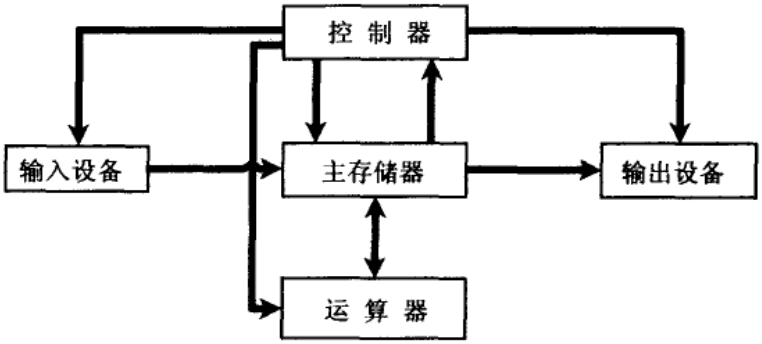
第一章

1. 什么是冯·诺依曼机？

答： 冯·诺依曼于 1945 年提出了存储程序的概念和二进制原理，利用这种概念和原理设计的电子计算机系统统称为冯·诺依曼机。

它包括运算器、控制器、存储器、输入设备和输出设备五个组成部分。

早期的冯·诺依曼机结构上以运算器和控制器为中心，随着计算机体系结构的发展，现在已演化为以存储器为中心的结构。



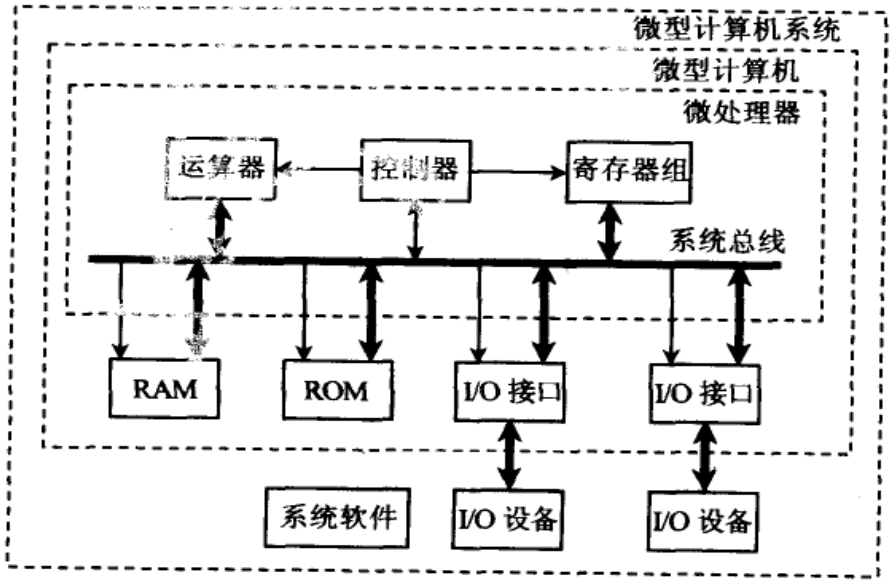
2. 微处理器，微型计算机，微型计算机系统有什么联系与区别？

答：微处理器是微型计算机系统的核心，也称为 CPU（中央处理器）。主要完成：①从存储器中取指令，指令译码；②简单的算术逻辑运算；③在处理器和存储器或者 I/O 之间传送数据；④程序流向控制等。

微型计算机由微处理器、存储器、输入/输出接口电路和系统总线组成。

以微型计算机为主体，配上外部输入/输出设备及系统软件就构成了微型计算机系统。

三者关系如下图：



3. 微处理器有哪些主要部件组成？其功能是什么？

答：微处理器是一个中央处理器，由算术逻辑部件 ALU、累加器和寄存器组、指令指针寄存器 IP、段寄存器、标志寄存器、时序和控制逻辑部件、内部总线等组成。

算术逻辑部件 ALU 主要完成算术运算及逻辑运算。

累加器和寄存器组包括数据寄存器和变址及指针寄存器，用来存放参加运算的数据、中间结果或地址。

指令指针寄存器 IP 存放要执行的下一条指令的偏移地址，顺序执行指令时，每取一条指令增加相应计数。

段寄存器存放存储单元的段地址，与偏移地址组成 20 位物理地址用来对存储器寻址。

标志寄存器 flags 存放算术与逻辑运算结果的状态。

时序和控制逻辑部件负责对整机的控制：包括从存储器中取指令，对指令进行译码和分析，发出相应的控制信号和时序，将控制信号和时序送到微型计算机的相应部件，使 CPU 内部及外部协调工作。

内部总线用于微处理器内部各部件之间进行数据传输的通道。

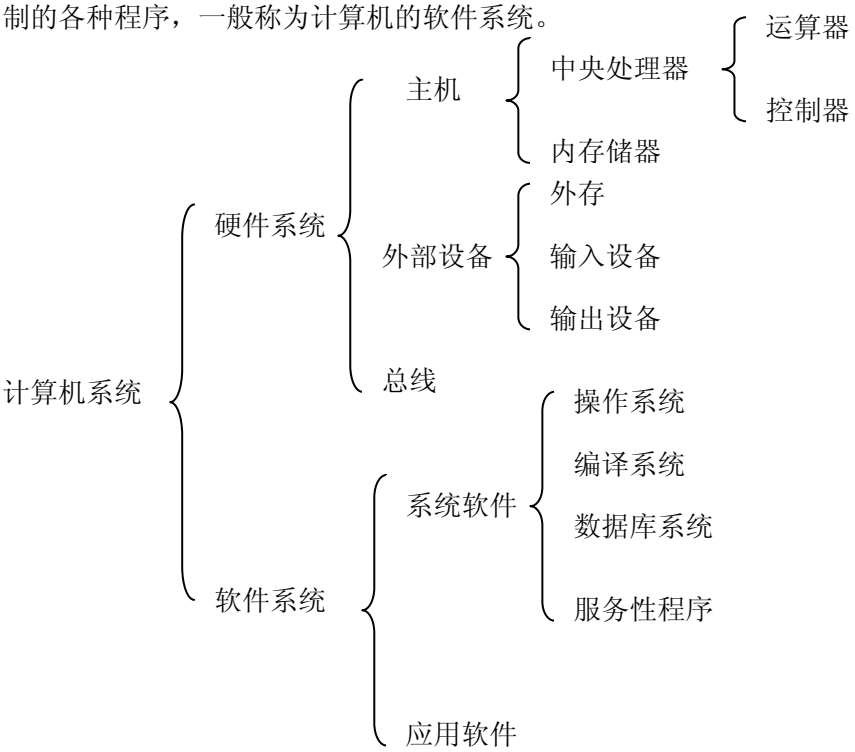
4. 画一个计算机系统的方框图，简述各部分主要功能。

答：计算机系统由硬件（Hardware）和软件（Software）两大部分组成。

硬件是指物理上存在的各种设备，如显示器、机箱、键盘、鼠标、硬盘和打印机等，是计算机进行工作的物质基础。

软件是指在硬件系统上运行的各种程序、数据及有关资料。

一个完整的计算机系统不仅应该具备齐全的基本硬件结构，还必须配备功能齐全的基本软件系统，后者是为了充分发挥基本硬件结构中各部分的功能和方便用户使用计算机而编制的各种程序，一般称为计算机的软件系统。



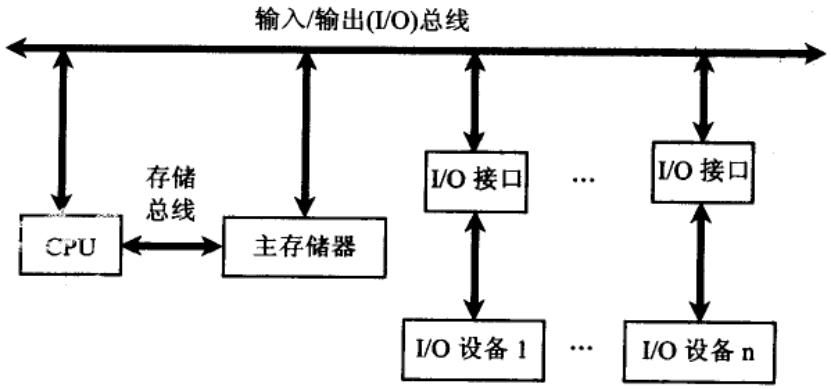
5. 列出计算机系统中的三种总线结构，画出面向存储器的双总线结构图。

答：（1）面向系统的单总线结构

（2）面向 CPU 的双总线结构

（3）面向主存储器的双总线结构

2010.05.09



6. 8086 微处理器可寻址多少字节存储器？Pentium II 微处理器可寻址多少字节存储器？

答：8086 微处理器有 20 条地址线，可寻址 2^{20}=1MB 存储器

Pentium II 处理器有 36 条地址线，可寻址 2^{36}=64GB 存储器

7. 什么是 PCI 总线？什么是 USB？

答：PCI 总线是微处理机机箱内的底板总线即系统总线的一种，是用来连接构成微处理机的各个插件板的一种数据传输标准。

PCI 全称为 Peripheral Component Interconnect，即外设互连局部总线，是 Intel 公司推出的 32/64 位标准总线。数据传输速率为 132MB/s，适用于 Pentium 微型机。PCI 总线是同步且独立于微处理器的，具有即插即用的特性，允许任何微处理器通过桥接口连接到 PCI 总线上。

USB总线，通用串行总线（Universal Serial Bus），属于外部总线的一种，用作微处理机系统与系统之间，系统与外部设备之间的信息通道。USB是在 1994 年底由英特尔、康柏、IBM、Microsoft等多家公司联合提出的，不过直到近期，才得到广泛应用，已成为目前电脑中的标准扩展接口。

USB 接口支持设备的即插即用和热插拔功能，具有传输速度快，使用方便，连接灵活，独立供电等优点。

8. 说明以下一些伪指令的作用。

（1）DB　（2）DQ　（3）DW　（4）DD

答：（1）在汇编语言中定义字节数据。

（2）在汇编语言中定义 4 字数据。

（3）在汇编语言中定义字数据。

（4）在汇编语言中定义双字数据。

9. 将下列二进制数转换为十进制数。

（1）1101.01B　（2）111001.0011B

（3）101011.0101B　（4）111.0001B

答：（1）13.25　（2）57.1875

（3）43.3125　（4）7.0625

10. 将下列十六进制数转换为十进制数。

（1）A3.3H　（2）129.CH

（3）AC.DCH　（4）FAB.3H

答：（1）163.1875　（2）297.75

（3）172.859375　（4）4011.1875

11. 将下列十进制数转换为二进制、八进制、十六进制。

（1）23　（2）107　（3）1238　（4）92

答：（1）23D=27Q=17H　　（2）107D=153Q=6BH
（3）1238D=2326Q=4D6H　（4）92D=134Q=5CH

12. 将下列十进制数转换为 8 位有符号二进制数。

（1）+32　　（2）-12　　（3）+100　　（4）-92

答：（1）$[+32]_{原}=[+32]_{反}=[+32]_{补}$=00100000B

（2）$[-12]_{原}$=10001100B　$[-12]_{反}$=11110011B　$[-12]_{补}$=11110100B

（3）$[+100]_{原}=[+100]_{反}=[+100]_{补}$=01100100B

（4）$[-92]_{原}$=11011100B　$[-92]_{反}$=10100011B　$[-92]_{补}$=10100100B

13. 将下列十进制数转换为压缩和非压缩格式的 BCD 码。

（1）102　　（2）44　　（3）301　　（4）1000

答：（1）$[102]_{压缩\ BCD}$=00000001 00000010B

$[102]_{非压缩\ BCD}$=00000001 00000000 00000010B

（2）$[44]_{压缩\ BCD}$=01000100B

$[44]_{非压缩\ BCD}$=00000100 00000100B

（3）$[301]_{压缩\ BCD}$=00000011 00000001B

$[301]_{非压缩\ BCD}$=00000011 00000000 00000001B

（4）$[1000]_{压缩\ BCD}$=00010000 00000000B

$[1000]_{非压缩\ BCD}$=00000001 00000000 00000000 00000000B

14. 将下列二进制数转换为有符号十进制数。

（1）10000000B　　（2）00110011B　　（3）10010010B　　（4）10001001B

答：（1）看作原码时真值为-0，看作反码时真值为-127，看作补码时真值为-128

（2）+51

（3）看作原码时真值为-18，看作反码时真值为-109，看作补码时真值为-110

（4）看作原码时真值为-9，看作反码时真值为-118，看作补码时真值为-119

15. 将下列十进制数转换为单精度浮点数。

（1）+1.5　　（2）-10.625　　（3）+100.25　　（4）-1200

答：（1）$+1.5=1.1B=1.1\times2^{0}$，符号为 0，移码阶 127+0=127=01111111B

尾数 1000000 00000000 00000000

故单精度浮点数为 0 01111111 1000000 00000000 00000000

（2）$-10.625=-1010.101B=-1.010101\times2^{3}$，符号为 1，

移码阶 127+3=130=10000010B，尾数 0101010 00000000 00000000

故单精度浮点数为 1 10000010 0101010 0000000 0000000

（3）$+100.25=1100100.01B=1.10010001\times2^{6}$，符号为 0

移码阶 127+6=133=10000101B，尾数 1001000 10000000 00000000

故单精度浮点数为 0 10000101 1001000 10000000 00000000

（4）$-1200=-10010110000B=-1.0010110000\times2^{10}$，符号为 1

移码阶 127+10=137=10001001B，尾数 0010110 00000000 00000000

故单精度浮点数为 1 10001001 0010110 00000000 00000000

16. 将下列单精度浮点数转换为十进制数。

（1）0 10000000 1100000 00000000 00000000

（2）1 01111111 0000000 00000000 00000000

（3）0 10000000 1001000 00000000 00000000

答：（1）$1.11\times2^{1}=11.1B=+3.5D$

（2）$-1.0\times2^{0}=-1B=-1D$

（3）1.1001×2^{1}=11.001B=+3.125D

第二章

1. 8086CPU 内部由哪两部分组成？它们的主要功能是什么？

答：8086CPU 内部由执行单元 EU 和总线接口单元 BIU 组成。

主要功能为：执行单元 EU 负责执行指令。它由算术逻辑单元(ALU)、通用寄存器组、16 位标志寄存器(FLAGS)、EU 控制电路等组成。EU 在工作时直接从指令流队列中取指令代码，对其译码后产生完成指令所需要的控制信息。数据在 ALU 中进行运算，运算结果的特征保留在标志寄存器 FLAGS 中。

总线接口单元 BIU 负责 CPU 与存储器和 I／O 接口之间的信息传送。它由段寄存器、指令指针寄存器、指令队列、地址加法器以及总线控制逻辑组成。

2. 8086CPU 中有哪些寄存器？各有什么用途？

答：8086CPU 内部包含 4 组 16 位寄存器，分别是通用寄存器组、指针和变址寄存器、段寄存器、指令指针和标志位寄存器。

（1）通用寄存器组　包含 4 个 16 位通用寄存器 AX、BX、CX、DX，用以存放普通数据或地址，也有其特殊用途。如 AX（AL）用于输入输出指令、乘除法指令，BX 在间接寻址中作基址寄存器，CX 在串操作和循环指令中作计数器，DX 用于乘除法指令等。

（2）指针和变址寄存器　BP、SP、SI 和 DI，在间接寻址中用于存放基址和偏移地址。

（3）段寄存器　CS、DS、SS、ES 存放代码段、数据段、堆栈段和附加段的段地址。

（4)指令指针寄存器IP　用来存放将要执行的下一条指令在现行代码段中的偏移地址。

（5）标志寄存器 Flags　用来存放运算结果的特征。

3. 8086CPU 和 8088CPU 的主要区别是什么？

答：8088CPU 的内部结构及外部引脚功能与 8086CPU 大部分相同，二者的主要不同之处如下：

（1）8088 指令队列长度是 4 个字节，8086 是 6 个字节。

（2）8088 的 BIU 内数据总线宽度是 8 位，而 EU 内数据总线宽度是 16 位，这样对 16 位数的存储器读/写操作需要两个读/写周期才能完成。8086 的 BIU 和 EU 内数据总线宽度都是 16 位。

（3）8088 外部数据总线只有 8 条 AD_7～AD_0，即内部是 16 位，对外是 8 位，故 8088 也称为准 16 位机。

（4）8088 中，用 $IO/\overline{M}$ 信号代替 $M/\overline{IO}$ 信号。

（5）8088 中，只能进行 8 位数据传输，$\overline{BHE}$ 不再需要，改为 $\overline{SS_0}$，与 $DT/\overline{R}$ 和 $IO/\overline{M}$ 一起决定最小模式中的总线周期操作。

4. 简要解释下列名词的意义：CPU，存储器，堆栈，IP，SP，BP，段寄存器，状态标志，控制标志，物理地址，逻辑地址，机器语言，汇编语言，指令，内部总线，系统总线。

答：CPU：中央处理器，是整个计算机系统的控制中心，主要功能是进行算术和逻辑运算，以及发出各种控制信号以协调整个系统正常工作。

存储器：是计算机系统中的记忆元件，用于存储指令和数据。

堆栈：在存储器中开辟的一个区域，用来存放需要暂时保存的数据。其操作特点是先进后出。

IP：指令指针寄存器，用来存放将要执行的下一条指令在现行代码段中的偏移地址。

SP：堆栈指针寄存器，用于指向当前栈顶单元。

BP：基址指针，间接寻址中用于存放基址，隐含段地址为 SS。

段寄存器：用于存放逻辑段的段地址。

状态标志：SF、ZF、OF、AF、PF 和 CF，反映运算结果的状态特征。

控制标志：IF、DF 和 TF，对可屏蔽中断、字符串操作指针变换方向和单步运行起控制作用。

物理地址：指存储器中存储单元的实际地址编码，是一种绝对地址，是 CPU 访问存储器的实际寻址地址，对于 8086 系统，地址范围为 00000H～FFFFFH。

逻辑地址：由段基址和偏移地址组成，均为无符号的 16 位二进制数，程序设计时采用逻辑地址，可由逻辑地址变换为物理地址，物理地址=段基址×16+偏移地址。

机器语言：直接用二进制代码指令表达的计算机语言，指令是用 0 和 1 组成的一串代码，计算机可以直接识别，不需要进行任何翻译。每台机器的指令，其格式和代码所代表的含义都是硬性规定的，故称之为面向机器的语言，也称为机器语言，是第一代计算机语言。

汇编语言：使用助记符表示的二进制代码指令语言，是一种符号化的机器语言，必须经编译程序将汇编语言编译成机器语言，计算机才能识别。

指令：能被计算机识别并执行的二进制代码，规定了计算机能完成的某一操作。

内部总线：微处理器内部各个部件之间传送信息的通道。

系统总线：微处理机机箱内的底板总线，用来连接构成微处理机的各个插件板，如 ISA 总线、EISA 总线、PCI 总线等。

5. 要完成下述运算或控制，用什么标志位判别？其值是什么？

（1）比较两数是否相等　　（2）两数运算后结果是正数还是负数

（3）两数相加后是否溢出　（4）采用偶校验方式，判定是否要补 1

（5）两数相减后比较大小　（6）中断信号能否允许

答：（1）ZF，两数相减，若 ZF=1，则相等。

（2）SF，SF=1 则为负，否则为正

（3）对有符号数：OF，OF=1 为溢出；对无符号数：CF，CF=1 为溢出

（4）PF，PF=1，不补 1

（5）对有符号数：无溢出时（OF=0），如 ZF=1，则两数相等；如 ZF=0 且 SF=0，则被减数大；如 ZF=0 且 SF=1，则减数大； 有溢出时（OF=1），如 SF=1，则被减数大；如 SF=0，则减数大； 对无符号数：如 ZF=1，则两数相等；如 CF=0，则被减数大；如 CF=1，则减数大

（6）IF，IF=1，允许中断

6. 8086 系统中存储器采用什么结构？用什么信号来选中存储体？

答：8086 系统中，存储器采用分体结构，1MB 的存储空间分成两个存储体：偶地址存储体和奇地址存储体，各为 512KB。

使用 A_0 和 $\overline{BHE}$ 来区分两个存储体。当 A_0=0 时，选中偶地址存储体，与数据总线低 8 位相连，从低 8 位数据总线读/写一个字节。

当 $\overline{BHE}$ =0 时，选中奇地址存储体，与数据总线高 8 位相连，从高 8 位数据总线读/写一个字节。

当 A_0=0，$\overline{BHE}$ =0 时，同时选中两个存储体，读/写一个字。

7. 用伪指令 DB 在存储器中存储 ASCII 码字符串“What time is it？”。并画出内存分布图。

答：　STR　DB　‘What time is it?’，’$’

STR	‘W’
	‘h’
	‘a’
	‘t’
	‘ ‘
	‘t’
	‘i’
	‘m’
	‘e’
	‘ ‘
	‘i’
	‘s’
	‘ ‘
	‘i’
	‘t’
	‘?’
	‘$’

8. 用伪指令将下列 16 位二进制数存储在存储器中，并画出内存分布图。

（1）1234H　　（2）A122H　　（3）B100H

答：　NUM　DW　1234H，A122H，B100H

NUM	34H
	12H
	22H
	A1H
	00H
	B1H

9. 段寄存器装入如下数据，写出每段的起始和结束地址。

（1）1000H　　（2）1234H　　（3）2300H　　（4）E000H　　（5）AB00H

答：（1）10000H~1FFFFH

（2）12340H~2233FH

（3）23000H~32FFFH

（4）E0000H~EFFFFH

（5）AB000H~BAFFFH

10. 根据下列 CS：IP 的组合，求出要执行的下一条指令的存储器地址。

（1）CS：IP=1000H：2000H　　（2）CS：IP=2000H：1000H

（3）CS：IP=1A00H：B000H　　（4）CS：IP=3456H：AB09H

答：（1）12000H　　（2）21000H　　（3）25000H　　（4）3F069H

11. 求下列寄存器组合所寻址的存储单元地址：

（1）DS=1000H，DI=2000H　　（2）SS=2300H，BP=3200H

（3）DS=A000H，BX=1000H　　（4）SS=2900H，SP=3A00H

答：（1）12000H　　（2）26200H　　（3）A1000H　　（4）2CA00H

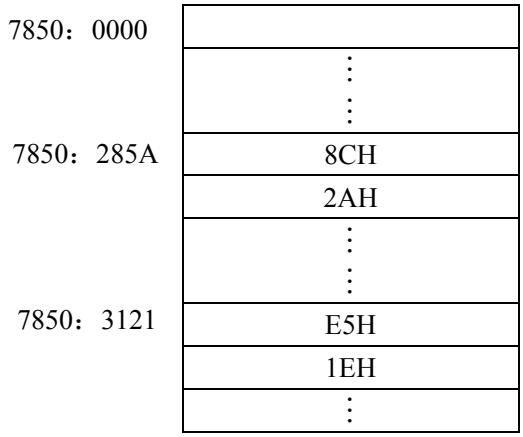
12. 若当前 SS=3500H，SP=0800H，说明堆栈段在存储器中的物理地址，若此时入栈 10 个字节，SP 内容是什么？若再出栈 6 个字节，SP 为什么值？

答：物理地址：35000H～35800H。入栈 10 个字节后 SP 为 7F6H。再出栈 6 个字节后 SP 为 7FCH。

13. 某程序数据段中存放了两个字，1EE5H 和 2A8CH，已知 DS=7850H，数据存放的偏移地址为 3121H 及 285AH。试画图说明它们在存储器中的存放情况。若要读取这两个字，需要对存储器进行几次操作？

答：1EE5H 的存储物理地址=78500H+3121H=7B621H，为奇数，故若要读取这个字，需要对存储器进行两次读操作。

2A8CH 的存储物理地址=78500H+285AH=7AD5AH，为偶数，故若要读取这个字，只需对存储器进行一次读操作。



14. 存储器中每段容量最多 64K 字节，若用 debug 调试程序中的 r 命令，在屏幕上有如下显示：

C：>debug

-r

AX=0000 BX=0000 CX=0079 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=10E4 ES=10F4 SS=21F0 CS=31FF IP=0100 NV UP DI PL NZ NA PO NC

（1）试画出此时存储器分段示意图

（2）写出状态标志 OF、SF、ZF、CF 的值

答：（1）代码段首地址：31FF0H　　当前指令地址：320F0H

数据段首地址：10E40H

堆栈段首地址：21F00H　　堆栈段栈顶地址：31EEEH

附加段首地址：10F40H

（2）OF=SF=ZF=CF=0

15. 说明 8086 系统中“最小模式”和“最大模式”两种工作方式的主要区别是什么？

答：为了便于组成不同规模的系统，在 8086 芯片中设计了两种工作模式，即最小模式和最大模式。

最小模式用于单机系统，系统中所需要的控制信号全部由 8086 直接提供；最大模式用于多处理机系统，系统中所需要的控制信号由总线控制器 8288 提供。

16. 8086 系统中为什么要用地址锁存器？8282 地址锁存器与 CPU 如何连接？

答：为了减少引脚的数量，8086CPU 的地址引脚和数据引脚分时复用，为了保证在总线操作周期中地址信号能有效而稳定的输出，必须使用地址锁存器。

由于 8086 有 20 条地址线，故需使用三片 8282，其中 $\overline{OE}$ 接地，STB 与 CPU 的 ALE 相连，前两片 8282 的 DI_0~DI_7 分别与 CPU 的 AD_0~AD_{15} 相连，第三片 8282 的 DI_0~DI_3 分别与 CPU 的 AD_{16}~AD_{19} 相连，DI_4 与 CPU 的 $\overline{BHE}$ 相连。

17. 哪个标志位控制 CPU 的 INTR 引脚？

答：IF，中断允许标志，IF=1 时，允许可屏蔽中断，IF=0 时，禁止可屏蔽中断。

18. 什么叫总线周期？在 CPU 读/写总线周期中，数据在哪个机器状态出现在数据总线上？

答：CPU 完成一次存储器访问或 I/O 端口操作所需要的时间称为一个总线周期，由几个 T 状态组成。在读/写总线周期中，数据在 T2～T4 状态出现在数据总线上。

19. 8284 时钟发生器共给出哪几个时钟信号？

答：OSC：振荡器输出信号，是内部振荡电路的 TTL 电平输出，其频率与晶振的频率相等，在 PC/XT 中，其频率为 14.318MHz

CLK：三分频 OSC 后的时钟，输出频率为 4.77MHz，占空比为 1/3，供 8086CPU 使用。

PCLK：二分频 CLK 后的时钟，输出频率为 2.38636MHz，TTL 电平，占空比为 1/2，供 PC/XT 机的外设使用。

20. 8086CPU 重新启动后，从何处开始执行指令？

答：重新启动后，CS=FFFFH，IP=0000H，故从物理地址为 FFFF0H 的位置开始执行指令。

21. 8086CPU 的最小模式系统配置包括哪几部分？

答：8086 最小模式系统配置包括：

8086CPU，存储器，I/O 接口芯片，1 片 8284 时钟发生器，3 片 8282 地址锁存器，2 片 8286 双向数据总线收发器。

第三章

1. 分别说明下列指令的源操作数和目的操作数各采用什么寻址方式。

（1）MOV　AX，2408H　　（2）MOV　CL，0FFH　　（3）MOV　BX，[SI]

（4）MOV　5[BX]，BL　　（5）MOV　[BP+100H]，AX　（6）MOV　[BX+DI]，’$’

（7）MOV　DX，ES：[BX+SI]　（8）MOV VAL[BP+DI]，DX

（9）IN　AL，05H　　（10）MOV　DS，AX

答：（1）立即数，寄存器　　（2）立即数，寄存器　　（3）寄存器间接，寄存器

（4）寄存器，寄存器相对　（5）寄存器，寄存器相对　（6）立即数，基址变址

（7）基址变址，寄存器　　（8）寄存器，相对基址变址

（9）直接端口寻址，寄存器　（10）寄存器，寄存器

2. 已知：DS=1000H，BX=0200H，SI=02H，内存 10200H～10205H 单元的内容分别为 10H，2AH，3CH，46H，59H，6BH。下列每条指令执行完后 AX 寄存器的内容各是什么？

（1）MOV　AX，0200H　　（2）MOV　AX，[200H]　　（3）MOV　AX，BX

（4）MOV　AX，3[BX]　　（5）MOV　AX，[BX+SI]　　（6）MOV　AX，2[BX+SI]

答：（1）0200H　（2）2A10H　（3）0200H

（4）5946H　（5）463CH　（6）6B59H

3. 设 DS=1000H，ES=2000H，SS=3500H，SI=00A0H，DI=0024H，BX=0100H，BP=0200H，数据段中变量名为 VAL 的偏移地址值为 0030H，试说明下列源操作数字段的寻址方式是什么？物理地址值是多少？

（1）MOV　AX，[100H]　　（2）MOV　AX，VAL　　（3）MOV　AX，[BX]

（4）MOV　AX，ES：[BX]　（5）MOV　AX，[SI]　　（6）MOV　AX，[BX+10H]

（7）MOV　AX，[BP]　（8）MOV　AX，VAL[BP][SI]

（9）MOV　AX，VAL[BX][DI]　　（10）MOV　AX，[BP][DI]

