

实验报告 Debug 用法实验

Hollow Man

一、实验环境

一台带有 MASM 软件的装有 Windows XP 系统的实验室计算机。

二、实验准备

用 Win+R 键打开“运行”，输入 cmd 并回车，打开“命令提示符”窗口程序。

在命令行中输入” cd /d D:\ “切换到 D 盘根目录。

输入“ MD JSL”创建 JSL 工作文件夹。

输入” cd JSL”切换到 JSL 工作目录

输入” copy C:\MASM* .”将程序文件拷贝进工作目录。

三、实验内容

1. 任务 1

按提示用 a 命令输入指令，得到以下运行结果：

```

命令提示符 - debug

D:\JSL>debug
-a
1381:0100 mov ax,4E20
1381:0103 add ax,1416
1381:0106 mov bx,2000
1381:0109 add ax,bx
1381:010B mov bx,ax
1381:010D add ax,001A
1381:0110 mov bx,0026
1381:0113 add al,b1
1381:0115 add ah,b1
1381:0117 add bh,al
1381:0119 mov ah,0
1381:011B add al,b1
1381:011D add al,9C
1381:011F
-t
AX=4E20 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=1381 ES=1381 SS=1381 CS=1381 IP=0103  NU UP EI PL NZ NA PO NC
1381:0103 051614      ADD     AX,1416
-t
AX=6236 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=1381 ES=1381 SS=1381 CS=1381 IP=0106  NU UP EI PL NZ NA PE NC
1381:0106 BB0020      MOV     BX,2000
-t
AX=6236 BX=2000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=1381 ES=1381 SS=1381 CS=1381 IP=0109  NU UP EI PL NZ NA PE NC
1381:0109 01D8      ADD     AX,BX
-t
AX=8236 BX=2000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=1381 ES=1381 SS=1381 CS=1381 IP=010B  OU UP EI NG NZ NA PE NC
1381:010B 89C3      MOV     BX,AX
-t
AX=8236 BX=8236 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=1381 ES=1381 SS=1381 CS=1381 IP=010D  OU UP EI NG NZ NA PE NC
1381:010D 051A00      ADD     AX,001A
-t
AX=8250 BX=8236 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=1381 ES=1381 SS=1381 CS=1381 IP=0110  NU UP EI NG NZ AC PE NC
1381:0110 BB2600      MOV     BX,0026
-

```

```

命令提示符 - debug
AX=8236 BX=2000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=1381 ES=1381 SS=1381 CS=1381 IP=010B  OV UP EI NG NZ NA PE NC
1381:010B 89C3      MOV     BX,AX
-t
AX=8236 BX=8236 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=1381 ES=1381 SS=1381 CS=1381 IP=010D  OV UP EI NG NZ NA PE NC
1381:010D 051A00    ADD     AX,001A
-t
AX=8250 BX=8236 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=1381 ES=1381 SS=1381 CS=1381 IP=0110  NU UP EI NG NZ AC PE NC
1381:0110 BB2600    MOV     BX,0026
-t
AX=8250 BX=0026 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=1381 ES=1381 SS=1381 CS=1381 IP=0113  NU UP EI NG NZ AC PE NC
1381:0113 00D8      ADD     AL,BL
-t
AX=8276 BX=0026 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=1381 ES=1381 SS=1381 CS=1381 IP=0115  NU UP EI PL NZ NA PO NC
1381:0115 00DC      ADD     AH,BL
-t
AX=8876 BX=0026 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=1381 ES=1381 SS=1381 CS=1381 IP=0117  NU UP EI NG NZ NA PO NC
1381:0117 00C7      ADD     BH,AL
-t
AX=8876 BX=7626 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=1381 ES=1381 SS=1381 CS=1381 IP=0119  NU UP EI PL NZ NA PO NC
1381:0119 B400      MOV     AH,00
-t
AX=0076 BX=7626 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=1381 ES=1381 SS=1381 CS=1381 IP=011B  NU UP EI PL NZ NA PO NC
1381:011B 00D8      ADD     AL,BL
-t
AX=009C BX=7626 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=1381 ES=1381 SS=1381 CS=1381 IP=011D  OV UP EI NG NZ NA PE NC
1381:011D 049C      ADD     AL,9C
-t
AX=0038 BX=7626 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=1381 ES=1381 SS=1381 CS=1381 IP=011F  OV UP EI PL NZ AC PO CY
1381:011F 1300      ADC     AX,[BX+SI]
DS:7626=0000

