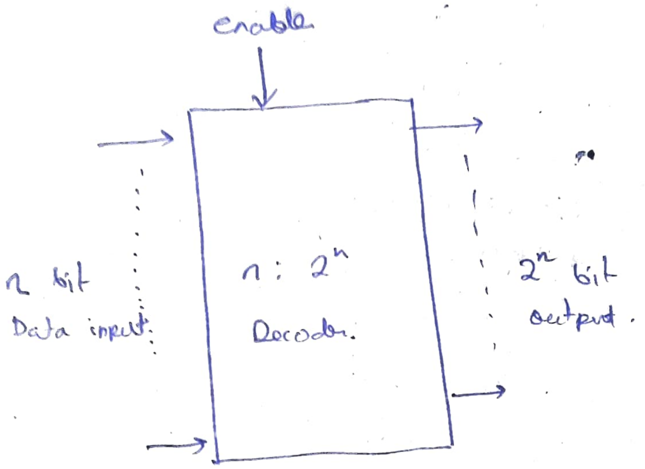
Question :- Design a 4x16 decoder by behavioural modelling, along with test bench.

Block Diagram :-



A decoder is a combinational circuit that has 'n' input lines and a maximum of 2^n output lines.

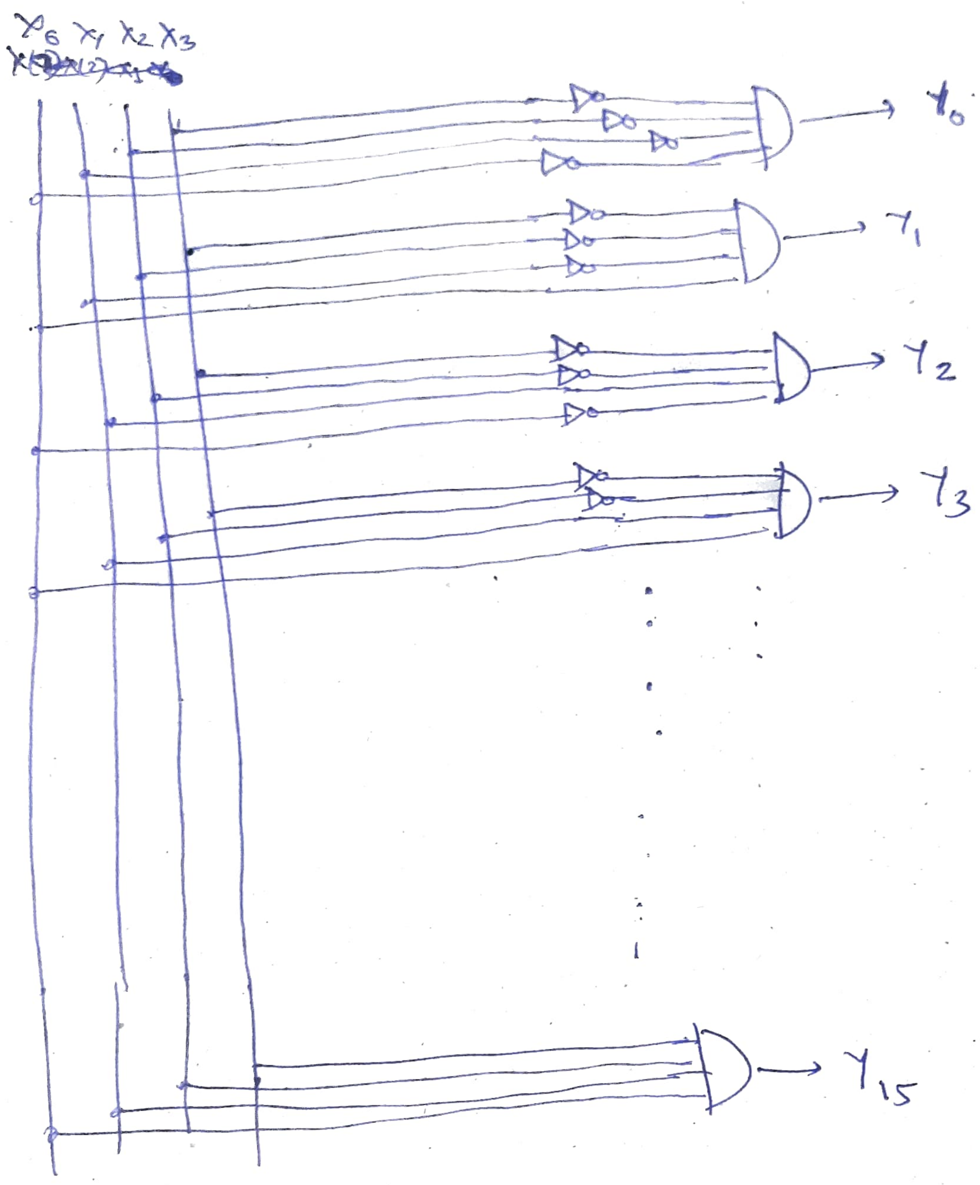
Only one of the output lines is high, based on the input when the decoder is enabled.

