Keegan Vaz
42
SECMPNB

Experiment 1

Aim: To solve various arithmetic problems on debug.

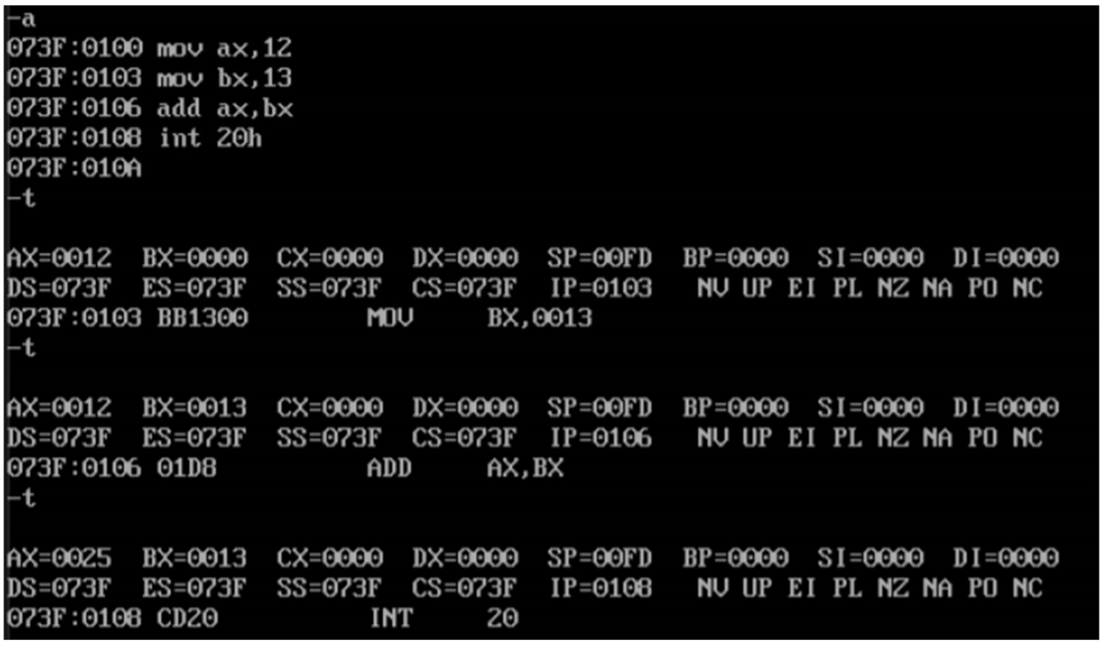
Prerequisites: Windows 7 or Virtual machine in which Windows 7-32 bit version.(Only when the system is not windows 7 32 bit.)

Theory:

The line-oriented debugger is an external command in operating systems such as DOS, OS/2 and Windows (only in 16-bit/32-bit versions). DEBUG can act as an assembler, disassembler, or hex dump program allowing users to interactively examine memory contents (in assembly language). The use of debug command is used to look at portions of your computer and write assembly code to perform certain tasks like arithmetic operations on your computer. We are able to add, subtract, multiply and divide by just writing the code in 3-4 lines. When we want to start writing a program we should insert -a. Then start the code by assigning the values to ax and bx, then we need to mention the operation and finally end with int 21h. While displaying, we just need to input the command called -t. The changes are seen whenever the command -t is runned. We need to enter the -t command till it completes the execution of each line.

Output:

1) Addition of two 8-bit numbers.



