SSRG International Journal of Electrical and Electronics Engineering
ISSN: 2348-8379/ https://doi.org/10.14445/23488379/IJEEE-V12I7P119

Volume 12 Issue 7, 264-273, July 2025
© 2025 Seventh Sense Research Group®

Original Article

Fast and Low Power Implementation of Ternary ALU

Nagarathna[1], Srividya B V[2*], Soumya S[3], Deepti Raj[4], Vinod B Durdi[5]

[1, 2, 5]*Department of Electronics and Telecommunication Engineering, Dayananda Sagar College of Engineering, Bengaluru, Karnataka, India*
[3]*Department of Electronics and Communication Engineering, Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal, India.*
[4]*Department of Electronics and Communication Engineering, Sri Venkateshwara College of Engineering, Bengaluru, Karnataka, India*

**Corresponding Author : srividyabv@gmail.com*

Received: 14 May 2025 Revised: 16 June 2025 Accepted: 15 July 2025 Published: 31 July 2025

Abstract *- A ternary ALU is an Arithmetic Logic Unit (ALU) that operates in a ternary (base-3) number system as opposed to the traditional binary (base-2) number system. In contrast to a binary ALU, which only processes 0s and 1s, a ternary ALU (Trit rather than bit) processes three possible values per digit. Every Trit contains more than a bit of information. For some ternary processes, utilizing fewer logic gates can reduce energy consumption. Trits represent the values of 0, 1, or 2 (an imbalanced ternary), making them efficient. A ternary ALU performs arithmetic and logical operations using ternary logic gates and ternary arithmetic circuits. The Ternary ALU is designed using forced-stack multi-threshold MOSFETS. On five-digit ternary data, operations like addition, subtraction, multiplication, and Trit-wise AND, OR, NAND, NOR, XOR, and NOT are carried out. To guarantee the least amount of energy usage, power analysis is done. Nevertheless, it only consumes 81.4% of the power, proving the effectiveness of low-power design techniques. Also, comparing ternary multipliers to binary multipliers, the performance analysis reveals a switching speed gain of roughly 5.35%. Cadence Virtuoso is used in the design and implementation of the ternary logic gates and circuits using 45nm technology.*

Keywords *- Low power, Forced Stack Multi Threshold Transistors (FSMT), Ternary, Decoder, Logic Gates, Arithmetic Unit, Logic Unit.*

1. Introduction

The growing use of Artificial Intelligence (AI) and the Internet of Things (IoT) is generating enormous amounts of data that require processing. It is anticipated that the entire amount of data generated annually will rise and surpass 180 zettabytes by 2025. Consequently, in order to satisfy the demands of data processing in the future, computers with a higher data density are very desirable. As Moore's law approaches its end, it becomes much more challenging to enhance chip performance by merely reducing the transistor feature size [1]. When it comes to having more logic states than its binary equivalent, multi-valued logic is a great choice. Compared to binary processors, ternary processors may be more efficient in computation and storage. Comparing some logical and arithmetic operations to binary logic, fewer transistors are required. Ternary logic lowers heat dissipation and increases energy efficiency by reducing needless state transitions. Arithmetic operations can be handled by ternary ALUs using fewer logic gate levels. For instance, ternary multiplication necessitates fewer partial products than binary multiplication. Low power dissipation and switching speed are the most important parameters of interest.

2. Methodology

Figure 1 depicts the 5-Trit ALU. Along with the logical units such as AND, OR, NAND, NOR, XOR, and NOT, the 5-Trit ALU has an array multiplier, parallel adder/subtractor, and 2:9 decoders. The 9 distinct operational units can be enabled or disabled using the 2:9 decoder's two inputs and nine outputs.

Table 1. Combinations for 5-Trit ALU

Sl.No	2-trit Opcode	Operation
1	00	A.B (TAND)
2	01	$\overline{A.B}$(TNAND)
3	02	A+B (TOR)
4	10	$\overline{A+B}$(TNOR)

5	11	$\bar{A}$(TNOT A)
6	12	$\bar{B}$(TNOT B)
7	20	A⨁B(TXOR)
8	21	A+B/A-B(ADD/SUB)
9	22	A*B (MUL)

