

## 熟悉指令系统

### 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：

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。

编辑器：EDIT.exe

编程序：MASM.exe

连接程序：LINK.exe

调试程序：DEBUG.exe

### 三、实验原理：

#### 汇编程序从写出到执行的过程

编程 → 1.asm → 编译 → 1.obj → 连接 → 1.exe → 加载 → 内存中的程序 → 运行  
( edit )                      ( masm )                      ( link )                      ( command )                      ( CPU )

### 四、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。

## 五、实验主要步骤:

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

MOV CX,AX

MOV BX,DX

MOV AX,Z

CWD

ADD CX,AX

ADC BX,DX

SUB CX,540

MOV AX,W

CWD

SUB AX,CX

SBB DX,BX

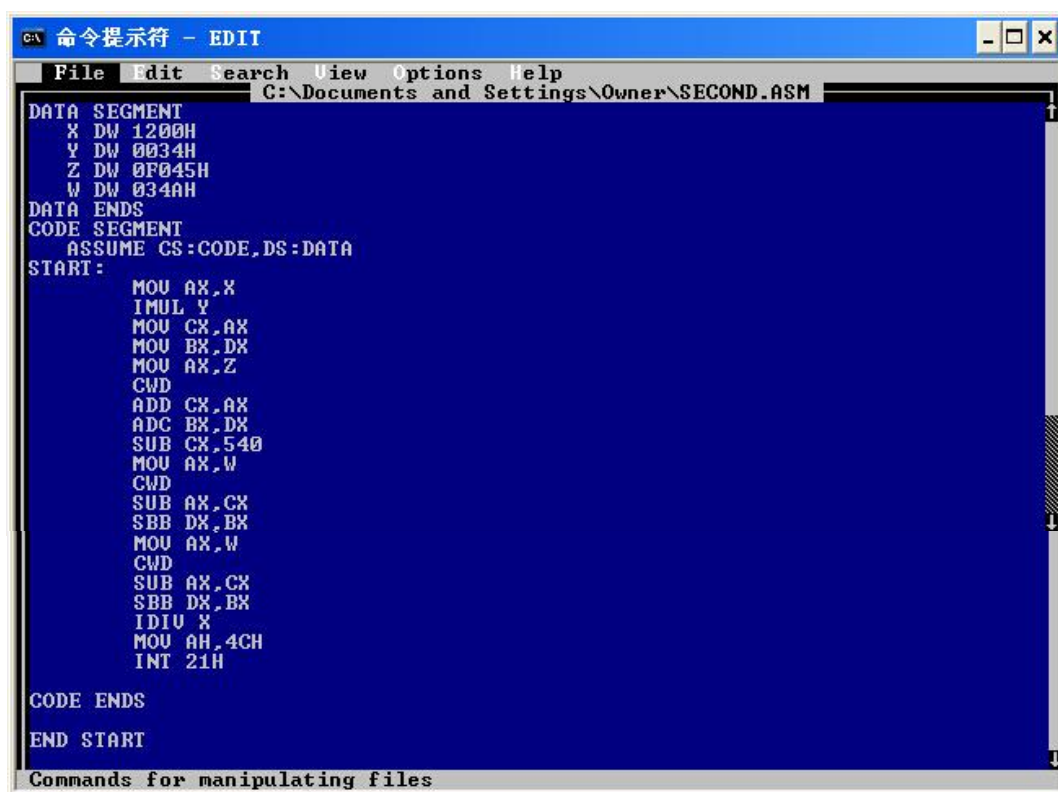
IDIV X

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。

依次为: 编译, 连接, 执行, 逐步调试

```
C:\DOCUME~1\Owner>MASM
Microsoft (R) Macro Assembler Version 5.10
Copyright (C) Microsoft Corp 1981, 1988. All rights reserved.

Source filename [.ASM]: SECOND
Object filename [SECOND.OBJ]:
Source listing [NUL.LST]:
Cross-reference [NUL.CRF]:

49318 + 449255 Bytes symbol space free

0 Warning Errors
0 Severe Errors
```

