Mansoura University
Faculty of Computers and Information
Department of Computer Science
First Semester: 2020-2021

[CS214P] Assembly Language

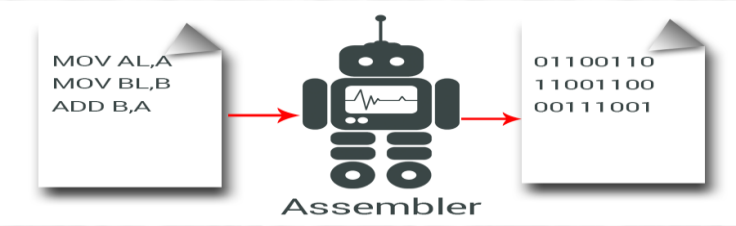
Grade: Third Year (Computer Science)

Sara El-Metwally, Ph.D.

Faculty of Computers and Information,

Mansoura University,

Egypt.



Computer Science Department
Faculty of Computers and Information
Mansoura University

Assembly Language

"Examining Computer Memory and Executing Instructions"

Sara El-Metwally, Ph.D.
Faculty of Computers and Information,
Mansoura University, Egypt.

Email: sarah_almetwally4@mans.edu.eg
sara.elmetwally.2007@gmail.com

Debug Program

- **DEBUG** program allows you to view memory, to enter programs in memory, and to trace their execution.
- **DEBUG** program is used for testing and debugging executable programs.
- **DEBUG** displays all program code and data in hexadecimal format.
- **DEBUG** has a single-step mode, which allows you to execute a program one instruction at a time.

Debug Commands

- **A:** Assemble symbolic instructions into machine code.
- **D:** Display the contents of an area of memory in hex format.
- **E:** Enter data into memory, beginning at a specific location.
- **G:** Run the executable program in memory (G means “Go”).
- **H:** Perform hexadecimal arithmetic.

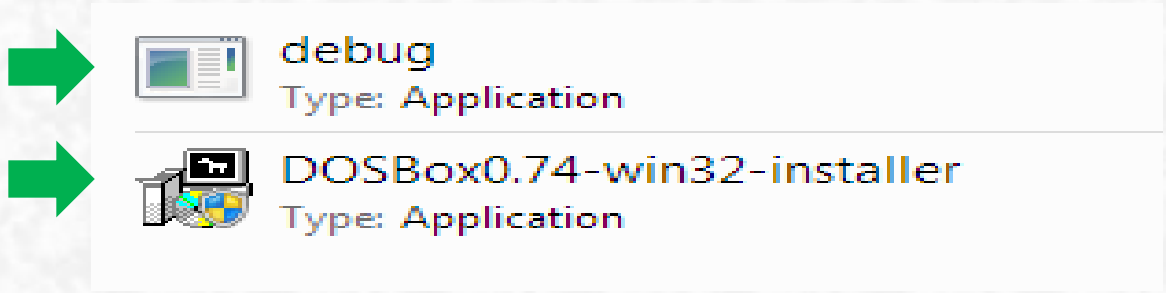
Debug Commands

- **N:** Name a program.
- **P:** Proceed, or execute a set of related instructions.
- **R:** Display the contents of one or more registers in hex format.
- **T:** Trace the execution of one instruction.
- **U:** Unassemble (or disassemble) machine code into symbolic code.