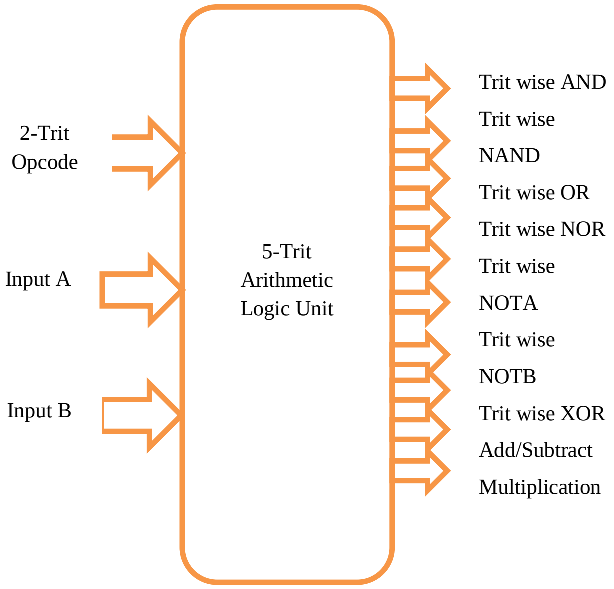


Fig. 1 5-Trit ALU

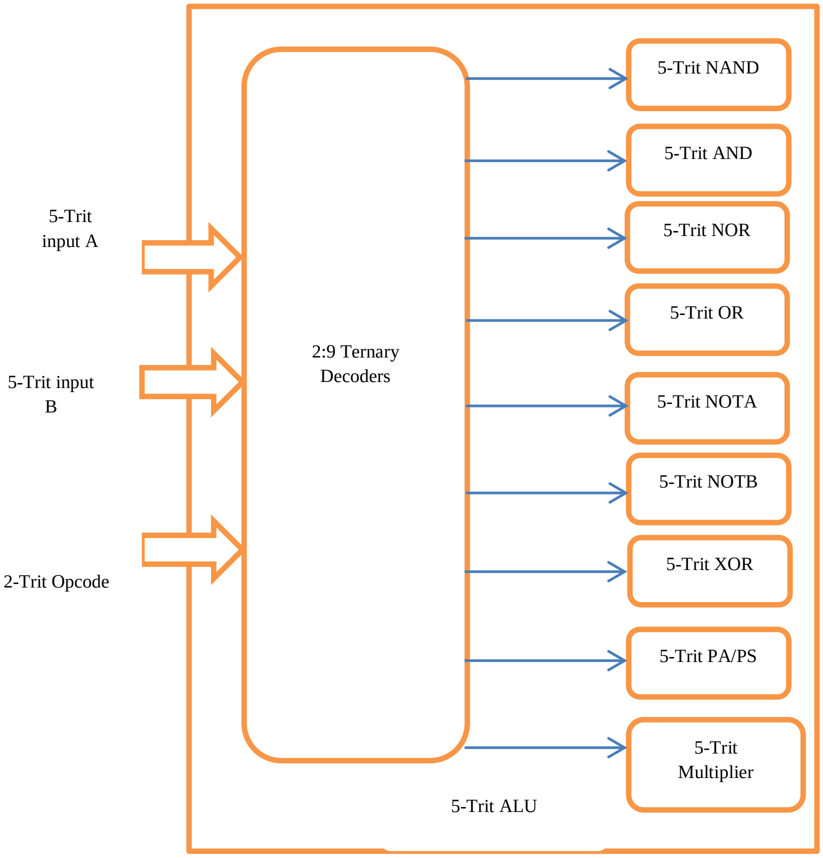


Fig. 2 5-Trit ALU with 2:9 decoder

All these circuits operate on unbalanced Ternary. FSMT Low-power Ternary AND gates are used in the design of each of these units to enable input. The output will be produced by the appropriate unit, taking the inputs into consideration if the enable input is at logic 2.

Inputs are set to 0 via the TAND gates, and the unit will produce 0 output if the enable is set to logic 0. To perform the corresponding operations as indicated in the diagram, the enable pins of each unit are turned on depending on the 2-trit opcode.

The procedures performed for various 2-trit opcode combinations are summarized in Table 1. The schematic for the same is in Figure 3. Each of the units in the ALU is designed to carry out Trit-wise logical operations, which are implemented using low-power Forced Stack Multi-Threshold (FSMT) ternary logic gates.

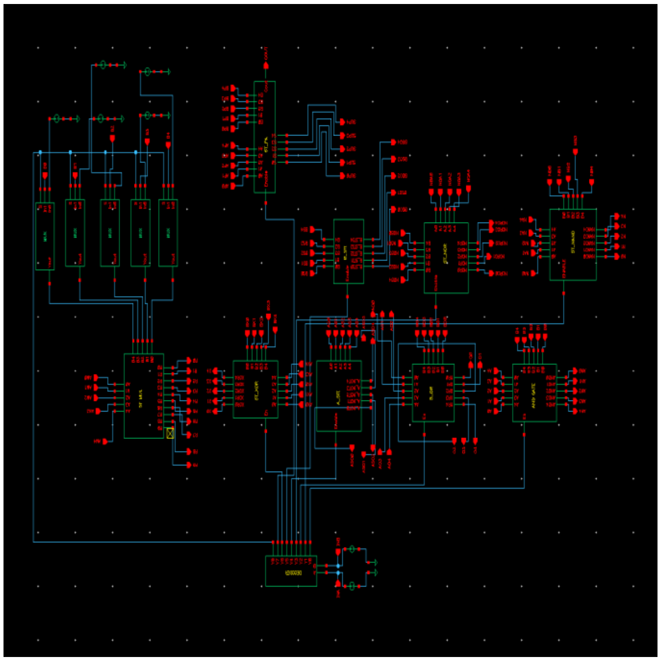


Fig. 3 5-Trit ALU schematic

3. Background Work

Reducing power consumption and chip delays is a digital circuit designer's primary goal. To cut down on delay, the majority of the designers sacrificed power and chip space. Ahmet Unutulmaz [2] suggests combining the ternary logic and threshold logic.

An ALU is implemented with a comparator and arithmetic operators as subparts. The suggested gate may be used to lower latency, power consumption, and chip space in ternary circuits, as the author demonstrated with the use of simulations. The authors of [3] have implemented a Ternary ALU using 180nm and have achieved a substantial reduction in power dissipation and propagation delay. By comparison, ternary representation has 36.90% fewer digits than binary representation [4]. GNRFET transistors operating at various threshold values are used by Badugu Divya Madhuri et al. to build gates, a half adder, and a full adder circuit. Controlling the width of the carbon nanoribbon can change the transistor's Vt value.

These designs exhibit decreased propagation latency, chip area, and power dissipation [5]. When considering the device level, the standby or leakage current is decreased by scaling the depth of the junction (W), the length of the channel (L), and the oxide thickness (Tox) [6-8].

Multiple threshold CMOS power gating, leakage-regulated transistors, forced stacking, sleep stacking, and other methods are some of the methods for minimizing leakage at the circuit level. Hence, the forced stack technique with multi-threshold CMOS transistors is used for implementation.

In our earlier research article entitled “Low-Power VLSI Architecture for Ternary Galois Field”, Multi-threshold transistors with a forced stack network have been designed for all the low-power Ternary logic gates. 56% reduction in power dissipation and an average of 50% reduction in chip area were achieved using the proposed technique. These designed Ternary logic gates are referred to as leaf nodes. Leaf nodes are used for designing Ternary Logic circuits, which are elaborated in the succeeding sections.

4. Implementation

The forced stack and multi-threshold MOSFET approaches are used in the implementation of different ternary gates, including the Inverter, AND, NAND, NOR, OR, and XOR, in order to achieve low-power operation. Leakage power is significantly reduced in both active and standby modes, according to transient and DC power assessments.

These low-power ternary gates have been expanded upon to provide basic arithmetic circuits such as 1-trit multipliers, 2-trit multipliers, half adders, half subtracters, full adders, and full subtractors. In order to demonstrate the potential of ternary computing for processing that uses less energy, these circuits are then combined to implement the arithmetic and logic functions of a Ternary ALU.

4.1. Arithmetic Circuits

4.1.1. 5- Trit Parallel Adder/Subtractor

In this work, a 5-Trit parallel adder is built by cascading 5 full adders using the ripple carry technique as depicted in Figure 4. 5-Trit from A0 to A4 are added in parallely with B0 to B4 along with carry generated from each stage by forwarding it to the next stage, generating sum S0-S4 with S5 as Carry out. The following examples demonstrate the addition of two 5-Trit numbers.

