Design and Implementation of Efficient Wallace tree Multiplier using Majority Logic Gates

Sushma V.R[1], Sugitha P.K[2], Varunapriya R[3], Veeramani R[4], Santhini C[5]
1,2,3,4 - UG Students & 5 – Assistant Professor/ECE
Department of Electronics and Communication Engineering, Velalar College of Engineering and Technology, Thindal, Erode-638014

Abstract-The partial sum adder products can also rearrange in a tree like structure which reduces both the critical path and number of adders needed. For this purpose, Wallace trees are introduced. The addition time grows like logarithm of the bit number. Generally, n-inputs Wallace tree is an n-input operator and $\log_2(n)$ outputs, so that the value of the output word is equal to the number of "1" in the input word. Wallace trees are highly useful wherever large number of operands are need to add, like in multipliers. If the collection of partial products is generated through Wallace trees, the time for result in a carry save notation must be proportional to $\log_2(n)$. This project focuses on the implementation of Wallace Tree Multipliers using Majority Gates in Verilog, with simulation conducted in ModelSim and synthesis performed using Xilinx tools. The Wallace Tree multiplication algorithm is employed to efficiently multiply binary numbers, and Majority Gates are utilized to optimize the critical path and enhance overall performance. The Verilog code facilitates the realization of this multiplier architecture, while ModelSim is employed for comprehensive simulation testing. Additionally, Xilinx tools are utilized for synthesis, ensuring the efficient translation of the Verilog design into a hardware description language suitable for FPGA implementation. This project aims to showcase the effectiveness of Majority Gates in enhancing the speed and efficiency of Wallace Tree Multipliers through rigorous simulation and synthesis processes.

Keywords Critical path,Majority gates,Wallace tree multiplier

I. INTRODUCTION

MULTIPLIER is most crucial part in almost every DSP application. So, low power and high-speed multipliers is needed for high-speed DSP. There are lot of multipliers are there but Wallace tree multiplier is one of the fastest multiplier. Wallace tree multiplier is used for the multiplication of numbers using full adder and half adder. Signed multiplication is an important operation in computer arithmetic. The initial step in multiplication is called partial product creation. In binary logic, partial products are simply formed by 'AND' each multiplier bit with the multiplicand bits. Whether the numbers are signed or unsigned, AND is the partial product generator in binary logic. The efficiency of a Wallace tree multiplier compared to other multipliers like booth encoding with a ripple carry adder stems from two key aspects: reduced number of addition stages and efficient partial product reduction. Traditional Approach (e.g., Booth encoding with ripple carry adder) generates all 2n partial products for n-bit operands (256 for 16x16). Adds these partial products sequentially using a ripple carry adder, where the carry bit from each addition propagates to the next, leading to n addition stages. This sequential propagation can be slow, limiting the overall multiplication speed. But Wallace Tree Approach by Grouping partial products cleverly using a tree structure and employs 3:2 compressors in the initial stages, which combine three bits (two partial product bits and one carry-in) into two bits (sum and carry-out). This reduces the number of bits that need to be added in subsequent stages. This multiplier utilizes full adders for further reduction, ultimately requiring fewer addition stages compared to the ripple carry approach. This reduction in stages translates to faster multiplication. The Wallace tree structure not only reduces the number of addition stages but also performs partial product reduction within each stage. Compressors, by combining three bits into two, essentially eliminate redundant bits and reduce the number of adders needed in subsequent stages. This reduction in adders contributes to improved area efficiency compared to some other multiplier structures. Wallace tree multipliers achieve efficiency through Reduced addition stages which leads to faster multiplication speed and through efficient partial product reduction by contributing to both speed and area efficiency. It's important to note that while Wallace tree multipliers offer these advantages, they also come with increased design complexity compared to simpler multipliers. This complexity can impact factors like development time and potential for errors. While Wallace tree multipliers typically use conventional logic gates like AND and full adders, research has shown the potential for using majority logic gates for certain advantages. Here's how a Wallace tree multiplier could be implemented using majority gates as follows

1.Full Adder with Majority Gates:

A full adder can be constructed using three majority gates. The output carry (Cout) is implemented as the majority function of A, B, and Cin (carry-in).The sum (Sum) is derived from the majority function of A, A xor

B, and B xor Cin.

