第一章　数和字符的表示法

复　习　提　要

1. 计算机只能识别以二进制形式存在的机器语言。计算机内部数的存储及运算也都是采用二进制。

2. 一个二进制数的值由 1 所在位置的权来确定。如$(1101)_2=2^3+2^2+2^0=(13)_{10}$

3. 二进制的负数用补码来表示。对一个二进制数求补(按位求反,末位再加 1),即得到这个数相反数的补码表示。

4. 补码的加法和减法的规则是:

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

5. 16 进制是一种很重要的短格式计数法,它把二进制数每 4 位分成一组,分别用 0～9 和 A～F 来表示 0000～1111。反之,16 进制数的每一位用四位二进制表示,就是相应的二进制数。

6. 十进制转换为二进制数的方法主要有降幂法和除法。计算机十化二程序中采用下面的算法:

$$N_mN_{m-1}\cdots N_1N_0=(\cdots((N_m+0)\times 10+N_{m-1})\times 10+\cdots+N_1)\times 10+N_0$$

7. 标志位 OF=1 表示带符号数的运算结果无效。CF=1 表示无符号数的运算结果无效。

8. 计算机中的字符数据用 ASCII 码表示,一个字符在存储器中占用一个字节(8 位二进制码)。

9. BCD 码是一种用二进制编码的十进制数,又称二-十进制数或 8421 码,它用 4 位二进制数表示一个十进制数码。BCD 码有压缩和非压缩两种格式。压缩的 BCD 码用 4 位二进制数表示一个十进制数位,如 95_{10}表示为 1001,0101。非压缩的 BCD 码用 8 位二进制数表示一个十进制数位。如 95_{10}表示为 00001001 00000101。

例　题　分　析

例 1.1　完成下列各式补码数的运算,并根据结果设置标志位 SF、ZF、CF 和 OF,指出运算结果有效否。

(1) 0100,1001b+1001,1101b

(2) 0100,0001b−1010,1011b

(3) 0A95Bh+8CA2h

(4) 6531h−42DAh

解:　(1)　0100,1001　　SF ZF CF OF

+1001,1101　　1　0　0　0　　运算结果有效

1110,0110

这两个二进制补码数相加，和为一个非零的负数(SF＝1，ZF＝0)；最高有效位无进位(CF＝0)；正负两数相加，结果不会溢出(OF＝0)。

(2)　0100,0001　　SF ZF CF OF

＋0101,0101　　1　0　1　1　　结果无效

1001,0110

两个补码数相减，先对减数求补，(1010,1011 求补后得 0101,0101)，再把减法转化为加法进行运算。运算结果为非零负数(SF＝1，ZF＝0)；最高位无进位(CF＝1)；结果符号与减数相同(均为负)，则 OF＝1，结果是错误的。

(3)　A95B　　SF ZF CF OF

＋8CA2　　0　0　1　1　　结果无效

1 35FD

这两个十进制数相加，所得结果是一个非零的正数(SF＝0，ZF＝0)；最高位有进位(CF＝1)；两负数相加，结果为正数，故 OF＝1，结果无效。

(4)　6531　　SF ZF CF OF

＋BD26　　0　0　0　0　　结果有效

1 2257

先对减数求补(42DA 求补后得 0BD26)，然后用加法进行运算。运算结果为非零正数(SF＝0，ZF＝0)；最高位有进位(CF＝0)；结果符号与减数不同(OF＝0)，结果正确。

例 1.2　把字符串“PART1:Memory”存放在 1100 开始的存储区中，请写出字符串的存储情况。

字　符：	P	A	R	T	1	:	M	e	m	o	r	y
ASCII 码：	50	41	52	54	31	3A	4D	65	6D	6F	72	79
地　址：	1100	1101	1102	1103	1104	1105	1106	1107	1108	1109	110A	110B

IBM PC 的存储器按字节编址，一个 ASCII 码字符占用一个字节。

例 1.3　写出十进制数 3590 的非压缩 BCD 码和压缩的 BCD 码，并分别把它们存入数据区 UNPAK 和 PAKED。

UNPAK＋0	0 0
＋1	0 9
＋2	0 5
＋3	0 3

PAKED＋0	9 0
＋1	3 5

3590 的非压缩 BCD 码为 0000 0011，0000 0101，0000 1001，0000 0000；压缩的 BCD 码为 0011 0101，1001，0000。它们以相反的顺序存储在数据区中。

