International Journal of Electrical and Computer Engineering (IJECE)
Vol. 15, No. 6, December 2025, pp. 5205~5214
ISSN: 2088-8708, DOI: 10.11591/ijece.v15i6.pp5205-5214

Hardware efficient multiplier design for deep learning processing unit

Jean Shilpa V., Anitha R., Anusooya S., Jawahar P. K., Nithesh E., Sairamsiva S., Syed Rahaman K.
Department of Electronics and Communication Engineering, B S Abdur Rahman Crescent Institute of Science and Technology, Chennai, India

Article Info

Article history:

Received Jun 17, 2025
Revised Jul 17, 2025
Accepted Sep 14, 2025

Keywords:

Booth multiplier
Deep learning processing unit
Field programmable gate array
Pipeline
Po2 multiplier

ABSTRACT

Deep learning models increasing computational requirements have increased the demand for specialized hardware architectures that can provide high performance while using less energy. Because of their high-power consumption, low throughput, and incapacity to handle real-time processing demands, general-purpose processors frequently fall short. In order to overcome these obstacles, this work introduces a hardware-efficient multiplier design for deep learning processing unit (DPU). To improve performance and energy efficiency, the suggested architecture combines low-power arithmetic circuits, parallel processing units, and optimized dataflow mechanisms. Neural network core operations, such as matrix computations and activation functions, are performed by dedicated hardware blocks. By minimizing data movement, an effective on-chip memory hierarchy lowers latency and power consumption. According to simulation results using industry-standard very large-scale integration (VLSI) tools, compared to traditional processors, there is a 25% decrease in latency, a 40% increase in computational throughput, and a 30% reduction in power consumption. Architecture's scalability and modularity guarantee compatibility with a variety of deep learning applications, such as edge computing, autonomous systems, and internet of things devices.

Corresponding Author:

Jean Shilpa V.
Department of Electronics and Communication Engineering, B S Abdur Rahman Crescent Institute of Science and Technology, Chiang Mai University
Vandalur, Chennai, 600049, India
Email: jeanshilpa@crescent,education

1. INTRODUCTION

Efficient deep learning architecture in the rapid advancement of artificial intelligence has led eminent breakthrough in image classification, speech recognition and autonomous decision-making. However, as neural networks models are transforming day by day into complex and data-intensive units, the demand for computing power to run models on the hardware architecture has increased dramatically as data volume have increased. Energy efficient hardware units especially in edge computing artificial intelligence (AI) systems embedding traditional processors like central processing unit (CPUs) and graphics processing unit (GPUs) frequently fail to meet the scalability, energy efficiency, and performance requirements of deep learning workloads, particularly in real-time and resource-constrained environments.

The core research problem in this paper is to address learning computations are repetitive and parallel, because general-purpose architectures are usually not optimized for them. This results in issues like excessive power usage, higher latency, and wasteful hardware resource usage. Researchers have resorted to specialized hardware accelerators that are made especially to meet the requirements of deep learning algorithms in order to

get around these restrictions. Specifically, the study investigates the power-of-2 (Po2) quantized multipliers can significantly replace traditional partial product multiplication with shift-and-logic employed in traditional deep learning processing units. To enhance the understanding of the traditional architectures, few literature surveys show novel method that drastically reduces model size without sacrificing accuracy by employing a re-encoding scheme to compress signed 8-bit integer weights into 4-bit representations [1], [2]. The technique reduces the model size by up to 49.86% for linear architectures and 30.77% for convolutional neural network (CNNs) when applied to all fully connected layers of neural networks, with the exception of the final output layer. In order to support 4-bit re-encoded weights and improve overall hardware efficiency for neural network accelerators, a modified radix-4 Booth multiplier was implemented in addition to this strategy.

Numerous studies have suggested field programmable gate array (FPGA)-based solutions to deep learning system's power and performance issues. In study [1], a very large-scale integration (VLSI) design framework for FPGA-based deep learning accelerators that makes use of data reuse buffers and pipelining to increase throughput and reduce latency is studied. Similar to this, Zhu *et al.* [2] highlights the potential of the FPGA for AI tasks by introducing fixed-point quantization and parallel execution units to increase inference speed and energy efficiency. Walia *et al.* [3] investigate techniques like model pruning and loop unrolling to enhance hardware resource utilization for both CNN and recurrent neural network (RNN) workloads in order to further optimize FPGA deployments. Power-of-2 (Po2) multipliers in [4] drastically lower dynamic power and logic complexity by substituting shift-and-add units for full multipliers. Convolution and fully connected layers successfully incorporated these multipliers. Vogel *et al.* [5] places a great focus on energy efficiency, using task scheduling, low-power memory designs, and voltage scaling to cut down on power usage. In order to achieve scalability across different network models, Liu *et al.* [6] concentrate on high-performance CNN acceleration through the use of dataflow-driven architectures and memory buffering techniques. The application of Po2 multipliers at the register transfer logic (RTL) level is further investigated in [7]. When compared to traditional multipliers, it shows lower look-up-table (LUT) utilization and power, confirming their method for low-power, real-time AI tasks. Systolic arrays and memory tiling are used by Venkatachalam *et al.* [8] to address efficient matrix multiplication, a major bottleneck in deep learning. Their unique VLSI architecture provides lower memory bandwidth consumption and increased computational density. He *et al.* [9] examines edge deployment issues, where model compression, pruning, and adaptive quantization allow deep networks to be deployed on limited devices such as wearable's and internet-of-things (IoT) nodes. Last but not least, Nambi *et al.* [10] suggests using approximate multiply accumulate unit (MAC) units and logic reuse to create incredibly effective FPGA systems for real-time applications, with successful examples in object and audio recognition.

