



DESIGN AND IMPLEMENTATION OF A FAST SEQUENTIAL MULTIPLIER BASED ON ITERATIVE ADDITION ARCHITECTURE

Amit Gupta, Ravishankar Singh, Madan Gopal, Rishikesh Tiwari

Assistant Professor, Student, Student, Student
Electronics And Communication Engineering

Galgotia College Of Engineering And Technology, Greater Noida, India

Abstract : Multiplication is a fundamental operation in digital signal processing, cryptography, and various computational tasks. The efficiency of multiplication directly impacts the overall performance of these applications. Traditional multiplication techniques such as array and Wallace tree multipliers often face challenges in terms of area overhead and speed. In recent years, iterative-based sequential multipliers have emerged as a promising alternative, offering improved performance and reduced resource utilization. This paper presents a comprehensive overview of iterative-based sequential multipliers, focusing on their design principles, architecture, and performance characteristics. Unlike conventional multipliers, which require complex hardware structures, iterative-based sequential multipliers employ a sequential approach, breaking down the multiplication process into simpler stages. This enables efficient resource utilization and facilitates high-speed operation. The proposed multiplier architecture comprises interconnected stages, each performing a specific computation step. By iteratively processing partial products, the multiplier achieves the desired multiplication result. Moreover, the sequential nature of the design enables pipelining, further enhancing throughput and reducing latency. Various optimization techniques, such as operand recoding and partial product reduction, are employed to minimize critical path delay and improve overall performance. Experimental results demonstrate the effectiveness of the iterative-based sequential multiplier in terms of speed, area efficiency, and power consumption. Comparative analysis with conventional multipliers showcases significant improvements in performance metrics, making it an attractive choice for applications demanding high-speed arithmetic operations within constrained hardware resources. In conclusion, iterative-based sequential multipliers offer a compelling solution for enhancing the efficiency and performance of multiplication operations in digital systems. Their simplicity, scalability, and adaptability make them suitable for a wide range of applications, from embedded systems to high-performance computing platforms. Further research in optimization techniques and architectural enhancements holds the potential to unlock even greater advancements in iterative-based sequential multiplication.

I. INTRODUCTION

The Iterative Addition Architecture is a computational framework utilized in the design of sequential multipliers, a crucial component in digital arithmetic. Unlike parallel multipliers that compute multiple partial products simultaneously, sequential multipliers generate partial products iteratively, employing a series of additions to accumulate the final product. This architecture operates on the principle of breaking down multiplication into a sequence of addition operations, making it well-suited for applications where speed, area efficiency, and power consumption are key considerations. At its core, the Iterative Addition Architecture consists of a single adder circuit that iteratively accumulates partial products over multiple clock cycles.

Each iteration of the addition process involves shifting one of the multiplicands and conditionally adding the other multiplicand based on the shifted bits. By repeating this process for each bit of the multiplier, the architecture effectively computes the product of the two input numbers. The simplicity and sequential nature of the Iterative Addition Architecture offer several advantages. It requires minimal hardware resources compared to parallel architectures, making it area-efficient and suitable for implementation in resource-constrained environments. Additionally, its sequential operation allows for straightforward pipelining, enabling high-speed performance without compromising on accuracy.

II. MOTIVATION

The motivation behind developing an iterative-based sequential multiplier arises from the need to overcome the limitations of traditional multiplication techniques, such as array and Wallace tree multipliers. These conventional methods, while effective for general-purpose arithmetic, often suffer from drawbacks like high area overhead, long critical paths, and limited scalability. As technology advances and applications demand higher computational performance within constrained resources, alternative

multiplication approaches are sought to offer improved efficiency and speed. Iterative-based sequential multipliers address these challenges through several key motivations:

Firstly, they aim for greater area efficiency by adopting a sequential approach and breaking down the multiplication process into simpler stages. This strategy allows them to achieve comparable or better performance with reduced hardware resources, making them particularly advantageous for applications with strict area constraints.

