东南大学

《微机系统与接口实验》

实验报告

实验一　数据传送

姓　　名：薛宇飞　　**学　　号**：04020235

同　　组：　　**学　　号**：

专　　业：信息工程　　**实 验 室**：金智楼硬件实验室

实验时间：2022 年 4 月 13 日　　**报告时间**：2022 年 4 月 13 日

评定成绩：　　**评阅教师**：裴文江

目录

一. 实验目的与内容

1. 熟悉 8086 指令系统的数据传送指令，进一步掌握传送指令的寻址方式；
2. 结合实验教材[book, guide]，利用 Turbo Debugger（TD）调试工具来调试汇编程序。

二. 实验任务

实验全部资料及完整代码详见薛宇飞的 GitHub 主页[mygit]。

(一) 执行代码段，熟悉 TD 的使用

程序段如下：

code

```
MOV BL,08H
MOV CL,BL
MOV AX,03FFH
MOV BX,AX
MOV DS:[0020],BX
```

实验结果如图 ??。

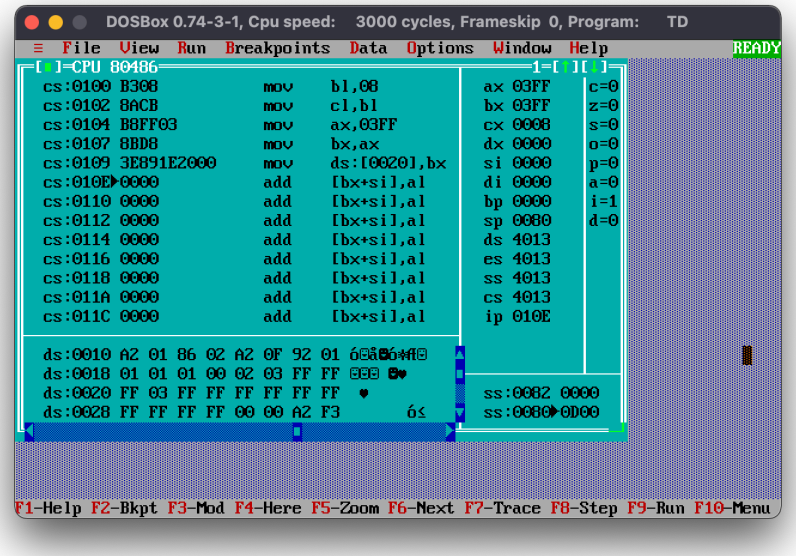

图 1: 实验结果

(二) 观察不同出栈方式的数据变化

预备程序段如下：

pre-code

```
MOV AX,0102H
MOV BX,0304H
MOV CX,0506H
MOV DX,0708H
PUSH AX
PUSH BX
PUSH CX
PUSH DX
```

第一种出栈方式：

way1

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

第二种出栈方式：

way2

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

第三种出栈方式：

way3

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

实验结果参见表 **??**。

注意 1: 验证要求

在验证不同的出栈方式之前，需要先执行预备程序段进行初始化。

(三) 指出并更正题目中的错误

具体过程详见表 **??**

表 1: 实验结果表格

	方式 1	方式二	方式三
AX	0102	0708	0304
BX	0304	0506	0102
CX	0506	0304	0708
DX	0708	0102	0506

表 2: 实验任务三

错误指令	错误类型	更正（不唯一）
MOV [BX],[SI]	目标和源操作数不能同时为寄存器	MOV BX,[SI]
MOV AH,BX	源操作数和目标操作数的字长不一致	MOV AX,BX
MOV AX,[SI][DI]	SI,DI 不能同时出现	MOV AX,[BX][DI]
MOV BYTE PTR [BX],2000H	源操作数和目标操作数的字长不一致	MOV WORD PTR [BX],2000H
MOV CS,AX	目标不能是 CS	MOV BX,AX
MOV DS,2000H	立即数不能直接送到段寄存器	MOV BX,2000H

