IT 451: Computer Organization and Architecture (Sessional)

Name – Anupam Sanidhya

Enrolment number – 510817049

Roll number – 42 (Hy)

Assignment 2

1.) Design and simulate the behavioral model of an UP/DOWN counter that is capable of counting up to 10. Use one input count_mode to control the mode of counting i.e. for count_mode = ‘1’ the counter will operate as an UP counter and for count_mode = ‘0’ the counter will behave as a DOWN counter. Use a reset input to reset the counter anytime to 0.

VHDL MODULE:

```
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 primitives in this code.

--library UNISIM;

--use UNISIM.VComponents.all;
```

```
entity Ass2Q1 is

    Port ( clk : in  STD_LOGIC;

           count_mode : in  STD_LOGIC;

           reset : in  STD_LOGIC;

           counter : out  STD_LOGIC_VECTOR(3 downto 0));

end Ass2Q1;



architecture Behavioral of Ass2Q1 is

signal c:std_logic_vector(3 downto 0);



begin

process(clk,reset)

begin

if(rising_edge(clk))

 then

  if(reset='1')

   then

               c<="0000";

              else

    if(count_mode='1')

     then

      if(c="1010")

       then

        c<="0000";

                              else

        c<=c+x"1";

      end if;

    else

     if(c="0000")

      then

       c<="1010";

                             else

       c<=c-x"1";

                end if;

    end if;

  end if;

end if;

end process;
```

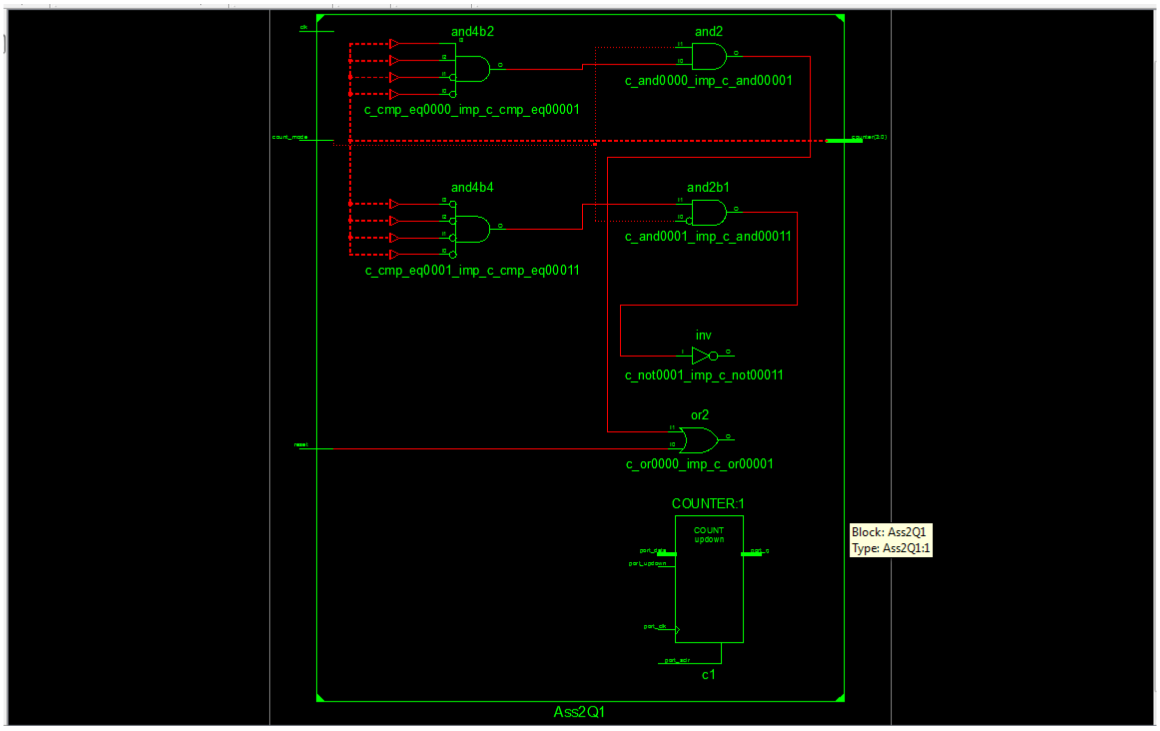
```
counter<=c;

end Behavioral;
```

TEST BENCH :

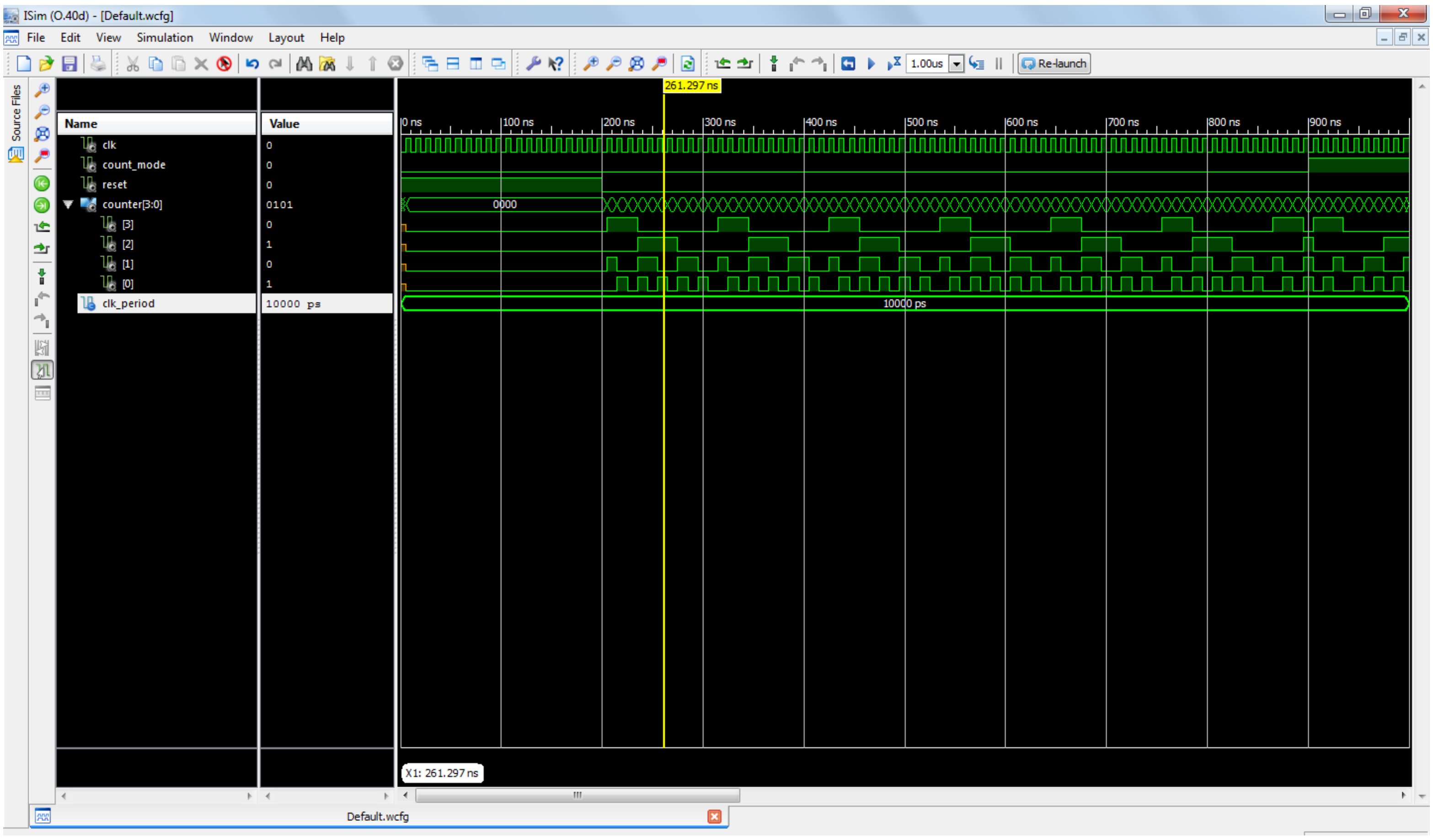
```
LIBRARY ieee;

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;


ENTITY test IS

END test;


ARCHITECTURE behavior OF test IS


    -- Component Declaration for the Unit Under Test (UUT)


    COMPONENT Ass2Q1

    PORT(

         clk : IN  std_logic;

         count_mode : IN  std_logic;

         reset : IN  std_logic;

         counter : OUT  std_logic_vector(3 downto 0)

        );

    END COMPONENT;


   --Inputs

   signal clk : std_logic := '0';

   signal count_mode : std_logic := '0';

   signal reset : std_logic := '0';


              --Outputs

   signal counter : std_logic_vector(3 downto 0);


   -- Clock period definitions

   constant clk_period : time := 10 ns;
```

```
BEGIN

             -- Instantiate the Unit Under Test (UUT)

   uut: Ass2Q1 PORT MAP (

          clk => clk,

          count_mode => count_mode,

          reset => reset,

          counter => counter

        );

   -- Clock process definitions

   clk_process :process

   begin

                          clk <= '0';

                          wait for clk_period/2;

                          clk <= '1';

                          wait for clk_period/2;

   end process;

   -- Stimulus process

   stim_proc: process

   begin

      -- hold reset state for 100 ns.

      reset<='1';

                          wait for 200 ns;

                          reset<='0';

                          wait for 300 ns;

                          count_mode<='0';

                          wait for 400 ns;

                          reset<='0';

                          count_mode<='1';

      wait for clk_period*10;

      -- insert stimulus here


      wait;

   end process;

END;
```

