ELE227 FUNDAMENTALS OF DIGITAL SYSTEM LABORATORY

PRELIMINARY WORK-3

Student

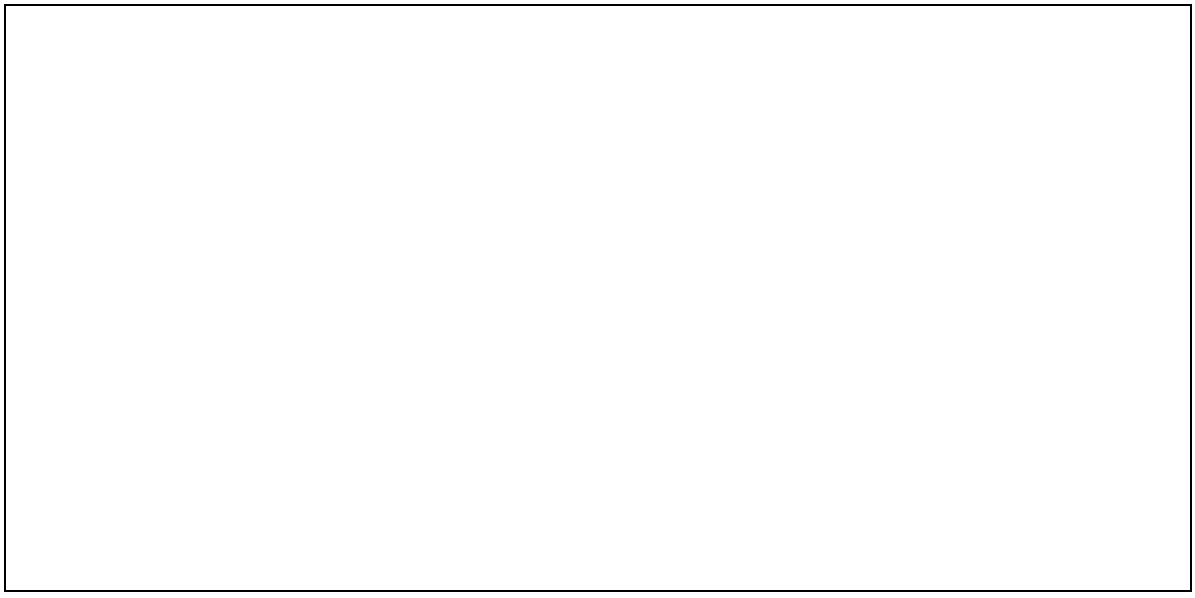
Name: Egemen Can

Surname: Ayduğan

ID: 21728036

1.)

Circuit Diagram:

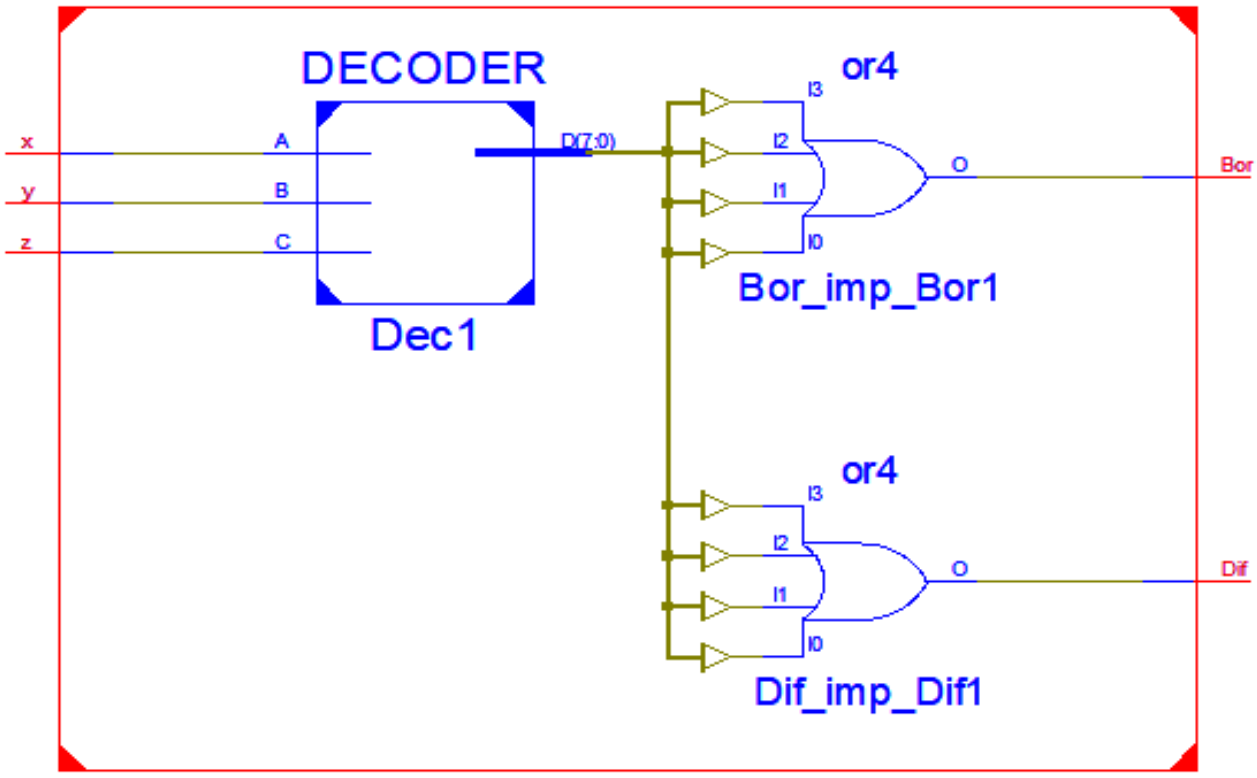


VHDL Implementation

```
entity DECODER is
    Port ( A : in  STD_LOGIC;
                B : in  STD_LOGIC;
                C : in  STD_LOGIC;
           D : out  STD_LOGIC_VECTOR (7
downto 0));
end DECODER;

architecture Behavioral of DECODER is

begin

D(0) <= (not A) and (not B) and (not C);
D(1) <= (not A) and (not B) and C;
D(2) <= (not A) and  B and (not C);
D(3) <= (not A) and  B and  C;
D(4) <=  A and (not B) and (not C);
D(5) <=  A and (not B) and C;
D(6) <=  A and B and (not C);
D(7) <=  A and B and C;

end Behavioral;
```

```
entity FullSubstractorByDecoder is
    Port ( x : in  STD_LOGIC;
           y : in  STD_LOGIC;
           z : in  STD_LOGIC;
           Dif : out  STD_LOGIC;
           Bor : out  STD_LOGIC);
end FullSubstractorByDecoder;

architecture Behavioral of

FullSubstractorByDecoder is

component DECODER
    Port ( A : in  STD_LOGIC;
                B : in  STD_LOGIC;
                C : in  STD_LOGIC;
           D : out  STD_LOGIC_VECTOR (7
downto 0));
end component;

signal T : STD_LOGIC_VECTOR (7 downto 0);

begin

Dec1 : DECODER port map(x,y,z,T(7 downto 0));

Dif <= T(1) or T(2) or T(4) or T(7);
Bor <= T(1) or T(2) or T(3) or T(7);

end Behavioral;
```

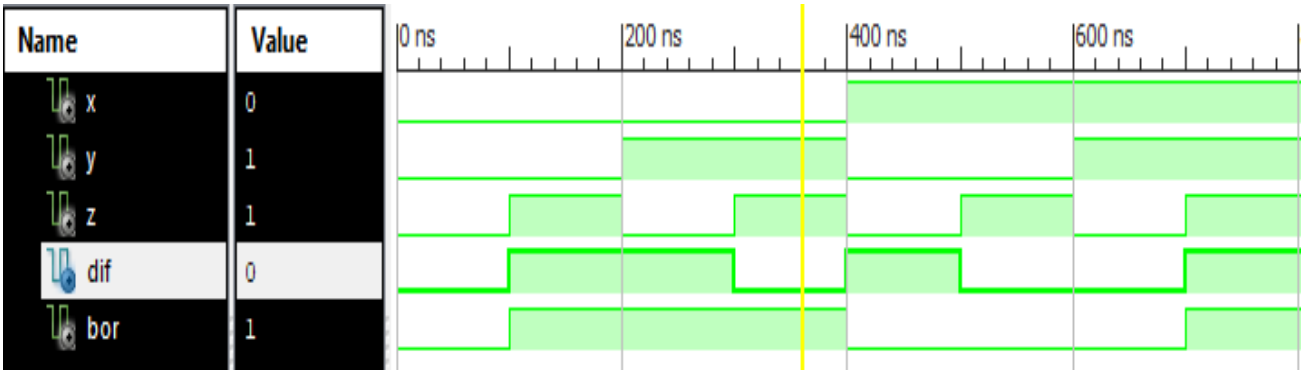
RLT Schematic:



TEST BENCH INPUT:

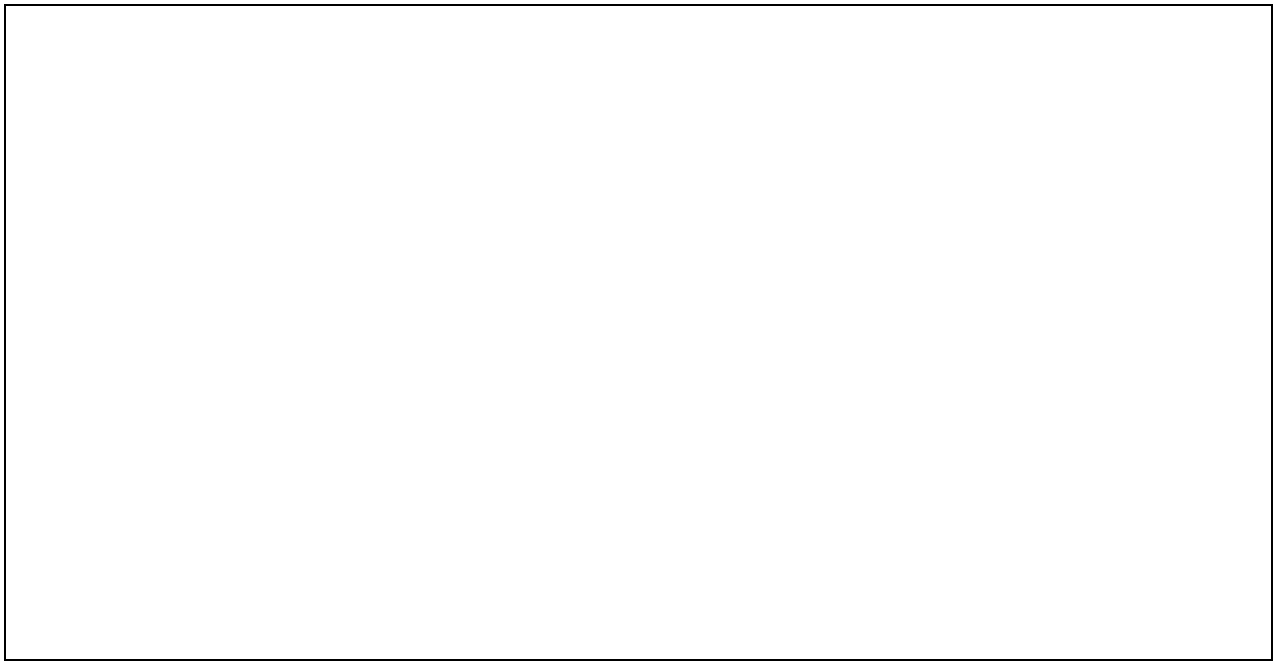
```
stim_proc: process
   begin
              x <='0';
              y <='0';
              z <='0';
      wait for 100 ns;
              x <='0';
              y <='0';
              z <='1';
      wait for 100 ns;
              x <='0';
              y <='1';
              z <='0';
      wait for 100 ns;
              x <='0';
              y <='1';
              z <='1';
      wait for 100 ns;
              x <='1';
              y <='0';
              z <='0';
              wait for 100 ns;
              x <='1';
              y <='0';
              z <='1';
              wait for 100 ns;
              x <='1';
              y <='1';
              z <='0';
              wait for 100 ns;
              x <='1';
              y <='1';
              z <='1';

      wait;
   end process;
```

TEST BENCH RESULTS:



2.)

