汇编复习题整理

1、80X86 微型计算机的组织

1-10　写出下列十进制数的十六进制表示。

　18　34　87　255　4095　62472

解：18=**12H**；34=**22H**；87=**57H**；

　　255=**0FFH**；4095=**0FFFH**；　62472=**0F408H**;

1-12　处理器的主要功能是实现所有指令的执行并处理数据。

1-14 指出处理器是如何存储的内存中的一个值

(a) hex 1234;　　(b) hex 01c3b5

解：（a）高位地址　　12H

　　低位地址　　34H

　　地址　　内存

　（b）高位地址　　01H

　　0c3H

　　地位地址　　0b5H

　　地址　　内存

1-15:　说出段，偏移量，地址边界的定义:

解：

段：一个程序段边界上开始的部分，大小为 64K，包含代码，数据，堆栈。

偏移量：是段边界上开始到段中其他位置的字节距离。

段边界：可以被 16 整除的位置;

1-18:　指出下列操作所需的寄存器:

(a) 段寻址; (b) 要执行的指令的偏移地址;

(c) 加法和减法操作;

(d) 乘法操作;

(d)乘法和除法操作;

(e)循环计数;

(f)标示结果为零。

解：(a)　CS,DS,SS;

　　(b)　IP;

　　(c)　AX,BX,CX,DX,DI,SI;

　　(d)　DX, AX;

　　(e)　CX;

　　(f)　ZF

二、单选

1.　　十六进制数 88H 可以被其他形式的数所表示, 下列哪种表示方法是错的?D

A. 无符号十进制 136　　C.压缩的 BCD 码 88

B. 带符号十进制–120　　D. 带符号数据 **–8**

解：错误的是 D 选项，A 选项 88=8*16^1+8=136，正确，B 选项 带符号的十六进制转换十进制方法：先将十六进制用二进制形式表示 88H=10001000B，然后按位取反，末位加 1 得 01111000，再转换为十进制得-120，正确；C 选项 正确。

2. 如果 DH=10H, 执行 NEG DH 指令, 正确的结果是(**D**.).

A. DH=10H　C=1　　C. DH=10H　C=0

B. DH=0F0H　C=0　　D. DH=0F0H　C=1

3.哪个是指令指针寄存器? A

A.IP　B.SP　C.BP　D.PSW(program status word)

4. 如果 AX=1000H,

NEG AX

NEG AX

上面两条指令执行后, AX= (C).

a.1001H　b.1002H　c.1000H　d.0F000H

解：指令 NEG 是取反的意思，两次取反自然为本身。故选 C

2.2 有两个 16 位的字 1EE5H 和 2A3CH 存储在 IBM PC 的 000B0H 和 000B03H 的内存单元中，请以图表的形式说明内存中的存储情况。

解：

内存地址	内容
000B4H	2AH
000B3H	3CH
000B2H	
000B1H	1EH
000B0H	E5H

2.3 如下图, 展示的是 IBM PC 的内存信息, 请说明 30022H 字节单元和 30024H 字节单元的内容,和 30021H 字单元和 30022H 字单元的内容。

解：

存储器

30020H	12H	30022H 字节单元的内容=0ABH
30021H	34H	30024H 字节单元的内容=0EFH
30022H	ABH	30021H 字单元的内容=0AB34H
30023H	CDH	30022H 字单元的内容=0CDABH
30024H	EFH	

2.4 3017:000A 的段地址的物理地址和偏移量是什么? 3015:002A 和 3010:007A 的段地址和偏移是什么？

解：段地址的物理地址是：3017AH；偏移量是：000AH

物理地址 1：PA= 3017 × 10H + 000AH = 3017AH

物理地址 2: PA= 3015 × 10H + 002AH = 3017AH

物理地址 3: PA= 3010 × 10H + 007AH = 3017AH

2.5 运行程序之前, (CS)=0A7F0H,(IP)=2B40H, 程序的第一个字的物理地址是什么?

解：

PA=（CS）×10H+（IP）= 0A7F00H + 2B40H = 0AAA40H

2.8 哪类型的寄存器可以用来显示内存地址?

解：CS, DS, ES, SS, IP, SP, BP, BX, DI, SI,

EAX, EBX, ECX, EDX, EBP, EIP, ESP, EDI,

5. 假定 DS = 5788H, 偏移量是 94H, 字节的 PA(物理地址) (A).

a.57974H b.57914H

c. 5883H d. 58ECH

解： PA=5788H*10H+94H=57974H

6.在段中寻址的时候，在 8086CPU 的寄存器中，哪些寄存器可以提供偏移地址(B)

