汇编语言（教材 王爽）期末考试复习：

考试题型：选择、填空、程序分析、编程题

一、 重点复习课本的检测点 1.1-3.2 与课后实验 1-14：

二、 需要掌握的指令见后面附录

三、汇编语言主要知识点：

寄存器与存储器

1. 寄存器功能

. 寄存器的一般用途和专用用途
. CS:IP 控制程序执行流程
. SS:SP 提供堆栈栈顶单元地址
. DS:BX(SI,DI) 提供数据段内单元地址
. SS:BP 提供堆栈内单元地址
. ES:BX(SI,DI) 提供附加段内单元地址
. AX,CX,BX 和 CX 寄存器多用于运算和暂存中间计算结果,但又专用于某些指令(查阅 指令表)。
. PSW 程序状态字寄存器只能通过专用指令（LAHF, SAHF)和堆栈(PUSHF,POPF)进行存取。

2. 存储器分段管理

. 解决了 16 位寄存器构成 20 位地址的问题
. 便于程序重定位
. 20 位物理地址=段地址 * 16 + 偏移地址
. 程序分段组织: 一般由代码段,堆栈段,数据段和附加段组成,不设置堆栈段时则使用系统内部的堆栈。

3. 堆栈

. 堆栈是一种先进后出的数据结构 ， 数据的存取在栈顶进行 ， 数据入栈使堆栈向地址减小的方向扩展。
. 堆栈常用于保存子程序调用和中断响应时的断点 以及暂存数据或中间计算结果 。
. 堆栈总是以字为单位存取

指令系统与寻址方式

1. 指令系统

. 计算机提供给用户使用的机器指令集称为指令系统,大多数指令为双操作数指令。执行指令后,一般源操作数不变,目的操作数被计算结果替代。
. 机器指令由 CPU 执行,完成某种运算或操作,8086/8088 指令系统中的指令分为 6 类: 数据传送,算术运算,逻辑运算,串操作,控制转移和处理机控制。

2. 寻址方式

. 寻址方式确定执行指令时获得操作数地址的方法
. 分为与数据有关的寻址方式(7 种)和与转移地址有关的寻址方式(4)种。

. 与数据有关的寻址方式的一般用途:
(1) 立即数寻址方式--将常量赋给寄存器或存储单元
(2) 直接寻址方式--存取单个变量 （直接给出地址值或变量名）
(3) 寄存器寻址方式--访问寄存器的速度快于访问存储单元的速度
(4) 寄存器间接寻址方式--访问数组元素
(5) 变址寻址方式
(6) 基址变址寻址方式 课本 P164
(7) 相对基址变址寻址方式
(5),(6),(7)都便于处理数组元素
. 与数据有关的寻址方式中,提供地址的寄存器只能是 BX,SI,DI 或 BP

. 与转移地址有关的寻址方式的一般用途:
(1) 段内直接寻址--段内直接转移或子程序调用
(2) 段内间接寻址--段内间接转移或子程序调用
(3) 段间直接寻址--段间直接转移或子程序调用
(4) 段间间接寻址--段间间接转移或子程序调用

汇编程序和汇编语言

1. 汇编程序
. 汇编程序是将汇编语言源程序翻译成二进制代码程序的语言处理程序,翻译的过程称为汇编。

2. 汇编语言
. 汇编语言是用指令助记符,各种标识变量,地址,过程等的标识符书写程序的语言, 汇编语言指令与机器指令一 一对应。
. 伪指令,宏指令不是由 CPU 执行的指令,而是由汇编程序在汇编期间处理的指令。
. 伪指令指示汇编程序如何完成数据定义,存储空间分配,组织段等工作。
. 宏指令可简化程序并减少程序书写量。 （不要求）
. 条件汇编伪指令的功能是确定是否汇编某段源程序 ,而不是实现程序分支,对未汇编的程序将不产生相应的目标代码。
. 结构作为一种数据结构可将一组类型不同但有逻辑关联的数据组织在一起,便于 整体处理数据。
. 记录可用于提高存储单元的利用率,将若干不足一个字节或字且有逻辑关联的信 息压缩存放在一个字节或字中。
. 指令中的表达式在汇编期间计算,并且只能对常量或地址进行计算。

