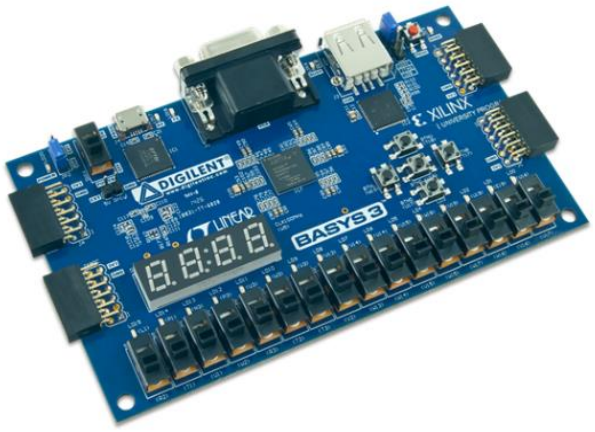
NANO-PROCESSOR DESIGN COMPETITION

220669E - Vishmitha W.D.L
220085J - Chanuka MGG
220382R - Manawadu N.D.
220099F - De Silva D.L.P.H.



In this lab, We designed a 4-bit processor capable of executing 4 instructions. Key outcomes include developing a 4-bit arithmetic unit for addition and subtraction, decoding instructions, and creating multiplexers. Components like the add/subtract unit and program counter will be modified from previous labs. Teamwork skills will be emphasized through communication, coordination, and integrating components developed by different team members.

ASSEMBLY PROGRAM AND ITS MACHINE CODE REPRESENTATION

TOTAL BETWEEN NUMBERS 1 & 3

Machine Code	Assembly Program
"100010000001"	MOV R1,1 ; R1 <-- 1 ;
"100100000010"	MOV R2,2 ; R2 <-- 2 ;
"100110000011"	MOV R3,3 ; R3 <-- 3 ;
"000010100000"	ADD R1,R2 ; R1 <---R1+R2;
"000010110000"	ADD R1,R3 ; R1 <---R1+R3;

Reduce one by one Register 1 (R1) initial value 10

"100010001010"	MOV R1,10
"100100000001"	MOV R2,1
"010100000000"	NEG R2
"000010100000"	ADD R1,R2
"110010000111"	JZR R1 ,7
"110000000011"	JZR R0,3
"000000000000"	
"ZZZZZZZZZZZZ"	

ALL VDHL CODES AND TIMING DIAGRAM

3 to 8 Decoder

- Design Source

```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date: 02/20/2024 03:20:27 PM
-- Design Name:
-- Module Name: Decoder_3_to_8 - 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 Decoder_3_to_8 is

 Port (
 I : in STD_LOGIC_VECTOR (2 downto 0);
 EN : in STD_LOGIC;
 Y : out STD_LOGIC_VECTOR (7 downto 0));
end Decoder_3_to_8;

architecture Behavioral of Decoder_3_to_8 is

 component Decoder_2_to_4
 port(
 I: in STD_LOGIC_VECTOR ;
 EN: in STD_LOGIC;
 Y: out STD_LOGIC_VECTOR );
 end component;

 signal I0,I1 : STD_LOGIC_VECTOR (1 downto 0);
 signal Y0,Y1 : STD_LOGIC_VECTOR (3 downto 0);
 signal en0,en1, I2 : STD_LOGIC;

begin

 Decode_2_to_4_0 : Decoder_2_to_4
 port map(
 I => I0,
 EN => en0,
 Y => Y0 );
```

```
Decode_2_to_4_1 : Decoder_2_to_4
port map(
I => I1,
EN => en1,
Y => Y1 );

en0 <= NOT(I2) AND EN;
en1 <= I2 AND EN;
I0 <= I(1 downto 0);
I1 <= I(1 downto 0);
I2 <= I(2);
Y(3 downto 0) <= Y0;
Y(7 downto 4) <= Y1;

end Behavioral;
```

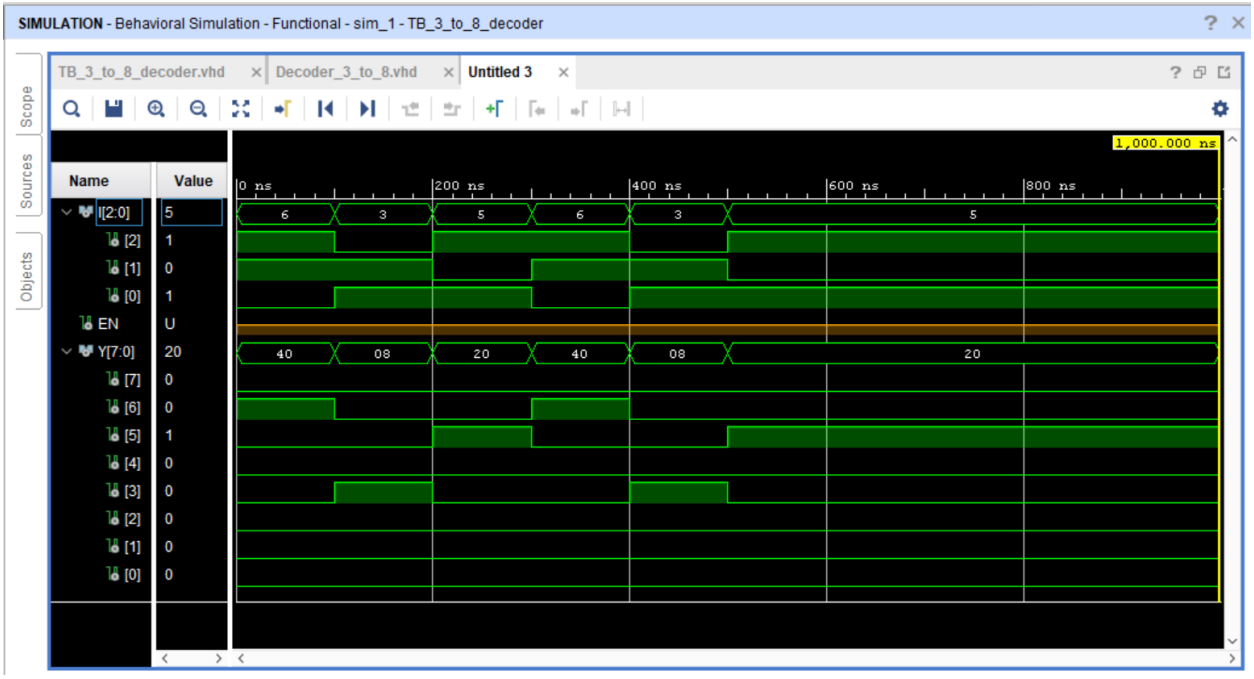
- Simulation Source

```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date: 04/10/2024 09:15:46 PM
-- Design Name:
-- Module Name: TB_3_to_8_decoder - 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 TB_3_to_8_decoder is
--  Port ( );
end TB_3_to_8_decoder;

architecture Behavioral of TB_3_to_8_decoder is
COMPONENT Decoder_3_to_8
Port (
   I : IN std_logic_vector(2 downto 0);
   EN : in STD_LOGIC;
   Y : OUT std_logic_vector(7 downto 0)
   );

END COMPONENT;

SIGNAL I : std_logic_vector(2 downto 0);
```

```
SIGNAL EN : std_logic;
SIGNAL Y : std_logic_vector(7 downto 0);

begin

uut: Decoder_3_to_8 PORT MAP (
  I => I,
  EN => '1',
  Y => Y
  );
process
-----  index to binary (220382---  110 101 110 011  011 110)
  begin
  I <= "110";

  wait for 100 ns;
  I <= "011";

  wait for 100 ns;
  I <= "101";

  wait for 100 ns;
  I <= "110";

  wait for 100 ns;
  I <= "011";

  wait for 100 ns;
  I <= "101";

  wait;
  end process;
end Behavioral;
```

- Timing Diagram



Add & Sub Unit

- Design Source

```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date: 03/15/2024 12:29:06 PM
-- Design Name:
-- Module Name: AU - 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 ADD_SUB_unit is
   Port ( INPUT_0: in STD_LOGIC_VECTOR (3 downto 0);
          INPUT_1: in STD_LOGIC_VECTOR (3 downto 0);
          AddSub_Select : in STD_LOGIC;
          OUTPUT:out STD_LOGIC_VECTOR(3 downto 0);
          Zero : out STD_LOGIC;
          OverFlow : out STD_LOGIC);
end ADD_SUB_unit;

architecture Behavioral of ADD_SUB_unit is

---RCA component
```

```
component RCA_4 Port (
    A0 : in STD_LOGIC;
    A1 : in STD_LOGIC;
    A2 : in STD_LOGIC;
    A3 : in STD_LOGIC;
    B0 : in STD_LOGIC;
    B1 : in STD_LOGIC;
    B2 : in STD_LOGIC;
    B3 : in STD_LOGIC;
    C_In : in STD_LOGIC;
    S0 : out STD_LOGIC;
    S1 : out STD_LOGIC;
    S2 : out STD_LOGIC;
    S3 : out STD_LOGIC;
    C_out : out STD_LOGIC;
    overflow:out STD_LOGIC;
    zero: out STD_LOGIC);
end component;

signal overflow_bit,C_out : STD_LOGIC;

begin

RCA_4_0 : RCA_4
port map (
  A0 => INPUT_0(0),
  A1 => INPUT_0(1),
  A2 => INPUT_0(2),
  A3 => INPUT_0(3),
  B0 => INPUT_1(0),
  B1 => INPUT_1(1),
  B2 => INPUT_1(2),
  B3 => INPUT_1(3),
```

```
    C_In => AddSub_Select,
    S0=> OUTPUT(0),
    S1=> OUTPUT(1),
    S2=> OUTPUT(2),
    S3=> OUTPUT(3),
    C_out=> C_out,
    overflow =>OverFlow,
    zero => Zero
);

end Behavioral;
```

- Simulation Source

```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date: 03/15/2024 12:29:06 PM
-- Design Name:
-- Module Name: AU - Behavioral
-- Project Name:
-- Target Devices:
-- Tool Versions:
-- Description:
--
-- Dependencies:
--
-- Revision:
-- Revision 0.01 - File Created
-- Additional Comments:
--
----------------------------------------------------------------------------------
```

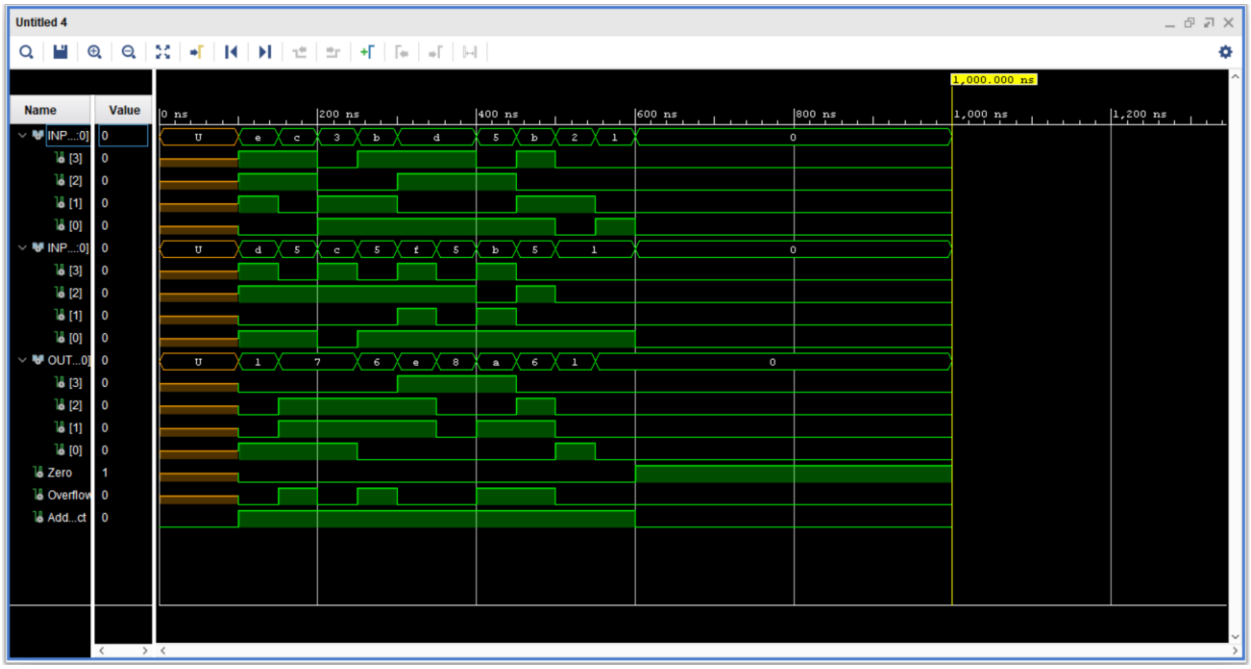
```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

-- 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 ADD_SUB_unit is
   Port ( INPUT_0: in STD_LOGIC_VECTOR (3 downto 0);
          INPUT_1: in STD_LOGIC_VECTOR (3 downto 0);
          AddSub_Select : in STD_LOGIC;
          OUTPUT:out STD_LOGIC_VECTOR(3 downto 0);
          Zero : out STD_LOGIC;
          OverFlow : out STD_LOGIC);
end ADD_SUB_unit;

architecture Behavioral of ADD_SUB_unit is

---RCA component
component RCA_4 Port (
      A0 : in STD_LOGIC;
      A1 : in STD_LOGIC;
      A2 : in STD_LOGIC;
      A3 : in STD_LOGIC;
      B0 : in STD_LOGIC;
      B1 : in STD_LOGIC;
      B2 : in STD_LOGIC;
```

```
    B3 : in STD_LOGIC;
    C_In : in STD_LOGIC;
    S0 : out STD_LOGIC;
    S1 : out STD_LOGIC;
    S2 : out STD_LOGIC;
    S3 : out STD_LOGIC;
    C_out : out STD_LOGIC;
    overflow:out STD_LOGIC;
    zero: out STD_LOGIC);
end component;

signal overflow_bit,C_out : STD_LOGIC;

begin

RCA_4_0 : RCA_4
port map (
  A0 => INPUT_0(0),
  A1 => INPUT_0(1),
  A2 => INPUT_0(2),
  A3 => INPUT_0(3),
  B0 => INPUT_1(0),
  B1 => INPUT_1(1),
  B2 => INPUT_1(2),
  B3 => INPUT_1(3),
  C_In => AddSub_Select,
  S0=> OUTPUT(0),
  S1=> OUTPUT(1),
  S2=> OUTPUT(2),
  S3=> OUTPUT(3),
  C_out=> C_out,
  overflow =>OverFlow,
  zero => Zero
```

