Nanotechnology Perceptions
ISSN 1660-6795
www.nano-ntp.com

Design and Implementation of an Optimized Error-Tolerant Adder (ETA) for Energy-Efficient FPGA Systems

Haritha C[1], Dr. Santhosh B Panjagal[2], Dr. Rasadurai[3]

[1]M.Tech Student, Department of Electronics and Communication Engineering , Kuppam Engineering College, India
[2]Associate Professor, Department of Electronics and Communication Engineering, Kuppam Engineering College, India
[3]Professor, Department of Electronics and Communication Engineering, Kuppam Engineering College, India
Email: agastyaharitha@gmail.com

In modern digital systems, power and area-efficient arithmetic units are crucial for optimizing FPGA-based applications. Approximate computing, particularly in error-tolerant adders (ETAs), provides a trade-off between computational accuracy and hardware efficiency. This paper presents the design and implementation of a high-performance ETA adder, compared with an existing hybrid approximate adder that integrates the low-error and area-efficient approximate adder (LEADx) and the area and power-efficient approximate adder (APEx). The proposed ETA adder significantly improves hardware efficiency by reducing the area and power consumption while maintaining acceptable computational accuracy. FPGA implementation results demonstrate that the ETA adder achieves a 65% reduction in area (LUTs), decreasing from 20 LUTs in the existing hybrid approximate adder to just 7 LUTs. Additionally, power analysis reveals a 28.6% reduction in total power consumption, with the proposed ETA adder consuming 3.651W compared to 5.117W in the existing design. The logical power is notably optimized, reducing from 0.098W to 0.026W, enhancing the overall energy efficiency of the system. These results highlight the suitability of ETA adders for power-sensitive FPGA Signal processing, machine learning accelerators, and image processing are just a few examples, where minor computational inaccuracies are acceptable in exchange for significant hardware optimizations.

Keywords: Low error, low power, FPGA, approximate computation, approximate adder, LUT etc.

1. Introduction

The demand for high-performance, energy-efficient computing has led to significant research in approximate computing, particularly in the design of arithmetic circuits for FPGA-based applications. Approximate computing is widely used in error-resilient applications such as

image processing, digital signal processing (DSP), machine learning, and video compression, where slight computational errors are tolerable in exchange for significant power and area savings [5], [12], [17-19].

Among arithmetic circuits, adders play a crucial role in computation-heavy tasks. Conventional adders are designed for maximum accuracy but often consume significant power and area. To address these limitations, approximate adders have been proposed as a promising solution. Various approximate adder designs, such as low-error and area-efficient approximate adders (LEADx), area and power-efficient approximate adders (APEx), and speculative adders, aim to achieve a balance between hardware efficiency and computational accuracy [15], [16].

While these designs have demonstrated notable improvements in power and area efficiency, further optimization is required to minimize logical power consumption, static power dissipation, and area overhead in FPGA-based implementations [7], [13]. Error-Tolerant Adders (ETAs) have emerged as a viable alternative, offering better energy efficiency by sacrificing minimal accuracy in applications where small computational errors do not significantly impact overall performance [9], [10].

Despite the progress in approximate computing, existing approximate adder designs face the following key challenges:

1. Tradeoff Between Accuracy and Power Efficiency – Many approximate adders, including LEADx and APEx, achieve reduced power consumption but introduce higher error rates, which can be unsuitable for certain error-sensitive applications [5], [12].

2. Logical and Dynamic Power Optimization – Although approximate adders reduce area usage, dynamic and logical power consumption still remains high in certain designs, making them inefficient for low-power FPGA-based applications [6], [14].

3. Area and LUT Utilization – Some existing designs, such as MACISH-based adders and accuracy-configurable adders, still exhibit high LUT utilization, leading to increased hardware complexity [1], [11].

4. Performance in FPGA-Based Systems – Many approximate adders focus on ASIC implementations, with limited research optimizing their design for FPGA-based applications, which require efficient LUT and power management [3], [7].

Given these challenges, the motivation of this work is to design an Error-Tolerant Adder (ETA) that achieves:

- Significant area reduction, optimizing LUT usage for FPGA-based implementations.
- Lower logical and dynamic power consumption, making it energy-efficient.
- Better tradeoff between accuracy and power efficiency, minimizing errors while maintaining computational reliability.

The key objectives of this research are:

1. To design and implement a high-performance ETA adder for FPGA-based applications.

2. To compare the proposed ETA adder with existing hybrid approximate adders (LEADx and APEx) in terms of area, power consumption, and performance.

