第 1 章　计算机基础知识

教材习题解答

1. 计算机中为什么都采用二进制数而不采用十进制数？

【解】计算机的基本功能是对数的运算和处理。计算机中，通过数字化编码技术，对所表示的数值、文字、符号及控制信息等进行数字编码，这种数字化表示方法不仅要适合于人的自然习惯，同时要满足机器中所用器件、线路的工作状态以及数据可靠传输与易于校验纠错等方面的要求。一个具有两种不同的稳定状态且能相互转换的器件，就可以用来表示一位二进制数，所以表示二进制的器件易于制造且工作可靠，并且二进制数的运算规则也最简单，因此目前计算机中均采用二进制数来表示各种信息及进行信息处理。

2. 写出下列用原码或补码表示的机器数的真值：

(1)01101101　(2)10001101　(3)01011001　(4)11001110

【解】

(1) $[X]_{原}=01101101=+109$　$[X]_{补}=01101101=+109$

(2) $[X]_{原}=10001101=-13$　$[X]_{补}=10001101=-115$

(3) $[X]_{原}=01011001=+89$　$[X]_{补}=01011001=+89$

(4) $[X]_{原}=11001110=-78$　$[X]_{补}=11001110=-50$

3. 填空：

(1) $(1234)_{10}=(\quad)_2=(\quad)_{16}$

(2) $(34.6875)_{10}=(\quad)_2=(\quad)_{16}$

(3) $(271.33)_{10}=(\quad)_2=(\quad)_{16}$

(4) $(1010110010001)_2=(\quad)_{10}=(\quad)_{16}$

(5) $(1AB.E)_{16}=(\quad)_{10}=(\quad)_2$

(6) $(10101010.0111)_2=(\quad)_{10}=(\quad)_{16}$

【解】

(1) $(1234)_{10}=(\ 10011010010\ )_2=(\ 4D2\ )_{16}$

(2) $(34.6875)_{10}=(\ 100010.1011\ )_2=(\ 22.B\ )_{16}$

(3) $(271.33)_{10}=(\ 100001111.010101\ )_2=(\ 10F.54\ )_{16}$

(4) $(1010110010001)_2=(\ 2761\ )_{10}=(\ AC9\ )_{16}$

(5) $(1AB.E)_{16}=(\ 427.875\ )_{10}=(110101011.111\ )_2$

(6) $(10101010.0111)_2=(\ 170.4375\ )_{10}=(\ AA.7\ )_{16}$

4. 已知 X=36，Y=−136，Z=−1250，请写出 X，Y，Z 的 16 位原码、反码和补码。

【解】

$[X]_{原}=0000\ 0000\ 0010\ 0100$

$[Y]_{原}=1000\ 0000\ 1000\ 1000$

$[Z]_{补}$=1000 0100 1110 0010

$[X]_{补}$=0000 0000 0010 0100

$[Y]_{补}$=1111 1111 0111 0111

$[Z]_{反}$=1111 1011 0001 1101

$[X]_{移}$=0000 0000 0010 0100

$[Y]_{移}$=1111 1111 0111 1000

$[Z]_{移}$=1111 1011 0001 1110

5. 已知$[X]_{补}$=01010101B，$[Y]_{补}$=10101010B，$[Z]_{补}$=1000111111111111B，求X、Y、Z及X+Y，Y−Z的十进制值为多少？

【解】Y−Z运算时Y需要扩展符号位

X=85

Y=−86

Z=−28673

X+Y=01010101B+10101010B=11111111B=−1

Y−Z=11111111 10101010B−10001111 11111111B

=11111111 10101010B+01110000 00000001B

=0110 1111 1010 1011B

=28587

6. 用8位补码进行下列运算，并说明运算结果的进位和溢出：

(1) 33+114　(2) 33−114　(3) (-33)+114　(4) (-33)-114

【解】

(1) $[33]_{补}$=00100001

$[114]_{补}$=01110010

00100001

+01110010

10010011

正确的运算结果147>127，出现溢出，使符号位发生变化，出现结果：−109

(2) $[33]_{补}$=00100001

$[-114]_{补}$=10001110

00100001

+10001110

10101111

运算结果：−81

(3) $[-33]_{补}$=11011111

$[114]_{补}$=01110010

11011111

+01110010

101010001

最高位进位，自然丢失，运算结果：+81

(4) $[-33]_{补}=11011111$

$[-114]_{补}=10001110$

```
  11011111
+ 10001110
 101101101
```

正确的运算结果－147<－127，出现溢出，使符号位发生变化，出现运算结果：＋109

7. 将下列十进制数表示为 8421BCD 码：

(1) 8609　(2) 5254　(3) 2730　(4) 2998

【解】

(1) 1000 0110 0000 1001

(2) 0101 0010 0101 0100

(3) 0010 0111 0011 0000

(4) 0010 1001 1001 1000

8. 将下列 8421BCD 码表示为十进制数和二进制数：

(1) 01111001　(2) 001010000101　(3) 011000000111　(4) 010110010000

【解】

(1) 79，　1001111B

(2) 285，　100011101B

(3) 607，　1001011111B

(4) 590，　1001001110B

9. 将下列数值或字符串表示为相应的 ASCII 码：

(1) 51　(2) 7FH　(3) C6H　(4) Computer　(5) how are you?

【解】

(1) 0110101 0110001

(2) 0110111 1100110 1101000

(3) 1100011 0110110 1101000

(4) 1100011 1101111 1101101 1110000 1110101 1110100 1100101 1110010

(5) 1101000 1101111 1110111 0100000 1100001 1110010 1100101 0100000 1111001 1101111 1110101 0111111

10. 定点数和浮点数表示方法各有什么特点？

【解】用浮点表示法比定点表示法表示数的范围大，浮点数的运算比定点数的运算复杂

11. 微处理器、微型计算机和微型计算机系统三者之间有什么不同？

【解】微处理器（CPU），由运算器和控制器组成。运算器完成算术运算和逻辑运算，控制器分析命令并指挥协调各部件统一行动完成命令规定的各种动作或操作。

微型计算机由运算器、控制器、存储器、输入设备、输出设备五大部分组成。

微型计算机系统包括微型计算机硬件和软件。

12. 微型计算机由哪几部分组成，各部分的功能是什么？

【解】微型计算机由运算器、控制器、存储器、输入设备、输出设备五大部分组成。

运算器完成算术运算和逻辑运算；控制器分析命令并指挥协调各部件统一行动完成命令

规定的各种动作或操作；存储器存放原始数据、中间结果和最终结果以及程序；输入设备、输出设备与外界交换信息。

13. CPU 在内部结构上由哪几部分组成，CPU 应具备什么功能？

【解】微处理器（CPU）由运算器和控制器组成。

CPU 应具备的功能：对数据进行处理并对处理过程进行控制。

14. 简述计算机执行指令和执行程序的过程，以书中的例子为例，说明在此二条指令执行中，哪些信号属于数据流，哪些信号属于控制流？

【解】计算机执行指令和执行程序的过程分为：(1) 取指阶段，从存储器中取出指令；(2) 分析执行阶段，由控制器进行分析译码，发出一系列控制信号完成该指令的执行。

以书中的例子为例，在此二条指令执行中，以下信号属于数据流：

① IP→M；

② IP+1→IP；

③ M→IR；

④ IR→ID；addr→M；M→ACC

⑤ ALU 结果→ACC。

以下信号属于控制流：

① 控制上述操作过程的信号流；

② 控制 IP 自动加 1

③ 存储器对地址译码，找到 100 单元；

④ ID 对指令译码后，由控制信号发生器产生一系列控制信号来执行这条指令

⑤ ALU 执行“ADD”运算；

15. 微型计算机外部为什么采用二总线结构？

【解】有了总线结构以后，系统中各功能部件之间的相互关系变为各个部件面向总线的单一关系。一个部件或设备只要符合总线标准，就可以连接到采用这种总线标准的系统中，便系统功能能很简便地得到扩展。

数据总线用来传输数据，地址总线专门用来传送地址信息，控制总线用来传输控制信号。

16. 数据总线和地址总线在结构和作用上有什么不同？

【解】数据总线用来传输数据，地址总线专门用来传送地址信息。从结构上看，数据总线是双向的，即数据既可以从 CPU 送到其它部件，也可以从其它部件传送到 CPU。因地址总是从 CPU 送出去的，所以地址总线是单向的。地址总线的位数决定了 CPU 可以直接寻址的内存范围。

17. 如果某几种 CPU 的地址总线分别有 8，16，20，32 条，它们各自能寻址的存储器的容量是多少？

【解】地址总线 8 条：2^8=256B

地址总线 16 条：2^{16}=64KB

地址总线 20 条：2^{20}=1MB

地址总线 32 条：2^{32}=4GB

18. 什么是硬件，什么是软件，硬件和软件的关系如何？

【解】硬件是指组成计算机的各种电子的、机械的、光磁学的物理器件和设备，它们构成了

计算机的物理实体。软件则是指为了运行，管理和维护计算机而编制的各种程序及其有关的文档资料的总称。硬件是基础，软件是灵魂，两者既相互独立，又相互依存，缺一不可。硬件和软件合起来才组成一个完整的计算机系统。

19. 说明位、字节、字长的概念及它们之间的关系。

【解】(1) 位（bit）。位是计算机所能表示的最基本最小的数据单位。它只能有两种状态"0"和"1"，即二进制位。

(2) 字（Word）。计算机中作为一个整体参与运算、处理和传送的一串二进制数，是计算机中信息的基本单位。

(3) 字长(Word Length)。计算机中每个字所包含的二进制位数称为字长。

它们之间的关系：字由位构成，字长指每个字所包含的位的个数。

20. 计算机的发展趋势有哪些？你如何看待冯·诺依曼计算机体系结构理论？

【解】计算机的发展趋势包括：微处理器的位数增加（4 位→64 位）；采用并行处理技术；集中式主机模式逐渐被客户/服务器模式所取代；网络技术的应用和普及；多媒体技术技术应用等。

冯·诺依曼计算机体系结构理论的核心是"存储程序"和"程序控制"。冯·诺依曼提出的这些基本概念奠定了现代计算机体系结构的基本框架，并由此产生了程序设计思想。尽管从计算机诞生到现在已经历了半个多世纪，计算机的体系结构已发生了很大变化，计算机的性能也有了巨大提高，但目前大多数计算机仍遵从冯·诺依曼体系结构理论。

21. 说出几种型号的 CPU，它们各有什么特点？

【解】

(1) 8088，8086

Intel 公司于 1981 年推出，16 位微处理器，地址线有 20 条，内存寻址范围为 1M 字节。它们的区别在于，8086 外部的数据也是 16 位，而 8088 的外部数据为 8 位。

(2) 80286

80286 也是 16 位处理器，其频率比 8086 更高，它有 24 条地址线，内存寻址范围是 16M 字节。

(3) 80386

80386 属于 32 位微处理器，其内部和外部数据总线都是 32 位，地址总线也是 32 位，可寻址 4GB 内存。它除具有实模式和保护模式外，还增加了虚拟 86 的工作方式，可以通过同时模拟多个 8086 处理器来提供多任务能力。386 处理器的主频有 16，20，25，33，40MHz 五种。

(4) 80486

于 1989 年由 Intel 公司首先出。其时钟频率从 25MHz 逐步提高到 33MHz，50MHz。它也属于 32 位处理器。80486 是将 80386 和数学协处理器 80387 以及一个 8KB 的高速缓存集成在一个芯片内，并且在 80X86 系列中首次采用了 RISC 技术，可以在一个时钟周期内执行一条指令。它还采用了突发总线方式，大大提高了 CPU 与内存的数据交换速度。

(5) Pentium 处理器

Pentium（奔腾）是 Intel 公司于 1993 年推出的新一代微处理器，Pentium 微处理器使用更高的时钟频率，最初为 60MHZ 和 66MHZ，后提高到 200MHZ，64 位数据总线，16KB 的

高速缓存。接着 Intel 推出使用 MMX 技术的 Pentium MMX 的多能奔腾。它增加了 57 条多媒体指令，内部高速缓存增加到 32KB。最高频率是 233MHz。MMX 是 Multimedia Extension 的缩写，意即多媒体扩展。一种基于多媒体计算以及通讯功能的技术，它能生成高质量的图像、视频和音频，加速对声音图像的处理。

(6) Pentium II

PentiumII 与以往的 Pentium 处理器使用了不同的封装方式，它将处理器放到了盒中。而且采用 SLOT 1 模式的插座。该形式的封装结构为系统总线与 L2 高级缓存之间的接口提供了独立的连接电路。然后再将处理器、高速缓存芯片、都放置在一个小型电路板上（SEC 卡盒）。

(7) 赛扬

赛扬属于 Pentium II 的低价位版本，被称为"Celeron"。它是将 Pentium II 处理器的二级 Cache 去掉，并简化了封装形式，没有塑料壳，另加一块散热片组成。因为没有了 Cache，其速度明显下降。

(8) 赛扬 300 A

Celeron 300A 处理器是包含了 128K 二级缓存的 Pentium II 处理器，其缓存是集成在 CPU 内部的，速度和 CPU 相同，比 Pentium II/III 的 Cache 速度还要高。这样 CPU 从二级缓存中读写数据时不需等待，可以大大提高计算速度。赛扬 300 A 仍没有塑料外壳，采用了 SLOT1 的结构，加了一个散热片和一块风扇。

(9) Pentium III

它采用了与 Pentium II 相同的 SLOT1 结构，具有 100MHz 的外频。其内部集成了 64K 的一级缓存，512K 的二级缓存仍然安装在 SLOT1 的卡盒内，工作频率是 CPU 的一半。提供了比 PentiumII 更强劲的性能。这主要表现在其新增加了 KNI 指令集。KNI 指令集中提供了 70 条个新的指令，可以大大提高 3D 运算、动画片、影像、音效等功能，增强了视频处理和语音识别的功能。

22. 说出目前流行的几种主机板的类型以及他们的性能特点。

【解】ATX 是目前市场上最常见的主板结构，扩展插槽较多，PCI 插槽数量在 4-6 个，大多数主板都采用此结构；

Micro ATX 又称 Mini ATX，是 ATX 结构的简化版，扩展插槽较少，PCI 插槽数量在 3 个或 3 个以下，多用于品牌机并配备小型机箱；

BTX 是英特尔制定的最新一代主板结构。

AT 主板的尺寸为 13"×12"，板上集成有控制芯片和 8 个 I/0 扩充插槽。由于 AT 主板尺寸较大，因此系统单元（机箱）水平方向增加了 2 英寸，高度增加了 1 英寸，这一改变也是为了支持新的较大尺寸的 AT 格式适配卡。AT 主板尺寸较大，板上能放置较多的元件和扩充插槽。

Baby/Mini AT 主板：随着电子元件集成化程度的提高，相同功能的主板不再需要个 AT 的尺寸。因此在 1990 年推出了规范，简称为 Baby AT 主板。Baby AT 主板是从最早的 XT 主板继承来的，它的大小为 15"×8.5"，比 AT 主板是略长，而宽度大大窄于 AT 主板。Baby AT 主板沿袭了 AT 主板的 I/0 扩展插槽，键盘插座等外设接口及元件的摆放位置，而对内存槽等内部元件结构进行了紧缩，再加上大规模集成电路使内部元件减少，使得 Baby AT 主板比 AT 主板布局紧凑而功能不减。

Micro ATX 主板把扩展插槽减少为 3-4 只，DIMM 插槽为 2-3 个，从横向减小了主板宽度，共总面积减小约 0.92 平方英寸，比 ATX 标准主板结构更为紧凑。按照 Micro ATX 标准，板上还应该集成图形和音频处理功能。目前很多品牌机主板使用了 Micro ATX 标准。

BTX 是英特尔提出的新型主板架构 Balanced Technology Extended 的简称，是 ATX 结构的替代者。BTX 具有如下特点：支持 Low-profile，也即窄板设计，系统结构将更加紧凑；针对散热和气流的运动，对主板的线路布局进行了优化设计；主板的安装将更加简便，机械性能也将经过最优化设计。

23　常用的外部设备有哪些，它们各有什么特点？如何衡量它们的性能？

【解】常用的外部设备有外存储器（如 CD-ROM、磁带存储器、硬盘、软盘）、键盘、鼠标、打印机、显示器、多媒体设备（如扫描仪、扬声器）、网络设备等。

CD-ROM 的特点是：存储容量大，只能读不能写，读写数据速度低于硬盘。

磁带存储器的特点是：存储容量大，顺序存取方式，主要用于在系统中备份数据。

硬盘的特点是：存储容量大，读写速度高

软盘的特点是：便宜，适宜于由用户保存数据，容易损坏，容量小，速度慢。

键盘：键盘是最主要的输入设备。

鼠标：鼠标器是控制显示屏上光标移动位置从而主机输入用户所选中的某个操作命令或操作对象的一种常用的输入设备。

打印机：打印机是产生硬拷贝输出的一种设备，供用户保存计算机处理的结果。

显示器：是用户与计算机对话的主要窗口，分辨率，彩色数目及屏幕尺寸是显示器的主要指标。

扫描仪：是一种输入图片和文字的外部设备。

24　计算机软件包括哪些种类，它们有什么不同？

【解】计算机软件包括系统软件和应用软件两大类。

系统软件指由机器的设计者提供的，使用和管理计算机的软件。系统软件包括：①各种语言的汇编或解释、编译程序。②机器的监控管理程序，操作系统、调试程序、故障诊断程序。③程序库。

应用软件指用户用各种语言编制的解决各种问题的软件，如财务管理软件、银行管理软件、文字处理软件等。

系统软件和应用软件的不同：系统软件指由机器的设计者提供的，其目的是让用户更方便地使用和管理计算机，而不必了解具体的计算机硬件，从而使用户编制各种源程序更为简单、方便和可靠。应用软件则是为了解决各种应用问题的软件，其目的为了扩大计算机的功能和应用领域，方便各应用领域的用户的使用。

25.　你知道或用过哪些系统软件，它们各有什么功能特点？

【解】

(1) DOS 系统是 1981 年由微软公司为 IBM 个人电脑开发的，它是一个单用户单任务的操作系统。在 1985 年到 1995 年间 DOS 占据操作系统的统治地位。

功能特点：小巧灵活，文件管理方便，外设支持良好，应用程序众多

(2) Windows

Windows 是一个为个人电脑和服务器用户设计的操作系统。它的第一个版本由微软公司

发行于 1985 年，并最终获得了世界个人电脑操作系统软件的垄断地位。

功能特点：具有友好的图形用户界面；具有强大的内存管理功能（可直接管理 4GB 内存），允许多任务操作（可同时运行多个程序），且速度较快；具有出色的多媒体功能；支持新硬件，如 DVD、数字相机等；可靠性更强。

(3) Unix

Unix 是一种分时计算机操作系统，1969 在 AT&TBell 实验室诞生。从此以后其优越性不可阻挡的占领网络。大部分重要网络环节都是 Unix 构造。

功能特点：具有强大的可移植性，适合多种硬件平台；可操作性很强；具有良好的用户界面和程序接口；为用户提供了数千条系统命令，有助于系统操作和系统管理；管道机制；为用户提供了良好的开发环境；具跨平台的文件系统和网络文件系统；具有强大的网络功能；完善的系统审计；增强的系统安全机制；系统备份功能完善；系统结构清晰，有利于操作系统的教学和实践；具有强稳定性和健壮的系统核心。

(4) Linux

Linux 是 Unix 克隆的操作系统，在源代码上兼容绝大部分 Unix 标准，是一个支持多用户、多进程、多线程、实时性较好的且稳定的操作系统。

功能特点：完全免费；完全兼容 POSIX 1.0 标准；多用户、多任务；良好的界面；丰富的网络功能；多进程、多线程、实时性较好；支持多种平台。

26. 说出你用过的一种计算机的主要性能指标。

【解】(1) 联想天骄 e1050x 的主要性能指标：

处理器类型 Celeron D 331，处理器主频 2660MHz，处理器的一级缓存 128KB，二级缓存 256KB，处理器运算位数 64 位

主板的系统总线频率 533MHz

内存类型 DDR，容量 256 MB

外围设备包括： DVD-ROM (16X)光驱，硬盘(7200rpm，80GB)，17 英寸液晶显示器，显卡，声卡，音响系统，10/100M 网卡，键盘，USB 光电鼠标

(2) 联想家悦 C 1066E 的主要性能指标：

处理器类型 SEM 2200+，处理器主频 2200MHz，处理器的一级缓存 128KB，二级缓存 256KB，处理器运算位数 64 位

主板的系统总线频率 533MHz

内存类型 DDR，容量 256 MB

外围设备包括： DVD-ROM (16X)光驱，硬盘(7200rpm，80GB)，17 英寸液晶显示器，显卡，声卡，音响系统，10/100M 网卡，键盘，USB 光电鼠标

第 2 章 8086 微处理器及其系统

教材习题解答

1. 8086 CPU 由哪两部分构成，它们的主要功能是什么?在执行指令期间,EU 能直接访问存储器吗，为什么?