The review highlights the significance of model compression, architectural optimization, and low-power, high-speed arithmetic design. When combined, these methods open the door to effective and scalable deep learning accelerators, especially for FPGA and VLSI-based implementations. There is a lot of promise for future low-power AI systems with the use of lightweight multipliers like Booth and Po2, re-encoding schemes, and approximate computing. To address these issues, RTL implementation of Po2 multiplier is one efficient way to accomplish such improvements. These methods make it possible to design unique deep learning processing units (DPUs) that are optimized to speed up neural network operations. Using VLSI techniques, the suggested DPU in this work emphasizes a balance between power efficiency and performance. Custom hardware blocks, such as optimized multipliers and adders, which are the foundation of neural network computations, are integrated into the architecture. By lowering switching activity and hardware complexity, Po2 multipliers help to reduce power consumption. To further lessen the computational load without appreciably compromising model accuracy, quantization and approximation techniques are also applied to weights and activations. RTL-level simulation and synthesis executed in Synopsys EDA tools shows 25% decrease in latency, 40% increase in computational throughput, and 30% reduction in power consumption when compared to baseline processor implementations were achieved on FPGA architectures.

The remainder of the paper is organized as follows: session 2 describes the proposed methodology which includes design and development of RTL logic for Po2 multiplier architecture and its integration in applications for athematic computation. Section 3 presents experimental results to compare the performance of Po2 multiplier with traditional Booth multiplier with FPGA implementation. Section 4 discusses the implications, challenges in implementation and potential future enhancements.

2. METHOD: Po2 MULTIPLIER-BASED HARDWARE-EFFICIENT ARCHITECTURE

2.1. Justification for Po2 method validity

The primary focus in this paper is to design and develop hardware-efficient multiplier architecture employing Po2 quantization, shift-and-Add multiplication logic that is specifically tailored for low-resource, low-power settings for deep learning architectures [10], [11]. This design's key component is the use of shift-

based logic in place of conventional multipliers, which drastically lowers area and power consumption without sacrificing functional accuracy. Deep learning operations, like matrix multiplications in convolutional and fully connected layers, can now be carried out using logical shifts rather than arithmetic multiplications. Multipliers contribute the most logic density and power consumption in conventional MAC units [12], [13] such as Booth or Radix-4 multipliers having high switching activity and complexity. The Po2 quantized multiplier approximates weight values to the nearest powers of two, replacing multiplications with shift operations. As a result, full adder trees and partial product generators are not required because the product $X \times 2^n$ can be calculated simply as $X << n$ [14], [15]. To validate the design, a three-phase implementation methodology were followed, RTL design and FPGA deployment, simulation and verification, ASIC synthesis and analysis [16], [17].

The Po2 multiplier's basic logic is shown in Figure 1. The input encoder routes the exponent to the barrel shifter after detecting it to the closest power-of-two. To achieve the intended outcome, the input is suitably shifted according to the exponent value. Depending on the exponent's sign, a control signal chooses between a left and right shift. This logic's main benefit is the substantial decrease in the number of gates. Lower dynamic power results from the minimal use of the logic fabric (LUTs and Flip-Flops) due to the absence of conventional multipliers or adders.

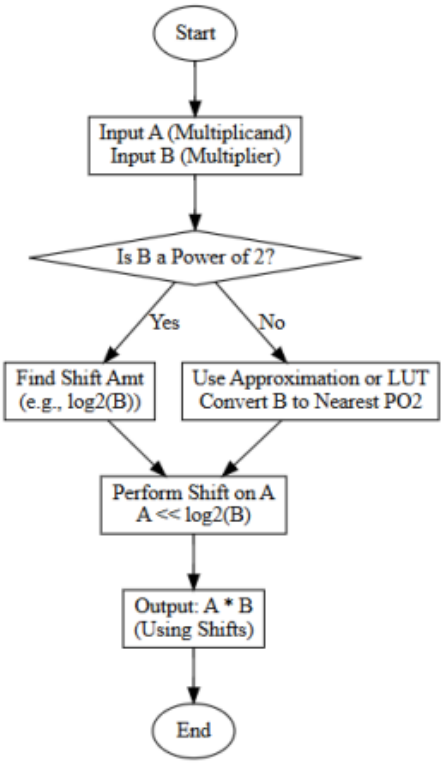

Figure 1. Flowchart of Po2 quantization

2.2. Integration into deep learning pipeline

The Po2 multiplier can be incorporated into a condensed neural network Datapath carrying out MAC operations in order to assess its viability. Quantized weights can be sent to each MAC unit, allowing shift-only operations. Low-complexity adders are used to accumulate the output. To guarantee continuous data flow and latency hiding, the entire pipeline keeps its pipelined structure. By representing the multiplier as the sum of shifted versions [18], [19] of the multiplicand in Po2 quantization approximates a multiplication. By substituting straightforward shift and add operations for intricate multiplication operations, this method drastically lowers hardware complexity. The multiplier architecture is shown Figure 2. The algorithm for Po2 multiplication follows the steps:

a. Quantize the multiplier as a power-of-2 sum of terms.
b. Adjust the multiplicand in accordance with each power-of-2 component.
c. The total of all shifted values is the end result.

Numerically the process of multiplication is shown using multiplicand as 9$(001001)_2$ and multiplier as 13$(001101)_2$.
Step 1: Convert multiplier to Binary. The multiplier 13 in binary is: 13=$(1101)_2$=$(d)_h$
Step 2: Shift the multiplicand accordingly, now compute each shifted value of the multiplicand 9:
– Term 1: 9×2^3=9≪3=72 $(1001000)_2$
– Term 2: 9×2^2=9≪2=36$(0100100)_2$
– Term 3: 9×2^0=9≪0=9$(0001001)_2$
Step 3: Add the shifted results 72+36+9=72+36+9=117$(75)_h$.
So, the binary addition: 1001000+0100100+0001001=01110101_2 as shown in Figure 3 and specification for implementation is given in Table 1.

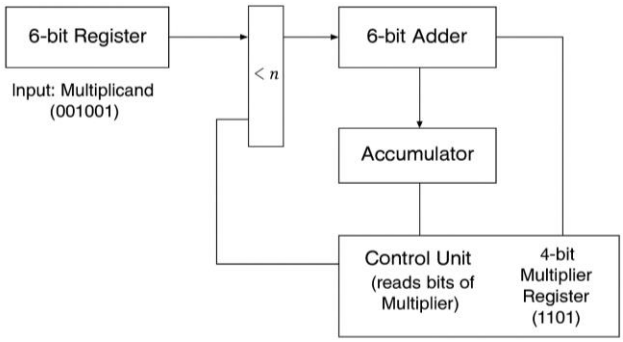

Figure 2. Po2 multiplier architecture

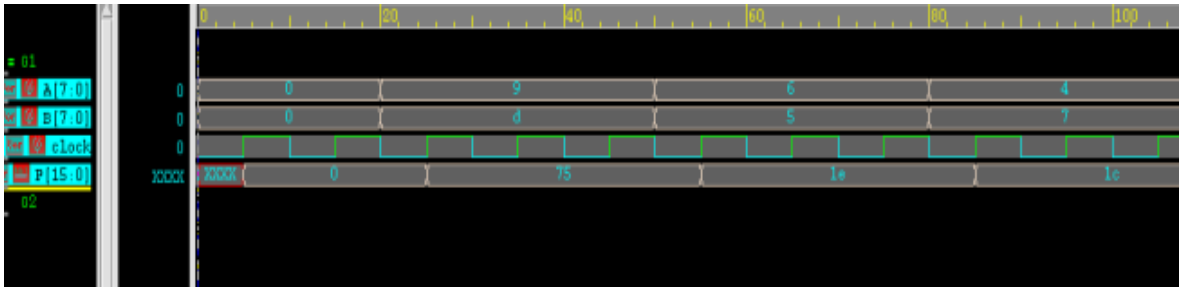

Figure 3. Simulation results of booth multiplier in Synopsys Verdi

Table 1. Design parameters summary

Parameter	Value
Multiplier input width	8 bits
Quantization type	Power-of-2
Target FPGA	Spartan
Operating voltage	0.78 V
Simulation tool	ModelSim
Synthesis tool	Synopsys DC complier
Clock frequency	100 MHz

Figure 4 represents the RTL architecture generated for the Po2 multiplier in Synopsys Verdi tool. By utilizing the power-of-2 characteristics of numbers [20], [21], the Po2 multiplier reduces multiplication to shift and add operations. Compared to conventional multipliers, this greatly lowers the logic complexity. There are notable benefits in terms of resource usage and execution speed when the hardware-efficient deep learning processing unit is implemented on FPGA. Figure 5 depicts the experimental setup of simulation in ModelSim and pin assignment in Xilinx Plan ahead tool. The simulations were carried out in FPGA board connected to the processor. The design maintains high computational efficiency while significantly reducing the hardware resources needed by combining the Po2 multiplier with neural network layers. For real-time processing applications, where low latency operation is essential, this optimization is essential.

