DESIGN OF ENSEMBLE SYSTOLIC MULTIPLIER WITH FAULT TOLERANT SELF-CHECKING ADDERS

Martha Shravya [1], Dr.Seetharam Khetavath [2]

[1] PG Scholar, Department of Electronics and Communication Engineering, Chaitanya (Deemed to be University) Hanmakonda, Warangal, Telangana-506001
Email: shravya.martha66@gmail.com

[2] Professor & Head, Department of Electronics and Communication Engineering, Chaitanya (Deemed to be University) Hanmakonda, Warangal, Telangana-506001.
Email: seetharamkhetavath@gmail.com

ABSTRACT

Multipliers are basic building blocks in various integrated circuits like microprocessors, micro controllers, and ALUs. The existing multipliers suffer from high power consumption and inefficient use of hardware resources. They often rely on traditional adder structures that are not tailored for specific operations, leading to suboptimal performance. Additionally, their fixed architectures limit adaptability and scalability in different applications. So, the proposed approach offers enhanced computational efficiency and reduced power consumption compared to conventional multiplier designs. By integrating multiplexer-dependent adders into the self-checking adders based Systolic Array, the proposed method optimizes resource utilization and delivers improved performance for various arithmetic operations. This integration allows for dynamic selection of adder types based on the specific multiplication operation, significantly reducing power consumption and latency. By adapting the hardware resources to the computational needs, the method achieves higher efficiency and flexibility, making it suitable for a wide range of applications in digital signal processing and data processing systems

Keywords: Systolic Multiplier, Fault Tolerant, Self-Checking Adders,Razor FF

1. INTRODUCTION

On a VLSI device, the complexity of the signal processing systems that are implemented is increasing all the time because the scale of integration is growing all the time. Not only do these signal processing applications need a large calculation capability, but they also use a significant amount of power. In today's VLSI system design, performance and area are still the two most important design considerations; nevertheless, power consumption has emerged as an issue of paramount importance []. There are two primary drivers that contribute to the need for low-power VLSI systems. First, as the operating frequency and processing capacity per chip continue to steadily increase, big currents need to be provided, and the heat caused by high power consumption has to be removed using appropriate cooling methods. Both of these challenges must be met. Second, the amount of time that portable electronic gadgets may operate on a single charge is restricted. A design that consumes less power results in these portable gadgets having an operational life that is much longer.

Addition is a fundamental mathematical operation that, in most cases, has a significant influence on the overall performance of digital systems. Adders are the most common kind of

calculator used in electronic applications. Multipliers and digital signal processors (DSPs) utilize these components to carry out a variety of algorithms, such as FFT, FIR, and IIR, respectively. Adders are introduced into the discussion whenever the notion of multiplication is discussed. As is well knowledge, microprocessors are capable of carrying out millions of instructions in a single second. When it comes to the design of multipliers, the most crucial factor to take into account is the maximum possible processing speed. Miniaturization of the gadget should be prioritized, and power consumption should be kept to a minimum, so that it may be easily transported. Mobile phones, laptops, and other electronic devices demand a larger battery backup.

2. LITERATURE SURVEY

Sadhineni, Harika, S. Josaph, et.al [1] proposed this paper; we designed the CMOS based multiplier using LFSR for high-speed operation. Basically, the memory cells, multidimensional memory circuits and memory frameworks were referenced with various sub-chips. The main intent behind using a Linear Feedback Shift Register (LFSR) Controller was to provide full speed operations. The proposed architecture planned full-speed activity of some non-straight calculations that were winded up increasingly more significant in present day memory testing, conclusion. The combinational logic and scan logic provides a solution to the system in a specified manner. Compared to existed design, the proposed system gives effective results.

Srinivas, Chundi Sai, M. S. Manohar, et.al [2] discussed main objective of this research article was to designed high efficient accurate DL-PO logic multiplier for low power applications. Primarily, the execution of DL-PO Multiplier was based on VLSI chips, which were used as critical element. In the multiplier, product information was utilised to create the propagator and generator signals in the same way. While designing the dual partial product unit, optimized multiplier were used most widely. The DL-PO logic Multiplier uses adequate hardware implemented. The DL-PO logic depended on multiple selection logic modules (MSLM). Hence compared to state-of-the-art system, the DL-PO logic Multiplier systems gives effective results in terms of speed, area and delay.

Kim, Sunwoong, Cameron J. Norris, et.al [3] discussed IEEE 754 standard for floating-point (FP) arithmetic was widely used for real numbers. Recently, a variant called posit was proposed to improve the precision around 1 and −1. Since FP multiplication requires high computational complexity, various algorithmic approaches and hardware accelerator solutions were explored. In this context, this article proposes a novel area-efficient logarithmic multiplier architecture for different real number formats, which also provides a significant and useful accuracy/latency tradeoff at runtime. To reduce the logic area in field-programmable gate arrays (FPGAs), this article offers two innovations: applying logarithm to only a single operand and mitigating the accuracy drop caused by this modification with advanced error converging and operand selection schemes. Our multiplier design for single-precision FP (SPFP) numbers uses 58% fewer hardware resources than the iterative Mitchell's multiplier (IMM) design of Babić et al. extended for SPFP numbers. The error falls within 0.5% when the number of iterations reaches 5. In JPEG, our SPFP multiplier with four iterations produces nearly identical image quality results to the conventional exact multiplier. We further show how to merge two SPFP multipliers for double-precision FP (DPFP) multiplication. This DPFP multiplier design reduced the hardware resources of the IMM design extended for DPFP numbers by 60%. Finally, we demonstrate how our SPFP multiplier design can be slightly modified for 32-bit posit multiplication. It achieves a significantly higher accuracy by

increasing the number of iterations compared to state-of-the-art approximate posits multiplier designs.

