

四川大学计算机学院、软件学院
实 验 报 告

学号：____2021141460301____ 姓名：____杨怡____ 专业：____计算机科学与技术____
____ 班级：____行政六班____ 第 12 周

课程名称	汇编语言程序设计课程设计	实验课时	4
实验项目	DEBUG 常用命令简介	实验时间	2022. 5. 11
实验目的	1、熟练掌握 DEBUG 的基本调试命令，能够使用 DEBUG 编写、调试汇编语言程序片段。 2、在理解数据传输指令的基础上按照实验内容中指定的程序片段对程序进行调试和记录		
实验环境	dosbox		
实验内容（算法、程序、步骤和方法）	<p>一、三个汇编指令</p> <p>1. 断点调试命令 G</p> <p>功能：从当前 CS、IP 指示的位置开始连续执行程序，执行到指定断点处（不包括断点地址指示的指令）暂停执行，在屏幕上显示 CPU 现场。</p> <p>使用方法： -g 连续执行，无断点</p> <p style="padding-left: 40px;">-g XXXX:XXXX 连续执行到断点位置</p> <p style="padding-left: 40px;">注：断点位置必须指定为某条指令的首字节地址</p> <p>2. 内存单元查看命令 D</p> <p>功能：查看指定地址开始的内存区域中各内存单元的内容。</p> <p>使用方法：</p> <p style="padding-left: 20px;">-D 从默认数据段内存地址开始显示（或接上次所显示的地址继续显示）</p> <p style="padding-left: 20px;">-D XXXX: XXXX 从指定内存地址开始显示</p>		

3. 内存单元修改命令 E

功能：修改指定地址内存单元的内容。

使用方法：

-E XXXX: XXXX

二、源代码

```
MOV WORD PTR [0100H], 5564H
```

```
MOV WORD PTR [0102H], 0C279H
```

```
MOV AX, [0100H]
```

```
PUSH AX
```

```
ADD AL, [0102H]//说明标志位是如何确定的
```

```
ADC AH, [0103H]//说明指令执行原理，包括标志位
```

```
MOV [0104H], AX
```

```
POP AX
```

```
SUB AL, [0102H]//说明标志位是如何确定的
```

```
SBB AH, [0103H]//说明指令执行原理，包括标志位
```

```
MOV [0106H], AX //说明程序执行完毕后，字单元[0104H]，[0106H]中的内容。
```

- 1、调试上述程序片段时，使用 r 命令先修改 DS、SS 段寄存器，使之与 CS 指向不同位置。将 DS 提供的原始段基值加 0100H, 将 SS 提供的原始段基值加 0200H，避免数据段、堆栈段内的操作覆盖代码段中已录入的指令，影响重复调试。

```
Z:\>c:
C:\>debug
-rds
DS 073F
:003f
-rss
SS 073F
:093f
```

- 2、进入 debug, 使用 a 命令输入本次实验代码，再使用 r 命令查看寄存器的值。

```
DOSBox 0.74, Cpu speed: 3000 cycles, Fra...
Z:\>c:
C:\>debug
-rds
DS 073F
:003f
-rss
SS 073F
:093f
-a
073F:0100 mov word ptr [0100],5564
073F:0106 mov word ptr [0102],0c279
073F:0106 mov word ptr [0102],c279 ^ Error
073F:010C mov ax,[0100]
073F:010F push ax
073F:0110 add al,[0102]
073F:0114 adc ah,[0103]
073F:0118 mov [0104]
073F:0118 mov [0104],ax ^ Error
073F:011B pop ax
073F:011C sub al,[0102]
073F:0120 sbb ah,[0103]
073F:0124 mov [0106],axS_
```

- 3、使用 t 指令单步调试运行程序。

```

DOSBox 0.74, Cpu speed: 3000 cycles, Fra...
AX=0000 BX=0000 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=003F ES=073F SS=093F CS=073F IP=0106 NU UP EI PL NZ NA PO NC
073F:0106 C706020179C2 MOV WORD PTR [0102],C279 DS:0102=0000
-t
AX=0000 BX=0000 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=003F ES=073F SS=093F CS=073F IP=010C NU UP EI PL NZ NA PO NC
073F:010C A10001 MOV AX,[0100] DS:0100=5564
-t
AX=5564 BX=0000 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=003F ES=073F SS=093F CS=073F IP=010F NU UP EI PL NZ NA PO NC
073F:010F 50 PUSH AX
-t
AX=5564 BX=0000 CX=0000 DX=0000 SP=00FB BP=0000 SI=0000 DI=0000
DS=003F ES=073F SS=093F CS=073F IP=0110 NU UP EI PL NZ NA PO NC
073F:0110 02060201 ADD AL,[0102] DS:0102=79
-t
AX=55DD BX=0000 CX=0000 DX=0000 SP=00FB BP=0000 SI=0000 DI=0000
DS=003F ES=073F SS=093F CS=073F IP=0114 OU UP EI NG NZ NA PE NC
073F:0114 12260301 ADC AH,[0103] DS:0103=C2
-SS_

