Praveen Kumar Sonu, 2021, 9:6
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

Design of 64-Bit Approximate Multiplier Algorithm Based On Compressors and Dadda Multipliers for VLSI-AI Application

Research Scholar Praveen Kumar Sonu, Associate Prof. Dr. Anshuj Jain
Department of Electronics and Communication Engineering,
Scope College of Engineering,
Bhopal, India.

Abstract- Various arithmetic operations such as multiplication, addition, and subtraction are important parts of a digital circuit to speed up the computation speed of the processor. This paper presents a 64bit approximate multiplier for high speed and low delay for advanced digital signal processing. Previously, it was designed for 16 and 32bit for various applications. The work focuses on hardware-level approximation by presenting the partial product perforation approach and the dada multiplier for creating approximate multiplication circuits. Xilinx ISE14.7 is used in the implementation.

Keywords- Approximate Digital, Multiplier, FPGA, DSP, Xilinx VLSI.

I. INTRODUCTION

An advanced multiplier can be implemented using a variety of computer arithmetic systems. The majority of these processes entail computing a large number of midway products and then combining the unfinished products. Different arithmetic operations, including as addition, subtraction, multiplication, and division, are important components of sophisticated circuits that help the processor calculate faster.

Careful computing units are not always required in applications that can tolerate inaccuracy, such as interactive media signal preparation and data mining. They can be replaced with their close relatives. Approximate computing for error-tolerant applications is becoming more popular. The main segments in these applications are adders and multipliers. At the transistor level, approximation complete adders are presented, and they are used in advanced sign preparation applications.

In fixed-width multiplier schemes, truncation is commonly used to reduce the equipment complex character of multipliers. A constant or variable rectification term is then added to compensate for the quantization error introduced by the shorter section. In multipliers, estimation processes focus on accumulating intermediate products, which is critical for power utilization.

The broken array multiplier is used to shorten the least crucial pieces of information sources while moulding incomplete products to reduce the multidimensional nature of the equipment.

The following characteristics should be present in a good multiplier:

- **Accuracy:** A decent multiplier should produce the desired result.
- **Speed:** Multiplier ought to perform activity at fast.
- **Area:** A multiplier should possess less number of cuts and LUTs.
- **Power:** Multiplier should devour less power.

Approximate computing has emerged as a potential solution for the design of energy-efficient digital systems [1] Multimedia, recognition, and data mining applications are inherently error-tolerant and do not necessitate absolute precision in processing.

Approximate circuits may play an essential part in these applications as a viable alternative for decreasing area, power, and latency in digital systems that can accept some loss of accuracy, resulting in improved energy efficiency. Adders have been extensively explored for approximate

implementation as one of the fundamental components in arithmetic circuits.

II. METHODOLOGY

Arithmetic units such as adders and multipliers are crucial components in a logic circuit and are used in multimedia, recognition, and other applications.

The speed and power consumption of arithmetic circuits have a substantial impact on a processor's performance. Carry look ahead adders (CLAs) and approximation multipliers are popular high-performance arithmetic circuits..

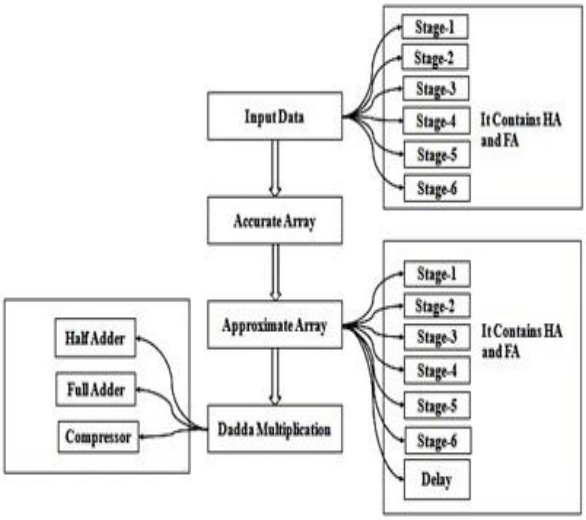

Fig 1. Flow Chart.

Figure 1 is showing proposed flow chart. According to this working flow it can be clear that proposed approximate multiplier is design and implemented according to following sub modules-

- Carry Save Adder
- Dadda Multiplier
- Compressor
- Full Adder
- Half Adder

1. Dadda Multiplier:

In a well-known multiplication scheme known as the cluster, the summation is carried out in a more traditional, but slower, manner to obtain the sum of the fractional components. At each phase of the summing, only one column of bits in the lattice is discarded using this strategy. The midway elements in a parallel multiplier are formed by displaying AND entryways.

The basic issue is the summation of the fractional components, and the time it takes to do so determines the maximum speed at which a multiplier can operate.

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. The incomplete item grid is formed in the primary stage by N2 AND stages. In the subsequent stage, the halfway item lattice is decreased to a tallness of two. Dadda supplanted Wallace Pseudo adders with parallel (n,m) counters.

A Parallel (n, m) counter is a circuit which has n inputs and produce m outputs which give a double tally of the ONEs present at the inputs. 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 implementation of the proposed algorithm is done over Xilinx ISE 14.7. The ISE package processing toolbox helps us to use the functions available in Xilinx Library.