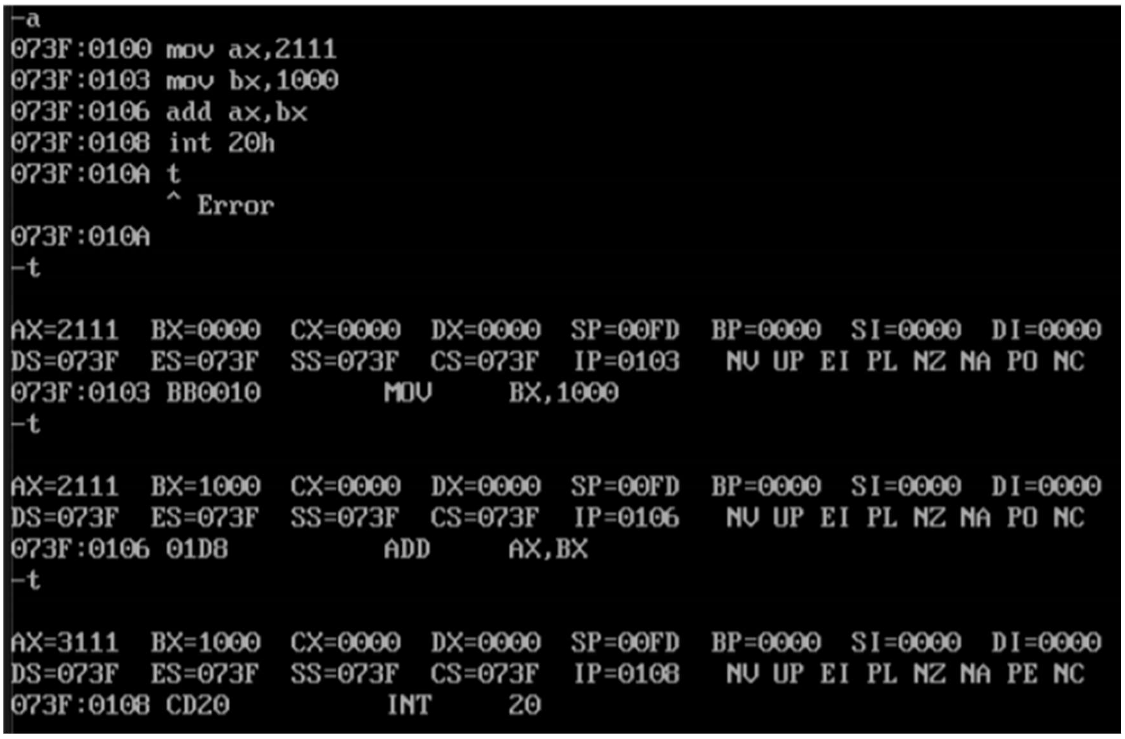
```
-a
073F:0100 mov ax,12
073F:0103 mov bx,13
073F:0106 add ax,bx
073F:0108 int 20h
073F:010A
-t

AX=0012  BX=0000  CX=0000  DX=0000  SP=00FD  BP=0000  SI=0000  DI=0000
DS=073F  ES=073F  SS=073F  CS=073F  IP=0103   NV UP EI PL NZ NA PO NC
073F:0103 BB1300        MOV     BX,0013
-t

AX=0012  BX=0013  CX=0000  DX=0000  SP=00FD  BP=0000  SI=0000  DI=0000
DS=073F  ES=073F  SS=073F  CS=073F  IP=0106   NV UP EI PL NZ NA PO NC
073F:0106 01D8          ADD     AX,BX
-t

AX=0025  BX=0013  CX=0000  DX=0000  SP=00FD  BP=0000  SI=0000  DI=0000
DS=073F  ES=073F  SS=073F  CS=073F  IP=0108   NV UP EI PL NZ NA PO NC
073F:0108 CD20          INT     20
```

2) Subtraction of two 8 bits numbers.