Secondly, these multipliers optimize speed by minimizing critical path delay through distributed computation across multiple stages and enabling pipelining. This enhances overall system performance, especially in high-speed computing applications. Scalability is another crucial factor driving the development of iterative-based sequential multipliers. Their inherent scalability allows them to easily support larger operand sizes without significantly impacting performance or resource utilization, making them suitable for a wide range of applications. Furthermore, power efficiency is enhanced through optimization of the hardware architecture and reduction of unnecessary computation. This is vital for prolonging battery life and reducing overall energy consumption, particularly in portable and IoT devices. Lastly, the adaptability and flexibility of iterative-based sequential multipliers make them suitable for various application requirements. Techniques such as operand recoding and partial product reduction enable optimization for different operand sizes, data formats, and performance targets, making them versatile across digital signal processing, cryptography, and scientific computing tasks.

III. CONTRIBUTION

The sequential multiplier represents a significant advancement in digital arithmetic, offering a range of contributions that address key challenges in multiplication operations. Firstly, it significantly enhances efficiency by streamlining the multiplication process into sequential stages, thereby reducing hardware complexity and area overhead. This makes it particularly well-suited for implementation in resource-constrained environments like embedded systems and mobile devices. Secondly, its utilization of pipelining techniques enables high-speed operation by allowing parallel processing of multiple operands, resulting in improved throughput and reduced latency. This boosts overall system performance, especially in applications requiring rapid arithmetic computations. Thirdly, the sequential multiplier exhibits inherent scalability, accommodating a wide range of operand sizes without necessitating substantial architectural modifications. This adaptability is crucial for meeting diverse application requirements and adapting to future technological advancements. Moreover, it optimizes power consumption by minimizing unnecessary computation and employing sequential processing techniques, making it suitable for power-sensitive applications such as battery-powered devices and IoT sensors. Lastly, its flexibility in design and optimization enables tailoring to specific application needs through techniques such as operand recoding, partial product reduction, and architectural enhancements. Overall, the sequential multiplier's contributions encompass efficiency, speed, scalability, power efficiency, and flexibility, making it a valuable asset in digital arithmetic operations across various domains.

IV. RELATED WORK

Research in the design and implementation of fast sequential multipliers based on iterative addition architecture encompasses various avenues for optimization and enhancement. Firstly, it involves an in-depth exploration of different iterative addition architectures proposed in the literature, each offering unique strategies to break down multiplication into sequential additions efficiently. Additionally, studies delve into algorithmic enhancements such as Booth's algorithm, Wallace tree multiplication, or Dadda multiplier, integrating them into iterative addition architectures to improve performance. Techniques involving parallelism and pipelining are also investigated to accelerate sequential multipliers, reducing latency by executing multiplication stages concurrently. Moreover, researchers focus on optimizing resource utilization and area efficiency by exploring different data path architectures, operand representations, and encoding schemes. Tailored optimizations for specific hardware technologies like FPGA or ASIC implementations are considered, along with strategies for error resilience and fault tolerance. Comparative performance evaluation and benchmarking studies play a crucial role in assessing the effectiveness of proposed approaches, while application-specific considerations ensure the adaptability of sequential multiplier designs to meet diverse performance requirements and constraints. By building upon existing research in these areas, novel methodologies for designing fast sequential multipliers can be developed, catering to a wide range of application domains.

Research Study no.	Model	Finding	Limitation
1.	8-by-8 Bit Shift/Add Multiplier	Evaluated different design variants to compare speed and area efficiency	The study didn't explore the optimization of constraints for efficient synthesis in-depth.
2.	A Review of Different Type of Multipliers and Multiplier-Accumulator	Compare	performance can vary based on application requirements
3.	DESIGN A POCKET MULTI-BIT MULTIPLIER IN FPGA	This multiplier can be used as an interface for microcomputer	The study focuses on various machine learning (ML) and deep learning (DL) approaches related to brain diseases
4.	RoBA Multiplier	achieved through rounding inputs to the nearest power of 2 (2^n)	introduces a small error in exchange for speed and energy efficiency

5.	DESIGN AND PERFORMANCE OF HIGH SPEED MULTIPLIER USING VERILOG HDL	multiplier architectures can be extended to implement larger multipliers, such as 32-bit and 64-bit multipliers	the performance of these multipliers in more complex and specific application scenarios is not discussed
6.	A Reliable Low Power Multiplier Using Fixed Width Scalable Approximation	including a 50% increase in speed, a 70% reduction in area, and a 60% decrease in energy consumption compared to traditional multipliers	it does so by reducing operand lengths and approximating values

TABLE I. MULTIPLIER WORK

V. Method

The digital circuits are one of the most essential parts in our everyday life and affect every aspect of it Hardware Description Language, is a language used in order to ease the designing process of the digital circuit as it deals with the digital circuit as if it is a programming language like c/c++ or python. Adder and Multipliers are one of the essential parts in any ALU circuit in any device as any instruction being carried out in any system is consists of simple math instructions of basic arithmetic operations. The objective of this is to implement an 8 by 8 sequential multiplier to get an output of 16 bit using a simple shift register circuits and Full Adder circuit.

