Assignment- 12

ELP - 718 Telecom Software Laboratory

Ch Krishna Chaitanya
2019JTM2674
2019-21

A report on Behavioural Modelling in VHDL

Bharti School of
Telecommunication Technology and Management
IIT Delhi
India

October 30, 2019

Contents

Objective Statement

To test our understanding and implementation of digital logic in VHDL using Vivado simulator.

1 Problem Statement -1

To model a hardware that detects pattern '10110101' in input bit stream.

1.1 Algorithm and Implementation

- Create Entity with required input ports and output ports.
- Draw a state diagram using Moore state machine
- Implement the logic in Vivado
- Write a test bench with required inputs
- Run simulation and observe output

1.2 State Diagram

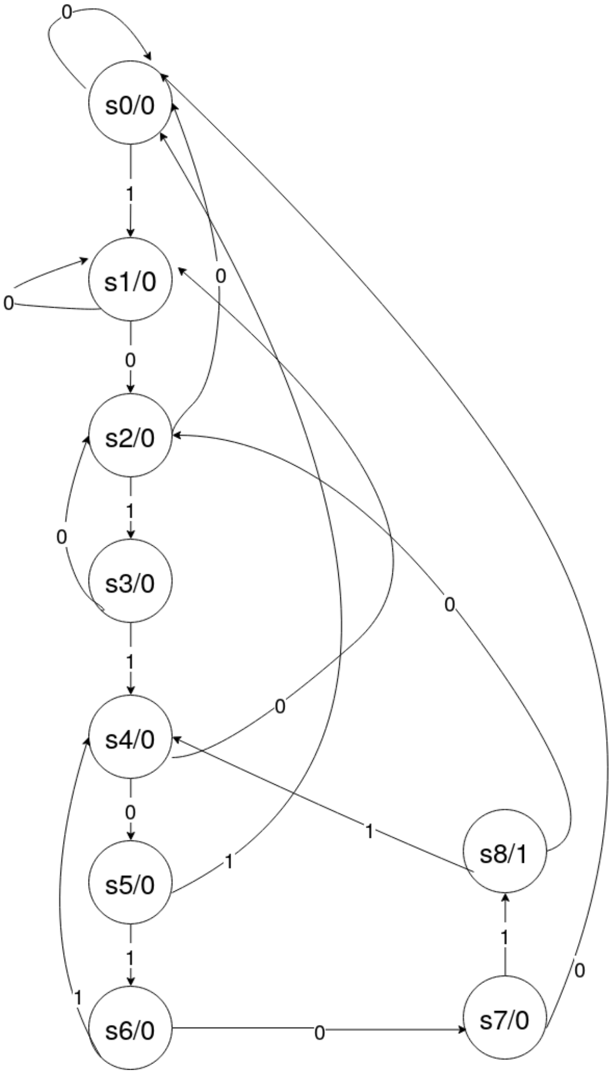
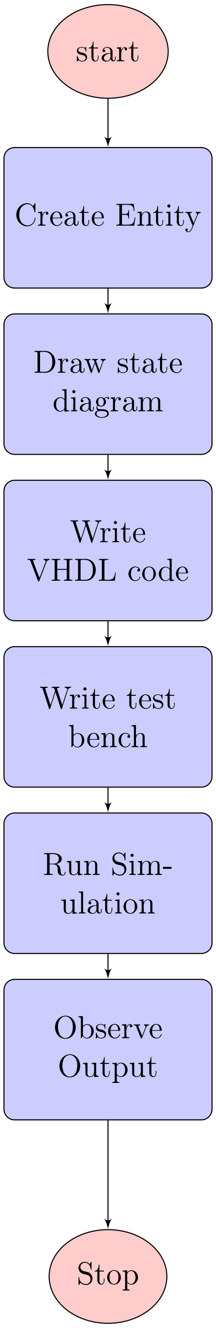


