TABLE OF CONTENTS

ABSTRACT

The importance of high-speed adders in digital systems cannot be overstated, as these components play a pivotal role in shaping the efficiency and performance of various applications. This abstract explores the significance of fast adders in the realm of digital signal processing, real-time computing, and communication systems.

Fast adders contribute to enhanced system performance by reducing computation times, enabling efficient parallel processing, and improving overall throughput. Their role becomes particularly crucial in applications requiring real-time processing, such as control systems and multimedia applications. Additionally, the abstract discusses how fast adders contribute to energy efficiency by minimizing active states in electronic circuits.

By delving into these aspects, the abstract underscores the broad impact of high-speed adders across diverse domains, emphasizing their indispensable role in meeting the demands of modern computing applications. High-speed adders play a crucial role in digital signal processing (DSP) for several reasons. DSP involves performing mathematical operations on signals to analyze, modify, or extract information.

CHAPTER 1

INTRODUCTION

1.1 Overall System/Scenario

Digital Signal Processors (DSPs) are specialized microprocessors designed to perform mathematical operations on digital signals efficiently. These operations include addition, subtraction, multiplication, and other transformations that are fundamental to processing audio, video, and communication data in real-time. High-speed adders are a critical component within DSPs because they play a pivotal role in performing arithmetic operations on digital signals.

In summary, high-speed adders are at the heart of DSPs, enabling them to efficiently process digital signals in real-time across a wide range of applications, from audio and video processing to wireless communication. Their speed, accuracy, are essential for delivering high-quality, responsive, and reliable signal processing in modern electronics and communication systems.

At present, the research continues on increasing the adder's delay performance. In many practical applications like mobile and telecommunications, the Speed and power performance improved in FPGAs is better than microprocessor and DSP's based solutions. Additionally, power is also an important aspect in growing trend of mobile electronics, which makes large-scale use of DSP functions. Because of the Programmability, structure of configurable logic blocks (CLB) and programming interconnects in FPGAs.

Major Issues Faced

When approximate parallel prefix adders are not employed, traditional, precise adders (such as ripple-carry or carry-lookahead adders) are typically used. Here are some major issues faced without having approximate parallel prefix adders:

- Traditional, slow adders can lead to slow signal processing speeds, limiting the system's ability to handle real-time requirements in applications such as audio processing, telecommunications or image processing. (Slow Signal Processing)

- Slow adders contribute to increased latency in arithmetic operations. This delay can be critical in real-time applications where timely responses are essential. (Increased Latency)
- Traditional adders might not fully exploit parallelism, leading to inefficient use of hardware resources. This limitation can result in suboptimal performance, especially in applications where parallel processing is beneficial. (Limited Parallelism)

- Slow adders may lead to increased power consumption, especially if the system needs to operate at a higher frequency to compensate for slow arithmetic operations. This can be undesirable in applications with strict power constraints. (Power Inefficiency)

- Slow adders may have difficulty adapting to dynamic workloads, especially in scenarios where computational demands vary. Real-time systems often require flexibility in handling changing workloads efficiently. (Challenges in Handling Dynamic Workloads)

- In real-time applications, particularly those dealing with streaming data, buffer overflows can result in data loss or system instability.(Risk of Buffer Overflows)

  Addressing these challenges may involve the adoption of high-speed adders or other optimized arithmetic units that better align with the stringent requirements of real-time applications

Ref : *Parallel prefix adders have better performance. S. K. Yezerla and B. Rajendra Naik, "Design and estimation of delay, power and area for Parallel prefix adders," 2014 Recent Advances in Engineering and Computational Sciences (RAECS), Chandigarh, India.*

M. M. A. d. Rosa, G. Paim, P. Ü. L. d. Costa, E. A. C. d. Costa, R. I. Soares and S. Bampi, "AxPPA: Approximate Parallel Prefix Adders," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 31, no. 1.

U. Penchalaiah and S. K. VG, "Design of High-Speed and Energy-Efficient Parallel Prefix Kogge Stone Adder," 2018 IEEE International Conference on System, Computation, Automation and Networking (ICSCA), Pondicherry, India.

A. K. Panda, R. Palisetty and K. C. Ray, "High-Speed Area-Efficient VLSI Architecture of Three-Operand Binary Adder," in IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 67.

1.2 Objectives

To design an approximated high-speed adder with improved performance in terms of speed, power dissipation and area occupancy in comparison with others. Analyzing the performance of various parallel prefix adders on the basis of power and computational delay.

Sub-Objectives:

- Research and Design Selection : To review on high-speed adders and evaluate the suitable semiconductor technologies, adder architectures and components for the intended application.
- Gate Level Optimization and Comparative/Trade-off analysis.
- Find the error rate of the obtained result.

1.3 Methodology

Conducting a review of the existing literature focused on the choice of approximate high-speed adder designs, followed by the utilization of software tools like Verilog or VHDL to simulate these designs. This simulation phase will involve optimizing the designs for either speed or area, and subsequently, assessing the outcomes through comprehensive simulations and comparative analyses.

The methodology involves configuring AxPOs for approximating specific parts of the parallel prefix adder, utilizing a combination of approximate and exact computing to achieve a balance between accuracy and efficiency.

The architecture and operation are demonstrated through specific examples and algorithms, and the proposed architectures provide flexibility in configuring the trade-off between accuracy and resource utilization.

The final step involves the practical validation of the chosen adder designs by implementing them on FPGA boards to accurately evaluate their performance in a hardware setting.

1.4 Expected Result

To propose a high-speed area-efficient adder technique and its VLSI architecture to perform the three operand binary addition for efficient computation of modular arithmetic used in image processing or audio processing applications. The proposed three-operand adder technique is a parallel prefix adder that uses four-stage structures to compute the addition of three input operands. The novelty of this proposed architecture is the reduction of delay and power in the prefix computation stages in PG logic and bit-addition logic that leads to an overall reduction in critical path delay and power-delay product (PDP).

Reduction in Delay in Prefix Computation:

The innovation lies in a systematic approach to minimize delay in the prefix computation stages, specifically in the PG logic. By employing advanced optimization techniques and efficient circuit configurations, the proposed architecture mitigates delays traditionally associated with parallel prefix adders.

Optimized Bit-Addition Logic:

The bit-addition logic is optimized for reduced power consumption without compromising speed. This is achieved through the implementation of advanced circuit design methodologies, resulting in a more energy-efficient and high-performance architecture.

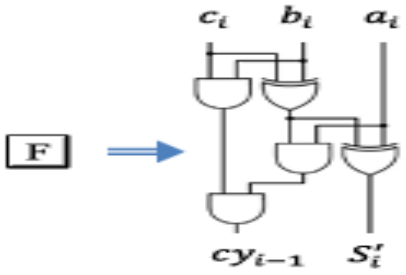

Fig : *Optimized Bit-Addition*

Overall Reduction in Critical Path Delay and PDP:

The combined effect of optimizing both PG logic and bit-addition logic is a substantial reduction in the critical path delay. Consequently, this leads to a lower power-delay product (PDP), enhancing the overall efficiency and speed of the three-operand binary addition.