The design for this project is based on the, firstly, behavioral structure as first this project starts by building the basic blocks of the circuit of the sequential multiplier.

First the project starts with implementing each block as shown in the previous figure. Then determining the signals that is going to connect the blocks with each other's lastly, map the connections and implement external processes like the “ANDing” process before the Full Adder block. The last part of the design is the system controller designed using Finite State Machine which is given in next figure.

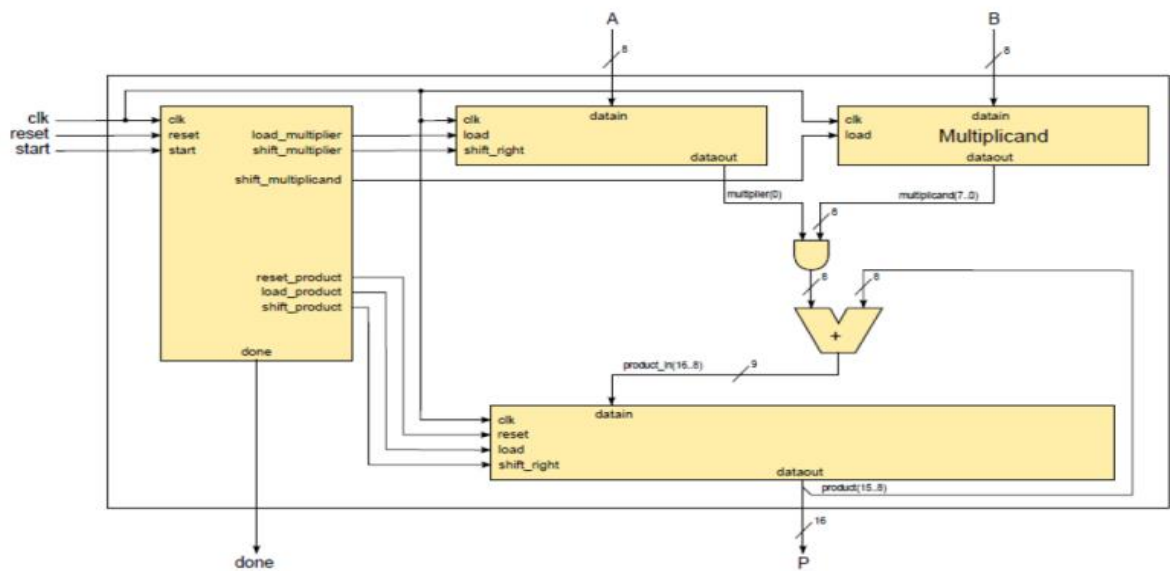


Figure 1 building block of sequential multiplier.

