**Design of Low Power 4X4 Magnitude Comparator Using GDI Technique**

1Abdalrahman Juber 2 Ismail Tarteer 3Robert Bannoura

Computer Engineering dept Computer Engineering dept Computer Engineering dept

Birzeit University Birzeit University Birzeit University

Birzeit, Palestine Birzeit, Palestine Birzeit, Palestine

1211769@student.birzeit.edu 1211243@student.birzeit.edu 1210574@student.birzeit.edu

Abstract-Design for Low power, high speed and reliable design is a very important role in emerging technologies in Integrated circuit. To reduce power consumption and propagation delay involved in the circuit, the paper presents GDI technique for implementation of digital logic circuit . In this paper, a 4x4 bit magnitude comparator is designed using conventional GDI logic style and GDI technique. The proposed GDI technique magnitude Comparator requires 130 transistors and implementation using CMOS logic requires 227 transistors. The power consumed by the 4x4 bit magnitude CMOS comparator is 2.5 μw and the power consumed by the 4x4 GDI technique magnitude Comparator is 0.35 μw which is very less compared to conventional CMOS Style. Delay present in the conventional CMOS magnitude comparator is 70.92 nsec whereas the delay produced by GDI technique magnitude Comparator 40.8 ns nsec. Delay is also reduced in the proposed GDI technique.GDI technique shows 86% efficiency for power measurement. All this circuit simulation is done by using ELECTRICTOOL EDA version 9.07 at 45 nm process technology.

***Keywords:*** *GDI, Comparator, VLSI, CMOS and MC.*

# Introduction

**(a) CMOS Technique**

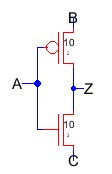
Complementary Metal-Oxide-Semiconductor CMOS logic circuits are most likely used in digital design for their low power consumption (But not always!!). CMOS technology utilizes a combination of PMOS NMOS transistors to make logic functions. PMOS are used to pull the output high, while NMOS transistors are used to pull the output low.

**(b) GDI Technique**

A GDI cell has three inputs: G (the common gate input), P (the input connected to the source/drain of the PMOS transistor), and N (the input connected to the source/drain of the NMOS transistor). Contrary to a CMOS inverter, the bulks of the NMOS and PMOS transistors in a GDI cell are connected to N and P, respectively, allowing for arbitrary biasing. It's important to note that not all functions can be realized using a standard P-Well CMOS

process, but they can be effectively implemented using Twin-Well CMOS or SOI technologies.

GDI Technique allows improvements in design complexity level, transistor counts, static power dissipation and logic level swing. Fig.1 represents the basic building block of GDI cell. In this cell Boolean expression of Z is . On the basis of this expression, any logic can be implemented by GDI cells [2].



**Fig.1** basic building block of GDI cell

**(c) Magnitude Comparator (MC)**

# A magnitude comparator is a important combinational logic circuit used to compare between two binary values. It determines whether the numbers are equal, or if one number is greater than or less than the other. The comparator has three outputs: G, E, and L. When A > B, only G is enabled. When A = B, only E is enabled. When A < B, only L is enabled. Also, in this project we have cascade outputs which are used to implement larger comparator by connecting each comparator with the other using these outputs Ei, Li and Gi.

# ANALYSIS OF GDI TECHNIQUE

The GDI method which is first proposed by A. Morgenstern,

A. Fish, and I. A. Wagner in 2001, is based on the use of a

simple cell as shown in figure.2. At first glance, the basic cell

reminds the standard CMOS inverter, but there are some

important differences:

1. The GDI cell contains three inputs: G (common gate

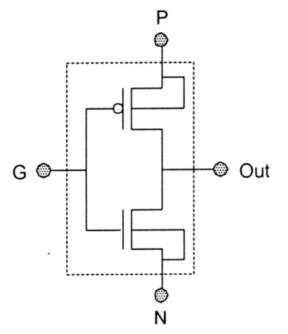
input of nMOS and pMOS), P (input to the source/drain of

pMOS), and N (input to the source/drain of nMOS).

2. Bulks of both nMOS and pMOS are connected to N

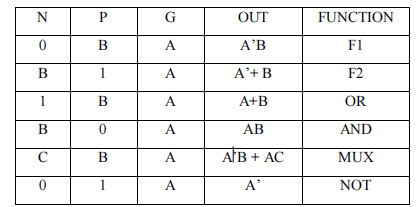
or P (respectively), so it can be used to implement gates similar CMOS technique.

.



**Fig.2.** GDI basic cell [2]

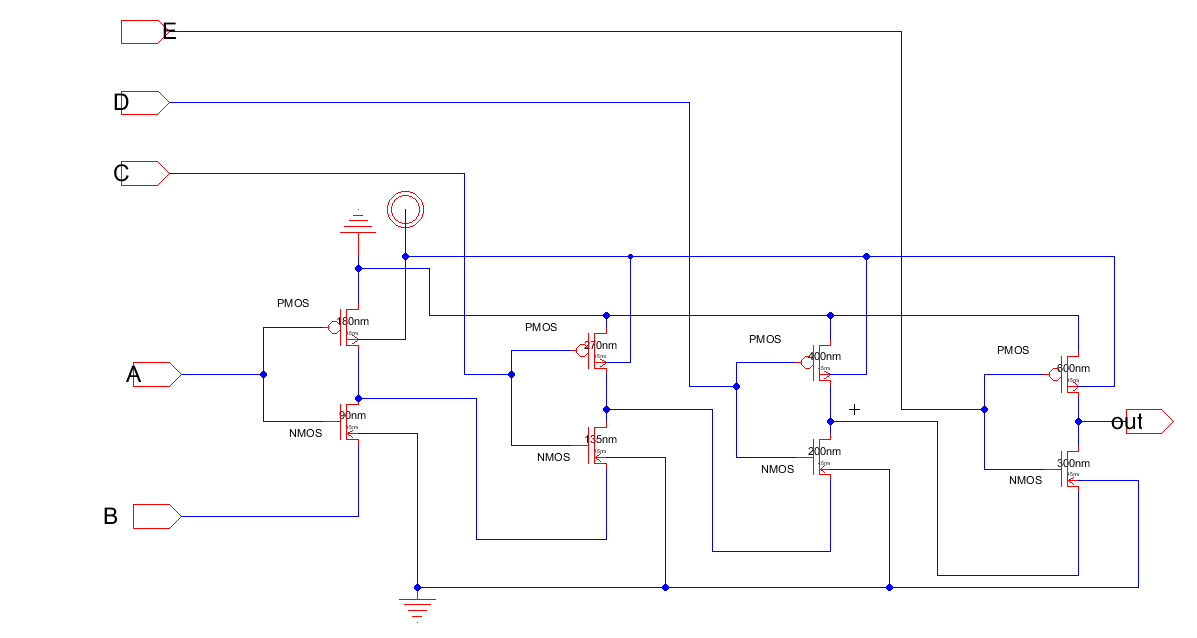
**Table I:** Various Logic Functions of GDI Cell

