International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435(Online) Volume 12, Issue 06, June 2023)

VLSI Implementation of Booth Multiplier for FPGA-IOT Applications: A Review

Shreepad Ganjiwale[1], Deepali Sahu[2], Vivek Bhargava[3]

[1]M.Tech Scholar, [2,3]Assistant Professor, Department of Electronics Engineering, Adina Institute of Science & Technology Sagar, India

***Abstract*— The many arithmetic operations, including multiplication, addition, and subtraction, are an essential component of digital circuits, since they help to accelerate the processing speed of the processor. Nevertheless, the multiplier unit of the CPU has a significant impact on the speed of the processor. The implementation of a digital multiplier may be accomplished via the use of a variety of computer arithmetic approaches. Many scholars have attempted, and continue to attempt, to construct multipliers in light of recent developments in technology. In this research, a review of the VLSI implementation of booth multiplier for FPGA-IOT applications is presented.**

***Keywords*—FPGA-VLSI, IOT, Multiplier, Booth, Perfromnace, ALU.**

I. INTRODUCTION

It is possible to use a variety of computer arithmetic systems in order to carry out an advanced multiplier. The majority of these processes involve calculating a large number of intermediate products and then combining the results of the computations involving the incomplete products. Before the 1970s, the vast majority of minicomputers lacked increase guidance [1]. In any case, centralised server computers had duplicate rules, but they executed the same kinds of movements and additions as a "increase schedule." Additionally, early chips lacked any kind of duplicate guidance. At that time, the Motorola 6809, which was released in 1978, was perhaps the first microchip to provide dedicated equipment increase direction [2]. It conducted the same kinds of motions and includes as a "duplicate daily practise," but the execution was done in the microcode of the MUL guide, which is what really important. It became possible to place sufficient adders on a single chip to total all of the fractional products without any delay, as opposed to reusing a single snake to deal with each intermediate product in turn, when more transistors became available per chip as a result of larger scale coordination [4]. This became possible as more transistors became available per chip as a result of larger scale coordination.

Currently, due to the fact that some standard advanced signal processing algorithms spend the majority of their energy duplicating, manufacturers of digital signal processors are forced to sacrifice a great deal of chip area in order to make the increase as quick as possible; a single cycle duplicate gather unit frequently occupied the majority of the chip area of early DSPs[5]. The creation of almost all computerised systems is a challenging and time-consuming endeavour. Restricted creators in any sector will break the initial project into logical subunits constructing squares, and they will use the standard subunits whenever there is a possibility to do so. Adders, registers, and multiplexers are some of the names given to the structural components that may be found in computerised equipment[6].

Various arithmetic operations, including as multiplication, expansion, and subtraction, are important components of an advanced circuit that increase the calculating speed of a processor. In any case, the multiplier unit of the CPU has a significant influence on the speed of the processor. The interest in the design of speedy multipliers in ALU and other sophisticated sign processors is therefore increased as a result of this development. In the course of the most recent few decades, a few novel multiplier engineering developments have been introduced. In the present VLSI structure, the Booth's multiplier [7] and the modified Booth's multiplier are widely recognised; nonetheless, each of these multipliers comes with their own unique set of drawbacks. In this multiplier, before putting up the final result, a few steps that are more or less in the middle of the road are necessary, which slows the pace of the processor down [9]. The performance of the processor slows down exponentially as the number of bits contained in the multiplier and multiplicand increases, therefore these middle of the road phases include a few moving chores, inspection, and subtraction. Because speed is now one of the most important concerns in the design of processors, it is necessary to develop new technologies that are faster than the multiplier that was discussed before. In order to overcome the challenges presented by the conventional multiplier booth's multiplier and the modified multiplier booth's multiplier, an additional engineering method depending on an approximate multiplier is being researched [10].

Computing units that are particularly cautious are not always required for use in applications such as the preparation of interactive media signals and information mining, both of which are tolerant of mistake. They are interchangeable with the partners that are most similar to them. The amount of research being done on approximation computing for applications that can tolerate mistake is growing. In these applications, the fundamental building blocks are composed of adders and multipliers. It has been suggested that entire adders can be implemented at the transistor level, and these adders are used in more complex sign preparation applications. The complete adders that they have suggested are put to use in the aggregate of halfway products in multipliers [11].

In fixed-width multiplier schemes, truncation is often used as a method to reduce the number of equipment features that are multidimensional in nature. When this occurs, a constant or variable rectification term is introduced in order to compensate for the quantization mistake that was caused by the shorter section. In multipliers, estimation techniques centre on the collection of intermediate products, which is important because of how much power is used. The broken array multiplier is run, which shortens the sources of information that are deemed to be of the least importance while simultaneously moulding incomplete products to reduce the multidimensional character of the equipment [12].

