CS 1050 Computer Organization and Digital Design

Lab 9-10 Nano processor Design Competition

Group Number: 23

Group members:

1. 220023U – Amarasinghe A.A.D.H.S.
2. 220503R – Ranasinghe P.I.
3. 220520P – Rathnayaka R.L.M.P.
4. 220548H – Samarakoon E.S.P.A.

Content

Contribution of each team member to the project

Member	Individual Contribution	Time Spent
220023U	Components made: • 4-bit Add/Subtract Unit • 3-bit adder • Program ROM • Top-level design Adding some additional features (Overflow flag) Creating the final report	28 hours
220503R	Components made: • 3-bit program counter • Multiplexers • Register Bank • Instruction Decoder • Top-level design Optimizing Designs and adding some additional features(Sign flag)	33 hours
220520P	Components made: • 4-bit Add/Subtract Unit • 3-bit adder • Program ROM • Instruction Decoder • Top-level design Adding some additional features (Overflow flag)	29 hours
220548H	Components made: • 3-bit program counter • Multiplexers • Register Bank • Instruction Decoder • Top-level design	31 hours

Introduction

The assigned laboratory task encompasses creating a 4-bit nano processor capable of executing a basic set of instructions. This involves developing various components such as an Add/Subtract unit, a Program Counter, multiplexers, a Register Bank, an Instruction Decoder, and a Program ROM. The goal is to implement the provided instruction set and construct a functional nanoprocessor circuit.

Assembly program and its machine code representation

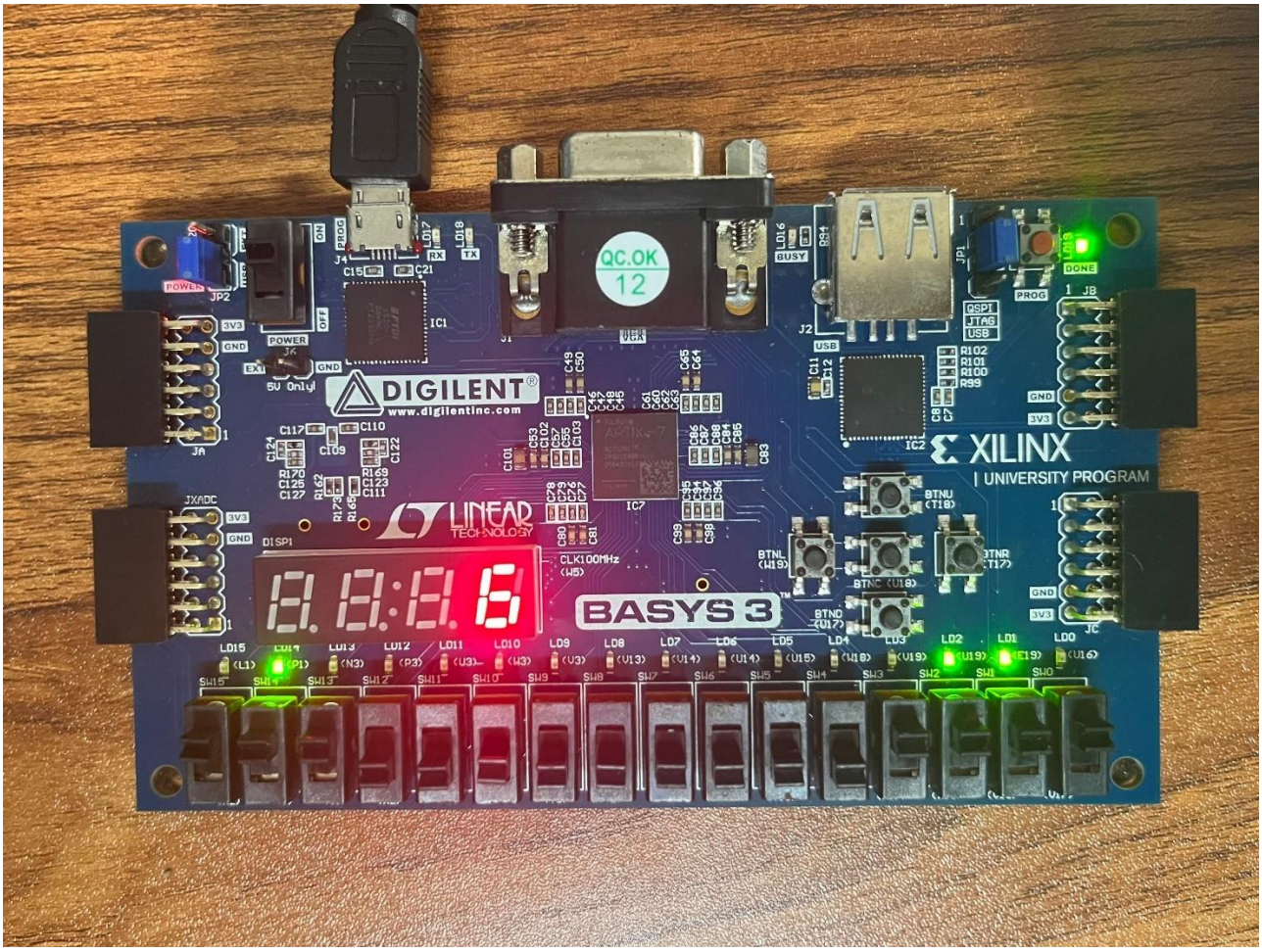
Assembly program:

```
MOVI   R2,2   ;   R2 <-- 2

MOVI   R3,3   ;   R3 <-- 3

MOVI   R7,1   ;   R7 <-- 1

ADD   R7,R1   ;   R7 <-- R7 + R1

ADD   R7,R2   ;   R7 <-- R7 + R2

JZR   R0,5   ;    If R0 = 0 jump to line 5
```

Machine code representation:

Instruction	Description	Machine code representation
MOVI R2,2	Move immediate value 2 to register R2	100010000010
MOVI R3,3	Move immediate value 3 to register R3	100100000011
MOVI R7,1	Move immediate value 1 to register R7	101110000001
ADD R7,R1	Add values in registers R7 and R1 and store the result in R7	001110010000
ADD R7,R2	Add values in registers R7 and R2 and store the result in R7	001110100000
JZR R0,5	Jump to line 5 if value in register R0 is 0	110000000101

Specific instructions for the practical operation of the machine



Allocated reset button

- The btnD button is allocated for the reset process.
- We reduced the clock speed of the internal clock of the board from 100MHz to 0.5MHz. Therefore, press and hold the reset button for at least 2-3 seconds to reset the whole program.

LED signal/ 7 -segment display mapping

LED0-LED3 Output of R7 register in Register Bank

- LED0-LED3 → Magnitude bits

LED12-LED15 Flags of 4-bit Add/Subtract Unit

- LED12 → Sign Flag
- LED13 → Overflow Flag
- LED14 → Zero Flag
- LED15 → Carry Flag

7 Segment Display

- In our design, the ***rightmost segment*** of the 7-Segment display will display the output from the R7 register in the Register Bank. The other three segments will not light up.
- It will display the magnitude of the 4-bit 2's complement number stored in R7 register.

Expected behavior of the program

- Each execution process needs a manual reset to start executing the program in the ROM.
- As the Program Counter and Registers should be initialized to Zero at the start, Program Counter will wait for a reset signal to change its count to start executing the program in the ROM.

Slice Logic and Primitives

Slice Logic

Site Type	Used	Fixed	Available	Util%
Slice LUTs*	54	0	20800	0.26
LUT as Logic	54	0	20800	0.26
LUT as Memory	0	0	9600	0.00
Slice Registers	75	0	41600	0.18
Register as Flip Flop	65	0	41600	0.16
Register as Latch	10	0	41600	0.02
F7 Muxes	0	0	16300	0.00
F8 Muxes	0	0	8150	0.00

Primitives

Ref Name	Used	Functional Category
FDRE	37	Flop & Latch
FDCE	28	Flop & Latch
LUT4	21	LUT
OBUF	19	IO
LUT3	19	LUT
LUT5	16	LUT
LUT6	12	LUT
LDCE	10	Flop & Latch
CARRY4	8	CarryLogic
LUT2	3	LUT
IBUF	2	IO
BUFG	2	Clock

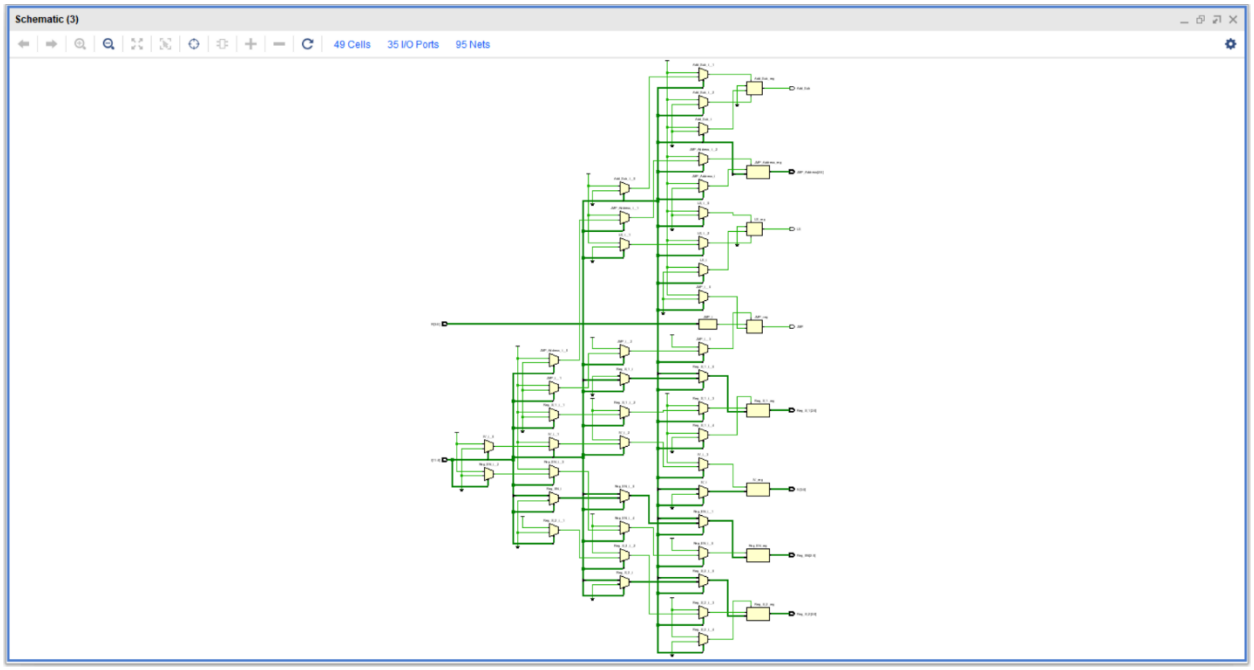
All VHDL codes

Instruction Decoder:

```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date: 17.04.2024 10:26:09
-- 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 ( I : in STD_LOGIC_VECTOR (11 downto 0);  --Instruction Bus
           R : in STD_LOGIC_VECTOR (3 downto 0);  -- Register check for jump
           Reg_EN : out STD_LOGIC_VECTOR (2 downto 0);  --Register enable
           LS : out STD_LOGIC; --Load select
           IV : out STD_LOGIC_VECTOR (3 downto 0);  -- Immediate value
           Reg_S_1 : out STD_LOGIC_VECTOR (2 downto 0);  --Register select 1
           Reg_S_2 : out STD_LOGIC_VECTOR (2 downto 0);  --Register select 2
           Add_Sub : out STD_LOGIC;  --Abb/Sub select
           JMP : out STD_LOGIC;  -- Jump Flag
           JMP_Address : out STD_LOGIC_VECTOR (2 downto 0));  -- Address to
jump
```

```
end Instruction_Decoder;

architecture Behavioral of Instruction_Decoder is

signal opcode: STD_LOGIC_VECTOR (1 downto 0);
signal Reg_A: STD_LOGIC_VECTOR (2 downto 0);
signal Reg_B: STD_LOGIC_VECTOR (2 downto 0);
signal data: STD_LOGIC_VECTOR (3 downto 0);

begin

opcode<=I(11 downto 10);
Reg_A<=I(9 downto 7);
Reg_B<=I(6 downto 4);
data<=I(3 downto 0);

process (opcode,Reg_A,Reg_B,data,R)
begin

    if opcode="10" then
            JMP<='0';
            LS<='0';
            Reg_EN<=Reg_A;
            IV<=data;

            Reg_S_1 <= "000";
            Reg_S_2 <= "000";
            Add_Sub <= '0';
            JMP_Address <= "000";

    elsif opcode="00" then
            JMP<='0';
            LS<='1';
            Reg_EN<=Reg_A;
            Reg_S_1<=Reg_A;
            Reg_S_2<=Reg_B;
            Add_Sub<='0';

            JMP_Address <= "000";
            IV<="0000";

    elsif opcode="01" then
            JMP<='0';
            LS<='1';
            Reg_EN<=Reg_A;
            Reg_S_1<="000";
            Reg_S_2<=Reg_A;
            Add_Sub<='1';
```

```
            JMP_Address <= "000";
            IV<="0000";

    elsif opcode="11" then
            JMP_Address<=data(2 downto 0);
            Reg_S_1<=Reg_A;

            if R="0000" then
                JMP<='1';
            else
                JMP<='0';
            end if;

            LS  <= '0';
            Add_Sub<='0';
            Reg_EN <= "000";
            IV <= "0000";
            Reg_S_2 <= "000";


    end if;

end process;

end Behavioral;
```



Half Adder:

```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date: 15.04.2024 21:25:20
-- Design Name:
-- Module Name: HA - 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 HA is
    Port ( A : in STD_LOGIC;
           B : in STD_LOGIC;
           S : out STD_LOGIC;
           C : out STD_LOGIC);
end HA;

architecture Behavioral of HA is

begin

S <= A XOR B;
C <= A AND B;


end Behavioral;
```

Full Adder:

```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date: 15.04.2024 21:27:24
-- Design Name:
-- Module Name: FA - 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 FA is
    Port ( A : in STD_LOGIC;
           B : in STD_LOGIC;
           C_in : in STD_LOGIC;
           S : out STD_LOGIC;
           C_out : out STD_LOGIC);
end FA;

architecture Behavioral of FA is

component HA
        port (
        A: in std_logic;
        B: in std_logic;
        S: out std_logic;
```

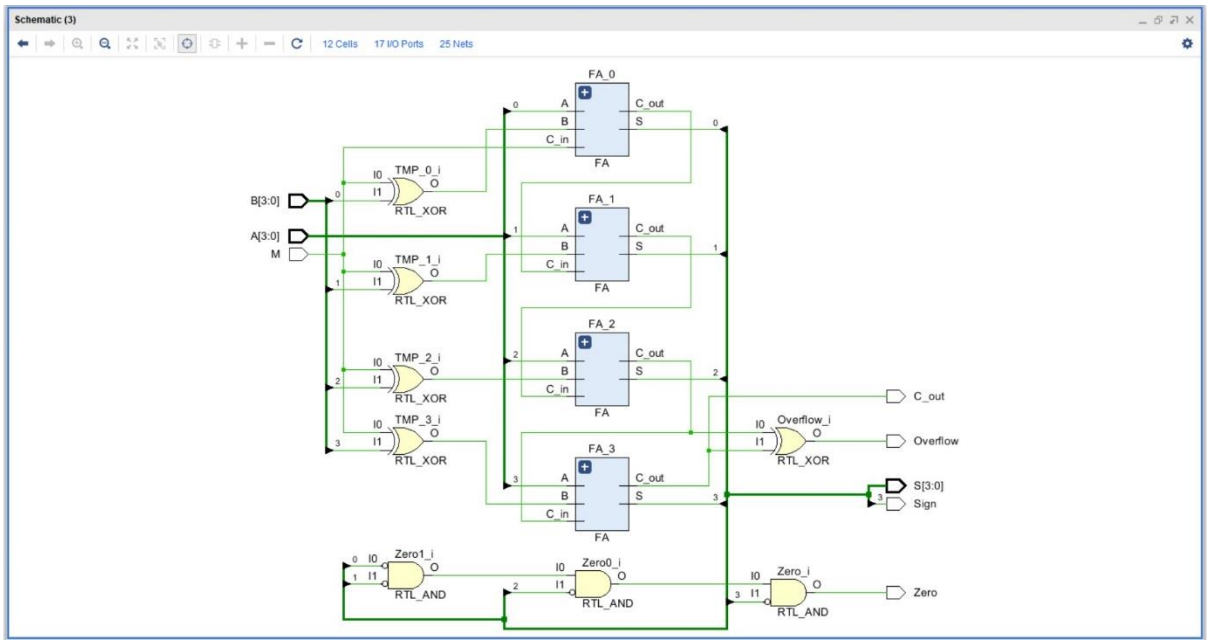
```
        C: out std_logic);
    end component;

SIGNAL HA0_S, HA0_C, HA1_S, HA1_C : std_logic;

begin

HA_0 : HA
    port map (
    A => A,
    B => B,
    S => HA0_S,
    C => HA0_C);

 HA_1 : HA
     port map (
    A => HA0_S,
    B => C_in,
    S => HA1_S,
    C => HA1_C);

 S <= HA1_S;
 C_out <= HA0_C OR HA1_C;


end Behavioral;
```

