

Hybrid Multi-Radix Approximate Booth Multipliers

Upkar Singh
Computer Engineering Undergraduate
University of Saskatchewan
Saskatoon, Canada
uoc167@mail.usask.ca

Ogo Alege
Computer Engineering Undergraduate
University of Saskatchewan
Saskatoon, Canada
egv062@mail.usask.ca

Krunal Chavda
Computer Engineering Undergraduate
University of Saskatchewan
Saskatoon, Canada
zcx592@mail.usask.ca

Abstract: Deep Neural Networks (DNNs) excel across various applications but are constrained by high power consumption and computational demands. This study investigates approximate computing in Booth multipliers, focusing on hybrid multi-radix designs. By integrating Radix-4 and Radix-8 encodings, the proposed R488 multiplier reduces hardware area and power usage while retaining high accuracy. Variants like R8ABE1, R8ABE2, and R8SUBR4_3 offer efficient trade-offs between precision and performance. Simulations demonstrate significant improvements in delay and power consumption, paving the way for energy-efficient DNN hardware implementations.

I. INTRODUCTION

Deep Neural Networks (DNNs) have achieved remarkable success in various fields, but their computational complexity and power consumption remain significant challenges, especially for resource-constrained devices. To address these issues, approximate computing has emerged as a promising approach, aiming to optimize DNNs by intentionally sacrificing some accuracy in exchange for improved efficiency and reduced power consumption.

This literature review focuses on the application of approximate computing techniques to Booth multipliers, which are widely used in DNNs for efficient multiplication operations. We will examine various approximate Booth multiplier designs, including hybrid multi-radix approaches, and analyze their performance based on factors such as accuracy, area, power consumption, and delay.

II. APPLIED APPROXIMATE TECHNIQUE

This research presents a hybrid Radix-4/8 Booth (R488) multiplier using approximate encoding to improve performance and power efficiency. Combining the efficiency of 3-bit Radix-4 encoding with the reduced partial products of 4-bit Radix-8 encoding, the design addresses Radix-8's complexity. Specifically, an approximate encoding scheme replaces the costly "3" digit in Radix-8 with "2" or "4", trading minimal accuracy for significant hardware and power reductions.

$d_{i+1}d_id_{i-1}$	PP_i
000	0
001	+1 x Multiplicand
010	+1 x Multiplicand
011	+2 x Multiplicand
100	-2 x Multiplicand
101	-1 x Multiplicand
110	-1 x Multiplicand
111	0

This table represents the Radix-4 Booth encoders.

The "R488" designation reflects the ordering of our hybrid design. In our 8-bit multiplier, it shows that the 3 most significant bits are encoded with a Radix-4 and the remaining 5 are encoded with 2 Radix-8 Booth encoders. These Radix-8 booth encoders generate the approximate partial products on less significant bits to ensure that the approximated values do not greatly deviate from the exact results.

Radix-8 encoding, while offering a reduced number of partial products, introduces complexity due to the handling of the "3" digit. Exact multiplication by 3 requires additional hardware and increases the critical path delay, leading to higher area, and power consumption.

The table compares partial product (PP) generation in different encoding techniques. The first column represents the Exact Radix-8 Encoder, which provides precise encoding by evaluating all combinations of input bits ($d_{i+2}d_{i+1}d_id_{i-1}$). The middle two columns, cited from [1], present approximate encoders R8ABE1 and R8ABE2, which simplify the encoding by approximating higher magnitude values (e.g., replacing $\pm 3\times$ with $\pm 2\times$ or $\pm 4\times$) to reduce complexity. The last column demonstrates the proposed design, where a Radix-4 Encoder is used as an approximate Radix-8 encoder, further reducing complexity and power by limiting high-magnitude partial products to simpler approximations (e.g., mapping $\pm 3\times$ to $\pm 2\times$) among other values.