);

end Behavioral;

- Timing Diagram



8 to 1 Mux

- Design Source

-- Company:

-- Engineer:

--

-- Create Date: 20.02.2024 15:36:34

-- Design Name:

-- Module Name: Mux_8_to_1 - 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 Mux_8_to_1 is
   Port ( S  : IN STD_LOGIC_VECTOR (2 downto 0);
          D0 : IN STD_LOGIC_VECTOR (3 downto 0);
          D1 : IN STD_LOGIC_VECTOR (3 downto 0);
          D2 : IN STD_LOGIC_VECTOR (3 downto 0);
          D3 : IN STD_LOGIC_VECTOR (3 downto 0);
          D4 : IN STD_LOGIC_VECTOR (3 downto 0);
          D5 : IN STD_LOGIC_VECTOR (3 downto 0);
          D6 : IN STD_LOGIC_VECTOR (3 downto 0);
          D7 : IN STD_LOGIC_VECTOR (3 downto 0);
```

```
        EN : IN STD_LOGIC;
        Y  : OUT STD_LOGIC_VECTOR (3 downto 0));
end Mux_8_to_1;

architecture Behavioral of Mux_8_to_1 is

COMPONENT Decoder_3_to_8
   PORT(
   I : IN STD_LOGIC_VECTOR (2 downto 0);
   EN : IN STD_LOGIC;
   Y : OUT STD_LOGIC_VECTOR (7 downto 0));
END COMPONENT;

SIGNAL Y_Y : STD_LOGIC_VECTOR (7 downto 0);

begin
   Decoder_3_to_8_0 : Decoder_3_to_8
   PORT MAP(
      I => S,
      EN => '1',
      Y => Y_Y);

   --Y <= ((D(0) AND Y0(0)) OR (D(1) AND Y0(1)) OR (D(2) AND Y0(2)) OR (D(3) AND Y0(3)) OR (D(4) AND Y0(4))
OR (D(5) AND Y0(5)) OR (D(6) AND Y0(6)) OR (D(7) AND Y0(7)));
   Y(0) <= ( D0(0) AND Y_Y(0)) OR ( D1(0) AND Y_Y(1)) OR ( D2(0) AND Y_Y(2)) OR (D3(0) AND Y_Y(3)) OR (D4(0)
AND Y_Y(4)) OR (D5(0) AND Y_Y(5)) OR (D6(0) AND Y_Y(6)) OR (D7(0) AND Y_Y(7));
   Y(1) <= ( D0(1) AND Y_Y(0)) OR ( D1(1) AND Y_Y(1)) OR ( D2(1) AND Y_Y(2)) OR (D3(1) AND Y_Y(3)) OR (D4(1)
AND Y_Y(4)) OR (D5(1) AND Y_Y(5)) OR (D6(1) AND Y_Y(6)) OR (D7(1) AND Y_Y(7));
   Y(2) <= ( D0(2) AND Y_Y(0)) OR ( D1(2) AND Y_Y(1)) OR ( D2(2) AND Y_Y(2)) OR (D3(2) AND Y_Y(3)) OR (D4(2)
AND Y_Y(4)) OR (D5(2) AND Y_Y(5)) OR (D6(2) AND Y_Y(6)) OR (D7(2) AND Y_Y(7));
   Y(3) <= ( D0(3) AND Y_Y(0)) OR ( D1(3) AND Y_Y(1)) OR ( D2(3) AND Y_Y(2)) OR (D3(3) AND Y_Y(3)) OR (D4(3)
AND Y_Y(4)) OR (D5(3) AND Y_Y(5)) OR (D6(3) AND Y_Y(6)) OR (D7(3) AND Y_Y(7));

end Behavioral;
```

- Simulation Source

```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date: 02/20/2024 04:43:07 PM
-- Design Name:
-- Module Name: TB_MUX_8_to_1 - 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;
```

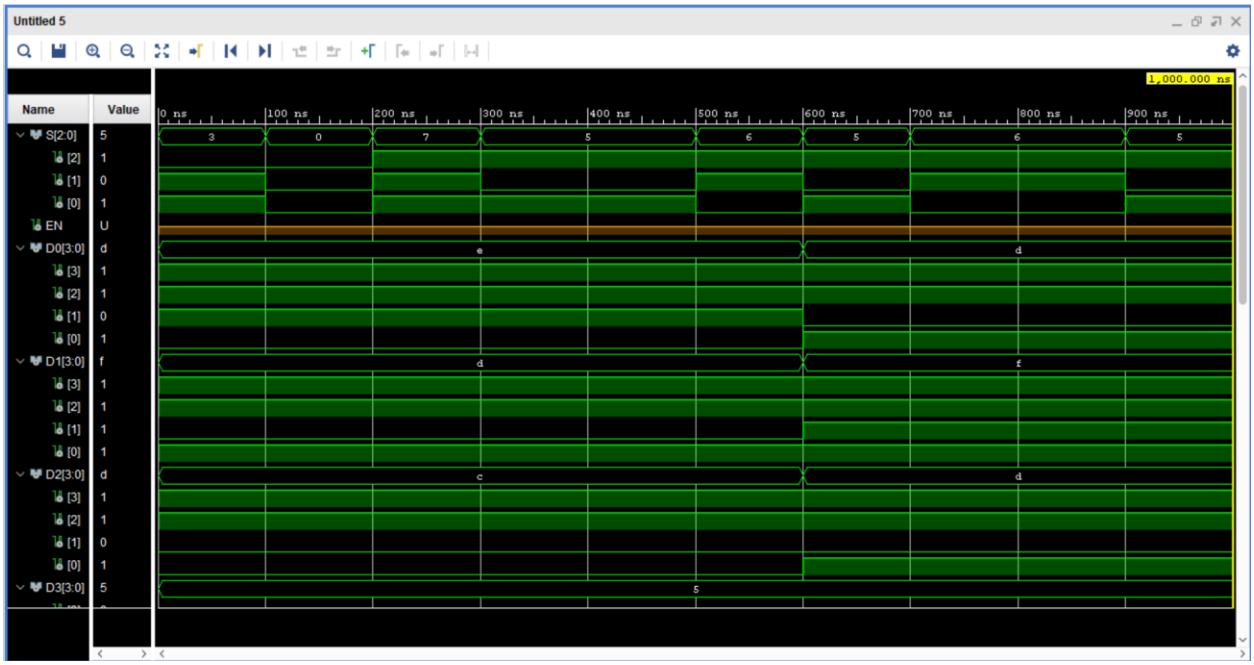
```
entity TB_MUX_8_to_1 is
--  Port ( );
end TB_MUX_8_to_1;

architecture Behavioral of TB_MUX_8_to_1 is
COMPONENT Mux_8_to_1
   Port ( S  : IN STD_LOGIC_VECTOR (2 downto 0);
          D0 : IN STD_LOGIC_VECTOR (3 downto 0);
          D1 : IN STD_LOGIC_VECTOR (3 downto 0);
          D2 : IN STD_LOGIC_VECTOR (3 downto 0);
          D3 : IN STD_LOGIC_VECTOR (3 downto 0);
          D4 : IN STD_LOGIC_VECTOR (3 downto 0);
          D5 : IN STD_LOGIC_VECTOR (3 downto 0);
          D6 : IN STD_LOGIC_VECTOR (3 downto 0);
          D7 : IN STD_LOGIC_VECTOR (3 downto 0);
          EN : IN STD_LOGIC;
          Y  : OUT STD_LOGIC_VECTOR (3 downto 0));
END COMPONENT;

SIGNAL S : std_logic_vector(2 downto 0);
SIGNAL EN : std_logic;
SIGNAL D0,D1,D2,D3,D4,D5,D6,D7,Y : std_logic_vector(3 downto 0);
 --Outputs

begin
uut: Mux_8_to_1  PORT MAP (
S=>S,
D0=>D0,
D1=>D1,
D2=>D2,
D3=>D3,
D4=>D4,
```

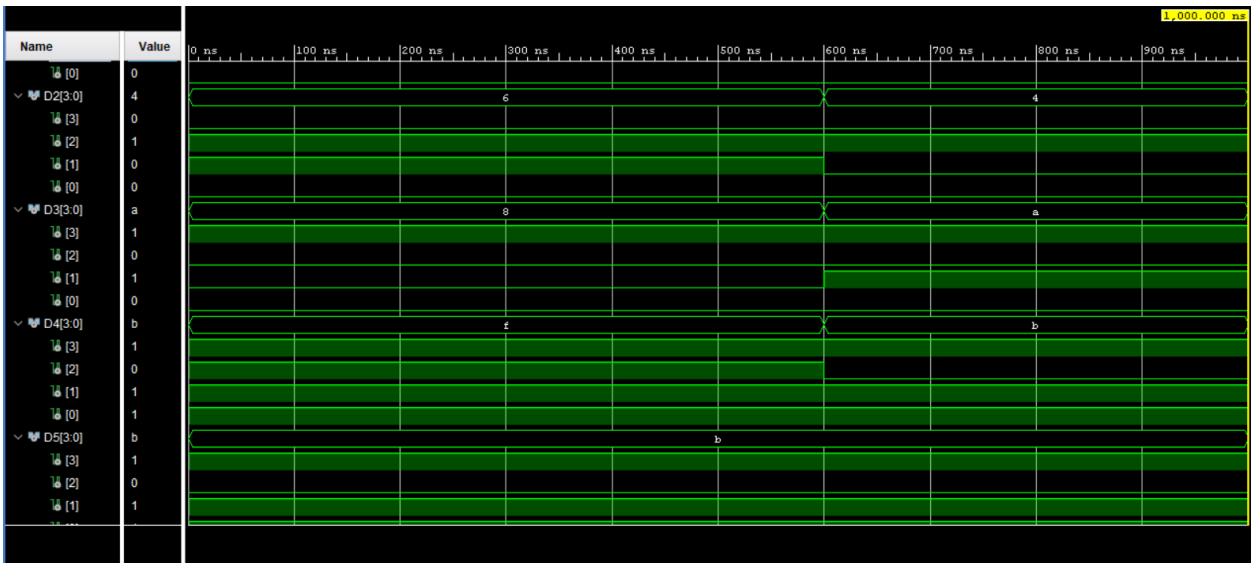
```
D5=>D5,
D6=>D6,
D7=>D7,
EN=>EN,
Y=>Y
 );
process
 begin

 --Index(Nimash) 11 0101 1100 1101 1110
 --Index(Pamoj)  110 101 101 111 000 011
 --Index(Lahiru) 11 0101 1101 1111 1101
 --Index(Gamith) 11 0101 1011 1011 0101

D0<= "1110";
D1<= "1101";
D2<= "1100";
D3<= "0101";
D4<= "0011";
D5<= "1100";
D6<= "1011";
D7<= "0101";

S<="011";


 wait for 100 ns;
 S<="000";


 wait for 100 ns;
 S<="111";
```

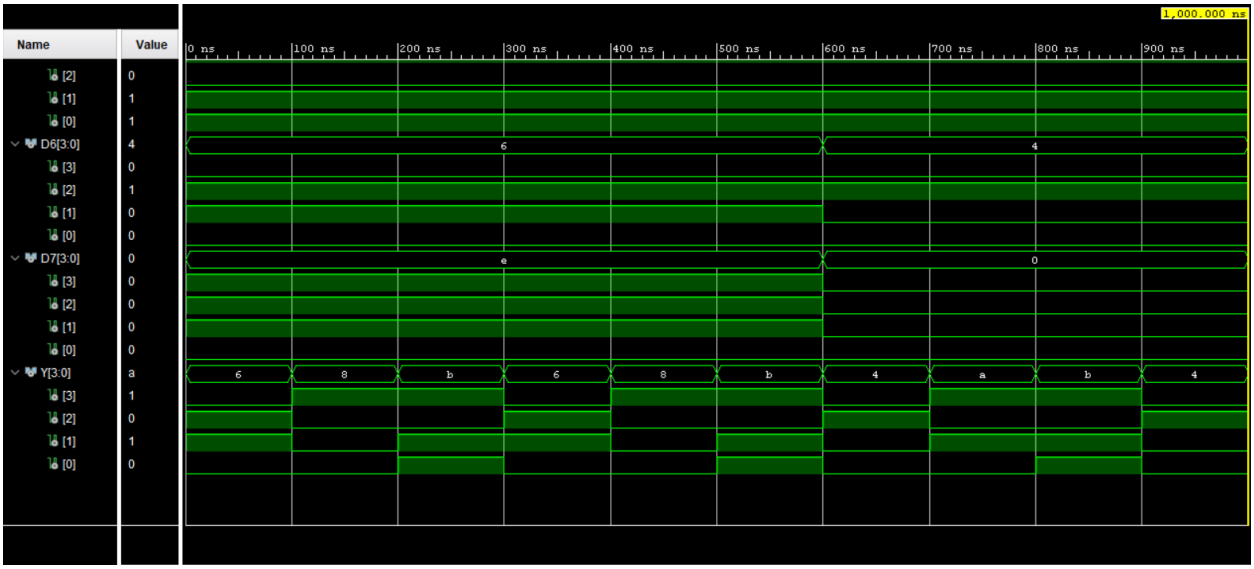
```
 wait for 100 ns;


