

汇编语言 第二次作业

2018302070001 沈思源

一、填空题

1. "后进先出"(LIFO)或者叫做"先进后出"
2. 立即数寻址方式
3. PUSH SI
4. LEA AX, A
5. EXTRN, PUBLIC
6. AX
7. 7FFFH; SF= 0, CF= 1, OF= 1
8. 8001H; SF=1, CF=0, OF= 0
9. 带符号数比较条件转移指令：

```
1  P: ....
2      ....
3      ....
4      CMP AX, BX
5      JNL P
```

10. AX , DX
11. 5F3FH, 5230H
12. IP ; CS,IP
13. 0136H , 013CH

二、选择题

1. A
2. D 需要对一些不确定的存储属性需要显式指定,BX和BP寄存器不允许出现在同一个[]内, SI和DI也不能同时出现。
3. C ;偏移地址为 09CH, 说明是短跳转指令, 指令长度为 2 bytes; 相对寻址的跳转指令中的偏移是以该跳转指令的下一条指令为基的,0135H+02H+9CH ???
4. B
5. C
6. C
7. D
8. A ; 12AB0H+00ABH+100H
9. D ;寄存器相对寻址
10. A

三、改错题

```

1.  CMP 0, AX
2.  INC  [BX]
3.  MOV  [DI], [SI]
4.  AND  AL, [DX]
5.  CMP CX, -10
   JB  L
6.  POP  AH
7. MOV  DS, DATA
8.  DIV  300

9.  LEA  M, BUF
10. MUL  30H

```

1. 错误, CMP指令和SUB指令操作相同, 双操作数指令中可以使用立即数寻址, 但是立即数只能位于源操作数字段, 这里位于OPD目的操作数字段, 故错误. `CMP AX,0`
2. 错误, 这里是不确定的存储属性, 需要显式指定, [BX]表示的是用BX寄存器值做地址, 必须指定这个数据的类型. `INC BYTE PTR[BX]`
3. 错误, 同样是没有指定存储属性,且**内存到内存不合法**,源操作数和目的操作数不能同时为存储器操作数, 即存储单元之间不能用MOV指令直接传送.可以使用寄存器进行中转.

```

1  MOV AX, [SI]
2  MOV [DI], AX

```

4. 类型不明确. `AND AL, BYTE PTR[DX]`
5. 功能执行不正确.既然是CX与有符号数进行比较, 条件跳转指令也需要用有符号数比较转移指令, 而不是无符号比较转移指令
`CMP CX,-10`
`JL L`
6. 出栈数据应该是一个16位(字)或者32位(双字)的数据, 而AH是8位寄存器. `POP AX`
7. 如果DATA是前面说明过的数据段, 那么MOV指令不允许在两个存储单元之间直接传送数据, 也不允许在两个段寄存器之间传送信息。
如果DATA是一个立即数数据, 立即数是不允许直接送段寄存器DS的

MOV AX,DATA

MOV DS,AX

8. 单操作数指令不允许使用立即数寻址方式.

MOV CX,300

DIV CX

9. **LEA OPD,OPS**指令要求OPD是16/32位的通用寄存器，不能用M主存储器

10. 单操作数指令不能直接使用立即数.

MOV CL,30H

MUL CL

四、分析题

数据段的段地址为 2000H，定义如下：

DATA SEGMENT

ORG 10H

A DW '3a','a','b','ab', D

B DB 2 DUP (2 DUP (9), 'A', 'B')

F EQU \$-B

C equ byte ptr A

D equ this word

E DD 44332211H , F, B

DATA ENDS

画出存储分配图

变量名	偏移地址	存储单元数据

变量名	偏移地址	存储单元数据
A	0010H	'a'
	0011H	'3'
	0012H	61H ('a')
	0013H	00H
	0014H	62H ('b')
	0015H	00H
	0016H	'b'
	0017H	'a'
	0018H	11H
	0019H	22H
	0020H	'B'
	0021H	'A'
B	0022H	09H
	0023H	09H
	0024H	'B'
	0025H	'A'
	0026H	09H
	0027H	09H
	0028H	11H
	0029H	22H
E	0030H	33H
	0031H	44H
	0032H	08H
	0033H	00H
	0034H	00H
	0035H	00H
	0036H	20H
	0037H	00H
	0038H	00H
	0039H	00H

