

21 世纪高等学校计算机教育实用规划教材

微型计算机原理及应用  
**导教 · 导学 · 导考**  
(第 2 版)

史新福 秦晓红 主编  
秦晓红 冯 萍 孔庆芸 史新福 编著

清华大学出版社  
北 京

# 前 言

---

随着计算机技术的不断发展，计算机的使用也在不断地普及和深入。微机原理及应用、微机原理与接口技术、微型计算机软硬件及其应用等课程的教材所涉及的内容已经从 8 位机、16 位机，发展到了 32 位机。

微型计算机原理及应用这门课程内容多而杂，学习起来比较困难，而且又是一门实践性很强的课程，必须进行大量的练习，逐步掌握编程的思路、方法和技巧。为了适应教学需求，针对学生在学习中可能出现的问题，我们收集了大量的例题、习题和试题，精选后编写了本书，其主要内容是对常见题型范例进行了详细的分析和必要的评注，学生读后可对该课程内容有一个总体性的认识，提高综合分析问题、解决问题的能力。考虑到 32 位机和 16 位机是向上兼容的，而且当前各高校普遍使用的是 16 位微机实验设备，所以书中保留了部分有关 8086 的内容。

本书由史新福、秦晓红主编，冯萍和孔庆芸参加了本书的编写及校对工作，在此表示感谢！

由于编者水平有限，书中难免有不少缺点和问题，恳请读者批评指正。

编 者  
2006 年 7 月

# 目 录

---

第 1 章 微型计算机的基本结构和运算基础	1
1.1 重点内容提要	1
1.2 常考题型范例精解	2
1.3 习题及全解	4
第 2 章 Intel 32 位微处理器	8
2.1 重点内容提要	8
2.2 常考题型范例精解	10
2.3 习题及全解	11
第 3 章 80x86 寻址方式和指令系统	15
3.1 重点内容提要	15
3.2 常考题型范例精解	19
3.3 习题及全解	26
第 4 章 汇编语言、程序设计及其与高级语言调用	38
4.1 重点内容提要	38
4.2 常考题型范例精解	42
4.3 习题及全解	55
第 5 章 内存储器及其管理	72
5.1 重点内容提要	72
5.2 常考题型范例精解	74
5.3 习题及全解	75
第 6 章 微型计算机的输入/输出	77
6.1 重点内容提要	77
6.2 常考题型范例精解	79
6.3 习题及全解	81

第7章 中断 .....	86
7.1 重点内容提要 .....	86
7.2 常考题型范例精解 .....	89
7.3 习题及全解 .....	94
第8章 总线技术 .....	100
8.1 重点内容提要 .....	100
8.2 常考题型范例精解 .....	100
8.3 习题及全解 .....	102
第9章 可编程接口芯片及其与 CPU 的接口 .....	103
9.1 重点内容提要 .....	103
9.2 常考题型范例精解 .....	106
9.3 习题及全解 .....	114
第10章 常用外围设备及人机接口 .....	137
10.1 重点内容提要 .....	137
10.2 常考题型范例精解 .....	138
10.3 习题及全解 .....	138
第11章 D/A 和 A/D 转换器及其与 CPU 的接口 .....	140
11.1 重点内容提要 .....	140
11.2 常考题型范例精解 .....	142
11.3 习题及全解 .....	147
第12章 微型计算机的应用 .....	153
12.1 重点内容提要 .....	153
12.2 常考题型范例精解 .....	153
12.3 习题及全解 .....	160
附录 硕士研究生入学“微机原理及应用”考试试题及参考答案 .....	164
附录1 硕士研究生入学考试模拟试题及答案 .....	164
附录2 2000年某高校硕士研究生入学“微机原理及应用” 考试试题及答案 .....	171
附录3 2001年某高校硕士研究生入学“微机原理及应用” 考试试题及答案 .....	178

附录 4	2002 年某高校硕士研究生入学 “ 微机原理及应用 ” 考试试题及答案 .....	186
附录 5	2003 年某高校硕士研究生入学 “ 微机原理及应用 ” 考试试题及答案 .....	193
附录 6	2004 年某高校硕士研究生入学 “ 微机原理及应用 ” 考试试题及答案 .....	200
附录 7	2005 年某高校硕士研究生入学 “ 微机原理及应用 ” 考试试题及答案 .....	207
附录 8	2006 年某高校硕士研究生入学 “ 微机原理及应用 ” 考试试题及答案 .....	214
参考文献 .....		223

# 第 1 章

## 微型计算机的基本结构和运算基础

### 1.1 重点内容提要

#### （一）微型计算机的基本结构

微型计算机系统是由硬件系统、软件系统以及通信网络系统组成的一个整体系统。

微型计算机硬件系统是指构成微机的所有实体部件的集合。这些部件包括集成电路芯片、机械等物理部件，通常称为“硬件”。微型计算机的硬件主要由输入设备、输出设备、运算器、存储器和控制器 5 个部分组成。

微型计算机的软件系统可分为系统软件和应用软件。

#### （二）数制转换

在计算机内部，一切信息的存取、处理和传送均采用二进制形式。但为了方便，常采用八进制、十六进制和十进制。这样它们之间就存在一种对应的转换关系。

任意进制数转换为十进制数就是按权展开求多项式之和。十进制转换为二进制数时，对于整数部分，采用除基数取余数法；对于小数部分，则采用乘基数取整数法。八进制、十六进制和二进制之间的转换非常简单，分别按 3 位二进制数对应 1 位八进制数、4 位二进制数对应 1 位十六进制数的关系转换即可。

#### （三）计算机中带符号数的表示方法

在计算机中表示的数叫机器数。数有带符号数和不带符号数之分，在计算机中，对于带符号数，其正和负必须符号化。带符号数的机器数最常用的有原码、反码和补码 3 种形式。

正数的原码、反码和补码形式一样，其符号位都为“0”，数值位同真值；对于负数，其符号位都为“1”，而数值位有区别，原码的数值位同真值，反码的数值位是其真值按位取反，补码的数值位为其反码末位加 1。

补码表示的机器数其符号位能和有效数值位一起参加数值计算，并能使减法运算变为加法运算，从而简化运算器的线路设计。

补码加法规则： $[X + Y]_{\text{补}} = [X]_{\text{补}} + [Y]_{\text{补}}$

$$[X - Y]_{\text{补}} = [X]_{\text{补}} + [-Y]_{\text{补}}$$

溢出判别：计算机在进行补码运算时，由于位数的限制可能产生溢出。对于带符号

数而言,溢出是由于数值位侵犯符号位造成的,可采用双高位法判别溢出。显然,在两个同号数相加或两个异号数相减时才可能溢出。溢出时,符号位的“1”和“0”已不能准确表示数的符号了。对于不带符号数,因所有位均是有效数值,可根据最高位是否产生进位或借位来判别溢出。

#### (四) 计算机中数的小数点表示方法

计算机中数的小数点表示方法有定点表示法和浮点表示法。

在定点表示法中,小数点在数中的位置是固定不变的。而在浮点表示法中,小数点的位置是不固定的,用阶码和尾数来表示。通常尾数为纯小数,阶码为整数,尾数和阶码均为带符号数。尾数的符号代表数的正、负;阶码的符号表明小数点的位置。

#### (五) 常用的二进制编码

计算机只能识别二进制数“0”和“1”,因此在计算机中任何信息都是通过一定的编码实现的。常用的二进制编码有BCD码、ASCII码、汉字国标码等。

## 1.2 常考题型范例精解

**例 1.1** 将十进制数 25.625 转换为二进制数、八进制数及十六进制数。

**【解】** 十进制数转换为二进制数时,对于整数部分,采用除 2 取余数法,即逐次用 2 去除要转换的十进制数,直至商为 0,每次所得的余数即为二进制数码,最先得到的为整数的最低有效位  $K_0$ ,最后得到的是整数的最高有效位  $K_{n-1}$ 。对于小数部分,采用乘 2 取整数法,即逐次用 2 去乘要转换的十进制小数,将每次所得的整数 0 或 1,依次记作  $K_{-1}, K_{-2}, \dots$ 。注意,十进制小数并不是都能用有限位的二进制数精确地表示的,这时只要根据精度要求,转换到一定的位数即可。

2   25	余数		整数
2   12	$K_0 = 1$	$0.625 \times 2 = 1.25$	$K_{-1} = 1$
2   6	$K_1 = 0$	$0.25 \times 2 = 0.5$	$K_{-2} = 0$
2   3	$K_2 = 0$	$0.5 \times 2 = 1$	$K_{-3} = 1$
2   1	$K_3 = 1$		
0	$K_4 = 1$		

故 25.625 对应的二进制数为 11001.101B。

八进制、十六进制和二进制之间的转换是非常简单的,分别按 3 位、4 位二进制数对应转换即可。方法是以小数点为界,整数部分自右至左,小数部分自左至右分组,若转换为八进制,3 位为一组,若转换为十六进制,4 位为一组,不足时补 0。

本题中:

$$11001.101B = 011, 001.101B = 31.5Q$$

$$11001.101B = 0001, 1001.1010B = 19.AH$$

所以, 25.625 对应的二进制数、八进制数及十六进制数分别为 11001.101B、31.5Q 和 19.AH。

**例 1.2** 将二进制数 10110B、八进制数 125Q 及十六进制数 5AF.8H 转换为十进制数。

**【解】** 将非十进制数转换为十进制数时, 一般是按其定义展开为多项式, 将系数与权用十进制表示, 然后进行相应的四则运算即可得到运算结果。

$$\begin{aligned} 10110\text{B} &= 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 = \\ &16 + 0 + 4 + 2 + 0 = 22\text{D} \end{aligned}$$

$$125\text{Q} = 1 \times 8^2 + 2 \times 8^1 + 5 \times 8^0 = 64 + 16 + 5 = 85\text{D}$$

$$\begin{aligned} 5\text{AF}.8\text{H} &= 5 \times 16^2 + 10 \times 16^1 + 15 \times 16^0 + 8 \times 16^{-1} = \\ &1280 + 160 + 15 + 0.5 = 1455.5\text{D} \end{aligned}$$

**例 1.3** 已知  $[X]_{\text{原}} = 11101011\text{B}$ ,  $[Y]_{\text{原}} = 01001010\text{B}$ , 求  $[X + Y]_{\text{补}}$  和  $[X - Y]_{\text{补}}$ , 并判断结果是否溢出。

**【分析】** 本题给出的已知条件是 X 和 Y 的原码形式, 根据补码运算规则  $[X + Y]_{\text{补}} = [X]_{\text{补}} + [Y]_{\text{补}}$ ,  $[X - Y]_{\text{补}} = [X]_{\text{补}} + [-Y]_{\text{补}}$ , 所以必须先求出  $[X]_{\text{补}}$ 、 $[Y]_{\text{补}}$  和  $[-Y]_{\text{补}}$ 。

**【解】** 由于正数的补码形式和原码形式一致, 负数的补码形式其符号位为 1, 数值部分是真值按位求反加 1, 所以:

$$[X]_{\text{补}} = 10010101\text{B} \quad [Y]_{\text{补}} = [Y]_{\text{原}} = 01001010\text{B}$$

在求  $[-Y]_{\text{补}}$  时, 只要对其相反数的补码连同符号位一起求反加 1 即可, 即  $[-Y]_{\text{补}} = 10110110\text{B}$ 。那么:

$$[X + Y]_{\text{补}} = [X]_{\text{补}} + [Y]_{\text{补}} = 10010101\text{B} + 01001010\text{B} = 11011111\text{B}$$

$$[X - Y]_{\text{补}} = [X]_{\text{补}} + [-Y]_{\text{补}} = 10010101\text{B} + 10110110\text{B} = 01001011\text{B}$$

可采用双高位法判断结果是否溢出。具体方法为用  $C_S$  表示符号位的进位情况;  $C_P$  表示最高数值位的进位情况。当有进位时,  $C_S$  或  $C_P$  为 1, 否则为 0; 溢出判别式 P 对两者进行异或运算, 即  $P = C_S \oplus C_P$ , 当其为 1 时, 表示溢出, 当其为 0 时, 表示不溢出。

本题中:

$$\begin{array}{r} 10010101\text{B} \quad [X]_{\text{补}} \\ + 01001010\text{B} \quad [Y]_{\text{补}} \\ \hline 11011111\text{B} \end{array}$$

$$C_S = 0, C_P = 0, P = C_S \oplus C_P = 0, \text{无溢出。}$$

因此  $[X + Y]_{\text{补}} = [X]_{\text{补}} + [Y]_{\text{补}} = 11011111\text{B}$  无溢出, 结果正确。

$$\begin{array}{r} 10010101\text{B} \quad [X]_{\text{补}} \\ + 10110110\text{B} \quad [-Y]_{\text{补}} \\ \hline 01001011\text{B} \end{array}$$

$$C_S = 1, C_P = 0, P = C_S \oplus C_P = 1, \text{有溢出。}$$

所以  $[X - Y]_{\text{补}} = [X]_{\text{补}} + [-Y]_{\text{补}} = 01001011\text{B}$  有溢出, 结果出错。



## 1.3 习题及全解

### 一、填空题

1. 与十进制数 45 等值的二进制数是\_\_\_\_\_。
2. 与二进制数 101110 等值的十六进制数是\_\_\_\_\_。
3. 若  $X = -1$ ,  $Y = -127$ , 字长  $n = 16$ , 则:  
 $[X]_{\text{补}} = \text{\_\_\_\_\_\_H}$ ,  $[Y]_{\text{补}} = \text{\_\_\_\_\_\_H}$   
 $[X + Y]_{\text{补}} = \text{\_\_\_\_\_\_H}$ ,  $[X - Y]_{\text{补}} = \text{\_\_\_\_\_\_H}$
4. 已知  $X = -65$ , 用 8 位机器数表示, 则  $[X]_{\text{原}} = \text{\_\_\_\_\_\_}$ ,  $[X]_{\text{反}} = \text{\_\_\_\_\_\_}$ ,  $[X]_{\text{补}} = \text{\_\_\_\_\_\_}$ 。
5. 已知  $X = 68$ ,  $Y = 12$ , 若用 8 位机器数表示, 则  $[X + Y]_{\text{补}} = \text{\_\_\_\_\_\_}$ ,  $[X - Y]_{\text{补}} = \text{\_\_\_\_\_\_}$ , 此时,  $OF = \text{\_\_\_\_\_\_}$ 。
6. 已知  $[X]_{\text{原}} = 01001001\text{B}$ ,  $[Y]_{\text{原}} = 10101010\text{B}$ , 求  $[X + Y]_{\text{补}} = \text{\_\_\_\_\_\_}$ ,  $[X - Y]_{\text{补}} = \text{\_\_\_\_\_\_}$ , 并判断是否溢出。
7. 已知  $X = -32$ ,  $Y = 66$ , 用 8 位机器数表示, 则  $[X]_{\text{补}} = \text{\_\_\_\_\_\_}$ ,  $[Y]_{\text{补}} = \text{\_\_\_\_\_\_}$ ,  $[X - Y]_{\text{补}} = \text{\_\_\_\_\_\_}$ 。
8.  $X = -32$ ,  $Y = 13$ , 则  $[X + Y]_{\text{补}} = \text{\_\_\_\_\_\_}$ ,  $[X - Y]_{\text{补}} = \text{\_\_\_\_\_\_}$ 。

#### 【解】

1. 101101B
2. 2EH
3. 0FFFF; 0FF81; 0FF80; 007E
4. 11000001B; 10111110B; 10111111B
5. 01010000B; 00111000B; 0
6. 00011111B (未溢出); 01110011B (未溢出)
7. 11100000B; 01000010B; 10011110B
8. 11101101B; 11010011B

### 二、选择题

1. 在计算机内部, 一切信息的存取、处理和传送都是以\_\_\_\_\_形式进行的。  
 A. 十进制      B. 八进制      C. 十六进制      D. 二进制
2. 在下面几个不同进制的数中, 最大的数是\_\_\_\_\_。  
 A. 1100010B      B. 225Q      C. 500      D. 1FEH
3. 在下面几个不同进制的数中, 最小的数是\_\_\_\_\_。  
 A. 1001001B      B. 75      C. 37Q      D. 0A7H
4. 十进制数 38 的 8 位机器数补码是\_\_\_\_\_。  
 A. 0011001      B. 10100110      C. 10011001      D. 00100110

5. 十进制数-38 的 8 位机器数补码是\_\_\_\_\_。
- A. 01011011      B. 11011010      C. 11011011      D. 01011010
6. 有一个 8 位机器数的补码是 11111101, 其相应的十进制真值是\_\_\_\_\_。
- A. -3      B. -2      C. 509      D. 253
7. 十进制数-75 用二进制数 10110101 表示, 其表示方式是\_\_\_\_\_。
- A. 原码      B. 补码      C. 反码      D. ASCII 码
8. 已知 $[X]_{\text{原}} = 10011010\text{B}$ ,  $[Y]_{\text{原}} = 11101011\text{B}$ , 则 $[X - Y]_{\text{补}} =$ \_\_\_\_\_。
- A. 溢出      B. 01111011B      C. 10000101B      D. 01010001B
9. 构成微机的主要部件除 CPU、系统总线、I/O 接口外, 还有\_\_\_\_\_。
- A. CRT      B. 键盘  
C. 磁盘      D. 内存 (ROM 和 RAM)
10. 下列数中为最小值的是\_\_\_\_\_。
- A.  $(28)_{10}$       B.  $(01100011)_2$       C.  $(10011000)_{\text{BCD}}$       D.  $(5A)_{16}$
11. 下列数中为最大值的是\_\_\_\_\_。
- A. 5AH      B. 01100011B      C. 28      D.  $(10011000)_{\text{BCD}}$
12. 目前, 在计算机中采用二进制数, 是因为\_\_\_\_\_。
- A. 容易实现      B. 算术四则运算规则简单  
C. 书写方便      D. 可进行二值逻辑运算
13. 计算机中常用的 BCD 码是\_\_\_\_\_。
- A. 二进制数      B. 十六进制数  
C. 二进制编码的十进制数      D. 不带符号数的二进制形式
14. 10001010 是\_\_\_\_\_。
- A. 带符号数      B. 原码、反码、补码表示的带符号数  
C. 不带符号数      D. BCD 码  
E. 无法确定

【解】

1. D    2. D    3. C    4. D    5. B    6. A    7. B    8. D    9. D    10. A    11. B    12. A、B、D    13. C    14. E

三、计算题

- 将下列十进制数转换成十六进制数、八进制数和二进制数: 128、241、511、372、1024 和 3000。
- 将下列无符号二进制数分别转换成十进制数、八进制数和十六进制数: 1011001010B、11110100B、01101001B 和 100100100B。
- 求 11010010B 和 01001110B 两数分别做“与”、“或”、“异或”操作的运算结果。
- 将下列十六进制数转换成十进制数和二进制数: 2ECH、325H、FFH、1ABH 和 FFFFH。
- 将下列十进制数转换成二-十进制数: 46、121、731 和 2 345。

6. 将下列补码转换成十进制数: 10010110B、01101100B、00101010B、11101110B、10000001B 和 11000000B。

7. 完成下列 BCD 码的运算: 01100001B - 01010110B、10011000B - 01111001B、00100110B + 01101000B、01000010B + 01010010B。

8. 求下列各数以 100H 为模的补码: -04H、-19H、-0FH、-2AH 和 -4BH。

9. 已知  $[X]_{\text{补}} = 11000000\text{B}$ ,  $[Y]_{\text{补}} = 01001000\text{B}$ ,  $[Z]_{\text{补}} = 00110010\text{B}$ 。求  $[-X]_{\text{补}}$ 、 $[-Y]_{\text{补}}$ 、 $[-Z]_{\text{补}}$ ; 并计算  $[X - Y]_{\text{补}}$  和  $[X - Z]_{\text{补}}$ , 若有溢出请给以说明。

【解】

1. 十进制数	十六进制数	八进制数	二进制数
128	80H	200Q	10000000B
241	F1H	361Q	11110001B
511	1FFH	777Q	11111111B
372	174H	564Q	101110100B
1 024	400H	2000Q	1000000000B
3 000	BB8H	5670Q	101110111000B

2. 二进制数	十进制数	八进制数	十六进制数
1011001010B	714	1312Q	2CAH
11110100B	244	364Q	F4H
01101001B	105	151Q	69H
100100100B	292	444Q	124H

3. “与”、“或”、“异或”后的结果分别为: 01000010B、11011110B 和 10011100B。

4. 十六进制数	十进制数	二进制数
2ECH	748	1011101100B
325H	805	1100100101B
FFH	255	11111111B
1ABH	427	110101011B
FFFFH	65 535	1111111111111111B

5. 十进制数	8421BCD 码
46	01000110B
121	000100100001B
731	011100110001B
2 345	0010001101000101B

6. 依次为: -106、+108、+42、-18、-127 和 -64。

7. 分别为: 00000101B、00011001B、10010100B 和 10010100B。

8. 依次为 FCH、E7H、F1H、D6H 和 B5H。

9.  $[-X]_{\text{补}} = 01000000\text{B}$   $[-Y]_{\text{补}} = 10111000\text{B}$   $[-Z]_{\text{补}} = 11001110\text{B}$

$[X - Y]_{\text{补}} = [X]_{\text{补}} + [-Y]_{\text{补}} = 01111000\text{B}$  (有溢出)

$[X - Z]_{\text{补}} = [X]_{\text{补}} + [-Z]_{\text{补}} = 10001110\text{B}$  (无溢出)

#### 四、问答题

1. 什么是字节？什么是计算机的字长？
2. 计算机是能够自动完成算术运算和逻辑运算等的电子装置，那么为什么在它的运算器中只有加法器？
3. 所谓 4 位、8 位、16 位、32 位微处理器是按什么划分的？
4. 计算机的硬件和软件各由哪几部分组成？
5. 微型计算机系统的硬件由哪几部分组成？简要叙述其功能。

#### 【解】

1. 字节指的是 8 位二进制信息。字长是计算机字所含的二进制位数。计算机字是作为一个整体被一次传送或运算最多的二进制位数。
2. 因为减法是通过加补码实现的；乘法是采用部分积右移加被乘数或“0”实现的；除法是采用部分余数左移加除数补码或“0”实现的。
3. 按处理器的字长来划分的。
4. 计算机的硬件由运算器、控制器、存储器、输入设备和输出设备组成。软件由系统软件和应用软件组成。
5. 微型计算机系统的硬件由输入设备、输出设备、运算器、控制器、存储器等 5 部分组成。输入设备用来输入原始数据和程序；输出设备用来输出处理结果；运算器完成算术或逻辑运算；控制器用来实现程序的自动执行；存储器用来存放数据和程序。

## 第 2 章

## Intel 32 位微处理器

### 2.1 重点内容提要

微型计算机中的运算器和控制器合起来称为微处理器，因微处理器通常集成在一块大规模集成电路芯片上，所以人们又把微处理器称作 CPU。Intel 公司生产的 80386、80486、Pentium 都是 32 位的微处理器（Pentium 4 已是 64 位）。32 位微处理器是指在微处理器内部以 32 位二进制数为单位进行数据处理。

#### （一）微处理器的基本结构

80486 微处理器内部由 8 大部件组成：总线接口部件、高速缓存部件、代码预取部件、指令译码部件、浮点数部件、执行部件、段部件、分页部件。图 2.1 为 80486 微处理器的内部基本结构图，它反映了微处理器内部的基本逻辑部件、部件之间的基本联系、各部件的主要功能、数据在微处理器中的主要流动方向。

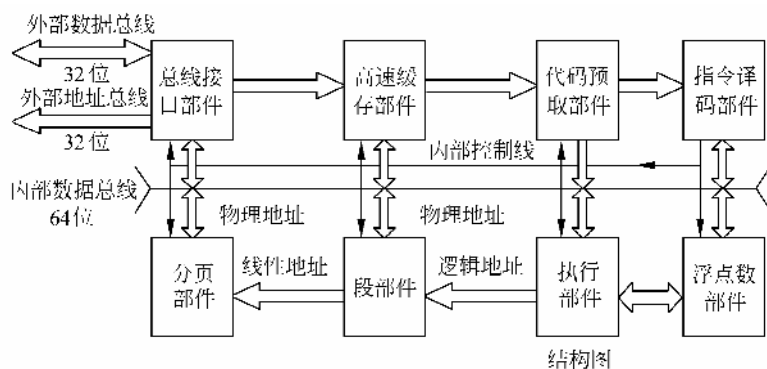


图 2.1 80486 内部基本结构图

微处理器外部引脚分为数据总线引脚、地址总线引脚及控制总线引脚。32 位微处理器都采用栅格阵列插针或封装，在微机主板上相应的插座安装。

80486 CPU 有 3 种工作方式：实地址方式、保护方式和虚拟 8086 方式。Pentium 微处理器除了上述 3 种工作方式外，增加了一种系统管理方式（SMM）。

#### （二）寄存器

寄存器是微处理器内部用来放置数据或地址的存储单元。在微处理器的各个部件

中，都有一些寄存器。有些寄存器是编程不可见的，有些是编程可见的。对于编程可见的寄存器，根据功能可分为 9 组，分别是通用寄存器、指令指针寄存器、标志寄存器、段寄存器、系统地址寄存器、调试寄存器、测试寄存器、控制寄存器及浮点寄存器。

在通用寄存器中，32 位寄存器有 EAX、EBX、ECX、EDX、EBP、ESP、ESI 和 EDI，其低 16 位分别是 AX、BX、CX、DX、BP、SP、SI 和 DI。其中 AX、BX、CX、DX 的每个字节均另有一个名字，高字节分别称为 AH、BH、CH、DH；低字节分别称为 AL、BL、CL、DL。

指令指针寄存器 EIP（低 16 位称为 IP），其内容为下一条要取入 CPU 的指令在内存中的偏移地址。

标志寄存器 EFLAGS（低 16 位称为 FLAGS），共有 3 类标志：状态标志、控制标志和系统标志。80486 中的标志包括进位标志 CF、奇偶标志 PF、辅助进位标志 AF、零标志 ZF、符号标志 SF、溢出标志 OF、方向标志 DF、中断允许标志 IF、陷阱标志 TF、嵌套任务标志 NT、输入输出特权级标志 IOPL、恢复标志 RF、虚拟 8086 方式标志 VM 和对准检查标志 AC。

段寄存器有 CS、DS、ES、SS、FS 和 GS。它们分为两部分，一部分是编程可见的选择寄存器，为 6 个 16 位寄存器，对应在另一部分有 6 个 64 位的编程不可见的描述符寄存器。在实地址方式或虚拟 8086 方式下，描述符寄存器不起作用，选择寄存器退化成 16 位 CPU 的段寄存器功能，存放内存段的段基址（段首地址的高 16 位）。其中 CS 对应于代码段，DS 对应于数据段，ES 对应于附加数据段，SS 对应于堆栈段，FS 和 GS 没有定义。在保护方式下，选择寄存器的低二位是特权标志。其中  $D_2$  位是描述符表类型标志，高 13 位是选择码，指出本段的段描述符在由  $D_2$  位指出的描述符表中的逻辑排序。

其他寄存器都是与系统有关的寄存器。控制寄存器和系统地址寄存器是实现保护模式功能的基础，控制着新的存储器管理方式、多任务切换等功能的实现；调试和测试寄存器用于调试微码执行、数据访问的断点自陷和测试芯片逻辑、高速缓冲存储器等；浮点寄存器主要用于浮点运算。

### （三）逻辑地址、线性地址和物理地址

一般来说，用户使用的是逻辑地址，由内存管理软件和相关硬件将逻辑地址变为物理地址。

在 16 位模式下，逻辑地址由 16 位的段基址和 16 位的偏移地址组成，物理地址 20 位。其关系为：

$$\text{物理地址} = \text{段基址} \times 10H + \text{偏移地址}$$

在 32 位模式下，地址转换比较复杂，逻辑地址是一个 16 位的段选择符和一个 32 位的偏移地址。段部件根据段选择符获得相应的段描述符，取出 32 位的段基址再加上 32 位偏移地址得到 32 位的线性地址。然后再由段部件传送给分页部件。由其转换为 32 位的物理地址。若分页部件被禁止，那么计算出的线性地址就是物理地址。

### （四）指令流水线操作

32 位微处理器的一个重要的特点就是采用了指令流水线技术。这一技术大大加快了

指令执行速度, 加大了信息流量。这是一种同时进行若干操作的并行处理方式。在流水线结构中, 每条指令分成若干步骤来执行, 每个子过程各在一个专门的硬件站点上执行, 这样一条指令的全部操作需顺序地经过流水线中多个站点的处理才能完成。但前后连续的几个操作可以依次进入流水线中, 在各个站点间重叠执行, 以此来实现并行操作。

在 80486 微处理器中设有 8KB 的内部高速缓存器、32 字节的预取指令队列、指令译码器、控制器及执行部件, 这就从硬件上支持了指令流水线操作。80486 使用 5 个步骤的流水线技术, 即指令预取 PF、译码  $D_1$ 、译码  $D_2$ 、执行 EX 和结果写回 WB。其执行过程如图 2.2 所示。

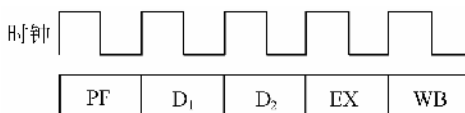


图 2.2 80486 指令执行过程

## 2.2 常考题型范例精解

**例 2.1** 80486 微处理器内部由哪些部件组成? 各部件主要完成什么功能?

**【解】** 80486 微处理器内部由总线接口部件、高速缓存部件、代码预取部件、指令译码部件、浮点数部件、执行部件、段部件和分页部件组成。

总线接口部件负责内部总线和外部总线的联系; 高速缓存部件用于程序或数据的缓存, 用以减少对内存的访问次数, 减少程序运行时间; 代码预取部件对代码做取入、排队分析、分解等译码的前期准备工作; 指令译码部件完成指令译码工作, 把指令的含义转换为相应的内部控制总线信号, 指挥各部分协同工作; 浮点数部件完成浮点数运算、双精度运算等执行部件不擅长的数学运算任务; 执行部件完成一般算术运算、逻辑运算及数据传送等任务; 段部件把指令指定的逻辑地址变换为线性地址; 分页部件把段部件输出的线性地址转换成物理地址。

**例 2.2** 设  $[X]_{\text{补}} = 10001110\text{B}$ 、 $[Y]_{\text{补}} = 10110011\text{B}$ , 试指出两数相加后, ZF、CF、AF、PF、SF 和 OF 的状态。

$$\begin{array}{r} \text{【解】} \quad 10001110\text{B} \\ + 10110011\text{B} \\ \hline 01000001\text{B} \end{array}$$

因为结果非零, 故  $ZF = 0$ ; 最高位  $D_7$  位上有进位, 故  $CF = 1$ ;  $D_3$  位向  $D_4$  位有进位, 故  $AF = 1$ ; 结果中“1”的个数为偶数, 故  $PF = 1$ ; 结果最高位  $D_7$  是“0”, 故  $SF = 0$ ; OF 的状态可用双高位法判断, 这里  $C_S = 1$ 、 $C_P = 0$ 、 $C_S \oplus C_P = 1$ , 故  $OF = 1$ 。

**【评注】** ①在 6 位状态标志中, 溢出标志 OF 最难判断, 它是由一个运算式判别的。这个运算式可记为  $OF = C_S \oplus C_P$ , 其中  $C_S$  是符号位产生的进位,  $C_P$  为最高数位向符号位产生的进位。这种判别方法叫双高位法。②CF 和 AF 都是有关进位的标志。CF 置位时, 表明 8 位或 16 位或 32 位数的最高位产生了进位或借位; 而 AF 置位时, 表明最低有效的 4 位向高位产生了进位或借位, 即数据  $D_3$  位向  $D_4$  位的进位状态。因此在使用这两个标志位时, 请注意使用方法。

**例 2.3** 在实方式和虚拟 8086 方式下, Intel 80486 中段寄存器中存放的值是什么?

而在保护方式下，它的功能又是什么？

**【解】** 在实方式和虚拟 8086 方式下，段寄存器中存放的是段基址，即段首地址的高 16 位。而在保护模式下，段寄存器分为两部分，一部分是编程可见的选择寄存器，为 16 位寄存器；对应的另一部分为 64 位的编程不可见的描述符寄存器。选择寄存器的低二位是特权标志。其中  $D_2$  位是描述符表类型标志，高 13 位是选择码，指出本段的段描述符在由  $D_2$  位指出的描述符表中的逻辑排序。描述符寄存器用来备份当一段被首次访问成功后该段的段描述符，以备下次访问时直接取到描述符。

**例 2.4** 在实地址方式下，假定内存数据段中有两个数据字 1234H 和 5678H；若已知当前  $DS = 5AA0H$ ，它们的偏移地址分别为 245AH 和 3245H，试用图说明它们在存储器中的物理地址。

**【解】** 因为在实地址方式下，DS 的内容为数据段的段基址，根据 16 位模式下的物理地址的定义式：

$$\text{物理地址} = \text{段基址} \times 10H + \text{偏移地址}$$

数据字 1234H 占两个连续的单元，为首的物理地址计算如下：

$$(DS) \times 10H + 245AH = 5AA00H + 245AH = 5CE5AH$$

数据字 5678H 也占两个连续的单元，为首的物理地址计算如下：

$$(DS) \times 10H + 3245H = 5AA00H + 3245H = 5DC45H$$

两个数据字在内存中存储如图 2.3 所示。

**【评注】** ①物理地址是数据或代码在存储器中的实际地址。每个存储单元都只有一个物理地址。物理地址是唯一的。但对应同一个物理地址，可以有不同的逻辑地址。②内存是以字节为单位编址的，即每个单元内只存一个字节的的信息。所以 16 位的数据字需占连续两个单元，低字节存于低地址单元中，高字节存于高地址单元中。故 1234H 中的 34H 存于物理地址为 5CE5AH 对应的单元中，而 12H 存于物理地址为 5CE5BH 对应的单元中。5678H 同理。

物理地址	内存
5CE5AH	00110100
5CE5BH	00010010
.....	.....
5DC45H	01111000
5DC46H	01010110

图 2.3 数据字存储图

**例 2.5** 若某数据区的开始地址为 A7F0H：2B40H，试问该数据区的首字单元和 32 个字的末字单元的物理地址分别为多少？

**【解】** 首地址为：

$$0A7F0H \times 10H + 2B40H = 0AAA40H$$

末地址为：

$$0AAA40H + (20H - 1) \times 2 = 0AAA7EH$$

**【评注】** 数据区最后一个字的地址为：首地址 + (字数-1) × 2。

## 2.3 习题及全解

### 一、填空题

1. 80486 微处理器中的执行部件完成\_\_\_\_\_等任务。



2. 80486 微处理器中的段部件把指令指定的\_\_\_\_\_地址变为\_\_\_\_\_地址。
3. 80486 微处理器的分页部件把\_\_\_\_\_地址变为\_\_\_\_\_地址。
4. 32 位微处理器中, 32 位通用寄存器的名称有\_\_\_\_\_。
5. 在实地址方式下, 逻辑地址中的段基址是由\_\_\_\_\_存储的。
6. 当微处理器进行数据输出时, 信号  $\overline{W}/\overline{R}$  为\_\_\_\_\_; 当微处理器进行数据输入时, 信号  $\overline{W}/\overline{R}$  为\_\_\_\_\_。
7. 80486 的数据总线是\_\_\_\_\_位, 地址总线是\_\_\_\_\_位。
8. 80486 在保护方式下, 虚拟空间为\_\_\_\_\_, 而实际地址空间为\_\_\_\_\_。
9. 80486 微处理器的工作方式有\_\_\_\_\_, \_\_\_\_\_和\_\_\_\_\_。
10. 80486 内部结构共有 8 个功能部件: \_\_\_\_\_、\_\_\_\_\_, \_\_\_\_\_、\_\_\_\_\_, \_\_\_\_\_、\_\_\_\_\_, \_\_\_\_\_和\_\_\_\_\_。
11. 80486 提供的能接收外部中断请求信号的引脚是\_\_\_\_\_和\_\_\_\_\_, 这两种信号的不同之处在于\_\_\_\_\_。
12. 在实地址方式下段寄存器的内容是\_\_\_\_\_, 在保护方式下段寄存器的内容是\_\_\_\_\_。

#### 【解】

1. 算术运算、逻辑运算、数据传送
2. 逻辑; 线性
3. 线性; 物理
4. EAX、EBX、ECX、EDX、EBP、ESP、ESI 和 EDI
5. 段寄存器
6. 高; 低
7. 32; 32
8. 64TB; 4GB
9. 实地址方式; 保护方式; 虚拟 8086 方式
10. 总线接口部件; 高速缓存部件; 代码预取部件; 指令译码部件; 浮点数部件; 执行部件; 段部件; 分页部件
11. INTR; NMI; 中断是否可屏蔽
12. 段基址; 选择码、特权标志和描述符表类型标志

#### 二、选择题

1. 指令指针寄存器(EIP)中存放的是\_\_\_\_\_。  
 A. 当前指令  
 B. 下一条要执行的指令  
 C. 操作数地址  
 D. 下一条要执行指令的地址
2. 指令队列的作用是\_\_\_\_\_。  
 A. 暂存操作数地址  
 B. 暂存操作数  
 C. 暂存指令地址  
 D. 暂存预取指令
3. 当  $\overline{M}/\overline{IO}=0$ ,  $\overline{W}/\overline{R}=0$  时, 微处理器完成的操作是\_\_\_\_\_。

- A. 存储器读    B. I/O 读    C. 存储器写    D. I/O 写
4. 80486 中有\_\_\_\_\_个 32 位通用寄存器。  
A. 4    B. 10    C. 16    D. 8
5. 主存和微处理器之间增加高速缓存的目的是\_\_\_\_\_。  
A. 解决微处理器和主存之间的速度匹配问题  
B. 扩大主存容量  
C. 既扩大主存容量，又提高存取速度
6. 溢出是两个带符号数\_\_\_\_\_, 结果超出规定的数值范围。  
A. 两同号数相减    B. 两同号数相加  
C. 两异号数相加    D. 两数异或
7. 80486 微处理器的工作方式有\_\_\_\_\_。  
A. 保护方式    B. 系统管理方式  
C. 实地址方式    D. 虚拟 8086 方式
8. 与堆栈有关的寄存器有\_\_\_\_\_。  
A. ESP    B. EBP    C. EBX    D. ESS

【解】

1. D    2. D    3. B    4. D    5. A    6. B    7. A、C、D    8. A、B、D

### 三、判断题

1. 80486 标志寄存器共有 32 位，每一位都有含义。  
2. 80486 的数据总线和地址总线都是 32 位的。

【解】

1. 错误    2. 正确

### 四、问答题

1. 80486 微处理器内部由哪几部分组成？  
2. 80486 有哪几种工作方式？  
3. 8086 存储器操作数的段基址、偏移地址、段的首地址、物理地址各是多少位？

80486 的段基址、偏移地址、段的首地址、物理地址又各是多少位？

4. 标志位寄存器中用户直接可以用指令改变其值的有哪几个？  
5. 计算机运算结果的溢出指的是什么？如何判断？  
6. Intel 80486 微处理器中是如何进行指令流水线操作的？

【解】

1. 由总线接口部件、高速缓存部件、代码指令预取部件、指令译码部件、浮点数部件、执行部件、段部件和分页部件 8 大部分组成。  
2. 有 3 种工作方式：实地址方式、保护方式和虚拟 8086 方式。  
3. 8086 存储器操作数的段基址是 16 位、偏移地址是 16 位、段的首地址是 20 位、物理地址是 20 位；80486 的段基址是 32 位、偏移地址是 32 位、段的首地址是 32 位、物理地址是 20 位；80486 的段基址是 32 位、偏移地址是 32 位、段的首地址是 32 位、物理地址是 20 位。

物理地址是 32 位。

4. 有 CF、DF 和 IF。

5. 溢出是指运算的结果超过了目的单元所能表示的数的范围。当 OF=“1”时，说明结果发生了溢出。

6. Intel 80486 微处理器中设有 8KB 内部高速缓存器、32 字节的预取指令队列、指令译码器、控制器及执行部件，这就从硬件上支持了指令流水线操作。80486 使用 5 个步骤的流水线技术，即指令预取 PF、译码 D<sub>1</sub>、译码 D<sub>2</sub>、执行 EX 和结果写回 WB。

## 第 3 章

## 80x86 寻址方式和指令系统

### 3.1 重点内容提要

#### (一) 80x86 数据寻址方式

通常，一条指令由操作码和操作数组成。数据寻址方式即指寻找指令中操作数所在地址的方法。80x86 操作数可以放在寄存器、存储器中，也可以以立即数的形式放在指令代码中。因此，从大体上来分，80x86 的寻址方式分为 3 大类，即立即寻址、寄存器寻址和存储器寻址。

##### 1. 立即寻址

操作数包含在指令中，作为指令的一部分，存放在代码段中。当取指令时，操作数也随之取出。如：

`MOV AL, 80H`

`MOV EAX, 0`

注意：立即寻址主要用来给通用寄存器或存储器赋值，它只能用于源操作数，不能用于目的操作数。

##### 2. 寄存器寻址

在这种方式中，操作数放在 32 位、16 位或 8 位的通用寄存器中。执行时不需使用总线周期，因而运算速度较快。如：

`MOV AX, BX`

##### 3. 存储器寻址

操作数在存储器中，其地址可以在指令中直接给出，也可以放在 16 位或 32 位规定的寄存器中间接给出。寻址所需要的偏移地址即为有效地址 EA，它代表被寻址的操作数在存储器中距离该段起点的字节数。有效地址 EA 的计算随所采用的寻址方式不同而异。

(1) 16 位指令模式下的存储器寻址方式：80x86 实方式为 16 位指令模式。寻址的寄存器为 BX、BP、SI、DI。如果有效地址在 BX、SI 或 DI 中，则以 DS 寄存器内容为段基址；如果有效地址在 BP 中，则以 SS 段寄存器之内容为段基址。如果使用段超越前缀（CS: ES: DS: SS:），操作数则可以放在冒号前指定的段。16 位指令模式寻址结构由 4 部分组成。

$$\text{段基址} \times 10H + \text{基址} + \text{变址} + \text{偏移量}$$

基址寄存器 **BX** 和 **BP** 存放基址, 变址寄存器 **SI** 和 **DI** 存放变址, 偏移量为 8 位或 16 位用补码形式表示的带符号数。操作数的有效地址为 16 位, 段内寻址范围为 64KB, 物理地址为 20 位, 使用存储器的第一个 1MB 空间。

① 直接寻址 操作数的有效地址 **EA** 就在指令中, 可以是数值地址, 也可以是符号地址。符号地址必须是赋过值的。如:

**MOV AX, [2000H]**

**MOV BX, BLOCK**

注意: **EA** 是在指令中, 而操作数本身在数据段的存储单元中。

② 寄存器间接寻址 操作数的有效地址在 **BX**、**BP** 或 **SI**、**DI** 中, 可表示为:

$$EA = \begin{bmatrix} (BX) \\ (BP) \\ (SI) \\ (DI) \end{bmatrix}$$

例如: **MOV AX, [BX]**

③ 寄存器相对寻址 操作数在存储单元中, 其有效地址是一个 8 位或 16 位的偏移量与一个基址或变址寄存器的内容之和, 偏移量和寄存器均由指令给出, 可表示为:

$$EA = \begin{bmatrix} (BX) \\ (BP) \\ (SI) \\ (DI) \end{bmatrix} + \begin{bmatrix} 8\text{位偏移量} \\ 16\text{位偏移量} \end{bmatrix}$$

例如: **MOV AL, [BX + 5]**

④ 基址变址寻址 操作数在存储单元中, 其有效地址是一个基址寄存器与一个变址寄存器的内容之和, 可表示为:

$$EA = \begin{bmatrix} (BX) \\ (BP) \end{bmatrix} + \begin{bmatrix} (SI) \\ (DI) \end{bmatrix}$$

例如: **MOV AX, [BP + DI]**

⑤ 相对基址变址寻址 操作数在存储单元中, 其有效地址是一个 8 位或 16 位偏移量、一个基址寄存器与一个变址寄存器 3 部分内容之和, 可表示为:

$$EA = \begin{bmatrix} (BX) \\ (BP) \end{bmatrix} + \begin{bmatrix} (SI) \\ (DI) \end{bmatrix} + \begin{bmatrix} 8\text{位偏移量} \\ 16\text{位偏移量} \end{bmatrix}$$

例如: **MOV AX, [BX+SI+2000H]**

(2) 32 位指令模式下的存储器寻址方式: 在保护方式下, 32 位指令模式寻址结构由 5 部分组成。

段址 + 基址 + 变址 × 比例因子 + 偏移量

其中, 基址寄存器或变址寄存器可以是除 **ESP** 以外的任何 32 位通用寄存器。当基址寄存器为 **EBP** 时, 默认段寄存器 **SS** 存放段选择符, 否则, 默认 **DS** 存放段选择符。也可使用段超越前缀来指定。比例因子为 1、2、4、8。偏移量为 8 位或 32 位。

① 直接寻址 指令中直接包含 32 位的有效地址。**EA** 为 32 位有效地址。

② 间接寻址 EA 为基址寄存器的内容。

③ 相对基址寻址 EA 为基址寄存器和一个 8 位或 32 位偏移量之和。

④ 相对比例变址寻址 EA 为变址寄存器的内容乘以比例因子，再加偏移量。

⑤ 相对比例基址变址寻址 EA 为变址寄存器的内容乘以比例因子，加上基址寄存器的内容，再加上偏移量。

## （二）80x86 指令系统

80x86 指令系统，按指令的功能可分为数据传送指令、算术运算指令、逻辑运算指令、控制转移类指令、串操作指令、处理器控制指令和保护方式指令。

### 1. 传送指令

这类指令完成数据、地址等的传送操作，除了标志寄存器传送指令外，其他传送指令都不影响标志位。

在使用传送指令 MOV、堆栈指令 PUSH 和 POP 以及数据交换指令 XCHG 时应注意：

①不允许将立即数直接送段寄存器；②不允许目的操作数为立即数和 CS 段寄存器；③不允许两个存储单元之间直接传送数据；④不允许两个段寄存器之间直接传送数据；⑤操作数数据类型要匹配；⑥对堆栈的操作不允许字节操作；⑦不能改变 CS 和指令指针寄存器（E）IP 的值。

地址传送指令 LEA、LDS 和 LES 的功能是将存储单元的地址送入特定的寄存器中。

标志传送指令 LAHF、SAHF、PUSHF、POPF、PUSHFD、POPFD 主要用来获取标志寄存器的值或改变标志寄存器的值，通过 AL 寄存器或堆栈来实现。

查表指令 XLAT 实现直接查表功能。规定 BX 寄存器存放表的首地址的偏移地址，AL 寄存器存放表内偏移量，查得结果存放在 AL 中。

符号扩展指令是将累加器中带符号数进行符号扩展，正数高半部分扩展为全“0”，负数高半部分扩展为全“1”。

### 2. 算术运算指令

算术运算指令用来完成加、减、乘、除等算术运算，包括 4 种基本算术运算操作及十进制算术调整指令。指令执行结果均影响标志位。

ADD 和 ADC 为加法指令，SUB 和 SBB 为减法指令，ADC 和 SBB 要考虑 CF 值。INC 和 DEC 分别为加 1 和减 1 指令，不影响 CF 值。比较指令 CMP 完成两个数的相减操作，但不回送相减的结果，只是使结果影响标志位，常用于分支程序设计中。

乘法指令分为无符号数乘 MUL 和带符号数乘 IMUL，只影响 CF 和 OF 标志。除法指令也分无符号数除 DIV 和带符号数除 IDIV，被除数必须是除数的双倍字长，一般要进行高位扩展。当进行无符号数除法时，被除数高位按 0 扩展为双倍除数字长。当进行有符号数除法时，被除数以补码表示，可使用扩展指令 CBW、CWD、CWDE 和 CDQ 进行高位扩展。

BCD 算术运算指令的功能是将二进制运算结果调整为 BCD 码表示形式。BCD 码分为组合型（压缩格式）的 BCD 码和非组合型的 BCD 码（ASCII 码）。组合型 BCD 码的加、减调整指令分别为 DAA 和 DAS；非组合型 BCD 码的加、减、乘、除调整指令分别

为 AAA、AAS、AAM 和 AAD。DAA 和 AAA 一般跟在 ADD 或 ADC 指令之后, DAS 和 AAS 一般跟在 SUB 或 SBB 之后, AAM 跟在 MUL 之后, 但 AAD 则用于 DIV 操作之前。注意, BCD 码总是被作为无符号数看待的。

使用算术运算指令时, 如果没有特别规定, 参与运算的两个操作数类型要匹配, 且不能同时为存储器操作数。操作数不允许为段寄存器。若操作数类型在指令中不明确, 要用 PTR 伪指令予以说明。

### 3. 逻辑运算指令

逻辑运算指令是按位操作, 除 NOT 指令不影响标志位外, 其余指令将使 CF=OF=0, 影响 SF、ZF 和 PF。TEST 和 AND 指令执行同样的操作, 但 TEST 不送回操作结果, 而仅仅影响标志位。TEST 指令用来测试某位的状态。AND 指令用来使某些位屏蔽, 即清 0。OR 指令常用来将一些位置 1。XOR 指令常用在使某个寄存器清 0 或使某些位变反。

移位指令包括有算术移位、逻辑移位和循环移位指令。算术移  $N$  位, 可相当于把二进制数乘以或除以  $2^N$ 。逻辑移位操作用于截取字节或字或双字的若干位。循环移位指令在移位时移出的目的操作数并不丢失, 而循环送回目的操作数的另一端。

### 4. 控制转移类指令

这类指令用来控制程序的执行流程, 即改变程序的走向。分为转移指令和循环控制指令。

转移指令可分为无条件转移指令和条件转移指令两类。

无条件转移指令 JMP 是指程序无条件地转移至其指定的地址去执行从该地址开始的指令。转移可分为段内转移和段间转移, 段内转移只改变 IP, 而段间转移改变 IP 和 CS。段内转移可以是段内直接转移(转向的有效地址是当前的 IP 寄存器的内容和一个偏移量之和), 也可以是段内间接寻址(转向的有效地址在一个寄存器或存储器单元中, 即用寄存器或存储器的内容取代 IP 的内容)。段间转移可以是段间直接转移(直接提供 CS 和偏移地址 IP), 也可以是段间间接转移(用存储器相邻两个字的内容取代 IP 和 CS 的内容)。

条件转移指令比无条件转移指令多了一个条件判断的功能, 若满足指令规定的条件, 则程序转移, 否则程序顺序执行。条件转移指令的条件主要是根据标志位来判别的, 对于 CF、ZF、SF、OF 和 PF 都有相应的判断指令。对于无符号数和带符号数大小的判别也有两组指令, 是根据标志位的组合状态来判别的。还有测试 CX 条件转移指令。条件转移指令只允许段内转移, 不允许段间转移。

循环控制指令 LOOP 用来管理程序循环的次数, 它与条件转移不同的是, 循环指令要对 (E) CX 寄存器的内容进行测试, 判断其是否为 0 作为转移条件。每循环执行一次, (E) CX 的内容自动减 1, 直到 (E) CX 为 0 时循环结束。

### 5. 串操作指令

串操作指令都用寄存器 (E) SI 对源操作数进行间接寻址, 并且默认是在 DS 段中; 用 (E) DI 对目的操作数进行间接寻址, 并且默认是在 ES 段中。(E) SI 和 (E) DI 这两个地址指针在每次串操作以后都自动进行修改, 这与方向标志 DF 有关。当 DF=1 时, (E) SI 和 (E) DI 做自动减量修改; 当 DF=0 时, (E) SI 和 (E) DI 做自动增量修改。

增、减量根据串元素的数据类型不同而不同，字节型对应是 1，字型对应是 2，双字型对应是 4。可以通过加重复前缀来实现串操作。

### 6. 输入/输出指令

使用 IN 或 OUT 指令时，要注意：一个操作数为 AL 或 AX 或 EAX，另一个操作数以 8 位端口地址方式直接给出，或者以 DX 寄存器间接给出。直接寻址范围为 256B，间接寻址范围为 64KB。每个 I/O 地址对应的端口的数据长度为 8 位，传送 8 位数据占用一个端口地址，传送 16 位数据占用两个端口地址，传送 32 位数据占用 4 个端口地址。

## 3.2 常考题型范例精解

**例 3.1** 设 DS = 2000H, SS = 3000H, BP = 0200H, SI = 4000H, BUF = 1000H, EAX = 00001000H, EBX = 00002000H, 假设按 16 位实模式操作，确定下列每条指令访问内存的物理地址，并且指出源操作数及目的操作数的寻址方式。

- |                        |                         |
|------------------------|-------------------------|
| (1) MOV AL, [1234H]    | (2) MOV EDX, [BX]       |
| (3) MOV CL, [BX+100H]  | (4) MOV[SI], EBX        |
| (5) MOV AH, BUF[BX+SI] | (6) MOV EAX, [BP+1234H] |

**【分析】** DS 给出数据段的段基址，因此，数据段的起始物理地址为 (DS) × 10H = 20000H。SS 给出堆栈段的段基址，因此，堆栈段的起始物理地址为 (SS) × 10H = 30000H 若是存储器寻址，根据寻址方式的不同，可求出相应的有效 EA，再加上段的起始物理地址即为将访问的内存单元的物理地址。

**【解】**

- (1) 源操作数在数据段中，为直接寻址方式。

$$EA = 1234H$$

$$\text{物理地址} = 20000H + 1234H = 21234H$$

目的操作数为寄存器寻址。

- (2) 源操作数在数据段中，为寄存器间接寻址。

$$EA = (BX) = 2000H$$

$$\text{物理地址} = 20000H + 2000H = 22000H$$

目的操作数为寄存器寻址。

- (3) 源操作数在数据段中，为寄存器相对寻址。

$$EA = (BX) + 100H = 2100H$$

$$\text{物理地址} = 20000H + 2100H = 22100H$$

目的操作数为寄存器寻址。

- (4) 源操作数为寄存器寻址。

目的操作数为寄存器间接寻址。

$$EA = (SI) = 4000H$$

$$\text{物理地址} = 20000H + 4000H = 24000H$$

- (5) 源操作数在数据段中，为相对基址变址寻址。



$$EA = (BX) + (SI) + BUF = 2000H + 4000H + 1000H = 7000H$$

$$\text{物理地址} = 20000H + 7000H = 27000H$$

目的操作数为寄存器寻址。

(6) 由于源操作数方括号中出现了 BP, 所以源操作数在堆栈段中, 为寄存器相对寻址。

$$EA = (BP) + 1234H = 0200H + 1234H = 1434H$$

$$\text{物理地址} = 30000H + 1434H = 31434H$$

目的操作数为寄存器寻址。

**【评注】** 注意操作数的位置, 是立即数还是在寄存器中, 或在存储器中。若为存储器寻址, 还要判断操作数是在数据段中还是在堆栈段中。

注意区别直接寻址方式和立即寻址方式, (1) 中源操作数是[1234H], 该指令是要把数据段中 1234H 单元的内容送 AL, 并非将 1234H 送 AL, 是一种直接寻址方式。

使用寄存器间接寻址时应注意和寄存器寻址方式的区别。(2) 中源操作数是[BX], BX 寄存器中的内容 2000H 是一个地址, 该指令是要把数据段中以 2000H 单元为首的一个双字数据送 EDI 寄存器中, 并非将 2000H 送 EDI。所以该指令源操作数为寄存器间接寻址。

**例 3.2** (AL) = 9AH, (BL) = 0BCH, 请分别进行下列指令运算, 并写出标志位的内容。

(1) ADD AL, BL; ZF=\_\_\_\_, PF=\_\_\_\_, CF=\_\_\_\_, SF=\_\_\_\_, OF=\_\_\_\_。

(2) SUB AL, BL; ZF=\_\_\_\_, PF=\_\_\_\_, CF=\_\_\_\_, SF=\_\_\_\_, OF=\_\_\_\_。

(3) AND AL, BL; ZF=\_\_\_\_, PF=\_\_\_\_, CF=\_\_\_\_, SF=\_\_\_\_, OF=\_\_\_\_。

(4) OR AL, BL; ZF=\_\_\_\_, PF=\_\_\_\_, CF=\_\_\_\_, SF=\_\_\_\_, OF=\_\_\_\_。

(5) XOR AL, BL; ZF=\_\_\_\_, PF=\_\_\_\_, CF=\_\_\_\_, SF=\_\_\_\_, OF=\_\_\_\_。

**【解】**

$$\begin{array}{r} \text{(1)} \quad \text{ADD AL, BL} \\ \quad 1\ 0\ 0\ 1\ 1\ 0\ 1\ 0\ B \quad (9AH) \\ + 1\ 0\ 1\ 1\ 1\ 1\ 0\ 0\ B \quad (0BCH) \\ \hline \quad 0\ 1\ 0\ 1\ 0\ 1\ 1\ 0 \quad (\text{有进位}) \end{array}$$

所以, ZF=0、PF=1、CF=1、SF=0、OF=1。

$$\begin{array}{r} \text{(2)} \quad \text{SUB AL, BL} \\ \quad 1\ 0\ 0\ 1\ 1\ 0\ 1\ 0\ B \quad (9AH) \\ - 1\ 0\ 1\ 1\ 1\ 1\ 0\ 0\ B \quad (0BCH) \\ \hline \quad 1\ 1\ 0\ 1\ 1\ 1\ 1\ 0 \quad (\text{有借位}) \end{array}$$

所以, ZF=0、PF=1、CF=1、SF=1、OF=0。

(3) 对于“与”、“或”和“异或”逻辑运算, 将使 CF=0、OF=0, 而 SF、ZF 和 PF 则根据运算结果设置。

$$\begin{array}{r} \text{AND AL, BL} \\ \quad 1\ 0\ 0\ 1\ 1\ 0\ 1\ 0\ B \quad (9AH) \\ \wedge 1\ 0\ 1\ 1\ 1\ 1\ 0\ 0\ B \quad (0BCH) \\ \hline \quad 1\ 0\ 0\ 1\ 1\ 0\ 0\ 0 \end{array}$$

所以, ZF = 0、PF = 0、CF = 0、SF = 1、OF = 0。

```
(4)          OR AL, BL
              1 0 0 1 1 0 1 0 B    (9AH)
            + 1 0 1 1 1 1 0 0 B    (0BCH)
            -----
              1 0 1 1 1 1 1 0
```

所以, ZF = 0、PF = 1、CF = 0、SF = 1、OF = 0。

```
(5)          XOR AL, BL
              1 0 0 1 1 0 1 0 B    (9AH)
            + 1 0 1 1 1 1 0 0 B    (0BCH)
            -----
              0 0 1 0 0 1 1 0
```

