Irish Interdisciplinary Journal of Science & Research (IIJSR)
Volume 7, Issue 3, Pages 98-105, July-September 2023

Design of Accuracy Configurable Booth Multiplier using Sorting based Compressors

Jason, K.[1*], Arivazhagan, P.[2] & Arunkumar, P.[3]

[1]PG Scholar, [2,3]Assistant Professor, [1-3]Department of Electronics and Communication Engineering, Sir Issac Newton College of Engineering and Technology, Nagapattinam, Tamilnadu-611 102, India.
Corresponding Author Email: json.bht@gmail.com*

DOI: https://doi.org/10.46759/IIJSR.2023.7311

Article Received: 11 July 2023 *Article Accepted: 22 September 2023* *Article Published: 30 September 2023*

ABSTRACT

An approximate computing technique has been applied in recent years to develop low-power design solutions for fault-tolerant applications. Since an application's accuracy requirements can change dynamically at runtime, it is necessary to design reconfigurable approximate circuits with varying power requirements proportional to computational accuracy. In this work, a new approximate booth multiplier circuit capable of accurately reconfiguring compressor based on sorting network has been designed. 4:2 compressor-based sorting network that minimizes propagation path delay during partial product reduction. The design involves partial error correction through the addition of sign bits in a broken array multiplier. The design space of the 16-bit broken array multiplier, sign included, is deeply analyzed to include only non-redundant precision modes. The horizontal and vertical breaks introduced into the booth multiplier circuit are controlled by external control signals stored in the ROM. The proposed reconfigurable multiplier provides significant power savings compared to state-of-the-art accurate multiplier circuits and precision configurable multiplier designs. The proposed ACBAM multiplier using a classification network achieves 57.81% efficiency in area and 68.61% efficiency in latency compared to the current system.

Keywords: Approximate Computing; Booth multiplier; Sorting Based Compressor.

1. Introduction

There are different types of multipliers used in computing, but we use booth multipliers because we can perform signed multiplications, and we can perform partial product reduction using modified booth encoding algorithms. Approximations are used here to reduce the area, the power and the delay in an architecture like signal and image processing application where there would be a need for more adders or multipliers and to reduce the hardware complexity. This approximate booth multipliers are used in error tolerant application. The Reconfigurable multipliers can be dynamically reconfigured during run time according to the accuracy requirements of the application [1-9]. Approximate computing is a promising approach to designing energy-efficient digital systems, as it allows for the tolerance of loss of quality or optimization in calculated results. 'Approximate Computing: An emerging paradigm for energy-efficient design' discusses recent advances in approximate arithmetic block design, error and quality measurements, and algorithm level techniques for approximate computing [10].

Speculative adders reduce critical path delays to sub-logarithmic delays by harnessing the tradeoffs between reliability and performance. The design of reliable variable latency adders combines speculative adders with error correction to achieve high performance for low area overlay compared to traditional adders. The authors describe Speculative Carry Select Adders (SCSA), a new function speculation technique to design low error rate speculative adder and low overhead, high performance, reliable variable latencies adders [11-12]. The authors propose a low-error, reduced-width Booth multiplier that uses the right compensation vector based on input data. The design results show that both the gate count and key path delay of these new reduced-width multipliers are 50.94% and 66.04%, respectively. They also create a module generator based on their proposed architecture that generates C code and Verilog codes for each of the reduced width multipliers [13-19].

Fixed-width multipliers are attractive to many media and digital signal processing systems that want to keep a fixed format and a small loss of accuracy to output data. The authors design high-accuracy, fixed-width modified Booth multipliers that reduce truncation error and derive an efficient error compensation function [11].

Approximate calculation reduces design complexity while improving performance and energy efficiency for error-resistant applications. The authors discuss a novel design approach for approximating multipliers, which introduces different probability terms and varies the logic complexity of approximation according to their probability [20].

