Lab 9-10 – Nanoprocessor Design Competition

CS1050 Computer Organization and Digital Design

Dept. of Computer Science and Engineering, University of Moratuwa

Group Members:

- Jayasooriya D.D.M. - 210250D
- Udayanga H.K.I. - 210661M

Lab task

We had to design a 4-bit processor which can execute instructions such as,

- MOVI R,d → Move immediate value d to register R
- ADD R_a , R_b → Add values in registers R_a and R_b and store the result in R_a
- NEG R → 2's complement of register R
- JZR R, d → Jump if value in register R is zero

First we had to design the following components.

- 4-bit Add/Subtract Unit
- 3-bit Adder
- 3-bit Program Counter
- 2-way 3-bit Multiplexer
- 2-way 4-bit Multiplexer
- 8-way 4-bit Multiplexer
- Register Bank
- Program ROM
- Instruction Decoder

After creating the components we connected them using buses.
Then we verified the functionality via simulation and on the development board.

Assembly Program
(To calculate the total of all integers between 1 and 3)

```
MOVI R1, 3      R1 ← 3
MOVI R2, 1           R2 ← 1
NEG R2          R2 ← (-R2)
ADD R7, R1      R7 ← R7 + R1
ADD R1, R2      R1 ← R1 + R2
JZR R1, 7       If R1=0 jump to line 7
JZR R0, 3       If R0=0 jump to line 3
```

Machine Code Representation

0. 100010000011
1. 100100000001
2. 010100000000
3. 001110010000
4. 000010100000
5. 110010000111
6. 110000000011
7. 110000000111

VHDL Codes

- **4-bit Add/Subtract unit**

```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date: 06/01/2023 10:02:29 AM
-- Design Name:
-- Module Name: ASU_4 - 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 ASU_4 is
    Port ( A : in STD_LOGIC_VECTOR (3 downto 0);
           B : in STD_LOGIC_VECTOR (3 downto 0);
           ASU_EN : in STD_LOGIC;
           AS_Sel : in STD_LOGIC;
           Sum : out STD_LOGIC_VECTOR (3 downto 0);
           Carry : out STD_LOGIC;
           Overflow : out STD_LOGIC;
           Zero : out STD_LOGIC);
end ASU_4;

architecture Behavioral of ASU_4 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 B0, B1, B2, B3, S0, S1, S2, S3, C0, C1, C2, C3 : STD_LOGIC;

begin

    B0 <= AS_Sel XOR B(0);
    B1 <= AS_Sel XOR B(1);
    B2 <= AS_Sel XOR B(2);
    B3 <= AS_Sel XOR B(3);

    FA_0 : FA
        port map(
        A => A(0),
        B => B0,
        C_in => AS_Sel, -- add 1 to LSB if subtracting to convert to two's
complement
        S => S0,
        C_out => C0);

    FA_1 : FA
        port map(
        A => A(1),
        B => B1,
        C_in => C0,
        S => S1,
        C_out => C1);

    FA_2 : FA
        port map(
        A => A(2),
        B => B2,
        C_in => C1,
        S => S2,
        C_out => C2);

    FA_3 : FA
        port map(
        A => A(3),
        B => B3,
        C_in => C2,
        S => S3,
        C_out => C3);
    Carry <= C3;
    Overflow <= C3 AND ASU_EN;
    Sum(0) <= S0 AND ASU_EN;
    Sum(1) <= S1 AND ASU_EN;
    Sum(2) <= S2 AND ASU_EN;
    Sum(3) <= S3 AND ASU_EN;

    process (S0, S1, S2, S3) begin
        if (S0 = '0' and S1 = '0' and S2 = '0' and S3 = '0') then
            Zero <= '1';
        else
            Zero <= '0';
        end if;

    end process;
end Behavioral;
```

- ## 3-bit adder

```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date: 06/01/2023 08:41:29 PM
-- Design Name:
-- Module Name: RCA_3 - 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 is
```

```
    Port ( I : in STD_LOGIC_VECTOR (2 downto 0);
           S_out : out STD_LOGIC_VECTOR (2 downto 0));
end RCA_3;

architecture Behavioral of RCA_3 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 C0, C1, C2, Carry : std_logic;
SIGNAL S : std_logic_vector(2 downto 0);

begin

    FA_0 : FA
        port map(
        A => '1',
        B => I(0),
        C_in => '0', -- grounded
        S => S(0),
        C_out => C0);

    FA_1 : FA
        port map(
        A => '0',
        B => I(1),
        C_in => C0,
        S => S(1),
        C_out => C1);

    FA_2 : FA
        port map(
        A => '0',
        B => I(2),
        C_in => C1,
        S => S(2),
        C_out => C2);

    S_out <= S;
    Carry <= C2;

end Behavioral;
```

- **3-bit Program Counter (PC)**

```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date: 06/01/2023 09:15:24 PM
```

```vhdl
-- Design Name:
-- Module Name: PC_3 - 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 PC_3 is
    Port ( D : in STD_LOGIC_VECTOR (2 downto 0);
           Res : in STD_LOGIC;
           Clk : in STD_LOGIC;
           Q : out STD_LOGIC_VECTOR (2 downto 0));
end PC_3;

architecture Behavioral of PC_3 is

component D_FF
    Port ( D : in STD_LOGIC;
           Res : in STD_LOGIC;
           Clk : in STD_LOGIC;
           Q : out STD_LOGIC;
           Qbar : out STD_LOGIC);
end component;

begin

    D_FF0 : D_FF
        port map (
            D => D(0),
            Res => Res,
            Clk => Clk,
            Q => Q(0));

    D_FF1 : D_FF
        port map (
            D => D(1),
            Res => Res,
            Clk => Clk,
            Q => Q(1));

    D_FF2 : D_FF
        port map (
            D => D(2),
            Res => Res,
            Clk => Clk,
            Q => Q(2));

end Behavioral;
```

- **tri-state buffer**

```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date: 06/02/2023 10:29:28 AM
-- Design Name:
-- Module Name: Tri_Buffer_3 - 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 Tri_Buffer_3 is
    Port ( I : in STD_LOGIC_VECTOR (2 downto 0);
           EN : in STD_LOGIC;
           I_out : out STD_LOGIC_VECTOR (2 downto 0));
end Tri_Buffer_3;

architecture Behavioral of Tri_Buffer_3 is

begin
```

```
    I_out <= I WHEN (EN = '1') ELSE "ZZZ";

end Behavioral;
```

- **2-way 3-bit multiplexer**

```
----------------------------------------------------------------------------------
-- Company: 
-- Engineer: 
-- 
-- Create Date: 06/02/2023 11:04:35 AM
-- Design Name: 
-- Module Name: MUX_2W_3B - 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_2W_3B is
    Port ( A : in STD_LOGIC_VECTOR (2 downto 0);
           B : in STD_LOGIC_VECTOR (2 downto 0);
           Sel : in STD_LOGIC;
           Y : out STD_LOGIC_VECTOR (2 downto 0));
end MUX_2W_3B;

architecture Behavioral of MUX_2W_3B is

component Tri_Buffer_3
```

```
    Port ( I : in STD_LOGIC_VECTOR (2 downto 0);
           EN : in STD_LOGIC;
           I_out : out STD_LOGIC_VECTOR (2 downto 0));
end component;

SIGNAL Y1, Y2 : STD_LOGIC_VECTOR (2 downto 0);
SIGNAL EN_A,EN_B : STD_LOGIC;

begin

    EN_A <= NOT(Sel);
    EN_B <= Sel;

    Buffer1 : Tri_Buffer_3
        port map (
        I => A,
        EN => EN_A,
        I_out => Y1);

    Buffer2 : Tri_Buffer_3
        port map (
        I => B,
        EN => EN_B,
        I_out => Y2);

    Y <= Y1 WHEN( Y2= "ZZZ") ELSE Y2;

end Behavioral;
```