II. Literature Survey

Y. -H. Chen et al.,[1] In order to minimise truncation errors, we suggest a data scaling technology (DST) that may be included into low-error fixed-width Booth multipliers (FWBMs). By eliminating unnecessary bits from the multiplicand, the suggested DST improves the performance of low-error FWBMs. By combining the suggested DST with an error-compensation circuit, truncation errors in FWBMs may be mitigated. We determined that the suggested DST-FWBM (1 bit) has a signal-to-noise ratio more than 1.05 dB better than an FWBM without the DST circuit. The precision of long-width DST-FWBMs was very near to the theoretical maximum of a post-truncated multiplier. We used a 0.18-m CMOS process to develop the DST-FWBM and test its performance on a VLSI device. It was shown that the suggested DST approach significantly improved the accuracy of FWBMs, making this technology applicable to digital signal processing methods.

P. Kavipriya et al.[2] Multipliers play a crucial role in digital signal processing and very large scale integration. The Booth multiplier is often used among traditional multipliers.

The goal of this study is to design a Booth multiplier that outperforms conventional multiplies in many respects, including speed, force, and footprint. In this paper, a Booth multiplier is considered that employs a variant of the Square Root Carrier-Select Adder (SQRT CSLA). Using a BEC (Binary to excess one converter) or a CBL in place of an RCA (Ripple Carry Adder) allows for the implementation of modified SQRT CSLA. When developing a Booth multiplier, CBL is preferable since it is more efficient than BEC.

H. Waris et al.[3] Approximate radix-8 Booth multipliers have already been designed with an emphasis on ASIC-based systems. Since these multipliers were designed for ASIC-based systems and are based on an approximation, they do not provide the same performance improvements when applied to FPGA-based hardware accelerators. This is because FPGAs and ASICs are fundamentally designed differently. This brief fills the need by suggesting high-performance approximate radix-8 Booth multipliers made specifically for use in FPGAs. As a result, we offer two approximations to the Booth multiplier in radix 8 (AxBM1 and AxBM2). To make the most of the FPGAs' carry chains and 6-input lookup table (LUT), an approximation scheme is put into place. Comparing AxBM2's latency to that of the previous best FPGA-targeted design (Booth-Approx), a 49% reduction is shown. AxBM2's ability to correct mistakes is a benefit; when paired with truncation, it may result in energy savings of up to 60%. Also, the prior state-of-the-art error-energy Pareto front is refined, allowing for greater energy benefits under a specified error bound. When the suggested multipliers were used in the Sobel edge detection application, AxBM2 recognised 98.45% of edges while conserving energy by 26.41 percent.

Y.-H. Chen et al.[4] offer a data scaling technique (DST) to minimise truncation errors. By eliminating unnecessary bits from the multiplicand, the suggested DST improves the performance of low-error FWBMs. By combining the suggested DST with an error-compensation circuit, truncation errors in FWBMs may be mitigated. We determined that the suggested DST-FWBM (1 bit) has a signal-to-noise ratio more than 1.05 dB better than an FWBM without the DST circuit. The precision of long-width DST-FWBMs was very near to the theoretical maximum of a post-truncated multiplier. We used a 0.18-m CMOS process to develop the DST-FWBM and test its performance on a VLSI device. It was shown that the suggested DST approach significantly improved the accuracy of FWBMs, making this technology applicable to digital signal processing methods.

P. K. Somayajulu et al.,[5]. Dynamic power dissipation hardly rises when technology decreases to 65nm, while leakage power rises dramatically. Therefore, low-power methods are required to mitigate the power drain. This decrease in power consumption is attainable on the system, algorithmic, and architectural levels. The booth multiplier is often used in very low power integrated circuits. Its short processing time, small footprint, and negligible energy use are all pluses. The purpose of this research is to create a pipelined 64-bit Booster booth. The typical procedure for reducing binary radix-16 partial products has a maximum column height of n+1/4, where n is the unsigned operand. In order to get the height down to n/4, the carry save adders are used. Modifications to the multiplier code in the booth are live. Here, the carry skip adder is used instead of the power- and space-saving carry save adder. Reduced energy usage by 11% and footprint by 9%.

