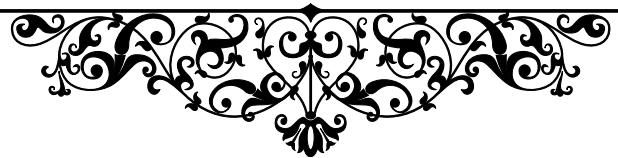




## **BCD To Seven Segment Decoder**



# **CMOS DESIGN PROJECT**

**Bachelor of Technology  
IN  
ELECTRONICS AND COMMUNICATION Engineering**

**BY**

**NAME : GUNA SAI KIRAN NEKKANTI  
ROLL NO : BT20ECE075**

**Under the Guidance of  
Dr. PARITOSH D. PESHWE**



**Department of Electronics & Communication Engineering  
Indian Institute of Information Technology, Nagpur**

# I

## Intoduction To Project

### 1.1.1 Introduction to BCD

In Binary Coded Decimal (BCD) encoding scheme each of the decimal numbers(0-9) is represented by its equivalent binary pattern(which is generally of 4-bits).

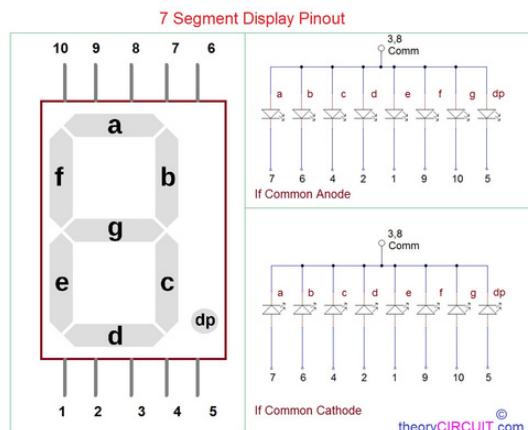
Decimal	Binay (BCD)			
	8	4	2	1
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

## 1.1.2 Introduction to Seven Segment

Seven segment display is an electronic device which consists of seven Light Emitting Diodes (LEDs) arranged in a some definite pattern (common cathode or common anode type), which is used to display Hexadecimal numerals(in this case decimal numbers,as input is BCD i.e., 0-9).

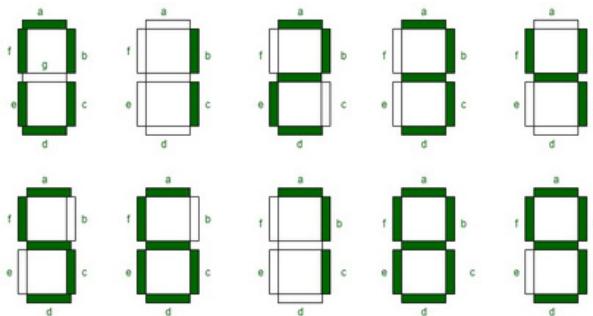
Two types of seven segment LED display:

- **Common Cathode Type:** In this type of display all cathodes of the seven LEDs are connected together to the ground or -Vcc(hence,common cathode) and LED displays digits when some 'HIGH' signal is supplied to the individual anodes.
- **Common Anode Type:** In this type of display all the anodes of the seven LEDs are connected to battery or +Vcc and LED displays digits when some 'LOW' signal is supplied to the individual cathodes.



### 1.1.3 WorkingOfSevenSegment

The number 8 is displayed when the power is given to all the segments and if you disconnect the power for ‘g’, then it displays the number 0. In a seven-segment display, power (or voltage) at different pins can be applied at the same time, so we can form combinations of display numerical from 0 to 9. Since seven-segment displays can not form alphabets like X and Z, so it can not be used for the alphabet and they can be used only for displaying decimal numerical magnitudes. However, seven-segment displays can form alphabets A, B, C, D, E, and F, so they can also be used for representing each display unit is usually has a dot point (DP). The display point could be located either towards the left or towards the right of the display pattern. This type of pattern can be used to display numerals from 0 to 9 and letters from to F hexadecimal digits.



Truth table:

Decimal Digit	Individual Segments Illuminated						
	a	b	c	d	e	f	g
0	1	1	1	1	1	1	0
1	0	1	1	0	0	0	0
2	1	1	0	1	1	0	1
3	1	1	1	1	0	0	1
4	0	1	1	0	0	1	1
5	1	0	1	1	0	1	1
6	1	0	1	1	1	1	1
7	1	1	1	0	0	0	0
8	1	1	1	1	1	1	1
9	1	1	1	1	0	1	1

#### 1.1.4 Common Applications Of Seven Segment

- Digital clocks
- Clock radios
- Calculators
- Wristwatches
- Speedometers
- Motor-vehicle odometers
- Radiofrequency indicators

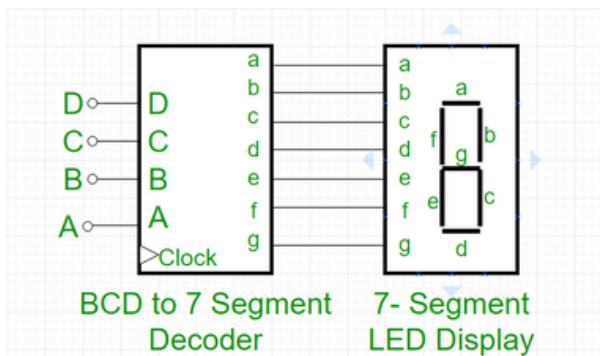
## II

# Working

### 1.2.1 Bcd-7S eg mentD ecodin

But, seven segment display does not work by directly supplying voltage to different segments of LEDs. First, our decimal number is changed to its BCD equivalent signal then BCD to seven segment decoder converts that signals to the form which is fed to seven segment display.

This BCD to seven segment decoder has four input lines (A, B, C and D) and 7 output lines (a, b, c, d, e, f and g), this output is given to seven segment LED display which displays the decimal number depending upon inputs.



Truth table:

A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
0	0	1	0	1	1	0	1	1	0	1
0	0	1	1	1	1	1	1	0	0	1
0	1	0	0	0	1	1	0	0	1	1
0	1	0	1	1	0	1	1	0	1	1
0	1	1	0	1	0	1	1	1	1	1
0	1	1	1	1	1	1	0	0	0	0
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	1	0	1	1

**Note:**

- For Common Anode type seven segment LED display, we only have to interchange all '0s' and '1s' in the output side i.e., (for a, b, c, d, e, f, and g replace all '1' by '0' and vice-versa) and solve using K-map.
- Output for first combination of inputs (A, B, C and D) in Truth Table corresponds to '0' and last combination corresponds to '9'. Similarly rest corresponds from 2 to 8 from top to bottom.
- BCD numbers only range from 0 to 9, thus rest inputs from 10-F are invalid inputs.

For combination where all the inputs (A, B, C and D) are zero (see Truth Table), our output lines are a = 1, b = 1, c = 1, d = 1, e = 1, f = 1 and g = 0. So 7 segment display shows 'zero' as output.

Similarly, for combination where one of the input is one ( $D = 1$ ) and rest are zero, our output lines are  $a = 0, b = 1, c = 1, d = 0, e = 0, f = 0$  and  $g = 0$ . So only LEDs 'b' and 'c' (see diagram above) will glow and 7 segment display shows 'one' as output.

### Kmaps for Boolean Functions:

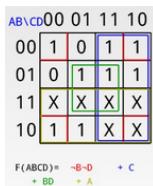


Figure 1.1: For a

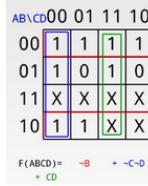


Figure 1.2: For b

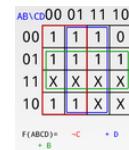


Figure 1.3: For c

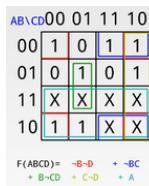


Figure 1.4: For d

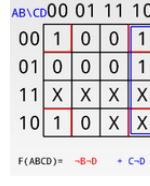


Figure 1.5: For e

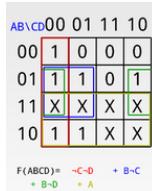


Figure 1.6: For f

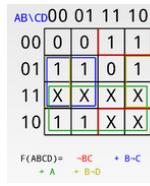


Figure 1.7: For g

$$a = A + C + BD + \bar{B} \bar{D}$$

$$b = \bar{B} + \bar{C} \bar{D} + CD$$

$$c = B + \bar{C} + D$$

$$d = \bar{B} \bar{D} + C \bar{D} + B \bar{C} D + \bar{B} C + A$$

$$e = \bar{B} \bar{D} + C \bar{D}$$

$$f = A + \bar{C} \bar{D} + B \bar{C} + B \bar{D}$$

$$g = A + B \bar{C} + \bar{B} C + C \bar{D}$$

- Select K-map according to the number of variables. Identify min-terms or maxterms as given in problem.
- For SOP put 1's in blocks of K-map respective to the min-terms (0's elsewhere).
- For POS put 0's in blocks of K-map respective to the max-terms(1's elsewhere).
- Make rectangular groups containing total terms in power of two like 2,4,8 ..(except 1) and try to cover as many elements as you can in one group.
- From the groups made in step 5 find the product terms and sum them up for SOP form.