A. AX, BX, CX, DX C. SP, IP, BP, DX

B. BX, BP, SI, DI D. CS, DS, ES, SS

概念总结：

CPU－-分析，控制并执行指令的部件，由算术逻辑部件 **ALU** 和寄存器组等组成。

存储器－－存储程序，数据等信息的记忆装置，**PC** 机有 **RAM** 和 **ROM** 两种。

堆栈－－以后进先出方式工作的存储空间。

IP－－指示 下一条要执行指令的地址。

SP－－保存 当前栈顶地址的寄存器。

状态标志－－记录指令操作结果的标志，共 **6** 位**;OF,SF,ZF,AF,PF,CF.**

控制标志－－控制操作的标志，**P** 机有三位，**DF,IF,TF.**

段寄存器－－保存各逻辑段的起始地址的寄存器，**PC** 机有 **4** 个：**CS,DS,SS,ES.**

物理地址－－唯一代表存储空间中每个字节单元的地址。

汇编语言－－用指令的助记符，符号地址，标号等符号书写程序的语言。

机器语言－－能被计算机直接识别执行和的语言。

汇编程序－－把汇编语言程序翻译成机器语言程序的系统程序。

连接程序－－把若干个模块连接起来成为可执行文件的系统程序。

指令－－告诉 **CPU** 要执行的操作（一般还要指出操作数的地址），在程序运行时执行。

伪指令－－由汇编程序在汇编过程中执行的指令。

3. 调试

3-7 用 DEBUG 的 E 命令来进入下列的机器语言程序:

机器码(在 00H 中):A0 00 D0 E0 F6 26 01 02 A3 02 02 90

数据(在 200H 中):2A 12 00 00
程序的实现的功能如下:
把 DS:0200(2A)中的一个字节的内容移到 AL 寄存器。
把 Al 中的一个位移动到左边(结果是 54。)
把在 DS:0201 中的一字节内容与 AL 相乘。
把结果从 AX 中移到 DS:0202 的开始的字;
键入程序后, 键入 D 命令来查看代码和数据。
键入 R 和连续的 T 命令来一步步运行程序直到到达 NOP。
这时, AX 应该包含了结果 05E8H。
键入另一个 D DS:0200,且注意到在 DS:0202 中的结果是 E805。
调试:

```
-E CS:100    A0 00 02 D0 E0 F6 26 01 02 A3 02 02 90                    ;输入 E 指令
-D CS:100 LC
1370:0100    A0 00 D0 E0 F6 26 01 02-A3 02 02 90                       .....&.....
-E DS:200 2A 12 00 00
-D DS:200 L4
1370:0200    2A 12 00 00
-R
AX=0000    BX=0000    CX=0000    DX=0000    SP=FFEE    BP=0000    SI=0000    DI=0000
DS=0B41    ES=0B41    SS=0B41    CS=0B41    IP=0100     NV UP EI PL NZ NA PO NC
0B41:0100 A00002            MOV        AL,[0200]                          DS:0200=2A
-T
AX=002A    BX=0000    CX=0000    DX=0000    SP=FFEE    BP=0000    SI=0000    DI=0000
DS=0B41    ES=0B41    SS=0B41    CS=0B41    IP=0103     NV UP EI PL NZ NA PO NC
0B41:0103 D0E0              SHL        AL,1
-T
AX=0054    BX=0000    CX=0000    DX=0000    SP=FFEE    BP=0000    SI=0000    DI=0000
DS=0B41    ES=0B41    SS=0B41    CS=0B41    IP=0105     NV UP EI PL NZ NA PO NC
0B41:0105 F6260102          MUL        BYTE PTR [0201]                    DS:0201=12
-0B41:0105 F6260102          MUL        BYTE PTR [0201]                   DS:0201=12
-T
AX=05E8    BX=0000    CX=0000    DX=0000    SP=FFEE    BP=0000    SI=0000    DI=0000
DS=0B41    ES=0B41    SS=0B41    CS=0B41    IP=0109     OV UP EI PL NZ NA PO CY
0B41:0109    A30202         MOV        [0202],AX                          DS:0202=0000
-T
AX=05E8    BX=0000    CX=0000    DX=0000    SP=FFEE    BP=0000    SI=0000    DI=0000
DS=0B41    ES=0B41    SS=0B41    CS=0B41    IP=010C     OV UP EI PL NZ NA PO CY
0B41:010C 90                NOP
-D DS:202    L2
0B41:0200             E8 05
```

3-8 问题 3-7, 键入命令进入写在硬盘上的程序 HEXMULT.COM.
调试:

```
-U 100 10C
0B41:0100 A00002          MOV       AL,[0200]
0B41:0103 D0E0            SHL       AL,1
0B41:0105 F6260102        MUL       BYTE PTR [0201]
0B41:0109 A30202          MOV       [0202],AX
0B41:010C 90              NOP
-N HEXMULT.COM
-R BX
BX 0000
:0
-R CX
CX 0000
:000B
-W
Writing 0000A bytes
```

