

# ECE3210 Microprocessor Engineering

## Practice Exam2

Name: \_\_\_\_\_

1. (35 points) **Arithmetic and Logic Instructions:** For each instruction, list register and flag bit new contents after executing the instruction. Assume real mode operation. Consider each instruction set separately based on the initial state provided below.

Register initial state:

AX: F000H      BX: 3456H      CX: 0001H      DX: E390H      BP: 0008H  
SI: 0004H      DI: 0000H      DS: 3170H      SS: 6001H      ES: 5000H  
FLAGS: 0000H

- a. ADC AX, DX

AX = \_\_D390H\_\_      CF = \_\_1\_\_

- b. NEG AX

AX = \_\_1000H\_\_

- c. SAR AX, CL

AX = \_\_F800H\_\_      CF= \_\_0\_\_

- d. MOV CL, 3  
RCL AX, CL

AX = \_\_8003H\_\_      CF= \_\_1\_\_

- e. AND DL, AL

DL = \_\_00H\_\_

- f. IMUL CX

AX = F000H      DX = FFFFH      CX = 0001H

g.

```
MOV AL, 0AAH
AND AL, 55H
CMP AL, 00
```

CF = 0      ZF = 1

2. (15 points) **Machine Language**

Assume 16-bit instruction mode.

- a. Convert an 89D8 to assembly language

Solution:

MOV AX, BX;

- b. What is the machine language for instruction

MOV SI, [DI+2]

Solution:

8b7502

3. (15 points) Fill in the contents in Hexadecimal format of the specified registers and memory locations after the instruction has been executed.

Assume that stack pointer has the value of SP = 0040H before the “PUSH AX” instruction in the main procedure is executed.

```
DATA_SEG SEGMENT
    X DW 0H
DATA_SEG ENDS
```

```
MAIN PROC FAR
    MOV AX, 30H
    MOV BX, 50H
```

```

    PUSH AX          ; SP= _003E_____
    PUSH BX
    CALL ADDM        ;
    MOV X, AX        ; AX= __0080H_____ SP=____0040H_____

.EXIT
MAIN ENDP

ADDM PROC NEAR
    PUSH BP
    MOV BP, SP;      BP = __0038H_____
    MOV AX, [BP+4];  AX = __0050H_____
    ADD AX, [BP+6];  AX = __0080H_____
    POP BP
    RET 4
ADDM ENDP

```

4. (15 points) Implement the following pseudo code into assembly language using conditional JMP. Assume signed number comparison.

IF (BX > AX) OR (BX < CX)

```

{
    X=1
}
ELSE
{
    X=2
}

```

```

        CMP BX, AX
        JG L1
        CMP BX, CX
        JG L2
L1:     MOV X, 1
        JMP ENDC
L2:     MOV X, 2
ENDC:

```

5. (20 points) Write a MACRO that clears all those element of X, whose 1<sup>st</sup> MSB (most significant bit) is 1, using instruction *loop*.

```

DATA_SEG SEGMENT

    X DB 100 DUP (0FFH)

DATA_SEG ENDS

```

```
CODE_SEG SEGMENT PARA 'CODE' PUBLIC USE16

    CLEAR MACRO COUNT, VALUE

    LOCAL BEGIN, L1
        MOV CX, COUNT
        MOV SI, 0
    BEGIN:    TEST VALUE[SI], 80H
        JZ L1
        MOV VALUE[SI], 0H
    L1:      INC SI
        LOOP BEGIN
    ENDM

MAIN PROC FAR
    ...

    CLEAR 100, X; CALL MACRO
    ...

.EXIT
MAIN ENDP

CODE_SEG ENDS
END MAIN
```