答：（1）直接，10100H　（2）直接，10030H　（3）寄存器间接,10100H

（4）寄存器间接，20100H　（5）寄存器间接,100A0H　（6）寄存器相对，10110H

（7）寄存器间接，35200H　（8）相对基址变址，352D0H

（9）相对基址变址，10154H　（10）基址变址，35224H

4. 写出下列指令的机器码

（1）MOV　AL，CL　　（2）MOV　DX，CX　　（3）MOV　[BX+100H]，3150H

答：（1）10001010 11000001B

（2）10001011 11010001B

（3）11000111 10000111 00000000 00000001 01010000 00110001B

5. 已知程序的数据段为：

```
DATA    SEGMENT
A       DB   ‘$’,10H
B       DB   ‘COMPUTER’
C       DW   1234H, 0FFH
D       DB   5   DUP(?)
E       DD   1200459AH
DATA    ENDS
```

求下列程序段执行后的结果是什么。

```
MOV   AL, A
MOV   DX, C
```

XCHG DL, A
MOV BX, OFFSET B
MOV CX, 3[BX]
LEA BX, D
LDS SI, E
LES DI, E

答：MOV AL, A　　AL=24H
MOV DX, C　　DX=1234H
XCHG DL, A　　DL=24H, A=34H
MOV BX, OFFSET B　　BX=2
MOV CX, 3[BX]　　CX=5550H
LEA BX, D　　BX=000EH
LDS SI, E　　DS=1200H, SI=459AH
LES DI, E　　ES=1200H, DI=459AH

6. 指出下列指令中哪些是错误的，错在什么地方。

（1）MOV DL，AX （2）MOV 8650H，AX （3）MOV DS，0200H
（4）MOV [BX]，[1200H] （5）MOV IP，0FFH （6）MOV [BX+SI+3]，IP
（7）MOV AX，[BX][BP] （8）MOV AL，ES：[BP] （9）MOV DL，[SI][DI]
（10）MOV AX，OFFSET 0A20H （11）MOV AL，OFFSET TABLE
（12）XCHG AL，50H （13）IN BL，05H （14）OUT AL，0FFEH

答：（1）长度不匹配 （2）立即数不能做目的操作数
（3）段寄存器不能用立即数赋值 （4）两个内存单元不能直接传送数据
（5）IP 不能用指令直接修改 （6）指令中不能出现 IP
（7）BX/BP 应与 SI/DI 搭配 （8）正确
（9）SI/DI 应与 BX/BP 搭配 （10）OFFSET 后应跟内存单元
（11）偏移地址为 16 位，AL 长度不够 （12）立即数不能用于 XCHG
（13）IN 必须用 AL/AX （14）操作数顺序反向；地址应为 8 位

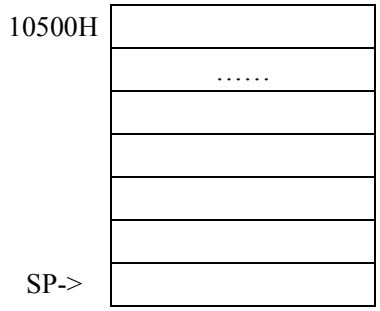
7. 已知当前数据段中有一个十进制数字 0～9 的 7 段代码表，其数值依次为 40H，79H，24H，30H，19H，12H，02H，78H，00H，18H。要求用 XLAT 指令将十进制数 57 转换成相应的 7 段代码值，存到 BX 寄存器中，试写出相应的程序段。

答：
```
DATA SEGMENT
TABLE DB 40H，79H，24H，30H，19H，12H，02H，78H，00H，18H
DATA ENDS
……
MOV AL，5
MOV BX，OFFSET TABLE
XLAT TABLE
MOV CL，AL
MOV AL，7
XLAT TABLE
MOV BL，AL
MOV BH，CL
```

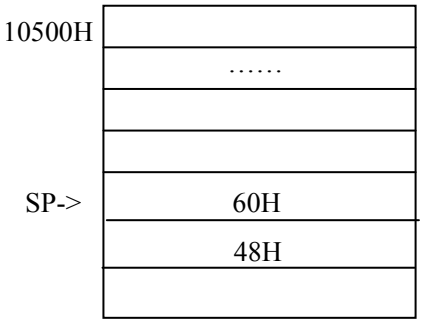
8. 已知当前 SS=1050H，SP=0100H，AX=4860H，BX=1287H，试用示意图表示执行下列指令过程中，堆栈中的内容和堆栈指针 SP 是怎样变化的。

PUSH AX
PUSH BX
POP BX
POP AX

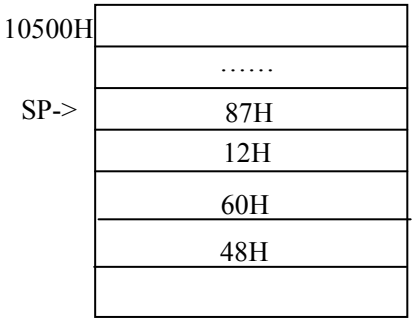
答：（1）指令执行前



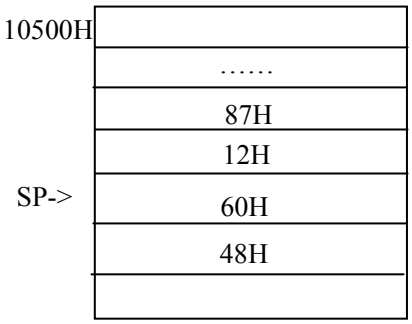
（2）执行 PUSH AX 后



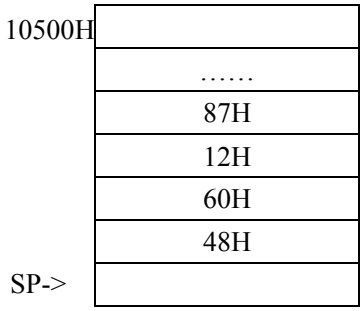
（3）执行 PUSH BX 后



（4）执行 POP BX 后



（5）执行 POP AX 后



9. 下列指令完成什么功能？

（1）ADD AL，DH （2）ADC BX，CX （3）SUB AX，2710H
（4）DEC BX （5）NEG CX （6）INC BL
（7）MUL BX （8）DIV CL

答：（1）AL+DH→AL （2）BX+CX+CF→BX （3）AX-2710H→AX
（4）BX-1→BX （5）0-CX→CX （6）BL+1→BL
（7）AX*BX→DX，AX （8）AX/CL 商→AL，余数→AH

10. 已知 AX=2508H，BX=0F36H，CX=0004H，DX=1864H，求下列每条指令执行后的结果是什么？标志位 CF 等于什么？

（1）AND AH，CL （2）OR BL，30H （3）NOT AX

（4）XOR　CX，0FFF0H　（5）TEST　DH，0FH　（6）CMP　CX，00H
（7）SHR　DX，CL　（8）SAR　AL，1　（9）SHL　BH，CL
（10）SAL　AX，1　（11）RCL　BX，1　（12）ROR　DX，CL
答：（1）AX=0408H，CF=0　（2）BX=0F36H，CF=0
（3）AX=0DAF7H，CF 不变　（4）CX=0FFF4H，CF=0
（5）DX=01864H，CF=0　（6）CX=0004H，CF=0
（7）DX=0186H，CF=0　（8）AX=2504H，CF=0
（9）BX=0F036H，CF=0　（10）AX=4A10H，CF=0
（11）BX=1E6C/1E6DH，CF=0　（12）DX=4186H，CF=0

11. 假设数据段定义如下：

```
DATA    SEGMENT
STRING    DB    'The Personal Computer & TV'
DATA    ENDS
```

试用字符串操作等指令编程完成以下功能：
（1）把该字符串传送到附加段中偏移量为 GET_CHAR 开始的内存单元中。
（2）比较该字符串是否与'The Computer'相同，若相同则将 AL 寄存器的内容置 1，否则置 0。并要求将比较次数送到 BL 寄存器中。
（3）检查该字符串是否有'&'，若有则用空格字符将其替换。
（4）把字符串大写字母传送到附加段中以 CAPS 开始的单元中，其余字符传到以 CHART 开始的单元中。然后将数据段中存储上述字符串的单元清 0。
答：（1）

```
        CLD
        LEA SI,STRING
        LEA DI,GET_CHAR
        MOV CX,26
        REP    MOVSB
```

（2）补充在附加段定义

```
        STRING1 DB 'The computer'
        MOV AL,1
        CLD
        LEA SI,STRING
        LEA DI,STRING1
        MOV CX,12
        REPZ CMPSB
        JZ SKIP
        MOV AL,0
    SKIP: MOV BX,12
        SUB BX,CX
```

（3）

```
        MOV AX, DATA
        MOV ES, AX
        CLD
        MOV AL,'&'
        LEA    DI,STRING
        MOV CX,26
 NEXT:    REPNE SCASB
```

```
        JZ    FIND
        JMP   EXIT
FIND:   MOV BYTE PTR [DI-1], '  '
        JCXZ  EXIT
        JMP   NEXT
EXIT:
```

（4）
```
        LEA SI,STRING
        LEA DI,CAPS
        LEA BX,CHART
        MOV CX,26
NEXT:   MOV AL, [SI]
        CMP AL,'A'
        JB OTHER
        CMP AL,'Z'
        JA OTHER
        MOV ES:[DI],AL
        INC DI
        JMP SKIP
OTHER:  MOV ES:[BX],AL
        INC BX
SKIP:   MOV BYTE PTR[SI],0
        INC SI
        LOOP NEXT
```

12. 编程将 AX 寄存器中的内容以相反的顺序传送到 DX 寄存器中，并要求 AX 中的内容不被破坏，然后统计 DX 寄存器中‘1’的个数是多少。

答：：
```
        MOV BL,0
        PUSH AX
        MOV DX,0
        MOV CX,16
NEXT: SHL AX,1
        JNC SKIP
        INC BL
SKIP:   RCR DX,1
        LOOP NEXT
STOP:   POP AX
```

13. 设 CS=1200H，IP=0100H，SS=5000H，SP=0400H，DS=2000H，SI=3000H，BX=0300H，（20300H）=4800H，（20302H）=00FFH，TABLE=0500H，PROG_N 标号的地址为 1200：0278H，PROG_F 标号的地址为 3400H：0ABCH。说明下列每条指令执行完后，程序将分别转移到何处执行？

（1）JMP　PROG_N

（2）JMP　BX

（3）JMP　[BX]

（4）JMP　FAR PROG_F

（5）JMP　DWORD PTR [BX]

如将上述指令中的操作码 JMP 改成 CALL，则每条指令执行完后，程序转向何处执行？并请画图说明堆栈中的内容和堆栈指针如何变化。

答：（1）1200H：0278H

（2）1200H：0300H

（3）1200H：4800H

（4）3400H：0ABCH

（5）00FFH：4800H

将操作码 JMP 改成 CALL 后：

（1）1200H：0278H

指令长度为 3，则 IP=0100H+3=0103H，入栈

5000H：0000H		
	……	
5000H：03FEH	03H	←SP
	01H	
5000H：0400H		

（2）1200H：0300H

指令长度为 2，则 IP=0100H+2=0102H，入栈

5000H：0000H		
	……	
5000H：03FEH	02H	←SP
	01H	
5000H：0400H		

（3）1200H：4800H

指令长度为 2，则 IP=0100H+2=0102H，入栈

5000H：0000H		
	……	
5000H：03FEH	02H	←SP
	01H	
5000H：0400H		

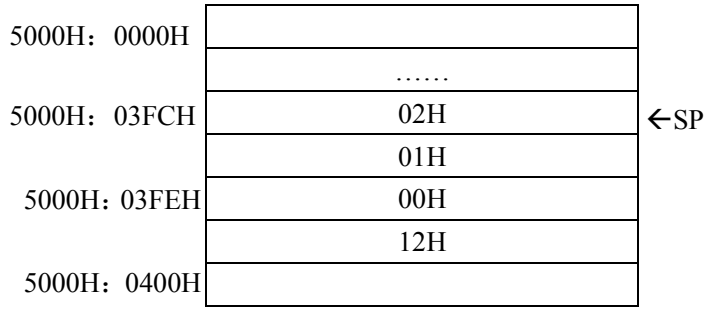
（4）3400H：0ABCH

指令长度为 5，则 IP=0100H+5=0105H，入栈

5000H：0000H		
	……	
5000H：03FCH	05H	←SP
	01H	
5000H：03FEH	00H	
	12H	
5000H：0400H		

（5）00FFH：4800H

指令长度为 2，则 IP=0100H+2=0102H，入栈



14. 在下列程序段括号中分别填入以下指令

（1）LOOP NEXT　　（2） LOOPE NEXT　　（3） LOOPNE NEXT

试说明在这三种情况下，程序段执行完后，AX，BX，CX，DX 的内容分别是什么。

```
START: MOV AX,01H
       MOV BX,02H
       MOV DX,03H
       MOV CX,04H
NEXT:  INC  AX
       ADD  BX,AX
       SHR  DX,1
       (   )
```

答：（1） AX=05H　BX=10H　CX=0　　DX=0

（2） AX=02H　BX=04H　CX=03H　DX=01H

（3） AX=03H　BX=07H　CX=02H　DX=0

15. 某班有 7 个同学英语成绩低于 80 分，分数存在 ARRAY 数组中，试编程完成以下工作：

（1）给每人加 5 分，结果存到 NEW 数组中

（2）把总分存到 SUM 单元中

答：

```
        DATA   SEGMENT
        ARRAY  DB  ？，？，？，？，？，？，？
        NEW    DB  7  DUP(?)
        SUM    DW  0
        DATA   ENDS

        MOV  AX，DATA
        MOV  DS, AX
（1）   LEA  SI,ARRAY
        LEA  DI,NEW
        MOV CX,7
NEXT:   MOV  AL，[SI]
        ADD  AL, 5
        MOV  [DI], AL
        INC  SI
        INC  DI
        LOOP  NEXT
（2）   LEA  SI，ARRAY
        CLC
```

```
        MOV    CX, 7
NEXT1：MOV    AL，[SI]
        MOV    AH, 0
        ADC    SUM, AX
        INC    SI
        LOOP   NEXT1
```

16. 中断向量表的作用是什么？它放在内存的什么区域内？中断向量表中的什么地址用于类型 3 的中断？

答：中断向量表用来存放中断服务程序的入口地址。8086 的 256 类中断的入口地址要占用 1K 字节，位于内存 00000H～003FFH 的区域内。中断向量表中 0000CH～0000FH 用于类型 3 的中断。

17. 设类型 2 的中断服务程序的起始地址为 0485：0016H，它在中断向量表中如何存放？

答：物理地址　内容

00008H　16H

00009H　00H

0000AH　85H

0000BH　04H

18. 若中断向量表中地址为 0040H 中存放 240BH，0042H 单元里存放的是 D169H，试问：

（1）这些单元对应的中断类型是什么？

（2）该中断服务程序的起始地址是什么？

答：（1）10H

（2）D169H:240BH

19. 简要说明 8086 响应类型 0～4 中断的条件是什么？

答：类型 0：除法错中断

执行除法指令时，若除数为 0 或所得商超过寄存器能容纳的范围，则自动产生类型 0 中断。

类型 1：单步中断

若 CPU 的单步标志 TF=1，则每执行完一条指令后，自动产生类型 1 中断。

类型 2：不可屏蔽中断 NMI

当 8086 的 NMI 引脚上接收到由低变高的电平变化时，将自动产生类型 2 中断。

类型 3：断点中断

若在程序中某位置设置断点，当程序执行到该断点时，则产生类型 3 中断。

类型 4：溢出中断

若溢出标志 OF=1，则可由溢出中断指令 INTO 产生类型 4 中断。

第四章

1. 下列变量各占多少字节？

```
A1   DW   23H，5876H
A2   DB   3 DUP（？），0AH，0DH, ‘$’
A3   DD   5 DUP(1234H, 567890H)
A4   DB   4 DUP(3 DUP(1, 2,’ABC’))
```

答：A1 占 4 个字节
A2 占 6 个字节
A3 占 40 个字节
A4 占 60 个字节

2. 下列指令完成什么功能？

```
MOV   AX，00FFH AND 1122H+3344H
MOV   AL，15 GE 1111B
MOV   AX，00FFH LE 255+6/5
AND   AL, 50 MOD 4
OR    AX, 0F00FH AND 1234 OR 00FFH
```

答：（1）将 0066H 传送给 AX
（2）将 0FFH 传送给 AL
（3）将 0FFFFH 传送给 AX
（4）AND AL，02H
（5）OR AX，00FFH

3. 有符号定义语句如下：

```
BUF   DB   3,4,5,’123’
ABUF DB   0
L     EQU   ABUF-BUF
```

求 L 的值为多少？
答：L=6

4. 假设程序中的数据定义如下：

```
PAR       DW   ?
PNAME   DB   16 DUP(?)
COUNT   DD   ?
PLENTH   EQU   $-PAR
```

求 PLENTH 的值为多少？表示什么意义？
答：PAR 的偏移地址为 0，PLENTH 当前偏移地址$=2+16+4=22， $-PAR=22，故 PLENTH 的值为 22。

若在 PLENTH 所在行有变量定义，则$表示该变量的偏移地址，即$表示 PLENTH 所在行的当前偏移地址。故 PLENTH 表示从当前行到 PAR 之间定义的变量所占的字节个数。

5. 对于下面的数据定义，各条 MOV 指令执行后，有关寄存器的内容是什么？

```
DA1   DB   ?
DA2   DW   10 DUP(?)
DA3   DB   ‘ABCD’
      MOV   AX, TYPE DA1
      MOV   BX, SIZE DA2
```

```
                MOV    CX, LENGTH DA3
```
答：AX=1,BX=20,CX=1

6. 下段程序完成后，AH 等于什么？

```
        IN      AL, 5FH
        TEST    AL, 80H
        JZ      L1
        MOV     AH, 0
        JMP     STOP
L1:     MOV     AH, 0FFH
STOP:   HLT
```

答：讨论从端口 5FH 输入的数据最高位的情况。若最高位为 1，则 AH=0；若最高位为 0，则 AH=0FFH。

7. 编程序完成下列功能：

（1）利用中断调用产生 5 秒延时。

（2）利用中断调用，在屏幕上显示 1～9 之间随机数。

答：（1）可以利用中断类型 1CH 来处理，因为在系统时钟的中断处理程序中，时钟中断一次要调用一次 INT 1CH，即每隔 55ms，产生一次 1CH 中断，要产生 5 秒延时，只要中断 5s/55ms=91 次即可。又因 1CH 中断处理程序中只有一条 IRET 指令，故可将用户的程序代替原有的 INT 1CH 程序。

```
DATA    SEGMENT
COUNT   DW   91              ;计数器
MESS    DB   '5 s delayed!',0AH,0DH,'$'
DATA    ENDS
CODE    SEGMENT
MAIN    PROC   FAR
  ASSUME   CS:CODE,DS:DATA,ES:DATA
START:  PUSH   DS
        MOV AX,0
        PUSH AX
        MOV AX,DATA
        MOV DS,AX
        MOV AL,1CH          ;得到原中断向量
        MOV AH,35H
        INT 21H
        PUSH ES             ;存储原中断向量
        PUSH BX
        PUSH DS
        MOV DX,OFFSET DELAY     ；DELAY 的偏移地址和段地址
        MOV AX,SEG DELAY
        MOV DS,AX
        MOV AL,1CH          ；设置中断向量
        MOV AH,25H
        INT 21H
        POP DS
```

```
            IN     AL,21H            ；设置中断屏蔽位
            AND AL,0FEH
            OUT 21H,AL
            STI
            MOV DI,2000H              ；主程序延迟，在执行此段程序期间
A1:         MOV SI,3000H              ；产生中断
A2:         DEC SI
            JNZ A2
            DEC DI
            JNZ A1
            POP DX                    ；取原中断向量
            POP DS
            MOV AL,1CH
            MOV AH,25H
            INT 21H
            RET
MAIN     ENDP
DELAY    PROC    NEAR
            PUSH DS
            PUSH AX
            PUSH CX
            PUSH DX
            MOV AX,DATA
            MOV DS,AX
            STI
            DEC COUNT                 ；5 秒计数
            JNZ EXIT
            MOV DX,OFFSET MESS        ；显示信息
            MOV AH,09H
            INT 21H
            MOV COUNT,91              ；5 秒的值
EXIT:       CLI
            POP DX
            POP CX
            POP AX
            POP DS
            IRET
DELAY    ENDP
CODE    ENDS
        END START
```

（2）可以利用 INT 1AH，读取当前时间的 1/100 秒为随机数。

```
CODE    SEGMENT
  ASSUME CS:CODE
START：MOV AH,0
```

```
        INT 1AH              ；读取当前时间 CH：CL=时：分
        MOV AL,DL            ；DH：DL=秒：1/100 秒
        MOV AH,0
        MOV BL,9
        DIV BL
        INC AH
        MOV DL,AH
        ADD DL,30H
        MOV AH,2
        INT 21H
        MOV AH,4CH
        INT 21H
CODE    ENDS
    END    START
```

8. 编两个通过过程完成将 AX 中存放的二进制数转换成压缩型 BCD 码以及将 BCD 码转换成二进制数。

答：（1）将 AX 中的二进制数先后除以 1000，100 和 10，每次除法所得的商，即是 BCD 数的千位、百位和十位数，余数是个位数。

子程序名：B2TOBCD

输入参数：AX=十六位二进制数

输出参数：CF=0，则 AX=4 位压缩型 BCD 码。CF=1，则要转换的数大于 9999，AX 不变。

使用寄存器：CX：存放除数，DX：存放中间结果。

```
B2TOBCD   PROC    FAR
          CMP    AX,9999          ；AX>9999，则 CF 置 1
          JBE     TRAN
          STC
          JMP    EXIT
TRAN:     PUSH    CX
          PUSH    DX
          SUB    DX,DX             ；DX 清 0
          MOV    CX,1000           ；计算千位数
          DIV    CX
          XCHG    AX,DX            ；商在 DX 中，余数在 AX 中
          MOV    CL,4
          SHL    DX,CL             ；DX 左移 4 位
          MOV    CL,100            ；计算百位数
          DIV    CL
          ADD    DL,AL             ；百位数加到 DL 中，DX 左移 4 位
          MOV    CL,4
          SHL    DX,CL
          XCHG    AL,AH            ；余数保留在 AL 中
          SUB    AH,AH
```

```
        MOV   CL,10              ；计算十位数
        DIV   CL
        ADD   DL,AL              ；十位数加到 DL 中，DX 左移 4 位
        MOV   CL,4
        SHL   DX,CL
        ADD   DL,AH              ；加个位数
        MOV   AX,DX              ；结果送到 AX 中
        POP   DX
        POP   CX
EXIT:   RET
B2TOBCD ENDP
```

（2）将 AX 中 4 位 BCD 码顺序乘以 1000，100，10 和 1，然后求和即得。

子程序名：BCDTOB2

输入参数：AX=4 位压缩 BCD 码

输出参数：AX=十六位二进制数

使用寄存器：BX：暂存数据，CX：存放乘数，DX：存放中间结果

```
BCDTOB2 PROC  FAR
        PUSH  BX
        PUSH  CX
        PUSH  DX
        MOV   BX,AX
        MOV   CL,4
        ROL   AX,CL
        AND   AX,000FH
        MOV   CX,1000            ；乘以 1000
        MUL   CX
        MOV   DX,AX
        MOV   AX,BX
        MOV   CL,8
        ROL   AX,CL
        AND   AX,000FH
        MOV   CL,100             ；乘以 100
        MUL   CL
        ADD   DX,AX
        MOV   AX,BX
        MOV   CL,4
        SHR   AX,CL
        AND   AX,000FH
        MOV   CL,10              ；乘以 10
        MUL   CL
        ADD   DX,AX
        AND   BX,000FH
        ADD   DX,BX
        MOV   AX,DX
```

```
        POP   DX
        POP   CX
        POP   BX
        RET
BCDTOB2  ENDP
```

9. 编写两个通用过程，一个完成 ASCII 码转换成二进制数功能，另一个完成 ASCII 字符显示输出功能。

答：（1）将 AX 中两位 ASCII 码先转化成数字，然后 AH*10+AL。

子程序名：ASCIITOB2

输入参数：AX=2 位 ASCII 码

输出参数：AX=转换后二进制数

使用寄存器：BX，CX，DX

```
ASCIITOB2  PROC   FAR
           PUSH   BX
           PUSH   CX
           PUSH   DX
           MOV    BX,AX
           SUB    AH,30H
           MOV    AL,AH
           MOV    CL,10          ；乘以 10
           MUL    CL
           MOV    DX,AX
           MOV    AX,BX
           SUB    AL,30H
           MOV    AH,0
           ADD    AX,DX
           POP    DX
           POP    CX
           POP    BX
           RET
ASCIITOB2  ENDP
```

（2）使用 2 号 DOS 功能调用显示字符。

子程序名：DISPLAY

输入参数：AX=2 位 ASCII 码

输出参数：无

使用寄存器：BX，DX

```
DISPLAY  PROC   FAR
         PUSH   BX
         PUSH   DX
         MOV    BX,AX
         MOV    DL,AH
         MOV    AH,2
         INT 21H
         MOV    AX,BX
```

```
        MOV   DL,AL
        MOV   AH,2
        INT 21H
        POP   DX
        POP   BX
        RET
DISPLAY ENDP
```

10. 编制两个通用过程，完成十六进制数转换成 ASCII 码并将 ASCII 码字符显示。

答：（1）子程序名：HEXTOASC

输入参数：AX：4 位十六进制数

输出参数：DX,AX：4 位 ASCII 码,DX 高 2 位，AX 低 2 位

使用寄存器：BX,CX,DX

```
HEXTOASC  PROC   FAR
          PUSH   BX
          PUSH   CX
          PUSH   DX
          MOV    BX,AX
          MOV    CL,4            ；转换 AH 高 4 位
          SHR    AX,CL
          AND    AX,0F00H
          ADD    AH,30H
          CMP    AH,3AH
          JB     A1
          ADD    AH,7
A1:       MOV    DH,AH
          MOV    AX,BX           ；转换 AH 低 4 位
          AND    AX,0F00H
          ADD    AH,30H
          CMP    AH,3AH
          JB     A2
          ADD    AH,7
A2:       MOV    DL,AH
          MOV    BH,BL           ；转换 AL 高 4 位
          MOV    CL,4
          SHR    BL,CL
          ADD    BL,30H
          CMP    BL,3AH
          JB     A3
          ADD    BL,7
A3:       MOV    AH,BL
          AND    BH,0FH          ；转换 AL 低 4 位
          ADD    BH,30H
          CMP    BH,3AH
          JB     A4
```

```
                ADD    BH,7
        A4:     MOV    AL,BH
                POP    DX
                POP    CX
                POP    BX
                RET
    HEXTOASC    ENDP
```

（2）子程序名：DISPLAYASC

输入参数：DX，AX：4 位 ASCII 码

输出参数：无

使用寄存器：BX,CX

```
DISPLAYASC  PROC    FAR
            PUSH    BX
            PUSH    CX
            MOV     BX,DX
            MOV     CX,AX
            MOV     DL,BH
            MOV     AH,02H
            INT     21H
            MOV     DL,BL
            INT     21H
            MOV     DL,CH
            INT     21H
            MOV     DL,CL
            INT     21H
            MOV     DL,'H'
            INT     21H
            POP     CX
            POP     BX
            RET
DISPLAYASC  ENDP
```

11. 某程序可从键盘接收命令（0～5），分别转向 6 个子程序，子程序入口地址分别为 P0～P5，编制程序，用跳转表实现分支结构。

答：

```
        DATA    SEGMENT
        TABLE   DW      6 DUP(?)
        DATA    ENDS
        CODE    SEGMENT
          ASSUME    CS:CODE,DS:DATA
        START:  MOV AX,DATA
                MOV DS,AX
                LEA SI, TABLE
                MOV WORD PTR[SI],OFFSET P0
                MOV WORD PTR[SI+2],OFFSET P1
                MOV WORD PTR[SI+4],OFFSET P2
```

```
        MOV WORD PTR[SI+6],OFFSET P3
        MOV WORD PTR[SI+8],OFFSET P4
        MOV WORD PTR[SI+10],OFFSET P5
        MOV AH,1
        INT 21H
        SUB AL,30H
        SHL AL,1
        MOV AH,0
        MOV SI,AX
        JMP TABLE[SI]
P0:
P1：
P2:
P3:
P4:
P5:
        MOV AH,4CH
        INT 21H
CODE    ENDS
    END START
```

12. 在首地址为 TABLE 的数组中按递增次序存放着 100 个 16 位补码数，编写一个程序，把出现次数最多的数及其出现次数分别存放于 AX 和 BL 中。

答：

```
DATA    SEGMENT
TABLE    DW 100 DUP (?) ； 数组中的数据是按增序排列的
NUM      DW ?
COUNT    DW 0
DATA    ENDS
CODE    SEGMENT
MAIN    PROC    FAR
   ASSUME CS: CODE, DS: DATA
START:    PUSH DS ； 设置返回 DOS
          SUB AX, AX
          PUSH AX
          MOV AX, DATA
          MOV DS, AX ； 给 DS 赋值
BEGIN: MOV CX, 100 ； 循环计数器
       MOV SI, 0
NEXT:    MOV DX, 0
       MOV AX, TABLE[SI]
COMP: CMP TABLE [SI], AX ； 计算一个数的出现次数
       JNE ADDR
       INC DX
       ADD SI, 2
       LOOP COMP
```

```
        ADDR: CMP DX, COUNT ； 此数出现的次数最多吗？
              JLE DONE
              MOV COUNT, DX ； 目前此数出现的次数最多，记下次数
              MOV NUM, AX ； 记下此数
        DONE: LOOP NEXT ； 准备取下一个数
              MOV CX, COUNT ； 出现最多的次数存入(CX)
              MOV AX, NUM ； 出现最多的数存入(AX)
              RET
        MAIN  ENDP
        CODE  ENDS ； 以上定义代码段
          END START
```

