

第1章 习题答案

1.答:

计算机硬件的基本组成部分有: 运算器、控制器、存储器、输入设备和输出设备五大组成部分;

运算器功能: 完成各种算术运算或逻辑运算;

控制器功能: 发出各种控制信息, 使计算机各部件协调工作;

存储器功能: 存储程序和数据;

输入设备: 将程序和数据输入的部件;

输出设备: 将结果数据输出的部件。

2.答:

以微型计算机为核心, 配以鼠标、键盘等外围设备、电源、接口电路, 以及控制计算机工作的软件构成微型计算机系统。

3.答:

主板与外部设备之间的数据传输必须通过系统总线, 所以系统总线包含的信号线必须满足下列各种输入/输出操作的需要: ① 访问分布于主板之外的存储器; ② 访问 I/O 接口; ③ 适应外部中断方式; ④ 适应存储器直接与外部设备交换信息。总线控制逻辑的任务就是产生和接受这些操作所需要的信号。

4.答:

计算机有运算器、控制器、存储器、输入设备和输出设备五大部分组成, 计算机内部采用二进制数据的格式表示数据和指令。程序将事先保存在内存储器中, 计算机在工作时, 不需要操作人员干预, 自动逐条取出指令并执行。

5.答:

CPU 首先执行主板上 BIOS 中的自检程序; 执行引导装入程序, 依次搜寻硬盘、光盘等; 读出引导记录, 装入引导程序; 由引导程序将操作系统装入内存; 执行操作系统。

6.答:

BIOS 是 Basic Input Output System 的缩写, 指的是基本输入输出系统。它是一组固化到计算机 ROM 芯片上的程序, 保存着计算机最重要的基本输入输出的程序、系统设置信息、开机后自检程序和系统自启动程序。

7.答: 略。

第1章 习题答案

1.答:

为了区别所使用的数制, 一般用以下两种书写格式表示:

① 用括号将数字括起, 后面加数制区分, 数制用下标的形式给出;

② 用后缀区分, 二进制数、十进制数、八进制数、十六进制数的后缀分别为字母 B (或 b)、D (或 d)、O (或 o) 或 Q (或 q)、H (或 h)。

例如: 十六进制数 56.78 可以表示成 $(56.78)_{16}$ 或 56.78H;

十进制数 56.78 可以表示成 $(56.78)_{10}$ 或 56.78D。

2.答:

123D 采用十进制, 0AFH 采用十六进制, 77Q 采用八进制, 1001110B 采用二进制。

3.答:

字长为 8 位的二进制数原码表示的最大值: 127D, 最小值: -127D;

补码表示的最大值: 127D, 最小值: -128D。

字长为 16 位的二进制数原码表示的最大值: 32767D, 最小值: -32767D;

补码表示的最大值: 32767D, 最小值: -32768D。

4.答:

(1) 125D=0111 1101B=7DH

(2) 255D=1111 1111B=FFH

(3) 72D=0100 1000B=48H

(4) 5090D=0001 0011 1110 0010B=13E2H

5.答:

(1) 1111 0000B=240D=F0H

(2) 1000 0000 B =128D =80H

(3) 1111 1111 B =255 D =FFH

(4) 0101 0101B=85D=55H

6.答:

(1) FFH=255D=1111 1111B

(2) ABCDH=43947D=1010 1011 1100 1101B

(3) 123H=291D=0000 0001 0010 0011B

(4) FFFFH=65535D=1111 1111 1111 1111B

7.答:

(1) 字长为 8 位时 $(16)_{原}=0001\ 0000$; $(16)_{补}=0001\ 0000$;

字长为 16 位时 $(16)_{原}=0000\ 0000\ 0001\ 0000$; $(16)_{补}=0000\ 0000\ 0001\ 0000$;

(2) 字长为 8 位时 $(-16)_{原}=1001\ 0000$; $(-16)_{补}=1111\ 0000$;

字长为 16 位时 $(-16)_{原}=1000\ 0000\ 0001\ 0000$; $(-16)_{补}=1111\ 1111\ 1111\ 0000$;

- (3) 字长为 8 位时 $(+0)_{\text{原}}=0000\ 0000$; $(+0)_{\text{补}}=0000\ 0000$;
 字长为 16 位时 $(+0)_{\text{原}}=0000\ 0000\ 0000\ 0000$; $(+0)_{\text{补}}=0000\ 0000\ 0000\ 0000$;
 (4) 字长为 8 位时 $(-0)_{\text{原}}=1000\ 0000$; $(-0)_{\text{补}}=0000\ 0000$;
 字长为 16 位时 $(-0)_{\text{原}}=1000\ 0000\ 0000\ 0000$; $(-0)_{\text{补}}=0000\ 0000\ 0000\ 0000$;
 (5) 字长为 8 位时 $(127)_{\text{原}}=0111\ 1111$; $(127)_{\text{补}}=0111\ 1111$;
 字长为 16 位时 $(127)_{\text{原}}=0000\ 0000\ 0111\ 1111$; $(127)_{\text{补}}=0000\ 0000\ 0111\ 1111$;
 (6) 字长为 8 位时-128 超过原码表示的范围; $(-128)_{\text{补}}=1000\ 0000$;
 字长为 16 位时 $(-128)_{\text{原}}=1000\ 0000\ 1000\ 0000$; $(-128)_{\text{补}}=1111\ 1111\ 1000\ 0000$;
 (7) 字长为 8 位时 $(121)_{\text{原}}=0111\ 1001$; $(121)_{\text{补}}=0111\ 1001$;
 字长为 16 位时 $(121)_{\text{原}}=0000\ 0000\ 0111\ 1001$; $(121)_{\text{补}}=0000\ 0000\ 0111\ 1001$;
 (8) 字长为 8 位时 $(-9)_{\text{原}}=1000\ 1001$; $(-9)_{\text{补}}=1111\ 0111$;
 字长为 16 位时 $(-9)_{\text{原}}=1000\ 0000\ 0000\ 1001$; $(-9)_{\text{补}}=1111\ 1111\ 1111\ 0111$;

8.答:

- (1) $[x]_{\text{补}}=1100\ 0010$; (2) $[-x]_{\text{补}}=0000\ 1101$; (3) $[x]_{\text{原}}=1100\ 0010$; (4) $[x]_{\text{反}}=1011\ 1101$ 。

9.答:

- (1) $A > B$; (2) $A < B$

10.答:

- (1) 溢出; (2) 13H; (3) 溢出; (4) EDH; (5) EDH。

11.答:

- (1) 1000 1000; (2) 1110 1110; (3) 0110 0110; (4) 1010 1100;
 (5) 0000 0000; (6) 1010 1100; (7) =0101 0011。

12.答:

- (1) 30H=48, 字符为 '0'; (2) 39H=57, 字符为 '9'; (3) 42H=66, 字符为 'B';
 (4) 62H=98, 字符为 'b'; (5) 20H=32, 字符为空格; (6) 7H=7, 字符为报警符;

13、答:

十进制	49	123	7	62
压缩 BCD 码	0100 1001B	0001 0010 0011B	0000 0111B	0110 0010B
非压缩 BCD 码	0000 0100 0000 1001B	0000 0001 0000 0010 0000 0011B	0000 0111B	0000 0110 0000 0010B
ASCII 码	3439H	313233H	37H	3632H

第2章 习题答案

1. (1) 答:

物理地址: 物理地址 (PA) 是 20 位无符号二进制数, 是 CPU 访问存储器的实际地址。每个存储单元对应一个物理地址。8086 存储空间的物理地址范围是: 00000H~FFFFFH。

逻辑地址: 采用分段结构的存储器中, 把通过段地址和偏移地址来表示的存储单元的地址称为逻辑地址, 记为: 段地址: 偏移地址。

段地址: 8086 规定各逻辑段从节的整数边界开始, 即段首地址二进制值的低 4 位是 0000, 把段首地址的高 16 位称为段基址或段地址。

偏移地址: 把某一存储单元相对于段地址的段内偏移量称为偏移地址 (也称有效地址 EA)。段地址和偏移地址都是 16 位无符号二进制数。

(2) 答:

时钟周期: 计算机的“时钟”是由振荡源产生的、幅度和周期不变的节拍脉冲, 每个脉冲周期称为时钟周期, 又称为 T 状态或 T 周期, 时钟周期是微机系统工作的最小时间单元。

总线周期: 当 CPU 访问存储器或输入/输出端口时, 需要通过总线进行读或写操作, 这个过程称为总线周期 (Bus Cycle)。总线周期是利用总线完成一次读/写所需要的时间。

指令周期: 执行一条指令所需要的时间称为指令周期 (Instruction Cycle)。指令周期由 1 个或多个总线周期组成。

(3) 答:

最小模式: 也称为单处理器模式, 是指系统中只有一片 8086 微处理器, 所连接的存储器容量不大、片子不多, 所要连接的 I/O 端口也不多, 系统的控制总线就直接由 CPU 的控制线供给, 从而使得系统中的总线控制电路减到最少。最小模式适用于较小规模的系统。