Minaeifar, Atefeh, Ebrahim Abiri, et.al [4] observed multipliers were most frequently employed components in a system, responsible for performing computations, while significantly contributing to power consumption. In that article, a new architecture was presented by removing the least significant bits to implement three multipliers (Mul-1, Mul-2, and Mul-3) with the aim of reducing complexity and power consumption. Mul-1 demonstrated the highest accuracy along with low energy consumption compared to previous works, achieving a favourable trade-off between accuracy and energy consumption. All proposed designs and existing multipliers were simulated and compared in 7 nm FinFET technology using the Hspice tool. Additionally, the accuracy and quality of the proposed approximate multipliers were evaluated using MATLAB. The results indicated that Mul-1 and Mul-3 were highly efficient in image processing applications. According to the findings, Mul-1 surpassed its counterpart by 10%, 50%, and 50% in terms of PDP, NMED, and MRED, respectively. Furthermore, Mul-3 exhibited satisfactory MSSIM in DSP applications, outperforming its counterpart by 23% and 16% in PDP and MRED. Meanwhile, Mul-2 improved PDP by nearly 53% compared to Mul-1 and had the lowest power consumption.

Hui, Yajuan, Qingzhen Li, LeiminWang, et.al [5] proposed In-memory computing represents an efficient paradigm for high-performance computing using crossbar arrays of emerging non-volatile devices. While various techniques have been emerged to implement Boolean logic in memory, the latency of arithmetic circuits, particularly multipliers, significantly increases with bit-width. In this work, we introduce an in-memory Wallace tree multiplier based on majority gates within voltage-gated spin-orbit torque (SOT) magneto resistive random access memory (MRAM) crossbar arrays. By utilizing a resistance sum, the majority gate is implemented during READ operations in voltage-gated SOT-MRAM crossbar arrays, resulting in reduced read currents and improved energy efficiency. We employ a series of READ and WRITE operations to perform multiplier calculations, leveraged the fast READ and WRITE speeds of voltage-gated SOT-MRAM devices. Furthermore, the use of five-input majority gates simplifies multiplication by employed uniform logic gates and reducing logic depth, thereby lowering the operation's complexity and the total number of occupied cells. Our experimental results demonstrate that the proposed in-memory Wallace tree multipliers consume three times less energy for in-memory operations than previously reported 4 × 4 multipliers. Moreover, the proposed method reduces the delay overhead from O (n2) to O (log2n), where n represents the number of bi.

Parmar, Rushik, Khushil Yadav, GauraangiAnand,et.al [6] introduced myocardial infarction (MI) was a cardiac abnormality in which the coronary artery gets blocked, causing millions of fatalities every year. MI has a very high mortality and disability rate; therefore, with the detection of MI, it was also imperative to determine the location of the blockage to provide on–time treatment to avoid any fatality. In this paper, for the first time, a VLSI architecture was proposed that can determine the location of the infarction in real-time. The proposed architecture classifies the electrocardiogram (ECG) into twelve classes and achieves an average accuracy, sensitivity, and specificity of 99.90%, 99.49%, and 99.94%, respectively. Its area utilization is 1.69 mm2 at SCL 180 nm Bulk CMOS technology node, and the power consumption was 268.9 μW at 250 KHz. The low area and power requirements and real-time classification capability make the classifier suitable for wearable devices.

Beura, Srikant Kumar, Sudeshna Manjari Mahanta, et.al [7] proposed inexact computing, was a modern approach for the development of

low power and high-performance digital circuits, which involves approximation stages to get the utmost accurate result and are very much applicable in some error-tolerant applications like image processing. Such applications were limited to human eyesight ability, which given us the freedom to make the approximate output. In this manuscript, two inexact partial product generators, viz.Namely IPPG1, IPPG2 were proposed for the design of radix-4 based 8 × 8 Booth multipliers IRBM1.1, IRBM1.2, respectively. Moreover, an inexact 4:2 compressor was also proposed to add such partial product bits generated by IPPG1 and IPPG2, and they were nomenclature as IRBM2.1 and IRBM 2.2, respectively. IPPG1 and IPPG2 were designed by introducing errors in the Karnaugh's map to reduce the circuit complexity and errors, respectively. Similarly, errors were introduced in the truth table of the exact 4:2 compressor to design a low power, high speed inexact 4:2 compressor. Error metrics estimation of such proposed inexact Booth multipliers and the state-of-the-art designs are performed using MATLAB and circuit performance parameters like area, computational delay and power dissipation are extracted using gpdk45 nm technology. While comparing, IRBM1.2 offers minimum error metrics than all other reported design so far. Moreover, in the circuit level prospect, IRBM1.1 consumes minimum power and IRBM2.2 is the fastest among its reported counter-parts. IRBM2.2 offers minimum power-delay-area (PDA) product over all of its reported counter-parts. For the validation of the proposed designs, an image multiplication, edge-detection using Sobel operator and convolution neural network-based application was incorporated and quality assessment parameters are measured using MATLAB.

