KURUKULASOORIYA KVR 200332X

COUNTER WITH EXTERNAL INPUT

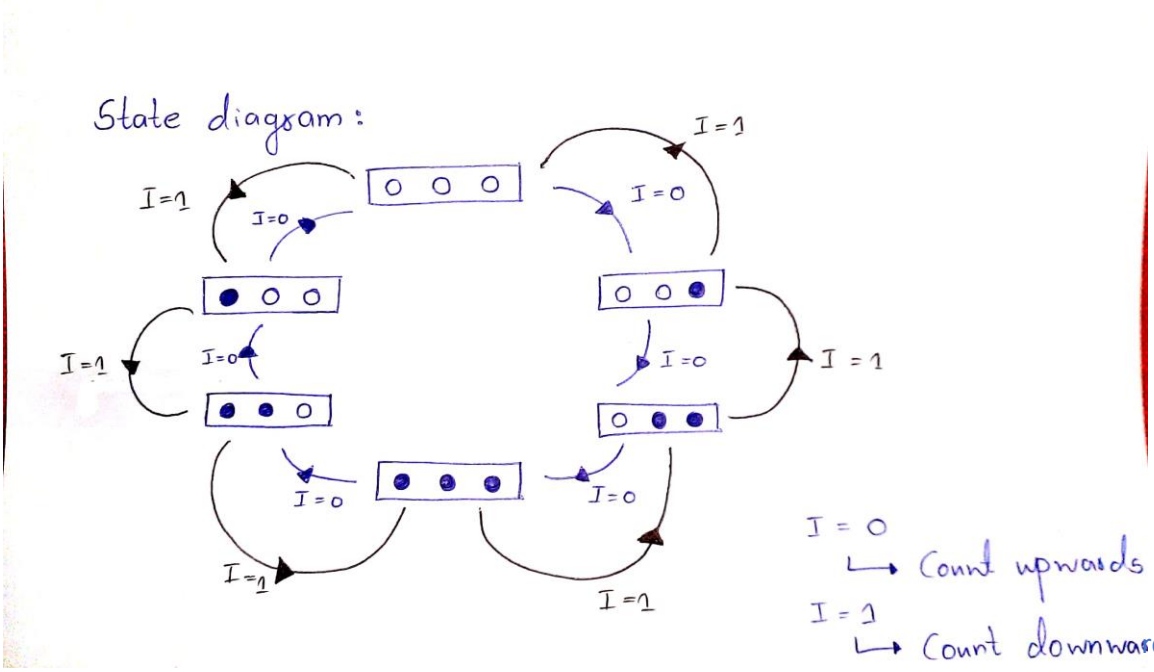
LAB 05 REPORT

Task Assigned:

- Identify the functionality of the 3 bit counter according to the state diagram given and construct it's excitation table using D flipflop's excitation table.
- Using Karnaugh maps, obtain the simplified logic expression for each bit in the counter.
- Design and develop the counter in Vivado software according to following steps:
    1. Develop a source code for a D flipflop in VHDL and verify it's functionality via a simulation.
    2. Develop a source code for a slow clock in VHDL and verify it's functionality via a simulation.
    3. Combining above two circuits (a slow clock and 3 D flipflops), develop a source code for the 3 bit counter and verify it's functionality via a simulation

Excitation table for the 3 bit counter:

I (Button)	Current Stage (Q_t)			Next Stage (Q_{t+1})			D_2	D_1	D_0
	Q_2	Q_1	Q_0	Q_2	Q_1	Q_0			
0	0	0	0	0	0	1	**0**	**0**	**1**
0	0	0	1	0	1	1	**0**	**1**	**1**
0	0	1	0	X	X	X	**X**	**X**	**X**
0	0	1	1	1	1	1	**1**	**1**	**1**
0	1	0	0	0	0	0	**0**	**0**	**0**
0	1	0	1	X	X	X	**X**	**X**	**X**
0	1	1	0	1	0	0	**1**	**0**	**0**
0	1	1	1	1	1	0	**1**	**1**	**0**
1	0	0	0	1	0	0	**1**	**0**	**0**
1	0	0	1	0	0	0	**0**	**0**	**0**
1	0	1	0	X	X	X	**X**	**X**	**X**
1	0	1	1	0	0	1	**0**	**0**	**1**
1	1	0	0	1	1	0	**1**	**1**	**0**
1	1	0	1	X	X	X	**X**	**X**	**X**
1	1	1	0	1	1	1	**1**	**1**	**1**
1	1	1	1	0	1	1	**0**	**1**	**1**



Derived Karnaugh maps & simplified logic expressions:

IQ_2 \ Q_1Q_0	00	01	11	10
00	0	0	1	X
01	0	X	1	1
11	1	X	0	1
10	1	0	0	X

$$D_2 = \overline{I}.Q_1 + \overline{Q}_0.I + Q_1.\overline{Q}_0$$

IQ_2 \ Q_1Q_0	00	01	11	10
00	0	1	1	X
01	0	X	1	0
11	1	X	1	1
10	0	0	0	X

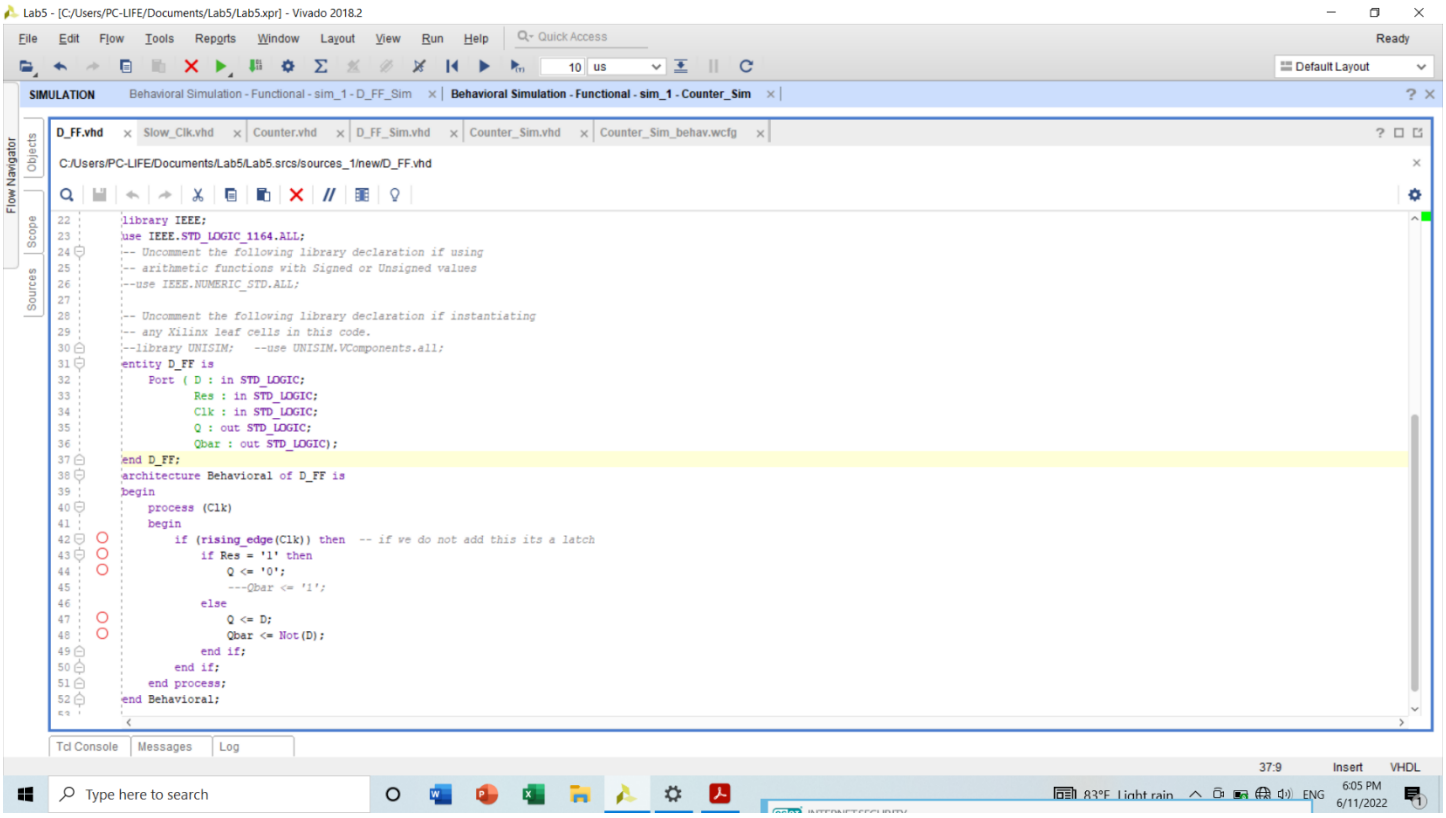
$$D_1 = Q_0.\overline{I} + Q_0.Q_2 + I.Q_2$$

IQ_2 \ Q_1Q_0	00	01	11	10
00	1	1	1	X
01	0	X	0	0
11	0	X	1	1
10	0	0	1	X

$$D_0 = Q_1.I + Q_1.\overline{Q}_2 + \overline{I}.\overline{Q}_2$$

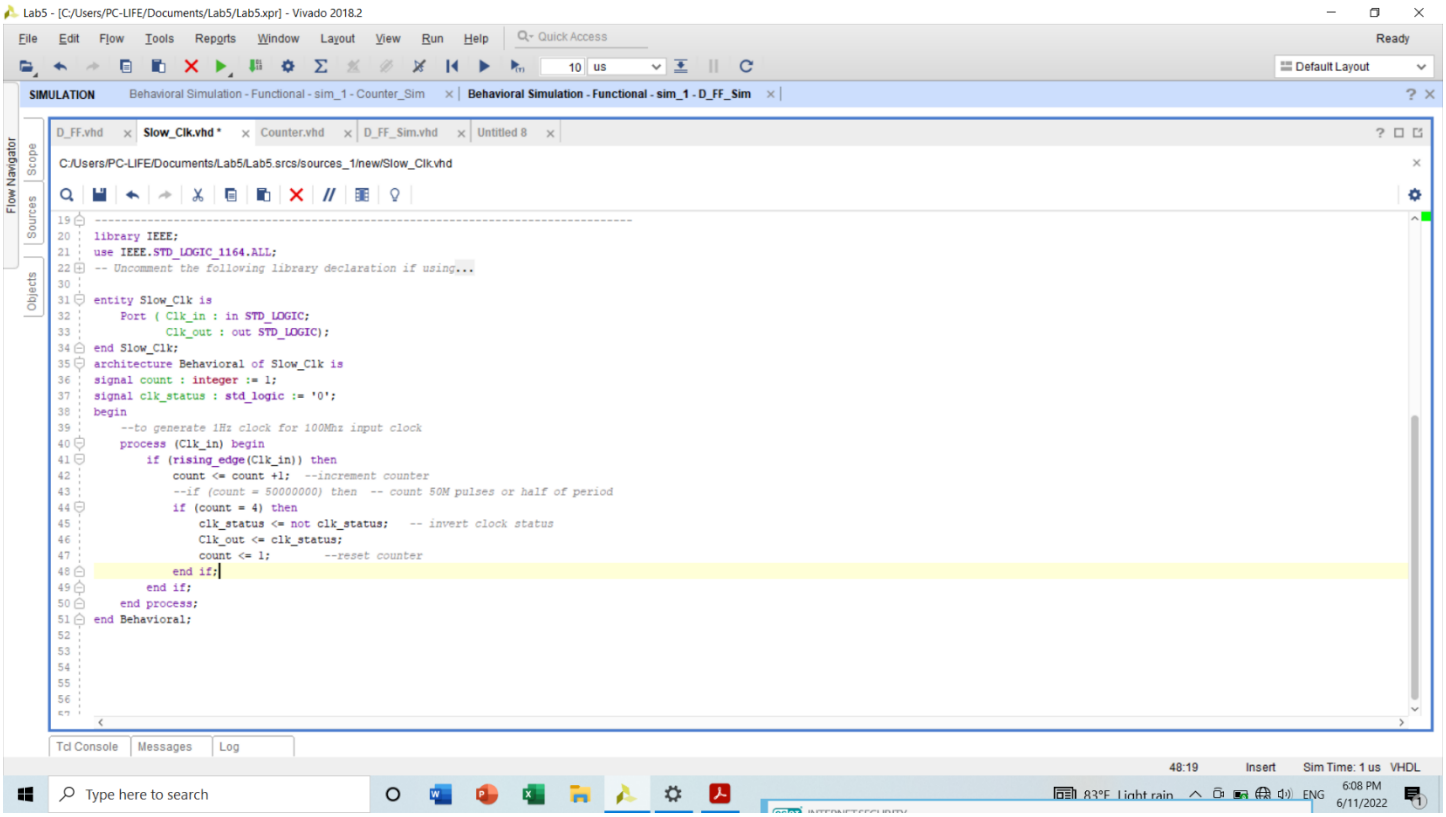
VHDL Codes:

01)VHDL SOURCE CODE FOR D FLIPFLOP



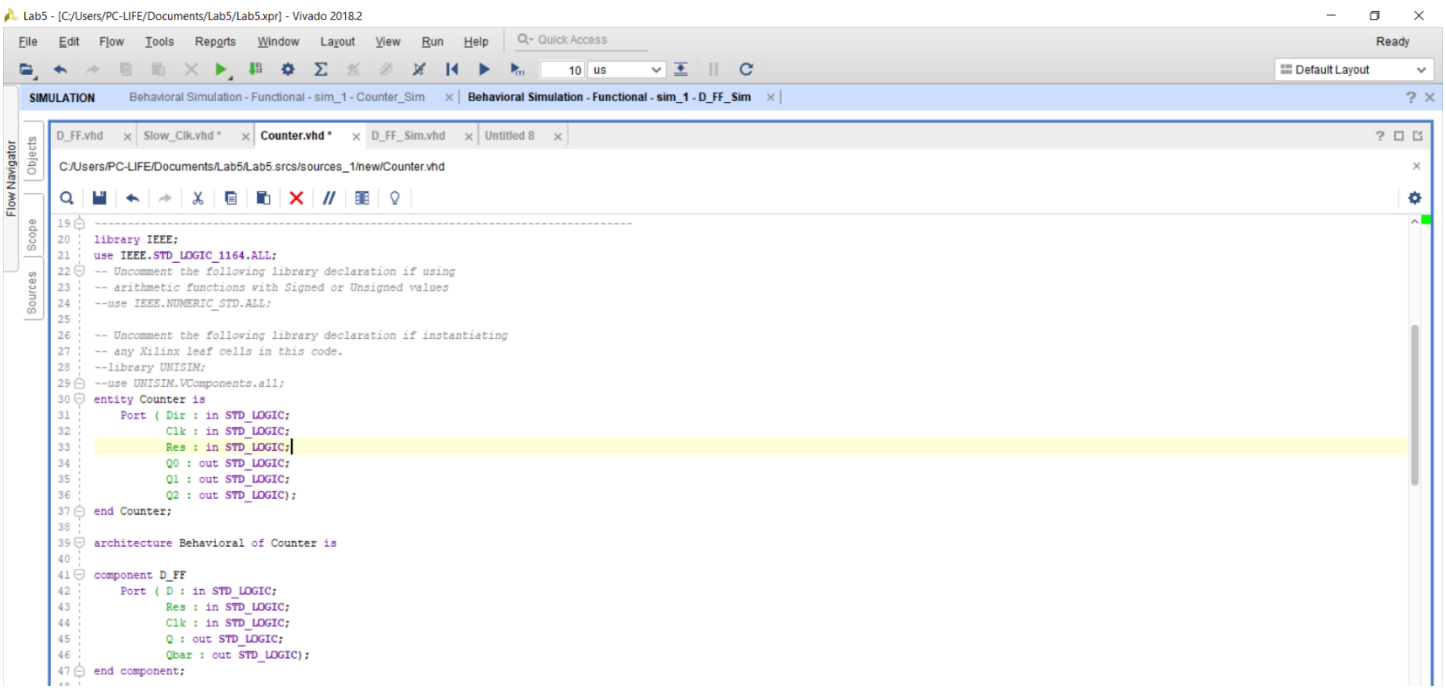
```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;

-- Uncomment the following library declaration if instantiating
-- any Xilinx leaf cells in this code.
--library UNISIM;   --use UNISIM.VComponents.all;
entity D_FF is
    Port ( D : in STD_LOGIC;
           Res : in STD_LOGIC;
           Clk : in STD_LOGIC;
           Q : out STD_LOGIC;
           Qbar : out STD_LOGIC);
end D_FF;
architecture Behavioral of D_FF is
begin
    process (Clk)
    begin
        if (rising_edge(Clk)) then  -- if we do not add this its a latch
            if Res = '1' then
                Q <= '0';
                ---Qbar <= '1';
            else
                Q <= D;
                Qbar <= Not(D);
            end if;
        end if;
    end process;
end Behavioral;
```