最大模式: 相对于最小模式而言, 适用于中、大型规模的系统。系统中有多个微处理器, 其中一个为主处理器 8086, 其他的处理器称为协处理器, 承担某方面专门的工作。需要增加一片 8288 来对 8086CPU 发出的控制信号进行变换和组合, 以得到对存储器或 I/O 端口的读/写信号和对锁存器、总线收发器的控制信号。

2. 答:

EU: 负责指令的执行, 即从总线接口部件 BIU 的指令队列取指令, 指令执行后向 BIU 送回运算结果, 同时把运算结果的状态特征保存到标志寄存器中。

BIU: 负责 CPU 与存储器、I/O 设备之间的数据传送。BIU 完成以下操作: 取指令送给指令队列、配合执行部件从指定的内存单元或者外设端口中取数据、将数据传送给执行部件或者把执行部件的操作结果传送到指定的内存单元或外设端口中。

8086 的 BIU 和 EU 在很多时候可以并行工作, 使得取指令、指令译码和执行指令这些操作构成操作流水线。

① 当指令队列中有两个空字节, 且 EU 没有访问存储器和 I/O 接口的要求时, BIU 会自动把指令取到指令队列中。

② 当 EU 准备执行一条指令时, 它会从指令队列前部取出指令执行。在执行指令的过程中, 如果需要访问存储器或者 I/O 设备, 那么 EU 会向 BIU 发出访问总线的请求, 以完成访问存储器或者 I/O 接口的操作。如果此时 BIU 正好处于空闲状态, 那么, 会立即响应 EU 的总线请求; 但如果 BIU 正在将某个指令字节取到指令队列中, 那么, BIU 将首先完成这

个取指令操作，然后再去响应 EU 发出的访问总线的请求。

③ 当指令队列已满，而且 EU 又没有总线访问时，BIU 便进入空闲状态。

④ 在执行转移指令、调用指令和返回指令时，下面要执行的指令就不是在程序中紧接着的那条指令了，而 BIU 往指令队列装入指令时，总是按顺序进行的。在这种情况下，指令队列中已经装入的指令就没有用了，会被自动消除。随后，BIU 会往指令队列中装入另一个程序段中的指令。

3.答：

地址信号是 CPU 发送给内存或 I/O 设备的，所以地址线是单向的；

数据信号可以从 CPU 发送给内存或 I/O 设备，也可以从内存或 I/O 设备发送给 CPU，故而数据线是双向的。

4.答：

8086CPU 中有 14 个寄存器。它们是：

① 4 个 16 位的通用寄存器：AX、BX、CX、DX；

② 8 个 8 位的通用寄存器：AH、AL、BH、BL、CH、CL、DH、DL；

③ 指针和变址寄存器 SP、BP、SI、DI；

④ 4 个 16 位的段寄存器 CS、DS、SS 和 ES；

⑤ 标志寄存器 FR；

⑥ 指令指针寄存器 IP。

功能说明详见教材。

5.答：

(1) 加法运算的结果为：CEACH；CF=0，PF=1，AF=0，ZF=0，SF=1，OF=1；

(2) 加法运算的结果为：68ACH；CF=0，PF=1，AF=0，ZF=0，SF=0，OF=0；

(3) 加法运算的结果为：DDDDH；CF=0，PF=1，AF=0，ZF=0，SF=1，OF=0。

6.答：

8086CPU 可寻址的存储器地址范围是 00000H~FFFFFFH；

可寻址的 I/O 端口地址范围是 0000H~FFFFH。

7.答：

由于 8086CPU 提供 20 位地址，但 8086 中可用来存放地址的寄存器，如 IP、SP、BX、SI 等都是 16 位的，只能直接寻址 64KB。为了寻址 1MB 存储空间，8086CPU 采用了典型的存储器分段技术，即将整个存储器空间分为许多逻辑段，每个逻辑段的容量小于或等于 64KB。分段后，对存储器的寻址操作不再直接用 20 位的物理地址，而是采用段地址加段内偏移地址的二级寻址方式。

8.答：

将 AH 的内容送存储单元的操作过程中： $M/\overline{IO}=1, \overline{WR}=0, \overline{RD}=1, DT/\overline{R}=1$ ；

将 I/O 端口的内容送 AL 的操作过程中： $M/\overline{IO}=0, \overline{WR}=1, \overline{RD}=0, DT/\overline{R}=0$ 。

9.答:

$\overline{\text{BHE}}$	A0	有效的数据引脚	操 作
0	0	AD15~AD0 (一个总线周期同时访问奇体和偶体, 从奇地址单元读/写字数据的高 8 位, 从偶地址单元读/写字数据的低 8 位)	从偶地址读/写一个字
1	0	AD7~AD0	从偶地址读/写一个字节
0	1	AD15~AD8	从奇地址读/写一个字节
0	1	AD15~AD8 (第一个总线周期从奇字数据的低 8 位)	从奇地址读/写一个字
1	0	AD7~AD0 (第二个总线周期从偶地址单元读/写字数据的高 8 位)	

10.答:

8086 系统复位后, 指令指针 (IP) 为 0000H; CS 寄存器为 FFFFH, 其他寄存器为 0000H; 指令队列清空。

11.答:

8086 的 $A_{19}/S_6 \sim A_{16}/S_3$ 和 $AD_{15} \sim AD_0$ 是复用信号, 需要地址锁存器将地址信息保存起来, 为外接存储器或外设提供地址信息。

需要锁存的信号是: 地址信号及 $\overline{\text{BHE}}$ 信号。

12.答:

8086CPU 一个基本的总线周期包含 4 个时钟周期: T1、T2、T3 和 T4。

在 T1 状态, $M/\overline{\text{IO}}$ 有效, 指示 CPU 访问的是存储器还是外设, 之后 CPU 往多路复用总线上发出地址信息, 以指出要寻址的存储单元或外设端口的地址。在 T1 状态, CPU 还必须在 ALE 引脚上输出一个正脉冲作为地址锁存信号。 $\overline{\text{BHE}}$ 信号也在 T1 状态送出, 它用来表示数据传送的字宽。

在 T2 状态, CPU 从总线上撤销地址, 使总线的低 16 位浮空, 置成高阻状态, 为传输数据作准备。总线的最高 4 位 ($A_{19} \sim A_{16}$) 用来输出本总线周期的状态信息。读信号 $\overline{\text{RD}}$ 或写信号 $\overline{\text{WR}}$ 在 T2 状态变为有效, 指示 CPU 将进行哪种操作 (读或写)。

在 T3 状态, 多路总线的高 4 位继续提供状态信息, 而多路总线的低 16 位上出现由 CPU 读出的数据或者 CPU 从存储器或端口写入的数据。

在 T4 状态和前一个状态的交界处, CPU 对数据总线进行采样, 获得数据, 总线周期结束。

13.答:

在有些情况下, 外设或存储器速度较慢, 不能及时地配合 CPU 传送数据。这时, 外设或存储器会通过 “READY” 信号线在 T3 状态启动之前向 CPU 发一个 “数据未准备好” 信号, 于是 CPU 会在 T3 之后插入 1 个或多个附加的时钟周期 T_w 。取决与外设或存储器速度。

14.答:

两种操作时序的不同之处发生在 T1 和 T2 状态。

① 在 T1 状态，读周期： $\overline{DT}/\overline{R}$ 应输出低电平；写周期： $\overline{DT}/\overline{R}$ 应输出高电平。

② 在 T2 状态，读周期： \overline{RD} 有效，而 \overline{WR} 无效， $AD_{15} \sim AD_0$ 为高阻态；写周期： \overline{RD} 变为无效，而写信号 \overline{WR} 变为有效， $AD_{15} \sim AD_0$ 在地址撤销之后立即送出要写入存储器或外设端口的数据。

15.答：

在最小模式下，读信号 \overline{RD} 、ALE 和 $\overline{DT}/\overline{R}$ 、 \overline{DEN} 等信号直接由 CPU 给出；

在最大模式下，总线控制器 8288 根据 $\overline{S_2}$ 、 $\overline{S_1}$ 和 $\overline{S_0}$ 状态信号产生读信号 \overline{MRDC} 和 \overline{IORC} ，

ALE 和 $\overline{DT}/\overline{R}$ 、DEN 也是由 8288 发出的，而且 DEN 信号的极性与 CPU 在最小模式下发出的 \overline{DEN} 信号正好相反。

电子工业出版社版权所有
盗版必究

第3章 习题答案

1.答:

指令中关于如何求出操作数有效地址的方法称为寻址方式。

8086CPU 支持多种寻址方式, 根据操作数的类型及来源大致分为 3 类: 数据寻址、转移地址寻址和 I/O 寻址。

2.答:

8086 汇编指令可以采用 7 种基本的数据寻址方式: ①立即寻址; ②寄存器寻址; ③直接寻址; ④寄存器间接寻址; ⑤寄存器相对寻址; ⑥基址变址寻址; ⑦相对基址变址寻址。与存储器寻址方式(后 5 种)相比, 寄存器寻址方式最快。

3.答:

如果指令中没有用前缀说明操作数存放在哪个段, 则操作数默认存放在数据段。

8086 系统允许操作数存放在代码段、堆栈段或附加段。此时, 就需要在指令中利用前缀指明段超越。例如:

MOV ES:[1225H],AX

4.答:

- (1) 寄存器寻址方式
- (2) 寄存器间接寻址方式, PA=10300H
- (3) 寄存器间接寻址方式, PA=20200H
- (4) 直接寻址方式, PA=10060H
- (5) 基址变址寻址方式, PA=10500H
- (6) 相对基址变址寻址方式, PA=12300H
- (7) 直接寻址方式, PA=11000H
- (8) 寄存器相对寻址方式, PA=10306H
- (9) 直接寻址方式, PA=10065H

5.答:

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)
×	×	×	√	×	√	×	×	√	×	×

6.答:

(1) 源操作数的寻址方式不同。MOV AX,3000H 指令的源操作数采用的是立即数寻址方式, MOV AX,[3000H]指令的源操作数采用的是直接寻址方式。

(2) 指令执行后, AX 的值不同。MOV AX,MEM 指令执行后, AX 得到的是 MEM 单元的内容, 而 MOV AX,OFFSET MEM 指令执行后, AX 得到的是 MEM 单元的地址。

(3) 指令不同。MOV AX,MEMMOV AX,MEM 指令执行后, AX 得到的是 MEM 单元的内容, 而 LEA AX,MEM 指令执行后, AX 得到的是 MEM 单元的地址。

(4) 转移的类型不同。JMP SHORT L1 指令实现的是段内转移，而 JMP NEAR PTR L1 指令实现的是段间转移。

(5) 指令的执行结果不同。CMP DX,CX 指令执行后，DX 的值不变，而 SUB DX,CX 指令执行后，DX 的值是两寄存器值的差。

(6) 操作数所在段不同。MOV [BP][SI],CL 指令访问的是堆栈段的存储单元，而 MOV DS:[BP][SI],CL 指令访问的是数据段的存储单元。

7.答：

(1) MOV BYTE PTR [BP],200, 该指令修改字节单元 (58200H) 的值，指令执行后，该单元的值：200。

(2) MOV WORD PTR [BX],2000 该指令修改字单元 (22400H) 的值，指令执行后，该单元的值：2000。

8.答：

当前栈顶地址是：2010:FE00H；

执行 PUSH BX 指令后，栈顶地址变为：2010:FDFEH，栈顶 2 字节内容是：3457H。

9.答：

(1) (DX) =3C62H, CF=1

(2) (DX) =03C6H, CF=0

(3) (DX) =18A0H, CF=1

(4) (DX) =2BC6H, CF=0

(5) (DX) =18B7H, CF=1

(6) (DX) =BCC5H, CF=0

10.答：

	(AX)	CF	ZF	OF	SF	PF
(1)	134EH	0	0	0	0	1
(2)	0821H	0	0	0	0	1
(3)	F142H	1	0	0	1	1
(4)	0A69	1	0	0	1	1
(5)	F596	-	-	-	-	-
(6)	0A69	0	0	0	0	1

11.答：

(1) CMP CX,DX

JA NEXT

(2) CMP AX,BX

JNA NEXT

(3) CMP DX,0

JZ NEXT

(4) CMP CX,DX

JBE NEXT

12.答:

- (1) 程序转向 L1。
- (2) 程序转向 L1。
- (3) 程序转向 L2。
- (4) 程序转向 L5。
- (5) 程序转向 L5。
- (6)

13.答:

因为普通运算指令执行的是二进制数的运算，而 BCD 码是十进制数，所以，需要对运算结果进行十进制调整。

在做 BCD 码的加、减和乘法运算时，十进制调整指令放在运算指令之后；而作 BCD 码的除法运算时，在运算指令之前用十进制调整指令对被除数进行调整。

14.答:

```
MOV CL,3
SHR BX,CL
```

15. 答: (略)

16.答:

```
DATA    SEGMENT
COUNT=100H
        ORG 1000H
D_DATA DB COUNT      DUP(?)
        ORG 2170H
S_DATA DB      COUNT/5  DUP(1,2,3,4,5)
DATA    ENDS
CODE    SEGMENT
        ASSUME CS:CODE,DS:DATA,ES:DATA
START:
MOV     AX,DATA
MOV     DS,AX
MOV     ES,AX
LEA     SI,S_DATA
LEA     DI,D_DATA
MOV     CX,COUNT
REP     MOVSB
MOV     AL,1
MOV     CX,COUNT
LEA     DI,D_DATA
```

17.答:

```
AGAIN:
CMP     BYTE PTR [DI],AL
JNZ     NEXT
MOV     BYTE PTR [DI],''
NEXT:
INC     DI
LOOP    AGAIN
EXIT:
MOV     AH,4CH
INT     21H
CODE    ENDS
END START
```

17. 答:

```
CODE    SEGMENT
        ASSUME CS:CODE
START:
        MOV     AX,5678H
        MOV     DX,1234H
        NOT     AX
        NOT     DX
        ADD     AX,1
        ADC     DX,0
EXIT:
        MOV     AH,4CH
        INT     21H
CODE    ENDS
        END     START
```

18. 答:

;本程序未考虑溢出的情况。

```
DATA    SEGMENT
        A1 DW 5050H
        A2 DW ?    ;存 A1 的反码
        A3 DW ?    ;存 A1 的补码
DATA    ENDS
CODE    SEGMENT
        ASSUME CS:CODE,DS:DATA
START:
        MOV     AX,DATA
        MOV     DS,AX
        MOV     AX,A1
        NOT     AX
        MOV     A2,AX
        INC     AX
        MOV     A3,AX
EXIT:
        MOV     AH,4CH
        INT     21H
CODE    ENDS
        END     START
```

19. 答:

```
DATA SEGMENT ;AT 5000H
    ORG 3481H
    DAT DB 12H
        DB ?,?,?
DATA ENDS
CODE SEGMENT
    ASSUME CS:CODE,DS:DATA
START:
    MOV AX,DATA
    MOV DS,AX
    MOV AL,DAT
    NEG AL
    MOV DAT+1,AL
    MOV AL,DAT
    XOR AL,00001111B
    MOV DAT+2,AL
    MOV AL,DAT
    OR AL,11110000B
    MOV DAT+3,AL
EXIT:
    MOV AH,4CH
    INT 21H
CODE ENDS
    END START
```

20. 答:

```
COUNT=1000
DATA SEGMENT
    ORG 1000H
    DATDB 10 DUP (12H,-5,-3,0,-128,56H,98H,4,128,200)
    ORG 2000H
    MINDAT DB ?
DATA ENDS
CODE SEGMENT
    ASSUME CS:CODE,DS:DATA
START:
    MOV AX,DATA
    MOV DS,AX
    LEA SI,DAT
    MOV CX,COUNT
    DEC CX
    MOV AL,[SI]
NEXT:
    INC SI
    CMP AL,[SI]
    JLE ISMIN
    MOV AL,[SI]
ISMIN:
    LOOP NEXT
    MOV MINDAT,AL
EXIT:
    MOV AH,4CH
    INT 21H
CODE ENDS
    END START
```

21. 答:

```
DATA    SEGMENT
    STRING1      DB  'HELLO!'
    COUNT1=$-STRING1
    STRING2      DB  'HELLO!'
    COUNT2=$-STRING2
    IM           DB  'MATCH$'
    NM           DB  'NOT MATCH$'
DATA    ENDS
CODE    SEGMENT
    ASSUME CS:CODE,DS:DATA
START:
    MOV  AX,DATA
    MOV  DS,AX
    LEA  SI,STRING1
    LEA  DI,STRING2
    MOV  CX,COUNT1
    MOV  BX,COUNT2
    CMP  CX,BX
    JNZ  DISPNOTMATCH
NEXT:
    MOV  AL,[SI]
    MOV  AH,[DI]
    CMP  AL,AH
    JNZ  DISPNOTMATCH
    INC  SI
    INC  DI
    LOOP NEXT
ISMATCH:
    MOV  DX,OFFSET IM
    MOV  AH,9
    INT  21H
    JMP  EXIT
DISPNOTMATCH:
    MOV  DX,OFFSET NM
    MOV  AH,9
    INT  21H
EXIT:
    MOV  AH,4CH
    INT  21H
CODE    ENDS
        END    START
```

22. 答:

```
DSEG  SEGMENT
    DATA  DB  5,6,7,8
           DW  ?
    DATA2 DB  1,10,100,20
DSEG  ENDS
CODE  SEGMENT
    ASSUME CS:CODE,DS:DSEG
START:
    MOV    AX,DSEG
    MOV    DS,AX
    MOV    DX,0
    MOV    CX,4
    LEA    SI,DATA
    LEA    DI,DATA2
NEXT:
    MOV    AL,[DI]
    MOV    BL,[SI]
    CALL   DOMUL
    ADD    DX,AX
    INC    DI
    INC    SI
    LOOP   NEXT
    MOV    WORD PTR DATA+4,DX
EXIT:
    MOV    AH,4CH
    INT    21H

DOMUL PROC
    MUL    BL
    RET
DOMUL ENDP
CODE  ENDS
    END    START
```