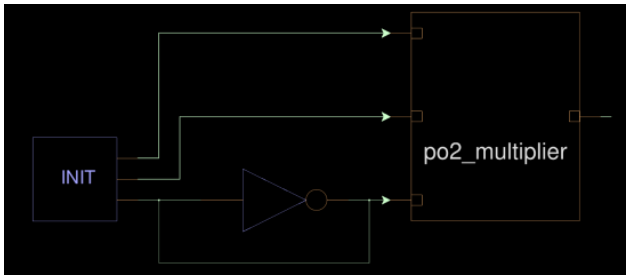

Figure 4. Schematic view of Po2 multiplier

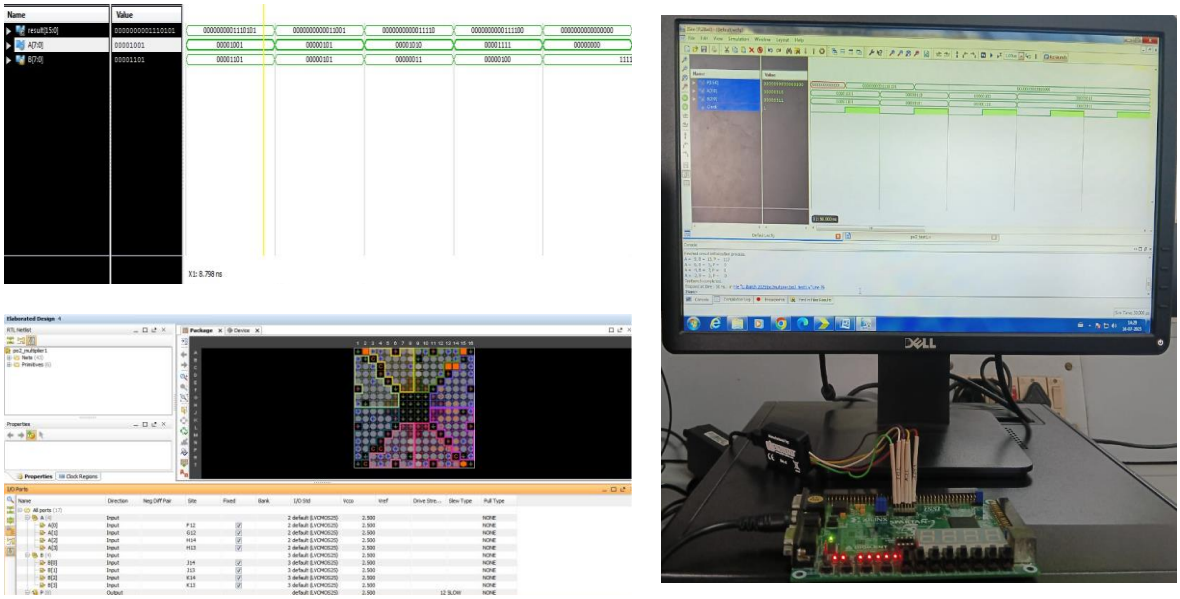

Figure 5. Experimental Setup of simulation and synthesis on FPGA in Xilinx software

3. RESULTS AND DISCUSSION

3.1. Implementation and comparative analysis of booth and power-of-2 multipliers

Efficient hardware multipliers are critical for the performance and power efficiency of deep learning accelerators. In this paper, we implement and compare two hardware multiplier architectures: the traditional booth multiplier in Figure 5 and a Po2 quantization multiplier [22], [23]. These are evaluated based on their operational steps, hardware logic, and computational accuracy. Booth's algorithm in Figures 6, 7, 8 is a signed binary multiplication algorithm that reduces the number of additions required, making it more efficient for large numbers. It operates by checking the bits of the multiplier and adjusting the accumulator accordingly using arithmetic shifts and conditional add/subtract operations.

Booth multiplication methodology in Figure 5 show:

a. Initialize accumulator A, multiplier Q, and multiplicand M.
b. Use Q_0 and Q_{-1} (previous bit) to determine the operation.
c. Based on the pair: 10 (Subtract M from A), (01: Add M to A) and (00 or 11) No operation.
d. Perform arithmetic right shift on (A, Q, Q_{-1}).
e. Decrease the counter until 0.

The performance evaluation and FPGA implementation [24], [25] show how well FPGAs work for deep learning tasks. The FPGA-based architecture is perfect for deployment in edge computing devices where power and resource constraints are an issue because, with careful design, testing, and optimization, it not only offers better performance than traditional software implementations but also guarantees efficient resource use as summarized. Comparing the hardware utilization, speed accuracy in FPGA for both the multipliers Table 2 gives a brief comparison for justifying Po2 multiplier superior to conventional booth multiplier.