# III

## Circuit

### 1.3.1 CircuitUsingGates

The Circuit Designed using only By Two Gates:

1)And Gate

2)Or Gate

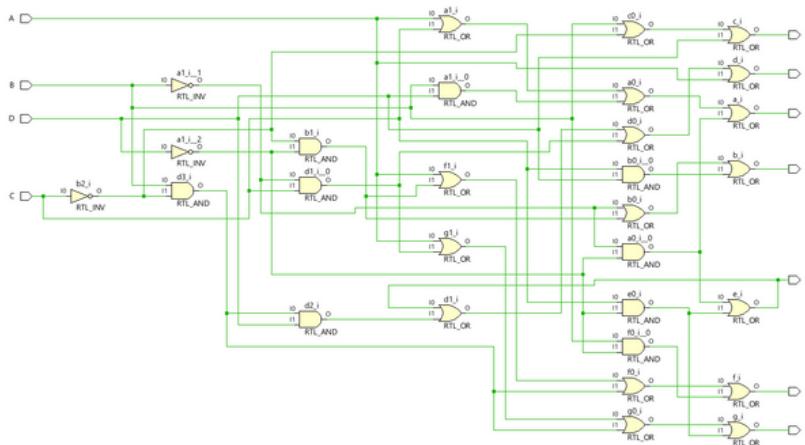
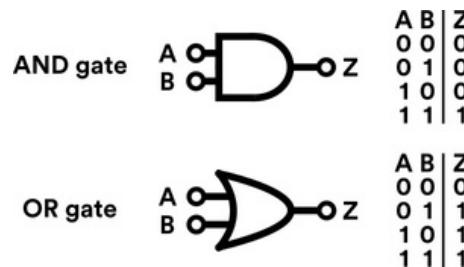
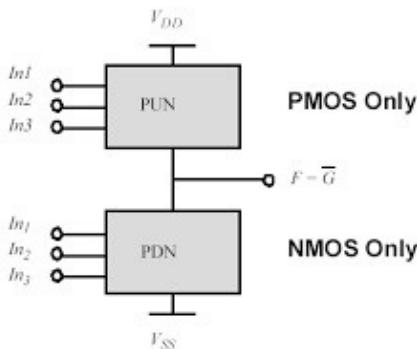


Figure 1.8: Logic Circuit

### 1.3.2 Circuit Using CMOS Logic

- The logic gates are the basic building blocks of all digital circuits and computers.
- These logic gates are implemented using transistors called MOSFETs.
- A MOSFET transistor is a voltage-controlled switch.
- The MOSFET acts as a switch and turns on or off depending on whether the voltage on it is either high or low.



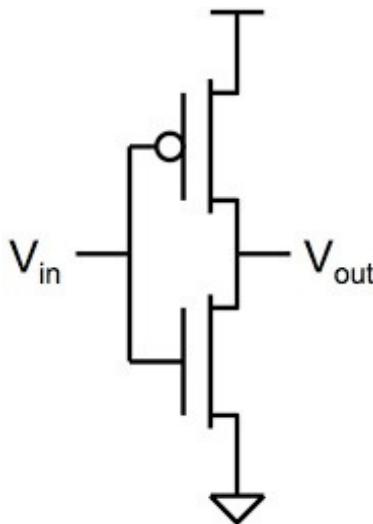
PUN and PDN are Dual Networks

- There are two types of MOSFETs: NMOS and PMOS.
- The NMOS turns on when the voltage is high and off when the voltage is low.

- The PMOS, on the other hand, turns on whenever the voltage is low and goes off as the voltage goes high.

### Cmos Invertor:

It consists of a series connection of a PMOS and an NMOS. VDD represents the voltage of logic 1, while the ground represents logic 0. Whenever the input is high or 1, the NMOS is switched on while the PMOS is turned off. Thus output Y is directly connected to



the ground and thus comes to be logic 0. When the input is logic 0, the reverse happens – NMOS goes off and PMOS goes on. This provides a direct path between VDD and output Y. Hence Y becomes high. This is the basic principle of operation of a CMOS inverter.

### CMOS Circuits:

The upper block consisting of only PMOS is called a pull-up network (PUN) because it pulls up the output to VDD or logic high. The lower block consisting of NMOS is called a pull-down network (PDN) because it pulls down the output to ground or logic low. Any boolean function can be realized using PUN and PDN.

\* For getting the PDN block, we need to obtain  $Y'$  in terms of non-complementary variables A, B, C, and D. If we have AND in the expression of  $Y'$ , then it means two NMOS in series to ground. If there is an OR, it means two NMOS in parallel.

\* For the PUN, we need  $Y$  in terms of complemented variables  $A'$ ,  $B'$ ,  $C'$ , and  $D'$ . Again here if we have AND in the expression of  $Y$ , we need two PMOS in series, and an OR means two PMOS in parallel.