02)VHDL SOURCE CODE FOR SLOW CLOCK

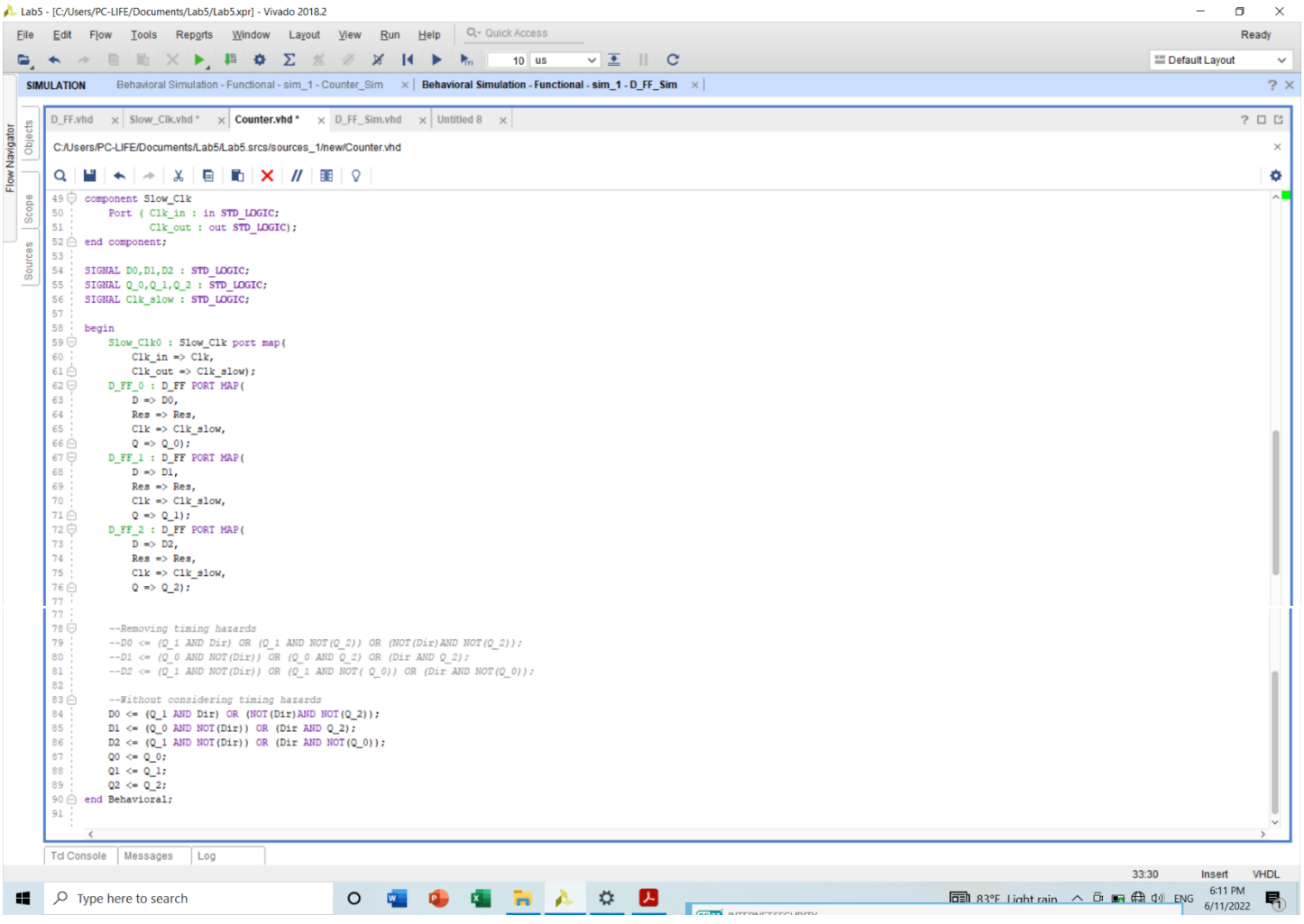


```
----------------------------------------------------------------------
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
-- Uncomment the following library declaration if using...

entity Slow_Clk is
    Port ( Clk_in : in STD_LOGIC;
           Clk_out : out STD_LOGIC);
end Slow_Clk;
architecture Behavioral of Slow_Clk is
signal count : integer := 1;
signal clk_status : std_logic := '0';
begin
    --to generate 1Hz clock for 100Mhz input clock
    process (Clk_in) begin
        if (rising_edge(Clk_in)) then
            count <= count +1;  --increment counter
            --if (count = 50000000) then  -- count 50M pulses or half of period
            if (count = 4) then
                clk_status <= not clk_status;   -- invert clock status
                Clk_out <= clk_status;
                count <= 1;         --reset counter
            end if;
        end if;
    end process;
end Behavioral;
```

