ASSIGNMENT REPORT

VLSI DESIGN (2EC601CC24)

TOPIC – 4 to 16 Decoder Layout in Microwind (using Verilog)

COMPILED BY:-
HARSHVARDHAN SINGH (22BEC120)

4 to 16 Decoder Using Verilog

Harshvardhan Singh
Department of Electronics & Communication, Institute of Technology
Nirma University, Ahmedabad, Gujarat-382481, India
22bec120@nirmauni.ac.in

Introduction:-

A 4-to-16 decoder is a combinational digital circuit that decodes a 4-bit binary input into one of 16 unique output lines, each corresponding to one specific input combination. It effectively acts as a translator, taking a 4-bit binary number (ranging from 0000 to 1111) and activating a single output line while keeping all others inactive. This functionality is useful in digital systems where a specific action or device needs to be selected based on binary inputs, such as in memory address decoding, multiplexing, and data routing. The 4-to-16 decoder circuit uses AND gates and inverters to implement its logic, where each output represents a unique minterm of the 4-bit input.

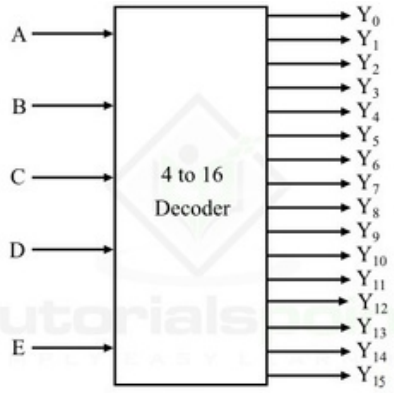
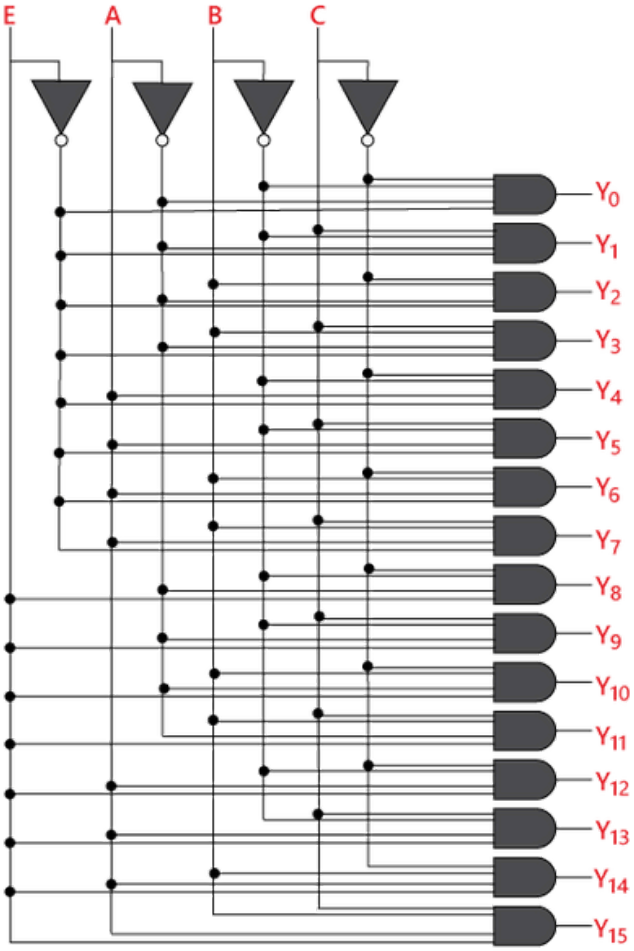


Figure 6 - 4 to 16 Decoder

Logic Diagram:-

The 4-to-16-line decoder can be constructed using either 2 to 4 decoder or 3 to 8 decoders. The logic diagram for the 4-to-16-line decoder is shown below. We have designed the 4 to 16 decoders using 4 not and 16 and gates.



