COE 205, Term 101

Computer Organization & Assembly Programming

HW# 3

# Consider a program that has the following data segment:

*I EQU 07Fh*

*J DB `1234`*

*K EQU 250*

*L DW 0FFh*

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

## 1. MOV AL, I+1 11. MOV [SI], WORD PTR [DI-1]

## 2. MOV [SI], I 12. INC [DI+1]

## 3. MOV AX, [BL] 13. MOV DS, ES

## 4. MOV AX, J +2 14. INC L+1

## 5. MOV BX, 2\*J 15. DEC Byte PTR [SI+DI]

## 6. MOV BL, K+6 16. ADD Byte PTR [BX], 2\*I+1

## 7. MOV L, I 17. SUB AH, [BX-SI-2]

## 8. MOV DS, I 18. IMUL K

## 9. SUB AX, DS 19. SUB CX, [AX]

## 10. ADD AX, J+2[BX] 20. ADC CX, [BP]2[SI]

# Suppose that you have the following initial content of the registers and memory locations:

AX=FE14H BX=7FEDH CX=F1A4H DX=00FFH

SI =0010H DI =0020H DS=4000H

Memory Address Contents (hex)

2000: 0010 FF

0011 1A

0012 BC

0013 06

0014 FE

0015 50

## 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 CF is initially set to 1.

### ADC BX, CX 6. NEG Word PTR [BX-7FDCh]

### INC Byte PTR [DI-16] 7. MUL DL

### SBB BL, AL 8. IMUL DL

### SUB AL, 2+[SI] 9. DIV Byte PTR [DI-13]

### DEC Byte PTR 4[SI] 10. IDIV CH

# 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.

### [EFA2:EFA1]🡨 [EFA2:EFA1]-FFh 4. [BX]🡨 0 - [BX]

### BX🡨 BX + [BX+1:BX] 5. DX:AX🡨 AX\*BX

### AH🡨 AH - CL - CF 6. BX🡨 [DI+BX-6:DI+BX-7]

# Write an 8086 assembly program that implements the following C code. Declare variables I, J, K, and L as either byte or word variables using the minimum size possible.

C version:

*Main()*

*{*

*int I, J, K, L;*

*I=-4;*

*J=30;*

*K=(4\*I\*J)+(I+5\*J)+1;*

*L=K/I;*

*I=I+1;*

*J=J-I-1;*

*}*