Rules of Debug Commands

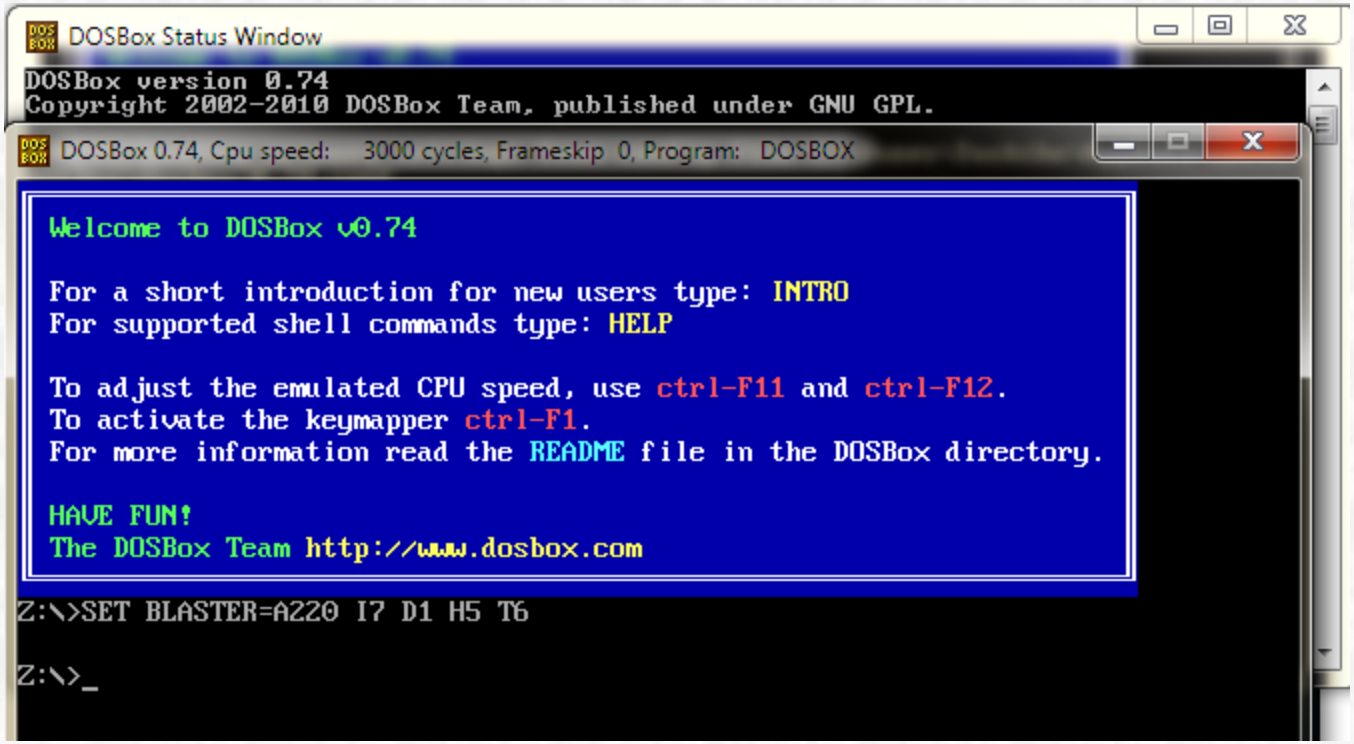
- **DEBUG:** lowercase = uppercase letters.
- **DEBUG:** assumes that all numbers in hexadecimal format.
- **DEBUG:** spaces in commands are used only to separate parameters.
- **DEBUG :** segments and offsets are specified with a colon, in the form segment: offset

Your Working environment



Your Working environment

- Install DOSBox
- Open DOSBox



Your Working environment

- Type:

mount <space> c <space> c:\

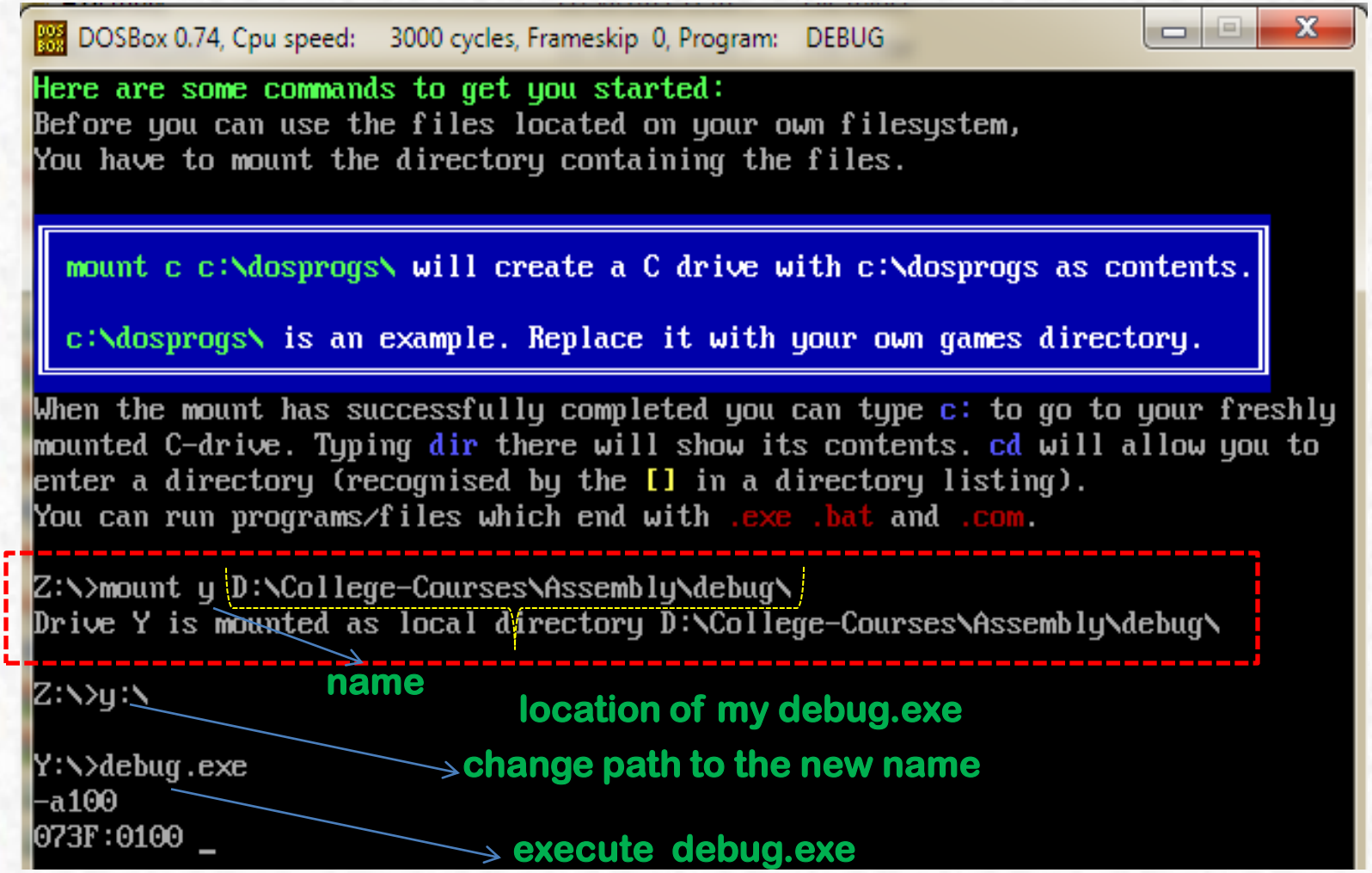
mount <space> c <space> c:\Documents\

could be any chars

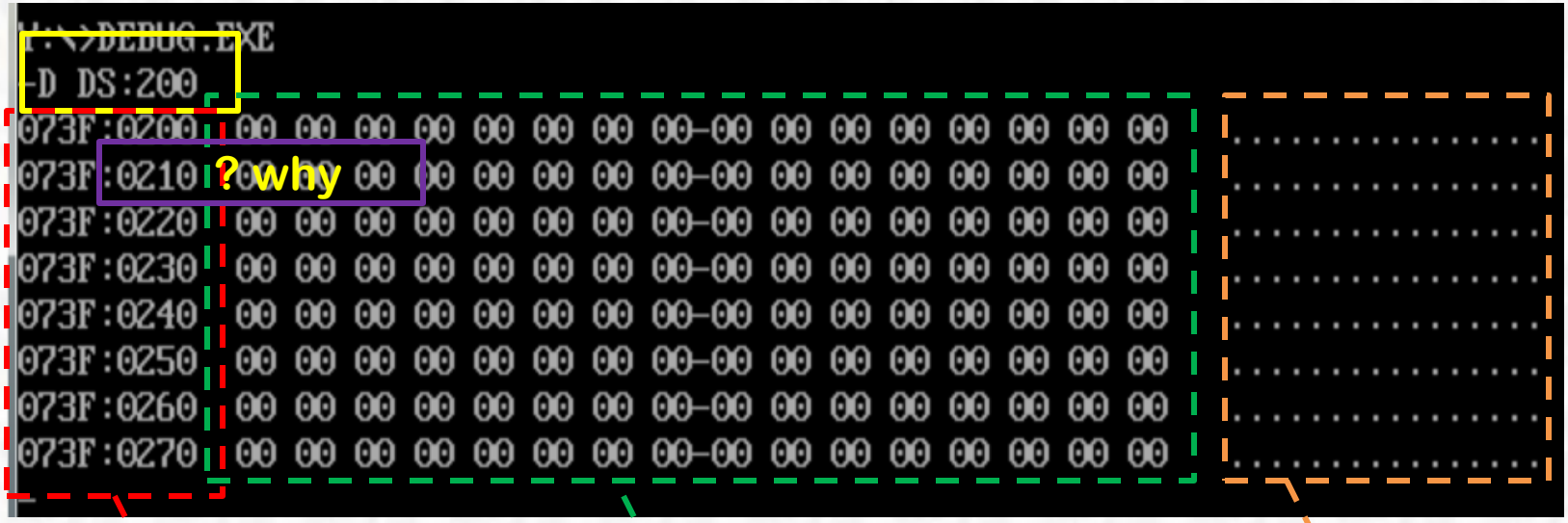
Location of debug.exe

- Then, change the path to the named directory i.e. c .
- Type: debug.exe

Your Working environment



Debug Display Command



The address of left most displayed byte segment : offset

Hex representation of the displayed area.

ASCII representation of the displayed area.

Debug Display Command (BIOS data area, 40[0]H)

Address of first serial port (?)

Address of second serial port (?)

Address of one parallel port (?)

4 words : addresses of serial ports COM1 through COM4.

The first 16 bytes: the addresses of the serial and parallel ports.

4 words : addresses of parallel ports LPT1 through LPT4.