S<="101";


 wait for 100 ns;

S<="101";


 wait for 100 ns;

S<="110";

  --Index(Lahiru) 11 0101 1101 1111 1101

 wait for 100 ns;
D0<= "1101";
 D1<= "1111";
 D2<= "1101";
 D3<= "0101";
 D4<= "1101";
 D5<= "1111";
 D6<= "1101";
 D7<= "0101";

  --Index(Gamith) 110 101 101 110 110 101

S<="101";
```

```
 wait for 100 ns;
 S<="110";


 wait for 100 ns;
 S<="110";


 wait for 100 ns;


S<="101";


 wait for 100 ns;

S<="101";


 wait for 100 ns;

S<="110";


 wait;

end process;
end Behavioral;
```

- Timing Diagram







3 Bit Ripple Carry Adder (RCA)

- Design Source

-- Company:

-- Engineer:

--

-- Create Date: 04/08/2024 01:22:22 PM

-- Design Name:

-- Module Name: RCA_3_bit_Adder - 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 RCA_3_bit_Adder is
   Port (
      A0 : in STD_LOGIC;
      A1 : in STD_LOGIC;
      A2 : in STD_LOGIC;
      B0 : in STD_LOGIC;
      B1 : in STD_LOGIC;
      B2 : in STD_LOGIC;
      C_in : in STD_LOGIC;
      S0 : out STD_LOGIC;
      S1 : out STD_LOGIC;
      S2 : out STD_LOGIC;
      C_out : out STD_LOGIC);
end RCA_3_bit_Adder;
```

```
architecture Behavioral of RCA_3_bit_Adder is
component FA
   port (
   A: in std_logic;
   B: in std_logic;
   C_in: in std_logic;
   S: out std_logic;
   C_out: out std_logic);
end component;

SIGNAL FA0_S, FA0_C, FA1_S, FA1_C, FA2_S, FA2_C: std_logic;

begin
FA_0 : FA
   port map (
      A => A0,
      B => '1',
      C_in => '0', -- Set to ground
      S => S0,
      C_out => FA0_C);
FA_1 : FA
   port map (
      A => A1,
      B => '0',
      C_in => FA0_C,
      S => S1,
      C_out => FA1_C);
 FA_2 : FA
   port map (
      A => A2,
      B => '0',
      C_in => FA1_C,
      S => S2,
```

```
      C_out => FA2_C);

end Behavioral;
```

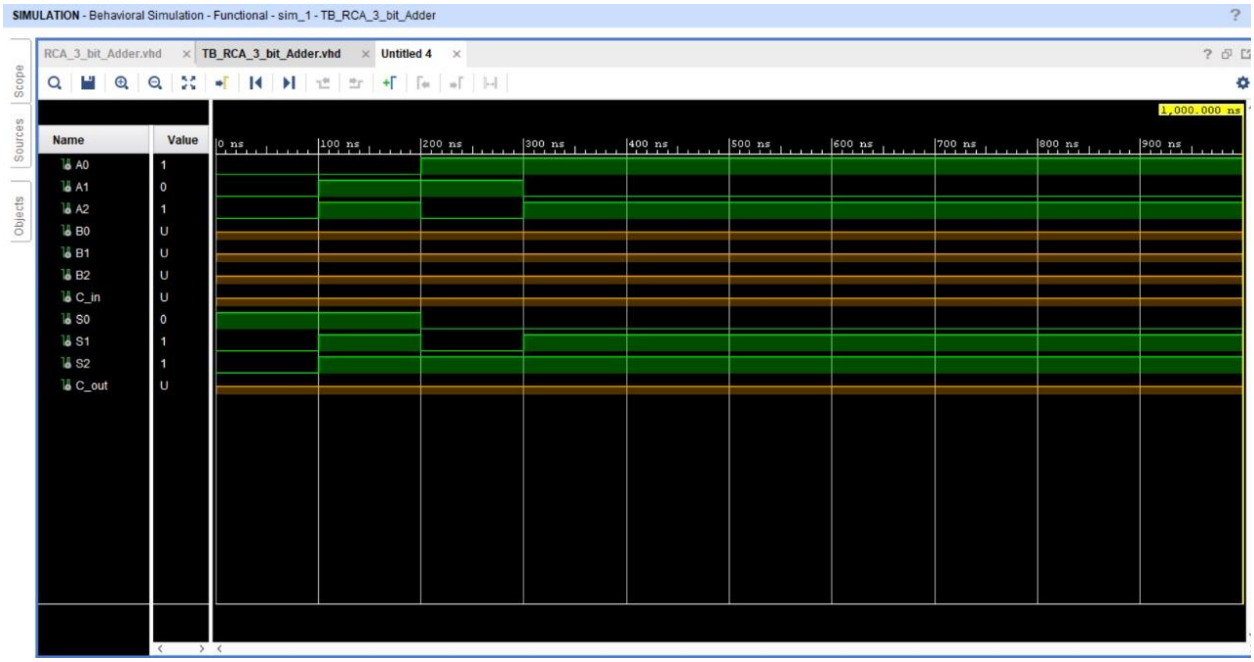
- Simulation Source

```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date: 04/08/2024 01:31:17 PM
-- Design Name:
-- Module Name: TB_RCA_3_bit_Adder - 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 TB_RCA_3_bit_Adder is
--  Port ( );
end TB_RCA_3_bit_Adder;

architecture Behavioral of TB_RCA_3_bit_Adder is
COMPONENT RCA_3_bit_Adder
   PORT(A0,A1,A2,B0,B1,B2,C_in:IN STD_LOGIC;
      S0,S1,S2,C_out:OUT STD_LOGIC);
END COMPONENT;
SIGNAL A0,A1,A2,B0,B1,B2,C_in :std_logic;
SIGNAL S0,S1,S2,C_out:std_logic;

begin
UUT: RCA_3_bit_Adder PORT MAP(
   A0=>A0,
   A1=>A1,
   A2=>A2,
   B0=>B0,
   B1=>B1,
   B2=>B2,
   C_in=>C_in,
   S0 => S0,
   S1=>S1,
   S2=>S2,
   C_out => C_out
);
process
begin
---In this simulation we use last 3 bits of 4 group members
```

```
A0 <= '0'; -- set initial values
A1 <= '0';
A2 <= '0';

WAIT FOR 100 ns; -- after 100 ns change inputs
--220382R's last 3 bits
A0 <= '0'; -- set initial values
A1 <= '1';
A2 <= '1';
WAIT FOR 100 ns; --change again
--220099F's last 3 bits
A0 <= '1'; -- set initial values
A1 <= '1';
A2 <= '0';
WAIT FOR 100 ns; --change again
--220669E's last 3 bits
A0 <= '1'; -- set initial values
A1 <= '0';
A2 <= '1';

WAIT FOR 100 ns; --change again
--220085's last 3 bits
A0 <= '1'; -- set initial values
A1 <= '0';
A2 <= '1';

WAIT; -- will wait forever
end process;

end Behavioral;
```

- Timing Diagram



2 way 3 bit Mux

- Design Source

-- Company:

-- Engineer:

--

-- Create Date: 04/09/2024 12:02:43 AM

-- Design Name:

-- Module Name: Mux_2_to_1_3bit - 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 Mux_2_to_1_3bit is
   Port ( S : in STD_LOGIC;
          D0 : in STD_LOGIC_VECTOR (2 downto 0);--if, s=0 => D0 select, & S=1 => D1 select
          D1 : in STD_LOGIC_VECTOR (2 downto 0);
          EN : in STD_LOGIC;
          Y : out STD_LOGIC_VECTOR (2 downto 0));
end Mux_2_to_1_3bit;

architecture Behavioral of Mux_2_to_1_3bit is

begin
Y(0) <= (D0(0) AND NOT(S)) OR (D1(0) AND S);
Y(1) <= (D0(1) AND NOT(S)) OR (D1(1) AND S);
```

```
Y(2) <= (D0(2) AND NOT(S)) OR (D1(2) AND S);


end Behavioral;
```

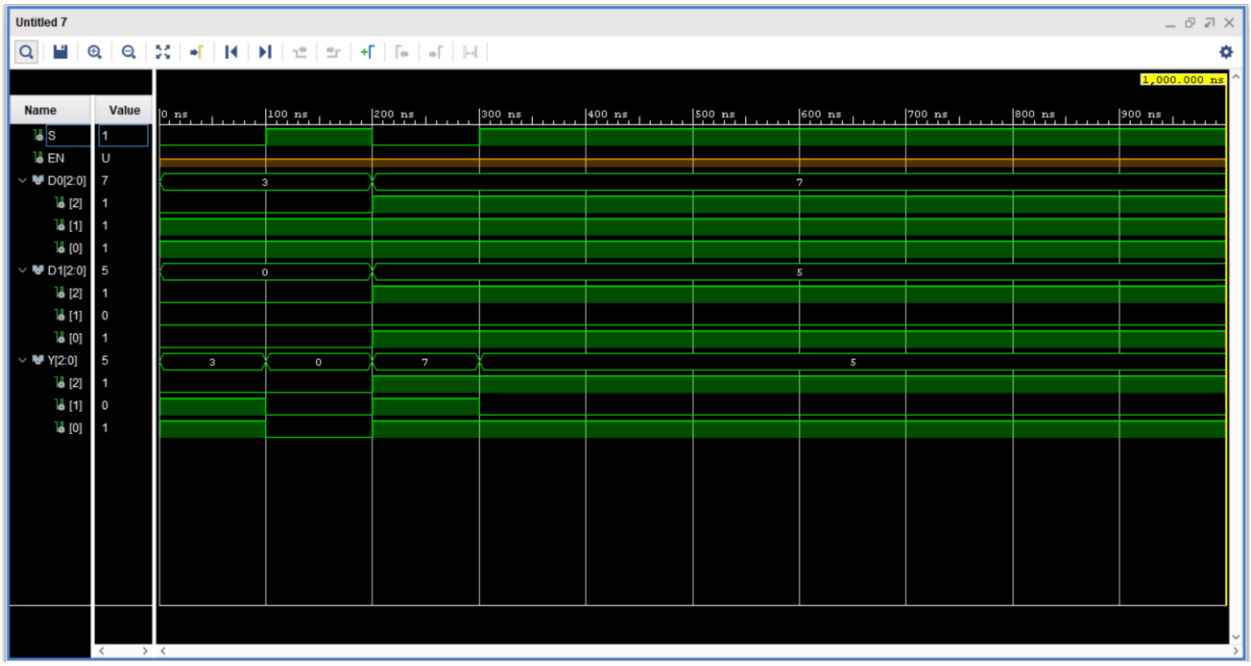
- Simulation Source

```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date: 04/09/2024 12:03:52 AM
-- Design Name:
-- Module Name: TB_Mux_2_to_1_3bit - 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 TB_Mux_2_to_1_3bit is
--  Port ( );
end TB_Mux_2_to_1_3bit;

architecture Behavioral of TB_Mux_2_to_1_3bit is

COMPONENT Mux_2_to_1_3bit

PORT(  S,EN : IN std_logic;
D0,D1 : IN std_logic_vector(2 downto 0);
Y: OUT std_logic_vector(2 downto 0));

END COMPONENT;


SIGNAL S,EN : std_logic;
SIGNAL D0,D1,Y : std_logic_vector(2 downto 0);

begin
UUT: Mux_2_to_1_3bit PORT MAP(
S =>S,
EN =>'1',
D0 => D0,
D1 => D1,
Y => Y
);
```

```
process
begin
D0 <= "011";
D1 <= "010";

S <= '0';
wait for 100 ns;
S <= '1';
wait for 100 ns;

D0 <= "101";
D1 <= "111";

S <= '0';
wait for 100 ns;
S <='1';
wait for 100 ns;

wait;
end process;
end Behavioral;
```

- Timing Diagram



2 way 4 bit Mux

- Design Source

-- Company:

-- Engineer:

--

-- Create Date: 04/08/2024 11:26:32 PM

-- Design Name:

-- Module Name: Mux_2_to_1_4bit - 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 Mux_2_to_1_4bit is
   Port ( Load_Select : in STD_LOGIC; --Load_Select=0 (D0/Instruction_immediate_value) ---Load_Select=1 (D1/Sum_Value)
          Mux_2_to_1_4bit_in_0 : in STD_LOGIC_VECTOR (3 downto 0);
          Mux_2_to_1_4bit_in_1 : in STD_LOGIC_VECTOR (3 downto 0);
          EN : in STD_LOGIC;
          Y : out STD_LOGIC_VECTOR (3 downto 0));
end Mux_2_to_1_4bit;

architecture Behavioral of Mux_2_to_1_4bit is

begin
Y(0) <= (Mux_2_to_1_4bit_in_0(0) AND NOT(Load_Select)) OR (Mux_2_to_1_4bit_in_1(0) AND Load_Select);
Y(1) <= (Mux_2_to_1_4bit_in_0(1) AND NOT(Load_Select)) OR (Mux_2_to_1_4bit_in_1(1) AND Load_Select);
Y(2) <= (Mux_2_to_1_4bit_in_0(2) AND NOT(Load_Select)) OR (Mux_2_to_1_4bit_in_1(2) AND Load_Select);
Y(3) <= (Mux_2_to_1_4bit_in_0(3) AND NOT(Load_Select)) OR (Mux_2_to_1_4bit_in_1(3) AND Load_Select);
```

end Behavioral;

