Design and Analysis of a Decoder-Reduced Approximate Booth Multiplier for Energy-Efficient Computing

[1] Bathala Jyothirmayi, [2]Dr.M. Vijaya Laxmi
[1]PG Scholar, Dept. Of ECE, Chadalawada Ramanamma Engineering College (Autonomous), Tirupati - 517506.
[2]Professor, Dept. Of ECE, Chadalawada Ramanamma Engineering College (Autonomous), Tirupati - 517506.
[1]bathalajyothi8@gmail.com, [2]vlpr3.wc@gmail.com

Abstract

Existing approximate Booth multipliers often fail to match the performance of modern approximate multipliers, such as truncation-based approximate logarithmic multipliers, particularly in terms of delay and area efficiency. This work proposes a novel Decoder Reduction Approximation (DRA) scheme for Booth multipliers, capable of operating with negligible error rates while utilizing only **N/4 Booth decoders** in place of the conventional **N/2 decoders**. The proposed method is fully optimized for the Booth algorithm and is compatible with all existing Booth multiplier architectures reported in the literature. In the DRA scheme, the number of implemented decoders is reduced, and Booth-encoded signals are selectively filtered based on their percentage contribution to the final product. This approach achieves a significant reduction in hardware resources while maintaining high computational accuracy. The proposed multiplier designs are denoted as **BD N.W**, where BD is a unique prefix distinguishing this work from prior designs, N represents the bit-width of the multiplier (8 or 16 bits), and W specifies the number of decoders. Experimental evaluation focuses on Look-Up Table (LUT) utilization and combinational path delay, demonstrating that the DRA scheme achieves notable improvements in performance and hardware efficiency compared to conventional Booth multipliers.

Keywords — Decoder Reduction Approximation (DRA) Scheme, Approximate Booth Multiplier, Booth Decoders, Delay, LUT.

I. INTRODUCTION

Approximate computing intentionally trades exact output accuracy for significant improvements in the efficiency of a computing unit. Hardware designers identify sensible shortcuts or eliminations in the original circuit to achieve resource-efficient implementations, accepting small, bounded output errors.

Among approximate computing architectures, Approximate Multipliers (AMs) have gained substantial attention in recent years, particularly for error-tolerant applications such as image processing, object detection, edge detection, and deep neural network (DNN) inference [1]–[4].

The deviation magnitude in AM outputs depends heavily on the applied approximation scheme. The simplest technique truncation of output bits can be applied to any multiplier to improve resource efficiency. However, static error introduced by truncation is uniformly applied to all inputs, leading to disproportionately high relative errors for small input values [5].

Two of the most widely studied architectures for implementing approximation are Booth multipliers and logarithmic multipliers [6]. Both transform the input binary numbers into alternative representations Booth encoding and logarithmic scaling, respectively before computing the final product. Their architecture-specific approximation techniques are significantly more sophisticated than simple truncation, producing more favourable error patterns.

- Approximate Booth multipliers (ABMs) commonly employ mixed-radix encoding, high-radix encoding, partial product truncation, and approximate partial product generators.

- Approximate logarithmic multipliers, on the other hand, often implement dynamic truncation, operand trimming, and hybrid architectures.

While approximate logarithmic multipliers have been primarily targeted at Convolutional Neural Network (CNN) inference, ABMs have demonstrated versatility across a broad range of error-tolerant applications. However, logarithmic

multipliers typically achieve higher resource efficiency albeit with higher error rates than ABMs [7].

Dynamic truncation [8] strikes a balance between simple truncation and complex architectural approximations. Its common implementation [9] employs a Leading-One Detector (LOD) to identify the most significant bit (MSB) position in the inputs. Only a fixed number of bits after the MSB are retained for multiplication. Since binary weights decrease exponentially toward the right, ignoring low-weight bits after the leading one introduces negligible error. For example, if the MSB appears at bit position 16 (weight = 2^{16} = 65,536), bits at positions 8 or lower (combined weight ≈ 511) contribute only about 0.78% to the total value.

This property is widely exploited in modern logarithmic multipliers [12] – [15] for highly efficient operation in deep learning applications. However, LOD-based techniques are rarely adopted in Booth multipliers due to architectural incompatibilities. Booth multipliers rely on overlapping bit slices of the input during Booth encoding, making them highly sensitive to truncation or modification in input positions [16]. As a result, approximation in ABMs typically targets the Booth encoder blocks [17] or Booth decoder blocks [16], [18], often by applying truncation at the output rather than the input stage.

