汇编语言程序设计

**习 题 册**

# 2019.07

院（系） 数学科学学院

|  |  |  |
| --- | --- | --- |
| 专 | 业 | 数理强化班 |
| 姓 | 名 | 庄镇华 |
| 学 | 号 | 1853790 |

2019年 07月 11日

1

# 第 1 章 基础知识

本章要点：数制及其转换，数及字符的编码（无符号数、有符号数、补码、ASCII 码）， 基本逻辑运算。了解汇编语言的特点及应用场合。

一、单项选择题

1.1.1 下列四个数均为有符号数，其中最小的数是（**B**）。

A. 1234H B. 8793H C. 9645H D. 5640H

1.1.2 下列四个无符号数中，数值最大的数是（**C**）。

A. 01001001B B. 4AH C. 75 D. 01001000B

1.1.3 十进制数 36.875 转换成二进制数是（**B**）。

A. 110100.01 B. 100100.111 C. 100110.11 D. 100101.101

1.1.4 已知 X=34，Y= -68，用补码计算 X+Y=（**D**）。

A. 11010001 B. 00011101 C. 10010110 D. 11011110

1.1.5 十六进制数 88H，可表示成下面几种形式，请找出正确的表示(**A**)。 A. 带符号十进制数-120 的补码 B. 数字 8 的 ASCII 码

C. 带符号十进制数-8 的补码表示 D. 无符号十进制数 88 1.1.6 回车符的 ASCII 码是（**C**）。

A. 10H B. 0AH C.0DH D. 20H

二、填空题

1.2.1 十进制数 25 转换为二进制数和十六进制数分别是 **11001B** 和 **19H** ；十进制数 78.3125 转换为二进制数和十六进制数分别是 **1001110.0101B**  和 **4E.5H**  。

1.2.2 二进制数 101101 转换为十进制数和十六进制数分别是 **45** 和 **2DH** ； 二进

制数 10000000 转换为十进制数和十六进制数分别是 **128** 和 **80H**  。

1.2.3 十六进制数 5B 转换为二进制数和十进制数分别是 **1011011B** 和 **91** ； 十六进制数 1234 转换为二进制数和十进制数分别是 **1001000110100B** 和  **4660** 。

1.2.4 D8 是十六进制表示的 8 位二进制数，当它被看作是用补码表示的带符号数时，所 表示的十进制数是 -**40**,当被看作是无符号数时所表示的十进制数是 **216** 。

1.2.5 如果用 24 位二进制数表示一个无符号数，这个数的范围是 **0~2^24-1** ；

如果用 24 位二进制补码表示一个有符号数，这个数的范围是 -2^23~2^23-1。

1.2.6. 4B 是十六进制表示的8位二进制数，当它被看作是用补码表示的数时所表示的十 进制数是 75 ；当被看作是字符的 ASCII 码时，表示的字符是 **K** 。

1.2.7 下列 6 个用十六进制数表示的二进制数：(1)6813H， (2)EAFAH，(3)CD06H， (4)1103H，(5)3BD6H，(6)B758H。如果作为无符号数，将它们从小到大排序为（将数据 的序号排序写）： **(4)(5)(1)(6)(3)(2)** ；如果作为带符号数，将它们

从小到大排序为：  **(6)(3)(2)(4)(5)(1)**  。

三、简答题

1.3.1 分别用降幂法和除法将十进制数 10000 转换为二进制数，要求写出运算过程，然 后将转换成的二进制数直接写出对应的十六进制数。

**降幂法：8192 4096 2048 1024 512 256 128 64 32 16 8 4 2 1**

**1 0 0 1 1 1 0 0 0 1 0 0 0 0**

**除2取余数法：10000/2=5000…0**

**5000/2=2500…0**

**2500/2=1250…0**

**1250/2=625…0**

**625/2=312…1**

**312/2=156…0**

**156/2=78…0**

**78/2=39…0**

**39/2=19…1**

**19/2=9…1**

**9/2=4…1**

**4/2=2…0**

**2/2=1…0**

**1/2=0…1**

**10000D=10011100010000B=2710H**

1.3.2 下列 8 位二进制数是补码表示的带符号数，请给出其等值的 16 位和 32 位形式 思

考：如果是无符号数呢？ **带符号数需要符号扩展，而无符号数只需要零扩展** ）

（1） 9FH

（2） 68H

（3） 0C0H

**FF9FH； FFFFFF9FH**。

**0068H； 00000068H**。

**FFC0H； FFFFFFC0H**。

\*1.3.3 两个 16 位无符号数相加、相减时，什么情况下运算结果会溢出？有什么判断的 方法？

**超出 0～65535 的范围时溢出。判断：有进位或借位时溢出**。

\*1.3.4 两个 16 位带符号数相加、相减时，什么情况下运算结果会溢出？有什么判断的 方法？

**超出-32768～+32767 的范围时溢出。判断：**

**相加时：异号不会溢出，正加正为负或负加负为正则溢出。**

**相减时：同号不会溢出，正减负为负或负减正为正则溢出。**

1.3.5 十进制数 76 和 85 的 8 位二进制补码分别是 **01001100B**和 **01010101B**；-76 和

-85 的 8 位二进制补码分别是 **10110100B**和 **10101011B**；

请用二进制补码计算下列 各题，要求写出运算过程，用十六进制数表示其运算结果，并回答结果是否溢出。

1. （-85）+76

**10101011**

**+ 01001100**

**11110111**

运算结果为 **F7H** 是否溢出？ **N**（Y/N）

1. (-85)-76

**10101011**

**+ 10110100**

**01011111**

运算结果为 **5FH** 是否溢出？ **Y**（Y/N）

（3）85 –76

**01010101**

**+ 10110100**

**00001001**

运算结果为 09**H** 是否溢出？ **N**（Y/N）

（4）85-（-76）

**01001100**

**+ 01010101**

**10100001**

运算结果为 A1H 是否溢出？ Y（Y/N）

\*1.3.6 说明下列二进制数可能表示的多种含义,至少说出 3 种：

（1） 00110001 **无符号数49； 有符号数49； ‘1’; 31BCD**。

（2） 11111111 **无符号数255； 有符号数-1； 扩展后的字符ASCII码； 逻辑真的八位二进制**；。

（3） 1011000010100001**无符号数45217； 有符号数-20318 ；具体的机器指令**；。

\*\*1.3.7 给出下列十进制数对应的压缩和非压缩 BCD 码形式：

（1） 16

（2） 328

**16H； 0106H**。

**328H； 030208H**。

\*1.3.8 给出实现下列功能的位操作（逻辑操作）：（提示：AL 为 8 位寄存器，已给出第

1 小题答案，根据提示完成其余各小题。）

（1）将 AL 寄存器中的数字字符’0’～’9’的 ASCII 码转换为对应的数字值 0～9：

AND AL,0FH

（2）将 AL 寄存器中的数字值 0～9 转换为对应数字字符’0’～’9’的 ASCII 码：

**OR AL,030H**

（3）将 AL 寄存器中的 8 位二进制数的低 4 位提取出来：

**AND AL,0FH**

（4）将 AL 寄存器中的 8 位二进制数的 0、2、4、6 位取反，其余位不变：

**XOR AL,55H**

思考题

1.0.1 简述学习汇编语言程序设计的目的。

**1.掌握汇编本身在直接控制硬件和高效时空效程序方面的优势。**

**2.编写运行速度快、代码量小的程序。**

**3.感知计算机的运行过程和原理。**

**4.加深理解高级语言语句的执行过程，即使使用高级语言编程，也有助于有意识地编写更高效、健壮的程序代码。**

1.0.2 简述机器语言、汇编语言与高级语言的区别。

**机器语言是最早使用的编程语言，是计算机唯一可以直接识别和执行的语言。**

**汇编语言是机器语言的符号表示，与机器语言无本质区别。也可以说是机器语言的符号化,与机器语言一一对应。**

**高级语言较为接近自然语言和数学公式的编程语言，基本脱离了机器的硬件系统，用人们更易理解的方式编写程序。**

# 第 2 章 80x86 编程的硬件基础

本章要点: 80x86 的寄存器的分类、作用以及有关寄存器的特定用法，内存及其分段， 逻辑地址和物理地址，I/0 端口地址。

一、单项选择题

2.1.1 80x86 的寄存器中，8 位的寄存器共有（**C**）个。(**AL AH BL BH CL CH DL DH**)

A. 4 B. 6 C. 8 D. 10

2.1.2 总是指向下一条要执行的指令，由此实现程序的自动执行的寄存器是（**B**）。

A. BP B. IP C. SP D. IR

2.1.3 标志寄存器用来保存算术逻辑运算的结果状态，其中用于表示当前运算结果是否 为 0 的标志位是（**D**）。

A. CF B. OF C. SF D. ZF **2.1.4** 80x86CPU 执行算术运算时，FLAGS 共有（**B**）个标志位受影响。

A. 5 B. 6 C. 7 D. 9

**2.1.5** 一个 16 位相对位移的范围是（**C(讲到相对时有正负之分**）。

A. -128～127 B. 0～65535 C. 8000H～7FFFH D. 8000H～FFFFH

2.1.6 如果某一存储单元的物理地址为 12345H，则它的逻辑地址为（**D**）：0345H。 A. 12000H B. 0012H C. 0120H D. 1200H

