

A Project Report on

DESIGN OF MEMRISTOR BASED LOW BASED ENCODER USING PGL TECHNIQUE

Submitted in partial fulfillment of the requirements for the award of the Degree of

Bachelor of Technology

in

Electronics and Communication Engineering

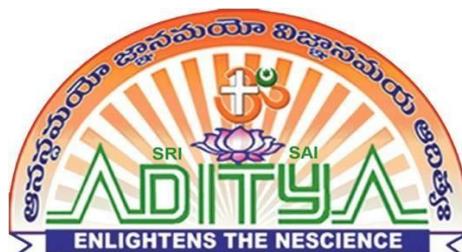
By

M.GANESH	20A91A04F4
M.KOMAL	21A95A0420
K. SAIRAM	20A91A04F1
S. RAZAK	20A91A04H5

Under the Esteemed Supervision of

Mr. P. BUJJIBABU

Professor



Department of Electronics and Communication Engineering

ADITYA ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Affiliated to JNTUK, Kakinada, Accredited by NBA (Tier I) and NAAC with 'A++' Grade)

Aditya Nagar, ADB Road, Surampalem

2020 – 2024

ADITYA ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Affiliated to JNTUK, Kakinada, Accredited by NBA (Tier I) and NAAC with 'A++' Grade)

Aditya Nagar, ADB Road, Surampalem

2020 – 2024

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING



CERTIFICATE

This is to certify that the thesis entitled "**DESIGN OF MEMRISTOR BASED ENCODER USING PGL TECHNIQUE**" is being submitted by

M.GANESH	20A91A04F4
M.KOMAL	21A95A0420
K. SAIRAM	20A91A04F1
S. RAZAK	20A91A04H5

in partial fulfillment of the requirements for the award of degree of **B. Tech** in Electronics and Communication Engineering from **Jawaharlal Nehru Technological University, Kakinada** is a record of bonafide work carried out by them at Aditya Engineering College(A).

The results embodied in this Project report have not been submitted to any other University or Institute for the award of any degree or diploma.

PROJECT GUIDE

(Mr. P. BUJJIBABU)

HEAD OF THE DEPARTMENT

(Dr. N. Radha)

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

We take great pleasure to express our deep sense of gratitude to our project guide **Mr. P. BUJJIBABU SIR**, Associate Professor, for his valuable guidance during the course of our project work.

We would like to thank **Dr. N Radha**, Head of the Department of Electronics and Communication Engineering for her encouragement.

We would like to express our heart-felt thanks to **Dr. M Sreenivasa Reddy**, Principal, Aditya Engineering College, Surampalem for providing all the facilities for our project.

Our utmost thanks to all the faculty members and Non-Teaching Staff of the Department of Electronics & Communication Engineering for their support throughout our project work.

Our Family Members and Friends receive our deepest gratitude and love for their support throughout our academic year.

M. GANESH
M. KOMAL
K. SAIRAM
S. RAZAK

20A91A04F4
21A95A0420
20A91A04F1
20A91A04H5



ADITYA ENGINEERING COLLEGE (A)

Aditya Nagar, ADB Road, Surampalem

VISION & MISSION OF THE INSTITUTE

Vision:

To emerge as a premier institute for quality technical education and innovation.

Mission:

M1: Provide learner centric technical education towards academic excellence

M2: Train on technology through collaborations

M3: Promote innovative research & development

M4: Involve industry institute interaction for societal needs

Sri Ram
↑

PRINCIPAL
PRINCIPAL
ADITYA ENGINEERING COLLEGE
SURAMPALEM - 533 437



ADITYA ENGINEERING COLLEGE (A)

Aditya Nagar, ADB Road, Surampalem

Department of Electronics & Communication Engineering

Vision and Mission of the Department

Vision:

To become a center of excellence in the field of Electronics and Communication Engineering with technological capability, professional commitment and social responsibility.

Mission:

M1: Provide quality education, well-equipped laboratory facilities and industry collaboration.

M2: Promote cutting edge technologies to serve the needs of the society and industry through innovative research.

M3: Inculcate professional ethics and personality development skills.

N. Ladha

Head of the Department

Head of the Department

Department of E.C.E.

Aditya Engineering College (A9)



ADITYA ENGINEERING COLLEGE (A)

Aditya Nagar, ADB Road, Surampalem

Department of Electronics & Communication Engineering

PROGRAM OUTCOMES(POs)

After successful completion of the program, the graduates will be able to

PO 1	Apply knowledge of mathematics, science, engineering fundamentals and an engineering specialization to the solution of complex engineering problems.
PO 2	Identify, formulate, research literature and analyze complex engineering problems, reaching substantiated conclusions using first principles of mathematics, natural sciences and engineering sciences.
PO 3	Design solutions for complex engineering problems and design systems, components or processes that meet specified needs with appropriate consideration for public health and safety, cultural, societal, and environmental considerations.
PO 4	Conduct investigations of complex problems using research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of information to provide valid conclusions.
PO 5	Create, select and apply appropriate techniques, resources, and modern engineering and IT tools, including prediction and modelling, to complex engineering activities, with an understanding of the limitations.
PO 6	Apply reasoning informed by contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to professional engineering practice.
PO 7	Understand the impact of professional engineering solutions in societal and environmental contexts and demonstrate knowledge of, and need for sustainable development.
PO 8	Apply ethical principles and commit to professional ethics and responsibilities and norms of engineering practice.
PO 9	Function effectively as an individual, and as a member or leader in diverse teams and in multidisciplinary settings.
PO 10	Communicate effectively on complex engineering activities with the engineering community and with society at large, such as being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
PO 11	Demonstrate knowledge and understanding of engineering management principles and apply these to one's own work, as a member and leader in a team and to manage projects in multidisciplinary environments.
PO 12	Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

N. Ladha
Head of the Department
Head of the Department
Department of E.C.E.
Aditya Engineering College (A9)



ADITYA ENGINEERING COLLEGE (A)

Aditya Nagar, ADB Road, Surampalem

Department of Electronics & Communication Engineering

Program Educational Objectives (PEOs)

The graduates of the Program will

PEO1:

Adapt the learning culture needed for a successful professional career and pursue research.

PEO2:

Build modern electronic systems by considering technical, environmental and social contexts.

PEO3:

Communicate effectively and demonstrate leadership qualities with professional ethics.

N. Ladha

Head of the Department

Head of the Department

Department of E.C.E.

Aditya Engineering College (A9)

Course Outcomes

After completion of the course the graduates will able to attain the following course outcomes

CO1: Identify a real life / engineering problem
CO2: Perform extensive investigation with prior knowledge
CO3: Interpret problem formulation and solution through critical thinking
CO4: Develop the work plan, schedule and estimate the cost
CO5: Identify the resources required to initiate project work
CO6: Apply the domain knowledge to arrive at a framework to solve the problem
CO7: Design solution using research-based knowledge and modern tools and interpret the results
CO8: Assess the obtained solution in the context of engineering framework addressing the societal and environmental concerns adhering to professional ethics
CO9: Demonstrate communication skills effectively to work as a team, for guide interaction and presentations.
CO10: Prepare technical documentation/reports with effective written communication skills

N. Dadha
Head of the Department
Head of the Department
Department of E.C.E.
Aditya Engineering College (A9)



ADITYA ENGINEERING COLLEGE (A)

Aditya Nagar, ADB Road, Surampalem

Department of Electronics & Communication Engineering

Project - PO Mapping

A.Y: 2023-24 (AR 20)

Sem.: VIII

Section & Batch: C15

Project Title: **DESIGN OF MEMRISTOR BASED LOW POWER ENCODER USING PGL TECHNIQUE**

Abstract

This project delves into the design and implementation of an innovative encoder architecture leveraging memristors integrated with Power Gating Logic (PGL), aiming to present a compelling alternative to traditional CMOS logic circuits. The study not only introduces a novel encoder architecture incorporating PGL based on memristors but also conducts comprehensive comparative analyses with established logic styles such as Memristor Logic (MRL), CMOS, and pseudo-NMOS. Through rigorous experimentation and evaluation, significant enhancements are demonstrated, including a noteworthy reduction in the power-delay product (PDP) from 10816×10^{-12} to 10666×10^{-12} and a substantial decrease in average power consumption from 80.7 milliwatts to 61.4 milliwatts. The utilization of Cadence Tools facilitates thorough evaluation of a 3-bit memristor encoder, employing the PGL methodology. This project not only showcases the feasibility and effectiveness of memristor-based logic circuits but also contributes valuable insights into optimizing power efficiency and performance in digital circuit design.

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
Overall Mapping	3	2	3	2	3	2	1	1	3	2	1	2

	PSO1	PSO2
Overall Mapping	3	1

Signature of Project Members:

1. 20A91A04F4

2. 21A95A0420

3. 20A91A04F1

4. 20A91A04H5

Guide Signature

INDEX

	Page No
List Of Figures	ii-iii
Abstract	v
CHAPTER 1 INTRODUCTION	1-6
1.1 Objective of project	1
1.2 Introduction of Memristor	1
1.3 Organization of the Project	6
CHAPTER 2 LITERATURE SURVEY	7-17
2.1 Survey	9
CHAPTER 3 ENCODER DESIGN ISSUE	18-63
3.1 Encoder	18
3.2 Encoder using CMOS Technology	19
3.3 Encoder using Pseudo NMOS Technology	21
3.4 Memristor working	23
3.5 Encoder Using MRL Technology	30
3.6 Power Gating Logic	31
3.7 Memristor based encoder using PGL Technique	32
3.8 Implementation in Cadence	35
CHAPTER 4 ENCODER DESIGN AND IMPLEMENTATION	63-73
4.1 Implementation of Encoder Using CMOS Technology	64
4.2 Implementation of Encoder Using Pseudo NMOS Technology	66
4.3 Implementation of Encoder Using MRL Technology	69
4.4 Memristor based encoder using PGL Technique	72
4.5 Advantages	73
4.6 Applications	73
CHAPTER 5 RESULT AND VALIDATION	74-75
5.1 Comparison of performance parameters	74
5.2 Analysis	75
CHAPTER 6 CONCLUSION AND FUTURE SCOPE	76
REFERENCES	77

LIST OF FIGURES

Fig No		Page No
1.1	The Symbol of Memristor	3
1.2	General Structure of passive components	3
3.1	3-Bit Encoder	18
3.2	Truth table of Encoder	19
3.3	Block Diagram of CMOS Logic	20
3.4	Encoder using CMOS Logic	21
3.5	Block diagram of Pseudo-NMOS Logic	22
3.6	Encoder using PSEUDO NMOS Logic	23
3.7	Hystersis loop for Memristor	24
3.8	Current direction for increased Resistance	25
3.9	Current direction for decreased Resistance	25
3.10	AND gate using Memristor	26
3.11	Memristive AND logic with 0 as inputs	26
3.12	Memristive AND logic with 1 as inputs	27
3.13	Memristive AND logic with one input is 1 another is 0	27
3.14	Memristive OR logic with one input is 1 another is 0	28
3.15	NAND gate using Memristor	29
3.16	NOR gate using Memristor	30
3.17	Encoder using memristor based logic	31
3.18	Block Diagram of Power Gating Approach	32
3.19	Memristor based encoder using PGL Technique	33
4.1	Block Diagram of CMOS Logic	64
4.2	Schematic Diagram of Encoder using CMOS Logic	65
4.3	Average power for encoder using CMOS Technology	66

4.4	Block Diagram of Pseudo NMOS	67
4.5	Schematic Diagram of Encoder using Pseudo NMOS Technology	68
4.6	Average power for encoder using Pseudo NMOS technology	69
4.7	Schematic Diagram of Encoder using MRL Technology	70
4.8	Average power consumed for the memristor based encoder	71
4.9	Schematic of memristor based PGL encoder	72
4.10	Input and output wave forms of memristor based encoder using PGL technique	74

ABSTRACT

This project delves into the design and implementation of an innovative encoder architecture leveraging memristors integrated with Power Gating Logic (PGL), aiming to present a compelling alternative to traditional CMOS logic circuits. Memristive devices, originally conceived for memory applications, present a promising avenue for advancing logic circuitry due to their inherent characteristics such as lower power consumption. The study not only introduces a novel encoder architecture incorporating PGL based on memristors but also conducts comprehensive comparative analyses with established logic styles such as Memristor Logic (MRL), CMOS, and pseudo-NMOS. Through rigorous experimentation and evaluation, significant enhancements are demonstrated, including a noteworthy reduction in the power-delay product (PDP) from 10816×10^{-12} to 10666×10^{-12} and a substantial decrease in average power consumption from 80.7 milliwatts to 61.4 milliwatts. The utilization of Cadence Tools facilitates the construction and thorough evaluation of a 3-bit memristor encoder, employing the PGL methodology. This project not only showcases the feasibility and effectiveness of memristor-based logic circuits but also contributes valuable insights into optimizing power efficiency and performance in digital circuit design.

CHAPTER 1

INTRODUCTION

Data protection is essential and difficult in the big data era since nobody can foresee when or where the next outage will happen. Technology for data backup is one of the greatest ways to reduce data loss and other undesirable effects. Therefore, the system's ability to sustain a steady and uninterrupted operation will depend on how well data backups and restores are executed. Data in the volatile memory will be moved to the nonvolatile memory during the backup process, and it will be restored after the downtime is over. The kind of memory and the system architecture have a major impact on data transfer latency. The effectiveness of data backup depends on how to reduce this latency.

1.1 OBJECTIVE OF THE PROJECT

The essential building blocks of nearly all digital electronic systems are combinational logic circuits. It suggests utilizing a memristor in a 3-bit combinational logic circuit for Power Gating Logic (PGL). Because memristor architecture consumes less power than CMOS logic, it provides an alternative to the current IC design for digital logic gates. One of the next computing architectures will be this one. Because a memristor can be created on top of an NMOS transistor's poly silicon gate, manufacturing MRL gates is straightforward. The density of transistors on a device is going to rise. The encoder design in three distinct configurations CMOS, Memristor, and Pseudo NMOS is shown in this project. When compared to other logic designs, the proposed 3-bit encoder design using MRL in PGL dissipates less power. This device is used to mimic various combinational logic circuits, and the primary goal of this project is to use Cadence Tool to construct and analyze a 3-bit encoder with various logics.

1.2 INTRODUCTION OF MEMRISTOR

One of these developing memories is the memristor, an acronym for "memory resistor." Leo Chua made the prediction in 1971, and HP Labs coined it in 2008. Numerous benefits, including nonvolatility, high density, low power consumption, excellent integration, and good compatibility with CMOS devices, have been investigated for this passive device. Additionally, it has been found that the volatile memory can store its intermediate states quickly and affordably when memristors are used in memory components.

So far, to design electronic circuits, passive elements like as capacitors, resistors and inductors are used, but a fourth fundamental element also exists, which is called a "memristor".

Memristance is just resistance that varies with charge, and its unit is the ohm.

The memristor was first proposed as a hypothetical non-linear passive two-terminal electrical component from the standpoint of circuit integrity. The remarkable nonvolatility attribute of a memristor is its ability to remember the amount and direction of electric charge that has passed through it in the recent past, which establishes its current resistance. The memristor retains its most recent resistance when the electrical power source is switched off and does not reset until the power is switched back on. A nonvolatile memory component can then be constructed by utilizing this attribute. Numerous models have been developed since HP Labs created the first useful memristor model to describe the electrical properties of memristors in terms of voltage, current, and state variables in various application scenarios. An ISQED survey contains a thorough overview of these models.

We use the voltage-controlled memristor model, which allows the memristor's resistance also known as its Memristance to abruptly change between values near the threshold voltage. In Figure 1.2.1, the memristor symbol is displayed. The negative terminal is the one on the right, indicated by the conspicuous black bar. The left terminal is called the positive terminal. The memristor's resistance state will change to R_{on} , often referred to as the Low Resistance State (LRS), when positive voltage exceeds the positive threshold, V_{set} . The memristor's resistance state will change to R_{off} , also known as High Resistance State (HRS), when the negative voltage is less than the negative threshold, V_{reset} . Otherwise, the resistance of the memristor remains unchanged. R_{off} and R_{on} have resistance levels that are several orders of magnitude different from one another. Because of this feature, it can be used to create nonvolatile devices and store binary values.



Fig1.1: The Symbol of Memristor

A memristor is an electrical component that limits or regulates the flow of electrical current in a circuit and remembers the amount of charge that has previously flowed through it. Memristors are important because they are non-volatile, meaning that they retain memory without power.

Memristors were first proposed as a nonlinear, passive two-terminal electrical component that connected electric charge and magnetic flux by Professor Leon Chua of the University of California, Berkeley in 1971.

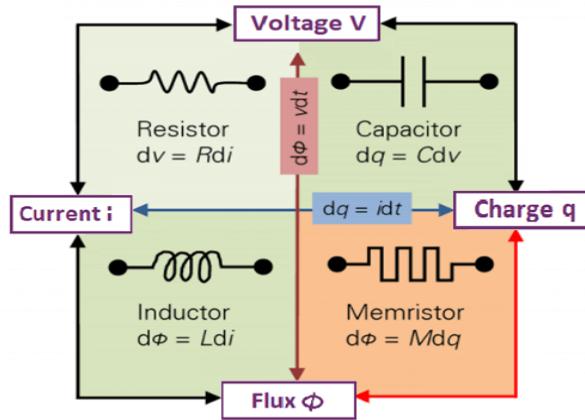


Fig 1.2 General structure of passive components

Since then, the term "memristor" has been expanded to refer to any type of non-volatile memory based on resistance switching, which results in an increase in one direction of current flow and a decrease in the other.

A memristor is frequently likened as a water-carrying imagined pipe. The diameter of the pipe expands as water flows in one direction, allowing the water to flow quicker; conversely, when water flows in the opposite way, the diameter of the pipe contracts, slowing the water's flow. The pipe's diameter does not change if the water is turned off until it is switched back on. To extend the analogy, a memristor maintains its resistance value even after its power is turned off. This implies that all open programs and documents would remain on the screen upon machine restarting in the event that the computer's power was abruptly switched off during a hard shutdown.

Memristors are one of numerous storage technologies that are expected to replace flash memory. They are regarded as a subcategory of resistive RAM. The first functional memristor was created in 2008 by scientists at HP Labs. Since then, memristors have been investigated by researchers at numerous major IT companies to see if they can be used to build computers that are quicker, smaller, and require less power consumption by transferring data between volatile and non-volatile memory. Theoretically, analog computers that could do calculations on the same chips that store data could be created if the storage hierarchy could be flattened by switching out DRAM and SSDs for memristors. Memristors are appealing to computer scientists because of a number of appealing features: They can store at least twice as much data in the same space as current solid-state storage

systems, and they work faster and with less energy use. Radiation can cause transistor-based technology to malfunction, while memristors are essentially impervious against it. Memristors can also be used to make computers that flicker on and off like light switches. Since a memristor has a voltage-dependent resistance as opposed to an electrical resistor's fixed resistance, a material's electric characteristics are crucial. The resistance of a memristor material needs to be able to vary reversibly with voltage. Memristors are made up of two metal electrodes separated by a tiny layer of titanium dioxide, which is its most basic structural element.

TYPES OF MEMRISTORS

1. Spintronic memristor.
2. Spin torque transfer magneto resistor.
3. Titanium dioxide memristor.
4. Polymeric memristor.
5. Spin mem system.
6. Magneto Memristive system.
7. Resonant tunneling dioxide memristor.

ADVANTAGES OF MEMRISTORS

- It is less expensive and faster than other devices like MRAM.
- There is more capacity to store more information.
- It does not lose information when the system is switched off.
- It has important information density.
- It produces less heat as it uses less energy.
- It has a greater data transfer rate.
- It uses a little amount of energy.
- It has the option to combine one short device, hard drives, and working memory.

APPLICATIONS OF MEMRISTORS

- Non-volatile memory applications.
- Digital circuits.
- Logic circuits.
- Biological and neuromorphic systems.
- Computer technology.
- Digital as well as analog memory.

1.3 ORGANIZATION OF PROJECT

This project is organized into 5 chapters.

Chapter 1: This chapter includes an overview of the project, an introduction to the memristor, and an explanation of its benefits and uses.

Chapter 2: The review of the literature and studies of encoder designs using CMOS, pseudo-NMOS, and MRL technologies are included in this chapter.

Chapter 3: This chapter includes a suggested way for creating encoders based on memristors utilizing the PGL technique. Additionally create the schematic diagrams for the encoders designed with CMOS, Pseudo NMOS, MRL, and PGL techniques, as well as the schematic diagram for the memristor-based encoder.

Chapter 4: This chapter includes simulation results for the encoder design using CMOS, pseudo-NMOS, MRL, and PGL techniques, as well as average power diagrams for the input and output waveforms.

Chapter 5: It consists of conclusion and future scope of the project.

CHAPTER -2

LITERATURE SURVEY

One of these developing memories is the memristor, an acronym for "memory resistor." Leo Chua made the prediction in 1971, and HP Labs coined it in 2008. Numerous benefits, including nonvolatility, high density, low power consumption, excellent integration, and good compatibility with CMOS devices, have been investigated for this passive device. Additionally, it has been found that the volatile memory can store its intermediate states quickly and affordably when memristors are used in memory components.

So far, to design electronic circuits, passive elements like as capacitors, resistors and inductors are used, but a fourth fundamental element also exists, which is called a "memristor".

Memristance is simply charge-dependent resistance and the unit of the memristance is the ohm.

As a hypothetical non-linear passive two-terminal electrical component, the concept of memristor was first proposed based on the perspective of the circuit integrity. Memristor has the notable non volatility property of having the memory of how much electric charge has flowed and in which direction through it in the recent past, which determines its present resistance. When the electric power supply is turned off, the memristor remembers its most recent resistance until the power is turned on again. This property can then be exploited to build a nonvolatile memory component. Since the first practical memristor model was built by HP Labs, many other models have been proposed to characterize the electrical characteristics of memristors in terms of voltage, current, and state variables in different application scenarios. A detailed description of these models can be found in an ISQED survey.

We adopt the voltage-controlled memristor model in which the resistance of the memristor (i.e., memristance) can be switched abruptly from one value to another around the threshold voltage. The right terminal, denoted by the bold black bar, is called the negative terminal. The left terminal is called the positive terminal. When positive voltage is greater than the positive threshold, V_{set} , the resistance state of the memristor will switch to R_{on} , which is known as the Low Resistance State (LRS). When the negative voltage on the memristor is smaller than the negative threshold, V_{reset} , the resistance state of the memristor will switch to R_{off} , also called High Resistance State (HRS). Otherwise, the resistance of the

memristor remains unchanged. The difference in the resistance values between Roff and Ron is several orders of magnitude. Such property makes it suitable to store binary values and implement nonvolatile devices.

A memristor is an electrical component that limits or regulates the flow of electrical current in a circuit and remembers the amount of charge that has previously flowed through it. Memristors are important because they are [non-volatile](#), meaning that they retain [memory](#) without [power](#).

The original concept for memristors, as conceived in 1971 by Professor Leon Chua at the University of California, Berkeley, was a nonlinear, passive two-terminal electrical component that linked [electric charge](#) and magnetic [flux](#).

Since then, the definition of memristor has been broadened to include any form of non-volatile memory that is based on resistance switching, which increases the flow of current in one direction and decreases the flow of current in the opposite direction.

A memristor is frequently likened as a water-carrying imagined pipe. The diameter of the pipe expands as water flows in one direction, allowing the water to flow quicker; conversely, when water flows in the opposite way, the diameter of the pipe contracts, slowing the water's flow. The pipe's diameter does not change if the water is turned off until it is switched back on. To extend the analogy, a memristor maintains its resistance value even after its power is turned off. This implies that all open programs and documents would remain on the screen upon machine restarting in the event that the computer's power was abruptly switched off during a hard shutdown.

Memristors are one of numerous storage technologies that are expected to replace flash memory; they are regarded as a subcategory of resistive RAM. Since the first functional memristor was created in 2008 by scientists at HP Labs, numerous major IT companies have conducted research on the potential applications of memristors in the creation of smaller, quicker, low-power computers that do not require the transfer of data between volatile and non-volatile memory. Theoretically, analog computers that could do calculations on the same chips that store data could be created if the storage heirarchy could be flattened by switching out DRAM and SSDs for memristors.

This project is designed by using the following papers as reference.

2.1 SURVEY

L. Chua, "Memristor-The missing circuit element," in IEEE Transactions on Circuit Theory, vol. 18, no. 5, pp. 507-519, September 1971, doi: 10.1109/TCT.1971.1083337.

The fourth fundamental circuit element is the introduction of the memristor, a novel two-terminal circuit element distinguished by a relationship between the charge and the flux-linkage. This relationship is explained in terms of a quasi-static expansion of Maxwell's equations in the context of an electromagnetic field. Memristors' numerous circuit-theoretical characteristics are deduced. It is demonstrated that this element displays some odd behavior distinct from that of capacitors, resistors, and inductors. These characteristics give rise to certain special applications that RLC networks alone are unable to accomplish. Even though there hasn't been a physical memristor device found yet, functional laboratory models have been constructed with the aid of active circuits. The characteristics and possible uses of memristors are illustrated through the presentation of experimental findings.

Liu, Gongzhi & Shen, Shuhang & Jin, Peipei & Wang, Guangyi & Liang, Yan. (2021). Design of Memristor-Based Combinational Logic Circuits. Circuits, Systems, and Signal Processing. 40. 1-22. 10.1007/s00034-021-01770-1.

Three modified memristor ratioed logic (MRL) gates are proposed in this paper: NOT, NOR, and A AND (NOR B) (i.e., $A \cdot \bar{B}$), each of which only requires one NMOS and one memristor. We design a few combinational logic circuits, such as a 4:1 multiplexer, 3-bit binary encoder, 3-bit binary decoder, and 1-bit comparator, based on the modified MRL gates. Additionally, a better multifunctional logic module that can do AND, OR, and XOR logic operations is provided. It consists of five memristors and one NMOS transistor. This versatile logic module is used to create a 1-bit full adder and a 4-bit comparator.

Singh memristor based XNOR for high-speed area efficient 1-bit full adder2017 international conference on computing, communication and 2017 ieeexplore.ieee.org

This study uses the Memristor nano-device characteristic to construct an area-efficient high-speed full adder. Compared to previous logic gate and full adder designs, the new full adder circuit has a simpler architecture, faster processing, and more power efficiency. Only fifteen NMOS transistors and fifteen memristors make up the adder. The primary objective of the project is to create a memristor-based alternative Computation-In-Memory architecture and an efficient extension of Moore's Law.

