

# DESIGN OF MEMRISTOR BASED LOW POWER ENCODER USING PGL TECHNIQUE

**Abstract**—This project focuses on devising an encoder utilizing memristors with Power Gating Logic (PGL), offering a potential alternative to conventional CMOS logic circuits due to memristors' lower power consumption. Memristive devices, primarily designed for memory applications, present an intriguing avenue for logic circuitry. This study introduces a novel encoder architecture that employs PGL based on memristors and comparative analyses with other logic styles like Memristor Logic (MRL), CMOS, and pseudo-NMOS. The project demonstrates notable improvements through experimentation, including reducing the power-delay product (PDP) from 10816 p to 10666 p and decreasing average power consumption from 80.7 mW to 61.4 mW. Utilizing Cadence Tools, the study aims to construct and evaluate the performance of a 3-bit memristor encoder using the PGL methodology.

**Index Terms**—CMOS Logic, Pseudo NMOS Logic, Memristor logic, Encoder, Memristor in PGL Technique

## I. INTRODUCTION

Moore's Law has fueled a rapid increase in transistor density on semiconductor chips over recent decades, but challenges like cost, energy consumption, and material limitations are straining the semiconductor industry. Some propose developing smaller components to replace traditional transistors to sustain this trend. However, shrinking CMOS cell sizes exacerbate issues like leakage currents, leading to higher static power dissipation and increased error rates.

Addressing these challenges requires innovative solutions, such as passive devices. The memristor, conceptualized by Professor Chua in 1971 and first realized by HP Labs in 2008, offers promise in this regard. Memristors have many possible uses in things like FPGAs, amplifiers, and memory devices, and they can help fix problems in manufacturing. However, making intricate circuits, such as Memristor-aided logic circuits (MAGIC) and Material Implication (IMPLY), can be tough because of irregular fan-outs.

In digital applications, memristors represent logic states, with high memristance corresponding to logic 0 and low memristance to logic 1. Encoders, vital in digital circuitry for signal encoding and selection, can integrate memristors using various logic paradigms. This study implements an encoder using memristor ratioed logic and compares its power analysis and design with traditional CMOS and pseudo-NMOS circuits. Such endeavors aim to harness the potential benefits of memristors in advancing digital logic circuitry.

## II. WORKING OF MEMRISTOR AND MODELS

The operating principle of a memristor involves two terminals and a unique feature of retaining its state without applied voltage. It's regarded as the fourth fundamental element due

to its ability to link magnetic flux and charge. Fig. 1 illustrates the schematic diagram of the Memristor model. To create

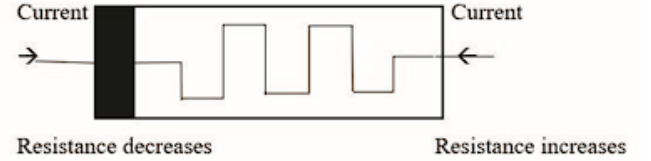


Fig. 1: Memristor Model

a memristor with dielectric properties, two layers of pure, undoped titanium dioxide (TiO<sub>2</sub>) are stacked atop each other. The memristance (M) is defined by the relationship  $M = d/q$ , where  $d$  represents magnetic flux (in Volt-seconds) and  $q$  denotes charge (in Coulombs). Doped titanium dioxide (TiO<sub>2</sub>x), placed on top of the insulator, behaves like a semiconductor. When electricity moves through the memristor, its resistance changes. Low resistance ( $R_{on}$ ) occurs when the width of the doped zone exceeds that of the undoped region within TiO<sub>2</sub>. Conversely, high resistance ( $R_{off}$ ) arises when the doped region's width is less than that of the undoped zone. Memristor polarity is typically indicated by a bold line. The equation representing the variable resistance of the memristor typically includes the parameter for off resistance ( $R_{off}$ ), which signifies the highest resistance achievable by the memristor.