- ## 2-way 4-bit multiplexer

```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date: 06/02/2023 11:53:50 AM
-- Design Name:
-- Module Name: MUX_2W_4B - 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_2W_4B is
    Port ( A : in STD_LOGIC_VECTOR (3 downto 0);
           B : in STD_LOGIC_VECTOR (3 downto 0);
           Sel : in STD_LOGIC;
           Y : out STD_LOGIC_VECTOR (3 downto 0));
end MUX_2W_4B;

architecture Behavioral of MUX_2W_4B is

component Tri_Buffer_4
    Port ( I : in STD_LOGIC_VECTOR (3 downto 0);
           EN : in STD_LOGIC;
           I_out : out STD_LOGIC_VECTOR (3 downto 0));
end component;

SIGNAL Y1, Y2 : STD_LOGIC_VECTOR (3 downto 0);
SIGNAL EN_A,EN_B : STD_LOGIC;

begin

    EN_A <= NOT(Sel);
    EN_B <= Sel;

    Buffer1 : Tri_Buffer_4
        port map (
        I => A,
        EN => EN_A,
        I_out => Y1);

    Buffer2 : Tri_Buffer_4
        port map (
        I => B,
        EN => EN_B,
        I_out => Y2);

    Y <= Y1 WHEN( Y2= "ZZZZ") ELSE Y2;


end Behavioral;
```

- **8-way 4-bit multiplexer**

```
----------------------------------------------------------------------------------
-- Company: 
-- Engineer: 
-- 
-- Create Date: 06/02/2023 02:00:13 PM
-- Design Name: 
-- Module Name: MUX_8W_4B - 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_8W_4B is
    Port ( Reg0 : in STD_LOGIC_VECTOR (3 downto 0);
           Reg1 : in STD_LOGIC_VECTOR (3 downto 0);
           Reg2 : in STD_LOGIC_VECTOR (3 downto 0);
           Reg3 : in STD_LOGIC_VECTOR (3 downto 0);
           Reg4 : in STD_LOGIC_VECTOR (3 downto 0);
           Reg5 : in STD_LOGIC_VECTOR (3 downto 0);
           Reg6 : in STD_LOGIC_VECTOR (3 downto 0);
           Reg7 : in STD_LOGIC_VECTOR (3 downto 0);
           RegSel : in STD_LOGIC_VECTOR (2 downto 0);
           RegOut : out STD_LOGIC_VECTOR (3 downto 0));
end MUX_8W_4B;

architecture Behavioral of MUX_8W_4B 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;

component Tri_Buffer_4
```

```
    Port ( I : in STD_LOGIC_VECTOR (3 downto 0);
           EN : in STD_LOGIC;
           I_out : out STD_LOGIC_VECTOR (3 downto 0));
end component;

SIGNAL Y0, Y1, Y2, Y3, Y4, Y5, Y6, Y7 : STD_LOGIC_VECTOR (3 downto 0);
SIGNAL Reg_EN : STD_LOGIC_VECTOR(7 downto 0);

begin

    Decoder: Decoder_3_to_8
        port map (
        I => RegSel,
        EN => '1',
        Y => Reg_EN);

    Buffer0 : Tri_Buffer_4
            port map (
            I => Reg0,
            EN => Reg_EN(0),
            I_out => Y0);

    Buffer1 : Tri_Buffer_4
            port map (
            I => Reg1,
            EN => Reg_EN(1),
            I_out => Y1);

    Buffer2 : Tri_Buffer_4
            port map (
            I => Reg2,
            EN => Reg_EN(2),
            I_out => Y2);

    Buffer3 : Tri_Buffer_4
            port map (
            I => Reg3,
            EN => Reg_EN(3),
            I_out => Y3);

    Buffer4 : Tri_Buffer_4
            port map (
            I => Reg4,
            EN => Reg_EN(4),
            I_out => Y4);

    Buffer5 : Tri_Buffer_4
            port map (
            I => Reg5,
            EN => Reg_EN(5),
            I_out => Y5);

    Buffer6 : Tri_Buffer_4
            port map (
            I => Reg6,
            EN => Reg_EN(6),
            I_out => Y6);

    Buffer7 : Tri_Buffer_4
```

```
            port map (
            I => Reg7,
            EN => Reg_EN(7),
            I_out => Y7);

    RegOut <= Y0 WHEN( Y0 /= "ZZZZ") ELSE
              Y1 WHEN( Y1 /= "ZZZZ") ELSE
              Y2 WHEN( Y2 /= "ZZZZ") ELSE
              Y3 WHEN( Y3 /= "ZZZZ") ELSE
              Y4 WHEN( Y4 /= "ZZZZ") ELSE
              Y5 WHEN( Y5 /= "ZZZZ") ELSE
              Y6 WHEN( Y6 /= "ZZZZ") ELSE
              Y7 WHEN( Y7 /= "ZZZZ");

end Behavioral;
```

- **Register Bank**

```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date: 06/02/2023 03:44:08 PM
-- Design Name:
-- Module Name: RegBank - 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 RegBank is
    Port ( D : in STD_LOGIC_VECTOR (3 downto 0);
           Reg_EN : in STD_LOGIC_VECTOR (2 downto 0);
           CLK : in STD_LOGIC;
           Res : in STD_LOGIC;
           Q0 : out STD_LOGIC_VECTOR (3 downto 0);
           Q1 : out STD_LOGIC_VECTOR (3 downto 0):= "0000";
           Q2 : out STD_LOGIC_VECTOR (3 downto 0);
           Q3 : out STD_LOGIC_VECTOR (3 downto 0);
           Q4 : out STD_LOGIC_VECTOR (3 downto 0);
           Q5 : out STD_LOGIC_VECTOR (3 downto 0);
           Q6 : out STD_LOGIC_VECTOR (3 downto 0);
           Q7 : out STD_LOGIC_VECTOR (3 downto 0));
end RegBank;

architecture Behavioral of RegBank 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;

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

SIGNAL R_EN : STD_LOGIC_VECTOR (7 downto 0);

begin

    Decoder : Decoder_3_to_8
    port map (
        I => Reg_EN,
        EN => '1',
        Y => R_EN );

    R0 : Reg
    port map (
        D => "0000",
        EN => R_EN(0),
        Clk => CLK,
        Q => Q0,
        Res => Res );

    R1 : Reg
    port map (
        D => D,
        EN => R_EN(1),
        Clk => CLK,
        Q => Q1,
        Res => Res );

    R2 : Reg
    port map (
        D => D,
        EN => R_EN(2),
        Clk => CLK,
        Q => Q2,
        Res => Res );
    R3 : Reg
    port map (
        D => D,
        EN => R_EN(3),
        Clk => CLK,
        Q => Q3,
        Res => Res );

    R4 : Reg
    port map (
        D => D,
        EN => R_EN(4),
        Clk => CLK,
        Q => Q4,
        Res => Res );

    R5 : Reg
    port map (
        D => D,
        EN => R_EN(5),
        Clk => CLK,
        Q => Q5,
        Res => Res );

    R6 : Reg
    port map (
        D => D,
        EN => R_EN(6),
        Clk => CLK,
        Q => Q6,
        Res => Res );

    R7 : Reg
    port map (
        D => D,
        EN => R_EN(7),
        Clk => CLK,
        Q => Q7,
        Res => Res );

end Behavioral;
```

- ## Program ROM

```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date: 06/02/2023 05:04:37 PM
-- Design Name:
-- Module Name: ProgROM - 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 ProgROM is
    Port ( M_Sel : in STD_LOGIC_VECTOR (2 downto 0);
           Instruction : out STD_LOGIC_VECTOR (11 downto 0));
end ProgROM;

architecture Behavioral of ProgROM is

    type rom_type is array (0 to 7) of std_logic_vector (11 downto 0);
        SIGNAL program_ROM : rom_type := (
            "100010000011", -- MOVI R1,3
            "100100000001", -- MOVI R2,1
            "010100000000", -- NEG R2
            "001110010000", -- ADD R7,R1
            "000010100000", -- ADD R1,R2
            "110010000111", --JZR R1,7
            "110000000011", --JZR R0,3
            "110000000111"  -- NULL
            );

begin

    Instruction <= program_ROM(to_integer(unsigned(M_Sel)));

end Behavioral;
```