Vakili, Bahareh, Omid Akbari, et.al [8] introduced Approximate computing was one of the promising techniques in error-resilient applications to overcome high-density integration challenges, such as energy consumption and performance. Multipliers constitute a significant portion of computer arithmetic units, leading to considerable energy and time consumption. In this paper, low-power and compact approximate compressors were proposed for composing approximate Dadda multiplier structures, including compressors, half adders, and full adders, which utilize three-phase partial product compression: truncated, approximation, and exact columns. In approximate columns, approximate compressors were considered, derived from the truth table of the exact 4:2 compressor and simplified K-map entries based on the probability of each combination of inputs. An error-correcting module (ECM) was designed to distinguish specific cases and reduce the error metrics. All circuits were simulated using ModelSim and then synthesized using Design Compiler with the 15 nm FinFET technology. When compared to state-of-the-art works, our multipliers exhibited approximately 30%, 43%, and 10% reductions in power, area, and delay, respectively. To evaluate their functionality, we conducted image multiplication and implemented a simple multi-layer perception (MLP) neural network using the modified National Institute of Standards and Technology (MNIST) dataset in MATLAB with 0.998 mean structural similarity index metric (MSSIM), 51 dB peak-signal noise ratio (PSNR), and 95% classification accuracy.

Haq, Shams Ul, Erfan Abbasian, et.al[9] developed appeal of portable electronics, embedded systems, and other smart devices steadily growing over time. The multi-valued logic (MVL) was primarily introduced to handle the interconnect issues in binary logic. The use of ternary logic reduces the circuit complexity, power consumption in interconnects, and total chip area. The carbon annotate field-effect transistor (CNTFET) offers some significant features like low leakage power, equal hole and electron mobility, and modulation in threshold voltage by adjusting the diameter of the carbon annotate (CNT). In this paper, a clocked

dynamic and pass transistor logic-based ternary 1-trit multiplier circuit with only 28 transistors was proposed. The proposed logic avoided the conventional approach of using logic gates, encoder/decoder, and multiplexers, which reduces the total device count to implement the multiplier. The reduced transistor count offers a reduction in power consumption, delay, power-delay-product (PDP), and energy-delay-product (EDP) in comparison to other state-of-the-art designs. Stanford 32-nm CNTFET model file and The Synopsis HSPICE simulator were used to carry out the simulations at a supply of 0.9 V. The proposed circuit shows a 34.16 %, 66.44 %, and 82.80 % reduction in power, PDP, and EDP as compared to the lowest power, delay, and PDP of the designs under consideration. The performance of the multiplier was evaluated under different temperatures, supply voltages, loads, and oxide thickness.

Rohani, Zahra, et.al [10] illustrated the multiplier circuit was considered a significant component of larger circuits, such as the arithmetic and logic unit (ALU), and it was crucial to enhance its energy efficiency. This objective can be easily achieved by utilizing grapheme nanoribbon field-effect transistor (GNRFET) devices and adopting ternary logic. Ternary circuit designs demonstrated superior energy efficiency and occupied less space compared to binary ones. The adjustability of the threshold voltage (Vth) in GNRFET devices was directly influenced by the width of the grapheme nanoribbon (GNR). This offers significant advantages for ternary circuit designs. This paper presents a 24-transistor low-energy GNRFET-based single-trit ternary multiplier. Our proposed design incorporates an enhanced voltage division technique to achieve logic '1' while minimizing power consumption. The primary design approach employed in our design involves the utilization of unary operators and specialized transistor configurations to reduce the number of transistors and shorten the critical path. We used the Hewlett simulation program with integrated circuit emphasis (HSPICE) and GNRFET technology with a 32- nm channel length operating at 0.9 V and 300˚ K to evaluate the efficiency of our circuit. We then compared it with similar existing ternary multiplier circuits. The suggested circuit displays favourable delay and power consumption characteristics and ranked as the second most optimal design in terms of energy efficiency.

Kuo, Chao-Tsung, et.al [11] developed multi-modulus architecture based on the radix-8 Booth encoding of a modulo $(2n - 1)$ multiplier, a modulo $(2n)$ multiplier, and a modulo $(2n + 1)$ multiplier was proposed in this paper. It uses the original single circuit and shared many common circuit characteristics with a small extra circuit to carry out multi-modulus operations. Compared with a previous radix-4 study, the radix-8 architecture increases the modulation multiplication encoding selection from three codes to four codes. This reduces the use of partial products from $\lfloor n/2 \rfloor$ to $\lfloor n/3 \rfloor + 1$, but it increases the operation complexity for multiplication by three circuits. A hard multiple generator (HMG) is used to address this problem. Two judgment signals in the multi-modulus circuit can be used to perform three operations of the modulo $(2n - 1)$ multiplier, modulo $(2n)$ multiplier, and modulo $(2n + 1)$ multiplier at the same time. The weighted representation is used to reduce the number of partial products. Compared with previously reported methods in the literature, the proposed approach achieved better performance by being more area-efficient, being faster, consuming low power, and having a lower area-delay product (ADP) and power-delay product (PDP). With the multi-modulus HMG, the proposed modified architecture saved 34.48–55.23% of hardware area. Compared with previous studies on the multi-modulus multiplier, the proposed architecture can save 22.78–35.46%, 4.12–11.15%, 12.59–24.73%, 27.88–38.88%, and 20.49–27.85% of hardware area, delay time, dissipation power, ADP, and PDP, respectively. Xilinx field programmable gate array (FPGA)