Figure 1: State Diagram

1.3 Flowchart



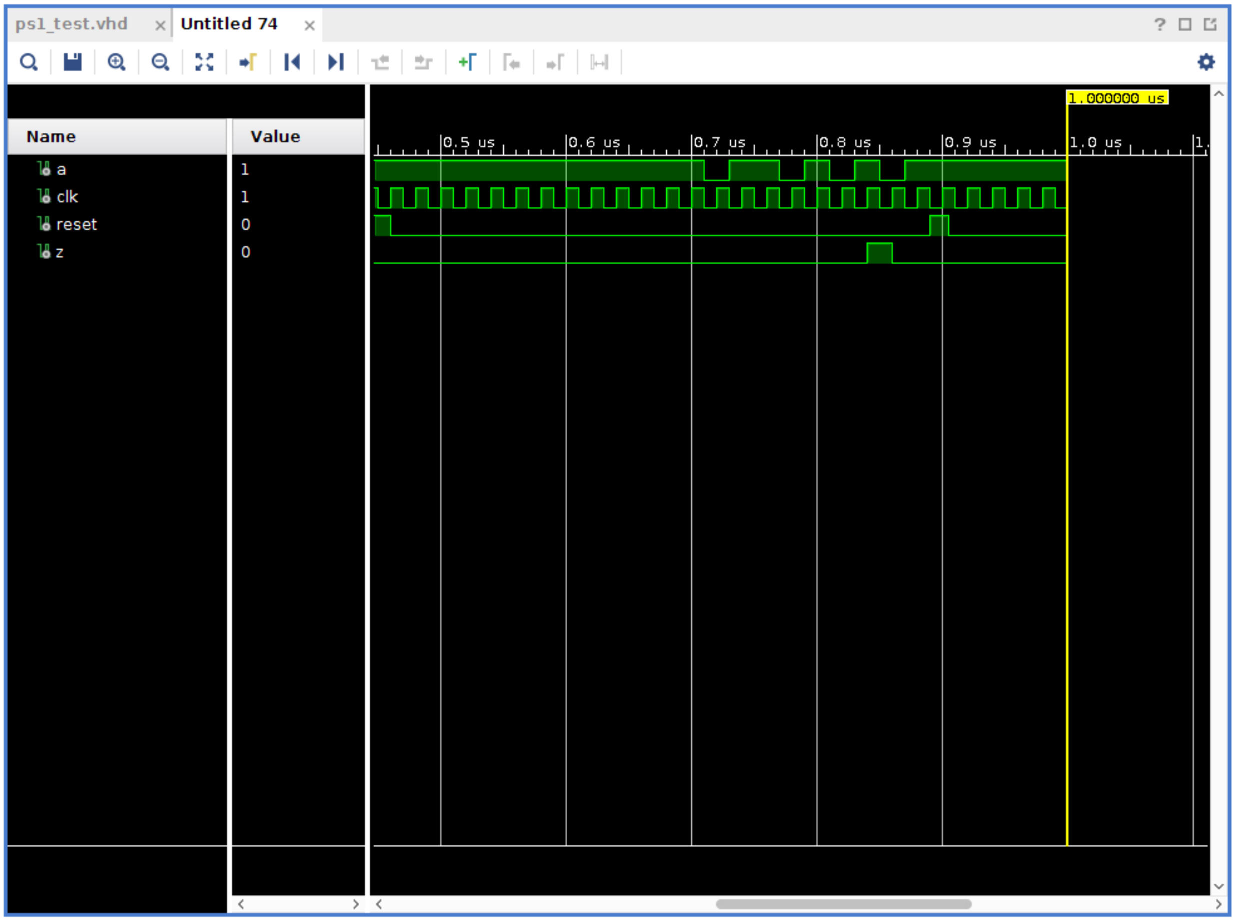
1.4 Inference

Afetr implementing both mealay and moore models, following things can be observed:

- Number of states in Mealy is one more than in Moore.
- Output in mealy depends on present state and input whereas in Moore it depends only on present state.
- Output in Mealy comes a clock cycle prior, compared to Moore.

