JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue

JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)

An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Design of 64-Bit Digital Approximate Multiplier for High Speed AI-VLSI Applications

[1]Sanjay Sharma, [2]Prof. Swati Gupta
[1]Research Scholar, [2]Assistant Professor
Department of Electronics and Communication Engineering
Vidhyapeeth Institute of Science & Technology, Bhopal, 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 Design of 64-Bit Digital Approximate Multiplier for High Speed AI-VLSI Applications. Simulation is done using Xilinx ISE software. Simulation results shows that the performance improvement in terms of speed and latency.

***IndexTerms* – FPGA-VLSI, Xilinx ISE, dada, Speed, Accuracy.**

I. INTRODUCTION

High speed multimedia applications have paved way for a whole new area in high speed error-tolerant circuits with approximate computing. These applications deliver high performance at the cost of reduction in accuracy. Furthermore, such implementations reduce the complexity of the system architecture, delay and power consumption. This paper explores and proposes the design and analysis of two approximate compressors with reduced area, delay and power with comparable accuracy when compared with the existing architectures [1].

Approximate 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].

Approximate 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]. Approximate 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, approximate computing has attracted a significant attention in recent years. Applications utilizing approximate computing can tolerate some loss of quality in the computed results for attaining high performance. Approximate arithmetic circuits have been extensively studied; however, their application at system level has not been extensively pursued. Furthermore, when approximate 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].

The fundamental attributes of the engineering are depicted as follows:

1) Control Unit: The Control Unit is a small 5 - arrange pipeline that brings and deciphers directions, and controls the information stream, arithmetic unit and the I/O ports DSP incorporates four universally useful registers inside its Control unit.

2) Custom Instruction Set: The arrangement of directions actualized has been intended to amplify the utilization of the Arithmetic Unit, improving the power decreases attainable by the multiplier. Directions are 32 bits wide and enable access to one or both memories obstructs in a single guidance. The guidance set is part into various functional areas, as explained below:

3) Memory blocks: Memory on the PTMAC engineering is framed by a 1024, 32 bit Program Memory square and two 512, 16 bit Information Memory squares. Each of the three recollections can be gotten to on a single clock cycle, as they are legitimately associated with the control unit.

4) Arithmetic Unit: The Arithmetic Unit is the center unit of this DSP. It comprises of an increase and structure with fundamental units as:

II. METHODOLOGY

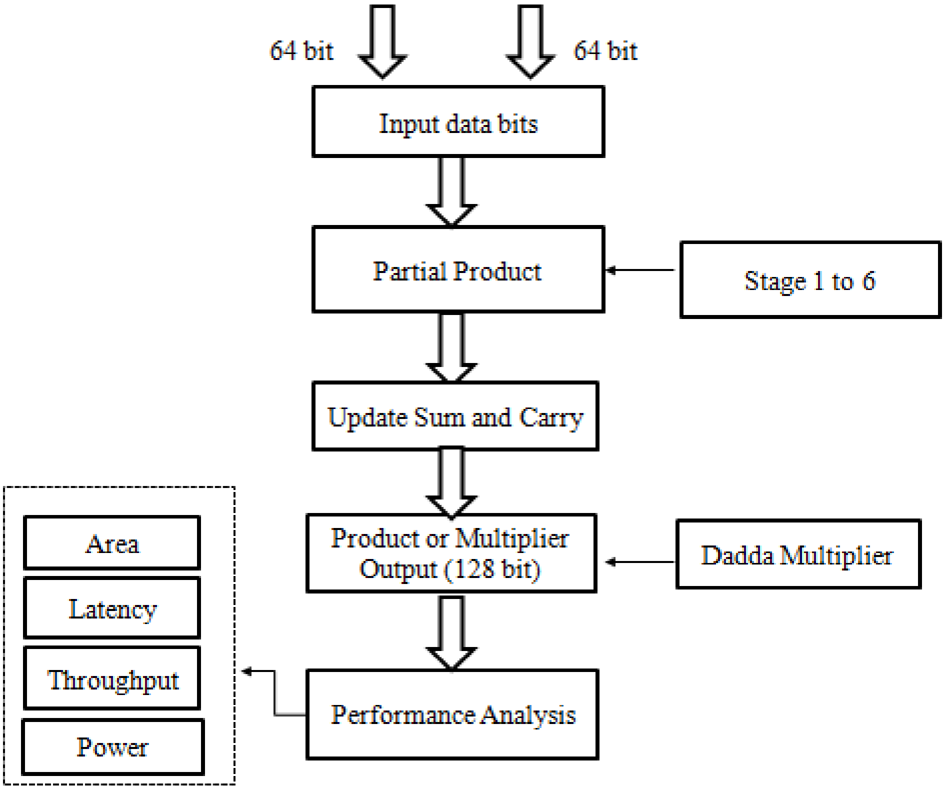

Figure 1: Flow Chart

The approximation 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. 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 proposed approximate 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.

