International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435(Online) Volume 12, Issue 08, August 2023)

Low Latency VLSI Implementation of Booth Multiplier for FPGA-IOT Applications

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

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

***Abstract*--The arithmetic logic unit (ALU) is key part of the any processor. The improvement is needed in the ALU for the advanced processor application. Digital multiplier is one of the key operations of the ALU of processor. The Xilinx seven series logic FPGA VLSI processor are using under 5G constraints. Research are continue going on various existing multipliers for enhancing in terms of performance improvement like high speed, low delay, low area, low power etc. This paper proposed low latency VLSI implementation of booth multiplier for FPGA-IOT applications. Simulation is done using Xilinx ISE software. Simulation results shows that the performance improvement in terms of speed and latency.**

***Keywords*-- FPGA-VLSI, 5G, Xilinx ISE, dada, speed, accuracy.**

I. INTRODUCTION

Different computer arithmetic systems can be utilized to execute an advanced multiplier. Out of these most procedures include computing a lot of halfway products, and afterward adding the incomplete products together [1]. Booth's multiplication algorithm is a multiplication algorithm that multiplies two signed binary numbers in two's complement notation. The algorithm was invented by Andrew Donald Booth in 1950 while doing research on crystallography at Birkbeck College in Bloomsbury, London. [1] Booth's algorithm is of interest in the study of computer architecture.

Booth Arithmetic has a great potential to design digital systems that consume less power and area without compromising delay. This technique is mainly utilized to design systems used in error tolerant Digital Signal Processing (DSP) applications as it simplifies the conventional circuit using certain approximation design strategy sacrificing the accuracy of the output [2]. Booth computing is an emerging technique in which power-efficient circuits are designed with reduced complexity in exchange for some loss in accuracy. Such circuits are suitable for applications in which high accuracy is not a strict requirement. Radix-4 modified Booth encoding is a popular multiplication algorithm which reduces the size of the partial product [5].

Booth results are required in many embedded data processors as they reduce time delay and power. As error tolerance adder (ETA) has decreased power drastically trading with accuracy. This work focuses on reducing delay on existing adders when replaced with a fast adder. When compared to the past works on ETA [7]. As one of the most promising energy-efficient emerging paradigms for designing digital systems, booth computing has attracted a significant attention in recent years. Applications utilizing booth computing can tolerate some loss of quality in the computed results for attaining high performance. Booth arithmetic circuits have been extensively studied; however, their application at system level has not been extensively pursued. Furthermore, when booth arithmetic circuits are applied at system level, error-accumulation effects and a convergence problem may occur in computation. Semi-supervised learning can improve accuracy and performance by using unlabeled examples [8].

II. METHODOLOGY

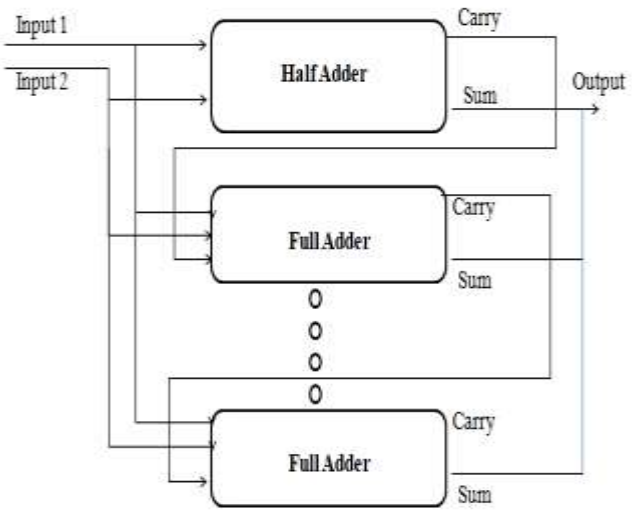

Figure 1: Flow Chart

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435(Online) Volume 12, Issue 08, August 2023)

The booth in multipliers leads to realisation of faster computations with reduced hardware complexity, delay and power, with accuracy in desirable levels. Partial product summation is the speed limiting operation in multiplication due to the propagation delay in adder net works. In order to reduce the propagation delay, compressors are introduced. Compressors compute the sum and carry at each level simultaneously. The resultant carry is added with a higher significant sum bit in the next stage. This is continued until the final product is generated.

Dada multiplier is a famous multiplication conspire the cluster, the summation continues in a more standard, yet more slow way, to getting the summation of the fractional items .Utilizing this plan just one column of bits in the lattice is disposed of at each phase of the summation. In a parallel multiplier the halfway items are created by utilizing exhibit of AND entryways. The fundamental issue is the summation of the fractional items, and it is the time taken to perform this summation which decides the greatest speed at which a multiplier may work. The Dadda plot basically limits the quantity of adder stages required to perform the summation of halfway items. This is accomplished by utilizing full and half adders to diminish the quantity of lines in the grid number of bits at every summation arrange. Dadda multipliers are a refinement of the parallel multipliers exhibited by Wallace. Dadda multiplier comprises of three phases. A full adder is a usage of a (3, 2) counter which takes 3 inputs and creates 2 outputs. Likewise a half adder is an execution of a (2, 2) counter which takes 2 inputs and delivers 2 outputs.

