DESIGN AND IMPLEMENTATION OF ADDER WITH SINGLE LUT FOR AERIAL IMAGE PROCESSING APPLICATIONS.

A PROJECT REPORT

Submitted by

RANJITH.R	(113021106081)
VINAY KUMAR.V	(113021106111)
MOHITH.M	(113021106055)

in partial fulfilment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

VELTECH HIGHTECH
DR. RANGARAJAN DR. SAKUNTHALA ENGINEERING COLLEGE
(An Autonomous Institution, Affiliated to Anna University, Chennai)
AVADI, CHENNAI-600062

ANNA UNIVERSITY: : CHENNAI 600 025
May 2023

ANNA UNIVERSITY: : CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report **"DESIGN AND IMPLEMENTATION OF ADDERS WITH SINGLE LUT FOR AERIAL IMAGE PROCESSING APPLICATIONS"** is the bonafide work of "RANJITH.R(113021106081),VINAYKUMAR.V(1130211060111),MOHITH.M (113021106055)" who carried out the work under my supervision.

SIGNATURE

Dr. S. Hemajothi

HEAD OF THE DEPARTMENT

Professor

Department of ECE

VEL TECH HIGH TEC
DR. RANGARAJAN DR. SAKUNTHALA
ENGINEERING COLLEGE

SIGNATURE

Mrs.Shofia priyadharshini

SUPERVISOR

Assistant Professor

Department of ECE

VEL TECH HIGH TECH
DR. RANGARAJAN DR. SAKUNTHALA
ENGINEERING COLLEGE

Submitted for the viva-voce held on …….at VEL TECH HIGH TECH DR. RANGARAJAN DR. SAKUNTHALA ENGINEERING COLLEGE.

INTERNAL EXAMINER EXTERNAL EXAMINER

ACKNOWLEDGEMENT

We wish to express our sincere thanks to the people who extended their help during the course of our project work.

We are greatly and profoundly thankful to our honourable **Chairman Col. Prof. Dr. Vel. R. RANGARAJAN, B.E., (Mechanical), B.E., (Electrical), M.S., (Automobile),D.Sc.,** and our **Vice Chairman Mrs. SAKUNTHALA RANGARAJAN,MBBS.,** Vel Tech High Tech Dr. Rangarajan Dr. Sakunthala Engineering college, for facilitating us with this opportunity.

We are greatly thankful to our **Dr.Mrs. MAHALAKSHMI KISHORE KUMAR B.E., M.B.A (U.S.A)., Ph.D., Chairperson and Managing Trustee,**and to our **Vice President Mr. K.V.D. KISHORE KUMAR, B.E, M.B.A(U.S.A).,.**VelTech High Tech Dr. Rangarajan Dr. Sakunthala Engineering college

We also record our sincere thanks to our honourable Principal **Dr. E. Kamalanaban Ph.D**. We also greatly thankful to **Dr. V.R Ravi PhD.,Dean- SoEC**, for their valuable suggestion.

We also record our sincere thanks to our honourable **Head of the Department, Dr. S. Hemajothi, Professor,** for her constant support and encouragement.

We would like to extend our sincere thanks to our **supervisor Mrs. Shofia priyadarshini M.E.,Assistant Professor**, Department of Electronics and communication, Vel Tech HighTech Dr.Rangarajan Dr. Sakunthala Engineering college for their constant technical support, valuable suggestions which enabled us to complete our project successfully.

Further the acknowledgement would be incomplete if we would not mention a word of thanks to our most beloved **PARENTS** whose continuous support and encouragement all the way through the course has led us to pursue the degree and confidently complete the project.

ABSTRACT

Aerial image processing plays a crucial role in various applications such as remote sensing, surveillance, and environmental monitoring. Real-time processing of high-resolution aerial images demands efficient and optimized hardware architectures. This paper proposes a design and implementation of an adder with a single Look-Up Table (LUT) for aerial image processing applications.The conventional approach for addition operations in digital systems involves cascading multiple arithmetic logic units (ALUs), which results in increased hardware complexity and power consumption. To address these challenges, the proposed design exploits the inherent parallelism and flexibility of LUTs, which are widely utilized for implementing combinational logic functions.The key idea is to utilize a single LUT to perform addition operations efficiently. The LUT is programmed to generate the sum of two input operands by storing precomputed values. By carefully mapping the inputs to the LUT, the proposed adder achieves high-speed performance and reduced area overhead compared to conventional adder architectures.The design methodology involves analyzing the requirements of aerial image processing applications and deriving the corresponding truth table for the adder operation. The LUT-based adder is implemented using hardware description languages and simulated using appropriate tools to verify its functionality and performance. Experimental results demonstrate that the proposed adder architecture effectively handles addition operations in aerial image processing applications. It achieves improved throughput and reduced power consumption compared to conventional ALU-based adders. Additionally, the design exhibits a smaller hardware footprint, making it suitable for integration into resource-constrained aerial image processing systems. In conclusion, the design and implementation of an adder with a single LUT provide an efficient solution for performing addition operations in aerial image processing applications. The proposed architecture offers improved performance, reduced power consumption, and a smaller hardware footprint, making it a promising choice for real-time processingofhigh-resolutionaerialimages.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

FPGA	Field Programmable Gate Array
ASIC	Application specific integrated circuits
LUT	Look Up table
RCA	Ripple carry adder
CSA	Carry Select adder
DCT	Discrete Cosine transform

CHAPTER-1

1. INTRODUCTION

1.1 INTRODUCTION

High-resolution aerial imaging and automated yet actionable video analytics are widely-being deployed by military and other government-sponsored law enforcement for on-demand Intelligence, Surveillance and Reconnaissance (ISR) operations, autonomous Combat, Search and Rescue missions, hostile target containment and border area patrol.

Apart from military use, light-weight and low cost aerial imaging solutions find great utility in diverse commercial applications, including self localization, target tracking, crowd management, drone swarms, collision avoidance and disaster relief operations. However, the first and foremost challenge for resource-starving aerial platforms, such as Unmanned/Micro Air Vehicles (UAVs/MAVs), is management of system resources to meet operational requirements within the allowable Size, Weight and Power (SWaP) constraints.

1.2 APPLICATION SPECIFIC INTEGRATED CIRCUITS (ASICs)

Application-specific integrated circuit is an integrated circuit (IC) chip customized for a particular use, rather than intended for general-purpose use. For example, a chip designed to run in a digital voice recorder or a high-efficiency bitcoin miner is an ASIC. Application-specific standard product (ASSP) chips are intermediate between ASICs and industry standard integrated circuits like the 7400 series or the 4000 series. ASIC chips are commonly created utilizing metal-oxide-semiconductor (MOS) innovation, as MOS coordinated circuit chips.

1.3 ADDERS

Adders are fundamental digital circuits used for arithmetic operations in computer systems and digital signal processing. They are responsible for adding or

combining binary numbers, and they form a crucial component in various computational tasks. There are several types of adders commonly used:Ripple Carry Adder (RCA): RCA is the simplest form of adder, where each bit of the inputs and the carry bits are added sequentially. It propagates the carry from one stage to the next, resulting in a delay that increases linearly with the number of bits. While simple to implement, RCA can be slower for larger numbers due to the carry propagation.Carry Look-Ahead Adder (CLA): CLA overcomes the carry propagation delay of RCA by precomputing carry snals for each stage using lookahead logic. It allows for parallel computation of carry signals, resulting in faster addition operations, independent of the number of bits. CLA is more complex than RCA but provides improved performance.Carry Select Adder (CSA): CSA is designed to reduce the carry propagation delay by using multiple RCA modules operating in parallel. It computes the addition using two sets of adders, each assuming a different carry-in value, and then selects the correct result based on the actual carry-in value. CSA provides a compromise between speed and complexity. Carry Skip/Adder (CLA): Also known as a carry-bypass adder, CLA aims to minimize the carry propagation delay by skipping the carry computation for certain groups of bits. It uses a combination of RCA and multiplexers to bypass carry propagation for specific groups of bits. CLA is more complex than RCA but provides improved speed for certain input patterns.Approximate Adder: Approximate adders are designed to trade off accuracy for improved performance or reduced hardware complexity. They introduce controlled errors in the output to achieve faster operation or lower power consumption. Approximate adders are commonly used in applications where minor inaccuracies can be tolerated, such as multimedia processing or machine learning.The choice of adder depends on the specific requirements of the application, considering factors such as accuracy, performance, area efficiency, power consumption, and design complexity. Each type of adder has its own trade-offs, allowing for customization based on the specific needs of the system.

1.4 FIELD PROGRAMMABLE GATE ARRAY

FPGA differs from ASIC because of its open architecture. In ASICs, computation is realized with logic gates. However, in case of FPGAs, logic is implemented through Configurable Logic Blocks (CLBs). CLB can be of two types i.e. SLICEL and SLICEM (Logic and Memory slices) and is connected to a switch matrix for access to the general routing matrix. Each slice comprises of Look-Up Tables (LUTs) and flip flops. Overall numbers of flip flops and LUTs are different for different FPGA chips.

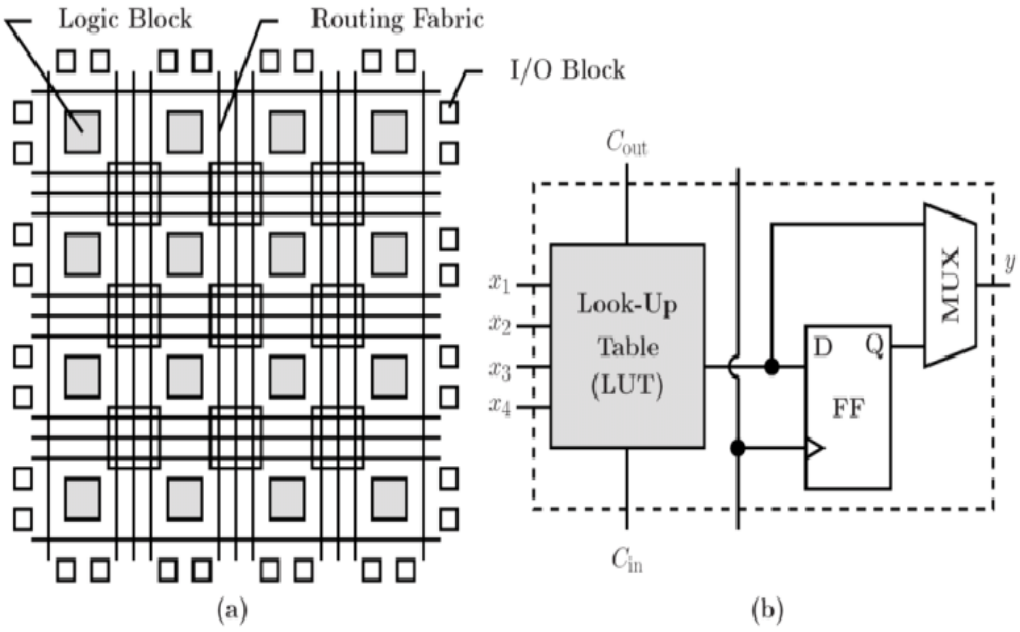


Fig 1.1 FPGA architecture

1.5 FPGA vs ASIC

ASICs are designed as a set of logic gates and minimize the number of logic gates to reduce silicon area and power consumption. However, ASIC-focused optimizations cannot be directly mapped to the FPGA architecture, primarily due to the differences in their underlying design principles.

