Design and Implementation of a 16x16-Bit Comparator Using Transmission Gates

Electric and Computer Engineering Department

Birzeit University, Ramallah, Palestine

Group Members :

Malak Moqbel 1210608 Noor Moadi 1211080 Hazar Michael 1201838

Abstract

The aim of this project is using VLSI application to design and implement a 16x16 bit comparator using CMOS Transmission gate method that describes several logic families used in the design of the comparator integrated circuit ,it also used to reduce the count of the transistors to implement the logic gates. Which typically leads to improve the speed, compact size , area ,cost, and power consumption.thus,In this project we use 0.6u technology,The transmission gate technique makes circuits work better by allowing smoother and more efficient switching, which reduces the chance of losing signals. This approach not only speeds up signal transmission but also makes the circuit more reliable. It's a smart way to cut down on power usage and shrink the size of the design, making it a great choice for projects that need to balance high performance with saving space.

KeyWords: VLSI Design, CMOS , 16x16 bit Comparator , speed, size , power consumption.

1. Introduction

A comparator is an electronic circuit that takes in two input signals and compares them to see how they relate to each other. It then produces an output that tells whether one input is equal to, greater than, or less than the other. .

1.1. Transmission Gate

A Transmission Gate is like a smart switch made with CMOS technology. It decides when to let a signal pass through or block it based on a control signal. When the control signal is low, the gate opens, allowing the input signal to flow to the output. But when the control signal is high, it closes the gate, isolating the input from the output and making the output "silent" or high impedance. The NMOS part of the gate is great at sending a strong '0', and the PMOS part is perfect for sending a strong '1'. Together, they ensure that signals pass through cleanly without losing strength or getting distorted.

In a 16-bit comparator, these Transmission Gates are essential for managing how signals move through the circuit. They make sure that both '0' and '1' signals stay clear and accurate, which is crucial for the comparator to work properly. By reducing any potential distortion, these gates help the comparator operate more smoothly, with less delay and lower power use. This means the comparator can compare signals faster and more efficiently, making the whole process more reliable.

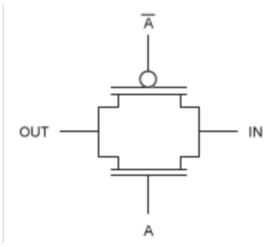


Figure 1

1.2. Comparators

Digital comparators, also known as binary or logic comparators, are essential components in many electronic devices, particularly within the arithmetic units of portable devices. These comparators compare two binary numbers to determine whether they are equal.

In CMOS technology, the operation of NMOS and PMOS transistors plays a crucial role. NMOS transistors are excellent at passing a strong logic 0 (low voltage) but struggle to pass a strong logic 1 (high voltage). Conversely, PMOS transistors pass a strong logic 1 effectively but are less efficient at passing a strong logic 0. This complementary behavior enables CMOS technology to handle both low and high voltages effectively, making it a better choice for circuits requiring both speed and efficiency.

Our approach involves designing various logic gates using CMOS technology and transmission gates. Transmission gates are particularly useful because they control the flow of signals, allowing them to pass through easily when needed and blocking them when not, thereby reducing power consumption and increasing circuit speed.

The complexity of a circuit in CMOS technology depends on the number and size of transmissions and wiring, which are determined by the number of connections and their lengths. As VLSI technology advances, the focus has been on creating systems that fit on a single chip, where comparators play a key role. These components ensure that electronic devices can make fast and accurate decisions, contributing to their overall proper functioning.

2. Design and Implementation

In this section, we'll discuss the key logic gates used to implement a 16x16 bit comparator using transmission gates, along with the simulation and layout considerations to ensure optimal performance.

