DESIGN AND IMPLEMENTATIONOF MULTIPLIER USING COMPRESSOR ARCHITECTUREWITH TERNARY LOGIC

Ganga Maheswari S.[1*] and Reddy Hemantha G.[2]

[1]Department of Electronics and Communication Engineering, Madanapalle Institute of Technology and Science(Autonomous),Madanapalle, Andhra Pradesh, 517325,India

[2]Department of Electronics and Communication Engineering, Madanapalle Institute of Technology andScience (Autonomous), Madanapalle, Andhra Pradesh, 517325, India

*Corresponding author: **Ganga Maheswari S**

Abstract

The design and implementation of a high-performance multiplier based on 5:2 compressors using ternary logic are presented in this study. Multipliers are crucial arithmetic units in digital systems that have a significant impact on the speed, power consumption, and efficiency of processors and signal-processing devices. Conventional binary multiplier systems that employ two logic levels are limited in terms of growing gate count, connectivity complexity, and power consumption as performance requirements continue to rise. Research into multi-valued logic systems to boost computer efficiency is motivated by the challenges in this field. This work blends three distinct logic levels in ternary logic to reduce the number of logic gates and signal connections while increasing information density. Partial product reduction, an essential role in fast multipliers, is efficiently carried out by the 5:2 compressor, which is intended for ternary operations. The devised ternary multiplier architecture shows much faster and more power-efficient arithmetic calculations because of the diminished propagation path lengths without affecting accuracy. To further boost speed and scalability, the multiplier architecture incorporates a hierarchical reduction tree with multiple ternary compressors. This ternary multiplier architecture employs Matrix Laboratory(MATLAB) simulations to simulate the proposed ternary multiplier. The major performance parameters, such as delay, area, and power, are evaluated and compared to the traditional binary systems. The results validate that the ternary compressor architecture can reduce circuit complexity and power dissipation without affecting accuracy, hence has potential use in digital technologies that require lower power dissipation and faster speed.

Keywords: Arithmetic Circuit Design, High-Speed Arithmetic Units, Low-Power Circuits, MultiValued Logic, Power-Efficient Computing

How to cite this article: Ganga Maheswari S, Reddy Hemantha G. Design and Implementation of Multiplier Using Compressor Architecture with Ternary Logic. Int J Drug Deliv Technol. 2026;16(64s):597-603. DOI: 10.25258/ijddt.16.64s.58

1. Introduction

Multipliers are essential components in digital signal processing, cryptography, and high-precision mathematical computations. Their efficiency strongly affects overall system performance, particularly in terms of speed, power consumption, and computational complexity. Conventional binary logic-based multipliers face several limitations, including high power dissipation, increased circuit complexity, and propagation delays due to extensive interconnections in modern designs. These issues become more significant with the continuous advancement of digital systems.

To overcome these challenges, ternary logic has been introduced as a promising alternative. Unlike binary logic, ternary logic uses three logic states instead of two, which helps reduce circuit complexity, interconnect overhead, and power consumption. This makes it suitable for designing efficient multipliers for high-speed and low-power applications [1].

Efficient partial product reduction is a key factor in multiplier design. In this context, ternary-optimized 5:2 compressor units are used to improve performance. These compressors effectively combine multiple partial products into fewer outputs, reducing logic depth and improving computational speed. As a result, they contribute to reduced delay, smaller silicon area, and lower energy consumption in advanced digital systems [2].

However, traditional compressor designs based on basic logic gates still suffer from higher power and area usage. To address this, majority logic has been proposed as an efficient alternative. Majority logic simplifies circuit operations by producing outputs based on the majority of input values, thereby reducing gate complexity and improving speed and energy efficiency [3][4].