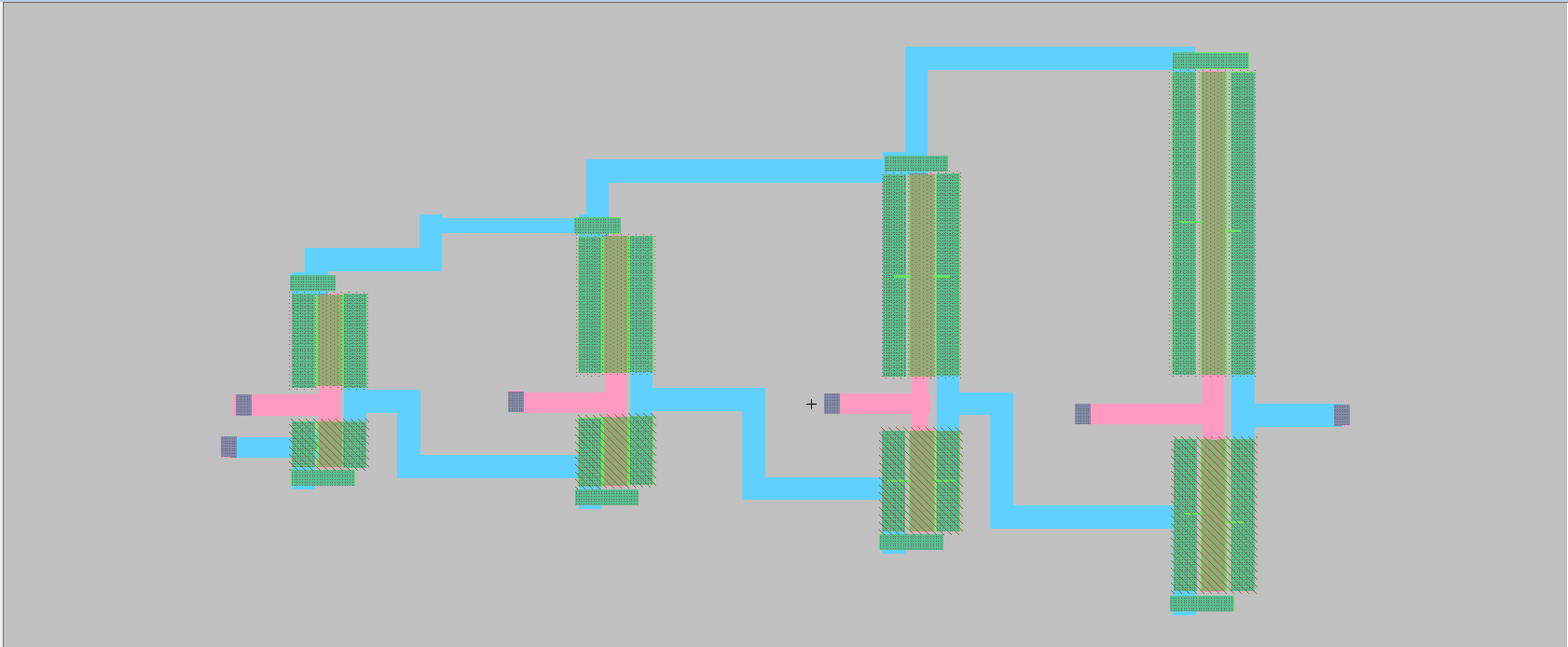


# Design Methodology

There are different technologies to construct integrated circuits such as bipolar integrated technology, CMOS technology, NMOS pass transistor logic, Transmission Gate(TG) technology and gate diffusion input(GDI) technology. In this paper we have used GDI technology to design 4-bit MC and comparison is made against CMOS and GDI technologies. The main reason of using GDI technique is due to its low propagation delay, low power consumption and low chip area.

We used basic components such as the inverter, 2-input, 3-input, 4-input, and 5-input AND gates, a 5-input OR gate, and a 2-input XOR gate. These components will be used while building our comparator. Figures 3, 4, and 5 represent the basic building blocks of our design, which include the 5-input AND gate, the 5-input OR gate, and the 2-input XOR gate.





**Fig.3** schematic and layout of 5-input AND gate (GDI tech)

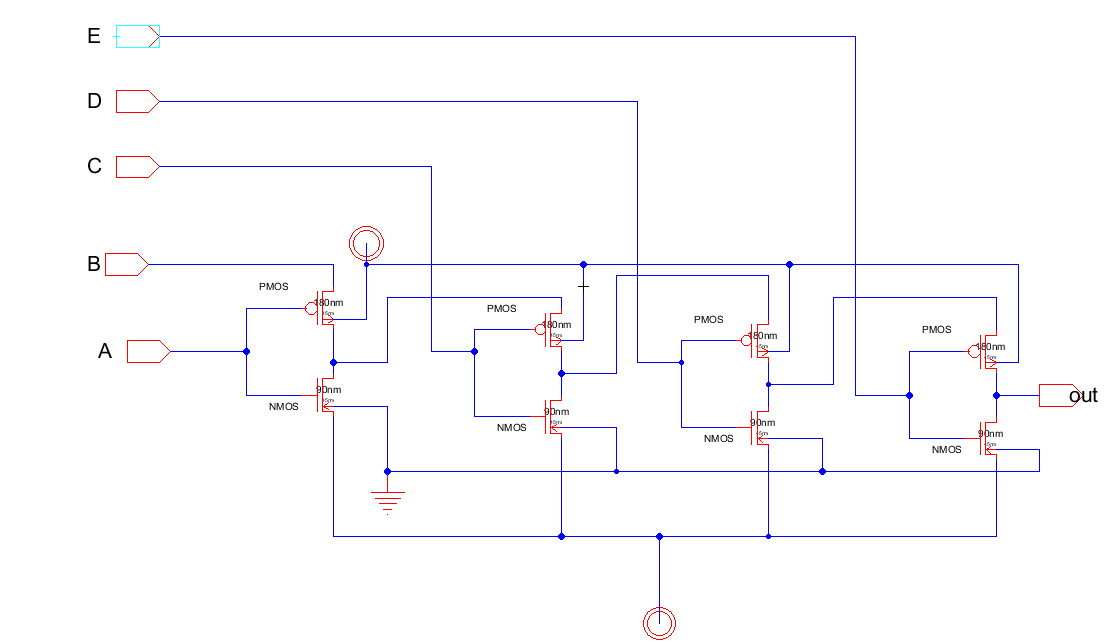
The sizes of transistors are (PMOS,NMOS) for and gate:

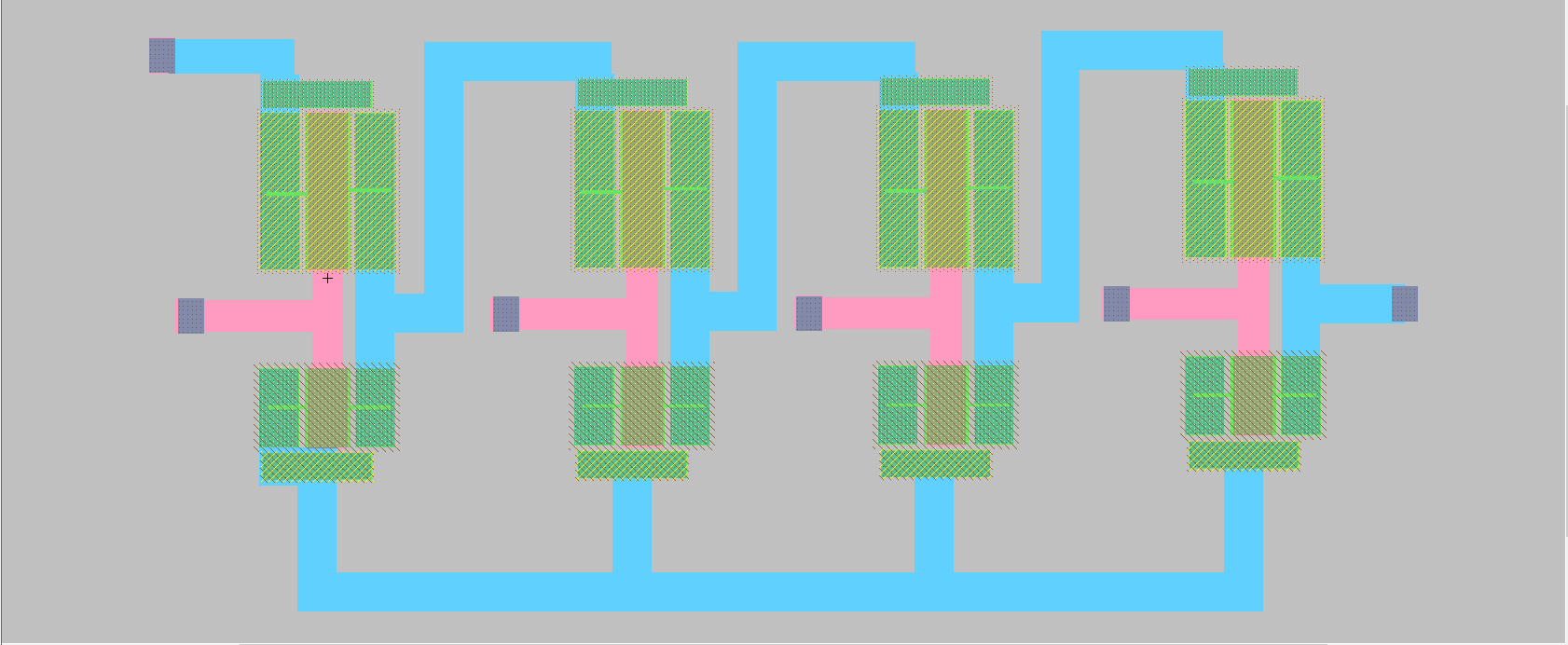
(180nm,90nm) , (270nm,135nm),

(400nm,200nm),(600nm,300nm).

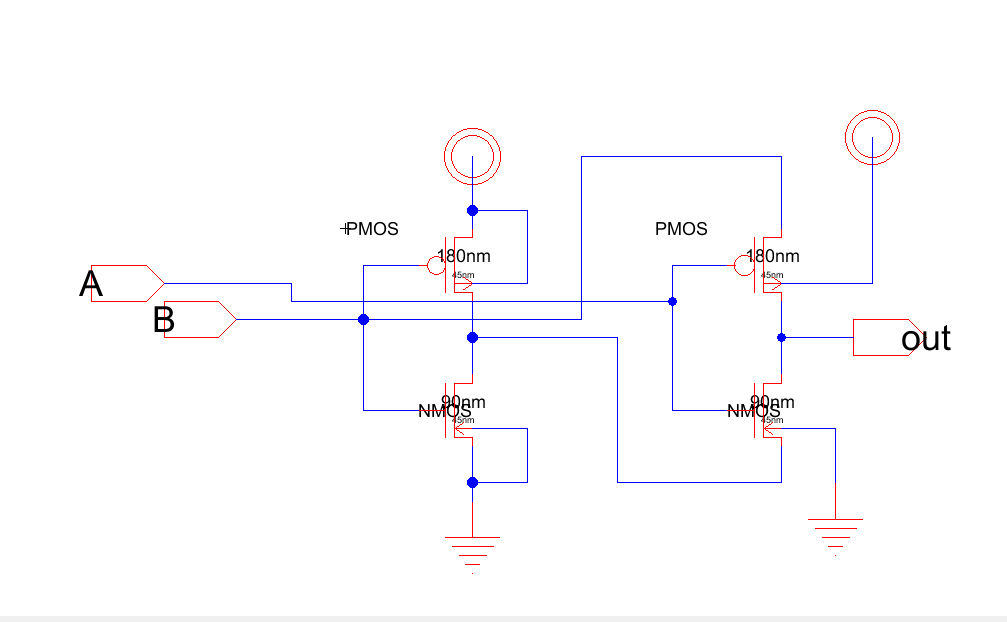
The sizes of transistors are (PMOS,NMOS) for or and xor

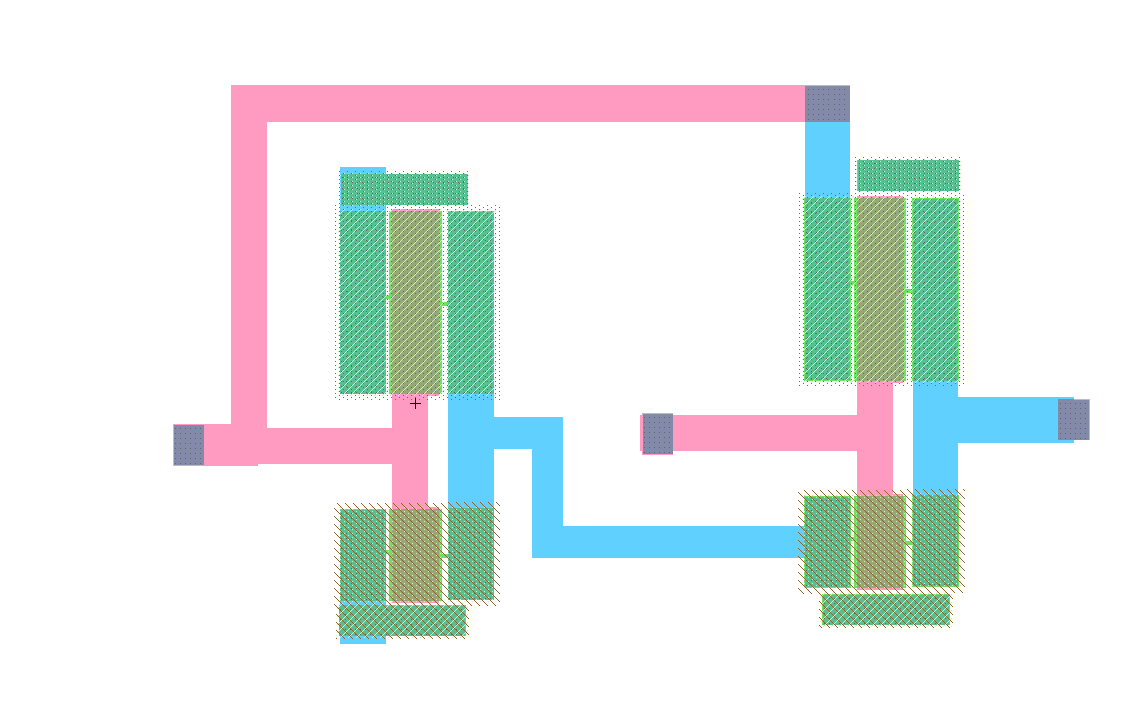
gates: (180nm,90nm)





