A Comprehensive Review of Approximate Multipliers in Power-Efficient Computing

Deepak Giri[1], Sachin Bandewar[2]
M.Tech.Scholar[#1], Assistant Professor[#2]
Department of Electronics and Communication Engineering
Sri Satya Sai College of Engineering, RKDF University, Bhopal, (M.P.) India
sachin.bandewar9@gmail.com

Abstract: - This review paper presented into the significance and application of multipliers in both economics and digital systems, emphasizing their role in amplifying or modifying original values. The burgeoning demand for power-efficient computing is explored, driven by concerns related to environmental sustainability, financial implications, and optimization of user experience in battery-powered devices. The focus then shifts to approximation computing, a strategy that seeks gains in energy efficiency, speed, and resource use at the expense of absolute precision. Several research studies and methodologies related to approximate multipliers are highlighted, ranging from unique designs grounded in decoder logic to algorithms optimized for specific applications. This paper also underscores the potential applications of approximate computing in diverse domains such as signal processing, machine learning, scientific computing, data analytics, and graphics processing, all while recognizing the challenges it presents.

Keywords: - Field Programmable Gate Array, Artificial Intelligence, Machine Learning, Internet of Things

INTRODUCTION

A multiplier serves as a factor that amplifies or modifies the original value of a given quantity. For example, a 2x multiplier doubles the initial value, while a 0.5x multiplier cuts it in half. In economic theories, multipliers are often tied to the concept of investment as a key driver of economic growth. The multiplier effect explains how an increase in investment leads to proportional rises in income and employment levels within the economy.

In the context of electronics and digital systems, multipliers are indispensable components for carrying out multiplication tasks. These devices come in a variety of forms and designs, each suited for particular applications and performance needs. A deep understanding of the various kinds of multipliers, along with their selection criteria, is vital for the efficient design of electronic systems. Multipliers are basic circuit elements used in both digital and analog systems to execute multiplication functions. They find applications in a wide range of areas, such as digital signal processing, analog signal manipulation, and communication systems, among others. Different methodologies can be used to implement multipliers, each with its own set of advantages and performance considerations. Typically, multipliers take in two input values—commonly known as the multiplicand and the multiplier—and generate an output that is the mathematical product of the two.

The Growing Demand for Power-Efficient Computing

The rising need for energy-efficient computation is fueled by multiple considerations such as environmental sustainability, financial savings, and enhanced performance across diverse use-cases. This form of computing focuses on creating hardware and software solutions that execute tasks while using as little energy as possible. Several factors contribute to its growing importance. Awareness about the urgency to combat climate change and preserve the environment has led to initiatives aimed at curbing energy use across various industries, including the computing sector. Facilities like data centers and high-performance computing (HPC) installations consume massive amounts of electricity, so making them more energy-efficient can have a considerable impact on lowering their ecological footprint.

Moreover, the operational costs for data centers and businesses heavily reliant on computing power can be substantial. Energy-efficient computing offers an avenue for reducing these expenses through lower energy bills. In today's world, mobile technology such as smartphones, laptops, and IoT devices are ubiquitous and usually run on batteries. Enhancing the energy efficiency of computing processes is crucial for extending the battery lifespan and optimizing the overall user experience for these gadgets. Artificial Intelligence (AI) and machine learning (ML) technologies, which often require heavy computational workloads, are becoming more common across sectors. The need for energy-efficient computing solutions is increasingly vital in these areas, particularly when resources are limited.

Approximation Computing

Approximation computing focuses on using imprecise methods to perform calculations, aiming

for gains in energy efficiency, speed, and resource use, rather than exact results. This approach is beneficial in scenarios where high precision is not essential, such as battery-powered devices, real-time applications, and resource-limited environments like IoT systems. The technique can significantly reduce energy consumption and increase computational speed. Various research papers have proposed different types of approximate multipliers, using techniques like error accumulation and truncation of least significant bits, to achieve these efficiencies.