03)VHDL SOURCE CODE FOR 3 BIT COUNTER



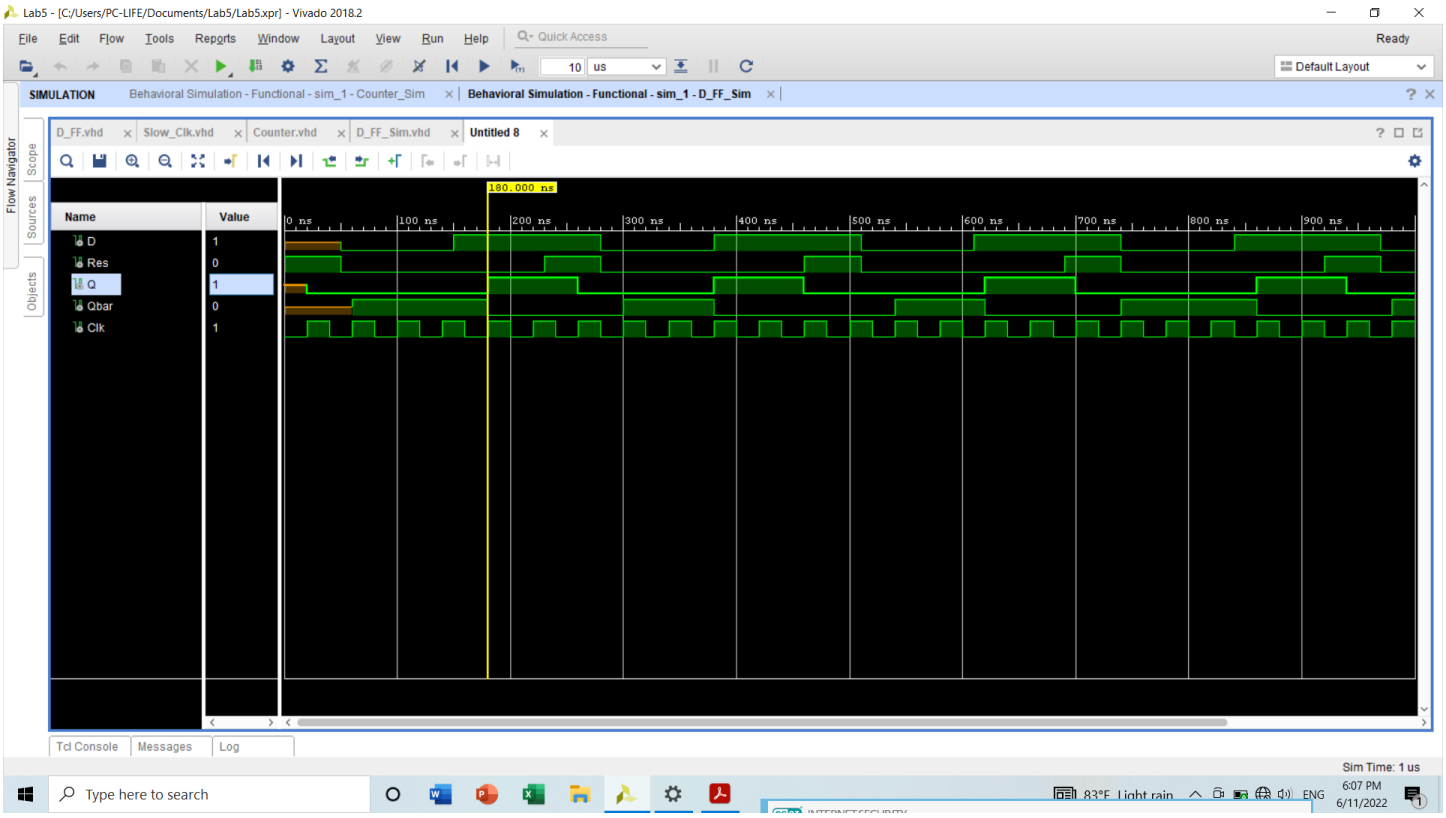
```
----------------------------------------------------------------------------------
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;

-- Uncomment the following library declaration if instantiating
-- any Xilinx leaf cells in this code.
--library UNISIM;
--use UNISIM.VComponents.all;
entity Counter is
    Port ( Dir : in STD_LOGIC;
           Clk : in STD_LOGIC;
           Res : in STD_LOGIC;
           Q0 : out STD_LOGIC;
           Q1 : out STD_LOGIC;
           Q2 : out STD_LOGIC);
end Counter;

architecture Behavioral of Counter is

component D_FF
    Port ( D : in STD_LOGIC;
           Res : in STD_LOGIC;
           Clk : in STD_LOGIC;
           Q : out STD_LOGIC;
           Qbar : out STD_LOGIC);
end component;
```



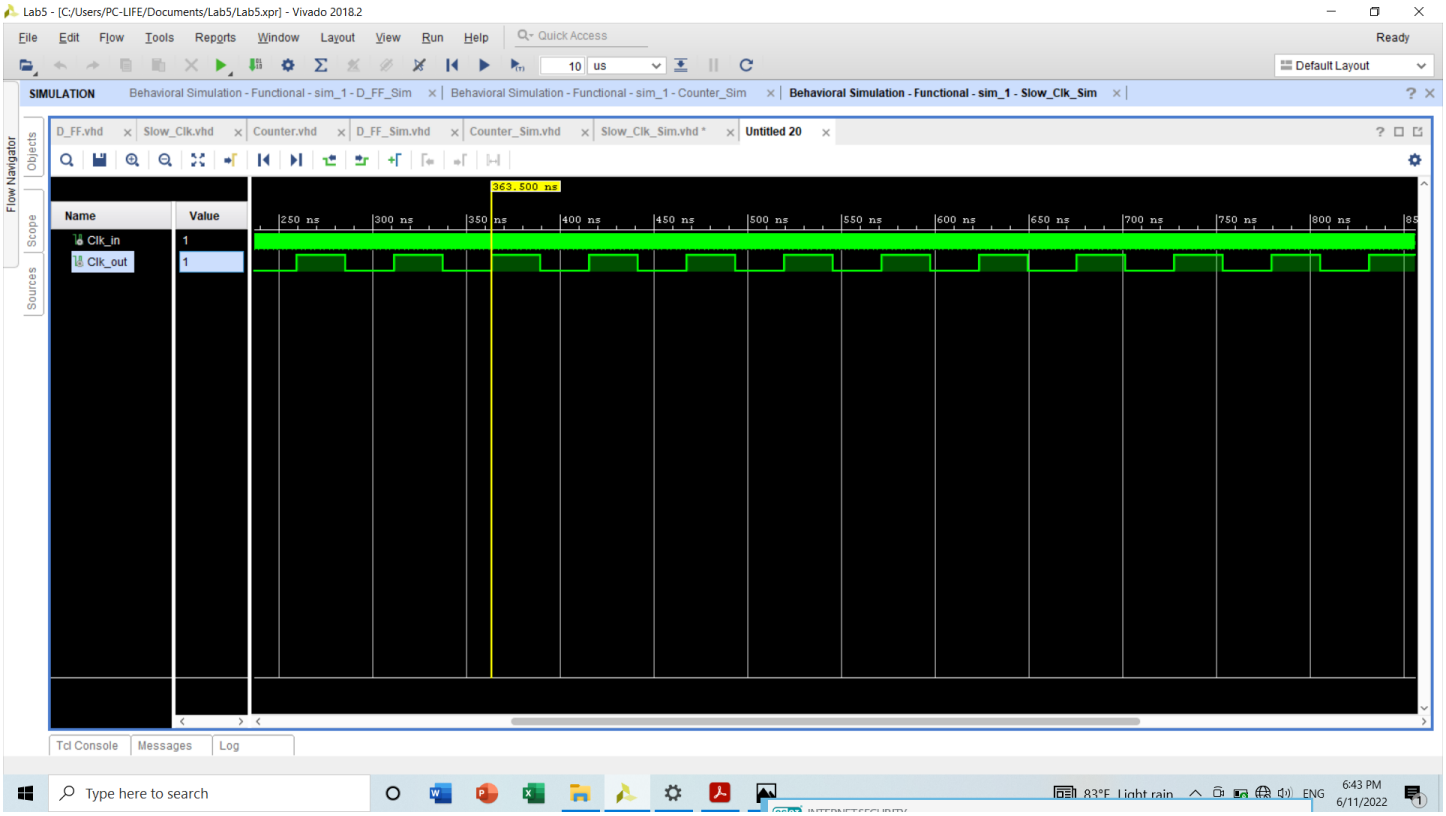
```
component Slow_Clk
    Port ( Clk_in : in STD_LOGIC;
           Clk_out : out STD_LOGIC);
end component;

SIGNAL D0,D1,D2 : STD_LOGIC;