RTL Schematic –



Simulation –



Name	Value
clk	0
count_mode	0
reset	0
counter[3:0]	0101
[3]	0
[2]	1
[1]	0
[0]	1
clk_period	10000 ps

2.) Design and simulate the behavioral model of a 4 bit shift register that can support the following modes of data transfer.

a. Serial in parallel out (SIPO)

b. Serial in serial out (SISO)

c. Parallel in serial out (PISO)

d. Parallel in parallel out (PIPO).

Use case statement for this design. Use one input reg_mode [std_logic_vector (1 downto 0)] to control the nature of behavior of the register. Use "00" for SIPO, "01" for SISO and so on.

VHDL MODULE:

Mux:

```
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 primitives in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity moxx is
   Port ( i0,i1 : in  STD_LOGIC;
          s : in  STD_LOGIC;
          o : out  STD_LOGIC);
end moxx;

architecture Behavioral of moxx is

begin
o<= i0 when s='0' else
 i1 when s='1';

end Behavioral;
```

Barrel Register:

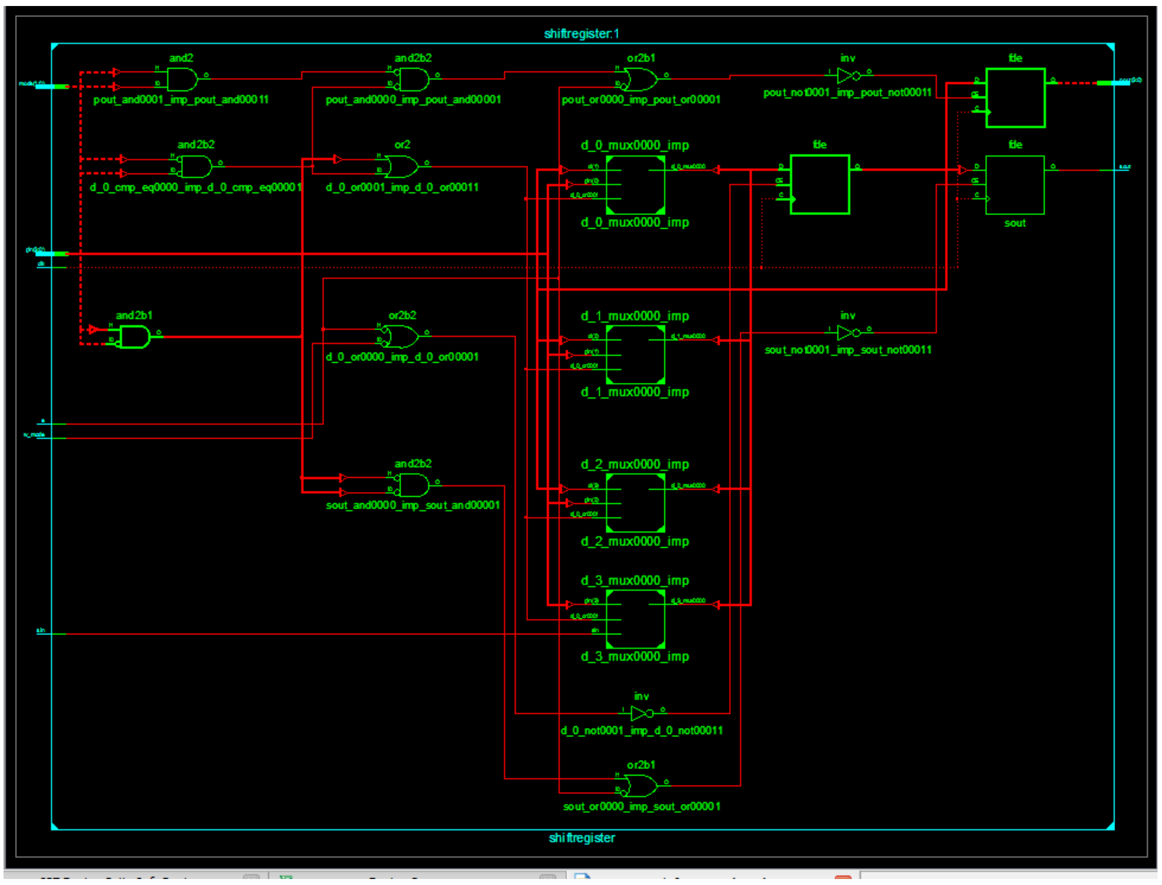
```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

-- 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 primitives in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity shiftregister is
    Port ( mode : in  STD_LOGIC_VECTOR (1 downto 0);
           sout : out  STD_LOGIC;
           sin,clk,e,w_mode : in  STD_LOGIC;
           pout : out  STD_LOGIC_VECTOR (3 downto 0);
           pin : in  STD_LOGIC_VECTOR (3 downto 0));
end shiftregister;

architecture Behavioral of shiftregister is
signal d:std_logic_vector(3 downto 0):="0000";
begin
process(e,clk)
begin
if(e='1' and clk'event and clk='1')then
if(w_mode='1')then
case mode is
when "00" | "01" =>
d(0)<=d(1);
d(1)<=d(2);
d(2)<=d(3);
d(3)<=sin;
when "10" | "11" =>
d<=pin;
when others =>
end case;
end if;
case mode is
when "01"| "10" =>
sout <= d(0);
when "00" | "11" =>
pout<=d;
when others =>
end case;
end if;
end process;
end Behavioral;
```

TEST BENCH :

```
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;

ENTITY test IS
END test;

ARCHITECTURE behavior OF test IS

    -- Component Declaration for the Unit Under Test (UUT)

    COMPONENT shiftregister
    PORT(
         mode : IN  std_logic_vector(1 downto 0);
         sout : OUT  std_logic;
         sin : IN  std_logic;
         clk : IN  std_logic;
         e : IN  std_logic;
         w_mode : IN  std_logic;
         pout : OUT  std_logic_vector(3 downto 0);
         pin : IN  std_logic_vector(3 downto 0)
```

