IT 451: Computer Organization and Architecture (Sessional)

Name – Anupam Sanidhya

Enrolment number – 510817049

Roll number – 42 (Hy)

Assignment 5

32 bit ALU -

VHDL MODULE:

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

entity ALU_32bit is
    Port ( A,B : in  STD_LOGIC_VECTOR(31 downto 0);
                                    CLK,carry : in STD_LOGIC;
           Set : in  STD_LOGIC_VECTOR(3 downto 0);
                                    O: out STD_LOGIC_VECTOR(31 downto 0));
end ALU_32bit;

architecture Behavioral of ALU_32bit is
signal temp: STD_LOGIC_VECTOR(31 downto 0);
```

```
begin
        process(CLK)
        begin
                if(rising_edge(CLK)) then
                        if(unsigned(Set)=0) then
                                temp<=A;
                        elsif(unsigned(Set)=1) then
                                temp<=A+x"1";
                        elsif(unsigned(Set)=2) then
                                temp<=A-x"1";
                        elsif(unsigned(Set)=3) then
                                temp<=B;
                        elsif(unsigned(Set)=4) then
                                temp<=B+x"1";
                        elsif(unsigned(Set)=5) then
                                temp<=B-x"1";
                        elsif(unsigned(Set)=6) then
                                temp<=A+B;
                        elsif(unsigned(Set)=7) then
                                temp<=A+B+carry;
                        elsif(unsigned(Set)=8) then
                                temp<= not A;
                        elsif(unsigned(Set)=9) then
                                temp<= not B;
                        elsif(unsigned(Set)=10) then
                                temp<= A and B;
                        elsif(unsigned(Set)=11) then
                                temp<= A or B;
```

```
                elsif(unsigned(Set)=12) then
                        temp<= A nand B;
                elsif(unsigned(Set)=13) then
                        temp<= A nor B;
                elsif(unsigned(Set)=14) then
                        temp<= A xor B;
                elsif(unsigned(Set)=15) then
                        temp<= A xnor B;
                end if;

            end if;
        end process;
        O<=temp;

end Behavioral;
```

TEST BENCH :

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
USE ieee.std_logic_UNSIGNED.ALL;
USE ieee.numeric_std.ALL;

ENTITY ALU_32bit_test IS
END ALU_32bit_test;

ARCHITECTURE behavior OF ALU_32bit_test IS
```

```
   -- Component Declaration for the Unit Under Test (UUT)

   COMPONENT ALU_32bit
   PORT(
        A : IN  std_logic_vector(31 downto 0);
        B : IN  std_logic_vector(31 downto 0);
        CLK : IN  std_logic;
        carry : IN  std_logic;
        Set : IN  std_logic_vector(3 downto 0);
        O : OUT  std_logic_vector(31 downto 0)
       );
   END COMPONENT;



  --Inputs
  signal A : std_logic_vector(31 downto 0) := (others => '0');
  signal B : std_logic_vector(31 downto 0) := (others => '0');
  signal CLK : std_logic := '0';
  signal carry : std_logic := '0';
  signal Set : std_logic_vector(3 downto 0) := (others => '0');

          --Outputs
  signal O : std_logic_vector(31 downto 0);

  -- Clock period definitions
  constant CLK_period : time := 10 ns;

BEGIN
```