Partial Product Perforation (PPP) is a promising approach to reduce power consumption of error-tolerant applications. The authors conduct extensive experimental testing using different multiplier architectures and reveal the optimal architecture-perforation configuration pair for various error constraints [8].

Approximate computing allows for the reduction of the requirement of accuracy by reducing the complexity of the circuit by reducing power consumption. 'Design of Approximate Radix-4 Booth Multipliers for Error-Tolerant Computing', approximate radix-4 modified Booth encoders (MBE) are proposed and analyzed [21-23].

1.1. Study Objectives

Digital signal processing relies heavily on multipliers and adders, which affect the speed of operations. Faster multipliers are recommended for high-speed applications. To increase multiplier speed, techniques are proposed, such as adding partial product terms. The implementation of multipliers involves three steps: generation of partial products, reduction tree generation, and vector merge addition. The second step consumes more power, but is applied in the reduction tree stage. Partial product accumulation in 4 x 4 multipliers is represented by two unsigned operands, α and β, and the product is denoted by γ. The bits of γ are produced in stages from the LSB to the MSB, with no further operation in Stage 0. A 4:2 compressor is required for generating and carrying in Stage 2.

Section 2 provides a Low Error Fixed Width Booth Multiplier. Section 3 is about the Result and Discussion and Section 4 is about Conclusion.

2. Accuracy configurable booth multiplier using sorting-based compressors

Booth's Algorithm is a multiplication algorithm that utilizes two's complement notation for signed binary numbers. It allows for both addition and subtraction, and can be implemented in multiple ways. Multiplication is a series of repeated additions, with the multiplier being the number to be added and the multiplier the number of times it is added. The product is generated after each step of addition, which is usually twice the length of operands to protect information content. Multipliers can be decomposed into two parts: the first part generates partial products, and the second part collects and adds them. The fundamental multiplication principle involves evaluating partial products and gathering shifted partial products through consecutive additions of columns of the shifted partial product matrix. The delayed, gated case of the multiplicand must all be in the same column of the shifted partial product matrix, then added to form the product bit for the particular form.

To expand multiplication to both signed and unsigned numbers, a suitable number system would be the depiction of numbers in two's complement format. A circuit that multiplies two unsigned n bit binary numbers uses a

2-dimensional array of identical sub-circuits, each containing a full adder and an and gate. However, this approach may not be appropriate for large numbers of bits due to the large number of gates needed. Booth's algorithm conserves the sign of the end result by shifting strings of 0s in the multiplier and adding or subtracting only at positions where there is a switch from 0 to 1 or from 1 to 0.

The pre-processing unit takes input Y and generates groups of 3 bits yi'= {y2i+1, y2i, y2i-1} depending on control signal k 1:0. By default, for any possible value of k, all three bits of y0 0 are set to zero. For k = 1, only bits corresponding to y0 are set to zero which is achieved by setting both control signals k1 and k0 to 1. All bits of y1' are forced to zero for k = 2, by setting {m3 = 1} and {m1=0}.

