

**DESIGN OF LOW POWER DIGITAL CIRCUITS USING MEMRISTOR WITH
CMOS LOGIC IN COMPARISION WITH FINFET & CMOS TECHNOLOGY**

FULL SEMESTER INTERNSHIP REPORT

Submitted by

M. LOKESH (20341A04B3)

N. MANOJ KUMAR (20341A04C7)

S. ANVESH (20341A04G7)

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

ELECTRONICS AND COMMUNICATION ENGINEERING

GMR Institute of Technology, Rajam

Andhra Pradesh, India

APRIL 2024

**DESIGN OF LOW POWER DIGITAL CIRCUITS USING MEMRISTOR WITH
CMOS LOGIC IN COMPARISION WITH FINFET & CMOS TECHNOLOGY**

Internship carried out at

MAHINDRA UNIVERSITY, HYDERABAD

Report Submitted by

M. LOKESH (20341A04B3)

N. MANOJ KUMAR (20341A04C7)

S. ANVESH (20341A04G7)

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

ELECTRONICS AND COMMUNICATION ENGINEERING

**GMR Institute of Technology,
RajamAndhra Pradesh, India**

APRIL 2024

Department of Electronics and Communication Engineering

CERTIFICATE

This is to certify that the thesis entitled **DESIGN OF LOW POWER DIGITAL CIRCUITS USING MEMRISTOR WITH CMOS LOGIC IN COMPARISION WITH FINFET & CMOS TECHNOLOGY** submitted by M. LOKESH (20341A04B3), N. MANOJ KUMAR (20341A04C7), S. ANVESH (20341A04G7), has been carried out in partial fulfillment of the requirement for the award of degree of Bachelor of Technology in Electronics and Communication of GMRIT, Rajam affiliated to JNTU-GV, Vizianagaram is a record of bonafide work carried out by them under my guidance & supervision. The results embodied in this report have not been submitted to any other University or Institute for the award of any degree.

Signature of Supervisor		Signature of HOD
Mrs. P. Revathi		Dr. V. Jagan Naveen
Assistant Professor		Head of the Department
ECE Department		ECE Department
GMRIT, Rajam		GMRIT, Rajam

The report is submitted for the viva-voce examination held on

Signature of Internal Examiner

Signature of External Examiner

ACKNOWLEDGEMENT

I would like to thank **Dr. P. Chaitanya Akshara**, Assistant Professor in Department of ECE in MAHINDRA UNIVERSITY, HYDERABAD for giving me the opportunity to do an internship within the organization. I also would like all the people that worked along with me in Mahindra university, Hyderabad with their patience and openness they created an enjoyable working environment. It is indeed with a great sense of pleasure and immense sense of gratitude that I acknowledge the help of these individuals.

I express my gratitude to our Director **Dr. J. GIRISH**, Principal **Dr. C. L. V. R. S. V. PRASAD** and Academic Dean **Dr. M.V. NAGESWARA RAO** for arranging the full semester internship as part of the academic schedule.

I also extend my gratitude to the Head of the Department **Dr. V. JAGAN NAVNEEN**, Professor, Department of ECE who supported us in successfully completing full semester internship.

I am deeply in debt to our central internship coordinator **Dr. M. KRISHNA PRASAD**, Professor for his moral support in successfully completing the full semester internship.

I am extremely thankful to our faculty supervisor **Mrs. P. Revathi**, Assistant Professor and our Department internship coordinator **Dr. A. SUDHAKAR**, Associate Professor for his valuable suggestions and guidance during our internship period.

I would like to thank all the members of GMRIT & Mahindra University for their direct or indirect support and their valuable suggestions and for providing excellent opportunities in the completion of this report.

M. Lokesh	(20341A04B3)
N. Manoj Kumar	(20341A04C7)
S. Anvesh	(20341A04G7)

ABSTRACT

The relentless demand for low-power electronic devices has spurred research into novel circuit design techniques. Memristors, with their unique properties, offer promising opportunities for achieving low power consumption in digital circuits. This work presents a comparative study of the design of low-power digital circuits using memristors with CMOS logic, contrasted against FinFET and conventional CMOS technology. The study includes an analysis of power consumption and propagation delay of logic gates and the Wallace and Dadda Multipliers. Simulation results demonstrate the potential of memristor-based designs to significantly reduce power consumption while maintaining competitive performance levels compared to existing technologies. Furthermore, the scalability and adaptability of memristor-based circuits for future technological advancements are discussed. This research contributes valuable insights into the integration of memristors with CMOS logic, paving the way for energy-efficient digital circuit designs in next-generation electronic systems.

Keywords: *CMOS, Dadda multiplier, FinFET, Memristor, MOSFET, Wallace multiplier.*

TABLE OF CONTENTS

CONTENTS	PAGE NO
1. INTRODUCTION	1
1.1. Internship	1
1.2. Benefits of Internship	
1.2.1. Benefits to the students	2
1.2.2. Benefits to the Industry	2
1.2.3. Benefits to the Institution	2
1.3. Ethics	3
1.4. Values	3
2. PROFILE OF THE COMPANY	4
2.1. Mahindra University	4
2.2. Department of ECE	
2.3. Industry Insights	5
3. PROBLEM STATEMENT AND BACKGROUND	6
3.1. MOSFET	6
3.1.1. MOSFET principles of operation	7
3.1.2. IDS vs VDS characteristics for a n-channel MOSFET	7
3.1.3. IDS vs VGS characteristics of a n-channel MOSFET	8
3.1.4. Subthreshold Swing (SS) of NMOS transistor	10
3.2. FinFET	8

3.2.1. Different configurations of finFET inverter	
3.2.1.1. Shorted gate configuration (SG) inverter	
3.2.1.2. Low Power Configuration (LP)	
3.2.1.3. Independent Gate Configuration (IG)	9
3.3. Memristor	12
3.3.1. Relation of electrical parameters	13
3.3.2. Hysteresis Loop	18
3.3.3. Window Functions	19
3.4. Total Power Analysis	
3.5. Propagation Delay	
4. METHODOLOGY	26
4.1. Procedure for Designing and Simulating a circuit	26
4.1.1. Invoke Custom Compiler	27
4.1.2. Create a Library	
4.1.3. Create a New Cell View	
4.1.4. Schematic Editor (SE) – Basic Schematic Cell Creation	
4.1.4.1. Editing Instance Parameters	
4.1.4.2. Add Wires	
4.1.4.3. Create Wire Name	
4.1.4.4. Add Pins	
4.1.4.5. Check and Saving the circuit	
4.1.5. Schematic Editor (SE): Symbol Creation	
4.1.6. Testbench Creation	
4.1.7. Prime Wave Design Environment	
4.1.7.1. Add variables	
4.1.7.2. Setup Model Files	
4.1.7.3. DC Sweep Analysis	
4.1.7.4. Transient Analysis	
4.1.7.5. Power and Delay Calculation	

4.2. Generation of Memristor Symbol	
4.3. Design of Fundamental Logic Gates	
4.3.1. Inverter	
4.3.2. OR gate	34
4.3.3. AND gate	35
4.3.4. NOR gate	35
4.3.5. NAND gate	
4.3.6. Half Adder	
4.3.7. Full Adder	
4.4. Design of Binary Multiplier Circuits	
4.4.1. Design of 4*4 Partial Product Generator	
4.4.2. Wallace Multiplier	
4.4.3. Dadda Multiplier	

5. RESULTS

6. CONCLUSION

APPENDIX

REFERENCES

LIST OF FIGURES

Fig. No.	NAME OF FIGURE	Page No.
2.1	Logo of the Organization	3
3.1	Physical structure of NMOS	6
3.2	Symbols for NMOS transistor	7
3.3	Symbols for PMOS transistor	7
3.4	I-V output characteristics of NMOS transistor	8
3.5	I-V transfer characteristics of an NMOS transistor	8
3.6	Demonstration of the subthreshold swing extraction	9
3.7	Fabrication of CMOS using N-WELL	9
3.8	Structure of FinFET	10
3.9	Symbol of n-channel and p- channel FinFET	10
3.10	Ids vs Vds for n-channel FinFET	11
3.11	Shorted gate Configuration	11
3.12	Low Power Configuration	12
3.13	Independent Gate Configuration	12
3.14	Relation between electrical parameters	13
3.15	HP memristor model	13
3.16	Symbol of Memristor	14
3.17	Hysteresis curve of the memristor	14
3.18	Propagation delay of an Inverter	17
4.1	Custom Compiler Window	
4.2	New Library Window	
4.3	New CellView Window	
4.4	Placing the Instances	
4.5	Schematic of Inverter	
4.6	Console Window	
4.7	Generate CellView from CellView	

Fig. No.	NAME OF FIGURE	Page No.
4.8	Created Symbol for the Schematic	
4.9	Symbol of Inverter	
4.10	Testbench View	
4.11	Vpulse, Vdc component parameters	
4.12	Prime Wave Design Environment after adding Variables	
4.13	Setting up Model Files	
4.14	DC Analysis Setup	
4.15	Transient Analysis Setup	
4.16	Simulation options	
4.17	DC Analysis Output Selection	
4.18	DC Analysis (Voltage Transfer Characteristic Curve of the Inverter)	
4.19	Transient Analysis Output Selection	
4.20	Transient Analysis Waveform	
4.21	Power calculation Setup	
4.22	Total Power of Inverter	
4.23	t_{pLH} Setup	
4.24	t_{pHL} Setup	
4.25	t_{pLH} of Inverter	
4.26	. t_{pHL} of Inverter	
4.27	Custom Compiler Window	
4.28	Library Manager	
4.29	Import text pop up window	
4.30	Netlist file	
4.31	New Library	
4.32	Pop up window for symbol generation	
4.33	Memristor Library	
4.34	Generated symbol	

Fig. No.	NAME OF FIGURE	Page No.
4.35	Memristor Symbol	
4.36	Symbol of Inverter	
4.37	Schematic of Inverter using CMOS	
4.38	Schematic of Inverter using finFET	
4.39	Transient Analysis of Inverter	
4.40	Symbol OR gate	
4.41	Schematic of OR gate using CMOS	
4.42	Schematic of OR gate using FinFET	
4.43	Schematic of OR gate - Memristor	
4.44	Testbench of OR gate	
4.45	Transient Analysis of OR gate	
4.46	Symbol AND gate	
4.47	Schematic of AND gate using CMOS	
4.48	Schematic of AND gate using finFET	
4.49	Schematic of AND gate-Memristor	
4.50	Testbench of AND gate	
4.51	Transient Analysis of AND gate	
4.52	Symbol NOR gate	
4.53	Schematic of NOR gate using CMOS	
4.54	Schematic of NOR gate using FinFET	
4.55	Testbench of NOR gate	
4.56	Transient Analysis of NOR gate	
4.57	Symbol NAND gate	
4.58	Schematic of NAND gate using CMOS	
4.59	Schematic of NAND gate using finFET	
4.60	Testbench of NAND gate	
4.61	Transient Analysis of NAND gate	

LIST OF TABLES

CHAPTER - 1

INTRODUCTION

1.1. Internship

An internship is a trained and supervised experience in a professional setting in which student is learning and gaining essential experience and expertise. Internship is meant for introducing candidates either full-time or part time to a real-world experience related to their career goals and interests. Internship is an excellent way to build those all-important connections that are invaluable in developing and maintaining a strong professional network for the future. Internships provide real world experience to those looking to explore or gain the relevant knowledge and skill required to enter a career field. Internship is relatively short term in nature with the primary focus on getting some on the job training and taking what's learning in the classroom and applying it to the real world.

1.2. Benefits of Internship

Students learn how their course of study applies to the real world and build valuable experiential that makes them stronger candidates for jobs after graduation.

- Internship at a start-up will benefit in improving team spirit, adapting to flexible working times and client services.
- You can get serious work experience, build a portfolio and establish a network of professional contacts which can help you after you graduate.
- The main advantage is to have practical knowledge. In our college we can have theoretical knowledge which doesn't help much. Working on a project gives the practical experience.
- Confidence can be increased when we were involved in solving problems and were succeeded in solving it.
- If you are willing to show initiative, enthusiasm and work hard, you will be given further opportunities to develop.
- Working on a project also improves communication skills and interpersonal skills. As we need to talk to higher authorities regarding the project our skills can be better when compared. Having several internships while in college can be very impressive to potential employers.
- Working in team for a project teaches us how to interact with our colleagues and how to deal them without hurting the feelings of both sides.

1.2.1. Benefits to the Students

- Learning by doing.
- All round development.
- Aid in career planning.
- Experience of professional working conditions.
- Smooth transition from campus to company.

1.2.2. Benefits to the Industry

- Steady stream of skilled manpower provides value addition and increased productivity.
- Human Resource Development benefits.
- Conduit for Industrial Partnership.
- Employer Branding.

1.2.3. Benefits to the Institution

- Inputs to quickly adapt curriculum to match the needs of industry.
- Opportunities for research and consultancy.
- Access to industrial expertise and infrastructure.

1.3. Ethics

- Help develop an organizational environment favorable to acting ethically.
- Improve their understanding of the software and related documents on which they work and of the environment in which they will be used.
- Accept full responsibility for their own work. Improve their ability to produce accurate, informative, and well-written documentation.
- Assist colleagues in professional development.
- Strive to fully understand the specifications for software on which they work.
- Improve their knowledge of the Code, its interpretation, and its application to their work.

1.4. Values

- Professional communications.
- Be proactive, and when invited to work functions introduce oneself to people.
- Taking constructive criticism well.
- Being able to work independently with little guidance is very important in the working world.
- Always work hard even the task is small and seems unimportant.

CHAPTER - 2

PROFILE OF THE ORGANIZATION

2.1. Mahindra University

The Mahindra University is located at Hyderabad is a university with a significant role in the global dynamics of R&D and higher learning, where students are motivated to attain their true potential. The faculty and staff aim to achieve excellence in pedagogy and contemporary frontiers of research and in providing services respectively while alumni strive to achieve global leadership.

Mahindra University is home to exceptional schools of Engineering, Management, Law, Education, Life Sciences, Media and Hotel Management that offer a world-class education. The university equips students with cutting-edge technical skills and fosters innovation through industry partnerships. Further, the schools nurture leaders by comprehensively understanding strategies and practices. All the schools emphasize experiential learning and research, preparing students to excel in their chosen fields. Mahindra University's schools empower students to become future-ready professionals.



Fig 2.1. Logo of the Organization

2.2. Department of ECE

Electrical and Computer Engineering (ECE) is a dynamic and integral field that is experiencing rapid growth and transformation driven by advancements in technology. It encompasses the design, development, and application of electrical and electronic systems. ECE professionals are in high demand in a wide range of industries, including telecommunications, healthcare, manufacturing, and transportation.

2.3. Industry Insights

The growth of the Internet of Things (IoT): The IoT is connecting billions of devices to the internet, and this is creating a huge demand for ECE professionals. ECE professionals are needed to design, develop, and maintain IoT devices and systems.

The development of quantum computing: Quantum computing is a new technology that has the potential to revolutionize many industries. ECE professionals are needed to develop the hardware and software that powers quantum computers.

The integration of advanced technologies, including smart meters, energy management systems, and grid automation, is driving the need for ECE professionals to optimize energy distribution, improve efficiency, and enable intelligent energy consumption.

The growth of technologies like data science, AI, ML, virtual reality and robotics are revolutionizing various sectors of electronics & computers and creating new opportunities for growth and advancement.

Areas like sensor networks, wireless communication, and cloud infrastructure are gaining exponential importance and focus.

Edge computing, which brings computing resources closer to the data source, has helped reduce latency, improve data privacy, and enable real-time processing for applications like autonomous vehicles, smart cities, and industrial automation.

The deployment of 5G networks and the ongoing development of future communication technologies will open new opportunities in wireless communication, network infrastructure, IoT applications, and autonomous systems.

CHAPTER - 3

TASKS TAKEN UP AND PROBLEM DEFINITION

In the realm of Very Large-Scale Integration (VLSI) circuit design and synthesis, the primary challenge lies in minimizing propagation delay and power dissipation (P_d). This objective hinges on crucial design parameters such as the W/L ratio, scaling technology, and load capacitance. Moore's Law, originally formulated by Gordon Moore in 1965, posits that the number of transistors on a microchip will double approximately every two years, leading to a continuous increase in computational power and efficiency. As technology nodes progress, the channel length of MOSFETs has consistently decreased, adhering to the principle of scaling down [1]. Smaller transistors enable a higher transistor count in the same area, promoting faster switching and reduced energy consumption, with sizes measured in nanometres. MOSFETs are preferred over BJTs as BJTs tend to have higher power consumption, especially in standby or idle states, accountable to continuous current flow and have thermal issues [2].

CMOS technology leverages the complementary operation of NMOS and PMOS transistors within the same circuit, where one type of transistor is used for the pull-up network (PMOS) and the other for the pull-down network (NMOS). This complementary arrangement ensures that only one type of transistor is conducting at any given time, leading to minimal power dissipation during idle states and significantly reduced static power consumption. This also ensures high noise margin and rapid switching speed, thermal efficiency [3].

The major disadvantage of the scaling of MOS transistor is the increase in the leakage current due to the short channel. In short channel MOSFET, the gate terminal is no longer able to control the channel current efficiently. Because of that the static power dissipation will get increased. It can be reduced by gate having a better control over the channel. The scaling down in a conventional MOSFET transistor becomes difficult under 32nm technology because of existing of short channel effects.

Memristors produce different characteristics in different applications. For example, an element that can compute, control, and store the data after calculation is needed in logic and memory applications. Memristors are two-terminal components with a size of only 10 nm and that can integrate devices at very high densities. In addition to this memristors having higher switching speed i.e., in nanoseconds. Because of its non-volatile resistance state, memristors are desirable options for usage as modern memory components.

Memristors are resistors with memory that allow you to change the resistance level by applying a voltage or current. Memristive devices may therefore solve many major problems in the semiconductor

industry, providing non-volatile, dense, fast, and power efficient memory.

Hybrid memristor is far better than the CMOS logic implementation. The power consumption is reduced to nearly 70 percent. Considering the hypothesis that memristors are smaller than MOSFETs, hybrid memristor-CMOS logic uses less area compared to the CMOS logic. Because the memristors under consideration for testing have a width of only 3 nm, significantly less than the width of 180 nm of a MOSFET, this revolutionary logic brings in the area saving. One MOSFET can contain several memristors due to the possibility of implementing them on the polysilicon layer of the MOSFET. A single MOSFET can execute a large number of complex functions, however CMOS logic requires a substantially larger number of transistors to compute a same function. There is a reduction of around 47% in area if Hybrid Memristor-CMOS logic is used.

3.1. MOSFET

3.1.1. MOSFET principles of operation

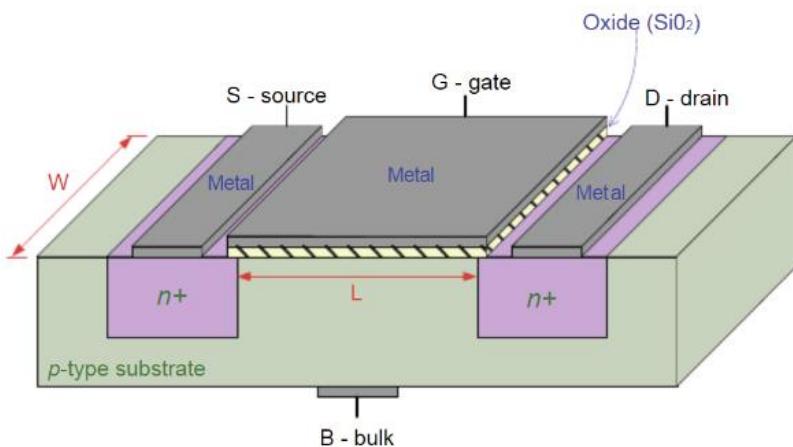


Fig. 3.1. Physical structure of NMOS [4]+

Fig. 1 shows the physical structure of a silicon based MOSFET cross section. It displays the silicon substrate, the heavily doped drain (D) and source (S) regions, the insulators between the channel and gate(G), and the thin gate insulator. The area between the source and drain is called the channel. The gate terminal controls the transistor (ON or OFF state) while, in the ON state, the channel current (I_{DS}) flows between the source and drain. Hence, a MOSFET can be thought of as a voltage-controlled current source. For an n-channel MOSFET (NMOS), the substrate used is a p-type silicon substrate and the source and drain regions are heavily doped n-type regions, while for a p-channel MOSFET (PMOS), the substrate is an n-type silicon substrate with heavily doped p-type drain and source regions. In the Fig. 3.1. W is the width of the gate and L denotes the length of the channel [5].

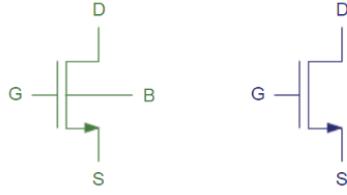


Fig. 3.2. Symbols for NMOS transistor [4]

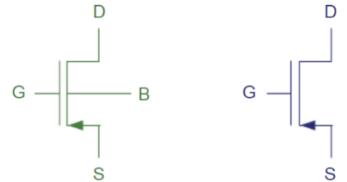


Fig. 3.3. Symbols for PMOS transistor [4]

3.1.2. I_{DS} vs V_{DS} characteristics for a n-channel MOSFET

Fig. 4 shows the I-V output characteristics curves of an NMOS. We have a voltage between the gate and the source (V_{GS}) and between drain and source (V_{DS}). Such I_{DS} vs. V_{DS} curves are obtained by grounding the source terminal, setting an initial V_{GS} value (typically zero), and sweeping V_{DS} from zero to the maximum DC voltage value provided by the power supply (V_{DD}), while stepping V_{GS} from zero to V_{DD} . For very low V_{GS} , I_{DS} is extremely small, while when V_{GS} gets higher, I_{DS} increases and will have the following dependency on V_{GS} and V_{DS} [6]:

- If $V_{GS} \leq V_{TH}$, then the MOSFET is in the OFF state and acts, ideally, as an open circuit.
- If $V_{GS} > V_{TH}$, then there are two modes of operation:

$$(i) \quad V_{DS} < V_{GS} - V_{TH} \quad \text{Linear, device acting as a resistance } (R_{ON})$$

$$I_{DS} = \mu_n \cdot C_{ox} \cdot \left(\frac{w}{l}\right) \cdot \left(V_{GS} - V_{TH} - \frac{V_{DS}}{2}\right) \cdot V_{DS} \quad (1)$$

$$R_{ON} = \left(\frac{1}{\mu_n \cdot C_{ox} \cdot \left(\frac{w}{l}\right) \cdot \left(V_{GS} - V_{TH} - \frac{V_{DS}}{2}\right)} \right) \quad (2)$$

$$(ii) \quad V_{DS} \geq V_{GS} - V_{TH} \quad \text{Saturation}$$

$$I_{DS} = \frac{1}{2} \cdot \mu_n \cdot C_{ox} \cdot \left(\frac{w}{l}\right) \cdot (V_{GS} - V_{TH})^2 \cdot (1 + \lambda V_{DS}) \quad (3)$$

In this operation region, where I_{DS} is high, the current is minimally dependent on the value of V_{DS} , but its maximum value is controlled by V_{GS} . λ is the channel length modulation, which accounts for the increase in I_{DS} with an increase in V_{DS} in field-effect transistors, because of pinch-off. Pinch-off occurs when both V_{DS} and V_{GS} determine the electric field pattern near the drain region. This effect shortens the effective

length of the channel and increases I_{DS} . Ideally, λ is equal to zero so that I_{DS} is completely independent of V_{DS} value in the saturation mode [6].