习　题

1.1　用降幂法和除法将下列十进制数转换为二进制数和 16 进制数。

(1) 369　　(2) 10000　　(3) 4095　　(4) 32767

1.2 将下列二进制数转换为 16 进制数和十进制数。

(1) 101101 (2) 10000000 (3) 1111111111111111 (4) 11111111

1.3 将下列 16 进制数转换为二进制数和十进制数。

(1) FA (2) 5B (3) FFFE (4) 12D4

1.4 完成下列二进制数的加法：

(1) 00010101 (2) 00111110 (3) 11111111 (4) 00111001

+00001101 +00101001 +00000001 +01000111

1.5 完成下列 16 进制数的加法：

(1) 23A6 (2) 51FD (3) 7779 (4) EABE

+0076 +0036 +0887 +26C4

1.6 完成下列八位二进制数的减法：

(1) 00011111 (2) 00011001 (3) 10101010 (4) 00111000

−00000101 −00010000 −00010101 −11111111

1.7 完成下列十六进制数的减法：

(1) FFFF (2) 12AA

−AAAA −02AB

1.8 写出下列二进制数的补码表示：

(1) −00010011 (2) −00111100 (3) −00111001 (4) −01000000

1.9 写出下列补码数的相反数：

(1) 11001000 (2) 10111101 (3) 10000000 (4) 00000000

1.10 下列各数均为十进制数，请用 8 位二进制补码计算下列各题，并用 16 进制数表示其运算结果，同时说明进位值，并判断是否溢出。

(1) (−85)+76 (2) 85+(−76) (3) 85−76 (4) 85−(−76)

(5) (−85)−76 (6) (−85)−(−76)

1.11 把下列数和字符用 16 进制表示出来：

(1) 字母 Q (2) ASCII 码 7 (3) 二进制数 10111101

(4) 十进制数 100 (5) 空格(space) (6) ? 符

1.12 下列各数均为用 16 进制表示的 8 位二进制数，请说明当它们分别看作补码表示的数及字符的 ASCII 码时，它们所表示的十进制数及字符是什么？

(1) 4F (2) 2B (3) 73 (4) 59

1.13 请写出下列字符串的 ASCII 码。

For example,

This is a number 3692.

1.14 分别用 8 位二进制和二位 16 进制数写出下列十进制数的补码表示：

(1) 16 (2) 100 (3) 127 (4) 0

(5) −16 (6) −100 (7) −128 (8) −1

1.15 16 位的二进制补码数所能表示的十进制最大数和最小数分别是什么？

1.16　16 位二进制数所能表示的无符号数的范围是多大?

1.17　假设两个二进制数 A=01101010,B=10001100,试比较它们的大小。

(1) A、B 两数均为带符号的补码数。

(2) A、B 两数均为无符号数。

1.18　有一个 16 位的数值 0101,0000,0100,0011,

(1) 如果它是一个二进制数,和它等值的十进制数是多少?

(2) 如果它们是 ASCII 码字符,则是些什么字符?

(3) 如果是压缩的 BCD 码,它表示的数是什么?

1.19　求出以下各 16 进制数与 62A0H 的和,并根据结果设置标志位 SF、ZF、CF 和 OF:

(1) 1234　　(2) 4321　　(3) CFA0　　(4) 9D60

1.20　求出以下各 16 进制数减去 4AE0H 的差值,并根据结果设置标志位 SF、ZF、CF 和 OF:

(1) 1234　　(2) 5D90　　(3) 9090　　(4) EA04

1.21　从键盘输入一个十进制数 4096 存入缓冲区 BUFFER,再将该数所对应的非压缩 BCD 码和压缩的 BCD 码分别存入 UNPAK 和 PAKED 数据项,请写出 BUFFER、UNPAK 和 PAKED 中各字节的内容。

1.22　变量 INCOME 用伪操作 DW 定义了一个 5 位的十进制数,请将此数以压缩和非压缩 BCD 码的格式,用伪操作 DB 定义在变量 INCOME1 和 INCOME2 中。

```
INCOME  DW  32767
```

第二章　IBM PC 计算机组织

复　习　提　要

1. PC XT/AT 的核心是 8088/80286 的微处理器，它能完成存取字或字节并对其进行处理等功能，也能对存储器进行管理和保护。

2. 两种类型的内部存储器是 ROM(只读存储器)和 RAM(随机存储器)。存储器按字节编址，存储器地址一般用 16 进制的无符号数表示。

3. 字数据在存储器中存放的顺序为高地址字节存放高 8 位，低地址字节存放低 8 位。

4. AX、BX、CX 和 DX 是通用寄存器，每个通用寄存器可作两个 8 位寄存器使用(如 AH 和 AL)。

5. 一个 20 位的物理地址可表示成段地址：偏移地址。计算存储器单元的物理地址，可将段地址乘以 10H，再加上偏移地址。

$$\text{物理地址} = (\text{段地址} \times 10\text{H}) + \text{偏移地址}$$

6. 一个程序可包括四个段：代码段包含可执行的指令；堆栈段包含一个后进先出的数据区，它可保存程序的返回地址，数据以及寄存器的内容；数据段是程序的数据区；附加段也是一个数据区，它通常和数据段定义在同一存储区。

7. 段寄存器 CS、SS、DS 和 ES 分别寄存代码段、堆栈段、数据段和附加段的段地址。

8. 指针寄存器 SP 和 BP，如无特殊说明，它们指示堆栈段内存储单元的地址。

9. 堆栈的最高地址叫做栈底，堆栈指示器 SP 总是指向栈顶。

10. 堆栈存取必须以字为单位。一个字存入堆栈时，SP 减 2，再把字数据存入 SP 所指示的字单元中。从堆栈中取出一个字时，将 SP 所指示的字单元中的数据取出，然后 SP 加 2。

