

1.1 计算机中常用的计数制有哪些?

解: 二进制、八进制、十进制(BCD)、十六进制。

1.2 什么是机器码? 什么是真值?

解: 把符号数值化的数码称为机器数或机器码, 原来的数值叫做机器数的真值。

1.3 完成下列数制的转换。

(1) $10100110\text{B} = (\quad)\text{D} = (\quad)\text{H}$

(2) $0.11\text{B} = (\quad)\text{D}$

(3) $253.25 = (\quad)\text{B} = (\quad)\text{H}$

(4) $1011011.101\text{B} = (\quad)\text{H} = (\quad)\text{BCD}$

解:

(1) 166, A6H

(2) 0.75

(3) 11111101.01B, FD.4H

(4) 5B.AH, (10010001.011000100101)BCD

1.4 8位和16位二进制数的原码、补码和反码可表示的数的范围分别是多少?

解:

原码 $(-127 \sim +127)$ 、 $(-32767 \sim +32767)$

补码 $(-128 \sim +127)$ 、 $(-32768 \sim +32767)$

反码 $(-127 \sim +127)$ 、 $(-32767 \sim +32767)$

1.5 写出下列真值对应的原码和补码的形式。

(1) $X = -1110011\text{B}$

(2) $X = -71\text{D}$

(3) $X = +1001001\text{B}$

解:

(1) 原码: 11110011 补码: 10001101

(2) 原码: 11000111 补码: 10111001

(3) 原码: 01001001 补码: 01001001

1.6 写出符号数 10110101B 的反码和补码。

解: 11001010, 11001011

1.7 已知 X 和 Y 的真值, 求 $[X+Y]$ 的补码。

(1) $X = -1110111B$ $Y = +1011010B$

(2) $X = 56D$ $Y = -21D$

解:

(1) 11100011

(2) 00100011

1.8 已知 $X = -1101001B$, $Y = -1010110B$, 用补码求 $X-Y$ 的值。

解: 11101101

1.9 请写出下列字符的 ASCII 码。

4A3=!

解: 34H, 41H, 33H, 3DH, 21H

1.10 若给字符 4 和 9 的 ASCII 码加奇校验, 应是多少?

解: 34H, B9H

1.11 上题中若加偶校验, 结果如何?

解: B4H, 39H

1.12 计算下列表达式。

(1) $(4EH + 10110101B) \times (0.0101)BCD = (\quad)D$

(2) $4EH - (24/08H + 'B'/2) = (\quad)B$

解:

(1) 129.5D

(2) 101010B

2.1 简述微型计算机的硬件系统结构？说明各部件的主要功能。

解：微型计算机的硬件系统主要包括以下几个部分：

中央处理器——包括控制器、运算器、寄存器组。主要功能是执行指令并根据指令发出相应的控制信号，以使各微机各部件协调工作。此外还完成各种算术逻辑运算功能。

存储器——包括 RAM 和 ROM。主要功能是存放当前运行的程序和数据。

I/O 接口——在外部设备与主机之间实现数据信息、控制信息和状态信息的缓存、变换、传送以及信号电平、速度的匹配等功能。

外部设备——主要实现人机交互(信息的输入输出)。

总线——把微机中各部件连接在一起的公共信息传输通道。

2.2 简述存储程序计算机的工作原理。

解：把要运行的程序和数据预先送到存储器中保存，开始工作时给出程序的第一条指令的地址，然后控制器根据存储器中的指令顺序周而复始地取出指令、分析指令、执行指令，直到按照程序的控制流程执行完全部所需执行的指令为止。

2.3 微型计算机采用总线结构的优点有哪些？

解：采用总线结构的优点在于设计简单、灵活性好、易于扩展、便于故障检测和维修，更易于标准化，使得部件制造成本大幅度降低。

2.4 典型的微机中有哪几种总线？它们各自传送什么类型的信息？

解：数据总线 DB——传输数据信息；地址总线 AB——传输存储器地址和 I/O 地址；控制总线 CB——传输控制信息和状态信息。

2.5 系统软件与应用软件的区别是什么？

解：系统软件主要用于微机软硬件资源的管理、调度，控制任务的运行，实现人机接口等。应用软件主要用于解决各种具体的实际应用问题(如办公软件、信息管理系统、游戏软件等)。

2.6 简述 CPU 执行程序的过程。

解：当程序的第一条指令所在的地址送入程序计数器后，CPU 就进入取指阶段准备取第一条指令。在取指阶段，CPU 从内存中读出指令，并把指令送至指令寄存器 IR 暂存。在取指阶段结束后，机器就进入执行阶段，这时，由指令译码器对指令译码，再经控制器发出相应的控制信号，控制各部件行指令所规定的具体操作。当一条指令执行完毕以

后,就转入了下一条指令的取指阶段。以上步骤周而复始地循环,直到遇到停机指令。

2.7 说明 8086 的 EU 和 BIU 的主要功能。在执行程序过程中它们是如何相互配合工作的?

解: 执行单元 EU 负责执行指令。EU 在工作时不断地从指令队列取出指令代码,对其译码后产生完成指令所需要的控制信息。数据在 ALU 中进行运算,运算结果的特征保留在标志寄存器 FLAGS 中。总线接口单元 BIU 负责 CPU 与存储器、I/O 接口之间的信息传送。BIU 取出的指令被送入指令队列供 EU 执行,BIU 取出的数据被送入相关寄存器中以便做进一步的处理。

当 EU 从指令队列中取走指令,指令队列出现空字节时,BIU 就自动执行一次取指令周期,从内存中取出后续指令代码放入队列中。当 EU 需要数据时,BIU 根据 EU 给出的地址,从指定的内存单元或外设中取出数据供 EU 使用。当运算结束时,BIU 将运算结果送入指定的内存单元或寄存器。当指令队列空时,EU 就等待,直到有指令为止。若 BIU 正在取指令,EU 发出访问总线的请求,则必须等 BIU 取指令完毕后,该请求才能得到响应。一般情况下,程序顺序执行,当遇到跳转指令时,BIU 就使指令队列复位,从新地址取出指令,并立即传给 EU 去执行。

指令队列的存在使 8086/8088 的 EU 和 BIU 并行工作,从而减少了 CPU 为取指令而等待的时间,提高了 CPU 的利用率,加快了整机的运行速度。另外也降低了对存储器存取速度的要求。

2.8 在执行指令期间,EU 能直接访问存储器吗?为什么?

解: 可以。因为 EU 和 BIU 可以并行工作,EU 需要的指令可以从指令队列中获得,这是 BIU 预先从存储器中取出并放入指令队列的。在 EU 执行指令的同时,BIU 可以访问存储器取下一条指令或指令执行时需要的数据。

2.9 8086 与 8088 CPU 的主要区别有哪些?

解: 主要区别有以下几点:

- ① 8086 的外部数据总线有 16 位,而 8088 的外部数据总线只有 8 位。
- ② 8086 的指令队列深度为 6 个字节,而 8088 的指令队列深度为 4 个字节。
- ③ 因为 8086 的外部数据总线有 16 位,故 8086 每个总线周期可以存取两个字节。而 8088 的外部数据总线因为只有 8 位,所以每个总线周期只能存取 1 个字节。

① 个别引脚信号的含义稍有不同。

2.10 8088 CPU 工作在最小模式下:

- (1) 当 CPU 访问存储器时,要利用哪些信号?
- (2) 当 CPU 进行 I/O 操作时,要利用哪些信号?
- (3) 当 HOLD 有效并得到响应时,CPU 的哪些信号置高阻?

解:

- (1) 要利用信号线包括 $\overline{WR}\#$ 、 $\overline{RD}\#$ 、 $\overline{IO}/\overline{M}\#$ 、ALE 以及 $AD0\sim AD7$ 、 $A8\sim A19$ 。
- (2) 同(1)。
- (3) 所有三态输出的地址信号、数据信号和控制信号均置为高阻态。

2.11 总线周期中,什么情况下要插入 T_w 等待周期? 插入 T_w 周期的个数,取决于

什么因素?

解: 在每个总线周期的 T_3 的开始处若 $READY$ 为低电平, 则 CPU 在 T_3 后插入一个等待周期 T_w 。在 T_w 的开始时刻, CPU 还要检查 $READY$ 状态, 若仍为低电平, 则再插入一个 T_w 。此过程一直进行到某个 T_w 开始时, $READY$ 已经变为高电平, 这时下一个时钟周期才转入 T_4 。

可以看出, 插入 T_w 周期的个数取决于 $READY$ 电平维持的时间。

2.12 若 8088 工作在单 CPU 方式下, 在教材第 91 页的表中填入不同操作时各控制信号的状态。

解: 结果如表 1-2-1 所示。

表 1-2-1

操 作	IO/\overline{M}	DT/R	\overline{DEN}	\overline{RD}	\overline{WR}
读存储器	0	0	0	0	1
写存储器	0	1	0	1	0
读 I/O 接口	1	0	0	0	1
写 I/O 接口	1	1	0	1	0

2.13 在 8086/8088 CPU 中, 标志寄存器包含哪些标志位? 各位为 0(为 1)分别表示什么含义?

解: 标志寄存器包含以下标志位:

CF 进位标志位。若算术运算时最高位有进(借)位则 $CF=1$, 否则 $CF=0$ 。

PF 奇偶标志位。当运算的结果低 8 位中“1”的个数为偶数时 $PF=1$, 为奇数时 $PF=0$ 。

AF 辅助进位位。在加(减)法操作中, b3 向 b4 有进位(借位)时, $AF=1$, 否则 $AF=0$ 。

ZF 零标志位。当运算结果为零时 $ZF=1$, 否则 $ZF=0$ 。

SF 符号标志位。当运算结果的最高位为 1 时 $SF=1$, 否则 $SF=0$ 。

OF 溢出标志位。当算术运算的结果溢出时, $OF=1$, 否则 $OF=0$ 。

TF 跟踪标志位。TF=1 时, 使 CPU 处于单步执行指令的工作方式。

IF 中断允许标志位。IF=1 使 CPU 可以响应可屏蔽中断请求。IF=0 时则禁止响应中断。

DF 方向标志位。DF=1 使串操作按减地址方式进行。DF=0 使串操作按增地址方式进行。

2.14 8086/8088 CPU 中, 有哪些通用寄存器和专用寄存器? 说明它们的作用。

解: 通用寄存器包含以下 8 个寄存器:

AX、BX、CX 和 DX 寄存器一般用于存放参与运算的数据或运算的结果。除此之外:

AX: 主要存放算术逻辑运算中的操作数, 以及存放 I/O 操作的数据。

BX: 存放访问内存时的基地址。

CX: 在循环和串操作指令中用作计数器。



DX: 在寄存器间接寻址的 I/O 指令中存放 I/O 地址。在做双字长乘法运算时, DX 与 AX 合起来存放一个双字长数。

SP: 存放栈顶偏移地址。

BP: 存放访问内存时的基地址。

SP 和 BP 也可以存放数据,但它们的默认段寄存器都是 SS。

SI: 常在变址寻址方式中作为源地址指针。

DI: 常在变址寻址方式中作为目标地址指针。

专用寄存器包括 4 个段寄存器和两个控制寄存器:

CS: 代码段寄存器,用于存放代码段的段基地址。

DS: 数据段寄存器,用于存放数据段的段基地址。

SS: 堆栈段寄存器,用于存放堆栈段的段基地址。

ES: 附加数据段寄存器,用于存放附加段的段基地址。

IP: 指令指针寄存器,用于存放下一条要执行指令的偏移地址。

FLAGS: 标志寄存器,用于存放运算结果的特征。

2.15 8086/8088 系统中,存储器为什么要分段? 一个段最大为多少字节? 最小为多少字节?

解: 分段的主要目的是便于存储器的管理,使得可以用 16 位寄存器来寻址 20 位的内存空间。一个段最大为 64KB,最小为 16B。

2.16 在 8086/8088 CPU 中,物理地址和逻辑地址是指什么? 已知逻辑地址为 1F00:38A0H,如何计算出其对应的物理地址?

解: 物理地址是 CPU 存取存储器所用的地址。逻辑地址是段和偏移形式的地址,即汇编语言程序中使用的存储器地址。

若已知逻辑地址为 1F00:38A0H,则对应的物理地址 $= 1F00 \times 16 + 38A0 = 228A0H$ 。

2.17 已知存储器物理地址为 78A00H,计算它所对应的逻辑地址。此结果惟一吗?

解: 物理地址可以对应于不同的逻辑地址。78A00H 对应的逻辑地址可以是 7000H: 8A00H,7800H:0A00H,78A0H:0000H 等。结果不是惟一的。

2.18 设当前数据段位于存储器的 A8000H~B7FFFH,DS 段寄存器的内容应是什么?

解: 因为 A8000H 到 B7FFFH 之间的地址范围大小为 64KB,未超过一个段的最大范围。故要访问此地址范围的数据,数据段的起始地址(即段首地址)应为 A8000H,则 DS 段寄存器为 A800H。

2.19 若 CS=8000H,则当前代码段可寻址的存储空间的范围是多少?

解: (CS)=8000H 时,当前代码段可寻址的存储空间范围为 80000H~8FFFFH。

2.20 8086/8088 CPU 在最小模式下构成计算机系统至少应包括哪几个基本部分(器件)?

解: 其至少应包括: 8088CPU、8284 时钟发生器、8282 锁存器(3 片)和 8286 双向总线驱动器。

2.21 总线传输需要解决哪几个主要问题?

解：总线传输需要解决以下几方面的问题：

总线传输同步。使信息发送部件与信息接收部件在规定的时刻交换数据。

总线仲裁控制。避免多个部件同时发送信息到总线而产生冲突。

出错处理。防止数据传送过程中可能产生的错误。

总线驱动。提供驱动总线所需的电压和电流。

2.22 在教材第 59 页的图 2-28 中,若设备接口 0 和设备接口 1 同时申请总线,哪一个设备接口将最先获得总线控制权?为什么?

解:设备接口 0 先获得总线控制权。因为设备接口 0 将截获总线回答信号 BG,使 BG 不会传送到设备接口 1。

2.23 在南北桥结构的 80x86 系统中,PCI 总线是通过什么电路与 CPU 总线相连的?ISA 总线呢?

解:PCI 总线通过北桥芯片与 CPU 总线相连,ISA 总线则通过南桥芯片与 PCI 总线相连。

2.24 80386 CPU 包含哪些寄存器?各有什么主要用途?

