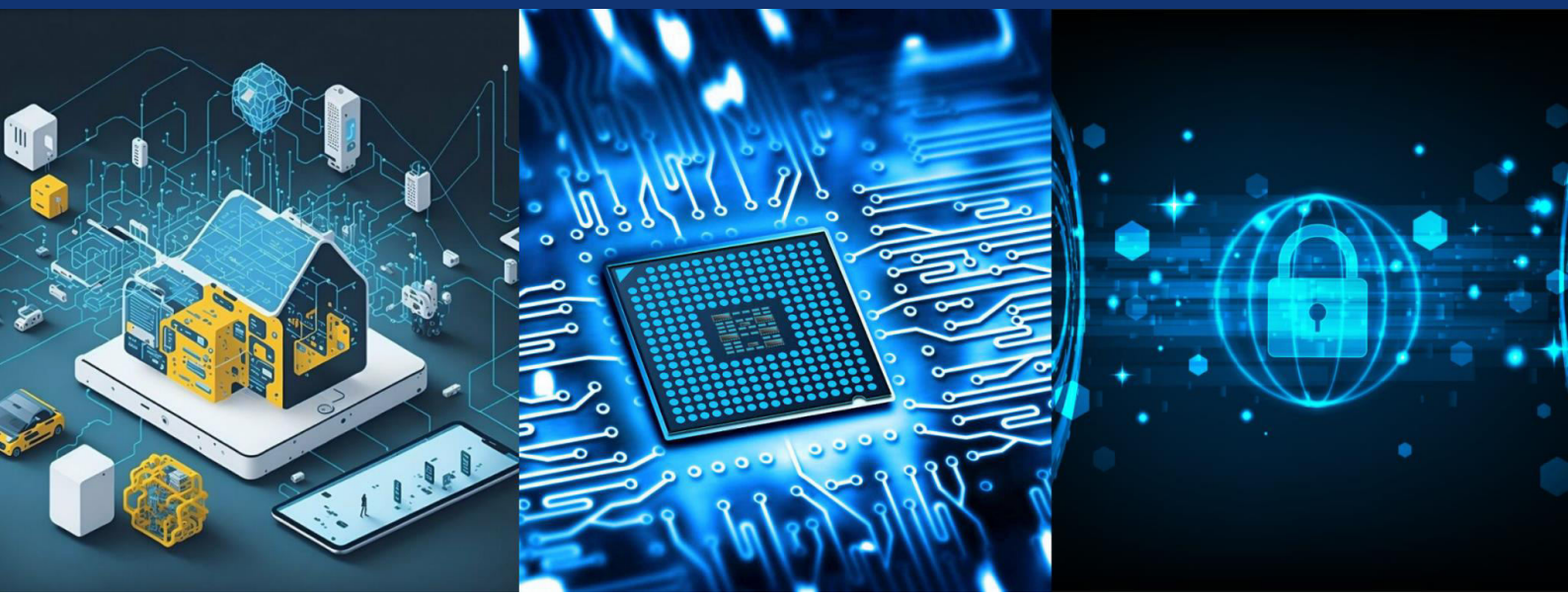
IJIRCCE

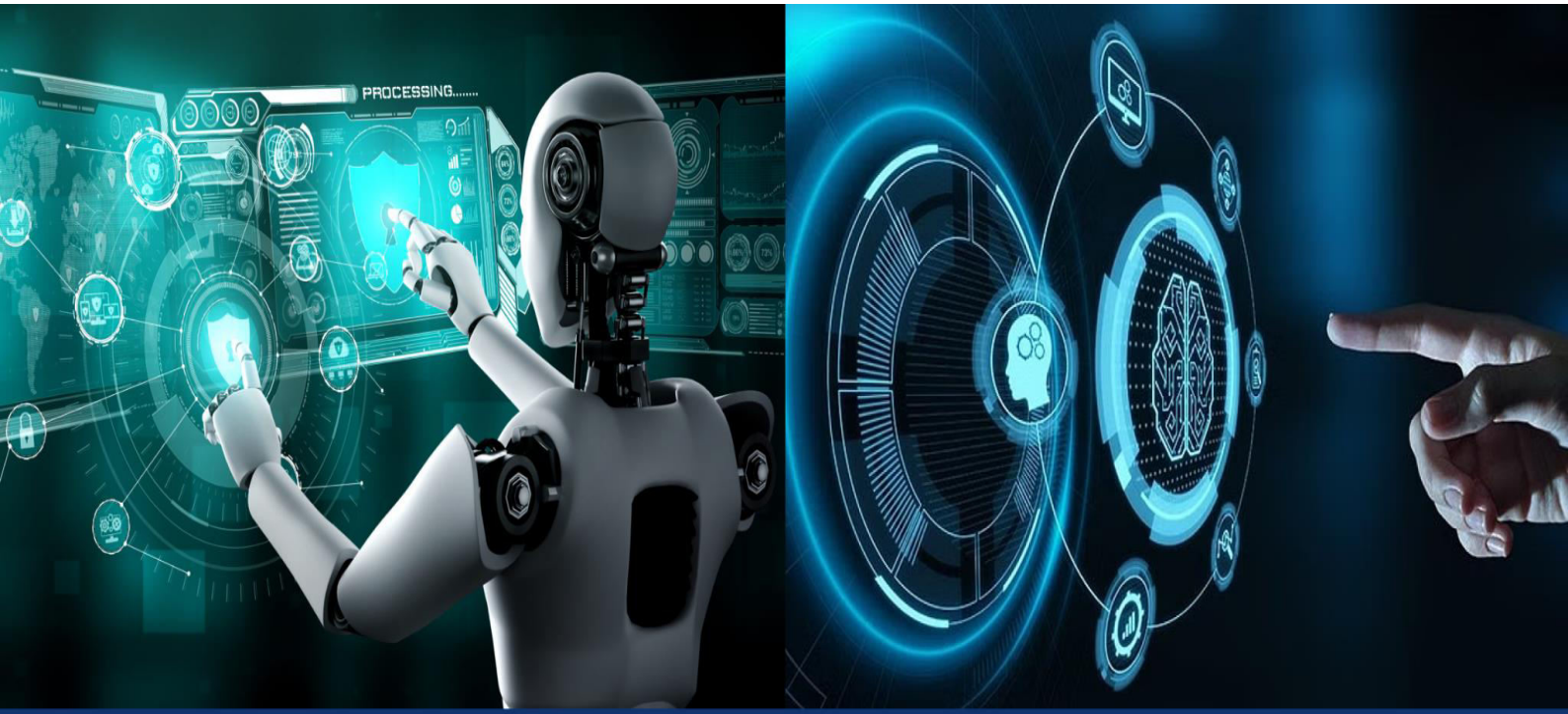
ISSN(O): 2320-9801

ISSN(P): 2320-9798



International Journal of Innovative Research in Computer and Communication Engineering

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 8.771

Volume 13, Issue 4, April 2025

© 2025 IJIRCCE | Volume 13, Issue 4, April 2025| DOI: 10.15680/IJIRCCE.2025.1304179

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Decoder Reduction Approximation Scheme for 16-Bit Accurate Booth Multipliers

D.Sridhar

Associate Professor, Department of ECE, Sri Vasavi Institute of Engineering & Technology (Autonomous), Nandamuru, Pedana, A.P, India

D.P.S.V.Surya Deep, P.Baby Harsha, M.Seshu Lakshmanarao, Ch.Anilkumar

Department of ECE, Sri Vasavi Institute of Engineering & Technology (Autonomous), Nandamuru, Pedana, A.P, India

ABSTRACT: In modern digital signal processing and arithmetic-intensive applications, high-performance multipliers play a critical role in achieving low power consumption, minimal area overhead, and faster computational speed. The Booth multiplier, known for reducing the number of partial products in multiplication, offers a significant improvement in performance compared to conventional methods. However, as the bit-width increases, the complexity and power consumption associated with partial product generation and decoding also increase substantially.This project proposes a **Decoder Reduction Approximation Scheme** for a **16-bit Accurate Booth Multiplier**, aiming to optimize the decoding logic involved in generating Booth-encoded partial products. By strategically simplifying the Booth decoder using approximation techniques while preserving computational accuracy, the design achieves notable reductions in logic complexity, area, and power without compromising the correctness of the final multiplication result.The proposed scheme has been modeled and synthesized using