- **Instruction Decoder**

```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date: 06/05/2023 06:59:12 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 : in STD_LOGIC_VECTOR (11 downto 0);
           Reg_Check_J : in STD_LOGIC_VECTOR (3 downto 0);
           Reg_EN : out STD_LOGIC_VECTOR (2 downto 0);
           Load_Sel : out STD_LOGIC;
           Im_Val : out STD_LOGIC_VECTOR (3 downto 0);
           Reg_Sel_1 : out STD_LOGIC_VECTOR (2 downto 0);
           Reg_Sel_2 : out STD_LOGIC_VECTOR (2 downto 0);
           Add_Sub_Sel : out STD_LOGIC;
           Jump_Flag : out STD_LOGIC;
           ASU_EN : out STD_LOGIC := '0';
           Jump_Address : out STD_LOGIC_VECTOR (2 downto 0));
end Instruction_Decoder;

architecture Behavioral of Instruction_Decoder is

begin

    process(Instruction, reg_check_j) begin
        case Instruction(11 downto 10) is

            when "00" =>
                -- sending addresses of the register to be enabled
                Reg_Sel_1 <= Instruction(9 downto 7);
                Reg_Sel_2 <= Instruction(6 downto 4);
                Add_Sub_Sel <= '0';
                Reg_EN <= Instruction(9 downto 7); -- Enabling the register to
store the answer
                Load_Sel <= '0'; -- selecting the output of the 4 bit Add sub unit
(ASU_4)
                Jump_Flag <= '0';
                Im_Val <= "0000";
                Jump_Address <= "000";
                ASU_EN <= '1';

            when "01" =>
                Reg_Sel_2 <= Instruction(9 downto 7);
                Reg_Sel_1 <= "000";
                Reg_EN <= Instruction(9 downto 7);
                Add_Sub_Sel <= '1'; -- Enable subtractor
                ASU_EN <= '1';
                Load_Sel <= '0';
                Im_Val <= "0000";
                Jump_Flag <= '0';
                Jump_Address <= "000";

            when "10" =>
                Reg_EN <= Instruction(9 downto 7);
                Im_Val <= Instruction(3 downto 0);
                Load_Sel <= '1'; -- selecting the immediate value
                Jump_Flag <= '0';
                reg_sel_2 <= "000";
                reg_sel_1 <= "000";
                Add_Sub_Sel <= '0';
                Jump_Address <= "000";

            when "11" =>
                Reg_Sel_1 <= Instruction(9 downto 7);
                reg_sel_2 <= "000";
                load_sel <= '0';
                Add_Sub_Sel <= '0';
                Im_Val <= "0000";
                reg_en <= "000";
                if (Reg_Check_J = "0000") then -- jump if value in register at
Reg_Sel_1 is 0
                    Jump_Flag <= '1';
                    Jump_Address <= Instruction(2 downto 0);
                end if;

            when others =>
        end case;
    end process;

end Behavioral;
```

- **Processor**

```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date: 06/07/2023 06:17:37 PM
-- Design Name:
-- Module Name: 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 NanoProcessor is
    Port ( CLK : in STD_LOGIC;
           Res : in STD_LOGIC;
           Zero: out STD_LOGIC;
           Overflow : out STD_LOGIC;
           Reg7_LED: out STD_LOGIC_VECTOR (3 downto 0);
           Reg7_7Seg : out STD_LOGIC_VECTOR (6 downto 0);
           Anode : out STD_LOGIC_VECTOR (3 downto 0):= "1110");
end NanoProcessor;

architecture Behavioral of NanoProcessor is

    component Instruction_Decoder
        Port ( Instruction : in STD_LOGIC_VECTOR (11 downto 0);
               Reg_Check_J : in STD_LOGIC_VECTOR (3 downto 0);
               Reg_EN : out STD_LOGIC_VECTOR (2 downto 0);
               Load_Sel : out STD_LOGIC;
               Im_Val : out STD_LOGIC_VECTOR (3 downto 0);
               Reg_Sel_1 : out STD_LOGIC_VECTOR (2 downto 0);
               Reg_Sel_2 : out STD_LOGIC_VECTOR (2 downto 0);
               Add_Sub_Sel : out STD_LOGIC;
               Jump_Flag : out STD_LOGIC;
               ASU_EN : out STD_LOGIC := '0';
               Jump_Address : out STD_LOGIC_VECTOR (2 downto 0));
    end component;

    component RegBank
        Port ( D : in STD_LOGIC_VECTOR (3 downto 0);
               CLK : in STD_LOGIC;
               Reg_EN : in STD_LOGIC_VECTOR (2 downto 0);
               Res : in STD_LOGIC;
               Q0 : out STD_LOGIC_VECTOR (3 downto 0);
               Q1 : out STD_LOGIC_VECTOR (3 downto 0);
               Q2 : out STD_LOGIC_VECTOR (3 downto 0);
               Q3 : out STD_LOGIC_VECTOR (3 downto 0);
               Q4 : out STD_LOGIC_VECTOR (3 downto 0);
               Q5 : out STD_LOGIC_VECTOR (3 downto 0);
               Q6 : out STD_LOGIC_VECTOR (3 downto 0);
               Q7 : out STD_LOGIC_VECTOR (3 downto 0));
    end component;

    component ProgROM
        Port ( M_Sel : in STD_LOGIC_VECTOR (2 downto 0);
               Instruction : out STD_LOGIC_VECTOR (11 downto 0));
    end component;

    component PC_3
        Port ( D : in STD_LOGIC_VECTOR (2 downto 0) := "000";
               Res : in STD_LOGIC;
               Clk : in STD_LOGIC;
               Q : out STD_LOGIC_VECTOR (2 downto 0) := "000");
    end component;

    component ASU_4
        Port ( A : in STD_LOGIC_VECTOR (3 downto 0);
               B : in STD_LOGIC_VECTOR (3 downto 0);
               AS_Sel : in STD_LOGIC:='0';
               ASU_EN : in STD_LOGIC;
               Sum : out STD_LOGIC_VECTOR (3 downto 0);
               Carry : out STD_LOGIC;
               Overflow : out STD_LOGIC;
               Zero : out STD_LOGIC);
    end component;

    component RCA_3
        Port ( I : in STD_LOGIC_VECTOR (2 downto 0) := "000";
               S_out : out STD_LOGIC_VECTOR (2 downto 0) := "000");
    end component;

    component MUX_8W_4B
        Port ( Reg0 : in STD_LOGIC_VECTOR (3 downto 0);
               Reg1 : in STD_LOGIC_VECTOR (3 downto 0);
               Reg2 : in STD_LOGIC_VECTOR (3 downto 0);
               Reg3 : in STD_LOGIC_VECTOR (3 downto 0);
               Reg4 : in STD_LOGIC_VECTOR (3 downto 0);
               Reg5 : in STD_LOGIC_VECTOR (3 downto 0);
               Reg6 : in STD_LOGIC_VECTOR (3 downto 0);
               Reg7 : in STD_LOGIC_VECTOR (3 downto 0);
               RegSel : in STD_LOGIC_VECTOR (2 downto 0);
               RegOut : out STD_LOGIC_VECTOR (3 downto 0));
    end component;

    component MUX_2W_3B
        Port ( A : in STD_LOGIC_VECTOR (2 downto 0) := "000";
               B : in STD_LOGIC_VECTOR (2 downto 0) := "000";
               Sel : in STD_LOGIC;
               Y : out STD_LOGIC_VECTOR (2 downto 0) := "000");
    end component;

    component MUX_2W_4B
        Port ( A : in STD_LOGIC_VECTOR (3 downto 0);
               B : in STD_LOGIC_VECTOR (3 downto 0);
               Sel : in STD_LOGIC;
               Y : out STD_LOGIC_VECTOR (3 downto 0));
    end component;
```