Vivado 2019.2 tools and the Verilog hardware description language are used for synthesis and implementation. The Xilinx Artix-7 XC7A35T-CSG324-1 chipset was adopted to evaluate the performance.

Hao Li, et.al [12] proposed Modular multiplication plays a crucial role in modern cryptography. Montgomery modular multiplication (MMM), one of the most classic and practical modular multiplication algorithms, has been widely used in cryptographic algorithms such as RSA, Diffie–Hellman algorithm, and Elliptic Curve Cryptography. In this paper, negative wrapped convolution (NWC was incorporate into the FFT-based Montgomery modular multiplication to avoid the issue of zero-padding and carry-save were used arithmetic's for parallel computation. By utilizing coefficient pairs (postpartum and neg_part), the final result was reconstructed and eliminate the restrictions imposed by nega-cyclic parts. Moreover, Karatsuba-like algorithm is introduced for building fine-grained large integer multipliers. The parameter specifications for the design were modified to meet requirements from diverse application scenarios. The design was implemented on Xilinx Virtex-7 FPGA under different conditions and the result were compared with the state-of-the-art MMM designs. The comparisons confirm that our design has the following characteristics: low latency for process, competitive area-latency-product (ALP), efficient DSP usage, and constant delay, which enhances security against timing attacks.

Balaji, M., et.al [13] presented in today's digital age, speed, and areas are the primary design concerns. Increasing the rate at which multiplication and addition are performed had always been a need for developing cutting-edge technologies. Wallace multipliers, and Dadda multipliers, were among the fastest multipliers used in many processors to accomplish fast arithmetic operations. A novel approach to design a Lookup Table (LUT) multiplier was proposed and implemented in Finite Impulse Response (FIR) filter. To improve the Residue Number System (RNS) based FIR filter's performance, several adders like Carry Look ahead (CLA) adder, Kogge Stone adder (KSA) and proposed adder architectures were implemented. When compared with the 16 tap with 32 bit proposed adder with LUT multiplier, the hardware resource utilization (Logic Elements) decreased by 5.97 % and in 32 tap with 16 bit combination, it decreased by 7.60 %.When compared with 32 tap with 4 bit word length, the proposed adder with LUT multiplier in the highlighted combinations, increased the Fmax by 19.28 % and in 32 tap with 16 bit, it increases by 29.74 %.

3. EXISTING SYSTEM

The baurn multiplier is a digital circuit used for binary multiplication, providing a systematic and efficient approach to compute the product of two binary numbers. Its design involves an array of individual processing elements, typically full adders, arranged in a grid-like structure. This organized arrangement enables parallel processing, significantly reducing the overall multiplication time.

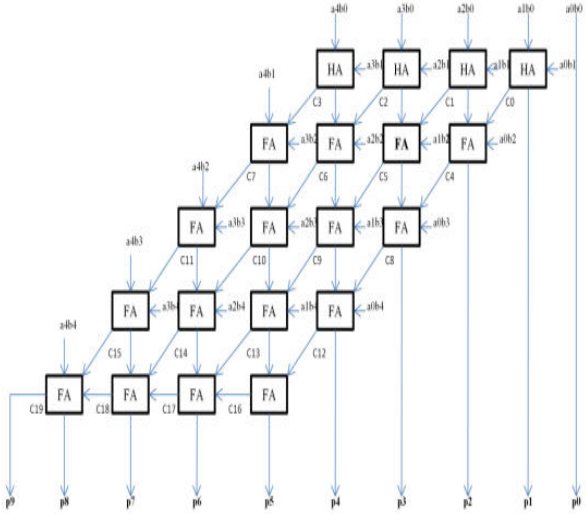


Figure: 3.1 Baurn multiplier

In the baurn multiplier, each cell within the grid corresponds to a partial product generated from the multiplication of a pair of bits from the

multiplicand and multiplier. These partial products are then added together using a set of interconnected adders, producing the final result. The parallel nature of the array multiplier makes it well-suited for hardware implementations, offering advantages in terms of speed and efficiency.

The efficiency of the baurn multiplier is particularly notable in scenarios where large binary numbers need to be multiplied. The array structure allows for a high degree of parallelism, enabling faster computations compared to sequential methods. Despite its effectiveness, the array multiplier's hardware complexity can increase with the size of the binary numbers involved, making it essential to strike a balance between computational speed and hardware resources in practical applications.

4. PROPOSED SYSTEM

Systolic architectures, inspired by the human cardiovascular system, are designed to achieve parallelism and pipelining in data processing. They are characterized by a regular grid of processing elements that collaboratively compute and propagate results through a structured flow of data.

