汇编语言程序设计

作业一

2005010130 侯杰

1.1 将下列十进制数转换为十六进制数：

100 8798 4096 777

答：100D=64H, 8798D=225EH, 4096D=1000H, 777D=309H

1.2 完成下列十六进制数的运算：

13B4 + 33 EDAC + BAF ABCD – EF 7AB * 6F

答：13B4+33=13E7, EDAC+BAF=F75B

ABCD-EF=AAFE, 7AB*6F=35325

1.3 下列各数均为十进制数，用 16bit 二进制补码进行计算，给出十六进制表示的运算结果，若结果出错，请注明：

20000 + 15000 （-128）+ 1024 （-4096）- 30000 256 + （-32768）

答：20000+15000 用补码表示并运算为

0100111000100000+ 0011101010011000

=1000100010111000(正加正得负，出错)

(-128)+1024 用补码表示并运算为

1111111110000000+0000010000000000

=0000001110000000=0380H

（-4096）- 30000 用补码表示并运算为

1111000000000000+1000101011010000

=0111101011010000(负加负得正，出错)

256 +（-32768）用补码表示并运算为

0000000100000000+1000000000000000

=1000000100000000=8100H

1.4 写出下列字符串的 ASCII 码值。

I love 2008’summer!

One world, One dream.

答：

I love 2008’summer!

ASCII:49,20,6C,6F,76,65,20,32,30,30,38,27,73,75,6D,6D,65,72,21

One world, One dream.

ASCII:4F,6E,65,20,77,6F,72,6C,64,2C,20,4F,6E,65,20,64,72,65,61,6D,2E

2.1 80x86 微机的存储器中存放如下信息，读出 10003H 和 10004H 字节单元的内容，10002H 字单元的内容以及 10001H 双字单元的内容。

1A	10000
2B	10001
3C	10002
4D	10003
5F	10004

答：字节单元 10003H 内容：4D，字节单元 10004H 内容：5F

字单元 10002H 内容：4D3C，双字单元 10001H 内容：5F4D3C2B

2.2 在实模式下，逻辑地址（段地址:偏移地址）为 0B2F:011B，计算物理地址；对应这个物理地址，有几种可能的逻辑地址？

答：物理地址为 0B2F0+011B=0B40B

对于物理地址 0B40B，考虑其由段地址和偏移地址组成。物理地址最低 4 位为 B，由于段地址对物理地址最低 4 位仅提供 0，所以偏移地址最低四位必须为 B。且偏移地址的范围为 000B~0B40B。从 000B 到 0B40B 中最低 4 位为 B 的地址可能有 0B41 种，即偏移地址可能有 0B41 种，所以逻辑地址同样可能有 0B41 种。

2.3 在实模式下，用 debug 调试 dis.exe 的截屏如下。（1）存储器是如何分段的？数据段有多长？（2）此时刚执行完哪条指令？（3）FLAGS 中

CF/OF/SF/ZF 的值是多少，有什么含义？（4）在第一条指令 PUSH DS 执行之前，DS 和 ES 会是多少？

```
C:\asm>debug dis.exe
-u
0BAA:0000 1E            PUSH    DS
0BAA:0001 2BC0          SUB     AX,AX
0BAA:0003 50            PUSH    AX
0BAA:0004 B8A70B        MOV     AX,0BA7
0BAA:0007 8ED8          MOV     DS,AX
0BAA:0009 8EC0          MOV     ES,AX
0BAA:000B B8FFFF        MOV     AX,FFFF
0BAA:000E 050100        ADD     AX,0001
0BAA:0011 8B0E2200      MOV     CX,[0022]
0BAA:0015 BB0000        MOV     BX,0000
0BAA:0018 8A17          MOV     DL,[BX]
0BAA:001A B402          MOV     AH,02
0BAA:001C CD21          INT     21
0BAA:001E 43            INC     BX
0BAA:001F E2F7          LOOP    0018
-g11

AX=0000  BX=0000  CX=0052  DX=0000  SP=FFFC  BP=0000  SI=0000  DI=0000
DS=0BA7  ES=0BA7  SS=0BA7  CS=0BAA  IP=0011   NV UP EI PL ZR AC PE CY
0BAA:0011 8B0E2200      MOV     CX,[0022]                      DS:0022=0015
-
```