Truth Table:-

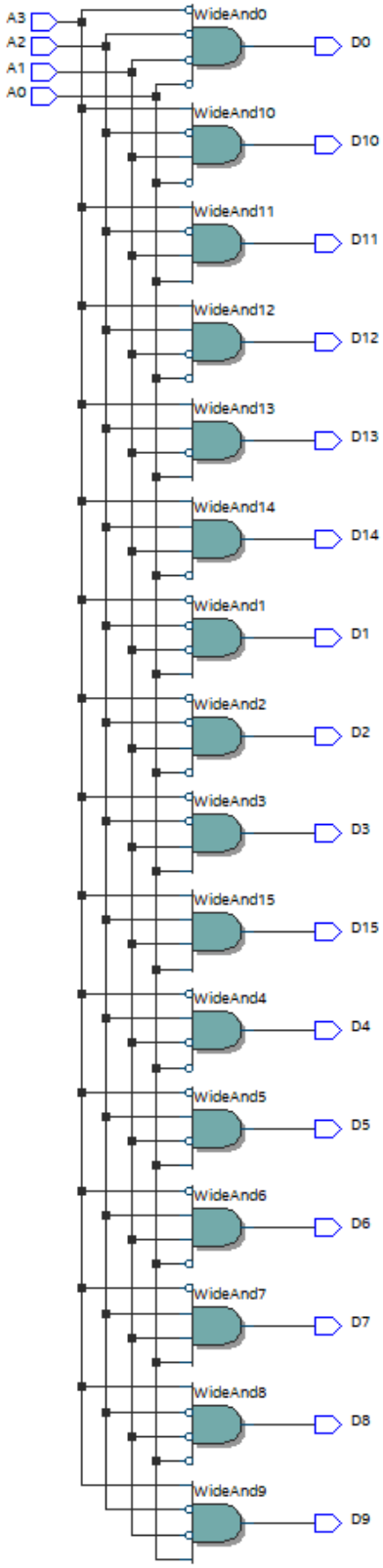
INPUTS				OUTPUTS															
A_3	A_2	A_1	A_0	Y_{15}	Y_{14}	Y_{13}	Y_{12}	Y_{11}	Y_{11}	Y_{10}	Y_9	Y_8	Y_7	Y_6	Y_5	Y_4	Y_3	Y_2	Y_1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
0	1	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
0	1	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
1	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
1	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

When all inputs are 0 the output of the decoder Y1 and when all inputs are 1 then the output is Y15. Decoder basically switches between 0 to 15 depending on the input. Now for implementing the decoder on the Microwind tool we write Verilog code on Quartus. The Verilog code for the 4-to-16-line decoder is given below.