$$R(w) = R_{on} \frac{w}{D} + R_{off} (1 - \frac{w}{D}) \quad (1)$$

In Figure 3, "D" shows the entire width of the TiO<sub>2</sub>, while "w" represents the width of the doped TiO<sub>2</sub>. The undoped width (w) of the memristor can either exceed or fall short of expectations based on the supplied current. The physical boundaries of the memristor define the undoped width, regulated by a window function

$$x = w/D \quad (2)$$

In this context, "x" indicates the proportion of the depletion region's width compared to the total width of the TiO<sub>2</sub>.

$$\frac{dx}{dt} = F(x) * K * I(t) \quad (3)$$

At a given time t,  $I(t)$  represents the current, while  $F(x)$  denotes the window function, with K representing a constant.

$$F(x) = 1 - (\frac{2w}{D-1})^2 p \quad (4)$$

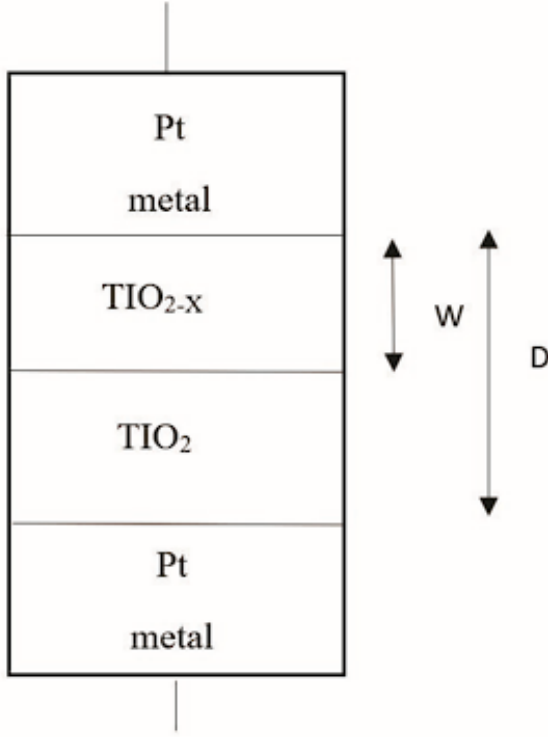


Fig. 2: Memristor Schematic

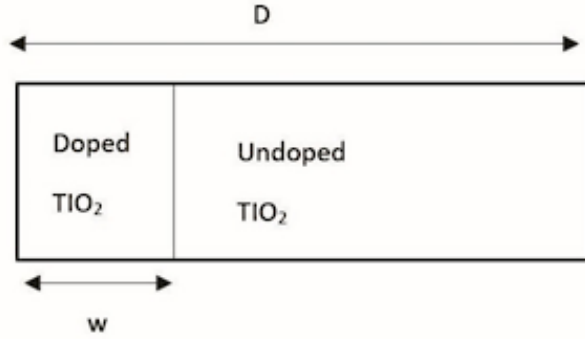


Fig. 3: Simple representation of a Memristor

In memristor ratioed logic, logic gates combine memristors (MRL) with CMOS technology, offering advantages over traditional CMOS designs. This integration seamlessly embeds memristors into the CMOS polysilicon layer, thereby reducing the overall design footprint effectively.. Additionally, fewer transistors are required. Figure 4 illustrates the CMOS inverter, which generates NAND gate logic, with the output obtained from the parallel connection between memristors M1 and M2. When both inputs A and B are set to 0, the voltage across the terminals becomes zero, leading to the operation of a single CMOS inverter. In contrast, Figure 5 demonstrates that reversing the memristor configuration yields NOR gate logic.

Advancing and researching memristors is pivotal for enhancing the efficacy of current-generating circuits. Memristors