2.1. XOR Implementaion

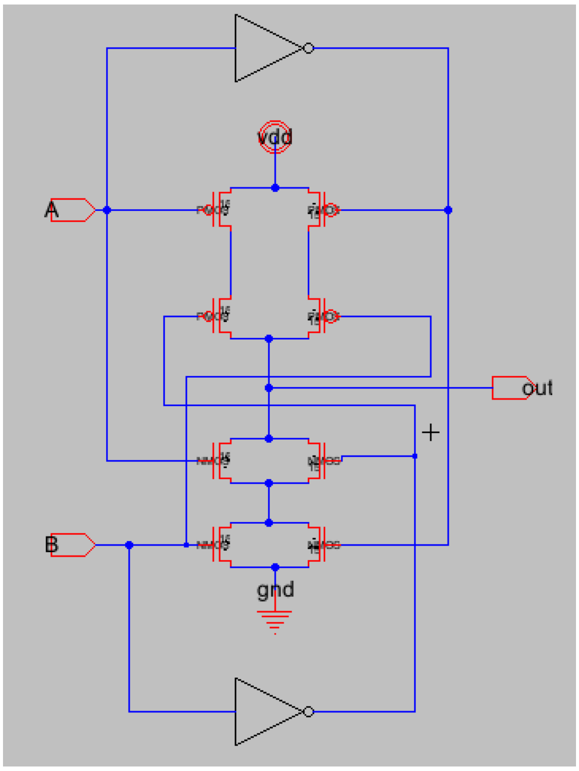


Figure 2 XOR Schematic

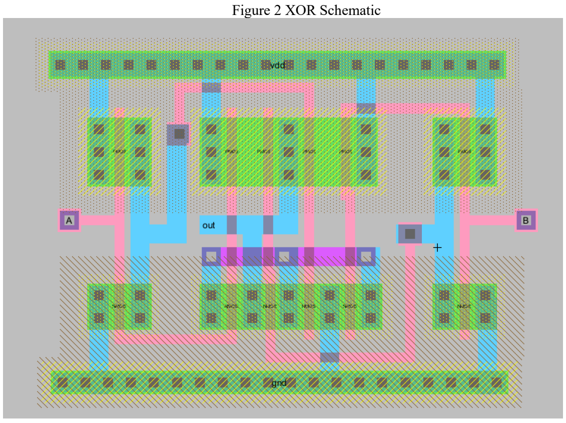


Figure 3 XOR Layout

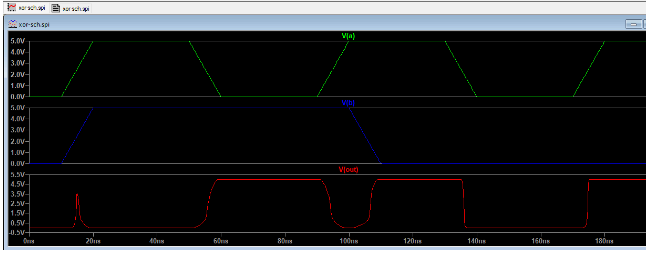


Figure 4 XOR simulation

The XOR gate that we implement has 2 inputs A and B , when they have different values then the output value is 1 , otherwise the output is 0 , the invertors are used to ensure proper signal inversion, allowing the transistor to switch on and off correctly to produce the XOR logic.

As shown in figure 4 The output delay occurs because the transistors need a moment to respond to input changes, which is normal in digital circuits.

2.2. Invertor Implementation

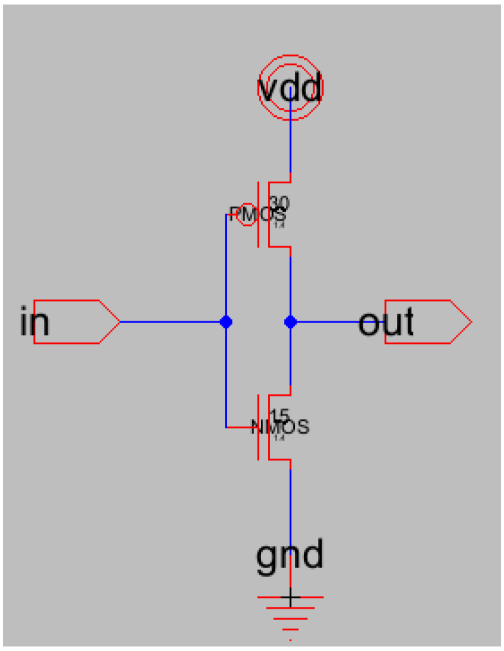


Figure 5 Invertor Schematic

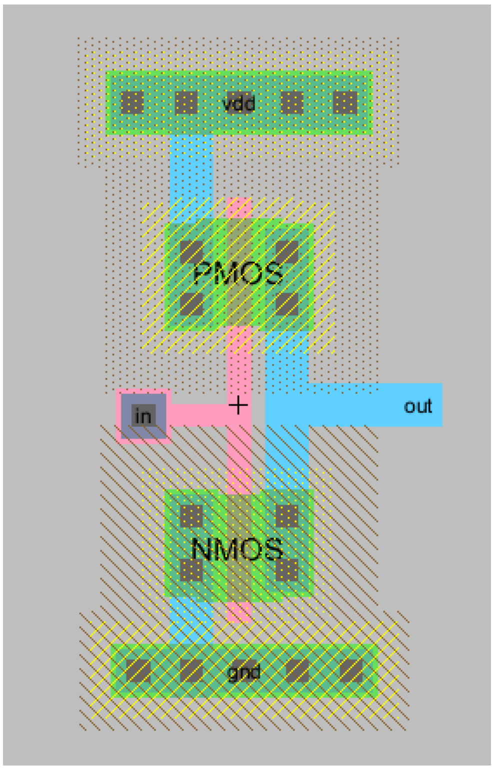


Figure 6 Invertor Implementation

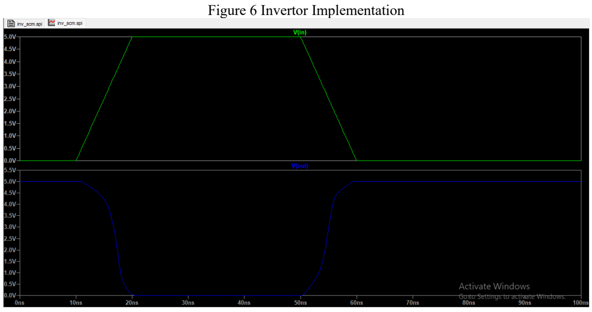


Figure 7 Invertor simulation

We designed an invertor using 1 PMOS with width of 30 and 1 NMOS with width of 15, the length is about 1.4.

