Digital Logic & Processor

Final Exam

Monsoon -2010

Roll number:

Name:

Seat:

Q_1	Q_2	Q_3	Q_4	Q_5	Q_6	Q_7	Q_8	Total

Time: 3 hrs

Maximum Marks: 120

Answer all questions. Marks are indicated against each question. The answers should be given in the space provided. State your assumptions clearly if there is ambiguity with question. ***All the best!***

1)

a) Give the design of a **JK** Master-Slave flip-flop with AND and NOR gates. Include a provision for asynchronous clear. **[5 marks]**

b) Consider a **JK'** flip-flop (a **JK** flip-flop with an inverter between the external input **K'** and the internal input **K**). Obtain its characteristic table, characteristic equation and the excitation table. **[5marks]**

2)

a) Using K-maps, design a circuit to divide a 2-bit number $\mathbf{N_1N_0}$ by another 2-bit number $\mathbf{D_1D_0}$ **(≠ 0)** to generate a 2-bit quotient $\mathbf{Q_1Q_0}$ and a 2-bit remainder $\mathbf{R_1R_0}$. **[10 marks]**

(i) Using only NAND gates

(ii) Using only NOR gates

b) Draw the circuit of a 4-bit register with parallel load and shift left. The mode bit **M** is 0 for load and 1 for shift. Parallel load inputs are $\mathbf{I_1}$ to $\mathbf{I_4}$. The outputs change on the rising edge of clock **CK**.

[5 marks]

3)

a) A stream of 1s and 0s appear at an input line. We need to design a recognizer for the sequence **0101** or **1010** with overlap. (That is the output should be 1 whenever the preceding 4 bits are 0101 or 1010). Show the state diagram for this sequence recognizer. **[5 marks]**

b) Design a 3-bit synchronous count-by-3 circuit that follows the sequence 0, 3, 6, 1, 4, 7, 2, 5, 0, 3, ... Let the outputs from left-to-right be x, y, and z. Use T flip-flops for the circuit. Give the expressions for T_x, T_y and T_z. in terms of x,y,z. Draw the circuit completely using any kind of gates. **[5 marks]**

4)

Perform the following arithmetic operations. All numbers are stored using 8-bit registers with 1's complements to represent negative numbers. Indicate clearly the carry on the row above.

[5x 1 marks]

a) -67 – 37

b) 25 – 97

c) -97 – 37

d) -126 − 2

e) Indicate in which of the above cases an overflow error has occurred.

5)

Consider the processor designed using the single bus architecture discussed in the class. Write down the microinstructions corresponding to only the execute cycle for the following instructions. Assume the instructions to be one byte long if no immediate operand is present. State the number of micro-cycles for executing each instruction. **[4 x 5 = 20 marks]**

a) **addr <r1>, <r2>, <r3>** : Add contents of registers **<r2>** and **<r3>** and store in **<r1>**

b) **loadr <r1>, <r2>** : move the contents of memory location in **[[r1]]** to memory location **[[r2]]**.

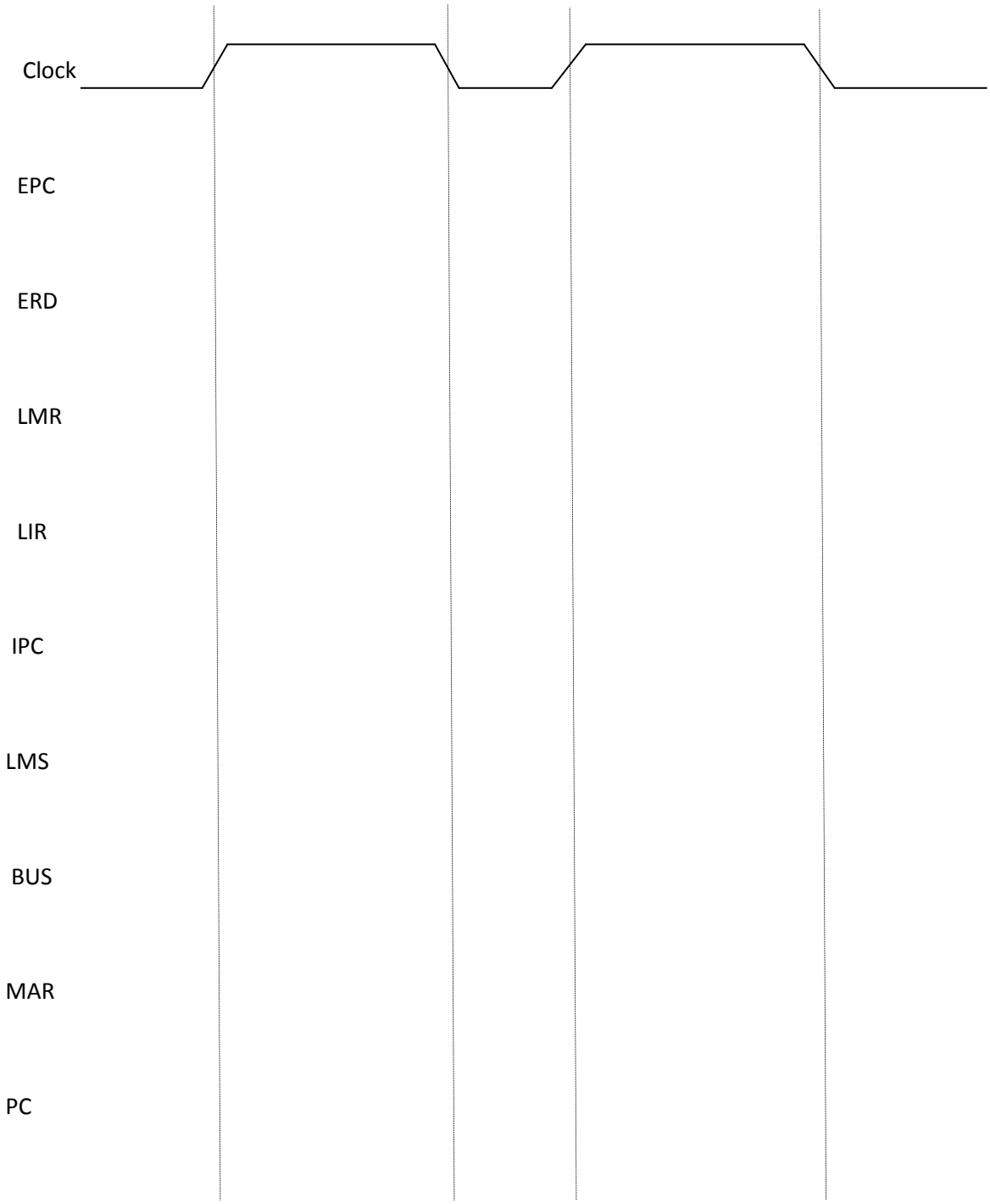
c) **loadx <r1>, <r2>:** Indexed load. Move the contents of memory location **[[r1]+[r2]]** to **AR**.