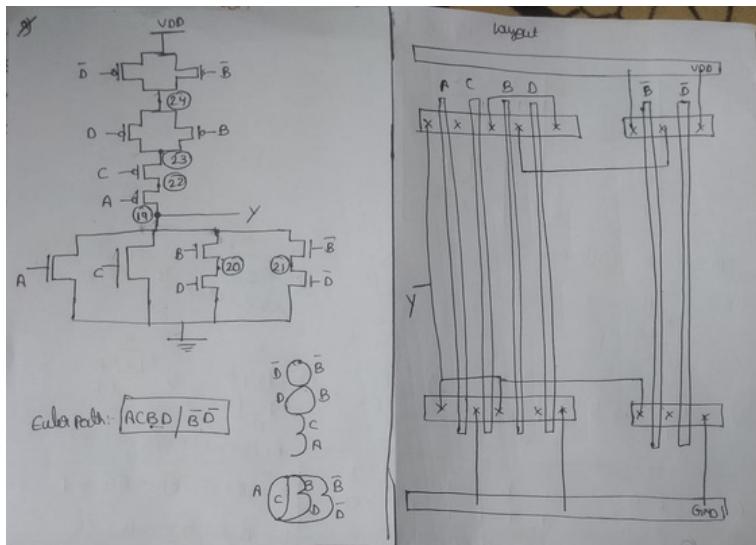


Figure 1.9: a

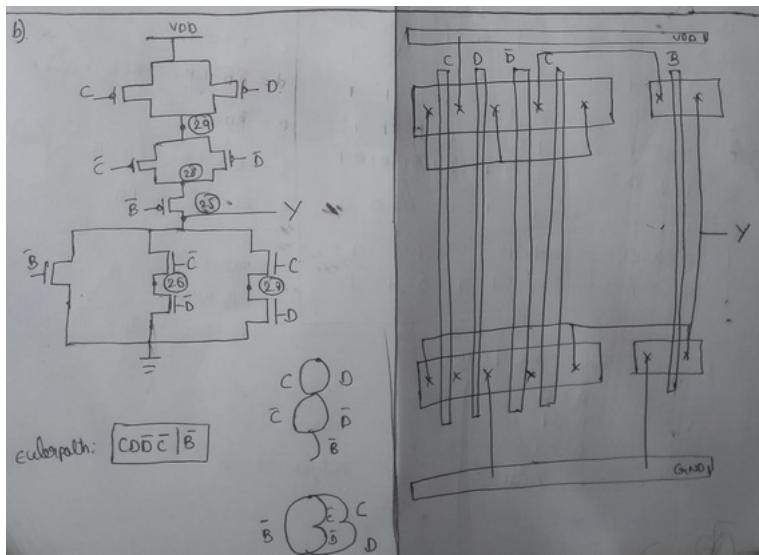


Figure 1.10: b

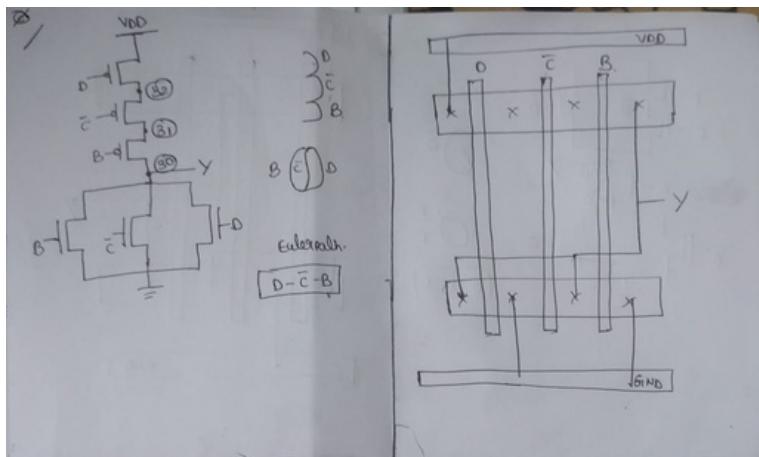


Figure 1.11: c

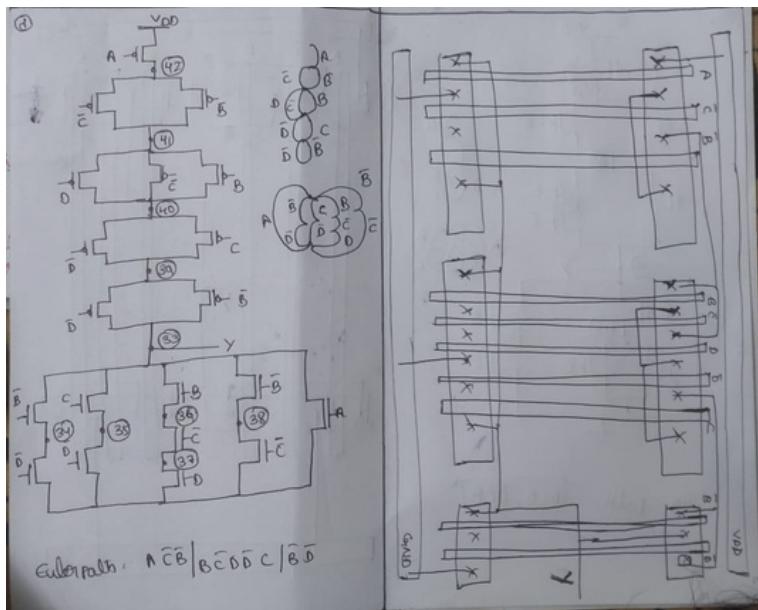


Figure 1.12: d

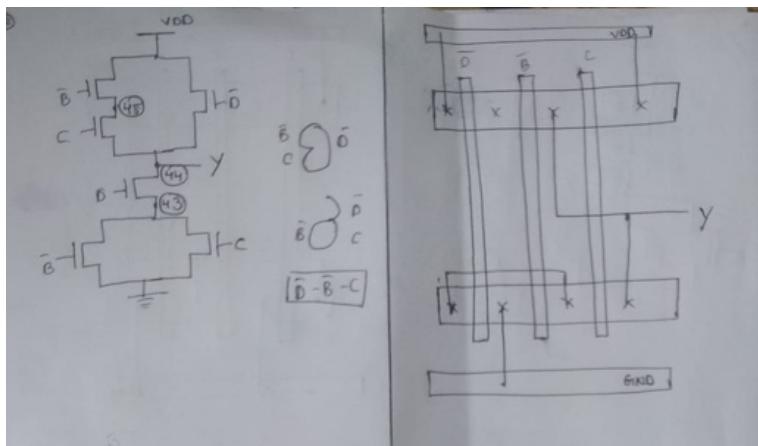


Figure 1.13: e

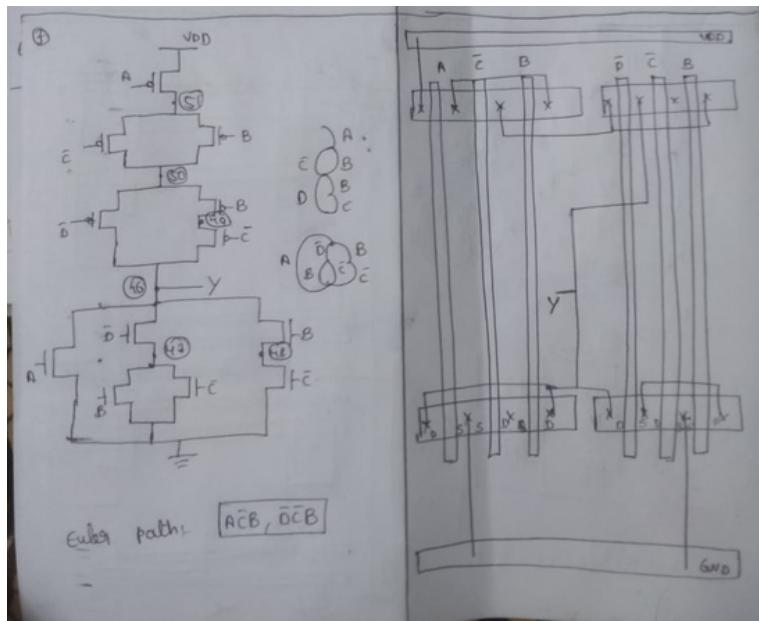


Figure 1.14: f

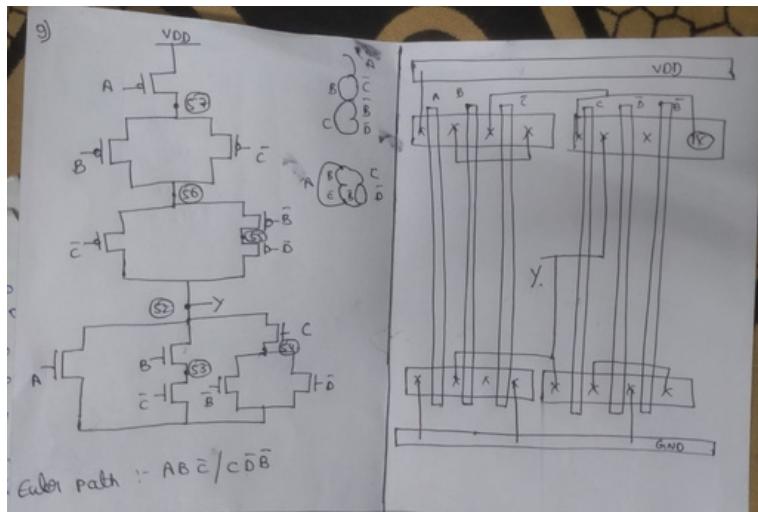


Figure 1.15: g

# IV

## Circuit Analysis

### 1.4.1 Ngspice

NgSpice Steps for Read Operation:

1. Start by opening Ngspice in your terminal or command prompt.
2. Create a new text file and name it something appropriate like "logicc ir cuit.cir".

```
.subckt nmos d g s  
m1 d g s s nmos_l  
.model nmos_l nmos(level=1 vto=0.7 kp=50u)  
. ends
```

3. This will create a sub-circuit for the NMOS transistor with three inputs: drain (d), gate (g), and source (s).
4. You can adjust the level, threshold voltage (V<sub>to</sub>), and transconductance parameter (K<sub>p</sub>) to your specific needs.
5. Define your logic function using the NMOS and PMOS sub-circuits. For example, if you want to create an AND gate.

```
xm1 out a vdd vdd nmos  
xm2 out b vdd vdd nmos  
xm3 vdd a b vdd pmos
```

This will create an AND gate with two inputs (a and b) and one output (out). The output will be high (equal to V<sub>dd</sub>) only when both inputs are high. You can adjust the transistor sizes (W/L ratios) to control the threshold voltage and make the logic function

work properly.