3.1.3. I_{DS} vs V_{GS} characteristics of a n-channel MOSFET

Fig. 6 shows the I-V transfer characteristics of an NMOS transistor (with a channel length of 100 nm), obtained by grounding the source terminal, stepping V_{DS} at two predefined values (low and high), while sweeping V_{GS} from zero to V_{DD} . At the low V_{DS} value (typically 50–100 mV), we obtain the transistor characteristics while operating in the linear mode. While at high V_{DS} values (typically 1.5–2 V), we obtain the transistor characteristics while operating in the saturation mode.

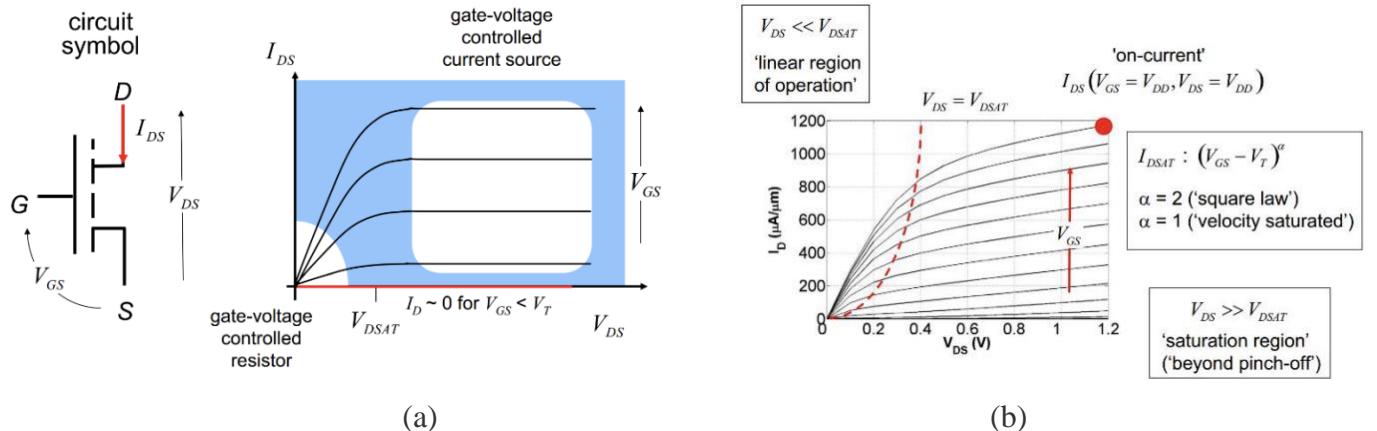


Fig. 3.4. I-V output characteristics of NMOS transistor (a) Ideal (b) with Channel Length Modulation [7]

The threshold voltage (V_{TH}) is the gate-source voltage V_{GS} at which drain-source charge flow begins. The pinch-off voltage ($V_{GS} - V_{TH}$) is a drain-source voltage V_{DS} at which saturation occurs.

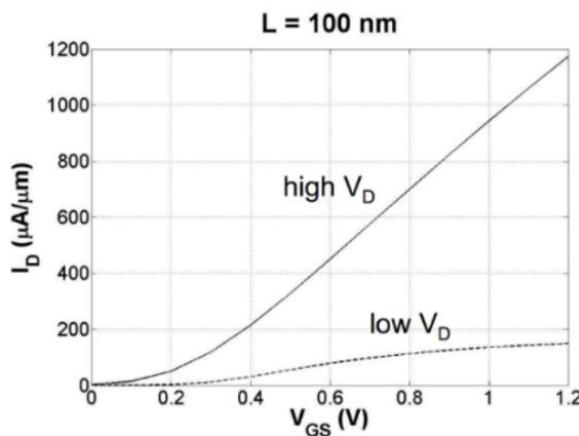


Fig. 3.5. I-V transfer characteristics of an NMOS transistor [7]

Finally, the transconductance (g_m) of a transistor is defined as a measure of the variation in I_{DS} with respect to V_{GS} at a constant V_{DS} :

$$g_m = \frac{\partial I_{DS}}{\partial V_{GS}} \Big|_{V_{DS}=\text{const.}} \quad (4)$$

3.1.4. Subthreshold Swing (SS) of NMOS transistor

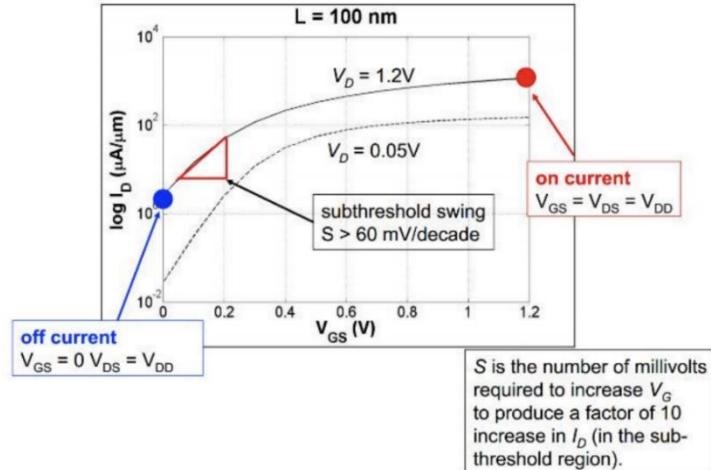


Fig. 3.6. Demonstration of the subthreshold swing extraction [7]

The subthreshold swing quantifies the ability of a transistor to make transition between the ON state and OFF state, and it is given as follows:

$$SS = \frac{\partial V_{GS}}{\partial (\log_{10} I_{DS})} \quad (5)$$

Sub-threshold Swing is defined as the change in V_{GS} per one decade of I_{DS} . Small subthreshold swing means better channel control, e.g. improved I_{on}/I_{off} , which usually means less leakage, and less energy. For subthreshold circuits it also means better performance. Silicon MOSFETs have a theoretical minimum subthreshold swing of about 60mV/decade for room temperature [7]. At a constant V_{DS} , the drain current I_{DS} at $V_{GS} = 0$ is called the OFF current and the same at $V_{GS} = V_{DD}$ is called the ON current. A high on-off ratio means a low leakage current i.e. an improved device performance.

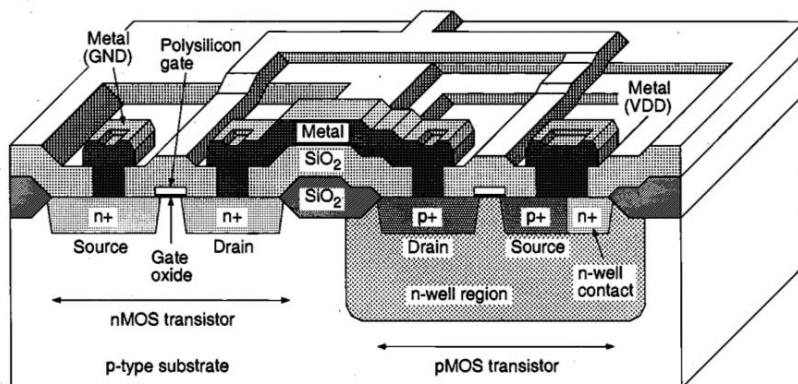


Fig. 3.7. Fabrication of CMOS using N-WELL [8]

3.2. FinFET

A FinFET (Fin Field-Effect Transistor) is a type of transistor structure used in semiconductor device fabrication. It is a three-dimensional transistor design that offers improved performance and power efficiency compared to traditional planar MOSFETs (Metal-Oxide-Semiconductor Field-Effect Transistors). In FinFETs, the conducting channel is formed by a thin silicon "fin" protruding from the substrate, which allows for better control of current flow. This design reduces leakage current and improves electrostatic control, enabling enhanced switching speed and lower power consumption. FinFET technology has become increasingly important in modern semiconductor manufacturing processes, particularly in advanced nodes, where continued scaling is challenging. It has enabled the development of smaller, faster, and more energy-efficient electronic devices across various applications, from mobile devices to high-performance computing systems.

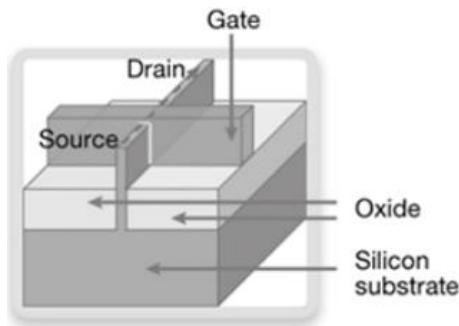


Fig. 3.8. Structure of FinFET [2]

This is the basic structure of basic finFET which is similar to traditional MOSFET. The same parameters 'W' and length 'L' is used for the finFET as well.

The width 'W' is given by [4]

$$W = 2 \times H_{fin} + W_{fin} \quad (6)$$

where, H_{fin} = vertical height of the fin

W_{fin} = width of the fin

Similar to MOSFET, n-channel finFETs and p-channel finFETs exists and are shown in Fig.3.9.

The operation of the n-channel finFET and the p-channel finFET are similar to that of p-mos and the n-mos transistors.

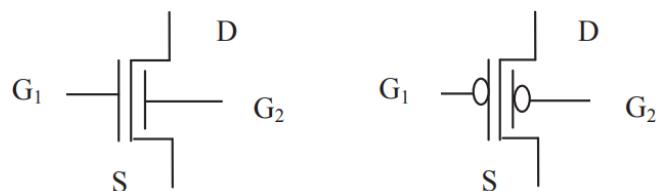


Fig. 3.9. Symbol of n-channel and p- channel FinFET [2]

The below plot represent the I_{ds} vs V_{ds} characteristics for n-channel finFET for different values of V_{gs} respectively.

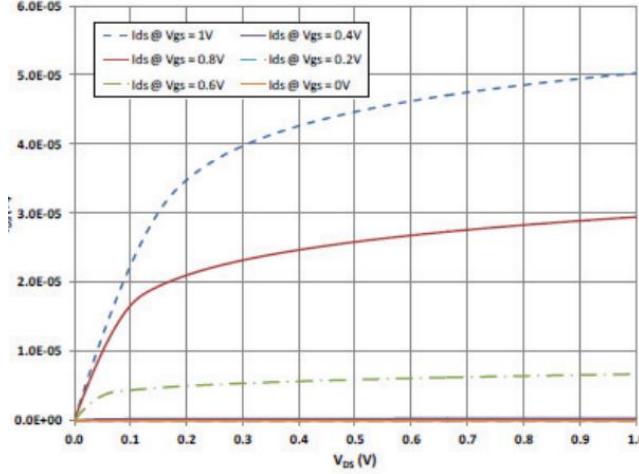


Fig. 3.10. I_{ds} vs V_{ds} for n-channel FinFET [2]

3.2.1. Different configurations of finFET Inverter

There are 3 different kinds of basic configurations of finFET are available [4].

- (i) Shorted gate configuration (SG)
- (ii) Low Power Configuration (LP)
- (iii)Independent Gate Configuration (IG)

3.2.1.1. Shorted gate configuration (SG) Inverter

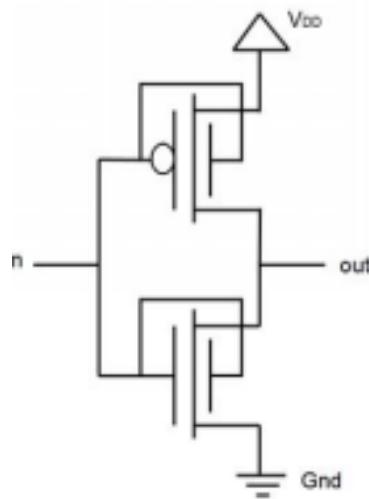


Fig. 3.11. Shorted gate configuration

In this configuration, the back gate and front gate for the n-channel finFET and p-channel finFET will be shorted means that both the gates will be given a common voltage. Since it is like having a single gate, the leakage and switching power consumption is more.

3.2.1.2. Low Power Configuration (LP) Inverter

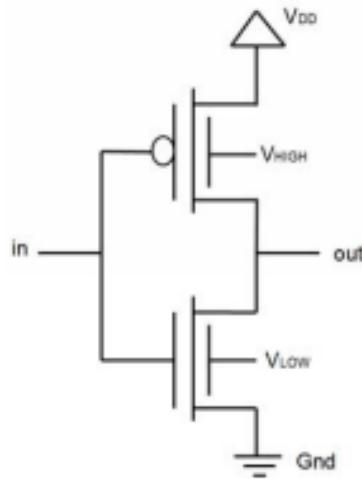


Fig. 3.12. Low Power Configuration (LP)

In this configuration, the back gate and front gate for the n-channel finFET and p-channel finFET are connected to voltage levels HIGH and LOW respectively.

3.2.1.3. Independent Gate Configuration (IG) Inverter

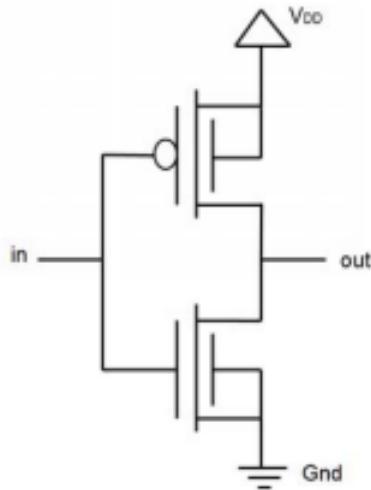


Fig. 3.13. Independent Gate Configuration (IG)

This configuration is similar to CMOS configurations. The front gate and back gate will be connected independently. Leaking currents are almost equal to that of SG configuration but the dynamic power is less than that of SG configuration.

3.3. Memristor

3.3.1. Relation of electrical parameters

Capacitor, resistor and inductor are three fundamental circuit elements. These circuit elements are defined by the relation between two of the four fundamental circuit variables current, voltage, charge and flux. Prof. Leon Chua [15] proposed that there should be a fourth fundamental circuit element which gives

the relation between flux and charge. Memristors are new passive circuit elements with interesting non-linear and memory properties that can be used for various forms of computations. Memristors have two primary properties: memory and resistance. The current through the memristor is a function of voltage and resistance because it is a passive element with resistive behavior. Unlike resistors, memristors can have varying resistances based on the previous voltages applied. Fig. 3.15 shows the relation between fundamental circuit elements [16].

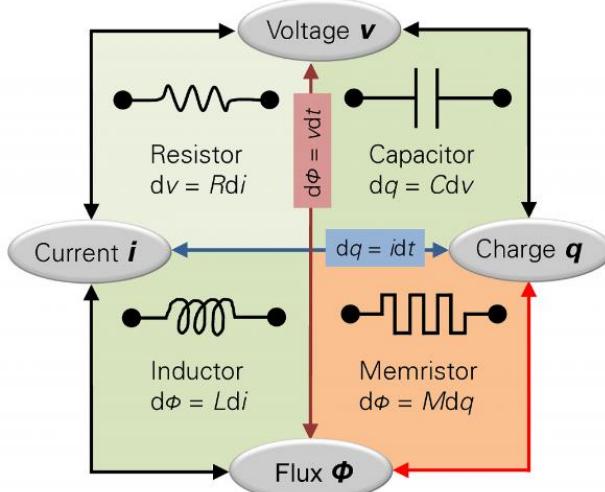


Fig. 3.14. Relation between electrical parameters

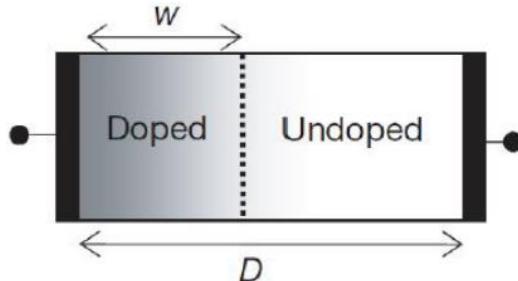


Fig. 3.15. HP memristor model

Memristor can be implemented by two layers of TiO_2 or MoO_x/MoS_2 or ferroelectric etc. in which one layer is doped with oxygen vacancies and the other layer is undoped.

$$R_{mem} = R_{on}x + R_{off}(1 - x) \quad (7)$$

Where R_{on} is the resistance in the doped region, R_{off} is the resistance in the undoped region, w is the width of the doped region and D is the memristor width.

$$x = \frac{w}{D} \quad (8)$$

Where x is the state variable of the device which is the ratio between the width of the doped region to the width of the memristor.



Fig. 3.16. Symbol of Memristor

In memristor dark colour thick line represents the polarity of the device. When the current flows into the device conductivity increases i.e., resistance decreases and when current flows out of the device conductivity decreases i.e., resistance increases.

A strong electric field is created in the device with only a few volts applied, leading to a high non-linearity in the ionic drift diffusion. Many attempts have been made to incorporate nonlinear behaviour into the state equation in order to solve this issue. The initial approach looked at multiplying the state equation's second part by a "window function," $f(x)$, in order to get overcome this limitation.

3.3.2. Hysteresis Loop

The nature of the nonlinearity can be seen clearly by tracing the response of the device to a sinusoidal signal. The plot starts at zero volts and zero amperes. As the voltage steadily increases, current rises at an accelerating rate reflecting the nonlinear memristance. Then, after the voltage reaches its maximum and starts to fall again, the current continues to rise briefly because the resistance of the TiO_2 film is still diminishing. When the current finally does pull back, the descending branch of the curve does not retrace the path of the ascending branch. Instead, it forms a loop, called a hysteresis loop [17].

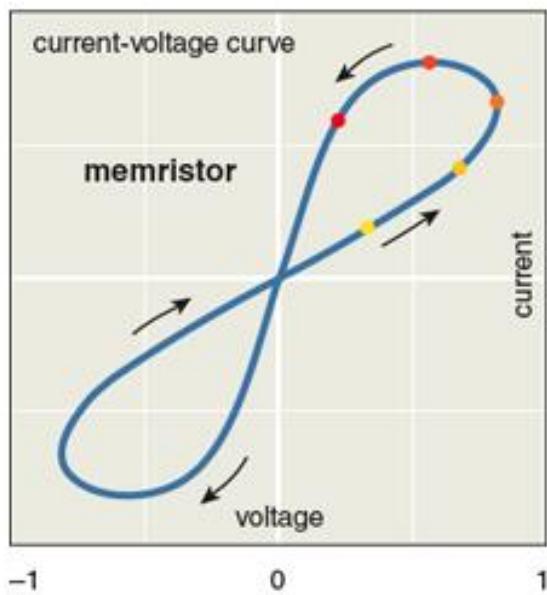


Fig. 3.17. Hysteresis curve of the memristor

3.3.3. Window Functions

In [18] different kind of window functions was presented for memristor model by Strukov, Joglekar, Biolek and Prodromakis.

First parabolic window function was proposed by Strukov as shown in equation (9).

$$f(x) = x - x^2 \quad (9)$$

This window function is capable to resolves the boundary effect problem, because at the boundary of the device (i.e. $x = 1$ or $x = 0$) interface drifting become zero with $f(x) = 0$, but it is poor in scalability. So, Joglekar proposed another window function with one control parameter to tune the non-linearity of the device given in equation (4).

$$f(x) = 1 - (2x - 1)^{2p} \quad (10)$$

Where ‘p’ should be positive integer only. Even then, it suffers from boundary stick issue, means that, if one’s interface hits any boundary of the device ($x = 0$ or 1), it is not capable to tune the resistance of the memristive device from its terminal resistance (R_{on} or R_{off}).

Thereafter, Biolek proposed a new window function to resolve the boundary stick issue by including device current parameter in window function. This window function is represented by expression-5, which have one control parameter.

$$f(x) = 1 - [x - stp(-i)]^{2p} \quad (11)$$

where p is a positive integer and $stp(-i)$ is a step function. However, as uncertainty in the direction of current, the continuous characteristic of the function cannot be satisfied. At the same time, the flexibility of the window function is limited because it lacks a parameter to control amplitude. By considering above limitations, Prodromakis recommended a window function with two control parameter which has given in expression-6.

$$f(x) = j \times [1 - \{(x - 0.5)_2 + 0.75\}]^p \quad (12)$$

where, ‘p’ and ‘j’ belongs to positive real number, which provides better flexibility to the window function. For all the windows functions, we can conclude the Joglekar model will result in higher currents, we can also conclude, by analyzing the previous figures that Joglekar model will also give the biggest range in the resistance values for the memristor.

Various mathematical concepts have been proposed to describe the behaviour of the memristor like Non-linear model, Linear Drift model, Simmons tunnelling model, Threshold adaptive model (TEAM), Voltage threshold adaptive model (VTEAM) [18].

Linear Drift model is inaccurate as compared to the physical memristive devices. Then, the Simmons tunnel model was analyzed. This model assumes nonlinear and asymmetric switching behavior. It gives a

higher accuracy in modeling real memristor but, it offers no explicit relation between current and voltage and its very complicated model, which results in it hard to simulate. The TEAM model was then developed to simplify the Simmons model. It achieves this by assuming two different thresholds for the current that passes through the device. It also assumes a polynomial dependence between the memristor current and the internal state derivative. Another advantage is that it can be fitted to different types of memristor models. The VTEAM model is an adaptation of the TEAM model, but is voltage controlled instead of current-controlled, it was created to take advantage of the simplicity and accuracy of the TEAM model. The primary change is the addition of voltage thresholds in place of current thresholds; this is significant while certain memory and logical applications require voltage thresholds.

3.4. Total Power Analysis

The total power of a circuit is the sum of Static power dissipation, short circuit power and Dynamic power [9].

$$P_{total} = P_{static} + P_{Dynamic} + P_{short-circuit} \quad (13)$$

Static power is the power consumed when there is no circuit activity, often known as a quiescent mode. It is mostly caused by leakage currents that flow from when the transistor is off state [9].

$$P_{static} = V_{DD} \times I_{leak} \quad (14)$$

Dynamic power will arise when the capacitive load charges and discharges. Therefore, it depends on the switching frequency of circuit if it is operating in active mode [9].

$$P_{Dynamic} = \alpha \times f_{switch} \times C_L \times V_{DD}^2 \quad (15)$$

Here α represents the probability that the output switches between 0 to 1 in single clock, V_{DD} is the supply voltage, C_L is the load capacitance and f_{switch} is the switching frequency, t_{sc} is the time for which both the NMOS and PMOS conduct, I_{peak} is the current due to short-circuit and I_{leak} is the leakage current.