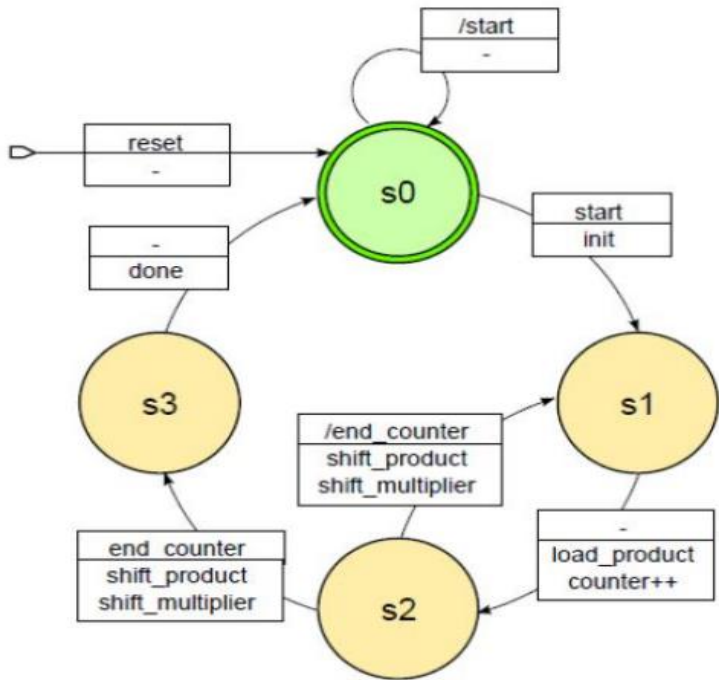


Figure 2 system controller FSM

The design was implemented as a finite state machine with states and transition logic as shown in Figure 2. The Start signal transitions the state machine out of the idle state and into the initialize state whereby it commands the multiplicand and multiplier to be loaded into registers. Once loaded, the state machine goes through a series of test and shift, or test, add and shift operations depending on the status of the LSB bit. Upon reaching the maximum count for the multiplication cycle, the state machine goes back to the idle state and outputs a Stop signal.

Controller is the control unit of the multiplier. It receives a START signal and consequently commands all other modules until the result is obtained and it outputs a STOP signal.

5.2 Simulation and Timing

The controller is synchronous to the clock and transitions through the various states occur on the rising clock edge. As can be seen from the timing diagram in Figure 3-3, the Start signal transitions the state machine out of the idle state only when sampled by the rising clock edge. Upon entering the initialize state, the LOAD_cmd is generated. During each test state, the LSB is sampled. If the LSB was high, the add state is entered and the controller generates the ADD_cmd. If the LSB was low, or once the add state is exited, the shift state is entered and the controller generates the SHIFT_cmd. Upon reaching the maximum count for the multiplication cycle, the state machine goes back to the idle state and outputs a stop signal.

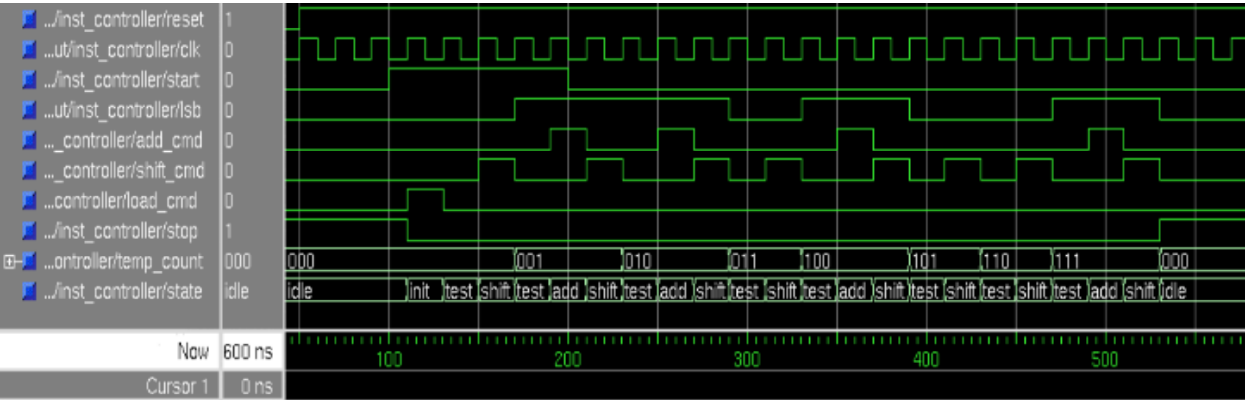
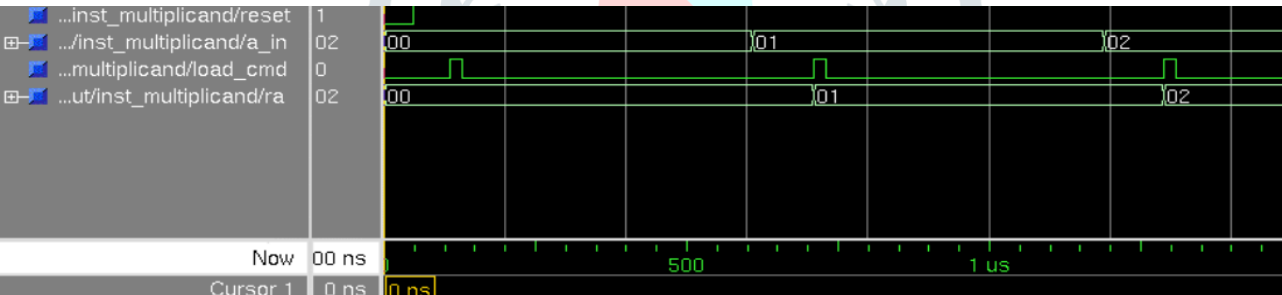
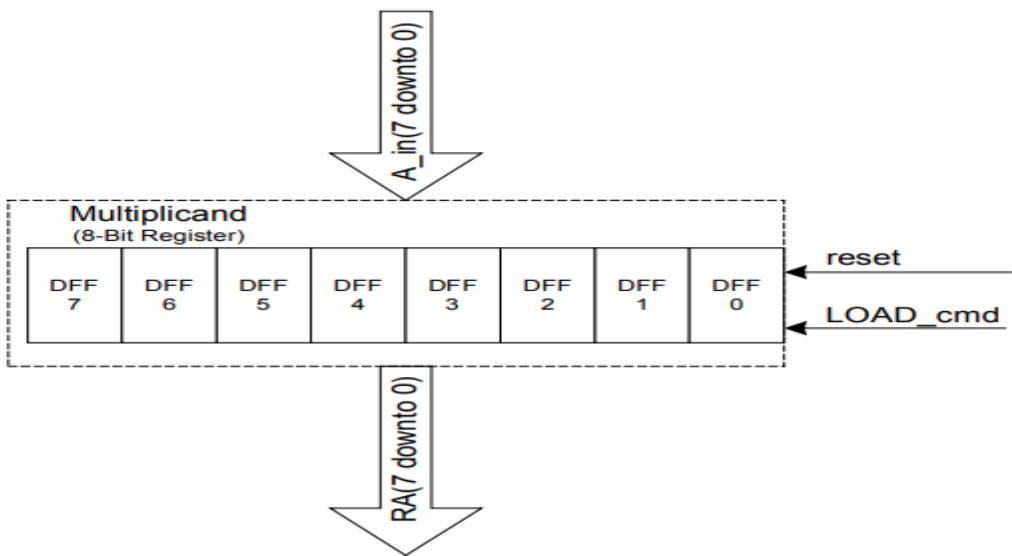


