

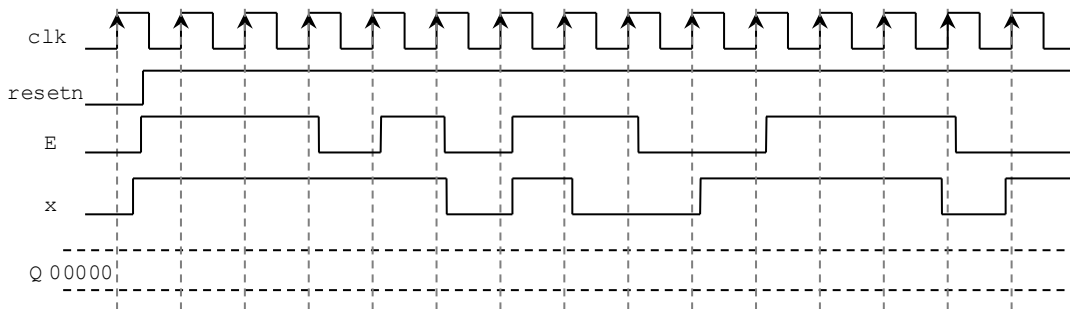
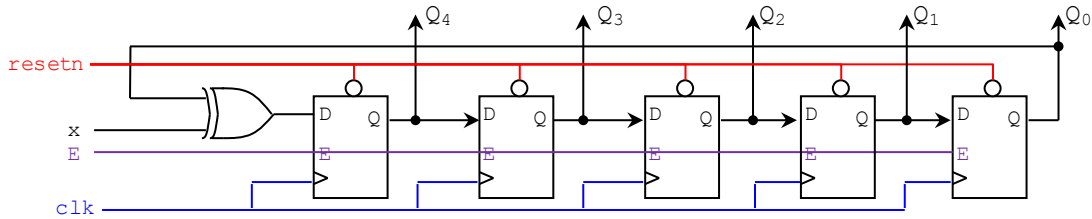
Final Exam (Online Section)

(Due Date: December 11th by 10:00 am)

Clarity is very important! Show your procedure!

PROBLEM 1 (12 PTS)

- Complete the timing diagram of the following circuit. $Q = Q_4Q_3Q_2Q_1Q_0$

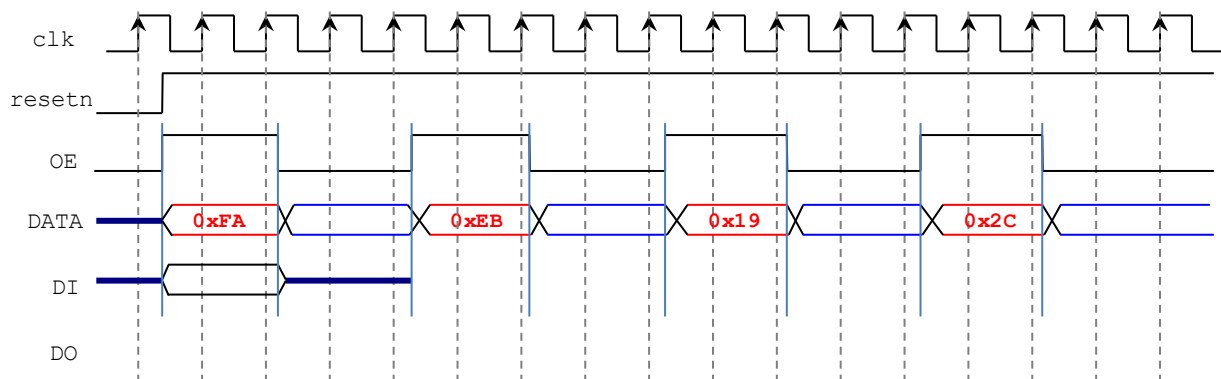
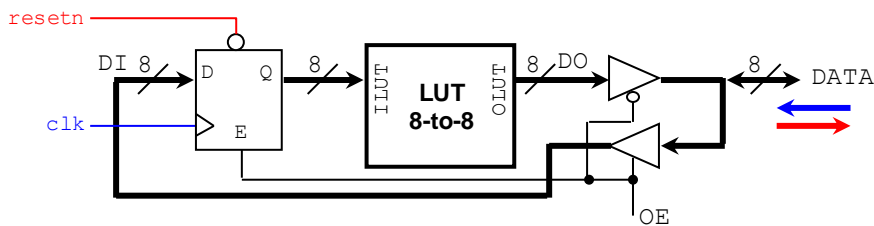


PROBLEM 2 (10 PTS)

- We want to design a counter *modulo-9* with enable using a State Machine. The counter must assert an output $z = '1'$ when the maximum count is reached.
- Provide the State Diagram (any representation) and the Excitation table $||Inputs|Present State||Next State|Outputs||$. Is this a Moore or a Mealy machine?

