IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

PARALLEL ARCHITECTURE OF POWER-OF-TWO MULTIPLIERS FOR FPGAs

Dr. U.Saravanakumar*1, Professor Head of the department, M.Madhumitha*2 PG Student, Muthayammal Engineering College, Department of ECE

ABSTRACT

Multiplication is a fundamental arithmetic operation that significantly influences the performance of modern digital systems such as processors, digital signal processing units, and embedded systems. Conventional multiplier architectures, including array and sequential multipliers, suffer from limitations such as high propagation delay, increased power consumption, and poor scalability for higher bit-width operations. This project presents the design and implementation of an efficient parallel multiplier architecture to improve multiplication speed and reduce computational delay. The proposed system generates partial products simultaneously and accumulates them using optimized adder structures. Modified Booth Encoding is employed to reduce the number of partial products, while carry save adders are used to minimize carry propagation during partial product accumulation. A high-speed final adder produces the accurate multiplication result. The proposed architecture supports both signed and unsigned multiplication using a unified design, thereby reducing hardware redundancy and improving flexibility. The design is described using hardware description language and verified through simulation and synthesis. The results demonstrate improved performance compared to conventional multiplier designs, making the proposed parallel multiplier suitable for modern VLSI and high-speed digital applications.

KEYWORDS: FPGA, Xlinx ISE, Modalism, Matrix Multiplication, Parallel Block-Scheduling.

INTRODUCTION

GENERAL INTRODUCTION

In modern digital systems, arithmetic units play a vital role in determining the overall performance of processors, digital signal processing (DSP) systems, and application-specific integrated circuits (ASICs). Among the various arithmetic operations, multiplication is one of the most fundamental and computationally intensive operations. Multipliers are extensively used in microprocessors, microcontrollers, digital filters, image processing systems, and communication hardware, where high speed and low power consumption are critical requirements.

With the rapid growth of portable and high-performance electronic systems, there is an increasing demand for arithmetic circuits that can operate at higher speeds while consuming minimal power and silicon area. Parallel multipliers have emerged as an effective solution to meet these requirements, as they perform multiplication by generating and adding partial products simultaneously, thereby significantly reducing computation time when compared to serial multiplication techniques.

The design of an efficient parallel multiplier requires careful consideration of speed, power consumption, and hardware complexity. As technology scales and clock frequencies increase, optimizing the multiplier architecture becomes essential to enhance system throughput and energy efficiency. Hence, the development of optimized parallel multiplier architectures has become a major research focus in the field of VLSI system design.

PROBLEM STATEMENT

Conventional multiplier architectures often suffer from large propagation delays, high power consumption, and increased area utilization due to the sequential generation and accumulation of partial products. As the operand bit-width increases, these issues become more pronounced, leading to degraded performance in high-speed digital systems.

In parallel multiplier designs, although higher speed can be achieved by performing multiple operations simultaneously, challenges such as increased switching activity, complex interconnections, and higher power dissipation arise. Moreover, achieving efficient multiplication for both signed and unsigned numbers using a single architecture further complicates the design.
Therefore, there is a need for a parallel multiplier architecture that can effectively reduce computation delay while maintaining acceptable power consumption and hardware complexity. Addressing these challenges is essential to improve the performance of arithmetic units used in modern digital and VLSI-based systems.

OBJECTIVE

The primary objective of this project is to design and analyze an efficient parallel multiplier architecture based on the generation and summation of partial products in parallel. The project aims to improve multiplication speed while optimizing power consumption and hardware utilization.

The specific objectives of the proposed work are:
1. To study existing multiplier architectures and identify their limitations in terms of speed, power, and area.
2. To design a parallel multiplier capable of performing high-speed multiplication for signed and unsigned numbers.
3. To implement the proposed parallel multiplier using hardware description language (HDL).
4. To analyze the performance of the multiplier in terms of delay, power consumption, and overall efficiency through simulation and synthesis.

EXISTING SYSTEM

In digital systems, multiplication is one of the most critical arithmetic operations, directly influencing the overall performance of processors, digital signal processors, and arithmetic logic units. Early digital designs primarily focused on correctness and simplicity, leading to the development of conventional multiplier architectures such as array multipliers and sequential multipliers. These designs were sufficient for low-speed and low-complexity applications but have become inadequate with the increasing demand for high-speed, low-power, and area-efficient computation.

The existing multiplier systems are largely based on straightforward partial product generation and accumulation techniques. In these architectures, the multiplication process involves generating partial products using logical AND operations and summing them using simple adder structures. Although these approaches are easy to design and implement, they do not efficiently utilize parallelism, which results in increased computation delay and power consumption. As the bit-width of operands increases, the performance degradation becomes more severe.