(四) 设置寄存器的内容并验证

BX=0010H，SI=0001H，DS:[0010]=12H，DS:[0011]=34H，DS:[0012]=56H，DS:[0013]=78H，DS:[0120]=0ABH，DS:[0121]=0CDH，DS:[0122]=0EFH。

请单步执行表 ?? 各条指令，写出指令执行后 AX 寄存器的值。指令中如有存储器操作数，请计算、写出存储器操作数的偏移地址和逻辑地址，并查看、写出对应存储单元的值。

表 3: 实验任务四

指令	AX	偏移地址/逻辑地址/值
MOV AX,1200H	1200	-
MOV AX,BX	0010	-
MOV AX,[0120]	CDAB	0120/4013*16+0120/AB
MOV AX,[BX]	3412	0010/4013*16+0010/12
MOV AX,0110[BX]	CDAB	0010+0110/4013*16+0010+0110/AB
MOV AX,[BX][SI]	5634	0010+0001/4013*16+0010+0001/34
MOV AX,0110[BX][SI]	EFCD	01110+0010+0001/4013*16+0110+0010+0001/CD

[1] 本实验中默认的 DS 为 4013。

注意 2: 数据传送

目标操作数是 2 字节，读取地址指向的 2 字节，遵循“高位高地址，低位低地址”原则进行数据传送。

具体实验界面见图 ??。

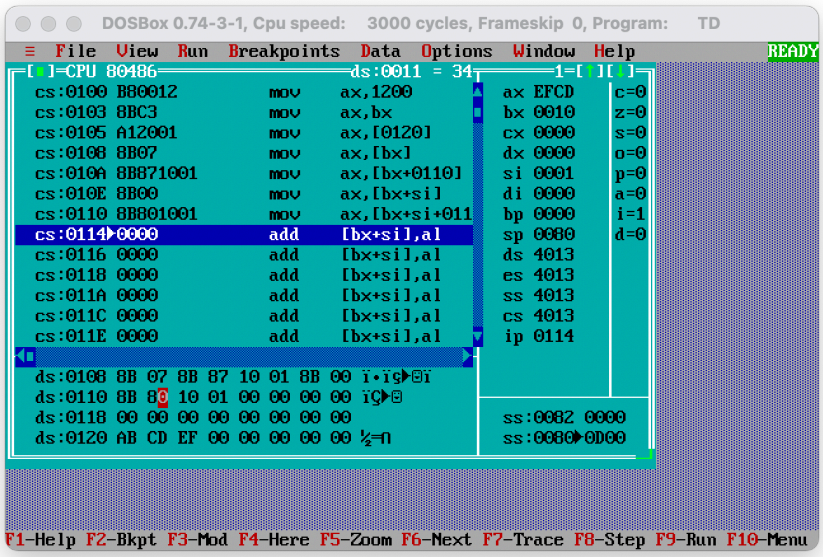

图 2: 实验四界面

(五) 不同寻址方式传递数据

实验前先检查寄存器情况:DS=4013H, BX=0000H, SI=0000H;初始化数据段情况:DS:[1000H]=FF, DS:[2020H]=00。传递数据的代码段如下:

code

```
% 直接寻址
MOV AL,[1000H]
MOV [2020H],AL
% 基址寻址
MOV AH,1000[BX]
MOV [2020H],AH
% 变址寻址
MOV AH,0100[SI]
MOV [2020H],AH
```

经过上机验证，全部正确。

注意 3: 语法错误

输入指令 MOV [2020H],[0100H] 会出现"Syntax error"，原因是 MOV 指令的源、目标操作数不能同时为储存器，需要借助寄存器间接传递。

(六) 数据的交换

设 AX 寄存器中的内容为 1111H,BX 寄存器中的内容为 2222H,DS:0010H 单元中的内容为 3333H。将 AX 和 BX 寄存器中内容进行交换，然后再将 BX 寄存器中的内容和 DS:0010H 单元中的内容进行交换。

code

```
% AX与BX交换
XOR AX,BX
XOR BX,AX
XOR AX,BX
% BX与储存器交换
MOV CX,BX
MOV BX,[0100H]
MOV [0100H],CX
```