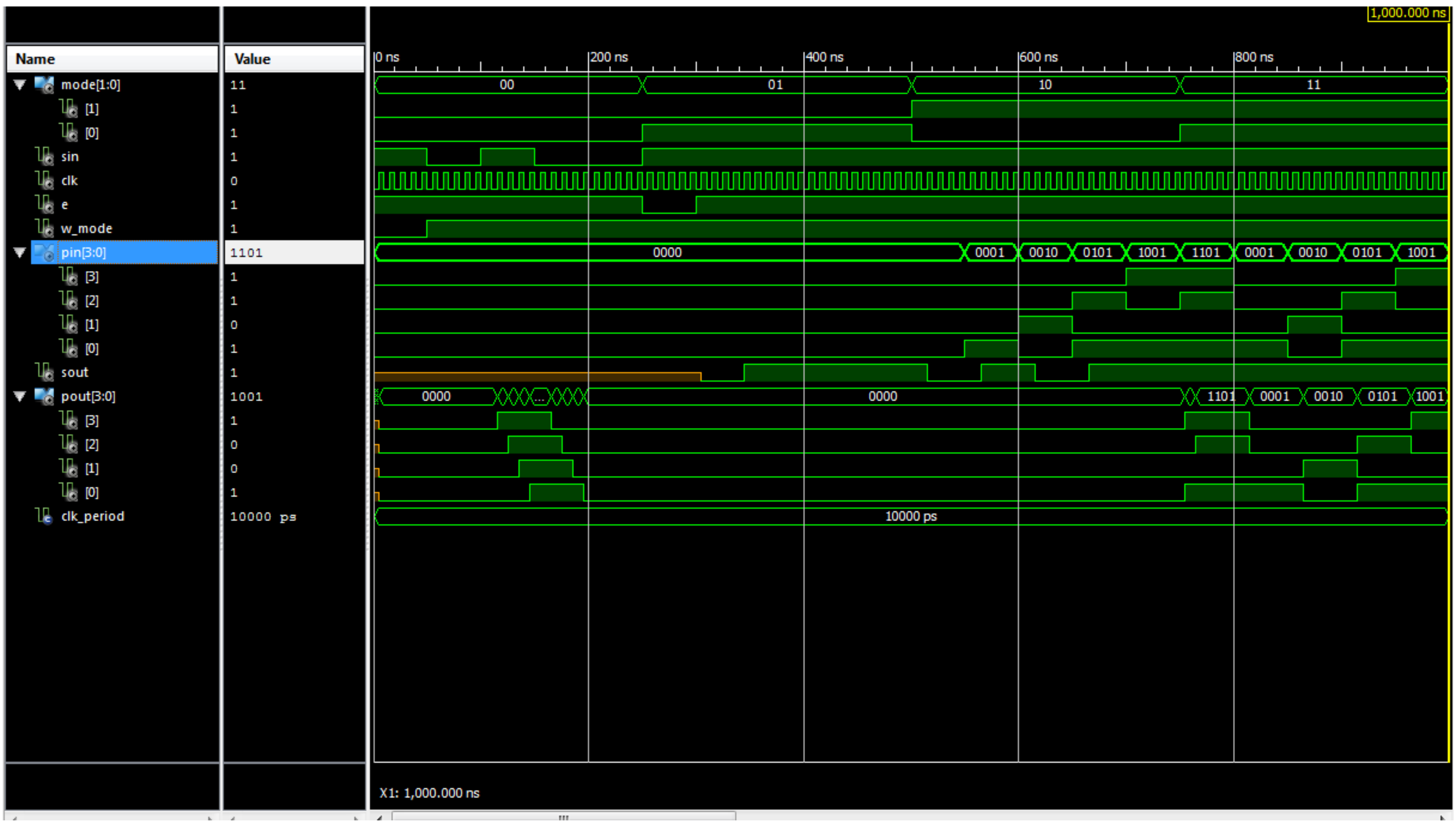
```
        );
    END COMPONENT;



   --Inputs
   signal mode : std_logic_vector(1 downto 0) := (others => '0');
   signal sin : std_logic := '0';
   signal clk : std_logic := '0';
   signal e : std_logic := '0';
   signal w_mode : std_logic := '0';
   signal pin : std_logic_vector(3 downto 0) := (others => '0');

  --Outputs
   signal sout : std_logic;
   signal pout : std_logic_vector(3 downto 0);

   -- Clock period definitions
   constant clk_period : time := 10 ns;

BEGIN

-- Instantiate the Unit Under Test (UUT)
   uut: shiftregister PORT MAP (
          mode => mode,
          sout => sout,
          sin => sin,
          clk => clk,
          e => e,
          w_mode => w_mode,
          pout => pout,
          pin => pin
        );

   -- Clock process definitions
   clk_process :process
   begin
clk <= '0';
wait for clk_period/2;
clk <= '1';
wait for clk_period/2;
   end process;



   -- Stimulus process
   stim_proc: process
   begin
      mode <= "00", "01" after 250 ns, "10" after 500 ns, "11" after 750 ns;
e <= '1', '0' after 250 ns, '1' after 300 ns;
w_mode <= '0', '1' after 50 ns;
sin <= '1', '0' after 50 ns, '1' after 100 ns, '0' after 150 ns, '1' after 250 ns;
pin <= "0000","0001" after 550 ns,"0010" after 600 ns,"0101" after 650 ns,"1001" after 700 ns,"1101" after 750 ns,
"0001" after 800 ns,"0010" after 850 ns,"0101" after 900 ns,"1001" after 950 ns,"1101" after 1000 ns;
      -- insert stimulus here

      wait;
   end process;

END;
```

RTL Schematic –



shiftregister:1
and2
pout_and0001_imp_pout_and00011
and2b2
pout_and0000_imp_pout_and00001
or2b1
pout_or0000_imp_pout_or00001
inv
pout_not0001_imp_pout_not00011
fde
and2b2
d_0_cmp_eq0000_imp_d_0_cmp_eq00001
or2
d_0_or0001_imp_d_0_or00011
d_0_mux0000_imp
fde
fde
sout
and2b1
or2b2
d_0_or0000_imp_d_0_or00001
d_1_mux0000_imp
inv
sout_not0001_imp_sout_not00011
and2b2
sout_and0000_imp_sout_and00001
d_2_mux0000_imp
d_3_mux0000_imp
inv
d_0_not0001_imp_d_0_not00011
or2b1
sout_or0000_imp_sout_or00001
shiftregister

Simulation –



3. Design the followings:

a.) Design and simulate the behavioral model of an 8 bit even parity generator.

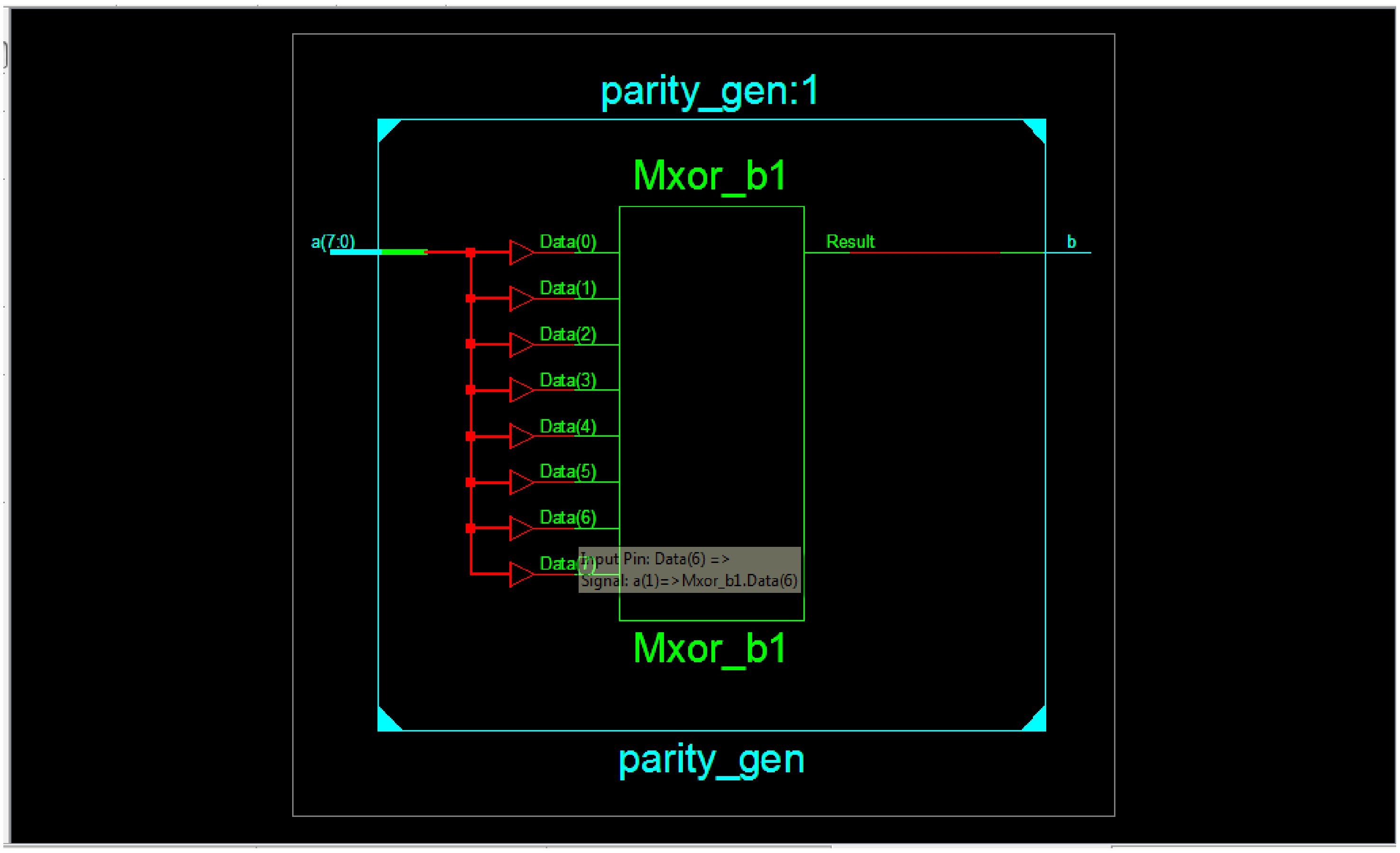
VHDL MODULE:

```
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 primitives in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity parity_gen is
   Port ( a : in  STD_LOGIC_VECTOR (7 downto 0);
          b : out  STD_LOGIC);
end parity_gen;

architecture Behavioral of parity_gen is

begin

process(a)
begin
b <= a(0) xor a(1) xor a(2) xor a(3) xor a(4) xor a(5) xor a(6) xor a(7);
end process;

end Behavioral;
```