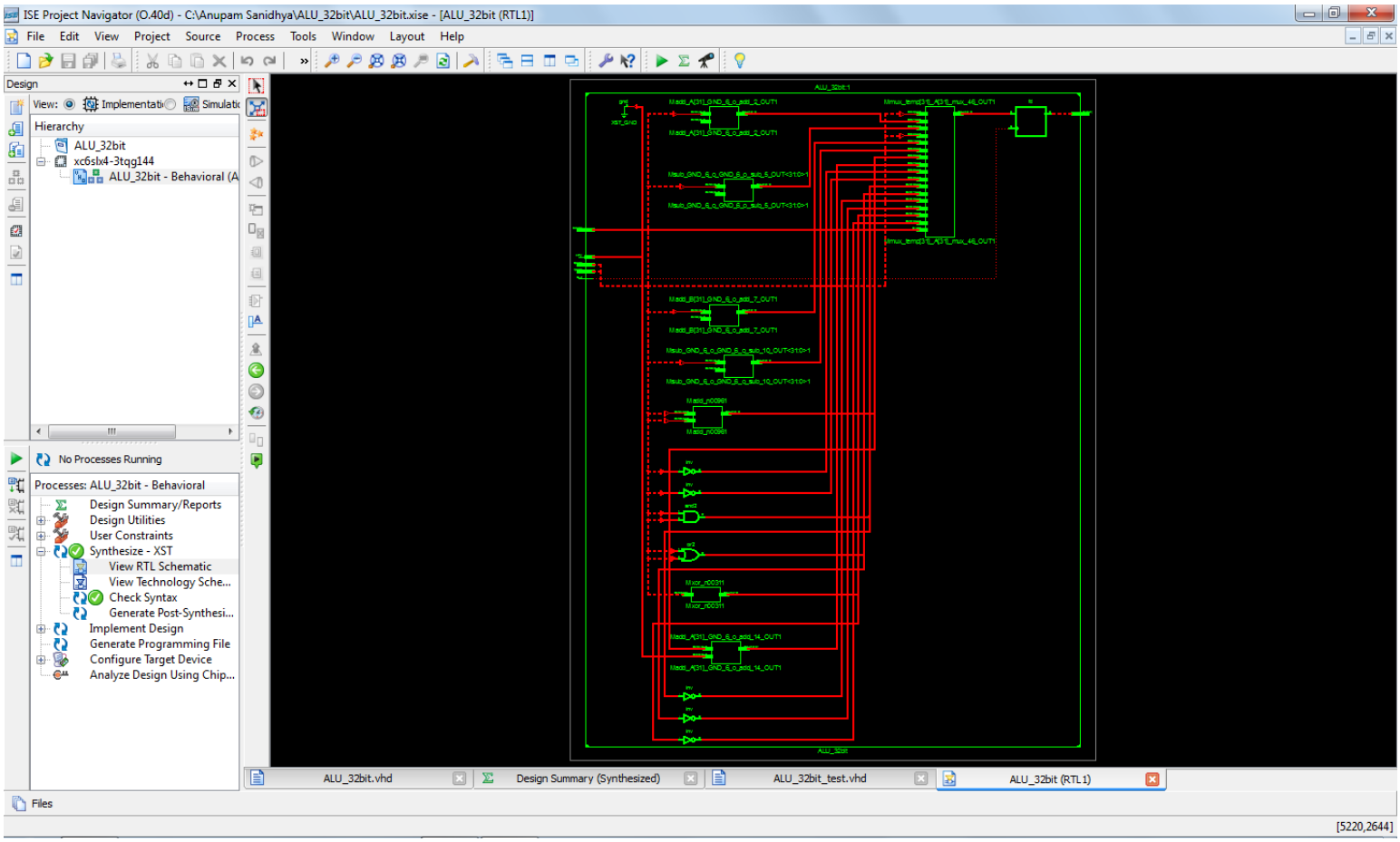
```
        -- Instantiate the Unit Under Test (UUT)
uut: ALU_32bit PORT MAP (
       A => A,
       B => B,
       CLK => CLK,
       carry => carry,
       Set => Set,
       O => O
     );

-- Clock process definitions
CLK_process :process
begin
                CLK <= '0';
                wait for CLK_period/2;
                CLK <= '1';
                wait for CLK_period/2;
end process;


-- Stimulus process
stim_proc: process
begin
   -- hold reset state for 100 ns.
   wait for 100 ns;

   wait for CLK_period;
```

```
A<="00000000000000000000000000001010";
B<="00000000000000000000000000010010";
Set<="0000";
wait for CLK_period;
A<="00000000000000000000000000001010";
B<="00000000000000000000000000010010";
Set<="0001";
wait for CLK_period;
A<="00000000000000000000000000001010";
B<="00000000000000000000000000010010";
Set<="0010";
wait for CLK_period;
A<="00000000000000000000000000001010";
B<="00000000000000000000000000010010";
Set<="0011";
wait for CLK_period;
A<="00000000000000000000000000001010";
B<="00000000000000000000000000010010";
Set<="0100";
wait for CLK_period;
A<="00000000000000000000000000001010";
B<="00000000000000000000000000010010";
Set<="0101";
wait for CLK_period;
A<="00000000000000000000000000001010";
B<="00000000000000000000000000010010";
Set<="0110";
wait for CLK_period;
```