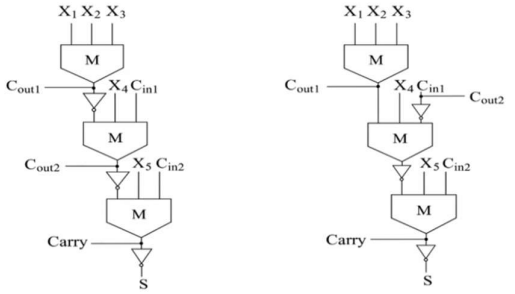
Approximate computing has also emerged as an important technique for improving efficiency by allowing controlled inaccuracies in computation. It is widely used in applications such as image processing, machine learning, and digital signal processing, where exact precision is not always required [5]. Majority logic further supports energy-efficient circuit design and is particularly useful in emerging nanotechnology systems like quantum dot cellular automata, where conventional CMOS designs are less effective. In future systems,

combining majority logic with approximate computing is expected to enable highly efficient, low-power architectures [6]. This work proposes a ternary multiplier using 5:2 compressors and ternary logic gates. MATLAB simulations validate the design and demonstrate improved efficiency with reduced hardware complexity.

2. Materials and Methods

2.1 Design of Efficient Approximate 5:2 Compressors and Multipliers

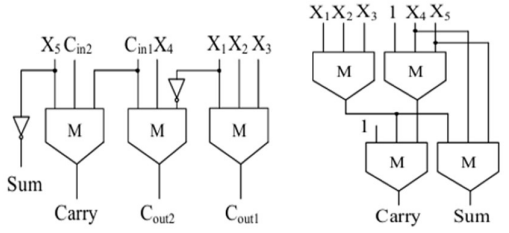
Majority logic-based approximate 5:2 compressors are designed to achieve lower power, reduced area, and decreased delay with acceptable accuracy. The MLAC-I and MLAC-II architectures improve the overall performance and are suitable for efficient arithmetic operations in modern low-power applications.



1(A) 1(B)

Figure 1(A) MLAC-I and 1(B) MLAC-IIdiagrammatic representations of MLAC [7].

The ternary full adder adds three inputs X_1, X_2, and X_3 (carry-in) and produces Sum (S) and Carry-out (Cout). Instead of conventional logic gates, it is implemented using multiplexers. Multiplexers act as switching devices that select one input based on control signals. In this design, a multi-level multiplexer structure is used to simplify ternary logic operations. The left-side multiplexer network generates intermediate signals for the Sum output, while the final multiplexer produces the Sum value. The right-side structure computes intermediate terms to generate the Carry-out output. This multiplexer-based implementation reduces circuit complexity, improves efficiency, and provides a suitable design approach for ternary arithmetic systems.



2(A) 2(B)

Figure 2 Displays the 5:2 MLAC as MLAC-III in 2(A) and MLAC-IV in 2(B) [7].

The output is generated as follows, as indicated in the illustration below:The multiplexers generate the logic operations involving the inputs X_1, X_2, X_3, X_4, and X_5, as well as the intermediate signals Cin_1, Cin_2, $Cout_1$, and $Cout_2$. The addition process can be further divided in smaller, more manageable chunks using the intermediate signals.The multiplexers on the right side of the adder are quickly connected to generate the final outputs [8]. The carry output is produced by one multiplexer, while the total output is delivered by another multiplexer that combines the necessary logic signals.

2.2 Ternary Logic Multipliers Using Advanced Adders and Compressors

Ternary multipliers are important in multi-valued logic; however, they suffer from high power consumption, increased delay, and complex circuit implementation. Conventional ternary adders face challenges such as carry propagation, complex logic structures, and higher transistor requirements. To overcome these limitations, optimized ternary adders and compressor-based architectures are developed to reduce delay, area, and power consumption. Although the lack of efficient ternary standard cells remains a challenge, improved compressor and adder designs can enhance the speed, energy efficiency, and practicality of ternary multipliers for future computing applications.

2.3Ternary logic and gates

Ternary logic is a multi-valued logic system that uses three states (0, 1, 2), in contrast to binary logic, which uses two states. Consequently, the information density increases and the circuit size lowers. Like binary gates, ternary gates are the fundamental building blocks of ternary circuits. Examples include ternary NOT, AND, OR, NAND, and majority gates. Complex circuits like multipliers and adders are built using these gates [9]. Ternary logic and gates enable higher efficiency and fewer connections, but circuit complexity and power constraints limit their practical application.

Table 1 Logic symbols of ternary logic[9]

VOLTAGE LEVEL	LOGIC LEVEL
0	0
½ VDD	1
VDD	2

Ternary logic uses three voltage levels—0, 1, and 2 to represent low, neutral, and high states. It helps reduce processing time, chip size, and routing complexity while improving data transfer efficiency. In this system, both inputs and outputs take values from {0, 1, 2}. Binary 0 is mapped to 0, binary 1 is mapped to 2, and 1 represents an intermediate state. These states are defined using supply voltages: Ground (0), VDD/2 (1), and VDD (2).