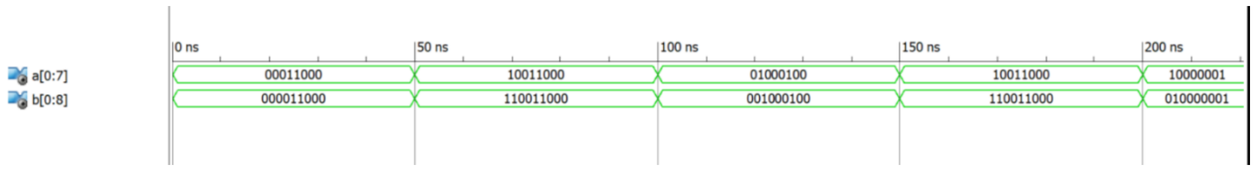
TEST BENCH :

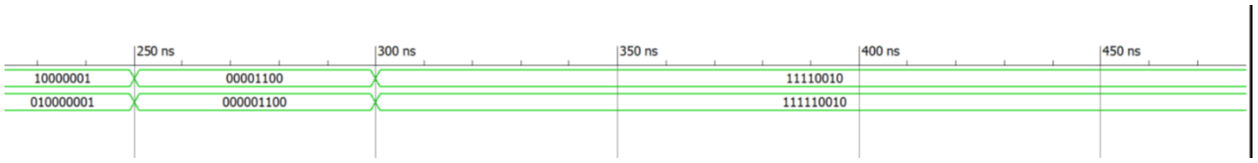
```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
ENTITY TEST IS
END TEST;
ARCHITECTURE behavior OF TEST IS
 COMPONENT parity_gen
   PORT(
      a: IN  std_logic_vector(0 to 7);
      b : OUT  std_logic_vector(0 to 8)
     );
   END COMPONENT;
signal a: std_logic_vector(0 to 7) := (others => '0');
signal b : std_logic_vector(0 to 8);
BEGIN
 uut: parity_gen PORT MAP (
        a => a,
        b =>b
      );
  stim_proc: process
  begin

 a<="00011000";
 wait for 50 ns;
 a<="10011000";
 wait for 50 ns;
 a<="01000100";
 wait for 50 ns;
 a<="10011000";
 wait for 50 ns;
 a<="10000001";
 wait for 50 ns;
 a<="00001100";
 wait for 50 ns;
 a<="11110010";
 wait for 50 ns;

     wait;
  end process;

END;
```

RTL Schematic –



Simulation –



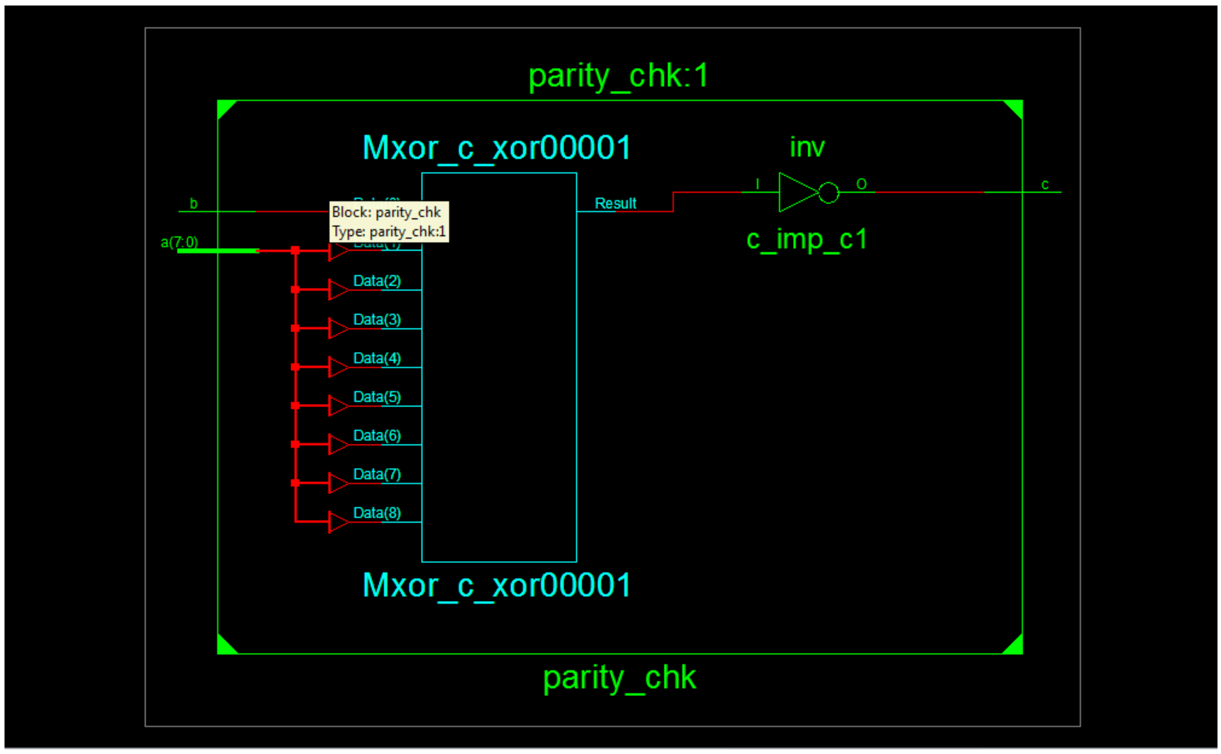


b.) Design and simulate the behavioral model an 8 bit even parity checker.