The operation od the NMOS and PMOS transistoes depends on the input voltage , if the input voltage is 1 then the PMOS is off and NMOS turns on , so the output voltage is pulled down to the gnd (zero) , in the other hand , of the input voltage is zero ,then PMOS turns on and NMOs turns off , so the output voltage is ulled to vdd (1)

2.3. 4-input NAND Implementaion

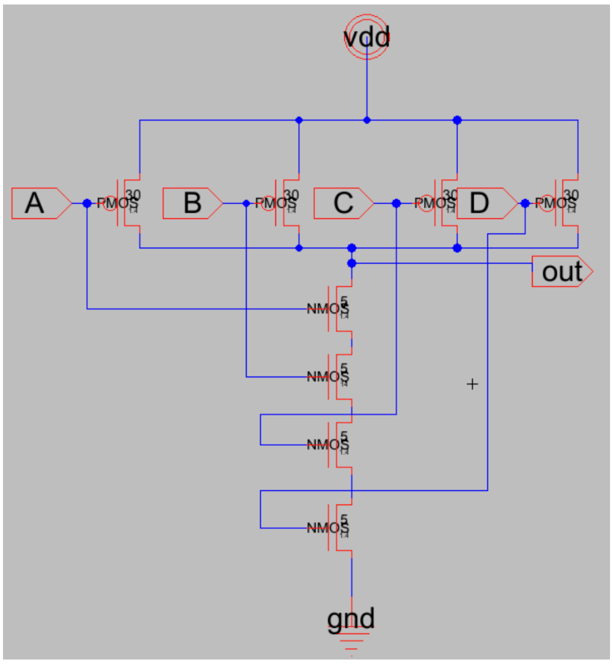


Figure 8 4-input NAND Schematic

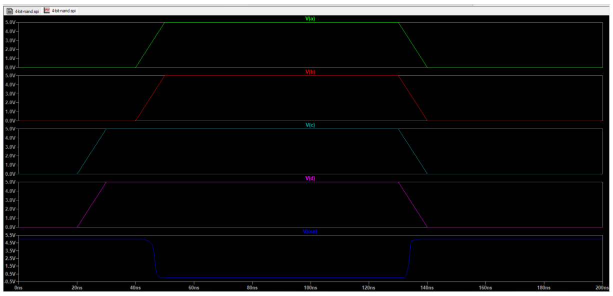


Figure 9 4-input NAND simulation

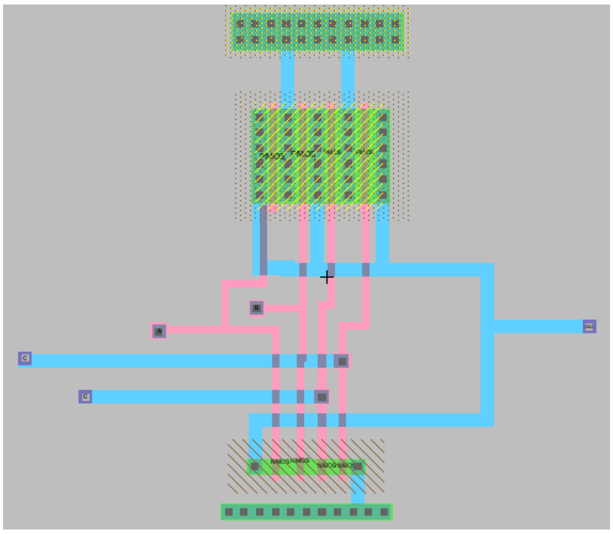


Figure 10 4-Input NAND Layout

In this part, We implement a 4-input NAND gate , by setting the new NMOS and PMOS size to 15 and 30, and the length to 1.4 As shown in figure 9 , the simulation of this gate is correct (when all the inputs are 1 the output value is 0)

2.4. 4-input AND Implementaion

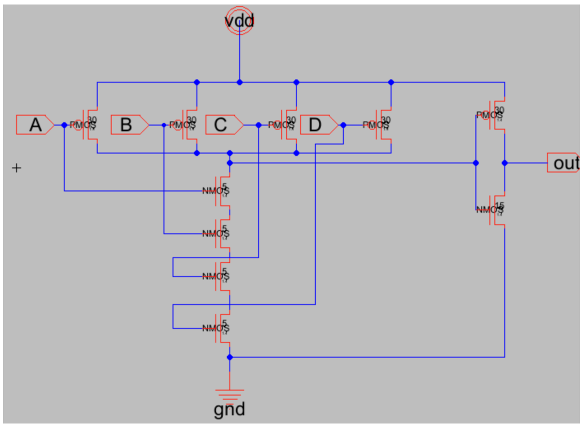


Figure 11 4-input AND Schematic

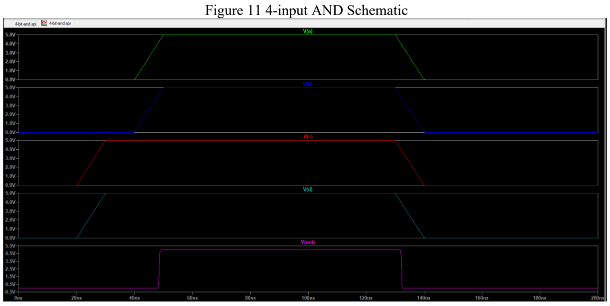


Figure 12 4-input AND Simulation

We designed a 4-input AND gate using a combination of NAND gates and inverters from a previous design. This gate plays a crucial role in verifying equality within the 16-bit comparator we're developing.