Comparative Analysis: Brent Kung vs. Kogge Stone

A detailed analysis is conducted on power and delay characteristics of the Brent Kung adder and Kogge Stone adder. Simulations and synthesis reports are employed to provide a comprehensive understanding of the performance trade-offs among the three architectures.

Efficiency in DSP Applications:

The suitability of each adder for implementation in a digital signal processor is evaluated based on the specific requirements of DSP applications. This includes considerations for power constraints, real-time processing demands, and overall computational efficiency. The study aims to identify the most optimal three-operand binary adder for DSP applications, considering both power and delay characteristics. Insights gained from the comparative analysis will guide designers in selecting the most suitable adder architecture for their specific requirements.

1.5 Time Plan

Month	**Module of work to be completed**
August 2023	Research and Design Selection
September 2023	Research and Design Selection September 2023 Preliminary Design and Simulation
October 2023	Optimization and Comparative Analysis
November 2023	Refinement and Validation
December 2023	Documentation and Presentation

CHAPTER 2

STATE OF THE ART

2.1 Introduction

Computational speed of basic data path elements such as adders plays a crucial role in performance of digital and signal processing systems. Objective is to design a fast adder after analyzing the performance of various parallel prefix adders on the basis of power and computational delay. These studies give an insight to do the same.

2.2 Basic background with few citations

The base paper explores the use of different Parallel-Prefix adders (PPAs) in the precise part of approximate adders. It compares various PPAs, including Brent-Kung, Kogge-Stone, Han Carlson, Ladner-Fisher, and Sklansky. The studies are generally conducted on 16-bit, 24-bit, and 32-bit approximate adders synthesized to 45 nm CMOS technology. Results indicate that PPA adders can improve clock frequency significantly while maintaining a reasonable energy consumption overhead.

The focus is on the synthesis of approximate parallel-prefix adders, providing a wide range of solutions with trade-offs in delay, area, and error. The automatic design space exploration leads to optimal solutions not achievable with known architectures.Comparisons with state-of-the-art adders show that the synthesized adders achieve better error frequency for random inputs and improve image quality metrics for image filtering. The proposed methodology also calculates arithmetic error rates for deterministic approximate adder architectures with restricted input bit calculations.

The paper introduces Approximate Parallel Prefix Adders (AxPPAs) by incorporating approximations in the prefix operators. Four AxPPA architectures are presented, namely AxPPA-BK, AxPPA-KS, AxPPA-LF, and AxPPA-SK, compared with energy-efficient approximate adders in stand-alone cases and embedded in signal processing application kernels. In summary, the research explores novel

approaches to approximate adders, considering different PPAs and introducing AxPPAs with promising results in terms of energy, quality, and area trade-offs.

Name of the Paper	Citation
Exploring the use of parallel prefix adder topologies into approximate adder circuits.	*Morgana Macedo†, Leonardo Soares*, Bianca Silveira†, Cla´udio M. Diniz†, Eduardo A. C. da Costa† †Graduate Program on Electronic Engineering and Computing - Catholic University of Pelotas (UCPel), Pelotas, Brazil*
Synthesis of Approximate Parallel-Prefix Adders	*Apostolos Stefanidis , Ioanna Zoumpoulidou, Dionysios Filippas , Giorgos Dimitrakopoulos , Member, IEEE, and Georgios Ch. Sirakoulis , Member, IEEE*
AxPPA: Approximate Parallel Prefix Adders	*Morgana Macedo Azevedo da Rosa , Student Member, IEEE, Guilherme Paim , Member, IEEE, Patrícia Ücker Leleu da Costa , Student Member, IEEE, Eduardo Antonio Ces´ar da Costa , Member, IEEE, Rafael I. Soares , Member, IEEE, and Sergio Bampi , Senior Member, IEEE*
Efficient Arithmetic Error Rate Calculus for Visibility Reduced Approximate Adders	*J. Echavarria, S. Wildermann, E. Potwigin and J. Teich, "Efficient Arithmetic Error Rate Calculus for Visibility Reduced Approximate Adders," in IEEE Embedded Systems Letters, vol. 10, no. 2, June 2018.*

Design and analysis of High speed wallace tree multiplier using parallel prefix adders for VLSI circuit designs	*Y. d. Ykuntam, K. Pavani and K. Saladi, "Design and analysis of High speed wallace tree multiplier using parallel prefix adders for VLSI circuit designs," 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Kharagpur, India, 2020, pp. 1-6, doi: 10.1109/ICCCNT49239.2020.9225404.*

2.2 Literature Survey

2.2.1 Exploring the use of parallel prefix adder topologies into approximate adder circuits.

Morgana Macedo†, Leonardo Soares, Bianca Silveira†, Cla´udio M. Diniz†, Eduardo A. C. da Costa† †Graduate Program on Electronic Engineering and Computing - Catholic University of Pelotas (UCPel), Pelotas, Brazil*

Objective:

- To explore different Parallel Prefix adders (PPA) topologies in the precise part of approximate adders .
- To investigate the Trade Off between area, performance and power dissipation in the approximate adders for different adder topologies in the precise part.

Methodology:

- Use of Brent-Kung, Kogge-Stone, Han Carlson, Ladner-Fisher, and Sklansky PPAs in the precise part of two well-known approximate adders

➢ Copy adder :

- Instead of using full adder cells and carry chain propagation scheme in the imprecise part (least significant bits), the simplification is performed by adopting buffers.

- Error Tolerant Adder I (ETA-I) :

- It consists of Half-adder and generate

operator for each bit position.

- The precise part is implemented without

any approximation technique.

Inference:

- This work proposed the exploration among different adder topologies to implement the precise part of state-of-the-art approximate adders.
- The Area analysis states that ETA-I approximate adder generally exhibited higher area than the Copy adder.
- On Analyzing the synthesis of different adder topologies Kogge Stone adder has the lowest delay because of immediate carry computations with a delay complexity of $(\log 2\, n)$.

Future Work:

- Further research could focus on enhancing the energy efficiency of the Kogge-Stone adder or other high-performance adder designs.
- This could involve exploring novel techniques, such as power gating, voltage scaling, or dynamic power management strategies, to mitigate the energy consumption overhead associated with these high-speed adder architectures.

2.2.2 Synthesis of Approximate Parallel-Prefix Adders

Apostolos Stefanidis , Ioanna Zoumpoulidou, Dionysios Filippas , Giorgos Dimitrakopoulos , Member, IEEE, and Georgios Ch. Sirakoulis , Member, IEEE

Objective:

- To automatically synthesize all possible approximate parallel-prefix adders, given a maximum allowed carry-chain length, the available number of prefix levels as well as the maximum allowed internal fan-out.

- Generate many candidate prefix graph structures for a given set of constraints that can be evaluated for their performance in physical implementation and numerical accuracy.
- Synthesize multiple forms of approximate adders that compare favourably to state-of- the-art adders including also split-accuracy alternatives.

Methodology:

The proposed approach for synthesizing approximate parallel-prefix adders extends the enumerating approach to include also approximate solutions that satisfy the given design constraints.

- The evaluation of the proposed approach is presented in four steps. In the first step, the relation between the designer's constraints and the number of derived solutions as well as the structure of the best choices is investigated.
- In the second step, the proposed approach to state-of-the-art is compared in terms of numerical accuracy on random inputs and hardware complexity.
- Next, the accuracy of the proposed adders on an image filtering application and a neural-network classifier is evaluated.
- In the last step, split-accuracy adders synthesized by the proposed method are compared to relevant state-of-the-art method for hardware complexity and accuracy for the same benchmark applications.

