IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 14, Issue 12, December 2025

DOI: 10.17148/IJARCCE.2025.1412129

FPGA IMPLEMENTATION OF BOOTH MULTIPLIER USING RADIX-4 ALGORITHM

Brunda A[1], Chakravarthi M N[2], Madhushree S[3], Dr. Samyuktha S[4]

Student, ECE, SJB Institute of Technology, Bengaluru, India[1]

Student, ECE, SJB Institute of Technology, Bengaluru, India[2]

Student, ECE, SJB Institute of Technology, Bengaluru, India[3]

Assistant Professor, ECE, SJB Institute of Technology, Bengaluru, India[4]

Abstract: High-speed arithmetic units are a critical requirement in modern digital signal processing systems and general-purpose processors. Among these units, multipliers and multiplier-and-accumulator (MAC) blocks significantly influence overall system performance. This paper presents a novel MAC architecture based on the Radix-4 Modified Booth multiplication algorithm, implemented on a Xilinx FPGA platform. The proposed design integrates multiplication and accumulation operations using an efficient hybrid adder structure, resulting in improved computational speed and reduced hardware complexity. The Modified Booth encoding technique minimizes the number of generated partial products by approximately half compared to conventional multiplication methods, thereby enhancing processing efficiency. By optimizing partial product generation and accumulation, the proposed architecture achieves faster arithmetic operations, making it suitable for high-performance DSP applications.

Keywords: Radix-4 Booth Multiplier, Multiplier-Accumulator (MAC), Modified Booth Encoding, FPGA Implementation, Digital Signal Processing, Hybrid Adder.

I. INTRODUCTION

The continuous growth of multimedia and communication technologies has significantly increased the demand for high-speed and real-time digital signal processing (DSP) systems. Applications such as audio processing, image and video processing, and large-scale data computation rely extensively on arithmetic operations, particularly multiplication and accumulation. In DSP architectures, the performance of fundamental operations like filtering, convolution, and transform computations is strongly influenced by the efficiency of multiplier and multiplier-accumulator (MAC) units. Most DSP algorithms, including widely used transforms such as the Discrete Cosine Transform (DCT) and Discrete Wavelet Transform (DWT), involve repeated multiplication and addition. As a result, the overall execution speed and computational efficiency of these systems depend largely on the speed and design of the multiplier architecture. Designing fast and area-efficient multipliers is therefore a critical requirement for modern VLSI-based DSP systems. High-performance arithmetic circuits demand compact designs to reduce propagation delay and interconnect complexity. Single-chip implementations are preferred as they minimize wire delays and facilitate seamless integration into larger processing systems. In this context, Modified Booth Encoding (MBE) based on the Radix-4 algorithm offers an effective solution for high-speed signed multiplication. By encoding operands in two's complement form, the Radix-4 Booth algorithm reduces the number of partial products by approximately half when compared to conventional multiplication methods, leading to improvements in speed and hardware efficiency. This paper presents the design and implementation of an extendable Radix-4 Booth multiplier architecture, focusing on the core building blocks such as the Booth encoder, partial product generator, and hybrid adder. The proposed approach aims to achieve reduced computation time and optimized hardware utilization, making it suitable for high-speed DSP and embedded system applications.

1.1 Motivation for work

Multiplication is a core arithmetic operation in digital signal and image processing systems, where speed and efficiency directly impact overall performance. As DSP applications such as digital filtering, image processing, and video processing demand high computational throughput, there is a strong need for optimized multiplier architectures. Implementing efficient Radix-2 and Radix-4 Booth multipliers on FPGA platforms enables reduced computation time and improved hardware utilization. The use of structured design techniques such as state diagrams and ASM charts further supports reliable and scalable Verilog implementations, motivating the development of high-speed Booth multiplier architectures for modern DSP applications.

1.2 Objective of work

- To design and implement high-speed Radix-2 and Radix-4 Booth multipliers optimized for FPGA platforms.
- To reduce the number of partial products and improve overall computational efficiency using Modified Booth Encoding.
- To develop the design using state diagrams and ASM charts for accurate Verilog implementation.
- To enhance performance in digital signal processing applications by optimizing speed, area, and power consumption.
- To efficiently utilize FPGA resources while ensuring reliability through thorough verification and testing.

II. LITERATURE SURVEY

1. **Multiplierless unity-gain SDF FFTs. Trans Very Large ScaleInteger (VLSI) System (Garrid M, Andersson R, 2016) :**

   Garrid M. and Andersson R. proposed a multiplierless FFT design using the SDF architecture. The method eliminates the need for explicit multipliers by carefully designing the rotators so that the entire FFT output can be scaled by powers of two. This scaling allows the FFT to achieve unity gain without requiring additional compensation circuits. The approach simplifies hardware implementation, reduces computational complexity, and maintains accuracy. It demonstrates an efficient solution for high-speed DSP applications where hardware resources are limited.