Verilog HDL, with post-synthesis analysis performed on industry-standard EDA tools. The performance of the optimized multiplier is evaluated in terms of delay, area, power, and overall efficiency. Comparative analysis with conventional Booth and accurate multiplier designs demonstrates the effectiveness of the decoder reduction approach in enhancing hardware efficiency while maintaining full accuracy in multiplication results.

KEYWORDS*:* Booth Multiplier, Decoder Reduction, Approximate Scheme, Verilog HDL, Radix-4 and Radix-8 Encoding, FPGA Implementation, Low-Power Design, Digital Single Processing(DSP).

I. INTRODUCTION

Multiplication is a fundamental arithmetic operation in modern computing systems, widely used in applications such as digital signal processing (DSP), machine learning, artificial intelligence (AI), and general-purpose computing. Efficient multiplication is crucial for achieving high-performance computing, reducing power consumption, and optimizing hardware resource utilization. Booth multipliers are among the most commonly used multiplier architectures due to their ability to handle signed numbers efficiently and reduce the number of partial products generated during multiplication. However, traditional Booth multipliers still suffer from certain limitations, such as increased hardware complexity, higher power consumption, and longer propagation delays.To address these challenges, approximation techniques have been introduced to optimize multiplier circuits while maintaining an acceptable level of accuracy. One such approach is the **Decoder Reduction Approximation (DRA) Scheme**, which aims to simplify the Booth multiplier's architecture by reducing the number of decoder units required for the encoding process. By implementing the DRA scheme, the total number of computational elements is reduced, leading to improved resource efficiency, reduced power consumption, and faster computation times.