3-9. 用 DEBUG 的 A 命令来进入下列指令:

```
MOV CX,3B
ADD CX,1C
SHL CX,01
SUB CX,36
NOP
```

反汇编这些指令跟踪指令的运行直到 NOP. 并且对每条指令，都查看 CX 寄存器的值。

调试：

```
-A
0B41:0100 MOV CX,3B
0B41:0103 ADD CX,1C
0B41:0106 SHL CX,1
0B41:0108 SUB CX,36
0B41:010B NOP
-U 100 10B
0B41:0100 B93B00          MOV       CX,003B
0B41:0103 83C11C          ADD       CX,+1C
0B41:0106 D1E1            SHL       CX,1
0B41:0108 83E936          SUB       CX,+36
0B41:010B 90              NOP
-T
AX=0000  BX=0000  CX=003B  DX=0000  SP=FFEE  BP=0000  SI=0000  DI=0000
DS=0B41  ES=0B41  SS=0B41  CS=0B41  IP=0103     NV UP EI PL NZ NA PO NC
0B41:0103 83C11C          ADD       CX,+1C
-T
AX=0000  BX=0000  CX=0057  DX=0000  SP=FFEE  BP=0000  SI=0000  DI=0000
DS=0B41  ES=0B41  SS=0B41  CS=0B41  IP=0106     NV UP EI PL NZ AC PO NC
```

```
0B41:0106 D1E1              SHL         CX,1
-T
AX=0000   BX=0000   CX=00AE   DX=0000   SP=FFEE   BP=0000   SI=0000   DI=0000
DS=0B41   ES=0B41   SS=0B41   CS=0B41   IP=0108      NV UP EI PL NZ AC PO NC
0B41:0108 83E936            SUB         CX,+36
-T
AX=0000   BX=0000   CX=0078   DX=0000   SP=FFEE   BP=0000   SI=0000   DI=0000
DS=0B41   ES=0B41   SS=0B41   CS=0B41   IP=010B      NV UP EI PL NZ NA PE NC
0B41:010B 90                NOP
```

4. 语句

4-3. 两种类型的标示符是什么?

答案**:**是标号和名称。

4-4.下列哪些命名是正确的:

(a) CX; (b)25C4; (c) @$_X; (d)$25; (e)AT&T.

如果不正确，解释原因

答案**:** 正确的命名是**:**

(c) @$_X (d)$25

不正确**:**

(b)25C4 以数字开头

(e)AT&T &是非法字符

(a)CX 只再与 **CX** 寄存器关联时是对的

4-9. (a) 解释过程的作用

(b) 怎么样定义一个过程的起始和结尾?

(c) 什么时候应该定义 FAR 过程?

(d)什么时候应该定义 NEAR 过程?

答案**:**

(a) 过程提供了相关的代码部分。

(b) 过程名 **PROC**

过程名 **ENDP**

(c) 在另一段中调用代码

(d)在同一段中调用代码

4-10. 哪些 END 语句与结束

(a)一个过程; (b)一个段; (c) 一个程序 相关?

答案**: (a)ENDP (b) ENDS (c) END**

4-11.区分结束汇编过程和结束程序执行的语句.

答案**: END** 表示汇编结束
MOV AX,4C00H　和 **INT 21H** 结束 **CPU** 的运行。

4-12. 把名称 STKSEG, DATSEG, 和 CDSEG 分别分配给堆栈,数据段, 和代码段, 编写所需的 ASSUME 语句.
答案**:**
ASSUME　SS:STKSEG, DS:DATSEG, CS:CDSEG

4-17.通过 AREA5 定义下列数据项中的名称为 ARE1 的数值。
分别是:
(a)一个 1 字节项包含的二进制数相当于十进制数 35。
(b)一个双字包好着连续的数 12,14,22,28,33 和 41。
(c)一个 2 字节的项 包含着未定义的值。
(d)一个 1 字节的项 包含的十六进制相当于十进制数 58。
(e)一个 4 字节的项包含的十六进制相当于十进制数 436。
答案**: (a) AREA1　DB　0010 0011B**
(b) AREA2　DW 12,14,22,28,33,41
(c) AREA3　DW ?
(d) AREA4　DB　3AH
(e) AREA5　DD　1B4H

单选:

1. 汇编程序的语句中，可以省略 (B　)

A. 名称　　B. 操作符　　C. 操作数　　d.注释

2. 正确结束伪指令的句子是(C):

A. 汇编程序把伪指令翻译成机器代码。
B. 伪指令在程序执行期间完成它的功能。
C. 伪指令的功能是告诉汇编程序在汇编过程实现特殊的处理

3. 对于 END 伪指令, 哪句话是对的?　　**C.**

a. END 伪指令是一个可执行指令。
b. END 伪指令指明执行程序的终止之处。
c. END 伪指令指明整个源程序的终止。
d. END 伪指令会在汇编过程生成机器码。

4. 执行下列指令后, 变量 DAB 的内容是 **C**

```
DAW　　DW　　2A05H
DAB　　DB　　0FAH
…
MOV　AL, BYTE　PTR　DAW
SUB　DAB,　AL
```

a. 0DAH　　b. 0FAH　　c. 0F5H　　d. 0D0H

7.　BUF　DW　10H　DUP (3 DUP (2, 10H), 3, 5)

汇编了语句后, 变量 BUF 所分配的地址是　(b).

a.　80H　　b. 100H　　c. 124H　　d.　192H

1.　ARY　DW　10 DUP(?)

…

MOV　AL, TYPE ARY

MOV　　BL, LENGTH　ARY

MOV　　AL,　SIZE　ARY

一系列 MOV 指令运行后，结果是 ？

答案**:**　　**(AL)=2,　　(BL)=10D,　　(al)= 20D**

填空:

1.　可汇编语句中,有两种类型的语句(指令语句)　和　(伪指令语句)

2.　ARY　DW　10DUP(?)

……

MOV　AL, TYPE　ARY　　;执行后(AL)= 2

MOV　CL, SIZE　　ARY　　;　执行后(CL)= 20

分析程序

DA_BY　DB　　83H, 73H, 61H, 94H, 5AH

MOV　CX, WORD　PTR　DA_BY

AND　CX, 0FH

MOV　AL, DA_BY+3

SHL　AL, CL

上述指令执行后,

AL=　**A0H**　,CL=　**03H**

7.1　指令和伪指令的不同点?伪指令的功能是什么?