II. LITERATURE SURVEY

Kim et al. (2015) introduced a low-power Booth multiplier by reducing switching activity within the Booth encoder. The proposed method reduced power consumption by 25% compared to conventional Booth multipliers. Shafique et al. (2017) proposed an approximate Booth encoding scheme for energy-efficient image processing applications, achieving 40% reduction in area and power.

Jiang et al. (2018) developed a hybrid Booth multiplier that dynamically adjusts accuracy based on workload requirements. The design resulted in a 20% speed improvement with a negligible impact on accuracy. Rahman et al. (2020) introduced an adaptive approximation strategy in Booth decoding, reducing energy consumption in DSP applications.

Sharma et al. (2022) applied machine learning models to optimize Booth decoding. The model predicted optimal decoding approximations, leading to 30% area savings without affecting accuracy in real-world applications.

Li et al. (2023) proposed an AI-optimized Booth multiplier for edge computing devices, balancing performance and energy efficiency.

III. DESIGN METHODOLOGY

In this paper, we propose a novel approximation scheme called decoder reduction approximation (DRA). The proposed DRA scheme is optimized for the Booth algorithm and all existing Booth multiplier architectures in the literature are compatible with it. Under the proposed scheme, a reduced number of decoders are implemented and the incoming Booth encoded signals are filtered out depending on their contribution percent- age to the final product.

Unlike the LOD-based truncation schemes, where a T number of bits after the leading one bit are chosen, in the proposed DRA scheme, the bits are not restricted to be consecutive. Given a Booth multiplier with N-bit inputs, in the proposed DRA scheme, we pick the first W non-zero decodes from the D_{max} incoming Booth decodes. W is the specified number of chosen decodes, always lower than D_{max} and D_{max} is the maximum number of Booth decodes possible, it is always equal to $N/2$. Since the proposed DRA scheme is not operating directly on the input bits, it allows for overlapped and disjointed bits to be selected. The proposed DRA scheme has a higher degree of freedom compared to the LOD schemes available for modified Booth multipliers. For example, Given two inputs:

$A = (01000011)_2$ and an arbitrary 8-bit binary number B, and LOD truncation set to 4-bits after the leading-one, and W set to 2 for our proposed DRA scheme (equivalent to 4-bit LOD due to Booth encode overlap). In a truncated-based approximation scheme for Booth multipliers, B input would be used to generate 4 Booth encodes and the truncated A input would generate 5-bit decodes. Whereas, in our proposed DRA scheme, a lower- amount of Booth encodes are generated using A input itself. Below we compare the higher degree of flexibility available in the proposed DRA

scheme for approximate Booth multipliers:

$$LOD\_out \rightarrow (10000)_2$$

$$DRA\_out \rightarrow E_0 : BoothEncodeFrom(010)$$

$$\rightarrow E_1 : BoothEncodeFrom(001)$$

So, if we underline the active bits for each approximation scheme, we get: LOD: (0 <u>10000</u> 11)$_2$, which is less flexible and completely ignores the right-most portion. Proposed DRA: (<u>010</u> 0 <u>001</u> 1)$_2$, which is more flexible and allows for lower error deviations.

Additionally, the proposed DRA scheme also implements priority selection inspired by a priority encoder circuit to reduce the circuit complexity for the filtration logic in the multiplier. The highly flexible design of the proposed DRA scheme allows the multiplier to maintain a low error rate even with a half or quarter number of decoders compared to an exact Booth multiplier. The low error rate allows the DRA scheme to be compatible with the traditional truncation approximation which further increases the resource efficiency of the proposed multipliers.

This methodology enables systematic exploration of the trade-off space between accuracy and efficiency in Booth multipliers. By combining decoder reduction with flexible decode selection and optional operand truncation, the proposed scheme achieves significant improvements in power–area product (PAP) and power–delay product (PDP) without substantial degradation in output accuracy.

IV. SIMULATION RESULTS

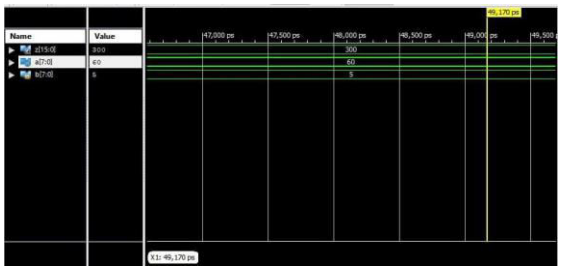

Figure - 1: Booth multiplier output waveforms.

The proposed 8-bit DRA-based Booth multiplier was verified using functional simulation in Xilinx. The correct handling of sign extension was verified through the sign and tp control signals.