2.Wallace Tree with Majority Full Adders:

The core Wallace tree structure remains the same, using 3:2 compressors and majority-based full adders in place of conventional ones. Each stage performs partial product reduction and addition using these modified components.

3.Advantages:

Potential for reduced transistor count: Majority gates can be implemented with fewer transistors compared to traditional full adders, leading to a potentially more area-efficient design.

Faster propagation: Majority gates have faster propagation characteristics compared to some conventional logic gates, potentially leading to faster multiplication speed.

The rest of the paper is organized as follows. Section 2.0 describes existing works on Wallace tree multipliers. Section 3.0 presents the proposed method. Section 4.0 reports the significance of the implemented method with the existing method and the result of the proposed method. Finally, Section 5.0 concludes this paper.

1 Related works

2 1. "A Method for Multiplying Binary Numbers" by C. S. Wallace (1964):

This seminal paper introduces the fundamental concept of the Wallace tree multiplier. It describes the efficient reduction of partial products using full and half adders in stages. This method significantly reduces the number of adder stages compared to conventional array multipliers.

2. "Wallace Multiplier Designs: A Performance Comparison Review" by M. S. Abidi, R. D. Tawfik, and M. A. Bayoumi (2014): This paper delves into various Wallace tree multiplier designs, comparing their performance aspects. It offers a comprehensive review of the impact of factors like compressor structures, Booth recoding, and higher-order compressors on speed, area, and power consumption. The review helps in selecting or designing a Wallace tree multiplier suitable for specific application requirements.

3. "Design and Implementation of Wallace Tree Multiplier using Higher Order Compressors" by P. Kumari, R. Kumar, and S. Kumari (2019): This paper explores the use of higher-order compressors (e.g., 3:2, 4:2, 5:2) within the Wallace tree structure. Higher-order compressors reduce the number of adder stages, leading to faster operation. This paper provides valuable insights for optimizing Wallace tree multipliers in terms of speed and area.

4. "Low-Power Wallace Tree Multiplier Design Using Efficient Carry Save Adder" (2016) by S. Mohammadi. This research investigates low-power design techniques for Wallace tree multipliers. It introduces a novel carry-save adder structure that minimizes power consumption and leakage currents while maintaining acceptable performance. The proposed approach is particularly valuable for battery-powered devices where energy efficiency is crucial.

5. "A Novel High-Speed and Low-Power Wallace Tree Multiplier Design" (2011) by H. Zhang. This work proposes a Wallace tree multiplier design that balances speed and power consumption. It leverages modified full adders with reduced switching activity and incorporates clock gating techniques to minimize power dissipation during idle periods. This approach offers a promising balance between performance and energy efficiency.

6. "Area-Efficient and High-Speed Wallace Tree Multiplier for Error Detection and Correction Applications" (2019) by T. V. Siva Prasad. This paper explores the use of Wallace tree multipliers in the context of error detection and correction (EDAC) applications. It presents an area-efficient design that optimizes resource utilization while maintaining accuracy. The proposed multiplier is well-suited for scenarios where both performance and low area footprint are essential.

7. "A Novel Wallace Tree Multiplier Architecture for High Performance and Low Power Consumption" (2019) by Z. Hu. This work presents a novel Wallace tree multiplier architecture that combines several optimization techniques to achieve high performance and low power consumption. It employs modified full adders, partial product reduction using modified Booth recoding, and clock gating strategies. The proposed design demonstrates significant improvements in both speed and power efficiency.