答案: 指令和伪指令的不同是是否生成可执行的机器码。

伪指令只是用来告诉汇编程序采取特殊的处理，不生成机器码。

7.3　　数值表达式跟地址表达式的不同点？

答案: 数值表达式是一个值可以在汇编过程中被汇编程序计算的表达式。

地址表达式表征着操作数内存项的地址。

7.4　　在汇编程序中,变量和标号的不同点?

答案**:**

标号表示机器指令码所在的内存位置**;**

变量表示着数值所在的内存位置**;**

7.5 计算下面表达式的值

①23H AND 45H OR 67　　②1234H/16+10H

⑤LOW 1234H OR HIGH 5678H　⑥23H SHL 4

答案**:**

① **43H**　　② **133H**

⑤ **76H**　　⑥ **230H**

7.7 如果有如下的一个程序段. 写出运行后 AX 寄存器的内容

```
                ORG     100H
100 VARW    DW      1234H,5678H
104 VARB    DB      3,4
106 VARD    DD      12345678H
10A BUFF    DB      10 DUP(?)
114 MESS    DB      'HELLO'
119 BEGIN:  MOV     AX,OFFSET VARB+OFFSET MESS      ;218H
            MOV     AX,TYPE BUFF+TYPE MESS+TYPE VARD ;6H
            MOV     AX,SIZE VARW+SIZE BUFF+SIZE MESS ;0DH
            MOV     AX,LENGTH VARW+LENGTH VARD ;2H
            MOV     AX,LENGTH BUFF+SIZE VARW    ;CH
            MOV     AX,  TYPE  BEGIN            ;-1
            MOV     AX,  OFFSET BEGIN           ;119H
```

选择**:**

3.　对于 END 伪指令, 那一句是对的?

a.　END 伪指令是一个可执行指令。

b.　END 伪指令指明执行程序的终止之处。

c.　END 伪指令指明整个源程序的终止。

d. END 伪指令会在汇编过程生成机器码

答案：C

```
3. DATA     SEGMENT
            ORG   20H
            NUM1 =   8
            NUM2 =   NUM1 + 10H
 DA1        DB    'COMPUTER'
            DB     0AH, 0DH
 COUNT      EQU   $-DA1
 DA2        DW     'CO', 'MP', 'UT', 'ER'
 DATA       ENDS
```

答案:　(1) DA1 的偏移量是 **20H**

(2) COUNT 的值是 **10D**

(3) DA2 的内容**+2** 字节的位置是 **P**

4. DA3 EQU WORD PTR DA4
DA4 DB 0ABH, 89H
…
SHR DA3, 1
MOV DX, DA3
SHL DA4, 1
MOV CX, DA3

程序执行之后：

CX= **44AAH**, DX=**44D5H**

7.12 如下程序段中有几个语句. 解释每一个符号的属性。

SYMB1 LABEL BYTE
SYM2 EQU THIS BYTE
SYMB3 DW ?
SYMB4 EQU BYTE PTR SYMB3

答案**:** **SYMB1 BYTE**
SYM2 BYTE
SYMB3 WORD
SYMB4 BYTE

4.7 请定义一个数据段 DATASG, 在段中定义字符变量和数据变量，要求如下：
(1) FLD1B 是一个字符变量: ’personal computer’;
(2) FLD2B 十进制数的字节变量 32;
(3) FLD6B 10 个 0 的字节变量;
(4) FLD7B 是一个硬件名称的列表 (ASCII code) 和数量(十进制数)
(10)FLD3W 是(7)中硬件列表地址变量。
(11)FLD4W 是一个字变量包含 5 个十进制数:5,6,7,8,9;
(13)FLD6W 是段中字节数据变量地址和字数据变量地址之差。

答案**:** **DATASG SEGMENT**
FLD1B DB ‘personal computer’
FLD2B DB 32
FLD6B DB 10 DUP(0)
FLD7B DB ‘PART1’,20,’PART2’,50,’PART3’,14
FLD3W DW FLD7B
FLD4W DW 5,6,7,8,9

FLD6W DW FLD3W-FLD1B

4.8　有一个数据段如下定义，PLENTH 的值是多少？意味着什么?

```
PARTNO      DW      ?
PNAME       DB      16   DUP(?)
COUNT       DD      ?
PLENTH      EQU     $-PARTNO
```

答案**:** 值是 **22.**相对 **PARTNO** 的偏移量

4.9　如下有几个语句，问：　L 的值?

```
BUFF     DB  1,2,3,'123'
EBUFF    DB  0
L        EQU     EBUFF-BUFF
```