$d_{i+2}d_{i+1}d_id_{i-1}$	Exact Radix-8 PP_i	R8ABE1 PP_i	R8ABE2 PP_i	Radix-4 as Radix-8 PP_i
0000	0	0	0	0
0001	+1 x Multiplicand	+1 x Multiplicand	+1 x Multiplicand	+1 x Multiplicand
0010	+1 x Multiplicand	+1 x Multiplicand	+1 x Multiplicand	0
0011	+2 x Multiplicand	+2 x Multiplicand	+2 x Multiplicand	+1 x Multiplicand
0100	+2 x Multiplicand	+2 x Multiplicand	+2 x Multiplicand	+1 x Multiplicand
0101	+3 x Multiplicand	+2 x Multiplicand	+2 x Multiplicand	+2 x Multiplicand
0110	+3 x Multiplicand	+2 x Multiplicand	+4 x Multiplicand	+1 x Multiplicand
0111	+4 x Multiplicand	+4 x Multiplicand	+4 x Multiplicand	+2 x Multiplicand
1000	-4 x Multiplicand	-4 x Multiplicand	-4 x Multiplicand	-2 x Multiplicand
1001	-3 x Multiplicand	-2 x Multiplicand	-4 x Multiplicand	-1 x Multiplicand
1010	-3 x Multiplicand	-2 x Multiplicand	-2 x Multiplicand	-2 x Multiplicand
1011	-2 x Multiplicand	-2 x Multiplicand	-2 x Multiplicand	-1 x Multiplicand
1100	-2 x Multiplicand	-2 x Multiplicand	-2 x Multiplicand	-1 x Multiplicand
1101	-1 x Multiplicand	-1 x Multiplicand	-1 x Multiplicand	0
1110	-1 x Multiplicand	-1 x Multiplicand	-1 x Multiplicand	-1 x Multiplicand

This table represents the Approximate Radix-8 Booth encoders alongside our Approximate Radix-8 encoding tables

All the approximate R488 8-bit multiplier designs we implemented combine different approximate Radix-8 encoders for the lower bits approximations. The values in the encoding tables are carefully designed to balance the outputs of the approximations. While the designs in [1] aimed to maintain a maximum error distance of 1, our approach to substitute a Radix-8 with a Radix-4 encoder prioritizes computational simplicity and efficiency, allowing larger deviations to achieve better area and power trade-offs. In this case, the Radix-4 encoder ignores the 2 LSB for generating its approximate result.

III. ACCURACY ANALYSIS

The accuracy and efficiency of the multipliers were assessed using various configurations, relative to mult_R4444, based on their performance on a model trained on the MNIST dataset to generate inference outputs. The exact multiplier mult_R488 maintained the same accuracy (96%) but required 22.17% more area (673.92 vs. 551.52) and consumed 19.29% more power (0.1082 mW vs. 9.07e-02 mW). Its delay, however, was slightly reduced by 4.6% (0.83 vs. 0.87).

Among the approximate designs, mult_R488_R8ABE1 achieved a higher accuracy (97%, +1% over R4444) while requiring only 5.45% more area (495.36 vs. 551.52) and consuming 14.1% less power (7.79e-02 mW). Its delay was 10.3% lower than R4444 (0.78 vs. 0.87), making it a balanced design for accuracy and efficiency.

mult_R488_R8ABE2 maintained a moderate increase in area (512.64, +9.92%) but traded accuracy (93%, -3% relative to R4444). Power consumption was also 11.91% lower than R4444 (7.99e-02 mW), and its delay improved by 9.2% (0.79 vs. 0.87).

The Sub-R4 designs, such as mult_R488_R8SUBR4_2, exhibited the most significant area and power savings. It required 27.86% less area (397.8 vs. 551.52) and consumed 28.41% less power (6.49e-02 mW). However, this came at the cost of reduced accuracy (84%, -12%) and a higher NMED of 0.03125. Conversely, mult_R488_R8SUBR4_3 balanced efficiency and accuracy, achieving 97% accuracy (+1%), 20.02% lower area (441.36 vs. 551.52), and 25.46% lower power (6.76e-02 mW). Its delay was reduced by 17.2% (0.72 vs. 0.87), making it a highly efficient design.

These comparisons highlight the trade-offs between accuracy and resource efficiency, with mult_R488_R8ABE1 and mult_R488_R8SUBR4_3 standing out as strong candidates for accuracy-focused and efficiency-optimized designs, respectively, especially in scenarios involving MNIST inference tasks.

IV. HARDWARE IMPLEMENTATION

Our hardware design for the 8-bit multipliers is structured with a modular approach, featuring a top-level module that accepts two 8-bit inputs (multiplicand and multiplier) and outputs a 16-bit product. At the core of the design is a multiplexer, implemented as an always @* block with a case statement. This multiplexer permutes the multiplicand based on the Booth grouping of the multiplier, effectively generating partial products for further processing.