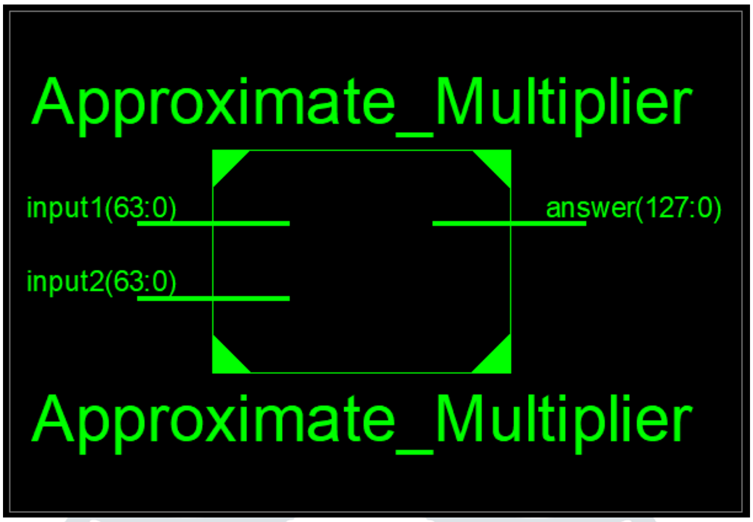

Figure 2: Top view of approximate multiplier

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

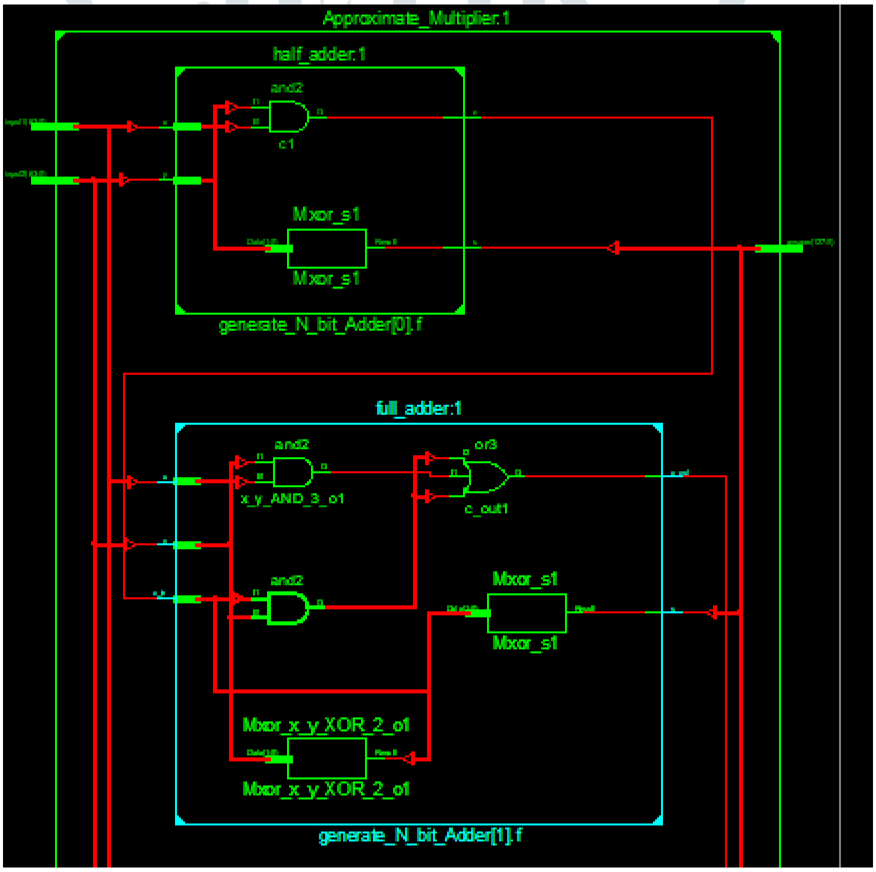

Figure 3: Combination of half and full adder

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.

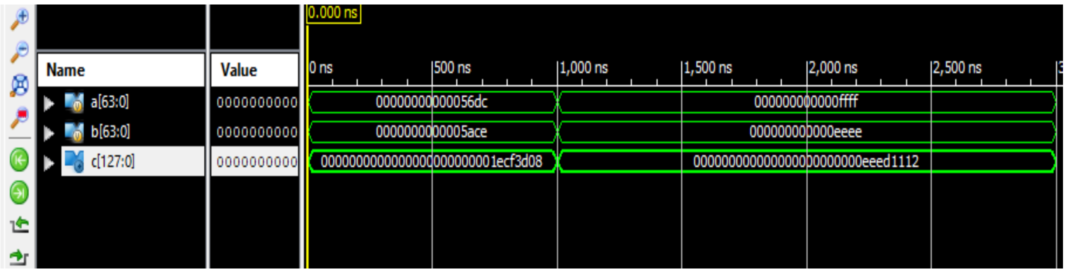

Figure 4: test bench results in hexadecimal

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

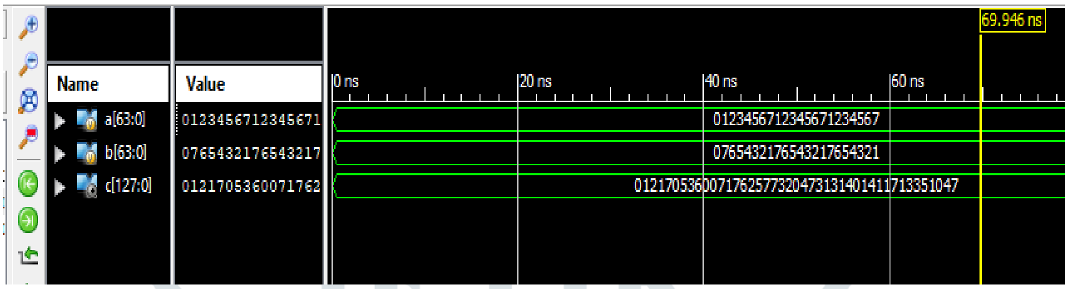

Figure 5: test bench results in octal

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) = 0121705360071762577320473131401411713351047

Table 1: Comparison of Simulation Results

Sr No.	Parameter	Previous Work [1]	Proposed Work
1	Order of multiplier	16 X 16	64 X 64
2	Area	18.37 mm^2	10.66 mm^2
3	Delay	0.624 ns	0.344 ns (16X16) 1.376 ns (64X64)
4	Power	218.8 micro watt	128 microwatt
5	Power Delay Product	136.62	44.032
6	Throughput	25.64 Gbps	46.51 Gbps

IV. CONCLUSION

Approximate multiplier are one of the fastest multiplier for the AI based FPGA-VLSI applications. The proposed research is presents the 64 X 64 bit approximate multiplier. The virtex 7 family FPGA IC is used to simulate the results. The proposed approximate multiplier id designed for the 64 X 64 bit multiplication while previous it is designed for the 16 X 16 bit multiplication. The total number of component or utilized area is 10.66 mm2 while previously it is 18.37 mm2. The total delay value is 0.344 ns (16X16) and 1.376 ns (64X64) for proposed work while 0.624 ns are for previous work. The throughput achieved by this research is 46.51 Gbps while 25.64 Gbps for previous simulation results.