2.1.7 通常我们所说的 32 位机，是指这种计算机的 CPU（**C**）。

A. 由 32 个运算器组成 B. 包含 32 个寄存器

C. 能够同时处理 32 位二进制数 D. 一共有 32 个运算器和控制器

2.1.8 下列寄存器组中，用于提供段内偏移地址的寄存器组是（**C**）。

A. AX，BX，CX，DX B. BX，BP，SI，DI

C. SP，BP，IP，DX D. CS，DS，ES，SS

2.1.9 在80x86 系统中，约定用于形成堆栈段数据物理地址的寄存器有（**B**）。

A. DS，DX，BX B. SS，BP，SP

C. SS，BX，BP D. DS，BP，SP

2.1.10 在程序的运行过程中，确定下一个指令的物理地址的计算表达式是（**C**）。

A. DS×16+SI B. ES×16+DI C. CS×16+IP D. SS×16+SP

二、填空题

**2.2.1**  在 80x86 的 16 位寄存器中，可以用来指示存储器地址的有 **10** 个，它们分别是

**BX BP SP SI DI IP CS SS DS ES** ；既可以用来指示存储器地址又可 以用来存放操作数的有 **5** 个，它们分别是 **BX BP DI SI SP** 。

2.2.2 在实模式下，段地址和偏移地址为 3017：000AH 的存储单元的物理地址是

**3017A** H；段地址和偏移地址为 3015：002AH 的存储单元的物理地址是 **3017A** H； 段地址和偏移地址为 3010：007AH 的存储单元的物理地址是 **3017A** H；这个结果说明了什么？ \_**同一物理地址可以由不同的段地址和偏移地址组成** 。

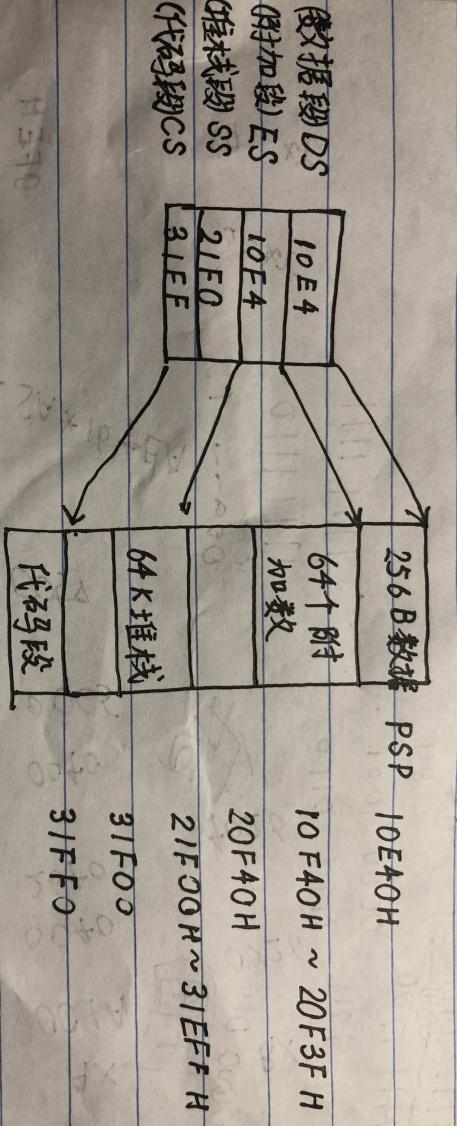
**2.2.4** 在实模式下，2.2.3 设有一个包含 20 个字的数据区，起始地址为 10F0:02A0H，则该数据区的首字单 元的物理地址是 **111A0** H；末字单元的物理地址是 **111DA** H。

存储器中每一段最多有 10000H（即 64K）个字节，在 DEBUG 下用

R 命令所显示的当前各寄存器的内容和各标志的状态如下:

AX=0000 BX=0200 CX=0014 DX=0020 SP=0E8C BP=0080 SI=0006 DI=0000 DS=10E4 ES=10F4 SS=21F0 CS=31FF IP=0106 NV UP EI NG ZR NA PE NC

请画出此时存储器分段的示意图，并回答以下问题：



（1）当前的下一条指令的物理地址是 **320F6** H；当前栈顶的物理地址是 **22D8C** H；

**（2）**状态标志 OF、SF、ZF、CF 的当前值（用 0 或 1 表示）分别是**0 ， 1 ， 1 ，0** ；

**2.2.5** 进行下列操作时，通常使用哪个或哪几个 16 位寄存器来完成？

（1）加、减运算：**AX BX CX DI SI BP** ；

（2）乘法、除法：**AL AX DX**  ；

（3）循环计数： **CX**  ；

（4）保存段地址： **CS SS DS ES** ；

（5）作为指针使用： **SI DI SP BP BX** ；

（6）存放端口地址：**DX**  ；

\*2.2.6 80x86 微机的 I/O 地址空间可达\_\_**64**  KB，端口地址的范围是 0000～FFFFH。 在输入输出指令中，端口号通常由

**DX** 寄存器提供；有时也可以在指令中直接指 定 00～FFH 的端口号，8 位的端口数共有 **256** 个。

\*2.2.7 以后进先出的方式工作的存储空间称为 **堆栈** ；能被计算机直接识别的语

言是**机器** 语言；用指令的助记符、符号地址表示的面向机器的语言称为**汇编**  语言； 把若干个模块连接起来成为可执行文件的系统程序是 **连接程序** 。

三、简答题

2.3.1 80x86 微机的存储器中存放信息如图所示，请写出 30022H 和 30024H 字节单元的 内容分别是什么？以及 30021H 和 30022H 字单元的内容分别是什么？

**0ABH,56H,0AB34H,0CDABH**

|  |  |
| --- | --- |
| 30020H  30021H  30022H  30023H  30024H | : |
| 12H |
| 34H |
| 0ABH |
| 0CDH |
| 56H |
| : |

2.3.2 有两个 16 位字 5EE1H 和 2A3CH 分别存放在 80x86 微机的存储器的 100B0H 和

100B3H 单元中，请用图表示出它们在存储器里的存放情况。

|  |  |
| --- | --- |
| 100B0H  100B1H  100B2H  100B3H  100B4H | : |
| **E1H** |
| **5EH** |
|  |
| **3CH** |
| **2AH** |
| : |

2.3.3 从内存地址 2000H 开始，依次存放着 3 个数：1（字）、2BH（字节）和 351DH（双 字），请用图表示出它们在存储器里的存放情况。

|  |  |
| --- | --- |
| 2000H  2001H  2002H  2003H  2004H  2005H  2006H | : |
| **01H** |
| **0** |
| **2BH** |
| **1DH** |
| **35H** |
| **0** |
| **0** |
| : |
|  |  |

**2.3.4** 给出下列 8 位二进制数相加后 CF、OF、SF、ZF 的值（用 0 或 1 表示）？如果把 它们看作是**无符号数相加**则如何判断溢出（用 1 表示有，0 表示无）？

CF OF SF ZF 看作无符号数时的溢出判断

（1） 0FFH + 01H： **1 0 0 1** **1 1 0 1** ；

（2） 0FEH + 01H： **0 0 1 0** **0 0 0 0** ；

（3） 80H + 81H： **1 1 0 0** **1 0 0 0**  ；

（4） 7FH + 02H： **0 1 1 0** **0 0 0 0**  ；

**2.3.5** 给出下列 8 位二进制数相减法 CF、OF、SF、ZF 的值（用 0 或 1 表示）？如果把 它们看作是无符号数相减则如何判断溢出（用 1 表示有，0 表示无）？

CF OF SF ZF 看作无符号数时的溢出判断

（1） 09H - 05H：  **0 0 0 0** **0 0 0 0**  ；

（2） 05H - 09H： **1 0 1 0**  **1 1 0 0** ；

（3） 80H - 01H：  **0 1 0 0 0 0 0 0** ；

（4） 7FH – 0FEH： **1 1 1 0 1 1 0 0** ；

# 第 3 章 数据定义与数据传送

本章要点: 数据在内存中的存放格式、常量、变量、标号的定义和使用，地址计数器， 汇编指令的格式、操作数的分类、操作数的寻址方式。常用数据传送指令、汇编语言程 序的编程格式和常用伪指令。汇编语言程序的上机过程，DEBUG 的使用。

