# **Solutions - Homework 3**

(Due date: November 2<sup>nd</sup> @ 11:59 pm)

Presentation and clarity are very important! Show your procedure!

#### PROBLEM 1 (12 PTS)

• Complete the timing diagram of the circuit shown below. (7 pts)



• Complete the timing diagram of the circuit shown below: (5 pts)



#### $Q_{t+1} \leftarrow T \oplus Q_t$

## PROBLEM 2 (16 PTS)

• Complete the timing diagram of the circuit shown below. Get the excitation equation for Q. (6 pts)



1

• Complete the timing diagram of the circuit shown below: (10 pts)





# ELECTRICAL AND COMPUTER ENGINEERING DEPARTMENT, OAKLAND UNIVERSITY ECE-2700: Digital Logic Design

#### PROBLEM 3 (16 PTS)

- With a D flip flop and logic gates, sketch the circuit whose excitation equation is given by:
  - $\checkmark \quad Q(t+1) \leftarrow \overline{y}Q(t) + x\overline{Q(t)} \text{ (4 pts)}$



• Given the following circuit, get the excitation equations for each flip flop output  $Q = Q_3 Q_2 Q_1 Q_0$  (6 pts)



• Complete the timing diagram of the circuit whose VHDL description is shown below. Also, get the excitation equation for q.



# PROBLEM 4 (15 PTS)

 Given the following circuit, complete the timing diagram (signals *DO*, *Q* and *DATA*). The LUT 6-to-6 implements the following function: *OLUT* = [*ILUT*<sup>0.85</sup>], where *ILUT* is an unsigned number. For example: *ILUT* = 35 (100011<sub>2</sub>) → *OLUT* = [35<sup>0.85</sup>] = 21 (010101<sub>2</sub>)



#### PROBLEM 5 (10 PTS)

Complete the timing diagram of the following 4-bit parallel access shift register with enable input. When E=1: If s = 0 (shifting operation). If s = 1 (parallel load). Note that  $Q = Q_3 Q_2 Q_1 Q_0$ .  $D = D_3 D_2 D_1 D_0$ 03 02 01 Q0 resetn ሰ D Q D Q D Q D О E E clk 0 0 0 0 1 din D3 s l D2 D1 D0 clk resetn Е s l din D 1101 1110 0101 1011 0111 Q 0000¦0000¦1000¦0100|1010|1010|1110|0111|0111|0101|0010|1001|1100|0110|0011|0011|0001|1000|

## PROBLEM 6 (23 PTS)

- For the following circuit, we have  $Q = Q_3 Q_2 Q_1 Q_0$ .  $P = P_3 P_2 P_1 P_0$ 
  - ✓ Write structural VHDL code. Create two files: i) flip flop, ii) top file (where you will interconnect the flip flops and the logic gates). (10 pts)
  - ✓ Write a VHDL testbench according to the timing diagram shown below. Complete the timing diagram by simulating your circuit (Behavioral Simulation). The clock frequency must be 50 MHz with 50% duty cycle. (13 pts)
- Upload (as a .zip file) the following files to Moodle (an assignment will be created):
  - ✓ VHDL code files and testbench.
  - ✓ A screenshot of your simulation showing the results for Q (this is on top of you completing the timing diagram below).



#### ✓ VHDL Code: Top File

```
library IEEE;
use IEEE.STD LOGIC 1164.ALL;
entity lfsr_mic_rev is
   generic (N: INTEGER:= 4);
   port ( resetn, clock: in std logic;
          E: in std_logic;
          P: in std logic vector (N-1 downto 0);
          Q: out std_logic_vector (N-1 downto 0));
end lfsr mic rev;
architecture structural of lfsr mic rev is
   component dffe
   port ( d : in STD LOGIC;
          clrn: in std_logic:= '1';
          prn: in std logic:= '1';
          clk : in STD LOGIC;
          ena: in std logic;
          q : out STD LOGIC);
   end component;
   signal D, Qt: std_logic_vector (N-1 downto 0);
begin
  D(0) \le Qt(N-1) \text{ xor } Qt(0) \text{ xor } P(0);
  g0: for i in N-1 downto 1 generate
         D(i) <= Qt(i-1) xor P(i);
      end generate;
  gl: for i in 0 to N-1 generate
        di: dffe port map (d => D(i), clrn =>resetn, prn => '1', clk => clock, ena => E, q => Qt(i));
      end generate;
  Q <= Qt;
```

end structural;

#### ✓ VHDL Code: D-Type flip flop

```
library IEEE;
use IEEE.STD LOGIC 1164.ALL;
use IEEE.STD LOGIC UNSIGNED.ALL;
entity dffe is
   port ( d : in STD LOGIC;
           clrn, prn, clk, ena: in std logic;
           q : out STD LOGIC);
end dffe;
architecture behaviour of dffe is
begin
    process (clk, ena, prn, clrn)
    begin
       if clrn = '0' then q <= '0';
       elsif prn = '0' then q <= '1';
       elsif (clk'event and clk='1') then
         if ena = '1' then q <= d; end if;
       end if;
    end process;
end behaviour;
```

#### ✓ VHDL Tesbench:

```
LIBRARY ieee;
USE ieee.std logic 1164.ALL;
ENTITY tb_lfsr_mic_rev IS
    generic (N: integer:= 4);
END tb lfsr mic rev;
ARCHITECTURE behavior OF tb_lfsr_mic_rev IS
   component lfsr_mic_rev
        port ( resetn, clock: in std logic;
               E: in std logic;
               P: in std_logic_vector (N-1 downto 0);
               Q: out std_logic_vector (N-1 downto 0));
   end component;
-- Inputs
  signal E : std logic := '0';
  signal resetn : std_logic := '0';
  signal clock : std_logic := '0';
  signal P: std logic vector (N-1 downto 0):= (others => '0');
-- Outputs
  signal Q : std_logic_vector(N-1 downto 0);
-- Clock period definitions
 constant T : time := 20 ns;
BEGIN
   -- Instantiate the Unit Under Test (UUT)
   uut: lfsr mic rev PORT MAP (resetn => resetn, clock => clock, E => E, P => P, Q => Q);
   -- Clock process definitions
   clock process :process
   begin
     clock <= '0'; wait for T/2;</pre>
    clock <= '1'; wait for T/2;</pre>
   end process;
   -- Stimulus process
   stim proc: process
   begin
        resetn <= '0'; P <= "1101"; wait for 100 ns;
        resetn <= '1';</pre>
        P <= "1101"; E <= '1'; wait for T;
        P <= "1011"; E <= '1'; wait for T;
        P <= "1011"; E <= '1'; wait for T;
        P <= "1011"; E <= '0'; wait for T;
        P <= "1011"; E <= '1'; wait for T;</pre>
        P <= "1101"; E <= '1'; wait for T;
        P <= "1101"; E <= '0'; wait for T;</pre>
        P <= "1101"; E <= '1'; wait for T;
        P <= "1101"; E <= '0'; wait for T;
        P <= "1101"; E <= '1'; wait for 2*T;
        P <= "1101"; E <= '0'; wait for 2*T;
        P <= "1000"; E <= '1'; wait for 4*T;
        wait;
   end process;
```

```
END;
```

Name	Value	100 ns	5 7 9 <i>8 6</i> 7 7 7	200 ns	300	) ns	400 ns
13 clock	1						
₩a resetn	1						
₩a E	1						
> 🐋 P[3:0]	1101	1101 X	1011	X	1101	X	1000
> 📢 Q[3:0]	0101	0000 1101	0001 1000 10	010 1000	1100 0100	0101	0011 1111 0110 0100

## PROBLEM 7 (8 PTS)

- Attach your Project Status Report (no more than 1 page, single-spaced, 2 columns, only one submission per group). This report should contain the initial status of your project. For formatting, use the provided template (Final Project - Report Template.docx). The sections included in the template are the ones required in your Final Report. At this stage, you are only required to:
  - ✓ Include a (draft) project description and title.
  - $\checkmark$  Include a <u>draft</u> Block Diagram of your hardware architecture.



- As a guideline, the figure shows a simple Block Diagram. There are input and output signals, as well as internal components along with their interconnection.
  - ✓ At this stage, only a rough draft is required. There is no need to go into details: it is enough to show the tentative toplevel components that would constitute your system as well as the tentative inputs and outputs.
- Only student is needed to attach the report (make sure to indicate all the team members).