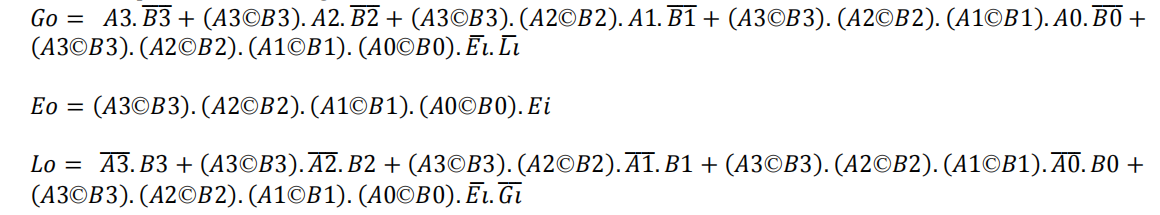
**Fig.4** schematic and layout of 5-input OR gate (GDI tech)





**Fig.5** schematic and layout of 2-input XOR gate (GDI tech)

Boolean equation of our design is:



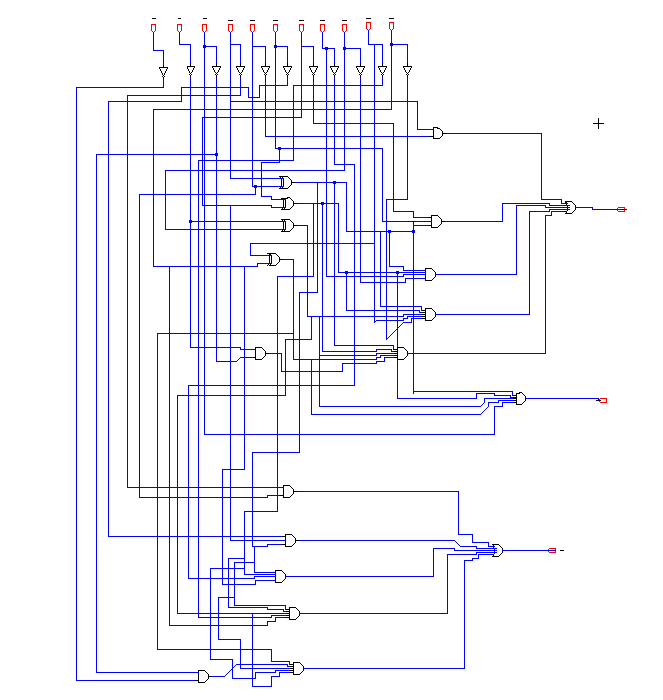
For the inputs A3, A2, A1, A0, B3, B2, B1, and B0 are being analyzed. Additionally, there are cascading inputs Gi, Ei, and Li, will be used to build bigger comparator as shown in table 3.

Also, we have implemented MC using its basic building blocks. Fig.6 and fig.7 are the schematic and layout of MC.

There are various applications of our design. This design is flexible by which higher order of MC can be implemented.

Cascading multiple ICs enables comparison of binary numbers over 4 bits.

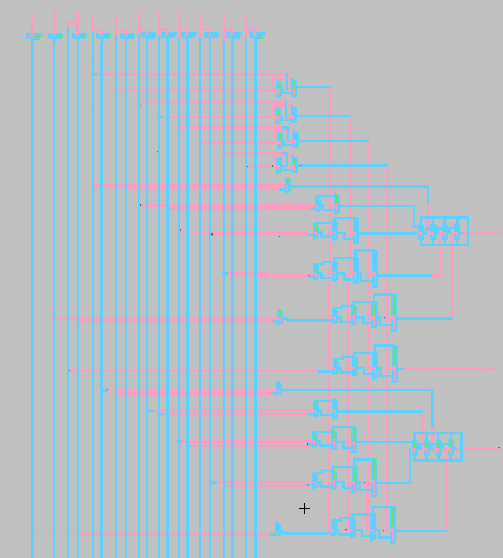
|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| COMPARING INPUTS | | | | CASCADING INPUTS | | | OUTPUTS | | |
| A3, B3 | A2, B2 | A1, B1 | A0, B0 | Gi | Li | Ei | Go | Lo | Eo |
| A3>B3 | X | X | X | X | X | X | 1 | 0 | 0 |
| A3=B3 | A2>B2 | X | X | X | X | X | 1 | 0 | 0 |
| A3=B3 | A2=B2 | A1>B1 | X | X | X | X | 1 | 0 | 0 |
| A3=B3 | A2=B2 | A1=B1 | A0>B0 | X | X | X | 1 | 0 | 0 |
| A3<B3 | X | X | X | X | X | X | 0 | 0 | 1 |
| A3=B3 | A2<B2 | X | X | X | X | X | 0 | 0 | 1 |
| A3=B3 | A2=B2 | A1<B1 | X | X | X | X | 0 | 0 | 1 |
| A3=B3 | A2=B2 | A1=B1 | A0<B0 | X | X | X | 0 | 0 | 1 |
| A3=B3 | A2=B2 | A1=B1 | A0=B0 | 1 | 0 | 0 | 1 | 0 | 0 |
| A3=B3 | A2=B2 | A1=B1 | A0=B0 | 0 | 1 | 0 | 0 | 1 | 0 |
| A3=B3 | A2=B2 | A1=B1 | A0=B0 | 0 | 0 | 1 | 0 | 0 | 1 |
| A3=B3 | A2=B2 | A1=B1 | A0=B0 | X | X | 1 | 0 | 0 | 1 |
| A3=B3 | A2=B2 | A1=B1 | A0=B0 | 1 | 1 | 0 | 0 | 0 | 0 |
| A3=B3 | A2=B2 | A1=B1 | A0=B0 | 0 | 0 | 0 | 1 | 1 | 0 |