编译

```
C:\DOCUME~1\Owner>LINK
```

```
Microsoft (R) Overlay Linker Version 3.64
Copyright (C) Microsoft Corp 1983-1988. All rights reserved.
```

连接

```
Object Modules [L.OBJ]: SECOND
Run File [SECOND.EXE]:
List File [NUL.MAP]:
Libraries [L.LIB]:
LINK : warning L4021: no stack segment
```

```
C:\DOCUME~1\Owner>SECOND
```

```
C:\DOCUME~1\Owner>_
```

执行

```
C:\DOCUME~1\Owner>DEBUG SECOND.EXE
```

```
Microsoft (R) Symbolic Debug Utility Version 4.00
Copyright (C) Microsoft Corp 1984, 1985. All rights reserved.
```

单步调试

```
Processor is [80286]
```

=====分界线=====

```
-U
0E81:0000 A10000      MOV     AX,[0000]
0E81:0003 F72E0200     IMUL   Word Ptr [0002]
0E81:0007 8BC8      MOV     CX,AX
0E81:0009 8BDA      MOV     BX,DX
0E81:000B A10400      MOV     AX,[0004]
0E81:000E 99        CWD
0E81:000F 03C8      ADD     CX,AX
0E81:0011 13DA      ADC     BX,DX
-U
0E81:0013 81E91C02     SUB     CX,021C
0E81:0017 A10600      MOV     AX,[0006]
0E81:001A 99        CWD
0E81:001B 2BC1      SUB     AX,CX
0E81:001D 1BD3      SBB     DX,BX
0E81:001F F73E0000     IDIU   Word Ptr [0000]
0E81:0023 B44C      MOV     AH,4C
0E81:0025 CD21      INT     21
```