As the input progressively changes, there will be a period when the pull-up and pull-down networks are switched ‘ON’ at the same time, providing a short-circuit path from VDD to GND results in Short Circuit power dissipation.

$$P_{short-circuit} = t_{sc} \times f_{switch} \times I_{peak} \times V_{DD} \quad (16)$$

3.5. Propagation Delay

The propagation delay (t_{PD}) of a gate is defined as the time taken by a gate to respond when there is change on its inputs [10]. It expresses the delay experienced by a signal when passing through a gate. It is measured between the 50% transition points of the input and output waveforms as shown in Fig. 9 for an inverter. As a gate shows different response times for rising or falling input waveforms the propagation delay is defined

by two terms. The t_{pLH} defines the response time of the gate for a low to high output transition while t_{pHL} refers to high to low transition. The propagation delay (t_{pD}) is defined as the average of these two terms.

$$t_{pD} = \frac{t_{pLH} + t_{pHL}}{2} \quad (17)$$

The propagation delay is function of both circuit technology and the slopes of the input and output signals of the gate. To consider this effect, the terms rise time (t_r) and fall time (t_f) are introduced. The rise and fall times are defined between the 10% and 90% points of the waveform as shown in Fig. 3.19.

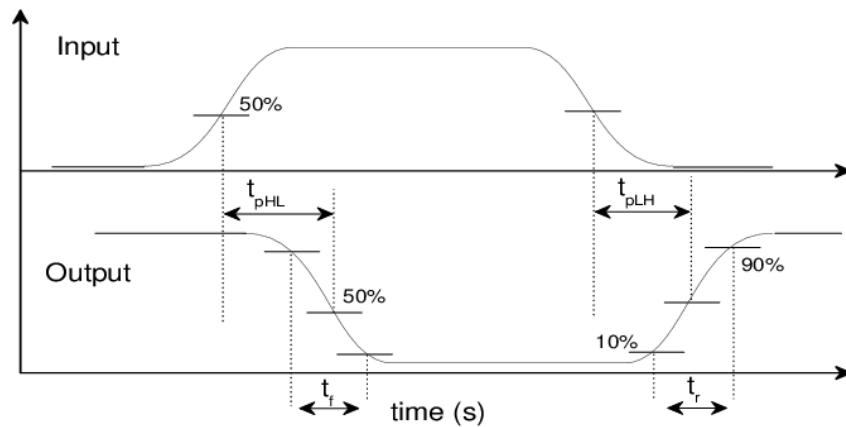


Fig. 3.18. Propagation delay of an Inverter

CHAPTER - 4

METHODOLOGY

4.1 Procedure for Designing and Simulating a circuit

4.1.1. Invoke Custom Compiler

Create a new working directory named CMOS. Start Custom Compiler from the working directory CMOS and run the command “cdesigner” in the Linux terminal window.

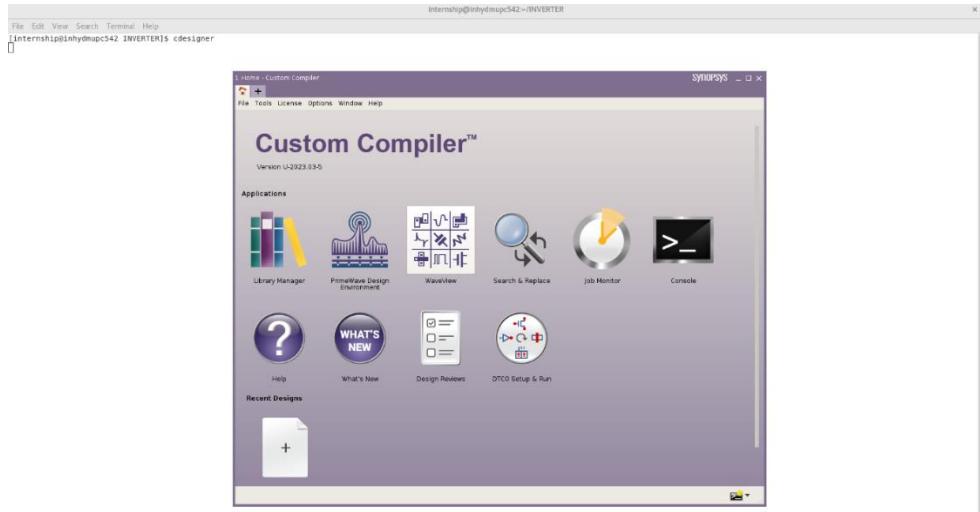


Fig. 4.1. Custom Compiler Window

- Select the Library Manager → File → New → Library in the Custom Compiler console window to create a new library.

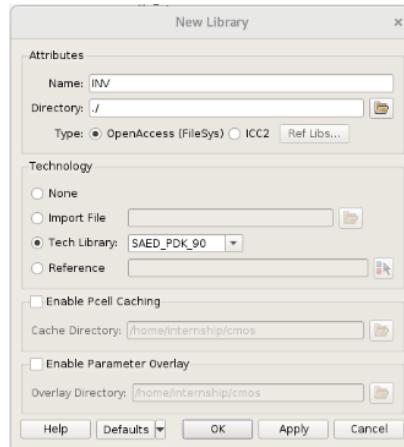


Fig. 4.2. New Library Window

4.1.2. Create a Library

Create a library named INV.

Check that the directory is in the current working directory by default.

Select SAED_PDK_90 in the Tech Library as shown in the Fig. 1.2. and press OK Technology File is required for the layout design.

4.1.3. Create a New Cell View

Select library (INV) in which a new cell is to be created.

Select File → New → CellView in Custom Compiler console window.

Type the name of the created cell under Cell Name

- Enter “inverter” under the Cell Name field.

Choose View Name of the cell as schematic.

Editor is automatically chosen.

After completing all the steps, verify the New CellView window as shown in Fig. 3 and press OK. The Schematic Editor window appears.

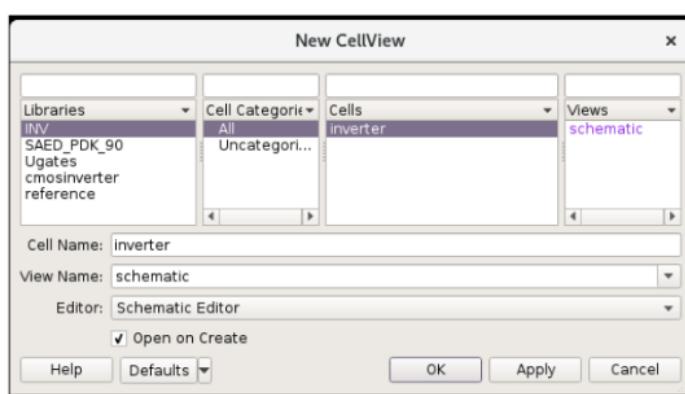


Fig. 4.3. New CellView Window

4.1.4. Schematic Editor (SE) – Basic Schematic Cell Creation

In the inverter Schematic editor window, select Add → Instance and add the instances of pmos4t and nmos4t from SAED_PDK_90 library and place them on the Schematic editor window as shown in the Fig.4.

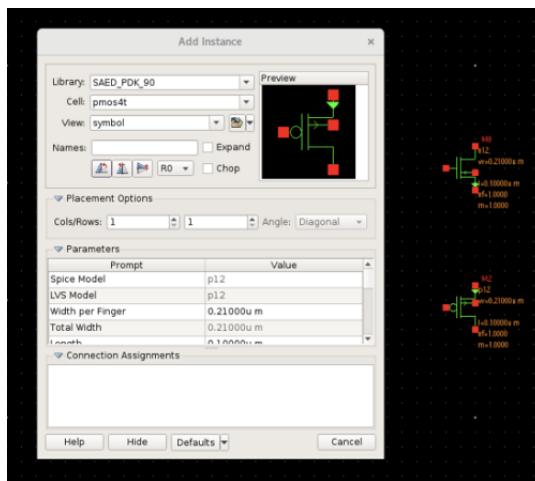


Fig.4.4. Placing the Instances

4.1.4.1. Editing Instance Parameters

To change the instance parameters, select the specific instance by single click and call Property Editor Window by pressing **Q**. Stick to the default parameters unless there is a specific significance of the parameters.

4.1.4.2. Add Wires

Connect the transistors as shown in the Fig. 6 with wires after pressing the key “**W**” on the keyboard.

4.1.4.3. Create Wire name

Wire name can be given manually by selecting **Add → Wire name**. Add the wire name to the schematic as shown in the below figure.

4.1.4.4. Add pins

Create pins to define the inputs and outputs of the circuit. Pins can be created navigating through **Add → Pin option**. Write the names “**in**, **vdd**, **vss**” in the Names line and select the pin type as **input** on the toolbar. Similarly, write “**out**” in the Names line and select the pin type as **output** on the toolbar and place them on the canvas as shown in the Fig. 5.

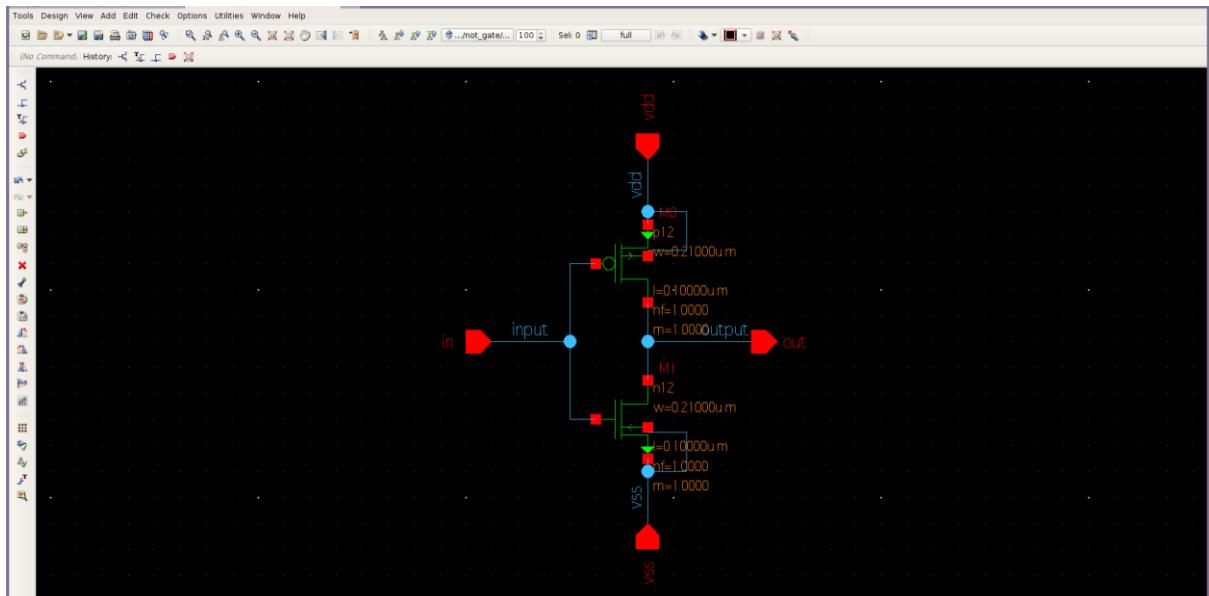


Fig. 4.5. Schematic of Inverter

4.1.4.5. Checking and Saving the circuit

Select **Design → Check and Save** and observe the **Console** window for any warning messages and if there are any conflicts in the circuit, markers are generated at the problematic regions. Make sure to clear them and save the circuit.

Checking "NOT_GATE/not_gate/schematic"
 No rule violations found in "NOT_GATE/not_gate/schematic"
 Information: Checking process is completed: marking the design violations
 Information: Marking of designs is completed

Fig. 4.6. Console Window

4.1.5. Schematic Editor (SE): Symbol Creation

Symbol can be generated for the schematic automatically by the following navigation option **Design → New CellView → From CellView.**

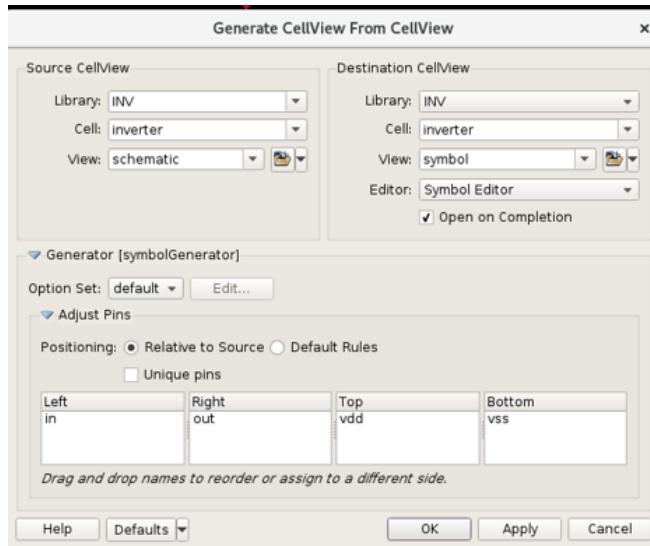


Fig. 4.7. Generate CellView from CellView

Library, Cell, and Schematic View are selected by default under Source and Destination Cell View. Pins positions can be adjusted irrespective of the default positions as shown in the Fig.7 and then press OK to generate the symbol view for circuit of inverter.

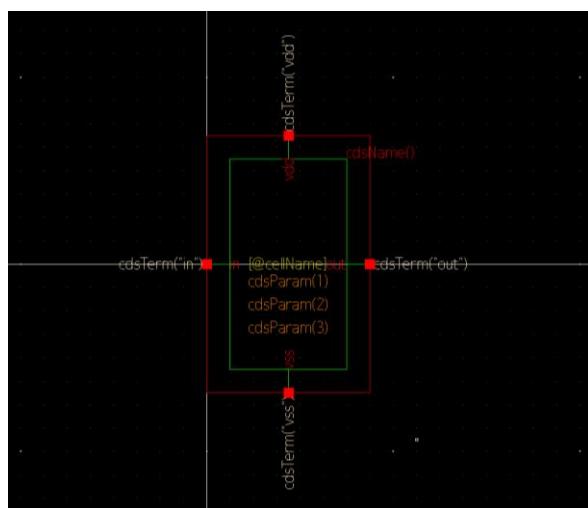


Fig. 4.8. Created Symbol for the Schematic

Edit the symbol as shown in the below Fig.9 and select the check→cross check view and observe the console window for warnings, if there are any warnings make sure to clear them and repeat the process to save the symbol.

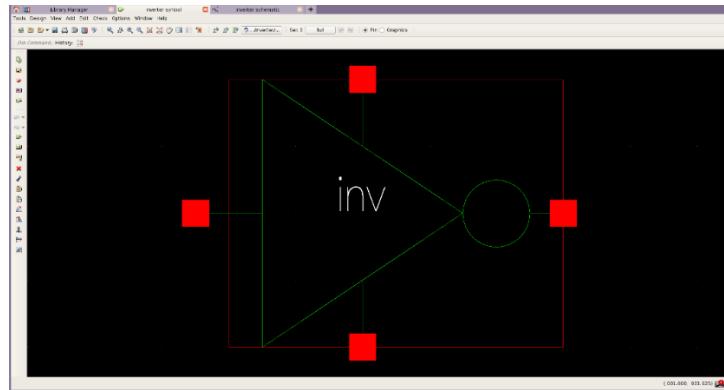


Fig. 4.9. Symbol of Inverter

4.1.6. Testbench Creation

For creating testbench, create a new schematic **cellview** named **inverter_tb**. Place the created **inverter** instance using **Add → Instance** option.

- Select Library **INV**
- Select Cell **inverter**

Click on the canvas to place inverter symbol. Add vdc, cap, vpluse, gnd components from the analogLib library and change their parameters. Save and close the design.

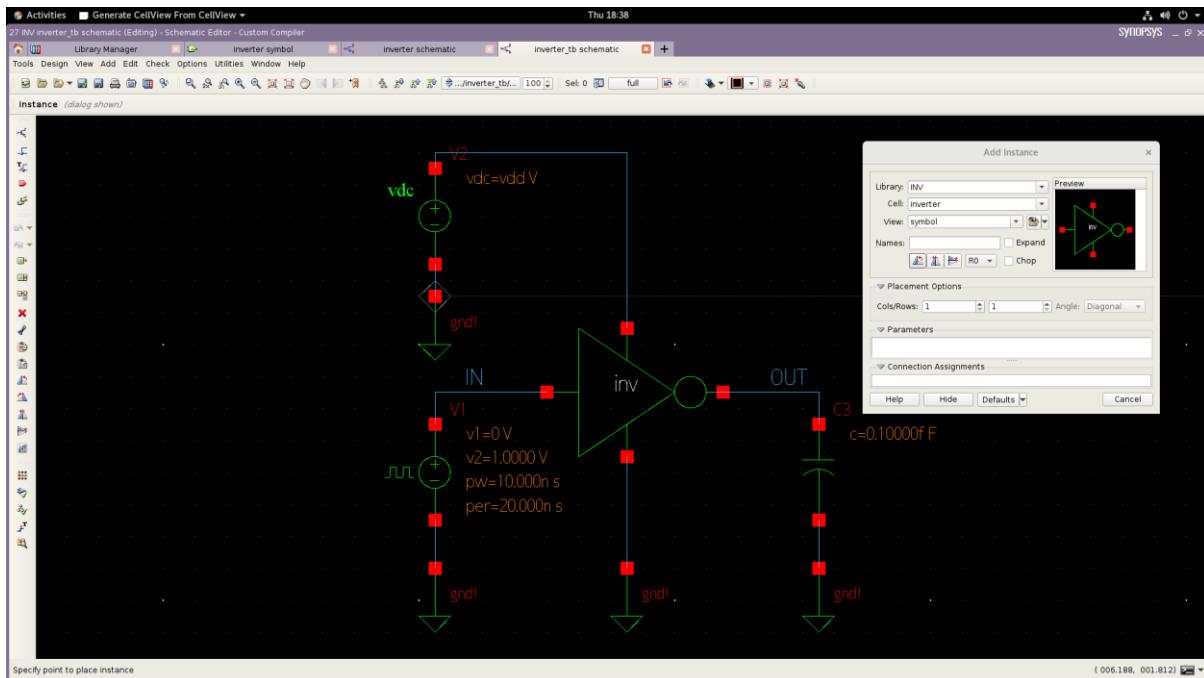


Fig. 4.10. Testbench View

The component parameters of the vpulse and vdc sources are shown in the below figure.

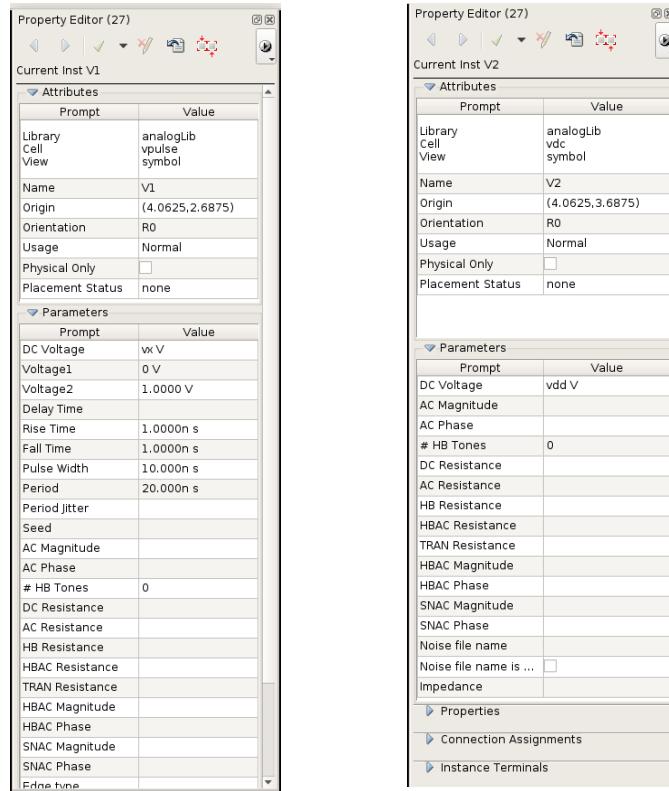


Fig. 4.11. vpulse, vdc component parameters from left to right.

4.1.7. Prime Wave Design Environment

From the Schematic Editor, choose Tools → PrimeWave. The PrimeWaveDesign Environment window opens.

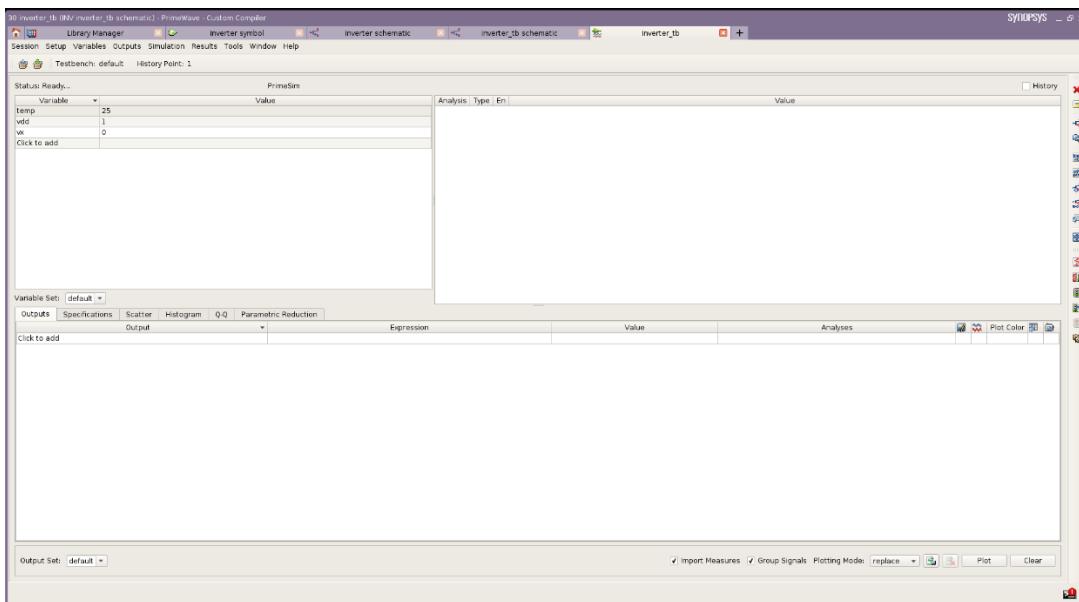


Fig. 4.12. Prime Wave Design Environment after adding the Variables

4.1.7.1. Add variables

- Click on the Variables menu of the testbench and Click on the Edit
- Click on Copy from Design. The variables are filled in automatically, Enter the values for the variables vdd and vx as shown in the Fig. 12.

4.1.7.2. Setup Model Files

Click on setup → Model files and include “..../SAED90_PDK/hspice/SAED90nm.lib” and Click in the Section field to open the Select Sections dialog box. Choose the TT_12 section as shown in figure.