**Fig.6** schematic of 4-bit MC (GDI tech)

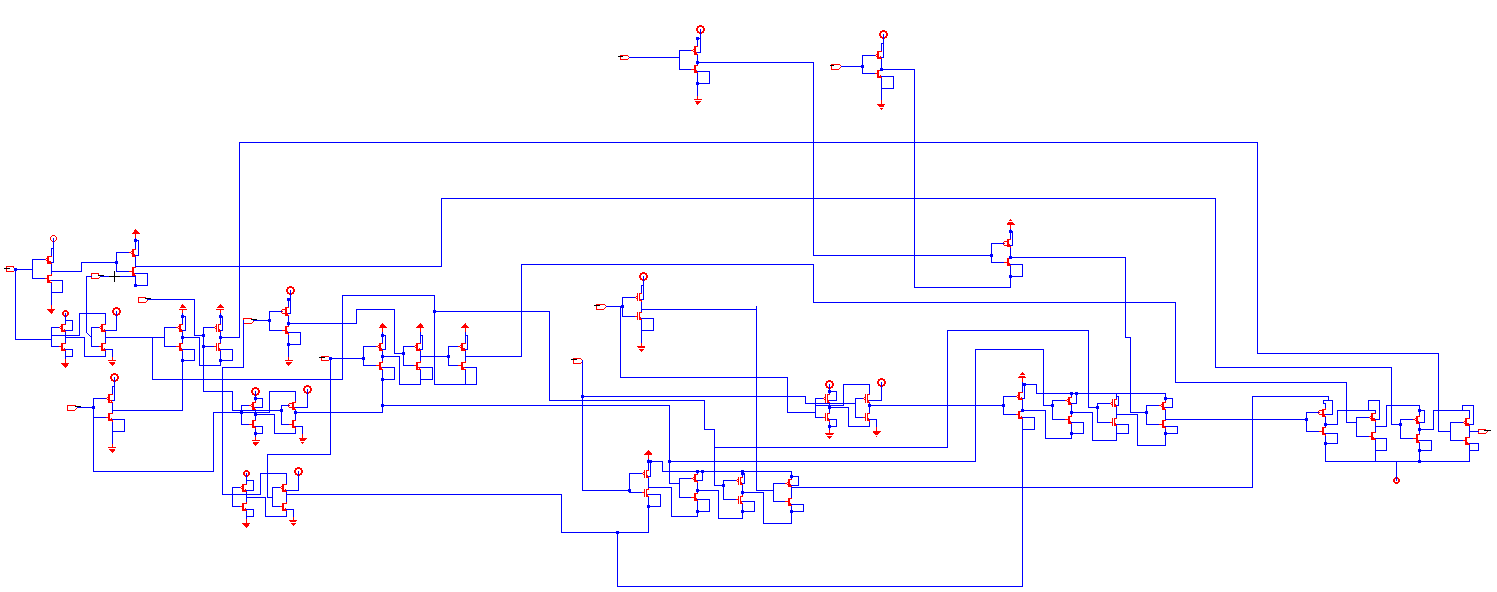
**Table.3** truth table of proposed MC

\

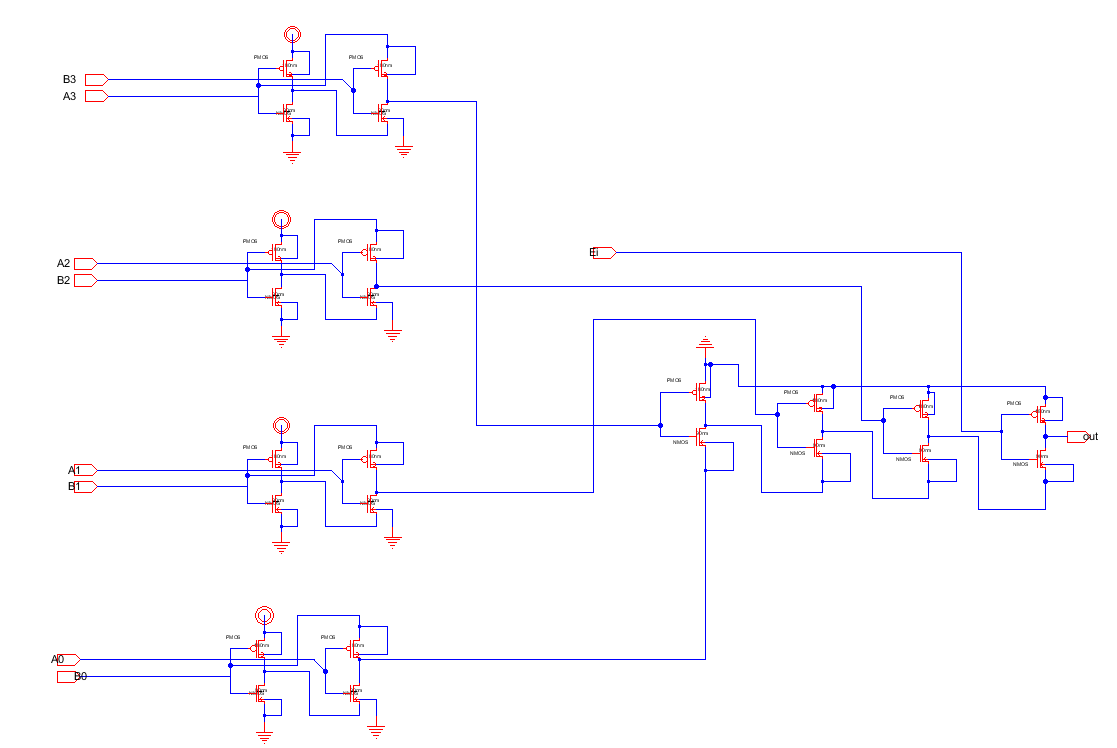


**Fig.7** layout of 4-bit MC (GDI tech)

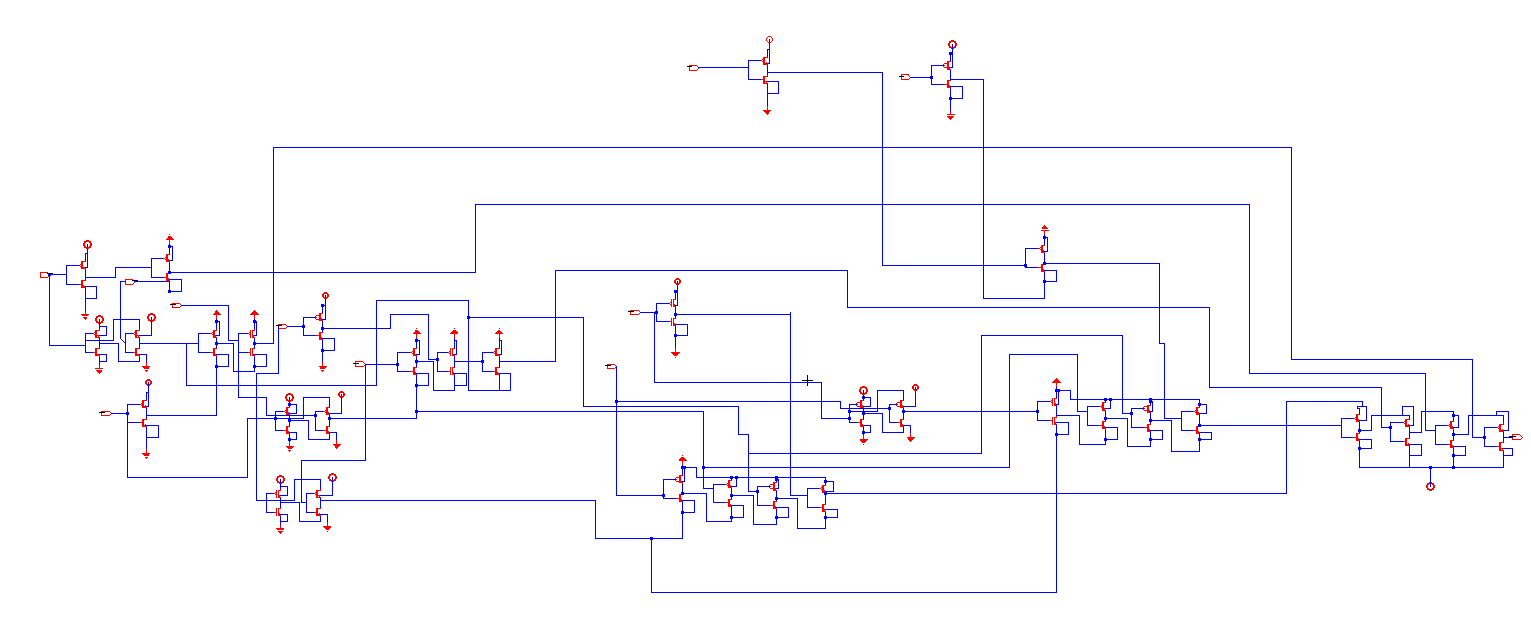
Finally we built schematic for different Boolean gates as shown in figure 8,9 and 10 for G0 , E0 and L0 respectively.