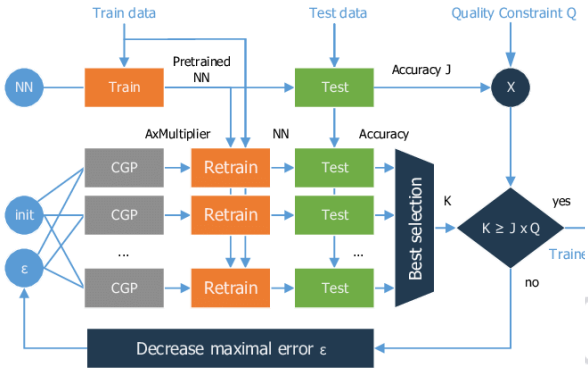

Fig 1. Approximate multiplier design for approximate NNs[9]

Various approximate compressors and multipliers are proposed in the literature to reduce delay and power consumption in computing tasks. These designs, like ACM3 and ACM4, apply approximation techniques mainly to the least significant bits (LSBs) while keeping the most significant bits (MSBs) exact. Various models such as broken-array, Wallace tree, and error-tolerant multipliers utilize approximation in different stages of the multiplication process. Some even use logarithmic approximations to further simplify calculations. These techniques aim to create more energy-efficient and faster computing systems.

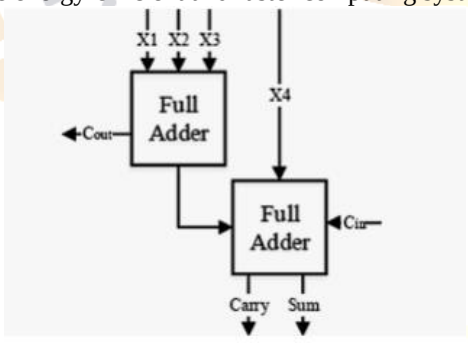

Fig 2. Approximate multiplexer [10]

LITERATURE REVIEW

Nimbi et al. [1] introduced a unique multiplier design grounded in decoder logic, aiming to minimize the number of partial products produced.

Sathish et al.[2] introduced a novel approach that leverages the segmentation of partial products through recursive multiplication by compressors in their approximate multipliers.

Liu et al. [3] introduced a Mitchell multiplication algorithm refined through piecewise linear logarithmic approximation.

Arya et al.[4] introduced the Reconfigurable Energy-efficient Approximate Divider (READ) built upon the fixed restoring array divider structure. When benchmarked against a precise 16⁄8 divider design, READ demonstrated a 49% boost in energy efficiency and operated 1.26 times faster, with only slight accuracy deviations.

SenthilKuma et al.[5] introduced a compressor design that simplifies system operations by cutting down on the number of XOR gates utilized. The results highlighted a notable reduction in power usage with only a slight compromise in accuracy.

Zhu et al.[6] introduced designs for approximate-truncated Booth multipliers (ATBMs) that leverage their novel approximate modified radix-4 Booth encoders (AMBEs), approximate 4-2 compressors (ACs), and a tiered truncation of partial products.

Qui et al.[7] introduced a pair of innovative approximate circuit design techniques anchored in machine learning. Instead of intricate manual evaluations, these methods prioritize input-error patterns to shape the approximate circuits.

Devi et al.[8] introduced a perspective on computational efficiency as technological scaling nears its boundaries. Their focus was on the approximate design of the RB-Normal Binary (NB) converter within the RB multiplier, taking into account the error characteristics of both the estimated Booth encoders and the RB compressors.

Mittal et al.[9] offer an overview of methods related to approximate computing (AC). The review delves into identifying sections of programs suitable for approximation, ways to ensure output quality, and the integration of AC across various processing units, including CPUs, GPUs, and FPGAs.

Narayanamoorthy et al.[10] introduced multiplier designs that allow a balance between computational precision and energy usage during the design phase.

Jiang et al.[11] offer an analysis and categorization of contemporary designs in approximate arithmetic circuits, encompassing elements like adders, multipliers, and dividers. The study conducts an exhaustive comparison of the error rates and circuitry traits across these designs, aiding in grasping the nuances of each.

Parhami et al.[12] have made significant contributions through their investigative studies and tool development, elevating computer architecture from a mere art form to a notably quantitative domain within computer science and engineering.