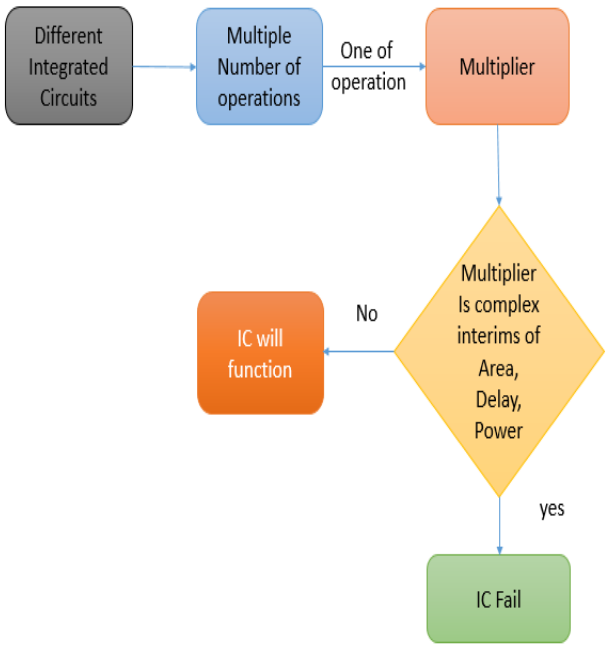


Figure.4 1. Systolic multiplier

When combined with hybrid multiplexer-dependent adders, which intelligently select and combine different addition techniques based on the operands, these systolic multipliers can deliver superior performance in terms of speed and area efficiency. The hybrid multiplexer-dependent adder introduces flexibility in selecting the most suitable adder architecture for a specific set of operands. This adaptability enables the systolic multiplier to dynamically adjust its operation based on the characteristics of the input data, minimizing the overall delay and power consumption.

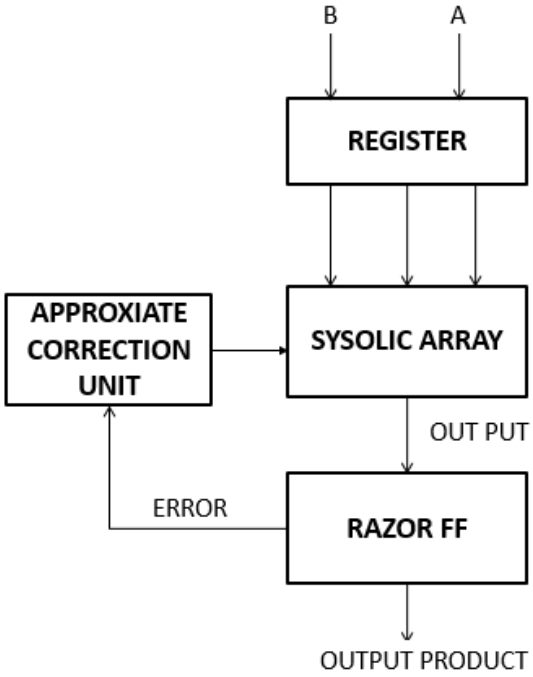


Figure 4.2. Systolic multiplier

A systolic multiplier is a specialized hardware component used for rapid multiplication of large numbers, commonly employed in digital signal processing applications. It operates within a self-checking adders based Systolic Array architecture, where data flows through a grid of interconnected processing elements. Each element conducts partial multiplication and addition operations, with results cascading to adjacent elements, facilitating parallel computation. This arrangement enables efficient handling of repetitive multiplication tasks such as matrix multiplication and digital filtering. Systolic multipliers offer notable advantages in terms of speed and efficiency compared to conventional methods, making them valuable for various computational tasks. They are often implemented in dedicated integrated circuits like ASICs or FPGAs to achieve optimized performance across different computing applications.

STEP-1: The initial step involves considering the input data, which comprises partial products derived from the bitwise AND operation between constants 'a' and an increasing set of values 'b'. The partial products, denoted as PP0 and PP1, are the results of the AND operation between the corresponding bits of 'a' and 'b'. Additionally, there is a carry input denoted as Cin.

STEP-2: The heart of the MDFA is the Full Adder, which is a fundamental building block in binary arithmetic. In the context of MDFA, the Full Adder is employed to process the partial products and generate the sum and carry-out (Cout). The sum is obtained through the XOR operation on the partial products and the carry-in (Cin), while the Cout is computed through a combination of AND and OR operations on the partial products and the carry-in.

Mathematically, the Full Adder operates as follows:

$$Sum = PP0 \oplus PP1 \oplus Cin$$

$$Cout = (PP0 \wedge PP1) \vee (PP1 \wedge Cin) \vee (Cin \wedge PP0)$$

These equations encapsulate the binary addition and carry generation logic within the Full Adder.

STEP-3: The Multiplexer 21 plays a crucial role in the MDFA design, acting as a selector between two possible outputs based on a control input. In the MDFA context, the control input is derived from the input data. The operation of the Multiplexer 21 is contingent upon the value of the data input.

If the data input is 0, the output is set to the product bit.

If the data input is 1, the output is determined by the sum produced by the Full Adder.