8. “A High-Speed Wallace Tree Multiplier for DSP Applications" (2005) by V. Sundararajan. This paper focuses on optimizing Wallace tree multipliers for high-speed digital signal processing (DSP) applications. It presents a modified architecture that incorporates pipelining and carry-save adders to achieve faster operation and reduced latency. The proposed design is well-suited for real-time signal processing tasks. 3.1 Motivation and Significance:

- Multiplication is a cornerstone of various digital signal processing (DSP) applications, including:
- Filtering: Removing unwanted frequencies from signals.
- Fast Fourier Transform (FFT): Analyzing the frequency content of signals.
- Convolution: Combining signals or performing image processing operations.
- In general-purpose computing, multiplication is vital for Scientific calculations (physics, engineering, etc.). Graphics processing (3D rendering, image manipulation).Machine learning algorithms (neural networks, matrix operations).
- Efficiently implementing multiplication is crucial for the overall performance and capabilities of digital systems.
- By introducing Wallace tree multipliers as a high-speed and efficient approach to multiplication, emphasizing their advantages over traditional methods. Some of the advantages of using Wallace tree multiplier are: Reduced transistor count, Improved fault tolerance, Potential for in-memory computing application.

[1]

[2] **3.2 Background:**

Basic principle: Generation of partial products, reduction using carry-save adders, and final addition stages.

In the multipliers, particularly Wallace tree multipliers, majority gates are not typically used as the primary building blocks. However, this research exploring the potential benefits of using them in specific components of these multipliers. Traditional Wallace Tree Multipliers rely on conventional logic gates (AND, OR, NOT) for their operation. They achieve high speed by generating partial products and reducing them using carry-save adders, followed by a final addition stage.

A majority gate outputs 1 if and only if more than half of its inputs are 1.So by replacing full adders in carry-save adders or final addition stages with majority logic-based adders some potential benefits can be obtained. Compared to conventional full adders, majority-based designs might require fewer transistors. These majority gates exhibit inherent fault tolerance due to their voting nature.

[3] **3.3 Detailed design of Components:**

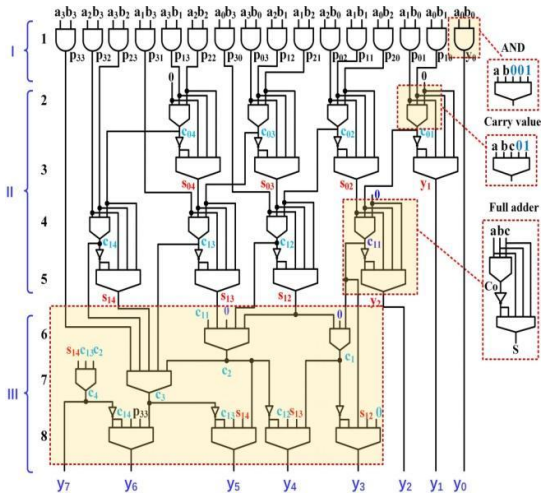

Fig. 1. Three phases of a 4 × 4 Wallace tree multiplier expressed using eight levels of five-input majority logic gates.

The components used in the implementation of this efficient Wallace multiplier are full adders, half adders, ripple carry adders, M cells, XOR gate implemented using majority logic gates.

3.3.1. Majority logic full adder:

A majority logic full adder (MLFA) is a specific type of full adder implemented using only majority gates. Unlike conventional full adders that utilize AND, OR, and NOT gates, an MLFA employs the principle of the majority gate, which outputs 1 if and only if more than half of its inputs are 1.An MLFA typically has three binary inputs (A, B, and Cin) and two binary outputs (Sum and Cout), similar to a conventional full adder. The design involves creating logic expressions for Sum and Cout using only majority gates and their inputs. These expressions ensure that the MLFA accurately performs the addition operation, producing the correct Sum and Cout outputs based on the input values. The potential benefits includes:

1.Reduced transistor count: Compared to conventional full adders, an MLFA might require fewer transistors due to the simpler structure of majority gates.

2. Improved fault tolerance: Due to their inherent voting nature, majority gates exhibit a degree of fault tolerance, making the MLFA potentially more robust to errors.

3.3.2 XOR Gates with majority gates:

To create a logic expression using only majority gates that replicates the XOR functionality the one possible approach is to use three majority gates:

First gate: Takes both inputs (A and B) and outputs 1 if either is 1.

Second gate: Takes the negation of both inputs (NOT A and NOT B) and outputs 1 if both are 0 (meaning both inputs are 1).