变量名	偏移地址	存储单元数据

(2) 执行指令 LDS SI, E 后 (SI) = _____, (DS) = _____。

执行指令 LEA SI, D-2 后 (SI) = _____。

执行指令: LEA SI, A MOV AX, 2[SI] 后 (AX) = _____。

1. 执行 LDS SI, E 后, EA=E=0028H, (EA)->SI, (EA+2)->DS

因此 (SI)=(EA)=2211H, (DS)=4433H

2. 执行指令 LEA SI, D-2 将 D-2 EA->SI, (SI)= 220FH ???

3. 执行 LEA SI, A 后, (SI)=EA_A=0010H

执行 MOV AX, 2[SI] 后, (AX)=(2+SI)=(0012H)=0061H='a'

执行指令 LEA SI, A LEA SI, 4[SI] 后 (SI) = _____。

4. 执行指令 LEA SI, A 后, (SI)=0010H

执行指令 LEA SI, 4[SI] 后, EA=4+SI=0014H, (SI)=0014H

五、阅读下列程序段，回答问题

1.

```

1  MOV DX, X+2    ; P高位->DX
2  MOV AX, X      ; P低位->AX
3  ADD AX, X      ; 2P低位->AX
4  ADC DX, X+2    ; 2P高位+来自低位相加的进位CF->DX
5  CMP DX, Y+2    ; 2P高位 COMPARE Q高位; 高位有符号判断
6
7  JL L2
8  JG L1
9
10 CMP AX, Y      ; 低位无符号判断
11
12 JBE L2
13
14 L1: MOV AX, 1
15     JMP EXIT
16 L2: MOV AX, 2
17 EXIT:
18

```

2.

```

1 OR BX,BX
2 JNS L1
3     MOV DL '-'
4     MOV AH,2
5     INT 21H
6     NEG BX
7 L1:CALL OUTBX16 ;将BX的无符号二进制数以16进制方式输出

```

问题：(a) 若运行前 (BX) = 7FF8H，运行后显示输出结果是什么？

(b) 若运行前 (BX) = 9FF8 H，运行后显示输出结果是什么？

(a): 运行前(BX)=7FF8H=0111 1111 1111 1000进行**或运算**后，符号位SF=0(正数)，结果为正则转移至L1，调用子程序，输出:7FF8H

(b): 运行前(BX)=9FF8H=1001 1111 1111 1000进行**或运算**后，符号位SF=1(负数)，不跳转

'-' -> DL

2号调用，将DL中的字符输出至屏幕上

对(BX)求补 -BX->BX;BX=0110 0000 0000 0111+1=0110 0000 0000 1000=6008H

输出：-6008H

六、分析以下程序段的功能

1、

```

1     MOV AH,10
2     LEA DX,L1 ;取L1的偏移地址->DX
3     INT 21H ;调用缓冲区多字符键盘输入 (L1+1存输入不含回车的字符数量,L1+2始存串)
4     LEA DX,L2 ;取L2的偏移地址->DX
5     INT 21H
6     MOV CL,L1+1 ;CL存入输入字符串的长度
7     CMP CL,L2+1 ;
8     JNZ N ;两者不相等跳转N
9     LEA SI,L1+2 ;L1+2处的偏移地址->SI
10    LEA DI,L2+2 ;L2+2处的偏移地址->DI
11    MOV CH,0 ;CH置0
12    CLD ;CLD即告诉程序SI, DI向前移动
13    REPZ CMPSE ;REPE/REPZ表示在CX不为0时, 且ZF=1重复执行后面的串处理指令SCAS或
    CMPS:
14    ;((SI)) - ((DI))两内存单元的内容相减根据比较结果置条件标志位: 相等
    ZF=1, 不等 ;ZF=0
15    JNZ N
16    MOV BX,0
17    JMP R
18 N: MOV BX,0FFFFH ;BX所有位置1
19 R: INC DI ;DI=DI+1