FPGAs comprise of Look-up Tables (LUTs) as their basic building block and therefore, require specialized LUT-specific approximation techniques. Nevertheless, Field Programmable Gate Arrays (FPGAs) are a preferred choice for hardware design of certain safety-critical and real-time applications (such as UAVs), due to their run-time re-configurability, rapid prototyping (less time to market) and low cost.

1.6 16 bit-Ripple Carry Adder:

A ripple-carry adder (shown in fig 1)is a basic adder architecture that uses a series of full-adder stages to perform the addition operation. This type of adder is relatively simple to implement and requires minimal circuitry, making it a popular choice for small-scale applications. However, in the context of DCT, ripple-carry adders can be slow and inefficient, especially for large data sets, due to the need to propagate carries through multiplestages**.**

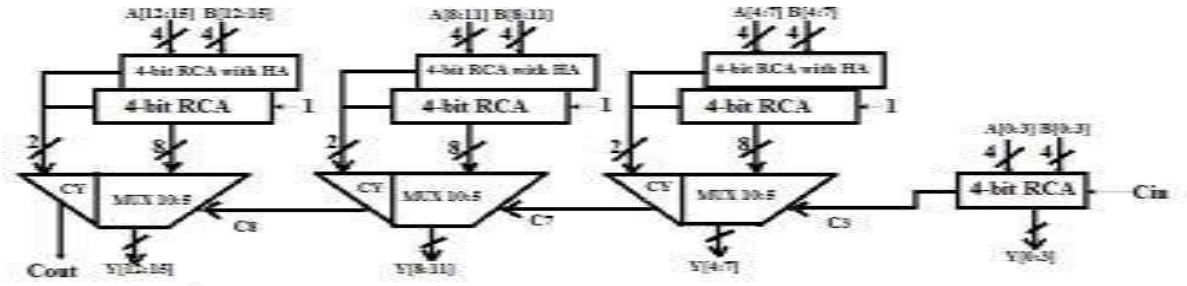


Fig 1.2: 16 bit Ripple Carry Adder

1.7 Carry Select Adder:

A carry select adder (shown in fig 2)is a type of adder that uses a set of precomputed carry bits to speed up the addition operation. This type of adder is particularly useful for computing the DCT because it can reduce the number of full-adder stages required to perform the addition, leading to a reduction in the overall delay of the circuit. However, carry-select adders (fig 2)are generally more complex than ripple-carry adders,

which can lead to higher power consumption and area requirement.

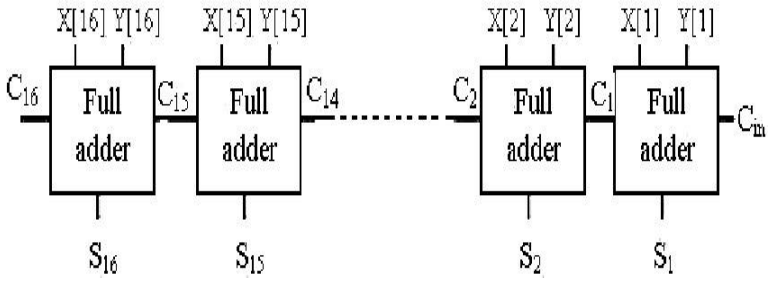


Figure 1.3 :Carry Select Adder

1.8 Approximate Adder:

An approximate adder(shown in fig 3) is a type of adder that sacrifices accuracy for performance. It uses simplified circuits to perform the addition operation, which reduces the amount of time and energy required to perform the operation. However, the resulting output may not be exact,leading to a loss in accuracy. In the context of DCT, using an approximate adder can result in a loss of precision in the computed DCT coefficients, leading to a degradation in the quality of the reconstructed signal.

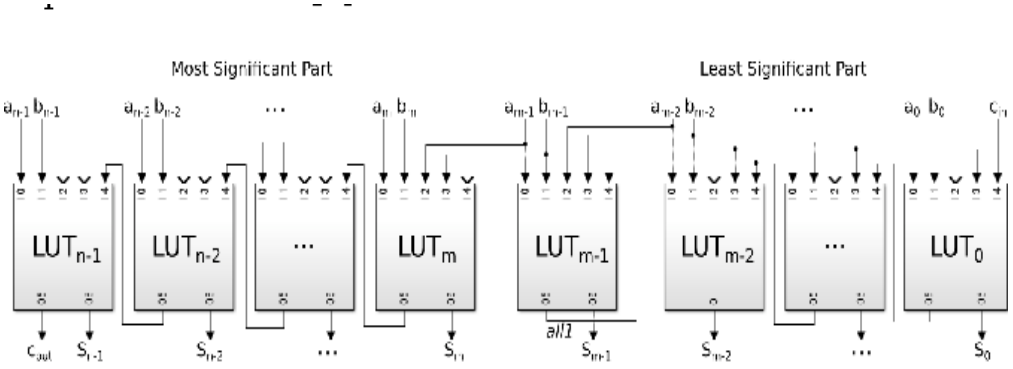


Fig 1.4 :Approximate Adder

1.9 Comparing adders

When comparing different adder designs such as the approximate adder, ripple carry adder, and carry look-ahead adder, several factors come into play: **Accuracy**: Ripple carry adders and carry look-ahead adders are designed to provide precise and accurate arithmetic operations. They guarantee exact results for addition but may suffer from longer propagation delays as the number of bits increases. On the other hand, approximate adders trade off accuracy for improved performance by allowing for small errors in the output. They are suitable for applications where minor inaccuracies can be tolerated, such as multimedia processing or machine learning.**Performance**: Ripple carry adders have a straightforward design, but their performance is limited due to the carry propagation delay. The delay increases linearly with the number of bits, resulting in longer overall computation time. Carry look-ahead adders address this issue by precomputing the carry signals, reducing the delay to a constant value. This design provides faster operation, making it more suitable for high-speed arithmetic operations compared to ripple carry adders.**Area Efficiency**: Approximate adders are often designed with reduced hardware complexity, resulting in smaller circuit area compared to precise adders like ripple carry and carry look-ahead adders. This advantage can be particularly valuable in resource-constrained applications or when a high density of adders is required in a design.**Power Efficiency**: Ripple carry adders have a simple structure but can consume significant power due to the large number of carry signals propagating through the circuit. Carry look-ahead adders can reduce power consumption by minimizing the number of carries, making them more power-efficient than ripple carry adders. Approximate adders can also offer power savings by utilizing simpler circuitry, reducing the dynamic power consumption.**Design Flexibility**: Ripple carry adders are relatively straightforward to design and implement, making them easily customizable for specific requirements. Carry look-ahead adders involve more complex logic but can be optimized for different performance or power goals. Approximate adders, depending on the desired level of approximation, can provide a range of trade-offs between accuracy and performance. **Application Suitability**: Ripple carry adders and carry look-ahead adders are commonly used in applications that require exact and precise arithmetic operations, such as critical numerical calculations or cryptographic systems. Approximate adders are suitable for applications where minor errors in the output

can be tolerated, such as multimedia processing, error-resilient computing, or low-power applications.In summary, the choice of adder design depends on the specific requirements of the application. Ripple carry adders and carry look-ahead adders are preferred for precise calculations, while approximate adders offer advantages in terms of performance, area efficiency, and power consumption, at the cost of sacrificing accuracy to some degree.

1.10 MOTIVATION

FPGA logic blocks are implemented using pure LUT. A k-input LUT is a block of SRAM, which can implement a truth table of k inputs and one output. For example, 3-input LUT allows 3 bit inputs and stores 23 possible single-bit outcomes, while mapping each unique input combination to deliver the correspondingly stored unique output.

The main aim of this project present a novel approximate adder design methodology, for FPGA-based systems with improved SWaP performance, besides preserving the accuracy requirements within acceptable thresholds. The design methodology proposed is focuses on the FPGA-specific Look-Up Table (LUT) architecture to introduce approximations while splitting the carry chain into LUT-based sub-adders, with flexible overlap to tune the adder's accuracy and achieve the overall latency of a single LUT.

1.11 PROJECT FLOW

Approximate adders are designed with increased efficiency in terms of power, area and delay using Xilinx 14.7 design suite, as introduces sub-adder-based designs which carry certain sub-adder output results, thus breaking the carry chain. Then error probability is calculated using the mathematical formula given. Then the adder design is introduced inside the DCT module for image compression in MATLAB.

The content provided focuses on the utilization of Xilinx 14.7 design suite to create approximate adders that offer improved efficiency in terms of power, area, and delay. These adders are designed with the aim of optimizing the resources while

maintaining acceptable performance levels. One approach employed is the use of sub-adder-based designs, which introduce a break in the carry chain by carrying certain sub-adder output results. By breaking the carry chain, these designs can reduce the critical path delay and enhance the overall performance of the adder.

In addition to improving the efficiency of the adders, the content mentions the calculation of error probability using a mathematical formula. This step allows for an evaluation of the accuracy and reliability of the approximate adder designs. By quantifying the error probability, it becomes possible to assess the trade-off between performance improvements and the potential introduction of errors in the adder operations.

Furthermore, the content highlights the integration of the approximate adder design within the Discrete Cosine Transform (DCT) module in MATLAB. The DCT module is commonly employed in image compression algorithms. By incorporating the adder design within this module, the efficiency gains achieved in terms of power, area, and delay can be leveraged to improve the overall image compression process. This integration showcases the practical application and benefits of the approximate adder design in a real-world scenario.

Overall, the content emphasizes the use of Xilinx 14.7 design suite to create efficient approximate adders that break the carry chain using sub-adder-based designs. It also discusses the calculation of error probability and the integration of the adder design within a DCT module for image compression in MATLAB.