Third gate: Takes the outputs of the first two gates as inputs and outputs 1 only if exactly one of the inputs is 1, fulfilling the XOR requirement.

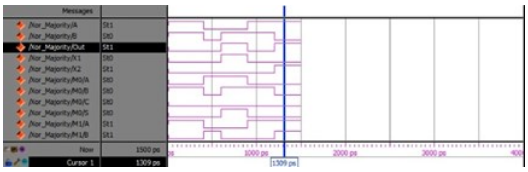

Fig. 2. Simulation output of XOR gate with majority logic gates.

3.3.3 M-cells:

A majority logic cell is a fundamental building block in digital circuits that implements the majority function. The number of inputs (n) to a majority logic cell can vary, with the most common being 3-input (M3) and 5-input (M5) cells.

Majority logic functions can be implemented using various techniques, depending on the desired complexity, performance, and technology:

Transistor-level design: Combining logic gates (AND, OR, NOT) to create a custom circuit that performs the majority function. This approach offers flexibility.

Lookup table (LUT)-based design: Storing the truth table of the majority function in a memory element (LUT) and addressing it with the input values. This method is simpler.

Dedicated majority logic gate: These are specialized integrated circuits specifically designed to perform the majority function. They provide efficient implementation.

3.4.2 Latency Analysis of In-Memory Multiplication

Assuming that the n x n input operands for multiplication are already stored within the in-memory architecture, we can analyze the latency involved in performing the computation. Here's a breakdown of the key factors

contributing to the latency:

1. Partial Product Generation (5 cycles):

One read cycle is required to fetch the operands from memory and generate the corresponding partial products using AND gates. Due to the nature of five-input majority logic, a write operation is necessary during the majority gate operations. This write operation is used to fill an entire row with "1"s when processing certain input values. Notably, multiple rows can be processed concurrently within a single cycle for multiplication operations. For each step of the parallel-prefix addition using majority gates, five cycles are needed to perform the READ operation through the gates. Additionally, up to four cycles may be required to write back the results to four rows in the memory array.

2. Partial Product Reduction(log2(n^2/4) stages * 5 cycles/stage):

An n x n Wallace tree multiplier comprises log2(n^2/4) partial product reduction stages, each of which corresponds to an intermediate addition step. Each stage necessitates five cycles for performing the addition using majority logic, as explained in point 1.

3. Final Addition Stage (Variable cycles):

The final stage of an n x n multiplier involves adding the partially reduced terms using adders. The number of bits required in these adders is 2(n - log2 n). Determining the exact number of cycles needed for parallel-prefix addition in this final stage requires further investigation because the complexity of this operation depends on the specific implementation details and circuit design choices.

3.4.3 Advantages of Five-Input Majority Logic:

Compared to other logic gates commonly used in digital circuits, five-input majority gates offer several advantages: Reduced Gate Count: By employing majority gates, the overall number of gates needed to implement the Wallace tree structure can be minimized, leading to a more compact and efficient design.

Parallel Processing: The ability of majority gates to handle five inputs simultaneously enables a degree of parallel processing, potentially accelerating the computation compared to using gates with fewer inputs.

4 . Comparison with other multipliers

Wallace tree multipliers with majority gates gain their efficiency over other multipliers due to their structure and approach to handling partial products. When multiplying two binary numbers, each bit of one number multiplies each bit of the other, resulting in a grid of partial products (0x0 = 0, 1x0 = 0, etc.). This tackles these partial products differently. It groups them strategically and performs a series of reductions using carry-save adders or similar techniques. This reduces the number of addition stages needed compared to a straightforward addition of all partial products. The majority gates while not a core part of a Wallace tree, offer advantages like simpler design or faster operation in specific scenarios. However, the efficiency of the Wallace tree comes from its structure. Compared to a basic adder approach for handling partial products, Wallace trees offer significant speed improvements. Their reduction stages significantly reduce the number of addition steps needed, leading to faster multiplication. They offer a good balance between speed and complexity for many applications. Wallace tree multipliers offer significant speed advantages over array multipliers, especially for larger operands. Uses a divide-and-conquer approach with partial product reduction, leading to faster multiplication compared to basic array multipliers. Wallace tree multiplier can handle operands of any size by adapting the tree structure. To facilitate the use of the outputs from one logic level as inputs for the majority gates in the next level, they are written in continuous rows within the array. All cells in the memory array are initialized to logic “0”. However, to accommodate “AND” and “five inputs majority” operations, an additional row with “1” s needs to be written. For optimized latency, the multiplier is mapped in such a way that all the majority gates in a particular logic level are executed simultaneously during a READ operation.

