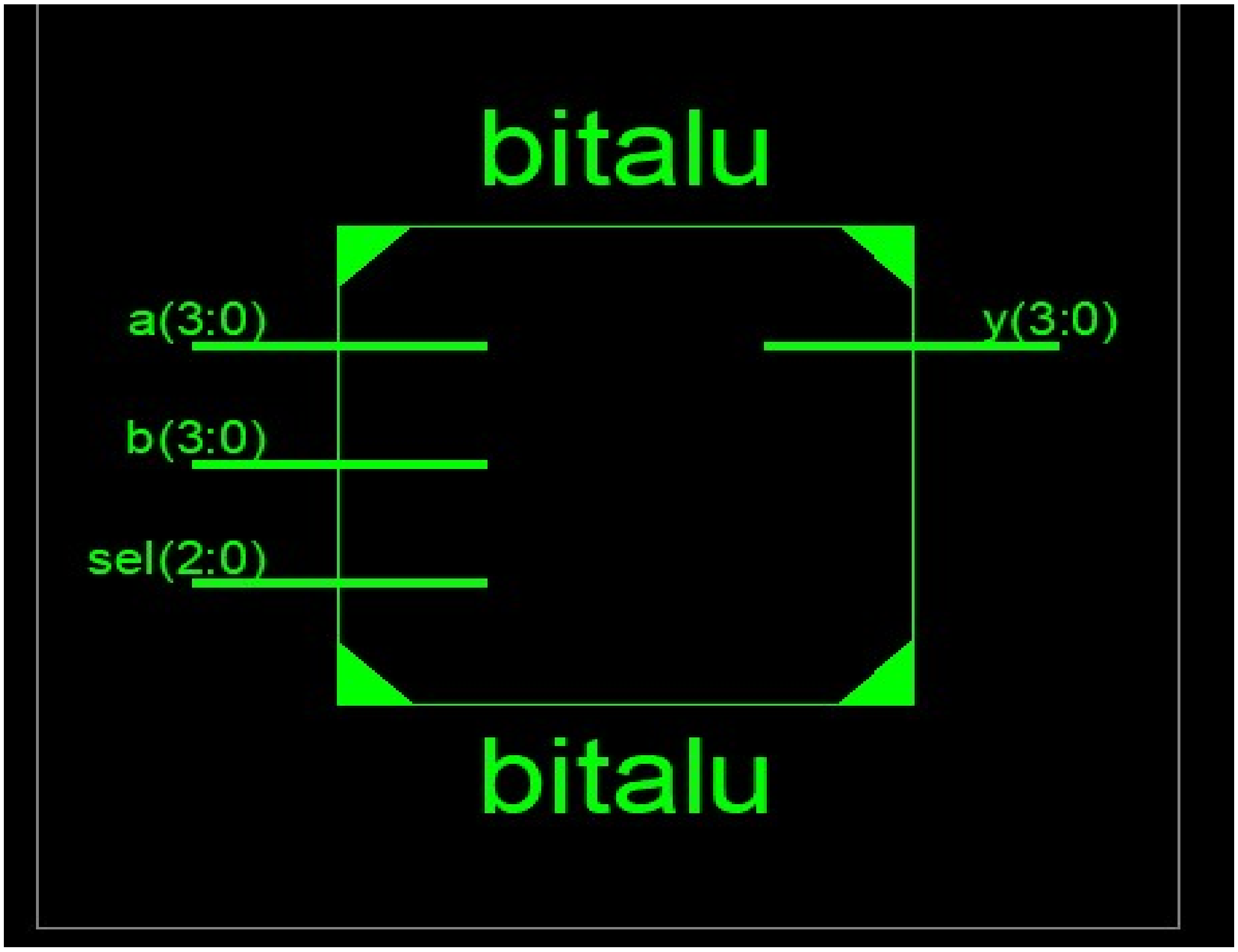
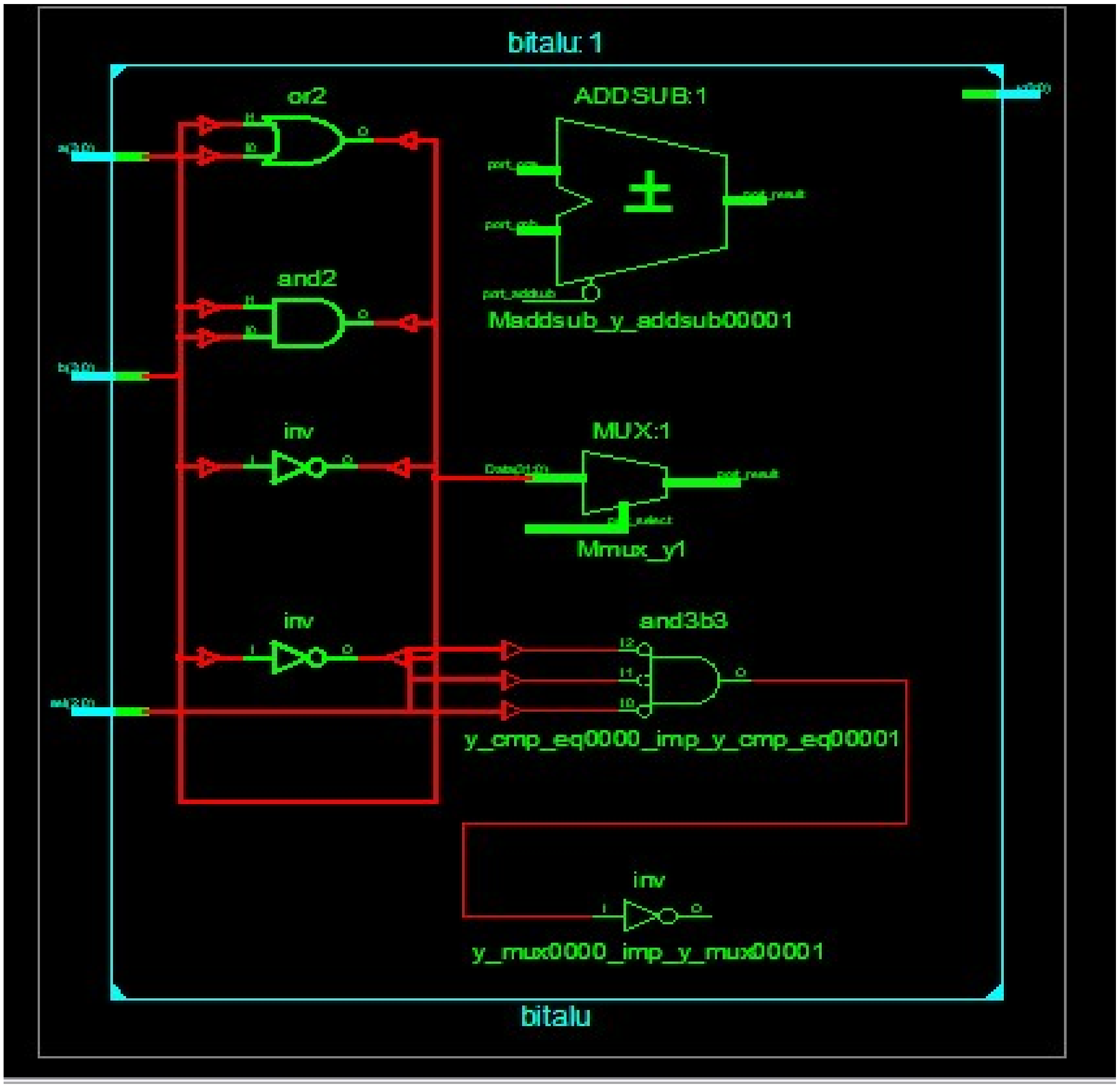
Practical No -: 1

4 BIT ALU

```
library IEEE;

use IEEE.STD_LOGIC_1164.ALL;

use IEEE.STD_LOGIC_UNSIGNED.ALL;

use IEEE.STD_LOGIC_ARITH.ALL;


entity bitalu is

        Port ( a : in     STD_LOGIC_VECTOR (3 downto 0);

                      b : in     STD_LOGIC_VECTOR (3 downto 0);

                      sel : in     STD_LOGIC_VECTOR (2 downto 0);

                      y : out     STD_LOGIC_VECTOR (3 downto 0));

end bitalu;


architecture bitalu_arch of bitalu is

begin

process(a,b,sel)

begin

case sel is

when"000"=>y<=a+b;

when"001"=>y<=a-b;

when"010"=>y<=a and b;

when"011"=>y<=a or b;

when"100"=>y<=a nand b;

when"101"=>y<=a nor b;
```

```
when"110"=>y<=a;

when"111"=>y<=b;

when others=>null;

end case;

end process;

end bitalu_arch ;
```





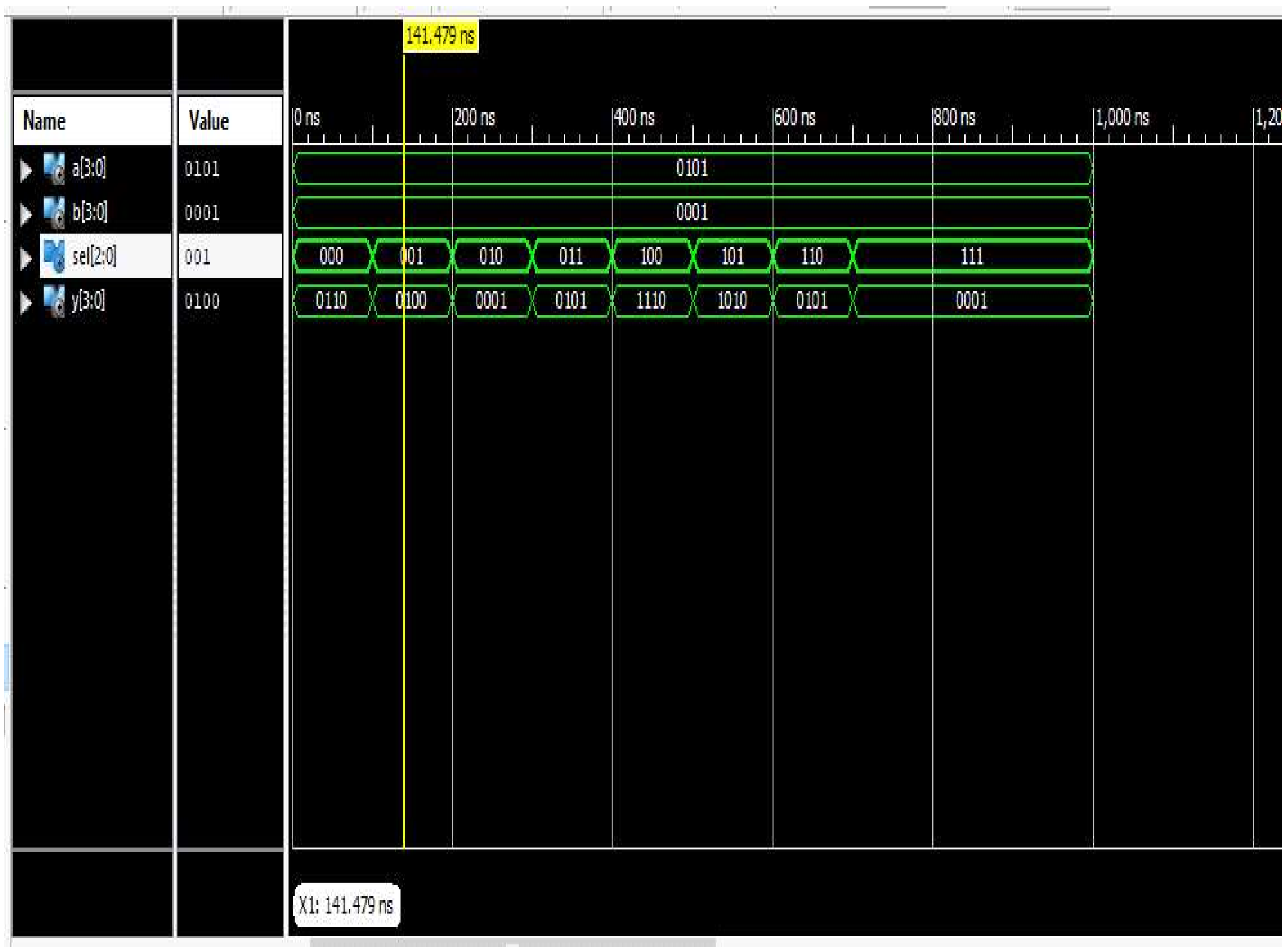
TESTBENCH -:

```
LIBRARY ieee;

USE ieee.std_logic_1164.ALL;

ENTITY bbbb IS

END bbbb;
```