程序设计基础

1. 分支程序设计
. 程序分支由条件转移指令或无条件转移指令实现
. 存放若干目的转移地址或跳转指令的跳转表常用于实现多路分支
. 条件转移指令只能实现偏移量为-128 至+127 字节范围的转移
. 无条件转移指令根据寻址方式可实现短转移(偏移量为-128 至+127 字节),段内转 移,段间转移。

2. 循环程序设计
. 可由循环控制指令或条件转移指令组织循环结构
. 内层循环结构必须完全包含在外层循环结构内,并不能发生从循环结构外向循环 结构内的转移。

3. 子程序设计
. 子程序中应保护寄存器内容,并正确使用堆栈, 成对执行 PUSH 和 POP 指令,保证执行 RET 指令时堆栈栈顶为返回地址。
. 主程序可通过寄存器,参数表,或堆栈传递参数给子程序

4. EXE 文件和 COM 文件 ?
. 二者都是可执行文件
. COM 文件源程序的特点是: 第一条可执行指令的起始存放地址必须是 100H,不能分段,不用定义堆栈,所有过程为 NEAR 类型,直接用 INT 20H 指令返回 DOS。

5. DOS 功能调用与 BIOS 中断调用
. 二者都是完成 DOS 系统提供给用户的输入/输出等常用功能,通过执行软中断指令 完成一次软中断服务。
. DOS 功能调用的中断服务程序是操作系统的一部分,存于 RAM 中; 而 BIOS 中断调用的中断服务程序存放在 ROM 中。

输入/输出与中断系统

1. 输入/输出的方式
. 程序直接 I/O 方式: 用 IN 和 OUT 指令直接在端口级上进行 I/O 操作,数据传送方式 分为无条件传送方式和查询传送方式。
. 中断传送方式: 由 CPU 响应中断请求完成中断服务。
. DMA 传送方式: 直接在存储器与外设之间传送数据。

2. 有关中断的概念
. 中断、中断源、中断请求、中断服务、中断向量、中断向量表、中断响应过程、中断指令、开中断、关中断、内部中断、外部中断、可屏蔽中断、非屏蔽中断。

四、程序编写及程序分析填空类的题目：(主要是书中的实验)

1、 编程，向内存 0:200~0:23f 依次传递数据 0~63(3fh)，程序中只能使用 9 条指令，9 条指中包括“mov ax,4c00h”和“int 21h”。
以及实验 4 的第 3 小题 将指令复制到 0：200 处

2、 编程，将 datasg 段中每个单词的前 4 个字母改写为大写字母。

```
assume cs:codesg,ss:stacksg,ds:datasg

stacksg segment

dw 0,0,0,0,0,0,0,0

stacksg ends

datasg segment

db '1. display '

db '2. brows '

db '3. replace '

db '4. modify '

datasg ends
```

3、 编程：在屏幕中间分别显示绿色、绿底红色、白底蓝色的字符串'welcome to masm!'。

4、 编写子程序：显示字符串

```
;名称：show_str
;功能：在屏幕的指定位置，用指定颜色，显示一个用 0 结尾的字符串
;参数：（dh）=行号，（dl）=列号（取值范围 0～80），（cl）=颜色，ds：si：该字符串的首地址
;返回：显示在屏幕上
```

5、 编写子程序：.解决除法溢出问题

```
;名称：divdw
;功能：除法，被除数 32 位，除数 16 位，商 32 位，余数 16 位，不会溢出
```

;参数：（dx）=被除数高 16 位，（ax）=被除数低 16 位，（cx）=除数
;返回：（dx）=商高 16 位，（ax）=商低 16 位，（cx）=余数

6、 数值显示

;名称：dtoc_word
;功能：将一个 word 型数转化为字符串
;参数：（ax）=word 型的数据，ds:si 指向字符串的首地址
;返回：ds:[si]放此字符串，以 0 结尾

7、 编写子程序

;名称：letterc
;功能：将以 0 结尾的字符串中的小写字母转变成大写字母
;参数：ds:si 开始存放的字符串
;返回：ds:si 开始存放的字符串

8、 编写 0 号中断的处理程序

编写 0 号中断的处理程序，使得在除法溢出发生时，在屏幕中间显示字符串“divide error!”,然后返回到 DOS。

五、参考习题

1 什么是堆栈操作？以下关于堆栈操作的指令执行后，SP 的值是多少？

```
PUSH  AX
PUSH  CX
PUSH  DX
POP   AX
PUSH  BX
POP   CX
POP   DX
```