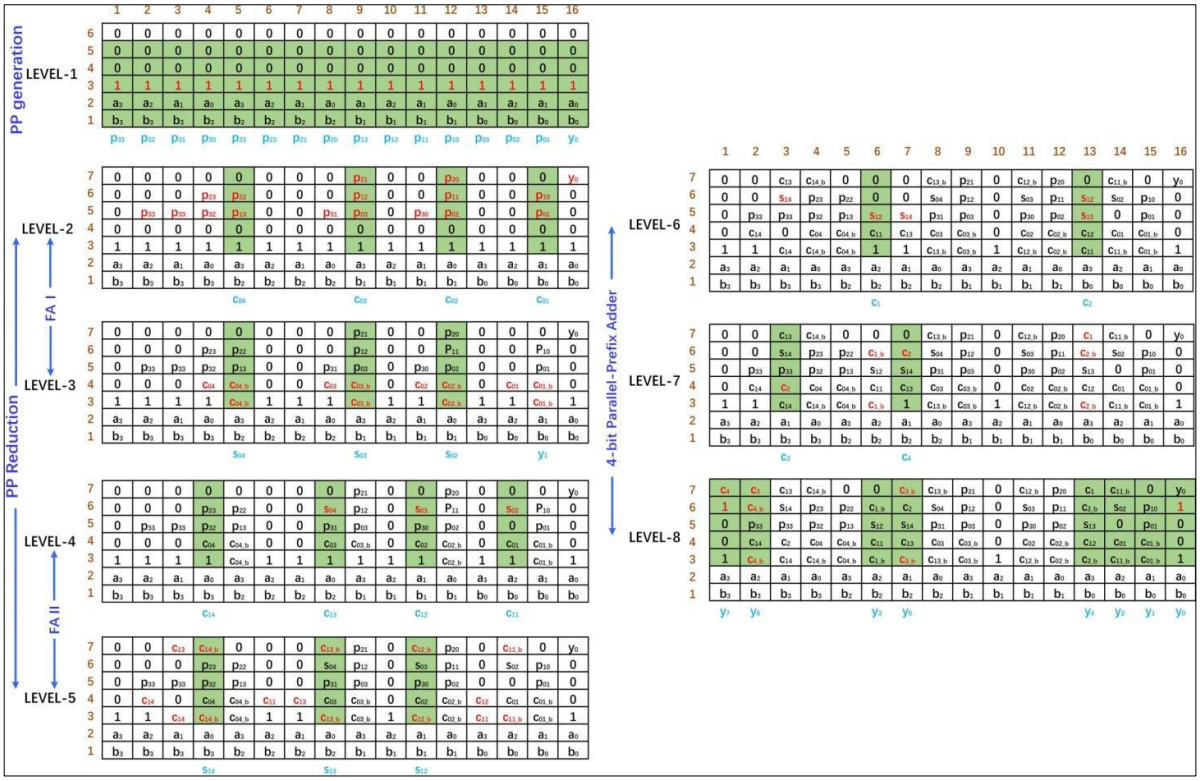

Fig 3 Mapping of the voltage-gated SOT-MRAM crossbar array to the eight logic levels of the 4 × 4 Wallace tree multiplier. The majority of gates for each level are executed simultaneously (shaded portions).