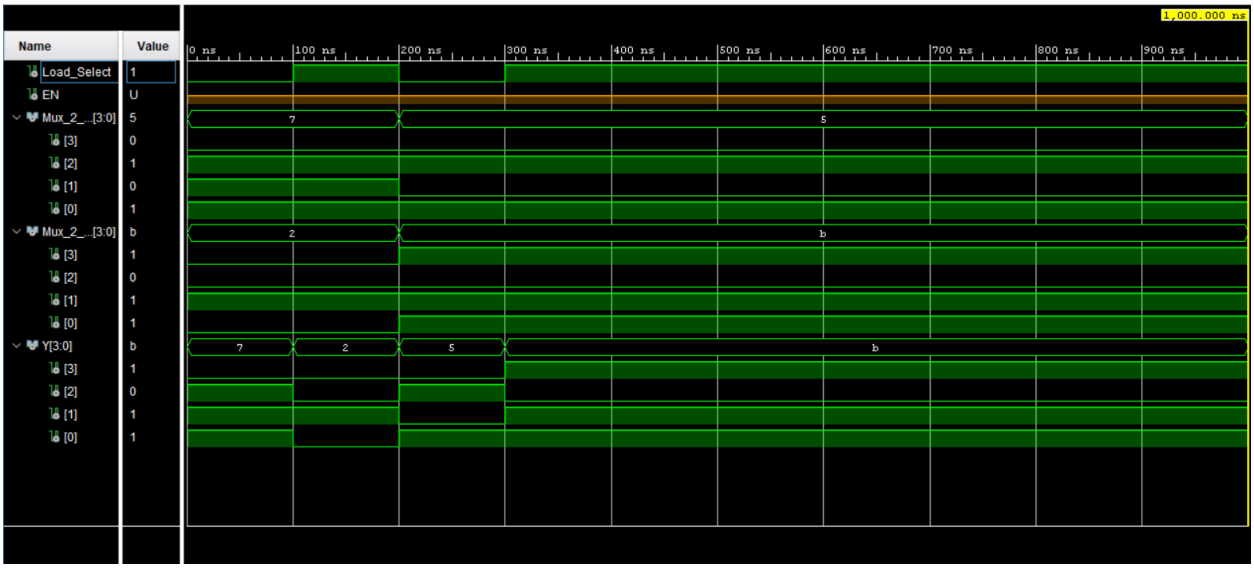
- Simulation Source

```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date: 04/08/2024 11:38:29 PM
-- Design Name:
-- Module Name: TB_Mux_2_to_1_4bit - 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 TB_Mux_2_to_1_4bit is
--  Port ( );
end TB_Mux_2_to_1_4bit;

architecture Behavioral of TB_Mux_2_to_1_4bit is

COMPONENT Mux_2_to_1_4bit

PORT(  Load_Select,EN : IN std_logic;
Mux_2_to_1_4bit_in_0,Mux_2_to_1_4bit_in_1: IN std_logic_vector(3 downto 0);
Y: OUT std_logic_vector(3 downto 0));

END COMPONENT;


SIGNAL Load_Select,EN : std_logic;
SIGNAL Mux_2_to_1_4bit_in_0,Mux_2_to_1_4bit_in_1,Y : std_logic_vector(3 downto 0);

begin
UUT: Mux_2_to_1_4bit PORT MAP(
Load_Select =>Load_Select,
EN =>'1',
Mux_2_to_1_4bit_in_0 => Mux_2_to_1_4bit_in_0,
Mux_2_to_1_4bit_in_1 => Mux_2_to_1_4bit_in_1,
Y => Y
);

process
begin
Mux_2_to_1_4bit_in_0 <= "0111";
```

```
Mux_2_to_1_4bit_in_1 <= "0010";

Load_Select <= '0';
wait for 100 ns;
Load_Select <= '1';
wait for 100 ns;

Mux_2_to_1_4bit_in_0 <= "0101";
Mux_2_to_1_4bit_in_1 <= "1011";

Load_Select <= '0';
wait for 100 ns;
Load_Select <='1';
wait for 100 ns;

wait;
end process;
end Behavioral;
```

- Timing Diagram



Program Counter

- Design Source

```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date: 04/07/2024 04:22:28 PM
-- Design Name:
-- Module Name: ProgramCounter - 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 ProgramCounter is
  Port ( Reset_button : in STD_LOGIC;
       EN:in STD_LOGIC;
       D : in STD_LOGIC_VECTOR (2 downto 0);
       Clk : in STD_LOGIC;
       Q : out STD_LOGIC_VECTOR (2 downto 0));
end ProgramCounter;

architecture Behavioral of ProgramCounter is
  component D_FF
    port(
      D : in std_logic;
      Res_D_FF : in std_logic;
      EN_D_FF :in std_logic;
      Clk : in std_logic;
      Q  : out std_logic;
      Qbar : out std_logic);
  end component;

begin
  D_FF0 : D_FF
    port map(
      D => D(0),
      EN_D_FF =>EN,
      Res_D_FF => Reset_button,
      Clk => Clk,
      Q => Q(0));

  D_FF1 : D_FF
    port map(
```

```
        D => D(1),
        EN_D_FF =>EN,
        Res_D_FF => Reset_button,
        Clk => Clk,
        Q => Q(1));

   D_FF2 : D_FF
      port map(
        D => D(2),
        EN_D_FF =>EN,
        Res_D_FF => Reset_button,
        Clk => Clk,
        Q => Q(2));



end Behavioral;
```

- Simulation Source

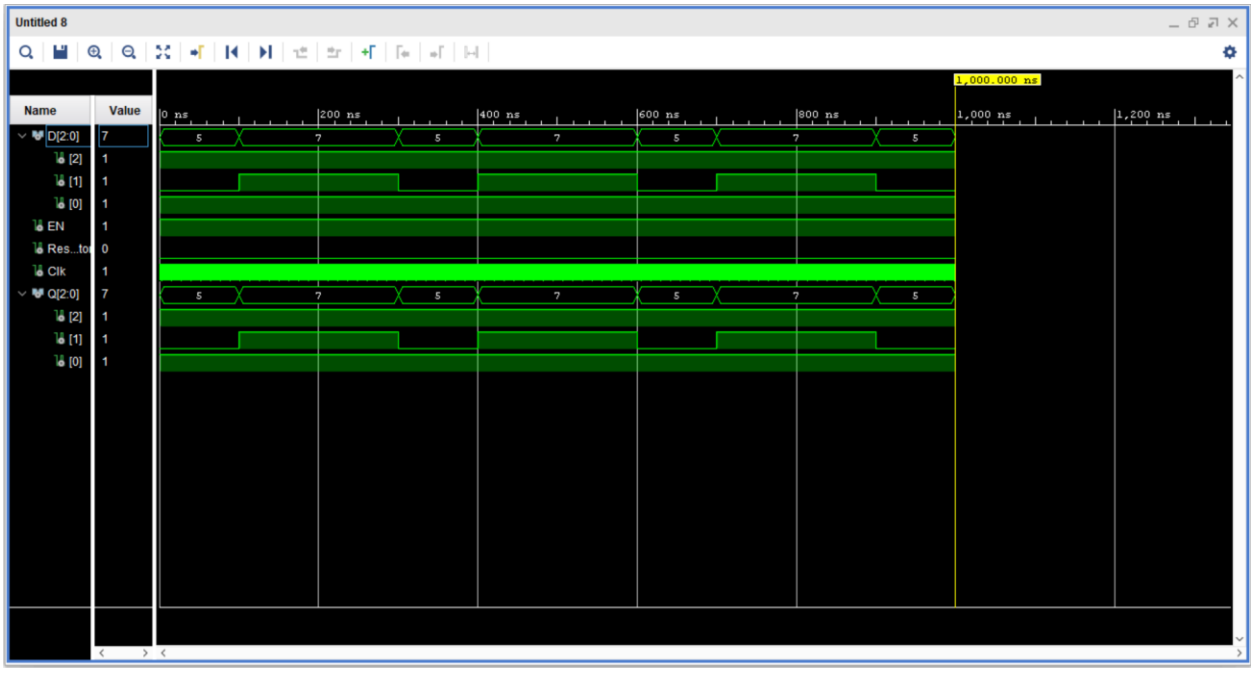
```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date: 04/10/2024 10:37:03 PM
-- Design Name:
-- Module Name: TB_ProgramCounter - Behavioral
-- Project Name:
-- Target Devices:
-- Tool Versions:
-- Description:
--
-- Dependencies:
--
```

```
-- 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 TB_ProgramCounter is
--  Port ( );
end TB_ProgramCounter;

architecture Behavioral of TB_ProgramCounter is
component ProgramCounter
  port (
    D : in STD_LOGIC_VECTOR(2 downto 0);
    EN : in STD_LOGIC;
    Reset_button : in STD_LOGIC;
    Clk : in STD_LOGIC;
    Q : out STD_LOGIC_VECTOR(2 downto 0)
  );
end component;
```

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

signal D : STD_LOGIC_VECTOR(2 downto 0);
signal EN : STD_LOGIC:='1';
signal Reset_button : STD_LOGIC := '0'; -- Ensure initial value is set
signal Clk : STD_LOGIC:='0';
signal Q : STD_LOGIC_VECTOR(2 downto 0);

begin
UUT:ProgramCounter  PORT MAP(
D => D,
Reset_button => Reset_button,
Clk => Clk,
Q => Q,
EN => EN
);

process begin
Clk <= NOT(Clk);
wait for 2ns;
end process;
process

--Index(Lahiru) 110 101 110 111 111 101

begin
D <= "101";
```

```
wait for 100 ns;
D <= "111";
wait for 100 ns;
D <= "111";
wait for 100 ns;
D <= "110";



end process;

end Behavioral;
```

- Timing Diagram



Program Rom

- Design Source

```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date: 04/07/2024 04:39:50 PM
-- Design Name:
-- Module Name: ProgramRom - 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.NUMERIC_STD.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
```

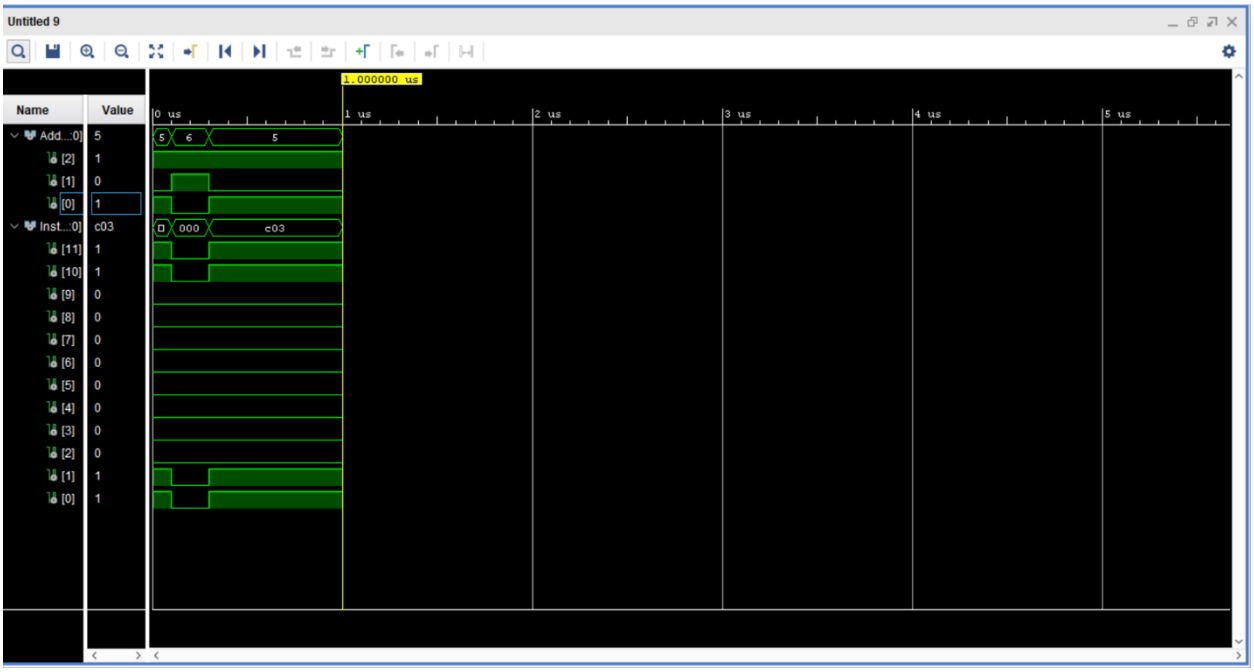
```
-- any Xilinx leaf cells in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity ProgramRom is
   Port ( Address : in STD_LOGIC_VECTOR (2 downto 0);
          Instruction : out STD_LOGIC_VECTOR (11 downto 0));
end ProgramRom;

architecture Behavioral of ProgramRom is

   type rom_type is array (0 to 7) of std_logic_vector(11 downto 0);
      signal Program_ROM : rom_type := (
                  "100010001010", --
                  "100100000001",
                  "010100000000",
                  "000010100000",
                  "110010000111",
                  "110000000011",
                  "000000000000",
                  "ZZZZZZZZZZZZ"
        );

begin

Instruction <= Program_ROM(to_integer(unsigned(Address)));


end Behavioral;
```

- Simulation Source

```
-- Company:
-- Engineer:
--
-- Create Date: 04/10/2024 10:29:12 PM
-- Design Name:
-- Module Name: TB_ProgramROM - 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 TB_ProgramROM is
```