【解】8086CPU 由执行部件（EU）和总线接口部件（BIU）两部分组成。

执行部件由内部寄存器组、算术逻辑运算单元（ALU）与标志寄存器（FR）及内部控制逻辑等二部分组成。寄存器用于存储操作数和中间结果；算术逻辑单元完成 16 位或 8 位算术逻辑运算，运算结果送上 ALU 内部数据总线。同时在标志寄存器中建立相应的标志；内部控制逻辑电路的主要功能是从指令队列缓冲器中取出指令，对指令进行译码，并产生各种控制信号，控制各部件的协同工作以完成指令的执行过程。

总线接口部件（BIU）负责 CPU 与存储器、I/O 设备之间传送数据，地址、状态及控制信息。

每当 EU 部件要执行一条指令时，它就从指令队列头部取出指令，后续指令自动向前推进。EU 要花几个时钟周期执行指令，指令执行中若需要访问内存或 I/O 设备，EU 就向 BIU 申请总线周期，若 BIU 总线空闲，则立即响应，若 BIU 正在取一条指令，则待取指令操作完成后再响应 EU 的总线请求。

2. 8086CPU 与传统的计算机相比在执行指令方面有什么不同？这样的设计思想有什么优点？

【解】8086 CPU 与传统的计算机相比增加了指令队列缓冲器，从而实现了执行部件（EU）与总线接口(BIU)部件的并行工作，因而提高了 8086 系统的效率。

3. 8086 CPU 中有哪些寄存器，各有什么用途？

【解】8086 共有 8 个 16 位的内部寄存器，分为两组：

① 通用数据寄存器。四个通用数据寄存器 AX、BX、CX、DX 均可用作 16 位寄存器也可用作 8 位寄存器。用作 8 位寄存器时分别记为 AH、AL、BH、BL、CH、CL、DH、DL。

AX（AH、AL）累加器。有些指令约定以 AX（或 AL）为源或目的寄存器。实际上大多数情况下，8086 的所有通用寄存器均可充当累加器。

BX（BH、BL）基址寄存器。BX 可用作间接寻址的地址寄存器和基地址寄存器。BH、BL 可用作 8 位通用数据寄存器。

CX（CH、CL）计数寄存器。CX 在循环和串操作中充当计数器，指令执行后 CX 内容自动修改，因此称为计数寄存器。

DX（DH、DL）数据寄存器。除用作通用寄存器外，在 I/O 指令中可用作端口地址寄存器，乘除指令中用作输助累加器。

② 指针和变址寄存器。

BP（Basic Pointer Register）基址指针寄存器。

SP（Stack Pointer Register）堆栈指针寄存器。

SI（Source Index Register）源变址寄存器。

DI（Destination Index Register）目的变址寄存器。

BP，SP 称为指针寄存器，用来指示相对于段起始地址的偏移量。BP 和 SP 一般用于堆栈段。SI、DI 称为变址寄存器，可用作间接寻址，变址寻址和基址变址寻址的寄存器。SI 一般用于数据段，DI 一般用于数据段或附加段。

标志寄存器（FR）：是一个 16 位寄存器，算术逻辑单元进行算术逻辑运算后，在标志寄存器中建立相应的标志。

段地址寄存器（CS、DS、SS、ES）：用于存放段地址，根据其主要用途，分为代码段寄存器 CS，数据段寄存器 DS，堆栈段寄存器 SS，附加段寄存器 ES。

代码段寄存器 CS：代码段是存放程序代码的存储区域，代码段寄存器用来存放代码段存储区域的起始地址。

数据段寄存器 DS：数据段是存放程序中所使用的数据的存储区域，数据段寄存器用来存放程序的数据存储区的起始地址。

堆栈段寄存器 SS：堆栈段寄存器用来存放堆栈存储区的起始地址。由堆栈段寄存器 SS 与堆栈指针寄存器 SP 来确定当前堆栈指令的操作地址。

附加段寄存器 ES：附加段是为某些字符串操作指令存放目的操作数而设置的一个附加的数据段，附加段寄存器用来存放该附加数据段存储区域的起始地址。

指令指针寄存器（IP），又称程序计数器，是 16 位寄存器。IP 中存放当前将要执行的指令的有效地址，每取出一条指令 IP 自动增量，即指向下一条指令。

指令队列缓冲器：是一个与 CPU 速度相匹配的高速缓冲寄存器。在 EU 执行指令的同时，BIU 可以从内存中取出下一条或下几条指令放到指令缓冲器中，EU 执行完一条指令后，可以立即从指令缓冲器中执行下一条指令。

4. 状态标志与控制标志有何不同，程序中是怎样利用这两类标志的？标志寄存器有哪些标志位，各在什么情况下置位？

【解】状态标志根据算术逻辑运算结果由硬件自动设定，它们反映运算结果的某些特征或状态，可作为后继操作（如条件转移）的判断依据。控制标志由用户通过指令来设定。它们可控制机器或程序的某些运行过程。

标志寄存器的内容如下：

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				OF	DF	IF	TF	SF	ZF	--	AF	--	PF	--	CF

CF（Carry Flag）进位标志，反映在运算结果的最高位有无进位或借位。如果运算结果的最高位产生了进位（加法）或借位（减法）则 CF=1，否则 CF=0。

PF（Parity Flag）奇偶标志，反映运算结果中“1”的个数的奇偶性，主要用于判断数据传送过程中是否出错。若结果的低 8 位中有偶数个“1”则 PF=1，否则 PF=0。

AF（Auxiliary Carry Flag）辅助进位标志，又称半进位标志。加减运算时，若 D_3 向 D_4 产生了进位或借位则 AF=1，否则 AF=0。在 BCD 码运算时，该标志用于十进制调整。

ZF（Zero Flag）零标志，反映运算结果是否为 0。若结果为零则 ZF=1，否则 ZF=0。

SF（Sign Flag）符号标志，反映运算结果最高位即符号位的状态。如果运算结果的最高

位为 1 则 SF=1（对带符号数即为负数），否则 SF=0（对带符号数即为正数）。

OF（Overflow Flag）溢出标志，反映运算结果是否超出了带符号数的表数范围。若超出了机器的表数的范围，即为产生溢出，则 OF=1，否则 OF=0。

DF（Direction Flag）方向标志，用于串处理指令中控制串处理的方向。当 DF=1 时，每次操作后变址寄存器 SI、DI 自动减量，因此处理方向是由高地址向低地址方向进行。当 DF=0 则 SI、DI 自动增量，处理方向由低地址向高地址方向进行。该标志由方向控制指令 STD 或 CLD 设置或清除。

IF（Interrupt Flag）中断允许标志，用于控制 CPU 是否允许响应可屏蔽中断请求。IF=1 为允许响应可屏蔽中断请求，IF=0 则禁止响应可屏蔽中断请求。该标志可由中断控制指令 STI 或 CLI 设置或清除。

TF（Trap Flag）陷阱标志，用于单步操作。TF=1 时，每执行一条用户程序指令后自动产生陷阱，进入系统的单步中断处理程序。TF=0 时，用户程序会连续不断地执行，不会产生单步中断。

5. 求出下列运算后各个标志的状态，并说明进位标志和溢出标志的区别。

(1) 1278H + 3469H　　(2) 54E3H - 27A0H

(3) 3881H + 3597H　　(4) 01E3H - 01E3H

【解】CF 进位标志，反映在运算结果的最高位有无进位或借位。OF 溢出标志，反映运算结果是否超出了带符号数的表数范围。机器实际处理时判断是否溢出的方法是根据最高位的进位（CF）与次高位的进位是否相同来确定，若两者不相同则 OF=1（表示有溢出），否则 OF=0（表示无溢出）。

(1)

0001 0010 0111 1000
+ 0011 0100 0110 1001
0100 0110 1110 0001

运算后各个标志的状态：

OF	DF	IF	TF	SF	ZF	--	AF	--	PF	--	CF
0				0	0		1		1		0

(2)

0101 0100 1110 0011
− 0010 0111 1010 0000
0010 1100 0100 0011

运算后各个标志的状态：

OF	DF	IF	TF	SF	ZF	--	AF	--	PF	--	CF
0				0	0		0		0		0

(3)

0011 1000 1000 0001
+ 0011 0101 1001 0111
0110 1110 0001 1000

运算后各个标志的状态：

OF	DF	IF	TF	SF	ZF	--	AF	--	PF	--	CF
0				0	0		0		1		0

(4)

```
  0000 0001 1110 0011
- 0000 0001 1110 0011
  0000 0000 0000 0000
```

运算后各个标志的状态：

OF	DF	IF	TF	SF	ZF	--	AF	--	PF	--	CF
0				0	1		0		1		0

6. 8086 CPU 中存储器的逻辑地址和物理地址之间有什么关系，各有多少位？

【解】物理地址为某一个存储单元的实际地址，对于 8086 它是一个 20 位的地址。物理地址从 00000H～FFFFFH 变化，对应 1MB 的空间。

逻辑地址，又称偏移地址或有效地址，即对段首的偏移量。偏移地址从 0000H～FFFFH 变化，对应 64KB 的空间。

物理地址的获得方法是：将段寄存器的内容左移 4 位（即×16），与逻辑地址相加，得到 20 位物理地址。根据寻址方式的不同，偏移地址可以来自指令计数器（IP）或其它寄存器。

7. 8086CPU 使用的存储器为什么要分段，怎样分段？为什么要设置段寄存器，有几个段寄存器？各段寄存器有什么意义？

【解】8086 CPU 内部数据结构是 16 位的，即所有的寄存器都是 16 位的，而外部寻址空间为 1MB，即需要 20 位地址线。为了能用内部寄存器中的 16 位地址来寻址 1MB 空间，8086 将 1MB 空间以 16 字节为一节内存节，共分成 64K 个节。节的起始地址分别为 00000H、00010H、00020H、…、FFFF0H，称为段采址。节的起始地址的后 4 位二进制数为全 0，称为节的段地址。

用于存放段地址的寄存器称为段寄存器，根据其主要用途，分为代码段寄存器 CS、数据段寄存器 DS、堆栈段寄存器 SS、附加段寄存器 ES。

代码段寄存器 CS：用来存放代码段存储区域的起始地址。

数据段寄存器 DS：数存放程序的数据存储区的起始地址。

堆栈段寄存器 SS：用来存放堆栈存储区的起始地址。由堆栈段寄存器 SS 与堆栈指针寄存器 SP 来确定当前堆栈指令的操作地址。

附加段寄存器 ES：附加段是为某些字符串操作指令存放目的操作数而设置的一个附加的数据段，附加段寄存器用来存放该附加数据段存储区域的起始地址。

8. 简述 A_0 与 $\overline{BHE}$ 在 8086 系统中的应用。

【解】8086 系统中将 1MB 存储空间分成两个 512KB 的物理存储体。一个存储体由偶数地址组成，另一个存储体由奇数地址组成。用 A0 位来区分两个存储体。

用 $\overline{BHE}$ 和 A_0 的组合来选择存储体，其组合关系及操作情况如下：

(1) 从偶地址读写一个字节（$\overline{BHE}$ A_0=10）。AD_{15}～AD_8 上的数据被忽略，字节内容通过 AD_7～AD_0 传送。

(2) 从奇地址读写一个字节（$\overline{BHE}$=0，A_0=1）。在 AD_{15}～AD_8 上传送的数据有效，AD_7～AD_0 上数据被忽略。

(3) 从偶地址开始读写一个字（$\overline{BHE}$=0，A_0=0）。在 AD_{15}～AD_0 上传送的数据同时有效。

(4) 从奇地址开始读写一个字。第一个总线周期 $\overline{BHE}$=0，A_0=1，从奇地址读写低字节，在 AD_{15}～AD_8 上传送的数据有效。第二个总线周期 $\overline{BHE}$=0，A_0=1，从偶地址读写高字节，在 AD_7～AD_0 上传送的数据有效。

9. 8086 系统中为什么要采用地址锁存器 8282？采用什么方法从分时复用地址/数据线中将数据和地址信号分离出来？

【解】8086 地址总线与数据总线是分时复用的，高 8 位数据有效信号 $\overline{BHE}$ 也是复用信号。在 T_1 状态，总线上输出 20 位地址信号及 $\overline{BHE}$ 信号，而在 T_2～T_4 状态，总线用于数据传送，$\overline{BHE}$ 信号也失效。为了正确地交换数据，地址信号及 $\overline{BHE}$ 信号在 T_2～T_4 期间必须保持，所以需要设一组地址锁存器（3 片 8282），用于锁存地址及 $\overline{BHE}$ 信号。

在 T_1 状态，CPU 送出地址锁存允许信号 ALE，将 ALE 接向 8282 的选通输入端 STB。当 ALE=1 时，8282 输出跟随输入变化，用 ALE 的下降沿将总线上已经稳定的地址信号锁入 8282。

10. 8086 和 8088 CPU 的主要区别是什么？

【解】8088 的内部结构和指令功能与 8086 完全相同，只是为了和原有的 8 位微处理器外围芯片兼容，其外部数据总线是 8 位的。

11. 8086 系统中的存储器采用什么结构？如何与地址、数据线连接？

【解】8086 系统中将 1MB 存储空间分成两个 512KB 的物理存储体。一个存储体由偶数地址组成，另一个存储体由奇数地址组成。用 A0 位来区分两个存储体。

12. 8086 的 I/O 端口寻址范围是多少？什么是 I/O 端口与内存分别独立编址？

【解】8086 的 I/O 端口使用 16 位地址 A15～A0，I/O 端口地址范围为 0000H～FFFFH，可寻址空间为 64KB。

I/O 端口与内存分别独立编址时，指令访问的是 I/O 端口还是内存，由地址信息无法区分，由 M/$\overline{IO}$ 信号区分是 I/O 端口的寻址与内存寻址。

13. 在对存储器和 I/O 设备读写时，要用到 $\overline{IOR}$（IO 读），$\overline{IOW}$（IO 写），$\overline{MR}$（存储器读），$\overline{MW}$（存储器写）信号。这些信号的作用是什么？它们在最小模式时可用怎样的电路得到？请画出示意图。

【解】$\overline{IOR}$：该信号有效时，对 IO 端口执行读操作

$\overline{IOW}$：该信号有效时，对 IO 端口执行写操作

$\overline{MR}$：该信号有效时，对存储器执行读操作

$\overline{MW}$：该信号有效时，对存储器执行写操作

在最小模式时可分别用以下电路得到上述信号：

14. 什么是基地址和位移量，它们之间有何联系？

【解】8086 CPU 内部数据结构是 16 位的，而外部寻址空间为 1MB。为了使用内部寄存器中的 16 位地址来寻址 1MB 空间，8086 将 1MB 空间以 16 字节为一个内存节（Paragraph），共分成 64K 个节。节的起始地址称为段基址。偏移地址是存储地址对段首的偏移量。偏移地址从 0000H～FFFFH 变化，对应 64KB 的空间。

它们之间有何联系：物理地址＝基地址×16＋位移量。

15. 设 CS=1200H，IP=0FF00H，此时指令的物理地址是多少？指向这一物理地址的 CS 和 IP 的值是惟一的吗？

【解】指令的物理地址：12000H＋0FF00H＝21F00H

指向这一物理地址的 CS 和 IP 的值不是惟一的。

16. 若 CS=1000H，指出当前代码段可寻址的存储空间的大小和地址范围。

【解】当前代码段可寻址的存储空间的大小：64KB

当前代码段可寻址的存储空间的地址范围：10000H～1FFFFH

17. 简述 8086 单 CPU 和多 CPU 系统各自主要特点，并说明有何差别。

【解】单 CPU 系统中只有一个微处理器 8086，所有总线控制信号由它产生，系统中总线控制逻辑信号可减少到最小。

多 CPU 系统中包括两个以上处理器，其中一个为 8086 作为主处理器，其它处理器作为协处理器，一般多用于复杂的大中型系统。与 8086 协同工作的协处理器有 8087、8089 两种，分别为数学协处理器和输入/输出协处理器。配置协处理器的系统，主处理器不用处理费时的复杂运算和 I/O 操作，因此可大大提高主处理器的运行效率。

18. 时钟周期、T 状态、总线周期、指令周期的定义是什么，什么情况下会出现空闲周期？

【解】计算机是由一串脉冲控制进行工作的，这一串脉冲称为计算机的时钟，每个脉冲的时间称为一个时钟周期，每个脉冲称为一个时钟脉冲或一个 T 状态。若干个时钟脉冲完成一个基本操作。一种基本操作称为一个总线周期。执行一条指令所需要的时间称为指令周期

19. 8086 CPU 读/写总线周期包含几个时钟周期，什么情况下需要插入 T_W 等待周期，插入 T_W 的数量取决于什么因素

【解】8086 CPU 读/写总线周期包含 4 个时钟周期。

读总线周期：在 T_1 状态内存或 I/O 端口将数据送上数据总线。CPU 准备读入数据。在 T_3 的前沿（下降沿），CPU 查询 READY 引脚，若内存或外设工作速度较慢，来不及在基本总线周期内完成数据传送工作，则应通过逻辑电路在 T_2 前沿之前产生 READY 低电平信号，T_3 前沿若查到 READY 为低电平，则在 T_3 后自动插入一个等待状态 T_W，在 T_W 前沿继续查询 READY 信号，若 READY 仍为低电平，则继续插入 T_W，直到 READY 上升为高电平，则等待状态结束，进入 T_4 状态。

写总线周期：在 T_2 状态中，T_2 状态有效的信号将继续保持有效。将被向外部写数据。在 T_3 的下降沿查询 READY，若内存或 I/O 端口在标准总线周期内来不及接收数据，则应通过逻辑电路在 T_2 前沿之前产生 READY 低电平信号。CPU 查到 READY 为低，则在 T_3 之后插入一个 T_W，并在 T_W 前沿继续查询 READY，直到 READY 上升为高电平，则结束等待进入 T_4 状态。

20. 8086 CPU 复位后，有哪些特征？8086 系统的起动程序如何去找？

【解】8086 CPU 复位后：所有内部寄存器、标志寄存器 IR 及 ES、SS、DS 段寄存器清 0，指令队列缓冲器清空，指令指针寄存器（IP）清 0，CS 被置为 FFFFH；复位时，所有三态输出总线变为高阻状态，这些三态总线包括：AD_{15}~AD_0，A_{19}/S_6~A_{16}/S_3，$\overline{BHE}/S_7$，$\overline{S_2}$（$M/\overline{IO}$），$\overline{S_1}$（$DT/\overline{R}$），$\overline{S_0}$（$\overline{DEN}$），$\overline{DEN}$，$\overline{WR}$，$\overline{RD}$，$\overline{INTA}$ 等。ALE，HLDA，QS_0，QS_1 等信号降为低电平，$\overline{RQ}/\overline{GT_0}$，$\overline{RQ}/\overline{GT_1}$ 等信号上升为高电平。

8086 系统的起动程序从 CS×16+IP 即 FFFF0H 地址开始执行。

21.8086 系统在最小模式时应该怎样配置？试画出这种配置并标出主要信号的连接关系。

【解】8086 系统在最小模式时的典型配置：一片 8284A 时钟发生器产生系统所需要的时钟信号 CLK，同时对外部 READY 信号和系统复位信号 RESET 进行同步，共输出送向 8086 应相引脚。三片 8282（或 74LS373）地址锁存器用于 20 位地址和 $\overline{BHE}$ 信号锁存，使得整个总线读写周期期间地址信号始终有效，以支持 8086CPU 地址/数据总线分时复用的工作方式。两片 8286 总线驱动器(又称总线收发器或总线驱动器)。当系统所连存储器和外设较多时，为了提高数据总线的驱动能力，可以接入 8286 芯片。