3. To evaluate the impact of the proposed design on logical power, static power, and dynamic power efficiency.

4. To validate the FPGA implementation results and demonstrate the suitability of ETA adders for low-power applications.

This paper makes the following contributions:

- Proposes a high-performance ETA adder that significantly reduces area utilization (LUTs), logical power, and dynamic power compared to existing approximate adders.
- Demonstrates a 65% reduction in LUTs and a 28.6% reduction in total power consumption compared to hybrid approximate adders such as LEADx and APEx.
- Optimizes logical power consumption, achieving a reduction from 0.098W to 0.026W, making it suitable for low-power FPGA applications.
- Validates the design through FPGA implementation, highlighting its suitability for power-sensitive applications such as image processing, machine learning accelerators, and digital signal processing.

The rest of this paper is organized as follows:

Section 2 presents a detailed literature survey, discussing prior work on approximate adders and their impact on FPGA-based applications. Section 3 describes the proposed ETA adder design and its implementation methodology. Section 4 provides a comparative analysis of the proposed and existing adders, including performance metrics such as LUT utilization, power consumption, and accuracy. Section 5 discusses the FPGA implementation results and their significance for real-world applications. Section 6 concludes the paper and suggests potential future research directions.

2. Related Works

In recent years, the design of approximate adders and computational circuits has gained attention due to the need for energy-efficient hardware for error-resilient applications. This survey reviews a selection of research articles that address various aspects of approximate computing, with a focus on adders, multipliers, and their application in FPGA-based systems.

Gillani et al. (2019) propose MACISH, a design for approximate MAC accelerators that incorporate internal self-healing mechanisms. This work emphasizes the tradeoff between computational accuracy and resource efficiency, making it a significant contribution to approximate computing in FPGA-based accelerators [1].

Kalali and Hamzaoglu (2020) focus on approximate HEVC intra angular prediction hardware. They explore the tradeoffs between power, performance, and accuracy in hardware implementations of video encoding algorithms. Their approach demonstrates a promising direction for approximate computation in multimedia applications [2].

Ayhan and Altun use neural networks and image processing as case studies to investigate circuit-aware approximate system design. An effective approach to approximate system design that prioritizes hardware cost while maintaining acceptable error levels for specific applications is presented in their work [3].

Ahmad and Hamzaoglu (2021) present an efficient approximate sum of absolute differences hardware for FPGAs. This work focuses on optimizing hardware for approximate computing in image processing tasks, particularly in the context of motion estimation for video compression [4].

An overview, classification, and comparison of approximate arithmetic circuits are provided by Jiang et al. (2017). Their work evaluates the efficacy of various approaches to approximate arithmetic design in a variety of computational tasks and provides a comprehensive overview of the various approaches [5].

Mert et al. (2019) introduce a novel approximate absolute difference hardware. Their design offers improvements in error reduction and power efficiency, particularly suited for video and image processing tasks [6].

Van Toan and Lee (2020) focus on FPGA-based multi-level approximate multipliers for high-performance, error-resilient applications. This work demonstrates how multiple approximation levels can be leveraged to optimize performance for FPGA-based systems in error-tolerant applications [7].

The design, evaluation, and use of approximate high-radix dividers are discussed in Chen et al. (2018). Their research focuses on digital signal processing, which calls for high-throughput arithmetic circuits and where approximate dividers can boost performance with little effect on accuracy [8].

Verma et al. (2008) introduce the concept of speculative addition in variable latency. For arithmetic circuit design it is new paradigm. This research looks at ways to adjust latency in addition to circuits for approximate computations, which can control error levels and reduce power consumption [9].

Zhu et al. (2010) propose an enhanced low-power high-speed adder for error-tolerant applications. Their design focuses on minimizing power consumption while maintaining low error rates, targeting applications such as signal processing and communications [10].

Kahng and Kang (2012) present an accuracy-configurable adder for approximate arithmetic designs. This work highlights the tradeoffs between accuracy and power consumption in adder designs, particularly in applications with varying tolerance for errors [11].

Gupta et al. (2013) explore low-power digital signal processing using approximate adders. Their research shows that approximate adders can achieve significant power savings without severely affecting the quality of signal processing applications [12].

Ahmad et al. (2020) compare approximate circuits for H.264 and HEVC motion estimation. Their work highlights the benefits of approximate adders and multipliers in video compression tasks, where small errors are often acceptable and can lead to power and area savings [13].

Mahdiani et al. (2010) propose bio-inspired imprecise computational blocks for efficient VLSI implementation of soft computing applications. Their approach focuses on building circuits that intentionally tolerate errors to reduce hardware complexity and power consumption [14].

Dalloo et al. (2018) systematically design an approximate adder known as the optimized lower part constant-OR adder. This work presents an adder architecture that minimizes error and power consumption while optimizing the hardware for specific application scenarios [15].