所以, ZF = 0、PF = 0、CF = 0、SF = 0、OF = 0。

**例 3.3** 判断下列指令是否正确, 若有错误, 请指出原因, 并改正。

- (1) MOV BL, OFFSET BLOCK
- (2) LEA DI, [BX+SI]
- (3) PUSH AL
- (4) POP CS
- (5) OUT CX, AL
- (6) MOV ES, DS
- (7) JMP BYTE PTR [SI]
- (8) MOV AX, [SI+DI]
- (9) MOV [BX+BP], AX
- (10) ADD AX, F000H

**【解】**

(1) 错误。OFFSET BLOCK 是指变量 BLOCK 的偏移地址, 不能送往 8 位的寄存器中, 应改为 MOV BX, OFFSET BLOCK。

(2) 正确。

(3) 错误。堆栈操作不能进行字节操作。应改为 PUSH AX。

(4) 错误。不能给 CS 赋值。可以改为别的段寄存器, 比如 POP DS。

(5) 错误。I/O 指令中只能用 DX 作为间接寻址的寄存器, 应该改为 OUT DX, AL。

(6) 错误。段寄存器之间不能直接传送数据。要完成该语句的功能, 应改为:

```
MOV AX, DS
MOV ES, AX
```

(7) 错误。转移地址必须是 16 位或 32 位。可改为 JMP NEAR/FAR PTR [SI]。

(8) 错误。SI 和 DI 两个变址不能同时出现在方括号里, 可以是一个基址加一个变址, 可改为 MOV AX, [BX+DI]。

(9) 错误。BX 和 BP 两个基址不能同时出现在方括号里。可以改为 MOV[BX+SI], AX。

(10) 错误。F000H 是一个以字母打头的数字, 为了和变量区别, 应书写成 0F000H, 所以应改为 ADD AX, 0F000H。

**例 3.4** 查表指令和串操作指令都采用隐含寻址, 试问它们的地址分别都有哪些约定?

**【解】** 查表指令 XLAT 隐含: 表首地址的偏移地址存于 BX 中, 表中偏移量在 AL 中。段基址在 DS 中, 有效地址为 BX 和 AL 之和。串操作指令隐含: 源串的段基址在 DS 中, 串元素偏移地址为 (E) SI; 目的串段基址在 ES 中, 串元素偏移地址在 (E) DI 中。每次串操作之后, (E) SI 和 (E) DI 都会自动修改。

**例 3.5** 根据下列题意各写一条指令。

- (1) 用逻辑指令将 EAX 清 0。
- (2) 将 BX 的高 8 位清 0。
- (3) 将 CX 的低 4 位和高 4 位取反。
- (4) 将 DX 的低 8 位置 1。
- (5) 对 AX 的内容算术右移两位。

**【分析】** 这些功能需要用逻辑运算指令来实现。AND 指令常用于将操作数中某些位清 0, 只需将要清 0 的位与 0, 其他不变的位与 1 即可; OR 指令常用于将操作数中某些位置 1, 只需将要置 1 的位或 1, 其他不变的位或 0 即可; XOR 指令常用于将操作数中某些位取反, 只需将要取反的位异或 1, 其他不变的位异或 0 即可; 也常用 XOR 指令将某个寄存器清 0。算术右移用 SAR 指令。

**【解】**

- (1) AND EAX, 0 或 XOR EAX, EAX
- (2) AND BX, 00FFH
- (3) XOR CX, 0F00FH
- (4) OR DX, 00FFH
- (5) SAR AX, 2

**例 3.6** 写出把首地址为 BUF 的字节缓冲区中第 5 个字节数送 AL 寄存器的指令, 要求使用以下几种寻址方式:

- (1) 寄存器间接寻址。
- (2) 寄存器相对寻址。
- (3) 基址变址寻址。

**【解】**

- (1) LEA BX, BUF+4  
MOV AL, [BX] ; 寄存器间接寻址
- (2) LEA BX, BUF  
MOV AL, [BX+4] ; 寄存器相对寻址
- (3) LEA BX, BUF  
MOV SI, 4  
MOV AL, [BX+SI] ; 基址变址寻址

**例 3.7** 试分别使用数据传送指令、交换指令和堆栈操作指令，实现将首地址为 BLOCK 的内存单元中两个数据字交换。BLOCK 变量定义如下：BLOCK DW 10H, 20H

**【解】**

(1) 用数据传送指令完成如下：

```
MOV AX, BLOCK
MOV BX, BLOCK+2      ; 由于是字数据，故 20H 的地址为 BLOCK+2
MOV BLOCK, BX
MOV BLOCK+2, AX
```

(2) 用交换指令完成如下：

```
XCHG AX, BLOCK
XCHG AX, BLOCK+2
XCHG BLOCK, AX
```

(3) 用堆栈指令完成如下：

```
PUSH BLOCK           ; 0010H 入栈
PUSH BLOCK+2         ; 0020H 入栈
POP BLOCK            ; 栈顶 0020H 出栈，存入 BLOCK
POP BLOCK+2          ; 0010H 出栈，存入 BLOCK+2
```

**【评注】** 注意：①因为操作数在存储器单元中，10H 和 20H 在指令中不能以立即数的形式出现；②由于是字数据，在取第二个字时，地址应为 BLOCK+2；③用堆栈指令来完成时，一定要按照先进后出的原则来操作。

**例 3.8** 试问下列程序段执行后，AL 和 CL 的内容分别是什么？

```
MOV AX, 1234H
MOV BX, 5678H
ADD AL, BL
DAA
MOV CL, AL
MOV AL, AH
ADC AL, BH
DAA
```

**【分析】** 由于使用了十进制调整指令 DAA，故相加的两个操作数必然是以 BCD 码的形式表示的，而且是以压缩格式的 BCD 码存放的。本题实际上是十进制 1234 和 5678 两个数相加，其和为 6912。根据程序，高位存于 AL，低位存于 CL 中。

**【解】** (AL) = 69, (CL) = 12

**例 3.9** 设一个字节数据 X 存放在 AL 寄存器中，试说明下列程序的功能。

```
XOR AH, AH
SAL AL, 1
```

```

MOV BX, AX
MOV CL, 2
SAL AX, CL
ADD AX, BX

```

【解】 SAL 为算术左移指令，如果无溢出，左移一位相当于乘以 2，左移两位相当于乘以 4，以此类推。本程序中首先对存于 AL 中的数 X 左移了一位，AX 中数变为了 2X，同时送入 BX，故 BX 中的数也为 2X；然后对 AX 左移了两位，即对 AX 扩大了 4 倍，这时 AX 中的数变为了 8X；再加上 BX 的值 2X，最后 AX 变为了 10X。故本程序段的功能为：X 扩大 10 倍后存入 AX 中。

例 3.10 下列程序是将两个 8 位组合 BCD 数（各占 4 个字节）相加求和的程序。设两 BCD 数已存入以 BUF1 和 BUF2 为首地址的内存单元中，且低位在低地址单元，高位在高地址单元，结果存回 BUF1 处。程序中含有若干错误，请改正。

```

.MODEL SMALL
.DATA
BUF1 DB 54H, 76H, 32H, 10H
BUF2 DB 23H, 88H, 06H, 20H
.CODE
.STARTUP
MOV CX, 4
LEA SI, BUF1
LEA DI, BUF2
CLC
NEXT: MOV AL, [SI]
      ADD AL, [DI]
      AAA
      MOV[SI], AL
      INC SI
      INC DI
      LOOP NEXT
.EXIT
END

```

【解】 本题的程序中有两个错误。

(1) ADD AL, [DI]指令不对。

因为对于多字节加法运算，高位部分对应相加时必须考虑低位上的进位情况，应改为 ADC AL, [DI]。

(2) AAA 指令不对。

由于 BUF1 和 BUF2 中存放的是组合 BCD 码，调整时就应该用组合型 BCD 码加法调整指令，而 AAA 为非组合型 BCD 码加法调整指令，应改为 DAA。

例 3.11 读程序段，说明其功能。（其中，DATA1、DATA2、DATA3 均为符号地址。）

```

(1) MOV AX, DATA1
    SUB AX, DATA2
    MOV DATA3, AX
    MOV AX, DATA1+2
    SBB AX, DATA2+2
    MOV DATA3+2, AX
(2) OR AX, 0FF00H
    XOR BX, 0F00FH
    AND CX, 00FFH
    XOR DX, DX

```

**【解】**

(1) 段程序的功能是：以 DATA1 为首地址的双字减去以 DATA2 为首地址的双字，结果存放在以 DATA3 为首的 4 个单元中。

(2) 段程序的功能：对 AX 的高 8 位置 1；对 BX 的高 4 位和低 4 位取反；对 CX 的高 8 位清 0；对 DX 寄存器清 0。

**例 3.12** 十进制数 0~9 与 BCD 码和字型码的相互转换关系见表 3.1。请编写程序，将 5 转换成共阳极 LED 显示的字型代码。

**【分析】**

(1) 首先应理解表 3.1：表中七段显示代码中高电平“1”表示不亮，低电平“0”表示亮。如：共阳极 LED 显示的字型码“0”（即十进制数“0”）只有 g 不亮，为高电平“1”。

表 3.1 BCD 码与字型码的相互关系

	十进制数	BCD 码	g	f	e	d	c	b	a	字型代码
TABLE+0	0	0000	1	0	0	0	0	0	0	40H
	1	0001	1	1	1	1	0	0	1	79H
	2	0010	0	1	0	0	1	0	0	24H
	3	0011	0	1	1	0	0	0	0	30H
	4	0100	0	0	1	1	0	0	1	19H
TABLE+5	5	0101	0	0	1	0	0	1	0	12H
	6	0110	0	0	0	0	0	1	0	02H
	7	0111	1	1	1	1	0	0	0	78H
	8	1000	0	0	0	0	0	0	0	00H
	9	1001	0	0	1	1	0	0	0	18H

(2) 对于顺序表，可用查表指令 XLAT 来实现查表功能。

**【解】**

(1) 设表格存放在内存的首址以标号 TABLE 表示。将表格首址的偏移地址放入 BX 中，将表格偏移量放入 AL 中。本例中偏移量为 05H

(2) 执行 XLAT 指令：EA= (BX) + (AL)

AL= (EA)

(3) 用如下指令实现：

```
MOV AX, SEG TABLE
MOV DS, AX
MOV BX, OFFSET TABLE
MOV AL, 05
XLAT
```

该程序执行后 (AL)=12H

【评注】查表指令用于查表很简便，应熟悉这个隐含寻址指令的执行过程。

例 3.13 算术运算综合举例。计算下式：

$$(S - (X \times Y + Z)) / X$$

其中 X、Y、Z、S 均为 16 位带符号数，已经分别装入 X、Y、Z、S 单元中，要求上式计算结果的商存入 AX，余数存入 DX 寄存器。试编制程序段完成算式功能。

【分析】X、Y、Z、S 均为 16 位带符号数，因此对于加、减运算要用到 ADC 和 SBB，对于乘、除运算要用到指令 IMUL 和 IDIV。X×Y 的积为 32 位数，Z 和 S 要进行双字扩展才能运算。因为除法运算被除数必须是除数位数的两倍。

【解】完成该算术运算式的程序段如下：

```
MOV AX, X
IMUL Y           ; X×Y
MOV CX, AX       ; CX 中存放积的低位
MOV BX, DX       ; BX 中存放积的高位
MOV AX, Z
CWD              ; Z 带符号扩展为 32 位
ADD CX, AX       ; 32 位 Z 与积的高、低 16 位分别相加
ADC BX, DX       ; X×Y+Z，高 16 位加时必须用 ADC 指令
MOV AX, S
CWD              ; S 带符号扩展为 32 位
SUB AX, CX       ; S-(X×Y+Z)
SBB DX, BX       ; 高 16 位减时必须用 SBB 指令
IDIV X           ; (S-(X×Y+Z))/X，商在 AX 中，余数在 DX 中
```

【评注】本例考查对有符号、无符号乘除指令及如何进行双字节加减运算的掌握情况。

### 3.3 习题及全解

#### 一、填空题

1. 计算机中的指令由\_\_\_\_\_和\_\_\_\_\_组成。
2. 指出下列指令源操作数的寻址方式：

(1) MOV AX, BLOCK[SI] \_\_\_\_\_

(2) MOV AX, [SI] \_\_\_\_\_

- (3) MOV AX, [6000H] \_\_\_\_\_
- (4) MOV AX, [BX+SI] \_\_\_\_\_
- (5) MOV AX, BX \_\_\_\_\_
- (6) MOV AX, 1500H \_\_\_\_\_
- (7) MOV AX, 80[BX+DI] \_\_\_\_\_
- (8) MOV AX, [DI+60] \_\_\_\_\_

3. 现有 (DS)=2000H, (BX)=0100H, (SI)=0002H, (20100H)=12H, (20101H)=34H, (20102H)=56H, (20103H)=78H, (21200H)=2AH, (21201H)=4CH, (21202H)=B7H, (21203H)=65H, 填入下列指令执行后 AX 寄存器的内容:

- (1) MOV AX, 1200H ; AX=\_\_\_\_\_
- (2) MOV AX, BX ; AX=\_\_\_\_\_
- (3) MOV AX, [1200H] ; AX=\_\_\_\_\_
- (4) MOV AX, [BX] ; AX=\_\_\_\_\_
- (5) MOV AX, 1100H[BX] ; AX=\_\_\_\_\_
- (6) MOV AX, [BX][SI] ; AX=\_\_\_\_\_
- (7) MOV AX, 1100H[BX][SI] ; AX=\_\_\_\_\_

4. 对于指令 XCHG BX, [BP+SI], 如果指令执行前, (BX)=6F30H, (BP)=0200H, (SI)=0046H, (SS)=2F00H, (2F246H)=154H, (2F247H)=41H, 则执行指令后: (BX)=\_\_\_\_\_, (2F246H)=\_\_\_\_\_, (2F247H)=\_\_\_\_\_。

5. 指令 LOOPZ/LOOPE 是结果\_\_\_\_\_且\_\_\_\_\_发生转移的指令; 而指令 LOOPNZ/LOOPNE 则是结果\_\_\_\_\_且\_\_\_\_\_发生转移的指令。

6. XLAT 指令规定 BX 寄存器中存放\_\_\_\_\_, AL 寄存器中存放\_\_\_\_\_。

7. 如果 BUF 为数据段中 5400H 单元的符号名, 从该单元开始连续两个单元存放的内容为 1234H, 执行指令 MOV BX, BUF 后, BX 的内容为\_\_\_\_\_; 而执行 LEA BX, BUF 后, BX 的内容为\_\_\_\_\_。

8. 串操作指令规定源串在\_\_\_\_\_段中, 用\_\_\_\_\_来寻址源操作数; 目的串在\_\_\_\_\_段中, 用\_\_\_\_\_来寻址目的操作数。

9. 近过程 (NEAR) 的 RET 指令把当前栈顶的一个字弹出到\_\_\_\_\_; 远过程 (FAR) 的 RET 指令弹出一个字到\_\_\_\_\_后又弹出一个字到\_\_\_\_\_。

10. 中断返回指令 IRET 执行后, 从堆栈顺序弹出 3 个字分别送到\_\_\_\_\_, \_\_\_\_\_, \_\_\_\_\_。

11. 设 (SS)=0FFA0H, (SP)=00B0H, (AX)=8057H, (BX)=0F79H, 执行指令 PUSH AX 后, (SP)=\_\_\_\_\_; 若再执行指令:

PUSH BX  
POP AX

后, (SP)=\_\_\_\_\_, (AX)=\_\_\_\_\_, (BX)=\_\_\_\_\_。

12. 阅读下面的程序。在\_\_\_\_\_的情况下, 本段程序的执行结果是



(AH)=0; 在\_\_\_\_\_的情况下, 其结果为(AH)=0FFH

```
IN AL, 20H
TEST AL, 80H
JZ A
MOV AH, 0
JMP B
A: MOV AH, 0FFH
B: HLT
```

13. 下列程序段是比较 AX、BX 和 CX 中带符号数的大小, 将最大的数放在 AX 中。请将程序填充完整。

```
CMP AX, BX
(1) _____ NEXT
XCHG AX, BX
NEXT: (2) _____ CX, AX
JLE OUT
(3) _____ AX, CX
OUT: .....
```

14. 若 (AX)=3F50H, (BX)=1728H, 执行 SUB AX, BX 指令后, (AX)=\_\_\_\_\_, 标志位 AF、SF、CF、ZF、OF、PF 的状态相应为\_\_\_\_\_。

15. 设 (SS)=2250H, (SP)=0140H, 若在堆栈中存入 5 个数据, 则栈顶的物理地址为\_\_\_\_\_, 如果再从堆栈中取出 3 个数据, 则栈顶的物理地址为\_\_\_\_\_。

16. MOV AL, 'A' 指令的源操作数的寻址方式为\_\_\_\_\_。

17. 若 BP=1500H, SI=2500H, DS=2000H, ES=3000H, SS=4000H, 假设按 16 位实模式操作, 则指令 MOV AL, [BP+SI+4200H], 其源操作数的寻址方式为\_\_\_\_\_, 访问内存的物理地址为\_\_\_\_\_。

18. MOV EAX, [BP+5000H], 若按实模式操作, 设 BP=1500H, DS=2000H, ES=3000H, SS=4000H, 则源操作数的物理地址为\_\_\_\_\_。

19. 指出下列每条指令中源操作数的寻址方式:

- (1) MOV AX, [4000H]; \_\_\_\_\_
- (2) MOV AL, [BX+DI]; \_\_\_\_\_
- (3) MOV EAX, [BP+5000H]; \_\_\_\_\_
- (4) MOV AL, [EBX\*4+80H]; \_\_\_\_\_

20. 实地址方式的数据寻址方式有\_\_\_\_\_。

21. 对于串操作指令, 要控制变址寄存器的内容自动递增, 用\_\_\_\_\_指令实现。

22. 将 AX 寄存器的内容清“0”的指令有\_\_\_\_\_。

23. 以 CX 寄存器内容为计数对象的指令有\_\_\_\_\_。

24. CPU 工作在实方式下, 若指令 MOV AL, [BP+SI], 其源操作数的寻址方式为\_\_\_\_\_。

\_\_\_\_\_，假设 BP=1500H，SI=2500H，ES=3000H，SS=4000H，则源操作数的物理地址为\_\_\_\_\_。

25. 输入/输出指令间接寻址必须通过\_\_\_\_\_寄存器。

【解】

1. 操作码；操作数
2. (1) 寄存器相对寻址；(2) 寄存器间接寻址；(3) 直接寻址；(4) 基址变址寻址；(5) 寄存器寻址；(6) 立即寻址；(7) 相对基址变址寻址；(8) 寄存器相对寻址
3. (1) 1200H；(2) 0100H；(3) 4C2AH；(4) 3412H；(5) 4C2AH；(6) 7856H；(7) 65B7H
4. 4154H；30H；6FH
5. 为零；相等；不为零；不相等
6. 表的首地址；表内偏移量
7. 1234H；5400H
8. 数据；(E)SI；附加；(E)DI
9. IP；IP；CS
10. IP；CS；标志寄存器 FR
11. 00AEH；00AEH；0F79H；0F79H
12. AL 中的最高位为 1；AL 中的最高位为 0
13. (1) JGE；(2) CMP；(3) XCHG
14. 2828H；1、0、0、0、0、1
15. 22636H；2263CH
16. 立即寻址方式
17. 相对基址变址寻址；47C00H
18. 46500H
19. (1) 直接寻址；(2) 基址变址寻址；(3) 相对变址寻址；(4) 相对比例变址寻址
20. 立即寻址、寄存器寻址和存储器寻址
21. CLD
22. ①MOV AX, 0；②XOR AX, AX；③SUB AX, AX；④AND AX, 0
23. 循环控制指令和串操作重复指令
24. 基址变址寻址；43A00H
25. DX

## 二、选择题

1. MOV AX, [BX+SI]的源操作数的物理地址是\_\_\_\_\_。  
A. (DS)×16+(BX)+(SI)      B. (ES)×16+(BX)+(SI)  
C. (SS)×16+(BX)+(SI)      D. (CS)×16+(BX)+(SI)
2. MOV AX, [BP+SI]的源操作数的物理地址是\_\_\_\_\_。  
A. (DS)×16+(BP)+(SI)      B. (ES)×16+(BP)+(SI)

- C.  $(SS) \times 16 + (BP) + (SI)$                       D.  $(CS) \times 16 + (BP) + (SI)$
3. MOV AX, ES: [BX+SI]的源操作数的物理地址是\_\_\_\_\_。
- A.  $(DS) \times 16 + (BX) + (SI)$                       B.  $(ES) \times 16 + (BX) + (SI)$
- C.  $(SS) \times 16 + (BX) + (SI)$                       D.  $(CS) \times 16 + (BX) + (SI)$
4. JMP NEAR PTR[DI]是\_\_\_\_\_。
- A. 段内间接转移                                      B. 段间间接转移
- C. 段内直接转移                                      D. 段间直接转移
5. JMP FAR PTR BLOCK (BLOCK 是符号地址) 是\_\_\_\_\_。
- A. 段内间接转移                                      B. 段间间接转移
- C. 段内直接转移                                      D. 段间直接转移
6. INC 指令不影响\_\_\_\_\_标志。
- A. OF                                      B. CF                                      C. SF                                      D. ZF
7. 条件转移指令 JNE 的测试条件是\_\_\_\_\_。
- A. ZF=1                                      B. CF=0                                      C. ZF=0                                      D. CF=1
8. 下列指令中, 有语法错误的是\_\_\_\_\_。
- A. MOV[SI], [DI]
- B. IN AL, DX
- C. JMP WORD PTR[BX+8]
- D. PUSH WORD PTR 20[BX+SI]
9. 假定(SS)=2000H, (SP)=0100H, (AX)=2107H, 执行指令 PUSH AX 后, 存放数据 21H 的物理地址是\_\_\_\_\_。
- A. 20102H                                      B. 20101H                                      C. 200FEH                                      D. 200FFH
10. 对于下列程序段:

```
AGAIN: MOV AL, [SI]
        MOV ES: [DI], AL
        INC SI
        INC DI
        LOOP AGAIN
```

也可用指令\_\_\_\_\_完成同样的功能。

- A. REP MOVSB                                      B. REP LODSB
- C. REP STOSB                                      D. REPE SCASB
11. 对于下列程序段:

```
AGAIN: MOV ES: [DI], AL
        INC DI
        LOOP AGAIN
```

可用指令\_\_\_\_\_完成。

- A. REP MOVSB                      B. REP LODSB  
C. REP STOSB                      D. REP SCASB
12. 交换寄存器 SI 和 DI 的内容，正确的程序段是\_\_\_\_\_。
- A. PUSH SI                      B. PUSH SI  
PUSH DI                      PUSH DI  
POP SI                      POP DI  
POP DI                      POP SI
- C. MOV AX, SI                      D. MOV AX, SI  
MOV SI, DI                      MOV BX, DI  
MOV DI, AX                      XCHG BX, AX
13. 在 ADD 指令中，两个操作数的物理位置可以安排在\_\_\_\_\_中。
- A. 两个存储单元                      B. 一个存储单元和一个数据寄存器  
C. 两个数据寄存器                      D. 一个堆栈单元和一个数据寄存器
14. 将字变量 ARRAY 的偏移地址送寄存器 BX 的正确结果是\_\_\_\_\_。
- A. LEA BX, ARRAY                      B. MOV BX, ARRAY  
C. MOV BX, OFFSET ARRAY                      D. MOV BX, SEG ARRAY
15. 将累加器 AX 的内容清 0 的正确指令是\_\_\_\_\_。
- A. AND AX, 0                      B. XOR AX, AX  
C. SUB AX, AX                      D. CMP AX, AX
16. 下列指令中，正确的是\_\_\_\_\_。
- A. MOV [DI], [SI]                      B. MOV DS, SS  
C. MOV AL, [EAX+EBX\*2]                      D. OUT BX, AX
17. 实现将 AL 寄存器中的低 4 位置 1 的指令为\_\_\_\_\_。
- A. AND AL, 0FH                      B. OR AL, 0FH  
C. TEST AL, 0FH                      D. XOR AL, 0FH
18. 指令 LOOPE X1, \_\_\_\_\_。
- A. 当 CX≠0 或者 ZF=0 时转移到 X1  
B. 当 CX≠0 或者 ZF=1 时转移到 X1  
C. 当 CX=0 而且 ZF=1 时转移到 X1  
D. 当 CX≠0 而且 ZF=0 时转移到 X1  
E. 当 CX≠0 而且 ZF=1 时转移到 X1  
F. 当 CX=0 而且 ZF=0 时转移到 X1
19. 80486 在实地址方式下，内存管理中段基址、段首地址、偏移地址、物理地址依次是\_\_\_\_\_。
- A. 16 位、20 位、16 位、20 位  
B. 均为 16 位  
C. 均为 20 位  
D. 20 位、16 位、20 位、16 位

20. 80486 在保护方式下, 内存管理中段基址、段首地址、偏移地址、物理地址依次是\_\_\_\_\_。

- A. 20 位、32 位、20 位、32 位
- B. 均为 32 位
- C. 均为 20 位
- D. 16 位、32 位、16 位、32 位

21. 采用 BCD 码表示的十进制数计算时, 先调整后运算的有\_\_\_\_\_。

- A. 加法      B. 减法      C. 乘法      D. 除法

22. 十进制调整指令 AAA\_\_\_\_\_。

- A. 跟在 SUB 或 SBB 之后
- B. 跟在 ADD 之后
- C. 跟在 MUL 或 ADC 之后
- D. 跟在 ADD 或 ADC 之后

【解】

1. A 2. C 3. B 4. A 5. D 6. B 7. C 8. A 9. D 10. A 11. C 12. A、C  
13. B、C、D 14. A、C 15. A、B、C 16. C 17. B 18. E 19. A 20. B 21. D  
22. B、D

### 三、判断题

1. MOV AX, [BP+SI]的源操作数的物理地址为(DS)×16+(BP)+(SI)。
2. 段内转移要改变 IP 和 CS 的值。
3. 立即寻址方式立即数不能用于目的操作数字段。
4. 不能给段寄存器进行立即数方式赋值。
5. OF 位用来表示带符号数的溢出, CF 位可以表示无符号数的溢出。
6. SP 的内容在任何时候都指向当前的栈顶, 要指向堆栈的其他位置, 可以使用 BP 指针。
7. REPE/REPZ 是相等/为零时重复操作, 其退出条件是: (CX)=0 或 ZF=1。
8. 指令中都必须有操作数。
9. 立即数可以直接送给寄存器、存储器或者段寄存器。
10. 在串指令使用前, 必须先将 DF 置 0。
11. 在用循环控制指令时, 必须将循环次数送 CX。
12. 十进制调整指令是把累加器中十六进制数转换成十进制数。
13. 加、减、乘、除运算指令都分带符号数和不带符号数运算指令。
14. 基址变址寻址是在基址寄存器和变址寄存器中寻找操作数。
15. 算术左移指令和逻辑左移指令在操作上是相同的, 而算术右移指令和逻辑右移指令在操作上是不相同的。
16. 请判断正误:  
(1) MOV AX, DX;

- (2) IN AL, DX;
- (3) MOV EBX, [BX+SI];
- (4) MOV [BX], [BP];
- (5) INC [EAX];
- (6) SUB AL, 2030H.
- (7) RCR AL;
- (8) XCHG AL, 80H;
- (9) ADD ECX, [BX+10H];
- (10) IDIV BI, ;

17. 在串操作中必须将串长度送 CX。

【解】

1. 错误 2. 错误 3. 正确 4. 正确 5. 正确 6. 正确 7. 错误 8. 错误 9. 错误 10. 错误 11. 正确 12. 正确 13. 错误 14. 错误 15. 正确 16. (1) 正确; (2) 正确; (3) 正确; (4) 错误; (5) 错误; (6) 错误; (7) 错误; (8) 错误; (9) 正确; (10) 正确 17. 错误

#### 四、问答题

1. 指出下列算术和逻辑指令执行后标志 CF、ZF、SF、PF、OF 和 AF 的状态。

```
MOV AL, 80H
DEC AL
ADD AL, 10H
SUB AL, 10H
MOV AL, 3AH
AND AL, 0F0H
OR AL, 0F0H
XOR AL, 0F0H
```

2. 试指出下列指令中的错误:

- |                     |                     |
|---------------------|---------------------|
| (1) MOV[BX], [SI]   | (2) MOV AH, DX      |
| (3) INC [BX]        | (4) MOV DS, SS      |
| (5) XCHG AX, 2000H  | (6) MOV AX, [BX+DX] |
| (7) XCHG[BP], ES    | (8) ADD[BX], BX     |
| (9) MOV AX, [DI+SI] | (10) IN AL, BX      |

3. 什么叫串? 串操作有哪些基本的指令? 在使用时它们的寻址方式有哪些约定? 串前缀在什么情况下使用?

4. 判断正误, 错者指出错误并改正。

- (1) MOV ES, DS
- (2) OUT BX, AX
- (3) XCHG AL, 66H

- (4) MOV EAX, [ECX\*4+80H]
- (5) INC [EBX]
- (6) MOV [BX], [DI]
- (7) SUB AL, 2000H
- (8) IDIV AX, BL
- (9) CMP [BX], 0
- (10) POP AL

## 【解】

## 1. MOV AL, 80H

DEC AL	; CF=0 ZF=0 SF=0 PF=0 OF=1 AF=1
ADD AL, 10H	; CF=0 ZF=0 SF=1 PF=0 OF=1 AF=0
SUB AL, 10H	; CF=0 ZF=0 SF=0 PF=0 OF=1 AF=0
MOV AL, 3AH	
AND AL, 0F0H	; CF=0 ZF=0 SF=0 PF=1 OF=0
OR AL, 0F0H	; CF=0 ZF=0 SF=1 PF=1 OF=0
XOR AL, 0F0H	; CF=0 ZF=1 SF=0 PF=0 OF=0

- 2. (1) 错误。源操作数和目的操作数不允许同时为存储器操作数。
- (2) 错误。源操作数和目的操作数的数据类型必须一致。
- (3) 错误。操作数的数据类型不明确，应使用 PTR 伪指令来定义。
- (4) 错误。源操作数和目的操作数不能同时为段寄存器。
- (5) 错误。交换指令中不能出现立即数。
- (6) 错误。源操作数寻址方式不对，可将 DX 改为 DI 或 SI。
- (7) 错误。不能对段寄存器交换数据。
- (8) 正确。
- (9) 错误。源操作数寻址方式不对。
- (10) 错误。输入指令若是间接寻址，端口地址不能用 BX 而只能用 DX。

3. 串是一组有序的在内存连续存放的数据。串基本操作指令有串传送如：MOVS/ MOVSW、串比较如：CMPSB/CMPSW、串扫描如：SCASB/SCASW、串存储（写串）如：STOSB/STOSW、串装入（读串）如：LODSB/LOSW 共 5 种。串前缀是在串操作过程中无需再做其他处理的情况下使用。

- 4. (1) 错误。因段寄存器之间不能互相传送，可改为：

```
MOV AX, DS
MOV ES, AX
```

- (2) 错误。应改为：OUTDX, AX
- (3) 错误。交换指令中不应有立即数，可改为：XCHG AL, [0066H]
- (4) 正确。
- (5) 错误。操作数类型不明确，可改为：INC BYTE PTR[EBX]

(6) 错误。两存储单元不能直接互相传送数据，可改为：

```
MOV AL, [DI]
MOV[BX], AL
```

(7) 错误。两操作数类型不匹配，应改为：SUB AX, 2000H

(8) 错误。IDIV 指令被除数隐含指定，应改为：IDIV BL

(9) 错误。操作数类型不明确，可改为：CMP BYTE PTR[BX], 0

(10) 错误。堆栈操作不允许字节操作，应改为：POP AX

## 五、编程题

1. 下面程序是查找 **STRING** 中是否有“A”这个字符，如果有则转向 **YES** 去执行；没有则转向 **NO** 去执行。请将程序填充完整。

```
        ; 假设 STRING 的长度为 20
        MOV CX, 20
        MOV BX, -1
        MOV AL, "A"
NEXT: (1) _____ BX
        CMP AL, STRING[BX]
        (2) _____ NEXT
        JNZ (3) _____
YES:   .....
        .....
        JMP (4) _____
NO:    .....
        .....
EXIT:  RET
```

2. 使 AX 寄存器清 0 有多种方式，试写出这多条指令。

3. 试编程实现：

(1) AL 寄存器的低 4 位清 0

(2) BL 寄存器的低 4 位置 1

(3) CL 寄存器的低 4 位取反

(4) 测试 DL 寄存器的最低两位是否为 0，若是将 0 送入 AL 寄存器；否则将 1 送入 AL 寄存器

4. 试编程统计在 AX 寄存器中有多少个 1，并将结果送 DL 寄存器中。

5. 试编程统计在内存 **BLOCK** 单元开始按字节存放的 100 个带符号数中有多少个负数，并将结果存放在 DL 寄存器中。

6. 有下列程序段：

```
SAL BX, 1
RCL AX, 1
```



RCI, DX, 1

(1) 说明此程序段实现的功能。

(2) 若(DX)=1002H, (AX)=3004H, (BX)=8006H, 问运行该段程序后相应 DX:

AX: BX 中的数据内容是多少?

7. 请写出实现下列功能的指令或伪指令:

(1) AL 寄存器的低 4 位清 0, 高 4 位不变的一条指令。

(2) BL 寄存器的低 4 位置 1, 高 4 位不变的一条指令。

(3) CL 寄存器的低 4 位取反, 高 4 位不变的一条指令。

(4) AX 寄存器 4 种方式清 0 的 4 条指令。

(5) BLOCK2 单元开始连续 10 单元清 0 的一条伪指令。

8. 读程序段, 说明其功能(其中, BLOCK 为符号地址)。

```
MOV AL, BLOCK
MOV AH, 0
SAL AX, 1
MOV BX, AX
SAL AX, 2
ADD AX, BX
```

#### 【解】

1. (1) INC (2) LOOPNE (3) NO (4) EXIT

2. (1) MOV AX, 0 (2) AND AX, 0 (3) XOR AX, AX (4) SHL AX, 16

(5) SHR AX, 16 (6) SAL AX, 16 (7) SUB AX, AX

3. (1) AND AL, 0F0H (2) OR BL, 0FH (3) XOR CL, 0FH

(4) TEST DL, 03H

JZ P1

MOV AL, 1

JMP P2

P1: MOV AL, 0

P2: HLT

4. .MODEL TINY

.CODE

.STARTUP

MOV DL, 0

MOV CL, 16

AGAIN: ROL AX, 1

JNC NEXT

INC DL

NEXT: DEC CL

JNZ AGAIN

```

.EXIT
END

5.  .MODEL SMALL
    .DATA
BLOCK DB 1, -2, 5, 6, -57, .....; 100 个带符号数
    .CODE
    .STARTUP
    MOV DL, 0
    LEA BX, BLOCK
    MOV CX, 100
AGAIN: CMP BYTE PTR [BX], 0
       JGE NEXT
       INC DL
NEXT:  INC BX
       LOOP AGAIN
    .EXIT
    END

```

6. (1) 48 位数乘以 2    (2) 2004H: 6009H: 000CH

7. (1) AND AL, 0F0H    (2) OR BL, 0FH    (3) XOR CL, 0FH    (4) XOR AX, AX/MOV AX, 0/AND AX, 0/SUB AX, AX    (5) BLOCK2 DB 10 DUP (0)

8. 将 BLOCK 单元中的数扩大 10 倍存入 AX 中。

## 第 4 章

# 汇编语言、程序设计 及其与高级语言调用

### 4.1 重点内容提要

#### （一）汇编语言

汇编语言是符号化了的机器语言。用汇编语言编写的程序被称为汇编语言源程序。源程序只有经过汇编过程才能得到目标程序。汇编时一般采用 MASM 汇编程序。

##### 1. 汇编语言格式

汇编语言的语句一般有两类，伪指令语句和指令语句。指令语句有其对应的机器码，汇编程序在对源程序进行汇编时，把指令语句翻译成机器指令。而伪指令没有其对应的机器码，只是指示汇编程序如何进行汇编。

指令语句格式：[标号:]助记符 操作数 ； 注释

伪指令语句格式：[名字]定义符 参数 ； 注释

程序中标号定义了指令的逻辑地址。标号具有段地址、偏移地址和类型属性。标号的段地址和偏移地址是指标号对应的指令首字节所在的段地址和段内的偏移地址。标号的类型属性有两种：NEAR 和 FAR 类型。NEAR 指示近程（段内）标号，该标号只能为本段内指令使用；FAR 指示远程（段间）标号，该标号可以为其他段使用。在转移和调用指令中常将标号作为转移目标地址使用。

变量定义了数据的逻辑地址。变量具有段地址、偏移地址和类型属性。类型属性有 BYTE、WORD、DWORD、PWORD、QWORD 等。

##### 2. 常用的伪指令

（1）数据定义伪指令：该指令格式为：变量 定义符 表达式 1，表达式 2，…

定义符一般是以下几种。

DB——定义字节变量。

DW——定义字变量。

DD——定义双字变量。

（2）简化的段定义伪指令：高版本的 MASM 6.X 提供了简化的段定义伪指令。简化的段结构中常用伪指令说明如下。

.X86——用于选择 80x86 指令系统。

.X86P——用于选择 80x86 保护模式指令系统。

- .STARTUP——指示程序开始，初始化 DS 和 SS 寄存器。
- .EXIT——使程序返回 DOS 操作系统。
- .CODE——定义程序段。
- .DATA——定义数据段。
- .STACK——定义堆栈段，其后可根据参数，定义堆栈大小。
- .MODEL——内存模式说明。内存模式如表 4.1 所示。

表 4.1 常用内存模式

内存模式	说明
TINY	程序和数据在 64KB 段内
SMALL	独立的代码段（64KB 内）和独立的数据段（64KB 内）
MEDIUM	单个数据段（64KB 内），多个代码段
COMPACT	单个代码段（64KB 内），多个数据段
LARGE	多个代码段和多个数据段

(3) 完整段定义伪指令：该指令格式为：

段名 SEGMENT [属性]

...

段名 ENDS

属性：[段合并属性][段类属性][段对齐属性][段长度属性]

当定义 DS、ES 和 SS 时，在 SEGMENT / ENDS 伪指令中间的语句，只能包括伪指令语句，不能包括指令语句。只有当定义 CS 时，中间的语句才能为指令语句以及与指令有关的伪指令语句。

段合并属性用来告诉连接程序本段与其他段的组合关系。可供选择的组合类型有 5 种。

PRIVATE——段不与其他同名段合并（默认）。

PUBLIC——段与其他同名段合并为单个连续段。

COMMON——定位段起始处放置其他同名段，段长为最长段长。

MEMORY——将它定位于其他段之上的地址。

AT 表达式——定位段起始于表达式指定的 16 位段值处。

段类属性表明该段的类别，有如下 4 种。

‘CODE’——代码段。

‘DATA’——数据段。

‘STACK’——堆栈段。

‘EXTRA’——附加段。

类别必须是用单引号括起来的字符串。具有相同类别的段，不管其段名是否相同，在连接时，将所有相同类别的逻辑段连接在一起，形成一个统一的物理段。

段对齐属性对段的起始边界进行定位。它有如下 5 种定位方式。

BYTE——段从任意字节开始。

WORD——段从下一字地址处开始。

DWORD——段从下一个双字地址处开始。

PARA——段从下一节地址处开始（默认 16B 为一节）。

PAGA——段从下一页地址处开始（256B 为一页）。

段长度属性有：

USE16——段为 16 位，段长 64KB。

USE32——段为 32 位，段长 4GB。

（4）段寄存器定义伪指令：其格式如下所示。

ASSUME 段寄存器：段名[，段寄存器：段名]

该指令用来规定给定段的段地址寄存器，段寄存器包括 CS、DS、ES、SS、FS 和 GS。

（5）段结束伪指令：如果采用简化的段定义结构，则其格式为 END。如果采用完整的段定义结构，则其格式为 END 标号。

这里，标号为第一条可执行语句的标号。

### 3. 控制汇编语言程序语句

MASM 6.X 版本提供了控制程序流程的 3 种汇编语句。

（1）IF 语句。

格式：

```
.IF      表达式
          语句 1
          .ELSE
          语句 2
          .ENDIF
```

（2）DO-WHILE 语句。

格式：

```
.WHILE  表达式
          语句
          .ENDW
```

（3）REPEAT-UNTIL 语句。

格式：

```
.REPEAT
          语句
          .UNTIL 表达式
```

## （二）汇编语言程序设计

### 1. 顺序程序设计

顺序程序是一种最简单的程序，每条指令按其在程序中的排列顺序执行。

### 2. 分支程序设计

程序的分支主要靠条件转移指令来实现。这里需要注意的是，条件转移指令都是近程跳转。若程序所要转移的地址超出其范围时，需利用一条无条件转移指令作为中转。

### 3. 循环程序设计

循环程序设计主要用于某些需要重复进行的操作，主要使用循环指令 LOOP 或条件

转移指令来实现循环。循环程序的设计可分为设置循环初始状态、循环体和循环控制条件 3 部分。

(1) 设置循环初始状态：该状态主要是指设置循环次数的计数初值，以及其他为使循环体正常工作而设置的初始状态等。

(2) 循环体：它是循环操作（重复执行）的部分，包括循环的工作部分及修改部分。循环的工作部分是实现程序功能的主要程序段；循环的修改部分是指当程序循环执行时，对一些参数如地址、变量的有规律的修正。

(3) 循环控制部分：它是循环程序设计的关键。每个循环程序必须选择一个控制循环程序运行和结束的条件。

#### 4. 子程序设计

子程序是一段独立的指令组，一般具有通用性，可共享、可被一个程序反复使用或被多个程序调用。采用子程序设计可以缩短程序长度、节省内存空间。在 80x86 汇编程序中，子程序常常以过程的形式出现。

(1) 过程定义

格式：过程名      PROC[TYPE][USES REG]  
                          ...  
                          RET  
                          过程名    ENDP

**TYPE**：过程类型，分为近程 NEAR 和远程 FAR，当类型为 NEAR 时，该过程可以被本段调用，当类型为 FAR 时，该过程可以被其他程序段调用。

**USES REG**：规定需要保护的寄存器，在过程开始将指定通用寄存器内容压入堆栈，在返回之前再恢复这些寄存器的内容。

(2) 过程调用指令

段内直接调用：CALL 过程名

段内间接调用：CALL NEAR PTR[REG]或 CALL REG

段间直接调用：CALL FAR PTR 过程名

段间间接调用：CALL FAR PTR [REG]

(3) 过程返回指令

格式：RET 或 RET N

过程返回指令一般放在过程体的末尾，用来返回主程序。RET 后面可以指定一个 N 值（为偶数），其功能为完成 RET 操作后根据 N 值修改堆栈指针（E）SP，目的是用于清除过程调用时入栈的参数。

(4) 寄存器内容的保护和恢复：即保护现场和恢复现场。除了在定义过程时用 USES REG 伪指令实现外，还可以在过程一开始，先把过程中要改变的寄存器的内容用 PUSH 指令压入堆栈，然后在返回前，用 POP 指令恢复这些寄存器的内容。

(5) 主程序和过程间的参数传递：主程序调用过程时，往往要将入口参数传递给子程序；子程序在执行完时，要将出口参数传递给主程序。参数传递可以有 3 种形式：①通过寄存器传递参数；②通过地址表传递参数；③通过堆栈传送参数或参数地址。

### （三）汇编语言与 C 语言的混合编程

由于使用高级语言开发效率高且移植性好，所以高级语言使用更广泛。但在要求执行速度快占用内存少或要求直接控制硬件的场合，仍然要用到汇编语言程序。对于一个具体任务，编程者往往是扬长避短使用高级语言和汇编语言混合编程。

汇编语言和 C 语言混合编程的方法有两种：在 C 语言程序中直接嵌入汇编代码或直接调用汇编语言子程序。

#### 1. 在 C 语言程序中直接嵌入汇编代码

在 C 语言程序中直接嵌入的汇编语句，要用关键字 `asm` 说明。其格式为：`asm 操作码 操作数`；

对于内嵌汇编指令的 C 程序可以采用命令行的编译连接方法。对于 TURBO C 程序，编译连接的命令行是：

`TCC_B_L: \LIB 文件名 库文件名`

注意汇编时 TCC 要用到 `TASM.exe` 程序，如果没有此程序，可以把 `MASM (3.0 以上)` 改名为 `TASM.exe` 代替，否则编译时会出错。

#### 2. 在 C 语言程序中直接调用汇编程序

C 语言程序可以调用汇编语言编写的子程序及定义的变量，汇编子程序也可以调用 C 语言编写的函数和定义的变量。若 C 程序调用汇编语言中的子程序或变量，则在汇编语言中应该用 `public` 进行说明，而在 C 语言中则应将其说明为 `extern`。C 程序调用汇编子程序时，参数是通过堆栈传递给汇编子程序的，要注意 C 语言程序参数入栈的顺序是从右到左，在执行汇编子程序前还要将返回地址压入堆栈。当被调用的汇编子程序有值返回给调用它的 C 程序时，这个值是通过 `AX` 和 `DX` 寄存器进行传递的。

对于用 C 语言和汇编语言分别独立编写的程序，要想使它们最终形成一个系统，需要对它们进行编译和连接，以便生成一个可执行文件。可以以工程的方式进行，其步骤如下所示。

（1）用汇编程序将汇编语言子程序汇编生成相应的目标文件 `XXX.obj`。

（2）在工程文件 `X.prj` 中加入编译连接的 C 语言程序及其调用的汇编子程序目标文件名。

（3）对工程文件进行编译连接，生成一个 `.exe` 可执行文件。

使用这种方法要注意 C 语言对大、小写敏感的问题，最好关闭大、小写敏感开关。

## 4.2 常考题型范例精解

**例 4.1** 读以下程序，指出它完成的功能。

```
.MODEL SMALL
.DATA
STR1 DB 300 DUP(?)
STR2 DB 100 DUP(?)
.CODE
```

```

        .STARTUP
        MOV CX,100
        MOV BX,200
        MOV SI,0
        MOV DI,0
NEXT:   MOV AL,STR1[BX][SI]
        MOV STR2[DI],AL
        INC SI
        INC DI
        LOOP NEXT
        .EXIT
        END

```

**【解】** 本程序完成将 STR1 的后 100 个数据传送到 STR2 中。

**【评注】** 这是一个简单的循环程序，注意其循环初始状态、循环体和循环控制条件。

**例 4.2** 下面程序的功能是：数组 A 包含 15 个互不相等的整数，数组 B 包含 20 个互不相等的整数，将既在数组 A 又在数组 B 中出现的整数存放在数组 C 中。请将程序填充完整。

```

        .MODEL SMALL
        .DATA
A   DW 15 DUP(?)
B   DW 20 DUP(?)
C   DW 15 DUP(?)
        .CODE
        .STARTUP
        MOV SI,0
        MOV BX,0
        MOV CX,15
NEXT1:  PUSH CX
        MOV DI,0
        MOV CX,20
        MOV AX,(1) _____
NEXT2:  CMP B[DI],AX
        JNE (2) _____
        MOV C[BX],AX
        INC BX
        INC BX
ABCD:   INC DI
        INC DI
        LOOP (3) _____
        INC SI
        INC SI
        (4) _____ CX
        LOOP NEXT1
        .EXIT
        END

```



**【解】** 这是一个两重循环程序。其内层循环完成一个数与数组 B 的每个元素进行比较,若相等,则送入数组 C 中,共比较 20 次;外层循环用来控制送入内层循环的待比较的数,将数组 A 中的每个数逐个送入,共进行 15 次。由于内层循环和外层循环都用 LOOP 来进行循环控制,并在两层循环的循环次数都要用到 CX,并在进入内层循环之前,必须保护 CX,回到外层循环时要恢复 CX。根据题意和程序,各空应该填: (1) A[SI]; (2) ABCD; (3) NEXT2; (4) POP。

**例 4.3** 在 STRING~STRING+99 的单元中存放着一个字符串,下列程序段实现测试该字符串中是否存在数字,若有数字则将 DL 的各位置 1,否则各位清 0。程序中有若干错误,试改正。

```
.MODEL SMALL
.DATA
STRING DB 100 DUP(?)
.CODE
.STARTUP
MOV CX,99
MOV SI,OFFSET STRING
REPEAT: MOV AL,[SI]
        CMP AL,30H
        JL NEXT
        CMP AL,39H
        JG NEXT
        AND DL,0FFH
        JMP DOWN
NEXT:   INC SI
        LOOP REPEAT
        OR DL,00H
DOWN:  .EXIT
END
```

**【解】** 本程序共有 5 处错误。

(1) MOV CX, 99 应改为 MOV CX, 100, 因为循环次数是从 STRING+0~STRING+99 的单元数,应是 100。

(2) JL NEXT 应改为 JB NEXT, 因为 ASCII 字符表示的十进制数的判断应用无符号数比较条件转移指令。

(3) JG NEXT 应改为 JA NEXT, 原因同上。

(4) AND DL, 0FFH 应改为 OR DL, 0FFH, 因为对 DL 各位置 1, 要用 OR 指令, AND 指令只能用来清 0。

(5) OR DL, 00H 应改为 AND DL, 00H, 因为对 DL 各位清 0, 要用 AND 指令, OR 指令只能用来置 1。

**例 4.4** 阅读程序:

```
.MODEL SMALL
```

```

        .DATA
BLOCK    DB  82H,97H,32H,0DBH
          DB  56H,9AH,0B7H,78H
BUFFER1  DB  8 DUP(?)
BUFFER2  DB  8 DUP(?)
        .CODE
        .STARTUP
MOV  AX,@DATA
MOV  ES,AX
CLD
LEA  SI,BLOCK
LEA  DI,BUFFER1
LEA  BX,BUFFER2
MOV  CX,8
LP:    LODSB
        TEST AL,80H
        JNZ MI
        STOSB
        JMP AGAIN
MI:     XCHG BX,DI
        STOSB
        XCHG BX,DI
AGAIN:  LOOP LP
        .EXIT
END

```

试问：若 DS=4000H，BLOCK=0000H，则程序执行结束后，内存单元[40007H]、[40009H]、[40013H]以及寄存器 AL、BX 的值分别是什么？

**【解】** 根据程序分析，本程序的功能是将以 BLOCK 为首的 8 个单元中的数据按正、负分开，正数存入以 BUFFER1 为首的单元中，负数存入以 BUFFER2 为首的单元中。因此以 BUFFER1 为首的 8 个单元中的数依次为 32H、56H、78H、?、?、?、?、?；以 BUFFER2 为首的 8 个单元中的数依次为 82H、97H、0DBH、9AH、0B7H、?、?、?。

因此，若 DS=4000H，BLOCK=0000H，则[40007H]=78H，[40009H]=56H，[40013H]=9AH，(AL)=78H，(BX)=0015H。

**例 4.5** 下面程序完成两个 ASCII 码的十进制乘法(ASCX×ASCY)，其结果为 ASCII 码的十进制数。

- (1) 试在程序的空白行填入正确的语句。
- (2) 给出程序运行后，以 PRODUCT 变量为地址的内存单元的内容。

```

        .MODEL SMALL
        .DATA
ASCX    DB  '1234'
ASCY    DB  '5'

```

```

PRODUCT DB 6 DUP(0)
        .CODE
        .STARTUP
        MOV CX,4
        LEA SI,ASCX
        ADD SI,CX
        DEC SI
        LEA DI,PRODUCT
        ADD DI,CX
        AND ASCY,0FH
NEXT:    MOV AL,[SI]
        AND AL,0FH
        MUL ASCY
        ①_____
        ADD AL,[DI]
        AAA
        ②_____
        MOV[DI],AL
        DEC DI
        MOV[DI],AH
        DEC SI
        ③_____
        .EXIT
END

```

**【分析】** ASCII 码表示的十进制数为 30H~39H, 屏蔽掉高 4 位, 变为 00H~09H, 再进行十进制的乘运算, 并在乘指令后加入十进制乘法调整指令 AAM, 加指令后跟非组合 BCD 码加法调整指令 AAA, 积再加 30H 变为 ASCII 码十进制数, 存入以 PRODUCT 为首地址的内存单元中。

**【解】**

(1) 根据题意, 填空如下: ①AAM; ②ADD AL, 30H; ③LOOP NEXT。

(2) 程序运行后, 以 PRODUCT 变量为地址的内存单元中的内容为: 36H、31H、37H、30H、00H、00H。

**例 4.6** 若 X、Y、Z 是连续存放在内存 BLOCK 开始的 3 个带符号字节数, 试编写出计算:  $X \times Y - Z$  的汇编语言源程序, 结果存入 RESULT 单元中。

```

【解】      .MODEL SMALL
              .DATA
BLOCK DB 15H,20H,30H
RESULT DW ?
              .CODE
              .STARTUP
              MOV BX,OFFSET BLOCK
              MOV AX,0

```

```

MOV AL,[BX]                ; X 为被乘数
IMUL BYTE PTR[BX+01H]      ; X×Y
MOV CX,AX                  ; 积存于 CX 中
MOV AL,[BX+02H]
CBW                        ; 对 Z 进行扩展
SUB CX,AX
MOV RESULT,CX              ; 结果存入 RESULT
.EXIT
END

```

【评注】注意带符号数的乘法要用 **IMUL**，字节相乘后结果为一个字，所以减 Z 时，16 位的 Z 要扩展为 32 位。

**例 4.7** 试编一个程序将字符串 ‘a’ ~ ‘j’ 转换为大写字母 ‘A’ ~ ‘J’。

【分析】由于小写字母的 ASCII 码比对应大写字母的 ASCII 码大 20H，小写字母可以通过减去 20H 得到相应的大写字母。

```

【解】 .MODEL SMALL
        .DATA
BUF     DB 'abcdefghij'
        .CODE
        .STARTUP
MOV SI,0
MOV CX,10
MOV AL,20H
NEXT:   SUB BUF[SI],AL
        INC SI
        LOOP NEXT
        .EXIT
END

```

**例 4.8** 若在内存中有两串 10 个字节无符号数，将其对应单元内容相减，结果再送回第一个串的位置。

```

【解】 .MODEL SMALL
        .DATA
BUFF1   DB 5,6,7,...; (10 个字节数)
BUFF2   DB 1,2,3,...; (10 个字节数)
        .CODE
        .STARTUP
MOV CX,10 ; 设置循环次数
LEA SI,BUFF1
IEA DI,BUFF2
A1:     MOV AL,[SI]
        SUB AL,[DI]
        MOV [SI],AL

```

```

INC SI
INC DI
LOOP A1
.EXIT
END

```

**例 4.9** 在以 BLOCK 为首地址的内存单元中存放着 15 个字节数（数据可以自己假设），试将其中所有的负数依次存放在以 DEST 开始的存储单元中。

**【解】**

```

.MODEL SMALL
.DATA
BLOCK DB 2,-3,5,-6...; (15 个带符号数)
DEST  DB 15 DUP(?)
.CODE
.STARTUP
MOV BX,OFFSET BLOCK
MOV DI,OFFSET DEST
MOV CX,15
AGAIN: MOV AL,[BX]
        CMP AL,0
        JGE NEXT
        MOV [DI],AL
        INC DI
NEXT:   INC BX
        LOOP AGAIN
        .EXIT
END

```

**例 4.10** 编写程序将包含 32 个数据的数组 ARRAY 分成两个数组：正数组 ARRAYP 和负数组 ARRAYN，并统计它们的长度。

**【解】**

```

.MODEL SMALL
.DATA
ARRAY DB 32 DUP(?)
ARRAYP DB 32 DUP(?)
PLEN  DB ?
ARRAYN DB 32 DUP(?)
NLEN  DB ?
.CODE
.STARTUP
MOV CX,32
MOV SI,OFFSET ARRAY
XOR BX,BX          ; BX 用来对正数计数
XOR DI,DI          ; DI 用来对负数计数
CLD
NEXT: LODSB          ; 从 ARRAY 中读一个数至 AL

```

```

        CMP AL,0
        JGE LOOP1          ; 若是正数或 0, 转 LOOP1 执行
        MOV ARRAYN[DI],AL
                                ; 否则为负数, 存入 ARRAYN 中
        INC DI              ; 对负数计数器加 1
        JMP CONT
LOOP1:  ARRAYP[BX], AL      ; 正数存入 ARRAYP 中
        INC BX              ; 正数计数器加 1
CONT:   LOOP NEXT
        MOV PLEN,BX        ; 存最终统计结果
        MOV NLEN,DI
        .EXIT
END

```

**例 4.11** 计算 $(V-(X \times Y + Z - 540)) / X$ , 结果商、余数分别存放在 R1 和 R2 单元中。  
其中, X、Y、Z、V 均为 16 位带符号数, 已分别存放在 X、Y、Z、V 单元中。

**【解】**

```

.MODEL SMALL
.DATA
X      DW 5000H
Y      DW 3000H
Z      DW 4000H
V      DW 6000H
R1     DW ?
R2     DW ?
.CODE
.STARTUP
MOV AX,X
IMUL Y      ; X×Y, 积在 DX: AX 中
MOV CX,AX
MOV BX,DX   ; 积转移至 BX: CX
MOV AX,Z
CWD         ; 对 Z 进行扩展, 存入 DX: AX
ADD CX,AX
ADC BX,DX   ; X×Y+Z
SUB CX,540
SBB BX,0    ; X×Y+Z-540, 结果在 BX: CX 中
MOV AX,V
CWD         ; 对 V 扩展, 存于 DX: AX
SUB AX,CX
SBB DX,BX   ; V-(X×Y+Z-540), 结果在 DX: AX 中
IDIV X      ; (V-(X×Y+Z-540))/X
MOV R1,AX   ; 存商
MOV R2,DX   ; 存余数
.EXIT

```

END

【评注】 注意带符号数的扩展问题。

**例 4.12** 在内存 AGE 开始的单元中存放了 15 个人的年龄, 编程取出其中最大的年龄, 送入 OLD 单元中。

【解】

```

.MODEL SMALL
.DATA
AGE      DB 12,34,76,...
OLD      DB ?
.CODE
.STARTUP
MOV CX,14      ; 循环次数送 CX
LEA BX,AGE
MOV AL,[BX]
AGAIN:  INC BX
        CMP AL,[BX]
        JAE NEXT
        MOV AL,[BX]      ; AL 中保存较大数
NEXT:   LOOP AGAIN
        MOV OLD,AL      ; 最大年龄送 OLD 单元
        .EXIT
END

```

**例 4.13** 在首地址为 BUF 的内存单元存放了 10 个字节的无符号数, 试编程求其和, 存入 SUM 单元。

【解】

```

.MODEL SMALL
.DATA
BUF      DB 10H,20H,40H,55H,33H
         DB 70H,80H,98H,35H,30H
SUM      DW ?
.CODE
.STARTUP
MOV CX,10
LEA BX,BUF
MOV AX,0
AGAIN:  ADD AL,[BX]
        ADC AH,0
        INC BX
        LOOP AGAIN
        MOV SUM,AX
        .EXIT
END

```

【评注】 在进行多个字节相加时, 要考虑进位问题。本程序用 AL 进行低位的累加,

用 AH 进行高位的累加，同时考虑低位相加时产生的进位。

**例 4.14** 在内存首地址为 TAB 开始的 16 个单元内连续存放着 0~15 的平方值，在内存 X 单元中任给其中的一个数，查表求其平方值，并把结果存入 Y 单元中。试编写程序。

```
【解】 .MODEL SMALL
        .DATA
TAB      DB 0,1,4,9,16,25,36,49,64,81
        DB 100,121,144,169,196,225
X        DB 8
Y        DB ?
        .CODE
        .STARTUP
        LEA BX,TAB
        MOV AL,X
        XLAT
        MOV Y,AL
        .EXIT
        END
```

**【评注】** 注意查表指令 XLAT 的地址隐含条件。

**例 4.15** 用查表指令实现“5”和“8”循环显示。设从“0”~“9”显示的代码表如表 4.2 所示。七段显示器的端口地址为 53H；“5”的代码连续送给显示器 100 次，以后又送“8”的代码 100 次，再循环执行。写出汇编源程序。

表 4.2 显示数字与七段代码对应表

显示字符	0	1	2	3	4	5	6	7	8	9
代码 (H)	3F	06	5B	4F	66	6D	7D	07	7F	6F

```
【解】 .MODEL SMALL
        .DATA
BLOCK    DB 3FH,06,5BH,4FH,66H,6DH,7DH
        DB 07H,7FH,6FH
        .CODE
        .STARTUP
XY:      MOV CX,100
        MOV AL,5
        MOV BX,OFFSET BLOCK
        XLAT
W1:      OUT 53H,AL           ; 显示“5”的代码
        LOOP W1
        MOV CX,100
        MOV AL,8
        MOV BX,OFFSET BLOCK
```



```

                XLAT
W2:             OUT 53H,AL          ; 显示“8”的代码
                LOOP W2
                JMP XY
                .EXIT
            END

```

**例 4.16** 将内存首地址为 BUF1 开始的 10 个带符号字节数,将其按由小到大的次序排列。

**【分析】** 可采用冒泡法来实现排序。

```

【解】      .MODEL SMALL
                .DATA
BUF1           DB 34H,63H,12H...
                .CODE
                .STARTUP
                MOV DX,9
X1:            MOV CX,DX
                MOV SI,0
X2:            MOV AL,BUF1[SI]
                CMP AL,BUF1[SI+1]
                JLE NEXT
                XCHG AL,BUF1[SI+1]
                MOV BUF1[SI],AL
NEXT:          INC SI
                LOOP X2
                DEC DX
                JNZ X1
                .EXIT
            END

```

**【评注】** 本程序用到两重循环,内层循环完成前后两个数两两比较,外层循环控制比较轮数(趟数)。

**例 4.17** 两位十进制数相乘:  $25 \times 80$ ,被乘数和乘数以组合 BCD 码形式存于 DATA1 和 DATA2 两个字节单元中,经乘法运算,积存放在 DATA3 定义的两个内存单元中。

**【分析】** 由于两个组合 BCD 码相乘后,没有相应的乘法调整指令,故只能用对 25 个 80H 相加的方法来实现。

```

【解】      .MODEL SMALL
                .DATA
DATA1          DB 25H
DATA2          DB 80H
DATA3          DW ?
                .CODE
                .STARTUP

```

```

MOV AL,DATA1
MOV BL,AL
AND BL,0FH
AND AL,0F0H
ROR AL,4
MOV BH,0AH
MUL BH
ADD AL,BL      ; 以上程序段将 25H 变为了 25
MOV CL,AL      ; 循环次数 25 送入 CX
MOV CH,0
MOV BL,DATA2
MOV AX,0
CLC
AGAIN:  ADC AL,BL
        DAA          ; 加法调整
        XCHG AH,AL
        ADC AL,0
        DAA
        XCHG AH,AL
        LOOP AGAIN
MOV DATA3,AX
.EXIT
END

```

**【评注】** ①在将 25H 变为 25 时，首先分离 BCD 码的高低位，高位 $\times 10$ +低位即为其对应的十进制数；②多个字节相加时，要考虑进位问题；③使用调整指令时，只能对 AL 中的数进行调整，所以对高位相加后的结果也必须先转移至 AL 才能调整。

**例 4.18** 试使用 .WHILE 和 .ENDIF 语句编程，从键盘接收字符存入内存 BUF 缓冲区，直到出现回车符时停止接收。

**【解】**

```

.MODEL SMALL
.DATA
BUF DB 100 DUP(?)
.CODE
.STARTUP
MOV BX,0
MOV AH,1
INT 21H
.WHILE AL!=0DH
MOV BUF[BX],AL
INC BX
MOV AH,1
INT 21H
.BREAK .IF BX==100

```

```
.ENDW
.EXIT
END
```

**例 4.19** 设有 10 个学生成绩分别为 52、64、83、85、70、89、98、60、100、86，试使用 .IF、.ELSEIF、.ENDIF 语句编程分别统计低于 60 分、60~69 分、70~79 分、80~89 分、90~99 分及 100 分的人数，并存放到 A5、A6、A7、A8、A9、A10 单元中。

**【解】**

```
.MODEL SMALL
.DATA
GRADE DB 52,64,83,85,70
      DB 89,98,60,100,86
A5     DB 0
A6     DB 0
A7     DB 0
A8     DB 0
A9     DB 0
A10    DB 0
.CODE
.STARTUP
MOV CX,10
MOV BX,OFFSET GRADE
NEXT:  MOV AL,[BX]
      .IF AL<60
      INC A5
      .ELSEIF AL<70
      INC A6
      .ELSEIF AL<80
      INC A7
      .ELSEIF AL<90
      INC A8
      .ELSEIF AL<100
      INC A9
      .ELSE
      INC A10
      .ENDIF
      INC BX
      LOOP NEXT
.EXIT
END
```

**例 4.20** 键盘每输入一个字符，若小于“A”，屏幕上显示字符“N”，否则显示字符“C”，直至输入回车结束。试编写程序。

**【分析】** 由于字符显示功能比较独立，可以作为子程序来设计。

**【解】**

```
.MODEL TINY
```

```

        .CODE
        .STARTUP
AGAIN:   MOV AH,1           ; 读输入字符
        INT 21H
        CMP AL,0DH         ; 判断是否回车
        JE DOWN           ; 是, 转 DOWN 退出
        CMP AL,41H         ; 字符与 'A' 比较
        JB NEXT1
        MOV BL,'C'         ; 若大于等于 'A', 则 BL='C'
        JMP NEXT2
NEXT1:   MOV BL,'N'         ; 若小于 'A', 则 BL='N'
NEXT2:   CALL DISPLAY      ; 调用显示子程序
        JMP AGAIN
DOWN:    NOP
        JMP A1
DISPLAY PROC NEAR USES DX,BX,AX
        MOV DL,BL          ; 要显示的字符通过 BL 传递
        MOV AH,2
        INT 21H
        RET
DISPLAY ENDP
A1:NOP
        .EXIT
END

```

## 4.3 习题及全解

### 一、填空题

1. 在汇编语言中标号的类型有\_\_\_\_\_, 变量的类型有\_\_\_\_\_。
2. 在汇编语言中, 一个过程的属性为 NEAR, 表明主程序和子程序\_\_\_\_\_; 若为 FAR, 则表明主程序和子程序\_\_\_\_\_。
3. 下列程序段运行后, X 单元的内容为\_\_\_\_\_。

```

        .MODEL SMALL
        .DATA
X       DW 10
Y       DW 20
Z       DW 100,40,66,80
        .CODE
        .STARTUP
        MOV BX,OFFSET Z

```

```

MOV AX,[BX]
MOV Y,AX
MOV AX,[BX+2]
ADD AX,Y
MOV X,AX
.EXIT
END

```

4. 以下程序段执行后, (AX)=\_\_\_\_\_。

```

...
A DW 124H,345H,128H,255H,512H,127H,678H,789H
B DW 5
...
MOV BX,OFFSET A
MOV SI,B
MOV AX,[BX+SI]
...

