Low-Power 4X4 Magnitude Comparator Using NAND-Inverter Logic For Modular Design

Dana Ghnimat [1], Maryan Qassis [2], Tareq Ewaida [3]
Department of Electrical and Computer Engineering, Birzeit University [1,2,3]
1200031@student.birzeit.edu [1], 1200861@student.birzeit.edu [2], 1192363@student.birzeit.edu [3]

***Abstract*—This paper aims to construct a 4X4 Magnitude comparator using NAND-Inverter Logic which is similar to Universal Gate logic but with the added use of inverters. The design strategy focuses on optimizing power efficiency and minimizing the comparator cell's area, to achieve a compromise between low power use and minimal delay. The design of this comparator starts with constructing the NAND and Inverter gates which are used to create 1-bit Comparator first which is used to construct the higher order comparators. Schematics and layouts are designed for each comparator using Electric EDA tool,**

Keywords—Low-Power, Comparator, Universal Gate, 2-bit Comparator, 4-bit Comparator, Modular Design

I. Introduction

Magnitude Comparators are digital circuits that compare two binary words, the output of a comparator indicates weather a word is greater than the other or if they're equal. Comparators are important in various applications, from CPU ALUs to Microcontrollers and DSP applications. All the applications mentioned require very fast components but in the last 20 years with computing devices becoming more mobile, and with the increase of transistors per chip, there has developed a real need for low power components to make the overall chips or devices less power dissipating. The aim of this project is to make magnitude comparators of various sizes starting at 1-bit comparators and using them in a modular design to construct the higher order comparators. The designs are being optimized for area, speed and power.

II. Design and Implementation

A. Theoretical Background

A magnitude digital comparator is a type of combinational circuit used to compare two binary or digital values to determine if they are more, equal, or less than each other. It makes sense for us to create a circuit with two inputs, one for A and one for B, and three output terminals: one for A > B, one for A = B, and one for A < B.

1) 1-bit Comparators are the building blocks for are all higher order comparators, they compare between only two binary bits A and B, where the outputs are:

- **Equal Condition**: A=B
- **Greater Condition**: A>B
- **Less Condition**: A<B

2) Multi-bit Comparators compare words of more than one bit, the comparason happens bit by bit from the LSB, if all are equal then they're equal, the bits that differ determine which word is greater. The process isn't sequencial it is implemented by combinational logic circuits.In this project these circuits are made modularly from lower size comparators starting at the 1-bit comparator.

N – bit Comparator

Beginning with the most significant bit (MSB) and working toward the least significant bit (LSB), the circuit compares the bits of the two values. The two equivalent bits of each integer are compared at each bit location. The A>B output is set to 1 and the circuit instantly detects that the first number is bigger than the second if the bit in the first number is greater than the equivalent bit in the second number. Similar to this, the circuit instantly identifies that the first number is smaller than the second if the bit in the second number is bigger than the corresponding bit in the first number and sets the A\B output to 1. The circuit advances to the next bit position and compares the following pair of bits if the two corresponding bits are equivalent. This procedure keeps going till every piece has been contrasted. The comparison is stopped and the appropriate output is produced if the circuit at any time throughout the comparison finds that the first number is higher or less than the second number.
The circuit produces an A=B output, which indicates that the two numbers are equal, if all the bits are equivalent.

3) NAND Logic

NAND gates are considered Universal Gates because they can be used to construct any other logic gate, but adding inverters to our designs makes the design process more flexible and reduces gate and transistor count in many cases where a simple inverter suffices. The main advantages of this approach are that it leads to simpler, more minimized designs, which can improve performance and reduce power consumption. It's also more reliable since only two gate constructions are used. Logic gates are small digital switching circuits that use two or more inputs Boolean functions to determine their binary output. Logical 0 denotes

False or Low in nature, whereas Logical 1 denotes True or High in nature. The result varies depending on the logical