III. Simulation Results

The proposed booth multiplier is implemented and simulated by using the Xilinx ISE 14.7 software, The Isim simulator is used to check the results validity in test bench.

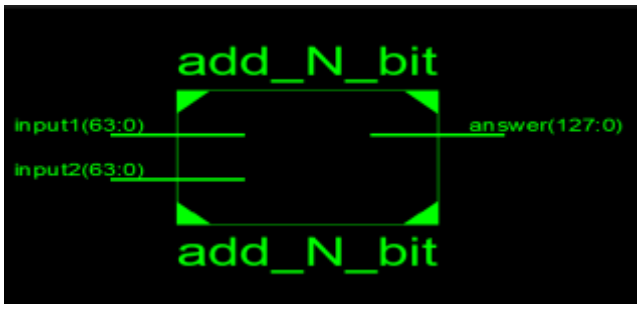

Figure 2: Top view of booth multiplier

Figure 2 is showing the top view of the proposed code, which includes the 64 bits input 1 and input 2and 128 bit output.

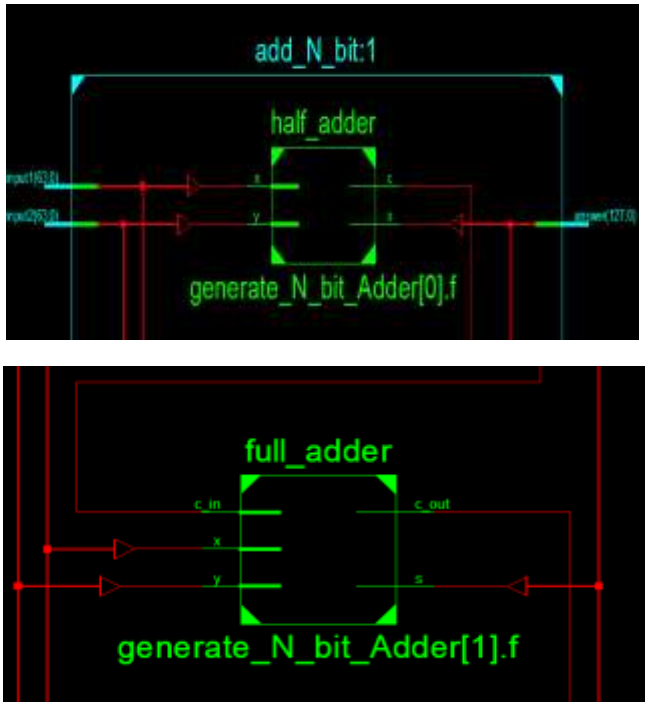

Figure 3: RTL view of half and full adder

Half adder- The 64 bit input x and y goes to the half adder and generate sum (s) and carry (c).

Full adder- The 64 bit input x and y goes to the full adder and carry (c) from the half adder. All the sums are added and generate final product.

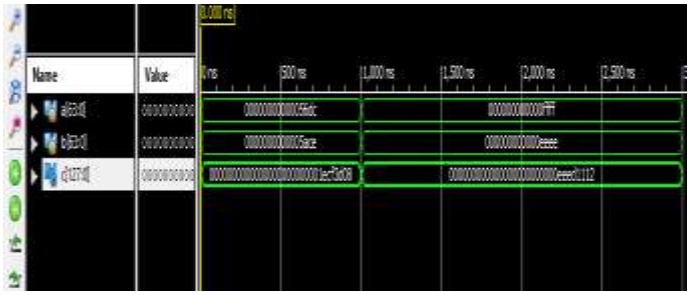

Figure 4: test bench results in hexadecimal

Figure 4 is showing the test bench results. Here the 64 bit input 1 and input 2 data bits is applied and generate the 128 bit output after multiplication.

Input 1 (a) = 56dc, Input 2 (b) = 5ace, Output (c) = 1ECF3D08

Input 1 (a) = ffff, Input 2 (b) = eeee, Output (c) = EEED1112

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435(Online) Volume 12, Issue 08, August 2023)

Figure 5: test bench results in octal

Figure 5 is showing the test bench results in octal. Here the 64 bit input 1 and input 2 data bits is applied and generate the 128 bit output after multiplication.