H. Waris et al.,[6] Radix-4 By making it simple to create partial products, Booth encoding is often used to realise power-efficient and low-area signed multipliers. But the radix-8 Booth encoding is inefficient since it needs the multiplicand to be generated in odd multiples. This problem is resolved in this concise treatment by rounding down the odd multiples of radix-8 to the closest power of two so that the mistakes cancel each other out. Two approximation Booth multipliers, called hybrid low radix (HLR)-BM1 and HLR-BM2, are developed in an effort to strike a balance between accuracy and energy consumption. HLR-BM2 was able to lower energy consumption by 22% while maintaining an identical MRED to the prior best error-optimized design (ABM1). In addition, compared to the greatest energy-optimized design before (RAD64), HLR-BM2 is 75% more efficient and uses 11% less energy for MRED. Image transformation performance is studied as an example of evaluation. The suggested multiplier designs achieve a peak signal-to-noise ratio (PSNR) of close to 50dB.

Patali, P., et al.,[7] The speed with which a DSP system can execute multiplication operations is crucial to the system's overall performance. Improving many performance indicators at once is challenging yet necessary. This includes delay, power, area, and energy efficiency. An effective carry saving multiplier (CSM) is presented, which uses a vector-merging addition performed by a modified square root carry select adder (MSCA) and a new and improved full adder (IFA) in lieu of the traditional full adder.

The proposed CSM improves upon the improved booth multiplier by 27.74 percent in terms of critical path delay (CPD), 19.4 percent in terms of power, 41.4 percent in terms of area, and 60.87 percent in terms of area delay product (ADP), and upon the low PDP booth multiplier by 46.43 percent, 31.46 percent, 36.9 percent, 63.05 percent, and 65.96 percent. The design and implementation processes are carried out using Cadence software and the gpdk 45nm standard cell library.

Sinha, A., et al.,[8] The use of approximate computing to provide energy-efficient design solutions for error-tolerant applications is an exciting new area of research. Approximations in power-hungry multiplier circuits have been the subject of much study. In this condensed paper, we present two variations of a design for a broken array approximate booth multiplier (SIBAM) that implements partial error correction by discarding sign bits during addition. Compared to precise multipliers with a Mean Relative Error Distance (MRED) constraint of 1.5%, experimental data demonstrate that up to 63% of energy savings may be realised. Energy savings of up to 24% are possible at the 0.3% MRED restriction, according to the research, making the suggested design superior than state-of-the-art multipliers.

S. Venkatachalam et al.[9] are shown to outperform the current approximate Booth multipliers in terms of precision and efficiency. With an MRED of 7:9 x 10-4 compared to the precise Booth multiplier, ABM-M1 may reduce both space and power consumption by up to 23 percent. With an MRED of 2:7 10 -2, ABM-M2 may reduce footprint by 51% and energy consumption by 46%. Area reductions of up to 56% and energy savings of up to 46% are possible with ABM-M3, which has an MRED of 3:4 10 -3. The suggested designs provide more space and power reductions than state-of-the-art current multipliers while retaining good accuracy. We show how well the suggested designs perform with certain common tasks, such transforming images, multiplying matrices, and implementing Finite Impulse Response (FIR) filters.

W. Liu et al. [10] To accommodate varying degrees of precision, approximate RB multipliers make use of both approximate and accurate regular partial product arrays. We also give the findings of our error analysis and hardware simulations. When the word size is big, the suggested approximate RB multipliers perform better than the approximate NB Booth multipliers, as shown by the comparison with earlier approximate Booth multipliers.

To further demonstrate the effectiveness of the suggested solutions, error-tolerant application case studies are provided.

D. J. M. Moss et al. [11]. To reduce latency, our multiplier only combines the nonzero Booth encodings and ignores the zero operations, making it a variation of the serial-parallel (SP) modified radix-4 Booth multiplier. Throughput and latency are enhanced for a subset of multiplier values by using two subcircuits with distinct critical routes. In comparison to conventional parallel-parallel and SP multipliers at four distinct process-voltage-temperature corners, the multiplier is tested on an Intel Cyclone V field-programmable gate array. We demonstrate that, depending on the input set, our optimisations may provide an area-time gain of 1.42 to $3.36 over the typical parallel Booth multiplier for bit widths of 32 and 64.

S. F. Sultana et al. [12] In the past, a complex multiplier or a gain factor other than unity was used to produce FFTs with a unity gain. For floating-point numbers, the IEEE 754 format is utilised. This work presents a reliable float reconstruction approach with improved latency and coverage. The binary multiplier is an algorithmic community consisting of Karatsuba and Urdhva-Tiryagbhyam algorithms. Verilog (HDL) is used to develop the algorithms, and the Xilinx ISE simulator for the FPGA Spartan-3E board is the intended target. The computer resources required by the FFT structure and design are therefore reduced.