答案：**L** 的值是 **6**

4.10　　有如下数据段。

（1）　用 MOV　指令把 LNAME 的有效地址移到 AX。

（2）　用一条指令来移动 CODE_LIST 前两个字节的内容到 SI。

（3）　编写伪指令让 CODE_LIST 的与段得长度相等.

```
LNAME           DB    30 DUP (?)
ADDRESS         DB    30 DUP(?)
CITY            DB    15 DUP(?)
CODE_LIST       DB    1,7,8,3,2
```

答案:

```
(1) MOV    AX, OFFSET    LNAME
(2) MOV    SI , WORD PTR CODE_LIST
(3) CODE_LENTH    EQU      $-CODE_LIST
```

4.11 尝试写出整个数据段 DATA_SEG, 它把 5 赋给一个字节, 且把整数–1,0,2,5 和 4 放入 10 个字的数组 DATA_LIST 的前 5 个单元 。然后写出整个代码段, 功能是把 DATA_LIST 的前 5 个数中的最大值和最小值放到 MAX 和 MIN。

答案:

```
DATA_SEG        SEGMENT
DATA1           DB    5
DATA_LIST       DW    -1,0,2,5,4, 5 DUP(?)
MAX             DB    ?
MIN             DB    ?
DATA_SEG        ENDS
CODE_SEG        SEGMENT
                ASSUME     DS:DATA_SEG,CS:CODE_SEG
```

```
BEGIN:          MOV    MAX, 5
                MOV    MIN, -1
CODE_SEG        ENDS
                END    BEGIN
```

6-1. 寻址

3.1 给出(BX)=637DH,(SI)=2A9BH,偏移量 D=7237H,尝试指出下列每一种寻址模式的有效地址。

(1) 立即寻址;

(2) 直接寻址

(3) 用 BX 的寄存器寻址;

(4) 用 BX 的寄存器间接寻址;

(5) 用 BX 的寄存器相对寻址;

(6) 基址变址寻

(7)相对基址变址寻

答案:(1)无

(2)EA=7237H

(3)无

(4)EA=637DH

(5)EA=D+[BX]=7237H+637DH=0D5B4H

(6)EA=[BX]+[SI]=637DH+2A9BH=8E18H

(7)EA=D+[BX]+[SI]=7237H+637DH+2A9BH=1004FH

3.2 根据下列要求, 写出相关的汇编语言指令。

(1) 把 BX 的内容和 DX 的内容相加，结果放入 DX 中。

(2) 把 AL 的内容加上内存地址的内容，并把结果放到 AL 中。内存地址由 BX 和 SI 进行基址变址寻址所得。

(3)把 CX 的内容加上内存地址的一个字，并把结果放到内存地址中。内存地址由 BX 和偏移量 0B2H 进行寄存器相对寻址所得。

(4) 把内存地址的内容与数 2A59H 相加，并把结果放入内存地址。内存地址由偏移量 0524H 进行直接寻址所得。

(5) 把数值 0B5H 与 AL 寄存器的内容相加, 结果放入 AL 寄存器。

答案: (1)ADD DX, BX

(2)ADD AL, BX[SI]

(3)ADD WORD PTR 0B2H [BX], CX

(4)ADD WORD PTR [0524H], 2A59H

(5)ADD AL, 0B5H

3.3 写出指令，把首址是 BLOCK 的字数组中的第六个字移入 DX 寄存器。使用如下的寻址方式。

(1) 寄存器间接寻址
(2) 寄存器相对寻址
(3) 基址变址寻址

答案:
(1) LEA　BX, BLOCK+10
　　MOV　DX, WORD PTR [BX]
(2) LEA　SI, BLOCK
　　MOV　DX, WORD PTR 0AH[SI]
(3)LEA　BX, BLOCK
　MOV　SI, 0AH
　MOV　DX, WORD PTR [BX][SI]

3.4　给出(DS)=2000H,(BX)=0100H,(SI)=0002H,(20100)=12H,　(20101)=34H,　(20102)=56H,　(20103)=78H, (21200)=2AH,(21201)=4CH,(21202)=B7H, (21203)=65H,
尝试解释：执行指令后，AX 寄存器的内容。
(1)　MOV　AX, 1200H
(3) MOV　AX, [1200H]
(5) MOV　AX, 1100[BX]
(7) MOV　AX, 1100[BX][SI]

答案:
(1) (AX)=1200H
(3)(AX)=4C2AH
(5)(AX)=4C2AH
(7)(AX)=65B7H

3.8　给出(DS)=2000H,(ES)=2100H,(SS)=00A0H,(BX)=0100H,(BP)=0010H,数据段中　VAL　的偏移地址是 0050H,指出源操作数段得寻址方式和物理地址。
(2) MOV　AX, BX
(4) MOV　AX, VAL
(6) MOV　AX, ES:[BX]
(8) MOV　AX, [SI]
(10)MOV　AX, VAL[BX]
(12)MOV　AX, VAL[BX][SI]

答案:
(2)寄存器寻址
(4)直接寻址,物理地址=20050H
(6)寄存器间接寻址,物理地址=21100H
(8)寄存器间接寻址,物理地址=200A0H
(10)寄存器相对寻址,物理地址=20150H

