भारतीय सूचना प्रौद्योगिकी संस्थान, नागपुर

Indian Institute of Information Technology, Nagpur

An Institution of National Importance By An Act of Parliament

Electronics and Communication Engineering Department

Designing

TWO BIT COMPARATOR

Submitted By:

ANURAG KUMAR

BT20ECE055

Semester VI

Submitted To:

Dr PARITOSH PESHWE

ACKNOWLEDGMENT

I am grateful for the opportunity to express my heartfelt appreciation and thanks to Dr Paritosh Peshwe Sir for his guidance and support during my project on designing a two-bit comparator.

Dr Paritosh's mentorship and expertise have been invaluable throughout this project. His guidance helped me develop a deeper understanding of the subject matter and enabled me to approach the project with greater confidence and clarity.

I am also grateful to the faculty and staff of the Department of Electronics and Communication Engineering for their valuable suggestions and inputs during the course of this project.

Moreover, Paritosh Peshwe's encouragement and support have been a constant source of inspiration, and I am deeply grateful for the opportunity he provided me to work on this project. His mentorship has been instrumental in shaping my academic and professional growth, and I am genuinely grateful for his guidance.

Once again, I extend my heartfelt thanks to Paritosh Peshwe for his unwavering support, encouragement, and guidance during this project.

INTRODUCTION

A two-bit comparator is a digital circuit that compares two binary numbers of two bits each and determines whether they are equal or not. The circuit has **two input terminals**, each of which receives a two-bit binary number, and an output terminal that indicates whether the two numbers are equal or not.

The operation of the two-bit comparator is based on comparing the individual bits of the two numbers. The comparator compares the most significant bit **(MSB)** of each number first. If the two MSBs are equal, then the comparator compares the least important bit **(LSB)** of each number. If both MSBs and LSBs are equal, then the two numbers are considered equal, and the output of the comparator is set to logic high. Otherwise, the output of the comparator is set to logic low, indicating that the two numbers are not equal.

The two-bit comparator is a **fundamental building block** in digital circuits and is used in a wide range of applications, including digital signal processing, computer arithmetic, and control systems. The design and implementation of a two-bit comparator are relatively simple, and the circuit can be easily extended to compare numbers of greater bit lengths.

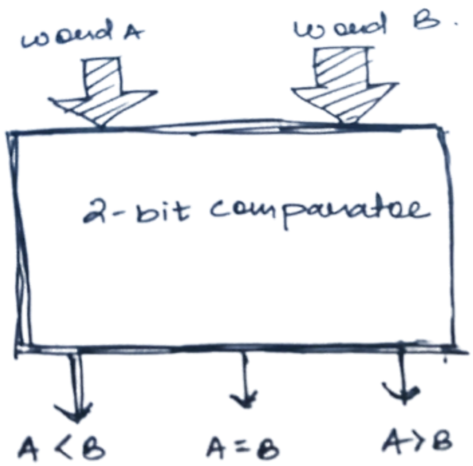


Figure 1

A two-bit comparator can be used in **various applications** where a comparison of two binary numbers of two bits each is required. Here are some of the common uses of a two-bit comparator:

Digital Signal Processing: In digital signal processing, a two-bit comparator can be used to compare two digital signals of two bits each. For instance, a two-bit comparator can be used in audio or video processing to determine if two signals are identical.

Computer Arithmetic: Two-bit comparators are used in digital computers' arithmetic and logical units. For example, in a computer's CPU, a two-bit comparator can be used to compare the operands of a binary addition or subtraction operation.

Control Systems: A two-bit comparator can be used in control systems to compare the input and output signals of a system. The comparator can be used to determine if the system is operating correctly or if adjustments need to be made.

Address Decoding: Two-bit comparators are used in address decoding circuits to select a specific memory location or device. In memory addressing, a two-bit comparator can be used to compare the two least significant bits of the memory address with the two-bit address input to select the correct memory location.

Overall, the two-bit comparator is a versatile and useful circuit that can be used in various applications that require binary number comparison.