反汇编

赋值: X =1200H; Y =0034H; Z=F045H; W=034AH

由反汇编知道数据段偏移地址开始为 0000。

注意: 高地址放高位, 低地址放地位

```
-E DS:0
```

```
0E70:0000 CD.00 20.12 FF.34 9F.00 00.45 9A.F0 F0.4A FE.03
```

```

-R
AX=0000 BX=0000 CX=0037 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=0E70 ES=0E70 SS=0E80 CS=0E81 IP=0000 NU UP EI PL NZ NA PO NC
0E81:0000 A10000 MOV AX,[0000] DS:0000=1200
-T
AX=1200 BX=0000 CX=0037 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=0E70 ES=0E70 SS=0E80 CS=0E81 IP=0003 NU UP EI PL NZ NA PO NC
0E81:0003 F72E0200 IMUL Word Ptr [0002] DS:0002=0034
-T
AX=A800 BX=0000 CX=0037 DX=0003 SP=0000 BP=0000 SI=0000 DI=0000
DS=0E70 ES=0E70 SS=0E80 CS=0E81 IP=0007 OU UP EI NG NZ NA PE CY
0E81:0007 8BC8 MOV CX,AX
-T
AX=A800 BX=0000 CX=A800 DX=0003 SP=0000 BP=0000 SI=0000 DI=0000
DS=0E70 ES=0E70 SS=0E80 CS=0E81 IP=0009 OU UP EI NG NZ NA PE CY
0E81:0009 8BDA MOV BX,DX
-T
AX=A800 BX=0003 CX=A800 DX=0003 SP=0000 BP=0000 SI=0000 DI=0000
DS=0E70 ES=0E70 SS=0E80 CS=0E81 IP=000B OU UP EI NG NZ NA PE CY
0E81:000B A10400 MOV AX,[0004] DS:0004=F045
-T
AX=F045 BX=0003 CX=A800 DX=0003 SP=0000 BP=0000 SI=0000 DI=0000
DS=0E70 ES=0E70 SS=0E80 CS=0E81 IP=000E OU UP EI NG NZ NA PE CY
0E81:000E 99 CWD
-T
AX=F045 BX=0003 CX=A800 DX=0003 SP=0000 BP=0000 SI=0000 DI=0000
DS=0E70 ES=0E70 SS=0E80 CS=0E81 IP=000E OU UP EI NG NZ NA PE CY
0E81:000E 99 CWD
-T
AX=F045 BX=0003 CX=A800 DX=FFFF SP=0000 BP=0000 SI=0000 DI=0000
DS=0E70 ES=0E70 SS=0E80 CS=0E81 IP=000F OU UP EI NG NZ NA PE CY
0E81:000F 03C8 ADD CX,AX
-T
AX=F045 BX=0003 CX=9845 DX=FFFF SP=0000 BP=0000 SI=0000 DI=0000
DS=0E70 ES=0E70 SS=0E80 CS=0E81 IP=0011 NU UP EI NG NZ NA PO CY
0E81:0011 13DA ADC BX,DX
-T
AX=F045 BX=0003 CX=9845 DX=FFFF SP=0000 BP=0000 SI=0000 DI=0000
DS=0E70 ES=0E70 SS=0E80 CS=0E81 IP=0013 NU UP EI PL NZ AC PE CY
0E81:0013 81E91C02 SUB CX,021C
-T
AX=F045 BX=0003 CX=9629 DX=FFFF SP=0000 BP=0000 SI=0000 DI=0000
DS=0E70 ES=0E70 SS=0E80 CS=0E81 IP=0017 NU UP EI NG NZ AC PO NC
0E81:0017 A10600 MOV AX,[0006] DS:0006=034A
-T
AX=034A BX=0003 CX=9629 DX=FFFF SP=0000 BP=0000 SI=0000 DI=0000
DS=0E70 ES=0E70 SS=0E80 CS=0E81 IP=001A NU UP EI NG NZ AC PO NC
0E81:001A 99 CWD
-T
AX=034A BX=0003 CX=9629 DX=FFFF SP=0000 BP=0000 SI=0000 DI=0000
DS=0E70 ES=0E70 SS=0E80 CS=0E81 IP=001A NU UP EI NG NZ AC PO NC
0E81:001A 99 CWD
-T
AX=034A BX=0003 CX=9629 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=0E70 ES=0E70 SS=0E80 CS=0E81 IP=001B NU UP EI NG NZ AC PO NC
0E81:001B 2BC1 SUB AX,CX
-T
AX=6D21 BX=0003 CX=9629 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=0E70 ES=0E70 SS=0E80 CS=0E81 IP=001D NU UP EI PL NZ NA PE CY
0E81:001D 1BD3 SBB DX,BX
-T
AX=6D21 BX=0003 CX=9629 DX=FFFC SP=0000 BP=0000 SI=0000 DI=0000
DS=0E70 ES=0E70 SS=0E80 CS=0E81 IP=001F NU UP EI NG NZ AC PE CY
0E81:001F F73E0000 IDIV Word Ptr [0000] DS:0000=1200
-T
AX=FFCE BX=0003 CX=9629 DX=F121 SP=0000 BP=0000 SI=0000 DI=0000
DS=0E70 ES=0E70 SS=0E80 CS=0E81 IP=0023 NU UP EI NG NZ AC PE CY
0E81:0023 B44C MOV AH,4C ;'L'
-T
AX=4CCE BX=0003 CX=9629 DX=F121 SP=0000 BP=0000 SI=0000 DI=0000
DS=0E70 ES=0E70 SS=0E80 CS=0E81 IP=0025 NU UP EI NG NZ AC PE CY
0E81:0025 CD21 INT 21 ;Terminate a Process
-