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

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

SIGNAL CLK_Slow, Load_Sel, AS_Sel, Jump_Flag, Carry, ASU_EN : STD_LOGIC;
SIGNAL Instruction : STD_LOGIC_VECTOR (11 downto 0);
SIGNAL A_Out, Im_Val, ASU_4_Sum, D, B_Out : STD_LOGIC_VECTOR (3 downto 0);
SIGNAL Reg0, Reg1, Reg2, Reg3, Reg4, Reg5, Reg6, Reg7 : STD_LOGIC_VECTOR (3 downto
0);
SIGNAL Reg_EN, Reg_Sel_A, Reg_Sel_B, Jump_Address, M_Sel,Incremented_Address :
STD_LOGIC_VECTOR (2 downto 0);
SIGNAL Next_Address : STD_LOGIC_VECTOR (2 downto 0) := "000";

begin

Slow_Clk_0 : Slow_Clk port map (
    Clk_in => CLK,
    Clk_out => CLK_Slow);

Instruction_Decoder_0 : Instruction_Decoder port map (
    Instruction => Instruction,
    Reg_Check_J => A_Out,
    Reg_EN => Reg_EN,
    Load_Sel => Load_Sel,
    Im_Val => Im_Val,
    Reg_Sel_1 => Reg_Sel_A,
    Reg_Sel_2 => Reg_Sel_B,
    Add_Sub_Sel => AS_Sel,
    Jump_Flag => Jump_Flag,
    ASU_EN => ASU_EN,
    Jump_Address => Jump_Address);


MUX_2W_4B_0 : MUX_2W_4B port map ( -- MUX to select Data to be sent to registers
    A => ASU_4_Sum,
    B => Im_Val,
    Sel => Load_Sel,
    Y => D);

Add_Sub_Unit : ASU_4 port map (
    A => A_Out,
    B => B_Out,
    AS_Sel => AS_Sel,
    ASU_EN => ASU_EN,
    Sum => ASU_4_Sum,
    Carry => Carry,
    Overflow => Overflow,
    Zero => Zero);

MUX_8W_4B_A : MUX_8W_4B port map (
    Reg0 => Reg0,
    Reg1 => Reg1,
    Reg2 => Reg2,
    Reg3 => Reg3,
    Reg4 => Reg4,
    Reg5 => Reg5,
    Reg6 => Reg6,
    Reg7 => Reg7,
    RegSel => Reg_Sel_A,
    RegOut => A_Out);


MUX_8W_4B_B : MUX_8W_4B port map (
    Reg0 => Reg0,
    Reg1 => Reg1,
    Reg2 => Reg2,
    Reg3 => Reg3,
    Reg4 => Reg4,
    Reg5 => Reg5,
    Reg6 => Reg6,
    Reg7 => Reg7,
    RegSel => Reg_Sel_B,
    RegOut => B_Out);

Adder_3bit : RCA_3 port map (
    I => M_Sel,
    S_out => Incremented_Address);



MUX_2W_3B_0 : MUX_2W_3B port map ( -- to select the address of the next instruction
    A => Incremented_Address,
    B => Jump_Address,
    Sel => Jump_Flag,
    Y => Next_Address);

Program_Counter :  PC_3 port map (
    D => Next_Address,
    Res => Res,
    Clk => CLK_Slow,
    Q => M_Sel);

Program_ROM : ProgROM port map (
    M_Sel => M_Sel,
    Instruction => Instruction);

Register_bank : RegBank port map (
    D => D,
    CLK => CLK_Slow,
    Reg_EN => Reg_EN,
    Res => Res,
    Q0 => Reg0,
    Q1 => Reg1,
    Q2 => Reg2,
    Q3 => Reg3,
    Q4 => Reg4,
    Q5 => Reg5,
    Q6 => Reg6,
    Q7 => Reg7);

LUT_7Seg : LUT_16_7 port map (
    address => Reg7,
    data => Reg7_7Seg);

Reg7_LED <= Reg7;

end Behavioral;
```

Test Bench Codes

- ### 4-bit Add/Subtract unit

```
----------------------------------------------------------------------------------
-- Company: 
-- Engineer: 
-- 
-- Create Date: 06/01/2023 10:30:45 AM
-- Design Name: 
-- Module Name: ASU_4_TB - 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 ASU_4_TB is
--  Port ( );
end ASU_4_TB;

architecture Behavioral of ASU_4_TB is

component ASU_4
    Port ( A : in STD_LOGIC_VECTOR (3 downto 0);
           B : in STD_LOGIC_VECTOR (3 downto 0);
           AS_Sel : in STD_LOGIC;
           ASU_EN : in STD_LOGIC;
           Sum : out STD_LOGIC_VECTOR (3 downto 0);
           Carry : out STD_LOGIC;
           Overflow : out STD_LOGIC;
           Zero : out STD_LOGIC);
end component;

SIGNAL A, B, Sum : STD_LOGIC_VECTOR (3 downto 0);
SIGNAL AS_Sel, Carry, Zero, Overflow, ASU_EN : STD_LOGIC;

begin

   UUT : ASU_4 PORT MAP (
            A => A,
            B => B,
            AS_Sel => AS_Sel,
            ASU_EN => ASU_EN,
            Sum => Sum,
            Carry => Carry,
            Overflow => Overflow,
            Zero => Zero);

    process begin

        -- 210250 -> 11 0011 0101 0100 1010
        -- 210661 -> 11 0011 0110 1110 0101

        ASU_EN <= '0';

        A <= "1010";
        B <= "0101";
        AS_Sel <= '0';
        wait for 100ns;

        AS_Sel <= '1';
        wait for 100ns;

        ASU_EN <= '1';

        A <= "1010";
        B <= "0101";
        AS_Sel <= '0';
        wait for 100ns;

        AS_Sel <= '1';
        wait for 100ns;

        A <= "0100";
        B <= "1110";
        AS_Sel <= '0';
        wait for 100ns;

        AS_Sel <= '1';
        wait for 100ns;

        A <= "0101";
        B <= "0110";
        AS_Sel <= '0';
        wait for 100ns;
        AS_Sel <= '1';
        wait for 100ns;

        A <= "0011";
        B <= "0011";
        AS_Sel <= '0';
        wait for 100ns;

        A <= "0000";
        B <= "0001";
        AS_Sel <= '1';
        wait for 100ns;

        AS_Sel <= '1';
        wait;
    end process;

end Behavioral;
```

- ### 3-bit adder

```
----------------------------------------------------------------------------------
-- Company: 
-- Engineer: 
-- 
-- Create Date: 06/01/2023 08:52:10 PM
-- Design Name: 
-- Module Name: RCA_3_TB - 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_TB is
--  Port ( );
end RCA_3_TB;

architecture Behavioral of RCA_3_TB is

component RCA_3
    Port ( I : in STD_LOGIC_VECTOR (2 downto 0);
           S_out : out STD_LOGIC_VECTOR (2 downto 0));
end component;

SIGNAL I, S_out : STD_LOGIC_VECTOR (2 downto 0);

begin

    UUT: RCA_3 PORT MAP (
            I => I,
            S_out => S_out);


    process begin

        -- 210250 -> 11 0011 0101 0100 1010
        -- 210661 -> 11 0011 0110 1110 0101

        I <= "010";
        wait for 100ns;

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

        I <= "001";
        wait for 100ns;

        I <= "100";
        wait;

    end process;

end Behavioral;
```

- ## 3-bit Program Counter (PC)