(12)相对基址变址寻址,物理地址=201F0H

6-2

单选：

1.当执行 POP [BX] 和寻找目标操作数时，段地址跟偏移地址是在(**B**)中。

a. 无段地址也无偏移地址

b. DS 和 BX 中

c. ES 和 BX 中

d. SS 和 SP 中

2.下列指令中, 哪个是错的?

a. MOV SS:[BX+DI],1000H

b. MOV DX, 1000H

c. MOV WORD PTR [BX],1000H

d. MOV DS, 2000H

答案**:**

d. 错误**:** 立即数到段寄存器

编写指令:

1. D1 DB 20H DUP(?)

D2 DW D1

给出三种方式，使用一条指令来把 D1 的偏移量载入到 SI 中。

答案:

1. LEA SI, D1

2. MOV SI, OFFSET D1

3. MOV SI, D2

简答**:**

1. 指出下列的错误**:**

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

解**:**

1) MOV AH, BX **;**操作数大小不同

2) MOV [BX], [SI] ;出错：内存到内存

3) MOV AX,[SI] [DI] ;出错：把**[SI][DI]**放到一起

4) MOV MYDAT [BX][SI], ES:AX ; 与 **2** 同**,**

5) MOV BYTE PTR[BX], 1000 ; 与 **1** 同,

6) MOV BX, OFFSET MYDAT[SI] ;删除**[SI]**

7) MOV CS, AX ;出错**: CS** 是第一操作数

填空:

```
.MODE   SMALL
.DATA
DFATA1     DW     2000H
DATA2   DW 3000H
.CODE
.STARTUP
LEA       SI,   DATA1
MOV       DI,    OFFSETDATA2
MOV       BX, [SI]
MOV       CX, [DI]
MOV       [SI], CX
MOV       [DI],BX
.EXIT
END
```

程序执行之后, (DATA1) = **3000H**

(DATA2) = **2000H**

3.10 TABLE 是一个在数据段中 0032 上的符号, 它的内容是 1234H, 下面指令有什么不同? 执行指令后 AX 寄存器的内容是什么?

MOV AX, TABLE

LEA AX,TABLE

答案**:** **(AX)=1234H**

(AX)=0032H

3.11 执行下列指令后 AX 寄存器的内容是什么?

```
TABLE   DW        10,20,30,40,50
ENTRY    DW       3
          …
          ……
         MOV      BX,OFFSET TABLE
         ADD      BX ENTRY
```

MOV AX [BX]

(BX)=0000H

(BX)=0003H

答案**:** **(AX)=1E00H**

7.8 下列指令中，符号 ABCD 是一个变量。说出两条指令的不同。

MOV AX,OFFSET ABCD

LEA AX,ABCD

答：两条指令都是取 ABCD 的偏移地址

但 MOV 指令能以更少的时间完成同样的功能。

7.10 如下的一个程序段。请改正错误的指令。

```
VARW    DW   1234, 5678H
VARB    DB   3,4
VARD    DD   12345678
        ……
        MOV     AX,VARB
        MOV     VARD,BX
        MOV     VARD+2,ES
        MOV     CL,VARW+3
        LES     DI,VAFW
```

改正如下：

```
MOV   AX , WORD PTR VARB
MOV   WORD PTR VARD ,   BX
MOV   WORD PTR VARD+2 , ES
MOV   CL , BYTE PTR WARW +3
```

8-1 、编程逻辑和控制要求

8-2

(a) 一个近 JMP, LOOP, 和条件跳转指令可以跳转的字节的最大值是什么？ (b) 机器代码的操作数的什么特点导致了这种限制？？

答案:

(a)JMP：两操作数字节.

LOOP：一个操作数字节

条件跳转：一个操作数字节

8-3 一条 JMP 指令从偏移位置 05C8H 开始。指出 JMP 的操作数跳转的偏移地址, 基于下列目标代码: (a) 14H;(b) 7DH;(c) A3H

答案:

(a) 05C8H+0014H+2H=05DEH.

(b) 05C8H+007DH+2H=0647H.

(c) 05C8H+FFA3H+2H=056DH.

8-5 假定 AX 和 BX 包含了无符号数据且 CX 和 DX 包含了带符号数据。指出下列所需的 CMP (必要时)和条件跳转指令 ：

(a) AX 等于或小于 BX ?

(b) CX 等于或小于 DX?

(c) CX 的值大于 DX?

(d) AX 的值大于 BX?

(e) DX 包含零吗?

(f)有溢出吗?

答案:

（a）CMP AX, BX

JBE Address

（b）CMP CX, DX

JLE Address

（c）CMP CX, DX

JG Address

（d）CMP AX, BX

JA Address

（e）CMP DX, 0

JE Address

（f）JO Address

8-10 在一个.EXE 程序中, F10 调用 G10, G10 调用 H10, H10 调用 J10.因为这些调用,堆栈中现在包含多少个地址?