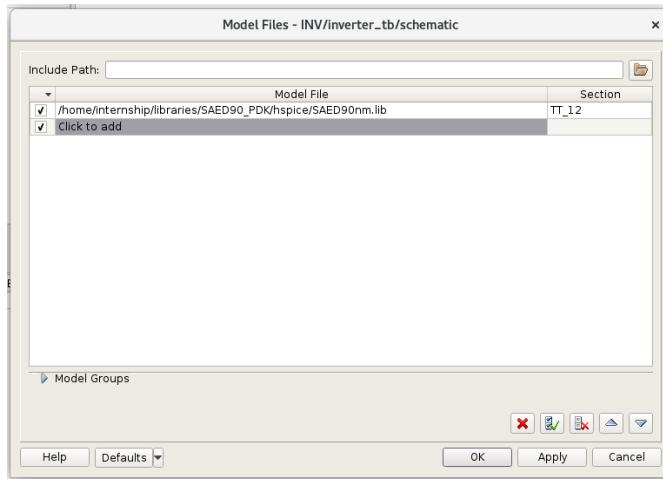


Fig. 4.13. Setting up Model Files

4.1.7.3. DC Sweep Analysis

Click on Setup and select Analyses in the testbench to open the Edit/Create Analyses dialog box. Choose dc as an analysis and check in the sweep option.

- In the sweep variable menu select Design Variable and choose vx variable, Setup the parameters as 0 to 1, number of steps is 20 as shown below and click OK.

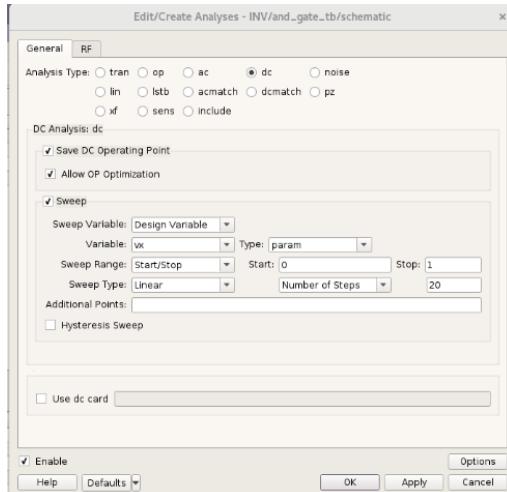


Fig. 4.14. DC Analysis Setup

Pick up the output, input node voltages from the design for the DC analysis simulation.

4.1.7.4. Transient Analysis

Click on Setup and select Analyses in the testbench to open the Edit/Create Analyses dialog box. Choose tran as an analysis and set up the window as shown below and click OK.

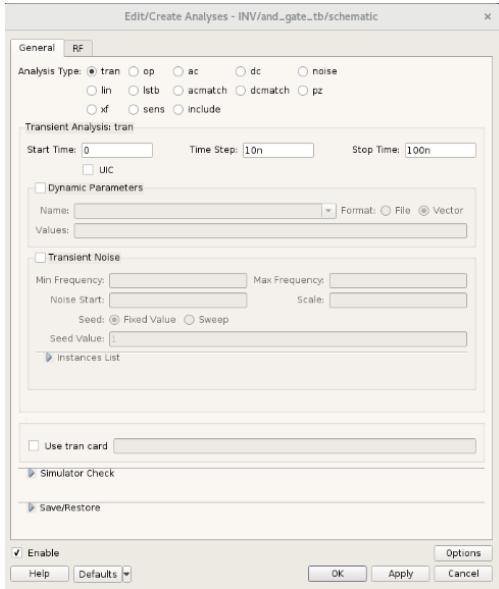


Fig. 4.15. Transient Analysis Setup

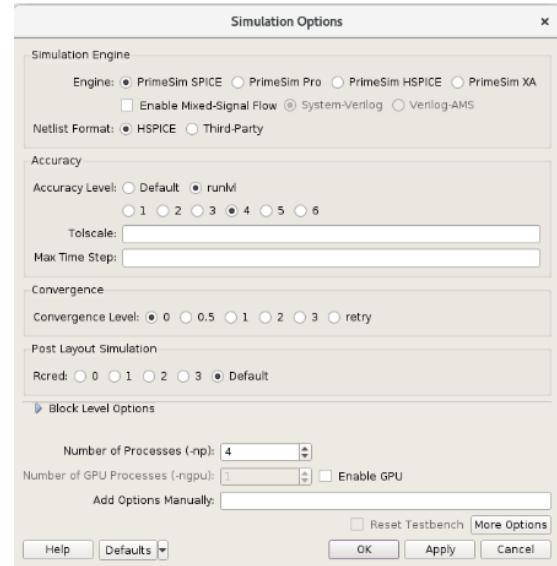


Fig. 4.16. Simulation options

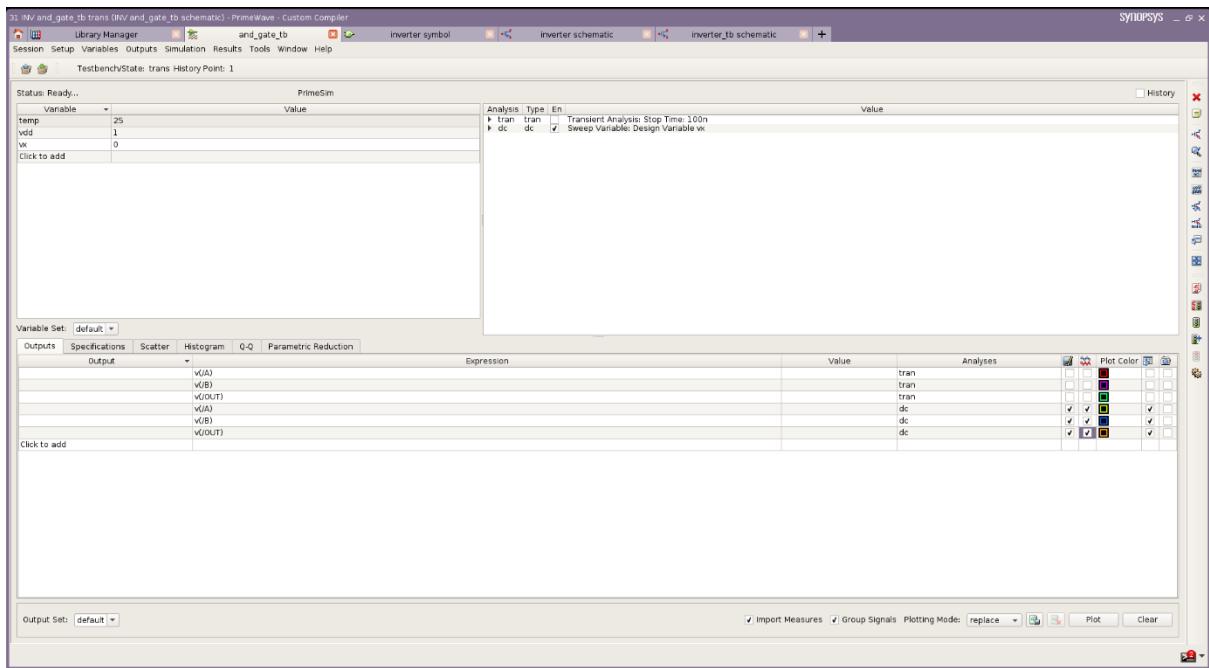


Fig. 4.17. DC Analysis Output Selection

Pick up the output, input node voltages simulation from the design for the transient analysis simulation.

- Click on Setup and select Simulator menu and make the changes in the Simulation Options window as

shown in the fig. 16.

The analysis and the outputs are to be selected as shown below.

- Choose Simulation → Netlist and Run, the waveform appears as shown in the following figure.

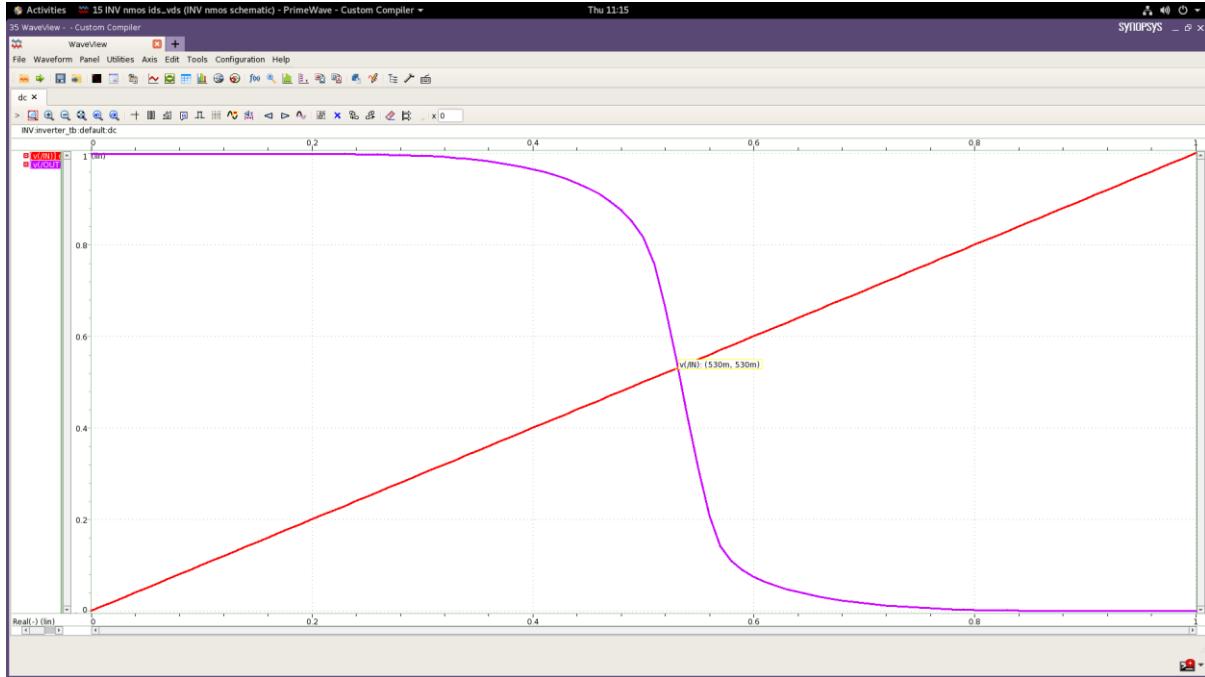


Fig. 4.18. DC Analysis (Voltage Transfer Characteristic Curve of the Inverter)

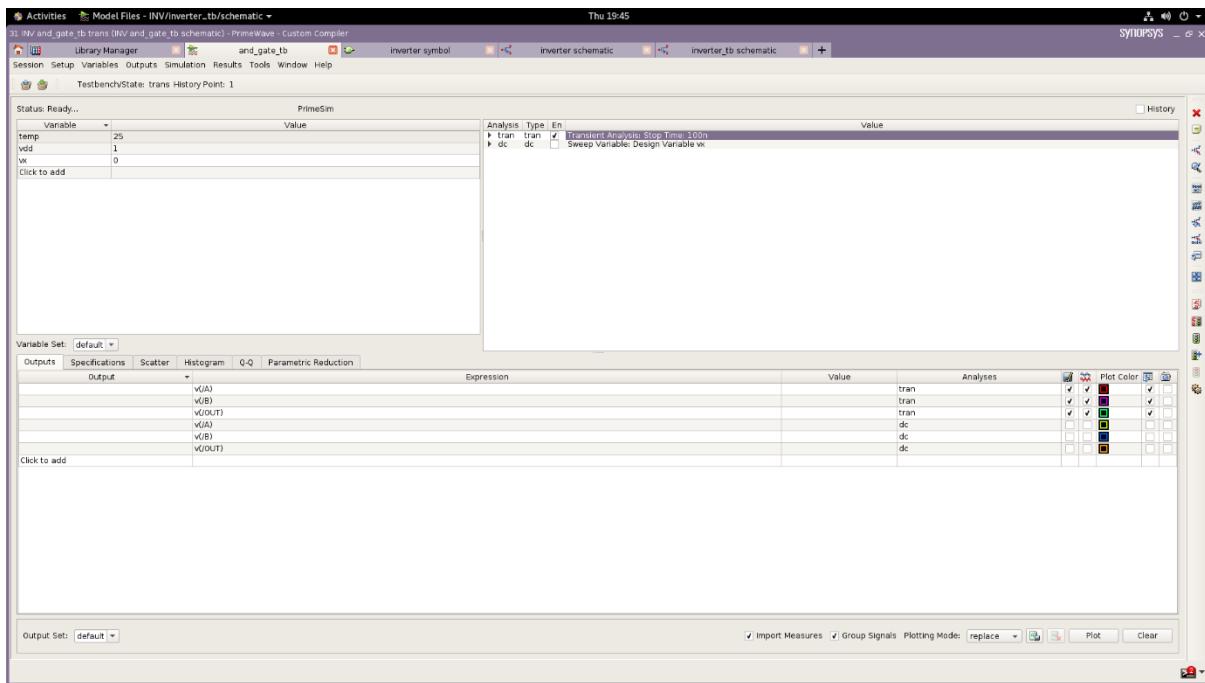


Fig. 4.19. Transient Analysis Output Selection

- Choose Simulation → Netlist and Run, the waveform appears as shown in the following figure.

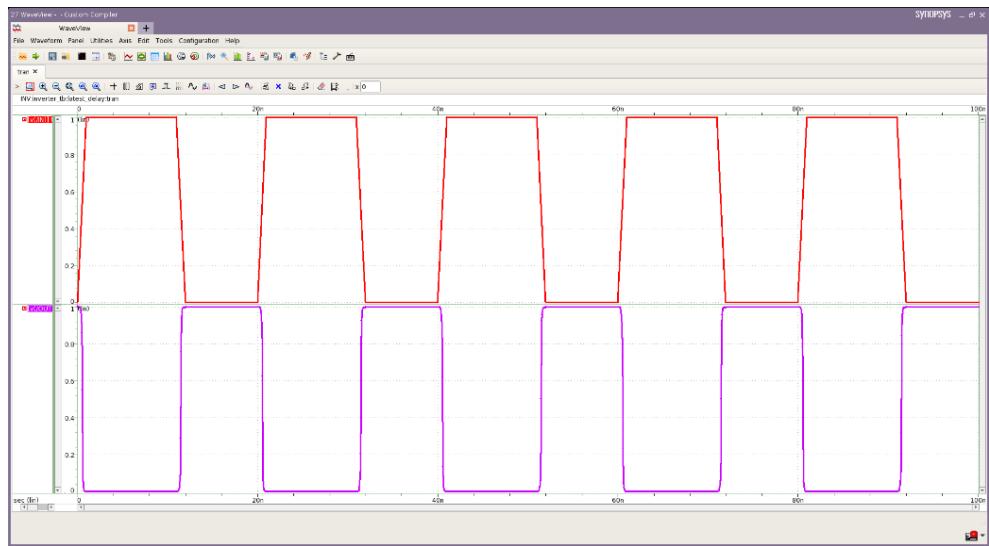


Fig. 4.20. Transient Analysis Waveform

4.1.7.5. Power and Delay Calculation

Power and delay are calculated considering only a single cycle of the input. Therefore, set the stop time to 20n and re-simulate the transient analysis.

In waveview window, after the wave appears, select utilities →calculator and Result Analyzer window opens. In the analysis, select trans and in the Plot Assist select power →Instance terminal and select the source terminal of the supply which is connected to the circuit. View the Results Analyzer window.

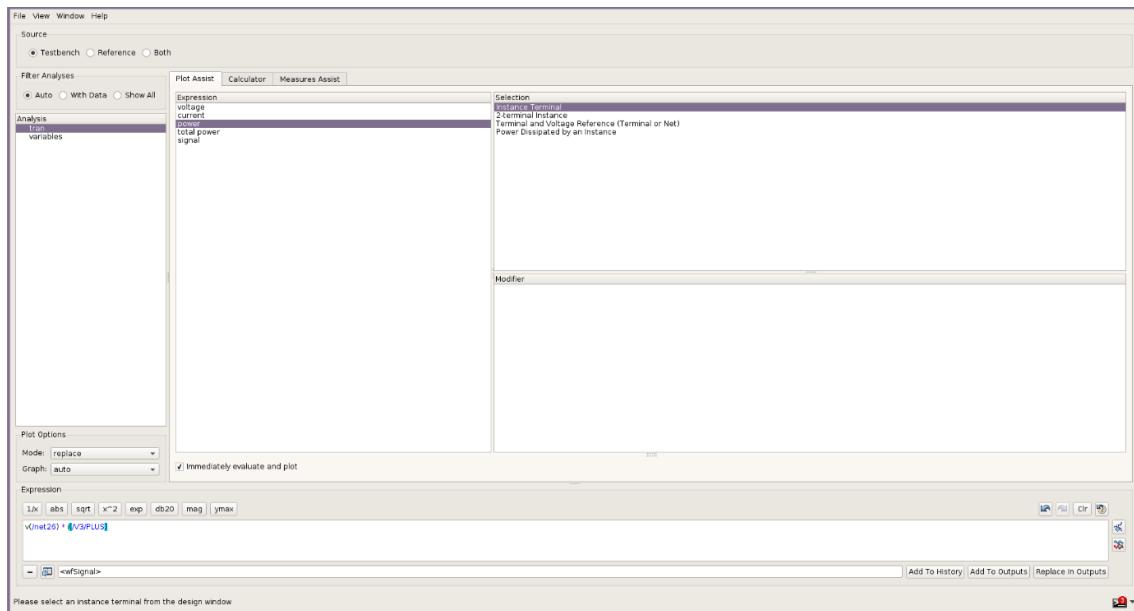


Fig. 4.21. Power calculation Setup

Tabulate the values by selecting table symbol besides the “=” symbol and in Waveview window click on equation and the following fig. appears.

$$\text{Total Power (P}_T\text{)} = | -6.63 | \mu\text{W} = 6.63 \mu\text{W}$$

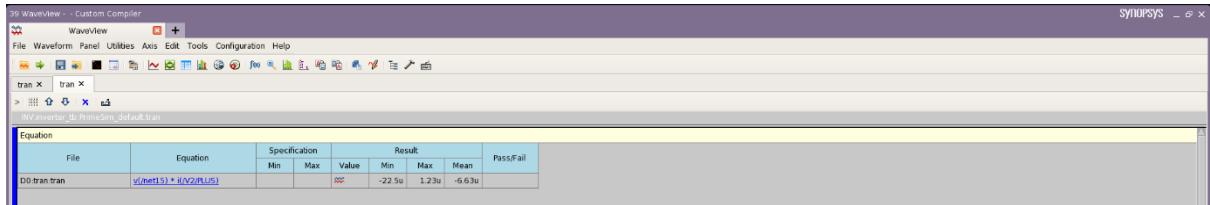


Fig. 4.22. Total Power of Inverter

To calculate the propagation delay, open the WaveView window, select the Utilities →Calculator →delay (double click).

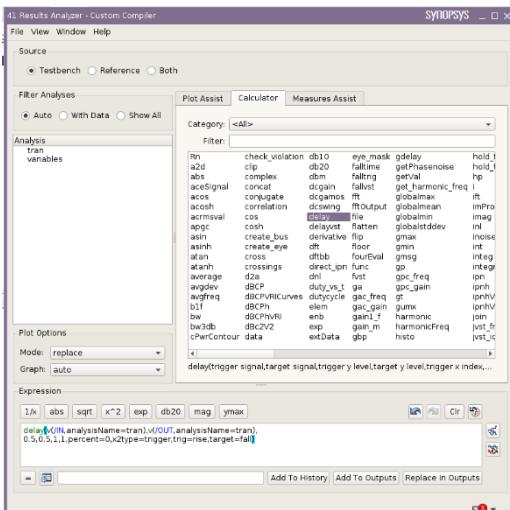


Fig. 4.23. tp_{LH} Setup

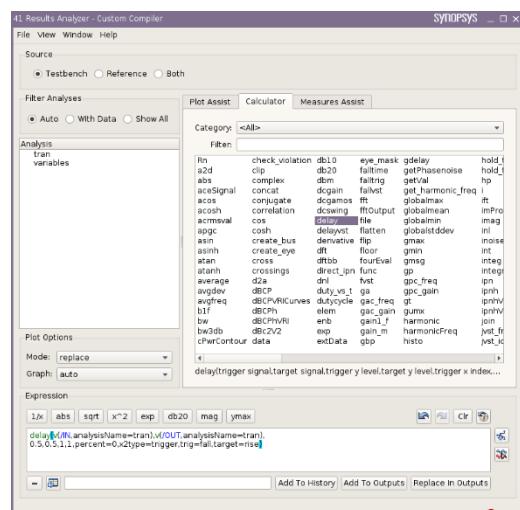


Fig. 4.24. tp_{HL} Setup

Tabluate the results as shown in the below Figures.

$$\text{Propagation Delay (t}_{pD}\text{)} = \text{Average (t}_{pLH}\text{, t}_{pH}\text{)} = \text{Average (56.4, -11.9)} = 22.25 \text{ psec.}$$



Fig. 4.25. t_{pLH} of Inverter

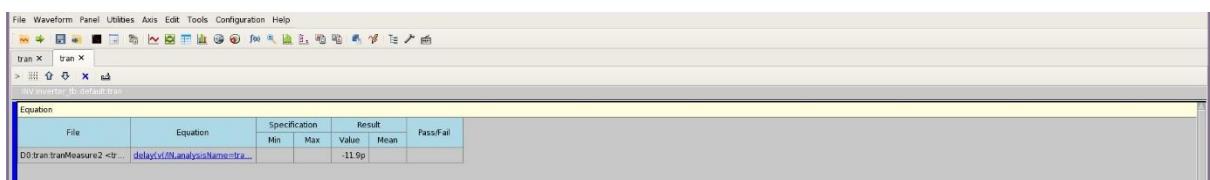


Fig. 4.26. t_{pHL} of Inverter

4.2 Generation of Memristor Symbol

- Create a new working directory named Memristor.
- In the Linux terminal window, change the current working directory to **Memristor**.
- Start Custom Compiler from the working directory **Memristor** and run the following command:



Fig.4.27. Custom Compiler Window

- Create a new library and Select **Library Manager** --> **File** --> **Import** --> **Netlist Text** in the Custom Compiler console window.

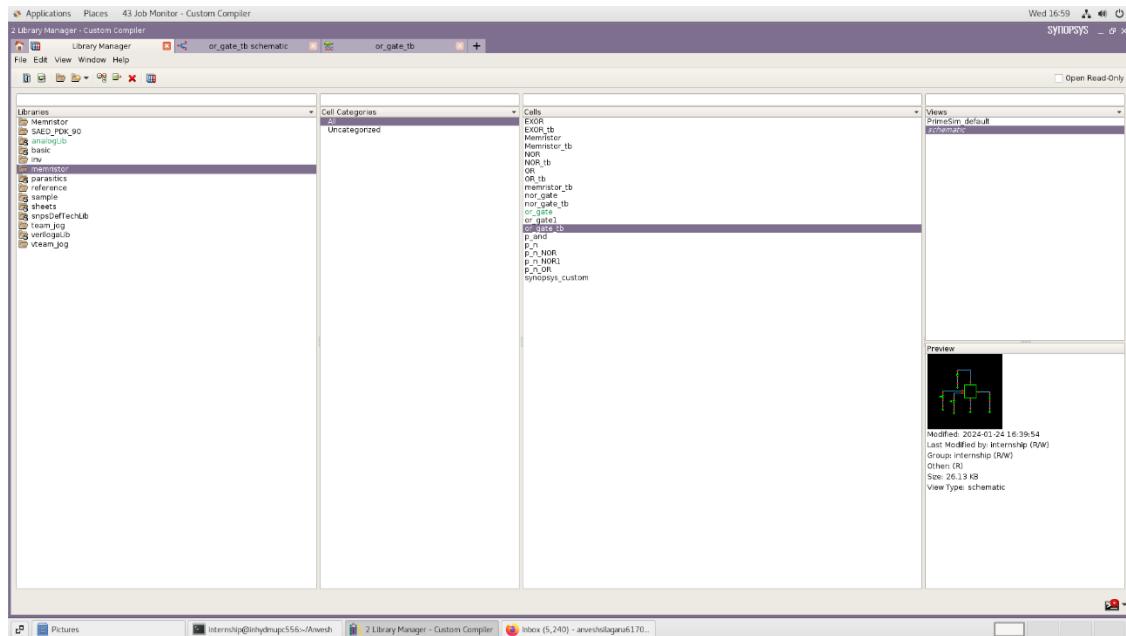


Fig.4.28. Library Manager

- After that a popup window Import Text will open.