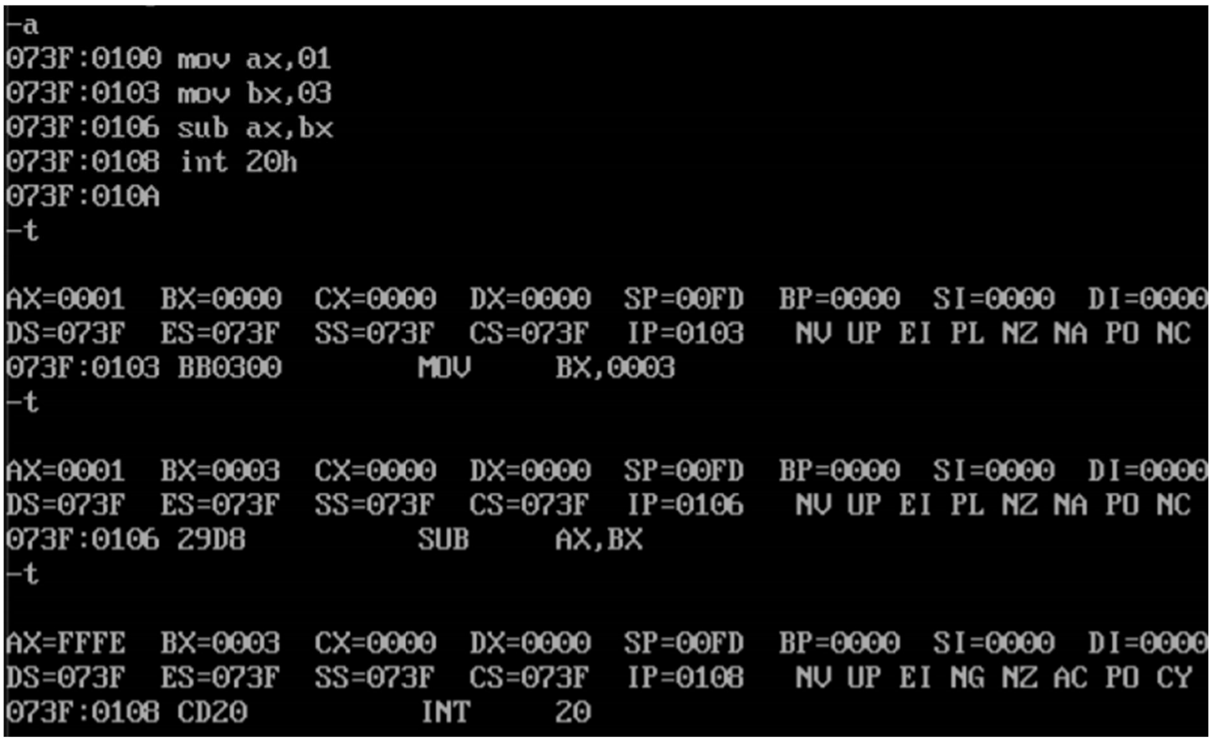
```
-a
073F:0100 mov ax,2111
073F:0103 mov bx,1000
073F:0106 add ax,bx
073F:0108 int 20h
073F:010A t
          ^ Error
073F:010A
-t

AX=2111  BX=0000  CX=0000  DX=0000  SP=00FD  BP=0000  SI=0000  DI=0000
DS=073F  ES=073F  SS=073F  CS=073F  IP=0103   NV UP EI PL NZ NA PO NC
073F:0103 BB0010        MOV     BX,1000
-t

AX=2111  BX=1000  CX=0000  DX=0000  SP=00FD  BP=0000  SI=0000  DI=0000
DS=073F  ES=073F  SS=073F  CS=073F  IP=0106   NV UP EI PL NZ NA PO NC
073F:0106 01D8          ADD     AX,BX
-t

AX=3111  BX=1000  CX=0000  DX=0000  SP=00FD  BP=0000  SI=0000  DI=0000
DS=073F  ES=073F  SS=073F  CS=073F  IP=0108   NV UP EI PL NZ NA PE NC
073F:0108 CD20          INT     20
```

3) Addition of two 16-bit numbers.