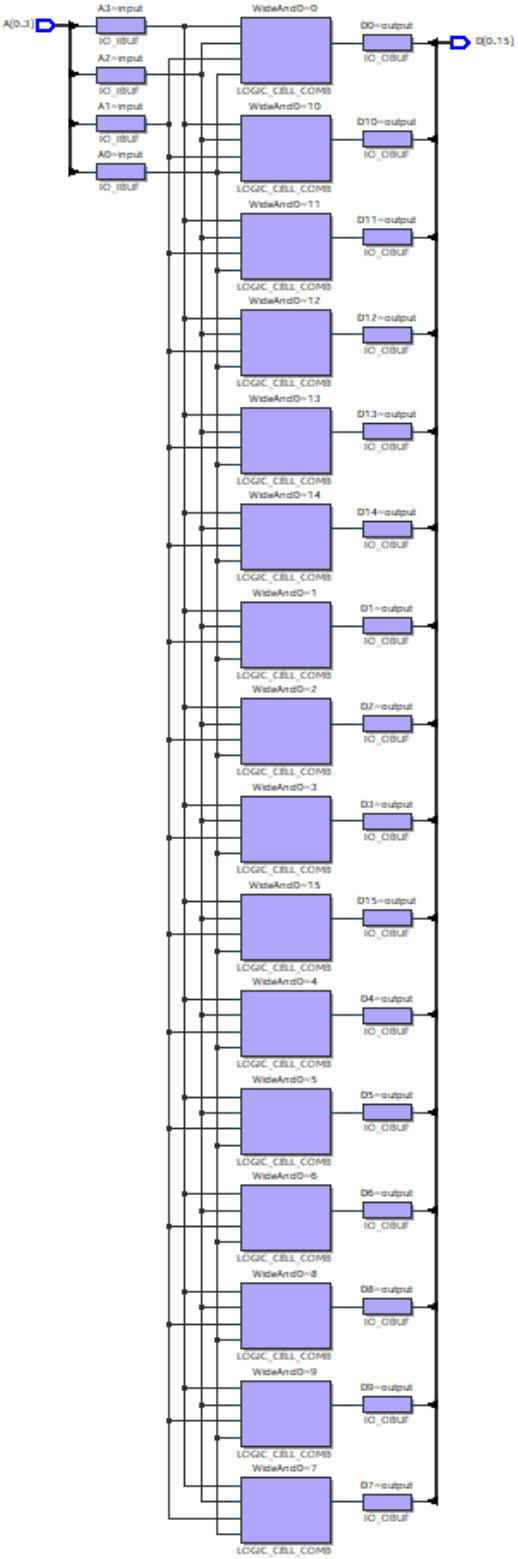
Verilog Code:-

```
module decoder_4_to_16 (
   input A3, A2, A1, A0,   // 4-bit input
   output D0, D1, D2, D3, D4, D5, D6, D7, D8, D9, D10, D11, D12, D13, D14, D15 // 16 outputs);

   wire nA3, nA2, nA1, nA0; // Not gates for each input
   // Inverting each input
   not (nA3, A3);
   not (nA2, A2);
   not (nA1, A1);
   not (nA0, A0);

   // AND gates for each output
   and (D0, nA3, nA2, nA1, nA0);  // 0000
   and (D1, nA3, nA2, nA1, A0);   // 0001
   and (D2, nA3, nA2, A1, nA0);   // 0010
   and (D3, nA3, nA2, A1, A0);    // 0011
   and (D4, nA3, A2, nA1, nA0);   // 0100
   and (D5, nA3, A2, nA1, A0);    // 0101
   and (D6, nA3, A2, A1, nA0);    // 0110
   and (D7, nA3, A2, A1, A0);     // 0111
   and (D8, A3, nA2, nA1, nA0);   // 1000
   and (D9, A3, nA2, nA1, A0);    // 1001
   and (D10, A3, nA2, A1, nA0);   // 1010
   and (D11, A3, nA2, A1, A0);    // 1011
   and (D12, A3, A2, nA1, nA0);   // 1100
   and (D13, A3, A2, nA1, A0);    // 1101
   and (D14, A3, A2, A1, nA0);    // 1110
   and (D15, A3, A2, A1, A0);     // 1111

endmodule
```