Circuit Diagram:

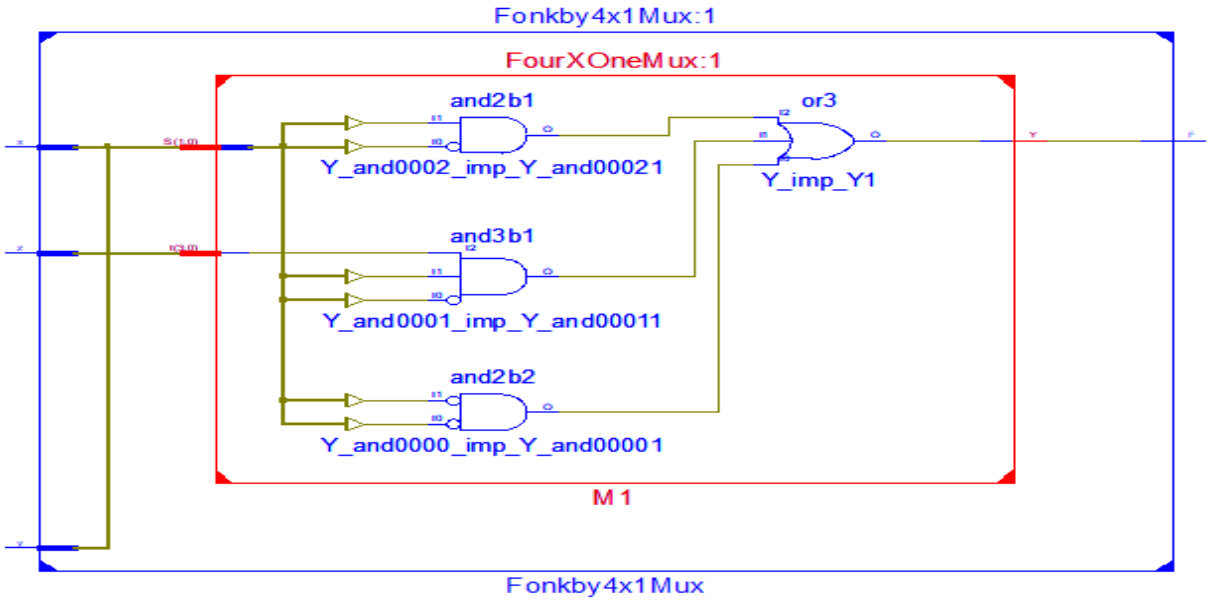


VHDL Implementation

```
entity FourXOneMux is
    Port ( I : in  STD_LOGIC_VECTOR (3 downto
0);
           S : in  STD_LOGIC_VECTOR (1 downto
0);
           Y : out  STD_LOGIC);
end FourXOneMux;

architecture Behavioral of FourXOneMux is

begin

Y <= ((not S(1)) and (not S(0)) and I(0)) or
  ((not S(1)) and S(0) and I(1)) or
  (S(1) and (not S(0)) and I(2)) or
  (S(1) and S(0) and I(3));

end Behavioral;
```

```
entity Fonkby4x1Mux is
    Port ( x : in  STD_LOGIC;
           y : in  STD_LOGIC;
           z : in  STD_LOGIC;
           F : out  STD_LOGIC);
end Fonkby4x1Mux;

architecture Behavioral of Fonkby4x1Mux is

component FourXOneMux
    Port ( I : in  STD_LOGIC_VECTOR (3 downto 0);
           S : in  STD_LOGIC_VECTOR (1 downto 0);
           Y : out  STD_LOGIC);
end component;
signal T : STD_LOGIC_VECTOR (3 downto 0);
signal K : STD_LOGIC_VECTOR (1 downto 0);
begin
 T(0) <= '1';
 T(1) <=  z;
 T(2) <= '1';
 T(3) <= '0';
 K(0) <=  y;
 K(1) <=  x;
M1: FourXOneMux port map (T(3 downto 0),K(1
downto 0),F);
end Behavioral;
```