```
--  Port ( );
end TB_ProgramROM;

architecture Behavioral of TB_ProgramROM is
component  ProgramRom
Port ( Address : in STD_LOGIC_VECTOR (2 downto 0);
     Instruction : out STD_LOGIC_VECTOR (11 downto 0));
end component;

signal Address : STD_LOGIC_VECTOR(2 downto 0);
signal Instruction : STD_LOGIC_VECTOR(11 downto 0);


begin
UUT: ProgramRom port map(
Address=>Address,
Instruction =>Instruction);
process
begin
--Index(Gamith) 110 101 101 110 110 101
   Address <= "101";
   wait for 100 ns;
   Address <= "110";
   wait for 100 ns;
   Address <= "110";
   wait for 100 ns;
   Address <= "101";

   wait;

end process;

end Behavioral;
```

- Timing Diagram



Register Bank

- Design Source

-- Company:

-- Engineer:

--

-- Create Date: 04/10/2024 04:31:09 PM

-- Design Name:

-- Module Name: Reg_Bank - 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 Reg_Bank is
   Port ( Reg_enable : in STD_LOGIC_VECTOR (2 downto 0);
          Mux_2_1_4bit_result : in STD_LOGIC_VECTOR (3 downto 0);
          Reset_button:in STD_LOGIC;
          Clk : in STD_LOGIC;
          Reg_out0 : out STD_LOGIC_VECTOR (3 downto 0);
          Reg_out1 : out STD_LOGIC_VECTOR (3 downto 0);
          Reg_out2 : out STD_LOGIC_VECTOR (3 downto 0);
          Reg_out3 : out STD_LOGIC_VECTOR (3 downto 0);
          Reg_out4 : out STD_LOGIC_VECTOR (3 downto 0);
          Reg_out5 : out STD_LOGIC_VECTOR (3 downto 0);
          Reg_out6 : out STD_LOGIC_VECTOR (3 downto 0);
          Reg_out7 : out STD_LOGIC_VECTOR (3 downto 0));
end Reg_Bank;
```

```
architecture Behavioral of Reg_Bank is

component Reg
Port (    D : in STD_LOGIC_VECTOR (3 downto 0);
          EN : in STD_LOGIC;
          Res_Reg : in STD_LOGIC;
          Clk : in STD_LOGIC;
          Q : out STD_LOGIC_VECTOR (3 downto 0));
end component;

component Decoder_3_to_8 port(
     I : in STD_LOGIC_VECTOR(2 downto 0);
     EN : in STD_LOGIC;
     Y : out STD_LOGIC_VECTOR (7 downto 0));
end component;



signal decorder_out: STD_LOGIC_VECTOR (7 downto 0);
begin

Decoder_3_to_8_0: Decoder_3_to_8
port map (
   I => Reg_enable,
   EN =>'1',
   Y =>decorder_out
);

Reg_0:Reg
port map (
D => Mux_2_1_4bit_result,
Res_Reg => Reset_button,
```

```
EN => '0',
Clk => Clk,
Q => Reg_out0);

Reg_1:Reg
port map (
D => Mux_2_1_4bit_result,
Res_Reg => Reset_button,
EN => decorder_out(1),
Clk => Clk,
Q => Reg_out1);

Reg_2:Reg
port map (
D => Mux_2_1_4bit_result,
Res_Reg => Reset_button,
EN => decorder_out(2),
Clk => Clk,
Q => Reg_out2);

Reg_3:Reg
port map (
D => Mux_2_1_4bit_result,
Res_Reg => Reset_button,
EN => decorder_out(3),
Clk => Clk,
Q => Reg_out3);

Reg_4:Reg
port map (
D => Mux_2_1_4bit_result,
Res_Reg => Reset_button,
EN =>decorder_out(4),
```

```
Clk => Clk,
Q => Reg_out4);

Reg_5:Reg
port map (
D => Mux_2_1_4bit_result,
Res_Reg => Reset_button,
EN =>decorder_out(5),
Clk => Clk,
Q => Reg_out5);

Reg_6:Reg
port map (
D => Mux_2_1_4bit_result,
Res_Reg => Reset_button,
EN =>decorder_out(6),
Clk => Clk,
Q => Reg_out6);

Reg_7:Reg
port map (
D => Mux_2_1_4bit_result,
Res_Reg => Reset_button,
EN => '1',
Clk => Clk,
Q => Reg_out7);

end Behavioral;
```

- Simulation Source

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

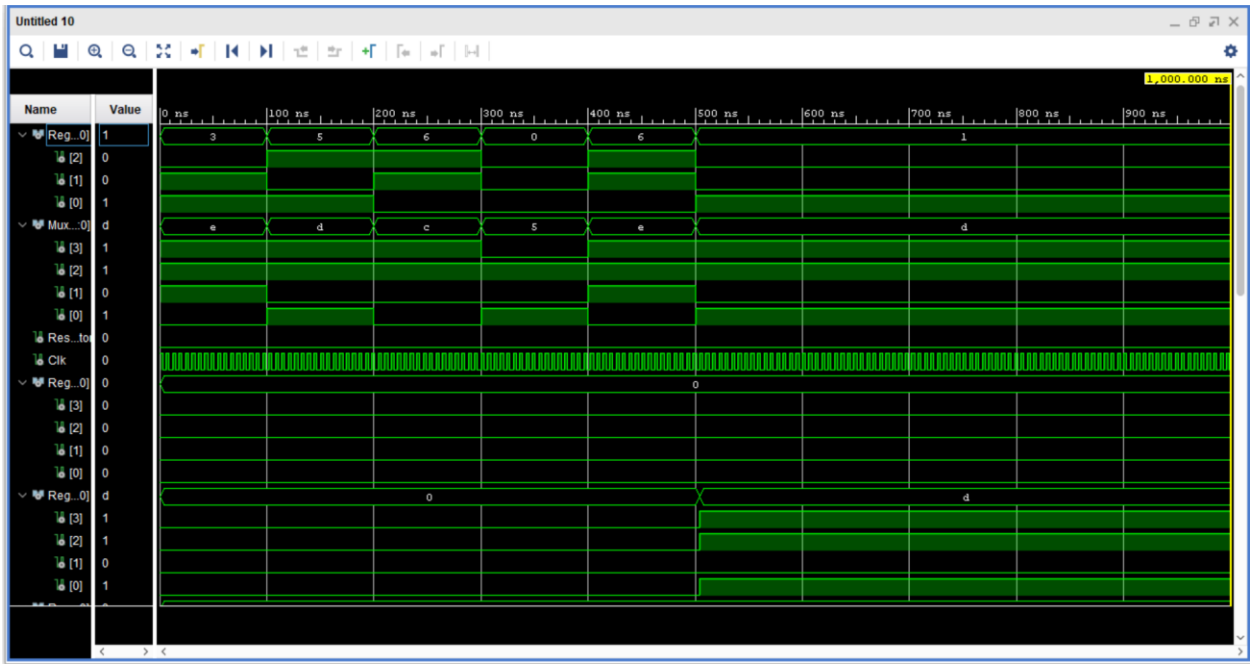
```
--
-- Create Date: 04/10/2024 05:35:07 PM
-- Design Name:
-- Module Name: TB_Reg_Bank - Behavioral
-- Project Name:
-- Target Devices:
-- Tool Versions:
-- Description:
--
-- Dependencies:
--
-- Revision:
-- Revision 0.01 - File Created
-- Additional Comments:
--
----------------------------------------------------------------------------------


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 TB_Reg_Bank is
--  Port ( );
end TB_Reg_Bank;
```

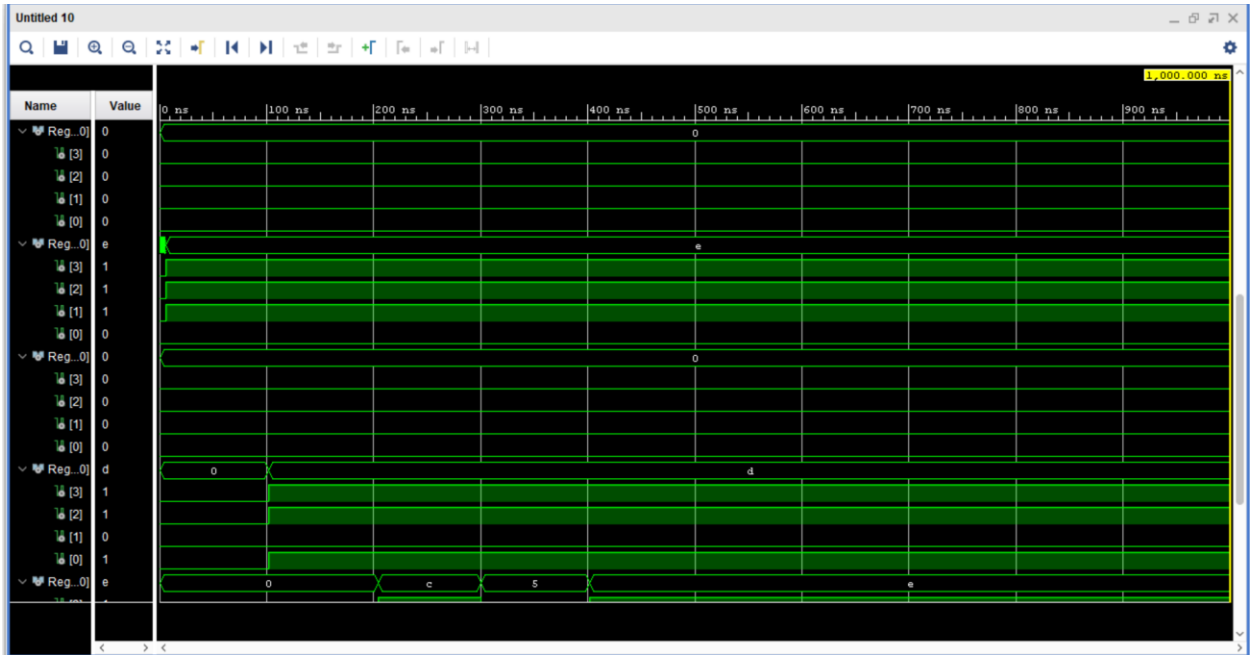
```
architecture Behavioral of TB_Reg_Bank is
component Reg_Bank
Port (    Reg_enable : in STD_LOGIC_VECTOR (2 downto 0);
       Mux_2_1_4bit_result : in STD_LOGIC_VECTOR (3 downto 0);
       Reset_button:in STD_LOGIC;
       Clk : in STD_LOGIC;
       Reg_out0 : out STD_LOGIC_VECTOR (3 downto 0);
       Reg_out1 : out STD_LOGIC_VECTOR (3 downto 0);
       Reg_out2 : out STD_LOGIC_VECTOR (3 downto 0);
       Reg_out3 : out STD_LOGIC_VECTOR (3 downto 0);
       Reg_out4 : out STD_LOGIC_VECTOR (3 downto 0);
       Reg_out5 : out STD_LOGIC_VECTOR (3 downto 0);
       Reg_out6 : out STD_LOGIC_VECTOR (3 downto 0);
       Reg_out7 : out STD_LOGIC_VECTOR (3 downto 0));

end component;

component Slow_Clk
     port (
       Clk_in : in STD_LOGIC;
       Clk_out : out STD_LOGIC

     );
end component;

 signal Reg_enable : STD_LOGIC_VECTOR(2 downto 0);
 signal Mux_2_1_4bit_result :  STD_LOGIC_VECTOR (3 downto 0);
 signal Reset_button: STD_LOGIC:='0';
 signal Clk: STD_LOGIC:='0';
 signal Reg_out0,Reg_out1,Reg_out2,Reg_out3,Reg_out4,Reg_out5,Reg_out6,Reg_out7 :
STD_LOGIC_VECTOR (3 downto 0);
 signal EN :  STD_LOGIC;
```