Mathematically, the operation of the Multiplexer 2*1 is expressed as:

$$Output = \begin{cases} Product\ bit, & if\ Data\ input = 0 \\ Full\ adder\ sum, & if\ Data\ input = 1 \end{cases}$$

This conditional output selection ensures that the MDFA behaves differently depending on the nature of the input data, dynamically adapting to the requirements of binary multiplication.

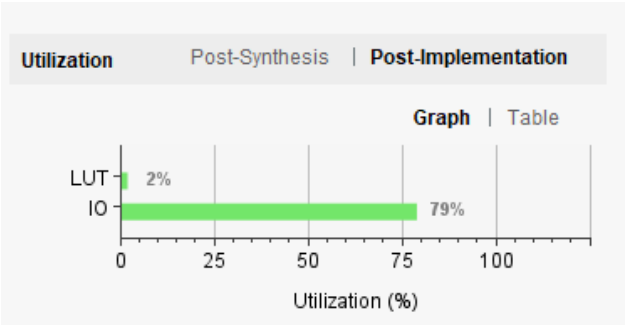
5. RESULTS

Figure 5.1. Shows, Proposed Simulation Result for N=32. Here a &b are the inputs, P indicates the product of the inputs, N indicates the size of the bits.

Figure 5.1. Proposed Simulation Result for N=32

Area

Figure 5.2. To shows the proposed area measurements for N=32. Here, 1790 numbers of LUT's are used out of available 134600 which consume 1.33 of utilization. Here, 128 number of IO are used out of available 500, which consumes 25.60 of utilization.



5.2 Proposed Area for N=32

Figure 5.3 shows proposed power measurement for N=32. Here, the total power is 123.260W, static power includes PL static is 1.235W, Dynamic power includes signal is 31.547W, Logic is 34.297 and I/O is 57.416W

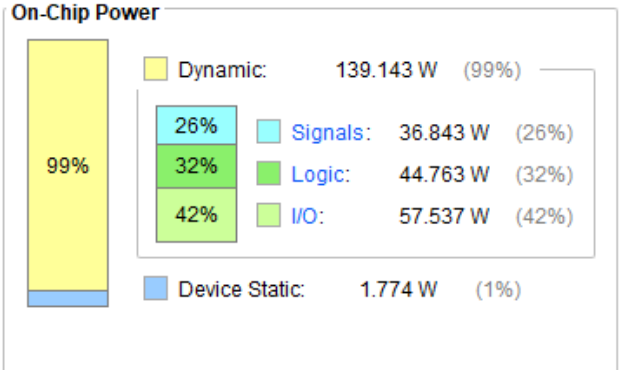


Figure 5.3. Proposed power for N=32

Setup Delay

Figure 5.4.showsProposed setup delay for N=32. Here Total Delay is 91.743, maximum Logic Delay is 17.214, and maximum Net Delay is 74.529.

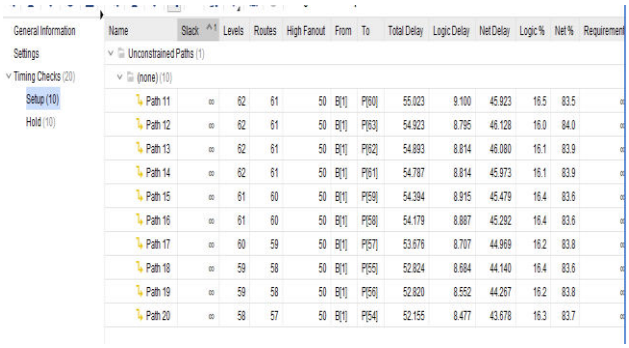


Figure 5.4.Proposed setup delay for N=32

Hold Delay

Figure 5.5.shows Proposed Hold delay for N=32. Here Total Delay is 3.484, maximum Logic Delay is 1.832, and maximum Net Delay is 1.651.

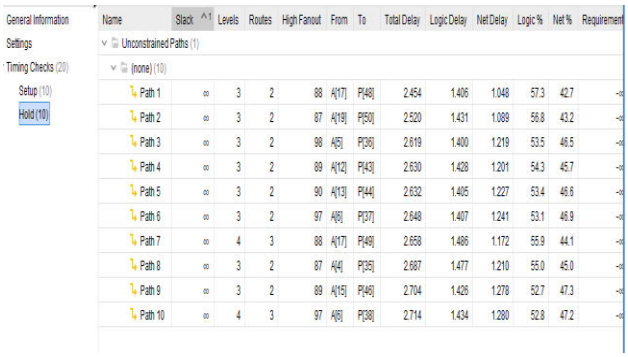


Figure 5.5 Proposed Hold delay for N=32

COMPARISION

Table 5.1.Perform compared of existing and proposed methods for N=32

METRIC	EXISTING	PROPOSED
LTU	131	80
IO	32	32
Static	0.154W	0.15W
Dynamic	14.147W	13.262W
Total delay	14.367	21.601
Logic delay	5.640	6.787
Net delay	13.727	14.816

In the comparison table provided, we evaluate the performance metrics of an existing system against a proposed one, focusing on key parameters. The first metric, LTU (presumably representing some form of processing units), reveals a notable improvement as it reduces from 131 units in the existing system to 80 units in the proposed one. Input/output (IO) remains unchanged at 32 units for both configurations.