解:80386 共有 7 类 34 个寄存器。它们分别是通用寄存器、指令指针和标志寄存器、段寄存器、系统地址寄存器、控制寄存器、调试和测试寄存器。

(1) 通用寄存器(8 个)。

EAX,EBX,ECX,EDX,ESI,EDI,EBP 和 ESP。每个 32 位寄存器的低 16 位可单独使用,同时 AX,BX,CX,DX 寄存器的高、低 8 位也可分别当作 8 位寄存器使用。它们与 8088/8086 中相应的 16 位通用寄存器作用相同。

(2) 指令指针和标志寄存器。

指令指针 EIP 是一个 32 位寄存器,存放下一条要执行的指令的偏移地址。

标志寄存器 EFLAGS 也是一个 32 位寄存器,存放指令的执行状态和一些控制位。

(3) 段寄存器(6 个)。

CS,DS,SS,ES,FS 和 GS。在实方式下,它们存放内存段的段地址。在保护方式下,它们被称为段选择符。其中存放的是某一个段的选择符。当选择符装入段寄存器时,80386 中的硬件会自动用段寄存器中的值作为索引从段描述符表中取出一个 8 个字节的描述符,装入到与该段寄存器相应的 64 位描述符寄存器中。

(4) 控制寄存器(4 个)。

CR0,CR1,CR2 和 CR3。它们的作用是保存全局性的机器状态。

(5) 系统地址寄存器(4 个)。

GDTR,IDTR,LDTR 和 TR。它们用来存储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。

(6) 调试寄存器(8 个)。

DR 0~DR 7。它们为调试提供硬件支持。

(7) 测试寄存器(8 个)。

TR 0~TR 7,其中 TR 0~TR 5 由 Intel 公司保留,用户只能访问 TR 6、TR 7。它们用于控制对 TLB 中的 RAM 和 CAM 相连存储器的测试。TR 6 是测试控制寄存器,TR 7



是测试状态寄存器,保存测试结果的状态。

2.25 什么是实地址模式?什么是保护模式?它们的特点是什么?

解:实地址模式是与 8086/8088 兼容的存储管理模式。当 80386 加电或复位后,就进入实地址工作模式。物理地址形成与 8088/8086 一样,是将段寄存器内容左移 4 位与有效偏移地址相加而得到,寻址空间为 1MB。

保护地址模式又称为虚拟地址存储管理方式。在保护模式下,80386 提供了存储管理和硬件辅助的保护机构,还增加了支持多任务操作系统的特别优化的指令。保护模式采用多级地址映射的方法,把逻辑地址映射到物理存储空间中。这个逻辑地址空间也称为虚拟地址空间,80386 的逻辑地址空间提供 2^{46} 的寻址能力。物理存储空间由内存和外存构成,它们在 80386 保护地址模式和操作系统的支持下为用户提供了均匀一致的物理存储能力。在保护模式下,用段寄存器的内容作为选择符(段描述符表的索引),选择符的高 13 位为偏移量,CPU 的 GDTR 中的内容作为基地址,从段描述符表中取出相应的段描述符(包括 32 位段基地址、段界限和访问权等)。该描述符被存入描述符寄存器中。描述符中的段基地址(32 位)与指令给出的 32 位偏移地址相加得到线性地址,再通过分页机构进行变换,最后得到物理地址。

2.26 80386 以上的 CPU 中,选择符中 3 个域的名字分别叫什么?大小各是多少位?各有什么作用?

解:称为 INDEX、TI 和 RPL。INDEX 有 13 位(bit15~bit3),它表示了描述符在描述符表中的位置。TI 有 1 位,它用于指示 INDEX 指向全局描述符表 GDT 还是局部描述符表 LDT。RPL 有 2 位,它定义了当前地址访问请求的特权级别。

2.27 80386 访问存储器有哪两种方式?各提供多大的地址空间?

解:实模式和保护模式。实模式可提供 1MB(2^{20})的寻址空间。保护模式可提供 4GB(2^{32})的线性地址空间和 64TB(2^{46})的虚拟存储器地址空间。

2.28 如果 GDT 寄存器值为 0013000000FFH,装入 LDTR 的选择符为 0040H,试问装入缓存 LDT 描述符的起始地址是多少?

解:根据(GDTR) = 0013000000FFH,得到全局描述符表的基地址为 00130000H;再根据 LDTR 选择符内容为 0040H(0000 0000 0100 0000B),得到索引值为 0 0000 0000 1000B,即 0008H。因为每个描述符为 8 个字节,故所装入的描述符在 GDT 中的偏移地址为 $(0008H-1) \times 8 = 0038H$ 。所以装入缓存的 LDT 描述符的起始地址为 00130038H。

2.29 如果允许分页,那么 80386、80486 和 Pentium 的地址空间可映射到多少页?页有多大?

解:页的大小为 4KB。允许分页时,32 位的线性地址分为 3 个部分,高 10 位用来查找页目录,页目录中的目录项指明了需要使用哪一个页表。中间 10 位用来查找对应的页表,页表中的页地址指明了所需页的基地址。最低 12 位是页内偏移量。由此可知,全部页目录和全部页表共可映射 $2^{10} \times 2^{10} = 1048576$ 页。

2.30 页转换产生的线性地址的三部分各是什么?

解:页目录索引、页表索引和页内偏移。

2.31 一个描述符中有几个字节?试说明其中每一个域名及大小。



解：每个描述符有 64 位(8 个字节)。其中：段基址 32 位，段界限 20 位，控制位 12 位。

2.32 选择符 022416H 装入了数据段寄存器，该值指向局部描述符表中从地址 00100220H 开始的段描述符。如果该描述符的内容为：

(00100220H)=10H,(00100221H)=22H

(00100222H)=00H,(00100223H)=10H

(00100224H)=1CH,(00100225H)=80H

(00100226H)=01H,(00100227H)=01H

则段基址和段界限各为多少？

解：把题目给出的内容按描述符格式可写为图 1-2-1 所示的形式。

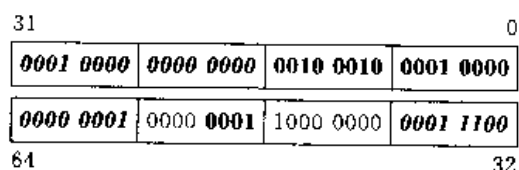


图 1-2-1

根据段描述符的构成可知，段基址为 0000 0001 0001 1100 0001 0000 0000 0000B (见图 1-2-1 中斜黑体字部分)，写成 16 进制数为 011C1000H。段界限为 0001 0010 0010 0001 0000B(见图 1-2-1 中黑体字部分)，写成 16 进制数为 12210H。

2.33 Pentium 4 的基本程序执行环境包含了哪些寄存器？

解：参见教材第 90 页图 2-52，此处略。

3.1 什么叫寻址方式? 8086/8088CPU 共有哪几种寻址方式?

解: 寻址方式主要是指获得操作数所在地址的方法。8086/8088CPU 具有: 立即寻址、直接寻址、寄存器寻址、寄存器间接寻址、寄存器相对寻址、基址—变址寻址、基址—变址—相对寻址以及隐含寻址等 8 种寻址方式。

3.2 设 $(DS) = 6000H$, $(ES) = 2000H$, $(SS) = 1500H$, $(SI) = 00A0H$, $(BX) = 0800H$, $(BP) = 1200H$, 数据变量 VAR 为 0050H。

请分别指出下列各条指令源操作数的寻址方式? 它的物理地址是多少?

- | | |
|------------------|-------------------------|
| (1) MOV AX, BX | (2) MOV DL, 80H |
| (3) MOV AX, VAR | (4) MOV AX, VAR[BX][SI] |
| (5) MOV AL, 'B' | (6) MOV DI, ES: [BX] |
| (7) MOV DX, [BP] | (8) MOV BX, 20H[BX] |

解:

(1) 寄存器寻址。因源操作数是寄存器, 故寄存器 BX 就是操作数的地址。

(2) 立即寻址。操作数 80H 存放于代码段中指令码 MOV 之后。

(3) 立即寻址。

(4) 基址—变址—相对寻址。

$$\begin{aligned}\text{操作数的物理地址} &= (DS) \times 16 + (SI) + (BX) + \text{VAR} \\ &= 60000H + 00A0H + 0800H + 0050H = 608F0H\end{aligned}$$

(5) 立即寻址。

(6) 寄存器间接寻址。

$$\begin{aligned}\text{操作数的物理地址} &= (ES) \times 16 + (BX) \\ &= 20000H + 0800H = 20800H\end{aligned}$$

(7) 寄存器间接寻址。

$$\begin{aligned}\text{操作数的物理地址} &= (SS) \times 16 + (BP) \\ &= 15000H + 1200H = 16200H\end{aligned}$$

(8) 寄存器相对寻址。

$$\begin{aligned}\text{操作数的物理地址} &= (DS) \times 16 + (BX) + 20H \\ &= 60000H + 0800H + 20H = 60820H\end{aligned}$$

3.3 假设 $(DS) = 212AH$, $(CS) = 0200H$, $(IP) = 1200H$, $(BX) = 0500H$, 位移量 $DATA = 40H$, $(217A0H) = 2300H$, $(217E0H) = 0400H$, $(217E2H) = 9000H$

试确定下列转移指令的转移地址。

(1) `JMP 2300H`

(2) `JMP WORD PTR[BX]`

(3) `JMP DWORD PTR[BX+DATA]`

解: 转移指令分为段内转移和段间转移, 根据其寻址方式的不同, 又有段内的直接转移和间接转移, 以及段间的直接转移和间接转移地址。对直接转移, 其转移地址为当前指令的偏移地址(即 IP 的内容)加上位移量或由指令中直接得出; 对间接转移, 转移地址等于指令中寄存器的内容或由寄存器内容所指向的存储单元的内容。

(1) 段内直接转移。转移的物理地址 $= (CS) \times 16 + (IP) + 2300H$

$$= 02000H \times 16 + 1200H + 2300H = 05500H$$

(2) 段内间接转移。转移的物理地址 $= (CS) \times 16 + [BX]$

$$= (CS) \times 16 + (217A0H)$$

$$= 02000H \times 16 + 2300H = 04300H$$

(3) 段间间接转移。转移的物理地址 $= [BX+DATA]$

$$= (217E2H) \times 16 + (217E0H)$$

$$= 90000H + 0400H = 90400H$$

3.4 试说明指令 `MOV BX, 5[BX]` 与指令 `LEA BX, 5[BX]` 的区别。

解: 前者是数据传送类指令, 表示将数据段中以 $(BX+5)$ 为偏移地址的 16 位数据送寄存器 BX 。

后者是取偏移地址指令, 执行的结果是 $(BX) = (BX) + 5$, 即操作数的偏移地址为 $(BX) + 5$ 。

3.5 设堆栈指针 SP 的初值为 $2300H$, $(AX) = 50ABH$, $(BX) = 1234H$ 。执行指令 `PUSH AX` 后, $(SP) = ?$, 再执行指令 `PUSH BX` 及 `POP AX` 之后, $(SP) = ?$ $(AX) = ?$ $(BX) = ?$

解: 堆栈指针 SP 总是指向栈顶, 每执行一次 `PUSH` 指令 $SP - 2$, 执行一次 `POP` 指令 $SP + 2$ 。所以, 执行 `PUSH AX` 指令后, $(SP) = 22FEH$; 再执行 `PUSH BX` 及 `POP AX` 后, $(SP) = 22FEH$, $(AX) = (BX) = 1234H$

3.6 指出下列指令的错误:

(1) `MOV AH, CX`

(2) `MOV 33H, AL`

(3) `MOV AX, [SI][DI]`

(4) `MOV [BX], [SI]`

(5) `ADD BYTE PTR[BP], 256`

(6) `MOV DATA[SI], ES: AX`

(7) `JMP BYTE PTR[BX]`

(8) `OUT 230H, AX`

(9) `MOV DS, BP`

(10) `MUL 39H`

解:

(1) 指令错。两操作数字长不相等。

(2) 指令错。MOV 指令不允许目标操作数为立即数。

(3) 指令错。在间接寻址中不允许两个间址寄存器同时为变址寄存器。

(4) 指令错。MOV 指令不允许两个操作数同时为存储器操作数。

(5) 指令错。ADD 指令要求两操作数等字长。

(6) 指令错。源操作数形式错,寄存器操作数不加段重设符。

(7) 指令错。转移地址的字长至少应是 16 位的。

(8) 指令错。对输入输出指令,当端口地址超出 8 位二进制数的表达范围(即寻址的端口超出 256 个)时,必须采用间接寻址。

(9) 指令正确。

(10) 指令错。MUL 指令不允许操作数为立即数。

3.7 已知 $(AL)=7BH$, $(BL)=38H$, 试问执行指令 ADD AL, BL 后, AF、CF、OF、PF、SF 和 ZF 的值各为多少?

解: $AF=1, CF=0, OF=1, PF=0, SF=1, ZF=0$

3.8 试比较无条件转移指令、条件转移指令、调用指令和中断指令有什么异同?

解: 无条件转移指令的操作是无条件地使程序转移到指定的目标地址, 并从该地址开始执行新的程序段, 其转移的目标地址既可以是在当前逻辑段, 也可以是在不同的逻辑段; 条件转移指令是在满足一定条件下使程序转移到指定的目标地址, 其转移范围很小, 在当前逻辑段的 $-128 \sim +127$ 地址范围内。

调用指令是用于调用程序中常用到的功能子程序, 是在程序设计中就设计好的。根据所调用过程入口地址的位置可将调用指令分为段内调用(入口地址在当前逻辑段内)和段间调用。在执行调用指令后, CPU 要保护断点。对段内调用是将其下一条指令的偏移地址压入堆栈, 对段间调用则要保护其下一条指令的偏移地址和段基地址, 然后将子程序入口地址赋给 IP(或 CS 和 IP)。

中断指令是因一些突发事件而使 CPU 暂时中止它正在运行的程序, 转去执行一组专门的中断服务程序, 并在执行完后返回原被中止处继续执行原程序。它是随机的。在响应中断后 CPU 不仅要保护断点(即 INT 指令下一条指令的段地址和偏移地址), 还要将标志寄存器 FLAGS 压入堆栈保存。

3.9 试判断下列程序执行后, BX 中的内容。

```
MOV CL, 3
MOV BX, 0B7H
ROL BX, 1
ROR BX, CL
```

解: 该程序段是首先将 BX 内容不带进位循环左移 1 位, 再循环右移 3 位。即相当于将原 BX 内容不带进位循环右移 2 位, 故结果为: $(BX)=0C02DH$ 。

3.10 按下列要求写出相应的指令或程序段。

- (1) 写出两条使 AX 内容为 0 的指令。
- (2) 使 BL 寄存器中的高 4 位和低 4 位互换。
- (3) 屏蔽 CX 寄存器的 b11、b7 和 b3 位。
- (4) 测试 DX 中的 b0 和 b8 位是否为 1。

解:

(1)

MOV AX,0

XOR AX,AX

;AX 寄存器自身相异或,可使其内容清 0

(2)

MOV CL,4

ROL BX,CL

;将 BX 内容循环左移 4 位,可实现其高 4 位和低 4 位的互换

(3)

AND CX,0F777H

;将 CX 寄存器中需屏蔽的位“与”0。也可用“或”指令实现

(4)

AND DX,0101H

;将需测试的位“与”1,其余“与”0 屏蔽掉

CMP DX,0101H

;与 0101H 比较

JZ ONE

;若相等则表示 b0 和 b8 位同时为 1

:

3.11 分别指出以下两个程序段的功能:

(1)

MOV CX,10

LEA SI,FIRST

LEA DI,SECOND

STD

REP MOVSB

(2)

CWD

LEA DI,[1200H]

MOV CX,0F00H

XOR AX,AX

REP STOSW

解:

(1) 该段程序的功能是:将数据段中 FIRST 为最高地址的 10 个字节数据按减地址方向传送到附加段 SECOND 为最高地址的向前 10 个单元中。

(2) 将附加段中偏移地址为 1200H 单元开始的 0F00H 个字单元清 0。

3.12 执行以下两条指令后,标志寄存器 FLAGS 的六个状态位各为什么状态?