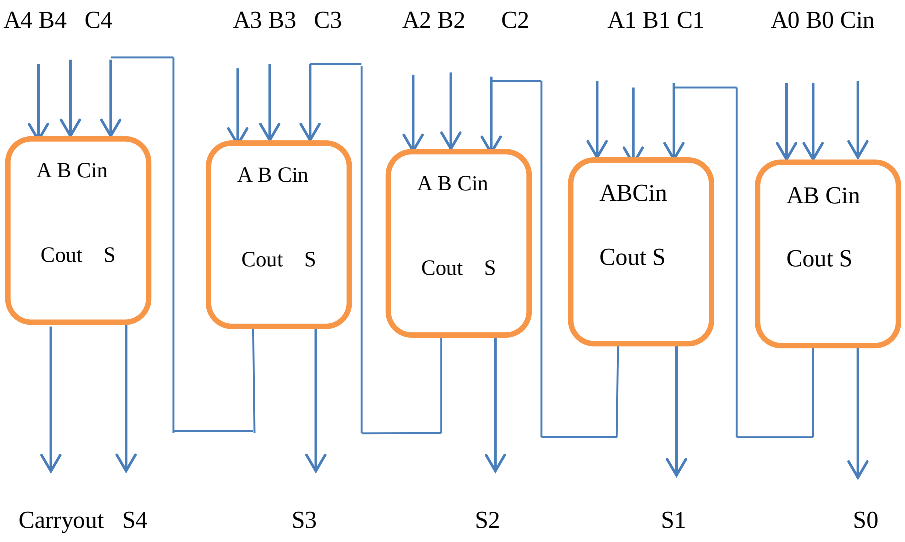


Fig. 4 5-trit parallel adder

As an Illustration, let us consider
Case 1: $(10222)_3$ +$(11012)_3$= $(22011)_3$ carryout=0
Case 2: $(12222)_3$+$(22222)_3$ =$(12221)_3$ carryout= 1

Parallel Subtractor: A Parallel subtractor can be implemented using the parallel adder by deploying the 3’s complement operation. Subtraction of two ternary numbers can be carried out by adding the minued with the 3's complement of the subtrahend. Consider the following example:

Case 1: $(21101)_3$-$(10101)_3$ is equivalent to
21101+3’s complement of (10101)
21101+(complement of (10101)+1)
∵(complement of 10101=12121)
21101+12121+1=11000 and carryout=1
Carryout =1 indicates the result is positive, and ignoring the carry, we get difference=11000 (108d)

Case 2: 10101(91d) - 21101(199d) is equivalent to
10101+3’s complement of (21101)
10101+(complelment of (21101)+1)
∵ (complement of 21101=01121)
10101+01122+1=12000 and carry out =0

Carryout=0 indicates the result is negative, and hence the 3’s complement must be taken to obtain the magnitude of the result.

Difference=3’s complement(12000)= -11000 (-108d)

The schematic of the parallel adder/subtractor is shown. In order to deploy adders as subtractors, one of the inputs (B0B1B2B3B4) is passed thorugh TXOR gates, whose one input is commonly connected to the mode input. TXOR gates act as a controlled inverter based on the mode, and cin is made as 0/ 1 based on addition/ subtraction by using a TAND gate. If mode=0, B⨁0=B and adder performs A+B+Cin(0), thereby producing the sum of A and B. Else if mode=2, B⨁2=$\overline{B}$ and adder performs A+$\overline{B}$ +Cin(1) thereby producing the difference A-B. 2:1 Mux is used to make mode = logic 0 or logic 2 to switch between addition/ subtraction.

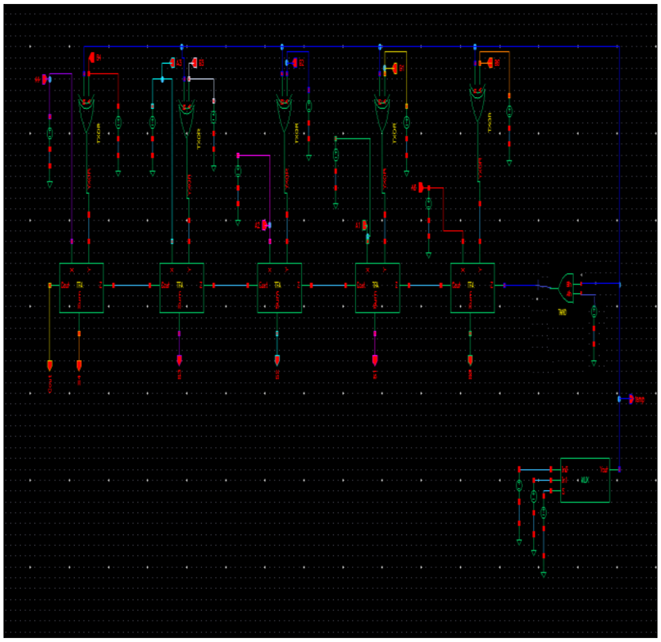


Fig. 5 Schematic representation of 5-trit parallel adder /subtractor

4.1.2: 5- Trit Array Multiplier

5-Trit multiplier provides a 10 Trit product for two 5-trits of data. Table 2 shows the samples of multiplication of pairs of 5 Trit value and their product. This is constructed using 25

1-trit multipliers and 17 adders. Partial products obtained from each of the multipliers are added with other partial products of the same weight (3^i) along with the carry generated from the previous stage addition, using either a Ternary Half Adder or Ternary Full Adder.

Table 2. Samples of 5-trit ternary multiplication

A	B	Product(PP0-PP9)
21201	12011	1110122211
22222	222222	2222100001
11111	11111	0202002021
10101	01201	0012202001
20101	20000	1102020000

5-Trit multiplier is implemented using 2 trit and 1 trit multipliers using the following expressions. Consider MP1 and MR1 as lower two trits of multiplicand (A1A0) and multiplier (B1B0), MP2 and MR2 as higher two trits of multiplicand (A3A2) and multiplier (B3B2) and MP3 and MR3 as most significant trit of multiplicand(A4) and multiplier(B4). 2 trit and 1 trit are multipliers used to obtain following partial products:

$MP_1X\ MR_1{=}P_3P_2P_1P_0$
$MP_1X\ MR_2{=}P_7P_6P_5P_4$
$MP_1X\ MR_3{=}\ P_{10}P_9P_8$
$MP_2X\ MR_1{=}P_{14}P_{13}P_{12}P_{11}$
$MP_2X\ MR_2{=}\ P_{18}P_{17}P_{16}P_{15}$
$MP_2X\ MR_2{=}P_{21}P_{20}P_{19}$
$MP_3X\ MR_1{=}\ P_{24}P_{23}P_{22}$
$MP_3X\ MR_2{=}\ P_{27}P_{26}P_{25}$
$MP_3X\ MR_3{=}\ P_{29}P_{28}$

Final products are obtained by adding intermediate partial products as per the following expressions:

$R_0{=}\ P_0$
$R_1{=}\ P_1$
$R_2{=}\ P_2 + P_4 + P_{11}$
$R_3{=}\ P_3 + P_5 + P_{12}{+}CY$
$R_4{=}\ P_6{+}\ P_8 + P_{13}{+}\ P_{15} + P_{22}{+}CY$
$R_5{=}\ P_7 + P_9 + P_{14}{+}\ P_{16}{+}\ P_{23}{+}CY$
$R_6{=}\ P_{10} + P_{17}{+}\ P_{19}{+}\ P_{24}{+}CY$
$R_7{=}\ P_{18} + P_{20} + P_{26}{+}CY$
$R_8{=}\ P_{21}{+}\ P_{27} + P_{28}{+}CY$
$R_9{=}\ P_{29}{+}CY$

The schematic representation of the multiplier is shown in Figure 6.

