

中国海洋大学计算机科学与技术系

实验报告

姓名：岳宇轩

年级：2019

专业：计算机科学与技术

科目：计算机系统基础

题目：熟悉指令系统

实验时间：2020 年 11 月 25 日

实验成绩：

实验教师：孙鑫

一、实验目的：

- 1.熟悉 8088/8086 的指令书写格式、功能。
- 2.能熟练运用数据传递、算术、逻辑三类指令。

二、实验环境(硬件或软件)：

在 DOS 或 Windows 的命令行窗口执行命令 Debug.exe，进入 Debug 程序环境，熟悉 Debug 各个命令的用法，Windows 8 版本没有 Debug 环境，需要配置虚拟机环境。

三、实验原理：

1. DEBUG 的加载及其常用命令（A、U、R、D、E、T、P、G、Q）的使用情况。（注意：微机进入 DEBUG 状态下之后，一切立即数和地址数据均被默认为十六进制数，在输入时数的后面不加后缀“H”。）
- 2.按照程序要求编写简单程序段。
- 3.在 Debug 中输入简单的汇编程序片断，并调试运行，得出结果。

四、实验内容(实验步骤或者程序编写)：

实验内容：

- 1.用 DEBUG 调试程序功能编程序计算 $(W - (X \times Y + Z - 540)) / X$ 。其中 X、Y、Z、W 均为 16 位带符号数，要求上式计算结果的商存入 AX 寄存器中，余数存入 DX 寄存器。掌握算术指令的用法。
- 2.用 DEBUG 编程并查看逻辑指令的运行结果，掌握逻辑指令的用法。
- 3.熟悉其它几组指令的功能和用法，并区分字、字节操作。

实验步骤：

- 1.预先编制程序片段，实现 $(W - (X \times Y + Z - 540)) / X$ ，将计算结果的商存入 AX 寄存器中，余数存入 DX 寄存器。

程序代码如下：

```
DATA SEGMENT
X DW 1200H
Y DW 0034H
```

```

Z DW 0F045H
W DW 034AH
DATA ENDS
CODE SEGMENT
    ASSUME CS: CODE, DS: DATA
    START:
    MOV AX,X
    IMUL Y;把 x*y 的值放到 DX,AX 中
    MOV CX,AX
    MOV BX,DX
    MOV AX,Z
    CWD;把 Z 放到 DX,AX 中
    ADD CX,AX
    ADC BX,DX;x*y+z 放到 BX,CX
    SUB CX,540
    SBB BX,0;x*y+z-540
    MOV AX,W
    CWD;W 放到 DX,AX 中
    SUB AX,CX
    SBB DX,BX;W-(x*y+z-540)
    IDIV X;(W-(x*y+z-540))/x 放到 AX, 商放到 DX
    MOV AH, 4CH
    INT 21H
CODE ENDS
    END START

```

2. 可以在 Debug 程序中使用 E 命令为变量 X、Y、Z、W 设置初值。

```

DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Pro...
Z:\>SET BLASTER=A220 I7 D1 H5 T6

Z:\>mount c c:\
Mounting c:\ is NOT recommended. Please mount a (sub)directory next time.
Drive C is mounted as local directory c:\

Z:\>c:\

C:\>cd tools

C:\TOOLS>DEBUG EXP2.EXE
Microsoft (R) Symbolic Debug Utility Version 4.00
Copyright (C) Microsoft Corp 1984, 1985. All rights reserved.

Processor is [80286]
-R
AX=FFFF BX=0000 CX=003A DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=0AA7 ES=0AA7 SS=0AB6 CS=0AB8 IP=0000 NU UP EI PL NZ NA PO NC
0AB8:0000 A10000 MOV AX,10000 DS:0000=20CD
-E DS:0
0AA7:0000 CD.04 20.00 FF.00 9F.00 00.1C EA.02 FF.0A FF.00
0AA7:0008 AD.