2. **FPGA mapping pipeline of FFT (Qureshi F, Gustafsson, 2017) :**

   Qureshi F. and Gustafsson proposed a pipelined FFT implementation optimized for FPGA platforms. By taking advantage of the architectural features of the target FPGA, the design achieved significantly improved performance in terms of speed and resource utilization. The approach carefully maps FFT computations onto the FPGA, enhancing throughput while minimizing hardware overhead. This method provides an efficient solution for high-performance digital signal processing applications on reconfigurable hardware.

3. **VLSI Implementation Real fast fourier transform (Satya Sai Ram M, 2019) :**

   Satya Sai Ram M. presented a VLSI implementation of real Fast Fourier Transform (FFT) using the Radix-2 Decimation-in-Frequency (DIF) approach. The design efficiently employs a single dual-port RAM (DRAM) to store intermediate results, avoiding the need for multiple memory units. This reduces hardware complexity, optimizes area utilization, and improves computational efficiency. The architecture provides a practical and resource-efficient solution for real-time DSP applications on VLSI platforms.

4. **A New Multichannel Parallel Real- time FFT Algorithm for a Solar Radio Observation System Based on FPGA (Lei Zhang and Yuan Y. Zhang, 2022) :**

   Lei Zhang and Yuan Y. Zhang presented a multichannel parallel real-time FFT algorithm (MPR-FFT) designed for FPGA-based solar radio observation systems. The proposed approach reduces FPGA resource utilization while improving real-time processing speed. By enabling parallel computation across multiple channels, it enhances throughput and efficiency, offering a practical solution for high-speed, real-time digital signal processing applications.

5. **VHDL implementation of a 16-bit Radix-4 using Xilinx ISE 14.4 FPGA platform (Sulbha Bhongale and Rajendra Patel, 2023) :**

   Sulbha Bhongale and Rajendra Patel presented a VHDL implementation of a 16-bit Radix-4 multiplier on the Xilinx ISE 14.4 FPGA platform. The design focuses on high-speed and area-efficient operation by minimizing the number of partial products, achieving a reduced delay of 26.32 ns compared to conventional designs. This work demonstrates an effective method for enhancing multiplication performance in digital signal processing applications while optimizing FPGA resource usage.

III. DESIGN AND IMPLEMENTATION

3.1 Proposed System Overview

The FPGA development board interfaces with a computer running design and simulation software, enabling FPGA programming and output monitoring. The board features switches, LEDs, and connectors for external connectivity. Real-time multiplication results are displayed on a 16×2 LCD screen. Ribbon cables link modules, facilitating communication between the multiplier logic and output devices. The Radix-4 Booth algorithm efficiently multiplies signed binary numbers, reducing partial products and boosting speed while conserving FPGA resources. LEDs and peripherals indicate intermediate signals or operational status. This setup enables practical verification of high-speed arithmetic operations on FPGA platforms, showcasing real-time multiplication with optimized area and speed.

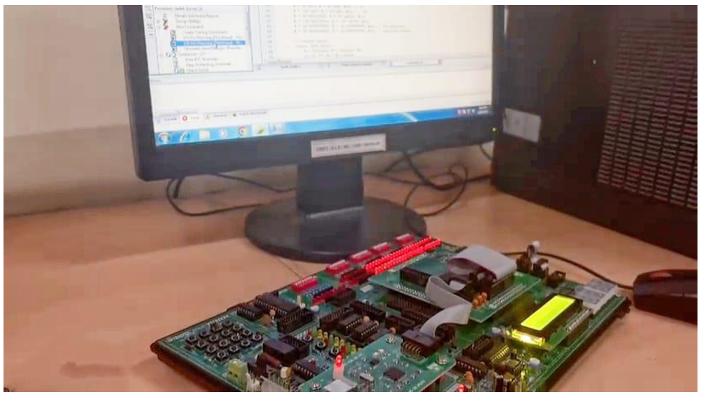


Fig. 1 FPGA multiplication experimental setup

3.2 Block Diagram Description

The Block Diagram represents the operational steps of a Radix-4 Booth multiplier. The process begins with two inputs: the multiplicand $A$and the multiplier B. The multiplier is first processed through the Booth Encoder, which converts it into a form suitable for generating partial products efficiently. Both the encoded multiplier and the multiplicand are then fed into the Booth Selector, which determines the appropriate multiples of the multiplicand according to the encoded multiplier values. These selected values are passed to the Partial Products Addition Array, where all the partial products are summed systematically. The resulting sum is then processed by the Final Result Forming Adder to produce the final multiplication output. The final result, Q, represents the product of the two signed binary numbers. This structured approach reduces the number of partial products, optimizing both speed and resource usage in hardware implementation.