答：

(1)数据段，附加段，堆栈端从内存地址 0BA7 开始存放。代码段从内存地址 0BAA 开始存放。数据段长度为 0030

(2) 此时刚刚执行完过 ADD AX,0001 指令

(3) CF 值为 1，表示产生进位

OF 值为 0，表示没有溢出

SF 值为 0，表示符号为正

ZF 值为 1，表示运算结果为 0

(4)0BA7-10H=0B97H,其中 10 为 PSP 的长度

所以最开始的 DS 和 ES 应为 0B97H

3.2 试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，记过存入 DX 寄存器中

ADD DX,BX

（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，并把结果送到 AL 寄存器中。

```
ADD AL,[BX][SI]
```

（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中。

```
ADD [0B2H][BX],CX
```

（4)用位移量为 0524H 的直接寻址方式把存储器中的一个字与数 2A59H 相加，并把结果送回该存储单元中。

```
ADD [0524H],2A59H
```

（5）把数 0B5H 与(AL)相加，并把结果送回 AL 中。

```
ADD AL,0B5H
```

3.4 现有

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

3.7 在 0624 单元内有一条二字节 JMP SHORT OBJ 指令，如其中位移量为(1)27H,(2)6BH,(3)0C6H,试问转向地址 OBJ 的值是多少？

答：(1)064BH　(2)068FH　(3)05EAH

3.14 设(DS)=1B00H,(ES)=2B00H,有关存储单元的内容如图所示（图略）。写出两条指令把字变量 X 装入 AX 寄存器。

答：指令如下

```
LES BX,DS:[2000H]
MOV AX,ES:[BX]
```

3.15 求出以下各十六进制数与 62A0H 的和，并根据结果设置标志位 SF,ZF,CF 和 OF 的值。

答：(1)1234H+62A0H=74D4　　SF:0　ZF:0　CF:0　OF:0

(2)4321H+62A0H=A5C1　　SF:1　ZF:0　CF:0　OF:1

(3)CFA0H+62A0H=3240　　SF:0　ZF:0　CF:1　OF:0

(4)9D60H+62A0H=0000　　SF:0　ZF:1　CF:1　OF:0

3.19 下列程序段中的每条指令执行完后，AX 寄存器以及 CF，SF，ZF 和 OF 的内容是什么？

（注：keep 表示保持原来的状态）

		AX	CF	SF	ZF	OF
MOV	AX,0	0000H	keep	keep	keep	keep
DEC	AX	FFFFH	keep	1	0	0
ADD	AX,7FFFH	7FFEH	1	0	0	0
ADD	AX,2	8000H	0	1	0	1
NOT	AX	7FFFH	0	1	0	1
SUB	AX,0FFFFH	8000H	1	1	0	1
ADD	AX,8000H	0000H	1	0	1	1
SUB	AX,1	FFFFH	1	1	0	0
AND	AX,58D1H	58D1H	0	0	0	0
SAL	AX,1	B1A2H	0	1	0	1
SAR	AX,1	D8D1H	0	1	0	0
NEG	AX	272FH	1	0	0	0
ROR	AX,1	9397H	1	0	0	1

3.21 写出对存放在 DX 和 AX 中的双字长数求补的指令序列

答：

```
NOT   AX
NOT   DX
ADD   AX,1H
ADC   DX,0
```

3.28 下列程序段执行完后，BX 寄存器中的内容是什么？

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

答：C02DH

3.31 假设程序中数据定义如下：

```
STUDENT_NAME   DB 30 DUP(?)
STUDENT_ADDR   DB 9 DUP(?)
PRINT_LINE     DB 132 DUP(?)
```

分别编写下列程序段：

（1）用空格符清除 PRINT_LINE 域

答：

```
      CLD
      MOV   AL,20H
      MOV   CX,132
      LEA   DI,PRINT_LINE
      REP   STOSB
```

（2）在 STUDENT_ADDR 中查找第一个'-'

答：

```
      CLD
      MOV   AL,'-'
      MOV   CX,9
      LEA   DI,STUDENT_ADDR
      REPNE SCASB
      JE    found    ;Found it
      JMP   notfound ;Not found it
found:
      …
notfound:
      …
```

（3）在 STUDENT_ADDR 中查找最后一个'-'

答：

```
      STD
      MOV   AL,'-'
      MOV   CX,9
      LEA   DI,STUDENT_ADDR+8
      REPNE SCASB
      JE    found    ;Found it
      JMP   notfound ;Not found it
found:
      …
notfound:
      …
```

（4）如果 STUDENT_NAME 域中全是空格符，填入'*'