Energy consumption is addressed through static and dynamic measurements. The static power, indicating the power consumed even when the system is idle, sees a marginal reduction from 0.154W to 0.15W in the proposed system. However, a more substantial improvement is observed in dynamic power, decreasing from 14.147W to 13.262W, signifying enhanced energy efficiency during active operations.

The total delay in the proposed system is noticeably higher at 21.601 compared to 14.367 in the existing setup. This increase is primarily attributed to higher logic delay (6.787 versus 5.640) and net delay (14.816 versus 13.727). While there is an apparent trade-off in total delay, the proposed system exhibits a refined balance in power consumption, particularly in dynamic power, contributing to potential advancements in energy-efficient computing.

Table 5.2 Perform compared of existing and proposed methods for N=8

METRIC	EXISTING	PROPOSED
LTU	596	344
IO	64	64
Static	0.493W	0.344W
Dynamic	46.175W	37.67W
Total delay	38.818	41.467
Logic delay	7.966	9.831
Net delay	30.852	31.635

In the comparison table, we assess the performance metrics between an existing system and a proposed one, shedding light on crucial aspects of their design. The first metric, LTU (likely denoting processing units), illustrates a substantial improvement in the proposed system, with the count dropping from 596 to 344 units. The Input/output (IO) remains consistent at 64 units for both configurations.

Energy consumption is meticulously examined through static and dynamic power measurements. The static power, representing power consumption at idle states, sees a significant reduction from 0.493W in the existing system to 0.344W in the proposed one. The dynamic power also exhibits a notable improvement, declining from 46.175W to 37.67W, indicating enhanced energy efficiency during active operations.

However, despite the improvements in power efficiency, the total delay in the proposed system slightly increases to 41.467 compared to 38.818 in the existing setup. This rise is primarily attributed to an increase in logic delay (9.831 versus 7.966) and a marginal change in net delay (31.635 versus 30.852). While there is

a trade-off in total delay, the proposed system showcases advancements in energy efficiency, which could be crucial for applications prioritizing power conservation in computing environments.

Table 5.3. Perform compared of existing and proposed methods for N=16

METRIC	EXISTING	PROPOSED
LTU	3109	1790
IO	128	128
Static	1.235W	1.235W
Dynamic	181.046W	1.235W
Total delay	81.139	91.743
Logic delay	12.339	17.214
Net delay	68.800	74.529

In the presented comparison table, we analyze the performance metrics of an existing system and a proposed alternative, offering insights into critical aspects of their design. The first metric, LTU (likely representing processing units), demonstrates a substantial improvement in the proposed system, with the count decreasing significantly from 3109 to 1790 units. Input/output (IO) remains consistent at 128 units for both configurations.

Energy consumption is scrutinized through static and dynamic power measurements. While the static power remains unchanged at 1.235W, a remarkable transformation is observed in dynamic power. In the proposed system, dynamic power drops dramatically from 181.046W in the existing system to 1.235W, indicating a substantial enhancement in energy efficiency during active operations.

Despite the improvements in energy efficiency, the total delay in the proposed system experiences a modest increase, rising to 91.743 compared to 81.139 in the existing setup. This is primarily due to an increase in logic delay (17.214 versus 12.339) and a slight change in net delay (74.529 versus 68.800). While there is a trade-off in total delay, the proposed system demonstrates a significant leap in energy efficiency, potentially offering advantages for applications emphasizing power conservation in computing environments.

6. CONCLUSION

A three-operand binary adder is the fundamental in wrapping up our exploration of the systolic multiplier, it's clear that we've uncovered a powerful tool in the realm of computational hardware. Through our journey, we've seen how systolic multipliers efficiently perform multiplication operations by exploiting parallelism and utilizing a regular, structured architecture. In essence, systolic multipliers excel in executing multiplication tasks by breaking down the process into smaller, manageable computations and executing them concurrently. This parallel processing capability leads to significant speed improvements compared to traditional sequential methods, making systolic multipliers invaluable in applications requiring fast arithmetic operations. Moreover, we've delved into the various design considerations and optimization techniques employed to enhance the performance of systolic multipliers While our journey has highlighted the strengths and advantages of systolic multipliers, it's important to acknowledge the challenges and limitations they may encounter. Factors such as resource constraints, power consumption, and complexity of implementation pose potential obstacles that need to be addressed in the design and deployment of systolic multipliers. Looking ahead, the future of systolic multipliers seems promising, with ongoing research focusing on advancements in hardware design, optimization techniques, and integration with emerging technologies such as artificial intelligence and machine learning.