```

由图示运行结果可以看到，每次执行完指令后，CS:IP 自动指向了下一个命令地址，并且按照指令的命令操作进行运算后，相关寄存器按照指令要求发生了数值的变化。

同理，用 e 命令直接输入机器码进内存，在运行前记得调整 CS:IP 指向命令开始语句的内存地址，也可得到同样的结果。

该程序指令含义详解：

- 将 4E20 写入 AX 中
- 将 1416 写入 AX 中
- 将 2000 写入 BX 中
- 将 AX+BX 的值写入 AX 中
- 将 AX 的值写入 BX 中
- 将 AX+BX 的值写入 AX 中
- 将 001A 写入 AX 中
- 将 0026 写入 BX 中
- 将 AL(AX 的后 2 个低位)+BL(BX 的后 2 个低位)的值相加写入 AL 中
- 将 BL 的值写入 AH(AX 的前 2 个高位)中
- 将 AL 的值写入 BH(BX 的前 2 个高位)中
- 将 0 写入 AH 中

- 将 BL 的值写入 AL 中
- 将 9C 和 AL 中的值相加，并写入 AL

2. 任务 2

按提示用 a 命令输入指令，得到以下运行结果：

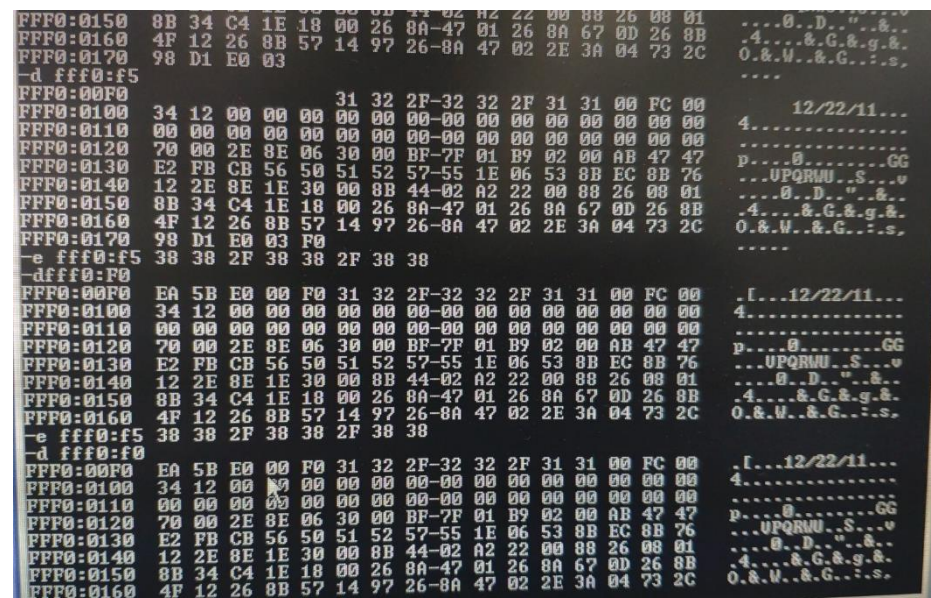
```
命令提示符 - debug
D:\JSL>debug
-a 2000:0
2000:0000 mov ax,1
2000:0003 add ax,ax
2000:0005 jmp 2000:0003
2000:0007
-r cs
CS 1381
:2000
-r ip
IP 0100
:t
AX=0001 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=1381 ES=1381 SS=1381 CS=2000 IP=0003  NU UP EI PL NZ NA PO NC
2000:0003 01C0      ADD     AX,AX
-t
AX=0002 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=1381 ES=1381 SS=1381 CS=2000 IP=0005  NU UP EI PL NZ NA PO NC
2000:0005 EBFC      JMP     0003
-t
AX=0002 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=1381 ES=1381 SS=1381 CS=2000 IP=0003  NU UP EI PL NZ NA PO NC
2000:0003 01C0      ADD     AX,AX
-t
AX=0004 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=1381 ES=1381 SS=1381 CS=2000 IP=0005  NU UP EI PL NZ NA PO NC
2000:0005 EBFC      JMP     0003
-t
AX=0004 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=1381 ES=1381 SS=1381 CS=2000 IP=0003  NU UP EI PL NZ NA PO NC
2000:0003 01C0      ADD     AX,AX
-t
AX=0008 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=1381 ES=1381 SS=1381 CS=2000 IP=0005  NU UP EI PL NZ NA PO NC
2000:0005 EBFC      JMP     0003
-t
AX=0008 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=1381 ES=1381 SS=1381 CS=2000 IP=0003  NU UP EI PL NZ NA PO NC
2000:0003 01C0      ADD     AX,AX
-t
```