2.5. 1-bit Comparator Implementaion

we designed a 1-bit comparator using transmission gates with a width of 30 and 15, and a length of 1.4, implemented in the Electric VLSI tool. The schematic, shown in Figure bellow, depicts how the comparator takes two 1-bit inputs (A and B) and produces three outputs: "awin" , "bwin" , and "Equal" . If the awin and bwin have the same value then A and B are equal , otherwise if awin is 1 then A is greater than B , and if bwin is 1 then B is greater than A. Transmission gates in this design ensure fast, accurate comparisons with minimal delay.

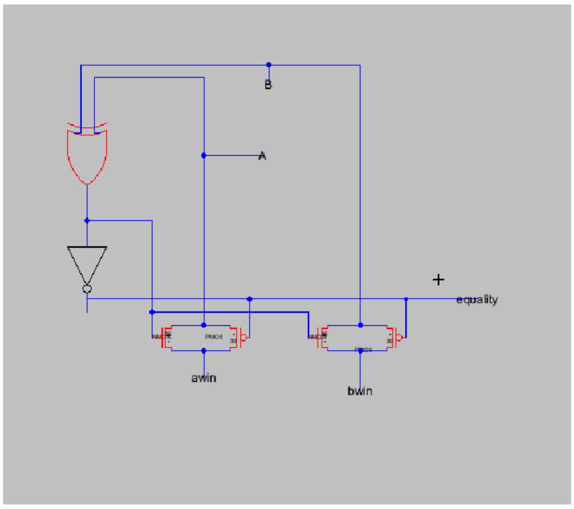


Figure 13 1-bit Comparator

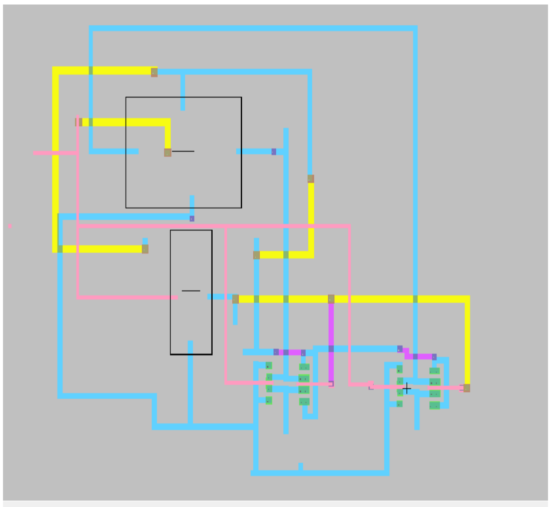


Figure 14 1-bit Comparator Layout

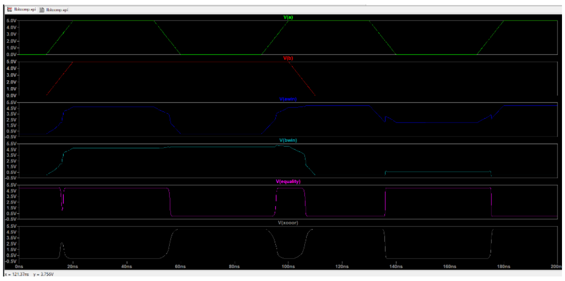


Figure 15 1-bit Comparator Simulation

As shown in figure 15 , when A and B are equal then (aout) and (bout) values are Don't Care , but V(Equal) Value is 1.

