



Electrical and Computer Engineering  
Computer Organization - Second Exam

Fall Semester 2014

Date: 16/12/2014

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

Time allowed: 60 minutes

ID: \_\_\_\_\_

Instructions:

- You have 60 minutes, budget your time carefully!
- Turn OFF your mobile phone.
- To make sure you receive credit, please write clearly and show your work.

Question	Maximum	Mark	Course Outcome
1	30		
2	20		
3	20		
4	30		
Total	100		

Question 1: [30; 2 each] (Multiple choice)

1. CPU checks for an interrupt signal during  
 (A) Starting of last Machine cycle  
 (B) During execution cycle  
 (C) Operand Fetch cycle  
 (D) Instruction fetch cycle
2. In 8086 the Overflow flag is set when  
 (A) The sum is more than 16 bits  
 (B) Signed numbers go out of their range after an arithmetic operation  
 (C) Carry and sign flags are set  
 (D) During subtraction
3. Which of the following is an illegal instruction  
 (A) MOV AX, [2000]                       (B) MOV CS, DX  
 (C) MOV CS, 10000                       (D) MOV DI, CX
4. Which of the following variables uses the most amount of memory:  
 (A) X DB 127,128                       (B) Y DB 20 dup(5 dup(127))  
 (C) Z DW 55 dup(0)                       (D) small DD 20 dup(20)

5. To invert 6<sup>th</sup> bit of AL register, we use:

- (A) AND AL, 40H                      (B) XOR AL, 40H  
(C) OR AL, 40H                        (D) OR AL, 0BFH

6. The effect of the following instructions on AX register is

```
push ax
add ax, 8
pop bx
sub bx, 2
push bx
pop ax
```

- (A) Leave it with its original value      (B) add 4 to it  
(C) Clear it                                    (D) double it

7. To copy the hexadecimal number A to the BH register we write

- (A) MOV 0BH, AH                        (B) MOV BH, 0AH  
(C) MOV BH, AH                         (D) MOV BH, [AH]

8. Given that AL contains the ASCII code of an uppercase letter, it can be converted to lowercase by

- (A) ADD AL, 32H                         (B) SUB AL, 32H  
(C) OR AL, 1101 1111B                 (D) ADD AL, 20H

9. The maximum segment size of an 8086 processor is

- (A) 2MB                                    (B) 64KB  
(C) 64GB                                  (D) 1MB

10. One of the following instruction is illegal:

- (A) MOV AL, [DX+1]                    (B) MOV [CX-1], SI  
(C) MOV [BX], 255                      (D) ADD AX, [1000]

11. Which register will be affected by the instruction imul BX

- (A) DS                                      (B) AX  
(C) DX                                      (D) BX

12. *model tiny* means:

- (A) One segment for code, one segment for data and one segment for stack  
(B) One segment for code, two segments for data, no stack  
(C) One segment for code, data and stack  
(D) None of the above

13. To declare (define) a matrix of 10 rows and 6 columns (10X6) of bytes we use:

- (A) Matrix DB 10, 6                    (B) Matrix DB 10 dup(6)  
(C) Matrix DW 10 dup( 6 dup(?))    (D) Matrix DB 60

14. The BP register is typically used for accessing

- (A) strings.                              (B) Extra segment  
(C) stack                                  (D) data segment

15. Consider the byte at address 13DDE within a 64K segment defined by segment address 10DE. What is its offset?

- (A) 20E1 (B) 2FFF  
 (C) 2FFE (D) 20EF

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
A	B	C	C	B	X	B	D	B	C	B	C	D	C	C

or  
C

**Question 2:** [20 marks, 4 each]

Assume (all values are in hex):

AX=0000 BX=00F3 CX=0003 DX=0000 SI=0050 DI=0000  
 CS=2000 SS=4000 DS=5000 ES=2000 SP=3000 BP=01C1  
 IP=0100

i) What is the physical address of the next instruction to be executed?

$$CS * 10h + IP = 20000 + 0100 = 20100H$$

ii) What is the physical address of the top of stack?

$$SS * 10h + SP = 40000 + 3000 = 43000H$$

iii) What is the highest possible physical address of data segment?

$$DS * 10h + FFFF = 50000 + ffff = 5FFFFH$$

vi) What is the physical address of the first operand of the following instruction?

MOV [BX+1221H], AL

$$DS * 10h + BX + 1221H = 50000 + 00F3 + 1221 = 51314H$$

v) What is the physical address of the source operand in the following instruction?

MOV DL, [BP+SI-2]

$$SS * 10h + BP + SI - 2 = 40000 + 01C1 + 0050 - 0002 = 4020FH$$

**Question 3:** [20 marks; 4pts each]

a) Show how the AL and Flags are affected by

```
MOV DL, 0CCH
MOV AL, 0BBH
ADD AL, DL
```

AL= 87H CF= 1 OF= 0 ZF= 0

b) What will be the value in AL and the following flags after executing the following instructions? Give the answer in both hexadecimal and binary.

```
mov al, 09Ah
mov bl, 073h
sub al, bl ;
```

AL= ( 27 )H, CF = 1 AF= 1 PF = 1

c) What will be the value in AX after executing the following instructions? Give the answer in hexadecimal:

```
.data
VAR DW 1122h, 3344h, 5566h, 7788h, 99AAh
.code
MOV BX, OFFSET VAR
MOV SI, 4
MOV AX, [BX+SI-1]
```

Address	Memory
02000	22H
02001	11H
02002	44H
02003	33H
02004	66H
02005	55H
02006	88H
02007	77H
02008	AAH
02009	99H

AX = ( 6633 )H

d) What will be the value in AX after executing the following instructions? Give the answer in hexadecimal:

```
.data
Table DB 1FH, 2EH, 3DH, 4CH, 5BH, 6AH, A7H, B8H, 9CH
.code
MOV SI, 5
LEA BX, Table[SI]
MOV AL, 2
xlatb
MOVSX AX, AL
```

AX = FFB8 H

e) Write assembly instruction/s for enabling interrupt flag (IF)? (Hint: IF is the 9<sup>th</sup> bit in FLAGS register).

```

pushf
pop AX
OR AX, 0100H OR STI
push AX
popf
    
```

**Question 4:** [30 marks]

a) Write a procedure to count number of Zero's in register AX? [15pts]

```

proc zeros_count near
    push cx
        mov bl, 0
        mov cx, 16
    next: rol ax, 1
           adc bl, 0 ; BL = BL + 0 + CF
           loop next
           mov bh, 16
           sub bh, bl
           ret
    pop cx
    ret
endp zeros_count
    
```

b) Write a procedure to reverse 10 bytes of an array addressed by register DX? [15pts]

```

proc Array-Reverse near
    push si
    push di
    push cx
    push ax
        xor si, si
        mov di, 9
        mov cx, 5
    next: mov al, [bx + si]
           mov [bx + di], al
           inc si
           dec di
           loop next
    pop ax
    pop cx
    pop di
    pop si
    ret
endp Array-Reverse
    
```