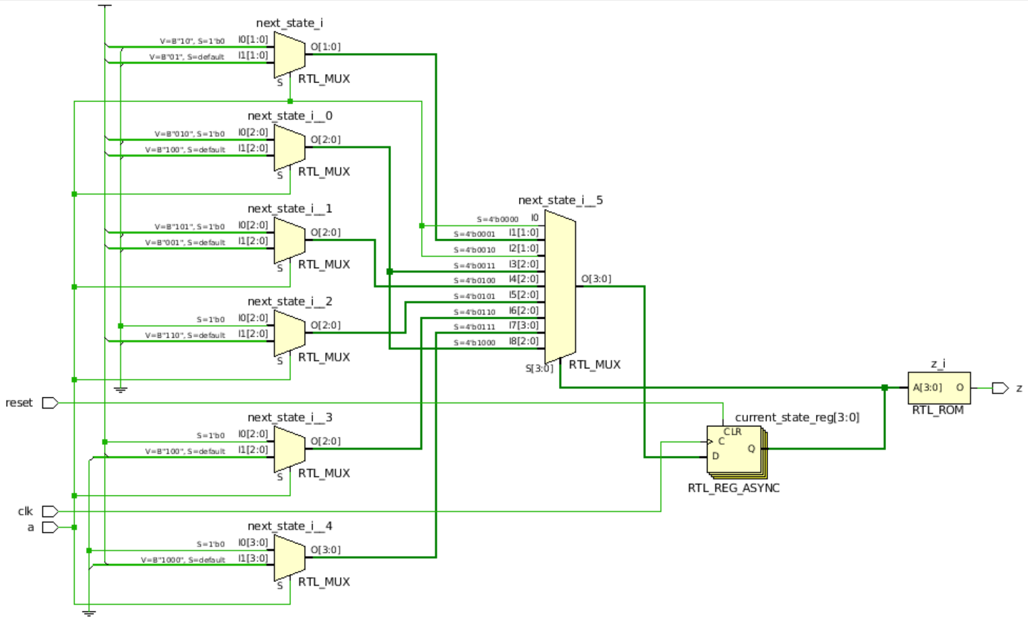
1.5 Screenshots

1.5.1 Simulation



Name	Value
a	1
clk	1
reset	0
z	0

1.5.2 Schematic



2 ProblemStatement-2

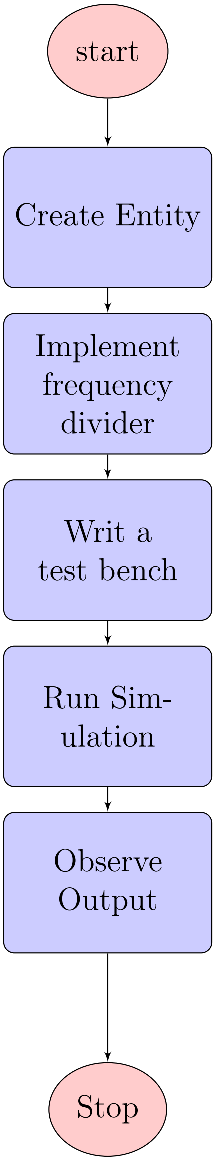
2.1 ProblemSatement

To write VHDL code for frequency divider that divides input frequency by 5.

2.2 Algorithm

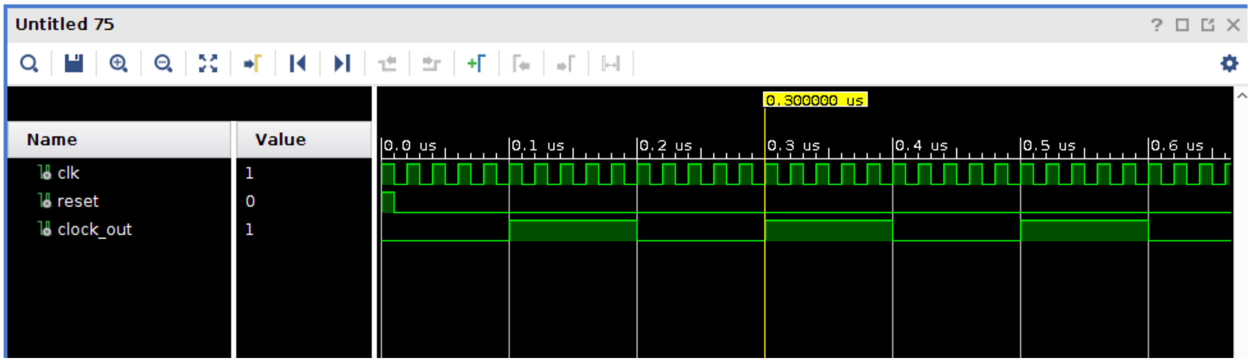
- Create Entity with required input ports and output ports.
- Writ logic in VHDL that divides frequency by 5.
- Implement the logic in Vivado
- Write a test bench with clock and reset
- Generate output clock using testbench
- Run simulation and observe output