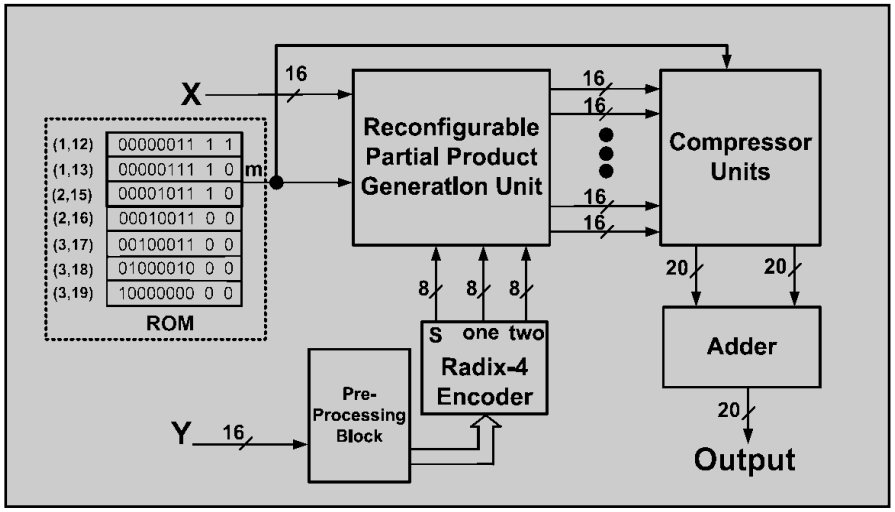

Figure 1. Block diagram of ACBAM

Thus, the partial product row corresponding to bits {y3; y2; y1} becomes zero. Similarly, for k = 3, all the bits of both groups y1'and y2' are forced to 0 by setting both control signals to 0. The pre-processing unit requires six AND gates. The reconfigurable PPG block drives all the partial products on the left of the vertical break to zero. The signal ppi;t corresponds to the tth partial product of (i+1)th row, where i ∈ [0, 7]. Here t refers to the (12-2i)th bit of input X corresponding to ith partial product row.

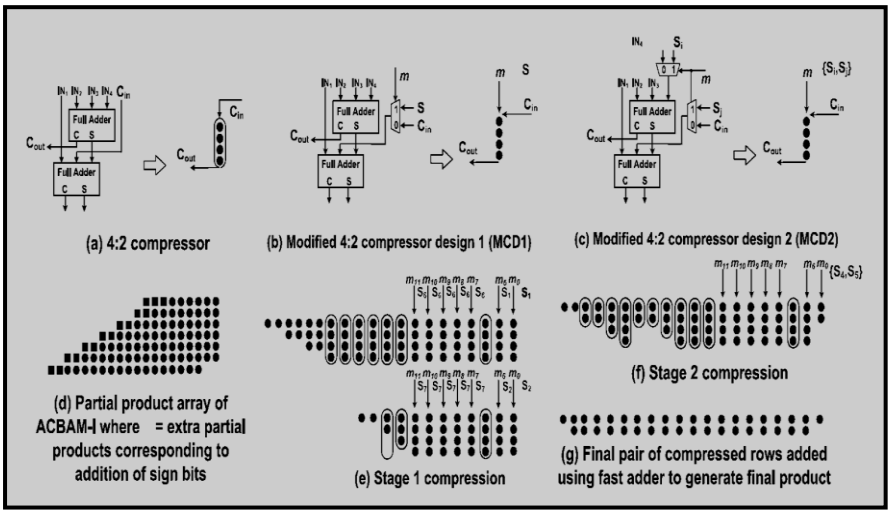

Figure 2. ACBAM-I Design

For m =12, all control signals from m0 to m5 are set to 1. When m is chosen as 13, m0 is set to 0 which means ppi,12 is forced to zero in each ith row. The AND gates placed in front of each PPG block ensures the partial product is forced to 0. The sorting network is a parallel hardware network used for data sorting, based on the famous 0,1 principle. This principle states that if a sorting network can sort a group of data with all 1-bit numbers, it can sort all types of numbers. This work focuses on 1-bit data sorting, using vertical lines representing sorters with two data inputs and outputs.

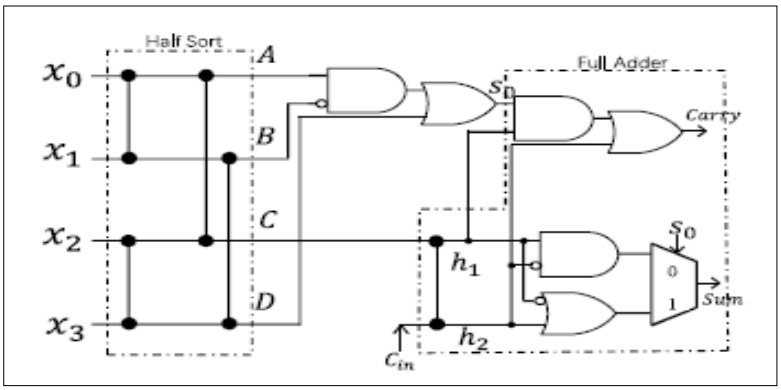

Figure 3. 4:2 Sorting based compressor

The sorter always puts the larger input up, the smaller one down. An example is given for four-way sorting network (4 SN) and three-way sorting network (3 SN). The input sequences are reordered after three layers of sorter.