Table 1 1-Bit Comparator truth table

A	B	awin	bwin	Equal
0	0	X	X	1
0	1	0	1	0
1	0	1	0	0
1	1	X	X	1

2.6. 4-bit Comparator Implementaion

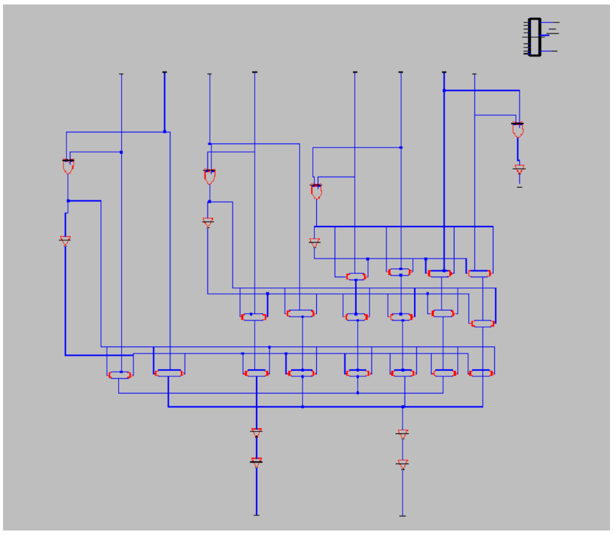


Figure 164-bit Comparator schematic

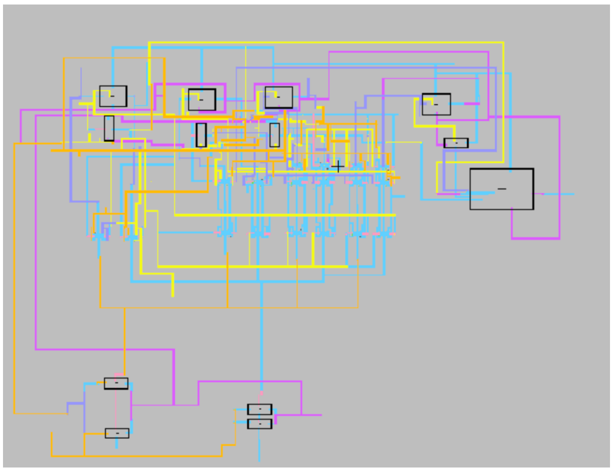


Figure 17 4-bit Comparator layout

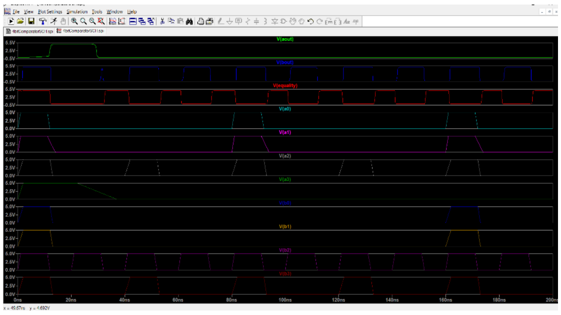


Figure 18 4-bit Compartor Simulation

This 4-bit comparator checks two 4-bit numbers, A and B, starting with the most significant bits to determine which one is greater. At first the MSBs have been compared with each other and if one is greater than the other, without consediring the LSBs values the bigger will give its output 1 ,however it they are not equal then it reverse to the previous bits because they are dertemined the results and so on , using transmission gates to direct the results. The outputs, Awin and Bwin, indicate whether A is greater than B or vice versa. The circuit's final result is based on the first bit where the numbers differ, with inverters ensuring the signals are clear and reliable. Additionally, a 4-input AND gate is used to check for equality between the two numbers, and the inverters at the end make sure these signals are clear and reliable.

2.7. 16-bit Comparator Implementaion

This 16-bit comparator is built using five 4-bit comparators. The first four compare the 16-bit inputs in chunks of 4 bits each. The fifth comparator then takes the results from these comparisons to decide which number is bigger overall and checks for equality using a 4-input AND gate. This setup ensures the entire 16-bit numbers are compared accurately and efficiently.

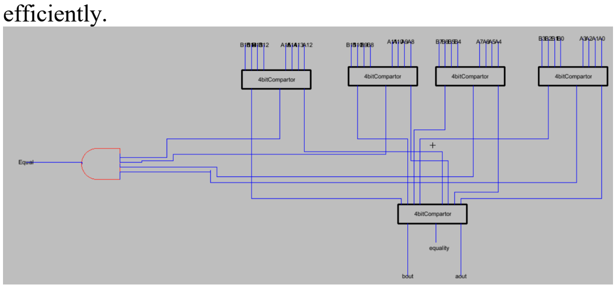


Figure 19 16-bit Comparator schematic

This waveform shows how the 16-bit comparator works as the inputs change, especially focusing on the most significant bits (V(a15) and V(b15)),to figure out which number is bigger. The signals V(aout) and V(bout) indicate which number is greater, and V(equal) is 1 when the numbers are equal. When V(bout) matches V(aout), V(equal) goes to 1, confirming the numbers are the same. This confirms that the comparator is working as expected.

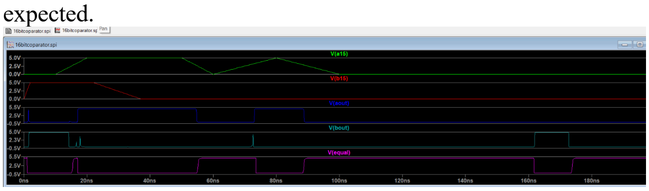


Figure 20 16-bit comparator simulation

This simulation (Figure 20)shows how all the input bits for the 16-bit comparator change over time. The outputs (V(aout), V(bout), and V(equal)) update to show which number is bigger or if they're equal.

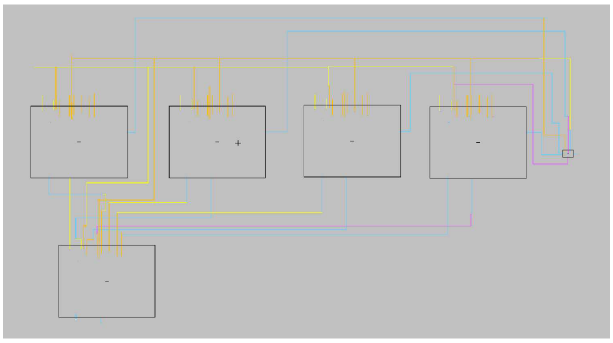


Figure 21 16 bit comparator layout

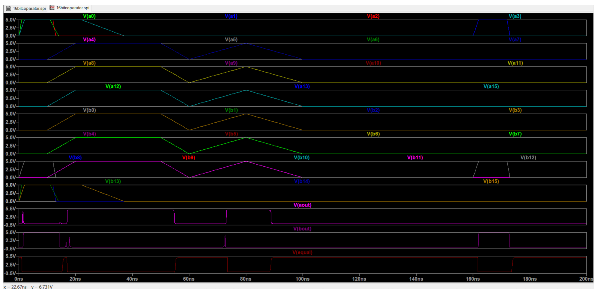


Figure 22 16-bit comparator simulation

This simulation shows what happens when the inputs to the 16-bit comparator are exactly the same. The V(equal) signal stays high the whole time, meaning the comparator correctly identifies that the inputs are equal.