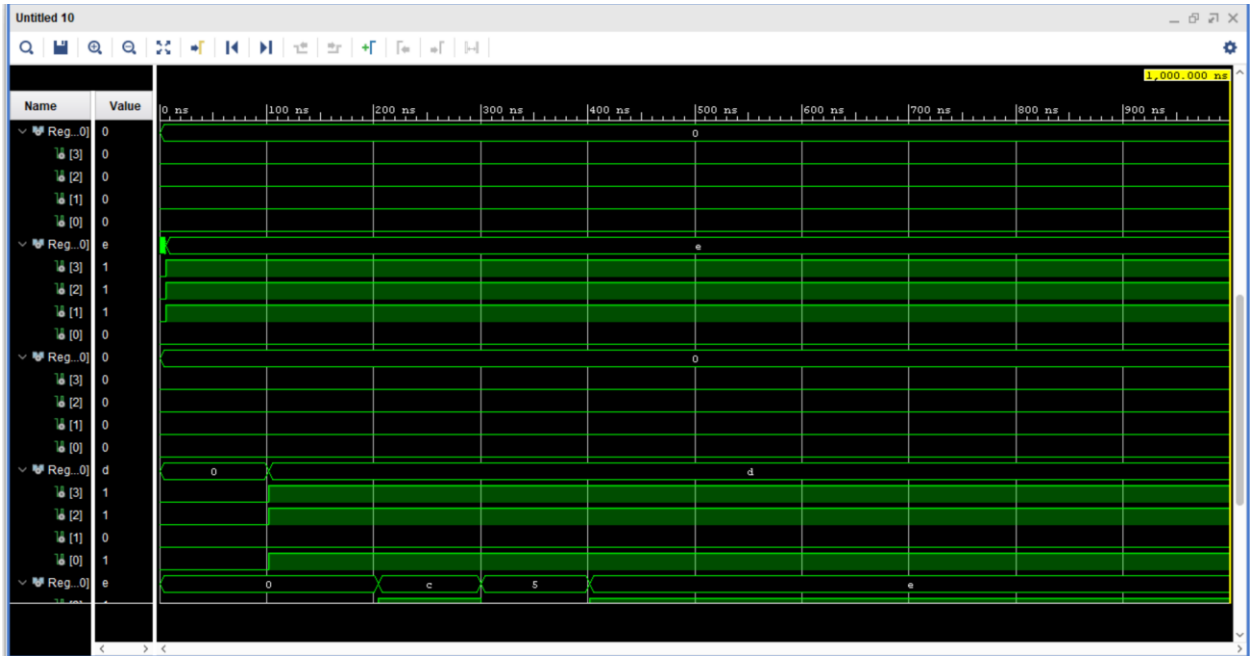
```
begin
UUT: Reg_Bank port map(

Reg_enable => Reg_enable,
Mux_2_1_4bit_result =>Mux_2_1_4bit_result,
Reset_button => Reset_button,
Clk => Clk,
Reg_out0 => Reg_out0,
Reg_out1 => Reg_out1,
Reg_out2 => Reg_out2,
Reg_out3 => Reg_out3,
Reg_out4 => Reg_out4,
Reg_out5 => Reg_out5,
Reg_out6 => Reg_out6,
Reg_out7 => Reg_out7

);

process begin
   Clk <= NOT(Clk);
   wait for 3ns;
end process;

--Index(Nimash) 11 0101 1100 1101 1110
process
begin
   Mux_2_1_4bit_result <="1110";

   Reg_enable <="011";

   wait for 100 ns;
   Mux_2_1_4bit_result <="1101";
```

```
  Reg_enable <="101";

  wait for 100 ns;
  Mux_2_1_4bit_result <="1100";

  Reg_enable <="110";

  wait for 100 ns;
  Mux_2_1_4bit_result <="0101";

  Reg_enable <="000";



  wait for 100 ns;
  Mux_2_1_4bit_result <="1110";

  Reg_enable <="110";
  wait for 100 ns;
  Mux_2_1_4bit_result <="1101";

  Reg_enable <="001";
  wait;




end process;

end Behavioral;
```

- Timing Diagrams


Untitled 10
Name
Value
1,000.000 ns
0 ns
100 ns
200 ns
300 ns
400 ns
500 ns
600 ns
700 ns
800 ns
900 ns
Reg...0]
Mux...0]
Res...to
Clk
Reg...0]
Reg...0]
Reg...0]


Untitled 10
Name
Value
1,000.000 ns
0 ns
100 ns
200 ns
300 ns
400 ns
500 ns
600 ns
700 ns
800 ns
900 ns
Reg...0]
Reg...0]
Reg...0]
Reg...0]



Instruction Decoder

- Design Source

```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date: 04/11/2024 07:48:04 PM
-- Design Name:
-- Module Name: Instruction_Decoder - 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 Instruction_Decoder is
  Port ( Instruction_input : in STD_LOGIC_VECTOR (11 downto 0);
      Register_check_for_jump : in STD_LOGIC_VECTOR (3 downto 0);
      Register_enable_output : out STD_LOGIC_VECTOR (2 downto 0);
      Load_select_output : out STD_LOGIC; -- load select is 0 choose store way else choose load immediate value way
      Immediate_value_output : out STD_LOGIC_VECTOR (3 downto 0);
      Register_select_output_0 : out STD_LOGIC_VECTOR (2 downto 0);
      Register_select_output_1 : out STD_LOGIC_VECTOR (2 downto 0);
      Add_Sub_select_output : out STD_LOGIC;
      Jump_flag : out STD_LOGIC; ---if jump flag is =0 : select normal sequence, else want to jump
      Address_to_jump : out STD_LOGIC_VECTOR (2 downto 0));
end Instruction_Decoder;

architecture Behavioral of Instruction_Decoder is
begin

process (Instruction_input,Register_check_for_jump)begin

 if (Instruction_input(11 downto 10) = "00") then ---- ADD instruction
   Register_select_output_0 <= Instruction_input(9 downto 7);
   Register_select_output_1 <= Instruction_input(6 downto 4);
   Load_select_output <='0';--Instruction_input(11);
   Add_Sub_select_output <= '0';
   Jump_flag <='0';
   Register_enable_output <= Instruction_input(9 downto 7);
   Immediate_value_output <="ZZZZ";
   Address_to_jump <="ZZZ";

 elsif(Instruction_input(11 downto 10) = "01") then ------ NEG instruction
   Register_enable_output <= Instruction_input(9 downto 7);
   Register_select_output_0 <= Instruction_input(6 downto 4);
   Register_select_output_1 <= Instruction_input(9 downto 7);
   Jump_flag <='0';
   Load_select_output <='0';
   Add_Sub_select_output <='1';
 Immediate_value_output <="ZZZZ";
 Address_to_jump <="ZZZ";

 elsif(Instruction_input(11 downto 10) = "10") then ------ MOV instruction
   Register_enable_output <= Instruction_input(9 downto 7);
   Jump_flag <='0';
```

```
    Load_select_output <='1';--Instruction_input(11);
    Immediate_value_output <=Instruction_input(3 downto 0);
    Address_to_jump <="ZZZ";
    Add_Sub_select_output <='Z';
    Register_select_output_0 <="ZZZ";
    Register_select_output_1 <="ZZZ";

  elsif(Instruction_input(11 downto 10) = "11")  then ------ JZR instruction
    Register_select_output_0 <=  Instruction_input(9 downto 7);
    Register_select_output_1 <="ZZZ";
    Add_Sub_select_output <='Z';
    Load_select_output <='Z';
    Register_enable_output <="ZZZ";
    if (Register_check_for_jump="0000")then
      Jump_flag <='1';
      Address_to_jump <= Instruction_input(2 downto 0);
    end if;
  elsif (Instruction_input(11 downto 0) = "ZZZZZZZZ") then
    Register_enable_output  <="ZZZ";
    Address_to_jump<="ZZZ" ;
    Jump_flag<='Z';
     Register_select_output_1 <="ZZZ";
        Add_Sub_select_output <='Z';
        Load_select_output <='Z';
        Register_select_output_0<="ZZZ";
        Immediate_value_output <="ZZZZ";

  end if;

end process;

end Behavioral;
```

- Simulation Source

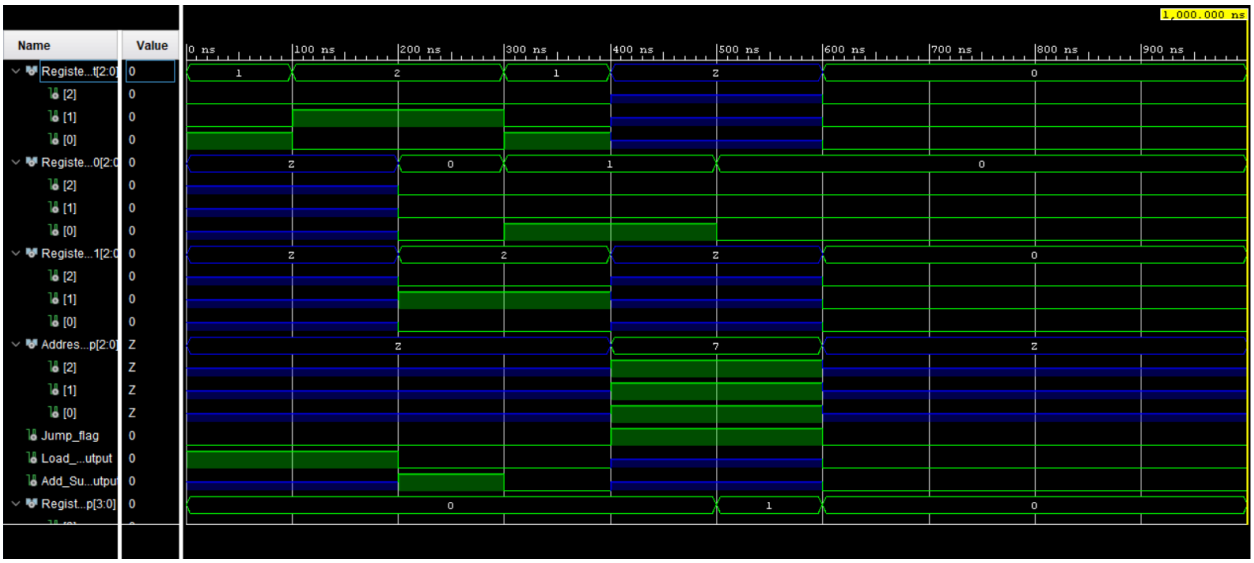
```
----------------------------------------------------------------------------------

-- Company:

-- Engineer:

--

-- Create Date: 04/11/2024 08:20:33 PM

-- Design Name:

-- Module Name: TB_Instruction_Decoder - Behavioral

-- Project Name:

-- Target Devices:

-- Tool Versions:
```

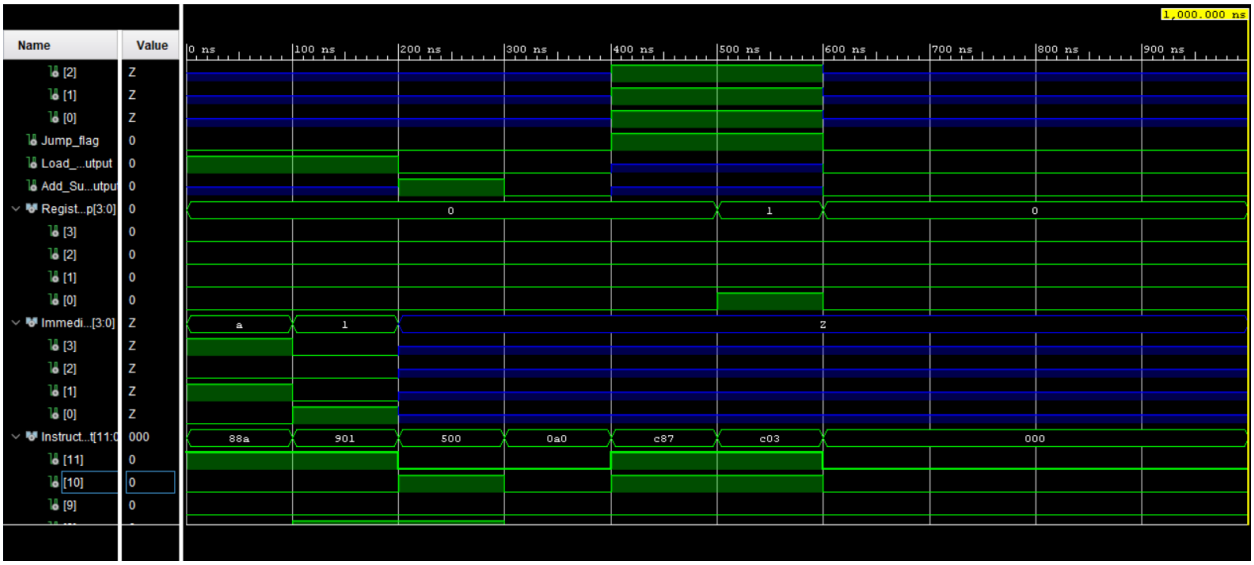
```
-- Description:
--
-- Dependencies:
--
-- Revision:
-- Revision 0.01 - File Created
-- Additional Comments:
--
----------------------------------------------------------------------------------


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 TB_Instruction_Decoder is
--  Port ( );
end TB_Instruction_Decoder;

architecture Behavioral of TB_Instruction_Decoder is
COMPONENT Instruction_Decoder
  Port(
    Instruction_input : in STD_LOGIC_VECTOR (11 downto 0);
    Register_check_for_jump : in STD_LOGIC_VECTOR (3 downto 0);
    Register_enable_output : out STD_LOGIC_VECTOR (2 downto 0);
```

```
    Load_select_output : out STD_LOGIC; -- load select is =0 choose store way else choose load immediate value  way
    Immediate_value_output : out STD_LOGIC_VECTOR (3 downto 0);
    Register_select_output_0 : out STD_LOGIC_VECTOR (2 downto 0);
    Register_select_output_1 : out STD_LOGIC_VECTOR (2 downto 0);
    Add_Sub_select_output : out STD_LOGIC;
    Jump_flag : out STD_LOGIC; ---if jump flag is =0 : select normal sequence, else want  to jump
    Address_to_jump : out STD_LOGIC_VECTOR (2 downto 0));

END COMPONENT;
SIGNAL Register_enable_output,Register_select_output_0,Register_select_output_1,Address_to_jump : std_logic_vector(2 downto 0);
SIGNAL Jump_flag,Load_select_output,Add_Sub_select_output: std_logic;
SIGNAL Register_check_for_jump,Immediate_value_output: std_logic_vector( 3 downto 0);
SIGNAL Instruction_input : std_logic_vector(11 downto 0);

begin

uut: Instruction_Decoder PORT MAP (
Instruction_input =>Instruction_input,
Register_check_for_jump => Register_check_for_jump,
Register_enable_output => Register_enable_output,
Load_select_output => Load_select_output,
Immediate_value_output => Immediate_value_output,
Register_select_output_0 =>Register_select_output_0,
Register_select_output_1 => Register_select_output_1,
Add_Sub_select_output => Add_Sub_select_output,
Jump_flag  => Jump_flag,
Address_to_jump => Address_to_jump );
process
-----  index to binary (220382---  110 101 110 011  011 110)
  begin
  Instruction_input <=  "100010001010";
  Register_check_for_jump <= "0000";
```

```
    wait for 100 ns;
    Instruction_input <=  "100100000001";
    Register_check_for_jump <= "0000";


    wait for 100 ns;
    Instruction_input <=  "010100000000";
    Register_check_for_jump <= "0000";


    wait for 100 ns;
    Instruction_input <=  "000010100000";
    Register_check_for_jump <= "0000";

    wait for 100 ns;
    Instruction_input <=  "110010000111";
    Register_check_for_jump <= "0000";

    wait for 100 ns;
    Instruction_input <=  "110000000011";
    Register_check_for_jump <= "0001";

    wait for 100 ns;
    Instruction_input <=  "000000000000";
    Register_check_for_jump <= "0000";


    wait;
    end process;

end Behavioral;
```