4-bit Add/Subtract unit:

```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date: 15.04.2024 22:43:30
-- Design Name:
-- Module Name: Add_Sub_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 Add_Sub_4bit is
    Port ( A : in STD_LOGIC_VECTOR (3 downto 0);
           B : in STD_LOGIC_VECTOR (3 downto 0);
           M : in STD_LOGIC;
           S : out STD_LOGIC_VECTOR (3 downto 0);
           C_out : out STD_LOGIC;
           Overflow : out STD_LOGIC;
           Sign: out STD_LOGIC;
           Zero : out STD_LOGIC );

end Add_Sub_4bit;

architecture Behavioral of Add_Sub_4bit 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_C, FA1_C, FA2_C, FA3_C  : std_logic;
signal TMP: std_logic_vector(3 downto 0);
signal sum: std_logic_vector(3 downto 0);
signal C_output:  std_logic;

begin

FA_0 : FA
    port map (
        A => A(0),
        B => TMP(0),
        C_in => M,
        S => sum(0),
        C_out => FA0_C);
 FA_1 : FA
    port map (
        A => A(1),
        B => TMP(1),
        C_in => FA0_C,
        S => sum(1),
        C_out => FA1_C);

FA_2 : FA
     port map (
        A => A(2),
        B => TMP(2),
        C_in => FA1_C,
        S => sum(2),
        C_out => FA2_C);
FA_3 : FA
     port map (
        A => A(3),
        B => TMP(3),
        C_in => FA2_C,
        S => sum(3),
        C_out => C_output);

TMP(0)<= M XOR B(0);
TMP(1)<= M XOR B(1);
TMP(2)<= M XOR B(2);
```

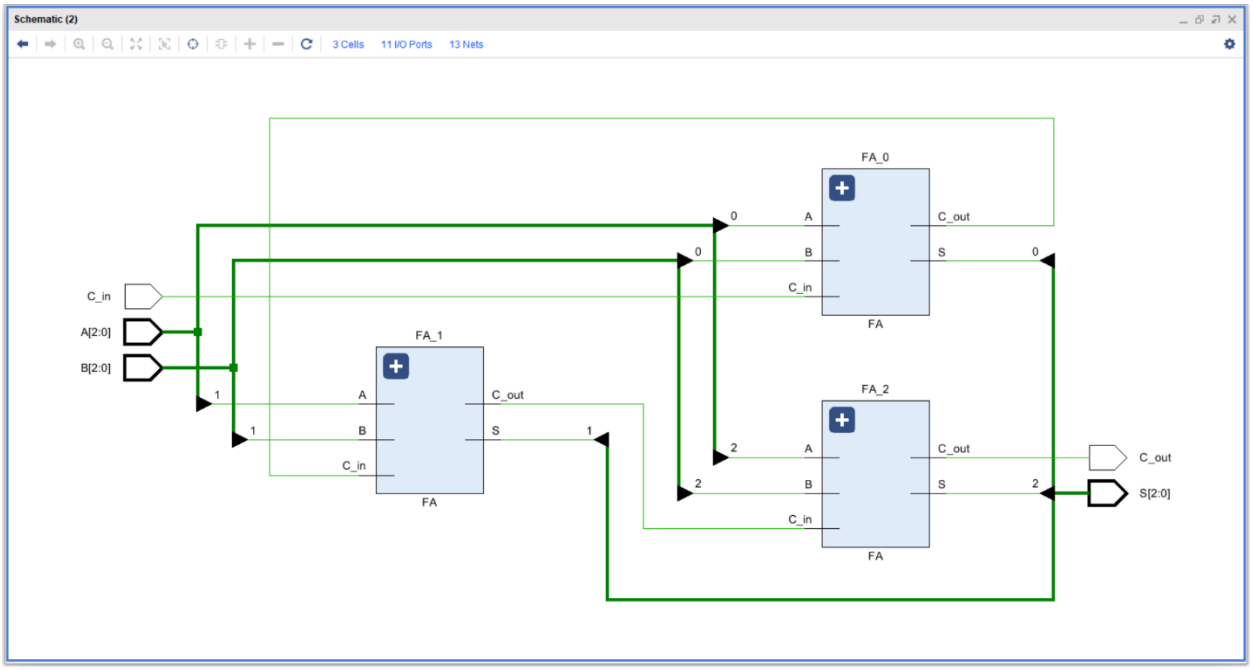
```
TMP(3)<= M XOR B(3);

C_out<=C_output;
S<=sum;

Overflow <= FA2_C XOR C_output;
Zero <= (NOT sum(0)) AND (NOT sum(1)) AND (NOT sum(2)) AND (NOT sum(3));
Sign<=sum(3);

end Behavioral;
```



3-bit adder:

```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
-- Create Date: 15.04.2024 21:39:15
-- Design Name:
-- Module Name: Adder_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 Adder_3bit is
    Port ( A : in STD_LOGIC_VECTOR (2 downto 0);
           B : in STD_LOGIC_VECTOR (2 downto 0);
           C_in : in STD_LOGIC;
           C_out : out STD_LOGIC;
           S : out STD_LOGIC_VECTOR (2 downto 0));
end Adder_3bit;

architecture Behavioral of Adder_3bit 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 => A(0),
        B => B(0),
        C_in => C_in,
        S => S(0),
        C_Out => FA0_C);


 FA_1 : FA
    port map (
        A => A(1),
        B => B(1),
        C_in => FA0_C,
        S => S(1),
        C_Out => FA1_C);

FA_2 : FA
     port map (
        A => A(2),
        B => B(2),
        C_in => FA1_C,
        S => S(2),
        C_Out => C_out);


end Behavioral;
```



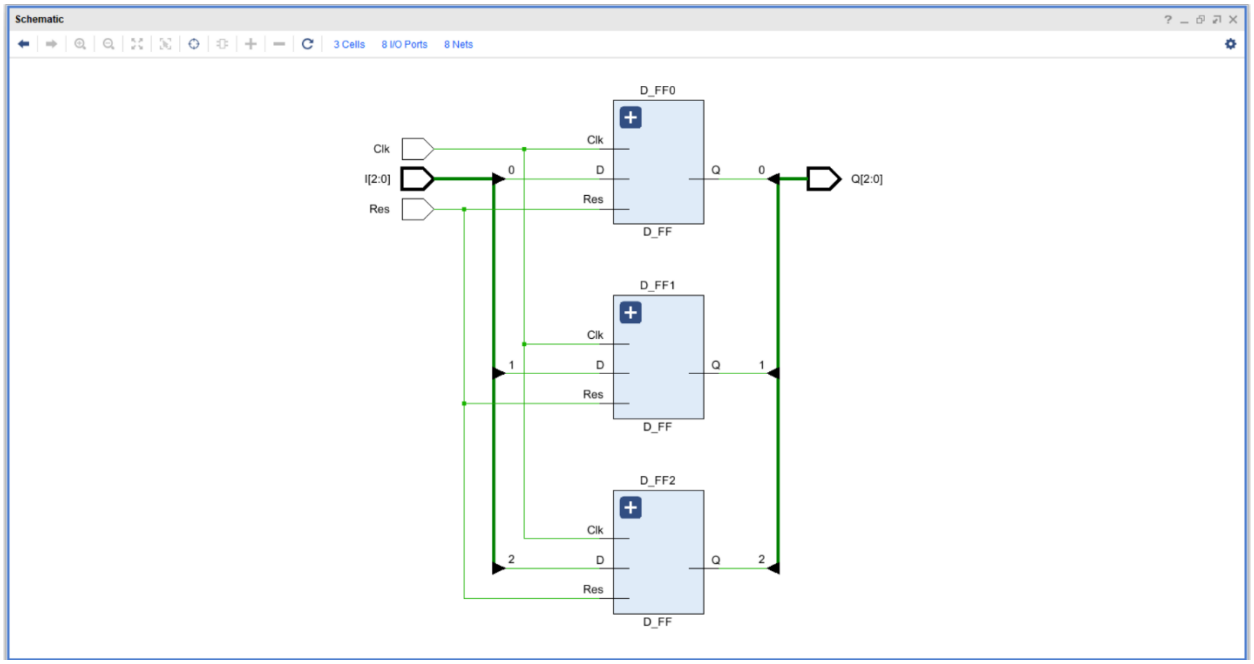
D flipflop :

```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
-- Create Date: 15.04.2024 11:38:59
-- Design Name:
-- Module Name: D_FF - 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 D_FF is
    Port ( D : in STD_LOGIC;
           Res : in STD_LOGIC;
           Clk : in STD_LOGIC;
           Q : out STD_LOGIC;
           Qbar : out STD_LOGIC);
end D_FF;

architecture Behavioral of D_FF is

begin

process (Clk) begin
    if (rising_edge(Clk)) then
        if Res='1' then
            Q <= '0';
            Qbar <= '1';
        else
            Q <= D;
            Qbar <= not D;
        end if;
    end if;
end process;

end Behavioral;
```

3-bit Program Counter (PC):

```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
-- Create Date: 15.04.2024 12:58:44
-- Design Name:
-- Module Name: PC - 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 is
    Port ( Res : in STD_LOGIC;
           Clk : in STD_LOGIC;
           I : in STD_LOGIC_VECTOR (2 downto 0);
           Q : out STD_LOGIC_VECTOR (2 downto 0));
end PC;

architecture Behavioral of PC 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 => I(0),
  Res => Res,
  Clk => Clk,
  Q => Q(0));

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

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

end Behavioral;
```



2-to-4 decoder:

```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date: 13.04.2024 00:48:08
-- Design Name:
-- Module Name: Decoder_2_to_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 Decoder_2_to_4 is
    Port ( I : in STD_LOGIC_VECTOR (1 downto 0);
           EN : in STD_LOGIC;
           Y : out STD_LOGIC_VECTOR (3 downto 0));
end Decoder_2_to_4;

architecture Behavioral of Decoder_2_to_4 is

begin

    Y(3) <=  EN AND I(1) AND I(0);
    Y(2) <=  EN AND  I(1) AND (NOT I(0));
    Y(1) <=  EN AND (NOT I(1)) AND I(0);
    Y(0) <=  EN AND (NOT I(1)) AND (NOT I(0));

end Behavioral;
```

3-to-8 decoder:

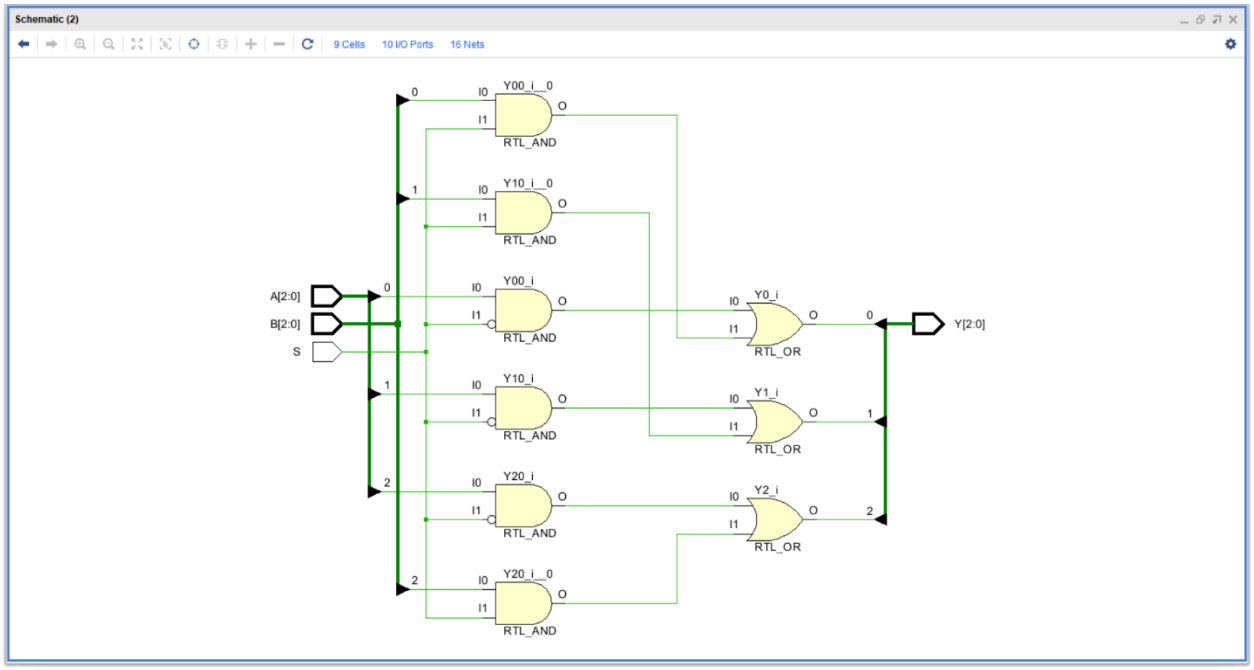
```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date: 13.04.2024 00:51:16
-- Design Name:
-- Module Name: Decoder_3_8 - Behavioral
-- Project Name:
-- Target Devices:
-- Tool Versions:
-- Description:
--
-- Dependencies:
--
-- 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 (1 downto 0);
EN: in STD_LOGIC;
Y: out STD_LOGIC_VECTOR (3 downto 0));
```

```
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

Decoder_2_to_4_0 : Decoder_2_to_4
port map(
I => I0,
EN => en0,
Y => Y0 );

Decoder_2_to_4_1 : Decoder_2_to_4
port map(
I => I1,
EN => en1,
Y => Y1 );

en0 <= NOT(I(2)) AND EN;
en1 <= I(2) 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;
```

2-way 3-bit multiplexer:

```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date: 13.04.2024 01:16:50
-- Design Name:
-- Module Name: Mux_2_way_3_bit - 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_way_3_bit is
    Port ( S : in STD_LOGIC;
           A : in STD_LOGIC_VECTOR (2 downto 0);
           B : in STD_LOGIC_VECTOR (2 downto 0);
           Y : out STD_LOGIC_VECTOR (2 downto 0));
end Mux_2_way_3_bit;

architecture Behavioral of Mux_2_way_3_bit is

signal Y0,Y1,Y2 : STD_LOGIC;

begin
```

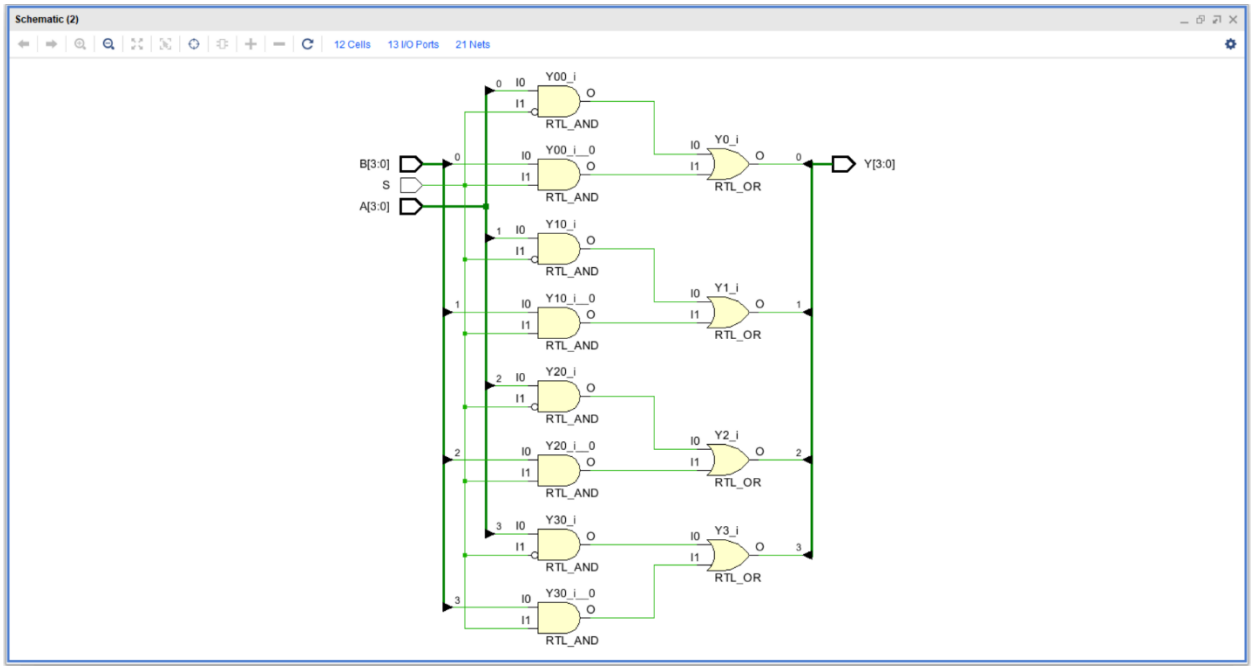
```
Y0 <= ((A(0) AND (NOT S)) OR (B(0) AND S));
Y1 <= ((A(1) AND (NOT S)) OR (B(1) AND S));
Y2 <= ((A(2) AND (NOT S)) OR (B(2) AND S));

Y(0) <= Y0;
Y(1) <= Y1;
Y(2) <= Y2;


end Behavioral;
```