2 ; if x is true
v(x)= {1;if x is perhaps true
Perhaps false
0; if x is false

The function v(x) represents multi-valued logic by mapping a variable x into discrete values. Unlike binary logic, ternary logic includes an intermediate state, enabling more flexible representation of uncertainty. In this system, v(x) = 2 represents true, v(x) = 0 represents false, and v(x) = 1 represents an intermediate or uncertain state [10]. This additional state allows digital systems to handle complex conditions more efficiently. Ternary logic is particularly useful in approximate computing, where controlled uncertainty is acceptable to improve performance and efficiency [11]. It also supports alternative arithmetic structures that enhance data representation and processing compared to traditional binary logic.

$$Y0=C0_{(x)}=\begin{cases}2, \square = 0\\ 0, \square \neq 0\end{cases} \quad (1)$$

$$Y1=C1_{(x)}=x=2-x \quad (2)$$

$$Y2=C2_{(x)}=\begin{cases}2, \square\square\ \square < 2\\ 0, \square\square\ \square = 2\end{cases} \quad (3)$$

The output functions Y0, Y1, and Y2 represent multi-valued logical operations in ternary systems. The function Y0 identifies the zero-logic state, Y1 = 2 − x provides a complementary inverse response, and Y2 performs threshold-based switching by becoming inactive at x = 2. These functions enable efficient detection, representation, and control of ternary logic states [12]. Together, they enhance ternary arithmetic and multi-valued logic systems by improving state transitions and logical processing in complex digital applications.

Table 2 Truth table of STI, PTI, NTI[13]

INPUT X	STI	PTI	NTI
0	2	2	2
1	1	2	0
200	0		

Ternary inverters operate on inputs X = {0, 1, 2} using three types: STI, PTI, and NTI.The STI (Standard Ternary Inverter) performs symmetric inversion by mapping 0 → 2, 1 → 1, and 2 → 0. The PTI (Positive Ternary Inverter) produces 2 for inputs 0 and 1, and 0 for input 2. The NTI (Negative Ternary Inverter) maps 0 → 2 and outputs 0 for inputs 1 and 2 [14]. These inverters are fundamental building blocks in ternary and multi-valued logic circuits. Ternary NAND and NOR gates are multi-input logic functions that require multiple input combinations, and their behaviour is defined through mathematical expressions and corresponding truth tables [15].

$$YNand = [\min \{X_1, X_2\}]' = 2-\min(X_1, X_2) \quad (4)$$

$$YNor = [\max \{X_1, X_2\}]' = 2-\max(X_1, X_2) \quad (5)$$

Table 3 Truth table of TNAND and TNOR gates[16]

X_1X_2	YNAND	YNOR
0 0	2	2
0 1	2	1
0 2	2	0
1 0	2	1
1 1	1	1
1 2	1	0
2 0	2	0
2 1	1	0
2 2	0	0

Ternary NAND (YNAND) and ternary NOR (YNOR) gates extend conventional logic by supporting three input states. YNAND maintains a higher output level and decreases only when both inputs reach the maximum state, whereas YNOR produces a high output only when both inputs are zero [16]. These characteristics make ternary gates suitable for efficient and compact multi-valued digital system designs.

2.4The design of the 5:2 Compressor Module

As seen in Figure 3. The 5:2 compressor is necessary for precise and quick multiplication. This stage uses optimal reduction strategies to further limit the number of partial products produced during multiplication.