一、单项选择题

3.1.1 下列选项中，不能作为变量名或标号的是（**B** ）。

A. FH B. 4G C. C3 D. FBCD 3.1.2 一个段最大可定义（**C** ）个字节。

A. 16K B. 32K C. 64K D. 1024K

3.1.3 在下列语句中的名字 N10，称为变量的是（**A** ）。

A. N10 DB 10,10H B. N10：SHL AX,CL

C. N10 EQU 10 D. N10 SEGMENT

3.1.4 执行 LINK 命令以后，可以生成一个以（**C** ）为扩展名的文件。

A. ASM B. COM C. EXE D. OBJ 3.1.5 汇编语言源程序以（**A** ）语句结束。

A. END B. ENDM C. ENDP D. ENDS 3.1.6 用来作为寄存器间接寻址的寄存器有（**A(基址寄存器：BX,BP变址寄存器：SI,DI)** ）个。

A. 4 B. 5 C. 6 D. 8

3.1.7 汇编语言中的变量有多种类型属性，下面错误的类型是（**C** ）。

A. 字节型 BYTE B. 字型 WORD

C. 字符型 CHAR D. 双字型 DWORD

**3.1.8** 数据存放在堆栈段，以基址变址寻址的地址可以存放在（**D** ）寄存器中。

A. AX、BP B. BX、SI C. DX、DI D. BP、SI 3.1.9 下列指令中的操作数在代码段中的是（**B(立即数占用代码段的空间** ）。

A. MOV AL,BL B. MOV AL,20H

C. MOV AX,[100H] D. MOV AX,CS

**3.1.10**  在绝大多数 8086/8088 指令系统的双操作数指令中.其中一个必须是（**B** ）。

A. 立即数 B. 寄存器操作数 C. 内存操作数 D. 端口操作数

3.1.11 在双操作数指令中，目的操作数不能使用（**A** ）。

A. 立即数 B. 寄存器 C. 存储器 D. 段寄存器

**3.1.12** 若定义“A DW 1,2,3 DUP(?)”，则该伪指令共分配（**C** ）个字节单元。

A. 5 B. 7 C. 10 D. 14

**3.1.13** 下列指令中，错误的指令是（**C** ）。

A. MOV AX,2000H B. MOV AL,20H

C. MOV DS,2000H D. MOV DS,AX

3.1.14 执行指令 MOV SP,3210H 和 PUSH AX 后，SP=（**D** ）H。

A. 3212 B. 3211 C. 320F D. 320E

**3.1.15** 设 A、B 为字变量，C 为标号，下面指令中不正确的是（**C** ）。

A. MOV AX,A B. MOV BX,B

C. JNC A D. JMP C

3.1.16 设 BX=2000H，SI=3008H，指令 MOV AX，[BX+SI+8]的源操作数的有效地址为

（**B** ）。

1. 5008H B. 5010H C. 5016H D.23008H

二、填空题

3.2.1 标号的类型有 **NEAR** 和 **FAR** ；变量的三个属性是段地址、 **偏移地址** 和类型。

3.2.2 伪指令 **END** 表示整个程序结束。

3.2.3 设有关寄存器及存储单元的内容如下:

DS=2000H，BX=0100H，SI=0002H，(20100H)=12H，(20101H)=34H，(20102H)=56H， (20103H)=78H，(21200H)=2AH，(21201H)=3CH，(21202H)=0B7H，(21203H)=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 | **1200H** |
| （2） | MOV | AX,BX | **100H** |
| （3） | MOV | AX,[1200H] | **3C2AH** |
| （4） | MOV | AX,[BX] | **3412H** |
| （5） | MOV | AX,1100H[BX] | **3C2AH** |
| （6） | MOV | AX,[BX][SI] | **7856H** |
| （7） | MOV | AX,1100H[BX][SI] | **65B7H** |

3.2.4 设有关寄存器及存储单元的内容如下:

(DS)=2000H，(ES)=2100H，(SS)=1500H，(SI)=00A0H，(BX)=0100H，(BP)=0010H，数

据段中变量名 VAL 的偏移地址值为 0050H，试指出下列各条指令源操作数的寻址方式 以及其物理地址:

|  |  |  |  |
| --- | --- | --- | --- |
| （1） | MOV | AX,0ABH | **立即寻址 ；物理地址： 无** |
| （2） | MOV | AX,BX | **寄存器寻址 ；物理地址： 无** |
| （3） | MOV | AX,[100H] | **直接寻址 ；物理地址： 20100H** |
| （4） | MOV | AX,VAL | **直接寻址 ；物理地址：20050H** |
| （5） | MOV | AX,[BX] | **间接寻址 ；物理地址： 20100H** |
| （6） | MOV | AX,ES:[BX] | **间接寻址 ；物理地址： 21100H** |
| （**7**） | MOV | AX,[BP] | **间接寻址 ；物理地址：15010H** |
| （8） | MOV | AX,[SI] | **间接寻址 ；物理地址：200A0H** |
| （**9**） | MOV | AX,[BX+10] | **相对寻址 ；物理地址： 2010AH** |
| （**10**） | MOV | AX,VAL[BX] | **相对寻址 ；物理地址： 20150H** |
| （**11**） | MOV | AX,[BX][SI] | **基址变址 ；物理地址： 201A0H** |
| （**12**） | MOV | AX,VAL[BX][SI] | **相对基址变址 ；物理地址： 201F0H** |

3.2.5 设(SI)=2A9BH，(BX)=637DH，16 位位移量 D=7237H，请仿照例子，写出以下各

种寻址方式下的 MOV 指令的格式，并确定有效地址：

|  |  |  |
| --- | --- | --- |
| （1）立即寻址 | MOV AX,1234H | 有效地址： 无 |
| （2）直接寻址 | **MOV AX,[7237H]** | 有效地址： **7237H** |
| （3）使用 BX 的寄存器寻址 | **MOV AX,BX** | 有效地址： **无** |
| （4）使用 BX 的间接寻址 | **MOV AX,[BX]** | 有效地址： **637DH** |

（5）使用 BX 的相对寻址  **MOV AX,D[BX]** 有效地址： **D5B4H**

（6）基址变址寻址 MOV AX,[BX][SI] 有效地址： 8E18H

（7）相对基址变址寻址 **MOV AX,D[BX][SI]**  有效地址： **004FH**

**3.2.6** 设 TABLE 为数据段中 0032H 单元的符号名，其中存放的内容为 1234H，请问以 下两条指令分别执行后 AX 寄存器的内容分别是 **1234H** 和**0032H**  。

（1）MOV AX,TABLE

（2）LEA AX,TABLE

这两条指令的区别是 **内容，地址** 。

* + 1. 已知 SP=1FFEH,BX=1565H,CX=7FFFH,则

（1）执行 PUSH BX 指令后，SP 的内容为 **1FFC** H；

（2）再执行 PUSH CX 和 POP AX 后，SP= **1FFC** H，AX= **7FFF** H。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 3.2.8 指出下列指令的错误: | | | | |
| （1） | MOV | | AX,BH | **类型不一致** |
| （2） | MOV | | [BX],5 | **类型不确定** |
| **（3）** | MOV | | AX,[SI][DI] | **SI,DI不能同时使用** |
| （**4**） | MOV | | [BX] ,[SI] | **不能同时为两个内存** |
| （5） | CMP | | 5, [BX] | **立即数不能做目的操作数** |
| （**6**） | MOV | | CS,AX | **CS不能做目的操作数** |
| 3.2.9 设数据定义如下，判断各语句是否正确，如有错误则改正: | | | | |
| DATA | | SEGMENT | | **正确** |
| A | | DB 123H | | **DW** |
| B | | DW 23，45678H | | **DD** |
| C | | DW ‘ABCD’ | | **DB** |
| D | | DB 100 DUP(‘ABC’) | | **正确** |
| E | | DB ($-D)/3 | | **正确** |
| DATA | | END | | **ENDS** |

\*3.2.10 对于下面的数据定义，各条 MOV 指令单独执行后，有关寄存器的内容是什么？

FLDB DB ?

TABLEA DW 20 DUP(?) TABLEB DB ‘ABCD’

|  |  |  |  |
| --- | --- | --- | --- |
| （1） | MOV | AX,TYPE FLDB | **1** |
| （2） | MOV | AX,TYPE TABLEA | **2** |
| （3） | MOV | CX,LENGTH TABLEA | **20** |
| （4） | MOV | DX,SIZE TABLEA | **40** |
| （5） | MOV | CX,LENGTH TANLEB | **1** |

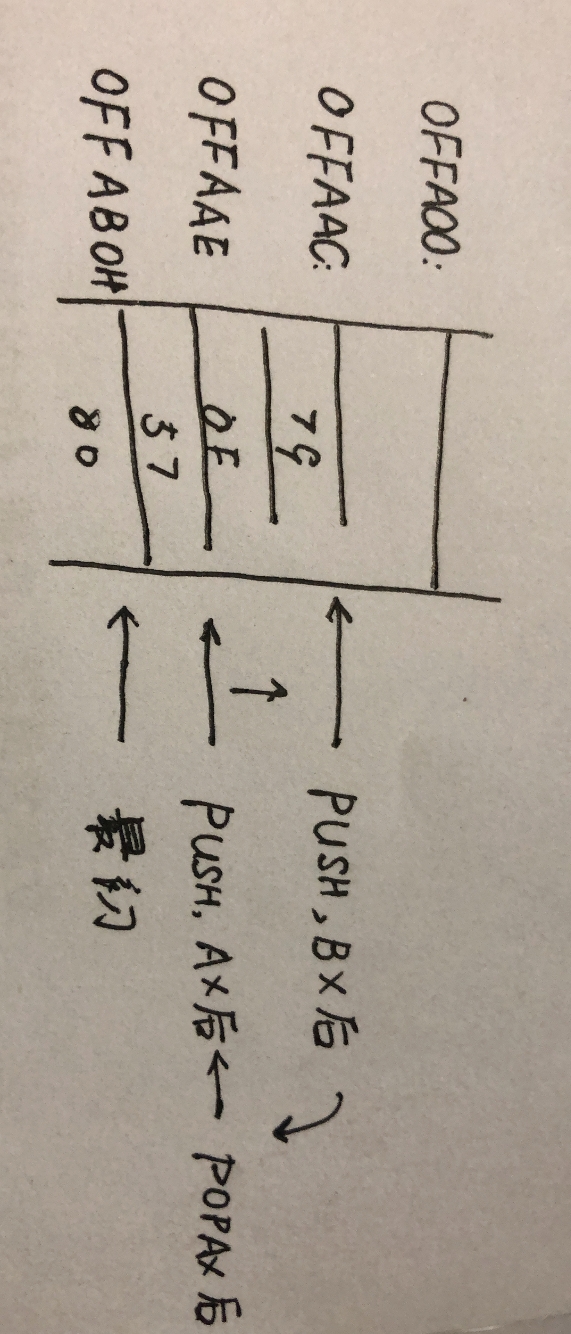
**3.2.11** 试说明下述指令中哪些需要加上 PTR 伪操作？需要的请填上正确的使用。

|  |  |  |  |
| --- | --- | --- | --- |
|  | BVAL WVAL | DB 20H,10H  DW 2000H |  |
| （1） | MOV | AL,BVAL | **正确** |
| （2） | MOV | DL,[BX] | **正确** |
| （3） | SUB | [BX],2 | **SUB BYTE PTR [BX],2** |
| （4） | MOV | CL,WVAL | **MOV CL,BYTE PTR WVAL**  **Z** |
| （5） | ADD | AL,BVAL+2 | **正确** |

三、简答题

3.3.1 已知(SS)=0FFA0H，(SP)=00B0H，(AX)= 8057H，(BX)= 0F79H，先执行 PUSH AX

和 PUSH BX，再执行 POP AX。试画出堆栈变化图，用物理地址标出 SP 的变化过程， 并回答 AX 的内容最后是多少？ **0F79H**



3.3.2 设程序中的数据定义如下: PARTNO DB ?