Input 1 (a) = 01123456712345671234567

Input 2 (b) = 07765432176543217654321

Output (c) = 012170536007176257732047313140141171335104 7

Table 1:
Result Comparison

Sr No.	Parameters	Previous work [1]	Proposed work
1	Area	320	287
2	Delay	4.92 ns	3.105 ns
3	Power	9.13 mW	8.2mW
4	PDP (Power delay product)	44.91	25.46

IV. CONCLUSION

Booth multiplier are one of the fastest multiplier for the AI based FPGA-VLSI applications. The proposed research is presents the 64 X 64 bit booth multiplier. The virtex 7 family FPGA IC is used to simulate the results. The proposed booth multiplier id designed for the 64 X 64 bit multiplication while previous it is designed for the 16 X 16 bit multiplication. The virtex 5 & 7 family FPGA IC is used to simulate the results. The total number of component or utilized area is 287 while previously it is 320. The total delay value is 0.344 ns in proposed and 4.92 ns in the previous. The utilized power consumption by the previous is 9.13 mW and 8.2mW in the proposed. The power delay product is 44.91 by the previous and 25.46 by the proposed work. Therefore the proposed booth multiplier provides the significant better results than the existing..

REFERENCES

[1] Y. -H. Chen, "Improvement of Accuracy of Fixed-Width Booth Multipliers Using Data Scaling Technology," in IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 68, no. 3, pp. 1018-1022, March 2021, doi: 10.1109/TCSII.2020.3023429.

[2] P. Kavipriya, S. Lakshmi, T. Vino, M. R. Ebenezar Jebarani and G. Jegan, "Booth Multiplier Design Using Modified Square Root Carry-Select-Adder," 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS), 2021, pp. 1647-1653, doi: 10.1109/ICAIS50930.2021.9396032.

[3] H. Waris, C. Wang, W. Liu and F. Lombardi, "AxBMs: Booth Radix-8 Booth Multipliers for High-Performance FPGA-Based Accelerators," in IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 68, no. 5, pp. 1566-1570, May 2021, doi: 10.1109/TCSII.2021.3065333.

[4] Y. -H. Chen, "Improvement of Accuracy of Fixed-Width Booth Multipliers Using Data Scaling Technology," in IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 68, no. 3, pp. 1018-1022, March 2021, doi: 10.1109/TCSII.2020.3023429.

[5] P. K. Somayajulu and S. R. Ramesh, "Area and Power Efficient 64-Bit Booth Multiplier," 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS), 2020, pp. 721-724, doi: 10.1109/ICACCS48705.2020.9074305.

[6] H. Waris, C. Wang and W. Liu, "Hybrid Low Radix Encoding-Based Booth Booth Multipliers," in IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 67, no. 12, pp. 3367-3371, Dec. 2020, doi: 10.1109/TCSII.2020.2975094.

[7] P. Patali and S. T. Kassim, "An Efficient Architecture for Signed Carry Save Multiplication," in IEEE Letters of the Computer Society, vol. 3, no. 1, pp. 9-12, 1 Jan.-June 2020, doi: 10.1109/LOCS.2020.2971443.

[8] A. Sinha Roy and A. S. Dhar, "SIBAM—Sign Inclusive Broken Array Multiplier Design for Error Tolerant Applications," in IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 67, no. 11, pp. 2702-2706, Nov. 2020, doi: 10.1109/TCSII.2020.2969903.

[9] S. Venkatachalam, E. Adams, H. J. Lee and S. Ko, "Design and Analysis of Area and Power Efficient Booth Booth Multipliers," in IEEE Transactions on Computers, vol. 68, no. 11, pp. 1697-1703, 1 Nov. 2019, doi: 10.1109/TC.2019.2926275.

[10] W. Liu et al., "Design and Analysis of Booth Redundant Binary Multipliers," in IEEE Transactions on Computers, vol. 68, no. 6, pp. 804-819, 1 June 2019, doi: 10.1109/TC.2018.2890222.

[11] D. J. M. Moss, D. Boland and P. H. W. Leong, "A Two-Speed, Radix-4, Serial–Parallel Multiplier," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 27, no. 4, pp. 769-777, April 2019, doi: 10.1109/TVLSI.2018.2883645.

[12] S. F. Sultana and B. Patil, "Area efficient VLSI architecture for reversible radix_2 FFT algorithm," 2021 International Conference on Emerging Smart Computing and Informatics (ESCI), 2021, pp. 136-141, doi: 10.1109/ESCI50559.2021.9396952.

[13] P. J. Edavoor, S. Raveendran and A. D. Rahulkar, "Booth Multiplier Design Using Novel Dual-Stage 4:2 Compressors," in IEEE Access, vol. 8, pp. 48337-48351, 2020, doi: 10.1109/ACCESS.2020.2978773.

[14] V. A and R. Dhavse, "Design of High Accuracy, Power Efficient and Area Efficient 16x16 Booth Multiplier," 2020 IEEE 17th India Council International Conference (INDICON), 2020, pp. 1-6, doi: 10.1109/INDICON49873.2020.9342223.