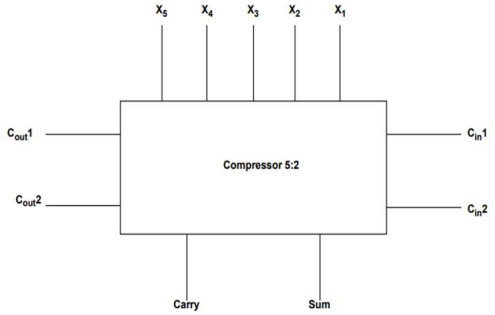


Figure 3 Blockdiagram of the 5:2 compressor [17]

A 5:2 compressor consists of five equal-weight inputs (X_1, X_2, X_3, X_4, and X_5), two carry-in signals (Cin_1 and Cin_2), and produces a Sum, two carry-out signals ($Cout_1$ and $Cout_2$), and a Carry output. The carry inputs are received from the previous stage, while the carry outputs are propagated to the next higher bit position [18]. Regardless of architectural variations, all 5:2 compressors follow the same basic mathematical principle for arithmetic computation.

$$X_1+X_2+X_3+X_4+X_5+Cin_1+Cin_2 = Sum +2(Carry + Cout_1+Cout_2) \quad (6)$$

2.5Design-1: Exact 5:2 compressor using a full adder

The accurate 5:2 compressor is constructed by cascading three Full Adder (FA) units. In this architecture, carry signals are propagated to the next higher-order bit position, while maintaining the required arithmetic operation. Design-1 implements the accurate 5:2 compression using Full Adders, and its output behavior can be represented by the corresponding mathematical equations.

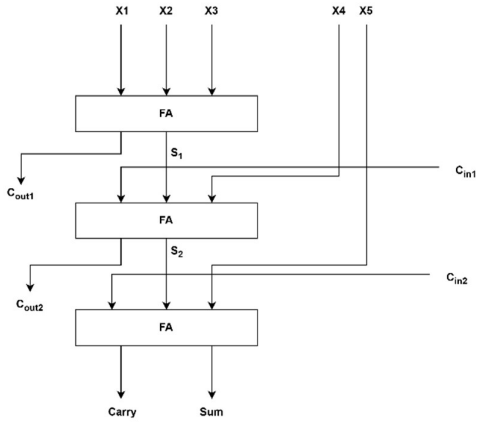


Figure 4 Design-1 exact 5:2 compressor [19].

Sum= $X_1 \oplus X_2 \oplus X_3 \oplus X_4 \oplus X_5 \oplus Cin_1 \oplus Cin_2$(7)

Carry= $X_1 \oplus X_2 \oplus X_3 \oplus X_4 \oplus X_5 \oplus Cin_1\ Cin_2 + (X_1 \oplus X_2 \oplus X_3 \oplus X_4 \oplus X_5 \oplus Cin_1)\ X_5$(8)

$Cout1=(X_1 \oplus X_2)$

$X_3+(X_1 \square X_2)\ X_1$ (9)

Cout2= $X_1 \oplus X_2 \oplus X_3 \oplus X_4\ Cin_1 + (X_1 \oplus X_2 \oplus X_3 \oplus X_4)\ X_5$(10)

2.6Inexact 5:2 Compressor

The graphic depicts the internal logic configuration of an estimated 5:2 compressor. The Sum and Carry outputs are generated by processing the five major inputs (A_1, A_2, A_3, A_4, and A_5) using simple logic circuits and a multiplexer.

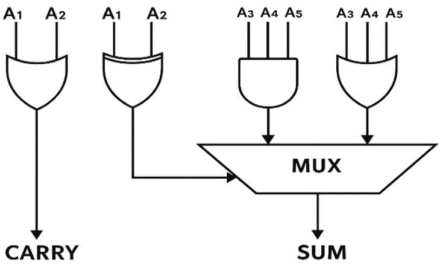


Figure5 Approximate 5:2 compressor [20].

The method reduces latency, power consumption, and device footprint by using approximation logic gates for full-precision adders.Although it introduces a minor degree of inaccuracy, this approach is quite helpful for applications that can tolerate small errors, such as machine learning algorithms and image and signal processing [21].