```

数据的高字节保存在高地址中，为小端模式

我这里给 XYZW 分别复制 4 0 540 10

这样运算 $(W - (X \times Y + Z - 540)) / X$ 得到商为 2 余数为 2

最终结果 AX 寄存器应该是 0002 DX 应该是 0002

```
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Pro...
-T
AX=021C BX=0000 CX=0000 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=0AA7 ES=0AA7 SS=0AB6 CS=0AB8 IP=001A NU UP EI PL ZR NA PE NC
0AB8:001A A10600 MOV AX,[0006] DS:0006=000A
-T
AX=000A BX=0000 CX=0000 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=0AA7 ES=0AA7 SS=0AB6 CS=0AB8 IP=001D NU UP EI PL ZR NA PE NC
0AB8:001D 99 CWD
-T
AX=000A BX=0000 CX=0000 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=0AA7 ES=0AA7 SS=0AB6 CS=0AB8 IP=001E NU UP EI PL ZR NA PE NC
0AB8:001E 2BC1 SUB AX,CX
-T
AX=000A BX=0000 CX=0000 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=0AA7 ES=0AA7 SS=0AB6 CS=0AB8 IP=0020 NU UP EI PL NZ NA PE NC
0AB8:0020 1BD3 SBB DX,BX
-T
AX=000A BX=0000 CX=0000 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=0AA7 ES=0AA7 SS=0AB6 CS=0AB8 IP=0022 NU UP EI PL ZR NA PE NC
0AB8:0022 F73E0000 IDIV Word Ptr [0000] DS:0000=0004
-T
AX=0002 BX=0000 CX=0000 DX=0002 SP=0000 BP=0000 SI=0000 DI=0000
DS=0AA7 ES=0AA7 SS=0AB6 CS=0AB8 IP=0026 NU UP EI PL ZR NA PE NC
0AB8:0026 B44C MOV AH,4C ;'L'
- ▲
```

↑ 程序运行结果与分析一致 ↑

3. 使用 A 命令汇编程序片段，并用 G 或 T 命令执行，用 R 或 D 命令查看结果。

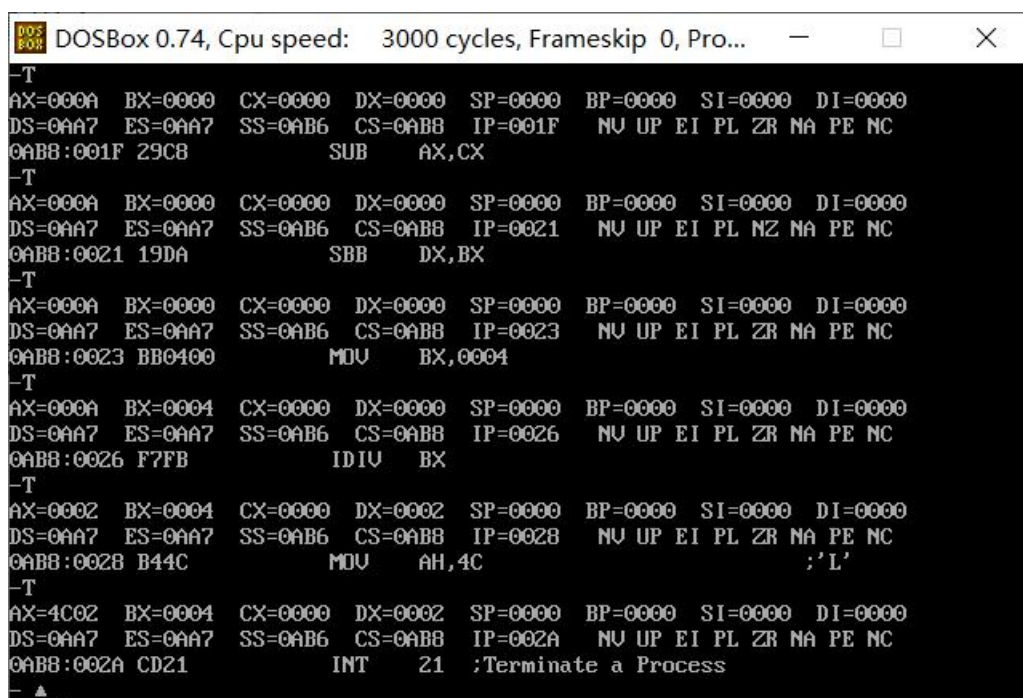
```
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Pro...
Copyright (C) Microsoft Corp 1984, 1985. All rights reserved.

Processor is I802861
-A
0AB8:0000 MOV AX,0004
0AB8:0003 MOV BX,0000
0AB8:0006 IMUL BX
0AB8:0008 MOV CX,AX
0AB8:000A MOV BX,DX
0AB8:000C MOV AX,021C
0AB8:000F CWD
0AB8:0010 ADD CX,AX
0AB8:0012 ADC BX,DX
0AB8:0014 SUB CX,021C
0AB8:0018 SBB BX,0000
0AB8:001B MOV AX,000A
0AB8:001E CWD
0AB8:001F SUB AX,CX
0AB8:0021 SBB DX,BX
0AB8:0023 MOV BX,0004
0AB8:0026 IDIV BX
0AB8:0028 MOV AH,4CH
0AB8:002A INT 21H
0AB8:002C
- ▲
```

