

COE 205, Term 991

Computer Organization & Assembly Programming

HW# 3

Due date: Wednesday, Oct. 13

Q.1. Consider a program that has the following data segment:

```
I    EQU 12h
J    DB  `12`
K    DW  1234h
```

Indicate whether the following are valid 8086 instructions. If invalid, give the reason:

- | | |
|---------------|-----------------------|
| 1. MOV AX, BL | 9. MOV CX, [AX] |
| 2. MOV I, BL | 10. INC [BX] |
| 3. MOV K, BL | 11. MOV ES, DS |
| 4. MOV AX, J | 12. INC J |
| 5. MOV K, J | 13. DEC [offset K] |
| 6. MOV AX, I | 14. ADD [BX], 120H |
| 7. MOV J, I | 15. SUB AH, [BX+SI-2] |
| 8. MOV DS, I | 16. MUL 10 |

Q.2. Suppose that you have the following initial content of the registers and memory locations:

AX=0010H	BX=0012H	CX=0312H	DX=00FFH
SI =0010H	DI =0002H	DS=2000H	

Memory Address	Contents (hex)
2000: 0010	12
0011	34
0012	56
0013	78
0014	9A
0015	BC

Show the content of the destination operand and the state of the flag bits (O, S, Z, A, P, and C) after the execution of the following instructions. Use the initial content of the registers and memory locations for the execution of each instruction. Suppose that the CF is initially set to 1.

- | | |
|-------------------------|--------------------------|
| 1. ADC SI, CX | 6. NEG Byte PTR [DI+12h] |
| 2. INC Byte PTR [0011] | 7. MUL DX |
| 3. SBB BL, AL | 8. IMUL DL |
| 4. SUB AL, [SI] | 9. DIV Byte PTR [SI+3H] |
| 5. DEC Word PTR [DI+BX] | 10. IDIV DL |

Q.3. Give a single 8086 instruction that performs each of the following operations. Use the appropriate type pointer whenever necessary to avoid ambiguity. CF is the value of the carry bit flag.

- | | |
|---|---|
| 1. $[1FE1:1FE0] \leftarrow [1FE1:1FE0] + 2Ah$ | 4. $[BX+1:BX] \leftarrow 0 - [BX+1:BX]$ |
| 2. $AX \leftarrow AX + [SI+1:SI]$ | 5. $DX:AX \leftarrow AX * BX$ |
| 3. $AH \leftarrow AH - BH - CF$ | 6. $CX \leftarrow [SI+BX+8:SI+BX+7]$ |

Q.4. Write an 8086 assembly program that implements the following C code. Declare all variables as byte variables.

```

C version:
Main()
{
    int I, J, K, L;
    I=5;
    J=15;
    K=I-J+4;
    L=3*K;
}

```