堆栈被定义为一种先进后出的数据结构，即最后进栈的元素将被最先弹出来。堆栈从一个称为栈底的位置开始，数据进入堆栈的操作称为压入（或压栈），数据退出堆栈的操作称为弹出，每进行一次弹出操作，堆栈就减少一个元素，最后一次压入的元素，称为栈顶元素，压入弹出操作都是对栈顶元素进行的堆栈的两种基本的操作。

在进行以上一系列堆栈操作后，SP 指针的值是原 SP-2。

2 用汇编语言指令实现以下操作。

（1） 将寄存器 AX、BX 和 DX 的内容相加，和放在寄存器 DX 中。

```
ADD  AX，BX
ADD  DX，AX
```

（2） 用基址变址寻址方式（BX 和 SI）实现 AL 寄存器的内容和存储器单

元 BUF 中的一个字节相加的操作，和放到 AL 中。

```
mov bx, offset buf
mov si,0
ADD  AL，BYTE PTR [BX][SI]
```

（3） 用寄存器 BX 实现寄存器相对寻址方式（位移量为 100H），将 DX 的内容和存储单元中的一个字相加，和放到存储单元中。

```
ADD  100H[BX]，DX
```

（4） 用直接寻址方式（地址为 0500H）实现将存储器中的一个字与立即数 3ABCH 相加，和放回该存储单元中。

```
ADD  word ptr  [0500H]，3ABCH
```

（5） 用串操作指令实现将内存定义好的两个字节串 BUF1 和 BUF2 相加后，存放到另一个串 BUF3 中的功能。

```
……
MOV  CX，COUNT
MOV  SI，OFFSET BUF1      源串
MOV  DI，OFFSET BUF3      目的串
MOV  BX，OFFSET BUF2      辅助
AGAIN：LODSB
ADD  AL，[BX]
STOSB
INC  BX            注意没有和 KEP 配合，所以 CX - 1
DEC  CX
JNZ  AGAIN
……
```

3 指出下列指令中，源操作数及目的操作数的寻址方式。

```
（1）   SUB  BX，[BP+35]    ；寄存器寻址、寄存器相对寻址
（2）   MOV  AX，2030H    ；寄存器寻址、立即寻址      串扫描指令
（3）   SCASB              ；隐含操作数为寄存器寻址和寄存器间接寻址
（4）   IN  AL，40H         ；寄存器寻址、立即寻址
（5）   MOV  [DI+BX]，AX   ；基址加变址寻址、寄存器寻址
（6）   ADD  AX，50H[DI]    ；寄存器寻址、寄存器相对寻址
（7）   MOV  AL，[1300H]    ；寄存器寻址、直接寻址
（8）   MUL  BL             ；寄存器寻址、目的操作数为隐含寄存器寻址
```

4 已知（DS）= 1000H，（SI）= 0200H，（BX）= 0100H，（10100H）= 11H，（10101H）= 22H，（10600H）= 33H，（10601H）= 44H，（10300H）= 55H，（10301H）= 66H，（10302H）= 77H，（10303H）= 88H，试分析下列各条指令执行完后 AX 寄存器的内容。

```
（1）   MOV  AX，2500H          （AX）=2500H
（2）   MOV  AX，500H[BX]       （AX）==4433H   [BX+500H] = [0600H]
（3）   MOV  AX，[300H]          （AX）=6655H
（4）   MOV  AX，[BX]            （AX）=2211H
（5）   MOV  AX，[BX][SI]         （AX）=6655H
（6）   MOV  AX，[BX+SI+2]       （AX）=8877H
```

5 判断下列指令是否有错，如果有错，说明理由。

（1） SUB BL，BX ；两个操作数的宽度不一样

（2） MOV BYTE PTR[BX]，3456H ；将 16 位的立即数传送到一个字节的内存单元

（3） SHL AX，CH ；移位指令的移位位数用 CL 给出，不能用 CH。

（4） MOV AH，[SI][DI] ；不能用两个变址寄存器来实现寻址操作

（5） SHR AX，4 ；只有当移位位数为 1 时，才能用立即数表达

（6） MOV CS，BX ；不能对 CS 实现传送操作

（7） MOV 125，CL ；立即数不能做目的操作数

（8） MOV AX，BYTE PTR[SI] ；源操作数限定为字节，与目的操作数宽度不一致

（9） MOV [DI]，[SI] ；两个操作数不能同时为内存数

6 设（DS）= 1000H，（ES）= 2000H，（SS）= 3000H，（SI）= 0080H，（BX）= 02D0H，（BP）= 0060H，试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，0CBH 立即寻址 DS*16 + 100H