```
ARCHITECTURE behavior OF bbbb IS

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

        COMPONENT bitalu

        PORT(

                 a : IN     std_logic_vector(3 downto 0);

                 b : IN     std_logic_vector(3 downto 0);

                 sel : IN     std_logic_vector(2 downto 0);

                 y : OUT     std_logic_vector(3 downto 0));

        END COMPONENT;

      --Inputs

      signal a : std_logic_vector(3 downto 0) := (others => '0');

      signal b : std_logic_vector(3 downto 0) := (others => '0');

      signal sel : std_logic_vector(2 downto 0) := (others => '0');

         --Outputs

      signal y : std_logic_vector(3 downto 0);


BEGIN

         -- Instantiate the Unit Under Test (UUT)

      uut: bitalu PORT MAP (

                   a => a,

                   b => b,

                   sel => sel,

                   y => y);

      -- Stimulus process

      stim_proc: process
```

```
        begin
a<="0101";     b<="0001";     sel<="000";
-- hold reset state for 100 ns.
wait for 100 ns;

a<="0101";     b<="0001";     sel<="001";
-- hold reset state for 100 ns.
wait for 100 ns;

a<="0101";     b<="0001";     sel<="010";
-- hold reset state for 100 ns.
wait for 100 ns;

a<="0101";     b<="0001";     sel<="011";
-- hold reset state for 100 ns.
wait for 100 ns;

a<="0101";     b<="0001";     sel<="100";
-- hold reset state for 100 ns.
wait for 100 ns;

a<="0101";     b<="0001";sel<="101";
-- hold reset state for 100 ns.
wait for 100 ns;
```

```
a<="0101";    b<="0001";    sel<="110";

-- hold reset state for 100 ns.

wait for 100 ns;


a<="0101";    b<="0001";    sel<="111";

-- hold reset state for 100 ns..

wait for 100 ns;

            wait;

      end process;

END;
```


141.479 ns
Name
Value
a[3:0]
b[3:0]
sel[2:0]
y[3:0]
0101
0001
001
0100
0 ns
200 ns
400 ns
600 ns
800 ns
1,000 ns
1,20
0101
0001
000
001
010
011
100
101
110
111
0110
0100
0001
0101
1110
1010
0101
0001
X1: 141.479 ns