ARCHITECTURE & OPERATIONS

The two-bit comparator is a digital circuit that compares two binary numbers of two bits each and determines whether they are equal or not. The comparator compares the most significant bit (MSB) of each number first. If the two MSBs are equal, then the comparator compares the least significant bit (LSB) of each number. If both MSBs and LSBs are equal, then the two numbers are considered equal, and the output of the comparator is set to logic high. Otherwise, the output of the comparator is set to logic low, indicating that the two numbers are not equal.

Here's a step-by-step operation of a two-bit comparator:

1. The two-bit binary numbers are applied to the two inputs of the comparator.
2. The comparator first compares the MSB of the two numbers. If the two MSBs are equal, the comparison proceeds to the LSB.
3. If the two MSBs are not equal, the comparator immediately sets the output to logic low, indicating that the two numbers are not equal.
4. If the two MSBs are equal, the comparator compares the LSB of the two numbers.
5. If the two LSBs are equal, the comparator sets the output to logic high, indicating that the two numbers are equal.
6. If the two LSBs are not equal, the comparator sets the output to logic low, indicating that the two numbers are not equal.

The output of the two-bit comparator is a binary signal that indicates whether the two input numbers are equal or not. The output is high if the two numbers are equal and low if they are not.

BOOLEAN EXPRESSION

i) For $A < B$:-

$$Y = \bar{A}_1 \bar{A}_0 B_0 + \bar{A}_1 B_1 + \bar{A}_0 B_1 B_0$$

ii) For $A = B$:-

$$Y = (A_1 \odot B_1)(A_0 \odot B_0)$$
$$= (\overline{A_1 \oplus B_1})(\overline{A_0 \oplus B_0})$$
$$= (\overline{\bar{A}_1 B_1 + A_1 \bar{B}_1})(\overline{\bar{A}_0 B_0 + A_0 \bar{B}_0})$$
$$Y = (A_1 + \bar{B}_1)(\bar{A}_1 + B_1)(A_0 + \bar{B}_0)(\bar{A}_0 + B_0)$$

iii) For $A > B$:-

$$Y = A_1 \bar{B}_1 B_0 + A_1 A_0 \bar{B}_0 + A_1 \bar{B}_1$$

TRUTH TABLE

A1	A0	B1	B0	A<B	A=B	A>B
0	0	0	0	0	1	0
0	0	0	1	1	0	0
0	0	1	0	1	0	0
0	0	1	1	1	0	0
0	1	0	0	0	0	1
0	1	0	1	0	1	0
0	1	1	0	1	0	0
0	1	1	1	1	0	0
1	0	0	0	0	0	1
1	0	0	1	0	0	1
1	0	1	0	0	0	1
1	0	1	1	0	1	0
1	1	0	0	0	0	1
1	1	0	1	0	0	1
1	1	1	0	0	0	1
1	1	1	1	0	1	0

In this table, A<B represents the case where A is less than B, A=B represents the case where A is equal to B, and A>B represents the case where A is greater than B.