SIGNAL Q_0,Q_1,Q_2 : STD_LOGIC;
SIGNAL Clk_slow : STD_LOGIC;

begin
    Slow_Clk0 : Slow_Clk port map(
        Clk_in => Clk,
        Clk_out => Clk_slow);
    D_FF_0 : D_FF PORT MAP(
        D => D0,
        Res => Res,
        Clk => Clk_slow,
        Q => Q_0);
    D_FF_1 : D_FF PORT MAP(
        D => D1,
        Res => Res,
        Clk => Clk_slow,
        Q => Q_1);
    D_FF_2 : D_FF PORT MAP(
        D => D2,
        Res => Res,
        Clk => Clk_slow,
        Q => Q_2);


    --Removing timing hazards
    --D0 <= (Q_1 AND Dir) OR (Q_1 AND NOT(Q_2)) OR (NOT(Dir)AND NOT(Q_2));
    --D1 <= (Q_0 AND NOT(Dir)) OR (Q_0 AND Q_2) OR (Dir AND Q_2);
    --D2 <= (Q_1 AND NOT(Dir)) OR (Q_1 AND NOT( Q_0)) OR (Dir AND NOT(Q_0));

    --Without considering timing hazards
    D0 <= (Q_1 AND Dir) OR (NOT(Dir)AND NOT(Q_2));
    D1 <= (Q_0 AND NOT(Dir)) OR (Dir AND Q_2);
    D2 <= (Q_1 AND NOT(Dir)) OR (Dir AND NOT(Q_0));
    Q0 <= Q_0;
    Q1 <= Q_1;
    Q2 <= Q_2;
end Behavioral;
```