该模式的配置图见教材图 2-8。

22. 画出最小模式时读存储器或 I/O 设备的总线周期时序。

【解】8086 最小模式下的读周期时序见教材图 2-15。

第 3 章 从 8086 到 Pentium 系列微处理器的技术发展

教材习题解答

1. 简述 80286 的特点和保护模式的保护功能。

【解】80286 的特点：

① CPU 内部分为四个处理部件：EU（执行部件），AU（地址部件），IU（指令部件）和 BU（总线部件）。这四个处理部件可以并行的进行操作，提高了处理速度。

② 数据线和地址线完全分离。在一个总线周期中，当有效数据出现在数据总线上的时候，下一个总线周期的地址已经送到地址总线，形成总线周期的流水作业。

③ 具有“实地址模式”（Real Address Mode，简称为“实模式”）和“保护虚地址模式”（Protected Virtual Address Mode，简称为“保护模式”）两种工作模式。

④ 能运行实时多任务操作系统，支持存储管理和保护功能。

⑤ 实现了虚拟存储管理。

⑥ 与 80286 配合使用的数学协处理器是 80287，它基本与 8087 相同，但适应 80286 的两种工作模式。

保护模式体现了 80286 的特色，主要是对存储器管理、虚拟存储和对地址空间的保护。在保护模式下，可为每个任务提供多达 1GB 的虚拟存储空间和保护机制，有力地支持了多用户、多任务的操作。那些内存装不下的逻辑段，将以文件形式存在外存储器中，当处理器需要对它们进行存取操作时就会产生中断，通过中断服务程序把有关的程序或数据从外存储器调入到内存，从而满足程序运行的需要。

保护模式为不同程序设置了四个特权级别，可让不同程序在不同的特权级别上运行。依靠这一机制，可支持系统程序和用户程序的分离，并可进一步分离不同级别的系统程序，大大提高了系统运行的可靠性。

2. 简述 80386 的特点、80386 引脚与 8086 的区别。

【解】80386 的特点：

80386 是个 32 位结构，它的外部数据总线和内部数据通道，包括寄存器、ALU 和内部总线都是 32 位的。

80386 有 3 种工作模式：实模式、虚拟 86 模式、386 的保护模式。

80386 的硬件结构可分成 6 个逻辑单元，它们以流水线方式工作，运行速度可达 4MIPS。其硬件设计有支持段页式存储管理部件，易于实现虚拟存储系统。在保护模式下的分段寻址体系，与操作系统相配合可以组成虚拟存储器系统，一个任务的最大虚拟空间可达 2^{46}=64 TB。

80386 硬件支持多任务处理，用一条指令就可以实现任务切换。

80386 设置了 4 级特权级，按优先顺序依次为 0 级、1 级、2 级、3 级，前 3 级用于操作系统程序，后 1 级用于用户程序。

80386 引脚与 8086 的区别见表 3-1。

表 3-1　80386 引脚与 8086 的区别

8086CPU	80386CPU
共有 40 个引脚	共有 132 个引脚
16 条地址/数据复用线 4 条地址线	24 条地址线 32 条数据线
	有总线宽度控制信号 BS#，由控制下，可选择 16 位或 32 位数据传送。
	字节控制信号 $\overline{BE_0}$～$\overline{BE_3}$
	协处理器接口信号 (1) PEREQ：协处理器向 80386 发出的请求信号，有效时表示协处理器请求与存储器之间传送数据。80386 响应该请求后，随即控制全部请求控制引脚进行读写。 (2) $\overline{BUSY}$：协处理器向 80386 发出的状态信号，有效时表示协处理器计算执行命令，处于忙状态，暂时不能接受新的指令。 (3) $\overline{ERROR}$：协处理器向 80386 发出的状态信号，有效时表示协处理器出错，80386 在检测到 $\overline{ERROR}$ 信号后，根据当前情况处理了程序发生异常事件。
	(1) D/$\overline{C}$：数据/控制信号，输出，表示当前是数据传送周期还是控制周期。 (2) $\overline{NA}$：“下一个地址”请求信号，输入，有效时表示请求地址流水线进行操作。 (3) $\overline{ADS}$：地址状态信号，三态输出，高阻于 8086 的 ALE 信号。

3. 简述 80386 CPU 寄存器的组成、特点及作用。

【解】80386 共有 34 个寄存器，按功能可分为：通用寄存器、段寄存器、状态和控制寄存器、系统地址寄存器、调试寄存器及测试寄存器。

80386 的 8 个通用寄存器和 8086 通用寄存器相同，只是扩展到 32 位，分别是：EAX（累加器）、EBX（基址寄存器）、ECX（计数寄存器）、EDX（数据寄存器，在 I/O 指令中可用作为口地址寄存器，乘除指令中用作辅助累加器）、ESI（源变址寄存器）、EDI（目的变址寄存器）、EBP（基址指针寄存器）、ESP（堆栈指针寄存器）

80386 的 6 个段寄存器分别是：CS 代码段寄存器，DS 数据段寄存器，SS 堆栈段寄存器，ES、FS、GS 为三个附加段寄存器。在实方式下，段寄存器的用法和 8086 系统相同，只是增加了两个附加段寄存器 FS、GS。在保护方式下，段寄存器称为段选择符，与描述符配合实现段寻址。

64 位的段描述符寄存器对程序员是不可见的。为了加快对内存中描述符表的查询速度，在段选择符内容装入时，段描述符同时装入段描述符寄存器。这样，只要段选择符内容不变，就不需要到内存中查描述符表，从而加快了段地址寻址的速度。描述符寄存器的内容包括段基地址、段限和段属性。段限指出本段的实际长度，与段属性一起主要用于段保护，防止不同任务进入不该进入的段进行操作。

80386 的状态和控制寄存器由标志寄存器 EFLAGS，指令指针寄存器 EIP 和四个控制寄存器 CR0～CR3 组成。

80386 有四个系统地址寄存器，用来保护操作系统需要的保护信息和地址转换表信息，定义目前正在执行任务的环境、地址空间和中断向量空间。

80386 为调试提供了硬件支持，芯片内设有 DR_0～DR_7 八个调试寄存器，调试寄存器主要为系统程序设计人员准备。

80386 有 8 个 32 位的测试寄存器，其中 TR0～TR5 保留备用。TR6～TR7 用于控制对转换后备缓冲器（TLB）中 RAM 和 CAM（内容可寻址寄存器）的测试。TR6 是测试命令寄存

器。TR7 为测试数据寄存器，其中保存测试结果的状态。

4. 简述 80386 的二种工作模式的特点和异同。

【解】80386 有 3 种工作模式：实地址模式（简称为实模式），保护虚拟地址模式（简称为保护模式），虚拟 8086 模式（简称为虚拟 86 模式）。

实模式：

80386 加电启动或复位后自动进入这一模式。实模式主要功能是初始化 80386，为建立保护模式做准备。在实模式下，80386 的工作方式与 8086 相似，可保持 80386 与 8086 兼容：地址总线仍为 20 位，不用虚拟地址的概念，存储器最大容量仍为 1MB，其寻址机制、存储器管理均与 8086 相同；数据总线为 32 位，数据总线与地址总线是相互独立的，内部寄存器主要作为 16 位使用，操作数默认长度是 16 位。也可以按 32 位使用，这时要在指令加上越权访问前缀；中断处理结构与 8086 相同；80386 具有 4 级特权级，程序运行在最高级（0 级）上，除少数几条指令外，80386 的绝大部分指令均可在实模式下执行。

保护模式：

保护模式是 80386 最常用的工作模式，通常在 80386 加电启动或复位后首先进入实模式，完成初始化工作后立即进入保护模式。所谓保护，主要是对存储器的保护，即对存储器中存放的程序和数据的保护。80386 运行在保护模式下，可实现对多任务、多道程序的复杂管理，也只有在保护模式下，80386 才能够真正发挥其强大的功能。

在保护模式下，采用虚拟存储器的概念，存储空间可使用虚拟地址空间、线性地址空间、物理地址空间。通过存储器管理部件，操作系统可以将磁盘等外存设备映射到内存，使程序员可使用的逻辑地址空间大大超过实际内存的物理地址空间。程序指令的操作数和段内的偏移地址都是 32 位，地址总线也是 32 位，物理地址空间为 2^{32}B=4GB。但对内存单元的访问要通过一种称为描述符的数据结构才能实现。80386 具有 4 级特权级，可实现程序与程序之间、用户程序与操作系统之间的隔离和保护，为多任务操作系统提供了有效的支持。

虚拟 86 模式：

在虚拟 86 模式下，不用虚拟地址的概念，存储器最大容量仍为 1MB，其寻址机制与 8086 相同。但存储管理机制与 8086 不同，它把 1MB 的存储空间分为 256 个页面，每页 4KB。这时，当多道程序同时运行时，可以使只中的一个或多个任务使用虚拟 86 模式，并使某一个任务占用存储器的某些页面，而另一个任务占用存储器的另外一些页面，这样就可将多个任务分别转换到物理存储器的不同存储位置，实现了多任务同时运行。在虚拟 86 模式下，程序运行在最低特权级（3 级）上，这时 80386 的一些特权指令是不能使用的。

80386 的上述 3 种工作模式可以相互转换。在实模式下，通过 LMSW 或数据传送指令，将控制寄存器 CR0 的第 0 位（即 PE，允许保护控制位）置为 1，即可进入保护模式。通过数据传送指令，将 PE 置为 0，即可从保护模式返回到实模式。在保护模式下，通过执行 IRETD 指令或进行任务转换，可以进入虚拟 86 模式。通过中断操作，可以从虚拟 86 模式转换到保护模式。

5. 什么是逻辑地址和物理地址，逻辑地址、线性地址和物理地址二者之间的关系是什么？

【解】逻辑地址：用户程序中所使用的地址称为逻辑地址。

物理地址：完成存储器单元或 I/O 端口寻址的实际地址。

程序提供的逻辑地址，包括偏移地址和段选择符两部分。逻辑地址由两部分组成：低 32

位为偏移地址，可指向 4GB 空间中的任何地址；高 16 位为选择符，指向段描述符表（段描述符表由操作系统管理）的一个表项，即一个段描述符。段描述符给出一个段基地址，该段基地址与偏移地址相加，产生线性地址。当不采用分页机制时，该线性地址就可用作存储器的物理地址，即出现在地址总线上的地址。当采用分页机制时，线性地址通过分页机构再转换成物理地址。

6. 简述 80486 CPU 的组成及各部分的作用。

【解】486 微处理器的内部结构包括九个功能单元，这些单元是：总线接口单元、高速缓存（CACHE）、指令预取单元、指令译码单元、控制单元、整数和数据通路单元、浮点单元、分段单元和分页单元等。

总线接口单元用于数据传输，指令预取和处理器内部单元与外部系统的控制功能。

CACHE 单元存储当前读入的指令、操作数及其它数据的副本。

指令预取单元：当指令执行中不使用总线周期时，指令预取单元就通过总线接口单元预取指令。

指令译码单元从指令预取单元接受指令，将其译码成低级控制信号和微代码入口指针。

控制单元的功能是解释指令字和从译码单元获得的微代码入口指针。

整数（数据通路）单元：数据在整数单元中存储并完成 386 处理器指令及几条新增指令的所有算术逻辑运算。

浮点单元执行协处理器 387 同样的指令组。

分段单元将程序发出的逻辑地址转换成线性地址，并将此线性地址发向分页单元和 CACHE。

分页单元用把程序和数据一部分存在存储器中，一部分存在磁盘上的方法，能够存取的数据结构远大于实际的物理空间。

7. Pentium 微处理器采用了哪些新的技术和结构？

【解】Pentium 新型体系结构的特点可以归纳为以下四个方面：

(1) 超标量流水线

超标量流水线（Superscalar）设计是 Pentium 处理器技术的核心。它由 U 与 V 两条指令流水线构成。每条流水线都拥有自己的 ALU、地址生成电路和数据 CACHE 的接口。这种流水线结构允许 Pentium 在单个时钟周期内执行两条整数指令。比相同频率的 486DX CPU 性能提高了一倍。

(2) 独立的指令 CACHE 和数据 CACHE

Pentium 片内有两个 8K CACHE B，一个作为指令 CACHE，另一个作为数据 CACHE，即双路 CACHE 结构，指令和数据分别使用不同的 CACHE，使 Pentium 的性能大大超过 486 微处理器。

(3) 重新设计的浮点单元

Pentium 的浮点单元在 486 的基础上进行了彻底的改进。其执行过程分为 8 级流水，使每个时钟周期能完成一个浮点操作。

(4) 分支预测

Pentium 提供一个称为分支目标缓冲器 BTB（Branch Target Buffer）的小 CACHE 来动态地预测程序分支，当一条指令导致程序分支时，BTB 记下这条指令和分支目标的地址，并用

这些信息预测这条指令再次产生分支时的路径，预先从此处预取指令，保证流水线的指令预取步骤不会空置。因此循环越多，BTB 的效益越明显。

第 4 章　指令系统

教材习题解答

1. 若 DS=3000H，BP=2000H，SI=1000H，[32000H]=00H，[32001H]=40H，SS=3000H，[31000H]=20H，[31001H]=60H，[33000H]=50H，[33001H]=60H，说明下列各条指令执行后，AX 中的内容是什么？并说明各条指令中源操作数的寻址方式。

(1) MOV　AX，DS　　(2) MOV AX，[2000H]

(3) MOV　AX，[SI]　　(4) MOV AX，[BP]

(5) MOV　AX，[BP+SI]

【解】

(1) AX＝3000H　寄存器寻址

(2) AX＝4000H　直接寻址

(3) AX＝6020H　寄存器间接寻址

(4) AX＝4000H　寄存器间接寻址

(5) AX＝6020H　基址加变址寻址

2. 指出下列指令中的非法指令。

(1) MOV　BX，AL　　(2) MOV　CS，2000H

(3) PUSH　4567H　　(4) XCHG VAR1，VAR2

(5) ADD　AL，148H　　(6) MOV　DS，2000H

(7) MOV　BH，SI　　(8) SUB　38H，AL

【解】非法指令：

(1) MOV　BX，AL　　BX 和 AL 的长度不一致

(2) MOV　CS，2000H　　CS 段寄存器不能作目的操作数

(3) PUSH4567H　　只能将寄存器或存储单元的内容压入堆栈，不能是立即数

(4) XCHG VAR1，VAR2　　两个存储单元之间不能直接交换数据

(5) ADD　AL，148H　　运算溢出

(6) MOV　DS，2000H　　立即数不能直接传送给 DS 段寄存器

(7) MOV　BH，SI　　BH 和 SI 的长度不一致

(8) SUB　38H，AL　　目的操作数不能是立即数

3. 若 SP=2000H，AX=3355H，BX=4466H，试指出下列指令或程序段执行后有关寄存器的内容。

(1) PUSH　AX

执行后 AX=？，SP=？

(2) PUSH　AX

PUSH　BX

POP　DX

POP CX

执行后 AX=？，CX=？，DX=？，SP=？

【解】(1) 执行后 AX=3355H，SP=1FFEH

(2) 执行后 AX=3355H，CX=3355H，DX=4466H，SP=2000H

4. 请按下面的要求写出相应的汇编指令或指令序列。

(1) 将 1234H 送入 DS 中。

(2) 将 5678H 与 AX 中的数相加，结果放在 AX 中。

(3) 将 DATAX 和 DATAY 相加，其和放在 DATAY 中。

(4) 将 AX 中的高 4 位变为全 0。

(5) 将 BX 中的低 2 位变为全 1。

(6) 将 CX 中的 D3～D7 位取反。

【解】

(1) MOV AX，1234H

MOV DS，AX

(2) ADD AX，5678H

(3) MOV AX，DATAX

ADD AX，DATAY

MOV DATAY，AX

(4) AND AX，0FFFH

(5) OR AX，0003H

(6) MOV AX，CX

AND AX，07H

XOR CX，0FFFFH

AND CX，0FFF8H

ADD CX，AX

5. 若 AL=0FFH，BL=03H，指出下列指令执行后标志 AF、OF、ZF、SF、PF、CF 的状态。

(1) ADD BL，AL (2) INC BL

(3) SUB BL，AL (4) NEG BL

(5) AND BL，AL (6) MUL BL

(7) CMP BL，AL (8) IMUL BL

(9) OR BL，AL (10)XOR BL，BL

【解】执行后：

	AF	OF	ZF	SF	PF	CF
(1) ADD BL，AL	1	0	0	0	0	1
(2) INC BL	0	0	0	0	0	不影响
(3) SUB BL，AL	1	0	0	0	0	1
(4) NEG BL	1	0	0	1	0	1
(5) AND BL，AL	任意值	0	0	0	1	0
(6) MUL BL	任意值	1	任意值	任意值	任意值	1

(7) CMP BL，AL	1	0	0	0	0	1
(8) IMUL BL	任意值	1	任意值	任意值	任意值	1
(9) OR BL，AL	任意值	0	0	1	1	0
(10) XOR BL，BL	任意值	0	0	1	1	0

6. 已知存储器中有两个压缩 BCD 码 6543 和 4672 存放在从 BUF 开始的连续 4 个单元中(高字节放在高地址单元)，试编制两数相减的程序，结果存放在后面的两个单元中。

【解】程序段如下：

```
MOV  BX，BUF
MOV  AX，[BX]
MOV  CX，[BX+2]
SUB  AL，CL
MOV  DL，AL
DAA
MOV  AL，AH
SBB  AL，CH
DAA
MOV  AH，AL
MOV  AL，DL
MOV  [BX+4]，AX
```

7. 假设 DX=36A5H，CL=3，CF=1，确定下列各条指令执行后 DX 和 CF 的值。

(1) SHR DX，1　　(2) SAR DX，CL
(3) SHL DX，CL　　(4) SHL DL，1
(5) ROR DX，CL　　(6) ROL DL，CL
(7) SAL DH，1　　(8) SAR DH，CL
(9) RCL DX，CL　　(10)RCR DX，CL

【解】

(1) 指令执行后 DX：1B52H，CF：1
(2) 指令执行后 DX：06D4H，CF：1
(3) 指令执行后 DX：6D4AH，CF：0
(4) 指令执行后 DX：364AH，CF：1
(5) 指令执行后 DX：0C6D4H，CF：1
(6) 指令执行后 DX：362DH，CF：1
(7) 指令执行后 DX：6CA5H，CF：0
(8) 指令执行后 DX：06A5H，CF：1
(9) 指令执行后 DX：0B529H，CF：1
(10) 指令执行后 DX：0A6D4H，CF：1

8. 编写程序段将寄存器 AL 中的 8 位二进制数的内容颠倒过来，即将原来的最高位变为最低位，次高位变为次低位，以此类推，若原 AL 的内容为 01110001B，则颠倒之后变为 10001110B。

【解】程序段如下：

```
        XOR AH，AH          ；AH 清零
        MOV CX,8            ；重复 8 次
LP1:    SHR AL，1           ；逻辑右移,将最最低位移入 CF
        RCL AH，1           ；带进位的循环左移，将 CF 移入 AH 的最低位
        LOOP LP1
        MOV AL，AH          ；存回 AL
```

9. 执行下列程序段，指出各相关寄存器的内容。

```
MOV     AX，0A0BH
DEC     AX
SUB     AX，0FFH
AND     AX，00FFH
MOV     CL，3
SAL     AL，CL
ADD     AL，25H
XCHG    AL，AH
PUSH    AX
POP     BX
INC     BL
```

【解】各相关寄存器的内容：

```
MOV     AX，0A0BH       ；AX：0A0BH
DEC     AX              ；AX：0A0AH
SUB     AX，0FFH        ；AX：090BH
AND     AX，00FFH       ；AX：000BH
MOV     CL，3           ；CL：03H
SAL     AL，CL          ；AX：0058H
ADD     AL，25H         ；AX：007DH
XCHG    AL，AH          ；AX：7D00H
PUSH    AX              ；AX：7D00H
POP     BX              ；BX：7D00H
INC     BL              ；BX：7D01H
```

10. 用串操作指令将 100H 个字符从 2100H 处搬到 1000H 处，并且从中检索与 AL 中所存字符相同的存储单元，并将该单元的内容替换成空格，本程序只替换检索到的第一个相同的单元。请在下列空格中填入合适的指令，使程序段完成上述功能。

```
        ⋮
MOV   SI，2100H
(1)_____
MOV   CX，100H
CLD
```

```
        (2)
        MOV   DI，1000H
        MOV   CX，100H
        (3)
        JNZ   K1
        (4)
        MOV   [DI]，20H
K1:     ┆
```

【解】

(1) MOV DI，1000H

(2) REP MOVSB

(3) REPNE SCASB

(4) DEC DI

11. 试分析下列程序段，如果 AX 和 BX 的内容分别为下列 5 种情况，问程序分别转向何处？

(1) AX=147FH，BX=80DCH

(2) AX=0B586H，BX=54B5H

(3) AX=42C0H，BX=608AH

(4) AX=0D023H，BX=9FD7H

(5) AX=94B7H，BX=0B568H

```
        ADD   AX，BX
        JNO   L1
        JNC   L2
        SUB   AX，BX
        JNC   L3
        JNO   L4
        JMP   L5
```