B. *Logic Design*

1) *1-bit Comparator logic design*

We used a NAND implementation of XOR and modified it with Inverters to get a simple 1-bit Comparator

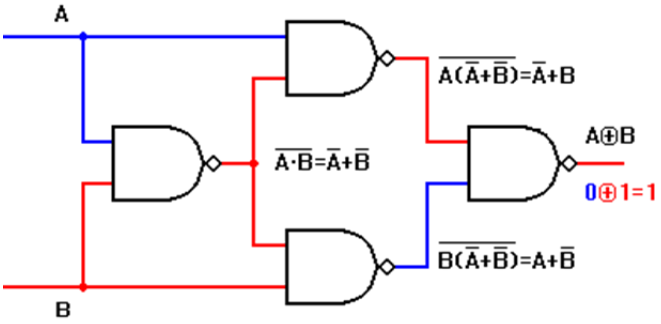

Fig. 1. XOR NAND implementation

The following modifications were made to the circuit in Fig.1

- To obtain A > B its simply (A'+B)' = A. B' using an inverter
- To obtain A < B, its (A+B')' = A'. B using an inverter
- To obtain A=B, we use (A ⊕ B)' which is (AB' + A'B)' , which can be written with NAND. Which was extracted from the following truth table:

Table 1 1-bit comparator

Inputs		Outputs		
A	***B***	***A>B***	***A=B***	***B>A***
0	0	0	1	0
0	1	0	0	1
1	0	1	0	0
1	1	0	1	0

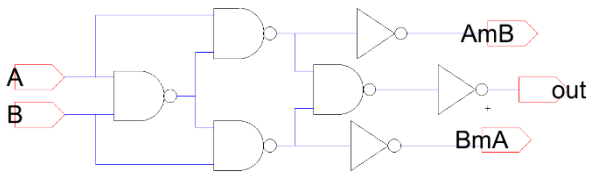

Fig. 2. Modified From Fig.1

2) *2-bit Comparator Logic Design*

The 2-bit comparator was constructed from 2 1-bit modules.

- A > B => $A_1B_1' + A_0B_1'B_0' + A_1A_0B_0'$
- A < B => $B_1A_1' + B_0B_1A_0' + A_1'A_0'B_0$
- A = B => $(A_0 \oplus B_0)(A_1 \oplus B_1)$

Using the 1-bit comparator, adding one NAND and an inverter creates the AND gate to connect the two Equal from the 1-bit Comp gates, gets the equal output for the 2-bit Comp.

- $A > B = A1 > B1\ or\ (A1 = B1\ and\ A0 > B0)$.
- $A = B = (A1 = B1\ and\ A0 = B0)$
- $B > A = B1 > A1\ or\ (A1 = B1\ and\ B0 > A0)$

Table 2 2-bit Comparator Truth Table

Inputs				Outputs		
A1	***A0***	***B1***	***B0***	***A>B***	***A=B***	***B>A***
0	0	0	0	0	1	0
0	0	0	1	0	0	1
0	0	1	0	0	0	1
0	0	1	1	0	0	1
0	1	0	0	1	0	0
0	1	0	1	0	1	0
0	1	1	0	0	0	1
0	1	1	1	0	0	1
1	0	0	0	1	0	0
1	0	0	1	1	0	0
1	0	1	0	0	1	0
1	0	1	1	0	0	1
1	1	0	0	1	0	0
1	1	0	1	1	0	0
1	1	1	0	1	0	0
1	1	1	1	0	1	0

NANDs and Inverters were used to construct the AND and OR gates.