23. 答:

```
MOV    CX,100
MOV    AX,SEG    LIST
MOV    DS,AX
MOV    AX,SEG    BLK
MOV    ES,AX
LEA    SI,LIST
ADD    SI,CX
LEA    DI,BLK
ADD    DI,CX
STD
REP    MOVSB
```

24. 答:

```
DSEG  SEGMENT
      BUFFER DW 8
          DB '12345678'
DSEG  ENDS
CODE  SEGMENT
      ASSUME CS:CODE,DS:DSEG
START:
      MOV     AX,DSEG
      MOV     DS,AX
      MOV     CX,BUFFER
      SAR     CX,1
      LEA     SI,BUFFER
      MOV     DI,SI
      ADD     DI,2
AGAIN:
      ADD     SI,2
      MOV     AL,[SI]
      AND     AL,0FH
      SAL     BYTE PTR [SI+1],1
      SAL     BYTE PTR [SI+1],1
      SAL     BYTE PTR [SI+1],1
      SAL     BYTE PTR [SI+1],1
      OR      AL,BYTE PTR [SI+1]
      MOV     [DI],AL
      INC     DI
      LOOP    AGAIN
EXIT:
      MOV     AH,4CH
      INT     21H
CODE  ENDS
      END     START
```

25. 答:

JMP SHORT LAB 指令是一条双字节指令。这条指令取出后, (IP)=0102H, 转移目标的偏移地址=(IP)+位移量。所以, 转移目标的物理地址=(CS)×16+(IP)+位移量。

- (1) 转移目标的物理地址=(CS)×16+0158H
- (2) 转移目标的物理地址=(CS)×16+0182H
- (3) 转移目标的物理地址=(CS)×16+017AH
- (4) 转移目标的物理地址=(CS)×16+01E2H

26. 答:

```
DSEG  SEGMENT
      ORG 10H
      DAT    DB 10
          DB ?
DSEG  ENDS
CODE  SEGMENT
      ASSUME CS:CODE,DS:DSEG
START:
      MOV     AX,DSEG
      MOV     DS,AX
      MOV     AL,DAT
      MOV     BL,AL
      SAL     BL,1
      ADD     AL,BL
      MOV     CL,2
      SAL     BL,CL
      ADD     AL,BL
      MOV     DAT+1,AL
EXIT:
      MOV     AH,4CH
      INT     21H
CODE  ENDS
      END     START
```


27. 答:

```
DSEG  SEGMENT
      ORG  10H
      DAT  DB  10H,20H
          DB  ?,?
DSEG  ENDS
CODE  SEGMENT
      ASSUME CS:CODE,DS:DSEG
START:
      MOV  AX,DSEG
      MOV  DS,AX
      MOV  AL,DAT
      MOV  AH,DAT+1
      MOV  CL,3
      SAR  AX,CL
      MOV  DAT+2,AL
      MOV  DAT+3,AH
EXIT:
      MOV  AH,4CH
      INT  21H
CODE  ENDS
      END START
```

28. 答:

```
DSEG  SEGMENT
      BLOCK DW 8 DUP (10H,20H,-3,-9)
DSEG  ENDS
CODE  SEGMENT
      ASSUME CS:CODE,DS:DSEG
START:
      MOV  AX,DSEG
      MOV  DS,AX
      LEA  SI,BLOCK
      MOV  CX,32
AGAIN:
      MOV  AX,[SI]
      CMP  AX,0
      JNS  NEXT
      NEG  AX
      MOV  [SI],AX
NEXT:
      INC  SI
      INC  SI
      LOOP AGAIN
EXIT:
      MOV  AH,4CH
      INT  21H
CODE  ENDS
      END START
```

29. 答:

```
DSEG  SEGMENT
        ORG  3030H
        DAT  DW  1234H,-1234H,?,?
DSEG  ENDS
CODE  SEGMENT
        ASSUME CS:CODE,DS:DSEG
START:
        MOV  AX,DSEG
        MOV  DS,AX
        MOV  AX,DAT
        MOV  BX,DAT+2
        IMUL BX
        MOV  DAT+4,AX
        MOV  DAT+6,DX
EXIT:
        MOV  AH,4CH
        INT  21H
CODE  ENDS
        END START
```

电子工业出版社版权所有
盗版必究

第4章 习题答案

1. 答:

标号是指令的符号地址，可用作控制转移指令的操作数。

标号具有 3 种属性：段属性、偏移属性和类型属性。

2. 答:

变量是存储单元的符号地址。

变量具有 3 种属性：段属性、偏移属性和类型属性。

3. 答:

伪指令语句，也称指示性语句，是不可执行语句，汇编后不产生目标代码，它仅仅在汇编过程中告诉汇编程序如何汇编源程序。

宏是一个以宏名定义的指令序列。一旦把某程序段定义成宏，则可以用宏名代替那段程序。在汇编时，要对宏进行宏展开，即把以宏名表示的地方替换为该宏对应的指令序列的目标代码。宏指令可以看成指令语句的扩展，相当于多条指令语句的集合。

4. 答:

汇编语言表达式中有如下运算符：算术操作符、逻辑操作符、移位操作符、关系操作符、数值回送操作符和属性操作符。

操作符所完成的运算在汇编阶段进行。

5. 答:

略。

6. 答:

(1) DB1 DB 10H DUP(1,2, 5 DUP(3),4)

(2) DB2 DB 'STUDENT'

(3) BD3 DW 12H,0ABCDH

(4) COUNT EQU DB2-DB1

7. 答:

第一个 OR 表示该指令是 OR 指令，在程序运行时，该 OR 操作被执行。

第二个 OR 是逻辑操作符 OR，在汇编时，OR 运算被执行。

8. 答:

(1) (AX) =1

(2) (AX) =2

(3) (CX) =5

(4) (DX) =0AH

(5) (CX) =1

(6) (DX) =4

9. 答: 略。

10. 答:

```
DSEG  SEGMENT
    STRING  DB  'NEXT123DF$'
DSEG  ENDS
CODE  SEGMENT
    ASSUME CS:CODE,DS:DSEG
START:
    MOV     AX,DSEG
    MOV     DS,AX
    LEA     SI,STRING
    MOV     CX,0
AGAIN:
    MOV     AL,[SI]
    CMP     AL,'$'
    JZ      NEXT
    INC     CX
    INC     SI
    JMP     AGAIN
NEXT:
    MOV     DX,CX    ;字符串长度暂存 DX
MVDAT:
    MOV     [SI]+2,AL
    DEC     SI
    MOV     AL,[SI]
    LOOP    MVDAT    ;从字符串最后一个字符开始移动
    MOV     [SI]+2,AL
    MOV     WORD PTR [SI],DX
EXIT:
    MOV     AH,4CH
    INT     21H
CODE  ENDS
END START
```

11. 答:

```
DSEG  SEGMENT
    STRING  DB  'It is FEB&03'
    COUNT=$-STRING
DSEG  ENDS
CODE  SEGMENT
    ASSUME CS:CODE,DS:DSEG
START:
    MOV     AX,DSEG
    MOV     DS,AX
    LEA     SI,STRING
    MOV     CX,COUNT
AGAIN:
    MOV     AL,[SI]
    CMP     AL,'&'
    JNZ     NEXT
    MOV     AL,''
    MOV     [SI],AL
NEXT:
    INC     SI
    LOOP    AGAIN
EXIT:
    MOV     AH,4CH
    INT     21H
CODE  ENDS
END START
```

12. 答:

```

DSEG    SEGMENT
        BLOCK DB 4,5,-2,7,8,9,2,1,0AH,1BH,87H,23H,44H,33H,45H,28H,0DH,8EH,66H,22H
        COUNT=$-BLOCK
DSEG    ENDS
CODE    SEGMENT
        ASSUME CS:CODE,DS:DSEG
START:
        MOV     AX,DSEG
        MOV     DS,AX
        LEA     SI,BLOCK
        MOV     CX,COUNT
AGAIN:
        MOV     AL,[SI]
        MOV     DI,SI
        MOV     DX,CX
NEXT1:
        CMP     AL,[DI]
        JGE     NEXT
        XCHG    AL,[DI]

NEXT:
        INC     DI
        DEC     DX
        JNZ     NEXT1
        MOV     [SI],AL
        INC     SI
        LOOP    AGAIN
EXIT:
        MOV     AH,4CH
        INT     21H
CODE    ENDS
        END START

```

13. 答:

(3)	1B50H
	1000H
(4)(2)	1A70H
(5)(1)	150BH
(6)	