S. Kvatinsky, G. Satat, N. Wald, E. G. Friedman, A. Kolodny, and U. C. Weiser, “**Memristor-based material implication (imply) logic: Design principles and methodologies,” IEEE Transactions on Very Large-Scale Integration (VLSI) Systems, vol. 22, no. 10, pp. 2054–2066, 2013.**

A natural method for using memristors to do out logic operations is an IMPLY logic gate. A memristor-based memory can be equipped with the entire logic family that this logic gate and FALSE together constitute. This memristive logic gate also opens the door to non-von Neumann designs, perhaps launching a new chapter in computer architecture history.. The potential density and power benefits of memristive circuits inspire more research in this field. The results in this paper can direct future research on array topologies, computing architectures, logic synthesis methods, and device structure optimization. A basic logical element that employs this technique is the material implication (IMPLY logic gate) [7], which combines a Boolean operator with state memory. There have been other logic families produced as well [8], [9], which are not included in this page. These families use variants of a standard memristor-based crossbar to extend the IMPLY logic gate. However, this study presents a specific modification of the crossbar structure to improve the logic gate's performance.

S Kvatinsky, A Kolodny, UC Weiser, EG Friedman “memristor-based imply logic design procedure”2011 IEEE 29th International Conference on Computer Design (ICCD), 2011 ieeexplore.ieee.org

A memristor-based IMPLY logic gate's logic design is shown. A number of design constraints and considerations are made clear by examining and analyzing the behavior of an IMPLY logic gate and memristor. The IMPLY logic gate compromises resilience (internal state drift) for efficiency (write time). This trade off requires the circuit to be occasionally refreshed. We suggest and arrange a set of heuristics for the construction of IMPLY logic gates using memristors. This design process takes into account the effects and trade-offs between the various input scenarios, beginning conditions, and memristor circuit characteristics.

E Lehtonen, JH Poikonen, M Laiho “Applications and limitations of memristive implication logic 2012 international workshop on cellular nanoscale network 2012 ieeexplore.iee.org

We have shown in this paper that a hardware architecture of the CMOL type can perform parallel stateful logic. In essence, bit-wise row or column vector operations within memristive crossbars are made possible by the methodologies presented. It is important to

remember that memristive logic can only apply one stateful logic operation at a time, despite its ability to handle multiple inputs and outputs. For instance, it is not feasible to execute stateful logic operations over any element inside a crossbar or to change the contents of a vector in a single step. Since there are only four microwires available for addressing, a CMOL architecture cannot handle stateful multi-input and multi-output operations, as was described in Subsection IV-C.

While it is possible to increase the number of address wires, this might not be a practical option in everyday situations. This appears to suggest that in order to pick many nanowires at once, the CMOS cells' logic levels should be raised. But adding local memory to the cells alone is insufficient because these local bits would always be stored sequentially, which would lessen or eliminate the advantages of multi-variable operations. This implies that the selection of the nanowires should be known to the CMOS cells from within the memory architecture itself, instead than being known externally through address decoders.

It is possible that an additional memristive crossbar layer exists, which serves to store the control signal patterns associated with the computational sequences. In order to facilitate basic vector operations like the shift operation, there should also be communication amongst the CMOS cells. In Section V, we looked into memristive binary Cellular Neural/Nanoscale Networks as an example of a design meeting these characteristics. CNNs are a good application option for this type of memristive logic since they can carry out stateful operations in a fully parallel fashion.

Suri Shanmukh, Sompalli RohitKumar, Paluru Hemaprasad[‡] and R. Sakthivel “Low Power 3-Bit Encoder Design using Memristor”2nd International Conference on Intelligent Technologies (CONIT) Karnataka, India. June 24-26, 2022

A memristor-based approach to digital logic gate design provides an alternative to the current integrated circuit architecture. One of the next Computer Architectures will be this one. Because a memristor may be created on top of an NMOS transistor's polysilicon gate, manufacturing MRL gates is straightforward. The density of transistors on a device is going to rise. In comparison to CMOS design, the proposed 3-bit encoder design with MRL dissipates less power. This device is designed to represent various combinational logic circuits, and the primary goal of this work is to use Cadence to construct and analyze a 3-bit encoder with various logics.

S. Smaili and Y. Massoud, “**Analytic modeling of memristor variability for robust memristor systems designs,**” in **2014 IEEE International Symposium on Circuits and Systems (ISCAS).** IEEE, 2014, pp. 794–797.

The nonlinear passive element called Memristor is used and an emulator has been designed. The linear and nonlinear characteristics of the device are studied and also simulation work has been carried out using Multisim. Indeed, no physical memristor device was not available in market, Multisim is used and emulator circuit has been designed and device characteristics are analyzed. A simulation result depicts the effectiveness of the memristor. This study explicitly shows that memristor is effective in all aspects of linear and nonlinear characteristics.

K. Mbarek, F. O. Rziga, S. Ghedira, and K. Besbes, “**Design and properties of logic circuits based on memristor devices,**” in **2020 IEEE International Conference on Design & Test of Integrated Micro & Nano Systems (DTS).** IEEE, 2020, pp. 1–5.

In this paper, a logic circuit based on memristor devices is discussed for future digital applications. Its I-\$V\$ characteristics are compared to a physical memristor to verify the model similarity with a real memristor device. An interface circuit is introduced between the memristor model and the CMOS components. The interfacing circuit is composed of a memristor (the resistance state) and a transistor. This interfacing circuit is very important to incorporate and integrate memristors into CMOS based digital applications. In this context, we examined the model behavior under the configuration of the 1T1R OxRRAM. We performed a circuit-level comparison of the proposed design with other designs proposed in the literature. We validate a non-volatile NAND gate based on 1T1R cell, then we discuss its possible configuration states “00”, “01”, “10”, and “11”. In addition, we implement this 1T1R cell on a crossbar array application and examine its operation. Thus, these memory-based CMOS circuits are potentially beneficial for future large-scale applications.

S. Mandal, J. Sinha, and A. Chakraborty, “**Design of memristor– cmos based logic gates and logic circuits,**” in **2019 2nd International Conference on Innovations in Electronics, Signal Processing and Communication (IESC).** IEEE, 2019, pp. 215–220.

A great deal of effort has recently been devoted to extending the usage of memristor technology from memory to computing. Memristor-based logic design is an emerging concept that targets efficient computing systems. Several logic families have evolved, each with different attributes. Memristor Ratioed Logic (MRL) has been recently introduced as a hybrid memristor–CMOS logic family. MRL requires an efficient design strategy that

takes into consideration the implementation phase. This paper presents a novel MRL-based crossbar design: X-MRL. The proposed structure combines the density and scalability attributes of memristive crossbar arrays and the opportunity of their implementation at the top of CMOS layer. The evaluation of the proposed approach is performed through the design of an X-MRL-based full adder. The design is presented with its layout and corresponding simulation results using the Cadence Virtuoso toolset and CMOS 6565nm process. The comparison with a pure CMOS implementation is promising in terms of the area, as our approach exhibits a 44.79% area reduction. Moreover, the combined Energy-Delay metric demonstrates a significant improvement with respect to the available literature.

K. Alammari, A. Ahmadi, and M. Ahmadi, “Hybrid memristor-cmos based up-down counter design,” in 2020 27th IEEE International Conference on Electronics, Circuits and Systems (ICECS). IEEE, 2020, pp. 1–4.

In this paper, an optimized memristor emulator circuit is designed, by using nine MOSFET transistors and a ground capacitor. Our area- and power-optimized emulator circuit can be used for basic data storage and processing at the monitoring edge, in real-time applications. The memristor shows a nonlinear voltage-current relationship, but no multiplier circuit provides the memristor's nonlinear characteristics. As a result, the proposed memristor emulator has a very low chip area. The memristor circuit is designed in LTSpice, using 16 nm and 45 nm CMOS technology parameters, and the operating voltage is ± 0.9 V. In this research, the theoretical derivations are validated using the simulated results of the memristor emulator circuit using different frequencies, capacitors, and input voltages in SPICE simulations.

A. Sasi, M. Ahmadi, and A. Ahmadi, “Low power memristor-based shift register design,” in 2020 27th IEEE International Conference on Electronics, Circuits and Systems (ICECS). IEEE, 2020, pp. 1–4.

The use of memristors is considered to be an appropriate alternative solution to Complementary Metal Oxide Semiconductor (CMOS) technology's scaling limitation. In digital design, shift registers are widely used and considered to be basic memory devices. In this paper, a fast and efficient area memristor-only-based shift register, as well as a hybrid CMOS/memristor-based shift register are proposed. Specifically, a 4-bit shift register with only 8 memristor devices and a hybrid CMOS /memristor with 64 memristor devices and 64 CMOS transistors were implemented and simulated using Cadence Virtuoso. The simulation results demonstrate the design's efficient functionality. Compared

to the implementation of a CMOS-memristor based shift register, the implementation of the proposed design is more efficient when concerning area and speed with respect to the implementation of the Memristor Based-Material-Implication (IMPLY) memristive shift register. In addition, the shift register with only memristor-based has a significant power reduction of over 30% compared to a CMOS design shift register.

D. B. Strukov, D. R. Stewart, J. Borghetti, X. Li, M. Pickett, G. M. Ribeiro, W. Robinett, G. Snider, J. P. Strachan, W. Wu et al., “Hybrid cmos/memristor circuits,” in 2010 IEEE International Symposium on Circuits and Systems (ISCAS). IEEE, 2010, pp. 1967–1970.

This paper is a brief review of recent work on the prospective hybrid CMOS/memristor circuits. Such hybrids combine the flexibility, reliability and high functionality of the CMOS subsystem with very high density of nanoscale thin film resistance switching devices operating on different physical principles. Simulation and initial experimental results demonstrate that performance of CMOS/memristor circuits for several important applications is well beyond scaling limits of conventional VLSI paradigm.

S. Kvatinsky, N. Wald, G. Satat, A. Kolodny, U. C. Weiser, and E. G. Friedman, “Mrl—memristor ratioed logic,” in 2012 13th International Workshop on Cellular Nanoscale Networks and their Applications. IEEE, 2012, pp. 1–6.

Memristive devices are novel structures, developed primarily as memory. Another interesting application for memristive devices is logic circuits. In this paper, MRL (Memristor Ratioed Logic) — a hybrid CMOS-memristive logic family — is described. In this logic family, OR and AND logic gates are based on memristive devices, and CMOS inverters are added to provide a complete logic structure and signal restoration. Unlike previously published memristive-based logic families, the MRL family is compatible with standard CMOS logic. A case study of an eight-bit full adder is presented and related design considerations are discussed.

L. O. Chua and S. M. Kang, “Memristive devices and systems,” Proceedings of the IEEE, vol. 64, no. 2, pp. 209–223, 1976.

A broad generalization of memristors--a recently postulated circuit element--to an interesting class of nonlinear dynamical systems called memristive systems is introduced. These systems are unconventional in the sense that while they behave like resistive devices, they can be endowed with a rather exotic variety of dynamic characteristics. While possessing memory and exhibiting small-signal inductive or capacitive effects, they are incapable of energy discharge and they introduce no phase shift between the input and

output waveforms. This zero-crossing property gives rise to a Lissajous figure which always passes through the origin. Memristive systems are hysteretic in the sense that their Lissajous figures vary with the excitation frequency. At very low frequencies, memristive systems are indistinguishable from nonlinear resistors while at extremely high frequencies, they reduce to linear resistors.

CHAPTER-3

ENCODER DESIGN ISSUE

3.1 ENCODER

A combinational circuit called an encoder carries out the opposite function of a decoder. There can be a maximum of 'n' output lines and 2^n input lines. It will output an equivalent binary code to the input, which is high when active. As a result, the encoder uses "n" bits to encode the first 2^n input lines. The encoder's representation of the enable signal is optional.

Binary bits that correspond to certain meanings are used in digital logic circuits to encode information. An encoder is a circuit that performs this encoding task. When one of the input bits is of an effective level, the encoder's job is to encode; its output varies based on the bits it receives. The circuit's "N" outputs and "M" inputs are connected by the formula $M = 2^N$.

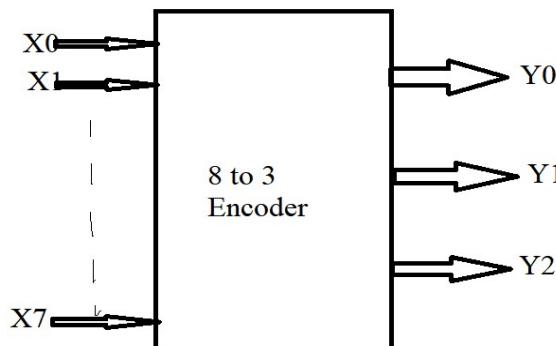


Fig 3.1: 3-bit encoder

X_0	X_1	X_2	X_3	X_4	X_5	X_6	X_7	Y_2	Y_1	Y_0
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1

Fig 3.2: Truth table encoder

From the Encoder truth table, the outputs and inputs are related by

$$Y_0 = X_1 + X_3 + X_5 + X_7$$

$$Y_1 = X_2 + X_3 + X_6 + X_7$$

$$Y_2 = X_4 + X_5 + X_6 + X_7$$

These relationships allow CMOS, Pseudo NMOS, and MRL to be used to create logic circuits. X0 through X7 are the input bits while Y2, Y1, and Y0 are the output bits in the encoder circuits. Equivalent binary information, or 000, is retrieved in the encoder circuit when the X0 input is enabled. When the X1 input is active output is the binary representation of 1, or 001 is obtained. In a similar manner, all inputs are applied, and output is produced based on input. The OR operation of the inputs X1, X3, X5, and X7 yields output Y0. Y1 is the result of the OR operation on X2, X3, X6, and X7. In a similar manner, X4, X5, X6, and X7 OR operations provide Y2.

3.2 ENCODER USING CMOS TECHNOLOGY

The most commonly used logic these days is CMOS logic (Complementary metal oxide semiconductor). In this logic both N-type and P-type are used. A transistor is turned on by a signal, while another transistor is turned off by the same signal. Instead of using pull-up resistors to implement this logic, just basic switches are needed. P-type transistor collections are arranged in CMOS logic as pull-up networks between the output and the high voltage rail, and N-type transistor collections are arranged as pull-down networks between the output and the low voltage rail. P-type transistors are ON and N-type transistors are OFF when their gates are coupled to the same input signal, and vice versa

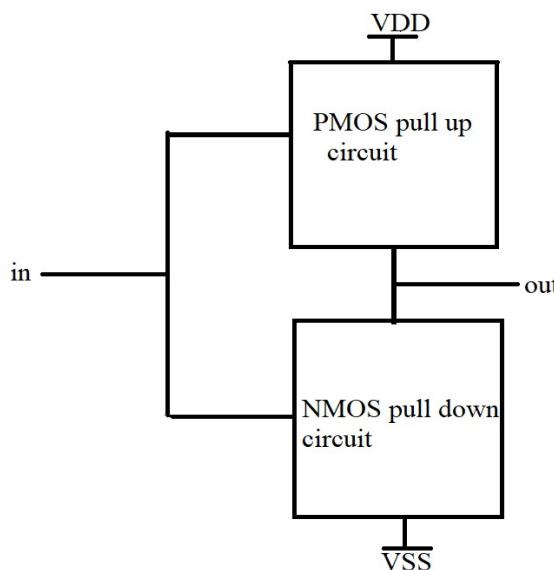


Fig 3.3: Block diagram of CMOS logic

Each logic gate in CMOS logic is implemented by a pair of complementary MOSFETs, one n-type and one p-type. Adders, multipliers, and memory systems are examples of sophisticated digital circuits that can be made by combining these gates.

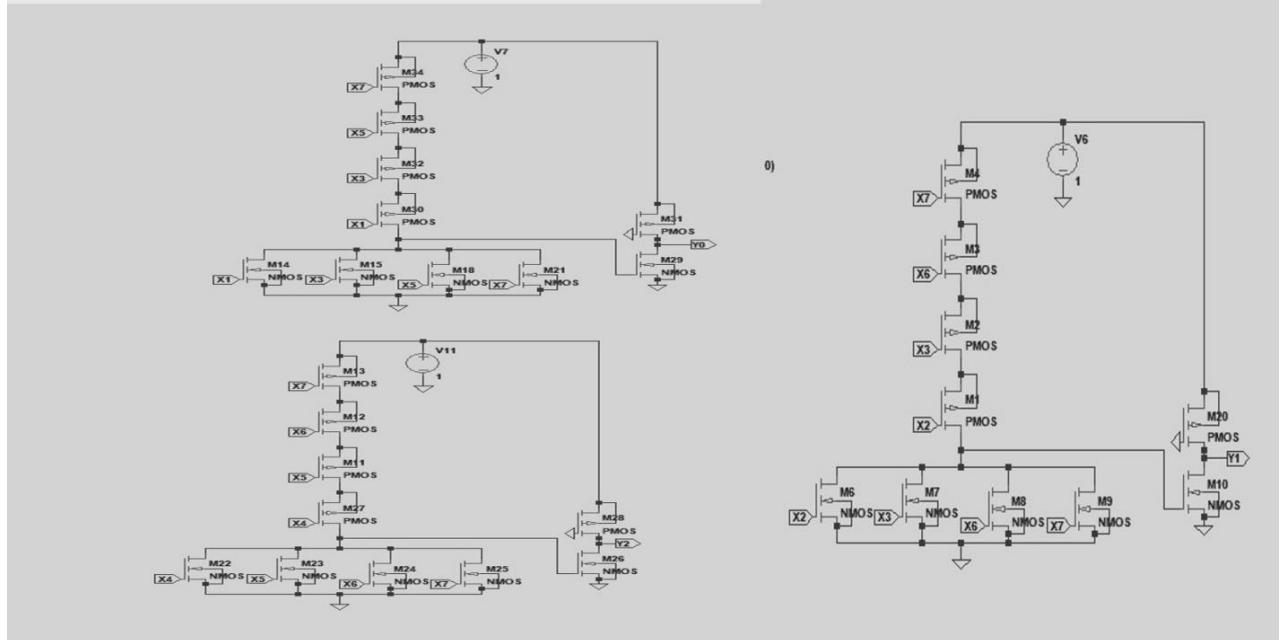


Fig 3.4: Encoder using CMOS Logic

3.3 ENCODER USING PSEUDO NMOS TECHNOLOGY

One PMOS device serves as the pull-up device for a multi-transistor N-Logic block in pseudo-NMOS logic. Consequently, $N+1$ transistors are needed to provide a N input gate circuit. The gate of the

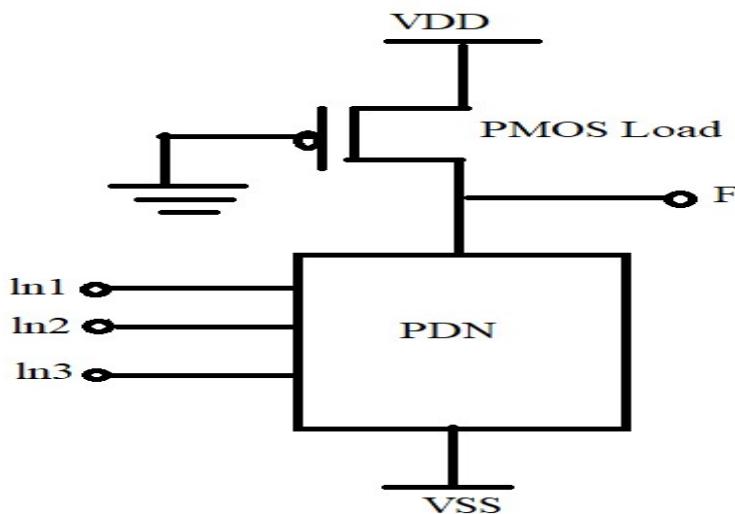


Fig 3.5 Block diagram of Pseudo-NMOS Logic

used PMOS device is linked to ground. Its V_{gs} is therefore set to -V_{dd}. As a result, the PMOS transistor is constantly in the "ON" state. The output of the N-Logic block is charged to V_{dd} (Logic '1') when it is "OFF". A significant current that flows from V_{dd} to ground when the N-block is "ON" can result in a significant amount of static power consumption. Pseudo-NMOS devices require precise device dimensions in order to allow the output to be discharged to ground. The PMOS device needs to be selected such that it is both thin enough to allow the N—block to safely pull down the output and wide enough to conduct a multiple of the leakage current of the N—block when the output is "high". As a result, ratioed logic characterizes pseudo-NMOS logic. The speed advantage of pseudo-NMOS logic over static CMOS logic is particularly noticeable in big fan-in NOR gates.

This is because the output rise time is only being contributed by one PMOS transistor. There has been a noticeable gain in speed overall, albeit with a little rise in power consumption.

Before CMOS technology was developed, pseudo NMOS logic was a common form of digital logic circuit utilized in the early days of integrated circuit design. Before CMOS technology was developed, a form of digital logic circuit known as pseudo NMOS logic was commonly employed in the early stages of integrated circuit design. When the input is high, the NMOS transistor in a pseudo NMOS gate pulls the output voltage down to ground. The pull-up resistor controls the output voltage while the transistor is off when the input is low.

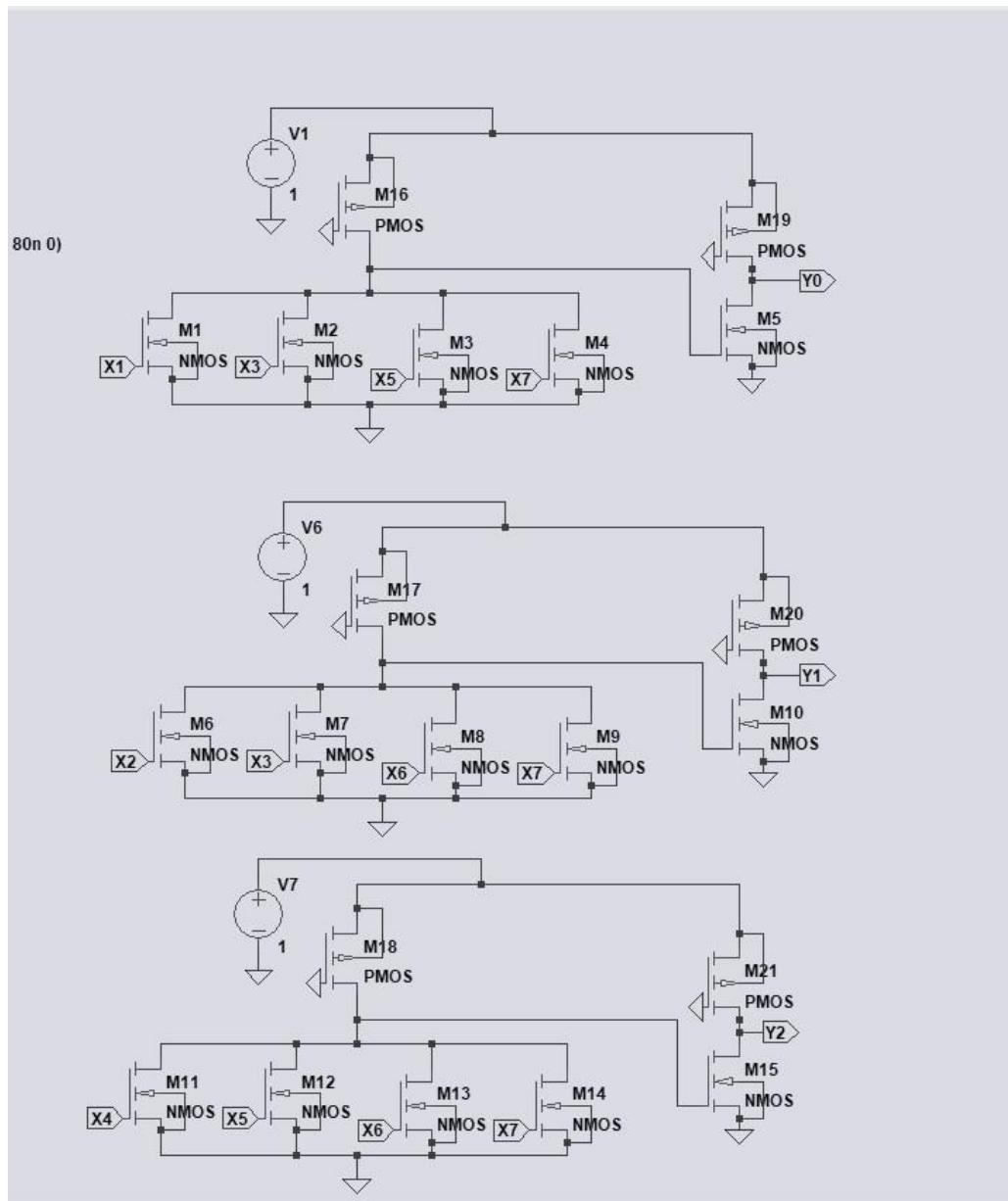


Fig 3.6: Encoder using PSEUDO NMOS Logic

3.4 MEMRISTOR WORKING

When we increase the positive voltage of the memristor, the current through it will increase very little and this state is called the high resistance state. If we further increase the voltage slightly, the current through it will increase rapidly and reached state called the low resistance state as shown in Fig. 2.5.1. We will be using these states for our switching. The memristor has a different voltage and current curve when compared to the basic resistor, inductor, and capacitor. This is known as the pinched hysteresis loop. The current decreases with increasing positive voltage and reaches zero when the voltage does. Similarly, decreasing negative voltage causes the current to drop slowly and reach a state of high

resistance. If the voltage drops further, however, the current drops quickly and reaches a state of low resistance. At this point, increasing negative voltage causes the current to increase as well. When the voltage This phenomenon of curve intersecting at the zero voltage and zero current is called as the pinched.