当 t 输入 16 次时，即程序循环了 8 次时，在 AX 寄存器中得到了 2 的 8 次方值 256。

该程序的原理是：

- 首先将 AX 赋值为 1，
- 然后将 AX 的数值变为 2 倍，
- 最后执行跳转，重复上一步。

3. 任务 3



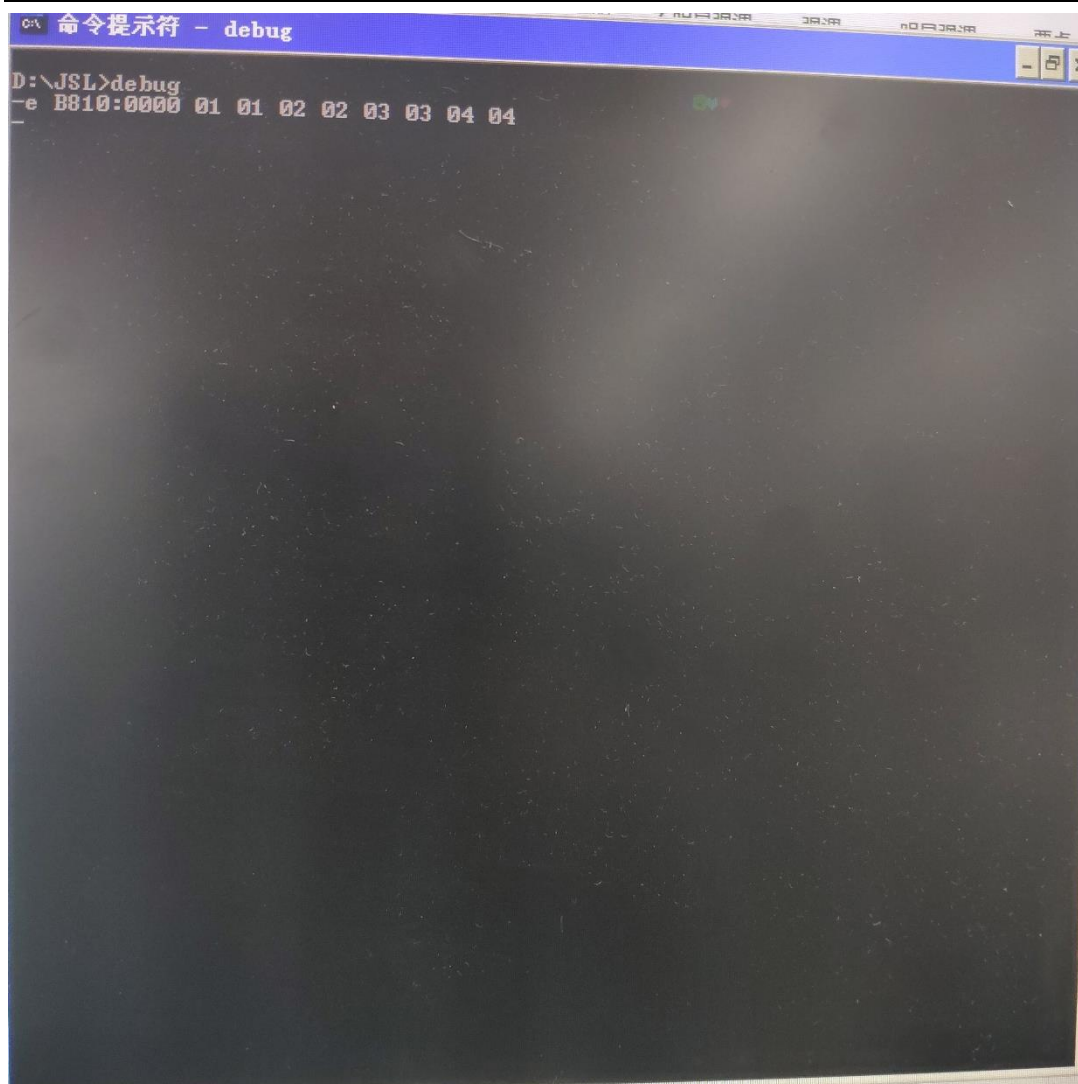
通过 d 指令我查找到了 PC 机主板 ROM 的生产日期存放在 FFF0:F5-FFF0:FC，是 2011 年 12 月 22 日生产的。