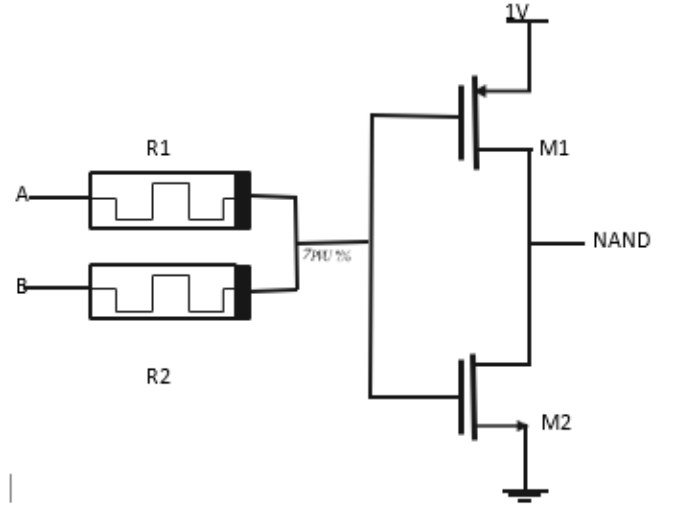


Fig. 4: Nand gate using Memristor

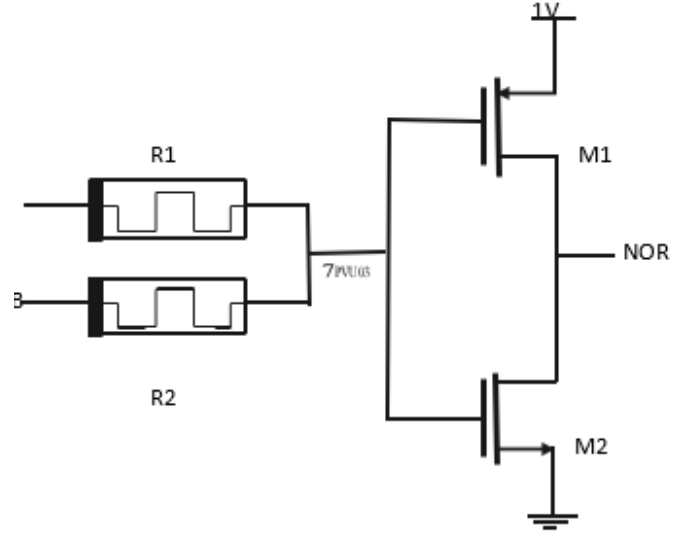


Fig. 5: Nor gate using Memristor

offer numerous advantages, such as improved design density, compatibility with CMOS technology, and nonvolatility. Their nonvolatile nature presents an excellent model for memory architecture. The implementation of Crossbar IMPLY logic gates has facilitated in-depth exploration of memory computations, potentially paving the way for a revolutionary architectural shift beyond the Von Neumann model in computer design. This could alleviate the existing bottleneck associated with Von Neumann architecture.

#### A. Linear Ion drift Model

$$R(w) = R_{on} \frac{w}{D} + R_{off} (1 - \frac{w}{D}) \quad (5)$$

$$\frac{dw}{dt} = \frac{u_v R_{on}}{Di(t)} \quad (6)$$

It utilizes the window function, is symmetric, and is simple to comprehend and apply.

$$F(x) = 1 - \left(\frac{2w}{D-1}\right)^2 p \quad (7)$$

### B. Simmons Tunnel Barrier

$$\frac{dw}{dt} = C_{off} \sinh\left(\frac{i}{i_{off}}\right) \exp\left(\frac{x - a_{off}}{\frac{w_e - |i|}{\frac{b-x}{w_c}}}\right), i > 0 \quad (8)$$

$$\frac{dw}{dt} = C_{on} \sinh\left(\frac{i}{i_{on}}\right) \exp\left(\frac{x - a_{on}}{\frac{w_e - |i|}{\frac{b-x}{w_c}}}\right), i < 0 \quad (9)$$

This precise model introduces increased complexity and asymmetric switching timings.