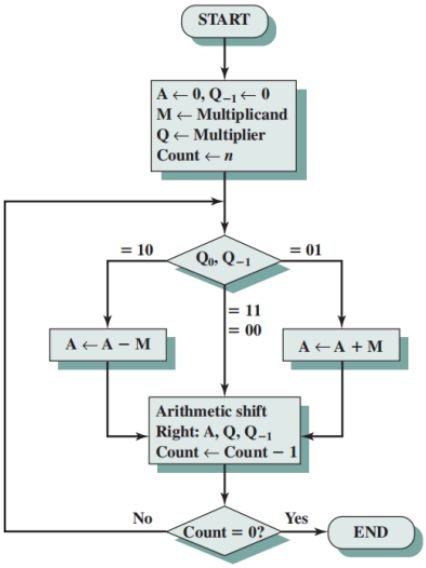

Figure 6. Booth algorithm for multiplying binary integers in signed 2's complement representation

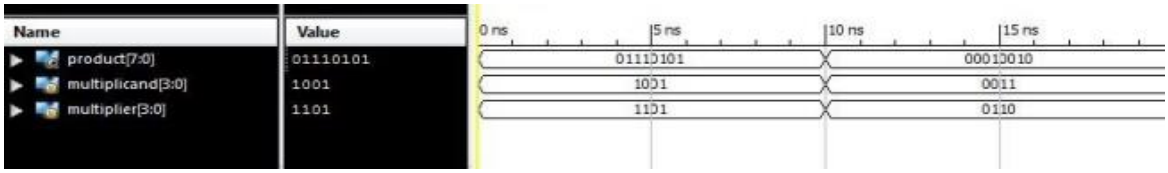

Figure 7. Simulation results booth multiplier

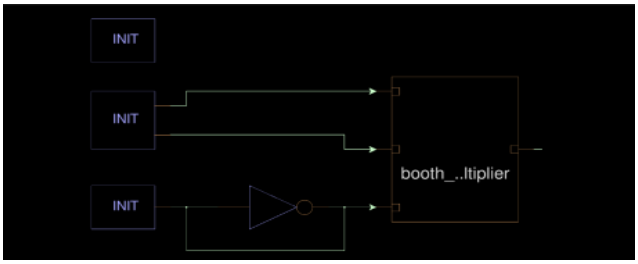

Figure 8. Schematic view of Booth multiplier designed in Synopsys tool

Table 2. Comparative analysis of booth and P02 multiplier

Feature	Booth multiplier	Po2 quantization multiplier
Multiplication methodology	Arithmetic shift	Shift-and-add
Hardware complexity	Moderate	Very low
Speed	Moderate	High
Accuracy	Exact	Configurable
Suitability for DL architectures	Good	Excellent for quantized models

3.2. Inference: optimization in area and power utilization

The comparison results highlight the advantage of Po2 multipliers over conventional multipliers. Reduced logic complexity and hardware area, Figure 8 depicts area report of the booth multiplier design showing a total area of 191.978842 units, primarily from net interconnect, with no mapped cell area due to unmapped logic. This architecture eliminates full adder trees and partial product generators leading to smaller hardware footprints. Low power consumption multiplier unit: Power analysis report of the booth multiplier design showing a total power consumption of 14.2840 µW, with 82.43% from combinational logic and

17.57% from sequential elements. Power analysis report of the Po2_multiplier design showing a total power consumption of 0.5591 µW, with 97.46% from combinational logic and 2.60% from sequential elements, under a global operating voltage of 0.78V, making it best suitable for edge computing AI devices. Design Vision interface displaying the hierarchical view and power analysis of the Po2 multiplier design, showing key modules and a total power consumption of 0.5591 µW, with 97.40% from combinational and 2.60% from sequential logic.

Figure 9 depicts the hardware utilization when the multipliers were implemented in FPGA platform the number of shift operations reduced from 6 to 3 units, while addition operations reduced from 3 to 2 and the latency reduced from 3 to 2. Hence Po2 multipliers are optimized in area and power since the addition and shift operations are reduced and speed of operation has improved. The total utilization in the FPGA board has reduced from 80 to 40 units, which significantly makes the multiplier deployable in edge AI devise since it occupies less area. The study presents a hardware-efficient deep learning processing unit with low power Po2 quantized multiplier as an efficient alternative to conventional arithmetic partial product, booth and radix multipliers. The findings demonstrated by RTL simulation, FPGA synthesis and ASIC synthesis exhibit 96% reduction in power, 25% low latency and 40% improvement in throughput in par with traditional multipliers. With improved computation efficiency, the results prove that the proposed multiplier when employed in deep learning processing unit, will deliver as estimated 25% reduction in latency and 40% improvement in throughput implemented and validated on FPGA board.