VHDL MODULE:

```
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 primitives in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity parity_chk is
   Port ( a : in  STD_LOGIC_VECTOR (7 downto 0);
          b : in  STD_LOGIC;
          c : out  STD_LOGIC);
end parity_chk;

architecture Behavioral of parity_chk is

begin

c <= not (a(0) xor a(1) xor a(2) xor a(3) xor a(4) xor a(5) xor a(6) xor a(7) xor b);

end Behavioral;
```

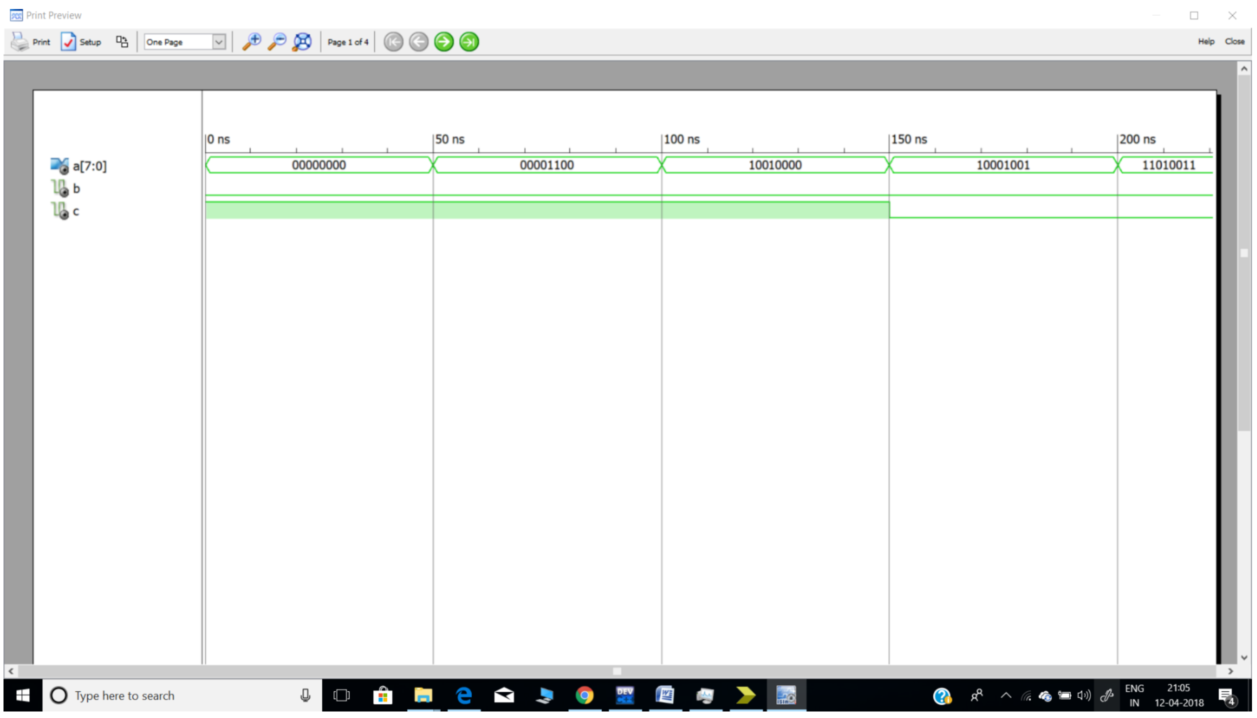
RTL Schematic –



TEST BENCH :

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
ENTITY s1_ttb IS
END s1_ttb;

ARCHITECTURE behavior OF s1_ttb IS
COMPONENT parity_chk
   PORT(
        a : IN  std_logic_vector(7 downto 0);
        b : IN  std_logic;
        c : OUT  std_logic
       );
   END COMPONENT;


  signal a : std_logic_vector(7 downto 0) := (others => '0');
  signal b : std_logic := '0';

  signal c : std_logic;

BEGIN

  uut: parity_chk PORT MAP (
        a => a,
        b => b,
        c => c
      );

  stim_proc: process
  begin

  a<="00000000";
wait for 50 ns;
a <="00001100";
wait for 50 ns;
a <="10010000";
wait for 50 ns;
a <="10001001";
wait for 50 ns;
a <="11010011";
wait for 50 ns;
a <="00001000";
wait for 50 ns;
a<="11011000";
wait for 50 ns;



    -- insert stimulus here

    wait;
  end process;

END;
```

Simulation –

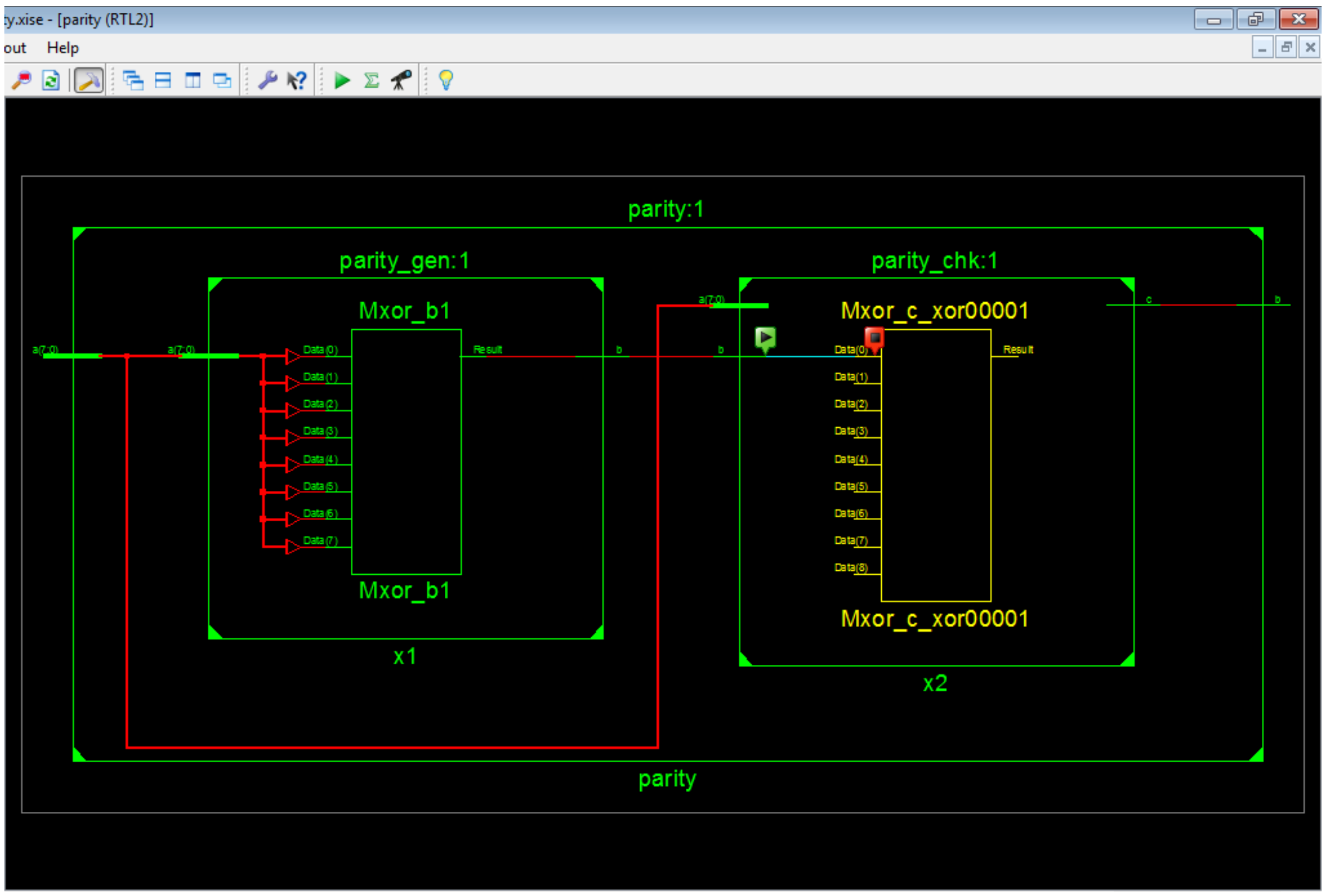


c.) Design a top level module called parity and use the previously designed two modules as components to check whether the functionality of the two modules are in synchronization or not.

VHDL MODULE:

```
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 primitives in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity parity is
   Port ( a : in  STD_LOGIC_VECTOR (7 downto 0);
          b : out  STD_LOGIC);
end parity;

architecture Structural of parity is

component parity_gen is
   Port ( a : in  STD_LOGIC_VECTOR (7 downto 0);
          b : out  STD_LOGIC);
end component;
component parity_chk is
   Port ( a : in  STD_LOGIC_VECTOR (7 downto 0);
          b : in  STD_LOGIC;
          c : out  STD_LOGIC);
end component;
signal t1 : std_logic := '0';

begin

x1 : parity_gen port map(a,t1);
x2 : parity_chk port map(a,t1,b);

end Structural;
```

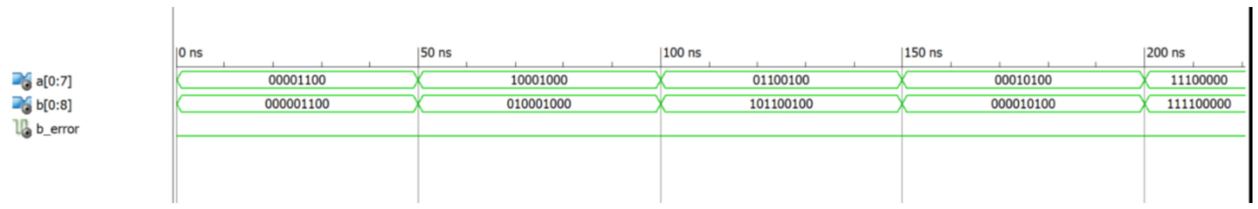
RTL Schematic –

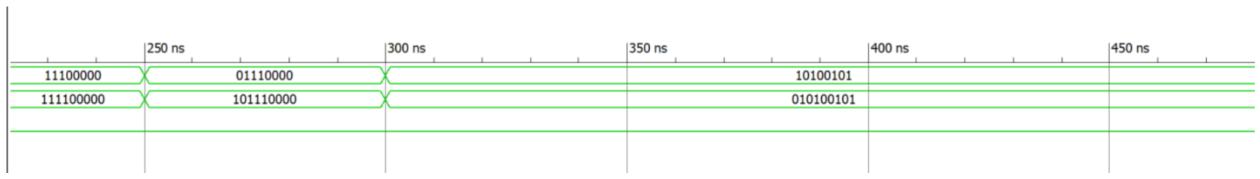


TEST BENCH :

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;

ENTITY TEST IS
END TEST;

ARCHITECTURE behavior OF TEST IS

   COMPONENT parity
   PORT(
      a : IN  std_logic_vector(0 to 7);
      b: OUT  std_logic_vector(0 to 8);
       b_error: OUT  std_logic
      );
   END COMPONENT;

  signal a: std_logic_vector(0 to 7) := (others => '0');

  signal b : std_logic_vector(0 to 8);
  signal b_error : std_logic;
BEGIN

  uut: parity PORT MAP (
        a=> a,
        b => b,
        b_error => b_error
      );
  stim_proc: process
  begin
    a<="00001100";
wait for 50 ns;
a<="10001000";
wait for 50 ns;
a<="01100100";
wait for 50 ns;
a <="00010100";
wait for 50 ns;
a <="11100000";
wait for 50 ns;
a <="01110000";
wait for 50 ns;
a <="10100101";
wait for 50 ns;
     wait;
  end process;

END;
```

Simulation –





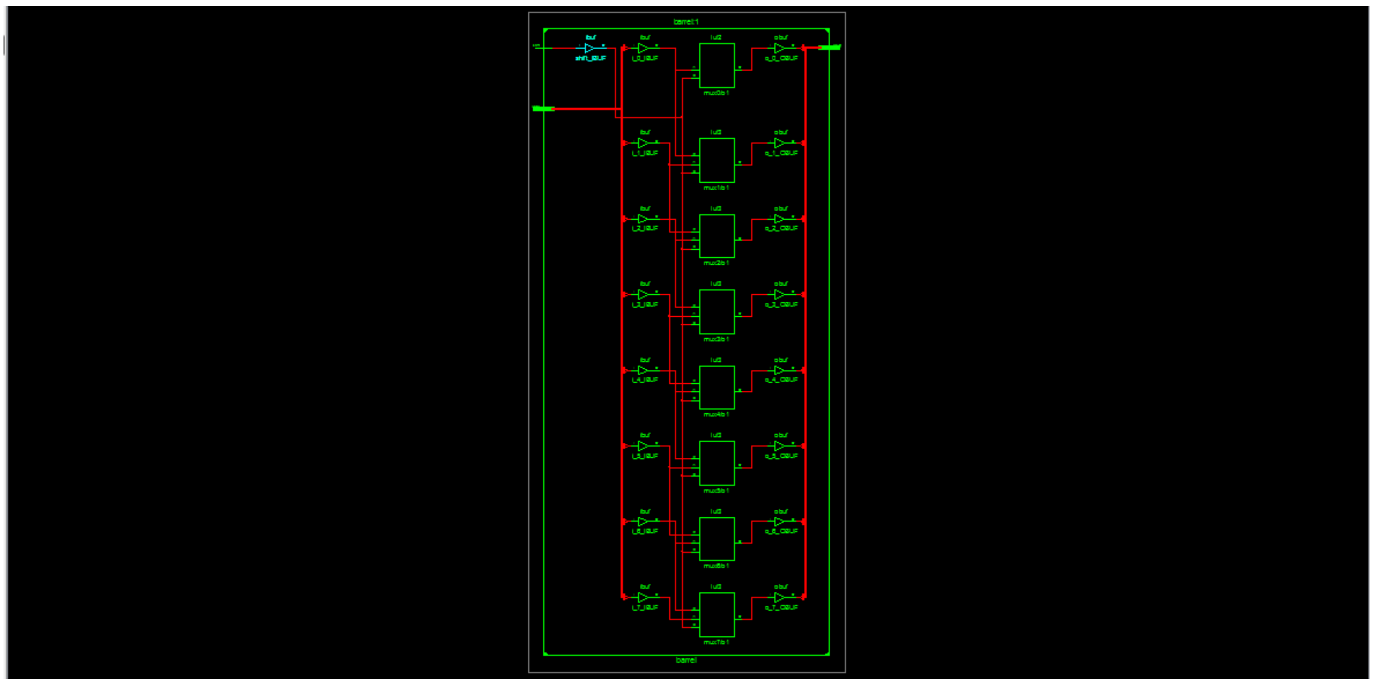
4.)The diagram of a barrel shifter is shown below: The circuit must shift the input vector of size 8 either 0 or 1 position to the left. When actually shifted (shift = 1), the LSB bit must be filled with ‘0’. If shift = 0, then out=in; else, if shift = 1, then out(0)=’0’ and out(i)=in(i-1), for 1≤i≤7. Write a concurrent code for this circuit and verify its behavior.

VHDL MODULE:

```
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 primitives in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity barrel is
   Port ( i : in  STD_LOGIC_VECTOR (7 downto 0);
          shift : in  STD_LOGIC;
          o : out  STD_LOGIC_VECTOR (7 downto 0));
end barrel;

architecture Behavioral of barrel is

begin
mux0 : entity work.moxx port map(i0=>'0',i1=>i(0),s=>shift,o=>o(0));
mux1 : entity work.moxx port map(i0=>i(0),i1=>i(1),s=>shift,o=>o(1));
mux2 : entity work.moxx port map(i0=>i(1),i1=>i(2),s=>shift,o=>o(2));
mux3 : entity work.moxx port map(i0=>i(2),i1=>i(3),s=>shift,o=>o(3));
mux4 : entity work.moxx port map(i0=>i(3),i1=>i(4),s=>shift,o=>o(4));
mux5 : entity work.moxx port map(i0=>i(4),i1=>i(5),s=>shift,o=>o(5));
mux6 : entity work.moxx port map(i0=>i(5),i1=>i(6),s=>shift,o=>o(6));
mux7 : entity work.moxx port map(i0=>i(6),i1=>i(7),s=>shift,o=>o(7));
end Behavioral;
```