Inference:

- Parallel-prefix computation of addition offers a versatile framework for designing adders of various area–power–delay characteristics.
- Generalised the synthesis of parallel-prefix adders to include also approximate parallel-prefix trees that compute carry chains of shorter length, under strict maximum fan-out and maximum prefix level constraints.

- The proposed synthesis approach allows to derive new approximate parallel-prefix adders that have not appeared in open literature supporting uniform or split accuracy.
- Most importantly, the introduced synthesis engine is avail- able under a permissive open-source license, which allows end-users to test it and extend it for future research.

Future Work:

Future work is planned to explore the interplay between approximation criteria and how they affect the characteristics of the synthesized parallel-prefix adders under process– voltage–temperature (PVT) variations

2.2.3 AxPPA: Approximate Parallel Prefix Adders

Morgana Macedo Azevedo da Rosa , Student Member, IEEE, Guilherme Paim , Member, IEEE, Patrícia Ücker Leleu da Costa , Student Member, IEEE, Eduardo Antonio Ces´ar da Costa , Member, IEEE, Rafael I. Soares , Member, IEEE, and Sergio Bampi , Senior Member, IEEE

Objective:

- To propose and evaluate the performance of approximate parallel prefix adders (AxPPAs) by introducing approximate prefix operators (AxPOs) in the architecture.
- To Compare the proposed AxPPA architectures with energy-efficient approximate adders (AxAs), such as Copy, lower-part OR adder (LOA), and Truncation (trunc), in terms of energy consumption, area utilization, and overall performance.
- To Integrate the AxPPAs into two critical signal processing application kernels, namely, the sum of squared differences (SSDs) video accelerator and the finite impulse response (FIR) filter kernel, to assess their impact on overall system performance and quality of results.

Methodology:

- The AxPPA is designed to perform approximate addition by approximating certain bits, which reduces the complexity and area of the computation.
- This AxPPA is divided into 3 steps:

1. Preprocessing
2. Approximate prefix computing
3. Postprocessing

- The approximate prefix computing only uses wires to connect values of generate and propagate carry of the preprocessing step to the post-processing step.

Inference:

- We have inferred the methodology used in the paper for approximating the parallel prefix adders by completely eliminating the Prefix Operators and further optimizing the prefix operations.
- Generalized synthesis results of different AxPPA and comparing them with PPAs based on many metrics such as performance, area and power.

Future Work:

- To develop error detection and correction mechanisms to ensure the reliability and precision of the computed results.
- To conduct hardware implementation and testing of the AxPPA to evaluate its performance in real-world scenarios.

2.2.4 Efficient Arithmetic Error Rate Calculus for Visibility Reduced Approximate Adders

J. Echavarria, S. Wildermann, E. Potwigin and J. Teich, "Efficient Arithmetic Error Rate Calculus for Visibility Reduced Approximate Adders," in IEEE Embedded Systems Letters, vol. 10, no. 2, June 2018.

Objective:

- To calculate the arithmetic error rate for deterministic approximate adder architectures for any input distribution i.e also nonuniformly distributed inputs, to demonstrate that the proposed technique has a computational complexity of O(n^2).
- To present the calculus for enabling Generic Algorithms GA based exploration of approximate adder architectures
- To Compare the Parallel Prefix Adder and the Carry Propagate Adder (involves examining the interaction of each output bit with previous inputs)

Methodology:

Given a few definitions to ease the understanding of our technique.

1. Definition 1 (Input Distributions)
2. Definition 2 (Carry Generation and Propagation)
3. Definition 3 (Visibility)
4. Definition 4 (Bit Error Rate)

Inference:

- We have inferred the methodology used in the paper for approximating the parallel prefix adders by completely eliminating the Prefix Operators and further optimizing the prefix operations.

- Generalized synthesis results of different AxPPA and comparing them with PPAs based on many metrics such as performance, area and power.

Future Work:

- To develop error detection and correction mechanisms to ensure the reliability and precision of the computed results.
- To conduct hardware implementation and testing of the AxPPA to evaluate its performance in real-world scenarios.

2.2.5 Design and analysis of High speed wallace tree multiplier using parallel prefix adders for VLSI circuit designs

Objective:

- Propose a new structure for the Wallace tree multiplier using parallel prefix adders (PPAs) to improve the speed of the multiplication process without compromising on the area parameter.
- Design and analyse five different Wallace tree multiplier structures using five different PPAs - Kogge stone adder, Sklansky adder, Brent Kung adder, Han Carlson adder, and Ladner Fischer adder.
- Compare the area (number of LUTs) and delay (ns) of the proposed multiplier structures with traditional multiplier design.

Methodology:

- Traditional Wallace Tree Multiplier: The operation of the traditional Wallace tree multiplier is explained, detailing the two phases of the multiplication process and the role of adders in generating final product terms.
- Design and Analysis: Five different Wallace tree multiplier structures using five different PPAs - Kogge stone adder, Sklansky adder, Brent Kung adder, Ladner Fischer adder, and Han Carlson adder - are designed and analyzed

using Verilog HDL in Xilinx 13.2 design suite. The simulation waveforms for the proposed

Inference:

- The paper suggests that the proposed multiplier structures using PPAs offer improved speed and, to some extent, reduced area compared to the traditional multiplier.
- The results of the proposed multiplier structures are analyzed in terms of area (number of LUTs) and delay (ns) compared to the traditional multiplier design.

Future scope:

- Designing multiplier structures with higher input sizes, such as 32-bit or 64-bit. By increasing the input size, it is anticipated that multiplier structures with fewer LUTs and better delay values can be achieved.

CHAPTER 3

DESIGN AND ANALYSIS

3.1 Introduction

In this chapter we will be discussing the design concept of Approximate Parallel Prefix Adders (AxPPAs). It introduces the concept of approximate computing (AxC) as a new design paradigm for increasing efficiency across the computing stack. Our base paper proposes AxPOs (Approximate Prefix Operators) logic to synthesize two different AxPPA architectures and evaluates their performance. The study demonstrates that AxPPAs offer significant improvements in energy and area efficiency compared to existing energy-efficient AxAs (Approximate Adders), and presents a new Pareto front in the trade-off between quality and hardware synthesis results when benchmarked against the most energy-efficient AxAs in the literature. Generally, approximate high-speed adders involve parallel processing of input bits and carry propagation techniques to achieve faster addition.

- Each architecture has its own set of logic blocks and circuitry arrangements that aims to optimize speed and minimize delays during addition operations.
- The project would involve understanding, implementing, and comparing these different architectures to determine their effectiveness in achieving high-speed

  The design and analysis of Approximate Parallel Prefix Adders (AxPPAs) represent a pioneering approach in this context.

  Background:

  Traditional adder designs prioritize precision, ensuring accurate computation of sums. However, in many applications, especially those tolerant to minor errors, sacrificing precision for gains in terms of power, area, and speed becomes a compelling consideration. Approximate computing leverages this trade-off, allowing for the exploration of adder architectures that intentionally introduce controlled inaccuracies.