（2） MOV AX，[100H] 直接寻址，物理地址为：10100H

（3） MOV AX，[BX] 寄存器间接寻址，物理地址为：102D0H

（4） MOV AX，[BP] 寄存器间接寻址，物理地址为：30060H SS*16+BP

（5） MOV AX，[BP+50] 寄存器相对寻址，物理地址为：300B0H

（6） MOV AX，[BX][SI] 基址加变址寻址，物理地址为：10350H

7 分别说明下列每组指令中的两条指令的区别.

（1） AND CL，0FH 按位相“与”，高 4 位为“0000”，低 4 位保留原值；
OR CL，0FH 按位相“或”，高 4 位为原值，低 4 位为“1111”。

（2） MOV AX，BX 将 BX 寄存器的内容传送到 AX 寄存器中；
MOV AX,[BX] 将 BX 寄存器所指的内存单元的内容送 AX 寄存器中。

（3） SUB BX，CX BX 寄存器内容减去 CX 寄存器的内容，结果送回到 BX；
CMP BX，CX BX 内容减去 CX 内容，但结果不送回，而根据标志位的情况做进一步的动作。

（4） AND AL，01H AL 内容与 01H 相“与”，结果为“0000000x”送回 AL 寄存器；
TEST AL，01H AL 内容与 01H 相“与”，结果为“0000000x”不送回 AL 寄存器，而根据标志位（ZF）情况做进一步的动作。

（5） JMP NEAR PTR NEXT NEXT 所指指令在当前指令的同段内（16 位地址范围）；
JMP SHORT NEXT NEXT 所指指令在当前指令的 8 位地址范围内。

（6） ROL AX，CL 循环左移，进位标志位不参与循环；
RCL AX，CL 循环左移，进位标志位参与循环。

（7） PUSH AX 将 AX 内容存入栈顶指针处，即进栈操作；
POP AX 将栈顶内容弹出装入 AX 寄存器中，即出栈操作。

8　试分析以下程序段执行完后 BX 的内容为何？

```
MOV  BX，1030H
MOV  CL，3
SHL  BX，CL
DEC  BX
```

程序执行完后，BX=817FH，执行过程如下。

9、在 BUF1 变量中依次存储了 5 个字数据，接着定义了一个名为 BUF2 的字单元，表示如下：

```
BUF1   DW  8765H，6CH，0，1AB5H，47EAH
BUF2   DW  ?
```

（1）　设 BX 中是 BUF1 的首地址，请编写指令将数据 50H 传送给 BUF2 单元。

```
ADD  BX，10
MOV  WORD PTR[BX+10]，50H
```

（2）　请编写指令将数据 FFH 传送给数据为 0 的单元。

```
ADD  BX，4
MOV  WORD PTR[BX+10]，0FFH
```

10、下面是一个数据段的定义，请用图表示它们在内存中存放的形式。

```
DATA  SEGMENT
A1    DB    25H，35H，45H
A2    DB    3 DUP（5）
A3    DW    200，3AB6H
A4    DW    3000H，6A6FH
DATA  ENDS
```

A1	25H
	35H
	45H
A2	05H
	05H
	05H
A3	00H
	02H
	B6H
	3AH
A4	00H
	30H
	6FH
	6AH

11、说明下列语句所分配的存储空间及初始化的数据值。

（1）BYTE_VAR DB ‘BYTE’，21，-42H，3 DUP(0，?，2 DUP(2，3)，?)