分析 1: 数据交换方式

数据交换有多种方法:

1. 经典使用变量来进行交换;
2. 通过三次异或来交换;

   思考 1: 异或特点

   (a) 优点:
   节省变量成本;

   (b) 缺点:
   会增加读写次数，使得程序运行时间较长;
   会影响标志位。

3. 直接使用指令 XCHG 来交换数据，需要注意其中一个操作数必须在寄存器中!
4. 可以利用命令 PUSH, POP 来进行交换，但需要将两个数据放在相邻的字节中（原因：两指令只能是字操作)。
5. 除此之外还可以利用一些花哨的方法交换数据，但没意义。所谓“技巧”，只会让程序运行的更慢!

(七) 传送数据段

设 DS=1000H，ES=2000H，有关存储器的内容见要求，将 DS 段的内容传送到 AX 寄存器；ES 段的内容传送到 BX 寄存器，编写程序段。

代码指令如下

code

```
MOV WORD PTR AX,[0010]
MOV WORD PTR BX,ES:[0020]
```

实验结果见图 ??。

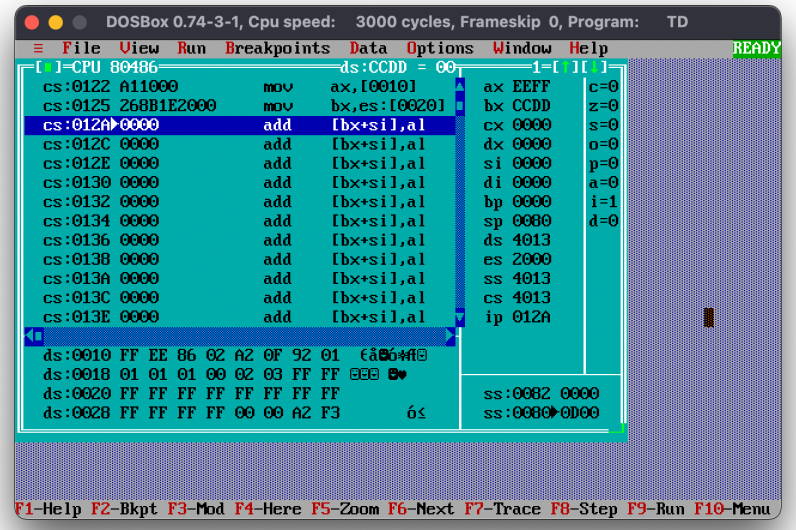

图 3: 实验结果

三. TD 使用方法小结

本课程实验中所需的 DOSBox 等软件搭建于 Mac OS 系统中，基本操作以及快捷键与 Windows 大致相同，不再赘述。全部文件已上传至我的 GitHub 主页[mygit]。

以下是符合个人喜好的个性化操作：

1. Mac OS 系统下的文件路径较于 Windows 系统较为复杂，因为实验前的自动挂载十分重要。具体方法为：通过 shift+cmd+. 找到根目录下/Library/Preferences/DOSBox 0.74-3-1 Preferences 的默认文件，在文件最后插入如下代码即可完成自动挂载：

code

```
mount c ~/DOSBox (your path)
path = path; c:\masm5
c:
```

2. 实验报告采用 LaTeX 模板进行编写，快捷键的设置也十分重要。在 vscode 中，使用组合键 fn+F1 搜索 keybingdings.json，进入可以添加快捷键，极大的提高了实验报告撰写效率。

3. 关于实验报告，所有的目录、引用以及交叉引用全部赋予 `Hyperlink`，方便快速定位查找，提高了回溯报告的效率。

四. 实验总结

实验总结已随文附在“注意”、“思考”、“分析”中。