Implementation of 64-Bit Approximate Multiplier for Accurate and High-Level Processing

Ragini Singh [1*] and Prof. Sandip Nemade [2]
[1] M. Tech. in Microelectronics and VLSI Design from RGPV Bhopal (M.P.) India.
[2] Asst. Prof. in Electronic and Communication Department in Technocrats Institute of Technology Bhopal (M.P.) India.
*Corresponding author email id: raginigarima@gmail.com

Abstract – **Approximate computing has received significant attention as a promising strategy to enhance performance of multiplication. Various arithmetic operations such as multiplication addition, subtraction are important part of digital circuit to speed up the computation speed of processor. This paper presents 64 bit approximate multiplier for high speed and low delay for advance digital signal processing. Previous it is designed for 16 bit and 32 bit for various applications. Research work is focus on hardware-level approximation by introducing the partial product perforation technique and dadda multiplier for designing approximate multiplication circuits. Xilinx 14.7 is used to implementation with verilog programming language.**

Keywords – **Approximate, Dadda, Multiplier, Verilog, Speed.**

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. Different arithmetic activities, for example, addition, subtraction, multiplication and division are significant piece of advanced circuit to accelerate the calculation speed of processor.

In applications like interactive media signal preparing and information mining which can endure error, careful computing units are not constantly essential. They can be supplanted with their approximate partners. Research on approximate computing for error tolerant applications is on the ascent. Adders and multipliers structure the key segments in these applications. In approximate full adders are proposed at transistor level and they are used in advanced sign preparing applications.

To decrease equipment multifaceted nature of multipliers, truncation is generally utilized in fixed-width multiplier plans. At that point a steady or variable rectification term is added to make up for the quantization error presented by the shortened part. Estimation procedures in multipliers center on gathering of halfway products, which is vital regarding power utilization. Broken array multiplier is executed, where the least critical bits of sources of info are shortened, while shaping incomplete products to diminish equipment multifaceted nature.

An effective multiplier ought to have following attributes : -

Accuracy:

A good multiplier should give right outcome.

Speed:

Multiplier ought to perform activity at fast.

Area:

A multiplier should possesses less number of cuts and LUTs.

Power:

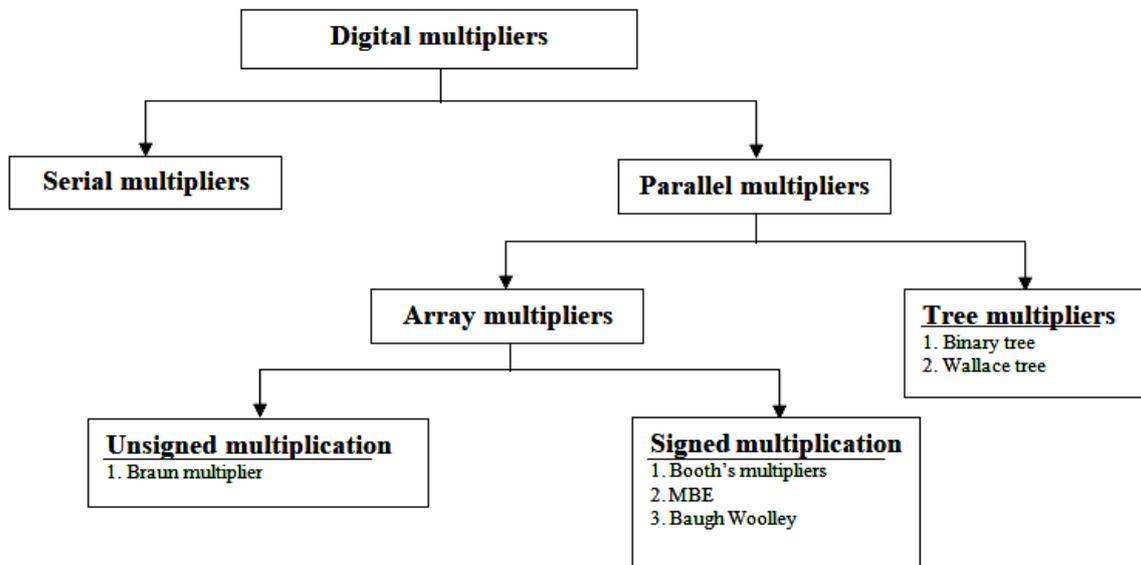Multiplier should devour less power.

