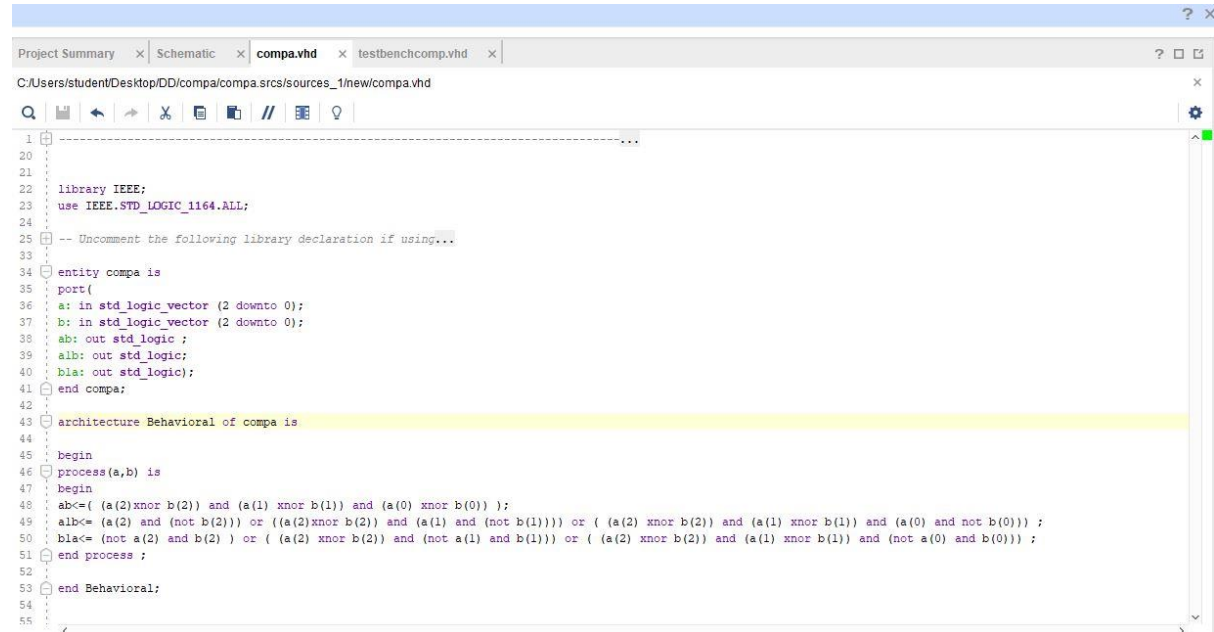


Experiment 7

Implementing 3-bit comparator

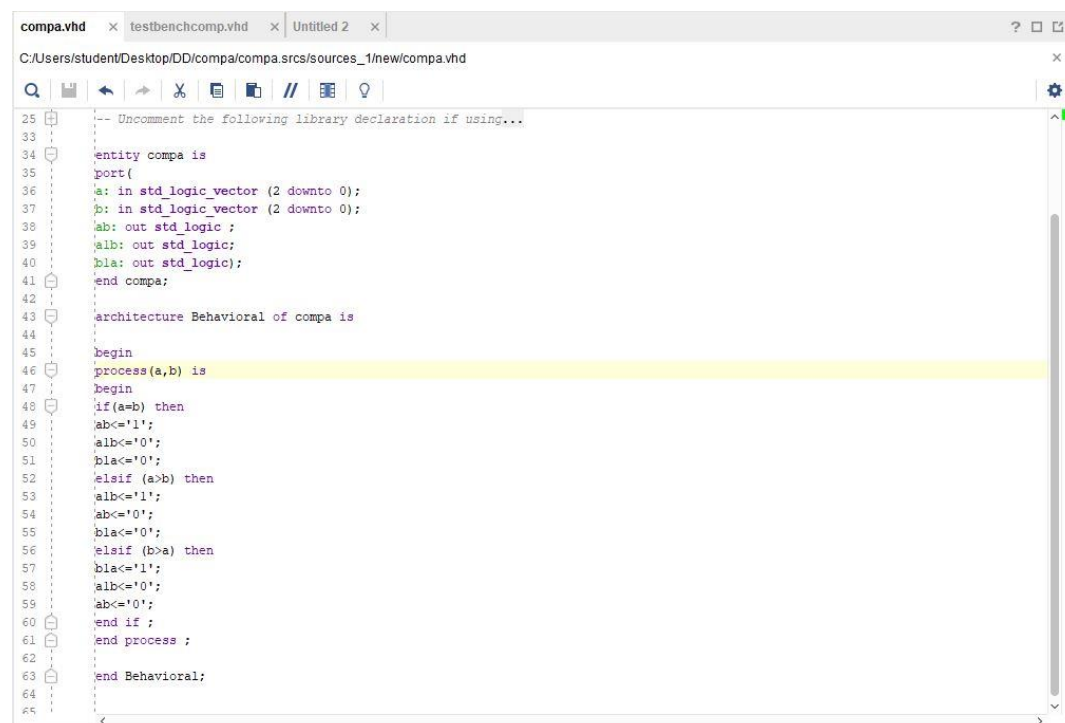
Design code: Structural



The screenshot shows a VHDL editor window with the file name `compa.vhd`. The code is structural, defining an entity `compa` with two 3-bit input ports `a` and `b`, and three output ports `ab`, `alb`, and `bla`. The architecture `Behavioral of compa is` contains a process `process(a,b) is` that implements the comparator logic using logical operations like `xnor`, `and`, `or`, and `not`.

```
1 20
21
22 library IEEE;
23 use IEEE.STD_LOGIC_1164.ALL;
24
25 -- Uncomment the following library declaration if using...
33
34 entity compa is
35 port(
36   a: in std_logic_vector (2 downto 0);
37   b: in std_logic_vector (2 downto 0);
38   ab: out std_logic ;
39   alb: out std_logic;
40   bla: out std_logic);
41 end compa;
42
43 architecture Behavioral of compa is
44
45 begin
46   process(a,b) is
47   begin
48     ab<= (a(2)xnor b(2)) and (a(1) xnor b(1)) and (a(0) xnor b(0)) ;
49     alb<= (a(2) and (not b(2))) or ((a(2)xnor b(2)) and (a(1) and (not b(1)))) or ((a(2) xnor b(2)) and (a(1) xnor b(1)) and (a(0) and not b(0))) ;
50     bla<= (not a(2) and b(2)) or ((a(2) xnor b(2)) and (not a(1) and b(1))) or ((a(2) xnor b(2)) and (a(1) xnor b(1)) and (not a(0) and b(0))) ;
51   end process ;
52
53 end Behavioral;
54
55
```