Figure 3 Controller Simulation Timing Diagram

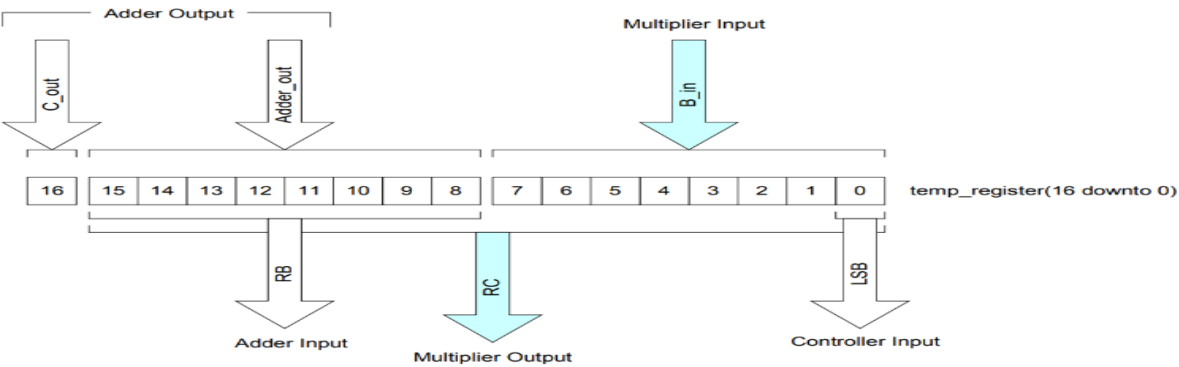
5.3 Multiplicand

The basic design for the Multiplicand block is that of an 8-bit register. The top-level multiplicand module generates an 8-bit register The byte is loaded into the register only when the LOAD_cmd is received from the Controller and the register is cleared when a global reset is applied



5.4 Multiplier

The Multiplier/Result block stores the multiplier (iB1 byte) as well as the accumulated output of the adder. It allows the register to be logically shifted right and provides one of the Adder’s inputs. The Multiplier/Result block consists of a 17-bit shift register and a multiplexer in order to provide this functionality.