然后我试图通过 e 命令更改生产日期为 88/88/88，结果无法更改，其原因：ROM 是只读的，不能进行写入操作。

查阅教材，获知其原理：在 16 位系统中，C0000-FFFF 的 24KB 空间是各类 ROM 的地址空间，自然也就能够查询到主板 ROM 生产日期，并且不能进行写入操作了。

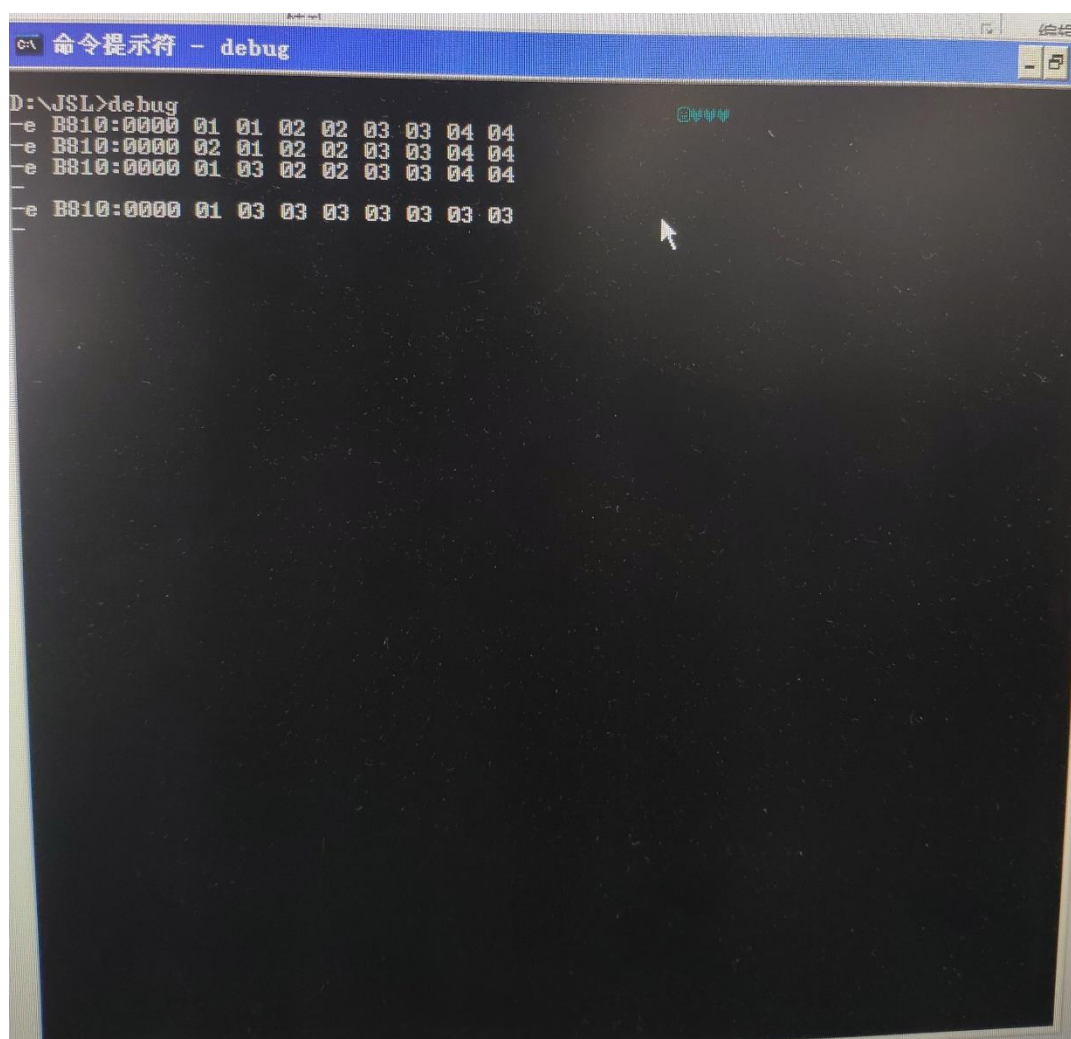
4. 任务 4

按照实验要求进行操作，得到如下结果：



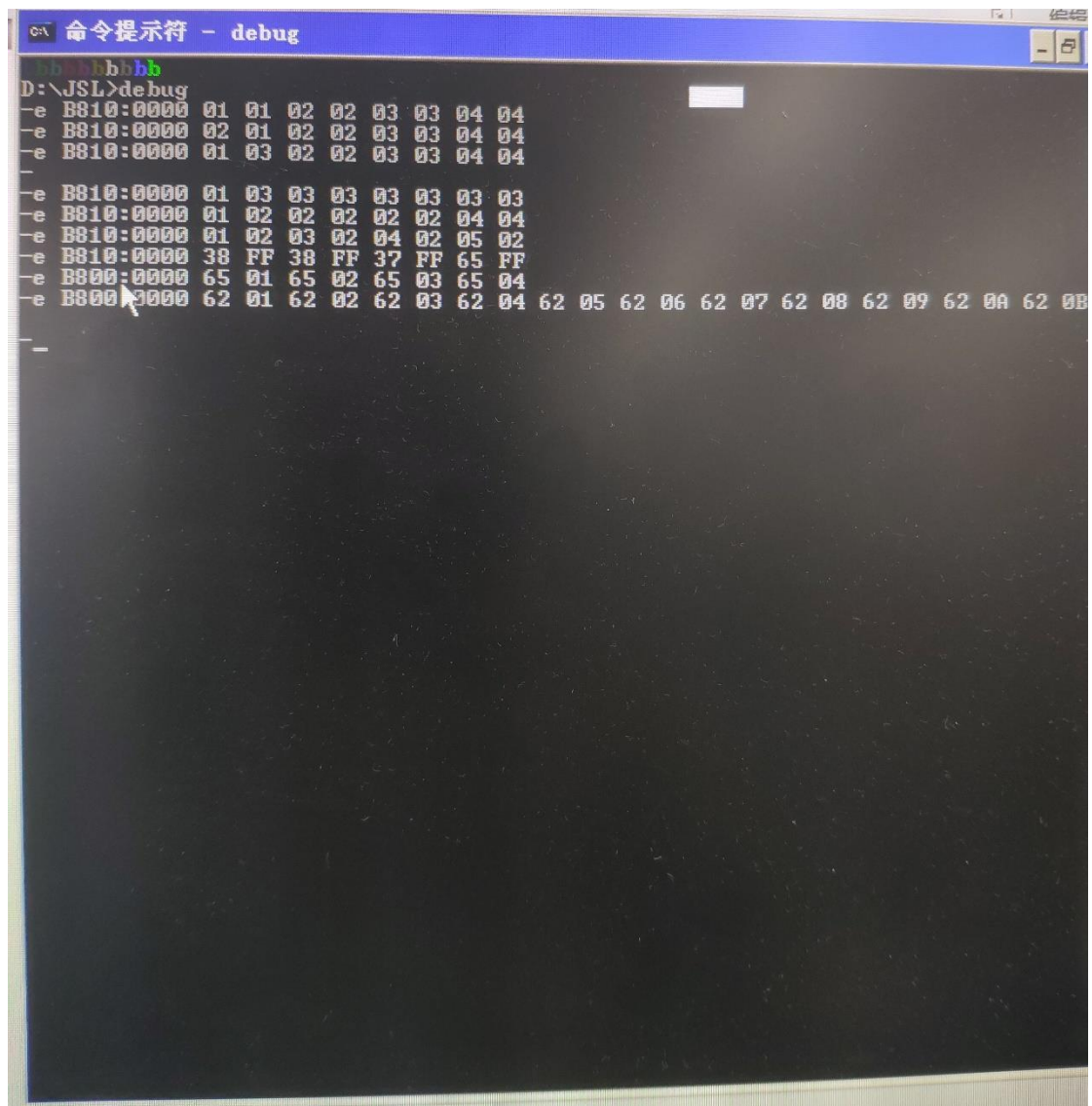
可以发现右上角出现了彩色字符。

继续改变数值，发现彩色字符的颜色和字符的内容都在变化：



```
命令提示符 - debug
D:\JSL>debug
-e B810:0000 01 01 02 02 03 03 04 04
-e B810:0000 02 01 02 02 03 03 04 04
-e B810:0000 01 03 02 02 03 03 04 04
-e B810:0000 01 03 03 03 03 03 03 03
```

改变地址值，发现字符的位置发生了移动：



同时通过多次实验，我还发现，写入内存地址的偶数位为显示字符的编码值，奇数位为显示的颜色值。

查阅教材，获知其原理：在 16 位系统中，A0000-BFFFF 的 8KB 空间是显存地址空间，而屏幕显示图像的原理是显卡读取显存中的内容显示到屏幕上，所以在强行更改显存内容时会出现这种现象。

四、实验总结

通过这次实验，我已经能够熟练使用 debug 的 r, d, e, u, a, t 命令进行程序的调试和内存地址值的修改。同时，我还了解到了不同内存地址对应的硬件设备和显卡的工作原理，收获颇丰。