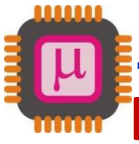

8086

Introduction to Assembly Language Programming

- ***Machine Language***
  - Programs consist of 0s and 1s are called machine language.
  - ***Assembly Language*** provides ***mnemonics*** for machine code instructions.
  - ***Mnemonics*** refer to codes and abbreviations to make it easier for the users to remember.

- ***Low/High-level languages***
  - Assembly Language is a low-level language. Deals directly with the internal structure of CPU. ***Assembler*** translates Assembly language program into machine code.
  - In high-level languages, Pascal, Basic, C; the programmer does not have to be concerned with internal details of the CPU. ***Compilers*** translate the program into machine code.

Introduction to Assembly Language Programming

- ***Machine Language***
  - ❖ Programs consist of 0s and 1s are called machine language.
  - ❖ Assembly Language program consists of series of lines of Assembly language ***instructions***.
  - ❖ ***Instructions*** consist of a ***mnemonic*** and ***operand(s).***

- ***MOV instruction***

MOV destination, source; *copy source operand to destination*

mnemonic *operands*

Introduction to Assembly Language Programming

- ***MOV instruction***

Example: (8-bit)

Comment sign

```
MOV     CL,55H    ;move 55H into register CL
MOV     DL,CL     ;move/copy the contents of CL into DL (now DL=CL=55H)
MOV     BH,DL     ;move/copy the contents of DL into BH (now DL=BH=55H)
MOV     AH,BH     ;move/copy the contents of BH into AH (now AH=BH=55H)
```

Example: (16-bit)

```
MOV     CX,468FH  ;move 468FH into CX (now CH =46 , CL=8F)
MOV     AX,CX     ;move/copy the contents of CX into AX (now AX=CX=468FH)
MOV     BX,AX     ;now BX=AX=468FH
MOV     DX,BX     ;now DX=BX=468FH
MOV     DI,AX     ;now DI=AX=468FH
MOV     SI,DI     ;now SI=DI=468FH
MOV     DS,SI     ;now DS=SI=468FH
MOV     BP,DS     ;now BP=DS=468FH
```

Introduction to Assembly Language Programming

- ***MOV instruction***
  - **Rules regarding MOV instruction**
    - Data can be moved among all registers except the ***flag*** register. There are other ways to load the flag registers. To be studied later.
    - Source and destination registers have to ***match in size***.
    - Data can be moved among all registers (except flag reg.) but data can be moved ***directly*** into ***nonsegment*** registers only. You can't move data segment registers directly.

Examples:

```
MOV     BX,14AFH        ;move 14AFH into BX                    (legal)
MOV     SI,2345H        ;move 2345H into SI                    (legal)
MOV     DI,2233H        ;move 2233H into DI                    (legal)
MOV     CS,2A3FH        ;move 2A3FH into CS                    (illegal)
MOV     DS,CS           ;move the content of CS into DS        (legal)
MOV     FR,BX           ;move the content of BX into FR        (illegal)
MOV     DS,14AFH        ;move 14AFH into DS                    (illegal)
```

Introduction to Assembly Language Programming

- ***MOV instruction***
  - **Important points**
    - ❖ Data values cannot be loaded directly into (***CS***,***DS***,***SS*** and ***ES***)

```
MOV     AX,1234H        ; load 1234H into AX
MOV     SS,AX           ; load the value in AX into SS
```

    - ❖ Sizes of the values:

```
MOV     BX,2H           ; BX=0002H,        BL:02H,  BH:00H

MOV     AL,123H         ; illegal (larger than 1 byte)
MOV     AX,3AFF21H      ; illegal (larger than 2 bytes)
```