BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI
II SEMESTER 2013-2014
EEE/CS/INSTR F241 MICROPROCESSOR PROGRAMMING AND INTERFACING
Mid-Semester Exam (OPEN BOOK)
07-03-2014

MARKS:80 DURATION: 90 MIN

| ID No: | Name: | Sec: |
|---|---|---|
| | | |

**Note: The paper contains 12 questions. Write your answer in the space provided.**

Q1. State which of the following instructions are valid or invalid for 8086 microprocessor. Justify. (10)

a) MOV AH,CX **Invalid. Operand Size mismatch**

b) MOV 1234H,AX **Invalid. Destination cannot be an immediate number**

c) MOV CS,[SI] **Invalid. CS cannot be changed**

d) MOV [1234H],[1234H] **Invalid. MEM to MEM data transfer is not allowed**

e) MOV [SI+DI],CX **Invalid. Use of two index register is not allowed**

Q2. What will be the contents of the CX register after execution of the following code snippets? Write 'X' if the contents cannot be determined. (6)

a)
```
MOV CX, 54H
MOV BX, 3300H
MOV AX, 7666H
MOV [BX], AX
ADD CX, 777H
MOV CX, [BX]
```

CX= 7666h

b)
```
MOV AX, 3H
MOV BX, 3H
MOV CX, 00H
ADD CX, 3H
INC CX,5
```

CX= X

Q3. After the x86 is reset, from which memory location does the x86 fetches the first instruction. (1)

**CS: ffffH IP:0000H => ffff0H**

Q4. Mention the two ways the Trap flag register can be set/reset. (2)

**Using interrupt and using stack (pushf/popf)**

Q5. If the microprocessor is working at 2 MHz and memory access time is 500 ns, then how many wait states are required? Assume set-up time for address and data to be zero. Ignore buffer delays also. (2)

**Zero wait states.**

Q6. Find the status of the flag register (8086) after the execution of the following instruction. Provide the answer using 1 or 0. Assume all the flags mentioned below are zero initially. (6)

MOV AL, 0FFH
INC AL

a) Carry flag [0]  b) parity flag [1]  c) zero flag [1]

d) Auxiliary carry flag [1]  e) sign flag [0]  f) overflow flag [0]

Q7. Mention true/False for the following statements applied to memory segments (4)

a) Two segments may overlap - **True**
b) Four segments may partially overlap - **True**
c) No segments may overlap - **True**
d) Four segments may completely overlap - **True**

Q8. For the following instructions, determine the addressing mode and the Machine code in hexadecimal. Assume instructions are in 16 bit mode of operation. (6)

a) MOV EBX,12340000H

**Machine code: 66bb00003412H Addr. Mode: immediate**

b) MOV AX,4020[BX+DI]

**Machine code: 8b812040H (for 4020 in hexadecimal)and 8b81b40fH (for 4020 in decimal) Addressing mode: Base relative -plus-indexed addressing**

Q9. If an 8086 processor is working at 10 MHz - how much time does two MEMR cycles take, (6)

a) If there are no wait states

**800ns**

b) If there are 3 wait states

**1400ns**

Q10. Write the machine cycles executed for the following instructions (cpu operating in 16 bit mode). State them in the right order (8)

a) PUSH 3324H **1 MEMR, 1MEMR, 1 MEMW**

b) MOV [DI], AX **1MEMR, 1MEMW**

c) ADD EAX, 09H 1MEMR, 1MEMR, 1MEMR
d) ADC EAX, [09H] 1MEMR, 1MEMR, 1MEMR, 1MEMR, 1MEMR

Q4. With reference to the 8086 assembly program given below.

```
.MODEL  TINY
.DATA

        PORT1   EQU 7
        LOC1    DB  28H, 0FFH, 76H, 05H, 0F1H
        ARRAY   DW  43ABH,66H,175,00H
        LOC2    DB  5 DUP(00H)
        DAT1 DB     'VALID'
        ALIGN 2
        DAT2 DD 4B3H
        DAT3 DQ 43ABH
        DAT5 DW 08H
        DAT6 DW   ?

.CODE
.STARTUP

        MOV SI,OFFSET LOC1
        ADD SI,PORT1
        MOV AL,[SI+1]
        MOV BX,DAT5
        MOV CL,[SI+BX]
        MOV DX,[SI+BX+5]

.EXIT
END
```

a) Fill the 40 locations in data segment starting from 0118H. **(10)**

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| DS:0118H | 28 | Ff | 76 | 05 | F1 | Ab | 43 | 66 |
| DS:0120H | 00 | Af | 00 | 00 | 00 | 00 | 00 | 00 |
| DS:0128H | 00 | 00 | 'V' (56) | 'A' (41) | 'L' (4c) | 'I' (49) | 'D' (44) | 00 (x) |
| DS:0130H | B3 | 04 | 00 | 00 | Ab | 43 | 00 | 00 |
| DS:0138H | 00 | 00 | 00 | 00 | 08 | 00 | 00/x | 00/x |

b) Determine the final values of AL,BX,CL,DX registers. **(4)**

| Register | AL | BX | CL | DX |
|---|---|---|---|---|
| Value | 00 | 0008 | 00 | IL(494c) |

Q2. Write a program that computes the square of a 16-bit number stored in memory at offset *dat1*. The result has to be stored again in memory at an offset *res1*. Write a near procedure to compute the square of the number and call this procedure from the main program. Use AX register to pass the parameter to the procedure. **(15)**

```
.model tiny
.data

DAT1   DW  0156H
RES1   DW  2 DUP(0)

ST1  DW 10 DUP(0)
ST2 DW 0
.code
.startup
        LEA SP, ST2
        MOV AX, DAT1

        CALL MYSQUARE

        LEA SI, RES1
        MOV WORD PTR [SI], AX
        INC SI
        INC SI
        MOV WORD PTR [SI], DX

.EXIT

MYSQUARE   PROC NEAR
        PUSHF
        MUL AX
        POPF
        RET
MYSQUARE   ENDP

END
```