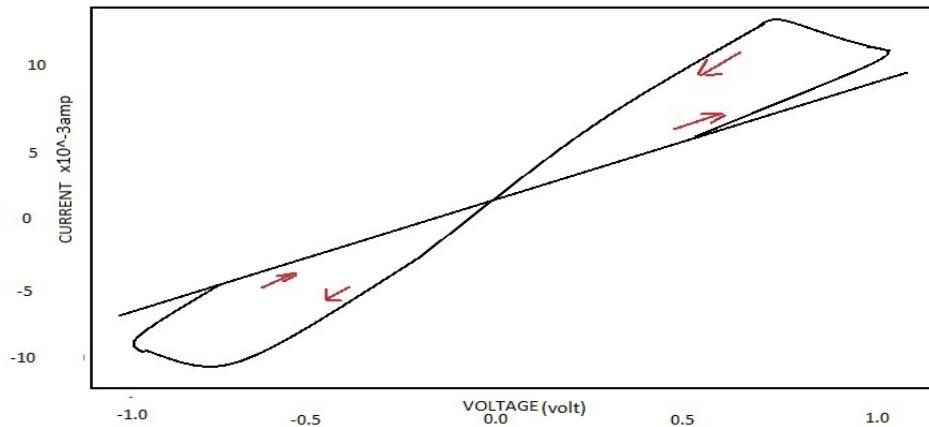


Fig 3.7 Hysteresis loop of the memristor

We will display the polarity of the memristor as it is depicted in Fig. 2.5.2. When current flows into the black spot, the device's resistance decreases, and when current flows out of the black mark, the device's resistance increases. Figure 2.5.3, an integrated memristor with standard logic, shows voltage as a logic state identical to that of normal CMOS logic, allowing the memristor to be used as a combinational element regardless of the device's memory characteristics and beginning state. In Fig. 2.5.1.1, an AND gate memristor circuit is displayed.

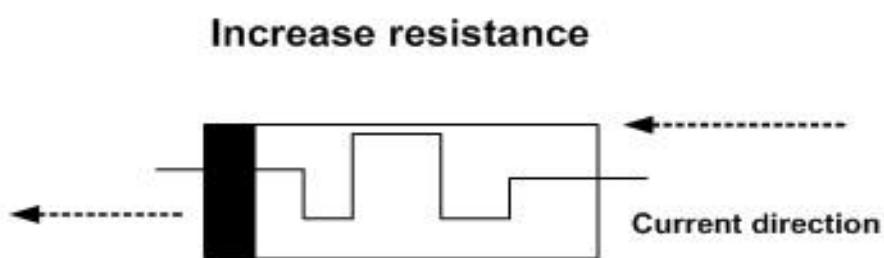


Fig 3.8 memristor current direction for increased resistance

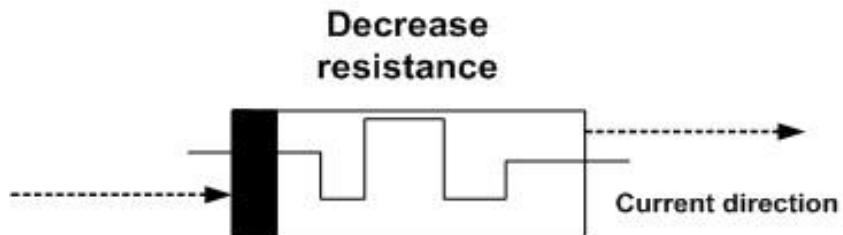


Fig 3.9 memristor current direction for decreased resistance

IMPLEMENTATION OF AND LOGIC GATE USING MEMRISTOR

The logic operation is illustrated in Fig. 2.5.1.1 below. If both inputs have logic 0 or are connected to ground, we assume that the output node is floating. Since there is no voltage applied to the memristor, no current flows through it, and the output is therefore either ground or logical 0. In Fig. 2.5.1.2, both input nodes are connected to VDD or have logical 1, so no current flows through the memristor again, causing the input voltage to appear across the logical 1 output. If, however, one input is logical 1 and the other is 0 (two scenarios), then current flows from the upper memristor to the bottom memristor. In this scenario, the resistance of the upper memristor is increasing and the resistance of the lower memristor is decreasing due to the polarity of the memristor. As a result, after a sufficient amount of time, the voltages will reach the maximum resistance of the It is simply a regular voltage divider because the lower memristor R_{on} has minimum resistance in the low resistance state and the upper memristor R_{off} in the high resistance state. As a result, if R_{off} is significantly higher than R_{on} , the voltage at the output is roughly equal to ground or logical 0.

$$V_{out} = [R_{on}/(R_{on}+R_{off})]V_{cc}$$

$$V_{out} = [R_{on}/R_{off}]V_{cc} \quad (R_{off} \gg R_{on})$$

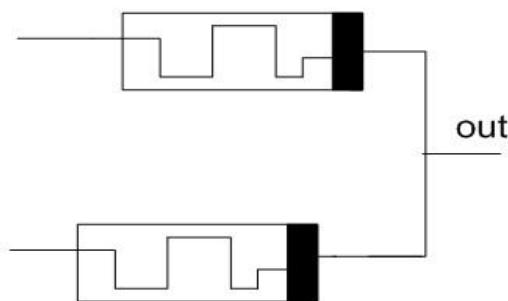


Fig 3.10 AND gate using memristor

- When both inputs are 0 are applied, no current flows through it.

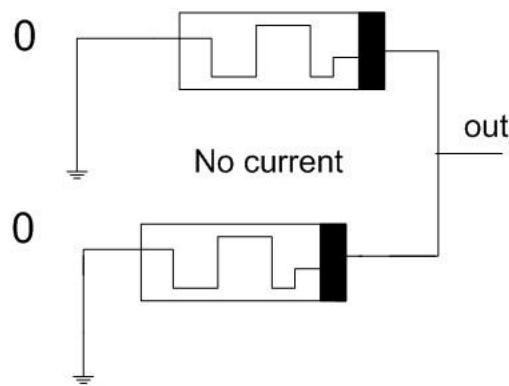


Fig 3.11 Memristive AND logic with 0 as Inputs

- When both inputs are applied 1, the output is high.

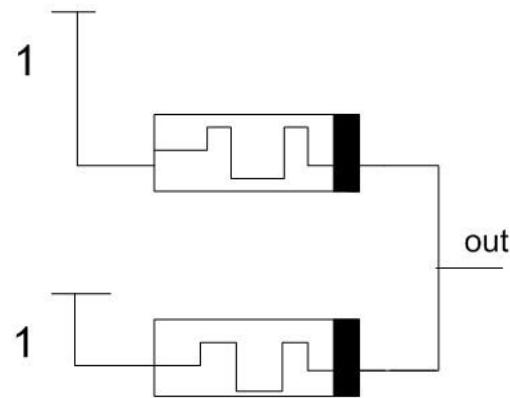


Fig 3.12 Memristive AND logic with 1 as Inputs

There will be no output and both increased and decreased resistance on the other side of the memristor (Fig. 2.5.1.4) when unusual inputs are applied.

.

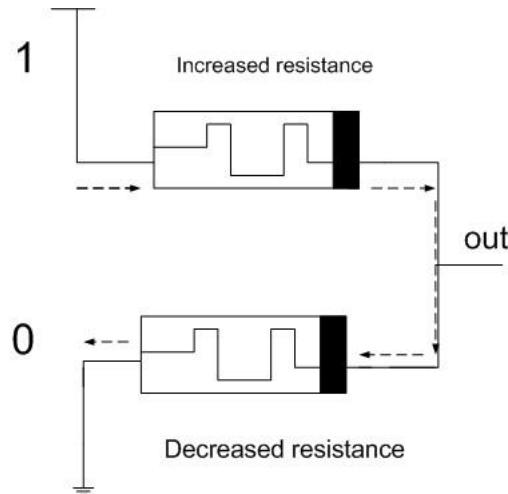


Fig 3.13 Memristive AND logic with one Input as 1 and another as 0

IMPLEMENTATION OF OR LOGIC GATE USING MEMRISTOR

Simply switch the memristor's terminals for the OR gate. In the event that both inputs are the same logical value, no current will flow, and the output will be the same voltage as the input voltage. In the event that both inputs are logical 0, the output will be logical 0, and in the event that both inputs are logical 1, the output will be logical 1. If you consider the other two scenarios, in which the inputs are logical 1 and logical 0 respectively, current will flow from the upper to the lower memristor, causing the resistance of the memristor to decrease and the resistance of the lower memristor to increase over time, resulting in the minimum resistance for the upper memristor and the maximum resistance for the lower memristor. the lower memristor and again it acts as a voltage divider output is logical 1(VCC) as shown in Fig 2.5.2.1

$$V_{out} = [R_{off} \div (R_{on} + R_{off})] V_{cc}$$

$$V_{out} = V_{cc} (R_{off} \gg R_{on})$$

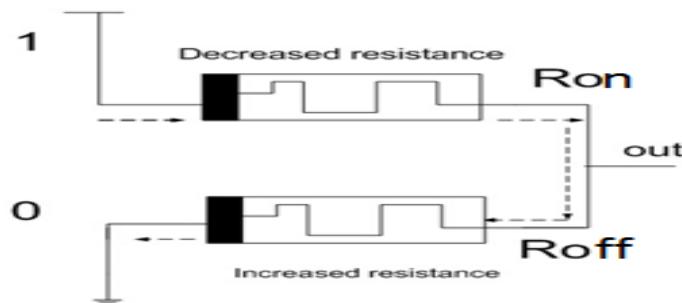


Fig 3.14 Memristive OR logic with one Input as 1 and another as 0.

IMPLEMENTATION OF NAND & NOR LOGIC GATE USING MEMRISTOR

Memristors are passive elements, so if there is a 1 percent loss in stage 1 and a 2 percent loss in stage 2, the 3 percent total loss is not restored in the three stages. This can be avoided by using memristors with standard CMOS logic, as shown in Fig. 2.5.3.1, because memristors have voltage as a logic state. This is the main drawback of the memristive chain. The idea is to use standard CMOS logic to amplify the voltage level using regular CMOS logic. To integrate memristor with standard CMOS logic, simply connect the AND and OR gates of the memristor, then follow the CMOS inverter. This will implement NAND and NOR logic in the memristor and CMOS logic, as illustrated in Fig. 2.5.3.2.

NAND LOGIC USING MEMRISTOR

All that is required to use a memristor with typical CMOS logic is to link the AND and NAND logic gates of the CMOS inverter and memristor. NAND gate logic is produced by connecting two memristors, M1 and M2, in parallel, and feeding the output of that to a CMOS inverter. The potential across the terminals is zero if we set A=0 and B=0. Subsequently, there is only one CMOS inverter. In the same way, the potential across the terminals is zero for A=0,B=1 and A=1,B=0. Subsequently, there is only one CMOS inverter. The potential across the terminals is 0 for A=1, B=1. The CMOS inverter is then zero out.

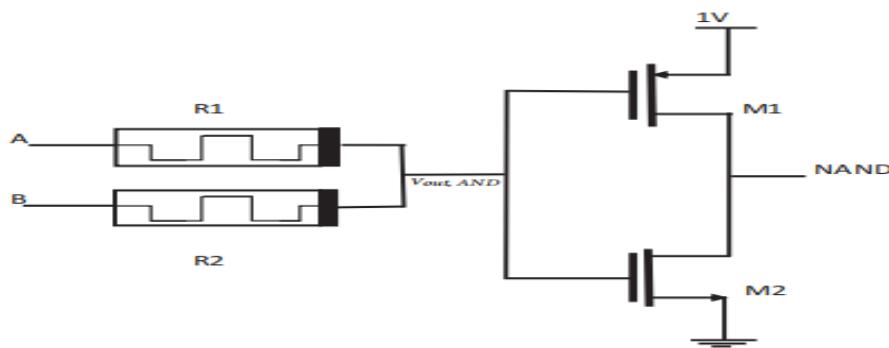


Fig 3.15 NAND gate using Memristor.

NOR USING MEMRISTOR

All that a memristor needs to do to work with typical CMOS logic is to attach an OR gate to it, then a NOR logic gate to the CMOS inverter. By connecting two memristors, M1 and M2, in parallel, and feeding the output to a CMOS inverter, NOR gate logic is produced. The potential across the terminals is zero if we set A=0 and B=0. Subsequently, there is only one CMOS inverter. In the same way, the potential across the terminals is one for A=0, B=1 and A=1,B=0. Then, there is zero out of the CMOS inverter. The potential across the terminals is one for A=1, B=1. Then, there is zero out of the CMOS inverter.

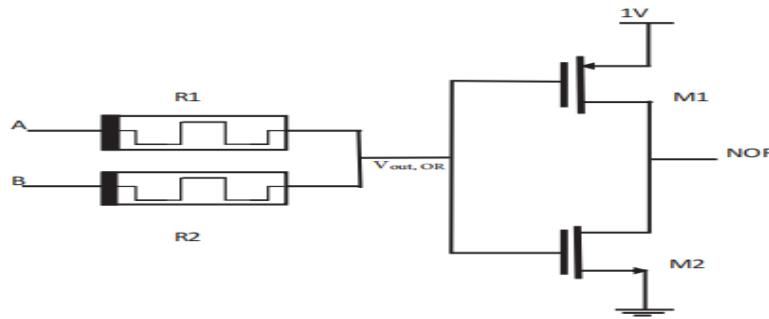
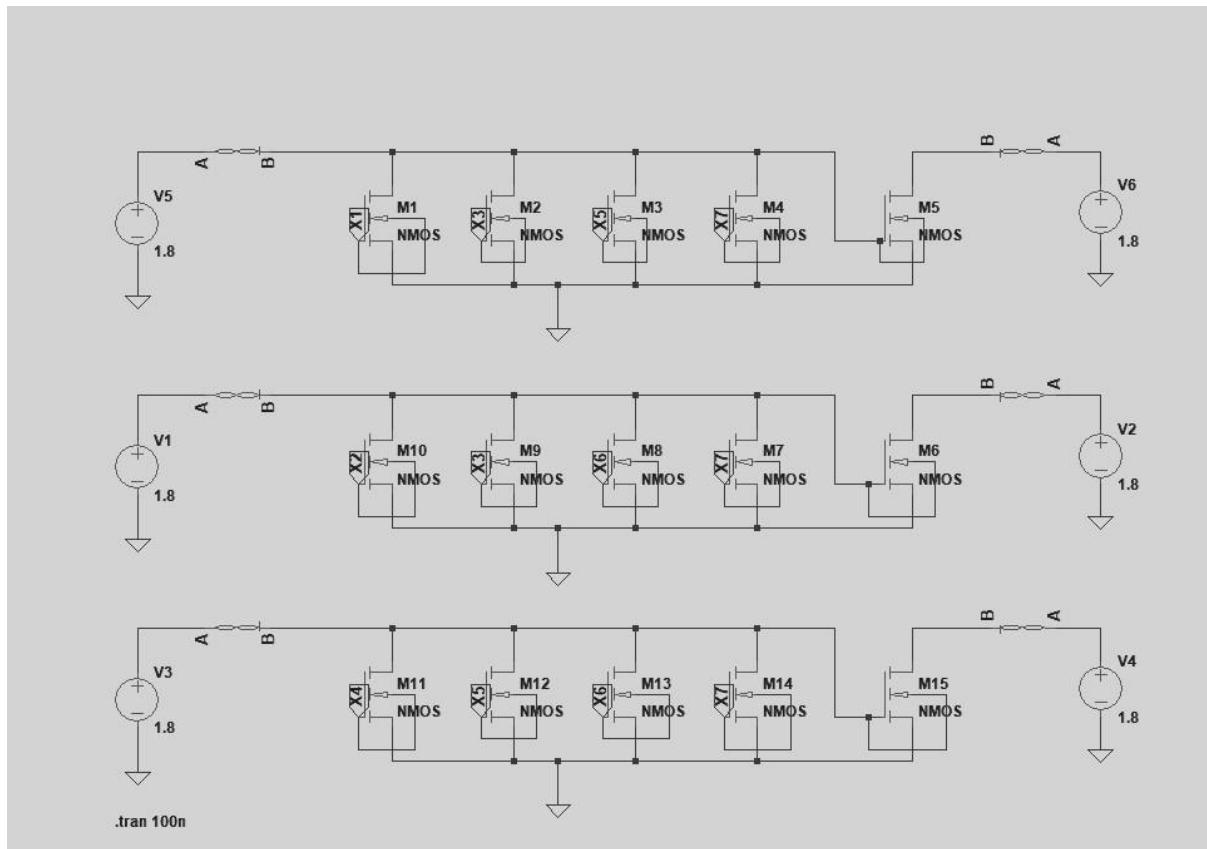


Fig. 3.16 NOR gate using Memristor.

3.5 ENCODER USING MRL TECHNOLOGY

A network of memristors configured in a certain pattern is subjected to a series of binary inputs in a memristor-based encoder. The memristors' resistances vary as current passes through them, producing a particular output voltage that is indicative of the encoded value. In order to improve the network topology and memristor properties, the memristor network design process combines simulation and optimization approaches, which is essential to obtaining accurate and dependable encoding. Nevertheless, memristor-based encoders come with certain drawbacks. The scarcity of memristor devices—which are still relatively new and not yet extensively available—is one of the primary obstacles. Furthermore, designing memristor-based encoders can be more difficult than designing standard encoders due to the non-linear behavior of memristors. Given that a memristor-based design takes up less space than a CMOS design, it can be said that this design is efficient in terms of area used. It is evident that the minimal amount of transistors needed for this design is


Fig 3.17 Encoder using memristor based logic

in contrast to Pseudo NMOS and traditional CMOS logic. The encoder circuit, which uses CMOS technology, consists of 30 transistors—15 PMOS and 15 NMOS. Six PMOS and fifteen NMOS transistors make up the encoder circuit's 21 transistors, which use pseudo-NMOS technology. The encoder features 21 transistors employing MRL technology, comprising 6 memristors and 15 NMOS.

3.6 POWER GATING LOGIC

One circuit in the power gating mechanism operates in two modes. The sleep transistors are active when they are turned on, and they can be thought of as functional redundant resistances. To lessen leakage power, the sleep transistors are turned off while the system is in the sleep mode. A sleep transistor is referred to as a "header switch" when it is situated at VDD and as a "footer switch" when it is put close to the ground. We employed the header power gating technique in this project.

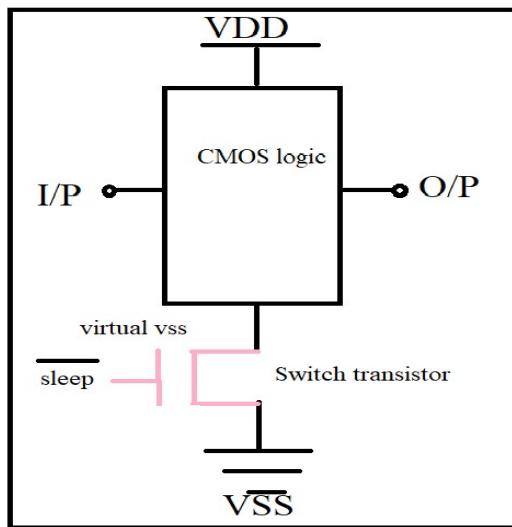


Fig 3.18: Block Diagram of Power Gating Approach

With the process scaling down to 65 nm and lower, power has become a fundamental design concern for current VLSI design. Leakage power has now arisen as a significant part of the total power dissipation in the CMOS circuits. [1]. As illustrated in fig., sleep transistors (ST) are employed as switches to cut off the power sources of the standby circuits. A high V_{th} transistor, known as a ST, can be either PMOS or NMOS and is used to link a permanent power source to a circuit power supply, often known as a "virtual power supply. The PMOS ST, often known as a "header switch," is used to switch the VDD supply. Hence, the NMOS ST Controls VSS supply is also known as a "footer switch." Only header or footer switches are utilized in sub-90nm designs because of the limitations imposed by the sub-1V power supply voltage and the STs' space penalty.

3.7 MEMRISTOR BASED ENCODER USING PGL

Clock gating has less of an impact on design architecture than power gating. Because power gated modes must be safely entered and departed, it causes longer delays. There are trade-offs in architecture between the energy dissipation required to enter and exit low power modes and the amount of leakage power savings. It is possible to shut down the blocks using hardware or software. The power down procedures can be scheduled by driver software. Hardware timers are also an option. An additional choice is a specialized power management controller.

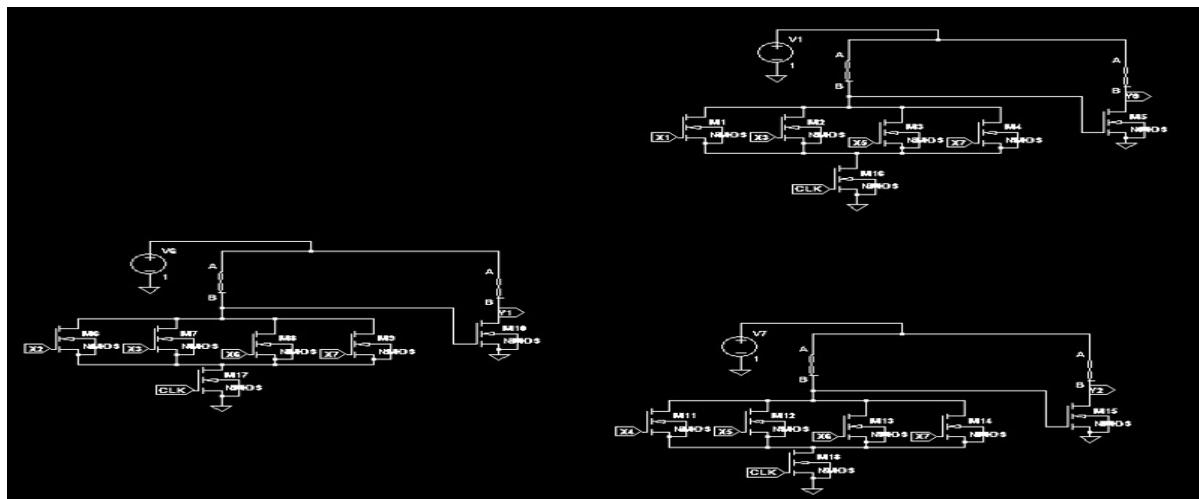


Fig 3.19 Memristor based Encoder using PGL technique

One of the simplest methods of power gating for long-term leakage power reduction is an externally switched power supply. Internal power gating is a better option for briefly turning off the block. Power gating controllers regulate the CMOS switches that supply power to the circuitry. The power gated block's outputs gradually drain. As a result, output voltage levels stay over the threshold voltage level for longer. A greater short circuit current may result from this.

Low-leakage PMOS transistors are used in power gating as header switches to cut off power to components of a design that are in standby or sleep mode. Sleep transistors can also be made from NMOS footer switches. By inserting the sleep transistors, the chip's power network is divided into two sections: a virtual power network that powers the cells and is switchable, and a permanent power network that is connected to the power source. Memristor-based encoder with three outputs and eight inputs that uses PGL. The block diagram is made up of four PMOS transistors, one memristor, and one input side memristor that work together to produce the output Y0. One additional sleep transistor is also attached to the circuit at the output side. To get one output Y0 totally 5 PMOS transistors and 2 memristors and one sleep transistor is required. Similarly, to get the second output Y1 it requires totally 5 PMOS transistors and 2 memristors and one sleep transistor. Similarly, to get the output Y2 again it requires 5 PMOS transistors and 2 memristors and one sleep transistor is needed.

WORKING

Using the PGL technique, a memristor-based encoder is designed. Input is supplied to PMOS transistors, which generate a NOR operation on the inputs to obtain the original

output. The output is then sent to another inverter circuit, which combines PMOS and memristor. To obtain the Y0, ORed together by X1, X3, X5, and X7, output $Y_0=X_1+X_3+X_5+X_7$. to carry out an OR procedure The NOR operation is performed on four PMOS transistors connected in series to obtain the Y0 output, which is then supplied to the inverter, which is made up of one PMOS transistor and a memristor. To cut down on power usage and speed up performance, an additional sleep transistor is linked to the input side. Every time electricity is used while the circuit is switched off, a sleep transistor blocks. To obtain the Y1, X2, X3, X6, X7 are ORed together in the output $Y_1=X_2+X_3+X_6+X_7$. Four PMOS transistors connected in series are used to execute an OR operation. A NOR operation is then performed to obtain the Y0 output of these four PMOS transistors, which is then supplied to an inverter, which is made up of one PMOS transistor and a memristor. In order to decrease power consumption and speed up performance, an additional sleep transistor is connected to the input side. Every time electricity is used when the circuit is switched off, a sleep transistor blocks. The output Y2 is obtained by ORing together X4, X5, X6, and X7. to carry out an OR procedure The NOR operation is performed on four PMOS transistors connected in series to obtain the Y0 output, which is then supplied to the inverter, which is made up of one PMOS transistor and a memristor. To boost performance and lower power consumption, an additional sleep transistor is added to the input side. Every time electricity is used while the circuit is switched off, a sleep transistor blocks.

3.8 Implementation in Cadence

Procedure

Cadence software operates only in UNIX/Linux server. To start off login to your Linux server which has the cadence license key.

Creating a directory

Now create a folder with appropriate name all the project logs will be saved into this folder.

Initiating cadence

After creating the folder open the folder, open this folder in terminal with the help of

options available from the right click on the empty space.

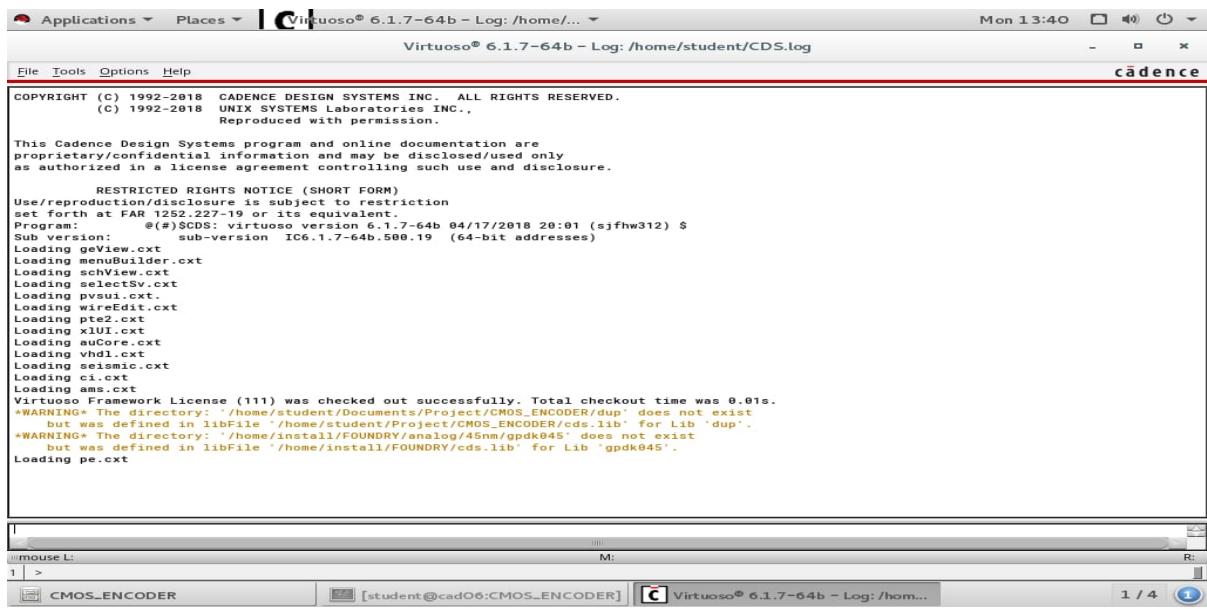
Now to initialize the cadence we have to initiate some commands in the terminal

~csh

~source home/install/cshrc

~virtuoso

Upon successful initialization of these commands the cadence tool will be opened.