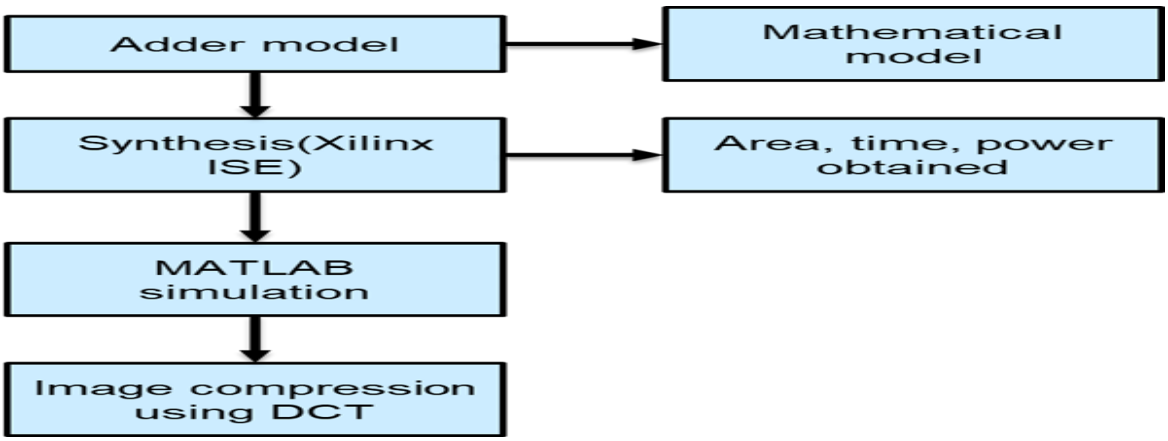


Fig 1.5 PROJECT FLOW

CHAPTER -2

LITERATURE SURVEY

2.1 Design and analysis of high speed and using carry select adder

AUTHOR : YaswanthD , S. Nagaraj , R.VishnuVijeth

The researchers used a variety of adders, including ripple carry, carry select, and Brent Kung, to create and analyse an eight-bit Vedic multiplier. In some of their designs, they employed the Ling Carry select (CS) adder and the binary system to Excess-1 (BEC) converter. The study evaluated speed as well as area capacity of several carry choose adder designs with that of the Vedic multiplier. In terms of both area and latency, the researchers found that an eight-bit multipliers employing Brent Kung (BK) with Carry Select Adder (CSL) offered the greatest results.

2.2 CMOS Implementation of Efficient 16-Bit Square Root Carry-Select Adder

AUTHOR: Shamim Akhter, Saurabh Chaturvedi, KilariPardhasardi

In comparison to a standard CSA, an a 16-bit Square Root Carry-Select Adder (SQRT CSA) was developed to use less space and power. By using a binary-to-excess-1 converter (BEC), the SQRT CSA substitutes the Cin=1 block seen in traditional CSA. Additional implementations include RCA and BEC with various bit widths. By applying Mentor Graphics Design Architect, transistor-level schematics are created, and the structure of the design is simulated using Eldo using TSMC 0.35m CMOS technologies and 3.3V supply voltage.

2.3 Simulation-Based Evaluation of Frequency Upscaled Operation of Exact/Approximate Ripple Carry Adders

AUTHOR: Junqi H, T.Nandha Kumar, Haider AbbasFabrizio Lombardi

The ripple carry adders and exact and approximation adder cells are evaluated in this study utilising the frequency upscaling method. With this technique, inputs to an adder cells are sent more often than they can handle, which results in addition mistakes. The findings demonstrate that the inaccurate adder can operate at a faster frequency and with less energy loss than an exact adder, as well as that the unsure and exact RCAs have extremely near normalised mean error distances and mean relative error distances.

2.4 A Power-Delay Efficient Hybrid Carry Lookahead/Carry- Select Based Redundant Binary to Two's Complement Converter

AUTHOR: YajuanHe,Chip-Hong Chang

The redundant binary (RB) format is presented in this study as an effective reverse converter for converting it into two's complement format. In comparison to its rivals, the suggested reverse converter completes a 64-bit conversion rates in 829 ps and dissipates just 5.84 mW at a data transmission speed of 1 GHz and a supplying voltage of 1.8 V in TSMC 0.18- m CMOS technology, according to a study.

2.5 Exploring Power-Performance-Quality Tradeoff of Approximate Adders for Energy Efficient Sobel Filtering

AUTHOR: Leonardo B, Morgana M, Claudio M, Eduardo A

In order to increase energy economy, this paper investigates the usage of various parallel-prefix adder topology in precision block of approximation adders. This study investigates many topologies and evaluates their performance using the Sobel picture application, in contrast to most comparable papers that employ only one topology.Results demonstrate that switching from Ripple Carry Adder to faster adder topology significantly reduces energy usage and dynamic power.

2.6 ACA-CSU: A Carry Selection Based Accuracy Configurable Approximate Adder Design

AUTHOR: Alish Kanani,Jigar Mehta, Neeraj

The novel approximation adder proposed in this study uses longer carry chains to provide programmable accuracy levels and includes a Carry Select Module for improved delay features. The design is flexible and can accommodate different degrees of precision. Experiments demonstrate that this novel design has superior latency and accuracy than previously suggested approximation adders.

2.7 Exploring the Use of Parallel Prefix Adder Topologies into Approximate Adder Circuits

AUTHOR: Morgana Macedo, Leonardo Soares , Bianca Silveira , Claudio M. Diniz , Eduardo A. C. da Costa

This research suggests investigating various Parallel-Prefix adders (PPA) topology in

the exact section for better performance as well as power efficiency. The majority of earlier efforts on approximation adders have employed Ripple Carry Adders (RCA) for the precision component. The findings demonstrate that PPA adders may dramatically increase the frequency of the clock with a low energy overhead.

2.8 Design of Low Power and High Speed Carry Select Adder Using Brent Kung Adder

AUTHOR: Pallavi Saxena

In order to develop Carry Select Adder (CSA) architectures, this study suggests employing a parallel prefix adder, especially a Brent Kung adder, as opposed to a Ripple Carry Adder (RCA). It was determined that the Modified SQRT BK CSA had the highest power output but a little speed disadvantage. Tanner EDA tool was used to create the designs at a 45nm scale.

2.9 Probabilistic Error Modeling for Approximate Adders

AUTHOR: Sana Mazahir, Osman Hasan, Rehan Hafiz, Muhammad Shafique, and Jorg Henkel

The methods presented in this study may be used to estimate the chance of error occurring and the possibility Mass Functions (PMF) of an error in a certain class of approximation adders, which can be utilised as measure of performance for comparison. The study illustrates how the suggested methodology may accurately forecast how various approximation adders would perform in real-world applications involving image processing.

2.10 Modified 16 bit Carry Select and Carry Bypass Adder Architectures for High Speed Operations

AUTHOR: Mary Christina Joy , Ansa Jimmy ,Tony C.Thomas, Manju I.Kollannur

This research utilises mathematical modelling as well as simulation methods to compare several 16-bit adders with respect to their latency. New carry bypass and carry select adders are suggested in light of the investigation; these adders demonstrate a considerable reduction in propagation latency when compared to existing adders.

2.11 Approximate Adder Synthesis for Area- and EnergyEfficient FIR Filters in CMOS VLSI

AUTHOR: Leonardo Bandeira Soares, Sergio Bampi

In order to increase the space and power consumption of the FIR filters installed in CMOS, the composition of approximation adders is presented in this research. Indicating energy per test. Our design strategy decreases hardware space in filters and saves money. Our approximation adder approach improves space and power efficiency in CMOS VLSI filters.

2.12 Low-Power Digital SignalProcessing Using Approximate Adders

AUTHOR: Vaibhav Gupta, Debabrata Mohapatra, Anand Raghunathan

The goal of this study is to find ways to save energy in portable multimedia systems that employ signal processing techniques. The suggested method employs imperfect full adder cells having less complicated transistors than accurate full adder cells, which leads to shorter critical pathways and permits voltage scaling. By creating structures for video and picture compression algorithms and generating mathematical representations for errors and consumption of power, the study assesses the effectiveness of this strategy. Comparing simulation findings to current accurate adder approaches, up to 69% power reductions are possible.

2.13 xUAVs: Towards Efficient Approximate Computing for UAVs—Low Power Approximate Adders With Single LUT Delay for FPGA-Based Aerial Imaging Optimization

AUTHOR: TUAHA NOMANI , MUJAHID MOHSIN ,ZAHID PERVAIZ,AND MUHAMMAD SHAFIQUE

The use of FPGAs, or field-programmable gate arrays, for the processing of images in Intelligence, Surveillance, and Reconnaissance (ISR) applications is discussed in this study, as well as the difficulties in implementing such algorithms on platforms with limited resources. The authors suggest a unique approximation adder design technique that

offers considerable performance advantages for error-tolerant applications while compromising the correctness of processed outputs. The most accurate design performs at least 9.9% better in terms of energy consumption when compared to current approximation adders, indicating that the suggested technique has the potential to significantly improve SWaP-index for computationally intensive UAV applications.

2.14 Block-Based Carry Speculative Approximate Adder for Energy-Efficient Applications

AUTHOR:Farhad Ebrahimi-Azandaryani, Omid Akbari , Mehdi Kamal , Ali Afzali-Kusha , and MassoudPedram

The brief suggests a block-based adder with low consumption of energy that divides the adder into a non-overlapping summation blocks and predicts the carry output based on the input operands of the block in question and the block after it. This results in a carry chain with fewer links and a shorter average delay. To improve accuracy and lower the output error rate, a system for mistake detection and recovery has also been presented. Modern approximation adders are outperformed by the suggested adder in terms of energy, latency, area, and output quality.

2.15 Design of Approximate Circuits by Fabrication of False Timing Paths: The Carry Cut-Back Adder

AUTHOR: Vincent Camus , Student Member, IEEE, Mattia Cacciotti , Jeremy Schlachter and Christian Enz

In order to loosen timing restrictions while retaining minimum behavioural change, a new technique for constructing approximation circuits that makes use of fake timing routes is presented in this study. The Carry Cut-Back Adder (CCBA), which increases performance and energy economy while assuring low worst-case mistakes, is used as an illustration of this strategy. The CCBA outperforms cutting-edge and truncated adders for high-accuracy and low-power circuits, achieving high accuracy with large circuit savings.

CHAPTER-3

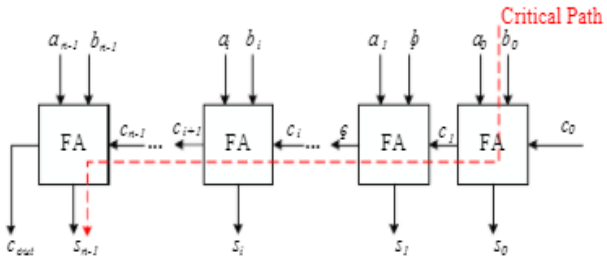
METHODOLOGY

This chapter address the theory about various adders and their methodologies of implementation of its logic blocks. The different adders and its functionality is explained in detail and image compression using DCT is also explained.

3.1 APPROXIMATE ADDERS

An adder performs the addition of two binary numbers. Two basic adders are the ripple-carry adder (RCA) (Figure 3.1) and the carry lookahead adder (CLA) (Figure 3. 2). In an n-bit RCA, the carry of each full adder (FA) is propagated to the next FA, thus the delay and circuit complexity increase proportionally with n (or O(n)). An n-bit CLA consists of that operate in parallel to produce the sum and the generate ($g = a_i b_i$) and propagate ($p = a + b_i$) signals for generating the lookahead carries. The delay of CLA is logarithmic in n (or O(log(n)), thus significantly shorter than for RCA.

However, a CLA requires a larger circuit area (in O (n log(n))), incurring a higher power dissipation. Many approximation schemes have been proposed by reducing the critical path and hardware complexity of an accurate adder. An early methodology is based on a speculative operation. In an n-bit speculative adder, each sum bit is predicted by its previous k less significant bits (LSBs) ($k < n$). As the carry chain is shorter than n, a speculative adder is faster than a conventional design. A segmented adder is implemented by several smaller adders operating in parallel. This type of adder is referred to as a carry select adder. Another method for reducing the critical path delay and power dissipation is by approximating a full adder.