11. 变址寄存器 SI 和 DI 一般指示数据段内单元的地址，有时也可作为数据寄存器用。

12. 16 位的标志寄存器包括 6 个状态标志(SF、ZF、PF、CF、AF、OF)和 3 个控制标志(DF、IF、TF)。CF、AF、SF、ZF 和 OF 反映了算术运算以及移位、循环、逻辑等操作的结果状态。

例　题　分　析

例 2.1　一个有 16 个字的数据区，它的起始地址为 70A0:DDF6，请写出这个数据区首末字单元的物理地址。

解：首地址为：

(70A0×10H)+0DDF6=7E7F6H

末地址为：

7E7F6H+(10H−1)×2=7E814H

物理地址是一个 20 位的数值，分别由 16 位的段地址和 16 位的偏移地址来表示。数据区的最后一个字的地址为：首地址+(字数−1)×2

例 2.2 假设堆栈段寄存器 SS 的内容为 2250，堆栈指示器 SP 的内容为 0140，如果在堆栈中存入 5 个数据，SS 和 SP 的内容各是什么？如果又从堆栈中取出 2 个数据，SS 和 SP 的内容又各是什么？

答：在堆栈中存入 5 个数据后，SP 的内容变为 0136(0140－2×5＝0136H)。如又取出 2 个数据，SP 的内容又成为 013A，(0136＋4＝013A)。SS 的内容不变。

往堆栈中存入一个数(只能以字为单位)时，(SP)←(SP)－2，然后把数存入 SP 指示的地址。从堆栈中取数据时，先将 SP 所指示的地址的内容取出，然后(SP)←(SP)＋2

习　题

2.1　一台微型计算机的字长为 16 位，如果采用字节编址，那么它可以访问的最大存储空间是多少字节？试用 16 进制数表示该机的地址范围。

2.2　IBM PC 机的 I/O 端口号通常是由 DX 寄存器来提供的，但有时也可以在指令中用一个字节来表示端口号。试问可以直接由指令指定的 I/O 端口数是多少？

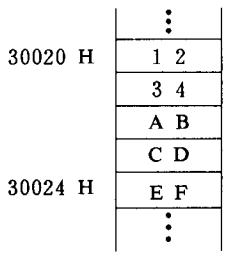
2.3　IBM PC 机有哪两种主要的存储器？它们所起的主要作用是什么？

2.4　有两个 16 位字 1EF5 和 2A3C 分别存放在 PC 机存储器的 000B0H 和 000B3H 单元中，请用图表示出它们在存储器里的存放情况。

2.5　将下列字符的 ASCII 码依次存入 00100 开始的字节单元中。(用图标出各单元的地址及其内容)。

IBM PC PERSONAL COMPUTER

2.6　在 PC 机的存储器中存放的信息如下图所示，试读出 30022，30024 字节单元的内容以及 30021、30022 字单元的内容。



2.7　写出下列存储器地址的段地址、偏移地址和物理地址。

(1) 2314：0035　　(2) 1FD0：000A

2.8　如果在一个程序段开始执行之前，(CS)＝0A7F0H，(IP)＝2B40H，试问该程序段的第一个字的物理地址是什么？

2.9　存储器中的每一段最多可含有 64K 个字节(1K＝1024)，假设用 DEBUG 命令显示出当前各寄存器的内容如下，请画出此时存储器分段的示意图，以及状态标志 OF、SF、ZF、CF 的值。

```
A> debug
-r
AX=0000  BX=0000  CX=0080  DX=0000  SP=0000
```

BP＝0000　SI＝0000　DI＝0000　DS＝10E4　ES＝10F4

SS＝21F0　CS＝31FF　IP＝0000　NV UP DI PL NZ NA PO NC

2.10　下列操作可使用哪些寄存器？

(1) 加法和减法；　(2) 循环计数；　(3) 乘法和除法；　(4) 保存段地址；

(5) 表示运算结果的特征；　(6) 将要执行的指令地址；　(7) 将要从堆栈中取出数据的地址。

2.11　IBM PC 有哪些寄存器可用来指示存储器的地址？

2.12　如果一个堆栈从地址 1250：0000 开始，它的最后一个字的偏移地址为 0100H，SP 的内容为 0052H，

问：(1) 栈顶地址是什么？

(2) 栈底地址是什么？

(3) 在 SS 中的段地址是什么？

(4) 存入数据 3445H 后，SP 的内容是多少？

2.13　请将左边的词汇和右边的说明联系起来，括号内填入所选的 A，B，C…。

(1)　CPU　　(　)
(2)　存储器　(　)
(3)　EU　　(　)
(4)　BIU　　(　)
(5)　堆栈　　(　)
(6)　IP　　(　)
(7)　SP　　(　)
(8)　状态标志　(　)
(9)　控制标志　(　)
(10) 段寄存器　(　)
(11) 物理地址　(　)
(12) 汇编语言　(　)
(13) 机器语言　(　)
(14) 汇编程序　(　)
(15) 连接程序　(　)
(16) 目标码　　(　)
(17) 指令　　(　)
(18) 伪指令　　(　)

