BCN 1043

COMPUTER ARCHITECTURE & ORGANIZATION

By
Dr. Mritha Ramalingam

Faculty of Computer Systems & Software Engineering
mritha@ump.edu.my

AUTHORS

- **Dr. Mohd Nizam Mohmad Kahar** (mnizam@ump.edu.my)
- **Jamaludin Sallim** (jamal@ump.edu.my)
- **Dr. Syafiq Fauzi Kamarulzaman** (syafiq29@ump.edu.my)
- **Dr. Mritha Ramalingam** (mritha@ump.edu.my)

Faculty of Computer Systems & Software Engineering

BCN 1043

COMPUTER ARCHITECTURE & ORGANIZATION

Chapter 4 continues…

CHAPTER 4

ASSEMBLY LANGUAGE

- Introduction
- The Computer Organization
- Instruction Format
- Addressing Mode
- DEBUG program

DEBUG Program

- The DEBUG program is used for testing and debugging executable programs which include to:

    1. viewing the content of the main memory (MM)
    2. enter programs in memory
    3. trace the execution of a program

- DEBUG also provides a single-step mode, which allows you to execute a program one instruction at a time, so that you can view the effect of each instruction on memory locations and registers.

DEBUG Commands

- The following are some 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

N : Name a program

P : Proceed or execute a set of related instructions

Q : Quit the DEBUG session

R : Display the contents of one or more registers in hex format

T : Trace the execution of one instruction

U : disassemble machine code into symbolic code

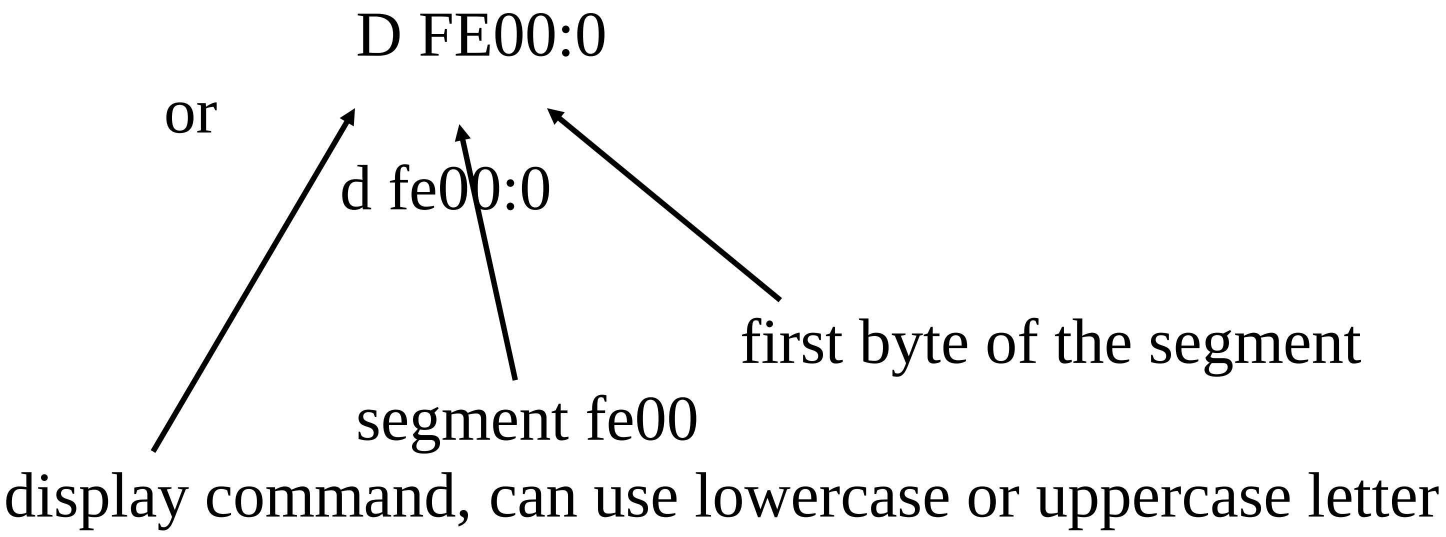
- **Note** : refer *appendix C* (from main reference) pg 513-519 for complete DEBUG commands

Rules of DEBUG Commands

- DEBUG does not distinguish between lowercase and uppercase letters.
- DEBUG assumes that all numbers are in hexadecimal format
- Spaces in commands are used only to separate parameters
- Segments and offset are specified with a colon, in the form segment:offset

Example :

To display the content in segment FE00$_{16}$ beginning from the first byte of the segment, in the DEBUG mode, type:



D FE00:0

or

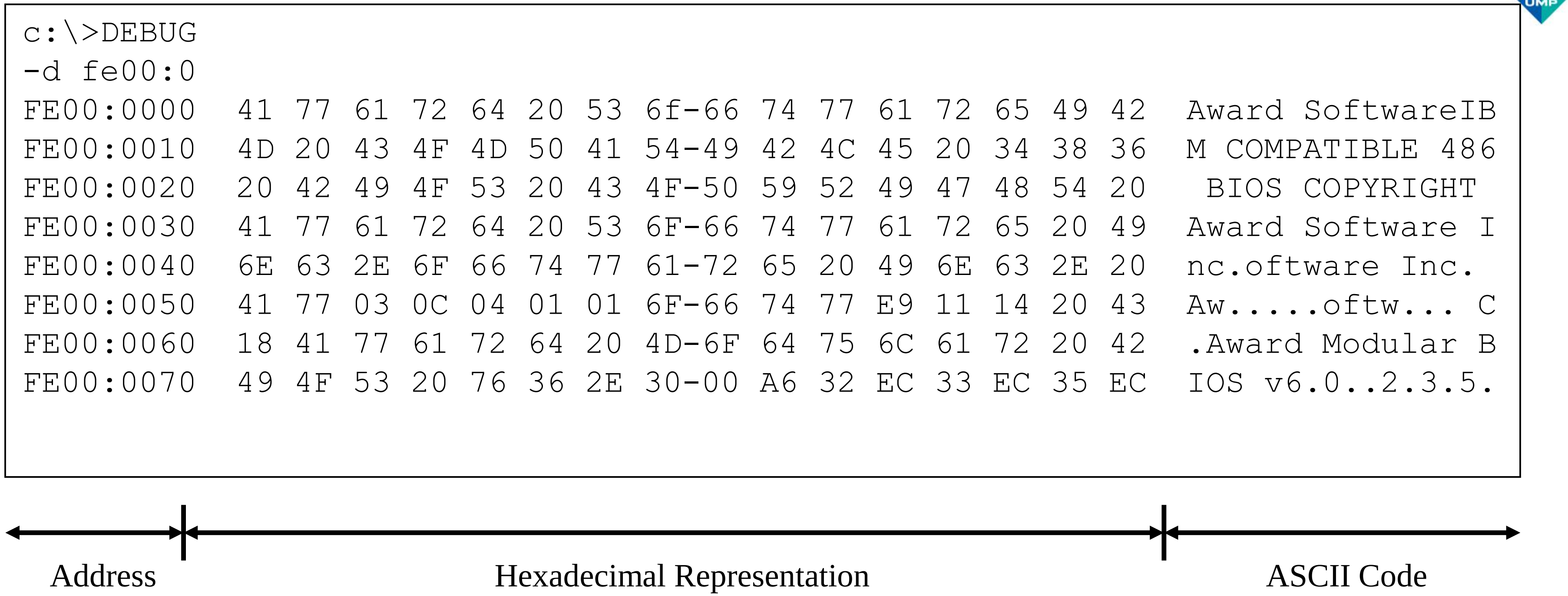
d fe00:0

first byte of the segment

segment fe00

display command, can use lowercase or uppercase letter

- the command d or D (D ⇒ Display) will display 8 rows of data and each row contains 16 bytes (32 digit hex) which adds up to a total of 128 bytes (8 rows), beginning from the address given



```
c:\>DEBUG
-d fe00:0
FE00:0000  41 77 61 72 64 20 53 6f-66 74 77 61 72 65 49 42  Award SoftwareIB
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  41 77 61 72 64 20 53 6F-66 74 77 61 72 65 20 49  Award Software I
FE00:0040  6E 63 2E 6F 66 74 77 61-72 65 20 49 6E 63 2E 20  nc.oftware Inc.
FE00:0050  41 77 03 0C 04 01 01 6F-66 74 77 E9 11 14 20 43  Aw.....oftw... C
FE00:0060  18 41 77 61 72 64 20 4D-6F 64 75 6C 61 72 20 42  .Award Modular B
FE00:0070  49 4F 53 20 76 36 2E 30-00 A6 32 EC 33 EC 35 EC  IOS v6.0..2.3.5.
```

Address

Hexadecimal Representation

ASCII Code

Machine Language Example: Using Immediate Data (Immediate Mode)

- the DEBUG program can also be used to enter the program code into the memory and trace its execution
- Below is an example of a program in machine language (written in hexadecimal) and assembly language (symbolic code) together with description about the instructions

Machine Instruction	Symbolic Code (Assembly Language)	Explanation
B82301	MOV AX,0123	Move value 0123H to AX
052500	ADD AX,0025	Add value 0025H to AX
8BD8	MOV BX,AX	Move contents of AX to BX
03D8	ADD BX,AX	Add contents of AX to BX
8BCB	MOV CX,BX	Move contents of BX to CX
2BC8	SUB CX,AX	Subtract contents of AX from CX
2BC0	SUB AX,AX	Subtract AX from AX
EBEE	JMP 100	Go back to the start

- - The first and second instructions in the program above use immediate addressing mode where the real data value is in the address field

```
 MOV  AX, 0123
ADD   AX, 002
```

Other instructions use register addressing mode (general purpose registers)

Assembly Language Example

- Assembly Language program can be written or entered into the memory using the DEBUG command of “A” or “a”. (A ⇒ Assemble)

- Example:

MOV CL, 42 (enter the value of 42H into the CL register)

MOV DL, 2A (enter the value of 2AH into the DL register)

