

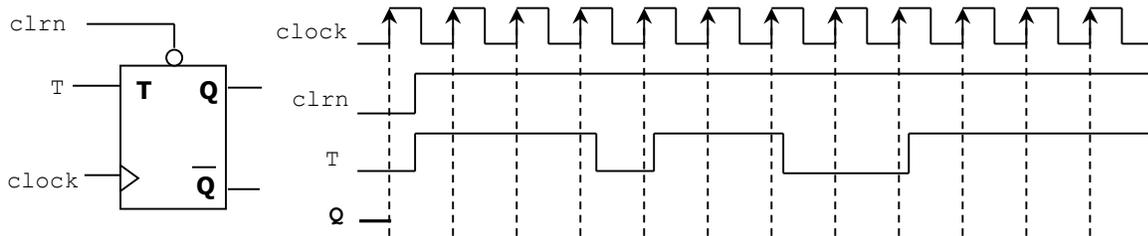
Homework 3

(Due date: March 12th @ 5:30 pm)

Presentation and clarity are very important! Show your procedure!

PROBLEM 1 (25 PTS)

a) Complete the timing diagram of the circuit shown below. (5 pts)



b) Complete the timing diagram of the circuit whose VHDL description is shown below: (5 pts)

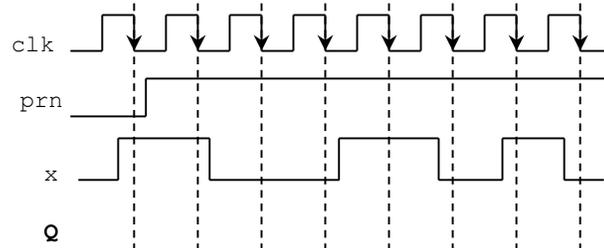
```
library ieee;
use ieee.std_logic_1164.all;

entity circ is
    port ( prn, x, clk: in std_logic;
          q: out std_logic);
end circ;

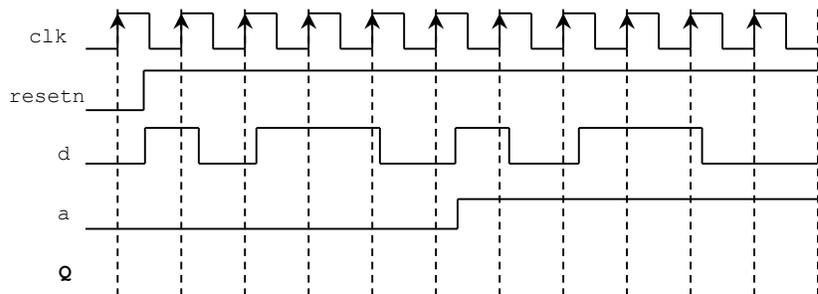
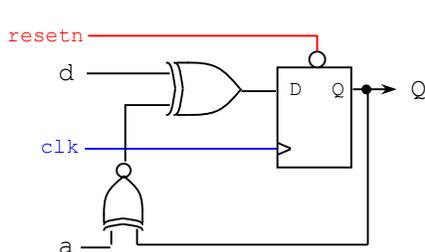
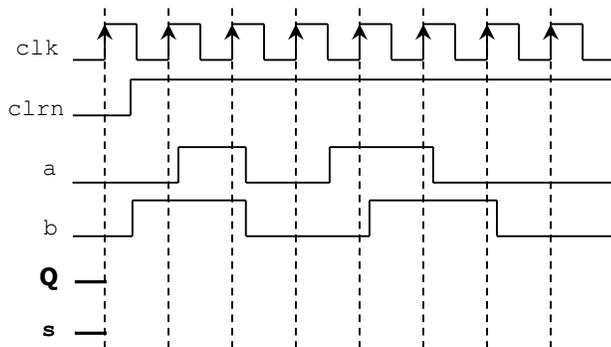
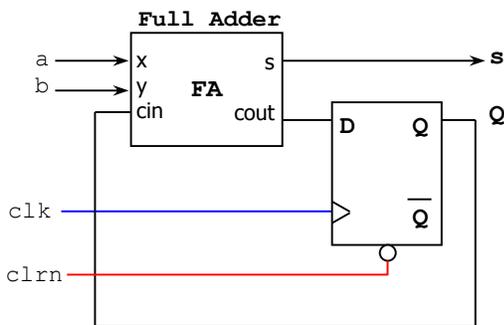
architecture a of circ is
    signal qt: std_logic;

begin
    process (prn, clk, x)
    begin
        if prn = '0' then
            qt <= '1';
        
```

```
        elsif (clk'event and clk = '0') then
            if x = '0' then
                qt <= not(qt);
            end if;
        end if;
    end process;
    q <= qt;
end a;
```

