Lab 9-10 – Nano Processor Design

Group 59

❖ Team members:

210613U-S.M.C.R. Siriwardhana:

- Register Bank
- 8-way 4-bit Mux
- Instruction decoder
- Program counter

210194H-S.L. Gunasekara

- 2-way 4-bit Mux
- 2-way 3-bit Mux
- 3-bit Adder
- 4-bit Add/Sub unit

❖ Lab Task:

design a 4-bit processor as given in the Lab capable of executing 4 instructions.

❖ Assembly Program

```
"100010001010", -- Movi R1,10
"100100000001", -- Movi R2,1
"010100000000", -- Neg R2
"100000000000", -- Movi R0,0
"000010100000", -- Add R1,R2
"110010000111",  -- Jzr R1,7
"110000000011" -- Jzr R0,3
```

➢ 2-way 3-bit Mux – Design code

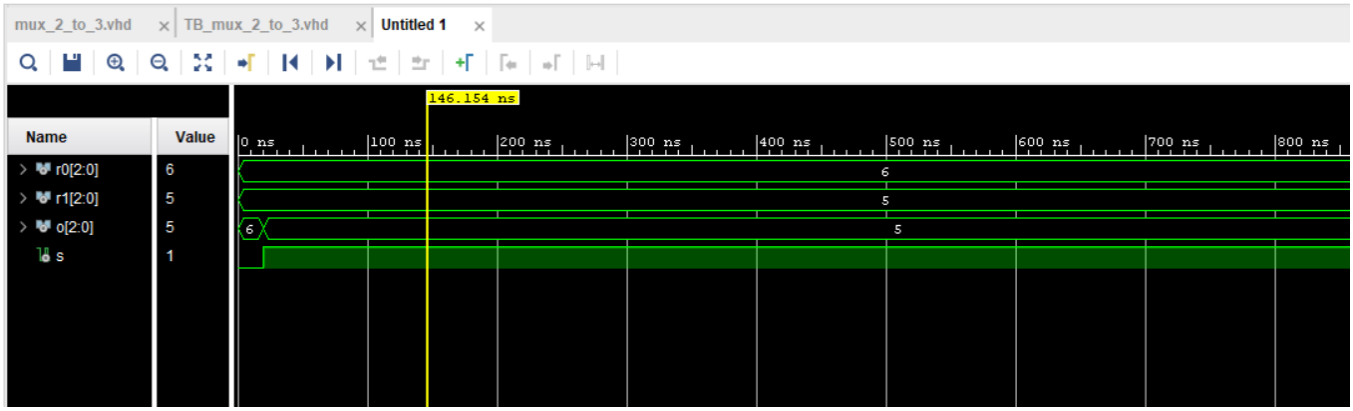
```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;

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

entity mux_2_to_3 is
   Port ( r0 : in STD_LOGIC_VECTOR (2 downto 0);
          r1 : in STD_LOGIC_VECTOR (2 downto 0);
          o : out STD_LOGIC_VECTOR (2 downto 0);
          s : in STD_LOGIC);
end mux_2_to_3;

architecture Behavioral of mux_2_to_3 is

begin
o <= r0 when s='0' else r1;
end Behavioral;
```

- Test bench code:

```
entity TB_mux_2_to_3 is
--  Port ( );
end TB_mux_2_to_3;

architecture Behavioral of TB_mux_2_to_3 is
component mux_2_to_3
   Port ( r0 : in STD_LOGIC_VECTOR (2 downto 0);
          r1 : in STD_LOGIC_VECTOR (2 downto 0);
          o : out STD_LOGIC_VECTOR (2 downto 0);
          s : in STD_LOGIC);
end component;
   signal r0: STD_LOGIC_VECTOR (2 downto 0);
   signal r1: STD_LOGIC_VECTOR (2 downto 0);
   signal o: STD_LOGIC_VECTOR (2 downto 0);
   signal s: STD_LOGIC;
begin
uut:mux_2_to_3
   port map(
      r0=>r0,
      r1=>r1,
      o=>o,
      s=>s);

   PROCESS
   BEGIN

   r0<="110";
   r1<="101";
```

```
    s<='0';
    wait for 20ns;
    s<='1';
    wait;
    END PROCESS;

end Behavioral;
```

simulation:



➢ 2-way 4-bit Mux- Design code

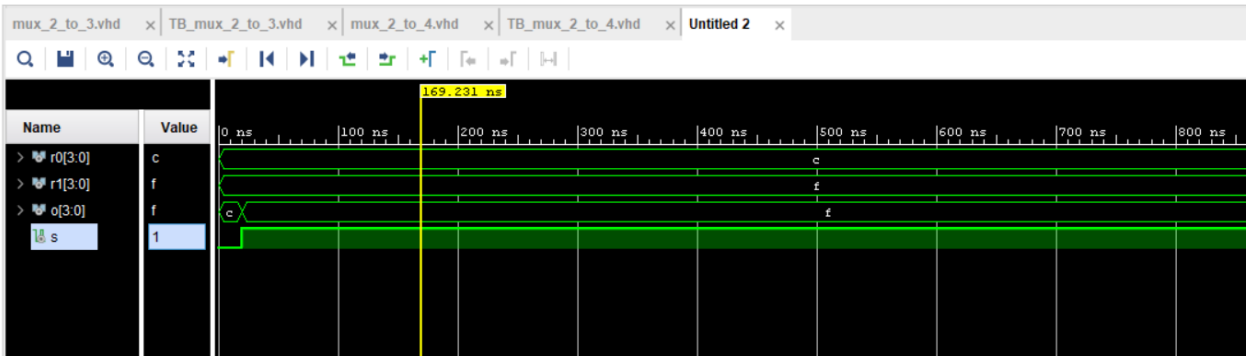
```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;

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

entity mux_2_to_4 is
Port ( r0 : in STD_LOGIC_VECTOR (3 downto 0);
       r1 : in STD_LOGIC_VECTOR (3 downto 0);
       o : out STD_LOGIC_VECTOR (3 downto 0);
       s : in STD_LOGIC);
end mux_2_to_4;

architecture Behavioral of mux_2_to_4 is

begin
o <= r0 when s='0' else r1;
end Behavioral;
```

- Test bench code

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

-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;

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

entity TB_mux_2_to_4 is
--  Port ( );
end TB_mux_2_to_4;

architecture Behavioral of TB_mux_2_to_4 is
component mux_2_to_4
    Port ( r0 : in STD_LOGIC_VECTOR (3 downto 0);
           r1 : in STD_LOGIC_VECTOR (3 downto 0);
           o : out STD_LOGIC_VECTOR (3 downto 0);
           s : in STD_LOGIC);
end component;
    signal r0: STD_LOGIC_VECTOR (3 downto 0);
    signal r1: STD_LOGIC_VECTOR (3 downto 0);
    signal o: STD_LOGIC_VECTOR (3 downto 0);
    signal s: STD_LOGIC;
```

```
begin
uut:mux_2_to_4
   port map(
      r0=>r0,
      r1=>r1,
      o=>o,
      s=>s);

PROCESS
BEGIN
r0<="1100";
r1<="1111";
s<='0';
wait for 20ns;
s<='1';
wait;
END PROCESS;



end Behavioral;
```

simulation:


mux_2_to_3.vhd
TB_mux_2_to_3.vhd
mux_2_to_4.vhd
TB_mux_2_to_4.vhd
Untitled 2
169.231 ns
Name
Value
r0[3:0]
r1[3:0]
o[3:0]
s
c
f
f
1
0 ns
100 ns
200 ns
300 ns
400 ns
500 ns
600 ns
700 ns
800 ns

➢ 8-way 4-bit Mux – Design code

```
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_Double is
    Port ( r0 : in STD_LOGIC_VECTOR (3 downto 0);
           r1 : in STD_LOGIC_VECTOR (3 downto 0);
           r2 : in STD_LOGIC_VECTOR (3 downto 0);
           r3 : in STD_LOGIC_VECTOR (3 downto 0);
           r4 : in STD_LOGIC_VECTOR (3 downto 0);
           r5 : in STD_LOGIC_VECTOR (3 downto 0);
           r6 : in STD_LOGIC_VECTOR (3 downto 0);
           r7 : in STD_LOGIC_VECTOR (3 downto 0);
           oa : out STD_LOGIC_VECTOR (3 downto 0);
           ob : out STD_LOGIC_VECTOR (3 downto 0);
           sa : in STD_LOGIC_VECTOR (2 downto 0);
           sb : in STD_LOGIC_VECTOR (2 downto 0));
end Mux_8_Way_Double;
```

```
architecture Behavioral of Mux_8_Way_Double is

begin
process (sa, r0, r1, r2, r3, r4, r5, r6, r7)
begin
if sa="000" then
   oa <= r0;
elsif sa="001" then
   oa <= r1;
elsif sa="010" then
   oa <= r2 ;
elsif sa="011" then
   oa <= r3;
elsif sa="100" then
   oa <= r4;
elsif sa="101" then
   oa <= r5;
elsif sa="110" then
   oa <= r6;
else
   oa <= r7;
end if;

end process;

process (sb, r0, r1, r2, r3, r4, r5, r6, r7)
begin
if sb="000" then
   ob <= r0;
```

```
elsif sb="001" then
   ob <= r1;
elsif sb="010" then
   ob <= r2 ;
elsif sb="011" then
   ob <= r3;
elsif sb="100" then
   ob <= r4;
elsif sb="101" then
   ob <= r5;
elsif sb="110" then
   ob <= r6;
else
   ob <= r7;
end if;

end process;

end Behavioral;
```

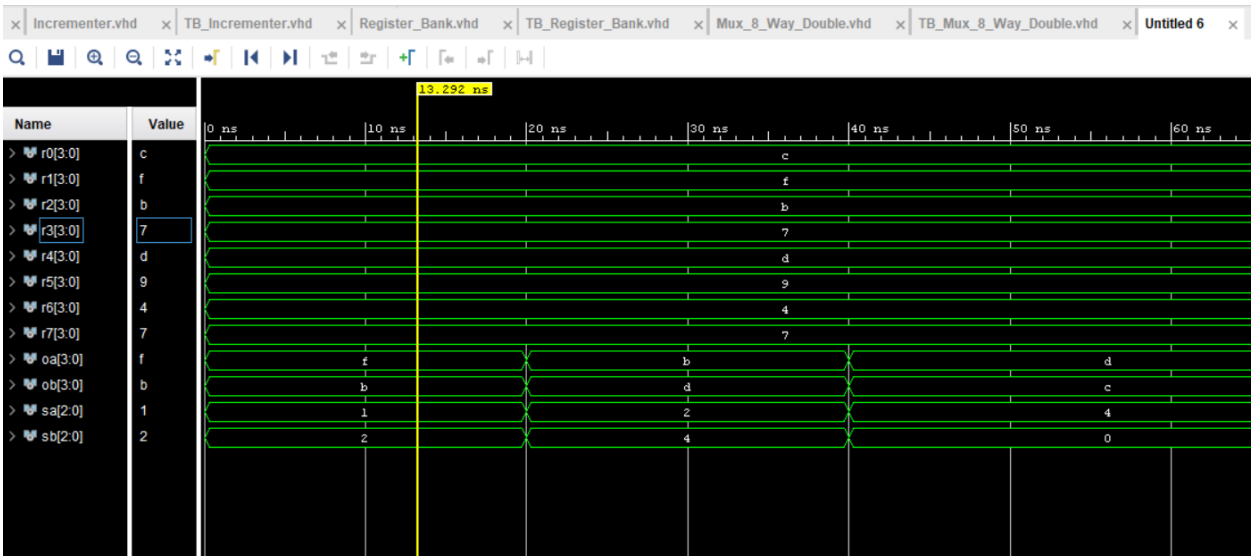
- Test bench code:

```
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_Double is
--  Port ( );
end TB_Mux_8_Way_Double;

architecture Behavioral of TB_Mux_8_Way_Double is

component Mux_8_Way_Double is
    Port ( r0 : in STD_LOGIC_VECTOR (3 downto 0);
           r1 : in STD_LOGIC_VECTOR (3 downto 0);
           r2 : in STD_LOGIC_VECTOR (3 downto 0);
           r3 : in STD_LOGIC_VECTOR (3 downto 0);
           r4 : in STD_LOGIC_VECTOR (3 downto 0);
           r5 : in STD_LOGIC_VECTOR (3 downto 0);
           r6 : in STD_LOGIC_VECTOR (3 downto 0);
           r7 : in STD_LOGIC_VECTOR (3 downto 0);
```

```
        oa : out STD_LOGIC_VECTOR (3 downto 0);
        ob : out STD_LOGIC_VECTOR (3 downto 0);
        sa : in STD_LOGIC_VECTOR (2 downto 0);
        sb : in STD_LOGIC_VECTOR (2 downto 0));
end component;

signal r0, r1, r2, r3, r4, r5, r6, r7, oa, ob : STD_LOGIC_VECTOR (3 downto 0);
signal sa, sb : STD_LOGIC_VECTOR (2 downto 0);

begin

mux_8_way_double_1: Mux_8_Way_Double port map(
   r0=>r0,
   r1=>r1,
   r2=>r2,
   r3=>r3,
   r4=>r4,
   r5=>r5,
   r6=>r6,
   r7=>r7,
   oa=>oa,
   ob=>ob,
   sa=>sa,
   sb=>sb);

PROCESS
   BEGIN
   r0<="1100";
   r1<="1111";
```

```
    r2<="1011";
    r3<="0111";
    r4<="1101";
    r5<="1001";
    r6<="0100";
    r7<="0111";

    sa<="001";
    sb<="010";
    wait for 20ns;
    sa<="010";
    sb<="100";
    wait for 20ns;
    sa<="100";
    sb<="000";
    wait;
END PROCESS;

end Behavioral;
```

simulation:


Incrementer.vhd
TB_Incrementer.vhd
Register_Bank.vhd
TB_Register_Bank.vhd
Mux_8_Way_Double.vhd
TB_Mux_8_Way_Double.vhd
Untitled 6
13.292 ns
Name
Value
0 ns
10 ns
20 ns
30 ns
40 ns
50 ns
60 ns
r0[3:0] c
r1[3:0] f
r2[3:0] b
r3[3:0] 7
r4[3:0] d
r5[3:0] 9
r6[3:0] 4
r7[3:0] 7
oa[3:0] f
ob[3:0] b
sa[2:0] 1
sb[2:0] 2

-

➢ Arithmetic unit – Design code

```
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.

entity Arithmatic_Unit is
    Port ( A : in STD_LOGIC_VECTOR (3 downto 0);
       B : in STD_LOGIC_VECTOR (3 downto 0);
       Add_Sub_Sel : in STD_LOGIC;
       S : out STD_LOGIC_VECTOR (3 downto 0);
       C_out : out STD_LOGIC;
       Zero : out STD_LOGIC);
--       attribute use_dsp : string;
--       attribute use_dsp of Arithmatic_Unit : entity is "yes";
end Arithmatic_Unit;

architecture Behavioral of Arithmatic_Unit is

component AddSubUnit Port (
          a:in std_logic_vector (3 downto 0);
          b:in std_logic_vector (3 downto 0);
```

```
        s:out std_logic_vector (3 downto 0);
        add : in STD_LOGIC;
        c_out : out STD_LOGIC);
end component;

signal O : STD_LOGIC_VECTOR (3 downto 0);

begin

add_sub_unit_0 : AddSubUnit
port map (
A => B,
B => A,
ADD => Add_Sub_Sel,
C_OUT => C_out,
S => O);

S <= O;
Zero <= NOR(O);

end Behavioral;
```

- Test bench code

```
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_Arithmatic_Unit is
--  Port ( );
end TB_Arithmatic_Unit;

architecture Behavioral of TB_Arithmatic_Unit is

component Arithmatic_Unit port
( a:in std_logic_vector (3 downto 0);
        b:in std_logic_vector (3 downto 0);
        s:out std_logic_vector (3 downto 0);
        Add_Sub_Sel : in STD_LOGIC;
        zero : out STD_LOGIC;
        C_out : out STD_LOGIC);
end component;
```

```
signal a,b,s:std_logic_vector (3 downto 0);
signal overflow,Add_Sub_Sel,zero:std_logic;

begin
uut : Arithmatic_Unit port map(
a=>a,
b=>b,
s=>s,
c_out=>overflow,
zero=>zero,
Add_Sub_Sel=>Add_Sub_Sel);

process
begin


Add_Sub_Sel<='0';
a<="1111";
b<="0101";
wait for 20ns;

Add_Sub_Sel<='1';
a<="1111";
b<="0011"; --randomly selected
wait for 20ns;

Add_Sub_Sel<='0';
a<="0000";
```

```
b<="0000";

wait;


end process;

end Behavioral;
```

Simulation:

➢ Incrementor -Design code

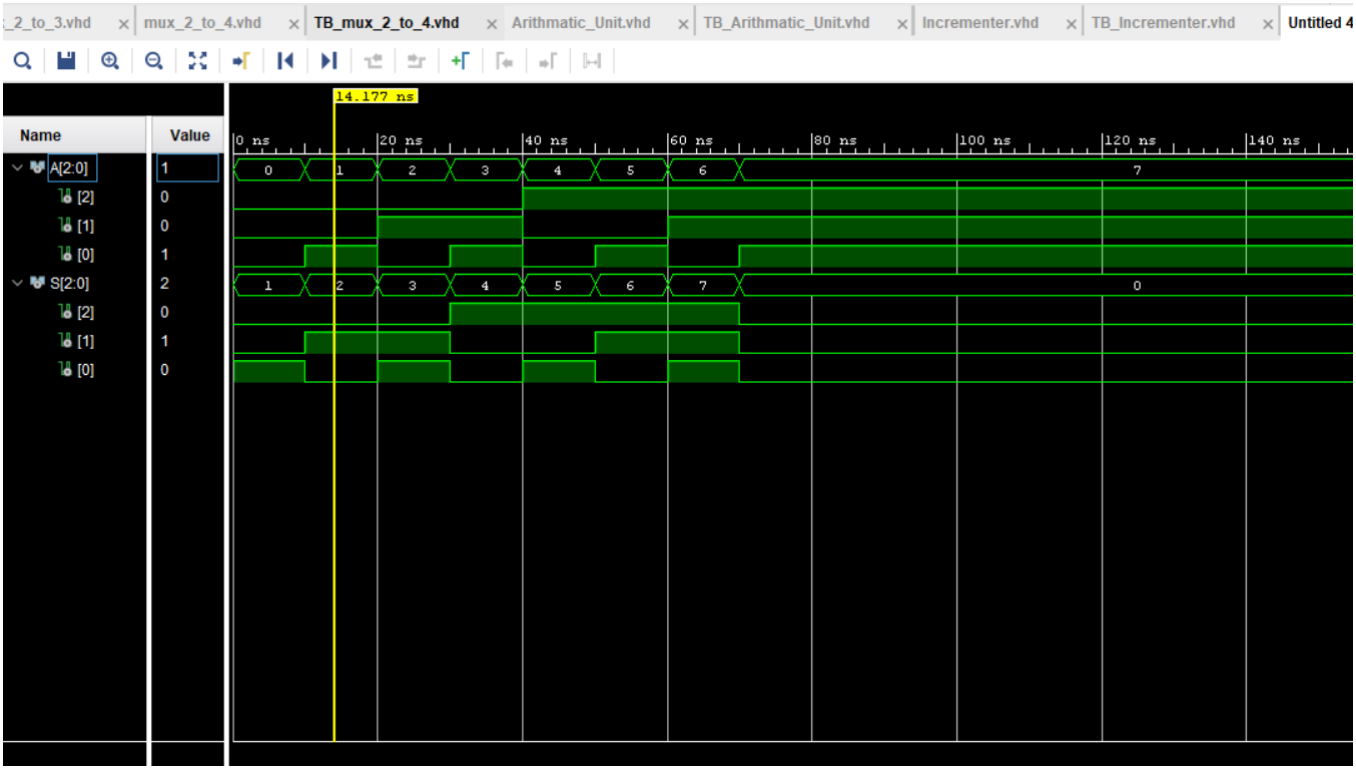
```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
-- Uncomment the following library declaration if instantiating
-- any Xilinx leaf cells in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity Incrementer is
   Port ( A:in std_logic_vector (2 downto 0);
          S:out std_logic_vector (2 downto 0));
--          attribute use_dsp : string;
--          attribute use_dsp of Incrementer : entity is "yes";

end Incrementer;
architecture Behavioral of Incrementer is

begin
S <= A + 1;
end Behavioral;
```

- Test bench code:

```
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_Incrementer is
--  Port ( );
end TB_Incrementer;

architecture Behavioral of TB_Incrementer is

component Incrementer
   Port ( A : in STD_LOGIC_VECTOR (2 downto 0);
          S : out STD_LOGIC_VECTOR (2 downto 0));
end component;

signal A, S : STD_LOGIC_VECTOR (2 downto 0);

begin

UUT : Incrementer
```

```
port map (
   A => A,
   S => S);

process begin
A <= "000";
wait for 10 ns;
A <= "001";
wait for 10 ns;
A <= "010";
wait for 10 ns;
A <= "011";
wait for 10 ns;
A <= "100";
wait for 10 ns;
A <= "101";
wait for 10 ns;
A <= "110";
wait for 10 ns;
A <= "111";
wait for 10 ns;
wait;

end process;

end Behavioral;
```

simulation:


_2_to_3.vhd
mux_2_to_4.vhd
TB_mux_2_to_4.vhd
Arithmatic_Unit.vhd
TB_Arithmatic_Unit.vhd
Incrementer.vhd
TB_Incrementer.vhd
Untitled 4
14.177 ns
Name
Value
A[2:0]
[2]
[1]
[0]
S[2:0]
[2]
[1]
[0]
1
0
0
1
2
0
1
0
0 ns
20 ns
40 ns
60 ns
80 ns
100 ns
120 ns
140 ns
0
1
2
3
4
5
6
7
1
2
3
4
5
6
7
0

➢ Register bank – Design code:

```
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 ( data : in STD_LOGIC_VECTOR (3 downto 0);
           clk : in STD_LOGIC;
           en : in STD_LOGIC_VECTOR (2 downto 0);
           r0 : out STD_LOGIC_VECTOR (3 downto 0) := "0000";
           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 reg port(
      D : in STD_LOGIC_VECTOR (3 downto 0);
      en : in STD_LOGIC;
      clk : in STD_LOGIC;
      Q : out STD_LOGIC_VECTOR (3 downto 0));
end component;

component decoder_3_to_8
Port ( I : in STD_LOGIC_VECTOR (2 downto 0);
      Y : out STD_LOGIC_VECTOR (7 downto 0));
end component;

signal d_en:std_logic_vector (7 downto 0);

begin

dec:decoder_3_to_8 port map(
   i=>en,
   y=>d_en);

reg1:reg port map(
   d=>data,
   en=>d_en(1),
   clk=>clk,
   q=>r1);
reg2:reg port map(
   d=>data,
   en=>d_en(2),
   clk=>clk,
```

```
    q=>r2);
reg3:reg port map(
    d=>data,
    en=>d_en(3),
    clk=>clk,
    q=>r3);
reg4:reg port map(
    d=>data,
    en=>d_en(4),
    clk=>clk,
    q=>r4);
reg5:reg port map(
    d=>data,
    en=>d_en(5),
    clk=>clk,
    q=>r5);
reg6:reg port map(
    d=>data,
    en=>d_en(6),
    clk=>clk,
    q=>r6);
reg7:reg port map(
    d=>data,
    en=>d_en(7),
    clk=>clk,
    q=>r7);

end Behavioral;
```

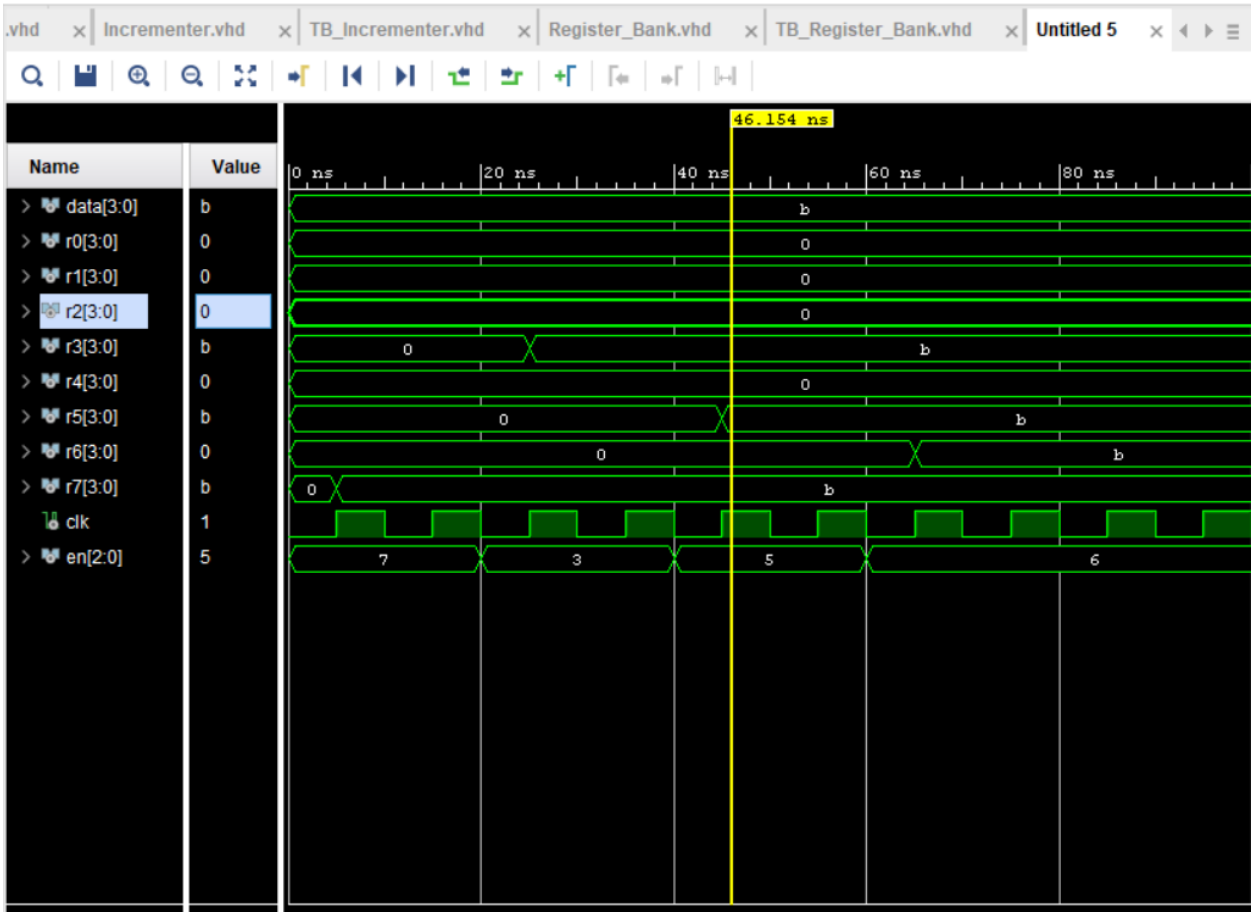
- Test bench code:

```
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
port (data : in STD_LOGIC_VECTOR (3 downto 0);
      clk : in STD_LOGIC;
      en : in STD_LOGIC_VECTOR (2 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 data : STD_LOGIC_VECTOR (3 downto 0);
signal r0, r1, r2, r3, r4, r5, r6, r7 : STD_LOGIC_VECTOR (3 downto 0) := "0000";
signal clk : STD_LOGIC := '0';
signal en : STD_LOGIC_VECTOR (2 downto 0);

begin
uut: Register_bank
       Port map (
       data=>data,
       clk=>clk,
       en=>en,
       r0=>r0,
       r1=>r1,
       r2=>r2,
       r3=>r3,
       r4=>r4,
       r5=>r5,
       r6=>r6,
       r7=>r7);

process
begin
   wait for 5ns;
   clk <= NOT clk;
end process;
```

```
process
begin
   data <= "1011";



   en <= "111";
   wait for 20ns;

   en <= "011";
   wait for 20ns;

   en <= "101";
   wait for 20ns;

   en <= "110";
   wait;

end process;
end Behavioral;
```

Simulation:



➢ Instruction decoder – Design code

```
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 (
   ins : in STD_LOGIC_VECTOR (11 downto 0);
         reg_en : out STD_LOGIC_VECTOR (2 downto 0) := "000";
         load_sel : out STD_LOGIC := '0';
         value : out STD_LOGIC_vector (3 downto 0) := "0000";
         mux_1 : out STD_LOGIC_VECTOR (2 downto 0) := "000";
         mux_2 : out STD_LOGIC_VECTOR (2 downto 0) := "000";
         aors : out STD_LOGIC := '0';
         jmp_flg : out STD_LOGIC := '0';
         jmp_add : out std_logic_vector (2 downto 0) := "000";
         check : in STD_LOGIC := '0');
--          attribute use_dsp : string;
--         attribute use_dsp of Instruction_Decoder : entity is "yes";
end Instruction_Decoder;

architecture Behavioral of Instruction_Decoder is
```

```
signal first_2_bits : std_logic_vector (1 downto 0);

begin

process (ins, check, first_2_bits)
begin
reg_en <= ins(9 downto 7);
jmp_add <= ins(2 downto 0);

first_2_bits <= ins(11 downto 10);
mux_1 <= ins(9 downto 7);
mux_2<=ins(6 downto 4);
value<=ins(3 downto 0);

if first_2_bits = "00" then
   aors<='1';
   load_sel<='0';
   jmp_flg<='0';
elsif first_2_bits = "01" then
   aors<='0';
   load_sel<='0';
   jmp_flg<='0';
elsif first_2_bits = "10" then
   aors<='1';
   load_sel<='1';
   jmp_flg<='0';
else
   aors<='1';
   load_sel<='0';
```

```
    jmp_flg<=check;
end if;


end process;


end Behavioral;
```

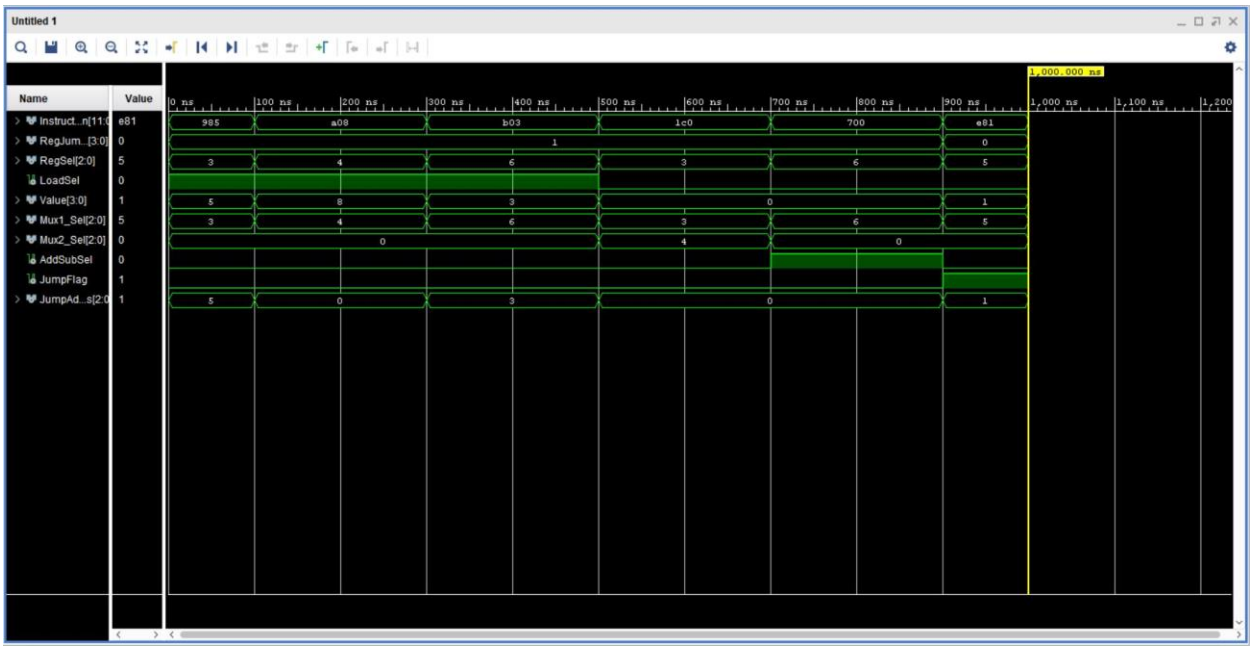
Test bench code:

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

-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--use IEEE.NUMERIC_STD.ALL;

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

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

architecture Behavioral of TB_Instruction_Decoder is

component Instruction_Decoder  Port ( ins : in STD_LOGIC_VECTOR (8 downto 0);
        reg_en : out STD_LOGIC_VECTOR (2 downto 0);
        load : out STD_LOGIC;
        value : out STD_LOGIC_vector (3 downto 0);
        mux_1 : out STD_LOGIC_VECTOR (2 downto 0);
        mux_2 : out STD_LOGIC_VECTOR (2 downto 0);
        aors : out STD_LOGIC;
        jmp_flg : out STD_LOGIC;
        jmp_add:out std_logic_vector (2 downto 0);
        check:in std_logic_vector (3 downto 0));
end component;
```

```
signal load, aors, jmp_flg : STD_LOGIC;
signal reg_en, mux_1, mux_2, jmp_add : STD_LOGIC_VECTOR (2 downto 0);
signal value, check : STD_LOGIC_vector (3 downto 0);
signal ins : STD_LOGIC_VECTOR (8 downto 0);

begin

uut: Instruction_Decoder port map(
   ins=>ins,
   reg_en=>reg_en,
   load=>load,
   value=>value,
   mux_1=>mux_1,
   mux_2=>mux_2,
   aors=>aors,
   jmp_flg=>jmp_flg,
   jmp_add=>jmp_add,
   check=>check);

process begin


   check <= "0000";
   ins <= "001011110"; --ADD R5, R7
   wait for 5ns;

   ins <= "100101001"; --MOVI R2, 9
   wait for 5ns;
```

```
    ins <= "010100000"; --NEG R2
    wait for 5ns;

    ins <= "110100100"; --JZR R2, 4
    wait for 5ns;
    check <= "1000";
    wait;
end process;

end Behavioral;
```

simulation:



➢ Program counter – Design code

```
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 Program_Counter is
    Port ( D : in STD_LOGIC_VECTOR (2 downto 0) := "000";
           Clk : in STD_LOGIC;
           Res : in STD_LOGIC := '0';
           Q : out std_logic_vector (2 downto 0) := "000");
--           attribute use_dsp : string;
--           attribute use_dsp of Program_Counter : entity is "yes";
end Program_Counter;
```

```
architecture Behavioral of Program_Counter is

begin
process (Clk, D) begin
   if (rising_edge(Clk)) then
      if Res = '1' then
       Q <= "000";
      else
       Q <= D;
      end if;
   end if;
end process;

end Behavioral;
```

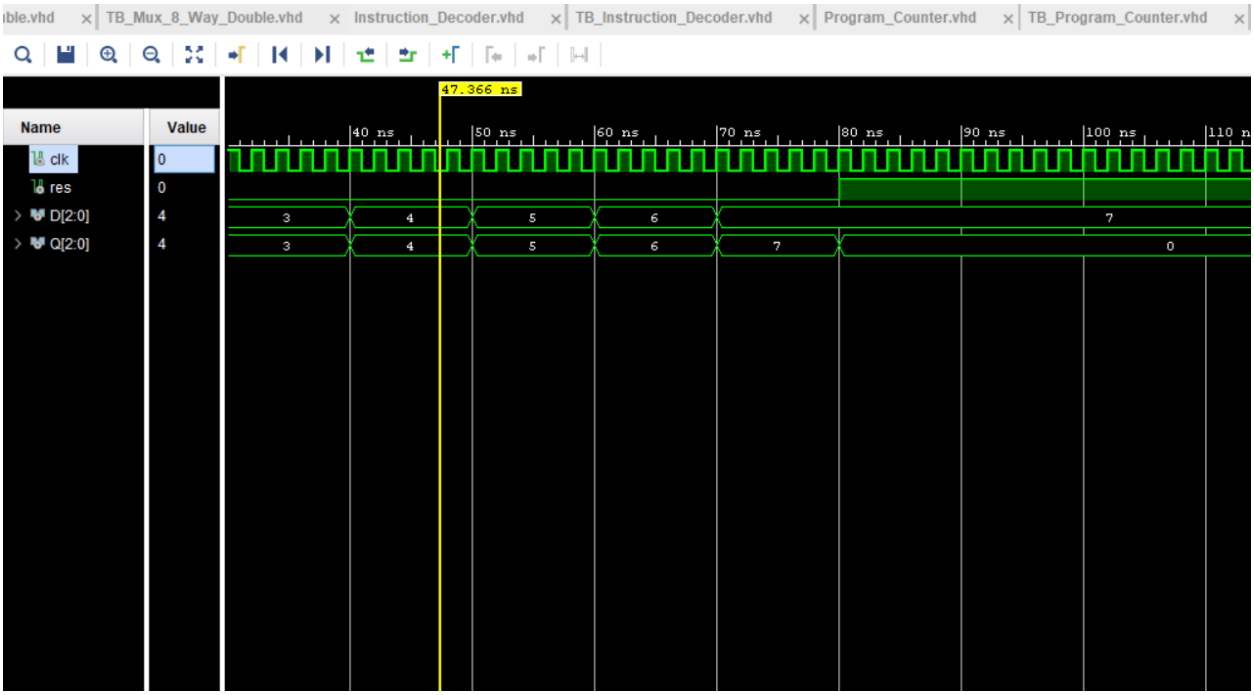
Test bench code:

```
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_Program_Counter is
--  Port ( );
end TB_Program_Counter;

architecture Behavioral of TB_Program_Counter is
component Program_Counter
   Port ( D : in STD_LOGIC_VECTOR (2 downto 0);
          clk : in STD_LOGIC;
          res : in STD_LOGIC ;
          Q : out std_logic_vector (2 downto 0));
end component;
signal clk : STD_LOGIC := '0';
signal res : STD_LOGIC := '0';
signal D,Q : STD_LOGIC_VECTOR (2 downto 0);

begin
UUT: Program_Counter
   port map(
      D=>D,
      clk=>clk,
      res=>res,
      Q=>Q);

process
begin
      clk<=not(clk);
      wait for 1ns;
end process;
```

```
process
begin
	wait for 80ns;
	res<=not(res);
end process;

process
begin
	D <= "000";
	wait for 10 ns;
	D <= "001";
	wait for 10 ns;
	D <= "010";
	wait for 10 ns;
	D <= "011";
	wait for 10 ns;
	D <= "100";
	wait for 10 ns;
	D <= "101";
	wait for 10 ns;
	D <= "110";
	wait for 10 ns;
	D <= "111";
	wait;
end process;

end Behavioral;
```

- Simulation:


ible.vhd
TB_Mux_8_Way_Double.vhd
Instruction_Decoder.vhd
TB_Instruction_Decoder.vhd
Program_Counter.vhd
TB_Program_Counter.vhd
47.366 ns
Name
Value
clk
0
res
0
D[2:0]
4
Q[2:0]
4
40 ns
50 ns
60 ns
70 ns
80 ns
90 ns
100 ns
3
4
5
6
7
0

➢ ROM- Design code

```
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 ( address : in STD_LOGIC_VECTOR (2 downto 0);
           data : 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 twelveSegment_ROM : rom_type := (
 "100010000011", --MOV R1, 3
"100100000001", --MOV R2, 1
"010100000000", --NEG R2
"001110010000", --ADD R7, R1
"000010100000", --ADD R1, R2
```

```
"110010000111", --JZR R1, 7
"110000000011", --JZR R0, 3
"110010000101"  --JZR R1, 5
 );

begin
data <= twelveSegment_ROM(to_integer(unsigned(address)));
end Behavioral;
```

Test bench code:

```
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
   Port ( address : in STD_LOGIC_VECTOR (2 downto 0);
          data : out STD_LOGIC_VECTOR (8 downto 0));
end component;

signal address: STD_LOGIC_VECTOR(2 downto 0);
signal data : STD_LOGIC_VECTOR (8 downto 0);

begin
uut: ROM
   Port map ( address => address,
              data => data);

process
begin


   address <= "111";
   wait for 20ns;

   address <= "011";
   wait for 20ns;

   address <= "101";
   wait for 20ns;

   address <= "000";
   wait for 20ns;
```

```
    address <= "110";
    wait;

end process;
end Behavioral;
```

Simulation:

- ## Microprocessor – Design code

```
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 MicroProcessor is
    Port ( reset : in STD_LOGIC := '0';
           clk : in STD_LOGIC;
           zero : out STD_LOGIC;
           overflow : out STD_LOGIC;
           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) := "1110");
end MicroProcessor;

architecture Behavioral of MicroProcessor is

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

```
component Register_Bank  Port ( data : in STD_LOGIC_VECTOR (3 downto 0);
        clk : in STD_LOGIC;
        en : in STD_LOGIC_VECTOR (2 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 Arithmatic_Unit Port ( a:in std_logic_vector (3 downto 0);
          b:in std_logic_vector (3 downto 0);
          s:out std_logic_vector (3 downto 0);
          add_sub_sel : in STD_LOGIC;
          C_out : out STD_LOGIC;
          zero:out std_logic);
end component;

component Incrementer  Port ( A:in std_logic_vector (2 downto 0);
        s:out std_logic_vector (2 downto 0));
end component;

component Program_Counter Port ( D : in STD_LOGIC_VECTOR (2 downto 0);
        clk : in STD_LOGIC;
        res : in STD_LOGIC;
        q:out std_logic_vector (2 downto 0));
```

```
end component;

component ROM  Port ( address : in STD_LOGIC_VECTOR (2 downto 0);
        data : out STD_LOGIC_VECTOR (11 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 Instruction_Decoder  Port ( ins : in STD_LOGIC_VECTOR (11 downto 0);
        reg_en : out STD_LOGIC_VECTOR (2 downto 0);
        load_sel : out STD_LOGIC;
        value : out STD_LOGIC_vector (3 downto 0);
        mux_1 : out STD_LOGIC_VECTOR (2 downto 0);
        mux_2 : out STD_LOGIC_VECTOR (2 downto 0);
        aors : out STD_LOGIC;
        jmp_flg : out STD_LOGIC;
        jmp_add:out std_logic_vector (2 downto 0);
        check:in std_logic);
end component;

component mux_2_to_3 Port ( r0 : in STD_LOGIC_VECTOR (2 downto 0);
        r1 : in STD_LOGIC_VECTOR (2 downto 0);
        o : out STD_LOGIC_VECTOR (2 downto 0);
        s : in STD_LOGIC);
end component;

component mux_2_to_4 Port ( r0 : in STD_LOGIC_VECTOR (3 downto 0);
```

```
        r1 : in STD_LOGIC_VECTOR (3 downto 0);
        o : out STD_LOGIC_VECTOR (3 downto 0);
        s : in STD_LOGIC);
end component;

component Mux_8_Way_Double Port ( r0 : in STD_LOGIC_VECTOR (3 downto 0);
        r1 : in STD_LOGIC_VECTOR (3 downto 0);
        r2 : in STD_LOGIC_VECTOR (3 downto 0);
        r3 : in STD_LOGIC_VECTOR (3 downto 0);
        r4 : in STD_LOGIC_VECTOR (3 downto 0);
        r5 : in STD_LOGIC_VECTOR (3 downto 0);
        r6 : in STD_LOGIC_VECTOR (3 downto 0);
        r7 : in STD_LOGIC_VECTOR (3 downto 0);
        oa : out STD_LOGIC_VECTOR (3 downto 0);
        ob : out STD_LOGIC_VECTOR (3 downto 0);
        sa : in STD_LOGIC_VECTOR (2 downto 0);
        sb : in STD_LOGIC_VECTOR (2 downto 0));
end component;

signal reg_en:std_logic_vector (2 downto 0);
signal data_mux:std_logic_vector (3 downto 0);
signal r0,r1,r2,r3,r4,r5,r6,r7 :std_logic_vector (3 downto 0);
signal mux4a,mux4b :std_logic_vector (3 downto 0);
signal regmuxa,regmuxb :std_logic_vector (2 downto 0);
signal muxout:std_logic_vector (3 downto 0);
signal addsub:std_logic;
signal imvalue:std_logic_vector (3 downto 0);
signal load_select :std_logic;
signal adder_out:std_logic_vector (2 downto 0);
```

```
signal jmp_add:std_logic_vector (2 downto 0);
signal jmp_flag:std_logic;
signal zero_check:std_logic;
signal s_clk:std_logic;
signal muxcounterout:std_logic_vector (2 downto 0);
signal prog_counter:std_logic_vector (2 downto 0);
signal instruction_bus:std_logic_vector (11 downto 0);

begin

slow_clock_1 : Slow_Clock port map(
   Clk_in=>clk,
   Clk_out=>s_clk);

register_bank_1: register_bank port map(
   clk=>s_clk,
   en=>reg_en,
   data=>data_mux,
   r0=>r0,
   r1=>r1,
   r2=>r2,
   r3=>r3,
   r4=>r4,
   r5=>r5,
   r6=>r6,
   r7=>r7);

mux_8_way_double_1: Mux_8_Way_Double port map(
   r0=>r0,
```

```
    r1=>r1,
    r2=>r2,
    r3=>r3,
    r4=>r4,
    r5=>r5,
    r6=>r6,
    r7=>r7,
    oa=>mux4a,
    ob=>mux4b,
    sa=>regmuxa,
    sb=>regmuxb);

arithmatic_unit_1: Arithmatic_Unit port map(
    a=>mux4a,
    b=>mux4b,
    s=>muxout,
    add_sub_sel=>addsub,
    zero=>zero_check,
    c_out=>overflow);

mux2to4_1: mux_2_to_4 port map(
    r0=>muxout,
    r1=>imvalue,
    o=>data_mux,
    s=>load_select);

mux2to3_1: mux_2_to_3 port map(
    r0=>adder_out,
    r1=>jmp_add,
```

```
    s=>jmp_flag,
    o=>muxcounterout);

incrementer_1: Incrementer port map(
    a=>prog_counter,
    s=>adder_out);

instruction_decoder_1: instruction_decoder port map(
    ins=>instruction_bus,
    reg_en=>reg_en,
    load_sel=>load_select,
    value=>imvalue,
    mux_1=>regmuxa,
    mux_2=>regmuxb,
    aors=>addsub,
    jmp_flg=>jmp_flag,
    jmp_add=>jmp_add,
    check=>zero_check);

rom_1: rom port map(
    address=>prog_counter,
    data=>instruction_bus);

program_counter_1: Program_Counter port map(
    d=>muxcounterout,
    clk=>s_clk,
    res=>reset,
    q=>prog_counter);
```

```
led_lut: LUT_16_7 port map(
   address=>r7,
   data=>S_7Seg);

led <= r7;
zero <= zero_check;

end Behavioral;
```

Test bench code:

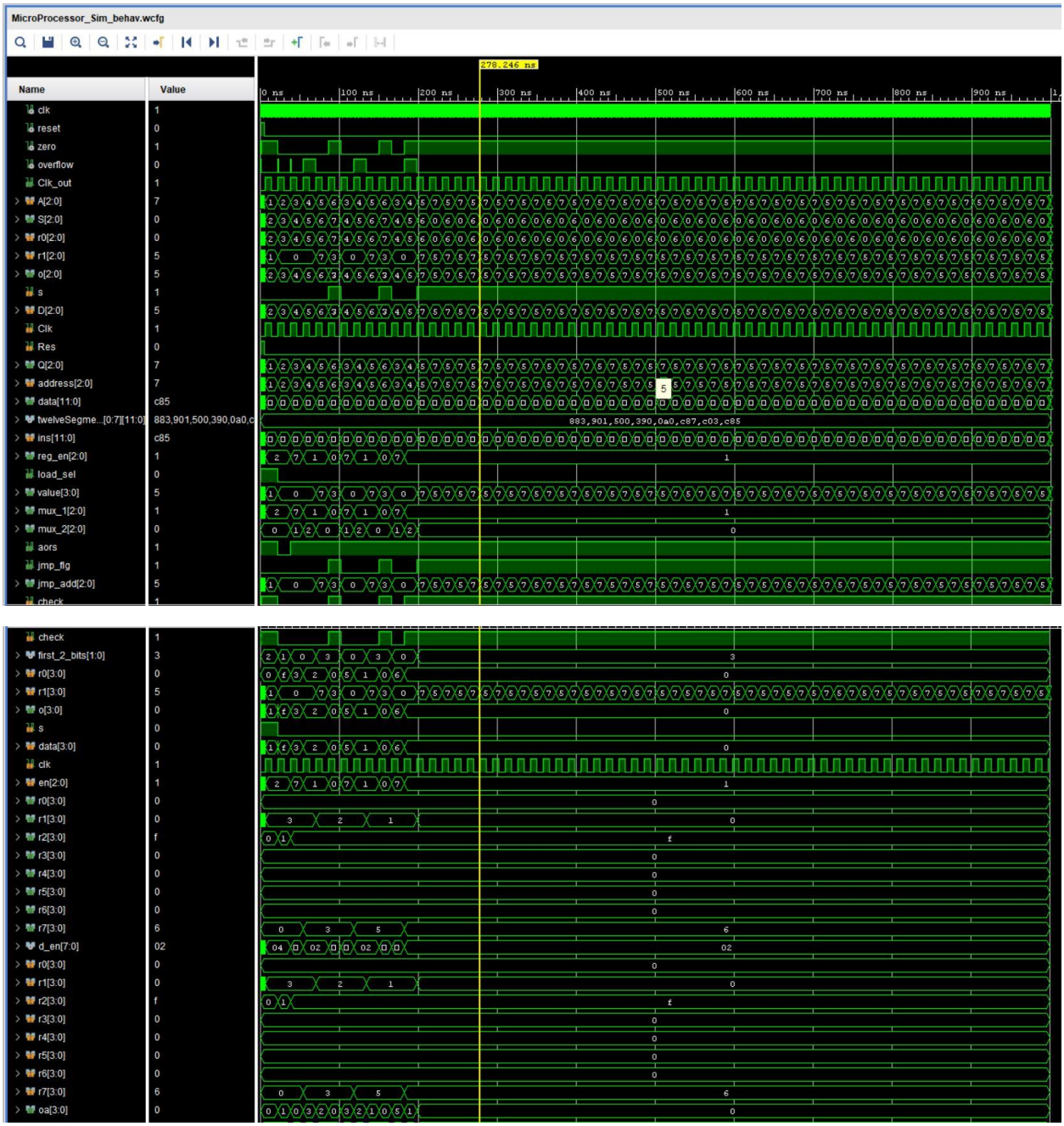
```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

-- 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 MicroProcessor_Sim is
--  Port ( );
end MicroProcessor_Sim;

architecture Behavioral of MicroProcessor_Sim is

component MicroProcessor  Port ( reset : in STD_LOGIC;
```

```
        clk : in STD_LOGIC;
        zero : out STD_LOGIC;
        overflow : out STD_LOGIC;
        led:out std_logic_vector (3 downto 0));
end component;

signal clk:std_logic:='0';
signal reset,zero,overflow:std_logic := '0';
signal led: std_logic_vector (3 downto 0);

begin
MicroProcessor_0 : MicroProcessor port map(
   reset=>reset,
   clk=>clk,
   zero=>zero,
   overflow=>overflow,
   led => led);
process
begin
   clk<=not(clk);
   wait for 1ns;
end process;

process
   begin
   reset<='1';
   wait for 5ns;
   reset<='0';
   wait;
```

end process;

end Behavioral;

simulation:



➢ Conclusions

- We can build a k-way n-bit MUX by adding n number of k to1 MUXs
- We can larger arithmetic operations by using a larger buses and registers
- We can subtract a number by adding the 2’s compliment of that number.
- We can expand the instruction set by expanding the instruction bus.
- We can improve number of instructions by expanding the ROM, program counter
- We cab define a negative flag by checking the MSB of a selected register.