13. 将键盘上输入的十六进制数转换成十进制数，在屏幕上显示。

答：

```
DATA SEGMENT
STRING DB 'INPUT 4 HEX NUM:',0AH,0DH,'$'
NUM    DB 10 DUP(?)
DATA   ENDS
CODE   SEGMENT
 ASSUME CS:CODE,DS:DATA
START: MOV AX,DATA
       MOV DS,AX
       LEA DX,STRING      ； 显示提示信息
       MOV AH,9
       INT 21H
       MOV BP,4
       MOV DX,0
 CONT: MOV CL,4           ； 输入 4 位十六进制数→DX
       SHL DX,CL
 D1:   MOV AH,1
       INT 21H
       CMP AL,'0'
       JB  D1
       CMP AL,'F'
       JA  D1
       CMP AL,'A'
       JB  A1
       SUB AL,7
 A1:   SUB AL,30H
       MOV AH,0
       ADD DX,AX
       DEC BP
       JNZ CONT
       MOV SI,0          ； 将 DX 转换成十进制数，再转换成 ASCII 码—>NUM
       MOV AX,DX
       MOV DX,0
```

```
        MOV BX,10
D2:     DIV BX
        ADD DL,30H
        MOV NUM[SI],DL
        INC SI
        CMP AX,0
        JZ  EXIT
        MOV DX,0
        JMP D2
EXIT:   MOV DL,0AH        ；显示十进制数
        MOV AH,2
        INT 21H
        MOV DL,0DH
        INT 21H
D3:     DEC SI
        MOV DL,NUM[SI]
        MOV AH,2
        INT 21H
        JNZ D3
        MOV AH,4CH
        INT 21H
CODE ENDS
  END START
```

14. 将 AX 中的无符号二进制数转换成 ASCII 字符串表示的十进制数。

答：将 13 题输入过程和输出过程去掉即得。

```
DATA SEGMENT
NUM     DB 10 DUP(?)        ；转换后 ASCII 码按倒序存放于 NUM
DATA    ENDS
CODE    SEGMENT
 ASSUME CS:CODE,DS:DATA
START: MOV AX,DATA
        MOV DS,AX
        MOV AX, ××××  ；无符号二进制数
        MOV SI,0            ；将 AX 转换成十进制数，再转换成 ASCII 码—>NUM
        MOV DX,0
        MOV BX,10
D2:     DIV BX
        ADD DL,30H
        MOV NUM[SI],DL
        INC SI
        CMP AX,0
        JZ  EXIT
        MOV DX,0
        JMP D2
```

```
EXIT:   MOV AH,4CH
        INT 21H
CODE ENDS
  END START
```

15. 从键盘输入 20 个有符号数，将它们排序并在屏幕上显示。

答：

```
DATA SEGMENT
NUM   DW 20 DUP(?)            ；存放 20 个有符号数
ASC   DB 10 DUP(?)            ；输出时暂时保存每一个数的 ASCII 码
DATA ENDS
CODE SEGMENT
 ASSUME CS:CODE,DS:DATA
START:  MOV AX,DATA
        MOV DS,AX
        MOV CX,20
        MOV BX,10
        MOV SI,0
        MOV BP,0
BEGIN:  MOV DX,0              ；输入 20 个有符号数（十进制），并
A1:     MOV AH,1              ；转化为二进制数存放于 NUM 数组
        INT 21H
        CMP AL,' '            ；判断是否空格
        JZ   A2
        CMP AL,0DH            ；是否回车符
        JZ   A2
        CMP AL,'-'            ；是否‘-’
        JZ   A3
        JMP A4
  A3:   MOV BP,1
        JMP A1
  A4:   PUSH AX
        MOV AX,DX
        MUL BX
        MOV DX,AX
        POP AX
        SUB AL,30H
        MOV AH,0
        ADD DX,AX
        JMP A1
A2:     CMP BP,1
        JNZ A5
        NEG DX                ；若为负数，则取负
A5:     MOV NUM[SI],DX
        MOV BP,0
        ADD SI,2
```

```
        LOOP BEGIN

        MOV DL,0DH          ；回车换行
        MOV AH,2
        INT 21H
        MOV DL,0AH
        INT 21H

        MOV BX,0            ；对 20 个有符号数按由小到大顺序排序
        MOV CX,19           ；采用冒泡法，排序后依然存放于 NUM 数组
L1:     MOV DX,CX
L2:     MOV AX,NUM[BX]
        CMP AX,NUM[BX+2]
        JLE CONT1
        XCHG AX,NUM[BX+2]
        MOV NUM[BX],AX
CONT1: ADD BX,2
        LOOP L2
        MOV CX,DX
        MOV BX,0
        LOOP L1

        MOV CX,20           ；将 20 个有符号数（二进制）转换为十进制数
        MOV SI,0            ；再转换为 ASCII 码并输出屏幕
D1:     MOV AX,NUM[SI]
        ADD SI,2
        TEST AX,1000H
        JZ   D4
        PUSH AX
        MOV DL,'-'
        MOV AH,2
        INT 21H
        POP AX
        NEG AX
D4:     MOV DI,0
        MOV DX,0
        MOV BX,10
D2:     DIV BX
        ADD DL,30H
        MOV ASC[DI],DL
        INC DI
        CMP AX,0
        JZ   D3
        MOV DX,0
```

```
        JMP D2
D3:     MOV DL,ASC[DI-1]
        MOV AH,2
        INT 21H
        DEC DI
        JNZ D3
        MOV DL,' '
        MOV AH,2
        INT 21H
        LOOP D1

        MOV AH,4CH      ；返回 DOS
        INT 21H
CODE ENDS
 END START
```

16. 编写多字节有符号数的加法程序，从键盘接收两个加数，在屏幕上显示结果。

答：

```
DATA SEGMENT
STRING1 DB 'INPUT FIRST NUM(HEX):',0DH,0AH,'$'
STRING2 DB 'INPUT SECOND NUM(HEX):',0DH,0AH,'$'
STRING3 DB 'THE RESULT IS(HEX):',0DH,0AH,'$'
NUM1 DW 0,0                 ；存放加数 1
NUM2 DW 0,0                 ；存放加数 2
RESULT DB 0,0,0,0           ；存放结果
DATA ENDS
CODE SEGMENT
   ASSUME CS:CODE,DS:DATA
START:   MOV AX,DATA
        MOV DS,AX
        LEA DX,STRING1      ；输入第一个加数（4 位十六进制数）
        MOV AH,9
        INT 21H
        MOV SI,2
        MOV CX,8
 CONT:   PUSH CX
        CMP CX,4
        JNZ   B1
        SUB SI,2
  B1:   MOV CL,4
        SHL NUM1[SI],CL
  C1:   MOV AH,1
        INT 21H
        CMP AL,'0'
        JB   C1
        CMP AL,'F'
```

```
        JA   C1
        CMP AL,'A'
        JB   A1
        SUB AL,7
A1:     SUB AL,30H
        MOV AH,0
        ADD NUM1[SI],AX
        POP CX
        LOOP CONT

        MOV DL,0DH          ；回车换行
        MOV AH,2
        INT 21H
        MOV DL,0AH
        INT 21H

        LEA DX,STRING2      ；输入第二个加数（4 位十六进制数）
        MOV AH,9
        INT 21H
        MOV SI,2
        MOV CX,8
CONT1:  PUSH CX
        CMP CX,4
        JNZ  B2
        SUB SI,2
B2:     MOV CL,4
        SHL NUM2[SI],CL
C2:     MOV AH,1
        INT 21H
        CMP AL,'0'
        JB   C2
        CMP AL,'F'
        JA   C2
        CMP AL,'A'
        JB   A2
        SUB AL,7
A2:     SUB AL,30H
        MOV AH,0
        ADD NUM2[SI],AX
        POP CX
        LOOP CONT1
        MOV DL,0DH          ；回车换行
        MOV AH,2
        INT 21H
```

```
        MOV DL,0AH
        INT 21H

        LEA SI,NUM1             ；两数相加
        LEA BX,NUM2
        LEA DI,RESULT
        MOV CX,4
        CLC
AD:     MOV AL,[SI]
        ADC AL,[BX]
        MOV [DI],AL
        INC SI
        INC BX
        INC DI
        LOOP AD

        LEA DX,STRING3          ；显示结果（4 位十六进制数）
        MOV AH,9
        INT 21H
        MOV CX,4
        MOV DI,3
TT:     PUSH CX
        MOV DL,RESULT[DI]
        MOV CL,4
        SHR DL,CL
        ADD DL,30H
        CMP DL,3AH
        JB  D1
        ADD DL,7
D1:     MOV AH,2
        INT 21H
        MOV DL,RESULT[DI]
        AND DL,0FH
        ADD DL,30H
        CMP DL,3AH
        JB  D2
        ADD DL,7
D2:     MOV AH,2
        INT 21H
        DEC DI
        POP CX
        LOOP TT

        MOV AH,4CH          ；返回 DOS
```

```
        INT 21H
CODE ENDS
  END START
```

17. 编写 2 位非压缩型 BCD 码相乘的程序。

答：转化成加法进行累加运算。

```
DATA SEGMENT
DA1    DB 09H,09H
DA2    DB 09H,08H
RESULT DB 4 DUP(0)
DATA ENDS
CODE SEGMENT
  ASSUME CS:CODE,DS:DATA
START:   MOV AX,DATA
         MOV DS,AX
         MOV AL,DA2+1     ；计算加法次数
         MOV BL,10
         MUL BL
         MOV BL,DA2
         MOV BH,0
         ADD AX,BX
         MOV CX,AX
CONT:    MOV AH,0         ；通过循环做累加
         MOV AL,RESULT
         ADD AL,DA1
         AAA
         MOV RESULT,AL
         MOV AL,RESULT+1
         ADD AL,AH
         MOV AH,0
         AAA
         ADD AL,DA1+1
         AAA
         MOV RESULT+1,AL
         MOV AL,RESULT+2
         ADD AL,AH
         MOV AH,0
         AAA
         MOV RESULT+2,AL
         MOV AL,RESULT+3
         ADD AL,AH
         MOV AH,0
         AAA
         MOV RESULT+3,AL
         LOOP CONT
```

```
        MOV AH,4CH        ；返回 DOS
        INT 21H
CODE ENDS
  END START
```

18. 编写完整的程序求 N!，求 N 大于 6 时的运算结果，并在屏幕上显示结果。

答：

```
DATA SEGMENT
    N DB 8              ;求 8 的阶乘
    DATA ENDS
    CODE SEGMENT
    ASSUME CS:CODE,DS:DATA
    START: MOV AX,DATA
           MOV DS,AX
           MOV AL,N
           MOV AH,0
           CALL FACT        ；调用过程求 N!，AX=N，BX=N!
           CALL B2TODEC   ；调用过程将结果转换为十进制，然后屏幕显示
           MOV AH,4CH
           INT 21H
    FACT PROC
           CMP AL,0
           JNZ CHN
           MOV BX,1
           RET
    CHN:   PUSH AX
           DEC AL
           CALL FACT        ；递归调用 N!=N*（N-1)!
           POP AX
           MUL BX
           MOV BX,AX
           RET
    FACT ENDP
    B2TODEC PROC
           MOV CX,10000
           CALL BIN
           MOV CX,1000
           CALL BIN
           MOV CX,100
           CALL BIN
           MOV CX,10
           CALL BIN
           MOV CX,1
           CALL BIN
           RET
    B2TODEC ENDP
```

```
BIN PROC
        MOV AX,BX
        MOV DX,0
        DIV CX
        MOV BX,DX
        MOV DL,AL
        ADD DL,30H
        MOV AH,2
        INT 21H
        RET
BIN ENDP
CODE ENDS
  END START
```

19. 在附加段有一个数组，首地址为 BUFF，数组中第一个字节存放了数组的长度。编一个程序在数组中查找 0，找到后把它从数组中删去，后续项向前压缩，其余部分补 0。

答：

```
DATA   SEGMENT
BUFF   DB     10,   1,0,2,3,4,0,5,6,7,0     ;10 个数
M      EQU    0
DATA   ENDS
CODE   SEGMENT
       ASSUME CS:CODE, ES:DATA, DS:DATA
START:    MOV    AX, DATA   ; 初始化 ES，DS
          MOV    ES, AX
          MOV    DS, AX
          MOV    AL, M          ; 关键字 M 存入 AL
          MOV    DI, OFFSET    BUFF
          MOV    CL, [DI]       ; 数组长度存入 CX
          MOV    CH, 0
          INC     DI              ;指向数组起始地址
          CLD                     ;清方向标志
  L1:     REPNE   SCASB           ;重复搜索关键字
          JNZ    STOP             ; 未找到，转 STOP 结束
          JCXZ    STOP            ; 最后一个数是 M，转 STOP
          PUSH    DI              ;关键字下一单元地址和循
          PUSH    CX              ;环次数入栈保护
  DEL:    MOV    BL, [DI]         ; 前移，末尾补 0
          MOV    [DI-1], BL
          INC    DI
          LOOP    DEL
          MOV    BYTE PTR[DI-1], 0
          POP    CX               ; 恢复 CX，DI
          POP    DI
          DEC    DI               ; 由于 REPNE SCASB 已自动加 1
          JMP    L1
```

```
    STOP:     MOV   AH, 4CH
              INT   21H
    CODE      ENDS
        END   START
```

20. 编程完成将第二个字符串插入到第一个字符串的指定位置上。

答：

```
DATA SEGMENT
STRING1   DB 'THIS IS THE FIRST STRING!','$'
          DB 100 DUP(0)        ;缓冲区
NUM1      DW ?                 ；字符串 1 的长度
STRING2   DB 'the second string!','$'
NUM2      DW  ?                ；字符串 2 的长度
POS       DW 8                 ；插入位置
DATA ENDS
CODE SEGMENT
 ASSUME CS:CODE,DS:DATA,ES:DATA
START:  MOV AX,DATA
        MOV DS,AX
        MOV ES,AX
        LEA DI,STRING1       ；求字符串 1 的长度存入 NUM1
        MOV AL,'$'
        MOV NUM1,0
        CLD
  D1:   SCASB
        JZ   D2
        INC NUM1
        JMP D1
  D2:   LEA DI,STRING2       ；求字符串 2 的长度存入 NUM2
        MOV AL,'$'
        MOV NUM2,0
        CLD
  D3:   SCASB
        JZ D4
        INC NUM2
        JMP D3
  D4:   LEA SI,STRING1       ；将字符串 1 自插入位置开始的字符向后移动，
        ADD SI,NUM1          ；空出位置以便插入字符串 2
        MOV DI,SI
        ADD DI,NUM2
        STD
        MOV CX,NUM1
        SUB CX,POS
        INC CX
        REP MOVSB
        LEA SI,STRING2     ；将字符串 2 插入到字符串 1 指定位置
```

```
        LEA DI,STRING1
        ADD DI,POS
        CLD
        MOV CX,NUM2
        REP MOVSB
        LEA DX,STRING1
        MOV AH,9
        INT 21H
        MOV AH,4CH
        INT 21H
CODE ENDS
  END START
```

21. 将学生的班级、姓名、学号、课程名、成绩定义为一个结构，用结构预置语句，产生 5 个学生的成绩登记表，编程序将成绩小于 60 分的学生姓名、成绩显示出来。

答：

```
STUDENT STRUC
        CLASS   DB ?                    ；班级
        NAM     DB 'ABCDE$'             ；姓名
        NUM     DB ?                    ；学号
        COURSE DB 'ABCD'                ；课程
        SCORE   DB ?                    ；成绩
STUDENT ENDS
DATA SEGMENT
STUDENT1 STUDENT <1,'XIAOA$',001,'MATH',70>
STUDENT2 STUDENT <1,'XIAOB$',002,'MATH',80>
STUDENT3 STUDENT <1,'XIAOC$',003,'MATH',50>
STUDENT4 STUDENT <1,'XIAOD$',004,'MATH',90>
STUDENT5 STUDENT <1,'XIAOE$',005,'MATH',55>
DATA ENDS
CODE SEGMENT
   ASSUME CS:CODE,ds:DATA
START:   MOV AX,DATA
        MOV DS,AX
        MOV CX,5                        ；通过循环结构进行筛选
        MOV SI,0
  CONT: MOV AL,STUDENT1.SCORE[SI]       ；取成绩
        CMP AL,60
        JAE D1
        LEA DX,STUDENT1.NAM[SI]         ；小于 60 则显示姓名和成绩
        MOV AH,9
        INT 21H
        MOV DL,' '
        MOV AH,2
        INT 21H
        MOV AL,STUDENT1.SCORE[SI]       ；成绩转换为 ASCII 码显示
```

```
        MOV AH,0
        MOV BL,10
        DIV BL
        PUSH AX
        MOV DL,AL
        ADD DL,30H
        MOV AH,2
        INT 21H
        POP AX
        MOV DL,AH
        ADD DL,30H
        MOV AH,2
        INT 21H
        MOV DL,0DH
        MOV AH,2
        INT 21H
        MOV DL,0AH
        INT 21H
  D1:   ADD SI,13
        LOOP CONT
        MOV AH,4CH
        INT 21H
CODE ENDS
  END START
```

22. 编程序统计学生的数学成绩，分别归类 90 分～99 分，80 分～89 分，70 分～79 分，60 分～69 分及 60 分以下，并将各段的人数送入内存单元中。

答：设学生人数为字节，成绩为压缩 BCD 码，且都是合法的。

```
    DATA   SEGMENT
    BUFF    DB    XXH,…….
    ANUM   EQU   $-BUFF
    SNUM   DB   5   DUP(0)          ；存放各类成绩统计结果
    BUFF1   DB   ANUM   DUP(0)   ；存放 60 分以下成绩
    BUFF2   DB   ANUM   DUP(0)   ；存放 60 分～69 分成绩
    BUFF3   DB   ANUM   DUP(0)   ；存放 70 分～79 分成绩
    BUFF4   DB   ANUM   DUP(0)   ；存放 80 分～89 分成绩
    BUFF5   DB   ANUM   DUP(0)   ；存放 90 分以上成绩
    DATA   ENDS
    CODE   SEGMENT
       ASSUME   CS:CODE,DS: DATA,ES: DATA
    START:   MOV   AX, DATA
             MOV   DS,AX
             MOV   ES,AX
             MOV   CL,ANUM        ；取学生人数
             MOV   CH,0
```

```
        MOV    BH,0
        MOV    SI,OFFSET BUFF
        MOV    DI,OFFSET SNUM
D1:     MOV    AL,[SI]
        CMP    AL,60H
        JAE    NEXT1
        MOV    BL,[DI]
        MOV    BUFF1[BX],AL
        INC    BYTE PTR [DI]
        JMP    NEXT5
NEXT1:  CMP    AL,69H
        JA     NEXT2
        MOV    BL,[DI+1]
        MOV    BUFF2[BX],AL
        INC    BYTE PTR [DI+1]
        JMP    NEXT5
NEXT2:  CMP    AL,79H
        JA     NEXT3
        MOV    BL,[DI+2]
        MOV    BUFF2[BX],AL
        INC    BYTE PTR [DI+2]
        JMP    NEXT5
NEXT3:  CMP    AL,89H
        JA     NEXT4
        MOV    BL,[DI+3]
        MOV    BUFF3[BX],AL
        INC    BYTE PTR [DI+3]
        JMP    NEXT5
NEXT4:  MOV    BL,[DI+4]
        MOV    BUFF4[BX],AL
        INC    BYTE PTR [DI+4]
NEXT5:  INC    SI
        LOOP   D1
        MOV    AH,4CH
        INT    21H
CODE    ENDS
        END    START
```

23. 编制宏定义，将存储器区中一个用'$'结尾的字符串传送到另一个存储器区中，要求源地址、目的地址、串结尾符号可变。

答：

```
SEND MACRO SCHARS, DCHARS , FLAG
    LOCAL NEXT, EXIT        ；LOCAL 用于解决宏定义内的标号问题
          PUSH AX           ；SCHARS 源串地址，DCHARS 目的串地址
          PUSH SI           ；FLAG 串结尾符号
          MOV SI, 0
```

```
NEXT:   MOV AL, SCHARS[SI]
        MOV DCHARS[SI], AL
        CMP AL, FLAG
        JZ EXIT
        INC SI
        JMP NEXT
EXIT:   POP SI
        POP AX
    ENDM
```

24. 定义宏指令名 FINSUM：它完成比较两个数 X 和 Y，若 X>Y，则执行 X+2*Y 结果送到 SUM，若 X≤Y，则执行 2*X+Y 结果送到 SUM。

答：

```
FINSUM  MACRO  X, Y, SUM
        IF  X  GT  Y
          MOV  AX, Y
          SHL  AX, 1
          ADD  AX, X
          MOV  SUM, AX
        ELSE
          MOV  AX, X
          SHL  AX, 1
          ADD  AX, Y
          MOV  SUM, AX
        ENDIF
    ENDM
```

25. DOS 功能调用需要在 AH 寄存器中存放不同的功能码，试将这些功能调用定义成宏指令 DOS，再定义宏指令 DISP，完成显示字符的功能，并展开宏调用 DISP ‘*’。

答：

```
DOS  MACRO  NUM
     MOV  AH, NUM
     INT  21H
   ENDM
   DISP  MACRO  ZIFU
     MOV  DL, ZIFU
     DOS  02H
   ENDM
```

宏调用：DISP ‘*’

宏展开：

```
MOV  DL, '*'
MOV  AH, 02H
INT  21H
```

26. 编一段程序产生乐曲。

答：演奏儿歌《一闪一闪亮晶晶》

1 1|5 5|6 6|5－|4 4|3 3|2 2|1－|

5 5|4 4|3 3|2－|5 5|4 4|3 3|2－|

1 1|5 5|6 6|5－|4 4|3 3|2 2|1－||

```
DATA SEGMENT
FREQUECY    DW 65535,262,294,330,349,392,440,494    ；各音阶频率，65535 对应 0，
            DW 523,578,659,698,784,880,988          ；表示不发声
            DW 1046,1175,1318,1397,1568,1760,1976
TABLE     DW    0,8,8,12,12,13,13,12,0,   11,11,10,10,9,9,8,0    ；乐谱各音符频率在
          DW    12,12,11,11,10,10,9,0,   12,12,11,11,10,10,9,0  ；FREQUECY 中的相对
          DW    8,8,12,12,13,13,12,0,   11,11,10,10,9,9,8,0     ；位置
DATA    ENDS
CODE SEGMENT
    ASSUME    CS:CODE,DS:DATA
START:    MOV AX,DATA
          MOV DS,AX
          MOV    BP,49       ;CI SHU
          LEA    SI,TABLE
CONT:     MOV    BX,[SI]
          INC    SI
          INC    SI
          SHL    BX,1
          MOV    DI,[FREQUECY+BX]
          CALL   PLAY
          DEC    BP
          JNZ    CONT
EXIT:     MOV    AH,4CH
          INT 21H
PLAY      PROC
          MOV AL,10110110B
          OUT 43H,AL
          MOV DX,12H
          MOV AX,34DEH
          DIV    DI
          OUT    42H,AL
          MOV    AL,AH
          OUT    42H,AL
          IN     AL,61H
          MOV    AH,AL
          OR     AL,03H
          OUT    61H,AL
          MOV CX,0FFFFH
DELAY:    MOV    DX,1700H
GOON:     DEC    DX
          JNZ    GOON
          LOOP   DELAY
          IN    AL,61H
          MOV AH,AL
```

```
        AND AL,0FCH
        OUT    61H,AL
        MOV    CX,0FFFFH
DELAY1: MOV    DX,100H
GOON1:    DEC    DX
        JNZ    GOON1
        LOOP DELAY1
        RET
PLAY    ENDP
CODE     ENDS
    END    START
```

第五章

1. 静态 RAM 与动态 RAM 有何区别？

答：（1）静态 RAM 内存储的信息只要电源存在就能一直保持，而动态 RAM 的信息需要定时刷新才能保持

（2）静态 RAM 的集成度比较低，运行速度快，而动态 RAM 的集成度高，运行相对较慢

（3）静态 RAM 造价成本高，动态 RAM 价格便宜

2. ROM、PROM、EPROM、EEPROM 在功能上各有何特点？

答：ROM 是只读存储器，根据写入方式的不同可以分为四类：掩膜型 ROM、PROM、EPROM 和 EEPROM。

掩膜型 ROM 中信息是厂家根据用户给定的程序或数据，对芯片图形掩膜进行两次光刻而写入的，用户对这类芯片无法进行任何修改。PROM 出厂时，里面没有信息，用户采用一些设备可以将内容写入 PROM，一旦写入，就不能再改变了，即只允许编程一次。

EPROM 可编程固化程序，且在程序固化后可通过紫外光照擦除，以便重新固化新数据。

EEPROM 可编程固化程序，并可利用电压来擦除芯片内容，以重新编程固化新数据。

3. DRAM 的 $\overline{\text{CAS}}$ 和 $\overline{\text{RAS}}$ 输入的用途是什么？

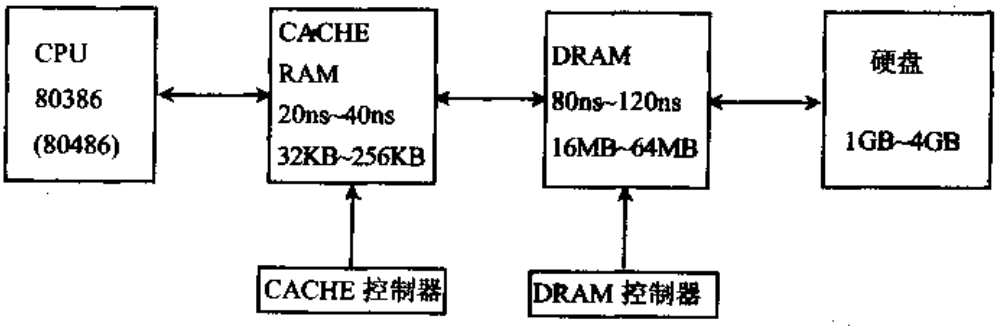
答：$\overline{\text{CAS}}$ 为列地址选通信号，用于指示地址总线上的有效数据为列地址；$\overline{\text{RAS}}$ 为行地址选通信号，用于指示地址总线上的有效数据为列地址。

4. 什么是 Cache？作用是什么？它处在微处理机中的什么位置？

答：Cache 也称为高速缓存，是介于主存和 CPU 之间的高速小容量存储器。

为了减少 CPU 与内存之间的速度差异，提高系统性能，在慢速的 DRAM 和快速 CPU 之间插入一速度较快、容量较小的 SRAM，起到缓冲作用，使 CPU 既可以以较快速度存取 SRAM 中的数据，又不使系统成本上升过高，这就是 Cache 的作用。

Cache 在微处理机中的位置如下图：



5. 直接映像 Cache 和成组相联 Cache 的组成结构有什么不同？

答：直接映象 Cache 是将主存储器中每一页大小分成和 Cache 存储器大小一致，Cache 中每一块分配一个索引字段以确定字段，这样可以通过一次地址比较即可确定是否命中，但如果频繁访问不同页号主存储器时需要做频繁的转换，降低系统性能；

成组相联 Cache 内部有多组直接映象的 Cache，组间采用全关联结构，并行地起着高速缓存的作用。访问时需要进行两次比较才能确定是否命中。

6. 为什么要保持 Cache 内容与主存储器内容的一致性？为了保持 Cache 与主存储器内容的一致性应采取什么方法？

答：由于 Cache 的内容只是主存部分内容的拷贝，故应当与主存内容保持一致。数据不一致问题通常是由于更新了 Cache 的数据而没有更新与其关联的存储器的数据，或更新了存储器数据却没有更新 Cache 的内容所引起的。

为了保持 Cache 与主存储器内容的一致性，有两种写入策略：

（1）通写法

在此方法中，当 CPU 写入数据到 Cache 中后，Cache 就立即将其写入主存中，使主存始终保持 Cache 中的最新内容。此方法简单，更新内容不会丢失，但每次对 Cache 的修改同时要写入主存储器，总线操作频繁，影响系统性能。

（2）回写法

此方法中，Cache 的作用好像缓冲区一样，当 CPU 写入数据到 Cache 中后，Cache 并不立即将其回写到主存中，而是等到系统总线空闲时，才将 Cache 中的内容回写到主存中，此方法使得 CPU 可以持续运行而不必等待主存的更新，性能比通写法要提高很多，但其 Cache 控制器复杂，价格高。

7. 用 1024×1 位的 RAM 芯片组成 16K×8 位的存储器，需要多少芯片？在地址线中有多少位参与片内寻址？多少位组合成片选择信号？（设地址总线为 16 位）

答：由于所用的芯片为 1024×1 位，构成 1024×8 位（即 1K×8 位）的存储器需要 8 片，因此组成 16K×8 位的存储器需要 16×8＝128 片。