One of the circuits in an Arithmetic Logic Unit (ALU) that uses the greatest power and time is the multiplier. Multiplication is a computer bottleneck because it requires numerous partial product generations, additions, and shifts, in contrast to more straightforward operations such as addition, subtraction, and bitwise logic. 5-Trit multiplier is implemented using 2 trit and 1 trit multipliers. Thus, calculations pertaining to power estimation and delay are performed for the 2-Trit Multiplier. A 3-bit binary multiplier's performance is also evaluated in terms of speed, power dissipation, and result correctness. The results obtained by considering the same parameters are tabulated in Table 3.

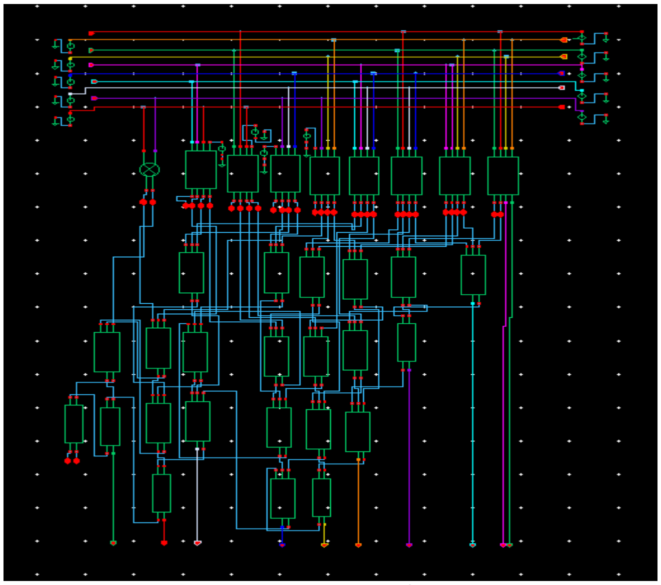


Fig. 6 Schematic representation of 5-trit multiplier

Table 3. Comparative analysis between binary and ternary multipliers

Sl. No.	Parameter	Binary Multiplier (3-bit)	Ternary Multiplier (2-Trit)
1	DC Power	789.266 µW	642.463µW
2	Transient Power	1221.437 µW	994.249µW
3	Multiplication Speed	105.728ns	100.067ns

Because there are more switching operations in a binary multiplier, power consumption is higher; however, ternary multipliers dissipate less power due to fewer state transitions. While ternary multipliers have fewer operations and therefore a faster computation, binary multipliers have a higher latency since they involve more partial products.

4.2. Logic Circuits

To implement Trit-wise logical operations, respective logical gates are utilized. The working of the low-power ternary logic gates has been functionally verified. These logic gates serve as leaf nodes while designing the logic unit of the ALU. Hence, only the logic circuits enabled by AND gates are depicted in the following section.

4.2.1. 5-Trit AND

The schematic of the 5-Trit AND unit is as in Figure 7. It uses five 3-input AND gates, with one of the inputs of each gate tied to the enable signal.

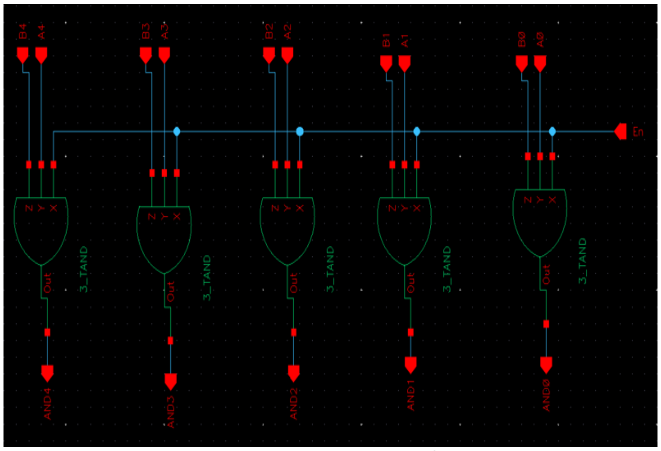


Fig. 7 Schematic representation of 5-trit AND

4.2.2. 5-Trit NAND

The 5-trit NAND unit is as in Figure 8, with every input passed through a set of AND gates, whose one input is connected to the enable.

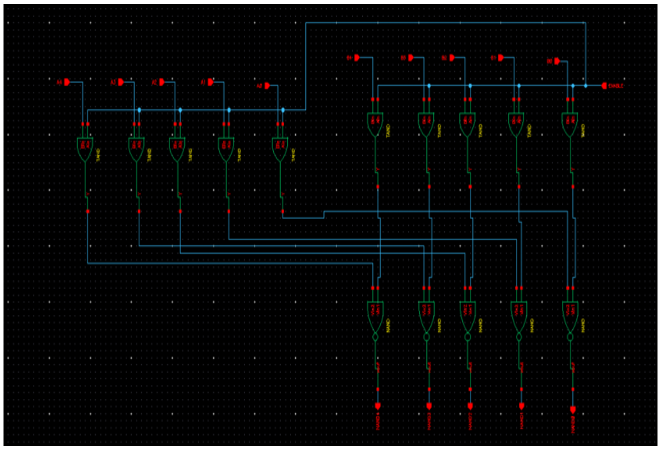


Fig. 8 Schematic representation of 5-trit NAND

4.2.3. 5-Trit OR

Figure 9 depicts the schematic representation of 5-Trit OR with an enable signal passed through TAND gates.