（2）WORD_VAR DW 5 DUP（4，2，0），？，-8，‘BY’，‘TE’ 256H

BYTE_VAR	42H	WORD_VAR	04H

59H
54H
45H
15H
10111110B
0
—
2
3
2
3
—
0
—
2
3
2
3
—
0
—
2
3
2
3
—

00H
02H
00H
00H
00H
04H
00H
02H
00H
00H
00H
04H
00H
02H
00H
00H
00H
04H
00H
02H
00H
00H
00H
04H
00H
02H
00H
00H
00H
—
—
11111000B
11111111B
42H
59H
54H
45H
56H
02H

12、在下列数据传送程序段中有些使用不当的语句，请改正之。

A　　DB　　10H，20H，‘OPQ’，4FH

B　　DB　　N DUP（？）　　；改为 B　　DB　 6 DUP（？）

```
        MOV  DI，A    ；改为 MOV  DI，OFFSET A
        MOV  SI，B    ；改为 MOV  SI，OFFSET B
        MOV  CX，LENGTH A  ；改为 MOV  CX，LENGTH B
    CC：  MOV  AL，[DI]
        MOV  [SI]，AL
        INC  SI
        INC  DI
        LOOP  CC
```

13、有一个数据段定义了如下 6 个变量，请写出该数据段。

（1）BUF1 为十进制数字节变量：64；

（2）BUF2 为字符串变量：‘Teacher’；

（3）BUF3 为十六进制数字节变量：2FH；

（4）BUF4 为双字变量：657AH；

（5）BUF5 为字变量：657AH；

（6）BUF6 为二进制数字节变量：10101101B。

```
DATA  SEGMENT
BUF1  DB 64
BUF2  DB  ‘Teacher’
BUF3  DB 2FH
BUF4  DD 657AH
BUF5  DW 657AH
BUF6  DB 10101101B
```

14 以下为用段基址:偏移量形式表示的内存地址，试计算它们的物理地址。

⑴ 12F8:0100 ⑵ 1A2F:0103 ⑶1A3F:0003 ⑷ 1A3F:A1FF

15 自 12FA:0000 开始的内存单元中存放以下数据(用十六进制形式表示)：03 06 11 A3 13 01，试分别写出 12FA:0002 的字节型数据、字型数据及双字型数据的值。

16 分别指出下列指令中源操作数和目标操作数的寻址方式。

⑴ MOV BX,12

⑵ MOV AL,128

⑶ MOV [BX],DX

⑷ MOV DS,AX

⑸ MOV VAR,8

⑹ MOV [1000H],DX

⑺ MOV 6[BX],CX

⑻ MOV AX,[BX][SI]

⑼ MOV TAB[BP][DI],AL

17 在 80X86 系统中，设(DS)＝1000H，(ES)＝2000H，(SS)＝1200H，(BX)＝0300H，(SI)＝0200H，(BP)＝0100H，VAR 的偏移量为 0060H，若目标操作数为存储器操作数，请计算目标操作数的物理地址是多少？

⑴ MOV BX,12　　⑵ MOV AL,128

⑶ MOV [BX],DX　　⑷ MOV ES:[SI],AX

⑸ MOV VAR,8　　⑹ MOV [1000H],DX
⑺ MOV 6[BX],CX　　⑻ MOV [BX][SI],AX
⑼ MOV 6[BP][SI],AL

18　指出下指令的不同：
⑴ MOV AX,3000H 与 MOV AX,[3000H]
⑵ MOV AX,MEM 与 MOV AX,OFFSET MEM
⑶ MOV AX,MEM 与 LEA AX,MEM

19 指出下列指令的错误：
⑴ MOV [AX],BX　　⑵ MOV AL,1280
⑶ MOV [BX],9　　⑷ MOV DS,1000H
⑸ MOV VAR,[BX]　　⑹ MOV M1,M2
⑺ MOV 6,CX ⑻ MOV AX,[SI][DI]
⑼ MOV CS,AX ⑽ MOV BX,OFFSET VAR[SI]

20 自 BUFFER 单元开始连续存放着两个字型数据，编程序求它们的和，并把结果存放在这两个数据之后。

21 写出把首址为 BUF 的字型数组的第 4 个字送到 AX 寄存器的指令，要求使用以下几种寻址方式：
⑴ 直接寻址方式
⑵ 使用 BX 的寄存器间接寻址方式
⑶ 使用 BX 的寄存器相对寻址方式

22 设(DS)＝1000H，(BX)＝0300H，(SI)＝0002H，(DI)＝0100H，自 1000:0300 单元开始存有以下数据（用十六进制形式表示）：12 34 56 78 90 AB CD EF 试说明下列各条指令执行后目标操作数的内容。
⑴ ADD BX,12 ⑵ MOV DX,[0300H]
⑶ SUB BYTE PTR [BX],8 ⑷ MOV AX,[BX][SI]
⑸ MOV CX,5[BX] ⑹ MOV DX,4[BX][SI]