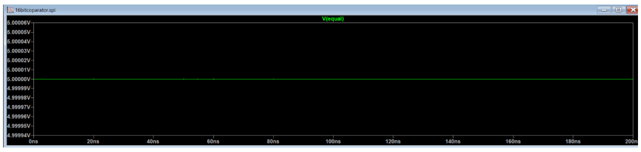


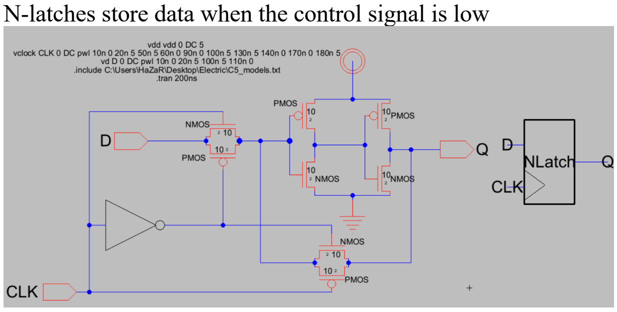
Figure 23 16-bit Comparator simulation (Equality)

3. Enhancements

In this project, we've tried to upgrade the 16-bit comparator by adding registers made from D flip-flops, built using P-latches and N-latches. This change lets us create a pipelined comparator, which speeds things up by allowing different stages of the comparison to happen at the same time. By breaking the process into smaller steps, each handled by its own register.

3.1. N-Latch Implementation

N-latches store data when the control signal is low



```
vdd vdd 0 DC 5
vclock CLK 0 DC pwl 10n 0 20n 5 50n 5 60n 0 90n 0 100n 5 130n 5 140n 0 170n 0 180n 5
vd D 0 DC pwl 10n 0 20n 5 100n 5 110n 0
.include C:\Users\HaZaR\Desktop\Electric\C5_models.txt
.tran 200ns
```

Figure 24 N-Latch Schematic

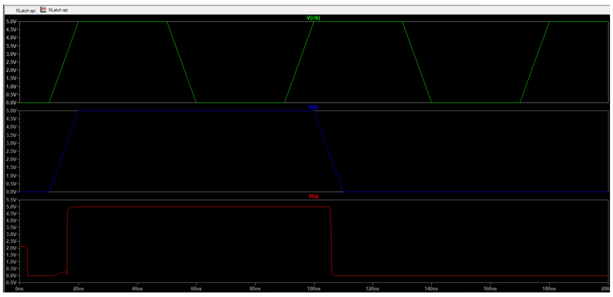
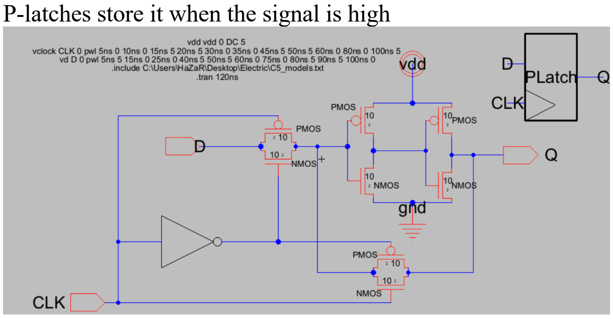


Figure 25 N-latch Simulation

3.2. P-Latch Implementation

P-latches store it when the signal is high



```
vdd vdd 0 DC 5
vclock CLK 0 pwl 5ns 0 10ns 0 15ns 5 20ns 5 30ns 0 35ns 0 45ns 5 50ns 5 60ns 0 80ns 0 100ns 5
vd D 0 pwl 5ns 5 15ns 0 25ns 0 40ns 5 50ns 5 60ns 0 75ns 0 80ns 5 90ns 5 100ns 0
.include C:\Users\HaZaR\Desktop\Electric\C5_models.txt
.tran 120ns
```

Figure 26 P-Latch Schematic

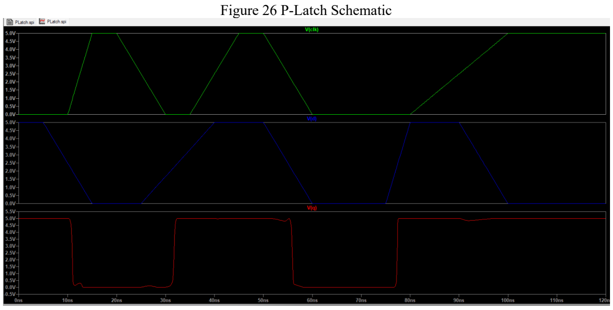
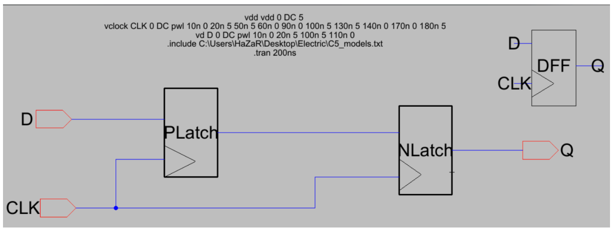


Figure 27 P-latch Simulation

If we had more time, we would have fully implemented the pipelined comparator. We did manage to build the schematic circuit and connect the D flip-flops as shown in figure below.