```

【解】

1. NEAR 和 FAR; BYTE、WORD、DWORD、PWORD、QWORD
2. 在同一代码段中; 不在同一代码段中
3. 140
4. 5501H

## 二、选择题

1. 完成 41H 送[2100H], 42H 送[2101H]的正确程序段是\_\_\_\_\_。
 

A. MOV AL, 41H	B. MOV SI, 2100H
MOV[2100H], AL	MOV[SI], 'A',
INC 41H	INC SI
MOV[2100H], AL	MOV[SI], 'B'
C. MOV AX, 4241H	D. MOV AX, 'AB'
MOV[2100H], AX	MOV[2100H], AX
2. 在汇编过程中不产生指令码, 只用来指示汇编程序如何汇编的指令是\_\_\_\_\_。
 

A. 汇编指令	B. 机器指令
C. 伪指令	D. 宏指令
3. 在汇编语言程序设计中, 标号和变量的类型属性分别有\_\_\_\_\_。
 

A. BIT、BYTE、WORD、DWORD
B. NEAR、FAR
C. NEAR、FAR 和 BIT、BYTE、WORD、DWORD
D. CODE、DATA 和 STACK、EXTRA

【解】

1. B、C
2. C
3. C

### 三、问答题

1. 什么是指令语句？什么是伪指令语句？它们的主要区别是什么？
2. 在汇编语言源程序中可用于设计简单分支程序的指令有哪些？循环程序结构有哪两类？
3. 汇编语言源程序中变量和标号的区别是什么？各有哪几个类型？
4. 试计算下列伪指令中各变量所分配的字节数。

A1 DW 20

A2 DW 8 DUP(?), 10, 20

A3 DD 10 DUP(?)

A4 DB 3 DUP(?), 4 DUP(0)

A5 DB 'Happy New Year!!'

#### 【解】

1. 指令语句经过汇编以后要产生机器码，而伪指令语句不产生机器码；指令语句和机器的一种操作相对应，而伪指令语句和机器的一种伪操作（汇编过程）相对应。

2. 用于设计分支程序的指令有：**CMP** 目的，源（比较指令）；**TEST** 目的，源（测试指令）；**SHL** 目的，源（移位指令）等。

用于设计循环程序的结构有：

(1) MOV CX,n ; 送循环次数

A1: ...

...循环体

...

LOOP A1

...

(2) MOV CX,n ; 送循环次数

B1: JCXZ A1

...

...循环体

...

JMP B1

A1: ...

3. 变量用来定义数据的逻辑地址，具有 **BIT**、**BYTE**、**WORD**、**DWORD**、**PWORD**、**QWORD** 等类型。标号用来定义指令的逻辑地址，具有 **NEAR** 和 **FAR** 类型。

4. 变量 A1、A2、A3、A4、A5 分配的字节数分别为 2、20、40、15、16。

### 四、读程序题

1. 仔细读下列程序，分析该程序执行完以后，**BLOCK3** 的偏移地址及其对应各字节单元的内容是什么。

```
.MODEL SMALL
```

```

        .DATA
COUNT  DB  4
BLOCK1  DB  96H,74H,32H,10H
BLOCK2  DB  65H,94H,87H,29H
BLOCK3  DB  4 DUP(4)
        .CODE
        .STARTUP
MOV SI,OFFSET BLOCK1
MOV DI,OFFSET BLOCK2
MOV BX,OFFSET BLOCK3
MOV CL,COUNT
MOV CH,0
CLC
NEXT:   MOV AL,[SI]
        INC SI
        ADC AL,[DI]
        INC DI
        DAA
        MOV[BX],AL
        INC BX
        LOOP NEXT
        .EXIT
END
    
```

2. 读下列程序。

```

        .CODE
        .STARTUP
MOV AX,01
MOV BX,02
MOV DX,03
MOV CX,04
AGAIN:  INC AX
        ADD BX,AX
        SHR DX,1
        _____
        .EXIT
END
    
```

若在横线上填入如下指令：

- (1) LOOP AGAIN
- (2) LOOPE AGAIN
- (3) LOOPNE AGAIN

指出程序执行完后 AX、BX、CX、DX 的内容。

3. 在以 ES: SOCl 为起始地址的表中存有字符串，它以 NULL(OOH)作为串结束符，

在 DS: CHAR 中存有关键字。从表中查找第一个此关键字的程序段如下。若找到此关键字,则寄存器 CX 中存放该关键字在 ES 段中的位移量;若串中无此关键字,则 CX 返回-1 值。请将该程序段补充完整。

```

                MOV DI,OFFSET SOC1
                MOV CX,0
CPO:            CMP ES:(1) _____,00H
                JNE COUNT
                MOV BX,CX
                MOV DI,OFFSET CHAR
                MOV AL,[DI]
                MOV DI,OFFSET SOC1
                CLD
                (2) _____SCASB
                JZ FOUND
                MOV CX,(3) _____
                JMP DOWN
FOUND:          (4) _____DI
                MOV CX,DI
                JMP DOWN
COUNT:         INC DI
                INC CX
                JMP CPO
DOWN:           HLT

```

4. 请阅读如下程序,指出其完成的功能。

```

                .MODEL SMALL
                .DATA
TAB DB 30H,31H,.....39H,41H,42H,.....46H
DISP DB 16 DUP(?)
                .CODE
                .STARTUP
                MOV CX,4
                MOV BX,OFFSET TAB
                MOV DI,OFFSET DISP
                MOV AL,0
HT:            SHL DX,1
                RCL AL,1
                SHL DX,1
                RCL AL,1
                SHL DX,1
                RCL AL,1
                SHL DX,1
                RCL AL,1
                SHL DX,1
                RCL AL,1

```



```

AND AL,0FH
XLAT
MOV[DI],AL
INC DI
LOOP HT
.EXIT
END

```

5. 下面的程序是将 10 个 8 位的无符号数按递减顺序排序。请选择正确的答案将程序填充完整。

```

.MODEL SMALL
.DATA
ARRAY DB 05H,78H,FFH,7BH,00H
       DB 8CH,20H,A0H,F0H,60H
.CODE
.STARTUP
MOV SI,OFFSET ARRAY
MOV BL,9
AB1:   MOV CX,BL
       (1) _____
AGAIN: MOV AL,[SI]
       INC SI
       CMP AL,[SI]
       (2) _____ CD1
       MOV AH,[SI]
       MOV[SI],AL
       DEC SI
       MOV[SI],AH
       INC SI
CD1:   LOOP AGAIN
       DEC BL
       (3) _____ AB1
       .EXIT
END

```

6. 读下面程序,指出程序完成的功能。

```

.MODEL SMALL
.DATA
FIRST DB 0BH,8AH,0H
SECOND DB 05H,0D7H
.CODE
.STARTUP
MOV CX,2

```

```

        MOV SI,0
        CLC
NEXT:   MOV AL,SECOND[SI]
        ADC FIRST[SI],AL
        INC SI
        LOOP NEXT
        MOV AL,0
        ADC AL,0
        MOV FIRST[SI],AL
        .EXIT
        END

```

7. 指出下列子程序完成的功能。

```

...
CHS:   PUSH AX
        PUSH DX
        MOV DX,390H
        IN AL,DX
        AND AL,0FH
        CMP AL,09H
        JG ATOF
        ADD AL,30H
        JMP SEND
ATOF:  ADD AL,37H
SEND:  OUT DX,AL
        POP DX
        POP AX
        RET

```

8. 下列程序段实现从键盘输入一系列字符，并以“\$”为结束符，然后对其中的非数字字符进行计数，并显示结果。程序中有若干错误，试改正。

```

...
BUFF  DB 50 DUP(?)
COUNT DW 0
...
        MOV BX,BUFF
        MOV COUNT,0
INPUT: MOV AH,01
        INT 21H
        MOV[BX],AL
        INC BX
        CMP AL,'$'
        JZ INPUT

```

```

        LEA BX,BUF
NEXT:   MOV CL,[BX]
        INC BX
        CMP CL,'$'
        JZ DISP
        CMP CL,30H
        JG CONT
        CMP CL,39H
        JB NEXT
CONT:   INC COUNT
        JMP NEXT
DISP:   ...

```

**【解】**

1. BLOCK3 的偏移地址为 0009H  
(0009H)=61H    (000AH)=69H  
(000BH)=20H    (000CH)=40H
2. (1) (AX)=05H    (BX)=10H    (CX)=00H    (DX)=00H  
(2) (AX)=02H    (BX)=04H    (CX)=03H    (DX)=01H  
(3) (AX)=03H    (BX)=07H    (CX)=02H    (DX)=00H
3. (1) DI    (2) REPNE    (3) -1 或 0FFFFH    (4) DEC
4. 它把 DX 中的 4 位十六进制值相应转换成 4 个 ASCII 字符, 从高位开始顺序存放于 DISP 中。
5. (1) DEC CX    (2) JAE    (3) JNZ
6. 这是一个多字节加法程序, 第一个数是 8A0BH, 第二个数是 D705H, 结果和进位存放在 FIRST 中。
7. 将端口 390H 输入的十六进制数的低 4 位转换成 ASCII 码再输出。
8. 共有 4 处错误。  
MOV BX, BUFF 改为 LEA BX, BUFF  
JZ INPUT 改为 JNZ INPUT  
JG CONT 改为 JB CONT  
JB NEXT 改为 JBE NEXT

**五、编程题**

1. 在内存 XYZ 中开始存放了 10 个带符号字节数, 判断它们中有多少个正数、多少个负数, 正数的个数放在 DH 中、负数的个数放在 DL 中。写出简化的汇编源程序。
2. 试编写查表程序, 将 BUF 中任意存放的 16 个 00H~0FH 之间的数转换为对应的表示十六进制的 ASCII 码, 并显示。
3. 试使用 REPEAT 和 UNTILCXZ 语句编程, 在首地址为 BUF 的缓冲区填入 100 个 00H。

4. 试使用 `.IF`、`.ELSE` 和 `.ENDIF` 语句编程, 将存于 `AL` 寄存器的 `00H~0FH` 之间的二进制数转换为 `ASCII` 码。

5. 设在数据段 `DATA`、附加段 `EXTRA` 和堆栈段 `STACK` 中分别定义了字变量 `X`、`Y` 和 `Z`, 试用完整段定义结构编写程序计算:  $Z \leftarrow X+Y+Z$ 。

6. 某班 30 个学生的汇编语言考试成绩存放在 `MARK` 开始的单元中, 试编程将其最高分和最低分取出, 分别送 `HIGH` 和 `LOW` 单元, 并且显示输出。

7. 试编程序完成两个 32 位带符号数相乘。

8. 编程序将数据串 `(-10, 0, 23, 12, -8, 9)` 中最大数和最小数找出来, 存入 `MAX` 和 `MIN` 中, 并计算数据串中各数据项绝对值之和。

9. 编写程序, 将字符串 `STRING` 中的 “&” 字符用空格符代替。字符串 `STRING` 为: “The day is cloud&rain.”。

10. 编制一个查表程序将十六进制数 `2A49H` 转换成 `ASCII` 码, 结果依次存入 `MEM` 数组的 4 个字节中, 即程序执行后 `MEM` 中 4 个字节内容成为: `39H`、`34H`、`41H`、`32H`。

11. 求分别放在内存单元 `X` 和 `Y` 中的两个数的平均值, 并把结果存放在 `Z` 单元中。

12. 编程实现把 `BX` 寄存器中的二进制数用十六进制的形式在屏幕上显示出来。

13. 在内存 `STRING` 单元中开始存放一串字符, 用回车符 (`ODH`) 作为结束标志。编程统计字符串的个数, 且把它送入 `NUM` 单元。

14. 在内存 `BLOCK` 单元中开始存放 10 个无符号数, 从中找出最大值与最小值分别送入 `MAX` 与 `MIN` 单元中。

15. 在内存 `BLOCK` 开始的连续 3 个单元中有 3 个带符号数 (字节) `a`、`b`、`c`, 写出计算  $a-b \times c$  的完整汇编源程序。

16. 若在内存同一段内有两串 10 个字节数, 将其对应单元内容相减, 结果存入目的串中, 写出简化汇编源程序。

17. 若在内存 `BLOCK1` 单元中开始存放着 10 个不带符号字节数, 在不改变原数存放位置的情况下, 从中找出最大数, 并把它存放在紧靠 10 个字节数之后的单元内。写出完整的段定义汇编语言源程序。

#### 【解】

```
1.      .MODEL SMALL
        .DATA
        XYZ DB -1, 2, -6, .....
        .CODE
        .STARTUP
        MOV BX, OFFSET XYZ      ; XYZ 的地址装入 BX
        MOV CX, 10              ; 循环次数送 CX
        MOV DX, 0               ; 用来计数的寄存器 DX 清 0
AGAIN:  CMP BYTE PTR[BX], 0     ; 数据和 0 进行比较
        JGE NEXT
        INC DL                  ; 为负数时, DL 加 1
        JMP X
NEXT:   INC DH                  ; 为正数时, DH 加 1
```

```

X:   INC BX
      LOOP AGAIN
      .EXIT
      END

2.   .MODEL SMALL
      .DATA
TABLE DB '0123456789ABCDEF'
      BUF DB 00H, 04H, 0EH, ..... ; (任意存放的 16 个 00H~0FH 之间的数)
      .CODE
      .STARTUP
      MOV CX,16
      MOV BX,OFFSET TABLE
      MOV SI,OFFSET BUF
AGAIN: MOV AL,[SI]
      XLAT
      MOV[SI],AL
      MOV DL,AL
      MOV AH,2
      INT 21H
      INC SI
      LOOP AGAIN
      .EXIT
      END

3.   .MODEL SMALL
      .DATA
      BUF DB 100 DUP(?)
      .CODE
      .STARTUP
      LEA BX,BUF
      MOV CX,100
      .REPEAT
      MOV BYTE PTR[BX], 0
      INC BX
      .UNTILCXZ
      .EXIT
      END

4.   .MODEI,SMALL
      .DATA
      BUF DB 00H,01H,02H, .....
      .CODE
      .STARTUP
      MOV BX,OFFSET BUF

```

```

        MOV CX,16
AGAIN:  MOV AL,[BX]
        .IF AL>=00H & AL<=09H
        ADD AL,30H
        .ELSE
        ADD AL,37H
        .ENDIF
        MOV DI,AL
        MOV AH,2
        INT 21H
        INC BX
        LOOP AGAIN
        .EXIT
        END

5. DATA SEGMENT
        X    DW 200
DATA    ENDS
EXTRA   SEGMENT
        Y    DW 300
EXTRA   ENDS
STACK   SEGMENT
SBUF    DW 1000 DUP(?)
        TOP   LABEL WORD
        Z    DW 100
STACK   ENDS
CODE    SEGMENT
        ASSUME CS: CODE,DS: DATA
        ASSUME ES: EXTRA,SS: STACK
START   MOV AX,DATA
        MOV DS,AX
        MOV AX,EXTRA
        MOV ES,AX
        MOV AX,STACK
        MOV SS,AX
        MOV SP,OFFSET TOP
        MOV AX,ES: Y
        ADD AX,X
        ADD AX,SS: Z
        MOV AH,4CH
        INT 21H
CODE    ENDS
        END START

6.      .MODEL SMALL
        .DATA

```

```

MARK      DB 60H,90H,80H,75H,……; (30 个分数的 BCD 码)
HIGH      DB ?
LOW       DB ?
          .CODE
          .STARTUP
          CALL SEARCH
          MOV BL,HIGH
          CALL DISPLAY
          MOV BL,LOW
          CALL DISPLAY

SEARCH    PROC
          MOV CX,29
          MOV AL,MARK
          MOV DL,MARK
          MOV BX,OFFSET MARK
AGAIN:    INC BX
          CMP AL,[BX]
          JAE NEXT1
          MOV AL,[BX]
NEXT1:    CMP DL,[BX]
          JBE NEXT2
          MOV DL,[BX]
NEXT2:    LOOP AGAIN
          MOV HIGH,AL
          MOV LOW,DL
          RET
SEARCH    ENDP
DISPLAY   PROC
          MOV DL,20H
          MOV AH,2
          INT 21H
          MOV CX,2
LP:       ROR BL,4
          MOV AL,BL
          AND AL,0FH
          OR AL,30H
          MOV DL,AL
          MOV AH,2
          INT 21H
          LOOP LP
          RET
DISPLAY   ENDP
          .EXIT
          END

7.        .MODEL SMALL
    
```

```

        .586
        .DATA
X      DD 00015002H
Y      DD 00042370H
        .CODE
        .STARTUP
        MOV EAX,X
        IMUL Y
        .EXIT
        END

8.      .MODEL SMALL
        .DATA
ARY    DB-10,0,23,12,-8,9
MAX    DB ?
MIN    DB ?
SUM    DW ?
        .CODE
        .STARTUP
        MOV AL,ARY
        MOV MAX,AL
        MOV MIN,AL
        MOV DX,0
        CLD
        MOV SI,OFFSET ARY
        MOV CX,6
L1:     LODSB
        CMP AL,MAX
        JNG LAB
        MOV MAX,AL
LAB:    CMP AL,MIN
        JNL LNL
        MOV MIN,AL
LNL:    AND AL,AL
        JNS LP
        NEG AL
LP:     ADD DL,AL
        ADC DH,0
        LOOP L1
        MOV SUM,DX
        .EXIT
        END

9.      .MODEL SMALL
        .DATA
STRING DB 'The day is cloud &rain. '

```



```

        .CODE
        .STARTUP
        MOV CX,24
        MOV SI,OFFSET STRING
NEXT:   MOV AL,[SI]
        CMP AL,'&'
        JNZ CONT
        MOV[SI],20H
CONT:   INC SI
        LOOP NEXT
        .EXIT
        END
    
```

```

10.     .MODEL SMALL
        .DATA
        MEM DB 4 DUP(?)
        TBL DB'0123456789ABCDEF'
        .CODE
        .STARTUP
        MOV AX, 2A49H
        MOV BX, OFFSET TBL
        MOV DX, AX
        MOV CX, 4
        MOV DI, 0
NEXT:   MOV AL, DL
        AND AL, 0FH
        XLAT
        MOV MEM[DI],AL
        PUSH CX
        MOV CL,4
        SHR DX,CL
        POP CX
        INC DI
        LOOP NEXT
        .EXIT
        END
    
```

```

11.     .MODEL SMALL
        .DATA
        X DB 6AH
        Y DB 8EH
        Z DB ?
        .CODE
        .STARTUP
        MOV AL,X
    
```

```

        ADD AL,Y
        MOV AH,0
        ADC AH,0
        MOV BL,2
        DIV BL
        MOV Z,AL
        .EXIT
    END

12.      .MODEL TINY
        .CODE
        .STARTUP
        MOV CH,4
AGAIN:    MOV CL,4
        ROL BX,CL
        MOV AL,BL
        AND AL,0FH
        OR AL,30H
        CMP AL,3AH
        JB NEXT
        ADD AL,07H
NEXT:    MOV DL,AL
        MOV AH,2
        INT 21H
        DEL CH
        JNZ AGAIN
        .EXIT
    END

13.      .MODEL SMALL
        .DATA
STRING    DB 'ABCDEFGH',0DH
NUM       DB ?
        .CODE
        .STARTUP
        LEA DI,STRING
        MOV DL,0
        MOV AL,0DH
        CLD
AGAIN:    SCASB
        JE DONE
        INC DL
        JMP AGAIN
DONE:    LEA BX,NUM
        MOV[BX],DL

```

```

        .EXIT
        END

14.      .MODEL SMALL
        .DATA
BLOCK    DB 60,90,0,50,.....
        MAX    DB ?
        MIN    DB ?
        .CODE
        .STARTUP
        MOV CX,9                ; 循环次数送 CX
        MOV AL,BLOCK
        MOV DL,BLOCK
        MOV BX,OFFSET BLOCK
AGAIN:   INC BX
        CMP AL,[BX]
        JAE NEXT1
        MOV AL,[BX]            ; AL 中存较大数
NEXT1:   CMP DL,[BX]
        JBE NEXT2
        MOV DL,[BX]            ; DL 中存较小数
NEXT2:   LOOP AGAIN
        MOV MAX,AL
        MOV MIN,DL
        .EXIT
        END

15. DATA SEGMENT
BLOCK    DB 3 DUP(?)
SULT     DW?
DATA     ENDS
CODE     SEGMENT
        ASSUME DS: DATA,CS: CODE
START:   MOV AX,DATA
        MOV DS,AX
        LEA SI,BLOCK
        MOV AL,[SI+1]
        MOV BL,[SI+2]
        IMUL BL
        MOV DX,AX
        MOV AL,[SI]
        CBW
        SUB AX,DX
        MOV SULT,AX
        MOV AH,4CH
    
```

```

        INT 21H
CODE    ENDS
        END START

16.     .MODEL SMALL
        .DATA
SRC      DB 20H,33H,.....
DEST     DB 10H,23H,.....
        .CODE
        .STARTUP
MOV  CX,10                ; 循环 10 次
MOV  AX,@DATA
MOV  ES,AX                ; 使附加段和数据段重叠
LEA  SI, SRC
LEA  DI, DEST
CLD                      ; 地址按增量修改
W:     LODSB
        SUB AL,[DI]
        STOSB
        LOOP W
        .EXIT
        END

17. DATA SEGMENT
BLOCK1  DB 10 DUP(?)
MAX      DB?
DATA     ENDS
CODE     SEGMENT
        ASSUME CS:CODE,DS:DATA
START    MOV AX,DATA
        MOV DS,AX
        MOV BX,OFFSET BLOCK1
        MOV CX,9
        MOV AL,[BX]
AGAIN:   CMP AL,[BX+1]
        JA NEXT
        MOV AL,[BX+1]      ; AL 中保存较大数
NEXT     INC BX
        LOOP AGAIN
        MOV MAX,AL        ; 最大数送入 MAX 单元中
        MOV AH,4CH
        INT 21H
CODE     ENDS
        END START

```

## 5.1 重点内容提要

### （一）存储器概述

#### 1. 存储器的类型

微型计算机系统存储器可分为两大类，即内存和外存。内存也称为主存储器，可通过系统总线直接与微处理器联系，用来存放正在执行的程序和正在处理的数据；外存也称辅助存储器，需通过专门的接口电路和微处理器联系，用来存放暂时不执行或不被处理的数据。

内存主要由半导体存储器组成，其类型如图 5.1 所示。

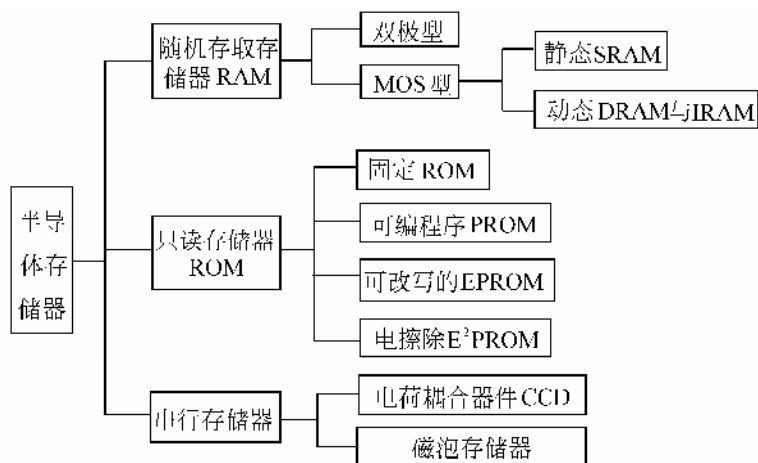


图 5.1 半导体存储器的分类

外存主要有磁盘、磁带和光盘。

#### 2. 存储器多级结构

在计算机的发展过程中，CPU 与主存储器速度匹配的矛盾越来越突出。为此，在 CPU 与主存之间再增加一级或多级高速缓冲存储器（Cache）来提高存储器的性能价格比。目前，微型计算机的存储器系统已经发展成多级结构，如图 5.2 所示。

#### 3. 32 位内存储器的组成方式

80486 微处理器分为 4 个存储体，依次存放 32 位数据的不同字节。每个存储体的 8 位数据线依次并行连接到系统数据线  $D_{31} \sim D_0$  上。每个存储体的 15 位地址  $A_{14} \sim A_0$  接

CPU 的地址线  $A_{16} \sim A_2$ , 片选信号由高位地址  $A_{31} \sim A_{17}$  的译码结果和  $\overline{BE_3} \sim \overline{BE_0}$  相“与”后产生。一旦地址确定,  $A_{31} \sim A_1$  将确定 4 个存储体中的相同地址单元,  $\overline{BE_3} \sim \overline{BE_0}$  决定某一个或某几个字节单元选中, 然后可对选中单元同时进行读/写操作。

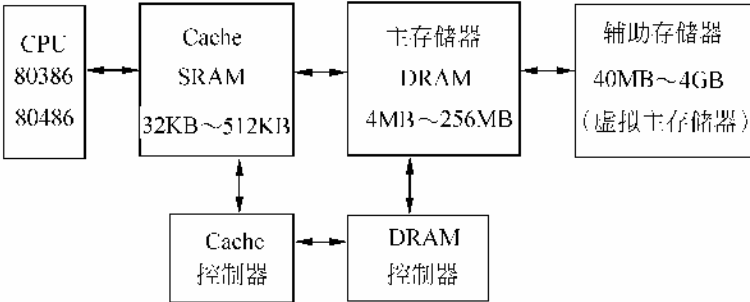


图 5.2 存储器多级结构

## (二) 内存储器的管理

32 位机采用分段和分页技术进行存储器管理。每段可高达 4GB, 每页为 4KB。

### 1. 分段管理

根据程序中指令与数据之间或各子程序之间的逻辑独立性, 可将程序分为若干个段。每个段的管理信息由段描述符来描述。

描述符按段的性质可分为 3 类: 程序段描述符、系统段描述符和门描述符, 如图 5.3 所示。

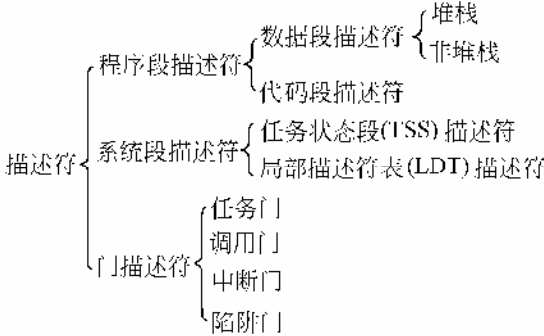


图 5.3 描述符的分类

段的描述符存储于被称为描述符表的专门定义的存储区内, 描述符表有 3 种: 全局描述符表 GDT、局部描述符表 LDT 和中断描述符表 IDT。

存储器的分段管理就是: 程序中使用的是逻辑地址, 包括 16 位段选择符和 32 位偏移地址 (由寻址方式决定) 两部分。任何信息都定义在某一段中, 通过段选择符, 可以找到它对应的段描述符, 从段描述符中可以取出该段的 32 位段基址、段的长度和关于该段的其他信息。用 32 位的段基址加上 32 位的偏移地址得到线性地址。如果分页部件被

禁止,那么线性地址就是物理地址。

## 2. 分页管理

分页管理实质是把线性地址空间和物理地址空间都看成由页组成,且线性地址空间中的任何一页均可映射到物理地址空间中的任何一页。页是使用存储器中固定大小的块,32位机中规定一页是4KB,并且它的起始地址总是安排在低12位地址为0处,即能被1000H整除的地址处。分页管理将把4GB的线性地址空间划分为 $2^{20}$ 个页面,并通过把线性地址空间的页重新定位到物理地址空间来管理。

分页与分段的主要不同是,一个段的长度可变,最大为4GB,而页的长度是固定的,每页4KB。

对于32位机,用户在虚拟空间中编程,虚拟空间为64TB,通过内存管理机制可以把编程的逻辑地址映射到4GB的物理地址空间。32位CPU还对内存中的程序设定了段限保护和特权保护。

## 3. 地址变换

地址变换过程是指将逻辑地址转换成物理地址的过程。地址变换过程如图5.4所示。

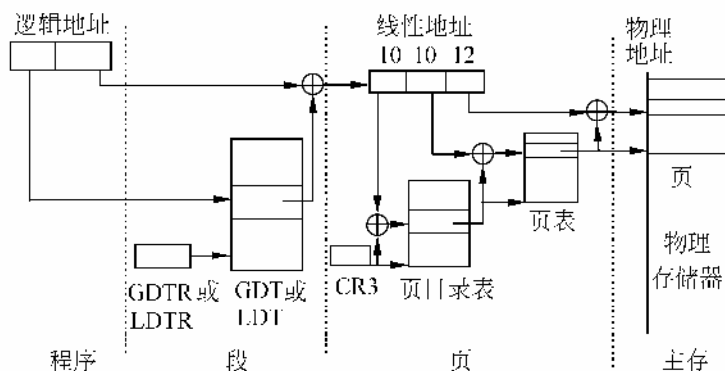


图 5.4 地址变换过程

## 5.2 常考题型范例精解

**例 5.1** 内存和外存有何区别? 内存中 RAM 和 ROM 有何区别?

**【解】** 内存用来存放正在执行的程序和处理的数据,CPU可直接访问它;而外存用来存放暂不执行的程序和不被处理的程序,CPU访问它时,必须通过接口电路把程序或数据成批地调入内存再进行处理。RAM是随机存取存储器,其内容可随机读出或写入;ROM是只读存储器,其内容只能读出,不能随机写入。

**例 5.2** 8086 对内存最大寻址范围是多大? 80486 对内存最大寻址范围是多大? 什么是虚拟存储器? 32 位 CPU 可寻址的最大虚拟空间是多大?

**【解】** 8086 的地址线为 20 位,所以对内存最大寻址范围是 1MB ( $2^{20}\text{B}$ ),80486 是 32 位的地址线,所以其内存最大寻址范围是 4GB ( $2^{32}\text{B}$ )。虚拟存储器是将外存的一部

分当做内存来使用,虚拟空间实际上是由外存和内存有机结合提供的。用户在虚拟空间里编程,通过地址变换将逻辑地址映射到内存物理地址。32 位 CPU 可寻址的最大虚拟空间是 64TB。

**例 5.3** 80486 CPU 保护模式下逻辑地址和实模式下的逻辑地址有何不同?

**【解】** 不同之处有:①保护模式下段基址和偏移地址都是 32 位的,而实模式下的都是 16 位的。②保护模式下的段基址包含在段描述符中。每个段描述符由 8 个字节组成,其中 32 位是本段的段基址,段的描述符存储于被称为描述符表的专门定义的存储区内。而在实模式下,段基址直接由 16 位的段寄存器提供。③在保护模式下段寄存器可以是 CS、SS、DS、ES、GS、FS。16 位编程可见的段寄存器称为选择寄存器,内部还对应有编程不可见的 64 位的描述符寄存器。其中选择寄存器的低 2 位是特权标志,  $D_2$  位是描述符表类型标志,高 13 位是选择码,指出本段的段描述符在由  $D_2$  指出的描述符表中的逻辑排序,依据此可以找到在描述符表中对应的段描述符,进而在段描述符中获得段基址。而在实模式下段寄存器只定义有 CS、SS、DS、ES。它直接存放内存段的段基址,这里段基址是指段首地址的高 16 位。

**例 5.4** 简述 32 位微机的地址变换过程。

**【解】** 32 位程序中,存储器地址往往是以逻辑地址的形式给出的,即一个段寄存器和一个由寻址方式确定的 32 位偏移量。在段寄存器中存放的是选择符,以其高 13 位选择段描述表中的某一项,送段描述符寄存器,同时将段描述符中 32 位的段基址与逻辑地址中的偏移量相加,产生线性地址。这时如果分页禁止,那么产生的线性地址就是存储器的物理地址。如果分页,首先根据线性地址的高 20 位查页描述符高速缓冲存储器 TLB,以获取物理地址,若不能命中,则由分页部件再将线性地址转换为物理地址。

## 5.3 习题及全解

### 一、填空题

1. 为实现在保护方式下工作,32 位 CPU 设置了 3 个描述符表,分别是\_\_\_\_\_、\_\_\_\_\_和\_\_\_\_\_。
2. 80486 直接可以访问的内存空间是\_\_\_\_B,它的一段最大空间是\_\_\_\_B,一页空间是\_\_\_\_B。

**【解】**

1. 全局描述符表 GDT; 局部描述符表 LDT; 中断描述符表
2. 4G; 4G; 4K

### 二、选择题

1. 和外存相比,内存的特点是\_\_\_\_\_。  
A. 容量小、速度快、成本高      B. 容量小、速度快、成本低  
C. 容量大、速度快、成本高      D. 容量大、速度快、成本低



2. 某计算机的内存为 3KB, 则内存地址寄存器只需\_\_\_\_位就足够了。

- A. 10                      B. 11                      C. 12                      D. 13

3. 计算机的内存可采用\_\_\_\_\_。

- A. RAM 和 ROM                      B. RAM  
C. ROM                      D. 磁盘

4. 采用虚拟存储器的目的是\_\_\_\_\_。

- A. 提高主存的速度                      B. 扩大外存的容量  
C. 扩大编程空间                      D. 提高外存的速度

【解】

1. A    2. C    3. A    4. C

### 三、问答题

1. 计算机的主存与辅存(即内存与外存)是以什么原则划分的? 80486 CPU 直接访问的内存空间(不考虑虚拟空间)最大是多少?

2. 从内存读取一个数据, 8086 系统的段基址由什么提供? 它是多少位的? 而 80486 系统的段基址由什么提供? 它是多少位的?

3. 80486 在 3 种工作方式下, “段”的长度有何差别?

4. 有一个描述符的选择字, 其内容为 1000011110000111, 请解释其含义。

5. 门描述符和段描述符有何异同?

【解】

1. 外存和内存是以 CPU 能否直接访问这一原则来划分的。80486 CPU 能直接访问的最大内存空间为 4GB。

2. 8086 系统的段基址是由段寄存器 DS 提供的, 它是 16 位寄存器。80486 系统的段基址是由段描述符提供的, 它是 32 位的。

3. 80486 在实模式与虚拟 8086 模式下段的长度为 64KB, 在保护模式下段的最大长度为 4GB。

4. 选择字的低 2 位是特权标志,  $D_2$  位是描述符表类型标志, 高 13 位是选择码, 该选择码指出本段的段描述符在由  $D_2$  指出的描述符表中的逻辑排序。根据选择字的含义, “1000011110000111”表示: 选择字的请求特权级为 3; 要访问的描述符在局部描述符表 LDT 中; 选择码为  $1 \times 16^3 + 15 \times 16 = 4336$ , 表示选择字所访问的描述符在描述符表中的 4336 表项中。

5. 段描述符和门描述符的主要区别是: 在 8 个字节的描述符格式中, 段描述符有 32 位的段基址, 门描述符中有 16 位的指向其他描述符的选择符。段描述符描述段的特征, 为访问段内信息提供了基础, 门描述符用于间接地控制转移和任务切换。

## 第 6 章

## 微型计算机的输入/输出

### 6.1 重点内容提要

#### （一）接口及端口

CPU 与外设之间不能直接进行通信，必须要借助于接口。这是因为 CPU 与外设之间速度不匹配或信息格式不同等原因。

对于 I/O 接口，在其内部往往有多个可寻址读写的寄存器，我们称之为端口。主机和外设之间交换信息都是通过操作接口电路的 I/O 端口来实现的。

CPU、I/O 接口和外设之间的接口信息如图 6.1 所示。

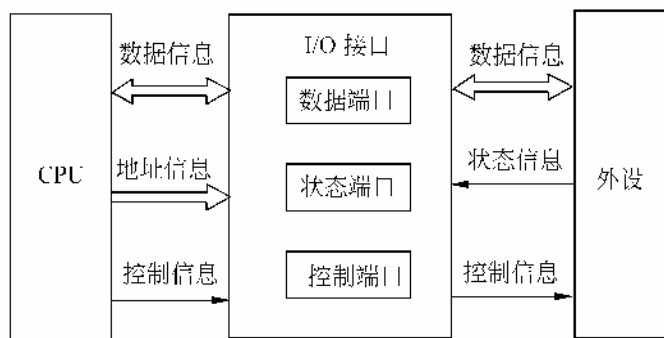


图 6.1 CPU、I/O 接口和外设之间的接口信息

#### （二）I/O 端口的编址

CPU 对 I/O 端口的编址分为存储器映象的 I/O 寻址和 I/O 映象的 I/O 寻址两种方式。

存储器映象的 I/O 寻址是把 I/O 端口的地址和存储单元的地址统一编址；I/O 映象的 I/O 寻址是将两者分开独立编址。所谓的“独立”和“统一”是相对于存储器地址空间而言的。独立编址时，I/O 地址空间完全独立于存储器空间，于是一个 I/O 端口的地址可以与一个存储单元的地址重叠，CPU 通过控制总线来确定要访问哪一个空间。

存储器映象的 I/O 寻址的优点是指令丰富、编程方便，但程序易读性差、存储器地址范围相对减小。I/O 映象的 I/O 寻址需要用专门的 I/O 指令，但程序方便阅读。80x86 采用的是 I/O 映象的 I/O 寻址。

### (三) I/O 端口地址的形成

当系统中使用存储器映象的 I/O 寻址方式时,系统中仅需一个译码器芯片,从译码器的输出端接至 I/O 接口芯片的控制端或片选端形成 I/O 端口地址,接至存储器芯片的片选端形成存储单元地址。注意,CPU 的  $\overline{M/\overline{IO}}$  线接在译码器的控制输入端时,要接在高电平有效的那一端,因为在执行存储器访问指令时,CPU 的  $\overline{M/\overline{IO}}$  管脚输出高电平有效信号。

当系统中使用 I/O 映象的 I/O 寻址方式时,系统中的 I/O 端口地址需要单独的一个译码器芯片,译码器的输出端只允许接至 I/O 接口芯片的控制端或片选端形成 I/O 端口地址。注意,CPU 的  $\overline{M/\overline{IO}}$  线接在译码器的控制输入端时,要接在低电平有效的那一端,这是因为在执行 IN 或 OUT 指令时,CPU 的  $\overline{M/\overline{IO}}$  管脚输出低电平有效信号。

### (四) 输入/输出方式

CPU 与外设通信时,数据传送的控制方式有程序控制传送方式、中断控制传送方式、直接存储器存取方式(DMA 方式)和 I/O 处理机方式。

程序控制传送方式分为同步传送方式和异步查询方式。同步传送方式只有在外设的各种操作时间已知且固定的场合下才能使用。异步查询方式在传送前,必须先查询一下外设的状态。当外设准备好了,CPU 就立即与外设进行数据传送,否则,CPU 就处于循环查询状态,直到 CPU 查询到外设准备好了才进行传送。这种方法能较好地解决外设与 CPU 速度上的差异,但 CPU 必须不断地循环测试外设的状态,这样,大大降低了 CPU 的运行效率。

在中断传送方式下,CPU 启动外设后,不用等待查询状态,CPU 继续执行自己的程序。当外设准备好了时,由外设向 CPU 发出一个中断请求。若 CPU 响应它,则去执行中断服务程序,待中断服务程序执行完后,再返回主程序。中断方式实现了 CPU 和外设的重叠工作,从而使 CPU 的效率得到提高。

若有大批量的数据要传送,采用 DMA 方式比较合适。DMA 方式是用一个硬件 DMAC 芯片来完成软件的工作。CPU 把总线使用权交给 DMAC,然后由 DMAC 快速地将大批量数据与内存进行信息交换。在这段时间里,CPU 可以处理一些内部工作。

要使 CPU 完全摆脱控制 I/O 设备的负担,则提出了 I/O 处理机方式。由 I/O 处理机完成一切 I/O 处理的工作,它有自己的指令系统,可以执行程序来实现对数据的处理。

### (五) DMA 控制器 8237A

DMA 控制器可以像 CPU 那样得到总线控制权,用 DMA 方式实现外部设备和存储器之间的数据高速传输。8237A 是高性能的可编程 DMA 控制器,它一方面可以控制系统总线,这时通常称它为总线主模块;另一方面又可以和其他接口一样,接受 CPU 对它的读/写操作,这时它就成了总线从模块。

(1) 理解 8237A 是如何向 CPU 申请总线控制权的,在控制总线期间完成什么工作。掌握 8237A 作为总线主模块(控制系统总线)和总线从模块(系统总线由 CPU 控制)

时, 8237A 的某些引脚的不同含义。其中  $\overline{\text{IOR}}$ 、 $\overline{\text{IOW}}$ 、 $A_3 \sim A_0$ 、 $DB_7 \sim DB_0$  等在 8237A 作为主模块和从模块时的定义不同, 甚至传送方向也不同。例如, 8237A 作为从模块时,  $\overline{\text{IOW}}$  是送入 DMA 控制器的, 此信号有效, 由 CPU 往 DMA 控制器的内部寄存器写入信息, 即进行编程; 在 8237A 作为主模块时,  $\overline{\text{IOW}}$  的方向是由 DMA 控制器送出的, 此信号有效时, 存储器中读出的数据被写入 I/O 接口中。8237A 作为从模块时,  $A_3 \sim A_0$  是输入端, 接受 CPU 发出的地址信息, 对 8237A 片内的 16 个端口进行寻址; 在 8237A 作为主模块时,  $A_3 \sim A_0$  是输出端, 是 8237A 输出的寻址存储器单元的地址信息的最低 4 位。

(2) 在 8237A 与 CPU 之间的连线中, 注意 ADSTB 和 AEN 信号的连接。简单地说, ADSTB 用于外部地址锁存器的锁存端相连, 而 AEN 则与锁存器的允许输出相连, 同时 AEN 信号还应控制 CPU 的地址锁存器的允许输出。当 8237A 作为主模块时, AEN 有效应允许 8237A 的地址锁存器有效输出, 而禁止 CPU 的地址锁存器输出。

(3) 8237A 仅提供 16 位地址线, 因此在 8086/8088 计算机系统中, 为能寻址 1MB 存储器空间, 最高 4 位地址  $A_{19} \sim A_{16}$  需在 DMA 传输之前, 要用指令将高 4 位地址送到一个 4 位的 I/O 端口中, 或采用页面寄存器 74LS670 来实现。

(4) 8237A 具有两种优先级管理方式: 固定优先级和循环优先级。但无论采用哪种优先级管理方式, 一旦某通道的请求获得服务之后, 即占有总线控制权后, 其他的通道请求均被禁止, 直到该通道的服务结束。

注意: 8237A 的优先管理不同于中断优先级的管理。

(5) 8237A 的 4 种工作模式, 除级联传输模式外, 其他 3 种模式的不同点都体现在 DMA 控制器占有总线后释放总线条件的不同上, 基于这一点认识, 便不难为不同的应用选择合适的工作模式。

8237A 具有 16 个端口地址, 学习时注意各端口的类型及信息格式。

## 6.2 常考题型范例精解

**例 6.1** I/O 端口有哪两种寻址方式? 它们各自的主要优点是什么?

**【解】** I/O 端口有存储器映象的 I/O 寻址方式和 I/O 映象的 I/O 寻址方式。前一种的主要优点是不需专门设置 I/O 指令, 对 I/O 的操作功能强; 后一种的主要优点是专门的 I/O 指令, 执行快、且不影响整个存储器空间。

**例 6.2** CPU 与外设传递的信息有哪几种? 为什么必须通过接口电路?

**【解】** CPU 与外设传递信息有数据、状态和控制 3 种信息。传递信息必须通过接口电路, 因为外设提供的信息种类多, 有数字信息 (数字信息还有串行的和并行的)、模拟信息、脉冲信息、开关信息等, 都要通过一些接口电路转换成计算机能处理的信息形式; 其次是外设工作速度慢, 必须通过接口电路实现与 CPU 速度匹配。另外还有一些如电平转换等功能。

**例 6.3** CPU 与外设通信时, 其控制方式有哪几种? 各有何特点?

**【解】** CPU 与外设通信时, 其控制方式有程序控制方式、中断控制方式、DMA 方式和 I/O 处理机方式。程序控制方式又分为同步传送方式和异步查询方式, 在此方式下,

其状态和数据的传送是由 CPU 执行一系列指令来完成的, CPU 效率低。中断控制方式实现了 CPU 与外设并行工作, 提高了 CPU 的效率。在 DMA 方式下, CPU 通过硬件 DMAC 来完成软件的工作, 使得外设和内存的信息传送速度大大提高, 从而 CPU 效率也得以提高。I/O 处理机方式利用 I/O 处理机承担 I/O 操作, 使 CPU 完全摆脱了管理和控制 I/O 设备的负担。

**例 6.4** 对一个 DMA 控制器的初始化工作包括哪些内容?

**【解】** 典型的 8237A 初始化工作包括:

- (1) 对 8237A 进行复位。
- (2) 将 DMA 传送中内存的起始地址的低 16 位写入基地址和当前地址寄存器, 从 I/O 端口输出最高 4 位地址 (或从页面寄存器输出最高 4 位地址)。
- (3) 将传送字节总长写入基地址和当前字节计数器。
- (4) 写入正确的模式字。
- (5) 写入正确的屏蔽字。
- (6) 写入正确的命令。

**例 6.5** 在某 8086 微机系统中有一外设, 使用 I/O 映象的 I/O 寻址方式, 要求该外设地址为 38H。试画出其译码器的连接电路, 使其译码器输出满足上述地址要求, 译码器使用 74LS138 芯片。

**【解】** 其译码器的连接电路如图 6.2 所示。

**【评注】** 注意: CPU 的  $\overline{M/\overline{IO}}$  线接在译码器的控制输入端时, 要接在低电平有效的那一端, 这是因为在执行 IN 或 OUT 指令时, CPU 的  $\overline{M/\overline{IO}}$  管脚输出低电平有效信号。

**例 6.6** 有 8 个发光二极管, 其阴极上加低电平则亮, 用 8212 芯片作 I/O 接口与 8086 CPU 通信。若要这些二极管同时亮或灭, 亮和灭的时间分别是 50ms 和 20ms, 用软件延时, 试编程序完成上述要求并画出其硬件接口电路。

**【解】** 软件编程如下所示。

```

...
A: MOV AL, 00H
   OUT CS8212, AL           ; CS8212 为 8212 的地址
   CALL RLY50               ; RLY50 为 50ms 延时子程序
   MOV AL, 0FFH
   OUT CS8212, AL
   CALL RLY20               ; RLY20 为 20ms 延时子程序
   JMP A
...