### C. Threshold Adaptive Model (TEAM)

In the TEAM model, the width of the undoped area acts as the state variable. It keeps the current thresholds, and you can tweak the model's parameters as needed. This model offers sufficient accuracy with reduced complexity, along with the utilization of window operations.

$$\frac{dx}{dt} = k_{off} \left(\frac{i(t)}{i_{off} - 1}\right)^a f_{off}(x), 0 < i_{off} < i \quad (10)$$

$$\frac{dx}{dt} = 0, i_{on} < i < i_{off} \quad (11)$$

$$\frac{dx}{dt} = k_{on} \left(\frac{i(t)}{i_{on} - 1}\right)^a f_{on}(x), i < i_{off} < 0 \quad (12)$$

The current is modeled as

$$V(t) = R_{on} i(t) \exp\left(\frac{y}{(x_{off} - x_{on})(x - x_{on})}\right) \quad (13)$$

$$\frac{R_{on}}{R_{off}} = e^y \quad (14)$$

To prevent the possibility of the memristor's width exceeding its physical dimensions, we employ window functions in simulation to constrain the width within the memristor's boundaries. In this study, we utilize the Joglekar window function of memristors along with the Ion Drift Model to model the encoder.

## III. WORKING OF ENCODER

An encoder, a specific type of combinational circuit, operates inversely to a decoder. With a maximum input line count of  $2n$  and a maximum output line count of ' $n$ ', it generates an equivalent active-high binary code corresponding to the input. Utilizing ' $n$ ' bits, it encodes the first  $2n$  input lines. Notably, encoders may not always necessitate the representation of the enable signal. In digital logic circuits, binary bits hold varying meanings for encoding data. The encoder's role involves executing this encoding task, triggered when one of the input bits reaches a significant level and affects the output, the circuit produces both ' $N$ ' and ' $M$ ' outputs. The relationship

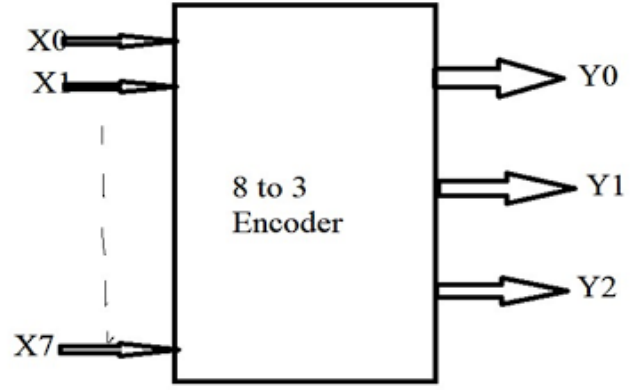


Fig. 6: 3-bit Encoder

$X_0$	$X_1$	$X_2$	$X_3$	$X_4$	$X_5$	$X_6$	$X_7$	$Y_2$	$Y_1$	$Y_0$
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	0	1
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. 7: 3-bit Encoder Truth Table

between the outputs and inputs in the Encoder truth table can be described as follows:

$$Y_0 = X_1 + X_3 + X_5 + X_7 \quad (15)$$

$$Y_1 = X_2 + X_3 + X_6 + X_7 \quad (16)$$

$$Y_2 = X_4 + X_5 + X_6 + X_7 \quad (17)$$

Based on these relationships, logic circuits for CMOS, Pseudo NMOS, and MRL (Memristor Ratioed Logic) can be designed. In the encoder circuits,  $Y_2$ ,  $Y_1$ , and  $Y_0$  represent the output bits, while  $X_0$  through  $X_7$  denote the input bits. Enabling the  $X_0$  input yields an equivalent binary information of 000 in the encoder circuit. Similarly, enabling the  $X_1$  input generates the binary representation of 1, or 001. All inputs are processed uniformly, producing an output corresponding to the input. Output  $Y_0$  is determined by utilizing inputs  $X_1$ ,  $X_3$ ,  $X_5$ , and  $X_7$ . Output  $Y_1$  is obtained through the OR operation on inputs  $X_2$ ,  $X_3$ ,  $X_6$ , and  $X_7$ . Likewise, OR operations on inputs  $X_4$ ,  $X_5$ ,  $X_6$ , and  $X_7$  yield  $Y_2$ .