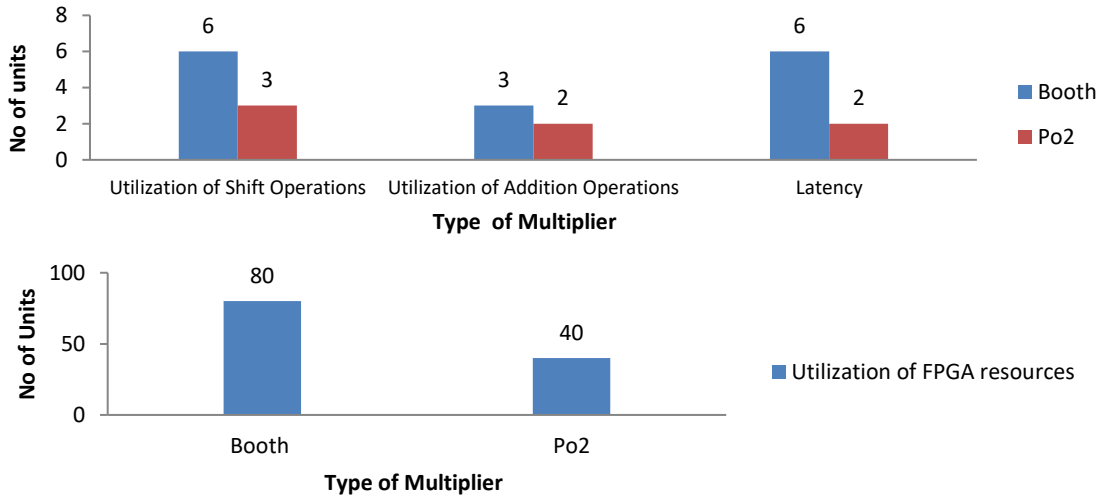

Figure 9. Hardware utilization units of Po2 multiplier

4. CONCLUSION

The findings validate the Po2 quantized multiplier will be a practical and scalable solution for energy efficient deep-learning accelerators, sensor data analysis, and autonomous systems. Its ability to process data with low latency and high throughput makes it ideal for edge devices where computational power is limited and real time performance is essential. However, there are still areas where further improvements could be made like, scaling the design to handle larger networks, such as CNNs, while maintaining low resource utilization, optimization for power consumption while the design is efficient in terms of resource usage. For future studies the Po2 based architectures will render to support processing of complex AI workloads. The architecture is relevant for edge devices, IoT systems, wearable devices where energy efficiency and real time performance is critical. The study contributes a validated, novel, scalable solution for the growing demand of efficient AI hardware for high performance intelligent systems. It is a low complexity, shift based Po2 multiplier providing a highly efficient AI hardware accelerator which can be deployed in edge AI technology.

FUNDING INFORMATION

The design and simulation were carried out in Synosys Simulation tool, funded by Chip-2-Start up scheme, funded by Meity, Govt of India.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Jean Shilpa V	✓	✓	✓	✓	✓	✓			✓			✓	✓	✓
Anitha R		✓		✓		✓			✓		✓			
Anusooya S	✓			✓			✓	✓		✓	✓			
Jawahar P K			✓							✓				✓
Nithesh E	✓	✓			✓	✓		✓						
Sairamsiva S	✓	✓			✓	✓		✓						
Syed Rahaman K	✓	✓			✓	✓	✓	✓						

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

Data availability is not applicable to this paper as no new data were created or analyzed in this study.

REFERENCES

[1] S. Deng, H. Zhao, W. Fang, J. Yin, S. Dustdar, and A. Y. Zomaya, "Edge intelligence: The confluence of edge computing and artificial intelligence," *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 7457–7469, Aug. 2020, doi: 10.1109/JIOT.2020.2984887.

[2] C. Zhu, S. Han, H. Mao, and W. J. Dally, "Trained ternary quantization," *arXiv preprint: arXiv:1612.01064*, 2016.

[3] S. Walia, B. V Tej, A. Kabra, J. Devnath, and J. Mekie, "Fast and lowpower quantized fixed posit high-accuracy DNN implementation," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 30, no. 1, pp. 108–111, Jan. 2022, doi: 10.1109/TVLSI.2021.3123456.

[4] C. Gong, Y. Chen, Y. Lu, T. Li, C. Hao, and D. Chen, "VecQ: Minimal loss DNN model compression with vectorized weight quantization," *IEEE Transactions on Computers*, vol. 70, no. 5, pp. 696–710, May 2021, doi: 10.1109/TC.2021.3056789.

[5] S. Vogel, J. Springer, A. Guntoro, and G. Ascheid, "Self-supervised quantization of pre-trained neural networks for multiplierless acceleration," in *Proc. Design, Automation & Test in Europe Conf. & Exhibition (DATE)*, 2019, pp. 1094–1099, doi: 10.23919/DATE.2019.8714973.

[6] 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 Transactions on Computers*, vol. 66, no. 8, pp. 1435–1441, Aug. 2017, doi: 10.1109/TC.2017.2708982.

[7] H. Waris, C. Wang, W. Liu, and F. Lombardi, "AxBMs: Approximate radix-8 booth multipliers for high-performance FPGA-based accelerators," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 68, no. 5, pp. 1566–1570, May 2021, doi: 10.1109/TCSII.2021.3061234.

[8] S. Venkatachalam, E. Adams, H. J. Lee, and S.-B. Ko, "Design and analysis of area and power efficient approximate booth multipliers," *IEEE Transactions on Computers*, vol. 68, no. 11, pp. 1697–1703, Nov. 2019, doi: 10.1109/TC.2019.2890612.

[9] Y. He, X. Yi, Z. Zhang, B. Ma, and Q. Li, "A probabilistic prediction-based fixed-width booth multiplier for approximate computing," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 67, no. 12, pp. 4794–4803, Dec. 2020, doi: 10.1109/TCSI.2020.3024567.