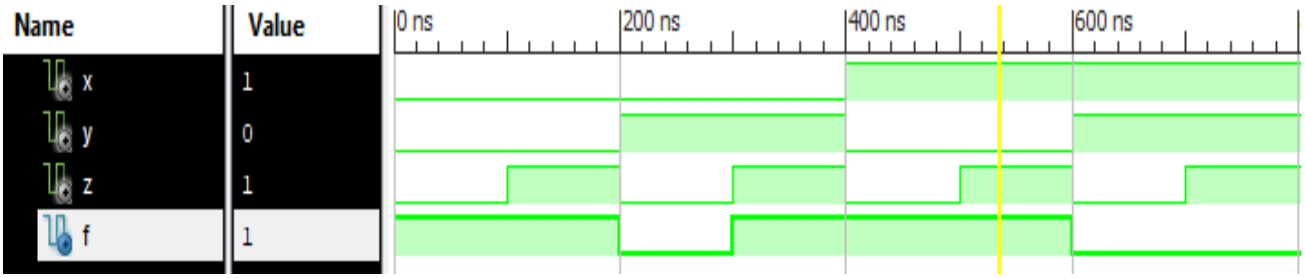
RLT Schematic:



TEST BENCH INPUT:

```
stim_proc: process
   begin
              x<='0';
              y<='0';
              z<='0';
      wait for 100 ns;
              x<='0';
              y<='0';
              z<='1';
      wait for 100 ns;
              x<='0';
              y<='1';
              z<='0';
      wait for 100 ns;
              x<='0';
              y<='1';
              z<='1';
      wait for 100 ns;
              x<='1';
              y<='0';
              z<='0';
      wait for 100 ns;
              x<='1';
              y<='0';
              z<='1';
      wait for 100 ns;
              x<='1';
              y<='1';
              z<='0';
      wait for 100 ns;
              x<='1';
              y<='1';
              z<='1';

      wait;
   end process;
```

TEST BENCH RESULTS:



3.)

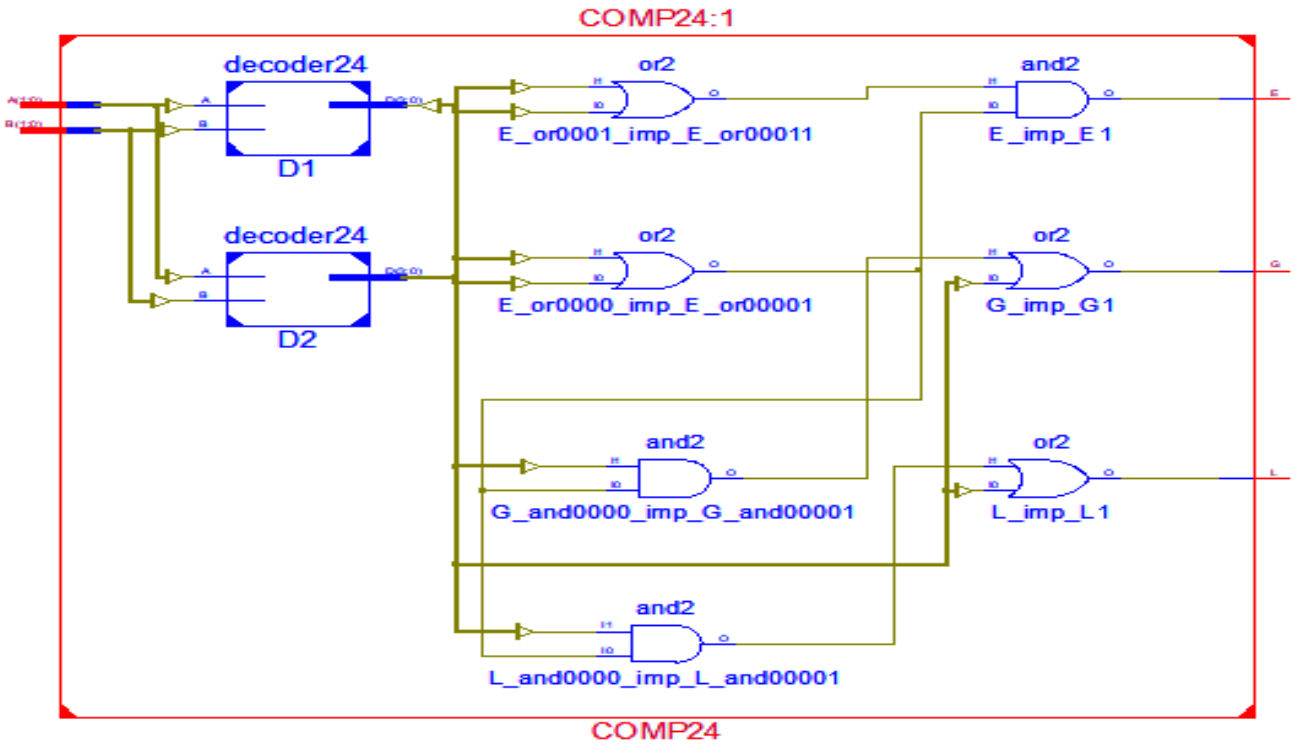
Circuit Diagram:

VHDL Implementation:

```
entity decoder24 is
    Port ( A : in  STD_LOGIC;
           B : in  STD_LOGIC;
           D : out  STD_LOGIC_VECTOR (3
downto 0));
end decoder24;

architecture Behavioral of decoder24 is

begin

D(0) <= (not A) and (not B);
D(1) <= (not A) and  B;
D(2) <= A and (not B);
D(3) <= A and  B;

end Behavioral;
```

```
entity COMP24 is
    Port ( A : in  STD_LOGIC_VECTOR (1
downto 0);
           B : in  STD_LOGIC_VECTOR (1
downto 0);
           G : out  STD_LOGIC;
           E : out  STD_LOGIC;
           L : out  STD_LOGIC);
end COMP24;

architecture Behavioral of COMP24 is

COMPONENT decoder24
    Port ( A : in  STD_LOGIC;
           B : in  STD_LOGIC;
           D : out  STD_LOGIC_VECTOR (3
downto 0));
end COMPONENT;

signal T : STD_LOGIC_VECTOR (7 downto 0);
begin

D1 : decoder24 port map(A(0),B(0),T(3
downto 0));
D2 : decoder24 port map(A(1),B(1),T(7
downto 4));

G <= T(6) or ((T(7) or T(4)) and T(2)) ;
L <= T(5) or ((T(7) or T(4)) and T(1)) ;
E <= (T(7) or T(4)) and (T(3) or T(0));
end Behavioral;
```

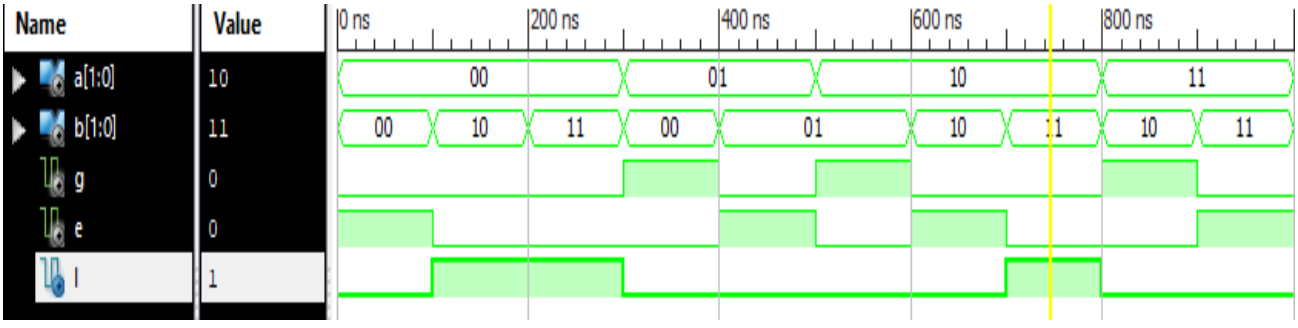
RLT Schematic:



TEST BENCH INPUT:

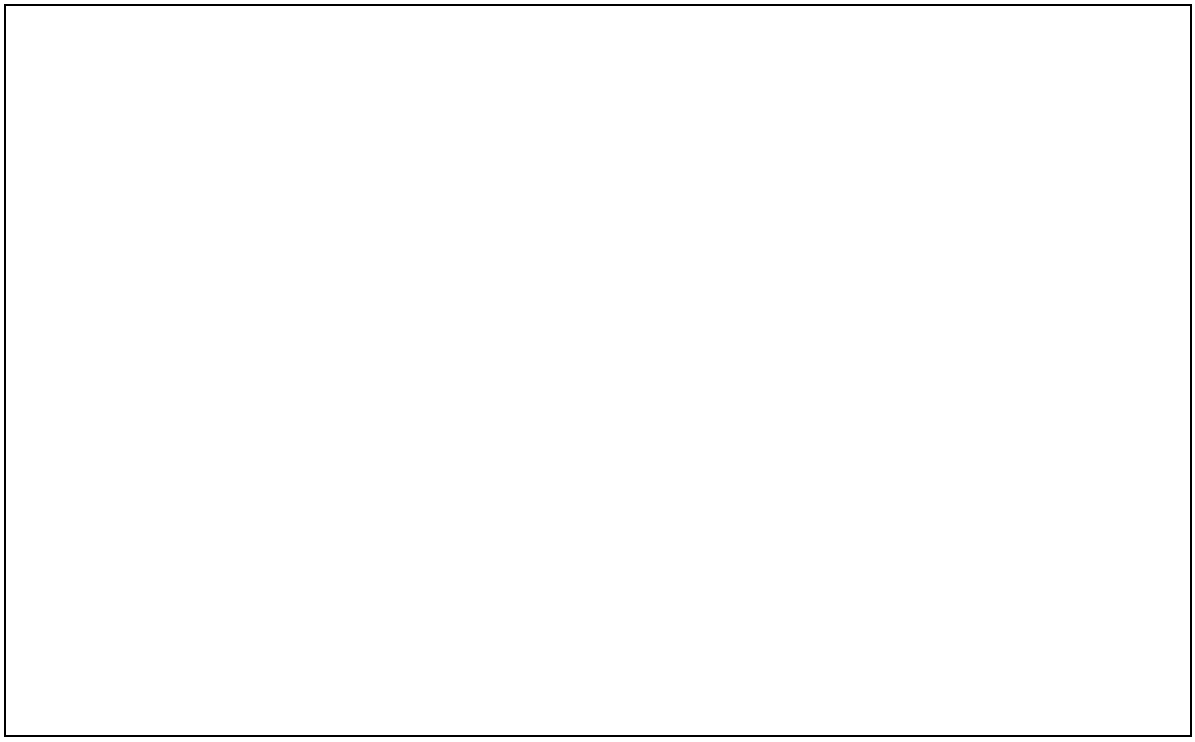
```
stim_proc: process
   begin
            A <= "00";
            B <= "00";
      wait for 100 ns;
            A <= "00";
            B <= "10";
      wait for 100 ns;
            A <= "00";
            B <= "11";
      wait for 100 ns;
            A <= "01";
            B <= "00";
      wait for 100 ns;
            A <= "01";
            B <= "01";
      wait for 100 ns;
            A <= "10";
            B <= "01";
      wait for 100 ns;
            A <= "10";
            B <= "10";
      wait for 100 ns;
            A <= "10";
            B <= "11";
      wait for 100 ns;
            A <= "11";
            B <= "10";
      wait for 100 ns;
            A <= "11";
            B <= "11";
      wait;
   end process;
```

TEST BENCH RESULTS:



4.)

Circuit Diagram:

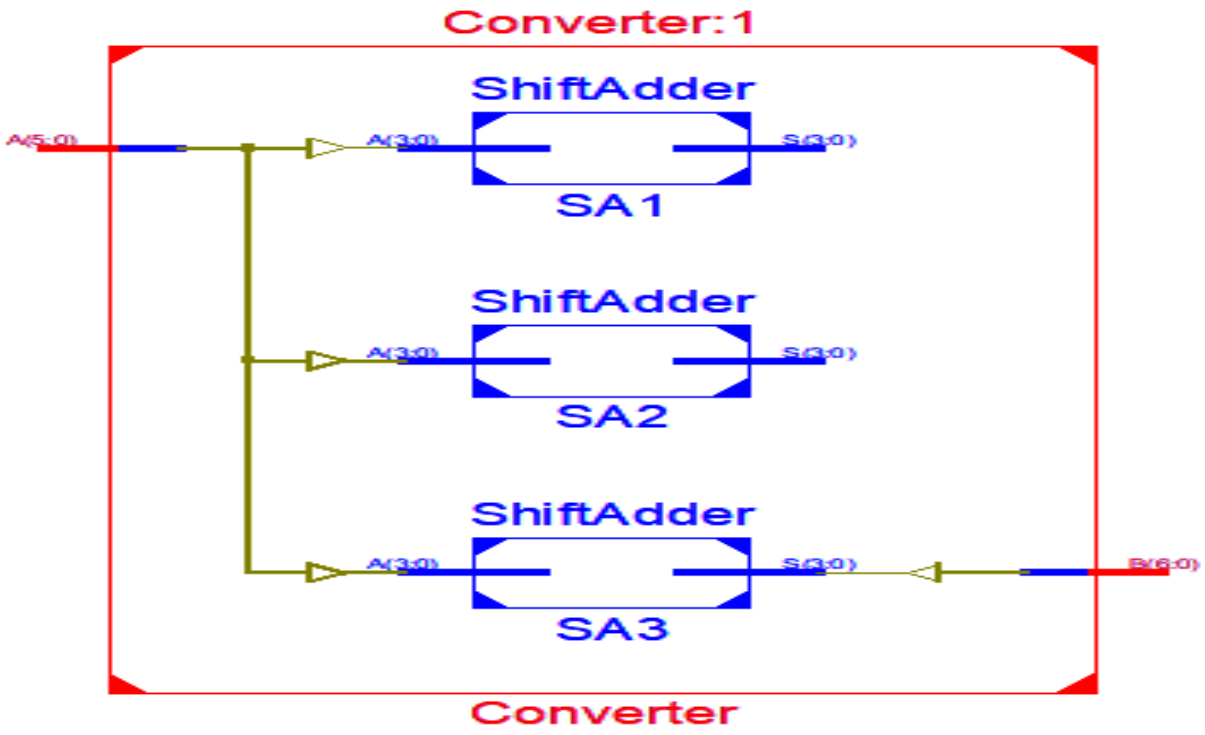


VHDL Implementation

```
entity ShiftAdder is
    Port ( A : in  STD_LOGIC_VECTOR (3 downto
0);
           S : out  STD_LOGIC_VECTOR (3 downto
0));
end ShiftAdder;

architecture Behavioral of ShiftAdder is

begin

 S(0)<= ((not A(3)) and (not A(2)) and A(0))or
((not A(3)) and A(2) and A(1) and (not A(0)))or
(A(3) and (not A(2)) and (not A(1)) and (not
A(0)));

 S(1)<= ((not A(3)) and (not A(2)) and
A(1))or ((not A(3)) and A(1) and A(0))or
(A(3) and (not A(2)) and (not A(1)) and (not
A(0)));

 S(2)<= (A(3) and (not A(2)) and (not A(1)) and
A(0))or ((not A(3)) and A(2) and (not A(1)) and
(not A(0)));

 S(3)<= ((not A(3)) and A(2) and A(0))or
 ((not A(3)) and A(2) and A(1))or (A(3) and
(not A(2)) and (not A(1)));

end Behavioral;
```

```
entity Converter is
    Port ( A : in  STD_LOGIC_VECTOR (5 downto 0);
           B : out  STD_LOGIC_VECTOR (6 downto
0));
end Converter;

architecture Behavioral of Converter is

component ShiftAdder
    Port ( A : in  STD_LOGIC_VECTOR (3 downto 0);
           S : out  STD_LOGIC_VECTOR (3 downto
0));
end component;
signal T : STD_LOGIC_VECTOR (9 downto 0);
signal K : STD_LOGIC_VECTOR (3 downto 0);
begin
K(3)<= '0';
K(2)<= A(5);
K(1)<= A(4);
K(0)<= A(3);
B(6)<= T(9);
T(5)<= A(2);
B(5)<= T(4);
T(0)<= A(1);
SA1 : ShiftAdder port map (K(3 downto 0),T(9
downto 6));
SA2 : ShiftAdder port map (T(8 downto 5),T(4
downto 1));
SA3 : ShiftAdder port map (T(3 downto 0),B(4
downto 1));
B(0) <= A(0);

end Behavioral;
```

RLT Schematic:



TEST BENCH INPUT:

```
stim_proc: process
   begin
            A <= "101010";
      wait for 100 ns;
            A <= "010101";
      wait for 100 ns;
            A <= "111111";
      wait for 100 ns;
            A <= "111110";
      wait for 100 ns;
            A <= "010110";
      wait for 100 ns;
            A <= "010001";
      wait for 100 ns;
            A <= "011011";
      wait for 100 ns;
            A <= "101101";

      wait;
   end process;
```

TEST BENCH RESULTS: