Low Power High Accuracy Approximate Multiplier Using Approximate High Order Compressors

T Ramya[1], SK Jafar Ameen[2]

[1]*PG Student, DSCE, Quba College of Engineering & Technology*

[2]*Assistant professor, DSCE, Quba College of Engineering & Technology*

Abstract— **This paper presents a novel approximate multiplier design that achieves low power consumption while maintaining high accuracy. The proposed design leverages approximate high-order compressors to reduce the complexity of partial product generation and accumulation. By relaxing the precision requirements of the compressors, significant power savings are achieved without compromising accuracy. The approximate multiplier is designed using a hybrid approach, combining algorithmic and circuit-level approximations. The proposed approximate multiplier is suitable for error-resilient applications, such as digital signal processing, image and video processing, and machine learning. The design demonstrates an optimal trade-off between power, area, and accuracy, making it an attractive solution for energy-efficient computing.**

Index Terms*— *Approximate Multiplier, Low Power Design, High Accuracy, Approximate Compressors, Digital Signal Processing, Error-Resilient Applications.

I. INTRODUCTION

The contemporary tendency in computer technology is oriented towards enhanced compactness and cost-effectiveness. The progress in VLSI technology has been credited with the attainment of higher chip density and operational frequency. The increase in energy usage has resulted in a close proximity to the limitations of both reliability and cost. Furthermore, when the system is scaled down to the Nano level, it presents various design resilience challenges, including signal integrity, soft errors, and process variability. Moreover, concerns pertaining to energy usage and resilience have a tendency to manifest over time. The situation described above has led to a complex scenario in the advancement of information systems, which is expected to impede future progress. As per the computer system specialists who are trailblazers in the domain, it is crucial to incorporate power consumption and system resilience into every phase of the design process. The meticulous evaluation of logic types holds significant significance in circuit design as it directly impacts power consumption, efficiency, and robustness. The present status of Static CMOS logic is considered inadequate to fulfill the upcoming computational demands. In CMOS technology, there exist two fundamental circuit architectures, namely static logic and dynamic logic. The CMOS technology, which is static in nature, is renowned for its exceptional energy efficiency and resilience. However, it has been observed that in critical and large designs, the speed is considerably reduced. On the other hand, the Domino logic methodology demonstrates noteworthy speed performance, albeit at the cost of consuming a substantial amount of power and exhibiting limited durability. As a result, there exists a necessity for an enhanced digital logic approach and construction that exhibits traits of energy efficiency, rapidity, and resistance to noise.

1.1 Motivation for low power Applications

In contemporary times, a multitude of research endeavours have focused on the development of low-power dependent VLSI systems with the aim of creating diverse computing platforms. The emergence of VLSI technology has facilitated the development and manufacture of low power dependent portable devices such as handheld communication devices, laptops, and personal digital assistants, which have been designed and utilized for diverse applications.

1.2 Problem Statement

The literature review suggests that there exists a possibility to enhance the efficiency of multipliers utilized in cryptography and DSP applications. To begin with, it is imperative to improve the effectiveness of the multiplier with respect to both power and delay. The second approach entails enhancing the performance of the multipliers with respect to power consumption and latency. The dissertation will focus on the development, implementation, and evaluation of approximate technique-based multipliers in relation to their area and power consumption. This will be achieved through the use of qualitative simulation and experimental analysis.

1.3 Approximate Computing as a Field of Research

Approximate computing in the most general sense involves the intentional introduction of error to a computation in order to improve performance. This section provides a survey of approximation techniques applied at all levels of the computing hierarchy. Software-level techniques are discussed in Section 1.3.1, architecture-level techniques are reviewed in Section 1.3.2, and circuit-level techniques are examined in Section 1.3.3. Because this project focuses on the application of approximate computing to arithmetic hardware.

1.3.1 Software-Level Techniques

Software approximation encompasses a wide variety of methods at varying levels of ab- straction. At the higher level, approximation-aware programming languages enable the programmer to directly indicate acceptable levels of accuracy for various computations executed in their program. In the programming language Eon, paths through a program occupy different energy states set by the programmer, which generally correspond to rate of execution priority. Eon’s automatic energy management then dynamically adapts these states according to currently available and predicted energy levels. an extension EnerJ uses type qualifiers to declare data for which approximate computations may be performed. These type qualifiers indicate to the system how the data should be handled, where approximate data is mapped to low-power storage, operated on using low-power computations, etc. The programming language Rely proposed on supports quantitative reliability specifications for results produced by a function, which essentially defines the minimum acceptable reliability for the function to be called. A static quantitative reliability analysis is used to verify the quantitative requirements on the reliability of a program, effectively verifying that the program satisfies its reliability specification when executed on the underlying unreliable hardware.

1.4 Objective of the Work

The Approximate Computing technique-based multipliers are much needed for cryptographic and signal processing applications for improving the power and area performances. The objectives of the research conducted for the dissertation are as follows:

- To design and develop a Radix-4 booth multiplier to enhance the technical parameters of the approximate multiplier circuit.
- To design a Radix-256 based approximate multiplier for high-speed applications.

1.5 Project Organization

The dissertation is broken into six chapters. The first chapter provides background and information for this research, as well as a summary of the relevant topic and work objectives. Chapter 2 explored the problem-related literature. The Chapter 3 involves existing compressors. Chapter 4 explored the implementation of the proposed Approximate Multipliers In the last part 6, the conclusion and future efforts are explored in order to improve the concept even more.

II. LITERATURE SURVEY

According to reference, multipliers are essential arithmetic operations that play a critical role in various fault algorithms. The utilization of approximate multipliers is commonly acknowledged as a superior approach for integrating power, speed, and accuracy. The authors present two methodologies for approximating multipliers with reduced power consumption and latency compared to conventional multiplication techniques, employing approximate tree compressors. The compressor that is recommended has the capability to reduce the size of the partial product trees by 50%, and also provides vectors that enable the restoration of precision. In comparison to traditional Wallace tree multiplication, an 8-bit approximate multiplier that is suggested exhibits a reduction in power consumption and critical path latency by approximately 59.9% and 36.3%, respectively. Furthermore, given a standardized mean error of 0.28%, the total chip size required for the design of the multiplier is decreased by 50.1%. The proposed multiplier topologies exhibit superior performance compared to their predecessors in terms of energy efficiency, critical path latency, and overall layout area.

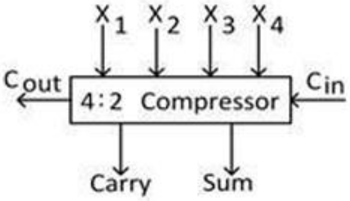
III. EXISTING COMPRESSORS

Arithmetic circuits are essential elements in various fields, such as digital circuits, microelectronics, and DSP applications. Performance metrics such as area, latency, and power are crucial attributes in electronic advancements, and therefore enhancing the modules in these domains is strongly recommended. The aforementioned methodology involved the incorporation of compressors into the sequencing process, in lieu of adders during multiplication operations. However, this approach resulted in increased utilization of silicon space and power consumption due to the partial product reduction step necessitated by the imposition of limitations on the accuracy of the multiplier. Different methodologies such as the modified booth encoding technique 4, ripple carry adders, and carry save adders were employed to minimize the generation of partial products and to decrease the complexity of partial product reduction. The aforementioned designs were rendered obsolete upon the implementation of compressor circuits that restrict carry propagation.

Compressors are increasingly significant components in multiplier designs utilized for the partial product reduction phase. Historically, various types of adders, particularly carry save adders, have been commonly employed for the purpose of reducing partial products. However, in light of the demand for reduced power consumption and a more streamlined design, adders have been progressively supplanted by alternative compressors, including those with ratios of 3:2, 4:2, and 5:2.

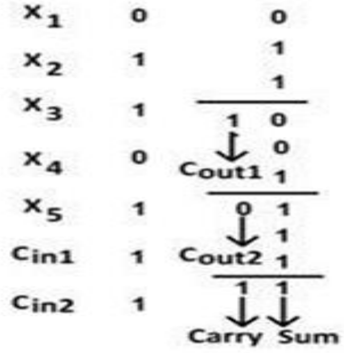
3.1 Existing 4:2 compressor

The compressor constitutes a fundamental component of a multiplier circuit, serving to reduce partial products. The basic form of data compression is achieved through employment of a full adder, which functions as a 3:2 compressor by condensing three inputs into two outputs. The 4:2 compressor, which features five inputs (X1, X2, X3, X4, Cin) and three outputs (Sum, Carry, Cout), has been particularly noteworthy. The schematic in Figure 3.1 illustrates the basic architecture of a 4:2 compressor.



3.2 Existing 5:2 compressor

In addition, a compressor can be defined as a type of digital circuit that takes in binary digits of equivalent weight and generates a single output bit that comprises multiple carry bits. One notable distinction between a compressor and an adder is that the former combines multiple bits 22 of equal significance, while the latter combines two inputs of differing significance. The operational mechanism of a 5:2 compressor is illustrated in Figure 3.5, as presented in reference.



IV. PROPOSED APPROXIMATE MULTIPLIERS

The paradigm of approximate computing has emerged as a significant engineering approach for numerous applications, including signal processing, multimedia computation, and deep learning. This approach enables the reduction of performance metrics such as power, latency, and area, similar to any circuit. Compressors serve as significant add-ons to the multiplier circuit, facilitating the compression of partial products and enhancing the overall circuit's speed. The present article showcases diverse configurations of 5:2 compressors. The present study showcases the utilization of imprecise 5:2 and 4:2 compressors in the development of an approximate multiplier.

4.1 Introduction

The process of multiplication involves three distinct stages, namely the generation and reduction of partial products, followed by the addition of all partial products. It is worth noting that the second stage of this process consumes the most chip area, power, and time.

Several methodologies, such as the modified booth encoding technique, ripple carry adders, and carry save adders, have been employed to minimize the generation of partial products and decrease the complexity of partial product reduction circuitry. The aforementioned designs were rendered obsolete following the implementation of compressor circuits, which restrict carry propagation. A compressor is a type of logic circuit that receives input bits of equal significance and generates a single Sum bit along with multiple Carry bits as output. One notable distinction between a compressor and an adder is that the former performs addition on multiple bits of equal significance, while the latter performs addition on two operands of differing significance. Figure 4.1 below illustrates an instance of a 5:2 compressor operation. The process of adding a 5:2 compressor involves the utilization of three Full Adders. The initial Full Adder (FA1) performs the addition operation on X1, X2, and X3, resulting in the generation of Carry1 and Sum1. The Carry1 is subsequently utilized as Cout1. The Full Adder denoted as FA2 performs the arithmetic operation of adding the binary digits of Sum1, X4, and X5, resulting in the generation of Cout2 and Sum2. The Carry and Sum outputs of a 5:2 compressor are obtained by incorporating Sum2 of FA2, Cin1, and Cin2. The increasing demand for low power architectures has led to a growing interest in inexact circuits, which prioritize energy/power, delay, and area over exactness of output, while still maintaining an acceptable level of accuracy.

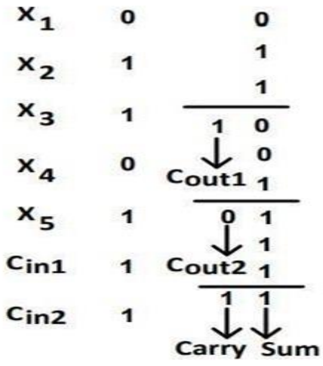


Figure 4.1: 5:2 Compressor Example

4.2 Proposed Approximate 5:2 Compressor

The present section deliberates on a novel 5:2 compressor and the suggested multiplier that employs it. The compressor depicted in this illustration employs the technique of probabilistic pruning to eliminate extraneous nodes and their corresponding interconnections from a precise 5:2 compressor. The 5:2 compressor, which utilizes XOR-XNOR gates and 2x1 Mux, incorporates the removal of an XNOR terminal in the supplementary XOR-XNOR gate. This guarantees that one of the input signals of the multiplexer remains grounded. The diagram presented in Figure 4.2 depicts an analogous representation of a contemporary compressor with a 5:2 ratio.

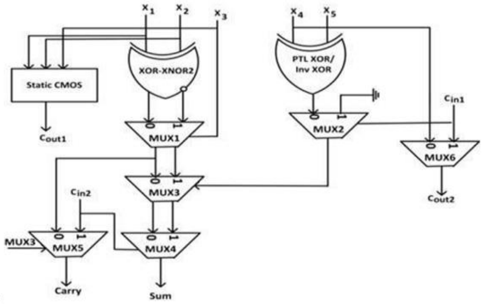
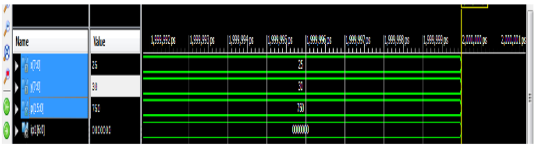


Figure 4.2 Inexact 5:2 Compressor using Probabilistic Pruning

V. SIMULATION RESULTS

The various 8 X 8 multipliers outlined above have been synthesized and simulated using an FPGA board utilizing the Xilinx Spartan 7 XC7S15- 1FTGB196C architecture. The inputs of the multiplier are connected to the input switches of the FPGA and outputs of the multiplier are connected to the LEDs present on the FPGA. For the implementation of the existing and proposed multipliers on to FPGA board, Xilinx ISE is used. By varying the switches on the FPGA, the corresponding LEDs are verified. However, this technique raises the delay time greater than the existing methods. In this case, simulation is performed for multipliers with an 8-bit width.