A. 保存当前栈顶地址的寄存器。

B. 指示下一条要执行的指令的地址。

C. 总线接口部件，实现执行部件所需要的所有总线操作。

D. 分析并控制指令执行的部件。

E. 存储程序、数据等信息的记忆装置，PC 机有 RAM 和 ROM 两种。

F. 以后进先出方式工作的存储器空间。

G. 把汇编语言程序翻译成机器语言程序的系统程序。

H. 唯一代表存储器空间中的每个字节单元的地址。

I. 能被计算机直接识别的语言。

J. 用指令的助记符、符号地址、标号等符号书写程序的语言。

K. 把若干个模块连接起来成为可执行文件的系统程序。

L. 保存各逻辑段的起始地址的寄存器。PC 机有四个寄存器 CS、DS、SS、ES。

M. 控制操作的标志，PC 机有三位：DF、IF、TF。

N. 记录指令操作结果的标志，共六位：OF、SF、ZF、AF、PF、CF

O. 执行部件，由算逻单元(ALU)和寄存器组等组成。

P. 由汇编程序在汇编过程中执行的指令。

Q. 告诉 CPU 要执行的操作(一般还要指出操作数地址)，在程序运行时执行。

R. 机器语言代码。

第三章　IBM PC 指令系统与寻址方式

复　习　提　要

1. 寻址方式小结

寻址方式	操作数地址(PA)	指令格式举例
立即寻址	操作数由指令给出	MOV DX,100H;(DX)←100
寄存器寻址	操作数在寄存器中	ADD AX,BX;(AX)←(AX)+(BX)

直接寻址	操作数的有效地址由指令直接给出	MOV AX,[100]
		MOV AX, VAR
		;(AX)←(100)或(VAR)

寄存器间接寻址	PA=(DS)×16+$\begin{pmatrix}(BX)\\(SI)\\(DI)\end{pmatrix}$ PA=(SS)×16+(BP)	MOV AX,[BX] ;(AX)←((DS)×16+(BX))
寄存器相对寻址	PA=(DS)×16+$\begin{pmatrix}(BX)\\(SI)\\(DI)\end{pmatrix}$+位移量 PA=(SS)×16+(BP)+位移量	MOV AL,MESS[SI] ;(AL)←$\begin{pmatrix}(DS)\times16+(SI)\\+OFFSET\ MESS\end{pmatrix}$

基址变址寻址	PA=(DS)×16+(BX)+$\begin{pmatrix}(SI)\\(DI)\end{pmatrix}$
	或=(SS)×16+(BP)+$\begin{pmatrix}(SI)\\(DI)\end{pmatrix}$
		MOV AX,[BX][DI]
		MOV AX,[BX+DI]
		;(AX)←((DS)×16+(BX)+(DI))

相对基址变址寻址	PA=(DS)×16+(BX)+$\begin{pmatrix}(SI)\\(DI)\end{pmatrix}$+位移量 或=(SS)×16+(BP)+$\begin{pmatrix}(SI)\\(DI)\end{pmatrix}$+位移量	MOV AX,BUFF[BX][DI] ;(AX)←$\begin{pmatrix}(DS)\times16+(BX)+(DI)\\+OFFSET\ BUFF\end{pmatrix}$

2. 编写指令时,应注意的几个问题

(1) 注意区别立即寻址方式和直接寻址方式。

如：MOV AX,126;将数据 126 送入 AX 寄存器

MOV AX,[126];将数据段中的 126 单元的内容送 AX.

(2) 使用寄存器间接寻址时应注意和寄存器寻址方式的区别。

如：MOV AX,BX ;BX 中的内容传送到 AX

MOV AX,[BX];BX 所指示的地址中的内容送 AX

(3) 在双操作数指令中,源操作数和目的操作数的地址不能同时为存储器地址。

如：M1 和 M2 为两个存储器变量,

则 ADD M1,M2 是错误指令。

(4) 段跨越前缀可修改操作数所在的段。

如：MOV DL,MESS1[SI]；源操作数地址为：

;(DS)×16+(SI)+OFFSET MESS1

MOV DL, ES：MESS2[SI]；源操作数地址为：

;(ES)×16+(SI)+OFFSET MESS2

应注意：段跨越前缀不能使用 CS。

(5) 代码段寄存器 CS 不能用作指令的目的寄存器。

3. 正确使用指令系统，关键要清楚每条指令的功能以及它们规定或限制使用的寄存器。下面是初学者易混淆的几个问题：

(1) 指令对地址还是对地址中的内容进行操作，这要严格加以区分。

```
如：LEA BX,MESS              ；(BX)←MESS 的偏移地址
    MOV BX,OFFSET MESS       ；(BX)←MESS 的偏移地址
    MOV BX,MESS              ；(BX)←字变量 MESS 中的内容
```