The following Boolean equations are used to calculate the SUM and CARRY signals:

SUM = $(A_1 \square A_2)\ A_3A_4A_5 + (A_1+A_2)\ (A_3+A_4+A_5)$ (11)

CARRY = (A_1+A_2) (12)

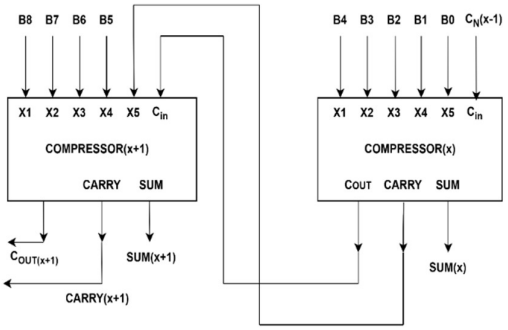
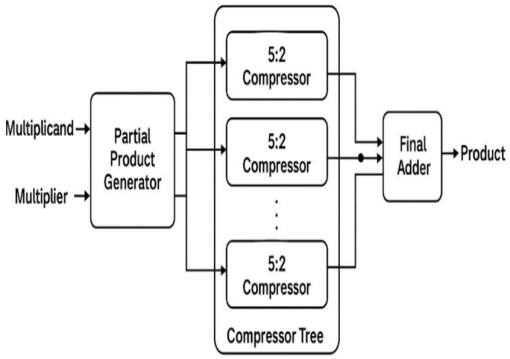


Figure6 Approximate 5:2 compressor chain [22].



The cascaded 5:2 compressor architecture consists of two stages, Compressor (x) and Compressor (x+1), for efficient multi-operand addition [23]. Each stage processes five input bits and a carry-in signal to generate Sum, Carry, and Carry-out outputs. This architecture reduces partial product rows, minimizes propagation delay and hardware complexity, and improves the performance of high-speed arithmetic units such as multipliers and accumulators [24]. Its modular design also supports scalable implementation in advanced digital and ternary logic systems.

Figure 7RTL design using 5:2 compressors with ternary logic gates [25].

The high-performance multiplier architecture utilizes a 5:2 compressor tree to efficiently reduce partial products generated by the Partial Product Generator. Multiple levels of 5:2 compressors combine input bits and carry signals, reducing partial product rows and minimizing propagation delay. The final reduced sum and carry outputs are processed by the Final Adder to produce the multiplication result. This compressor-based architecture improves speed, reduces hardware complexity, and enhances power efficiency, making it suitable for VLSI arithmetic and high-speed digital signal processing applications.

3. Results

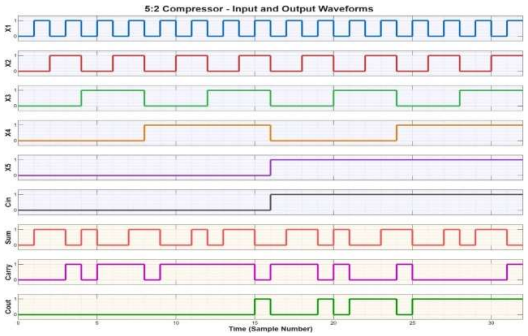


Figure8 Input and output waveforms of 5:2 compressors.

The waveform analysis of the 5:2 compressor verifies its correct operation using inputs X1–X5 and Cin. Different input combinations are applied to test the design, and the outputs Sum, Carry, and Cout represent proper arithmetic processing. The Sum output reflects bitwise addition, while Carry and Cout indicate intermediate and final carry propagation. The waveform results show accurate and smooth transitions, confirming that the compressor correctly reduces five input bits into fewer outputs. The simulation proves the reliability and correctness of the design for high-speed arithmetic applications such as partial product reduction and multiplier implementation. MATLAB modelling validates the functional performance before hardware realization.

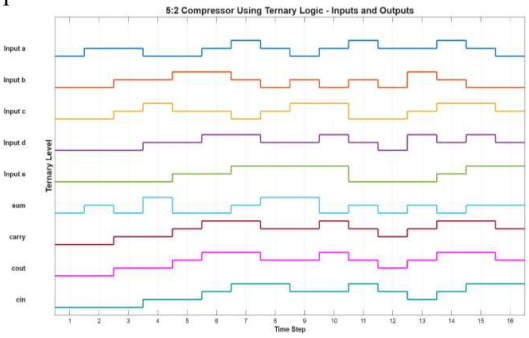


Figure 9 Input and output waveforms of ternary 5:2 compressor.