Balasubramanian et al. (2021) introduce an approximate adder with a near-normal error distribution. They analyze the error characteristics and practical applications of their adder design, which aims to balance error performance with power efficiency [16].

The surveyed literature reveals the growing interest in approximate computing, particularly in the design of adders and multipliers for FPGA-based applications. The various approaches reviewed provide insights into how approximation techniques can be applied to enhance performance, reduce power consumption, and maintain acceptable error levels in applications ranging from video encoding to signal processing and neural networks. These studies demonstrate the potential of approximate hardware designs to meet the demands of modern, error-tolerant applications.

3. Existing Method

In existing focuses on designing low-error, efficient approximate adders optimized for FPGA implementations. The primary objective is to minimize the Mean Square Error (MSE) while improving area and power efficiency. The method leverages hierarchical carry chain optimization, ensuring reduced error accumulation and efficient use of FPGA resources.

1. Dual-Component Approximate Adder Design

Each approximate adder consists of two parts: Accurate Portion: Responsible for handling critical high-order bits with full precision. Approximate Portion: Implements a controlled error approximation strategy to reduce hardware complexity.

2. Development of Two Approximate Adders

APEx (Area and Power Efficient Approximate Adder)

Prioritizes area and power efficiency while maintaining an acceptable error margin. Best suited for low-power, resource-constrained applications. LEADx (Low Error and Area Efficient Approximate Adder) Focuses on minimizing MSE while keeping area efficiency. Provides higher accuracy than APEx, making it ideal for error-sensitive applications.

FPGA-specific area-efficient and low-error approximate adder

In the least significant m 2 bits of the approximate adder architecture depicted in Fig.2, the proposed LEADx approximate adder makes use of "(m 2)/2" instances of the AAd2 adder. Fig 2 & 4 bit LEADx. In LEADx, Cm−2 is set equal to Am−3. AAd2 executes a 5-to-2 logic function that is mapped to a single Look-Up Table (LUT), like AAd1. Consequently, ⌈m/2⌉ LUTs are employed for the Least Significant Part (LSP), working simultaneously. As a result,

the LSP's delay is equivalent to that of a single LUT (tLUT). From the input Am2 to the output Sn1, LEADx's critical path is a one-way one-way system.

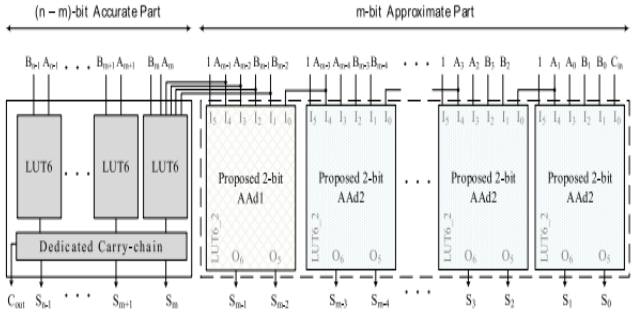

Fig. 1. n-bit error area

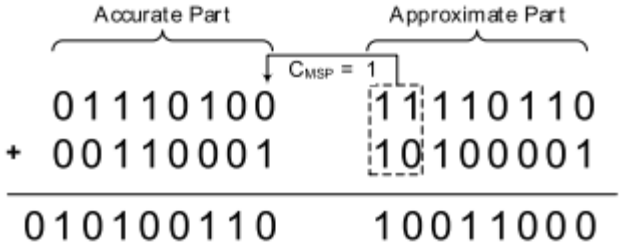

Fig. 2. Example of 16 bit

Power- and area-efficient FPGA approximate adder

In the proposed APEx, the outputs S0 to Cm−3 are set to 1, and Cm−2 is set to 0. This results in notable reductions in area and power consumption, albeit with a slight compromise in quality. This method must be distinguished from the bit truncation method, which sets both the carry and sum outputs to 0.