## IV. APPROACH AND TECHNIQUE

In the power gating configuration, a circuit can function in one of two modes. When the sleep transistors are active, they act as functional redundant resistances and thus remain operational. However, when the system enters sleep mode, disabling the sleep transistors reduces leakage power. The terms "Header switch" and "Footer switch" refer to sleep

transistors positioned at VDD and close to ground, respectively. In this project, we employed the header power gating technique. Power gating has a more pronounced effect on

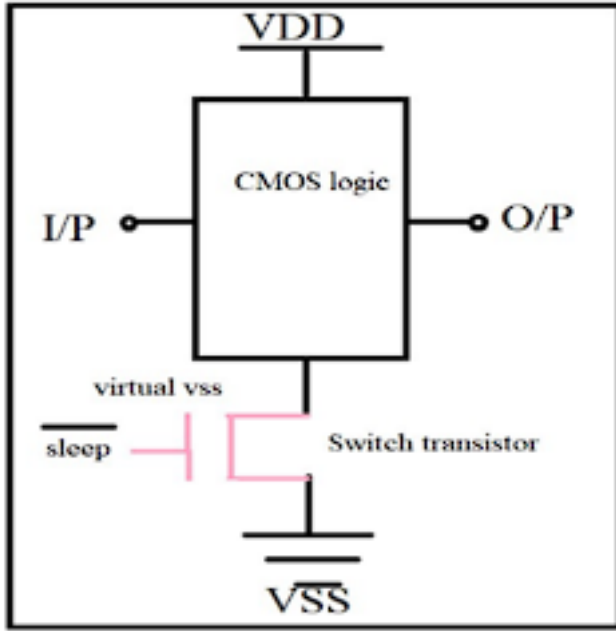


Fig. 8: Block Diagram of Power Gating Approach

design architecture when contrasted with clock gating. Longer delays occur due to the necessity of securely transitioning into and out of power-gated states. There are trade-offs in architecture between how much leakage power is saved and the energy needed to switch between low-power modes.

Blocks can be disabled using either software or hardware methods. Software-driven driver programs can schedule power-down procedures, while hardware timers offer alternative options. Additionally, employing a dedicated power management controller provides an extra option for efficient power management. Using an externally switched power source to

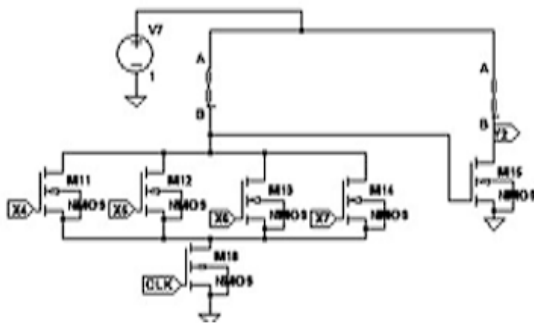


Fig. 9: Basic connection using PGL technique

implement power gating offers a straightforward approach to

decreasing long-term leakage power. However, internal power gating, which temporarily disables blocks, offers greater efficiency. Power gating controllers oversee the CMOS switches responsible for delivering power to the circuitry. As power-gated blocks turn off, their outputs slowly discharge, keeping output voltage levels above the threshold voltage for longer durations. However, this can potentially result in a rise in short-circuit current. In power gating, low-leakage PMOS transistors function as header switches, disconnecting power to components during standby or sleep mode, while NMOS sleep transistors act as footer switches. Sleep transistors divide the chip's power network into a permanent power network linked to the power source and a virtual power network that switches to power the cells as needed.

In a PGL-based memristor encoder with eight inputs and three outputs, the block diagram's input side for generating output Y0 consists of four PMOS transistors and one memristor. Additionally, an associated sleep transistor is connected to the circuit on the output side. To generate output Y0, five PMOS transistors, one sleep transistor, and two memristors are required. Likewise, to obtain output Y1, five PMOS transistors, two memristors, and one sleep transistor are needed. Similarly, for output Y2, five PMOS transistors, two memristors, and one sleep transistor are necessary.

## V. RESULTS AND DISCUSSION

"Memristor ratioed logic" (MRL) denotes the logic type utilized with memristors. In the circuits of 3-bit encoders, the PGL technique integrates CMOS, MRL, and memristors.