答案：3 个地址

8-13 假定 BX 包含二进制 10111001 10111001 和 CL 包含 3. 指出下列指令运行后 BX 的十六进制值 ：

a) SHL BL 1;

b) SHL BX，CL;

c) SHR BX,CL;

d) SHR BX,1;

e) SAL BH,1;

f) ROR BX,CL;

g) ROR BL,CL

答案：

a) B972H（BL 逻辑左移一位）

b) 0CDC8H（BX 逻辑左移 3 位）

c) 1737H（BX 逻辑右移 3 位）

d) 5CDCH（BX 逻辑右移 1 位）
e) 72B9H（BX 算术左移 1 位）
f) 3737H（BX 循环右移 3 位）
g) B937H（BL 循环右移 3 位）

8-14 用移位, 移动, 和加法指令来以 40H 初始化 CX 并且乘于 10.
答案：

```
MOV CX, 40H
MOV AX, CX
MOV BX, CX
MOV    CL, 3
SHL    AX, CL
SHL    BX, 1
ADD AX, BX
```

8-15 命名为“旋转位”的章节末尾有一个例子，把 DX：AX 乘于 2。修改这个程序 a) 乘于 4 b) 除于 4 c) 把 DX:AX:BX 中的 48 位乘于 2。
答案：

a)
```
SHL AX, 1
RCL DX, 1
SHL AX, 1
RCL DX, 1
```
b)
```
SAR DX, 1
RCR AX, 1
SAR DX, 1
RCR AX, 1
```
c)
```
SHL BX, 1
RCL AX, 1
RCL DX, 1
```

2.假定 VAR1 和 VAR2 是字变量, LAB 是标号, 指出下列的错误:

```
ADD VAR1,VAR2
SUB    AL, VAR1
JMP     LAB[SI]
JNZ     VAR1
JMP     NEAR    LAB
```

错误:
1.ADD 内存, 内存
2. 两个操作数大小不同
3.删除[SI] 或者 删除 LAB
4. 条件跳转范围<= 字节, 不是字

5.　应该是 JMP NEAR PTR LAB

3、A DW 1234H
B DW 5678H
……
PUSH A
PUSH B
POP A
POP B
回答：
① 执行之后（A）＝5678H ,(B)＝ 1234H
② 执行程序之前 SP＝200H，
　　执行之后 SP＝ 200H

8-2

分析指令：
1、MOV AX，6540H
MOV DX，3210H
MOV CL，04
SHL DX，CL
MOV BL，AH
SHL AX，CL
SHR BL，CL
OR DL，BL

执行指令之后，
（AX）＝5400H,（BL）＝06H,
（DX）＝2106H

2、MOV AL，200
SAR AL，1
MOV BL，AL
MOV CL，2
SAR AL，CL
ADD AL，BL
　执行指令之后
（BL）＝0E4H
（AL）＝0DDH

3 .BLK1 DB 46，84，34，－5，20，122，73

```
MOV CX，7
LEA S1，BLK1
NEXT：  MOV AL，[SI]
INC SI
TEST AL，81H
LOOPZ NEXT
MOV BL，[SI]
```

执行指令之后：
（AL）＝0FBH（BL）＝20 / 14H

6、

```
AND   AL，AL
JZ      BRCH1
RCR   AL，1
JZ      BRCH2
RCL   AL，1
INC   AL
JZ    BRCH3
```

执行指令之后, 回答：
（1）当（AL）＝0 时，程序跳到 BRCH1
（2）当（AL）＝1 时，程序跳到 BRCH2
（3）当（AL）＝0FFH 时，程序跳到 BRCH3

12.串

单选:
1.当在串指令之前使用 REPE 立即前缀时,当 （**C**），串指令将会停止。
A. CX=0 and ZF=0 B. CX=0 and ZF=1
C. CX=0 or ZF=0 D. CX=0 or ZF=1
2.下列哪条指令有合理和充分的意义? D
A. REP LODSB B. REP SCASB
C. REP CMPSB D. REP MOVSB

13.算术

13.1
对于无符号和带符号的数据。(a)字节的最大值是什么(b)字的最大值是什么？
答案:
(a):对于带符号数据:+127and128;
  对于无符号数据:0 and 255
(b):对于带符号数据:-32768 and 32767
  对于无符号数据:0 and 65535:

13-2 区分算术操作的进位和溢出。
答案:
C 位代表无符号数据的溢出**;**
C=1，有进位/借位; 其他情况 C=0。
O 代表带符号数据的溢出。
加法**:** 当两个操作数的符号相同，但是结果的符号与操作数不同时，**O** 位被置位 (O=1)。
其他情况 O = 0。
减法**:** 如果两个操作数符号不同, 结果和减数的符号相同，那么 O =1;其他情况 O = 0。

13-3 对于下列的二进制加法, 指出二进制的和作为带符号和无符号的十进制的和， 指出溢出标志和进位标志的值:

a) 0011 0011 + 0001 1000
b) 0111 0110 + 0001 1001
c) 1101 0110 + 0101 1001

答案:

	a	b	C
Sum	0100 1011	1000 1111	0010 1111
CF	0	0	1
OF	0	1	0

13.5 编写代码:
a): 字 VALUE1 和字 VALUE2 相加;

```
MOV AX, VALUE2
ADD AX, VALUE1
```

(b): 以 VALUE1 开始的双字 和双字 VALUE2 相加;

```
MOV EAX, VALUE1
ADD EAX, VALUE2
```

13.7 编写代码(MUL)
(a) 字 VALUE1 和 VALUE2 相乘;

```
MOV AX, VALUE1
MUL VALUE2
```

(b) 以 VALLE1 开始的双字 与字 VALUE2 相乘；

```
MOV   AX, VALUE1
MUL   VALUE2
MOV   RESULT, AX
MOV   RESULT+2, DX
MOV   AX, VALUE1+2
MUL   VALUE2
ADD   RESULT+2, AX
```

```
    ADC     RESULT+4, DX
    ADC     RESULT+6, 0
```