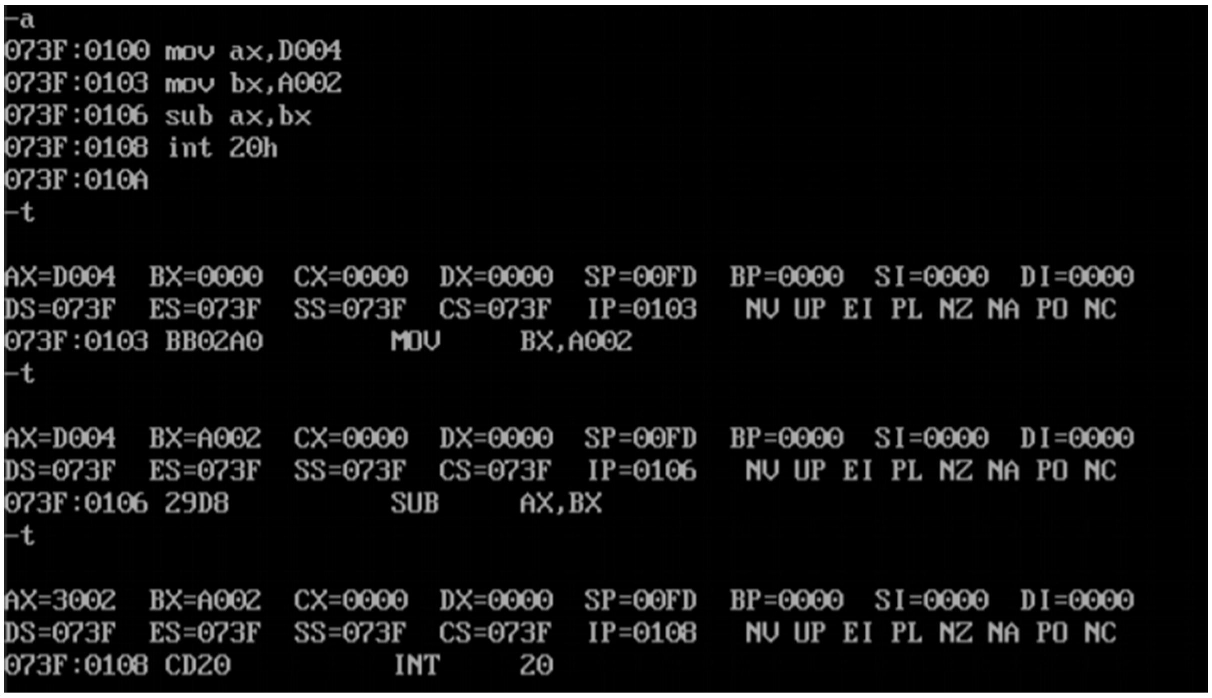
```
-a
073F:0100 mov ax,01
073F:0103 mov bx,03
073F:0106 sub ax,bx
073F:0108 int 20h
073F:010A
-t

AX=0001  BX=0000  CX=0000  DX=0000  SP=00FD  BP=0000  SI=0000  DI=0000
DS=073F  ES=073F  SS=073F  CS=073F  IP=0103   NV UP EI PL NZ NA PO NC
073F:0103 BB0300        MOV     BX,0003
-t

AX=0001  BX=0003  CX=0000  DX=0000  SP=00FD  BP=0000  SI=0000  DI=0000
DS=073F  ES=073F  SS=073F  CS=073F  IP=0106   NV UP EI PL NZ NA PO NC
073F:0106 29D8          SUB     AX,BX
-t

AX=FFFE  BX=0003  CX=0000  DX=0000  SP=00FD  BP=0000  SI=0000  DI=0000
DS=073F  ES=073F  SS=073F  CS=073F  IP=0108   NV UP EI NG NZ AC PO CY
073F:0108 CD20          INT     20
```

4) Subtraction of two 16-bit numbers.