3.3. DFF Implementation



```
vdd vdd 0 DC 5
vclock CLK 0 DC pwl 10n 0 20n 5 50n 5 60n 0 90n 0 100n 5 130n 5 140n 0 170n 0 180n 5
vd D 0 DC pwl 10n 0 20n 5 100n 5 110n 0
.include C:\Users\HaZaR\Desktop\Electric\C5_models.txt
.tran 200ns
```

Figure 28 DFF schematic

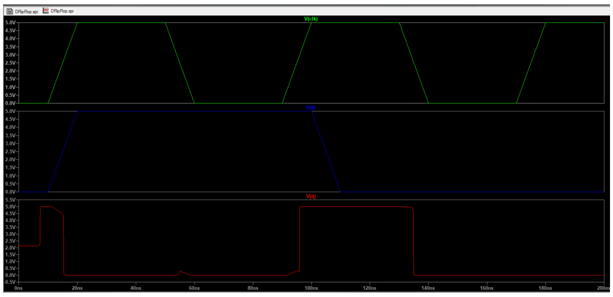


Figure 29 DFF simulation

3.4. 16-bit pipeline Comparator

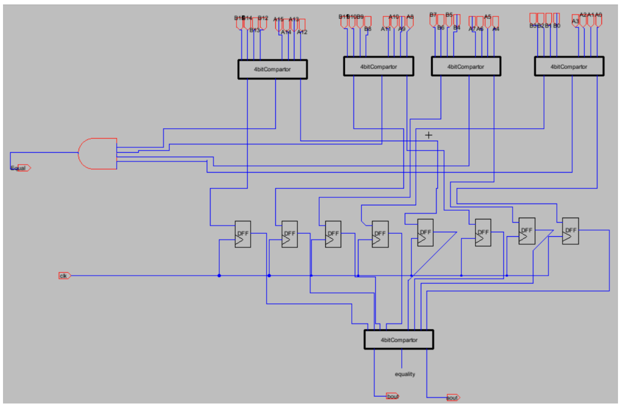


Figure 30 16 - bit Pipeline Comparator

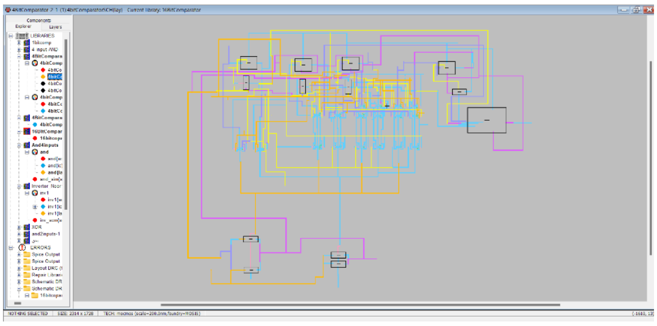
4. Speed, area and power optimization

4.1. Area

We know that electrons move much faster in NMOS transistors than holes do in PMOS transistors, with typical mobility values of 1350 cm²/V·s for electrons and 500 cm²/V·s for holes. To keep performance on par, the PMOS transistors need to be about 2.7 times wider than their NMOS

We focused on making our design as compact as possible while ensuring top performance. Initially, we set the NMOS width at 15 microns and the PMOS at 30 microns.

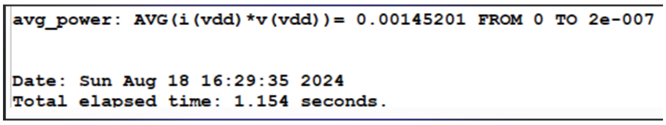
We carefully followed spacing rules for components, wires, and contacts to fit everything into the smallest area without breaking any design rules. We also built the gates from scratch rather than using pre-built ones, giving us more control. The final 4-bit comparator design is compact, measuring 2314 x 1728 units, with the unused area, so the actual area is less than that, achieving a balance between performance and space efficiency.



4.2. Power

CMOS technology reduces capacitor coupling, which lowers dynamic power use. A longer channel length cuts down on leakage current, reducing static power loss. Keeping NMOS and PMOS transistors at minimum width also helps save power by minimizing both dynamic and static losses.

To calculate the power for the comparator, we use the code .measure tran avg_power avg I(Vdd)*V(Vdd).



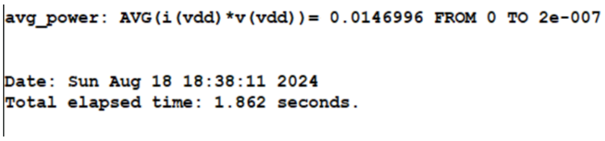
```
avg_power: AVG(i(vdd)*v(vdd))= 0.00145201 FROM 0 TO 2e-007

Date: Sun Aug 18 16:29:35 2024
Total elapsed time: 1.154 seconds.
```

Figure 31 4 bit comparator avg power

For the 4-bit comparator: The simulation shows that the comparator consumes an average power of 1.452 milliwatts over 200 nanoseconds. The simulation was completed in 1.154 seconds.

For the 16-bit comparator: The simulation shows that the comparator consumes an average power of 0.14699 milliwatts over 200 nanoseconds. The simulation was completed in 1.862 seconds.



```
avg_power: AVG(i(vdd)*v(vdd))= 0.0146996 FROM 0 TO 2e-007

Date: Sun Aug 18 18:38:11 2024
Total elapsed time: 1.862 seconds.
```

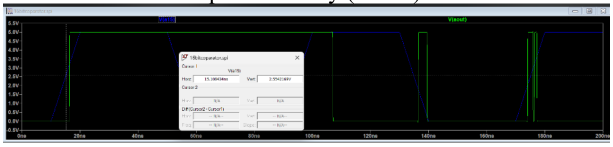
Figure 32 6 bit comparator avg power

4.3. Delay

The propagation delay of the 16x16-bit comparator was determined by calculating the time difference between the 50% point of the input signal and the 50% point of the output signal during the transition of rising or falling edges. This method provides an accurate measurement of the delay for different comparison conditions.