3. Results and Discussion

The segmentation process is shown below where the 16x16 operands are used where the segmentation is done and the decimal values are shown as output. The proposed reconfigurable 16-bit multiplier ACBAM is designed using sorting-based compressors. Figures 4 and 5 shows the simulation result of ACBAM with and without sorting network-based compressors. X and Y is the input operands of the multiplier and OP is the product. SEL is the selection line for selecting the ROM data which is used for performing different modes of operation.

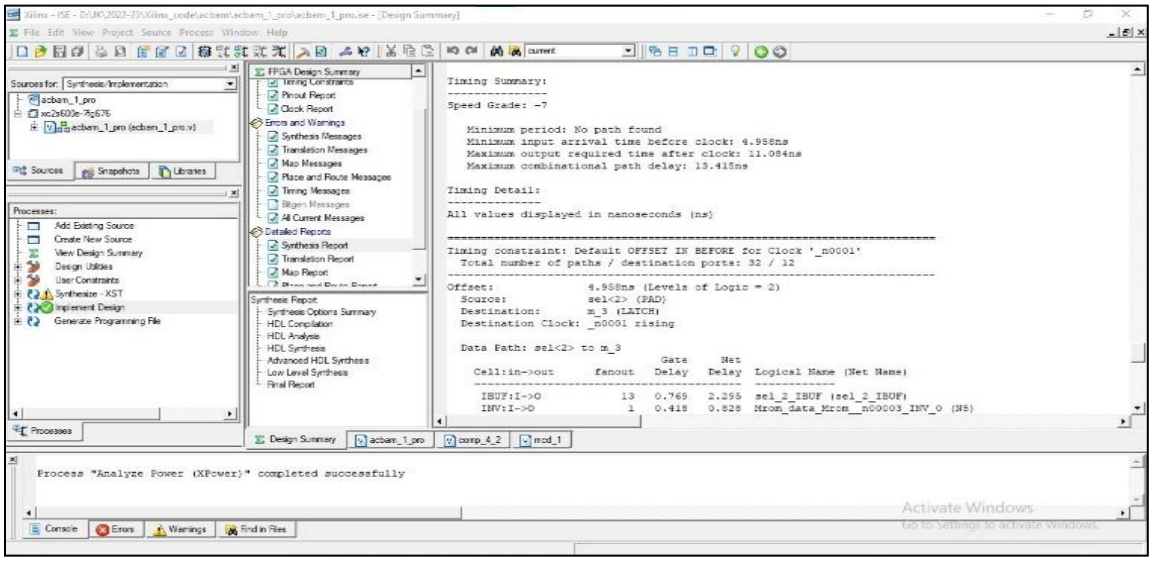

Figure 4. Design Summary of Proposed 16x16 ACBAM Multiplier with sorting network using Xilinx

The multiplier designs are implemented in Verilog- HDL and synthesized with Xilinx ISE. The report also demonstrates the area utilization, power consumption and delay analysis of the existing and proposed method. The proposed sorting-based compressor used for partial product result in reduction of all three parameters compared to existing design.

