ECE 424 – INTRODUCTION TO VLSI DESIGN + LAB

LABORATORY WORK 7

In this laboratory work, we will study on D flip-flops, T flip-flops and counters.

D-Flip Flop:

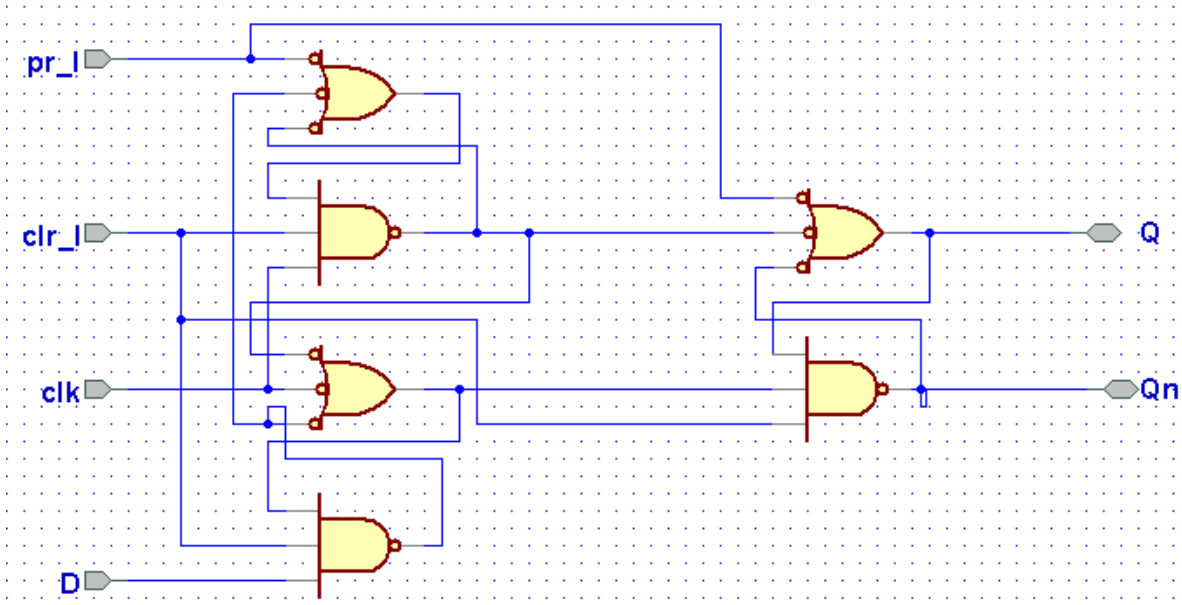

Figure 1 (D Flip Flop)

clr_l	pr_l	Clk	d	q	qn
0	0	X	X	1	1
0	1	X	X	0	1
1	0	X	X	1	0
1	1	↑	0	0	1
1	1	↑	1	1	0

Figure 2 (Truth Table)

```
library IEEE;
use ieee.std_logic_1164.all;
entity dff is
        port (
                pr_l: in STD_LOGIC;      -- active low preset input
                clr_l:in STD_LOGIC;      -- active low clear input
                clk   :in STD_LOGIC;     -- clock input
                d     :in STD_LOGIC;     -- D input
                q     :inout STD_LOGIC;  -- output of D flip flop
                qn    :inout STD_LOGIC   -- inverted output
        );
end dff;
architecture dff of dff is
signal e,f,g,h:std_logic;
component nand3
 port (
                a,b,c: in STD_LOGIC;
                d     : out STD_LOGIC
        );
end component;
begin
  g1:nand3 port map(pr_l,h,f,e);      -- creates g1 gate
  g2:nand3 port map(clr_l,e,clk,f);   -- creates g2 gate
  g3:nand3 port map(f,clk,h,g);       -- creates g3 gate
  g4:nand3 port map(g,clr_l,d,h);     -- creates g4 gate
  g5:nand3 port map(pr_l,f,qn,q);     -- creates g5 gate
  g6:nand3 port map(q,g,clr_l,qn);    -- creates g6 gate
  end dff;
```

Figure 3 (VHDL Code for Dff)

```
library IEEE;
use IEEE.std_logic_1164.all;
entity nand3 is
        port (
                a,b,c: in STD_LOGIC;
                d     : out STD_LOGIC
              );
end nand3;
architecture \nand\ of nand3 is
begin
 d<= not (a and b and c); -- creates a 3 i/p nand gate
end \nand\;
```

Figure 4 (3-input nand gate)

Write thee codes and simulate it by using computer. Try to understand the working principles of the code. Then, write D flip flop behavioral codes.

