Lab 5 - Counter

CS1050 Computer Organization and Digital Design

Name : **Dissanayake D.M.A.K.**
Index No. : **220135N**
Group : **44**

Lab Task:

Designing a 3 bit up / down counter using D flip-flops and slow down clock. First we completed the truth table and got expressions for D0, D1, D2. Then we created D flip flop and slow down clock. By using these D flip flops and slow down clock we created the counter.

Introduction

We initiate the lab by examining D flip-flop excitation tables and streamlining equations using Karnaugh Maps. Following this, we progress to VHDL modeling of D flip-flops. Additionally, we establish a slowing counter model to effectively regulate clock frequencies, ensuring optimal visibility.

The central objectives of the lab involve constructing the 3-bit counter circuit, incorporating the derived Boolean expressions, and integrating the slowing counter to manage clock pulses efficiently. Rigorous simulation is employed to validate the counter's performance and functionality before advancing to hardware verification using the BASYS 3 board.

Excitation Table of D Flip-Flop

Q_t	Q_{t+1}	D
0	0	0
0	1	1
1	0	0
1	1	1

State Table of Counter

Q_t			Button	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	1	0	0	1	0	0
0	0	1	0	0	1	1	0	1	1
0	0	1	1	0	0	0	0	0	0
0	1	0	0	X	X	X	X	X	X
0	1	0	1	X	X	X	X	X	X
0	1	1	0	1	1	1	1	1	1
0	1	1	1	0	0	1	0	0	1
1	0	0	0	0	0	0	0	0	0
1	0	0	1	1	1	0	1	1	0
1	0	1	0	X	X	X	X	X	X
1	0	1	1	X	X	X	X	X	X
1	1	0	0	1	0	0	1	0	0
1	1	0	1	1	1	1	1	1	1
1	1	1	0	1	1	0	1	1	0
1	1	1	1	0	1	1	0	1	1

Karnaugh Maps for three Inputs

We can derive the boolean expressions for the inputs D2, D1 and D0 based Q_t and B.

For D0

Q_2Q_1 / Q_0B	00	01	11	10
00	1	X	0	0
01	0	X	1	0
11	0	1	1	X
10	1	1	0	X

$D_0 = Q_1.B + Q'_2.B'$

For D1

Q_0B \ Q_2Q_1	00	01	11	10
00	0	X	0	0
01	0	X	1	1
11	0	0	1	X
10	1	1	1	X

$$D_1 = Q_2.B + Q_0.B'$$

For D2

Q_0B \ Q_2Q_1	00	01	11	10
00	0	X	1	0
01	1	X	1	1
11	0	0	0	X
10	0	1	1	X

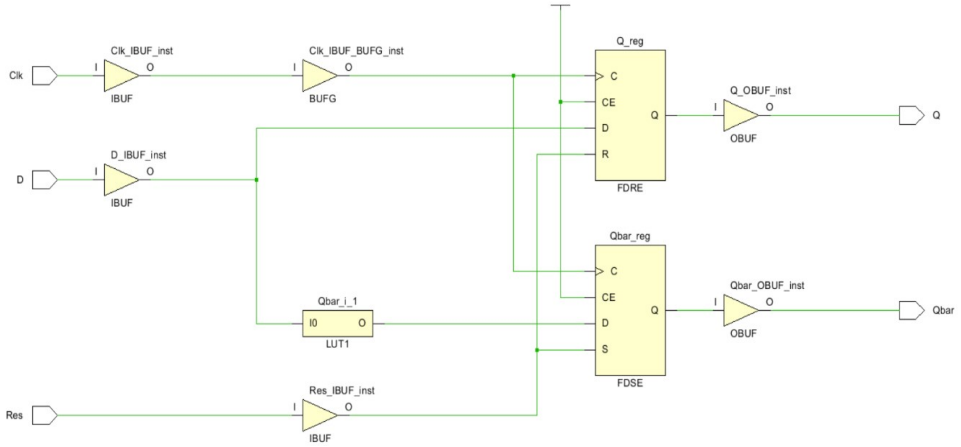
$$D_2 = Q'_0.B + Q_1.B'$$

1)D Flip Flop

Design Source File

```
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 Res = '1'then
      Q <= '0';
      Qbar <= '1';
    else
      Q <= D;
      Qbar <= not D;
    end if;
  end if;
end process;

end Behavioral;
```