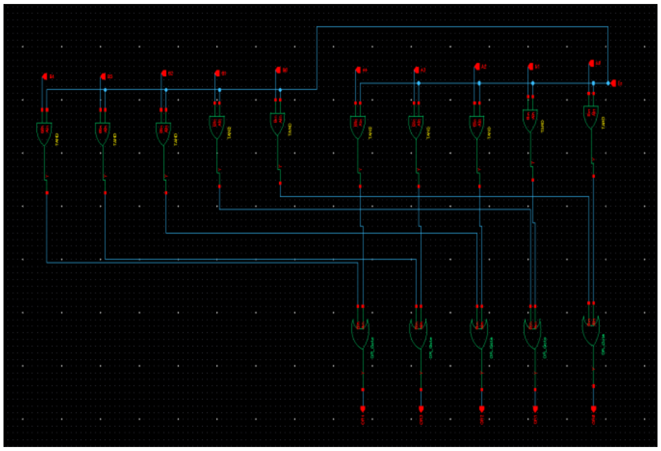


Fig. 9 Schematic representation of 5-trit OR

4.2.4: 5-Trit NOR

Figure 10 depicts the schematic representation of 5-Trit NOR with an enable signal passed through TAND gates.

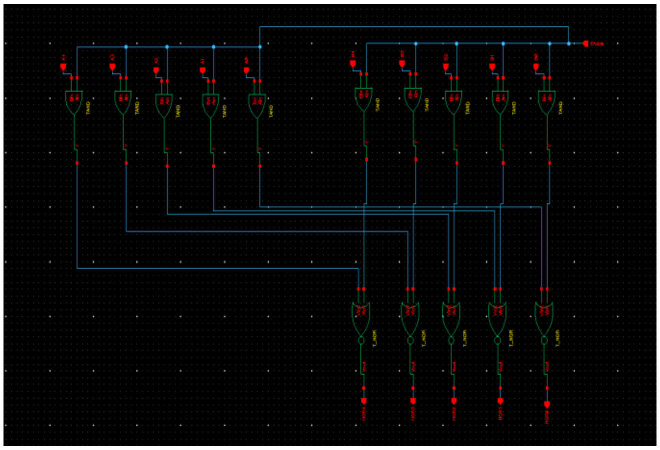


Fig. 10 Schematic representation of 5-trit NOR

4.2.5. 5-Trit NOT

Figures 11and 12 show 5 trit NOT and 5 trit XOR logical units with an enable signal passed through TAND gates.

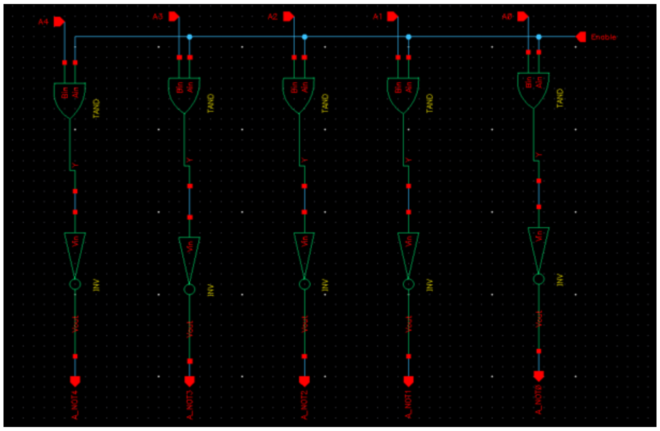


Fig. 11 Schematic representation of 5-trit NOT

4.2.6. 5-Trit XOR

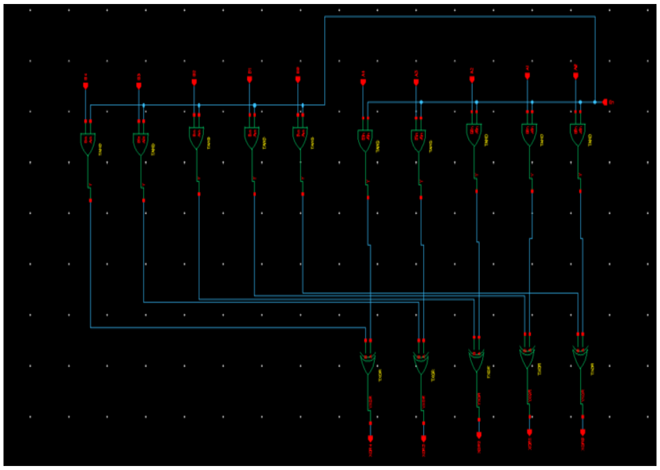


Fig. 12 Schematic representation of 5-trit XOR

5. Results and Discussion

Results of Decoder: Figure 13 shows the results of the decoder when the inputs A and B are at logic 1(0.9V) and logic 0(0V), as per the function table, when AB=10, Y3 will be at logic 2, and the same is depicted in Figure 13.

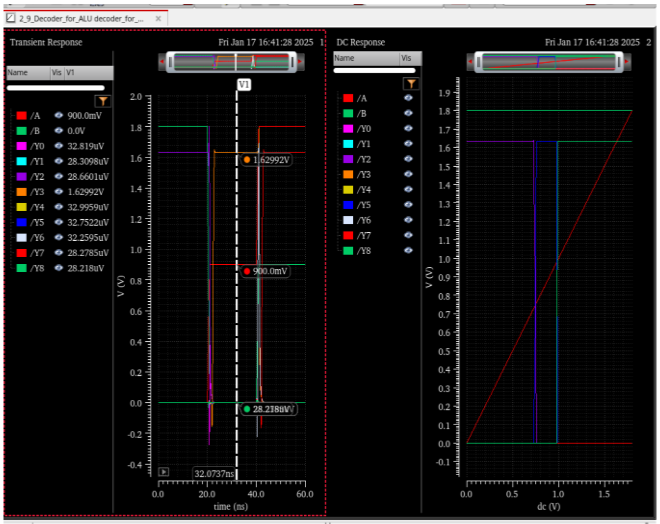


Fig. 13 Results of 2:9 decoder

Figure 14 depicts the T-ALU output when the opcode (INA,INB) is at “12”, and the B inputs are at “11111” by generating a 5-trit complemented output of B as “11111”.

This is as per the table. Likewise, all operations of the 5-trit ALU are verified for different opcodes as per the table.