上表左侧的序号，表示对应操作完成后 SP 指向的栈顶的位置。

14. 答:

```
SUB1 PROC
MOV BX,AX
MOV CL,4
ROL BX,CL
MOV AX,BX
AND AX,000FH
MOV CX,1000
MUL CX ;千位在 DX,AX,有效值在 AX 中
PUSH AX
MOV CL,4
ROL BX,CL
MOV AX,BX
AND AL,0FH
MOV CL,100
MUL CL ;百位在 AX
POP CX
ADD AX,CX ;千百位在 AX
PUSH AX
MOV CL,4
ROL BX,CL
MOV AX,BX
AND AL,0FH
MOV CL,10
MUL CL ;十位在 AX
POP CX
ADD AX,CX ;千百十位在 AX
MOV CL,4
ROL BX,CL
AND BX,0FH
ADD AX,BX
RET
SUB1 ENDP
```

```
SUB2 PROC
MOV DX,0
MOV CX,1000
DIV CX ;千位数字在 AX 中
MOV BX,AX
MOV CL,4
ROR BX,CL ;千位数字移至 BX 的高 4 位
MOV AX,DX
MOV CL,100
DIV CL ;百位数字在 AL 中
OR BH,AL ;千百位置 BH
MOV AL,AH
MOV AH,0
MOV CL,10
DIV CL ;十位数字在 AL 中,个位在 AH 中
MOV CL,4
SAL AL,CL
MOV BL,AL ;十位数字置 BL 高 4 位
OR BL,AH ;十位个位置 BL
MOV AX,BX
RET
SUB2 ENDP
```

SUB3	PROC			N5:	
	PUSH	BX			MOV AL,AH
	MOV	BX,0			MOV AH,0
	LEA	DI,ASCNUM			MOV CL,10
	CMP	AX,0			DIV CL
	JNS	CHG			CMP BX,0
	NEG	AX			JNZ N6
	MOV	BYTE PTR [DI], '-'			CMP AL,0
	JMP	NEXT			JZ N7
CHG:					MOV BX,2
	MOV	BYTE PTR [DI], '+'		N6:	
NEXT:					OR AL,30H
	MOV	DX,0			INC DI
	MOV	CX,10000			MOV [DI],AL ;存十位
	DIV	CX ;万位数字在 AX 中		N7:	
	CMP	AL,0			OR AH,30H
	JZ	N1			INC DI
	MOV	BX,5			MOV [DI],AH
	OR	AL,30H			CMP BX,0
	INC	DI			JNZ SN
	MOV	[DI],AL ;存万位			MOV BX,1
N1:				SN:	
	MOV	AX,DX			MOV CX,BX ;存数位长度
	MOV	DX,0			INC CX ;串长=数位+1 位符号位
	MOV	CX,1000			LEA DX,ASCNUM
	DIV	CX			POP BX
	CMP	BX,0			RET
	JNZ	N2			SUB3 ENDP
	CMP	AL,0			
	JZ	N3			
	MOV	BX,4			
N2:					
	OR	AL,30H			
	INC	DI			
	MOV	[DI],AL ;存千位			
N3:					
	MOV	AX,DX			
	MOV	CL,100			
	DIV	CL			
	CMP	BX,0			
	JNZ	N4			
	CMP	AL,0			
	JZ	N5			
	MOV	BX,3			
N4:					
	OR	AL,30H			
	INC	DI			
	MOV	[DI],AL ;存百位			

15. 答:

```
CODE    SEGMENT
        ASSUME CS:CODE
START:
        MOV     AH,1
        INT     21H
        CMP     AL,'a'
        JB      NOCHG
        CMP     AL,'z'
        JA      NOCHG
        SUB     AL,20H
NOCHG:
        MOV     DL,AL
        MOV     AH,2
        INT     21H

EXIT:
        MOV     AH,4CH
        INT     21H

CODE    ENDS
        END START
```

16. 答:

<pre>;键盘输入的成绩以逗号分割 ;输入完毕按回车结束 DSEG SEGMENT STRING DB 40,?,40 DUP(?) CJ DB 10 DUP(?) S6 DB ? S7 DB ? S8 DB ? S9 DB ? S10 DB ? DSEG ENDS CODE SEGMENT ASSUME CS:CODE,DS:DSEG START: MOV AX,DSEG MOV DS,AX LEA DX,STRING MOV AH,10 INT 21H LEA DI,CJ LEA SI,STRING+2 MOV BH,0 MOV AL,0 AGAIN: MOV BL,[SI] CMP BL',' JZ NEXT CMP BL,0DH JZ NEXT SUB BL,30H MOV CL,10 MUL CL ADD AL,BL INC SI JMP AGAIN</pre>	<pre>NEXT: MOV [DI],AL INC DI MOV AL,0 INC BH INC SI CMP BL,0DH JNZ AGAIN TJ: LEA DI,CJ TJ1: MOV AL,[DI] CMP AL,100 JZ ADDS10 CMP AL,90 JAE ADDS9 CMP AL,80 JAE ADDS8 CMP AL,70 JAE ADDS7 CMP AL,60 JAE ADDS6 ADDS10: INC S10 JMP TJNEXT ADDS9: INC S9 JMP TJNEXT ADDS8: INC S8 JMP TJNEXT ADDS7: INC S7</pre>	<pre>JMP TJNEXT ADDS6: INC S6 TJNEXT: INC DI DEC BH JNZ TJ1 EXIT: MOV AH,4CH INT 21H CODE ENDS END START</pre>
--	---	--

17. 答：略。

18. 答：略。

19. 答：略。

20. 答：略。

21. 答：略。

：

电子工业出版社版权所有
盗版必究

第5章 习题答案

略。

第6章 习题答案

1. 半导体存储器的分类如下：



2. 随机存取存储器 RAM，又称为读/写存储器，其每个存储单元的内容可以随时按需要进行读/写操作。RAM 主要用来保存各种输入/输出数据、中间结果、与外存交换的信息，也可作堆栈使用。而 ROM 的内容只能读出，不能写入或改写，一般用来存放固定的程序和数据。

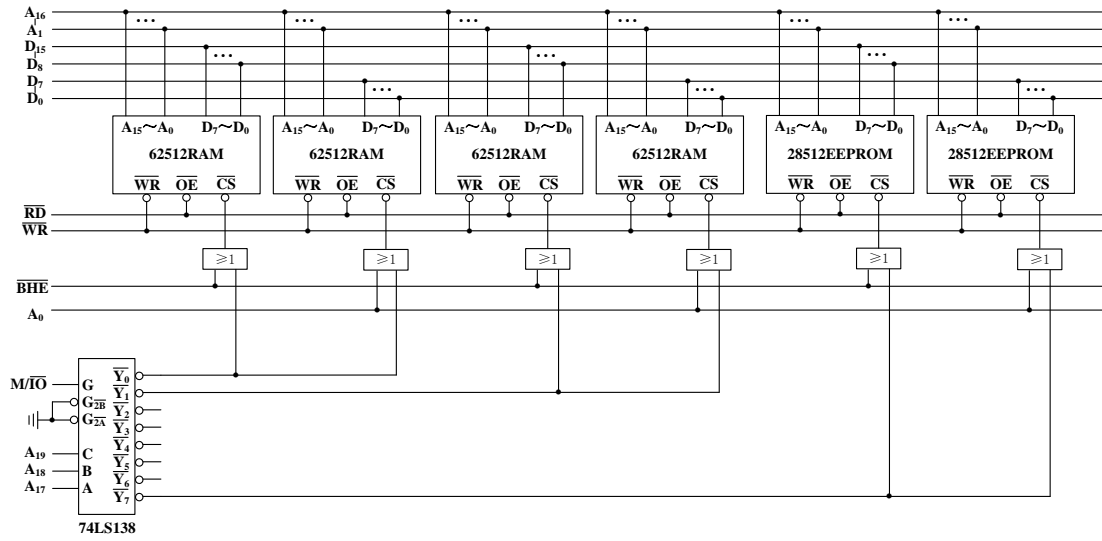
3. 半导体存储器的主要技术指标：存储容量、读/写速度（存取时间和存储周期）、可靠性。

4. 8086CPU 和存储器连接时要考虑：存储芯片数，AB、DB 和一些控制信号的连接，奇、偶片的设置，地址译码方式等。

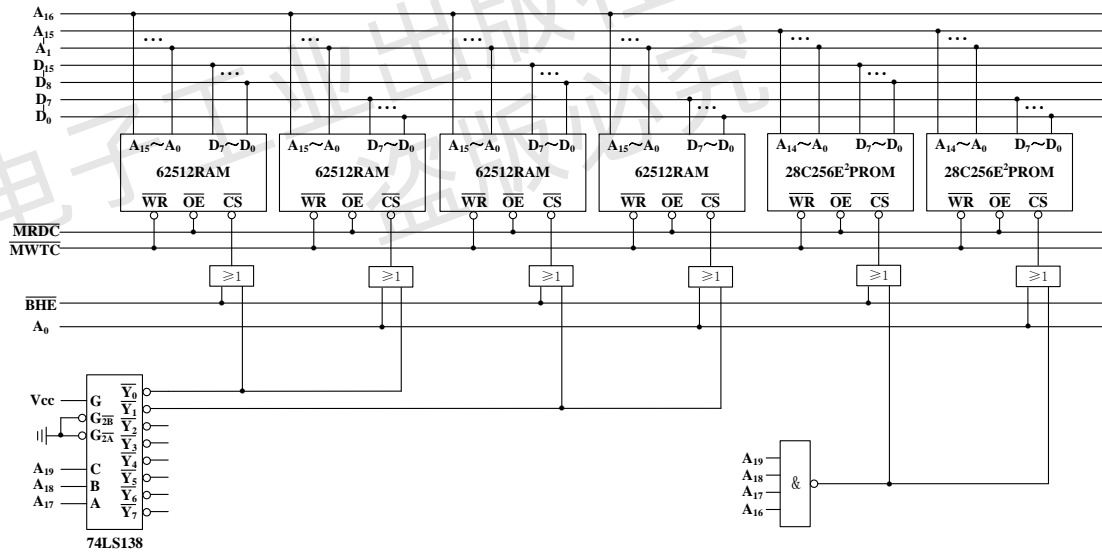
5. 常用的存储器地址译码方式有：全地址译码、部分地址译码、线选法。

6. 需要 $(16k \times 8)/(1024 \times 1) = 128$ 片，CPU 地址线中片内寻址 10 位，片外 4 位。

7.