```

### 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。

```

-A
0E3C:0100 MOV AX,1200
0E3C:0103 MOV BX,0034
0E3C:0106 IMUL BX
0E3C:0108 MOV CX,AX
0E3C:010A MOV BX,DX
0E3C:010C MOV AX,F045
0E3C:010F CWD
0E3C:0110 ADD CX,AX
0E3C:0112 ADC BX,DX
0E3C:0114 SUB CX,21C
0E3C:0118 MOV AX,034A
0E3C:011B CWD
0E3C:011C SUB AX,CX
0E3C:011E SBB DX,BX
0E3C:0120 MOV BX,1200
0E3C:0123 IDIV BX
0E3C:0125 MOV AH,4CH
0E3C:0127 INT 21H
0E3C:0129
-
-R
AX=0000 BX=0000 CX=0000 DX=0000 SP=F0B3 BP=0000 SI=0000 DI=0000
DS=0E3C ES=0E3C SS=0E3C CS=0E3C IP=0100 NU UP EI PL NZ NA PO NC
0E3C:0100 B80012 MOV AX,1200
-T
AX=1200 BX=0000 CX=0000 DX=0000 SP=F0B3 BP=0000 SI=0000 DI=0000
DS=0E3C ES=0E3C SS=0E3C CS=0E3C IP=0103 NU UP EI PL NZ NA PO NC
0E3C:0103 BB3400 MOV BX,0034
-T
AX=1200 BX=0034 CX=0000 DX=0000 SP=F0B3 BP=0000 SI=0000 DI=0000
DS=0E3C ES=0E3C SS=0E3C CS=0E3C IP=0106 NU UP EI PL NZ NA PO NC
0E3C:0106 F7EB IMUL BX
-T
AX=A800 BX=0034 CX=0000 DX=0003 SP=F0B3 BP=0000 SI=0000 DI=0000
DS=0E3C ES=0E3C SS=0E3C CS=0E3C IP=0108 OU UP EI NG NZ NA PE CY
0E3C:0108 89C1 MOV CX,AX
-T
AX=A800 BX=0034 CX=A800 DX=0003 SP=F0B3 BP=0000 SI=0000 DI=0000
DS=0E3C ES=0E3C SS=0E3C CS=0E3C IP=010A OU UP EI NG NZ NA PE CY
0E3C:010A 89D3 MOV BX,DX
-T
AX=A800 BX=0003 CX=A800 DX=0003 SP=F0B3 BP=0000 SI=0000 DI=0000
DS=0E3C ES=0E3C SS=0E3C CS=0E3C IP=010C OU UP EI NG NZ NA PE CY
0E3C:010C B845F0 MOV AX,F045
-
AX=F045 BX=0003 CX=A800 DX=0003 SP=F0B3 BP=0000 SI=0000 DI=0000
DS=0E3C ES=0E3C SS=0E3C CS=0E3C IP=010F OU UP EI NG NZ NA PE CY
0E3C:010F 99 CWD
-T
AX=F045 BX=0003 CX=A800 DX=FFFF SP=F0B3 BP=0000 SI=0000 DI=0000
DS=0E3C ES=0E3C SS=0E3C CS=0E3C IP=0110 OU UP EI NG NZ NA PE CY
0E3C:0110 01C1 ADD CX,AX
-T
AX=F045 BX=0003 CX=9845 DX=FFFF SP=F0B3 BP=0000 SI=0000 DI=0000
DS=0E3C ES=0E3C SS=0E3C CS=0E3C IP=0112 NU UP EI NG NZ NA PO CY
0E3C:0112 11D3 ADC BX,DX
-T
AX=F045 BX=0003 CX=9845 DX=FFFF SP=F0B3 BP=0000 SI=0000 DI=0000
DS=0E3C ES=0E3C SS=0E3C CS=0E3C IP=0114 NU UP EI PL NZ AC PE CY
0E3C:0114 81E91C02 SUB CX,021C
-T
AX=F045 BX=0003 CX=9629 DX=FFFF SP=F0B3 BP=0000 SI=0000 DI=0000
DS=0E3C ES=0E3C SS=0E3C CS=0E3C IP=0118 NU UP EI NG NZ AC PO NC
0E3C:0118 B84A03 MOV AX,034A
-T
AX=034A BX=0003 CX=9629 DX=FFFF SP=F0B3 BP=0000 SI=0000 DI=0000
DS=0E3C ES=0E3C SS=0E3C CS=0E3C IP=011B NU UP EI NG NZ AC PO NC
0E3C:011B 99 CWD
-T
AX=034A BX=0003 CX=9629 DX=0000 SP=F0B3 BP=0000 SI=0000 DI=0000
DS=0E3C ES=0E3C SS=0E3C CS=0E3C IP=011C NU UP EI NG NZ AC PO NC
0E3C:011C 29C8 SUB AX,CX
-T
AX=6D21 BX=0003 CX=9629 DX=0000 SP=F0B3 BP=0000 SI=0000 DI=0000
DS=0E3C ES=0E3C SS=0E3C CS=0E3C IP=011E NU UP EI PL NZ NA PE CY
0E3C:011E 19DA SBB DX,BX
-T
AX=6D21 BX=0003 CX=9629 DX=FFFC SP=F0B3 BP=0000 SI=0000 DI=0000
DS=0E3C ES=0E3C SS=0E3C CS=0E3C IP=0120 NU UP EI NG NZ AC PE CY
0E3C:0120 BB0012 MOV BX,1200
-T
AX=6D21 BX=1200 CX=9629 DX=FFFC SP=F0B3 BP=0000 SI=0000 DI=0000
DS=0E3C ES=0E3C SS=0E3C CS=0E3C IP=0123 NU UP EI NG NZ AC PE CY
0E3C:0123 F7FB IDIV BX
-T
AX=FFCE BX=1200 CX=9629 DX=F121 SP=F0B3 BP=0000 SI=0000 DI=0000
DS=0E3C ES=0E3C SS=0E3C CS=0E3C IP=0125 NU UP EI NG NZ AC PE CY
0E3C:0125 B44C MOV AH,4C ;'L'
-T
AX=4CCE BX=1200 CX=9629 DX=F121 SP=F0B3 BP=0000 SI=0000 DI=0000
DS=0E3C ES=0E3C SS=0E3C CS=0E3C IP=0127 NU UP EI NG NZ AC PE CY
0E3C:0127 CD21 INT 21 ;Terminate a Process
-

