Assembly language programming

Introduction to 8086 Programming

The 8086 microprocessor is one of the family of 8086,80286,80386,80486,Pentium,PentiumI,II,III …. Also referred to as the X86 family.)

Learning any imperative programming language involves

mastering a number of common concepts:

Conti.

- Variables: declaration/definition
- Assignment: assigning values to variables
- Input/Output: Displaying messages
- Displaying variable values
- Control flow: if-then
- Loops
- Subprograms: Definition and Usage

Conti.

The 8086 has 14 registers. Each of these is a 16-bit register.

Initially, we will use four of them – the so called the general

purpose registers:

ax, bx, cx, dx

These four 16-bit registers can also be treated as eight 8-bit

registers:

ah, al, bh, bl, ch, cl, dh, dl

Assignment

In Java, assignment takes the form:

x = 42 ;

y = 24;

z = x + y;

In assembly language we carry out the same operation but we use an instruction to denote the assignment operator ("=" in Java). The above assignments would be carried out in 8086 assembly langauge as follows

mov x, 42

mov y, 24

add z, x

add z, y

The mov instruction carries out assignment

Example

Store the ASCII code for the letter A in register bx.

mov bx, ‘A’

The mov instruction also allows you to copy the contents of one register into another register.

Example:

mov bx, 2

mov cx, bx

Explanation

The first instruction loads the value 2 into bx where it is stored as a binary number. [a number such as 2 is called an integer constant]

The Mov instruction takes two operands, representing the destination where data is to be placed and the source of that data.

General Form of Mov Instruction

mov destination, source

conti.

where destination must be either a register or memory location and source may be a constant, another register or a memory location.

Note: The comma is essential. It is used to separate the two operands.

A missing comma is a common syntax error.

Comments:

Anything that follows semi-colon (;) is ignored by the assembler. It is called a comment. Comments are used to

make your programs readable. You use them to explain what you are doing in English.

8086 PROGRAM FOR ADD and SUB.

ADDITION:

```
MOV AX,05
MOV BX,03
ADD AX,BX
MOV SI,8000
MOV [SI],AX
INT 03
```

SUBTRACTION:

```
MOV AX,05
MOV BX,03
SUB AX,BX
MOV SI,8000
MOV [SI],AX
INT 03
```

8086 PROGRAM FOR MUL AND DIV

MULTIPLICATION:

```
MOV AX,05
MOV BX,03
MUL BX
MOV SI,8000
MOV [SI],AX
INT 03
```

DIVISION:

```
MOV AX,05
MOV BX,03
DIV BX
MOV SI,8000
MOV [SI],AX
INT 03
```

Example

Write a code fragment to display the character ’a’ on the screen:

```
mov dl, ‘a‘  ;  dl = ‘a‘
mov ah, 2h ; character output subprogram
int 21h        ; call ms-dos output character
```

As you can see, this simple task is quite complicated in assembly language.

Example 2

Write a code fragment to read a character from the keyboard:

```
mov  ah, 1h ; keyboard input subprogram
int    21h      ; character input character is stored in al
```

The following example combines the two previous ones, by

reading a character from the keyboard and displaying it.

Example 3:

- Reading and displaying a character:
- mov ah, 1h ; keyboard input subprogram
- int 21h ; read character into al
- mov dl, al ; copy character to dl
- mov ah, 2h ; character output subprogram
- int 21h ; display character in dl

Conti.

- The first instruction loads the value 2 into bx where it is stored as a binary number. [a number such as 2 is called an integer constant]