1. Motivation for Approximate Multipliers: Traditional exact multipliers provide precise computation but often require significant hardware resources, leading to increased power consumption and area overhead. As computing systems become more complex, especially in **low-power embedded systems, AI accelerators, and IoT devices**, the demand for resource-efficient arithmetic units has grown. Approximate multipliers provide a trade-off between accuracy and hardware efficiency by reducing circuit complexity at the cost of a controlled error margin.Approximate computing is particularly effective in error-tolerant applications where minor inaccuracies do not significantly impact overall performance. Examples of such applications include:

- **Convolutional Neural Networks (CNNs):** Deep learning models involve massive matrix multiplications, and small errors in multiplication results do not affect final accuracy significantly.
- **Image and Video Processing:** Image filters, compression algorithms, and video encoding techniques can tolerate minor inaccuracies while achieving significant hardware optimizations.
- **Low-Power Embedded Systems:** Battery-operated devices, such as IoT sensors and wearables, benefit from approximate multipliers as they consume less energy.

2. Booth Multipliers and Logarithmic Multipliers in Approximate Computing

Booth multipliers and logarithmic multipliers are two widely used architectures for implementing approximate multiplication. Both architectures convert input binary numbers into an alternate representation before computing the final product. While Booth multipliers rely on **Booth encoding**, logarithmic multipliers use **logarithmic scaling** to approximate multiplication using addition and shifting operations.

Comparison of Approximate Booth Multipliers and Approximate Logarithmic Multipliers

1. **Approximate Booth Multipliers (ABM):**
   - Use **mixed-radix encoding, high-radix encoding, partial product truncation, and approximate partial product generators** to simplify computations.
   - Suitable for a wide range of error-tolerant applications.
   - Generally more accurate than logarithmic multipliers.
2. **Approximate Logarithmic Multipliers:**
   - Utilize **dynamic truncation, operand trimming, and hybrid architectures** for approximation.
   - Typically offer **higher resource efficiency** but at the cost of greater error rates.
   - Primarily used in deep learning applications where accuracy can be compromised for efficiency.

3. Existing Approximation Techniques in Multipliers:Truncation-Based Approximation: The simplest form of approximation in multipliers is **truncation of output bits**, which reduces the number of bits in the final result. While this technique effectively reduces power consumption and circuit area, it introduces a **static error** across all input values. The truncation technique performs poorly for **small input values**, as the relative error increases significantly when fewer bits contribute to the final result.**Dynamic Truncation with Leading-One Detector (LOD):**Dynamic truncation improves upon simple truncation by **identifying the most significant bits** in the input and retaining only the necessary bits for multiplication. A **Leading-One Detector (LOD)** circuit is used to locate the highest-order '1' in a binary number, allowing the system to **discard less significant bits** dynamically.

- LOD circuits are commonly used in **approximate logarithmic multipliers** to minimize hardware usage.
- The effectiveness of LOD-based truncation relies on the exponential weighting of bits, where **lower-order bits contribute minimally to the final result**.
- The major limitation of LOD techniques in **Booth multipliers** is that Booth encoding uses **overlapping bit slices**, making direct truncation impractical.

4. Proposed Decoder Reduction Approximation (DRA) Scheme: The **Decoder Reduction Approximation (DRA) Scheme** introduces a novel method to simplify Booth multipliers while maintaining an acceptable level of computational accuracy. The key idea is to **reduce the number of Booth decoders** required in the multiplication process by strategically simplifying the encoding and decoding stages.

Key Features of the DRA Scheme:

1. **Reduction of Booth Decoders:**
   - Traditional Booth multipliers require **N/2 decoders** for an N-bit multiplication.
   - The proposed DRA scheme **reduces this requirement to N/4 decoders**, achieving a **50% reduction** in decoder complexity.
2. **Circuit Simplification:**
   - The scheme eliminates redundant computations, reducing the **height of adder trees** used to accumulate partial products.
   - A more compact design leads to **lower power consumption and faster computation times**.
3. **Minimal Accuracy Loss:**
   - The approximation method ensures that the error rate remains within an acceptable range.
   - The proposed scheme achieves **better accuracy** compared to approximate logarithmic multipliers, which have higher error rates.

© 2025 IJIRCCE | Volume 13, Issue 4, April 2025| DOI: 10.15680/IJIRCCE.2025.1304179

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

4. **Scalability and Adaptability:**
    - The DRA scheme is **configurable**, allowing designers to **adjust the level of approximation** based on the application requirements.
    - Suitable for various **error-tolerant applications**, including **AI, DSP, and low-power VLSI circuits**.