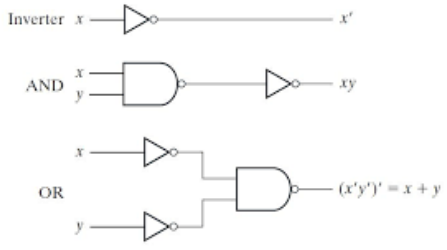

Fig 3. Gates Constructed with NAND and Inverter

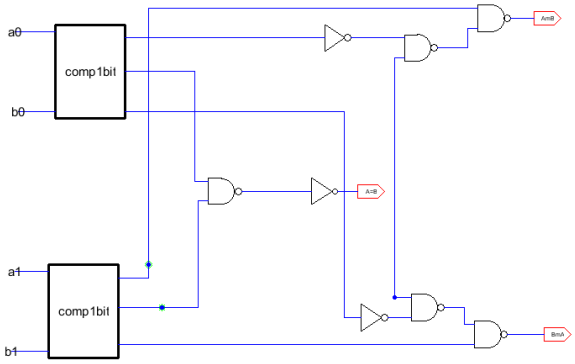

Fig. 4. 2-bit Comparator Logic Design

3) *4-bit Comparator Design*

Table 3 4-bit Comparator Truth Table

Inputs				Outputs		
A3, B3	***A2, B2***	***A1, B1***	***A0, B0***	***A>B***	***B>A***	***A=B***
A3>B3	X	X	X	1	0	0
A3<B3	X	X	X	0	1	0
A3=B3	A2>B2	X	X	1	0	0
A3=B3	A2<B2	X	X	0	1	0
A3=B3	A2=B2	A1>B1	X	1	0	0
A3=B3	A2=B2	A1<B1	X	0	1	0
A3=B3	A2=B2	A1=B1	A0>B0	1	0	0
A3=B3	A2=B2	A1=B1	A0<B0	0	1	0
A3=B3	A2=B2	A1=B1	A0=B0	0	0	1

(A=B):

A3=B3 AND A2=B2 AND A1=B1 AND A0=B0
EQ3=¬(¬(¬(A3∧B3) ∧B3) ∧¬(¬(A3∧B3) ∧A3))
EQ2=¬(¬(¬(A2∧B2) ∧B2) ∧¬(¬(A2∧B2) ∧A2))
EQ1=¬(¬(¬(A1∧B1) ∧B1) ∧¬(¬(A1∧B1) ∧A1))
EQ0=¬(¬(¬(A0∧B0) ∧B0) ∧¬(¬(A0∧B0) ∧A0))
(A=B) = EQ3 AND EQ2 AND EQ1 AND EQ0

(A>B):

A3>B3
GT3=¬(¬(¬(A3∧B3) ∧A3))
A3=B3 AND A2>B2
GT2=¬(¬(¬(A2∧B2) ∧A2))
A3=B3 AND A2=B2 AND A1>B1
GT1=¬(¬(¬(A1∧B1) ∧A1))
A3=B3 AND A2=B2 AND A1=B1 AND A0>B0
GT0=¬(¬(¬(A0∧B0) ∧A0))
(A>B) = GT3 OR (EQ3 AND GT2) OR (EQ3 AND EQ2 AND GT1) OR (EQ3 AND EQ2 AND EQ1 AND GT0)

(B>A):

B3>A3
LT3=¬(¬(¬(A3∧B3) ∧B3))
B3=A3 AND B2>A2
LT2=¬(¬(¬(A2∧B2) ∧B2))
B3=A3 AND B2=A2 AND B1>A1
LT1=¬(¬(¬(A1∧B1) ∧B1))
B3=A3 AND B2=A2 AND B1=A1 AND B0>A0
LT0=¬(¬(¬(A0∧B0) ∧B0))
(B>A) = LT3 OR (EQ3 AND LT2) OR (EQ3 AND EQ2 AND LT1) OR (EQ3 AND EQ2 AND EQ1 AND LT0)