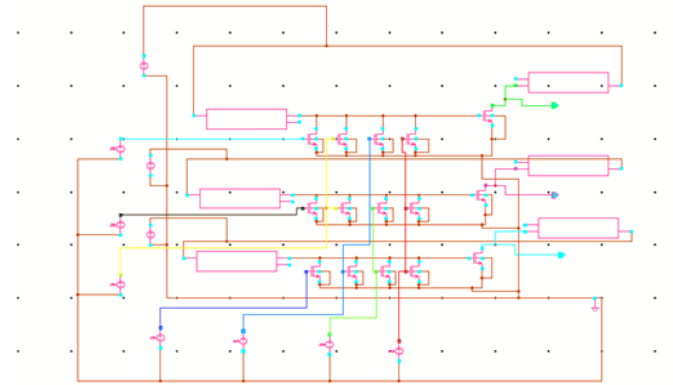


Fig. 10: Encoder using Memristor-based logic technique

Both the encoder's table and the output can be displayed similarly. The encoder output was derived using a memristor with an initial resistance of  $R_{init}=8M$ , where  $R_{on}=1$  and  $R_{off}=10M$ . Verilog-AMS was employed to model the circuit component. The memristor symbol was created using Cadence tools, as depicted in Fig. 7. Moreover, the 3-bit encoder output was verified using pseudo-NMOS and CMOS hardware, as depicted in Figures 11 and 14, respectively. The average power consumption measured during the experiment was 784.67W. Figure 15 displays the power loss in the pseudo-NMOS

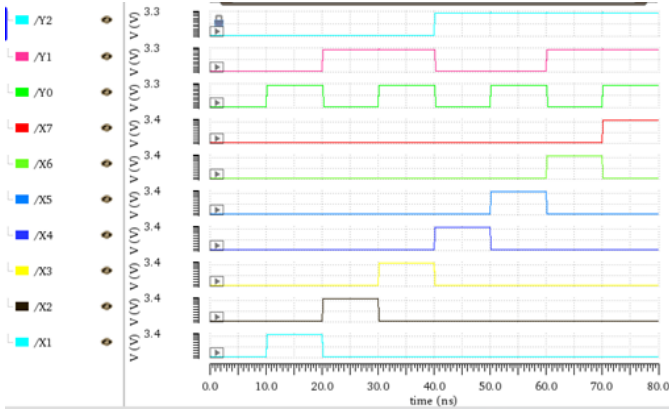


Fig. 11: output of memristor logic

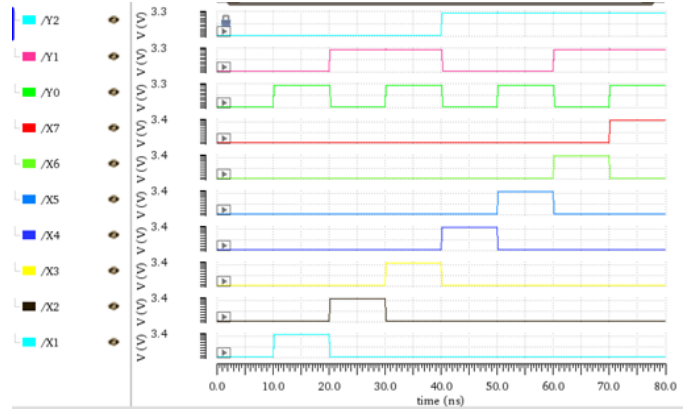


Fig. 13: output of CMOS logic

circuit, whereas Figure 12 illustrates the power dissipation in a standard CMOS circuit.

Considering that the known area occupied by a memristor-based architecture is smaller than that of a CMOS architecture, it can be demonstrated that it is more energy-efficient in terms of area consumption. In comparison to traditional Pseudo

CMOS logic, this analysis results in a reduction of the power-delay product (PDP) to 10666pW from 10816pW and an average power of 80.7mW from 61.4mW. The primary goals of this project are to create and analyze a 3-bit encoder using the PGL technique in Cadence Tool.