The Partial Product Generating Booth Encoders, instantiated in the top-level module, take the permuted multiplicand and the grouped multiplier bits as inputs. These encoders compute the partial products using a combination of exact or approximate logic, depending on the multiplier configuration. The partial products are then summed in a tree structure of adders within the top-level module to produce the final product.

V. RESULTS/ SIMULATION RESULTS

	EXACT		APPROXIMATE			
	mult_R4444	mult_R488	mult_R488_R8ABE1	mult_R488_R8ABE2	mult_R488_R8SUBR4_2	mult_R488_R8SUBR4_3
Area	551.52	673.92	495.36	512.64	397.8	441.36
Power	9.07e-02mW	0.1082mW	7.79e-02mW	7.99e-02mW	6.49e-02mW	6.76e-02mW
Delay	0.87	0.83	0.78	0.79	0.7	0.72
Accuracy	96	96	97	93	84	97
NMED	0	0	0.008842	0.009057	0.03125	0.010742

Table depicting the numerical value comparison of different algorithms

The table compares the performance of different designs for hybrid multi-radix approximate Booth multipliers based on four key metrics: accuracy, area reduction, power reduction, and delay improvement.

- Accuracy (%) : Indicates how close the design's output is to the exact result. While exact designs achieve consistent 96 accuracy, approximate designs like R8SUBR4_3 outperform with 97 accuracies by optimizing resource usage.
- Area Reduction (%) : Represents the decrease in chip area usage compared to the exact design. Approximate designs reduce area significantly, with R8SUBR4_3 achieving the highest reduction,

making it efficient for space-constrained hardware implementations.

- Power Reduction (%) : Shows the percentage reduction in power consumption. R8SUBR4_3 leads with a greater degree of reduction, demonstrating its energy efficiency, which is critical for resource-constrained applications.
- Delay Improvement (%) : Highlights the improvement in processing speed. R8SUBR4_3 again performs the best, enhancing the system's overall performance.

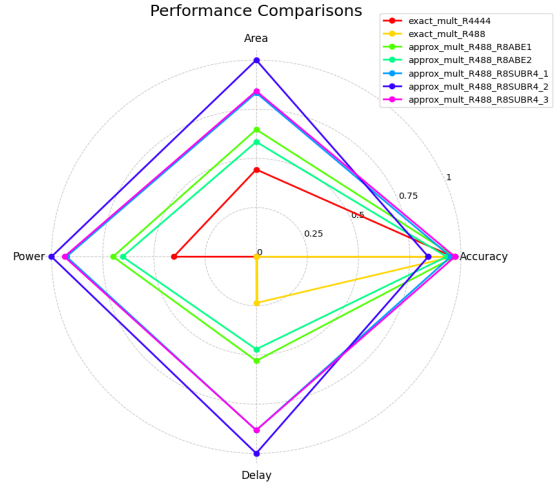


Figure depicting the differences in usage stats of various algorithms

The graph highlights in a way that the further away from the origin we are for a certain parameter, the better usability and results it portrays. A small explanation being that the further we are away from the origin, the less is the Power used, the less is the Area under usage and less is the Delay in the process alongside showcasing higher Accuracy.

CONCLUSION

This study delves into the optimization of hybrid multi-radix Booth multipliers through the implementation of approximate computing techniques. By combining Radix-4 and Radix-8 encodings, the proposed R488 design demonstrates a balanced approach, achieving significant improvements in area and power efficiency while maintaining high accuracy. The research explores both exact and approximate multiplier variants, with approximate designs such as R8SUBR4_3 offering practical trade-offs for real-world applications.

Through systematic hardware implementations, the study examines the trade-offs between precision and performance, providing insights into the impact of reduced encoding complexity on accuracy and computational efficiency. Results indicate that approximate multipliers not only reduce hardware requirements but also enhance performance metrics such as delay and power consumption.

REFERENCES

- [1] H. Waris, C. Wang and W. Liu, "Hybrid Low Radix Encoding-Based Approximate Booth Multipliers," in IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 67, no. 12, pp. 3367-3371, Dec. 2020, doi: 10.1109/TCSII.2020.2975094.