**Fig.8** schematic of G0 (GDI tech)



**Fig.9** schematic of E0 (GDI tech)

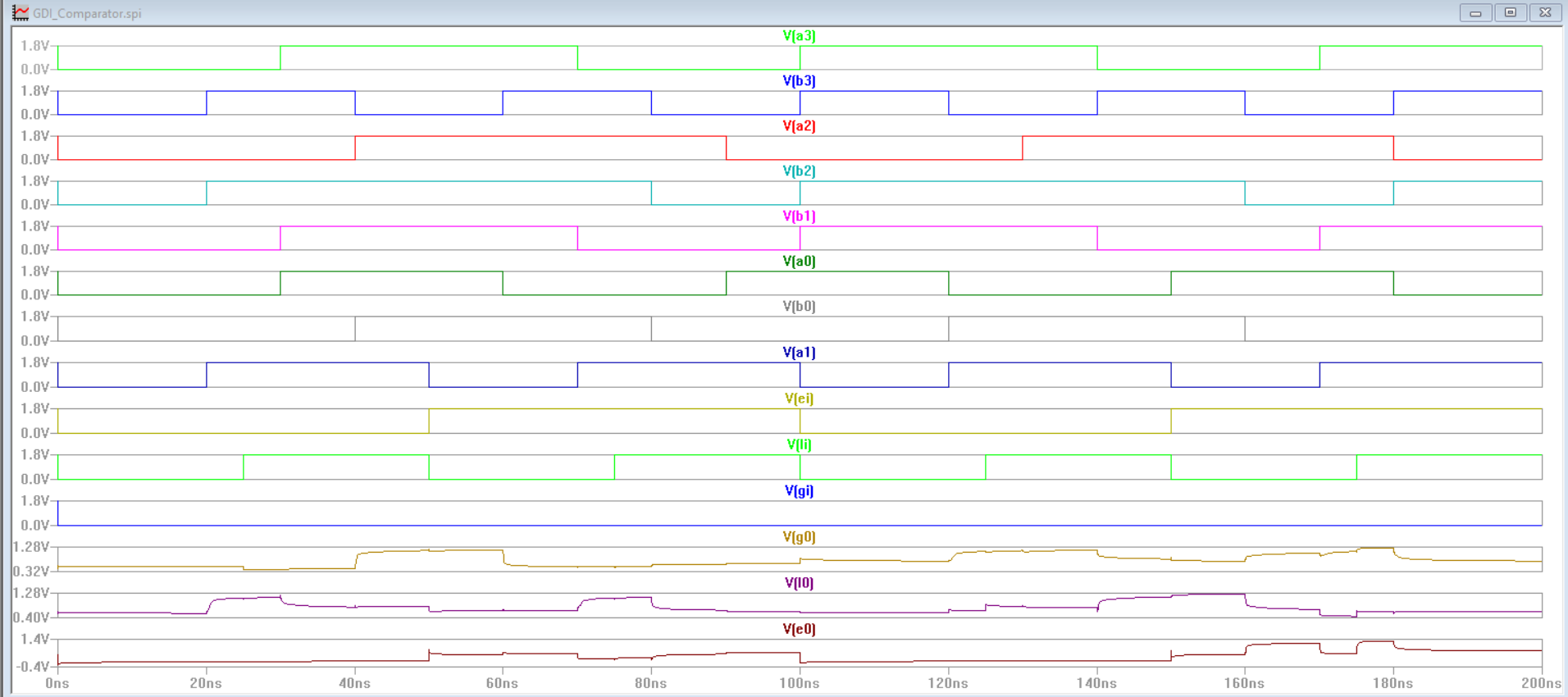


**Fig.10** schematic of L0 (GDI tech)

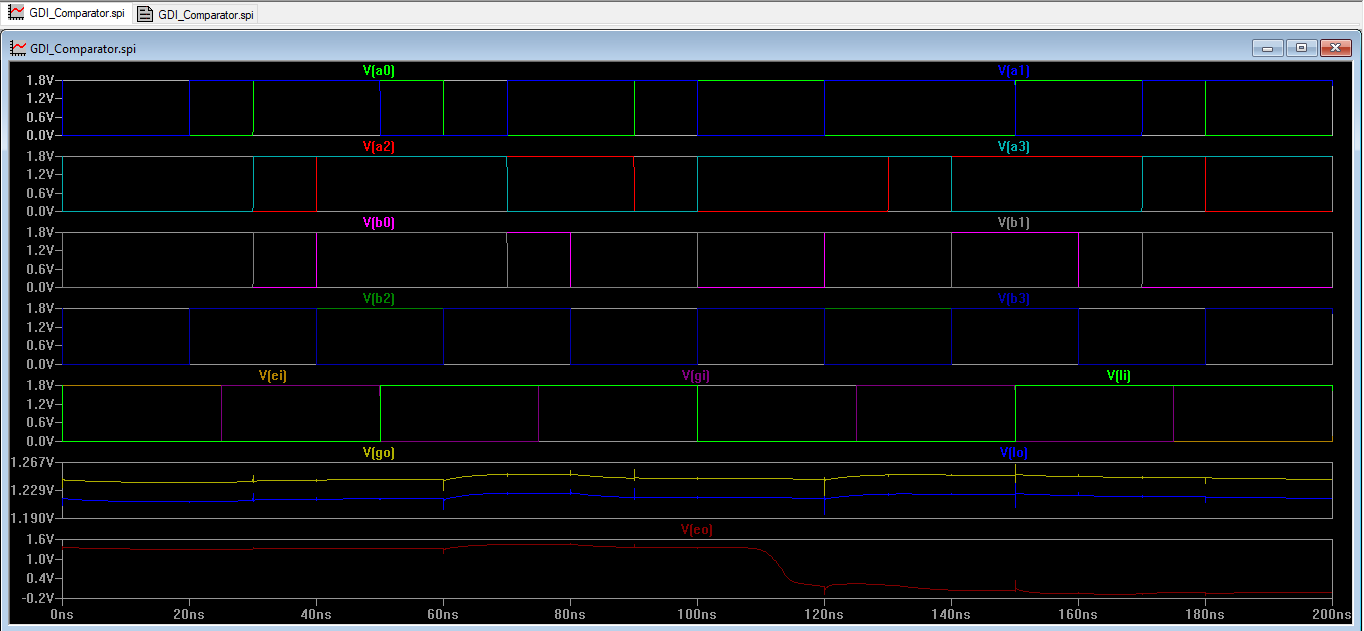
**IV.Analysis Of Simulation Result**

## Wave form

The simulation results of the MC are illustrated in figure 11. Additionally, a detailed description of how the MC operates can be found in table 3, providing a comprehensive overview of its functions and capabilities.