ADD CL, DL (add the value in the CL register with the value in the DL register and keep the result in the CL register)

Enter the program above into the memory:

```
-A 100
2090 : 0100     MOV CL, 42
2090 : 0102     MOV DL, 2A
2090 : 0104     ADD CL, DL
2090 : 0106     JMP 100
2090 : 0108
```

To view machine code for the assembly language entered, use the “u” or “U” command. (U ⇒ Un-assemble)

```
-U  100, 107
2090 : 0100    B142                 MOV        CL, 42
2090 : 0102    B22A                 MOV        DL, 2A
2090 : 0104    00D1                 ADD        CL, DL
2090 : 0106    EBF8                 JMP        0100
```

↑

the machine code for the instruction entered

- To execute the above program, as usual, use the “r” or “R” command followed by the “T” or “t” command.

Example 1: Getting the Current Date and Time

- the instruction to access the current date is INT 21H function code 2AH. The function code 2AH must be moved to AH register. The instructions are as the following:

```
MOV AH, 2A
INT 21
JMP 100
```

- use command A to enter the above instructions into the code segment.

- type R to display the registers and T to execute the MOV.

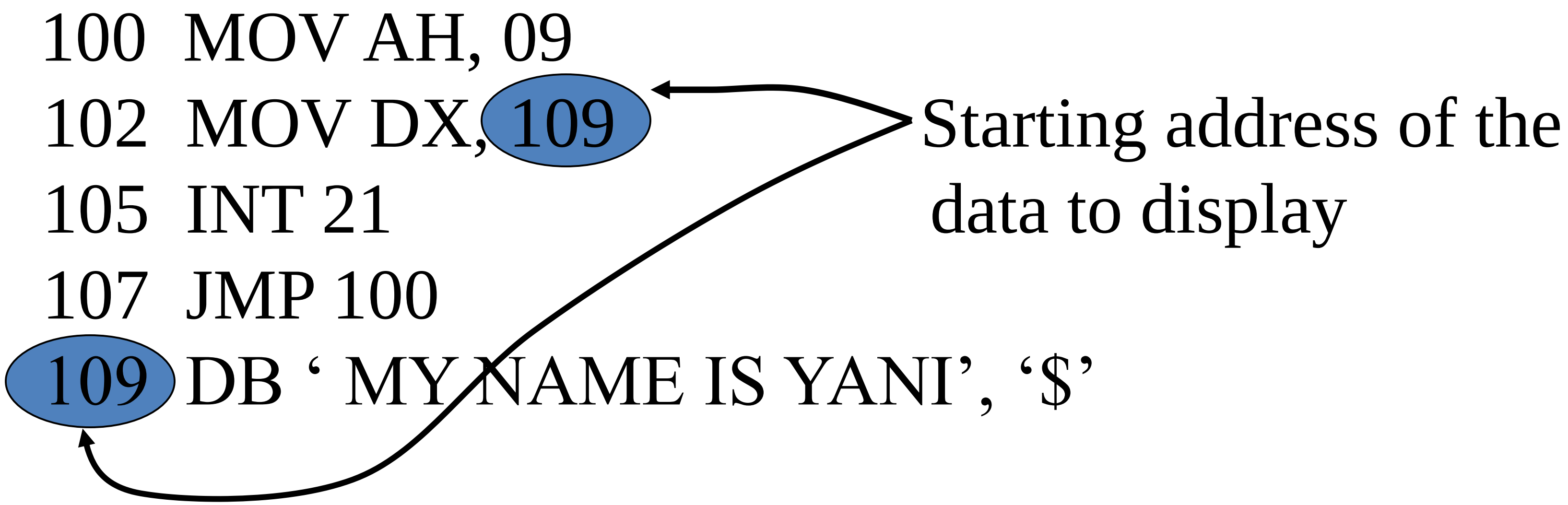
- type P to proceed directly through the interrupt routine; the operation stops at the JMP.

- - the registers contain the following information in hex format:

AL: Day of the week, where 0 = Sunday
CX: Year (for example, 07D4H = 2004)
DH: Month (01H through 0CH)
DL: Day of the month (01H through 1FH)

Example 2: Displaying

- to display data on screen.
- enter the following instructions using A 100 command.



```
100  MOV AH, 09
102  MOV DX, 109
105  INT 21
107  JMP 100
109  DB ' MY NAME IS YANI', '$'
```

Starting address of the data to display

- key in R to display the registers and first instruction, and key in T commands for the two MOVs. Key in P to execute INT 21 and MY NAME IS YANI will display on the screen.

Example 3: Keyboard Input

- to accept characters from the keyboard.
- Type in the DEBUG command A and then these assembly instructions:

```
100   MOV AH, 10
 102  INT 16
 104  JMP 100           <enter> twice
```

- the first instruction, MOV, provides function code 10H that tells INT 16H to accept data from the keyboard.

Chapter 4 Review

- **Introduction**
- **The Computer Organization**
- **Instruction Format**
- **Addressing Mode**
- **DEBUG program**

Chapter 4 ends!