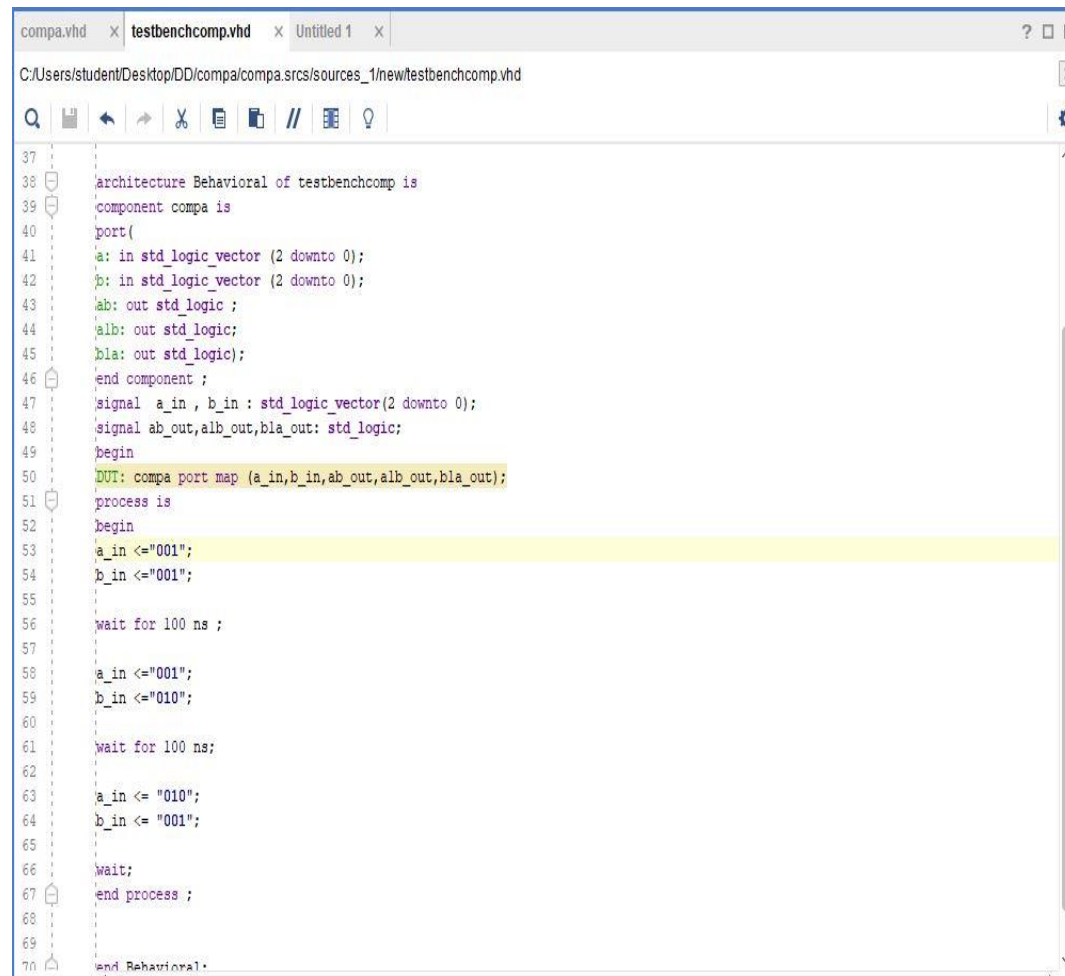
Design code: behavioral



The screenshot shows a VHDL editor window with the file name `compa.vhd`. The code is behavioral, defining an entity `compa` with two 3-bit input ports `a` and `b`, and three output ports `ab`, `alb`, and `bla`. The architecture `Behavioral of compa is` contains a process `process(a,b) is` that implements the comparator logic using conditional statements (`if`, `elsif`, `else`) to compare the bits of `a` and `b`.

```
25 -- Uncomment the following library declaration if using...
33
34 entity compa is
35 port(
36   a: in std_logic_vector (2 downto 0);