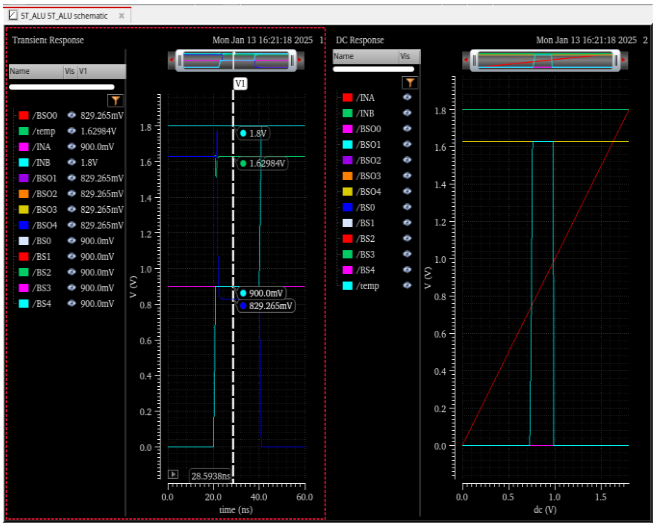


Fig. 14 Results of ALU for a specific opcode

5.1. Results of Parallel Adder/Subtractor

Functioning of the 5-trit parallel adder/subtractor is verified for some of the combinations of 5 Trit inputs A and B. Figure 15(a) shows the sum of 5-Trit data 02111(A0-A4) and 10000(B0-B4) as 12111(S0-S4) with Cout=0. Addition is selected by making mode(temp)=0.

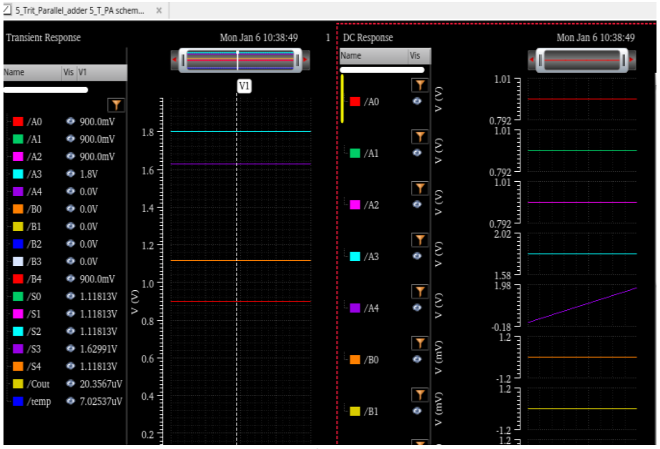


Fig. 15(a) Results of addition (with cout=0)

Figure 15(b) shows the result of the addition S=02111 and carry out=1, for A=22111, B=10000 combination.

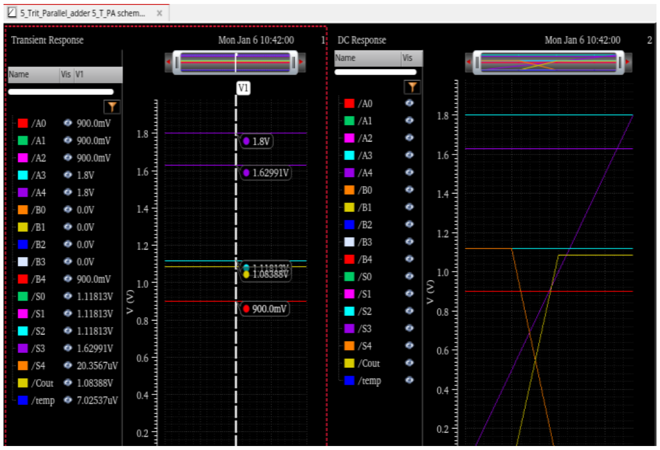


Fig. 15(b) Results of addition (with cout=1)

Figure 15(c) shows the result of A-B for the combination 21101-10101, generating a a difference of 11000 with Cout=0, indicating a positive result. Subtraction can be selected by forcing Mode(temp)=logic 2.

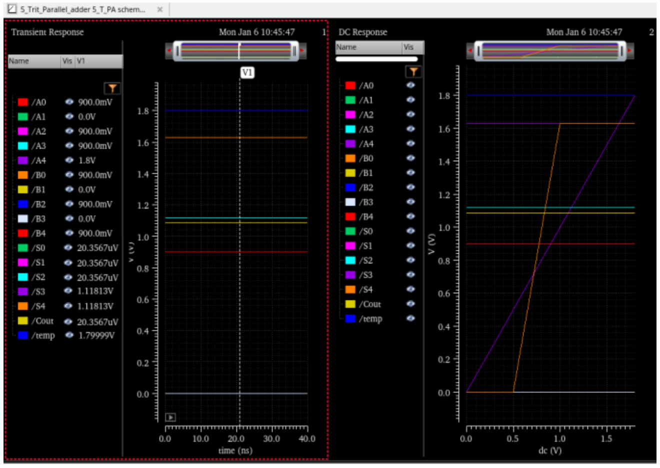


Fig. 15(c) Results of subtraction (result is +ve)

Figure 15(d) shows the result of A-B for the combination 10101 -21101, generating a difference of 12000 with Cout=logic 1, indicating a negative result. The magnitude of this can be obtained by taking the 3's complement, which is 11000.

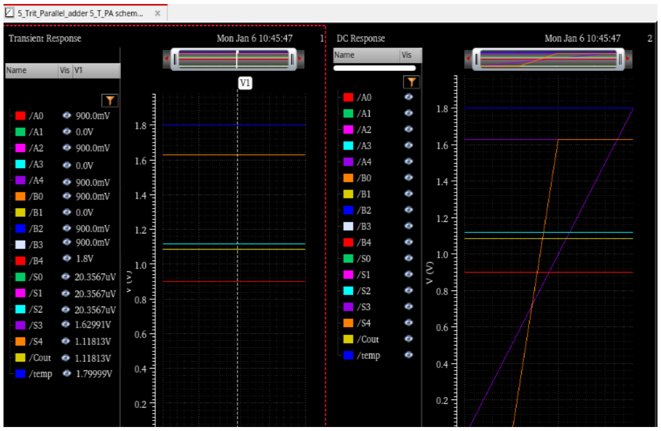
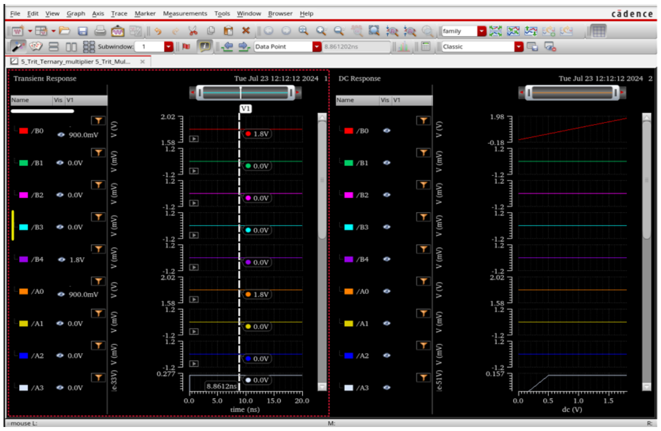
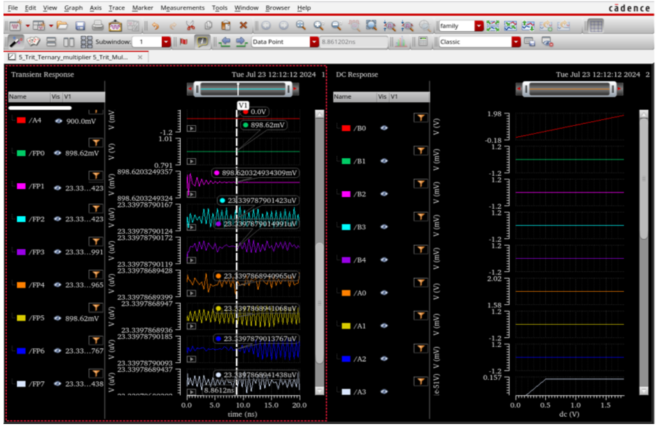