The n-bit ripple-carry adder (RCA). FA: a 1-bit full adder.

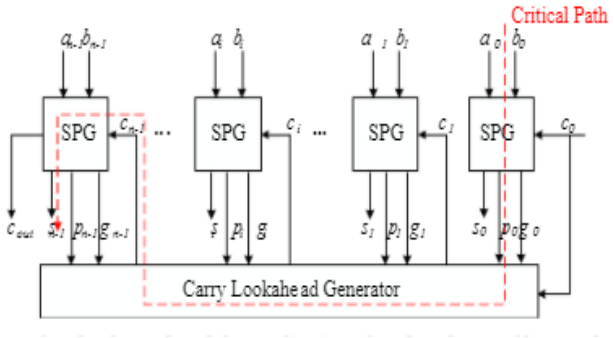


Fig 3.1 and 3.2

3.2 2-BIT & 3-BIT SUB-ADDER BASED DESIGNS

The design implementation in FPGAs is through LUTs. Considering sum of two operands of three bits each, the addition requires computations over six input bits to yield three output bits (ignoring the carry in/out), which can be handled using three six-input LUTs. Each of these three LUTs shall have a six-input mapping onto a single bit output. In this way, by using three six-input LUTs, we calculate the result with a delay of single LUT. Using a maximum of 6-input LUT structure, the viable sizes of sub-adder can be either 2 or 3 bits only. Hence, the proposed sub-adder model variants with sub-adder sizes of 2 and 3 bits, implemented with single LUT delay, are depicted in (Fig. 3.8).

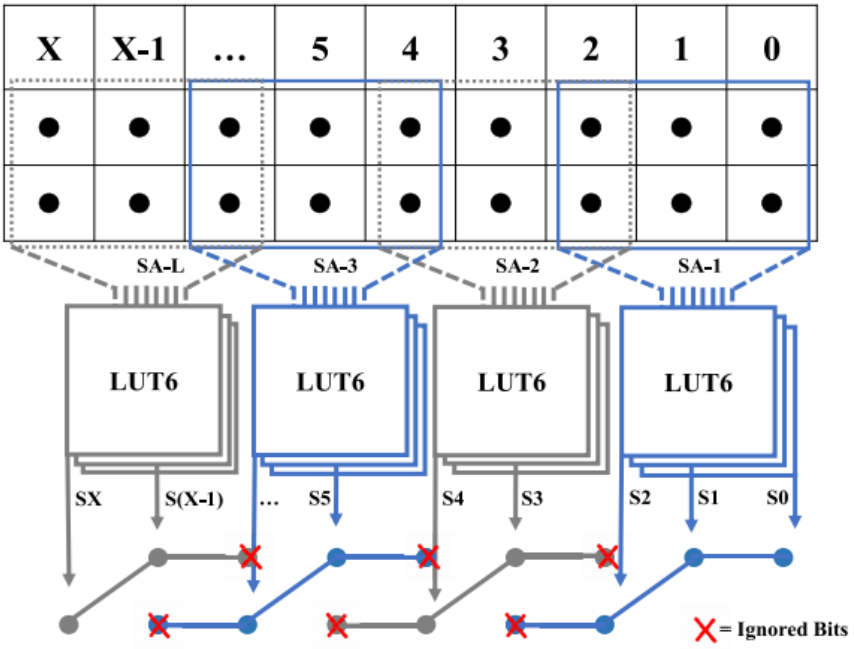


Fig 3.3 LUT6 to perform submodule addition

3.3 5-BIT SUB-ADDER BASED DESIGNS

Modern FPGAs (such as Virtex-7) offer 5-LUT combinational blocks, which can function as 6-input LUTs with common input, as displayed in Fig. 5(a). Thus, the logic implemented within the two 5-input LUTs can be tuned in such a way that one LUT5 is configured for possible carry-in as ‘1’, while the other calculates result if the carry is ‘0’, as presented in Fig. 5(b). This LUT-specific subadder design seeks its motivation from the accurate Carry Save Adder [15]. Using this configuration, the size of this subadder is increased to 5 bits with a latency overhead of 1 built-in FPGA multiplexer.

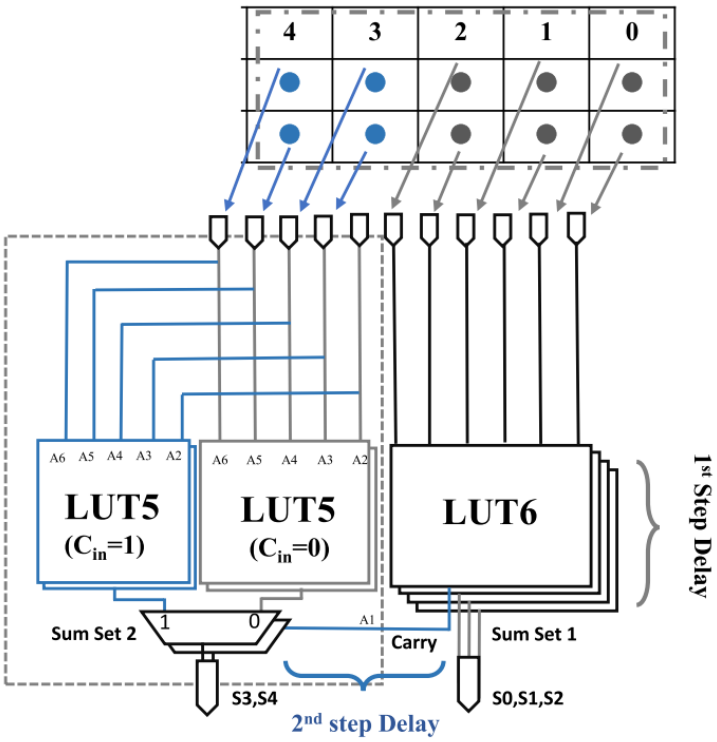


Figure 3.4 Proposed 5 bit sub-adder

3.4 DISCRETE COSINE TRANSFORM

Image is stored or transmitted with having pixel value. It can be compressed by reducing the value its every pixel contains. Image compression is basically of two types:

3.4.1. Lossless compression

In this type of compression, after recovering image is exactly become same as that was before applying compression techniques and so, its quality didn't gets reduced.

3.4.2. Lossy compression:

In this type of compression, after recovering we can't get exactly as older data and that's why the quality of image gets significantly reduced. But this type of compression results in very high compression of image data and is very useful in transmitting image over

network.

The discrete cosine transform (DCT) represents an image as a sum of sinusoids of varying magnitudes and frequencies. The dct2 function computes the two-dimensional discrete cosine transform (DCT) of an image. The DCT has the property that, for a typical image, most of the visually significant information about the image is concentrated in just a few coefficients of the DCT. For this reason, the DCT is often used in image compression applications. For example, the DCT is at the heart of the international standard lossy image compression algorithm known as JPEG. (The name comes from the working group that developed the standard: Joint Photographic Experts Group).

The two-dimensional DCT of an M-by-N matrix A is defined as follows.

$$T_{pq} = \begin{cases} 1/\mathrm{sqrt}(M) & p=0, \quad 0 \le q \le M-1 \\ \mathrm{sqrt}(2/M)\,\cos\dfrac{\pi(2q+1)p}{2M}, & 1 \le p \le M-1, \quad 0 \le q \le M-1 \end{cases}$$

To perform DCT Transformation on an image, first we have to fetch image file information (pixel value in term of integer having range 0 – 255) which we divides in block of 8 X 8 matrix and then we apply discrete cosine transform on that block of data. In the compression addition need to add the subadder design. Then the operation continues and inverse DCT done to obtain reconstructed image.

CHAPTER-4

SIMULATION AND RESULTS

This chapter deals with the evaluation of the approximate adders ,carry select adder , ripple carry adder and their performance in terms of area, power and critical path delay using Xilinx 14.7 tool, Quartus and image compression of those adders are done using MATLAB .

4.1 Implementation of carry select adder

In this module carry select adder is implemented. The RTL view of carry select adder is shown in (Fig 4.1) and simulation is shown in (Fig 4.4).