RTL View :-



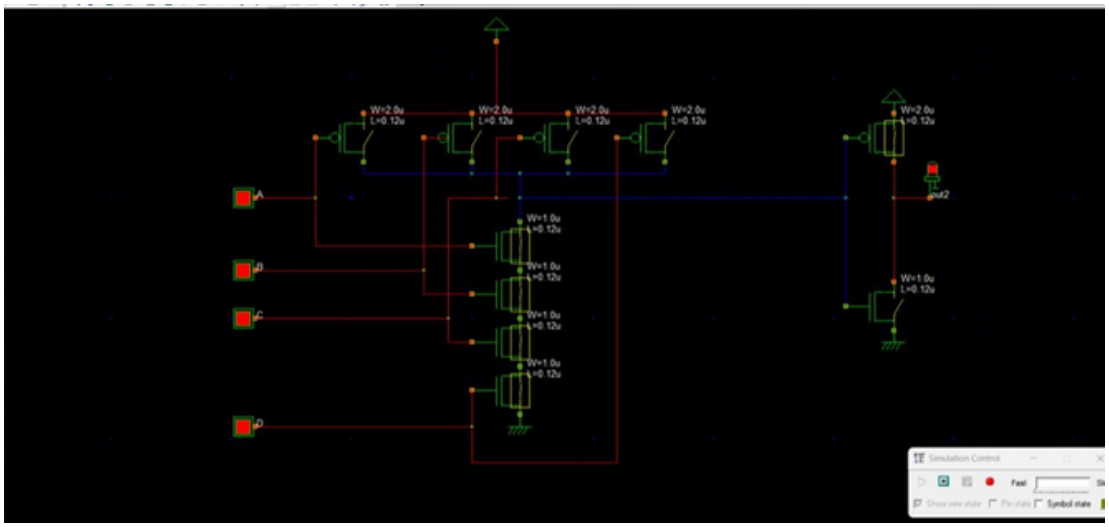
TTL View :-

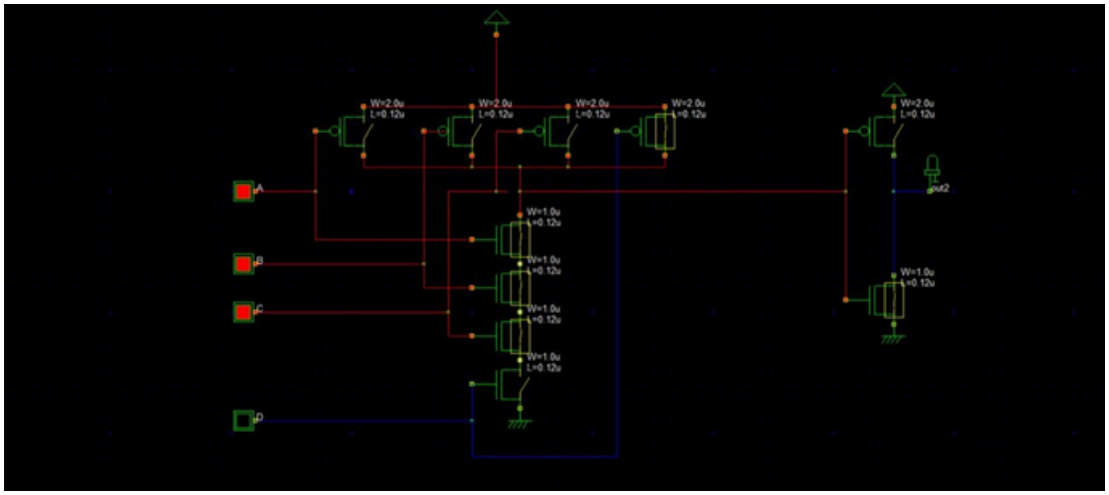


Optimized Boolean Equations:-

1. D0 = A3'A2'A1'A0'
2. D1 = A3'A2'A1'A0
3. D2 = A3'A2'A1A0'
4. D3 = A3'A2'A1A0
5. D4 = A3'A2A1'A0'
6. D5 = A3'A2A1'A0
7. D6 = A3'A2A1A0'
8. D7 = A3'A2A1A0
9. D8 = A3A2'A1'A0'
10. D9 = A3A2'A1'A0
11. D10 = A3A2'A1A0'
12. D11 = A3A2'A1A0
13. D12 = A3A2A1'A0'
14. D13 = A3A2A1'A0
15. D14 = A3A2A1A0'
16. D15 = A3A2A1A0