```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date: 06/01/2023 09:25:00 PM
-- Design Name:
-- Module Name: PC_3_TB - 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 PC_3_TB is
--  Port ( );
end PC_3_TB;

architecture Behavioral of PC_3_TB is

component PC_3
    Port ( D : in STD_LOGIC_VECTOR (2 downto 0);
           Res : in STD_LOGIC;
           Clk : in STD_LOGIC;
           Q : out STD_LOGIC_VECTOR (2 downto 0));
end component;

SIGNAL D, Q : STD_LOGIC_VECTOR (2 downto 0);
SIGNAL Res, Clk : STD_LOGIC := '0';

begin

    UUT : PC_3 PORT MAP (
        D => D,
        Res => Res,
        Clk => Clk,
        Q => Q);

    process begin
        Clk <= NOT(Clk);
        wait for 10ns;
    end process;

    process begin
        -- 210250 -> 11 0011 0101 0100 1010
        -- 210661 -> 11 0011 0110 1110 0101

        D <= "010";
        wait for 100ns;

        Res <= '1';
        wait for 100ns;

        D <= "101";
        wait for 100ns;
        Res <= '1';
        wait for 100ns;

        Res <= '0';
        wait;
    end process;


end Behavioral;
```

- **tri-state buffer**

```
----------------------------------------------------------------------------------
-- Company: 
-- Engineer: 
-- 
-- Create Date: 06/02/2023 10:41:54 AM
-- Design Name: 
-- Module Name: Tri_Buffer_3_TB - 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 Tri_Buffer_3_TB is
--  Port ( );
end Tri_Buffer_3_TB;

architecture Behavioral of Tri_Buffer_3_TB is

component Tri_Buffer_3
    Port ( I : in STD_LOGIC_VECTOR (2 downto 0);
           EN : in STD_LOGIC;
           I_out : out STD_LOGIC_VECTOR (2 downto 0));
end component;

SIGNAL I, I_out : STD_LOGIC_VECTOR (2 downto 0);
SIGNAL EN : STD_LOGIC;

begin

    UUT : Tri_Buffer_3 PORT MAP (
            I => I,
            EN => EN,
            I_out => I_out);


    process begin
        -- 210250 -> 11 0011 0101 0100 1010
        -- 210661 -> 11 0011 0110 1110 0101
        EN <= '1';
        I <= "010";
        wait for 100ns;

        EN <= '0';
        wait for 100ns;

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

        EN <= '1';
        wait;

    end process;

end Behavioral;
```

- **2-way 3-bit multiplexer**

```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date: 06/02/2023 11:25:00 AM
-- Design Name:
-- Module Name: MUX_2W_3B_TB - 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_2W_3B_TB is
--  Port ( );
end MUX_2W_3B_TB;

architecture Behavioral of MUX_2W_3B_TB is

component MUX_2W_3B
    Port ( A : in STD_LOGIC_VECTOR (2 downto 0);
           B : in STD_LOGIC_VECTOR (2 downto 0);
           Sel : in STD_LOGIC;
           Y : out STD_LOGIC_VECTOR (2 downto 0));
end component;

SIGNAL A, B, Y : STD_LOGIC_VECTOR (2 downto 0);
SIGNAL Sel : STD_LOGIC;

begin

    UUT : MUX_2W_3B PORT MAP (
            A => A,
            B => B,
            Sel => Sel,
            Y => Y);

    process begin
        -- 210250 -> 11 0011 0101 0100 1010
        -- 210661 -> 11 0011 0110 1110 0101

        A <= "010";
        B <= "101";
        Sel <= '0';
        wait for 400ns;

        Sel <= '1';
        wait;

    end process;

end Behavioral;
```

- **2-way 4-bit multiplexer**

```
----------------------------------------------------------------------------------
-- Company: 
-- Engineer: 
-- 
-- Create Date: 06/02/2023 11:58:35 AM
-- Design Name: 
-- Module Name: MUX_2W_4B_TB - 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_2W_4B_TB is
--  Port ( );
end MUX_2W_4B_TB;

architecture Behavioral of MUX_2W_4B_TB is

component MUX_2W_4B
    Port ( A : in STD_LOGIC_VECTOR (3 downto 0);
           B : in STD_LOGIC_VECTOR (3 downto 0);
           Sel : in STD_LOGIC;
           Y : out STD_LOGIC_VECTOR (3 downto 0));
end component;

SIGNAL A, B, Y : STD_LOGIC_VECTOR (3 downto 0);
SIGNAL Sel : STD_LOGIC;


begin

    UUT : MUX_2W_4B PORT MAP (
            A => A,
            B => B,
            Sel => Sel,
            Y => Y);

    process begin
        -- 210250 -> 11 0011 0101 0100 1010
        -- 210661 -> 11 0011 0110 1110 0101

        A <= "1010";
        B <= "0101";
        Sel <= '0';
        wait for 400ns;

        Sel <= '1';
        wait;

    end process;

end Behavioral;
```

- **8-way 4-bit multiplexer**

```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date: 06/02/2023 02:35:32 PM
-- Design Name:
-- Module Name: MUX_8W_4B_TB - 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_8W_4B_TB is
--  Port ( );
```

```
end MUX_8W_4B_TB;

architecture Behavioral of MUX_8W_4B_TB is

component MUX_8W_4B
    Port ( Reg0 : in STD_LOGIC_VECTOR (3 downto 0);
           Reg1 : in STD_LOGIC_VECTOR (3 downto 0);
           Reg2 : in STD_LOGIC_VECTOR (3 downto 0);
           Reg3 : in STD_LOGIC_VECTOR (3 downto 0);
           Reg4 : in STD_LOGIC_VECTOR (3 downto 0);
           Reg5 : in STD_LOGIC_VECTOR (3 downto 0);
           Reg6 : in STD_LOGIC_VECTOR (3 downto 0);
           Reg7 : in STD_LOGIC_VECTOR (3 downto 0);
           RegSel : in STD_LOGIC_VECTOR (2 downto 0);
           RegOut : out STD_LOGIC_VECTOR (3 downto 0));
end component;

SIGNAL Reg0, Reg1, Reg2, Reg3, Reg4, Reg5, Reg6, Reg7, RegOut :  STD_LOGIC_VECTOR
(3 downto 0);
SIGNAL RegSel : STD_LOGIC_VECTOR (2 downto 0);

begin

    UUT : MUX_8W_4B PORT MAP (
           Reg0 => Reg0,
           Reg1 => Reg1,
           Reg2 => Reg2,
           Reg3 => Reg3,
           Reg4 => Reg4,
           Reg5 => Reg5,
           Reg6 => Reg6,
           Reg7 => Reg7,
           RegSel => RegSel,
           RegOut => RegOut);


    process begin
        -- 210250 -> 11 0011 0101 0100 1010
        -- 210661 -> 11 0011 0110 1110 0101

        Reg0 <= "1010";
        Reg1 <= "0101";
        Reg2 <= "0100";
        Reg3 <= "1110";
        Reg4 <= "0111";
        Reg5 <= "0110";
        Reg6 <= "0011";
        Reg7 <= "1011";
        RegSel <= "000";
        wait for 100ns;

        RegSel <= "001";
        wait for 100ns;

        RegSel <= "010";
        wait for 100ns;

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

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

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

        RegSel <= "111";
        wait;

    end process;


end Behavioral;
```

- **Register Bank**

```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date: 06/05/2023 04:27:43 PM
-- Design Name:
-- Module Name: RegBank_TB - 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 RegBank_TB is
--  Port ( );
end RegBank_TB;
```

```
architecture Behavioral of RegBank_TB is

component RegBank is
    Port ( D : in STD_LOGIC_VECTOR (3 downto 0);
           CLK : in STD_LOGIC;
           Reg_EN : in STD_LOGIC_VECTOR (2 downto 0);
           Res : in STD_LOGIC;
           Q0 : out STD_LOGIC_VECTOR (3 downto 0);
           Q1 : out STD_LOGIC_VECTOR (3 downto 0);
           Q2 : out STD_LOGIC_VECTOR (3 downto 0);
           Q3 : out STD_LOGIC_VECTOR (3 downto 0);
           Q4 : out STD_LOGIC_VECTOR (3 downto 0);
           Q5 : out STD_LOGIC_VECTOR (3 downto 0);
           Q6 : out STD_LOGIC_VECTOR (3 downto 0);
           Q7 : out STD_LOGIC_VECTOR (3 downto 0));
end component;