The screenshot shows the Virtuoso 6.1.7-64b software interface. The title bar reads "Virtuoso® 6.1.7-64b - Log: /home/student/CDS.log". The main window displays a log message from Cadence Design Systems. The log includes copyright information, a proprietary notice, and a restricted rights notice. It then lists the loading of various configuration files (geView.cxt, memModeler.cxt, etc.) and a warning about a framework license checkout. The bottom status bar shows the project name "CMOS_ENCODER" and the log file path "student@cad06:CMOS_ENCODER".

```
COPYRIGHT (C) 1992-2018 CADENCE DESIGN SYSTEMS INC. ALL RIGHTS RESERVED.  
(C) 1992-2018 UNIX SYSTEMS Laboratories INC.  
Reproduced with permission.  
  
This Cadence Design Systems program and online documentation are  
proprietary/confidential information and may be disclosed/used only  
as authorized in a license agreement controlling such use and disclosure.  
  
RESTRICTED RIGHTS NOTICE (SHORT FORM)  
Use/reproduction/disclosure is subject to restriction  
set forth at FAR 1252.227-19 or its equivalent.  
Program: @(#)SCDS: virtuoso version 6.1.7-64b 04/17/2018 20:01 (sjfhw312) $  
Sub version: sub-version IC6.1.7-64b.500.19 (64-bit addresses)  
Loading geView.cxt  
Loading memModeler.cxt  
Loading netVisor.cxt  
Loading selectSv.cxt  
Loading pvsui.cxt.  
Loading wireEdit.cxt  
Loading pte2.cxt  
Loading libC.cxt  
Loading auroC.cxt  
Loading vhdl1.cxt  
Loading seismic.cxt  
Loading ci.cxt  
Loading ams.cxt  
Virtual Framework License (111) was checked out successfully. Total checkout time was 0.01s.  
+WARNING+ The directory: '/home/student/Documents/Project/CMOS_ENCODER/dup' does not exist  
but was defined in libFile '/home/student/Project/CMOS_ENCODER/cds.lib' for Lib 'dup'.  
+WARNING+ The directory: '/home/install/FOUNDRY/analog/45nm/gpdk045' does not exist  
but was defined in libFile '/home/install/FOUNDRY/cds.lib' for Lib 'gpdk045'.  
Loading pe.cxt  
  
mouse L: M: R:  
CMOS_ENCODER [student@cad06:CMOS_ENCODER] Virtuoso® 6.1.7-64b - Log: /hom... 1 / 4
```

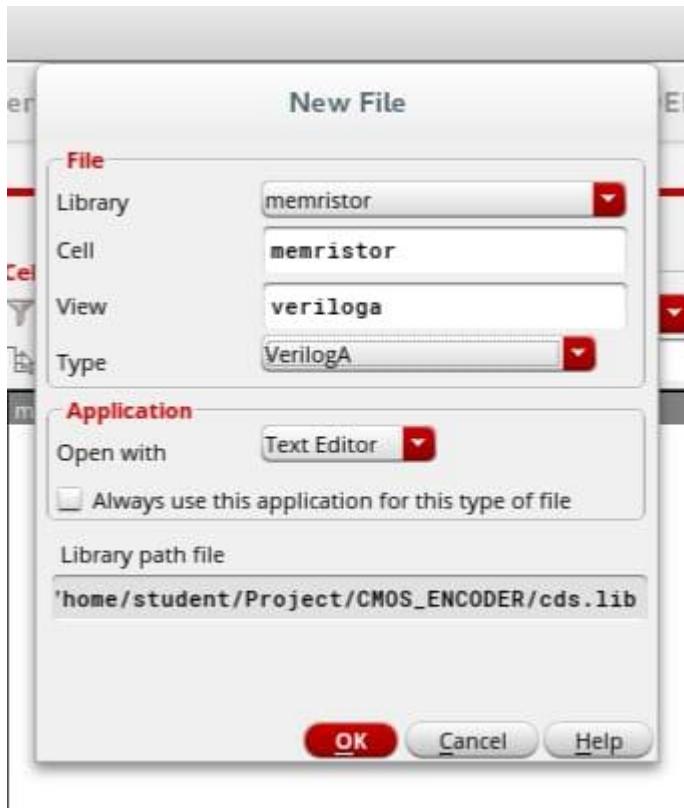
In order to setup the environment for this project we have all the components needed in the standard library i.e., gpdk90 except for our major component memristor.

Procedure for memristor

Creating new Library

From the top left menu select FILE -> LIBRARY -> CREATE NEW LIBRARY.

Now here we can give the properties for the library. Since this library will be used to initiate memristor we are going to select DO NOT NEED PROCESS INFORMATION and select type as **Verilog a**.



This library now allows us to code in Verilog A in order produce schematic.

SOURCE FOR MEMRISTOR

To initialize the memristor we referred to external source for memristor code and we came across VTEAM memristor code. And implemented onto our design. The memristor code is as follows

```

IV. VERILOG-A CODE

/////////////////////////////
// VerilogA model for memristor
// kerentalis@gmail.com
// Dimafilter@gmail.com
// skv@tx.technion.ac.il
// Technion - Israel institute of technology
// EE Dept. December 2011
/////////////////////////////

`include "disciplines.vams"
`include "constants.h"

// define meter units for w parameter
nature distance;
access = Metr;
units = "m";
abatol = 0.01n;
endnature

discipline Distance
    potential distance;
enddiscipline

module Memristor(p, n,w_position);
    input p;//positive pin
    output n;//negative pin
    output w_position;// w-width pin
    electrical p, n,gnd;
    Distance w_position;
    ground gnd;

    parameter real model = 0;
    // define the model:
    // 0 - Linear Ion Drift;
    // 1 - Simmons Tunnel Barrier;
    // 2 - Team model;
    // 3 - Nonlinear Ion Drift model
    parameter real window_type=0;
    // define the window type:
    // 0 - No window;
    // 1 - Joglekar window;
    // 2 - Bielek window;
    // 3 - Prodromakis window;
    // 4 - Kvatincky window (Team model only)
    parameter real dt=0;
    // user must specify dt same as max step size in
    // transient analysis & must be at least 3 orders
    // provoke the w width not to get stuck at
    // 0 or D with p window
    parameter real threshold_voltage=0;
    // local variables
    real w;
    real dwt;
    real dlast;
    real Rj;
    real sign_multiply;
    real stp_multiply;
    real first_iteration;

////////////////// Simmons Tunnel Barrier model ///////////////////
//parameters definitions and default values
//for Simmons Tunnel Barrier model
parameter real c_off = 3.5e-6;
parameter real c_on = 40e-6;
parameter real l_coff = 150e-6;
parameter real l_con = 8.4e-6;
parameter real x_c = 107e-12;
parameter real b = 500e-6;
parameter real a_on = 2e-9;
parameter real a_off = 1.2e-9;

// local variables
real x;
real dzt;
real x_last;

//////////////////TEAM model/////////////////
parameter real K_on=-8e-13;
parameter real K_off=-8e-13;
parameter real Alpha_on=3;
parameter real Alpha_off=12;
parameter real IV_relation=0;
// IV_relation=0 means linear V=IR,
// IV_relation=1 means nonlinear V=I*exp(..)
parameter real x_on=0;
parameter real x_off=3e-09; // equals D
// local variables
real lambda;

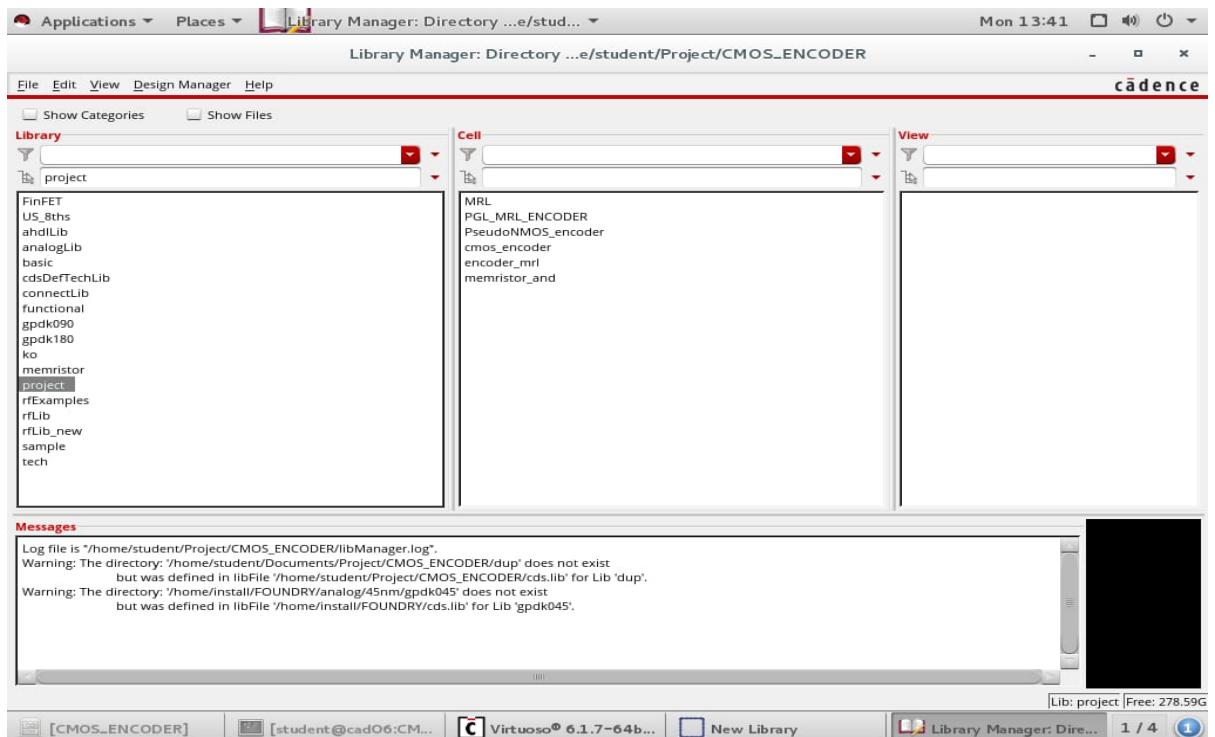
//////////////////Nonlinear Ion Drift model ///////////////////
parameter real alpha = 2;
parameter real beta = 9;
parameter real gamma = 0.01;
parameter real g = 4;

```

The code file that will be available will not be properly organized so it is essential to convert the code with proper indentation and the code is error free.

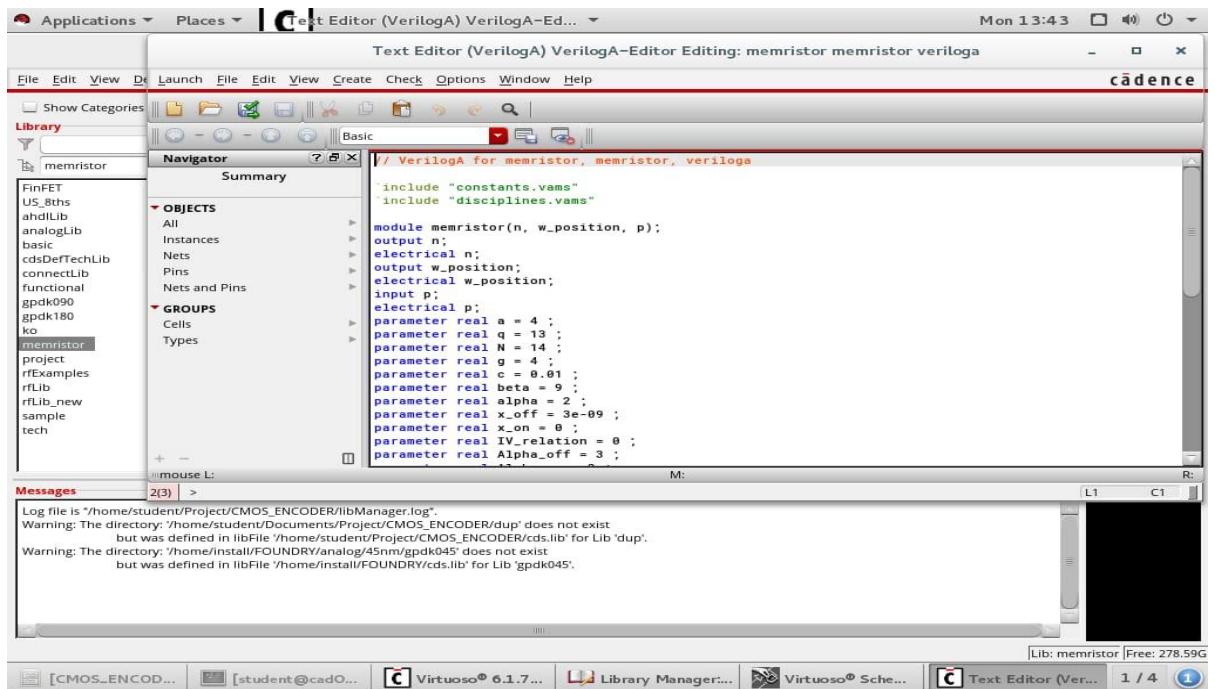
Creating Cell view

After creation of library select FILE->LIBRARY MANAGER from the top left menu. Now we are able to see the library that we have created.



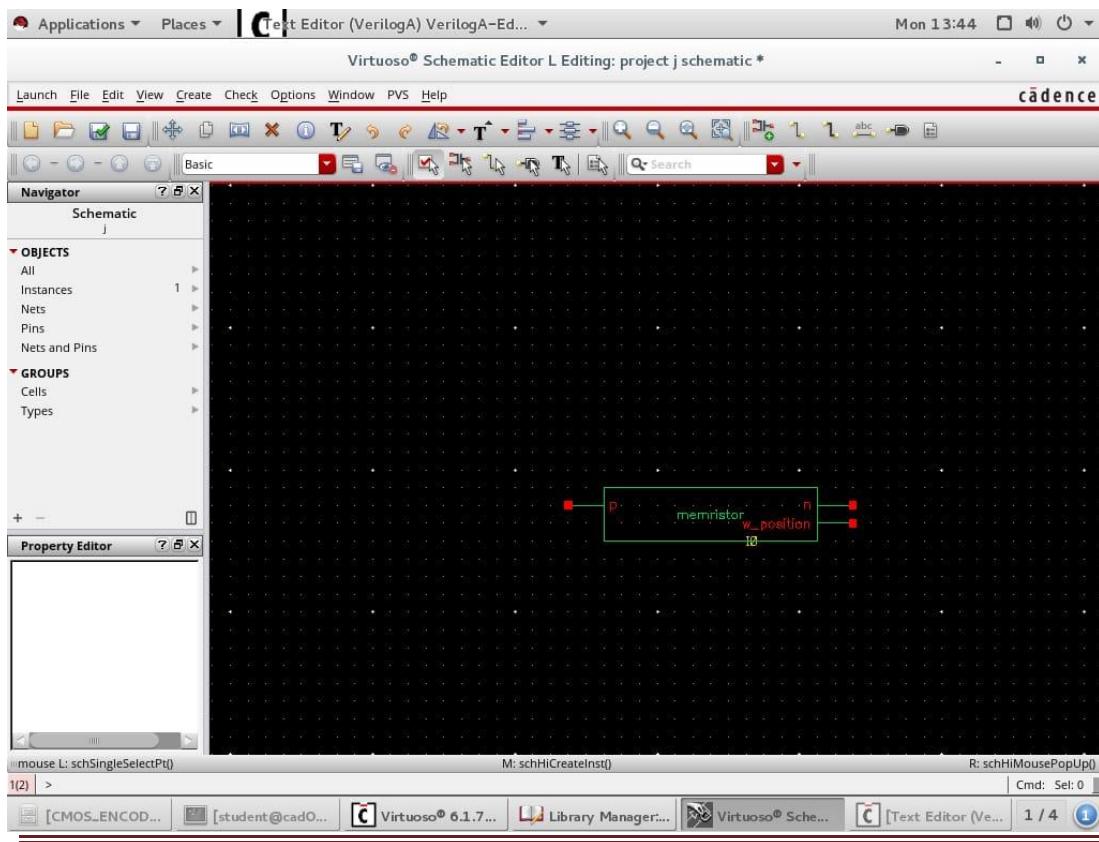
Select that library and from the top menu select FILE->NEW->CELLVIEW. Provide a appropriate name for the cell click OK. Now the cell view to implement Verilog A is created.

Design of memristor based low power encoder using PGL technique



Now copy the Verilog A code into the cell view that has been created. And make sure the indentation and syntax are error free.

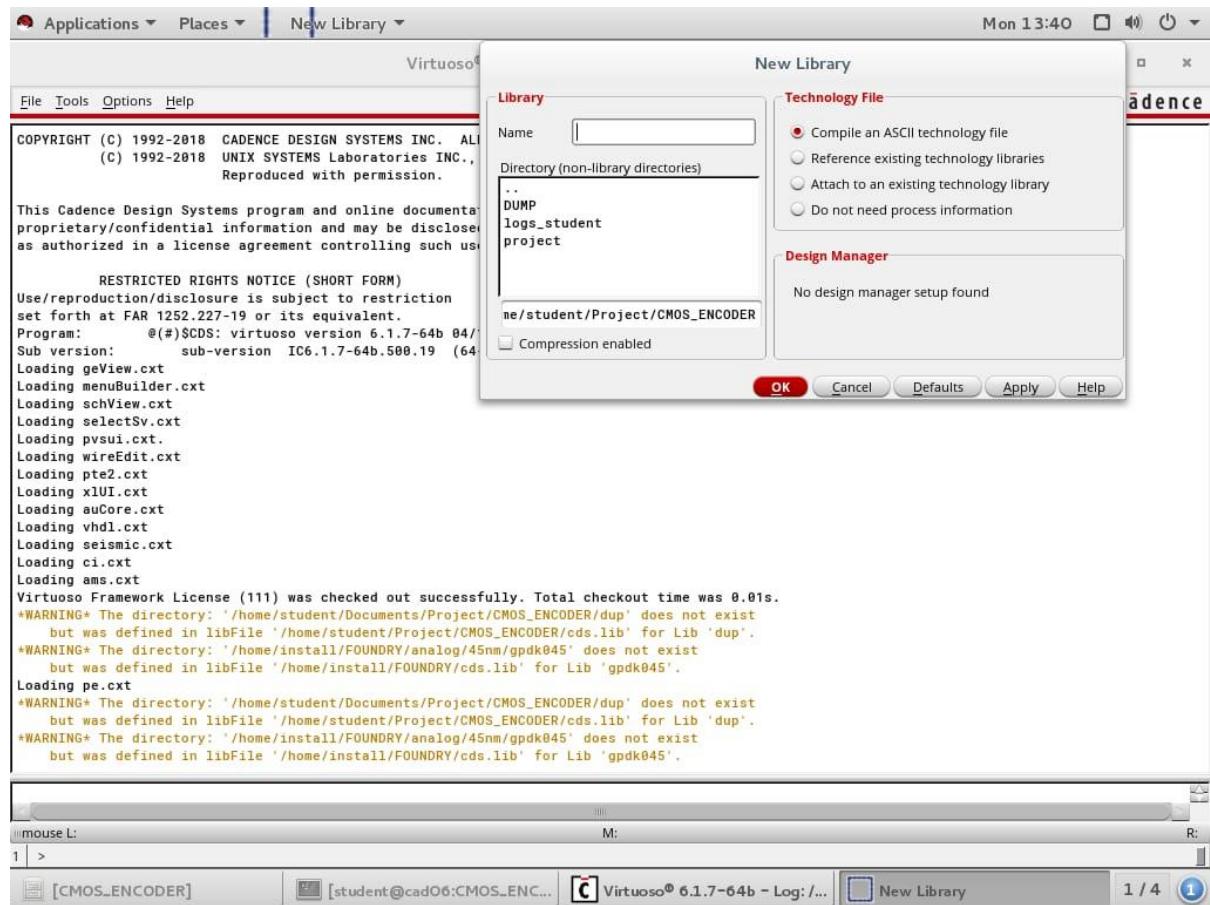
Check and save the cell view from the option available on top. And RUN the cell. Upon successful implementation we are able to see the memristor symbol in an empty cell view.



During the logic implementation the third terminal produces the warning and since there will be no need of it. The warning can be neglected and logic implementation can be done. Now in order to initialize the encoder logical circuit. Let us create another library in which we are going to implement cmos encoder, pseudocmos encoder, memristor based encoder in short MRL and POWER GATING LOGIC implemented on MRL. Our aim is to reduce the power and delay factor of the MRL encoder using the power gating logic.

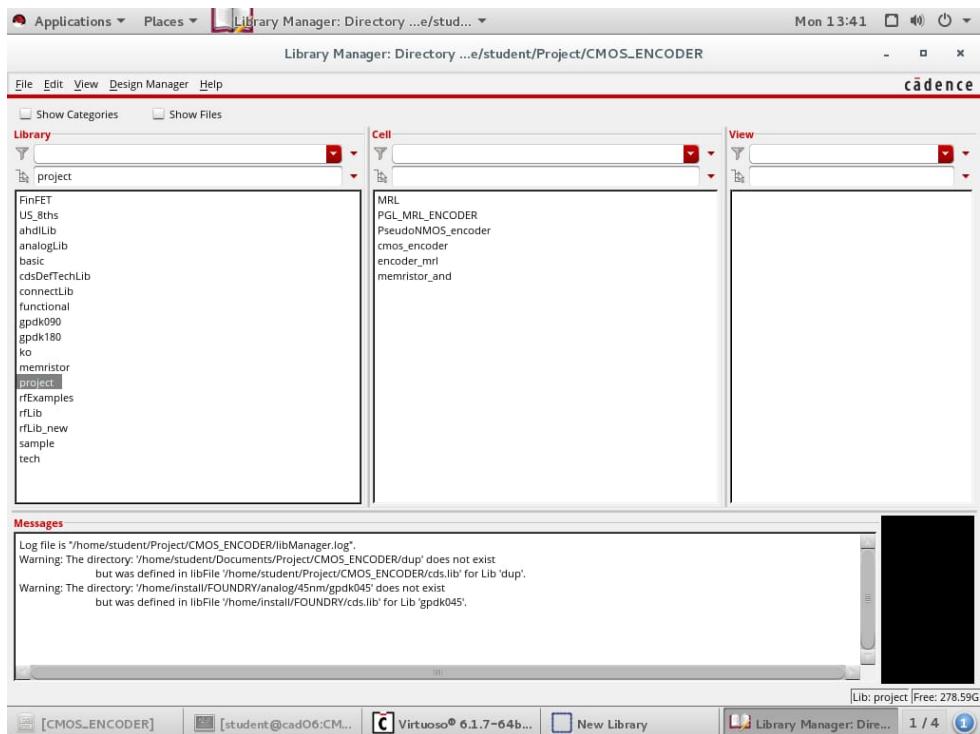
Now in order to create new library we select FILE -> NEW -> LIBRARY.

And in the technology file select ATTACH TO AN EXISTING LIBRARY and select the gpdk90 technology.

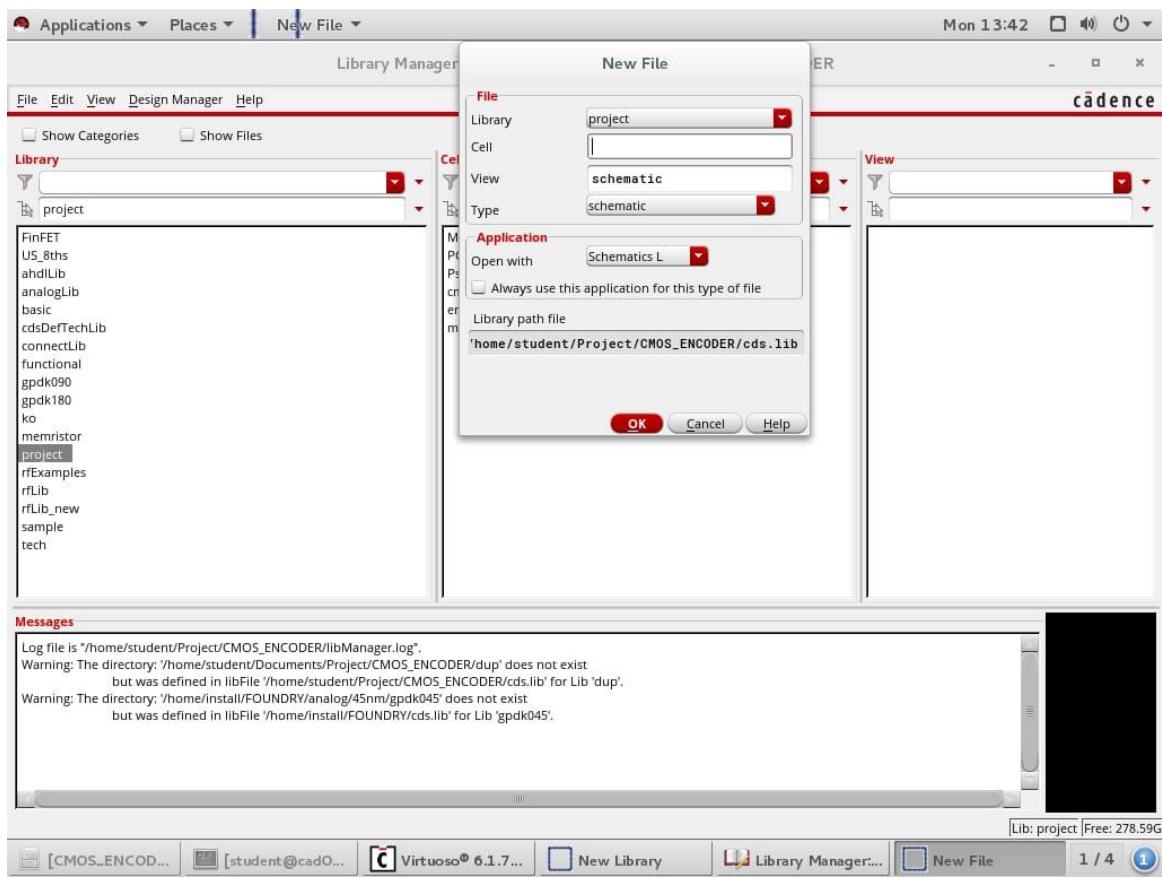


Now select FILE -> LIBRARY MANAGER from the top left menu. And select the created library and select FILE -> NEW -> CELL VIEW from the top left menu.