5. Performance Comparison and Benefits of the DRA Scheme: The proposed **DRA-based approximate Booth multipliers** have been extensively analyzed and compared against existing state-of-the-art multipliers. The key benefits observed are:

- **50% Reduction in Hardware Complexity:** By reducing the number of Booth decoders, the circuit requires **fewer logic gates** and occupies **less silicon area**.
- **Lower Power Consumption:** With fewer active components, the multiplier operates at a **lower power budget**, making it suitable for embedded applications.
- **Higher Speed and Lower Latency:** The optimized Booth encoding process results in **faster computation times**, crucial for high-performance computing applications.
- **Better Accuracy vs. Approximate Logarithmic Multipliers:** While logarithmic multipliers tend to introduce significant errors, the **DRA-based Booth multipliers maintain accuracy with minimal degradation**.

6. Applications of the DRA-Based Booth Multipliers: The **DRA scheme** is designed to improve computational efficiency in various applications, including:

1. **Neural Network Inference:**
    - Deep learning models require high-speed multiplication operations, and the proposed scheme provides an **optimized balance between accuracy and efficiency**.
2. **Image and Video Processing:**
    - Applications such as **image compression, filtering, and transformation** can benefit from the reduced power consumption of approximate Booth multipliers.
3. **Cryptographic Applications:**
    - Encryption and decryption involve modular multiplication, where optimized multipliers can **enhance security processing speeds**.
4. **Embedded Systems and IoT Devices:**
    - The low-power nature of the **DRA scheme** makes it ideal for battery-powered devices, where energy efficiency is critical.

The **Decoder Reduction Approximation (DRA) Scheme** for Booth multipliers presents an innovative approach to optimizing **power consumption, hardware complexity, and computational speed** in arithmetic circuits. By **reducing the number of Booth decoders by 50%**, the scheme achieves **significant resource savings** without introducing substantial errors. The proposed design outperforms existing approximate logarithmic multipliers, making it a viable solution for **AI accelerators, DSP applications, and low-power VLSI systems**. Future research can explore **adaptive approximation techniques**, where the degree of approximation can be dynamically adjusted based on real-time performance requirements.

II. PROPOSEDSYSTEM

The **Decoder Reduction Approximation Scheme** for a **16-bit Accurate Booth Multiplier** is a design methodology that aims to reduce power consumption, area, and delay by approximating certain operations involved in Booth encoding and decoding without significantly sacrificing accuracy. This technique is especially useful in applications like image processing, neural networks, and DSP systems, where slight inaccuracies are tolerable in exchange for improved efficiency.

1. Overview of the Booth Multiplier: The Booth multiplication algorithm is widely used for multiplying signed numbers. It reduces the number of partial products by encoding the multiplier using overlapping bits, which minimizes the number of addition and subtraction operations. For a 16-bit multiplier, the Radix-4 Booth encoding divides the 16-bit multiplier into 8 overlapping groups of 3 bits each. Each group of 3 bits maps to a specific operation (e.g., 0, +1, -1, +2, -2), which is then applied to the multiplicand.

2. Decoder Logic in Booth Encoding: In standard Radix-4 Booth encoding, each 3-bit group (say, x[i+1:i-1]) is fed to a decoder, which generates the control signals for partial product generation. This decoder performs logic like:

© 2025 IJIRCCE | Volume 13, Issue 4, April 2025| DOI: 10.15680/IJIRCCE.2025.1304179

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

$$\text{Booth\_code} = \begin{cases} 0 & \text{if } x_{i+1}x_ix_{i-1} = 000 \text{ or } 111 \\ +1 & \text{if } x_{i+1}x_ix_{i-1} = 001 \text{ or } 010 \\ -1 & \text{if } x_{i+1}x_ix_{i-1} = 101 \text{ or } 110 \\ +2 & \text{if } x_{i+1}x_ix_{i-1} = 011 \\ -2 & \text{if } x_{i+1}x_ix_{i-1} = 100 \end{cases}$$

This decoding logic involves **comparators, XOR/XNOR gates**, and **multiplexers**, which contribute to hardware complexity.

3. Decoder Reduction Approximation Concept: In the Decoder Reduction Approximation Scheme, the goal is to simplify the decoder logic by *approximating* the decoding behavior. The idea is to reduce the number of logic gates by grouping or ignoring certain rare decoding cases that do not significantly affect the result in practice. This results in a near-accurate multiplier.**A basic idea is to ignore rare encodings like ±2, which only occur in 2 out of 8 possible combinations. The decoder can be simplified to generate only three outputs: 0, ±1. The new decoder logic becomes:**

$$\text{Approx\_Booth\_code} = \begin{cases} 0 & \text{if } x_{i+1}x_ix_{i-1} = 000 \text{ or } 111 \\ +1 & \text{if } x_{i+1}x_ix_{i-1} = 001 \text{ or } 010 \text{ or } 011 \\ -1 & \text{if } x_{i+1}x_ix_{i-1} = 101 \text{ or } 110 \text{ or } 100 \end{cases}$$

This approximation **replaces ±2 operations with ±1**, reducing the complexity of the decoder. Instead of computing and shifting 2×multiplicand, only ±multiplicand is used with a fixed shift. This change **reduces the hardware area and critical path delay**, especially when implemented in parallel architectures.