PNAME DW 16 DUP(?) COUNT DD ?

PLENTH EQU $-PARTNO

问 PLENTH 的值是多少？ 22 \_；它表示什么意义？ 以上数据的总字节数

3.3.3 有符号定义语句如下:

BUFF DB 1,2,0,‘12’ EBUFF DW ?

L EQU EBUFF-BUFF

问 L 的值是多少？5 ；它表示什么意义？BUFF所占的字节数

四、程序分析题

3.4.1 下列程序段执行后，AX 寄存器的内容是1E00 H。

|  |  |  |
| --- | --- | --- |
| TABLE | DW | 10,20,30,40,50 |
| ENTRY | DW | 3 |
|  | : |  |
|  | MOV ADD | BX,OFFSET TABLE  BX,ENTRY ;加法指令，BX←(BX)+(ENTRY) |
|  | MOV | AX,[BX] |

3.4.2 设数据定义如下：

DATA SEGMENT ORG 100H

1. EQU 10
2. DW ‘AB’,12
3. DB 101B
4. EQU B+3

DATA ENDS

请回答：（1）B、C 的偏移地址分别是 100 H 和104 H；

（2）执行“MOV BX，D”后，BX= 500 H。 3.4.3 设数据定义如下：

DATA SEGMENT

A DB 2，3

B DW ‘AB’，5，6，7

1. EQU 10
2. DB 20 DUP(?)
3. EQU $-A
4. DW D

DATA ENDS

请回答：

（1）该数据段占用的内存有多少个字节？ 32 ；

（2）E 的内容是多少？ 30 ；

（3）F 的内容是多少？0A H；

（4）执行指令“MOV AX,WORD PTR A+1”后 AX=4203 H；（5）执行指令“MOV EAX, DWORD PTR B+1” 后 EAX= 06000541 H。

# 第 4 章 数据运算与输入输出

本章要点: 算术运算指令、逻辑运算与移位指令，数值计算和表达式计算。字符和字符 串的输入输出，在程序中完成将计算结果在屏幕上显示输出。

一、单项选择题

4.1.1 逻辑移位指令 SHR 用于（D ）；算术移位指令 SAR 用于（B ）。

A. 带符号数乘以 2 B. 带符号数除以 2

C. 无符号数乘以 2 D. 无符号数除以 2 4.1.2 INC 指令不影响（A ）标志。

* + - 1. CF B. OF C. SF D. ZF 4.1.3 要完成(AX)×7/2 运算，应在下来 4 条指令后添加（C ）指令。

MOV BX,AX MOV CL,3 SAL AX,CL SUB AX,BX

A. DIV AX,2 B. SAL AX,1 C. SAR AX,1 D. ROR AX,1 4.1.4 指令 MOV AX,[BP+SI]中的源操作数使用（D ）段寄存器。

A. CS B. DS C. ES D. SS 4.1.5 使 BX 寄存器内容清零的错误指令是（B ）。

A. MOV BX,0 B. OR BX,0 C. XOR BX,BX D. SUB BX,BX

\*4.1.6 已知 AL 和 BX 中存放的都是带符号数据，现要计算 AL×BX，应该使用下列程 序段（C ）。

A. MOV AH,0 B. MOV AH,0 C. CBW D. IMUL AL,BX MUL BX IMUL BX IMUL BX

4.1.7 执行“（A ）AX,BX”指令中不改变 AX 寄存器的内容。

A. CMP B. ADD C. XOR D. OR 4.1.8 十进制数字 85 所对应的非压缩型 BCD 码形式是（D ）。

A. 0085 B. 0085H C. 0805 D. 0805H

4.1.9 设 AX=1000H，执行“NEG AX”指令后，AX=（D ）。

A. 1000H B. 1001H C. 0E000H D. 0F000H

4.1.10 执行“DIV BL”指令后，商存放在（A ）寄存器中。

A. AL B. AH C. BL D. AX

二、填空题

4.2.1 现有程序段如下：

MOV AX,N MOV DX,M

SHR DX,1

RCR AX,1

请回答：（1）该程序段完成的功能是 将N:M表示的32位数右移一位 ；

（2）若 N=1234H，M=5678H，程序运行后 DX= 091AH ，AX=2B3CH \_\_。

4.2.2 下列程序段中的每条指令执行完后，AX 寄存器及 CF、SF、ZF 和 OF 的内容是什

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 么？请在下面相应的地方填入结果： | AX | CF | SF | ZF | OF |
| MOV AX,0 ； | 0H | × | × | × | × |
| DEC AX ； | FFFFH | × | 1 | 0 | 0 |
| ADD AX,7FFFH ； | 7FFEH | 1 | 0 | 0 | 0 |
| ADD AX,2 ； | 8000H | 0 | 1 | 0 | 1 |
| NOT AX ； | 7FFFH | × | × | × | × |
| SUB AX,0FFFFH ； | 8000H | 1 | 1 | 0 | 1 |
| ADD AX,8000H ； | 0 | 1 | 0 | 1 | 1 |
| SUB AX,1 ； | FFFF | 1 | 1 | 0 | 0 |
| AND AX,58D1H ； | 58D1 | 0 | 0 | 0 | 0 |
| SAL AX,1 ； | B1A2 | 0 | 1 | 0 | 1 |
| SAR AX,1 ； | D8D1 | 0 | 1 | 0 | 0 |
| NEG AX ； | 272F | 1 | 0 | 0 | 0 |
| ROR AX,1 ； | 9397 | 1 | × | × | 1 |

提示：如果某指令对标志位有影响填 0 或 1，没有影响的则填 “×” 。

4.2.3 下列程序段中的每条指令执行完后，AL 寄存器的内容,以及 CF、AF、PF、SF、 ZF 和 OF 标志的值是什么？请在下面相应的地方填入结果：

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | AL | CF | AF | PF | SF | ZF | OF |
| MOV | AL,45H | ； | 45H | × | × | × | × | × | × |
| ADD | AL,AL | ； | 8AH | 0 | 0 | 0 | 1 | 0 | 1 |
| ADC | AL,9FH | ； | 29H | 1 | 1 | 0 | 0 | 0 | 1 |
| CMP | AL,0ACH | ； | 29H | 1 | 1 | 1 | 0 | 0 | 0 |
| SUB | AL,34H | ； | 0F5H | 1 | 0 | 1 | 1 | 0 | 0 |
| DEC | AL | ； | 0F4H | × | 0 | 0 | 1 | 0 | 0 |
| NEG | AL | ； | 0CH | 1 | 1 | 1 | 0 | 0 | 0 |

提示：如果某指令对标志位有影响填 0 或 1，没有影响的则填 “×” 。

4.2.4 用 移 位 指 令 把 AL 寄 存 器 中 的 十 进 制 数 +53 乘 以 2 ， 应 该 用 的 指 令 是

SHL AL,1 ，得到的结果是 6A H；用移位指令把 AL 寄存器中 的-49 除以 2，应该用指令 SAR AL,1 ，得到的结果是 E7 H； 4.2.5 写出实现下列要求的单条指令：

（1）把 2345H 传送给寄存器 AX MOV AX,2345H ；

（2）从 AX 中减去 1234H SUB AX,1234H ；

（3）把字节变量 BVAR 的偏移地址送入 SI LEA SI,BVAR ；

（4）把字变量 WVAR 的内容送入 BX MOV BX,WVAR ；

（5）将 AX 的高 4 位清 0 AND AX,0FFFH ；

（6）把 BX 的低 4 位置 1 OR BX,OFH ；

三、简答题

* + 1. 简述指令 AND 和 TEST，NOT 和 NEG 之间的区别： 1，TEST将两个数进行与运算，但不回送结果 2，NEG为取反加一，NOT为取反，NOT不影响标志位