[10] S. Nambi, U. A. Kumar, K. Radhakrishnan, M. Venkatesan, and S. E. Ahmed, "DeBAM: Decoder-based approximate multiplier for low power applications," *IEEE Embedded Systems Letters*, vol. 13, no. 4, pp. 174–177, Dec. 2021, doi: 10.1109/LES.2021.3126543.

[11] H. Waris, C. Wang, and W. Liu, "Hybrid low radix encoding-based approximate booth multipliers," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 67, no. 12, pp. 3367–3371, Dec. 2020, doi: 10.1109/TCSII.2020.3034568.

[12] P. Yin, C. Wang, H. Waris, W. Liu, Y. Han, and F. Lombardi, "Design and analysis of energy-efficient dynamic range approximate logarithmic multipliers for machine learning," *IEEE Transactions on Sustainable Computing*, vol. 6, no. 4, pp. 612–625, 2021, doi: 10.1109/TSUSC.2021.3089123.

[13] R. Pilipovic, P. Bulić, and U. Lotrič, "A two-stage operand trimming approximate logarithmic multiplier," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 68, no. 6, pp. 2535–2545, Jun. 2021, doi: 10.1109/TCSI.2021.3067894.

[14] 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 Transactions on Computers*, vol. 68, no. 5, pp. 660–675, May 2019, doi: 10.1109/TC.2019.2903456.

[15] L. M. Ang, K. P. Seng, G. K. Ijemaru, and A. M. Zungeru, "Deployment of IoV for smart cities: applications, architecture, and challenges," *IEEE Access*, vol. 7, pp. 6473–6492, 2019, doi: 10.1109/ACCESS.2018.2887076.

[16] D. Przewlocka-Rus, S. S. Sarwar, H. E. Sumbul, Y. Li, and B. De Salvo, "Power-of two quantization for low bitwidth and hardware compliant neural networks," *arXiv preprint: arXiv:2203.05025*, 2022.
[17] S. Vahdat, M. Kamal, A. Afzali-Kusha, and M. Pedram, "TOSAM: An energy-efficient truncation- and rounding-based scalable approximate multiplier," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 27, no. 5, pp. 1161–1173, May 2019, doi: 10.1109/TVLSI.2019.2891234.
[18] M. Asadikouhanjani and S.-B. Ko, "Enhancing the utilization of processing elements in spatial deep neural network accelerators," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 40, no. 9, pp. 1947–1951, Sep. 2021, doi: 10.1109/TCAD.2021.3076541.
[19] M. Asadikouhanjani, H. Zhang, L. Gopalakrishnan, H.-J. Lee, and S.-B. Ko, "A realtime architecture for pruning the effectual computations in deep neural networks," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 68, no. 5, pp. 2030–2041, May 2021, doi: 10.1109/TCSI.2021.3054321.
[20] F. Tu, S. Yin, P. Ouyang, S. Tang, L. Liu, and S. Wei, "Deep convolutional neural network architecture with reconfigurable computation patterns," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 25, pp. 2220–2233, 2017, doi: 10.1109/TVLSI.2017.2679784.
[21] Y.-H. Chen, T. Krishna, J. S. Emer, and V. Sze, "Eyeriss: an energy-efficient reconfigurable accelerator for deep convolutional neural networks," *IEEE Journal of Solid-State Circuits*, vol. 52, no. 1, pp. 127–138, 2017, doi: 10.1109/JSSC.2016.2625978.
[22] Y. Umuroglu, D. Conficconi, L. Rasnayake, T. B. Preusser, and M. Själander, "Optimizing bit-serial matrix multiplication for reconfigurable computing," *ACM Transactions on Reconfigurable Technology and Systems*, 2019, doi: 10.1145/3326361.
[23] J. Garland and D. Gregg, "Low complexity multiply accumulate unit for weight-sharing convolutional neural networks," *IEEE Computer Architecture Letters*, vol. 16, no. 2, pp. 132–135, 2017, doi: 10.1109/LCA.2017.2718506.
[24] A. Parashar *et al.*, "SCNN: an accelerator for compressed-sparse convolutional neural networks," in *Proceedings of the 44th Annual International Symposium on Computer Architecture (ISCA)*, 2017, pp. 27–40, doi: 10.1145/3079856.3080243.
[25] S. Lee, D. Kim, D. Nguyen, and J. Lee, "Double MAC on a DSP: boosting the performance of convolutional neural networks on FPGAs," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2019, doi: 10.1109/TCAD.2018.2871231.