这是用 A 命令

汇编的程序片段，要注意的两点是：1.A 指令中把如 XYZW 这样的变量用立即数代替 2.IMUL IDIV 指令的源操作数 SRC 不能为立即数，所以我把立即数先存到一个寄存器（BX）中（注意这个寄存器需要是无用的，即不存储需要用的数据，这样用它存立即数后不会对程序运行产生影响。在这个程序中我用 BX 的原因是 BX 寄存器存了立即数使用后又立即被其它数据覆盖掉了，因此无影响）

- 我这里给 XYZW 的复制是 4 0 540 10 运算结果商 2 余 2



```
DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Pro...
-T
AX=000A BX=0000 CX=0000 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=0AA7 ES=0AA7 SS=0AB6 CS=0AB8 IP=001F  NU UP EI PL ZR NA PE NC
0AB8:001F 29CB          SUB    AX,CX
-T
AX=000A BX=0000 CX=0000 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=0AA7 ES=0AA7 SS=0AB6 CS=0AB8 IP=0021  NU UP EI PL NZ NA PE NC
0AB8:0021 19DA          SBB    DX,BX
-T
AX=000A BX=0000 CX=0000 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=0AA7 ES=0AA7 SS=0AB6 CS=0AB8 IP=0023  NU UP EI PL ZR NA PE NC
0AB8:0023 B0400        MOV    BX,0004
-T
AX=000A BX=0004 CX=0000 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=0AA7 ES=0AA7 SS=0AB6 CS=0AB8 IP=0026  NU UP EI PL ZR NA PE NC
0AB8:0026 F7FB          IDIV   BX
-T
AX=0002 BX=0004 CX=0000 DX=0002 SP=0000 BP=0000 SI=0000 DI=0000
DS=0AA7 ES=0AA7 SS=0AB6 CS=0AB8 IP=0028  NU UP EI PL ZR NA PE NC
0AB8:0028 B44C          MOV    AH,4C          ;'L'
-T
AX=4C02 BX=0004 CX=0000 DX=0002 SP=0000 BP=0000 SI=0000 DI=0000
DS=0AA7 ES=0AA7 SS=0AB6 CS=0AB8 IP=002A  NU UP EI PL ZR NA PE NC
0AB8:002A CD21          INT     21          ;Terminate a Process
- ^ _
```

从上图可以看到执行完 IDIV 后 AX 存的 0002，DX 存的 0002，与预期结果一致

4. 使用 A 命令汇编几条逻辑指令，并运行，查看结果。

NOT 指令

```

DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Pro...
0AB8:002B B44C      MOV     AH,4C      ;'L'
-T
AX=4C02 BX=0004 CX=0000 DX=0002 SP=0000 BP=0000 SI=0000 DI=0000
DS=0AA7 ES=0AA7 SS=0AB6 CS=0AB8 IP=002A  NU UP EI PL ZR NA PE NC
0AB8:002A CD21      INT     21      ;Terminate a Process
-Q

C:\TOOLS>DEBUG EXP2.EXE
Microsoft (R) Symbolic Debug Utility Version 4.00
Copyright (C) Microsoft Corp 1984, 1985. All rights reserved.

Processor is [80286]
-A
0AB8:0000 MOV BX,FFFF
0AB8:0003 NOT BX
0AB8:0005
-T
AX=FFFF BX=FFFF CX=003A DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=0AA7 ES=0AA7 SS=0AB6 CS=0AB8 IP=0003  NU UP EI PL NZ NA PO NC
0AB8:0003 F7D3      NOT     BX
-T
AX=FFFF BX=0000 CX=003A DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=0AA7 ES=0AA7 SS=0AB6 CS=0AB8 IP=0005  NU UP EI PL NZ NA PO NC
0AB8:0005 0200      ADD     AL,[BX+SI]      DS:0000=CD
- ▲