NETLIST

```
Vx 2 0 pulse(0 5 0 0 0 10m 20m)
Vy 3 0 pulse(0 5 0 0 0 10m 20m)
Vz 4 0 pulse(0 5 0 0 0 10m 20m)
Vw 5 0 pulse(0 5 0 0 0 10m 20m)
vdd 1 0 dc 5v


.model nmod nmos version=54 level=4.7
.model pmod pmos version=54 level=4.7


m11 6 2 0 0 nmod w=100u l=10u
m12 6 2 1 1 pmod w=100u l=10u


m13 7 3 0 0 nmod w=100u l=10u
m14 7 3 1 1 pmod w=100u l=10u


m15 8 4 0 0 nmod w=100u l=10u
m16 8 4 1 1 pmod w=100u l=10u


m17 9 5 0 0 nmod w=100u l=10u
m18 9 5 1 1 pmod w=100u l=10u


m19 10 5 0 0 nmod w=100u l=10u
m20 11 7 10 10 nmod w=100u l=10u
m21 11 7 1 1 pmod w=100u l=10u
m22 11 5 1 1 pmod w=100u l=10u


m23 12 11 0 0 nmod w=100u l=10u
m24 12 11 1 1 pmod w=100u l=10u


m25 15 6 14 14 nmod w=100u l=10u
m26 14 5 13 13 nmod w=100u l=10u
m27 13 4  0 0 nmod w=100u l=10u
m28 15 6 1 1 pmod w=100u l=10u
m29 15 5 1 1 pmod w=100u l=10u
m30 15 4 1 1 pmod w=100u l=10u


m31 16 15 0 0 nmod w=100u l=10u
```

```
m32 16 15 1 1 pmod w=100u l=10u


m33 19 7 18 18 nmod w=100u l=10u
m34 18 6 17 17 nmod w=100u l=10u
m35 17 4  0 0 nmod w=100u l=10u
m36 19 7 1 1 pmod w=100u l=10u
m37 19 6 1 1 pmod w=100u l=10u
m38 19 4 1 1 pmod w=100u l=10u


m39 20 19 0 0 nmod w=100u l=10u
m40 20 19 1 1 pmod w=100u l=10u


m41 23 12 0 0 nmod w=100u l=10u
m42 23 16 0 0 nmod w=100u l=10u
m43 23 20 0 0 nmod w=100u l=10u
m44 23 20 22 22 pmod w=100u l=10u
m45 22 16 21 21 pmod w=100u l=10u
m46 21 12 1 1 pmod w=100u l=10u


m47 24 23 0 0 nmod w=100u l=10u
m48 24 23 1 1 pmod w=100u l=10u


.tran 0.1m 100m
.control
run
set color0 = white
set color1 = black
set xbrushwidth =4.5
plot V(24)  title "s1"
plot V(2)  title "a"
plot V(3)  title "b"
plot V(4)  title "c"
plot v(5)   title "d"

.end
```

NgSpice OUTPUT

NGSPICE is an open-source circuit simulator that can be used for simulating and analyzing electronic circuits. It can be used for both analogue and digital circuits, as well as for mixed-signal circuits.