【解】

(1) L1：不溢出，转移到 L1

(2) L1：不溢出，转移到 L1

(3) L2：溢出，不转移到 L1，进位为 0，转移到 L2

(4) L3：溢出，不转移到 L1，进位为 1，不转移到 L2，借位为 0，转移到 L3

(5) L4：溢出，不转移到 L1，进位为 1，不转移到 L2，借位为 1，不转移到 L3，不溢出，转移到 L4

12. 编程求出 AX 中存放的 16 位二进制数中‘1’的个数，将其存入 CL 中（若 AX=1001010011001011B 则将 8 存入 CL）。

【解】程序段如下：

```
        MOV CX，16
        XOR BX，BX
```

```
L1:   SHL AX，1          ；最低位移入 CF
      RCL BL，1          ；CF 移入最低位
      ADD BH，BL
      XOR BL，BL
      LOOP L1
      MOV CL，BH
```

13. 已知 BUF 单元有一个单字节无符号数 X，按要求编写一程序段计算 Y（仍为单字节数），并将其存于零加器。

$$Y=\begin{cases}3X, & X<20\\ X-20, & X\geqslant 20\end{cases}$$

【解】程序段如下：

```
      XOR AX，AX
      MOV AL，BUF
      CMP AL，20
      JNB L1
      MOV BL，3
      MUL BL
      JMP END
L1:   SUB AX，20
END:  HLT
```

第 5 章　汇编语言程序设计

教材习题解答

1. 下列语句在存储器中分别为变量分配多少字节？

```
VR1     DW  9
VR2     DW  4  DUP(? )
CONT    EQU 10
VR3     DD  CONT  DUP(? )
VR4     DB  2  DUP(? , CONT DUP(0))
VR5     DB  'HOW ARE YOU? '
```

【解】VR1：2B ；　VR2：8B ；　CONT：1B ；　VR3：40B；　VR4：44B；　VR5：12B

2. 根据下列数据定义，写出各条指令执行后的结果

```
TABLE   DW 100  DUP(? )
ARRAY   DB 'ABCD'
RES     DB ?
(1)  MOV    AX, TYPE  RES        AX=（    ）
(2)  MOV BX, TYPE  TABLE BX=（    ）
(3)  MOV CX, LENGTH  TABLE  CX=（    ）
(4)  MOV    SI, SIZE  TABLE      SI=（    ）
(5)  MOV    DI, LENGTH  ARRAY    DI=（    ）
```

【解】(1)　AX=1　　(2)　BX=2　　(3)　CX=100　　(4)　SI=200　　(5)　DI=4

3. 下面定义的是一个数据段，请画示它们在存储器中的存放形式。

```
DATA  SEGMENT
A     DB   1, 2, 3, 4
B     DB   'ABCD'
C     DB   4  DUP(0)
N     EQU  12
X     DW   33, 0302H
Y     DD   0ABCDH
DATA  ENDS
```

【解】注：括号中为段内偏移地址

(0000H) A

1
2
3
4

(0008H) C

0
0
0
0

(0011H) Y

03H
0CDH
0ABH
00H

(0004H) B	A	(000CH) N	12		00H
	B	(000DH) X	17H		...
	C		00H		
	D		02H		

4. 试定义一个完整的数据段，首先将 10 个压缩的 BCD 码 29 存放在 ARRAY 变量字节单元，紧接着把-28，6，45，39，3 存放在 ALPHA 数组变量的字单元中，最后从字节变量 BUFFER 单元开始预留 100 个字单元备用。

【解】

```
DATA    SEGMENT
ARRAY   DB    10 DUP(29H)
ALPHA   DW    -28，6，45，39，3
BUFFER  DB    100 DUP(？)
DATA    ENDS
```

5. 请定义一个结构，描述一个学生的简况。该结构应含有以下内容：姓名，年龄，性别，籍贯，民族，入学成绩，名次。说明如何定义结构变量和引用结构变量。

【解】

```
STUDENT  STRUC
NAME     DW 5 DUP(？)
AGE      DB  (？)
SEX      DW  (？)
HOMEPLA  DW 10 DUP(？)
RACE     DW 5 DUP(？)
GRADE    DB  (？)
SORT     DB  (？)
STUDENT  ENDS
```

定义结构变量的格式：结构变量名　结构名　〈字段值表〉

例如：STU1　STUDENT　(1，'ZHANG')

引用结构变量的格式：结构变量名．结构字段名，例如：STU1.NAME

6. 实现满足下面要求的宏定义。

(1) 任意两个单元中的数据相加存于第三个单元中。

(2) 任意 8 位寄存器中的数据转换为 ASCII 码并在屏幕上显示。

【解】(1) 宏定义的代码段如下：

```
ADDM    MACRO   M1，M2，M3
        MOV     AX，[M1]
        ADD     AX，[M2]
        MOV     [M3]，AX
        ENDM
```

(2) 设 8 位寄存器存储的是无符号数，最高位是自位。宏定义的代码段如下：

```
ADDM    MACRO   Y                    ；形式参数 Y 代表一个 8 位寄存器
```

```
        MOV     AL，Y
        XOR     AH，AH
        MOV     DL，100
        DIV     DL
        ADD     AL，30H
        MOV     DL，AL
        MOV     AL，AH                  ；存余数
        MOV     AH，2
        INT     21H                     ；显示百位数
        XOR     AH，AH
        MOV     DL，10
        DIV     DL
        ADD     AL，30H
        MOV     DL，AL
        MOV     AL，AH                  ；存余数
        MOV     AH，2
        INT     21H                     ；显示十位数
        ADD     AL，30H
        MOV     DL，AL
        MOV     AH，2
        INT     21H                     ；显示个位数
        ENDM
```

7. 对下面程序进行注释，并说明其功能。

```
DATA    SEGMENT
A       DB  '123ABC'
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS：CODE，DS：DATA
START：  MOV  AX，DATA
        MOV  DS，AX
        LEA  BX，A
        MOV  CX，6
        MOV  AH，2
LP：     MOV  AL，[BX]
        XCHG  AL，DL
        INC  BX
        INT  21H
        LOOP  LP
        MOV  AH，4CH
```

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

【解】注释如下：

```
DATA    SEGMENT                 ；数据段开始
A       DB  '123ABC'
DATA    ENDS                    ；数据段结束
CODE    SEGMENT                 ；代码段开始
        ASSUME  CS：CODE，DS：DATA
START：  MOV  AX，DATA
        MOV  DS，AX             ；装填数据段
        LEA  BX，A              ；传送变量的地址
        MOV  CX，6
        MOV  AH，2
LP：  MOV  AL，[BX]
        XCHG  AL，DL
        INC  BX
        INT  21H                ；系统功能调用，功能号：2，
                                ；将字符 1、2、3、"A"、"B"、"C" 送屏幕显示
        LOOP  LP                ；CX=6，循环 6 次
        MOV  AH，4CH
        INT  21H                ；返回 DOS
CODE    ENDS                    ；代码段结束
        END  START              ；源程序结束
```

程序的功能：显示从 A 开始的 6 个字节单元中的字符。

8. 编程将 CX，DX 作为双字联合右移四位（设 CX 为高 16 位），最高 4 位送入全 1。

【解】程序代码如下：

```
CODE    SEGMENT                 ；代码段开始
        ASSUME  CS：CODE
START：  MOV  AX，CX
        MOV  CX，4
LP：    SHR  AX，1
        RCR  DX，1
        LOOP LP
        OR   AX，0F000H
        MOV  CX，AX
        MOV  AH，4CH
        INT  21H                ；返回 DOS
CODE    ENDS                    ；代码段结束
```

```
        END   START                    ；源程序结束
```

9. 编程把从 A 单元开始存放的 3 个单字节无符号数按递增顺序排序后存回原处。

【解】程序代码如下：

```
DATA    SEGMENT                        ；数据段开始
A       DB   3 DUP(? )
DATA    ENDS                           ；数据段结束
CODE    SEGMENT                        ；代码段开始
        ASSUME  CS：CODE，DS：DATA
START：  MOV   AX，DATA
        MOV   DS，AX                   ；装填数据段
        LEA   BX，A                    ；作近变量偏地址
        MOV   AL，[BX]
        CMP   AL，[BX＋1]
        JAE   EX1                      ；[BX]>[BX＋1]，交换
CMP2：   CMP   AL，[BX＋2]
        JAE   EX2                      ；[BX]>[BX＋2]，交换
        JMP   CONT1
EX1：    XCHG  AL，[BX＋1]
        XCHG  AL，[BX]
        MOV   AL，[BX]
        JMP   CMP2
EX2：    XCHG  AL，[BX＋2]
        XCHG  AL，[BX]
        MOV   AL，[BX]
CONT1：  MOV   AL，[BX＋1]
        CMP   AL，[BX＋2]
        JB    CONT2
        XCHG  AL，[BX＋2]              ；[BX＋1]>[BX＋2]，交换
        XCHG  AL，[BX＋1]
CONT2：  MOV   AH，4CH
        INT   21H                      ；返回 DOS
CODE    ENDS                           ；代码段结束
        END   START                    ；源程序结束
```

10. 试编写一段程序比较从 ARRAY 开始存放的 3 个 16 位有符号数，根据比较结果置 FLAG 标志。

(1) 如果 3 个数都不相等，置 FLAG 为 0。

(2) 如果 3 个数中有 2 个数相等，置 FLAG 为 1。

(3) 如果 3 个数都相等，置 FLAG 为 2。

【解】程序代码如下：

```
DATA    SEGMENT                    ；数据段开始
ARRAY   DW  3 DUP(? )
FLAG    DB  ?
DATA    ENDS                       ；数据段结束
CODE    SEGMENT                    ；代码段开始
        ASSUME  CS：CODE，DS：DATA
START：  MOV  AX，DATA
        MOV  DS，AX                ；装填数据段
        LEA  BX，ARRAY             ；指向变量的地址
        MOV  AX，[BX]
        CMP  AX，[BX＋2]
        JE   EQU1                  ；[BX]－[BX＋2]，执行 MOV DL，1
        MOV  DL，0
        JMP  CONT1
EQU1：  MOV  DL，1
CONT1：  CMP  AX，[BX＋4]
        JE   EQU2                  ；[BX]－[BX＋4]，DL＋1
        JMP  CONT2
EQU2：  ADD  DL，1
CONT2：  MOV  AX，[BX＋2]
        CMP  AX，[BX＋4]
        JE   EQU3
        JMP  CONT3
EQU3：  ADD  DL，1
CONT3：  CMP  DL，3
        JNZ  EQU4                  ；[BX]－[BX＋2]－[BX＋4]，DL－1
        SUB  DL，1
EQU4：  MOV  FLAG，DL
        MOV  AH，4CH
        INT  21H                   ；返回 DOS
CODE    ENDS                       ；代码段结束
        END  START                 ；源程序结束
```

11. 分析下列程序，指出运行结果。

```
DATA    SEGMENT
SUM     DW  ?
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS：CODE，DS：DATA
START：  MOV  AX，DATA
```

```
        MOV   DS，AX
        XOR   AX，AX
        MOV   CX，10
        MOV   BX，2
LP：    ADD   AX，BX
        INC   BX
        INC   BX
        LOOP  LP
        MOV   SUM，AX
        MOV   AH，4CH
        INT   21H
CODE    ENDS
        END   START
```

请问：

(1) 该程序完成的功能是_____。

(2) 程序执行后，SUM 单元的内容是_____。

【解】

(1) 该程序完成的功能是计算 0～20 的偶数和。

(2) 程序执行后，SUM 单元的内容是 110。

12. 从 ARRAY 开始的单元中存有 10 个 16 位无符号数，试编一完整程序找出其中最小数并存入 MIN 单元。

【解】程序代码如下：

```
DATA    SEGMENT                    ；数据段开始
ARRAY   DW 0F454H，4560H，0D214H，8354H，8210H，
        0A673H，5656H，0021H，0567H，422BH
MIN     DW ?
DATA    ENDS                       ；数据段结束
CODE    SEGMENT                    ；代码段开始
        ASSUME  CS：CODE，DS：DATA
START：MOV  AX，DATA
        MOV   DS，AX               ；装填数据段
        LEA   BX，ARRAY            ；作这变至首地址
        MOV   AX，[BX]
        MOV   CX，9
LP1：   INC   BX
        INC   BX
        CMP   AX，[BX]
        JB    CONT1                ；AX<[BX]，转移
        MOV   AX，[BX]             ；AX>=[BX]，AX←[BX]
```

```
CONT1:  LOOP  LP1
        MOV   MIN, AX
        MOV   AH, 4CH
        INT   21H                ; 返回 DOS
CODE    ENDS                     ; 代码段结束
        END   START              ; 源程序结束
```

13. 从 DAT 开始的数据区中放有 100 个 8 位的无符号数，编程统计其中奇数的个数，将结果存入 RESULT 单元。

【解】程序代码如下：

```
DATA    SEGMENT                  ; 数据段开始
DAT     DB  26, 45, 44, 32, 90, 2, 1, 81, 2, 120, 5……
RESULT  DB  ?
DATA    ENDS                     ; 数据段结束
CODE    SEGMENT                  ; 代码段开始
        ASSUME  CS: CODE, DS: DATA
START:  MOV  AX, DATA
        MOV  DS, AX              ; 装填数据段
        LEA  BX, DAT             ; 传送变量的地址
        XOR  AX, AX
        XOR  DX, DX
        MOV  DL, 2
        MOV  CX, 100
LP1:    MOV  AL, [BX]
        DIV  DL                  ; 余数在 AH 中
        CMP  AH, 0
        JE   CONT1
        INC  DH                  ; DH 用作计数器
CONT1:  INC  BX
        LOOP  LP1
        MOV   RESULT, DH
        MOV  AH, 4CH
        INT  21H                 ; 返回 DOS
CODE    ENDS                     ; 代码段结束
        END  START               ; 源程序结束
```

14. 利用 DOS 系统功能调用，将键盘输入的小写字母转换成大写字母输出显示，直到输入“$”字符时停止输出。

【解】调用 1 号系统功能，键入的字符送 AL 寄存器，并送屏幕显示输出。调用 2 号系统功能，将 DL 寄存器中的内容送屏幕显示输出。程序代码如下：

```
CODE    SEGMENT                  ; 代码段开始
```

```
        ASSUME  CS：CODE
START：  XOR   AX，AX
INPU：   MOV   AH，1
        INT    21H              ；键盘输入单字符送 AL
        MOV    BL，'$'
        CMP    AL，BL
        JE     EXT
        SUB    AL，20H          ；转换成大写字母
        MOV    DL，AL
        MOV    AH，2
        INT    21H              ；将 DL 中的字符送屏幕显示
        JMP    INPU
EXT：   MOV   AH，4CH
        INT    21H              ；返回 DOS
CODE    ENDS                    ；代码段结束
        END    START            ；程序结束
```

15. 建立两个过程：一个过程将 16 进制数变换成 ASCII 码，一个过程将 ASCII 码字符在屏幕上显示输出。

【解】程序代码如下：

```
DATA    SEGMENT                 ；数据段开始
LIST    DB   ?                  ；存储一个十六进制数(1 位)
ASC     DB   ?                  ；存储转换后 ASCII 码
DATA    ENDS                    ；数据段结束
CODE    SEGMENT                 ；代码段开始
        ASSUME  CS：CODE，DS：DATA，ES：DATA
START：  MOV   AX，DATA
        MOV    DS，AX           ；装填数据段
        CALL   CHANGE
        CALL   OUTPTR
        MOV    AH，4CH
        INT    21H              ；返回 DOS
CHANGE  PROC
        MOV    AL，LIST
        CMP    AL，9
        JA     NEXT1            ；AL>9，转移
        OR     AL，30H
        JMP    NEXT2
NEXT1：  ADD    AL，37H
NEXT2：  MOV    ASC，AL
```

```
        RET
CHANGE  ENDP
OUTPTR  PROC
        MOV   DL，ASC
        MOV   AH，2
        INT   21H
        RET
OUTPTR  ENDP
CODE    ENDS                    ；代码段结束
        END   START             ；源程序结束
```

16. 要求编写程序比较两个最大长度为 100 个字符的字符串 ALFA1 和 ALFA2 是否相等，若相等则输出显示“All Right!”，否则输出显示“Error!”。

【解】完成输出显示功能时，调用 9 号系统功能，功能是将指定的内存缓冲区中的字符串在屏幕上显示出来，缓冲区的字符串以“$”为结束标志。

LEN 指定的内存中存储两字符串中长串的长度，否则，即使两串相等，由于字符串后面的随机字符，将影响判断结果。

程序代码如下：

```
DATA    SEGMENT                 ；数据段开始
ALFA1   DB  100 DUP(?)
ALFA2   DB  100 DUP(?)
RESULT1 DB  'All Right!$'
RESULT2 DB  'Error!$'
LEN     DB  100                 ；字符串长度
DATA    ENDS                    ；数据段结束
CODE    SEGMENT                 ；代码段开始
        ASSUME  CS：CODE，DS：DATA，ES：DATA
START:  MOV   AX，DATA
        MOV   DS，AX            ；装填数据段
        MOV   ES，AX            ；装填附加段
        LEA   BX，ALFA1         ；传送源串的地址
        MOV   SI，BX
        LEA   BX，ALFA2         ；传送目的串的地址
        MOV   DI，BX
        CLD                     ；前方向标志，使正向传送
        XOR   CX，CX
        MOV   CL，LEN
        REPNZ  CMPSB
        JZ    EQQ               ；若 ZF=1，说明两串相等，转 EQQ
        LEA   DX，RESULT2       ；输出显示“Error!”
```

```
        JMP   OUT1
EQQ:    LEA   DX，RESULT1    ；输出显示“All Right!”
OUT1:   MOV   AH，9
        INT   21H
STOP:   MOV   AH，4CH
        INT   21H            ；返回 DOS
CODE    ENDS                 ；代码段结束
        END   START          ；源程序结束
```

17. 有一个最大长度为 80 个字符的字符串 STRING，试编写程序找出第一个空格的位置（用 00H～4FH 表示），并存入 CL 中。若该串无空格，则将 0FFH 存入 CL 中。

【解】程序代码如下：

```
DATA    SEGMENT                ；数据段开始
STRING  DB  80 DUP(?)
LEN     DB  80
DATA    ENDS                   ；数据段结束
CODE    SEGMENT                ；代码段开始
        ASSUME  CS：CODE，DS：DATA，ES：DATA
START:  MOV   AX，DATA
        MOV   DS，AX           ；装填数据段
        MOV   ES，AX           ；装填附加段
        MOV   AL，' '          ；要查找的关键字（空格）
        LEA   BX，STRING       ；待查串的地址
        MOV   DI，BX
        XOR   CX，CX
        MOV   CL，LEN
        CLD                    ；地方向标志，使正向传送
        REPNZ  SCASB           ；搜索关键字
        JZ     FOUND           ；若 ZF＝1，说明找到，转 FOUND
        MOV   CL，0FFH
        JMP   STOP
FOUND:  MOV   AL，LEN
        SUB   AL，CL           ；CL 中是剩余的字符数
        MOV   CL，AL
STOP:   MOV   AH，4CH
        INT   21H              ；返回 DOS
CODE    ENDS                   ；代码段结束
        END   START            ；源程序结束
```

18. 在 AX 中存放着压缩 BCD 码的十进制数。要求：

(1) 将 AH 中的数转换成二进制数。

(2) 求 AH 与 AL 中数的和，将结果转换成 ASCII 码，然后在屏幕上显示出来。

【解】(1) 将 AH 中的数转换成二进制数，结果存于 DX 中。程序代码如下：

```
CODE    SEGMENT                 ；代码段开始
        ASSUME CS：CODE
START：  MOV   BX，AX            ；保存 AX 中的数据
        AND   AH，0FH           ；千位置 0
        MOV   AL，AH
        MUL   100
        MOV   DX，AX            ；百位结果存于 DX 中
        MOV   AH，BH
        MOV   CL，4             ；移位次数
        SHR   AH，CL            ；千位移到低 4 位
        MOV   AL，AH
        MUL   1000
        ADD   AX，DX
        MOV   DX，AX            ；百位加千位结果存于 DX 中
        MOV   AH，4CH
        INT   21H               ；返回 DOS
CODE    ENDS                    ；代码段结束
        END   START             ；源程序结束
```