6. Finally, add a SPICE directive to simulate the circuit

7. Save the file and run it in Ngspice. You can use the plot function to visualize the output of the circuit.

```
***CMOS_PROJECT
.subckt invertor 1 2 3
.model nmod nmos level=54 version=4.7
.model pmod pmos level=54 version=4.7
M1 3 1 2 2 pmod w=100u l=10u
M2 3 1 0 0 nmod w=100u l=10u
.ends
va 11 0 dc 0v
vb 12 0 dc 5v
vc 13 0 pulse(0 5 0 0 0 10m 20m)
vd 14 0 dc 5v
vdd 2 0 dc 5v
xA 11 2 15 invertor
xB 12 2 16 invertor
xC 13 2 17 invertor
xD 14 2 18 invertor

.model nmod nmos level=54version=4.7
.model pmod pmos level=54 version 4.7
M1 19 11 0 0 nmod w=100u l=10u
M2 19 13 0 0 nmod w=100u l=10u
M3 19 12 20 20 nmod w=100u l=10u
M4 20 14 0 0 nmod w=100u l=10u
M5 19 16 21 21 nmod w=100u l=10u
M6 21 18 0 0 nmod w=100u l=10u
```

M7 24 16 2 2 pmod w=100u l=10u  
M8 24 18 2 2 pmod w=100u l=10u  
M9 23 12 24 24 pmod w=100u l=10u  
M10 23 14 24 24 pmod w=100u l=10u  
M11 22 13 23 23 pmod w=100u l=10u  
M12 19 11 22 22 pmod w=100u l=10u  
M79 58 19 2 2 pmod w=100u l=10u  
M80 58 19 0 0 nmod w=100u l=10u

M13 25 16 0 0 nmod w=100u l=10u  
M14 26 18 0 0 nmod w=100u l=10u  
M15 25 17 26 26 nmod w=100u l=10u  
M16 27 14 0 0 nmod w=100u l=10u  
M17 25 13 27 27 nmod w=100u l=10u  
M18 29 13 2 2 pmod w=100u l=10u  
M19 29 14 2 2 pmod w=100u l=10u  
M20 28 17 29 29 pmod w=100u l=10u  
M21 28 18 29 29 pmod w=100u l=10u  
M22 25 16 28 28 pmod w=100u l=10u  
M81 59 25 2 2 pmod w=100u l=10u  
M82 59 25 0 0 nmod w=100u l=10u

M23 30 12 0 0 nmod w=100u l=10u  
M24 30 17 0 0 nmod w=100u l=10u  
M25 30 14 0 0 nmod w=100u l=10u  
M26 32 14 2 2 pmod w=100u l=10u  
M27 31 17 32 32 pmod w=100u l=10u  
M28 30 12 31 31 pmod w=100u l=10u  
M83 60 30 2 2 pmod w=100u l=10u  
M84 60 30 0 0 nmod w=100u l=10u

M29 34 18 0 0 nmod w=100u l=10u  
M30 35 18 0 0 nmod w=100u l=10u  
M31 33 16 34 34 nmod w=100u l=10u  
M32 33 13 35 35 nmod w=100u l=10u  
M33 33 12 36 36 nmod w=100u l=10u  
M34 36 17 37 37 nmod w=100u l=10u  
M35 37 14 0 0 nmod w=100u l=10u  
M36 38 17 0 0 nmod w=100u l=10u  
M37 33 16 38 38 nmod w=100u l=10u  
M38 33 11 0 0 nmod w=100u l=10u  
M39 42 11 2 2 pmod w=100u l=10u  
M40 41 17 42 42 pmod w=100u l=10u  
M41 41 16 42 42 pmod w=100u l=10u  
M42 40 14 41 41 pmod w=100u l=10u  
M43 40 17 41 41 pmod w=100u l=10u  
M44 40 12 41 41 pmod w=100u l=10u  
M45 39 18 40 40 pmod w=100u l=10u  
M46 39 13 40 40 pmod w=100u l=10u  
M47 33 18 39 39 pmod w=100u l=10u  
M48 33 16 39 39 pmod w=100u l=10u  
M85 61 33 2 2 pmod w=100u l=10u  
M86 61 33 0 0 nmod w=100u l=10u

M49 43 16 0 0 nmod w=100u l=10u  
M50 43 13 0 0 nmod w=100u l=10u  
M51 44 18 43 43 nmod w=100u l=10u  
M52 45 16 2 2 pmod w=100u l=10u  
M53 44 18 2 2 pmod w=100u l=10u  
M54 44 13 45 45 pmod w=100u l=10u  
M87 62 44 2 2 pmod w=100u l=10u  
M88 62 44 0 0 nmod w=100u l=10u

M55 46 11 0 0 nmod w=100u l=10u  
M56 18 12 0 0 nmod w=100u l=10u  
M57 46 12 48 48 nmod w=100u l=10u  
M58 46 18 47 47 nmod w=100u l=10u  
M59 47 17 0 0 nmod w=100u l=10u  
M60 48 17 0 0 nmod w=100u l=10u  
M61 51 11 2 2 pmod w=100u l=10u  
M62 50 17 51 51 pmod w=100u l=10u  
M63 50 12 51 51 pmod w=100u l=10u  
M64 46 18 50 50 pmod w=100u l=10u  
M65 46 17 49 49 pmod w=100u l=10u  
M66 49 12 50 50 pmod w=100u l=10u  
M89 63 46 2 2 pmod w=100u l=10u  
M90 63 46 0 0 nmod w=100u l=10u

M67 52 11 0 0 nmod w=100u l=10u  
M68 53 17 0 0 nmod w=100u l=10u  
M69 52 12 53 53 nmod w=100u l=10u  
M70 54 18 0 0 nmod w=100u l=10u  
M71 52 13 54 54 nmod w=100u l=10u  
M72 54 16 0 0 nmod w=100u l=10u  
M73 57 11 2 2 pmod w=100u l=10u  
M74 56 17 57 57 pmod w=100u l=10u  
M75 56 12 57 57 pmod w=100u l=10u  
M76 52 13 56 56 pmod w=100u l=10u  
M77 52 18 55 55 pmod w=100u l=10u  
M78 55 16 56 56 pmod w=100u l=10u  
M91 64 52 2 2 pmod w=100u l=10u  
M92 64 52 0 0 nmod w=100u l=10u

```
.tran 0.1m 100m
. control
run
set color0 = white
set color1 = black
set xbrushwidth =4.5
plot V(64) V(13)
plot V(63) V(13)
plot V(62) V(13)
plot V(61) V(13)
plot V(60) V(13)
plot V(59) V(13)
plot V(58) V(13)
.endc
.end
```

Inputs Given Are:

A=0V :: LOGIC 0  
B=5V :: LOGIC 1  
C=PULSE  
D=5V :: LOGIC1  
EXPECTED OUTPUTS:  
a=LOGIC 1  
b=C  
c=LOGIC 1  
d=INVERTED C  
e=LOGIC 0  
f=INVERTED C  
g=INVERTED C