  Motivation:

The motivation behind designing AxPPAs lies in the quest for more power-efficient and area-optimized digital systems. By strategically incorporating approximation techniques, designers aim to achieve computational efficiency while meeting the acceptable error tolerance for specific applications. This approach is particularly relevant in scenarios where absolute precision is not critical, such as in machine learning or multimedia processing.

Design Principles:

AxPPAs are designed with a focus on configurable approximations. Parameters like the number of approximated bits (K) and the choice of AxPO configurations are critical design choices. The architecture involves a combination of exact and approximate computing components, enabling flexibility in tailoring the adder's behavior to match the precision requirements of diverse applications.

Analysis and Performance Metrics:

The analysis of AxPPAs involves assessing their performance against traditional adder architectures. Key metrics include delay, gate area, power consumption, and the impact on accuracy. Designers must strike a balance between achieving acceptable accuracy levels and optimizing resource utilization. Comparative studies with exact adders and across various AxPPA configurations provide insights into the trade-offs involved.

3.2 System Architecture

The methodology and architecture of an Approximate Parallel Prefix Adder (AxPPA) along with the associated Approximate Prefix Operators (AxPOs).

1. **General Architecture Overview:**

The AxPPA is designed for parallel prefix addition with a focus on approximate computing. The precision of the adder is configurable at design time using a parameter K, representing the number of approximated bits. The AxPPA is divided into an approximate part (AxPOs) and an exact part. Two specific architectures (AxPPA-BK, AxPPA-KS) are proposed, each with its own design characteristics.

2. **AxPO (Approximate Prefix Operator) Configuration:**

The AxPOs are used to approximate the logic of a subset of the Partial Products (POs) in the parallel prefix adder. The number of approximate POs can be configured using the parameter K. AxPOs use a combination of wires for processing the prefix computing, connecting preprocessing to postprocessing.

3. **AxPPA Operation:**

The AxPPA processes W-bit inputs, where W is the size of the words input. The AxPPA comprises both an approximate part (AxPO-based) and an exact part (utilizing an exact adder structure). The AxPPA-BK, AxPPA-KS architectures provide variations in how the AxPOs are applied within the adder structure.

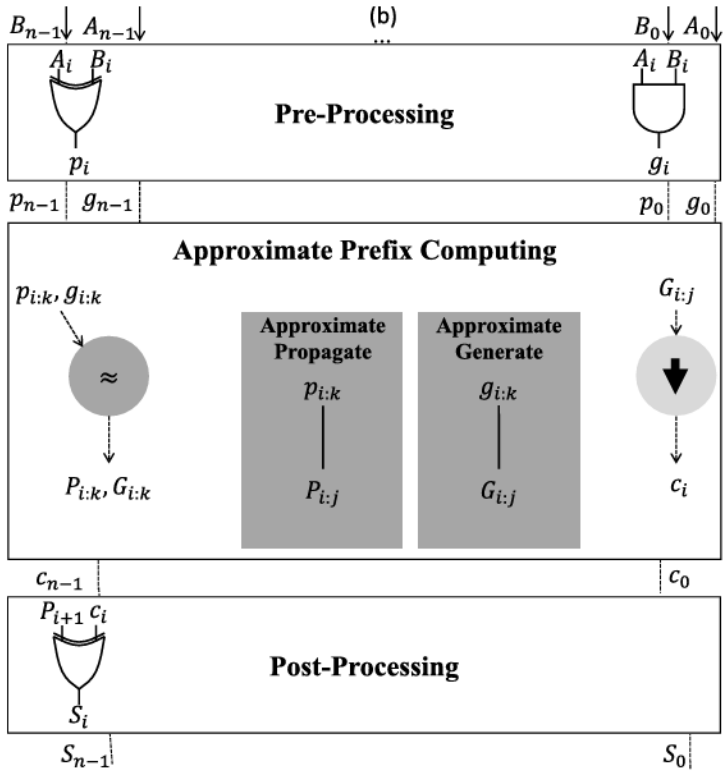

Fig 3.2: *Generic description of the pre-processing, prefix computation, and postprocessing steps of our AxPPA proposal.*

In summary, the architecture involves configuring AxPOs for approximating specific parts of the parallel prefix adder, utilizing a combination of approximate and exact computing to achieve a balance between accuracy and efficiency. The

proposed architecture provides flexibility in configuring the trade-off between accuracy and resource utilization.

3.3 Design and Analysis :

We are implementing a three-operand binary adder technique designed for modular arithmetic, employing a parallel prefix adder with a distinctive four-stage structure. Each stage serves a specific purpose in the computation process, involving bit-addition logic, base logic, PG (propagate and generate) logic, and sum logic. The logical expressions defining these stages are outlined, providing insights into the internal workings of the proposed adder.

Stage Descriptions:

1. Bit Addition Logic (Stage-1):

- Computes the bitwise addition of three n-bit binary input operands using a series of full adders. Logical expressions for sum (*Si*) and carry (*cyi*) signals are defined as:

$$S_i = a_i \oplus b_i \oplus c_i$$
$$cy_i = a_i \cdot b_i + b_i \cdot c_i + c_i \cdot a_i$$

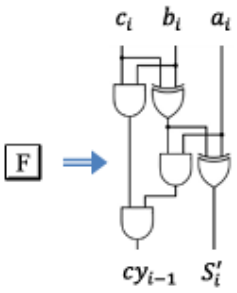

Fig-3.3.1 *The logical diagram of the bit-addition logic*.

2. Base Logic (Stage-2):

Utilizes the outputs of the bit-addition logic to compute generate (G_i) and propagate (P_i) signals. The "squared saltire-cell" is employed for computation, involving n+1 saltire-cells.

Logical expressions for generating and propagating signals are given by:

$$G_i = S_i \cdot cy_{i-1}$$
$$P_i = S_i \oplus cy_{i-1}$$

External carry-input signal (Cin) is considered in the base logic.

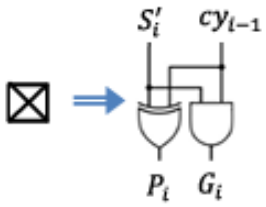

Fig:3.3.2 *The logical diagram of the base logic*.

3. PG (Generate and Propagate) Logic (Stage-3):

Carries out carry computation using a combination of black and grey cell logics. Logical expressions for generating and propagating signals are expressed as:

$$G_{i:j} = G_{i:k} + P_{i:k} \cdot G_{k-1:j}$$
$$P_{i:j} = P_{i:k} \cdot P_{k-1:j}$$

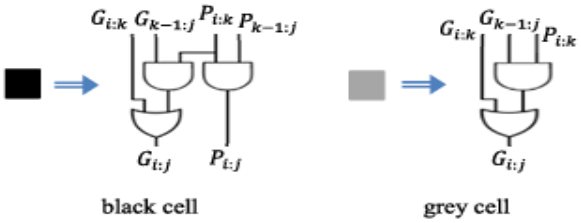

Fig:3.3.3 *The logical diagram of the black cell(P) and grey cell(G)*.

4. Sum Logic (Stage-4):

- Computes the final "sum (S_i)" bits using carry generate ($G_{i:j}$) and carry propagate ($P_{i:j}$)signals.