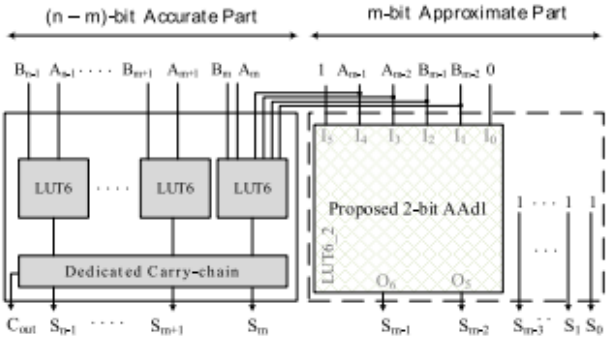

Fig. 3. n-bit area & power efficient

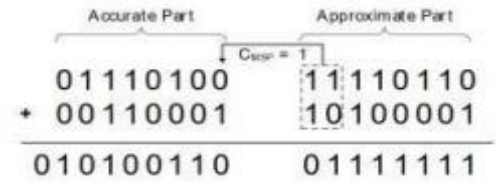

Fig. 4. Example

4. Proposed Method

The proposed method introduces a High-Performance Error-Tolerant Adder (ETA), designed to enhance area efficiency, power consumption, and computational speed in FPGA-based applications Fig.5. Unlike traditional hybrid approximate adders (such as LEADx and APEx), which focus on reducing power and area while maintaining low error rates, the proposed ETA adder prioritizes error tolerance and hardware simplicity, achieving significant reductions in logical power, dynamic power, and area utilization (LUTs).

- The ETA adder is designed by segmenting the addition process into two parts:
- Most Significant Part (MSP): Performs approximate addition to reduce computational complexity.
- Least Significant Part (LSP): Uses an error-tolerant mechanism to maintain acceptable accuracy.

This approach helps to achieve a tradeoff between accuracy and efficiency, making the ETA adder particularly suitable for FPGA-based applications where minor computation errors are acceptable in exchange for better power and area optimization shown in Fig.6

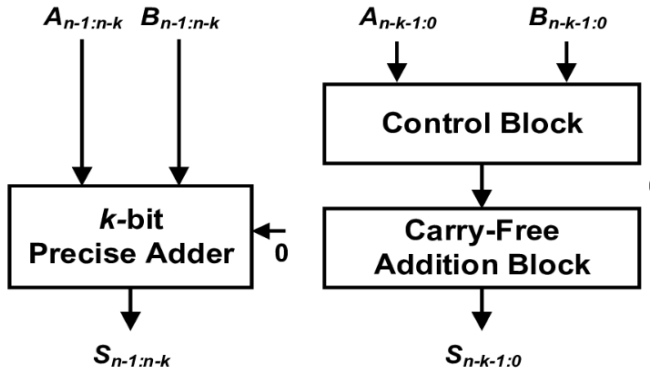

Fig. 5. Propsoed method Architecture

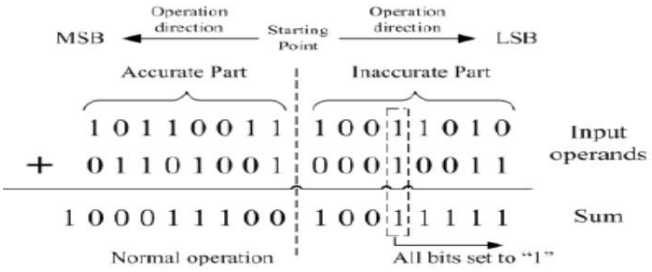

Fig. 6. Example for propsoed method

The methodology for designing and implementing the High-Performance Error-Tolerant Adder (ETA) focuses on optimizing area, power consumption, and computational efficiency for FPGA-based applications. The process consists of multiple stages, including problem definition, adder design, FPGA implementation, and performance evaluation.

Problem Definition

The creation of an area-optimized and energy-efficient ETA adder for use in FPGA-based applications is the primary objective of this work.. The main limitations of existing approximate adders (such as LEADx and APEx) include:

- High logical power consumption (0.098W)
- Excessive LUT utilization (20 LUTs)
- Increased total power consumption (5.117W)

To overcome these challenges, the proposed ETA adder is designed to:

Design of the ETA Adder

The design process of the ETA adder involves partitioning the addition process to optimize power and area efficiency Fig.7. The addition is split into two parts:

1. Most Significant Part (MSP) – Exact Addition:

o The MSB segment is computed accurately to ensure minimal error impact on the final result.

o This preserves accuracy in critical computations while maintaining computational efficiency.

2. Least Significant Part (LSP) – Approximate Addition:

o The LSB segment is computed using an approximation method to reduce power consumption and circuit complexity.

o This results in lower delay and energy usage, making the adder more power-efficient.

This hybrid approach maintains a balance between accuracy and hardware efficiency, making the ETA adder ideal for FPGA-based implementations.

FPGA Implementation Flow

The proposed ETA adder is implemented on an FPGA platform using the following steps:

Step 1: HDL Design (Verilog)

- The ETA adder architecture is designed using VHDL/Verilog.
- The exact (MSP) and approximate (LSP) computation units are coded separately for modularity.

Step 2: Functional Simulation

- The design is tested using functional simulations in ModelSim/Xilinx Vivado to verify:
  - Correct adder functionality
  - Bit-wise accuracy in computation
  - Error impact on different applications

Step 3: FPGA Synthesis and Implementation

- The hardware synthesis is performed using Xilinx Vivado/Quartus Prime.
- The area utilization (LUTs), power consumption, and timing analysis are obtained from FPGA reports.