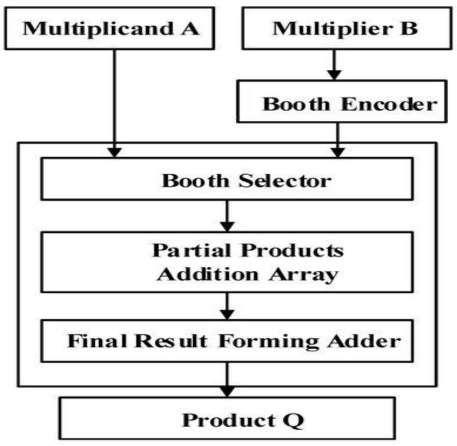


Fig 3.2 Block Diagram

3.3 Implementation

The Radix-4 Booth multiplier was implemented on an FPGA platform to evaluate its performance in terms of speed, accuracy, and hardware efficiency. The multiplier design employs Booth encoding to process signed binary inputs and generate reduced partial products, which significantly lowers computational complexity. These partial products are generated, shifted, and accumulated in a clock-controlled manner to ensure proper synchronization and reliable operation. The use of sequential control logic resulted in stable and deterministic outputs after the required clock cycles. Functional verification confirmed correct handling of both positive and negative operands across various test cases. Due to the reduced number of adders and shift operations, the overall resource utilization on the FPGA was minimized. The implementation demonstrated improved processing speed and efficient logic usage, making the design well suited for high-performance arithmetic operations. The results validate the effectiveness of the Radix-4 Booth algorithm for FPGA-based multiplier architectures.

3.4 Hardware Environment:

- FPGA development board with configurable logic blocks for parallel arithmetic operations
- Onboard clock source for synchronization
- Input switches for test inputs
- LEDs/display units for output observation
- Programming cable for FPGA-computer connection

3.5 Software Environment:

- FPGA design tool for coding, synthesis, and implementation
- Simulation tool for pre-testing verification
- Timing analysis for operational correctness
- Device programming tool for FPGA configuration

3.6 Data Preparation:

- Binary inputs with two's complement for signed numbers
- Fixed bit-width for multiplier/multiplicand
- Test data: positive, negative, zero values
- Inputs via simulation vectors or FPGA switches
- Radix-4 Booth encoding and partial product generation
- Control signals for alignment and summation
- Output verified against expected results

3.7 System workflow Summary:

- Radix-4 Booth multiplier designed using HDL in Xilinx
- Simulated for functional correctness
- Testbench created and executed for validation
- Design synthesized and implemented
- Programmed onto FPGA development kit
- Output displayed on onboard LEDs, confirming implementation

IV. RESULTS

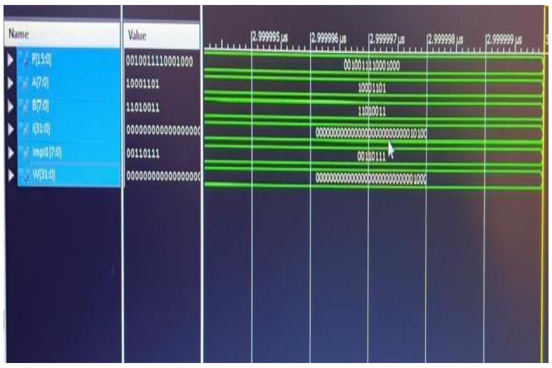
The FPGA implementation of the Radix-4 Booth multiplier was successfully verified through simulation and hardware synthesis. The obtained results confirm correct signed multiplication with proper generation and accumulation of partial products. Compared to conventional multiplication, the Radix-4 approach significantly reduces the number of partial products, leading to faster computation and efficient hardware utilization. The output waveforms demonstrate stable, glitch-free operation synchronized with the system clock, validating the reliability of the proposed design for high-speed digital arithmetic applications.

4.1 Simulation Results

The simulation results verify the functional correctness efficiency of the FPGA-based Booth multiplier designed using the Radix-4 algorithm. Signed binary values are applied as inputs to the multiplicand and multiplier, which are processed through Radix-4 Booth encoding by grouping the multiplier bits in overlapping sets of three. This approach significantly decreases the number of generated partial products when compared to conventional multiplication techniques, thereby improving computational speed and reducing hardware complexity. The simulation waveforms illustrate the sequential formation, alignment, and accumulation of partial products under clock-controlled operation. All control and data signals

DOI: 10.17148/IJARCCE.2025.1412129

exhibit proper synchronization throughout the computation cycle. The output signal stabilizes to the correct product value after the completion of arithmetic operations, with no observable glitches or timing violations. These results demonstrate reliable timing behavior and confirm that the proposed design achieves accurate signed multiplication with enhanced performance, making it well suited for high-speed digital arithmetic and signal processing applications.



```
Name        Value
P[15:0]     0010011110001000
A[7:0]      10001101
B[7:0]      11010011
i[31:0]     000000000000000000
impl1[7:0]  00110111
W[31:0]     000000000000000000
```

```
0010011110001000
10001101
11010011
00000000000000000000000000010100
00110111
00000000000000000000000000001000
```

Fig Output waveform of the Radix-4 Booth multiplier

Fig FPGAboard showing the implemented design.