13.8 编写代码(DIV) :
(a): 字 VALUE1 除于 36;

```
MOV AX, VALUE1
MOV BL, 36
DIV BL
```

(b): 以双字开始的 VALUE1 除于字 VALUE2;

```
MOV     AX, VALUE1
MOV     DX, VALUE1+2
DIV       VALUE2
```

13.9
除了除以零之后，除以什么数会导致溢出错误？
答案: 除数必须比被除数小。例如除于 1，如除以 1 生成一个商数，是与被除数相同，也可能导致中断和溢出错误。

3.40 分析下列代码段:

```
ADD   AX, BX
JNO     L1
JNC     L2
SUB     AX, BX
JNC     L3
JNO     L4
JMP     SHORT L5
```

AX 的内容和 CX 的内容如下 :

```
AX          BX
B568        54B7
```

程序执行后，会跳转到哪里?
答案:

```
L1
L3
```

3.47 指令填空:
(1) LOOP L20 (2)LOOPE L20 (3)LOOPNE L20
尝试指出在 3 个不同情况下，程序执行之后 AX,BX,CX,DX 寄存器的内容 ？

```
TITLE           EXLOOP.COM
CODESG          SEGMENT
                ASSUME CS: CODESG, DS: CODESG,SS:CODESG
                OGR       100H
BEGIN:          MOV     AX, 01
                MOV     BX, 02
```

```
                MOV     DX, 03
                MOV     CX, 04
L20:            INC     AX
                ADD     BX, AX
                SHR     DX, 1
                (   )
                RET
CODESG          ENDS
                END     BEGIN
```

答案: (1) LOOP　L20 :AX=05H ; BX=10H ; CX=00H ; DX= 00H

(2) LOOPNE　L20: AX=03H; BX=7H; CX=2H ; DX=00H

(3)LOOPE　L20:没有进行循环

AX=02H; BX=4H; CX=3H; DX=1H

4-1　写出 ADD 指令，完成如下操作：

(a)　把 BX 加到 AX

ADD AX, BX

(b) 把 12H 加到 AL

ADD AL, 12H

(c) 把 EDI 加到 EDP

ADD EDP, EDI

(d) 把 22H 加到 CX

ADD CX. 22H

(e) 把 SI 的地址数据加到 AL

ADD AL, [SI]

(f)把 CX 加到 FROG 地址的数据上

ADD FROG, CX

4.2 指出指令 ADD ECX,AX 的错误?

答案:　ECX 和 AX 的大小不一致。

4.3 可以用 ADD 指令把 CX 加到 DS 吗?

答案:　不可以，因为 DS 是段寄存器。

4.4　给出 AX=1001H,DX=20FFH,执行 ADD AX，DX 后，列出标志寄存器的总和和每一位的内容。

答案:

AX=3100H;

C=0;最高位没有进位

A=1;第三和第四位没有进位

S=0;结果为正

Z=0;结果非零

O=0;结果没有溢出

4.6 设计一个简短的程序，把 AX,SX,CX,DX 和 SP 累加在一起,把结果存进 DI 中。

```
LEIJIA    PROC    NEAR
          ADD    AX, BX
          ADD    AX, CX
          ADD    AX, DX
          ADD    AX, SP
          MOV    DI, AX
          RET
LEIJIA ENDP
```

4.9　写出把 sp 的内容加 1 的指令。
指令：INC SP

4.10 写出 SUB 指令，完成如下操作。

(a) 从 AX 中减去 BX
指令：SUB AX, BX
(b) 从 DH 中减去 0EEH
指令：SUB DH, 0EEH

4.11 解释 SBB [DI-4],DX 的结果
答案: 从由 DI-4 寻址的内存单元中，减去 CX，同时也减去借位。

4.12 解释 SUB 和 CMP 指令的不同
答案: 指令 SUB 的功能是 从源操作数减去目标操作数，然后把结果存储到目标操作数，
而 CMP 指令并不会改变两个操作数的内容,只会改变标志位 。

4.13 当 8-位 操作数加上另外一个数,结果存在哪里?
答案: 在 AX 寄存器中。

4.14 MUL EDI 的结果存在哪里？
答案 :在 EDX-EAX 中。

4.15 当 8-bit 操作数被除时,结果存在哪里?
答案: AX.

4.16　　写一个简短的程序, 把 BL 中的数据除以 CL 中的数据, 然后把结果加上 2,最后的结果存在 DX 中, 是 16-位数。

答案:
MOV AL, BL

```
MOV AH, 0
DIV CL
MOV AH, 0
ADD AX, AX
MOV DX, AX
```