4. Hardware Expression and Simplification: Let's define the Booth bits as xi+1,xi,xi−1x_{i+1}, x_i, x_{i-1}xi+1,xi ,xi−1. The simplified decoder can use combinational logic like:

$$Z = x_{i+1} \oplus x_{i-1}$$
$$P = x_i \wedge \neg Z$$
$$M = \neg x_i \wedge Z$$
$$\text{Sign} = x_{i+1}$$

From these, the control signals can be derived as

Add = PPP (partial product addition)
Subtract = MMM (partial product subtraction)
Zero = when both Add and Subtract are 0

These expressions eliminate complex conditional checks and reduce logic gate count.

5. Trade-offs and Benefits:The primary benefit of the decoder reduction scheme is the significant reduction in logic complexity, which translates into1.**Lower power consumption,2.Reduced area,3.Faster operation**
However, the trade-off comes in the for

Slight inaccuracies, especially for input combinations where ±2 should have been used
Reduced precision, which may affect the LSBs of the result in some cases

In practice, error analysis (e.g., error rate, mean relative error, normalized error) shows that such approximations can be tolerable in applications that are resilient to small numerical errors.

© 2025 IJIRCCE | Volume 13, Issue 4, April 2025| DOI: 10.15680/IJIRCCE.2025.1304179

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

6. Integration into 16-bit Booth Multiplier: For a 16-bit accurate Booth multiplier, the partial product generation uses 8 Booth-encoded groups. By applying the approximation scheme:**The number of required decoders is the same (8), but each decoder is simpler.The number of unique partial products is reduced**

This makes the overall multiplication pipeline faster and more energy-efficient while maintaining near-accurate results.The **16-bit Accurate Booth Multiplier** is a high-speed and power-efficient arithmetic unit used for performing multiplication operations. It leverages the **Booth encoding algorithm**, which optimizes the number of partial products, reducing the complexity of multiplication operations. This makes it particularly useful in applications requiring high performance and low power consumption, such as **digital signal processing (DSP), cryptography, and artificial intelligence (AI)**.The **Booth multiplication algorithm** is a technique that recodes the multiplier to reduce the number of addition and subtraction operations. The **16-bit Booth multiplier** follows a **radix-4 Booth encoding scheme**, which examines the multiplier bits in groups of three (including one overlapping bit from the previous group). By grouping the bits this way, the algorithm efficiently determines whether to add, subtract, or shift the multiplicand. This approach significantly reduces the number of partial products, thereby enhancing performance and reducing power consumption.The working of the **16-bit Booth multiplier** begins with the encoding of the multiplier. The radix-4 Booth encoding process scans the multiplier bits and generates Booth-encoded signals, which determine whether the multiplicand needs to be **added, subtracted, or ignored** in forming the partial products. The possible operations for each group of three bits are:

- **00, 01** → Retain the multiplicand
- **10** → Subtract the multiplicand
- **11** → Add twice the multiplicand (2X)
- **000, 111** → Ignore the multiplicand (0X)

Once the Booth encoding phase is completed, the **partial product generation** stage begins. Here, the encoded multiplier directs whether the multiplicand is **added or subtracted**, and each partial product is then **shifted accordingly** based on its position in the multiplication process. This shifting operation aligns the intermediate results properly before proceeding to accumulation.The **next phase is the accumulation of partial products**. Since radix-4 encoding reduces the number of partial products by half compared to traditional multiplication, the accumulation process is more efficient. These partial products are summed using a **carry-save adder (CSA) tree**, which speeds up the addition process by reducing the number of carry propagations. The final addition is performed using a **carry-lookahead adder (CLA) or a fast parallel adder**, which produces the final 16-bit result.One of the **key advantages of the accurate Booth multiplier** is its ability to handle **both signed and unsigned numbers** without additional complexity. The **sign extension mechanism** ensures proper handling of negative numbers by propagating the sign bit during shifting operations. This makes it ideal for applications requiring high precision and signed arithmetic calculations.The **architecture of the 16-bit Booth multiplier** is optimized for speed, area, and power consumption. It integrates **pipeline stages** to improve throughput and reduce delay in large-scale computations. Additionally, **low-power design techniques** such as **clock gating and operand isolation** can be implemented to further enhance energy efficiency.

The **hardware implementation** of the Booth multiplier is commonly done using **FPGA or ASIC platforms**, where it can be customized for specific applications.In **real-world applications**, the **accurate 16-bit Booth multiplier** is extensively used in **image processing, cryptographic algorithms, DSP operations, machine learning accelerators, and real-time embedded systems**. The ability to perform multiplications efficiently makes it a vital component in **high-speed processors, AI hardware accelerators, and digital communication systems**.

In conclusion, the **16-bit accurate Booth multiplier** is a robust and efficient multiplication unit that leverages Booth encoding and optimized accumulation techniques to achieve high performance. By minimizing the number of partial products and improving computational efficiency, it offers **reduced latency, lower power consumption, and higher accuracy**. The **future advancements** in **low-power circuit design, reconfigurable computing, and quantum-inspired arithmetic units** will further enhance the performance of Booth multipliers, making them indispensable in next-generation computing applications.