Debug Display Command (BIOS data area, 40[0]H)

Equipment status word
(indication of installed devices)

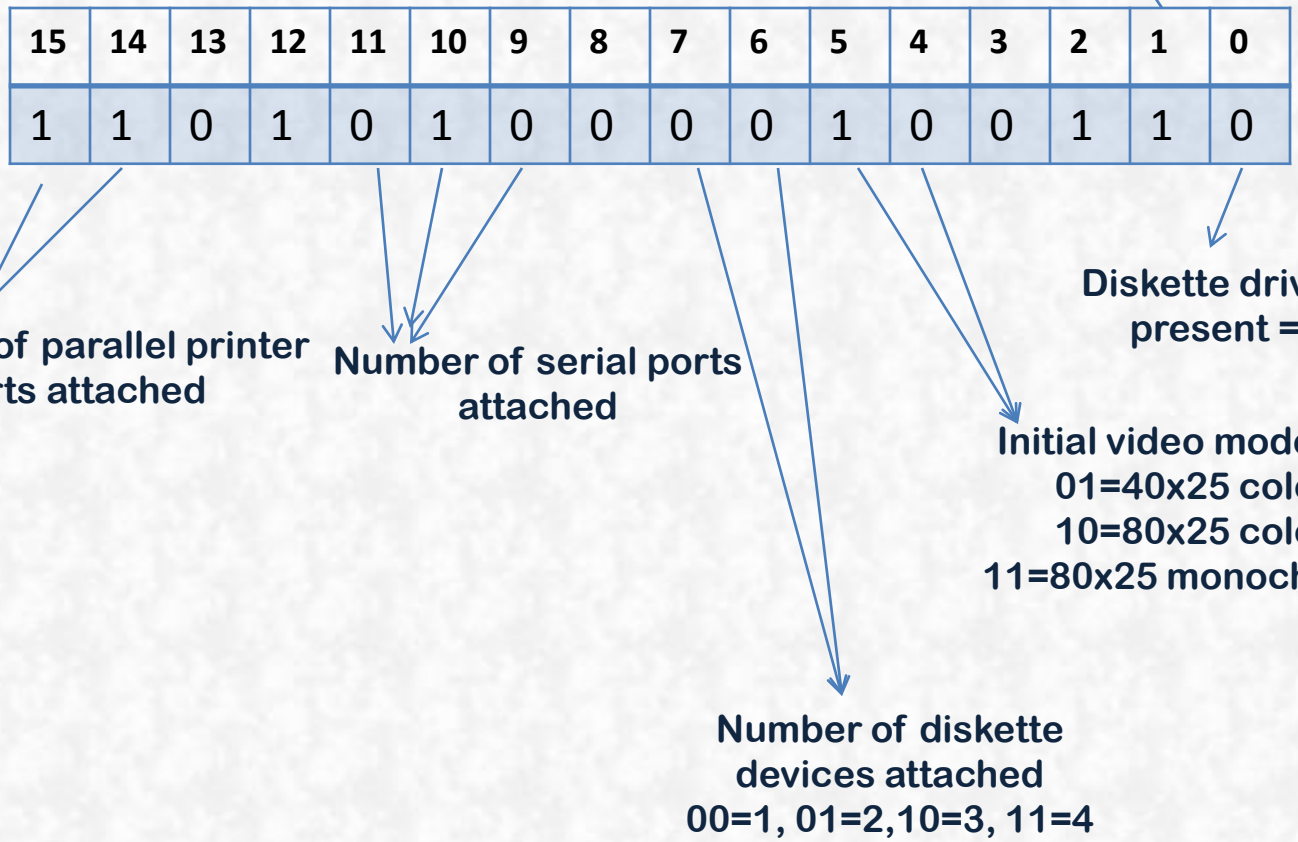
D426

```
-D 40:10
0040:0010  26 D4 00 80 02 00 00 40-00 00 22 00 22 00 30 0B   &......@..".".0.
0040:0020  0D 1C 34 05 30 0B 3A 27-30 0B 30 0B 0D 1C 44 20   ..4.0.:'0.0...D
0040:0030  20 39 31 02 08 0E 34 05-30 0B 3A 27 31 02 00 00    91...4.0.:'1...
0040:0040  00 00 00 00 00 00 00 00-00 03 50 00 00 10 00 00   ..........P.....
0040:0050  00 10 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
0040:0060  07 06 00 D4 03 29 30 00-00 00 00 00 6F 1A 14 00   .....)0.....o...
0040:0070  00 00 00 00 00 02 00 00-01 01 01 00 01 01 01 01   ................
0040:0080  1E 00 3E 00 18 10 00 60-09 51 0B 00 00 00 00 00   ..>....`.Q......
```