2-way 4-bit multiplexer:

```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date: 13.04.2024 13:03:34
-- Design Name:
-- Module Name: Mux_2_way_4_bit - 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_way_4_bit is
    Port ( S : in STD_LOGIC;
           A : in STD_LOGIC_VECTOR (3 downto 0);
           B : in STD_LOGIC_VECTOR (3 downto 0);
           Y : out STD_LOGIC_VECTOR (3 downto 0));
end Mux_2_way_4_bit;

architecture Behavioral of Mux_2_way_4_bit is

signal Y0,Y1,Y2,Y3 : STD_LOGIC;

begin
```

```
Y0 <= ((A(0) AND (NOT S)) OR (B(0) AND S));
Y1 <= ((A(1) AND (NOT S)) OR (B(1) AND S));
Y2 <= ((A(2) AND (NOT S)) OR (B(2) AND S));
Y3 <= ((A(3) AND (NOT S)) OR (B(3) AND S));

Y(0) <= Y0;
Y(1) <= Y1;
Y(2) <= Y2;
Y(3) <= Y3;

end Behavioral;
```



8-way 1-bit multiplexer:

```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date: 13.04.2024 00:53:53
-- 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);
           D : in STD_LOGIC_VECTOR (7 downto 0);
           EN : in STD_LOGIC;
           Y : out STD_LOGIC);
end Mux_8_to_1;


architecture Behavioral of Mux_8_to_1 is
```

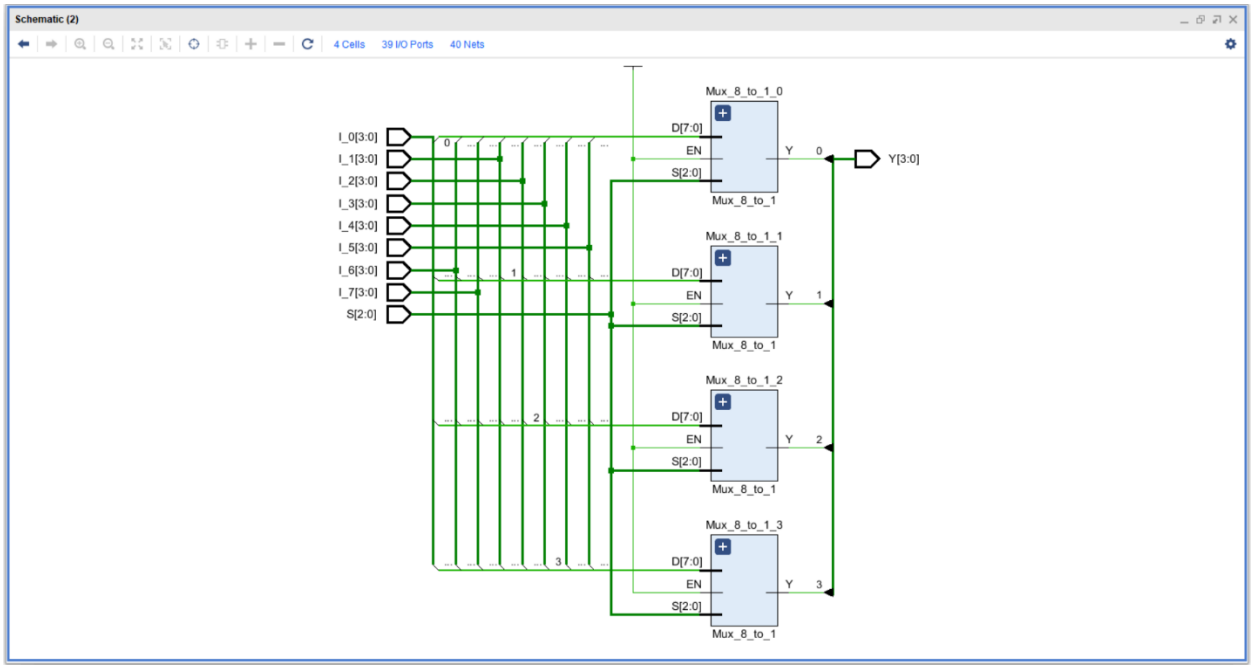
```
component Decoder_3_to_8


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


 end component;

signal Y0: STD_LOGIC_VECTOR(7 downto 0);

begin

Decoder_3_to_8_0 : Decoder_3_to_8
 port map(
 I => S,
 EN => EN,
 Y => Y0

);

 Y <= EN AND ((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)));

end Behavioral;
```

8-way 4-bit multiplexer:

```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
-- Create Date: 13.04.2024 13:24:28
-- Design Name:
-- Module Name: Mux_8_way_4_bit - 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_way_4_bit is
    Port ( S : in STD_LOGIC_VECTOR (2 downto 0);
           I_0 : in STD_LOGIC_VECTOR (3 downto 0);
           I_1 : in STD_LOGIC_VECTOR (3 downto 0);
           I_2 : in STD_LOGIC_VECTOR (3 downto 0);
           I_3 : in STD_LOGIC_VECTOR (3 downto 0);
           I_4 : in STD_LOGIC_VECTOR (3 downto 0);
           I_5 : in STD_LOGIC_VECTOR (3 downto 0);
           I_6 : in STD_LOGIC_VECTOR (3 downto 0);
           I_7 : in STD_LOGIC_VECTOR (3 downto 0);
           Y : out STD_LOGIC_VECTOR (3 downto 0));

end Mux_8_way_4_bit;

architecture Behavioral of Mux_8_way_4_bit is

component Mux_8_to_1 is
    Port ( S : in STD_LOGIC_VECTOR (2 downto 0);
```

```
            D : in STD_LOGIC_VECTOR (7 downto 0);
            EN : in STD_LOGIC;
            Y : out STD_LOGIC);

end component;


signal Y0,Y1,Y2,Y3: STD_LOGIC;

begin

Mux_8_to_1_0 : Mux_8_to_1
 port map(
 S => S,
 D(0)=> I_0(0),
 D(1)=> I_1(0),
 D(2)=> I_2(0),
 D(3)=> I_3(0),
 D(4)=> I_4(0),
 D(5)=> I_5(0),
 D(6)=> I_6(0),
 D(7)=> I_7(0),
 EN => '1',
 Y => Y0
);

 Mux_8_to_1_1 : Mux_8_to_1
 port map(
 S => S,
 D(0)=> I_0(1),
 D(1)=> I_1(1),
 D(2)=> I_2(1),
 D(3)=> I_3(1),
 D(4)=> I_4(1),
 D(5)=> I_5(1),
 D(6)=> I_6(1),
 D(7)=> I_7(1),
 EN => '1',
 Y => Y1
);

 Mux_8_to_1_2 : Mux_8_to_1
  port map(
  S => S,
  D(0)=> I_0(2),
  D(1)=> I_1(2),
  D(2)=> I_2(2),
  D(3)=> I_3(2),
  D(4)=> I_4(2),
```

```
  D(5)=> I_5(2),
  D(6)=> I_6(2),
  D(7)=> I_7(2),
  EN => '1',
  Y => Y2
);

 Mux_8_to_1_3 : Mux_8_to_1
   port map(
   S => S,
   D(0)=> I_0(3),
   D(1)=> I_1(3),
   D(2)=> I_2(3),
   D(3)=> I_3(3),
   D(4)=> I_4(3),
   D(5)=> I_5(3),
   D(6)=> I_6(3),
   D(7)=> I_7(3),
   EN => '1',
   Y => Y3
);

Y(0) <= Y0;
Y(1) <= Y1;
Y(2) <= Y2;
Y(3) <= Y3;

end Behavioral;
```



Register:

```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date: 16.04.2024 10:39:00
-- Design Name:
-- Module Name: Reg - 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 is
    Port ( D : in STD_LOGIC_VECTOR (3 downto 0);
           En : in STD_LOGIC;
           Res : in STD_LOGIC;
           Clk : in STD_LOGIC;
           Q : out STD_LOGIC_VECTOR (3 downto 0));
end Reg;

architecture Behavioral of Reg is

begin

process (Clk,Res) begin
```

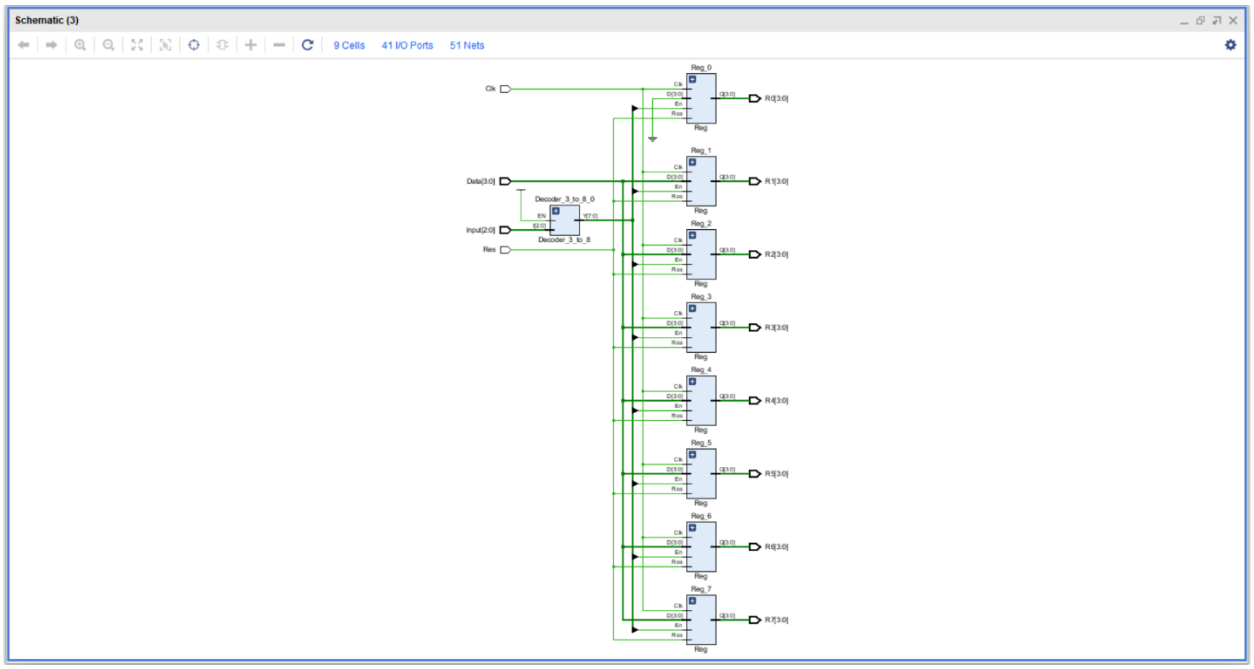
```
    if (Res='0') then
        if (rising_edge(Clk)) then
            if En = '1' then
                Q <= D;
            end if;
        end if;
    else
        Q<= "0000";
    end if;


end process;

end Behavioral;
```

Register Bank:

```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date: 15.04.2024 20:41:44
-- Design Name:
-- Module Name: Register_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 Register_Bank is
    Port ( Res : in STD_LOGIC;
           Clk : in STD_LOGIC;
           Input : in STD_LOGIC_VECTOR (2 downto 0);
           Data : in STD_LOGIC_VECTOR (3 downto 0);
           R0 : out STD_LOGIC_VECTOR (3 downto 0);
           R1 : out STD_LOGIC_VECTOR (3 downto 0);
           R2 : out STD_LOGIC_VECTOR (3 downto 0);
           R3 : out STD_LOGIC_VECTOR (3 downto 0);
           R4 : out STD_LOGIC_VECTOR (3 downto 0);
           R5 : out STD_LOGIC_VECTOR (3 downto 0);
           R6 : out STD_LOGIC_VECTOR (3 downto 0);
           R7 : out STD_LOGIC_VECTOR (3 downto 0));
```

```
end Register_Bank;

architecture Behavioral of Register_Bank is

 component 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 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 Y0 : STD_LOGIC_VECTOR (7 downto 0);

begin

Decoder_3_to_8_0 : Decoder_3_to_8
 port map(
 I => Input,
 EN => '1',
 Y => Y0 );

Reg_0 : reg
     port map(
         D =>"0000",
         Res => Res,
         En =>Y0(0),
         Clk => Clk,
         Q => R0 );

Reg_1 : reg
     port map(
         D =>Data ,
         Res => Res,
         En =>Y0(1),
         Clk => Clk,
         Q => R1 );

Reg_2 : reg
     port map(
         D =>Data ,
         Res => Res,
```

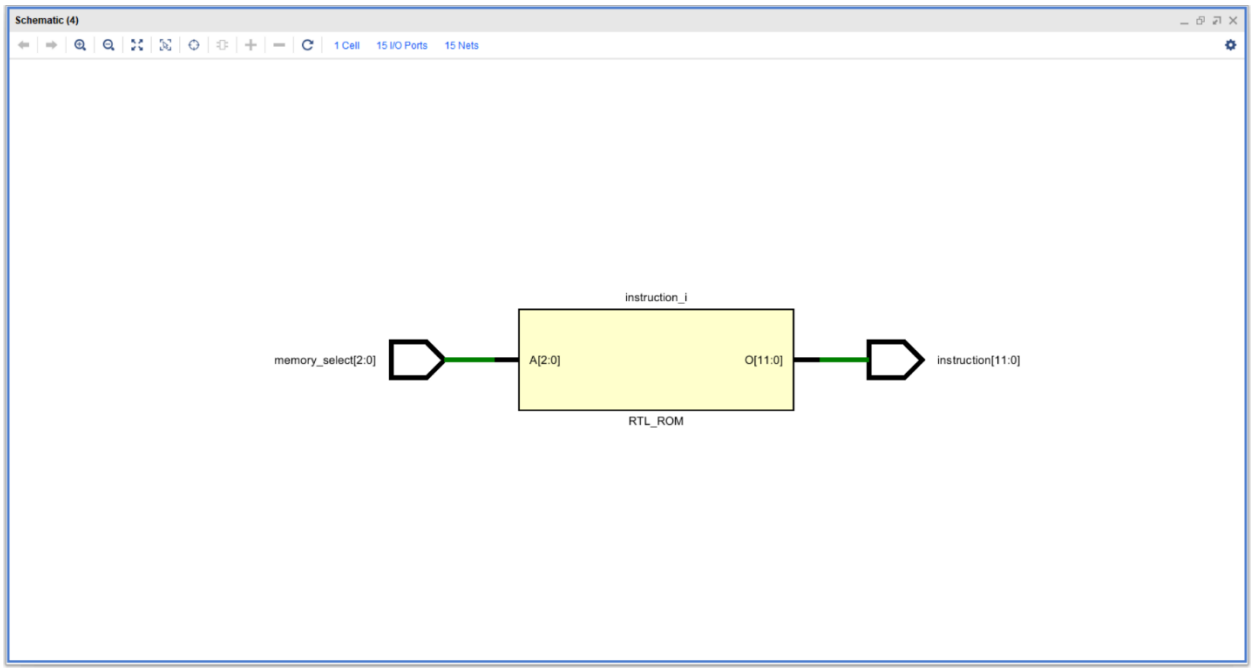
```
          En =>Y0(2),
          Clk => Clk,
          Q => R2 );

Reg_3 : reg
     port map(
          D =>Data ,
          Res => Res,
          En =>Y0(3),
          Clk => Clk,
          Q => R3 );

Reg_4 : reg
     port map(
          D =>Data ,
          Res => Res,
          En =>Y0(4),
          Clk => Clk,
          Q => R4 );

Reg_5 : reg
     port map(
          D =>Data ,
          Res => Res,
          En =>Y0(5),
          Clk => Clk,
          Q => R5 );

Reg_6 : reg
     port map(
          D =>Data,
          Res => Res,
          En =>Y0(6),
          Clk => Clk,
          Q => R6 );

Reg_7 : reg
     port map(
          D =>Data ,
          Res => Res,
          En =>Y0(7),
          Clk => Clk,
          Q => R7 );

end Behavioral;
```


Schematic (3)
9 Cells
41 I/O Ports
51 Nets
Clk
Data[3:0]
Input[2:0]
Res
Decoder_3_to_8_0
Decoder_3_to_8
Reg_0
Reg_1
Reg_2
Reg_3
Reg_4
Reg_5
Reg_6
Reg_7
R0[3:0]
R1[3:0]
R2[3:0]
R3[3:0]
R4[3:0]
R5[3:0]
R6[3:0]
R7[3:0]

Program ROM:

```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date: 16.04.2024 17:46:48
-- Design Name:
-- Module Name: ROM - 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 ROM is
    Port ( memory_select : in STD_LOGIC_VECTOR (2 downto 0);
           instruction : out STD_LOGIC_VECTOR (11 downto 0));
end ROM;

architecture Behavioral of ROM is

type rom_type is array(0 to 7) of std_logic_vector(11 downto 0);

signal PROGRAM_ROM : rom_type := (

"100010000010",  --MOV 2 to R1
"100100000011",  --MOV 3 to R2
"101110000001",  --MOV 1 to R7
```

```
"001110010000",   --ADD R7 to R1 and store in R7
"001110100000",   --ADD R2 to R7 and store in R7
"110000000101",   --Jump to instruction 5 using R0
"000000000000",
"000000000000"
);

begin

instruction <= PROGRAM_ROM(to_integer(unsigned(memory_select)));

end Behavioral;
```