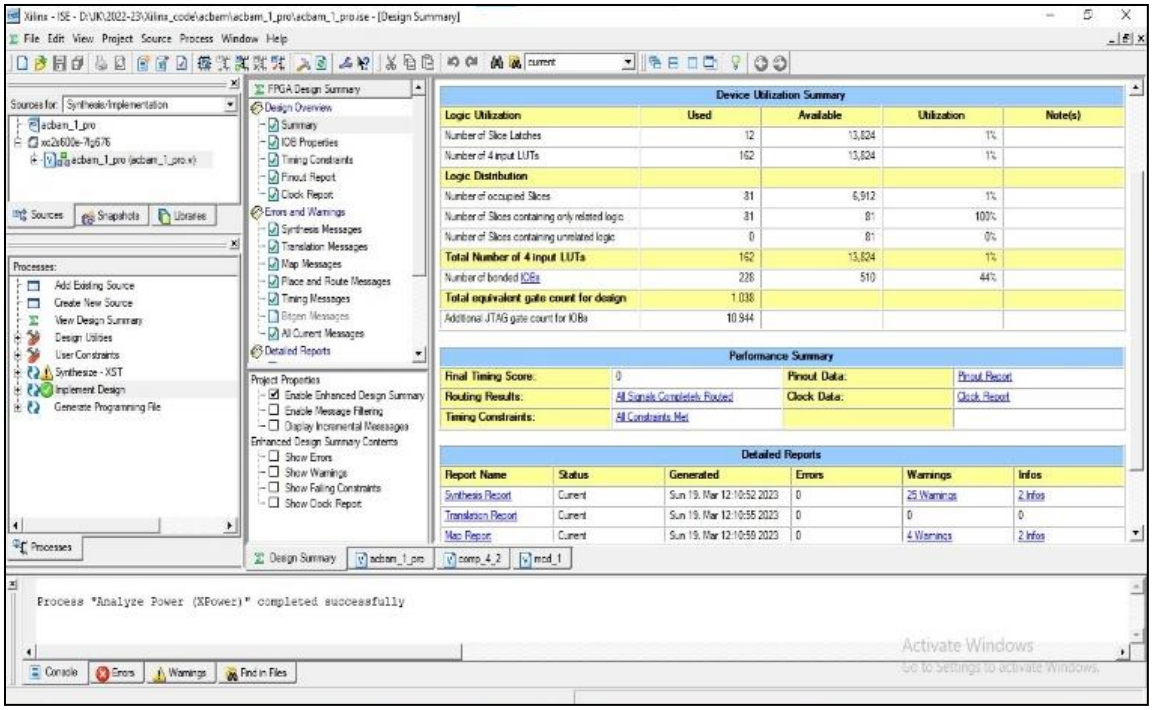

Figure 5. Timing Report for 16 x 16 ACBAM Multiplier using Xilinx

The proposed ACBAM Multiplier using Sorting network achieves better performance of 57.81% in terms of Area and 68.61% in terms of delay when compared with the existing system.