The waveform analysis of the ternary 5:2 compressor demonstrates its operation using five ternary inputs (a, b, c, d, and e) and a carry input (Cin) with three logic levels: 0, 1, and 2. The compressor generates three outputs, namely Sum, Carry, and Carry-out (Cout), which represent the ternary arithmetic operation and carry propagation. The simulation results confirm that the compressor efficiently combines five ternary inputs into reduced output signals with accurate state transitions. Compared with binary compressors, the ternary design provides a more compact representation and can reduce circuit complexity. MATLAB-based waveform analysis is performed over multiple time steps to validate the functionality, reliability, and suitability of the ternary 5:2 compressor for high-speed ternary arithmetic and multiplier applications.

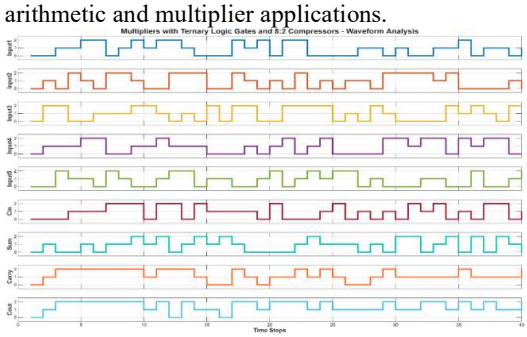


Figure 10 Multiplier simulation using 5:2 compressors and ternary logic.

The waveform analysis of the ternary multiplier based on a 5:2 compressor confirms its correct operation using five ternary inputs and a carry input (Cin). The inputs operate in three logic levels (0, 1, 2), enabling efficient multi-valued arithmetic processing. The outputs Sum, Carry, and Cout represent proper compression of partial products and carry propagation during multiplication. The simulation results show accurate transitions and stable output behaviour under different input conditions, verifying correct arithmetic functionality. The use of ternary logic with a 5:2 compressor reduces carry propagation delay and improves overall computational efficiency. MATLAB-based waveform analysis validates the design in terms of speed, reliability, and low-power performance. The results confirm that the proposed ternary multiplier is suitable for high-performance VLSI arithmetic circuits and advanced digital systems, where efficient multiplication and reduced complexity are essential.

Table 4 Waveform data of 5:2 compressors using ternary logic.

Time	I1	I2	I3	I4	I5	Cin	Sum	Carry	Cout	Decimal	Input_Sum
1	0	0	0	0	0	0	0	0	0	0	0
2	0	1	2	1	0	0	1	1	1	13	4
3	1	0	2	1	2	0	0	2	2	24	6
4	1	2	0	1	1	1	0	2	2	24	6
5	2	1	0	2	1	1	1	2	2	25	7
6	2	0	1	2	0	1	0	2	2	24	6
7	0	2	1	0	2	2	1	2	2	25	7
8	1	2	1	0	1	2	1	2	2	25	7
9	2	1	2	1	0	2	2	2	2	26	8
10	1	0	2	1	0	0	1	1	1	13	4
11	2	0	1	2	1	2	2	2	2	26	8
12	2	2	0	1	2	2	0	2	0	6	9
13	1	2	1	1	2	0	1	2	2	25	7
14	0	2	0	1	0	2	2	1	1	14	5
15	0	0	2	0	1	1	1	0	1	10	4
16	0	0	0	0	1	1	2	0	0	2	2
17	2	2	2	0	0	1	1	2	2	25	7
18	1	1	2	1	0	1	0	1	2	21	6
19	2	0	0	0	1	0	0	0	1	9	3
20	0	1	0	1	2	2	0	1	2	21	6

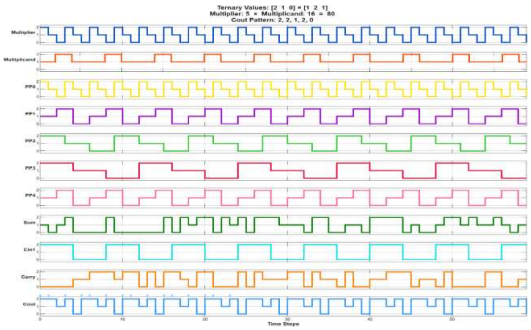


Figure 11 Simulation of input and output multiplicand with ternary logic based 5:2 compressors.

The timing waveform of the ternary multiplier shows partial product generation, summation, and carry propagation over time. The multiplier and multiplicand operate using ternary logic levels (0, 1, 2), providing higher information density than binary systems. Based on ternary inputs, partial products (PP_0–PP_4) are generated and combined to form the Sum output according to their weights.