Another major characteristic of existing systems is the reliance on carry- propagation-based addition for partial product accumulation. This creates long critical paths, limiting the maximum operating frequency of the multiplier. Furthermore, traditional designs often require separate hardware blocks for signed and unsigned multiplication, increasing design complexity and silicon area. These issues collectively restrict the scalability and applicability of conventional multiplier architectures in modern VLSI systems.

The following subsections discuss the key elements and approaches used in existing multiplier systems, along with their operational principles and inherent drawbacks.

LIMITATIONS OF EXISTING SYSTEM

- The existing multiplier architectures suffer from several critical limitations that hinder their performance in modern digital systems. One of the primary limitations is high propagation delay caused by sequential partial product accumulation and long carry chains. As operand size increases, the delay grows rapidly, limiting scalability and high-frequency operation.

- Another major limitation is excessive power consumption resulting from unnecessary switching activity during partial product generation and accumulation. Conventional designs do not efficiently exploit data patterns, leading to wasted energy. Furthermore, the use of separate architectures for signed and unsigned multiplication increases hardware complexity and area overhead.

- **PROPOSED SYSTEM**

The proposed system introduces an optimized parallel multiplier architecture designed to overcome the performance limitations of conventional multiplier systems. With the rapid advancement of digital processing applications such as signal processing, image processing, cryptography, and high-performance computing, the demand for fast and power-efficient arithmetic units has increased significantly. Multiplication, being a fundamental arithmetic operation, directly impacts system throughput and energy efficiency, making multiplier optimization a critical design concern

In the proposed system, multiplication is carried out by generating and accumulating partial products simultaneously, rather than sequentially. This parallelism drastically reduces computation time and improves overall performance. Unlike traditional array and sequential multipliers, where partial products are generated and summed over multiple stages with long carry propagation delays, the proposed architecture exploits concurrent operations to shorten the critical path and increase operating frequency

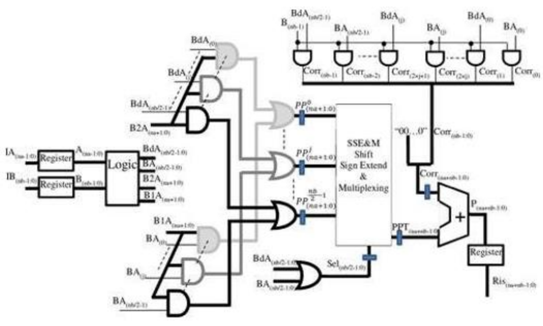

Fig 3.3 Proposed Parallel Multiplier Architecture

A key feature of the proposed parallel multiplier is the use of Modified Booth Encoding for partial product generation. Modified Booth Encoding recodes the multiplier bits to reduce the number of partial products generated during multiplication. By grouping adjacent bits of the multiplier and encoding them into signed digits, the number of partial products is reduced by approximately half when compared to conventional binary multiplication. This reduction directly contributes to lower delay, reduced switching activity, and improved power efficiency.

Following partial product generation, the proposed system employs carry save adders (CSA) arranged in a tree-like structure to accumulate partial products efficiently. Carry save adders are well suited for parallel multiplier architectures because they eliminate immediate carry propagation during intermediate addition stages. Instead of propagating carries through multiple adder levels, CSAs store carry and sum outputs separately, allowing multiple additions to occur concurrently. This significantly reduces the critical path delay associated with carry propagation.

The final summation stage of the proposed multiplier uses a high-speed adder, such as a carry look-ahead adder or a fast parallel prefix adder, to combine the final sum and carry outputs produced by the CSA tree. This ensures that the final result is generated with minimal delay. By restricting carry propagation to the final stage only, the overall latency of the multiplier is greatly reduced.

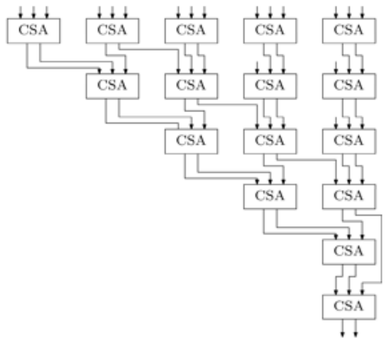

Fig 3.4 Carry Save Adder Based Partial Product Reduction

Another important aspect of the proposed system is its ability to support both signed and unsigned multiplication within a unified architecture. Proper sign extension and encoding logic are incorporated to handle negative operands without requiring separate hardware blocks. This unified approach reduces hardware redundancy, simplifies control logic, and improves flexibility in arithmetic logic units where both signed and unsigned operations are frequently required.