Liang et al.[13] introduced novel criteria for assessing both the dependability and energy efficiency of probabilistic and approximate adders.

Yi et al.[14] introduced an innovative circuit design that alters the circuit's architecture without affecting its operation. AWM3 achieves area reductions of up to 48.077% and minimizes delay by up to 46.633%.

Meanwhile, AWM4 showcases reductions of up to 53.846% in size and up to 56.482% in processing time

Bhardwaj, et al.[15] introduces an approximate multiplication technique that takes into account bit-width for enhancing the design of our multiplier. Through simulations, the team found that the designs have an average accuracy ranging between 99.85% and 99.965%.

Rao et al.[16] introduced the Rounding-based AM (RAM) by adapting the Karatsuba algorithm. Through simulations, it was revealed that the newly introduced RAM designs for 8 and 16 bits showcased significant improvements. Specifically, when integrated with the ISFA, the RAM's SSIM and PSNR values ranged from 1.44% to 84.47% and 0.28% to 24.4% respectively, outperforming the ISFA-integrated existing AMs.

Yang et al.[17] introduced three innovative compressors designed for partial product reduction (PPR) within multipliers, ensuring they maintain a level of accuracy. These designs incorporate both approximation and truncation strategies.

Ha et al.[18] introduced an enhanced design that builds upon a prior approximate 4–2 compressor model by incorporating an error correction component.

Hashimi et al.[19] developed a multiplier with an emphasis on achieving an unbiased error distribution. The errors introduced have a Gaussian distribution, characterized by an almost negligible average and standard deviations ranging from 0.45% to 3.61%.

Venkatachalam et al.[20] introduced optimized approximate multipliers where the multiplier's partial products undergo modifications using generate and propagate signals. These approximations were incorporated into two distinct 16-bit multiplier models. Synthesis data indicates that the newly proposed multipliers offer power efficiencies of 72% and 38% when set against a standard multiplier.

Design of Cost-Efficient Approximate Log Multipliers

The Mitchell Log Multiplier algorithm is introduced, highlighting its original lack of a specified digital circuit implementation. A low-cost implementation is developed and optimized through various techniques. The Truncated Mitchell Log Multiplier is then presented as a further approximation of the Mitchell Log Multiplier. It involves locating leading ones and significant portions of input operands to truncate them after taking the logarithm. This approach allows for reasonably accurate multiplication while significantly reducing hardware costs.

The issue of bias in the log multiplier is addressed, as it typically produces negative errors and lacks positive errors. To create unbiased designs with a mean error closer to zero, both error sources (w truncation and approximate logarithm) need to be considered.

For CNN architectures using depthwise separable convolution, the Truncated Iterative Log Multiplier is discussed. This structure aims to enhance the accuracy of the Mitchell Log Multiplier but also focuses on reducing the cost of basic blocks to make iterative logarithmic multiplication more affordable.

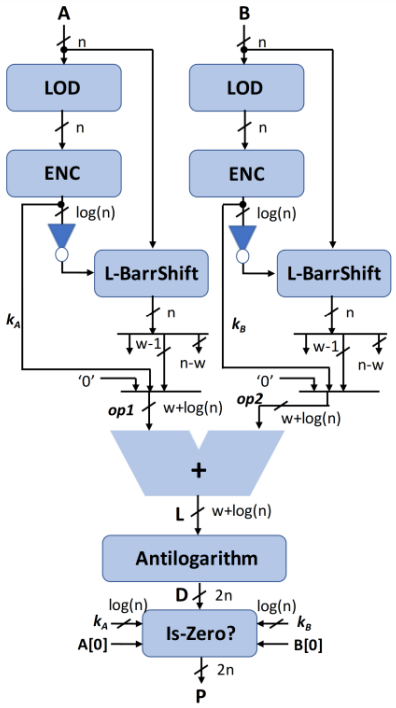

Figure 3: Truncated Mitchell Log Multiplier

Effects of Approximate Multiplication on Convolutional Neural Networks

CNNs consist of convolution layers followed by fully connected layers for predictions. They've grown in size for better accuracy, with more convolution layers and fewer fully connected layers. Some use 1x1 convolutions as classifiers. CNNs excel in image recognition, as they leverage spatial locality to detect features. Each convolution layer performs operations on input channels with kernels, creating output channels for different features. Computationally expensive convolution layers are optimized using approximate multipliers, without altering the CNN architecture. Approximate computing enhances digital circuit performance by allowing less precise calculations, particularly in adders.