T Flip-Flop:

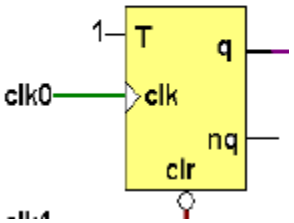

Figure 5 (T Flip Flop)

```
library IEEE;
use IEEE.std_logic_1164.all;

entity tff is
        port (
        t      : in STD_LOGIC;--input to the T-flip flop
        clk    : in STD_LOGIC;--Clock signal for T-flip flop
        clr_1 : in STD_LOGIC;--active low clear input
        q,nq : out STD_LOGIC--actual and complemented outputs of T-flip flop

        );

end tff;

architecture tff of tff is
begin
 process(t,clk,clr_1)
   variable temp:STD_LOGIC:='0';
 begin
  if (clr_1='0') then
        temp:='0';

  elsif ((clr_1='1') and (clk'event and clk='0')) then--perfoms during falling edge
        if ( t='0') then
        temp:=temp;
        else temp:= not temp;
        end if;
  end if;
  q<= temp;
  nq<= not temp;
end process;
end tff;
```

Figure 6 (Behavioral Code of T flip-flop)

Write this code in a new project and simulate it. Don't delete it, you will use it in next part.

4 – Bit Binary Counter:

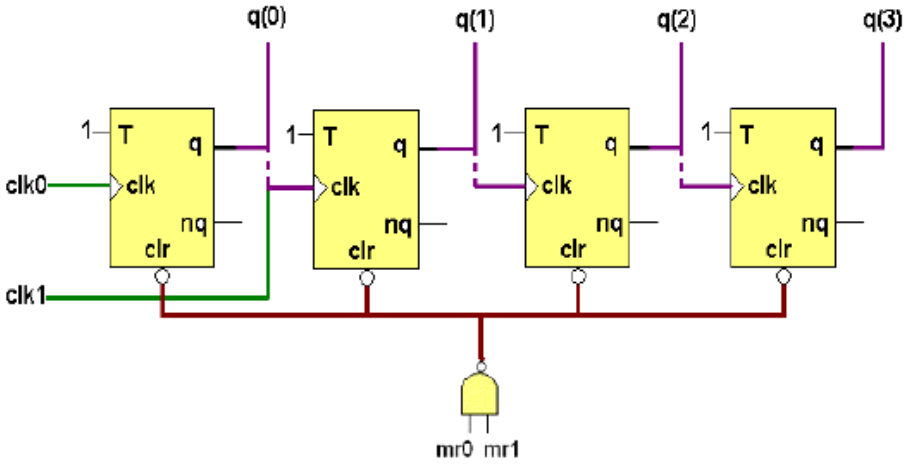

Figure 7 (4 - Bit Binary Counter)

```
library IEEE;
use IEEE.std_logic_1164.all;

entity cnt is
        port (
                clk0: in STD_LOGIC;
                mr0: in STD_LOGIC;
                mr1: in STD_LOGIC;
                clk1: inout STD_LOGIC;
                Q:inout STD_LOGIC_VECTOR(3 downto 0)

        );
end cnt;

architecture cnt of cnt is

Component tff  -- T- flip flop instantiation
        port (
                t      : in STD_LOGIC;
                clk    : in STD_LOGIC;
                clr_1 : in STD_LOGIC;
                q,nq : out STD_LOGIC

        );
end component;
signal clear : std_logic;
begin
        clear<= mr0 nand mr1;  -- common reset inputs for mod2 and mod8
                               --counters
        CLK1<=q(0); --to work as asynchronous mod16 counter
        t1:tff port map('1',clk0,clear,Q(0),open);--t1,t2,t3,t4 create four T-flip flops
        t2:tff port map('1',clk1,clear,Q(1), open);
        t3:tff port map('1',Q(1),clear,Q(2), open);
        t4:tff port map('1',Q(2),clear,Q(3), open);

 end cnt;
```

Figure 8 (VHDL Codes for 4-bit counter)

Write the code and simulate it.

Homework: Write N-bit counter by using “generic” method. Make a research about the usage of this method and write a behavioral code for N bit counter.

(For simulation, you can take N=8).

Send me whole project folders. (Not screenshots or just codes)

Deadline: 24.12.2013 Tuesday, 21:00.

No late submission is going to be accepted!

Prepared By: **İbrahim BOZKURT**
e-mail: ibrahimbozkurt@cankaya.edu.tr