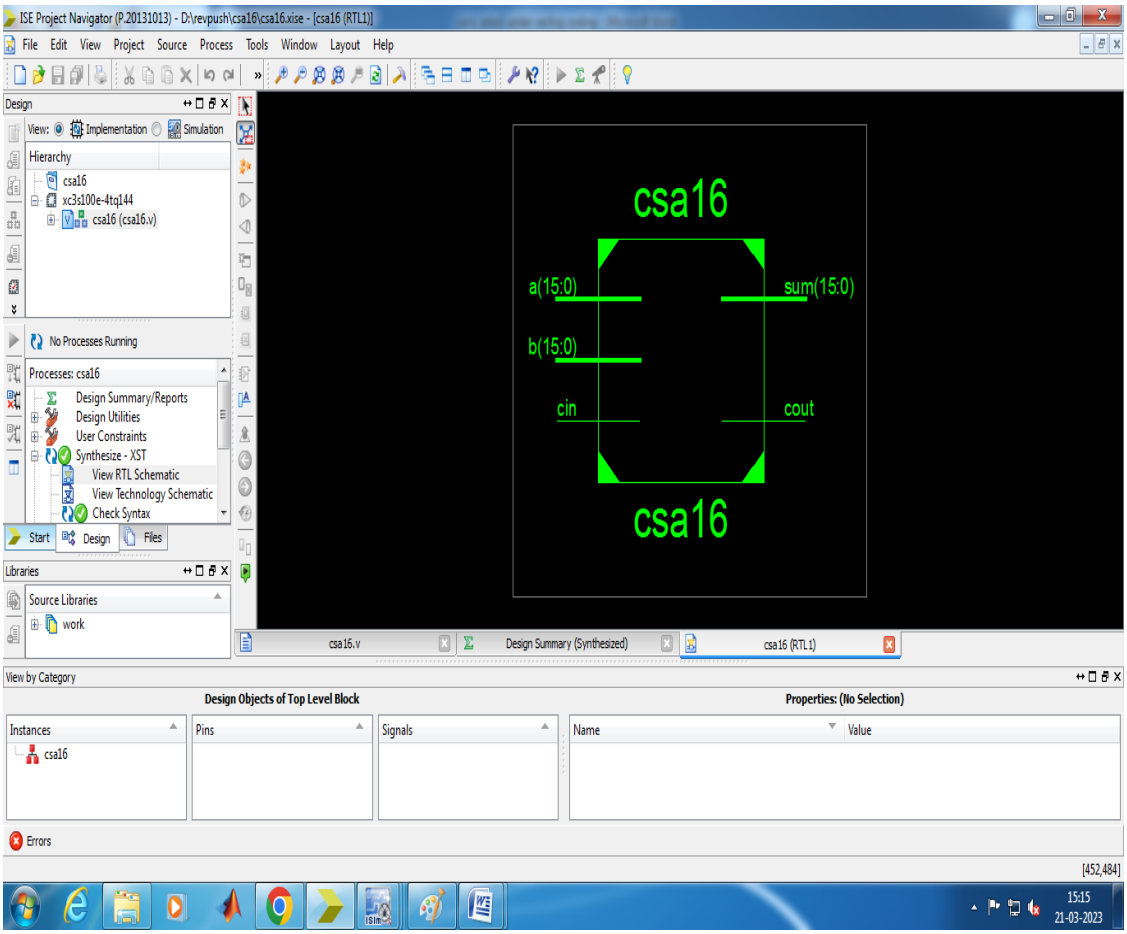


Fig 4.1 : RTL Schematics view of carry select adder

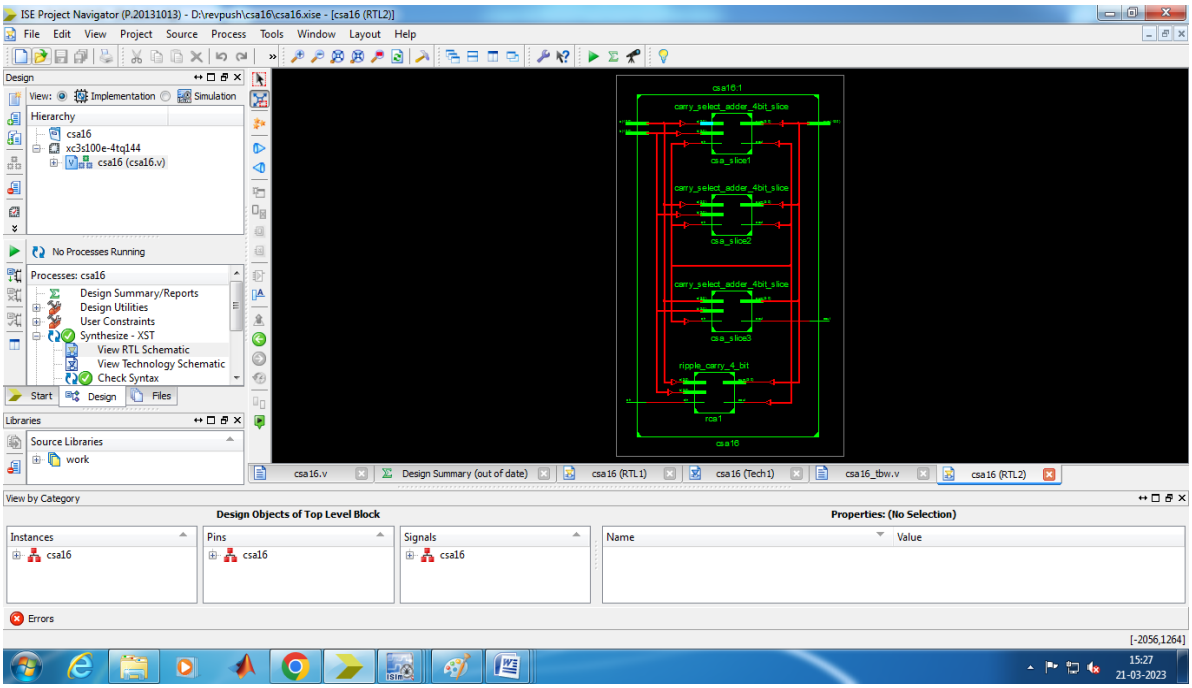


Fig 4.2 LUT schematic view of carry select adder

This (Fig 4.2) shows the how many LUT present in this carry select adder and it shows the complete view of the LUT present in this.

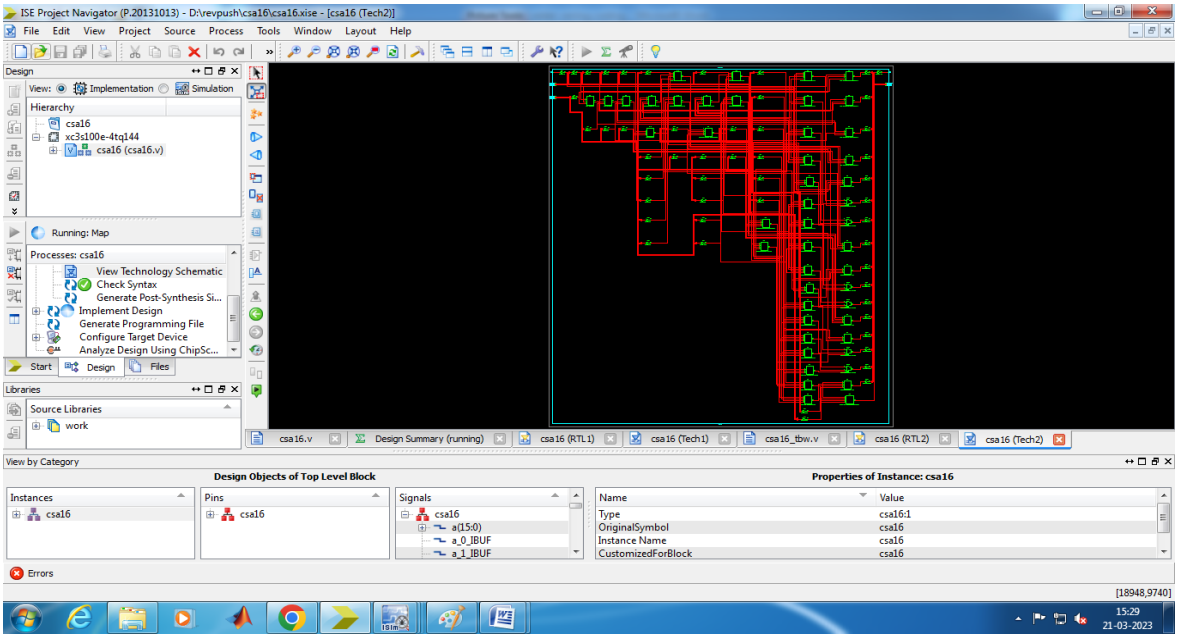


Fig 4.3 Technology mapping of carry select adder

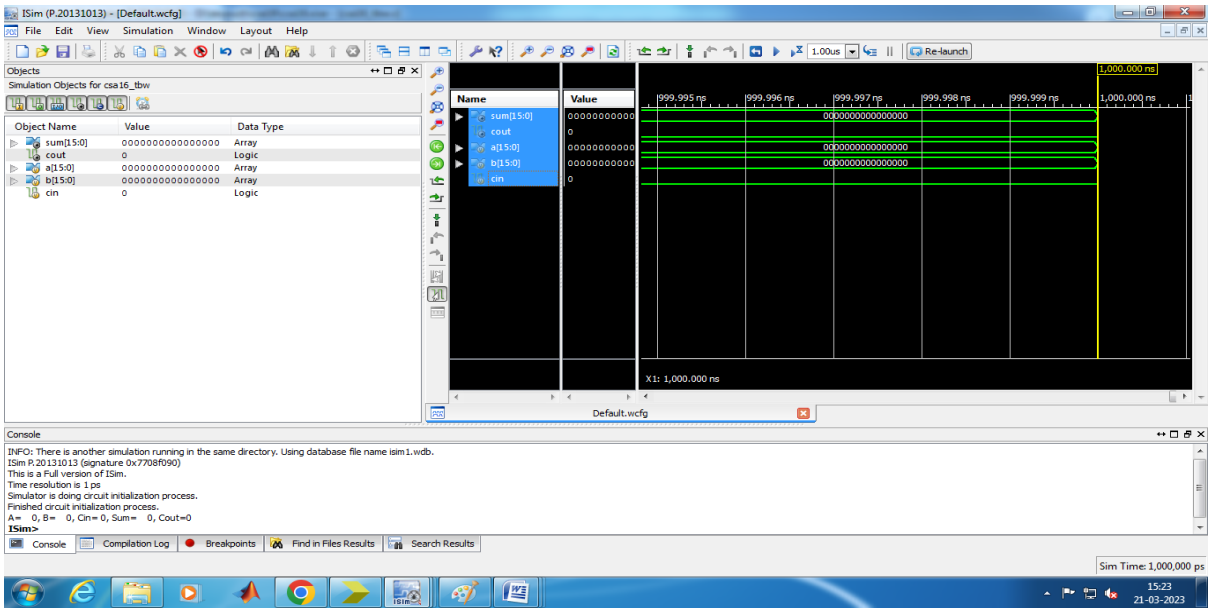


Fig 4.4Simulation of carry select adder

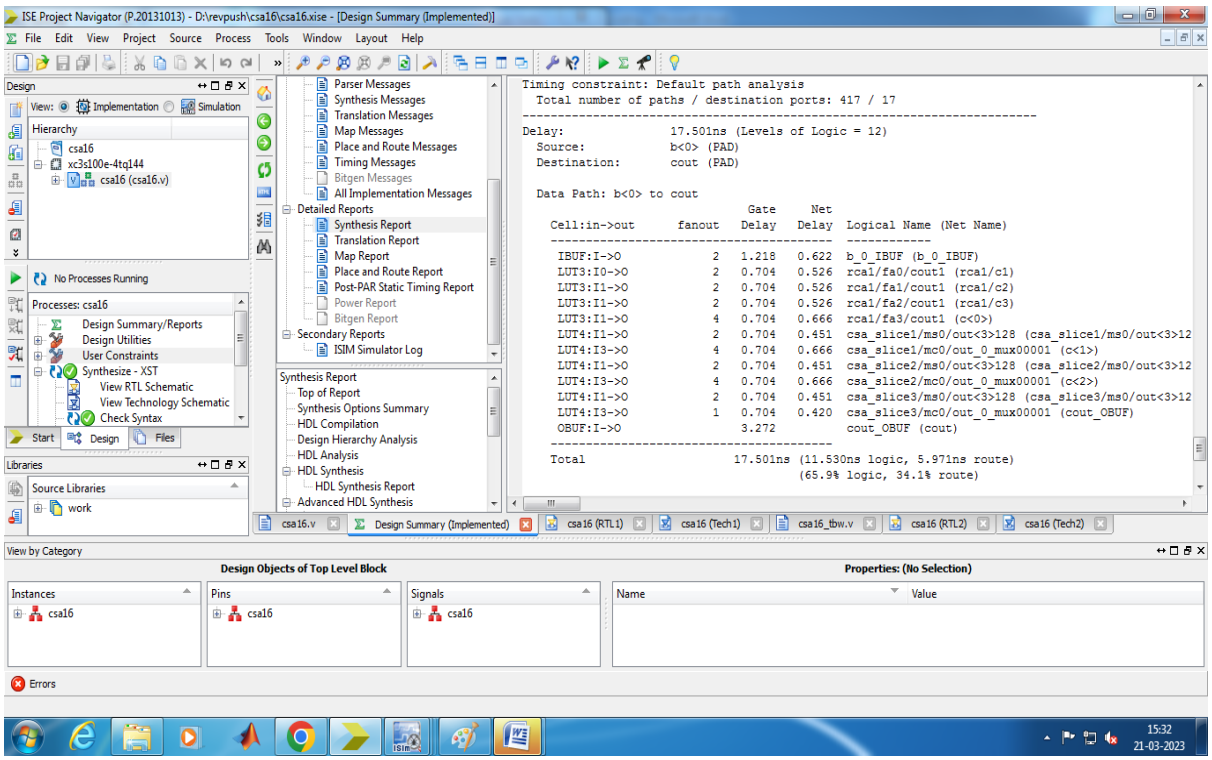


Fig 4.5 output of carry select adder

This output shows (Fig 4.5) the values of time delay , area of the carry select adder.