REFERENCES

[1]Sadhineni, Harika, S. Josaph, and N. Vaishnavi. "Design and analysis of CMOS

based multiplier using LFSR." In AIP Conference Proceedings, vol. 2512, no. 1. AIP Publishing, 2024.
[2]Srinivas, Chundi Sai, M. S. Manohar, and U. Anusha Rani. "High efficient accurate DL-PO logic multiplier design for low power applications." In AIP Conference Proceedings, vol. 2512, no. 1. AIP Publishing, 2024.
[3]Kim, Sunwoong, Cameron J. Norris, James I. Oelund, and Rob A. Rutenbar. "Area-Efficient Iterative Logarithmic Approximate Multipliers for IEEE 754 and Posit Numbers." IEEE Transactions on Very Large Scale Integration (VLSI) Systems (2024).
[4]Minaeifar, Atefeh, Ebrahim Abiri, Kourosh Hassanli, Mehrzad Karamimanesh, and Farshid Ahmadi. "Energy efficient approximate multipliers compatible with error-tolerant application." Computers and Electrical Engineering 114 (2024): 109064.
[5]Hui, Yajuan, Qingzhen Li, Leimin Wang, Cheng Liu, Deming Zhang, and Xiangshui Miao. "In-Memory Wallace Tree Multipliers Based on Majority Gates Within Voltage-Gated SOT-MRAM Crossbar Arrays." IEEE Transactions on Very Large Scale Integration (VLSI) Systems (2024).
[6]Parmar, Rushik, Khushil Yadav, Gauraangi Anand, and Gaurav Trivedi. "An SNN Inspired Area and Power Efficient VLSI Architecture of Myocardial Infarction Classifier for Wearable Devices." IEEE Transactions on Circuits and Systems II: Express Briefs (2024).
[7]Beura, Srikant Kumar, Sudeshna Manjari Mahanta, Bishnulatpam Pushpa Devi, and Prabir Saha. "Inexact radix-4 Booth multipliers based on new partial product generation scheme for image multiplication." Integration 94 (2024): 102096.
[8]Vakili, Bahareh, Omid Akbari, and Behzad Ebrahimi. "Efficient approximate multipliers utilizing compact and low-power compressors for error-resilient applications." AEU-International Journal of Electronics and Communications 174 (2024): 155039.
[9]Haq, Shams Ul, Erfan Abbasian, Vijay Kumar Sharma, Tabassum Khurshid, and Hanaa Fathi. "Energy-Efficient High-Speed dynamic logic-based One-Trit multiplier in CNTFET technology." AEU-International Journal of Electronics and Communications 175 (2024): 155088.
[10]Rohani, Zahra, and Azadeh Alsadat Emrani Zarandi. "Simulation for a low-energy ternary multiplier cell based on Graphene nanoribbon field-effect transistor." Int. J. Nano Dimens 15, no. 1 (2024): 49-62.
[11]Kuo, Chao-Tsung, and Yao-Cheng Wu. "Area-Power-Delay-Efficient Multi-Modulus Multiplier Based on Area-Saving Hard Multiple Generator Using Radix-8 Booth-Encoding Scheme on Field Programmable Gate Array." Electronics 13, no. 2 (2024): 311.
[12]Ke, Hongfei, Hao Li, and Peiyong Zhang. "High-performance montgomery modular multiplier with NTT and negative wrapped convolution." Microelectronics Journal 144 (2024): 106085.
[13]Balaji, M., and N. Padmaja. "Area and delay efficient RNS-based FIR filter design using fast multipliers." Measurement: Sensors (2024): 101014.
[14]Malathi, L., A. Bharathi, and A. N. Jayanthi. "FPGA design of FFT based intelligent accelerator with optimized Wallace tree multiplier for image super resolution and quality enhancement." Biomedical Signal Processing and Control 88 (2024): 105599.
[15]Yassin, Hoda, Arash Akhoundi, El-Sayed Hasaneen, and Dante G. Muratore. "A Power-Efficient Oscillatory Synchronization Feature Extractor for Closed-Loop Neuromodulation." IEEE Transactions on Circuits and Systems II: Express Briefs (2024).
[16]Keerthi, A., S. Manoj, G. Manjula, and K. Kalshetti. "Low Power High Performance 8bit Vedic Multiplier Using 16nm." International Journal 11, no. 5 (2023).
[17]Krishna, Motkuri, and Bal Chand Nagar. "DTMOS based four-quadrant multiplier/divider with voltage difference transconductance

amplifier." Analog Integrated Circuits and Signal Processing (2024): 1-16.
[18]Arora, Gurmeet. "Desing of VLSI Architecture for a flexible testbed of Artificial Neural Network for training and testing on FPGA." Journal of VLSI circuits and systems 6, no. 1 (2024): 30-35.s
[19]Maleknejad, Mojtaba, Fazel Sharifi, and Hojjat Sharifi. "A fast and energy-efficient hybrid 4–2 compressor for multiplication in nanotechnology." The Journal of Supercomputing (2024): 1-23.
[20]Priyanka, C., Bala Krishna Reddy Annapureddy, J. Lakshmi Prasanna, Bathula Murali Krishna, Banda Sai Sandeep, Chandra Vamsi Bodanki, Rishit Pallav, and K. Durga Bhavani. "High speed FFT computation using symmetric stacking." In AIP Conference Proceedings, vol. 2512, no. 1. AIP Publishing, 2024.
[21]Chhabra, Aakriti, Bhawna Aggarwal, and Raj Senani. "A low-voltage MOS translinear loop, biased using FVF and its applications in realizing square-rooter and squarer circuits
[22]Sona, M. Kivi, and V. Somasundaram. "Vedic multiplier implementation in VLSI." Materials Today: Proceedings 24 (2020): 2219-2230.