**Fig.11 (a)** simulation result of 4-bit MC (Schematic)



**Fig.11 (b)** simulation result of 4-bit MC (layout)

## Spice code

V3 Gi 0 DC 1.8 PULSE 0 1.8 0 0 1ps 1ps 100ns 200ns

V4 Ei 0 DC 1.8 PULSE 1.8 0 0 1ps 1ps 50ns 100ns

V6 Li 0 DC 1.8 PULSE 1.8 0 0 1ps 1ps 25ns 50ns

V7 A3 0 DC 1.8 PULSE 1.8 0 0 1ps 1ps 30ns 70ns

V8 A2 0 DC 1.8 PULSE 1.8 0 0 1ps 1ps 40ns 90ns

V9 A1 0 DC 1.8 PULSE 1.8 0 0 1ps 1ps 20ns 50ns

V10 A0 0 DC 1.8 PULSE 1.8 0 0 1ps 1ps 30ns 60ns

V11 B3 0 DC 1.8 PULSE 1.8 0 0 1ps 1ps 20ns 40ns

V12 B2 0 DC 1.8 PULSE 1.8 0 0 1ps 1ps 20ns 80ns

V13 B1 0 DC 1.8 PULSE 1.8 0 0 1ps 1ps 30ns 70ns

V14 B0 0 DC 1.8 PULSE 1.8 0 0 1ps 1ps 40ns 80ns

.tran 1ps 200ns

V0 Vdd 0 DC 1.8

\*\*\*\*\*G0\*\*\*\*\*

.meas time1\_G0 FIND time WHEN V(B3) = 0.9V RISE=1

.meas time2\_G0 FIND time WHEN V(G0) = 566mV RISE=1

.meas tPHL\_G0 param time2\_G0-time1\_G0

.meas time3\_G0 FIND time WHEN V(B3) = 0.9V FALL=1

.meas time4\_G0 FIND time WHEN V(G0) = 566mV FALL=1

.meas tPLH\_G0 param time4\_G0-time3\_G0

.meas Propogation\_delay\_Go param (tPLH\_G0+tPHL\_G0)/2

\*\*\*\*\*L0\*\*\*\*\*

.meas time1\_L0 FIND time WHEN V(B3) = 0.9V RISE=1

.meas time2\_L0 FIND time WHEN V(L0) = 566mV RISE=1

.meas tPHL\_L0 param time2\_L0-time1\_L0

.meas time3\_L0 FIND time WHEN V(B3) = 0.9V FALL=1

.meas time4\_L0 FIND time WHEN V(L0) = 566mV FALL=1

.meas tPLH\_L0 param time4\_L0-time3\_L0

.meas Propogation\_delay\_L0 param (tPHL\_L0+ tPLH\_L0)/2

\*\*\*\*\*E0\*\*\*\*\*

.meas time1\_E0 FIND time WHEN V(B3) = 0.9V RISE=1

.meas time2\_E0 FIND time WHEN V(E0) = 566mV RISE=1

.meas tPHL\_E0 param time2\_E0-time1\_E0

.meas time3\_E0 FIND time WHEN V(B3) = 0.9V FALL=1

.meas time4\_E0 FIND time WHEN V(E0) = 566mV FALL=1

.meas tPLH\_E0 param time4\_E0-time3\_E0

.meas Propogation\_delay\_E0 param (tPHL\_E0+tPLH\_E0)/2

.include C:\Users\Desktop\C5\_models.txt

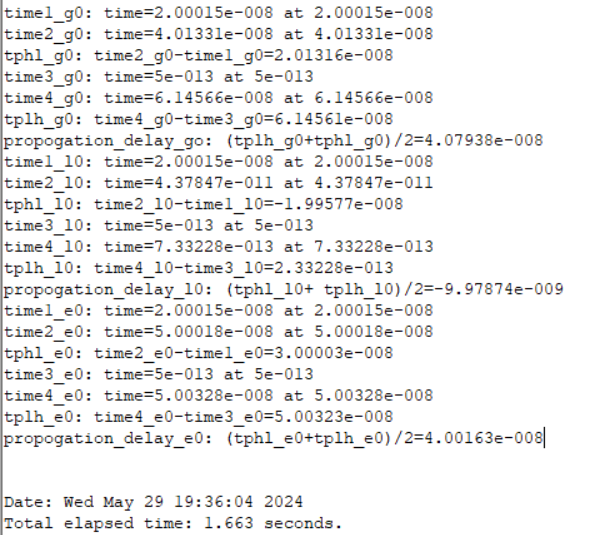
In the Spice code we used 1.8 voltage for both inputs and vdd,

Also, we used. meas to measure the propagation delay for all outputs.

## Finding critical path

### Using simulation

To find the critical path we need to find the highest propagation delay



**Fig.12** Simulation of Tp

As seen for the previous outputs we can conclude the propagation delay for all outputs are:

Tp(G0) = 40.8 ns

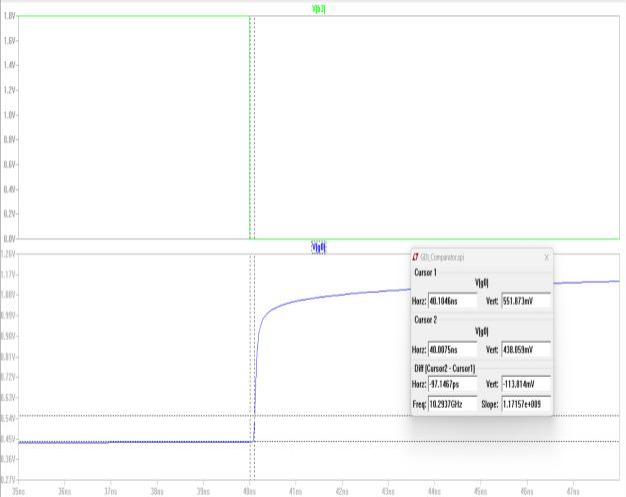
Tp(L0) = 0.998 ns

Tp(E0) = 40 ns

**So, the worst case propagation delay is 40.8ns from G0 output.**

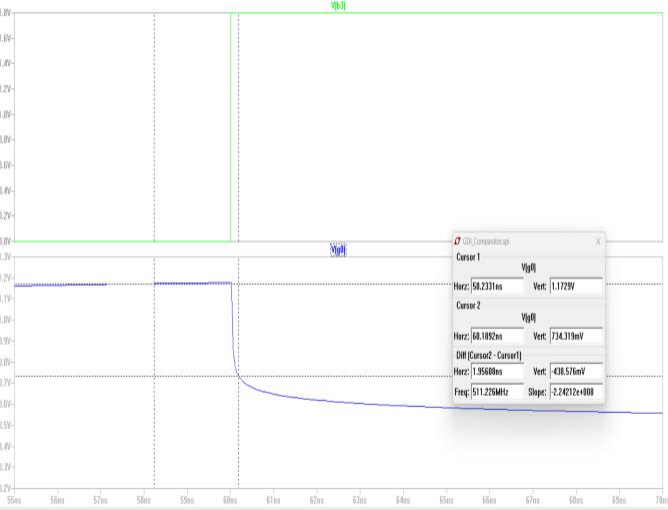
### Using hand calculation

1. *Propagation* *delay for* *G0*



**Fig.13** tPHL for G0

Tphl-G0=40.1046-40=0.1046

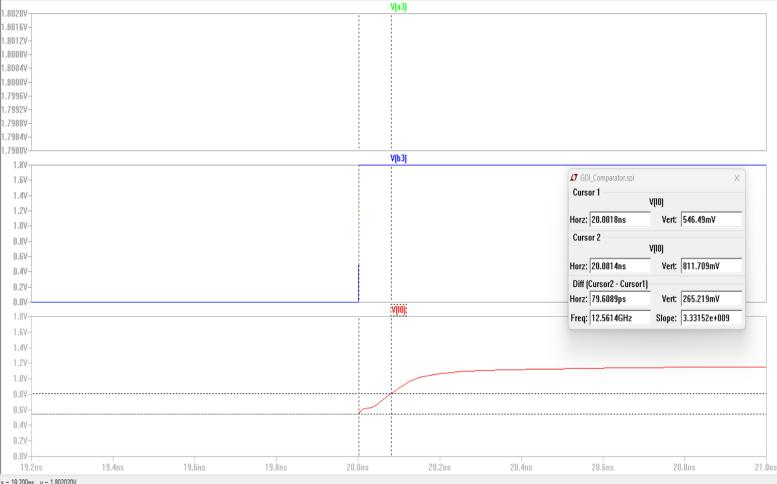


**Fig.14** TPLH for G0

Tplh-G0 = 60.1892-60=0.1892

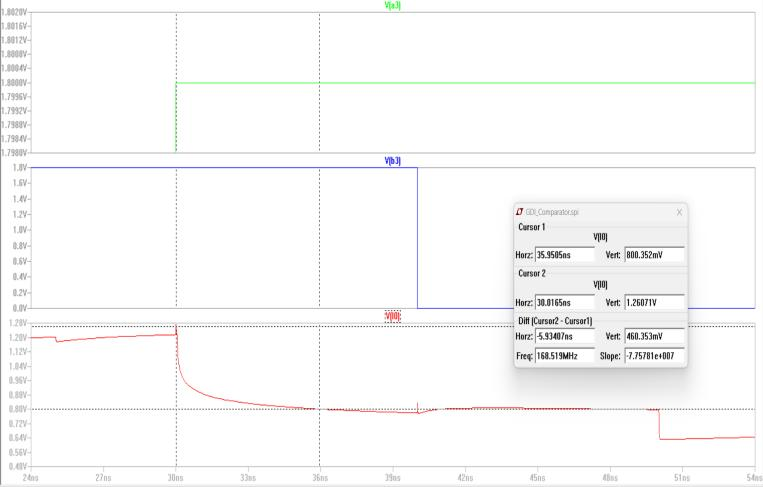
Tp = (0.1046 + 0.1892) / 2= 0.1469 ns

### 2) Propagation delay for L0



**Fig.15** tPHL for L0

Tphl-L0=20.0814ns-20=0.0814ns



**Fig.16** tPLH for G0

Tplh-L0 = 35.9505ns-30.0165ns=5.93407ns

Tp = (0.0814ns + 5.93407ns) / 2= 3.01ns

We didn’t calculate E0 because it has little transistors which it the faster output. And as it was calculated before using simulation.

**Also, in the figure before we have shown the critical signals with each of its output.**

**Also, the critical path was built which is Go in figure 8.**

## Finding average power

## To find average power we need to find the current that passes through the comparator, we calculated it as shown in fig 17.

## 

**Fig.17** Measure I(vdd)

**Average Power = Ivdd \* VDD = 1.8 \* 0.2u = 0.35** µW

# Comparison

Today’s processor demands to develop various new design techniques in order to reduce the propagation delay, chip area and power consumption. By selecting best techniques. Table.4 provides comparing against different technologies where many inputs gates This table compare between CMOS and GDI techniques which the last is more efficient for all outputs.

**Table.5** Transistors usage and delay of MC

has implemented. This table basically provide idea to develop the design by the use of CMOS, TG and GDI techniques

**Table.4** Transistors usage in multi input Gates by different technologies[1]

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **SL NO** | GATE | **TECHNOLOGIES** | | | |
| **CMOS** | | **GDI** | |
| **SCHEMATIC** | **TRANSI STORS USAGE** | **SCHEMATIC** | **TRANSIS TORS USAGE** |
| 1 | 3-  input AND |  | 8 |  | 4 |
| 2 | 3-  input OR |  | 8 |  | 4 |
| 5 | 2-  input XOR |  | 12 |  | 4 |

Our design lastly provides following result whose details have given in table.5. CMOS tech. Uses 226 transistors whereas GDI tech. uses only 130 transistors to implement our design (MC).

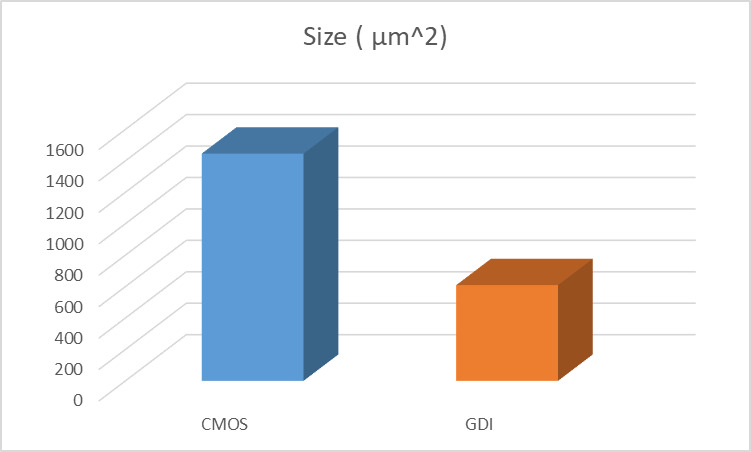
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Technology | Transistors Usage | Average Delay (in ns) | Power consumption (µW) | Area(µm\*µm) |
| CMOS | 226 | 70.92 | 2.5 | 1446 |
| GDI | 130 | 40.8 | 0.35 | 609.32 |

has implemented. This table basically provide idea to develop the design by the use of CMOS, TG and GDI techniques.

# Conclusion

GDI technique is used for Basic Logic Gates and some Digital circuits. It is compared to standard CMOS and pass transistor logics, showing that GDI is a novel and effective way to reduce power consumption and transistor count, ultimately shrinking chip size. GDI enables high-density fabrication, making it advantageous in terms of chip area, power consumption, and transistor count compared to other techniques.

GDI tech. reduces 57.85% of chip area, 5.64% of average delay time and also 86% of power consumption over CMOS tech. which has shown in fig.17..



**Figure 17.** Comparison Graph of Delay, Area, Transistor

Count and Power Dissipation between GDI and CMOS Circuits

##### References

1. https://www.researchgate.net/publication/316479699\_IC\_Layout\_Design\_of\_4bit\_Magnitude\_Comparator\_using\_Electric\_VLSI\_Design\_SystemJ. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.
2. https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=c35d770942f22b4a1595a225c518335e3acdd494K. Elissa, “Title of paper if known,” unpublished.
3. R. Nicole, “Title of paper with only first word capitalized,” J. Name Stand. Abbrev., in press.
4. Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, “Electron spectroscopy studies on magneto-optical media and plastic substrate interface,” IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].