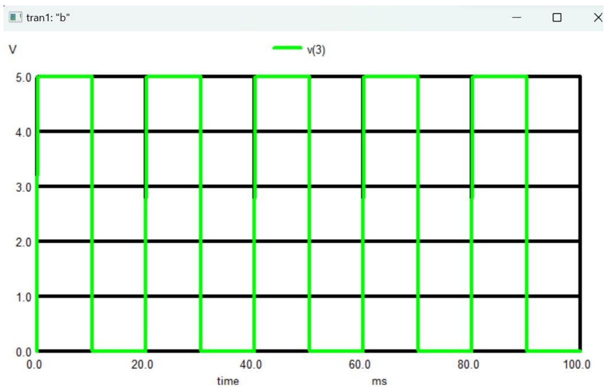


Figure 2.1 Input Graph for a0

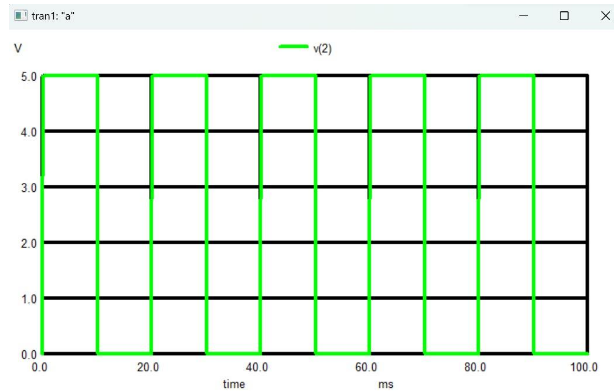


Figure 2.2 Input Graph for a1

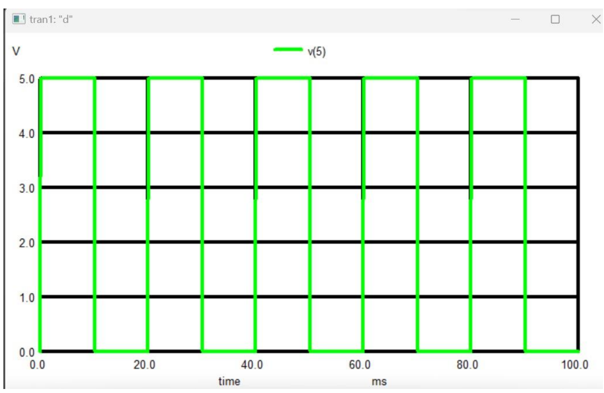


Figure 2.3 Input Graph for b1

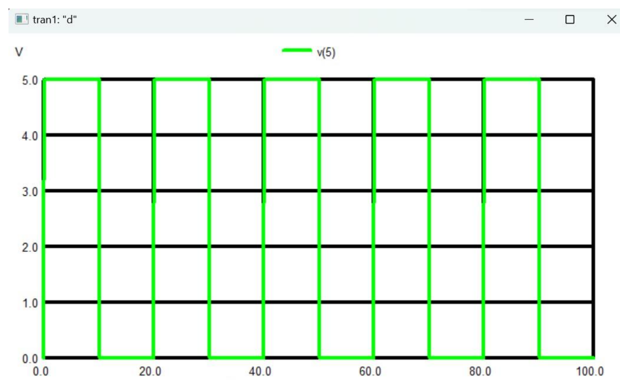


Figure 2.4 Input Graph for b0

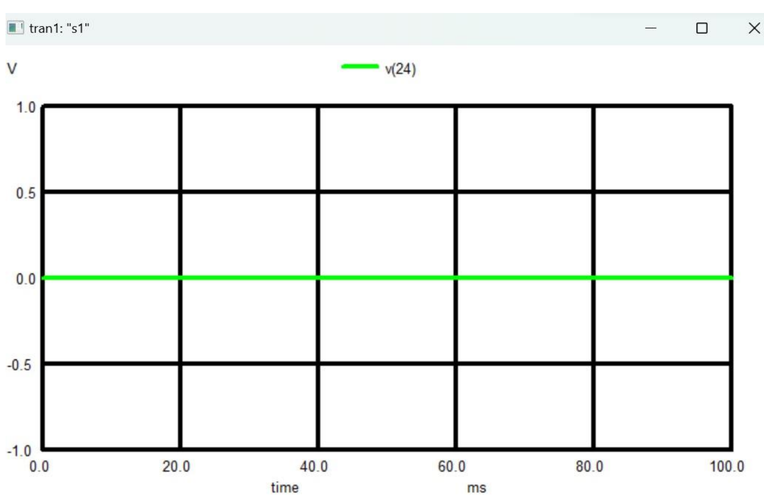


Figure 2.5 OUTPUT GRAPH

NOTE:
** The given output is true for the first and the last entry of the truth table as the output for A<B is low (0) in both the cases.*
** Further we can analyse the circuit just by changing the A0 A1 and B0 B1 as DC inputs. While repeating it several times till we get all the inputs.*