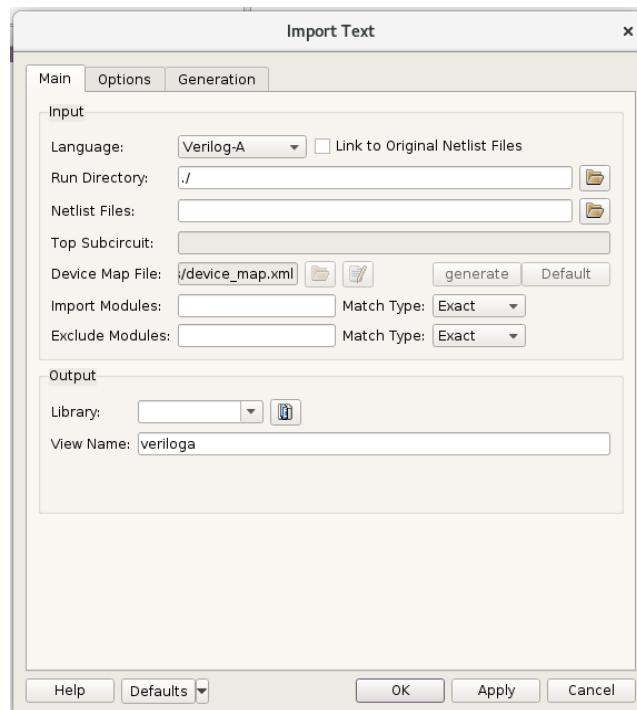


Fig.4.29. Import text pop up window

- Select the language as Verilog-a and then click on the Netlist Files and include Verilog-a code file.

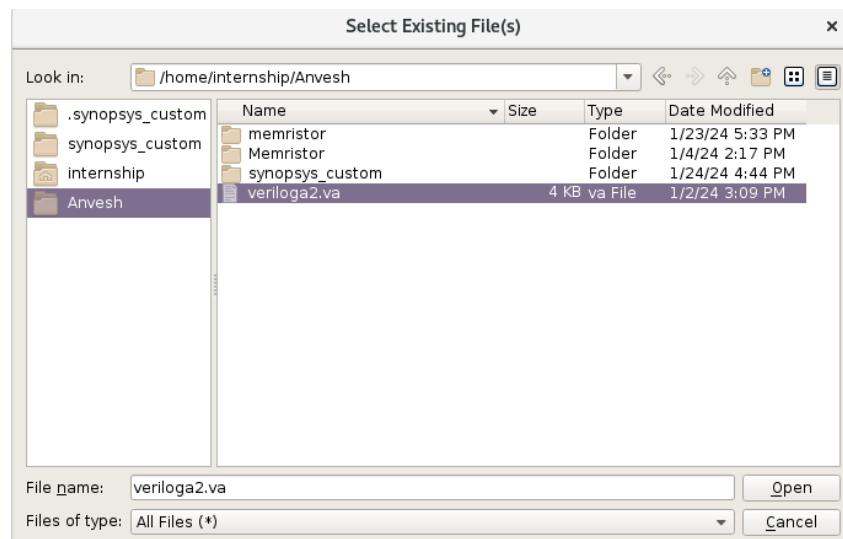


Fig.4.30. Netlist file

- Create a library named **Memristor**.
- Check that the directory is in the current working directory by default.
- Enter **saed90nm_1p9m_cd.tf** technology file available in <environment_path>/SAED90_PDK/techfiles as shown in the following figure.

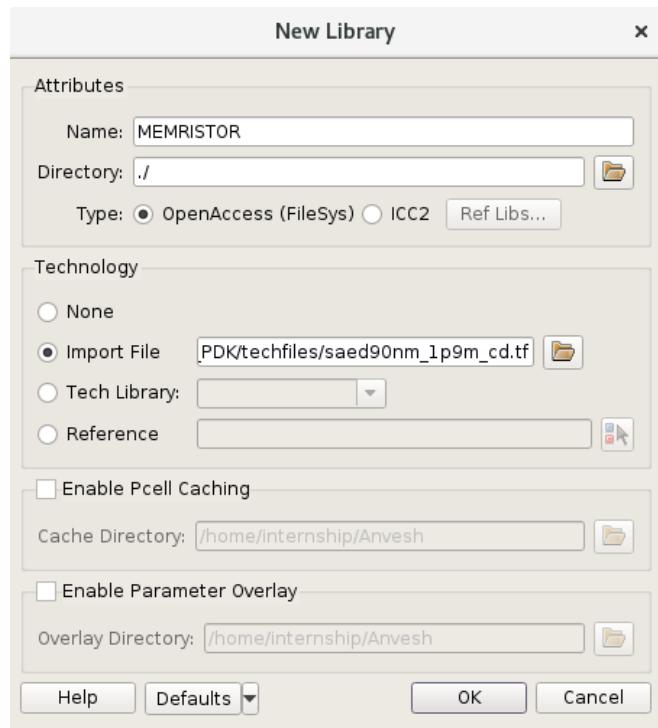


Fig.4.31. New Library

- Tick on the generate under the symbols in Generation.

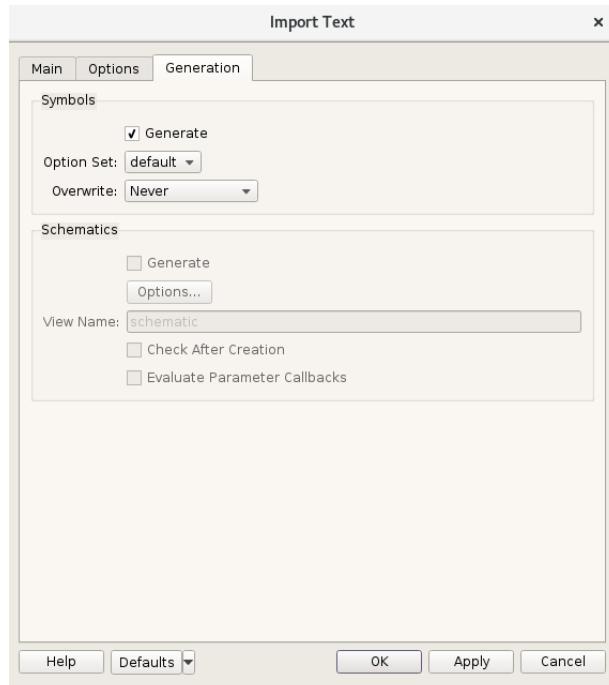


Fig.4.32. Pop up window for symbol generation

- After that Library named memristor is appeared in the library manager and its symbol also generated.

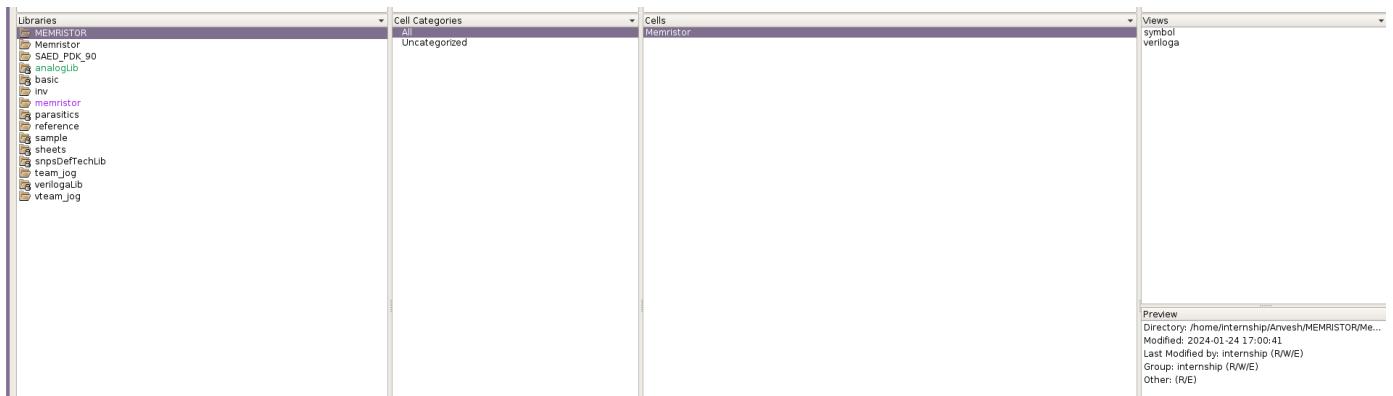


Fig.4.33. Memristor Library

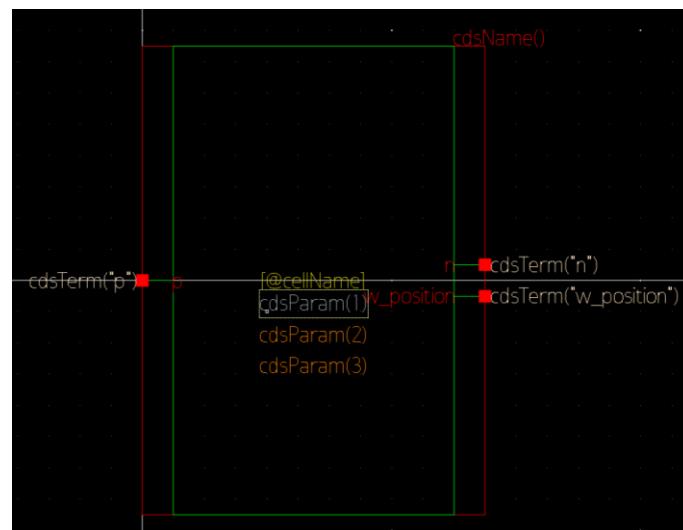


Fig.4.34. Generated symbol

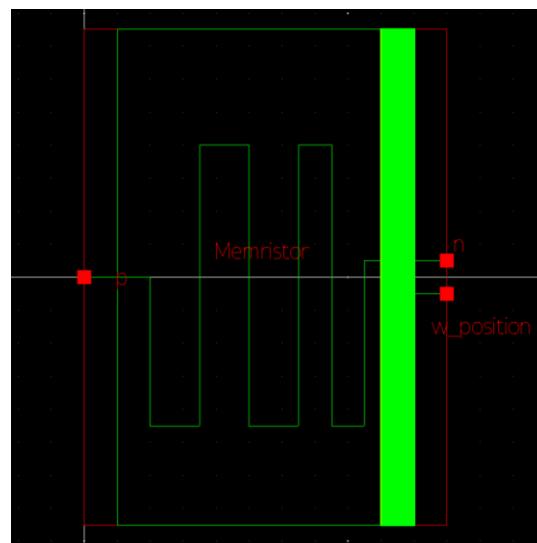


Fig.4.35. Memristor Symbol

4.3. DESIGN OF FUNDAMENTAL LOGIC GATES

4.3.1. Inverter

- A NOT gate, also known as an inverter, is a fundamental logic gate in digital electronics. Its primary function is to produce the logical complement of its input signal. In other words, if the input is high (1), the output will be low (0), and vice versa.

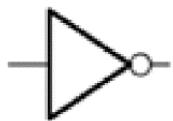


Fig. 4.36. Symbol of Inverter [11]

INPUT	OUTPUT
A	
0	1
1	0

TABLE 1: Truth table of Inverter [11]

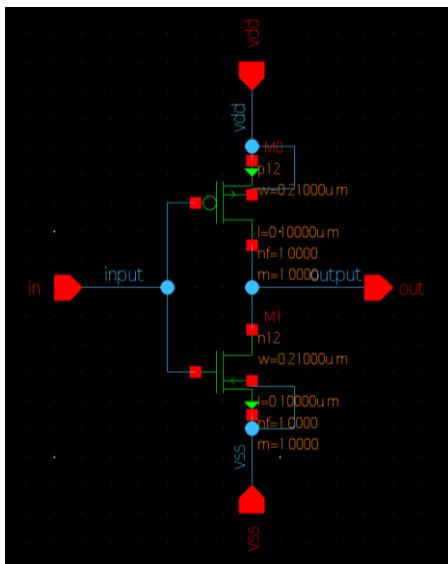


Fig.4.37. Schematic of Inverter using CMOS

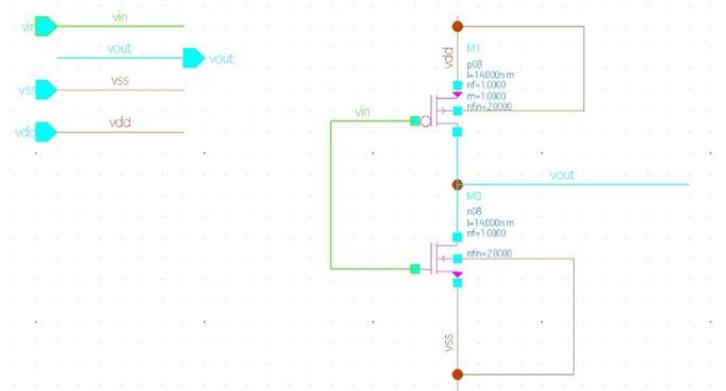


Fig.4.38. Schematic of Inverter using finFET

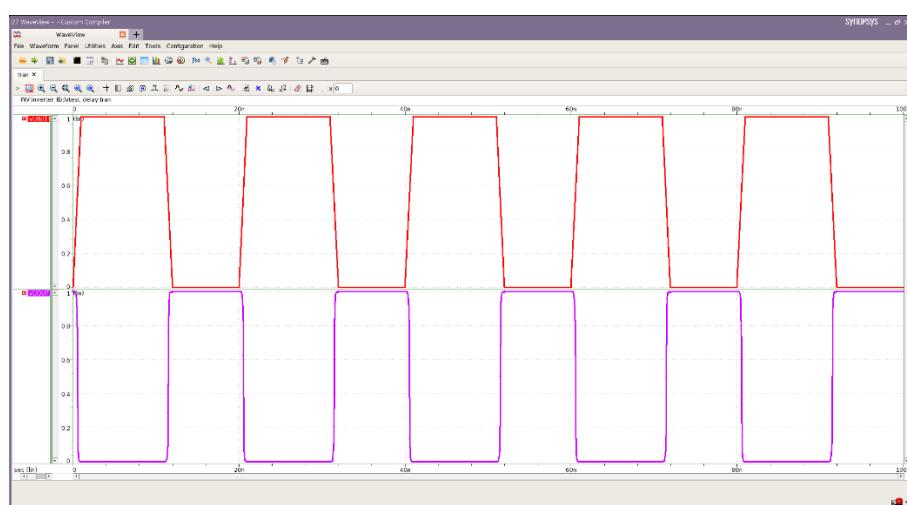


Fig. 4.39. Transient Analysis of Inverter

4.3.2. OR gate

- The operation of an OR gate is based on the logical OR function. If any of the input signals (A or B) is high (1), the output (Y) will be high (1). Only when both inputs are low (0), the output will be low (0).
- Mathematically, the output (Y) of an OR gate with two inputs (A and B) is given by the Boolean expression: $Y = A + B$.
- The OR gate implements logical disjunction, where the output is true if at least one input is true.



Fig. 4.40. Symbol OR gate [11]

INPUT		OUTPUT
A	B	
0	0	0
1	0	1
0	1	1
1	1	1

TABLE 2: Truth table of OR gate [11]

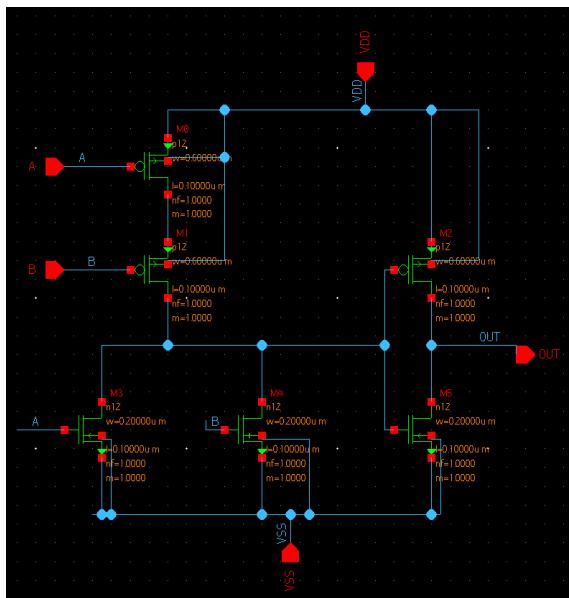


Fig. 4.41. Schematic of OR gate using CMOS

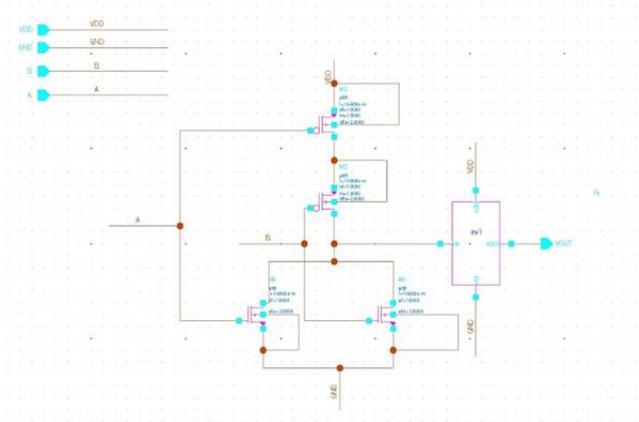


Fig. 4.42. Schematic of OR gate using FinFET

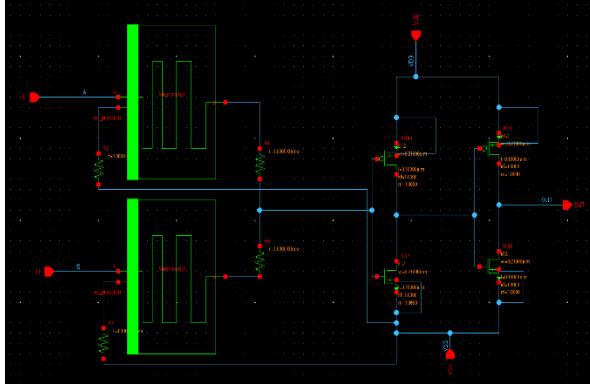


Fig. 4.43. Schematic of OR gate - Memristor

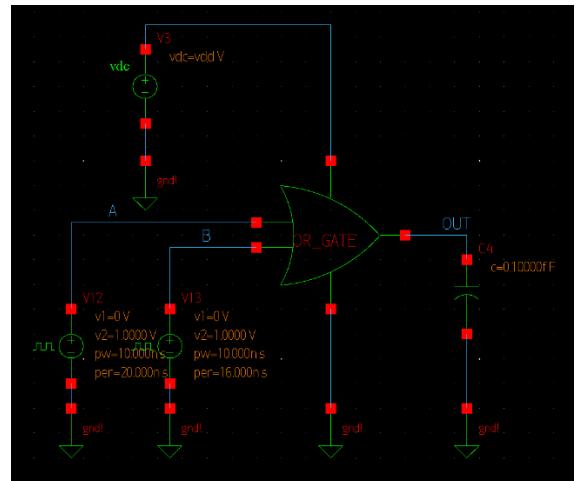


Fig. 4.44. Testbench of OR gate

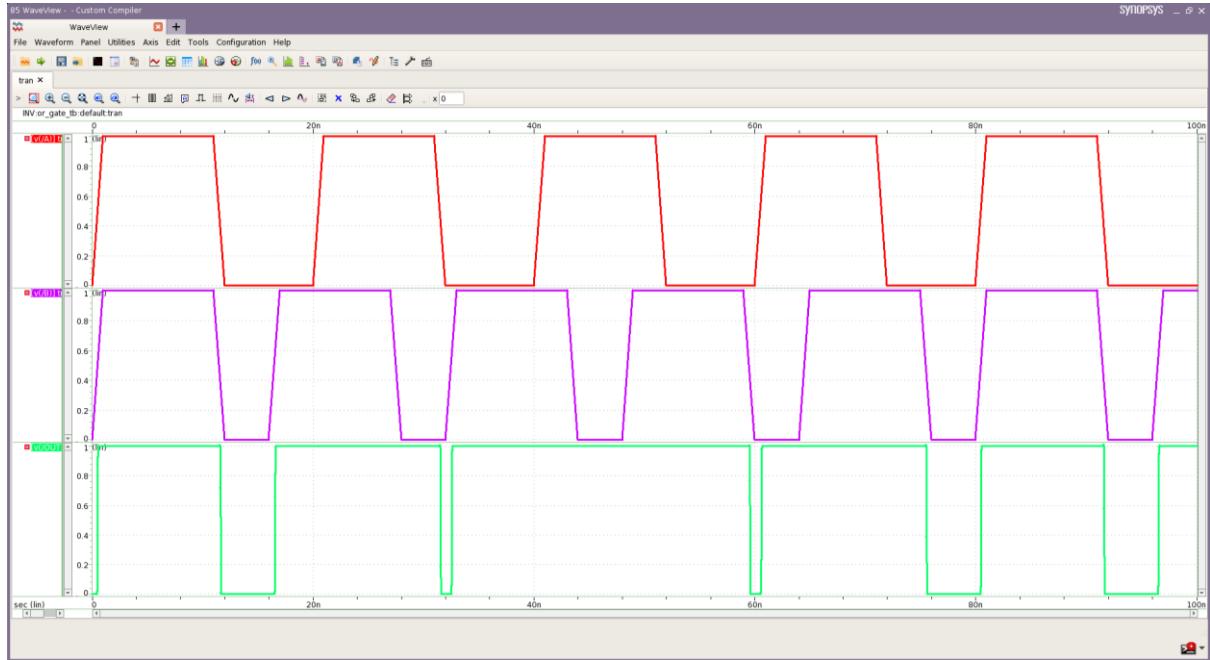
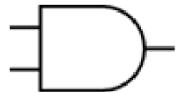


Fig. 4.45. Transient Analysis of OR gate

4.3.3. AND gate

- The operation of an AND gate is based on the logical AND function. If any of the input signals (A or B) is low (0), the output (Y) will be low (0).
- Only when both inputs are high (1), the output will be high (1). Mathematically, the output (Y) of an AND gate with two inputs (A and B) is given by the Boolean expression: $Y = A \cdot B$.
- The AND gate implements logical conjunction, where the output is true only if all inputs are true (1).