2.3 Flowchart



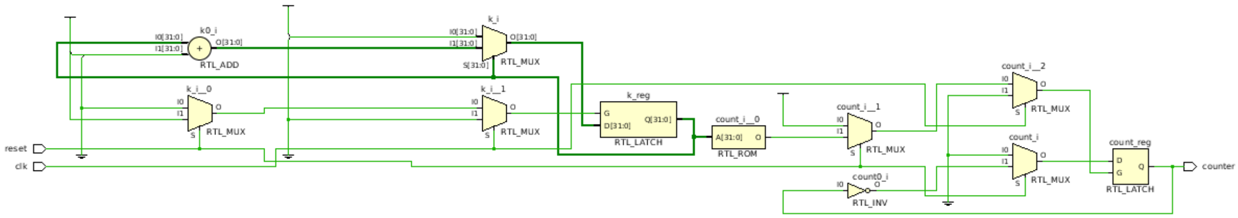
2.4 Screenshots

2.4.1 Simulation



Name	Value
clk	1
reset	0
clock_out	1

2.4.2 Schematic



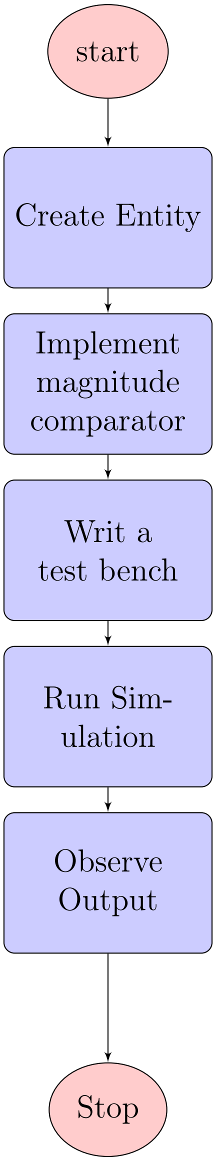
3 ProblemStatement-3

To model and simulate 15bit digital comparator with given requirements.

3.1 Algorithm

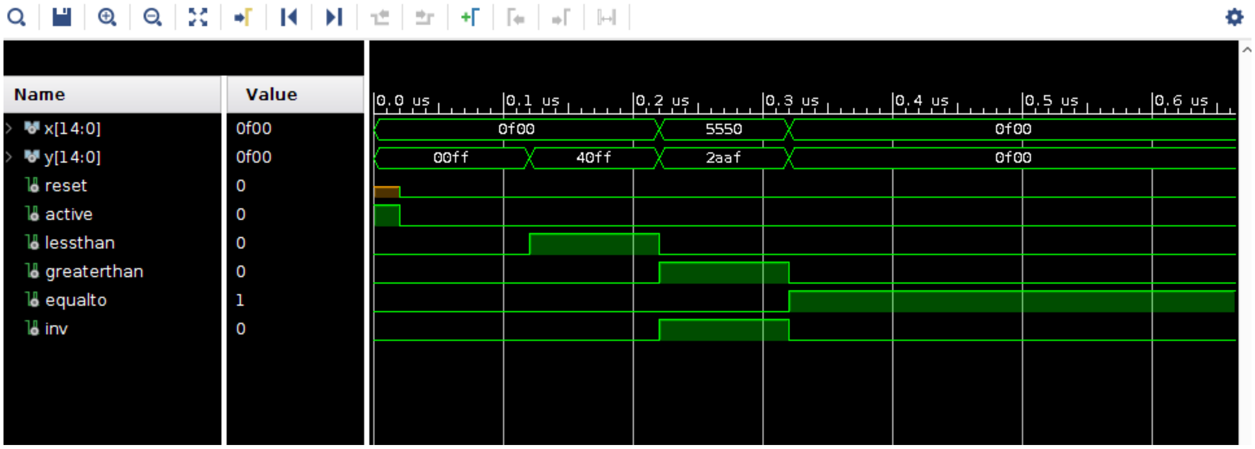
- Create Entity with required input ports and output ports.
- Writ logic that implements magnitude comparator
- Implement the logic in Vivado
- Write a test bench with required inputs
- Run simulation and observe output