Step 4: Power and Area Optimization

- Techniques such as logic folding, pipelining, and resource sharing are applied to further optimize LUT utilization and power consumption.
- The proposed design is compared with existing adders (LEADx and APEx) to measure improvements.

Performance Evaluation Metrics

To validate the effectiveness of the ETA adder, the following performance metrics are evaluated:

1. Area Utilization (LUTs)

- Measures the number of Look-Up Tables (LUTs) required for hardware implementation.
- The ETA adder achieves a 65% reduction in LUT utilization compared to LEADx/APEx.

Power Consumption

- The logical, static, dynamic, and total power consumption are analyzed using Xilinx Power Analyzer.
- The ETA adder reduces logical power by 73% and total power by 28.6%.

Accuracy Evaluation

- The computational accuracy of the ETA adder is compared with LEADx and APEx by measuring the Mean Absolute Error (MAE).
- The design is optimized to ensure minimal impact on real-world applications.

Delay and Speed Performance

- Propagation delay and clock frequency are analyzed using timing analysis tools.
- The ETA adder improves computation speed by reducing unnecessary logic gates.

Implementation

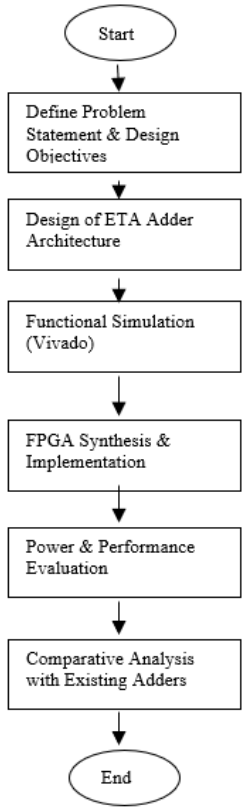

Fig. 7. n-bit area & power efficient

5. Simulation Results

Existing Method

The simulation waveform for the existing method (LEADx and APEx adders) demonstrates its functionality Fig.8. The output sum and carry signals are generated based on the input

values, showing the operation of the approximate addition process.

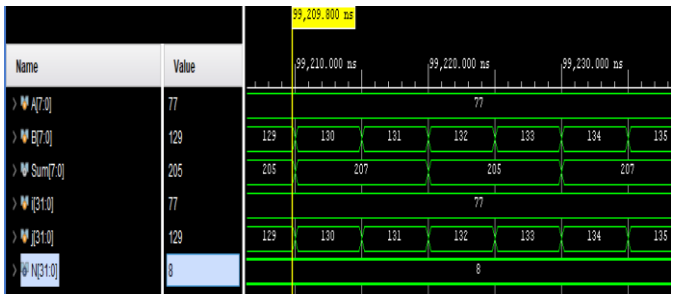

Fig. 8. Simulation Result for existing method

The RTL (Register Transfer Level) schematic provides a hardware-level representation of the existing adder Fig.9. The design consists of a large number of logic gates and interconnections, leading to higher area utilization (20 LUTs).