LUT:

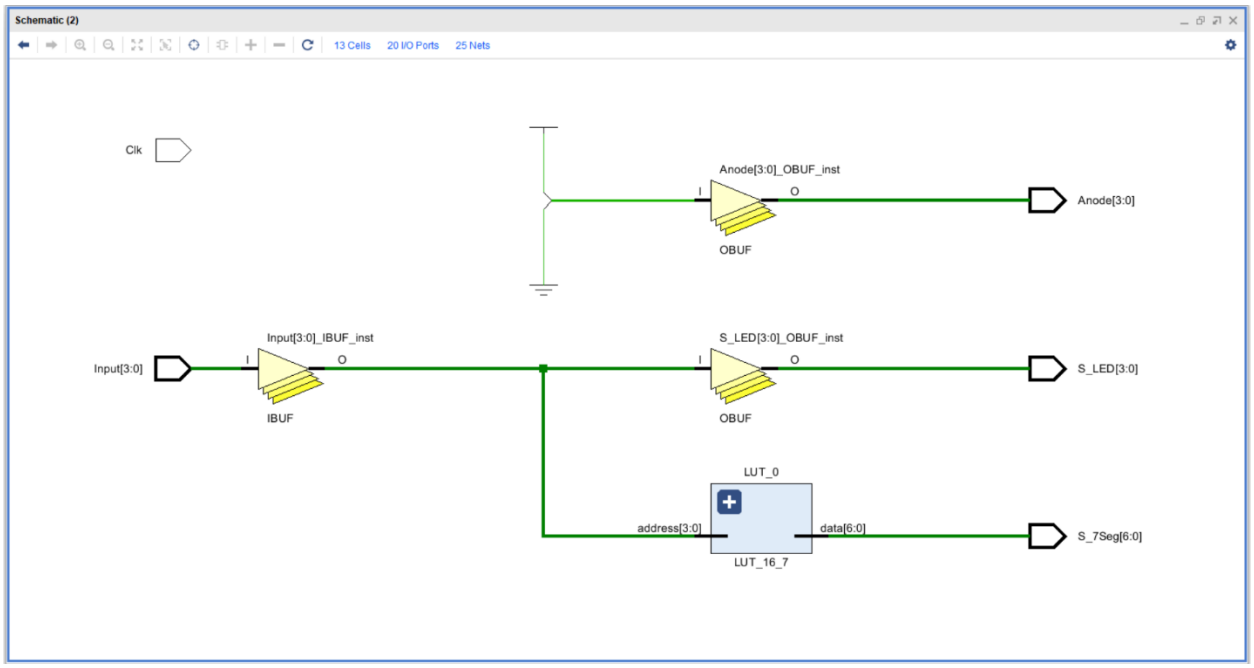
```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date: 17.04.2024 07:53:36
-- Design Name:
-- Module Name: LUT_16_7 - Behavioral
-- Project Name:
-- Target Devices:
-- Tool Versions:
-- Description:
--
-- Dependencies:
--
-- Revision:
-- Revision 0.01 - File Created
-- Additional Comments:
--
----------------------------------------------------------------------------------

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;


-- 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;
```

7 segment display:

```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date: 17.04.2024 08:07:10
-- Design Name:
-- Module Name: Add_Sub_7_seg - 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 R7_7_seg is
    Port ( Input : in STD_LOGIC_VECTOR (3 downto 0);
           Clk : in STD_LOGIC;
           S_LED : out STD_LOGIC_VECTOR (3 downto 0);
           S_7Seg : out STD_LOGIC_VECTOR (6 downto 0);
           Anode : out STD_LOGIC_VECTOR (3 downto 0));
end R7_7_seg;

architecture Behavioral of R7_7_seg is

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

```
begin

LUT_0: LUT_16_7
   port map(
      address=>Input,
      data=>S_7Seg
);

S_LED <= Input;
Anode <= "1110";

end Behavioral;
```



Slow Clock:

```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date: 15.04.2024 10:37:44
-- 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;
```

Top-level design (Nano processor):

```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
-- Create Date: 17.04.2024 18:54:05
-- 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;
           Carry_out:out STD_LOGIC;
           Overflow : out STD_LOGIC;
           Sign: out STD_LOGIC;
           Zero : out STD_LOGIC;
           S_LED : out STD_LOGIC_VECTOR (3 downto 0);
           S_7Seg : out STD_LOGIC_VECTOR (6 downto 0);
           Anode : out STD_LOGIC_VECTOR (3 downto 0));
end NanoProcessor;

architecture Behavioral of NanoProcessor is

component PC is
    Port ( Res : in STD_LOGIC;
           Clk : in STD_LOGIC;
           I : in STD_LOGIC_VECTOR (2 downto 0);
```

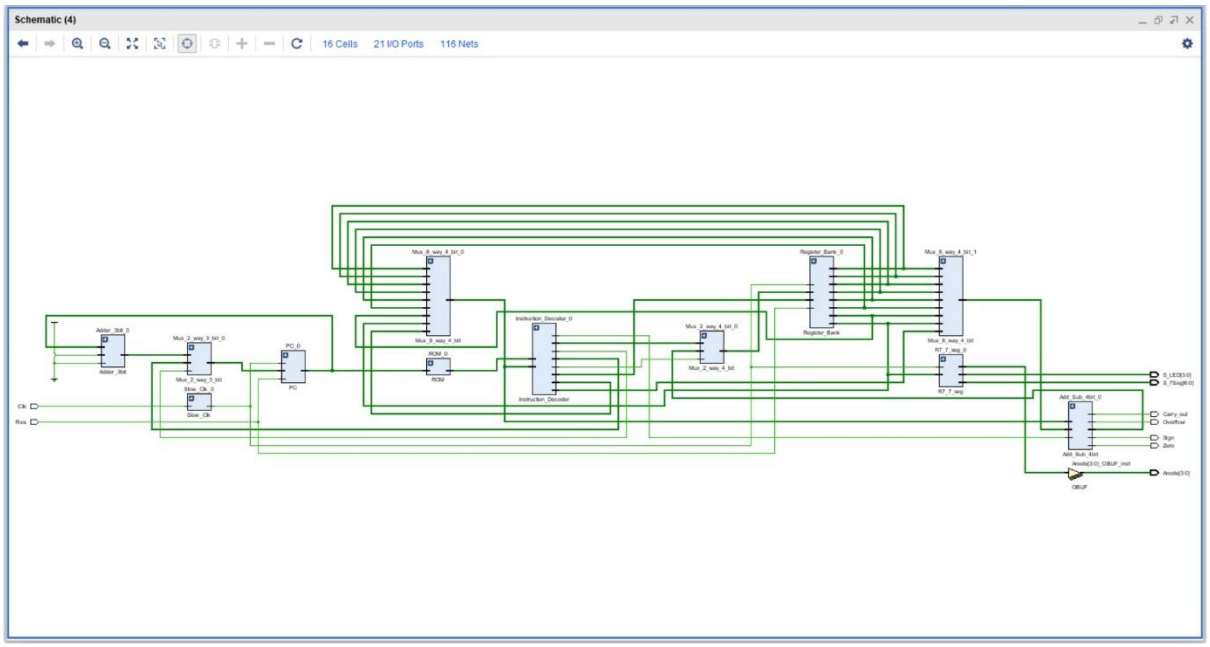
```
            Q : out STD_LOGIC_VECTOR (2 downto 0));
end component;

component Adder_3bit is
    Port ( A : in STD_LOGIC_VECTOR (2 downto 0);
           B : in STD_LOGIC_VECTOR (2 downto 0);
           C_in : in STD_LOGIC;
           C_out : out STD_LOGIC;
           S : out STD_LOGIC_VECTOR (2 downto 0));
end component;

component Mux_2_way_3_bit is
    Port ( S : in STD_LOGIC;
           A : in STD_LOGIC_VECTOR (2 downto 0);
           B : in STD_LOGIC_VECTOR (2 downto 0);
           Y : out STD_LOGIC_VECTOR (2 downto 0));
end component;

component ROM is
    Port ( memory_select : in STD_LOGIC_VECTOR (2 downto 0);
           instruction : out STD_LOGIC_VECTOR (11 downto 0));
end component ROM;

component Instruction_Decoder is
    Port ( I : in STD_LOGIC_VECTOR (11 downto 0);  --Instruction Bus
           R : in STD_LOGIC_VECTOR (3 downto 0);  -- Register check for jump
           Reg_EN : out STD_LOGIC_VECTOR (2 downto 0);  --Register enable
           LS : out STD_LOGIC; --Load select
           IV : out STD_LOGIC_VECTOR (3 downto 0);  -- Immediate value
           Reg_S_1 : out STD_LOGIC_VECTOR (2 downto 0);  --Register select 1
           Reg_S_2 : out STD_LOGIC_VECTOR (2 downto 0);  --Register select 2
           Add_Sub : out STD_LOGIC;  --Abb/Sub select
           JMP : out STD_LOGIC;  -- Jump Flag
           JMP_Address : out STD_LOGIC_VECTOR (2 downto 0));  -- Address to
jump
end component;

component Mux_2_way_4_bit is
    Port ( S : in STD_LOGIC;
           A : in STD_LOGIC_VECTOR (3 downto 0);
           B : in STD_LOGIC_VECTOR (3 downto 0);
           Y : out STD_LOGIC_VECTOR (3 downto 0));
end component;

component Register_Bank is
    Port ( Res : in STD_LOGIC;
           Clk : in STD_LOGIC;
           Input : in STD_LOGIC_VECTOR (2 downto 0);
           Data : in STD_LOGIC_VECTOR (3 downto 0);
```

```
            R0 : out STD_LOGIC_VECTOR (3 downto 0);
            R1 : out STD_LOGIC_VECTOR (3 downto 0);
            R2 : out STD_LOGIC_VECTOR (3 downto 0);
            R3 : out STD_LOGIC_VECTOR (3 downto 0);
            R4 : out STD_LOGIC_VECTOR (3 downto 0);
            R5 : out STD_LOGIC_VECTOR (3 downto 0);
            R6 : out STD_LOGIC_VECTOR (3 downto 0);
            R7 : out STD_LOGIC_VECTOR (3 downto 0));
end component;

component Mux_8_way_4_bit is
    Port ( S : in STD_LOGIC_VECTOR (2 downto 0);
            I_0 : in STD_LOGIC_VECTOR (3 downto 0);
            I_1 : in STD_LOGIC_VECTOR (3 downto 0);
            I_2 : in STD_LOGIC_VECTOR (3 downto 0);
            I_3 : in STD_LOGIC_VECTOR (3 downto 0);
            I_4 : in STD_LOGIC_VECTOR (3 downto 0);
            I_5 : in STD_LOGIC_VECTOR (3 downto 0);
            I_6 : in STD_LOGIC_VECTOR (3 downto 0);
            I_7 : in STD_LOGIC_VECTOR (3 downto 0);
            Y : out STD_LOGIC_VECTOR (3 downto 0));
end component;

component R7_7_seg is
    Port ( Input : in STD_LOGIC_VECTOR (3 downto 0);
            Clk : in STD_LOGIC;
            S_LED : out STD_LOGIC_VECTOR (3 downto 0);
            S_7Seg : out STD_LOGIC_VECTOR (6 downto 0);
            Anode : out STD_LOGIC_VECTOR (3 downto 0));
end component;

component Add_Sub_4bit is
    Port ( A : in STD_LOGIC_VECTOR (3 downto 0);
            B : in STD_LOGIC_VECTOR (3 downto 0);
            M : in STD_LOGIC;
            S : out STD_LOGIC_VECTOR (3 downto 0);
            C_out : out STD_LOGIC;
            Overflow : out STD_LOGIC;
            Sign: out STD_LOGIC;
            Zero : out STD_LOGIC );

end component;

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

```
signal
Mux_2_3_out,PC_out,Adder3_out,JMP_Address,Register_enable,Register_select1,Re
gister_select2: STD_LOGIC_VECTOR (2 downto 0);
signal Jump_flag,Load_select, Add_sub_select,slow_clk_out: STD_LOGIC;
signal ROM_out : STD_LOGIC_VECTOR (11 downto 0);
signal Mux1_8_4_out,Mux2_8_4_out,Mux_2_4_out,Immediate_value,Add_sub_out:
STD_LOGIC_VECTOR (3 downto 0);
signal R_0,R_1,R_2,R_3,R_4,R_5,R_6,R_7 : STD_LOGIC_VECTOR (3 downto 0);

begin

PC_0: PC port map(
    Res=>Res,
    Clk=>slow_clk_out,
    I=>Mux_2_3_out,
    Q=>PC_out
);

Adder_3bit_0: Adder_3bit port map(
    A=>PC_out,
    B=>"001",
    C_in=>'0',
    S=>Adder3_out
);

Mux_2_way_3_bit_0: Mux_2_way_3_bit port map(
    S=>Jump_flag,
    A=>Adder3_out,
    B=>JMP_Address,
    Y=>Mux_2_3_out
);

ROM_0: ROM port map(
    memory_select=>PC_out,
    instruction=>ROM_out
);

Instruction_Decoder_0: Instruction_Decoder port map(
    I=>ROM_out,
    R=>Mux1_8_4_out,
    Reg_EN=>Register_enable,
    LS=>Load_select,
    IV=>Immediate_value,
    Reg_S_1=>Register_select1,
    Reg_S_2=>Register_select2,
    Add_Sub=>Add_sub_select,
    JMP=>Jump_flag,
    JMP_Address=>JMP_Address
);
```

```
Mux_2_way_4_bit_0: Mux_2_way_4_bit port map(
    S=>Load_select,
    A=>Immediate_value,
    B=>Add_sub_out,
    Y=>Mux_2_4_out
);

Register_Bank_0: Register_Bank port map(
    Res=>Res,
    Clk=>slow_clk_out,
    Input=>Register_enable,
    Data=>Mux_2_4_out,
    R0=>R_0,
    R1=>R_1,
    R2=>R_2,
    R3=>R_3,
    R4=>R_4,
    R5=>R_5,
    R6=>R_6,
    R7=>R_7
);

R7_7_seg_0:  R7_7_seg port map(
    Input =>R_7,
    Clk =>slow_clk_out,
    S_LED=>S_LED,
    S_7Seg=>S_7Seg,
    Anode =>Anode
);

Mux_8_way_4_bit_0: Mux_8_way_4_bit port map(
    S=>Register_select1,
    I_0=>R_0,
    I_1=>R_1,
    I_2=>R_2,
    I_3=>R_3,
    I_4=>R_4,
    I_5=>R_5,
    I_6=>R_6,
    I_7=>R_7,
    Y=>Mux1_8_4_out
);

Mux_8_way_4_bit_1:Mux_8_way_4_bit port map(
    S=>Register_select2,
    I_0=>R_0,
    I_1=>R_1,
    I_2=>R_2,
```

```
    I_3=>R_3,
    I_4=>R_4,
    I_5=>R_5,
    I_6=>R_6,
    I_7=>R_7,
    Y=>Mux2_8_4_out
);

Add_Sub_4bit_0: Add_Sub_4bit port map(
    A => Mux1_8_4_out,
    B => Mux2_8_4_out,
    M => Add_sub_select,
    S => Add_sub_out,
    C_out=>Carry_out,
    Overflow =>Overflow,
    Sign=>Sign,
    Zero =>Zero
);

Slow_Clk_0: Slow_Clk port map(
    Clk_in =>Clk,
    Clk_out =>slow_clk_out
);

end Behavioral;
```



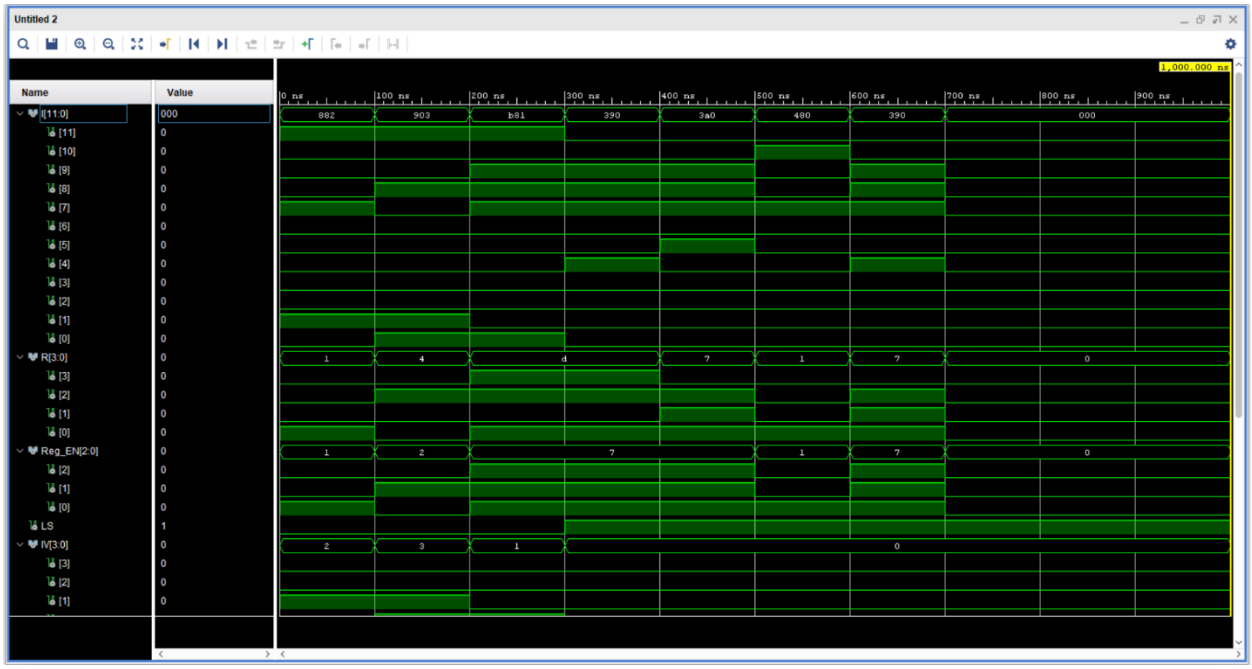
All timing diagrams

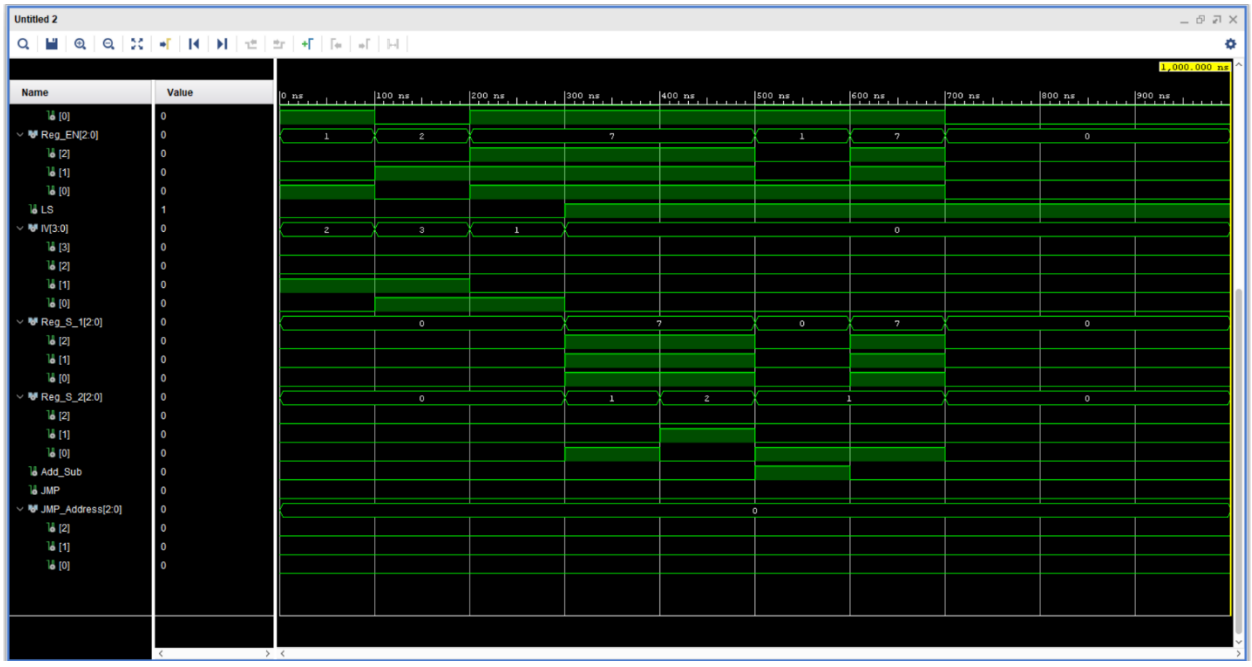
Instruction Decoder:

```
----------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date: 17.04.2024 12:03:55
-- 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 is


    Port ( I : in STD_LOGIC_VECTOR (11 downto 0);
           R : in STD_LOGIC_VECTOR (3 downto 0);  -- Register check for jump
```

```
            Reg_EN : out STD_LOGIC_VECTOR (2 downto 0);
            LS : out STD_LOGIC;
            IV : out STD_LOGIC_VECTOR (3 downto 0);
            Reg_S_1 : out STD_LOGIC_VECTOR (2 downto 0);
            Reg_S_2 : out STD_LOGIC_VECTOR (2 downto 0);
            Add_Sub : out STD_LOGIC;
            JMP : out STD_LOGIC;
            JMP_Address : out STD_LOGIC_VECTOR (2 downto 0));

end component;

signal I :  STD_LOGIC_VECTOR (11 downto 0);
signal  R : STD_LOGIC_VECTOR (3 downto 0);
signal  Reg_EN : STD_LOGIC_VECTOR (2 downto 0);
signal LS : STD_LOGIC;
signal IV : STD_LOGIC_VECTOR (3 downto 0);
signal Reg_S_1 : STD_LOGIC_VECTOR (2 downto 0);
signal Reg_S_2 : STD_LOGIC_VECTOR (2 downto 0);
signal Add_Sub : STD_LOGIC;
signal JMP : STD_LOGIC;
signal  JMP_Address : STD_LOGIC_VECTOR (2 downto 0);

begin

UUT: Instruction_Decoder port map(
    I=>I,
    R=>R,
    Reg_EN=>Reg_EN,
    LS=>LS,
    IV=>IV,
    Reg_S_1=>Reg_S_1,
    Reg_S_2=>Reg_S_2,
    Add_Sub=>Add_Sub,
    JMP=>JMP,
    JMP_Address=>JMP_Address
);

process


begin

     I<="100010000010";
     R<="0001";

     wait for 100ns;

     I<="100100000011";
     R<="0100";
```

```
      wait for 100ns;

      I<="101110000001";
      R<="1101";

      wait for 100ns;

      I<="001110010000";
      R<="1101";

      wait for 100ns;

      I<="001110100000";
      R<="0111";

      wait for 100ns;

      I<="0100100000000";
      R<="0001";

      wait for 100ns;

     I<="001110010000";
     R<="0111";

     wait for 100ns;

     I<="000000000000";
     R<="0000";

     wait for 100ns;

      wait;

end process;

end Behavioral;
```


Untitled 2
Name
Value
I[11:0]
000
R[3:0]
Reg_EN[2:0]
LS
IV[3:0]
0 ns
100 ns
200 ns
300 ns
400 ns
500 ns
600 ns
700 ns
800 ns
900 ns
1,000.000 ns
882
903
b81
390
3a0
480
390
000

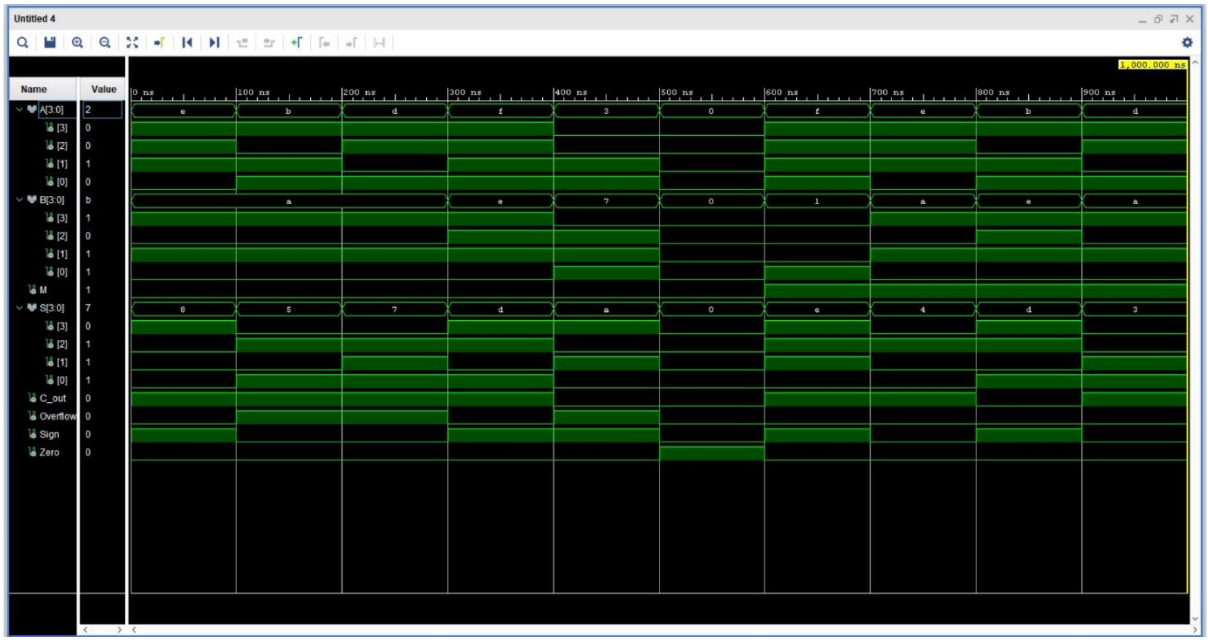

Untitled 2
Name
Value
Reg_EN[2:0]
LS
IV[3:0]
Reg_S_1[2:0]
Reg_S_2[2:0]
Add_Sub
JMP
JMP_Address[2:0]
0 ns
100 ns
200 ns
300 ns
400 ns
500 ns
600 ns
700 ns
800 ns
900 ns
1,000.000 ns

4-bit Add/Subtract unit:

```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
-- Create Date: 15.04.2024 23:21:38
-- Design Name:
-- Module Name: TB_Add_Sub_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_Add_Sub_4bit is
--  Port ( );
end TB_Add_Sub_4bit;

architecture Behavioral of TB_Add_Sub_4bit is

component Add_Sub_4bit is
    Port ( A : in STD_LOGIC_VECTOR (3 downto 0);
           B : in STD_LOGIC_VECTOR (3 downto 0);
           M : in STD_LOGIC;
           S : inout STD_LOGIC_VECTOR (3 downto 0);
           C_out : inout STD_LOGIC;
           Overflow : out STD_LOGIC;
           Sign: out STD_LOGIC;
           Zero : out STD_LOGIC );
end component;

signal A : STD_LOGIC_VECTOR (3 downto 0);
```

```
signal  B : STD_LOGIC_VECTOR (3 downto 0);
signal M : STD_LOGIC;
signal  S : STD_LOGIC_VECTOR (3 downto 0);
signal C_out :  STD_LOGIC;
signal Overflow : STD_LOGIC;
signal Sign: STD_LOGIC;
signal  Zero : STD_LOGIC ;

begin
UUT: Add_Sub_4bit port map(
A=>A,
B=>B,
M=>M,
S=>S,
C_out=>C_out,
Overflow=>Overflow,
Sign=>Sign,
Zero=>Zero
);

process
begin
        --ADD--

        A<="1110";
        B<="1010";
        M<='0';
        WAIT FOR 100 ns;

        A<="1011";
        B<="1010";
        WAIT FOR 100 ns;

        A<="1101";
        B<="1010";
        WAIT FOR 100 ns;

        A<="1111";
        B<="1110";
        WAIT FOR 100 ns;

        A<="0011";
        B<="0111";
        WAIT FOR 100 ns;

        A<="0000";
        B<="0000";
        WAIT FOR 100 ns;
```

```
        --SUBSTRACT--

        A<="1111";
        B<="0001";
        M<='1';
        WAIT FOR 100 ns;

        A<="1110";
        B<="1010";
        WAIT FOR 100 ns;

        A<="1011";
        B<="1110";
        WAIT FOR 100 ns;

        A<="1101";
        B<="1010";
        WAIT FOR 100 ns;

        A<="0010";
        B<="1011";
        WAIT FOR 100 ns;

        A<="1111";
        B<="1111";
        WAIT FOR 100 ns;

        WAIT;
end process;
end Behavioral;
```

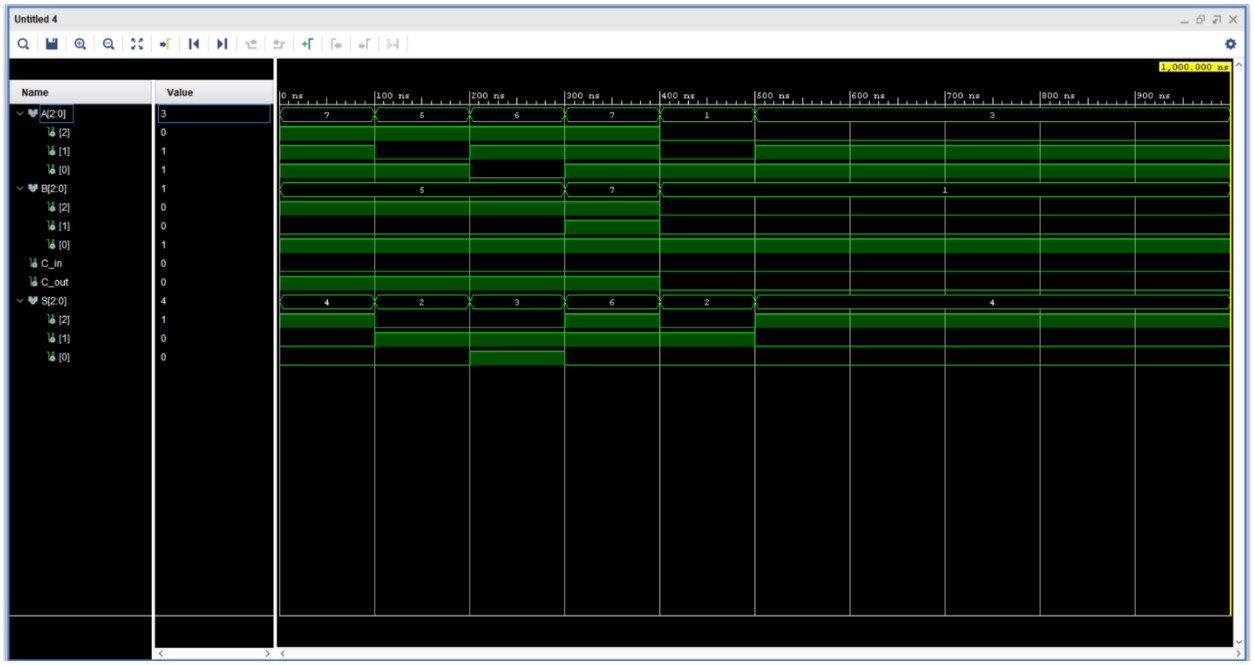


3-bit adder:

```
----------------------------------------------------------------------------------
-----
-- Company:
-- Engineer:
--
-- Create Date: 15.04.2024 22:16:42
-- Design Name:
-- Module Name: TB_Adder_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_Adder_3bit is
--  Port ( );
end TB_Adder_3bit;

architecture Behavioral of TB_Adder_3bit is

component Adder_3bit is
    Port ( A : in STD_LOGIC_VECTOR (2 downto 0);
           B : in STD_LOGIC_VECTOR (2 downto 0);
           C_in : in STD_LOGIC;
           C_out : out STD_LOGIC;
           S : out STD_LOGIC_VECTOR (2 downto 0));
end component;
```

```
signal A : STD_LOGIC_VECTOR (2 downto 0);
signal B :  STD_LOGIC_VECTOR (2 downto 0);
signal C_in : STD_LOGIC;
signal C_out : STD_LOGIC;
signal S : STD_LOGIC_VECTOR (2 downto 0);

begin

UUT: Adder_3bit port map(
    A=>A,
    B=>B,
    C_in=>C_in,
    C_out=>C_out,
    S=>S

);

process
begin

A<= "111";
B<= "101";
C_in<='0';

wait for 100ns;

A<= "101";
B<="101";

wait for 100ns;

A<="110";
B<="101";

wait for 100ns;

A<="111";
B<="111";

wait for 100ns;

A<="001";
B<="001";

wait for 100ns;

A<="011";
B<="001";
```

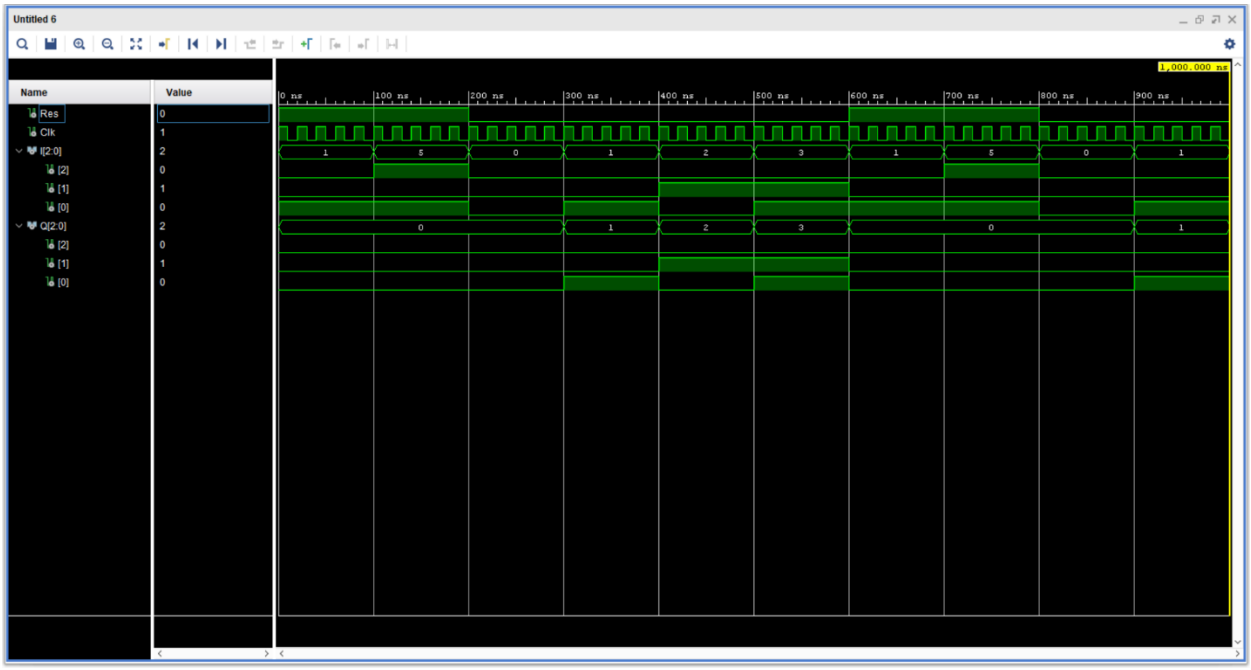
```
wait for 100ns;

wait;
end process;

end Behavioral;
```



3-bit Program Counter (PC):

```
----------------------------------------------------------------------------------
-----
-- Company:
-- Engineer:
--
-- Create Date: 15.04.2024 13:06:22
-- Design Name:
-- Module Name: TB_PC - 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_PC is
--  Port ( );
end TB_PC;

architecture Behavioral of TB_PC is

component PC is
    Port ( Res : in STD_LOGIC;
           Clk : in STD_LOGIC;
           I : in STD_LOGIC_VECTOR (2 downto 0);
           Q : out STD_LOGIC_VECTOR (2 downto 0));
end component;
```

```
signal Res : STD_LOGIC;
signal Clk : STD_LOGIC := '0';
signal I : STD_LOGIC_VECTOR (2 downto 0);
signal Q : STD_LOGIC_VECTOR (2 downto 0);

begin

UUT: PC port map(
    Res => Res,
    Clk => Clk,
    I => I,
    Q => Q
);

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

end process;

process
begin

    Res <= '1';
    I <= "001";
    wait for 100 ns;

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

    Res <= '0';
    I <= "000";
    wait for 100 ns;

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

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

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


end process;

end Behavioral;
```

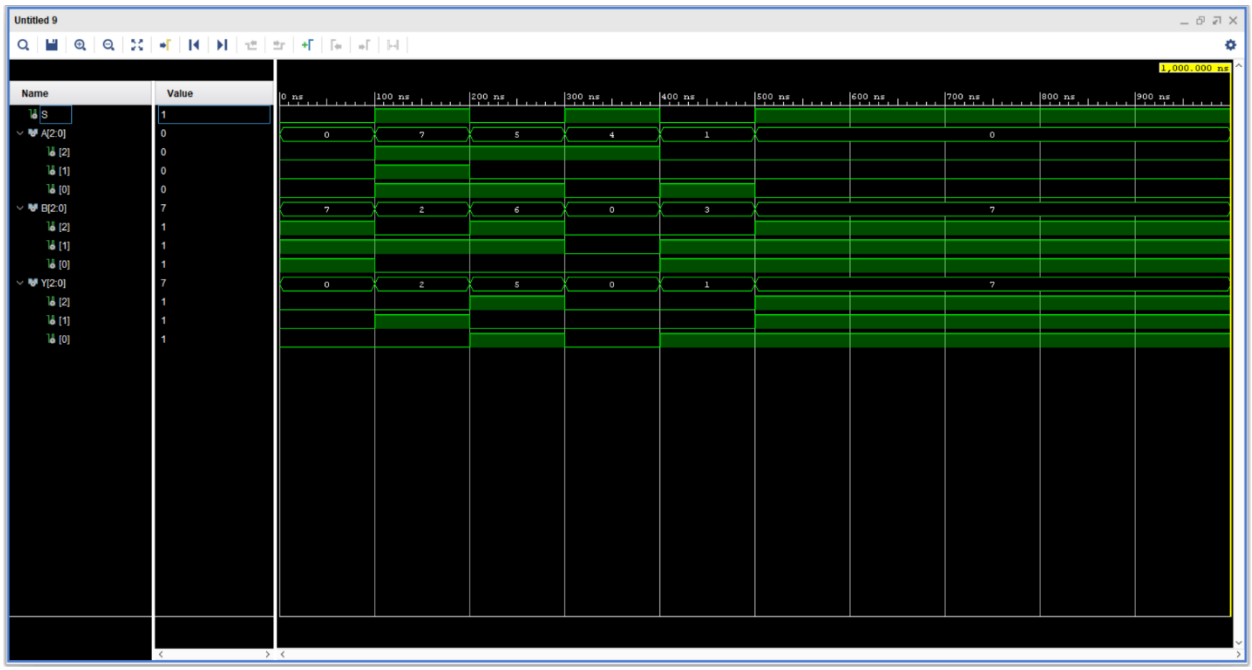

Untitled 6
Name
Value
Res
Clk
I[2:0]
[2]
[1]
[0]
Q[2:0]
[2]
[1]
[0]
0
1
2
0
1
0
2
0
1
0
0 ns
100 ns
200 ns
300 ns
400 ns
500 ns
600 ns
700 ns
800 ns
900 ns
1,000.000 ns
1 5 0 1 2 3 1 5 0 1
0 1 2 3 0 1

2-way 3-bit multiplexer:

```
----------------------------------------------------------------------------------
-----
-- Company:
-- Engineer:
--
-- Create Date: 13.04.2024 01:47:45
-- Design Name:
-- Module Name: TB_Mux_2_way_3_bit - 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_way_3_bit is
--  Port ( );
end TB_Mux_2_way_3_bit;

architecture Behavioral of TB_Mux_2_way_3_bit is

component Mux_2_way_3_bit is
    Port ( S : in STD_LOGIC;
           A : in STD_LOGIC_VECTOR (2 downto 0);
           B : in STD_LOGIC_VECTOR (2 downto 0);
           Y : out STD_LOGIC_VECTOR (2 downto 0));
end component;
```

```
signal S :  STD_LOGIC;
signal A :  STD_LOGIC_VECTOR (2 downto 0);
signal B : STD_LOGIC_VECTOR (2 downto 0);
signal  Y : STD_LOGIC_VECTOR (2 downto 0);

begin


UUT:Mux_2_way_3_bit PORT MAP(
S => S,
A => A,
B => B,
Y => Y

);

process
begin

S<='0';
A<= "000";
B<="111";

wait for 100 ns;

S<='1';
A<="111";
B<="010";

wait for 100 ns;

S<='0';
A<= "101";
B<="110";

wait for 100 ns;

S<='1';
A<="100";
B<="000";

wait for 100 ns;

S<='0';
A<="001";
B<="011";

wait for 100 ns;
```

```
S<='1';
A<="000";
B<="111";

wait for 100 ns;

wait;
end process;

end Behavioral;
```

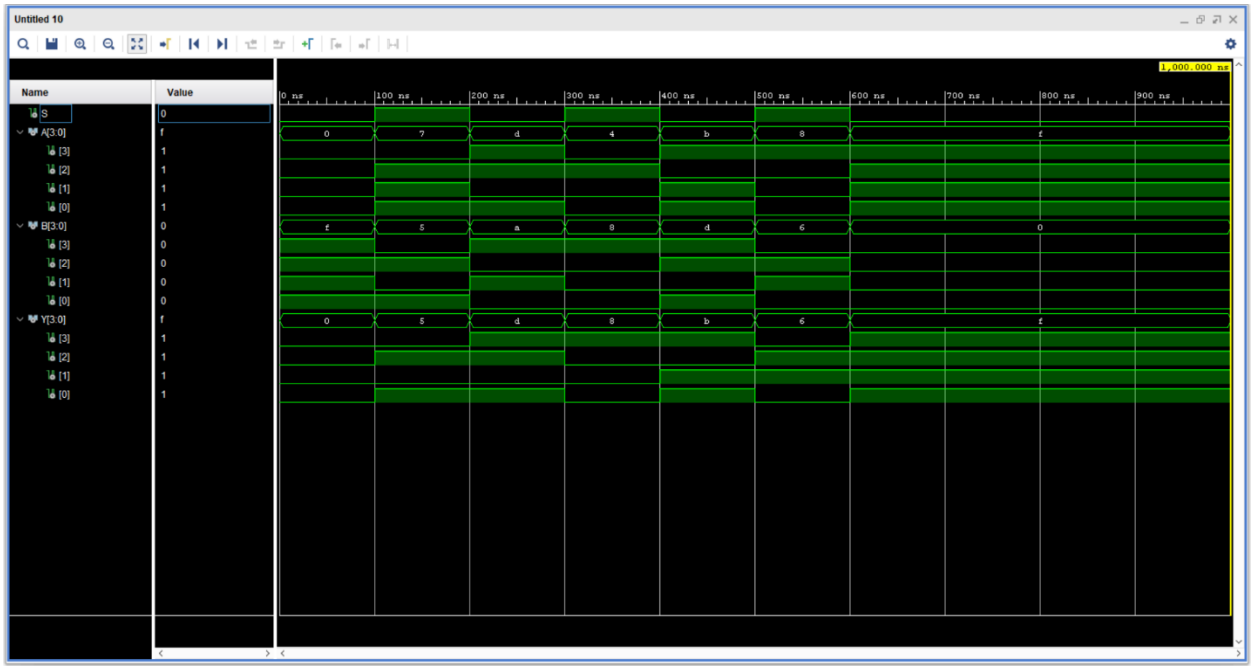


2-way 4-bit multiplexer:

```
----------------------------------------------------------------------------------
-----
-- Company:
-- Engineer:
--
-- Create Date: 13.04.2024 13:08:28
-- Design Name:
-- Module Name: TB_Mux_2_way_4_bit - 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_way_4_bit is
--  Port ( );
end TB_Mux_2_way_4_bit;

architecture Behavioral of TB_Mux_2_way_4_bit is

component Mux_2_way_4_bit is
    Port ( S : in STD_LOGIC;
           A : in STD_LOGIC_VECTOR (3 downto 0);
           B : in STD_LOGIC_VECTOR (3 downto 0);
           Y : out STD_LOGIC_VECTOR (3 downto 0));
end component;
```

```
signal S :  STD_LOGIC;
signal A :  STD_LOGIC_VECTOR (3 downto 0);
signal B : STD_LOGIC_VECTOR (3 downto 0);
signal  Y : STD_LOGIC_VECTOR (3 downto 0);

begin

UUT:Mux_2_way_4_bit PORT MAP(
S => S,
A => A,
B => B,
Y => Y

);

process
begin

S<='0';
A<= "0000";
B<="1111";

wait for 100 ns;

S<='1';
A<="0111";
B<="0101";

wait for 100 ns;

S<='0';
A<= "1101";
B<="1010";

wait for 100 ns;

S<='1';
A<="0100";
B<="1000";

wait for 100 ns;

S<='0';
A<="1011";
B<="1101";

wait for 100 ns;
```

```
S<='1';
A<="1000";
B<="0110";

wait for 100 ns;

S<='0';
A<="1111";
B<="0000";

wait for 100 ns;

wait;
end process;

end Behavioral;
```



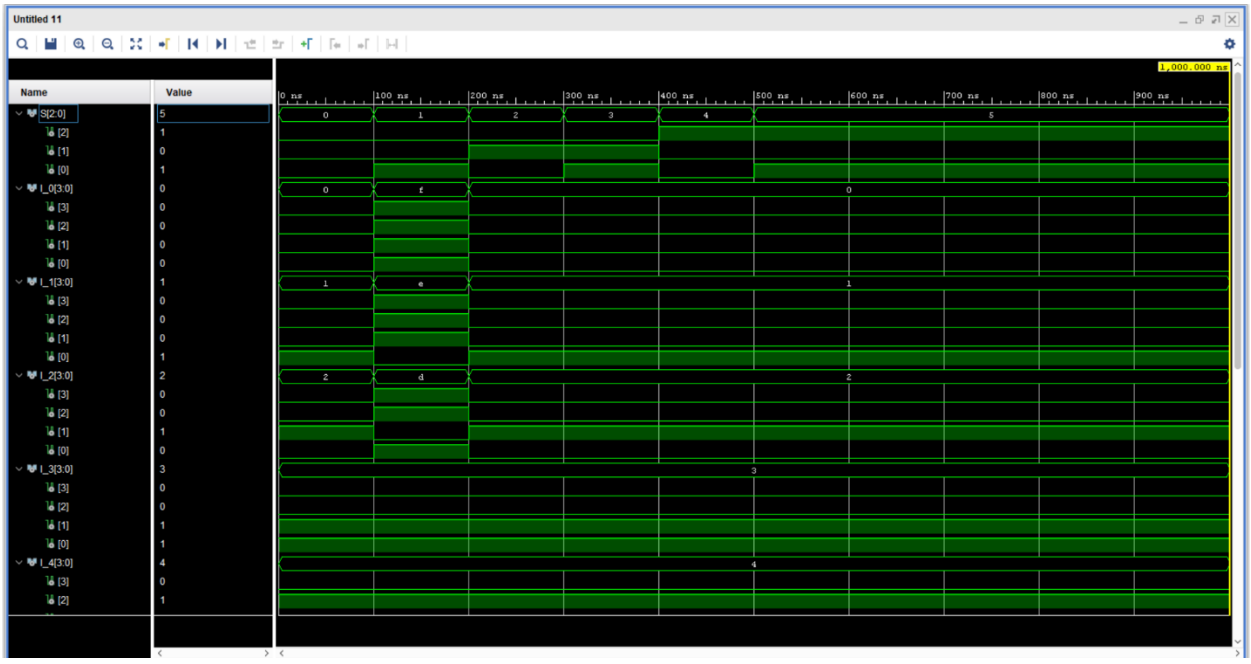
8-way 4-bit multiplexer:

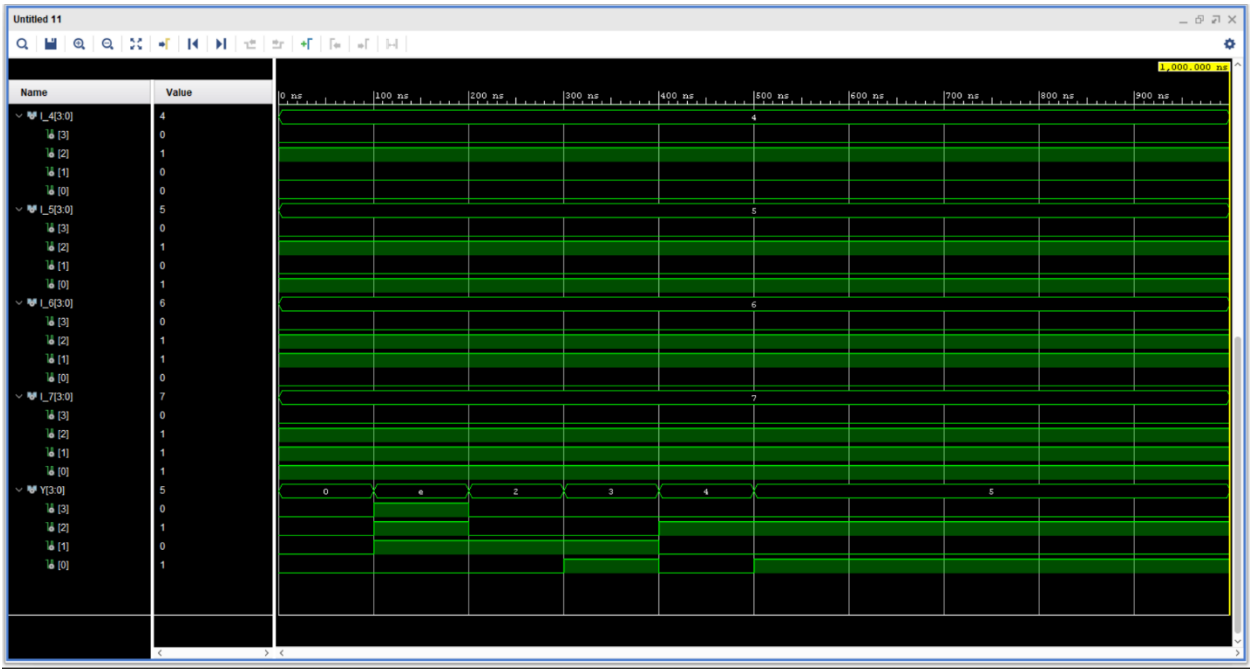
```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date: 14.04.2024 23:49:04
-- Design Name:
-- Module Name: TB_Mux_8_way_4_bit - Behavioral
-- Project Name:
-- Target Devices:
-- Tool Versions:
-- Description:
--
-- Dependencies:
--
-- 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_way_4_bit is
--  Port ( );
end TB_Mux_8_way_4_bit;

architecture Behavioral of TB_Mux_8_way_4_bit is

component Mux_8_way_4_bit is
    Port ( S : in STD_LOGIC_VECTOR (2 downto 0);
           I_0 : in STD_LOGIC_VECTOR (3 downto 0);
           I_1 : in STD_LOGIC_VECTOR (3 downto 0);
           I_2 : in STD_LOGIC_VECTOR (3 downto 0);
           I_3 : in STD_LOGIC_VECTOR (3 downto 0);
           I_4 : in STD_LOGIC_VECTOR (3 downto 0);
           I_5 : in STD_LOGIC_VECTOR (3 downto 0);
```

```
            I_6 : in STD_LOGIC_VECTOR (3 downto 0);
            I_7 : in STD_LOGIC_VECTOR (3 downto 0);
            Y : out STD_LOGIC_VECTOR (3 downto 0));
end component;

signal S : STD_LOGIC_VECTOR (2 downto 0);
signal I_0 :  STD_LOGIC_VECTOR (3 downto 0);
signal I_1 :  STD_LOGIC_VECTOR (3 downto 0);
signal I_2 :  STD_LOGIC_VECTOR (3 downto 0);
signal I_3 : STD_LOGIC_VECTOR (3 downto 0);
signal I_4 :  STD_LOGIC_VECTOR (3 downto 0);
signal I_5 :  STD_LOGIC_VECTOR (3 downto 0);
signal I_6 :  STD_LOGIC_VECTOR (3 downto 0);
signal I_7 :  STD_LOGIC_VECTOR (3 downto 0);
signal Y : STD_LOGIC_VECTOR (3 downto 0);

begin

UUT:Mux_8_way_4_bit PORT MAP(
S => S,
I_0 => I_0,
I_1 => I_1,
I_2 => I_2,
I_3 => I_3,
I_4 => I_4,
I_5 => I_5,
I_6 => I_6,
I_7 => I_7,
Y => Y );

process
begin

S <="000";
I_0<= "0000";
I_1<= "0001";
I_2<= "0010";
I_3<= "0011";
I_4<= "0100";
I_5<= "0101";
I_6<= "0110";
I_7<= "0111";

wait for 100 ns;
S <="001";
I_0<= "1111";
I_1<= "1110";
I_2<= "1101";
I_3<= "0011";
```

```
I_4<= "0100";
I_5<= "0101";
I_6<= "0110";
I_7<= "0111";

wait for 100 ns;

S <="010";
I_0<= "0000";
I_1<= "0001";
I_2<= "0010";
I_3<= "0011";
I_4<= "0100";
I_5<= "0101";
I_6<= "0110";
I_7<= "0111";

wait for 100 ns;

S <="011";
I_0<= "0000";
I_1<= "0001";
I_2<= "0010";
I_3<= "0011";
I_4<= "0100";
I_5<= "0101";
I_6<= "0110";
I_7<= "0111";

wait for 100 ns;

S <="100";
I_0<= "0000";
I_1<= "0001";
I_2<= "0010";
I_3<= "0011";
I_4<= "0100";
I_5<= "0101";
I_6<= "0110";
I_7<= "0111";

wait for 100 ns;

S <="101";
I_0<= "0000";
I_1<= "0001";
I_2<= "0010";
I_3<= "0011";
I_4<= "0100";
```

```
I_5<= "0101";
I_6<= "0110";
I_7<= "0111";

wait for 100 ns;

wait;
end process;

end Behavioral;
```



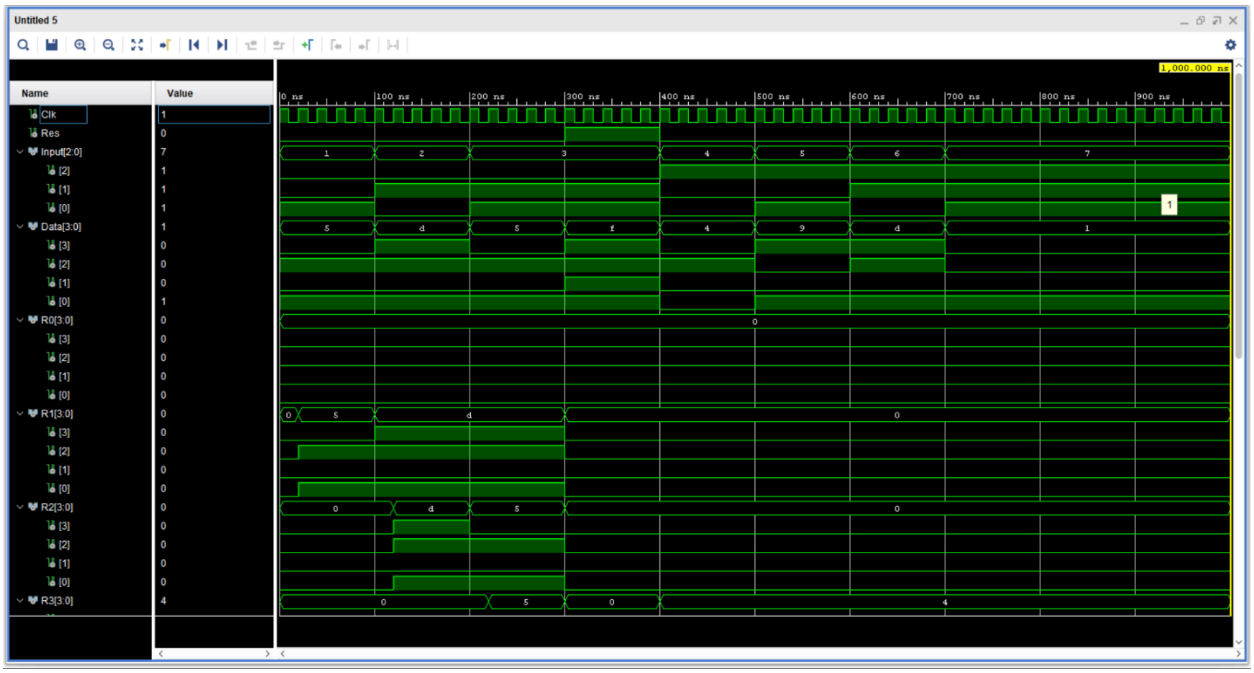


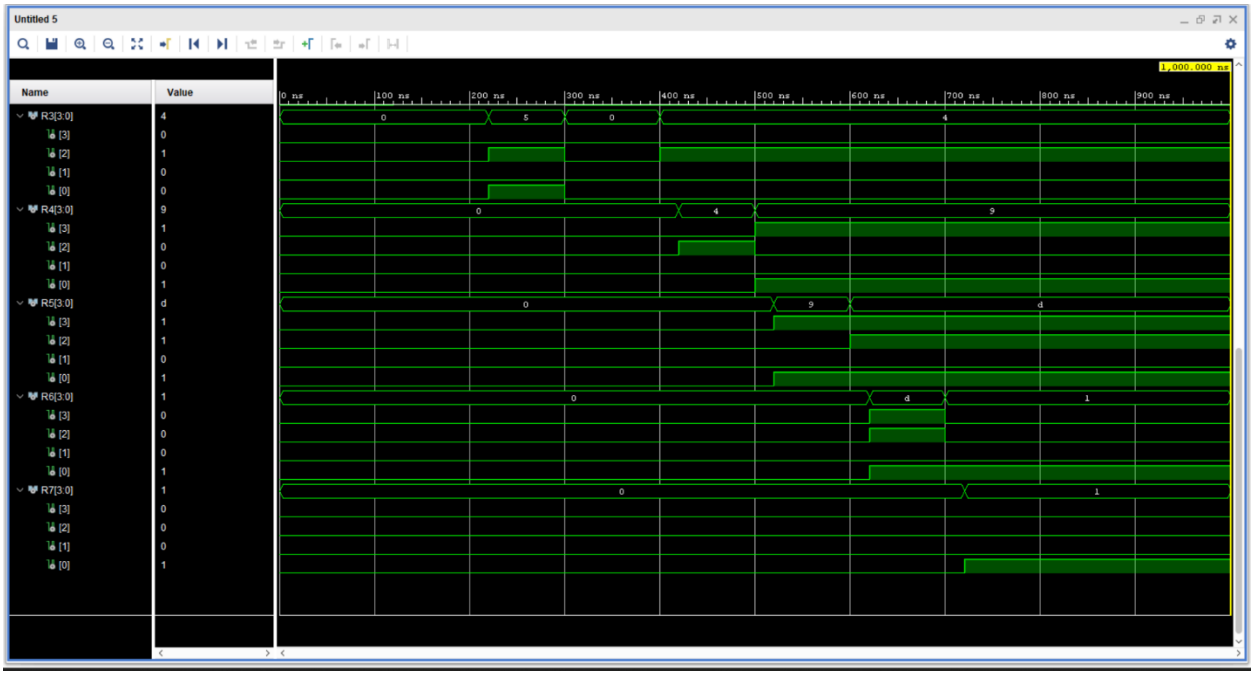
Register Bank:

```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
--
-- Create Date: 16.04.2024 11:18:14
-- Design Name:
-- Module Name: TB_Register_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_Register_Bank is
--  Port ( );
end TB_Register_Bank;

architecture Behavioral of TB_Register_Bank is

component Register_Bank is
    Port ( Res : in STD_LOGIC;
           Clk : in STD_LOGIC;
           Input : in STD_LOGIC_VECTOR (2 downto 0);
           Data : in STD_LOGIC_VECTOR (3 downto 0);
           R0 : out STD_LOGIC_VECTOR (3 downto 0);
           R1 : out STD_LOGIC_VECTOR (3 downto 0);
           R2 : out STD_LOGIC_VECTOR (3 downto 0);
```

```
            R3 : out STD_LOGIC_VECTOR (3 downto 0);
            R4 : out STD_LOGIC_VECTOR (3 downto 0);
            R5 : out STD_LOGIC_VECTOR (3 downto 0);
            R6 : out STD_LOGIC_VECTOR (3 downto 0);
            R7 : out STD_LOGIC_VECTOR (3 downto 0));
end component;

signal Clk :  STD_LOGIC :='0';
signal Res :  STD_LOGIC ;
signal Input : STD_LOGIC_VECTOR (2 downto 0);
signal Data : STD_LOGIC_VECTOR (3 downto 0);
signal R0,R1,R2,R3,R4,R5,R6,R7 : STD_LOGIC_VECTOR (3 downto 0);

begin

UUT: Register_Bank port map(
Res=>Res,
Clk=>Clk,
Input=>Input,
Data=>Data,
R0=>R0,
R1=>R1,
R2=>R2,
R3=>R3,
R4=>R4,
R5=>R5,
R6=>R6,
R7=>R7

);

process
begin


    Clk <= (not Clk);
    wait for 10 ns;

end process;

process
begin

    Res<='0';
    Input<="001";
    Data<="0101";

    wait for 100ns;

    Input<="010";
```

```
    Data<="1101";

    wait for 100ns;

    Input<="011";
    Data<="0101";

    wait for 100ns;

    Res<='1';
    Input<="011";
    Data<="1111";

    wait for 100ns;

    Res<='0';
    Input<="100";
    Data<="0100";

    wait for 100ns;

    Input<="101";
    Data<="1001";

    wait for 100ns;

    Input<="110";
    Data<="1101";

    wait for 100ns;

    Input<="111";
    Data<="0001";

    wait for 100ns;

    wait;

end process;

end Behavioral;
```

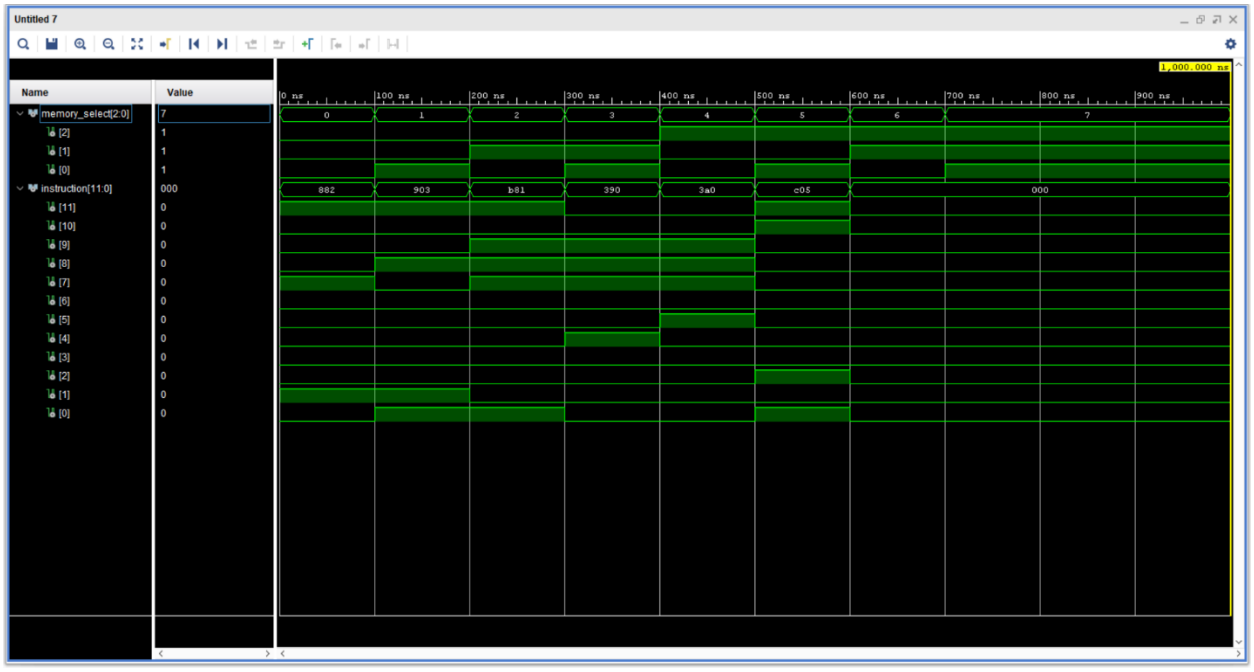

Untitled 5
Name
Value
Clk
Res
Input[2:0]
Data[3:0]
R0[3:0]
R1[3:0]
R2[3:0]
R3[3:0]
1,000.000 ns


Untitled 5
Name
Value
R3[3:0]
R4[3:0]
R5[3:0]
R6[3:0]
R7[3:0]
1,000.000 ns

Program ROM:

```
----------------------------------------------------------------------------------
-- Company:
-- Engineer:
-- Create Date: 16.04.2024 22:20:51
-- Design Name:
-- Module Name: TB_ROM - 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_ROM is
--  Port ( );
end TB_ROM;

architecture Behavioral of TB_ROM is

component ROM is
    Port ( memory_select : in STD_LOGIC_VECTOR (2 downto 0);
           instruction : out STD_LOGIC_VECTOR (11 downto 0));
end component;

signal memory_select :  STD_LOGIC_VECTOR (2 downto 0);
signal instruction : STD_LOGIC_VECTOR (11 downto 0);

begin
```

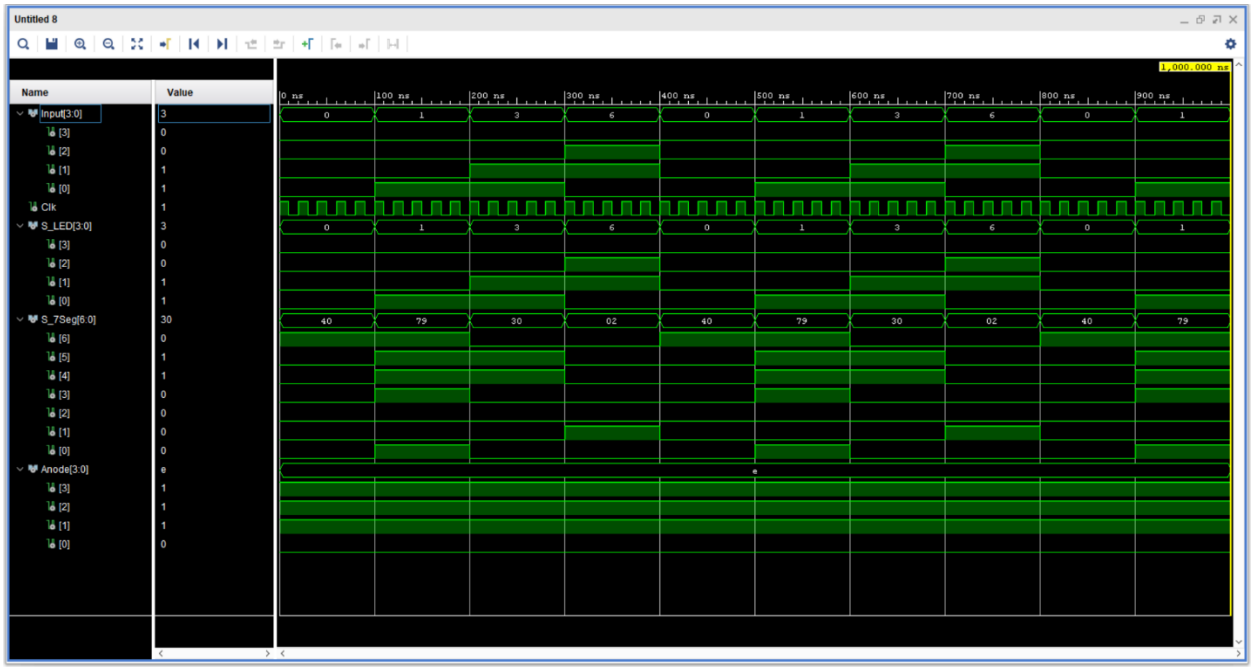
```
UUT:ROM port map(
     memory_select=>memory_select,
     instruction=>instruction);

process
begin
   memory_select <= "000";
   wait for 100ns;
   memory_select <= "001";
   wait for 100ns;
   memory_select <= "010";
   wait for 100ns;
   memory_select <= "011";
   wait for 100ns;
   memory_select <= "100";
   wait for 100ns;
   memory_select <= "101";
   wait for 100ns;
   memory_select <= "110";
   wait for 100ns;
   memory_select <= "111";
   wait for 100ns;
   wait;

end process;


end Behavioral;
```