- Timing Diagrams


Name
Value
1,000.000 ns
0 ns
100 ns
200 ns
300 ns
400 ns
500 ns
600 ns
700 ns
800 ns
900 ns
Registe...t[2:0]
[2]
[1]
[0]
Registe...0[2:0]
Registe...1[2:0]
Addres...p[2:0]
Jump_flag
Load_...utput
Add_Su...utput
Regist...p[3:0]


Name
Value
1,000.000 ns
0 ns
100 ns
200 ns
300 ns
400 ns
500 ns
600 ns
700 ns
800 ns
900 ns
Jump_flag
Load_...utput
Add_Su...utput
Regist...p[3:0]
Immedi...[3:0]
Instruct...t[11:0]
[11]
[10]
[9]


[0]
Immedi...[3:0]
[3]
[2]
[1]
[0]
Instruct...t[11:0]
[11]
[10]
[9]
[8]
[7]
[6]
[5]
[4]
[3]
[2]
[1]
[0]

Slow Clock

- Design Source

```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date: 03/02/2024 08:21:05 PM
-- Design Name:
-- Module Name: Slow_Clk - 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 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;
```

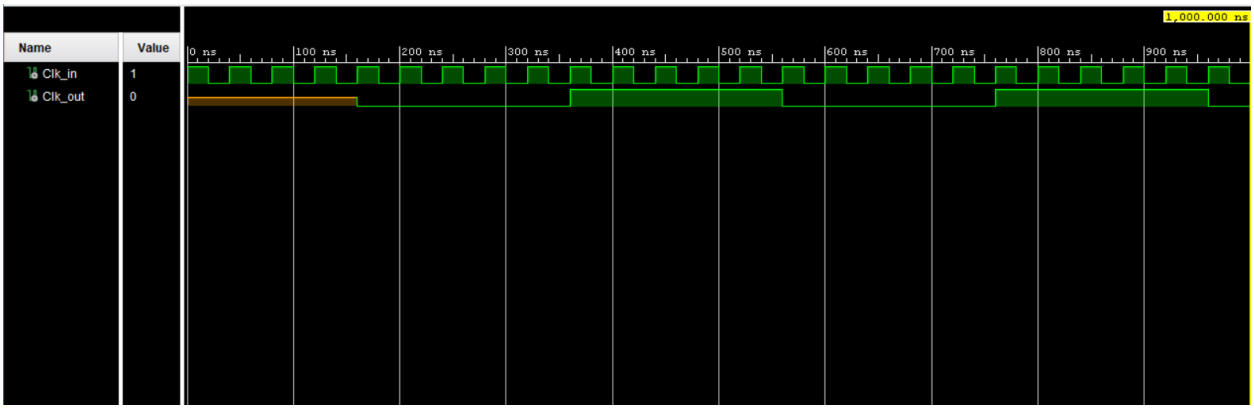
- Simulation Source

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

```
--
-- Create Date: 03/02/2024 08:30:45 PM
-- Design Name:
-- Module Name: Slow_Clk_Sim - 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 Slow_Clk_Sim is
--  Port ( );
end Slow_Clk_Sim;
```

```
architecture Behavioral of Slow_Clk_Sim is

COMPONENT Slow_Clk
  PORT ( Clk_in : in STD_LOGIC:='0';
       Clk_out : out STD_LOGIC);
END COMPONENT;
  SIGNAL Clk_in : STD_LOGIC := '0';
  SIGNAL Clk_out : STD_LOGIC;
begin
  UUT : Slow_Clk PORT MAP(
  Clk_in => Clk_in,
  Clk_out => Clk_out
  );
process begin
  Clk_in <= not Clk_in;
  wait for 20 ns;

end process;



end Behavioral;
```

- Timing Diagram



Lookup Table

- Design Source

-- Company:

-- Engineer:

--

-- Create Date: 03/21/2024 12:34:31 PM

-- Design Name:

-- Module Name: LUT_16_7 - 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.numeric_std.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 LUT_16_7 is
   Port ( address : in STD_LOGIC_VECTOR (3 downto 0);
          data : out STD_LOGIC_VECTOR (6 downto 0));
end LUT_16_7;

architecture Behavioral of LUT_16_7 is
type rom_type is array (0 to 15) of std_logic_vector(6 downto 0);
signal sevenSegment_ROM : rom_type := (
 "1000000", -- 0
 "1111001", -- 1
 "0100100", -- 2
 "0110000", -- 3
 "0011001", -- 4
 "0010010", -- 5
 "0000010", -- 6
 "1111000", -- 7
```

```
"0000000", -- 8
"0010000", -- 9
"0001000", -- a
"0000011", -- b
"1000110", -- c
"0100001", -- d
"0000110", -- e
"0001110"  -- f
);

begin
data <= sevenSegment_ROM(to_integer(unsigned(address)));


end Behavioral;
```

Nano_processor

- Design Source

```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date: 04/13/2024 07:27:14 PM
-- Design Name:
-- Module Name: Nano__Processor - 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 Nano_Processor is
   Port ( Reset_Push_Button:in STD_LOGIC;
          Clk:in STD_LOGIC;
          LED_out : out STD_LOGIC_VECTOR (3 downto 0);
          LUT_out:out STD_LOGIC_VECTOR(6 downto 0);
          Zero_Flag : out STD_LOGIC;
          Anode:out STD_LOGIC_VECTOR(3 downto 0);
          --pc:out STD_LOGIC_VECTOR(2 downto 0);
         -- Reg_1_out: out STD_LOGIC_VECTOR (3 downto 0);
         -- Reg_2_out: out STD_LOGIC_VECTOR (3 downto 0);
          Overflow_Flag : out STD_LOGIC);
```

```
end Nano_Processor;

architecture Behavioral of Nano_Processor is

component LUT_16_7
   Port ( address : in STD_LOGIC_VECTOR (3 downto 0);
          data : out STD_LOGIC_VECTOR (6 downto 0));
end component;

component Reg_Bank
   Port (
     Reg_enable : in STD_LOGIC_VECTOR (2 downto 0);
     Mux_2_1_4bit_result : in STD_LOGIC_VECTOR (3 downto 0);
     Reset_button:in STD_LOGIC;
     Clk : in STD_LOGIC;
     Reg_out0 : out STD_LOGIC_VECTOR (3 downto 0);
     Reg_out1 : out STD_LOGIC_VECTOR (3 downto 0);
     Reg_out2 : out STD_LOGIC_VECTOR (3 downto 0);
     Reg_out3 : out STD_LOGIC_VECTOR (3 downto 0);
     Reg_out4 : out STD_LOGIC_VECTOR (3 downto 0);
     Reg_out5 : out STD_LOGIC_VECTOR (3 downto 0);
     Reg_out6 : out STD_LOGIC_VECTOR (3 downto 0);
     Reg_out7 : out STD_LOGIC_VECTOR (3 downto 0));

end component;

component Mux_8_to_1
   Port ( S  : IN STD_LOGIC_VECTOR (2 downto 0);
     D0 : IN STD_LOGIC_VECTOR (3 downto 0);
     D1 : IN STD_LOGIC_VECTOR (3 downto 0);
     D2 : IN STD_LOGIC_VECTOR (3 downto 0);
     D3 : IN STD_LOGIC_VECTOR (3 downto 0);
```

```
    D4 : IN STD_LOGIC_VECTOR (3 downto 0);
    D5 : IN STD_LOGIC_VECTOR (3 downto 0);
    D6 : IN STD_LOGIC_VECTOR (3 downto 0);
    D7 : IN STD_LOGIC_VECTOR (3 downto 0);
    EN : IN STD_LOGIC;
    Y  : OUT STD_LOGIC_VECTOR (3 downto 0));
end component;

component ADD_SUB_unit
  Port (
    INPUT_0: in STD_LOGIC_VECTOR (3 downto 0);
    INPUT_1: in STD_LOGIC_VECTOR (3 downto 0);
    AddSub_Select : in STD_LOGIC;
    OUTPUT:out STD_LOGIC_VECTOR(3 downto 0);
    Zero : out STD_LOGIC;
    OverFlow : out STD_LOGIC);
end component;

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

component Mux_2_to_1_4bit
Port ( Load_Select : in STD_LOGIC; --Load_Select=0 (D0/stored_value(unit_result)) ---Load_Select=1
(D1/immediate_value)
    Mux_2_to_1_4bit_in_0 : in STD_LOGIC_VECTOR (3 downto 0);
    Mux_2_to_1_4bit_in_1 : in STD_LOGIC_VECTOR (3 downto 0);
    EN : in STD_LOGIC;
    Y : out STD_LOGIC_VECTOR (3 downto 0));
end component;
```

```
component Instruction_Decoder
Port ( Instruction_input : in STD_LOGIC_VECTOR (11 downto 0);
    Register_check_for_jump : in STD_LOGIC_VECTOR (3 downto 0);
    Register_enable_output : out STD_LOGIC_VECTOR(2 downto  0); -- load select is  0 choose store way else choose load immediate value  way
    Immediate_value_output : out STD_LOGIC_VECTOR (3 downto 0);
    Load_select_output : out STD_LOGIC;
    Register_select_output_0 : out STD_LOGIC_VECTOR (2 downto 0);
    Register_select_output_1 : out STD_LOGIC_VECTOR (2 downto 0);
    Add_Sub_select_output : out STD_LOGIC;
    Jump_flag : out STD_LOGIC; ---if jump flag is =0 : select normal sequence, else want  to jump
    Address_to_jump : out STD_LOGIC_VECTOR (2 downto 0));
end component;

component ProgramRom
 Port ( Address : in STD_LOGIC_VECTOR (2 downto 0);
      Instruction : out STD_LOGIC_VECTOR (11 downto 0));
end component;


component ProgramCounter
Port ( Reset_button : in STD_LOGIC;
    EN:in STD_LOGIC;
    D : in STD_LOGIC_VECTOR (2 downto 0);
    Clk : in STD_LOGIC;
    Q : out STD_LOGIC_VECTOR (2 downto 0));
end component;

component  Mux_2_to_1_3bit
Port ( S : in STD_LOGIC;
    D0 : in STD_LOGIC_VECTOR (2 downto 0);
    D1 : in STD_LOGIC_VECTOR (2 downto 0);
    EN : in STD_LOGIC;
```

```
    Y : out STD_LOGIC_VECTOR (2 downto 0));
 end component;


component RCA_3_bit_Adder
Port (
  A0 : in STD_LOGIC;
  A1 : in STD_LOGIC;
  A2 : in STD_LOGIC;
  B0 : in STD_LOGIC;
  B1 : in STD_LOGIC;
  B2 : in STD_LOGIC;
  C_in : in STD_LOGIC;
  S0 : out STD_LOGIC;
  S1 : out STD_LOGIC;
  S2 : out STD_LOGIC;
  C_out : out STD_LOGIC);
end component;



signal Clk_slow,select_operation,load_select,jump_or_not:STD_LOGIC;
signal Instruction_Bus_Input:STD_LOGIC_VECTOR(11 downto 0);
signal
reg_select_0,reg_select_1,reg_enb,Mux_2_to_1_3bit_output,RCA_3_bit_Adder_output,Jump_Address,Progra
mCounter_output:STD_LOGIC_VECTOR(2 downto 0);
signal
reg_0_output,reg_1_output,reg_2_output,reg_3_output,reg_4_output,reg_5_output,reg_6_output,reg_7_outp
ut:STD_LOGIC_VECTOR (3 downto 0);
signal
ALU_MUX_output_0,ALU_MUX_output_1,AU_out,Choose_Register_Store_value,Immediate_value:STD_LOGI
C_VECTOR (3 downto 0);
signal lut_output:std_LOGIC_VECTOR(6 downto 0);


begin
```

```
LUT_16_7_0:LUT_16_7
port map(
address  =>reg_7_output,
data =>lut_output
);