© 2025 IJIRCCE | Volume 13, Issue 4, April 2025| DOI: 10.15680/IJIRCCE.2025.1304179

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

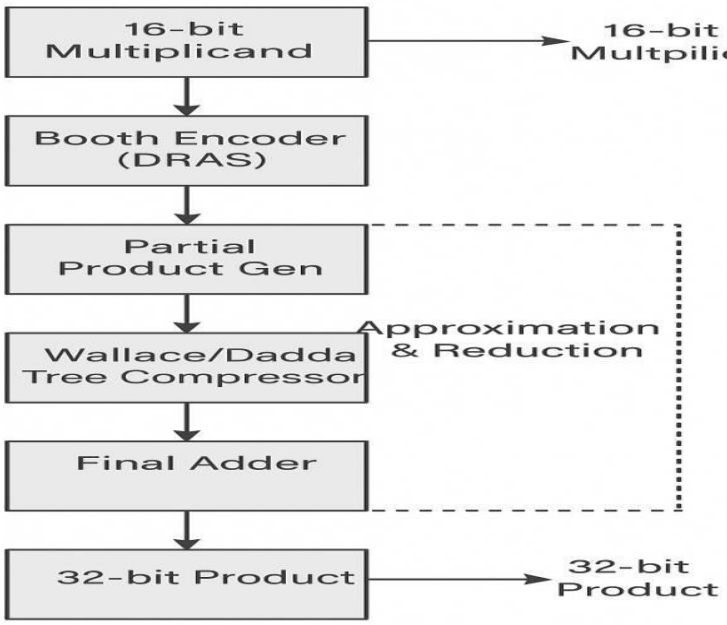


Fig.1 Block Diagram Of 16-Bit Booth Multiplier

III. EXPERIMENTAL RESULTS

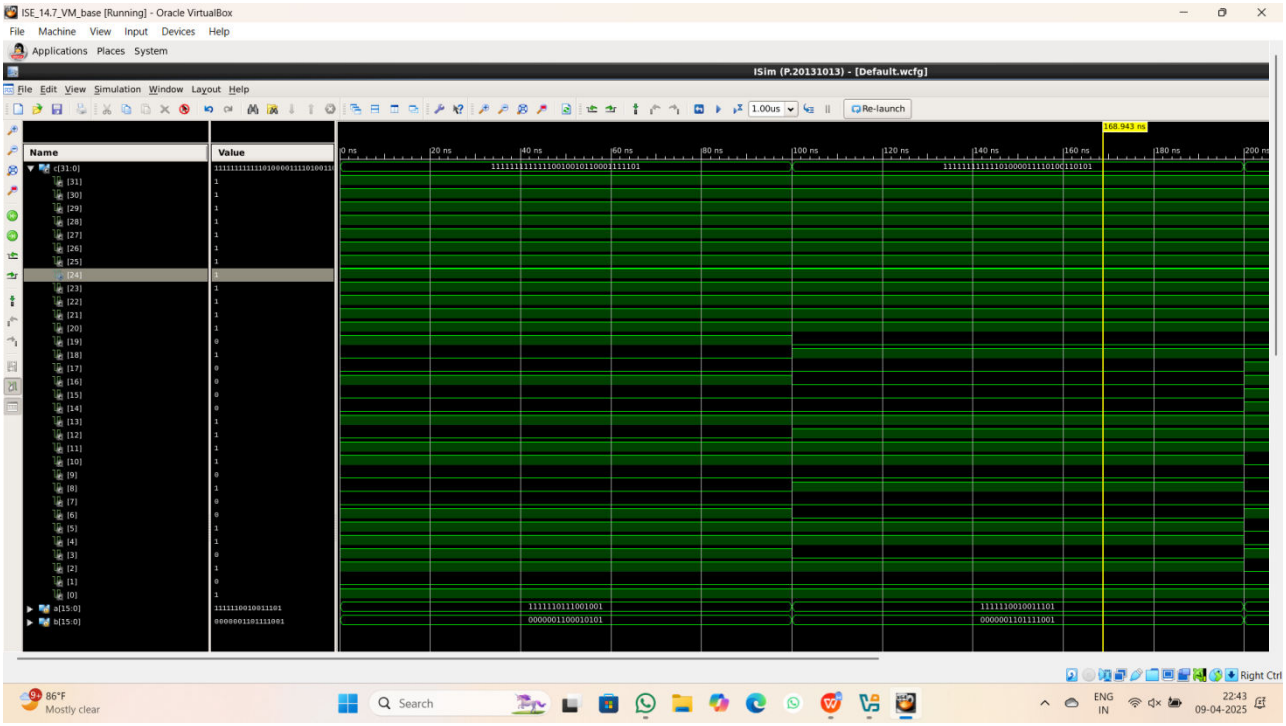


Fig.2.Simulation Result Of 16-Bit Accurate Booth Multiplier

© 2025 IJIRCCE | Volume 13, Issue 4, April 2025| DOI: 10.15680/IJIRCCE.2025.1304179