3.2 Flowchart



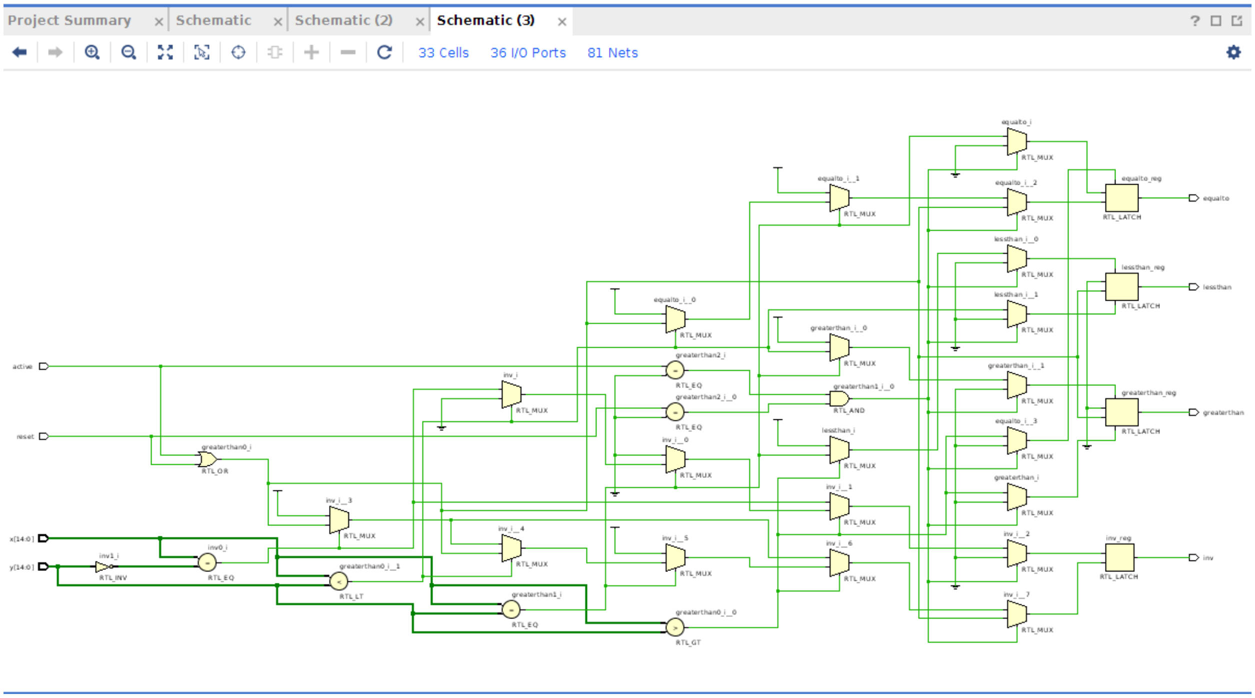
3.3 Screenshots

3.3.1 Simulation



Name	Value
x[14:0]	0f00
y[14:0]	0f00
reset	0
active	0
lessthan	0
greaterthan	0
equalto	1
inv	0

3.3.2 Schematic



Appendices

Problem 1

code:

```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date: 30.10.2019 09:24:23
-- Design Name:
-- Module Name: ps1 - Behavioral
-- Project Name:
-- Target Devices:
-- Tool Versions:
-- Description:
--
-- Dependencies:
--
-- Revision:
-- Revision 0.01 - File Created
-- Additional Comments:
--
----------------------------------------------------------------------------------


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 ps1 is
    Port ( a : in STD_LOGIC;
           clk : in STD_LOGIC;
           reset : in STD_LOGIC;
           z : out STD_LOGIC);
end ps1;

architecture Behavioral of ps1 is
    type state_type is (s0,s1,s2,s3,s4,s5,s6,s7,s8);
    signal next_state, current_state: state_type;

begin
--Process for reset and clock
    process(clk,reset)
    begin
        if(reset = ’1’) then
            current_state <= s0;
        elsif rising_edge(clk) then
            current_state <= next_state;
         end if;
    end process;
    --Process for assigning states in Moore model
    process(current_state,a)
    begin
        case current_state is
        when s0=>
            if a= ’0’ then
                next_state <= s0;
            else
                next_state <= s1;
             end if;
         when s1=>
             if a= ’0’ then
                next_state <= s2;
             else
                 next_state <= s1;
              end if;
          when s2=>
              if a= ’0’ then
                  next_state <= s0;
              else
                  next_state <= s3;
               end if;
          when s3=>
               if a =’0’ then
                   next_state <= s2;
               else
                  next_state <=s4;
                end if;
          when s4=>
                if a ='0' then
                    next_state <= s5;
                else
                    next_state <=s1;
                end if;
          when s5=>
                 if a ='0' then
                    next_state <= s0;
                 else
                    next_state <=s6;
                 end if;
          when s6=>
               if a='0' then
                  next_state <= s7;
               else
                  next_state<=s4;
               end if;
          when s7=>
              if a='0' then
                  next_state<= s0;
              else
                  next_state<= s8;
              end if;
            when s8 =>
               if a='0' then
                  next_state<= s2;
               else
                  next_state<=s4;
                end if;
           end case;
    end process;
    process(current_state)
    begin
    -- Asigning output in moore model
    case current_state is
    when s0=>
        z<='0';
    when s1=>
         z<=’0’;
    when s2=>
          z<=’0’;
    when s3=>
          z<=’0’;
    when s4=>
         z<=’0’;
    when s5=>
         z<=’0’;
    when s6=>
         z<=’0’;
    when s7=>
         z<=’0’;
    when s8=>
          z<=’1’;
    end case;
    end process;
end Behavioral;
```