(2) 求 AH 与 AL 中数的和，将结果转换成 ASCII 码，然后在屏幕上显示出来。程序代码如下：

```
CODE    SEGMENT                 ；代码段开始
        ASSUME CS：CODE
START：  MOV   BX，AX            ；保存 AX 中的数据
        MOV   CL，4             ；移位次数
        SHR   AH，CL            ；千位移到 AH 低 4 位
        ADD   AH，30H           ；转换成 ASCII 码
        MOV   DL，AH
        MOV   AH，2
        INT   21H               ；调用 2 号系统功能，输出单字符
        MOV   AH，BH
        AND   AH，0FH           ；取低 4 位
        ADD   AH，30H           ；转换成 ASCII 码
        MOV   DL，AH
        MOV   AH，2
        INT   21H               ；调用 2 号系统功能，输出单字符
        MOV   AL，BL
        MOV   CL，4             ；移位次数
```

```
        SHR   AL，CL                ；十位移到低 4 位
        ADD   AL，30H               ；转换成 ASCII 码
        MOV   DL，AL
        MOV   AH，2
        INT   21H                   ；调用 2 号系统功能，输出单字符
        MOV   AL，BL
        AND   AL，0FH               ；取低 4 位
        ADD   AL，30H               ；转换成 ASCII 码
        MOV   DL，AL
        MOV   AH，2
        INT   21H                   ；调用 2 号系统功能，输出单字符
        MOV   AH，4CH
        INT   21H                   ；返回 DOS
CODE    ENDS                        ；代码段结束
        END   START                 ；源程序结束
```

19. 已知从 BUF 单元开始存放着 10 个 8 位无符号数，要求编写汇编语言程序将这 10 个数去掉一个最大的，再去掉一个最小的，将其余的数的算术平均值计算出来并存于 AVERG 单字节单元。

【解】注意：10 个 8 位无符号数的累加和有可能超过 8 位，需要 16 位的寄存器存储。寄存器分配：DL：最小值；DH：最大值；CX：循环次数；AX：累加和。

本题的思路：找出最小值，暂时存于 DL，找出最大值，暂时存于 DH，求 10 个数的和，再减去最小值和最大值，求 8 个数的平均值。

程序代码如下：

```
DATA    SEGMENT                     ；数据段开始
BUF     DB  23H，26H，44H，98H，32H，72H，39H，62H，75H，48H
AVERG   DB  ?
DATA    ENDS                        ；数据段结束
CODE    SEGMENT                     ；代码段开始
        ASSUME  CS：CODE，DS：DATA
START：  MOV   AX，DATA
        MOV   DS，AX                ；装填数据段
        LEA   BX，BUF               ；将首元素偏移地址
        XOR   AX，AX
        MOV   CX，9
        MOV   DL，[BX]
        MOV   DH，[BX]
        MOV   AL，[BX]
LP1：  INC   BX
        CMP   DL，[BX]              ；if DL>[BX], then DL←[BX]
```

```
        JBE   CONT1                ; if DL<=[BX], 转移
        MOV   DL，[BX]
CONT1:  CMP   DH，[BX]             ; if DH-[BX], then DH←[BX]
        JAE   CONT2                ; if DL>=[BX], 转移
        MOV   DX，[BX]
CONT2:  ADD   AL，[BX]
        ADC   AH，0
        LOOP  LP1
        SUB   AL，DL
        SBB   AH，0
        SUB   AL，DH
        SBB   AH，0
        MOV   BL，8
        DIV   BL                   ; 平均值在 AL 中
        MOV   AVERG，AL
        MOV   AH，4CH
        INT   21H                  ; 返回 DOS
CODE    ENDS                       ; 代码段结束
        END   START                ; 源程序结束
```

20. 编程：

(1) 键入某班学生（30 人）的计算机考试成绩。成绩按学号（1~30）排放在 SCORE 数据区中。

(2) 按考分排序（降序），列出相应学号到 ORDER 数据区中。

(3) 在屏幕上显示前三名学生及成绩。

【解】程序代码如下：

```
DATA    SEGMENT                    ; 数据段开始
SCORE   DB  30 DUP(?)              ; 存储按学号排序的成绩
ORDER   DB  30 DUP(?)              ; 按成绩排序的学号
BUFF    DB  30 DUP(?)              ; 暂存降序排序的成绩
DATA    ENDS                       ; 数据段结束
CODE    SEGMENT                    ; 代码段开始
        ASSUME  CS：CODE，DS：DATA
START:  MOV   AX，DATA
        MOV   DS，AX               ; 装填数据段
        LEA   BX，ORDER            ; 作这变量的地址
        MOV   CX，30
        MOV   AL，1
LP1:    MOV   [BX]，AL
        INC   BX
```

```
        INC   AL
        LOOP  LP1                ；存储学号初始值
                        ；以下按价接收键盘输入的成绩，成绩以回车符 0DH 间隔
        LEA   BX，SCORE          ；传送变量的地址
        XOR   AL，AL
        XOR   DL，DL
        MOV   CX，30
INPUT:  MOV   AH，1
        INT   21H                ；等待键盘输入，ASCII 码送 AL 寄存器
        CMP   AL，0110000B
        JB    NEXT1              ；输入的不是数字，有可能是回车符
        CMP   AL，0111001B
        JA    STOP               ；输入的不是数字，也不是回车符，结束程序
        SUB   AL，30H
        MOV   DH，10
        XCHG  AL，DL
        MUL   DH
        ADD   DL，AL             ；暂存输入的数字
        JMP   INPUT
NEXT1:  CMP   AL，0DH
        JNE   STOP               ；输入的不是回车符，也不是数字，结束程序
        MOV   [BX]，DL           ；是回车符，成绩存入存储区
        INC   BX
        XOR   DL，DL             ；DL，准备转换下一个输入的数据
        LOOP  INPUT
                        ；以下成绩存入暂存区
        LEA   BX，SCORE          ；传送数据表首地址
        LEA   SI，BUFF           ；SI←暂存区首地址
        MOV   CX，30
LP2:    MOV   AL，[BX]
        MOV   [SI]，AL
        INC   BX
        INC   SI
        LOOP  LP2
                        ；以下用冒泡法排序
        MOV   CH，30             ；CH←数据长度
        DEC   CH                 ；CH←外循环次数
LOP0:   LEA   SI，BUFF           ；SI←数据表首地址（成绩）
        LEA   BX，ORDER          ；BX←数据表首地址（学号）
```

```
        MOV   CL，CH                ；CL←内循环（比较）次数
        XOR   DL，DL                ；交换标志 DL←0
LOP1：  MOV   AL，[SI]
        CMP   AL，[SI+1]            ；比较相邻两个数
        JGE   NEXT                  ；顺序正确（小数在后），转 NEXT
        XCHG  AL，[SI+1]
        MOV   [SI]，AL              ；顺序不对（大数在后），交换两数
        MOV   AL，[BX]
        XCHG  AL，[BX+1]
        MOV   [BX]，AL              ；交换学号
        OR    DL，01H               ；交换标志 BL←1
NEXT：  INC   SI                    ；调整地址指针
        INC   BX
        DEC   CL
        JNZ   LOP1                  ；内循环未完，转 LOP1 继续内循环
        AND   DL，DL
        JZ    STOP                  ；交换标志为 0，说明顺序排好，转 STOP
        DEC   CH
        JNZ   LOP0                  ；外循环未完，转 LOP0 继续外循环
                                    ；以下显示前三名学号和成绩
        LEA   BX，ORDER
        MOV   CX，3
LP33：  CALL  DISPLAY
        INC   BX
        LOOP  LP33
        MOV   DL，0001101B
        MOV   AH，2
        INT   21H                   ；显示回车
        MOV   DL，0001010B
        MOV   AH，2
        INT   21H                   ；显示换行
        LEA   BX，BUFF
        MOV   CX，3
LP43：  CALL  DISPLAY
        INC   BX
        LOOP  LP43
        MOV   DL，0001101B
        MOV   AH，2
        INT   21H                   ；显示回车
```

```
        MOV   DL，00010108
        MOV   AH，2
        INT   21H                    ；显示换行
        JMP   STOP
DISPLAY PROC                         ；定义显示子程序
        XOR   AX，AX
        MOV   AL，[BX]
        MOV   DL，0AH
        DIV   DL
        ADD   AL，30H
        MOV   DL，AL
        MOV   AH，2
        INT   21H                    ；显示十位数
        XOR   AX，AX
        MOV   AL，[BX]
        MOV   DL，0AH
        DIV   DL
        ADD   AH，30H
        MOV   DL，AH
        MOV   AH，2
        INT   21H                    ；显示个位数
        MOV   DL，01000008
        MOV   AH，2
        INT   21H                    ；显示空格
        RET
DISPLAY ENDP                         ；定义显示子程序结束
STOP:   MOV   AH，4CH
        INT   21H                    ；返回 DOS
CODE    ENDS                         ；代码段结束
        END   START                  ；源程序结束
```

21. 从 TABLE 字节单元开始存有 100 个无符号数，试编程把数组中出现次数最多的数存入 CH 中，其出现次数存入 CL 中。

【解】程序代码如下：

```
DATA    SEGMENT                      ；数据段开始
TABLE   DB   100 DUP(? )
DATA    ENDS                         ；数据段结束
CODE    SEGMENT                      ；代码段开始
        ASSUME CS：CODE，DS：DATA
START:  MOV  AX，DATA
```

```
        MOV   DS，AX                ；装填数据段
        LEA   BX，TABLE             ；传送变量的地址
        XOR   AX，AX
        XOR   DX，DX
        MOV   CX，99
LP2：   PUSH  BX
        PUSH  CX
        MOV   AH，[BX]
        MOV   AL，1
        LP1：   INC   BX
        CMP   AH，[BX]
        JNE   NEXT1                 ；AH!=[BX]，转移
        INC   AL
NEXT1：LOOP   LP1
        POP   CX
        POP   BX
        INC   BX
        CMP   AL，DL
        JB    NEXT2                 ；(AL<DL，本次计数的数字不是最多的
        MOV   DL，AL
        MOV   DH，AH
NEXT2：LOOP   LP2
        MOV   CX，DX
        MOV   AH，4CH
        INT   21H                   ；返回 DOS
CODE    ENDS                        ；代码段结束
        END   START                 ；源程序结束
```

22. 有两个长度不等的字符串，分别存于 STRN1 和 STRN2 单元开始的存储区，字串长度分别存放于 LS1 和 LS2 字节单元，要求编程将短串接在长串之后，并将连接后的串长度存于 LS1 和 LS1+1 单元。

【解】设两串连接后最大长度为 256 个字符，连接后的字符串存于 STRN3 单元开始的存储区。程序代码如下：

```
DATA    SEGMENT                     ；数据段开始
STRN1   DB   'i swear by the moon and the stars in the sky.'
STRN2   DB   'and i swear like the shadow thats by your side. '
STRN3   DB   256 DUP(? )
LS1     DB   45
LS2     DB   48
DATA    ENDS                        ；数据段结束
```

```
CODE    SEGMENT                    ；代码段开始
        ASSUME CS：CODE，DS：DATA
START：  MOV  AX，DATA
        MOV  DS，AX                ；装填数据段
        XOR  AX，AX
        XOR  BX，BX
        MOV  AL，LS1
        MOV  BL，LS2
        CMP  AL，BL
        JA   LINK2                 ；AL>BL，大数在 LS1 中，转移到 LINK2
LINK1：  LEA  BX，STRN2            ；传送变量偏地址
        LEA  SI，STRN3             ；传送变量偏地址
        XOR  CX，CX
        MOV  CL，LS2
LOP1：   MOV  AL，[BX]
        MOV  [SI]，AL
        INC  BX
        INC  SI
        LOOP LOP1
        LEA  BX，STRN1             ；传送变量偏地址
        MOV  CL，LS1
LOP2：   MOV  AL，[BX]
        MOV  [SI]，AL
        INC  BX
        INC  SI
        LOOP LOP2
        JMP  NEXT
LINK2：  LEA  BX，STRN1            ；传送变量偏地址
        LEA  SI，STRN3             ；传送变量偏地址
        XOR  CX，CX
        MOV  CL，LS1
LOP3：   MOV  AL，[BX]
        MOV  [SI]，AL
        INC  BX
        INC  SI
        LOOP LOP3
        LEA  BX，STRN2             ；传送变量偏地址
        MOV  CL，LS2
LOP4：   MOV  AL，[BX]
```

```
        MOV   [SI], AL
        INC   BX
        INC   SI
        LOOP  LOP4
NEXT:   XOR   AX, AX
        XOR   BX, BX
        MOV   AL, LS1
        MOV   BL, LS2
        ADD   AX, BX
        LEA   BX, LS1
        MOV   [BX], AX          ; 将运算后的单元应存于 LS1 和 LS1+1 单元。
STOP:   MOV   AH, 4CH
        INT   21H               ; 返回 DOS
CODE    ENDS                    ; 代码段结束
        END   START             ; 源程序结束
```

23. 从 BUF 字节单元开始存有按增序排好顺序的一个带符号数组，数组长度在 LEN 字节单元。在 POSI 字节单元存有一个正数，要求将其按顺序插入到数组中，并修改数组长度。

【解】程序代码如下：

```
DATA    SEGMENT                 ; 数据段开始
BUF     DB  -33, -29, -6, -4, 4, 21, 33, 34, 34, 49.....
LEN     DB  9
POSI    DB  32
DATA    ENDS                    ; 数据段结束
CODE    SEGMENT                 ; 代码段开始
        ASSUME CS: CODE, DS: DATA
START:  MOV   AX, DATA
        MOV   DS, AX            ; 装填数据段
        LEA   BX, BUF           ; 将这变量的地址
        XOR   CX, CX
        MOV   CL, LEN
        MOV   AH, POSI
LP1:    MOV   AL, [BX]
        CMP   AH, AL
        JL    NEXT              ; AH < AL, 转移
        INC   BX
        LOOP  LP1
        MOV   [BX], AH          ; 新数据插入到数组尾
        JMP   STOP
NEXT:   ADD   BX, CX            ; 指针移向数组尾
```

```
LP2:    MOV   AL，[BX]
        MOV   [BX+1]，AL
        DEC   BX
        LOOP  LP2
        MOV   [BX]，AH          ；插入新数据
STOP:   MOV   AL，LEN
        INC   AL
        MOV   LEN，AL           ；修改数组长度
        MOV   AH，4CH
        INT   21H               ；返回 DOS
CODE    ENDS                    ；代码段结束
        END   START             ；源程序结束
```

24. 从 DAT 单元开始存有 20 个 8 位带符号数构成的数组，要求编程去掉数组中的负奇数，并生成一个新的数组存于从 RES 字节单元开始的存储区中。

【解】程序代码如下：

```
DATA    SEGMENT                 ；数据段开始
DAT     DB  -29，34，4，33，-4，-6，49，-33，34，21，
        67，83，-17，-12，-81，34，85，32，1，91
RES     DB  20 DUP(?)
DATA    ENDS                    ；数据段结束
CODE    SEGMENT                 ；代码段开始
        ASSUME CS：CODE，DS：DATA
START:  MOV   AX，DATA
        MOV   DS，AX            ；装填数据段
        LEA   BX，DAT           ；存源变量的地址
        LEA   SI，RES
        MOV   CX，20
LP1:    MOV   AL，[BX]
        INC   BX
        AND   AL，81H
        CMP   AL，81H
        JZ    NEXT              ；是负奇数，转移
        MOV   AL，[BX-1]
        MOV   [SI]，AL
        INC   SI
NEXT:   LOOP  LP1
        MOV   AH，4CH
        INT   21H               ；返回 DOS
CODE    ENDS                    ；代码段结束
```

```
        END   START                    ；源程序结束
```

25. 已知某数组 ARRAY 中有 50 个 8 位带符号数，试用汇编语言编写一个完整的程序统计该数组中相邻两数之间符号变化（正数变负数或负数变正数）的次数，并将其存于 NUM 存储单元。

【解】程序代码如下：

```
DATA    SEGMENT                        ；数据段开始
ARRAY   DB  -20, 34, 4, 33, -4, -6, 49, -33, 34, 21,
        67, 83, -17, -12, -81, 34, 85, 32, 1, 91,
        15, 73, 95, 14, 28, 16, -41, -34, -9, 54,
        2, 2, 4, 67, -33, -51, -21, 4, 3, 10,
        67, 83, -20, -17, -81, 34, 85, 32, 1, 90
NUM     DB  ?
DATA    ENDS                           ；数据段结束
CODE    SEGMENT                        ；代码段开始
        ASSUME CS: CODE, DS: DATA
START:  MOV  AX, DATA
        MOV  DS, AX                    ；装填数据段
        LEA  BX, ARRAY                 ；传送变量的偏址
        XOR  DL, DL
        MOV  CX, 49
LP1:    MOV  AL, [BX]
        MOV  AH, [BX+1]
        AND  AL, 80H                   ；留符号位，屏蔽其它位
        AND  AH, 80H                   ；留符号位，屏蔽其它位
        CMP  AH, AL
        JE   CONT
        INC  DL
CONT:   INC  BX
        LOOP LP1
        MOV  NUM, DL
        MOV  AH, 4CH
        INT  21H                       ；返回 DOS
CODE    ENDS                           ；代码段结束
        END  START                     ；源程序结束
```

26. 编写完整的汇编语言程序完成如下功能：

首先在屏幕上显示提示行“Input number key, CR or Space return”，显示后回车换行等待用户输入。若用户输入的数字 N 在 1—9 之间，则响铃 N 次（每次要稍延时以作间隔）；若键入的是 0 或非数字字符则不响铃；若键入的是回车或空格，则直接返回 DOS。

【解】程序代码如下：

```
DATA    SEGMENT                 ；数据段开始
STR1    DB  'Input number key, CR or Space return$'
DATA    ENDS                    ；数据段结束
CODE    SEGMENT                 ；代码段开始
        ASSUME  CS：CODE，DS：DATA
START：  MOV   AX，DATA
        MOV   DS，AX            ；装填数据段
        LEA   DX，STR1          ；传送变量的地址
        MOV   AH，9
        INT   21H               ；调用 9 号系统功能，显示内存缓冲区中的字符串
                                ；（以“$”为结束标志）
        MOV   DL，0001101B      ；回车控制符
        MOV   AH，2
        INT   21H               ；调用 2 号系统功能，显示回车
        MOV   DL，0001010B      ；换行控制符
        MOV   AH，2
        INT   21H               ；调用 2 号系统功能，显示换行
CONT：  MOV   AH，7
        INT   21H               ；调用 7 号系统功能，等待用户输入单字符，送 AL
        CMP   AL，0110001B      ；与‘1’比较
        JB    NEXT              ；小于‘1’，转移（CF=1）
        CMP   AL，0111001B      ；与‘9’比较
        JA    NEXT              ；大于‘1’，转移（CF=0，且 ZF=0）
        XOR   CX，CX            ；CX 清零
        MOV   CL，AL            ；响铃的次数
        SUB   CL，30H           ；ASCII 码转换成数值
LP1：   MOV   DL，0000111B      ；响铃控制符
        MOV   AH，2
        INT   21H               ；调用 2 号系统功能，响铃 1 次
        PUSH  CX
        CALL  DELAY             ；调用延时子程序，形成响铃间隔
        POP   CX
        LOOP  LP1               ；响铃 N 次
NEXT：  CMP   AL，0001101B      ；与‘回车符’比较
        JZ    STOP              ；相等，转移
        CMP   AL，0100000B      ；与‘空格’比较
        JZ    STOP              ；相等，转移
        JMP   CONT              ；继续等待下一个输入
STOP：  MOV   AH，4CH
```

```
        INT   21H              ；返回 DOS
DELAY   PROC                   ；定义延时子程序
        MOV   CX，10000        ；向 CX 中送延时常数，决定延时的时间
DELAY1：NOP
        PUSH  CX
        MOV   CX,65535
DELAY2：NOP
        LOOP  DELAY2
        POP   CX
        NOP
        LOOP  DELAY1
        RET
DELAY   ENDP                   ；定义延时子程序结束
CODE    ENDS                   ；代码段结束
        END   START            ；源程序结束
```

第 6 章　微机存储器系统

教材习题解答

1. 内存和外存的功能如何，有什么联系和区别？

【解】内存的存取速度快而容量有限，用于存放 CPU 现行程序和数据；外存的容量大但存取速度较慢，用于存放 CPU 暂时不用或尚未用过的程序。弥补内存的容量不足。当某个时刻 CPU 需要执行那部分程序时，可将程序从外存调入内存，以供 CPU 执行，或者将 CPU 暂时不用的那部分程序从内存调出，存入外存以备特用。