Name	Value
x[7:0]	25
y[7:0]	30
p[15:0]	750
tp1[6:0]	0000000

Figure 5.1 Simulation results of 8X8 Array Multiplier

A multiplication using the array multiplier with inputs X and Y as 8 bits each is depicted in Figure 5.1. The output result of the multiplication process is represented by P.

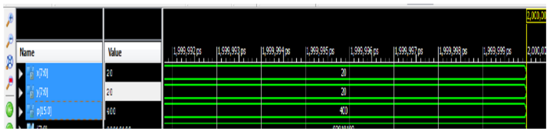
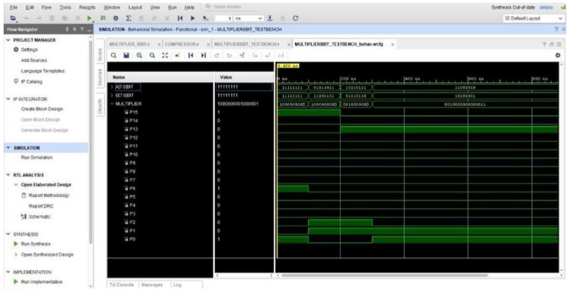


Figure 5.2 Simulation results of 8X8 Booth Multiplier

A multiplication using the Booth multiplier with inputs X and Y as 8 bits each is depicted in Figure 5.3. The output result of the multiplication process is represented by P.



A multiplication using the proposed multiplier with inputs X and Y as 16bits each is depicted in Figure 5.3. The output result of the multiplication process is represented by P. The multiplier implementations are synthesized using Xilinx vivado/ISE, and also the resulting delay & area parameters are shown in Table 5.1. Table 5.1 clearly show that the proposed Multiplier occupies less area than the other designs.

Table 5.1 Performance analysis of various multipliers

Multiplier Name	Area	Power(mw)
Existing Wallace tree Booth Multiplier	877	0.693
Existing Booth Approximate Multiplier	350	0.311
Proposed Approximate Multiplier	209	0.042

VI. CONCLUSION AND FUTURE SCOPE

Conclusion

The Project demonstrated a low-power high-accuracy approximate 8 x 8 multiplier design. To achieve high accuracy, we use accurate (i.e., exact) 4:2 compressors in the higher significance weights. To reduce power consumption, we use high-order approximate compressors in the middle significance weights. Compared with the Dadda tree multiplier, experimental results show that the proposed approximate multiplier design can save 80 percent of area.

Future Scope

Furthermore, when the speed of technological advancement quickens, new inventive solutions and design elements may be created to fulfil technological scalability standards. Furthermore, as CMOS technology progresses, researchers may concentrate on CMOS IC design methodologies with a high degree of assurance and power efficiency for wireless healthcare and medical application sectors. There is space to create building techniques at both the circuit block and system levels throughout the whole foundation of technical breakthroughs. Approaches for designing ultra-low- power CMOS circuits for the full network impact of wireless health and medical care that are strategically positioned at low-technological nodes

REFERENCES

[1] Suganthi.V, et.al, "Design of Power and Area Efficient Approximate Multipliers," IEEE Transactions on VLSI Systems, 2016, pp.1-5.

[2] Darjn.E, et.al, "Approximate Multipliers Based on New Approximate Compressors," IEEE Transactions on CAS-I: Regular Papers, PP (99), 2018, pp. 1-14.

[3] Avinash.L, et.al, "Parsimonious Circuits for Error-Tolerant Applications through Probabilistic Logic Minimization", International Workshop on PATMOS 2011, pp.204213.

[4] Liang.J, et.al, "New Metrics for the Reliability of Approximate and Probabilistic Adders," IEEE Transactions on Computers, 63(9), 2013, pp. 1760-1771.

[5] Zervakis.G, et.al, "Design-Efficient Approximate Multiplication Circuits through Partial Product Perforation," IEEE Trans. on VLSI Systems, 24(10), 2016, pp. 3105-3117.

[6] Peter. A, et.al, "Opportunities for Machine Learning in Electronic Design Automation," in ISCAS, 2018, Italy.

[7] Veeramachaneni, et.al., "Novel architectures for high-speed and low-power 3-2, 4-2 and 5- 2 compressors," Proc. Int. Conf.