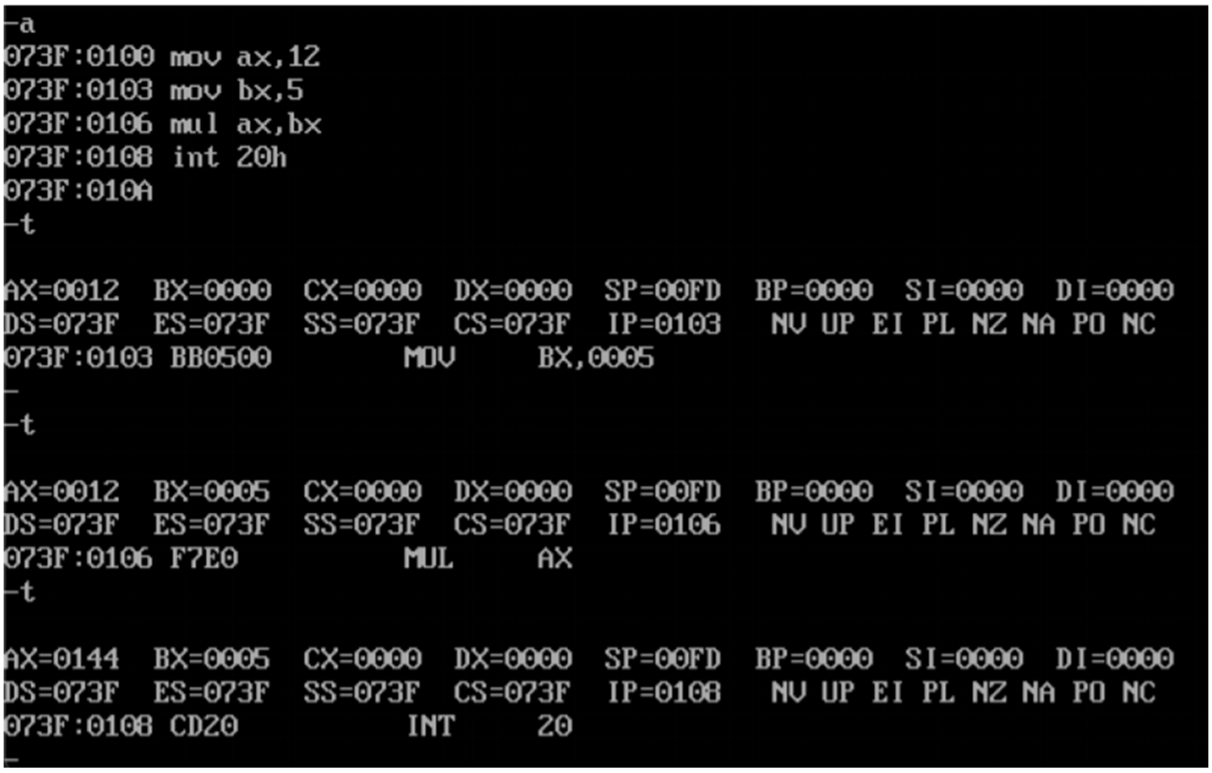
```
-a
073F:0100 mov ax,D004
073F:0103 mov bx,A002
073F:0106 sub ax,bx
073F:0108 int 20h
073F:010A
-t

AX=D004  BX=0000  CX=0000  DX=0000  SP=00FD  BP=0000  SI=0000  DI=0000
DS=073F  ES=073F  SS=073F  CS=073F  IP=0103   NV UP EI PL NZ NA PO NC
073F:0103 BB02A0        MOV     BX,A002
-t

AX=D004  BX=A002  CX=0000  DX=0000  SP=00FD  BP=0000  SI=0000  DI=0000
DS=073F  ES=073F  SS=073F  CS=073F  IP=0106   NV UP EI PL NZ NA PO NC
073F:0106 29D8          SUB     AX,BX
-t

AX=3002  BX=A002  CX=0000  DX=0000  SP=00FD  BP=0000  SI=0000  DI=0000
DS=073F  ES=073F  SS=073F  CS=073F  IP=0108   NV UP EI PL NZ NA PO NC
073F:0108 CD20          INT     20
```

5) Multiplication of two 8-bit numbers.