Fig. 15(d) Results of subtraction

5.2. Results of Array Multiplier

The following figures show the product for inputs $20001_{(3)}$ and $10001_{(3)}$, resulting in the product as $0200100001_{(3)}$. Response of the multiplier is verified for some of the sample inputs.





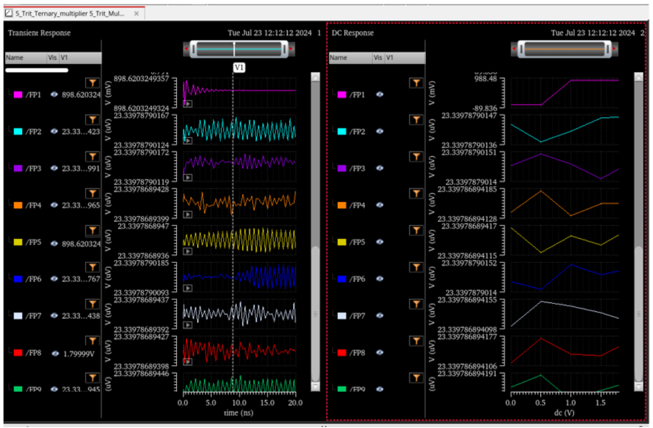


Fig. 16 Results of array multiplier

The power analysis of binary and ternary multipliers is carried out. From the results, it is observed that the 2-Trit ternary multiplier consumes only 81.4% of the total power consumed by the 3-bit binary multiplier.

5.3. Results of Logic Unit

Each of the logical and arithmetic units is tested and verified individually for different combinations of 5 trit inputs. Figure 16 shows the result of the 5-Tritwise AND unit, with enable input at 1.8V (logic 2) and inputs A and B at "22222", thereby producing the result of "22222". Figure 18 shows the result is "00000" when the enable is set at 0V (logic 0). Similarly, functional verification of all the logical gates is carried out for enable=0 and enable=1.

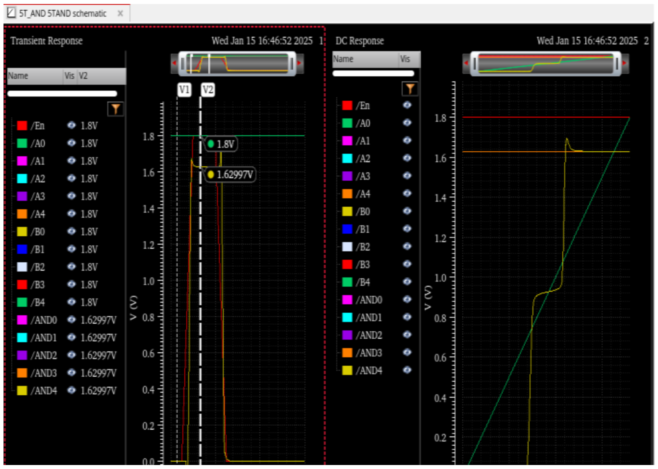


Fig. 17 Results of 5-trit AND with enable=1

Figure 19 depicts the T-ALU output of Ternary NOT, when the opcode (INA,INB) is at "12" and the B inputs are at "11111" by generating a 5-trit complemented output of B as "11111". This is as per the table. Likewise, all the operations of 5-trit ALU are verified for different opcodes as per the table. The T-ALU's results are validated for various input and opcode combinations. Power analysis is done on 5-trit Ternary AND OR units and 8-bit binary AND, OR units. The low-power Ternary logical units that are suggested lead to five-trit logical units' power consumption, which is analyzed and contrasted with that of matching eight-bit logical units. It has been observed that five trit AND logical units and five trit OR units use 57.41µW and 137.85µW of power, respectively. Likewise, the 8-bit AND and OR logical units use 1.277 and 1.237 mW of power, respectively.