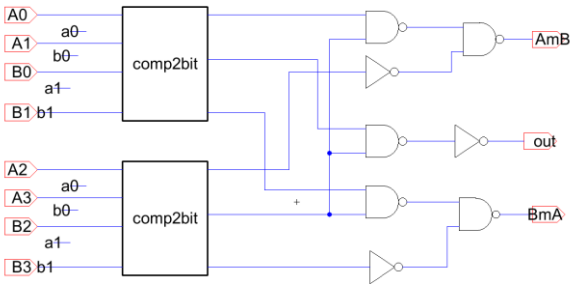

Fig. 5. 4-bit Comparator Logic Design

C. *Schematics*

1) *NAND Schematic*

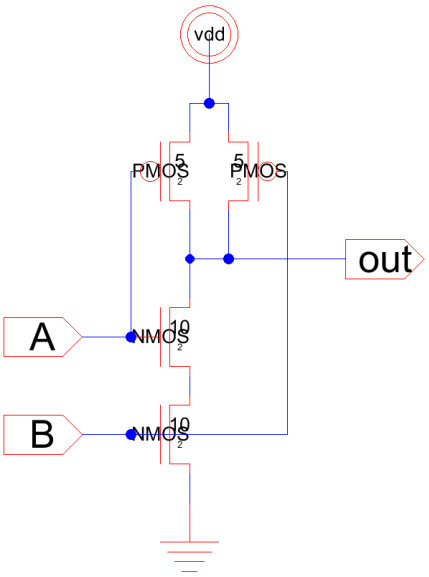

Fig. 6. NAND Schematic

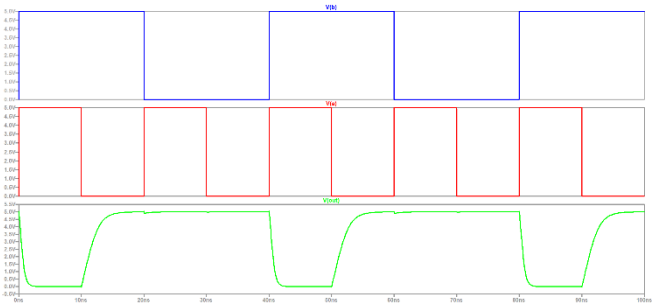

Fig. 7. NAND Simulation

2) Inverter Schematic

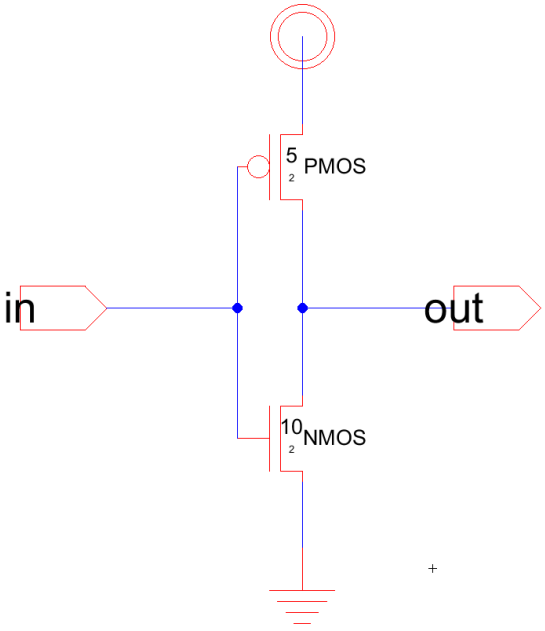

Fig. 8. Inverter Schematic

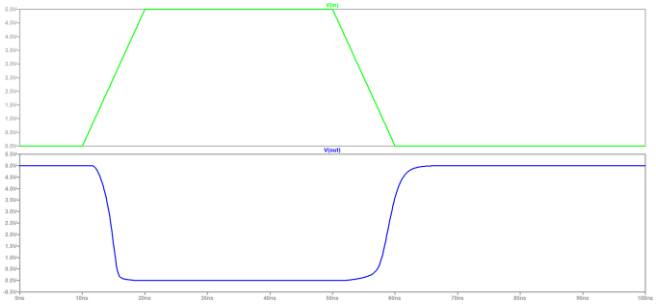

Fig. 9. Inverter Simulation