d) **pushi, xx** : Push Immediate : Decrement **SP** by 1 and put [**xx]** in the memory location specified by **SP**

Op-Code	Micro-Instructions	Selects
addr		
loadr		
loadx		
pushi		

6)

a) Write the micro-instructions corresponding to the 2-step fetch – cycle for the processor discussed in the class. Complete the following timing diagram for the fetch cycle. Assume **PC** contains 100 (decimal) and memory location at 100 contains 05 (decimal). **[10 marks]**



b) Consider the following ***program fragment*** having the op-codes discussed in the class. Note that the 2-byte instructions have been separated by commas. **[5 marks]**

```
        movi r1 , COUNT
LOOP:   movs r1
        subi,01
        movd r1
        jmpdnz, LOOP
```

Write down the micro-instructions and the number of clock cycles corresponding to each of the op-codes.

Op-code	Micro-Instructions	Selects
movi		
movd		
movs		
subi		
jmpd		

c) Assume the Clock frequency is 400MHz (i.e. Clock period = 2.5ns), calculate the time taken by each **LOOP** in the above program and hence determine the value of **COUNT** required to give an overall delay of 10µs. **[5 marks]**

7) Consider the following ***program*.** Please note that all 2-byte instructions are separated by commas. Also note that **n, N,** and **g** are positive integers.

a) Write down the micro-instructions corresponding to **cd** and **ret** and mention the number of clock cycles for each. **[3 + 2 marks]**

b) Analyze the program carefully and determine its output. **[5 marks]**

```
        movi r0, n
        movi r1, A0
        movi r2, N
LOOP1:  cdu ,  NEXT1
        cdu,  NEXT2
        movs r2
        subi, 01
        movd r2
        jmpdnz, LOOP1
        stop
NEXT1:  movs r1
        store r0
        adi, 01
        movd r1
        retu
NEXT2:  movi r3, g
        movi r4,00
LOOP2:  movs r4
        add r0
        movd r4
        movs r3
        subi, 01
        movd r3
        jmpdnz, LOOP2
        movs  r4
        movd r0
        retu
```

Op-Code	Micro-Instructions	Selects
cd<FL>		
ret<FL>		

c) Trace the following ***program fragment*** for 11 steps of execution. **SP** is initialized to 255 and **PC** to 0 at start. The program is loaded starting at address 0. Show the memory location in which each instruction is stored. Indicate the instruction executed in sequence in the table and show the new values of **all** internal registers and memory which are changing as each instruction is executed. **[10 marks]**

```
movi r1, 22
movs r1
adi, 42
push r1
cdu, 11
pop r2
movs r2
stop
add r1
movd r3
retu
```

Step No	Instruction	Changed Values
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		
11		

d) Write an assembly language program using the op-codes discussed in the class that reads a number '**n'** stored in memory location **M1** and checks if its divisible by '**r'** (**r >= 3)**

[10 marks]

8) Write down an assembly language program using the op-codes discussed in the class that generates a straight line ***y = mx*** with a variable slope '***m***' starting from origin. The input to your program should be the slope '***m***' stored at memory location **M1**. The program should return the co-ordinates of the line **(x_1, y_1), (x_2, y_2) ….upto N** co-ordinates and store them starting from memory location **M2** onwards. **(**write 2-byte instructions on the same line using commas to separate the op-code and data) **[10 marks]**