片内有 1024 个单元，需要 10 根地址线。

16 组（每组 8 片）存储器需要 16 根片选信号，至少需要 4 根地址线经译码器输出。

8. 现有一存储体芯片容量为 512×4 位，若要用它组成 4KB 的存储器，需要多少这样的芯片？每块芯片需要多少寻址线？整个存储系统最少需要多少寻址线？

答：4K×8 / 512×4=16 片

每块芯片内有 512 个单元，故需要 9 根地址线

整个存储系统最少需要 12 根地址线

9. 利用 1024×8 位的 RAM 芯片组成 4K×8 位的存储器系统，试用 A_{15}～A_{12} 地址线用线性选择法产生片选信号，存储器的地址分配有什么问题，并指明各芯片的地址分配。

答：组成 4K×8 的存储器，那么需要 4 片这样的芯片：将 A_{15} 取反后分配芯片 1 的 $\overline{CS}$；将 A_{14} 取反后分配给芯片 2 的 $\overline{CS}$；将 A_{13} 取反后分配芯片 3 的 $\overline{CS}$；将 A_{12} 取反后分配给芯片 4 的 $\overline{CS}$。

芯片 1 的地址范围 8000H~83FFH、8400H~87FFH、8800H~8BFFH、8C00H~8FFFH

芯片 2 的地址范围 4000H~43FFH、4400H~47FFH、4800H~4BFFH、4C00H~4FFFH

芯片 3 的地址范围 2000H~23FFH、2400H~27FFH、2800H~2BFFH、2C00H~2FFFH

芯片 4 的地址范围 1000H~13FFH、1400H~17FFH、1800H~1BFFH、1C00H~1FFFH

这样会造成地址的重叠。

10. 当从存储器偶地址单元读一个字节数据时，写出存储器的控制信号和它们的有效逻辑电平信号。（8086 工作在最小模式）

答：8086 发出 20 位地址信息和 $\overline{BHE}$ =1，通过地址锁存信号锁存至 8282，然后发出 $M/\overline{IO}$ =1 和 $\overline{RD}$ =0 等控制信号，20 位地址信号和 $\overline{BHE}$ =1 送给存储器，经过译码，选中偶地址单元一字节，将其数据读出，送至数据总线，经过由 $\overline{DEN}$ =0 和 $DT/\overline{R}$ =0 控制的数据收发器 8286 传送至 CPU。

11. 当要将一个字写入到存储器奇地址开始的单元中去，列出存储器的控制信号和它们的有效逻辑电平信号。（8086 工作在最小模式）

答：此时要启动 2 个写总线周期，第一个写周期将字的低 8 位写入存储器奇地址单元，第二个写周期将字的高 8 位写入存储器奇地址单元下一个单元。

第一个写周期中，$\overline{BHE}$=0，第二个写周期中，$\overline{BHE}$=1。

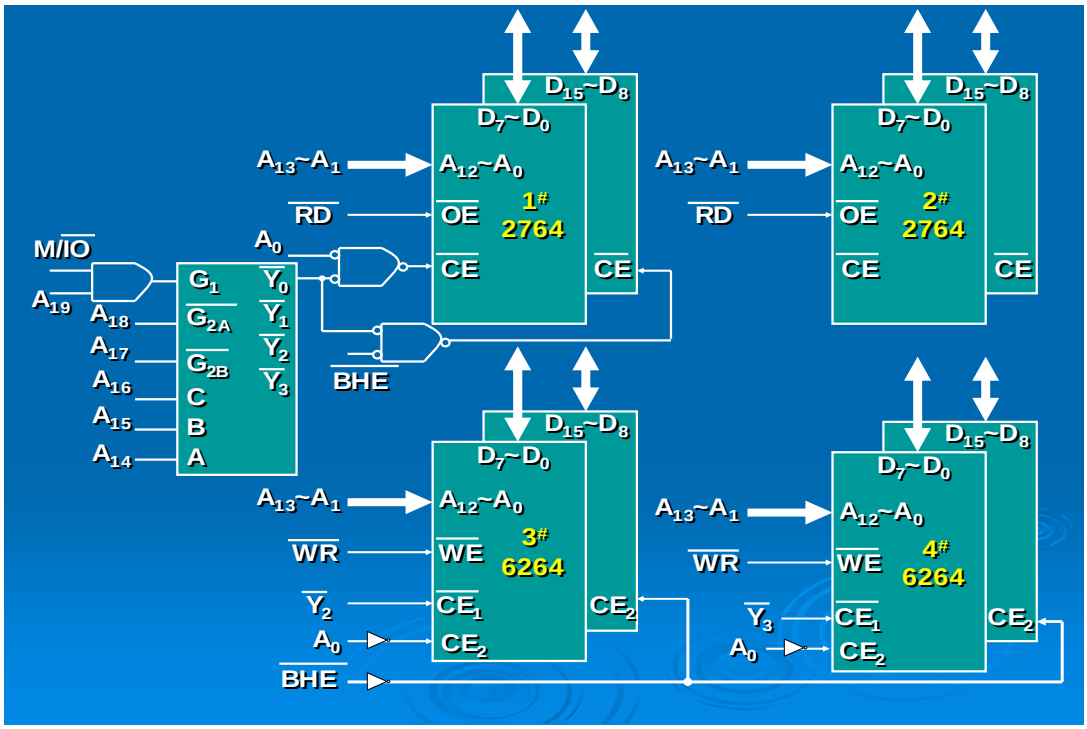
其余信号 M/$\overline{IO}$=1，$\overline{WR}$=0，$\overline{DEN}$=0，DT/$\overline{R}$=1

12. 设计一个 64K×8 存储器系统，采用 74LS138 和 EPROM2764 器件，使其寻址存储器的地址范围为 40000H～4FFFFH。

答：因为 EPROM2764 是 8K×8 的 ROM，所以要构成 64K×8 的存储器系统，需要 8 片 EPROM2764。其中 CPU 的 A_{12}~A_0 直接与 8 片 EPROM 的 A_{12}~A_0 相连（没有考虑驱动能力问题），A_{15}、A_{14} 和 A_{13} 与 138 的 A、B、C 三个端口相连，其他地址线（A_{19}～A_{16}）和 M/$\overline{IO}$ 组合连到 G1、G_{2A} 和 G_{2B} 上，确保 A_{19}=0、A_{18}=1、A_{17}=0 和 A_{16}=0 即可。

13. 用 8K×8 位的 EPROM2764、8K×8 位的 RAM6264 和译码器 74LS138 构成一个 16K 字 ROM、16K 字 RAM 的存储器子系统。8086 工作在最小模式，系统带有地址锁存器 8282，数据收发器 8286。画出存储器系统与 CPU 的连接图，写出各块芯片的地址分配。

答：



			C	B	A		
A19	A18	A17	A16	A15	A14	A13~A0	
1	0	0	0	0	0		Y0 有效 80000~83FFFH
1	0	0	0	0	1		Y1 有效 84000~87FFFH
1	0	0	0	1	0		Y2 有效 88000~8BFFFH
1	0	0	0	1	1		Y3 有效 8C000~8FFFFH

14. 上题中若从 74LS138 的 Y2 开始选择 ROM 和 RAM 芯片，写出各块芯片的地址分配。

答：

			C	B	A		
A19	A18	A17	A16	A15	A14	A13~A0	
1	0	0	0	1	0		Y2 有效 88000~8BFFFH
1	0	0	0	1	1		Y3 有效 8C000~8FFFFH
1	0	0	1	0	0		Y4 有效 90000~93FFFH
1	0	0	1	0	1		Y5 有效 94000~97FFFH

第六章

1. CPU 与外设交换数据时，为什么要通过 I/O 接口进行？I/O 接口电路有哪些主要功能？

答：CPU 和外设之间的信息交换存在以下一些问题：速度不匹配；信号电平不匹配；信号格式不匹配；时序不匹配。

I/O 接口电路是专门为解决 CPU 与外设之间的不匹配、不能协调工作而设置的，处于总线和外设之间，一般应具有以下基本功能：

⑴设置数据缓冲以解决两者速度差异所带来的不协调问题；

⑵设置信号电平转换电路，来实现电平转换。

⑶设置信息转换逻辑，如模拟量必须经 A/D 变换成数字量后，才能送到计算机去处理，而计算机送出的数字信号也必须经 D/A 变成模拟信号后，才能驱动某些外设工作。

⑷设置时序控制电路；

⑸提供地址译码电路。

2. 在微机系统中，缓冲器和锁存器各起什么作用？

答：缓冲器多用在总线上，可提高总线驱动能力、隔离前后级起到缓冲作用，缓冲器多半有三态输出功能。

锁存器具有暂存数据的能力，能在数据传输过程中将数据锁住，然后在此后的任何时刻，在输出控制信号的作用下将数据传送出去。

3. 什么叫 I/O 端口？一般的接口电路中可以设置哪些端口？计算机对 I/O 端口编址时采用哪两种方法？在 8086/8088CPU 中一般采用哪些编址方法？

答：在 CPU 与外设通信时，传送的信息主要包括数据信息、状态信息和控制信息。在接口电路中，这些信息分别进入不同的寄存器，通常将这些寄存器和它们的控制逻辑统称为 I/O 端口。

一般的接口电路中可以设置数据端口、状态端口和命令端口。

计算机对 I/O 端口编址时采用两种方法：存储器映像寻址方式、I/O 单独编址方式。

在 8086/8088CPU 中一般采用 I/O 单独编址方式。

4. CPU 与外设间传送数据主要有哪几种方式？

答：CPU 与外设间的数据传送方式主要有：程序控制方式、中断方式、DMA 方式。

程序控制传送方式：CPU 与外设之间的数据传送是在程序控制下完成的。⑴无条件传送方式：也称为同步传送方式，主要用于对简单外设进行操作，或者外设的定时是固定的或已知的场合。 ⑵条件传送：也称为查询式传送方式，在开始传送前，必须先查询外设已处于准备传送数据的状态，才能进行传送。

采用中断方式：CPU 平时可以执行主程序，只有当输入设备将数据准备好了，或者输出端口的数据缓冲器已空时，才向 CPU 发中断请求。CPU 响应中断后，暂停执行当前的程序，转去执行管理外设的中断服务程序。在中断服务程序中，用输入或输出指令在 CPU 和外设之间进行一次数据交换。等输入或输出操作完成之后，CPU 又回去执行原来的程序。

DMA 方式：也要利用系统的数据总线、地址总线和控制总线来传送数据。原先，这些总线是由 CPU 管理的，但当外设需要利用 DMA 方式进行数据传送时，接口电路可以向 CPU 提出请求，要求 CPU 让出对总线的控制权，用 DMA 控制器来取代 CPU，临时接管总线，控制外设和存储器之间直接进行高速的数据传送。这种控制器能给出访问内存所需要的地址信息，并能自动修改地址指针，也能设定和修改传送的字节数，还能向存储器和外设发出相应的读/写控制信号。在 DMA 传送结束后，它能释放总线，把对总线的控制权又交还给 CPU。

5. 说明查询式输入和输出接口电路的工作原理。

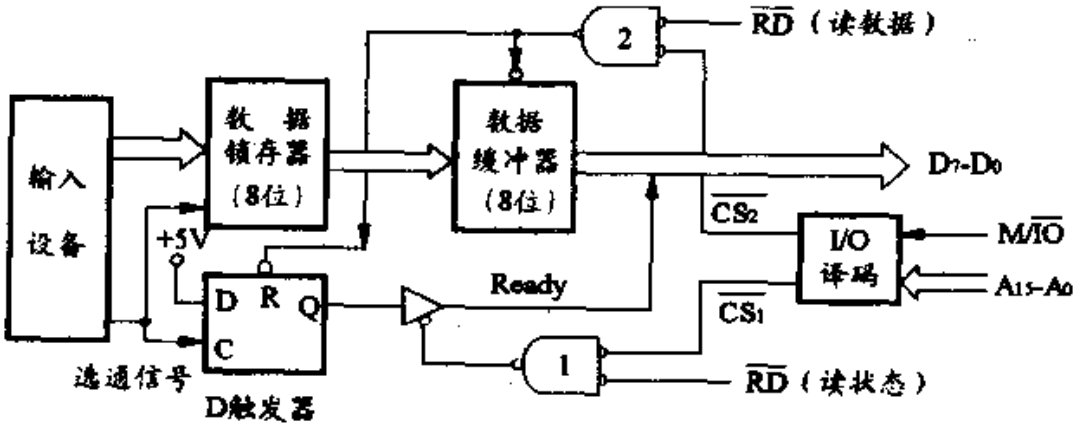
答：查询式传送方式也称为条件传送方式。一般情况下，当 CPU 用输入或输出指令与外设

交换数据时，很难保证输入设备总是准备好了数据，或者输出设备已经处在可以接收数据的状态。为此，在开始传送前，必须先确认外设已处于准备传送数据的状态，才能进行传送，于是就提出了查询式传送方式。

查询式传送方式的工作过程：

在传送数据前，CPU 要先执行一条输入指令，从外设的状态口读取它的当前状态。如果外设未准备好数据或处于忙碌状态，则程序要转回去反复执行读状态指令，不断检测外设状态；如果该外设的输入数据已经准备好，CPU 便可执行输入指令，从外设读入数据。

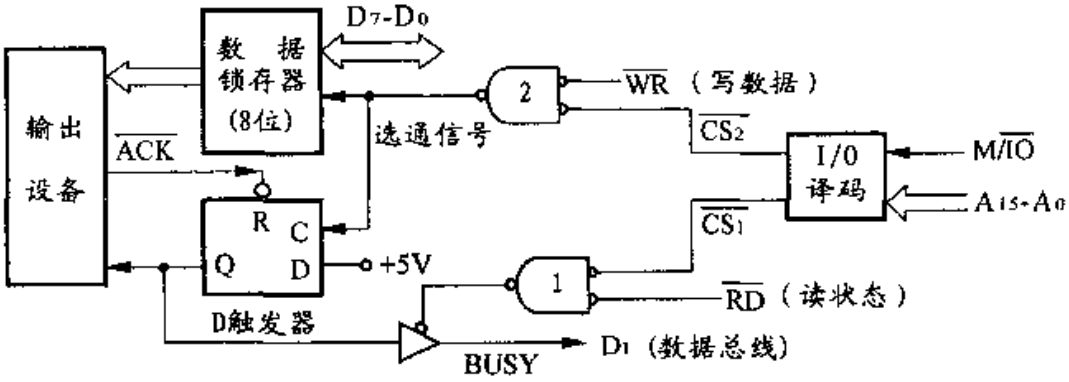
查询式输入方式的接口电路如下：



查询式输入方式的工作过程：

当输入设备准备好数据后，就向 I/O 接口电路发一个选通信号。此信号有两个作用：一方面将外设的数据打入接口的数据锁存器中，另一方面使接口中的 D 触发器的 Q 端置 1。CPU 首先执行 IN 指令读取状态口的信息，这时 M/$\overline{IO}$ 和 $\overline{RD}$ 信号均变低，M/$\overline{IO}$ 为低，使 I/O 译码器输出低电平的状态口片选信号 $\overline{CS1}$。$\overline{CS1}$ 和 $\overline{RD}$ 经门 1 相与后的低电乎输出，使三态缓冲器开启，于是 Q 端的高电平经缓冲器(1 位)传送到数据线上的 READY(D_0)位，并被读入累加器。程序检测到 READY 位为 1 后，便执行 IN 指令读数据口。这时 M/$\overline{IO}$ 和 $\overline{RD}$ 信号再次有效，先形成数据口片选信号 $\overline{CS2}$，$\overline{CS2}$ 和 $\overline{RD}$ 经门 2 输出低电平。它一方面开启数据缓冲器，将外设送到锁存器中的数据经 8 位数据缓冲器送到数据总线上后进入累加器，另一方面将 D 触发器清 0，一次数据传送完毕。接着就可以开始下一个数据的传送。当规定数目的数据传送完毕后，传送程序结束，程序将开始处理数据或进行别的操作。

查询式输出方式的接口电路如下：



查询式输出方式的工作过程：

当 CPU 准备向外设输出数据时，它先执行 IN 指令读取状态口的信息。这时，低电平的 $M/\overline{IO}$ 和有效的端口地址信号使 I/O 译码器的状态口片选信号 $\overline{CS1}$ 变低，$\overline{CS1}$ 再和有效的 $\overline{RD}$ 信号经门 1 相与后输出低电平，它使状态口的三态门开启，从数据总线的 D1 位读入 BUSY 状态。若 BUSY＝l，表示外设处在接收上一个数据的忙碌状态。只有在 BUSY＝0 时，CPU 才能向外设输出新的数据。当 CPU 检查到 BUSY＝0 时，便执行 OUT 指令将数据送向数据输出口。这时低电平的 $M/\overline{IO}$ 使 I / O 译码器的状态口片选信号 $\overline{CS2}$ 变低，$\overline{CS2}$ 再和 $\overline{WR}$ 信号经门 2 相与后输出低电平的选通信号，它用来选通数据锁存器，将数据送向外设。同时，选通信号的后沿还使 D 触发器翻转，置 Q 为高电平，即把状态口的 BUSY 位置成 l，表示忙碌。当输出设备从接口中取走数据后，就送回一个应答信号 $\overline{ACK}$，它将 D 触发器清 0，即置 BUSY＝0，允许 CPU 送出下一个数据。

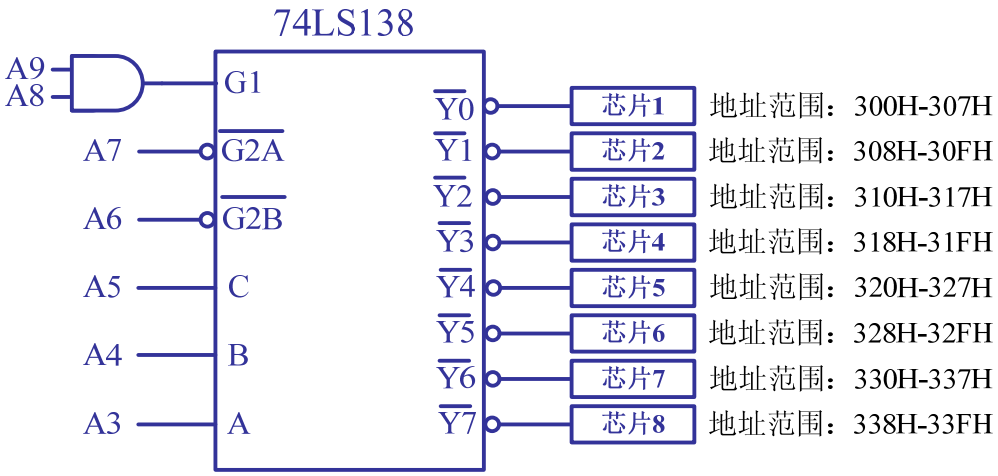
6. 简述在微机系统中，DMA 控制器从外设提出请求到外设直接将数据传送到存储器的工作过程。

答：DMA 方式，外设向内存传输数据的过程：

当一个接口中有数据要输入时，就向 DMA 控制器发送 DMA 请求； DMA 控制器接收到请求后，便往控制总线上发一个总线请求；如果 CPU 允许让出总线，则发一个总线允许信号；DMA 控制器接到此信号后，就将地址寄存器的内容送到地址总线上，同时往接口发一个 DMA 回答信号，并发一个 1/O 读信号和一个内存写信号；接口接到 DMA 回答信号以后，将数据送到数据总线上，并撤除 DMA 请求信号；内存在接收到数据以后，一般往 DMA 控制器回送一个准备好信号，DMA 控制器的地址寄存器内容加 1 或减 1，计数器的值减 1，而且撤除总线请求信号，这样，就完成了对一个数据的 DMA 输入传输。DMA 传输结束时，往接口发一个结束信号，向 CPU 交回总线控制权。其状态寄存器的传输结束标志置“1”。查询时，CPU 在主程序中通过查询状态寄存器的传输结束标志，决定是否进行后续处理。

7. 某一个微机系统中，有 8 块 I/O 接口芯片，每个芯片占有 8 个端口地址，若起始地址为 300H，8 块芯片的地址连续分布，用 74LS138 做译码器，试画出端口译码电路，并说明每块芯片的端口地址范围。

答：



8. 什么叫总线？总线分哪几类？在微型计算机中采用总线结构有什么好处？

答：在微型计算机系统中，将用于各部件之间传送信息的公共通路称为总线（BUS）。

总线分三类：片级总线、系统总线、外部总线。

在微型计算机中采用总线的好处有：

（1）总线具有通用性，只要按统一的总线标准进行设计或连接，不同厂家生产的插件板可以互换，不同系统之间可以互连和通信，很适合于大批量生产、组装和调试，也便于更新和扩充系统。

（2）对于用户来说，可以根据自身需要，灵活地选购接口板和存储器插件，还可以根据总线标准的要求，自行设计接口电路板，来组装成适合自己应用需要的系统或更新原有系统。

9. PC 总线和 ISA 总线各用于何种类型的微型计算机中？它们的数据总线各有多少根？

答：IBM PC/XT 使用的总线称为 PC 总线，有 8 条数据线。

ISA 总线又称 AT 总线，是以 80286 为 CPU 的 PC/AT 机及其兼容机所用的总线，具有 16 条数据线。

10. PCI 总线有哪些主要特点？根据 PCI 总线引脚图和典型时序图，说明如何完成 PCI 突发读交易。

答：PCI 总线的特点：

（1）PCI 总线传输数据的位数为 32 位，也可扩展到 64 位。32 位 PCI 总线在读写传送中，以 33MHz 的频率进行，传输速率为 132MB/s，当数据宽度为 64 位时，以 66MHz 的频率运行，传输速率高达 528MB/s。

（2）PCI 总线支持突发传送方式（Burst Transfer）。

（3）PCI 总线支持即插即用（Plug and Play，PnP）功能。

（4）PCI 总线与微处理器之间不直接相连，而是通过与 CPU 结构无关的中间部件桥接器相连。

PCI 总线突发读交易的过程：

PCI 突发读交易以周期帧信号 FRAME#有效后开始进行 PCI 交易，交易的第一个时钟周期为地址时段，此时主设备通过驱动地址总线寻址目标，驱动 PCI 命令确定交易类型。因 PCI 的地址线/数据线（AD）和命令信号（C/BE#）都是分时复用的，所以每个 PCI 目标设备在下一个时钟周期的上升沿将地址信号和命令信号锁存，然后经译码，确定自己是否是本次寻找的目标和将要交易的类型是什么。PCI 目标确定自己是交易的目标后，则将设备选择信号 DEVSEL#置为有效状态，向交易的启动方声明本次交易有效。在主设备的 IRDY#和目标设备的 TRDY#都有效的情况下可以连续进行交易，传送多个数据，直至交易结束为止。

第七章

1. 什么叫中断？什么叫可屏蔽中断和不可屏蔽中断？

答：当 CPU 正常运行程序时，由于微处理器内部事件或外设请求，引起 CPU 中断正在运行的程序，转去执行请求中断的外设(或内部事件)的中断服务子程序，中断服务程序执行完毕，再返回被中止的程序，这一过程称为中断。

可屏蔽中断由引脚 INTR 引入，采用电平触发，高电平有效，INTR 信号的高电平必须维持到 CPU 响应中断才结束。可以通过软件设置来屏蔽外部中断，即使外部设备有中断请求，CPU 可以不予响应。当外设有中断申请时，在当前指令执行完后，CPU 首先查询 IF 位，若 IF＝0，CPU 就禁止响应任何外设中断；若 IF＝1，CPU 就允许响应外设的中断请求。

不可屏蔽中断由引脚 NMI 引入，边沿触发，上升沿之后维持两个时钟周期高电平有效。不能用软件来屏蔽的，一旦有不可屏蔽中断请求，如电源掉电等紧急情况，CPU 必须予以响应。

2. 列出微处理器上的中断引脚和与中断有关的指令。

答：INTR：可屏蔽中断请求输入引脚。

NMI：不可屏蔽中断请求输入引脚

$\overline{\text{INTA}}$：可屏蔽中断响应引脚

INT n ：软件中断指令，其中 n 为中断类型号

INTO：溢出中断，运算后若产生溢出，可由此指令引起中断。

CLI：中断标志位 IF 清 0

STI：置位中断标志位为 1

3. 8086/8088 系统中可以引入哪些中断？

答：（1）外部中断

两种外部中断：不可屏蔽中断 NMI 和可屏蔽中断 INTR

（2）内部中断

内部中断又称软件中断，有三种情况引起：

①INT n ：中断指令引起的中断

②CPU 的某些运算错误引起的中断：包括除法错中断和溢出中断

③由调试程序 debug 设置的中断：单步中断和断点中断。

4. CPU 响应中断的条件是什么？简述中断处理过程。

答：CPU 响应中断要有三个条件：

外设提出中断申请；本中断位未被屏蔽；中断允许。

可屏蔽中断处理的过程一般分成如下几步：

中断请求；中断响应；保护现场；转入执行中断服务子程序；恢复现场和中断返回。

CPU 在响应外部中断，并转入相应中断服务子程序的过程中，要依次做以下工作：

⑴从数据总线上读取中断类型号，将其存入内部暂存器。

⑵将标志寄存器 PSW 的值入钱。

⑶将 PSW 中的中断允许标志 IF 和单步标志 TF 清 0，以屏蔽外部其它中断请求，避免 CPU 以单步方式执行中断处理子程字。

⑷保护断点，将当前指令下面一条指令的段地址 CS 和指令指针 IP 的值入栈，中断处理完毕后，能正确返回到主程序继续执行。

⑸根据中断类型号到中断向量表中找到中断向量，转入相应中断服务子程序。

⑹中断处理程序结束以后，从堆栈中依次弹出 IP、CS 和 PSW，然后返回主程序断点

处，继续执行原来的程序。

5. 中断服务子程序中中断指令 STI 放在不同位置会产生什么不同结果？中断嵌套时，STI 指令应如何设置？

答：由于响应中断时 CPU 自动关闭中断（IF＝0），故在中断服务子程序中 STI 指令后方可实现中断嵌套。一般在中断服务子程序中保护现场后即设置开中断指令 STI（IF＝1），以便实现中断嵌套。

6. 中断结束命令 EOI 放在程序不同位置处会产生什么不同结果？

答：中断结束命令 EOI 后，清除中断服务寄存器中的标志位，即允许响应同级或低级中断，为避免错误，一般将中断结束命令 EOI 置于中断服务子程序结束前。

7. 中断向量表的功能是什么？

答：中断向量表又称中断服务程序入口地址表。

将每个设备的中断服务程序入口地址(矢量地址)集中，依次放在中断向量表中。当 CPU 响应中断后，控制逻辑根据外设提供的中断类型号查找中断向量表，然后将中断服务程序的入口地址送到段寄存器和指令指针寄存器，CPU 转入中断服务子程序，这样可大大加快中断处理的速度。

8086/8088 系统允许处理 256 种类型的中断，对应类型号为 0~FFH。在存储器的 00000H~003FFH，占 1K 字节空间，用作存放中断向量。每个类型号占 4 个字节，高 2 个字节存放中断入口地址的段地址，低 2 个字节存放段内偏移地址。

8. 假定中断类型号 15 的中断处理程序的首地址为 ROUT15，编写主程序为其建立一个中断向量。

```
PUSH    ES
MOV     AX，0
MOV     ES，AX
MOV     DI，54H                ；15H*4
MOV     AX，OFFSET ROUT15      ；中断处理程序的偏移→AX
CLD
STOSW
MOV     AX，SEG ROUT15         ；中断处理程序的段地址→AX
STOSW
POP   ES
```

9. 8086/8088CPU 如何获得中断类型号？

答：对于内部中断

①INT　n ：n 即为中断类型号

②除法错中断自动获得类型号 0，INTO 溢出中断自动获得中断类型号 4

③单步中断自动获得类型号 1，断点中断自动获得类型号 3。

对于外部中断：

① 不可屏蔽中断 NMI：自动获得中断类型号 2

② 可屏蔽中断 INTR：

CPU 响应可屏蔽中断，对中断接口电路发出两个中断响应信号 $\overline{\text{INTA}}$，中断接口电路收到第二个 $\overline{\text{INTA}}$ 以后，通过数据线向 CPU 送中断类型号。

10. 给定 SP=0100H，SS=0500H，PSW=0240H，在存储单元中已有内容为(00024H)=0060H，(00026H)=1000H，在段地址为 0800H 及偏移地址为 00A0H 的单元中有一条中断指令 INT 9，

试问执行 INT 9 指令后，SP、SS、IP、PSW 的内容是什么？栈顶的三个字是什么？
答：执行 INT 9 指令，标志寄存器 PSW、下一条指令的段地址 CS 和指令指针 IP 的值分别入栈，PSW 中的中断允许标志 IF 和单步标志 TF 清 0，中断向量表的中断入口地址送 CS 和 IP，转入中断服务子程序。所以此时 SP＝00FAH，SS＝0500H，CS＝1000H，IP＝0060H，PSW＝0040H。栈顶的三个字是：(0500H:00FAH)＝00A2H、(0500H:00FCH)＝0800H、(0500H:00FEH)＝0240H

标志寄存器 PSW	X X X X	OF	DF	IF	TF	SF	ZF	X	AF	X	PF	X	CF
执行 INT 9 前	0 0 0 0	0	0	1	0	0	1	0	0	0	0	0	0
执行 INT 9 后	0 0 0 0	0	0	0	0	0	1	0	0	0	0	0	0