RTL Schematic –

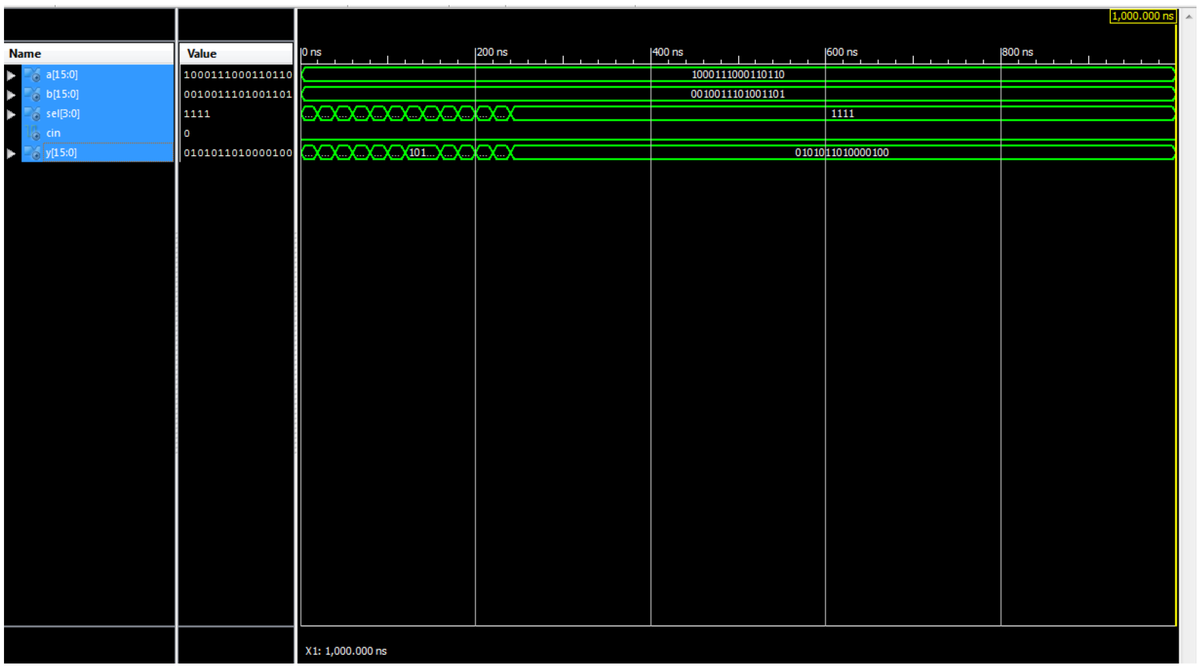


TEST BENCH :

```
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;

ENTITY test IS
END test;

ARCHITECTURE behavior OF test IS

    -- Component Declaration for the Unit Under Test (UUT)

    COMPONENT ALU_m
    PORT(
         a : IN  std_logic_vector(15 downto 0);
         b : IN  std_logic_vector(15 downto 0);
         sel : IN  std_logic_vector(3 downto 0);
         cin : IN  std_logic;
         y : OUT  std_logic_vector(15 downto 0)
        );
    END COMPONENT;


   --Inputs
   signal a : std_logic_vector(15 downto 0) := (others => '0');
   signal b : std_logic_vector(15 downto 0) := (others => '0');
   signal sel : std_logic_vector(3 downto 0) := (others => '0');
   signal cin : std_logic := '0';

 --Outputs
   signal y : std_logic_vector(15 downto 0);
   -- No clocks detected in port list. Replace <clock> below with
   -- appropriate port name

   --constant <clock>_period : time := 10 ns;

BEGIN

-- Instantiate the Unit Under Test (UUT)
   uut: ALU_m PORT MAP (
          a => a,
          b => b,
          sel => sel,
          cin => cin,
          y => y
        );

   -- Clock process definitions
   --<clock>_process :process
   --begin
--<clock><= '0';
--wait for <clock>_period/2;
--<clock><= '1';
--wait for <clock>_period/2;
   --end process;


   -- Stimulus process
   stim_proc: process
   begin
      -- hold reset state for 100 ns.
      --wait for 100 ns;

      --wait for <clock>_period*10;

      -- insert stimulus here
a<="1000111000110110";
b<="0010011101001101";
cin<='0';
sel<="0000";
wait for 20 ns;
```

```
sel<="0001";
wait for 20 ns;
sel<="0010";
wait for 20 ns;
sel<="0011";
wait for 20 ns;
sel<="0100";
wait for 20 ns;
sel<="0101";
wait for 20 ns;
sel<="0110";
wait for 20 ns;
    sel<="0111";
wait for 20 ns;
sel<="1000";
wait for 20 ns;
sel<="1001";
wait for 20 ns;
sel<="1010";
wait for 20 ns;
    sel<="1011";
wait for 20 ns;
sel<="1111";
wait for 20 ns;
    wait;
  end process;

END;
```

Simulation –



Name	Value
a[15:0]	1000111000110110
b[15:0]	0010011101001101
sel[3:0]	1111
cin	0
y[15:0]	0101011010000100

5.) Design an 16 bit ALU (Arithmetic Logic Unit) as shown in the following figure and verify it's behavior with simulation.

The arithmetic and logical operations that are supported are listed below:

Unit	Function	Sel
Arithmetic	Transfer a	0000
	Increment a	0001
	Decrement a	0010
	Transfer b	0011
	Increment b	0100
	Decrement b	0101
	Add a and b	0110
	Add a and b with carry	0111
Logical	Complement a	1000
	Complement b	1001
	AND	1010
	OR	1011
	NAND	1100
	NOR	1101
	XOR	1110
	XNOR	1111

VHDL MODULE:

MUX:

```
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 primitives in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity mux is
   Port ( i0 : in  STD_LOGIC_VECTOR (15 downto 0);
          i1 : in  STD_LOGIC_VECTOR (15 downto 0);
          y : out  STD_LOGIC_VECTOR (15 downto 0);
          sel : in  STD_LOGIC_VECTOR (3 downto 0));
end mux;

architecture Behavioral of mux is
```

```
begin
y<= i0 when sel(3) = '0' else
i1 when sel(3) = '1' ;

end Behavioral;
```

Logic Unit:

```
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;
use ieee.std_logic_unsigned.all;
-- Uncomment the following library declaration if instantiating
-- any Xilinx primitives in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity logic_unit is
   Port ( sel : in  STD_LOGIC_VECTOR (3 downto 0);
          a : in  STD_LOGIC_VECTOR (15 downto 0);
          b : in  STD_LOGIC_VECTOR (15 downto 0);
          o : out  STD_LOGIC_VECTOR (15 downto 0));
end logic_unit;

architecture Behavioral of logic_unit is

begin
o<= not a when sel= "1000" else
not b when sel= "1001" else
a and b when sel="1010" else
a or b when sel="1011" else
a nand b when sel="1100" else
a nor b when sel="1101" else
a xor b when sel="1110" else
a xnor b when sel="1111";

end Behavioral;
```