3) 1-bit Comparator Schematic

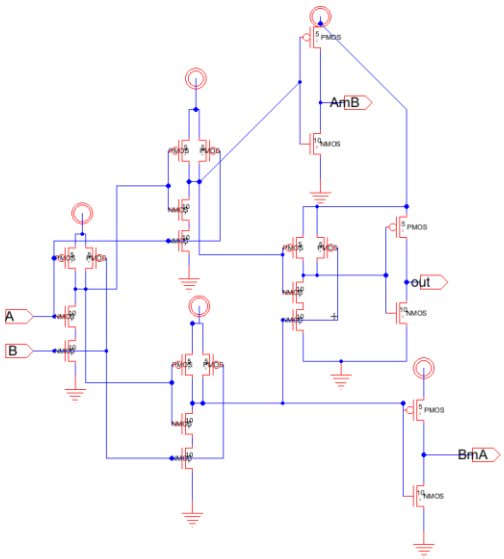

Fig.10. 1-bit Comparator Schematic

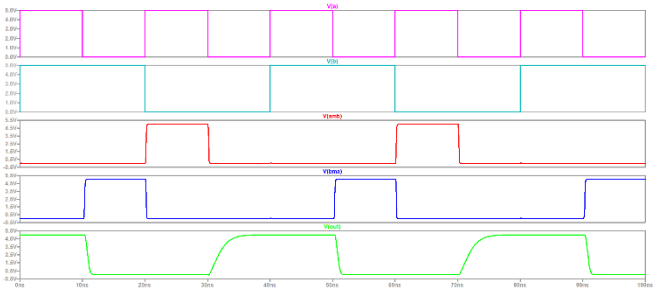

Fig. 11. 1-bit Comparator Simulation

4) 2-bit Comparator Schematic

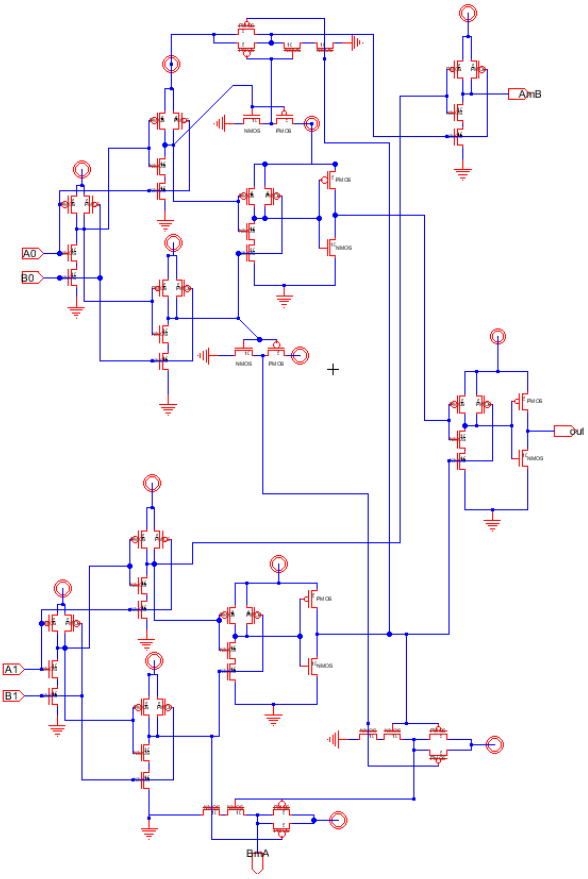

Fig. 12. 2-bit Comparator Schematic

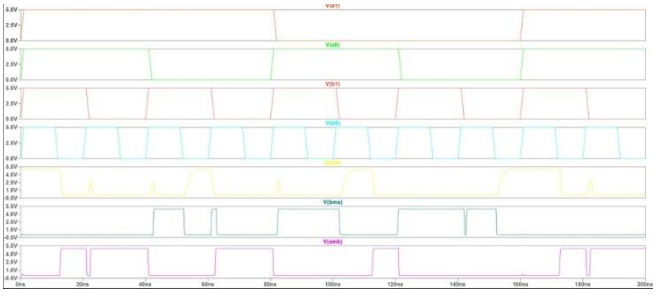

Fig.13. 2-bit Comparator Simulation

5) 4-bit Comparator Schematic