MOV AX,84A0H

ADD AX,9460H

解:执行 ADD 指令后,6 个状态标志位的状态分别为:

在两个 16 位数进行加法运算时,对 CF、ZF、SF 和 OF 会产生影响,但对 PF 和 AF 标志位,只有其低 8 位的运算影响它们的状态。各标志位的状态分别为:AF=0,PF=1,CF=1,ZF=0,SF=0,OF=1。

3.13 将+46 和-38 分别乘以 2,可应用什么指令来完成? 如果除以 2 呢?

解:因为对二进制数,每左移一位相当于乘以 2,右移一位相当于除以 2。所以,将+46 和-38 分别乘以 2,可分别用逻辑左移指令(SHL)和算术左移指令(SAL)完成。SHL 指令针对无符号数,SAL 指令针对有符号数。

当然,也可以分别用无符号数乘法指令 MUL 和有符号数乘法指令 IMUL 完成。

如果是除以 2,则进行相反操作,即用逻辑右移指令 SHR 或无符号数除法指令 DIV 实现+46 除以 2 的运算,用算术右移指令 SAR 或有符号数除法指令 IDIV 实现-38 除



以 2 的运算。

3.14 已知 $AX=8060H$, $DX=03F8H$, 端口 $PORT1$ 的地址是 $48H$, 内容为 $40H$; $PORT2$ 的地址是 $84H$, 内容为 $85H$ 。请指出下列指令执行后的结果。

- (1) `OUT DX,AL`
- (2) `IN AL,PORT1`
- (3) `OUT DX,AX`
- (4) `IN AX,48H`
- (5) `OUT PORT2,AX`

解:

- (1) 将 $60H$ 输出到地址为 $03F8H$ 的端口中。
- (2) 从 $PORT1$ 读入一个字节数据, 执行结果: $(AL)=40H$ 。
- (3) 将 $AX=8060H$ 从地址为 $03F8H$ 的端口输出。
- (4) 由 $48H$ 端口读入 16 位二进制数。
- (5) 将 $8060H$ 从地址为 $84H$ 的端口输出。

3.15 试编写程序, 统计 $BUFFER$ 为起始地址的连续 200 个单元中 0 的个数。

解: 将 $BUFFER$ 为首地址的 200 个单元的数依次与 0 进行比较, 若相等则表示该单元数为 0, 统计数加 1; 否则再取下一个数比较, 直到 200 个单元数全部比较完毕为止。程序如下:

	<code>LEA SI,BUFFER</code>	;取 $BUFFER$ 的偏移地址
	<code>MOV CX,200</code>	;数据长度送 CX
	<code>XOR BX,BX</code>	;存放统计数寄存器清 0
AGAIN:	<code>MOV AL,[SI]</code>	;取一个数
	<code>CMP AL,0</code>	;与 0 比较
	<code>JNE GOON</code>	;不为 0 则准备取下一个数
	<code>INC BX</code>	;为 0 则统计数加 1
GOON:	<code>INC SI</code>	;修改地址指针
	<code>LOOP AGAIN</code>	;若未比较完则继续比较
	<code>HLT</code>	

3.16 写出完成下述功能的程序段:

- (1) 从地址 $DS:0012H$ 中传送一个数据 $56H$ 到 AL 寄存器。
- (2) 将 AL 中的内容左移两位。
- (3) AL 的内容与字节单元 $DS:0013H$ 中的内容相乘。
- (4) 乘积存入字单元 $DS:0014H$ 中。

解:

- (1) `MOV DS, BYTE PTR[0012H],56H`
`MOV AL,[0012H]`
- (2) `MOV CL,2`
`SHL AL,CL`
- (3) `MUL DS: BYTE PTR[0013H]`

(4) MOV DS: [0014H], AX

3.17 若 $(AL) = 96H$, $(BL) = 12H$, 在分别执行指令 MUL 和 IMUL 后, 其结果是多少? OF=? CF=?

解: MUL 是无符号数的乘法指令, 它将两操作数示为无符号数; IMUL 是有符号数的乘法指令, 此时, 两操作数被看作有符号数。在该题中, $(AL) = 96H$, 其最高位为 1, 是负数。IMUL 指令的执行原理是先求出它的真值(即对它求补), 再做乘法运算。

执行 MUL BL 指令后, $(AX) = 0A8CH$, $CF = OF = 1$

执行 IMUL BL 指令后, $(AX) = F88CH$, $CF = OF = 1$

4.1 请分别用 DB、DW、DD 伪指令写出在 DATA 开始的连续 8 个单元中依次存放数据 11H、22H、33H、44H、55H、66H、77H、88H 的数据定义语句。

解：DB、DW、DD 伪指令分别表示定义的数据为字节型、字类型及双字型。其定义形式分别为：

```
DATA DB 11H,22H,33H,44H,55H,66H,77H,88H
```

```
DATA DW 2211H,4433H,6655H,8877H
```

```
DATA DD 44332211H,88776655H
```

4.2 若程序的数据段定义如下，写出各指令语句独立执行后的结果：

```
DSEG SEGMENT
```

```
DATA1 DB 10H,20H,30H
```

```
DATA2 DW 10 DUP(?)
```

```
STRING DB '123'
```

```
DSEG ENDS
```

```
(1) MOV AL,DATA1
```

```
(2) MOV BX,OFFSET DATA2
```

```
(3) LEA SI,STRING
```

```
ADD DI,SI
```

解：

(1) 取变量 DATA1 的值。指令执行后，(AL)=10H。

(2) 变量 DATA2 的偏移地址。指令执行后，(BX)=0003H。

(3) 先取变量 STRING 的偏移地址送寄存器 SI，之后将 SI 的内容与 DI 的内容相加并将结果送 DI。指令执行后，(SI)=0017H；(DI)=(DI)+0017H。

4.3 试编写求两个无符号双字长数之和的程序。两数分别在 MEM1 和 MEM2 单元中，和放在 SUM 单元。

解：

```
DSEG SEGMENT
```

```
MEM1 DW 1122H,3344H
```

```
MEM2 DW 5566H,7788H
```



```

SUM    DW 2 DUP(?)
DSEG  ENDS
CSEG  SEGMENT
        ASSUME CS, CSEG, DS, DSEG
START: MOV AX, DSEG
        MOV DS, AX
        LEA BX, MEM1
        LEA SI, MEM2
        LEA DI, SUM
        MOV CL, 2
        CLC
AGAIN:  MOV AX, [BX]
        ADC AX, [SI]
        MOV [DI], AX
        ADD BX, 2
        ADD SI, 2
        ADD DI, 2
        LOOP AGAIN
        HLT
CSEG  ENDS
        END START

```

4.4 试编写程序,测试 AL 寄存器的第 4 位(bit4)是否为 0?

解: 测试寄存器 AL 中某一位是否为 0,可使用 TEST 指令、AND 指令、移位指令等几种方法实现。

```

如: TEST AL, 10H
      JZ NEXT
      :
NEXT: ...

```

```

或者: MOV CL, 4
      SHL AL, CL
      JNC NEXT
      :
NEXT: ...

```

4.5 试编写程序,将 BUFFER 中的一个 8 位二进制数转换为 ASCII 码,并按位数高低顺序存放在 ANSWER 开始的内存单元中。

解:

```

DSEG SEGMENT
BUFFER    DB  ?                ;要转换的数

```

```

ANSWER  DB  3 DUP(?)           ;ASCII 码结果存放单元
DSEG    ENDS
CSEG    SEGMENT
        ASSUME CS,CSEG,DS,DSEG
START:  MOV  AX,DSEG
        MOV  DS,AX
        MOV  CX,3               ;最多不超过 3 位十进制数(255)
        LEA  DI,ANSWER          ;DI 指向结果存放单元
        XOR  AX,AX
        MOV  AL,BUFFER          ;取要转换的二进制数
        MOV  BL,0AH             ;基数 10
AGAIN:  DIV  BL                  ;用除 10 取余的方法转换
        ADD  AH,30H             ;十进制数转换成 ASCII 码
        MOV  [DI],AH            ;保存当前位的结果
        INC  DI                  ;指向下一个位保存单元
        AND  AL,AL               ;商为 0? (转换结束?)
        JZ   STO                 ;若结束,退出
        MOV  AH,0
        LOOP AGAIN              ;否则循环继续
STO:    MOV  AX,4CH
        INT  21H                 ;返回 DOS
CSEG    ENDS
        END  START

```

4.6 假设数据项定义如下:

DATA1 DB 'HELLO! GOOD MORNING!'

DATA2 DB 20 DUP(?)

用串操作指令编写程序段,使其分别完成以下功能:

- (1) 从左到右将 DATA1 中的字符串传送到 DATA2 中。
- (2) 传送完后,比较 DATA1 和 DATA2 中的内容是否相同。
- (3) 把 DATA1 中的第 3 和第 4 个字节装入 AX。
- (4) 将 AX 的内容存入 DATA2+5 开始的字节单元中。

解:

(1)

```

MOV AX,SEG DATA1
MOV DS,AX
MOV AX,SEG DATA2
MOV ES,AX
LEA SI,DATA1
LEA DI,DATA2
MOV CX,20

```

```

CLD
REP MOVSB
(2)
LEA SI,DATA1
LEA DI,DATA2
MOV CX,20
CLD
REPE CMPSB

```

...

```

(3)
LEA SI,DATA1
ADD SI,2
LODSW
(4)
LEA DI,DATA2
ADD DI,5
STOSW

```

4.7 执行下列指令后,AX 寄存器中的内容是多少?

```

TABLE DW 10,20,30,40,50
ENTRY DW 3
:
MOV BX,OFFSET TABLE
ADD BX,ENTRY
MOV AX,[BX]

```

解: (AX)=1E00H

4.8 编写程序段,将 STRING1 中的最后 20 个字符移到 STRING2 中(顺序不变)。

解: 首先确定 STRING1 中字符串的长度,因为字符串的定义要求以 '\$' 符号结尾,可通过检测 '\$' 符确定出字符串的长度,设串长度为 COUNT,则程序如下:

```

LEA SI,STRING1
LEA DI,STRING2
ADD SI,COUNT-20
MOV CX,20
CLD
REP MOVSB

```

4.9 假设一个 48 位数存放在 DX:AX:BX 中,试编写程序段,将该 48 位数乘以 2。

解: 可使用移位指令来实现。首先将 BX 内容逻辑左移一位,其最高位移入进位位 CF,之后 AX 内容带进位位循环左移,使 AX 的最高位移入 CF,而原 CF 中的内容(即 BX 的最高位)移入 AX 的最低位,最后再将 DX 内容带进位位循环左移一位,从而实现 AX

的最低位移入 DX 的最低位。

```
SHL BX,1
RCL AX,1
RCL DX,1
```

4.10 试编写程序,比较 AX, BX, CX 中带符号数的大小,并将最大的数放在 AX 中。

解:比较带符号数的大小可使用符号数比较指令 JG 等。

```
      CMP AX,BX
      JG NEXT1
      XCHG AX,BX
NEXT1: CMP AX,CX
      JG STO
      MOV AX,CX
STO:  HLT
```

4.11 若接口 03F8H 的第 1 位(b1)和第 3 位(b3)同时为 1,表示接口 03FBH 有准备好的 8 位数据,当 CPU 将数据取走后,b1 和 b3 就不再同时为 1 了。仅当又有数据准备好时才再同时为 1。

试编写程序,从上述接口读入 200 字节的数据,并顺序放在 DATA 开始的地址中。

解:即当从输入接口 03F8H 读入的数据满足 $\times\times\times\times1\times1\times B$ 时可以从接口 03FBH 输入数据。

```
      LEA SI,DATA
      MOV CX,200
NEXT:  MOV DX,03F8H
WAITT: IN AL,DX
      AND AL,0AH           ;判断 b1 和 b3 位是否同时为 1
      CMP AL,0AH
      JNZ WAITT            ;b1 和 b3 位同时为 1 则读数据,否则等待
      MOV DX,03FBH
      IN AL,DX
      MOV [SI],AL
      INC SI
      LOOP NEXT
      HLT
```

4.12 画图说明下列语句分配的存储空间及初始化的数据值。

(1) DATA1 DB 'BYTE',12,12H,2 DUP(0,?,3)

(2) DATA2 DW 4 DUP(0,1,2),?,-5,256H

解:

(1) 存储空间分配情况如图 1-4-1(a)所示。

(2) 存储空间分配情况如图 1-4-1(b)所示。

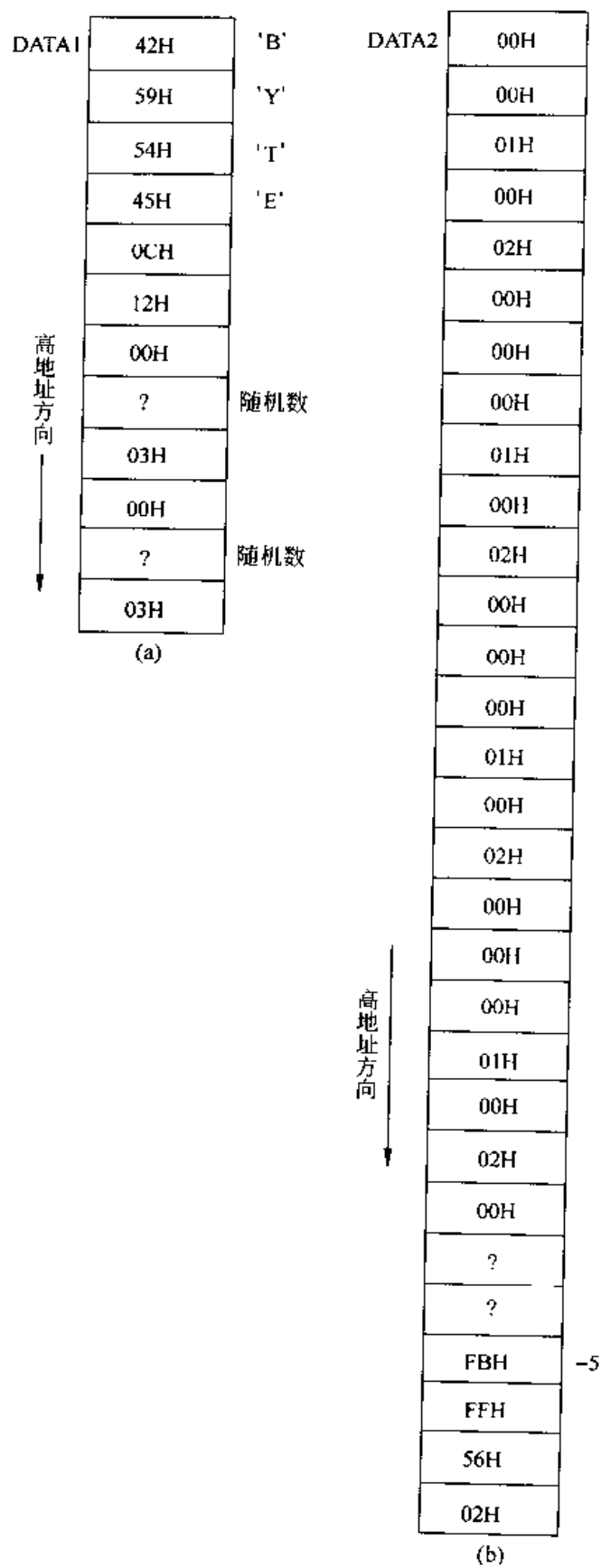


图 1-4-1

4.13 请用子程序结构编写如下程序：从键盘输入一个二位十进制的月份数(01~12),然后显示出相应的英文缩写名。

解：可根据题目要求编写如下几个子程序：

INPUT 从键盘接收一个二位数,并将其转换为二进制数。

LOCATE 通过字符表查找将输入数与英文缩写对应起来。

DISPLAY 将缩写字母在屏幕上显示。

程序如下：

```
DSEG SEGMENT
DATA1 DB 3
DATA2 DB 3,?,3 DUP(?)
ALFMON DB '???'','$'
MONTAB DB 'JAN','FEB','MAR','APR','MAY','JUN'
        DB 'JUL','AUG','SEP','OCT','NOV','DEC'
DSEG ENDS

;
SSEG SEGMENT STACK 'STACK'
        DB 100 DUP(?)
SSEG ENDS

;
CSEG SEGMENT
ASSUME CS: CSEG,DS: DSEG,ES: DSEG,SS: SSEG
;
MAIN    PROC FAR
        PUSH DS                ;恢复断点
        XOR AX,AX
        PUSH AX
        MOV AX,DSEG            ;段初始化
        MOV DS,AX
        MOV ES,AX
        MOV AX,SSEG
        MOV SS,AX
        CALL INPUT
        CALL LOCATE
        CALL DISPLAY
        RET
MAIN    ENDP

;
INPUT   PROC NEAR
        PUSH DX
        MOV AH,0AH            ;从键盘输入月份数
        LEA DX,DATA2
```

```

        INT 21H
        MOV AH,DATA2+2      ;输入月份数的 ASCII 码送 AX
        MOV AL,DATA2+3
        XOR AX,3030H        ;将月份数的 ASCII 码转换为二进制数
        CMP AH,00H          ;确定是否为 1~9 月
        JZ RETURN
        SUB AH,AH            ;若为 10~12 月则清高 8 位
        ADD AL,10            ;转为二进制码
RETURN;POP DX
        RET
INPUT  ENDP
;
LOCATE PROC NEAR
        PUSH SI
        PUSH DI
        PUSH CX
        LEA SI,MONTAB
        DEC AL
        MUL DATA1          ;每月为 3 个字符
        ADD SI,AX            ;指向月份对应的英文缩写字母的地址
        MOV CX,03H
        CLD
        LEA DI,ALFMON
        REP MOVSB
        POP CX
        POP DI
        POP SI
        RET
LOCATE ENDP
;
DISPLAY PROC
        PUSH DX
        LEA DX,ALFMON
        MOV AH,09H
        INT 21H
        POP DX
        RET
DISPLAY ENDP
;
CSEG ENDS
END MAIN

```



4.14 给出下列等值语句：

```
ALPHA EQU 100
BETA EQU 25
GRAMM EQU 4
```

试求下列表达式的值：

- (1) $ALPHA * 100 + BETA$
- (2) $(ALPHA + 4) * BETA - 2$
- (3) $(BETA / 3) MOD$
- (4) $GRAMM OR 3$

解：

- (1) $10000 + 25 = 10025$
- (2) $104 \times 25 - 2 = 2598$
- (3) $(25 / 3) MOD 5 = 3$
- (4) $4 OR 3 = 7$

4.15 图示以下数据段在存储器中的存放形式：

```
DATA SEGMENT
DATA1 DB 10H,34H,07H,09H
DATA2 DW 2 DUP(42H)
DATA3 DB 'HELLO!'
DATA4 EQU 12
DATA5 DD 0ABCDH
DATA ENDS
```

解：符号定义伪指令 EQU 所定义的变量 DATA4 不占用内存，故内存数据区有如图 1-4-2 存放形式。

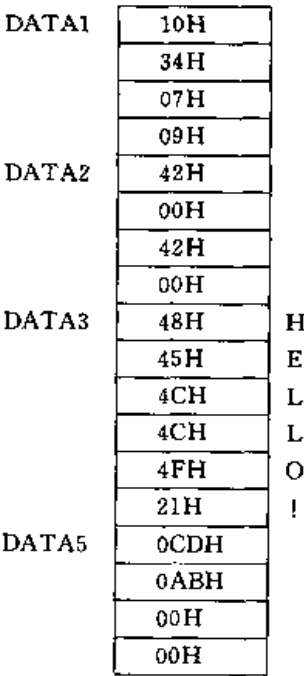


图 1-4-2

4.16 阅读下边的程序段,试说明它实现的功能是什么?

```
DATA SEGMENT
DATA1 DB 'ABCDEFGH'
DATA ENDS
CODE SEGMENT
    ASSUME CS, CODE, DS, DATA
AAAA: MOV AX, DATA
      MOV DS, AX
      MOV BX, OFFSET DATA1
      MOV CX, 7
NEXT:  MOV AH, 2
      MOV AL, [BX]
      XCHG AL, DL
      INC BX
      INT 21H
      LOOP NEXT
      MOV AH, 4CH
      INT 21H
CODE ENDS
      END AAAA
```

解:该程序段是将 ABCDEFGH 7 个字母依次显示在屏幕上。

4.17 编写一程序段,把从 BUFFER 开始的 100 个字节的内存区域初始化成 55H、0AAH、55H、0AAH、...、55H、0AAH。

解:可用串存储指令实现。

```
DSEG SEGMENT
BUFFER DB 100 DUP(?)
DSEG ENDS
CSEG SEGMENT
    ASSUME CS, CSEG, DS, DSEG, ES, DSEG
BEGIN: MOV AX, DSEG
      MOV DS, AX
      MOV ES, AX
      MOV AX, 0AA55H
      LEA DI, BUFFER
      CLD
      MOV CX, 50
      REP STOSW
      HLT
CSEG ENDS
      END BEGIN
```



4.18 有 16 个字节,编程序将其中第 2、5、9、14、15 个字节内容加 3,其余字节内容乘 2(假定运算不会溢出)。

解:

```
DSEG  SEGMENT
DATA  DB 16 DUP(?)
DSEG  ENDS
CSEG  SEGMENT
      ASSUME CS, CSEG, DS, DSEG
BEGIN: MOV AX, DSEG
      MOV DS, AX
      LEA SI, DATA
      MOV CL, 1
AGAIN: MOV AL, [SI]
      CMP CL, 2
      JE ADDD
      CMP CL, 5
      JE ADDD
      CMP CL, 9
      JE ADDD
      CMP CL, 14
      JE ADDD
      CMP CL, 15
      JE ADDD
      SHL AL, 1
      JMP GOON
ADDD:  ADD AL, 3
GOON:  MOV [SI], AL
      INC SI
      INC CL
      CMP CL, 16
      JNG AGAIN
      HLT
CSEG  ENDS
      END BEGIN
```

4.19 编写计算斐波那契数列前 20 个值的程序。斐波那契数列的定义如下:

$$\begin{cases} F(0) = 0, \\ F(1) = 1, \\ F(n) = F(n-1) + F(n-2), n \geq 2 \end{cases}$$

解: 根据斐波那契数列的定义,将计算出的前 20 个值放在 DATA1 为首地址的内存单元中。程序如下:

```

DATA SEGMENT
DATA1 DB 0,1,18 DUP(?)
DATA ENDS
CODE SEGMENT
        ASSUME CS: CODE,DS: DATA
START:  MOV AX,DATA
        MOV DS,AX
        LEA BX,DATA1
        MOV CL,18
        CLC
NEXT:   XOR AX,AX
        MOV AL,[BX]
        MOV DL,[BX+1]
        ADC AL,DL
        MOV [BX+2],AL
        INC BX
        DEC CL
        JNZ NEXT
        HLT
CODE    ENDS
        END START

```

4.20 试编写将键盘输入的 ASCII 码转换为二进制数的程序。

解：

```

DATA    SEGMENT
BUFFER DB 100 DUP(?)
DATA    ENDS
CODE    SEGMENT
        ASSUME CS: CODE,DS: DATA
START:  MOV AX,DATA
        MOV DS,AX
        LEA SI,BUFFER
NEXT:   MOV AH,1                ;从键盘输入一个数
        INT 21H
        AND AL,7FH              ;去掉最高位
        CMP AL,'0'
        JL STO                  ;若小于0则不属于转换范围
        CMP AL,'9'
        JG ASCB1
        SUB AL,30H              ;对0~9之间的数减去30H转换为二进制数
        JMP ASCB2
ASCB1:  CMP AL,'A'              ;对大于9的数再与A比较

```

```

        JL STO
        CMP AL,'F'
        JG STO
        SUB AL,37H           ;对 A~F 之间的数减去 37H 转换
ASC2:   MOV [SI],AL          ;转换结果存放在 BUFFER 为首地址的单元中
        INC SI
        STO: CMP AL,'$'      ;输入字符'$'则结束
        JNE NEXT
        HLT
CODE    ENDS
        END START

```



5.1 内部存储器主要分为哪两类？它们的主要区别是什么？

解：

(1) 分为 ROM 和 RAM。

(2) 它们之间的主要区别是：

- ROM 在正常工作时只能读出，不能写入。RAM 则可读可写。
- 断电后，ROM 中的内容不会丢失，RAM 中的内容会丢失。

5.2 为什么动态 RAM 需要定时刷新？

解：DRAM 的存储元以电容来存储信息，由于存在漏电现象，电容中存储的电荷会逐渐泄漏，从而使信息丢失或出现错误。因此需要对这些电容定时进行“刷新”。

5.3 CPU 寻址内存的能力最基本的因素取决于_____。

解：地址总线的宽度。

5.4 试利用全地址译码将 6264 芯片接到 8088 系统总线上，使其所占地址范围为 32000H~33FFFH。

解：将地址范围展开成二进制形式如图 1-5-1 所示。

0011 000	1 0000 0000 0000
~	
0011 001	0 1111 1111 1111

图 1-5-1

6264 芯片的容量为 $8 \times 8\text{KB}$ ，需要 13 根地址线 $A_0 \sim A_{12}$ （见图 1-5-1 中虚线框内的部分）。而剩下的高 7 位地址应参加该芯片的地址译码。

但是，仔细分析一下就可以看出，上述地址范围跨越了两个 8KB 地址空间：30000H~31FFFH 的高 4KB 和 32000H~33FFFH 的低 4KB。也就是说，在该地址空间内高位地址不惟一，直接利用高 7 位进行片选译码是行不通的。为解决这个问题，我们可以观察一下该地址范围的特征，bit13 和 bit12 总是相反的。因此，可以把这两位进行异或运算后作为译码器的一个输入。电路如图 1-5-2 所示。

5.5 内存地址从 20000H~8BFFFH 共有多少字节？

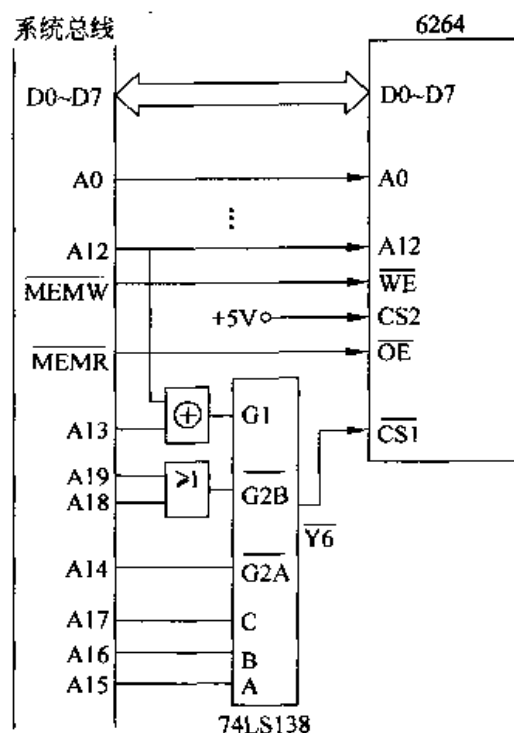


图 1-5-2

解：共有 $8\text{BFFFH} - 20000\text{H} + 1 = 6\text{C000H}$ 个字节。或 432KB。

5.6 若采用 6264 芯片构成上述的内存空间,需要多少片 6264 芯片?

解：每个 6264 芯片的容量为 8KB,故需 $432/8 = 54$ 片。

5.7 设某微型机的内存 RAM 区的容量为 128KB,若用 2164 芯片构成这样的存储器,需多少 2164 芯片? 至少需多少根地址线? 其中多少根用于片内寻址? 多少根用于片选译码?