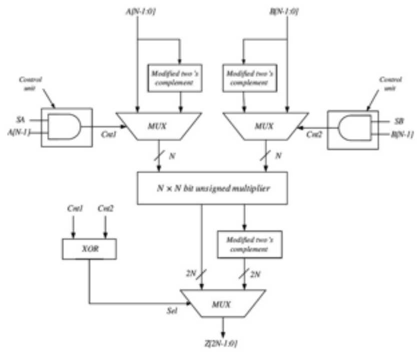

Fig 3.5 Signed and Unsigned Operation in Parallel Multiplier

The proposed parallel multiplier architecture is described using a hardware description language and is suitable for implementation on FPGA as well as ASIC platforms. During synthesis, the design can be optimized for speed, power, or area depending on application requirements. Simulation and synthesis results demonstrate that the proposed system achieves significant improvement in operating frequency and power efficiency when compared to existing multiplier architectures.
Furthermore, the proposed system is scalable and can be extended to higher bit- width multipliers without major architectural changes. As operand size increases, the benefits of parallelism and reduced

partial product generation become more pronounced. This makes the proposed parallel multiplier suitable for modern high- performance digital systems that demand fast, reliable, and energy-efficient arithmetic computation.

Overall, the proposed system provides an effective solution to the challenges faced by conventional multiplier architectures by combining reduced partial product generation, efficient accumulation, and unified signed–unsigned operation. These features collectively enhance performance, reduce power consumption, and improve hardware utilization, making the proposed parallel multiplier a strong candidate for next-generation VLSI systems.

NEED FOR PROPOSED SYSTEM

- The increasing complexity of modern digital systems has created a strong need for efficient arithmetic units capable of performing high-speed computations with minimal power consumption. Multipliers are extensively used in processors, digital signal processing units, and embedded systems, where their performance directly influences overall system efficiency. Conventional multiplier architectures fail to meet these requirements due to high delay, excessive power consumption, and limited scalability.

- One of the primary needs for the proposed parallel multiplier system is the demand for high-speed multiplication. Applications such as real-time signal processing, multimedia processing, and cryptographic algorithms require fast arithmetic operations to meet stringent timing constraints. The proposed system addresses this need by exploiting parallelism in partial product generation and accumulation, thereby significantly reducing computation time.

- Another important requirement is power efficiency. With the widespread use of portable and battery-powered devices, reducing power consumption has become a critical design objective. Traditional multiplier architectures suffer from excessive switching activity, leading to high dynamic power dissipation. By reducing the number of partial products and minimizing carry propagation, the proposed system lowers switching activity and improves energy efficiency.

- The need for hardware optimization and flexibility also motivates the proposed design. Supporting both signed and unsigned multiplication using a single architecture eliminates the need for separate hardware blocks, reducing silicon area and design complexity. This unified approach simplifies integration into arithmetic logic units and improves overall system flexibility.

- Scalability is another crucial factor driving the need for the proposed system. As operand sizes increase in advanced computing applications, conventional multipliers experience significant performance degradation. The proposed parallel multiplier architecture scales efficiently with bit-width, making it suitable for high-resolution and high-precision applications.

- Finally, the proposed system addresses the need for modern VLSI compatibility. The architecture is well suited for implementation using contemporary design tools and technologies, enabling optimization at both RTL and physical design levels. This makes the proposed parallel multiplier a practical and efficient solution for current and future digital systems.

SOFTWARE REQUIREMENTS

The design and implementation of the proposed parallel multiplier require a reliable and efficient software environment that supports hardware description, simulation, synthesis, and performance analysis. Software tools play a crucial role in validating the correctness of the design and analyzing key parameters such as delay, power consumption, and resource utilization. In this project, industrystandard electronic design automation (EDA) tools are used to model, simulate, and verify the parallel multiplier architecture.

The selected software tools provide support for hardware description languages, enable behavioral and structural modeling, and allow detailed timing and functional verification. These tools ensure that the proposed multiplier design meets the required performance objectives and can be efficiently implemented on programmable logic devices or ASIC platforms.