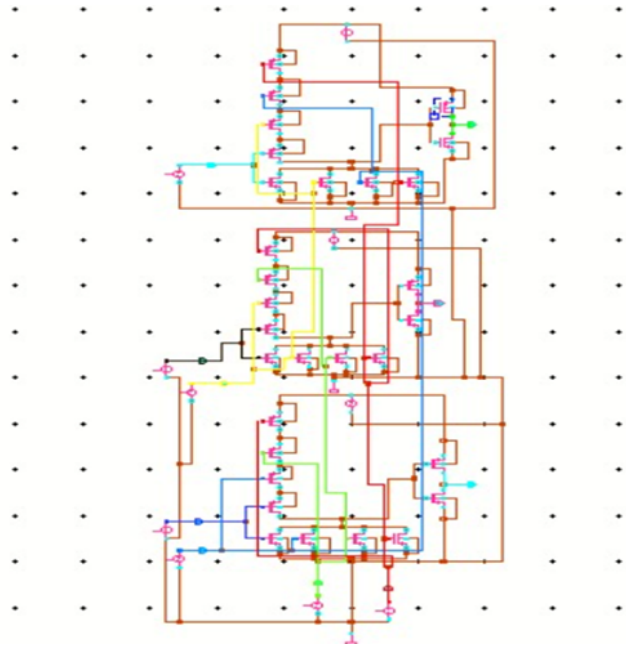


Fig. 12: Encoder using CMOS logic

NMOS and CMOS logic, it is clear that this design employs the fewest transistors. The CMOS encoder circuit contains thirty transistors, fifteen of which are PMOS and fifteen of which are NMOS. The encoder circuit includes 21 transistors total—15 NMOS and 5 PMOS—thanks to pseudo-NMOS technology. The encoder comprises a total of 21 transistors, including six memristors and 15 NMOS transistors, made possible by MRL technology. The final encoder is memristor-based, incorporating six memristors along with eighteen transistors featuring power-gating logic. Additionally, the encoder

## VI. CONCLUSION AND FUTURE SCOPE

When it comes to power and area efficiency, The traditional CMOS logic and pseudo-NMOS logic PGL technique is much inferior to the encoder design that makes use of a memristor-based logic architecture. There is always a trade-off between a circuit's power, area, and speed. Employing this design method leads to a higher circuit fabrication efficiency and the use of fewer transistors.

### A. Power Analysis

Utilizing the PGL technique in a memristor-based encoder achieves significant power savings compared to various other circuits: it consumes 91.4% less power than a CMOS encoder circuit, 41.3% less power than a pseudo-NMOS encoder circuit, and 23.9% less power than a memristor-based encoder alone. Hence, to attain a 23.9% reduction in power consumption, implementing the PGL technique with the memristor-based encoder is recommended.

Further optimization of the encoder design using the PGL technique can be pursued to enhance its performance metrics such as power consumption, speed, and area efficiency. Extending the encoder design to handle higher bit widths, beyond the 3-bit encoder, can be a promising direction. Scaling up to 4-bit, 8-bit, or even higher-bit encoders would demonstrate the scalability and versatility of the PGL technique with memristors. Investigating novel applications and use cases for memristor-based encoders beyond conventional digital logic circuits. This could include applications in emerging fields such as neuromorphic computing, machine learning accelerators, or Internet of Things (IoT) devices.



