Digital Locha

The world of Digital computing explored by Tushar B. Kute -http://tusharkute.com

Home ▼

Monday, January 6, 2014

Print a number using Assembly Language

Let's see something simple but interesting stuff in assembly language. To print a value on screen, various high level programming languages use there library function and statement. For example, C uses 'printf', C++ uses 'cout', Java uses 'println' and Python uses 'print'. It is single line statements that does our task. But, microprocessor implements several sequential steps to print a number on screen.

Here, I have used the widely used MASM to demonstrate this work. Basically, to print a number microprocessor don't have any function for this. It is only possible to print a single character with function number 02 an interrupt number 21h. The character that you have to print, must be present in DL register.
For example,

```
MOV DL,'a'
MOV AH,02H
INT 21H
```

This code will print 'a' on the screen (without quote!). Here DL register of processor is storing ASCII value of character 'a'. So, following code will also do the same task, as ASCII value of character 'a' is 41 in hexadecimal.

```
MOV DL,41H
MOV AH,02H
INT 21H
```

Here, 02 is function number and it is necessary to store it in AH register before invoking an interrupt. Now, to print number on screen it is not directly possible using assembly language. We need to do it using ASCII values of hexadecimal digit. Remember ASCII values of 0 to 9 are 30 to 39 respectively. So, if you want to print 5 on screen, we need to store 35 in DL register before invoking the interrupt. Now, following code will print 5 on screen.

```
MOV DL,35H
MOV AH,02H
INT 21H
```

It means that, we need code conversion by adding 30 in hex to the respective single digit number. If you want to print 2 digit number the same procedure can be followed for both digit by rotating. Then one after the another following algorithm will do the task for us.

1. Get a two digit number in temporary register say BH.
2. Rotate the bits of the number by 4 position so hex digit would be swapped.
3. Copy the rotated number in DL register.
4. Mask high order 4 bits to zero.
5. Add 30h in DL.
6. Use function 02h and int 21h to print the number.
7. Repeat the steps 2 to 6 for second digit also.

Code:

```
MOV BH,96H ;number to print
MOV CH,02H ;number of digits
MOV CL,04H ;rotation count
UP:ROL BH,CL ;swap digits
MOV DL,BH
AND DL,0FH ;mask MSB digit
ADD DL,30H ;add 30 in DL
MOV AH,02H ;function number
```

```
 9. INT 21H
10. DEC CH ;do twice
11. JNZ UP
```

Here, we simulate the code stepwise.

1. BH=96H
2. CH=02H
3. CL=04H
4. BH=69H
5. DL=69H
6. DL=09H
7. DL=39H
8. AH=02H
9. It will print DL=39 i.e 9 on screen.
10. CH=01H
11. Condition true as ZF=0. Jump to 4th Statement.

4. BH=96H
5. DL=96H
6. DL=06H
7. DL=36H
8. AH=02H
9. It will print DL=36 i.e 6 on screen.
10. CH=00H
11. Condition false so terminate.

Now, here we have printed the hex digits only between 0-9. But, if digits are between A-F then we need to check this condition also. The ASCII value of numbers and characters have difference of 07H. So, if the hex digit of number is greater than 9 then to have to add 07H also in DL register. The following code will do the task.

```
MOV BH,5CH ;number to print
MOV CH,02H
MOV CL,04H
UP:
ROL BH,CL
MOV DL,BH
AND DL,0FH
```