Fig. 1. Types of Multiplier.

Approximate computing has emerged as a potential solution for the design of energy-efficient digital systems [1]. Applications such as multimedia, recognition and data mining are inherently error-tolerant and do not require a perfect accuracy in computation. For these applications, approximate circuits may play an important role as a promising alternative for reducing area, power and delay in digital systems that can tolerate some loss of accuracy, thereby achieving better performance in energy efficiency. As one of the key components in arithmetic circuits, adders have been extensively studied for approximate implementation.

II. BACKGROUND

There are many multipliers designed by researchers. The basic multiplication algorithm is discussed here. The multiplication algorithm for an N bit multiplicand by N bit multiplier is shown below:

$$Y = Y_{n-1}\, Y_{n-2} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots Y_2\, Y_1\, Y_0 \quad \text{Multiplicand}$$

$$X = X_{n-1}\, X_{n-2} \ldots\ldots\ldots\ldots\ldots\ldots\ldots X_2\, X_1\, X_0 \quad \text{Multiplier}$$

Also, gates are utilized to create the Fractional Items, PP, On the off chance that the multiplicand is N-bits and the Multiplier is M-bits, at that point there is N* M halfway item. How the fractional items are produced or summed up is the distinction between the diverse models of different multipliers.

Duplication of paired numbers can be disintegrated into increments. Consider the increase of two 8-bit numbers A and B to create the 16 bit item P.

The equation for the addition is:

$$P(m + n) = A(m)B(n) = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} a_i b_j 2^{i+j}$$

- If the LSB of Multiplier is '1', at that point include the multiplicand into a collector.

- Shift the multiplier one piece to the right and multiplicand one piece to one side.

- Stop when all bits of the multiplier are zero.

Equipment usage of digital signal preparing (DSP) algorithms and mixed media applications in advances, for example, field programmable gate arrays (FPGAs) and digital signal processors requires countless. Frequently, the general execution of the plan is restricted by imperatives on the speed, vitality utilization, and zone necessities of the accessible multiplier structure choices. This is especially valid for applications revolved around current handheld sight and sound gadgets, where physical size, chip area and power. Therefore, research has been centered on the improvement of effective, propelled multiplier procedures to help these requesting applications.

III. PROPOSED METHODOLOGY

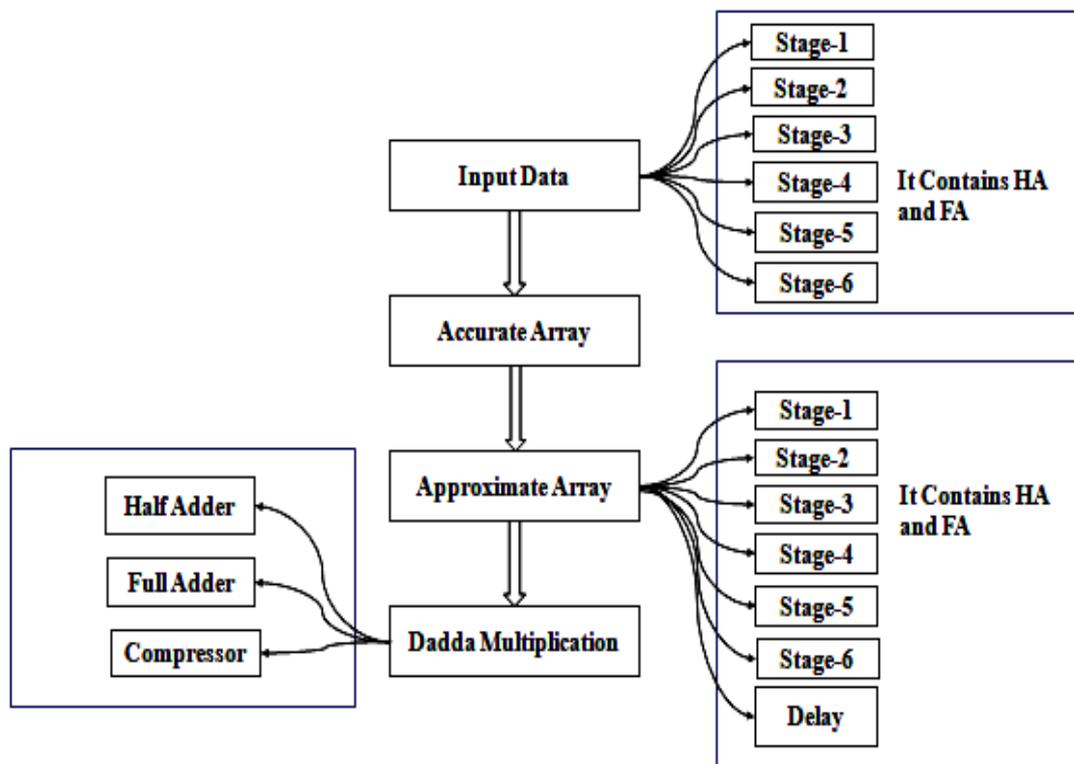Proposed 64-bit approximate multiplier is design according to following flow chart.