2. 高速缓冲区有什么功能？

【解】随着计算机各部件与工艺的发展，主存储器的速度与 CPU 的速度出现一定的差距，从而成为影响整机提高速度的重要因素。为解决主存储器与 CPU 的速度匹配，在 CPU 与主存储器之间增设一个容量不大而速度很高的存储器，通常叫做“高速缓冲存储器”。CPU 在某一小段时间内要执行的程序，事先从主存储器调入高速缓冲存储器中。当 CPU 执行这些程序时，就直接从高速缓冲存储器中取得，这样减少了访问主存储器的次数，显著提高了 CPU 执行指令的速度。高速缓冲存储器多采用与 CPU 相同类型的半导体集成电路工艺，如双极型集成电路器件，以保证尽量与 CPU 的速度相匹配。

3. RAM 和 ROM 有什么区别？

【解】对于随机存储器（RAM），CPU 或 I/O 设备在某一时刻可按地址去访问任一个存储单元，而且在一个存取周期内能进行一次访问，信息读出的时间对全部地址都是相等的，即与信息所在地址位置无关。RAM 既可以读出，又可写入信息。

只读存储器（ROM）是随机存储器的一种特殊方式。其特点是：每次访问它只能读出信息，而不能写入新的内容，因此用它存放那些固定不变的系统程序和子程序等。

4. 利用若干 2114 芯片和 2716 芯片采用 74LS138 译码，构成 7K 容量的存储器，要求有 5KRAM 及 2KROM，请设计电路图。

【解】2114 芯片为 1K×4 的存储芯片，所以每两片构成一个 1K 容量的存储器。5KRAM 需要 10 个 2114 芯片。2KROM 需要 1 个 2716 芯片。电路图如图 6-5 所示。其中 ROM 的地址范围是：80000H～807FFH，RAM 的地址范围是：80800H～81BFFH。

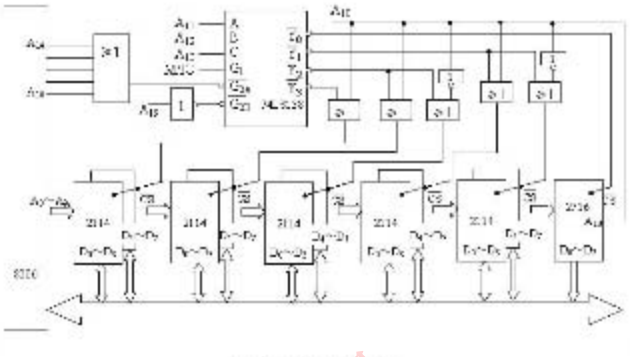


图 6-5　存储器连接示意图

5. 常见的存储芯片有哪些，它们的容量是多少？

【解】静态 RAM 芯片：Intel 2114，存储容量为 1K×4 位

动态 RAM 芯片：Intel 2188，存储容量为 16K×1 位

掩膜式只读存储器 ROM：8508，8316 芯片等。

一次性可编程的只读存储器 PROM：Intel 3036，存储容量为 2K×8 位

紫外线擦除的 EPROM：Intel 2716，存储容量为 2K×8 位

电可擦除只读存储器 EEPROM：AT24C01，存储容量为 128×8 位

第 7 章　输入/输出和中断

教材习题解答

1. CPU 与外设交换数据的方式有几种，各有什么特点？

【解】CPU 与外设交换数据的方式有两种：程序控制传递方式和 DMA（直接存储器存取）传递方式，其中程序控制的数据传送分为无条件传送、查询传送和中断传送三种。

程序控制传递方式的特点是：以 CPU 为中心，数据传送的控制来自 CPU，通过预先编制好的输入或输出程序（传送指令和 I/O 指令）实现数据的传送。这种传送方式的数据传送速度较低，传送路径经过 CPU 内部的寄存器，同时数据的输入输出的响应比较慢。

DMA（Direct Memory Access）传递方式是一种不需要 CPU 干预也不需要软件介入的高速数据传送方式。由于 CPU 只启动而不干预这一传送过程，同时整个传送过程只由硬件完成而不需要软件介入，所以其数据传送速率可以很高。

2. 比较中断传送和 DMA 传送的区别。

【解】在中断方式下，外设需要与主机传输数据时要请求主机给予中断服务，中断当前主程序的执行，自动转向对应的中断处理程序，控制数据的传输，过程始终在处理器所执行的指令控制之下进行。

DMA 传送方式下，系统中有一个 DMA 控制器，它是一个可驱动总线的主控部件。当外设与主存储器之间需要传输数据时，外设向 DMA 控制器发出 DMA 请求，DMA 控制器向中央处理器发出总线请求；取得总线控制权后，DMA 控制器按照总线时序控制外设与存储器间的数据传输，而不是通过指令来控制数据传输，传输速度高于中断方式。

3. 编写一段程序，使从终端上输入 80 个字符，存放 BUFFER 开始的内存缓冲区中；并要求每读入一个字符，在终端上显示键入的字符。

【解】程序如下：

```
DATA    SEGMENT
BUFFER  DB  80 DUP（？）
DATA    ENDS
CODE    SEGMENT
        ASSUME  CS：CODE，DS：DATA
START：  MOV   AX，DATA
        MOV   DS，AX
        MOV   CX，80
        MOV   BX，BUFFER
LP：     MOV   AH，1
        INT   21H
        MOV   [BX]，AL
```


1

```
        INC   BX
        LOOP  LP
        MOV   AH，4CH
        INT   21H
CODE    ENDS
        END   START
```

4. 什么叫中断，采用中断技术有哪些好处？

【解】中断是外设随机地（指主程序运行到任何一条指令时）或程序预先安排产生中断请求信号，暂停CPU正在运行的程序，转入执行称为中断服务的子程序，中断服务完毕后，返回到主程序被中断处继续执行的过程。

CPU与外设采用中断技术交换数据，可以克服无条件传递和查询传送的缺点，即CPU和外设只能串行工作，这样可以提高系统的工作效率，充分发挥CPU的高速运算的能力。

5. 什么叫中断系统，它具备哪些功能？

【解】计算机系统中完成中断处理过程的软硬件称为中断系统。

中断系统应具有以下功能：

(1) 能实现中断响应、中断服务和中断返回

当有中断源发出中断请求时，CPU能决定是否响应此中断，若接受这个中断请求，CPU能在保护断点（现行CS：IP值）后，转去执行相应的中断服务程序，中断处理完成后能恢复断点，继续执行程序。

(2) 能实现中断优先级排队

当两个或多个中断源同时提出中断请求时，CPU要能根据各中断请求的轻重缓急程度，分别处理，即给每个中断源一个优先级别，保证首先处理优先级高的中断申请。

(3) 能实现中断嵌套

若中断处理过程中，又有新的优先级较高的中断请求，且当中断允许时，CPU应能暂停正在执行的中断服务程序，转去响应与处理优先级较高的中断申请，待处理结束后，再返回原先较低的中断处理过程。

6. CPU在什么情况下才响应中断，中断处理过程一般包括哪些步骤？

【解】CPU每执行完一条指令后，当查询到有中断请求且此时CPU允许中断（即中断允许标志位IF=1，开中断），CPU才能响应中断。

CPU中断处理的具体过程一般包括以下步骤：中断请求、中断判优、中断响应、中断处理及中断返回。

7. 何谓非屏蔽中断和可屏蔽中断？

【解】何谓非屏蔽中断和可屏蔽中断都是硬件中断（也称外部中断）。非屏蔽中断通过CPU的NMI端引入，它不受内部中断允许标志位IF的屏蔽，一般在一个系统中只允许有一个非屏蔽中断。可屏蔽中断是通过CPU的INTR引入，它受内部中断允许标志IF的控制。只有在IF=1时，CPU才能响应中断源的请求。当IF=0时，中断请求被屏蔽。通常在一个系统中，通过中断控制器（8259A）的配合，可屏蔽中断可以有一个或多个。

8. CPU如何识别中断源？

【解】通过读取8259中断服务寄存器ISR中的内容识别中断源。ISR寄存所有正在被服务的

中断线，优先权电路对保存在 IRR 中的各个中断请求，经过判断确定最高的优先权，并在中断响应周期把它选通送至中断服务寄存器 ISR。

9. 什么叫中断优先权，有哪些解决中断优先权的办法？

【解】由于中断请求是随机的，在某一瞬间有可能出现两个或两个以上中断源同时提出请求的情况。这时必须根据中断源的轻重缓急，给每个中断源确定一个中断级别，这个级别称为中断优先权。在系统中如果有多个中断源，就要考虑其优先权的问题。通常 CPU 只有一条中断请求线，当有多个中断源同时请求中断服务时，就要求 CPU 能识别出哪些中断源有中断请求，同时辨别和比较它们的优先权，先响应中断源中优先权级别最高的中断请求。另外，当 CPU 正在处理中断时，也要能响应更高级的中断申请，并屏蔽同级或较低的中断请求。

确定中断的中断优先权一般可以采用软件和硬件两种方法。

10. 简述 8086/8088CPU 的中断结构及中断处理过程。

【解】8086/8088CPU 可以处理 256 种类型的中断源，这些中断源可分为硬件中断和软件中断两大类。CPU 响应中断后，把正在执行程序的当前地址（CS：IP）压入堆栈保存，将中断服务程序入口地址送入 CS：IP，由此开始执行中断服务程序。在 8086 系统中，允许引入 256 种类型中断源（类型码为 0—255），相应有 256 个中断服务程序首址，存放中断地址的一段内存空间称中断向量表。断类型码与中断向量所在位置（中断向量地址指针）之间的对应关系为：中断向量地址指针=4*中断类型码

内部中断的处理过程：

（1）CPU 取得中断类型码，将类型码乘 4 作为向量表指针。（2）把 CPU 标志寄存器入栈，保护各个标志。此操作类似于 PUSHF 指令。（3）清除 IF 和 TF 标志，屏蔽新的 INTR 中断和单步中断。（4）保存断点，即把断点处 IP 和 CS 压栈，先压 CS 值后压 IP 值。（5）从中断向量表中取中断服务程序入口地址分别送入 IP 和 CS 中。（6）按新的地址执行中断服务程序。

CPU 执行中断服务程序的最后一条中断返回指令 IRET 后，返回到断点处继续执行原程序。

11. 8086/8088 的中断系统分哪几类，其优先顺序如何？

【解】8086/8088CPU 可以处理 256 种类型的中断源，这些中断源可分为硬件中断和软件中断两大类。硬件中断是由外部硬件产生的，它又可分为非屏蔽中断和可屏蔽中断。一般在一个系统中只允许有一个非屏蔽中断，可屏蔽中断可以有一个或多个。软件中断是 CPU 根据软件的某些指令或者软件对标志寄存器某个标志位的设置而产生的。

8086/8088 系统中，中断优先权排队次序从高到低为：除法出错、指令中断、溢出中断、非屏蔽中断、可屏蔽中断、单步中断。

12. 中断入口地址表的功能是什么？已知中断类型码分别为 84H 和 0FAH，它们的中断入口在中断入口地址表的什么位置上？

【解】CPU 响应中断后，把正在执行程序的当前地址（CS：IP）压入堆栈保存，将中断服务程序入口地址送入 CS：IP，由此开始执行中断服务程序。在 8086 系统中，允许引入 256 种类型中断源（类型码为 0—255），相应有 256 个中断服务程序首址。这些地址就存放在中断入口地址表中。

中断类型码分别为 84H 的中断源所对应的中断服务程序首址存放在 0000：0210H（4×

3

84H＝210H）开始的4个单元中。

中断类型码分别为0FAH的中断源所对应的中断服务程序首址存放在0000：03E8H（4×0FAH＝3E8H）开始的4个单元中。

13. 若在一个系统中有5个中断源，它们的优先排序为：1，2，3，4，5。它们的中断服务程序入口地址分别为$LOOP_1$、$LOOP_2$、$LOOP_3$、$LOOP_4$、$LOOP_5$。试编写一中断服务程序，当有中断请求CPU响应时，能用软件查询办法转到优先权最高的中断源。

【解】设中断请求寄存器地址为20H。中断服务程序如下：

```
        PUSH  CS
        PUSH  IP            ；保护现场
        IN    AL，20H
        TEST  AL，80H
        JNE   LOOP1         ；执行1号中断服务程序
        TEST  AL，40H
        JNE   LOOP2
        TEST  AL，20H
        JNE   LOOP3
        TEST  AL，10H
        JNE   LOOP4
        TEST  AL，08H
        JNE   LOOP5
              ；
              ；
              ；
LOOP1：       ；
              ；
        IRET
LOOP2：       ；
              ；
        IRET
LOOP3：       ；
              ；
        IRET
LOOP4：       ；
              ；
        IRET
LOOP5：       ；
              ；
        IRET
```

14. 8086/8088CPU如何获得中断类型码？

4

【解】对于专用中断，中断类型码是自动形成的。几种类型码为：类型 0，1，3，4。

对于 INT n 指令，其类型码为指令中给定的 n。

对于外部非屏蔽中断请求，CPU 自动提供中断类型号 2。

对于外部可屏蔽中断请求，CPU 从数据线中获取外部设备的中断类型码。

15. 简述中断控制器 8259A 的内部结构和主要功能。

【解】8259A 的内部结构见教材图 7-11。它由中断请求寄存器 IRR（INTERRUPT Request Register）、优先权电路、中断服务寄存器 ISR（IN—Service Register）、中断屏蔽寄存器 IMR（Interrupt Mask Register）、数据总线缓冲器、读写电路、控制逻辑和级连缓冲/比较器组成。

8259A 的主要功能为：

(1) 具有 8 级优先权控制，通过级连可扩展至 64 级优先权控制。

(2) 每一级中断都可以屏蔽或允许。

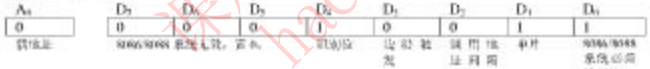
(3) 在中断响应周期，8259A 可提供相应的中断向量号（中断类型号）。

(4) 可通过编程来进行选择工作方式。

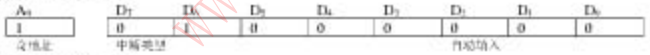
16. 某系统中有 5 个中断源，它们从中断控制器 8259A 的 IR_4～IR_0 中以脉冲方式引进系统，它们的中断类型码分别为 40H，41H，42H，43H 和 44H，中断入口分别为 3500H，4080H，4505H，5540H 和 6000H。允许它们以完全嵌套方式工作，请编写相应的初始化程序，使 CPU 响应任何一级中断时，能正确地进入各自中断服务程序入口。

【解】设在 I/O 地址空间中分配给 8259A 的端口地址为 20H 和 21H。中断结束为 EOI 命令方式。边沿触发方式，缓冲方式。

ICW_1 的内容：13H



ICW_2 的内容：40H



ICW_4 的内容：0DH

根据系统要求初始化编程如下：

```
MOV  AL，13H        ；设置 ICW1 为边沿触发，单片 8259A，需要 ICW4。
OUT  20H，AL
MOV  AL，40H        ；设置 ICW2，类型码为 40H（IR0）。
OUT  21H，AL
MOV  AL，0DH        ；设置 ICW4，全嵌套方式，缓冲方式，正常 EIO
OUT  21H，AL
```

初始化完成后，8259A 处于全嵌套工作方式，可以响应外部中断请求。根据操作要求，设置中断入口地址分别为 3500H，4080H，4505H，5540H 和 6000H：

5

17. 试编写出只有一片 8259A 的 8086 系统中 8259A 的初始化程序。8259A 的地址为 02C0H 和 02C1H，要求：（1）中断请求采用电平触发。（2）IRQ 请求的中断类型是 16。（3）采用缓冲器方式。（4）采用普通的 EOI 命令。

【解】

ICW_1的内容：1BH

A_0	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
0	0	0	0	1	1	0	1	1
设地址	8086/8088 系统无效，置 0。			切换位	电平触发	调用地址间隔为 8	单片	8086/8088 系统必须设 ICW_4。

6

ICW_1的内容：10H

A_0	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	1	0	0	0	0
写端口	中断类型				自动填入			

ICW_4的内容：0DH

A_0	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	0	1	1	0	1
写端口	未定义，置 0。			全嵌套方式	缓冲方式/正		非自动EOI	下任于 8086/8088
							结束	系统

根据系统要求初始化编程如下：

```
MOV  AL，18H          ；设置 ICW1为电平触发，单片 8259A，需要 ICW4。
MOV  DX，02C0H
OUT  DX，AL
MOV  AL，10H          ；设置 ICW2，类型码为 10H（IR0 ）。
MOV  DX，02C1H
OUT  DX，AL
MOV  AL，0DH          ；设置 ICW4，全嵌套方式，缓冲方式，正常 EIO
OUT  DX，AL
```


7

第 8 章　接口技术

教材习题解答

1．填空题

(1) 8255A 内部包括两组控制电路，其中 A 组控制_____的工作方式和读写操作，B 组控制_____的工作方式和读写操作。

(2) 8255A 的端口 A 工作于方式 2 时，使用端口 C 的_____位作为与 CPU 的外部设备的联络信号。

(3) 当 8255A 的端口 A 和端口 B 均工作于方式 1 输出时，端口 C 的 PC_6 和 PC_2 可以作为_____使用。

(4) 8255A 的端口 A 工作于方式 2 时，端口 B 可以工作于_____。

(5) 8255A 中，可以按位进行置位/复位的端口是_____，其置位/复位操作是通过向_____口地址写入_____实现的。

(6) 8251A 工作在同步方式时，最大波特率为_____；工作在异步方式时，最大波特率为_____。

(7) 8251A 工作在异步方式时，每个字符的数据位长度可以是_____，停止位长度可以是_____。

(8) 8251A 从串行输入线接收到一个字符后，将信号_____置为有效。

(9) 8251A 工作在同步方式时，同步检测引脚 SYNDET 可以作为输入或者输出信号使用。若工作在外同步方式，该引脚作为_____；若工作在内同步方式，该引脚作为_____。

(10)8253 工作于方式 0 下，控制信号 GATE 变成低电平后，对计数器的影响是_____。

【解】

(1) 端口 A、端口 C 高 4 位（PC_7~PC_4）
端口 B，端口 C 低 4 位（PC_3~PC_0）

(2) PC_3~PC_7

(3) $\overline{STB}_A$（端口 A 的选通输入信号）、IBF_A（输入缓冲器满信号）

(4) 方式 0 或方式 1

(5) 端口 C_7；控制；控制字

(6) 64 Kbit/s ；19.2 Kbit/s

(7) 5、6、7、或 8 位 ；1 个、1.5 或 2 位

(8) R_XRDY

(9) 输入或；输出或

(10)计数暂停

2．已知 8255A 的端口 A、B、C 和控制口地址分别为 120H、122H、124H 和 126H，试按下列要求设计初始化程序：

(1) 将端口 A 和端口 B 设置成方式 0，端口 A、端口 C 作为输出口，端口 B 作为输入口。

(2) 将端口 A 设置成方式 2，端口 B 设置成方式 1，端口 B 作为输出口。

(3) 将端口 A 和端口 B 均设置成方式 1 的输入状态，且 PC6、PC7 设置成输出状态。

【解】

(1) 方式选择控制字的内容：82H

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	0	0	1	0
控制字特征	端口 A 设置成方式 0		端口 A 为输出口	端口 C 高 4 位作为输出口	端口 B 设置成方式 0	端口 B 为输入口	端口 C 低 4 位作为输出口

初始化程序：

```
MOV AL，82H        ；方式选择控制字送 AL
OUT 126H，AL       ；方式选择控制字输出给 8255A 控制端口
```

(2) 方式选择控制字的内容：0C4H

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	1	×	×	×	1	0	×
控制字特征	端口 A 设置成方式 2				端口 B 设置成方式 1	端口 B 为输出口	

初始化程序：

```
MOV AL，0C4H       ；方式选择控制字送 AL
OUT 126H，AL       ；方式选择控制字输出给 8255A 控制端口
```

(3) 方式选择控制字的内容：0B6H

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	1	1	0	1	1	×
控制字特征	端口 A 设置成方式 1		端口 A 为输入口	端口 C 高 4 位作为输出口	端口 B 设置成方式 1	端口 B 为输入口	

初始化程序：

```
MOV AL，0B6H       ；方式选择控制字送 AL
OUT 126H，AL       ；方式选择控制字输出给 8255A 控制端口
```