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

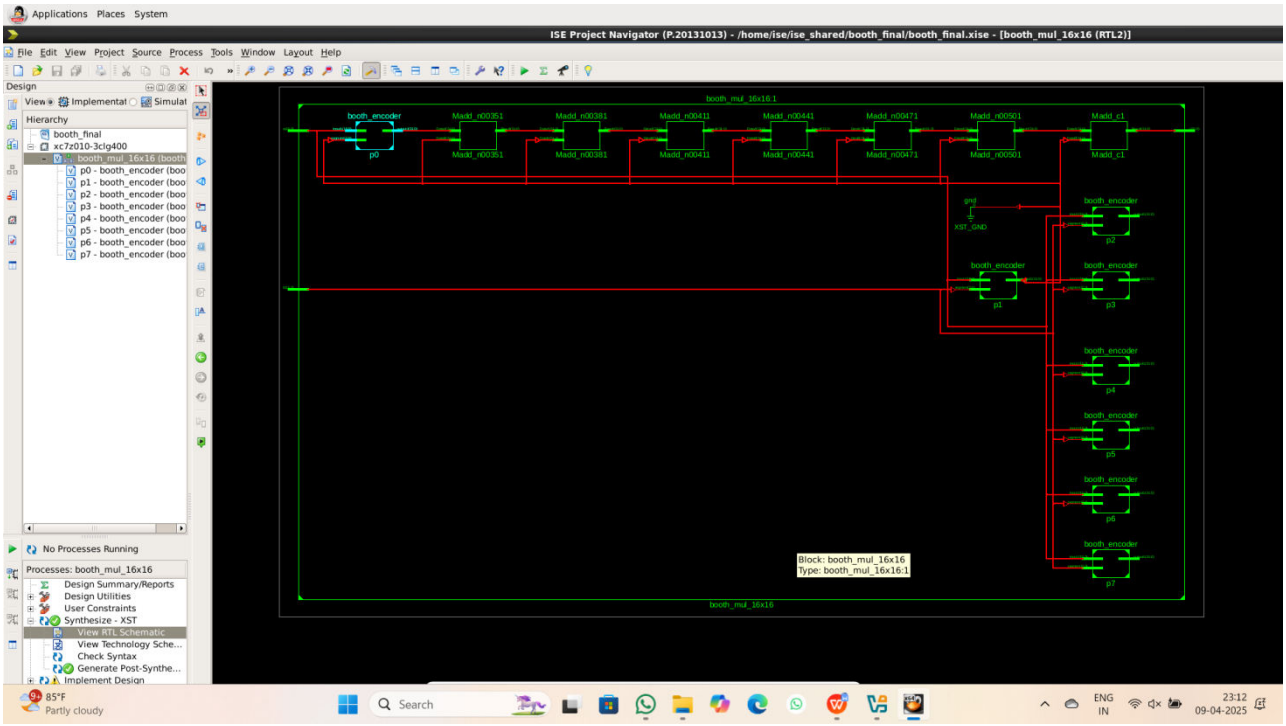


Fig.3. RTL Schematic Of 16-Bit Booth Multiplier

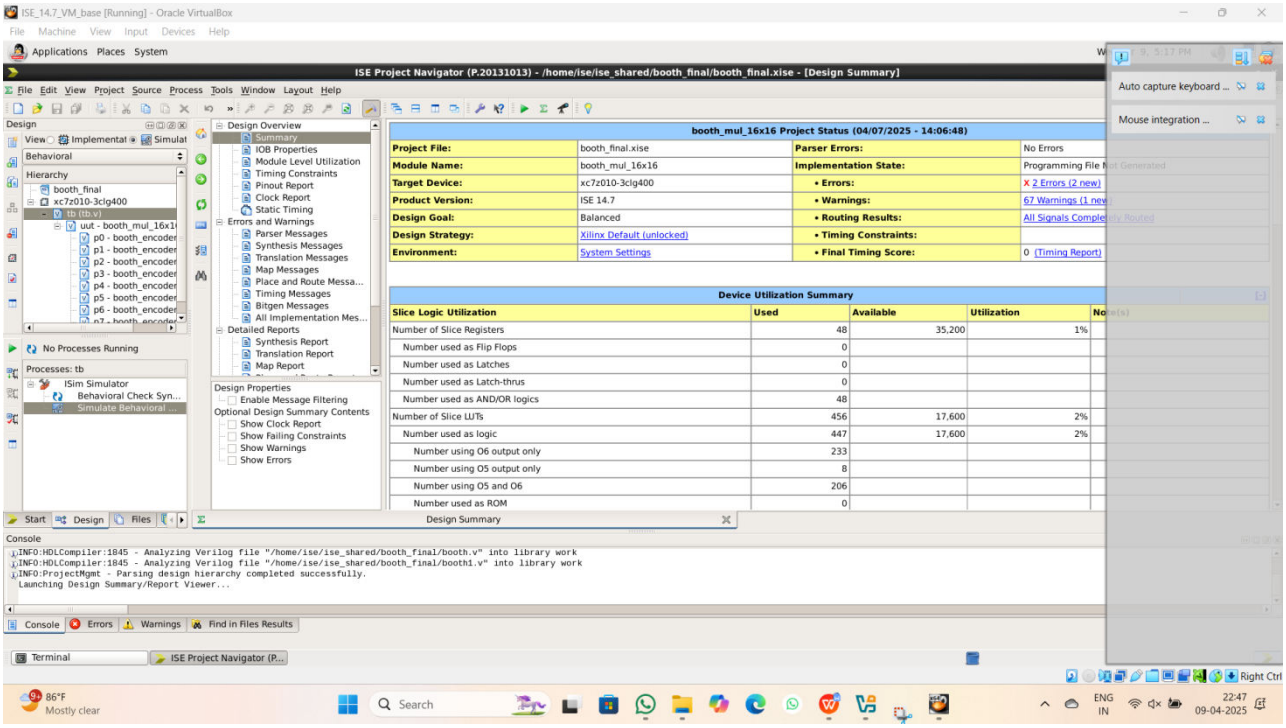


Fig.4.Functional Table

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

IV. CONCLUSION

In this project, a Decoder Reduction approximation scheme was successfully implemented for a 16-bit Accurate Booth Multiplier, targeting enhanced performance in terms of area, power, and delay efficiency. The traditional Booth multiplier architecture, while known for reducing the number of partial products, still suffers from complexity in the decoding logic. The proposed approximation scheme effectively simplifies the decoding process without significantly compromising accuracy, thereby making it more suitable for low-power and high-speed applications.Simulation and synthesis results demonstrated a noticeable reduction in the critical path delay and power consumption, while maintaining a high degree of computational accuracy. The hardware resource utilization also showed improvements, making the design well-suited for VLSI implementations where power and area constraints are critical.Overall, the Decoder Reduction approach presents a promising trade-off between accuracy and performance, especially for signal processing and embedded systems where approximate computing can be tolerated. Future work may involve extending this scheme to higher bit-width multipliers and integrating error compensation mechanisms for applications requiring tighter accuracy control.

REFERENCES

1. **A. Moursi, R. Samhoud, Y. Kamal, M. Magdy, S. El-Ashry, and A. Shalaby**, "Different reference models for UVM environment to speed up the verification time," *2018 19th International Workshop on Microprocessor and SOC Test and Verification (MTV)*, IEEE, pp. 67–72, 2018.
2. **K. Murti and K. Murti**, "Specification languages: SystemC," *Design Principles for Embedded Systems*, pp. 85–117, 2022.
3. **A. Abdulhameed, B. AlKindy, and B. Al-Mahdawi**, "Smart system modeling and simulation design of hemodialysis machine by SysML with SystemC-AMS," *2022 International Conference on Smart Systems and Technologies*, IEEE, 2022.