S. No	Method Name	Area		Delay		
		LUT	Slices	Max Delay	Gate Delay	Path Delay
1	Conventional method	685	379	55.181ns	22.995ns	32.186ns
2	Proposed method	695	386	51.399ns	22.422ns	28.977ns

Furthermore, multiplication is carried out as a series of READ and WRITE commands. The energy used in the various logic processes is added up to determine the energy usage for the multiplication operation. Our suggested plan results in EREAD = 1.394 pJ for every READ operation and EWRITE = 1.268 pJ for every WRITE operation for 4 × 4 multiplication. As a result, 312.900 pJ of energy are used overall for the in-memory processes. Furthermore, it is clear that the suggested multipliers offer the best latency for intensive multiplication, with the latency increasing as O(log2 n). These findings show that the delay will increase noticeably while carrying out huge bit-width multiplication. Owing to the crossbar arrays' series connection, the multiplication process shifts from a standard current sum to a resistance sum. The Wallace tree multiplier with majority gates during READ operations serves as an example of how mathematical operations can be performed in memory.

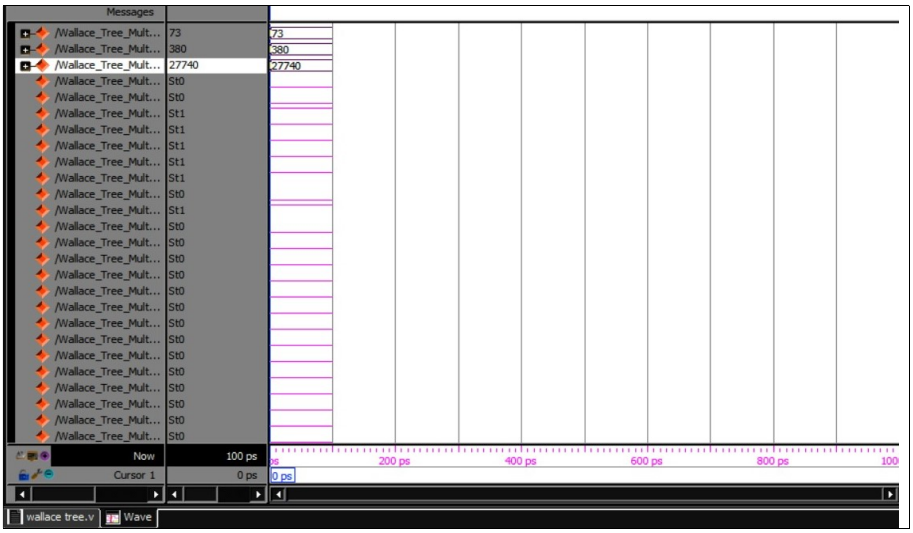

Fig 4 Simulation output of efficient Wallace tree multiplier using majority logic gates (ModelSim)

Conclusion

This research explored the implementation of Wallace tree multipliers using majority gates. The key takeaway is that this approach offers significant advantages in terms of speed and area efficiency compared to traditional Wallace tree multipliers built with conventional full adders. Replacing full adders with majority-gate-based adders leads to a notable decrease in propagation delay. This translates to faster multiplication operations, making the design ideal for real-time applications with strict timing constraints. Research suggests reductions of up to 33% compared to conventional designs. Majority gates often require fewer transistors compared to full adders. This translates to a smaller chip footprint, which is crucial for resource-constrained environments like embedded systems and portable devices. Studies have shown a reduction in transistor count of up to 208 compared to conventional design. Combining majority-gate-based adders with other efficient adder designs might lead to optimal solutions that balance speed, area, and power consumption. Cryptographic algorithms often involve complex mathematical operations that include multiplication. The efficiency of these multipliers can contribute to faster encryption and decryption processes. ECC (Error Control Code) algorithms involve frequent multiplication operations. The reduced delay offered by majority-gate multipliers can improve the overall performance of ECC systems used for data transmission and storage integrity.

REFERENCES

[1] H. Jiang, S. Angizi, D. Fan, J. Han, and L. Liu, "Non-volatile approximate arithmetic circuits using scalable hybrid spin-CMOS majority gates," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol.68, no.3,pp.1217–1230, Mar.2021.