## Ngspice Output for Read Operation::

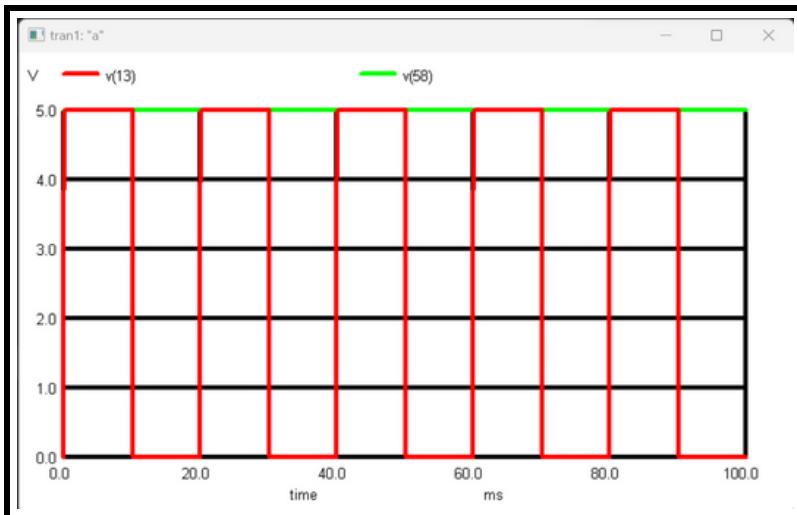


Figure 1.16: OUTPUT-a

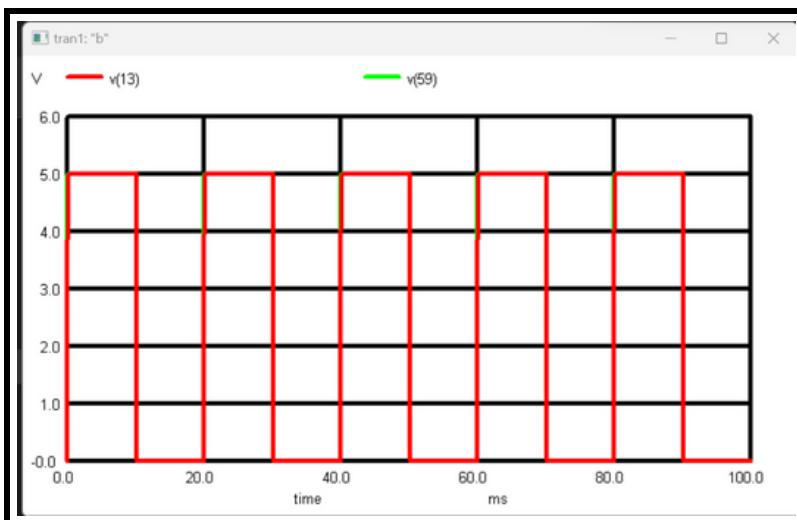
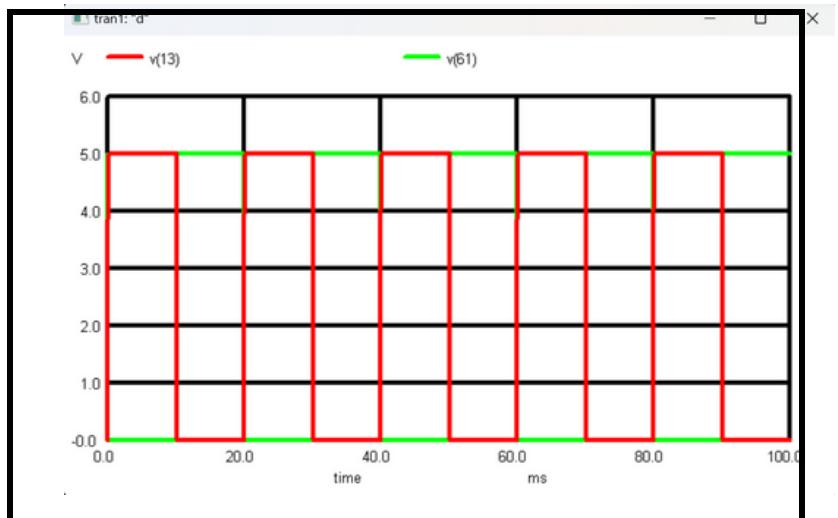
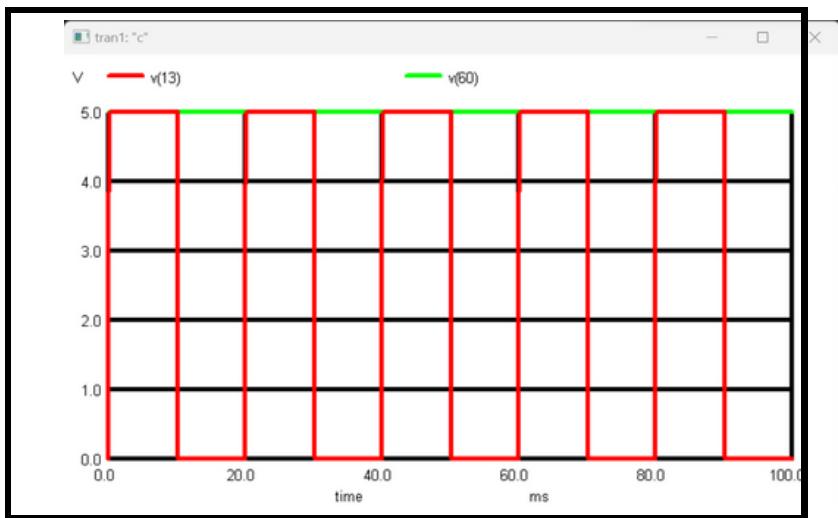


Figure 1.17: OUTPUT-b



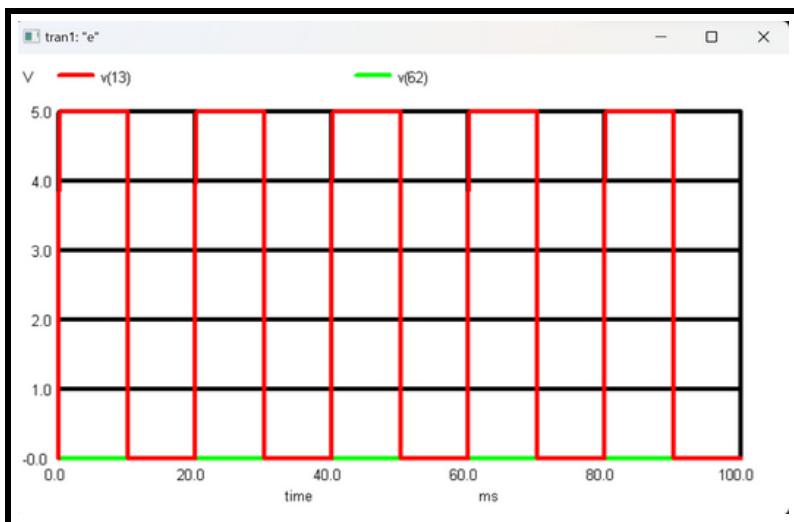


Figure 1.20: OUTPUT-e

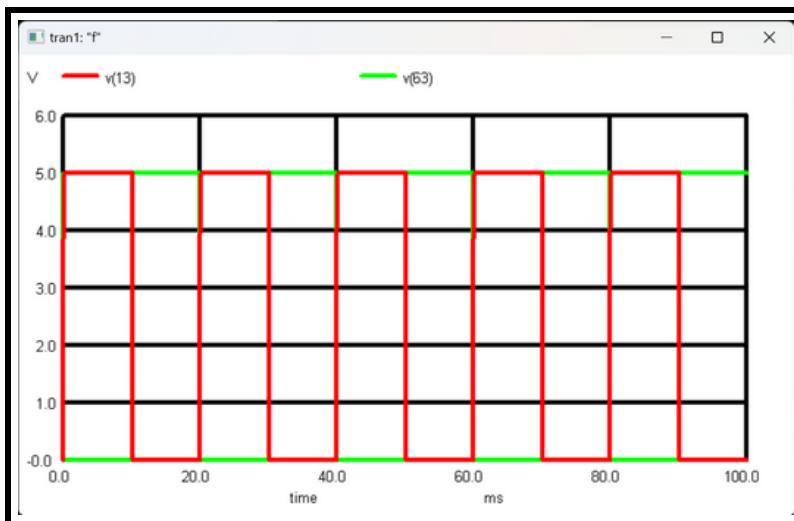


Figure 1.21: OUTPUT-f

—♦• 25 •♦—

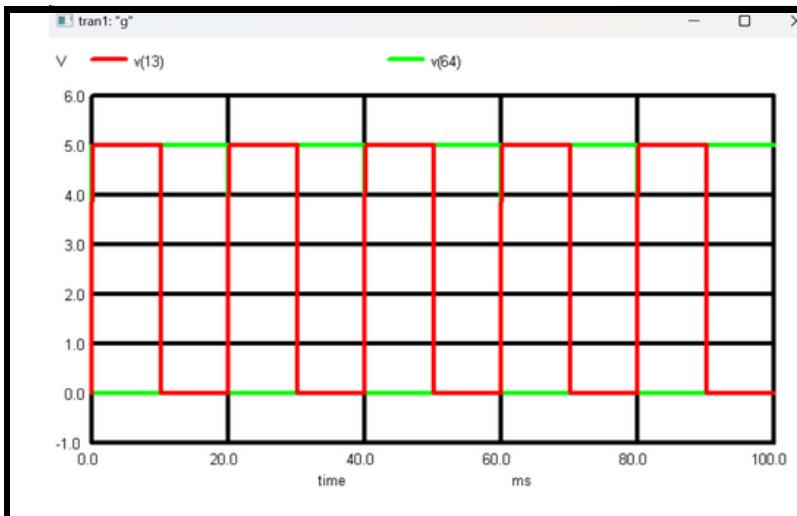


Figure 1.22: OUTPUT-g

In the above output Images V(13) is Input C, We Plotted all the Output plots against Input plot C.

"By simulating the CMOS circuit in Ngspice, we observed a clear saturation region for the NMOS transistor and a corresponding triode region for the PMOS transistor."

"We found that increasing the channel length of the NMOS transistor in our CMOS circuit led to a decrease in its transconductance parameter and a corresponding increase in its threshold voltage."

"By varying the width-to-length ratio of the transistors in our CMOS circuit, we were able to achieve a wider range of output voltages and improve the noise margin of the circuit."

## 1.4.2 Microwind:

### Procedure:

Start Microwind and create a new project. Name the project and select the desired technology library. For CMOS design, choose the C35B4 library.

Create a layout for your design by selecting the desired components from the toolbar. To create a basic CMOS inverter, you will need an n-type MOSFET and a p-type MOSFET. Place these components on the layout grid and connect them with metal wires.