解：

(1) 每个 2164 芯片的容量为 64KB,共需 $128/64 \times 8 = 16$ 片。

(2) 128KB 容量需要地址线 17 根。

(3) 16 根用于片内寻址。

(4) 1 根用于片选译码。

注意,用于片内寻址的 16 根地址线要通过二选一多路器连到 2164 芯片,因为 2164 芯片是 DRAM,高位地址与低位地址是分时传送的。

5.8 现有两片 6116 芯片,所占地址范围为 $61000\text{H} \sim 61\text{FFFH}$,试将它们连接到 8088 系统中。并编写测试程序,向所有单元输入一个数据,然后再读出与之比较,若出错则显示“Wrong!”,全部正确则显示“OK!”。

解：连接如图 1-5-3 所示。测试程序段如下：

```
OK      DB 'OK!', $
WRONG   DB 'Wrong!', $
...
```

```

MOV AX,6100H
MOV ES,AX
MOV DI,0
MOV CX,1000H
MOV AL,55H
REP STOSB
MOV DI,0
MOV CX,1000H
REPZ SCASB
JZ DISP_OK
LEA DX,WRONG
MOV AH,9
INT 21H
HLT
DISP_OK:
LEA DX,OK
MOV AH,9
INT 21H
HLT

```

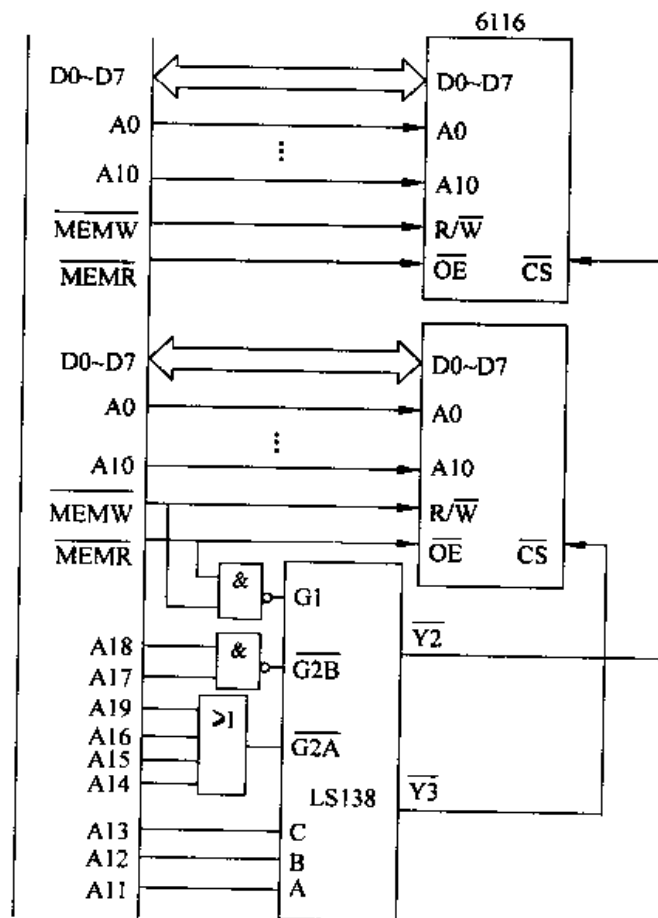


图 1-5-3

5.9 什么是字扩展？什么是位扩展？ 用户自己购买内存条进行内存扩充，是在进行何种存储器扩展？

解：

(1) 当存储芯片的容量小于所需内存容量时，需要用多个芯片构成满足容量要求的存储器，这就是字扩展。

(2) 当存储芯片每个单元的字长小于所需内存单元字长时，需要用多个芯片构成满足字长要求的存储模块，这就是位扩展。

(3) 用户在市场上购买内存条进行内存扩充，所做的是字扩展的工作。

5.10 74LS138 译码器的接线图如教材第 245 页的图 5-47 所示，试判断其输出端 $\overline{Y0}$ 、 $\overline{Y3}$ 、 $\overline{Y5}$ 和 $\overline{Y7}$ 所决定的内存地址范围。

解：因为是部分地址译码(A17 不参加译码)，故每个译码输出对应 2 个地址范围：

$Y0\#$ ：00000H ~ 01FFFH 和 20000H ~ 21FFFH

$Y3\#$ ：06000H ~ 07FFFH 和 26000H ~ 27FFFH

$Y5\#$ ：0A000H ~ 0BFFFH 和 2A000H ~ 2BFFFH

$Y7\#$ ：0E000H ~ 0FFFFH 和 2E000H ~ 2FFFFH

5.11 某 8088 系统用 2764 ROM 芯片和 6264 SRAM 芯片构成 16KB 的内存。其中，ROM 的地址范围为 0FE000H ~ 0FFFFFFH，RAM 的地址范围为 0F000H ~ 0F1FFFH。试利用 74LS138 译码，画出存储器与 CPU 的连接图，并标出总线信号名称。

解：连接如图 1-5-4 所示。

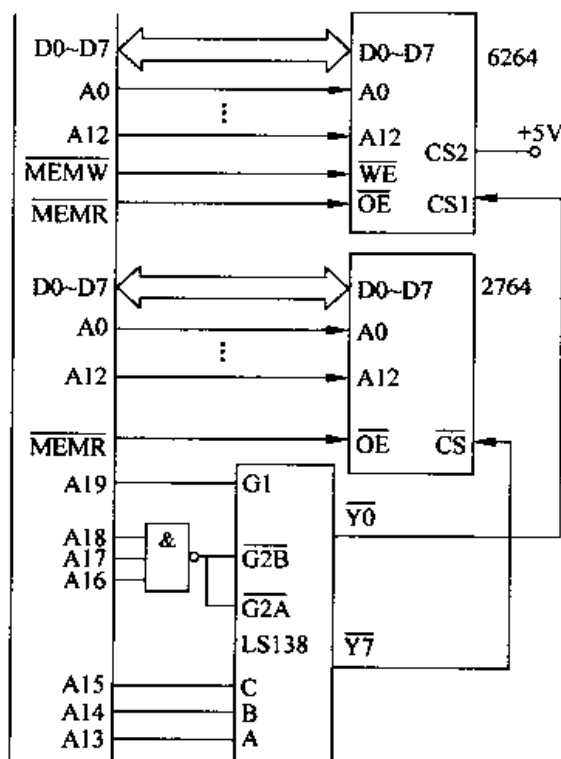


图 1-5-4

5.12 叙述 EPROM 的编程过程,并说明 EPROM 和 EEPROM 的不同点。

解:

(1) 对 EPROM 芯片的编程过程详见教材第 215~217 页。

(2) EPROM 与 EEPROM 的不同之处为:

- EPROM 用紫外线擦除,EEPROM 用电擦除。
- EPROM 是整片擦除,EEPROM 可以整片擦除,也可以逐个字节地擦除。

5.13 试说明 FLASH EEPROM 芯片的特点及 28F040 的编程过程。

解:

(1) 特点是:它结合了 RAM 和 ROM 的优点,读写速度接近于 RAM,断电后信息又不会丢失。

(2) 28F040 的编程过程详见教材第 222~223 页。

5.14 什么是 Cache? 它能够极大地提高计算机的处理能力是基于什么原理?

解:

(1) Cache 是位于 CPU 与主存之间的高速小容量存储器。

(2) 它能够极大地提高计算机的处理能力,是基于程序和数据访问的局部性原理。

5.15 若主存 DRAM 的存取周期为 70ns,Cache 的存取周期为 5ns,Cache 的命中率为 90%,由它们构成的存储器的平均存取周期大约是多少?

解:平均存取周期约为 $70 \times 0.1\text{ns} + 5 \times 0.9\text{ns} = 11.5\text{ns}$ 。

5.16 如何解决 Cache 与主存内容的一致性问题?

解:读和写各有 2 种方式。读出:贯穿读出式和旁路读出式;写入:写穿式和回写式。

5.17 新购买的或擦除干净的 EPROM 芯片,其各单元的内容是什么?

解:0FFH。

5.18 在微型计算机中,CMOS RAM 的作用是什么?

解:保存硬件配置信息。

5.19 简述磁盘存储器的几种接口标准,并分析各自的特点。

解:IDE 接口标准:每个接口支持 2 台设备(主设备和从设备)。传输模式可采用 PIO 或 DMA。成本低,传输速度低,CPU 占用率高,可靠性低。

SCSI 接口标准:每个接口支持 7~15 台设备。采用总线主控方式传送数据。成本高,传输速率高,CPU 占用率低,可靠性高。

5.20 请解释下列名词:磁道、扇区、磁盘容量、磁盘驱动器、磁盘适配器、磁盘控制器。

解:磁道:磁盘表面用于记录信息的一系列同心圆。

扇区:每个磁道平均划分成的磁道片段。

磁盘容量:整个磁盘所能存储的字节数。

磁盘驱动器:对软盘来说,是驱动软盘片转动并负责读写盘片信息的设备。对硬盘来说,是驱动装置、读写电路、盘片的组合。



磁盘适配器和磁盘控制器：一般是指控制磁盘读写的接口电路。

5.21 DOS 能够访问的硬盘最大容量是多少？对于超出 DOS 管理能力的硬盘如何在 DOS 下使用？

解：（略）

5.22 新买的硬盘需要经过哪几个步骤才能存放数据？

解：分区和格式化。

5.23 简述光驱的工作原理。

解：见教材第 242 页。

6.1 I/O 接口的主要功能有哪些？有哪两种编址方式？在 8088/8086 系统中采用哪一种编址方式？

解：I/O 接口主要需具有以下几种功能：

(1) I/O 地址译码与设备选择。保证任一时刻仅有一个外设与 CPU 进行数据传送。

(2) 信息的输入输出，并对外设随时进行监测、控制和管理。必要时，还可以通过 I/O 接口向 CPU 发出中断请求。

(3) 命令、数据和状态的缓冲与锁存。以缓解 CPU 与外设之间工作速度的差异，保证信息交换的同步。

(4) 信号电平与类型的转换。I/O 接口还要实现信息格式变换、电平转换、码制转换、传送管理以及联络控制等功能。

I/O 端口的编址方式通常有两种：一是与内存单元统一编址，二是独立编址。8088/8086 系统采用 I/O 端口独立编址方式。

6.2 试比较 4 种基本输入输出方法的特点。

解：在微型计算机系统中，主机与外设之间的数据传送有 4 种基本的输入输出方式：无条件传送方式、查询工作方式、中断工作方式、直接存储器存取(DMA)方式。

它们各自具有以下特点：

(1) 无条件传送方式适合于简单的、慢速的、随时处于“准备好”接收或发送数据的外部设备，数据交换与指令的执行同步，控制方式简单。

(2) 查询工作方式针对并不随时“准备好”、且需满足一定状态才能实现数据的输入/输出的简单外部设备，其控制方式也较简单，但 CPU 的效率比较低。

(3) 中断工作方式是由外部设备作为主动的一方，在需要时向 CPU 提出工作请求，CPU 在满足响应条件时响应该请求并执行相应的中断处理程序。这种工作方式使 CPU 的效率较高，但控制方式相对较复杂。

(4) DMA 方式适合于高速外设，是 4 种基本输入/输出方式中速度最高的一种。

6.3 主机与外部设备进行数据传送时，采用哪一种传送方式，CPU 的效率最高？

解：使用 DMA 传送方式 CPU 的效率最高。这是由 DMA 的工作性质所决定的。

6.4 利用三态门芯片 74LS244 作为输入接口，接口地址为 40FBH，试画出其与 8088 系统总线的连接图。

解：16 位地址信号通过译码电路与 74LS244 芯片连接。其连接如图 1-6-1 所示。

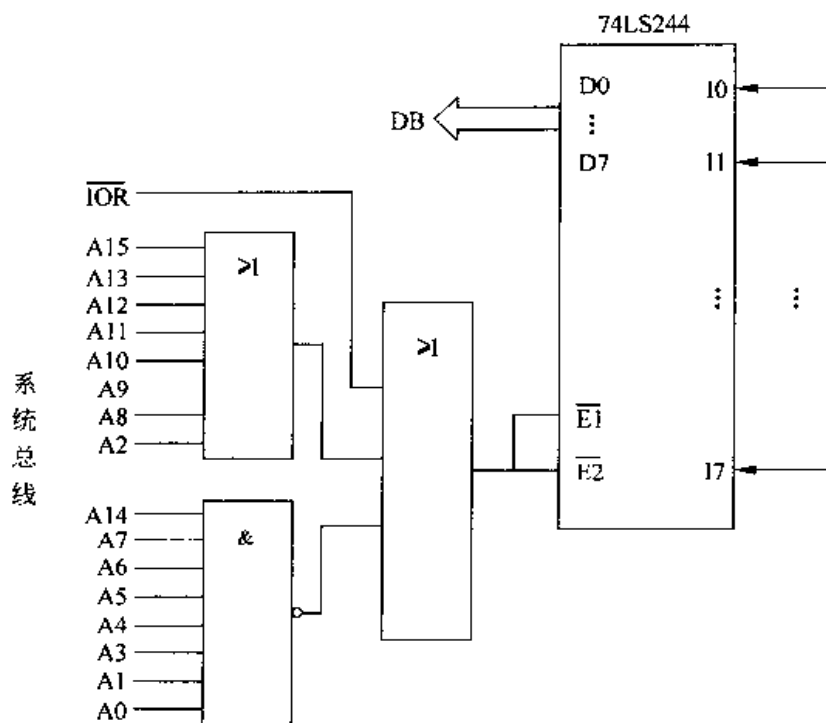


图 1-6-1

6.5 某输入接口的地址为 0E54H,输出接口的地址为 01FBH,分别利用 74LS244 和 74LS273 作为输入和输出接口。试编写程序,使当输入接口的 bit1、bit4 和 bit7 位同时为 1 时,CPU 将内存中 DATA 为首址的 20 个单元的数据从输出接口输出;若不满足上述条件则等待。

解：首先判断由输入接口读入数据的状态,若满足条件,则通过输出接口输出一个单元的数据;之后再判断状态是否满足,直到 20 个单元的数据都从输出接口输出。

```

LEA SI,DATA           ;取数据偏移地址
MOV CL,20             ;数据长度送 CL
AGAIN: MOV DX,0E54H
WAITT: IN AL,DX        ;读入状态值
AND AL,92H            ;屏蔽掉不相关位,仅保留 bit1、bit4 和 bit7 位状态
CMP AL,92H            ;判断 bit1、bit4 和 bit7 位是否全为 1
JNZ WAITT             ;不满足 bit1、bit4、bit7 位同时为 1 则等待
MOV DX,01FBH
MOV AL,[SI]
OUT DX,AL             ;满足条件则输出一个单元数据
INC SI                ;修改地址指针
LOOP AGAIN            ;若 20 个单元数据未传送完则循环

```

6.6 8088/8086 系统如何确定硬件中断服务程序的入口地址?