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	1	0	1	0	0	0	0	1	0	0	1	1	0

Debug Display Command (BIOS data area, 40[0]H)

D426



Debug Display Command

(ROM BIOS, copyright, FE00:0)

```
-D FE00:0000
FE00:0000  00 00 00 00 00 00 00 00-00 00 00 00 00 00 49 42   ..............IB
FE00:0010  4D 20 43 4F 4D 50 41 54-49 42 4C 45 20 34 38 36   M COMPATIBLE 486
FE00:0020  20 42 49 4F 53 20 43 4F-50 59 52 49 47 48 54 20    BIOS COPYRIGHT
FE00:0030  54 68 65 20 44 4F 53 42-6F 78 20 54 65 61 6D 2E   The DOSBox Team.
FE00:0040  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
FE00:0050  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   ................
FE00:0060  00 44 4F 53 42 6F 78 20-46 61 6B 65 42 49 4F 53   .DOSBox FakeBIOS
FE00:0070  20 76 31 2E 30 00 00 00-00 00 00 00 00 00 00 00    v1.0...........
-
```

Debug Display Command

(ROM BIOS Date, FFFF:5)

```
-D FFFF:0005
FFFF:0000                30 31 2F-30 31 2F 39 32 00 FC 55      01/01/92..U
FFFF:0010  60 10 00 F0 08 00 70 00-08 00 70 00 08 00 70 00    `.....p...p...p.
FFFF:0020  08 00 70 00 60 10 00 F0-60 10 00 F0 60 10 00 F0    ..p.`...`...`...
FFFF:0030  A5 FE 00 F0 87 E9 00 F0-55 FF 00 F0 60 10 00 F0    ........U...`...
FFFF:0040  60 10 00 F0 60 10 00 F0-80 10 00 F0 60 10 00 F0    `...`.......`...
FFFF:0050  00 13 00 F0 00 11 00 F0-20 11 00 F0 40 11 00 F0    ........ ...@...
FFFF:0060  A0 11 00 F0 C0 11 00 F0-E0 11 00 F0 20 12 00 F0    ............ ...
FFFF:0070  C0 12 00 F0 C0 12 00 F0-40 12 00 F0 60 10 00 F0    ........@...`...
FFFF:0080  60 12 00 F0 A4                                     `....
-
```