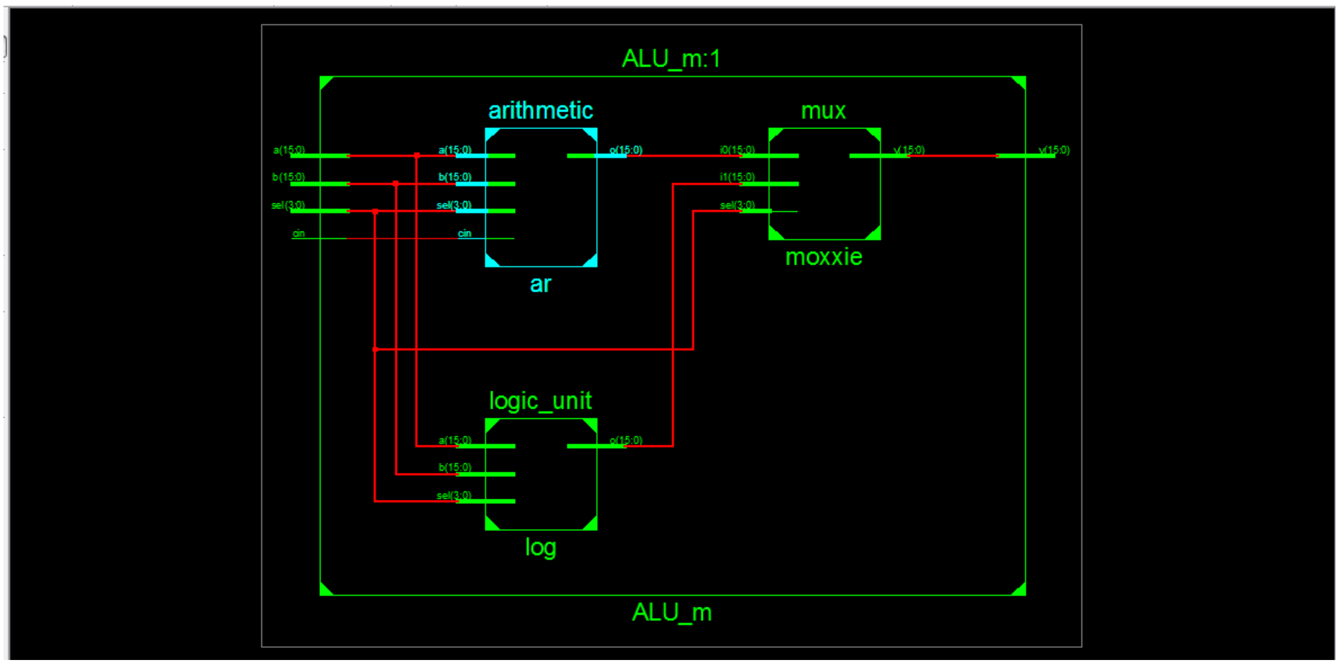
Arithmetic Unit:

```
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;
use ieee.std_logic_unsigned.all;
-- Uncomment the following library declaration if instantiating
-- any Xilinx primitives in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity arithmetic is
   Port ( a : in  STD_LOGIC_VECTOR (15 downto 0);
          b : in  STD_LOGIC_VECTOR (15 downto 0);
          sel : in  STD_LOGIC_VECTOR (3 downto 0);
          o : out  STD_LOGIC_VECTOR (15 downto 0);
  cin: in STD_LOGIC);
end arithmetic;

architecture Behavioral of arithmetic is

begin
o<= a when sel= "0000" else
a+1 when sel= "0001" else
a-1 when sel="0010" else
b when sel="0011" else
b+1 when sel="0100" else
b-1 when sel="0101" else
a+b when sel="0110" else
a + b + cin when sel="0111";

end Behavioral;
```

ALU:

```
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;
use ieee.std_logic_unsigned.all;
-- Uncomment the following library declaration if instantiating
-- any Xilinx primitives in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity ALU_m is
   Port ( a : in  STD_LOGIC_VECTOR (15 downto 0);
          b : in  STD_LOGIC_VECTOR (15 downto 0);
          sel : in  STD_LOGIC_VECTOR (3 downto 0);
          cin : in  STD_LOGIC;
          y : out  STD_LOGIC_VECTOR (15 downto 0));
end ALU_m;

architecture Behavioral of ALU_m is
signal o1 : std_logic_vector(15 downto 0);
signal o2 : std_logic_vector(15 downto 0);
begin
ar : entity work.arithmetic port map(
a=>a,
b=>b,
sel=> sel,
cin=>cin,
o=>o1);
log : entity work.logic_unit port map(
a=>a,
b=>b,
sel=>sel,
--cin=>cin,
o=>o2);
moxxie : entity work.mux port map(
i0=> o1,
i1=> o2,
sel=>sel,
y=>y);
end Behavioral;
```

RTL Schematic –



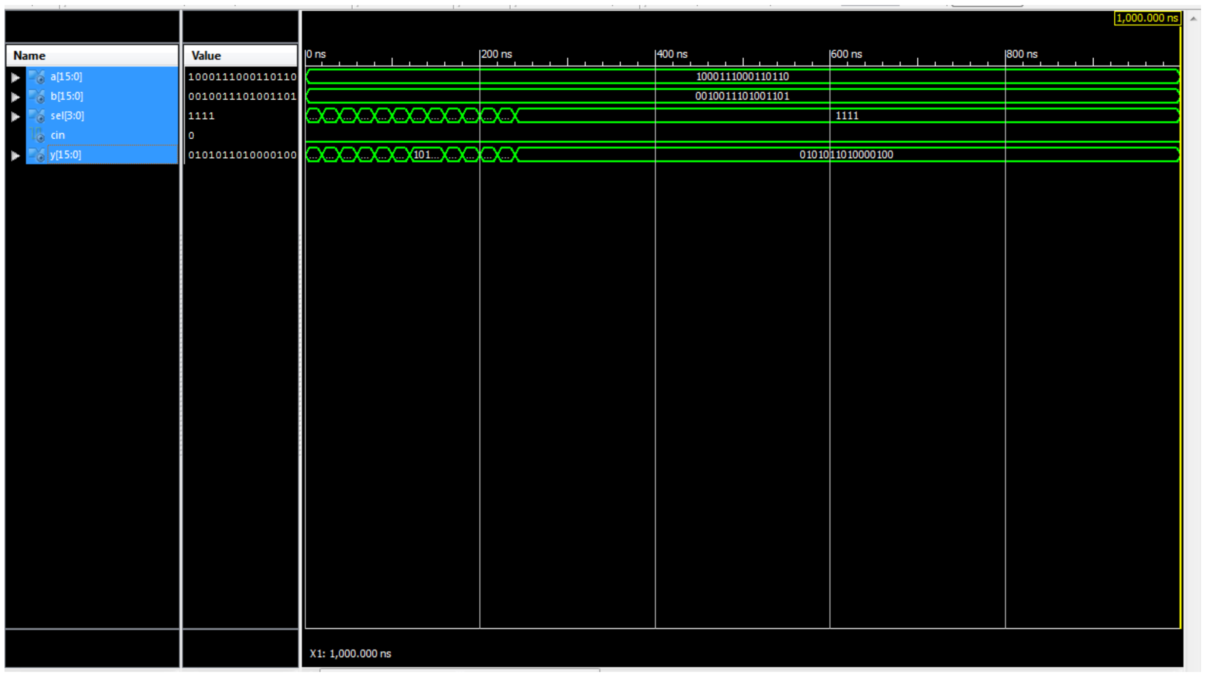
TEST BENCH :

```
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;

ENTITY test IS
END test;

ARCHITECTURE behavior OF test IS

    -- Component Declaration for the Unit Under Test (UUT)

    COMPONENT ALU_m
    PORT(
         a : IN  std_logic_vector(15 downto 0);
         b : IN  std_logic_vector(15 downto 0);
         sel : IN  std_logic_vector(3 downto 0);
         cin : IN  std_logic;
         y : OUT  std_logic_vector(15 downto 0)
        );
    END COMPONENT;
```

```
  --Inputs
  signal a : std_logic_vector(15 downto 0) := (others => '0');
  signal b : std_logic_vector(15 downto 0) := (others => '0');
  signal sel : std_logic_vector(3 downto 0) := (others => '0');
  signal cin : std_logic := '0';

 --Outputs
  signal y : std_logic_vector(15 downto 0);
  -- No clocks detected in port list. Replace <clock> below with
  -- appropriate port name

  --constant <clock>_period : time := 10 ns;

BEGIN

-- Instantiate the Unit Under Test (UUT)
  uut: ALU_m PORT MAP (
        a => a,
        b => b,
        sel => sel,
        cin => cin,
        y => y
      );

  -- Clock process definitions
  --<clock>_process :process
  --begin
--<clock><= '0';
--wait for <clock>_period/2;
--<clock><= '1';
--wait for <clock>_period/2;
  --end process;


  -- Stimulus process
  stim_proc: process
  begin
    -- hold reset state for 100 ns.
    --wait for 100 ns;

    --wait for <clock>_period*10;

    -- insert stimulus here
a<="1000111000110110";
b<="0010011101001101";
cin<='0';
sel<="0000";
wait for 20 ns;
sel<="0001";
wait for 20 ns;
sel<="0010";
wait for 20 ns;
sel<="0011";
wait for 20 ns;
sel<="0100";
wait for 20 ns;
```

```
sel<="0101";
wait for 20 ns;
sel<="0110";
wait for 20 ns;
    sel<="0111";
wait for 20 ns;
sel<="1000";
wait for 20 ns;
sel<="1001";
wait for 20 ns;
sel<="1010";
wait for 20 ns;
    sel<="1011";
wait for 20 ns;
sel<="1111";
wait for 20 ns;
    wait;
  end process;

END;
```

Simulation –



Name	Value
a[15:0]	1000111000110110
b[15:0]	0010011101001101
sel[3:0]	1111
cin	0
y[15:0]	0101011010000100