Applications of Approximation Computing

Approximation techniques find applications across various domains to enhance speed and energy efficiency:

Signal Processing: Used in digital signal processing for quicker and more energy-efficient image and audio algorithms.

Machine Learning: Allows for faster computations in AI and ML without significantly affecting model accuracy.

Scientific Computing: Speeds up simulations and calculations while maintaining acceptable precision.
Data Analytics: Helps in managing large-scale data processing tasks more efficiently.
Graphics Processing: Used in GPUs for real-time performance in gaming and computer graphics through approximate rendering techniques.

Challenges in Approximation Computing

Approximate computing is an evolving field that focuses on balancing accuracy and efficiency to achieve energy-saving and high-performance results in various applications like multimedia processing, machine learning, and scientific simulations. While this approach is promising, it presents challenges such as ensuring quality control, designing adaptable algorithms, analyzing errors, and creating programmer-friendly tools. The paradigm leverages the error-resilience of certain applications, allowing for controlled approximations in computation, memory, and data transfer to enhance energy efficiency and speed. Techniques like loop perforation and using reduced data precision are among the methods employed. As energy efficiency becomes increasingly important, especially in AI and IoT contexts, approximate computing is expected to gain more relevance.
Decoders find a wide range of applications in various domains:
Code Conversions: Used for transforming one type of code into another, such as binary to decimal conversions.
Memory Systems: Integral in computer memory systems for de-multiplexing and data distribution.
Data Routing: Employed in applications that require minimal propagation delay for quick data routing.
Specific Output Lines: Utilized for directing input data to specified output lines, commonly in addressing core memory in computers.
Format Conversions: Used in converting formats, like turning a 2-bit binary input into a 4-bit output.
These applications demonstrate the versatility and utility of decoders in computing and data management.

CONCLUSION

Approximate computing is a computational paradigm that presents several advantages, including reductions in area, delay, and power consumption. This particular research focuses on harnessing the inherent error-resilient nature of image processing applications. This paper reviews work done in Approximate computing and also gives its applications and challenges. The future scope of this research lies in further optimizing and adapting architecture for a broader range of computational tasks beyond image processing, exploring its potential applications in fields like machine learning, signal processing, and scientific simulations. Additionally, the development of automated techniques for determining when and where to employ approximate computing in different contexts can be an avenue for future investigation, enhancing its practicality and impact in resource-constrained computing environments.

REFERENCES

[1] S. Nambi, U. A. Kumar, K. Radhakrishnan, M. Venkatesan, and S. E. Ahmed, “DDM: Decoder-Based Approximate Multiplier for Low Power Applications,” IEEE Embed. Syst. Lett., vol. 13, no. 4, pp. 174–177, Dec. 2021, doi: 10.1109/LES.2020.3045165.

[2] V. A and D. S. K. G. A, “Approximate Multiplier for Low Power Applications,” Int. J. Eng. Res. Technol., vol. 4, no. 14, Jul. 2018, doi: 10.17577/IJERTCONV4IS14039.

[3] H. Liu, M. Wang, L. Yao, and M. Liu, “A Piecewise Linear Mitchell Algorithm-Based Approximate Multiplier,” Electron. 2022, Vol. 11, Page 1913, vol. 11, no. 12, p. 1913, Jun. 2022, doi: 10.3390/ELECTRONICS11121913.

[4] N. Arya, T. Soni, M. Pattanaik, and G. K. Sharma, “READ: A fixed restoring array based accuracy-configurable approximate divider for energy efficiency,” Integration, vol. 76, pp. 1–12, Jan. 2021, doi: 10.1016/J.VLSI.2020.08.002.

[5] K. K. Senthilkumar et al., “Approximate Multiplier based on Low power and reduced latency with Modified LSB design”, doi: 10.1051/e3sconf/202339901009.