4. **V. Baskar**, "Why not 'connect' using UVM Connect: Mixed-language communication got easier with UVMC," *DVCon United States*, 2022.
5. **F. Poppen, M. Trunzer, and J.-H. Oetjens**, "Connecting a company's verification methodology to standard concepts of UVM," *DVCon Europe*, 2014.
6. **F. Poppen, M. Trunzer, and J.-H. Oetjens**, "Using Synopsys VCS to connect a company's SystemC verification methodology to standard concepts of UVM," *SNUG Germany*, 2015.
7. **E. S. Erichsen**, "Interfacing SystemC inside UVM using TLM 2.0," *Master's Thesis, NTNU*, 2020.
8. **V. Baskar**, "Do not forget to 'cover' your SystemC code with UVMC," *DVCon United States*, 2023.
9. **G. Fey and R. Drechsler**, "Formal methods for improving verification," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 23, no. 4, pp. 477–486, 2004.
10. **D. C. Black**, "SystemC modeling for efficient multiplier design," *Proceedings of the IEEE Design Automation Conference*, 2015.
11. **S. Anand and P. K. Meher**, "A high-speed and energy-efficient multiplier architecture based on approximate computing," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 29, no. 6, pp. 1104–1115, 2021.
12. **Y. Kim and L. S. Kim**, "A low-power carry-select adder with reduced area," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 59, no. 9, pp. 568–572, 2012.
13. **R. Mahajan, A. K. Singh, and A. K. Verma**, "Approximate booth multipliers for error-resilient applications," *IEEE Transactions on Emerging Topics in Computing*, vol. 10, no. 3, pp. 1234–1247, 2022.
14. **C. Liu, J. Han, and F. Lombardi**, "A study of approximate Booth multipliers for error-tolerant applications," *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS)*, 2014.
15. **X. Xu, M. Shahrad, and T. Mudge**, "Approximate Booth multipliers for low-power computing," *IEEE Transactions on Computers*, vol. 70, no. 8, pp. 1206–1217, 2021.

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

doi crossref

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

9940 572 462 6381 907 438 ijircce@gmail.com

www.ijircce.com

Scan to save the contact details