The FPGA implementation of the Radix-4 Booth Multiplier successfully produced accurate outputs for all tested input combinations, including both positive and negative numbers. The Radix-4 encoding reduced the number of partial products, resulting in faster computation compared to conventional multiplication methods. Clock-controlled operations ensured synchronized generation, shifting, and accumulation of partial products, producing stable and glitch-free outputs. Hardware resource utilization was optimized, requiring fewer adders and shift operations, making the design efficient for FPGA deployment. Overall, the implementation demonstrated improved processing speed, reduced computational complexity, and reliable performance, confirming the effectiveness of the Radix-4 Booth algorithm for high-speed arithmetic applications.

V. CONCLUSION

The project demonstrates the design and implementation of a modified Radix-4 Booth multiplier. Simulation results were analyzed for both Radix-2 and Radix-4 algorithms, showing that Radix-4 reduces the number of partial products and improves computation speed. The multiplier was successfully implemented on an FPGA using appropriate synthesis tools, achieving accurate results, efficient hardware utilization, and reliable performance. This confirms that the Radix-4 Booth algorithm is an effective solution for high-speed arithmetic operations in FPGA-based systems.

VI. FUTURE SCOPE

The algorithm has been implemented using a hybrid adder to sum the partial products in parallel, generating the final output efficiently. The hybrid adder combines the advantages of a carry look-ahead adder and a carry select adder, achieving faster addition compared to conventional methods. This approach can be further enhanced by combining other adder architectures to reduce propagation delay even more. For larger input sizes, Radix-4 multipliers are expected to provide improved performance and higher speed. Such optimizations make the design more suitable for high-performance digital applications requiring fast and reliable multiplication.

REFERENCES

[1]. J. J. F. Cavanagh, *Digital Computer Arithmetic*. New York: McGraw-Hill, 1984.
[2]. ISO/IEC, *Information Technology—Coding of Moving Picture and Associated Audio, MPEG-2 Draft International Standard*, Parts 1–3, 1994.
[3]. ISO/IEC JTC1/SC29 WG1, *JPEG 2000 Part I, Final Draft*, 2000.
[4]. O. L. MacSorley, "High-speed arithmetic in binary computers," *Proc. IRE*, vol. 49, pp. 67–91, Jan. 1961.
[5]. A. D. Booth, "A signed binary multiplication technique," *Quart. J. Math.*, vol. IV, pp. 236–240, 1952.
[6]. G. Goto, T. Sato, M. Nakajima, and T. Sukemura, "A 54×54 regular structured tree multiplier," *IEEE J. Solid-State Circuits*, vol. 27, no. 9, pp. 1229–1236, Sep. 1992.
[7]. J. Fadavi-Ardekani, "M×N Booth-encoded multiplier generator using optimized Wallace trees," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 1, no. 2, pp. 120–125, Jun. 1993.

DOI: 10.17148/IJARCCE.2025.1412129

[8]. N. Ohkubo, M. Suzuki, T. Shinbo, T. Yamanaka, A. Shimizu, K. Sasaki, and Y. Nakagome, "A 4.4 ns CMOS 54×54 multiplier using pass-transistor multiplexer," *IEEE J. Solid-State Circuits*, vol. 30, no. 3, pp. 251–257, Mar. 1995.

[9]. K. Kim and P. Beerel, "A synchronous matrix-vector multiplier for discrete cosine transform," in *Proc. Int. Symp. Low Power Electronics and Design*, pp. 256–261, Jul. 2000.

[10]. T. Y. Tang, C. S. Choy, P. L. Siu, and C. F. Chan, "Design of self-timed asynchronous Booth's multiplier," in *Proc. Asia South Pacific Design Automation Conf.*, pp. 15–16, Jan. 2000.

[11]. M. N. Fahmi, F. Elguibaly, E. Abdel-Raheem, and A. Tawfik, "Area-time efficient fixed-point multiplier-accumulators for inner-product computation," in *Proc. IEEE Int. Conf. Microelectronics*, Dhahran, Saudi Arabia, pp. 189–192, Dec. 1999.

[12]. S. Kim, C. H. Ziesler, and M. C. Thymiou, "Design, verification, and test of a true single-phase 8-bit adiabatic multiplier," in *Proc. 19th Conf. Advances in Research VLSI*, Salt Lake City, UT, pp. 42–58, Mar. 2001.