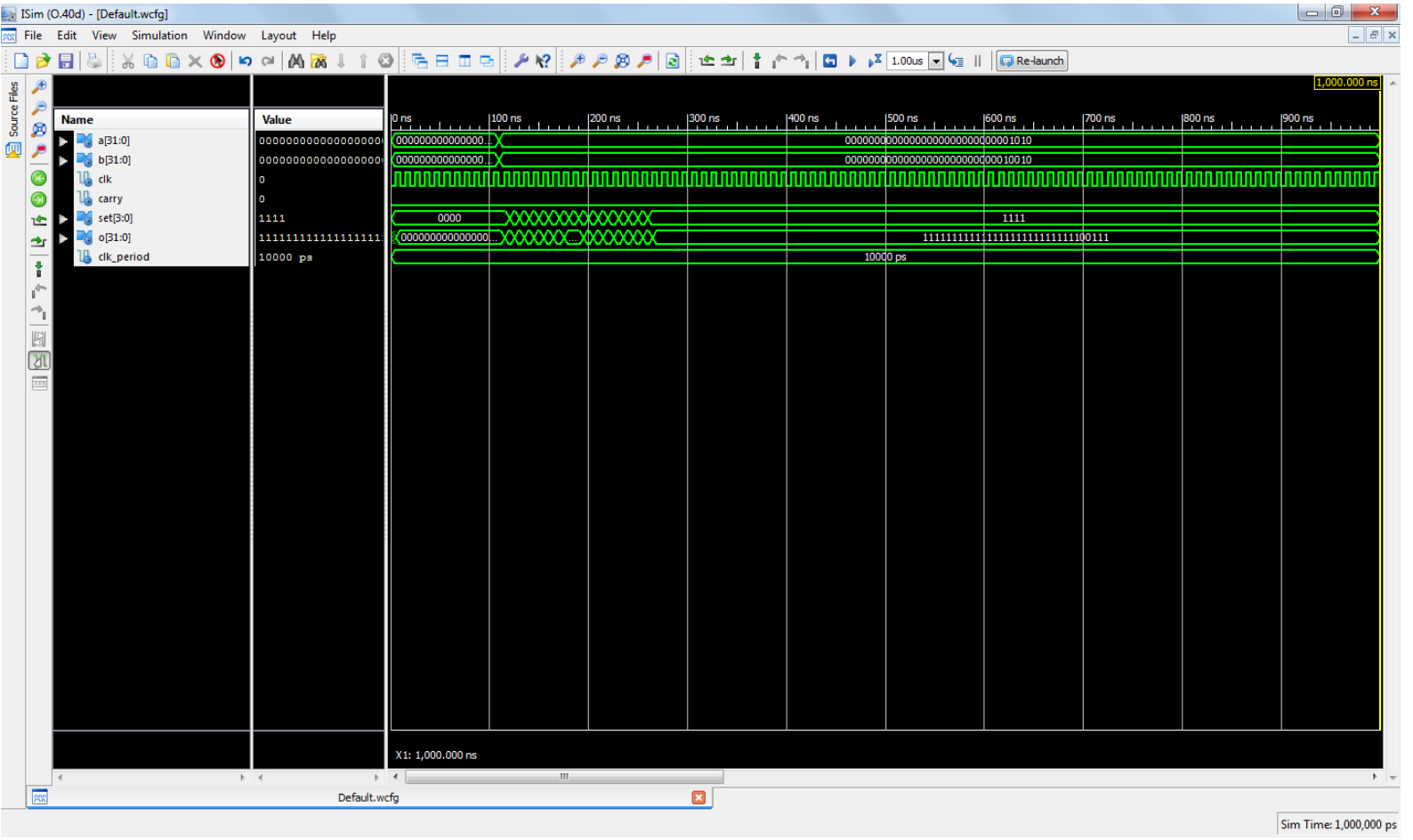
```
A<="00000000000000000000000000001010";
B<="00000000000000000000000000010010";
Set<="0111";
wait for CLK_period;
A<="00000000000000000000000000001010";
B<="00000000000000000000000000010010";
Set<="1000";
wait for CLK_period;
A<="00000000000000000000000000001010";
B<="00000000000000000000000000010010";
Set<="1001";
wait for CLK_period;
A<="00000000000000000000000000001010";
B<="00000000000000000000000000010010";
Set<="1010";
wait for CLK_period;
A<="00000000000000000000000000001010";
B<="00000000000000000000000000010010";
Set<="1011";
wait for CLK_period;
A<="00000000000000000000000000001010";
B<="00000000000000000000000000010010";
Set<="1100";
wait for CLK_period;
A<="00000000000000000000000000001010";
B<="00000000000000000000000000010010";
Set<="1101";
wait for CLK_period;
```

```
        A<="00000000000000000000000000001010";
        B<="00000000000000000000000000010010";
        Set<="1110";
        wait for CLK_period;
        A<="00000000000000000000000000001010";
        B<="00000000000000000000000000010010";
        Set<="1111";
    -- insert stimulus here

     wait;
  end process;

END;
```