Machine Language Example

(Using Immediate Data)

Machine Instruction	Symbolic Code
B82301	MOV AX, 0123
052500	ADD AX, 0025
8BD8	MOV BX,AX
03D8	ADD BX,AX
8BCB	MOV CX,BX
2BC8	SUB CX,AX
2BC0	SUB AX,AX
EBEE	JMP 100

Machine Language Example

(Using Immediate Data)

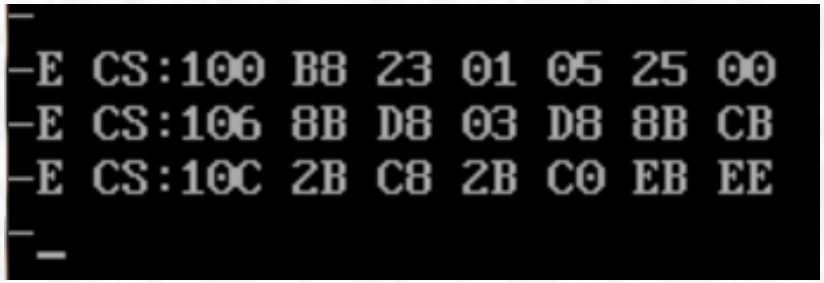
```
B82301
052500
8BD8
03D8
8BCB
2BC8
2BC0
EBEE
```