SCHEMATICS

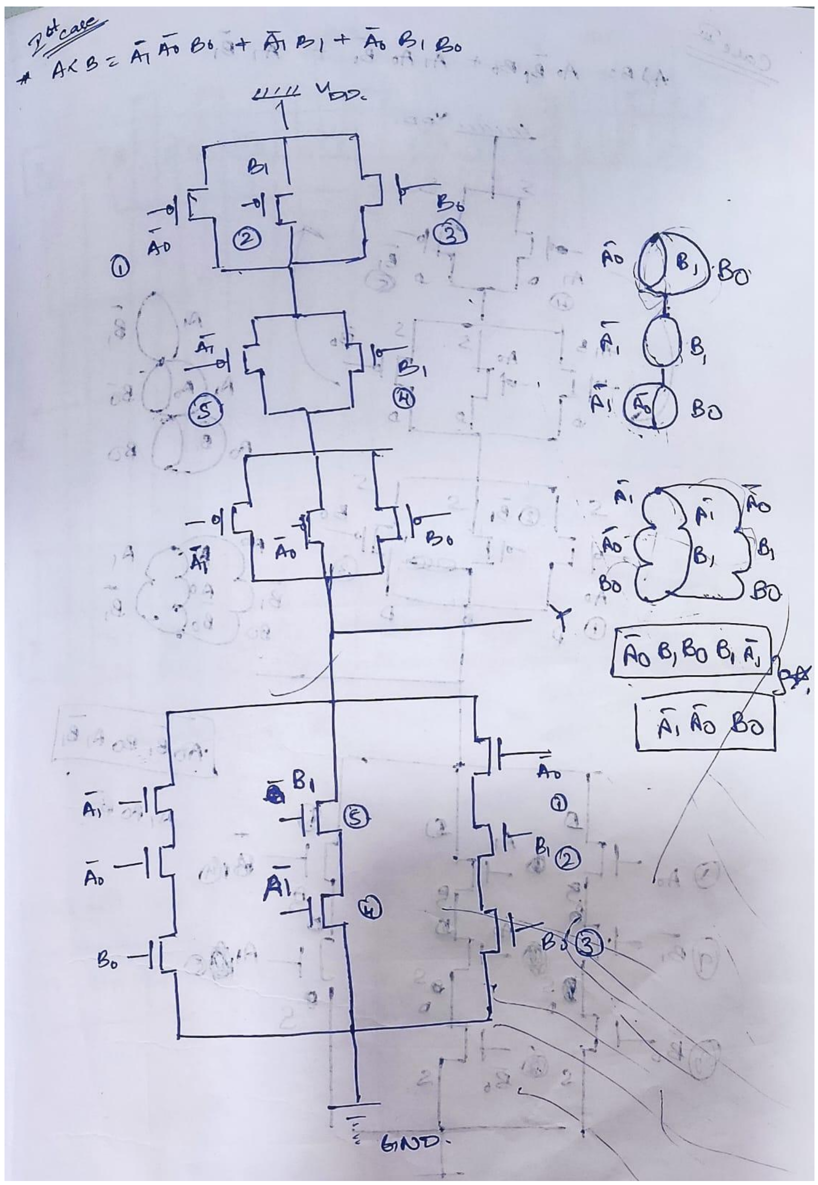


Figure 3.1

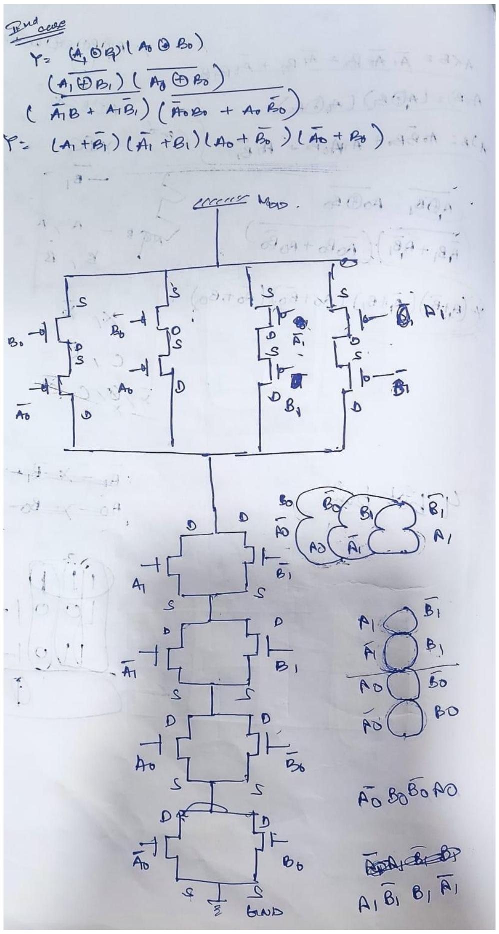


Figure 3.2

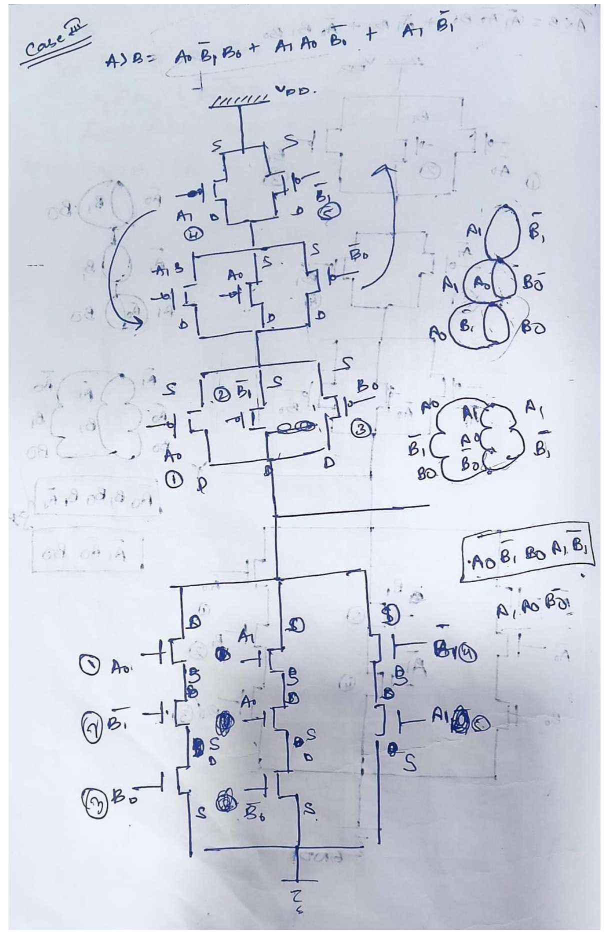


Figure 3.3

LAYOUT DIAGRAM

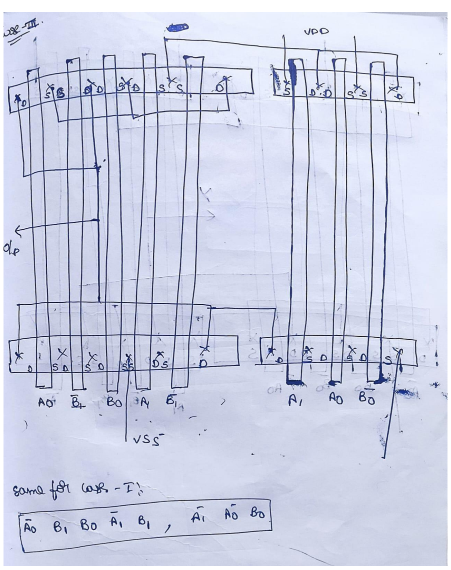


Figure 4.1

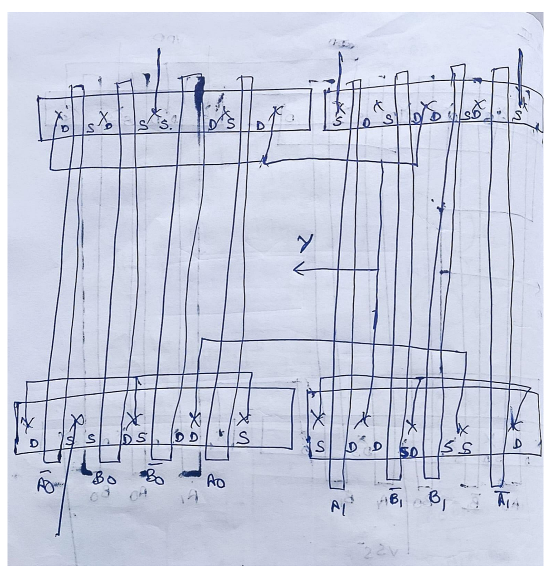


Figure 4.2

MICROWIND LAYOUT

Microwind is a powerful and flexible tool for designing and simulating digital and analog circuits. It can be useful for both professionals and students and can help to streamline the circuit design process and improve circuit performance.