INPUT		OUTPUT
A	B	
0	0	0
1	0	0
0	1	0
1	1	1

Fig. 4.46. Symbol AND gate [11]

TABLE 3: Truth table of AND gate [11]

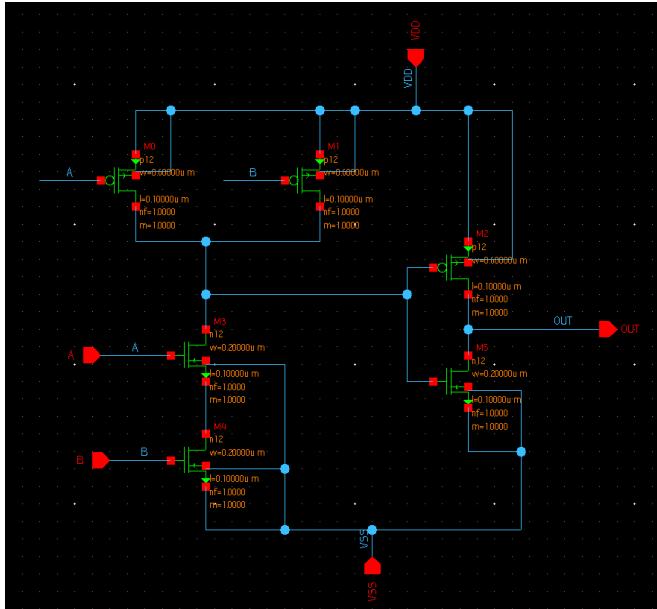


Fig. 4.47. Schematic of AND gate using CMOS

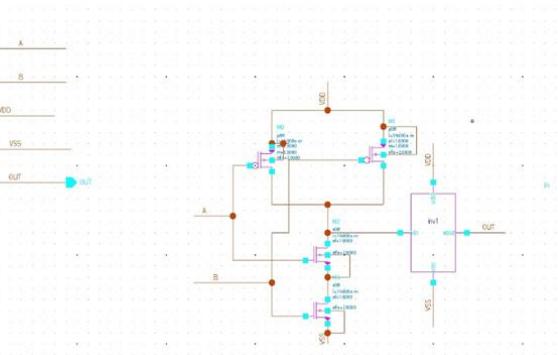


Fig. 4.48. Schematic of AND gate using finFET

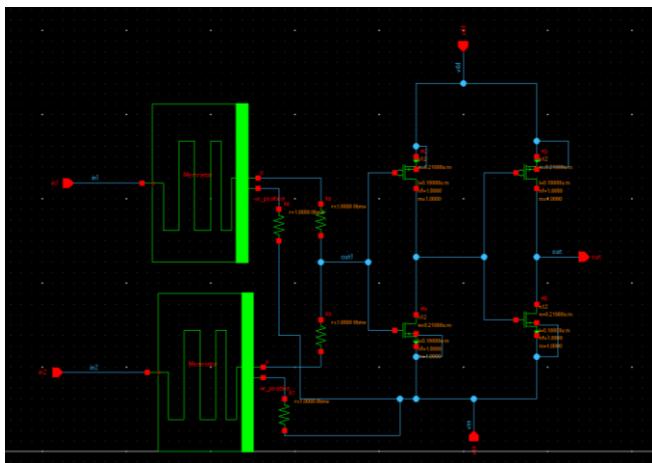


Fig. 4.49. Schematic of AND gate-Memristor

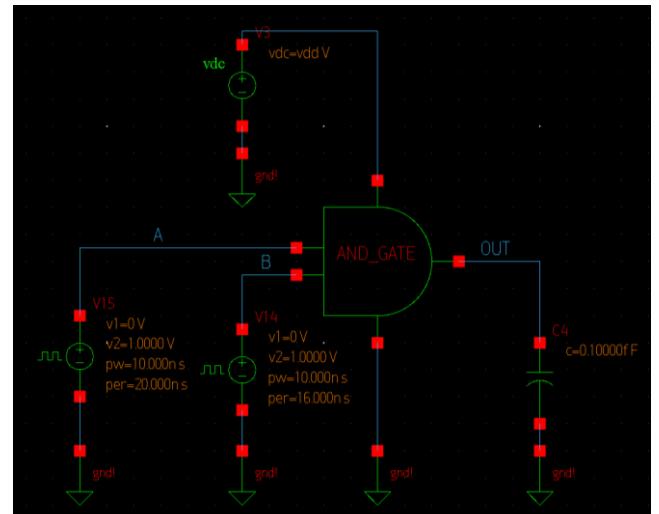


Fig. 4.50. Testbench of AND gate

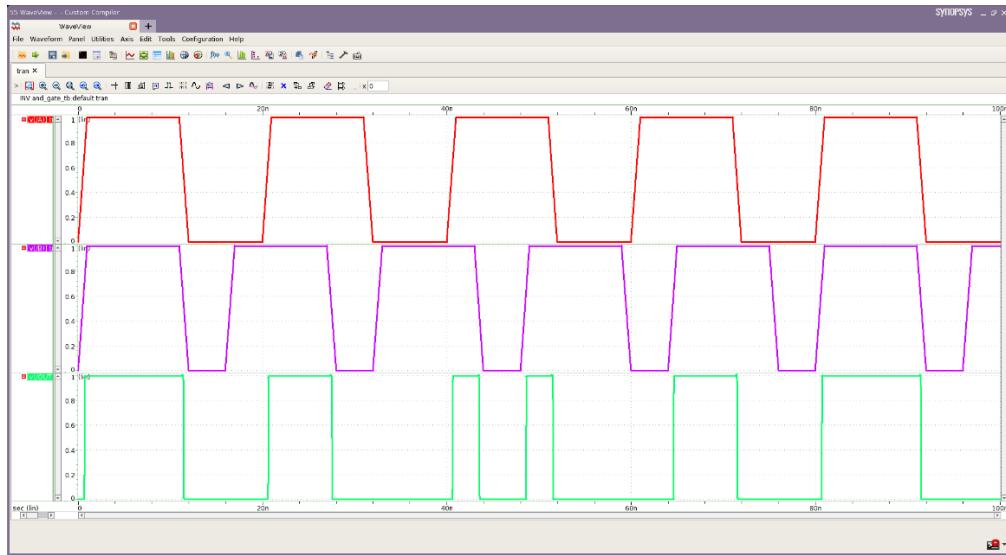


Fig. 4.51. Transient Analysis of AND gate

4.3.4. NOR gate

- The NOR gate is the complement of the OR gate. It produces a high output only when all its input signals are low; otherwise, it produces a low output.



Fig. 4.52. Symbol NOR gate [11]

INPUT		OUTPUT
A	B	
0	0	1
1	0	0
0	1	0
1	1	0

TABLE 4: Truth table of NOR gate [11]

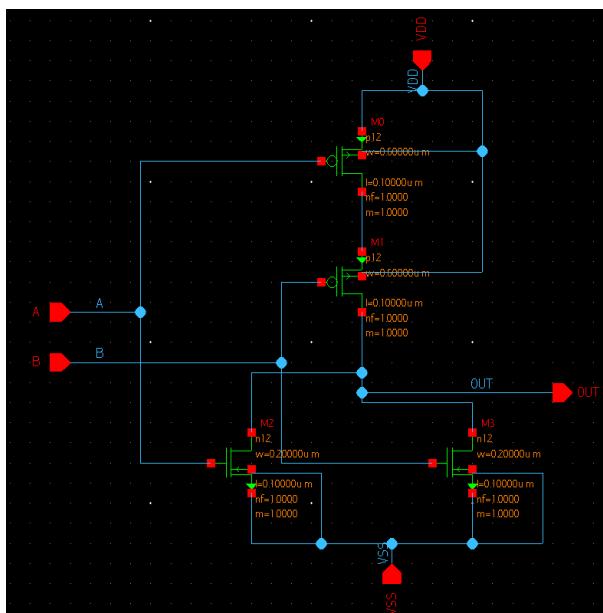


Fig. 4.53. Schematic of NOR gate using CMOS

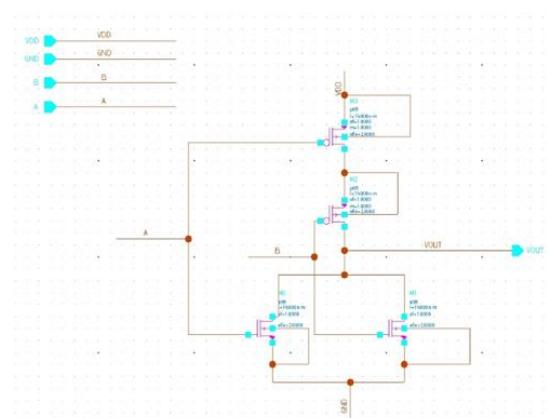


Fig. 4.54. Schematic of NOR gate using FinFET

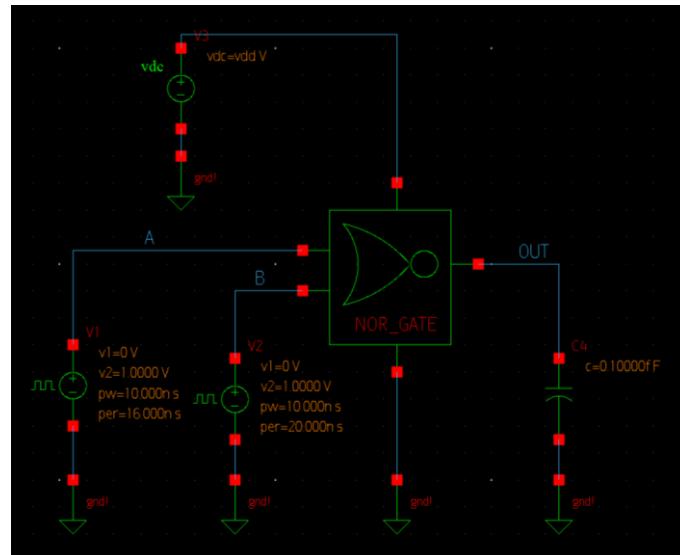


Fig. 4.55. Testbench of NOR gate

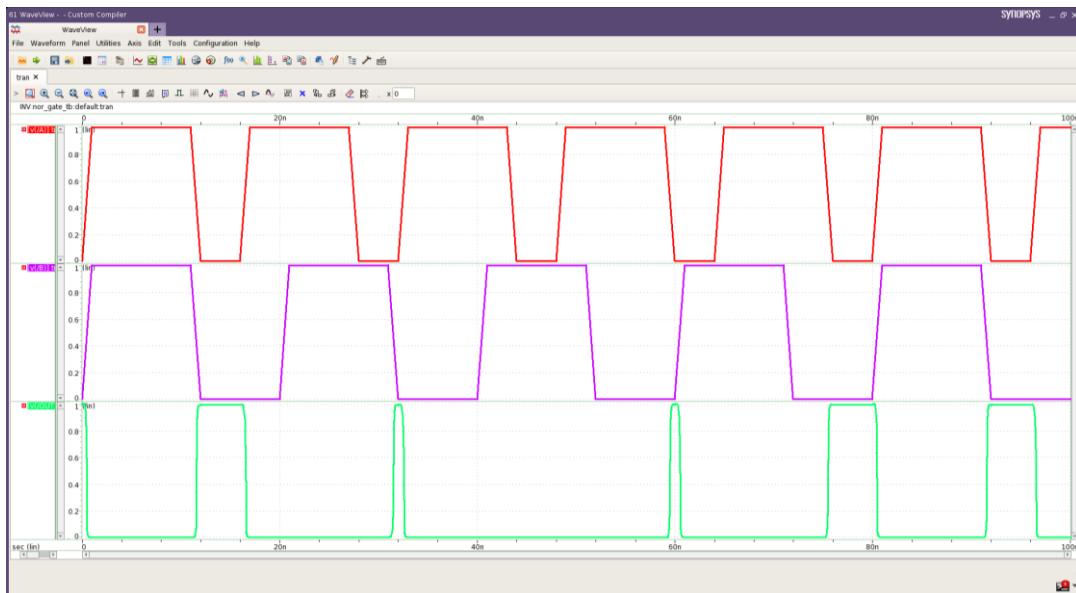


Fig. 4.56. Transient Analysis of NOR gate

4.3.5. NAND gate

The NAND gate is the complement of the AND gate. It produces a low output only when all its input signals are high; otherwise, it produces a high output.



Fig. 4.57. Symbol NAND gate [11]

INPUT		OUTPUT
A	B	
0	0	1
1	0	1
0	1	1
1	1	0

TABLE 5: Truth table of NAND gate [11]

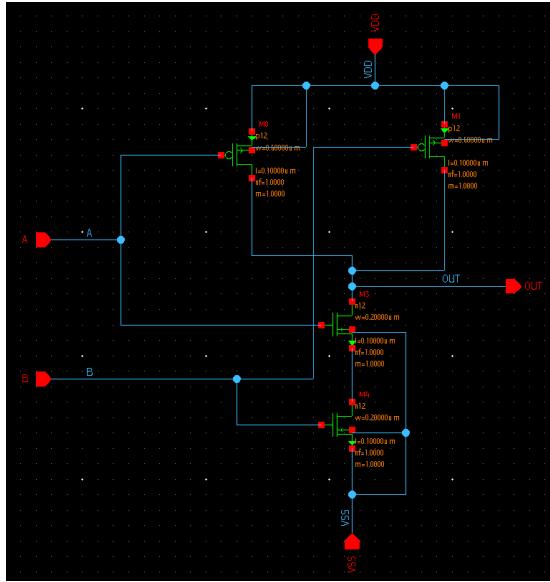


Fig. 4.58. Schematic of NAND gate using CMOS

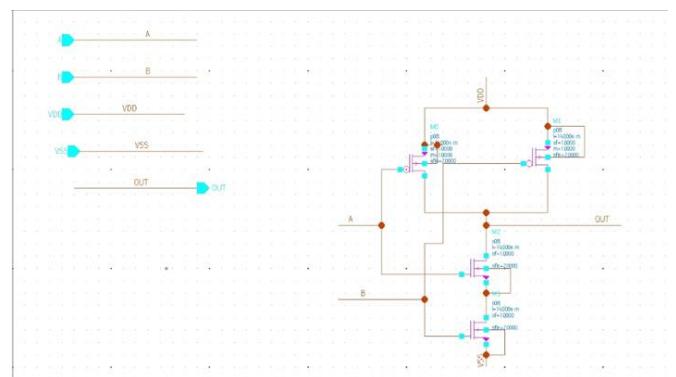


Fig. 4.59. Schematic of NAND gate using finFET

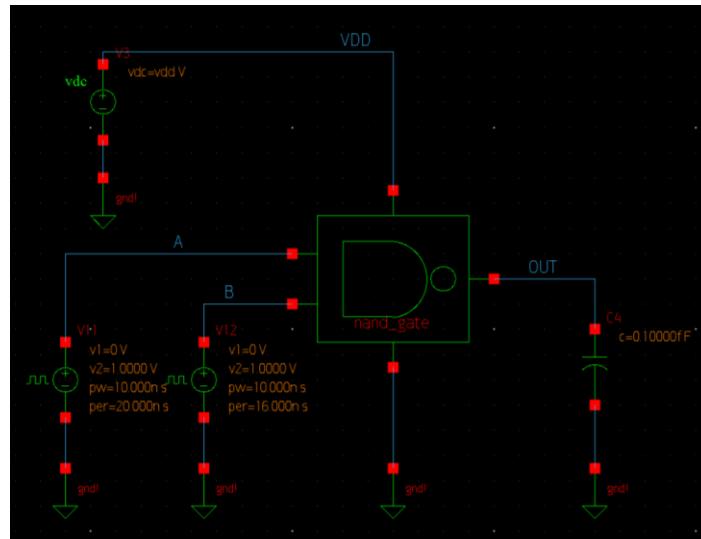


Fig. 4.60. Testbench of NAND gate

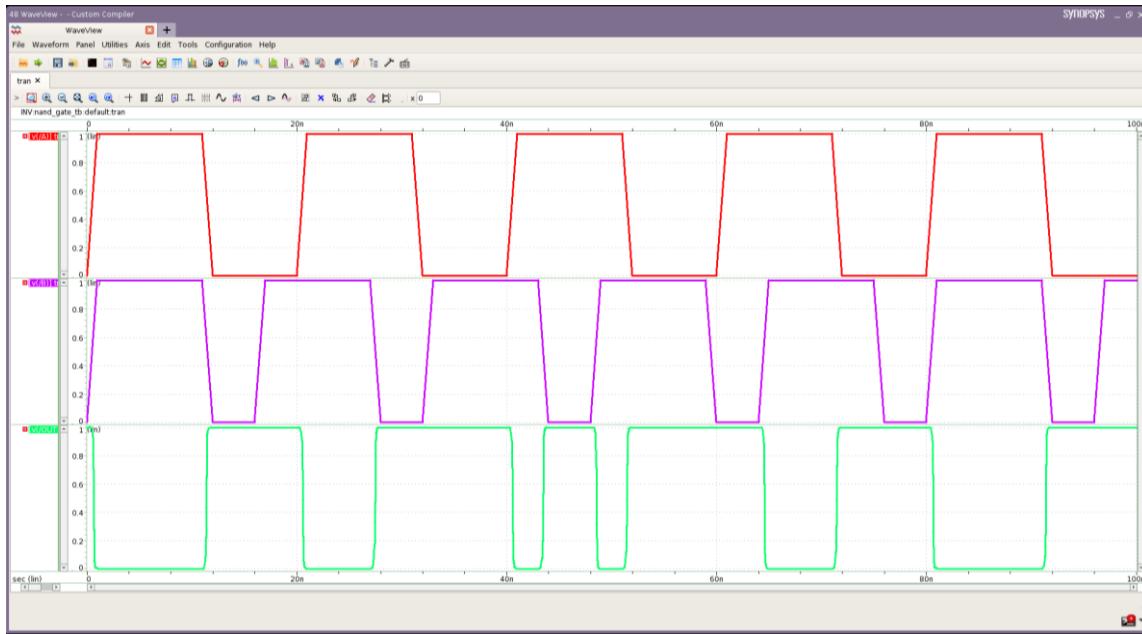


Fig. 4.61. Transient Analysis of NAND gate

4.3.6. Half Adder

A half adder has two single bit inputs A, B, and outputs Sum(S) and Carry(C)[12]. The schematic and the truth table are shown below.

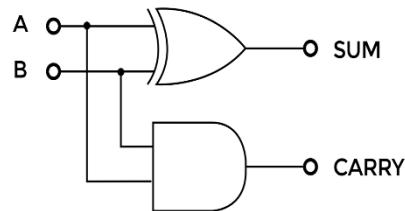


Fig. 4.62. Half-Adder [12]

Inputs		Outputs	
A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

TABLE 6: Truth table of Half-Adder [12]

INPUT		OUTPUT
A	B	
0	0	0
1	0	1
0	1	1
1	1	0

TABLE 7: Truth table of XOR gate [11]

From the above truth table, the Boolean expressions of the Sum and Carry are given as follows:

$$S = A'B + AB' \text{ or } A \oplus B$$

$$C = A \cdot B$$

Exclusive OR (XOR) is not a fundamental gate, it can be designed using several other gates. There are multiple ways to implement the XOR gate using the CMOS logic [13]. Out of them, the best to implement the half adder is shown below:

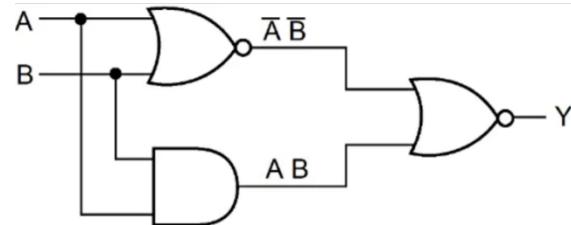


Fig. 4.63. XOR gate schematic [13]

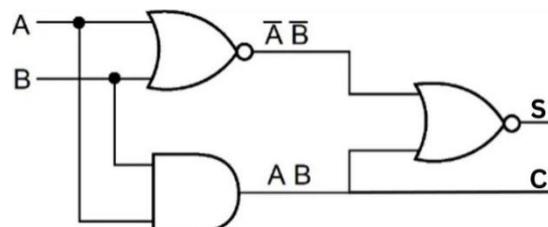


Fig. 4.64. Schematic of Half-Adder

In the above figure, Y is the XOR output of A and B. Therefore, in a half-adder, Y can be used in sum generation and an additional output pin can be connected to AB which is already available in XOR for the carry generation. The resultant half adder schematic is shown below:

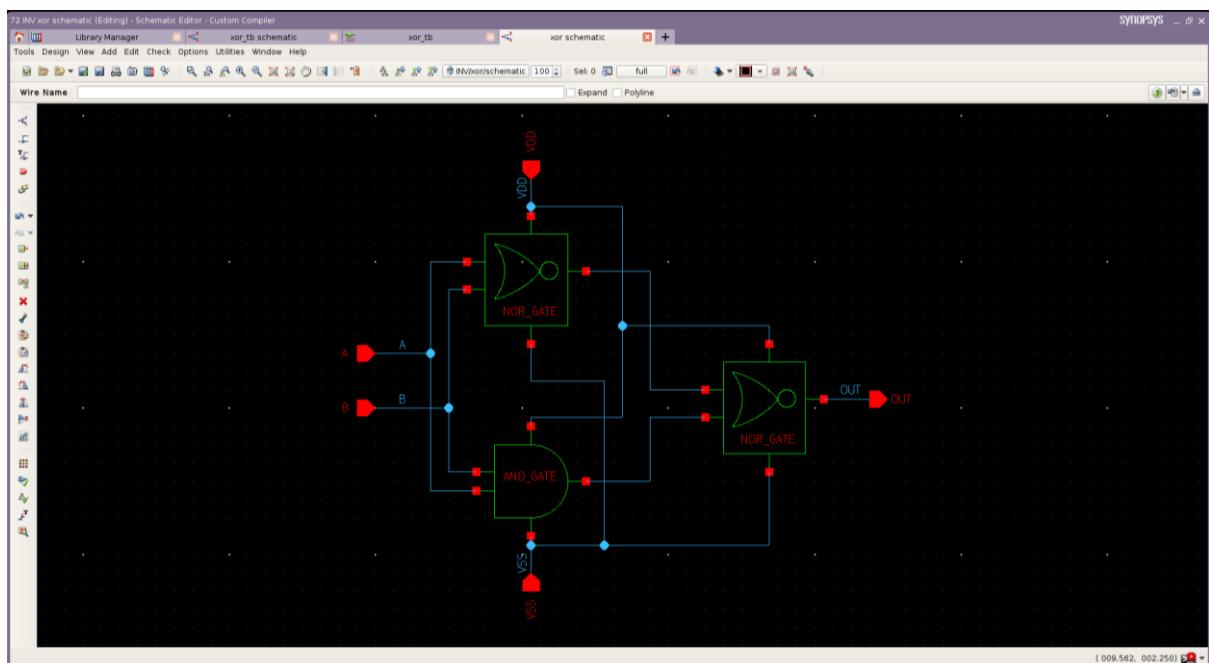


Fig. 4.65. Schematic of XOR gate

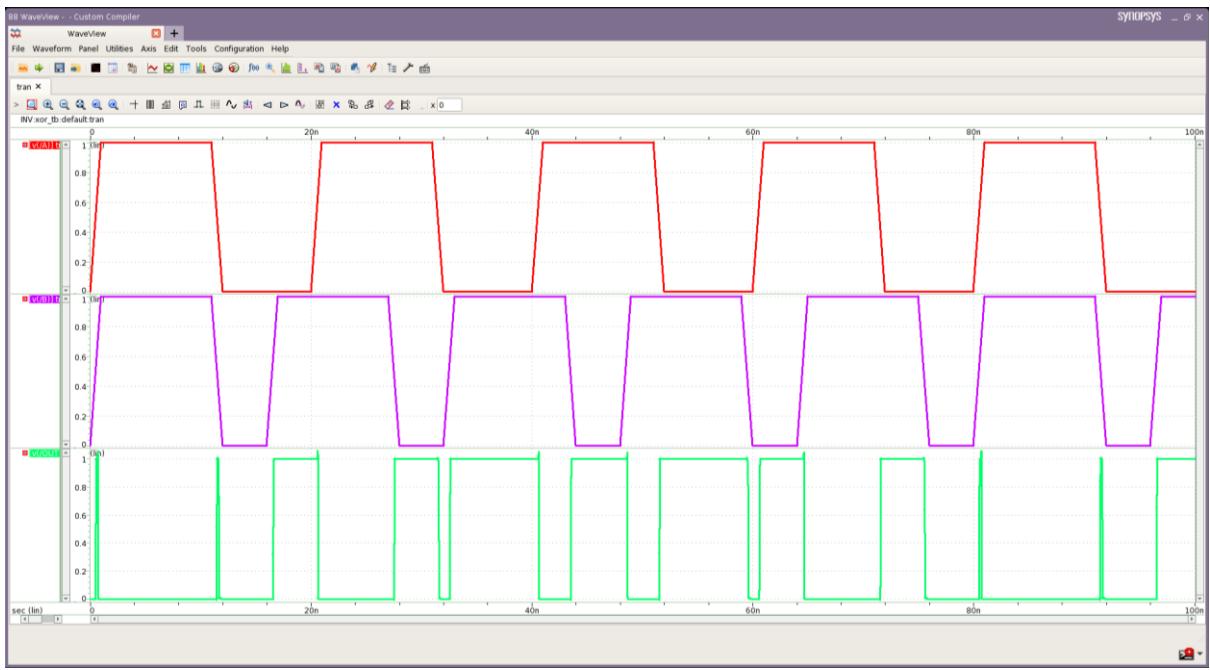


Fig. 4.66. Transient analysis of XOR gate

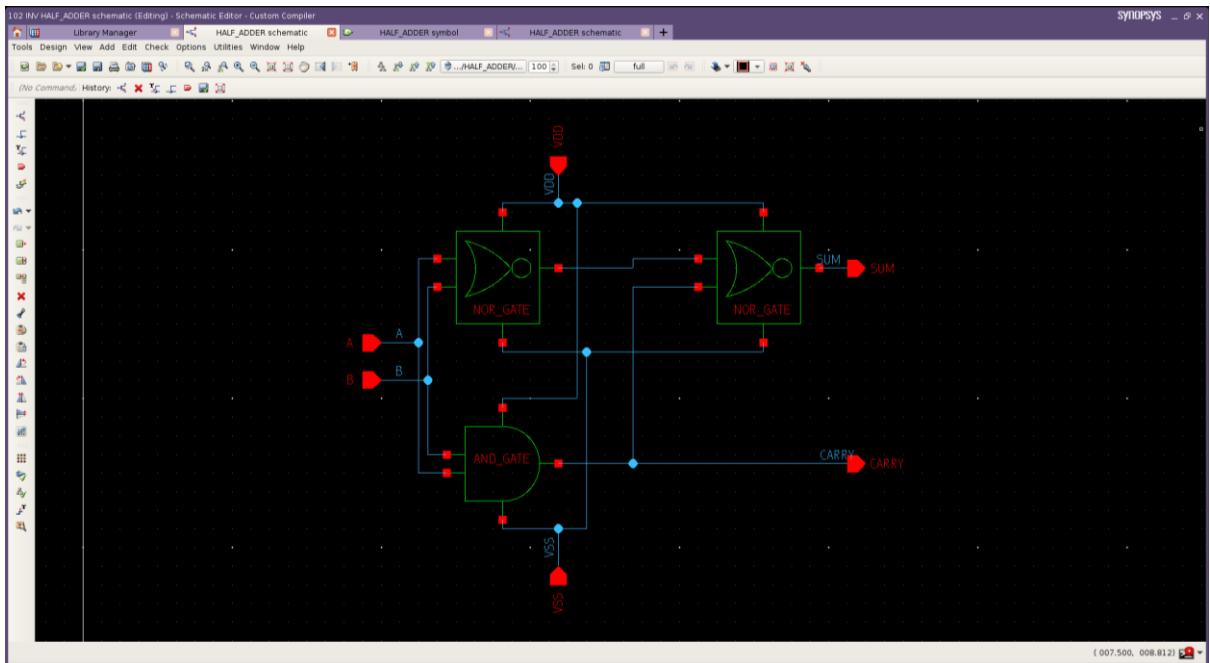


Fig. 4.67. Schematic of Half Adder

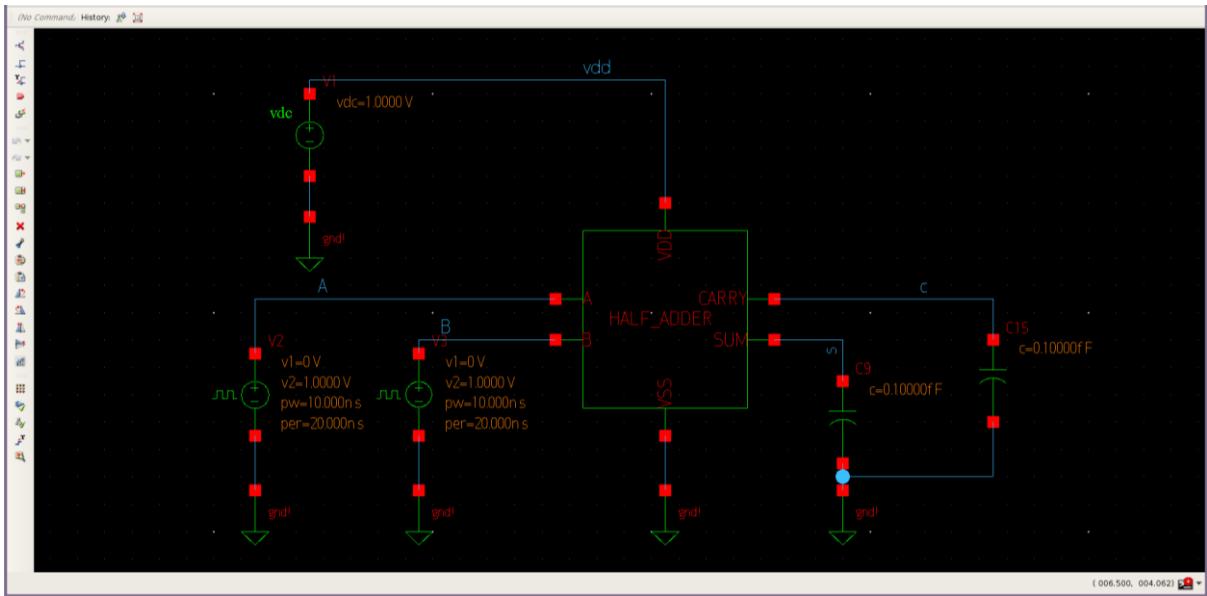


Fig. 4.68. Testbench of Half Adder

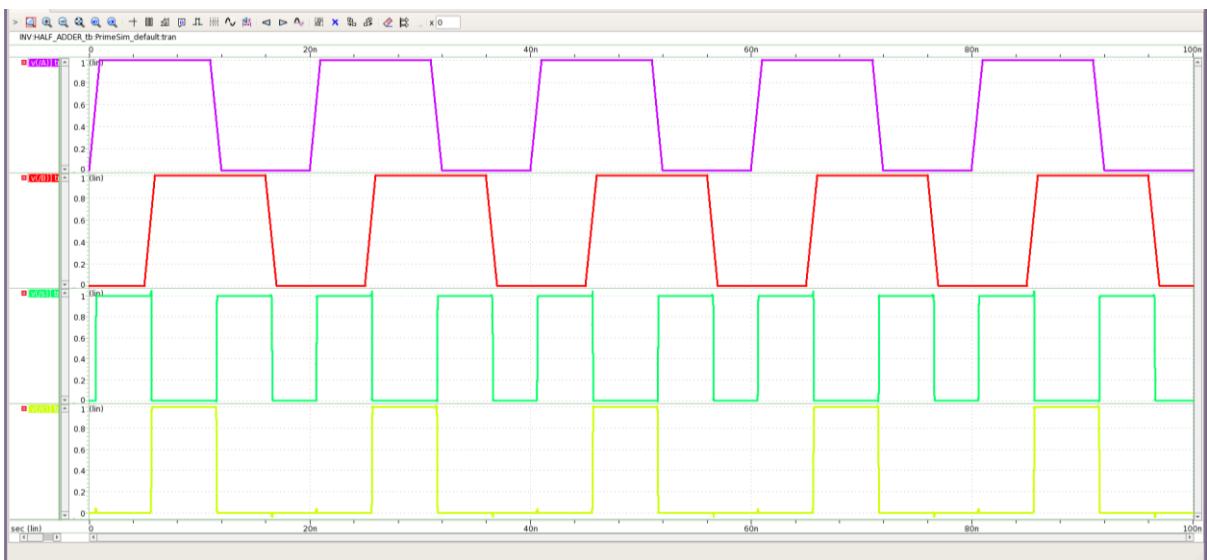


Fig. 4.69. Transient Analysis of Half Adder

4.3.7. Full Adder

Full-Adder is a combinational circuit that performs the addition operation of three single bit inputs, A, B and a Carry in (Cin). There are 8 possible input combinations for each case of sum (S) and carry out (Cout) [13]. Truth table and corresponding Boolean equations are given below along with the design of Full-Adder using the Half adders.

$$Sum(S) = A \oplus B \oplus C_{in}$$

$$Carry\ out\ (C_{out}) = A.B + B.C_{in} + C_{in}.A$$

Inputs			Outputs	
A	B	C_{in}	S	C_{out}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

TABLE 8: Truth table of Full-Adder [13]

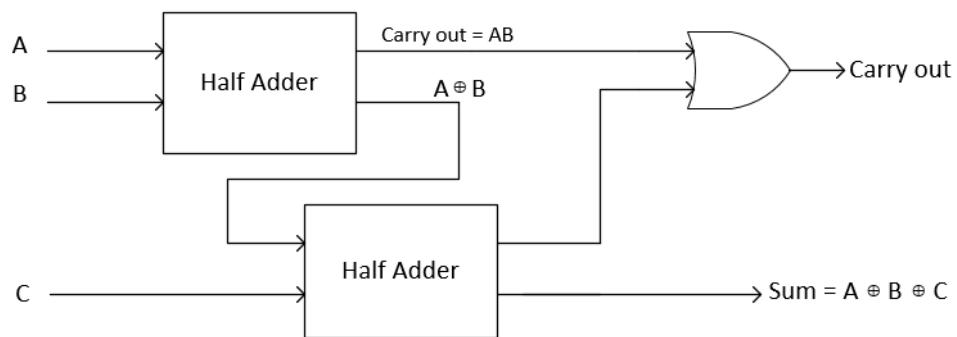


Fig. 4.70. Full-Adder using Half-Adders [13]

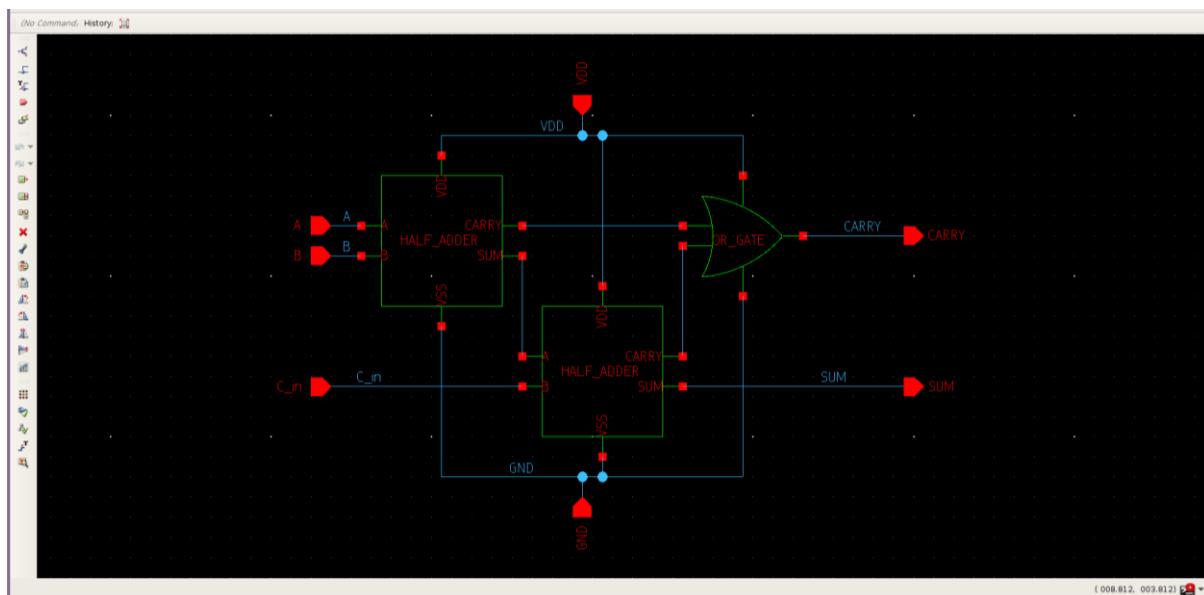


Fig. 4.71. Schematic of Full Adder

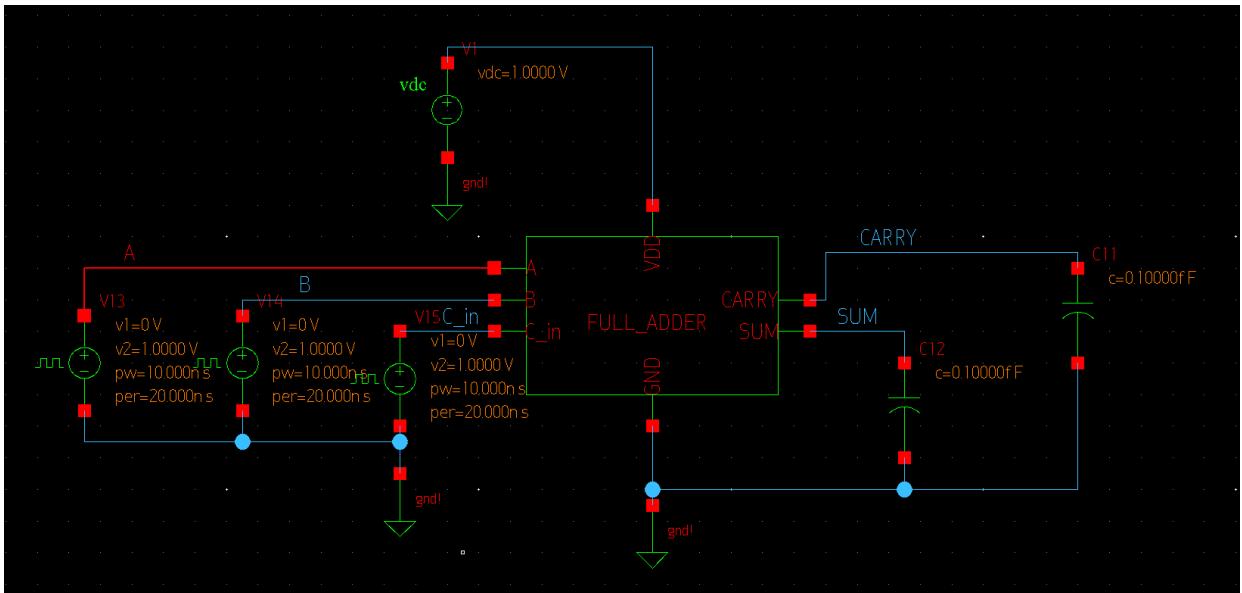


Fig. 4.72. Testbench of Full Adder

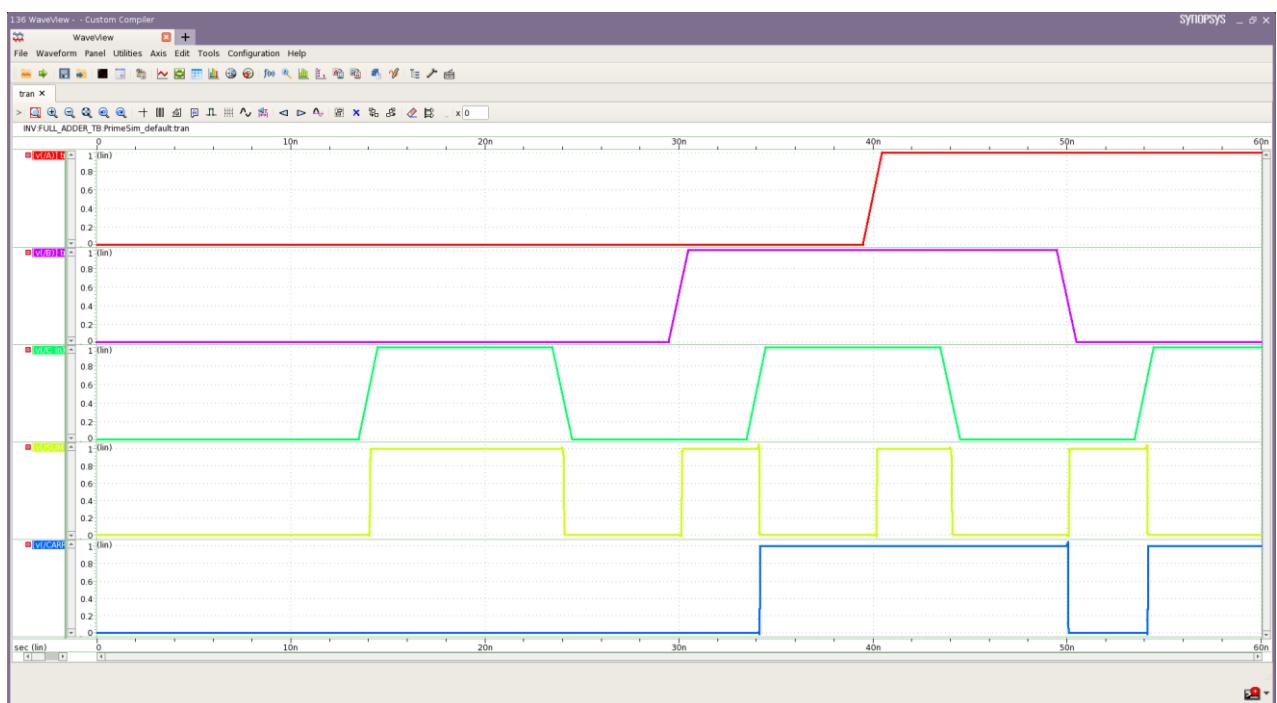


Fig. 4.73. Transient Analysis of Full Adder

4.4. Design of Binary Multiplier Circuits

4.4.1. Design of 4*4 Partial Product Generator

Importance of Partial Product Generation:

- (i) **Foundation of Binary Multiplication:** Partial product generation is the foundation for binary multiplication in hardware and software implementations. It's a systematic way to break down the multiplication process into simpler steps.
- (ii) **Efficiency in Multiplier Design:** In hardware implementations like digital circuits, generating partial products efficiently can significantly impact the speed and resource utilization of the multiplier.
- (iii) **Parallel Processing:** Partial products can be generated in parallel, allowing for faster multiplication of large numbers. This is particularly important in applications requiring high-speed computations.
- (iv) **Scalability:** The concept of partial product generation scales well to larger bit lengths, making it suitable for handling multiplication operations involving multi-bit numbers.
- (v) **Optimization Opportunities:** By optimizing the generation and accumulation of partial products, overall multiplication performance can be improved, especially in high-performance computing environments.

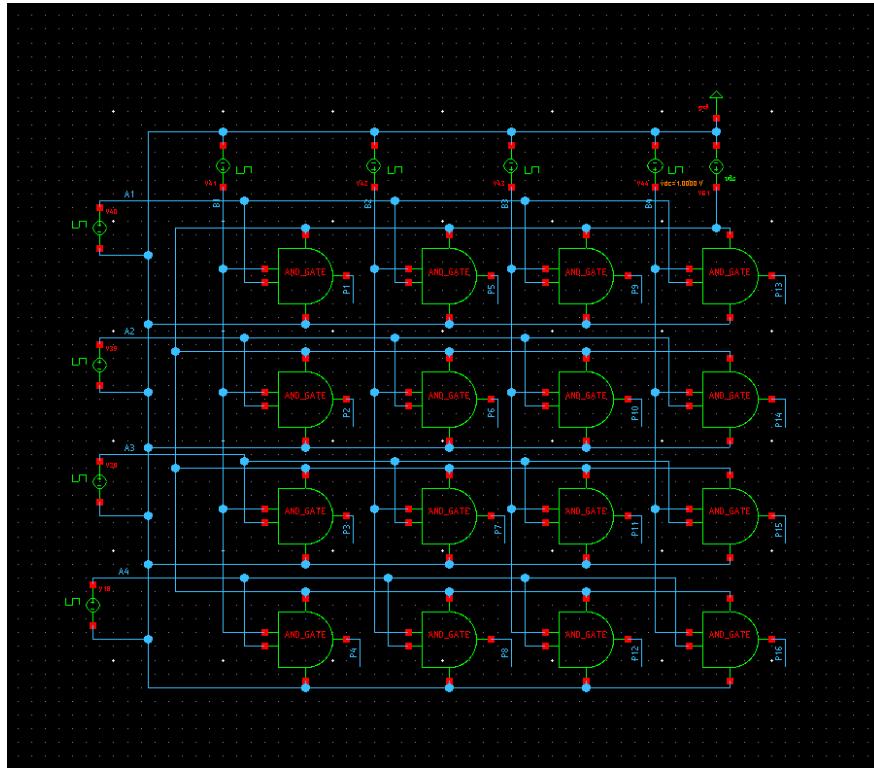


Fig. 4.74. Partial product generator for 4 * 4 multiplier

4.4.2. Wallace Multiplier

A Wallace multiplier is a hardware implementation of a binary multiplier, a digital circuit that multiplies two integers.