Below are the delay measurements for the 16-bit comparator under various scenarios:

- 16-bit Comparator Delay (A > B)



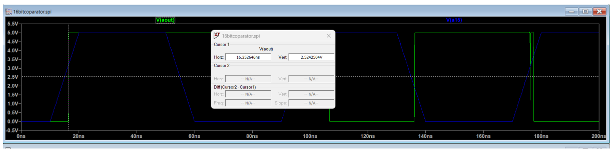
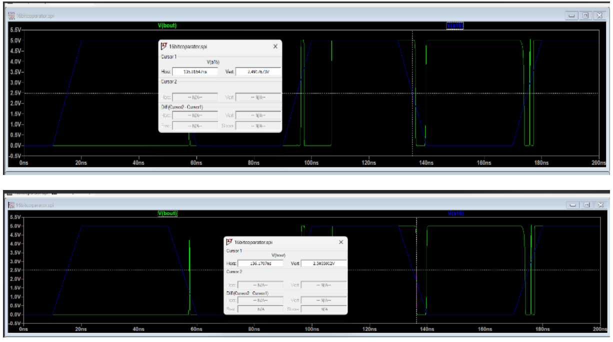


Figure 33 4 bit comparator Delay A>B

Delay = 16.35ns – 15.10ns = 1.25ns

- 16-bit Comparator Delay (B > A)

Figure 34 4 bit comparator Delay B>A



Delay = 136.17ns – 135.01ns = 1.16ns

- 16-bit Comparator Delay (Equality)

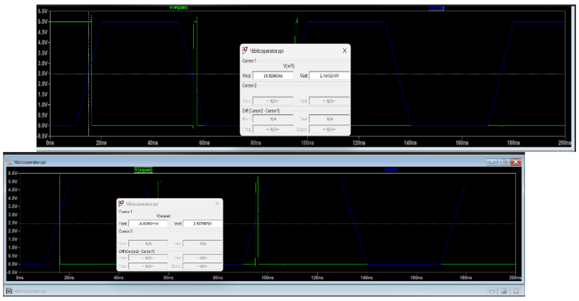


Figure 35 4 bit comparator Delay A=B

Delay = 16.09ns – 14.92ns = 1.17ns

4.4. Speed

We focused on speed in our comparator, by using transmission gate and start comparing by the MSBs also making sure the NMOS and PMOS transistors work equally well. Since NMOS electrons move faster than PMOS holes, we made the PMOS width about 2 times the NMOS width.

To calculate the speed from propagation delay, you can use the following formula:

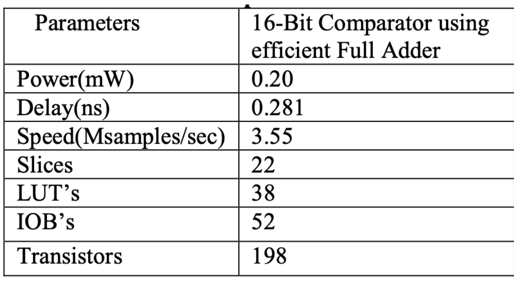
$$\text{Speed} = \frac{Distance}{\text{Propagation Delay}}$$

**Distance is the length of the path over which the signal travels.

Speed 40.28 Samples/ sec

4.5. Comparing with reference

Comparing with reference [5] , which use an effectient Full adder, we find that our power and delay results was higher than it results, on the otherhand our speed was much higher as a result of using Transmission gate, also the technology difference was a main factor of these unclose results.



Parameters	16-Bit Comparator using efficient Full Adder
Power(mW)	0.20
Delay(ns)	0.281
Speed(Msamples/sec)	3.55
Slices	22
LUT’s	38
IOB’s	52
Transistors	198

Figure 36 Comparing results

5. Conclusions

In this project, we successfully designed and implemented a 16-bit comparator using transmission gates. The transmission gate approach allowed us to create a fast, efficient, and reliable comparator while minimizing power consumption and size. We also tried to enhance the design by introducing pipelining through D flip-flops using N-latches and P-latches. Although we couldn’t complete the full pipelined comparator due to time constraints, the groundwork we laid shows promising potential for future development. Overall, this project demonstrates how using transmission gates can lead to a more efficient and effective comparator design.

[1]

6. Refernces

[1] "Semantic Scholar," [Online]. Available: https://www.semanticscholar.org/paper/Design-of-Low-Power-and-High-Speed-4-Bit-Comparator-Prajpat-Joshi/96a8839bcfda55afd57371e1ae5d43f0dbda57c2. [Accessed 11 8 2024].

[2] "ResearchGate," [Online]. Available: https://www.researchgate.net/figure/2-Bit-Comparator-using-Transmission-Gate-logic-5_fig1_260632302. [Accessed 10 8 2024].

[3] "IEEE Xplore," [Online]. Available: https://ieeexplore.ieee.org/document/6918857. [Accessed 11 8 2024].

[4] "IJSRD," [Online]. Available: https://www.ijsrd.com/Article.php?manuscript=IJSRDV10I100045. [Accessed 12 8 2024].

[5] Design and Implementation of 16-Bit Magnitude Comparator Using Efficient Low Power High Performance Full Adders