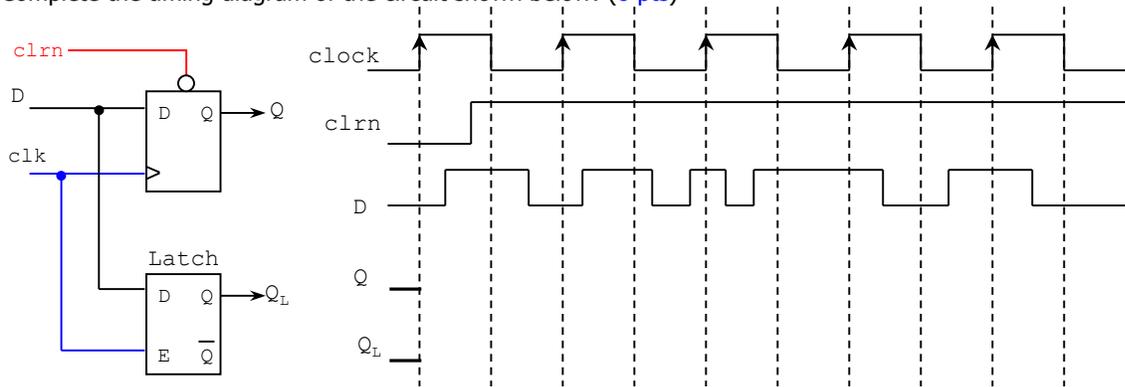


c) Complete the timing diagram of the circuits shown below: (15 pts)



PROBLEM 2 (15 PTS)

- Complete the timing diagram of the circuit shown below: (8 pts)



- Complete the VHDL description of the synchronous sequential circuit whose truth table is shown below: (7 pts)

```

library ieee;
use ieee.std_logic_1164.all;

entity my_ff is
  port ( a, b, c: in std_logic;
         clrn, clk: in std_logic;
         q: out std_logic);
end my_ff;

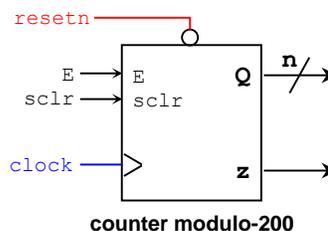
architecture a of my_ff is

begin
  -- ???
end a;
    
```