答:

```
      CLD
      MOV   AL,20H
      MOV   CX,30
      LEA   DI,STUDENT_NAME
      REPNE SCASB
      JE    fill      ;There are all spaces, ready to fill '*'
      JMP   otherwise;Otherwise
fill:
      MOV   AL,'*'
      MOV   CX,30
      LEA   DI,STUDENT_NAME
      REP   STOSB
      …
otherwise:
      …
```

（5）把 STUDENT_NAME 移到 PRINT_LINE 的前 30 个字节中，把 STUDENT_ADDR 移到 PRINT_LINE 的后 9 个字节中

答:

```
      CLD
      ;step 1
      LEA   SI,STUDENT_NAME
      LEA   DI,PRINTLINE
      MOV   CX,30
      REP   MOVSB
      ;step 2
      STD
      LEA   SI,STUDENT_ADDR+8
      LEA   DI,PRINT_LINE+131
      MOV   CX,9
      REP   MOVSB
```

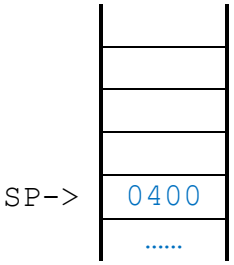
3.35 指令 CMP AX，BX 后面跟着一条格式为 J… L1 的指令。其中…可以是 B，NB，BE,NBE,L,NL,LE 和 NLE 中的任一个。如果 AX 和 BX 的内容给定如下，请问这八条转移指令中的哪几条将引起转移到 L1？

```
   AX    BX
(1)1F52  1F52     JNB,JBE,JNL,JLE
(2)88C9  88C9     JNB,JBE,JNL,JLE
(3)FF82  007E     JNBE,JNB,JL,JLE
```

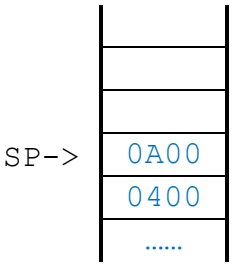
(4)58BA 020E JNBE,JNB,JNLE,JNL
(5)FFC5 FF8B JNBE,JNB,JNLE,JNL
(6)09A0 1E97 JB,JBE,JL,JLE
(7)8AEA FC29 JB,JBE,JL,JLE
(8)D367 32A6 JNBE,JBE,JL,JLE

3.39 考虑一下的调用序列，并画出每次调用及返回时的堆栈状态

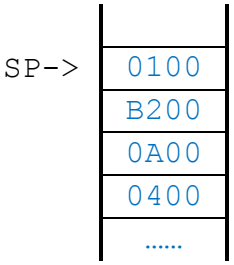
（1）MAIN 调用 NEAR 的 SUBA 过程（返回的偏移地址为 0400）



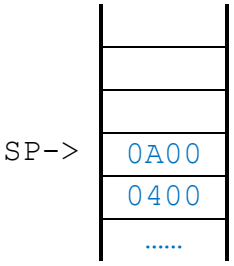
（2）SUBA 调用 NEAR 的 SUBB 过程（返回的偏移地址为 0A00）



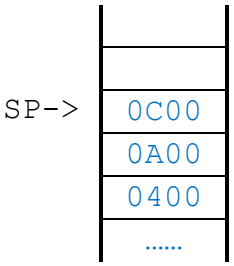
（3）SUBB 调用 FAR 的 SUBC 过程（返回的段地址为 B200，偏移地址为 0100）



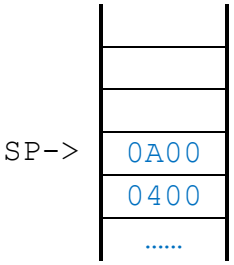
（4）从 SUBC 返回 SUBB



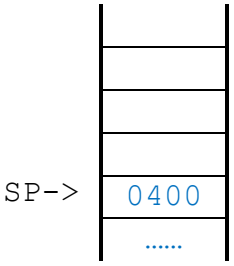
（5）SUBB 调用 NEAR 的 SUBD 过程（返回的偏移地址为 0C00）



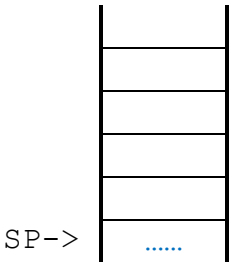
（6）从 SUBD 返回 SUBB



（7）从 SUBB 返回 SUBA



（8）从 SUBA 返回 MAIN



（9）从 MAIN 调用 SUBC（返回的段地址为 1000，偏移地址为 0600）