```

获取用户输入的一串字符，存在缓冲区L1中，与L2指向的程序原有字符串进行比较，若两者不完全相同，则置BX为0FFFFH，否则置BX为0

2、

```
1 DATA SEGMENT
2 BUF DB 'add ax,bx sub cx,10 mov 1234h END $'
3 DATA ENDS
4 CODE SEGMENT
5 ASSUME CS:CODE,DS:DATA,SS:STACK
6 BEGIN:
7     MOV AX,DATA
8     MOV DS,AX
9     LEA BX,BUF ;将BUF的EA->BX
10    MOV DL,[BX] ;将当前BUF的字节->DL
11    LOPA: CMP DL,'$' ;判断该字节数据是否是'$'
12          JE EXIT ;是'$'跳转至EXIT
13          CMP DL,'a' ;不再比较是不是'a'
14          JB N ;如果比a小跳转N
15          CMP DL,'z'
16          JA N ;比z大也跳转N
17          SUB DL,20H ;DL的字节数据-20H
18    N: MOV AH,2
19        INT 21H ;2号调用;显示输出DL的字符
20        INC BX ;BX=BX+1
21        JMP LOPA ;跳转回LOPA
22    EXIT: MOV AH,4CH ;DOS正常结束
23          INT 21H
24 CODE ENDS
25     END BEGIN
26
```

作用:将BUF指向的存储区存储的字符串的小写字母转换为大写，输出最后的英文字母全为大写形式的字符串

七、定义以下宏指令

1. 将X单元中的ASCII码转换为16进制数，结果送Y单元

```
1 ;要求ASCII码字母大写，字符集合{0-9, A-F}
2 TRANSFORM MACRO X,Y
3     PUSH DL
4     PUSH DI
5     MOV DI,0
6     LOPA: MOV DL,X[DI]
7           CMP DL,'$'
8           JE EXIT
9           CMP DL,'A'
10          JB N
11          SUB AH,1FH
12          JMP R
13    N: SUB AH,30H
14    R: MOV Y[DI],DL
15        INC DI
16        JMP LOPA
17    EXIT: POP DI
```

```
18 | POP DL
19 |
```

2. 将A缓冲区的N个字/字节送到B缓冲区

```
1  ;KIND==0传送字，KIND==1传送字节
2  DELIVERY MACRO A,B,N,KIND
3          PUSH AL
4          PUSH DI
5          PUSH BX
6          PUSH CL
7          XOR AL,AL
8          XOR DI,DI
9          CMP AL,KIND
10         JS L2
11         L1: CMP DI,N
12             JNL EXIT
13             MOV BX,A[DI+DI]
14             MOV B[DI+DI],BX
15             INC DI
16             JMP L1
17         L2: CMP DI,N
18             JNL EXIT
19             MOV CL,A[DI]
20             MOV B[DI],CL
21             INC DI
22             JMP L2
23     EXIT: POP CL
24           POP BX
25           POP DI
26           POP AL
```

八、定义以下子程序

1. 求N个字之和

```
1  SUM PROC
2      PUSH DI
3      PUSH BX
4      PUSH SI
5      XOR BX,BX
6      XOR AX,AX
7      XOR DX,DX
8      XOR DI,DI
9
10     L1: CMP DI,CX
11         JNL EXIT
12         MOV BX,[SI]
13         ADD SI,2
14         ADC AX,BX
15         JC L2
16         INC DI
17         JMP L1
18     L2: ADD DX,CF
```



```

19      INC DI
20      JMP L1
21  EXIT: POP SI
22      POP BX
23      POP DI
24  SUM ENDP

```

2.

在BUF缓冲区内造出有符号数的最大数

```

1  FINDMAX PROC
2      PUSH SI
3      PUSH BX
4      PUSH DI
5      XOR DI,DI
6      XOR BX,BX
7      MOV AX,[SI]
8      ADD SI,2
9      INC DI
10  L1:
11      CMP DI,CX
12      JNL EXIT
13      MOV BX,[SI]
14      ADD SI,2
15      INC DI
16      CMP AX,BX
17      JL L2
18      JMP L1
19  L2: MOV AX,BX
20      JMP L1
21  EXIT: POP DI
22      POP BX
23      POP SI

```