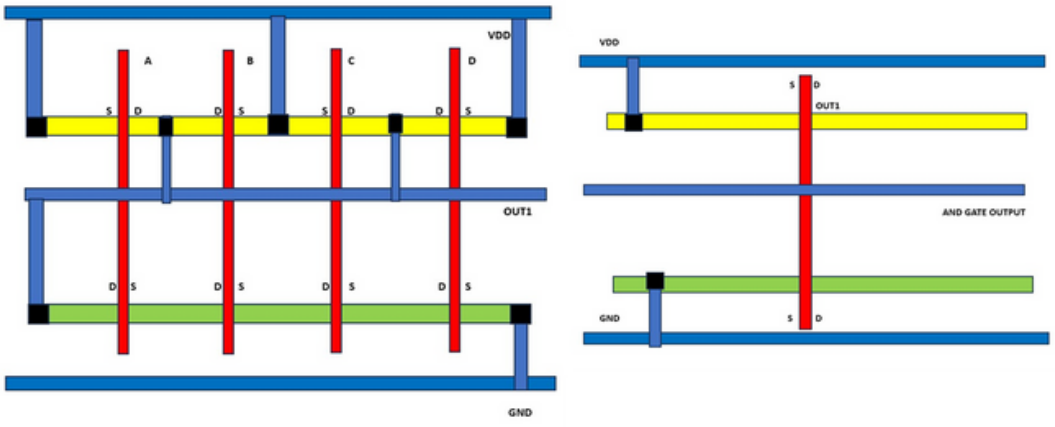
Transistor Level Schematic for CMOS/MOS implementation:-





As, I formed 4 input AND gate schematic, for first Boolean function. All the 16 are like the first one. This schematic is drawn or simulated on dsch03 software, as we can see that, if all buttons are pressed than led will glow, if single button is not pressed than led will not glow.

Stick Diagram for above implementation level using proper color code:-



Various levels of VOL corresponding to various transistor statuses:-

In a 4-to-16 decoder, the levels of VOL (Output Low Voltage) can vary depending on the transistor statuses, which are influenced by the logic levels of the input lines. VOL represents the maximum voltage at which an output is considered "low." Here, I'll provide an overview of the transistor statuses and the corresponding VOL levels for each output line based on the input conditions:

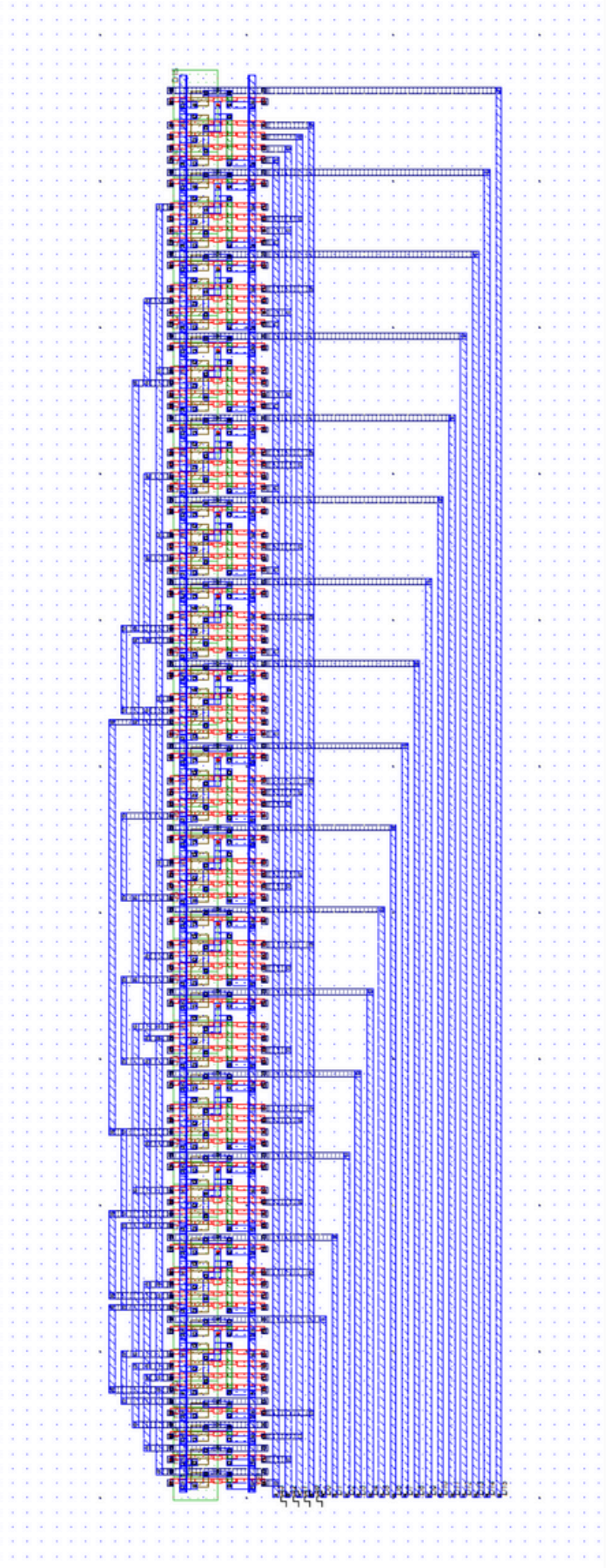
Transistor ON (Active): When a transistor is ON (active), it means that the input combination selects that output line. In this case, the output level is typically close to ground (0V) since the transistor actively connects the output to ground. Therefore, VOL for an active output is typically very low, approaching 0V.