SIGNAL Res, CLK : STD_LOGIC := '0';
SIGNAL D, Q0, Q1, Q2, Q3, Q4, Q5, Q6, Q7 : STD_LOGIC_VECTOR (3 downto 0);
SIGNAL Reg_EN : STD_LOGIC_VECTOR (2 downto 0);

begin

    UUT : RegBank PORT MAP (
        D => D,
        CLK => CLK,
        Reg_EN => Reg_EN,
        Res => Res,
        Q0 => Q0,
        Q1 => Q1,
        Q2 => Q2,
        Q3 => Q3,
        Q4 => Q4,
        Q5 => Q5,
        Q6 => Q6,
        Q7 => Q7 );

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

    process begin

        -- 210250 -> 11 0011 0101 0100 1010
        -- 210661 -> 11 0011 0110 1110 0101

        D <= "1010";
        Reg_EN <= "101";
        wait for 100ns;

        D <= "0101";
        Reg_EN <= "100";
        wait for 100ns;

        D <= "1110";
        Reg_EN <= "110";
        wait for 100ns;
        Res <= '1';
        wait;

    end process;


end Behavioral;
```

- ## Program ROM

```
----------------------------------------------------------------------------------
-- Company: 
-- Engineer: 
-- 
-- Create Date: 06/05/2023 06:31:46 PM
-- Design Name: 
-- Module Name: ProgROM_TB - 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 ProgROM_TB is
--  Port ( );
end ProgROM_TB;

architecture Behavioral of ProgROM_TB is

component ProgROM
    Port ( M_Sel : in STD_LOGIC_VECTOR (2 downto 0);
           Instruction : out STD_LOGIC_VECTOR (11 downto 0));
end component;

SIGNAL M_Sel : STD_LOGIC_VECTOR (2 downto 0);
SIGNAL Instruction : STD_LOGIC_VECTOR (11 downto 0);

begin

    UUT : ProgROM PORT MAP (
        M_Sel => M_Sel,
        Instruction => Instruction);

    Process begin

    -- 210250 -> 11 0011 0101 0100 1010
    -- 210661 -> 11 0011 0110 1110 0101
        M_Sel <= "010";
        wait for 300ns;

        M_Sel <= "101";
        wait for 300ns;

        M_Sel <= "001";
        wait;

    end process;

end Behavioral;
```

- ## Instruction Decoder

```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date: 06/06/2023 03:39:39 PM
-- Design Name:
-- Module Name: Instruction_Decoder_TB - 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;
```

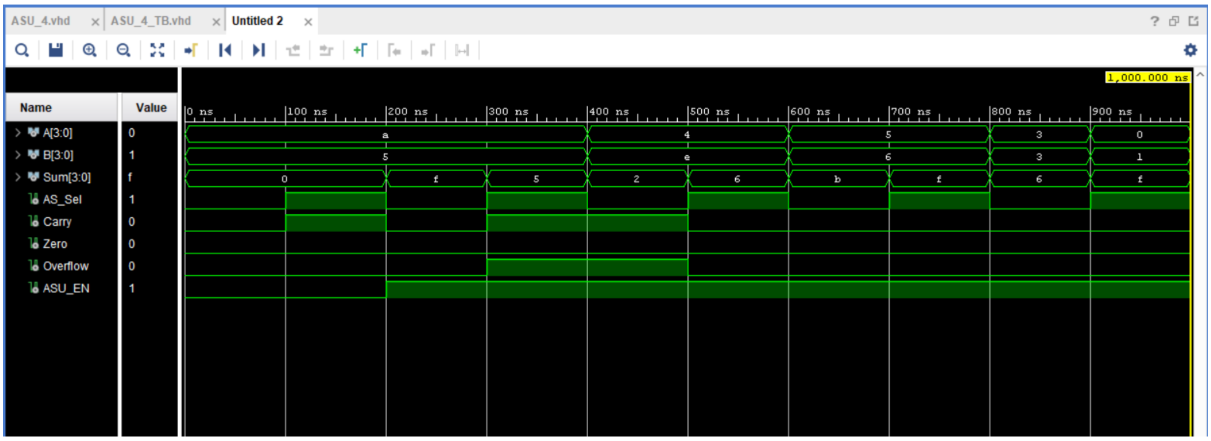
```
-- Uncomment the following library declaration if instantiating
-- any Xilinx leaf cells in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

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

architecture Behavioral of Instruction_Decoder_TB is

component Instruction_Decoder
    Port ( Instruction : in STD_LOGIC_VECTOR (11 downto 0);
           Reg_Check_J : in STD_LOGIC_VECTOR (3 downto 0);
           Reg_EN : out STD_LOGIC_VECTOR (2 downto 0);
           Load_Sel : out STD_LOGIC;
           Im_Val : out STD_LOGIC_VECTOR (3 downto 0);
           Reg_Sel_1 : out STD_LOGIC_VECTOR (2 downto 0);
           Reg_Sel_2 : out STD_LOGIC_VECTOR (2 downto 0);
           Add_Sub_Sel : out STD_LOGIC;
           -- RCA_3_EN : out STD_LOGIC := '0';
           Jump_Flag : out STD_LOGIC;
           Jump_Address : out STD_LOGIC_VECTOR (2 downto 0));
end component;


SIGNAL Instruction : STD_LOGIC_VECTOR (11 downto 0);
SIGNAL Reg_Check_J, Im_Val : STD_LOGIC_VECTOR (3 downto 0);
SIGNAL Reg_EN, Reg_Sel_1, Reg_Sel_2, Jump_Address : STD_LOGIC_VECTOR (2 downto 0);
SIGNAL Load_Sel, Add_Sub_Sel, Jump_Flag : STD_LOGIC;

begin

    UUT : Instruction_Decoder PORT MAP (
        Instruction => Instruction,
        Reg_Check_J => Reg_Check_J,
        Reg_EN => Reg_EN,
        Load_Sel => Load_Sel,
        Im_Val => Im_Val,
        Reg_Sel_1 => Reg_Sel_1,
        Reg_Sel_2 => Reg_Sel_2,
        Add_Sub_Sel => Add_Sub_Sel,
        Jump_Flag => Jump_Flag,
        Jump_Address => Jump_Address);

    process begin

        -- 210250 -> 11 0011 0101 0100 1010
        -- 210661 -> 11 0011 0110 1110 0101
        Instruction <= "100010001010";
        wait for 200ns;

        Instruction <= "000010100000";
        wait for 200ns;

        Instruction <= "010010000000";
        wait for 200ns;