四、程序分析题

4.4.1 下列程序段执行后，BX 寄存器的内容是 00DA H。

MOV BX,6D16H MOV CL,7

SHR BX,CL

4.4.2 下列程序段执行后，BX 寄存器的内容是 802D H。

MOV CL,3 MOV BX,0B7H

ROL BX,1

RCR BX,CL

4.4.3 分析下面程序段，请回答（1）程序段运行后，DX= 2304 H；AX=\_ 5670 H；

BL= 04 H；（2）该程序段完成的功能是 将DX:AX的双字左移4位 。

|  |  |  |  |
| --- | --- | --- | --- |
| MOV | AX,4567H | MOV | BL,AH |
| MOV | DX,1230H | SHL | AX,CL |
| MOV | CL,4 | SHR | BL,CL |
| SHL | DX,CL | OR | DL,BL |

4.4.4 已知 AX=4A50H,CX=5402H,CF=1,下列程序段执行后，AX= 0010 H；CF= 1 。

RCL AX,CL

AND AH,CH

RCR AX,CL

4.4.5 已知 AX=0CF49H,CX=0302H,CF=0,下列程序段执行后，AX= 9E80 H；CF= 1 。 SAR AX,CL

XCHG CH,CL SHL AX,CL

4.4.6 现有下面程序段：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | : |  | SUB | AX,Y |
| X | DW | 100 | MOV | CL,4 |
| Y | DW | 20 | SAL | AX,CL |
| Z | DW | ? | ADD | AX,20 |
|  | : |  | SAR | AX,1 |
|  | MOV | AX,X | MOV | Z,AX |

请回答：（1）该程序完成的功能是 \_计算 Z=((X-Y)\*16+20)/2\_ ；

（2）程序执行后，Z= 028A H。

**五、编写程序题**

4.5.1 写出执行以下计算的指令序列，其中 X、Y、Z、R 和 W 均为存放 16 位带符号数 单元的地址。（不考虑溢出）

（1）Z←W+(Z-X) （2）Z←(W\*X)/(Y+6)，R←余数

MOV AX,Z MOV AX,W IDIV BX

SUB AX,X IMUL X

MOV Z,AX

ADD AX,W MOV BX,Y

ADD BX,6

MOV Z,AX

IDIV BX

MOV Z,AX

MOV R,DX

4.5.2 写出对存放在 DX 和 AX 中的双字长数**求补**的指令序列：（请使用多种方法）

1. NOT AX

NOT DX

ADD AX,1

ADC AX,0

(2)

NEG AX

NEG DX

SBB DX,0

4.5.3 写出完成下列功能的指令序列：（请使用 2 个以上的方法）

（1）将 AL 与 BX 中的两个无符号数相加，结果存入 AX。

ADD BL,AL ADC BH,0 MOV AX,BX

MOV AH,0 ADD AX,BX

（2）将 AL 与 EBX 中的两个带符号数相加，结果存入 EAX。

CBW CWDE ADD EAX,EBX

MOVSX EAX,AL ADD EAX,EBX

\*4.5.4 编写指令序列，实现将 80x86 标志寄存器中的标志位 IF 置 1，DF 变反，CF、PF、

AF、ZF、SF、OF 和 TF 清 0，其它位保持不变。

PUSHF

POP AX

XOR AX,400H

OR AX,200H

AND AX,0F62AH

PUSH AX

POPF

# 第 5 章 分支、循环程序设计

本章要点: 转移指令的寻址方式及其执行过程，控制转移类指令的使用，分支和循环程 序的设计和应用。程序调试的方法，常见问题的程序设计方法。

一、单项选择题

5.1.1 条件转移是根据标志寄存器中的标志位来判断的，条件判断的标志位共有(B)位。

A. 4 B. 5 C. 6 D. 9

5.1.2 用一条条件转移指令一次可以实现(A )个分支

* + - 1. 2 B. 3 C. 4 D. N 5.1.3 条件转移指令的转移范围是（A ）。

A. -128～127 B. 0～255 C. 0～65535 D. -32768～32767

5.1.4 设 A 为字变量，B 为标号，下列指令中不正确的是（D ）。

A. MOV AX，A B. JNZ B C. JMP [SI] D. JMP B[BX] 5.1.5 下述指令中影响 CF 标志位的是（A ）。

A. SHL AL,1 B. MOV AL,1 C. JC L D. JNC L 5.1.6 下述指令中不影响 CF 标志位的是（A ）。

A. INC SI B. SUB SI,0 C. NEG AL D. TEST AL,1

5.1.7 在多重循环程序设计中，每次通过外层循环进入内层循环时，其内层循环的初始 条件（B ）。

A. 不必考虑 B. 必须重新设置 C. 必须清 0 D. 必须置 1 5.1.8 当设计一个程序时，最重要的是（B ）。

A. 程序的结构化 B. 能使程序正常运行并实现功能

C. 程序的执行速度快 D. 程序占用的存储空间小

\*5.1.9 如果“JNC L”指令的操作码放在 0040H，转移后在 0020H 处取下一指令的操作

码，那么这条指令的位移量是（C ）。

A. 1EH B. 20H C. 0DEH D. 0E0H

\*5.1.10 如果“JGE P”指令的操作码放在 0050H，该指令的位移量是 34H，执行完这条 指令转移取下一条指令的偏移地址是（C ）。

A. 82H B. 84H C. 86H D. 88H

二、填空题

5.2.1 当下面循环程序中的划线处填上一个什么数字时，执行的循环次数最多？

MOV CX, 0

MOV AX,0

L: INC AX LOOP L

5.2.2 当两个数进行比较后，执行 JE L 表示两数相等则转移到 L。

5.2.3 循环程序的基本结构主要由 初始化 、循环体 和循环控制 三个部分组成。

5.2.4 循环控制部分的连续两条指令“DEC CX” 和“JNZ L”可以用一条 LOOP L

指令来代替。

5.2.5 分析下面程序段：

ADD AX,BX

JNO L1

JNC L2

SUB AX,BX

JNC L3

JNO L4

JMP SHORT L5

如果 AX 和 BX 的初始值为以下 5 种情况，请问程序段运行后，程序转向哪里？

|  |  |  |
| --- | --- | --- |
| AX | BX | 转向： |
| （1）147BH | 80DCH | L1 |
| （2）B568H | 54B7H | L1 |
| （3）42C8H | 608DH | L2 |
| （4）D023H | 9FD0H | L5 |
| （5）94B7H | B568H | L5 |

5.2.6 指令 MOV AX,A[BX]的源操作数的寻址方式是 相对寻址 ；指令 JMP

DWORD PTR [BX]属于 段间间接 \_寻址。

三、简答题

5.3.1 简述条件转移指令大体上可以分为哪几类？各在什么情况下使用？

（1）单个标志 根据单个标志位转移时

（2）无符号 比较两个无符号数的大小时

（3）有符号 比较两个有符号数的大小时

5.3.2 为了实现程序的多路分支，除了使用条件转移指令，还有哪些多分支的程序设计 方法？

地址表法

跳转表法

逻辑分解法

5.3.3 在循环程序中，循环控制的方法主要有哪几个？各自在什么情况下使用？

计数循环：循环次数知道

条件循环：循环次数未知

四、程序分析题

5.4.1 假设 X 和 X+2 单元的内容是双精度数 p，Y 和 Y+2 单元存放着双精度数 q，其中 X 和 Y 中存放低位字，试分析下面程序段的功能。

当2p>q时，AX=1

2p<=q,AX=2

|  |  |  |
| --- | --- | --- |
|  | MOV | DX,X+2 |
| MOV | AX,X |
| ADD | AX,X |
| ADC | DX,X+2 |
| CMP | DX,Y+2 |
| JL | L2 |
| JG | L1 |
| CMP | AX,Y |
| JBE | L2 |
| L1: | MOV | AX,1 |
|  | JMP | SHORT EXIT |
| L2: | MOV | AX,2 |
| EXIT: | INT | 20H ；INT 20H 为程序结束中断 |
| 当 |  |  |

5.4.2 在下列程序的括号中分别填入如下指令：

|  |  |  |
| --- | --- | --- |
| （1） | LOOP | L20 |
| （2） | LOOPE | L20 |
| （3） | LOOPNE | L20 |

试说明在三中情况下，当程序段执行完后，AX、BX、CX 和 DX 寄存器内容分

别是什么？

MOV AX,1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | AX | BX | CX | DX |
| (1) | 5 | 16 | 0 | 0 |
| (2) | 2 | 4 | 3 | 1 |
| (3) | 3 | 7 | 2 | 0 |

MOV BX,2

MOV DX,3

MOV CX,4

L20： INC AX

ADD BX,AX

SHR DX,1

（ ）

5.4.3 现有程序段如下：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ： |  | N1: | CMP | AL,CL |
| BUF DB  ： | 12H,56H,23H |  | JAE XCHG | N2 AL,CL |
| LEA | SI,BUF | N2: | CMP | BL,CL |
| MOV | AL,[SI] |  | JAE | N3 |
| MOV | BL,[SI+1] |  | XCHG | BL,CL |
| MOV | CL,[SI+2] | N3: | MOV | [SI],AL |
| CMP | AL,BL |  | MOV | [SI+1],BL |
| JAE XCHG | N1 AL,BL |  | MOV  ： | [SI+2],CL |

请回答：（1）该程序完成的功能是 将BUF中三个数从大到小排列 ；

（2）程序运行后，BUF 中的内容依次是 56H,23H,12H 。

5.4.4 现有程序段如下：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| BUF S | DB DB  : MOV | 0D2H  ?  AL,BUF | L: | JZ NEG MOV MOV | L AL S,AL  AH,4CH |
|  | TEST | AL,80H |  | INT | 21H |

请回答：（1）该程序完成的功能是 求BUF的绝对值 ；

（2）程序运行后，S 中的内容为 1DH 。

5.4.5 现有程序段如下：

：

NUM DB 30H,54H,07H,11H,68H,98H,8BH,0,56H SUM DW ?