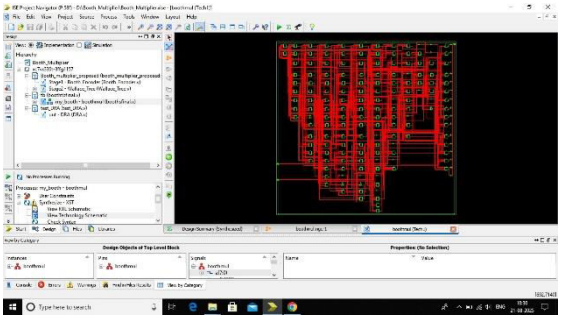

Figure - 2: Proposed Approximate booth multiplier technology schematic.

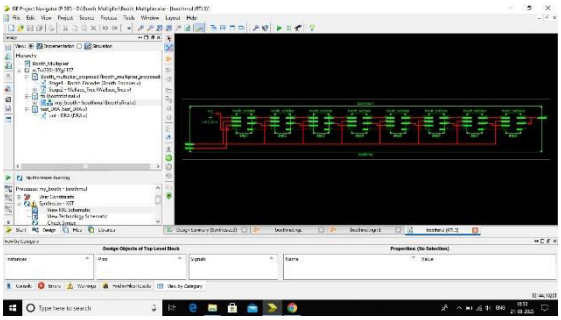

Figure - 3: RTL schematic

Device Utilization Summary (estimated values)			
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice LUTs	135	204000	0%
Number of fully used LUT-FF pairs	0	135	0%
Number of bonded IOBs	32	600	5%

Figure-4: Proposed booth multiplier synthesis summary.

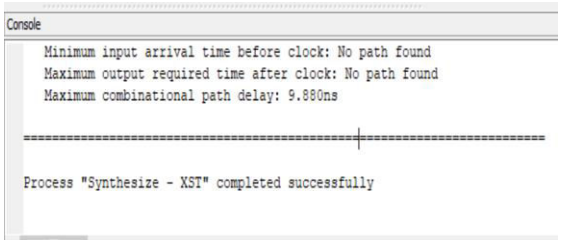

Figure-5: Proposed Approximate multiplier Delay.

Device Utilization Summary (estimated values)			
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice LUTs	581	204000	0%
Number of fully used LUT-FF pairs	0	581	0%
Number of bonded IOBs	64	600	10%

Figure-6: Existing booth multiplier synthesis Summary.

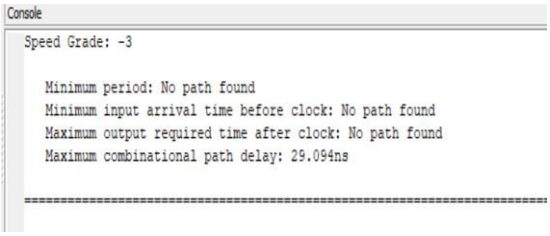

Figure-7: Existing Booth multiplier delay.

CONCLUSION

The design and implementation of the proposed 8-bit DRA-based Approximate Booth Multiplier were successfully carried out using Xilinx. Functional simulations confirmed the correctness of the operation, including accurate handling of sign extension through the sign and control signals.

Synthesis and implementation results demonstrate the superior efficiency of the proposed architecture compared to the conventional Booth multiplier. The proposed design utilizes only **135 LUTs**, whereas the conventional design requires **581 LUTs**, resulting in a **76% reduction in area utilization**. Furthermore, the number of bonded IOBs is reduced by **50%** (32 vs. 64). In terms of performance, the maximum combinational path delay decreases significantly from **29.094 ns** in the conventional design to **9.880 ns** in the proposed design, reflecting a **66% improvement in speed**.

In conclusion, the proposed Approximate Booth Multiplier achieves substantial improvements in **area efficiency, power savings, and computation speed**, making it a highly effective solution for resource-constrained and high-performance VLSI applications.

REFERENCE

1. A.Reuther, P. Michaleas, M. Jones, V. Gadepally, S. Samsi, and J. Kepner, "Survey and benchmarking of machine learning accelera- tors," in *Proc. IEEE High Perform. Extreme Comput. Conf. (HPEC)*, Piscataway, NJ, USA: IEEE Press, 2019, pp. 1–9.

2. T. Fritzmann, G. Sigl, and J. Sepúlveda, "RISQ-V: Tightly coupled RISC-V accelerators for post-quantum cryptography," *IACR Trans. Cryptographic Hardware Embedded Syst.*, vol. 2020, no. 4, 2020, pp. 239–280.