4.2 Implementation of ripple carry adder

In this module ripple carry adder is implemented. The RTL view of ripple carry adder is shown in (Fig 4.6) and simulation is shown in (Fig 4.9).

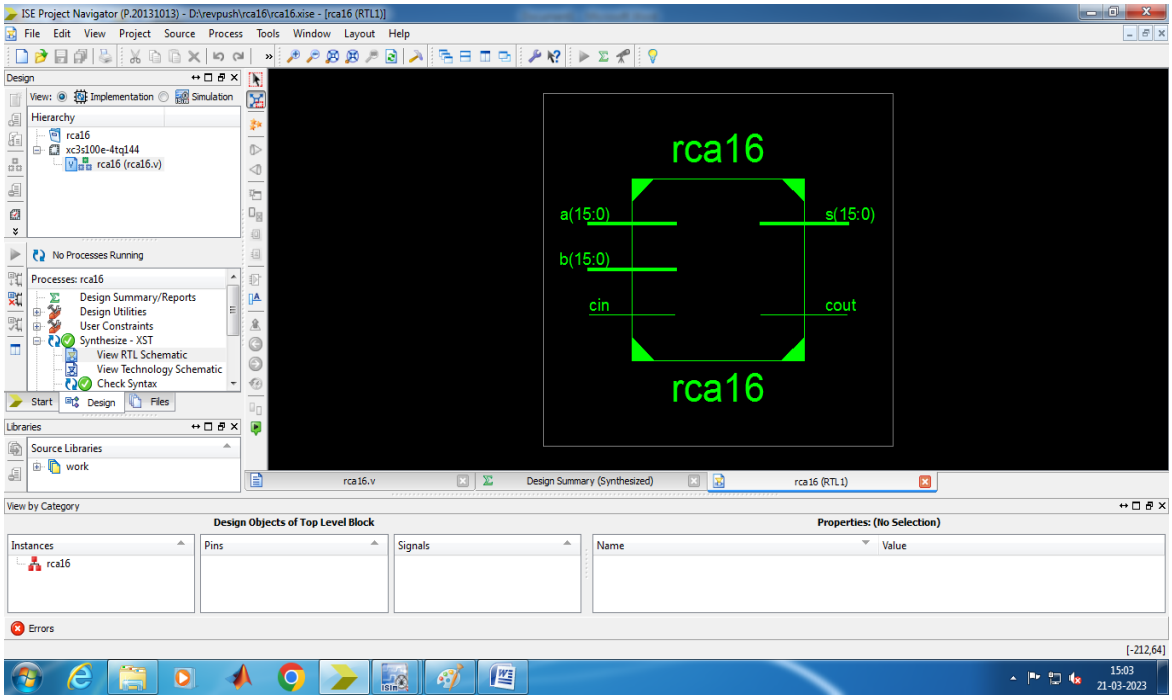


Fig 4.6 RTL schematic view of ripple carry adder

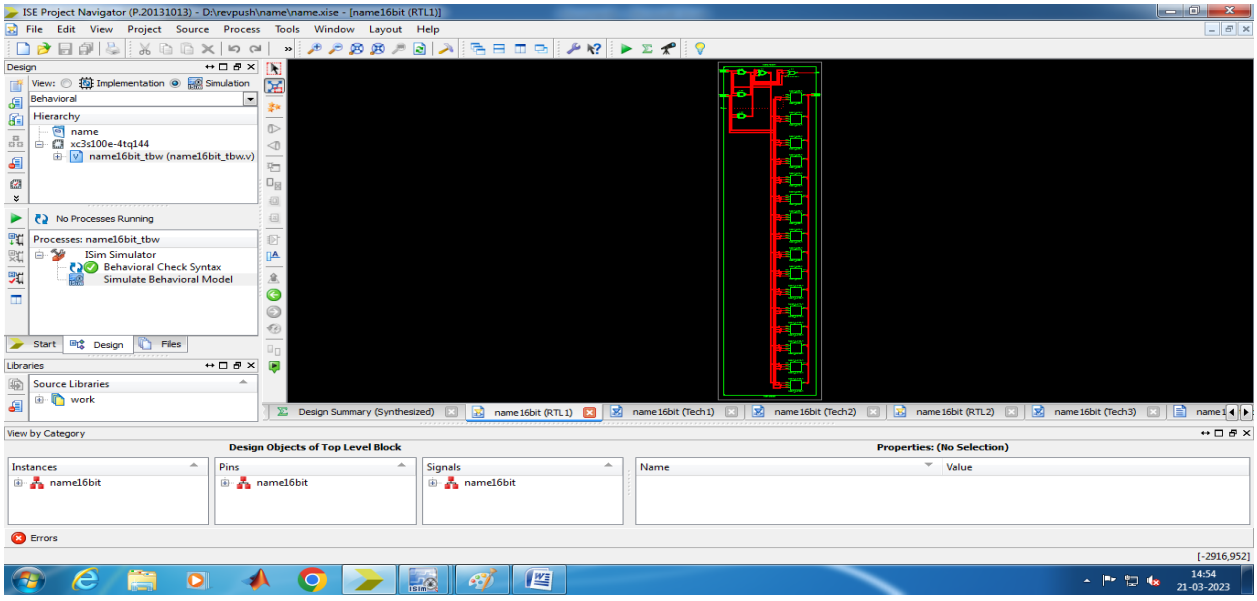


Fig 4.7 LUT schematic view of ripple carry adder

This (Fig 4.2) shows the how many LUT present in this ripple carry adder and it shows the complete view of the LUT present in this.