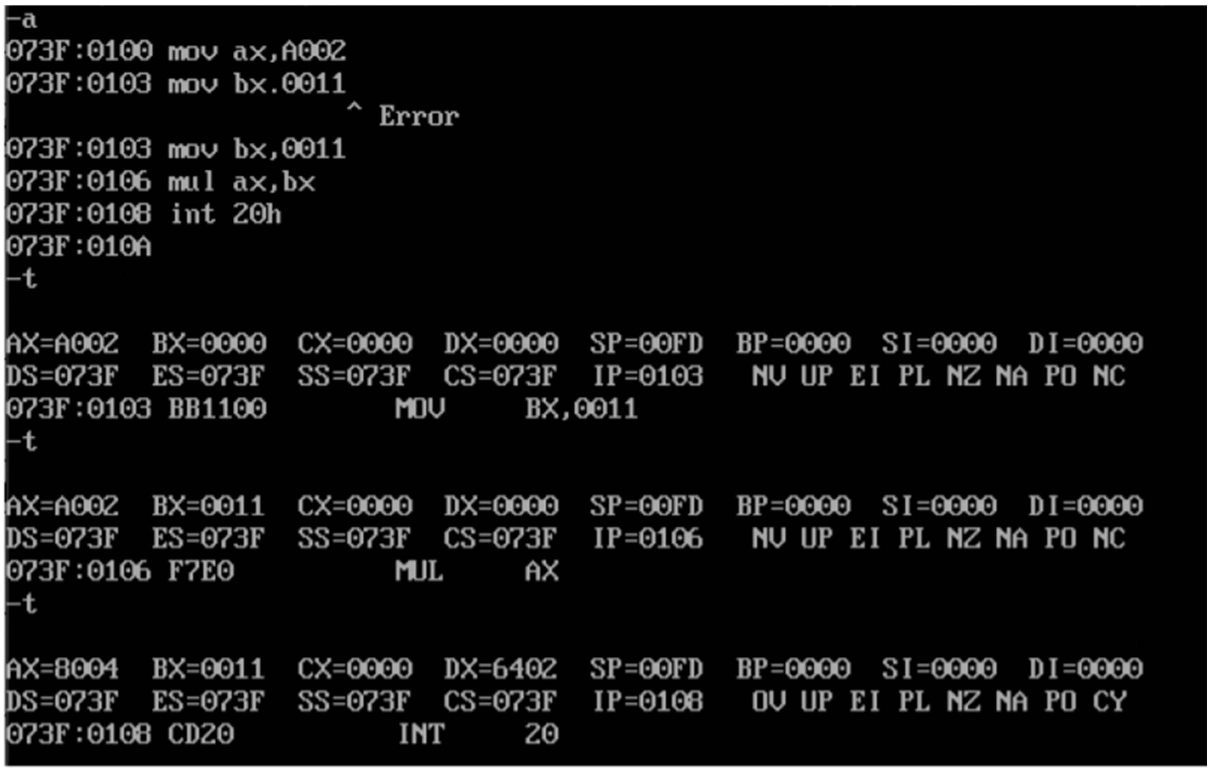
```
-a
073F:0100 mov ax,12
073F:0103 mov bx,5
073F:0106 mul ax,bx
073F:0108 int 20h
073F:010A
-t

AX=0012  BX=0000  CX=0000  DX=0000  SP=00FD  BP=0000  SI=0000  DI=0000
DS=073F  ES=073F  SS=073F  CS=073F  IP=0103   NV UP EI PL NZ NA PO NC
073F:0103 BB0500        MOV     BX,0005
-
-t

AX=0012  BX=0005  CX=0000  DX=0000  SP=00FD  BP=0000  SI=0000  DI=0000
DS=073F  ES=073F  SS=073F  CS=073F  IP=0106   NV UP EI PL NZ NA PO NC
073F:0106 F7E0          MUL     AX
-t

AX=0144  BX=0005  CX=0000  DX=0000  SP=00FD  BP=0000  SI=0000  DI=0000
DS=073F  ES=073F  SS=073F  CS=073F  IP=0108   NV UP EI PL NZ NA PO NC
073F:0108 CD20          INT     20
-
```

6) Multiplication of one 16-bit number with one 8-bit number.