(2) 使用指令时，要清楚指令隐含的操作寄存器。

如在乘法和除法指令中，只指出源操作数地址，但要清楚目的操作数必须存放在(AX)或(AL)中(乘法)，或(AX)、(DX：AX)中(除法)。又如串指令(MOVS、STOS、LODS、CMPS、SCAS)，它们的寻址方式也是隐含的，指令规定操作是在数据段中 SI 所指示的地址和附加段中 DI 所指示的地址之间进行串处理的；在存取串时，AL 是隐含的存取寄存器。

十进制调整指令(DAA、DAS、AAA、AAS、AAM、AAD)也隐含地使用了 AL 寄存器。

类似这些在指令语句中不反映出隐含操作数的指令还有换码指令 XLAT、循环指令 LOOP、LOOPE、LOOPNE 等，它们都要求预先在规定的寄存器内设置好操作数地址或计数值。

(3) 对带符号数和无符号数的操作应正确选择相应的条件转移指令。

(4) 用移位指令来倍增或倍减一个值是很方便的，但要注意对带符号数和无符号数所使用的指令应是不同的。

如：(AX)＝8520H，当(AX)为无符号数时，(AX)/2 可用指令 SHR AX,1，结果是(AX)＝4290H。

当(AX)为带符号数时，(AX)/2 应用指令 SAR AX,1，结果为(AX)＝0C290H

(5) 标号是程序中指令的符号地址，要注意和变量(数据符号)的区别.

如定义 VAR 是一个变量，LAB 是程序中的一个标号，则 JMP LAB 指令的转移地址为 LAB，而 JMP VAR 是一条非法指令。

例题分析

例 3.1 程序在数据段中定义的数组如下：

```
NAMES   DB      'TOM..'
        DB      20
        DB      'ROSE.'
        DB      30
        DB      'KATE.'
```

```
                DB      25
```

请指出下列指令是否正确？为什么？

```
(1) MOV BX,OFFSET NAMES
    MOV AL,[BX+5]
(2) MOV AX,NAMES
(3) MOV AX,WORD PTR NAMES+1
(4) MOV BX,6
    MOV SI,5
    MOV AX,NAMES [BX][SI]
(5) MOV BX,6 * 2
    MOV SI,5
    MOV AX, OFFSET NAMES [BX][SI]
    INC [AX]
(6) MOV BX, 6
    MOV SI, 5
    LEA DI,NAMES [BX][SI]
    MOV AL,[DI]
```

解：(1) 两条指令都是合法指令。第一条指令取得 NAMES 的偏移地址，第二条 MOV 指令使用间接寻址方式，将地址为 (DS)×10H+(BX)+5 字节中的数据传送给 AL，结果 (AL)=20。

(2) 这条指令不正确，因为 NAMES 的属性为字节，而目的寄存器是 AX，所以类型不匹配。

(3) 为合法指令。指令中将已定义的字节变量用伪操作 PTR 改变为字类型，所以避免了类型不匹配的错误。操作结果(AX)=4D4FH (即 M 和 O 的 ASCII 码)。

(4) 前两条指令使用的是立即数方式，第三条指令的源操作数字段使用的是相对基址变址方式，但形成的数据段地址中的数据属性为字节，而源操作数据寄存器为 AX，故出现“类型不匹配”的错误。如 AX 改为 AL，这条指令就是合法指令。

(5) 前两条指令是正确的，后两条指令有错误。在汇编过程中，OFFSET 操作将得到变量的偏移值，但对相对基址变址寻址方式形成的值在汇编指令时还是未知的。同样 MOV BX, OFFSET NAMES[SI]也是错误指令。第四条指令中，AX 不能作为基址寄存器用。

(6) 均为合法指令。第三条指令中的 DI 取得一个字节地址：(BX)+(SI)+OFFSET NAMES 然后再按 DI 中的偏移地址，在数据段中将一字节内容传送给 AL 寄存器。操作结果 (AL)=30。

例 3.2 两个数据段 DSEG1 和 DSEG2 的定义如下：

```
;-------------------------------
DSEG1   SEGMENT  'DATA'     ;define data segment 1
L1      EQU      100         ;Length of STR1
STR1    DB       L1 DUP(?)
ASTR2   DB       5 DUP(?)    ;For storing STR2
DSEG1   ENDS
```

```
;-------------------------------------
DSEG2   SEGMENT  'DATA'     ;define data segment 2
L2      EQU      5          ;Length of STR2
STR2    DB       L2 DUP(?)
COUNT   EQU      L1-L2+1
DSEG2   ENDS
;-------------------------------------
```