Carry-in (Cin), Carry, and Carry-out (Cout) signals manage intermediate and final carry propagation during multiplication. The waveform confirms

```
Input A (ternary LSB->MSB): [2 1 0]
Input B (ternary LSB->MSB): [1 2 1]
Output Product (ternary LSB->MSB): [2 2 1 2 0]
Decimal equivalent: 71
```

correct synchronization, accurate partial product accumulation, and proper carry handling throughout the operation. Simulation results validate the correct functionality of the ternary multiplier architecture and demonstrate its efficiency for high-speed multi-valued arithmetic applications.

Figure 12MATLAB result of ternary multiplication.

The ternary multiplication process is performed from the least significant bit (LSB) to the most significant bit (MSB) using the given input operands. The obtained output demonstrates the approximate behaviour of the ternary multiplier, where minor deviations from the exact result occur due to approximations in partial product generation and carry handling. These approximations reduce circuit complexity, delay, and power consumption, making the design suitable for low-power and error-tolerant arithmetic applications.

4. Discussion

Compared to conventional binary multipliers, the construction of a multiplier that combines ternary logic with 5:2 compressors hasshown notable gains in efficiency and performance. Because ternary logic has fewer transistors and simpler connections, it uses less power and has a smaller silicon footprint.According to simulation results, the suggested architecture reduces propagation delays while enabling scalability for bigger bit-width multipliers. High-performance digital designs help to accelerate the accumulation and simplify the processing stages using the 5:2 compressors in the partial product reduction phase. The design and implementation of the multiplier using the 5:2 compressors and ternary approach have improved the performance and efficiency of the design much beyond the binary multipliers.

5. Conclusion

This notion and realization of a multiplier by using 5:2 compressors with ternary logic perform much more effectively and efficiently compared with binary multipliers. The power consumption and size of ternary logic are smaller because it uses fewer transistors and simpler interconnects than binary logic. The simulation results indicate that the proposed architecture can reduce propagation delay and improve scaling for multipliers with a wider bit-width. 5:2 compressors are used in high-performance digital systems to carry out the partial product reduction stage, enabling quicker accumulation with fewer stages.

6. Acknowledgements

The authors would like to thank the Department of Electronics and Communication Engineering at the Madanapalle Institute of Technology and Science for their invaluable support and facilities provided during this research. The authors also express their sincere gratitude to G. Reddy Hemantha for his guidance and valuable suggestions throughout the study. In addition, the authors acknowledge the support of colleagues and fellow researchers during the simulation and verification phases of this work.

7. References

1. Yoon J, Baek S, Kim S, Kang S. Optimizing ternary multiplier design with fast ternary adder. IEEE Transactions on Circuits and Systems II: Express Briefs. 2022 Sep 28;70(2):766-70.
2. Zahoor F, Jaber RA, Isyaku UB, Sharma T, Bashir F, Abbas H, Alzahrani AS, Gupta S, Hanif M. Design implementations of ternary logic systems: A critical review. Results in engineering. 2024 Sep 1;23:102761.
3. Zhao G, Zeng Z, Wang X, Qoutb AG, Coquet P, Friedman EG, Tay BK, Huang M. Efficient ternary logic circuits optimized by ternary arithmetic algorithms. IEEE Transactions on Emerging Topics in Computing. 2023 Oct 19;12(3):826-39.
4. Vijeyakumar KN, Sumathy V, Devi MG, Tamilselvan S, Nair RR. Design of hardware efficient high speed multiplier using modified ternary logic. Procedia Engineering. 2012 Jan 1;38:2186-95.
5. Mohammadi Mohaghegh S, Sabbaghi-Nadooshan R, Mohammadi M. Design of a