：

MOV SI,OFFSET NUM MOV AX,0

NEXT: CMP [SI],BYTE PTR 0 JZ NDO

ADD AL,[SI]

ADC AH,0

INC SI

JMP NEXT NDO: MOV SUM,AX

请回答：（1）该程序完成的功能是 求NUM 中 0 前面的数的和 ；

（2）如果删除程序中的指令“ADC AH，0”，则程序的运行结果如何？

丢失进位，27H 。

5.4.6 现有程序段如下：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ： |  |  |  | ADD | BX,AX |
| STR0 | DB | 'ZERO',13,10,'$' |  | JMP | WORD PTR [BX] |
| STR1 | DB | 'ONE',13,10,'$' | D0: | LEA | DX,STR0 |
| STR2 | DB | 'TWO',13,10,'$' |  | JMP | EXIT |
| STR3 | DB | 'THREE',13,10,'$' | D1: | LEA | DX,STR1 |
| X | DB | 02H |  | JMP | EXIT |
| TAB | DW  ： | D0,D1,D2,D3 | D2: | LEA JMP | DX,STR2 EXIT |
|  | LEA | BX,TAB | D3: | LEA | DX,STR3 |
|  | MOV | AL,X | EXIT: | MOV | AH,9H |
|  | AND | AL,3 ;\* |  | INT | 21H |
|  | MOV | AH,0 |  | MOV | AH,4CH |
|  | SHL | AX,1 |  | INT | 21H |

请回答：（1）该程序完成的功能是\_

根据X的值显示不同字符串 ；

（2）程序运行后显示输出的是什么？ TWO ； （3）程序中指令 “AND AL,3”的作用是什么？ 将数限制在0~3 。

5.4.7 现有程序段如下：

:

|  |  |  |
| --- | --- | --- |
| ARRAY | DW | 1234H,5678H,9ABCH,-90,0,-234,1234,89,57ADH,0BC3H |
| LEN | EQU | $-ARRAY |
| MAX | DW | ? |
| MIN | DW | ? |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | : |  |  | JNL | NEXT1 |
| LEA | BX,ARRAY |  | MOV | MAX,AX |
| MOV | MAX,-32768 | NEXT1: | CMP | MIN,AX |
| MOV | MIN,32767 |  | JNG | NEXT2 |
| MOV | CX,LEN |  | MOV | MIN,AX |
| SHR | CX,1 | NEXT2: | ADD | BX,2 |
| AGAIN: | MOV | AX,[BX] |  | LOOP | AGAIN |
|  | CMP | MAX,AX |  | : |  |

请回答：（1）该程序完成的功能是 找出ARRAY数组的最值 ；

（2）程序运行后，MAX 和 MIN 单元中的内容分别为为 57AD H 和 9ABC H。

5.4.8 现有程序段如下：

：

BUF DW 18 DUP(?)

：

LEA SI,BUF MOV CX,18 MOV AX,0

MOV BX,1

LOP: MOV [SI],AX

MOV DX,AX ADD AX,BX MOV BX,DX ADD SI,2 LOOP LOP

：

请回答：（1）该程序完成的功能是 在BUF中产生FIBO数列 ；

（2）程序运行后，BUF 中的前 10 个数的内容为 0,1,1,2,3,5,8,13,21,34 。

五、程序填空题

5.5.1 下面的程序段是判断两个无符号字数据 X 和 Y 的大小，当 X>Y 时计算 X-Y，当 X<Y 时计算 Y-X，当 X=Y 时计算 X+Y，运算后的结果存入字变量 W 中。请在程序的 空格处填写适当的指令。

MOV AX,X

MOV BX,Y

CMP AX,BX

JA L ;(1) JB M

ADD AX,BX

JMP DOWN ;(2)

L: SUB AX,BX ;(3) JMP DOWN

M: XCHG AX,BX SUB AX,BX

DOWN: MOV W,AX

MOV AH,4CH INT 21H

5.5.2 下面的程序是将 BUF1 缓冲区的有符号字节数据中小于 100 的数送 BUF2 自己字

节缓冲区中保存。请在程序的空格处填写适当的指令。

：

|  |  |  |
| --- | --- | --- |
| BUF1 COUNT BUF2 | DB EQU DB | 12H,56H,9DH,0D7H,0,45H,78H,89H,0F5H,60H  $-BUF1 COUNT DUP(?) |
|  | ： |  |
|  | MOV | SI,OFFSET BUF1 |
| MOV DI,OFFSET BUF2 ;(1) | | |
|  | MOV | CX,COUNT |
| L: | MOV | AL,[SI] |
|  | CMP | AL,100 |
| JGE NEXT ;(2) | | |

MOV [DI],AL

INC DI

NEXT: INC SI LOOP L

：

;(3)

5.5.3 下面的程序是是统计以 0 为结尾的 STR 字符串中大写字母、小写字母、数字和非 数字字母的个数，分别送 A、B、C、D 字节单元保存。请在程序的空格处填写适当的指 令。

：

;(3)

|  |  |  |
| --- | --- | --- |
| STR | DB | '12+34/QWeRas?.Df"56UiOPj..kA[,%78ZcB&M=huzl9',0 |
| A | DB | ? |
| B | DB | ? |
| C | DB | ? |
| D | DB  ： | ?  JA NEXT2 |
| LEA AGAIN:MOV  CMP  CMP JB CMP JA INC  NEXT1:CMP  JB | | SI,STR INC A  AL,[SI] JMP DOWN  AL,0 NEXT2:MOV AL,[SI]  JE EXIT;(1) CMP AL,61H;( AL,30H JB OTHER  OTHER CMP AL,7AH  AL,39H JA OTHER  NEXT1 INC B  C JMP DOWN  JMP DOWN2) OTHER:INC D  AL,41H DOWN: INC SI OTHER JMP AGAIN |

;(4)

CMP AL,5AH EXIT: ：

5.5.4 下面程序段的功能是统计字节变量 X 中为 1 的位数，并将统计结果显示出来。请 在程序的空格处填写适当的指令。

|  |  |  |  |
| --- | --- | --- | --- |
| X | DB | 5EH | |
|  | : |  | |
|  | MOV | AL,X | |
|  | MOV | DL,0 | |
|  | MOV | CX,8 | |
| L1: | SHL | AL,1 | |
|  | JNC L2 | | ;(1) |
|  | INC DL | |  |
| L2: | CMP AL ,0 | | ;(2) |
|  | JNZ L1 | |  |
|  | ADD DL,30H | | ;(3) |

MOV AH,2

INT 21H

六、编写程序题（编写完整的汇编语言程序，要求各段定义完整，能正常返回 DOS，

经汇编、连接后能正常运行。根据自己的情况自行编程练习，并且使用上机手段验证程 序是否实现规定的功能。）

5.6.1 编写汇编语言程序，要求对键盘输入的小写字母用大写字母显示出来，注意判断 输入小写字母的有效性。

data segment

mess db:'inpup error!',0ah,0dh,'$'

data ends

code segment

assume ds:data,cs:code

start:mov ax,data

mov ds,ax

mov ah,01h

int 21h

cmp al,'A'

jb err

cmp al,'z'

ja err

add al,20h

mov dl,al

mov ah,02h

int 21h

jmp exit

err:mov dx,offset mess

mov ah,09h

int 21h

exit:mov ah,02h

int 21h

code ends

end start

5.6.2 编写汇编语言程序，从键盘接收小写字母，然后找出它的前导和后继字符，再按 顺序显示这三个字符，注意字母 a 的前导是 z，z 的后继字母是 a。

data segment

n db 0

data ends

code segment

assume cs:code,ds:data

p proc far

mov ax,data

mov ds,ax

mov ah,01h

int 21h

cmp al,'a'

jb input

je behind

cmp al,'z'

ja input

je blow

jmp normal

input: xchg dl,al

mov ah,02h

int 21h

jmp endd

normal : xchg dl,al

dec dl

mov ah,02h

int 21h

inc dl

mov ah,02h

int 21h

inc dl

mov ah,02h

int 21h

jmp endd

behind: xchg dl,al

mov ah,02h

int 21h

inc dl

mov ah,02h

int 21h

jmp endd

blow: xchg dl,al

dec dl

mov ah,02h

int 21h

inc dl

mov ah,02h

int 21h

jmp endd

endd:

mov ah,4ch

int 21h

p endp

code ends

end p

5.6.3 编一汇编语言程序，将一个包含有 20 个数据的数组 M 分成两个数组：正数组 P 和负数组 N，并分别把这两个数组中数据的个数显示出来（可以按十六进制数显示）。

DATA SEGMENT

M    DB 5 DUP(-1,-2,-1,2)

P    DW 0

N    DW 0

P1   DB "P=$"

N1   DB 0AH,"N=$"

DATA ENDS

STACK SEGMENT PARA STACK

          DW 128 DUP(?)