If the decoder is disabled ~~&~~ all the output lines are Low.

Truth table :-

X	Y
0 0 0 0	0000.0000.0000.0001
0 0 0 1	0000000000000010
0 0 1 0	0000000000000100
0 0 1 1	0000000000001000
0 1 0 0	0000000000010000
0 1 0 1	0000 0000 0010 0000
0 1 1 0	0000 0000 0100 0000
0 1 1 1	0000 0000 1000 0000
1 0 0 0	0000 0001 0000 0000
1 0 0 1	0000 0010 0000 0000
1 0 1 0	0000 0100 0000 0000
1 0 1 1	0000 1000 0000 0000
1 1 0 0	0001 0000 0000 0000
1 1 0 1	0010 0000 0000 0000
1 1 1 0	0100 0000 0000 0000
1 1 1 1	1000 0000 0000 0000

Circuit diagram for ~~4 bit~~ 4x16 decoder.



code :-

```
entity decoder_5x16 is
    Port ( x: in std_logic_vector (3 downto 0);
           E: in std_logic;
           Y: out std_logic_vector (15 downto 0));

end decoder_4x16;

architecture Behavioural of ~~de~~ decoder_5x16 is
begin
    p1: process (x,E)
    begin
        if E = '0' then
            Y = "0000 0000 0000 0000";
        else
            if x="0000" then ~~0000 0000 0000~~ Y <= "0000 0000 0000 0001";
        elsif x ="0001" then Y <= "0000 0000 0000 0010";
        elsif x ="0010" then Y <= "0000 0000 0000 0100";
        elsif x ="0011" then Y <= "0000 0000 0000 1000";
        elsif x ="0100" then Y <= "0000 0000 0001 0000";
        elsif x ="0101" then Y <= "0000 0000 0010 0000";
        elsif x ="0110" then Y <= "0000 0000 0100 0000";
        elsif x ="0111" then Y <= "0000 0000 1000 0000";
        elsif
        elsif x ="1000" then Y <= "0000 0001 0000 0000";
              x ="1001" then Y <= "0000 0010 0000 0000";
        elsif x ="1010" then Y <= "0000 0100 0000 0000";
        elsif x ="1011" then Y <= "0000 1000 0000 0000";
        elsif
         elsif x ="1100" then Y <= "0001 0000 0000 0000";
          elsif x ="1101" then Y <= "0010 0000 0000 0000";
          elsif x ="1110" then Y <= "0100 0000 0000 0000";
          elsif x ="1111" then Y <= "1000 0000 0000 0000";
        end if;
    end if;
    end process;
end Behavioral.
```

Test Bench :-

```
stim_proc : process.
begin
    x <= "0000" ; wait for 1ps;
    x <= "0001" ; wait for 1ps;
    x <= "0010" ; wait for 1ps;
    x <= "0011" ; wait for 1ps;
    x <= "0100" ; wait for 1ps;
    x <= "0101" ; wait for 1ps;
    x <= "0110" ; wait for 1ps;
    x <= "0111" ; wait for 1ps;
    x <= "1000" ; wait for 1ps;
    x <= "1001" ; wait for 1ps;
    x <= "1010" ; wait for 1ps;
    x <= "1011" ; wait for 1ps;
    x <= "1100" ; wait for 1ps;
    x <= "1101" ; wait for 1ps;
    x <= "1110" ; wait for 1ps;
    x <= "1111" ; wait for 1ps;
end process;
```