Machine Instruction

First 6 bytes of machine codes starting from 100 ending with 105

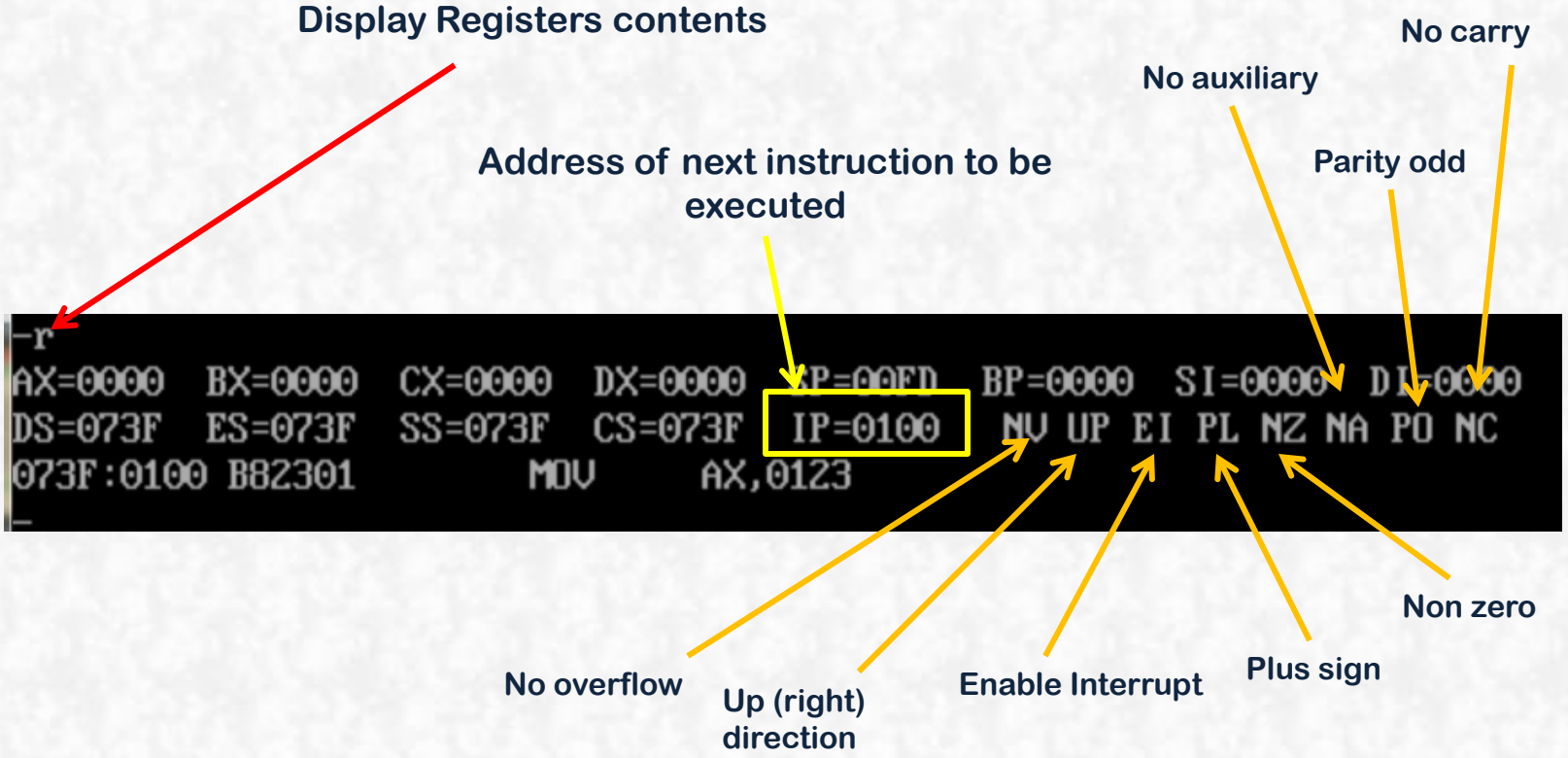
Starting memory address CS:100

Enter data into memory beginning at specific location



Machine Language Example

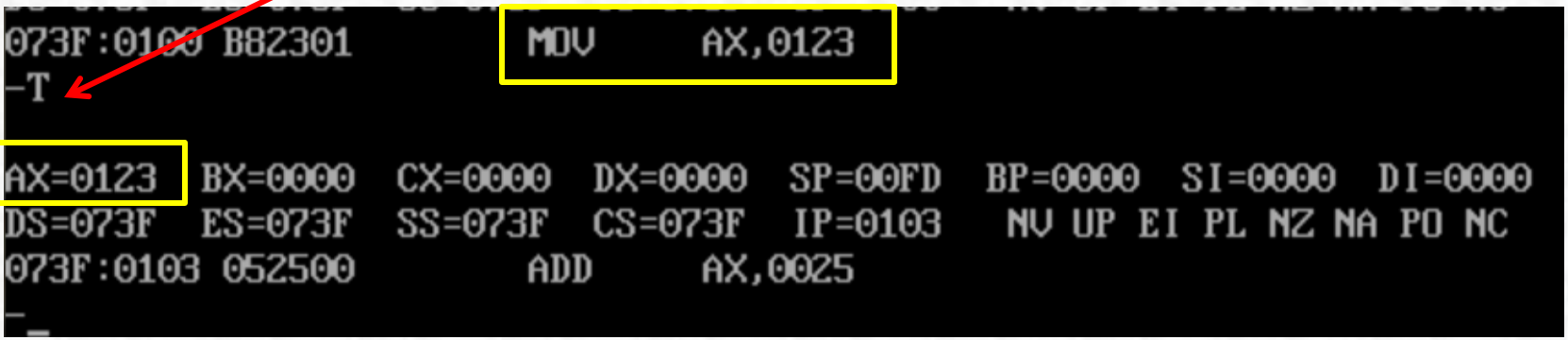
(Using Immediate Data)



Machine Language Example

(Using Immediate Data)