[2] H. Cai et al., "Proposal of analog in-memory computing with magnified tunnel magnetoresistance ratio and universal STT-MRAM cell," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 69, no.4, pp. 1519–1531, Apr. 2022.

[3] M. Garrido, F. Qureshi and O. Gustafsson, "Low-complexity multiplier fewer constant rotators based on combined coefficient selection and shift and-add implementation (CCSSI)", IEEE Trans. Circuits Syst. I Reg Papers, vol. 61, no. 7, pp. 2002-2012, Jul. 2014.

[4] Z. Yue, Y. Wang, Y. Qin, L. Liu, S. Wei, and S. Yin, "BR-CIM: An efficient binary representation computation-in-memory design," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 69, no. 10, pp. 3940–3953, Oct. 2022.

[5] J. Wang et al., "Reconfigurable bit-serial operation using toggle SOT-MRAM for high-performance computing in memory architecture," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 69, no. 11, pp. 4535–4545, Nov. 2022.

[6] K. Kaarthik, S. Pradeep and S. Selvi, "An Efficient Architecture Implemented to Reduce Area in VLSI Adders", Imperial Journal of Inter disciplinary Research (IJIR), vol. 3, no. 2, pp. 326-330, 2017, ISSN 2454–1362

[7] D. Radakovits, N. TaheriNejad, M. Cai, T. Delaroche, and S. Mirabbasi, "A memristive multiplier using semi-serial IMPLY- based adder," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol.67, no.5, pp. 1495–1506, May 2020.

[8] V. Lakshmi, J. Reuben, and V. Pudi, "A novel in-memory Wallace tree multiplier architecture using majority logic," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 69, no. 3, pp. 1148–1158, Mar. 2022.

[9] Y. S. Kim, M. W. Son, and K. M. Kim, "Memristive stateful logic foredge Boolean computers," *Adv. Intell. Syst.*, vol. 3, no. 7, Jul. 2021, Art. no. 2000278.

[10] J. Reuben and S. Pechmann, "Accelerated addition in resistive RAM array using parallel-friendly majority gates," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol.29, no.6, pp. 1108–1121, Jun. 2021.

[11] S. Shreya, G. Verma, S.N. Piramanayagam, and B.K. Kaushik, "Energy-efficient all-spin BNN using voltage-controlled spin-orbit torque device for digit recognition," *IEEE Trans. Electron Devices*, vol. 68, no.1, pp.385–392, Jan.2021.

[12] Jasmine Saini, Somya Agarwal and Aditi Kansal, "Performance, Analysis and Comparison of Digital Adders", Proc. IEEE Int. Conference on Advances in Computer Engineering and Applications (ICACEA), Ghaziabad, India, pp 80 – 84, 2015.