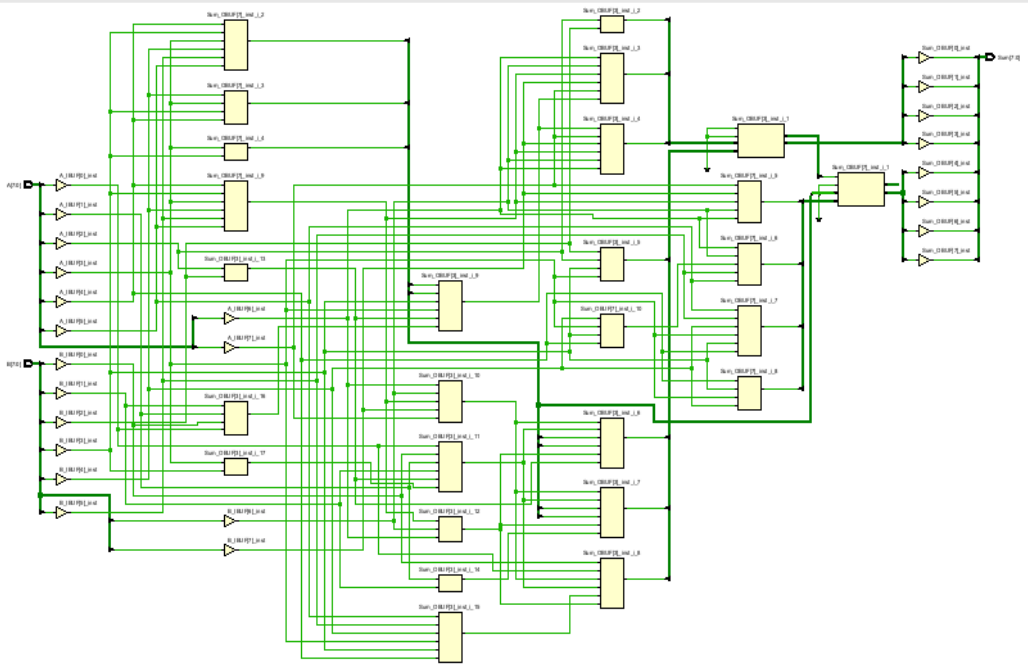

Fig. 9. RTL Schematic for Existing method

The power consumption analysis (Fig.10) is obtained using Xilinx Power Analyzer. The following power metrics are observed: Static Power: 0.117W Logical Power: 0.098W Dynamic Power: 5.060W Total Power Consumption: 5.117W The high logical and dynamic power consumption is due to larger logic complexity.

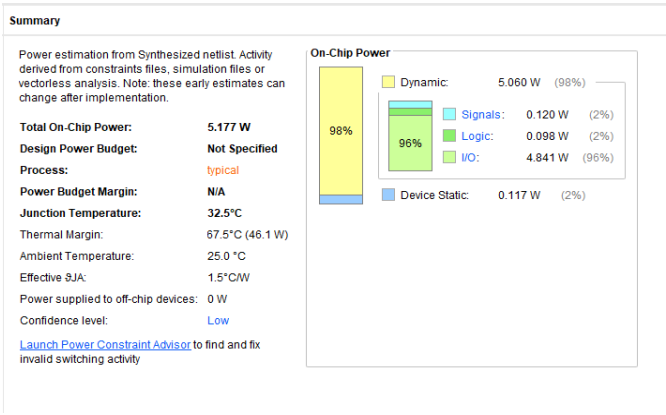

Fig. 10. Power utilization of existing method

The area report Fig. 11 shows that the existing adder uses 20 LUTs.The higher area consumption affects FPGA resource efficiency.

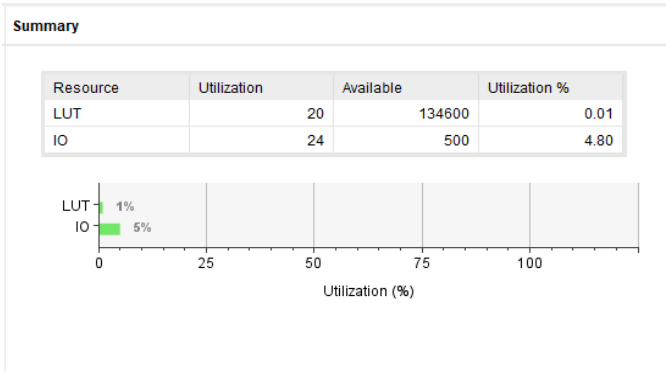

Fig. 11. Area utilization of existing method

Proposed Method

The simulation waveform for the proposed ETA adder Fig.12 demonstrates correct functionality with reduced complexity. The output signals indicate proper approximate addition with lower error impact.

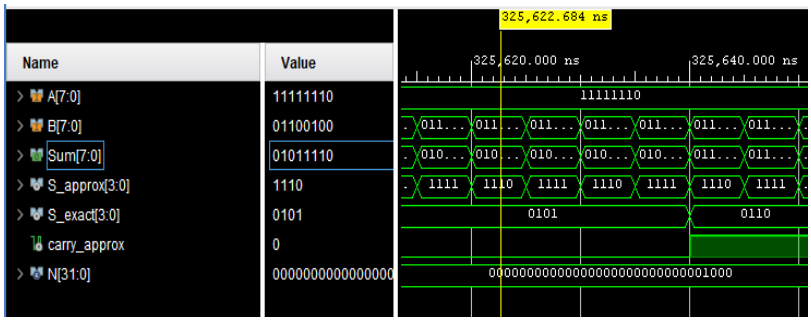

Fig. 12. Simulation result of Proposed method

The RTL schematic of the ETA adder Fig.13 shows a simplified logic structure compared to the existing method. The reduced gate count contributes to lower area utilization (7 LUTs) and optimized FPGA resource usage.

