EEL 3701C – Digital Logic & Computer System

LAB 3

(53 Pts, Due 10/30/2022, 11:59 PM)

Goal:

By the end of this lab, you should be able to: -

- Design and implement an SR latch.
- Shift Registers
- Asynchronous Counters

Problem 1: Design and Implementation of an alarm circuit using SR latch on FPGA
An SR latch is an asynchronous circuit that works independently of a clock/control signal and relies only on the state of the S and R inputs. An SR latch can be created using two NOR gates that have a cross-feedback loop. The value (current state) stored in the latch is readable from the output (Q).

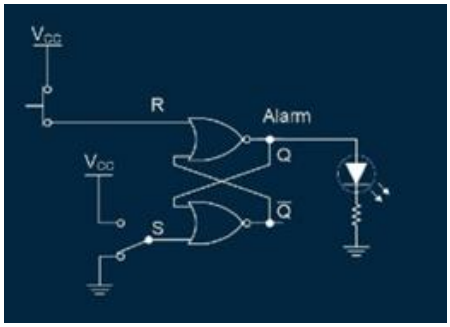

Figure 1 SR Latch

In this problem, we want to develop an alarm system for your home. The system consists of two switches, one of which is connected to the door and the other connected to the control panel. After arming the alarm, if the door is opened, the buzzer will turn on. Once the alarm is off, closing the door will not stop the buzz. Only by pressing the reset on the panel can the alarm be stopped.

Pre-Lab Homework (7 pts)

1. Explain why the SR-latch is enough to make your alarm work (Consider connecting the R and S with two switches and connect the output Q with the Buzzer/LED on the extension board). **(3 pts)**
2. Implement the design in **VHDL CODE: Problem 1** by instantiating a latch and building the logic around it. Compile and make sure that the written code is error free. **(4 pts)**

In-Lab Implementation (5 pts)

1. Synthesize your circuit, download and test your design on the Max10 for different inputs and outputs. Make sure that the logic of the alarm is working.

VHDL CODE: Problem 1

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;
use ieee.std_logic_unsigned.all;
        entity SR_latch is
                Port ( S : in    STD_LOGIC;
                       R : in    STD_LOGIC;
                       Q : inout STD_LOGIC);
        end SR_latch ;

        architecture Behavioral of SR_latch is
        signal notQ : STD_LOGIC;
        begin
        Q    <= R nor notQ;
        notQ <= S nor Q;

end Behavioral;
```

Figure 2 VHDL code for problem 1

Problem 2 – Shift Registers

Below is the code of a 4-bit shift right register that uses 4 D-Flip-Flops.

```
Library IEEE;
USE IEEE.Std_logic_1164.all;

entity SLR_DFF is
  port(
    Clk :in std_logic;
    reset: in std_logic;
    D :in std_logic;
    Q : out std_logic_vector(3 downto 0)
  );
end SLR_DFF;

architecture Behavioral of SLR_DFF is
Component DFF is
  port(
    Clk :in std_logic;
    reset: in std_logic;
    D :in  std_logic;
    Q : out std_logic
```

```
   );
end component;

Signal q3tod2, q2od1, q1tod0: std_logic;

begin
R3: DFF Port map(Clk => Clk, reset => reset, D => D, Q => q3tod2);

R2: DFF Port map(Clk => Clk, reset => reset, D => q3tod2, Q => q2od1);

R1: DFF Port map(Clk => Clk, reset => reset, D => q2od1, Q => q1tod0);

R0: DFF Port map(Clk => Clk, reset => reset, D => q1tod0, Q => Q(0));

Q(3) <= q3tod2;

Q(2) <= q2od1;

Q(1) <= q1tod0;

end Behavioral;
```

Pre-Lab Homework (8 pts)

1. Understand and explain the behavior of the circuit. **(4 pts)**
2. Simulate the circuit and annotate the results. **(4 pts)**

In-Lab Implementation (17 pts)

1. Map the inputs of your circuit to 4 switches and the output to 4 LEDs on your board.

2. Synthesize your circuit and demonstrate the functionality of you D-FF to your PI. **(5 pts)**

3. Using the main clock, you will not see any changes because they are happening to fast. Use the clock divider circuit of lab 2 to slow down your design to a reasonable frequency for your eyes to see the changes. **(5 pts)**
4. **Bonus**: Modify the circuit so that it can do a shift left and a shift right based on the value of an input. That control input will be mapped to another switch. Simulate and implement your design in FPGA. **(7 Pts).**

Problem 3 – Asynchronous Counter

Pre-Lab Homework (10 pts)

Design and simulate the mod-16 asynchronous counter on slide 25 of chapter 03 “Latches, Flip-Flops, Memory**”. (10 pts)**

In-Lab Implementation (6 pts)

Map the input to switches and outputs to LEDs, synthesize and demonstrate your circuit on your board.

Deliverables: Prepare and submit your report in canvas, using the template provided on the webpage. Include supporting documents (pictures, video, or compressed sources after cleaning up your project) in your report.

N.B All Pre-Lab HomeWorks must be completed before your lab session. You will not be admitted to the lab without completing this part.