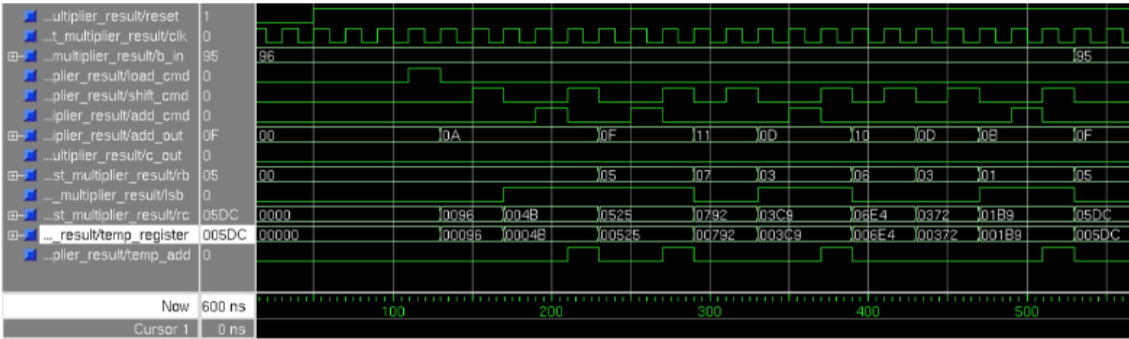


Figure 7 Multiplier Simulation Timing Diagram

5.5 Adder
The adder design is the most influential part of the multiplier in terms of the area and speed achievable. This is due to the large iterations of additions performed in the multiplication cycle.