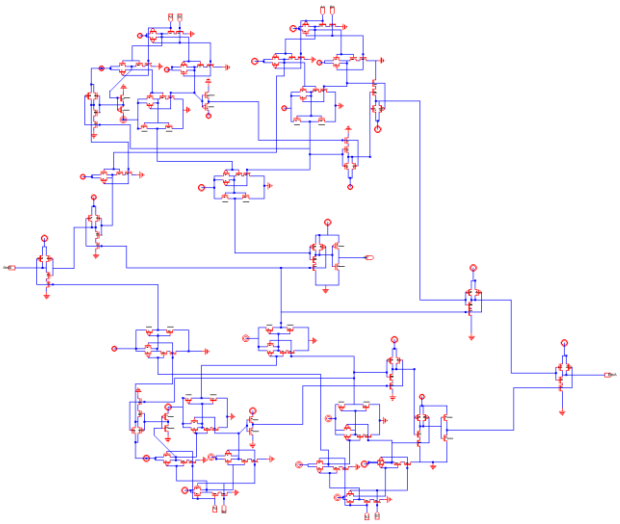

Fig. 14. 4-bit Comparator Schematic

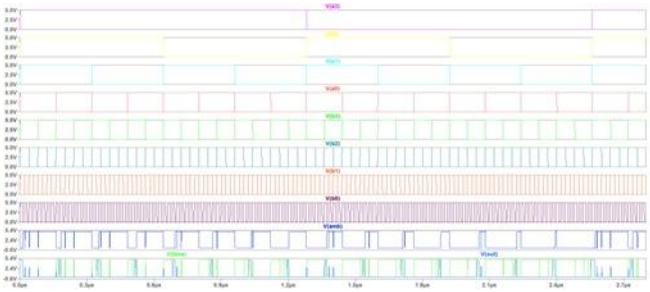

Fig. 15. 4-bit Comparator Simulation

D. Layouts

1) NAND Layout

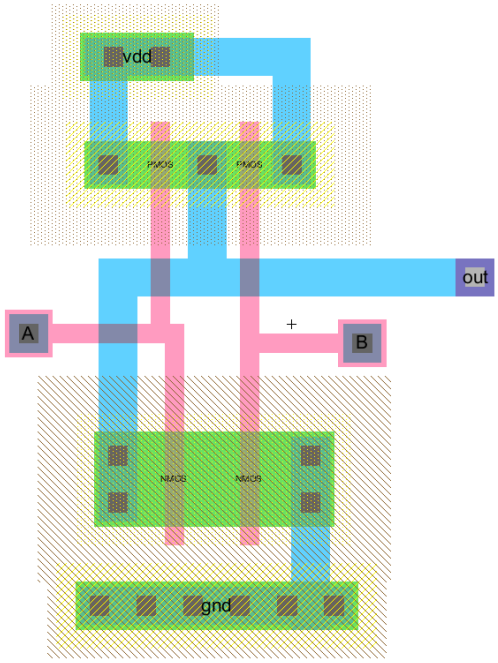

Fig. 16. NAND Layout

2) Inverter Layout

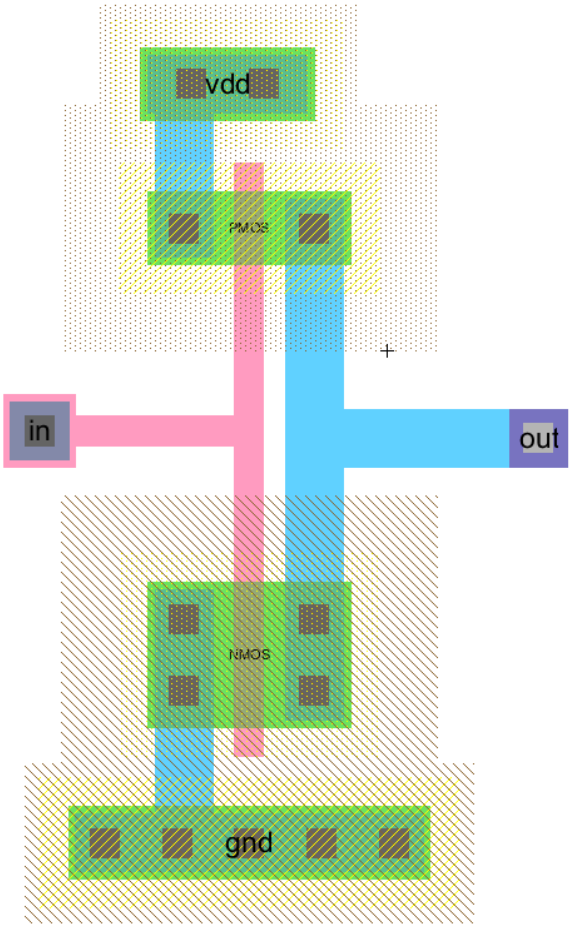

Fig. 17. Inverter Layout

3) 1-bit Comparator Layout

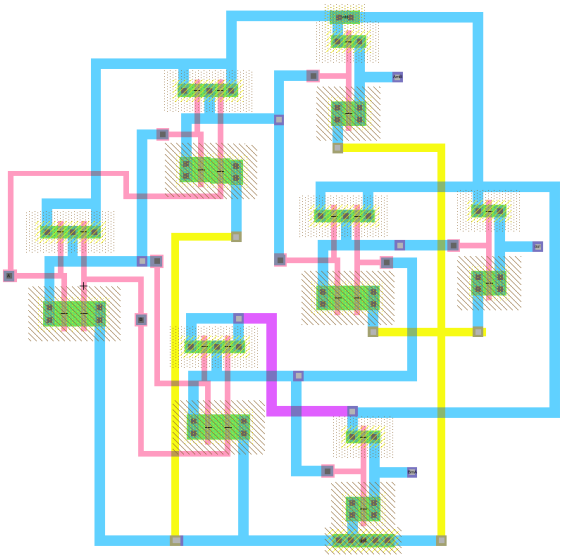