clrn	clk	A	B	Q_{t+1}
1		0	0	C
1		0	1	Q_t
1		1	0	1
1		1	1	Q_t
0	X	X	X	0

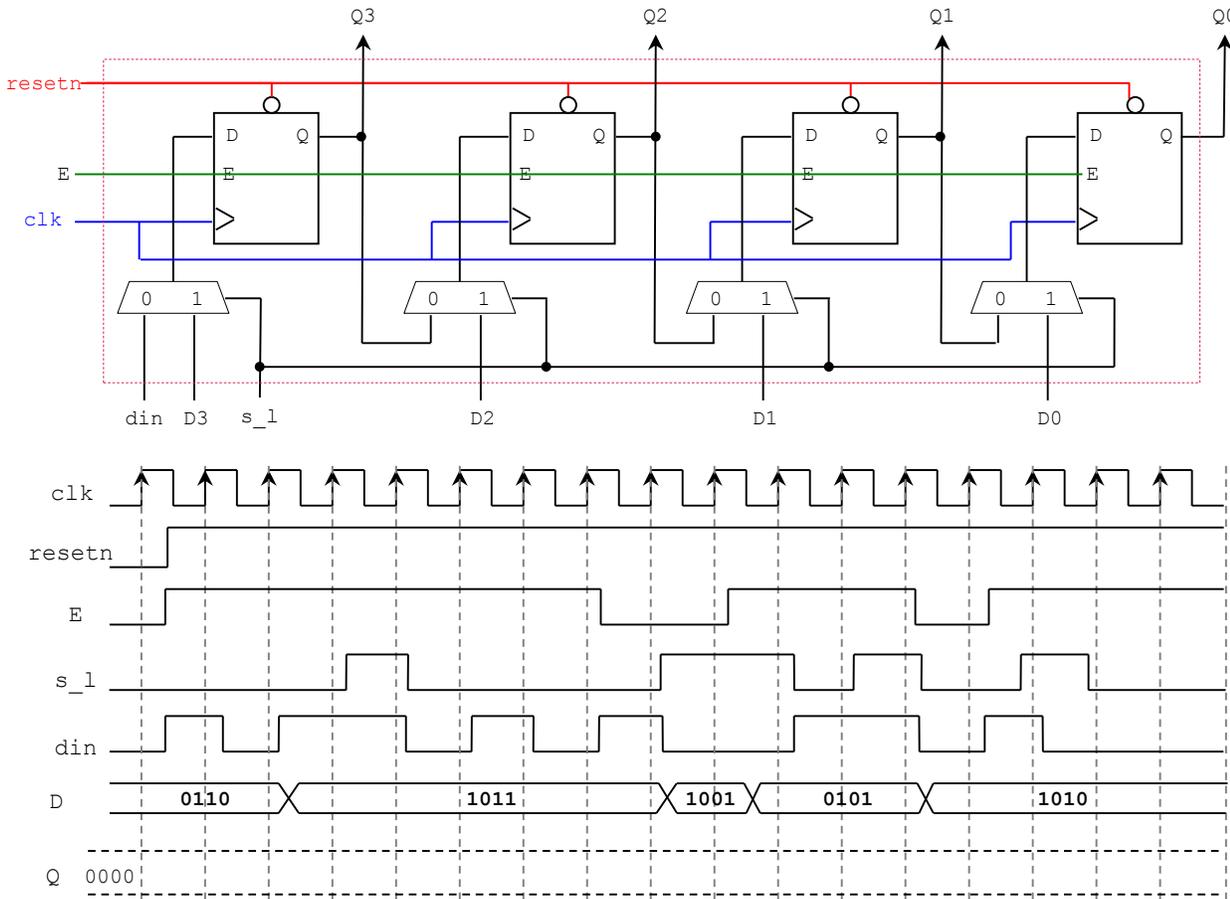
PROBLEM 3 (15 PTS)

- Design a modulo-200 counter with enable, synchronous clear, and synchronous load.
- Asynchronous input: `resetn`
- Synchronous inputs:
 - ✓ `E`: This is the enable input. It increases the count every time it is asserted ($E = 1$).
 - ✓ `sclr`: It clears the count (it requires $E = 1$)
- Outputs:
 - ✓ `Q`: This is the count.
 - ✓ `z`: It is asserted only when the maximum count is reached.
- You need to determine the minimum number of bits n that it is required for the count.
- You can use adder units, registers, logic gates, and MUXes.



PROBLEM 4 (35 PTS)

- The following circuit is a parallel/serial load shift register with enable input. Shifting operation: $s_1=0$. Parallel load: $s_1=1$.
 - Write a structural VHDL code. You MUST create a file for: i) flip flop, ii) MUX 2-to-1, and iii) top file (where you will interconnect the flip flops and MUXes). Provide a printout. (15 pts)
 - Write a VHDL testbench according to the timing diagram shown below. Complete the timing diagram by simulating your circuit (Timing Simulation). The clock frequency must be 100 MHz with 50% duty cycle. Provide a printout. (20 pts)



PROBLEM 5 (10 PTS)

- Attach a printout of your Initial Project Report (no more than a page). This report should contain the project title, the project description, and the current status of the project. Use the provided template (Final Project - Report Template.docx).