Logical expression for computing the sum (S_i) is given by:

$$S_i = (P_i \oplus G_{i-1:0})$$

The carryout signal (Cout) is directly obtained from the carry generate bit ($G_{n:0}$).

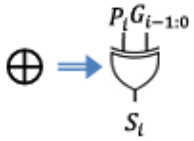

Fig:3.3.4 *The logical diagram of the Sum Logic*

Additional Considerations:

The use of an additional carry-input signal (Cin) is emphasized, providing flexibility for three-operand addition in the proposed adder technique. The logical expressions and diagrams provide a comprehensive overview of the internal workings, allowing for detailed analysis and implementation considerations.

Brent-Kung Adder Vs Kogge Stone Adder (Comparative Analysis):

1. Less Computation Nodes:

The Brent Kung adder is noted for having fewer computation nodes compared to other adder architectures. Computation nodes refer to the units responsible for executing the arithmetic operations within the adder.

2. Maximum Depth and Increased Latency:

While having fewer computation nodes, the Brent Kung adder is mentioned to have maximum depth, contributing to increased latency. The term "depth" in this context likely refers to the length of the longest path in the circuit, influencing the overall delay or latency of the adder.

3. Interconnection Complexity:

The interconnection complexity of black and grey cells in the Brent Kung adder is highlighted as being less than that of the Kogge Stone adder. This suggests that the connectivity between different components within the Brent Kung adder is simpler.

4. Delay Equation:

The delay of the Brent Kung adder structure is given by a specific equation: $2 \cdot (\log_2 n) - 2$ with $2n - 2 - \log_2 n$ computation nodes. This equation provides an analytical way to estimate the delay characteristics of the adder.

5. Structure Comparison with Kogge Stone Adder:

A 16-bit Brent Kung adder structure is mentioned, consisting of 12 black cells and 15 grey cells. The implication is that this structure has a smaller area when compared to both Kogge Stone and Sklansky adders.

6. Simplicity and Fan Out:

- The structure of the Brent Kung adder is described as simpler compared to the Kogge Stone adder. Additionally, the fan-out (the number of gates or inputs a single output is connected to) for the Brent Kung adder is noted to be less than that of the Sklansky adder.

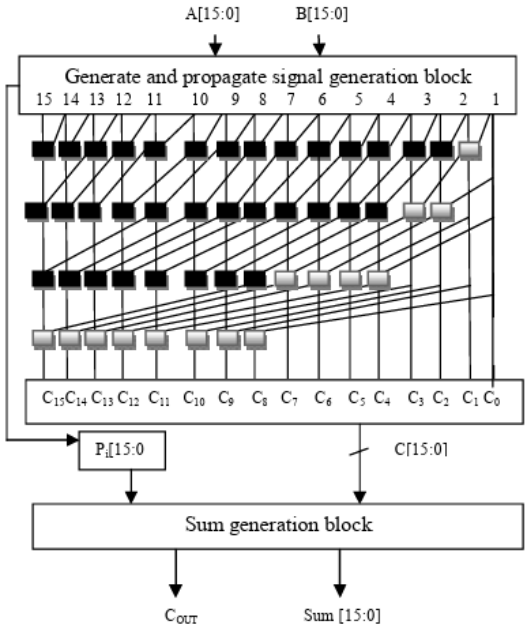

Fig 3.3.4: Structure of 16-bit Kogge Stone Adder

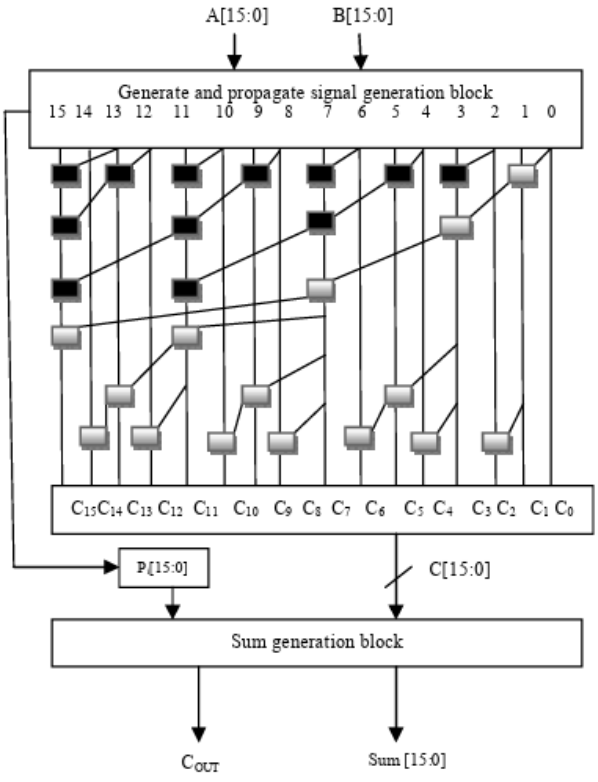

Fig 3.3.5: Structure of 16-bit Brent Kung Adder

In summary, the Brent Kung adder is praised for its reduced interconnection complexity, simpler structure, and potential advantages in terms of area and fan-out compared to other adder architectures. However, it is acknowledged that it may have increased latency due to its maximum depth. These considerations make it a suitable choice depending on the specific requirements and trade-offs in a given application.