        Instruction <= "110010100000";
        Reg_Check_J <= "0000";
        wait for 200ns;

    end process;

end Behavioral;
```

- ## Processor

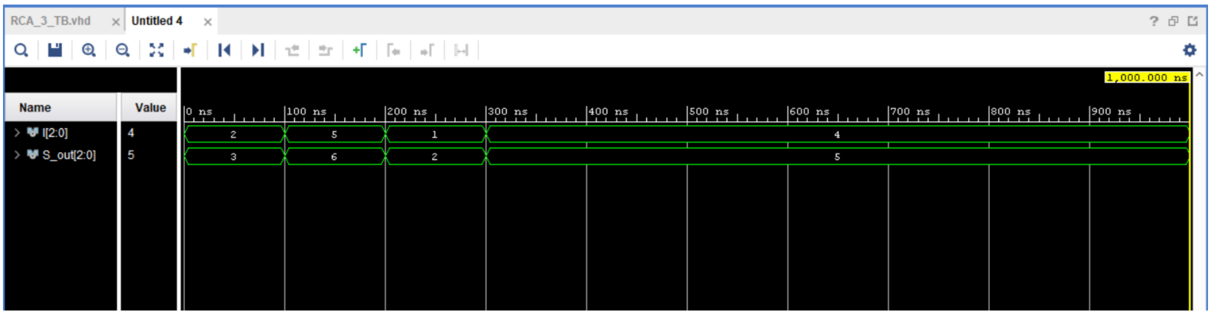
```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date: 06/07/2023 10:08:06 PM
-- Design Name:
-- Module Name: Processor_TB - Behavioral
-- Project Name:
-- Target Devices:
-- Tool Versions:
-- Description:
--
-- Dependencies:
--
-- 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 Processor_TB is
--  Port ( );
end Processor_TB;

architecture Behavioral of Processor_TB is

component NanoProcessor
    Port ( CLK : in STD_LOGIC;
           Res : in STD_LOGIC;
           Zero: out STD_LOGIC;
           Overflow: out STD_LOGIC;
           Reg7_LED: out STD_LOGIC_VECTOR (3 downto 0);
           Reg7_7Seg : out STD_LOGIC_VECTOR (6 downto 0);
           Anode : out STD_LOGIC_VECTOR (3 downto 0):= "0111");
end component;

SIGNAL CLK, Res : STD_LOGIC := '0';
SIGNAL Overflow, Zero: STD_LOGIC;
SIGNAL Anode, Reg7_LED: STD_LOGIC_VECTOR (3 downto 0);
SIGNAL Reg7_7Seg : STD_LOGIC_VECTOR (6 downto 0);

begin

UUT : NanoProcessor PORT MAP (
    CLK => CLK,
    Res => Res,
    Zero => Zero,
    Overflow => Overflow,
    Reg7_LED => Reg7_LED,
    Reg7_7Seg => Reg7_7Seg,
    Anode => Anode);

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

    process begin

        Res <= '1';
        wait for 100ns;

        Res <= '0';
        wait;

    end process;