Timing Diagrams:

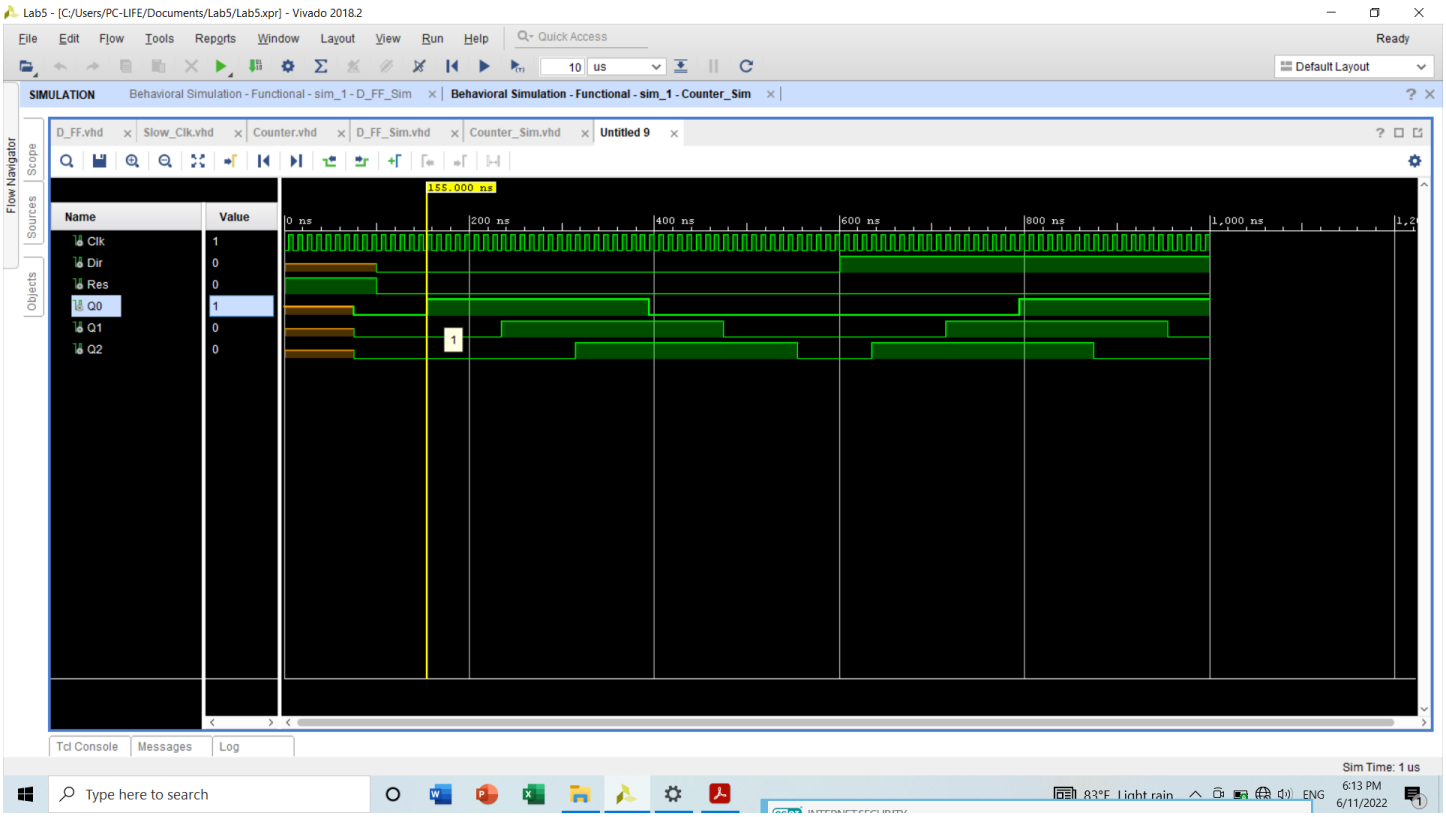
01)TIMING DIAGRAM FOR D FLIPFLOP



02)TIMING DIAGRAM FOR SLOW CLOCK

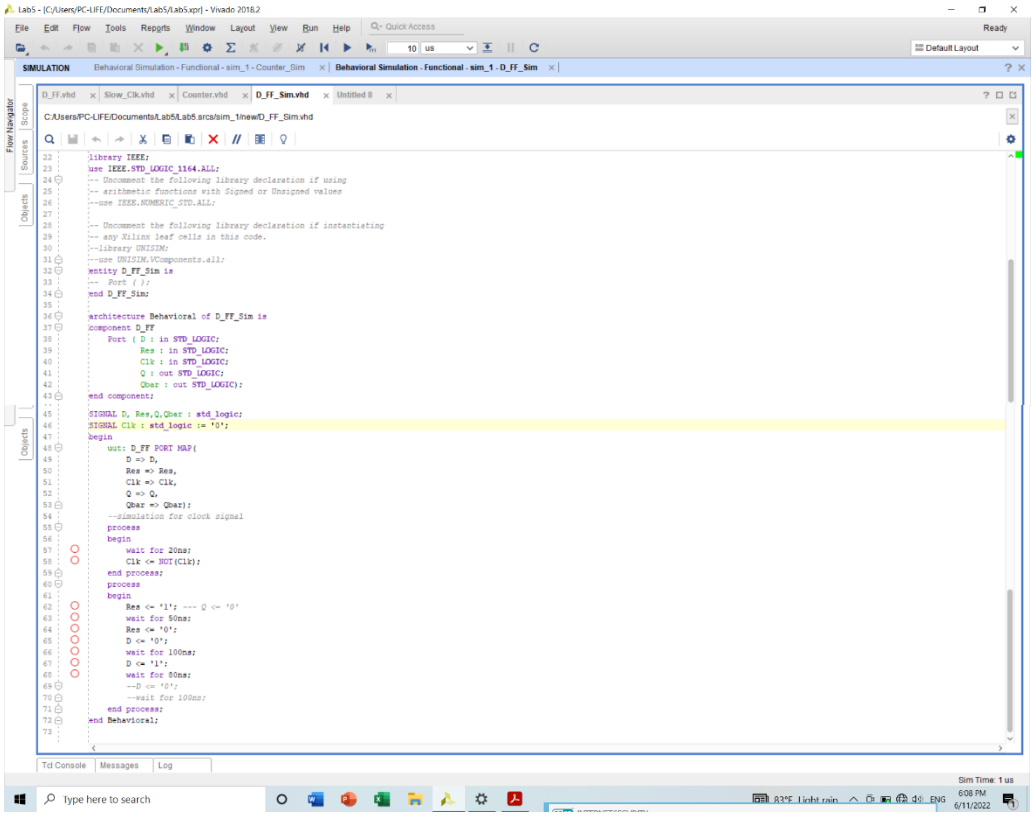


03)TIMING DIAGRAM FOR 3 BIT COUNTER

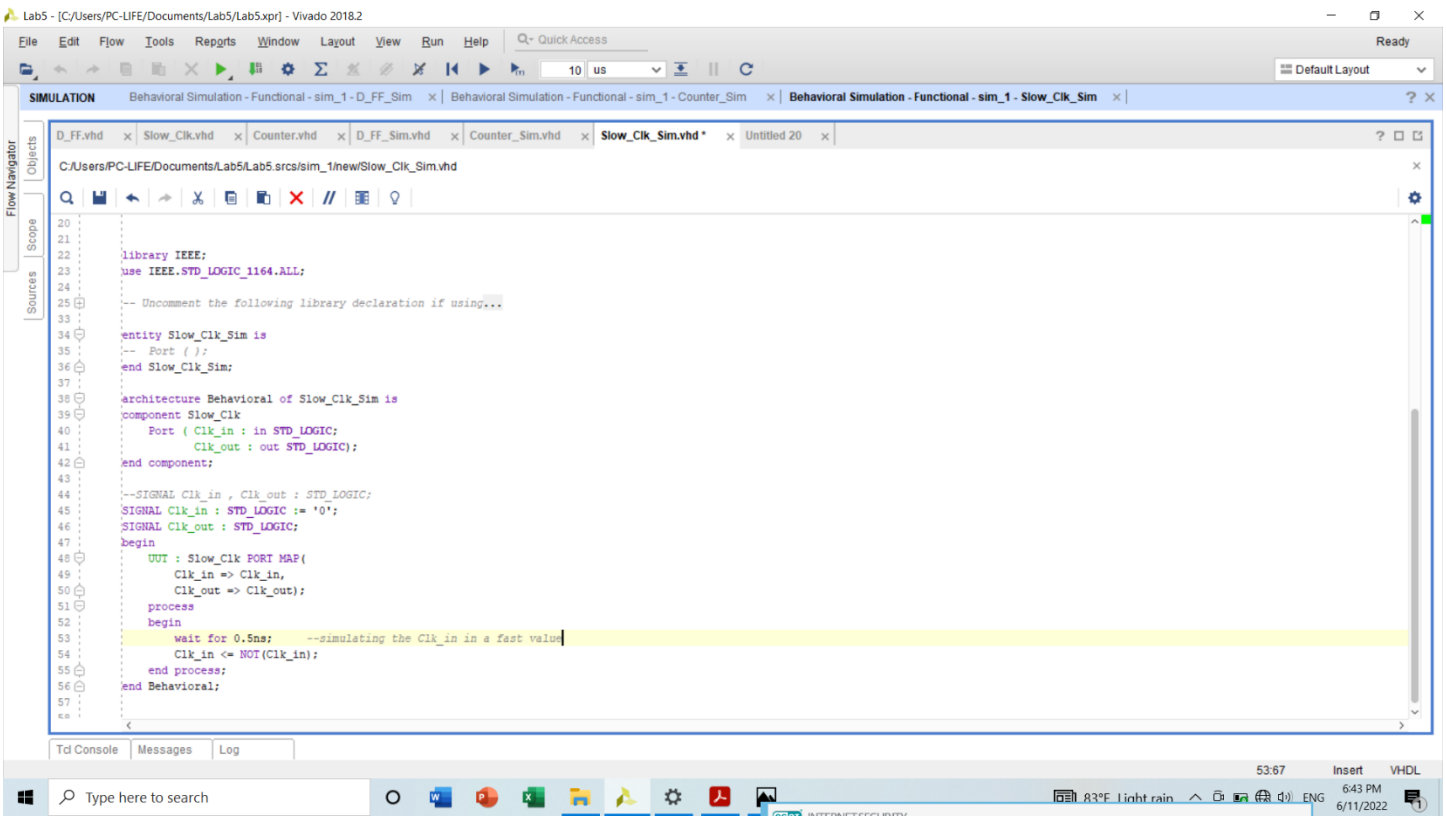


Test bench codes for simulation:

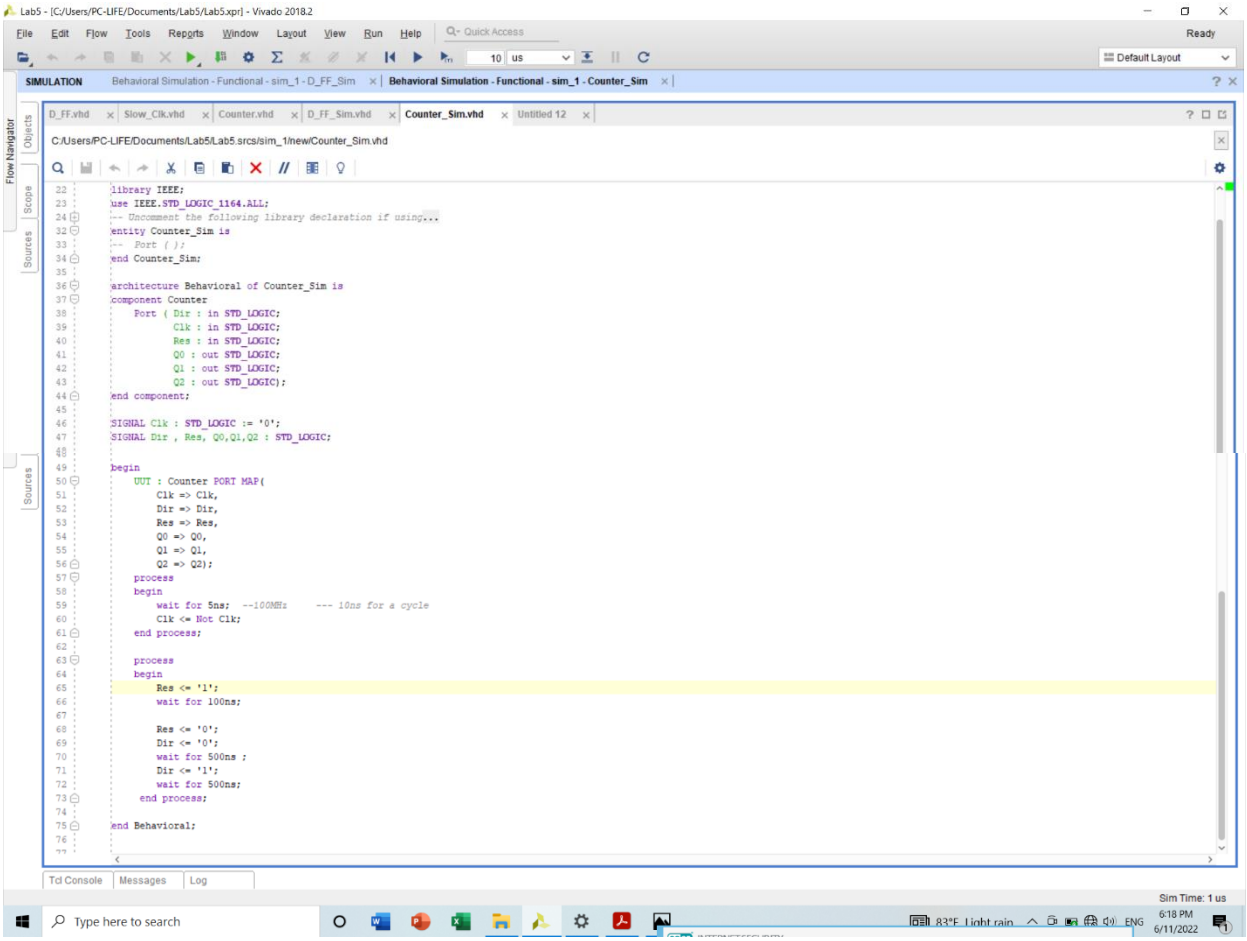
01)TB FOR D FLIPFLOP'S SIMULATION



02)TB FOR SLOW CLOCK'S SIMULATION



03)TB FOR 3 BIT COUNTER'S SIMULATION



Conclusions:

- Sequential logic circuits **(We used D flipflops for the lab)** allow it's output to be depend not only on the external input given but also on the past sequence of outputs.

In the lab, a 3 bit counter is developed using 3 D flipflops which could maintain a memory of past outputs and also could response to the external input on the counting direction whether it's clockwise or anticlockwise.

- Clock divider **(Slow_Clk in our lab)** allows to take an input signal of frequency f_{in} and generate an output signal of a frequency $f_{out} = \frac{fin}{n}$ where n is an integer.
- Vivado's simulation allows to verify the functionality of the source codes of the circuits designed.