37   b: in std_logic_vector (2 downto 0);
38   ab: out std_logic ;
39   alb: out std_logic;
40   bla: out std_logic);
41 end compa;
42
43 architecture Behavioral of compa is
44
45 begin
46   process(a,b) is
47   begin
48     if(a=b) then
49       ab<='1';
50       alb<='0';
51       bla<='0';
52     elsif (a>b) then
53       alb<='1';
54       ab<='0';
55       bla<='0';
56     elsif (b>a) then
57       bla<='1';
58       alb<='0';
59       ab<='0';
60     end if ;
61   end process ;
62
63 end Behavioral;
64
65
```

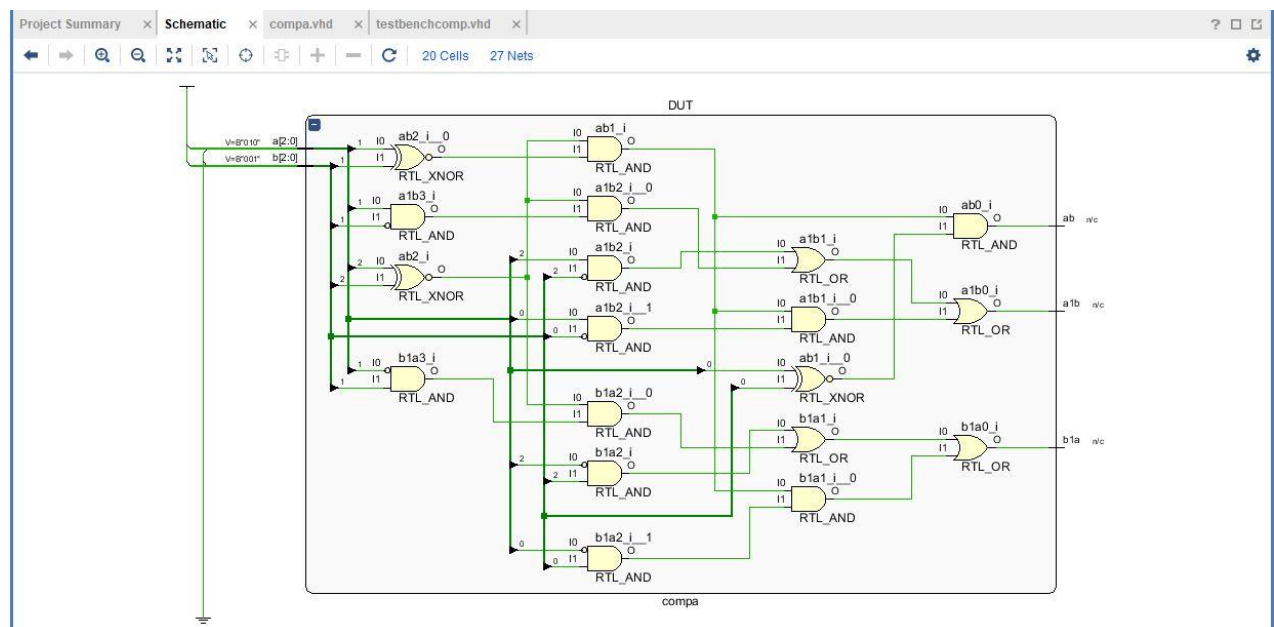
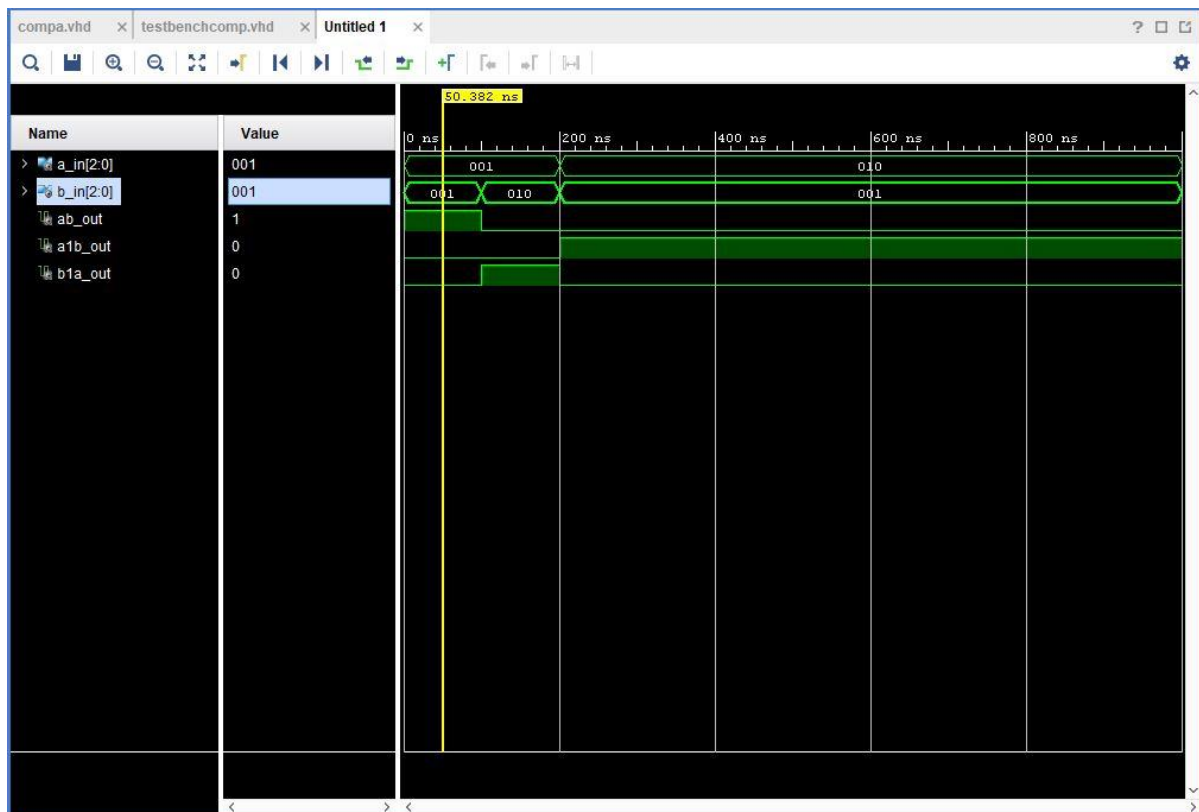
Testbench:



The image shows a screenshot of a VHDL testbench code in an IDE. The code is written in VHDL and is titled "testbenchcomp.vhd". The code defines a testbench for a component named "compa". The testbench architecture is "Behavioral of testbenchcomp is". It defines a component "compa" with two input ports, "a" and "b", both of type "std_logic_vector (2 downto 0)". It also defines three output ports, "ab", "alb", and "bla", all of type "std_logic". The testbench then defines a signal "a_in" and "b_in" of type "std_logic_vector(2 downto 0)". It also defines three signals, "ab_out", "alb_out", and "bla_out", of type "std_logic". The testbench then defines a process "process is" which starts by assigning "a_in" to "001" and "b_in" to "001". It then waits for 100 ns. After the wait, it assigns "a_in" to "001" and "b_in" to "010". It then waits for 100 ns. After the wait, it assigns "a_in" to "010" and "b_in" to "001". It then waits. Finally, it ends the process. The code is highlighted in yellow.

```
37
38 architecture Behavioral of testbenchcomp is
39 component compa is
40 port(
41   a: in std_logic_vector (2 downto 0);
42   b: in std_logic_vector (2 downto 0);
43   ab: out std_logic ;
44   alb: out std_logic;
45   bla: out std_logic);
46 end component ;
47 signal a_in , b_in : std_logic_vector(2 downto 0);
48 signal ab_out,alb_out,bla_out: std_logic;
49 begin
50   DUT: compa port map (a_in,b_in,ab_out,alb_out,bla_out);
51 process is
52 begin
53   a_in <="001";
54   b_in <="001";
55
56   wait for 100 ns ;
57
58   a_in <="001";
59   b_in <="010";
60
61   wait for 100 ns;
62
63   a_in <= "010";
64   b_in <= "001";
65
66   wait;
67 end process ;
68
69
70 end Behavioral;
```

Simulation:



Result:- 3-bit comparator implemented successfully, and their test results are shown above.