7 segment display:

```
----------------------------------------------------------------------------------
-----
-- Company:
-- Engineer:
--
-- Create Date: 17.04.2024 08:37:24
-- Design Name:
-- Module Name: TB_R7_7_seg - 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_R7_7_seg is
--  Port ( );
end TB_R7_7_seg;

architecture Behavioral of TB_R7_7_seg is

component R7_7_seg is
    Port ( Input : in STD_LOGIC_VECTOR (3 downto 0);
           Clk : in STD_LOGIC;
           S_LED : out STD_LOGIC_VECTOR (3 downto 0);
           S_7Seg : out STD_LOGIC_VECTOR (6 downto 0);
           Anode : out STD_LOGIC_VECTOR (3 downto 0));
end component;
```

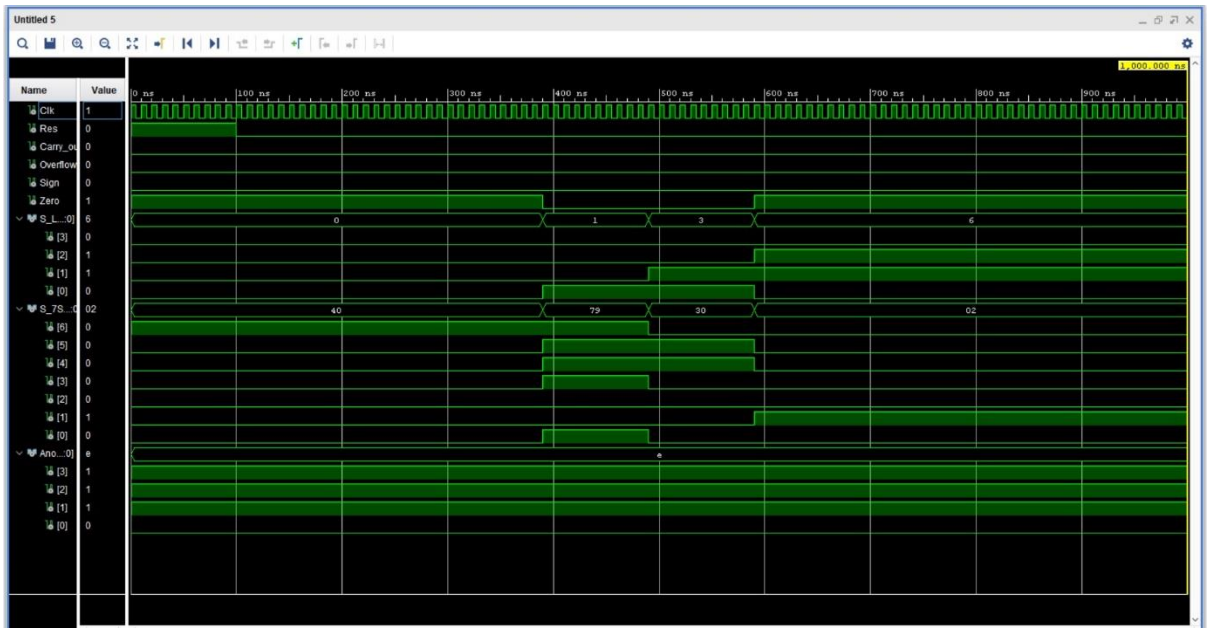
```
signal Input : STD_LOGIC_VECTOR (3 downto 0);
signal Clk : STD_LOGIC:='0';
signal S_LED : STD_LOGIC_VECTOR (3 downto 0);
signal S_7Seg : STD_LOGIC_VECTOR (6 downto 0);
signal Anode : STD_LOGIC_VECTOR (3 downto 0);

begin


UUT: R7_7_seg port map(
    Input=>Input,
    Clk=>Clk,
    S_LED=>S_LED,
    S_7Seg=>S_7Seg,
    Anode=>Anode
);

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

process
begin

     Input<="0000";
     wait for 100ns;

     Input<="0001";
     wait for 100ns;

     Input<="0011";
     wait for 100ns;

     Input<="0110";
     wait for 100ns;

end process;

end Behavioral;
```


Untitled 8
Name
Value
input[3:0]
[3]
[2]
[1]
[0]
Clk
S_LED[3:0]
[3]
[2]
[1]
[0]
S_7Seg[6:0]
[6]
[5]
[4]
[3]
[2]
[1]
[0]
Anode[3:0]
[3]
[2]
[1]
[0]
0 ns
100 ns
200 ns
300 ns
400 ns
500 ns
600 ns
700 ns
800 ns
900 ns
1,000.000 ns

Top-level design (Nano processor):

```
----------------------------------------------------------------------------------
-----
-- Company:
-- Engineer:
--
-- Create Date: 17.04.2024 19:28:06
-- 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 NanoProcessor is
    Port ( Clk : in STD_LOGIC;
           Res : in STD_LOGIC;
           Carry_out:out STD_LOGIC;
           Overflow : out STD_LOGIC;
```

```
            Sign: out STD_LOGIC;
            Zero : out STD_LOGIC;
            S_LED : out STD_LOGIC_VECTOR (3 downto 0);
            S_7Seg : out STD_LOGIC_VECTOR (6 downto 0);
            Anode : out STD_LOGIC_VECTOR (3 downto 0));
end component;

signal Clk : STD_LOGIC:='0';
signal Res :  STD_LOGIC;
signal Carry_out :  STD_LOGIC;
signal  Overflow : STD_LOGIC;
signal Sign: STD_LOGIC;
signal  Zero : STD_LOGIC;
signal S_LED : STD_LOGIC_VECTOR (3 downto 0);
signal S_7Seg : STD_LOGIC_VECTOR (6 downto 0);
signal Anode : STD_LOGIC_VECTOR (3 downto 0);

begin

UUT: NanoProcessor port map(
Clk=>Clk,
Res=>Res,
Carry_out=>Carry_out,
Overflow=>Overflow,
Sign=>Sign,
Zero=>Zero,
S_LED=>S_LED,
S_7Seg=>S_7Seg,
Anode=>Anode);

process
begin
    Clk <= (not Clk);
    wait for 5 ns;

end process;

process
begin
    Res<='1';
    wait for 100ns;
    Res<='0';
    wait;

end process;
end Behavioral;
```

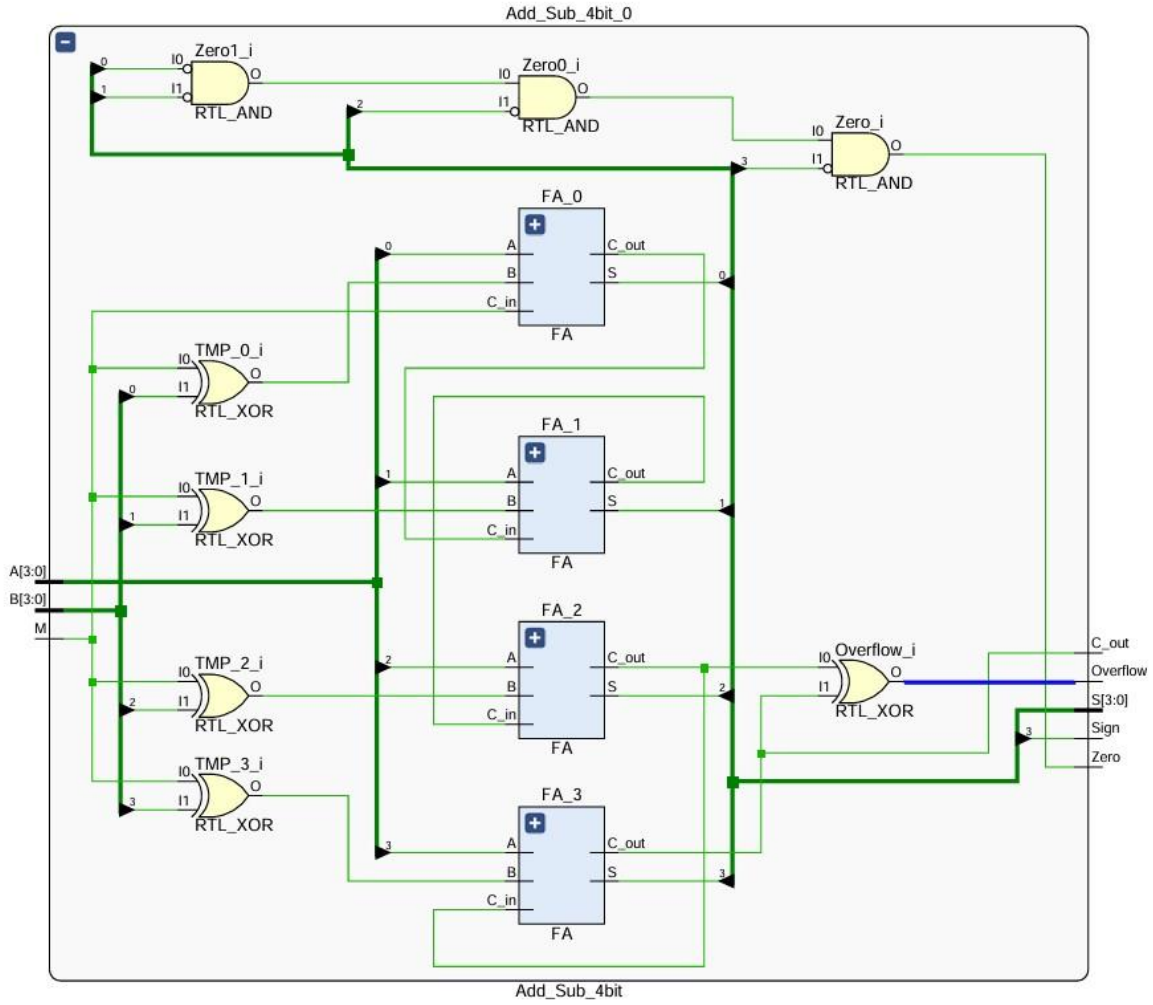

Untitled 5
Name
Value
Clk
Res
Carry_ou
Overflow
Sign
Zero
S_L...:0]
[3]
[2]
[1]
[0]
S_7S...:0]
[6]
[5]
[4]
[3]
[2]
[1]
[0]
Ano...:0]
[3]
[2]
[1]
[0]
0 ns
100 ns
200 ns
300 ns
400 ns
500 ns
600 ns
700 ns
800 ns
900 ns
1,000.000 ns
0
1
3
6
40
79
30
02
e

Constraints file

```
## Clock signal
set_property PACKAGE_PIN W5 [get_ports Clk]
      set_property IOSTANDARD LVCMOS33 [get_ports Clk]
      create_clock -add -name sys_clk_pin -period 10.00 -waveform {0 5} [get_ports Clk]


## LEDs
set_property PACKAGE_PIN U16 [get_ports {S_LED[0]}]
      set_property IOSTANDARD LVCMOS33 [get_ports {S_LED[0]}]
set_property PACKAGE_PIN E19 [get_ports {S_LED[1]}]
      set_property IOSTANDARD LVCMOS33 [get_ports {S_LED[1]}]
set_property PACKAGE_PIN U19 [get_ports {S_LED[2]}]
      set_property IOSTANDARD LVCMOS33 [get_ports {S_LED[2]}]
set_property PACKAGE_PIN V19 [get_ports {S_LED[3]}]
      set_property IOSTANDARD LVCMOS33 [get_ports {S_LED[3]}]


set_property PACKAGE_PIN P3 [get_ports {Sign}]
      set_property IOSTANDARD LVCMOS33 [get_ports {Sign}]
set_property PACKAGE_PIN U1 [get_ports {Overflow}]
      set_property IOSTANDARD LVCMOS33 [get_ports {Overflow}]
set_property PACKAGE_PIN P1 [get_ports {Zero}]
      set_property IOSTANDARD LVCMOS33 [get_ports {Zero}]
set_property PACKAGE_PIN L1 [get_ports {Carry_out}]
      set_property IOSTANDARD LVCMOS33 [get_ports {Carry_out}]



##7 segment display
set_property PACKAGE_PIN W7 [get_ports {S_7Seg[0]}]
      set_property IOSTANDARD LVCMOS33 [get_ports {S_7Seg[0]}]
```

```
set_property PACKAGE_PIN W6 [get_ports {S_7Seg[1]}]
      set_property IOSTANDARD LVCMOS33 [get_ports {S_7Seg[1]}]
set_property PACKAGE_PIN U8 [get_ports {S_7Seg[2]}]
      set_property IOSTANDARD LVCMOS33 [get_ports {S_7Seg[2]}]
set_property PACKAGE_PIN V8 [get_ports {S_7Seg[3]}]
      set_property IOSTANDARD LVCMOS33 [get_ports {S_7Seg[3]}]
set_property PACKAGE_PIN U5 [get_ports {S_7Seg[4]}]
      set_property IOSTANDARD LVCMOS33 [get_ports {S_7Seg[4]}]
set_property PACKAGE_PIN V5 [get_ports {S_7Seg[5]}]
      set_property IOSTANDARD LVCMOS33 [get_ports {S_7Seg[5]}]
set_property PACKAGE_PIN U7 [get_ports {S_7Seg[6]}]
      set_property IOSTANDARD LVCMOS33 [get_ports {S_7Seg[6]}]


set_property PACKAGE_PIN U2 [get_ports {Anode[0]}]
      set_property IOSTANDARD LVCMOS33 [get_ports {Anode[0]}]
set_property PACKAGE_PIN U4 [get_ports {Anode[1]}]
      set_property IOSTANDARD LVCMOS33 [get_ports {Anode[1]}]
set_property PACKAGE_PIN V4 [get_ports {Anode[2]}]
      set_property IOSTANDARD LVCMOS33 [get_ports {Anode[2]}]
set_property PACKAGE_PIN W4 [get_ports {Anode[3]}]
      set_property IOSTANDARD LVCMOS33 [get_ports {Anode[3]}]


##Buttons
set_property PACKAGE_PIN U17 [get_ports Res]
      set_property IOSTANDARD LVCMOS33 [get_ports Res]
```

Additional features of our design



- We included an **overflow flag** and a **sign flag** additionally.
- **Sign Flag**: Signed number are represented on computers as 2's complements. In this representation, the most significant bit (MSB) is 1 for negative numbers and zero for positive numbers. The sign bit of the output of the last math or logic operation is copied to the sign flag.
- **Overflow Flag**: When the target sign suddenly changes during addition or subtraction, set the overflow flag. When a transfer from the MSB varies from a transfer to the MSB, the overflow flag is activated.

Problems faced and how we managed to solve them

1. Problems faced while implementing the Register Bank

Register bank needs to be implemented with a reset because all the registers need to be reset to zero when the reset button is pressed. To reset a 4-bit register, we can first construct a D Flip Flop with reset and then connect four of them. However, using that method will require up to 28 D Flip Flops for the necessary 7 registers. This will increase the number of components used. But, we could simply do it as below.

```
process (Clk,Res) begin
if (Res='0') then
    if (rising_edge(Clk)) then
        if En = '1' then
            Q <= D;
        end if;
    end if;
else
    Q <= "0000";
end if;
end process;
```

With the above modification, we can optimize our register bank instead of creating 04 D flip flops for each register.

2. Problems faced while implementing the Instruction Decoder.

When constructing the instruction decoder we only gave values to the ports that we needed for the current execution process. So, the values stored during the previous execution process in those extra ports are still there. As a result, we got an incorrect output. We solved this problem by initializing the unused ports for the current execution process into "0"

MOVI INSTRUCTION

```
Reg_S_1 <= "000";
Reg_S_2 <= "000";
Add_Sub <= '0';
JMP_Address <= "000";
```

ADD INSTRUCTION

```
JMP_Address <= "000";
IV<="0000";
```

NEG INSTRUCTION

```
JMP_Address <= "000";
IV<="0000";
```

JZR INSTRUCTION

```
LS  <= '0';
Add_Sub<='0';
Reg_EN <= "000";
IV <= "0000";
Reg_S_2 <= "000";
```

Conclusions

The lab concludes with successfully creating and implementing a 4-bit nano processor capable of executing a specific set of instructions. Through this project, we gained practical experience in digital design, including developing arithmetic units, decoding instructions, and constructing multiplexers. The final stage involves testing the nano processor's functionality on a development board. Additionally, teamwork is emphasized, providing opportunities to improve communication, coordination, and collaboration skills.