```

NOT 指令的操作数不能为立即数，所以我先把 FFFF 放到了 BX 里，再对 BX 进行 NOT，得到 0000，实验结果与预期一致

AND 指令

```

-A
0AB8:001E MOV BX,0F0F
0AB8:0021 AND BX,F0F0
0AB8:0025
-T
AX=FFFF BX=0F0F CX=003A DX=F00F SP=0000 BP=0000 SI=0000 DI=0000
DS=0AA7 ES=0AA7 SS=0AB6 CS=0AB8 IP=0021  NU UP EI NG NZ NA PE NC
0AB8:0021 81E3F0F0   AND     BX,F0F0
-T
AX=FFFF BX=0000 CX=003A DX=F00F SP=0000 BP=0000 SI=0000 DI=0000
DS=0AA7 ES=0AA7 SS=0AB6 CS=0AB8 IP=0025  NU UP EI PL ZR NA PE NC
0AB8:0025 00B4CCD    ADD     [SI+CD4C],DH      DS:CD4C=00
- ▲

```

同样的，AND 的目的操作数不能是立即数，先存寄存器。我这里 $BX = 0F0F$ ，AND BX,F0F0 后，放到目的操作数 BX 中，可以看到在执行 AND 后 BX 的值为 0000

OR 指令

```

-A
0AB8:0025 MOV BX,0F0F
0AB8:0028 OR BX,F0F0
0AB8:002C
-T
AX=FFFF BX=0F0F CX=003A DX=F00F SP=0000 BP=0000 SI=0000 DI=0000
DS=0AA7 ES=0AA7 SS=0AB6 CS=0AB8 IP=0028 NU UP EI PL ZR NA PE NC
0AB8:0028 81CBF0F0 OR BX,F0F0
-T
AX=FFFF BX=FFFF CX=003A DX=F00F SP=0000 BP=0000 SI=0000 DI=0000
DS=0AA7 ES=0AA7 SS=0AB6 CS=0AB8 IP=002C NU UP EI NG NZ NA PE NC
0AB8:002C C9 LEAVE
-A ▲

```

与 AND 指令差不多，BX=0F0F,OR BX,F0F0 后 BX=FFFF

XOR 指令

```

-A
0AB8:002C MOV BX,00FF
0AB8:002F XOR BX,F00F
0AB8:0033
-T
AX=FFFF BX=00FF CX=003A DX=F00F SP=0000 BP=0000 SI=0000 DI=0000
DS=0AA7 ES=0AA7 SS=0AB6 CS=0AB8 IP=002F NU UP EI NG NZ NA PE NC
0AB8:002F 81F30FF0 XOR BX,F00F
-T
AX=FFFF BX=F0F0 CX=003A DX=F00F SP=0000 BP=0000 SI=0000 DI=0000
DS=0AA7 ES=0AA7 SS=0AB6 CS=0AB8 IP=0033 NU UP EI NG NZ NA PE NC
0AB8:0033 65 DB 65
-A ▲

```

BX=00FF,XOR BX,F00F 后 BX=F0F0，实验结果与预期一致

TEST 指令

```

-A
0AB8:0049 MOV BX,0000
0AB8:004C TEST 0001
^ Error
0AB8:004C TEST BX,0001
0AB8:0050
-T
AX=FFFF BX=0000 CX=003A DX=F00F SP=0000 BP=0000 SI=FFFF DI=0000
DS=0AA7 ES=0AA7 SS=0AB6 CS=0AB8 IP=004C NU UP EI PL NZ NA PO NC
0AB8:004C F7C30100 TEST BX,0001
-T
AX=FFFF BX=0000 CX=003A DX=F00F SP=0000 BP=0000 SI=FFFF DI=0000
DS=0AA7 ES=0AA7 SS=0AB6 CS=0AB8 IP=0050 NU UP EI PL ZR NA PE NC
0AB8:0050 3007 XOR IBX,AL DS:0000=CD
-A

```

BX=0000

TSET BX,0001 后 BX 值不变，标志寄存器结果与预期一致

5.写出实验报告。

五、实验结果(实验截图)及分析：

写在上面了