3. 某一外部输入设备，当它准备好一个数据时，那个发出一个数据准备好的状态信号 READY（高电平有效）。当 CPU 把数据取走后，要求 CPU 通过 $\overline{ACK}$ 线向外设发一负脉冲，以便外设清除 READY 信号。试用 8255A 作为接口芯片，分别用查询和中断方式从外设读入 100 个数据，将其存入从 DAT_BEG 开始的内存区。要求：画出 8255A 与外设之间的连线，并进行编程。

【解】

(1) 用查询方式从外设读入数据时，8255A 与外设之间的连线如图 8-30 所示，使用端口 C 传送控制和状态信息。

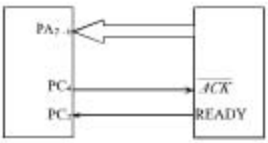


图 8-30　8255A 与外设之间的连线

方式选择控制字的内容：91H

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	1	0	×	×	1
控制字标志	端口 A 设置为方式 0		端口 A 为输入口	端口 C 的一4 位为输出口	端口 B	端口 B	端口 C 的 3—0 位为输入口

设在系统中 8255A 的端口地址为：

端口 A：00E0H　　端口 B：00E2H　　端口 C：00E4H　　控制口：00E6H

读入数据的程序：

```
DATA    SEGMENT                    ；数据段开始
DAT_BEG  DB   100 DUP(? )
DATA    ENDS                       ；数据段结束
CODE    SEGMENT                    ；代码段开始
        ASSUME  CS：CODE，DS：DATA
START：  MOV  AX，DATA
         MOV  DS，AX               ；装填数据段
         LEA  BX，DAT_BEG          ；代表变量的地址
         MOV  AL，91H              ；方式控制字送 AL
         OUT  0E6H，AL             ；方式控制字写入 8255A 控制口
         MOV  AL，09H              ；用置位方式使 PC4=1
         OUT  0E6H，AL             ；使 ACK 为高
         MOV  CX，9
RDLP：   IN   AL，0E4H             ；读入端口 C 的内容（READY 信号）
         AND  AL，04H
         JZ   RDLP                 ；未准备好，则等待
         IN   AL，0E0H             ；已准备好，则从端口 A 输入数据
         MOV  [BX]，AL             ；将输入数据保存到内存中
         INC  BX
         MOV  AL，08H              ；用置位方式使 PC4=0
         OUT  0E6H，AL             ；使 ACK 为低
         CALL  DELAY               ；  延定时程序，形成负脉冲
         MOV  AL，09H              ；用置位方式使 PC4=1
         OUT  0E6H，AL             ；使 ACK 为高
         LOOP  RDLP
         MOV  AH，4CH
         INT  21H                  ；返回 DOS
DELAY    PROC                      ；定义延时子程序
         MOV   CX，100             ；向 CX 中送延时常数，决定延时的时间
DELAY2：NOP
         LOOP   DELAY2
         RET
```

```
DELAY   ENDP                    ；定义延时子程序结束
CODE    ENDS                    ；代码段结束
        END   START             ；源程序结束
```

(2) 用中断方式从外设读入数据时，8255A 与外设之间的连线如图 8-31 所示，使用端口 C 传送控制和状态信息。将 PC_3 连到 8259A 的中断请求信号输入端 IR_3，其对应的中断类型号为 0BH，由于 0BH×4=002CH，所以应该将中断向量写入 0000：002CH 开始的 4 个单元中。假设 8259A 在系统程序中已经完成初始化。

方式选择控制字的内容：0B0H

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	1	1	×	×	×	×
控制字标志	端口 A 选择方式 1		端口 A 为输入口	端口 C	端口 B	端口 B	端口 C

设在系统中 8255A 的端口地址为：

端口 A：00E0H　　端口 B：00E2H　　端口 C：00E4H　　控制口：00E6H

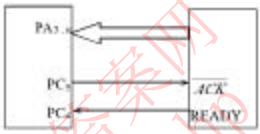


图 8-31　8255A 与外设之间的连线

读入数据的程序：

```
DATA    SEGMENT                 ；数据段开始
DAT_BEG DB   100 DUP(? )
DATA    ENDS                    ；数据段结束
CODE    SEGMENT                 ；代码段开始
        ASSUME  CS：CODE，DS：DATA
START：  MOV   AX，DATA
         MOV   DS，AX           ；装填数据段
         MOV   AL，0B0H         ；方式控制字送 AL
         MOV   AL，0BH          ；用置位方式使 PC3=1
         OUT   0E6H，AL         ；使 ACK 为高
         MOV   AX，0
         MOV   ES，AX           ；ES 为 0，指向中断向量表
         MOV   AX，0100H        ；取中断服务子程序入口地址偏移量
         MOV   ES：[002CH]，AX
         MOV   AX，3200H        ；取中断服务子程序入口地址段值
         MOV   ES：[002EH]，AX
         MOV   AL，0DH
         OUT   0E6H，AL         ；置 8255A 之 INTEA=1，允许端口 A 中断
```

```
        STI                     ；80186CPU 开放中断
        ┆
```

中断服务子程序如下：

```
        PUSH    AX
        PUSH    BX
        LEA     BX，DAT_BEG     ；存送发送的地址
        IN      AL，0E0H        ；已准备好，则从端口 A 输入数据
        MOV     [BX]，AL        ；将输入数据保存到内存中
        INC     BX
        MOV     AL，0AH         ；用置位方式使 PC5=0
        OUT     0E6H，AL        ；使 ACK 为低
DELAY:  MOV     CX，100         ；  向 CX 中送延时常数；一段延时程序，形成负脉冲
DELAY1: LOOP    DELAY1
        MOV     AL，0BH         ；用置位方式使 PC5=1
        OUT     0E6H，AL        ；使 ACK 为高
        POP     BX
        MOV     AL，20H         ；向 8259A 发 EOI
        OUT     20H，AL
        POP     AX
        STI
        IRET                    ；中断返回
```

4．试用 8255A 和 8253 芯片配合，设计一个交通信号自动控制系统。设计要求如下：

(1) 在某一南北方向和东西方向的十字路口，每个方向都有红、黄、绿三色信号灯，如图 8-79 所示。要求信号灯按下列规则变化：

① 首先使某一方向绿灯亮，另一方向红灯亮。当出现下述情况之一时，开始变灯：其一是红灯已亮 30s；其二是红灯路口停留的车辆已达 5 辆。

② 变灯遵循的规律：首先使绿灯闪烁 5s（注：0.5s 灭，0.5s 亮，重复 5 次），接着绿灯灭，黄灯亮，黄灯亮 5s 后，使得黄灯灭，红灯亮，紧接着使另一方向的红灯变为绿灯。在一个方向由绿灯变红灯的过程中，另一方向的红灯保持不变。

(2) 画出硬件框图，编写控制程序。

提示：用 8253 来实现定时和检测车辆。用 8255A 某一端口的 6 根线控制交通灯。对于检测车辆，假设另有传感器电路配合，此传感器电路每当有一辆车时会发出一个脉冲信号。

【解】用 8255 的端口 A 的 6 根线控制交通灯：（输出 1 时灯亮）

PA_7	PA_6	PA_5	PA_4	PA_3	PA_2	PA_1	PA_0
未用	未用	东西红灯	东西黄灯	东西绿灯	南北红灯	南北黄灯	南北绿灯

设 8253 的端口地址分别为：

0D0H（端口 A），0D2H（端口 B），0D4H（端口 C），0D6H（控制口）。

设 8255 的端口地址分别为：

1E0H（通道 0），1E 2H（通道 1），1E 4H（通道 2），1E6H（控制口）。

硬件框图如图 8-32 所示。

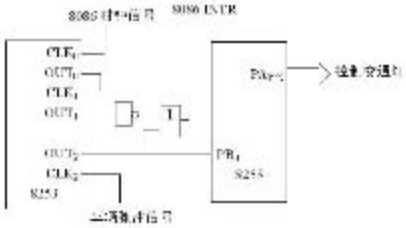


图 8-32　交通灯控制硬件框图

8253 方式选择控制字的内容：82H

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	×	0	1	×
控制字标志	端口 A 被置成方式 0		端口 A 为输出口	端口 C	端口 B 被置成方式 0		端口 B 为输入口	端口 C

8255 控制字的内容：34H

计数初值：0FFFFH（65535），使用 8086 的时钟信号（频率：6MHz ）

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
0	0	1	1	×	1	0	0
计数第 0		先写低 8 位		方式 2			二进制

8255 控制字的内容：54H，计数初值：46

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
0	1	0	1	×	1	0	0
计数第 1		只取写低 8 位		方式 2			二进制

8255 控制字的内容：90H，计数初值：05H

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	1	0	0	0	0
计数第 2		只取写低 8 位		方式 0			二进制

在程序中使用如下寄存器作为标志量：

BL：最低位为 0 时，表示当前状态是“东西方向红灯亮”

　　最低位为 1 时，表示当前状态是“南北方向红灯亮”

BH：红灯计时单元，初值为 0，每 0.5 秒加 1，值为 60 时，说明某一红灯已亮 30 秒。

DL：绿灯闪烁计数单元，初值为 0，每 0.5 秒加 1，值为 10 时，说明某一绿灯已闪烁 5 秒。

DH：黄灯计时单元，初值为 0，每 0.5 秒加 1，值为 10 时，说明某一黄灯已亮 5 秒。

CH：绿灯闪烁时标志当前绿灯的状态，最低位为 0 时，表示当前状态是“灭”。

中断服务程序流程图如图 8-33 所示，初始化程序段如下：

```
MOV  AL，82H          ；初始化 8255
OUT  0D6H，AL
MOV  AL，34H          ；初始化 8253
OUT  1E6H，AL
```

```
        MOV  AL，54H
        OUT  1E6H，AL
        MOV  AL，90H
        OUT  1E6H，AL
        MOV  AL，FFH          ；8253 通道 0 计数初值
        OUT  1E0H，AL
        OUT  1E0H，AL
        MOV  AL，46           ；8253 通道 1 计数初值
        OUT  1E2H，AL
        MOV  AL，05H          ；8253 通道 2 计数初值
        OUT  1E4H，AL
        XOR  DX，DX           ；清各标志单元
        XOR  BX，BX
        MOV  AL，21H
        OUT  0D0H，AL         ；设初始状态：东西方向红灯亮，南北方向绿灯亮
        MOV  CH，01H          ；初始时，某一绿灯亮
        STI                   ；开中断
        ……  ……
```

中断服务程序如下：

```
        IN   AL，0D2H
        AND  AL，01H
        JNZ  CHANGE           ；不为 0，说明车辆计数≥5，转移到变灯处理
        INC  BH
        CMP  BH，60
        JAE  CHANGE           ；≥60，转移到变灯处理
        AND  BL，01H
        JNZ  CHANG1           ；BL0=1，当前状态：南北红灯亮
        MOV  AL，21H
        OUT  0D0H，AL         ；东西红灯亮，不变灯
        JMP  RETURN           ；中断返回
CHANG1：MOV  AL，0CH
        OUT  0D0H，AL         ；南北红灯亮，不变灯
        JMP  RETURN           ；中断返回
CHANGE：INC  DL               ；变灯处理开始
        CMP  DL，10
        JA   YELLO            ；绿灯已闪烁 5 秒，转移到变黄灯处理
        AND  BL，01H
        JNZ  GREEN1           ；BL0=1，当前状态：南北红灯亮，转移
        AND  CH，01H
```

```
        JNZ   GREEN2          ；CH4=1，当前状态：绿灯亮
        MOV   AL，21H
        OUT   0D0H，AL        ；南北绿灯亮，东西红灯亮（不变）
        MOV   CH，01H         ；改变绿灯亮的标志
        JMP   RETURN          ；中断返回
GREEN2：MOV   AL，20H
        OUT   0D0H，AL        ；南北绿灯灭，东西红灯亮（不变）
        MOV   CH，00H         ；改变绿灯亮的标志
        JMP   RETURN          ；中断返回
GREEN1：AND   CH，01H
        JNZ   GREEN3          ；CH4=1，当前状态：绿灯亮
```

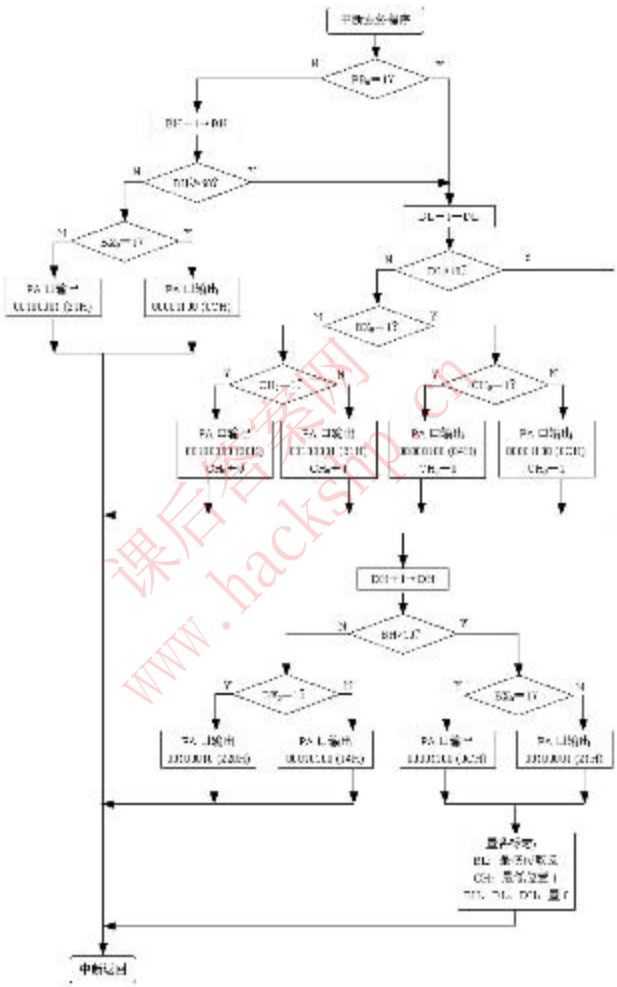


图 8-33　交通灯控制中断服务程序流程图

```
        MOV  AL，0CH
        OUT  0D0H，AL          ；东西绿灯亮，南北红灯亮（不变）
        MOV  CH，01H           ；改变绿灯亮的标志
        JMP  RETURN            ；中断返回
GREEN3：MOV  AL，04H
        OUT  0D0H，AL          ；东西绿灯灭，南北红灯亮（不变）
        MOV  CH，00H           ；改变绿灯亮的标志
        JMP  RETURN            ；中断返回
YELLO:  INC  DH                ；黄灯亮处理开始
        CMP  DH，10
        JA   YELLO1            ；黄灯已亮5秒，转移到变红灯处理
        AND  BL，01H
        JNZ  YELLO2            ；BL0=1，当前状态：南北红灯亮，转移
        MOV  AL，22H
        OUT  0D0H，AL          ；东西红灯亮，南北黄灯亮
        JMP  RETURN            ；中断返回
YELLO2：MOV  AL，14H
        OUT  0D0H，AL          ；南北红灯亮，东西黄灯亮
        JMP  RETURN            ；中断返回
YELLO1：AND  BL，01H
        JNZ  RED1              ；原状态：南北红灯亮
        MOV  AL，0CH
        OUT  0D0H，AL          ；南北红灯亮，东西绿灯亮
        JMP  RED
RED1:   MOV  AL，21H
        OUT  0D0H，AL          ；东西红灯亮，南北绿灯亮
RED:    ADD  BL，01H           ；南北标志取反
        XOR  DX，DX
        XOR  BH，BH
        MOV  CH，01H
RETURN：STI
        IRET                   ；中断返回
```

5. 图 8-80（见教材）是一个检测开关状态并控制相应的继电器通断的电路。要求当开关 S_0～S_7之一闭合时，使相应的继电器 K_0～K_7之一吸合（即让驱动电流流过继电器线圈）；若开关处于断开状态，则使相应的继电器释放。系统每隔 20ms 应检测一遍开关状态，并对继电器做相应控制。图中 8255A 的 4 个端口地址分别为 2C0H，2C2H，2C4H 和 2C6H。试完成：

(1) 8255A 的初始化编程（初始状态所有继电器的线圈均无电流通过）。

(2) 设系统中具有一个 2MHz 的时钟信号源，另由一片 8253 来实现 20ms 的定时，每当 20ms

判时自动向 CPU 申请中断。编写中断服务程序，并在其中完成开关的检测和继电器的控制。

【解】

(1) 8255A 的初始化编程：

8255A 的方式选择控制字的内容：82H

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	×	0	1	×
控制字标志	端口 A 组方式 0		端口 A 为输出口	端口 C	端口 B 组方式 0	端口 B 为输入口	端口 C

初始化程序：

```
MOV  AL，82H     ；方式选择控制字送 AL
OUT  2C6H，AL    ；方式选择控制字输出给 8255A 控制端口
MOV  AL，00H     ；
OUT  2C0H，AL    ；  8255A PA 端口输出 0，使所有继电器的线圈均无电流通过
```

(2) 开关 S_0~S_7 之一闭合时，相应的 PB_0~PB_7 的输入为 1；PA_0~PA_7 输出高电平（1）时，可以使相应的继电器 K_0~K_7 吸合；

中断服务程序：

```
PUSH   AX
IN     AL，2C2H        ；从端口 B 输入数据
OUT    2C0H，AL        ；从端口 A 输出数据
POP    AX
STI
IRET                   ；中断返回
```

6. 设 8086 系统中有一片 8251A 芯片，其端口地址分别为 130H 和 132H，请按以下要求分别编出 8251A 的初始化程序：

(1) 令双工异步方式通信，波特率系数为 16，每个字符数据为 7 位，偶校验，1.5 个停止位，传送过程错误不复位，且不使用调制解调器。

(2) 令双工同步方式通信，每个字符数据为 8 位，不带校验，内同步，两个同步字符分别为 EFH 和 FEH。

【解】

(1) 8251A 的方式选择控制字的内容：0BAH

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	1	1	1	0	1	0
1.5 个停止位		偶校验		每个字符数据为 7 位		波特率系数为 16	

初始化程序：

```
MOV  AL，0BAH    ；方式选择控制字送 AL
OUT  132H，AL    ；方式选择控制字输出给 8255A 控制端口
```

(2) 8251A 的方式选择控制字的内容：0CH

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
0	0	×	0	1	1	0	0
内同步，两个同步字符		无校验		每个字符数据为 8 位		同步方式	

初始化程序：

```
MOV  AL，40H
OUT  132H，AL    ；复位 8251A
MOV  AL，0CH     ；方式选择控制字送 AL
OUT  132H，AL    ；方式选择控制字输出给 8255A 控制端口
MOV  AL，0EFH
OUT  132H，AL    ；写入第一个同步字符
MOV  AL，0FEH
OUT  132H，AL    ；写入第二个同步字符
```

7. 两台计算机均利用 8251A 芯片进行串行通信，通信规则为：一方进行发送，另一方进行接收的半双工异步通信方式，波特率系数取 16，每个字符传送 7 位，奇校验，2 个停止位。试用查询方式实现串行通信，设计出逻辑电路图，并编写通信程序。

【解】

【解】逻辑电路图如图 8-34 所示，设 8251A 的口地址：1A0H、1A2H。

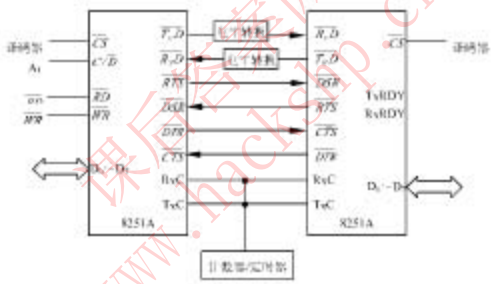


图 8-34　两台计算机串行通信的逻辑电路图

8251A 的方式选择控制字的内容：0DAH

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	1	0	1	1	0	1	0
2 个停止位	奇校验			每个字符数据为 7 位		波特率系数为 16	

8251A 的操作命令控制字（内部复位）的内容：40H

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
0	1	0	0	0	0	0	0

使 8251A 返回到接受方式选择控制字状态

8251A 的操作命令控制字（允许发送）的内容：37H

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
0	0	1	1	0	1	1	1
		请求发送	令错误标志复位	不给下传	接收允许	数据存储准备好	允许发送