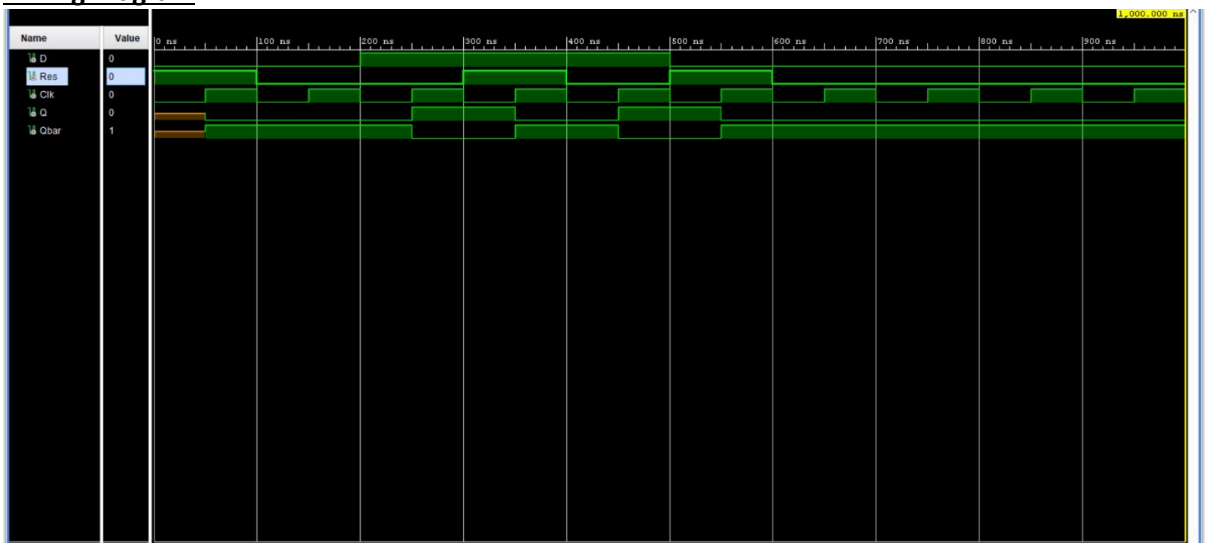
Elaborated design schematic.



Simulation source file

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity D_FF_Sim is
--  Port ( );
end D_FF_Sim;

architecture Behavioral of D_FF_Sim 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;

    Signal D : STD_LOGIC;
    Signal Res : STD_LOGIC;
    Signal Clk : STD_LOGIC;
    Signal Q : STD_LOGIC;
    Signal Qbar : STD_LOGIC;

begin

UUT: D_FF port map(
    D => D,
    Res => Res,
    Clk => Clk,
    Q => Q,
    Qbar => Qbar);

Clk_process:
process
     begin
     Clk <= '0';
     wait for 50ns;

     Clk <= '1';
     wait for 50ns;

end process;