Fig. 2. Flow Chart.

Figure 2 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

A. *Dadda Multiplier*

In 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. 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.
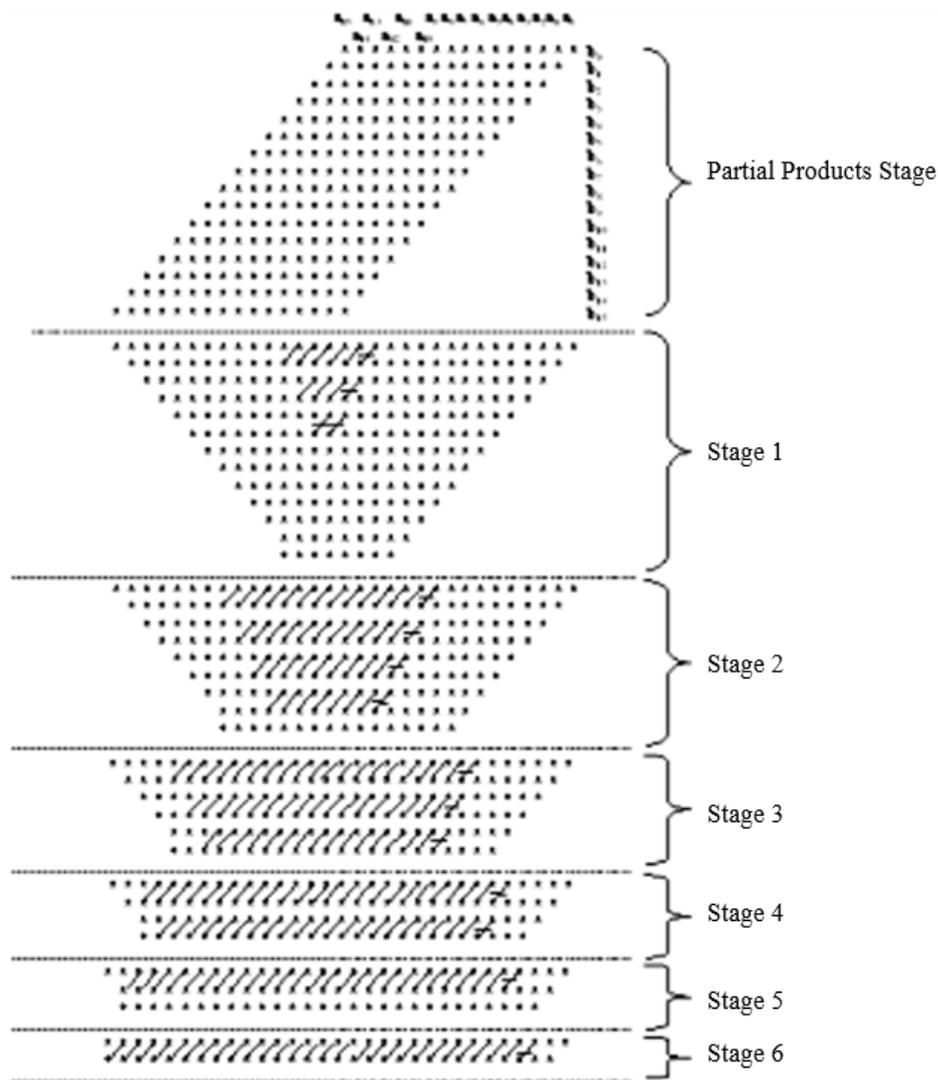

Fig. 3. Dot diagram of proposed 64-bit dadda multiplier.