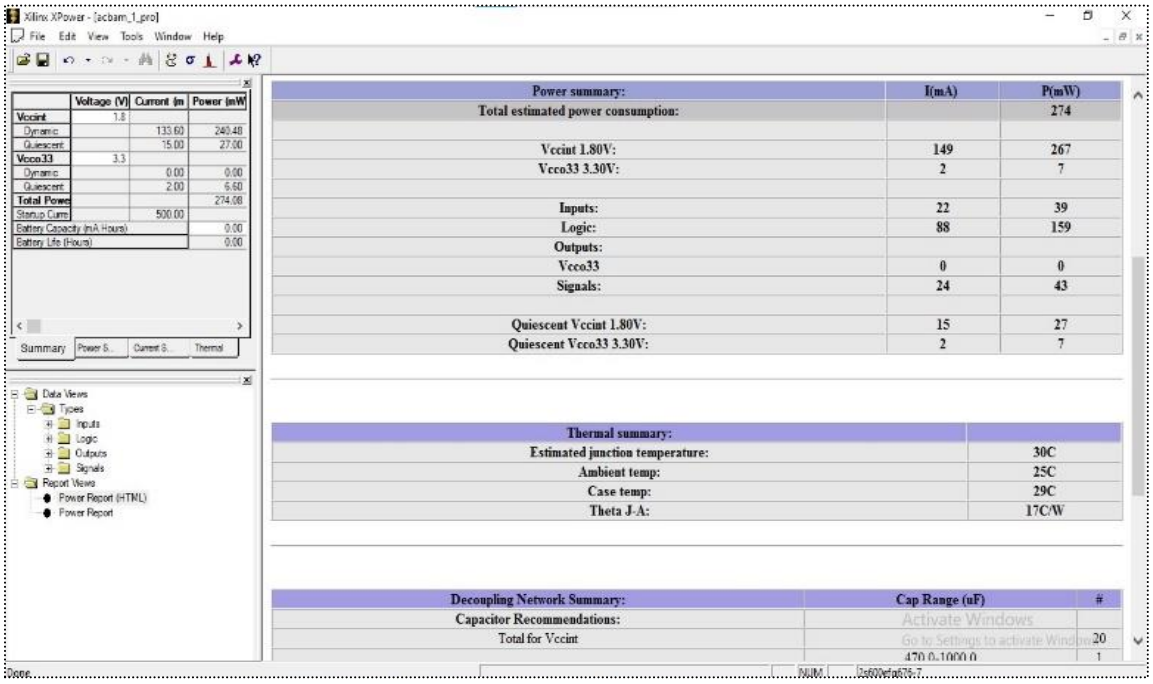

Figure 6. X Power Analyzing Report for 16 x 16 ACBAM Multiplier using Xilinx

Irish Interdisciplinary Journal of Science & Research (IIJSR)
Volume 7, Issue 3, Pages 98-105, July-September 2023

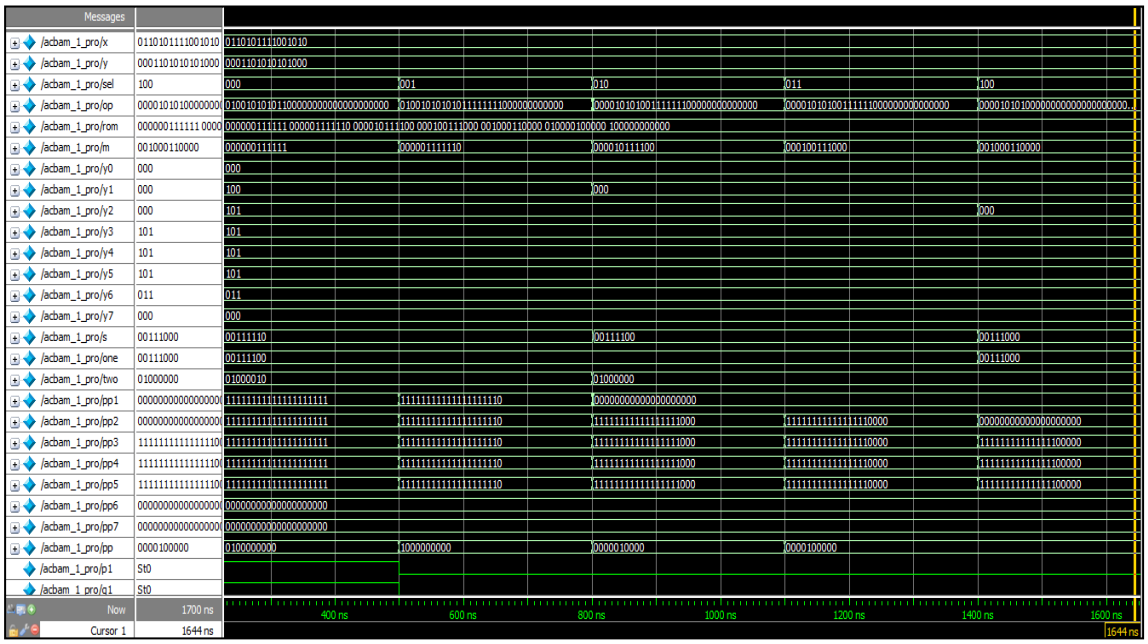

Figure 7. Simulation waveform for proposed ACBAM Multiplier with sorting network using ModelSim Tool