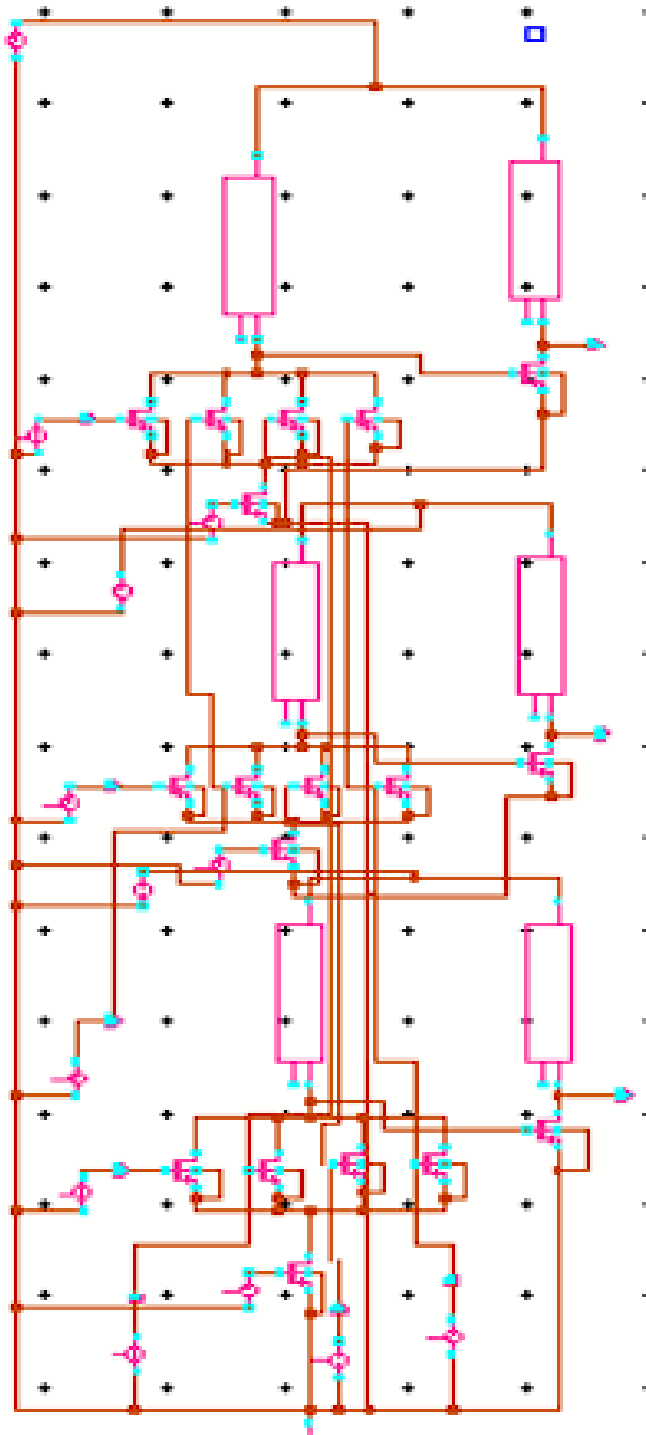


Fig. 14: Encoder using PGLTechnique

#### REFERENCES

[1] G. Liu, S. Shen, P. Jin, G. Wang, and Y. Liang, "Design of memristor-based combinational logic circuits," *Circuits, Systems, and Signal Processing*, vol. 40, no. 12, pp. 5825–5846, 2021.

[2] S. Smali and Y. Massoud, "Analytic modeling of memristor variability for robust memristor systems

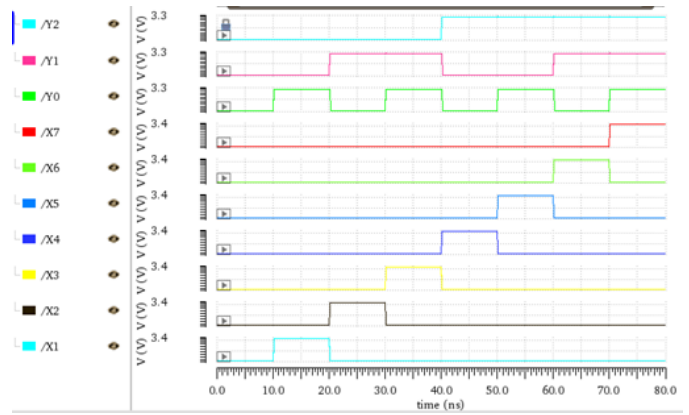


Fig. 15: output of CMOS logic

Logic used for designing	Average Power Dissipated
CMOS	719.8mW
Pseudo NMOS	104.7mW
Memristor based design	80.7mW
Memristor based encoder using PGL technique	61.4mW

Fig. 16: Average Power Table

designs," in *2014 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2014, pp. 794–797.

[3] 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.

K. Mbarek, F. O. Rzig, 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 NanoSystems (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. Kvatinsky, A. Kolodny, 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. 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.

[12] V. A. Slipko and Y. V. Pershin, “Importance of the window function choice for the predictive modeling 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.