解：8088/8086 系统的硬件中断包括非屏蔽和可屏蔽两种中断请求。每个中断源都

有一个与之相对应的中断类型码 n 。系统规定所有中断服务子程序的首地址都必须放在中断向量表中,其在表中的存放地址 $= n \times 4$, (向量表的段基地址为 $0000H$)。即子程序的入口地址为 $(0000H: n \times 4)$ 开始的 4 个单元中,低位字(2 个字节)存放入口地址的偏移量,高位字存放入口地址的段基地址。

6.7 中断向量表的作用是什么? 如何设置中断向量表?

解: 中断向量表用于存放中断服务子程序的入口地址,位于内存的最低 1K 字节(即内存中 $00000H \sim 003FFH$ 区域),共有 256 个表项。

设置中断向量表就是将中断服务程序首地址的偏移量和段基址放入中断向量表中。如: 将中断服务子程序 CLOCK 的入口地址置入中断向量表的程序如下:

```
MOV AX,0000H
MOV DS,AX                      ;置中断向量表的段基地址
MOV SI,<中断类型码×4>          ;置存放子程序入口地址的偏移地址
MOV AX,OFFSET CLOCK
MOV [SI],AX                    ;将子程序入口地址的偏移地址送入中断向量表
MOV AX,SEG CLOCK
MOV [SI+2],AX                  ;将子程序入口地址的段基址送入中断向量表
```

6.8 INTR 中断和 NMI 中断有什么区别?

解: INTR 中断为可屏蔽中断,中断请求信号高电平有效。CPU 能否响应该请求要看中断允许标志位 IF 的状态,只有当 $IF=1$ 时,CPU 才可能响应中断。

NMI 中断为非屏蔽中断,请求信号为上升沿有效,对它的响应不受 IF 标志位的约束,CPU 只要当前指令执行结束就可以响应 NMI 请求。

6.9 在中断服务程序的入口处,为什么常常要使用开中断指令?

解: 中断服务程序分为两种,一种是在进入服务子程序后不允许被中断,另一种则可以被中断。在入口处使用开中断指令表示该中断服务程序是允许被中断的服务程序,即在进入服务子程序后允许 CPU 响应比它级别高的中断请求。

6.10 试说明 8088CPU 可屏蔽中断的响应过程。

解: 可屏蔽中断的响应过程主要分为 5 个步骤,即:

(1) 中断请求。外设需要在需要时向 CPU 的 INTR 端发出一个高电平有效的中断请求信号。

(2) 中断判优。若 $IF=1$,则识别中断源并找出优先级最高的中断先源予以响应,在其处理完后,再响应级别较低的中断源的请求。

(3) 中断响应。中断优先级确定后,发出中断的中断源中优先级别最高的中断请求就被送到 CPU 的中断。

(4) 中断处理。

(5) 中断返回。中断返回需执行中断返回指令 IRET,其操作正好是 CPU 硬件在中断响应时自动保护断点的逆过程。即 CPU 会自动地将堆栈内保存的断点信息弹出到 IP、CS 和 FLAG 中,保证被中断的程序从断点处继续往下执行。

6.11 CPU 满足什么条件能够响应可屏蔽中断?



解:

- (1) CPU 要处于开中断状态,即 $IF=1$,才能响应可屏蔽中断。
- (2) 当前指令执行结束。
- (3) 当前没有发生复位(RESET)、保持(HOLD)和非屏蔽中断请求(NMI)。
- (4) 若当前执行的指令是开中断指令(STI)和中断返回指令(IRET),则在执行完该指令后再执行一条指令,CPU 才能响应 INTR 请求。
- (5) 对前缀指令,如 LOCK、REP 等,CPU 会把它们和它们后面的指令看作一个整体,直到这个整体指令执行完,方可响应 INTR 请求。

6.12 8259 有哪几种优先级控制方式? 一个外中断服务程序的第一条指令通常为 STI,其目的是什么?

解: 8259 有两类优先级控制方式,即固定优先级和循环优先级方式。

CPU 响应中断时会自动关闭中断(使 $IF=0$)。若进入中断服务程序后允许中断嵌套,则需用指令开中断(使 $IF=1$),故一个外中断服务程序的第一条指令通常为 STI。

6.13 试编写 8259 芯片的初始化程序: 系统中仅有一片 8259 芯片,允许 8 个中断源边沿触发,不需要缓冲,一般全嵌套方式工作,中断向量为 40H。

解: 设 8259 芯片的地址为 FF00H ~ FF01H。其初始化顺序为: ICW1, ICW2, ICW3, ICW4。对单片 8259 芯片,不需初始化 ICW3。程序如下:

```
SET8259; MOV DX,0FF00H      ;置 ICW1,A0=0
          MOV AL,13H          ;单片,边沿触发,需要 ICW4
          OUT DX,AL
          MOV DX,0FF01H      ;置 ICW2,A0=1
          MOV AL,40H          ;中断向量码=40H
          OUT DX,AL
          MOV AL,03H          ;ICW1,8086/8088 模式,一般全嵌套,非缓冲
          OUT DX,AL
          HLT
```

6.14 单片 8259 芯片能够管理多少级可屏蔽中断? 若用 3 片级联能管理多少级可屏蔽中断?

解: 因 8259 芯片有 8 位可屏蔽中断请求输入端,故单片 8259 芯片能够管理 8 级可屏蔽中断。若用 3 片级联,即 1 片用作主控芯片,两片作为从属芯片,每一片从属芯片可管理 8 级,则 3 片级联共可管理 22 级可屏蔽中断。

6.15 8259A 芯片的中断请求有哪两种触发方式? 对请求信号有什么要求?

解: 根据实际需要,8259 芯片的中断请求有两种触发方式,即: 边沿触发方式和电平触发方式。

对边沿触发,要求在 8259A 芯片的引脚 IR_n 上出现一个上升沿的信号,高电平并不表示有中断请求。而对电平触发方式,要求在 8259A 芯片的引脚 IR_n 上要出现高电平。当然,此时应注意高电平的宽度,以免引起不应该有的第二次中断。

6.16 具备何种条件能够作输入接口? 何种条件能够作输出接口?

解：对输入接口要求具有对数据的控制能力，对输出接口要求具有对数据的锁存能力。

6.17 已知 $(SP) = 0100H$, $(SS) = 3500H$, $(CS) = 9000H$, $(IP) = 0200H$, $(00020H) = 7FH$, $(00021H) = 1AH$, $(00022)H = 07H$, $(00023)H = 6CH$, 在地址为 $90200H$ 开始的连续两个单元中存放一条两字节指令 $INT\ 8$ 。试指出在执行该指令并进入相应的中断例程时, SP 、 SS 、 IP 、 CS 以及 SP 所指向的字单元的内容是什么?

解：CPU 在响应中断请求时首先要进行断点保护, 即要依次将 $FLAGS$ 和 INT 下一条指令的 CS 、 IP 寄存器内容压入堆栈, 亦即栈顶指针减 6, 而 SS 的内容不变。 INT 指令是一条两字节指令, 故其下一条指令的 $IP = 0200H + 2 = 0202H$ 。

中断服务子程序的入口地址则存放在中断向量表 (8×4) 所指向的连续 4 个单元中。所以, 在执行中断指令并进入相应的中断例程时, 以上各寄存器的内容分别为:

$$SP = 0100H - 6 = 00FAH$$

$$SS = 3500H$$

$$IP = [8 \times 4]$$

$$CS = [(8 \times 4) + 2]$$

$$[SP] = 0200H + 2 = 0202H$$



7.1 一般来讲,接口芯片的读写信号应与系统的哪些信号相连?

解:一般来讲,接口芯片的读写信号应与系统总线信号中的 \overline{IOR} (接口读)或 \overline{IOW} (接口写)信号相连。

7.2 试说明 8253 芯片的六种工作方式。其时钟信号 CLK 和门控信号 GATE 分别起什么作用?

解:可编程定时/计数器 8253 具有六种不同的工作方式,其中:

方式 0: 软件启动、不自动重复计数。在写入控制字后 OUT 端变低电平,计数结束后 OUT 端输出高电平,可用来产生中断请求信号,故也称为计数结束产生中断的工作方式。

方式 1: 硬件启动、不自动重复计数。所谓硬件启动是在写入计数初值后并不开始计数,而是要等门控信号 GATE 出现由低到高的跳变后,在下一个 CLK 脉冲的下降沿才开始计数,此时 OUT 端立刻变为低电平。计数结束后,OUT 端输出高电平,得到一个宽度为计数初值 N 个 CLK 脉冲周期宽的负脉冲。

方式 2: 既可软件启动,也可以硬件启动。可自动重复计数。

在写入控制字后,OUT 端变为高电平。计数到最后一个时钟脉冲时 OUT 端变为低电平,再经过一个 CLK 周期,计数值减到零,OUT 又恢复为高电平。之后再自动装入计数初值,并重新开始新一轮计数。方式 2 下 OUT 端会连续输出宽度为 T_{CLK} 的负脉冲,其周期为 $N \times T_{CLK}$,所以方式 2 也称为分频器,分频系数为计数初值 N 。

方式 3: 也是一种分频器,也可有两种启动方式,自动重复计数。当计数初值 N 为偶数时,连续输出对称方波(即 $N/2$ 个 CLK 脉冲低电平, $N/2$ 个 CLK 脉冲高电平),频率为 $(1/N) \times F_{CLK}$ 。若 N 为奇数,则输出波形不对称,其中 $(N+1)/2$ 个时钟周期高电平, $(N-1)/2$ 个时钟周期低电平。

方式 4 和方式 5 都是在计数结束后输出一个 CLK 脉冲周期宽的负脉冲,且均为不自动重复计数方式。区别在方式 4 是软件启动,而方式 5 为硬件启动。

时钟信号 CLK 为 8253 芯片的工作基准信号。GATE 信号为门控信号。在软件启动时要求 GATE 在计数过程中始终保持高电平;而对硬件启动的工作方式,要求在写入计数初值后 GATE 端出现一个由低到高的正跳变,启动计数。

7.3 8253 可编程定时/计数器有两种启动方式,在软件启动时,要使计数正常进行,

GATE 端必须为()电平,如果是硬件启动呢?

解:在软件启动时,要使计数正常进行,GATE 端必须为(高)电平;如果是硬件启动,则要在写入计数初值后使 GATE 端出现一个由低到高的正跳变,以启动计数。

7.4 若 8253 芯片的接口地址为 D0D0H~D0D3H,时钟信号频率为 2MHz。现利用计数器 0、1、2 分别产生周期为 $10\mu\text{s}$ 的对称方波及每 1ms 和 1s 产生一个负脉冲,试画出其与系统的电路连接图,并编写包括初始化在内的程序。

解:根据题目要求可知,计数器 0(CNT0)工作于方式 3,计数器 1(CNT1)和计数器 2 (CNT2)工作于方式 2。时钟频率 2MHz,即周期为 $0.5\mu\text{s}$,从而得出各计数器得计数初值分别为:

CNT0:
 $10\mu\text{s}/0.5\mu\text{s}=20$

CNT1:
 $1\text{ms}/0.5\mu\text{s}=2000$

CNT2:
 $1\text{s}/0.5\mu\text{s}=2\times 10^6$

显然,计数器 2 的计数初值已超出了 16 位数的表达范围,需经过一次中间分频,可将 OUT1 端的输出脉冲作为计数器 2 的时钟频率。这样,CNT2 的计数初值就等于 $1\text{s}/1\text{ms}=1000$ 。线路连接如图 1-7-1 所示。

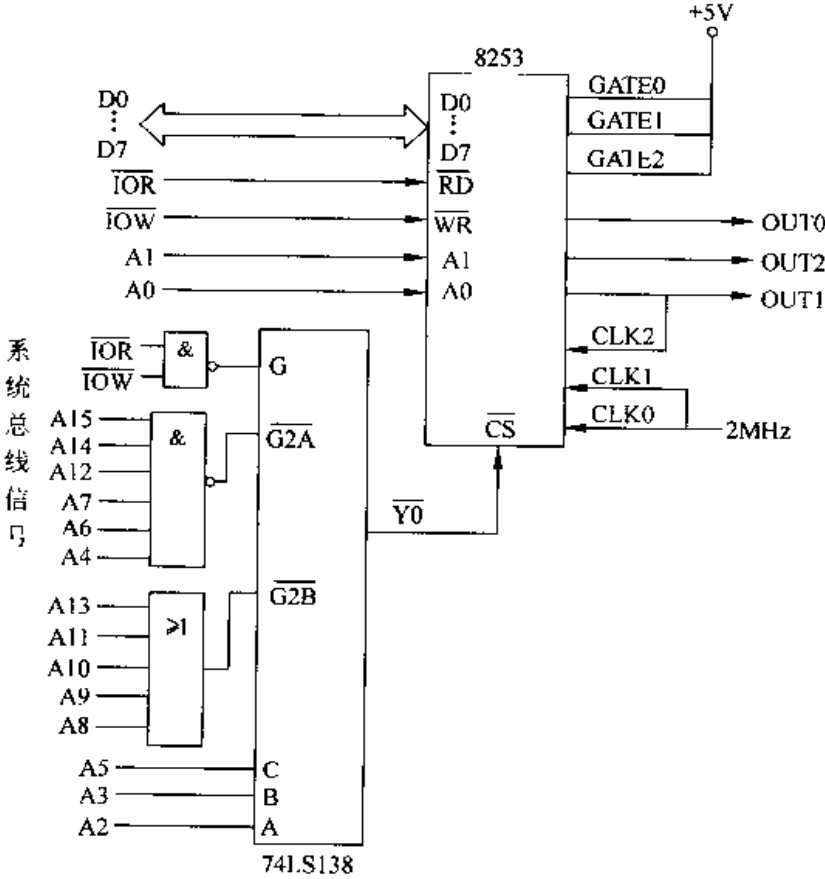


图 1-7-1

8253 的初始化程序如下:

```
MOV DX,0D0D3H
MOV AL,16H           ;计数器 0,低 8 位计数,方式 3
OUT DX,AL
MOV AL,74H           ;计数器 1,双字节计数,方式 2
OUT DX,AL
MOV AL,0B4H          ;计数器 2,双字节计数,方式 2
OUT DX,AL
MOV DX,0D0D0H
MOV AL,20             ;送计数器 0 计数初值
OUT DX,AL
MOV DX,0D0D1H
MOV AX,2000           ;送计数器 1 计数初值
OUT DX,AL
MOV AL,AH
OUT DX,AL
MOV DX,0D0D2H
MOV AX,1000           ;送计数器 2 计数初值
OUT DX,AL
MOV AL,AH
OUT DX,AL
```

7.5 某一计算机应用系统采用 8253 芯片的计数器 0 作频率发生器,输出频率为 500Hz;用计数器 1 产生 1000Hz 的连续方波信号,输入 8253 的时钟频率为 1.19MHz。试问:初始化时送到计数器 0 和计数器 1 的计数初值分别为多少?计数器 1 工作于什么方式下?