LUT_out <=lut_output;

Mux_2_to_1_3bit_0:Mux_2_to_1_3bit
port map(
S =>jump_or_not,
D0 =>RCA_3_bit_Adder_output,
D1 =>Jump_Address,
EN =>'1',
Y =>Mux_2_to_1_3bit_output
);

RCA_3_bit_Adder_0:RCA_3_bit_Adder
port map(
A0 =>ProgramCounter_output(0),
A1 =>ProgramCounter_output(1),
A2 =>ProgramCounter_output(2),
B0 =>'1',
B1 =>'0',
B2 =>'0',
C_in =>'0',
S0 =>RCA_3_bit_Adder_output(0),
S1 =>RCA_3_bit_Adder_output(1),
S2 =>RCA_3_bit_Adder_output(2)
);
```

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


ProgramRom_0:ProgramRom
port map(
Address =>ProgramCounter_output,
Instruction =>Instruction_Bus_Input
);


ProgramCounter_0:ProgramCounter
port map(
Reset_button=>Reset_Push_Button,
EN =>'1',
D=>Mux_2_to_1_3bit_output,
Clk=>Clk_slow,
Q =>ProgramCounter_output
);

--pc <=ProgramCounter_output;
Instruction_Decoder_0:Instruction_Decoder
port map(
Instruction_input =>Instruction_Bus_Input,
Register_check_for_jump =>ALU_MUX_output_0,
Register_enable_output=>reg_enb,
Load_select_output =>load_select,
Immediate_value_output =>Immediate_value,
Register_select_output_0 =>reg_select_0,
Register_select_output_1 =>reg_select_1,
Add_Sub_select_output =>select_operation,
```

```
Jump_flag =>jump_or_not,
Address_to_jump =>Jump_Address
);


Reg_Bank_0:Reg_Bank
port map(
Reg_enable=>reg_enb,
Mux_2_1_4bit_result => Choose_Register_Store_value,
Reset_button =>Reset_Push_Button,
Clk => Clk_slow,
Reg_out0=>reg_0_output,
Reg_out1=>reg_1_output,
Reg_out2=>reg_2_output,
Reg_out3=>reg_3_output,
Reg_out4=>reg_4_output,
Reg_out5=>reg_5_output,
Reg_out6=>reg_6_output,
Reg_out7=>reg_7_output
);

LED_out <=reg_7_output;
--Reg_1_out <=reg_1_output;
--Reg_2_out <=reg_2_output;

Anode <="1110";

Mux_8_to_1_0:Mux_8_to_1
port map(
S =>reg_select_0,
D0 => reg_0_output,
D1 => reg_1_output,
D2 => reg_2_output,
```

```
D3 => reg_3_output,
D4 => reg_4_output,
D5 => reg_5_output,
D6 => reg_6_output,
D7 => reg_7_output,
EN => '1',
y =>ALU_MUX_output_0
);

Mux_8_to_1_1:Mux_8_to_1
port map(
S =>reg_select_1,
D0 => reg_0_output,
D1 => reg_1_output,
D2 => reg_2_output,
D3 => reg_3_output,
D4 => reg_4_output,
D5 => reg_5_output,
D6 => reg_6_output,
D7 => reg_7_output,
EN => '1',
y =>ALU_MUX_output_1
);

ADD_SUB_unit_0:ADD_SUB_unit
Port map(
INPUT_0 =>ALU_MUX_output_0,
INPUT_1 =>ALU_MUX_output_1,
AddSub_Select =>select_operation,
OUTPUT =>AU_out,
Zero =>Zero_flag,
OverFlow =>Overflow_Flag
);
```

```
Mux_2_to_1_4bit_0:Mux_2_to_1_4bit
port map(
Load_Select =>load_select,
Mux_2_to_1_4bit_in_0=>AU_out,
Mux_2_to_1_4bit_in_1 =>Immediate_value,
EN =>'1',
Y =>Choose_Register_Store_value
);



end Behavioral;
```

- Simulation File

```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date: 04/13/2024 09:49:25 PM
-- Design Name:
-- Module Name: TB_NanoProcessor - Behavioral
-- Project Name:
-- Target Devices:
-- Tool Versions:
-- Description:
--
-- Dependencies:
--
-- Revision:
-- Revision 0.01 - File Created
-- Additional Comments:
```

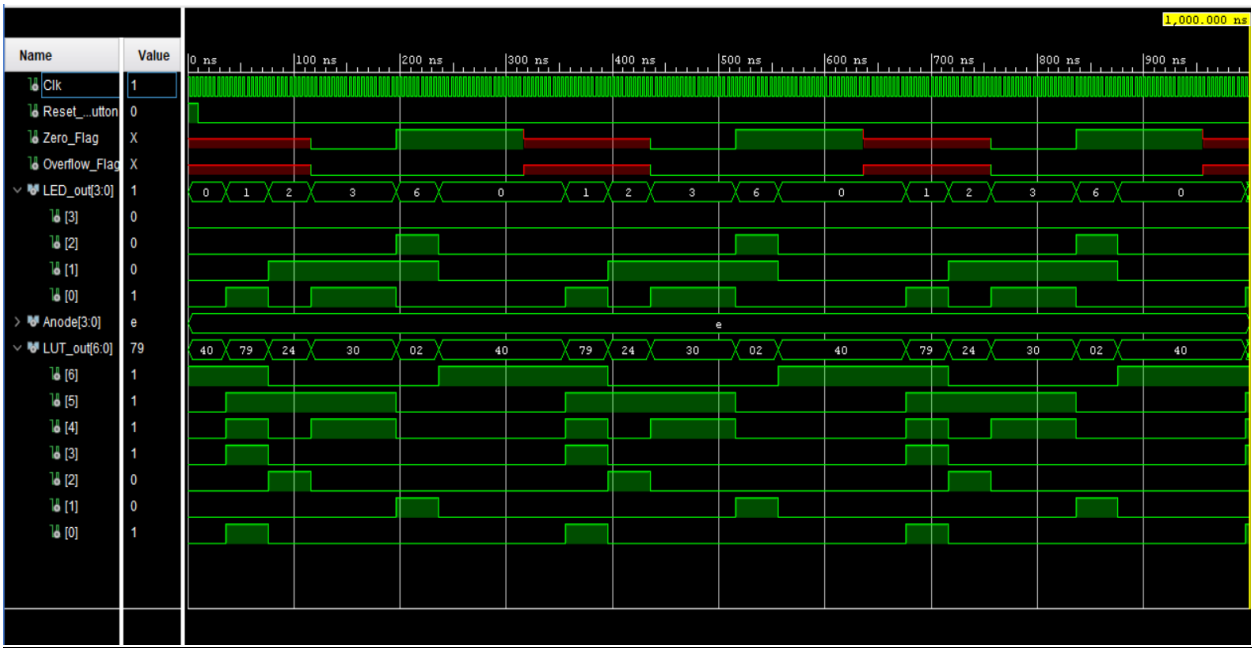
```
--
----------------------------------------------------------------------------------


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 TB_NanoProcessor is
--  Port ( );
end TB_NanoProcessor;

architecture Behavioral of TB_NanoProcessor is
component Nano_Processor
 Port (
    Reset_Push_Button:in STD_LOGIC;
    Clk:in STD_LOGIC;
    LED_out : out STD_LOGIC_VECTOR (3 downto 0);
    LUT_out:out STD_LOGIC_VECTOR(6 downto 0);
    Zero_Flag : out STD_LOGIC:='0';
    Overflow_Flag : out STD_LOGIC:='0';
    --pc:out STD_LOGIC_VECTOR(2 downto 0);
    --Reg_1_out: out STD_LOGIC_VECTOR (3 downto 0);
    --Reg_2_out: out STD_LOGIC_VECTOR (3 downto 0);
    Anode:out STD_LOGIC_VECTOR(3 downto 0));
```

```
end component;

component Slow_Clk
    port (
      Clk_in : in STD_LOGIC;
      Clk_out : out STD_LOGIC

    );
end component;

signal Clk,Reset_Push_Button:STD_LOGIC:='0';
signal Zero_Flag,Overflow_Flag:STD_LOGIC:='0';
signal LED_out,Anode:STD_LOGIC_VECTOR( 3 downto 0);
--signal pc:STD_LOGIC_VECTOR( 2 downto 0);
signal LUT_out:STD_LOGIC_VECTOR(6 downto 0);

begin
UUT: Nano_Processor port map(
Reset_Push_Button=>Reset_Push_Button,
Clk =>Clk,
Anode =>Anode,
LED_out => LED_out,
zero_Flag=>Zero_Flag,
Overflow_Flag =>Overflow_Flag,
--pc =>pc,
--Reg_1_out=>Reg_1_out,
--Reg_2_out=>Reg_2_out,
LUT_out=>LUT_out
);

process begin
  Clk <= NOT(Clk);
```

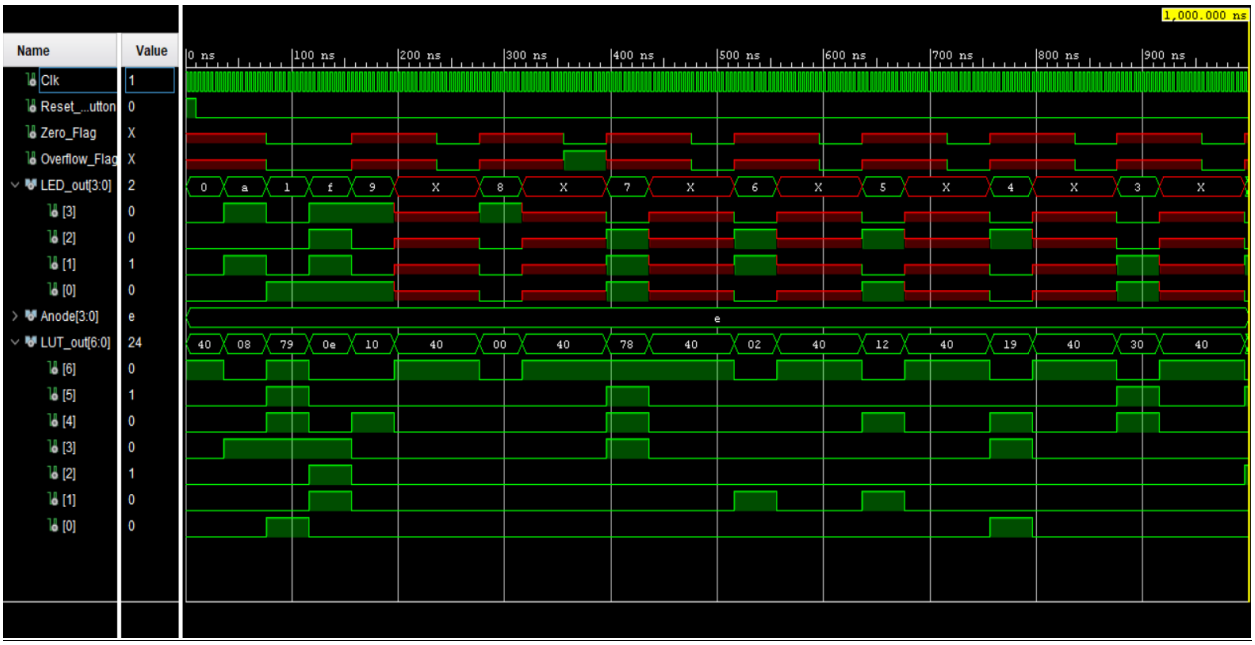
```
    wait for 2ns;
end process;

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

end process;

end Behavioral;
```

- Timing Diagrams

According to 1 st instruction set



According to 2 nd instruction set



CONCLUTION

In conclusion, this lab provided valuable hands-on experience in designing a simple microprocessor, or nanoprocessor, capable of executing basic instructions. Through teamwork, students developed essential skills in communication, coordination, and sharing responsibilities. By extending and modifying existing components, such as the arithmetic unit and program counter, students gained practical knowledge in digital circuit design. Overall, this project enhanced understanding of processor architecture and teamwork dynamics.

THE CONTRIBUTION OF EACH TEAM MEMBER TO PROJECT AND NUMBER OF HOURS SPENT

Team Member	Contribution to Project	Hours Spent
Vishmitha W.D.L	• As a leader Organized team meetings, delegated tasks, and ensured deadlines were met. •Register Bank with 3 to 8 Decoder •8 way 4 bit Mux • Troubleshooting: Helped resolve technical issue in the instruction decoder.	21 hours
Manawadu N.D.	• Design and Development of Instruction Decoder, Mux and Clock • Integrated all components into the Nano processor design. • Helped resolve technical issue (mentioned above)in the instruction decoder.	23 hours
Silva D.L.P.H.	• Design and Development of 4-bit Add/Sub unit , 2 way 3 bit Mux and create a 3-bit adder for the Program Counter. •Conducted simulations to verify the functionality of all components •Worked on transferring nano processor design to the development board (BASYS 3) and ensured its proper functioning.	20 hours
Chanuka MGG	•Design and Development of Program ROM and Program Counter. • Worked on transferring nano processor design to	20 hours

	the development board (BASYS 3) and ensured its proper functioning. • Documented the design process and simulation results to submit a report . • Assisted in resolving issue related to the Program ROM.	

END.