RTL Schematic –



Simulation –



==========

* Design Summary *

==========

Top Level Output File Name : ALU_32bit.ngc

Primitive and Black Box Usage:

BELS : 796

GND : 1

INV : 64

LUT1 : 95

LUT2 : 33

LUT4 : 64

LUT6 : 64

MUXCY : 186

MUXF7 : 64

MUXF8 : 32

VCC : 1

XORCY : 192

FlipFlops/Latches : 32

FD : 32

Clock Buffers : 1

BUFGP : 1

IO Buffers : 101

IBUF : 69

OBUF : 32

Device utilization summary:

Selected Device : 6slx4tqg144-3

Slice Logic Utilization:

Number of Slice LUTs: 320 out of 2400 13%

Number used as Logic: 320 out of 2400 13%

Slice Logic Distribution:

Number of LUT Flip Flop pairs used: 320

Number with an unused Flip Flop: 320 out of 320 100%

Number with an unused LUT: 0 out of 320 0%

Number of fully used LUT-FF pairs: 0 out of 320 0%

Number of unique control sets: 1

IO Utilization:

Number of IOs: 102

Number of bonded IOBs: 102 out of 102 100%

IOB Flip Flops/Latches: 32

Specific Feature Utilization:

Number of BUFG/BUFGCTRLs: 1 out of 16 6%

16 bit ALU –

VHDL MODULE:

MUX:

```
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 primitives in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity mux is
   Port ( i0 : in  STD_LOGIC_VECTOR (15 downto 0);
          i1 : in  STD_LOGIC_VECTOR (15 downto 0);
          y : out  STD_LOGIC_VECTOR (15 downto 0);
          sel : in  STD_LOGIC_VECTOR (3 downto 0));
end mux;

architecture Behavioral of mux is

begin
y<= i0 when sel(3) = '0' else
i1 when sel(3) = '1' ;

end Behavioral;
```

Logic Unit:

```
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 primitives in this code.
--library UNISIM;
```

```
--use UNISIM.VComponents.all;

entity logic_unit is
   Port ( sel : in  STD_LOGIC_VECTOR (3 downto 0);
          a : in  STD_LOGIC_VECTOR (15 downto 0);
          b : in  STD_LOGIC_VECTOR (15 downto 0);
          o : out  STD_LOGIC_VECTOR (15 downto 0));
end logic_unit;

architecture Behavioral of logic_unit is

begin
o<= not a when sel= "1000" else
not b when sel= "1001" else
a and b when sel="1010" else
a or b when sel="1011" else
a nand b when sel="1100" else
a nor b when sel="1101" else
a xor b when sel="1110" else
a xnor b when sel="1111";

end Behavioral;
```

Arithmetic Unit:

```
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 primitives in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity arithmetic is
   Port ( a : in  STD_LOGIC_VECTOR (15 downto 0);
          b : in  STD_LOGIC_VECTOR (15 downto 0);
          sel : in  STD_LOGIC_VECTOR (3 downto 0);
          o : out  STD_LOGIC_VECTOR (15 downto 0);
  cin: in STD_LOGIC);
end arithmetic;

architecture Behavioral of arithmetic is
```