图 3.01 （例 3.2）

请编写一段程序，在串 STR1 中查找子串 STR2(子串 STR2 的长度小于 STR1)，如果 STR1 中包含有子串 STR2，则程序结束；如果 STR1 中不包含子串 STR2，则将 STR2 加在 STR1 之后。

```
;-------------------------------------
CSEG    SEGMENT
MAIN    PROC     FAR
        ASSUME CS : CSEG, DS : DSEG2, ES : DSEG1
START:
        PUSH     DS                ;for return to DOS
        SUB      AX,AX
        PUSH     AX
        MOV      AX,DSEG2          ;put dseg2 in DS
        MOV      DS,AX
        MOV      AX,DSEG1          ;put dseg2 in ES
        MOV      ES,AX

        CLD                        ;set direction flag to forward
        MOV      CX,COUNT          ;put count in CX
        MOV      BX,OFFSET STR1
NEXT:   MOV      DI,BX             ;dest addr in DI
        MOV      SI,OFFSET STR2    ;source string addr
        PUSH     CX
        MOV      CX,L2
        REP      CMPSB             ;compare STR2 in STR1
        POP      CX
        JE       MATCH             ;STR1 contain STR2
        INC      BX                ;no match,point next addr
        LOOP     NEXT              ;compare again
NO_MATCH:
        LEA      DI,ASTR2          ;STR1 not contain STR2
        LEA      SI,STR2
        MOV      CX,L2
        REP      MOVSB             ;move STR2 at after STR1
MATCH:  RET                        ;return to DOS
MAIN    ENDP
```

```
CSEG        ENDS
;-------------------------------------
END         START
```

图 3.02 (例 3.2)

如果 STR1 和 STR2 定义在同一个数据段，而且把 ES 和 DS 都初始化为同一数据段如下，则使用串指令将更方便

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

习　题

寻址方式

3.1 假定(DS)＝2000H，(ES)＝2100H，(SS)＝1500H，(SI)＝00A0H，(BX)＝0100H，(BP)＝0010H，数据变量 VAL 的偏移地址为 0050H，请指出下列指令的源操作数字段是什么寻址方式？它的物理地址是多少？

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

(4) MOV AX,VAL　(5) MOV AX,[BX]　(6) MOV AX, ES：[BX]

(7) MOV AX,[BP]　(8) MOV AX,[SI]　(9) MOV AX,[BX＋10]

(10) MOV AX,VAL[BX]　(11) MOV AX,[BX][SI]　(12) MOV AX,VAL[BX][SI]

3.2 假定(DS)＝212AH，(CS)＝0200H，(IP)＝2BC0H，(BX)＝1200H，位移量D＝5119H，(224A0)＝0600H，(275B9)＝098AH，试确定 JMP 指令的转移地址。

(1) 段内直接寻址。

(2) 使用 BX 及寄存器寻址方式的段内间接寻址。

(3) 使用 BX 及寄存器相对寻址方式的段内间接寻址。

3.3 设有关寄存器及存储单元的内容如下：

(DS)＝2000H，(BX)＝0100H，(SI)＝0002H，(20100)＝12H，(20101)＝34H，(20102)＝56H，(20103)＝78H，(21200)＝2AH，(21201)＝4CH，(21202)＝0B7H，(21203)＝65H，试说明下列各条指令执行完后 AX 寄存器的内容。

(1) MOV AX,1200H　(2) MOV AX,BX　(3) MOV AX, [1200H]

(4) MOV AX,[BX]　(5) MOV AX,1100[BX]　(6) MOV AX, [BX][SI]

(7) MOV AX,1100[BX][SI]

3.4 在 ARRAY 数组中依次存储有 7 个字数据：23,36,2,100,32000,54,0，紧接着是 ZERO 字变量单元：

```
ARRAY DW 23,36,2,100,32000,54,0
ZERO  DW  ?
```

(1) 如果 BX 包含数组 ARRAY 的初始地址，请编写指令，将数据 0 传送给 ZERO 单元。

(2) 如果 BX 包含数据 0 在数组 ARRAY 中的位移量，请编写指令将数据 0 传送给 ZERO 单元。

3.5 下面有四条等值语句：

```
C1  EQU  1000
C2  EQU  1
C3  EQU  20000
C4  EQU  25000
```

下列指令哪些是不对的？请说明原因。

(1) ADD AL,C1−C2　　(2) MOV AX,C3+C4　　(3) SUB BX,C4−C3

(4) SUB AH,C4−C3−C1　　(5) ADD AL,C2

3.6 下列 ASCII 码字符串(包括空格符)依次存储在首地址为 CSTRING 的字节单元中：

```
CSTRING DB 'BASED ADDRESSING'
```

请编写指令将字符串中的第 1 个和第 7 个字符传送给 DX 寄存器。

3.7 编写指令将附加段中的一个字节变量 COUNT 送给 AL 寄存器。

3.8 编写指令将 BX 寄存器初始化为变量 MYDAT 的偏移地址。

3.9 下列程序段完成什么工作？

```
DATX1  DB   300 DUP (?)
DATX2  DB   100 DUP (?)
  ⋮
       MOV CX,100
       MOV BX,200
       MOV SI, 0
       MOV DI, 0
NEXT:  MOV AL, DATX1 [BX][SI]
       MOV DATX2 [DI],AL
       INC SI
       INC DI
       LOOP NEXT
```