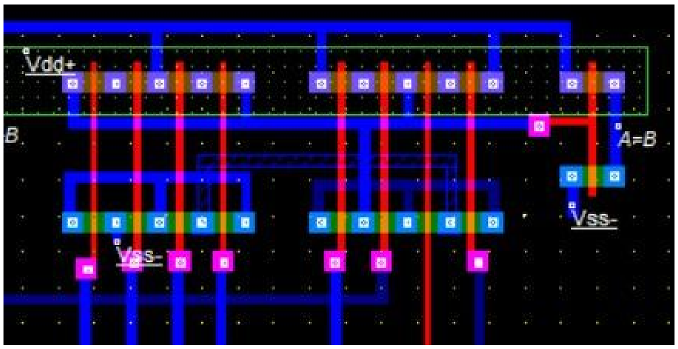


Figure 5.1

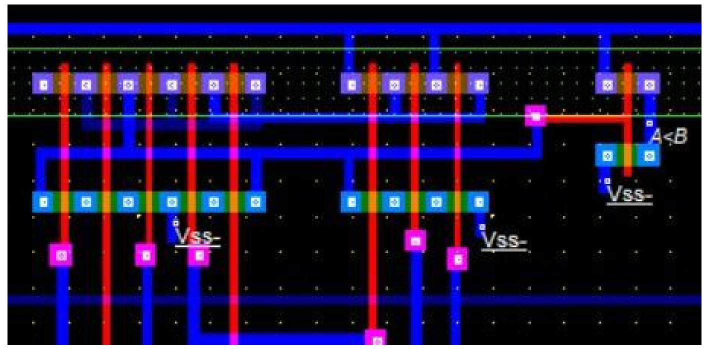


Figure 5.2

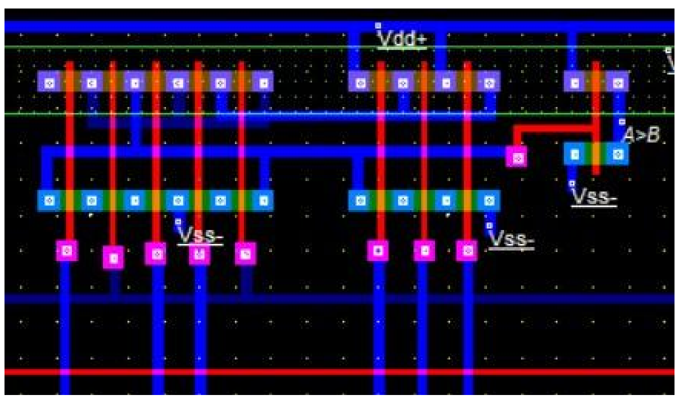


Figure 5.3

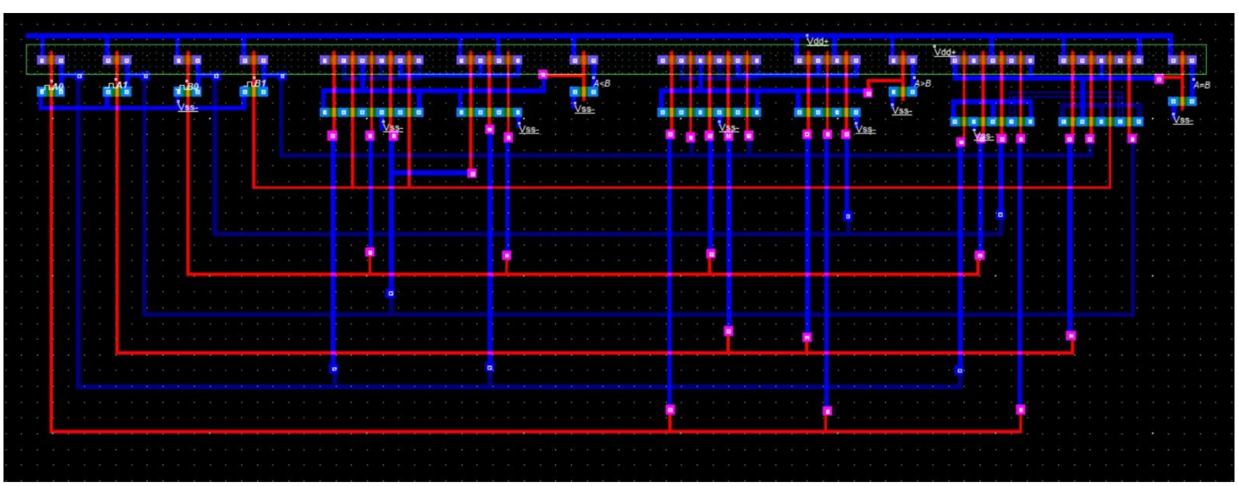
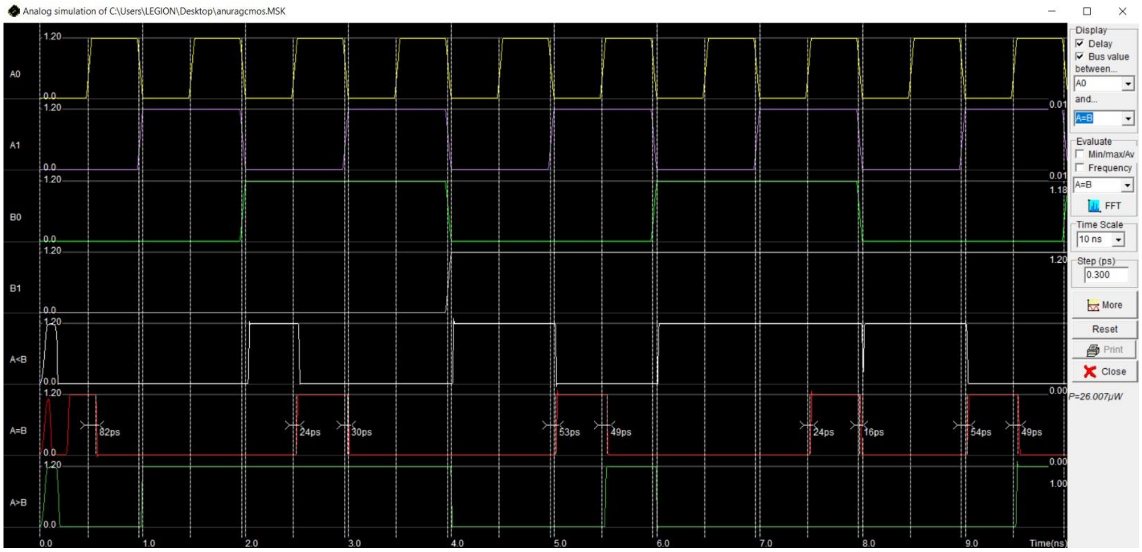


Figure 5.4

LAYOUT OUTPUT



CONCLUSION

Based on the design and simulation of a two-bit comparator using Microwind, it can be concluded that the comparator functions effectively and accurately. The project successfully achieved its objectives, which included designing a comparator using Microwind, simulating its operation, and analyzing its performance.

The methodology involved creating a schematic of the two-bit comparator, which was then simulated using Microwind to verify its functionality. The design was optimized to achieve the desired performance specifications, such as speed and power consumption. Although there were some challenges encountered during the project, such as optimizing the design to meet the required specifications, these were successfully addressed.

The results of the project demonstrated that the two-bit comparator designed using Microwind operates with high accuracy and speed while consuming minimal power. The simulation results showed that the comparator was able to accurately compare two-bit binary numbers, producing the expected output.

In conclusion, the design and simulation of a two-bit comparator using Microwind were successful, demonstrating that Microwind can be a useful tool for designing and simulating digital circuits. The project provides a foundation for future research on optimizing and improving the performance of digital circuits.