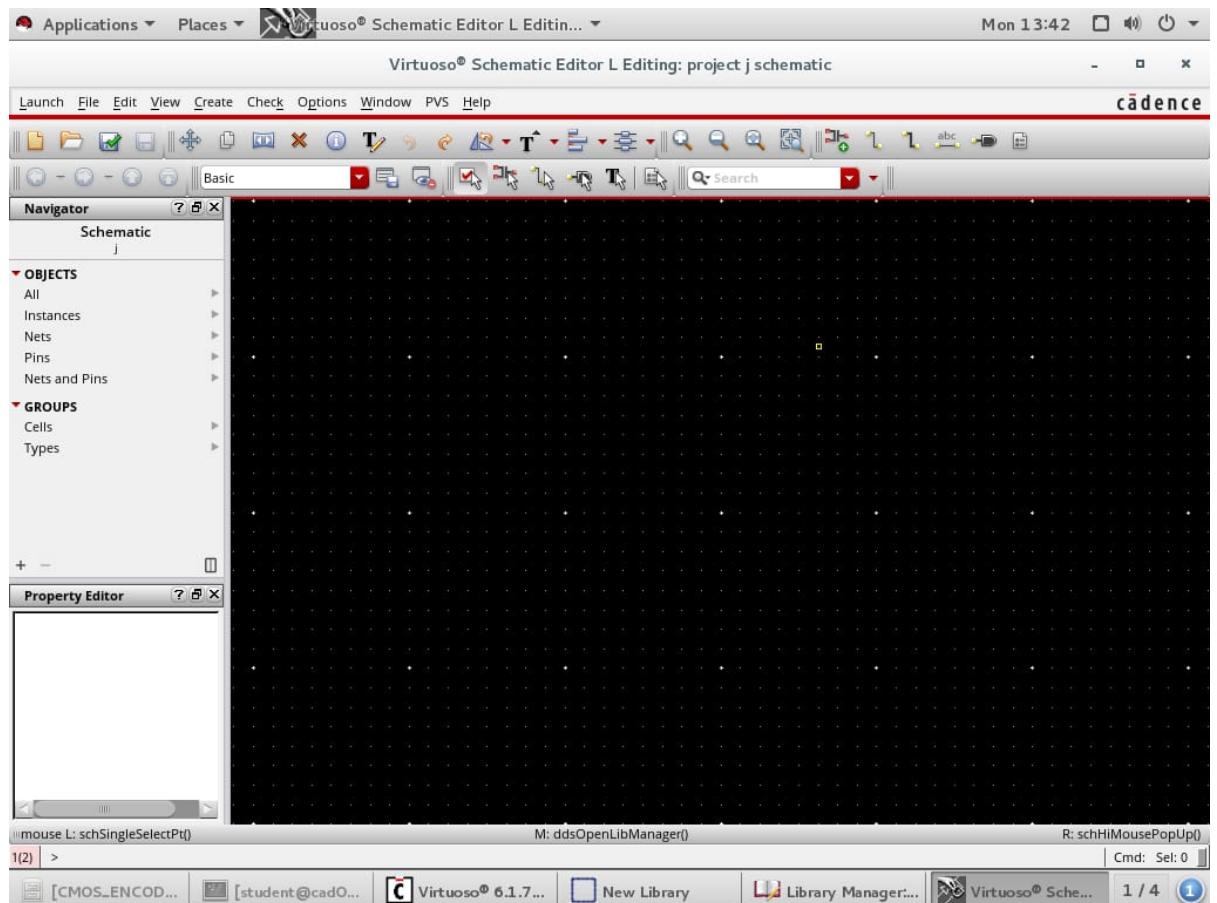
Design of memristor based low power encoder using PGL technique



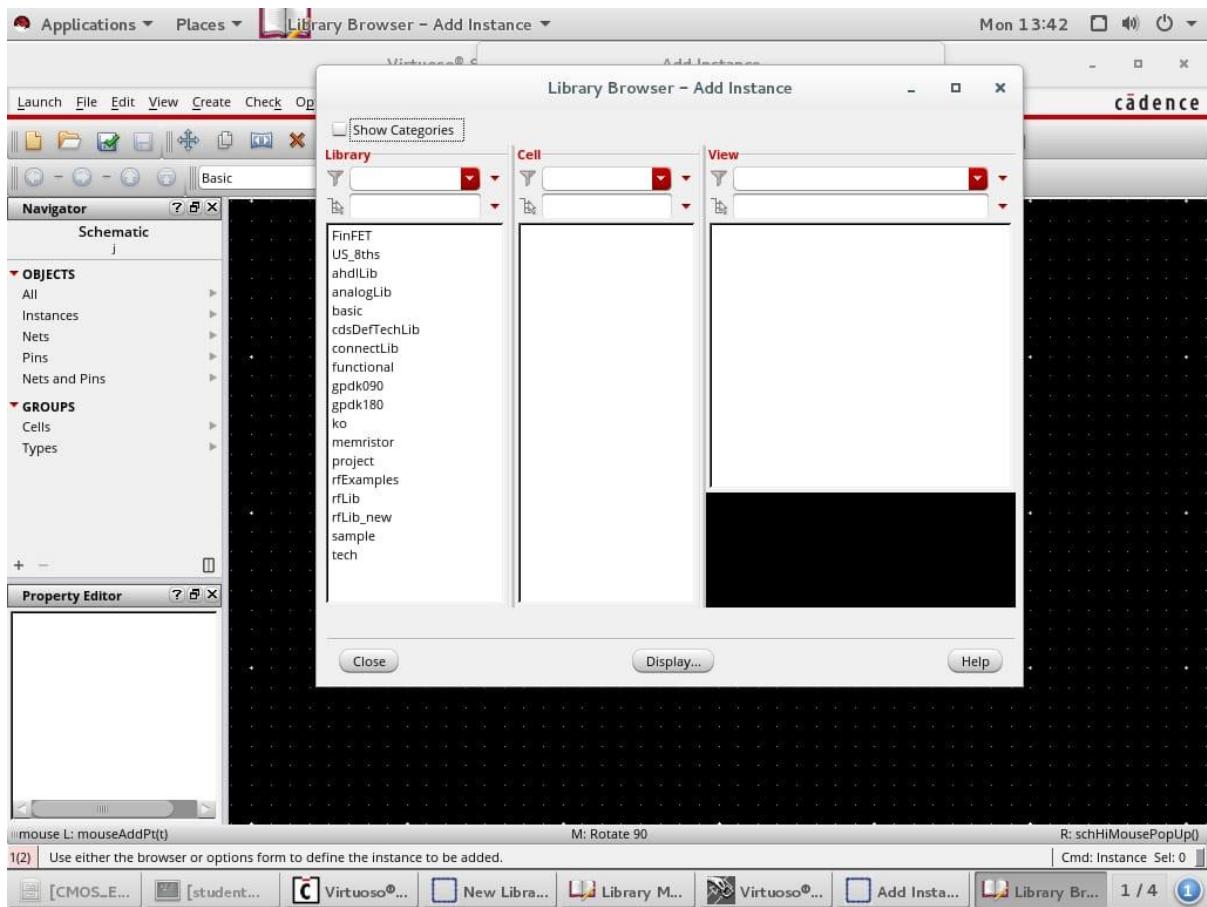
A new pop-up window appears from the window provide the cell an appropriate name and select type as schematic.



Check the path, library and type of the cell view that is being created and click OK button.
Now an empty cell view is created and we can able to perform circuit design in this cell view that is created.



In this cell view we can design our encoder circuit. Now in order to add components onto the cell view we can use the **KEY I** or we can find ADD INSTANCE button at top right of the cell view.



The components required to implement the encoder logic circuit are available in these libraries.

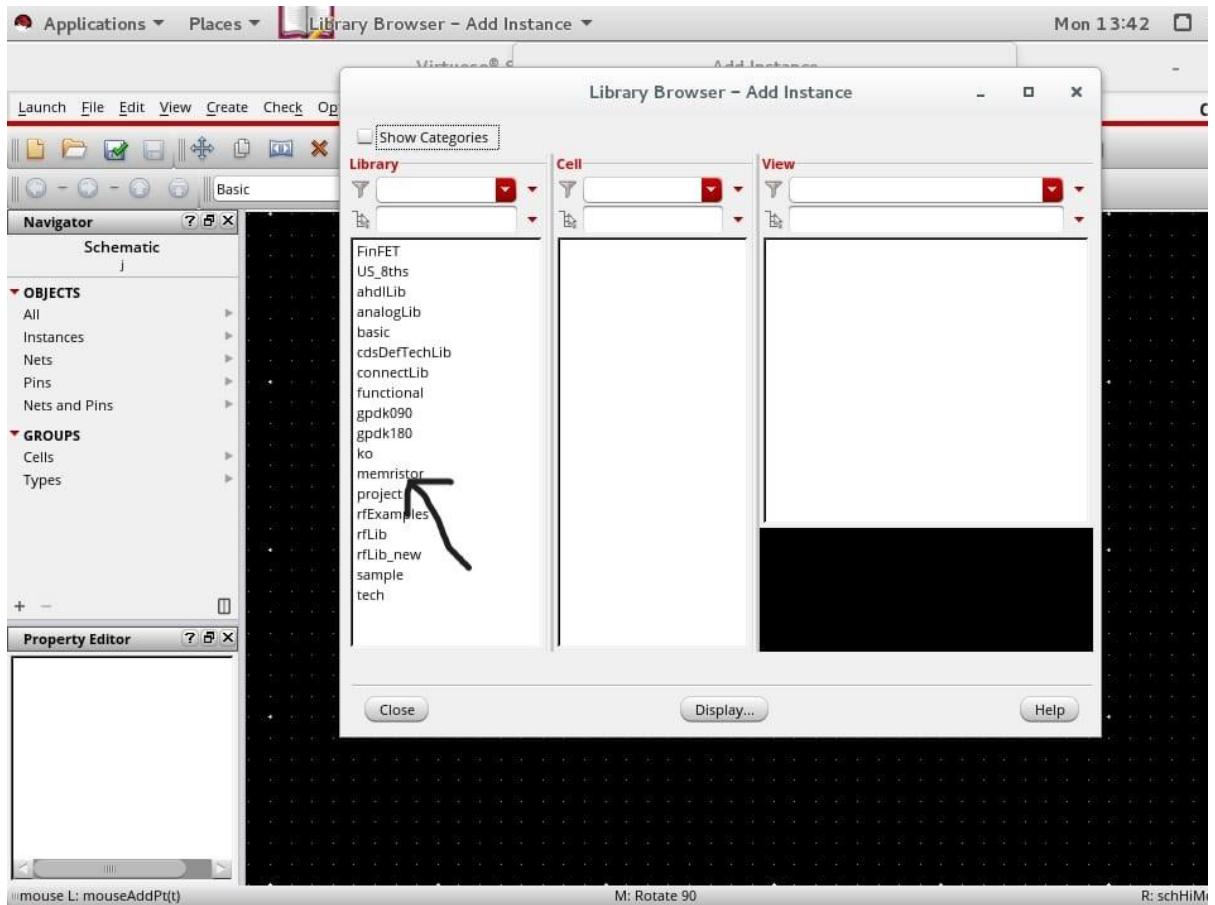
NMOS, PMOS TRANSISTORS ARE AVAILABLE IN THE GPDK090 TECHNOLOGY.

VPWL, GND, VDC ARE AVAILABLE IN THE analogLib TECHNOLOGY FILE.

MEMRISTOR IS CREATED WITH THE HELP OF VERILOG A CODE REFERRED FROM EXTERNAL SOURCE (VTEAM)

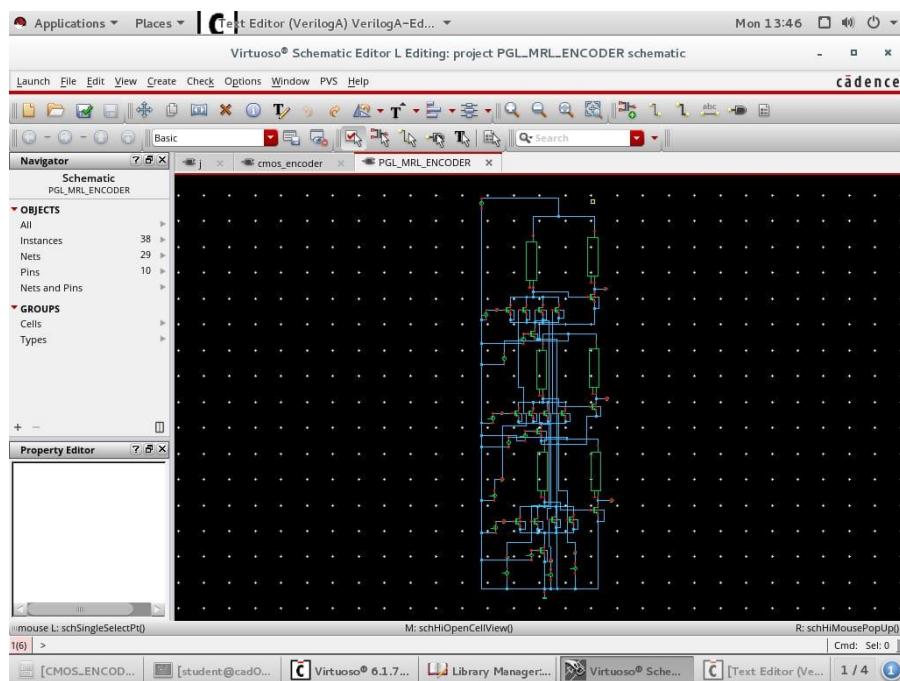
As you can see the memristor library is also included with the standard modules.

Design of memristor based low power encoder using PGL technique



To implement the memristor onto our design we can use the memristor library that we created.

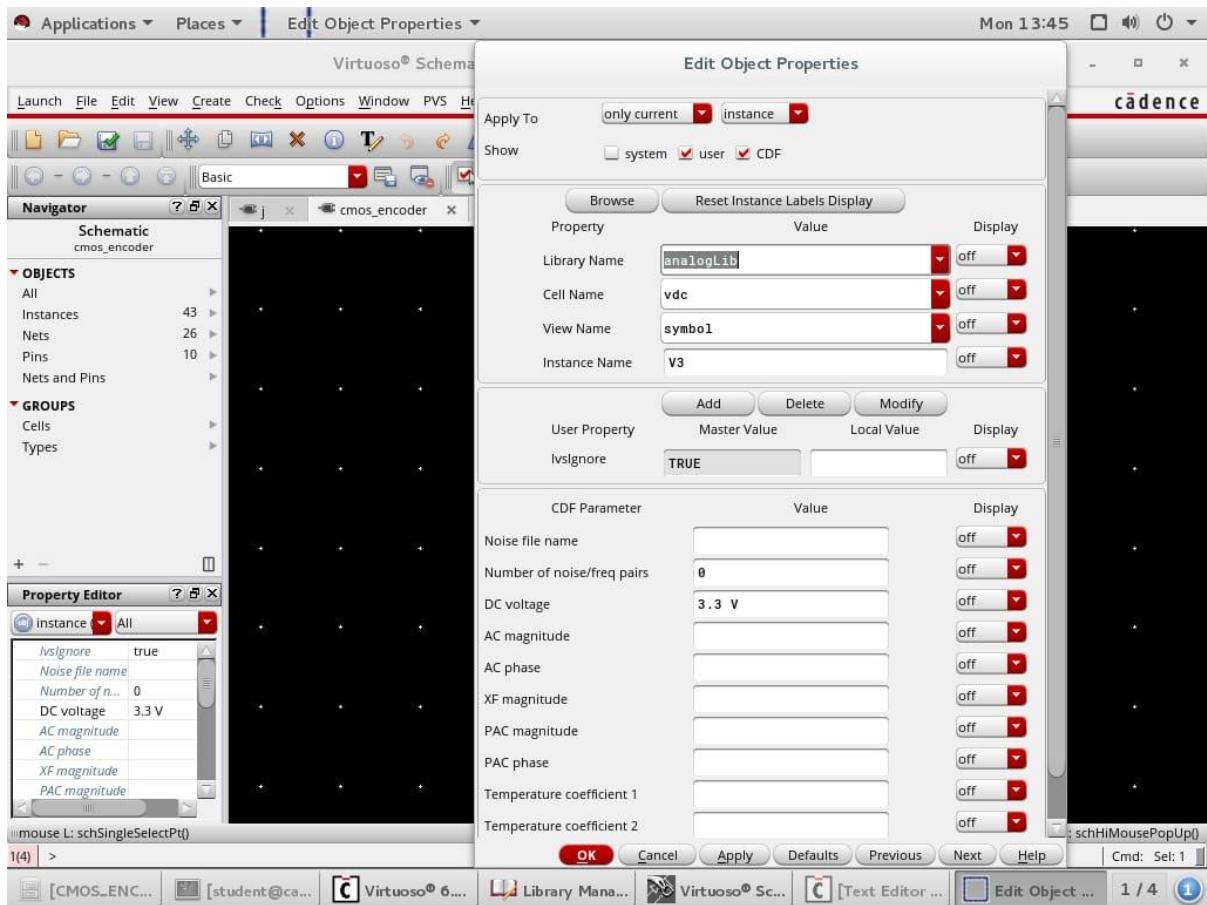
After successful component retrieval to connect the components, we can use the KEY W that stands for wire or we can use the wire option from menu available.



This is the circuit diagram for the power gating logic memristor based encoder circuit.

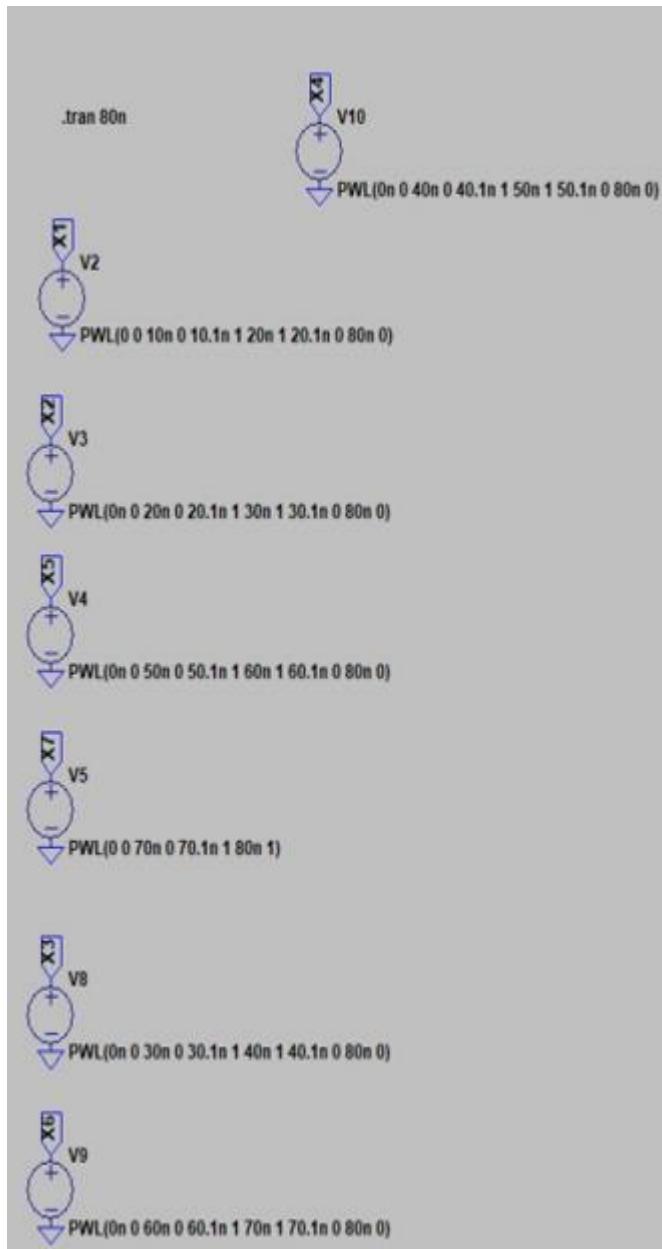
After the circuit diagram to verify the functionality of the circuit we must provide vdc and vpwl logics and provide the pins to identify the inputs and outputs.

In order to provide the parameters select the component and select **KEY Q** to alter the properties for the selected component. Since we are using gpdk090 the VDC voltage can be set to 3.3V. And it can be set using edit properties pop-up after initiating Q key on the required component.

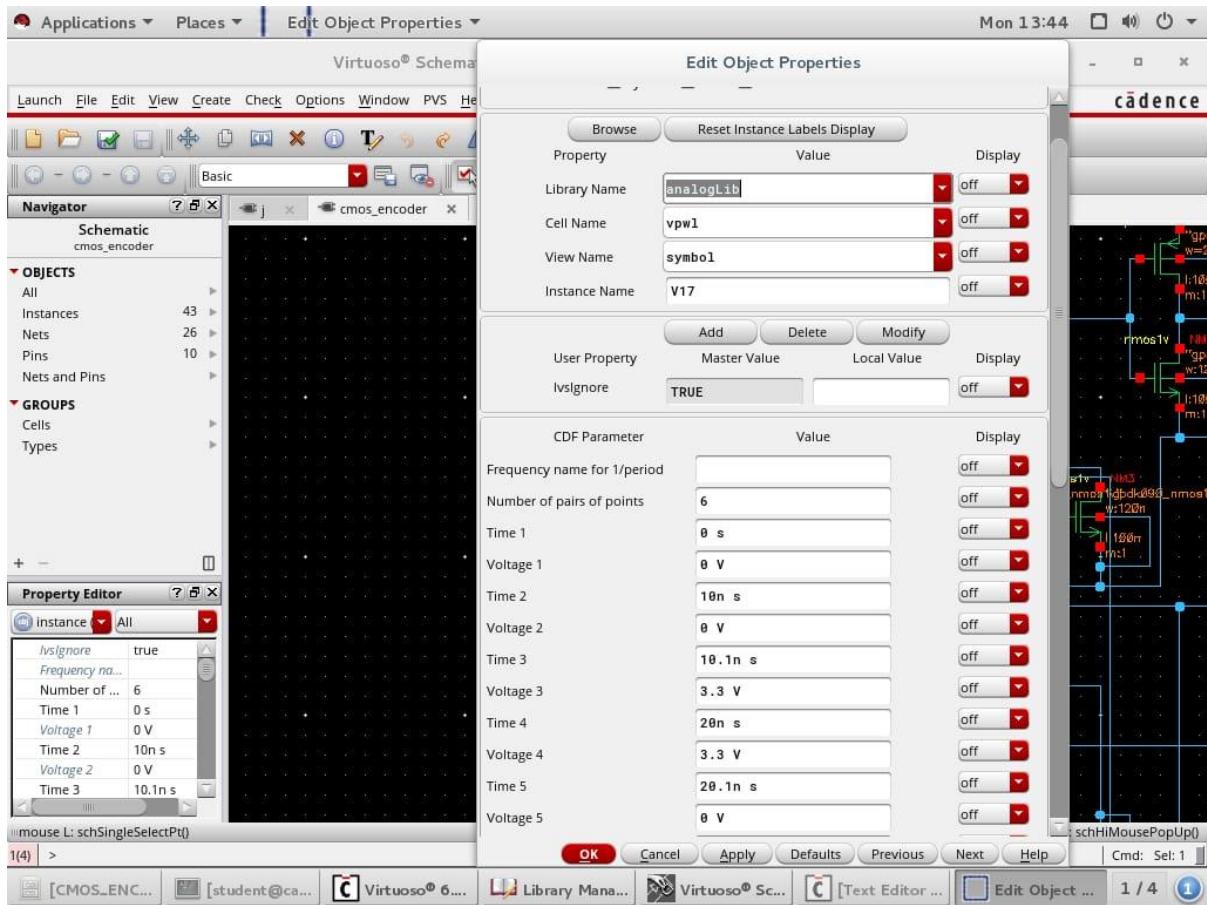


As to test the functionality of the encoder circuit we will be using VPWL component which allows us to provide INPUT HIGH at required time.