process
begin
     D <= '0';      Res <= '1';      wait for 100 ns;
     Res <= '0';      D <= '0';      wait for 100 ns;
     Res <= '0';      D <= '1';      wait for 100 ns;
     Res <= '1';      D <= '1';      wait for 100 ns;
     Res <= '0';      D <= '1';      wait for 100 ns;
     Res <= '1';      D <= '0';      wait for 100 ns;
     Res <= '0';      D <= '0';      wait for 100 ns;
     wait;
end process;
end behavioral;
```

Timing Diagram

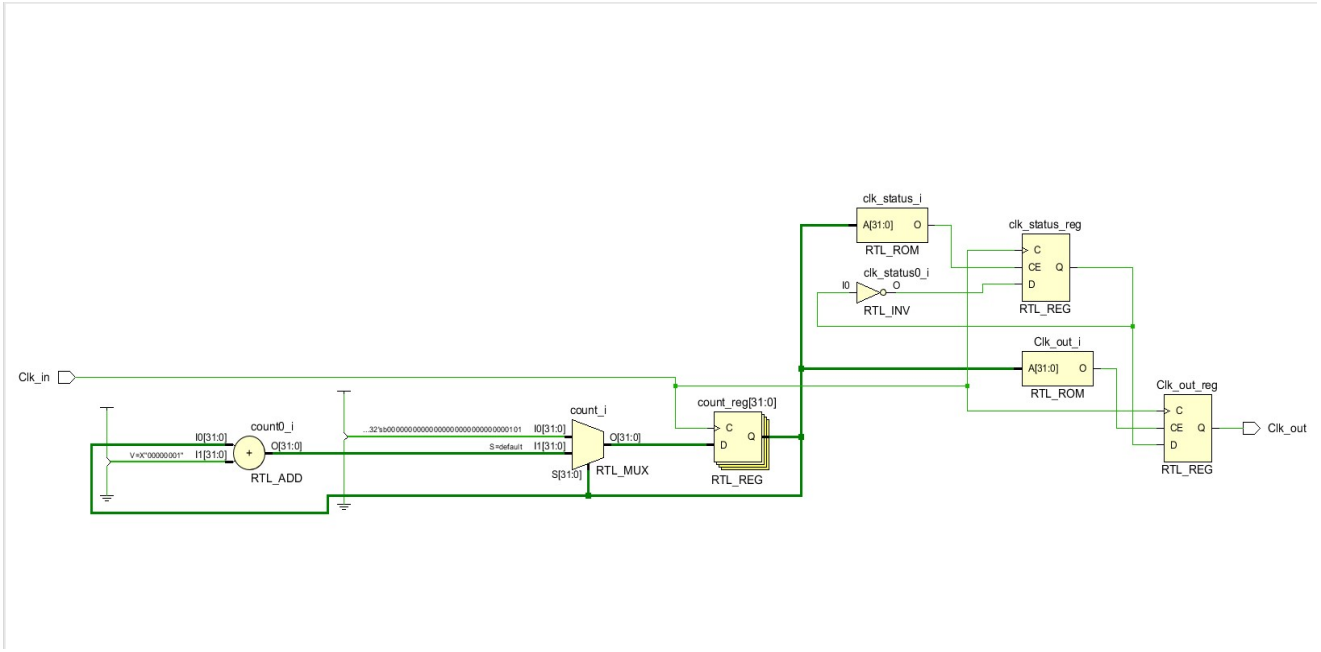


2) Slow Clock

Design Source File

```
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 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
process (Clk_in) begin
         if (rising_edge(Clk_in)) then
             count <= count + 1;
             if(count = 5) then
                 clk_status <= not clk_status;
                 Clk_out <= clk_status;
                 count <= 1;
             end if;
          end if;
     end process;
end Behavioral;
```

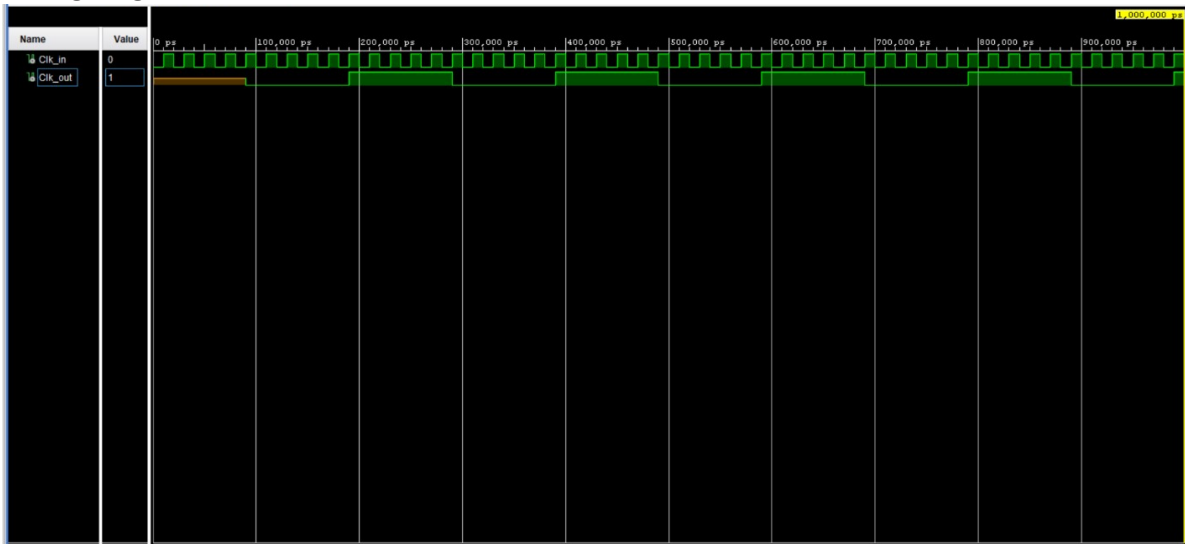
Elaborated design schematic



Simulation File

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity Slow_Clk_Sim is
-- Port ( );
end Slow_Clk_Sim;

architecture Behavioral of Slow_Clk_Sim is

signal Clk_in: STD_LOGIC := '0';
signal Clk_out: STD_LOGIC;

begin
 -- Instantiate the Unit Under Test (UUT)

UUT:Slow_Clk port map(
 Clk_in => Clk_in,
 Clk_out => Clk_out
 );
 -- Clock process definitions

Clk_process:
Process
 begin
  Clk_in <= '0';
  wait for 10ns;
  Clk_in <= '1';
  wait for 10ns;
  end process;
 end Behavioral;
```