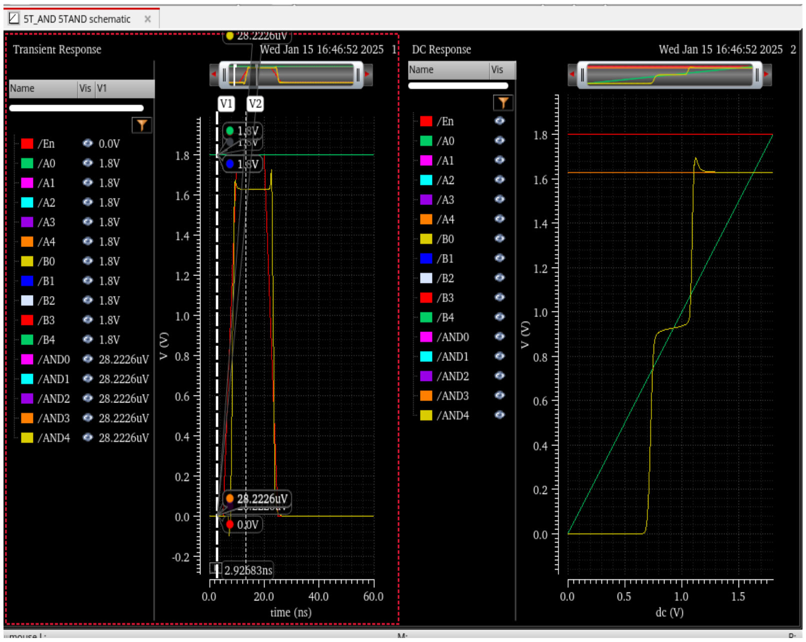


Fig. 18 Simulation results of 5-trit AND with enable=0

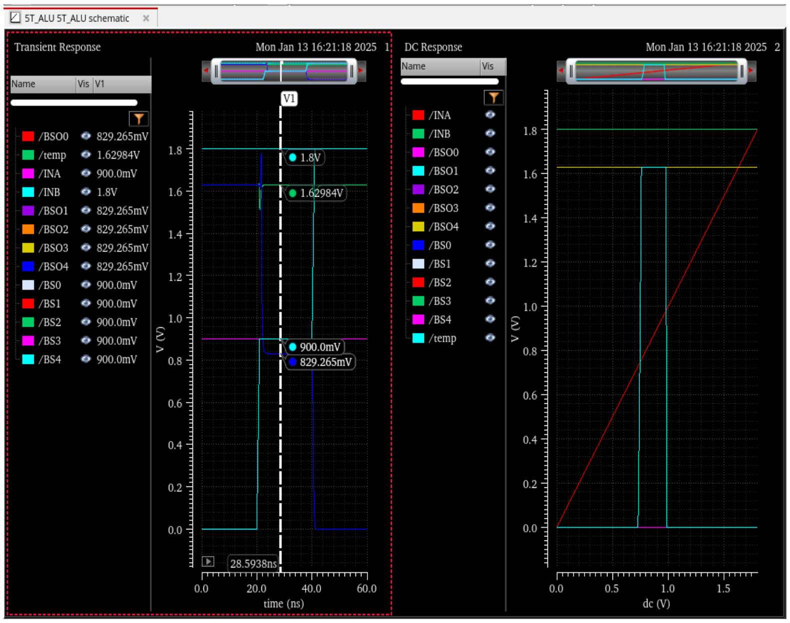


Fig. 19 Simulation results of TNOT

6. Conclusion

The logic levels of the ternary number system are 0, 1, and 2. One trit is equivalent to 1.58 bits of information. 5-Trit ALU typically has 3^5=243 distinct values. Despite using a ternary system (base-3), the 2-trit multiplier can handle nine different values (3^2), while a 3-bit binary multiplier can handle eight different values (2^3). However, it only uses 81.4% of the power, demonstrating how well low-power design strategies work. By employing forced stack and multi-threshold MOSFET techniques, ternary gates are implemented, resulting in a power reduction. In both active and standby phases, these methods are very good at lowering leakage power. The performance analysis of ternary multipliers shows a switching speed improvement of approximately 5.35% compared to binary multipliers.

References

[1] Thomas N. Theis, and H.-S. Philip Wong, “The End of Moore’s Law: A New Beginning for Information Technology,” *Computing in Science & Engineering*, vol. 19, no. 2, pp. 41-50, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[2] Ahmet Unutulmaz, and Cem Ünsalan, “Implementation and Applications of a Ternary Threshold Logic Gate,” *Circuits, Systems, and Signal Processing*, vol. 43, no. 2, pp. 1192-1207, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[3] Guangchao Zhao et al., “Efficient Ternary Logic Circuits Optimized by Ternary Arithmetic Algorithms,” *IEEE Transactions on Emerging Topics in Computing*, vol. 12, no. 3, pp. 826-839, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[4] Xiao-Yuan Wang et al., “A Review on the Design of Ternary Logic Circuits,” *Chinese Physics B*, vol. 30, no. 12, pp. 1- 12, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[5] Zarin Tasnim Sandhie, Farid Uddin Ahmed, and Masud H. Chowdhury, “Design of Ternary Logic and Arithmetic Circuits Using GNRFET,” *IEEE Open Journal of Nanotechnology*, vol. 1, pp. 77-87, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[6] A. Steegen et al., “65nm CMOS Technology for Low Power Applications,” *IEEE International Electron Devices Meeting, 2005. IEDM Technical Digest*, Washington, DC, USA, pp. 64-67, 2005. [CrossRef] [Google Scholar] [Publisher Link]

[7] S. Zhao et al., “Transistor Optimization for Leakage Power Management in a 65nm CMOS Technology for Wireless and Mobile Applications,” *Digest of Technical Papers. 2004 Symposium on VLSI Technology*, Honolulu, HI, USA, pp. 14-15, 2004. [CrossRef] [Google Scholar] [Publisher Link]

[8] K. Koh et al., “Highly Manufacturable 100nm 6T Low Power SRAM with Single Poly-Si Gate Technology,” *2003 International Symposium on VLSI Technology, Systems and Applications. Proceedings of Technical Papers. (IEEE Cat. No.03TH8672)*, Hsinchu, Taiwan, pp. 64-67, 2003. [CrossRef] [Google Scholar] [Publisher Link]

[9] A.P. Dhande, V.T. Ingole, and V.R. Ghiye, *Ternary Digital System: Concepts and Applications*, SM Group, SM Medical Technologies Private Limited, pp. 1-131, 2014. [Google Scholar]

[10] Furqan Zahoor et al., “Design Implementations of Ternary Logic Systems: A Critical Review,” *Results in Engineering*, vol. 23, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[11] Sneh Lata Murotiya, and Anu Gupta, “Design of CNTFET-Based 2-Bit Ternary ALU for Nanoelectronics,” *International Journal of Electronics*, vol. 101, no. 9, pp. 1244-1257, 2014. [CrossRef] [Google Scholar] [Publisher Link]

[12] A.P. Dhande, and V.T. Ingole, “Design and Implementation of 2 Bit Ternary ALU Slice,” *SETIT 2005 3rd International Conference: Sciences of Electronic, Technologies of Information and Telecommunications*, Tunisia, vol. 17, 2005. [Google Scholar] [Publisher Link]

[13] Chetan Kumar Vudadha, and MB Srinivas, “Design Methodologies for Ternary Logic Circuits,” *2018 IEEE 48th International Symposium on Multiple-Valued Logic (ISMVL)*, Linz, Austria, pp. 192-197, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[14] Vijay Kumar Sharma, “A Survey of Leakage Reduction Techniques in CMOS Digital Circuits for Nanoscale Regime,” *Australian Journal of Electrical and Electronics Engineering*, vol. 18, no. 4, pp. 217-236, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[15] Malachy Eaton, “Design and Construction of a Balanced Ternary ALU with Potential Future Cybernetic Intelligent Systems Applications,” *2012 IEEE 11th International Conference on Cybernetic Intelligent Systems (CIS)*, Limerick, Ireland, pp. 30-35, 2012. [CrossRef] [Google Scholar] [Publisher Link]