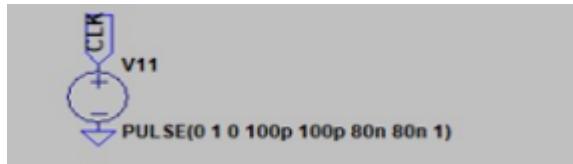
The inputs for the VPWL components is taken as follows:



Design of memristor based low power encoder using PGL technique



The clock input for the power gating logic implementation we will be using VPULSE from the analogLib the properties of the v-pulse will be as follows:



voltage 1 – 0V
rise time – 100p
pulse delay – 80n

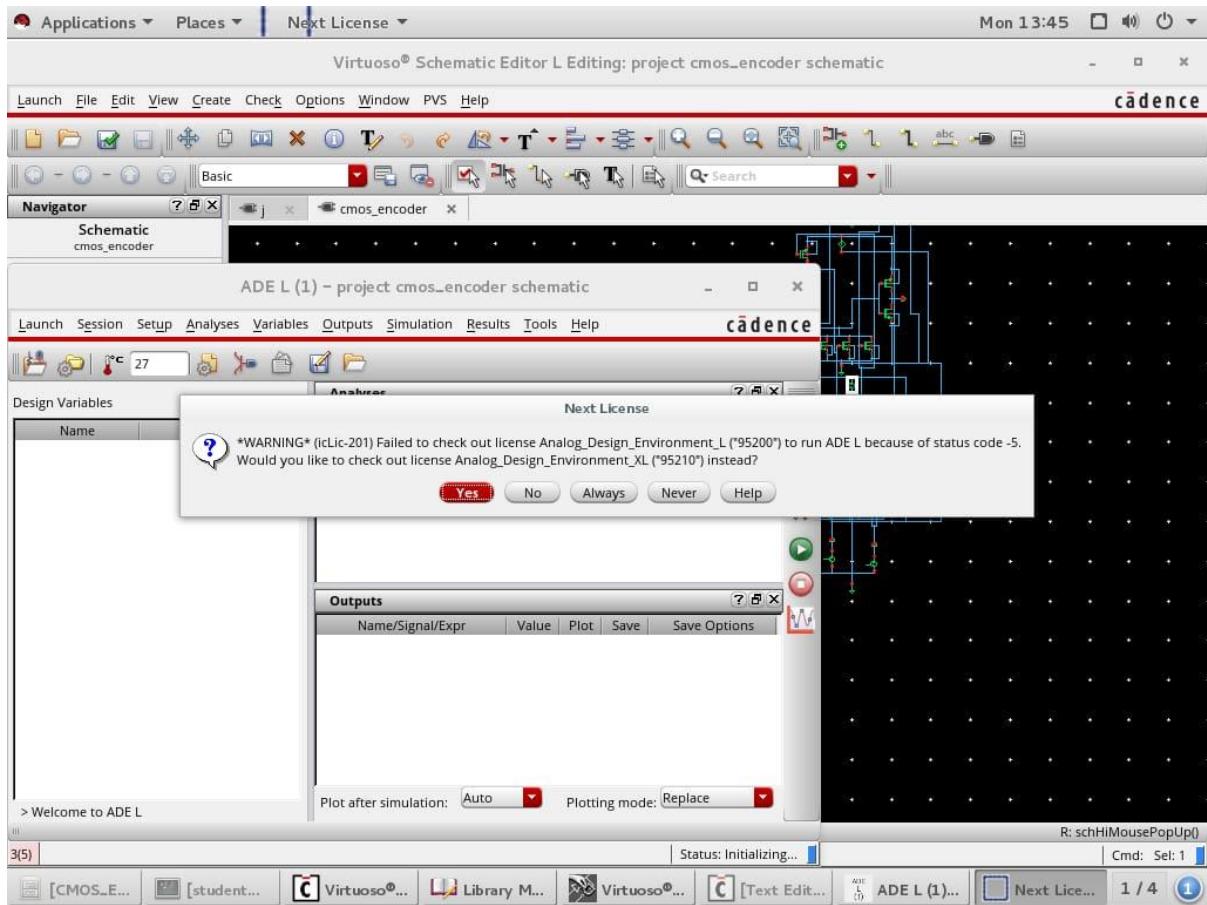
voltage 2 - 1V
fall time – 100p
pulse period – 80n

After editing the properties of the components we save the schematic with the help of CTRL + S key.

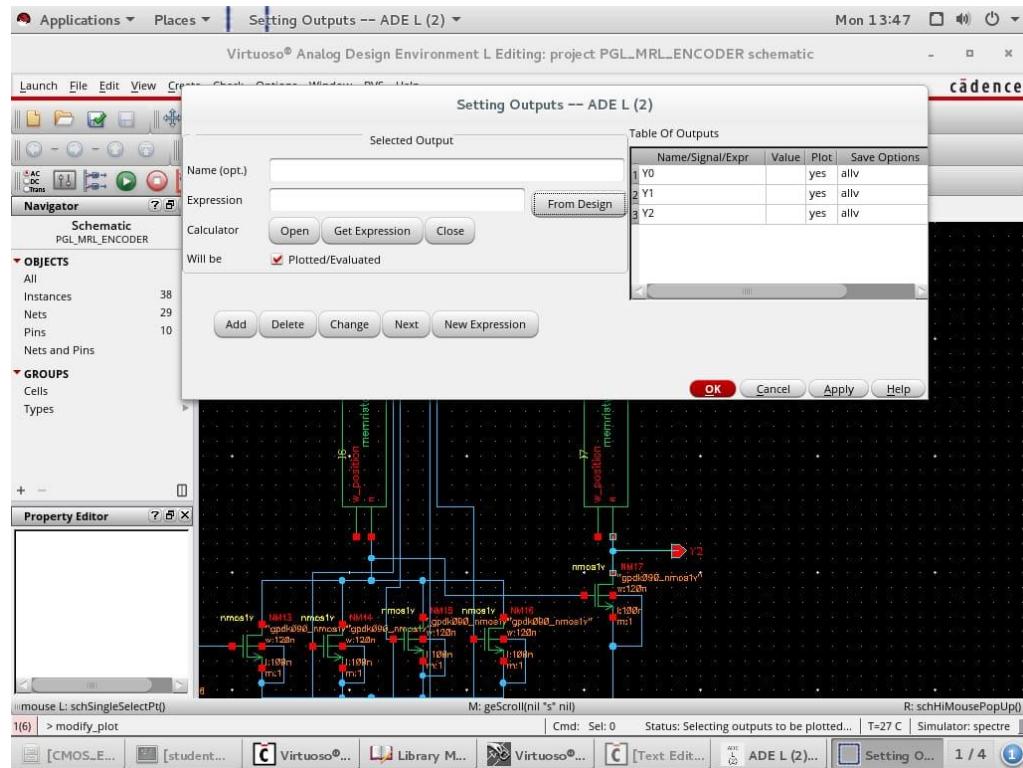
SIMULATING THE RESULTS

After saving the schematic from the top left corner select LAUNCH->ADE L. This will enable us to simulating the results for the circuit we constructed. After launching we get multiple warning ignore and click OK for all of the warnings.

Design of memristor based low power encoder using PGL technique

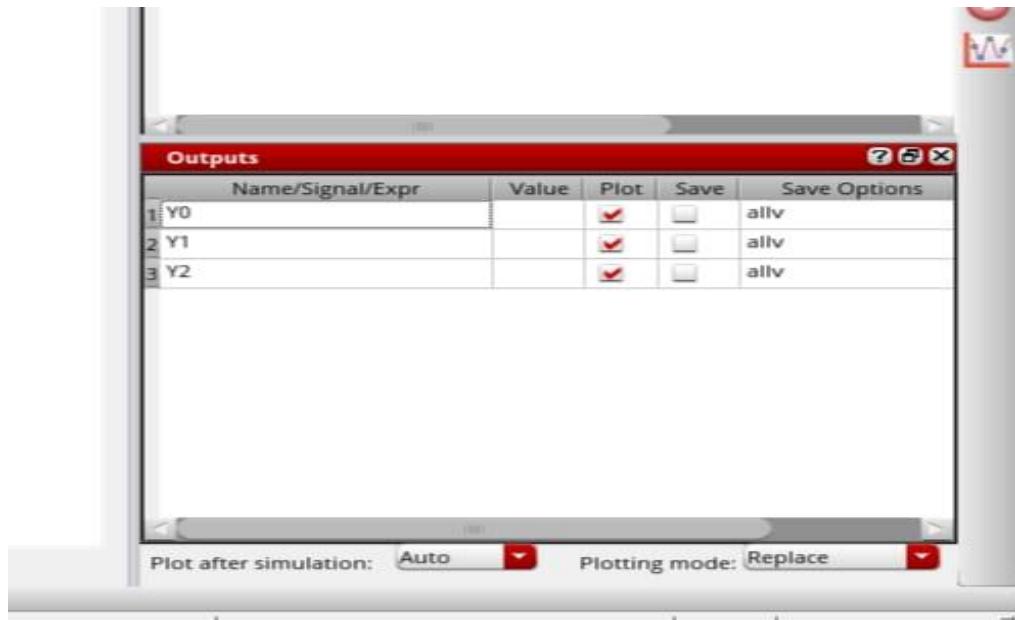


Now in order to plot the output graphs right click on the output sub window in the ADE L simulation and select EDIT. A new pop – window appears as follow:

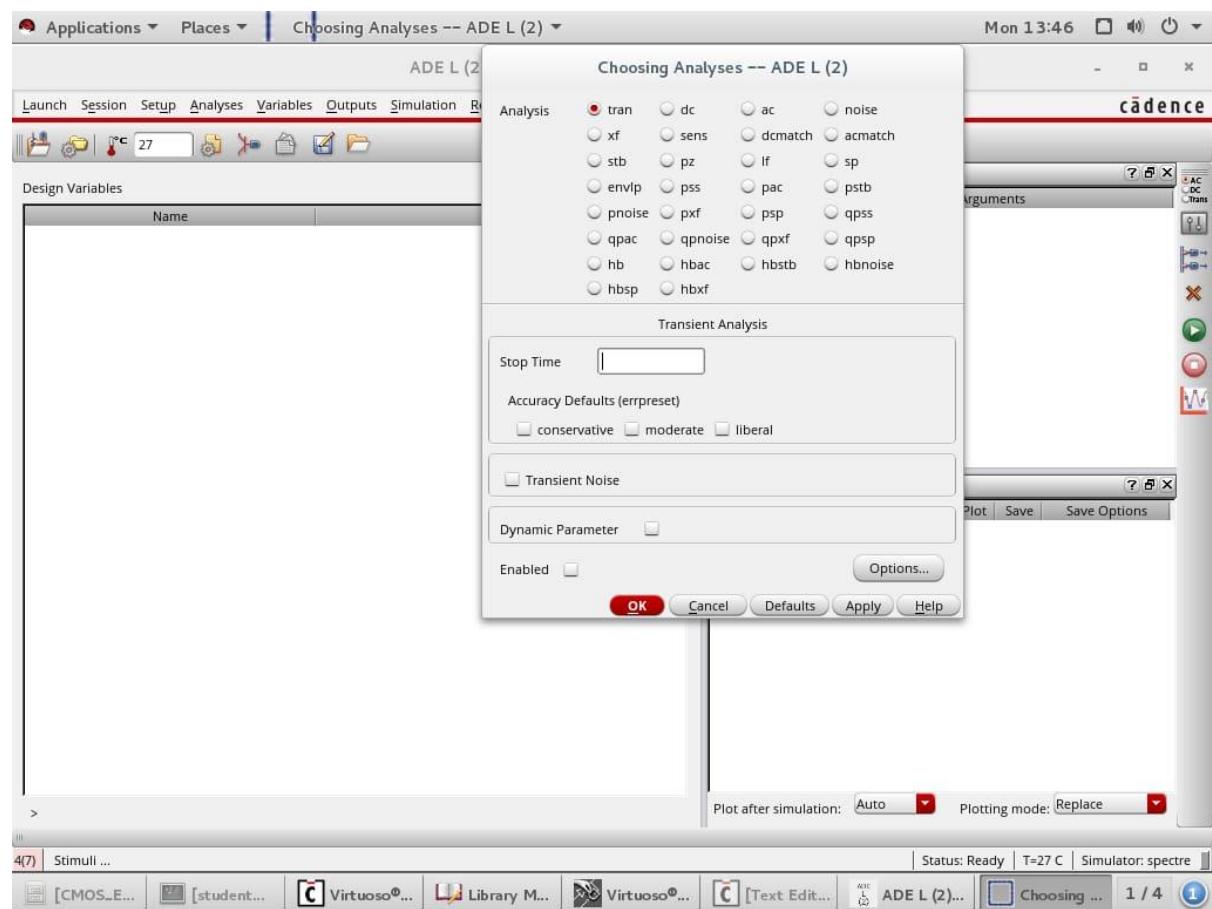


Design of memristor based low power encoder using PGL technique

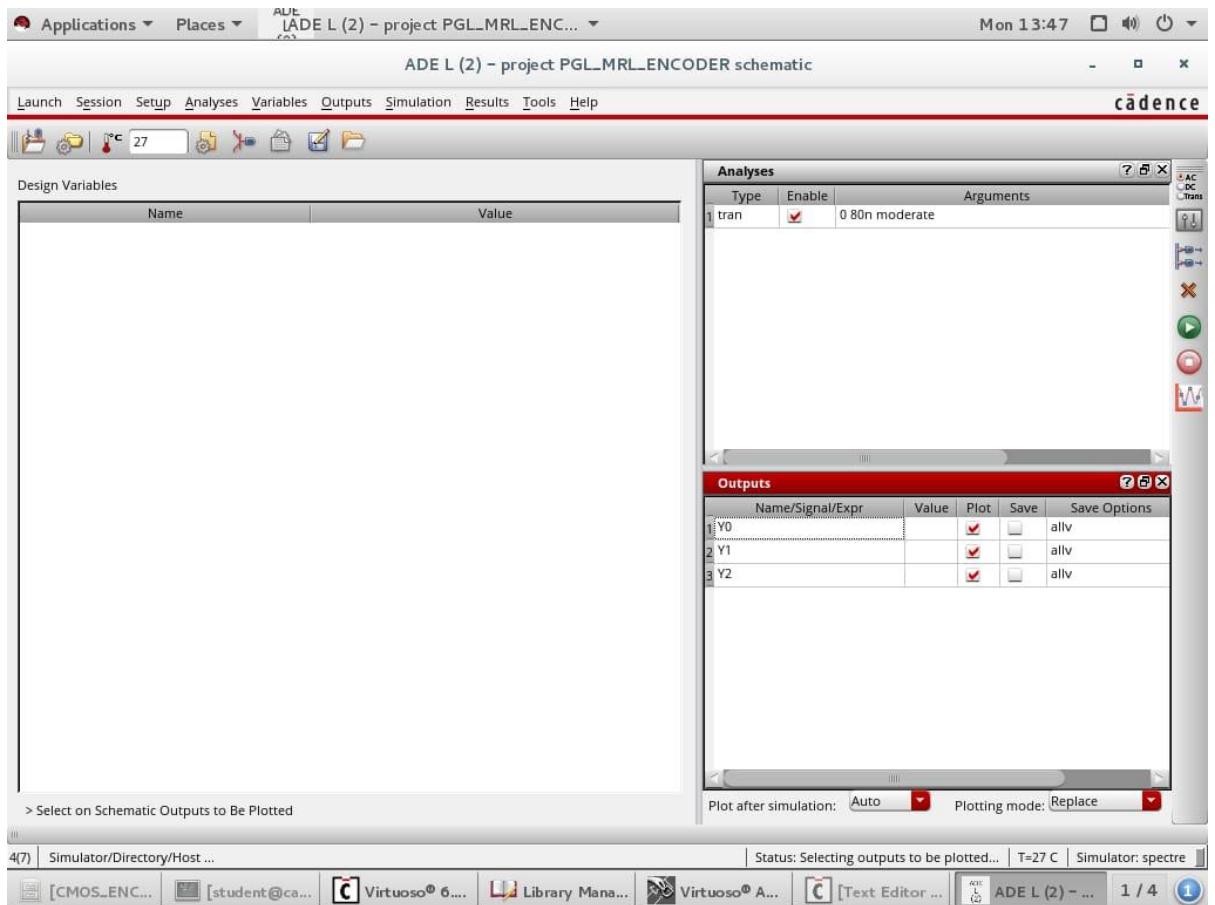
Select ‘FROM DESIGN’ option from the window it enables us to select the pins that needs to be plotted. After selecting all the required pins on schematic click on OK. This will generate the following output window.



Since we are doing transient analysis edit the analysis sub window and select EDIT, a new window appears as follows in this select analysis type as **trans** stop time as 80n s.



Design of memristor based low power encoder using PGL technique



And we have setup our test bench in order verify the logic circuit behavior. And to simulate the resultant waveforms we can click green run button which is at right side of the whole window.

This will produce the waveforms for all the selected pins from 0ns to 80ns time period.

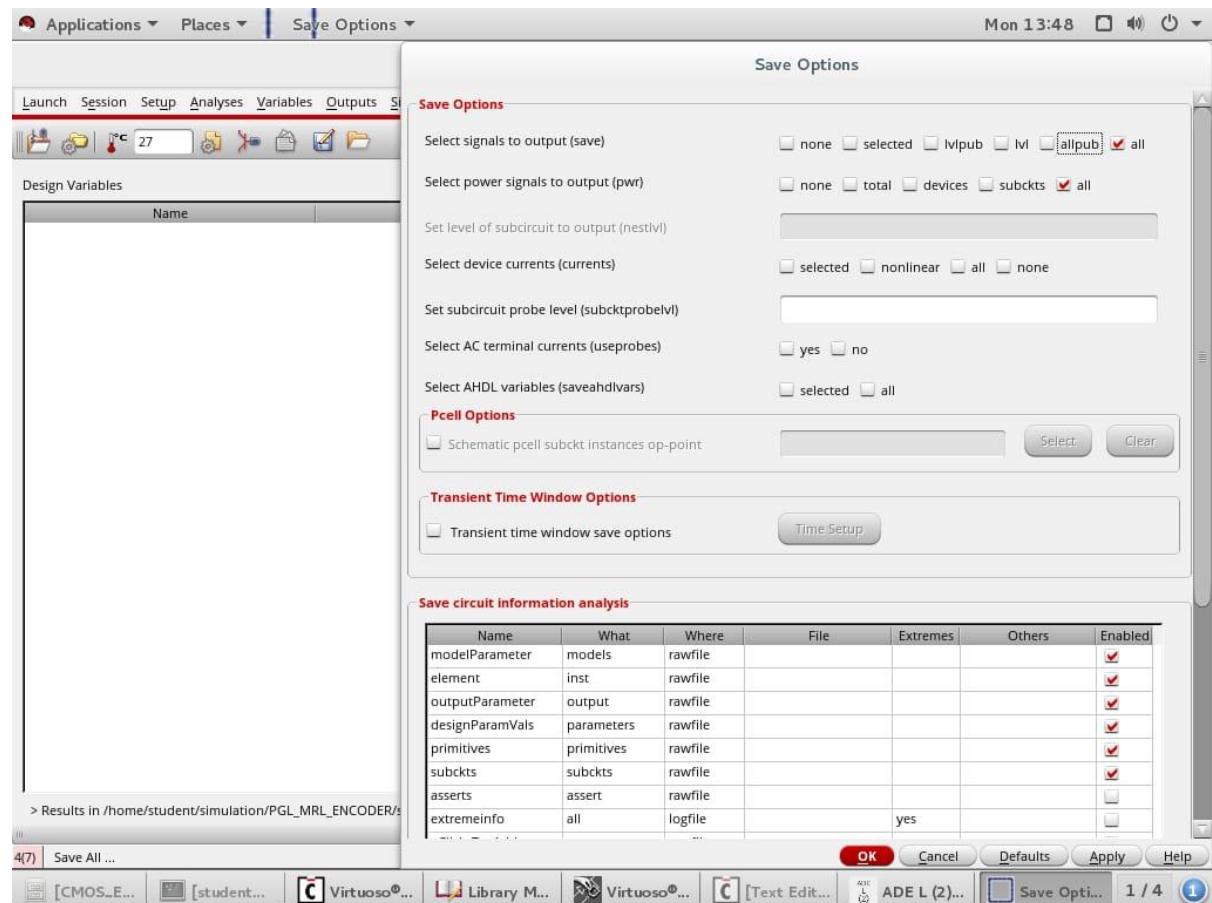


AVERAGE POWER AND DELAY ANALYSIS

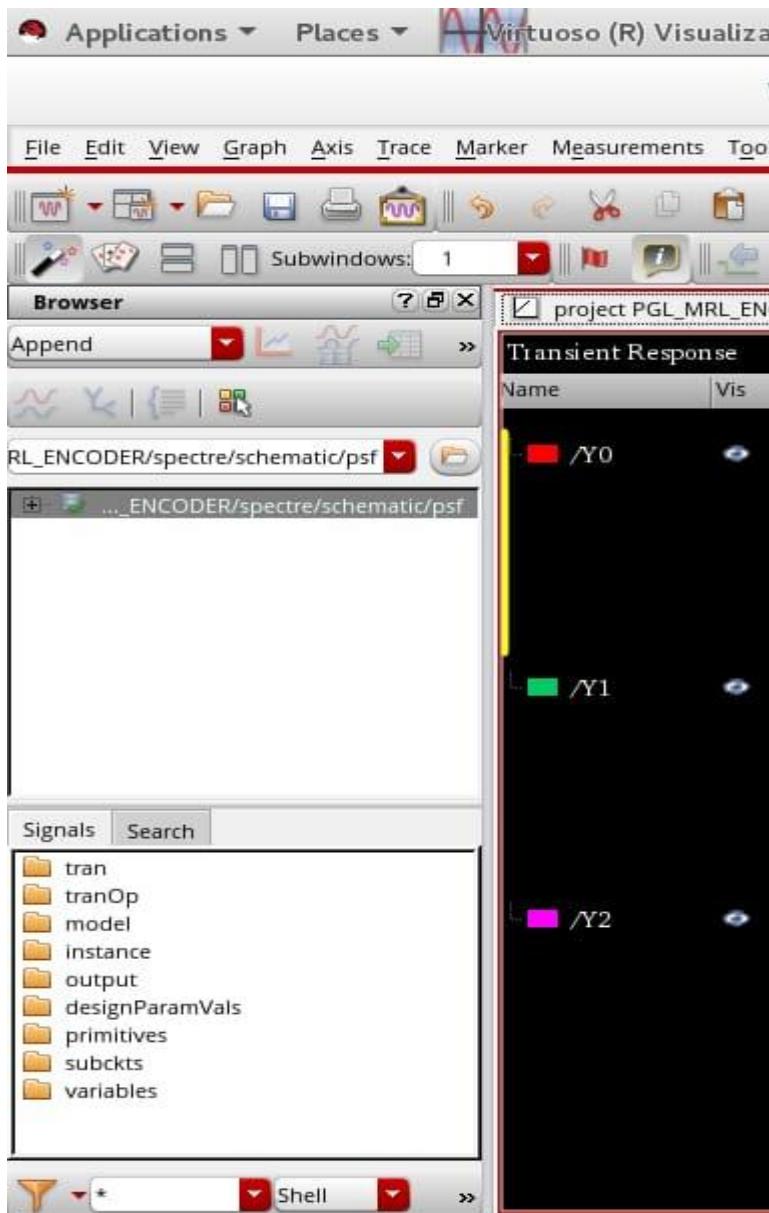
Design of memristor based low power encoder using PGL technique

In order to estimate the average power and delay for each logical circuit ADE L is very helpful with its tools like calculator. Both average power and delay can be calculated with the help of calculator.

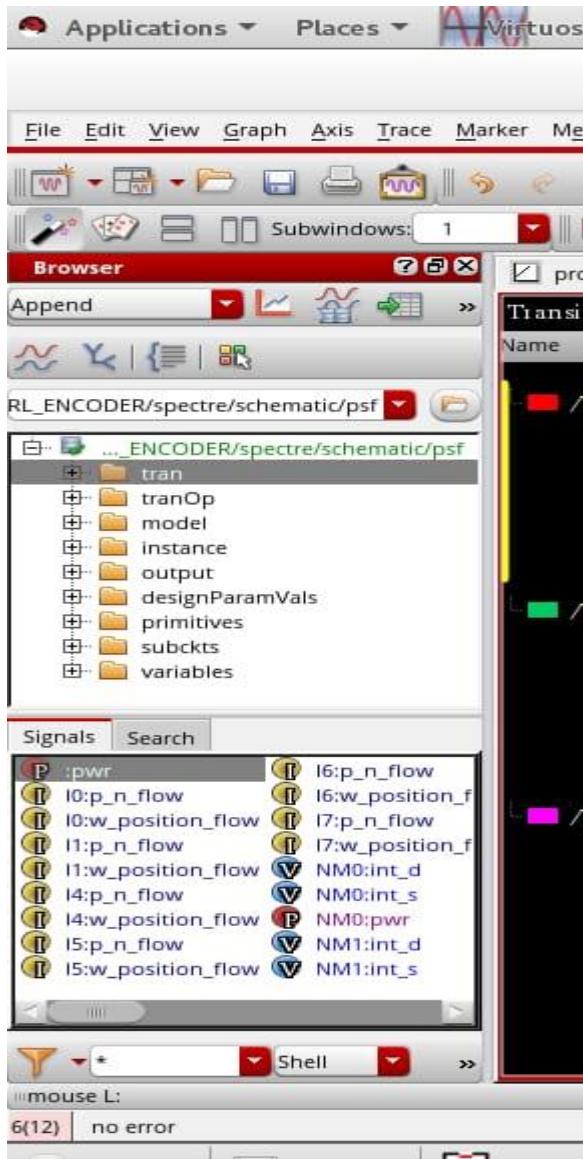
In the ADE L tab in the top menu select OUTPUTS →SAVE ALL option. And in the new window enable the power signals to output option to all and click OK.