BIOGRAPHIES OF AUTHORS

Jean Shilpa V. received her B.E. degree in electronics and communication engineering from Visvesvaraya Technological University, Karnataka, M.Tech degree in VLSI Design from Vellore Institute of Technology and Ph.D. Degree from B.S. Abdur Rahman Crescent Institute of Science and Technology, Chennai, India. She is currently working as associate professor, in the Department of Electronics and Communication Engineering, B.S. Abdur Rahman Crescent Institute of Science and Technology. Her main research interest includes, hybrid FPGA-CPU based designs, and memristor based designs. She is a Certified Lab-view Associate developer. She has more than 25 papers in peer reviewed journals and more than 40 papers in national and international journals. She has three patents, authored a book on "VLSI Design" and four book chapters published. She has been granted a funded proposal from Chip-2-Startup, Meity in 2024. He can be contacted at email: jeanshilpa@crescent.education.

Anitha R. received her B.E. degree in electronics and communication engineering from Madras University, Tamil Nadu, M.Tech. degree in communication system from S.A. Engineering College and Ph.D. Degree from B.S. Abdur Rahman Crescent Institute of Science and Technology, Chennai, India. She is currently working as assistant professor (Selection Grade), in the Department of Electronics and Communication Engineering, B.S. Abdur Rahman Crescent Institute of Science and Technology. Her main research interest includes, signal processing and cryptography. She is a Certified Lab-view associate developer. She has more than 20 papers in peer reviewed journals and more than 40 papers in national and international journals. She has three patents, three book chapters published. He can be contacted at email: r.anitha@crescent.education.

Anusooya S. received her B.E. degree in electronics and communication engineering from Anna University and M.Tech. degree in applied electronics from Anna University. She received her Ph.D degree in the Department of Electronics and Communication Engineering of B.S. Abdur Rahman Crescent Institute of Science and Technology, Chennai, India. She is currently working as assistant professor (Sel.Gr), in the Department of Electronics and Communication Engineering in B.S. Abdur Rahman Crescent Institute of Science and Technology. Her current interest includes, analog electronics, low power VLSI and mixed signal design. She is a Certified LabVIEW Associate Developer. She has more than 30 papers in national and international journals. She has two patents and two book chapters published. He can be contacted at email: anusooya@crescent.education.

Jawahar P. K. is a professor with over 32 years of academic and research experience, currently serving at BSACIST since July 2000. He holds a B.E. in electronics and communication engineering from Coimbatore Institute of Technology (1989), an M.Tech in the same discipline from Pondicherry Engineering College (1998), a Ph.D. in information and communication engineering from Anna University through MIT, Chromepet (2010), and a Postgraduate Diploma in VLSI from Accel Technologies, Chennai (2002). His areas of expertise include VLSI, IoT, embedded systems, and computer networking. Dr. Jawahar has guided five Ph.D. scholars and is currently supervising three more, with one thesis submitted and another synopsis completed. His research interests lie in VLSI system design, IoT, and computer networks, with a citation count of 65 and an h-index of 5. He is a senior member of IEEE, a member of ACM and IAEng, and a Fellow of both the Institution of Engineers (India) and the Institution of Electronics and Telecommunication Engineers. He is also a Life Member of ISTE. He can be contacted at email: jawahar@crescent.education.

Nithesh E. was born on 18th January 2004 in Chengalpattu, Tamil Nadu. He completed his schooling at Infant Jesus Matric Higher Secondary School, Kalpakkam, and graduated in the year 2021. He is currently pursuing a Bachelor of Technology degree in electronics and communication engineering at B.S. Abdur Rahman Crescent Institute of Science and Technology, Chennai. His areas of interest include VLSI, artificial intelligence (AI), the internet of things (IoT), cloud computing, communication systems, and embedded systems. He is passionate about leveraging emerging technologies to solve real-world challenges, with a particular focus on automation and networking systems. He aspires to contribute to the development of intelligent, efficient, and connected solutions. He can be contacted at email: nnithesh542@gmail.com.

Sairamsiva S. was born on 5th July 2004 in Kumbakonam, Tamil Nadu. He completed his schooling at Town Higher Secondary School, Kumbakonam, and graduated in the year 2021. He is currently pursuing a Bachelor of Technology degree in electronics and communication engineering at B.S. Abdur Rahman Crescent Institute of Science and Technology, Chennai. His areas of interest include VLSI design, embedded systems, IoT-based automation, wireless communication, and edge-cloud computing for smart systems. He is passionate about integrating hardware and software to develop intelligent, efficient, and scalable solutions. He aspires to contribute to the advancement of modern ECE technologies with a focus on real-time applications and innovation in automation. He can be contacted at email: sairamsiva05@gmail.com.

Syed Rahaman K. was born on 8th June 2004 in Villupuram, Tamil Nadu. He completed his schooling at Saraswathi Matric Higher Secondary School, Villupuram, and graduated in the year 2021. He is currently pursuing a Bachelor of Technology degree in electronics and communication engineering at B.S. Abdur Rahman Crescent Institute of Science and Technology, Chennai. His areas of interest include VLSI, artificial intelligence (AI), the internet of things (IoT), cloud computing, communication systems, and embedded systems. He is passionate about leveraging emerging technologies to solve real-world challenges, with a particular focus on automation and networking systems. He aspires to contribute to the development of intelligent, efficient, and connected solutions. He can be contacted at email: syedrahman2004@gmail.com.