Figure 3 indicating spot chart of proposed 64-bit dadda multiplier [1]. Generation of incomplete items, halfway items decrease tree, and at last, a vector consolidate expansion to deliver last item from the entirety and convey columns created from the decrease tree. Second step expends more force. In this short, estimate is applied in decrease tree arrange.

Fig. 4. Top level View.

In figure 4, 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.

IV. RESULT AND DISCUSSION

The implementation of the proposed 64-bit approximate multiplier is Xilinx 14.7 version using verilog language. Isim simulator is used for simulation and validation of result in test bench. Behavioral modeling style used to develop proposed algorithm. Artix Family is used to implementation.

Binary Number Input-1 [64-bit]

Input (a) = 1010111111000011010101010111110000111100001111000011111111111110

Input (b) = 1101011111111100000011111000001111111000000111111000000111111100

Fig. 5. Result validation in Test Bench-1.

Now simulation is done using Isim simulator. Figure 5 is showing test bench window. Here 'a' and 'b' is 64 bit inputs and 'c' is 128 bit output. Value of 'a' and 'b' is mentioned above. Now the output of 'c' is multiply of 'a' and 'b'.

Output (c) = 0111000101100100001001010001001010000110111011101010100010100011001111001000000 0000000000000000000000000000000000000000000000000000000000

Hexadecimal Number Input-2

Fig. 6. Result validation in Test Bench-2.

Here 'a' and 'b' is 64 bit hexadecimal inputs and 'c' is 128 bit hexadecimal output. Value of 'a' is aaaa and value of 'b' is bbbb. The output of 'c' is multiply of 'a' and 'b'. So value of 'c' is 7D26D82E.

Table 1. Simulation Parameter.

Sr. No.	Parameters	Proposed Work
1	Type of Multiplier	64-bit Approximate-Multiplier
2	Area	30.33%
3	Delay	13.745ns
4	Accuracy rate	97%
5	Simulation Time	19.00 Secs
6	Power	0.082mW
7	PDP (Power delay product)	1127
8	Frequency	72.7 MHz
9	Throughput	4.6 GHz
10	Memory	4625700 kilobytes
11	Global Maximum Fanout	100000

Fig. 7. Comparison of multiplier bit.

Proposed multiplier is designed for 64 bit approximate multiplication while in previous work it is designed only for 16-bit.

So it can be say that proposed 64-bit multiplier is advanced of previous multiplier in terms of calculated parameters.

V. Conclusion

Multiplier is an important part in arithmetic processors, the current mobile applications and DSP applications need ICs with high-speed operations but low power consumption. This paper presents 64-bit efficient approximate multiplier. It is implemented and result validation using xilinx software successfully. Various parameters calculated like power, area, latency, throughput, frequency and power delay product to identify the improved architectures for high speed applications. The proposed multiplier plans can be utilized in applications with negligible misfortune in yield quality while sparing huge power and area. The 64-bit inputs binary and hexadecimal will be given and 64-bit multiplier output will be resultant. Proposed multiplier save 70% area and consume 0.08mW power.

References

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

[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.

[15] I. Park and T. Kim, "Multiplier-less and table-less linear approximation for square and square-root," *2009 IEEE International Conference on Computer Design*, Lake Tahoe, CA, 2009, pp. 378-383.

Author's Profile

First Author
Ragini Singh is Electronics and Communication Engineer, currently she is perusing M. Tech. in Microelectronics and VLSI Design from RGPV Bhopal (M.P.) India.

Second Author
Prof. Sandip Nemade holds degree in VLSI Design and currently working as Asst. Prof. in Electronic and Communication Department in Technocrats Institute of Technology Bhopal (M.P.) India.