```

4、程序分析：

（1）MOV WORD PTR [0100H]，5564H：将数据 5564H 存入 WORD PTR[0100H] 所指向的内存单元中

（2）MOV WORD PTR [0102H]，0C279H：将数据 0C279H 存入 WORD PTR[0102H] 所指向的内存单元中

（3）MOV AX，[0100H]：将[0100H]指向的内存单元中的数据存入 AX 中

（4）PUSH AX：将 AX 的数据压入堆栈中

（5）ADD AL，[0102H]//说明标志位是如何确定的：将地址[0102H]的数据（79H）与 AL 的数据（64H）相加并存入 AL 中。

$$79H+64H=01111001+01100100=11011101$$

CF=0，第一位未产生进位；

OF=1，产生溢出；

ZF=0，结果不为 0；

PF=1，低 8 位有偶数个 0；

AF=0，第三位未产生进位；

	<p>SF=1, 符号位为 1;</p> <p>(6) ADC AH, [0103H]//说明指令执行原理, 包括标志位: 将地址[0103H]的数据(0C2H)与AH的数据(55H)相加, 再加上CF中的数据, 结果存入AH中。$11000010+01010101+0=100010111$</p> <p>CF=1, 产生进位;</p> <p>OF=0, 正数加负数没有溢出;</p> <p>ZF=0, 结果不为 0;</p> <p>PF=1, 低 8 位有偶数个 0;</p> <p>AF=0, 第三位未产生进位;</p> <p>SF=0, 符号位为 0;</p> <p>(7) MOV [0104H], AX: 将 AX 中的值移入地址为 0104H 的内存中</p> <p>(8) POP AX: 将栈顶的数据取出, 存放到 AX 中</p> <p>(9) SUB AL, [0102H] //说明标志位是如何确定的: 将 AL 中的数据减去地址为 0102H 处的数据, 并把结果存入 AL 中</p> <p>$01100100-01111001=11101000$</p> <p>CF=1, 第一位产生借位;</p> <p>OF=0, 未产生溢出;</p> <p>ZF=0, 结果不为 0;</p> <p>PF=1, 低 8 位有偶数个 0;</p> <p>AF=1, 第三位产生借位;</p> <p>SF=1, 符号位为 1;</p> <p>(10) SBB AH, [0103H]//说明指令执行原理, 包括标志位: 带借位的减法运算, 将 AH 的数据(55H)减去地址[0103H]的数据(0C2H), 再减去 CF 中的数</p>
--	--

据，结果存入 AH 中。

01010101-11000010-1=1 10010010

CF=1, 第一位产生借位;

OF=1, 产生溢出;

ZF=0, 结果不为 0;

PF=0, 低 8 位有奇数个 0;

AF=0, 第三位未产生借位;

SF=1, 符号位为 1;

(11) MOV [0106H], AX: 将 AX 的数据存入地址为 0106H 的内存中

5、使用 r 指令修改 IP 的值到程序初始处，使用 g 命令并设置断点(最后一条指令的后三个字节)运行上述程序。

```
AX=92EB BX=0000 CX=0000 DX=0000 SP=00FD BP=0000 SI=0000 DI=0000
DS=003F ES=073F SS=093F CS=073F IP=0129  NU UP EI NG NZ NA PE NC
073F:0129 0000      ADD     [BX+SI],AL      DS:0000=EB
-r ip
IP 0129
:0100
-g 073f:0127
```

6、使用 d 命令查看内存单元的值。

```
-d
073F:0100  C7 06 00 01 64 55 C7 06-02 01 79 C2 A1 00 01 50  ....dU...y...P
073F:0110  02 06 02 01 12 26 03 01-A3 04 01 58 2A 06 02 01  ....&.....X*...
073F:0120  1A 26 03 01 A3 06 01 00-00 00 00 00 00 00 00 00  .&.....
073F:0130  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00  .....
073F:0140  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00  .....
073F:0150  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00  .....
073F:0160  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00  .....
073F:0170  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00  .....
-a 0104
003F:0100  DD 17 EB 92-00 00 00 00 00 00 00 00 00 00 00  .....
003F:0110  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00  .....
003F:0120  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00  .....
003F:0130  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00  .....
003F:0140  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00  .....
003F:0150  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00  .....
003F:0160  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00  .....
003F:0170  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00  .....
003F:0180  00 00 00 00
-S
```

结 论 (结 果)	该程序片段计算了 5564H+C279H 的值并将其存入[0104H],计算了 5564H-C279H 的值并将其存入 [0106H]中。
小 结	通过此次实验,我进一步的了解到如何用 DEBUG 编写及调试一个程序,学会了如何使用 DOXBOX 操作平台进行 DEBUX. EXE 的使用,并初步掌握了与汇编学习有关的常用 DEBUG 命令。第二次实验让我成功的进行了汇编实验中的简单计算,对程序的流程更加明确和清晰。
指导老 师评 议	<p>成绩评定:</p> <p>指导教师签名:</p>