BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI
II SEMESTER 2013-2014
EEE/CS/INSTR F241 MICROPROCESSOR PROGRAMMING AND INTERFACING
QUIZ #2 (OPEN BOOK)

MARKS:10 **12-02-2014** **DURATION: 30 MIN**

ID:	NAME: **SOLUTIONS**	SEC:

Note: Each question carries one mark.

Q1. For string instructions, the SI offset address accesses data, by default, in the data segment, which may be changed with a segment override prefix and the DI offset address accesses data in extra segment and this cannot be changed with segment override prefix. (**True/False ?**)

TRUE

Q2. Under what two conditions will the REPNE CMPSB instruction pass control to the following instructions?

CX = 0
Destination byte = Source byte

Q3. Which logical instruction should be used to force bits D0 and D1 of register DH (eight bit reg,D7 to D0)zero without changing any of the other bits?

AND 0FCH

Q4. For an 8086 CPU, which two instructions will have the same effect of dividing register BX by 8? (can't use DIV/IDIV instruction)?

```
MOV CL,03  (IT CAN NOT BE CX)
SHR BX,CL
```

Q5. Using no more than three instructions, write a program (just the three instructions) to multiply the contents of AX by 7. (can't use MUL/IMUL instruction).

```
MOV BX,AX
SHL AX,3
SUB AX,BX
```

Q6. Contrast the operation of a JMP [DI] with a JMP FAR PTR [DI].

JMP [DI]:INTRA-SEGMENT JUMP. ONLY IP REPLACED WITH WORD IN CS POINTED BY DI
JMP FAR PTR [DI]: INTER-SEGMENT JUMP. BOTH IP AND CS REPLACED WITH DOUBLE WORD POINTED BY DI

Q7. Write the contents of the register/memory location as depicted in the following section of ALP:

```
.data
sum dw 0cfh
.code
.startup
clc
mov ax,1234h
add ax,sum            --AX = 1303H
mov ax,1234h
adc ax,sum            --AX = 1303H
inc word ptr sum      - sum = 00D0H
inc byte ptr sum      - sum = 00D1H
```

Q8. For the following code, write the contents of the data segment in the table. Write 'X' if the contents are not known. The data segment starts at physical address 02000H.

```
.model tiny
.data
count1 equ 10
count2 equ 20
org 100h
count3 equ 30
.code
.startup
mov ax,count1
.exit
end
```

02000H→	X	X	X	X	X

Q9. The conditional jump can only be to a location within the range of -128 to +127 bytes from the instruction after the conditional jump instruction. If the program requires a conditional jump beyond this range, suggest a way to overcome this limitation.

```
CMP AX,03H
JE X1
-----
-----
MOV BX,CX
------
X1: MOV BH,04H
```

Target is to jump to X1 if AX=03H. If Conditional jump (JE) to X1 goes beyond the -127 to +128 bytes limit?

```
CMP AX,03H
JNE X2
JMP X1
------
X2:MOV BX,CX
------
X1: MOV BH,04H
```

Programmer ensures Conditional jump (JNE) to X2 falls within limit and if condition not met, then uses unconditional JMP to X1 with no limitation

Q10. For an 8086 cpu, if the MUL instruction is used to multiply a byte by a word, can the CBW instruction be used to convert the byte to a word before doing the multiplication? Justify.

CBW can not be used. The MUL instruction multiplies unsigned data. If AL contained a data which had Most Significant bit set to 1, then CBW will fill AH with all 1's instead of 0's which will give incorrect result.