After enabling the power to output to all. From the menu select TOOLS -> RESULT BROWSER. This will enable the power graph and various other pins outputs to result window. Now open the waveforms window and we can observe sub windows at left side as follows:

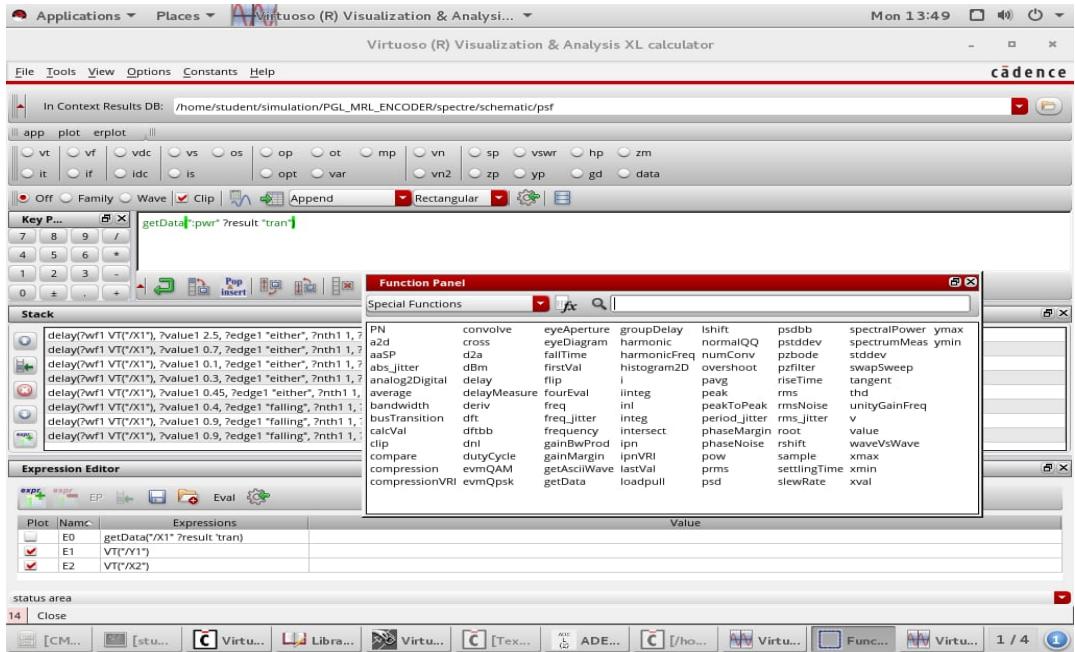


Select **trans** from the signals and we can observe various pins that the circuit has. And also we can observe a power as pwr signal

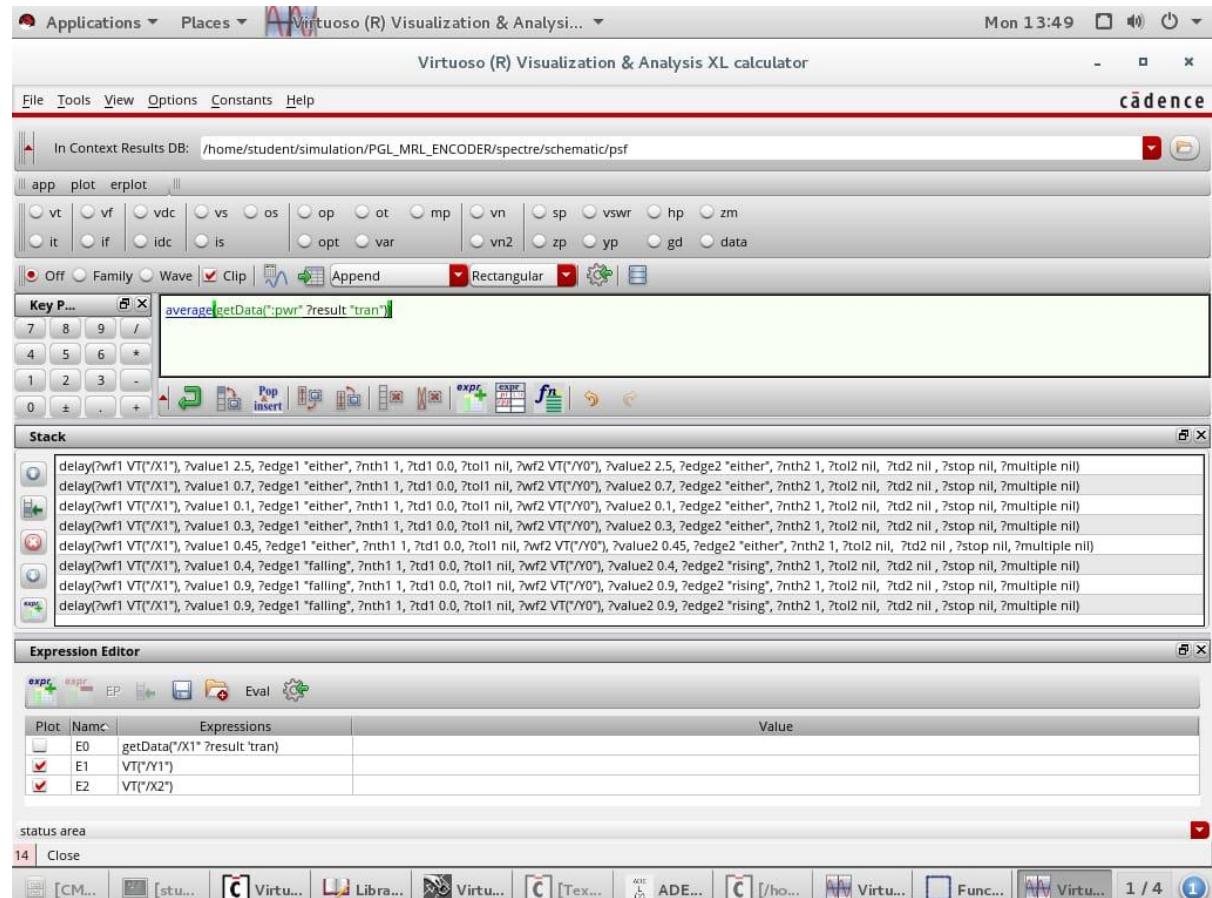


Right click on the pwr signal and select TO CALCULATOR option this will send the power equation to the calculator.

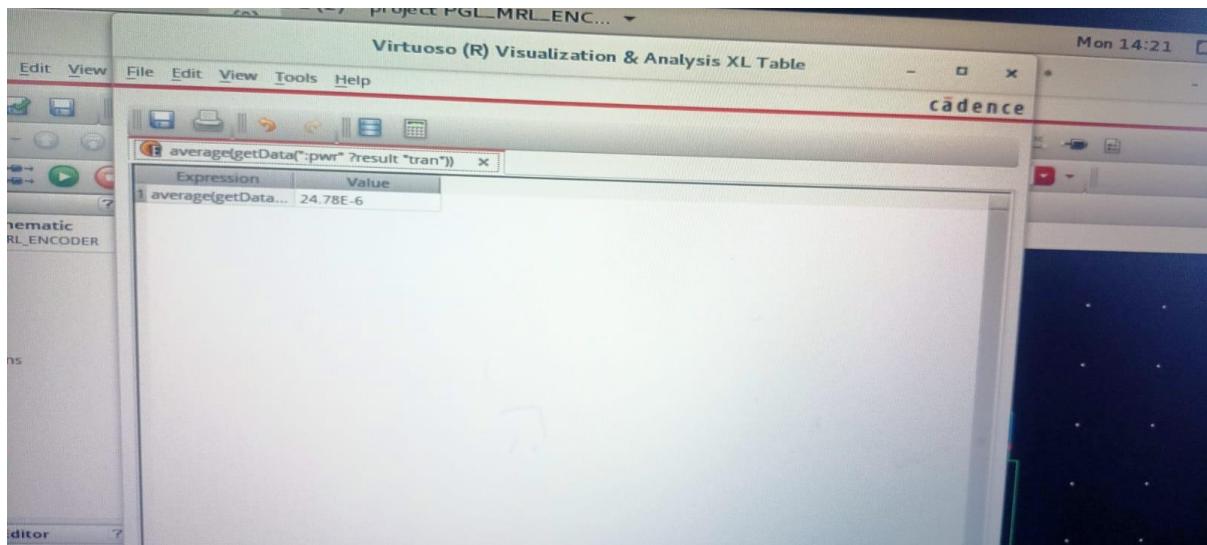
Design of memristor based low power encoder using PGL technique



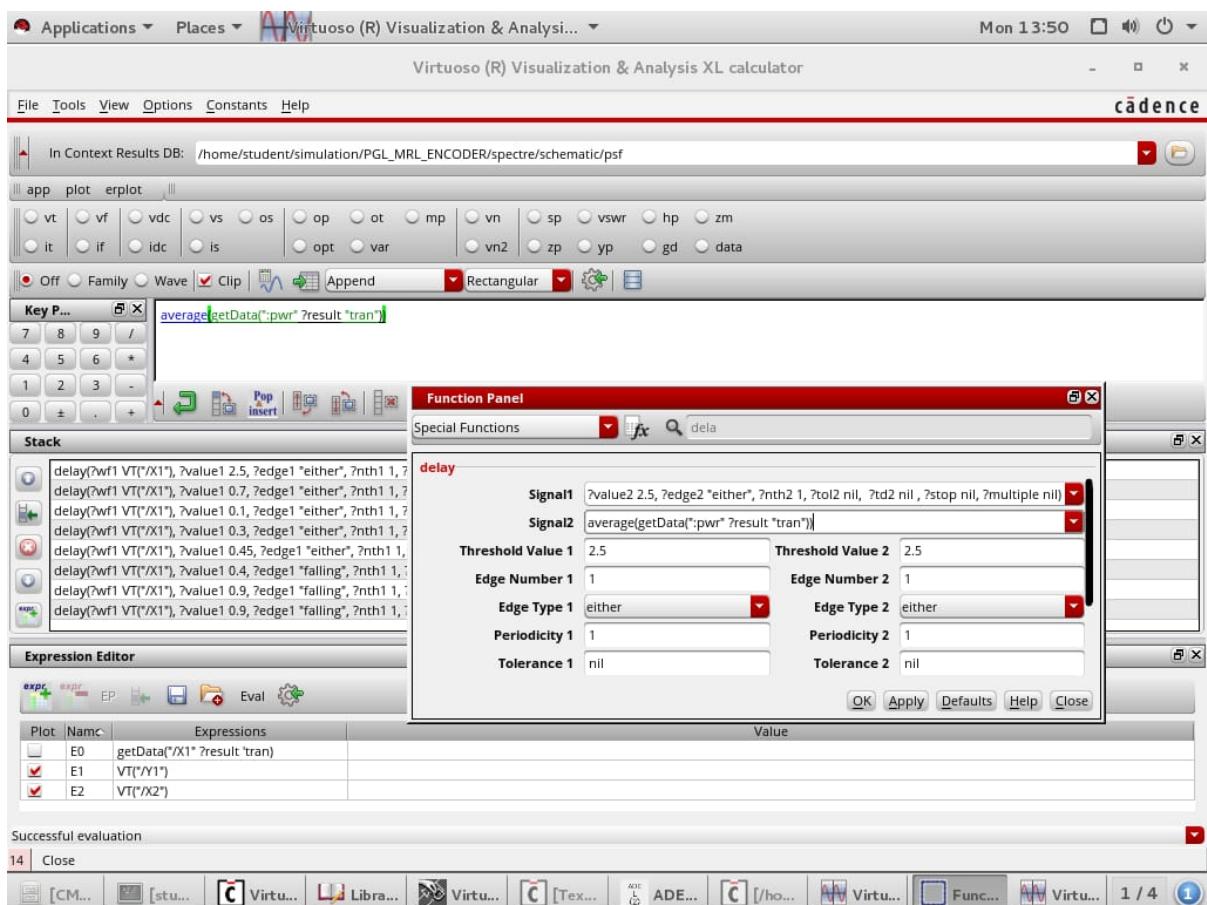
Now in order to find the average power from the function panel select average function through which we can find the average of the power.



To analyze the value you can table with in arrow icon.



To find the delay of the circuit in function panel of the calculator we have a function called delay through which you can find the delay.



The threshold value of the signals can be set to average DC voltage i.e., 1.6. To observe the same value you can use the mentioned icon to send to the table.

CHAPTER 4

ENCODER DESIGN AND IMPLEMENTATION

4.1 IMPLEMENTATION OF ENCODER USING CMOS TECHNOLOGY

These days, complementary metal oxide semiconductor logic, or CMOS logic, is the most widely used type of logic. Both N-type and P-type are employed in this reasoning. A transistor is turned on by a signal, while another transistor is turned off by the same signal. Instead of using pull-up resistors to implement this logic, just basic switches are needed. P-type transistor collections are arranged in CMOS logic as pull-up networks between the output and the high voltage rail, and N-type transistor collections are arranged as pull-down networks between the output and the low voltage rail. P-type transistors are ON and N-type transistors are OFF when their gates are coupled to the same input signal, and vice versa.

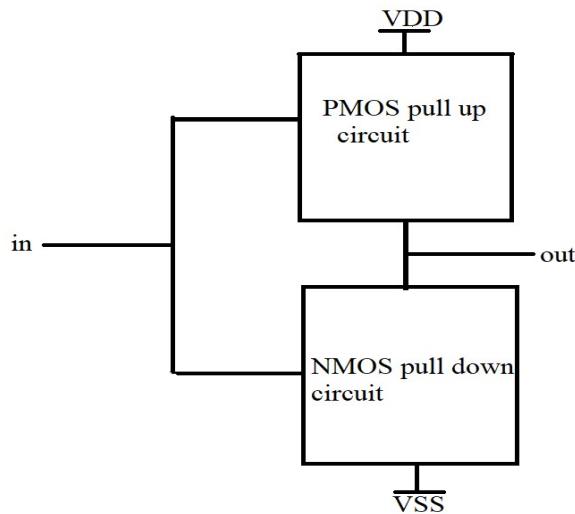


Fig 4.1: Block diagram of CMOS logic

Each logic gate in CMOS logic is implemented by a pair of complementary MOSFETs, one n-type and one p-type. Complex digital circuits like adders, multipliers, and memory systems can be made by combining these gates.

Design of memristor based low power encoder using PGL technique

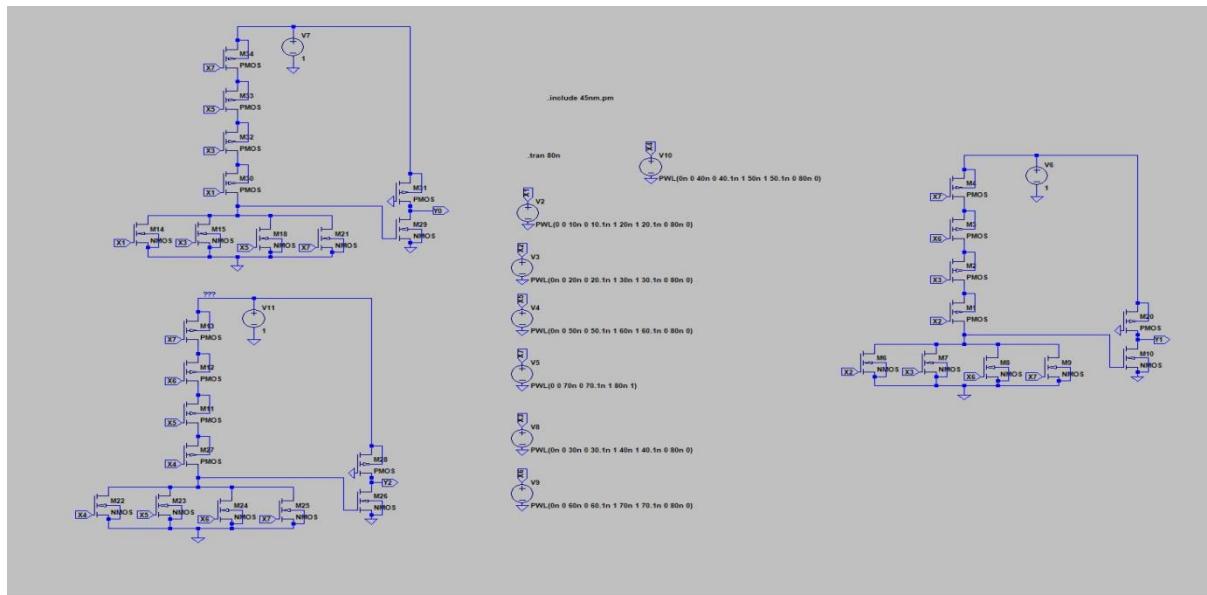
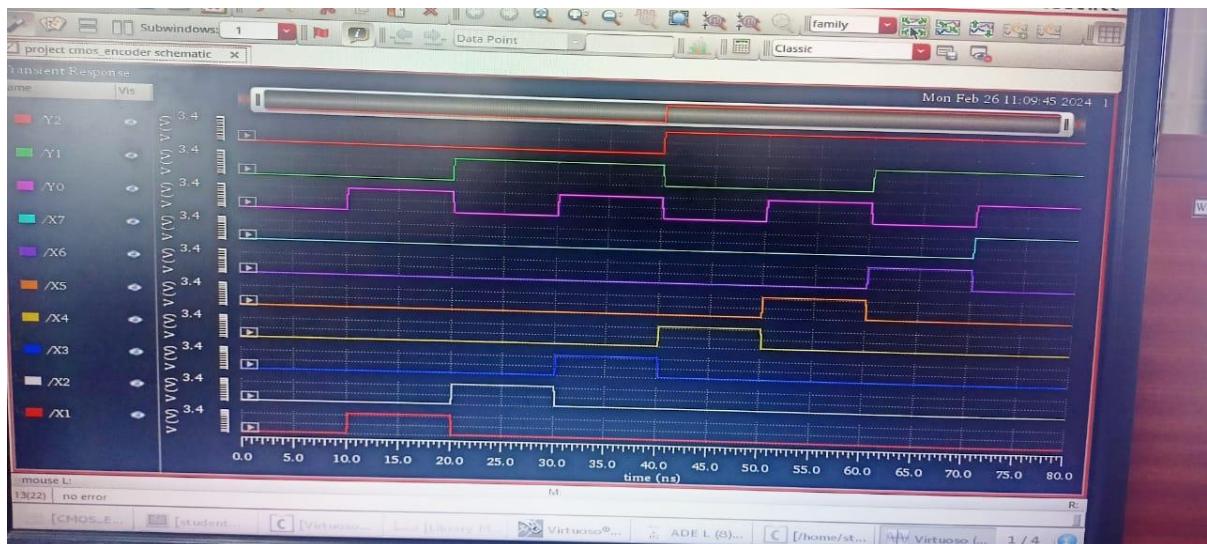


Fig 4.2 Schematic diagram of Encoder Using CMOS Technology



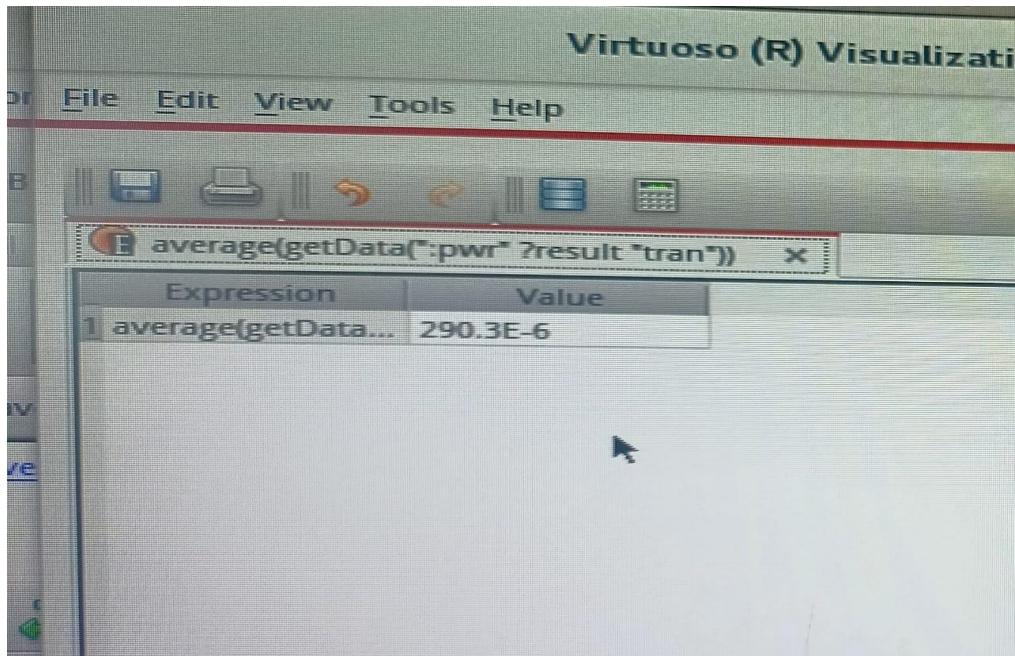


Fig 4.3 Average power for Encoder using CMOS technology

Y0, Y1, and Y2 are the encoder's CMOS technology outputs. This technique has an average power consumption of 290.3 e^{-6} which is equal to **719.5 mW** and a latency of 129.57ps. The CMOS circuit receives inputs X0, X1, X2, X3, X4, X5, X6, and X7. The inputs of the CMOS circuit are X1, X3, X5, and X7. To obtain the output Y0, these inputs must be active HIGH. Likewise, X2, X3, X6, X7 were Active HIGH to obtain the Y1 result. X4, X5, X6, and X7 inputs are Active HIGH to produce the Y2 output.

4.2 ENCODER USING PSEUDO NMOS TECHNOLOGY

Before CMOS technology was developed, pseudo NMOS logic was a common form of digital logic circuit utilized in the early days of integrated circuit design. Its foundation is the creation of a logic gate by using a single NMOS (n-type MOSFET) transistor as a pull-down device in conjunction with a resistive load. When the input is high in a pseudo NMOS gate, the NMOS transistor is employed to pull the output voltage down to ground. The pull-up resistor controls the output voltage when the input is low, turning off the transistor. The supply voltage, which is normally 5 volts in the majority of digital systems, and the output voltage fluctuate between each other. The supply voltage, which is normally 5 volts in the majority of digital systems, and the output voltage fluctuate between each other.

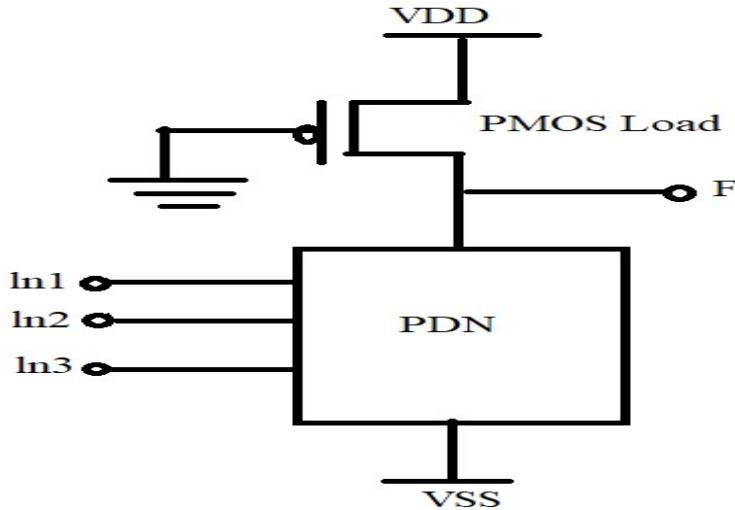


Fig 4.4 Block diagram of pseudo NMOS

The input signal drives a pull-down or driver for n devices. This is referred to as "Pseudo-NMOS" since it is basically equal to using a depletion on load in NMOS technology. Numerous CMOS logic circuits employ the circuit. In this, PMOS will typically be in a linear region. As a result, the RC time constant is low due to low resistance. An uninterrupted DC current flows through the circuit when the driver is turned on. One PMOS device serves as the pull-up device for a multi-transistor N-Logic block in pseudo-NMOS logic. As a result, a $N+1$ transistor is needed for every N input gate that operates. The used PMOS device is always "ON" because its gate is connected to ground. The output is charged to Vdd while the N-Logic block is in the "OFF" state. A substantial current that flows from Vdd to ground while the N-block is "ON" can consume a lot of power. Pseudo-NMOS devices need to have carefully specified dimensions so that the output can be discharged to ground. The PMOS device needs to be selected such that it is both thin enough to allow the N-block to safely pull down the output and wide enough to conduct a multiple of the leakage current of the N-block when the output is "high". Pseudo-NMOS logic is hence a limited logic. The speed advantage of pseudo-NMOS logic over static CMOS logic is particularly noticeable in big fan-in NOR gates. This is because the output rise time is only being contributed by one PMOS transistor. The total speed boost is significant, but at the expense of a marginal rise in power usage.