end Behavioral;
```

Timing Diagrams

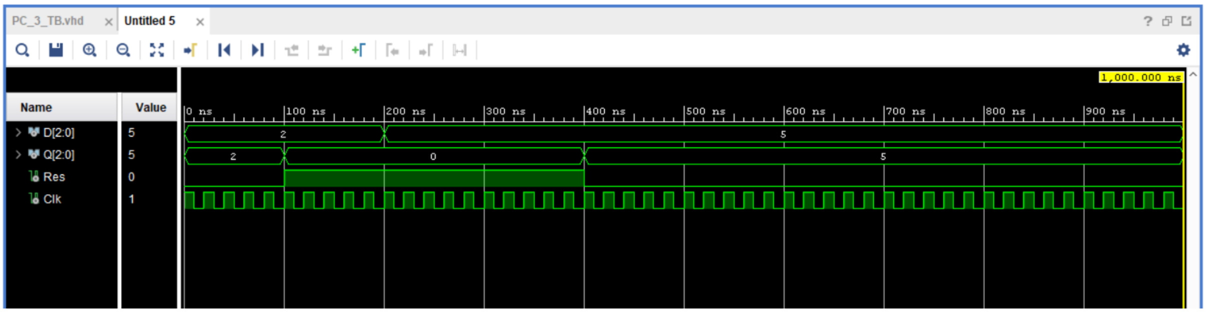
- ### 4-bit Add/Subtract unit



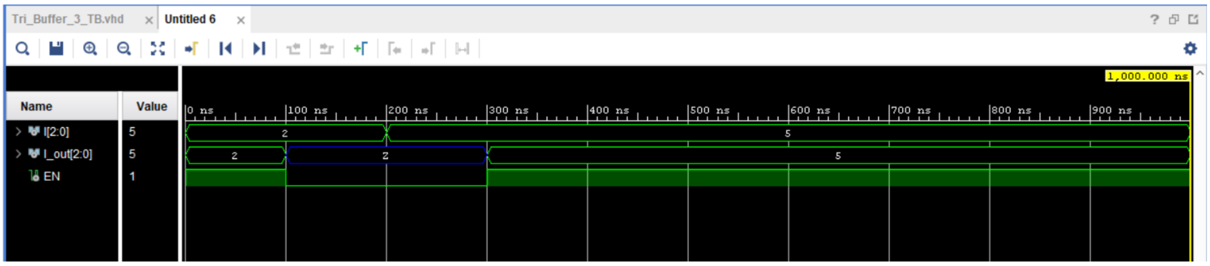
- **3-bit adder**



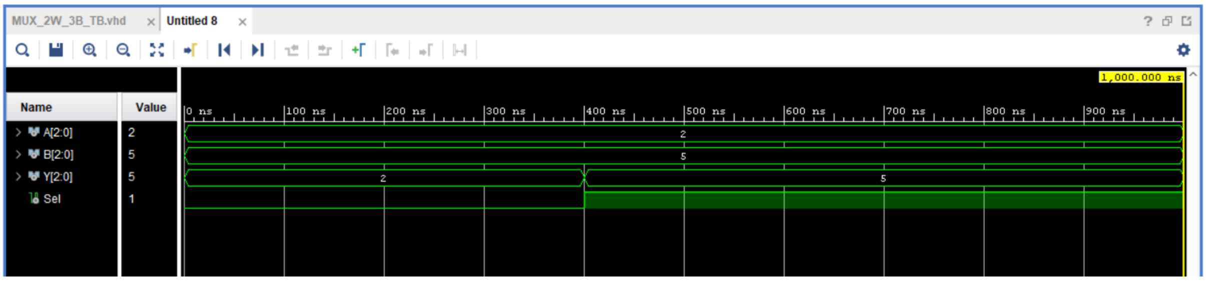
- **3-bit Program Counter (PC)**



- **tri-state buffer**



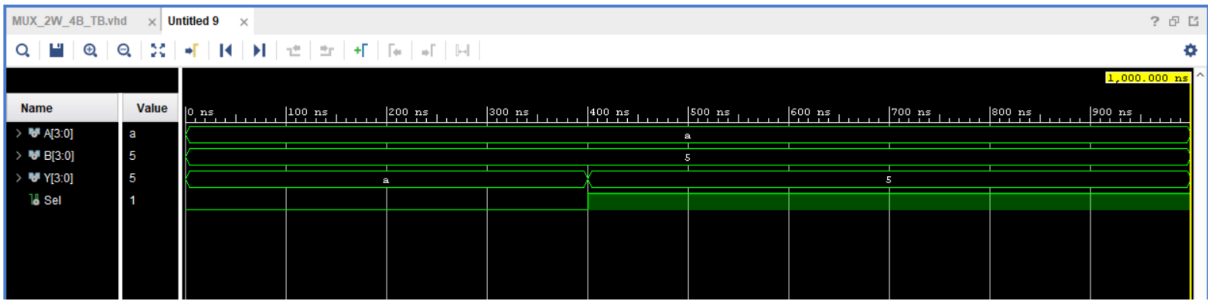
- **2-way 3-bit multiplexer**



MUX_2W_3B_TB.vhd × Untitled 8 ×

Name	Value
A[2:0]	2
B[2:0]	5
Y[2:0]	5
Sel	1

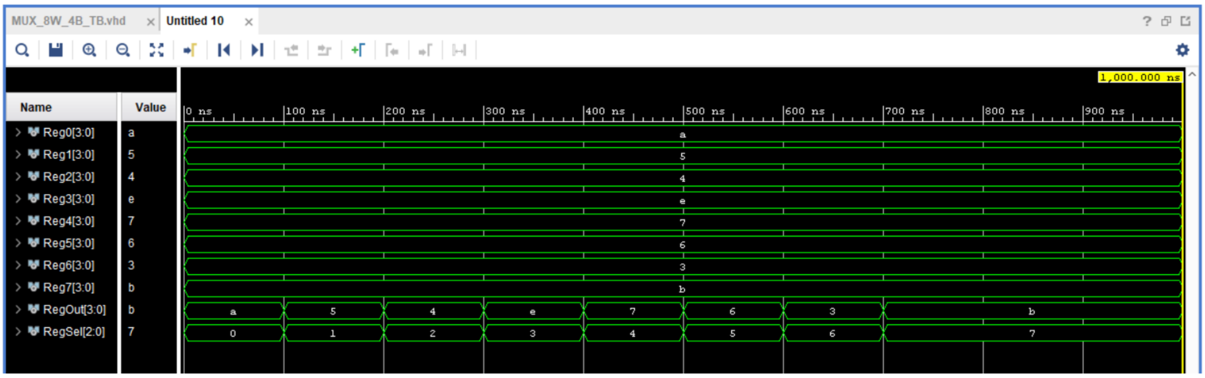
- **2-way 4-bit multiplexer**



MUX_2W_4B_TB.vhd × Untitled 9 ×

Name	Value
A[3:0]	a
B[3:0]	5
Y[3:0]	5
Sel	1

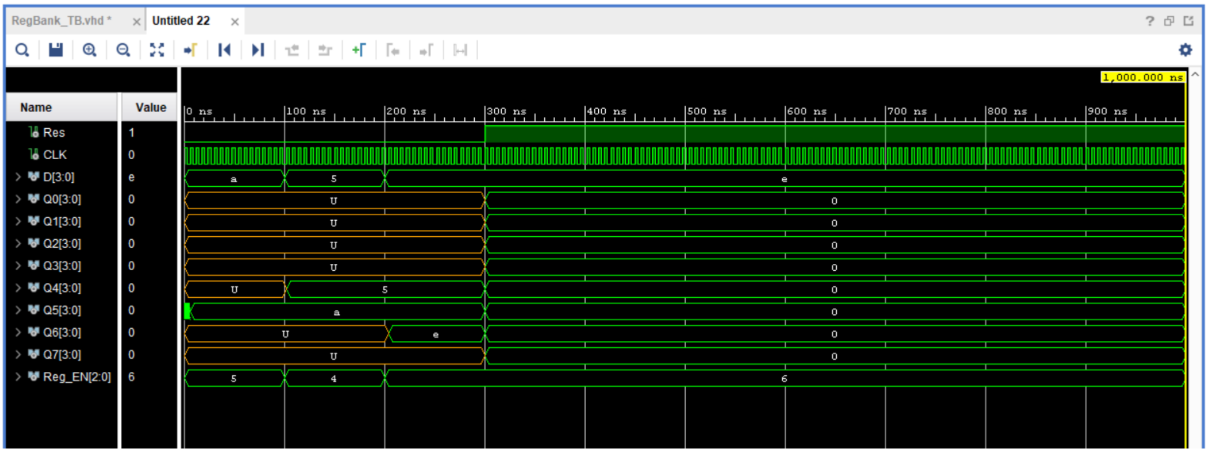
- **8-way 4-bit multiplexer**



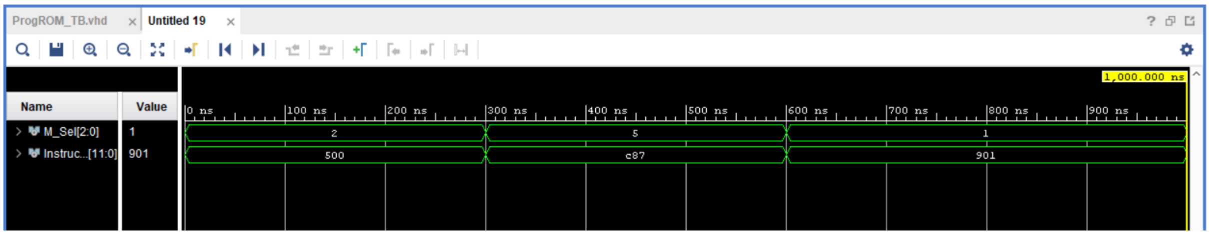
MUX_8W_4B_TB.vhd × Untitled 10 ×

Name	Value
Reg0[3:0]	a
Reg1[3:0]	5
Reg2[3:0]	4
Reg3[3:0]	e
Reg4[3:0]	7
Reg5[3:0]	6
Reg6[3:0]	3
Reg7[3:0]	b
RegOut[3:0]	b
RegSel[2:0]	7

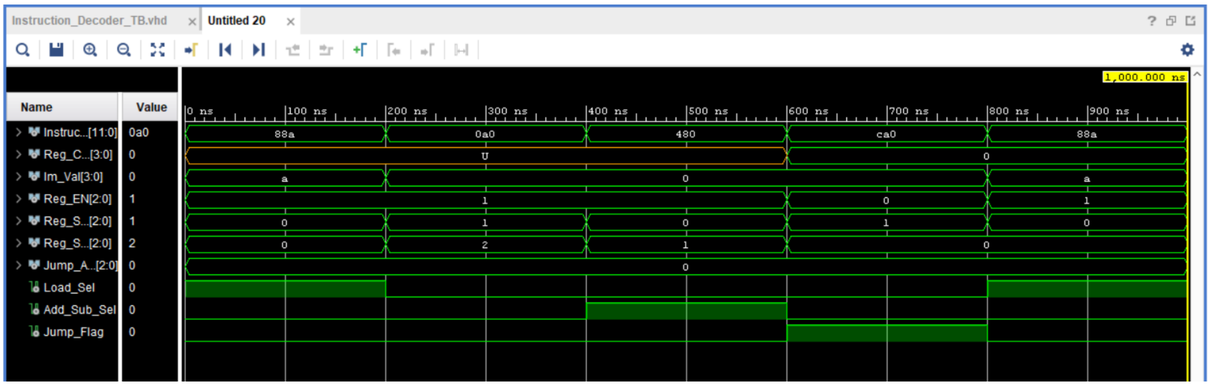
- **Register Bank**



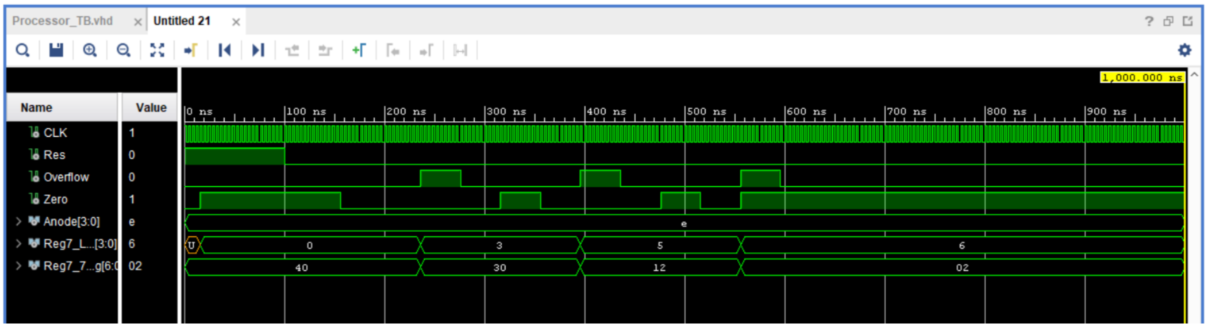
- **Program ROM**



- **Instruction Decoder**



- **Processor**



Conclusions

- We need to hardcode assembly instructions as binary values because the microprocessor only understands machine language.
- Simulating and testing each and every component helped us to make sure the correct functionality of our design.
- Since some of the components were already designed in previous labs, we only had to develop them further without starting from scratch.
- When designing circuits, we can use buses instead of many parallel wires.
- This project helped us to understand the functionality of a microprocessor

Contribution

210250D (10 hours)	210661M(10 hours)
4-bit Add/Subtract Unit	3-bit Adder
2-way 3-bit Multiplexer	3-bit Program Counter
2-way 4-bit Multiplexer	Register Bank
8-way 4-bit Multiplexer	Instruction Decoder
Program ROM	
Tri-state buffer	