testbench:

```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date: 30.10.2019 10:05:56
-- Design Name:
-- Module Name: ps1_test - Behavioral
-- Project Name:
-- Target Devices:
-- Tool Versions:
-- Description:
--
-- Dependencies:
--
-- Revision:
-- Revision 0.01 - File Created
-- Additional Comments:
--
----------------------------------------------------------------------------------


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 ps1_test is
--  Port ( );
end ps1_test;

architecture Behavioral of ps1_test is

signal a,clk,reset: std_logic;
    signal z : std_logic;

    component ps1
    port(
        a,clk,reset : in std_logic;
        z: out std_logic
     );
     end component;

     begin
     uut: ps1 port map(a,clk,reset,z);

     process
        begin
            clk <= ’1’;
            wait for 10ns;
            clk <= ’0’;
             wait for 10ns;
      end process;

      process
           begin
              reset<=’1’;
              wait for 15ns;
              reset<=’0’;
              a<=’1’;
              wait for 250ns;

              a<=’0’;
              wait for 20ns;

              a<=’1’;
              wait for 20ns;

              a<=’1’;
              wait for 20ns;

              a<=’0’;
              wait for 20ns;

              a<=’1’;
              wait for 20ns;

              a<=’0’;
              wait for 20ns;

              a<=’1’;
              wait for 20ns;

              a<=’0’;
              wait for 20ns;

              a<=’1’;
              wait for 20ns;
           end process;

end Behavioral;
```

Problem 2

code:

```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
```

```
--
-- Create Date: 30.10.2019 10:39:36
-- Design Name:
-- Module Name: ps2 - Behavioral
-- Project Name:
-- Target Devices:
-- Tool Versions:
-- Description:
--
-- Dependencies:
--
-- Revision:
-- Revision 0.01 - File Created
-- Additional Comments:
--
----------------------------------------------------------------------------------


library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_UNSIGNED.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 ps2 is
    Port ( clk : in STD_LOGIC;
           reset : in STD_LOGIC;
           counter : out STD_LOGIC);
end ps2;

architecture Behavioral of ps2 is

signal count:std_logic;
signal k:integer:=1;
begin
-- Process for clock and reset
```

```
process(clk,reset)
begin
if(clk='1') then
    if(reset='1') then
        count<='0';
    elsif(reset='0') then
        if(k=5) then               -- Terminates if count>5
            k<=1;
            count<=not(count);    -- Negating clock
        else
            k <= k+1;
        end if;
    end if;
end if;
end process;
counter<= count;
end Behavioral;
```

testbench:

```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date: 30.10.2019 10:55:49
-- Design Name:
-- Module Name: ps2_test - Behavioral
-- Project Name:
-- Target Devices:
-- Tool Versions:
-- Description:
--
-- Dependencies:
--
-- Revision:
-- Revision 0.01 - File Created
-- Additional Comments:
--
----------------------------------------------------------------------------------


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 ps2_test is
--  Port ( );
end ps2_test;

architecture Behavioral of ps2_test is
signal clk,reset: std_logic;
    signal clock_out: std_logic;

    component ps2
    port(
    clk,reset: in std_logic;
    counter: out std_logic
    );
    end component;


begin
    uut: ps2 port map(clk,reset,clock_out);

    process
    begin
        clk<='1';
        wait for 10ns;
        clk<='0';
        wait for 10ns;

    end process;

    process
    begin
        reset<='1';
        wait for 10ns;
        reset<='0';
        wait;
    end process;

end Behavioral;
```