PSW＝0040H

11. 8259A 优先权管理方式有哪几种？中断结束方式又有哪几种？
答：8259A 优先权管理方式有如下 4 种：
（1）全嵌套方式
此方式下，中断优先级分配固定级别 0～7 级，IR0 具有最高优先级，IR7 优先级最低。
（2）特殊全嵌套工作方式
此种工作方式主要用于 8259A 级联情况。此方式与全嵌套工作方式基本相同，区别在于当处理某级中断时，有同级中断请求进入，8259A 也会响应，从而实现了对同级中断请求的特殊嵌套。
（3）优先级自动循环方式
优先级可以改变，初始优先级次序为 IR0～IR7，当任何一级中断被处理完后，它的优先级别变为最低，将最高优先级赋给原来比它低一级的中断请求。
（4）优先级特殊循环方式
特殊循环方式下，初始时优先级由程序指定，而不是固定的。
8259A 中断结束方式有如下 3 种：
（1）普通 EOI 结束方式
在全嵌套工作方式下，任何一级中断，处理结束返回上一级程序前，CPU 向 8259A 传送 EOI 结束命令字，8259A 收到 EOI 结束命令后，自动将 ISR 寄存器中级别最高的置 1 位清 0。
（2）特殊 EOI 结束方式
在非全嵌套工作方式下，中断服务寄存器是无法确定哪一级中断为最后响应和处理的，此时要采用特殊 SEOI 结束方式。CPU 向 8259A 发特殊 EOI 结束命令字，命令字中将当前要清除的中断级别也传给 8259A，此时 8259 将 ISR 寄存器中指定级别的对应位清 0。
（3）自动 EOI 结束方式
在自动 AEOI 方式中，任何一级中断被响应后，ISR 寄存器对应位置 1，但在 CPU 进入中断响应周期，发第二个 $\overline{\text{INTA}}$ 脉冲后，自动将 ISR 寄存器中对应位清 0。

12. 单片 8259A 在全嵌套中断工作方式下，要写哪些初始化命令字及操作命令字？
答：初始化命令字需要写：ICW1，ICW2，ICW4。
操作命令字需要写：OCW1 中断屏蔽操作命令字（根据需要，若不需要可不写）

OCW2 优先权循环方式和中断结束方式操作字

13. 系统中新增加一个中断源，在软件上应增加哪些内容，此中断系统才能正常工作？

答：需要确定中断源的中断类型，然后将中断服务程序的入口地址放入中断向量表中。编写相应的中断服务程序。

14. 系统中有 3 个中断源，从 8259A 的 IR0、IR2、IR4 端引入中断，以边沿触发，中断类型号分别为 50H、52H、54H，中断入口地址分别为 5020H、6100H、3250H，段地址为 1000H。使用完全嵌套方式，普通 EOI 结束，试编写初始化程序，使 CPU 能正确响应任何一级中断；并编写一段中断服务子程序，保证中断嵌套的实现和正确返回。

答：;初始化程序 设 8259A 的端口地址为 20H 和 21H

```
;设置中断向量表
MOV     AX, 1000H    ;段地址
MOV     DS, AX
MOV     DX, 5020H    ;偏移地址
MOV     AL, 50H      ;中断类型号
MOV     AH, 25H
INT     21H          ;写 50H 的中断入口地址
MOV     DX, 6100H    ;偏移地址
MOV     AL, 52H      ;中断类型号
INT     21H          ;写 52H 的中断入口地址
MOV     DX, 3250H    ;偏移地址
MOV     AL, 54H      ;中断类型号
INT     21H          ;写 54H 的中断入口地址

;8259A 初始化命令字
MOV     AL, 13H      ;定义 ICW1，单片，边沿触发
OUT     20H, AL
MOV     AL, 50H      ;定义 ICW2，中断号 50H~57H
OUT     21H, AL
MOV     AL, 01H      ;定义 ICW4，完全嵌套, 非缓冲, 普通 EOI
OUT     21H, AL
MOV     AL, 0EAH     ;定义 OCW1，屏蔽 IR1,3,5,6,7
OUT     21H, AL

;中断服务子程序
PUSH    AX           ；保护现场
……
STI                  ；开中断
……                   ；中断处理
CLI                  ；关中断
MOV     AL，20H          ；定义 OCW2，普通 EOI 结束命令
OUT     20H，AL
……
POP  AX              ；恢复现场
IRET                     ；中断返回
```

15. 如外设 A1、A2、A3、A4、A5 按完全嵌套优先级排列规则，外设 A1 的优先级最高，A5 最低。若中断请求的次序如下所示，试给出各外设的中断处理程序的次序。（假设所有的中断处理程序开始后就有 STI 指令）

（1）外设 A3 和 A4 同时发出中断请求；

（2）在外设 A3 的中断处理中，外设 A1 发出中断请求；

（3）在外设 A1 的中断处理未完成前，发出 EOI 结束命令，外设 A5 发出中断请求

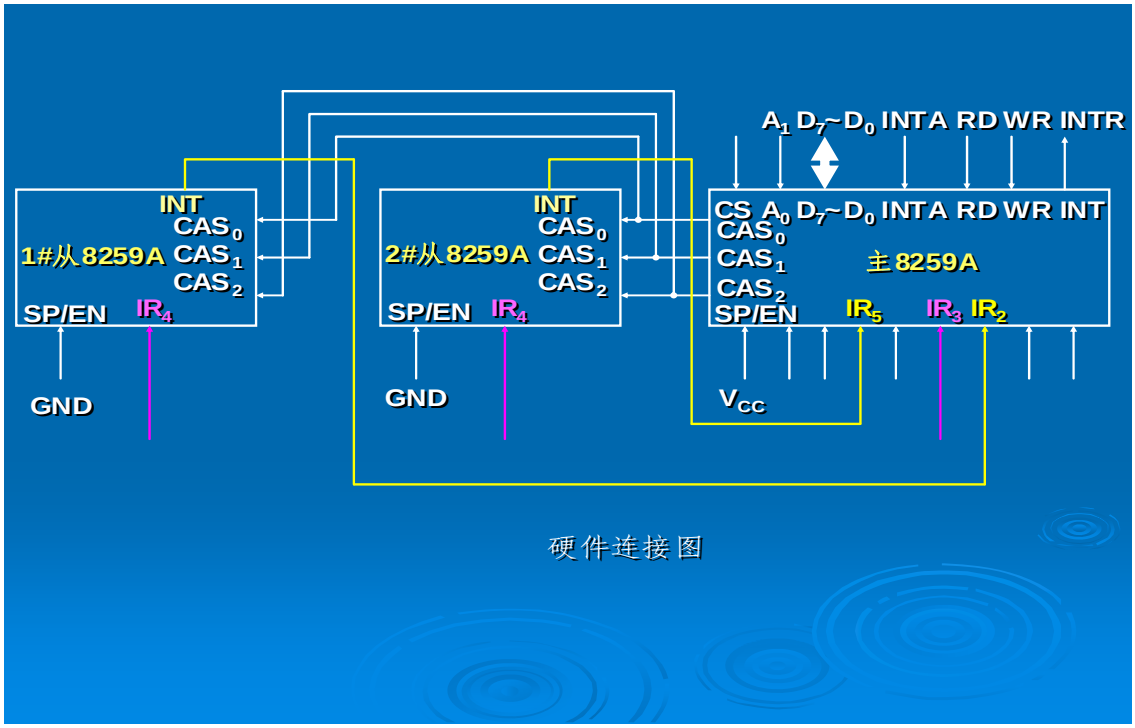
答：外设的中断处理程序的次序为：A3→A1→A3→A4→A5

16. 某系统中有 3 片 8259A 级联使用，1 片为 8259A 主片，2 片为 8259A 从片，从片接入 8259A 主片的 IR2 和 IR5 端，并且当前 8259A 主片的 IR3 及两片 8259A 从片的 IR4 各接有一个外部中断源。中断类型基号分别为 80H，90H，A0H，中断入口段地址为 2000H，偏移地址分别为 1800H，2800H，3800H，主片 8259A 的端口地址为 F8H，FAH。一片 8259A 从片的端口地址为 FCH，FEH，另一片为 FEECH，FEEEH。中断采用电平触发，全嵌套工作方式，普通 EOI 结束。

（1）画出硬件连接图。

（2）编写初始化程序。

答：



硬件连接图

（1）中断向量表形成

```
MOV     AX，2000H
MOV     DS，AX        ；DS 中为段地址
MOV     DX，1800H     ；DX 中为偏移地址
MOV     AL，83H       ；中断类型号为 83H
MOV     AH，25H
INT 21H               ；设置类型号 83H 的中断向量
MOV     DX，2800H
MOV     AL，94H
INT 21H               ；设置类型号 94H 的中断向量
```

```
    MOV     DX，3800H
    MOV     AL，0A4H
    INT 21H             ；设置类型号 A4H 的中断向量
```
(2)主片 8259A 初始化编程：端口地址为 F8H 和 FAH
```
    MOV     AL，00011001B（19H）；定义 ICW1，主片级联，电平触发
    OUT     0F8H，AL          ；发 ICW1 命令
    MOV     AL，80H           ；IR0 的中断类型号为 80H
    OUT     0FAH，AL          ；发 ICW2 命令
    MOV     AL，00100100B（24H）；定义 ICW3，IR2 和 IR5 接从片
    OUT     0FAH，AL
    MOV     AL，00000001B（01H）；定义 ICW4，完全嵌套，非缓冲
    OUT     0FAH，AL          ；非自动 EOI 结束方式
    MOV     AL，11010011B（0D3H）；定义 OCW1，允许 IR2、IR3
    OUT     0FAH，AL          ；IR5 中断，其余中断请求屏蔽
```
(3)1＃从片 8259A 初始化编程：端口地址为 FCH 和 FEH
```
    MOV     AL，00011001B（19H）；定义 ICW1，片从级联，电平触发
    OUT     0FCH，AL          ；发 ICW1 命令
    MOV     AL，90H           ；IR0 的中断类型号为 90H
    OUT     0FEH，AL          ；发 ICW2 命令
    MOV     AL，00000010B（02H）；定义 ICW3，1#从片接主片的 IR2
    OUT     0FEH，AL
    MOV     AL，00000001B（01H）；定义 ICW4，完全嵌套，非缓冲
    OUT     0FEH，AL          ；非自动 EOI 结束方式
    MOV     AL，11101111B（0EFH）；定义 OCW1，允许 IR4 中断，
    OUT     0FEH，AL          ；其余中断请求屏蔽
```
(4)2＃从片 8259A 初始化编程：端口地址为 FEECH 和 FEEEH
```
    MOV     AL，00011001B（19H）；定义 ICW1，片从级联，电平触发
    MOV     DX，0FEECH
    OUT     DX，AL      ；发 ICW1 命令
    MOV     AL，0A0H          ；IR0 的中断类型号为 A0H
    MOV     DX，0FEEEH
    OUT     DX，AL      ；发 ICW2 命令
    MOV     AL，00000101B（05H）；定义 ICW3，2#从片接主片的 IR5
    OUT     DX，AL
    MOV     AL，00000001B（01H）；定义 ICW4，完全嵌套，非缓冲
    OUT     DX，AL      ；非自动 EOI 结束方式
    MOV     AL，11101111B（0EFH）；定义 OCW1，允许 IR4 中断，
    OUT     DX，AL      ；其余中断请求屏蔽
```

第八章

1. 8253 芯片有哪几个计数通道？每个计数通道可工作于哪几种工作方式？这些操作方式的主要特点是什么？

答：8253 内部包含 3 个完全相同的计数器/定时器通道，即 0～2 计数通道，对 3 个通道的操作完全是独立的。8253 的每个通道都有 6 种不同的工作方式。

方式 0——计数结束中断方式：当对 8253 的任一个通道写入控制字，并选定工作于方式 0 时，该通道的输出端 OUT 立即变为低电平。要使 8253 能够进行计数，门控信号 GATE 必须为高电平。经过 n 十 1 个脉冲后，计数器减为 0，这时 OUT 引脚由低电平变成高电平。OUT 引脚上的高电平信号，一直保持到对该计数器装入新的计数值，或设置新的工作方式为止。在计数的过程中，如果 GATE 变为低电平，则暂停减 1 计数，计数器保持 GATE 有效时的值不变，OUT 仍为低电平。待 GATE 回到高电平后，又继续往下计数。

方式 1——可编程单稳态输出方式 ：当 CPU 用控制字设定某计数器工作于方式 1 时，该计数器的输出 OUT 立即变为高电平。GATE 出现一个上升沿后，在下一个时钟脉冲的下降沿，将 n 装入计数器的执行部件，同时，输出端 OUT 由高电平向低电平跳变。当计数器的值减为零时，输出端 OUT 产生由低到高的正跳变，在 OUT 引脚上得到一个 n 个时钟宽度的负单脉冲。在计数过程中，若 GATE 产生负跳变，不会影响计数过程的进行。但若在计数器回零前，GATE 又产生从低到高的正跳变，则 8253 又将初值 n 装入计数器执行部件，重新开始计数，其结果会使输出的单脉冲宽度加宽。

方式 2——比率发生器：当对某一计数通道写入控制字，选定工作方式 2 时，OUT 端输出高电平。如果 GATE 为高电平，则在写入计数值后的下一个时钟脉冲时，将计数值装入执行部件，此后，计数器随着时钟脉冲的输入而递减计数。当计数值减为 1 时，OUT 端由高电乎变为低电平，待计数器的值减为 0 时，OUT 引脚又回到高电平，即低电平的持续时间等于一个输入时钟周期。与此同时，还将计数初值重新装入计数器，开始一个新的计数过程，并由此循环计数。如果装入计数器的初值为 n，那么在 OUT 引脚上，每隔 n 个时钟脉冲就产生一个负脉冲，其宽度与时钟脉冲的周期相同，频率为输入时钟脉冲频率的 n 分之一。在操作过程中，任何时候都可由 CPU 重新写入新的计数值，不影响当前计数过程的进行。当计数值减为 0 时，一个计数周期结束，8253 将按新写入的计数值进行计数。在计数过程中，当 GATE 变为低电平时，使 OUT 变为高电平，禁止计数；当 GATE 从低电平变为高电平，GATE 端产生上升沿，则在下一个时钟脉冲时，把预置的计数初值装入计数器，从初值开始递减计数，并循环进行。

方式 3——方波发生器 ：方式 3 和方式 2 的工作相类似，但从输出端得到的是对称的方波或基本对称的矩形波。 如果写入计数器的初值为偶数，则当 8253 进行计数时，每输入一个时钟脉冲，均使计数值减 2。计数值减为 0 时，OUT 输出引脚由高电平变成低电平，同时自动重新装入计数初值，继续进行计数。当计数值减为 0 时，OUT 引脚又回到高电平，同时再一次将计数初值装入计数器，开始下一轮循环计数；如果写入计数器的初值为奇数，则当输出端 OUT 为高电平时，第一个时钟脉冲使计数器减 1，以后每来一个时钟脉冲，都使计数器减 2，当计数值减为 0 时，输出端 OUT 由高电平变为低电平，同时自动重新装入计数初值继续进行计数。这时第一个时钟脉冲使计数器减 3，以后每个时钟脉冲都使计数器减 2，计数值减为 0 时，OUT 端又回到高电平，并重新装入计数初值后，开始下一轮循环计数。

方式 4——软件触发选通：当对 8253 写入控制宇，进入工作方式 4 后，OUT 端输出变为高电平，如果 GATE 为高电平，那么，写入计数初值后，在下一个时钟脉冲后沿将自动把计数初值装入执行部件，并开始计数。当计数值成为 0 时，OUT 端输出变低，经过一个

时钟周期后，又回到高电平，形成一个负脉冲。若在计数过程中写入一个新的计数值，则在现行计数周期内不受影响，但当计数值回 0 后，将按新的计数初值进行计数，同样也只计一次。　　如果在计数的过程中 GATE 变为低电平，则停止计数，当 GATE 变为高电平后，又重新将初值装入计数器，从初值开始计数，直至计数器的值减为 0 时，从 OUT 端输出一个负脉冲。

方式 5——硬件触发选通：编程进入工作方式 5 后，OUT 端输出高电平。当装入计数值 n 后，GATE 引脚上输入一个从低到高的正跳变信号时，才能在下一个时钟脉冲后沿把计数初值装入执行部件，并开始减 1 计数。当计数器的值减为 0 时，输出端 OUT 产生一个宽度为一个时钟周期的负脉冲，然后 OUT 又回到高电平。计数器回 0 后，8253 又自动将计数值 n 装入执行部件，但并不开始计数，要等到 GATE 端输入正跳变后，才又开始减 1 计数。计数器在计数过程中，不受门控信号 GATE 电平的影响，但只要计数器未回 0，GATE 的上升沿却能多次触发计数器，使它重新从计数初值 n 开始计数，直到计数值减为 0 时，才输出一个负脉冲。如果在计数过程中写入新的计数值，但没有触发脉冲，则计数过程不受影响。当计数器的值减为 0 后，GATE 端又输入正跳变触发脉冲时，将按新写入的初值进行计数。

2. 8253 的最高工作频率是多少？8254 与 8253 的主要区别是什么？

答：8253 的最高计数频率能达到 2MHz。

Intel 8254 是 8253 的增强型产品，它与 8253 的引脚兼容，功能几乎完全相同，不同之处在于以下两点：

（1）8253 的最大输入时钟频率为 2MHz，而 8254 的最大输入时钟频率可高达 5MHz。

（2）8254 有读回（Read-back）功能，可以同时锁存 1～3 个计数器的计数值及状态值，供 CPU 读取，而 8253 每次只能锁存和读取一个通道的计数器，且不能读取状态值。

3. 对 8253 进行初始化编程分哪几步进行？

答：（1）写入控制字

用输出指令向控制字寄存器写入一个控制字，以选定计数器通道，规定该计数器的工作方式和计数格式。写入控制字还起到复位作用，使输出端 OUT 变为规定的初始状态，并使计数器清 0.

（2）写入计数初值

用输出指令向选中的计数器端口地址中写入一个计数初值，初值设置时要符合控制字中有关格式规定。

4. 设 8253 的通道 0～2 和控制端口地址分别为 300H，302H，304H 和 306H，定义通道 0 工作在方式 3，CLK0=2MHz。试编写初始化程序，并画出硬件连接图。要求通道 0 输出 1.5KHz 的方波，通道 1 用通道 0 的输出作计数脉冲，输出频率为 300Hz 的序列负脉冲，通道 2 每秒钟向 CPU 发 50 次中断请求。

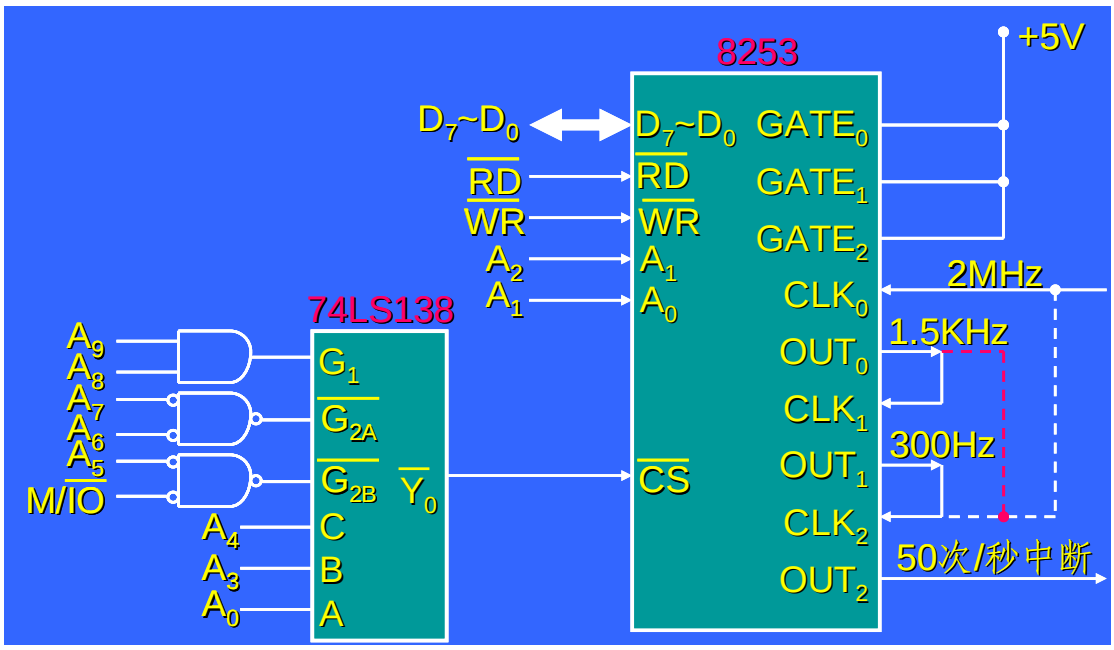
答：通道 0 工作在方式 3，n0＝2MHz/1.5KHz＝1334

通道 1 工作在方式 2，n1＝1.5KHz/300Hz＝5

通道 2 工作在方式 0，当 CLK2＝2MHz 时， n2＝2MHz/50Hz－1＝39999;

当 CLK2＝OUT0＝1.5KHz 时， n2＝1.5KHz/50Hz－1＝29 ;

当 CLK2＝OUT1＝300Hz 时， n2＝300Hz/50Hz－1＝5



初始化程序如下：

通道 0 初始化：

```
MOV     DX, 306H
MOV     AL, 00110111B(37H)          ；方式 3，先读/写低 8 位，
                                    ；后读/写低 8 位， BCD 计数
OUT     DX, AL
MOV     DX, 300H
MOV     AL, 34H                     ；初值低 8 位
OUT     DX, AL
MOV     AL, 13H                     ；初值高 8 位
OUT     DX, AL
```

通道 1 初始化：

```
MOV     DX, 306H
MOV     AL, 01010101B(55H)      ；方式 2，只读/写低 8 位， BCD 计数
OUT     DX, AL
MOV     DX, 302H
MOV     AL, 05H                     ；初值
OUT     DX, AL
```

通道 2 初始化：

```
MOV     DX, 306H
MOV     AL, 10010001B(91H)          ；方式 0，只读/写低 8 位，BCD 计数
OUT     DX, AL
MOV     DX, 304H
MOV     AL, 29H(或 05H) ；初值
OUT     DX, AL
```

5. 某微机系统中，8253 的端口地址为 40H～43H。要求通道 0 输出方波，使计算机每秒钟产生 18.2 次中断；通道 1 每隔 15 μ s 向 8237A 提出一次 DMA 请求；通道 2 输出频率为 2000Hz 的方波。试编写 8253 的初始化程序，并画出有关的硬件连接图。

答：此微机系统为 IBM PC/XT 系统，通道 0 作实时时钟，每秒钟产生 18.2 次中断。

通道 0 工作于方式 3，n0=1.19318MHz/18.2=65536 即 0

初始化编程：

```
MOV   AL, 00110110B    ;通道 0，先写低字节，后写高字节，方式 3，2 进制计数
OUT   43H, AL          ;写入控制字
MOV   AX, 0000H        ;预置计数值 n=65536
OUT   40H，AL          ；先写低字节
MOV   AL，AH
OUT   40H，AL          ；后写高字节
```

通道 1 工作于方式 2，周期为 15μs，频率为 66.2878KHz，

初值 n1=1.19318MHz/66.2878KHz=18

初始化编程：

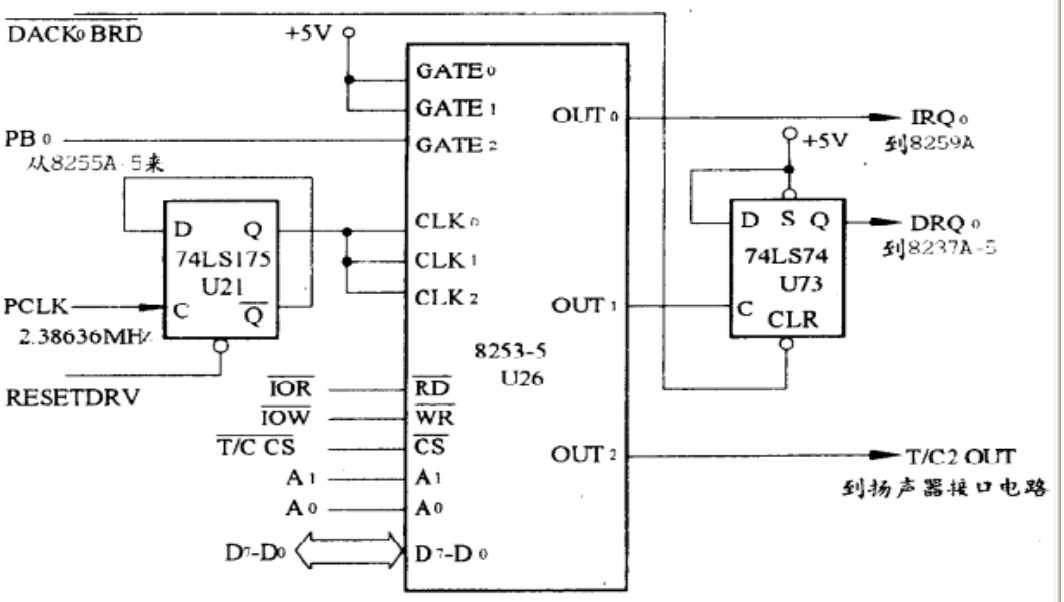
```
MOV   AL，01010101B    ；控制字，计数 1，只写低字节，方式 2，BCD 计数
OUT   43H，AL          ；写入控制字
MOV   AL，18H          ；计数初值 BCD 数 18H
OUT   41H，AL          ；送初值
```

通道 2 用于扬声器音调控制，要求输出频率 2000Hz 的方波，故工作于方式 3，

n2=1.19318MHz/2000Hz=596

初始化编程：

```
MOV   AL，10110111B    ；控制字，计数器 2，先写低字节，后写高字节，方式 3，BCD 计数
OUT   43H，AL
MOV   AX，596H         ；预置初值
OUT   42H，AL          ；先送低字节
MOV   AL，AH
OUT   42H，AL          ；后送高字节
```



8253-5在 PC/XT 机中的连线图

6. 设某系统中 8254 芯片的基地址为 F0H，在对 3 个计数通道进行初始化编程时，都设为先读写低 8 位，后读写高 8 位，试编程完成下列工作：

（1）对通道 0～2 的计数值进行锁存并读出来。

（2）对通道 2 的状态值进行锁存并读出来。

答：（1）利用 8254 的读回功能锁存计数值

```
MOV   AL，11011110B     ；锁存 3 个计数通道计数值
OUT   0F3H，AL
IN    AL，0F0H          ；读通道 0 低 8 位
MOV   AH，AL
IN    AL，0F0H          ；读通道 0 高 8 位
XCHG  AH，AL            ；将计数值置入 AX
PUSH  AX                ；入栈保存
IN    AL，0F1H          ；读通道 1 低 8 位
MOV   AH，AL
IN    AL，0F1H          ；读通道 1 高 8 位
XCHG  AH，AL
PUSH  AX                ；入栈
IN    AL，0F2H          ；读通道 2 低 8 位
MOV   AH，AL
IN    AL，0F2H          ；读通道 2 高 8 位
XCHG  AH，AL
PUSH  AX
```

（2） 利用 8254 的读回功能锁存状态

```
MOV   AL，11101000B     ；锁存通道 2 状态
OUT   0F3H，AL
IN    AL，0F2H          ；读通道 2 状态
```

第九章

1. 8255A 的 3 个端口在功能上各有什么不同的特点？8255A 内部的 A 组和 B 组控制部件各管理哪些端口？

答：端口 A 包含一个 8 位的数据输出锁存器/缓冲器，一个 8 位的数据输入锁存器，因此 A 口作输入或输出时数据均能锁存。

端口 B 包含一个 8 位的数据输入/输出锁存器/缓冲器，一个 8 位的数据输入缓冲器。

端口 C 包含一个 8 位的数据输出锁存器/缓冲器，一个 8 位的数据输入缓冲器，无输入锁存功能，当它被分成两个 4 位端口时，每个端口有一个 4 位的输出锁存器。

端口 A 和端口 C 的上半部分（PC7～PC4）由 A 组控制逻辑管理。

端口 B 和端口 C 的下半部分（PC3～PC0）由 B 组控制逻辑管理。

2. 8255A 有哪几种工作方式？各用于什么场合？端口 A、端口 B 和端口 C 各可以工作于哪几种工作方式？

答：8255A 具有 3 种基本的工作方式，在对 8255A 进行初始化编程时，应向控制字寄存器写入方式选择控制字，用来规定 8255A 各端口的工作方式。这 3 种基本工作方式是：

方式 0——基本输入输出方式：适用于不需要用应答信号的简单输入输出场合。这种方式 A 口和 B 口可作为 8 位的端口，C 口的高 4 位和低 4 位可作为两个 4 位的端口。

方式 1——选通输入输出方式：A 口和 B 口作为数据口，均可工作于输入或输出方式。端口 C 的 6 根线用来产生或接受联络信号。

方式 2——双向总线 I/O 方式：只有 A 口可以工作于这种方式。端口 A 工作于方式 2 时，端口 C 的 5 位（PC3~PC7）作 A 口的联络控制信号。