8.



第7章 习题答案

1. I/O 接口的功能：对输入/输出数据进行缓冲和锁存、对信号的形式和数据的格式进行变换、对 I/O 端口进行寻址、提供联络信号等。

2. CPU 与外设之间的数据传输方式有程序控制方式（无条件传输方式和程序查询方式）、中断方式、DMA、通道控制和 I/O 处理器。

无条件传输方式：CPU 不需要了解外设状态，直接与外设传输数据，适用于按钮开关、发光二极管等简单外设与 CPU 的数据传送过程。

程序查询方式：CPU 与外设传输数据之前，先检查外设状态，如果外设处于“准备好”状态（输入设备）或“空闲”状态（输出设备），才可以传输数据。

中断方式：外设数据准备好之后再通知 CPU，这样，CPU 在没接到外设通知前只管做自己的事情，只有接到通知时才执行与外设的数据传输工作。

DMA：直接存储器存取方式，将外设与内存间建立起直接的通道，CPU 不再直接参加外设与内存间的数据传输，而是在系统需要进行 DMA 传输时，将 CPU 对地址总线、数据总线及控制总线的管理权交由 DMA 控制器进行控制。

通道控制和 I/O 处理器：主 CPU 启动通道工作，通道控制器即从主存或通道存储器中取出相应的程序，控制数据的输入/输出。

3. 每个 I/O 接口内部一般由 3 类寄存器组成，CPU 与外设进行数据传输时，各类信息在接口中进入不同的寄存器，一般称这些寄存器为 I/O 端口。包括数据端口、状态端口、控制端口。端口寻址方式有存储器映像的 I/O 寻址、I/O 映像的 I/O 寻址。80X86 系统采用后者。

4. $\overline{M/\overline{IO}}$ ：低电平，ALE：高电平， \overline{DEN} ：低电平；CPU 从 I/O 接口读取数据时， DT/\overline{R}

为低电平；CPU 往 I/O 接口写入数据时， DT/\overline{R} 为高电平。

5. 在外设接口电路中，经常需要对传输过程中的信息进行锁存或缓冲，所以需要锁存器和缓冲器。

6.

```
DATA SEGMENT AT 2000H
```

```
    ORG 2000H
```

```
    BUF DB 100 DUP(?)
```

```
DATA ENDS
```

```
CODE SEGMENT
```

```
    ASSUME CS:CODE,DS:DATA
```

```
MAIN PROC FAR
```

```
START: MOV AX,DATA
```

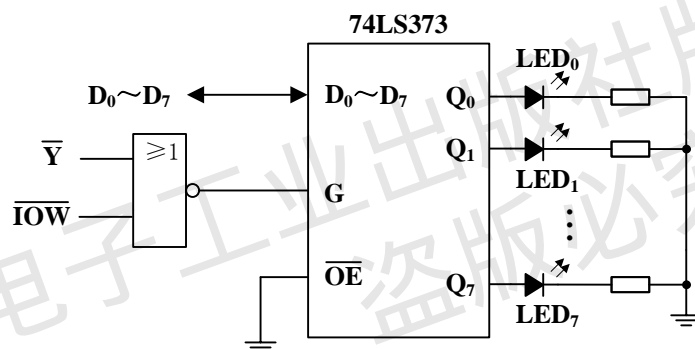
```
        MOV DS,AX
```

```

        LEA SI,BUF
        MOV CX,100
INPUT:  MOV DX,0FFE2H
        IN AL,DX          ;读状态端口
        TEST AL,01H       ;测输入状态 D0 位
        JZ INPUT          ;未“准备好”转 INPUT
        MOV DX,0FFE0H     ;读取输入字符
        IN AL,DX
        MOV [SI],AL       ;输入字符存缓冲区
        INC SI
        LOOP INPUT
        MOV AH,4CH        ;返回 DOS
        INT 21H
CODE    ENDS
        END START

```

7.



```

CODE    SEGMENT
        ASSUME CS:CODE
MAIN    PROC    FAR
START:  PUSH    DS
        MOV     AX,0
        PUSH    AX
        MOV     DX,200H          ;设 I/O 端口为 200H
AGAIN:  MOV     AH,1             ;读键盘缓冲区字符
        INT     16H
        CMP     AL,1BH           ;若为“Esc”键,则退出
        JZ      EXIT
        MOV     AL,0FFH          ;灯全亮
        OUT     DX,AL            ;输出控制 LED
        MOV     BX,5              ;向子程序传递参数,实现 50ms 软延时
        CALL    DELAY            ;子程序 DELAY 实现 10ms 延时
        NOT     AL                ;灯全灭
        MOV     BX,2              ;向子程序传递参数,实现 20ms 软延时
        CALL    DELAY

```

```

JMP AGAIN
EXIT: RET
MAIN ENDP
DELAY PROC NEAR
PUSH CX
WAIT0: MOV CX,2801
WAIT1: LOOP WAIT1
DEC BX
JNZ WAIT0
POP CX
RET
DELAY ENDP
CODE ENDS
END START

```

8.

```

CODE SEGMENT
    ASSUME CS:CODE
MAIN PROC FAR
START: PUSH DS
MOV AX,0
PUSH AX
MOV DX,200H ;设 I/O 端口为 200H
MOV CL,01H ;设置输出初值
AGAIN: MOV AH,1 ;读键盘缓冲区字符
INT 16H
CMP AL,1BH ;若为“Esc”键,则退出
JZ EXIT
MOV AL,CL
OUT DX,AL ;输出控制 LED
MOV BX,50 ;向子程序传递参数,实现 0.5s 软延时
CALL DELAY ;子程序 DELAY 实现 10ms 延时
ROL CL,1 ;循环左移 1 位
JMP AGAIN
EXIT: RET
MAIN ENDP
DELAY PROC NEAR
PUSH CX
WAIT0: MOV CX,2801
WAIT1: LOOP WAIT1
DEC BX
JNZ WAIT0
POP CX
RET

```


DELAY ENDP
CODE ENDS
END START

电子工业出版社版权所有
盗版必究

第8章 习题答案

1. 答：可编程芯片具有灵活的多种工作方式，其工作方式的设置通过软件编程实现。
2. 答：8255A 的内部结构如图 9-1 所示，它由数据总线缓冲器，数据端口 A、端口 B 和端口 C，A 组和 B 组控制电路和读/写控制逻辑 4 部分组成。
3. 答：8255A 有方式 0、方式 1、方式 2 三种工作方式。8255A 方式 0 是基本输入/输出方式。在方式 0 下，每一个端口都可作为基本的输入或输出口，端口 C 的高 4 位和低 4 位以及端口 A、端口 B 都可独立地设置为输入或输出口。方式 1 是单向选通输入/输出方式。8255A 在工作方式 1 时，把 3 个数据端口分为 A、B 两组，分别称为 A 组控制和 B 组控制。此时，端口 A 和端口 B 仍作为数据的输入或输出口，而端口 C 作为联络控制信号，被分成两部分，一部分作为端口 A 和端口 B 的联络信号，另一部分仍可作为基本的输入/输出口。8255A 方式 2 是双向选通输入/输出方式，只有端口 A 可以工作于方式 2。8255A 端口 A 工作于方式 2 时，可使 8255A 与外设进行双向通信，可采用查询方式或中断方式进行传输。
4. 答：通过 JMP AGAIN 指令实现循环检测。
5. 答：软件定时和硬件定时。软件定时是通过执行一段循环程序来实现的。通过调整循环次数可以控制时间间隔的长短。其特点是：不需要专用硬件电路、成本低，但是耗费 CPU 的时间，降低了 CPU 的工作效率。硬件定时是采用定时/计数器或单稳延时电路实现的。其特点是：定时时间长、使用灵活而且不占用 CPU 的时间，适用范围广。
6. 答：由于 8253 的读/写操作对系统时钟没有特殊的要求，因此它几乎可以应用于任何一种微处理器系统中，可作为可编程的方波频率发生器、分频器、实时时钟、事件计数器或单脉冲发生器等。每片 8253 定时/计数器有 3 个独立的 16 位计数通道，每个计数通道最高计数速率可达 2.6MHz。每个计数器可编程设定 6 种工作方式，使用时可以根据需要选择其中的一种工作方式。每个计数通道可按二进制或十进制来计数。
7. 答：8253 的每个计数器都有 6 种工作方式：方式 0~方式 5。采用方式 0 工作方式，8253 可完成计数功能，且计数器只计一遍。方式 1 是硬件触发单稳态方式，采用这种工作方式可在输出端 OUT 输出单个负脉冲信号，脉冲的宽度可通过编程来设定。采用方式 2 可产生连续的负脉冲信号，可用作频率发生器。采用方式 3 可产生连续的方波信号，可用作方波发生器。方式 4 是软件触发的选通方式。采用方式 4 可产生单个负脉冲信号，负脉冲宽度为一个时钟周期，由 OUT 指令触发。方式 5 是硬软件触发的选通方式。采用方式 5 可产生单个负脉冲信号，负脉冲宽度为一个时钟周期。由 GATE 的上升沿触发。
8. 答：8253 在初始化编程时，必须遵守两条原则：
- ① 在对某个计数器设置初值之前，必须先写入控制字；
 - ② 在设置计数初始值时，要符合控制字的规定，即只写低位字节，还是只写高位字节，还是高、低位字节都写（分两次写，先低位字节后高字节）。
9. 答：8253 初始化编程步骤是：先写控制字到 8253 的控制端口，再写计数器初值到相应的计数器端口。