Transistor OFF (Inactive): When a transistor is OFF (inactive), it means that the input combination does not select that output line. In this case, the output line remains high or at its inactive state. VOL for an inactive output is typically closer to the supply voltage (VCC) or the high logic level, depending on the technology used. It's important to note that the VOL for an inactive output is not zero; it's closer to VCC.

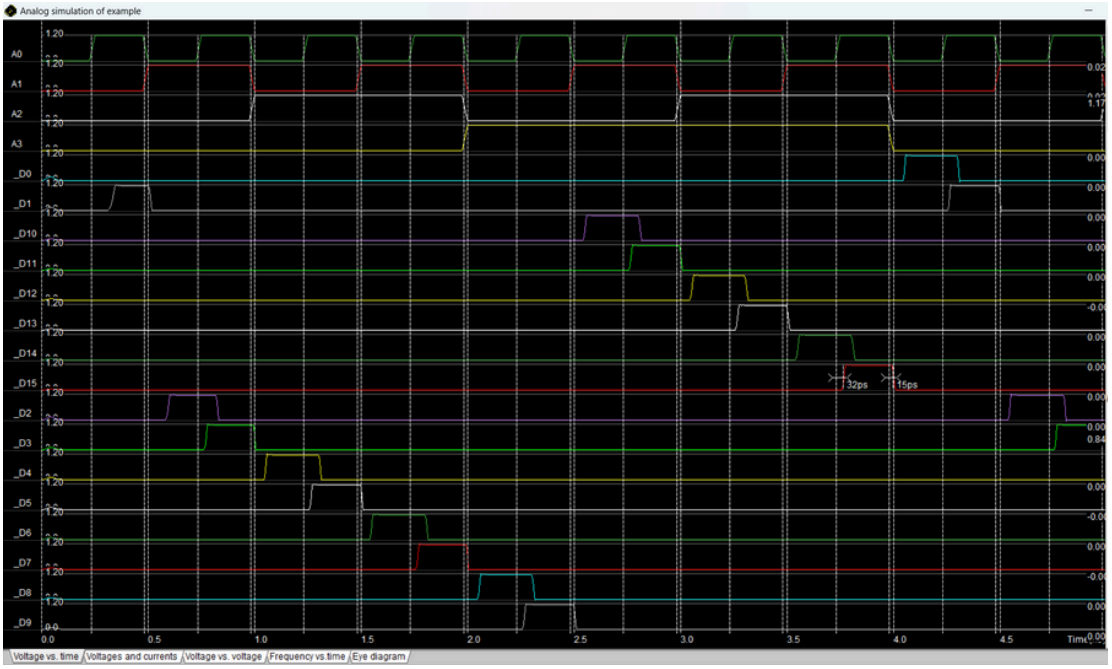
It's crucial to understand that VOL for the active outputs is low (close to 0V), while VOL for the inactive outputs is relatively higher, typically close to the supply voltage (VCC). The specific voltage levels can vary based on the technology and the actual circuit implementation, but this general behavior is consistent with most digital logic circuits.