3.10 执行下列指令后，AX 寄存器中的内容是什么？

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

3.11 指出下列指令的错误：

(1) MOV AH, BX　　(2) MOV [BX],[SI]

(3) MOV AX,[SI][DI]　　(4) MOV MYDAT [BX][SI],ES : AX

(5) MOV BYTE PTR [BX],1000　　(6) MOV BX, OFFSET MYDAT [SI]

(7) MOV CS, AX　　(8) MOV DS, BP

3.12 按下列要求编写指令，将 BLOCK 数组中的第六个字数据存放在 DX 寄存器中。

(1) 寄存器间接寻址。

(2) 寄存器相对寻址。

(3) 基址变址寻址。

3.13 根据以下要求写出相应的汇编语言指令。

(1) 把 BX 寄存器和 DX 寄存器的内容相加，结果存入 DX 寄存器中。

(2) 用寄存器 BX 和 SI 的基址变址寻址方式，把存储器中的一个字节与 AL 寄存器的内容相加，并保存在 AL 寄存器中。

(3) 用寄存器 BX 和位移量 0B2H 的寄存器相对寻址方式把存储器中的一个字和(CX)相加，并把结果送回存储器单元中。

(4) 用位移量 0524 的直接寻址方式把存储器中的一个字与数 2A59 相加，并把结果送回该存储单元中。

(5) 把数 0B5H 与(AL)相加，结果送回 AL 中。

指令练习

3.14
```
DATA SEGMENT
    TABLE_ADDR DW 1234H
DATA ENDS
    ⋮
    MOV AX, TABLE_ADDR
    LEA AX, TABLE_ADDR
```
请写出上述两条指令执行完后，AX 寄存器中的内容。

3.15 完成下列操作，选用什么指令？

(1) 把4629H 传送给 AX 寄存器。

(2) 从 AX 寄存器中减去036A。

(3) 把数组 MYDAT 的段地址和偏移地址保存在 DS 和 BX 中。

3.16 设(DS)＝1B00H，(ES)＝2B00H，有关存储器地址及其内容如下图所示。请用两条指令把字变量 X 装入 AX 寄存器。

1B00:2000	8000
1B00:2002	2B00
	⋮
2B00:8000	X

3.17 写出完成下述功能的程序段：

(1) 传送25H 到 AL 寄存器

(2) 将 AL 的内容乘以2

(3) 传送15H 到 BL 寄存器。

(4) AL 的内容乘以 BL 的内容

问最后结果 (AX)＝？

3.18 写出完成下述功能的程序段：

(1) 从地址 DS:00中传送一个数据58H 到 AL 寄存器。

(2) 把 AL 的内容右移二位

(3) AL 的内容与字节单元 DS:01中的内容相乘

(4) 将乘积存入字单元 DS:02中

3.19 变量 DATAX 和变量 DATAY 的定义如下：

```
DATAX   DW   0148H
        DW   2316H
DATAY   DW   0237H
        DW   4052H
```

按下述要求写出指令序列：

(1) DATAX 和 DATAY 中的两个字数据相加，和存放在 DATAY 和 DATAY+2中。

(2) DATAX 和 DATAX 两个双字数据相加，和存放在 DATAY 开始的字单元中。

(3) 解释下列指令的作用：

```
STC
MOV BX, DATAX
ADC BX, DATAY
```

(4) DATAX 和 DATAY 两个字数据相乘（用 MUL)。

(5) DATAX 和 DATAY 两个双字数据相乘（用 MUL)。

(6) DATAX 除以23（用 DIV)。

(7) DATAX 双字除以字 DATAY（用 DIV)。

3.20 完成下面的双字乘法程序：

```
CODE     SEGMENT
         ASSUME CS:CODE, DS:CODE
         ORG 100H
BEGIN: JMP SHORT MAIN
;-----------------------
MULTCND  DW 3206H        ;被乘数低位字
         DW 2521H        ;被乘数高位字
MULTPLR  DW 6400H        ;乘数
PRODUCT  DW 0            ;乘积
         DW 0
         DW 0
;-----------------------
;Doubleword * word
MAIN PROC
       :    }双字乘法程序
MAIN ENDP
```

3.21 求双字长数 DX:AX 的相反数。

3.22 下面哪些指令是错误的?(假设 OP1,OP2是已经用 DB 定义的变量)。

(1) CMP 15, BX (2) CMP OP1, 25 (3) CMP OP1, OP2 (4) CMP AX, OP1

3.23 若(AL)=96H,(BL)=12H,指令 MUL BL 和 IMUL BL 分别执行后，它们的结果为何值? OF、CF 为何值?