CHAPTER 4

SIMULATION RESULTS

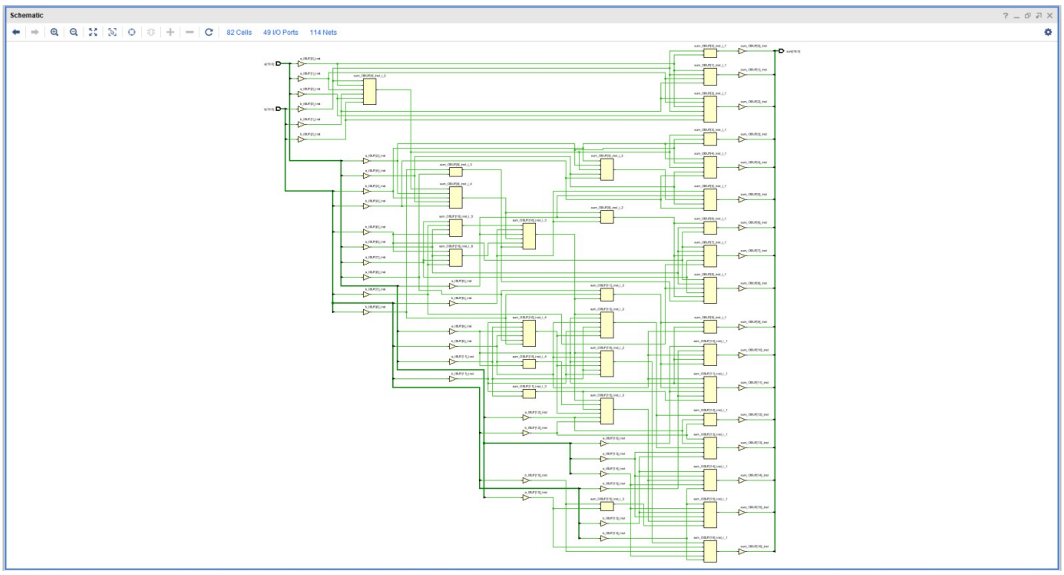

Fig 5.1: Schematic Of 16 bit Brent Kung Adder

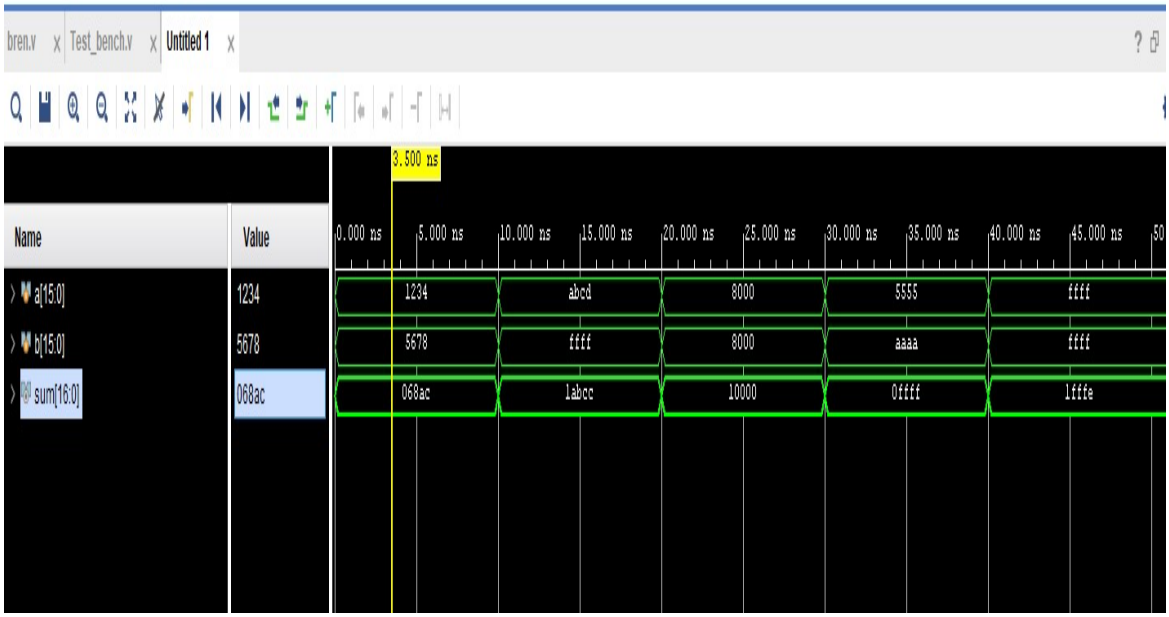

Fig 5.2: Timing Waveform Of 16 bit Brent Kung Adder

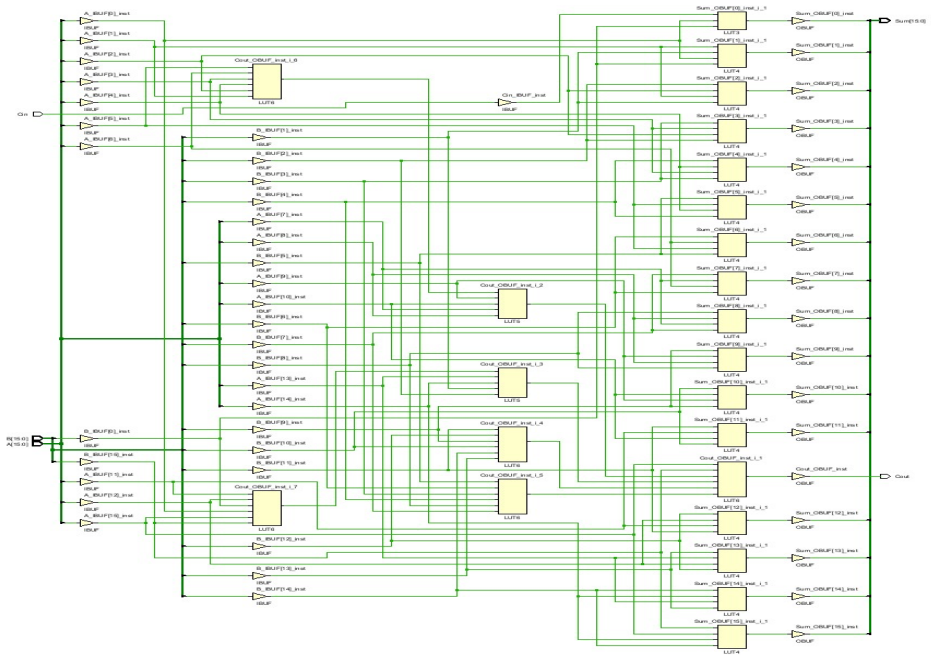

Fig 5.3: Schematic Of 16 bit Kogge Stone Adder

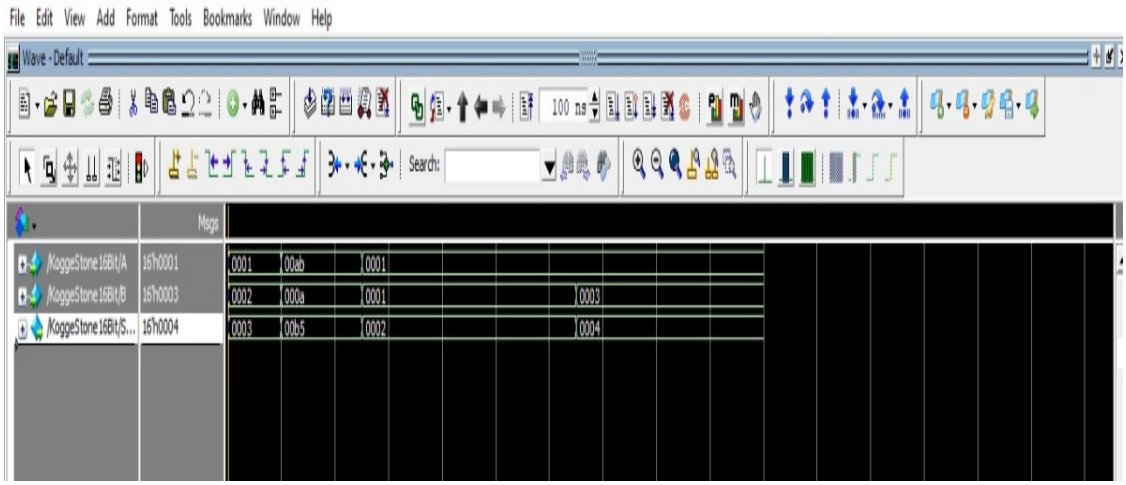

Fig 5.4 : Timing Waveform Of 16 bit Kogge-Stone Adder:

Approximate Adders Simulations

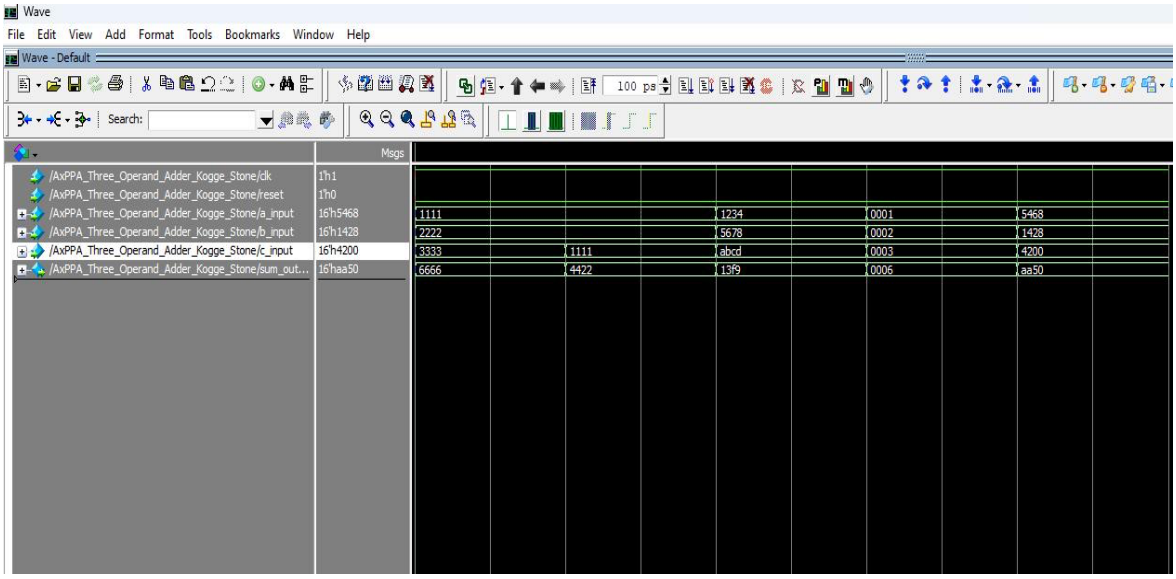

Fig 5.5 Timing diagram of Approximate Kogge Stone Adder

Device Utilization Summary

Slice Logic Utilization	Used	Available	Utilization	Note(s)
Number of Slice LUTs	24	28,800	1%	
Number used as logic	24	28,800	1%	
Number using O6 output only	16			
Number using O5 and O6	8			
Number of occupied Slices	17	7,200	1%	
Number of LUT Flip Flop pairs used	24			
Number with an unused Flip Flop	24	24	100%	
Number with an unused LUT	0	24	0%	
Number of fully used LUT-FF pairs	0	24	0%	
Number of slice register sites lost to control set restrictions	0	28,800	0%	
Number of bonded IOBs	64	560	11%	
Average Fanout of Non-Clock Nets	1.66			

Fig 5.6 Area Report of Approximate Kogge Stone Adder

```
--------------------------------------------------------------------------------
Delay:               6.428ns (Levels of Logic = 6)
  Source:            a_input<8> (PAD)
  Destination:       sum_output<13> (PAD)

  Data Path: a_input<8> to sum_output<13>
                              Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     IBUF:I->O             4    0.694   0.914  a_input_8_IBUF (a_input_8_IBUF)
     LUT6:I0->O            3    0.086   0.609  BL32/BL9/g_output1 (g<8>)
     LUT5:I2->O            3    0.086   0.609  PG_LOGIC/MSBx1/gc_2/o_g1 (gx<11>)
     LUT5:I2->O            3    0.086   0.828  gx<13>1 (gx<13>)
     LUT5:I0->O            1    0.086   0.286  SLG/Mxor_sum_output_Result<13>1 (sum_output_13_OBUF)
     OBUF:I->O                  2.144          sum_output_13_OBUF (sum_output<13>)
```

Fig 5.7 Delay Report of Approximate Kogge Stone Adder

On-Chip	Power (W)	Used	Available	Utilization (%)
Logic	0.000	24	28800	0
Signals	0.000	80	---	---
IOs	0.000	64	560	11
Leakage	0.525			
Total	0.525			

Fig 5.8 Power Report of Approximate Kogge Stone Adder

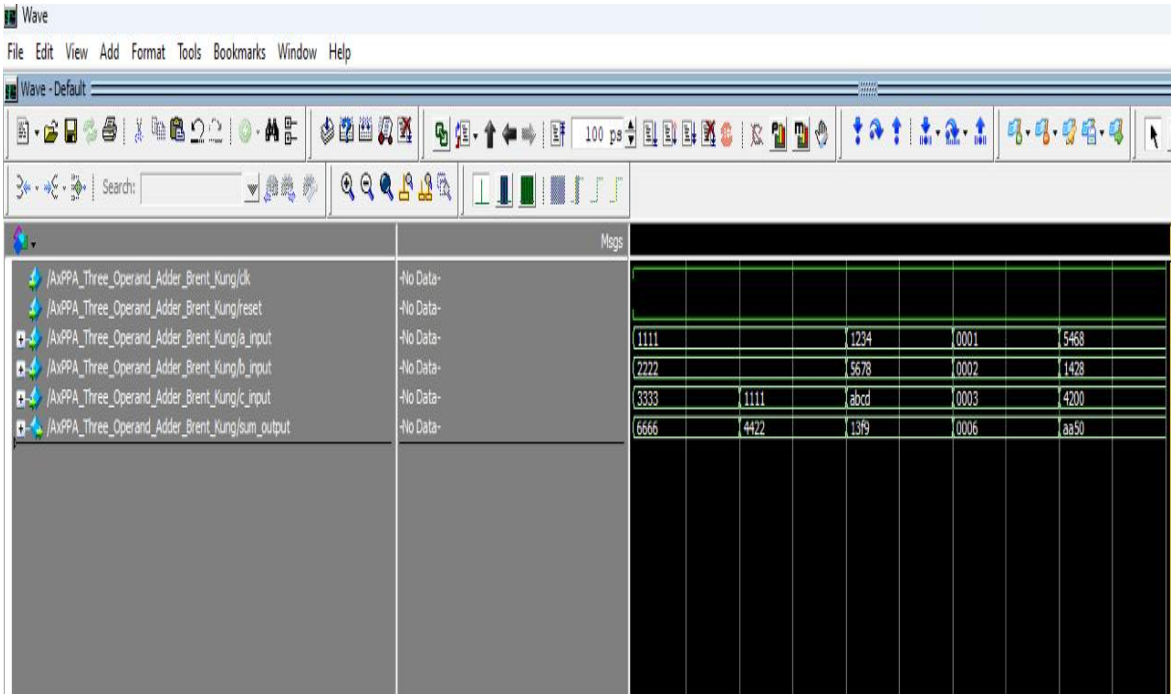

Fig 5.9 Timing diagram of Approximate Brent Kung Adder

Device Utilization Summary				
Slice Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of Slice LUTs	24	28,800	1%	
Number used as logic	24	28,800	1%	
Number using O6 output only	17			
Number using O5 and O6	7			
Number of occupied Slices	19	7,200	1%	
Number of LUT Flip Flop pairs used	24			
Number with an unused Flip Flop	24	24	100%	
Number with an unused LUT	0	24	0%	
Number of fully used LUT-FF pairs	0	24	0%	
Number of slice register sites lost to control set restrictions	0	28,800	0%	
Number of bonded IOBs	64	480	13%	
Average Fanout of Non-Clock Nets	1.71			