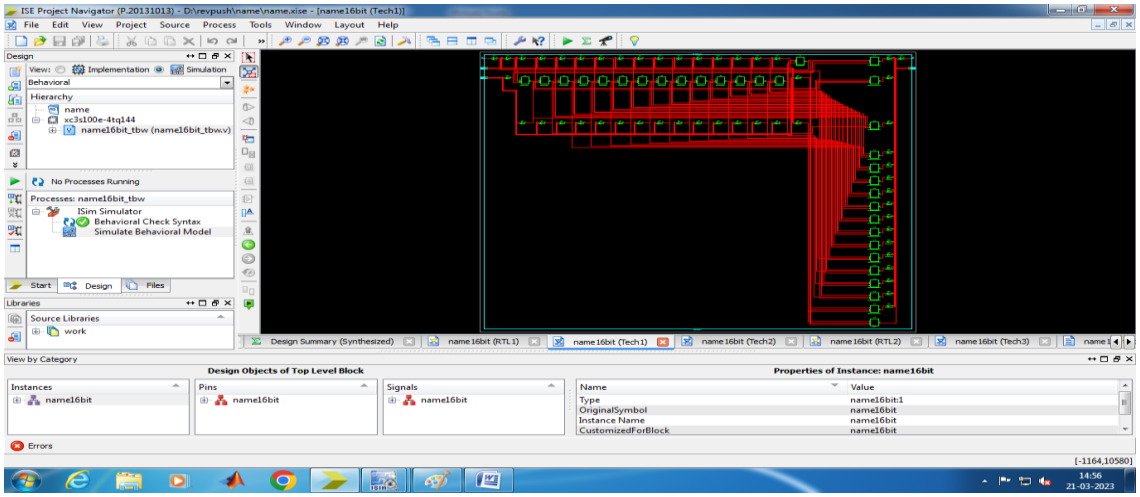


Fig 4.8 Technology mapping of ripple carry adder

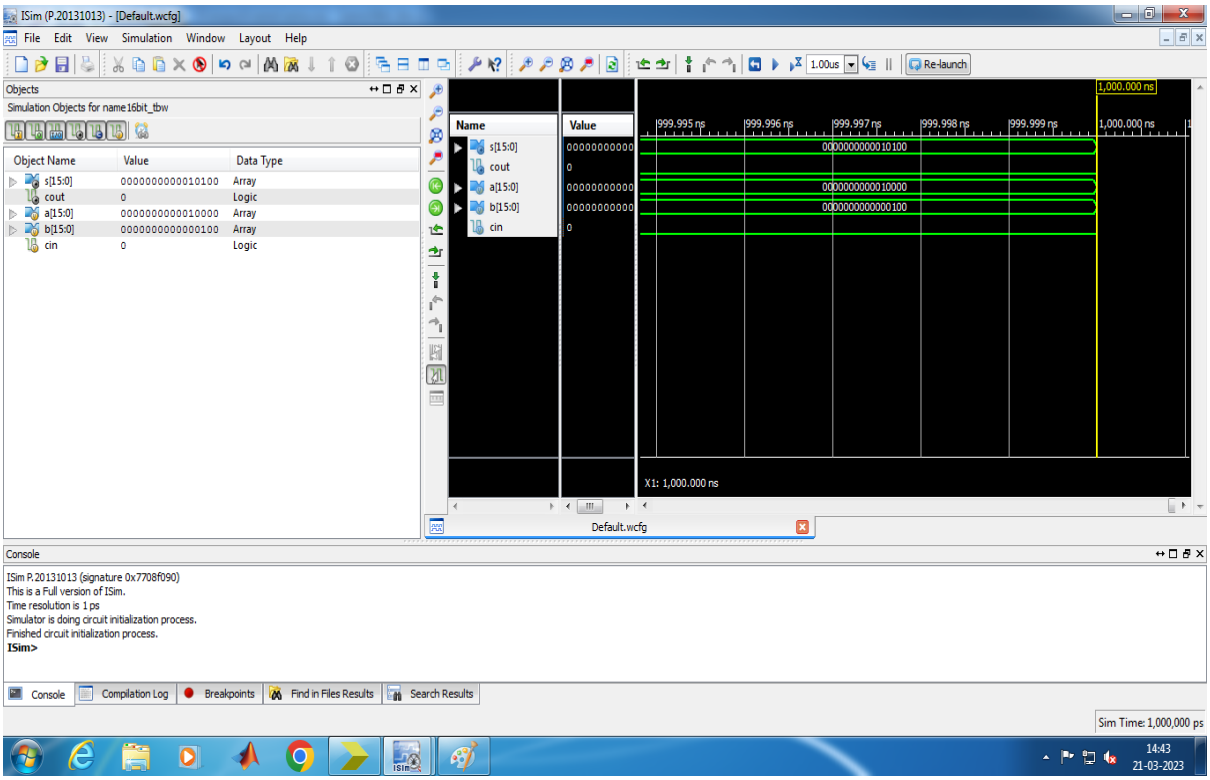


Fig 4.9 simulation of ripple carry adder

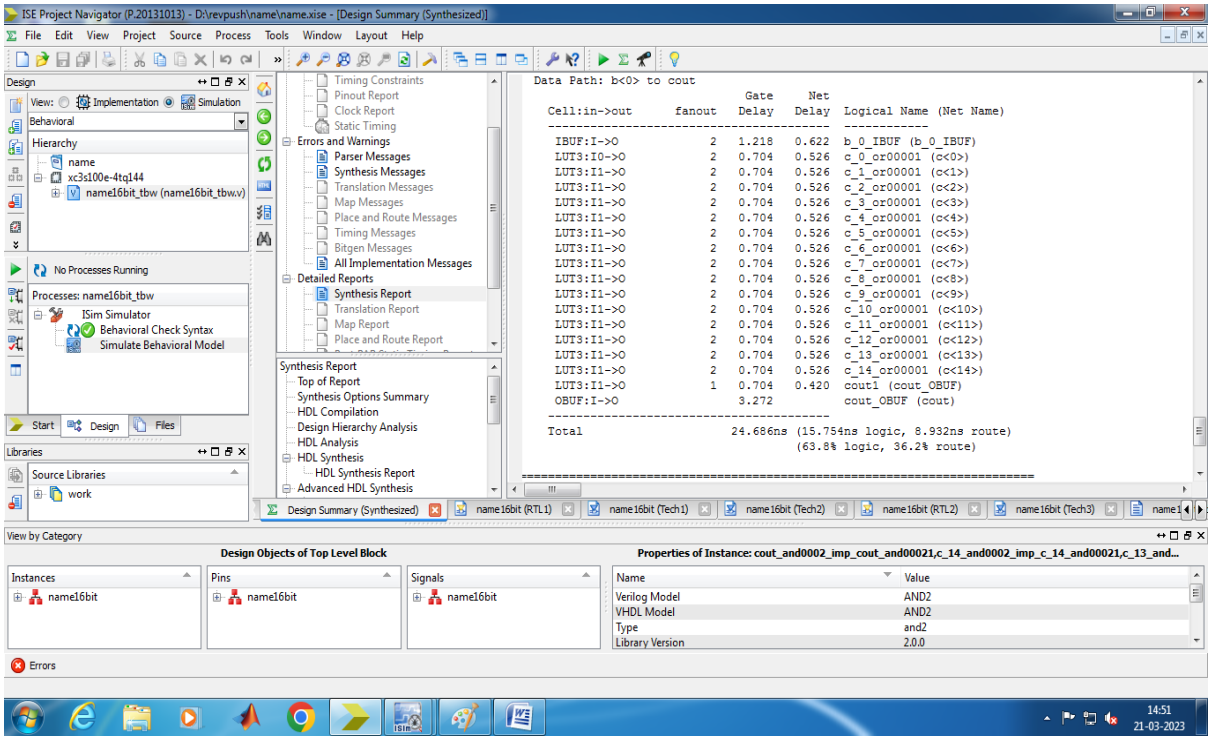


Fig 4.10 output of ripple carry adder

This output shows the values of time delay,area of the ripple carry adder.

4.3 Implementation of Approximate adder

In this module appoximate adder is implemented. The LUT view of appoximate adder is shown in (Fig 4.11) and simulation is shown in (Fig 4.12).

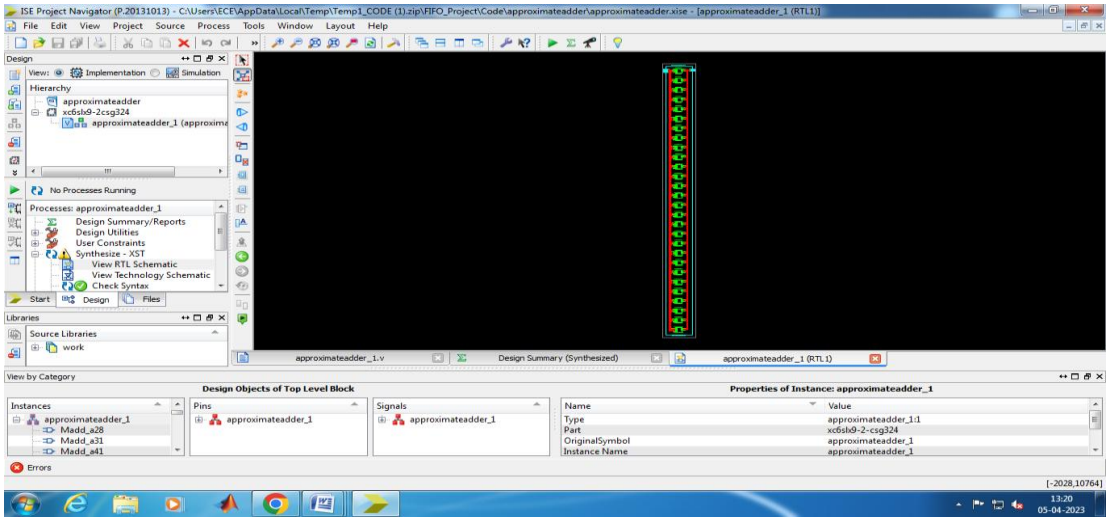


Fig 4.11 LUT view of APPROXIMATE ADDER

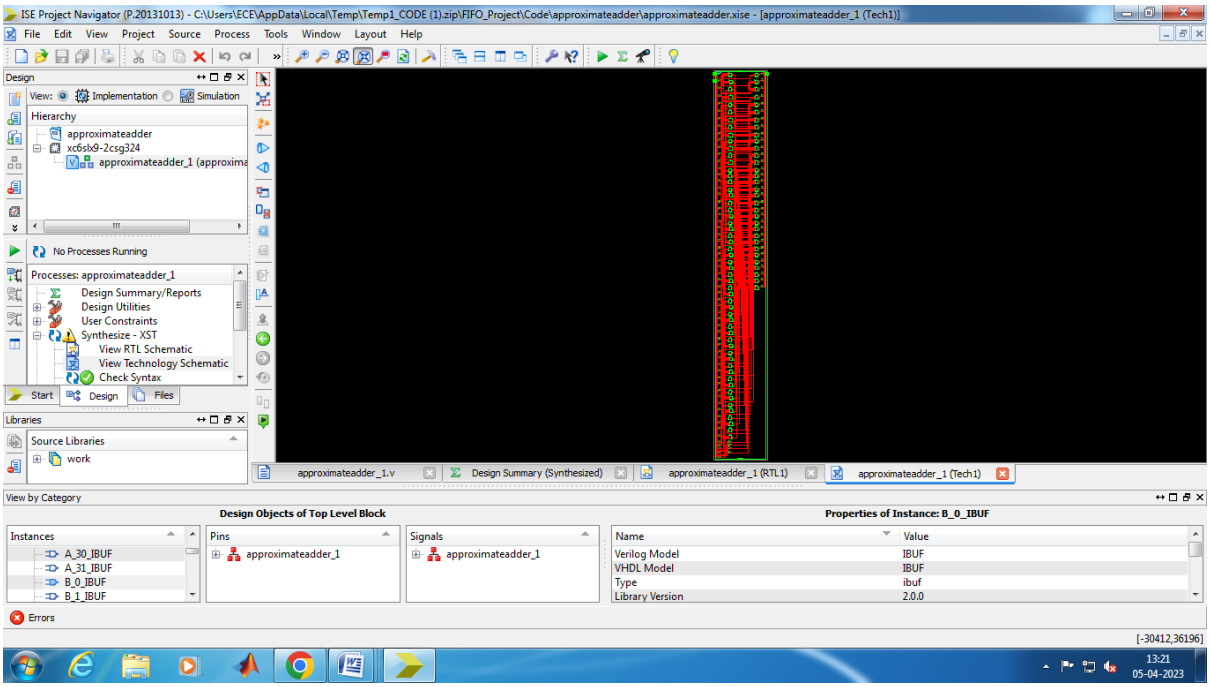


Fig 4.12 Technology mapping of approximate adder

```
Slice Logic Utilization:
 Number of Slice LUTs:                   87  out of   5720     1%
    Number used as Logic:                87  out of   5720     1%

Slice Logic Distribution:
 Number of LUT Flip Flop pairs used:     87
   Number with an unused Flip Flop:      87  out of     87   100%
   Number with an unused LUT:             0  out of     87     0%
   Number of fully used LUT-FF pairs:     0  out of     87     0%
   Number of unique control sets:         0

IO Utilization:
 Number of IOs:                          96
 Number of bonded IOBs:                  96  out of    200    48%

Specific Feature Utilization:
=========================================================================
Timing constraint: Default path analysis
  Total number of paths / destination ports: 346 / 32
-------------------------------------------------------------------------
Delay:               7.595ns (Levels of Logic = 4)
  Source:            A<27> (PAD)
  Destination:       Z<27> (PAD)

  Data Path: A<27> to Z<27>
                             Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     IBUF:I->O             5   1.328   0.949  A_27_IBUF (A_27_IBUF)
     LUT2:I0->O            3   0.250   1.221  Madd_a24_lut<4>1 (Madd_a24_lut<4>)
     LUT6:I0->O            1   0.254   0.681  Madd_a24_xor<4>11 (Z_27_OBUF)
     OBUF:I->O                 2.912          Z_27_OBUF (Z<27>)
    ----------------------------------------
    Total                      7.595ns (4.744ns logic, 2.851ns route)
                               (62.5% logic, 37.5% route)

=========================================================================
```

Fig 4.13 output of approximate adder

This output shows the values of time delay,area of the approximate adder.

ADDER MODEL	TIME	POWER	AREA
APPROXIMATE ADDER	7.595ns	0.44mW	58nm
CARRY SELECT ADDER	17.501ns	0.63mW	45nm
16- BIT RIPPLE CARRY ADDER	24.686ns	0.92mW	38nm

Table 4.4 Performance analysis of various components of adders

From the Table 4.4, it is found that area, power and time are increasing depending upon the adder size.

Carry Select Adder offers fast operation but consumes more power and occupies more area. Ripple Carry Adder is a simpler design with power consumption and smaller area, but it operates at a slower speed. Approximate adders trade accuracy for improved performance in time, area, and power, with varying degrees of approximation. The selection of an appropriate adder type depends on the specific trade-offs desired for a given application, considering the importance of time, area, power consumption, and the required level of accuracy.

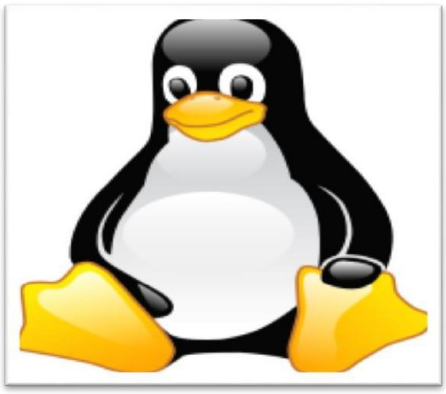
4.4 DCT implementation of Accurate adder:

INPUT IMAGE

DCT IMAGE

In DCT implementation of accurate adder above input image is converted into the clear image which is DCT image this implementaion is done in the MATLAB.