Timing Diagram



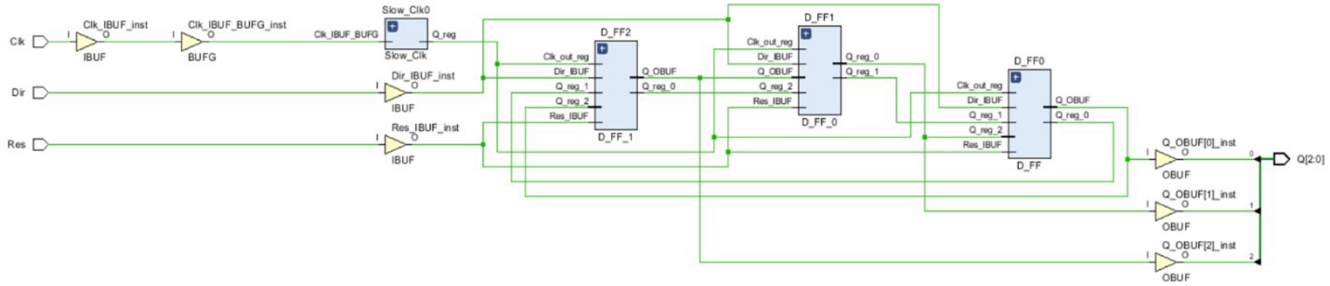
3) Counter

Design Source File

```
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;
           Res : in STD_LOGIC;
           Clk : in STD_LOGIC;
           Q : out STD_LOGIC_VECTOR (2 downto 0));
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; -- Internal signals
signal Q0, Q1, Q2 : std_logic; -- Internal signals
signal Clk_Slow   : std_logic; -- Internal clock

begin
Slow_Clk0 : Slow_Clk
port map (
         Clk_in => Clk,
         Clk_out => Clk_slow);

     D0 <= ((not Q2) and (not Dir)) or (Q1 and Dir);
     D1 <= ((not Dir) and Q0) or (Dir and Q2);
     D2 <= ((not Dir) and Q1) or (Dir and (not Q0));

D_FF0 : D_FF
port map (
         D => D0,
         Res => Res,
         Clk => Clk_slow,
         Q => Q0);

D_FF1 : D_FF
port map (
         D => D1,
         Res => Res,
         Clk => Clk_slow,
         Q => Q1);

D_FF2 : D_FF
port map (
         D => D2,
         Res => Res,
         Clk => Clk_slow,
         Q => Q2);

     Q(0) <= Q0;
     Q(1) <= Q1;
     Q(2) <= Q2;

end Behavioral;
```