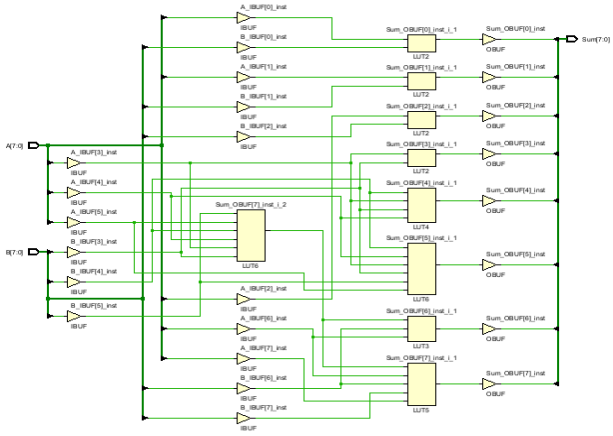

Fig. 13. RTL Schematic for Propsoed method

The ETA adder significantly reduces power consumption:

- Static Power: 0.114W (Slightly improved)
- Logical Power: 0.026W (73% lower than the existing method)
- Dynamic Power: 3.537W (30% lower than existing method)
- Total Power Consumption: 3.651W (28.6% reduction)
- The lower logical and dynamic power is due to optimized logic partitioning in ETA shown in Fig.14.

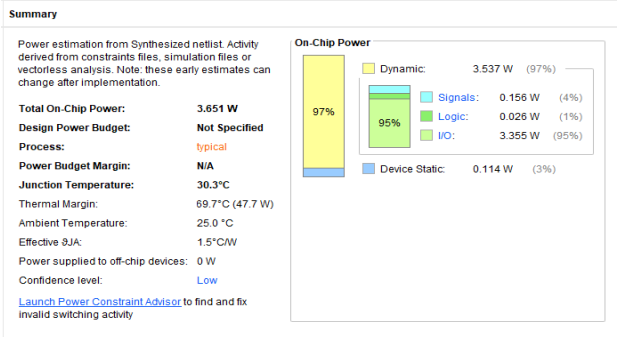

Fig. 14. Power utilization of Proposed method

The area report Fig.15 shows that the proposed adder uses only 7 LUTs, compared to 20 LUTs in the existing method. The 65% reduction in LUT utilization improves FPGA scalability and efficiency.

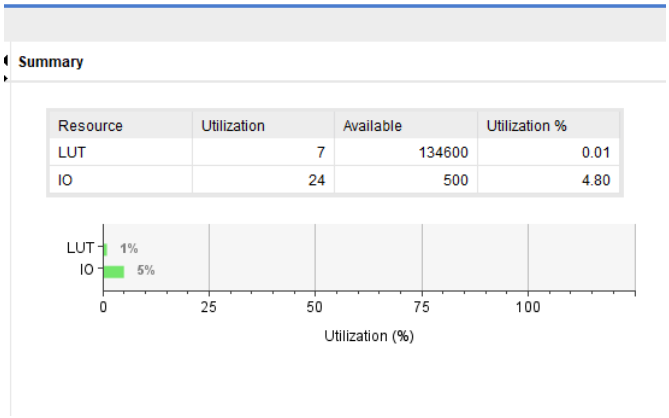

Fig. 15. Area utilization of existing method

Performance Comparision Table

TABLE I. PERFORMANCE COMPARISION OF EXISTING AND PROPOSED METHOD

S.N	Parameter	Existing method	Proposed method	Utilization (%)
1	Area (LUTs)	20	7	70 % Lower
2	Static Power (W)	0.117	0.114	-
3	Logical Power (W)	0.098	0.026	73% Lower
4	Dynamic Power (W)	5.060	3.537	30% Lower
5	Total Power (W)	5.117	3.651	28.6% Lower

6. Conclusion and Future scope

In this work, a high-performance Error-Tolerant Adder (ETA) for FPGA-based applications has been designed and implemented. The proposed ETA adder significantly reduces power consumption and area utilization compared to the existing hybrid approximate adders (LEADx and APEx). Through FPGA synthesis and simulation, the following key improvements were observed:65% reduction in area utilization (LUTs) – from 20 LUTs to 7 LUTs. 73% lower logical power consumption – from 0.098W to 0.026W.30% reduction in dynamic power – from 5.060W to 3.537W.28.6% decrease in total power consumption – from 5.117W to 3.651W. These optimizations make the ETA adder an efficient solution for FPGA-based applications that require low power and high performance while tolerating small computational errors. The proposed method is suitable for applications in signal processing, neural networks, and image processing because it achieves an excellent balance between accuracy and hardware efficiency.

Future scope

In future the proposed method can be extended with Using AI/ML algorithms to optimize approximate adder designs for different workloads.