3.24 写出执行以下计算的指令序列：

(1) Z←W+(Z−X)　　(2) Z←W−(X+6)−(R+9)

(3) Z←(W * X)/(R+6)　　(4) Z←((W−X)/5 * Y) * 2

3.25 已知在N到N+i的存储区中有一组ASCII码字符串(共i+1个),试编写一个汇编语言程序,将此字符串传送到NI到NI+i单元中,并使字府串的顺序与原来的顺序相反。

3.26 试编写一程序求出双字长数的绝对值。双字长数在A和A+2单元中,结果存放在B和B+2单元中。

3.27 假定(BX)=11100011B,变量VALUE的值为01111001B,确定下列各条指令单独执行后的结果。

(1) XOR BX, VALUE (2) AND BX, VALUE (3) OR BX, VALUE

(4) XOR BX, 1111,1111B (5) AND BX, 0 (6) TEST BX, 0000,0001B

3.28 编写指令序列:测试DL寄存器的低4位是否为0。

3.29 如果要检查BX寄存器中的第13位是否为1,应该用什么指令?

3.30 (1) 用一条逻辑指令清除AX寄存器。

(2) 用一条逻辑指令使DX寄存器的高3位为1,其余位不变。

(3) 写一条逻辑指令使BL寄存器的低4位为0,其它位不变。

(4) 用一条逻辑指令将AX中与BX中的对应位不相同的位均置为1。

3.31 假定(DX)=10111001b, (CL)=03, (CF)=1, 试确定下列各条指令单独执行后,DX中的值。

(1) SHR DX,1 (2) SAR DX, CL (3) SHL DX, CL

(4) SHL DL,1 (5) ROR DX, CL (6) ROL DL,CL

(7) SAL DH,1 (8) RCL DX, CL (9) RCR DL, 1

3.32 利用移位、传送和加指令完成(AX)与10的乘法运算。

3.33 把DX:AX中的双字右移4位。

3.34 下列程序段执行后,BX寄存器中的内容是什么?

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

3.35 用两条循环指令将DX:AX中的双字长数乘以2。

3.36 试分析下面的程序段完成什么功能?

```
MOV CL, 04
SHL DX, CL
MOV BL, AH
SHL AX, CL
SHR BL, CL
OR  DL, BL
```

3.37 假定一个48位数存放在DX:AX:BX中,请编写一段程序,把这个48位数乘以2。

3.38 试写出执行下列指令后,BX寄存器的内容。

```
MOV CL, 7
MOV BX, 8D16H
SHR BX, CL
```

3.39 如果把AX,BL和DH中的内容分别乘以8,连用下面的指令序列能完成此工作

吗？为什么？

```
MOV CL, 3
SHL AX, CL
SHL BL, CL
SHL DH, CL
```

3.40 要求测试字节变量 STATUS，如果第1位、或第3位、或第5位为1，则转移到 ROUTINE_1；如果第1位和第3位同时为1，则转移到 ROUTINE_2；如果第1位和第3位同时为0，则转移到 ROUTINE_3，此外的其它情况都继续执行 ROUTINE_4，试画出流程图，并编制相应的程序段。

3.41 用其它指令完成和下列指令一样的功能：

(1) REP MOVSB (2) REP LODSB (3) REP STOSB (4) REP SCASB

3.42 假设数据项定义如下：

```
CONAME  DB 'SPACE EXPLORERS INC.'
PRLINE  DB 20 DUP (' ')
```

用串指令编写程序段分别完成以下功能：

(1) 从左到右把 CONAME 中字符串传送到 PRLINE。

(2) 从右到左把 CONAME 中字符串传送到 PRLINE。

(3) 把 CONAME 中的第3个和第4个字节装入 AX。

(4) 把 AX 中的内容存入 PRLINE+5开始的字节单元中。

(5) 比较 CONAME 和 PRLINE 字符串是否相同。

(6) 扫描 CONAME 字符串中有无空格符(space)，如有，把第一个空格符的地址传送给 BX 寄存器。

3.43 编写程序段将 STRING1中的20个字符移到 STRING2中，假设 DS 和 ES 都初始化为同一数据段。

3.44 编写程序段，将字符串 STRING 中的'&'字符用空格符代替。

字符串 STRING 为：'The date is FEB&03'

3.45 有一段程序如下：

```
MOV CX,10
LEA SI, FIRST
LEA DI, SECOND
REP MOVSB
```

(1) 这个程序段完成什么动作？

(2) REP 和 MOVSB 哪条先执行？

(3) MOVB 第一次执行时，要完成什么动作？

(4) REP 指令第一次执行时，要完成什么工作？

3.46 编写一段程序，比较两个5字节的字符串 OLDS 和 NEWS，如果 OLDS 字符串不同于 NEWS 字符串，则执行 NEW_LESS，否则顺序执行程序。

3.47 编写一段程序，用空格符将字符区 CHAR_FIFLD 中的字符全部清除。字符数存在 COUNT 单元中。

3.48 编写一程序段：查找 TELEPHONE 中的电话号码(10位)有无'-'符，如有则转向