USER INTERFACE

The primary user interface of the selected design tool is a project-based graphical environment that allows designers to manage source files, simulation settings, and synthesis processes efficiently. The interface provides a hierarchical view of the design modules, enabling easy navigation and organization of hardware description language files used in the parallel multiplier design.
The user interface typically includes an integrated code editor for writing and modifying HDL source files, a console window for displaying compilation and simulation messages, and a process window for controlling synthesis and implementation steps. This organized interface helps designers track design progress and identify errors during compilation or simulation. The availability of waveform viewers further aids in analyzing the functional behavior of the multiplier by visualizing signal transitions over time.

TECHNOLOGY

Xilinx designs, develops and markets programmable logic products, including integrated circuits (ICs), software design tools, predefined system functions delivered as intellectual property (IP) cores, design services, customer training, field engineering and technical support. Xilinx sells both FPGAs and CPLDs for electronic equipment manufacturers markets such as communications, industrial, consumer, automotive and data processing.

Xilinx's FPGAs have been used for the ALICE (A Large Ion Collider Experiment) at the CERN European laboratory on the French-Swiss border to map and disentangle the trajectories of thousands of subatomic particles.

Xilinx has also engaged in a partnership with the United States Air Force Research Laboratory’s Space Vehicles Directorate to develop FPGAs to withstand the damaging effects of radiation in space, which are 1,000 times less sensitive to space radiation than the commercial equivalent, for deployment in new satellites.

PROJECT DESIGN

Project design is a crucial phase in the development of any digital system, as it translates theoretical concepts into a practical and implementable architecture. In the proposed work, the design phase focuses on defining the structure of the parallel multiplier, identifying input and output requirements, and evaluating the feasibility of implementing the design using modern VLSI tools and technologies. The project design ensures that the proposed parallel multiplier meets the desired performance objectives in terms of speed, power efficiency, and scalability.

The design of the parallel multiplier is carried out by carefully organizing partial product generation, accumulation, and final summation stages. Emphasis is placed on achieving high throughput while maintaining efficient hardware utilization. The design methodology follows a systematic approach to ensure correctness, reliability, and ease of implementation.

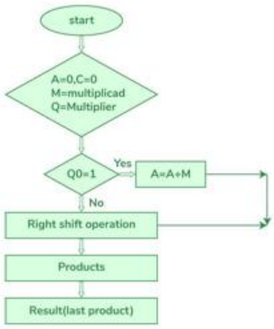

Fig 5.1 Overall Design Flow of Parallel Multiplier

SYSTEM TESTING

System testing is a critical phase in the development of the proposed parallel multiplier, as it ensures that the design operates correctly under all specified conditions and meets the required performance objectives. In digital hardware design, testing primarily focuses on functional correctness, timing behavior, and performance validation rather than traditional software-based testing methods. The purpose of system testing in this project is to verify that the parallel multiplier produces accurate results for both signed and unsigned operations and operates within acceptable timing and power constraints.

Testing is performed after the completion of design and simulation stages to identify errors, verify design logic, and ensure reliability. A well-defined testing strategy helps confirm that the multiplier architecture functions as intended and is suitable for integration into larger digital systems.

TYPES OF TESTS

- Functional Testing
- Unit Testing
- Integration Testing
- Test Strategy And Approach
- Test Objectives
- Features To Be Tested

CONCLUSION

In this project, an efficient parallel multiplier architecture has been designed, implemented, and analyzed to address the limitations of conventional multiplier systems. Multiplication being a fundamental arithmetic operation in digital systems, its performance has a significant impact on overall system efficiency, particularly in high-speed and low-power applications. The proposed work focuses on improving multiplication speed while maintaining acceptable power consumption and hardware utilization.

The design of the parallel multiplier is based on simultaneous generation and accumulation of partial products, which significantly reduces computation time compared to sequential and array-based multiplier architectures. The use of Modified Booth Encoding reduces the number of partial products generated, while carry save adders enable fast and efficient partial product accumulation by minimizing carry propagation during intermediate stages. A high-speed final adder ensures accurate and timely generation of the final multiplication result.

Functional verification through simulation confirms the correctness of the proposed multiplier for both signed and unsigned operations. Timing and power analysis demonstrate that the proposed architecture offers improved performance compared to traditional multiplier designs. The unified support for signed and unsigned multiplication reduces hardware redundancy and enhances flexibility in arithmetic logic units.

Overall, the proposed parallel multiplier successfully meets the design objectives of high speed, improved efficiency, and scalability. The results obtained from simulation and synthesis validate the effectiveness of the proposed architecture, making it suitable for use in modern digital processing systems and VLSI applications.