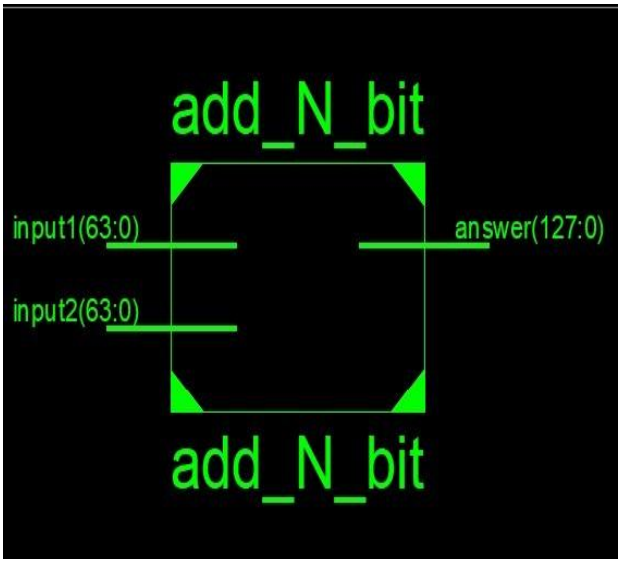

Fig 2. Top level View.

In figure 2, showing top level view of proposed 64-bit approximate multiplier. There is 64 bit at input 'a' and 64 bit at input 'b'. The output 'c' of this multiplier will be 128 bit. In digital multiplier the output bit is total addition of input bit.

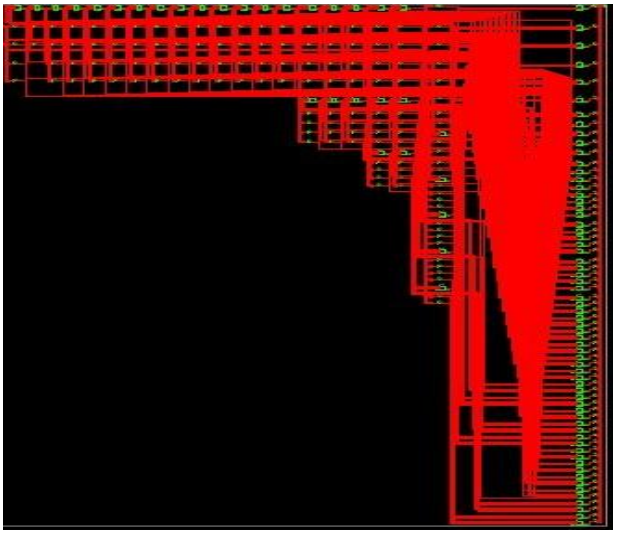

Fig 3. Completer register transfer level technology view.

Figure 3 is showing technological register transfer level view. There are 6 stages of operation so many numbers of wires, full adder and other component are using.

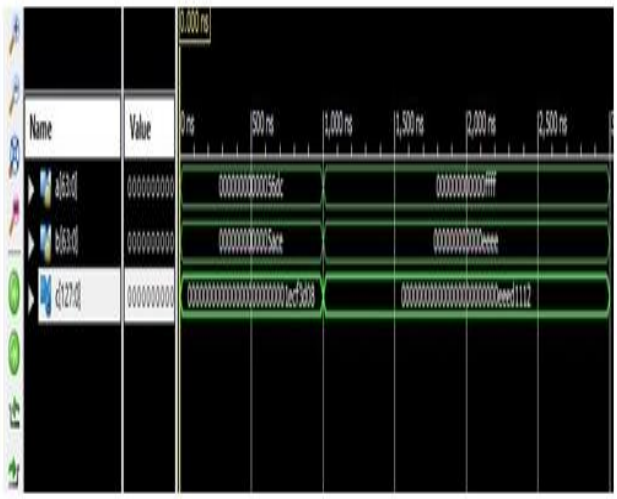

Fig 4. Test bench results in hexadecimal.

The test bench findings are shown in Figure 4. After multiplication, the 64 bit input 1 and input 2 data bits are used to form the 128 bit output.

Input 1 (a) = 56dc, Input 2 (b) = 5ace, Output (c) = 1ECF3D08 Input 1 (a) = ffff, Input 2 (b) = eeee, Output (c) = EEED1112

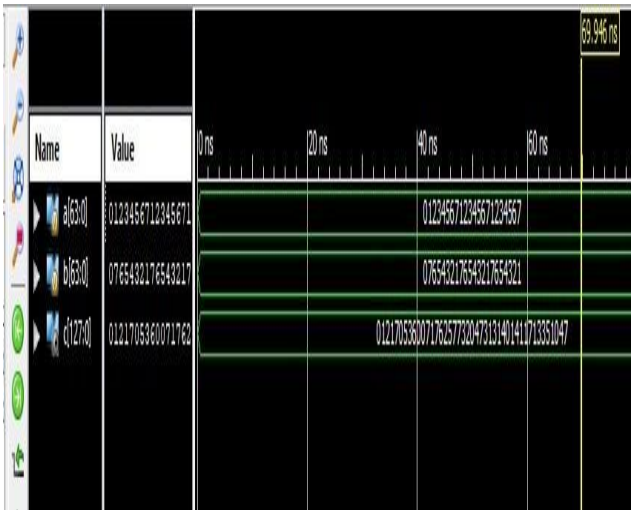

Fig 5. Test bench results in octal.

