Notre Dame University

ECCE Department

Exam # 2

Spring Semester (2004-2005)

Microprocessor Design Systems

EEN324

Name:_____A. KAssem_____

Student ID:____Prof._____

Question1		10
Question2		10
Question3		10
Question4		20
Total		/50

Question 1. (**10 points**)

Answer True or False for each of the following statements (1 point each)

1. The following piece of code copies the contents of C into D: **PUSH C** **POP D**	T	(F)
2. The following instructions perform the same operation **XRA A** **XRI 00H**	(T)	F
3. The instruction '**LDA BC**' copies the contents of memory location pointed by BC register pair INTO register A.	T	(F)
4. The **MVI C, A** instruction copy the content of register A into the register C.	T	(F)
5. **INR M** instruction increments by 1 the content of memory pointed by the register pair HL.	(T)	F
6. **LXI H, 3210H** and **MVI 32H, H** **MVI 10H, L** Perform the same operation.	T	(F)
7. **INX D** instruction increments the register **D** by 1.	T	(F)
8. Programmers can set the **S**, **Z**, **AC** and **CY** flags directly.	(T)	F
9. The SP register is decremented only once during the execution of a **CALL** instruction.	T	(F)
10. Both **SLA** and **SRA** instructions modify the contents of the **AC** flag.	T	(F)
11. Can we perform the following **MOV L, M** **DCR M**	(T)	F

Question 2. (10 points)

2.1. Execute the following program and show the flags state after each instruction.(**5 Points**)

			Z	S	CY
HL = FE00	**LXI**	**H,FE00**	X	X	X
A = 7F	**MOV**	**A, M**	X	X	X
A = 7F + 11 = 90H	**ADI**	**11H**	0	1	0
HL = FE01	**INX**	**H**	0	1	0
A = 90 . 0F = 00H	**ANI**	**0FH**	1	0	0
B = FF	**MOV**	**B, M**	1	0	0
C = 88	**MVI**	**C, 88H**	1	0	0
A = 00 + 88 = 88	**ADD**	**C**	0	1	0
Yes it is negative	**JM**	**STP**	0	1	0
Go to HLT	**LDAX**	**B**	X	X	X
	RAR		X	X	X
STP	**HLT**				

Given the memory contents shown below

000F	00H
0F00	01H
FE00	7FH
FE01	FFH
FF01	02H
FF88	81H
88FF	FFH
88FE	7FH

2.2. The starting address of the following program is 0000H. (**5 Points**)

```
0000H    Start:    MVI   A, Byte1    ; 3 bytes
0003H              ORA   A           ; 1 byte
0004H              JP    NEXT        ; 3 bytes
0007H              XRA   A           ; 1 byte
0008H    NEXT:     OUT   F2H         ;
                   HLT
```

a) Specify the address of the label **NEXT:** and explain the type of numbers that can be displayed at the port.

$0008, It displays only the numbers ≥ 00H

b) If **Byte1** = 92H, what is the output?

00H

c) If **Byte1** = 09H, what is the output?

09H

Question 3. (10 points)

Using a 8085 microprocessor with a frequency of 2MHz, calculate the **MAXIMUM** time, that, it takes to execute the following program:
The time delay for DELAY subroutine is 100.46 ms.

		# of T-St.	
To = 85 T-States	**LXI H, 9001H**	**10**	
	LDA 9000H	**13**	
	OUT 40H	10	
	CALL DELAY	18	+ 100.46ms
	MVI C, 0FH	7	
	ANA C	4	
	MOV C, A	4	
	MOV A, B	4	
	MVI B, F0H	7	
	ANA B	4	
	MOV B, A	4	
Loop	MOV A, B	4	
	RRC	4	
	MOV B, A	4	
	MOV A, C	4	
	RLC	4	
Ti = 76 T-States + Subroutine Delay	MOV C, A	4	
	ORA B	4	
	OUT 40H	10	
	CALL DELAY	18	+ 100.46ms
	DCR M	10	
	JNZ Loop	10-7	
	HLT		

Given the memory contents shown below

9000	FFH
9001	81H
F000	FFH
F001	02H
000F	81H
F000	FFH
0040	00H
4000	FFH

Counter is equal to the memory M content (81H = 129_{10})

T_{out} = 85 T-States + 100.46 ms * 2 MHz = 201005

T_{in} = (129 * 76) – 3 + 129 * 100.46 ms * 2 MHz. = 25928481

$T_{Total} = T_{out} + T_{in}$ = 26129486

Delay = 26129486 / 2 MHz = 13.064743 s

Question 4. (**20 points**)

Write a program to load a binary number (<64H) from input port address 01H, and start counting the loaded number until 99 in **BCD**; finally display the result at the output port address FFH, with a delay between each count.

Assume that you have a subroutine called DELAY.

Example: if the input number is $(\mathbf{0000\ 1010})_2$ the **1st** output is $\mathbf{10}_{10}$, the **2nd** is $\mathbf{11}_{10}$, the **3rd** is $\mathbf{12}_{10}$ until 99_{10}.

```
START   IN    01H              ; Input Port
        MVI   D, 64H           ; # to be compared if ≥ 100
        MOV   C,  A            ; save input in C
        CALL  PWR10            ; Start convert to BCD
        HLT
PWR10:
        LXI   H,  OUTBUF
        MVI   B,   0AH
        CALL  BINBCD
        RET
BINBCD:
        MVI   M,  FFH
NXTBUF:
        INR   M                ; [M] = BCD2
        SUB   B                ;
        JNC   NXTBUF           ;
        ADD   B                ; A = BCD1
        RLC                    ; Rotates A 4 times
        RLC                    ;
        RLC                    ;
        RLC                    ;
        ORA   M                ; Get 2 digits BCD2 & BCD1
        OUT    A               ; Output the BCD2 & BCD1
        INR   C                ; Increments the loaded input by 1
        MOV   A, C
        CMP   D                ; compares it, if =
        JNZ   PWR10            ; in not converts the new incremented #
        RET                    ; if yes go out
```