FUTURE ENHANCEMENT

Although the proposed parallel multiplier achieves significant improvements over conventional designs, there are several directions in which the work can be extended to further enhance performance and applicability. One possible future enhancement is the implementation of the multiplier with higher bit-width operands, such as 32-bit or 64-bit multiplication, to meet the requirements of advanced computing and signal processing applications.

Another important enhancement is the introduction of pipelining techniques to further increase operating frequency and throughput. By dividing the multiplier into multiple pipeline stages, the design can support higher clock speeds and be better suited for real-time and high-performance systems.

Power optimization can be further improved by incorporating low-power techniques such as clock gating, operand isolation, or dynamic voltage scaling. These techniques can reduce unnecessary switching activity and improve energy efficiency, particularly in portable and battery-operated devices.

Future work may also explore FPGA and ASIC implementations using advanced technology nodes to evaluate the performance of the proposed architecture under real hardware constraints. Additionally, the use of alternative adder structures or compressor-based partial product reduction techniques could further reduce delay and area.

Finally, integrating the proposed parallel multiplier into larger systems such as arithmetic logic units, digital signal processors, or cryptographic engines would demonstrate its practical applicability and performance benefits in real-world applications.

REFFERENCE

[1] J. Sithara, R. Kannan, and S. Prakash, “A high-speed parallel multiplier architecture using hybrid Booth–Wallace reduction,” International Journal of VLSI Design and Communication Systems, vol. 16, no. 2, pp. 45–56, Mar. 2025.

[2] C. Kalamani and P. Arulmozhivarman, “Energy-efficient Wallace tree multiplier using optimized compressor structures,” Microelectronics Journal, vol. 141, pp. 106– 118, Jan. 2024.

[3] S. Cekli and M. Yilmaz, “High-speed pipelined radix-16 Booth multiplier architecture for FPGA implementation,” IEEE Access, vol. 12, pp. 38821–38833, Feb. 2024.

[4] S. N. Leela and R. Anand, “Optimized parallel multiplier design using compressor- based partial product reduction,” Journal of Circuits, Systems and Computers, vol. 32, no. 8, pp. 1–15, Aug. 2023.

[5] A. Sadeghi and M. Hosseini, "A counter-based Wallace-tree multiplier with hybrid full-adder architecture," Frontiers of Information Technology & Electronic Engineering, vol. 23, no. 4, pp. 521–533, Apr. 2022.

[6] B. Srinivasu and K. Sridharan, "Unified signed and unsigned parallel multiplier using modified Booth encoding," International Journal of Electronics, vol. 108, no. 11, pp. 1924–1938, Nov. 2021.

[7] H. Sirugudi, S. Gadgil, and C. Vudadha, "A high-speed modified Booth encoder multiplier for signed and unsigned numbers," International Journal of Engineering Research & Technology, vol. 9, no. 6, pp. 112–118, June 2020.

[8] S. S. Saste and A. G. Sawant, "Design and implementation of radix-4 based multiplication on FPGA," International Journal of Engineering Research & Technology, vol. 5, no. 9, pp. 245–250, Sept. 2016.

[9] R. Mudassir and Z. Abid, "New parallel multipliers based on low power adders," International Journal of Computer Applications, vol. 14, no. 5, pp. 19–24, Feb. 2015.

[10] M. C. Wen, S. J. Wang, and Y. N. Lin, "Low-power parallel multiplier with column bypassing," Electronics Letters, vol. 37, no. 7, pp. 430–432, Apr. 2001.

[11] J. Zhang, H. Zhang, Q. Sun, and R. Zhang, "Design of a high-speed low-power parallel multiplier using advanced modified Booth encoding," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 33, no. 1, pp. 88–99, Jan. 2025.

[12] D. K. Jayaraman Rajanediran, "Hybrid radix-16 Booth encoding based parallel multiplier for FFT applications," Integration, the VLSI Journal, vol. 88, pp. 65–75, Mar. 2024.

[13] J. Ponraj, M. Karthikeyan, and S. Ravi, "High-performance multiply–accumulate unit using counter-based Wallace-tree multiplier," Microprocessors and Microsystems, vol. 95, pp. 104–116, Oct. 2023.

[14] S. Tabrizchi, H. Sharifi, and K. Navi, "A novel design approach for compressor- based multipliers," Circuits, Systems, and Signal Processing, vol. 41, no. 6, pp. 3124– 3142, June 2022.

[15] C. Adiono, A. Taufiq, and R. V. Wicaksana, "Compact radix-4 Booth multiplier design for low-power digital systems," International Journal of Electronics, vol. 106, no. 9, pp. 1425–1437, Sept. 2019.