23 编程序使：
⑴ 使 AL 的各位变反。
⑵ BX 寄存器低四位置 1。
⑶ AX 寄存器的低四位清 0。
⑷ 使 CX 寄存器的低四位变反。
⑸ 用 TEST 指令测试 AL 寄存器的位 0 和位 6 是否同时为 0，若是则把 0 送 DH 寄存器，否则把 1 送 DH 寄存器。

24.　下列语句各为变量分配了多少字节？
⑴ N1 DB 5
⑵ N2 DB 123
⑶ N3 DB　‘123’
⑷ N4 DB 80,?,80 DUP (0)
⑸ N5 DW 1,2,3
⑹ N6 DD 6,100
⑺ N7 DD N3
⑻ N8 DW N4
⑼ N9 DW N4+2

25.　对于第 12 题，若 N1 的地址为 1470:0000，其后的变量依次连续存放，试

给出各变量的偏移量。N7~N9 变量的值是多少？

14. 如果定义了如第 12 题所示的变量，下列指令执行的结果是什么(若语句正确，则给出执行结果；若不正确，说明错误原因)？

⑴ ADD N1,10
⑵ MOV AL,N2
⑶ SUB N3,N2
⑷ MOV AL,N3+2
⑸ LEA DX,N4+2
⑹ XOR N5,0FH
⑺ MOV N4+1,CX
⑻ AND N1,01234H
⑼ NUM1 EQU 89
ADD NUM1,10
⑽ NUM2 EQU 100
NUM2 EQU 120

26. 假设有下列数据定义语句：

⑴ M1 DB 10
⑵ M2 DW 2345H
⑶ M3 DD 987865H

试写出汇编程序对这些语句汇编后所生成的值（例如 M2 单元、M2＋1 单元的内容)。

27. 有以下程序片段，试问汇编后符号 L1 和 L2 的值各为多少？

BUF1 DB 1,2,3
BUF2 DW 5，6，7
L1 EQU $－BUF2
L2 EQU BUF2－BUF1

28. 若 AX=0ABCDH ， BX=7F8FH ， CF=1 。 求分别执行 8086 CPU 指令

⑴ ADD AX,BX
⑵ ADC AX,BX
⑶ SBB AX,BX
⑷ NEG AX
⑸ AND AX,BX
⑹ OR AX,BX
⑺ XOR AX, BX
⑻ IMUL BL

后， AX 寄存器中的内容，

29. 指出下列指令中哪些是错误的，并指出错在何处？

⑴ MOV DL,[DX]
⑵ MOV ES,2000H
⑶ SUB [BX],[SI]
⑷ ADD AX,[BX+CX]
⑸ XCHG DS,[2400H]

(6) DEC 15H
(7) IN AL,DX
(8) OUT 300H,AX
解答
(1) 错， DX 不能作为间接寻址的寄存器。
(2) 错， 立即数不能直接送给段寄存器。
(3) 错， 两个存储单元间内容不能直接相减。
(4) 错，CX 寄存器不能作为变址寄存器。
(5) 错， 不允许段寄存器与存储单元内容进行交换
(6) 错， 减 1 指令不能对段寄存器直接操作
(7) 对
(8) 错， 输出指令的地址若是 16 位， 必须用 DX 间址。

30. 下列程序段执行后 AX=?
设数据段有：TABLE DW 100,200,300,400
ENTRY DW 3
代码段 对 DS 初始化
```
MOV BX,OFFSET TABLE
MOV  SI,ENTRY
MOV AX,[BX+SI]
```

31. 若 SS=1000H ， SP=2000H ， AX=1234H ， BX=5678H ， FR=2115 ， 试说明执行指令

```
PUSH  AX
PUSH  BX
PUSHF
POP CX
POP  DX
```

之后， SP=? SS=? CX=? DX=? 并画图指出堆栈中各单元的内容。
解答： SS=1000H
SP=1FFEH
CX=2115H
DX=5678H
栈中各单元内容如下所示：

地址	RAM
11FFAH	2115
11FFCH	5678
11FFEH	1234
12000H	

32、读下列程序段：写出； (1)每条指令连续执行后，哪些寄存器发生变化?内容是什么? (2)程序段执行完后，BX 寄存器的肉容是什么?