III. Challenges

There are some challenges into the booth digital multiplication, which is as followings-

- Implemented only 16 and 32 bit digital booth multiplier while the advance processor uses 64 bit operations.
- Ultra large scale integration is most advanced technology in the fabrication of integrated circuits. More than ten lakh components are integrated together in a single chip. More area size required for implementations by the existing research.
- Due to widespread application of portable electronic devices and the evaluation of microelectronic technology, power dissipation has become a critical parameter in low power VLSI circuit designs. In emerging VLSI technology, the circuit complexity and high speed imply significant increase in the power consumption.
- The network latency is an estimate of the delay of the clock tree before clock tree synthesis. After clock tree synthesis, the total clock latency from the clock source to a clock in of a flip flop is the source latency plus actual delay of the clock tree from the clock definition point to the flip flop. More delay or response time by existing research.

IV. Conclusion

In this work, a review of a low-latency VLSI implementation of a booth multiplier for FPGA-IOT applications is presented. In the earlier study effort, many parameter analyses such as power, area, latency, throughput, frequency, and power delay product were explored in order to discover the optimised designs for high speed applications. The multiplier schemes that have been offered may be used in applications with just a marginal decrease in yield quality, all while saving a significant amount of power and space. Inputs in binary and hexadecimal will be provided for the 64-bit multiplier, and the output of the multiplier will be 128 bits.

REFERENCES

[1] S. Majumder, S. Bhattacharyya, P. Debnath and M. Chanda, "Power Delay Analysis of CMOS Reversible Gates for Low Power Application," 2020 International Conference on Computational Performance Evaluation (ComPE), 2020, pp. 620-625, doi: 10.1109/ComPE49325.2020.9200136.

[2] M. Awais, A. Razzaq, A. Ahmed and G. Masera, "LDPC check node implementation using reversible logic," in IET Circuits, Devices & Systems, vol. 13, no. 4, pp. 443-455, 7 2019.

[3] C. Bandyopadhyay, R. Das, A. Chattopadhyay and H. Rahaman, "Design and synthesis of improved reversible circuits using AIG- and MIG-based graph data structures," in IET Computers & Digital Techniques, vol. 13, no. 1, pp. 38-48, 1 2019.

[4] M. Soeken, M. Roetteler, N. Wiebe and G. D. Micheli, "LUT-Based Hierarchical Reversible Logic Synthesis," in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 38, no. 9, pp. 1675-1688, Sept. 2019.

[5] V. Shukla, O. P. Singh, G. R. Mishra and R. K. Tiwari, "A novel approach for reversible realization of 8-bit adder-subtractor circuit with optimized quantum cost," 2016 International Conference on Emerging Trends in Engineering, Technology and Science (ICETETS), Pudukkottai, 2016, pp. 1-6.

[6] V. Shukla, O. P. Singh, G. R. Mishra and R. K. Tiwari, "Performance parameters optimization and implementation of adder/subtractor circuit using reversible logic approach," 2016 11th International Conference on Industrial and Information Systems (ICIIS), Roorkee, 2016, pp. 323-328.

[7] K. Ghosh, M. M. Haque and S. Chakraborty, "Design of reversible ternary adder/subtractor and encoder/priority encoder circuits," 2016 International Conference on Communication and Signal Processing (ICCSP), Melmaruvathur, 2016, pp. 1290-1295.

[8] M. Sangsefidi, M. Karimpour and M. Sarayloo, "Efficient Design of a Coplanar Adder/Subtractor in Quantum-Dot Cellular Automata," 2015 IEEE European Modelling Symposium (EMS), Madrid, 2015, pp. 456-461.

[9] R. Bardhan, T. Sultana and N. J. Lisa, "An efficient design of adder/subtractor circuit using quantum dot cellular automata," 2015 18th International Conference on Computer and Information Technology (ICCIT), Dhaka, 2015, pp. 495-500.

[10] N. J. Lisa and H. M. H. Babu, "Design of a Compact Ternary Parallel Adder/Subtractor Circuit in Quantum Computing," 2015 IEEE International Symposium on Multiple-Valued Logic, Waterloo, ON, 2015, pp. 36-41.

[11] A. K. Chowdhury, D. Y. W. Tan, S. L. B. Yew, G. L. C. Wyai, B. Madon and A. Thangarajah, "Design of full adder/subtractor using irreversible IG-A gate," 2015 International Conference on Computer, Communications, and Control Technology (I4CT), Kuching, 2015, pp. 103-107.

[12] A. Rahman, Abdullah-Al-Kafi, M. Khalid, A. T. M. S. Islam and M. Rahman, "Optimized hardware architecture for implementing IEEE 754 standard double precision floating point adder/subtractor," 2014 17th International Conference on Computer and Information Technology (ICCIT), Dhaka, 2014, pp. 147-152.