References

1. G. A. Gillani, M. A. Hanif, B. Verstoep, S. H. Gerez, M. Sha_que, and A. B. J. Kokkeler, ``MACISH: Designing approximate MAC accelerators with internal-self-healing," IEEE Access, vol. 7, pp. 77142_77160, 2019.
2. E. Kalali and I. Hamzaoglu, ``An approximate HEVC intra angular prediction hardware," IEEE Access, vol. 8, pp. 2599_2607, 2020.
3. T. Ayhan and M. Altun, ``Circuit aware approximate system design with case studies in image processing and neural networks," IEEE Access.
4. W. Ahmad and I. Hamzaoglu, ``An ef_cient approximate sum of absolute differences hardware for FPGAs," in Proc. IEEE Int. Conf. Consum.Electron. (ICCE), Las Vegas, NV, USA, Jan. 2021, pp. 1_5.
5. H. Jiang, C. Liu, L. Liu, F. Lombardi, and J. Han, ``A review,classification,and comparative evaluation of approximate arithmetic circuits,'ACM J. Emerg. Technol. Comput. Syst., vol. 13, no. 4, pp. 1_34, Aug.2017.
6. A. C. Mert, H. Azgin, E. Kalali, and I. Hamzaoglu, ``Novel approximate absolute difference hardware," in Proc. 22nd Euromicro Conf.Digit. Syst.Design (DSD), Kallithea, Greece, Aug. 2019, pp.190_193.
7. N. Van Toan and J.-G. Lee, ``FPGA-based multi-level approximate multipliers for highperformance error-resilient applications,"IEEE Access,vol. 8, pp. 25481_25497, 2020.
8. L. Chen, J. Han, W. Liu, P. Montuschi, and F. Lombardi, ``Design, evaluation and application of approximate high-radix dividers,"IEEE Trans.Multi-Scale Comput. Syst., vol. 4, no. 3, pp. 299_312, Jul. 2018.
9. A. K. Verma, P. Brisk, and P. Ienne, ``Variable latency speculative addition:A new paradigm for arithmetic circuit design," in Proc. Design, Automat.Test Eur. (DATE), Munich, Germany, Mar. 2008, pp. 1250_1255.
10. N. Zhu, W. L. Goh, G. Wang, and K. S. Yeo,``Enhanced low-power high-speed adder for errortolerant application," in Proc. Int. SoC Design Conf.,Incheon, South Korea, Nov. 2010, pp. 323_327.
11. A. B. Kahng and S. Kang, ‘‘Accuracy-configurable adder for approximate arithmetic designs,’’ in Proc. 49th Annu. Design Automat. Conf. (DAC), New York, NY, USA, 2012, pp. 820–825. [12] V. Gupta, D. Mohapatra,
12. A. Raghunathan, and K. Roy, ‘‘Low-power digital signal processing using approximate adders,’’ IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 32, no. 1, pp. 124–137, Jan. 2013.
13. W. Ahmad, B. Ayrancioglu, and I. Hamzaoglu, ‘‘Comparison of approximate circuits for H.264 and HEVC motion estimation,’’ in Proc. 23rd Euromicro Conf. Digit. Syst. Design (DSD), Kranj, Slovenia, Aug. 2020, pp. 167–173.
14. H. R. Mahdiani, A. Ahmadi, S. M. Fakhraie, and C. Lucas, ‘‘Bio-inspired imprecise computational blocks for efficient VLSI implementation of softcomputing applications,’’ IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 57, no. 4, pp. 850–862, Apr. 2010.
15. A. Dalloo, A. Najafi, and A. Garcia-Ortiz, ‘‘Systematic design of an approximate adder: The optimized lower part constant- OR adder,’’ IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 26, no. 8, pp. 1595–1599, Aug. 2018.
16. P. Balasubramanian, R. Nayar, D. L. Maskell, and N. E. Mastorakis, ‘‘An approximate adder with a near-normal error distribution: Design, error analysis and practical application,’’ IEEE Access, vol. 9, pp. 4518–4530, 2021.
17. J.V.Preethi, G. N. Kodandaramaiah, and K.Rasadurai, "FPGA Implementation of 256 Bit Key AES Algorithm using A Modified S-BOX Sharing Algorithm", 1st Edition Security Issues in Communication Devices, Networks and Computing ModelsVolume 2, ISBN 9781032970271.
18. K. Revathi, K.Rasadurai, "An Improved Noise Filter Design for Canny Edge Detection with Pipeline Architecture", 1st Edition Security Issues in Communication Devices, Networks and Computing ModelsVolume 2, ISBN 9781032970271.
19. Jenifer.J K. Rasadurai, "Enhanced on-chip Memory Reliability 2 Demensional Error Detection & Correction scheme",6th IEEE International Conference on Systems Computation, Automation and Networking, Volume. 1, Issue.1,Pages.92, 28.012.2024, ICSCAN 2024 Pondicherry.