Trace the execution of one instruction

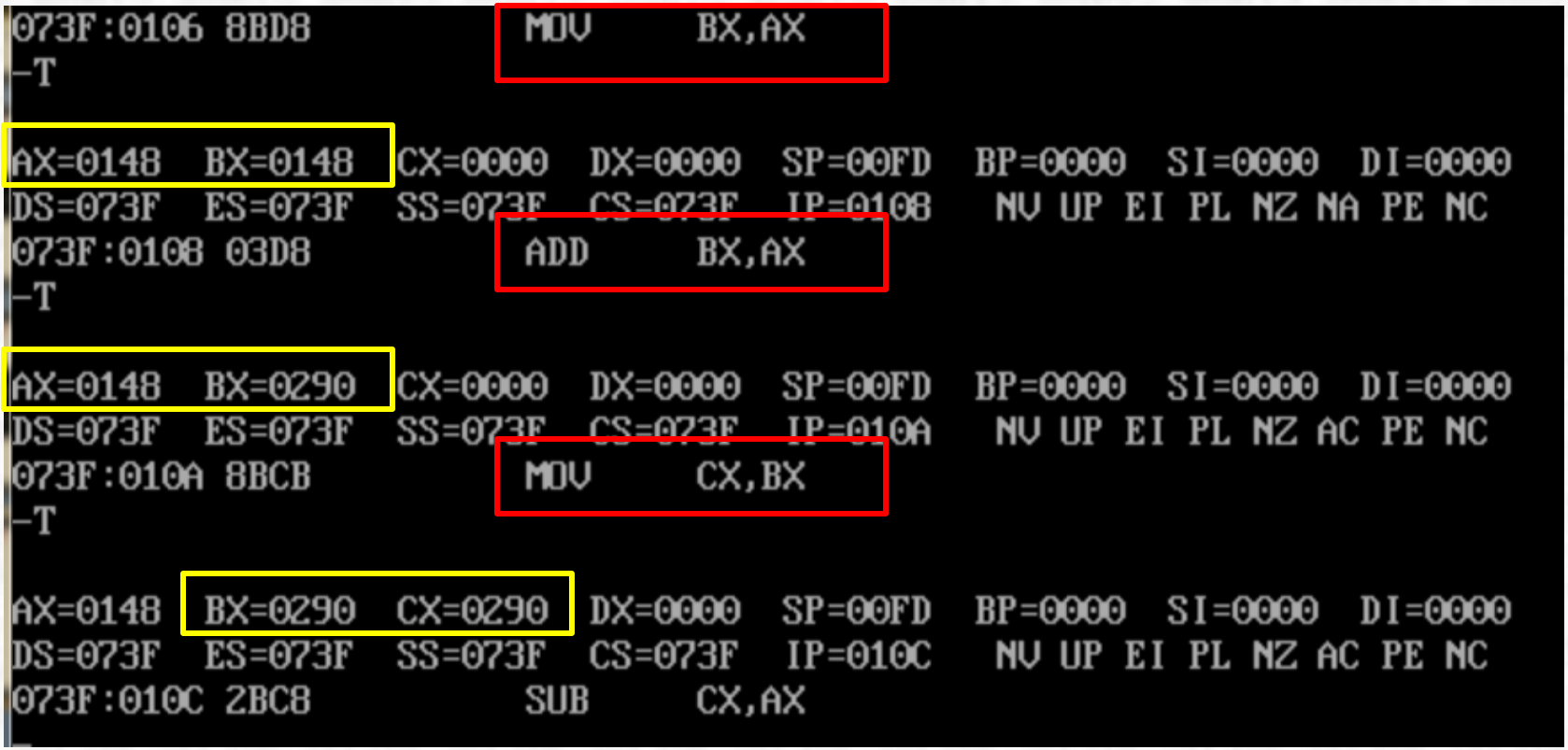


Machine Language Example

(Using Immediate Data)

Machine Language Example

(Using Immediate Data)



Reset IP value

(R IP)


```
Y:\>DEBUG.EXE
-R IP
IP 0100
:0200
-R
AX=0000  BX=0000  CX=0000  DX=0000  SP=00FD  BP=0000  SI=0000  DI=0000
DS=073F  ES=073F  SS=073F  CS=073F  IP=0200   NV UP EI PL NZ NA PO NC
```

Assembly Language Program

(A: assemble command)

```
-A 100
073F:0100
```

A command tells DEBUG to begin accepting symbolic assembly instructions and to convert them into machine code.

```
-A 200
073F:0200 MOV CL,42
073F:0202 MOV DL,2A
073F:0204 ADD CL,DL
073F:0206 JMP 100
073F:0209
-G=200 206

AX=0000  BX=0000  CX=006C  DX=002A  SP=00FD  BP=0000  SI=0000  DI=0000
DS=073F  ES=073F  SS=073F  CS=073F  IP=0206   NV UP EI PL NZ NA PE NC
073F:0206 E9F7FE        JMP     0100
-
```

Assembly Language Program

(U: unassemble command)