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. V. A and R. Dhavse, "Design of High Accuracy, Power Efficient and Area Efficient 16x16 Approximate Multiplier," 2020 IEEE 17th India Council International Conference (INDICON), 2020, pp. 1-6, doi: 10.1109/INDICON49873.2020.9342223.
3. R. Bhattacharjya, A. Kanani and N. Goel, "ReARM: A Reconfigurable Approximate Rounding-Based Multiplier for Image Processing," 2020 24th International Symposium on VLSI Design and Test (VDAT), 2020, pp. 1-4, doi: 10.1109/VDAT50263.2020.9190474.

4. H. Waris, C. Wang and W. Liu, "Hybrid Low Radix Encoding-Based Approximate 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.
5. 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, doi: 10.1109/TC.2019.2926275.
6. S. T. Bala, D. Shangavi and P. Sangeetha, "Area and Power Efficient Approximate Wallace Tree Multiplier using 4:2 Compressors," 2018 International Conference on Intelligent Computing and Communication for Smart World (I2C2SW), 2018, pp. 287-290, doi: 10.1109/I2C2SW45816.2018.8997160.
7. V. V. Kavipranesh, J. Janarthanan, T. N. Amruth, T. M. Harisuriya and E. Prabhu, "Power And Delay Efficient Exact Adder For Approximate Multiplier," 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2018, pp. 1896-1899, doi: 10.1109/ICACCI.2018.8554552.
8. P. Huang, C. Wang, R. Ma, W. Liu and F. Lombardi, "A Hardware/Software Co-design Method for Approximate Semi-Supervised K-Means Clustering," 2018 IEEE Computer Society Annual Symposium on VLSI (ISVLSI), 2018, pp. 575-580, doi: 10.1109/ISVLSI.2018.00110.
9. T. Su, C. Yu, A. Yasin and M. Ciesielski, "Formal Verification of Truncated Multipliers Using Algebraic Approach and Re-Synthesis," 2017 IEEE Computer Society Annual Symposium on VLSI (ISVLSI), 2017, pp. 415-420, doi: 10.1109/ISVLSI.2017.79.
10. A. Bonetti, A. Teman, P. Flatresse and A. Burg, "Multipliers-Driven Perturbation of Coefficients for Low-Power Operation in Reconfigurable FIR Filters," in IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 64, no. 9, pp. 2388-2400, Sept. 2017, doi: 10.1109/TCSI.2017.2698138.
11. R. Zendegani, M. Kamal, M. Bahadori, A. Afzali-Kusha and M. Pedram, "RoBA Multiplier: A Rounding-Based Approximate Multiplier for High-Speed yet Energy-Efficient Digital Signal Processing," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 25, no. 2, pp. 393-401, Feb. 2017, doi: 10.1109/TVLSI.2016.2587696.
12. P. Yin, C. Wang, W. Liu and F. Lombardi, "Design and Performance Evaluation of Approximate Floating-Point Multipliers," 2016 IEEE Computer Society Annual Symposium on VLSI (ISVLSI), 2016, pp. 296-301, doi: 10.1109/ISVLSI.2016.15.
13. A. Mehta, S. Maurya, N. Sharief, B. M. Pranay, S. Jandhyala and S. Purini, "Accuracy-configurable approximate multiplier with error detection and correction," TENCON 2015 - 2015 IEEE Region 10 Conference, 2015, pp. 1-4, doi: 10.1109/TENCON.2015.7372902.
14. V. Mrazek and Z. Vasicek, "Automatic Design of Low-Power VLSI Circuits: Accurate and Approximate Multipliers," 2015 IEEE 13th International Conference on Embedded and Ubiquitous Computing, 2015, pp. 106-113, doi: 10.1109/EUC.2015.20.