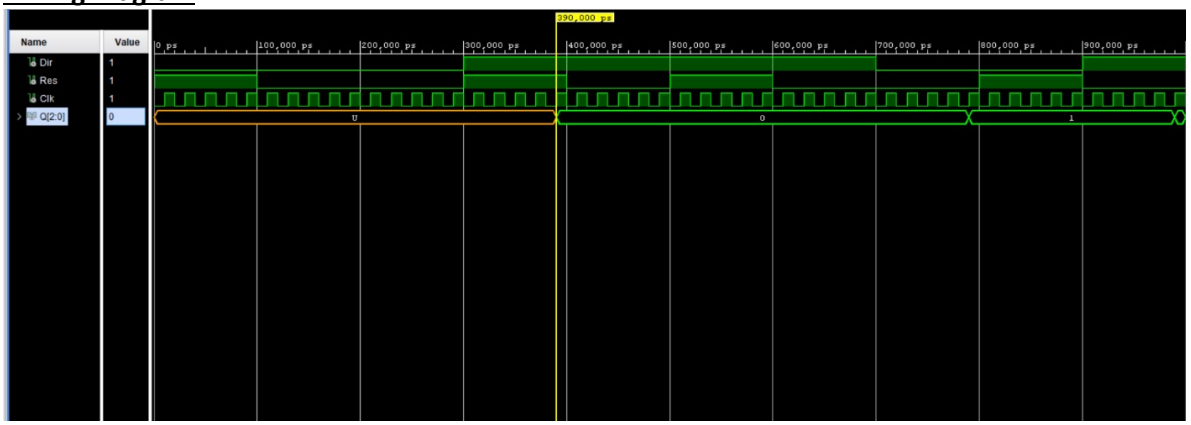
Elaborated design schematic



Simulation File

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity Counter_sim is
--  Port ( );
end Counter_sim;

architecture Behavioral of Counter_sim is

  -- Component declarations
component Counter
    Port ( Dir : in STD_LOGIC;
           Res : in STD_LOGIC;
           Clk : in STD_LOGIC;
           Q : out STD_LOGIC_VECTOR (2 downto 0));
end component;

  -- Signal declarations
signal Dir : in STD_LOGIC;
signal Res : in STD_LOGIC;
signal Clk : in STD_LOGIC;
signal Q : out STD_LOGIC_VECTOR (2 downto 0));


begin
UUT: Counter
  Port map (Dir => Dir,
            Res => Res,
            Clk => Clk,
            Q => Q);

Clk_process:
process
begin
      Clk <= '0';           wait for 10ns;
      Clk <= '1';           wait for 10ns;
end process;

process
begin
      Res <= '1';      Dir <= '0';     wait for 100ns;
      Dir <= '0';      Res <= '0';     wait for 100ns;
      Res <= '0';      wait for 100ns;
      Res <= '1';      Dir <= '1';     wait for 100ns;
      Dir <= '1';      Res <= '0';     wait for 100ns;
      Res <= '1';      wait for 100ns;
      Res <= '0';      Dir <= '1';     wait for 100ns;
      Dir <= '0';      Res <= '0';     wait for 100ns;
      Res <= '1';      wait for 100ns;
      Res <= '0';      Dir <= '1';     wait for 100ns;
      Dir <= '1';      Res <= '1';     wait for 100ns;
      Res <= '0';      wait for 100ns;
      Res <= '0';      Dir <= '0';     wait for 100ns;
      Dir <= '0';
      wait;

end process;

end Behavioral;
```

Timing Diagram



Constraints File

```
## Clock signal
set_property PACKAGE_PIN W5 [get_ports Clk]
set_property IOSTANDARD LVCMOS33 [get_ports Clk]
create_clock -add -name sys_clk_pin -period 10.00 -waveform
{0 5} [get_ports Clk]

## Switches
set_property PACKAGE_PIN V17 [get_ports {Dir}]
set_property IOSTANDARD LVCMOS33 [get_ports {Dir}]

## LEDs
set_property PACKAGE_PIN U16 [get_ports {Q(0)}]
set_property IOSTANDARD LVCMOS33 [get_ports {Q(0)}]
set_property PACKAGE_PIN E19 [get_ports {Q(1)}]
set_property IOSTANDARD LVCMOS33 [get_ports {Q(1)}]
set_property PACKAGE_PIN U19 [get_ports {Q(2)}]
set_property IOSTANDARD LVCMOS33 [get_ports {Q(2)}]

##Buttons
set_property PACKAGE_PIN U17 [get_ports Res]
set_property IOSTANDARD LVCMOS33 [get_ports Res]
```

Conclusion

A counter is like a digital gadget that can either increase or decrease its number with each tick of a clock. You can decide if it should count up or down using an external switch. It basically goes through different stages, and each time the clock ticks, the number it shows changes. Depending on what kind of counter it is, you can make it change by giving it a clock tick or using other buttons. People use counters in lots of electronic things to measure time or how often something happens.

-End-