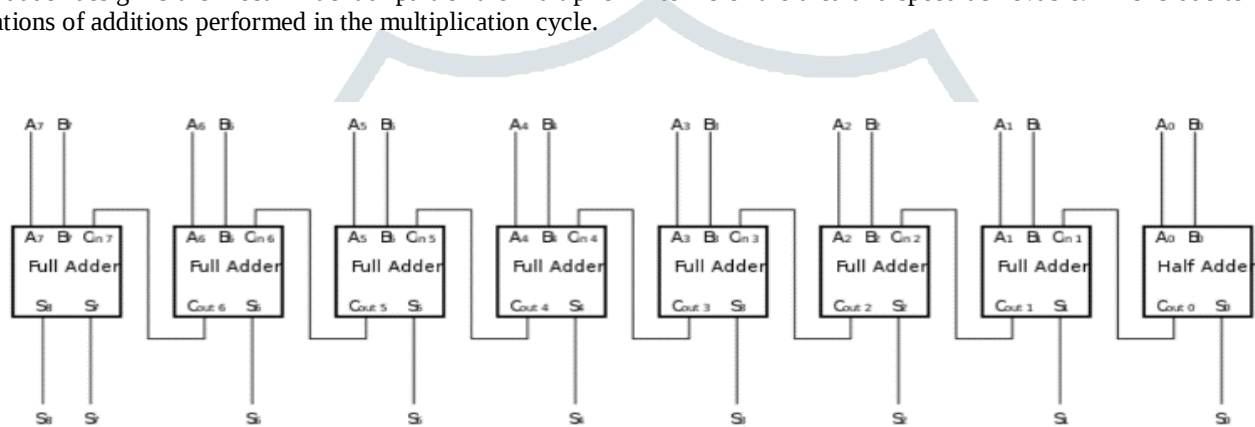


Figure 8 Adder

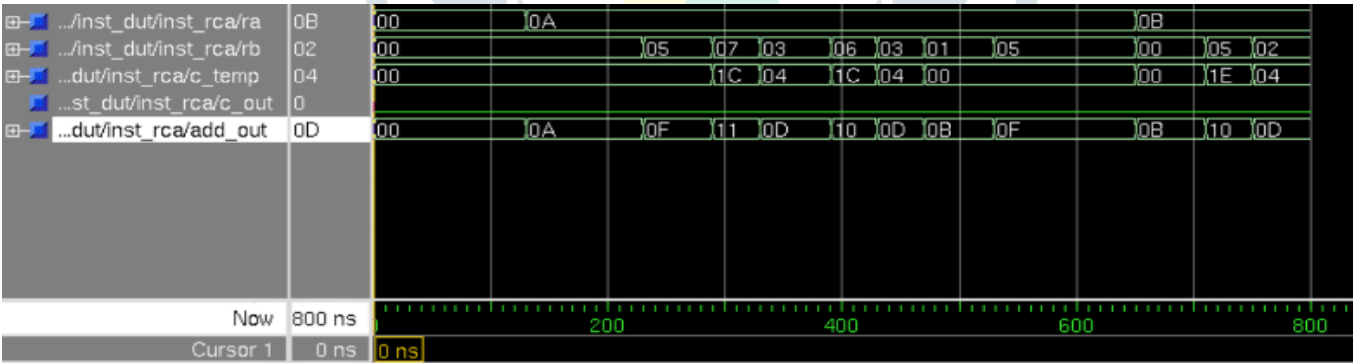


Figure 9 Adder Simulation Timing Diagram

5.6 Complete Multiplier
Simulation Result Files contains the multiplicand and multiplier sent to the Multiplier as well as the result received

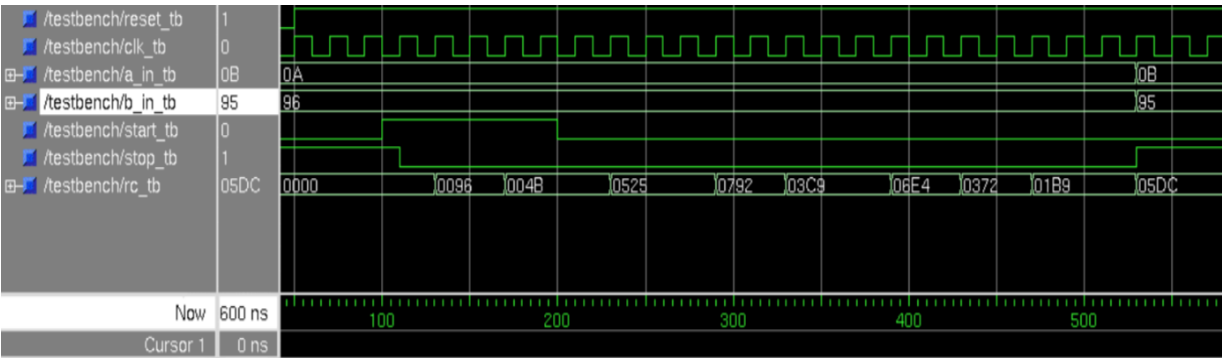


Figure 10 Complete Multiplier Simulation Timing Diagram

VI. RESULTS AND DISCUSSION

The designs of 8-bit size multipliers (including the design of both proposed and conventional sequential multiplier) are synthesized with Verilog implementation of Xilinx software. Following features were utilized in this sequential multiplier.

Structural design implementations of two components for conventional and proposed sequential multipliers were built using Verilog HDL synthesis. A top-level component is built to instantiate all the lower level components of each multiplier. The designs were implemented and synthesized based on the structures explained previously. **Our main motive is to minimize the error in RoBA multiplier it is 14.% which is reduced by our proposed sequential multiplier(iterative addition). Our worst time delay is 7.959ns and total power consumed is 32.97mW which is better than the other combinational multiplier(Array Multiplier).**

Based on the simulation results, our proposed design shows better performance than the RoBA multiplier in terms of error. Results of simulation show that the proposed design has a slightly larger number of logic elements and the delay time, and power consumption are significantly high. Since the standard design of adder (i.e. Ripple Carry Adder RCA) is used in implementing the accumulator unit of the proposed multiplier, the area of the proposed multiplier is slightly bigger than the area of RoBA multiplier.

VII. CONCLUSION

In this paper, a new approach is proposed for implementing a fast sequential multiplier. The new approach comes up with a new implementation for sequential multiplier. Conventional and proposed designs of sequential multiplier are simulated in Xilinx software tool using Verilog implementations. Results of simulation demonstrate that the proposed design of the sequential multiplier is faster than the conventional design. The proposed sequential multiplier is found to have a **reduction in error of RoBA which is 14.%**. As a future work reducing the total logic elements of the implementation could be achieved by applying more enhancements to the design of adder that occupied most of the area of the proposed multiplier.