Fig. 18. 1-bit Comparator Layout

4) 2-bit Comparator Layout

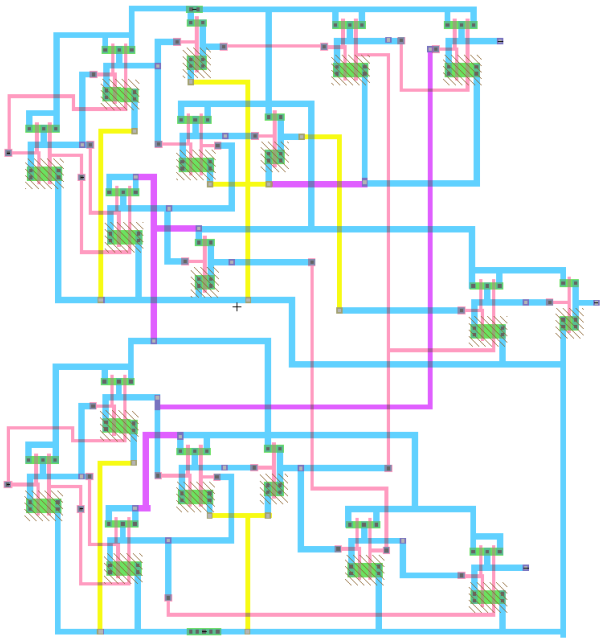

Fig. 19. 2-bit Comparator Layout

III. Results

Table 4 Simulation results

No	Parameter	1-bit Comp.	2-bit Comp	4-bit Comp Schm
1	***Delay* (ns)**	5.873 *ns*	7.161 *ns*	6.995 *ns*
2	***Area* (μm^2)**	228 x 231.5 μm^2	493 x 476 μm^2	307 x 260
3	***Power* (μW)**	0.1 μW	0.2 μW	2.8 μW