```

硬件接口电路设计如图 6.3 所示。

**例 6.7** 对 8237A 的 4 个通道基地址和当前地址寄存器、基和当前字节计数器进行测试检查, 先写入 FFFFH 后再读出, 比较读/写操作是否正确。若正确, 则再写入 0000H,

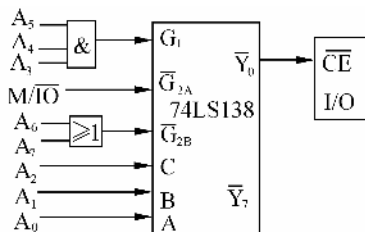


图 6.2 译码器连接电路图

同样读出校验，若仍正确，则认为 8237A 工作正常，就可以对其初始化；若有错，则执行暂停命令。

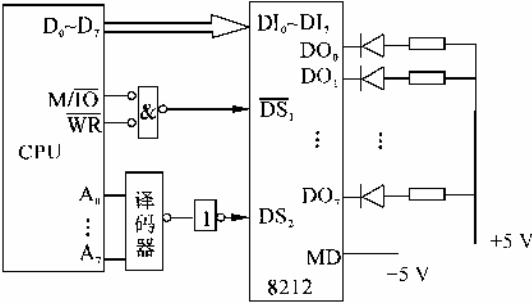


图 6.3 硬件接口电路

【解】 程序中用标号 DMA 表示首地址。程序段如下所示。

```
MOV AL,04H           ; 写命令寄存器
OUT DMA+8H,AL        ; 禁止 DMA 工作
OUT DMA+0DH,AL       ; 送复位命令
MOV AL,0FFH
C16: MOV BL,AL
    MOV BH,AL
    MOV CX,8
    MOV DX,DMA
C17: OUT DX,AL        ; 写入低字节
    OUT DX,AL        ; 写入高字节
    IN AL,DX         ; 读出低字节
    MOV AH,AL
    IN AL,DX         ; 读出高字节
    CMP BX,AX        ; 判断是否正确
    JE C18
    HLT
C18: INC DX           ; 口地址加 1, 指向下一寄存器
    LOOP C17
    INC AL            ; 使 AL=0, 再循环一次
    JZ C16
    JZ C16
```

6.3 习题及全解

一、填空题

1. CPU 与 I/O 接口间的信息一般包括\_\_\_\_\_、\_\_\_\_\_和\_\_\_\_\_3 种类型。  
三类信息的传送方向应该是（从哪到哪）\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_。

2. CPU 从 I/O 接口的\_\_\_\_\_中获取外部设备的“准备好”、“忙”或“闲”等信息。
3. I/O 数据缓冲器主要用于协调 CPU 与外设的\_\_\_\_\_上的差异。
4. 一般 I/O 端口的编址方式可以分为\_\_\_\_\_和\_\_\_\_\_两种方式。
5. 8237A 有\_\_\_\_\_个完全独立的 DMA 通道。
6. 8237A 一共占用\_\_\_\_\_个输入/输出端口地址。
7. CPU 与外设传送数据时, 输入/输出方式有\_\_\_\_\_, \_\_\_\_\_, \_\_\_\_\_和\_\_\_\_\_。

**【解】**

1. 数据信息; 状态信息; 控制信息; 从 CPU 到 I/O 接口或从 I/O 接口到 CPU (双向); 从 I/O 接口到 CPU; 从 CPU 到 I/O 接口
2. 状态寄存器
3. 速度
4. I/O 端口单独编址; 存储器统一编址
5. 4
6. 8
7. 程序控制传送方式; 中断控制传送方式; DMA 方式; I/O 处理机方式

**二、选择题**

1. 主机与外设传送数据时, 采用\_\_\_\_\_, CPU 的效率最高。  
A. 程序查询方式                      B. 中断方式  
C. DMA 方式                            D. 同步方式
2. 主机与外围设备传送数据时, 采用\_\_\_\_\_, 主机与外设是串行工作的。  
A. 程序查询方式                      B. 中断方式  
C. DMA 方式                            D. I/O 处理机方式
3. 在 DMA 传送过程中, 实现总线控制的部件是\_\_\_\_\_。  
A. CPU                                    B. 外部设备  
C. DMAC                                 D. 存储器
4. 在 DMA 方式下, CPU 与总线的关系是\_\_\_\_\_。  
A. 只能控制数据总线                  B. 只能控制地址总线  
C. 成隔离状态                          D. 成短接状态
5. 如果采用两级 8237 级联方式, 最多可以构成\_\_\_\_\_个 DMA 通道。  
A. 2                                        B. 4                                        C. 8                                        D. 16
6. 可作为简单输入接口的电路是\_\_\_\_\_。  
A. 三态缓冲器                          B. 锁存器  
C. 反相器                                D. 译码器

**【解】**

1. C    2. A    3. C    4. C    5. D    6. A

### 三、问答题

1. CPU 与外部设备之间通信为什么要通过接口?
2. CPU 与外设交换信息的控制方式有哪几种?

【解】

1. 因为外设提供的信息种类繁多, 提供数据的速度也与 CPU 速度不匹配, 所以要通过 I/O 接口对信息形式进行转换 (如串行与并行数据的转换、正负逻辑的转换、电平转换等) 以及数据传送速度匹配等作用。

2. 控制方式有程序控制传送方式、中断控制传送方式、DMA 方式和 I/O 处理机方式。

### 四、接口设计题

1. 在某 8086 微机系统中有一外设, 使用存储器映象的 I/O 寻址方式, 要求该外设地址为 01000H。试画出其译码器的连接电路, 使其译码器输出满足上述地址要求, 译码器使用 74LS138 芯片。

2. 现有两个输入设备, 使用程序查询方式与 CPU 通信。当状态位  $D_0=1$  时, 为第 1<sup>#</sup>设备输入字符; 当状态位  $D_1=1$  时, 为第 2<sup>#</sup>设备输入字符; 当状态位  $D_3=1$  时, 两个设备中任意一个结束输入过程。设状态端口地址为 0024H, 1<sup>#</sup>设备数据端口地址为 0026H, 2<sup>#</sup>设备数据端口地址为 0028H, 输入字符缓冲区首地址分别为 BUFF1 和 BUFF2, 试编一程序完成从输入设备输入 100 个字符。

3. 现有一台硬币兑换器, 平时等待纸币输入, 只有从状态端口 FAH 查到  $D_2=1$  时, 才有纸币输入, 此时可从数据输入端口中测出纸币品种, 1 角纸币代码为 01, 2 角纸币代码为 02, 5 角纸币代码则为 03。状态端口  $D_3=1$  后, 把兑换的 5 分硬币数 (十六进制) 从数据输出端口输出, 设数据输入端口地址为 FCH, 数据输出端口地址为 FEH。编程完成以上要求。

4. 利用 8237A 的通道 2, 由磁盘输入 32 KB 的一个数据块, 传送至内存 68000H 开始的区域, 采用增量、块连续的方式, 传送完不自动初始化, 磁盘的 DREQ 和 DACK 都是高电平有效。试编写初始化程序。

【解】

1. 译码器的连接电路如图 6.4 所示。

```
2.      .MODEL SMALL
      .DATA
BUFF1  DB 100 DUP(?)
BUFF2  DB 100 DUP(?)
      .CODE
      .STARTUP
      MOV SI,OFFSET BUFF1
      MOV DI,OFFSET BUFF2
      MOV CX,100
```

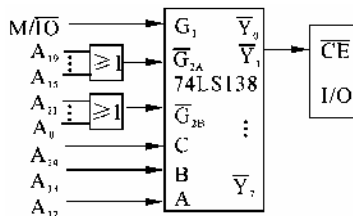


图 6.4 译码器的连接电路图



```

TEST0:  MOV DX,0024H
        IN AL,DX
        TEST AL,01H
        JZ TEST1
IN1:    MOV DX,0026H
        IN AL,DX
        MOV[SI],AL
        INC SI
        DEC CX
        MOV DX,0024H
        IN AI,DX
        TEST AL,08H
        JZ IN1
        JMP NEXT
TEST1:  TEST AI,02H
        JZ NEXT
IN2:    MOV DX,0028H
        IN AI,DX
        MOV[DI],AL
        INC DI
        DEC CX
        MOV DX,0024H
        IN AL,DX
        TEST AL,08H
        JZ IN2
NEXT:   CMP CX,0
        JNZ TEST0
        .EXIT
        END

```

```

3.      ...
TEST1:  IN AL,0FAH
        TEST AL,04H
        JZ TEST1
        IN AL,0FCH
        CMP AL,01
        JE P1
        CMP AL,02
        JE P2
        CMP AL,03
        JE P3
        JMP DOWN
P1:     MOV BL,02H
        JMP TEST2

```

```

P2:      MOV BL,04H
          JMP TEST2
P3:      MOV BL,0AH
TEST2:   IN AL,0FAH
          TEST AL,08H
          JZ TEST2
          MOV AL,BL
          OUT 0FEH,AL
          JMP TEST1
DOWN:    .EXIT
          END

```

4. 设首地址为 DMA，初始化程序片段如下所示。

```

MOV AL,06H
OUT DMA+0DH,AL      ; 送复位命令
MOV AI,00H
OUT DMA+4,AL        ; 送基地址和当前地址低 8 位
MOV AI,80
OUT DMA+4,AL        ; 送基地址和当前地址高 8 位
MOV DX,0081H
MOV AL,06H
OUT DX,AI           ; 送最高 4 位的地址
MOV AL,0FFH
OUT DMA+5,AL
MOV AL,7FH
OUT DMA+5,AI        ; 送基字节和当前字节计数器初值
MOV AI,86H
OUT DMA+0BH,AI      ; 根据题意写入方式字
MOV AL,02H
OUT DMA+0AH,AI      ; 写入屏蔽字(2号通道去除屏蔽)
MOV DX,DMA+08H      ; DMA+08H 为控制寄存器的端口
MOV AL,10000000      ; 对 8237A 设置控制命令字，DREQ 和 DACK 都是高电平有效，固定优
                      ; 优先级，启动工作
OUT DX,AL

```

## 7.1 重点内容提要

### （一）中断和异常

中断是指由于某个事件的发生（硬件的或软件的），计算机暂停执行当前的程序，转而执行另一程序，以处理发生的事件，处理完毕后又返回原程序继续原作业的过程。

通常，把因外部事件而改变程序执行的流程，去处理外部事件的过程叫做中断，也称硬件中断或外部中断；把因内部意外而改变程序执行的流程，以报告出错情况和异常状态的过程叫做异常，也称软件中断或内部中断。中断和异常其实都是中断过程，本章将它们统称为“中断”。

### （二）中断源

能引起中断的原因，或是能发出中断请求信号的来源，称为中断源。CPU 外部中断源有 I/O 设备、数据通道、实时时钟和故障源等；内部中断源有 CPU 指令产生了异常（如除 0、溢出、单步调试等），或执行了软件中断指令 INT n。

### （三）中断类型

根据中断源的不同，中断源可按图 7.1 所示分类。

可屏蔽中断和非屏蔽中断的区别是：CPU 对可屏蔽中断是否响应要根据标志寄存器中 IF 位来决定，若 IF=1，则响应，否则不响应；非屏蔽中断不受 IF 的影响，常用于 CPU 感知突发致命事件。

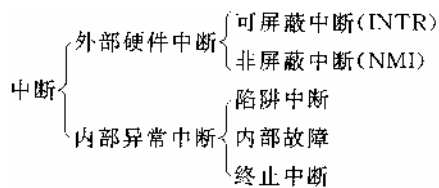


图 7.1 中断类型

### （四）中断优先顺序

CPU 的中断优先权从高到低排列为：除法出错中断、溢出中断，INT n→NMI→INTR→单步中断。

### （五）中断类型码的分配

中断类型码是连接中断源和中断处理程序的唯一桥梁。80x86 可处理 256 级中断，中断类型码可以是 0~255，一部分由系统占用；一部分由用户支配。

### （六）中断处理的一般过程

对各种中断的响应和处理过程是不完全相同的，主要区别在于如何获取相应的中断类型码。对于 **INT n**、**NMI**、单步中断等中断类型码是系统规定好的自动形成的。而 **INTR**，则必须判断 **IF** 是否为 1，若是，CPU 才响应，并读取该中断源的中断类型码。在获取了中断类型码以后，中断处理过程是一样的。

- （1）把 CPU 的标志寄存器压栈保护。
- （2）陷阱标志 **TF** 的状态送入暂存器，并对 **TF** 和 **IF** 清 0。
- （3）保护断点。将 **CS** 和 **IP** 压栈保护。
- （4）根据中断类型码取中断服务程序的入口地址。
- （5）执行中断服务程序。
- （6）恢复断点地址。
- （7）恢复标志寄存器的内容。
- （8）返回原程序，顺序执行下一条指令，再判断有无中断。

### （七）中断向量和中断描述符表

每个中断都分配有中断类型码，CPU 通过中断类型码经过处理要找到对应的中断服务程序的入口地址。在实模式下，由中断向量表将中断类型码和中断服务程序的入口地址联系起来；而在保护模式下，是由中断描述符表来对两者进行联系。

中断向量表设置在系统 **RAM** 的最低端 **00000H~003FFH** 的 **1KB** 内，表中共有 256 个中断类型码对应的向量值，每个向量占用 4 个字节。前两个字节为中断服务程序入口地址的偏移值 **IP**，后两个字节为服务程序的段基址值 **CS**。这 4 个单元的地址中的最小地址称为向量地址。CPU 可以通过 **CS** 和 **IP** 的值得到该中断类型的中断服务程序的入口地址。

中断描述符表（**IDT**）在内存中的首地址保存在 CPU 内部的 **IDTR** 系统寄存器中。在 CPU 响应中断过程中，CPU 把中断类型码乘以 8，与 **IDTR** 中的基地址相加，指示中断描述符表中的某一中断门或陷阱门。CPU 将门描述符中的选择符送 **CS** 寄存器，并根据选择符中 **TI** 位从 **LDT** 或 **GDT** 中选择一个段描述符，送入 **CS** 的段描述符寄存器中。这时，由段描述符中的基地址确定中断服务程序的入口地址。

### （八）中断向量的建立

为了让 CPU 响应中断后正确转入中断服务，建立中断向量表是非常重要的。在建立中断向量表时，可以采用以下方法。

- （1）绝对地址置入法。
- （2）串指令装入法。
- （3）**DOS** 调用法。
- （4）直接接入法。

### （九）可编程控制器 8259A

**8259A** 是一个可编程的中断控制器，用于实现中断优先权管理、中断屏蔽、自动输

送中断向量码等功能。单片 8259A 可管理 8 级硬件中断，级联时可管理 64 级中断。

### 1. 8259A 的内部结构

8259A 的内部结构如图 7.2 所示。

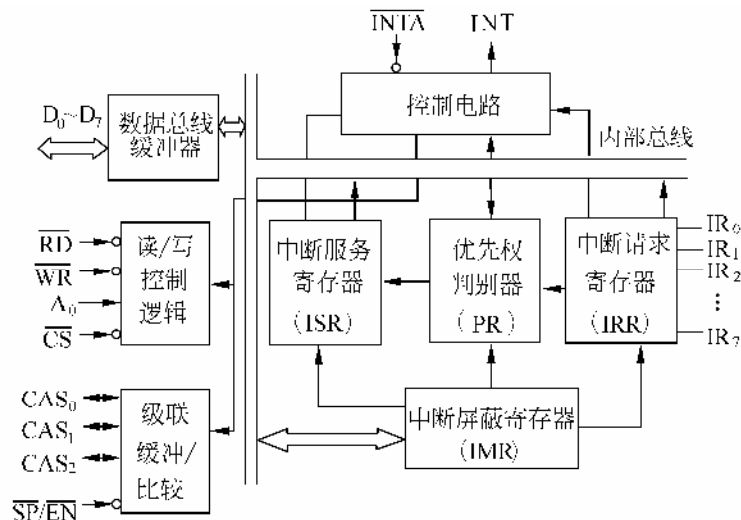


图 7.2 8259A 的内部结构框图

### 2. 有关寄存器

- 中断请求寄存器 IRR：用于存放所有从  $IR_i$  引脚输入的有效的中断请求信号，即当  $IR_i$  引脚上出现高电平或上跳沿时，IRR 中对应的  $D_i$  位置 1，并且在该请求的中断响应周期中被清除。
- 中断屏蔽寄存器 IMR：决定是否让 IRR 锁存的中断请求进入优先权判别器。
- 优先权判别器 PR：根据中断请求的优先等级，从未屏蔽的 IRR 各请求位中选出优先等级最高的一个，在第一个  $\overline{INTA}$  信号到来时，将其对应的中断服务寄存器 ISR 中的相应位置 1，并通过控制电路在第二个  $\overline{INTA}$  信号到来时送出该中断源的中断类型码。
- 中断服务寄存器 ISR：用于指明请求已被响应、服务尚未结束的中断源。

### 3. 8259A 的编程

8259A 的编程包括两部分：初始化命令字和操作命令字的设置。初始化命令字有 4 个，称为 ICW1~ICW4，用于完成对 8259A 的预置。操作命令字有 OCW1~OCW3，在操作过程中可通过其来定义 8259A 的操作方式，并允许重新改变操作方式。

初始化命令字要按规定顺序写入 8259A 中，而操作命令字无需按顺序写入。一般情况下，初始化命令字一旦设定，工作过程中就不再改变。操作命令字则是由应用程序设定的，用于对中断处理过程做动态控制。在一个系统运行过程中，可多次改写操作命

令字。

#### 4. 中断管理方式

- 中断优先级有 4 种中断优先级：正常全嵌套方式、特殊全嵌套方式、优先级自动循环方式及优先级特殊循环方式（指定优先级循环）。
- 中断屏蔽：普通屏蔽方式是通过操作命令字 OCW1 设置中断屏蔽寄存器中的位来达到屏蔽/允许目的的，而特殊屏蔽方式是通过操作命令字 OCW3 设置特殊屏蔽允许位和特殊屏蔽方式位来完成的。
- 中断结束方式：分为自动结束方式和非自动结束方式。是在初始化时由初始化命令字 ICW4 的 AEOI 决定的。在中断自动结束方式中，系统一进入中断过程，8259A 就自动将当前中断服务寄存器 ISR 中的对应位清 0，这样，尽管系统正在为某个设备进行中断服务，但对 8259A 来说，当前中断服务寄存器中却没有对应位的指示，好像已经结束了中断服务一样。因此，中断自动结束方式一般用于中断没有嵌套的情况。在非自动结束方式下，当中断服务程序返回之前，必须向 8259A 发出中断结束命令 EOI，才能将 ISR 中的当前服务位复位。
- 数据缓冲方式：分为缓冲方式和非缓冲方式。在缓冲方式下， $\overline{SP/EN}$  是输出信号，与总线驱动器的输出允许相连；在非缓冲方式下，它是输入信号，用于确定本片是 8259A 的主片还是从片。在缓冲方式下，主片和从片是由 ICW4 的 M/S 位决定的。
- 中断请求触发方式：分为边沿触发方式和电平触发方式。在边沿触发方式下，8259A 决定中断请求为正跳变（边沿）触发；电平触发是 8259A 决定中断请求为高电平触发。

## 7.2 常考题型范例精解

**例 7.1** 什么是中断类型码？什么是向量地址？什么是中断向量？它们之间有何关系？

**【解】** 中断类型码是对中断源的编码。向量地址是向量在中断向量表中对应的连续几个单元的最小地址。中断向量是指存入的中断处理服务程序的入口地址。

中断系统根据中断类型码能唯一地确定中断向量，80x86 CPU 都拥有 256 个中断类型码。从 0~255 号。其中，都规定了中断向量表中各中断向量等长，且中断服务程序入口地址在向量表中按中断源的中断类型码排序。因此，向量表中共有 256 个向量，而且任意一个类型码乘上向量单元数再加上向量表的首地址可得到向量地址，从中取得中断服务程序的入口地址。

**例 7.2** 分别简述 8086 CPU 和 80486 CPU 得到了中断类型码之后，如何找到中断服务程序的入口地址。

**【解】** 对于 8086 CPU 来说，中断类型码乘 4 得到中断向量表的向量地址的偏移地址，其段基址是 DS=0000H。从向量表中取出 4 个单元的数据。其中前两个单元余中断

服务程序的首地址的偏移地址。后两个单元是中断服务程序的首地址的段基址,分别装入 IP 和 CS 中,从而得到中断服务程序的入口地址。而对于 80486 CPU 是通过中断描述符表 IDT 而不是中断向量表来协助处理。中断类型码乘 8 得到一个 32 位的地址,从 IDT 表中查它的选择符,用得到的选择符从 GDT 或 LDT 表中查段的描述符,描述符中包含着中断服务程序的段基址,段基址再和选择符中的 32 位偏移地址相加得到中断服务程序的入口地址。

**例 7.3** 中断控制器 8259A 的基本功能是什么? 有哪些命令字?

**【解】** 8259A 的基本功能有:

- (1) 接收外部中断的可屏蔽中断,并实施屏蔽。
- (2) 判断没有被屏蔽掉的所有请求服务的设备中谁的优先权最高。
- (3) 确定请求的设备优先权是否比当前正在服务的中断具有更高的优先权,若高,就向 CPU 发送一个中断请求。
- (4) 输送中断类型码。
- (5) 可编程确定 8259A 的工作方式。
- (6) 单片时可管理 8 级中断,级联时最多可管理 64 级中断。

8259A 有初始化命令字 ICW1、ICW2、ICW3、ICW4; 操作命令字 OCW1、OCW2、OCW3。

**例 7.4** 请描述 8259A 对 INTR 中断的处理过程。

**【解】** 8259A 的中断请求寄存器 IRR 接受外部的中断请求,IRR 有 8 位,它们分别和引脚  $IR_0 \sim IR_7$  相对应。接收来自某一引脚的中断请求并锁存,同时 IRR 寄存器中的对应位置 1。中断请求锁存之后,逻辑电路根据中断屏蔽寄存器 IMR 中的对应位决定是否让此请求通过。如果没有被屏蔽,将此中断请求输入中断优先级判别器 PR,和当前正在处理的中断相比较,从而决定哪个优先级更高。而当前中断服务寄存器就是用来存放现在正在处理的中断请求的。如果判断出新进来的中断请求具有足够高的优先级,那么,相应的逻辑电路会使 8259A 的 INT 端为 1。向 CPU 发出一个中断请求。如果此时 IF 为 1,即处于开中断状态,则 CPU 会连续发出两个  $\overline{INTA}$  信号,进入两个连续的中断响应周期。第一个  $\overline{INTA}$  信号到达时,8259A 完成以下 3 个动作:①使 IRR 的锁存功能失效,这样,在  $IR_0 \sim IR_7$  线上的中断请求信号就不予接受,直到第二个  $\overline{INTA}$  信号到来时,才又使 IRR 的锁存功能有效;②使当前中断服务寄存器 ISR 中相应位置 1,以便为优先级判别器 PR 以后的工作提供判断依据;③使 IRR 寄存器中的相应位清 0。第二个  $\overline{INTA}$  信号到达时,8259A 完成以下动作:①将当前中断源的中断类型码送到数据总线,CPU 据此转向相应的中断服务程序;②如果是中断自动结束方式,那么 8259A 将会在第一个  $\overline{INTA}$  信号到来时设置当前中断服务寄存器 ISR 的相应位清 0。

**例 7.5** 若 8086 系统中采用单片 8259A 作为外部可屏蔽中断的优先级管理,正常全嵌套方式、边沿触发、非缓冲连接、非自动中断结束,端口地址为 20H 和 21H。其中某中断源的中断类型码为 0AH,其中断服务子程序的入口地址是 2000:3A40H。

- (1) 请为 8259A 设置正确的初始化命令字,并编写初始化程序。
- (2) 请问本题中的中断源应与 IR 的哪一个输入端相连,其中断矢量地址是多少,

矢量区对应的 4 个单元的内容是什么？

【解】

(1) ICW1=00010011 (单片、边沿触发、必须设置 ICW4)。

ICW2=00001000 (中断类型码基值为 08H, 高 5 位为 00001, 低 3 位任意设置)。

无 ICW 3, 因采用的是单片 8259A 系统, 无需设置 ICW<sub>3</sub>

ICW4=00000001 (正常全嵌套方式, 非缓冲连接, 非自动中断结束)。

初始化程序如下所示。

```
MOV AL,13H          ; 设置 ICW1
OUT 20H,AL
MOV AL,08H          ; 设置 ICW2
OUT 21H,AL
MOV AL,01H          ; 设置 ICW4
OUT 21H,AL
```

(2) 本题中的中断源应与 IR<sub>2</sub> 相连, 其中断矢量地址为 0000: 0028H, 矢量区对应的 4 个单元的内容依次为 40H、3AH、00H、20H。

【评注】 8259A 的 ICW2 用来提供中断类型码, 中断类型码的高 5 位由用户指定, 低 3 位由 8259A 的中断请求输入端的 IR<sub>0</sub>~IR<sub>7</sub> 的编码决定。中断类型码基值规定最低 3 位为 0, 故本例中断类型码基值为 08H。本题中的中断类型码为 0AH, 即 00001010B, 所以此中断源一定连接 8259A 的 IR<sub>2</sub> 端。

中断向量表规定从内存的 00000H 开始按中断类型码顺序存放中断向量, 每个向量占用 4 个单元, 所以, 该中断源的向量地址为 0AH×4=28H, 即 0000: 0028H; 这 4 个单元存放中断服务程序入口地址, 规定 IP 在前、CS 在后, 且低位在前、高位在后, 故 4 个单元的内容依次为 40H、3AH、00H、20H。

**例 7.6** 下面为对一个主从式 8259A 系统进行初始化的程序段。请对以下程序段详细注释, 并具体说明各初始化命令字的含义。

主片初始化程序 (设主片的端口地址为 140H 和 141H) 如下所示。

```
MOV AL,11H
MOV DX,140H
OUT DX,AL
MOV AL,08H
INC DX
OUT DX,AL
MOV AX,04H
OUT DX,AL
MOV AL,01H
OUT DX,AL
```

从片初始化程序 (设从片的端口地址为 190H 和 191H) 如下所示。

```
MOV DX,190H
```



```

MOV AL,11H
OUT DX,AL
MOV AL,70H
INC DX
OUT DX,AL
MOV AL,02H
OUT DX,AL
MOV AL,01H
OUT DX,AL

```

**【解】** 根据各命令字的格式可以进行如下分析。

主片初始化程序（设主片的端口地址为 140H 和 141H）如下所示。

```

MOV AL,11H
MOV DX,140H
OUT DX,AL           ; 设置 ICW1, 中断请求信号边沿触发、级联方式, 须设 ICW4
MOV AL,08H
INC DX
OUT DX,AL           ; 设置 ICW2, 中断类型码为 08H~0FH
MOV AL,04H
OUT DX,AL           ; 设置 ICW3, IR2 与从片的 INT 相连
MOV AL,01H
OUT DX,AL           ; 设置 ICW4, 完全嵌套方式、非缓冲、正常中断
                     ; 结束

```

从片初始化程序（设从片的端口地址为 190H 和 191H）如下所示。

```

MOV DX,190H
MOV AL,11H
OUT DX,AL           ; 同主片 ICW1
MOV AL,70H
INC DX
OUT DX,AL           ; 设置 ICW2, 中断类型码为 70H~77H
MOV AL,02H
OUT DX,AL           ; 设置 ICW3, 从片 INT 与主片的 IR2 相连
MOV AL,01H
OUT DX,AL           ; 同主片 ICW4

```

**例 7.7** 若 8259A 以三片级联方式管理 22 级中断, A 从片的 INT 接到主片的 IR<sub>2</sub> 上, B 从片的 INT 接到主片的 IR<sub>6</sub> 上, 主、从片的中断请求采用边沿触发方式, 选用全嵌套、自动结束、非缓冲方式, 主片中断类型码为 60H~67H, A 片的中断类型码为 70H~77H, B 片的中断类型码为 80H~87H, 各 8259A 的 A<sub>0</sub> 均接到 CPU 的 A<sub>0</sub> 口地址可合理假设, 试对该中断管理系统进行初始化编程。

**【解】** 主片初始化程序（设主片地址位 80H 和 81H）如下所示。

```

MOV AL,11H
OUT 80H,AL                ; 设置 ICW1
MOV AL,60H
OUT 81H,AL                ; 设置 ICW2
MOV AL,01000100B
OUT 81H,AL                ; 设置 ICW3
MOV AL,00000011B
OUT 81H,AL                ; 设置 ICW4
; 从片 A 初始化 (设从片 A 地址为 82H 和 83H)
MOV AL,11H
OUT 82H,AL                ; ICW1
MOV AL,70H
OUT 83H,AL                ; ICW2
MOV AL,02H
OUT 83H,AL                ; ICW3
MOV AL,03H
OUT 83H,AL                ; ICW4
; 从片 B 初始化 (设从片 B 地址为 84H 和 85H)
MOV AL,11H
OUT 84H,AL                ; ICW1
MOV AL,80H
OUT 85H,AL                ; ICW2
MOV AL,06H
OUT 85H,AL                ; ICW3
MOV AL,03H
OUT 85H,AL                ; ICW4

```

**例 7.8** 在 IBM PC/XT 系统中，通过读取 ISR 的值，判断是否为硬件中断。若是，试清除正在服务的中断优先级，并发非特殊 EOI 命令结束中断。

**【解】**

```

MOV AL,0BH                ; 0CW3, 置读 ISR 状态
OUT 20H,AL
NOP
IN AL,20H                 ; 读 ISR
MOV AH,AL                 ; 保存正在服务的优先级
OR AL,AH
JNZ HW-INT                ; 不为 0, 硬件中断, 转 HW-INT
...                        ; 为 0, 为非硬件中断
HW-INT: IN AI,21H          ; 读 IMR
OR AL,AH                  ; 屏蔽当前服务的优先级
OUT 21H,AL
MOV AL,20H                ; 发出非特殊结束命令
OUT 20H,AL

```

## 7.3 习题及全解

### 一、填空题

1. 硬件中断可分为\_\_\_\_\_和\_\_\_\_\_两种。
2. CPU 响应可屏蔽中断的条件是\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_。
3. 当 CPU 在响应外设中断, 并送入中断子程序的过程中, 要完成: \_\_\_\_\_、\_\_\_\_\_  
\_\_\_\_\_, \_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_。
4. 8259A 有两种中断触发方式, 分别是\_\_\_\_\_和\_\_\_\_\_。
5. 8259A 可编程控制器, 当其单片使用时可同时接收\_\_\_\_\_个外设的中断请求; 当级联使用时, 其主片的\_\_\_\_\_应与从片的\_\_\_\_\_连接。
6. 中断矢量表的 1KB 空间中, 可供用户使用的指针范围为 00080H~003FCH, 其对应的类型号为\_\_\_\_\_~\_\_\_\_\_。
7. 8259A 有\_\_\_\_\_个命令字, 3 片 8259A 接成级联可管理\_\_\_\_\_级中断。
8. 中断系统中, 实方式下的中断类型码与该中断类型码对应的中断服务程序入口地址之间的连接表称为\_\_\_\_\_, 保护方式下的称为\_\_\_\_\_。
9. 若某外设的中断类型码为 3AH, 则该中断源的中断请求信号应连在 8259A 的\_\_\_\_\_  
\_\_\_\_\_端, 且对应的中断向量地址为\_\_\_\_\_。
10. 若某外设的中断类型码为 48H, 该中断源对应的中断向量地址为\_\_\_\_\_。

#### 【解】

1. 可屏蔽中断; 非屏蔽中断
2. IF=1; 有中断请求; 现行指令执行完毕
3. 从数据线上读取中断类型码; 将标志寄存器的值压入堆栈; 将 IF 和 TF 清 0; 保护断点; 根据中断类型码, 找到中断服务程序的入口地址, 转入中断子程序
4. 电平触发方式; 边沿触发方式
5. 8; IR 端; INT 端
6. 32 (或 20H); 255 (或 FFH)
7. 7; 22
8. 中断向量表; 中断描述符表
9. IR<sub>2</sub>; 00E8H
10. 0000: 0120H

### 二、选择题

1. 中断向量可以提供\_\_\_\_\_。  
A. 被选中设备的起始地址  
B. 传送数据的起始地址  
C. 中断服务程序的入口地址  
D. 主程序的断点地址
2. 中断向量地址是\_\_\_\_\_。

- A. 子程序入口地址
  - B. 中断服务程序入口地址
  - C. 中断服务程序入口地址的地址
  - D. 主程序的断点地址
3. 在中断响应周期内, 将 IF 置 0 是由\_\_\_\_\_。
- A. 硬件自动完成的
  - B. 用户在编制中断服务程序时设置的
  - C. 关中断指令完成的
4. 对于掉电的处理, CPU 是通过\_\_\_\_\_来处理的。
- A. 软件中断
  - B. 可屏蔽中断
  - C. 非屏蔽中断
  - D. DMA
5. 下面的中断中, 只有\_\_\_\_\_需要硬件提供中断类型码。
- A. INT 0
  - B. INT n
  - C. NMI
  - D. INTR
6. 一片 8259A 只占两个 I/O 地址, 可以用地址码 A1 来选择端口。如果其中一个端口地址为 92H, 则另一个端口地址为\_\_\_\_\_。
- A. 90H
  - B. 91H
  - C. 93H
  - D. 94H
7. 当多片 8259A 级联使用时, 对于主 8259A, 信号 CAS<sub>0</sub>~CAS<sub>2</sub> 是\_\_\_\_\_。
- A. 输入信号
  - B. 输出信号
  - C. 输入/输出信号
8. 当多片 8259A 级联使用时, 对于从 8259A, 信号 CAS<sub>0</sub>~CAS<sub>2</sub> 是\_\_\_\_\_。
- A. 输入信号
  - B. 输出信号
  - C. 输入/输出信号
9. 8259A 中的中断服务寄存器用于\_\_\_\_\_。
- A. 指示有外设向 CPU 发中断请求
  - B. 指示有中断正在进行
  - C. 开放或关闭中断系统
10. 8259A 中的中断请求寄存器用于\_\_\_\_\_。
- A. 指示有外设向 CPU 发中断请求
  - B. 指示有中断正在进行
  - C. 开放或关闭中断系统
11. 通常情况下, 一个外中断服务程序的第一条指令是 STI, 其目的是\_\_\_\_\_。
- A. 开放所有屏蔽中断
  - B. 允许低一级中断产生
  - C. 允许高一级中断产生
  - D. 允许同一级中断产生
12. PC 采用向量中断方式处理 8 级外中断, 中断号依次为 08H~0FH, 在 RAM0: 2CH 单元开始依次存放 23H、FFH、00H 和 F0H 四个字节, 该向量对应的中断号和中断服务程序入口地址是\_\_\_\_\_。
- A. 0CH, 23FF: 00F0H
  - B. 0BH, F000: FF23H
  - C. 0BH, 00F0: 23FFH
  - D. 0CH, F000: FF23H
13. 已知中断类型为 14H, 它的中断向量存放在存储器的向量单元\_\_\_\_\_中。
- A. 00050H, 00051H, 00052H, 00053H
  - B. 00056H, 00057H, 00058H, 00059H
  - C. 0000: 0050H, 0000: 0051H, 0000: 0052H, 0000: 0053H
  - D. 0000: 0056H, 0000: 0057H, 0000: 0058H, 0000: 0059H

- 【解】

- 【解】

3. 80486 采用和 8086 相同的方式获取中断矢量号。

在实方式下, 80486 采用和 8086 相同的方式处理中断, 即在内存 0 段设置一个中断矢量表, 中断响应时, 根据中断矢量号从中断矢量表中相应的位置取出中断服务程序的首地址装入 IP 和 CS 中, 引导程序转至中断服务程序。

在保护方式下, 80486 通过中断描述符表来协助中断响应和处理。中断类型号作为中断描述符表 IDT 的索引, 从 IDT 表中获取一个 8 字节的门描述符, CPU 将门描述符中的选择符送 CS 寄存器, 并根据选择符中的 IT 位是 0 或 1 从 GDT 或 LDT 中选择一个段描述符, 送 CS 的描述符寄存器中。这时, 由段描述符中的基地址确定中断服务程序所在位置的起始地址, 由门描述符中的偏移地址确定中断服务程序的入口地址。

4. 从 8259A 内部编程部件的工作过程可知, 8259A 的中断屏蔽寄存器 IMR 决定是否让中断请求寄存器 IRR 锁存的中断请求进入优先权判别器 PR。也就是说, IMR 中的 8 位二进制控制的是对应的 8 个中断源请求是否通过 8259A 向 CPU 申请中断; 而中断允许标志 IF 是 CPU 对外部可屏蔽中断的总屏蔽/允许位, 控制了 CPU 是否响应 8259A 的中断请求。所以, 只有当 IF 被设置为 1, 8259A 某引脚外部中断源在 IMR 中对应位为 0 时, 该引脚上有效的中断请求才有可能得到 CPU 的响应。

5. 该中断源应与 8259A 的 IR<sub>6</sub> 相连, 其中断矢量地址是 0000: 0118H, 矢量区对应的 4 个单元内容依次是: D4H、23H、A0H、16H。

6. 在 8259A 初始化编程中, 当系统是 8269A 级联情况时, 必须设置 ICW3, 且主片和从片的 ICW3 格式不同。主片的 ICW3 表明主 8259A 的哪个 IR 端接有从 8259A, 从片的 ICW3 表明它接在主 8259A 的哪个 IR 端。

7. 中断向量表和中断描述符表都是中断类型码与该中断类型相对应的中断服务程序的入口地址之间的连接表。中断向量表对应 16 位模式下, 表内内容为中断服务程序入口地址的 IP 和 CS 值, 每个中断类型码对应 4 个单元; 中断描述表对应 32 位模式下, 表内内容为中断门描述符, 每个中断类型码对应 8 个单元。

#### 四、8259A 的初始化编程题

1. 在某应用中, 8259A 工作于正常全嵌套方式, 要求在为中断源 IR<sub>4</sub> 服务时, 设置特殊屏蔽方式, 开放较低级的中断请求, 请编写有关程序片段。已知 8259A 的端口地址为 20H 和 21H。

2. 请为中断类型码为 0FH 的外部中断源设置中断矢量, 中断服务子程序名为 INTER。

3. 试按照如下要求对 8259A 设置初始化命令字: 系统中有一片 8259A, 中断请求信号用电平触发方式; 下面要用 ICW4 设置嵌套及中断结束方式等状态, 中断类型码为 60H~67H, 用特殊全嵌套方式、无缓冲、采用中断自动结束方式。8259A 的端口地址为 93H 和 94H。

4. 以下程序用来检查系统中 8259A 的正确性。请对该程序进行详细注释。

```
; 8259A 的端口地址为 20H 和 21H
MOV AL, 0
OUT 21H, AL
```

```

IN AL,21H
OR AL,AL
JNZ ERROR
MOV AL,0FFH
OUT 21H,AL
TN AL,21H
ADD AL,1
JNZ ERROR
...

```

5. 某系统采用级联方式,主 8259A 的中断类型码从 30H 开始,端口地址为: 20H 和 21H。从 8259A 的 INT 接主片的  $IR_7$ ,从片的中断类型码从 40H 开始,端口地址为: 22H 和 23H。均不要 ICW4。试对其进行初始化编程。

### 【解】

1. 根据题意,程序片段为:

```

IR4SER: ...
CII                      ; 关中断,保证设置命令时不响应中断
MOV AL,68H               ; 设置特殊屏蔽方式(OCW3=68H)
OUT 20H,AL
IN AL,21H                ; 读 IMR 状态
OR AL,00010000B          ; 屏蔽  $IR_4$  和 OCW1
OUT 21H,AL
STI                      ; 开中断
...                      ; 继续对  $IR_4$  进行中断处理
...                      ; 若有较低级的中断请求,CPU 给予响应,并在中断处理返回
...                      ; 继续对  $IR_4$  进行中断处理
CLT                      ; 为以下设置命令而关中断
IN AL,21H                ; 读 IMR 状态
AND AL,11101111B         ; 开放  $IR_4$  的中断请求
OUT 21H,AL               ; 恢复原来的屏蔽字
MOV AL,48H               ; 复位特殊屏蔽方式(OCW3=48H)
OUT 20H,AL
...
...
MOV AL,20H               ; 发送中断结束命令
OUT 20H,AL
IRET                     ; 返回主程序

```

2. 根据题意,程序片段为:

```

MOV AX,0
MOV ES,AX                ; 段地址 0 送 ES
MOV DI,0FH*4             ; 将类型为 0FH 的中断矢量的偏移地址送 DI
MOV AX,OFFSET INTER      ; 中断服务程序首地址的偏移地址
CLD

```

```

STOSW
MOV AX,SEG INTER          ; 中断服务程序首地址的段地址
STOSW

3.  MOV AL,00011011B
    OUT 94H,AL              ; ICW1
    MOV AL,01100000B
    OUT 93H,AL              ; ICW2
    MOV AL,00010011B
    OUT 93H,AL              ; ICW4

4. ; 8259A 的端口地址为 20H,21H
    MOV AL,0
    OUT 21H,AL              ; OCW1, IMR 清 0
    IN AL,21H               ; 读 IMR
    OR AL,AL
    JNZ ERROR               ; 有错, 转出错处理程序
    MOV AL,0FFH             ; 若正常, 置 IMR 全 1
    OUT 21H,AL
    IN AL,21H               ; 读 IMR
    ADD AL,1
    JNZ ERROR               ; 有错, 转出错处理程序
    ...                     ; 正常, 继续执行程序

5. ; 主片初始化程序
    MOV AL,00010001B
    OUT 20H,AL
    MOV AL,30H
    OUT 21H,AL
    MOV AL,80H
    OUT 21H,AL
    ; 从片初始化程序
    MOV AL,00010001B
    OUT 22H,AL
    MOV AL,40H
    OUT 23H,AL
    MOV AL,07H
    OUT 23H,AL

```



### 8.1 重点内容提要

#### （一）总线的分类

总线是一组信号线的集合，是一种在各模块之间传送信息的公共通路。按在系统的不同层次位置上分类，总线可分为4种：片内总线、在板局部总线、系统总线和通信总线。

#### （二）总线标准

为了便于不同厂家生产的模块能灵活地组成系统并具有通用性，总线必须有一定的标准。每种总线都有详细的规范，包括机械结构规范、功能结构规范和电气规范。

局部总线标准如 IBM PC、ISA、EISA、VL 和 PCI 等。

系统总线标准如 S-100、MULTIBUS I、MULTIBUS II、VME、STDBUS、STD32 等。

通信总线标准如 IDE、SCSI、Centronics、RS-232C、VXI、IEEE-488 等。

#### （三）总线数据传输

总线上完成一次数据传输要经过4个阶段：申请占用总线阶段、寻址阶段、传数阶段和结束阶段。

总线传输控制方式可以是同步传输、异步传输和半同步传输。

### 8.2 常考题型范例精解

**例 8.1** 总线规范的基本内容是什么？

**【解】**

- （1）机械结构规范。规定模块尺寸、总线插头、边沿连接器等规格。
- （2）功能结构规范。确定引脚名称与功能，以及其相互作用的协议。
- （3）电气规范。规定信号逻辑电平、负载能力及最大额定值、动态转换时间等。

**例 8.2** 根据在微机系统的不同层次上的总线分类，共有哪几类总线？各类总线的用途是什么？

**【解】** 可分为4类。

- (1) 片内总线。此类总线在集成电路芯片内部, 用来连接各功能单元的信息通路。
- (2) 在板局部总线。用于在印刷电路板上连接各芯片之间的公共通路。
- (3) 系统总线。又称内总线, 用来连接构成微机的各插件板。
- (4) 通信总线。又称外总线, 用于微机系统与系统之间、微机系统与外设、微机系统和仪器仪表之间的通信通道。

**例 8.3** 总线数据传输的同步、异步和半同步控制各有何特点?

**【解】** 同步控制的特点是简单, 数据传送在一个共同的时钟信号控制下进行, 总线操作有固定的时序, 只适应于存取时间匹配的各设备之间传输数据。

异步控制利用“握手”信号来实现, 对高速设备能高速操作, 而对低速设备能低速操作, 但传输延迟是同步总线的两倍。

半同步控制结合了同步和异步总线的优点, 既具有同步总线的速度。又具有异步总线的适应性。

**例 8.4** 举出几种典型的通信总线, 并说明它们主要用于哪些应用场合。

**【解】** 典型的通信总线有 IEEE 488、VXI、SCSI、IDE、Centronics、RS-232C 等。

IEEE 488 和 VXI 总线一般用于自动测试系统, 用来连接仪器仪表; SCSI 总线用于计算机与软盘、硬盘、光盘、扫描仪等外设的连接; IDE 总线只支持硬盘驱动器; Centronics 总线用于与打印机或绘图仪的连接; RS-232C 是数据终端设备和数据通信设备之间的接口标准。

**例 8.5** IBM PC 总线(或称为 XT 总线)、ISA 总线和 EISA 总线的总信号线、地址线、数据线有何区别?

**【解】** XT 总线共有 62 个引脚信号, 其中有 8 位数据线、20 位地址线。ISA 总线在 XT 总线的基础上增加了 36 个信号, 共有 98 个引脚信号, 其中数据线 16 位、地址线 24 位, 寻址能力达 16MB。EISA 总线的引脚在 ISA 的基础上扩展到了 198 个, 数据线和地址线都为 32 位, 寻址能力可达 4GB。

**例 8.6** 继 ISA 和 EISA 总线之后, 为什么要推出 VL 和 PCI 总线? 哪种局部总线更有发展前途?

**【解】** 随着 80486 和 Pentium 等高性能 CPU 的问世, 因其内部速度大大提高了, 再加上集成高速缓存和数字协处理器 FPU, 高速的 CPU 和内存访问同慢速 I/O 操作成为 PC 技术的瓶颈。多媒体的出现, 对于图形和高速显示要求更快、更大量地进行信息传输。而这些问题, ISA 和 EISA 总线都无法解决。为此, 一些厂商为主板设计了一种特殊的高速插槽, 将高速外部设备控制卡直接挂到 CPU 局部总线上, 并以 CPU 速度运行。这种总线有 VL 总线和 PCI 总线, 主要支持高速的外部设备板。对于其他慢速设备仍保持原来的 ISA 和 EISA 总线标准。这样既保持了兼容性, 又解决了瓶颈问题。

相对来说, PCI 总线更有发展前途, 因为 PCI 总线开放性好, 不受处理器类型的限制, 具有广泛的兼容性。

**例 8.7** STD 总线主要用于什么场合?

**【解】** STD 总线主要用于工业控制微型机。STD 总线采用小板结构、高度模块化, 采用一套高可靠性措施, 使该总线构成的工业控制系统, 可以长期可靠地工作于恶劣环

境下。STD 总线不仅是国际上流行的工业控制标准总线，也是国内工业控制机首选的标准总线。

## 8.3 习题及全解

### 一、填空题

1. 按照总线在系统的不同层次位置上分类，总线可分为\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_和\_\_\_\_\_4 大类。
2. 总线上要完成一次数据传输要经过\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_和\_\_\_\_\_、4 个阶段。
3. 总线传输控制方式可以是：\_\_\_\_\_、\_\_\_\_\_和\_\_\_\_\_。

#### 【解】

1. 片内总线；在板局部总线；系统总线；通信总线
2. 申请占用总线；寻址；传数；结束
3. 同步传输；异步传输；半同步传输

### 二、问答题

1. 采用标准总线结构组成微机系统有何优点？
2. 目前有哪几种系统总线？
3. 在系统总线上完成一次数据传输要经历哪几个阶段？
4. 什么叫内总线？什么叫外总线？各写出两个常用总线的名字。

#### 【解】

1. 采用标准总线结构组成微机系统可以简化系统设计，简化系统结构、提高系统可靠性，便于系统的扩充和更新。
2. 有 S-100、MULTI bus I、MULTI bus II、STD bus、VME bus 和 FUTURE bus。
3. 分为 4 个阶段：申请占用总线阶段、寻址阶段、传送阶段和结束阶段。
4. 内总线就是系统总线，指微机机箱内的底板总线，用来连接各插件板。如 MULTIBUS、STD、VME 等。外总线即通信总线，用于微机与系统之间、微机与外设之间的通信通道。如 IEEE-488、VXI、RS-232C、Centronics 等。

## 第 9 章

## 可编程接口芯片及其与 CPU 的接口

### 9.1 重点内容提要

#### (一) 可编程并行输入/输出接口芯片 8255A

##### 1. 8255A 的基本结构

8255A 内部由 4 部分组成：并行输入/输出端口、A 组和 B 组控制部件、数据总线缓冲器和读/写控制电路。

并行输入/输出端口有 3 个，分别是 A 口、B 口和 C 口。这 3 个端口均为 8 位的端口，在使用中，A 口和 B 口常常作为独立的输入端口或输出端口，端口 C 则配合 A 口和 B 口工作，为它们提供控制信号和状态信号。C 口可分成两个独立的 4 位端口来使用。

3 个端口分成两组，一组由 A 口和 C 口高 4 位组成。另一组由 B 口和 C 口低 4 位组成。分别由 A 组控制部件和 B 组控制部件来控制，根据它们来定义各端口的操作方式。

数据总线缓冲器是一个 8 位的双向三态的缓冲器，它是 8255A 和 CPU 之间的数据接口。

读/写控制电路能接收 CPU 的控制命令，并根据命令向片内各功能部件发出操作命令。

##### 2. 8255A 的外部引脚

8255A 是一个有 40 条引脚的并行接口电路。除电源和地线外，外部引脚可分为两组。

与外设联系的引脚：A 口有  $PA_0 \sim PA_7$  八个引脚，同样 B 口和 C 口均有  $PB_0 \sim PB_7$ 、 $PC_0 \sim PC_7$ ，八个引脚。

与 CPU 联系的引脚：数据和命令通道线  $D_0 \sim D_7$ ，读/写控制引脚  $\overline{RD}$ 、 $\overline{WR}$ 、 $\overline{CS}$ 、 $A_0$ 、 $A_1$ 、RESET。

##### 3. 8255A 端口地址的选通

8255A 提供给用户的端口地址选通线有片选信号  $\overline{CS}$  和地址输入线  $A_0 \sim A_7$ ，以选通 8255A 内部的 4 个数据寄存器（端口 A、B、C 和控制寄存器）。4 个端口地址分配如下所示。

$\overline{CS}$	$A_1$	$A_0$	端口
0	0	0	A 口地址
0	0	1	B 口地址
0	1	0	C 口地址
0	1	1	控制端口地址

#### 4. 8255A 的工作方式

8255A 有方式 0 (基本 I/O 方式)、方式 1 (选通 I/O 方式) 和方式 2 (双向方式) 3 种工作方式。在使用时应注意: ①A 口有 3 种工作方式, 可以工作在其中任一种工作方式。B 口有两种工作方式, 可工作在方式 0 或方式 1。C 口只有一种工作方式, 只能工作在方式 0。②在方式 0 下, 可以不需要“联络”信号, 当需要“联络”信号时, 可由用户定义 C 口的任意位为“联络”信号, CPU 和外设之间的通信可以是同步方式或异步查询方式。当工作于方式 1 或方式 2 时, C 口的某些固定位用做“联络”信号。CPU 和外设之间的通信可以是异步查询方式或中断方式。③若采用中断方式与 CPU 交换信息时, 一定要将 8255A 片内的相应中断允许 INTE 置“1”。注意, 当用 C 口的按位置“1”/清“0”控制字来设置 C 口对应位时, 只影响 INTE 的状态, 不影响对应的引脚状态。

5. 8255A 的控制字 8255A 的控制字有两个, 方式控制字和 C 口置“1”/清“0”控制字。两个控制字以最高位  $D_7$  的状态作为标志。8255A 内部仅有一个控制寄存器, 当仅选择一个端口时, 其他端口的控制字也确定下来, 故设置控制字时务必小心。C 口置“1”/清“0”控制字, 可使端口 C 工作于位控方式, 但 C 口却是以字节形式读/写的。

### (二) 可编程定时器/计数器芯片 8253

1. 8253 的内部结构 8253 的基本结构包括计数器、数据总线缓冲器、读/写控制逻辑及控制字寄存器。

8253 内部有 3 个独立的计数器, 3 个计数器完全相同。每个计数器都有 3 个内部部件 (计数值锁存器、递减计数器和输出锁存器), 以及 3 个外部引脚 (CLK、GATE 和 OUT)。

数据总线缓冲器是 8253 用于和 CPU 数据总线接口的 8 位双向三态缓冲器。

读写控制逻辑接受来自 CPU 的读写控制信号, 以决定 3 个计数器和控制字寄存器中哪一个进行工作, 并控制内部总线上数据传送的方向。

控制字寄存器只写不读, 用于存放 8253 可编程控制字, 每个计数器都有一个独立的控制字寄存器, 但共用一个公用地址, 对哪一个计数器进行控制字的写入要通过控制字格式中的最高两位的状态来决定。

#### 2. 8253 的功能

8253 具有计数和定时功能。8253 不论作为计数器和定时器, 其本质都是减 1 计数, 是对 CLK 脉冲进行减 1 计数。两者的差别是: 作为计数器时, 在减到“0”以后, 输出一个信号便结束; 而作为定时器时, 将按定时常数不断地输出为 CLK 信号周期整数倍的定时间隔。计数仅仅是数脉冲的个数; 而定时是把计数值与计数脉冲周期联系了起来, 即  $T = n \times T_c$  ( $T$  为定时时间,  $n$  为计数值,  $T_c$  为计数脉冲周期)。

#### 3. 8253 的编程

8253 在工作之前, 必须确定每个计数器的工作方式及各自的计数初值。注意对计数器赋计数初值之前必须先写控制字, 在设置计数初值时, 要符合控制字中的格式规定。

编程命令有读出和写入两种。读出是用来读计数器的当前值; 写入命令有 3 个, 除写控制字和设定计数初值外, 还有锁存命令。在读计数器的计数值之前, 必须先用锁存

命令将当前计数值锁存到输出锁存器中，这样不会影响计数器的正常计数。

#### 4. 8253 几种工作方式的比较

8253 共有 6 种工作方式。

(1) 方式 0 和方式 4 都是软件触发计数，无自动重装能力，除非重写初值。门控信号 GATE 用于开始计数控制，高电平时正常计数、低电平时暂停计数。

两种方式的区别主要在于 OUT 的输出波形。方式 0 的 OUT 在计数过程中输出为低电平，计数结束时变为高电平。方式 4 的 OUT 在计数过程中保持高电平，而在计数结束时输出单个 CLK 脉宽的负脉冲。

(2) 方式 1 和方式 5 均是硬件触发计数。写入初值之后并不马上计数，必须在门控信号 GATE 的触发下，才开始计数。GATE 信号只是上升沿起作用。

两种方式的区别也在于 OUT 的输出波形。方式 1 在计数过程中 OUT 输出低电平并维持  $n$  个 CLK 脉冲，计数结束后变高。方式 5 在计数过程中 OUT 输出高电平，计数结束后输出 1 个 CLK 脉宽的负脉冲。

(3) 方式 2 和方式 3 共同的特点是具有自动重装入初值的能力，即当计数结束时，计数器又继续重新开始计数。这两种方式的输出都是连续波形。

两种方式的区别在于：每当计数器减到 1 时，方式 2 输出一个 CLK 脉宽的负脉冲，方式 3 则输出方波信号。若计数初值  $n$  是偶数时，则输出的 OUT 是对称方波；若  $n$  是奇数时，则输出的 OUT 在  $(n+1)/2$  计数期间保持为高电平，而在  $(n-1)/2$  计数期间保持为低电平。

### (三) 可编程串行输入/输出接口芯片 8251A

#### 1. 串行通信的一般概念

串行通信是计算机与外围设备之间传递信息的一种方式，采用这种方式可远距离传递信息。计算机要实现串行数据传送，在发送端要进行数据的并-串转换，在接收端必须把数据进行串-并转换。传递过程有异步和同步两种方式。异步传送，数据按字符方式进行传送，字符的有效长度可以是 5、6、7、8 位。为能识别字符还需要加上起始位和停止位。为保证数据传输的正确性，很多场合还须加上奇偶校验位。这种方式容易实现，但有效数据率较低。同步传送，数据按块组合，数据块起始位采用两个同步字符。显然有效数据率较高，但收发双方的时钟必须严格同步，控制电路复杂。在串行通信中常用波特率来表示数据传递的速率。串行数据传递线路的方式有单工、半双工和全双工 3 种。

#### 2. 8251A 的基本功能

8251A 是一个可完成串-并、并-串转换，以及设置传送工作方式、检测通信错误及决定传送速率等工作的接口芯片。它的基本功能包括：①可用于异步或同步串行数据传送，传送方式可由编程控制字决定；②异步传送时，可传送 5~8 位字符，时钟频率为传送波特率的 1、16、64 倍，可产生 1、1.5、2 位的停止位；③同步传送时，可选择内同步和外同步，并能自动插入同步字符；④能对奇偶错、帧错、溢出错等通信错误进行检测；⑤能以全双工的方式进行通信，发送器和接收器都是双缓冲的；⑥异步时传送波特率为 DC19.2K，同步时为 DC64K。

### 3. 8251A 的内部结构

8251A 内部由 5 大部分组成,即数据总线缓冲器、读/写控制逻辑、发送器、接收器及调制/解调控制电路。

数据总线缓冲器采用双向三态结构,是 8251A 内部所有数据、控制字及状态信息与 CPU 进行传输的通道,通过  $D_7 \sim D_0$  与 CPU 的数据总线相连。

发送器由发送缓冲器和发送控制电路两部分组成。它的功能是接收 CPU 传送来的并行数据,进行并-串转换,经移位寄存器逐位串行输出。

接收器由接收缓冲器和接收控制电路组成。从外部接收的串行数据逐位进入接收移位寄存器中,待一组数据接收完毕,将移位寄存器中的数据并行置入接收数据缓冲器中,等待向 CPU 传送。

读/写控制逻辑接收来自 CPU 的控制信号。从而确定 8251A 处于什么状态,并对数据在内部总线的传输方向进行控制。

在进行远距离通信时,一般采用模拟信号传送。调制/解调控制电路是 8251A 将数据输出端的数字信号调制成模拟信号、将数据接收端的模拟信号解调为数字信号的接口电路。

### 4. 8251A 的编程

8251A 规定了 3 种控制字:方式选择控制字、操作命令控制字和状态控制字。

方式选择控制字规定了 8251A 的工作方式。需要注意的是,方式选择控制字仅仅对 8251A 的工作方式做了规定,并不能使其启动工作。因此在方式字后必须写入一个命令字,使 8251A 处于运行状态,以便接收和发送数据。命令控制字对是否允许发送和接收、是否复位、是否搜索同步字及相关引脚的状态都做了明确规定。方式选择控制字和操作命令控制字是在 8251A 初始化编程时必须完成的,否则 8251A 将不工作。为了监视和检查 8251A 的工作状态,在 8251A 工作过程中,CPU 可以用 IN 指令读取当前 8251A 的状态控制字,从中获取一些状态信号。

## 9.2 常考题型范例精解

**例 9.1** 设某系统中有两片 8255A 芯片,由 74LS138 译码器产生两个芯片的片选信号,如图 9.1 所示。要求:第一片 8255A ( $J_1$ ) 的 A 口工作在方式 0 输出,B 口工作在方式 0 输入,C 口高 4 位为输出,低 4 位为输入。第二片 8255A ( $J_2$ ) 的 A 口为方式 0 输入,B 口为方式 1 输出,C 口高 4 位输出,C 口低 4 位为 B 口控制信号。

- (1) 试指出两片 8255A 芯片各自的端口地址。
- (2) 试写出两片 8255A 芯片各自的方式控制字。
- (3) 试写出两片 8255A 芯片各自的初始化程序。

**【解】**

- (1) 根据译码电路图,可分析出:

$J_1$  的 A、B、C 和控制端口的地址分别为 0138H、013AH、013CH、013EH。

$J_2$  的 A、B、C 和控制端口的地址分别为 0130H、0132H、0134H、0136H。

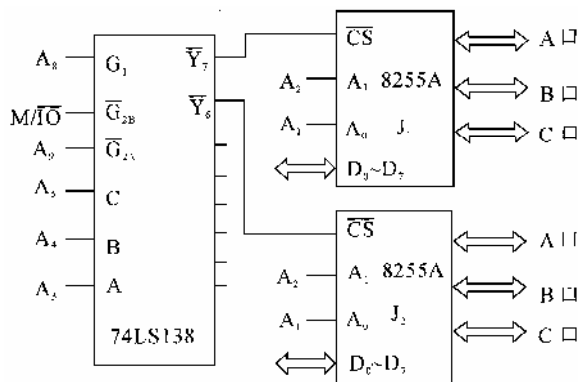


图 9.1 译码电路图

(2) 按照 8255A 方式控制字的格式，两片 8255A 的控制字分别为：

J<sub>1</sub>: 10000011B                      J<sub>2</sub>: 10010100B 或 10010101B

(3) J<sub>1</sub> 的初始化程序为：      J<sub>2</sub> 的初始化程序为：

```
MOV DX, 013EH                      MOV DX, 0136H
MOV AL, 83H                        MOV AL, 94H (或 95H)
OUT DX, AL                         OUT DX, AL
```

【评注】 注意 8255A 方式控制字的标志为 D<sub>7</sub>=1。控制字要写入控制端口，这里由于端口地址为 16 位，输出指令中必须采用 DX 间接寻址。

**例 9.2** 8253 是计数器/定时器。试问计数和定时的区别是什么？8253 各种工作方式在 OUT 输出特性及 GATE 信号上有什么区别？

【解】 作为计数器时，在减到“0”以后，输出一个信号便结束；而作为定时器时，将按定时常数不断地输出为 CLK 信号周期整数倍的定时间隔。计数仅仅是数脉冲的个数；而定时是把计数值与计数脉冲周期联系了起来。也就是说，定时和输入的 CLK 脉冲的频率（周期）有关。而计数和 CLK 频率无关。

8253 各种工作方式在 OUT 输出特性及 GATE 信号上的区别如表 9.1 所示。

表 9.1 8253 各种工作方式在 OUT 输出特性及 GATE 信号上的区别

	OUT	GATE
方式 0	一次性负方波	电平控制
方式 1	一次性负方波	脉冲触发
方式 2	周期性负脉冲	电平控制
方式 3	周期性方波	电平控制
方式 4	一次性负脉冲	电平控制
方式 5	一次性负脉冲	脉冲触发

