

Q No. 1 Briefly answer each of the following: [6 x 2 + 1 x 4 = 16 points]

- (i) Differentiate between SAR and SHL instructions with one example each.

Shift Arithmetic Right shift destination to the right preserving the MSB (sign-bit) i.e. insertion from left will be the value of MSB. LSB is move to CF as well. Shift Left shifts destination to the left and zero are inserted from the right. MSB is moved to CF in this case. (example as per book and lecture slides)

- (ii) Using shift and add instructions multiply a decimal number  $X_{10}$  by  $23_{10}$ . Assume that the result does not exceed the range of a 16-bit register.

$$X * 23 \rightarrow X * (16 + 8 - 1) \rightarrow X * (2^4 + 2^3 - 1) \rightarrow (X \ll 4) + (X \ll 3) + (-X)$$

```
MOV EAX, X
MOV EBX, 0
SHL EAX, 3 ; X << 3
ADD EBX, EAX
SHL EAX, 1 ; X << 4 (already shift 3 so shifted only one more)
ADD EBX, EAX
NEG X
ADD EBX, X ; subtract X but we have to use ADD instruction only.
```

- (iii) Why an assembly procedure cannot be implemented using JMP instructions?

JMP instruction does not push return address on top of the stack. Programmer can keep track of the return address while program is running first before making procedure call using JMP and later return to that address.

- (iv) Why stack parameters are considered more convenient than register parameters. Justify your answer with examples.

Register parameter are very fast way to pass parameters but are few in numbers. Therefore, parameters are passed through stack. The use of stack also facilitate creation of local variables during the procedure lifetime and automatically cleared upon return. The use of stack reserve stack space for parameters, return address and local variables called stack frame. This concept also support recursion.