Layout using Microwind tool:-

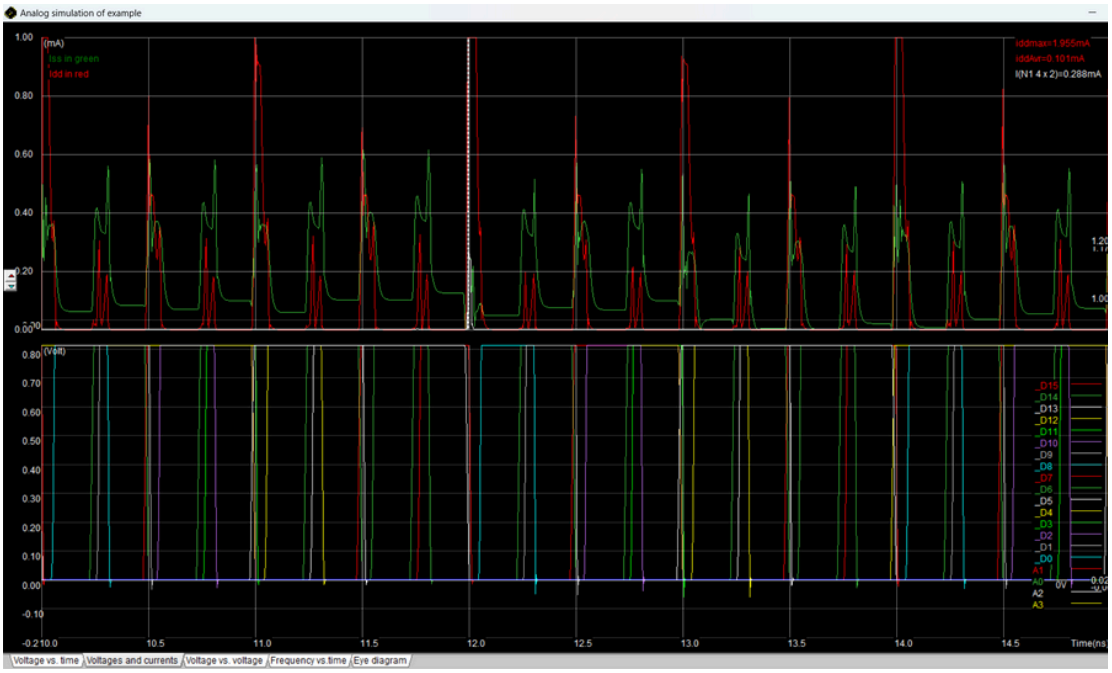
For 4 to 16 Decoder layout in Microwind, we compile the Verilog code into microwind and get these results. Here A3, A2, A1, A0 are the inputs and D1 to D15 are output is shown. We prepare the 4-to-16-line decoder in 0.12 micrometer technology.



Simulation Results:-



Voltage v/s Voltage



Voltage v/s Current

Delay(low to high, Delay(high to low) and Propagation delay :-

	Delay(LH)	Delay(HL)	P.D.
D0	67	70	68.5
D1	27	16	21.5
D2	61	71	66
D3	28	13	20.5
D4	65	72	68.5
D5	29	17	23
D6	64	72	68
D7	29	13	21
D8	69	73	71
D9	30	19	24.5
D10	64	74	69
D11	31	16	23.5
D12	68	74	71
D13	31	20	25.5
D14	65	76	70.5
D15	32	15	23.5

Conclusion:-

The primary goal of this project is to design and generate the layout for a 4-to-16 decoder. To accomplish this, we began by developing a truth table and constructed the corresponding gate-level circuit. Given the complexity of manually creating a transistor-level schematic for such an extensive circuit, we utilized Verilog to implement the design. The Verilog code, written in Quartus II using gate-level modeling, produced the optimized RTL representation of the decoder.

Next, we imported the Verilog code into Microwind, where the tool automatically generated the necessary CMOS and metal connections based on the code. Through simulation, we verified the functionality, obtaining the expected results: 16 unique outputs, labeled D0 to D15, corresponding to each of the 16 input combinations. The performance analysis showed the delay (high to low), delay (low to high), and the overall propagation delay for the circuit for all output combinations (D0 to D15), indicating efficient switching behavior.