**例 9.3** 某系统采用串行异步方式与外设通信，发送字符格式由 1 位起始位、7 位数据位、1 位奇偶校验位和 2 位停止位组成，波特率为 2 200 bps。试问，该系统每分钟发送多少个字符？若选波特率因子为 16，问发时钟频率为多少？



## 【解】

(1) 因为发送字符格式由 1 位起始位、7 位数据位、1 位奇偶校验位和 2 位停止位组成，所以一帧数据长度为 11 位，而波特率为 2 200bps，由此计算出该系统每秒钟发送  $2\,200 \div 11 = 200$  字符，每分钟为  $200 \times 60 = 12\,000$  字符。

(2) 发送时钟频率为  $2\,200 \times 16 = 35\,200\text{Hz}$ 。

例 9.4 图 9.2 所示为一利用 8255A 实现的打印机接口。

(1) 设 8255A 的 A 口工作在方式 0 输入，B 口工作在方式 0 输出。写出初始化程序片段。

(2) 写出查询方式下输出一个字符至打印机的程序片段（假设输出字符已经在 AL 中）。

【解】 根据接口图可知 8255A 的地址为 200H~203H

(1) 8255A 初始化程序为：

```
MOV DX,203H
MOV AL,10010001B
OUT DX,AI
```

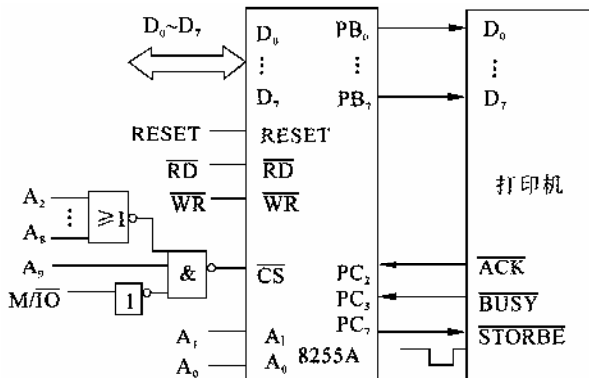


图 9.2 打印机接口图

(2) 打印控制程序为：

```
PUSH AX                ;暂存 AL 中的打印字符
MOV DX,202H
WAIT: IN AL,DX          ;读 C 口内容
TEST AL,08H            ;测 BUSY 信号
JZ WAIT                ;打印机忙,则等待
MOV DX,201H
POP AX
OUT DX,AL              ;打印字符输出到 B 口
MOV DX,203H            ;从 PC7 发一个负脉冲,将
MOV AL,0EH              ;打印字符锁入打印机
OUT DX,AL
```

```
MOV AL,0FH
OUT DX,AL
```

【评注】 注意控制打印机的工作流程。首先判断打印机是否忙，若忙继续检测，否则就可以建立数据并发出选通，将数据锁入打印机。其中输入锁存信号为一负脉冲，由软件产生，注意发脉冲信号的编程方法。

例 9.5 接口原理图如图 9.3 所示。输入机启动一次输入一个数据；打印机启动一次打印一个数据。要求从输入机输入一个数并存入存储器中，然后再由打印机将该数打印出来。一共输入 10 个数，打印 10 个数。

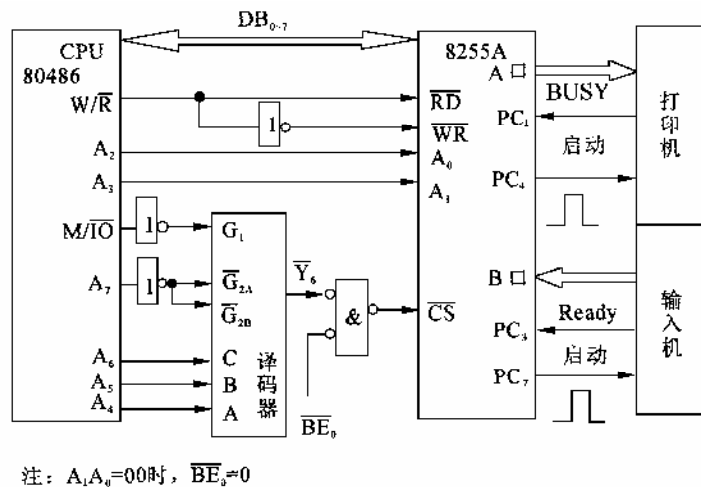


图 9.3 接口原理图

编写出满足上述要求的源程序。

【解】 由接口图分析可知，8255A 的端口地址为 E0H、E4H、E8H、ECH。程序如下：

```
.MODEL SMALL
.DATA
WP DB 10 DUP(?)
.CODE
.STARTUP
MOV AL,83H
OUT 0ECH,AL ;初始化 8255A
MOV CX,10 ;循环次数送 CX
MOV BX,OFFSET WP
AGAIN: MOV AL,00001111B
OUT 0ECH,AL ;启动输入机
PRT: IN AL,0E8H
TEST AL,08H ;检测 PC3,判断输入数据是否准备好
JZ PRT ;若无,继续检测等待
```

```

        IN AL,0E4H                ;否则,读 B E1 数据
        MOV[BX],AL                ;送入内存单元
        MOV AL,00001110B
        OUT 0ECH,AL                ;关闭输入机
PST:    IN AL,0E8H
        TEST AL,02H                ;检测 PCI,判断打印机忙否
        JNZ PST                    ;若忙,则继续检测等待
        MOV AL,[BX]
        OUT 0E0H,AL                ;否则,存入内存的数据送入 A 口
        MOV AL,00001001B
        OUT 0ECH,AL                ;启动打印机
        MOV AL,00001000B
        OUT 0ECH,AL                ;断开打印机
        INC BX
        LOOP AGAIN
        .EXIT
    END
    
```

【评注】 注意控制输入机的工作流程。首先启动输入机，然后判断输入机是否将数据准备好，如果没有准备好，则继续循环检测，否则读取数据。打印机的工作流程同上题。

例 9.6 接口原理图如图 9.4 所示，设：8259A 边沿触发，8 个中断类型码为 60~67H。写出对以内存 DSR 开始连续 10 个字节数进行打印的汇编程序片段。

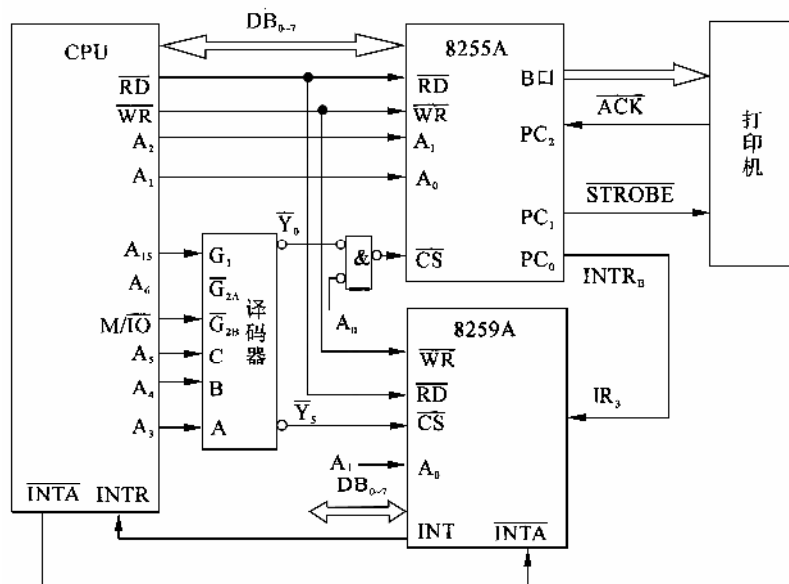


图 9.4 接口原理图

注意：未用的地址线可设为 0。

【解】 根据接口图，8255A 的 A、B、C 控制口地址分别为 8000H、8002H、8004H、8006H，B 口工作在方式 1。8259A 的地址为 8028H 和 802AH。

```
        ; 初始化 8259A
MOV DX,8028H
MOV AL,13H
OUT DX,AL          ; ICW1
MOV DX,802AH
MOV AL,60H
OUT DX,AL          ; ICW2
MOV AL,03H
OUT DX,AL          ; ICW4
        ; 建立中断向量表
PUSH DS
MOV AX,0
MOV DS,AX
MOV SI,OFFSET XY
MOV DI,SEG XY
MOV [018CH],SI
MOV [018EH],DI
POP DS
        ; 初始化 8255A
MOV DX,8006H
MOV AL,84H
OUT DX,AL
MOV AL,05H
OUT DX,AL          : INTE 置 1
        ; 循环 10 次
MOV CX,10
LEA BX,DSR
W:      HLT
        LOOP W
        ... ..
        ; 中断服务程序
XY:     MOV AL,[BX]
        MOV DX,8002H
        OUT DX,AL
        INC BX
        IRET
```

【评注】 由于中断源从 8259A 的  $IR_3$  接入，故该中断源的中断类型码为 63H。由此计算出中断向量地址为  $63H \times 4 = 018CH$

**例 9.7** 在某系统中。8253 芯片的计数器 0、计数器 1、计数器 2 及控制端口地址分别为 70H、71H、72H、73H。若利用计数器 0 对外部事件计数，其 GATE 接高电平，当计数计满 3 000 次，向 CPU 发出中断申请；且利用计数器 1 输出频率为 100Hz 的方波， $\text{CLK}_1=2\text{MHz}$ 。试编写 8253 的初始化程序。

**【解】** 根据题意分析，计数器 0 工作在方式 0，计数置 3 000 次；计数器 1 工作在方式 3，计数值  $n=20\ 00=4\text{E}20\text{H}$  次。

；8253 初始化程序片段

```
MOV AL, 21H
OUT 73H, AL           ; 设置计数器 0 的控制字
MOV AL, 30H
OUT 70H, AL           ; 给计数器 0 送计数初值
MOV AL, 76H
OUT 73H, AL           ; 设置计数器 1 的控制字
MOV AL, 20H
OUT 71H, AL
MOV AL, 4EH
OUT 71H, AL           ; 给计数器 1 送计数初值
```

**【评注】** 对每一个计数器，其控制字必须写在它的计数值之前。计数值写入计数器的格式，必须按照控制字的  $\text{RL}_1$  和  $\text{RL}_0$  的规定格式来写。

**例 9.8** 接口原理图如图 9.5 所示。要求发光二极管  $L_0$  亮 5s 后就熄灭； $L_1$  在  $K_1$ ，启动后亮 6s 后就熄灭； $L_2$  亮 4s 灭 4s 交替进行。写出源程序片段（可以省略段定义语句）。

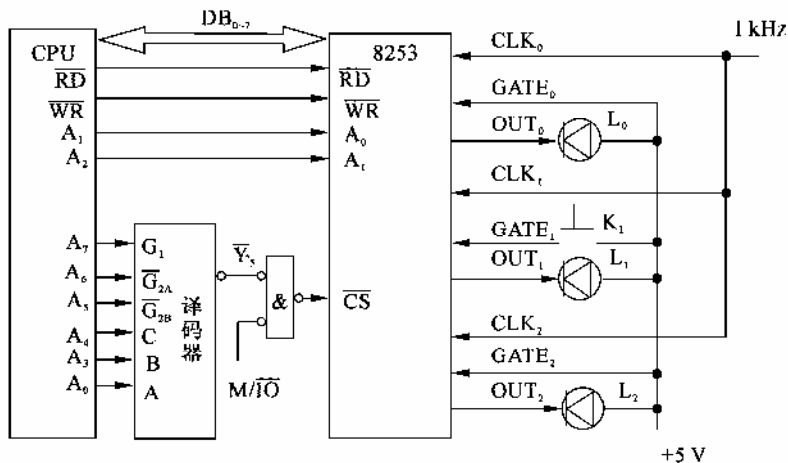


图 9.5 接口原理图

**【解】** 由译码电路分析可知，8253 芯片计数器 0、计数器 1、计数器 2 及控制端口的地址分别为 91H、93H、95H、97H。根据接口原理图及题目要求，8253 的计数器 0 应该工作在方式 0，计数初值  $N_0=5\ 000$ ；计数器 1 应工作在方式 1， $N_1=6\ 000$ ；计数器 2 应工作在方式 3， $N_2=8\ 000$ 。

根据以上分析，可以编写出 8253 初始化程序。

8253 初始化程序片段如下所示。

```
, 初始化计数器 0
MOV AL, 21H
OUT 97H, AL
MOV AL, 50H
OUT 91H, AL
, 初始化计数器 1
MOV AL, 63H
OUT 97H, AL
MOV AL, 60H
OUT 93H, AL
, 初始化计数器 2
MOV AL, 0A7H
OUT 97H, AL
MOV AL, 80H
OUT 95H, AL
```

**例 9.9** 设 8251A 为异步工作方式，波特率因子为 16，7 位字符，奇校验，两位停止位，CPU 对 8251A 输入 80 个字符，试对其进行初始化编程，设 8251A 的控制口地址为 0F2H，数据口地址为 0F0H。

**【解】** 初始化程序片段为：

```
MOV AL, 0DAH          ; 方式选择字
OUT 0F2H, AL
MOV AL, 35H           ; 命令控制字, 启动发送器和接收器, 清除出错标志
OUT 0F2H, AL
MOV DI, 0
MOV CX, 80
WAIT: IN AL, 0F2H      : 读状态字, 并测试 RXRDY 是否为 1, 当为 0 时等待
TEST AL, 02H
JZ WAIT
IN AL, 0F0H           ; 输入字符
MOV LINE[DI], AL      ; 存入内存
INC DI
IN AL, 0F2H
TEST AL, 38H          ; 测试各种错误
JNZ ERROR             ; 若出错, 则至错误处理
LOOP WAIT
ERROR: CALL ERR-OUT    ; 调出错处理程序
```

**例 9.10** 8086 CPU 通过 8251A 与电传打字机相连，采用中断、异步方式工作，波特率因子为 64，字符长度 7 位，偶校验，两位停止位。试对其初始化编程。设 8251A 的

控制口地址为 0F1H, 数据口地址为 0F0H。

【解】 8251A 的初始化程序片段为:

```
MOV AL,0FBH
OUT 0F1H,AL
MOV AL,15H
OUT 0F1H,AL
```

8251A 在初始化编程后做好了运行准备, 可接收和发送数据。8251A 的 TxRDY 和 RxRDY 引脚通过或门经 8259A 向 8086 CPU 发出中断请求信号, 如果条件满足, 则 CPU 进入中断服务程序。其过程如下所示。

```
PUSH AX
IN AL,0F1H          ; 读入状态
TEST AL,01H
JNZ SEND            ; TxRDY=1 转发送
TEST AL,02H
JNZ RECE            ; RxRDY=1 转接收
REN: POP
IRET
RECE: IN AL,0F0H
TEST AL,38H         ; 检查有无传送错误
JNZ ERROR           ; 若出错,则至出错处理程序
IN AL,0F0H
MOV BL,AL
JMP REN
SEND: MOV AL,BL      ; 要输出的数据在 BL 中
OUT 0F0H,AL
JMP REN
```

## 9.3 习题及全解

### 一、填空题

1. 8255A 是一个\_\_\_\_\_接口芯片。
2. 8255A 的内部包括两组控制电路, 其中 A 组控制\_\_\_\_\_, B 组控制\_\_\_\_\_。
3. 8255A 控制字的最高位为\_\_\_\_\_时, 表示该控制字为方式控制字。
4. 8255A 端口 C 的按位置位复位功能是由控制字中最高位为\_\_\_\_\_来决定的。
5. 8255A 的端口 A 工作在方式 2 时, 使用端口 C 的\_\_\_\_\_作为与 CPU 和外设的联络信号。
6. 某一测控系统要使用一个连续的方波信号, 如果使用 8253 可编程定时/计数器来实现此功能, 则 8253 应工作在方式\_\_\_\_\_。

7. 异步串行通信规程规定, 传送数据的基本单位是\_\_\_\_\_, 其中最先传送的是\_\_\_\_\_, 信号电平为\_\_\_\_\_。
8. 8251A 工作在异步方式时, 每个字符的数据位长度为\_\_\_\_\_, 停止位的长度为\_\_\_\_\_。
9. 8251A 从串行输入线上接收好了一个字符后, 将信号\_\_\_\_\_置为有效。
10. 8255A 内部具有\_\_\_\_\_个输入输出端口, 每个端口的数据寄存器长度为\_\_\_\_\_位, 其中端口\_\_\_\_\_没有输入锁存功能。
11. 8255A 与 CPU 连接时, 地址线一般与 CPU 的地址总线的\_\_\_\_\_连接。
12. 8255A 的端口 C 按位置位复位控制字的\_\_\_\_\_位用来指定要置位或复位的端口 C 的具体位置。
13. 异步串行通信规定, 传送的每个字符的最后是\_\_\_\_\_位, 其宽度为\_\_\_\_\_, 信号电平为\_\_\_\_\_。
14. 同步串行通信包括\_\_\_\_\_和\_\_\_\_\_两类。
15. 当 8251A 工作在同步方式时, 引脚同步检测 SYNDET 可作为输入或输出信号使用。若工作在外同步方式, 该引脚为\_\_\_\_\_, 若工作在内同步方式, 该引脚为\_\_\_\_\_。
16. 如果 8251A 设定为异步通信方式, 发送器时钟输入端和接收器时钟输入端都连接到频率为 19.2kHz 的输入信号, 波特率为 1 200bps, 字符数据长度为 7 位, 1 位停止位, 采用偶校验。则 8251A 的方式控制字为\_\_\_\_\_。
17. 8255A 工作在方式 1 或方式 2 时, INTE 为\_\_\_\_\_, 它的置 1/清 0 由\_\_\_\_\_进行控制。
18. 利用 8253 芯片每 25ms 产生一个方波输出, 若输入的时钟频率为 2MHz, 8253 的计数初值应取\_\_\_\_\_, 工作方式选\_\_\_\_\_。
19. 在串行通信中波特率是指\_\_\_\_\_。
20. 若 8255A 的 A 口工作在方式 1, 那么 CPU 通过它与外设通信可以采用\_\_\_\_\_传送方式或\_\_\_\_\_传送方式或\_\_\_\_\_传送方式。
21. 利用 8253 芯片每 20ms 产生一个脉冲输出, 若输入的时钟频率为 2MHz, 8253 的计数初值应该取\_\_\_\_\_, 工作方式应选择\_\_\_\_\_。
22. 8255A 的 A 口工作方式有\_\_\_\_\_, B 口的工作方式有\_\_\_\_\_。

#### 【解】

1. 可编程的通用并行输入输出
2. 端口 A 和端口 C 的上半部分; 端口 B 和端口 C 的下半部分
3. 1
4. 0
5. PC<sub>3</sub>~PC<sub>7</sub>
6. 3
7. 字符; 起始位; 低电平
8. 5~8 位; 1 位或 1.5 位或 2 位



9. RxRDY
10. 3; 8; C
11.  $A_0$  和  $A_1$
12. D3、D2、D1
13. 停止; 1 位或 1.5 位或 2 位; 高电平
14. 面向字符型; 面向比特型
15. 输入; 输出
16. 01111010B
17. 中断允许位; C 口按位置位清零控制字
18. 50 000 次; 方式 3
19. 每秒钟传送的二进制位数
20. 同步; 查询(异步); 中断
21. 40 000 次; 方式 2。
22. 方式 0、方式 1 和方式 2; 方式 0 和方式 1

## 二、选择题

1. 在数据传送过程中, 数据由串行变为并行, 或由并行变为串行, 这种转换是通过接口电路中的\_\_\_\_\_实现的。
  - A. 数据寄存器
  - B. 移位寄存器
  - C. 锁存器
  - D. 缓冲器
2. 8255A 在方式 0 工作时, 端口 A、端口 B 和端口 C 的输入输出有\_\_\_\_\_种组合。
  - A. 4
  - B. 8
  - C. 16
  - D. 6
3. 8255A 工作在方式 1 时, 端口 A 和端口 B 作为数据输入输出使用, 而端口 C 的各位分别作为端口 A 和端口 B 的控制信息和状态信息。其中作为端口 A 和端口 B 的中断请求信号的分别是端口 C 的\_\_\_\_\_。
  - A.  $PC_4$  和  $PC_2$
  - B.  $PC_5$  和  $PC_1$
  - C.  $PC_6$  和  $PC_7$
  - D.  $PC_3$  和  $PC_0$
4. 8255A 的端口 A 或端口 B 工作在方式 1 输入时, 端口与外设的联络信号有\_\_\_\_\_。
  - A. 选通输入  $\overline{STB}$
  - B. 中断请求信号 INTR
  - C. 中断允许信号 INTE
  - D. 输入缓冲器满信号 IBF
5. 当 8255A 的端口 A 和端口 B 都工作在方式 1 输入时, 端口 C 的  $PC_7$  和  $PC_6$ \_\_\_\_\_。
  - A. 被禁止使用
  - B. 只能作为输入使用
  - C. 只能作为输出使用
  - D. 可以设定为输入或输出使用
6. 8255A 的端口 A 和端口 B 工作在方式 1 输出时, 与外设的联络信号将使用\_\_\_\_\_。
  - A. INTR 信号
  - B.  $\overline{ACK}$  信号
  - C.  $\overline{OBF}$  信号
  - D. IBF 信号
7. 8255A 的端口 A 工作在方式 2 时, 如果端口 B 工作在方式 1, 则固定用做端口 B

的联络信号的端口 C 的信号是\_\_\_\_\_。

- A.  $PC_2 \sim PC_0$
- B.  $PC_6 \sim PC_4$
- C.  $PC_7 \sim PC_5$
- D.  $PC_3 \sim PC_1$

8. 如果选择波特率因子为 16, 在接收时, 采用波特率的 16 倍频率作为接收时钟, 其目的是\_\_\_\_\_。

- A. 提高取样精度
- B. 取样信号的峰值
- C. 提高接收速度
- D. 识别正确的起始位

9. 在异步串行输入/输出接口中, 实现并行数据与串行数据转换的主要功能部件是\_\_\_\_\_。

- A. 发送移位寄存器
- B. 发送数据寄存器
- C. 接收数据寄存器
- D. 接收移位寄存器

10. 在异步串行传输时, 如果规定采用 5 位数据位、偶校验和两位停止位, 在接收时, 如果收到 5 位数据和 1 位校验位后, 再接收到一位高电平信号和一位低电平信号, 其结果表示\_\_\_\_\_。

- A. 已正确接收一个字符的数据
- B. 已经开始接收下一个数据
- C. 传输中出现了奇偶错
- D. 传输中出现了帧错

11. 在异步串行输入/输出接口电路中, 发生溢出错是指\_\_\_\_\_。

- A. 发送数据寄存器的内容被新送来的数据覆盖
- B. 发送移位寄存器的内容被新送来的数据覆盖
- C. 接收数据寄存器的内容被新送来的数据覆盖
- D. 接收移位寄存器的内容被新送来的数据覆盖

12. 8251A 做好发送的准备时, 信号  $TxRDY$  有效。产生此信号有效的条件是\_\_\_\_\_。

- A. 发送缓冲器空信号  $TxEN$  有效
- B. 同步检测信号  $SYNDET$  有效
- C. 数据终端准备好信号  $\overline{DTR}$  有效
- D. 清除发送信号  $\overline{CTS}$  有效

13. 如果 8251A 设定为异步通信方式, 发送器时钟输入端和接收器时钟输入端都连接到频率为 19.2kHz 的输入信号, 波特率因子为 16, 则波特率为\_\_\_\_\_ bps。

- A. 1 200
- B. 2 400
- C. 9 600
- D. 19 200

14. 8251A 中控制发送和接收字符的速度的时钟信号是\_\_\_\_\_。

- A. CPU 的工作时钟
- B. 接收器时钟  $\overline{RxC}$
- C. 发送器时钟  $\overline{TxC}$
- D. CLK 输入

15. 如果 8251A 的方式字寄存器的地址为 2F1H, 则它的控制字寄存器的地址为\_\_\_\_\_。

- A. 2F0H
- B. 2F2H
- C. 2F3H
- D. 2F1H

16. 当 8253 可编程定时/计数器工作在\_\_\_\_\_下时, 需要由外部脉冲触发开始计数。
- A. 方式 0    B. 方式 1    C. 方式 2  
D. 方式 3    E. 方式 4    F. 方式 5
17. 若 8253 处于计数过程中, 当 CPU 对它装入新的计数初值时, 其结果将是\_\_\_\_\_。
- A. 8253 禁止编程  
B. 8253 允许编程, 并改变当前的计数初值  
C. 8253 允许编程, 但不改变当前的计数初值  
D. 8253 允许编程, 是否影响当前计数过程随工作方式不同而不同
18. 当 8253 工作在方式 0, 在初始化编程时, 一旦写入控制字后, \_\_\_\_\_
- A. 输出信号端 OUT 变为高电平  
B. 输出信号端 OUT 变为低电平  
C. 输出信号保持原来的电位值  
D. 立即开始计数
19. 当 8253 工作在方式 0 时, 控制信号 GATE 变为低电平后, 对计数器的影响是\_\_\_\_\_。
- A. 结束本次计数循环, 等待下一次计数的开始  
B. 暂时停止现行计数工作  
C. 不影响本次计数  
D. 终止本次计数过程, 立即开始新的计数循环
20. 8255A 的 A 口工作在方式 2 时, B 口\_\_\_\_\_。
- A. 可工作在方式 0 或方式 1                      B. 可工作在方式 1 或方式 2  
C. 只能工作在方式 1                              D. 只能空着
21. 利用 8253 每 20ms 产生一次中断, 若 CLK 为 2MHz, 则 8253 可采用的工作方式及所取的计数初值分别为\_\_\_\_\_。
- A. 方式 0; 65536                                  B. 方式 3; 40000H  
C. 方式 5; 40000                                  D. 方式 2; 40000
22. 8253\_\_\_\_\_。
- A. 只能实现计数                                  B. 只能实现定时  
C. 既可实现计数, 又可实现定时              D. 以上工作均不能完成
23. 8255A 用户可以用命令字设置\_\_\_\_\_
- A. A 口和 B 口均可工作在方式 0 或方式 1 或方式 2  
B. A 口工作在方式 0, B 口工作在方式 1  
C. A 口工作在方式 1, B 口工作在方式 0 或方式 1  
D. A 口工作在方式 2, B 口只能工作在方式 1
24. 8255A 能实现双向传送功能的工作方式为\_\_\_\_\_。
- A. 方式 0    B. 方式 1  
C. 方式 2    D. 不能实现

25. 当 8255A 工作在方式 1 时, 端口 C 被分为两个部分, 分别作为端口 A 和端口 B 的控制信息和状态信息。这两个部分的划分是\_\_\_\_\_。

- A. 端口 C 的高 4 位和低 4 位                      B. 端口 C 的高 5 位和低 3 位  
C. 端口 C 的高 3 位和低 5 位                      D. 端口 C 的高 6 位和低 2 位

【解】

1. B    2. C    3. D    4. A、D    5. D    6. B、C    7. A    8. D  
9. A、D    10. D    11. C    12. A、D    13. A  
14. B、C    15. D    16. B、F    17. D    18. B    19. B  
20. A    21. D    22. C    23. B、C    24. C    25. B

### 三、问答题

1. 试指出下列工作方式组合使用时, 8255A 芯片 C 端口各位的作用。

- (1) A 口工作在方式 2, B 口工作在方式 0, 输入;  
(2) A 口工作在方式 2, B 口工作在方式 1, 输入;  
(3) A 口工作在方式 2, B 口工作在方式 1, 输出;  
(4) 说明  $\overline{STB}$ 、IBF、 $\overline{OBF}$ 、 $\overline{ACK}$ 、INTE 以及 INTR 的作用。

2. 8255A 在微机系统中起什么作用? 它有几几种工作方式? 它有几个口地址? 各由什么来决定?

3. 当数据从 8255A 的端口 C 往数据总线上读出时, 8255A 的几个控制信号  $\overline{CS}$ 、 $A_1$ 、 $A_0$ 、 $\overline{RD}$ 、 $\overline{WR}$  分别是什么?

4. 8253 工作于方式 2 和方式 3 时有什么异同?

5. 当 8255A 工作在方式 2 时, CPU 是如何来区分输入或输出的?

6. 在 8253 计数器/定数器中, 时钟信号 CLK、门控信号 GATE 分别起什么作用?

7. 在串行通信中奇/偶校验的概念是什么?

8. 8255A 的 A 口工作在方式 1 输入方式和输出方式, 中断请求信号是从 C 口哪一端引出的? 中断允许信号又是如何实现的?

9. 8253 中有 3 个计数器, 用户怎么用它完成定时?

【解】

1. (1)  $PC_0 \sim PC_2$  为自由 I/O 位;  $PC_3$  为中断请求信号;  $PC_4$  和  $PC_5$  提供 A 口在输入时的联络信号;  $PC_6$  和  $PC_7$  提供 A 口在输出时的联络信号。

(2)  $PC_0$  为中断请求信号;  $PC_1$  和  $PC_2$  提供 B 口输入时的联络信号;  $PC_3$  为中断请求信号;  $PC_4$  和  $PC_5$  提供 A 口在输入时的联络信号;  $PC_6$  和  $PC_7$  提供 A 口在输出时的联络信号。

(3)  $PC_0$  为中断请求信号;  $PC_1$  和  $PC_2$  提供 B 口输出时的联络信号;  $PC_3$  为中断请求信号;  $PC_4$  和  $PC_5$  提供 A 口在输入时的联络信号;  $PC_6$  和  $PC_7$  提供 A 口在输出时的联络信号。

2. 8255A 是并行可编程接口芯片。它有 3 种工作方式: 方式 0、方式 1 和方式 2。

有 4 个口地址, 由地址译码输出线接在它的  $\overline{\text{CS}}$  上, 对应的地址值和  $A_0$ 、 $A_1$  一起决定 4 个地址。

3. 分别为 0、1、0、0、1。

4. 8253 的方式 2 和方式 3 都是对 CLK 的  $N$  分频, OUT 的初始电平一样都为高, GATE 控制也一样, 只是 OUT 输出的信号波形不同, 方式 2 的 OUT 是  $(N-1)$  个 CLK 的高电平加一个 CLK 的低电平; 而方式 3 的 OUT 是方波 ( $N$  为偶数), 高、低电平各为  $N/2$  个 CLK, 或近似方波 ( $N$  为奇数), 高电平为  $(N+1)/2$  个 CLK, 低电平为  $(N-1)/2$  个 CLK。

5. 在输入、输出均允许中断的情况下, 可在中断服务程序中查询判断  $\overline{\text{OBF}}$  和  $\overline{\text{IBF}}$  位状态, 来区分是输入中断还是输出中断, 并进行相应操作。

6. CLK 输入计数/定时器脉冲信号, 8253 就是对 CLK 端的时钟脉冲信号计数。门控信号 GATE 作为控制信号, 为高电平允许对 CLK 端脉冲计数, 低电平将停止计数, 对方式 1 和方式 5 上升沿触发启动计数, 对方式 2 和方式 3 上升触发重新初始计数。

7. 为了在串行通信长距离传送中避免发生错误, 由奇偶校验检查每个要传送的字符中“1”的个数, 根据设定的奇校验或偶校验, 自动在奇偶校验位上送“1”或“0”。在接收后, 检查字符的每一位和奇偶校验位的“1”的个数和是否满足奇或偶校验, 以确认是否发生了错误。

8. 中断请求信号是从  $\text{PC}_3$  引出的, 当 A 口工作在方式 1 的输入时, 中断允许通过  $\text{PC}_4$  置 1 实现; 当 A 口工作在方式 1 的输出时, 中断允许通过  $\text{PC}_6$  置 1 实现。

9. 利用计数器定时, 与时钟脉冲输入的频率  $f_c$  有关。若定时时间为  $T$ , 则计数器的计数值  $N=T \times f_c$

#### 四、接口设计题

1. 某系统中 8253 芯片的通道 0~通道 2 和控制端口的地址分别为 FFF0H~FFF6H, 定义通道 0 工作在方式 2,  $\text{CLK}_0=2\text{MHz}$ , 要求输出  $\text{OUT}_0$  为 1KHz 的速率波; 定义通道 1 工作在方式 0, 其  $\text{CLK}_1$  输入外部计数事件, 每计满 1 000 个向 CPU 发出中断请求。试写出 8253 通道 0 和通道 1 的初始化程序。

2. 在某微机系统与 CRT 通信中, 采用异步方式, 设波特率为 200bps, 波特率因子为 16, 利用 8253 芯片产生发送和接收时钟。设 8253 的通道 1 工作在方波方式,  $\text{CLK}_1=1.2288\text{MHz}$ , 试写出 8253 的初始化程序。

3. 8251A 工作在异步方式, 选波特率系数为 64, 其传送字符格式为: 7 位字符, 1 位奇校验, 1.5 位停止位。其端口地址要求为 2CH 和 2EH, 使用 74LS138 译码器产生端口地址, 试编写 8251A 的初始化程序, 画出端口地址形成电路。

4. 有一打印机引脚信号如图 9.6 所示。其中:

$\text{DB}_0 \sim \text{DB}_7$  打印机数据输入线。

$\overline{\text{STB}}$  输入数据选通信号。该信号有效, 则打印机接收数据。

BUSY “忙”信号输出线, 此信号有效, 表示打印机正在打印一个数据。

$\overline{\text{ACK}}$  是对  $\overline{\text{STB}}$  信号的“应答”信号输出线, 此信号有效, 表示打印机已得到数据。

$\overline{\text{ERR}}$  “出错”信号输出线，此信号有效，表示打印命令有错。

注意：只有在  $\overline{\text{ACK}}$  有效、 $\text{BUSY}$  和  $\overline{\text{ERR}}$  无效情况下，打印机才能接收新的数据进行打印。

请用 8255A 作为接口芯片，采用查询传送方式，写出连续打印 10 个字节数据的程序。

（用简化汇编语言程序写出，用 8086 CPU 或 80486 CPU 均可，画出接口图，口地址可合理假设。）

2	4	6	8	10	12	14	16	18	20
GND	GND	GND	GND	GND	GND	GND	GND	$\overline{\text{ACK}}$	$\overline{\text{ERR}}$
$\overline{\text{STB}}$	$\text{DB}_5$	$\text{DB}_1$	$\text{DB}_6$	$\text{DB}_7$	$\text{DB}_4$	$\text{DB}_5$	$\text{DB}_6$	$\text{DB}_7$	$\text{BUSY}$
1	3	5	7	9	11	13	15	17	19

图 9.6 打印机引脚信号图

5. 某微机控制系统中扩展一片 8255A 作为并行口，如图 9.7 所示。其中，A 口为方式 1 输入，以中断方式与 CPU 交换数据，中断类型为 0FH；B 口工作于方式 0 输出，C 口的普通 I/O 线作为输入。请编写 8255A 的初始化程序，并设置 A 口的中断矢量。
6. 设 8255A 的 4 个端口地址为 00C0H、00C2H、00C4H 和 00C6H，要求用置位/复位方式对 PC<sub>6</sub> 置 1，对 PC<sub>4</sub> 清 0。

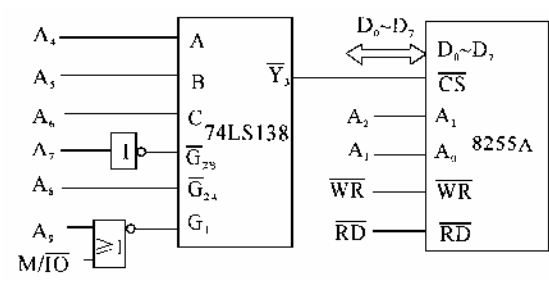


图 9.7 接口图

7. 对 8255A 设置工作方式，8255A 的控制端口地址为 00C6H。要求端口 A 工作在方式 1 输入，端口 B 工作在方式 0 输出，端口 C 的普通 I/O 线为输入。
8. 编程将 8253 的计数器 0 设置为方式 1，计数初值为 3000H；计数器 1 设置为方式 2，计数初值为 2010H；计数器 2 设置为方式 4，计数初值为 50，8253 的端口地址为 3B0H~3B3H。
9. 用 8255A 作为 A/D 并行接口，如图 9.8 所示。8255A 的 A 口工作于方式 1，为输入口，端口 C 的 PC<sub>7</sub> 位设定为输出端，与 A/D 转换器的转换启动信号相连，由 PC<sub>7</sub> 端产生正脉冲信号以启动 A/D 转换。A/D 的转换结束端经反相后作为采样保持器的控制信号，并且转换结束端经下降沿触发单稳电路，再经反相输出一个负方波的波形到 PC<sub>4</sub> 端，作为端口 A 的数据输入锁存信号，将 A/D 转换结果存入 A 口的数据输入寄存器。假设 8255A 的 A、B、C 和控制寄存器的 I/O 地址分别为 300H、301H、302H、303H。

写出 8255A 的初始化程序及启动 A/D 转换的程序片段。

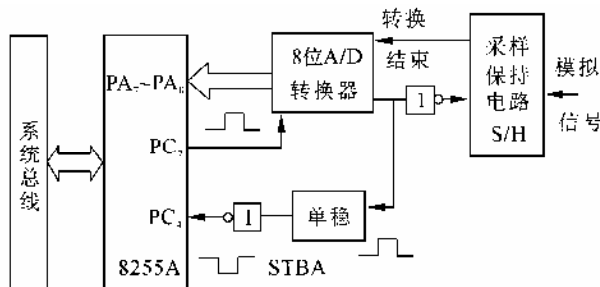


图 9.8 接口图

10. 用程序段对 8251A 进行同步方式设置。端口地址为 66H, 写出用内同步方式, 同步字符为 2 个, 用奇校验, 7 个数据位。

11. 若 8251A 的收发时钟 (RxC 和 TxC) 频率为 38.4kHz, 它的  $\overline{\text{RTS}}$  和  $\overline{\text{CTS}}$  引脚相连, 试编写出以半双工异步通信, 每帧字符的数据位数为 7, 停止位数为 1, 偶校验, 波特率为 600bps, 处于发送状态的初始化程序 (设 8251A 的地址为 02C0H 和 02C1H)。

12. 设有某微机控制系统, 采用定时器 8253 产生定时中断信号, CPU 响应中断后便执行数据采集、数字滤波和相应的控制算法, 以控制输出, 如图 9.9 所示。采用两个计数器串联的方法实现定时控制。一旦定时时间到,  $\text{OUT}_1$  信号由高变低, 经反向送 8259A 的  $\text{IR}_2$ 。  $\text{IR}_2$  的中断类型码为 0AH, 中断处理程序首地址存储在 28H~2BH。8253 端口地址为 230H~233H。试编制 8253 的初始化程序及设置中断处理程序首地址程序段。

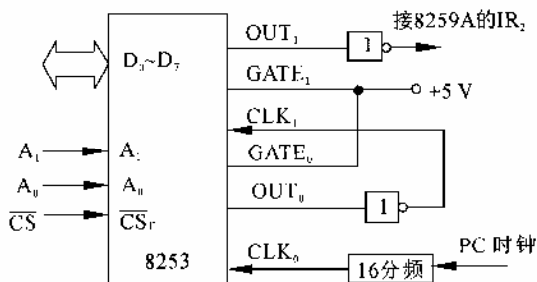


图 9.9 接口图

13. 试编程, 将 8253 计数器 0 设置为方式 1, 计数初值为 3000, 计数器 1 设置为方式 2, 计数初值为 2010H, 计数器 2 设置为方式 4, 计数初值为 4030H。

14. 利用 8253 的 2<sup>#</sup> 计数器周期性地每隔 10ms 产生一次中断, 已知 CLK 频率为 2MHz。试选择工作方式, 并编写出相应初始化程序。

15. 图 9.10 所示的为开关状态检测电路和继电器控制电路。当开关 K 闭合时, 将驱动对应的继电器 (即动作); 若开关处于断开状态, 则无电流流过继电器线圈, 继电器不动作。若系统每隔 10ms 检测一次开关状态和对继电器做相应控制, 定时控制由 8253 完

成，试编写对应 8255A 的初始化程序（初始状态时应保证继电器不动作）和完成上述功能的检测、控制程序。

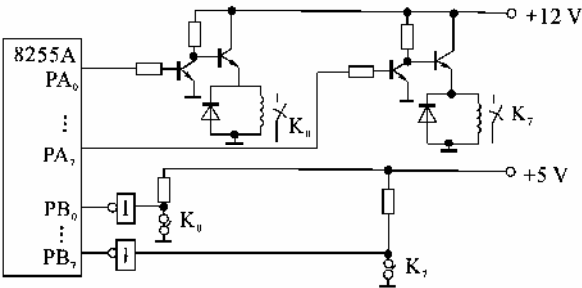


图 9.10 接口图

16. 利用 8255A 作为 CPU 与打印机的接口，硬件连线如图 9.11 所示。若要打印字符 ‘A’，试编写此接口程序（STB 为选通信号，BUSY 为忙信号）。

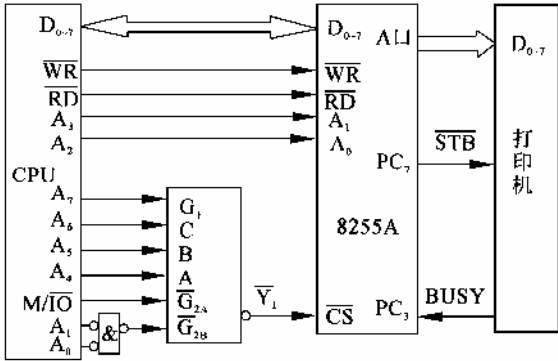


图 9.11 接口图

17. 接口原理图如图 9.12 所示。设 8255A 的 A 口地址为 50H，B 口地址为 51H，C 口地址为 52H，控制端口地址为 53H。输入机和输出机（打印机）启动信号为高电平，Ready 为输入机就绪信号；BUSY 为打印机“忙”信号。要求编写出从输入机连续输入 100 个数据并存入存储器中，再由打印机打出该 100 个数据的简化汇编源程序。

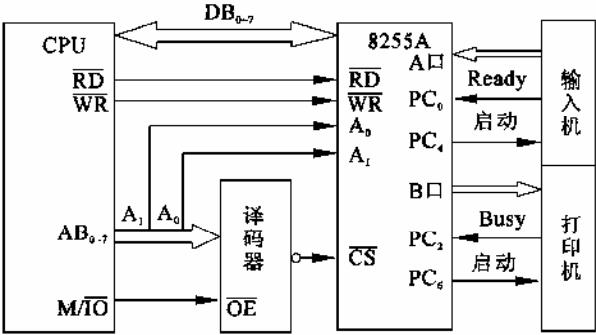
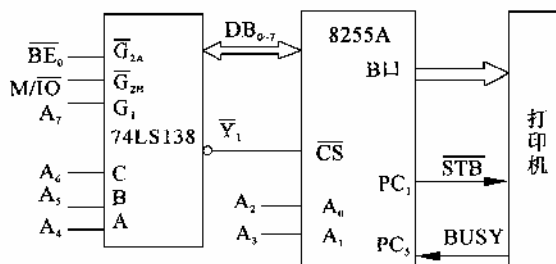


图 9.12 接口原理图



[illegible]

19. 接口连接图如图 9.14 所示, 写出向打印机输出字符 'A' 的接口程序。提示: 当 CPU 的  $A_0$  和  $A_1$  均为 0 时,  $\overline{BE}_0 = 0$  )。



20. 接口连接图如图 9.15 所示, 写出连续向打印机输送 10 个字节的程序 (提示: 当 CPU 的  $A_0$  和  $A_1$  均为 0 时,  $\overline{BE}_0 = 0$ )

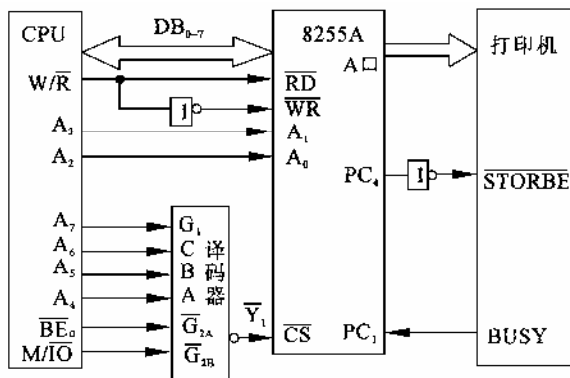


图 9.15 接口图

21. 已知 8253 芯片的  $A_0$  和  $A_1$  上分别连接到 CPU 的  $A_0$  和  $A_1$  上, 要求该芯片的 1<sup>#</sup> 计数器输出频率为 1kHz 的方波,  $CLK_1=20MHz$ , 可合理假设端口地址。试编写 8253 的初始化程序。

22. 设 8255A 的 A 口工作在方式 1, 作为中断方式下的字符打印机的接口, 此时, 若 CPU 发出控制信号通过 8255A 的 C 口向打印机发出数据锁存信号, 则 CPU 送来打印的数据被锁存到打印机等待打印。打印机收到打印数据后向 8255A 发出应答信号, 此时 8255A 就发出中断请求, 中断信号从 8259A 的  $IR_3$  引入, 中断类型为 5BH。设中断服务程序必须装配在 2000H: 2340H 处。I/O 端口地址任选。

(1) 试写出 8255A 的初始化程序。

(2) 设 8259A 以固定优先权、完全嵌套、自动结束方式工作, 试写出 8259A 的预置命令。

(3) 用直接置入法, 将中断服务程序的地址置入矢量表中。

(4) 写出中断服务程序。

(5) 画出连接线路图。

23. 设 8253 的通道 2 工作在计数方式, 外部事件从  $CLK_2$  引入, 通道 2 每计若干个脉冲向 CPU 发出中断请求, CPU 响应这一中断后继续与入计数值, 重新开始计数, 保持每 1s 向 CPU 发出中断请求。假设条件如下所示。

(1) 8253 的通道 2 工作在方式 4

(2) 外部计数事件频率为 1kHz

(3) 中断类型为 54H

(4) 8253 的通道 0~通道 2 和控制端口地址分别为 FFF0~FFF6H

(5) 用 8212 芯片产生中断类型号

试编写程序完成以上任务, 并画出硬件连接图。

【解】

1. ; 8253 通道 0 的初始化程序

```
MOV AL,00100101B
```

```
MOV DX,0FFF6H
```

```
OUT DX,AL
```

```
MOV AL,20H
```

```
MOV DX,0FFF0H
```

```
MOV DX,AL
```

; 8253 通道 1 的初始化程序

```
MOV AL,01100001B
```

```
MOV DX,0FFF6H
```

```
OUT DX,AL
```

```
MOV AL,10H
```

```
MOV DX,0FFF2H
```

```
OUT DX,AL
```

2. 依题意, 通道 1 的工作方式应选方式 3, 计数初值为:

$$1.228\ 8\ \text{MHz}/(200\times 16\text{Hz})=384$$

设 8253 地址为 70H~73H，则 8253 的初始化程序如下所示。

```
MOV AL,01110111B
OUT 73H,AL
MOV AL,84H
OUT 71H,AL
MOV AL,03H
OUT 71H,AL
```

3. ; 8251A 的初始化程序

```
MOV AL,10011011B
OUT 2EH,AL
MOV AL,15H
OUT 2EH,AL
```

端口地址形成电路如图 9.16 所示。

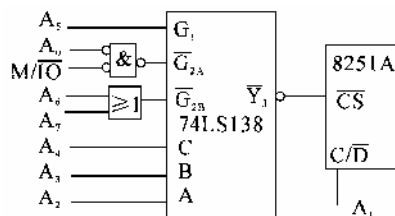


图 9.16 端口地址形成电路

4. . MODEL SMALL

. DATA

BLOCK DB 1,2,3,..... ; (10 个字节数)

. CODE

. STARTUP

MOV BX,OFFSET BLOCK

MOV AL,10000010B ; 8255A 初始化

OUT 53H,AL ; 设 8255A 的 A、B、C 及控制口地址为 50H~53H

MOV CX,10

IN AL,51H ; 读 B 口

TEST AL,04H ; 测试 ERR

JZ ERR ; 错误处理

P1: IN AL,51H ; 读 B 口

TEST AL,02H ; 测试 BUSY

JNZ P1

P2: MOV AL,51H ; 测 ACK

TEST AL,01H

JNZ P2

MOV AL,[BX] ; 送数

OUT 50H,AL

MOV AL,00001100B ; PC6=0

OUT 53H,AL

```

INC BX
LOOP P1
.EXIT
END

```

电路如图 9.17 所示。

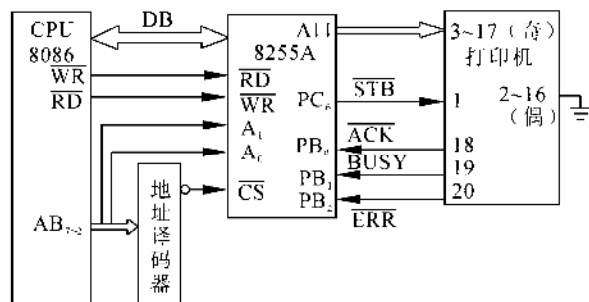


图 9.17 接口图

5. 从图 9.17 可得到 8255A 的一组地址为 00B0H、00B2H、00B4H、00B6H，或另一组地址为 00B8H、00BAH、00BCH、00BEH。

```

; 8255A 初始化程序
MOV AL,10111001B      ; 方式控制字
MOV DX,00B6H
OUT DX,AL
MOV AL,00001001B      ; PC4 置 1，开放 A 口的输入中断请求
OUT DX,AL
; 中断矢量设置程序（使用串装入法）
MOV AX,0
MOV ES,AX
MOV DI,0FH*4
MOV AX,OFFSET SERV    ; SERV 为中断服务程序
CLD
STOSW
MOV AX,SEG SERV
STOSW

```

6. 根据题意设计程序片段，如下所示。

```

MOV DX,00C6H          ; 置位/复位控制字是写入控制端口的
MOV AL,00001101B      ; PC6 置位控制字
OUT DX,AL
MOV AL,00001000B      ; PC4 复位控制字
OUT DX,AL

```

7. 初始化程序片段如下所示。

```
MOV AL,10111001B
MOV DX,00C06H
OUT DX,AL
```

8. 根据题意设计程序片段如下所示。

```
ADDR EQU 3B0H
MOV DX,ADDR+3          ; 控制寄存器地址
MOV AL,00110010B       ; 将计数器 0 设置为方式 1，二进制计数
OUT DX,AL
MOV DX,ADDR             ; 计数器 0 地址
MOV AL,0
OUT DX,AL               ; 设置 0#初值低 8 位
MOV AL,30H
OUT DX,AL               ; 设置 0#初值高 8 位
MOV DX,ADDR+3           ; 控制寄存器地址
MOV AL,01110100B       ; 1#计数器设置为方式 2，二进制计数
OUT DX,AL
MOV DX,ADDR+1           ; 1#计数器地址
MOV AL,10H
OUT DX,AL               ; 设置 0#初值低 8 位
MOV AL,20H
OUT DX,AL               ; 设置 0#初值高 8 位
MOV DX,ADDR+3           ; 控制寄存器地址
MOV AL,10011001B       ; 2#计数器设置为方式 4，十进制计数，只读/写低 8 位字节
OUT DX,AL
MOV DX,ADDR+2           ; 2#计数器地址
MOV AL,50H
OUT DX,AL               ; 设置 2#初值低 8 位
```

9.