Figure 5 depicts the octal test bench findings. After multiplication, the 64 bit input 1 and input 2 data bits are used to form the 128 bit output.

Input 1 (a) = 01123456712345671234567
Input 2 (b) = 07765432176543217654321
Output(c)=012170536007176257732047313140141171335104 7

Table 1. Comparison of Simulation Results.

Sr No.	Parameters	Previous work [1]	Proposed work	Improvement
1	Order of Approximate Multiplier	16 X 16	64 X 64	4 times
2	Area (mm2)	18.17	10.16	40%
3	Delay (ns)	24.48	12.68	46%
4	Power (microwatt)	2.51	1.79	27%
5	PDP (Power Delay product)	62.99	25.01	60%

IV. CONCLUSION

This Multiplier is a crucial component in arithmetic processors; today's mobile and DSP applications require ICs that operate at fast speeds while consuming little power. This study examines a variety of characteristics such as power, area, latency, throughput, frequency, and power delay product in order to identify superior architectures for high-speed applications. The suggested multiplier is for a 64 bit X 64 bit approximates multiplier, whereas the previous one was for a 16 bit X 16 bit approximate

multiplier. The simulation was run using Xilinx 14.7 software, and the results reveal that the proposed multiplier saves 40% of the area, 46% of the latency, and 27% of the power.

REFERENCES

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

[2] F. Sabetzadeh, M. H. Moaiyeri and M. Ahmadinejad, "A Majority-Based Imprecise Multiplier for Ultra-Efficient Approximate Image Multiplication," in IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 66, no. 11, pp. 4200- 4208, Nov. 2019.

[3] H. Saadat, H. Bokhari and S. Parameswaran, "Minimally Biased Multipliers for Approximate Integer and Floating-Point Multiplication," in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 37, no. 11, pp. 2623- 2635, Nov. 2018.

[4] V. Mrazek, Z. Vasicek, L. Sekanina, H. Jiang and J. Han, "Scalable Construction of Approximate Multipliers With Formally Guaranteed Worst Case Error," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 26, no. 11, pp. 2572-2576, Nov. 2018.

[5] S. Venkatachalam and S. Ko, "Design of Power and Area Efficient Approximate Multipliers," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 25, no. 5, pp. 1782-1786, May 2017.

[6] S. Mazahir, O. Hasan, R. Hafiz and M. Shafique, "Probabilistic Error Analysis of Approximate Recursive Multipliers," in IEEE Transactions on Computers, vol. 66, no. 11, pp. 1982-1990, 1 Nov. 2017.

[7] W. Liu, L. Qian, C. Wang, H. Jiang, J. Han and F. Lombardi, "Design of Approximate Radix-4 Booth Multipliers for Error-Tolerant Computing," in IEEE Transactions on Computers, vol. 66, no. 8, pp. 1435-1441, 1 Aug. 2017.

[8] A. Mokhtari, W. Shi, Q. Ling and A. Ribeiro, "DQM: Decentralized Quadratically Approximated Alternating Direction Method of Multipliers," in IEEE Transactions on Signal Processing, vol. 64, no. 19, pp. 5158-5173, 1 Oct.1, 2016.

[9] H. Jiang, J. Han, F. Qiao and F. Lombardi, "Approximate Radix-8 Booth Multipliers for Low-Power and High-Performance Operation," in IEEE Transactions on Computers, vol. 65, no. 8, pp. 2638-2644, 1 Aug. 2016.

[10] B. Shao and P. Li, "Array-Based Approximate Arithmetic Computing: A General Model and Applications to Multiplier and Squarer Design," in IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 62, no. 4, pp. 1081-1090, April 2015.

[11] Z. Zhi-Qing Lü and X. An, "Non-conforming finite element tearing and interconnecting method with one Lagrange multiplier for solving large-scale electromagnetic problems," in IET Microwaves, Antennas & Propagation, vol. 8, no. 10, pp. 730-735, 15 July 2014.

[12] D. De Caro, N. Petra, A. G. M. Strollo, F. Tessitore and E. Napoli, "Fixed-Width Multipliers and Multipliers-Accumulators With Min-Max Approximation Error," in IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 60, no. 9, pp. 2375-2388, Sept. 2013.

[13] J. Wang, S. Kuang and S. Liang, "High-Accuracy Fixed-Width Modified Booth Multipliers for Lossy Applications," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 19, no. 1, pp. 52-60, Jan. 2011.

[14] F. Auger, Z. Lou, B. Feuvrie and F. Li, "Multiplier-Free Divide, Square Root, and Log Algorithms [DSP Tips and Tricks]," in IEEE Signal Processing Magazine, vol. 28, no. 4, pp. 122-126, July 2011.