Assignment 1

Microprocessor Interfacing and Programming

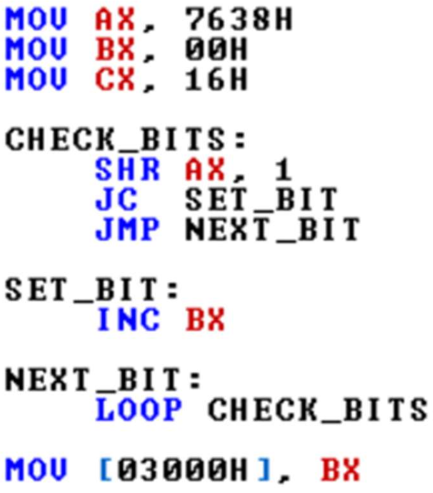
1) **Demonstrate** using assembly language code how to count the number of set bits in a given hex number
2) **Demonstrate** using assembly language code how to divide a given number by 16 without using the DIV instruction.

3) **Illustrate** how to determine if a provided number is a power of 2 using assembly language code.

4) **Show** the process of identifying the position of the first set bit (Most significant set bit) in a given number using assembly language code.

Note1: Explain the flow and logic of each code (in a paragraph or bullets)
Note2: All the codes should be from the 8086 Instruction set

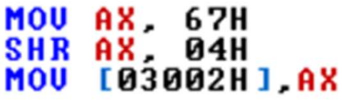
Q1



```
MOV AX, 7638H
MOV BX, 00H
MOV CX, 16H

CHECK_BITS:
    SHR AX, 1
    JC  SET_BIT
    JMP NEXT_BIT

SET_BIT:
    INC BX

NEXT_BIT:
    LOOP CHECK_BITS

MOV [03000H], BX
```

- Move the 16-bit number into AX
- Set BX (bit counter) to 0
- CX (loop counter=> 16-bit number)
- Shift AX right by 1 (moving the LSB into carry flag)
- If carry flag is set, bit is 1 and BX is incremented
- If carry flag is clear, bit is 0 and code continues the loop till CX becomes 0
- BX i.e count of set bits is stored in memory [03000H]

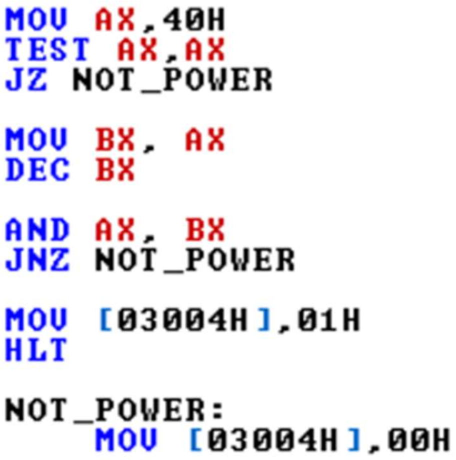
Q2



```
MOV AX, 67H
SHR AX, 04H
MOV [03002H],AX
```

- Move the 16-bit number into AX
- In Hexadecimal, if we shift left once, the number is multiplied by 2. Similarly, if we shift right once, the number is divided by 2
- Therfore, to divide the number by 16, we shift right the number 4 times i.e 2^4=16
- AX i.e result after division is stored in memory [03002H]

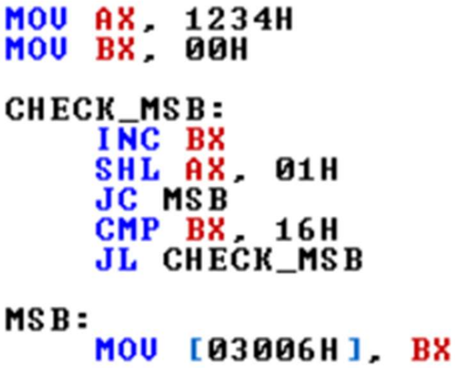
Q3



```
MOV AX,40H
TEST AX,AX
JZ NOT_POWER

MOV BX, AX
DEC BX

AND AX, BX
JNZ NOT_POWER

MOV [03004H],01H
HLT

NOT_POWER:
    MOV [03004H],00H
```

- Move the 16-bit number into AX
- By testing number with itself, we check if it is 0 or not. If it is 0, it is not power of 2, hence 0 is stored at memory [03004H]
- AX is copied to BX and BX is decremented
- For a power of 2, n & (n - 1) will always be 0. This is because subtracting 1 from a power of 2 will turn off the single set bit and turn on all the lower bits.
- If number is power of 2, 1 is stored at memory [03004H]

Q4



```
MOV AX, 1234H
MOV BX, 00H

CHECK_MSB:
    INC BX
    SHL AX, 01H
    JC MSB
    CMP BX, 16H
    JL CHECK_MSB

MSB:
    MOV [03006H], BX
```

- Move the 16-bit number into AX
- Set BX (bit counter) to 0
- BX is incremented. It shows the position of current bit
- AX is shift left by 1 (moving LSB into carry)
- If carry flag is clear, bit is 0
- If carry flag is set, bit is 1 (MSB found), code jumps to MSB
- Compare checks if we've shifted 16 times (max for 16-bit number). If not, code continues finding the MSB
- BX i.e position of most significant set bit is stored in memory [03006H]