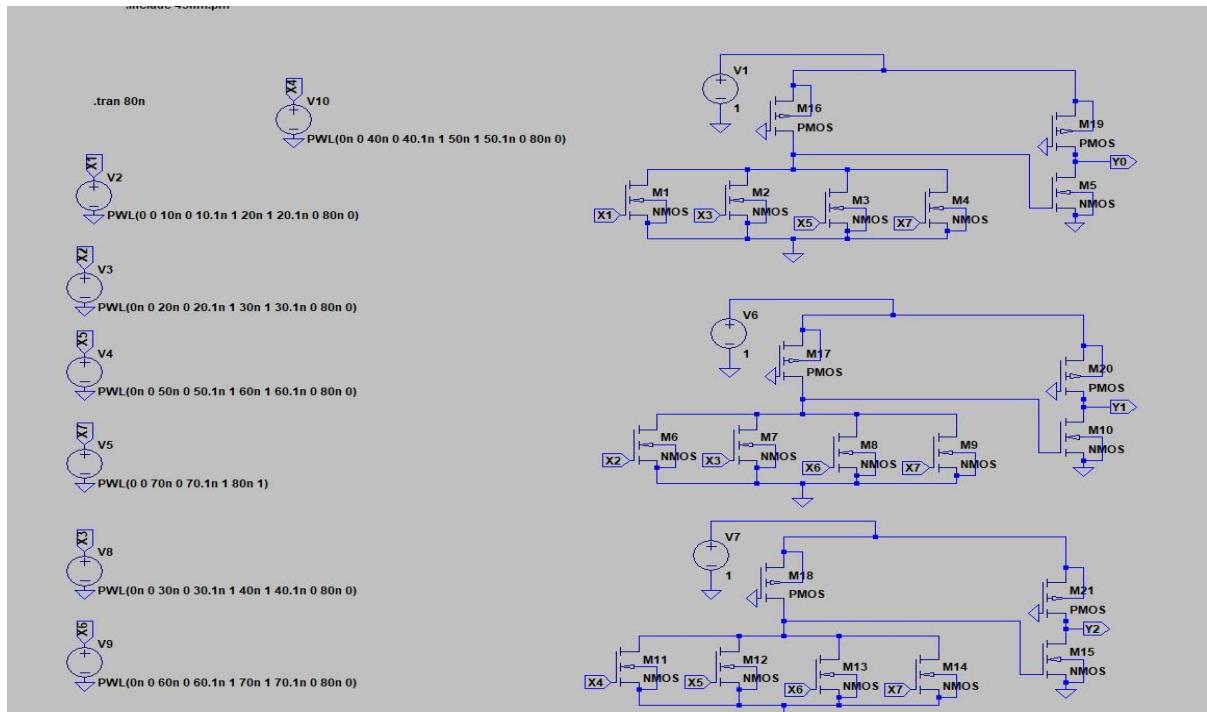


Fig 4.5 Schematic diagram of Encoder Using Pseudo NMOS Technology

OUTPUT WAVEFORM OF PSEUDO NMOS BASED ENCODER

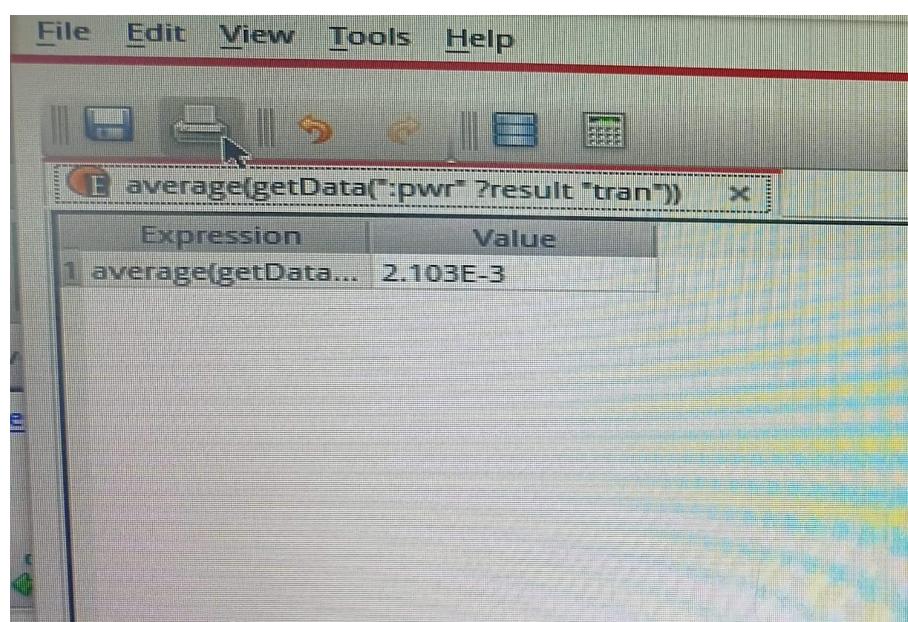
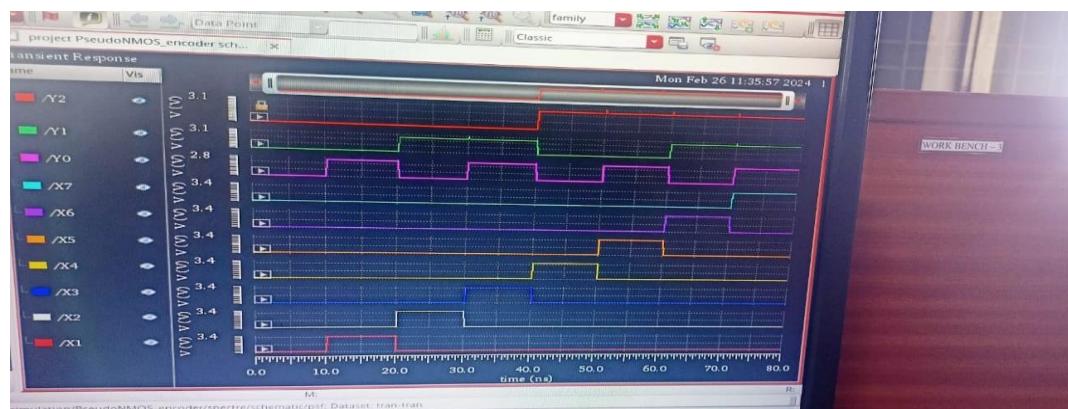


Fig 4.6 Average power for the encoder design using pseudo NMOS technology

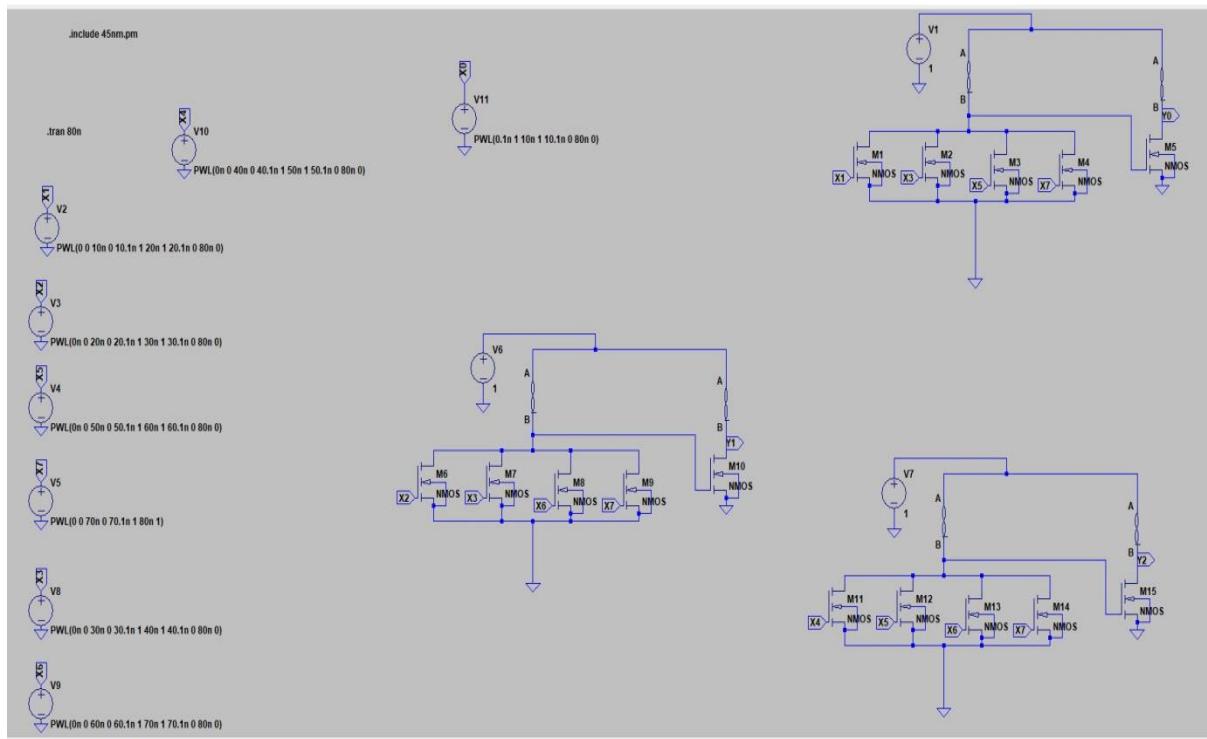
When compared to CMOS technology, pseudo NMOS technology consumes significantly less power on average 2.103 e^{-1} which equals to **104.7 mW**, with a delay of 177.94ps. The CMOS circuit receives inputs X0, X1, X2, X3, X4, X5, X6, and X 7. The inputs of the CMOS circuit are X1, X3, X5, and X7. To obtain the output Y0, these inputs must be Active HIGH. Likewise, X2, X3, X6, X7 were Active HIGH to obtain the Y1 result. Y2 is the result of ORing inputs X4, X5, X6, and X 7.

4.3 ENCODER USING MRL TECHNOLOGY

In a memristor-based encoder, a network of memristors arranged in a certain pattern is subjected to a series of binary inputs. The memristors' resistances vary as current passes through them, producing a particular output voltage that is indicative of the encoded value. In order to improve the network topology and memristor properties, the memristor network design process combines simulation and optimization approaches, which is essential to obtaining accurate and dependable encoding. Nevertheless, memristor-based encoders come with certain drawbacks. The scarcity of memristor devices—which are still relatively new and not yet extensively available—is one of the primary obstacles. Furthermore, the design of memristor-based encoders may be more difficult than that of typical encoders due to the non-linear behavior of memristors, necessitating the use of more advanced simulation and circuit design tools.

The linking of two memristors, M1 and M2, in parallel is shown in Fig. 3.4.1. The output of this is provided to the CMOS inverter, which produces NOR gate logic. The potential across the terminals is one if A and B are both zero. In this case, the CMOS inverter output is zero. In a similar vein, the CMOS inverter's output is zero for A=0 and B=1. The output of the CMOS inverter is one when the voltage across the terminals is one, which occurs when A and B are both one.

Design of memristor based low power encoder using PGL technique



**Fig 4.7 Schematic diagram of Encoder Using MRL Technology
OUTPUT WAVEFORM OF MRL BASED ENCODER**



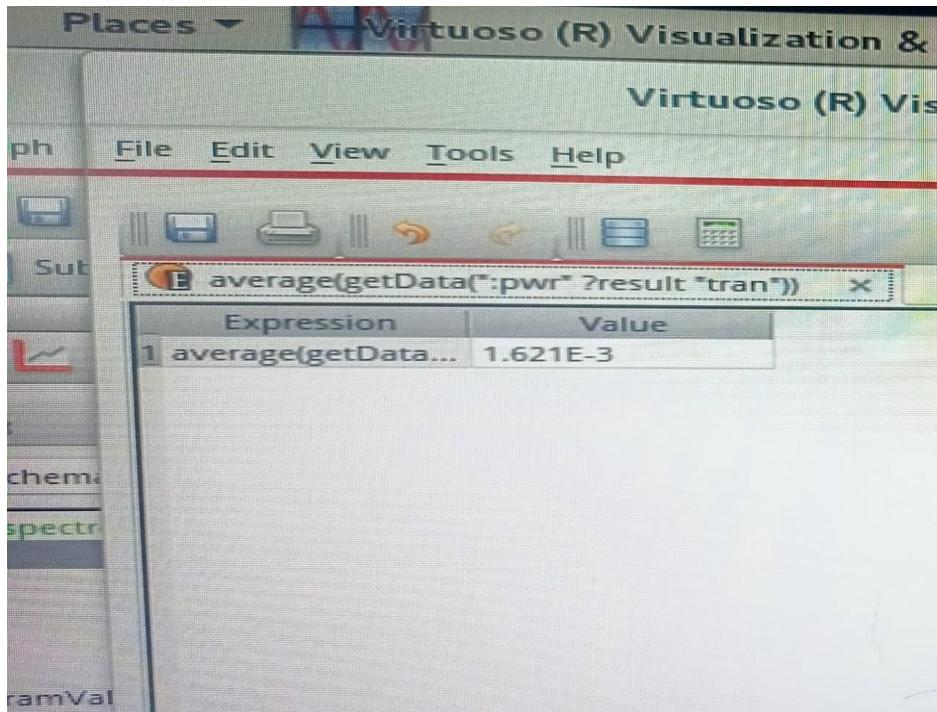


Fig 4.8 Average power consumed for the memristor based encoder

The memristor-based encoder has an average power consumption of 1.621 e^{-3} which is equals to **80.7 mW**, which is 88.7% less than that of the CMOS-based encoder and 22.9% less than that of the PSEUDO NMOS-based encoder. The memristor-based encoder uses PGL, which results in a 124.02ps delay. The CMOS circuit receives inputs X0, X1, X2, X3, X4, X5, X6, and X7. The inputs of the CMOS circuit are X1, X3, X5, and X7. To obtain the output Y0, these inputs must be Active HIGH. Likewise, X2, X3, X6, X7 were Active HIGH to obtain the Y1 result. X4, X5, X6, and X7 inputs are Active HIGH to produce the Y2 output.

4.4 IMPLEMENTATION OF MEMRISTOR BASED ENCODER USING PGL TECHNIQUE

The schematic diagram of a PGL-based memristor encoder in Cadence Tool displayed in Figure. Eight sources of voltage There are eight additional transistors. In Cadence, a schematic diagram is created. Comparing encoders using CMOS, pseudo-NMOS, and MRL yields different findings from those with a memristor-based encoder employing PGL.

Design of memristor based low power encoder using PGL technique

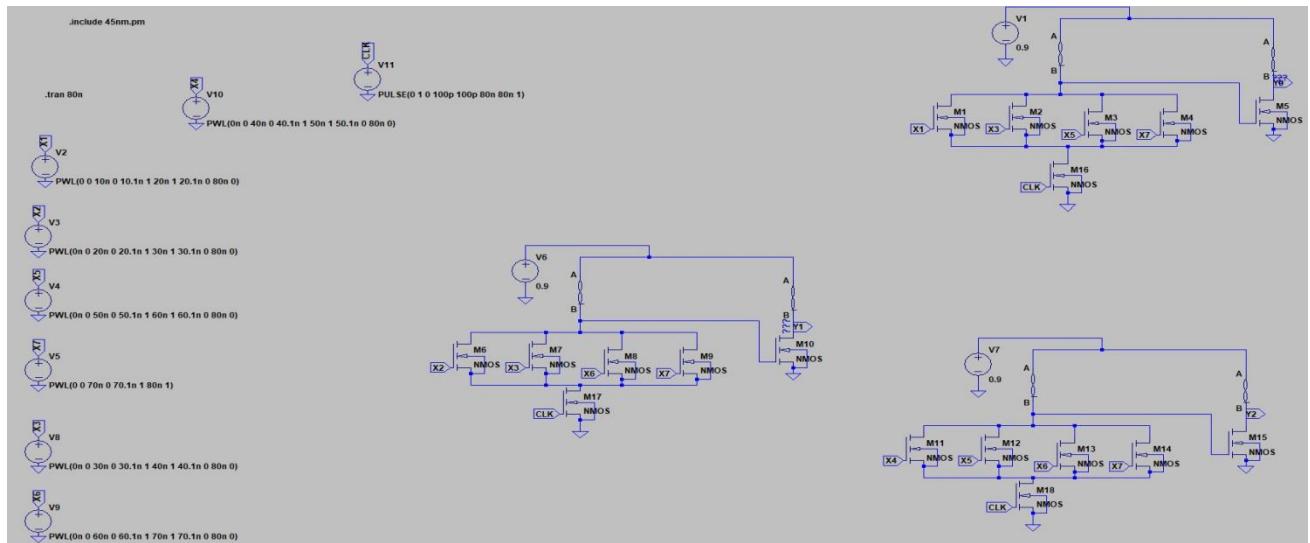
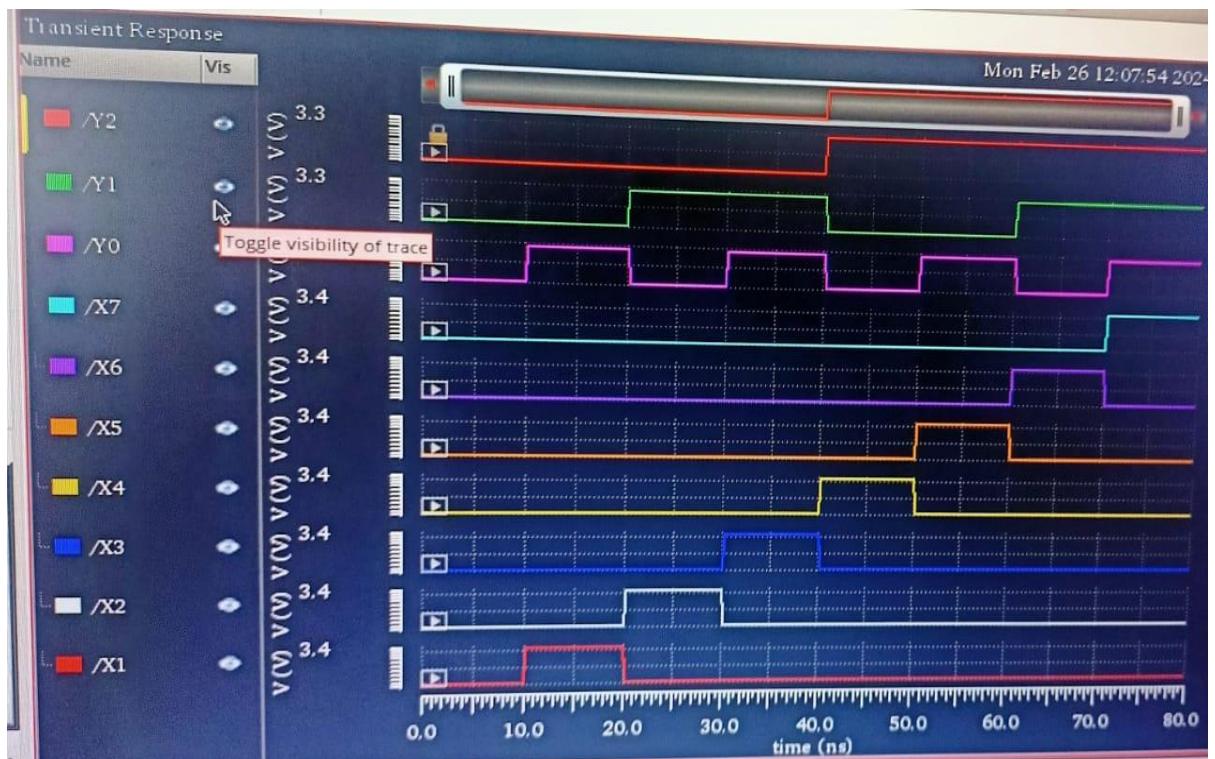


Fig 4.9 Schematic of Memristor based PGL encoder



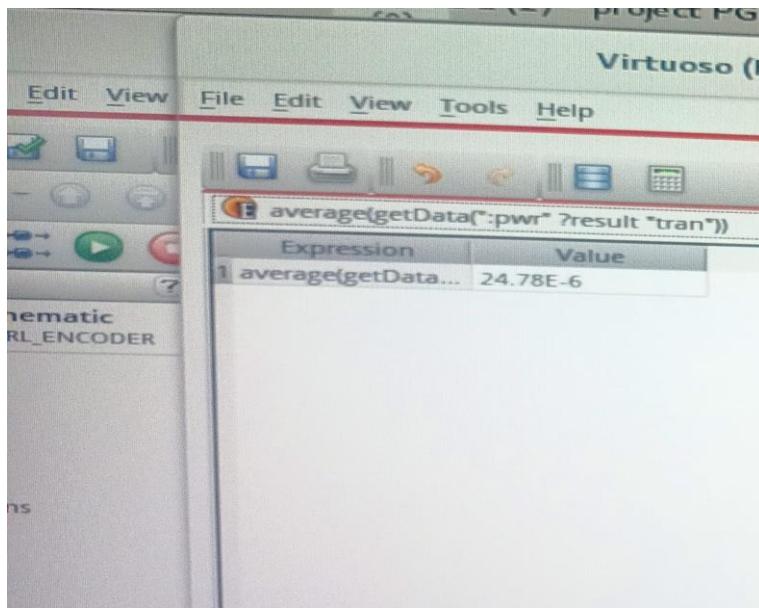


Fig 4.10 Input and output wave forms of memristor based encoder using PGL technique

The PGL technique output waveform of a memristor-based encoder reveals that the average power consumption is 24.78 e^{-6} which equals to **61.4 mW**, which is 23.9% less than that of a memristor-based encoder with a delay of 173.74ps.

X4 demonstrated a high voltage level between 40 and 50 ns, X5 demonstrated a high voltage level between 50 and 60 ns, X6 demonstrated a high voltage level between 60 and 70+ ns, and X7 was high voltage between 70 and 80 ns. The memristor-based encoder's output waveforms are Y1, Y2, and Y3. When the X1, X3, X5, and X7 are high, Y1 also gets Active HIGH. The peak of Y2 occurs when the X2, X3, X6, X7, and Y3 occur when the X4, X5, X6, X7, and Y3 are at their highest points.

4.5 ADVANTAGES

1. Lower power consumption
2. Less Delay
3. Less power-delay (PDP) factor

4.6 APPLICATIONS

Data backup technology

Communication

CHAPTER 5

RESULT AND VALIDATION

Memristor research and development is essential to improving the performance of circuits made in the current generation. Memristors provide many benefits, including higher density designs and non-volatility with CMOS. Memristors' non-volatility is very inspiring for memory design. In memory calculations are being thoroughly studied with the aid of PGL Technique implementation, opening the door for a possible revolution in computer architecture known as Beyond-Newman architecture. This could eliminate the Von Newman architecture's existing bottleneck. Memristor using PGL method requires less power and delays less time.

5.1 COMPARISON OF PERFORMANCE PARAMETERS

When compared to encoders designed with CMOS, pseudo-NMOS, or MRL technology, memristor-based encoders using PGL approach have lower power consumption and less latency.

COMPARISION TABLE

	Average power(<i>mW</i>)	Delay(ps)	Power-delay Product (PDP)
CMOS based encoder	719.5	129.57	93225 x 10^-12
Pseudo NMOS encoder	104.7	177.94	18630 x 10^-12
MRL based encoder	80.7	134.031	10816 x 10^-12
MRL based encoder using PGL	61.4	173.74	10667 x 10^-12

5.2 ANALYSIS

In terms of average power and latency, the table compares the pseudo-NMOS based encoder, MRL based encoder, CMOS based encoder design, and Memristor based encoder utilizing PGL approach. The MRL based encoder has an average power and delay of 80.7mW and 134.031ps; the pseudo NMOS encoder has an average power and delay of 104.7mW and 177.94ps; the MRL based encoder employing PGL has an average power and delay of

61.4 mW and 173.74ps. This investigation shows that a PGL-based memristor encoder uses 23.9% less power compared to the memristor based encoder alone.

POWER

The memristor based encoder with power gating logic (PGL) uses 91.4% less power than CMOS encoder circuit and 41.3% less power than pseudo NMOS encoder circuit and 23.9% less than the memristor based encoder. So, therefore 23.9% of power saving is achieved by implementing power gating technique in the memristor based encoder.

DELAY OVERHEAD

The delay of the memristor based encoder with PGL technique is more than the memristor based encoder (MRL logic) but it is less than the Pseudo NMOS encoder by 2.36%. But in terms of performance without delay CMOS encoder logical circuit is at the top than any other logical circuit.

POWER-DELAY PRODUCT

As represented in the tabular form the power-delay product (PDP) factor which is used to determine the performance of digital integrated circuit of the PGL induced memristor based encoder has smaller value than any other design. Thereby declaring memristor based encoder using PGL technique is much more efficient from the remaining logical circuits.

CHAPTER 6

CONCLUSION & FUTURESCOPE

6.1 CONCLUSION

When compared to pseudo NMOS and regular CMOS logic, memristor-based encoders in Power Gating logic are significantly more efficient in terms of both power and area. A circuit's power, area, and speed are always being traded off. It is therefore possible to draw the conclusion that this design method requires fewer transistors overall. By employing the MRL Technique, the encoder's total power consumption is lowered by 23.9% to 61.4mW from 80.7mW. This benefit makes the process of constructing a digital circuit more effective. Memristor devices execute a detection calculation in real time and function as artificial synapses with reversible short-term plasticity. Typical We demonstrate real-time adaptive control with remarkable reproducibility using our monitor-compute-actuate paradigm on in vitro hippocampal neural cells. The creation of memristive neural networks is one of the many fields in which memristor emulators are used in computation. Memristor neural networks are neural networks that achieve low power consumption and great processing density by using memristors as weights.

6.2 FUTURE SCOPE

Artificial Intelligence and machine learning (ML) are likely to play a bigger role in 6G networks and the 6G devices that link to them. Artificial Intelligence and machine learning (ML) are likely to play a bigger role in 6G networks and the 6G devices that link to them. Memristive computational memory can minimize data transportation both inside 6G networks and within 6G devices by processing data close to the sensor. In addition to improving user privacy, this type of local data processing would help to protect sensor data privacy. Inspired by these and other instances, our goal is to apply this study as a foundation for the effective deployment of AI applications in the future, thereby contributing to the long-term advancement of computing beyond CMOS in the 6G era.

REFERENCES

- [1]. G. Liu, S. Shen, P. Jin, G. Wang, and Y. Liang, “Design of memristorbased combinational logic circuits,” Circuits, Systems, and Signal Processing, vol. 40, no. 12, pp. 5825–5846, 2021.
 - [2]. S. Smaili and Y. Massoud, “Analytic modeling of memristor variability for robust memristor systems designs,” in 2014 IEEE International Symposium on Circuits and Systems (ISCAS). IEEE, 2014, pp. 794–797.
 - [3]. S. Kvativinsky, G. Satat, N. Wald, E. G. Friedman, A. Kolodny, and U. C. Weiser, “Memristor-based material implication (imply) logic: Design principles and methodologies,” IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 22, no. 10, pp. 2054–2066, 2013.
 - [4]. K. Mbarek, F. O. Rziga, S. Ghedira, and K. Besbes, “Design and properties of logic circuits based on memristor devices,” in 2020 IEEE International Conference on Design & Test of Integrated Micro & Nano Systems (DTS). IEEE, 2020, pp. 1–5.
 - [5]. S. Mandal, J. Sinha, and A. Chakraborty, “Design of memristor–cmos based logic gates and logic circuits,” in 2019 2nd International Conference on Innovations in Electronics, Signal Processing and Communication (IESC). IEEE, 2019, pp. 215–220.
 - [6]. A. Singh, “Memristor based xnor for high speed area efficient 1-bit full adder,” in 2017 International Conference on Computing, Communication and Automation (ICCCA). IEEE, 2017, pp. 1549–1553.
 - [7]. K. Alammari, A. Ahmadi, and M. Ahmadi, “Hybrid memristor-cmos based up-down counter design,” in 2020 27th IEEE International Conference on Electronics, Circuits and Systems (ICECS). IEEE, 2020, pp. 1–4.
 - [8]. A. Sasi, M. Ahmadi, and A. Ahmadi, “Low power memristor-based shift register design,” in 2020 27th IEEE International Conference on Electronics, Circuits and Systems (ICECS). IEEE, 2020, pp. 1–4.
 - [9]. D. B. Strukov, D. R. Stewart, J. Borghetti, X. Li, M. Pickett, G. M. Ribeiro, W. Robinett, G. Snider, J. P. Strachan, W. Wu et al., “Hybrid cmos/memristor circuits,” in 2010 IEEE International Symposium on Circuits and Systems (ISCAS). IEEE, 2010, pp. 1967–1970.
 - [10]. S. Kvativinsky, A. 83Kolodny, U. C. Weiser, and E. G. Friedman, “Memristor-based imply logic design procedure,” in 2011 IEEE 29th International Conference on Computer Design (ICCD). IEEE, 2011, pp. 142–147.
-

- [11]. S. Kvatincky, N. Wald, G. Satat, A. Kolodny, U. C. Weiser, and E. G. Friedman, “Mrl—memristor ratioed logic,” in 2012 13th International Workshop on Cellular Nanoscale Networks and their Applications. IEEE, 2012, pp. 1–6.
- [12]. V. A. Slipko and Y. V. Pershin, “Importance of the window function choice for the predictive modelling of memristors,” IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 68, no. 6, pp. 2167– 2171, 2019.
- [13]. L. O. Chua and S. M. Kang, “Memristive devices and systems,” Proceedings of the IEEE, vol. 64, no. 2, pp. 209–223, 1976.
- [14]. E. Lehtonen, J. H. Poikonen, and M. Laiho, “Applications and limitations of memristive implication logic,” in 2012 13th International Workshop on Cellular Nanoscale Networks and their Applications. IEEE, 2012, pp. 1–6.
- [15]. L. Chua, “Memristor—the missing circuit element,” IEEE Transactions on circuit theory, vol. 18, no. 5, pp. 507–519, 1971.