10. 答: (1) MOV DX,203H

MOV AL,50H

OUT DX,AL

MOV DX,201H

MOV AL,120

OUT DX,AL

(2) MOV DX,203H

MOV AL,33H ;00110011B

OUT DX,AL

MOV DX,200H

MOV AX,2011H

OUT DX,AL

MOV AL,AH

OUT DX,AL

(3) MOV DX,203H

MOV AL,B4H ;10110100B

OUT DX,AL

MOV DX,202H

MOV AX,0F050H

OUT DX,AL

MOV AL,AH

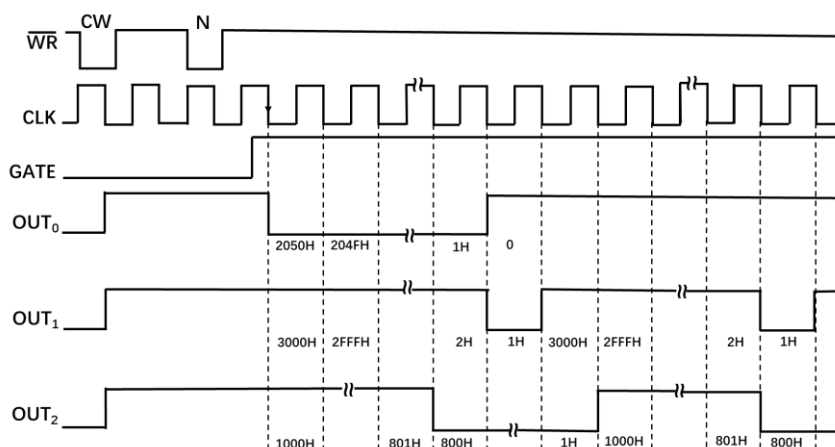
OUT DX,AL

11. 答:

计数器 0 工作于方式 1, 在 GATE 上升沿之后的下一个 CLK 脉冲的下降沿, OUT₀ 变为低电平, 计数器开始减 1 计数。当计数值减到 0 时, OUT₀ 变为高电平。OUT₀ 输出为单个负脉冲, 负脉冲宽度为 2050H 个时钟周期。

计数器 1 工作于方式 2, 若 GATE 为高电平, 在下一个 CLK 的下降沿, 计数器开始减 1 计数。当计数值减到 1 时, 输出端 OUT₁ 输出低电平, 经过一个时钟周期, OUT₁ 输出高电平, 并自动开始一个新的计数过程。输出为连续负脉冲, 负脉冲宽度为 1 个时钟周期。

计数器 2 工作于方式 3, 若 GATE 为高电平, 在下一个 CLK 的下降沿, 计数器开始减 1 计数。当计数值减到一半时, OUT₂ 输出低电平, 计数器继续减 1 计数, 计数到 0 时, OUT₂ 变为高电平, 并自动开始一个新的计数过程。因为初值 1000H 为偶数, 所以输出为对称方波。



12. 答: 参考程序段:

```
MOV AL,31H ;方式控制字 00110001B=31H
MOV DX,406H
OUT DX,AL
MOV DX,400H
MOV AX,1000H
OUT DX,AL
MOV AL,AH
OUT DX,AL
Lea bx, VALUE
MOV AL,01H
MOV DX,406H
OUT DX,AL
MOV DX,400H
IN AL, DX
Mov cl,al
IN AL, DX
Mov ch,al
Mov [bx],cx
Inc bx
Inc bx
Cmp cx,0
Jnz lop1
.....
```

13. 答: 参考程序段:

.....

```
Mov al ,90H
Mov dx,203H
Out dx,al
again:mov dx,200H
in al,dx
test al,03h ;
jz exit
test al,01h
jz diap_1
test al,02h
jz diap_2
jmp again
diap_2:mov al,06h
mov dx,201H
out dx,al
jmp again
```

```
diap_1:mov al,3FH  
mov dx,201h  
out dx,al  
jmp again  
exit:  
.....
```

14. 答：根据在串行通信中数据定时和同步的不同，串行通信方式分为同步串行通信和异步串行通信。

同步传送要求对传送的每一位在收发两端保持严格同步，发送端、接收端使用同一时钟源以保证同步。

异步串行通信中的异步是指发送端和接收端不使用共同的时钟，也不在数据中传送同步信号，但接收方与发送方之间必须约定传送数据的帧格式和波特率。

15. 答：双工通信方式，两设备之间有两根传输线，对于每一个设备来讲都有专用的一条发送线和一条专用的接收线，实现双向数据传送。

半双工通信方式数据可从设备 1 发送到设备 2，此时设备 1 是发送方，设备 2 是接收方；也可以从设备 2 发送到设备 1，此时设备 2 是发送方，设备 1 是接收方。但某一设备不能同时接收和发送数据。

单工方式，数据只能向单一方向传送，只能从设备 1 发送到设备 2，设备 1 是发送方，设备 2 是接收方。

电子工业出版社版权所有
盗版必究

16. 答：每秒能传输的最大字符数为： $1200 / (1+7+1+1) = 120$ 字符/s

17. 答：至少需要时间为： $2000 * (1+8+1) / 9600 \approx 2.1$ 秒

18. 答：初始化程序：

```
MOV AL,0DAH
```

```
OUT 0C2H,AL ;设置方式选择命令字
```

```
MOV AL,37H
```

```
OUT 0C2H,AL ;
```

电子工业出版社版权所有
盗版必究

第9章 习题答案

1.答：采用中断方式。因为采用中断方式时，CPU 可以进行其它的工作，只在外设准备好数据发出中断请求之后，才予以响应，暂时中断当前工作转去执行传送数据，传送完成后又返回执行原来的程序，这样就大大地提高了计算机系统的效率。

2.答：计算机在执行正常程序的过程中，系统内发生任何非寻常的或非预期的急需处理事件，处理器就暂时中断正在执行的正常程序，而转去执行相应的事件处理程序。当处理完毕后，CPU 再返回到被暂时中断的程序，接着往下继续执行，这个过程称为中断。

产生中断请求的设备或事件称为“中断源”。从中断源来看，中断一般可分为三类：

- ① 由计算机硬件异常或故障引起的中断，也称为内部异常中断。
- ② 由程序中执行了中断指令引起的中断，也称为软中断。
- ③ 外部设备（如输入输出设备）请求引起的中断，也称为外部中断或 I/O 中断。

3.答：中断技术是十分重要而复杂的技术，由计算机的软、硬件协同完成，称之为中断系统。一个完整的中断系统应具备如下功能：设置中断源、中断源识别、中断源判优、中断嵌套和中断处理与返回。

4.答：Intel 80X86 系列微机有一个灵活的中断系统，中断可分为外部中断（或硬件中断）和内部中断（或软件中断）两类。可以处理 256 种中断源，每个中断源都有对应的中断类型码（0~255）供 CPU 识别。

5.答：当外设通过屏蔽中断请求信号向微处理器提出中断请求时，微处理器在当前指令执行结束后，判断有无中断请求，CPU 对可屏蔽中断请求的响应是有条件的，它受中断允许标志位 IF 的控制。当 IF=1 时，允许 CPU 响应 INTR 请求；当 IF=0 时，禁止 CPU 响应 INTR 请求。

8086 系统把中断向量表安排在内存地址 00000H~003FFH 区域（1K），每四个连续字节存放一个中断向量，其中高地址 2 个字节单元放段地址（CS），低地址 2 个字节单元中放偏移地址（IP）。

6.答：当外设通过非屏蔽中断请求信号向微处理器提出中断请求时，微处理器在当前指令执行结束后，就立即无条件的予以响应，这样的中断就是非屏蔽中断。由 NMI 引脚出现上升沿触发，高电平维持 2 个 T。不受中断允许标志 IF 的限制，中断类型号固定为 2。

可屏蔽中断由外部设备产生，从 CPU 的 INTR 端引入高电平触发。CPU 对可屏蔽中断请求的响应是有条件的，它受中断允许标志位 IF 的控制。当 IF=1 时，允许 CPU 响应 INTR 请求；当 IF=0 时，禁止 CPU 响应 INTR 请求。可屏蔽中断用于 CPU 与外设进行数据交换。CPU 响应时送出两个 INTA 脉冲，外设则送出中断类型码到 CPU。

7.答：1) 用传送指令直接装入

CLI	; IF=0, 关中断
MOV AX, 0	; ES 指向 0 段
MOV ES, AX	
MOV BX, 0AH*4	; 向量表地址送 BX


```

MOV  AX, OFFSET  INT_PA      ; 中断服务子程序的偏移地址送 AX
MOV  ES:WORD PTR[BX], AX    ; 中断服务子程序的偏移地址写入向量表
MOV  AX, SEG INT_PA      ; 中断服务子程序的段基址送 AX
MOV  ES:WORD PTR[BX+2], AX  ; 中断服务子程序的段基址写入向量表

STI                                ; IF=1, 开中断
...