```

依次为：（从左到右，从上到下）

用 A 命令汇编程序片段、用 T 逐步调试

注意：由于 imul 和 idiv 指令的操作数不能为立即数，所以用 BX 储存 Y 和 X 的值

结果与用 EDIT 编写程序、用 MASM 编译、用 LINK 连接、用 DEBUG 调试的挤过是一样的。

答案均为：

AX=FFCE

DX=F121

由于最后要 mov ah,4c

所以 AX=4CCE

## 五、实验结果与分析：

结果均为：AX=FFCE；DX=F121，由于最后要 mov ah,4c 所以 AX=4CCE。

通过实验，我基本掌握了数据传递、算术指令的运用，熟悉了 8088/8086 的指令书写格式、功能，知道了会如何编写一个汇编程序，并把它编译、连接成一个可执行文件。

尤其是对于 imul 指令了解，知道了它如果两个 16 位数相乘得到的是 32 位乘积，存放在 DX，AX 中，其中 DX 存放高位字，AX 存放低位字。同时在编写程序是我们需要注意 imul 默认被乘数，idiv 是有默认被除数的，它们都默认是 AX 里的内容，所以编写程序时需要注意协调好哪些数据存放在哪些寄存器中。

在 debug 模式下调式，进一步熟悉 debug 的 R、D、E、U、A、T 等常用指令，更深刻地知道 debug 工具对于汇编程序实验的重要性。

可我还存在疑问，在单步调式时，1200H\*34H 已经进位（高位是 3），标志寄存器也显示他溢出，可当我们使用 adc 把高位相加时，我发现并没有把 CF（当前是 1）加上，理论上执行 adc 指令后，BX 的值应该为 3+0+1=4，可是 BX 的值一直是 3。