PROBLEM 3 (15 PTS)

- Given the following system, complete the Timing Diagram.
- The LUT 8-to-8 implements the following function: $OLUT = [sqrt(ILUT)]$



PROBLEM 4 (18 PTS)

- Provide the State Diagram (any representation) of the FSM whose VHDL description is shown below.
- Complete the Timing Diagram.

```
library ieee;
use ieee.std_logic_1164.all;

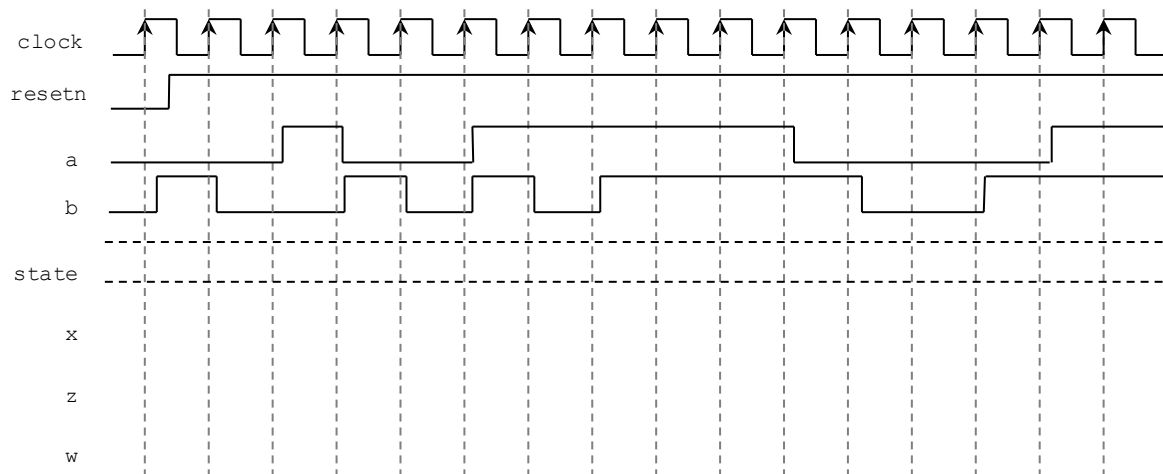
entity circ is
  port ( clk, resetn: in std_logic;
        a, b: in std_logic;
        x,w,z: out std_logic);
end circ;
```

```
architecture behavioral of circ is
  type state is (S1, S2, S3);
  signal y: state;
begin
  Transitions: process (resetn, clk, a, b)
  begin
    if resetn = '0' then y <= S1;
    elsif (clk'event and clk = '1') then
      case y is
        when S1 =>
          if a = '1' then
            if b = '1' then y <= S2;
            else y <= S1; end if;
          else
            y <= S1;
          end if;

          when S2 =>
            if a = '1' then y <= S3;
            else y <= S1; end if;

          when S3 =>
            if b = '1' then y <= S3;
            else y <= S1; end if;
        end case;
      end if;
    end process;

  Outputs: process (y,a,b)
  begin
    x <= '0'; w <= '0'; z <= '0';
    case y is
      when S1 =>
        if a = '1' and b = '1' then
          z <= '1'; end if;
      when S2 => x <= '1';
      when S3 =>
        if b = '1' then w <= '1'; end if;
    end case;
  end process;
end behavioral;
```



PROBLEM 5 (10 PTS)

- Sequence detector (with overlap): Draw the state diagram (in ASM form) of a circuit (with an input 'x') that detects the following sequence: 1011011. The detector must assert and output z='1' when the sequence is detected.

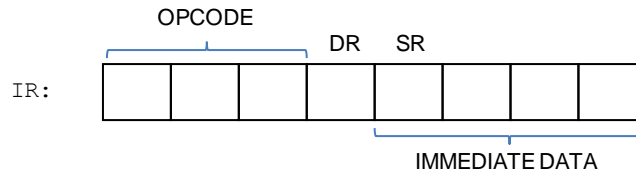
PROBLEM 6 (15 PTS)

Basic Processor:

Available Registers: R0 (register 0, 4 bits), R1 (register 1, 4 bits), PC (program counter, 4 bits),
OUT (output register, 4 bits)
IR (instruction register, 8 bits)

Instruction Memory: Stores up to 16 8-bit instructions.

Instruction Set: Instructions are specified on the Instruction Register (IR):



DR=0 \Rightarrow R0 is the destination register, DR=1 \Rightarrow R1 is the destination register.

SR=0 \Rightarrow R0 is the source register, SR=1 \Rightarrow R1 is the source register.

OPCODE (IR[7..5])	Instruction	Operation
000	MOV DR, SR	DR \leftarrow SR
001	LOADI DR, DATA	DR \leftarrow DATA, DATA = IR[3..0]
010	ADD DR, SR	DR \leftarrow DR + SR
011	ADDI DR, DATA	DR \leftarrow DR + DATA, DATA = IR[3..0]
100	SR0 DR, SR	DR \leftarrow 0&SR[3..1]
101	IN DR	DR \leftarrow IN
110	OUT DR	OUT \leftarrow DR
111	JNZ DR, ADDRESS	PC \leftarrow PC + 1 if DR=0 PC \leftarrow IR[3..0] if DR \neq 0 * ADDRESS = IR[3..0]

- Write an assembly program for a counter from 2 to 13: 2, 3, ..., 13, 2, 3, ... The count must be shown on the output register (OUT). Use labels to specify any address where your program jumps. Note that you can have only up to 16 instructions.
- Provide the contents of the Instruction Memory.

address	INSTRUCTION MEMORY
0000	
0001	
0010	
0011	
0100	
0101	
0110	
0111	
1000	
1001	
1010	
1011	
1100	
1101	
1110	
1111	

PROBLEM 7 (20 PTS)

- Complete the timing diagram of the following digital circuit that includes an FSM (in ASM form) and a datapath circuit.