Table 1. Performance analysing Report

Types	Area			Delay(ns)	Power (mW)
	Slices	LUT	Gate		
Existing System	192	354	2,184	42.740	289.61
Proposed System	81	162	1038	13.415	274.08

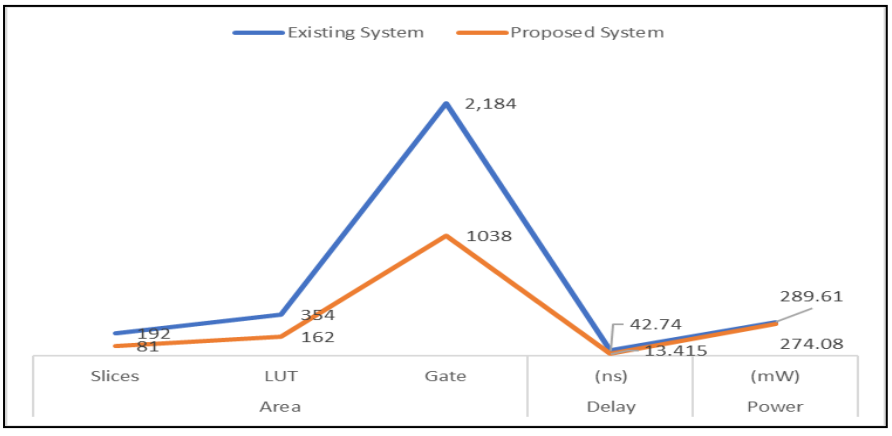

Figure 8. Performance Analysis Graph

4. Conclusion

The booth multiplier operation is performed using multisim software and the power, area and delay are calculated using the Xilinx suite where we could analyze that compared to the ACBAM using sorting compressors, in static segmentation-based booth multiplication the area, power, delay are more efficiently used.

We have also developed simple and effective error correction techniques, able to significantly reduce the approximation error, at reduced hardware cost. And these could be used in varying applications involving image processing. Design analysis shows that the proposed design outperforms the accurate multiplier with energy savings to a greater percentage.

Declarations

Source of Funding

This study has not received any funds from any organization.

Conflict of Interest

The authors declare that they have no conflict of interest.

Consent for Publication

The authors declare that they consented to the publication of this study.

Authors' Contribution

All the authors took part in literature review, research, and manuscript writing equally.

References

[1] Jay Kumar S.R., Arivazhagan P., Saranya K., & Manikandaprabu N. (2014). Genetic Algorithm based test Pattern Generation for Asynchronous Circuits with Handshake Controllers. Unique Journal of Engineering and Advanced Sciences, 2(1).

[2] Ramesh J., Arivazhagan P., & Thiruvenkadam K. (2015). Design of high speed and low power modified booth add multiply operator for FFT application. International Journal of Applied Engineering Research, 10(29).

[3] Suganthi M., Kavitha M., & Arivazhagan P. (2022). An Efficient VLSI Design of fixed width booth multiplier for edge detection scheme. International Journal of Novel Research and Development, 7(9).

[4] Avishek Sinha Roy, Hardik Agrawal & Anindya Sundar Dhar (2022). ACBAM – Accuracy – Configurable Sign Inclusive Broken Array Booth Multiplier Design. IEEE Transactions on Emerging Topics in computing, 10(4).

[5] Georgios Zervakis, Kostas Tsoumanis, Sotirios Xydis, Dimitrios Soudris & Kiamal Pekmestzi (2016). Design-Efficient Approximate Multiplication Circuits Through Partial Product Perforation. IEEE Transactions on Very Large Scale Integration (VLSI) Systems.

[6] Ha M., & Lee S. (2018). Multipliers with approximate 4–2 compressors and error recovery modules. IEEE Embedded Syst. Lett., 10(1): 6–9.