解:计数器 0 工作于方式 2,其计数初值 $=1.19\text{MHz}/500\text{Hz}=2380$

计数器 1 工作于方式 3,起计数初值 $=1.19\text{MHz}/1\text{KHz}=1190$

7.6 若所用 8253 芯片用软件产生一次性中断,最好采用哪种工作方式?现用计数器 0 对外部脉冲计数,当计满 10000 个脉冲时产生中断,请写出工作方式控制字及计数值。

解:若 8253 用软件产生一次性中断,最好采用方式 0,即计数结束产生中断的工作方式。其方式控制字为:00110000B

计数初值=10000

7.7 试比较并行通信与串行通信的特点。

解:并行通信是在同一时刻发送或接收一个数据的所有二进制位。其特点是接口数据的通道宽,传送速度快,效率高。但硬件设备的造价较高,常用于高速度、短传输距离的场合。

串行通信是将数据逐位的传送。其特点是传送速度相对较慢,但设备简单,需要的传输线少,成本较低。所以常用于远距离通信。



7.8 8255 各端口可以工作在几种方式下? 当端口 A 工作在方式 2 时, 端口 B 和 C 工作于什么方式下?

解: 8255 各端口均可以工作在方式 0 和方式 1 下, 而 A 口则可以工作在方式 0、方式 1 及方式 2 三种方式下。当端口 A 工作在方式 2 时, 端口 B 和端口 C 的剩余端口可工作于方式 0 或方式 1。

7.9 在对 8255 的 C 口进行初始化为按位置位或复位时, 写入的端口地址应是 () 地址。

解: 应是(8255 的内部控制寄存器)地址。

7.10 某 8255 芯片的地址范围为 A380H~A383H, 工作于方式 0, A 口、B 口为输出口, 现欲将 PC4 置“0”, PC7 置“1”, 试编写初始化程序。

解: 该 8255 芯片的初始化程序包括置方式控制字及 C 口的按位操作控制字。程序如下:

```
MOV DX, 0A383H      ;内部控制寄存器地址送 DX
MOV AL, 80H          ;方式控制字
OUT DX, AL
MOV AL, 08H          ;PC4 置 0
OUT DX, AL
MOV AL, 0FH          ;PC7 置 1
OUT DX, AL
```

7.11 设 8255 芯片的接口地址范围为 03F8H~03FBH, A 组 B 组均工作于方式 0, A 口作为数据输出口, C 口低 4 位作为控制信号输入口, 其他端口未使用。试画出该片 8255 芯片与系统的电路连接图, 并编写初始化程序。

解: 8255 芯片与系统的电路连接如图 1-7-2 所示。

由题目知, 不需对 C 口置位控制字, 只需对 8255 置方式控制字, 故其初始化程序如下:

```
MOV DX, 03FBH
MOV AL, 81H
OUT DX, AL
```

7.12 已知某 8088 微机系统的 I/O 接口电路框图如教材中图 7-47 所示。试完成:

(1) 根据图中接线, 写出 8255 芯片、8253 芯片各端口的地址。

(2) 编写 8255 芯片和 8253 芯片的初始化程序。其中, 8253 芯片的 OUT1 端输出 100Hz 方波, 8255 芯片的 A 口为输出, B 口和 C 口为输入。

(3) 为 8255 芯片编写一个 I/O 控制子程序, 其功能为: 每调用一次, 先检测 PC0 的状态, 若 PC0=0, 则循环等待; 若 PC0=1, 可从 PB 口读取当前开关 K 的位置(0~7), 经转换计算从 A 口的 PA0~PA3 输出该位置的二进制编码, 供 LED 显示。

解:

(1) 8255 芯片的地址范围为: 8000H~FFFFH



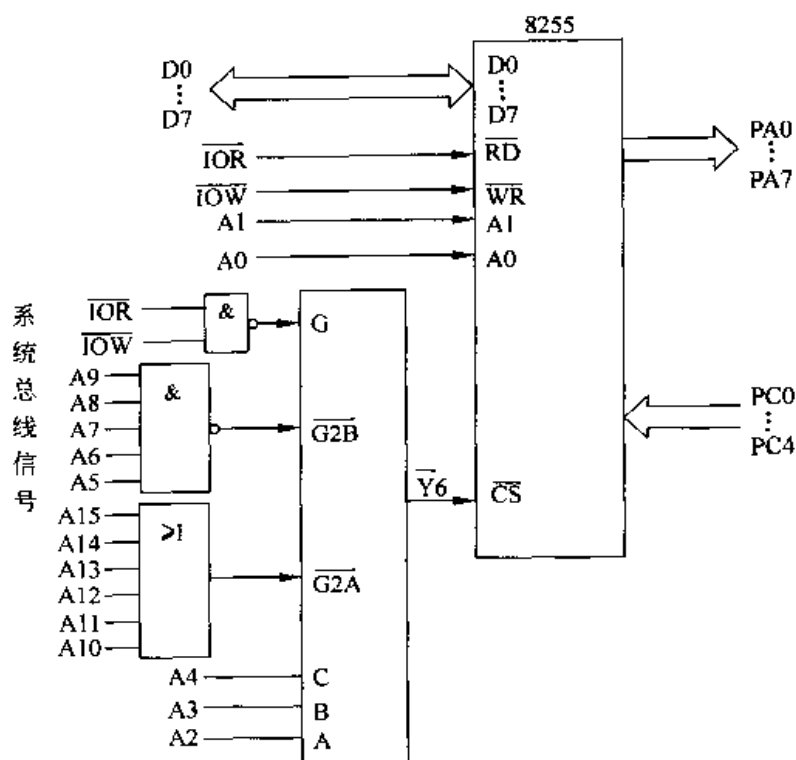


图 1-7-2

8253 芯片的地址范围为: 0000H~7FFFH

(2)

;初始化 8255 芯片

MOV DX,8003H

MOV AL,8BH

;方式控制字,方式 0,A 口输出,B 口和 C 口输入

OUT DX,AL

;初始化 8253

MOV DX,0003H

;内部寄存器口地址

MOV AL,76H

;计数器 1,先写低 8 位/后写高 8 位,方式 3,二进制计数

OUT DX,AL

MOV DX,0001H

;计数器 1 端口地址

MOV AX,10000

;设计数初值=10000

OUT DX,AL

MOV AL,AH

OUT DX,AL

(3)

;8255 芯片的控制子程序

;定义显示开关位置的字形译码数据

DATA SEGMENT

BUFFER DB 3FH,06H,5BH,0FH,66H,6DH,7CH,07H

```

DATA ENDS
;
CODE SEGMENT
    ASSUME CS: CODE, DS: DATA
MAIN  PROC
    PUSH DS
    MOV AX, DATA
    MOV DS, AX
    CALL DISP
    POP DX
    RET
MAIN  ENDP
; 输出开关位置的二进制码程序
DISP  PROC
    PUSH CX
    PUSH SI
    XOR CX, CX
    CLC
    LEA SI, BUFFER
    MOV DX, 8002H
WAITT: IN AL, DX
    TEST AL, 01H
    JZ WAITT
    MOV DX, 8001H
    IN AL, DX
NEXT:  SHR AL, 1
    INC CX
    JC NEXT
    DEC CX
    ADD SI, CX
    MOV AL, [SI]
    MOV DX, 8000H
    OUT DX, AL
    POP SI
    POP CX
    RET
DISP ENDP
CODE  ENDS
    END MAIN

```

7.13 试说明串行通信的数据格式。

解：串行通信通常包括两种方式，即同步通信和异步通信。二者因通信方式的不同



而有不同的数据格式,其数据格式可参见教材第 326 页及 327 页图 7-34 和图 7-35。

7.14 串行通信接口芯片 8250 的给定地址为 83A0H~83A7H,试画出其与 8088 系统总线的连接图。若采用查询方式由该 8250 芯片发送当前数据段、偏移地址为 BUFFER 的顺序 100B 的数据,试编写发送程序。

解:8250 芯片与系统连接如图 1-7-3 所示。假设要写入除数锁存器的除数为 96,即 0060H。

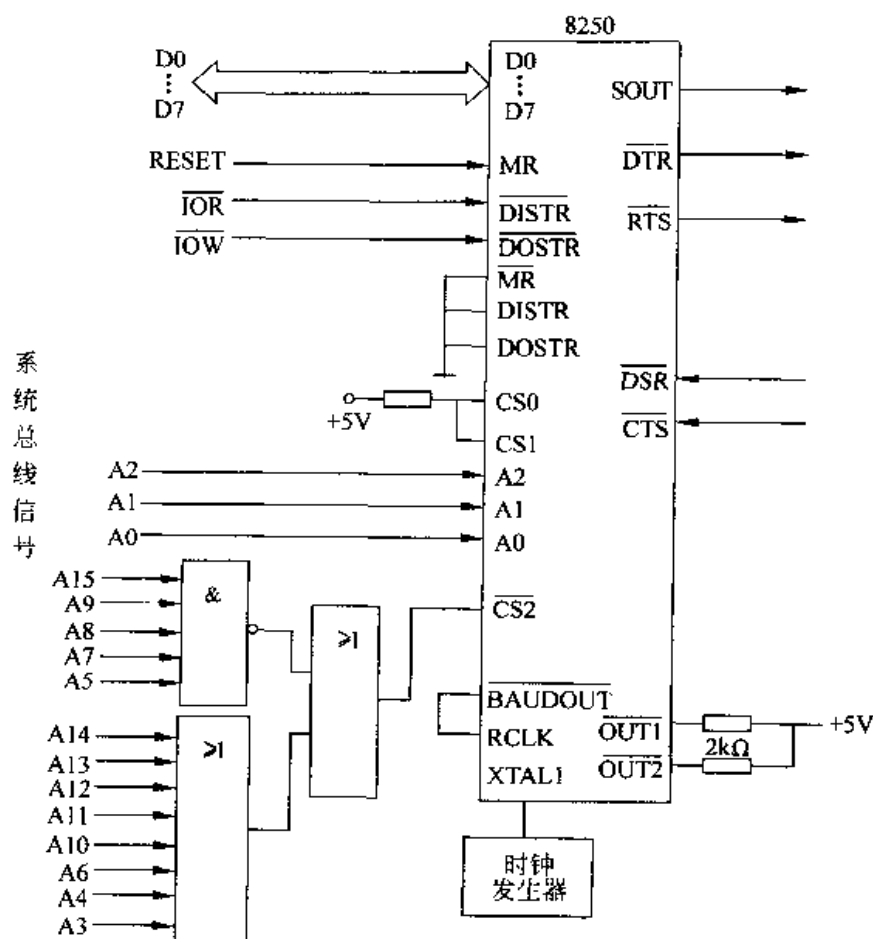


图 1-7-3

程序如下:

;8250 芯片的初始化程序

```
BEGIN:  MOV DX,83A3H      ;通信控制寄存器地址
        MOV AL,80H       ;使通信控制寄存器的 D7=1
        OUT DX,AL
        MOV DX,83A0H     ;除数锁存器地址
        MOV AL,60H       ;除数为 0060H
        OUT DX,AL        ;写除数低 8 位
        INC DX
        MOV AL,0
```

```

        OUT DX,AL                ;写除数高8位
        MOV DX,83A3H            ;通信控制寄存器地址
        MOV AL,0AH              ;1位停止位,7位数据位,奇校验
        OUT DX,AL              ;初始化通信控制寄存器
        MOV DX,83A4H            ;MODEM控制寄存器地址
        MOV AL,03H              ;使DTR和RTS有效
        OUT DX,AL
        MOV DX,83A1H            ;中断允许寄存器地址
        MOV AL,0                ;禁止所有中断
        OUT DX,AL

;数据发送程序
SENDATA: LEA SI,BUFFER
        MOV CX,100
WAITT:  MOV DX,83A5H            ;通信状态寄存器地址
        IN AL,DX
        TEST AL,20H            ;检查发送数据寄存器是否空
        JZ WAITT
        MOV DX,83A0H            ;发送数据寄存器地址
        MOV AL,[SI]
        OUT DX,AL              ;发送一个字节
        INC SI
        DEC CX
        JNZ WAITT

```

7.15 上题中若采用中断方式接收数据,试编写将接收到的数据放在数据段 DATA 单元的中断服务子程序。

解:同上题一样,首先要对 8250 芯片进行初始化,其初始化程序与上题基本相同,只是要将中断允许寄存器的 D_0 位置“1”。程序如下:

```

        BEGIN:  ...
                :
                MOV DX,83A1H      ;中断允许寄存器地址
                MOV AL,01H        ;允许接收数据寄存器满产生中断
                OUT DX,AL
                STI
;接收数据子程序
RECDATA:  PUSH AX
          PUSH BX
          PUSH DX
          PUSH DS
          MOV DX,83A5H
          IN AL,DX
          MOV AH,AL              ;保存接收状态

```



```

MOV DX,83A0H
IN AL,DX           ;读入接收到的数据
AND AL,7FH
TEST AH,1EH       ;检查有无错误产生
JNZ ERROR
SAVEDATA: MOV DX,SEG DATA
MOV DS,DX
MOV SI,OFFSET DATA
MOV [SI],AL
MOV DX,中断控制器 8259 端口地址
MOV AL,20H        ;发送中断结束命令给 8259
OUT DX,AL
POP DS
POP DX
POP BX
POP AX
STI
IRET

```


8.1 试说明将一个工业现场的非电物理量转换为计算机能够识别的数字信号主要需经过哪几个过程?

解: 将工业现场的非电物理量转换为计算机能够识别的数字信号的过程就是模拟量的输入通道,主要需经过以下几个环节:

(1) 由传感器将非电的物理量转换为电信号或可进一步处理的电阻值、电压值等非电量。

(2) 变送器将传感器输出的微弱电信号或电阻值等非电量转换成统一的电信号。

(3) 信号处理。去除叠加在变送器输出信号上的干扰信号,并将其进行放大或处理成与 A/D 转换器所要求的输入相适应的电压水平。

(4) 如果是多路模拟信号共享一个 A/D 转换器,则需添加多路转换开关。

(5) 采样保持。因完成一次 A/D 转换需要一定的时间,而转换期间要求保持输入信号不变,所以增加采样保持电路,以保证在转换过程中输入信号始终保持在其采样时的值。

(6) A/D 变换。将输入的模拟信号转换为计算机能够识别的数字信号。

8.2 什么是 A/D 变换器? 什么是 D/A 变换器? 它们的主要作用是什么?