3. 8255A 的方式选择字和置位复位字都写入什么端口？用什么方式区分它们？

答：8255A 的方式选择字和置位复位控制字都被写入控制字寄存器端口中，但通过控制字的 D7 位进行区分，方式控制字的 D7=1，置位复位控制字的 D7=0。

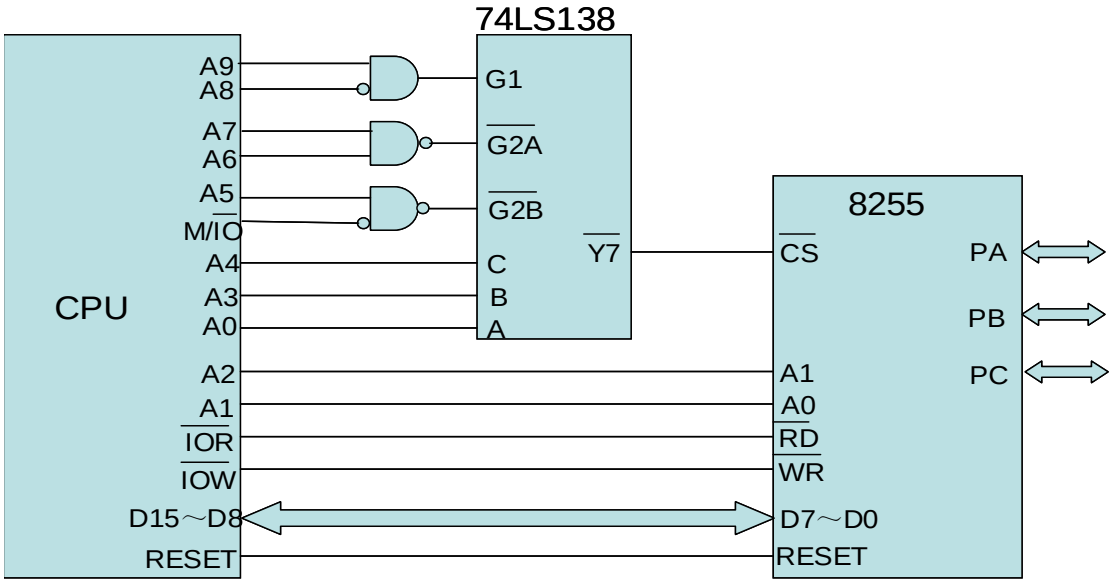
4. 若 8255A 的系统基地址为 2F9H，且各端口都是奇地址，则 8255A 的 3 个端口和控制寄存器的地址各是多少？

答：端口 A 地址：2F9H

端口 B 地址：2FBH

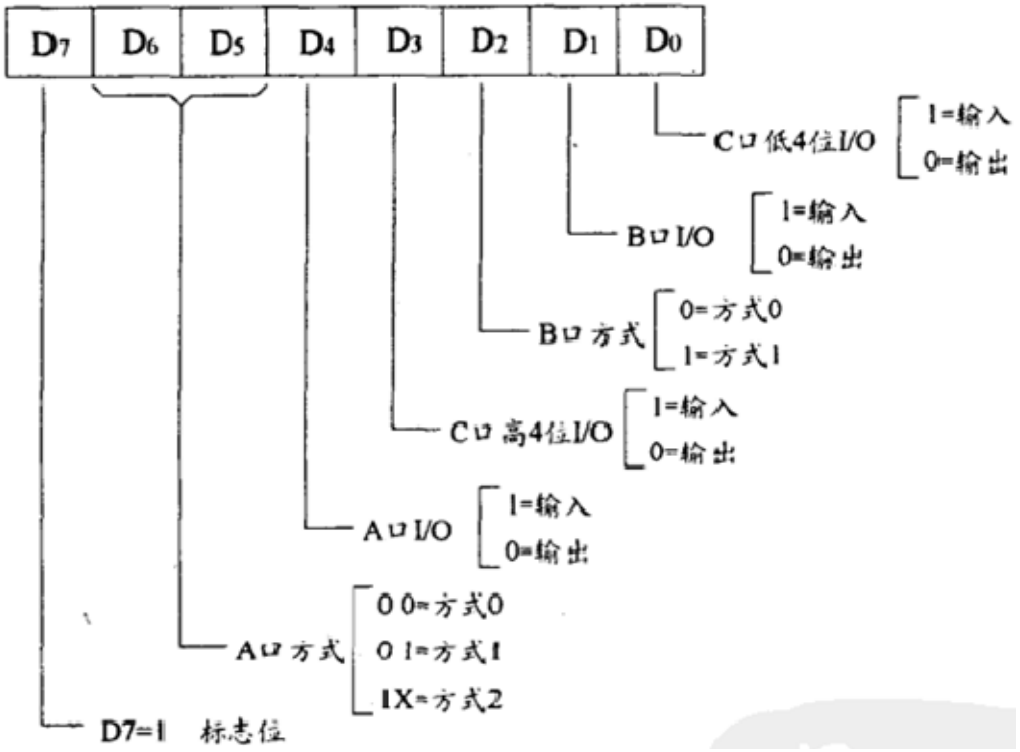
端口 C 地址：2FDH

控制字地址：2FFH



5. 设 8255A 的 A 口、B 口、C 口和控制字寄存器的端口地址分别为 80H、82H、84H 和 86H。要求 A 口工作在方式 0 输出，B 口工作在方式 0 输入，C 口高 4 位输入，低 4 位输出，试编写 8255A 的初始化程序。

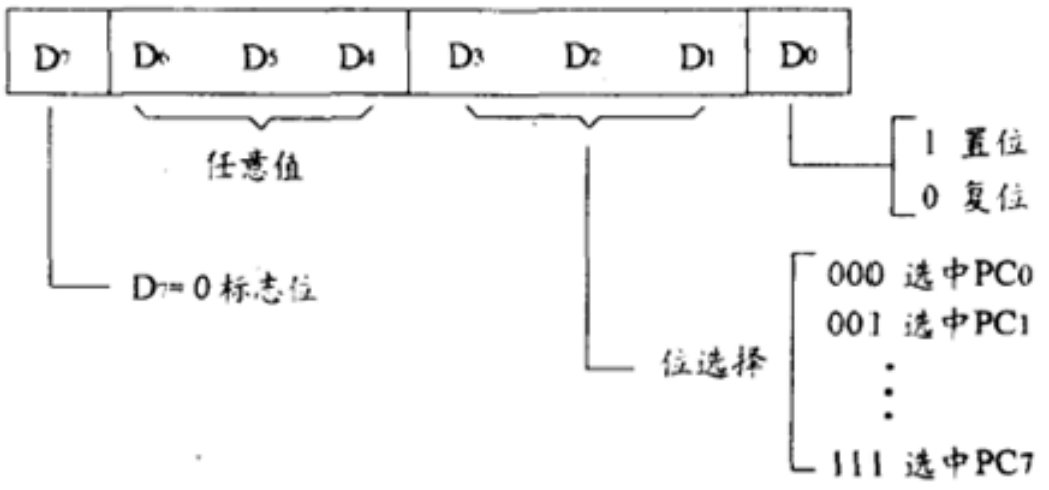
答：




```
MOV   AL，10001010B      ；方式控制字
OUT   86H，AL
```

6. 8255A 的端口地址同第 5 题，要求 PC4 输出高电平，PC5 输出低电平，PC6 输出一个正脉冲，试写出完成这些功能的指令序列。

答：




```
MOV   AL，00001001B      ；PC4 输出高电平
OUT   86H，AL
MOV   AL，00001010B      ；PC5 输出低电平
```

OUT　86H，AL
MOV　AL，000001100B　　；PC6 先输出低电平
OUT　86H，AL
MOV　AL，00001101B　　；PC6 再输出高电平
OUT　86H，AL
MOV　AL，00001100B　　；PC6 再输出低电平，形成一个正脉冲
OUT　86H，AL

7. 8255A 端口地址同第 5 题，若 A 口工作在方式 0 输入，B 口工作在方式 1 输出，C 口各位的作用是什么？控制字是什么？若 B 口工作在方式 0 输出，A 口工作方式 1 输入，C 口各位的作用是什么？控制字是什么？

答：A 口工作在方式 0，不需要 C 口做应答信号，B 口工作在方式 1 输入，需要 C 口做联络信号，PC1 连接 $\overline{\text{OBF}}$，输出缓冲器满信号，PC2 连接 $\overline{\text{ACK}}$，外设的应答信号信号，PC0 连接 INTR，向 CPU 发出的中断请求信号。控制字为 1001X10XB。

B 口工作在方式 0 输出，A 口工作方式 1 输入，需要 C 口做联络信号，PC4 连接 $\overline{\text{STB}}$，由外设输入的选通信号，PC5 连接 IBF，输入缓冲器满信号，PC3 连接 INTR，向 CPU 发出的中断请求信号。控制字为 1011X00XB。

8. 若 A 口工作在方式 2，B 口工作在方式 1 输入，C 口各位作用是什么？若 A 口工作在方式 2，B 口工作在方式 0 输出，C 口各位的作用是什么？

答：若 A 口工作在方式 2，B 口工作在方式 1 输入，C 口各位作用如下：

PC0 连接 INTR_B，PC1 连接 IBF_B，PC2 连接 $\overline{\text{STB}_B}$

PC3 连接 INTR_A，PC4 连接 $\overline{\text{STB}_A}$，PC5 连接 IBF_A，PC6 连接 $\overline{\text{ACK}_A}$，PC7 连接 $\overline{\text{OBF}_A}$

若 A 口工作在方式 2，B 口工作在方式 0 输出，C 口各位的作用如下：
PC0，PC1 和 PC2 可任意

PC3 连接 INTR_A，PC4 连接 $\overline{\text{STB}_A}$，PC5 连接 IBF_A，PC6 连接 $\overline{\text{ACK}_A}$，PC7 连接 $\overline{\text{OBF}_A}$

9. 8255A 的口地址为 80H～83H，8253 的口地址为 84H～87H，

（1）若 A 口接 8 个开关 K7～K0，B 口接 8 个指示灯 LED7～LED0，当开关合上时相应的指示灯亮，断开时灯灭，要求每隔 0.5s 检测一次开关状态，并在开关上显示出来，试画出硬件连线图，编写实现这种功能的程序。

（2）若把接在端口 A 上的开关去掉，要求接在端口 B 上的指示灯轮流熄灭，每只灯熄灭 1 秒钟，请编程实现这种功能。

答：（1）8255A：A 口输入，B 口输出

8253：2MHz/2Hz＝1000000

通道 0 工作于方式 2，取 N0＝1000

通道 1 工作于方式 0，取 N1＝999，即得 OUT1 每 0.5 秒中断一次。

本题用 8253 定时中断，中断处理时检测开关状态，并点亮相应得 LED。

假设 8259A 已初始化，主程序如下：

MOV　AX, SEG　INTR　；形成中断矢量表
MOV　DS, AX
MOV　DX, OFFSET　INTR

```
        MOV     AL, N
        MOV     AH, 25H
        INT   21H
        MOV     AL, 10010000B    ；8255 初始化
        OUT     83H,AL
        MOV     AL, 00110101B    ；通道 0 方式 2，BCD 计数
        OUT     87H,AL
        MOV     AL, 00H          ；置初值 1000
        OUT     84H,AL
        MOV     AL, 10H
        OUT     84H,AL
        MOV     AL, 01110001B    ；通道 1 方式 0，BCD 计数
        OUT     87H,AL
        MOV     AL, 99H          ；置初值 999
        OUT     85H,AL
        MOV     AL, 09H
        OUT     85H,AL
        STI
 AGAIN:HLT
        JMP AGAIN
```

中断服务程序：

```
INTR:   PUSH    AX
        STI
        IN   AL, 80H       ；检测开关，合上为 0
        NOT     AL         ；取反
        OUT     81H,AL     ；点亮相应 LED（合上）
        MOV     AL, 01110001B    ；通道 1 方式 0，BCD 计数
        OUT     87H,AL
        MOV     AL, 99H          ；置初值 999
        OUT     85H,AL
        MOV     AL, 09H
        OUT     85H,AL
        CLI
        MOV     AL, 20H          ；普通 EOI 命令
        OUT     20H,AL
        POP AX
        IRET
```

（2）8255A：B 口输出

8253：通道 0 工作于方式 2，取 N0＝2000

通道 1 工作于方式 3，取 N1＝1000，即得 OUT1 频率为 1Hz（周期为 1 秒）的方波，接到 8259A 的 IRi，用沿触发中断请求。

主程序如下：

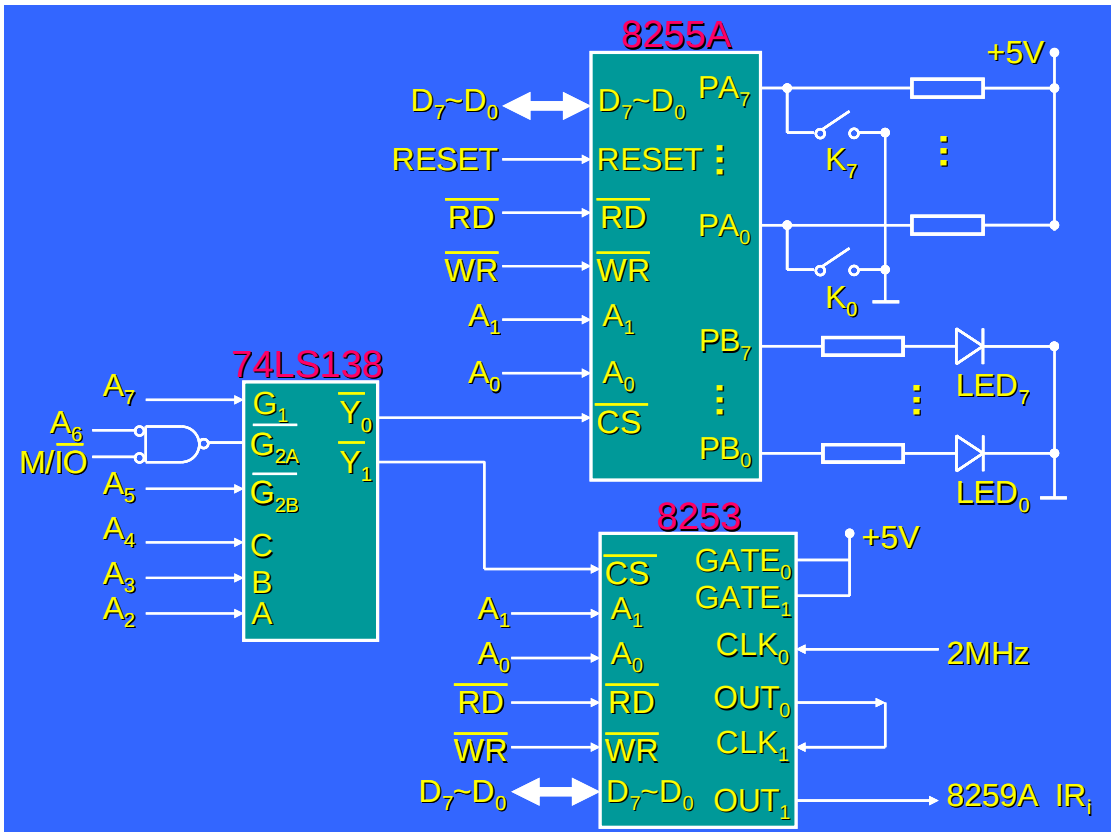
```
        MOV     AX, SEG   INTR  ；形成中断矢量表
        MOV     DS, AX
```

```
        MOV     DX, OFFSET  INTR
        MOV     AL, N
        MOV     AH, 25H
        INT 21H
        MOV     AL, 10000000B    ；8255 初始化
        OUT     83H,AL
        MOV     AL, 00110101B    ；通道 0 方式 2，BCD 计数
        OUT     87H,AL
        MOV     AL, 00H      ；置初值 2000
        OUT     84H,AL
        MOV     AL, 20H
        OUT     84H,AL
        MOV     AL, 01110111B    ；通道 1 方式 0，BCD 计数
        OUT     87H,AL
        MOV     AL, 00H      ；置初值 1000
        OUT     85H,AL
        MOV     AL, 10H
        OUT     85H,AL
        MOV     AL, 0FEH         ；熄灭 LED0
        OUT     81H,AL
        STI
  GOON:HLT
        JMP
```

中断服务程序如下：

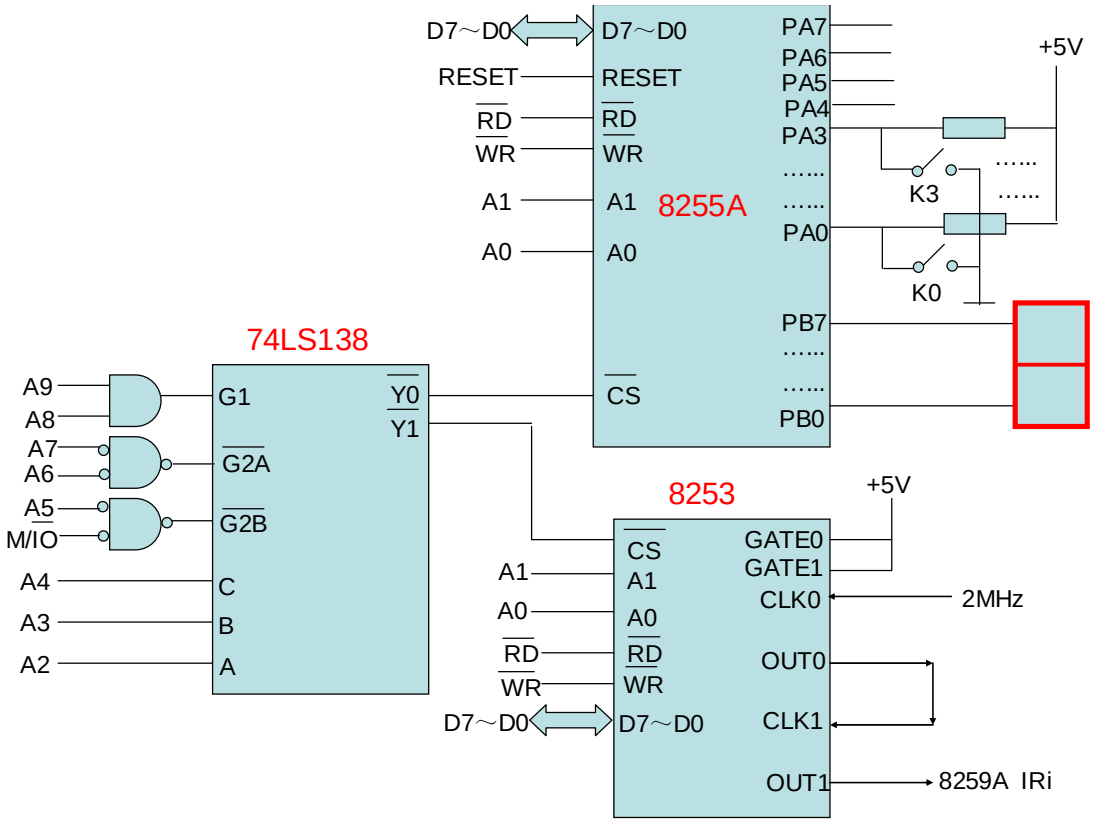
```
INTR:   PUSH    AX
        STI
        ROL     AL, 1        ；AL 左循环移位 1 位
        OUT     81H,AL       ；点亮下一位 LED
        CLI
        MOV     AL, 20H
        OUT     20H,AL       ；普通 EOI 结束命令
        POP   AX
        IRET
```

硬件连接图如下：



10. 设 8255A 的口地址为 300H～303H，A 口接 4 个开关 K3～K0，B 口接一个七段 LED 显示器，用来显示 4 个开关所拨通的 16 进制数字 0～F，开关都合上时，显示 0，都断开时显示 F，每隔 2 秒钟检测一次，试画出硬件连线图，并编写实现这种功能的程序。

答：



与上题思路相同

8255A：A 口方式 0 输入，B 口方式 0 输出，端口地址 300H～303H

8253：用于 2 秒定时，端口地址为 304H～307H

2MHz/0.5Hz＝4000000

通道 0 工作于方式 2，取 N0＝4000

通道 1 工作于方式 0，取 N1＝999，即得 OUT1 每 2 秒中断一次。

本题用 8253 定时中断，中断处理时检测开关状态，并点亮相应得 LED。

共阴极 LED 七段显示码表

十六进制数字	七段显示码	十六进制数字	七段显示码
0	3FH	8	7FH
1	06H	9	6FH
2	5BH	A	77H
3	4FH	b	7CH
4	66H	C	39H
5	6DH	d	5EH
6	7DH	E	79H
7	07H	F	71H

假设 8259A 已初始化，主程序如下：

```
TABLE   DB   3FH, 06H, 5BH, 4FH, 66H, 6DH, 7DH, 07H   ;七段码表格
        DB   7FH, 6FH, 77H, 7CH, 39H, 5EH, 79H, 71H

MOV     AX, SEG   INTR  ；形成中断矢量表
MOV     DS, AX
MOV     DX, OFFSET   INTR
MOV     AL, N
MOV     AH, 25H
INT   21H
MOV     AL, 10010000B    ；8255 初始化
MOV   DX，303H
OUT     DX, AL
MOV     AL, 00110101B    ；8253 初始化，通道 0 方式 2，BCD 计数
MOV   DX，307H
OUT     DX，AL
MOV     AL, 00H      ；置初值 4000H
MOV   DX，304H
OUT     DX, AL
MOV     AL, 40H
OUT     DX, AL
MOV     AL, 01110001B    ；通道 1 方式 0，BCD 计数
MOV   DX，307H
OUT     DX, AL
MOV     AL, 99H      ；置初值 999
MOV   DX，305H
```

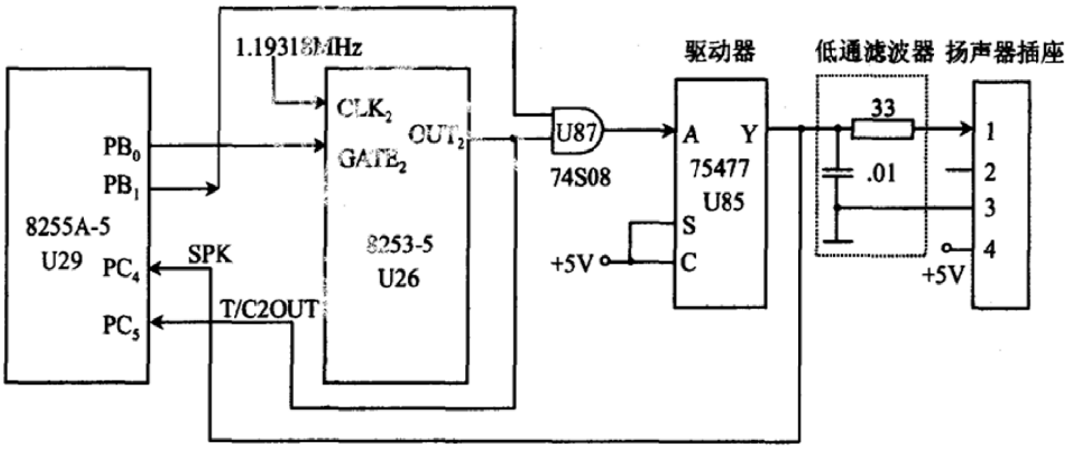
```
        OUT     DX, AL
        MOV     AL, 09H
        OUT     DX, AL
        STI
AGAIN:HLT
        JMP  AGAIN
```

中断服务程序：

```
INTR:   PUSH    AX
        STI
        MOV   DX，300H
        IN   AL, DX        ；检测开关，合上为 0
        AND     AL，0FH     ；只保留低 4 位 PA3～PA0
        MOV   BX，OFFSET TABLE
        XLAT   TABLE
        MOV   DX，301H
        OUT     DX，AL      ；点亮 LED，显示数字
        MOV     AL, 01110001B    ；通道 1 方式 0，BCD 计数
        MOV   DX，307H
        OUT     DX，AL
        MOV     AL, 99H      ；置初值 999
        MOV   DX，305H
        OUT     DX，AL
        MOV     AL, 09H
        OUT     DX，AL
        CLI
        MOV     AL, 20H      ；普通 EOI 命令
        OUT     20H,AL
        POP  AX
        IRET
```

11. 说明 PC 机中扬声器发声电路的工作原理，编写产生频率为 1000Hz 的发声程序。

答：在 PC 机中，扬声器接口电路由 8255A，8253，驱动器和低通滤波器构成，8253 是音频信号源，8255 作控制器，驱动器用来增大 8253 输出的 TTL 电平信号的驱动能力，低通滤波器将脉冲信号转换成接近正弦波的音频信号，去驱动扬声器发声。

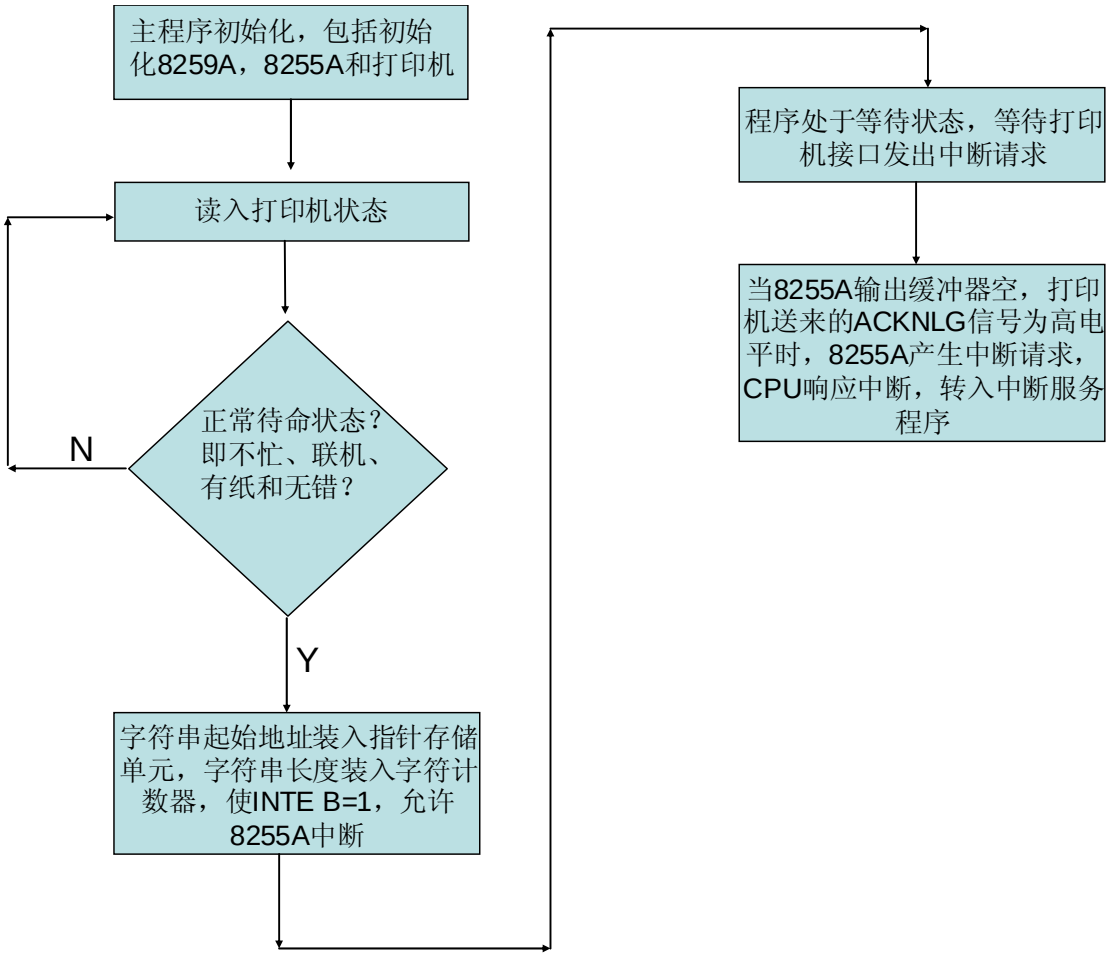


产生频率为 1000Hz 的发声程序：

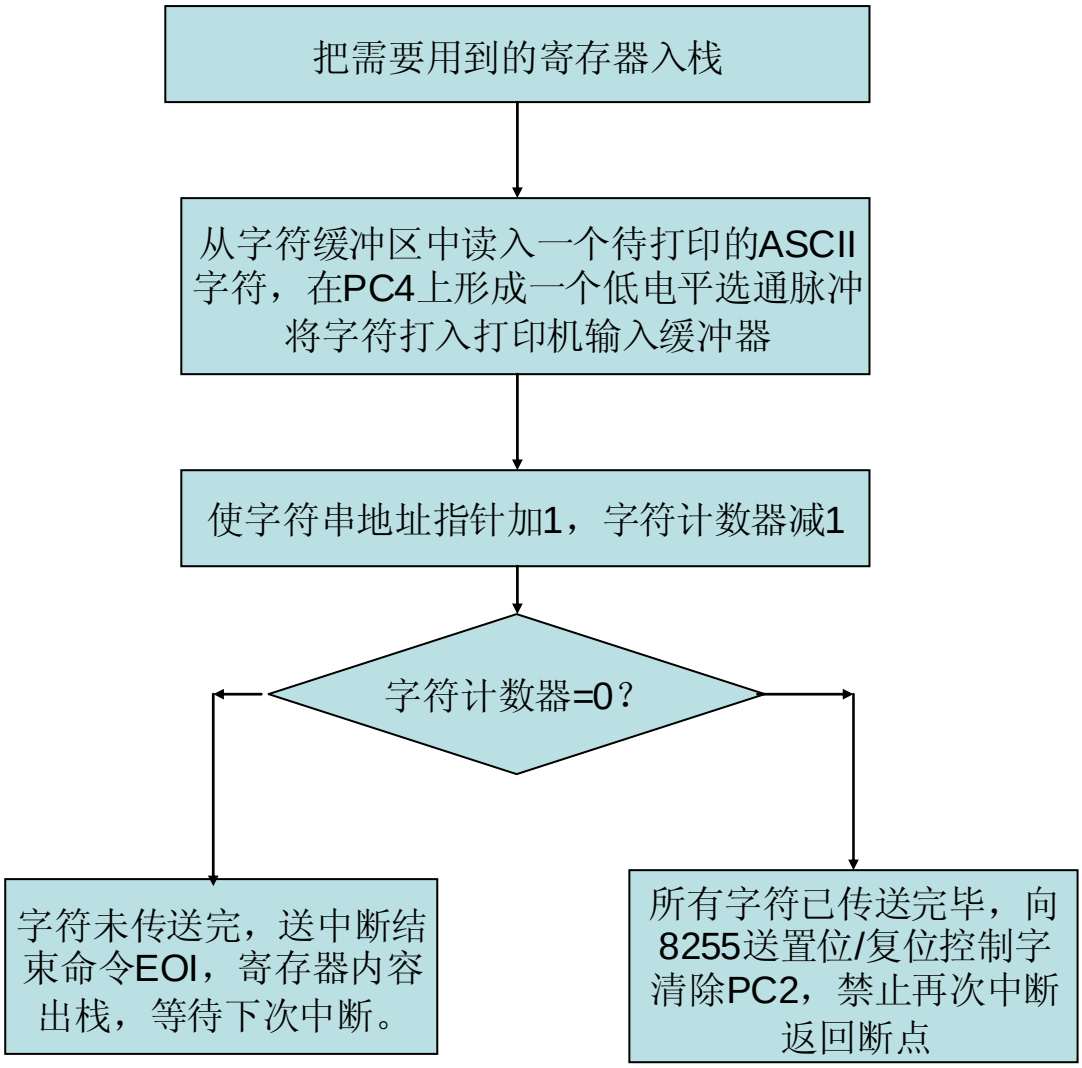
```
MOV   DI，1000          ；频率 1000Hz
MOV   AL，10110110B     ；8253 控制字，通道 2，先写低字节，后写高字节，
                        ；方式 3，二进制计数
OUT   43H，AL           ；写入控制字
MOV   DX，0012H         ；被除数高位
MOV   AX，34DEH         ；被除数低位
DIV   DI                ；求计数初值 n，结果在 AX 中
OUT   42H，AL           ；送出低 8 位
MOV   AL，AH
OUT   42H，AL           ；送出高 8 位
IN    AL，61H           ；读入 8255A 端口 B 的内容
MOV   AH，AL            ；保护 B 口的原状态
OR    AL，03H           ；使 B 口后两位置 1，其余位保留
OUT   61H，AL           ；接通扬声器，发声
```