Problem 2

code:

```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date: 30.10.2019 12:30:17
-- Design Name:
-- Module Name: ps3 - Behavioral
-- Project Name:
-- Target Devices:
-- Tool Versions:
-- Description:
--
-- Dependencies:
--
-- Revision:
-- Revision 0.01 - File Created
-- Additional Comments:
--
----------------------------------------------------------------------------------


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 ps3 is
    Port ( x : in STD_LOGIC_VECTOR (14 downto 0);
           y : in STD_LOGIC_VECTOR (14 downto 0);
           reset: in std_logic;
           active:in std_logic;
           lessthan : out STD_LOGIC;
           greaterthan : out STD_LOGIC;
           equalto : out STD_LOGIC;
           inv:out STD_LOGIC:='0');
end ps3;

architecture Behavioral of ps3 is

begin
    -- Process to find greater, lesser or equalto
    process(x,y)
    begin
    if(active='1' or reset='1') then
        greaterthan<='0';
        lessthan<='0';
        equalto<='0';
        inv<='0';
    end if;
    if(active='0' and reset='0' ) then
        if(x>y) then
            if(x = not(y)) then            -- Checking invert condition
            inv<='1';
            end if;
            greaterthan<='1';
            lessthan<='0';
            equalto<='0';
        elsif(x=y) then
            greaterthan<='0';
            lessthan<='0';
            equalto<='1';
            inv<='0';
        elsif(x<y) then
            greaterthan<='0';
            lessthan<='1';
            equalto<='0';
            if(x = not(y)) then
                inv<='1';
            end if;
        end if;

     end if;
     end process;


end Behavioral;
```

testbench:

```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date: 30.10.2019 12:41:26
-- Design Name:
-- Module Name: ps3_test - Behavioral
-- Project Name:
-- Target Devices:
-- Tool Versions:
-- Description:
--
-- Dependencies:
--
-- Revision:
-- Revision 0.01 - File Created
-- Additional Comments:
--
----------------------------------------------------------------------------------


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 ps3_test is
--  Port ( );
end ps3_test;

architecture Behavioral of ps3_test is

    component ps3
    port(
        x: in std_logic_vector(14 downto 0);
        y: in std_logic_vector(14 downto 0);
        reset: in std_logic;
        active:in std_logic;
        lessthan: out std_logic;
        greaterthan : out STD_LOGIC;
        equalto : out STD_LOGIC;
        inv:out STD_LOGIC
        );
    end component;

    signal x,y: std_logic_vector(14 downto 0);
    signal reset,active: std_logic;
    signal lessthan,greaterthan,equalto,inv: std_logic;

begin
    UUT: ps3 port map (x,y,reset,active,lessthan,greaterthan,equalto,inv);
    process
    begin
        active <=’1’;
        x <= "000111100000000";
        y <= "000000011111111";
        wait for 20ns;

        active<=’0’;
        reset<=’0’;
        x <= "000111100000000";
        y <= "000000011111111";
        assert (greaterthan=’1’) report "Comparison Error detected!" ;
        wait for 100ns;

        x <= "000111100000000";
        y <= "100000011111111";
        assert (lessthan=’1’) report "Comparison Error detected!" ;
        wait for 100ns;

        x<= "101010101010000";
        y<="010101010101111";
        wait for 100ns;

        x <= "000111100000000";
        y <= "000111100000000";
        assert (equalto=’1’) report "Comparison Error detected!" ;
        wait for 100ns;

        wait;
    end process;

end Behavioral;
```

References

[1] Flowchart using Latex
Kjell Magne Fauske
http://www.texample.net/tikz/examples/simple-flow-chart/

[2] VHDL Basics by Example
http://esd.cs.ucr.edu/labs/tutorial/

[3] Finite State Machines
https://vhdlguide.readthedocs.io/en/latest/vhdl/fsm.html

[4] Git Hub
https://help.github.com/en/articles/fork-a-repo

[5] VHDL Test Bench
https://www.youtube.com/watch?v=2wMj-JmHDQQ