① MOV BX，3F93H

② MOV CL ，4

③ ADD BX ，CL

④ MOV AL ，BL

⑤ AND AL ，0FH

⑥ OR AL ，30H

六 附录：汇编指令

红色为需要掌握的汇编指令：

一、数据传输指令

它们在存贮器和寄存器、寄存器和输入输出端口之间传送数据.

1. 通用数据传送指令.

MOV 传送字或字节.

MOVSX 先符号扩展,再传送.

MOVZX 先零扩展,再传送.

PUSH 把字压入堆栈.

POP 把字弹出堆栈.

PUSHA 把 AX,CX,DX,BX,SP,BP,SI,DI 依次压入堆栈.

POPA 把 DI,SI,BP,SP,BX,DX,CX,AX 依次弹出堆栈.

PUSHAD 把 EAX,ECX,EDX,EBX,ESP,EBP,ESI,EDI 依次压入堆栈.

POPAD 把 EDI,ESI,EBP,ESP,EBX,EDX,ECX,EAX 依次弹出堆栈.

BSWAP 交换 32 位寄存器里字节的顺序

XCHG 交换字或字节.(至少有一个操作数为寄存器,段寄存器不可作为操作数)

CMPXCHG 比较并交换操作数.(第二个操作数必须为累加器 AL/AX/EAX)

XADD 先交换再累加.(结果在第一个操作数里)

XLAT 字节查表转换.

—— BX 指向一张 256 字节的表的起点，AL 为表的索引值 (0-255,即 0-FFH)；返回 AL 为查表结果. ([BX+AL]->AL)

2. 输入输出端口传送指令.

IN I/O 端口输入. (语法: IN 累加器, {端口号 | DX})

OUT I/O 端口输出. (语法: OUT {端口号 | DX},累加器)

输入输出端口由立即方式指定时，其范围是 0-255；由寄存器 DX 指定时，

其范围是 0-65535.
3. 目的地址传送指令.
LEA 装入有效地址.
例: LEA DX,string ;把偏移地址存到 DX.
LDS 传送目标指针,把指针内容装入 DS.
例: LDS SI,string ;把段地址:偏移地址存到 DS:SI.
LES 传送目标指针,把指针内容装入 ES.
例: LES DI,string ;把段地址:偏移地址存到 ES:DI.
LFS 传送目标指针,把指针内容装入 FS.
例: LFS DI,string ;把段地址:偏移地址存到 FS:DI.
LGS 传送目标指针,把指针内容装入 GS.
例: LGS DI,string ;把段地址:偏移地址存到 GS:DI.
LSS 传送目标指针,把指针内容装入 SS.
例: LSS DI,string ;把段地址:偏移地址存到 SS:DI.
4. 标志传送指令.
LAHF 标志寄存器传送,把标志装入 AH.
SAHF 标志寄存器传送,把 AH 内容装入标志寄存器.
PUSHF 标志入栈.
POPF 标志出栈.
PUSHD 32 位标志入栈.
POPD 32 位标志出栈.

二、算术运算指令

ADD 加法.
ADC 带进位加法.
INC 加 1.
AAA 加法的 ASCII 码调整.
DAA 加法的十进制调整.
SUB 减法.
SBB 带借位减法.
DEC 减 1.
NEC 求反(以 0 减之).
CMP 比较.(两操作数作减法,仅修改标志位,不回送结果).
AAS 减法的 ASCII 码调整.
DAS 减法的十进制调整.
MUL 无符号乘法.
IMUL 整数乘法.
以上两条,结果回送 AH 和 AL(字节运算),或 DX 和 AX(字运算),
AAM 乘法的 ASCII 码调整.
DIV 无符号除法.
IDIV 整数除法.
以上两条,结果回送:

商回送 AL,余数回送 AH, (字节运算);
或 商回送 AX,余数回送 DX, (字运算).
AAD 除法的 ASCII 码调整.
CBW 字节转换为字. (把 AL 中字节的符号扩展到 AH 中去)
CWD 字转换为双字. (把 AX 中的字的符号扩展到 DX 中去)
CWDE 字转换为双字. (把 AX 中的字符号扩展到 EAX 中去)
CDQ 双字扩展. (把 EAX 中的字的符号扩展到 EDX 中去)