12. 试画出打印机驱动程序的流程图。

答：



中断服务程序负责将一个字符送到打印机接口



第十章

1. 串行通信与并行通信的主要区别是什么？各有什么优缺点？

答：计算机与外部的信息交换称为通信，基本的通信方式有两种，并行通信和串行通信。

并行通信：数据各位同时传送，此方式传输数据的速度快，但使用的通信线多，若要并行传送 8 位数据，需要用 8 根数据线，另外还需一些控制信号线。随着传输距离的增加，通信线成本的增加将成为突出的问题，而且传输的可靠性随着距离的增加而下降。因此，并行通信适用于近距离传送数据的场合。

串行通信：将要传送的数据或信息按一定的格式编码，然后在单根线上按一位接一位的先后顺序进行传送。发送完一个字符后，再发送第二个。接收数据时，每次从单根线上一位接一位的接收信息，再把它们拼成一个字符，送给 CPU 作进一步处理。适用于远距离通信，需要的通信线少和传送距离远等优点。

2. 在串行通信中，什么叫单工、半双工、全双工工作方式？

答：串行通信时，数据在两个站 A 与 B 之间传送，按传送方向分成单工、半双工和全双工三种方式。

单工数据线仅能在一个方向上传输数据，两个站之间进行通信时，一边只能发送数据，另一边只能接收数据，也称为单向通信。

在半双工方式中，数据可在两个设备之间向任一个方向传输，但两个设备之间只有一根传输线，故同一时间内只能在一个方向上传输数据，不能同时收发。

全双工：对数据的两个传输方向采用不同的通路，可以同时发送和接收数据。

3. 什么叫同步工作方式？什么叫异步工作方式？哪种工作方式的效率更高？为什么？

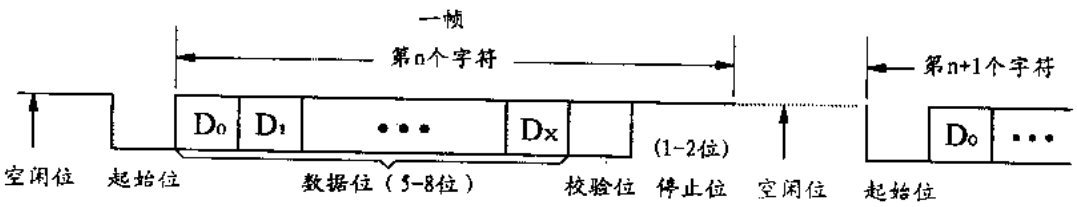
答：串行通信有两种基本工作方式：异步方式和同步方式

异步方式：不发送数据时，数据信号线总是呈现高电平，称为 MARK 状态，也称空闲状态。当有数据要发送时，数据信号线变成低电平，并持续一位的时间，用于表示字符的开始，称为起始位。起始位后，在信号线上依次出现待发送的每一位字符数据，最低有效位 D0 最先送出，根据不同编码，有效数据位可由 5 位、6 位、7 位或 8 位构成，数据位后面有一个奇偶校验位，校验位后至少有一位高电平表示停止位，用于指示字符的结束。由此可见，异步方式发送一个 7 位的 ASCII 码时，实际需发送 10 位、10.5 位或 11 位信息，故影响传输效率。

同步方式：没有数据传送时，传输线处于 MARK 状态，为了表示数据传输的开始，发送方式发送一个或两个特殊字符，称为同步字符。当发送法和接收方达到同步后，就可以一个字符接一个字符发送一大块数据，不再需要用起始位和停止位了，这样就可以明显的提高数据的传输速率。同步方式传送数据时，在发送过程中，收发双发还必须用同一个时钟进行协调，用于确定串行传输中每一位的位置。接收数据时，接受方可利用同步字符将内部时钟与发送方保持同步，然后将同步字符后面的数据逐位移入，并转换成并行格式，供 CPU 读取，直至收到结束符为止。

4. 用图表示异步串行通信数据的位格式，标出起始位，停止位和奇偶校验位，在数字位上标出数字各位发送的顺序。

答：



5. 什么叫波特率？什么叫波特率因子？常用的波特率有哪些？

答：在串行通信中，波特率表示数据传送的速率，每秒钟内所传送数据的位数称为波特率，单位为波特 bps（Bd）。

在波特率指定后，输入移位寄存器/输出移位寄存器在接收时钟/发送时钟控制下，按指定的波特率速度进行移位。一般几个时钟脉冲移位一次。要求：接收时钟/发送时钟是波特率的 16、32 或 64 倍。波特率因子就是发送／接收 1 个数据（1 个数据位）所需要的时钟脉冲个数，其单位是个／位。

常用的波特率为 110，300，600，1200，2400，4800，9600，19200，28800，36400，57600 波特。

6. 若某一终端以 2400 波特的速率发送异步串行数据，发送 1 位需要多少时间？假设一个字符包含 7 个数据位、1 个奇偶校验位、1 个停止位，发送 1 个字符需要多少时间？

答：1/2400=0.416ms，

一个字符包含 7 个数据位、1 个奇偶校验位、1 个停止位，1 个起始位，发送 1 个字符需要 10/2400=4.16ms

7. 什么叫 UART？什么叫 USART？列举典型芯片的例子。

答：仅用于异步通信的接口芯片，称为通用异步收发器 UART（Universal Asynchronous Receiver-Transmitter），典型芯片如 INS 8250。

既可以工作于异步方式，又可工作于同步方式，称为通用同步异步收发器 USART（Universal Synchronous-Asynchronous Receiver-Transmitter），典型芯片如 Intel 8251A。

8. 什么叫 MODEM？用标准电话线发送数字数据为什么要用 MODEM？调制的形式主要有哪几种？

答：能将数字信号转换成音频信号及将音频信号恢复成数字信号的器件称为调制解调器，即 MODEM。

标准电话线只能传送带宽为 300Hz～3000Hz 的音频信号，不能直接传送频带很宽的数字信号，为了解决此问题，在发送数据时，先把数字信号转换成音频信号后，称为调制，再利用电话线进行传输，接收数据时又将音频信号恢复成数字信号，称为解调。

调制的形式主要有：

幅度（Amplitude）调制或幅移键控 ASK（Amplitude-Shift Keying）简称“调幅”

频率键移 FSK（Frequency-Shift Keying，简称“调频”）

相位键移 PSK（Phase-Shift Keying，简称“调相”）

多路载波（Multiple Carrier）

9. 若 8251A 以 9600 波特的速率发送数据，波特率因子为 16，发送时钟 $\overline{\text{TxC}}$ 频率为多少？

答：发送时钟 $\overline{\text{TxC}}$ 频率=9600*16=153600Hz

10. 8251A 的 SYNDET/BRKDET 引脚有哪些功能？

答：SYNDET/BRKDET(Synchronous Detect/Break Detect)　同步检测/断点检测，输出/输入，高电平有效。

（1）8251A 工作于同步方式该引脚表示 SYNDET，内同步时该引脚为输出，有效状态（高电平）表示 8251A 已经检测到同步字符；外同步时该引脚为输入，由该引脚输入同步脉冲，上升沿启动 8251A 接收数据。

（2）8251A 工作于异步方式该引脚表示 BRKDET，是输出信号，有效状态（高电平）表示接收端检测到间断点（编程规定长度的全“0”字符）；恢复正常数据接收时该引脚被复位。

11. 如果系统中无 MODEM，8251A 与 CPU 之间有哪些连接信号？

答：8251A 和 CPU 之间的连接信号可以分为四类：

（1）片选信号

$\overline{CS}$：片选信号，它由 CPU 的地址信号通过译码后得到。

（2）数据信号

D0～D7： 8 位，三态，双向数据线，与系统的数据总线相连。传输 CPU 对 8251A 的编程命令字和 8251A 送往 CPU 的状态信息及数据。

（3）读 / 写控制信号

$\overline{RD}$：读信号，低电平时， CPU 当前正在从 8251A 读取数据或者状态信息。

$\overline{WR}$：写信号，低电乎时， CPU 当前正在往 8251A 写入数据或者控制信息。

$C/\overline{D}$：控制 / 数据信号，用来区分当前读 / 写的是数据还是控制信息或状态信息。该信号也可看作是 8251A 数据口 / 控制口的选择信号。

（4）收发联络信号

TXRDY：发送器准备好信号，用来通知 CPU， 8251A 已准备好发送一个字符。

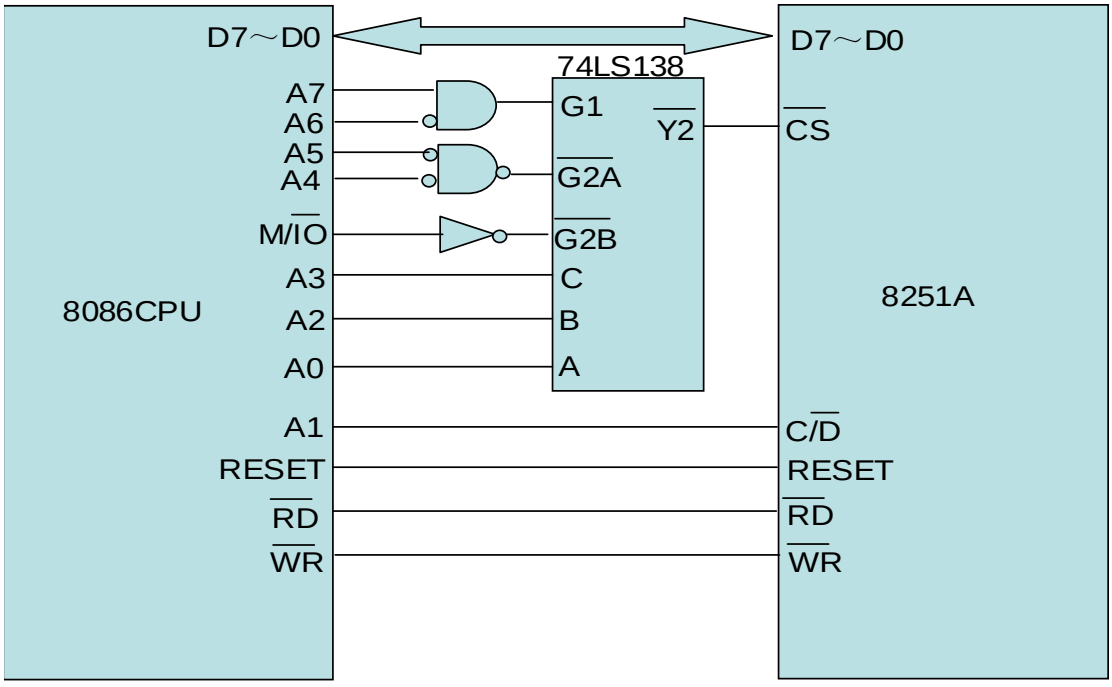
TXE ：发送器空信号， TXE 为高电平时有效，用来表示此时 8251A 发送器中并行到串行转换器空，说明一个发送动作已完成。

RXRDY ：接收器准备好信号，用来表示当前 8251A 已经从外部设备或调制解调器接收到一个字符，等待 CPU 来取走。因此，在中断方式时， RXRDY 可用来作为中断请求信号；在查询方式时， RXRDY 可用来作为查询信号。

SYNDET ：同步检测信号，只用于同步方式。

12. 在一个以 8086 为 CPU 的系统中，若 8251A 的数据端口地址为 84H，控制口和状态口的地址为 86H，试画出地址译码电路、数据总线和控制总线的连接图。

答：



13. 设 8251A 的端口地址如题 12，要求 8251A 工作于内同步方式，同步字符为 2 个，用偶校验，7 个数据位，试对 8251A 进行初始化编程。

答：
```
MOV   AL，0
OUT   86H，AL
OUT   86H，AL
OUT   86H，AL      ；向控制口写入三个 0
MOV   AL，40H
OUT   86H，AL      ；写入复位字
MOV   AL，00111000B
OUT   86H，AL      ；写入方式字
MOV   AL，SYSN   ；同步字符
OUT   86H，AL
OUT   86H，AL      ；两个同步字符
MOV   AL，10010101B
OUT   86H，AL      ；送命令字
```

14. 若 8251A 的端口地址为 FF0H，FF2H，要求 8251A 工作于异步工作方式，波特率因子为 16，有 7 个数据位，1 个奇校验位，1 个停止位，试对 8251A 进行初始化编程。

答：
```
MOV   AL，0
MOV   DX，0FF2H
OUT   DX，AL
OUT   DX，AL
OUT   DX，AL       ；向控制口写入三个 0
MOV   AL，40H
OUT   DX，AL       ；写入复位字
MOV   AL，01011010B
OUT   DX，AL       ；写入方式字
MOV   AL，00010101B
OUT   DX，AL       ；写入命令字
```

15. RS-232C 的逻辑高电平与逻辑低电平的范围是多少？怎么与 TTL 电平的器件相连？规定用什么样的接插件？最少用哪几根信号线进行通信？

答：逻辑高电平：有负载时为-3V～-15V，无负载时为-25V

逻辑低电平：有负载时为+3V～+15V，无负载时为+25V

通常用±12V 作为 RS-232C 的电平。

计算机及其接口芯片多采用 TTL 电平，即 0～0.8V 为逻辑 0，+2.0V～+5V 为逻辑 1，与 RS-232C 电平不匹配，必须设计专门的电路进行电平转换，常用的电平转换电路为 MAX232 和 MAX233。

RS-232C 使用 25 芯的 D 型插头插座和 9 芯的 D 型接插件。

常用的信号线有：TxD 发送数据，RxD 接收数据，$\overline{\text{RTS}}$ 请求发送，$\overline{\text{CTS}}$ 清除发送，$\overline{\text{DSR}}$ 数据装备准备好等信号。

16. 某微机系统用串行方式接收外设送来的数据，再把数据送到 CRT 去显示，若波特率为 1200，波特率因子为 16，用 8253 产生收发时钟，系统时钟频率为 5MHz，收发数据个数为 COUNT，数据存放到数据段中以 BUFFER 为始址的内存单元中。8253 和 8251A 的基地址

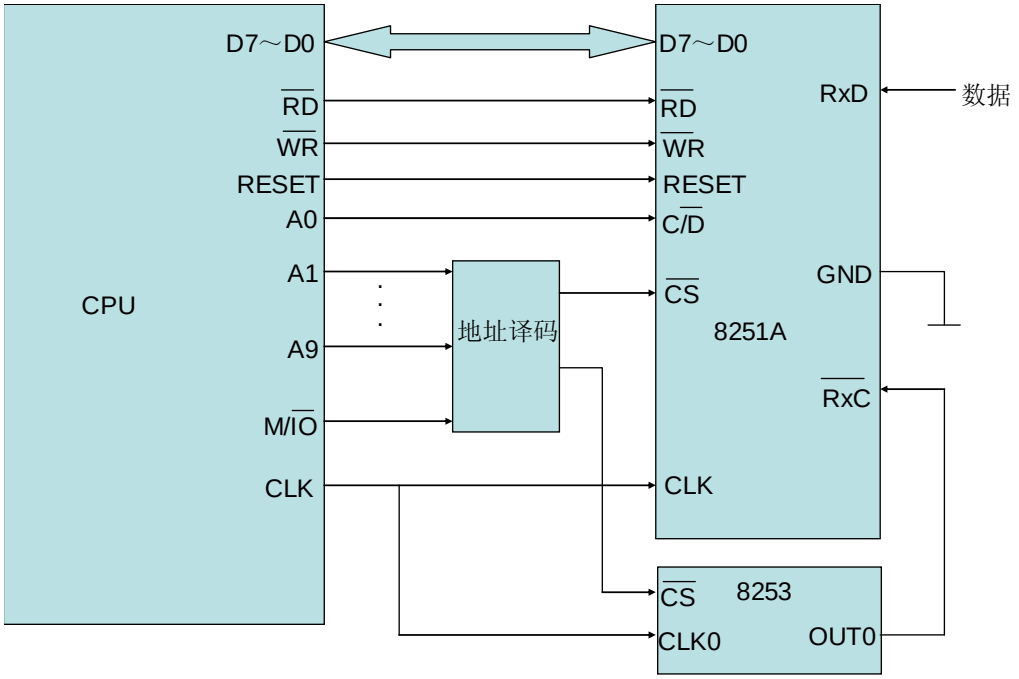
分别为 300H 和 304H。

（1）画出系统硬件连线图。

（2）编写 8253 和 8251A 的初始化程序。

（3）编写接收数据的程序。

答：（1）8251 的接收时钟 $\overline{RxC}$ =1200*16=19200Hz，用 8253 产生收发时钟，使用通道 0 产生频率为 19200Hz 的方波，故计数初值为 n0=5MHz/19200Hz≈260



（2）8253 的初始化程序

```
    MOV   DX，303H
    MOV   AL，00110111B
    OUT   DX，AL          ；8253 控制字，通道 0，方式 3，BCD 计数
    MOV   DX，300H
    MOV   AL，60H         ；初值低字节
    OUT   DX，AL
    MOV   AL，02H         ；初值高字节
    OUT   DX，AL
    8251A 的初始化程序
    MOV   AL，0
    MOV   DX，305H
    OUT   DX，AL
    OUT   DX，AL
    OUT   DX，AL          ；向控制口写入 3 个 0
    MOV   AL，40H         ；复位字
    OUT   DX，AL
    MOV   AL，01111010B ；方式字，异步，1 停止位，偶校验，7 位数据
    OUT   DX，AL
    MOV   AL，14H
```

```
    OUT   DX，AL      ；输出命令字，清错误标志，允许接收
```
（3）接收数据的程序

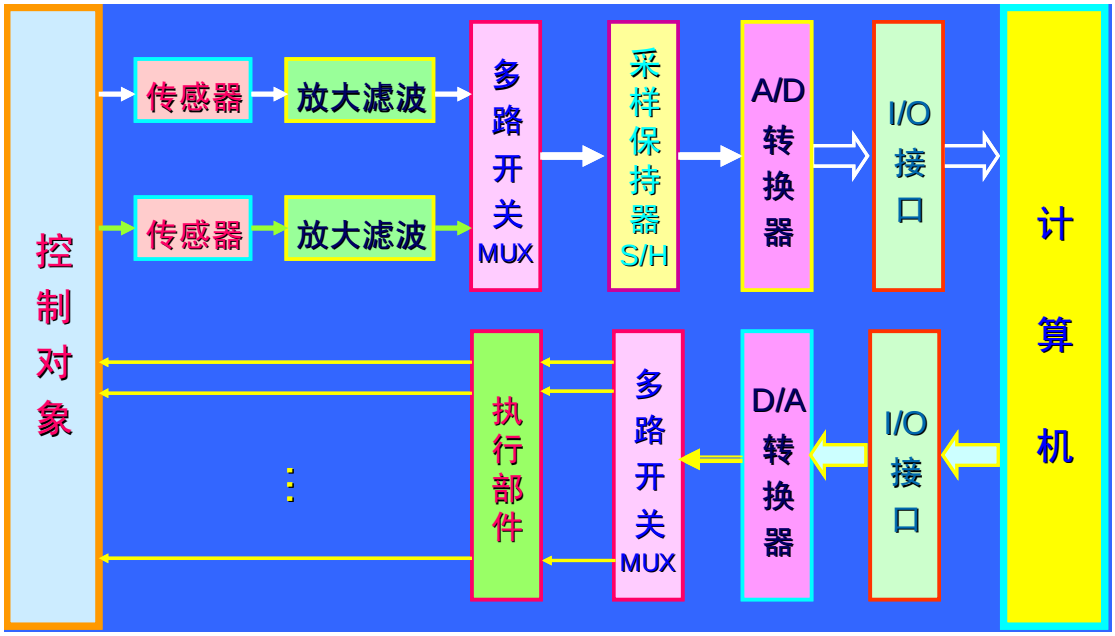
采用查询方式，异步传送

```
        LEA   DI，BUFFER      ；接收数据缓冲区
        MOV   CX，COUNT     ；接收数据个数
        MOV   DX，305H
NEXT：IN    AL，DX            ；读入状态字
        TEST   AL，02H         ；RxRDY 有效吗
        JZ    NEXT             ；否，循环等待
        TEST   AL，38H         ；是，检查是否有错
        JNZ   ERROR            ；有错，转出错处理程序
        MOV   DX，303H         ；数据端口
        IN    AL，DX           ；无错，读入一个数据
        MOV   [DI]，AL         ；送缓冲区
        INC    DI
        MOV   DX，305H
        LOOP   NEXT            ；数据没传完则继续
         …..                   ；完成
ERROR：……                     ；出错处理
```

第十一章

1. 包含 A/D 和 D/A 的实时控制系统主要由哪几部分组成？什么情况下要用多路开关？什么时候要用采样保持器?

答：



对多个变化较为缓慢的模拟信号进行 A/D 转换时，利用多路开关将各路模拟信号轮流与 A/D 转换器接通，使一个 A/D 转换器能完成多个模拟信号的转换，节省硬件开销。

一个模数转换器完成一次模数转换，要进行量化、编码等操作，每种操作均需化费一定的时间，这段时间称为模数转换时间 tc。在转换时间 tc 内, 输入模拟信号 x(t)变化速率较高时，在转换过程中，输入模拟量有一个可观的△x，结果将会引入较大的误差。也就是说，在 A/D 转换过程中，加在转换器上的电平在波动，这样，就很难说输出的数字量表示 tc 期间输入信号上哪一点的电压值，在这种情况下就要用采样保持器来解决这个问题。

2. 什么叫采样、采样率、量化、量化单位？12 位 D/A 转换器的分辨率是多少？

答：采样就是按相等的时间间隔 t 从电压信号上截取一个个离散的电压瞬时值，t 越小，采样率 f_s 越高。

对一个被采样的信号电压的幅度变化范围进行分层，确定某一个采样电压所在的层次，该分层的起始电平就是该采样的数字量，此过程称为量化，每个分层所包含的最大电压值与最小电压值之差，称为量化单位，用 q 表示，量化单位越小，精度越高。

12 位 D/A 转换器，2^n=4096，其分辨率为 1/4096*FSR=0.0244%FSR

3. 某一 8 位 D/A 转换器的端口地址为 220H，已知延时 20ms 的子程序为 DELAY_20MS，参考电压为+5V，输出信号（电压值）送到示波器显示，试编程产生如下波形：

（1）下限为 0V，上限为+5V 的三角波

（2）下限为 1.2V，上限为 4V 的梯形波。

答：（1）由于 1LSB=5V/256=0.019V，所以下限电压对应的数据为

0/0.019V=0

上限电压对应的数据为

5V/0.019V=256

程序段如下：

```
BEGIN：MOV   AL，0       ；下限值
       MOV   DX，220H
UP：    OUT   DX，AL     ；D/A 转换
       CALL  DELAY
       INC   AL          ；数值增 1
       CMP   AL，00H     ；超过上限了吗？
       JNZ    UP         ；没有，继续转换
       DEC   AL          ；超过了，数值减量
DOWN：OUT   DX，AL     ；D/A 转换
       CALL  DELAY
       DEC   AL          ；数值减 1
       CMP   AL，00H     ；低于下限了吗？
       JNZ   DOWN        ；没有
       JMP   BEGIN       ；低于，转下一个周期
```

（2）下限电压对应的数据为 1.2V/0.019V=61=3DH

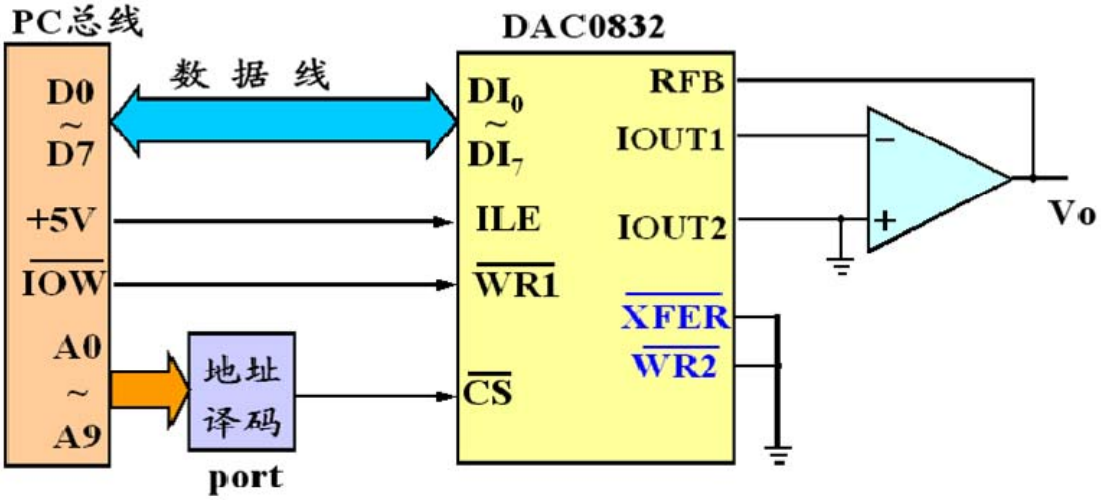
上限电压对应的数据为 4V/0.019V=205=0CDH

产生梯形波的程序如下：

```
BEGIN：MOV   AL，3DH      ；下限值
       MOV   DX，220H
UP：    OUT   DX，AL      ；D/A 转换
       CALL  DELAY
       INC   AL          ；数值增 1
       CMP   AL，0CDH     ；到达上限了吗？
       JNZ   UP          ；没有，继续转换
       OUT   DX，AL
       CALL  DELAY_20MS ；到达上限延时输出方波
       DEC   AL
DOWN：OUT   DX，AL      ；D/A 转换
       CALL  DELAY
       DEC   AL          ；数值减 1
       CMP   AL，3DH     ；到达下限了吗？
       JNZ   DOWN        ；没有，继续
       OUT   DX，AL
       CALL  DELAY_20MS ；到达下限延时输出方波
       INC   AL
       JMP   UP
```