(1) 8255A 的初始化程序如下所示。

```
MOV DX,303H
MOV AL,10110000B
OUT DX,AL
```

(2) 启动 A/D 转换器及读取转换数据。

```
MOV DX,303H
MOV AL,00001111B
OUT DX,AL
MOV AL,00001110B
OUT DX,AL          ; 送置/复位控制字，在 PC7 产生一个正
                   ; 波输出，启动 ADC

MOV DX,302H
AGAIN: IN AL,DX
TEST AL,00010000B
```

```

JZ AGAIN                ; 等待 A/D 转换结束
MOV DX,300H
IN AL,DX                ; 从 A 口中读取转换数据

```

10. 根据题意，程序段如下所示。

```

XOR AX,AX
MOV DX,66H
OUT DX,AL
OUT DX,AL
OUT DX,AL
MOV AL,40H
OUT DX,AL                ; 连续向 8251A 送 3 个 00H 及一个 40H 是为了复位
MOV AL,00011000B
OUT DX,AL                ; 送方式字
MOV AL,SYNC              ; SYNC 为同步字
OUT DX,AL
OUT DX,AL                ; 送两个同步字
MOV AL,10111111B
OUT DX,AL                ; 送控制字

```

11. 因  $R_{xC}$  和  $T_{xC}$  为 38.4kHz，而波特率为 600bps，故分频系数=38 400/600=64  
初始化程序片段如下所示。

```

XOR AX,AX
MOV DX,02C1H
OUT DX,AL
OUT DX,AL
OUT DX,AL                ; 往 8251A 的控制端口送 3 个 00H
MOV AL,40H
OUT DX,AL                ; 往 8251A 的控制端口送 40H，使它复位
MOV AL,01111011B
OUT DX,AL                ; 方式字，1 个停止位，偶校验，7 位数据，分频系数=64
MOV AL,00110001B
OUT DX,AL                ; 送控制字

```

12. 根据题意，程序段如下所示。

```

CS8253 DW 230H
TIME DB xxH              ; 计数器 1 的时间常数为 xxH
...
MOV BX,DS: CS8253
MOV DX,BX
ADD DX,03H                ; 形成控制寄存器地址
MOV AL,14H
OUT DX,AL                ; 设置计数器 0 的工作为方式 2，二进制计数

```

```

MOV DX,BX
MOV AL,0
OUT DX,AL                ; 对计数器 0 送时间常数, 0 为最大值
ADD DX,03H
MOV AL,54H
OUT DX,AL                ; 设置计数器 1 的工作方式, 同计数器 0
ADD BX,01H
MOV DX,BX
MOV AL,DX: BYTE PTR TIME
OUT DX,AL                ; 对计数器 1 送时间常数
CLI                      ; 禁止中断
MOV AX,0
MOV ES,AX                ; 置向量表段基址
MOV DI,28H               ; 置 0AH 号中断向量地址
MOV AX,OFFSET INTER      ; 中断服务程序偏移地址
CLD
STOSW
MOV AX,SEG INTER          ; 中断服务程序段基址
STOSW
STI                      ; 开中断
    
```

13. 设 8253 的地址为 70H~73H。编写程序段如下所示。

```

MOV AL,00100011B
OUT 73H,AL                ; 送计数器 0 控制字
MOV AL,30H
OUT 70,AL                 ; 送计数器 0 的计数初值
MOV AL,01110100B
OUT 73H,AL                ; 送计数器 1 的控制字
MOV AL,10H
OUT 71H,AL
MOV AL,20H
OUT 71H,AL                ; 送计数器 1 的计数初值
MOV AL,10111000B
OUT 73H,AL                ; 送计数器 2 的控制字
MOV AL,30H
OUT 72H,AL
MOV AL,40H
OUT 72H,AL                ; 送计数器 2 的计数初值
    
```

14. 设 2<sup>#</sup>计数器的地址为 72H, 控制寄存器的地址为 73H。根据题意要产生周期性的中断信号, 可选择方式 2。计数初值为:

$$n=10\text{ms}\times\text{MHz}=10\,000=4\text{E}20\text{H}$$

初始化程序片段如下所示。

```

MOV AL,10110100B
OUT 73H,AL
MOV AL,20H
OUT 72H,AL
MOV AL,4EH
OUT 72H,AL

```

15. 假设 8255A 的 A 口地址为 CS8255, 8253 的 0<sup>#</sup>计数器的地址为 CS8253。程序片段如下所示。

```

...
CLI
MOV AL,10000010B      ; 8255A 初始化
OUT CS8255+3,AL       ; 设 A, B 口均为方式 0, A 口输出, B 口输入
MOV AL,00H
OUT CS8255,AL         ; 置 A 口为全 0, 保证继电器不动作
MOV AL,00110000B      ; 8253 的初始化
OUT CS8253+3,AL       ; 设计数器 0 为方式 0, 16 位计数, 二进制方式
MOV AX,20000
OUT CS8253,AL
MOV AL,AH
OUT CS8253,AL         ; 送计数初值 20000 (设 CLK0=2MHz)
STI
...
INTCHECK: PROC NEAR   ; 中断服务程序 (定时中断检测、控制)
    PUSH AX
    IN AL,CS8255+1     ; 从 B 口读入开关状态
    OUT CS8255,AL      ; 从 A 口输出, 使相应的继电器动作
    MOV AX, 20000
    OUT CS8253,AL
    MOV AL,AH
    OUT CS8253,AL      ; 置计数初值; 重新启动定时
    POP
    IRET
INTCHECK ENDP

```

16. 分析图 9.11 可知: A 口、B 口、C 口和控制口地址分别为 90H、94H、98H 和 9CH。

接口程序如下所示。

```

MOV AL,81H
OUT 9CH,AL      ; 8255A 初始化
W: IN AL,98H    ; 读 C 口
TEST AL,08H     ; 检测 PC3, 判断打印机是否忙
JNZ W           ; 若“忙”, 则等待

```



```

MOV AL, 'A'                ; 否则, 字符“A”送 A 口
OUT 90H, AL
MOV AL, 0EH
MOV 9CH, AL
INC AL
OUT 9CH, AL                ; 选通打印机

```

17. 根据题意, 编写汇编源程序如下所示。

```

. MODEL SMALL
. DATA
BLOCK DB 100 DUP (?)
. CODE
. STARTUP
MOV AL, 10010001B
OUT 53H, AL                ; 8255 初始化
MOV CX, 64H
LEA BX, BLOCK
X:  MOV AL, 00001001B
    OUT 53H, AL            ; 启动输入机
W1:  IN AL, 52H
    TEST AL, 01H
    JZ W1                  ; 检测 Ready=0?若是, 循环检测等待, 直到 Ready=1
    IN AL, 50H             ; 若 Ready 有效, 则从 A 口读数
    MOV[BX], AL            ; 存入内存
    INC BX
    MOV AL, 08H
    OUT 53H, AL            ; PC4 清 0, 关闭输入机
    LOOP X
    MOV CX, 100
    LEA BX, BLOCK
W2:  IN AL, 52H
    TEST AL, 04H           ; 检测 Busy 信号
    JNZ W2                 ; 若忙, 继续检测
    MOV AL, [BX]
    OUT 51H, AL            ; 否则, 数据写入 B 口
    INC BX
    MOV AL, 00001101B
    OUT 53H, AL            ; 断开打印机
    MOV AL, 00001100B
    OUT 53H, AL
    LOOP W2
. EXIT
END

```

18. 根据译码电路, 8253 的端口地址为 B0H、B4H、B8H、BCH。根据题目要求, 经分析得出: 计数器 1 工作在方式 1, 计数初值 N=6 000; 计数器 2 工作在方式 3, 计数初值 N=8 000。初始化程序片段如下所示。

```

MOV AL,01100011B
OUT 0BCH,AL
MOV AL,10100111B
OUT 0BCH,AL
MOV AL,60H
OUT 0B4H,AL
MOV AL,80H
OUT 0B8H,AL

```

19. 根据接口图, 可得知 8255A 的地址为: A 口, 90H; B 口, 94H; C 口, 98H; 控制口, 9CH, 程序片段如下所示。

```

MOV AL,10001000B    ; 初始化 8255A
OUT 9CH,AL
W1:  IN AL,98H
     TEST AL,20H      ; 检测 PC5, 打印机是否忙
     JNZ W1           ; 若忙, 则继续查询
     MOV AL, 'A'      ; 否则, 字符“A”送 B 口
     OUT 94H,AL
     MOV AL,02H       ; 选通打印机
     OUT 9CH,AL
     MOV AL,03H       ; 断开打印机
     OUT 9CH,AL

```

20. 由接口图分析可知, 8255A 的地址为: A 口, 90H; B 口, 94H; C 口, 98H; 控制口, 9CH, 设 10 个字节在以 BLOCK 为首地址的内存缓冲区中, 连续向打印机输送 10 个字节的程序如下所示。

```

. MODEL SMALL
. DATA
BLOCK DB 10 DUP(?)
. CODE
. STARTUP
MOV AL,81H      ; 初始化 8255A
OUT 9CH,AL
MOV BX,OFFSET BLOCK
MOV CX,10
W:  IN AL,98H    ; 查询打印机忙否
     TEST AL,02H
     JNZ W
     MOV AL,[BX] ; 若不忙, 则当前数据送入 A 口
     OUT 90H,AL

```

```

MOV AL,09H          ; 选通打印机
OUT 9CH,AL
DEC AL              ; 断开打印机
OUT 9CH,AL
INC BX              ; 指向下一个数据
LOOP W
. EXIT
END

```

21. 假设端口地址为 70H~73H, 依题意: 1<sup>#</sup>计数器应选方式 3, n=20 000 次=4E20H 次, 初始化程序片段如下所示。

```

MOV AL,01110110B
OUT 73H,AL
MOV AL,20H
OUT 71H,AL
MOV AL,4EH
OUT 71H,AL

```

22. (1) 假设 8255A 的端口地址为 80H~83H, 初始化程序如下所示。

```

MOV AL,0A0H        ; 方式控制字
OUT 83H,AL
MOV AL,0DH          ; INTE 置 1
OUT 83H,AL

```

(2) 假设 8259A 的地址为 90H, 91H, 预置命令如下所示。

```

MOV AL,13H          ; ICW1
OUT 90H,AL
MOV AL,5BH          ; ICW2
OUT 91H,AL
MOV AL,03H          ; ICW4
OUT 91H,AL

```

(3) 中断向量表的建立如下所示。

```

PUSH DX
MOV AX,0
MOV DX,AX
MOV BX,2340H
MOV WORD PTR[016CH],BX
MOV BX,2000H
MOV WORD PTR[016EH],BX
POP DS

```

(4) 中断服务程序如下所示。

```

PRINTER: PUSH AX
          STI
          MOV AL,BL
          OUT 80H,AL
          CLI
          POP AX
          IRET

```

(5) 接口电路图如图 9.18 所示。

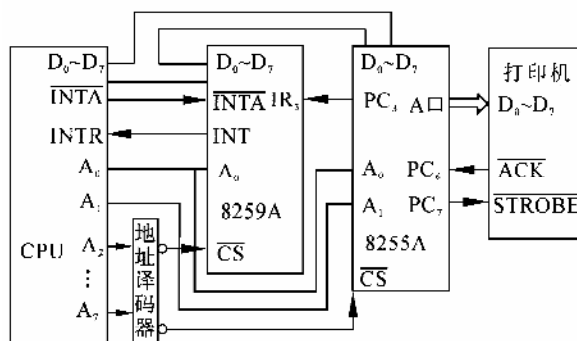


图 9.18 接口图

23. 根据题意，编写程序如下所示。

```

PUSH DS
MOV AX,0
MOV DS,AX
MOV BX, OFFSET XY      ; XY 为中断服务程序
MOV SI,SEG XY
MOV [0150H],BX
MOV [0152H],SI
POP DS
MOV DX,0FFF6H
MOV AL,10101001B      ; 8253 控制字
OUT DX,AL
MOV DX,0FFF4H
MOV AL,10H             ; 计数器 2 的计数值为 1000
OUT DX,AL
HLT
                        ; 中断服务程序
XY:  MOV DX,0FFF4H
      MOV AL,10H

```

OUT DX,AL

IRET

电路图如图 9.19 所示。

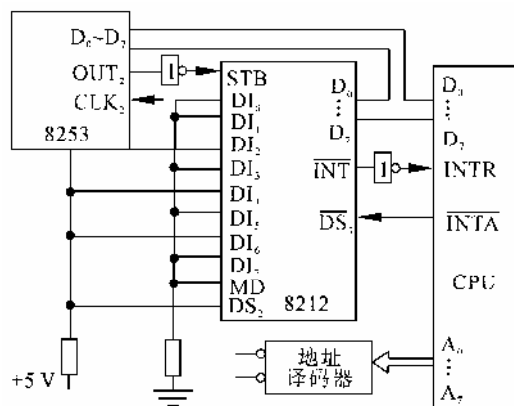


图 9.19 接口图

## 第 10 章

## 常用外围设备及人机接口

### 10.1 重点内容提要

#### （一）外围设备的功能

计算机外围设备是指除 CPU 和内存以外的计算机系统的其他部件。外围设备的功能如下：

- （1）对信息的形式进行交换。
- （2）为人和计算机系统提供联系通路。
- （3）作为信息存储的场所。
- （4）促进计算机在各个领域的应用。

#### （二）常用的外围设备

常用的输入设备有键盘、鼠标、扫描仪、光笔、触摸屏、条形码阅读器、数字化仪、声音/图像识别器，等等。

输出设备有显示器、打印机及绘图仪等。

外存设备有软盘存储器、硬盘存储器、光盘存储器等。

通信设备有终端设备、调制解调器、交换机、路由器等。

还有其他设备，如 A/D 和 D/A 转换器、开关设备等。

#### （三）外围设备的接口技术

##### 1. 键盘

早期的 PC 与键盘的接口采用移位寄存器来接收键盘发送的串行扫描码，通过并行接口 8255A 将装配好的数据送给 CPU，同时向 8259A 发中断请求。后来的机型，一般都使用增强型 101 键扩展键盘，它的键盘扫描码和断开扫描码均与标准键盘有很大差距。扩展键盘的接口采用了单片机（如 8242）作为键盘控制器，负责键盘接口的全部工作，实现键盘与主机之间的双向数据传送。

##### 2. 鼠标

串行通信鼠标一般采用 RS-232C 标准接口进行通信。通常带有一个 9 针的 D 型插头，插到微机上的一个串行口 COM1 或 COM2。

总线式鼠标需要在主机系统总线扩展槽中插一专门接口板。鼠标和接口板之间采用 9 针圆形插头连接。

PC/2 鼠标要使用专用的 6 芯 DIN 型接口插座。

### 3. 显示器

CRT 显示器与主机之间通过显卡连接。

随着计算机的发展,出现了许多视频显示标准,从 MDA、CGA、EGA 到 VGA 等,每一种视频标准都有相应的显卡与之对应。目前,显卡上还集成了图形加速功能。

### 4. 打印机

打印机分为针式打印机、喷墨打印机和激光打印机。

大多数打印机与主机并行接口连接。按照 Centronics 标准来定义插头插座的引脚。

## 10.2 常考题型范例精解

**例 10.1** 按功能分,外围设备可分为哪几类?对每一类举出几种外设名称。

**【解】** 按照功能,外围设备大致可分为输入设备、输出设备、外存设备、通信设备和其他设备。输入设备如键盘、鼠标、扫描仪等;输出设备如打印机、绘图仪等;外存设备如硬盘、软盘、光盘存储器等;通信设备如终端、调制解调器等。其他设备,如 A/D 和 D/A 转换器、开关设备等。

**例 10.2** 试说明 101 键盘接口电路的基本组成与作用。

**【解】** 101 键盘接口采用单片机 8042 作为键盘控制器,负责键盘接口的全部工作。可接收和校验键盘送来的键盘扫描码,可向 CPU 发出中断请求,还可向键盘发送控制命令并接收键盘发回的响应信号。

**例 10.3** 试说明在图形模式下,显示缓冲区是如何组织的。

**【解】** 在图形模式下,显示缓冲区以位的形式为每个像素保存信息,每个像素仅具有独立的颜色属性,没有形状、无闪烁、无背景色、也无数据。显示图形时,用描述像素颜色属性的二进制数的位数决定可同时显示的颜色数。

**例 10.4** 激光打印机由哪几部分组成?各部分作用是什么?

**【解】** 激光打印机由接口、激光扫描和电子照相等 3 部分组成。接口部分接收 CPU 送来的欲打印的信息,经字形发生器变换成不同频率的超声波。激光扫描部分利用不同频率的超声波使入射的激光束产生“0”级和“1”级衍射光,并在感光体的指定位置上形成扫描光点。电子照相部分对感光体形成静电影像,并转移到打印纸上。

## 10.3 习题及全解

### 一、问答题

1. 什么是外围设备?外围设备有何功能?
2. 什么是编码键盘?什么是非编码键盘?
3. 试说明文本模式下显示缓冲区是如何组织的。

**【解】**

1. 通常把除 CPU 和内存以外的计算机系统的其他部件都称为外围设备。外围设备具备如下功能：为人和计算机系统提供联系通路；可对信息的形式进行变换；可作为信息大量且长久存储的场所；还可促进计算机在各个领域的应用。

2. 编码键盘是指对每一个按键，由编码电路产生与其对应的唯一编码。也就是说，它是一种用硬件电路来识别按键代码的键盘。而非编码键盘是指利用简单的硬件和一套专用键盘程序来识别按键的位置，然后通过查表程序转换成相应的编码。

3. 文本模式下，显示缓冲区内用两个字节地址为每个字符保存信息，其中一个字节用来保存字符的 ASCII 码（或扩充的 ASCII 码）值，另一字节用来存放字符的属性，如前景色、背景色及闪烁等。



## 第 11 章

# D/A 和 A/D 转换器及其与 CPU 的接口

### 11.1 重点内容提要

#### （一）D/A 和 A/D 转换器概述

数/模（D/A）和模/数（A/D）转换技术主要用于计算机控制和测量仪表中。

在工业控制和参数测量时，经常遇到有关的参量是一些连续变化的模拟量，而计算机处理这些模拟量时，一般先利用传感器将它们变为模拟电流或模拟电压，然后通过 A/D 转换器变成计算机能够接收的数字量。对于控制系统来说，最终的目的是要根据当时现场情况进行控制。所以，计算机还应该发出信号送到控制目标。控制目标只能接收模拟量，所以需要通过 D/A 转换器将计算机的数字量转换为模拟电流或模拟电压。

#### （二）D/A 转换器及其接口技术

##### 1. D/A 转换器

D/A 转换器的功能是把数字量转换为模拟量。D/A 芯片是由集成在单一芯片上的解码网络并根据需要附加上一些功能电路所构成的。有些 D/A 芯片转换的结果是电流直接作为模拟量输出，有些芯片把电流经运算放大器转换成电压后输出。典型的 D/A 芯片如 DAC0832、DAC1210、AD558、AD7522 等。D/A 转换器的主要性能指标有分辨率、绝对精度、相对精度、线性误差、建立时间、温度系数、电源敏感度及输出电压一致性等。

##### 2. D/A 转换器与 CPU 的接口

D/A 转换器芯片一般提供如下接口信号：数据线、片选信号和写控信号。数据线可直接与微机系统的数据总线相连，写控信号可与 CPU 的写控信号相连或经过读写信号与其他信号组合产生。片选信号由 CPU 的地址译码产生。此外还有两条模拟信号输出线及基准电源线等。

##### 3. D/A 转换器与 CPU 的接口时应注意的问题

（1）数据锁存问题：当 CPU 送出一个数字信号给 D/A 时，这个数据在数据总线上只出现很短暂的一段时间。为了保证 D/A 完成转换，并在数据总线上数字信号消失后能保持有稳定的模拟输出，必须有一组锁存器保持住原输入的数字信号。有的 D/A 芯片提供了数据锁存器。如 DAC1210、DAC0832 等，在其芯片内部均采用两极锁存。这是比

较典型的 D/A 转换器芯片，必须掌握其结构和工作方式。还有的 D/A 芯片不提供数据锁存器，这时，就要在接口设计中予以外加。

(2) 模拟信号输出问题：D/A 芯片输出的模拟信号有些是电流信号，有些是电压信号。对于电流输出型 D/A 转换器，使用时需外加运算放大器或者负载电阻。另外，D/A 转换器输出通常是单极性的，但有些控制系统的模拟量是双极性的，因此要求 D/A 转换器的输出是双极性的。任何单极性 D/A 转换器都可以改为双极性的转换器。这时，只要在单极性 D/A 转换器输出运算放大器的求和点上加上一个能抵消半个量程电流的偏移电流即可。

(3) D/A 位数（分辨率）多于数据总线位数问题：如果 D/A 的位数多于微处理器数据总线的位数，则被转换的数据必须分几次进行。这就需要多个锁存器来锁存分几次送来的完整的数字数据，可以采用多级缓冲锁存来实现。

### （三）A/D 转换器及其接口技术

#### 1. A/D 转换器

A/D 转换的方法很多，归纳为直接比较型和间接比较型两大类。前者典型的有逐次逼近型，后者典型的双积分型转换。逐次逼近型转换速度比较快，性能适应大部分的应用场合。双积分型转换速度低，但抗干扰能力较强。

逐次逼近型 A/D 转换器一般由输入电压比较器、逐次比较寄存器、比较基准源及控制逻辑组成。一般还带有三态输出数据锁存器。这类 A/D 转换器芯片很多。如 8 位的 ADC0809、AD570、12 位的 AD574、ADC1210，还有 16 位的 ADC1140 等。要求掌握带 8 个模拟开关的 ADC0809 的工作原理和应用。

A/D 转换器的性能参数有分辨率、量化误差、转换时间、绝对精度、相对精度及漏码等。

#### 2. A/D 转换器与 CPU 的接口

一般的 A/D 转换器引脚信号有如下几种：模拟量输入、数据输出、启动转换信号、转换结束信号、输出允许和工作时钟。当 A/D 芯片与系统连接时，主要考虑这几种信号的连接。当数据输出线与系统数据总线连接时，要考虑三态输出能力。

#### 3. A/D 和 CPU 间的时间配合问题

设计 A/D 和微处理器间的接口中，要解决的突出问题是启动转换和读取结果数据这两步操作间的时间配合问题。可采用 5 种方法来解决此问题。

(1) 固定延时等待法：已知完成一次 A/D 转换所需的时间，可用定时器或软件延时一个固定时间，然后读取转换结果。这种方法的优点是接口简单，缺点是等待时间较长，且在等待期间微处理器不能去做别的工作。

(2) 保持等待法：用 A/D 启动后发出的状态信号去控制微处理器插入一些等待周期，以延缓处理器的读操作。转换结束解除延缓后再去读取数据。这种方法的先决条件是 CPU 必须有“等待”引脚。

(3) 中断响应法：用转换结束信号作为 CPU 的中断请求信号，以中断方式请求 CPU 读取转换结果。此方法可提高 CPU 的利用率。

(4) 程序查询法: 启动 A/D 后, 通过程序不断地读取转换结束信号, 若该信号有效, 表明完成了一次转换。这时读取转换结果。当读取结果的急迫性不是很高时, 可以采用此方法。

(5) 双重缓冲法: 在 A/D 和微处理器之间加一个具有三态输出能力的锁存器, 使这个锁存器中总是保持着最新的转换结果。任何时候 CPU 只要打开它即可读取最新结果数据。此方式可提高 A/D 转换结果数据的利用率。

## 11.2 常考题型范例精解

**例 11.1** 当用带两级数据缓冲器的 D/A 转换器时, 为什么 8 位 CPU 有时要用 3 条输出指令才完成 16 位或 12 位数据转换?

**【解】** 如果有一个 D/A 转换器超过 8 位, CPU 则通过两条输出指令向两个锁存器对应的端口地址输出数据。但是第一次执行输出指令后, D/A 转换器会得到一个局部的输入, 由此, 输出端得到一个局部的、实际上不需要的模拟量输出。因此用两级缓冲结构来解决, CPU 先用两条输出指令把数据送到第一级数据缓冲器, 通过第三条输出指令把数据送到第二级数据缓冲器, 从而使 D/A 转换器一次得到所有位待转换的数据。这就是用带两级缓冲器的 D/A 转换器必须要用 3 条输出指令才能完成 16 位或 12 位数据转换的理由。

**例 11.2** 设输入模拟信号的最高有效频率为 5 kHz, 试问应选用转换时间为多少的 A/D 转换器对它进行转换?

**【解】** 根据采样定理, 当采样频率高于或等于输入模拟信号最高有效频率的两倍时, 所采样的信号才不会失真。于是, 当输入模拟信号的最高有效频率为 5 kHz 时, 采样频率至少为 10 kHz。所以, 进行 A/D 转换时, 应采用转换时间最多为 1/10 kHz、0.1ms 的 ADC。

**例 11.3** 设计一个电路和相应程序, 完成一个锯齿波发生器的功能, 使锯齿波呈负向增长, 并且锯齿波周期可调。

**【分析】** 利用程序和电路产生一个锯齿波, 必须用到 D/A 转换技术。可以由 CPU 向 D/A 转换器送入适当的数据, 通过 D/A 转换, 得到相应的模拟量波形。

**【解】** 电路设计如图 11.1 所示, 这里采用 DAC0832 作为 D/A 转换器。

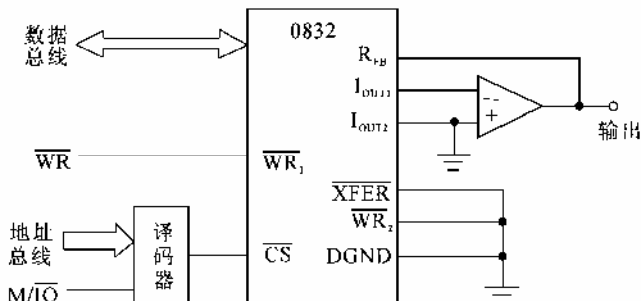


图 11.1 电路设计图

完成题目要求的程序片段如下所示。

```
MOV DX,DAPORT      ; DAPORT 为 D/A 转换器的端口号
MOV AL,00H          ; 初值
ROTATE:DEC AL        ; 负向锯齿波
OUT DX,AL           ; 向 A/D 转换器输出数据
CALL DELAY          ; 调用延时子程序
JMP ROTATE

DELAY PROC          ; 延时子程序
MOV CX,DATA         ; 向 CX 中送延时常数, 调节时间常数, 即可调节锯齿波周期
WAIT: LOOP WAIT
RET
DELAY ENDP
```

【评注】 由于 DAC0832 是一个 8 位的 D/A 转换器芯片，满量程电压对应的数字量为 11111111B，最小电压对应的数字量为 0。程序中，0 减去 1 变为 FFH。

例 11.4 若 DAC1210 接口电路如图 11.2 所示，问此 DAC1210 工作在何种方式？设模拟输出满量程为 10V，试编写出利用它产生 0~10V 三角波电压的程序段。

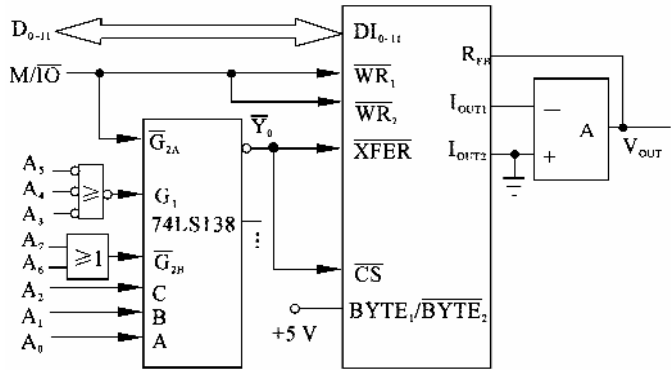


图 11.2 DAC1210 接口电路图

【解】 从几个控制信号的接法看出此 DAC1210 工作在单缓冲方式，通过对其中译码电路分析可知，DAC1210 的端口地址为 38H。产生三角波电压的程序段如下所示。

```
MOV AX,0
W1: OUT 38H,AX
NOP
NOP
INC AX
CMP AX,0FFFFH      ; 比较电压是否升到最高点
JNZ W1             ; 若不是最高点, 继续上升
W2: OUT 38H, AX
```

```

NOP
NOP
DEC AX
CMP AX,0           ; 比较电压是否升到最低点
JNZ W2             ; 若不是最低点，继续下降
JMP W1
    
```

【评注】程序中所加的 NOP 指令用于延时，可用 NOP 的多少来调节输出电压波形的周期。

例 11.5 接口原理图如图 11.3 所示。编写出对 IN<sub>0</sub>~IN<sub>7</sub> 八个模拟量各采样 100 个点的简化汇编源程序。

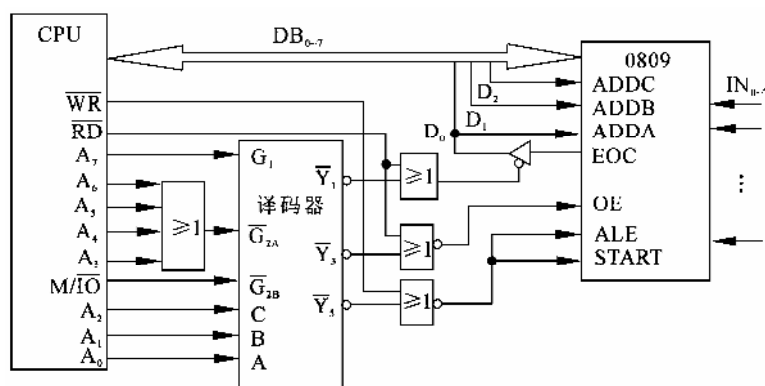


图 11.3 ADC0809 接口图

【解】根据译码电路，ADC0809 的 START 地址为 85H，EOC 地址为 81H，OE 地址为 83H。完成题目要求的程序如下所示。

```

.MODEL SMALL
.DATA
BLOCK DB 800 DUP(?)
.CODE
.STARTUP
LEA DI,BLOCK
MOV CX,100           ; 循环 100 次，次数送 CX
PP: MOV BL,0
LL: MOV AL,BL        ; BL 用来选择通道
    OUT 85H,AL        ; 启动转换器
    NOP
    NOP               ; 延迟，避免虚假 EOC 信号
W:  IN AL,81H
    TEST AL,01H
    JZ W              ; 检测 EOC，判断转换结束否，若转换未结束，则继续检测
    IN AL,83H
    
```

```

MOV [DI],AL          ; 若转换好，则读数并存入内存
INC DI
INC BL                ; 下一个通道
CMP BL,08H
JNZ LL               ; 8个通道是否转换完，若否，则继续下一个通道转换
LOOP PP
.EXIT
END

```

【评注】 由于转换结束信号 EOC 的变低相对于启动信号与 START 有 2us+8 个时钟周期的延迟，所以程序中在 A/D 启动后要加上 NOP 指令，使它不致产生虚假的转换结束信号。

例 11.6 利用 8255A 间接与 0809 接口，接口图如图 11.4 所示。采用查询方式，编写转换通道 IN5 的模拟量的程序。

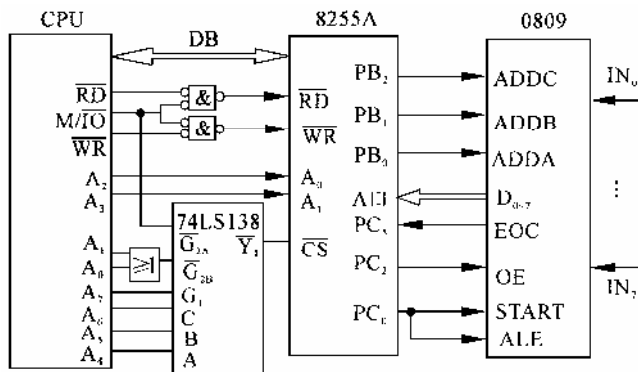


图 11.4 8255A 间接与 0809 接口的接口图

【解】 由接口图分析可知 8255A 的 A 口、B 口、C 口和控制口的地址分别为 90H、94H、98H 和 9CH，程序片段如下所示。

```

; 8255 初始化程序
MOV AL,10011000B
OUT 9CH,AL
; A/D 转换
MOV AL,05H
OUT 94H,AL          ; 选 IN5 通道
MOV AL,01H          ; 启动 A/D
OUT 9CH,AL
NOP                  ; 延时，避免虚假信号
NOP
W: IN AL,98H         ; 测 EOC
TEST AL,00100000B
JZ W
MOV AL,00000101B    ; OE 置 1

```

```

OUT 9CH,AL
IN AL,90H
MOV BL,AL          ; 转换结果存入 BL 中
MOV AL,00000000B   ; OE 清 0
OUT 9CH,AL

```

【评注】 注意 ADC0809 以异步查询方式与 CPU 通信时的工作流程，8255A 以此来进行接口控制。

例 11.7 有如图 11.5 所示的接口原理图。设该系统 8259A 的 8 个中断类型码为 70H~77H，边沿触发。写出对  $IN_1$  一个模拟量采样并转换 10 个点的程序片段。

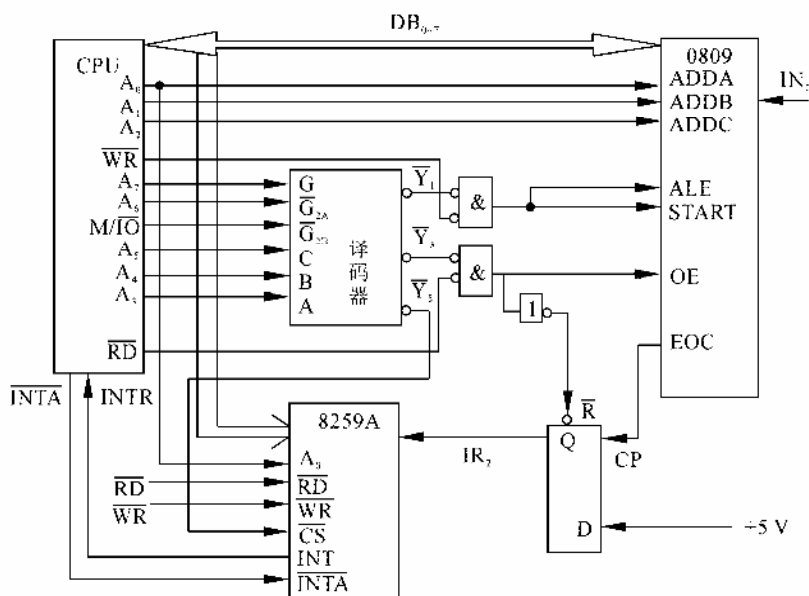


图 11.5 8259A 与 ADC0809 的接口图

【解】 由接口图分析可知 ADC0809 的 START 地址为 89H，OE 地址为 98H；8259A 的地址为 A8H 和 A9H。完成对  $IN_1$  一个模拟量采样并转换 10 个点的程序片段如下所示。

； 建立向量表

```

PUSH DS
MOV AX,0
MOV DX,AX
MOV BX,OFFSET XY          ; XY 为中断服务程序首地址
MOV [01C8H],BX           ; 72H×4=01C8H
MOV BX,SEG XY
MOV [01CAH],BX
POP DS

```

```

                                ; 初始化 8259A
MOV AL, 13H
OUT 0A8H,AL                    ; ICW1
MOV AL,70H
OUT 0A9H, AL                  ; ICW2
MOV AL,03H
OUT 0A9H,AL                    ; ICW4
MOV BX,OFFSET WP              ; 设 WP 为内存缓冲区首地址
MOV CX,10
W: OUT 89H,AL                  ; 启动 0809
NOP
NOP
HLT                            ; 等待中断
MOV [BX],AL
INC BX
LOOP W
; 中断服务程序
XY: IN AL, 98H                 ; 读转换好的数据
IRET

```

## 11.3 习题及全解

### 一、填空题

1. DAC0832 是一个\_\_\_\_\_位的 D/A 转换器, 具有\_\_\_\_\_级锁存功能, 输出的是\_\_\_\_\_ (电流/电压) 信号。
2. DAC1210 是\_\_\_\_\_位的 D/A 转换器, 可工作在\_\_\_\_\_方式和\_\_\_\_\_方式。
3. 若要将数字量转换为模拟电压, 需要用到\_\_\_\_\_转换器, 若该转换芯片为电流输出, 则须外接\_\_\_\_\_。
4. A/D 转换时, 若输入模拟信号的最高有效频率为 20 kHz, 采样频率最小为\_\_\_\_\_, 应选用转换时间为\_\_\_\_\_的 A/D 转换器。
5. A/D 转换时, 若输入模拟信号的最高有效频率为 10 kHz, 应选用转换时间为\_\_\_\_\_, 的 A/D 转换器, 若此 A/D 转换器是 12 位的, 则其分辨率为\_\_\_\_\_。

#### 【解】

1. 8; 2; 电流
2. 8, 单缓冲方式; 双缓冲方式
3. 数/模; 运算放大器
4. 40 kHz; 25  $\mu$ s



5. 小于等于 0.05ms; 12 位

## 二、选择题

1. 一个 8 位 D/A 转换器的分辨能力可以达到满量程的\_\_\_\_\_。  
A. 1/8      B. 1/256      C. 1/16      D. 1/32
2. 设有一现场模拟信号, 其最大可能变化频率为 2kHz, 则计算机在收集此信息时, 最多每隔\_\_\_\_\_输入一次, 才能反映出输入信号的变化。  
A. 0.5 ms      B. 0.25 ms      C. 0.15 ms      D. 2 ms
3. 某一测控系统要求计算机输出的模拟控制信号的分辨率必须达到千分之一, 则选择的 D/A 转换器的位数应该至少是\_\_\_\_\_。  
A. 4 位      B. 8 位      C. 10 位      D. 12 位
4. 使用 A/D 转换器对一个频率为 4 kHz 的正弦波信号进行输入, 要求在一个信号周期内采样 5 个点, 则应选用 A/D 转换器的转换时间最大为\_\_\_\_\_。  
A. 1  $\mu$ s      B. 100  $\mu$ s      C. 10  $\mu$ s      D. 50  $\mu$ s
5. 当 CPU 使用中断方式从 ADC0809 读取数据时, ADC0809 向 CPU 发出中断请求的信号是\_\_\_\_\_。  
A. START      B. OE      C. EOC      D. INTR
6. 关于 DAC0832 或 DAC1210, 下列\_\_\_\_\_的说法正确。  
A. 是电流输出  
B. 是电压输出  
C. 外加一个放大器后可得到单向电压输出  
D. 外加一个放大器和一个运算放大器可得到双向电压输出
7. 关系 ADC0809, 下列\_\_\_\_\_的说法正确。  
A. 只能接一个模拟量输入      B. 可以接 8 个模拟量输入  
C. 某时刻只对一个模拟量采样      D. 同时对 8 个模拟量采样
8. 关于 DAC0832 或 DAC1210, 下列\_\_\_\_\_的说法正确。  
A. 是 A/D 转换器, 可接 8 个模拟量输入  
B. 是 D/A 转换器, 电流输出  
C. 具有一级缓冲器的 D/A 转换器  
D. 具有两级缓冲器的 D/A 转换器  
E. 是 A/D 和 D/A 均可实现的转换器

### 【解】

1. B    2. B    3. C    4. D    5. C    6. A、C、D    7. B、C    8. B、D

## 三、接口设计题

1. 编写用 AD558 D/A 转换器芯片产生方波的程序。试问如何控制方波频率?
2. 编写用 DAC0832 D/A 转换器芯片产生三角波的程序, 其变化范围为 0~10V。若要在 -5~+5V 之间变化要采用什么措施实现?

3. 试设计一个采用固定延时等待法并每采集一个数据只需一条 I/O 指令的 CPU 和 ADC0809 的接口电路，并编制程序使之把所采集的 8 个通道的数据送入给定的内存区。
4. 试设计一个采用查询法并用数据线选择通道的 CPU 和 ADC0809 的接口电路，并编制程序使之把所采集的 8 个通道的数据送入给定的内存区。
5. 若采用两级缓冲锁存方式，将 10 位 D/A 转换器与 8 位 CPU 接口，试编出转换 10 个数的程序。
6. 有接口电路如图 11.6 所示，用 0809 对接在  $IN_3$  上的  $V_x(t)$  采样并转换 10 个点。若每个点转换后的数大于 0F0H，则送 D/A，否则不送。数都要保存在内存。写出汇编源程序。

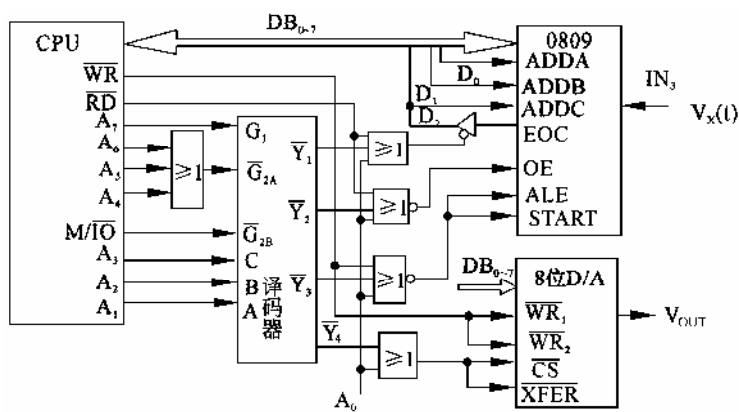


图 11.6 接口电路图

### 【解】

1. 设 AD558 端口地址为 PORT，程序如下所示。

```

MOV DX, PORT
LP: MOV AL, 0
OUT DX, AL
CALL RLY ; RLY 为延时子程序
MOV AL, 0FFH
OUT DX, AL
CALL RLY
JMP LP

```

方波频率由延时子程序来控制，延时时间越短、方波频率越高。

2. 设 0832 采用缓冲方式，且端口地址为 PORT，程序如下所示。

```

MOV DX, PORT
XOR AL, AL
W1: OUT DX, AL
INC AL
CMP AL, 0FFH

```

```

JNZ W1
W2: OUT DX,AL
DEC AL
CMP AL,00H
JNZ W2
JMP W1

```

若要双极性输出，可在 0832 转换器的输出运算放大器后面加一级比例求和电路，通过电压偏移变为双极性电压输出。

3. 接口电路设计如图 11.7 所示，相应的程序编制如下所示。

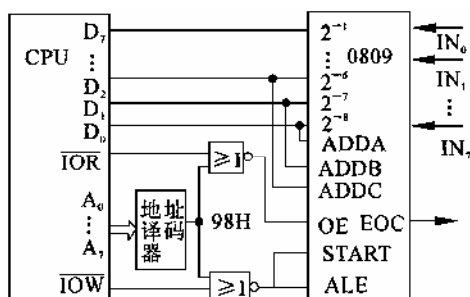


图 11.7 接口电路图

```

MOV SI,OFFSET BLOCK      ; 设 BLOCK 为缓冲区首地址
MOV BL,0
LL: MOV AL,BL
OUT 98H,AL                ; 启动 A/D
CALL RLY                  ; RLY 为延时子程序
IN AL,98H
MOV [SI],AL
INC SI
INC BL
CMP BL,08H
JNZ LL
HLT

```

电路图如图 11.7 所示。

4. 硬件设计图如图 11.8 所示，由图可知，各通道由数据线选择，译码器对应的地址分别为 80H、81H 和 82H。下面程序采用查询法对 IN<sub>0</sub>~IN<sub>7</sub> 各采样一次，转换的数据送入以 BLOCK 为首地址的内存区中。

```

MOV SI,OFFSET BLOCK      ; 设置数据存储指针
MOV BL,0                 ; BL 用来设置通道号
P: MOV AL,BL              ; 选通通道，启动 A/D
OUT 82H,AL
NOP                       ; 延时，避免 EOC 虚假信号

```

NOP

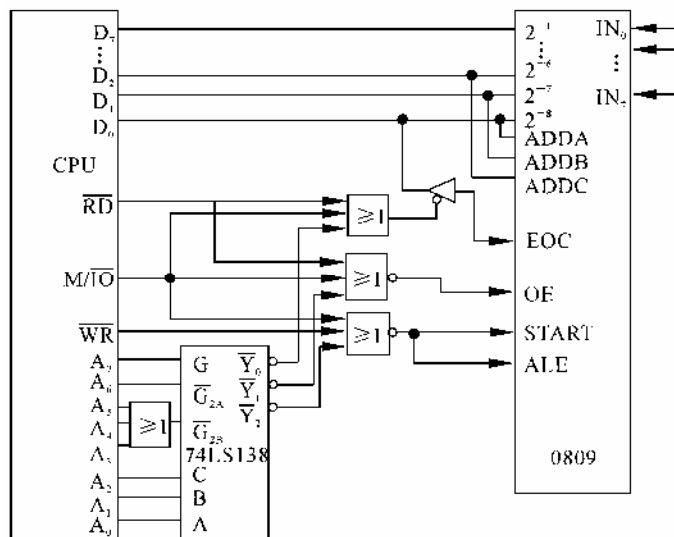


图 11.8 硬件设计图

```

W: IN AL,80H          ; 输入 EOC 标志
   TEST AL,01H        ; 测试状态
   JZ W               ; 未结束, 返回等待
   IN AL,81H          ; 若转换结束, 读数据
   MOV [SI],AL        ; 存数
   INC SI
   INC BL
   CMP BL,08H
   JNZ P
   HLT

```

电路图如图 11.8 所示。

5. 程序如下所示。

```

DATA DW 1010H,3407H,... ; (10 个数)
...
MOV CX,10              ; 置计数初值为 10
MOV SI,0               ; 地址指针为 0
AGAIN: MOV BX,DATA[SI] ; 取出内存缓冲区的数据
MOV AL,BL
OUT PORTL,AL           ; 向第一级缓冲锁存器送低 8 位数据
MOV AL,BH
OUT PORTH,AL           ; 向第一级缓冲锁存器送高 2 位数据
OUT PORT,AL            ; 选通第二级缓冲锁存器
CALL DELAY             ; 调延时子程序

```

```

                INC SI                ; 地址指针加1
                LOOP AGAIN            ; 计数减1, 若没有转换完10个数据, 则跳至 AGAIN, 否则结束
                RET
DELAY: PUSH CX                ; 保护 CX
                MOV CX,COUNT
WAIT:  LOOP WAIT                ; 延时
                POP CX              ; 恢复 CX
                RET                ; 返回

```

6. 根据译码电路分析可知, 0809 的 START 地址为 86H, EOC 地址为 82H, OE 地址为 84H; D/A 转换器的地址为 88H。程序如下所示。

```

.MODEL SMALL
.DATA
BLOCK DB 10 DUP(?)
.CODE
.STARTUP
MOV BX,OFFSET BLOCK
MOV CX,10
AGAIN: MOV AL,03H
        OUT 86H,AL                ; 启动 0809, 选择 IN3 通道
        NOP
        NOP                        ; 延时, 避免虚假 EOC 信号
W:  IN AL,82H
        TEST AL,04H
        JZ W                        ; 检测 EOC 信号, 若未转换完毕, 继续检测等待
        IN AL,84H                    ; 若转换完毕, 则读数据
        CMP AL,0F0H                  ; 转换数据与 0F0H 比较
        JBE DOWN
        OUT 88H,AL                    ; 若大于 0F0H, 则送 D/A 转换
DOWN: MOV[BX],AL                    ; 数据存入内存
        INC BX
        LOOP AGAIN
        .EXIT
END

```

## 第 12 章

## 微型计算机的应用

### 12.1 重点内容提要

#### （一）计算机的应用范围

计算机的应用范围几乎涉及各行各业，大到卫星发射、矿产开发、石油勘探，小到儿童玩具、电子称、电子表等。

传统的计算机的应用范围主要包括科学计算、数据处理和过程控制等领域。

#### （二）微型计算机的应用

计算机能渗透到各个部门、各个单位乃至居民家庭，是随着微型计算机被推广使用才真正成为现实。微型计算机在过程控制、计算机辅助设计、辅助制造和辅助教学、人工智能等方面都得到了广泛的应用。下面对此作简要介绍。

（1）在辅助科学实验室中的应用：如美国西北太平洋拜特尔实验室用计算机作为计算和控制装置，成功地完成了原子能反应堆内的模拟实验。

（2）在生物科学中的应用：比如进行呼吸率的测试时，呼吸率根据计算两次呼吸之间的时间间隔来决定，呼吸量则要将外部模拟量通过传感器转换成电信号，再通过 A/D 转换由计算机测量。

（3）在过程控制中的应用：比如数据采集、开环控制、闭环监控、直接数字控制、模糊控制等这些应用都离不开微型计算机的应用。

（4）在临床医疗仪器中的应用：临床使用的仪器要求高度可靠、使用方便，而且尽可能减少成本，微型计算机是最佳选择。比如在超声波仪器中，微机应完成测量传输时间，确定超声波束相对于某一坐标系的位置和方向，完成必要的计算等工作。

### 12.2 常考题型范例精解

**例 12.1** 设计一个采用中断传送方式，从 ADC0809 八个通道各采样 100 个点的数据采集系统。画出接口图，用简化汇编语言写出源程序，可合理假设接口地址，中断类型为 60H~67H。

**【分析】** 由于此数据采集系统要求采用中断传送方式，可以考虑用中断控制器

8259A 进行中断控制。

【解】 接口电路图可设计如图 12.1 所示。

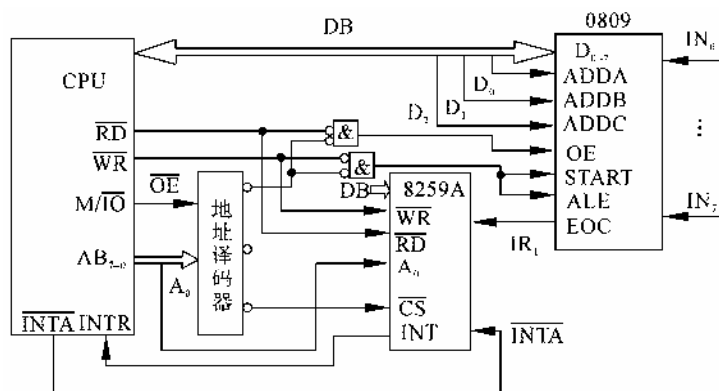


图 12.1 数据采集系统接口电路图

完成数据采集的程序如下所示。

```
.MODEL SMALL
.DATA
BLOCK DB 800 DUP(?)
.CODE
.STARTUP
MOV CX,100
MOV BX,OFFSET BLOCK
MOV AL,13H                ; 8259A 初始化, ICW1
OUT 20H,AL
MOV AL,60H                ; ICW2
OUT 21H,AL
MOV AL,01H                ; ICW4
OUT 21H,AL
PUSH DS                    ; 建立中断向量表
MOV AX,0
MOV DS,AX
MOV SI,OFFSET XY
MOV DI,SEG XY
MOV [0184H],SI            ; 61H×4=0184H
MOV [0186H],DI
POP DS
PP1: MOV DL,0
PP2: MOV AL,DL
OUT 98H,AL                ; 启动 0809
NOP
HLT                        ; 等待中断
INC DL
```

```

CMP DL,08H
JNZ PP2
DEC CX
JNZ PP1
.EXIT
END

```

； 中断服务程序

```

XY:  PUSH AX
      IN  AL,98H
      MOV [BX],AL
      INC BX
      POP AX
      IRET

```

**例 12.2** 用 8255A 控制 DAC0832 进行 D/A 转换，控制 8253 产生方波。(1) 试根据图 12.2 所示的连线，给出 8255A 和 8253 的端口地址，并为 8253 选择合适的工作方式，确定计数初值；(2) 编程要求：设 8255 工作在方式 0，需转换的数字量在 BL 中存放，试编写程序段，使得 DAC0832 产生模拟量输出，8253 产生所要求的方波。

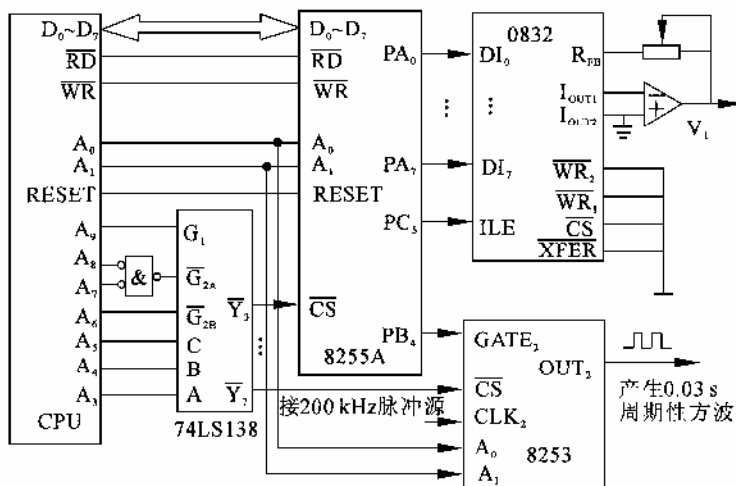


图 12.2 接口电路图

### 【解】

(1) 从图 12.2 中的译码电路可知：

8255A 的地址为 0218H~021BH

8253 的地址为 0238H~023BH

根据题意，8253 的计数器应工作于方式 3，计数初值为：

$$n = 0.03 \times (200 \times 1\,000) = 6\,000$$

(2) 完成题目要求的程序段如下所示。

```

MOV DX,39FH          ; 8255A 控制字端口

```



```

MOV AL,10000000B
OUT DX,AL
MOV DX,39CH          ; 8255A 的 A 端口
MOV AL,BL
OUT DX,AL
MOV AL,00001011B     ; PC5 置 1, 0832 进行 D/A 转换
MOV DX,39FH
OUT DX,AL
MOV AL,00001010B     ; PC5 清 0
OUT DX,AL
MOV DX,3BFH          ; 8253 控制字端口
MOV AL,10110110B     ; 计数器 2, 方式 3, 二进制计数
OUT DX,AL
MOV AX,6000
MOV DX,3BEH
OUT DX,AL             ; 送计数初值低 8 位
MOV AL,AH
OUT DX,AL             ; 送计数初值高 8 位
MOV DX,39DH          ; 8255A 的 B 端口
MOV AL,00010000B     ; PB4 置 1, GATE2 有效
OUT DX,AL

```

或者, 8253 的计数器 2 可以用 BCD 码格式计数, 则 8253 程序片段如下所示。

```

MOV DX,3BFH
MOV AL,10110111B
OUT DX,AL
MOV DX,3BEH
MOV AL,00H
OUT DX,AL
MOV AL,60H
OUT DX,AL
MOV DX,39DH
MOV AL,00010000B
OUT DX,AL

```

**例 12.3** 某 12 位 A/D 转换器, 工作时需首先在 SC 端送入低电平有效的启动转换信号, 这时在数据输出端 D<sub>0</sub>~D<sub>11</sub> 可得到 12 位的转换结果。采用 8255A、8253 和该 ADC 实现定时采样功能, 图 12.3 给出了一模块的部分信号连接图。

(1) 将其余的信号线与 CPU 的系统总线连接起来(要求 8255A 的地址为 220H~223H, 8253 的地址为 228H~22BH)。

(2) 该模块中 8255A 的 A 口、B 口均工作在方式 0、输入方式。8253 采用方式 3, 实现定时中断信号 IR<sub>2</sub>, 定时时间为 200μs, 写出 8255A 及 8253 初始化程序片段。

(3) 写出对输入模拟量进行采样的 A/D 转换的程序, 将转换结果低 8 位保存在 ADL

内存单元，将高 4 位保存在 ADH 内存单元。

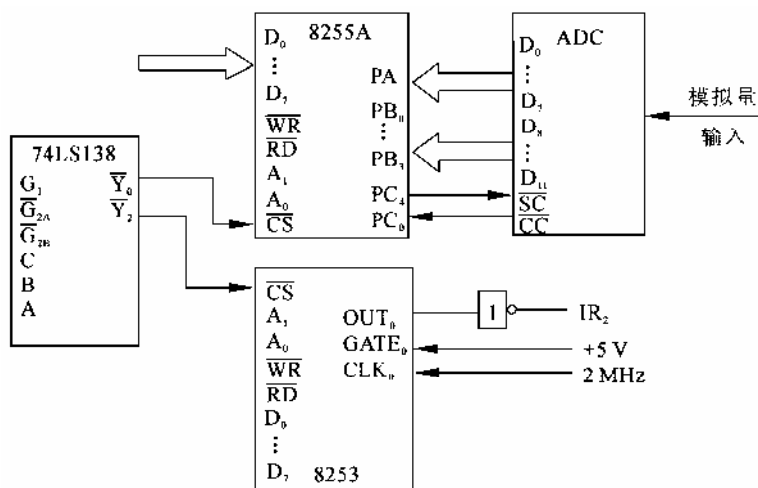


图 12.3 部分信号连接图

【解】

(1) 系统连线如图 12.4 所示。

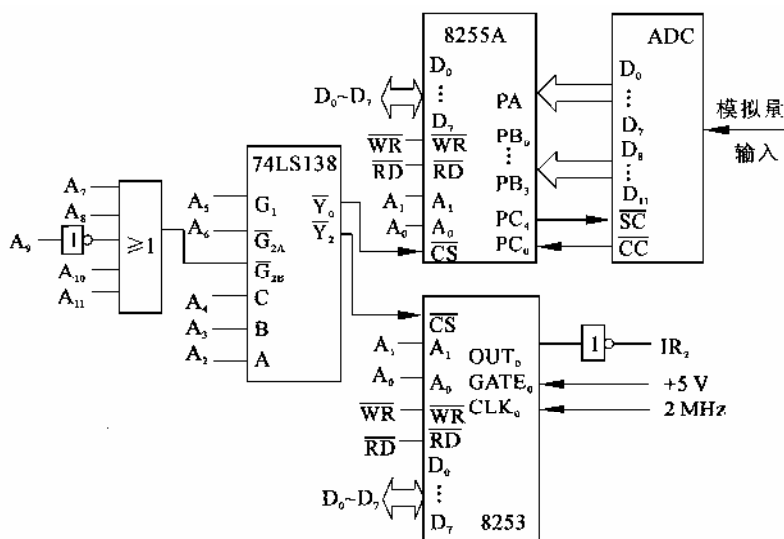


图 12.4 系统连线图

(2) 8255A 与 8253 的初始化程序如下所示。

```
MOV AL,10010011B      ; 送 8255A 工作方式字
MOV DX,223H
OUT DX,AL
MOV AL,00110111B      ; 送 8253 控制字
```

```

MOV DX,22BH
OUT DX,AL
MOV DX,228H           ; 送计数初值
MOV AL,00H
OUT DX,AL
MOV AL,04H
OUT DX,AL

```

(3) 定时采样程序如下所示。

```

MOV DX,223H
MOV AL,00001000B
OUT DX,AL             ; PC4复位,启动 A/D
MOV DX,222H
K1: IN AL,DX
AND AL,01H
JNZ K1                ; 读入 PC0 的状态,不为低电平时需等待
MOV DX,220H
IN AL,DX              ; 从 A 口读入低 8 位数据
MOV ADI,AL
INC DX
IN AL,DX              ; 从 B 口读入高 4 位数据
MOV ADI,AL
RET

```

**例 12.4** 接口图如图 12.5 所示,写出每 1ms 向 DAC1210 输送一次数据,使之输出 0 至满量程电压的斜线波形。注:8259A 设计为单片,边沿触发工作方式,中断类型码为 80H~87H,接口地址可合理假设。

**【解】** 假设 8259A 的地址:20H,21H

DAC1210 地址:80H

8253 口地址:70H~73H

由于要求每 1ms 向 DAC1210 输送一次数据,也就是说 8253 的定时周期为 1ms,那么计数初值  $n = T \times fc = 1\text{ms} \times 4\text{MHz} = 4\,000$  次,完成题目要求的程序片段如下所示。

```

; 建立中断向量
PUSH DS
MOV AX,0
MOV DS,AX
MOV BX,OFFSET XY
MOV SI,SEG XY
MOV [0214H],BX       ; 85H×4=0214H

```

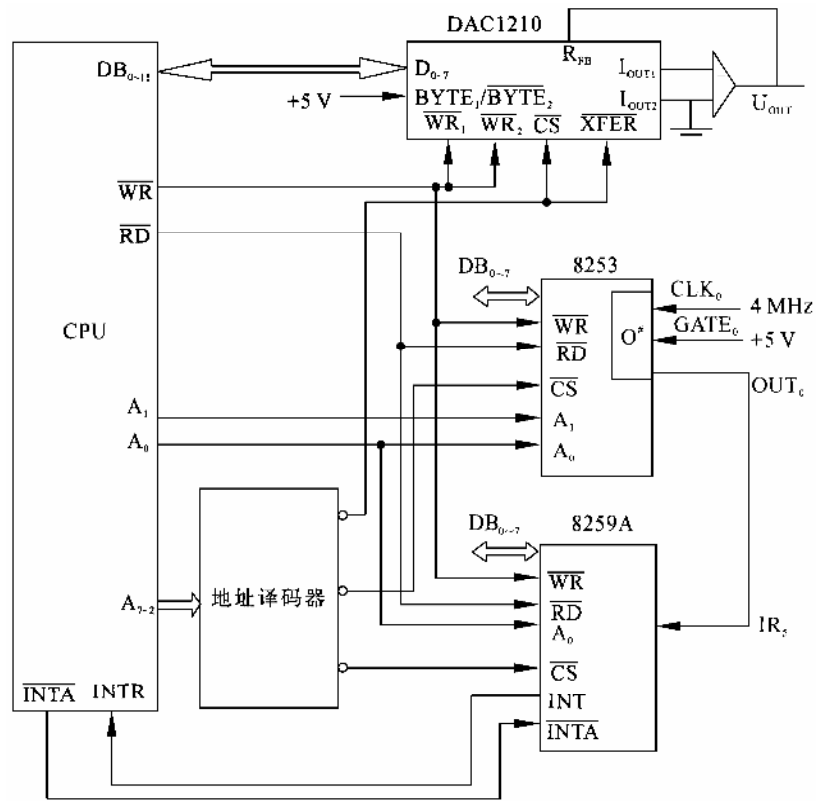


图 12.5 接口图

```

MOV [0216H],SI
POP DS
; 初始化 8259A
MOV AL,00010011B          ; 8259A ICW1
OUT 20H,AL
MOV AL,80H                 ; ICW2
OUT 21H,AL
MOV AL,01H                 ; ICW4
OUT 21H,AL
; 初始化 8253
MOV AL,00100101B          ; 设置控制字
OUT 73H,AL
MOV 40H,AL                 ; 给计数器 0 设置计数初值
OUT 70H,AL
; 主程序
MOV AX,0
X1: HLT                    ; 等待中断
INC AX
CMP AX,0FFFFH

```

```

        JNZ X1                ; 判断电压是否达满量程，若否，则继续转换
        ...
; 中断服务程序
        STI
        OUT 80H,AL
        NOP
        NOP
XY:     CLI
        IRET

```

## 12.3 习题及全解

1. 有一接口电路图如图 12.6 所示。8253 的 1<sup>#</sup> 计数器定时 1s,  $CLK_1 = 500\text{ Hz}$ ;  $CLK_2 = 0.1\text{ MHz}$ 。要求在 1 秒钟之内由 CPU 向 0832 送 100 字节数。编写出简化的汇编语言源程序（可合理假设接口地址）。

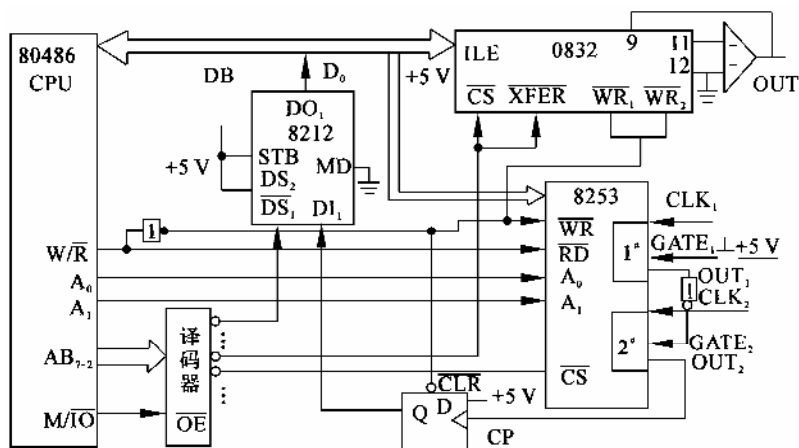


图 12.6 接口电路图

【解】

```

.MODEL SMALL
.DATA
BLOCK DB 1, 2, ..., ... ; 100 字节数
.CODE
.STARTUP
MOV CX, 100
MOV AL, 01100011B ; 1#初始化，方式 1，计数值 500
OUT 73H, AL ; 设 8253 的地址为 70H~73H
MOV AL, 05H
OUT 71H, AL

```

```

MOV AL,10100101B          ; 2#初始化, 方式 2, 计数值 1000
OUT 73H,AL
MOV AL,10H
OUT 72H,AL
MOV BX,OFFSET BLOCK
P:  IN AL,80H                ; 设 8212 的地址为 80H
    TEST AL,01H
    JZ P
    MOV AL,[BX]
    OUT 20H,AL              ; 设 0832 的地址为 20H
    INC BX
    DEC CX
    JNZ P
    .EXIT
END

```

2. 在某微机控制系统中, 8251A 作为 CPU 的全双工异步串行通信接口, 采用中断方式工作, 波特率因子为 16, 字符长度为 8, 偶校验, 1 位停止位, 发送和接收数据的波特率均为 1 200。8253 的 2<sup>#</sup>计数器为 8251A 提供发送、接收时钟 T<sub>x</sub>C 和 R<sub>x</sub>C, 其 CLK 输入为 8MHz 信号。已知 8251A 的端口地址为 0F0H 和 0F1H, 8253 的端口地址为 230H~233H。请对 8251A 和 8253 进行初始化。

**【解】**

(1) 8251A 的初始化程序片段为。

```

MOV DX,0F1X
MOV AL,01111110B          ; 方式控制字
OUT DX,AL
MOV AL,35H                 ; 操作命令字
OUT DX,AL

```

(2) 8253 的 2<sup>#</sup>计数器用作串行通信的波特率发生器, 所以选择方式 3。又输入 CLK 为 8 MHz, 要求输出方波为 16 (波特率因子) × 1200 (波特率) = 19.2 kHz, 所以 2<sup>#</sup>计数器的计数初值为:

$$8 \text{ MHz} / 19.2 \text{ kHz} = 417 = 1\text{A}1\text{H}$$

8253 的初始化程序片段为:

```

MOV AL,10110110B          ; 2#计数器, 方式 3, 二进制计数
MOV DX,233H
OUT DX,AL
MOV DX,232H
MOV AL,0A1H;                ; 计数初值低 8 位
OUT DX,AL
MOV AL,01H                 ; 初值高 8 位
MOV DX,AL

```

Figure 1 illustrates the hardware connection of the printer interface. It features two integrated circuits: an 8255A PPI and an 8253 PPI. The 8255A chip is connected to a printer (打印机) via its parallel ports PA<sub>0</sub>~PA<sub>7</sub> (data bus D<sub>7</sub>~D<sub>0</sub>), PC<sub>6</sub> (ACK), PC<sub>7</sub> (STB), and OBF. Its chip select (CS) is connected to address 220H~223H, and its address lines A<sub>1</sub> and A<sub>0</sub> are connected to address lines A<sub>1</sub> and A<sub>0</sub>. The 8253 chip is connected to the 8255A's RD and WR signals. Its OUT<sub>1</sub> is connected to RD, and OUT<sub>0</sub> is connected to WR. The 8253's CS is connected to address 228H~22B, and its data bus D<sub>7</sub>~D<sub>0</sub> is connected to DB<sub>7</sub>~DB<sub>0</sub>. A 1.193 181 6 MHz clock signal is connected to CLK<sub>0</sub>, and a +5V supply is connected to GATE<sub>0</sub>.