[7] Jiang H., Han J., Qiao F., & Lombardi F. (2016). Approximate radix-8 booth multipliers for low-power and high-performance operation. IEEE Trans. Comput., 65(8): 2638–2644.

[8] Jiang H., Liu C., Lombardi F., Leibo L., & Han J. (2017). A review, classification, and comparative evaluation of approximate arithmetic circuits. ACM J. Emerg. Technol. Comput. Syst., 13(4): 1–34.

[9] Jiao X., Jiang Y., Rahimi A., & Gupta R.K. (2017). SLoT: A supervised learning model to predict dynamic timing errors of functional units. In Proc. Conf. Des. Automat. Test Eur., Pages 1183–1188.

[10] Jie Han & Michael Orshansky (2013). Approximate Computing: An Emerging Paradigm For Energy-Efficient Design. 18th IEEE European Test Symposium (ETS).

[11] Jiun-Ping Wang, Shiann-Rong Kuang & Shish-Chang Liang (2011). High-Accuracy Fixed-Width Modified Booth Multipliers for Lossy Applications. IEEE Transactions on Very Large-Scale Integration (VLSI) Systems, 9(1).

[12] Kai Du, Peter Varman & Kartik Mohanram (2012). High Performance Reliable Variable Latency Carry Select Addition.

[13] Kulkarni P., Gupta P., & Ercegovac M. (2011). Trading accuracy for power with an under designed multiplier architecture. In Proc. 24th Int. Conf. VLSI Des., Pages 346–351.

[14] Leon V., Zervakis G., Soudris D., & Pekmestzi K. (2018). Approximate hybrid high radix encoding for energy-efficient inexact multipliers. IEEE Trans. Very Large Scale Integr. Syst., 26(3): 421–430.

[15] Leon V., Zervakis G., Xydis S., Soudris D., & Pekmestzi K. (2019). Cooperative arithmetic-aware approximation techniques for energy-efficient multipliers. In Proc. ACM/IEEE. Des. Automat. Conf., Pages 1–6.

[16] Leon V., Zervakis G., Xydis S., Soudris D., & Pekmestzi K. (2018). Walking through the energy-error pareto frontier of approximate multipliers. IEEE Micro, 38(4): 40–49.

[17] Ragavan R., Barrois B., Killian C., & Sentieys O. (2017). Pushing the limits of voltage over-scaling for error-resilient applications. In Proc. Des., Automat. Test Eur. Conf. Exhib., Pages 476–481.

[18] Roy A.S., & Dhar A.S. (2010). SIBAM-Sign inclusive broken array multiplier design for error tolerant applications. IEEE Trans. Circuits Syst. II: Exp. Briefs, 67(11): 2702–2706.

[19] Shyh-Jye Jou, Meng-Hung Tsai, & Ya-Lan Tsao (2003). Low-Error Reduced-Width Booth Multipliers for DSP Applications. IEEE Transactions on Circuits and Systems—I: Fundamental Theory And Applications, 50(11).

[20] Suganthi Venkatachalam & Seok-Bum Ko (2017). Design of Power and Area Efficient Approximate Multipliers. IEEE Transactions on Very Large Scale Integration (VLSI) Systems.

[21] Weiqiang Liu, Liangyu Qian, Chenghua Wang, Honglan Jiang, Jie Han & Fabrizio Lombardi (2016). Design of Approximate Radix-4 Booth Multipliers for Error-Tolerant Computing. IEEE Transactions on Computers.

[22] Wenbo Guo & Shuguo Li (2021). Fast Binary Counters and Compressors Generated by Sorting Network. IEEE Transactions on Very Large-Scale Integration Systems, 29(6).

[23] Yuan-Ho Chen & Tsin-Yuan Chang (2012). A High-Accuracy Adaptive Conditional-Probability Estimator for Fixed-Width Booth Multipliers. IEEE Transactions on Circuits and Systems—I: Regular Papers, 59(3).