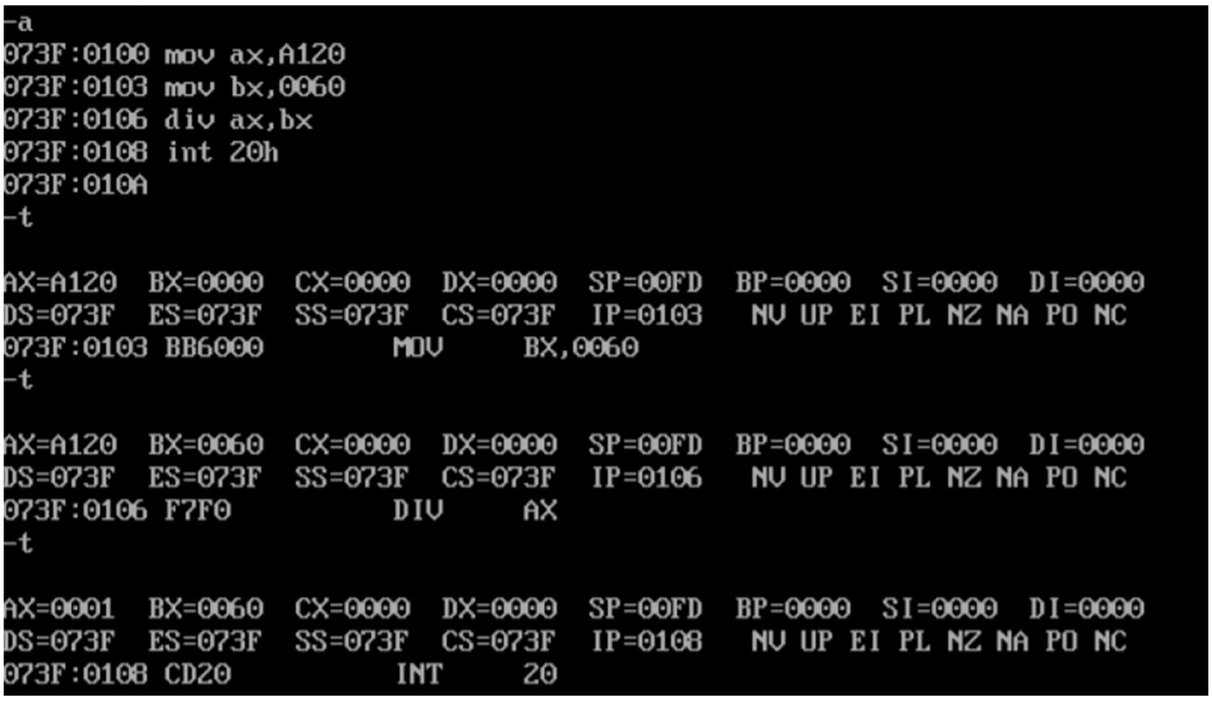
```
-a
073F:0100 mov ax,A002
073F:0103 mov bx.0011
                    ^ Error
073F:0103 mov bx,0011
073F:0106 mul ax,bx
073F:0108 int 20h
073F:010A
-t

AX=A002  BX=0000  CX=0000  DX=0000  SP=00FD  BP=0000  SI=0000  DI=0000
DS=073F  ES=073F  SS=073F  CS=073F  IP=0103   NV UP EI PL NZ NA PO NC
073F:0103 BB1100        MOV     BX,0011
-t

AX=A002  BX=0011  CX=0000  DX=0000  SP=00FD  BP=0000  SI=0000  DI=0000
DS=073F  ES=073F  SS=073F  CS=073F  IP=0106   NV UP EI PL NZ NA PO NC
073F:0106 F7E0          MUL     AX
-t

AX=8004  BX=0011  CX=0000  DX=6402  SP=00FD  BP=0000  SI=0000  DI=0000
DS=073F  ES=073F  SS=073F  CS=073F  IP=0108   OV UP EI PL NZ NA PO CY
073F:0108 CD20          INT     20
```

7) Division of one 16-bit number with one 8-bit number.



```
-a
073F:0100 mov ax,A120
073F:0103 mov bx,0060
073F:0106 div ax,bx
073F:0108 int 20h
073F:010A
-t

AX=A120  BX=0000  CX=0000  DX=0000  SP=00FD  BP=0000  SI=0000  DI=0000
DS=073F  ES=073F  SS=073F  CS=073F  IP=0103   NV UP EI PL NZ NA PO NC
073F:0103 BB6000        MOV     BX,0060
-t

AX=A120  BX=0060  CX=0000  DX=0000  SP=00FD  BP=0000  SI=0000  DI=0000
DS=073F  ES=073F  SS=073F  CS=073F  IP=0106   NV UP EI PL NZ NA PO NC
073F:0106 F7F0          DIV     AX
-t

AX=0001  BX=0060  CX=0000  DX=0000  SP=00FD  BP=0000  SI=0000  DI=0000
DS=073F  ES=073F  SS=073F  CS=073F  IP=0108   NV UP EI PL NZ NA PO NC
073F:0108 CD20          INT     20
```

Conclusion:

Debug helps us to write assembly programs in an ordered manner. We are able to implement arithmetic problems using debug. We just need to learn some commands that are needed to implement programs in debug. So with debug we are able to learn to write and run assembly programs easily.