- (1) 编写 8253 的初始化程序。
- (2) 编写 8255A 的初始化程序及打印程序。

(1) 16 位的计数器在时钟频率为 1.193 181 6MHz 时的最大定时时间为:

$2^{16}/1.193\ 181\ 6\ \text{M}=55\ \text{ms}$

所以, 须采用串接两个计数器  $0^\#$  和  $1^\#$  来完成 30 分钟的定时任务。

计数器 0<sup>#</sup> 和 1<sup>#</sup> 都工作于方式 2, 对时钟信号进行分频, 在计数器 1<sup>#</sup> 的 OUT 端得到一个每隔 30 分钟发出的负脉冲, 用于向 CPU 申请中断, 在中断服务程序中调用打印程序。

$$N_0 \times N_1 = 30 \times 60 \times 1.193\,181\,6M = 2\,147\,726\,880 = 65\,536 \times 32\,772$$

$$N_0=0FFFFH, \quad N_1=8004H$$

8253 初始化程序如下所示。

```
MOV AL,00110100B ; 0# 计数器, 方式2, 二进制计数
MOV DX,22BH
OUT DX,AL
MOV AL; 0FFH ; 计数值低8位
MOV DX,228H
OUT DX,AL
```

```

OUT DX,AL                ; 设置高 8 位
MOV AL,01110100B         ; 1# 计数器, 方式 2, 二进制计数
MOV DX,22BH
OUT DX,AL
MOV AL,04H                ; 设置低 8 位初值
MOV DX,229H
OUT DX,AL
MOV AL,80H                ; 设置高 8 位初值
OUT DX,AL

```

(2) 8255A 的初始化程序如下所示。

```

MOV AL,10100011B
MOV DX,233H
OUT DX,AL

```

打印程序为:

```

PRINT: MOV CX,32          ; 打印字符长度计数
        MOV AX,SEG BUF    ; 字符缓冲区首地址送入 DS: SI
        MOV DS,AX
        MOV AX,OFFSET BUF
        MOV SI,AX
PT1:    MOV AL,[SI]        ; 取打印字符
        MOV DX,220H       ; 从 A 口输出字符
        OUT DX,AL
        MOV DX,222H
WAIT:   IN AL,DX           ; 从 C 口读取状态字
        TEST AL,10000000B ; 测输出缓冲器满信号  $\overline{\text{OBF}}$  变高了吗?
        JZ WAIT           ; 否, 再查
        INC SI            ; 是, 传递下一个字符
        LOOP PT1
        RET

```



## 附 录

# 硕士研究生入学“微机原理及应用” 考试试题及参考答案

## 附录 1 硕士研究生入学考试模拟试题及答案

### 一、填空题（每空 1 分，共 15 分）

1. 如果 BUF 为数据段中 5400H 单元的符号名，其中存放的内容为 1234H，执行指令 MOV BX, BUF 后，BX 的内容为\_\_\_\_\_；而执行 LEA BX, BUF 后，BX 的内容是\_\_\_\_\_。
2. 若  $X = -1$ ,  $Y = -127$ , 字长  $n = 16$ , 则  $[X+Y]_{补} =$  \_\_\_\_\_ H,  $[X-Y]_{补} =$  \_\_\_\_\_ H。
3. 指出下列每条指令中源操作数的寻址方式：  
MOV AX, [4000H]; \_\_\_\_\_  
MOV AL, [BX+DI]; \_\_\_\_\_  
MOV EAX, [BP+5000H]; \_\_\_\_\_
4. 输入/输出指令间接寻址必须通过\_\_\_\_\_寄存器。
5. CPU 响应可屏蔽中断的条件是\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_。
6. CPU 从 I/O 接口的\_\_\_\_\_中获取外部设备的“准备好”、“忙”或“闲”等信息。
7. DAC0832 是一个\_\_\_\_\_位的 D/A 转换器，具有\_\_\_\_\_级锁存功能，输出是\_\_\_\_\_（电流/电压）信号。

### 二、选择题（单选或多选，每小题 3 分，共 15 分）

1. 在 ADD 指令中，两个操作数的物理位置可以安排在\_\_\_\_\_中。  
A. 两个存储单元  
B. 一个存储单元和一个数据寄存器  
C. 两个数据寄存器  
D. 一个堆栈单元和一个数据寄存器
2. 主机与外围设备传送数据时，采用\_\_\_\_\_，主机与外设是串行工作的。  
A. 程序查询方式  
B. 中断方式  
C. DMA 方式  
D. I/O 处理机方式
3. 下列指令中，有语法错误的是\_\_\_\_\_。  
A. MOV [SI], [DI]

- B. IN AL, DX
- C. JMP WORD PTR[BX+8]
- D. PUSH BYTE PTR 20[BX+SI]

4. 可做简单输入接口的电路是\_\_\_\_\_。

- A. 三态缓冲器
- B. 锁存器
- C. 反相器
- D. 译码器

5. CPU 响应外部中断请求是\_\_\_\_\_。

- A. 在一条指令执行结束后
- B. 在一个机器周期结束后
- C. 一旦请求, 立即响应
- D. 由中断类型码 n 引起

### 三、问答题 ( 每小题 4 分, 共 20 分 )

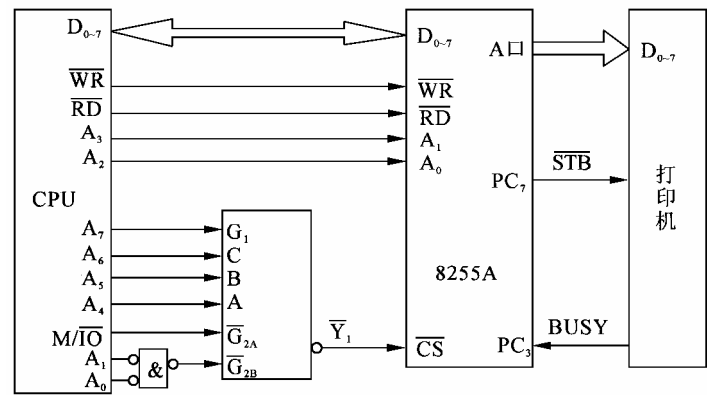
1. I/O 编址方式中, 什么是存储器映像 ( 统一编址 ) 和 I/O 映像 ( 单独编址 ) ?
2. 计算机的主存与辅存 ( 即内存与外存 ) 是以什么原则划分的 ? 80486 CPU 直接访问的内存空间 ( 不考虑虚拟空间 ) 最大是多少 ?
3. 8253 是计数器/定时器, 那么计数和定时的区别是什么 ? 8253 的各种工作方式在 OUT 输出特性及 GATE 信号上有什么区别 ?
4. 8259A 的中断屏蔽寄存器 IMR 和中断允许标志 IF 有什么差别 ? 在中断系统中它们是如何起作用的 ?
5. 在实方式和虚拟 8086 方式下, Intel 80486 中段寄存器中存放的值是什么 ? 而在保护方式下, 它的功能又是什么 ?

### 四、汇编程序设计题 ( 每小题 15 分, 共 45 分 )

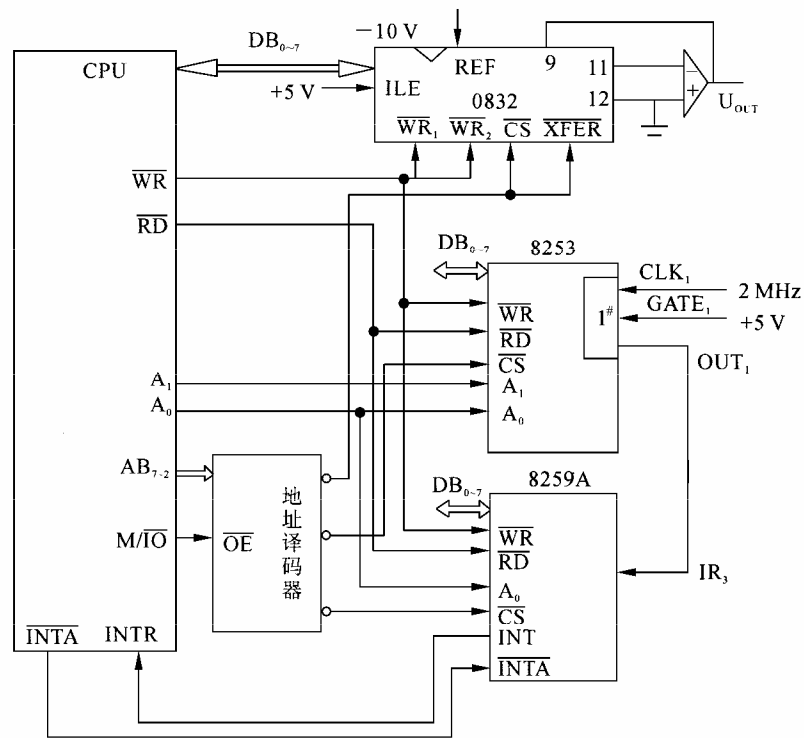
1. 在内存 BLOCK 单元开始存放 10 个无符号数, 从中找出最大值与最小值分别送入 MAX 与 MIN 单元中。
2. 在以 BLOCK 为首地址的内存单元中存放着 15 个字节数 ( 数据可以自己假设 ), 试将其中所有的负数依次存放在以 DEST 开始的存储单元中。
3. 在内存 BLOCK 开始存放了两个不带符号字, 求它们的积, 并存放在该两数之后的区域内。

### 五、接口程序设计题 ( 第 1 小题 15 分, 第 2 小题和第 3 小题各 20 分, 共 55 分 )

1. 利用 8255A 作为 CPU 与打印机的接口, 硬件连线如附图 1 所示。若要打印字符 “ A ”, 试编写此接口程序 (  $\overline{\text{STB}}$  为选通信号, BUSY 为忙信号 )。
2. 接口图如附图 2 所示, 写出每 10ms 向 0832 输送一次数据, 共输送 20 次的程序 ( 提示: 接口地址可合理假设; 20 个数据可以任意假设 )。注: 8259A 设置为单片, 边沿触发工作方式, 中断类型码为 60H ~ 67H。
3. 如附图 3 所示, 用 0809 对接在  $\text{IN}_3$  上的  $V_x(t)$  采样并转换 10 个点。若每个点转换后的数大于 0F0H, 则送 D/A, 否则不送。数都要保存在内存。写出汇编语言源程序。



附图1 硬件连线图

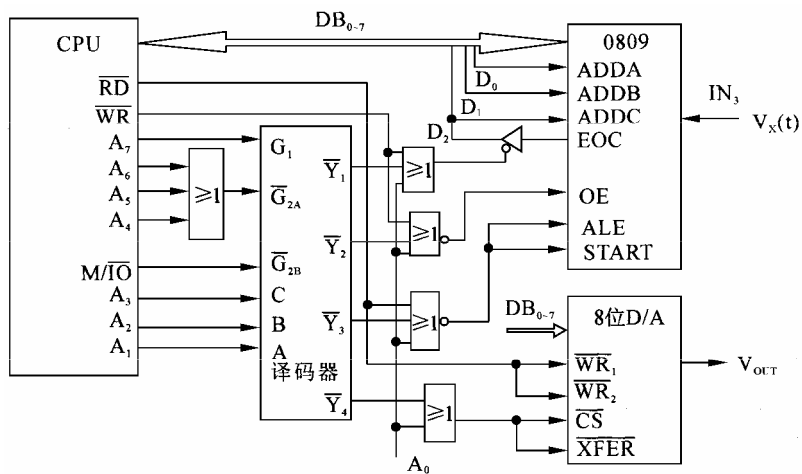


附图2 接口图

参 考 答 案

一、填空题

- 1 . 1234H ; 5400H
- 2 . 0FF80 ; 007E



附图 3 线路图

3. 直接寻址； 基址变址寻址； 相对基址寻址

4. DX

5. IF=1；有中断请求；现行指令执行完毕

6. 状态寄存器

7. 8；2；电流

## 二、选择题

1. B、C、D

2. A

3. A、D

4. A

5. A

## 三、问答题

1. 存储器映像寻址：指系统把 I/O 端口的地址和存储单元统一编址，此时，一个 I/O 口地址就是一个存储单元地址，对 I/O 端口的访问只能用存储器访问指令。I/O 映像寻址：指系统把 I/O 端口地址和存储单元地址分开编址，它们各有各的地址，使用不同的指令。

2. 外存和内存是以 CPU 能否直接访问这一原则来划分的。80486 CPU 能直接访问的最大内存空间为 4GB。

3. 定时和输入的 CLK 脉冲的频率（周期）有关，而计数和 CLK 频率无关。

	OUT	GATE
方式 0	一次负方波	电平控制
方式 1	一次负方波	脉冲触发
方式 2	周期性负脉冲	电平控制
方式 3	周期性方波	电平控制
方式 4	一次负脉冲	电平控制
方式 5	一次负脉冲	脉冲触发

4. 从 8259A 内部编程部件的工作过程可知, 8259A 的中断屏蔽寄存器 IMR, 决定是否让中断请求寄存器 IRR 锁存的中断请求进入优先权判别器 PR, 也就是说, IMR 中的 8 位二进制控制的是对应的 8 个中断源请求是否通过 8259A 向 CPU 申请中断; 而中断允许标志 IF 是 CPU 对外部可屏蔽中断的总屏蔽/允许位, 控制了 CPU 是否响应 8259A 的中断请求。所以, 只有当 IF 被设置为 1, 8259A 某引脚外部中断源在 IMR 中对应位为 0 时, 该引脚上有效的中断请求才有可能得到 CPU 的响应。

5. 在实方式和虚拟 8086 方式下, 段寄存器中存放的是段基址, 即段首地址的高 16 位; 而在保护模式下, 段寄存器中存放的是选择符, 包括选择码、描述符表类型标志和特权标志。

#### 四、汇编程序设计题

```
1.      .MODEL SMALL
        .DATA
BLOCK   DB 60,90,0,50.....
MAX     DB ?
MIN     DB ?
        .CODE
        .STARTUP
        MOV CX,9                ; 循环次数送 CX
        MOV AL,BLOCK
        MOV DL,BLOCK
        MOV BX,OFFSET BLOCK
AGAIN:  INC BX
        CMP AL,[BX]
        JAE NEXT1
        MOV AL,[BX]             ; AL 中存较大数
        JMP NEXT2
NEXT1:  CMP DL,[BX]
        JBE NEXT2
        MOV DL,[BX]             ; DL 中存较小数
NEXT2:  LOOP AGAIN
        MOV MAX,AL
        MOV MIN,DL
        .EXIT
        END

2.      .MODEL SMALL
        .DATA
BLOCK   DB 2,-3,5,-6.....      ; (15 个带符号数)
DEST    DB 15 DUP(?)
        .CODE
        .STARTUP
```

```

MOV BX,OFFSET BLOCK
MOV DI,OFFSET DEST
MOV CX,15
AGAIN: MOV AL,[BX]
      CMP AL,0
      JGE NEXT
      MOV[DI],AL
      INC DI
NEXT:  INC BX
      LOOP AGAIN
      .EXIT
      END
3.    .MODEL SMALL
      .DATA
BLOCK DW 1010H, 2030H,?,?
      .CODE
      .STARTUP
      MOV BX,OFFSET BLOCK
      MOV AX,[BX]
      MUL WORD PTR[BX+2]
      MOV[BX+4],AX           ; 积的低 16 位
      MOV[BX+6],DX           ; 积的高 16 位
      .EXIT
      END

```

### 五、接口程序设计题

1 . 分析接口图可知 : A 口、B 口、C 口及控制口地址分别为 90H、94H、98H、9CH , 接口程序如下所示。

```

MOV AL,81H
OUT 9CH,AL           ; 8255A 初始化
W:  IN AL,98H         ; 读 C 口
    TEST AL,08H       ; 检测 PC3 , 判断打印机是否忙
    JNZ W             ; 若 “ 忙 ” , 则等待
    MOV AL, 'A'       ; 否则 , 字符 “ A ” 送 A 口
    OUT 90H,AL
    MOV AL,00001110B   ; PC7=0 , 选通打印机
    OUT 9CH,AL
    INC AL
    OUT 9CH,AL

```

2. 假设 8253 的 0<sup>#</sup>、1<sup>#</sup>、2<sup>#</sup> 计数器及控制寄存器的地址分别为 70H、71H、72H、73H；设 8259A 的地址为 20H 和 21H，设 0832 的地址为 50H，并设 20 个数据已存放在以 BLOCK 为首的内存缓冲区中。程序片段如下所示。

```

        PUSH DS                ; 建立中断向量表
        MOV AX, 0
        MOV DS, AX
        MOV SI, OFFSET XY      ; XY 为中断服务程序入口处
        MOV DI, SEG XY
        MOV[018CH], SI        ; 63H×4=018CH
        MOV[018EH], DI
        POP DS                ; 以上指令建立中断向量
        MOV AL, 13H            ; 初始化 8259A
        OUT 20H, AL            ; ICW1
        MOV AL, 60H
        OUT 21H, AL            ; ICW2
        MOV AL, 01H
        OUT 21H, AL            ; ICW4
        MOV AL, 01110100B      ; 初始化 8253
        OUT 73H, AL            ; 设置控制字
        MOV AL, 20H
        OUT 71H, AL
        MOV AL, 4EH
        OUT 71H, AL            ; 给 1# 计数器送计数初值 N=20000=4E20H
        MOV CX, 20
        LEA BX, BLOCK
AGAIN:  HLT                    ; 等待中断
        LOOP AGAIN
; 中断服务程序
XY:     MOV AL, [BX]
        OUT 50H, AL
        INC BX
        IRET

```

3. 根据译码电路分析可知，0809 的 START 地址为 86H，EOC 地址为 82H，OE 地址为 84H；D/A 转换器的地址为 88H。程序如下所示。

```

        .MODEL SMALL
        .DATA
BLOCK   DB 10 DUP(?)
        .CODE
        .STARTUP
        MOV BX, OFFSET BLOCK
        MOV CX, 10

```

```

AGAIN: MOV AL,03H
        OUT 86H,AL          ; 启动 0809, 选择 IN3 通道
        NOP
        NOP                  ; 延时, 避免虚假 EOC 信号
W:      IN AL,82H
        TEST AL,04H
        JZ W                ; 检测 EOC 信号, 若未转换完毕, 继续检测等待
        IN AL,84H            ; 若转换完, 读数据
        CMP AL,0F0H          ; 转换数据与 0F0H 比较
        JBE DOWN
        OUT 88H,AL           ; 若大于 0F0H, 送 D/A 转换
DOWN:   MOV[BX],AL           ; 数据存入内存
        INC BX
        LOOP AGAIN
        .EXIT
        END

```

## 附录 2 2000 年某高校硕士研究生入学“微机原理及应用”考试试题及答案

### 一、填空题（共 15 分）

1. 已知 AL=10011100B, BL=01101000B, 执行指令 SUB AL, BL 后, AL=\_\_\_\_\_, 标志位 CF=\_\_\_\_\_, OF=\_\_\_\_\_, PF=\_\_\_\_\_。
2. 8086 CPU 在执行 8 位带符号数运算后, 产生溢出是由于运算结果超出了\_\_\_\_\_。
3. 若指令 IMUL DL 执行后 CF=OF=1, 则表示\_\_\_\_\_。
4. 8086 CPU 判断两个带符号数大小的标志位为\_\_\_\_\_, \_\_\_\_\_、\_\_\_\_\_。
5. 若 SUBPROC 为过程名, 设当前 SP=6000H, 8086 CPU 执行指令 CALL FAR PTR SUBPROC 后, SP=\_\_\_\_\_。
6. 若 AGAIN 为标号, 指令 JMP SHORT AGAIN 为\_\_\_\_\_寻址方式, 其转移地址为\_\_\_\_\_。
7. 若 8086 CPU 可屏蔽中断的类型码为 60H, 则其中断向量表的入口地址指针为\_\_\_\_\_。若该中断源的中断服务程序入口地址为 5A000H 时, 则该中断向量表入口地址指针对应的 4 个字节, 从地址低位开始依次为\_\_\_\_\_, \_\_\_\_\_、\_\_\_\_\_, \_\_\_\_\_（设 CS=5000H）。
8. 若异步传送字符的 ASCII 码, 每帧 7 位数据, 偶校验, 1 位停止位, 当波特率为 2 400 bps 时, 每秒可传送\_\_\_\_\_个字符。

### 二、问答题（7 分）

简述 8086 CPU 总线接口部件 BIU 和执行部件 EU 并行工作的过程。



## 三、(10分)

试编写程序统计数据段字节变量 NUM 单元中“1”的个数,并且将统计值存入字节变量 COUNT 单元。

## 四、(10分)

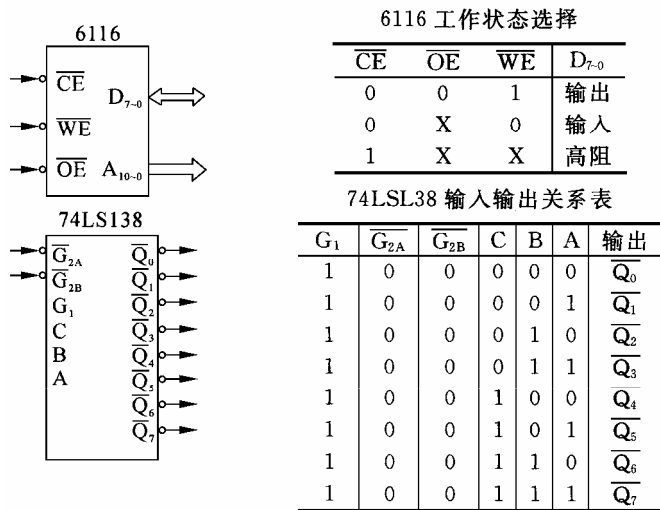
在数据段有一字类型数组,其首地址 BUFFER,数组中第一个字单元存放着数组长度,要求编写一个程序,使该数组按照从小到大的顺序排列整齐。如果经过若干次排序,数组已经整序完毕,则立刻退出排序操作。程序中有3处空白,请填上适当的语句,并且请对程序中4处错误进行改正。

```
1) DATA    SEGMENT
2) BUFFER DB 6,88H,22H,66H,44H,55H,77H
3) DATA    ENDS
4) CODE     SEGMENT
5)          ASSUME CS:CODE,DS:DATA
6) START:   MOV AX,DATA
7)          MOV DS,AX
8)          MOV DX,BUFFER
9)          LEA DI,BUFFER
10) LOOP1:  MOV BX,1
11)          DEC DX
12)          JNZ DONE
13) LOOP2:  ADD DI,2
14)          MOV AX,[DI]
15)          CMP[DI+2],AX
16)          JAE NEXT
17)          (1)_____
18)          (2)_____
19)          SUB BX,BX
20) NEXT:   LOOP(3)_____
21)          MOV CX,DX
22)          CMP BX,0
23)          JE LOOP1
24) DONE:   MOV AH,4CH
25)          INT 21H
26) CODE    ENDS
27)          END START
```

## 五、(10分)

某8位微机系统内存RAM为32KB,现须扩充内存RAM16KB,要求扩充地址与原系统地址连续,RAM芯片采用2K×8位RAM6116,地址译码器采用74LS1383-8译码器,

所需门电路可自行选择，试画出扩充内存的系统连接图。并说明扩充内存地址范围（设地址线 16 位，见附图 4 所示）。



附 图 4

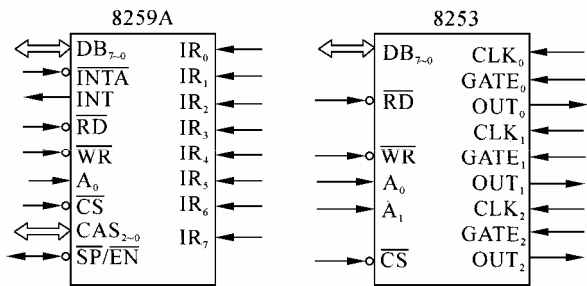
#### 六、(23 分)

试采用 8253 和 8259A 实现下述功能。

(1) 采用 8253 计数器 0 作为外部控制对象的硬件时钟，当控制对象发正脉冲信号启动计数器 0 工作后，计数器 0 延时 5ms 向控制对象发出负脉冲时钟信号（8253  $CLK_0$  时钟频率为 1MHz，由外部时钟发生器提供）。

(2) 采用 8253 计数器 1 接收外部控制对象计数脉冲，每接收 100 个计数脉冲向 8259A 送一次中断请求，该中断源中断类型码为 63H，8259A 工作在单片、全嵌套、非缓冲和边沿触发工作方式。

要求：试画出 8253、8259A 与 8086 CPU 的系统连接图，并且编写 8253 和 8259A 初始化程序（如附图 5 所示）。



附 图 5

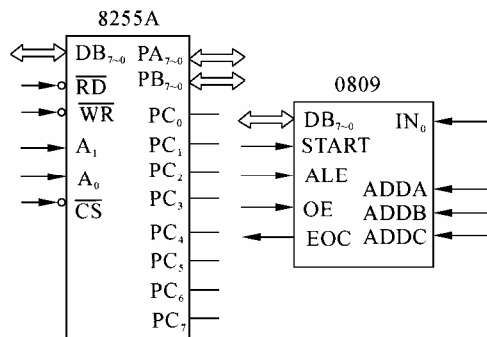
### 七、(25分)

(如附图6所示)试采用8255A和0809实现对控制对象的控制,具体要求如下所示。

电路设计实现功能:8086 CPU经8255A和0809连接。控制对象将一路模拟量 $V_x$ 送0809。8086 CPU经8255A接收0809输入的数字量,并且由8255A采用查询工作方式控制0809工作。8086 CPU经8255A向控制对象发送电平控制信号。

具体对控制对象的控制过程为:8086 CPU接收控制对象送出的数据,如果数值小于80H,则向控制对象发送高电平控制信号;如果数值大于80H,则向控制对象发送低电平控制信号。

根据上述要求,试画出8086 CPU与8255A、0809及控制对象的系统连接图,并且编写初始化及工作程序(可合理假设端口地址)。



附图 6

## 参 考 答 案

### 一、填空题

1. 34H ; 0 ; 1 ; 0
2. -128 ~ +127
3. 乘积高位 AH 数据有效
4. SF ; OF ; ZF
5. 5FFCH
6. 段内短转移; 当前 IP 值与目标地址 AGAIN 确定的 8 位位移量之和
7. 180H ; 00H ; A0H ; 00H ; 50H
8. 240

### 二、问答题

- (1) 当指令队列中有两个空字节时, BIU 会自动取指到指令队列。
- (2) EU 不需等待, 直接从指令队列取指、译码和执行指令。

(3) EU 在执行指令时,如需访问内存或外设,则 EU 会向 BIU 发出访问内存或外设的请求。

(4) 若 BIU 当前无取指操作,则响应 EU 的请求,完成访问内存或外设的操作。

(5) 指令队列已满,EU 又无请求,BIU 处于空闲状态。

(6) 当 EU 执行调用或转移类指令时,指令队列内容被 BIU 消除。

(7) BIU 以新地址取指立即送 EU,接着取入后继指令填满指令队列。

### 三、答案

```
DATA        SEGMENT
NUM          DW 5566H
COUNT      DW ?
DATA        ENDS
CODE        SEGMENT
            ASSUME CS:CODE,DS:DATA
START:      MOV AX,DATA
            MOV DS,AX
            MOV CX,0
            MOV AX,NUM
REPEAT:     TEST AX,0FFFFH
            JZ  DONE
            JNS SHIFT
            INC CX
SHIFT:      SHL AX,1
            JMP REPEAT
DONE:       MOV COUNT,CX
            MOV AH,4CH
            INT 21H
CODE        ENDS
            END  START
```

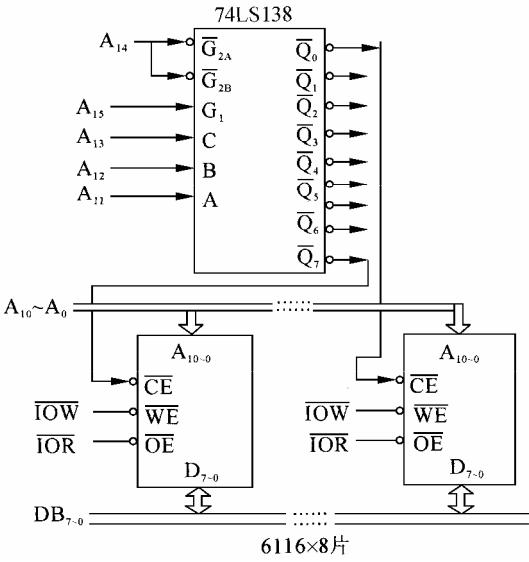
### 四、

```
1) DATA    SEGMENT
2) *BUFFER  DW 6, 88H, 22H, 66H, 44H, 55H, 77H
3) DATA    ENDS
4) CODE     SEGMENT
5)          ASSUME CS:CODE,DS:DATA
6) START:   MOV AX, DATA
7)          MOV DS, AX
8)          MOV DX,BUFFER
9) * LOOP1: LEA DI,BUFFER
10)         MOV BX,1
11)         DEC DX
```

```
12) *      JZ  DONE
13) *      MOV CX, DX
14) LOOP2:  ADD DI, 2
15)        MOV AX, [DI]
16)        CMP [DI+2], AX
17)        JAE NEXT
18)        (1) XCHG [DI+2], AX
19)        (2) MOV [DI], AX
20)        SUB BX, BX
21) NEXT:   LOOP (3) LOOP2
22)        CMP BX, 0
23)        JE LOOP1
24) DONE:   MOV AH, 4CH
25)        INT 21H
26) CODE    ENDS
27)        END START
```

五、

扩充内存地址范围为 8000H ~ 0BFFFH。扩展图如附图 7 所示。



附图 7 扩展图

六、

(1) 计数器 0，方式 5，N=5000

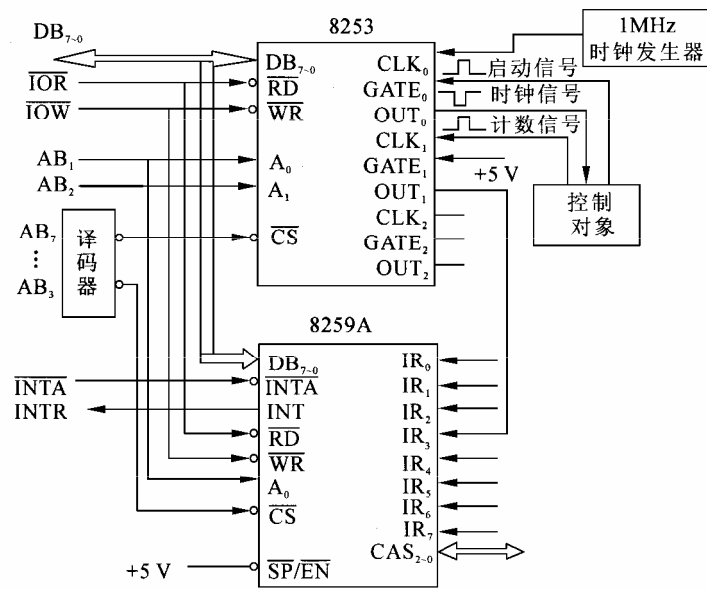
```
MOV AL,00101011B      ; 8253 初始化
OUT 86H,AL
```

```
MOV AL,50H
OUT 80H,AL
```

(2) 计数器 1，方式 2，N=100

```
MOV AL,01010100B      ; 8253 初始化
OUT 86H,AL
MOV AL,100
OUT 82H,AL
MOV AL,00010011B      ; 8259A 初始化
OUT 88H,AL
MOV AL,60H
OUT 8AH,AL
MOV AL,01H
OUT 8AH,AL
```

接口图如附图 8 所示。



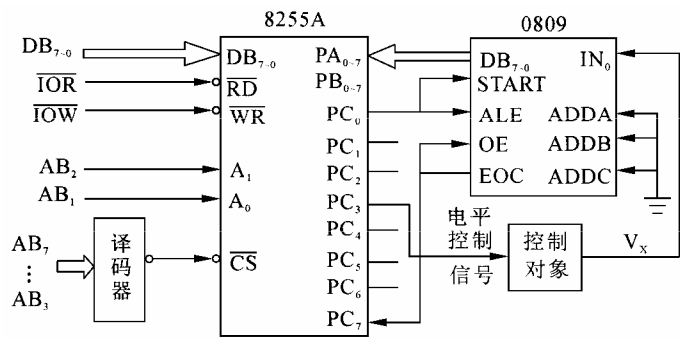
附 图 8

七、

```
CODE SEGMENT
ASSUME CS:CODE
START: MOV AL, 10011000B      ; A 入 , C 上入 , C 下出
      OUT 86H, AL            ; 设 8255A 的 A、B、C 及控制口地址为
                              ; 80H、82H、84H、86H
LP:    MOV AL, 0
```

```
OUT 86H, AL
INC AL
OUT 86H, AL          ; PC0置1,启动 0809
DEC AL
OUT 86H, AL
NOP
WAIT: IN AL,84H      ; 延时 8T+2us 后
TEST AL,80H
JZ WAIT              ; 测试 EOC
IN AL,80H             ; A 口读入
CMP AL,80H
JB LOW
CMP AL,0F0H
JA LP
MOV AL, 06H
OUT 86H,AL           ; PC3置0
JMP LP
LOW:  MOV AL,07H
OUT 86H,AL           ; PC3置1
JMP LP
CODE ENDS
END START
```

接口图如附图 9 所示。



附图 9 接口图

附录 3 2001 年某高校硕士研究生入学“微机原理及应用”考试试题及答案

一、问答题（每小题 2 分，共 30 分）

- (1) 什么是微处理器的字长？8086 和 80486 的字长是多少位？
- (2) 计算机是能够自动完成算术运算和逻辑运算等的电子装置，那为什么在它的运

算器中只有加法器？

- (3) 80486 CPU 内部由哪几部分组成？
- (4) 80486 有哪几种工作方式？
- (5) 8086 存储器操作数的段基址、偏移地址、段的首地址、物理地址各是多少位？80486 的段基址、偏移地址、段的首地址、物理地址又各是多少位？
- (6) 什么叫串？串操作有哪些基本的指令？在使用时它们的寻址方式有哪些约定？在什么情况下使用串前缀？
- (7) 什么是指令语句？什么是伪指令语句？它们的主要区别是什么？
- (8) 在汇编语言源程序中可用于设计简单分支程序的指令有哪些？循环程序结构有哪两类？
- (9) 8086 对内存最大寻址范围是多大？80486 对内存最大寻址范围是多大？什么是虚拟存储器？32 位 CPU 可寻址的最大虚拟空间是多大？
- (10) CPU 与外设传递的信息有哪几方面内容？为什么必须通过接口电路？
- (11) CPU 与外设传递信息的控制方式有哪几种？
- (12) 分别简述 8086 CPU 和 80486 CPU 得到了中断类型码之后，如何找到中断服务程序的入口地址。
- (13) 简述 8259A 中断控制器的主要功能是什么？
- (14) 8255A 在微机系统中起什么作用？它有几种工作方式？它有几个口地址，由什么来决定？
- (15) 8253 是计数器/定时器，那么计数和定时的区别是什么？8253 各种工作方式在 OUT 输出特性及 GATE 信号上有什么区别？

二、(10 分)

若 X、Y、Z 是连续存放在内存 BLOCK 开始的 3 个带符号字节数，试编写出计算  $X \times Y - Z$  的完整的汇编语言源程序。

三、(10 分)

若在内存中有两串 10 个字节不带符号数，将其对应单元内容相减，结果再送回两串中任一串的位置。写出简化的汇编语言源程序。

四、(15 分)

一打印机引脚信号如附图 10 所示。

2	4	6	8	10	12	14	16	18	20
GND	GND	GND	GND	GND	GND	GND	GND	ACK	ERR
STB	DB <sub>0</sub>	DB <sub>1</sub>	DB <sub>2</sub>	DB <sub>3</sub>	DB <sub>4</sub>	DB <sub>5</sub>	DB <sub>6</sub>	DB <sub>7</sub>	BUSY
1	3	5	7	9	11	13	15	17	19

附图 10 打印机引脚信号



STB : 输入数据选通信号。该信号有效, 则打印机接收数据。

ACK : 是对信号的“ 应答 ”信号输出线，此信号有效，表示打印机已得到数据。

注：只有在  $\overline{\text{ACK}}$  有效、 $\text{BUSY}$  和  $\overline{\text{ERR}}$  无效情况下，打印机才能接收新的数据进行  
]。

五、(15分)

附 图 11

六、(20 分)

## 参 考 答 案

(1) 一次运算或传送的最长的二进制位数称为字长。8086 和 80486 的字长都定义为 16 位, 但 80486 一次能处理的二进制位数可达 32 位。

(2) 因为减法是通过加补码实现的；乘法是采用部分积右移即右移和加实现的；除法是采用部分余数左移加除数补码实现的。

(3) 由总线接口部件、高速缓存、指令预取、译码电路、浮点部件、执行部件、分段管理部件和分页部件 8 大部分组成。

(4) 有 3 种工作方式：实地址方式、保护方式和虚拟 8086 方式。

(5) 8086 存储器操作数的段基址是 16 位、偏移地址是 16 位、段的首地址 20 位、物理地址是 20 位；80486 的段基址 32 位、偏移地址是 32 位、段的首地址 32 位、物理地址是 32 位。

(6) 串是一组有序连续存放的数据。串基本操作指令有串传送 MOVSB/MOVSX、串比较 CMPSB/CMPSX、串扫描 SCASB/SCASX、串存储（写串）STOSB/STOSX、串装入（读串）LODSB/LODX 共 5 种。串前缀是在串操作过程中无须再做其他处理的情况下使用的。

(7) 指令语句经过汇编以后要产生机器码，而伪指令语句不产生机器码；指令语句和机器的一种操作相对应，而伪指令语句和机器的一种伪操作（汇编过程）相对应。

(8) 用于设计分支程序的指令有：CMP 目的，源（比较指令）；TEST 目的，源（测试指令）；SHL 目的，源（移位指令）等。

用于设计循环程序的结构有：

```

MOV CX, n          ; 送循环次数
A1: ...
    ... 循环体
    ...
    LOOP A1
    ...
MOV CX, n          ; 送循环次数
B1: JCXZ A1
    ...
    ... 循环体
    ...
    JMP B1
A1: ...

```

(9) 8086 对内存最大寻址范围是 1MB，80486 对内存最大寻址范围是 4GB，虚拟存储器是将外存的一部分当作内存来使用。32 位 CPU 可寻址的最大虚拟空间是 64KB。

(10) 有数据、状态、控制 3 种信息。必须通过接口电路，因为外设提供的信息种类多，有数字信息（数字信息还有串行的和并行的）、模拟信息、脉冲信息、开关信息等，都要通过一些接口电路转换成计算机能处理的信息形式。其次是外设工作速度慢，必须通过接口电路实现与 CPU 速度匹配。另外还有一些如电平转换等功能。

(11) 有无条件传送方式、查询传送方式、中断传送方式、DMA 传送方式和 I/O 处理机方式。

(12) 在 8086 中, 中断类型码 $\times 4$  得到矢量表的矢量地址的偏移地址, 其段基址是 DS=0000H, 从矢量表中得到中断服务程序的入口地址。而在 80486 中, 中断类型码 $\times 8$  得到一个 32 位的地址, 从 IDT 表中查它的选择符, 用得到的选择符从 GDT 或 LDT 表中查段的描述符, 描述符中包含着中断服务程序的段基址, 段基址再和选择符中的 32 位偏移地址相加得到中断服务程序的入口地址。

(13) 8259A 中断控制器的主要功能是: 接受中断请求信号, 实施对中断请求信号的屏蔽、判优, 比较目前提出请求而且没有被屏蔽的中断中最高优先级是否比正处理的中断具有更高的优先权, 向 CPU 发出中断请求, 接收 CPU 中断响应信号, 输送中断类型码, 接受 CPU 的编程。

(14) 8255A 是并行可编程接口芯片。它有 3 种工作方式: 方式 0、方式 1 和方式 2。有 4 个口地址, 由地址译码输出线接在它的  $\overline{CS}$  上, 线对应的地址值和 A0 和 A1 一起决定 4 个地址。

(15) 定时和输入的 CLK 脉冲的频率 (周期) 有关, 而计数和 CLK 频率无关。

	OUT	GATE
方式 0	一次负方波	电平控制
方式 1	一次负方波	脉冲触发
方式 2	周期性负脉冲	电平控制
方式 3	周期性方波	电平控制
方式 4	一次负脉冲	电平控制
方式 5	一次负脉冲	脉冲触发

二、

```
DATA SEGMENT
BLOCK DB X, Y, Z
DATA ENDS
CODE SEGMENT
ASSUME CS: CODE, DS: DATA
START: MOV AX, DATA
MOV DS, AX
MOV BX, OFFSET BLOCK
MOV AX, 0
MOV AL, [BX]
IMUL BYTE PTR[BX+01H]
MOV CX, AX
MOV AL, [BX+02H]
CBW
SUB CX, AX
MOV AH, 4CH
INT 21H
CODE ENDS
```

END START

三、

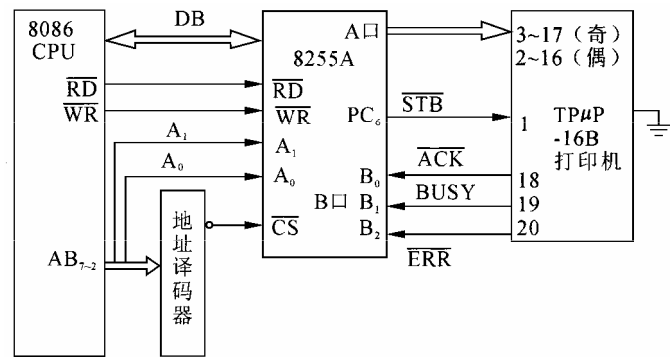
```
        .MODEL SMALL
        .DATA
SRC      DB 5,6,7 ..... (10 个字节数)
DEST     DB 1,2,3 ..... (10 个字节数)
        .CODE
        .STARTUP
MOV CX,10
LEA SI, SRC
LEA DI, DEST
MOV AX, @DATA
MOV ES, AX
CLD
A1:      LODSB
        SUB AL, [DI]
        STOSB
        LOOP A1
        .EXIT
END
```

四、

```
        .MODEL SMALL
        .DATA
BLOCK    DB 1,2,3 .....           ; (10 个字节数)
        .CODE
        .STARTUP
MOV BX, OFFSET BLOCK
MOV AL, 10000010B                   ; 8255A 初始化, 设地址为 50H ~ 53H
OUT 53H, AL
MOV CX, 10
IN AL, 51H                          ; 读 B 口
TEST AL, 04H
JZ ERR                               ; 错误处理
P1:      IN AL, 51H                  ; 读 B 口
        TEST AL, 02H                ; 测试 BUSY
        JNZ P1
P2:      MOV AL, 51H                 ; 测  $\overline{\text{ACK}}$ 
        TEST AL, 01H
        JNZ P2
```

```
MOV AL,[BX]                ; 送数
INC BX
OUT 50H,AL
MOV AL,00001100B           ; PC6=0, 选通  $\overline{STB}$ 
OUT 53H,AL
DEC CX
JNZ P1
.EXIT
END
```

接口图如附图 12 所示。



附 图 12

五、

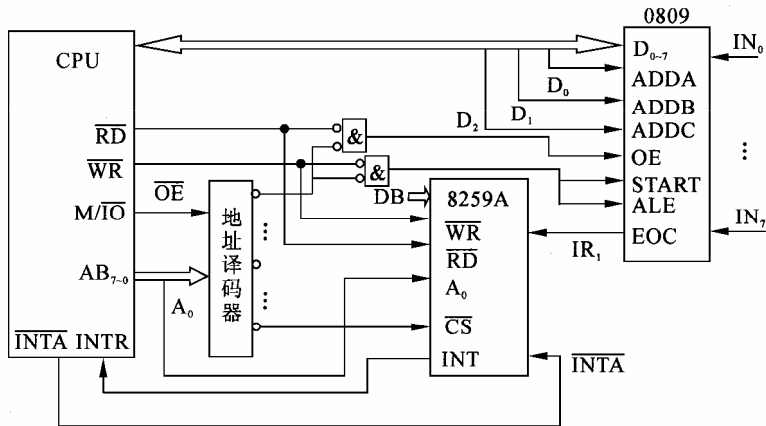
```
.MODEL SMALL
.DATA
BLOCK DB 1,2 .....      ; (100 字节数)
.CODE
.STARTUP
MOV CX,100
MOV AL,01100011B         ; 设 8253 地址为 70H~73H, 计数器 1 初始化
OUT 73H,AL
MOV AL,05H
OUT 71H,AL                ; 计数器 1 初值=500
MOV AL,10100101B
OUT 73H,AL
MOV AL,10H
OUT 72H,AL                ; 计数器 2 初值=1000
MOV BX,OFFSET BLOCK
P: IN AL,80H              ; 设 8212 地址为 80H
```

```

TEST AL,01H
JZ P
MOV AL,[BX]
OUT 20H,AL          ; 设 0809 地址为 20H
INC BX
DEC CX
JNZ P
.EXIT
END

```

六、  
接口图设计如附图 13 所示。



附图 13 接口图

```

.MODEL SMALL
.DATA
BLOCK DB 800 DUP(?)
.CODE
.STARTUP
MOV CX,100
MOV BX,OFFSET BLOCK
MOV AL,00010011B          ; 设 8259A 地址为 20H、21H、ICW1
OUT 20H,AL
MOV AL,60H                ; ICW2
OUT 21H,AL
PUSH DS                    ; 建立中断向量表
MOV AX,0
MOV DS,AX
MOV SI,OFFSET XY
MOV DI,SET XY

```

```

MOV[0184H],SI          ; 61H×4=0184H
MOV[0186H],DI
POP DS
PP1:  MOV DL,0          ; 通道 0
PP2:  MOV AL,DL
      OUT 98H,AL        ; 选择通道,并启动
      NOP
      HLT              ; 等待中断
      INC DL
      CMP DL,08H
      JNZ PP2
      DEC CX
      JNZ PP1
      .EXIT
END

```

中断服务程序如下所示。

```

XY:   PUSH AX
      IN AL,98H
      MOV[BX],AL
      INC BX
      POP AX
      IRET

```

## 附录4 2002年某高校硕士研究生入学 “微机原理及应用”考试试题及答案

### 一、问答题(每小题2分,共10分)

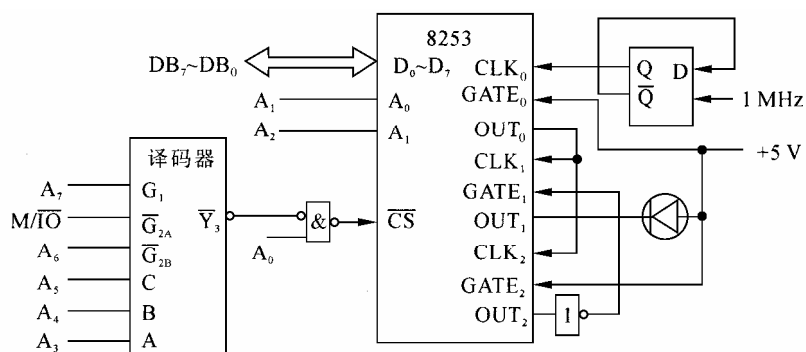
1. 微型计算机的主要特点是什么?
2. 为什么在计算机中采用二进制数?
3. 微型计算机主要由哪几部分组成?各部分的主要功能是什么?
4. 80486内部主要由哪几部分组成?有哪些通用寄存器?
5. 32位微型计算机内存管理有分段分页管理,分页管理主要解决了什么问题?

### 二、选择题(每小题2分,共10分,有单选题,也有多选题)

1. 80486中操作数的直接寻址是指\_\_\_\_\_。  
 A. 在某寄存器中寻找操作数      B. 操作数是在指令中  
 C. 操作数地址在指令中      D. 在寄存器中寻找指令
2. 80486的 $\overline{W}/\overline{R}$ 引脚的功能是\_\_\_\_\_。  
 A. 执行读操作

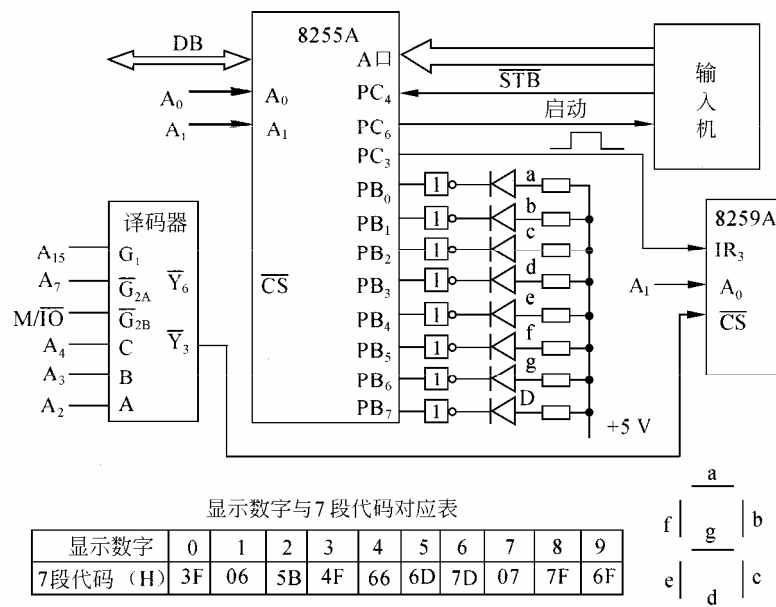






附图 14 接口简化图

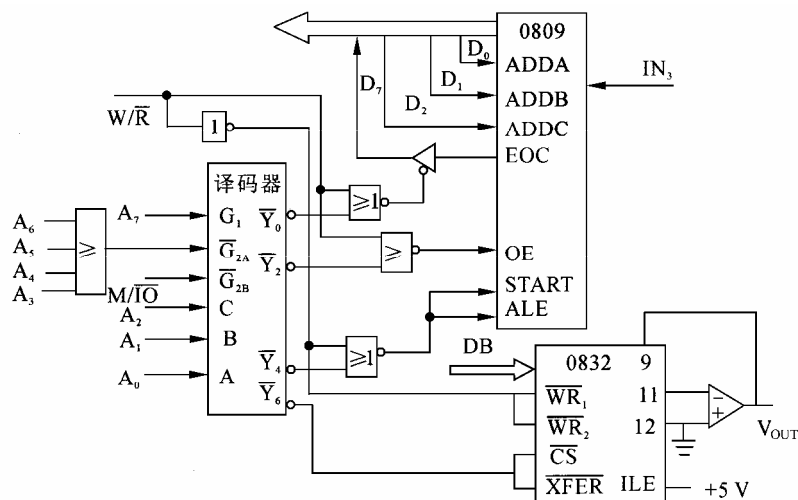
接口简化图如附图 15 所示, 8255A 的 B 口接七段显示器发光二极管 a、b、c、d、e、f、g, 要求从 0~9 这 10 个数字不断重复显示, 在显示的过程中 DP 一直不亮。每个数字显示的时间是 DEC 和 JNZ 指令重复执行 500 次的时间。在重复显示 0~9 数字的同时, 从输入机读入 100 个数据。100 个数据输入完, 显示也停止。设 8259A 的 8 个中断类型码为 70H~77H, 边沿触发。编写出实现以上要求的程序(可省略伪指令)。



附图 15 接口简化图

有一接口简化图(如附图 16), 在 0809 的  $IN_3$  上接了一个  $V(t)$  电压, 要求连续对该电压采样并转换 100 个点。同时要求, 每转换好一个数据, 在保存到内存之后再送

给 D/A 转换器 0832，写出实现该要求的程序（可省略伪指令）。



附图 16 接口简化图

## 参 考 答 案

### 一、问答题

1. 体积小、重量轻、可靠性高、价格便宜、使用方便。
2. 容易实现，运算规则简单。
3. 由运算器、控制器、存储器、I/O 接口及输入设备和输出设备等组成。运算器是对信息加工的部件，控制器是指挥各部件工作的部件，存储器是信息仓库，所有程序、原始数据、运算的中间结果和最终结果都保留在存储器中，I/O 接口及输入和输出设备是用来完成计算机与外部传递信息的。
4. 由总线接口部件、高速缓存部件、代码预取部件、指令译码部件、浮点数组成。部件、执行部件、分段部件、分页部件及总线等组成。通用寄存器有 EAX、EBX、EDX、ECX、ESP、EBP、ESI、EDI；若当作 16 位使用，相应为 AX、BX、CX、DX、SP、BP、SI、DI；若当作 8 位使用，为 AH、AL、BH、BL、CH、CL、DH、DL。
5. 因为 32 位机可直接对内存的寻址最大范围为  $2^{32}=4\text{GB}$ ，一段最大范围可达 4GB，为了减少内存碎片，所以要分页，一页为 4KB。

### 二、选择题

1. C
2. C
3. C
4. B、E、F
5. A

### 三、填空题

1.  $[X+Y]_{\text{补}}=[X]_{\text{补}}+[Y]_{\text{补}}=00000000\text{B}$ （无溢出）； $[X-Y]_{\text{补}}=[X]_{\text{补}}+[-Y]_{\text{补}}=10010110\text{B}$ （溢出）

2. 1; 串长度

3. CX; CL

4. 高; 低

5. CMP 指令是比较 DEST 和 SRC 两个操作数的大小或是否相等, 而 TEST 指令是测试 DEST 某一位是 1 还是 0, 或者某些位是否同时为 0。

四、

```

        .MODEL SMALL
        .DATA
XYZ      DW 1020,.....(10 个数)
A1       DW 10DUP(?)
B1       DW 10DUP(?)
        .CODE
        .STARTUP
MOV CL,10
MOV BX,OFFSET XYZ
MOV SI,OFFSET A1
MOV DI,OFFSET B1
P:       MOV AX,[BX]
        TEST AX,8000H
        JZ W1
        MOV[DI],AX
        INC DI
        INC DI
        JMP Q
W1:      MOV[SI],AX
        INC SI
        INC SI
Q:       INC BX
        INC BX
        DEC CL
        JNZ P
        .EXIT
END
    
```

五、

```

        .MODEL TINY
        .DATA
WP       DB 10H,20H,30H
        .CODE
        .STARTUP
MOV BX,OFFSET WP
MOV AL,[BX]
MOV DL,[BX+1]
    
```

```

MUL DL
MOV CL,[BX+2]
MOV CH,0
SUB AX,CX
DIV CL
MOV[BX+3],AL           ; 商
MOV[BX+4],AH           ; 余数
.EXIT
END

```

### 六、

由接口图分析可知，0<sup>#</sup>、1<sup>#</sup>、2<sup>#</sup>计数器及控制寄存器的地址分别为 99H、9BH、9DH、9FH。根据题意，0<sup>#</sup>计数器应工作在方式 2，计数值 N<sub>0</sub>=2000；1<sup>#</sup>计数器应工作在方式 3，计数值 N<sub>1</sub>=1000；2<sup>#</sup>计数器应工作在方式 0，计数值 N<sub>2</sub>=5000。程序片段如下所示。

```

MOV AL,00100101B
OUT 9FH,AL
MOV AL,01100111B
OUT 9FH,AL
MOV AL,10100001B
OUT 9FH,AL
MOV AL,20H
OUT 99H,AL
MOV AL,10H
OUT 9BH,AL
MOV AL,50H
OUT 9DH,AL

```

### 七、

由接口图分析可知，8255A 的地址为 8018H~801BH，8259A 的地址为 800CH 和 800EH。程序片段如下所示。

```

TAB      DB 3FH,06H,5BH,4FH,66H,6DH,7DH,07H,7FH,6FH
          DB 100 DUP(?)
MOV AL,10110000B           ; 8255A 命令字
MOV DX,801BH
OUT DX,AL
MOV AL,00001001B           ; PC4=1，允许中断
OUT DX,AL
PUSH DS                     ; 建立中断向量表
MOV AX,0
MOV DS,AX
MOV BX,OFFSET XY
MOV DI,SEG XY
MOV[01CCH],BX              ; 73H×4=01CCH

```

```

MOV[01CEH],DI
POP DS
MOV DX,800CH
MOV AL,00010011B           ; 8259A ICW1
OUT DX,AL
MOV DX,800EH
MOV AL,70H                 ; ICW2
OUT DX,AL
MOV AL,03H                 ; ICW3
OUT DX,AL
MOV AL,03H                 ; ICW3
OUT DX,AL
MOV CL,100
MOV BX,OFFSET TAB
MOV SI,BX+10
MOV DI,SI
MOV AL,00001101B           ; PC6=1, 启动
MOV DX,801BH
OUT DX,AL
MOV DX,8019H
Q:  MOV AX,[BX]              ; 取显示码
    OUT DX,AL               ; 送 B 口
    MOV AX,500
P:   DEC AX
    JNZ P                   ; 延时, 每个数字显示的时间
    INC BX
    CMP BX,SI
    JB Q
    SUB BX,10
    JMP Q
                                ; 中断服务程序
XY:  PUSH AX
    PUSH CX
    PUSH BX
    MOV DX,8018H
    IN AL,DX
    MOV[DI],AL
    INC DI
    DEC CL
    JNZ RP
    JMP PQ
RP:  POP BX
    POP CX
    POP AX
    IRET
PQ:  MOV AL,00001100B         ; PC6=0, 关输入机