ternary QCA multiplier and multiplexer: a model-based approach. Analog Integrated Circuits and Signal Processing. 2019 Oct;101(1):23-9.
6. Srinivasu B, Sridharan K. A synthesis methodology for ternary logic circuits in emerging device technologies. IEEE Transactions on Circuits and Systems I: Regular Papers. 2017 Apr 12;64(8):2146-59.
7. Kim S, Lee SY, Park S, Kim KR, Kang S. A logic synthesis methodology for low-power ternary logic circuits. IEEE Transactions on Circuits and Systems I: Regular Papers. 2020 May 7;67(9):3138-51.
8. Wen W, Zhao G, Hu W, Li Z, Wang X, Friedman EG, Tay BK, Ke S, Huang M. High Efficiency Multiply-Accumulator Using Ternary Logic and Ternary Approximate Algorithm. IEEE Transactions on Circuits and Systems I: Regular Papers. 2024 Nov 18.
9. Soliman N, Fouda ME, Alhurbi AG, Said LA, Madian AH, Radwan AG. Ternary functions design using memristive threshold logic. Ieee Access. 2019 Apr 16;7:48371-81.
10. Kim S, Kang Y, Baek S, Choi Y, Kang S. Low-power ternary multiplication using approximate computing. IEEE Transactions on Circuits and Systems II: Express Briefs. 2021 Mar 26;68(8):2947-51.
11. Unutulmaz A, Ünsalan C. Implementation and applications of a ternary threshold logic gate. Circuits, Systems, and Signal Processing. 2024 Feb;43(2):1192-207.
12. Rohani Z, Emrani Zarandi AA. Simulation for a low-energy ternary multiplier cell based on Graphene nanoribbon field-effect transistor. International Journal of Nano Dimension. 2024 Jan 1;15(1):49-62.
13. Gao S, Al-Khalili D, Chabini N. Implementation of large size multipliers using ternary adders and higher order compressors. In2009 International Conference on Microelectronics-ICM 2009 Dec 19 (pp. 118-121). IEEE.
14. Kumm M, Hardieck M, Willkomm J, Zipf P, Meyer-Baese U. Multiple constant multiplication with ternary adders. In2013 23rd International Conference on Field programmable Logic and Applications 2013 Sep 2 (pp. 1-8). IEEE.
15. Xu K, Geng S, Yang H, Cui A. Design and Analysis of Majority Logic-Based Approximate 5: 2 Compressors and Multipliers. In2024 5th International Conference on Computer Engineering and Application (ICCEA) 2024 Apr 12 (pp. 455-459). IEEE.
16. Abbasian E, Sofimowloodi S. A high-performance and energy-efficient ternary multiplier using CNTFETs. Arabian Journal for Science and Engineering. 2023 Nov;48(11):14365-79.
17. Panahi MM, Hashemipour O, Navi K. A novel design of a multiplier using reversible ternary gates. IETE Journal of Research. 2021 Nov 2;67(6):744-53.
18. Naga Jyothi G, Debanjan K, Anusha G. ASIC implementation of fixed-point iterative, parallel, and pipeline CORDIC algorithm. InSoft Computing for Problem Solving: SocProS 2018, Volume 1 2019 Nov 28 (pp. 341-351). Singapore: Springer Singapore.
19. R. A. Jaber, A. El-Maleh, H. H. Saleh, and M. A. N. Al-Qutayri, “Ternary full adder designs employing unary operators and ternary multiplexers,” Micromachines, vol. 14, no. 5, p. 1064, 2023.
20. Macsorley OL. High-speed arithmetic in binary computers. Proceedings of the IRE. 2007 Jan 22;49(1):67-91.
21. Lakshmi V, Reuben J, Pudi V. A novel in-memory wallace tree multiplier architecture using majority logic. IEEE Transactions on Circuits and Systems I: Regular Papers. 2021 Nov 30;69(3):1148-58.
22. Labrado C, Thapliyal H, Lombardi F. Design of majority logic based approximate arithmetic circuits. In2017 IEEE international symposium on circuits and systems (ISCAS) 2017 May 28 (pp. 1-4). IEEE.
23. Shafiabadi MA, Sharifi F. Approximate 5-2 Compressor Cell Using Spin-Based Majority Gates. InSpin 2020 Jun 25 (Vol. 10, No. 02, p. 2040004). World Scientific Publishing Company.
24. Liu W, Zhang T, McLarnon E, O’Neill M, Montuschi P, Lombardi F. Design and analysis of majority logic-based approximate adders and multipliers. IEEE Transactions on Emerging Topics in Computing. 2019 Jul 17;9(3):1609-24.
25. Balobas D, Konofaos N. Low-power high-performance CMOS 5-2 compressor with 58 transistors. Electronics Letters. 2018 Mar;54(5):278-80.