TOP   LABEL WORD

STACK ENDS

CODE SEGMENT

           ASSUME CS:CODE,SS:STACK,DS:DATA

MAIN PROC FAR

START:

     MOV AX,DATA

     MOV DS,AX

     MOV AX,STACK

     MOV SS,AX

     LEA SP,TOP

     LEA SI,M

     MOV CX,20

L0: MOV AL,[SI]

     CMP AL,0

     JL  L1

     JG  L2

L1: INC N

     JMP L3

L2: INC P

     JMP L3

L3: INC SI

     LOOP L0

     MOV AH,9

     LEA DX,P1    ;

     INT 21H

     MOV AX,P     ;

     MOV BL,10

     DIV BL

     MOV DL,AL     ;

     ADD DL,30H    ;

     MOV BH,AH     ;显示P商

     MOV AH,2

     INT 21H

     MOV DL,BH     ;显示P余数

     ADD DL,30H

     MOV AH,2

     INT 21H

     MOV AH,9

     LEA DX,N1

     INT 21H

     MOV AX,N

     MOV BL,10

     DIV BL

     MOV DL,AL     ;显示N商

     ADD DL,30H

     MOV BH,AH

     MOV AH,2

     INT 21H

     MOV DL,BH     ;显示N余数

     ADD DL,30H

     MOV AH,2

     INT 21H

     MOV AH,4CH

     INT 21H

CODE ENDS

MAIN ENDP

      END START

5.6.4 试编写一个汇编语言程序，找出首地址为 DATE 的 100 个字数组中的最小偶数， 把它存放在 AX 寄存器中。（选做：并把该最小偶数显示出来）

DATAS SEGMENT;

ARRAY DW 4,5,6,7,8,9,3,6,1,10,34,12,45,58,6,32,15,17,20,'$'

Z DB 100,10

DATAS ENDS

STACKS SEGMENT;此处输入堆栈段代码

DW 100 DUP(?)

STACKS ENDS

CODES SEGMENT

ASSUME CS:CODES,DS:DATAS,SS:STACKS

START:

MOV AX,DATAS

MOV DS,AX ;

LEA BX,ARRAY

MOV DX,[BX]

MOV AX,4 ;AX初始值设为4

TEST1: MOV CX,DX ;判断是否是偶数

RCR CX,1 ;带进位循环右移一位

JNC OUSHU ;偶数跳转

JC CONTINUE ;奇数跳转

OUSHU:

CMP AX,DX

JL CONTINUE ;如果小于AX,则将AX和CX的值交换，比较完之后AX中始终是最小的偶数

JNL JIAOHUAN ;较下一个数

JIAOHUAN:MOV AX,DX

JMP CONTINUE

CONTINUE:ADD BX,2

MOV DX,[BX]

CMP DX,24H ;与结束符$比较，看是否结束

JNZ TEST1 ;不相等跳转到比较处

JMP SHUCHU

SHUCHU:

DIV Z

PUSH AX

CMP AL,0

JZ AA

ADD AL,30H

MOV DL,AL

MOV AH,02H

INT 21H

AA:POP AX

MOV AL,0

ADD AH,30H

MOV DL,AH

MOV AH,02H

INT 21H

MOV AH,4CH

INT 21H

CODES ENDS

END START

5.6.5 编一汇编语言程序，从键盘输入一系列字符（小于 80 个，以回车符结束），按字 母、数字以及其他字符分类统计，并显示（可以按十六进制数显示）出这三类的计数结 果。（选做：按十进制数显示计数结果）

见实验报告三

5.6.6 试编写一个汇编语言程序，要求比较两个字符串 STRING1 和 STRING2 所含字符 是否相同，若相同则显示“MATCH”，若不相同则显示“NO MATCH”。（提示：显示信息 用 9 号功能调用；比较字符串可以用串操作指令，也可以逐个比较。）

code ends

end maindata segment

str1 db '123456','$'

len1 equ $-str1

str2 db '123456','$'

len2 equ $-str2

match db 'MATCH!','$'

nomatch db 'NO MATCH!','$'

data ends

stack segment stack

dw 20h dup(?)

top label word

stack ends

code segment

assume cs:code,ds:data,ss:stack

main proc far

mov ax,data

mov ds,ax

mov es,ax ;this is important

mov ax,stack

mov ss,ax

lea sp,top

mov ah,len1

mov al,len2

cmp al,ah

jne no

mov ch,0

mov cl,al

lea si,str1

lea di,str2

cld

repe cmpsb

jne no

yes:

lea dx,match

mov ah,09h

int 21h

jmp exit

no: lea dx,nomatch

mov ah,09h

int 21h

exit:

mov ah,4ch

int 21h

main endp

5.6.7 试编写一个汇编语言程序，要求从键盘接收一个四位的十六进制数，并在屏幕上 显示出与它等值的二进制数。注意判断输入的有效性。

DATA SEGMENT

STR1 DB 'INPUT A HEX NUM: $'

HEX DB 5 ;计划输入的字符个数(含回车符)

NNN DB 0 ;存放实际输入的字符个数

DB 5 DUP(?)

BIN DW ? ;输入数字的二进制形式的存放单元

ERR DB 0AH, 0DH, ' ERROR !'

DB 0AH, 0DH, ' I HAVE NO TIME TO PLAY WITH YOU ! $'

STR2 DB 13, 10, 'BINARY IS: $'

DATA ENDS

STACK SEGMENT PARA STACK 'STACK'

DB 100 DUP(?)

STACK ENDS

CODE SEGMENT

ASSUME CS:CODE, SS:STACK, DS:DATA

START:

MOV AX, DATA

MOV DS, AX

MOV AX, STACK

MOV SS, AX

LEA DX, STR1 ;'INPUT A HEX NUM'

MOV AH, 9 ;显示字符串

INT 21H

LEA DX, HEX

MOV AH, 10 ;等着输入四位HEX数

INT 21H

MOV BIN, 0 ;先把二进制数字单元清零

LEA SI, HEX + 2;取输入字符串的首地址

MOV CH, NNN ;取输入字符的个数，用来控制循环处理的次数

CONV:

MOV BL, [SI] ;取来一个字符

CMP BL, 0DH ;和回车符进行比较

JZ END0 ;是, 即结束

CMP BL, '0'

JB ERROR ;小于'0'转移

CMP BL, 'F'

JA ERROR ;大于'F'转移

CMP BL, '9'

JNA A2B ;不大于'9', 就去变换

CMP BL, 'A'

JNB A2B ;不小于'A', 就去变换

JMP ERROR ;其它符号都是错误

A2B:

SUB BL, 30H ;ASC-->BIN

CMP BL, 10

JB A2B\_END

SUB BL, 07H ;ASC-->BIN

A2B\_END:

MOV BH, 0 ;弄成16位数

PUSH CX

MOV AX, BIN ;把原来的数字乘以16

MOV CX, 16

MUL CX

ADD AX, BX ;加上新输入的

MOV BIN, AX ;存结果，这就是二进制数

INC SI

POP CX

DEC CH ;处理次数减一

JNZ CONV ;非零转移

DISP:

LEA DX, STR2 ;'BINARY IS: $'

MOV AH, 9 ;显示字符串

INT 21H

MOV AX, BIN ;取来数字

MOV CX, 16 ;显示16位数

DDDD:

MOV DL, '0' ;先计划显示0

TEST AX, 8000H ;测试最高位

JZ DDD1

INC DL ;非0就显示1

DDD1:

PUSH AX

MOV AH, 2 ;显示DL中的ASCII码

INT 21H

POP AX

SHL AX, 1 ;左移

LOOP DDDD ;CX-1，非零转移

JMP END0 ;16位显示完毕，转到结束

ERROR:

LEA DX, ERR

MOV AH, 09H ;错误信息提示

INT 21H

END0:

MOV DL, 10

MOV AH, 2 ;显示DL中的ASCII码

INT 21H

MOV DL, 13

MOV AH, 2 ;显示DL中的ASCII码

INT 21H

MOV AX, BIN ;取来数字

CALL DISP\_AX

MOV AH, 4CH ;返回DOS

INT 21H

DISP\_AX:

MOV BX, 10 ;10进制.

MOV CX, 5 ;5位.

ABC:

MOV DX, 0

DIV BX

PUSH DX ;保存余数

LOOP ABC

MOV CX, 5 ;显示5位.

ASD:

POP DX

CMP DL, 10

JB A48

ADD DL, 7

A48:

ADD DL, 48

MOV AH, 2

INT 21H ;输出.