```

```
MOV DX,801BH
OUT DX,AL
HLT
```

八、

```
TAB DB 100 DUP(?)
MOV CL,100
MOV BX,OFFSET TAB
Q: MOV AL,03H
OUT 84H,AL ;启动 0809
NOP
P: IN AL,80H ;测 EOC 状态
TEST AL,80H
JZ P
IN AL,82H ;读数
MOV[BX],AL
INC BX
OUT 86H,AL ;送 0832
DEC CL
JNZ Q
HLT
```

## 附录 5 2003 年某高校硕士研究生入学 “微机原理及应用”考试试题及答案

一、问答题（每小题 3 分，共 15 分）

1. 计算机设计采用二进制的优点是什么？
2. 什么是运算溢出？什么是串行通信溢出错？
3. 80486 CPU 对于外部的可屏蔽中断和非屏蔽中断是分别通过哪两个引脚接收的？在什么条件下分别响应它们？
4. 指令语句和伪指令语句的结构和功能有什么不同？
5. 串的概念是什么？DF 的作用是什么？串操作的寻址方式是什么？串前缀在什么情况下使用？

二、填空题（每小题 4 分，共 20 分）

1. 已知 $[X]_{\text{补}}=10011101\text{B}$ ， $[Y]_{\text{补}}=11001001\text{B}$ ， $[Z]_{\text{补}}=00100110\text{B}$ ，计算 $[X+Y]_{\text{补}}$ ，并指出是否溢出；计算 $[X-Z]_{\text{补}}$ ，并指出是否溢出。
2. 指令 LOOPZ/LOOPE 是\_\_\_\_\_且\_\_\_\_\_发生循环的指令；而指令 LOOPNZ/LOOPNE 是\_\_\_\_\_且\_\_\_\_\_发生循环的指令。
3. 80486 在实地址方式下段基址为\_\_\_\_\_位，而在保护方式下段基址为\_\_\_\_\_。

193

附  
录

\_\_\_\_\_位,它们分别由\_\_\_\_\_和\_\_\_\_\_提供。

4. 串行通信中波特率是指\_\_\_\_\_,波特率系数是指\_\_\_\_\_。

5. 设  $SS=0FFA0H$ ,  $SP=00B0H$ ,  $AX=8057H$ ,  $BX=0F79H$ , 执行指令  $PUSH\ AX$  后,  
 $SP=$ \_\_\_\_\_ ; 若再执行指令:

$PUSH\ BX$

$POP\ AX$

后,  $SP=$ \_\_\_\_\_ ;  $AX=$ \_\_\_\_\_ ;  $BX=$ \_\_\_\_\_。

### 三、选择题(每小题3分,共15分,有单选和多选)

1. 可以用一条指令改变的标志位有\_\_\_\_\_。

A. CF      B. ZF      C. DF      D. IF      E. AF

2. 实现  $AL$  寄存器中  $D7$  和  $D0$  都取反的指令是\_\_\_\_\_。

A.  $AND\ AL, 7EH$       B.  $OR\ AL, 81H$   
 C.  $TEST\ AL, 81H$       D.  $XOR\ AL, 81H$

3. 8253 需要由外部脉冲触发才开始计数的工作方式是\_\_\_\_\_。

A. 方式 0      B. 方式 1      C. 方式 2  
 D. 方式 3      E. 方式 4      F. 方式 5

4. 8255A 工作在方式 0 时,通过它可与外设进行信息传递的控制有\_\_\_\_\_。

A. 无条件传送方式      B. 中断传送方式  
 C. 查询传送方式      D. DMA 传送方式

5. 8259A 是用来实现\_\_\_\_\_。

A. 接收中断请求      B. 执行中断服务处理  
 C. 中断管理与通用数据传送      D. 中断源判优  
 E. 向 CPU 输送中断类型码

### 四、汇编程序设计(每小题20分)

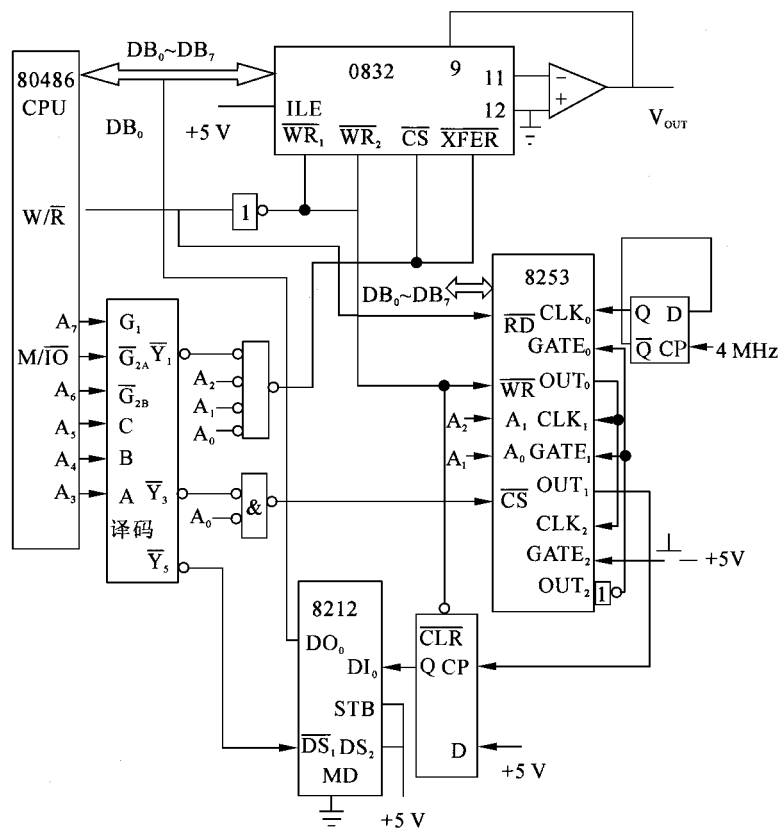
1. 有两个十进制数:  $X=7654433221$  和  $Y=8776655443$ , 分别以组合 BCD 码形式存放在内存  $A1$  和  $A2$  开始的缓冲区内。计算  $Z=X+Y$ , 并将  $Z$  存放在  $A3$  开始的缓冲区内。

2. 在内存  $BLOCK$  开始的缓冲区内有 5 个无符号字  $a$ 、 $b$ 、 $c$ 、 $d$ 、 $e$ , 计算  $[a+b-c] \times d/e$ , 并将结果存放在数的后面。

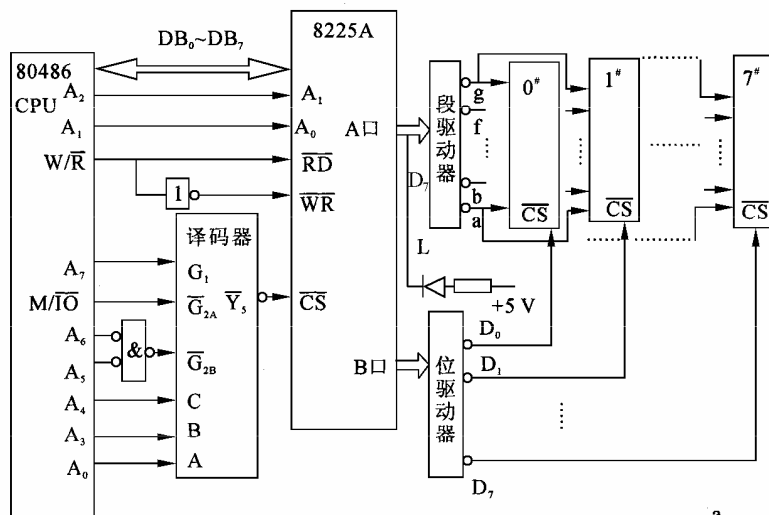
### 五、接口程序设计(每小题20分)

1. 硬件电路如附图 17 所示,要求每 50 ms 向 0832 送一个数据,该过程共进行 10 s,数据存放在内存  $BLOCK$  开始的缓冲区内。

2. 硬件电路如附图 18 所示,要求在 8 个七段显示器上依次不断重复显示 19860524,在显示的过程中,要求发光二极管  $L$  一直亮。

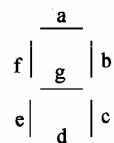


附图 17 硬件电路图



显示数字与7段代码对应表

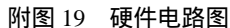
显示数字	0	1	2	3	4	5	6	7	8	9
7段代码 (H)	3F	06	5B	4F	66	6D	7D	07	7F	6F



附图 18 硬件电路图



196



## 参 考 答 案

1. 容易实现, 运算规则简单。

3. 可屏蔽中断请求是通过 INTR, 非屏蔽中断是通过 NMI; 可屏蔽中断是在 IF=1 开中断, 并且当前指令执行完了响应它; 而非屏蔽中断是在当前指令执行完了就响应它。

4. 伪指令语句符号名与定义符之间无冒号, 而指令语句标号与助记符之间有冒号; 伪指令语句不产生目标码, 是在汇编过程中起伪操作功能的, 而指令语句要产生目标

码，它和机器的一种操作相对应。

5. 串是指内存中有序的一组数据；DF 是控制变址寄存器的内容在串操作之后是递增还是递减；源串是以 DS 内容为段基址，目的串是以 ES 的内容为段基址；源串偏移地址是以 SI (ESI) 的内容为段基址，目的串是以 DI (EDI) 的内容为段基址；前缀是在串操作过程中再无别的操作，使在串操作过程中一直进行到 CX=0 为止。

二、填空题

- 1.  $[X+Y]_{补}=01100110B$  (溢出);  $[X-Z]_{补}=01110111B$  (溢出)
- 2.  $CX \neq 0$ ;  $ZF=1$ ;  $CX \neq 0$ ;  $ZF=0$
- 3. 16 位; 32 位; 段寄存器; 描述符
- 4. 每秒传递数据的位数; 传递数据的位数与传送脉冲数之比
- 5. 00AEH; 00AEH; 0F79H; 0F79H

三、选择题

- 1. A、C、D    2. D    3. B、F    4. A、C    5. A、D、E

四、汇编程序设计

```
1.      .MODEL SMALL
        .DATA
A1      DB  21H,32H,43H,54H,76H
A2      DB  43H,54H,65H,76H,87H
A3      DB  6 DUP(?)
        .CODE
        .STARTUP
        LEA BX,A1
        LEA SI,A2
        LEA DI,A3
        CLC
        MOV AH,0
        MOV CL,5
W:      MOV AL,[BX]
        ADC AL,[SI]
        DAA
        MOV[DI],AL
        INC BX
        INC SI
        INC DI
        DEC CL
        JNZ W
```

```

        ADC AH,0
        MOV[DI],AH
        .EXIT
    END

2 .      .MODEL SMALL
        .DATA
    BLOCK DW 20,30,40,50,60
        DW ?,?
        .CODE
        .STARTUP
    CLC
    MOV BX,OFFSET BLOCK
    MOV DX,0
    MOV AX,[BX]
    ADD AX,[BX+2]
    SUB AX,[BX+4]
    MUL WORD PTR[BX+6]
    DIV WORD PTR[BX+8]
    MOV[BX+10],AX          ;商
    MOV[BX+12],DX          ;余数
        .EXIT
    END
    
```

## 五、接口程序设计

1. 0832 地址为 88H; 8212 地址为 A8H~AFH 范围内任意一个均可。

8253 地址: 0<sup>#</sup>, 1<sup>#</sup>, 2<sup>#</sup>计数器及控制寄存器分别为 98H、9AH、9CH、9EH。

8253 计数值: N<sub>0</sub>=4000, N<sub>1</sub>=25, N<sub>2</sub>=5000

BLOCK DB?,.....

```

    MOV BX,OFFSET BLOCK
    MOV AL,00100101B
    OUT 9EH,AL
    MOV AL,40H
    OUT 98H,AL
    MOV AL,01010101B
    OUT 9EH,AL
    MOV AL,25H
    OUT 9AH,AL,
    MOV AL,10100011B
    OUT 9EH,AL
    MOV AL,50H
    OUT 9CH,AL
    
```

```

W:      IN AL,A8H
        TEST AL,01H
        JZ W
        MOV AL,[BX]
        OUT 88H,AL
        INC BX
        JMP W
        .EXIT
        END

```

2 . 8255A 的地址：A 口为 91H；B 口为 93H；控制口为 97H。

```

A1      DB 06H,6FH,7FH,7DH,3FH,6DH,5BH,66H
        LEA BX,A1
        MOV AL,10000000B
        OUT 97H,AL
        MOV AH,01H
W:      MOV AL,[BX]
        OUT 91H,AL
        MOV AL,AH
        OUT 93H,AL
        INC BX
        ROL AH,1
        JMP W

```

3 . 8255A 的地址：A 口为 84H；B 口为 85H；C 口为 86H；控制口为 87H。  
8259A 的地址为 98H 和 9AH。

```

        MOV AL,10110000B           ; 8255A 初始化
        OUT 87H,AL
        MOV AL,00001001B           ; PC4=1，中断允许
        OUT 87H,AL
        MOV AL,13H                  ; 8259A 初始化，ICW1
        OUT 98H,AL
        MOV AL,60H                  ; ICW2
        OUT 9AH,AL
        MOV AL,03H                  ; ICW4
        OUT 9AH,AL
        PUSH DS                     ; 建立中断向量表
        MOV AX,0
        MOV DS,AX
        MOV SI,OFFSET XY
        MOV[0188H],SI              ; 62H×4=0188H
        MOV SI,SEG XY

```

```

MOV[018AH],SI
POP DS
STI
LOO:  MOV AL,00001111B
      OUT 87H,AL                ; PG7=1, 启动 0809
      MOV AL,00001110B
      OUT 87H,AL
      HLT                      ; 等待中断
      CMP AL,0F6H
      JGE P
      CMP AL,78H
      JL Q
      JMP LOO
P:    MOV AL,00001100B
      OUT 87H,AL                ; PC6=0, 开关 Switch 低电平
      JMP LOO
Q:    MOV AL,00001101B
      OUT 87H,AL                ; PC6=1, 开关 Switch 高电平
      JMP LOO
XY:   IN AL,84H
      IRET

```

## 附录6 2004年某高校硕士研究生入学 “微机原理及应用”考试试题及答案

### 一、选择题(每小题3分,共15分)

1. 有一个八位二进制数的补码是11111101B,其相应的十进制真值是\_\_\_\_\_。  
A. -3            B. -2            C. 509           D. 253
2. 10001010B是\_\_\_\_\_。  
A. 带符号数                      B. 原码、反码、补码表示的带符号数  
C. 不带符号数                    D. BCD码  
E. 无法确定
3. 指令指针寄存器(EIP)中存放的是\_\_\_\_\_。  
A. 当前指令                      B. 下一条要执行的指令  
C. 操作数地址                    D. 下一条要执行指令的地址
4. 当 $\overline{M/\overline{IO}}=0$ ,  $\overline{W/\overline{R}}=0$ 时,CPU完成的操作是\_\_\_\_\_。  
A. 存储器读                      B. I/O读  
C. 存储器写                      D. I/O写
5. 假定(SS)=2000H,(SP)=0100H,(AX)=2107H,执行指令PUSH AX后,存放数据21H的物理地址是\_\_\_\_\_。

- A . 20102 H                      B . 20101H  
C . 200FEH                      D . 200FFH

二、填空题（每小题 3 分，共 15 分）

- 1 . 80486 直接可以访问的内存空间是\_\_\_\_\_B，它的一段最大空间是 B，一页空间是\_\_\_\_\_B。  
2 . 一般 I/O 端口的编址方式可以分为\_\_\_\_\_和\_\_\_\_\_两种方式。  
3 . 硬件中断可分为\_\_\_\_\_和\_\_\_\_\_两种。  
4 . 8255A 的 A 口工作在方式 1，那么 CPU 通过它与外设通信可以采用\_\_\_\_\_传送方式或\_\_\_\_\_传送方式或\_\_\_\_\_传送方式。  
5 . 模/数转换时，若输入模拟信号的最高有效频率为 20 kHz，采样频率最小为\_\_\_\_\_，应选用转换时间为\_\_\_\_\_的 A/D 转换器。

三、问答题（每小题 3 分，共 15 分）

- 1 . CPU 与外设交换信息的控制方式有哪几种？  
2 . 简述 8259A 中断控制器的主要功能是什么？  
3 . 在串行通信中奇/偶校验的概念是什么？  
4 . 什么是字节？什么是计算机的字长？  
5 . 80486 CPU 内部由哪几部分组成？

四、简化汇编源程序设计题（每小题 15 分，共 45 分）

- 1 . 下面程序的功能是：数组 A 包含 15 个互不相等的整数，数组 B 包含 20 个互不相等的整数，将既在数组 A 又在数组 B 中出现的整数存放在数组 C 中。请将程序填写完整。

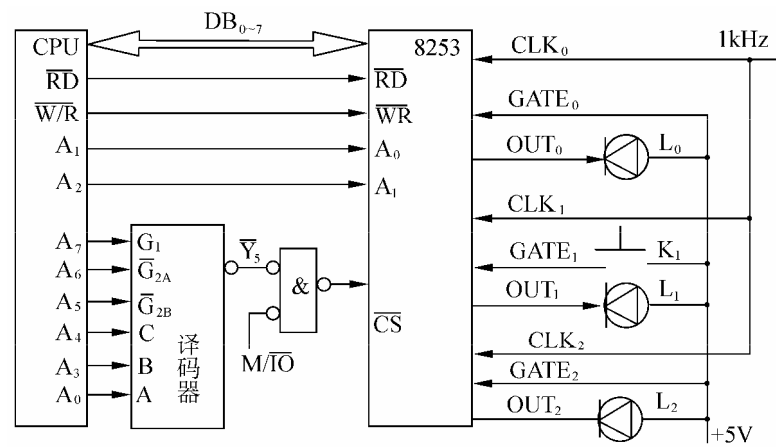
```
.MODEL SMALL
.DATA
A      DW 15 DUP(?)
B      DW 20 DUP(?)
C      DW 15 DUP(?)
.CODE
.STARTUP
MOV SI,0
MOV BX,0
MOV CX,15
NEXT1: PUSH CX
        MOV DI,0
        MOV CX,20
        MOV AX,(1)_____
NEXT2:  CMP B[DI],AX
```

```
JNE(2)_____  
MOV C[BX],AX  
INC BX  
INC BX  
ABCD: INC DI  
      INC DI  
      LOOP(3)_____  
      INC SI  
      INC SI  
(4)_____CX  
      LOOP NEXT1  
      .EXIT  
END
```

- 2. 试编一段程序将字符串 ‘a’ ~ ‘j’ 转换为大写字母 ‘A’ ~ ‘J’。
- 3. 计算  $(V - (X \times Y + Z - 540)) / X$ ，结果商、余数分别存放在 R1 和 R2 单元中。其中，X、Y、Z、V 均为 16 位带符号数，已分别存放在 X、Y、Z、V 单元中。

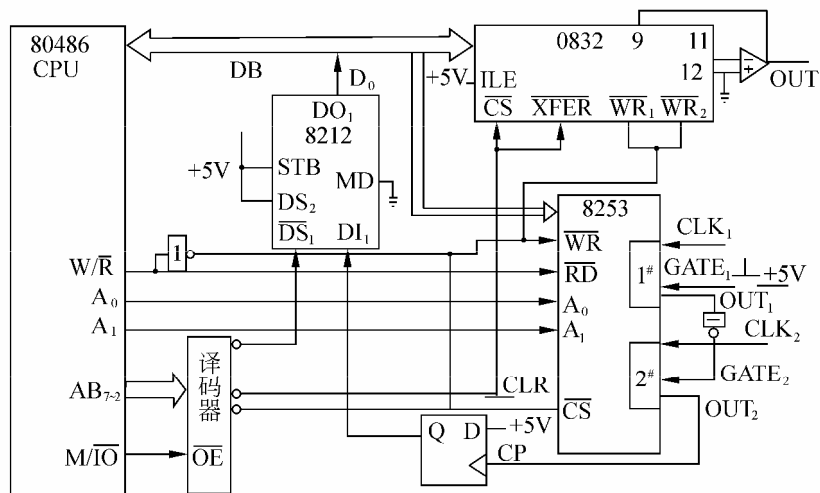
五、接口编程题(可省略伪指令)(每小题 20 分，共 60 分)

1. 接口原理图，如附图 20 所示。要求发光二极管 L<sub>0</sub> 亮 5s 后就熄灭；L<sub>1</sub> 在 K<sub>1</sub> 启动后亮 6s 后就熄灭；L<sub>2</sub> 亮 4s 灭 4s 交替进行。写出源程序片段。

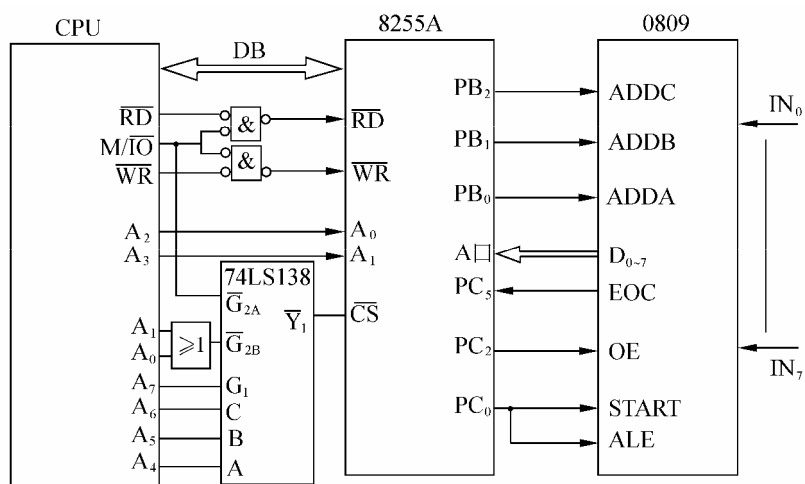


附 图 20

- 2. 有一接口电路图如附图 21 所示。8253 的 1<sup>#</sup>计数器定时 1s，CLK<sub>1</sub>=500Hz；CLK<sub>2</sub>=0.1 MHz。要求在 1s 之内由 CPU 向 0832 送 100 字节数。编写出简化的汇编语言源程序(可合理假设地址口)。
- 3. 利用 8255A 间接与 0809 接口，接口图如附图 22 所示。采用查询方式，编写转换通道 IN<sub>5</sub> 的模拟量的程序。



附 图 21



附 图 22

## 参 考 答 案

### 一、选择题

1 . A    2 . E    3 . D    4 . B    5 . D

### 二、填空题

- 1 . 4G ; 4G ; 4K
- 2 . I/O 端口单独编址 ; 与存储器统一编址
- 3 . 可屏蔽中断 ; 非屏蔽中断



4. 同步; 查询(异步); 中断

5. 40kHz; 25 $\mu$ s

### 三、问答题

1. 控制方式有程序控制传送方式、中断控制传送方式、DMA 方式和 I/O 处理机方式。

2. 8259A 的主要功能是: 接受中断请求信号, 实施对中断请求信号的屏蔽、判优, 比较目前提出请求而且没有被屏蔽的中断中最高优先级是否比正处理的中断具有更高的优先权, 向 CPU 发出中断请求, 接收 CPU 中断响应信号, 输送中断类型码, 接受 CPU 的编程。

3. 为了在串行通信长距离传送中避免发生错误, 由奇偶校验检查每个要传送的字符中“1”的个数, 根据设定的奇或偶校验, 自动在奇偶校验位上送“1”或“0”。在接收时, 检查字符的每一位和奇偶校验位的“1”的个数和是否满足奇或偶校验, 以确认是否发生错误。

4. 字节指的是八位二进制信息。字长是计算机字所含的二进制位数。计算机是作为一个整体被一次传送或运算最多的二进制位数。

5. 由总线接口部件、高速缓存、指令预取、译码电路、浮点部件、执行部件、分段管理部件、分页部件 8 大部分组成。

### 四、简化汇编源程序设计题

1. (1) A[SI]; (2) ABCD; (3) NEXT2; (4) POP

```
2.          .MODEL SMALL
           .DATA
BUF         DB 'abcdefghij'
           .CODE
           .STARTUP
MOV SI,0
MOV CS,10
MOV AL,20H
NEXT: SUB BUF[SI],AL
      INC SI
      LOOP NEXT
           .EXIT
      END
```

```
3. MODEL SMALL
   .DATA
X   DW 500H
Y   DW 3000H
Z   DW 4000H
V   DW 6000H
```

```

R1  DW ?
R2  DW ?
    . CODE
    . STARTUP
MOV  AX,X
IMUL Y                ; X×Y, 积在 DX:AX 中
MOV  CX,AX
MOV  BX,DX            ; 积转移至 BX: CX
MOV  AX,Z
CWD                  ; 对 Z 进行扩展, 存入 DX: AX
ADD  CX,AX
ADC  BX,DX            ; X×Y+Z
SUB  CX,540
SBB  BX,0             ; X×Y+Z-540, 结果在 BX: CX
MOV  AX,V
CWD                  ; 对 V 扩展, 存于 DX: AX
SUB  AX,CX
SBB  DX,BX           ; V-(X×Y+Z-540), 结果在 DX: AX
IDIV X                ; (V-(X×Y+Z-540))/X
MOV  R1,AX           ; 存商
MOV  R2,DX           ; 存余数
    . EXIT
END

```

## 五、接口编程题

1. 由译码电路分析可知, 8253 芯片计数器 0、计数器 1、计数器 2 及控制端口的地址分别为 91H、93H、95H、97H。根据接口图及题目要求, 8253 的计数器 0 应该工作在方式 0, 计数初值  $N_0=5000$ ; 计数器 1 应工作在方式 1,  $N_1=6000$ ; 计数器 2 应工作在方式 3,  $N_2=8000$ 。

根据以上分析, 可以编写出 8253 初始化程序。

8253 初始化程序片段如下所示。

```

; 初始化计数器 0
MOV  AL,21H
OUT  97H,AL
MOV  AL,50H
OUT  91H,AL
; 初始化计数器 1
MOV  AL,63H
OUT  97H,AL
MOV  AL,60H
OUT  93H,AL
; 初始化计数器 2

```

```

MOV AL,0A7H
OUT 97H,AL
MOV AL,80H
OUT 95H,AL

2 .. MODEL SMALL
. DATA
BLOCK DB 1,2,..... ; (100 字节数)
. CODE
. STARTUP
MOV CX,100
MOV AL,01100011B ; 设 8253 地址为 70H~73H , 计数器 1 初始化
OUT 73H,AL
MOV AL,05H
OUT 71H,AL ; 计数器 1 初值=500
MOV AL,10100101B
OUT 73H,AL
MOV AL,10H
OUT 72H,AL ; 计数器 2 初值=1000
MOV BX,OFFSET BLOCK
P: IN AL,80H ; 设 8212 地址为 80H
TEST AL,01H
JZ P
MOV AL,[BX]
OUT 20H,AL ; 设 0832 地址为 20H
INC BX
DEC CX
JNZ P
. EXIT
END

```

3 .由接口图分析可知 8255A 的 A 口、B 口、C 口和控制口的地址分别为 90H、94H、98H、9CH , 程序片段如下所示。

```

; 8255 初始化
MOV AL,10011000B
OUT 9CH,AL

; A/D 转换
MOV AL,05H
OUT 94H,AL ; 选 IN5 通道
MOV AL,01H ; 启动 A/D
OUT 9CH,AL
MOV AL,00H
OUT 9CH,AL
NOP ; 延时 , 避免虚假信号

```

```

NOP
W:  IN AL,98H                ;测 EOC
    TEST AL,00100000B
    JZ W
    MOV AL,00000101B        ;OE 置 1
    OUT 9CH,AL
    IN AL,90H
    MOV BL,AL                ;转换结果存入 BL 中
    MOV AL,00000000B        ;OE 清 0
    OUT 9CH,AL

```

## 附录 7 2005 年某高校硕士研究生入学 “微机原理及应用”考试试题及答案

### 一、填空题（每小题 2 分，共 30 分）

1. 已知 $[X]_{\text{原}}=01001001\text{B}$ ， $[Y]_{\text{原}}=10101010\text{B}$ ，则 $[X+Y]_{\text{补}}=$ \_\_\_\_\_， $[X-Y]_{\text{补}}=$ \_\_\_\_\_（并判断是否溢出）。
2. 80486 CPU 中的执行部件完成\_\_\_\_\_等任务。
3. 当 CPU 进行数据输出时，信号  $\overline{W}/\overline{R}$  \_\_\_\_\_；当 CPU 进行数据输入时，信号  $\overline{W}/\overline{R}$  为\_\_\_\_\_。
4. 80486 在保护方式下，虚拟空间为\_\_\_\_\_，而实际地址空间为\_\_\_\_\_。
5. 对于指令  $\text{XCHG BX}, [\text{BP}+\text{SI}]$ ，如果指令执行前， $(\text{BX})=6\text{F}30\text{H}$ ， $(\text{BP})=0200\text{H}$ ， $(\text{SI})=0046\text{H}$ ， $(\text{DS})=2456\text{H}$ ， $(\text{SS})=2\text{F}00\text{H}$ ， $(2\text{F}246\text{H})=4154\text{H}$ ，则执行指令后： $(\text{BX})=$ \_\_\_\_\_， $(2\text{F}246\text{H})=$ \_\_\_\_\_。
6. 串操作指令规定源串在\_\_\_\_\_段中，用\_\_\_\_\_来寻址源操作数；目的串在\_\_\_\_\_段中，用\_\_\_\_\_来寻址目的操作数。
7. 近过程（NEAR）的 RET 指令把当前栈顶的一个字弹出到\_\_\_\_\_；远过程（FAR）的 RET 指令弹出一个字到\_\_\_\_\_后又弹出一个字到\_\_\_\_\_。
8. 下列程序段是比较 AX、BX、CX 中带符号数的大小，将最大的数放在 AX 中。请将程序填充完整。

```

CMP AX, BX
(1) _____NEXT
XCHG AX, BX
NEXT:(2) _____CX, AX
JLE OUT1
(3) _____AX, CX
OUT1: _____

```

9. 将 AX 寄存器的内容清“0”的指令有(写出4条指令)

\_\_\_\_\_。

10. 在汇编语言中标号的类型有\_\_\_\_\_, 变量的类型有\_\_\_\_\_。

11. 80486 直接可以访问的内存空间是\_\_\_\_\_B, 它的一段最大空间是\_\_\_\_\_B, 一页空间是\_\_\_\_\_B。

12. 一般 I/O 端口的编址方式可以分为\_\_\_\_\_和\_\_\_\_\_两种方式。

13. 硬件中断可分为\_\_\_\_\_和\_\_\_\_\_两种。

14. 在串行通信中波特率是指\_\_\_\_\_。

15. 8255A 的 A 口工作方式有\_\_\_\_\_, B 口的工作方式有\_\_\_\_\_。

二、判断下列程序所实现的功能(10分)

```

        .MODEL SMALL
        .DATA
BLOCK   DB 60,90,0,50,.....
MAX     DB ?
MIN     DB ?
        .CODE
        .STARTUP
MOV     CX,9
MOV     AL,BLOCK
MOV     DL,BLOCK
MOV     BX,OFFSET BLOCK
AGAIN:  INC BX
        CMP AL,[BX]
        JAE NEXT1
        MOV AL,[BX]
NEXT1:  CMP DL,[BX]
        JBE NEXT2
        MOV DL,[BX]
NEXT2:  LOOP AGAIN
        MOV MAX,AL
        MOV MIN,DL
        EXIT
        END
    
```

三、汇编编程题(50分)

1.(15分)在内存 XYZ 开始存放了 10 个带符号字节数,判断它们中有多少个正数,多少个负数,正数的个数放在 DH 中,负数的个数放在 DL 中。写出汇编源程序。

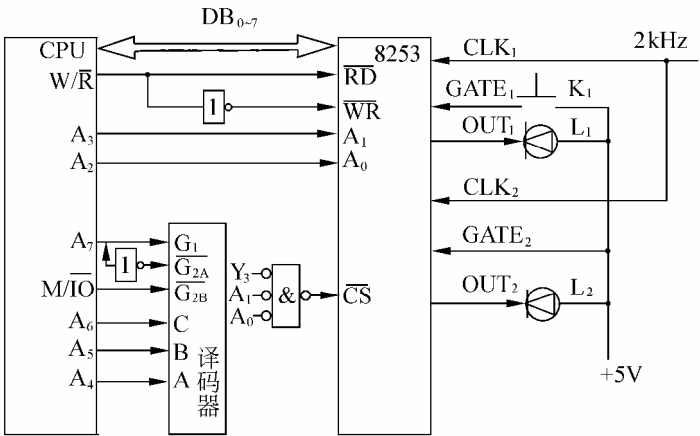
2.(15分)在内存 BLOCK 开始的连续 3 个单元中有 3 个带符号数(字节) a、b、c,写出计算  $a-b \times c$  的汇编源程序。

3.(20分)在内存同一段内有两串不带符号字节数各 10 个,用串操作指令从内存

取数和存入，计算对应项之和，并将和存入目的串；若出现溢出则停止计算。写出汇编源程序。

四、(20分)

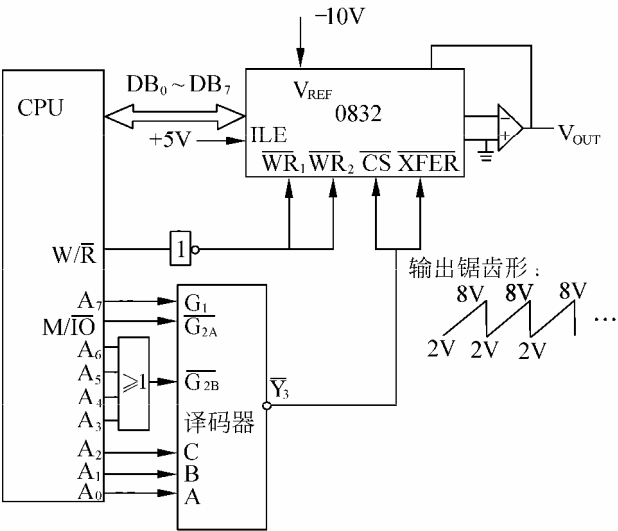
有接口原理图如附图 23 所示。要求发光二极管  $L_1$  在  $K_1$  启动后亮 3s 就熄灭；发光二极管  $L_2$  亮 2 秒灭 2s 交替进行。编写出汇编源程序（可省略伪指令）。



附图 23

五、(20分)

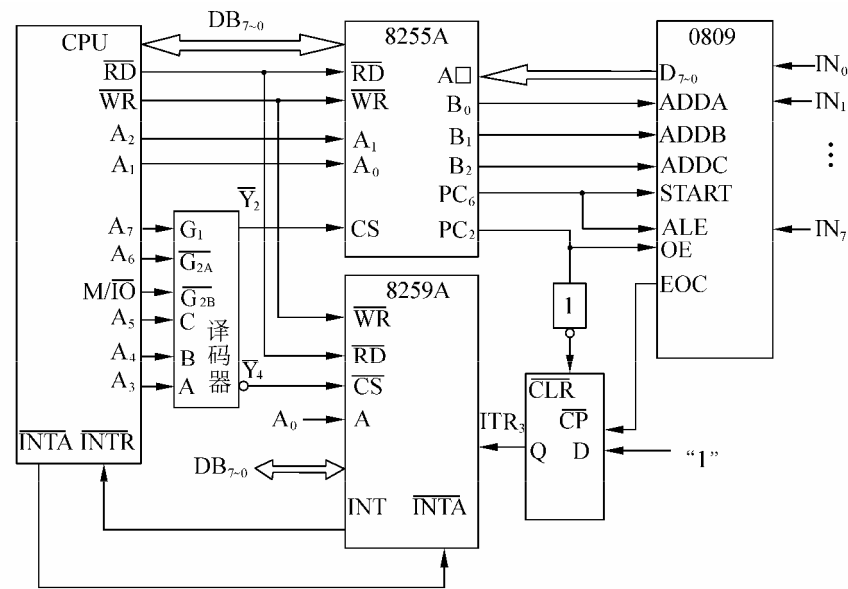
D/A 转换器与 CPU 接口图如附图 24 所示，要求输出 2~8V 的连续锯齿波（量程为 0~10V），写出汇编源程序（可省略伪指令）。



附图 24

六、(20分)

接口图如附图 25 所示,要求从 0809A/D 转换器 8 个通道各采样 100 个点。8259A 采用边沿触发、非缓冲方式、自动结束方式,这 8 个中断类型码假设为 60H~67H。写出汇编源程序(可省略伪指令)。



附图 25 接口图

参 考 答 案

一、填空题

- 1. 00011111B (未溢出); 01110011B (未溢出)
- 5. 4154H; 6F30H
- 6. 数据; (E) SI; 附加; (E) DI      7. IP; IP; CS
- 8. JGE; CMP; XCHG
- 9. MOV AX, 0; XOR AX, AX; SUB AX, AX; AND AX, 0
- 10. NEAR、FAR; BYTE、WORD、DWORD、PWORD、QWORD
- 11. 4G; 4G; 4K
- 12. I/O 端口单位编址; 存储器统一编址
- 13. 可屏蔽中断; 非屏蔽中断
- 14. 一秒钟传送的二进制位数
- 15. 方式 0、方式 1、方式 2; 方式 0、方式 1

## 二、判断下列程序所实现的功能

在 10 个无符号字节数中寻找最大数和最小数，并将最大数存入 MAX、最小数存入 MIN 单元中。

## 三、汇编编程题

```
1 .      .MODEL  SMALL
          .DATA
XYZ      DB  -1,-2,-6,...
          .CODE
          .STARTUP
MOV  CX,10
MOV  BX,OFFSET XYZ
MOV  DX,0
AGAIN:  CMP  BYTE  PTR[BX],0
        JGE  NEXT
        INC  DL
        JMP  X
NEXT:   INC  DH
X:      INC  BX
        LOOP AGAIN
        .EXIT
END
```

```
2 .      .MODEL  TINY
          .DATA
BLOCK DB 3 DUP(?)
SULT  DW ?
          .CODE
          .STARTUP
LEA  SI,BLOCK
MOV  AL,[SI+1]
MOV  BL,[SI+2]
IMUL BL
MOV  DX,AX
MOV  AX,[SI]
CBW
SUB  AX,DX
MOV  SULT,AX
        .EXIT
END
```

```
3 .      .MODEL  SMALL
```



```

        . DATA
BLOCK1  DB  10 DUP(?)
BLOCK2  DB  10 DUP(?)
        . CODE
        . STARTUP
MOV  AX,@DATA
MOV  ES,AX
LEA  SI,BLOCK1
LEA  DI,BLOCK2
MOV  CX,10
CLD
W:   LODSB
      ADD AL,[DI]
      JC  P1
      STOSB
      LOOP W
P1:   HLT
      . EXIT
      END
    
```

#### 四、

8253 的端口地址: 0B0H 0<sup>#</sup> 0B4H 1<sup>#</sup>  
                                   0B8H 2<sup>#</sup> 0BCH 控制

计数值:  $n_1=6000$

$n_2=8000$

```

        . MODEL TINY
        . CODE
        . STARTUP
MOV  AL,63H
OUT  0BCH,AL
MOV  AL,0A7H
OUT  0BCH,AL
MOV  AL,60H
OUT  0B4H,AL
MOV  AL,80H
OUT  0B8H,AL
      . EXIT
      END
    
```

#### 五、

D/A 地址: 83H

$$\frac{10V}{256} = 0.039 \text{ V/步}$$

$$\frac{2V}{0.039V} = 51.3 \quad \text{取整 } 51$$

$$\frac{8V}{0.039V} = 205.1 \quad \text{取整 } 205$$

```

.MODEL TINY
.CODE
.STARTUP
Q: MOV AL, 51
W: OUT 83H, AL
    INC AL
    CMP AL, 205
    JBE W
    JMP Q
.EXIT
END

```

六、

8255A 地址：A 口 90H    B 口 92H    C 口 94H  
                  控制口 96H  
8259A 地址：偶地址：0A0H  
                  奇地址：0A1H  
8255A 命令字：90H  
8259A 命令字：ICW1 13H  
                  ICW2 取 60H  
                  ICW4 03H

```

.MODEL TINY
.DATA
WP DB 800 DUP(?)
.CODE
.STARTUP
LEA BX, WP
MOV CX, 100
MOV AL, 90H           ; 8255A 初始化
OUT 96H, AL
MOV AL, 13H           ; 8259A 初始化
OUT 0A0H, AL
MOV AL, 60H
OUT 0A1H, AL
MOV AL, 03H
OUT 0A1H, AL
PUSH DS               ; 建立矢量表

```

```

MOV    AX,0
MOV    DS,AX
MOV    AX,OFFSET XY
MOV    [63H*4],AX
MOV    AX,SEG XY
MOV    [63H *4+2],AX
POP    DS
Q: MOV  DL,O
W: MOV  AL,DL
OUT    92H,AL
MOV    AL,0DH          ; PC6=1, 启动 0809
OUT    96H,AL
DEC    AL
OUT    96H,AL
HLT
INC    DL
CMP    DL,07H
JBE    W
LOOP   Q
HLT
XY: MOV  AL,,05H          ; PC2=1
OUT    96H,AL
IN     AL,90H            ; 读数
MOV    [BX],AL
INC    BX
MOV    AL,04H          ; PC2=0
OUT    96H,AL
IRET
.EXIT
END

```

## 附录8 2006年某高校硕士研究生入学 “微机原理及应用”考试试题及答案

### 一、问答题(每小题3分,共15分)

1. 什么是字节?什么是计算机的字长?
2. 80486 CPU 内部由哪几部分组成?
3. 什么叫串?串操作有哪些基本的指令?串前缀在什么情况下使用?
4. 计算机的主存与辅存(即内存与外存)是以什么原则划分的?80486 CPU 直接访问的内存空间(不考虑虚拟空间)最大是多少?
5. 中断向量和中断描述符表有何异同?

## 二、填空题（每空 1 分，共 15 分）

1. 在汇编语言中标号的类型有\_\_\_\_\_，变量的类型有\_\_\_\_\_。
2. 在汇编语言中，一个过程的属性为 NEAR，表明主程序和子程序\_\_\_\_\_；若为 FAR，则表明主程序和子程序\_\_\_\_\_。
3. 指出下列指令源操作数的寻址方式：  
MOV AX, BLOCK [SI] \_\_\_\_\_  
MOV AX, [SI] \_\_\_\_\_  
MOV AX, [6000H] \_\_\_\_\_  
MOV AX, [BX+SI] \_\_\_\_\_  
MOV AX, 1500H \_\_\_\_\_
4. 阅读下面的程序。在\_\_\_\_\_的情况下，本段程序的执行结果是 (AH)=0；在\_\_\_\_\_的情况下，其结果为 (AH)=0FFH。

```
IN AL, 20H
TEST AL, 80H
JZ A
MOV AH, 0
JMP B
A: MOV AH, 0FFH
B: HLT
```

5. 对于串操作指令，源串是以\_\_\_\_\_内容为段基址，以\_\_\_\_\_内容为偏移地址；目的串是以\_\_\_\_\_内容为段基址，以\_\_\_\_\_内容为偏移地址。

## 三、汇编语言编程题（简化段定义或完整段定义均可，每小题 20 分，共 40 分）

1. 在内存数据段中的 A1 开始缓冲区内有两个不带符号的字节数 X 和 Y，计算  $X \times Y$  并将其积存入 B1 开始的缓冲区内。
2. 在内存数据段中的 C1 开始缓冲区内有两个带符号的字节数 W 和 P，计算  $W/P$ ；并将其商和余数分别存入 SHU 单元和 YU 单元内。

## 四、汇编程序填空题（每个空格 3 分，共 30 分）

1. 下面程序的功能是实现把 BX 寄存器内的二进制数用十六进制数的形式在屏幕上显示出来。请将该程序填充完整。

```
· MODEL TINY
· CODE
· STARTUP
MOV CH, 4
AGAIN: MOV (1) _____, 4
ROL BX, CL
```

```

        MOV AL, BL
        AND AL, ( 2 ) _____
        OR AL, 30H
        CMP AL, 3AH
        ( 3 ) _____ NEXT
        ADD AL, 07H
NEXT:   MOV DL, AL
        MOV AH, ( 4 ) _____
        INT 21H
        DEC CH
        ( 5 ) _____ AGAIN
        .EXIT
        END

```

2. 将内存首地址为 BLOCK 开始的 10 个带符号字节数, 将其按由大到小的次序排列。

```

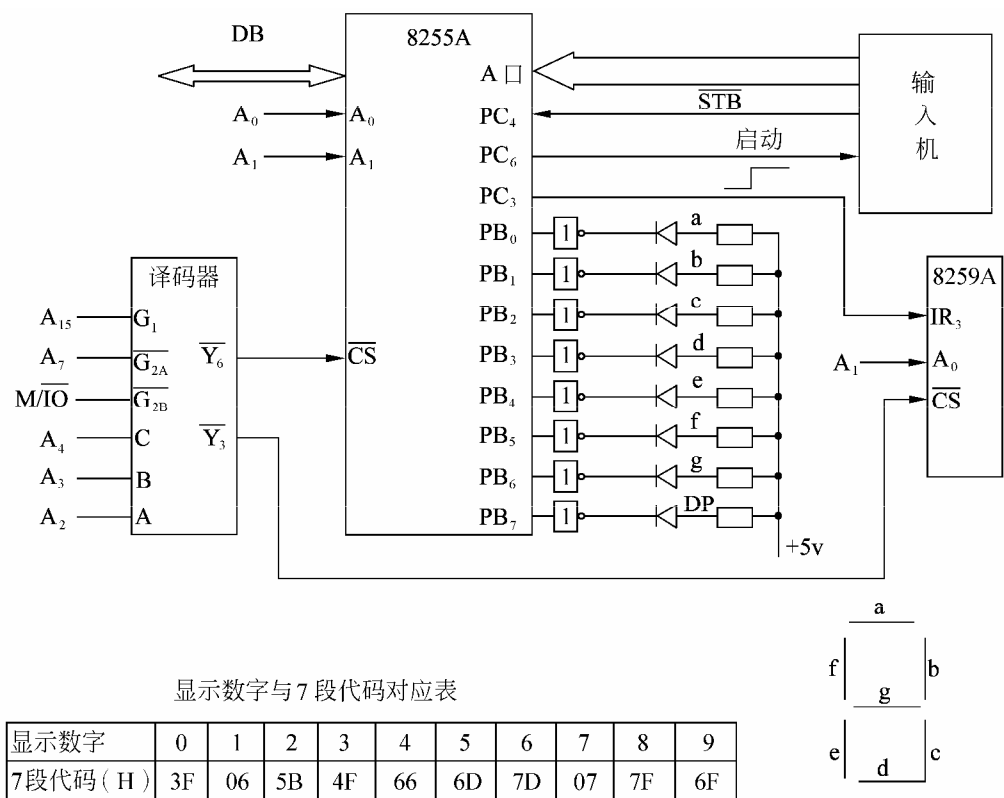
        .MODEL SMALL
        .DATA
BLOCK DB 66H, 88H, 10H, 20H, 30H, 40H, 50H, 60H, 70H, 80H
        .CODE
        .STARTUP
        MOV DI, 9
LOOP1: MOV CX, DI
        MOV BX, ( 1 ) _____
LOOP2: MOV AL, BLOCK [BX]
        CMP AL, BLOCK [ ( 2 ) _____ ]
        ( 3 ) _____ NEXT
        XCHG AL, BLOCK [ BX+1 ]
        MOV BLOCK [BX], AL
NEXT: ( 4 ) _____ BX
        LOOP LOOP2
        DEC ( 5 ) _____
        JNZ LOOP1
        .EXIT
        END

```

### 五、接口程序填空题(每个空格 2 分, 共 30 分)

1. 接口如附图 26, 8255A 的 B 口接七段显示器 a、b、c、d、e、f、g 七个发光二极管, 要求不断重复显示 0~9 十个数字, 在显示的过程中 DP 一直不亮。每个数字显示的时间是 DEC 和 JNZ 指令重复执行 300 次的时间。在重复显示 0~9 数字的同时, 从输入机读入 50 个数据。50 个数据输入完, 显示也将停止。设 8259A 八个中断类型码为 60H~67H, 边沿触发、非缓冲方式、自动结束方式。下列程序片段实现以上要求, 请补

充完整。



附 图 26

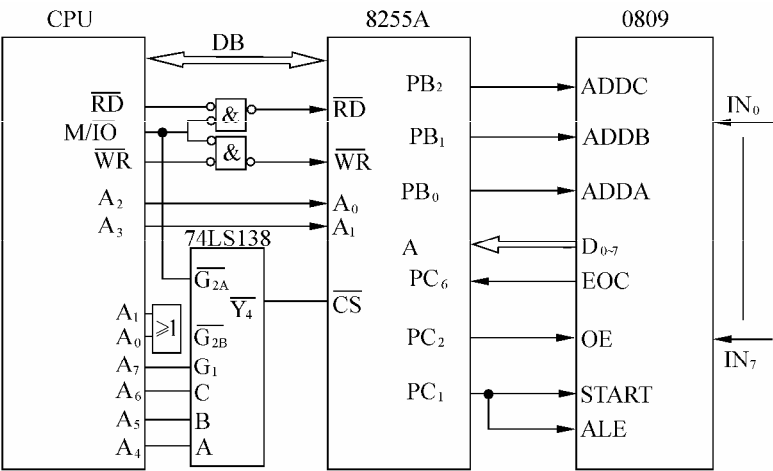
；程序片段如下

```
TAB DB 3FH,06H,5BH,4FH,66H,6DH,7DH,07H,7FH,6FH
      DB 50 DUP ( ? )
      MOV AL,(1) _____
      MOV DX,801BH
      OUT DX,AL
      MOV AL,(2) _____
      OUT DX,AL
      PUSH DS
      MOV AX,0
      MOV DS,AX
      MOV BX,OFFSET XY
      MOV DI,SEG XY
      MOV [ (3) _____ ],BX
      MOV [ (4) _____ ],DI
      POP DS
      MOV DX,800CH
      MOV AL,00010011B
```

```

        OUT  DX,AL
        MOV  DX,800EH
        MOV  AL,( 5 ) _____
        OUT  DX,AL
        MOV  AL,03H
        OUT  DX,AL
        MOV  CL,50
        MOV  BX,( 6 ) _____
        MOV  SI,BX+10
        MOV  DI,SI
        MOV  AL,00001101B
        MOV  DX,801BH
        OUT  DX,AL
        DEC  DX
        DEC  DX
Q:      MOV  AL,[BX]
        OUT  DX,AL
        MOV  AX,300
P:      DEC  AX
        JNZ  P
        INC  BX
        CMP  BX,SI
        JB   Q
        SUB  BX,10
        IMP  Q
; 中断服务程序
XY:     PUSH  AX
        PUSH  CX
        PUSH  BX
        MOV  DX,8018H
        IN   AL,DX
        MOV  [DI],AL
        ( 7 ) _____
        DEC  CL
        JNZ  RP
        JMP  PQ
RP:     POP  BX
        POP  CX
        POP  AX
        ( 8 ) _____
PQ:     MOV  AL,00001100B
        MOV  DX,801BH
        OUT  DX,AL
        HLT
    
```

2. 利用 8255A 间接与 0809 接口，接口图如附图 27 所示。采用查询方式，编写转换通道 IN6 的一个点的模拟量程序。



附图 27 接口图

```

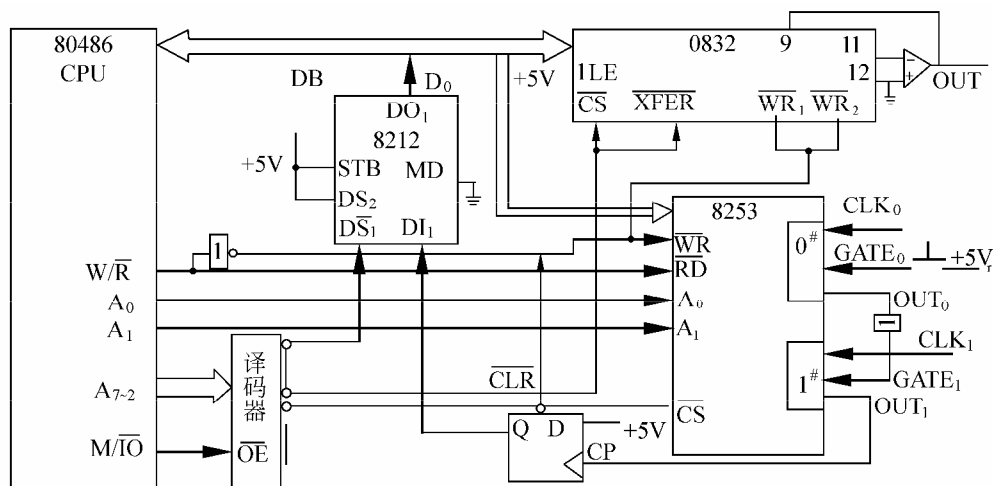
; 程序片段如下
MOV AL, ( 1 ) _____
OUT 0CCH,AL
MOV AL, ( 2 ) _____
OUT 0C4H,AL
MOV AL,03H
OUT 0CCH,AL
MOV AL,02H
OUT 0CCH,AL
NOP
W: IN AL, ( 3 ) _____
TEST AL, ( 4 ) _____
JZ W
MOV AL, ( 5 ) _____
OUT 0CCH,AL
IN AL, ( 6 ) _____
MOV BL,AL
MOV AL, ( 7 ) _____
OUT 0CCH,AL

```

六、接口编程题（20 分）

有一接口电路如附图 28 所示，8253 的 0# 计数器定时 1s，CLK0=300HZ；CLK1=0.2MHz。要求在 1s 之内由 CPU 向 0832 送 40 个字节数。编写出简化的汇编语言源程序（可合理假设口地址）。





附图 28 接口电路图

## 参 考 答 案

### 一、问答题

1. 字节指的是八位二进制信息。字长是计算机字所含的二进制位数。计算机字是作为一个整体被一次传送或运算最多的二进制位数。
2. 由总线接口部件、高速缓存、指令预取、译码电路、浮点部件、执行部件、分段管理部件、分页部件 8 大部分组成。
3. 串是一组有序的连续存放的数据。串基本操作指令有串传送 MOVSB/MOVSX/MOVSQ、串比较 CMPSB/CMPSX/CMPSQ、串扫描 SCASB/SCASX/SCASQ、串存储(写串) STOSB/STOSX/STOSQ、串装入(读串) LODSB/LODSX/LODSQ 共 5 种。串前缀是在须要连续进行串操作的情况下使用。
4. 外存和内存是以 CPU 能否直接访问这一原则来划分的。80486 CPU 能直接访问的最大内存空间为 4GB。
5. 中断向量和中断描述符表都是中断类型码与该中断类型相对应的中断服务程序的入口地址之间的连接表。中断向量表对应 16 位模式下,表内内容为中断服务程序入口地址的 IP 和 CS 值,每个中断类型码对应 4 个单元;中断描述符表对应 32 位模式下,表内内容为中断门描述符,每个中断描述符对应 8 个单元。

### 二、填空题

1. NEAR 和 FAR; BYTE、WORD、DWORD、PWORD、QWORD
2. 在同一代码段中;不在同一代码段中
3. 寄存器相对寻址; 寄存器间接寻址; 直接寻址; 基址变址寻址; 立

即寻址

- 4. AL 中的最高位为 1 ; AL 中的最高位为 0
- 5. DS ; (E) SI ; ES ; (E) DI

### 三、汇编语言编程题

```
1. .MODEL SMALL
   .DATA
   A1 DB 65H, 30H
   B1 DW ?
   .CODE
   .STARTUP
   MOV AL, A1
   MOV BL, A1+1
   MUL BL
   MOV B1, AX
   .EXIT
   END

2. .MODEL SMALL
   .DATA
   C1 DB 80H, 54H
   SHU DB ?
   YU DB ?
   .CODE
   .STARTUP
   MOV AL, C1
   CBW
   MOV BL, C1+1
   IDIV BL
   MOV SHU, AL
   MOV YU, AH
   .EXIT
   END
```

### 四、汇编程序填空题

- 1. CL ; 0FH ; JB ; 2 ; JNZ
- 2. 0 ; BX+1 ; JGE ; INC ; DI

### 五、接口程序填空题

- 1. 0B0H ; 09H ; 018CH ; 018EH ; 60H ; OFFSET TAB ; INC DI ; IRET
- 2. 10011000B ; 06H ; 0C8H ; 01000000B ; 00000101B ; 0C0H ; 00000100B

## 六、接口编程题

假设 8253 的端口地址为 70H~73H, 8212 的端口地址为 80H, 0832 的端口地址为 20H。

```

· MODEL SMALL
· DATA
BLOCK DB 1,2,..... ; (40 字节数)
· CODE
· STARTUP
MOV CX,40
MOV AL,00100011B
OUT 73H,AL
MOV AL,03H
OUT 70H,AL
MOV AL,01100101B
OUT 73H,AL
MOV AL,50H
OUT 71H,AL
MOV BX,OFFSET BLOCK
P: IN AL,80H
TEST AL,01H
JZ P
MOV AL,[BX]
OUT 20H,AL
INC BX
DEC CX
JNZ P
· EXIT
END
    
```

## 参 考 文 献

- 1 史新福等. 32 位微型计算机原理接口技术及其应用 (第 2 版). 西安: 西北工业大学出版社, 2001
- 2 李广军等. 微型计算机原理. 成都: 电子科技大学出版社, 2001
- 3 乔瑞萍, 欧文. 微型计算机原理典型题解析及自测试题. 西安: 西北工业大学出版社, 2001
- 4 武自芳. 微型计算机原理常见题型解析及模拟题 (第 2 版). 西安: 西北工业大学出版社, 2001
- 5 朱庆保等. 微型计算机系统及接口应用技术. 南京: 南京大学出版社, 1997
- 6 张雷兰. 汇编语言程序设计. 北京: 北京理工大学出版社, 2001
- 7 艾德才等. 80486/80386 系统原理与接口大全. 北京: 清华大学出版社, 1995
- 8 李大友等. 微型计算机接口技术. 北京: 清华大学出版社, 1998