4. 利用 DAC0832 产生锯齿波，试画出硬件连线图，并编写有关的程序。

答：



设下限为 1.2V，上限为 4V，端口地址为 300H

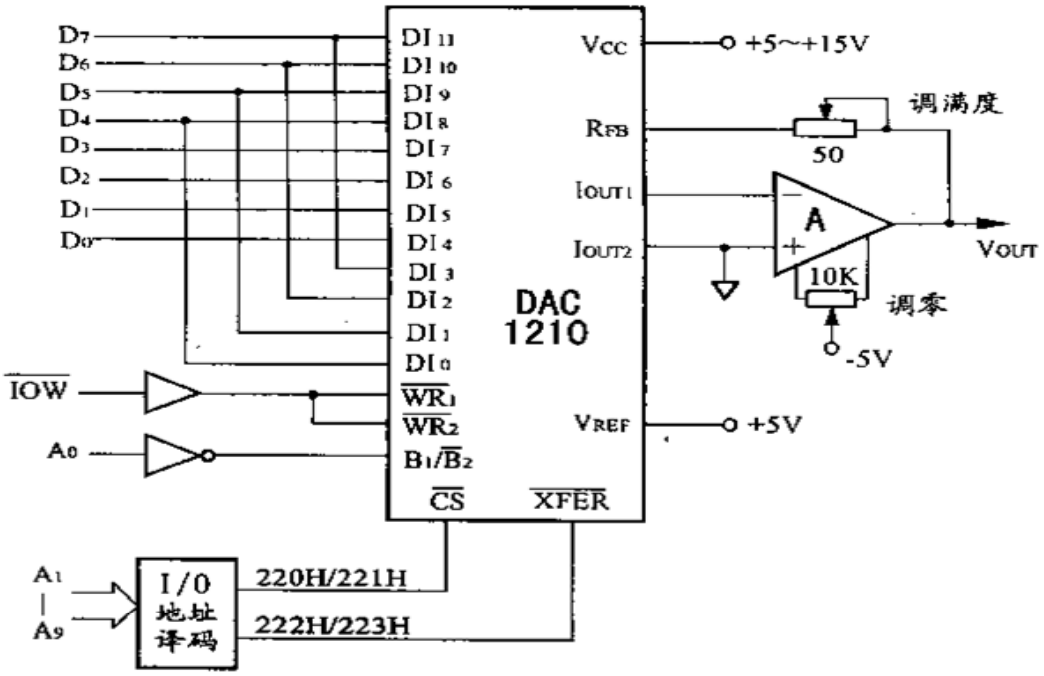
产生锯齿波的程序如下：

```
BEGIN：MOV   AL，3CH
       MOV   DX，300H
AGAIN：INC   AL
       OUT   DX，AL      ；D/A 转换
       CALL  DELAY
       CMP   AL，0CDH
       JNZ   AGAIN
       JMP   BEGIN
```

5. （1）画出 DAC1210 与 8 位数据总线的微处理器的硬件连接图，若待转换的 12 位数字是存在 BUFF 开始的单元中，试编写完成一次 D/A 转换的程序。

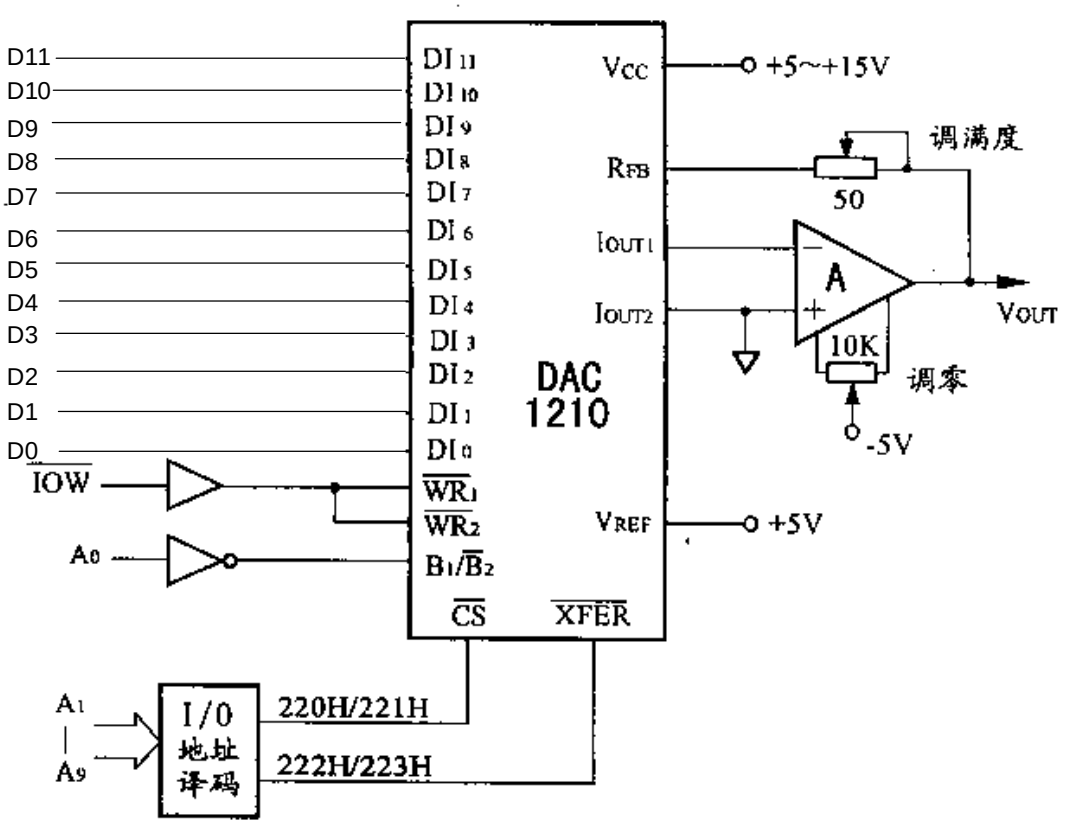
（2）将 DAC1210 与具有 16 位数据总线的 8086 相连，其余条件同（1），画出该硬件连线和编写 D/A 转换程序。

答：（1）



```
START：MOV   DX，220H      ；指向 220H 端口
       MOV   CL，4         ；移位次数
       MOV   BX，BUFF      ；取要转换的数据
       SHL   BX，CL        ；BX 中数左移 4 次后向左对齐
       MOV   AL，BH        ；取高 8 位
       OUT   DX，AL        ；写入 8 位输入寄存器
       INC   DX            ；口地址为 221H
       MOV   AL，BL        ；取低 4 位
       OUT   DX，AL        ；写入 4 位输入寄存器
       INC   DX            ；口地址为 222H
       OUT   DX，AL        ；启动 D/A 转换，AL 中可为任意值
```

（2）



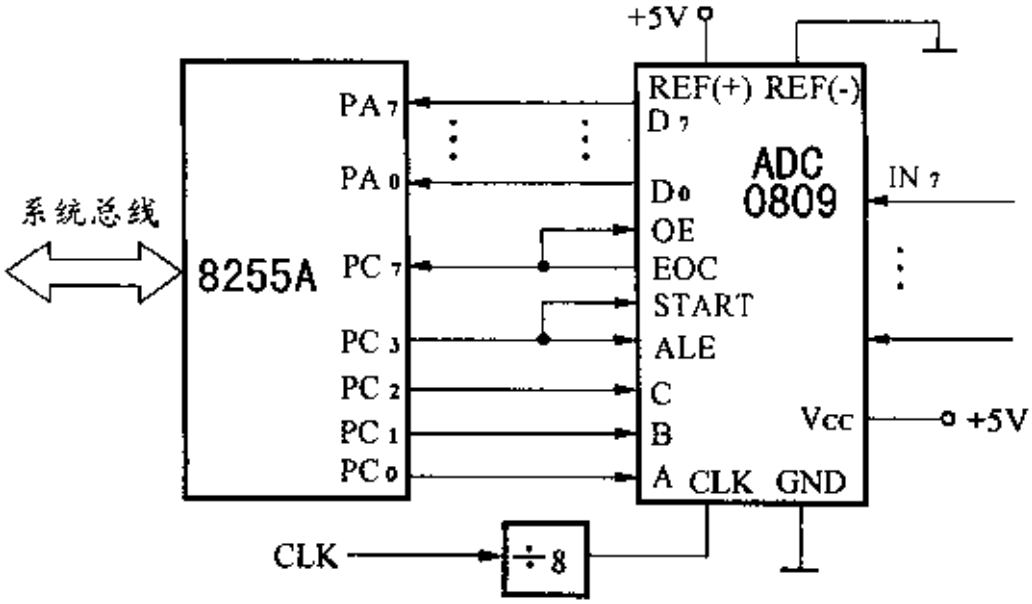

```
START：MOV   DX，220H      ；指向 220H 端口
       MOV   AX，BUFF      ；取要转换的数据
       OUT   DX，AX        ；写入 8 位输入寄存器
       INC   DX
       INC   DX            ；口地址为 222H
       OUT   DX，AL        ；启动 D/A 转换，AL 中可为任意值
```

6. 利用 8255A 和 ADC0809 等芯片设计 PC 机上的 A/D 转换卡，设 8255A 的口地址为 3C0H～3C3H，要求对 8 个通道各采集 1 个数据，存放到数据段中以 D_BUF 为始址的缓冲器中，试完成以下工作：

（1）画出硬件连接图。

（2）编写完成上述功能的程序。

答：（1）



（2）AD_SUB PROC NEAR
MOV CX，8 ；CX 作数据计数器
MOV BL，00H ；模拟通道号存在 BL 中
LEA DI，D_BUF ；缓冲区
NEXT_IN： MOV DX，3C2H ；8255A 端口 C 地址
MOV AL，BL
OUT DX，AL ；输出通道号
MOV DX，3C3H ；指向控制口
MOV AL，00000111B ；PC3 置 1
OUT DX，AL ；送出开始启动信号
NOP ；延时
NOP
NOP
MOV AL，00000110B ；PC3 复位
OUT DX，AL ；送出结束启动信号
MOV DX，3C2H ；C 口
NO_CONV： IN AL，DX ；读入 C 口内容
TEST AL，80H ；PC7，EOC 信号
JNZ NO_CONV ；PC7=1，未开始转换，等待
NO_EOC： IN AL，DX ；PC7=0，已启动转换
TEST AL，80H ；再查 PC7
JZ NO_EOC ；PC7=0，转换未结束，等待
MOV DX，3C0H ；PC7=1，转换结束，DX 指向 A 口
IN AL，DX ；读入数据
MOV [DI]，AL ；存入缓冲区
INC DI
INC BL ；指向下个通道
LOOP NEXT_IN

RET

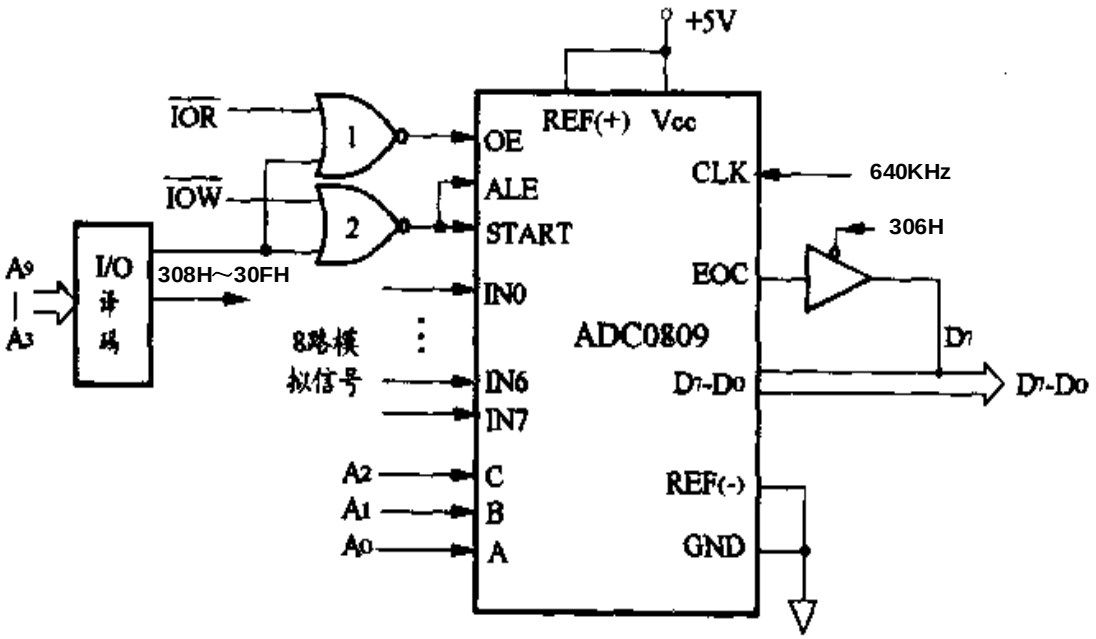
AD_SUB ENDP

7. 试利用 ADC0809，8253 和 8259A 等芯片设计 8 通道 A/D 转换电路。系统中用 8253 作定时器，采用中断方式控制采样率，采样率为 500Hz。设 8253 的通道 0 输入时钟脉冲为 2MHz，输出端 OUT0 接 8259A 的 IR2，8253 的口地址为 300H～303H，8259A 的口地址为 304H 和 305H，ADC0809 的 8 个输入通道的口地址为 308H～30FH，查询 EOC 信号和状态口地址为 306H，ADC0809 的输入时钟频率为 640KHz，A/D 转换的结果依次存入数据段中以 BUFFER 为始址的内存中，从通道 0 开始先存入各通道的第一个数据，再存放第二个数据，采集 10 秒钟后停止工作。要求：

（1）画出硬件连线图，可以不画具体的译码电路。

（2）编写 8253，8259A（只需写入中断屏蔽字）的初始化程序及采集 8 路模拟信号的中断服务程序。

答：（1）



（2）因为 8253 的时钟输入频率为 2MHz，而要求的采样频率 f=500Hz，即用 8253 定时，每隔 2ms 中断一次，因此 8253 的分频次数（时间常数）N=2MHz/500Hz=4000。

采集 10 秒钟，共采集 10s/2ms=5000 次，即 8253 中断次数为 5000 次。

```
DATA    SEGMENT
BUFFER    DB    8*5000 DUP（？）
DATA    ENDS
；数据采集子程序
8253 初始化编程，通道 0，方式 2，先写低字节，后高字节，BCD 计数，时间常数 4000
        MOV    DX，303H
        MOV    AL，00110101B
        OUT    DX，AL
        MOV    DX，300H          ；通道 0
        MOV    AX，4000H         ；时间常数
        OUT    DX，AL
        MOV    AL，AH
```

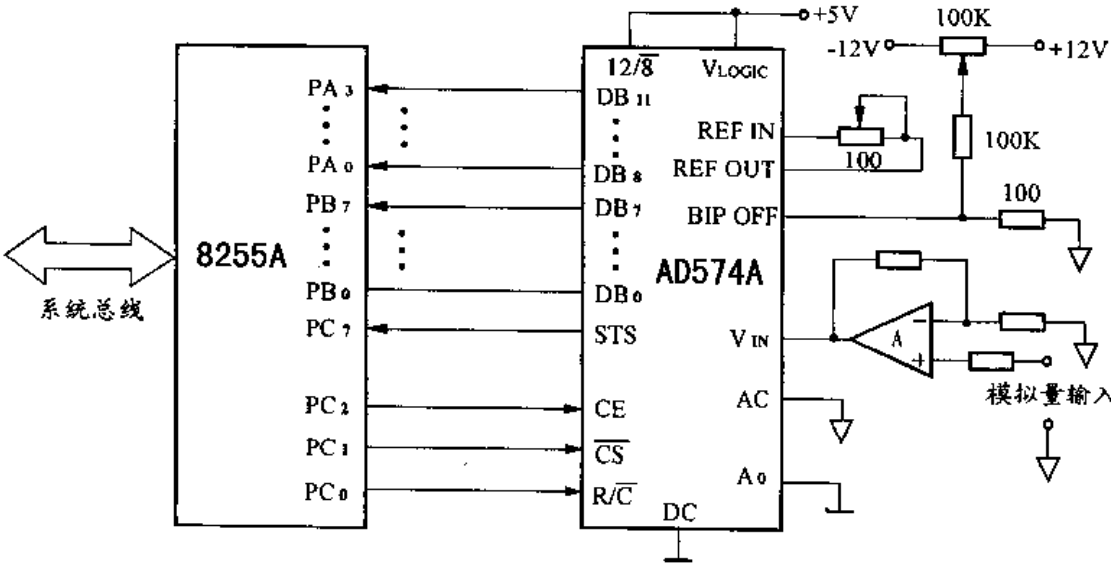
```
        OUT   DX，AL
；8259A 设置屏蔽字，仅允许 8259A 的 IR2 和键盘中断，其余禁止
        MOV   AL，11111001B          ；屏蔽字
        MOV   DX，305H
        OUT   DX，AL                 ；向屏蔽寄存器输出屏蔽字
；设置数据缓冲区始址到 SI 中，计数初值到 BX 中，等待中断，每通道采完 5000 次后结束中断
        MOV   SI，OFFSET   BUFFER    ；SI 指向数据缓冲区
        MOV   BX，5000
        STI                          ；开中断
AGAIN：CMP   BX，0
        JNZ    AGAIN
        MOV   AL，11111101B          ；采集完，禁止 IR2 中断
        MOV   DX，305H
        OUT   DX，AL
        MOV   AH，4CH                ；退出中断
        INT   21H
；中断服务程序，对每个通道均采集一个数据，存入 BUFFER
ADINT   PROC   NEAR
        MOV   CX，0008H              ；设置通道计数器初值
        MOV   DX，308H               ；DX 指向 ADC 通道 0
 NEXT：OUT   DX，AL                  ；启动一次转换
        PUSH   DX                    ；保存通道号
        MOV   DX，306H               ；DX 指向状态口
 POLL：IN   AL，DX                   ；读入 EOC 状态
        TEST   AL，80H               ；EOC（D7）=0？即开始转换了
        JNZ   POLL                   ；非 0，循环等待
 NO_END：IN   AL，DX                 ；EOC=0，开始转换
          TEST   AL，80H             ；再查 EOC 是否为 1
           JZ    NO_END              ；EOC=0，等待转换结束
           POP   DX                  ；EOC=1，恢复通道地址
           IN   AL，DX               ；读取结果
           MOV   [SI]，AL            ；存储到缓冲区
           INC   DX                  ；DX 指向下一个通道
           INC   SI                  ；地址指针指向下一个缓冲单元
           LOOP   NEXT               ；通道计数器减 1，结果非 0 则循环
           DEC   BX                  ；为 0，缓冲数据计数器减 1
           MOV   AL，20H
           MOV   DX，304H
           OUT   DX，AL
           STI                       ；开中断
           IRET
 ADINT     ENDP
```

8. 利用 8255A 和 AD574A 设计数据采集系统，输入模拟电压为 0～+10V，若每秒采集 100 个数据，转换后的数据字存放在 W_BUF 开始的缓冲器中，低字节在前，高字节在后，采满 16K 字节的数据后停止工作，要求：

（1）画出硬件连线图。

（2）编写启动 AD574A 工作和读取转换结果的子程序。

答：（1）



（2）16K 字节的数据共需要 8K=8192 次采集

```
    ；8255A 的端口地址
      PORT_A    EQU   0F0H       ；A 口地址
      PORT_B    EQU   0F1H       ；B 口地址
      PORT_C    EQU   0F2H       ；C 口地址
      PORT_CTL  EQU   0F3H     ；控制口地址
    ；8255A 控制字：A 口和 B 口工作于方式 0，A 口、B 口和 C 口的上半部分为输入，C 口的下半部分为输出
          LEA   DI，W_BUF
          MOV   CX，8192
AD_CONT：MOV   AL，10011010B        ；方式字
          OUT   PORT_CTL，AL        ；输出方式字
        ；启动 A/D 转换
          MOV   AL，00H
          OUT   PORT_C，AL          ；使 CS，CE，R/C 均为低
          NOP                       ；延时
          NOP
          MOV   AL，04H
          OUT   PORT_C，AL          ；使 CE=1，启动 A/D 转换
          NOP
          NOP
          MOV   AL，03H
          OUT   PORT_C，AL          ；使 CE=0，CS=R/C=1，结束启动状态
```

```
READ_STS：IN   AL，PORT_C      ；读 STS 状态
          TEST  AL，80H        ；转换(STS=0)完了吗？
          JNZ   READ_STS       ；否，则循环等待
  ；转换完成，启动读操作
          MOV   AL，01H
          OUT   PORT_C，AL     ；使 CS=0，CE=0， R/C=1
           NOP
          MOV   AL，05H        ；使 CE=1， CS=0， R/C=1
          OUT   PORT_C，AL
  ；读取数据，存入 BX 中
          IN   AL，PORT_A      ；读入高 4 位数据
          AND   AL，0FH
          MOV   BH，AL         ；存入 BH
          IN   AL，PORT_B      ；读入低 8 位
          MOV   BL，AL         ；存入 BL
          MOV   [DI]，BX       ；存入缓冲区
          INC   DI
          INC   DI
；结束读操作
          MOV   AL，03H        ；使 CE=0， CS=1
          OUT   PORT_C，AL     ；结束读操作
          CALL   DELAY_10MS    ；延时 10 个毫秒
          LOOP   AD_CONT
```

第十二章

1. 一般 DMA 控制器应具有哪些基本功能？

答：（1）能向 CPU 提出 DMA 请求，请求信号加到 CPU 的 HOLD 引脚上。

（2）CPU 响应 DMA 请求后，DMA 控制器从 CPU 那儿获得对总线的控制权。在整个 DMA 操作期间，由 DMA 控制器管理系统总线，控制数据传送，CPU 暂停工作。

（3）能提供读/写存储器或 I/O 设备的各种控制命令。

（4）确定数据传输的起始地址和数据的长度，每传送一个数据，能自动修改地址，使地址增 1 或减 1，数据长度减 1.

（5）数据传送完毕，能发出结束 DMA 传送的信号。

2. 什么是 8237DMA 控制器的主态工作方式？什么是从态工作方式？在这两种工作方式下，各控制信号的功能是什么?

答：（1）在 DMA 控制器未取得总线控制权时，必须由 CPU 对 DMA 控制器进行编程，以确定通道的选择、数据传送的方式和类型、内存单元起始地址、地址是递增还是递减以及要传送的总字节数，CPU 也可以读取 DMA 控制器的状态。此时，CPU 处于主控状态，而 DMA 控制器就和一般的 I/O 芯片一样，是系统总线的从属设备，DMA 控制器的这种工作方式称为从态方式。

（2）当 DMA 控制器取得总线控制权以后，系统就完全在它的控制之下，使 I/O 设备和存储器之间或者存储器与存储器之间进行直接的数据传送，DMA 控制器的这种工作方式称为主态方式。

（3）$\overline{\text{CS}}$：从态方式下片选信号

A3～A0：从态时，输入地址信号，寻址 DMA 控制器的内部寄存器，主态时，输出要访问内存的最低 4 位地址。

A7～A4：主态时，输出 4 位地址信息

DB7～DB0：从态时，用于 8237DMA 编程，主态时，输出高 8 位地址和用来传送数据

$\overline{\text{IOR}}$：从态时作为输入控制信号，读取内部寄存器，主态时作为输出控制信号，与 $\overline{\text{MEMW}}$ 配合，控制数据由外设传送到存储器中

$\overline{\text{IOW}}$：从态时输入控制信号，对 8237A 进行初始化编程，主态时，输出控制信号，与 $\overline{\text{MEMR}}$ 配合，把数据从存储器传送到外设

$\overline{\text{MEMR}}$：主态时，与 $\overline{\text{IOW}}$ 配合把数据从存储器读出送外设，也用于控制内存间数据传送，使数据从源地址单元读出。从态时该信号无效。

$\overline{\text{MEMW}}$：主态时，可与 $\overline{\text{IOR}}$ 配合把数据从外设写入存储器，也用于内存间数据传送，控制把数据写入目的单元。从态时该信号无效。

DREQ3～DREQ0：通道 3～0 的 DMA 请求信号。

HRQ：保持请求信号

HLDA：保持响应信号

DACK3～DACK0：通道 3～0 的 DMA 响应信号。

$\overline{EOP}$：传输过程结束信号。

3. 8237A DMA 控制器的当前地址寄存器、当前字节寄存器、基地址寄存器和基字节寄存器各保存什么值？

答：当前地址寄存器用于存放 DMA 传送的存储器地址值，每传送一个数据，地址值自动增 1 或减 1，以指向下一个存储单元。

当前字节寄存器也称为当前字计数寄存器，其初值比实际传送的字节数少 1，该值在编程时由 CPU 写入的，用于保存本次 DMA 传送的字节数。

基地址寄存器：用于存放对应通道当前地址寄存器的初值。

基字节寄存器：也称基字计数寄存器，用于存放对应通道当前字计数器的初值，主要用于自动预置操作时使当前字计数器恢复初值。

4. 8237A 具有几个 DMA 通道？每个通道有哪几种传送方式？各用于什么场合？什么叫自动预置方式？

答：8237A 具有 4 个 DMA 通道，每个通道有 4 种传送方式：

（1）单字节传送方式，此种方式下，每进行一次 DMA 操作，只传送一个字节的数据。此方式能保证在两次 DMA 传送之间，CPU 可执行一次完整的总线操作。

（2）数据块传输方式，此方式可使 DMA 操作连续传输数据，一直到一批数据传送完毕，8237A 才释放总线。

（3）请求传送方式，此方式与数据块传送方式类似，可连续传送数据，但与其不同之处在于，每传送一个字节后，8237A 要对 DREQ 端进行测试，一旦检测到 DREQ 信号无效，则立即停止传送，当 DREQ 有效后，可使 DMA 传输从断点处继续进行。

（4）级联传送方式，此方式可将多个 8237A 连在一起，以便扩充系统的 DMA 通道。

当 DMA 处于自动预置方式时，每当产生有效的 $\overline{EOP}$ 信号后，该通道将自动把基地址寄存器和基字计数器的内容分别重新置入当前地址寄存器和当前字计数器中，达到重新初始化的目的，这样既不需要 CPU 的干预，又能自动执行下一次 DMA 操作。

5. 8237A 可执行哪几条软件命令？

答：8237A 设置了 3 条软件命令，只要对特定的端口地址进行一次写操作，命令就会生效。

（1）清除先/后触发器

8237 内部设有一个先/后触发器，用于控制读/写次序，当触发器清 0 时，读写低 8 位数据，随后先/后触发器自动置成 1，读写高 8 位数据，随后触发器清 0，如此循环。为了按正确的顺序访问寄存器中的高 8 位字节和低 8 位字节，CPU 应使用清除先/后触发器指令，将触发器清 0。

（2）主清命令

也称复位命令，与 RESET 功能相同，可使命令寄存器、状态寄存器、请求寄存器、暂存寄存器和内部先/后触发器均清 0，将屏蔽寄存器置 1.

（3）清除屏蔽寄存器

该命令能清除 4 个通道的全部屏蔽位，允许各通道接收 DMA 请求。

6. 根据图 12.12，表 12.3 和表 12.4，说明如何产生 DMA 传送的 20 位存储器地址。

答：在 DMA 服务期间，直接从 8237A 的 A7～A4 和 A3～A0 输出低 8 位地址，在整个 DMA 传输周期中这些地址信号都是稳定的。

在 S1 和 S2 状态，从数据线 DB7～DB0 输出高 8 位地址 A15～A8，要用三态地址锁存器锁存。

8237A 只能提供 16 位地址，最大寻址 64KB 内存空间，为了实现对全部内存空间的寻

址，在 PC/XT 中设置了一个页面寄存器 74LS670，用来产生存储器的高 4 为地址 A19～A16，8237A 管理低 16 位地址 A15～A0.通过对页面寄存器的编程，既可以在 1M 内存范围内寻址，但在 DMA 传输过程中，页面寄存器的值是固定的，即总是指向内存中某个 64KB 的地址范围。

7. 若 8237A 的端口基地址为 000H，要求通道 0 和通道 1 工作在单字节读传输，地址减 1 变换，无自动预置功能。通道 2 和通道 3 工作在数据块传输方式，地址加 1 变化，有自动预置功能。8237A 的 DACK 为高电平有效，DREQ 为低电平有效，用固定优先级方式启动 8237A 工作，试编写 8237A 的初始化程序。

答：初始化程序如下：

```
DMA    EQU    000H                    ；8237A 的基地址为 00H
；输出主清除命令
OUT    DMA+0DH，AL                     ；发总清命令
；写入方式字：单字节读传输，地址减 1 变化，无自动预置功能，选择通道 0
MOV    AL，01101000B                   ；方式字
OUT    DMA+0BH，AL                     ；写入方式字
；写入方式字：单字节读传输，地址减 1 变化，无自动预置功能，选择通道 1
MOV    AL，01101001B                   ；方式字
OUT    DMA+0BH，AL                     ；写入方式字
；写入方式字：数据块传输方式，地址加 1 变化，有自动预置功能，选择通道 2
MOV    AL，10010010B                   ；方式字
OUT    DMA+0BH，AL                     ；写入方式字
；写入方式字：数据块传输方式，地址加 1 变化，有自动预置功能，选择通道 3
MOV    AL，10010010B                   ；方式字
OUT    DMA+0BH，AL                     ；写入方式字
；写入命令字：DACK 为高电平有效，DREQ 为低电平有效，用固定优先级方式
MOV    AL，11000000 B                  ；命令字
OUT    DMA+08H，AL                     ；写入 8237A
```

（完）