[6] Y. Zhu, W. Liu, P. Yin, T. Cao, J. Han, and F. Lombardi, “Design, evaluation and application of approximate-truncated Booth multipliers,” IET Circuits, Devices Syst., vol. 14, no. 8, pp. 1305–1317, Nov. 2020, doi: 10.1049/IET-CDS.2019.0398.

[7] L. Qiu, “Designing Approximate Computing Circuits with Scalable and Systematic Data-Driven Techniques Recommended Citation”, Accessed: Sep. 19, 2023. [Online]. Available: https://tigerprints.clemson.edu/all_theses

[8] Da. Devi, P. Sai Kumar, R. Swapna, and A. Siddhartha, “International Journal of Computer Science and Mobile Computing Design Analysis of Approximate Redundant Binary Multipliers,” Int. J. Comput. Sci. Mob. Comput., vol. 11, no. 1, pp. 74–94, 2022, doi: 10.47760/ijcsmc.2022.v11i01.010.

[9] S. Mittal, “A Survey of Techniques for Approximate Computing,” ACM Comput. Surv., vol. 48, no. 4, Mar. 2016, doi: 10.1145/2893356.

[10] S. Narayanamoorthy, H. A. Moghaddam, Z. Liu, T. Park, and N. S. Kim, “Energy-

efficient approximate multiplication for digital signal processing and classification applications," IEEE Trans. Very Large Scale Integr. Syst., vol. 23, no. 6, pp. 1180–1184, Jun. 2015, doi: 10.1109/TVLSI.2014.2333366.

[11] H. Jiang, F. Lombardi, and J. Han, "60 A Review, Classification and Comparative Evaluation of Approximate Arithmetic Circuits," 2017, doi: 10.1145/3094124.

[12] S. E. C O N D E D I T I O N Behrooz Parhami and N. York Oxford, "COMPUTER ARITHMETIC Algorithms and Hardware Designs," 2010, Accessed: Sep. 19, 2023. [Online]. Available: http://www.oup.com

[13] J. Liang, J. Han, and F. Lombardi, "New metrics for the reliability of approximate and probabilistic adders," IEEE Trans. Comput., vol. 62, no. 9, pp. 1760–1771, 2013, doi: 10.1109/TC.2012.146.

[14] X. Yi, H. Pei, Z. Zhang, H. Zhou, and Y. He, "Design of an energy-efficient approximate compressor for error-resilient multiplications," Proc. - IEEE Int. Symp. Circuits Syst., vol. 2019-May, 2019, doi: 10.1109/ISCAS.2019.8702199.

[15] K. Bhardwaj, P. S. Mane, and J. Henkel, "Power- and area-efficient Approximate Wallace Tree Multiplier for error-resilient systems," Fifteenth Int. Symp. Qual. Electron. Des., pp. 263–269, 2014, doi: 10.1109/ISQED.2014.6783335.

[16] E. J. Rao, K. T. Rao, K. S. Ramya, D. Ajaykumar, and R. Trinadh, "Efficient Design of Rounding-Based Approximate Multiplier Using Modified Karatsuba Algorithm," J. Electron. Test., vol. 38, no. 5, pp. 567–574, Oct. 2022, doi: 10.1007/S10836-022-06029-4.

[17] Z. Yang, J. Han, and F. Lombardi, "Approximate compressors for error-resilient multiplier design," Proc. 2015 IEEE Int. Symp. Defect Fault Toler. VLSI Nanotechnol. Syst. DFTS 2015, pp. 183–186, Nov. 2015, doi: 10.1109/DFT.2015.7315159.

[18] HaMinho and LeeSunggu, "Multipliers With Approximate 4–2 Compressors and Error Recovery Modules," IEEE Embed. Syst. Lett., Mar. 2018, doi: 10.5555/3195730.3195769.

[19] S. Hashemi, "DRUM: A Dynamic Range Unbiased Multiplier for Approximate Applications," 2015. doi: 10.1109/ICCAD.2015.7372600.

[20] S. Venkatachalam and S.-B. Ko, "Design of Power and Area Efficient Approximate Multipliers," 2017, doi: 10.1109/TVLSI.2016.2643639.