From simulation results, its noticed that the power in each comparator is still low, with a low delay in nanoseconds, that's due many factors one of them the metals that has been used to create the layout of each comparator except the 4-bit comparator, and it is noticed how small the 4-bit comparator, and it also have a lower delay because it only contain the wires and the cmos, while 2-bit and 1-bit comparator contain other metals adding up to the delay.

IV. Conclusion

To summarize, the entire project is about creating low-power 4x4 Magnitude Comparator which the use of NAND-Inverter Logic guarantees the flexibility and modularity of the design. The design plan mainly focused on the improvement of power efficiency as well as decreasing the size of the comparator cell, thus maintaining the minimum power loss and delay. In this project the designing and simulating of 1-bit, 2-bit, and 4-bit comparators were carried out, and the diagrams prepared by the Electric EDA tool were accurate as well as the layouts. It is clearly visible through the output of the comparator that the system has been tuned on such metrics as delay, area, and power consumption; this shows how the NAND-Inverter logic has become for low-power applications. This approach is made possible by the modular design methodology which introduces predictable solutions for scaling. As a result, the design is convenient for larger digital systems in which they need magnitude comparison in an efficient manner.

V. Future Work and Improvements

The design and implementation of the 4x4 magnitude comparator using NAND-Inverter logic have demonstrated significant power efficiency and modularity benefits. Future improvements could involve scaling the comparator to higher bit-widths, such as 8-bit or 16-bit, while optimizing interconnections for minimal delay and power consumption. Additionally, exploring advanced power reduction techniques like dynamic voltage and frequency scaling (DVFS), power gating, and clock gating could further enhance efficiency. Investigating the impact of process variations on performance and developing robust strategies against manufacturing and environmental fluctuations will be crucial. Comparative analysis with other comparator designs using different logic styles will help identify strengths and guide future enhancements, ensuring the comparator's applicability in various high-performance, power-efficient digital systems.

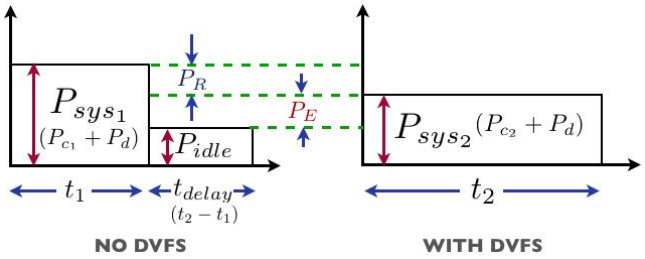

Figure 20 Power Consumption and Execution Times

References

[1] "Comparators - Digital Electronics," Testbook, [Online]. Available: https://testbook.com/digital-electronics/comparators. [Accessed: 13-Jun-2024].

[2] https://www.geeksforgeeks.org/magnitude-comparator-in-digital-logic/ [Accessed: 15-Jun-2024]

[3] "Gate Level Implementation - Digital Electronics (Part 8)," Engineers Garage, [Online]. Available: https://www.engineersgarage.com/gate-level-implementation-de-part-8. [Accessed: 13-Jun-2024].

[4] "Analysis of Dynamic Voltage Scaling for System Level Energy Management"[Online]Available:https://www.usenix.org/legacy/event/hotpower08/tech/full_papers/dhiman/dhiman_html/index.html [Accessed: 14-Jun-2024].