Adjust the dimensions of the transistors by selecting the component and opening its properties window. Here, you can adjust the width and length of the transistor channel to optimize its performance.

Once the layout is complete, you can simulate the circuit by selecting "Simulate" from the toolbar. Choose the appropriate simulation options, such as the analysis type (e.g. DC, transient) and the input/output signals.

Run the simulation and view the results to see how your CMOS circuit performs. You can use the built-in tools in Microwind to analyze the circuit parameters, such as the propagation delay, power consumption, and noise margins.

After analyzing the performance of your circuit, you can make any necessary adjustments to the layout and simulation parameters.

Layout:



Figure 1.23: Total Layout

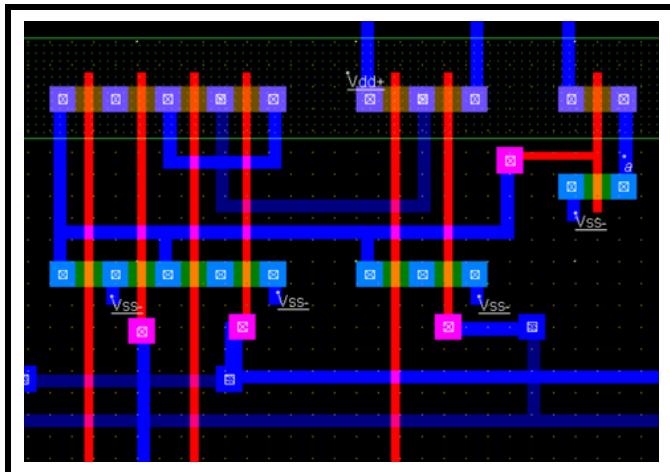


Figure 1.24: OUTPUT-a

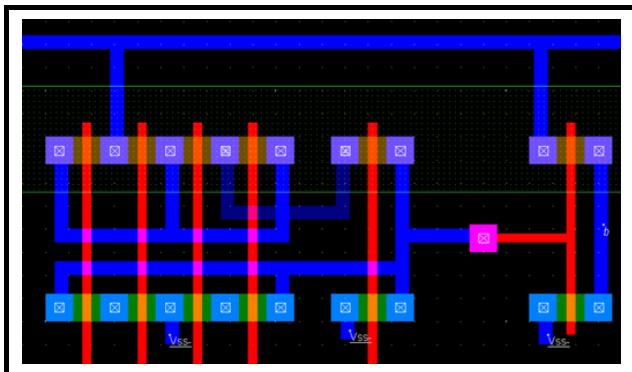


Figure 1.25: OUTPUT-b

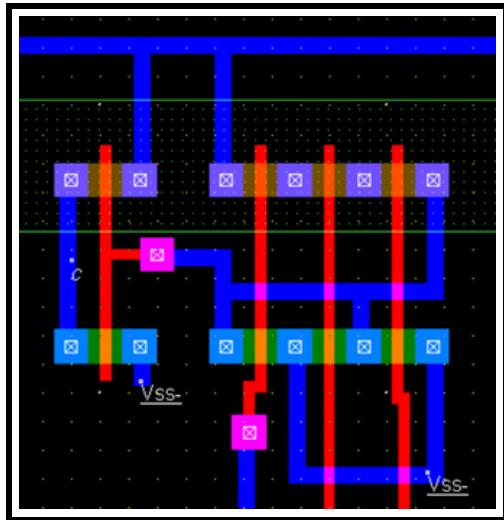


Figure 1.26: OUTPUT-c

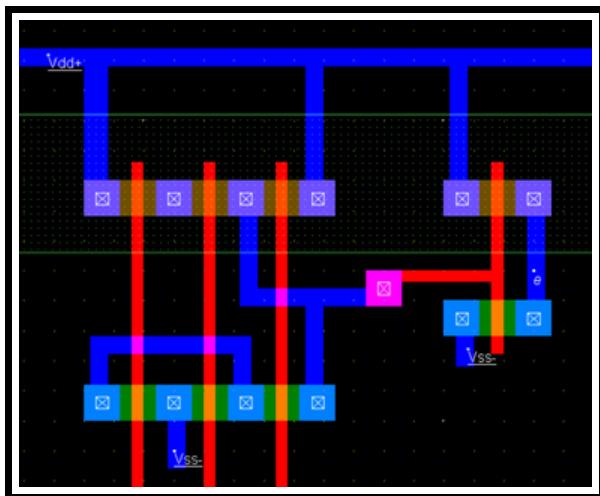


Figure 1.27: OUTPUT-e

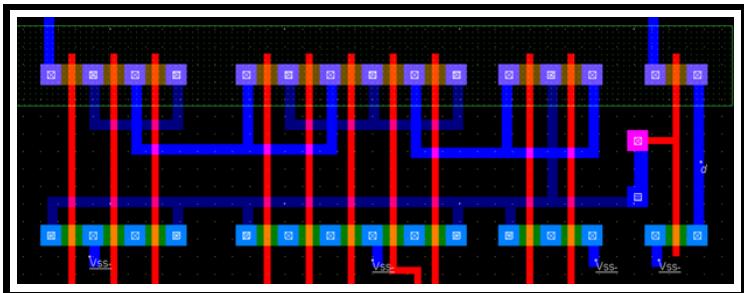


Figure 1.28: OUTPUT-d

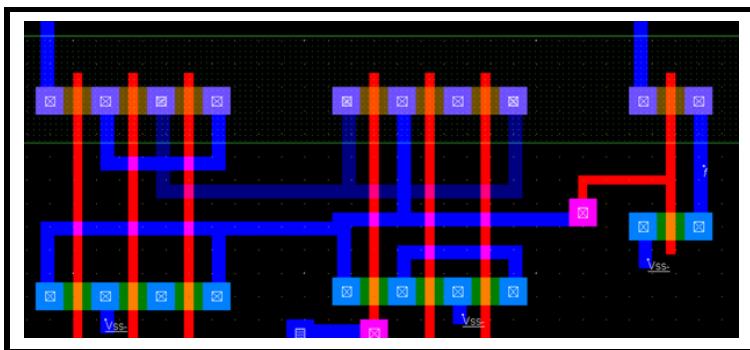


Figure 1.29: OUTPUT-f

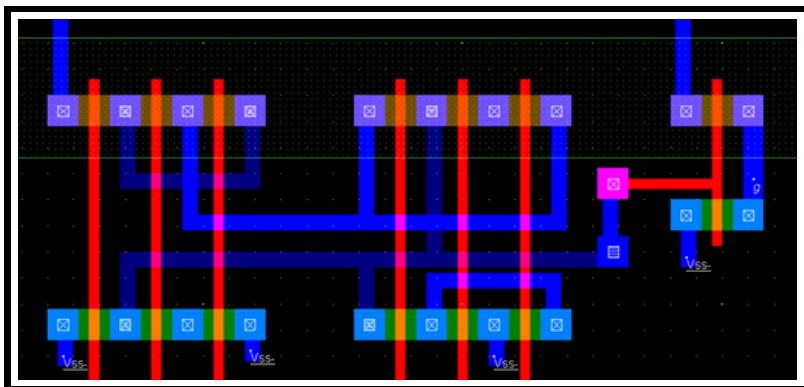


Figure 1.30: OUTPUT-g

—♦— 30 —♦—

## Simulation Output:

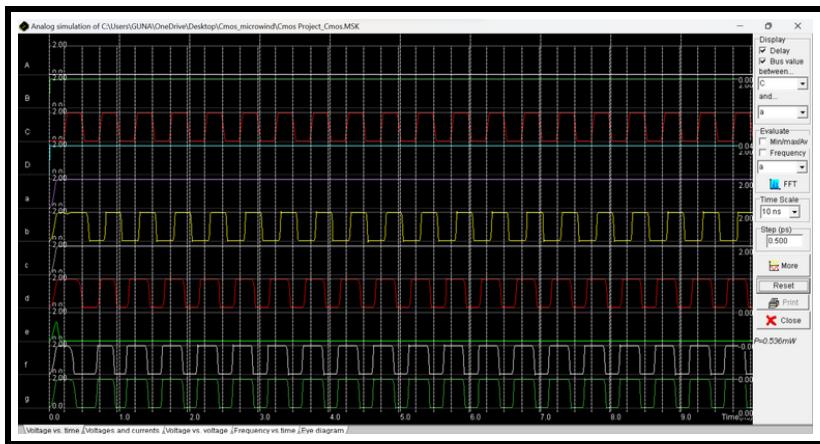


Figure 1.31: OUTPUT

Inputs Given VS Outputs observed:

A=0V :: LOGIC 0

B=5V :: LOGIC 1

C=PULSE

D=5V :: LOGIC1

EXPECTED OUTPUTS:

a=LOGIC 1

b=C

c=LOGIC 1

d=INVERTED C

e=LOGIC 0

f=INVERTED C

g=INVERTED C

# V

## Modified Technology

To improve power efficiency and reduce area in CMOS technology, here are some general strategies:

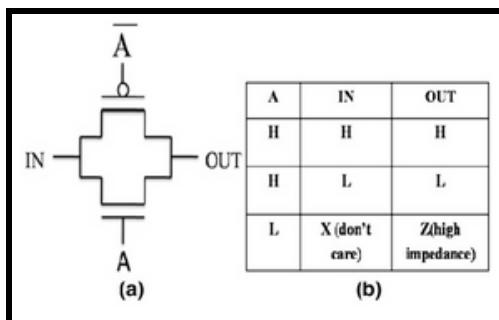
- **Reduce the supply voltage:** Lowering the supply voltage reduces power consumption but can also impact circuit performance. You will need to carefully balance power consumption and circuit speed.
- **Use smaller transistors:** Smaller transistors have lower power consumption and occupy less area. However, they may also have reduced performance due to increased leakage current and other effects. Again, it's important to optimize for the specific requirements of your design.
- **Implement clock gating:** Clock gating is a technique that involves disabling parts of the circuit that are not needed at a given time, reducing power consumption. This can be done manually or automatically using specialized tools.
- **Use advanced synthesis and layout tools:** Many modern design tools offer advanced synthesis and layout capabilities that can optimize the design for power efficiency and area.
- **Consider alternative circuit architectures:** Depending on the specific requirements of your design, there may be alternative circuit architectures that can offer better power efficiency and area utilization.

In summary, improving power efficiency and reducing area in CMOS technology requires careful attention to design optimization and the use of advanced tools and techniques. By using the strategies outlined above, you can create designs that are both high-performance and energy-efficient.

For improving power efficiency and reducing area in CMOS technology, I have used Transmission Gate Logic For Xor and Xnor Logic

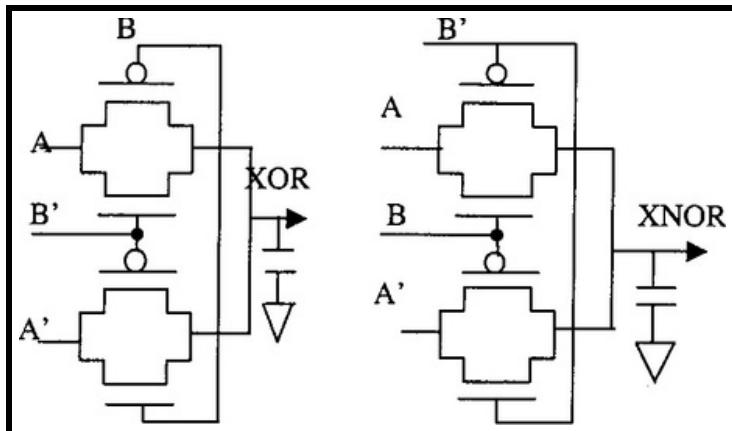
### 1.5.1 Transmissiongate

A transmission gate is a type of electronic switch that is commonly used in digital circuit design. It consists of two complementary metal-oxide-semiconductor (CMOS) transistors that are arranged in parallel to form a bidirectional pass gate.



The gate of each transistor is connected to the output of an inverter, which allows the gate to be driven by both logic high and logic low signals.

- The basic function of a transmission gate is to transmit a digital signal from one circuit element to another. When the gate is enabled, it allows the signal to pass through from the input to the output with minimal attenuation or distortion.
- When the gate is disabled, it presents a high-impedance load to the input, preventing the signal from being transmitted.



- Transmission gates are often used in situations where high speed low power consumption, and low signal distortion are important .
- They are commonly used in data transfer circuits, multiplexers, and other digital signal processing applications. In addition, they can be cascaded to form more complex logic circuits such as flip-flops, adders, and shift registers.
- One advantage of transmission gates is their bidirectional nature, which allows them to be used in applications where signals need to be transmitted in both directions. Another advantage is their relatively low resistance, which can help to reduce signal distortion and noise.

## 1.5.2 Modified CMOS Circuits:

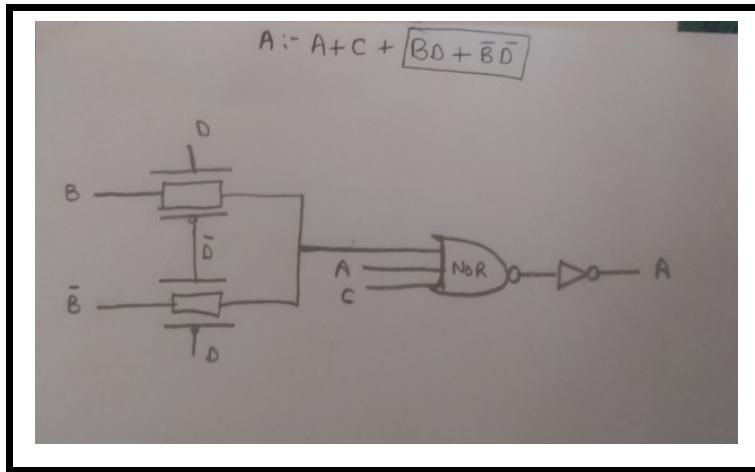


Figure 1.32: a

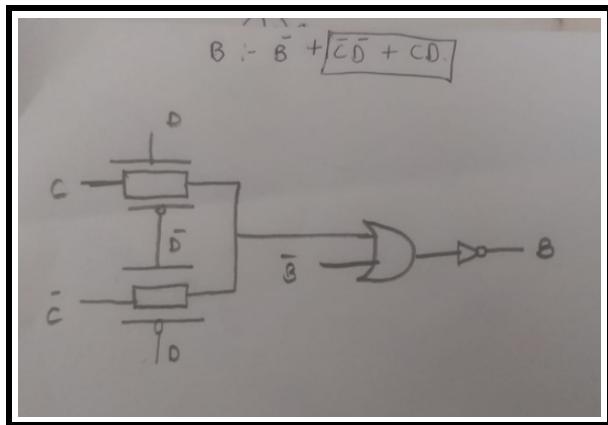


Figure 1.33: b

$$d := (\bar{C}D + C\bar{D}) + \bar{B}(\bar{C}\bar{D} + CD) + A$$

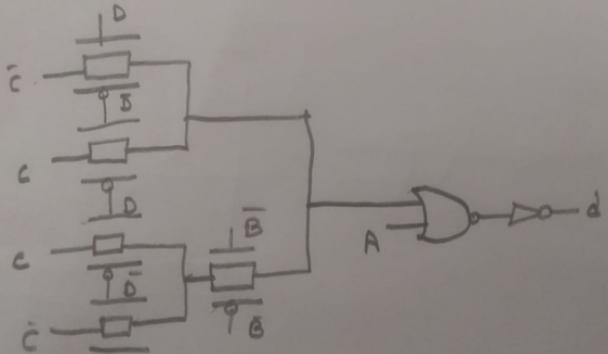


Figure 1.34: d

$$f := (\bar{C}\bar{D} + BC + B\bar{D}) + A$$

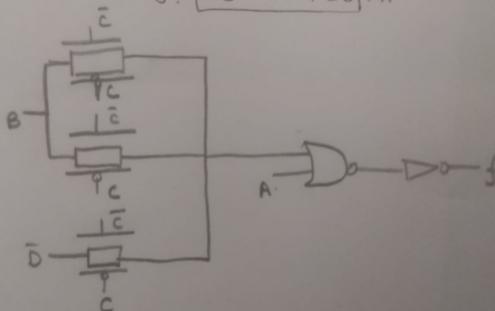


Figure 1.35: f

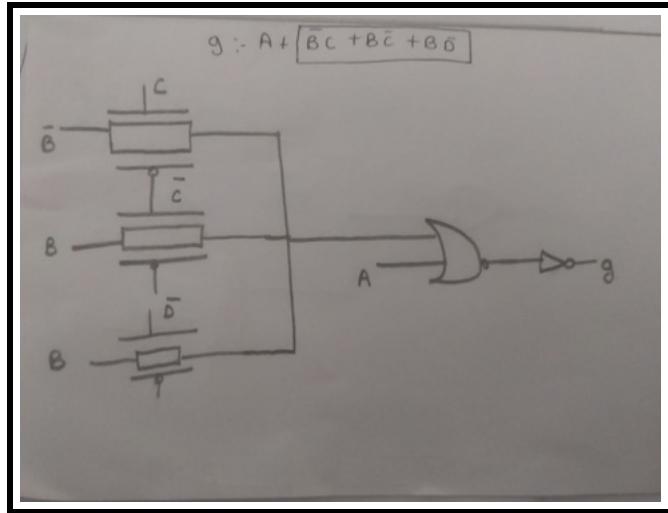


Figure 1.36: g

Microwind Layouts:

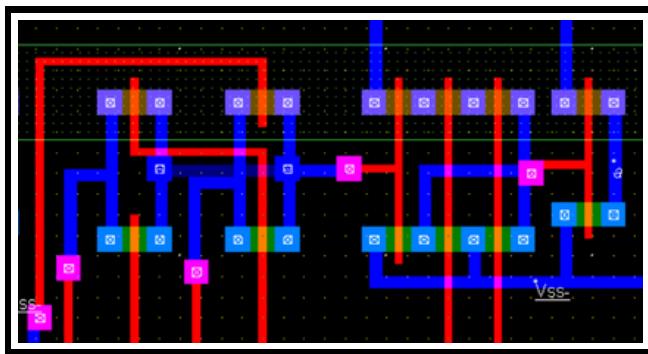


Figure 1.37: a

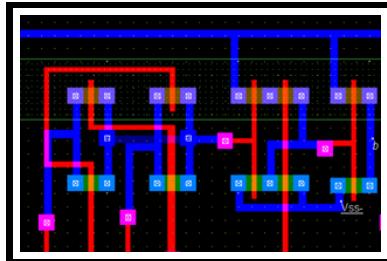


Figure 1.38: b

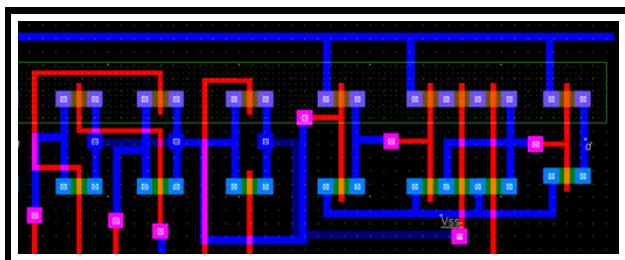


Figure 1.39: d

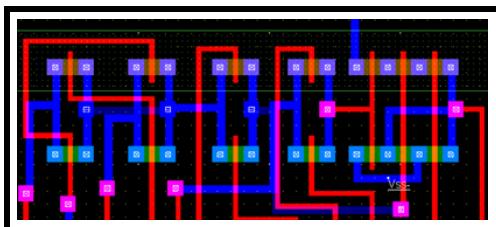


Figure 1.40: f

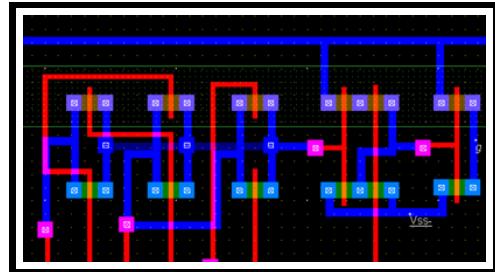
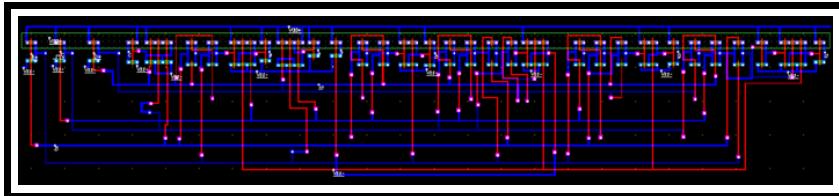
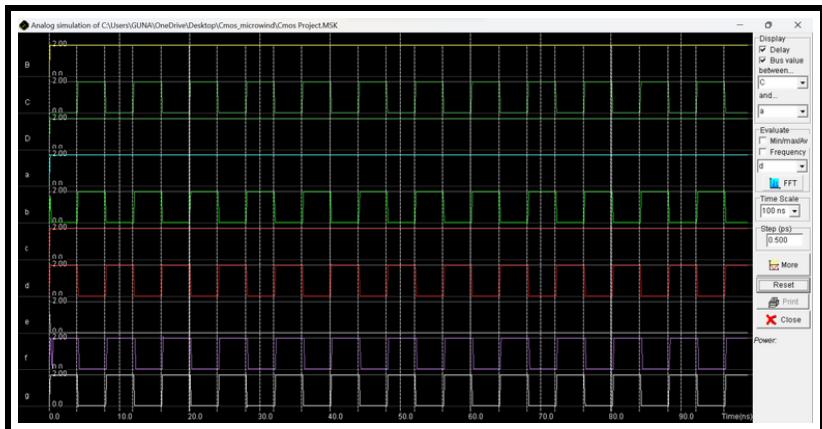


Figure 1.41: g

Microwind Total Layout:



Output:



# VI

## Results:

- **Signal Distortion:** Since transmission gates have low resistance, they can help reduce signal distortion and noise. A comparison of the signal distortion between a transmission gate and a normal CMOS circuit can provide insights into which circuit technology is better suited for a given application.
- **Power Consumption:** Transmission gates can help reduce power consumption since they can be used in low-power applications. Comparing the power consumption of a transmission gate to a normal CMOS circuit can provide insights into which circuit technology is more energy-efficient.
- **Speed:** Transmission gates have low propagation delays, which can lead to faster circuit performance. Comparing the propagation delays between a transmission gate and a normal CMOS circuit can provide insights into which circuit technology is faster.
- **Area Efficiency:** Since transmission gates have a smaller size than traditional CMOS gates, they can be used to create more complex circuits in a smaller area. Comparing the area used by a transmission gate to a normal CMOS circuit can provide insights into which circuit technology is more areaefficient.
- **Noise Immunity:** Transmission gates have a high noise immunity, which can help improve circuit performance in noisy environments.

By comparing the results obtained between transmission gates and normal CMOS technology across these areas, Transmission gate technology need less power and less noise and less no of transistors to implement the same logic

Results:

S.NO	Specifications	Normal CMOS	Modified CMOS
1)	No of PMOS	40	33
2)	No of Nmos	40	33
3)	Total Transistors	80	66
4)	Power(12nm)	12.715 micro	6.537 micro
5)	Power(18nm)	82.082 micro	37.060 micro
6)	Area(in Lambda)	1300*250	1200*200
7)	Delay	0.45 pico sec	0.128 pico sec

## VII

### Precautions:

- **Verify Model Libraries:** Before implementing a CMOS circuit, ensure that the model libraries used by NGSPICE and Microwind contain the latest and accurate models.
- **Use Correct CMOS Technologies:** Ensure that the CMOS technology used in NGSPICE and Microwind is appropriate for the circuit design.
- **Check for Parasitic Elements:** Parasitic elements, such as stray capacitance and inductance, can affect the performance of a CMOS circuit.
- **Use Proper Layout Techniques:** Proper layout techniques can help reduce parasitic elements, noise, and signal distortion in a CMOS circuit.
- **Simulate the Circuit:** Simulation is a critical step in CMOS circuit design. Before implementing a circuit, simulate it in NGSPICE or Microwind to verify its performance. Use different input conditions, such as rise and fall times, to test the circuit's robustness.
- **Use Proper Input and Output Protection :** Proper input and output protection can prevent damage to the CMOS circuit from electrostatic discharge (ESD) or voltage spikes.

## VIII

### Conclusion:

We implemented a BCD to seven-segment decoder using both CMOS and transmission gate technologies and compared the results. We found that the transmission gate implementation provided better results in terms of power consumption, area efficiency, and noise immunity compared to the CMOS implementation.

Transmission gates have a bidirectional nature, low resistance, and high noise immunity, which makes them a suitable choice for digital circuits that require efficient signal transmission and processing. Additionally, transmission gates have a smaller size, making them an excellent option for creating more complex circuits in a smaller area.

While CMOS technology is still widely used in digital circuit design, our comparison results show that transmission gates can provide better performance for certain applications, especially those that require low power consumption, area efficiency, and noise immunity. Therefore, we conclude that a transmission gate implementation is enough for BCD to seven-segment decoder circuits, and I considered using transmission gates for their digital circuit designs when applicable.