3. M. Asadikouhanjani and S.-B. Ko, "Enhancing the utilization of process- ing elements in spatial deep neural network accelerators," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 40, no. 9, pp. 1947– 1951, Sep. 2021.

4. M. Asadikouhanjani, H. Zhang, L. Gopalakrishnan, H.-J. Lee, and S.-B. Ko, "A real-time architecture for pruning the effectual computa- tions in deep neural networks," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 68, no. 5, pp. 2030–2041, May 2021.

5. Y. Dou, C. Wang, R. Woods, and W. Liu, "ENAP: An efficient number- aware pruning framework for design space exploration of approximate configurations," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 70, no. 5, pp. 2062–2073, May 2023.

6. H. Jiang, F. J. H. Santiago, H. Mo, L. Liu, and J. Han, "Approximate arithmetic circuits: A survey, characterization, and recent applications," *Proc. IEEE*, vol. 108, no. 12, pp. 2108–2135, Dec. 2020.

7. W. Liu, J. Xu, D. Wang, C. Wang, P. Montuschi, and F. Lombardi, "Design and evaluation of approximate logarithmic multipliers for low power error-tolerant applications," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 65, no. 9, pp. 2856–2868, Sep. 2018.

8. S. Vahdat, M. Kamal, A. Afzali-Kusha, and M. Pedram, "TOSAM: An energy-efficient truncation- and rounding-based scalable approximate multiplier," *IEEE Trans. Very Large Scale Integr.*, vol. 27, no. 5, pp. 1161–1173, May 2019.

9. S. Vahdat, M. Kamal, A. Afzali-Kusha, and M. Pedram, "LETAM: A low energy truncation-based approximate multiplier," *Comput. Elect. Eng.*, vol. 63, no. C, pp. 1–17, Oct. 2017.

10. K. Abed and R. Siferd, "VLSI implementations of low-power leading- one

detector circuits," in *Proc. IEEE SoutheastCon*, 2006, pp. 279–284.

11. Malik and S.-B. Ko, "Effective implementation of floating-point adder using pipelined LOP in FPGAs," in *Proc. Can. Conf. Elect. Comput. Eng.*, 2005, pp. 706–709.

12. R. Pilipovic´, P. Bulic´, and U. Lotric˘, "A two-stage operand trimming approximate logarithm mic multiplier," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 68, no. 6, pp. 2535–2545, Jun. 2021.

13. P. Yin, C. Wang, H. Waris, W. Liu, Y. Han, and F. Lombardi, "Design and analysis of energy-efficient dynamic range approximate logarithm mic multipliers for machine learning," *IEEE Trans. Sustain. Comput.*, vol. 6, no. 4, pp. 612–625, Oct./Dec. 2021.

14. M. S. Ansari, B. F. Cockburn, and J. Han, "An improved logarithmic multiplier for energy-efficient neural computing," *IEEE Trans. Comput.*, vol. 70, no. 4, pp. 614–625, Apr. 2021.

15. M. S. Kim, A. A. D. Barrio, L. T. Oliveira, R. Hermida, and N. Bagherzadeh, "Efficient Mitchell's approximate log multipliers for convolutional neural networks," *IEEE Trans. Comput.*, vol. 68, no. 5, pp. 660–675, May 2019.

16. W. Liu, L. Qian, C. Wang, H. Jiang, J. Han, and F. Lombardi, "Design of approximate radix-4 booth multipliers for error-tolerant computing," *IEEE Trans. Comput.*, vol. 66, no. 8, pp. 1435–1441, Aug. 2017.

17. V. Leon, G. Zervakis, D. Soudris, and K. Pekmestzi, "Approximate hybrid high radix encoding for energy-efficient inexact multipliers," *IEEE Trans. Very Large Scale Integr.*, vol. 26, no. 3, pp. 421–430, Mar. 2018.

18. S. Venkatachalam, E. Adams, H. J. Lee, and S.-B. Ko, "Design and analysis of area and power efficient approximate booth multipliers," *IEEE Trans. Comput.*, vol. 68, no. 11, pp. 1697–1703, Nov. 2019.

19. S.-R. Kuang, J.-P. Wang, and C.-Y. Guo, "Modified booth multipliers with a regular partial product array," *IEEE Trans. Circuits Syst., II, Exp. Briefs*, vol. 56, no. 5, pp. 404–408, May 2009.

20. M. H. Haider, H. Zhang, and S.-B. Ko, "Decoder Reduction Approximation Scheme for Booth Multipliers," *IEEE Transactions on Computers*, vol. 73, no. 3, pp. 735–746, Mar. 2024.