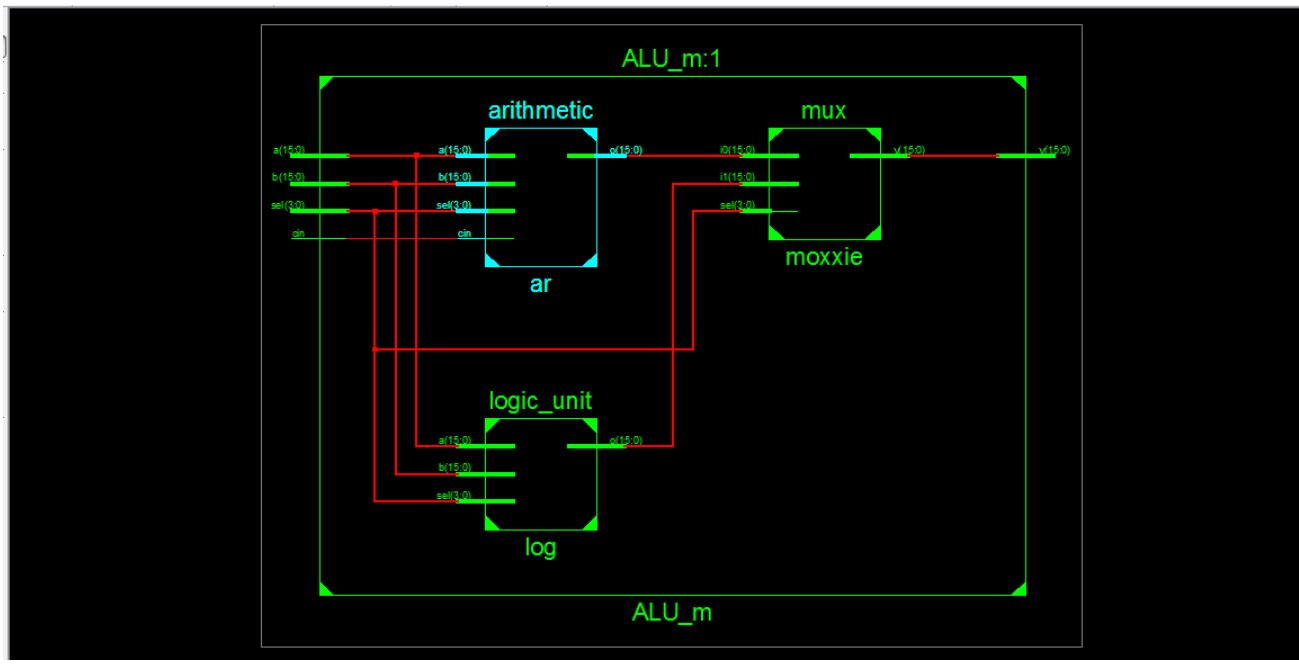
```
begin
o<= a when sel= "0000" else
a+1 when sel= "0001" else
a-1 when sel="0010" else
b when sel="0011" else
b+1 when sel="0100" else
b-1 when sel="0101" else
a+b when sel="0110" else
a + b + cin when sel="0111";

end Behavioral;
```

ALU:

```
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 primitives in this code.
--library UNISIM;
--use UNISIM.VComponents.all;

entity ALU_m is
   Port ( a : in  STD_LOGIC_VECTOR (15 downto 0);
          b : in  STD_LOGIC_VECTOR (15 downto 0);
          sel : in  STD_LOGIC_VECTOR (3 downto 0);
          cin : in  STD_LOGIC;
          y : out  STD_LOGIC_VECTOR (15 downto 0));
end ALU_m;

architecture Behavioral of ALU_m is
signal o1 : std_logic_vector(15 downto 0);
signal o2 : std_logic_vector(15 downto 0);
begin
ar : entity work.arithmetic port map(
a=>a,
b=>b,
sel=> sel,
cin=>cin,
o=>o1);
log : entity work.logic_unit port map(
a=>a,
b=>b,
```

```
sel=>sel,
--cin=>cin,
o=>o2);
moxxie : entity work.mux port map(
i0=> o1,
i1=> o2,
sel=>sel,
y=>y);
end Behavioral;
```