```

2) 用 DOS 系统功能调用装入法

```

CLI                                ; IF=0, 关中断
MOV  AL, 0AH                      ; 中断类型号 40H 送 AL
MOV  DX, SEG INT_PA               ; 中断服务子程序的段基址送 DS
MOV  DS, DX
MOV  DX, OFFSET  INT_PA          ; 中断服务子程序的偏移地址送 DX
MOV  AH, 25H                     ; 25H 功能调用
INT  21H
STI                                ; IF=1, 开中断

```

8.答: CPU 在响应中断后, 都要保护现场和断点, 然后转入相应的中断服务程序。因此, 中断操作要解决的一个首要问题就是: 找到与中断源相对应的中断服务子程序的入口地址。在 8086 系统中, 是通过采用中断类型号和中断向量的办法来解决这个问题。对于可屏蔽中断, 在中断响应周期的第二个总线周期中, 由中断控制器给出中断服务子程序的入口地址 (即将中断处理程序所在段的段地址及第一条指令的有效地址), 装入 CS 与 IP 两个寄存器。一旦装入完毕, 中断服务程序就开始执行。

9.答: CPU 响应中断后, 发出中断响应信号的同时将状态标志寄存器 FR 的内容压入堆栈保护起来, 然后将 FR 中的中断标志位 IF 与陷阱标志位 TF 清零, 从而自动关闭外部硬件中断, 以避免在中断过程中或进入中断服务程序后受到其他中断源的干扰。只有中断处理程序中出現开中断指令 STI 时, 才一允许 CPU 接收其他设备的中断请求。

10.答: Intel 8259A 的主要功能有:

- ① 可管理具有 8 级优先权的中断源, 通过级联可扩充至管理 64 级优先权的中断源。
- ② 可通过编程对每一级中断源都可实现屏蔽或允许。
- ③ 在中断响应周期, 8259A 能向 CPU 提供相应的中断向量, 从而能迅速地转至中断服务程序。

11.答: 边沿触发方式和电平触发两种触发方式

12 答: 单片 8259A 能管理 8 级可屏蔽中断, 8 片级联可管理 64 级可屏蔽中断。

13 答: 8259A 的初始化编程是在中断系统进入正常运行之前, 通过设置初始化命令字 Initialization Command Word (ICW)来预置 8259A 的工作方式。

14.答: 8259A 的初始化命令字 ICW 用于预置 8259A 的工作方式, 而操作命令字 OCW 则用于实现 8259A 运行中的操作控制, OCW 可以在 8259A 被初始化之后的任何时候被使用。

15.答：中断向量表的入口地址为：20H *4=0080H

中断向量表的内容为：

0000:0080H	0000H
0000:0082H	143FH

16.答：

MOV AL, 00010011B

OUT 80H, AL

MOV AL, 00100000B

OUT 81H, AL

MOV AL, 00000011B

OUT 81H, AL

电子工业出版社版权所有
盗版必究

第10章 习题答案

1.答: DAC0832 的工作方式有 3 种: (1) 直通方式 (2) 单缓冲方式 (3) 双缓冲方式。

2.答: DAC0832 是 CMOS 工艺制成的 8 位双缓冲型 D/A 转换器, 其逻辑电平与 TTL 电平相兼容。内部阶梯电阻网络形成参考电流, 由输入二进制数控制 8 个电流开关, CMOS 的电流开关漏电很小保证了转换器的精度。DAC0832 使用单一电源, 功耗低, 建立时间为 $1\mu\text{s}$ 。输入数据为 8 位并行输入, 有两级数据缓冲器及使能信号、数据锁存信号等, 与 CPU 接口方便。

3.答: D/A 转换器在微机控制系统中作用: 将计算机处理后的数字量结果转换成模拟量, 然后控制执行部件; 和 A/D 转换器在微机控制系统中作用: 将工业设备的模拟量转换成数字量便于计算机计算和处理。

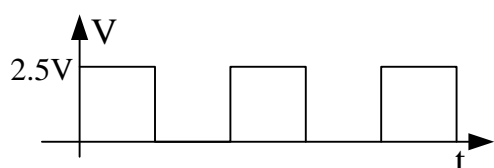
4.答: 模拟量是连续的信号, 采样可以将时间上连续变化的信号, 转换为时间上离散的信号, 即将时间上连续变化的模拟量转换为一系列等间隔的脉冲, 脉冲的幅度取决于输入模拟量的大小。

5.答: 分辨率; 转换精度; 建立时间; 线性度; 温度系数; 输入代码; 输出电平

6.答: 量程; 分辨率; 量化误差; 转换误差; 转换精度; 转换时间

7.答:

```
MOV    DX, 24C0H
AGANT:
MOV     AL, 128                ;方波高电平
OUT     DX, AL                ;转换数据送 D/A 的数据口
CALL    DELAY                 ;调用延时子程序段,也可用几条 NOP 指令
MOV     AL, 0                  ;方波低电平
OUT     DX, AL                ;转换数据送 D/A 的数据口
CALL    DELAY
JMP     AGANT
DELAY   PROC                  ;软件延时子程序
MOV     CX, 10
DELAY1: LOOP DELAY1
RET
DELAY   ENDP
```

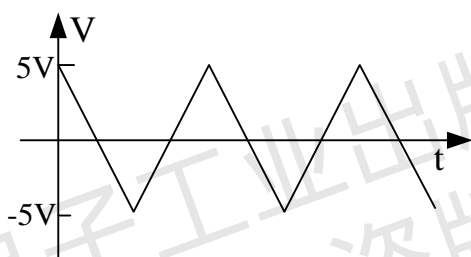


8.答:

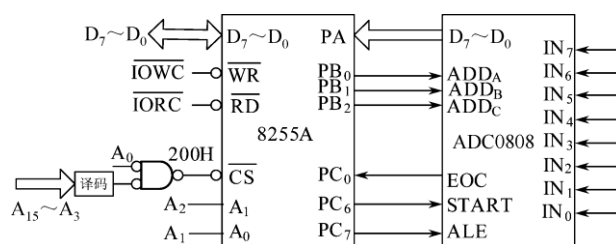
```

L1: MOV    CX, 256      ;256 个数字量
    MOV    AL, 0        ;初值
    MOV    DX, 24C0H    ;24C0H 为 0832 的端口地址
L2:  OUT    DX, AL      ;三角波形上升段
    CALL   DELAY        ;调用延时子程序
    INC    AL
    LOOP   L2
    MOV    CX, 256
    MOV    AL, 255      ;初值
    MOV    DX, 400H
L3:  OUT    DX, AL      ;三角波形下降段
    CALL   DELAY
    DEC    AL
    LOOP   L3
    JMP    L1

```



9、答:



DATA SEGMENT

BUF DB 20 DUP(?) ;预留 20 字节空间,存放采样后结果

DATA ENDS

CODE SEGMENT

ASSUME CS:CODE,DS:DATA

START: MOV AX,DATA

MOV DS,AX

MOV AL,10010001B ;8255A 初始化

MOV DX,206H

OUT DX,AL

MOV AL,00H

MOV DX,204H

```

        OUT  DX,AL                ;START、ALE=0
        MOV  BX,OFFSET BUF        ;BUF 是数据区首地址
        MOV  CX,20                ;CX 中是采样次数
        MOV  AL,05H
        MOV  DX,202H
        OUT  DX,AL                ;通过 PB2~PB0 选中采样通道 IN5
AGAIN:   MOV  AL,0FH
        MOV  DX,206H
        OUT  DX,AL                ;通过 PC7 使 ALE=1
        MOV  AL,0DH
        MOV  DX,206H
        OUT  DX,AL                ;通过 PC6 使 START=1
        MOV  AL,00H
        MOV  DX,204H                ;START、ALE=0
WAIT0:   IN   AL,DX                ;循环检测 PC 口的 PC0(即 EOC 信号)
        AND  AL,01H
        JNZ  WAIT0                ;若 EOC 为低,则开始转换
WAIT1:   IN   AL,DX                ;继续循环检测 PC 口的 PC0(即 EOC 信号)
        AND  AL,01H
        JZ   WAIT1                ;若 EOC 为高,则转换结束,可以读数据
        MOV  DX,200H
        IN   AL,DX                ;从 PA 口输入数据
        MOV  [BX],AL              ;存入内存
        INC  BX
        LOOP AGAIN                ;循环 20 次采样
        RET
CODE  ENDS
      END  START

```

第11章 习题答案

略。