解: A/D 变换器是模拟量转换为数字量的集成电路芯片,在模拟量的输入信道中用于将工业现场采集的模拟信号转换为计算机能够识别的数字信号。常用于数据采集系统。

D/A 变换器的功能正好相反,它是将计算机输出的数字量转换为模拟信号,用以驱动执行机构。常用于死循环控制系统或信号发生器。

8.3 D/A 变换器主要有哪些技术指标? 影响其转换误差的主要因素是什么?

解: D/A 变换器主要技术指标有:分辨率、转换精度、转换时间、线性误差和动态范围等。影响其转换误差的主要因素除由位数产生的转换误差(即分辨率)外,非线性误差,温度系数误差、电源波动误差及运算放大器误差等。

8.4 对于一个 10 位的 D/A 变换器,其分辨率是多少? 如果输出满刻度电压值为 5V,其一个最低有效位对应的电压值等于多少?

解: D/A 变换器的分辨率 $= 1/(2^n - 1) \times 100\%$, 其中 n 为 D/A 变换器的位数。

所以,一个 10 位的 D/A 变换器的分辨率 $= 1/1023 \times 100\% \approx 0.0978\%$

(分辨率也可用 D/A 变换器的位数表示,即可以说该 D/A 变换器的分辨率是 10 位。)

若输出满刻度电压值为 5V,则其一个 LSB 对应的电压值 $= 5 / (2^n - 1) = 5 / 1023 \approx 4.89\text{mV}$

8.5 某一测控系统要求计算机输出模拟控制信号的分辨率必须达到 0.1%,则应选用的 D/A 芯片的位数至少是多少?

解: 因为 D/A 芯片的分辨率 $= 1 / (2^n - 1) \times 100\%$,所以要使计算机输出模拟控制信号的分辨率达到 0.1%,则应选用的 D/A 芯片的位数至少是 10 位。

8.6 DAC0832 芯片在逻辑上由哪几个部分组成? 可以工作在哪几种模式下? 不同工作模式在线路连接上有什么区别?

解: DAC0832 芯片在逻辑上包括一个 8 位的输入寄存器、一个 8 位的 DAC 寄存器和一个 8 位的 D/A 转换器等 3 个部分。可以工作在 3 种模式下,即:双缓冲模式、单缓冲模式及直通模式。

在双缓冲模式下,CPU 对 DAC0832 要进行两步写操作:先将数据写入输入寄存器,再将输入寄存器的内容写入 DAC 寄存器,并进行一次变换。即此时 DAC0832 占用两个接口地址,可将 ILE 固定接 +5V, $\overline{\text{WR1}}$ 、 $\overline{\text{WR2}}$ 接到 $\overline{\text{IOW}}$, $\overline{\text{CS}}$ 和 $\overline{\text{XFER}}$ 分别接到两个端口的地址译码信号线。

当工作于单缓冲模式时,数据写入输入寄存器后将直接进入 DAC 寄存器,并进行一次变换。此时 DAC0832 仅占用一个接口地址,故在线路连接上,只需通过 ILE, $\overline{\text{WR1}}$ 和 $\overline{\text{CS}}$ 进行控制,通常仍将 ILE 固定接 +5V, $\overline{\text{WR1}}$ 接 $\overline{\text{IOW}}$, $\overline{\text{CS}}$ 接到地址译码器的输出端。 $\overline{\text{WR2}}$ 和 $\overline{\text{XFER}}$ 直接接地。

直通工作方式是将 $\overline{\text{CS}}$ 、 $\overline{\text{WR1}}$ 、 $\overline{\text{WR2}}$ 以及 $\overline{\text{XFER}}$ 引脚都直接接数字地,ILE 接 +5V,芯片处于直通状态,只要有数字量输入,就立刻转换为模拟量输出。

8.7 如果要求同时输出 3 路模拟量,则 3 片同时工作的 DAC0832 芯片最好采用哪一种工作模式?

解: 考虑到 3 路模拟量需同步输出,可使 3 片 DAC0832 芯片工作于双缓冲模式。使 3 路数字量先分别锁存到 3 片 DAC0832 芯片的输入寄存器,再同时打开各自的 DAC 寄存器,使 3 路模拟量同时输出。

8.8 某 8 位 D/A 变换器,输出电压为 0~5V。当输入的数字量为 40H、80H 时,其对应的输出电压分别是多少?

解: 当输出电压为 0V 时,对应的数字量输入为 00H;输出为 5V 时,输入为 FFH。所以,当输入的数字量为 40H、80H 时,其对应的输出电压分别约为 1.255V 和 2.451V。

8.9 ADC0809 芯片是完成什么功能的芯片? 试说明它的变换原理。

解: ADC0809 芯片是完成将输入的模拟量转换为数字量并输出的集成电路芯片。其工作原理为逐位反馈型(或称逐位逼近型)。内部主要由逐次逼近寄存器、D/A 转换器、电压比较器和一些时序控制逻辑电路等组成。逐次逼近寄存器的位数就是 0809 芯片的位数。转换开始前,先将逐次逼近寄存器各位清零,然后设其最高位为 1(即为 10000000B),逐次逼近寄存器中的数字量经 D/A 变换器转换为相应的模拟电压 V_c ,并与

模拟输入电压 V_x 进行比较,若 $V_x \geq V_c$,则逐次逼近寄存器中最高位的 1 保留,否则就将最高位清零。然后再使次高位置 1,进行相同的过程,直到逐次逼近寄存器的所有位都被确定。转换过程结束后,该寄存器中的二进制码就是 A/D 变换器的输出。

8.10 设 DAC0832 芯片工作在单缓冲模式下,端口地址为 034BH,输出接运算放大器。试画出其与 8088 系统的线路连接图,并编写输出三角波的程序段。

解: DAC0832 芯片工作在单缓冲模式下与系统的线路连接图如图 1-8-1 所示。

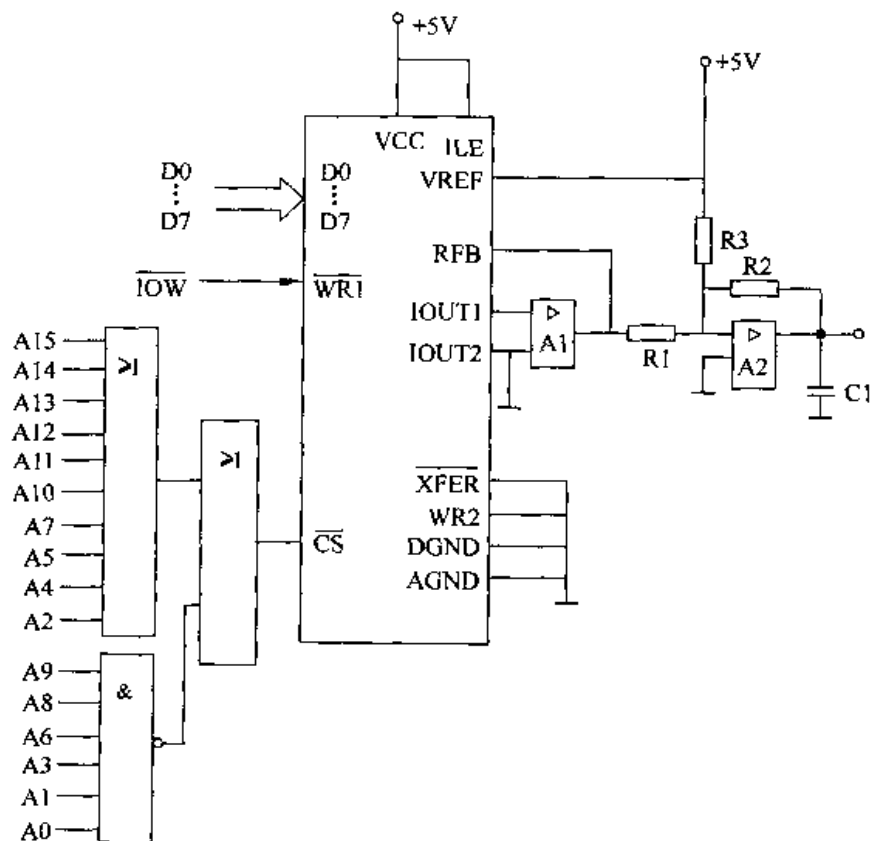


图 1-8-1

因 DAC0832 芯片为 8 位,故其最大输出对应的二进制码是 0FFH,而最小输出对应 00H。现利用该芯片输出连续的三角波的程序如下:

```
START: MOV DX,034BH
NEXT1: INC AL
      OUT DX,AL
      CMP AL,0FFH           ;比较是否达到最大值
      JNE NEXT1
NEXT2: DEC AL               ;达到最大值则减 1
      OUT DX,AL
      CMP AL,00H           ;比较是否达到最小值
      JNE NEXT2
      JMP NEXT1
```

8.11 对 8 位、10 位和 12 位的 A/D 变换器,当满刻度输入电压为 5V 时,其量化间



隔各为多少？绝对量化误差又为多少？

解：量化间隔分别为： $\Delta = 5\text{V}/255 \approx 19.6\text{mV}$

绝对量化误差： $\Delta/2 \approx 9.8$

$\Delta = 5\text{V}/1023 \approx 4.89\text{mV}$

$\Delta/2 \approx 2.45$

$\Delta = 5\text{V}/4095 \approx 1.22\text{mV}$

$\Delta/2 \approx 0.61$

8.12 某工业现场的三个不同点的压力信号经压力传感器、变送器及信号处理环节等分别送入 ADC0809 芯片的 IN0、IN1 和 IN2 端。计算机巡回检测这三点的压力并进行控制。试编写数据采集程序。

解：ADC0809 芯片的数据采集程序可参见教材第 363 页，只是教材中完成的是对 8 路模拟量的采集，本题目中只有 3 路，即第 363 页程序中的 CL 要赋值 3。

8.13 设被测温度的变化范围为 $0 \sim 100^\circ\text{C}$ ，若要求测量误差不超过 0.1°C ，应选用分辨率为多少位的 A/D 变换器？

解：由题目知， $(1/2)\Delta = 0.1$ ，即 $(1/2)(100/2^n - 1) = 0.1$

从而得： $n \approx 9$

即：至少应选用分辨率为 9 位的 A/D 变换器

8.14 某 11 位 A/D 变换器的引线及工作时序如教材第 365 页图 8-21 所示。利用不小于 $1\mu\text{s}$ 的后沿脉冲 (START) 启动变换。当 BUSY 端输出低电平时表示正在变换，BUSY 变高则变换结束。为获得变换好的二进制数据，必须使 OE 为低电平。现将该 A/D 变换器与 8255 相连，8255 芯片的地址范围为 $03\text{F}4\text{H} \sim 03\text{F}7\text{H}$ 。试画线路连接图，编写包括 8255 芯片初始化程序在内的、完成一次数据变换并将数据存放在 DATA 中的程序。

解：A/D 变换器通过 8255 与系统的线路连接如图 1-8-2 所示。

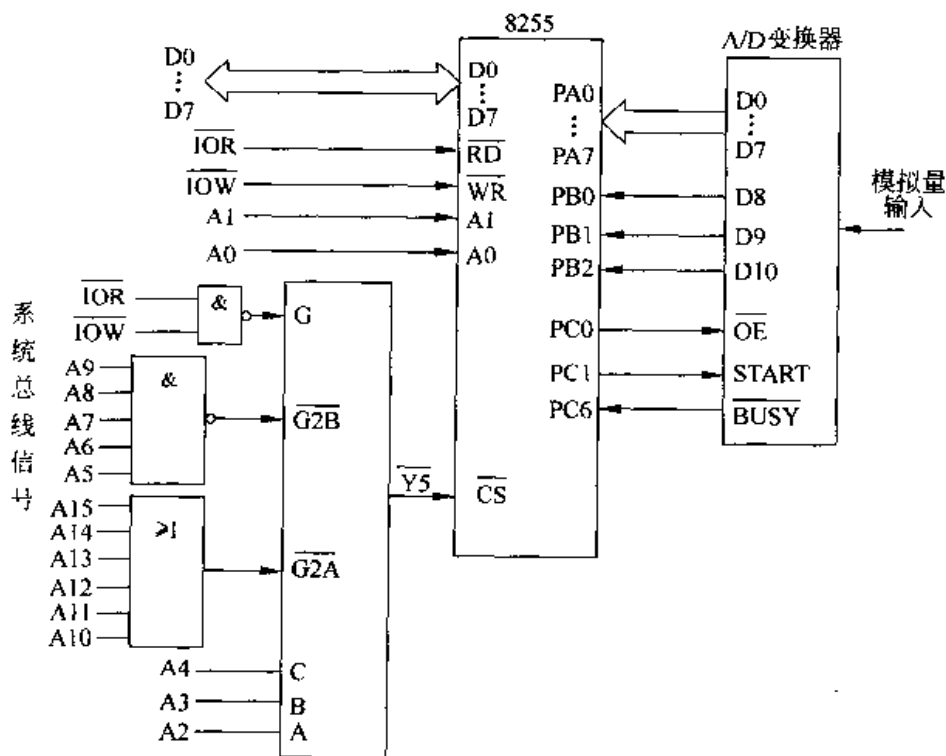


图 1-8-2

程序设计如下:

;8255 的初始化程序

INIT PROC NEAR

PUSH DX

PUSH AX

MOV DX,03F7H

MOV AL,9AH ;方式 0,A、B 口输入,C 口高 4 位输入,低 4 位输出

OUT DX,AL

MOV AL,01H

;PC₀初始置 1

OUT DX,AL

MOV AL,02H

OUT DX,AL

;PC₁初始置 0

POP AX

POP DX

RET

INIT ENDP

;完成一次数据采集程序

START: MOV AX,SEG DATA

MOV DS,AX

MOV SI,OFFSET DATA

CALL INIT

;初始化 8255

MOV DX,03F6H

MOV AL,03H

;输出 START 信号

OUT DX,AL

NOP

;空操作使 START 脉冲不小于 1 μ s

MOV AL,01H

OUT DX,AL

;空操作等待转换

WAITT: IN AL,DX

;读 BUSY 状态

AND AL,40H

JZ WAITT

;若 BUSY 为低电平则等待

AND AL,0FEH

OUT DX,AL

;EOC 端为高电平则输出读允许信号 OE=0

MOV DX,03F5H

IN AL,DX

;读入变换结果的高 3 位

MOV [SI],AL

;将转换的数字量送存储器

INC SI

;修改指针

MOV DX,03F4H

IN AL,DX

;读入变换结果的低 8 位

MOV [SI],AL

HLT