LOOP ASD

RET

CODE ENDS

END START

# 第 6 章 子程序设计

本章要点: 子程序的定义、调用和返回，寄存器的保护盒恢复，参数传递方法尤其是堆 栈传递参数，模块化程序设计的基本方法，DOS 功能调用。

一、单项选择题

6.1.1 下列叙述不正确的是（A ）。

A. 在子程序中的保护现场只能用堆栈来实现

B. 在子程序中的保护现场用堆栈来实现是其中的一种方法

C. 在子程序中的保护现场可以有多种实现方法

D. 在子程序中的保护现场可以将要保护的内容送内存变量来实现

6.1.2 下列叙述不正确的是（B ）。

A. 在汇编语言程序中，每一个过程允许有多条 RET 指令

B. 在汇编语言程序中，每一个过程只允许出现一条 RET 指令

C. 在汇编语言程序中，每一个过程结束之前一定有一条 RET 指令

D. 在汇编语言程序中，以过程形式表示的代码段一定有一条 RET 指令存在

6.1.3 下列叙述正确的是（B ）。

A. 执行一条段内返回指令，先从堆栈弹出两个字节的内容，然后 SP-2。 B. 执行一条段内返回指令，先从堆栈弹出两个字节的内容，然后 SP+2。

C. 执行一条段内返回指令，先从 SP-2，然后从堆栈弹出两个字节的内容。 D. 执行一条段内返回指令，先从 SP+2，然后从堆栈弹出两个字节的内容。

6.1.4 在进行 DOS 功能调用时，其功能号应先送（C ）。

A. AL 寄存器 B. BL 寄存器 C.AH 寄存器 D. DH 寄存器

二、填空题

6.2.1 在子程序的设计中，通常用堆栈来保护现场和恢复现场。而堆栈的操作原则是

先进后出 。

6.2.2 如果主程序和子程序在同一代码段中，则主程序调用子程序时只改变 偏移 地址；如果主程序和子程序不在同一代码段中，则主程序调用子程序时要改变偏移 地址和段 地址。

6.2.3 一个子程序调用另一个子程序称为 子程序的嵌套 ；一个子程序直接或间接 调用该子程序本身称为 递归 。

6.2.4 以过程定义的子程序有两种类型的属性，它们分别是 NEAR 和FAR 。

6.2.5 在用 9 号 DOS 功能调用进行字符串输出时，要求输出字符串以 $ 为结束符。

三、简答题

6.3.1 简述一个完整的子程序结构应该包含哪几个方面的内容？

子程序定义，保存恢复现场，子程序主体，RET

6.3.2 调用程序和子程序之间一般使用哪几种参数传递方法？它们各自的特点是什么？

利用寄存器传递参数：最直接简便常用，但是只适合传递较少的参数

利用存储器传递参数：可以建立一个由参数地址构成的地址表以便传输

利用堆栈传递参数：适用于参数较多，或子程序有多层嵌套，递归调用的情况

四、程序分析题

6.4.1 现有程序如下：

MOV DL,0AH END START

|  |  |  |  |
| --- | --- | --- | --- |
| DATA SEGMENT |  | INT | 21H |
| BUF DB 80 DUP(0) |  | MOV | BYTE PTR [DI],'$' |
| DATA ENDS |  | LEA | DX,BUF |
| CODE SEGMENT |  | MOV | AH,9 |
| ASSUME CS:CODE,DS:DATA |  | INT | 21H |
| START: MOV AX,DATA |  | MOV | AH,4CH |
| MOV DS,AX |  | INT | 21H |
| LEA DI,BUF | STO | PROC |  |
| NEXT: MOV AH,1 |  | CMP | AL,30H |
| INT 21H |  | JB | DOWN |
| CMP AL,0DH |  | CMP | AL,39H |
| JZ EXIT |  | JA | DOWN |
| CALL STO |  | MOV | [DI],AL |
| JMP NEXT |  | INC | DI |
| EXIT: MOV AH,2  MOV DL,0DH INT 21H | DOWN: RET STO ENDP CODE ENDS | | |

请回答：（1）STO 子程序完成的功能是 判断从键盘输入的是否为 0~9 的数字字符

（2）该程序完成的功能是输入以回车结束的字符串，将数字字符存入BUF,并显示这些数字字符 。

6.4.2 现有程序如下：

STACK SEGMENT STACK 'STACK' DB 64 DUP(?)

STACK ENDS DATA SEGMENT A DB 30

1. DB 9
2. DW 5

DATA ENDS CODE SEGMENT

ASSUME CS:CODE,DS:DATA,SS:STACK MAIN PROC FAR

PUSH DS MOV AX,0 PUSH AX

JNZ L2

POP DX

POP AX RET

MOV AX,DATA MOV DS,AX MOV CX,C

MOV BH,B

MOV BL,A

L1: PUSH BX CALL SUB1 CALL SUB2 POP BX

INC BL

SUB BH,2 LOOP L1 RET

MAIN ENDP SUB1 PROC

PUSH AX PUSH DX

L2: MOV DL,20H MOV AH,2

INT 21H

DEC BL

SUB1 ENDP SUB2 PROC

PUSH AX PUSH DX

L3: MOV DL,'\*' MOV AH,2

INT 21H

DEC BH

JNZ L3 MOV DL,0DH MOV AH,2

INT 21H MOV DL,0AH INT 21H

POP DX

POP AX RET

SUB2 ENDP CODE ENDS

END MAIN

请回答：（1）SUB1 子程序完成的功能是 打印BL寄存器所表示的空格数 ；

（2）SUB2 子程序完成的功能是 打印BH寄存器所表示的‘\*’ ；

（3）该程序完成的功能是 打印倒三角图形，共5行，第一行9个‘\*’，以后逐行减二 。

五、程序填空题

6.5.1 下面的程序是将 BUF1 缓冲区的 100 个字单元的内容送 BUF2 缓冲区的子程序。

请在程序的空格处填写适当的指令。

MOVE PROC

PUSH AX

ADD SI, 2 ;(3) ADD DI,2

PUSH SI LOOP L ;(4)

PUSH DI ;(1) PUSH CX

POP CX

POP DI

MOV CX，100 POP SI ;(5)

LEA SI,BUF1

LEA DI,BUF2 L: MOV AX,[SI]

MOV [DI], AX ;(2)

POP AX RET

MOVE ENDP

6.5.3 下面的程序是将 AX 寄存器中的 16 位无符号二进制数转换为十进制数显示输出的 子程序。请在程序的空格处填写适当的指令。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| DISP | PROC |  | DISP0: | CMP CX,0 |
|  | PUSH AX |  |  | JZ RE |
|  | PUSH BX |  |  | POP DX |
|  | PUCH CX |  |  | ADD DX, 30H ;(3) |
|  | PUSH DX |  |  | MOV AH,2 |
|  | MOV CX,0 |  |  | INT 21H |
|  | MOV BX,10 |  |  | DEC CX |
| L: | MOV DX, 0 | ;(1) |  | JMP DISP0 ;(4) |
|  | CMP AX,0 |  | RE: | POP DX |
|  | JZ DISP0 |  |  | POP CX |
|  | DIV BX |  |  | POP BX |
|  | PUSH DX | ;(2) |  | POP AX |
|  | INC CX |  |  | RET ;(5) |
|  | JMP L |  | DISP | ENDP |

6.5.2 下面的程序通过子程序调用完成在 2 个数组中找出最大数，并将最大数存放在各 自数组的后一个字单元。主子程序之间采用堆栈传递参数方式。请在程序的空格处填写 适当的指令。

STACK SEGMENT STACK 'STACK' DB 64 DUP(?)

STACK ENDS DATA SEGMENT

BUF1 DW 2001H,45E5H,1234H,678AH,905DH,08F3H N1 EQU ($-BUF1)/2

MAX1 DW ?

BUF2 DW 5678H,5E32H,3412H,8086H,0E234H,6635H,7329H N2 EQU ($-BUF2)/2

MAX2 DW ? DATA ENDS CODE SEGMENT

ASSUME CS:CODE,DS:DATA,SS:STACK MAIN PROC FAR

PUSH DS MOV AX,0

PUSH SI

PUSHF ;(2) MOV SI,[BP+6]

PUSH AX

MOV AX,DATA MOV DS,AX LEA AX,BUF1 PUSH AX

MOV AX,N1 PUSH AX CALL SMAX LEA AX,BUF2

MOV CX, [BP+4] MOV AX,[SI]

DEC CX NEXT: ADD SI,2

CMP AX,[SI]

JGE MAX MOV AX,[SI]

MAX: LOOP NEXT

ADD SI,2

;(3)

PUSH AX MOV [SI], AX ;(4)

MOV AX,N2 PUSH AX CALL SMAX RET

MAIN ENDP

POPF POP SI

POP CX

POP AX

POP BP

SMAX PROC RET 4H ;(5)

PUSH BP

MOV BP, SP (1)

PUSH AX

PUSH CX

SMAX ENDP CODE ENDS

END MAIN

六、程序设计题

6.6.1 编写子程序 DISPBX，能将 BX 寄存器中的 16 位二进制数转换为十六进制数在屏 幕上显示输出。

PUBLIC DISPBX

CODE SEGMENT

ASSUME CS:CODE

DISPBX PROC FAR

MOV CH,4

MOV CL,4

ROTATE: ROL BX,CL

MOV AL,BL

AND AL,0FH

ADD AL,30H

CMP AL,3AH

JL PRINTIT

ADD AL,7

PRINTIT:MOV DL,AL

MOV AH,2

INT 21H

DEC CH

JNZ ROTATE

RET

DISPBX ENDP

CODE ENDS

END

# 第 7 章 输入/输出程序设计

本章要点: CPU 与外设交换数据的 3 种方法，查询传送方式，中断传送方式，中断向量 表的使用，以及中断程序设计的基本方法，直接使用 IN/OUT 指令进行 I/O 程序设计。 一、单项选择题

7.1.1 执行“INT 10H”指令时，其中断处理程序的入口地址在（D ）。

A. 10H～13H B. 20H～23H C. 30H～33H D. 40H～43H

7.1.2 执行“INT 10H”指令时，入栈的内容有（D ）。

A. CS、IP B. PSW、IP C. PSW、CS D. PSW、CS、IP

三、简答题

7.3.1 写出分配给 INT 8 的中断向量在中断向量表中的物理地址。20H~23H

7.3.2 中断类型号为 14H 的中断向量在存储器的哪些单元中？ 0:50H~0:53H

7.3.3 写出指令将一个字节数据输出到端口 25H。 OUT 25H, AL

7.3.4 写出指令将一个字数据从端口 1000H 输入。 MOV DX, 1000H IN AX, DX