三、逻辑运算指令

AND 与运算.
OR 或运算.
XOR 异或运算.
NOT 取反.
TEST 测试.(两操作数作与运算,仅修改标志位,不回送结果).
SHL 逻辑左移.
SAL 算术左移.(=SHL)
SHR 逻辑右移.
SAR 算术右移.(=SHR)
ROL 循环左移.
ROR 循环右移.
RCL 通过进位的循环左移.
RCR 通过进位的循环右移.
以上八种移位指令,其移位次数可达 255 次.
移位一次时, 可直接用操作码. 如 SHL AX,1.
移位>1 次时, 则由寄存器 CL 给出移位次数.
如 MOV CL,04
SHL AX,CL

四、串指令

DS:SI 源串段寄存器 :源串变址.
ES:DI 目标串段寄存器:目标串变址.
CX 重复次数计数器.
AL/AX 扫描值.
D 标志 0 表示重复操作中 SI 和 DI 应自动增量; 1 表示应自动减量.
Z 标志 用来控制扫描或比较操作的结束.
MOVS 串传送.
(MOVSB 传送字符. MOVSW 传送字. MOVSD 传送双字.)
CMPS 串比较.
(CMPSB 比较字符. CMPSW 比较字.)
SCAS 串扫描.

把 AL 或 AX 的内容与目标串作比较,比较结果反映在标志位.
LODS 装入串.
把源串中的元素(字或字节)逐一装入 AL 或 AX 中.
(LODSB 传送字符. LODSW 传送字. LODSD 传送双字.)
STOS 保存串.
是 LODS 的逆过程.
REP 当 CX/ECX<>0 时重复.
REPE/REPZ 当 ZF=1 或比较结果相等,且 CX/ECX<>0 时重复.
REPNE/REPNZ 当 ZF=0 或比较结果不相等,且 CX/ECX<>0 时重复.
REPC 当 CF=1 且 CX/ECX<>0 时重复.
REPNC 当 CF=0 且 CX/ECX<>0 时重复.

五、程序转移指令

1>无条件转移指令 (长转移)
JMP 无条件转移指令
CALL 过程调用
RET/RETF 过程返回.
2>条件转移指令 (短转移,-128 到+127 的距离内)
(当且仅当(SF XOR OF)=1 时,OP1
JA/JNBE 不小于或不等于时转移.
JAE/JNB 大于或等于转移.
JB/JNAE 小于转移.
JBE/JNA 小于或等于转移.
以上四条,测试无符号整数运算的结果(标志 C 和 Z).
JG/JNLE 大于转移.
JGE/JNL 大于或等于转移.
JL/JNGE 小于转移.
JLE/JNG 小于或等于转移.
以上四条,测试带符号整数运算的结果(标志 S,O 和 Z).
JE/JZ 等于转移.
JNE/JNZ 不等于时转移.
JC 有进位时转移.
JNC 无进位时转移.
JNO 不溢出时转移.
JNP/JPO 奇偶性为奇数时转移.
JNS 符号位为 "0" 时转移.
JO 溢出转移.
JP/JPE 奇偶性为偶数时转移.
JS 符号位为 "1" 时转移.
3>循环控制指令(短转移)
LOOP CX 不为零时循环.
LOOPE/LOOPZ CX 不为零且标志 Z=1 时循环.

LOOPNE/LOOPNZ CX 不为零且标志 Z=0 时循环.
JCXZ CX 为零时转移.
JECXZ ECX 为零时转移.
4>中断指令
INT 中断指令
INTO 溢出中断
IRET 中断返回
5>处理器控制指令
HLT 处理器暂停，直到出现中断或复位信号才继续.
WAIT 当芯片引线 TEST 为高电平时使 CPU 进入等待状态.
ESC 转换到外处理器.
LOCK 封锁总线.
NOP 空操作.
STC 置进位标志位.
CLC 清进位标志位.
CMC 进位标志取反.
STD 置方向标志位.
CLD 清方向标志位.
STI 置中断允许位.
CLI 清中断允许位.

六、伪指令

DW 定义字(2 字节).
PROC 定义过程.
ENDP 过程结束.
SEGMENT 定义段.
ASSUME 建立段寄存器寻址.
ENDS 段结束.
END 程序结束