Name: Shubham Vilas Malve
Roll No.: E42067

```
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 LCD1 is

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

          Y : out  STD_LOGIC_VECTOR(6 downtO 0));

end LCD1;


architecture Behavioral of LCD1 is



begin

 process(A)

 begin

  if A="0000" then
```

```
Y<="1111110";

elsif A="0001" then
y<="0110000";

elsif A="0010" then
Y<="1101101";

elsif A="0011" then
Y<="1111001";

elsif A="0100" then
Y<="0110011";

elsif A="0101" then
Y<="1011011";

elsif A="0111" then
Y<="1011111";

elsif A="1000" then
Y<="1110000";

elsif A="1001" then
Y<="1111111";
```

```
 elsif A="1011" then
 Y<="1111011";

 else
 Y<="0000000";
 end if;
end process;

end Behavioral;
```

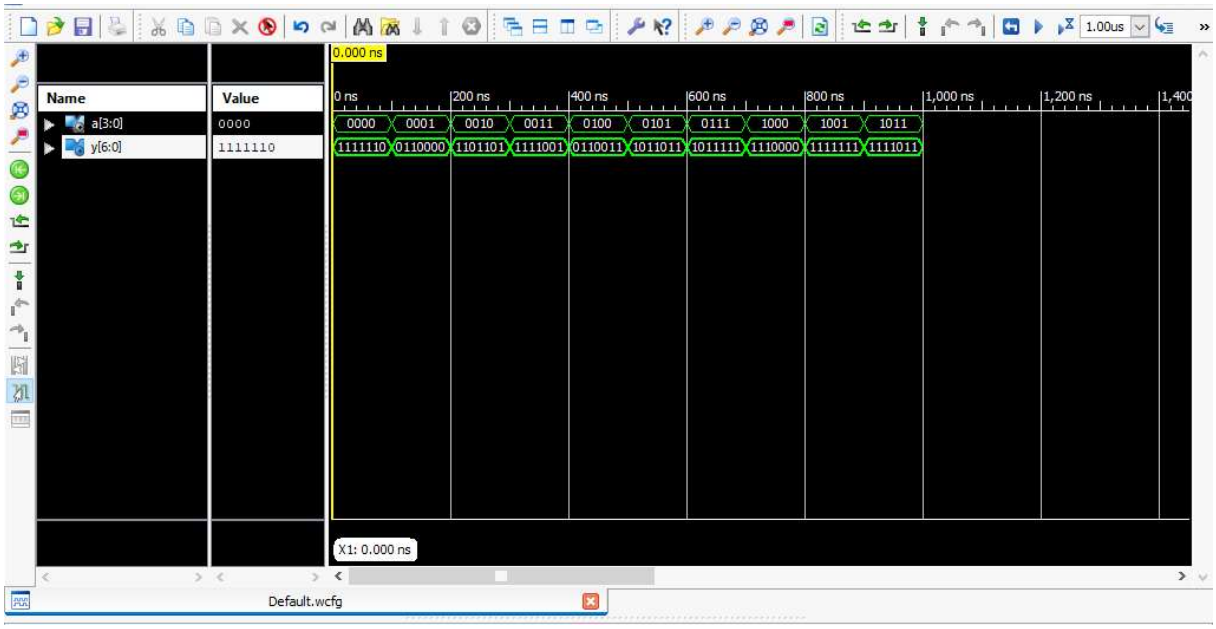
TEST BENCH :

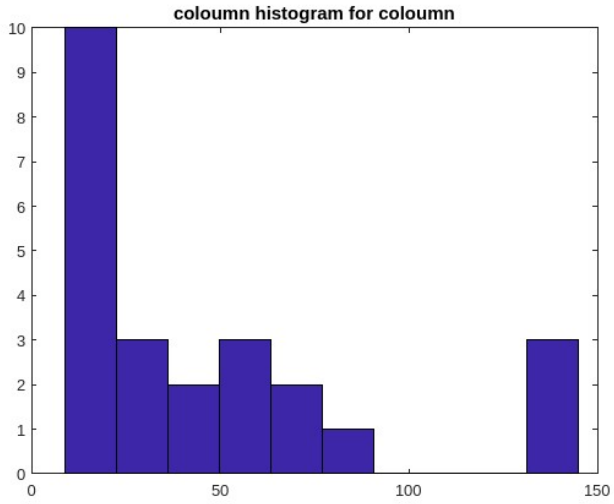
```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;

-- Uncomment the following library declaration if using
-- arithmetic functions with Signed or Unsigned values
--USE ieee.numeric_std.ALL;

ENTITY segment IS
END segment;
```

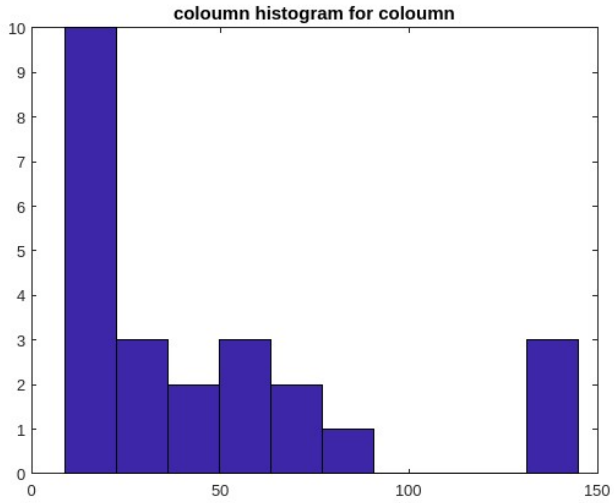
```
ARCHITECTURE behavior OF segment IS

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

   COMPONENT LCD1
   PORT(
        A : IN  std_logic_vector(3 downto 0);
        Y : OUT  std_logic_vector(6 downto 0)
       );
   END COMPONENT;



  --Inputs
  signal A : std_logic_vector(3 downto 0) := (others => '0');

            --Outputs
  signal Y : std_logic_vector(6 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: LCD1 PORT MAP (
```

```
        A => A,
        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.
                    A<="0000";
   wait for 100 ns;
                    A<="0001";
                    wait for 100 ns;
                    A<="0010";
                    wait for 100 ns;
                    A<="0011";
                    wait for 100 ns;
```

```
                A<="0100";
                wait for 100 ns;
                A<="0101";
                wait for 100 ns;
                A<="0111";
                wait for 100 ns;
                A<="1000";
                wait for 100 ns;
                A<="1001";
                wait for 100 ns;
                A<="1011";
                wait for 100 ns;

    --wait for <clock>_period*10;

    -- insert stimulus here

  end process;

END;
```


Name
Value
a[3:0]
y[6:0]
0000
1111110
0.000 ns
0 ns
200 ns
400 ns
600 ns
800 ns
1,000 ns
1,200 ns
0000 0001 0010 0011 0100 0101 0111 1000 1001 1011
1111110 0110000 1101101 1111001 0110011 1011011 1011111 1110000 1111111 1111011
X1: 0.000 ns
Default.wcfg


coloumn histogram for coloumn
0
1
2
3
4
5
6
7
8
9
10
0
50
100
150


coloumn histogram for coloumn
0
1
2
3
4
5
6
7
8
9
10
0
50
100
150