Fig 5.10 Area Report of Approximate Brent KungAdder

```
--------------------------------------------------------------------------------
Delay:               6.838ns (Levels of Logic = 7)
  Source:            a_input<9> (PAD)
  Destination:       sum_output<13> (PAD)

  Data Path: a_input<9> to sum_output<13>
                                Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     IBUF:I->O             3   0.694   0.609  a_input_9_IBUF (a_input_9_IBUF)
     LUT3:I0->O            2   0.086   0.823  BAL32/BAL10/Mxor_sum_Result1 (s<9>)
     LUT5:I0->O            2   0.086   0.416  PG_LOGIC/MSBx1/gc_12/o_g1 (gx<10>)
     LUT5:I4->O            3   0.086   0.609  PG_LOGIC/MSBx1/gc_1/o_g1 (gx<11>)
     LUT5:I2->O            3   0.086   0.828  PG_LOGIC/MSBx1/gc_7/o_g1 (gx<13>)
     LUT5:I0->O            1   0.086   0.286  SLG/Mxor_sum_output_Result<13>1 (sum_output_13_OBUF)
     OBUF:I->O                 2.144          sum_output_13_OBUF (sum_output<13>)
    ----------------------------------------
    Total                      6.838ns (3.268ns logic, 3.570ns route)
```

Fig 5.11 Delay Report of Approximate Brent Kung Adder

On-Chip	Power (W)	Used	Available	Utilization (%)
Logic	0.000	24	28800	0
Signals	0.000	79	---	---
IOs	0.000	64	480	13
Leakage	0.560			
Total	0.560			

Fig 5.12 Power Report of ApproximateBrent Kung Adder

COMPARISION TABLE

	High Speed 16-Bit Three Operand Approximate Parallel Prefix Adder (AxPPA) using Vertex-5 XC5VLX50-2ff1153 FPGA	
	Brent Kung Adder	Kogge Stone Adder
Number of Slice LUTs	24	32
Number of Occupied Slices	19	21
Number of Bonded IOBs	64	64
Delay (ns)	6.838	6.428
Power (mW)	0.560	0.525

REFERENCES:

- M. Macedo, L. Soares, B. Silveira, C. M. Diniz and E. A. C. da Costa, "Exploring the use of parallel prefix adder topologies into approximate adder circuits," 2017 24th IEEE International Conference on Electronics, Circuits and Systems (ICECS), Batumi, Georgia, 2017.
- A. Stefanidis, I. Zoumpoulidou, D. Filippas, G. Dimitrakopoulos and G. C. Sirakoulis, "Synthesis of Approximate Parallel-Prefix Adders," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 31, no. 11.
- R. Pandey, "Implementation of Approximate Adder circuit of Ladner Fischer Adder (16 bit)," 2020 4th International Conference on Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, 2020.
- J. Echavarria, S. Wildermann, E. Potwigin and J. Teich, "Efficient Arithmetic Error Rate Calculus for Visibility Reduced Approximate Adders," in IEEE Embedded Systems Letters, vol. 10, no. 2.
- M. M. A. d. Rosa, G. Paim, P. Ü. L. d. Costa, E. A. C. d. Costa, R. I. Soares and S. Bampi, "AxPPA: Approximate Parallel Prefix Adders," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 31, no. 1.
- S. K. Yezerla and B. Rajendra Naik, "Design and estimation of delay, power and area for Parallel prefix adders," 2014 Recent Advances in Engineering and Computational Sciences (RAECS), Chandigarh, India, 2014.
- A. K. Panda, R. Palisetty and K. C. Ray, "High-Speed Area-Efficient VLSI Architecture of Three-Operand Binary Adder," in IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 67, no. 11.
- M. Pashaeifar, M. Kamal, A. Kusha, and M. Pedram, “A theoretical framework for quality estimation and optimization of DSP applications

using low-power approximate adders," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 66, no. 1

- P. Stanley-Marbell et al., "Exploiting errors for efficiency: A survey from circuits to applications," ACM Comput. Surv., vol. 53, no. 3.
- H. B. Seidel, M. M. A. da Rosa, G. Paim, E. A. C. da Costa, S. J. M. Almeida, and S. Bampi, "Approximate pruned and truncated Haar discrete wavelet transform VLSI hardware for energy-efficient ECG signal processing," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 68, no. 5.
- P. Pereira et al., "Energy-quality scalable design space exploration of approximate FFT hardware architectures," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 69, no. 11.
- G. Paim, L. M. G. Rocha, G. M. Santana, L. B. Soares, E. A. C. da Costa, and S. Bampi, "Power-, area-, and compression efficient eight-point approximate 2-D discrete tchebichef transform hard ware design combining truncation pruning and efficient transposition buffers," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 66, no. 2
- L. Soares, J. Oliveira, E. Costa, and S. Bampi, "An energy-efficient and approximate accelerator design for real-time Canny edge detection," Circuits Syst. Signal Process., vol. 39, no. 12, pp. 6098–6120, 2020.
- G. Paim, H. Amrouch, E. A. C. D. Costa, S. Bampi, and J. Henkel, "Bridging the gap between voltage over-scaling and joint hardware accelerator-algorithm closed-loop," IEEE Trans. Circuits Syst. Video Technol., vol. 32, no. 1, pp. 398–410, Jan. 2022.
- Z. G. Tasoulas, G. Zervakis, I. Anagnostopoulos, H. Amrouch, and J. Henkel, "Weight-oriented approximation for energy efficient neural network inference accelerators," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 67, no. 12, pp. 4670–4683, Dec. 2020.

- B. Silveira et al., “Power-efficient sum of absolute differences hardware architecture using adder compressors for integer motion estimation design,” IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 64, no. 12, pp. 3126–3137, Aug. 2017.
- K. M. Reddy, M. H. Vasantha, Y. B. N. Kumar, and D. Dwivedi, “Design of approximate booth squarer for error-tolerant computing,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 28, no. 5, pp. 1230–1241, May 2020.
- M. M. A. da Rosa, G. Paim, R. I. S. J. Castro-Godínez, E. A. C. Costa, and S. and Bampi, “AxRSU: Approximate radix-4 squarer uni,” in Proc. IEEE Int. Symp. Circuits Syst. (ISCAS), Austin, TX, USA, Jun. 2022, pp. 1–4.
- D. Esposito, A. G. M. Strollo, E. Napoli, D. De Caro, and N. Petra, “Approximate multipliers based on new approximate compressors,” IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 65, no. 12, pp. 4169–4182, Dec. 2018.
- W. Liu, J. Xu, D. Wang, C. Wang, P. Montuschi, and F. Lombardi, “Design and evaluation of approximate logarithmic multipliers for low power error-tolerant applications,” IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 65, no. 9, pp. 2856–2868, Sep. 2018.
- H. Jiang, L. Liu, P. P. Jonker, D. G. Elliott, F. Lombardi, and J. Han, “A high-performance and energy-efficient FIR adaptive fil ter using approximate distributed arithmetic circuits,” IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 66, no. 1, pp. 313–326, Jan. 2019.
- A. G. M. Strollo, E. Napoli, D. De Caro, N. Petra, and G. Di Meo, “Comparison and extension of approximate 4–2 compressors for low power approximate multipliers,” IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 67, no. 9.