REFERENCES

- [1] Jhamb M, Garima and Lohani H 2016 Design, implementation and performance comparison of multiplier topologies in power-delay space Engineering Science and Technology, an International Journal 19 355–63
- [2] Sabeetha S, Ajayan J, Shriram S, Vivek K and Rajesh V 2015 A study of performance comparison of digital multipliers using 22nm strained silicon technology 2015 2nd International Conference on Electronics and Communication Systems (ICECS)
- [3] Salman E H 2013 Fast convolution and correlation algorithm for multidimensional signals Diyala Journal of Engineering Sciences 6 75–89
- [4] Mogheer H S 2018 A new technology for reducing power consumption in synchronous digital design using tri-state buffer Diyala Journal of Engineering Sciences 11 60–6
- [5] Qian L, Wang C, Liu W, Lombardi F and Han J 2016 Design and evaluation of an approximate Wallace-Booth multiplier 2016 IEEE International Symposium on Circuits and Systems (ISCAS) 1974–7
- [6] Prabhu A S and Elakya V 2012 Design of modified low power booth multiplier 2012 International Conference on Computing, Communication and Applications
- [7] Kuang S-R, Wang J-P and Guo C-Y 2009 Modified Booth Multipliers With a Regular Partial Product Array IEEE Transactions on Circuits and Systems II: Express Briefs 56 404–8
- [8] Govekar D and Amonkar A 2017 Design and implementation of high speed modified booth multiplier using hybrid adder 2017 International Conference on Computing Methodologies and Communication (ICCMC)
- [9] Sukowati A I, Putra H D and Wibowo E P 2016 Usage area and speed performance analysis of booth multiplier on its FPGA implementation 2016 International Conference on Informatics and Computing (ICIC)
- [10] Patterson D A and Hennessy J L 2018 Chapter 3 Arithmetic for Computers Computer organization and design the hardware/software interface RISC-V (Cambridge, MA, MA: Elsevier, Morgan Kaufmann Publishers) pp 177–81
- [11] Navabi Z 2005 Sequential Multiplier Digital Design and Implementation with Field Programmable Devices ed Z Navabi (Boston, MA: Springer US) pp 229–45
- [12] J. N. Mitchell, "Computer multiplication and division using binary logarithms," IRE Trans. Electron. Comput., vol. EC-11, no. 4, pp. 512–517, Aug. 1962.
- [13] V. Mahalingam and N. Ranganathan, "Improving accuracy in Mitchell's logarithmic multiplication using operand decomposition," IEEE Trans. Comput., vol. 55, no. 12, pp. 1523–1535, Dec. 2006.
- [14] Nangate 45nm Open Cell Library, accessed on 2010. [Online]. Available: <http://www.nangate.com/>
- [15] H. R. Myler and A. R. Weeks, The Pocket Handbook of Image Processing Algorithms in C. Englewood Cliffs, NJ, USA: Prentice-Hall, 2009.
- [16] S. Narayanamoorthy, H. A. Moghaddam, Z. Liu, T. Park, and N. S. Kim, "Energy-efficient approximate multiplication for digital signal processing and classification applications," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 23, no. 6, pp. 1180–1184, Jun. 2015.

- [17] S. Hashemi, R. I. Bahar, and S. Reda, "DRUM: A dynamic range unbiased multiplier for approximate applications," in Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD), Austin, TX, USA, 2015, pp. 418–425.
- [18] C.-H. Lin and I.-C. Lin, "High accuracy approximate multiplier with error correction," in Proc. 31st Int. Conf. Comput. Design (ICCD), 2013, pp. 33–38.
- [19] A. B. Kahng and S. Kang, "Accuracy-configurable adder for approximate arithmetic designs," in Proc. 49th Design Autom. Conf. (DAC), Jun. 2012, pp. 820–825.
- [20] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," IEEE Trans. Image Process., vol. 13, no. 4, pp. 600–612, Apr. 2004.
- [21] J. Liang, J. Han, and F. Lombardi, "New metrics for the reliability of approximate and probabilistic adders," IEEE Trans. Comput., vol. 62, no. 9, pp. 1760–1771, Sep. 2013.
- [22] Ali, A. 2001. Macroeconomic variables as common pervasive risk factors and the empirical content of the Arbitrage Pricing Theory. Journal of Empirical finance, 5(3): 221–240.