RTL Schematic –



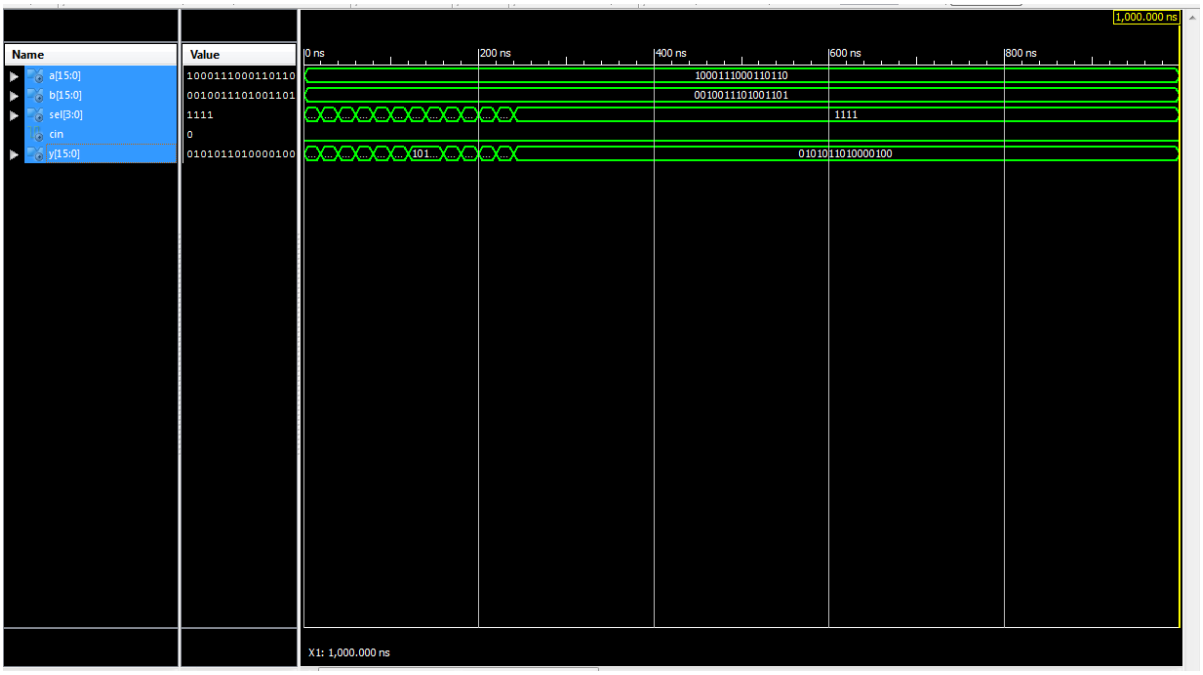
TEST BENCH :

```
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;

ENTITY test IS
END test;

ARCHITECTURE behavior OF test IS
```

```
   -- Component Declaration for the Unit Under Test (UUT)

   COMPONENT ALU_m
   PORT(
        a : IN  std_logic_vector(15 downto 0);
        b : IN  std_logic_vector(15 downto 0);
        sel : IN  std_logic_vector(3 downto 0);
        cin : IN  std_logic;
        y : OUT  std_logic_vector(15 downto 0)
       );
   END COMPONENT;


  --Inputs
  signal a : std_logic_vector(15 downto 0) := (others => '0');
  signal b : std_logic_vector(15 downto 0) := (others => '0');
  signal sel : std_logic_vector(3 downto 0) := (others => '0');
  signal cin : std_logic := '0';

 --Outputs
  signal y : std_logic_vector(15 downto 0);
  -- No clocks detected in port list. Replace <clock> below with
  -- appropriate port name

  --constant <clock>_period : time := 10 ns;

BEGIN

-- Instantiate the Unit Under Test (UUT)
  uut: ALU_m PORT MAP (
         a => a,
         b => b,
         sel => sel,
         cin => cin,
         y => y
        );

  -- Clock process definitions
  --<clock>_process :process
  --begin
--<clock><= '0';
--wait for <clock>_period/2;
--<clock><= '1';
--wait for <clock>_period/2;
  --end process;
```

```
  -- Stimulus process
  stim_proc: process
  begin
    -- hold reset state for 100 ns.
    --wait for 100 ns;

    --wait for <clock>_period*10;

    -- insert stimulus here
a<="1000111000110110";
b<="0010011101001101";
cin<='0';
sel<="0000";
wait for 20 ns;
sel<="0001";
wait for 20 ns;
sel<="0010";
wait for 20 ns;
sel<="0011";
wait for 20 ns;
sel<="0100";
wait for 20 ns;
sel<="0101";
wait for 20 ns;
sel<="0110";
wait for 20 ns;
    sel<="0111";
wait for 20 ns;
sel<="1000";
wait for 20 ns;
sel<="1001";
wait for 20 ns;
sel<="1010";
wait for 20 ns;
    sel<="1011";
wait for 20 ns;
sel<="1111";
wait for 20 ns;
    wait;
  end process;

END;
```

Simulation –


Name
Value
a[15:0]
1000111000110110
b[15:0]
0010011101001101
sel[3:0]
1111
cin
0
y[15:0]
0101011010000100
0 ns
200 ns
400 ns
600 ns
800 ns
1,000.000 ns
1000111000110110
0010011101001101
1111
101...
0101011010000100
X1: 1,000.000 ns