(1) 发送程序

```
MOV  DX，1A2H
MOV  AL，00
OUT  DX，AL          ；复位→方式命令→工作命令
```

```
        MOV  AL，40H
        OUT  DX，AL                ；内部复位（D6=1）
        NOP
        MOV  AL，0DAH
        OUT  DX，AL                ；方式选择控制字输出给 8251A 控制端口
        MOV  AL，37H
        OUT  DX，AL                ；工作命令：D0=1 允许发送
        MOV  CX，2DH               ；发送字节数
        MOV  SI，300H              ；发送首址
L1：    MOV  DX，1A2H
        IN   AL，DX
        AND  AL，01H               ；D0=1：发送准备好
        JZ   L1                    ；发送状态（TXRDY）未准备好
        MOV  DX，1A0H
        MOV  AL，[SI]
        OUT  DX，AL                ；发送数据
        INC  SI
        LOOP L1
        MOV  AX，4C00H
        INT  21H
```

8251A 的操作命令控制字（允许接收）的内容：14H

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
0	0	0	1	0	1	0	0
			错误标志复位	不空下作	接收允许		

(2) 接收程序

```
        MOV  DX，1A2H              ；复位→方式命令→工作命令
        MOV  AL，00
        OUT  DX，AL
        MOV  AL，40H
        OUT  DX，AL                ；内部复位（D6=1）
        NOP
        MOV  AL，0DAH
        OUT  DX，AL                ；方式命令
        MOV  AL，14H               ；D2=1 允许接收，D4=1 清错误标志
        OUT  DX，AL                ；工作命令
        MOV  CX，2DH               ；接收字节数
        MOV  DI，400H              ；接收首址
L2：    MOV  DX，1A2H
        IN   AL，DX
        AND  AL，02H               ；D1 接收准备好
```

```
        JZ      L2                  ；接收未准备好转
        MOV     DX，1A0H
        IN      AL，DX              ；取数据
        MOV     [DI]，AL
        INC     DI
        LOOP    L2
STOP:   MOV     AX，4C00H
        INT     21H
```

8．已知某系统中 8253 的口地址为 1E0H，1E2H，1E4H，1E6H，系统时钟为 2MHz，试编写 8253 的初始化程序，使其通道 0 产生周期为 1ms 的方波输出。

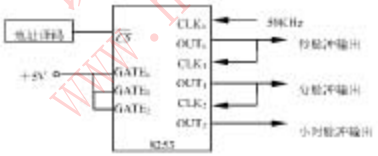
【解】通道 0 工作于方式 3（方波发生器），计数值为 2000（07D0H）

初始化程序如下：

```
MOV     AL，36H             ；控制字送 AL
OUT     1E6H，AL            ；向控制口写入控制字
MOV     AL，0D0H            ；低 8 位计数值是 0D0H
OUT     1E0H，AL            ；向通道 0 写入计数初值的低 8 位
MOV     AL，07H             ；高 8 位计数值为 07H
OUT     1E0H，AL            ；向通道 0 写入计数初值的低 8 位
```

9．已知某时钟信号源频率为 50KHz，试利用 8253 设计一个实时钟系统。试画出硬件电路，并编程序。

【解】硬件电路如图所示：



分别利用 CLK_0、CLK_1、CLK_2 输出秒，分钟和小时信号脉冲。三个计数器均工作在方式 2 下，计数器 0 的计数初值：50000（0C350H），计数器 1 的计数初值：60（3CH），计数器 0 的计数初值：60（3CH），控制字如下：

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
0	0	1	1	×	1	0	0
计数器 0		读写 16 位		方式 2			二进制

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
0	1	0	1	×	1	0	0
计数器 1		只读写低 8 位		方式 2			二进制

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	1	×	1	0	0
计数器 2		只读写低 8 位		方式 2			二进制

设 8253 的端口地址分配是：通道 0 为 120H，通道 1 为 122H，通道 2 为 124H，控制端口为 126H。初始化程序如下：

```
MOV   AL，34H          ；控制字送 AL
MOV   DX，126H         ；控制口地址送 DX
OUT   DX，AL           ；向控制口写入控制字
MOV   AL，50H          ；低 8 位计数值是 50H
MOV   DX，120H         ；通道 0 端口地址送 DX
OUT   DX，AL           ；向通道 0 写入计数初值的低 8 位
MOV   AL，0C3H         ；高 8 位计数值为 0C3H
OUT   DX，AL           ；向通道 0 写入计数初值的高 8 位
MOV   AL，54H          ；控制字送 AL
MOV   DX，126H         ；控制口地址送 DX
OUT   DX，AL           ；向控制口写入控制字
MOV   AL，3CH          ；低 8 位计数值是 3CH
MOV   DX，122H         ；通道 1 端口地址送 DX
OUT   DX，AL           ；向通道 0 写入计数初值的低 8 位
MOV   AL，94H          ；控制字送 AL
MOV   DX，126H         ；控制口地址送 DX
OUT   DX，AL           ；向控制口写入控制字
MOV   AL，3CH          ；低 8 位计数值是 3CH
MOV   DX，124H         ；通道 1 端口地址送 DX
OUT   DX，AL           ；向通道 0 写入计数初值的低 8 位
```

10. 试利用 DAC0832 芯片产生三角波信号输出，要求三角波的上下限为 0～+3V。试画出硬件电路图，并设计 D/A 转换程序。

【解】DAC0832 芯片输出+3V 时对应的输入为：

$255 \times 3/5 = 153 = 99H$

D/A 转换程序：

```
        MOV   DX，DAPORT    ；D/A 端口地址送入 DX
        MOV   AL，0         ；初值送 AL 寄存器
STAR:   MOV   CX，99H       ；循环变量
LP1:    INC   AL
        OUT   DX，AL        ；向 D/A 转换器输出数据
        LOOP  LP1           ；三角波的上升沿
        MOV   CX，99H       ；循环变量
LP2:    DEC   AL
        OUT   DX，AL        ；向 D/A 转换器输出数据
        LOOP  LP2           ；三角波的下降沿
        JMP   STAR
```

11. 试利用 8255A，ADC0809 设计一个数据采集系统，要求对于 8 个 0～5V 范围内的模拟信号进行采样，可以采用查询方式工作。试进行硬件电路设计，并编写 8255 初始化程序，ADC0809 实现 A/D 转换的工作程序。

【解】

设 8255 的四个端口地址为 0E0H、0E2H、0E4H、0E6H。ADC0809 的端口地址为 0A0H，对 8 路模拟量轮流采集，共结果存入地址为 2500H 开始的内存缓冲区。

8255 端口 A 和端口 B 设为方式 0，输入口，方式选择控制字的内容：92H

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	1	×	0	1	×
控制字标志	端口 A 被置成方式 0		端口 A 为输入口	端口 C	端口 B 被置成方式 0	端口 B 为输入口	端口 C

8255 初始化程序如下：

```
MOV  AL，92H      ；方式选择控制字送 AL
OUT  0E6H，AL     ；方式选择控制字输出给 8255A 控制端口
```

ADC0809 实现 A/D 转换的工作程序：

```
START:  MOV   BX，  2500H        ；设定输入缓冲区指针
        MOV   CX，8              ；设定 A/D 转换次数
        MOV   AH，0
LP1:    MOV   AL，AH
        OUT   0A0H，AL           ；启动一路模拟量转换
        MOV   DX，0E2H
WAIT:   IN    AL，DX             ；查询转换是否结束
        CMP   AL，80H
        JNE   WAIT
        IN    AL，0A0H
        MOV   [BX]，AL           ；存入内存一个字节
        INC   BX
        INC   AH                 ；改变选择模拟量的编码
        LOOP  LP1                ；若不够 8 次，返回重新启动
                                 ；若完则此已完成一遍采集
```

第9章　微机总线技术

教材习题解答

1. 什么是总线标准和接口标准？二者有何差异？

【解】所谓总线标准是指国际组织正式公布或推荐的应用各种不同模块组成各种计算机系统时必须遵守的规范。即计算机系统各模块之间通过微机总线连接和传输信息时，必须遵守的协议。总线标准一般从硬件和软件两个方面予以规范。

接口标准指的是外设接口的规范和定义。其中涉及到外设接口信号线的定义及排序，传输速率，传输方向，电器和机械特性等方面内容。不同外设有不同的接口标准，只有符合该标准的外设，才能使用该接口标准。

二者的差异：

(1) 微机总线具有公用性的特点，在微机系统内不同类型的功能模块均可并接到总线上，并分时复用总线共享总线资源。而接口标准具有专用性的特点，一般情况下一种接口只连接一类设备。

(2) 总线信号传输形式一般采用并行传输方式，而接口标准的信号传输形式既有并行传输又有串行传输。

(3) 接口标准一般实施在微机箱外以插头或插座形式提供用户使用。总线的信号线一般在微机箱内部。

2. 微机总线和接口标准可分为哪几类？

【解】按照总线所处的位置可将微机总线分为五种类型：内部总线，系统总线，元件级总线，局部总线，通信总线。

接口标准依其所连接设备的性质和功能分为以下五类：串/并行接口标准，外存接口标准，视频显示接口标准，测试仪器接口标准，新型的通用外设接口标准。

3. 微机总线的规范是什么？评价一种总线的性能有哪几种指标？

【解】微机总线的规范一般包括以下内容：

(1) 信号功能规范。包括信号线分类，数据传送方式及位数，控制信号的功能等。

(2) 电气规范。包括驱动类型，每槽输入负载，供电电源，直流特性，交流特性等。

(3) 定时规范。包括各种操作的时间参数。

(4) 机械规范。包括插槽布置，插件板尺寸等。

评价一种总线的性能一般包括以下几种指标：

(1) 数据传输性能。数据传输始终是总线的基本功能，比较重要的性能参数包括传输类型（同步传输还是异步传输），传输速率（兆字节/秒），数据线位数，地址线宽度等。

(2) 中断处理功能。几乎每种总线都支持中断功能，其关键参数是中断线数量，直接中断能力，中断类型等。

(3) 多重主控功能。相当一部分总线支持多重主控。当多个主控设备同时请求总线进行

信息传输时，需要总线仲裁来确定下一个使用总线的主控设备。其重要指标是总线仲裁系统构成、仲裁时间和仲裁算法等。

(4) 其它实用功能。包括电源故障处理、总线错误检测、备用电池能力等。

4. ISA 总线有哪几类总线组成？各有多少条？

【解】地址线 SA0S-A19（共 20 条）；数据线 SD0-SD7（共 8 条）；控制线（共 21 条）；状态线（共 2 条）；辅助线和电源线（共 11 条）。

5. 什么是 PCI 设备的配置空间？起什么作用？

【解】为了实现自动配置，PCI 总线规范要求每个 PCI 设备都要有 256 个字节的空间来存放设备说明等配置信息，这个空间称为配置空间。配置空间分成预定义区和设备关联区。预定义区占 64 个字节，对这个区域，各 PCI 设备必须服从统一的数据结构，共余 192 字节由设备规定。

配置空间是 PCI 局部总线和其它总线的最大区别之一，它用于扩展卡和器件的自动配置。当 PCI 扩展卡插入系统时，系统能根据配置空间的信息自动为扩展卡及设备分配存储地址、端口地址和中断线等。

6. 试比较 PCI 总线和 AGP 总线？

【解】PCI 局部总线是一种高性能、32 位或 64 位地址数据线复用的总线，它的用途是在高度集成的外设控制器器件、扩展板和处理器/存储器系统之间提供一种内部连接机制。

AGP 加速图形端口(accelerated graphics port)是由 AGP 执行者论坛与 1996 年 8 月提出的一种新型视频接口标准，它是专门为 3D 加速而设置的加速图形接口，其技术支持 3D 图形数据跨过了 PCI 总线，把主存和显存连接起来，解决了 PCI 总线设计中对于超高速系统的瓶颈问题。

采用 AGP 的目的是为了解决 PCI 总线形成的系统瓶颈，而不是用 AGP 去取代 PCI。它既具有 PCI 的一些特性又有一些性能超过 PCI。

7. 为什么要使用 AGP 图形加速端口？AGP 总线采用哪些先进技术？

【解】采用 AGP 的目的是为了解决 PCI 总线设计中对于超高速系统的瓶颈问题。它是对 PCI 总线的增强和扩充，实现高性能的 3D 图像处理。

AGP 总线采用的先进技术包括：

(1) 采用双泵（双时钟）技术。利用 66.6MHz 的时钟信号上升沿和下降沿都可以同时存取数据，相当于使工作时钟频率提高为原来的两倍，即达到 133MHz。

(2) AGP 允许图形控制器直接访问系统主内存。因此 AGP 可直接对系统主存中的图像进行处理，可减少主内存和图形控制器之间的数据传输量，从而加快数据传输速度。

(3) 采用流水线技术进行内存读写，减少了总线阻塞，提高了图形处理效率。

(4) 采用多路信号分离技术，把总线上的地址信号与数据信号分离，通过使用边带寻址 SBA(sideband address)总线来提高随机内存的访问速度。

(5) 采用 DIME 技术，将纹理数据置于帧缓冲区之外的系统主内存，从而让出帧缓冲区和带宽供其他功能使用，以获得更高的屏幕分辨率，或者允许 Z 缓冲产生更大的屏幕面积。

8. ATA/ATA-2 接口标准的功能是什么？二者有何不同？

【解】ATA/ATA-2 接口标准是硬盘控制器的接口标准。该接口把早期的硬盘接口中的控制器部分设置到驱动器中，故在硬盘适配卡中减少了控制器部分，大大地简化了硬盘适配器逻辑

电路。驱动器中借助于控制器逻辑可采用不同编码方式和定位方式，大幅度提高了磁道密度和位密度，从而大大增加了磁盘存储空间的容量。控制器设置到驱动器中解决了驱动器和控制器之间的数据丢失问题，增强了数据传输的可靠性且提高了数据传输速率。

二者的不同：

(1) ATA 标准仅是一个硬盘标准，故其只支持硬盘。ATA-2 标准支持符合 ATAPI(AT attachment packet interface)接口标准的硬盘，磁带驱动器及不 CD-ROM 驱动器等。

(2) ATA 标准提供了一个和主板连接器插口相接的 ATA 插座，最多只能挂两个硬盘。ATA-2 能提供两个接口插座，可连接四个符合 ATAPI 接口的设备。

(3) ATA 标准最多允许硬盘的磁头数为 16 个，这样就限定了硬盘最大容量为 528MB。ATA-2 支持大容量硬盘，磁带机等，其中硬盘最大容量达 8.4GB。

(4) ATA 标准中的 DMA 方式是一种可选方式，一般很少使用。它支持 PIO0、PIO1、PIO2 三种 PIO 方式传送数据。该标准采用的三种 PIO 方式的最小总线周期分别为 600ns、383ns、240ns。ATA-2 标准增加了 PIO3 和 PIO4 两种新 PIO 方式传送数据，总线周期分别为 180ns 和 120ns，速度更快。ATA-2 标准中 DMA 方式得到充分利用，DMA 方式分别为单字 DMA 方式和多字 DMA 方式。

9. SCSI 总线的信号线是如何定义的？其总线操作过程分为哪几个阶段？

【解】以 8 位数据 SCSI 总线、单端方式传输信号为例，传输线是一根 50 芯电缆，接口信号定义如下：

DB（0-7）：8 位数据总线。

DB（P）：奇偶校验位。

TERMPWR：终端电源（Terminator Power）是 SCSI 设备的片外电源供电电路接入线。

ATN：注意（Attention），这是主设备发出的有信息要传送的信号，希望引起目标设备的注意。

BSY：该信号为真，表明 SCSI 设备处于“忙”（Busy）状态。

ACK：启动设备对目标设备的应答（Acknowledge）信号。

RST：要求总线上所有设备复位。

MSG：在消息（Message）阶段由目标设备置为真。

SEL：选择（Select）信号，启动设备选择目标设备或目标设备选择启动设备。

C/D：表明数据总线所送信息的类型（Control/Data）。

REQ：目标设备向启动设备发出数据传送请求（Request）。

I/O：表明数据传送的方向，也用以区分选择阶段和重选阶段。

其余信号线用作接地线和保留线。

SCSI 总线操作分为 8 个阶段：总线空闲阶段、仲裁阶段、选择阶段、重选阶段、命令阶段、数据阶段、状态阶段和消息阶段。

10. 什么是 USB？简述 USB 的系统组成。

【解】USB(Universal Serial Bus)是一个通用串行总线接口标准，用一种通用的连接器连接多种类型的外设，其通用连接器为 4 针插头，支持热插拔连接，即插即用，具备较快的传输速度。

USB 系统是由硬件和软件两大部分组成。USB 硬件部分包括 USB 主机、USB 设备和连

接电缆。USB 系统软件部分主要由二个驱动程序来组成，分述是 USB 设备驱动程序，USB 驱动程序、主控制器驱动程序。

11. USB 有哪几种数据传输方式？

【解】USB 系统信息流中有四种类型的数据流，与其对应的数据传输方式亦有四种方式。

(1) 控制传输(Control)：控制传输主要用作与配置设备也可以用作设备的其他用途。控制传输是双向传输，一般传输经历两个阶段，分别为 setup 阶段、data transmission 阶段和 status 阶段。setup 阶段 USB 主机发出命令给设备；data transmission 阶段是传输上阶段所设定的数据；status 阶段主机接收设备返回的握手信号。USB 协议规定 USB 设备必须用端点零完成控制传送。其时间是当 USB 设备第一次被 USB 主机检测到时和 USB 主机交换信息。其内容是提供设备配置、设定外设和传送状态的双向通信。

(2) 批传输(bulk)：它是用来传送大批数据，可以是单向也可以是双向，批传输传送的时间性要求不高，但要确保数据的正确性。

(3) 中断传输(interrupt)：它是用于随机的、少量的数据传送，其信息传输是单向的。USB 中断传输是用查询方式实现，主机要频繁地请求端点输入，当查询到输入时进行数据传送。

(4) 等时传输(isochronous)：等时传输用于传送连续性、实时性较强的信息，可单向也可双向传输。此种方式的特点是要求传输速率固定，时间性强，不计传送错误，也就是传输中出现错误也不重复执行。

12. USB 总线上每一次交换不少需要哪几个包才能完成？

【解】USB 总线上的每一次交换一般都需要三个传送包方能实现，它们分别是标志包、数据包、握手包，但是等时传输不需要握手包。

USB 总线上的每一次交换的过程是：首先由主机发出标志传送包，该包中有 USB 设备地址码、端点号、传输方向和传输类型等信息。然后数据源向数据目的发送数据传送包或发送无数据传送的指示信息信号，在每一次交换中，数据传送包中的数据最多为 1023B。最终是数据目的方向数据源发送一个握手信息包，提示数据是否正常发送的反馈信息，若出现发送错误，则需重复执行。

13. IEEE 1394 的协议层次结构是什么？

【解】IEEE 1394 标准用一组三层协议使主控机与外设交互标准化。其三层协议为物理层，链接层和交换层。由串行总线管理将这三个层次连接起来。

(1) 物理层(physical layer)

物理层根据不同总线的物理介质，将数据链接层的逻辑信号转换成实际的物理电信号，并提供了保证每次只有一种设备传输数据的仲裁服务。该层又包含物理协议子层和物理介质相关子层，其中物理协议子层的功能是控制和管理总线仲裁方式，提供了使用本地时钟再同步数据以及数据传输速度的自动检测。物理介质相关子层定义了机械和电器接口以及信号传输的方法。

(2) 链接层(link layer)

链接层定义了以数据传输包形式进行的数据传输服务。该层为异步传输数据包和等时传输数据包提供了包传送功能。该层主要完成数据的编址、校验和数据包的制作。

(3) 交换层(transaction layer)

交换层定义了请求应答协议用以执行总线传输。该层支持对异步传输协议的读/写和锁

定，读命令使接收端向发送端返回数据，写命令使发送端发送数据到接收端。锁定命令综合了读/写两种功能，该层隐藏了 IEEE 1394 较低层的细节方便了用户的使用。

14.有哪几种控制器结构？

【解】VXI 总线常用的控制器有以下几种：

(1) IEEE－488 控制器。它可以接几个主机箱，亦可同时接 IEEE－488 仪器。其特点是具有最普通的仪器界面，容易将 VXI 总线仪器与 IEEE－488 仪器混合使用，最高传输速率为 1MB/s，速度最慢，最多可控制 168 个标准模块，价格最低。

(2) 嵌入式（亦称内置式）VXI 总线控制器。它能直接放入主机箱中，并具有一台通用计算机（486 或 586 微机）的全部功能。其主要特点是外形尺寸小，能直接控制 VXI 总线及 12 个标准模块，传输速率可达 40MB/s，速度最快。

(3) MXI 控制器。其价格与 IEEE－488 控制器相当，传输速率可达(20～33)MB/s。