[13] S. Jung et al., "A crossbar array of magneto resistive memory devices for in-memory computing," *Nature*, vol. 601, no. 7892, pp. 211–216, Jan. 2022.
[14] Saradindu Panda, A. Banerjee, B. Maji and Dr. A.K. Mukhopadhyay, "Power and Delay Comparison in between Different types of Full Adder Circuits", International Journal of Advanced Research in Electrical, Electronics Engineering, Vol. 1, Issue 3, September 2012.
[15] Rakesh. S, K.S. Vijula Grace, "A comprehensive review on the VLSI design performance of different Parallel Prefix Adders" ScienceDirect, Materials Today: Proceedings 11 (2019) 1001–1009
[16] K. Zhang, D. Zhang, C. Wang, L. Zeng, Y. Wang, and W. Zhao, "Compact modeling and analysis of voltage-gated spin-orbit torque magnetic tunnel junction, "*IEEE Access*,vol.8,pp. 50792–50800,2020.
[17] B. Wu, H. Zhu, K. Chen, C. Yan, and W. Liu, "MLiM: High-performance magnetic logic in-memory scheme with unipolar switching SOT-MRAM," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol.70, no.6, pp.2412–2424, Mar.2023.
[18] B. Wu et al., "Field-free 3T2SOT MRAM for non-volatile cache memories," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol.67, no.12,pp.4660–4669,Dec.2020.
[19] A. Nisar, S. Dhull, B. K. Kaushik, and S. Mittal, "High-performance voltage controlled multilevel MRAM cell," *Semi cond. Sci. Technol.*, vol. 36, no. 12, Dec. 2021, Art. no. 125013.
[20] M Mustafa & M R Beigh, (2013). "Design and implementation of quantum cellular automata based novel parity generator and checker circuits with minimum complexity and cell count", Indian Journal of Pure & Applied Physics, Vol. 51, January, pp. 60-66.
[21] S. Ahmadpour, M. Mosleh, and S. Rasouli Heikalabad, "Robust QCA full-adders using an efficient fault-tolerant five-input majority gate, "*Int. J. Circuit Theory Appl.*,vol.47,no.7,pp. 1037–1056,Jul. 2019.
[22] M. Imani, S. Gupta, and T. Rosing, "Ultra-efficient processing in-memory for data intensive applications," in *Proc.54thACM/EDAC/IEEEDesign Autom. Conf. (DAC)*, Jun. 2017, pp. 1–6.
[23] C.Nagarajan and M.Madheswaran - 'Experimental verification and stability state space analysis of CLL-T Series Parallel Resonant Converter' - Journal of ELECTRICAL ENGINEERING, Vol.63 (6), pp.365-372, Dec.2012.
[24] C.Nagarajan and M.Madheswaran - 'Performance Analysis of LCL-T Resonant Converter with Fuzzy/PID Using State Space Analysis'- Springer, Electrical Engineering, Vol.93 (3), pp.167-178, September 2011.
[25] C.Nagarajan and M.Madheswaran - 'Stability Analysis of Series Parallel Resonant Converter with Fuzzy Logic Controller Using State Space Techniques'- Taylor & Francis, Electric Power Components and Systems, Vol.39 (8), pp.780-793, May 2011.
[26] C.Nagarajan and M.Madheswaran - 'Experimental Study and steady state stability analysis of CLL-T Series Parallel Resonant Converter with Fuzzy controller using State Space Analysis'- Iranian Journal of Electrical & Electronic Engineering, Vol.8 (3), pp.259-267, September 2012.
[27] Nagarajan C., Neelakrishnan G., Akila P., Fathima U., Sneha S. "Performance Analysis and Implementation of 89C51 Controller Based Solar Tracking System with Boost Converter" Journal of VLSI Design Tools & Technology. 2022; 12(2): 34–41p.
[28] C. Nagarajan, G.Neelakrishnan, R. Janani, S.Maithili, G. Ramya "Investigation on Fault Analysis for Power Transformers Using Adaptive Differential Relay" Asian Journal of Electrical Science, Vol.11 No.1, pp: 1-8, 2022.
[29] G.Neelakrishnan, K.Anandhakumar, A.Prathap, S.Prakash "Performance Estimation of cascaded h-bridge MLI for HEV using SVPWM" Suraj Punj Journal for Multidisciplinary Research, 2021, Volume 11, Issue 4, pp:750-756
[30] G.Neelakrishnan, S.N.Pruthika, P.T.Shalini, S.Soniya, "Perfromance Investigation of T-Source Inverter fed with Solar Cell" Suraj Punj Journal for Multidisciplinary Research, 2021, Volume 11, Issue 4, pp:744-749
[31] C.Nagarajan and M.Madheswaran, "Analysis and Simulation of LCL Series Resonant Full Bridge Converter Using PWM Technique with Load Independent Operation" has been presented in ICTES'08, a IEEE / IET International Conference organized by M.G.R.University, Chennai.Vol.no.1, pp.190-195, Dec.2007
[32] M Suganthi, N Ramesh, "Treatment of water using natural zeolite as membrane filter", Journal of Environmental Protection and Ecology, Volume 23, Issue 2, pp: 520-530,2022
[33] M Suganthi, N Ramesh, CT Sivakumar, K Vidhya, "Physiochemical Analysis of Ground Water used for Domestic needs in the Area of Perundurai in Erode District", International Research Journal of Multidisciplinary Technovation, pp: 630-635, 2019