For the conventional Wallace reduction method, once the partial product array (of N^2 bits) is formed, adjacent rows are collected into nonoverlapping groups of three.

Each group of three rows is reduced by

1) applying a full adder to each column that contains three bits.

2) applying a half adder to each column that contains two bits.

3) passing any single bit columns to the next stage without processing. This reduction method is applied to each successive stage until only two rows remain. The final two rows are summed with a carry propagating adder.

The Wallace tree Multiplier utilizes a lesser number of half adders and full adders and operates faster when compared to the conventional Array Multiplier.

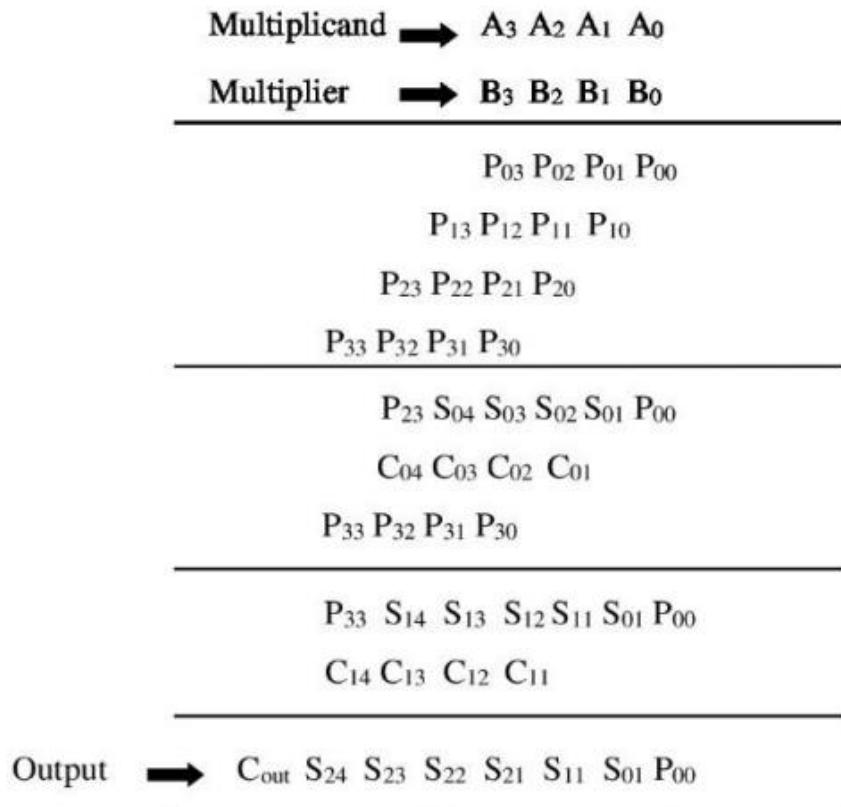


Fig. 4.75. Operation of 4*4 Wallace Multiplier

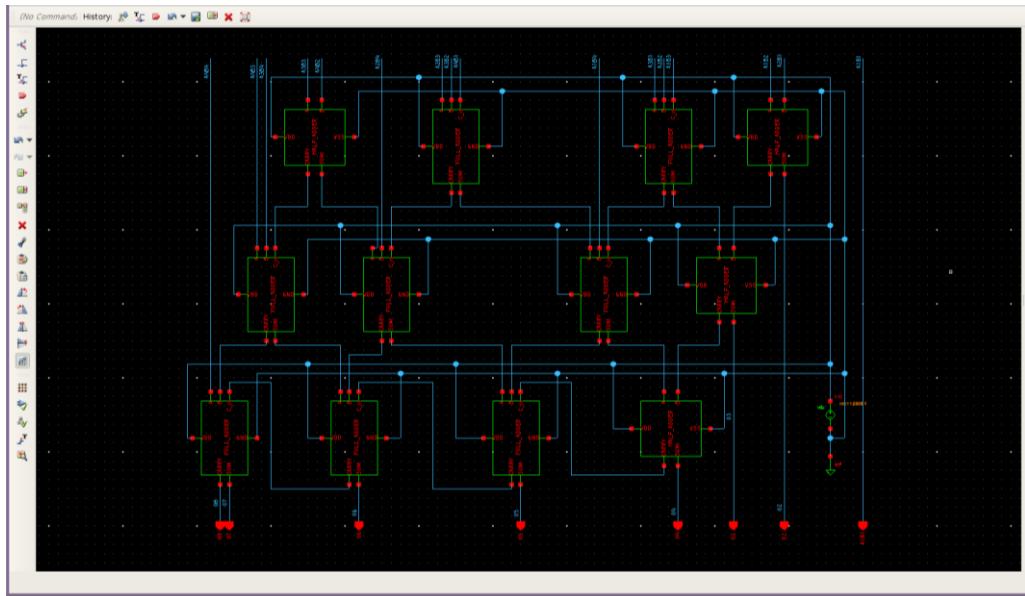


Fig. 4.76. Schematic of 4×4 Wallace Multiplier

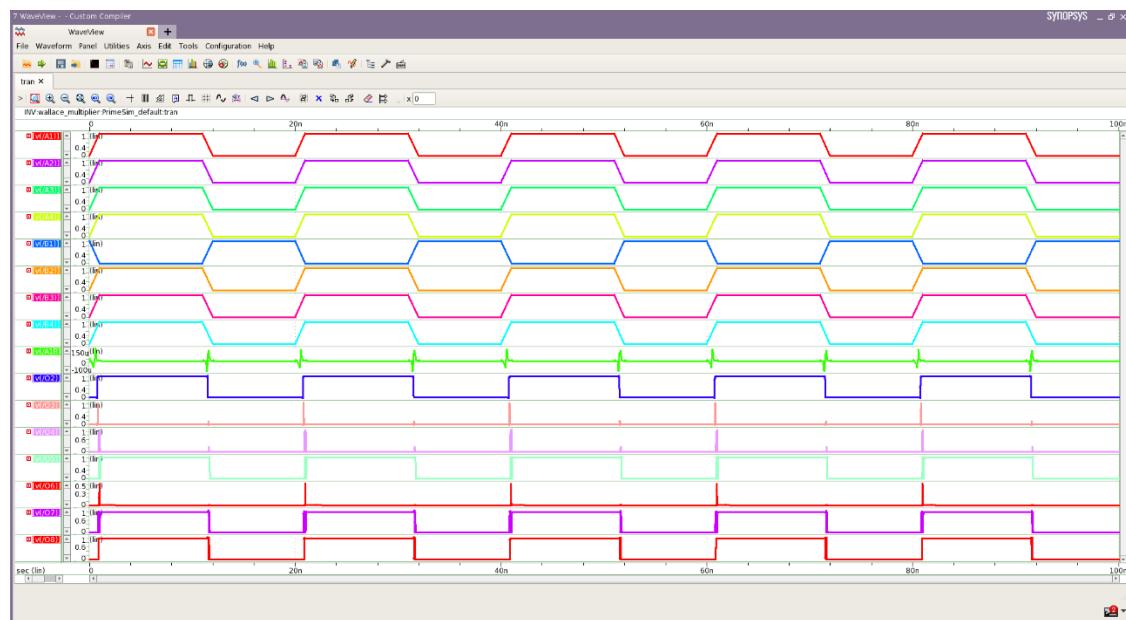


Fig. 4.77. Transient Analysis of 4×4 Wallace Multiplier

Array multiplier has more power consumption and propagation delay compared to other multipliers. Wallace multiplier has large area wastage problem due to irregularity in the structure. To overcome these disadvantages in the existing multipliers Dadda multiplier is proposed. Different reduction tree reduces the number of gates required and makes it slightly faster. Reduce the adding steps required to add some products. It is obtained by reducing the number of rows in the matrix by a half adder in each addition step.

4.4.3. Dadda Multiplier

For an $N \times M$ multiplier, we choose the d which is less than the smaller number of N and M . $d+1$ is the maximum number of partial products in a column in each stage.

Each preceding stage height can be no larger than $(3^* \text{ successor height}/2)$ successor.

There is a special sequence proposed by Dadda for $d+1$, which must be maintained in each successive stage of reduction of partial products. It is given as follows [16][15][17]:

2, 3, 4, 6, 9, 13, 19, 28, etc

Steps involved in stage reduction [17]:

- (i) If the position has bits less than the d , move on.
- (ii) If, the position has bits equal to $d+1$, use HA to compress two bits, pushing one result to the next position. The current position would be reduced to d , allowing us to move to the next bit position.
- (iii) If the position has any higher number of bits, use FA to compress, pushing bits and repeat again from the first step.

Stage 1: Partial Product Generation

P33	P23	P13	P03	P02	P01	P00
	P32	P22	P12	P11	P10	
		P31	P21	P20		
			P30			

Here, $d+1=4$.

As per the algorithm, for the next stage $d+1$ should be reduced to 3.

Stage 2:

P33	P23	S-2	S-1	P02	P01	P00
	P32	P31	P21	P11	P10	
	C-2	C-1	P30	P20		

Here, $d+1=3$.

As per the algorithm, for the next stage $d+1$ should be reduced to 2.

Stage 3: Ripple Carry Adder

P33	S-6	S-5	S-4	S-3	P01	P00
C-6	C-5	C-4	C-3	P20	P10	

Half Adder - 

Full Adder - 

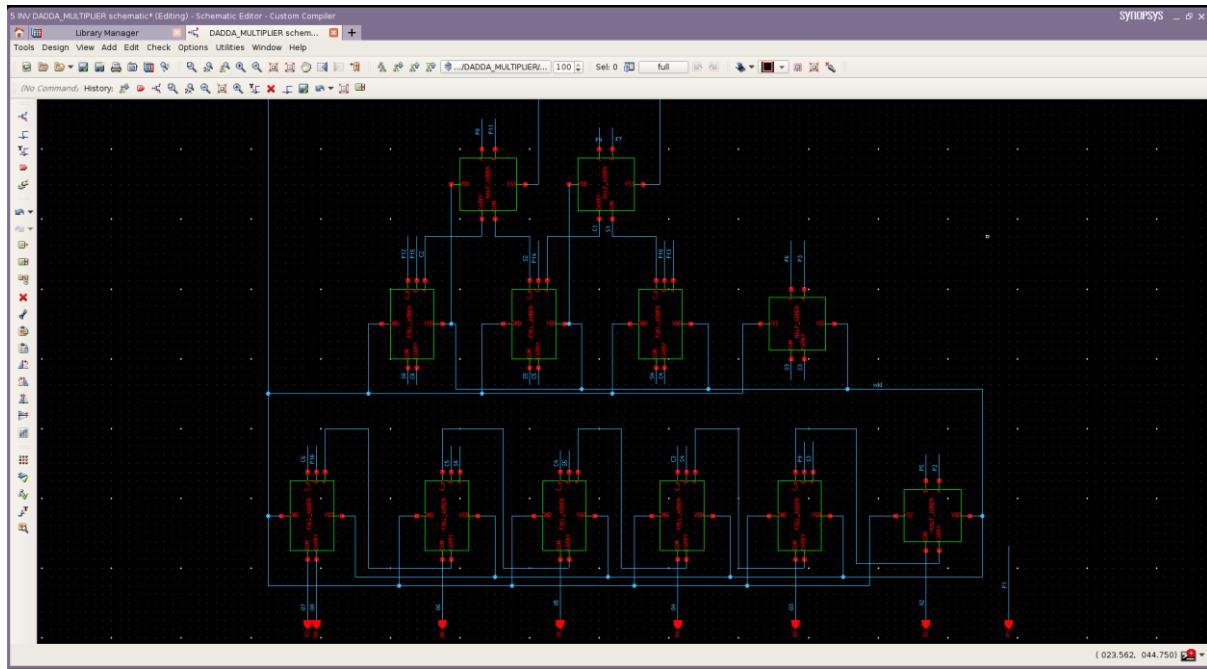


Fig.4.78. Schematic of 4*4 Dadda Multiplier

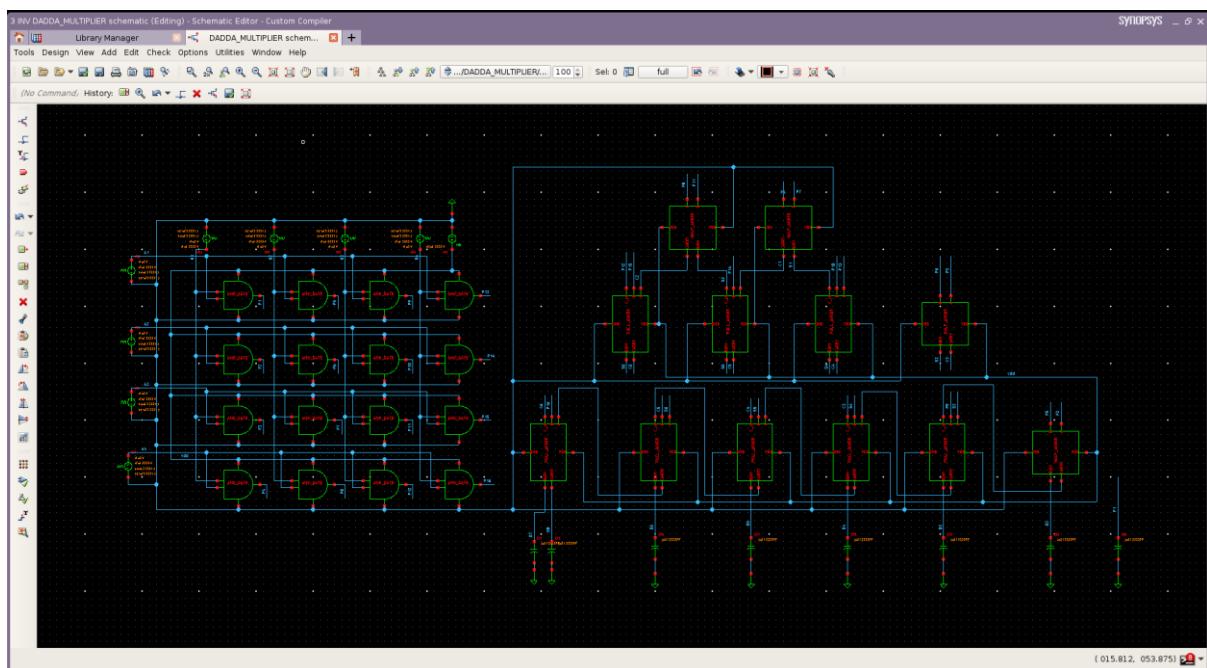


Fig.4.79. Schematic of 4*4 Dadda Multiplier with Partial Product Generator

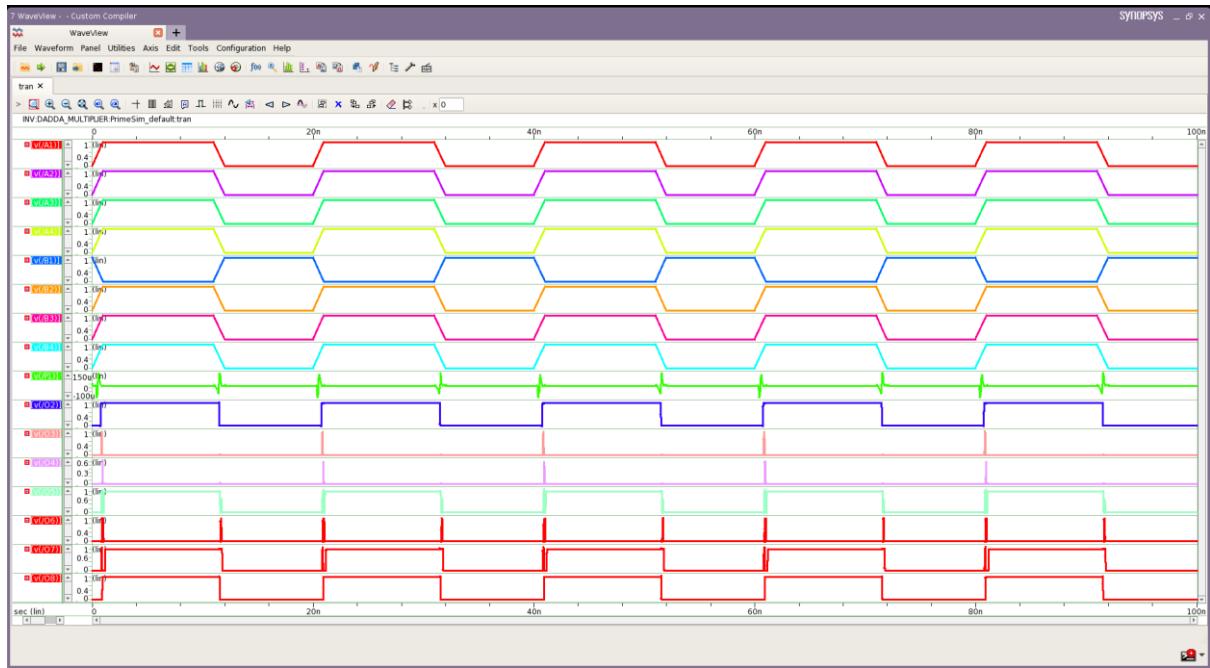


Fig.4.80. Transient Analysis of 4*4 Dadda Multiplier

CONCLUSION

When compared to traditional technologies like FinFET and CMOS, the integration of memristors with CMOS logic circuits offers interesting paths for creating low-power digital circuitry. We have investigated the advantages and disadvantages of this integration through our project. Our research concludes that there are a number of benefits to using memristors in CMOS logic circuits. First of all, memristors have the ability to store information without losing it due to their non-volatile memory properties. This lowers the power used on standby. Furthermore, memristors' intrinsic qualities such as their low power consumption and high integration density make them appealing options for implementing energy-efficient digital circuits. Additionally, our comparison study shows that memristor-based CMOS circuits perform better than both in terms of power efficiency.

REFERENCES

- [1] N. A. Badiger, S. Iyer and S. Gejji, "Power and Delay Analysis of a CMOS Inverter," 2023 International Conference on Data Science and Network Security (ICDSNS), Tiptur, India, 2023, pp. 1-6, doi: 10.1109/ICDSNS58469.2023.10244825.
- [2] P. U. Jain and V. K. Tomar, "FinFET Technology : As A Promising Alternatives for Conventional MOSFET Technology," 2020 International Conference on Emerging Smart Computing and Informatics (ESCI), Pune, India, 2020, pp. 43-47, doi: 10.1109/ESCI48226.2020.9167646.
- [3] A. Islam, "Technology scaling and its side effects," 2015 19th International Symposium on VLSI Design and Test, Ahmedabad, India, 2015, pp. 1-1, doi: 10.1109/ISVDAT.2015.7208164.
- [4] A. K. Kuna, K. Kandpal and K. B. R. Teja, "An investigation of FinFET based digital circuits for low power applications," 2017 International Conference on Circuit ,Power and Computing Technologies (ICCPCT), Kollam, India, 2017, pp. 1-6, doi: 10.1109/ICCPCT.2017.8074280.
- [5] H. Farkhani, A. Peiravi, J. M. Kargaard and F. Moradi, "Comparative study of FinFETs versus 22nm bulk CMOS technologies: SRAM design perspective," 2014 27th IEEE International System-on-Chip Conference (SOCC), Las Vegas, NV, USA, 2014, pp. 449-454, doi: 10.1109/SOCC.2014.6948971.
- [6] A. N. Bhoj and N. K. Jha, "Design of Logic Gates and Flip-Flops in High-Performance FinFET Technology," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 21, no. 11, pp. 1975-1988, Nov. 2013, doi: 10.1109/TVLSI.2012.2227850.
- [7] A. Verma, V. Singh, A. K. Upadhyay, A. Upadhyay and S. A. Taya, "The Power Dissipation of Complementary Metal Oxide Semiconductor (CMOS) Inverter and Propagation Delay for Various Technologies," 2023 International Conference on Sustainable Emerging Innovations in Engineering and Technology (ICSEIET), Ghaziabad, India, 2023, pp. 25-30, doi: 10.1109/ICSEIET58677.2023.10303532.
- [8] Esa, Mohd & Achyut, Konasagar. (2019). Design and Verification of 4 X 4 Wallace Tree Multiplier. 10.5281/zenodo.3757325.
- [9] Gowri, Mangala & H., Girish. (2023). DESIGN OF 3-BIT CMOS WALLACE MULTIPLIER. Seybold Report. 18. 1260-1271.
- [10] Singh, Shikha & Shukla, Yagnesh. (2023). Implementation of FinFET technology based low power 4×4 Wallace tree multiplier using hybrid full adder. TELKOMNIKA (Telecommunication Computing Electronics and Control). 21. 1139. 10.12928/telkomnika.v21i5.24304.
- [11] S. T. Prasad and S. K. Sinha, "Power Efficient and High-performance 4-bit Dadda Multiplier using Multiplexer based BEC-1 converter," 2021 International Conference on Recent Trends on

Electronics, Information, Communication & Technology (RTEICT), Bangalore, India, 2021, pp. 749-754, doi: 10.1109/RTEICT52294.2021.9573750.

- [12] Mathews, Manoj. (2022). Mathematical Modelling of TEAM and VTEAM Memristor Model Using Verilog-A. 10.48550/arXiv.2205.12472.
- [13] Yang, J., Strukov, D. & Stewart, D. Memristive devices for computing. *Nature Nanotech* 8, 13–24 (2013). <https://doi.org/10.1038/nano.2012.240>.
- [14] Emara, A.S., Madian, A.H., Amer, H.H., Amer, S.H. and Abdelhalim, M.B. (2016) On the Production Testing of Memristor Ratioed Logic (MRL) Gates. *Circuits and Systems*, 7, 3016-3025.
- [15] Singh, Tejinder. (2015). Hybrid Memristor-CMOS (MeMOS) based Logic Gates and Adder Circuits. arXiv [cs.ET]. 1506. 1-11.
- [16] Agarwal, Tarun. “What Is a Memristor? Types of Memristors, Construction and Applications.” *ElProCus - Electronic Projects for Engineering Students*, 4 Mar. 2019, www.elprocus.com/what-is-a-memristor-types-of-memristors-and-their-applications.
- [17] “The Memristor.” *American Scientist*, 3 Feb. 2018, www.americanscientist.org/article/the-memristor.
- [18] Singh, C.P., Raghvendra & Pandey, S.K. An efficient and flexible window function for a memristor model and its analog circuit application. *J Comput Electron* 21, 1425–1433 (2022). <https://doi.org/10.1007/s10825-022-01939-0>.

CO-PO Mapping

Course Outcomes for Project

At the end of the project work the students will be able to

1. Identify a contemporary engineering application to serve the society at large
2. Apply complex engineering concepts and use computational tools to get the desired solution
3. Analyze the assembled/fabricated/developed products intended.
4. Prepare documents and present the project report articulating the applications of the concepts and ideas coherently
5. Demonstrate ethical and professional attributes during the project implementation.
6. Evaluate & execute the project in a collaborative environment.

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	2				3	2							
CO2	3	3			3									
CO3	3	3	3	2							2			
CO4									3			2		
CO5							3							
CO6									3					