4.5 Implementation of DCT algorithm for sa3ov1:



INPUT IMAGE

DCT IMAGE

In DCT implementation of Algorithm for sub adder above input image is converted into the clear image which is DCT image this implementaion is done in the MATLAB.

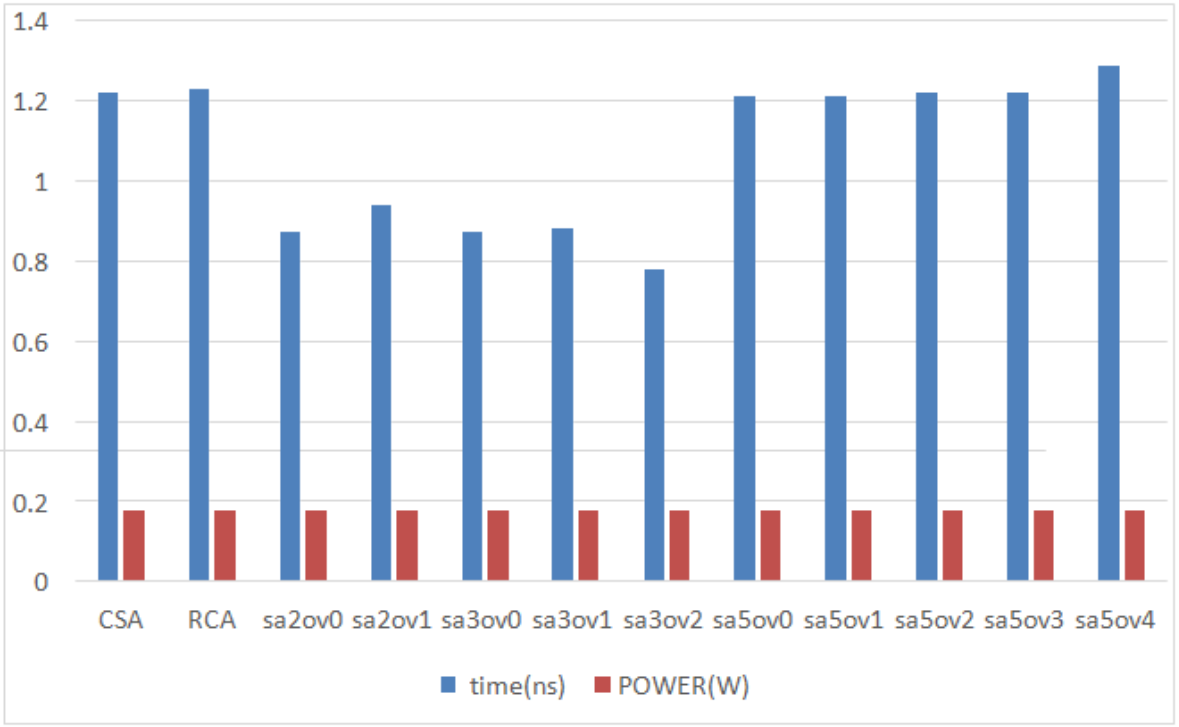
4.6 DCT implementation for sa5ov3

INPUT IMAGE

DCT IMAGE

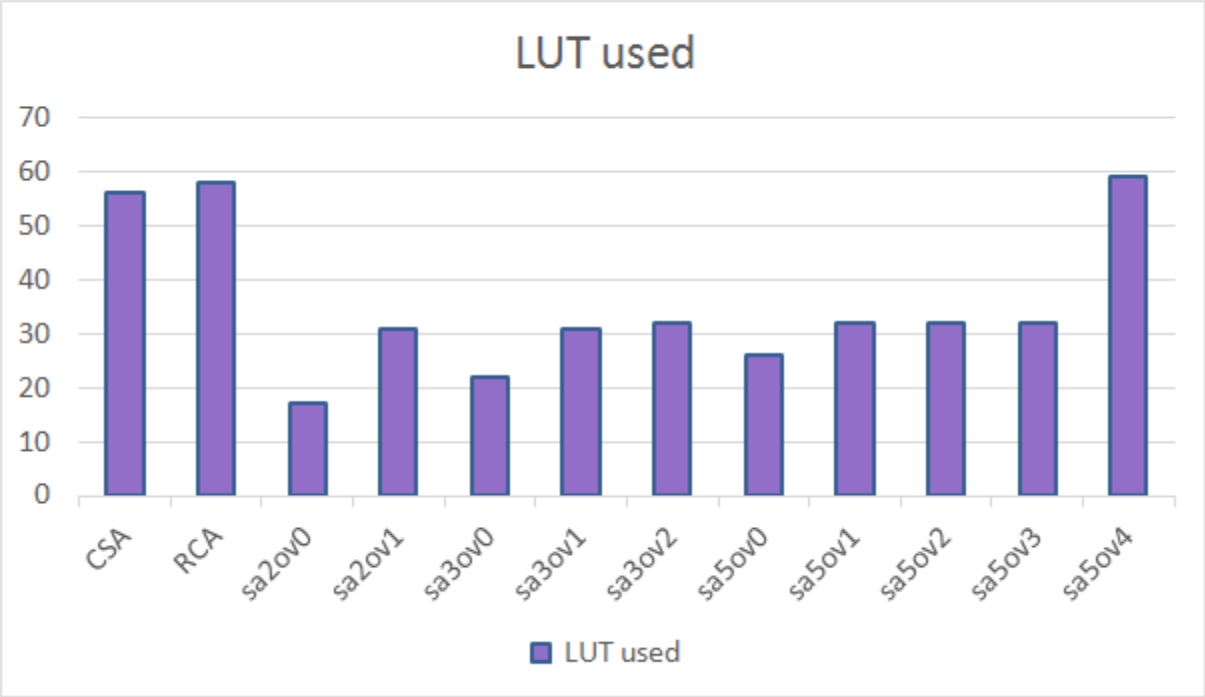
In DCT implementation of sub adder above input image is converted into the clear image which is DCT image this implementaion is done in the MATLAB.

4.7 Time and power consumption:



The above figure shows the time and power comparision of carry select adder and ripple carry adder and approximat e adder.

4.8 Performance results for proposed design:



The above figure shows the performance analysis comparision of carry select adder and ripple carry adder and approximate adder.

CHAPTER -5

CONCLUSION

Efficiency: The use of a single LUT for the adder design improves computational efficiency compared to conventional adders. By storing precomputed values in the LUT, the adder can perform addition operations with reduced delay and complexity, making it suitable for real-time aerial image processing applications.Resource Optimization: The single LUT design optimizes the utilization of resources, particularly in terms of hardware requirements. Compared to adders that rely on complex logic gates or multiple LUTs, the single LUT design reduces the need for additional components, resulting in a more compact and resource-efficient implementation. Accuracy: The adder with a single LUT maintains accuracy in the image processing applications. The precomputed values stored in the LUT are carefully chosen to ensure precise and reliable addition operations, thereby minimizing errors introduced during the processing of aerial images.Flexibility: The single LUT design offers flexibility in adapting to different image processing algorithms and requirements. The LUT can be customized and reprogrammed to accommodate various data types, precision levels, or specific processing tasks, allowing for versatility in handling different aerial image processing applications.Speed and Throughput: By leveraging the lookup table, the adder achieves fast computation and high throughput, enabling efficient processing of large volumes of aerial image data. This speed advantage contributes to real-time or near-real-time processing, crucial for applications such as object detection, image enhancement, or navigation systems in aerial imagery.Power Efficiency: The use of a single LUT for addition operations can lead to improved power efficiency compared to traditional adder designs. By reducing the complexity and number of logic gates, the overall power consumption of the adder can be minimized, which is beneficial for battery-operated or energy-constrained aerial image processing systems.

We can include application in conclusion

Applications

In addition to aerial image processing applications, the design and implementation of an adder with a single LUT can find applications in various other fields. Some of these applications include:

1. Digital Signal Processing: The adder can be used in digital signal processing applications such as audio and video processing, speech recognition, and telecommunications, where efficient and low-power arithmetic operations are essential.

2. Computer Vision: The adder can be employed in computer vision tasks, including object detection and tracking, facial recognition, augmented reality, and robotics vision systems. Its compact size and low power consumption make it suitable for integration into vision-based devices

3. Medical Imaging: In medical imaging applications like MRI (Magnetic Resonance Imaging), CT (Computed Tomography), and ultrasound, the adder can contribute to faster and more efficient image processing tasks, enabling quicker diagnoses and improving overall medical imaging systems' performance.

4. Data Compression: The adder can be utilized in data compression algorithms, such as JPEG (Joint Photographic Experts Group) and MPEG (Moving Picture Experts Group), to perform arithmetic operations required for encoding and decoding processes, resulting in more efficient and compact data representation.

5. Cryptography: The adder can play a role in cryptographic algorithms, such as symmetric and asymmetric key encryption, where arithmetic operations are crucial for secure data transmission, authentication, and privacy protection.

6. Machine Learning: The adder can be integrated into hardware accelerators for machine learning applications, including neural networks and deep learning algorithms, to perform arithmetic operations within the network layers efficiently, contributing to faster training and inference times.

7. Internet of Things (IoT): In IoT devices with limited computational resources and power constraints, the adder can be used for lightweight arithmetic operations, enabling efficient data processing and analytics at the edge, reducing the need for data transmission to

the cloud.

8. Financial and Trading Systems: The adder can be employed in financial and trading systems, where rapid and accurate arithmetic operations are required for real-time data analysis, risk assessment, and algorithmic trading.

9. Robotics: The adder can be utilized in robotics applications, such as robot control systems, motion planning, and sensor fusion, where efficient arithmetic operations are essential for real-time decision-making and precise control.

10. Gaming: In the gaming industry, the adder can be employed in graphics processing units (GPUs) for tasks like image rendering, physics simulations, and collision detection, contributing to faster and more realistic gaming experiences.

11. Wireless Communication: The adder can find applications in wireless communication systems, including modulation and demodulation schemes, error correction codes, and channel equalization, where efficient arithmetic operations are required for reliable and high-speed data transmission.

12. Audio Processing: The adder can be used in audio processing applications, such as audio mixing, filtering, and equalization, where real-time arithmetic operations are necessary for audio signal manipulation and enhancement.

13. Video Processing: In video processing tasks, such as video encoding and decoding, video compression, and video analytics, the adder can facilitate efficient arithmetic operations, contributing to improved video quality, reduced bandwidth requirements, and faster processing times.

14. Scientific Computing: The adder can be applied in scientific computing applications, such as simulations, numerical analysis, and data processing in fields like physics, chemistry, and biology, where accurate and efficient arithmetic operations are fundamental.

15. Automotive Systems: In automotive systems, the adder can be utilized in various applications, including driver assistance systems, sensor fusion, autonomous driving algorithms, and in-vehicle communication networks, where fast and low-power arithmetic operations are crucial for real-time decision-making and vehicle control.

16. Financial Analysis: The adder can find applications in financial analysis and modeling, where complex mathematical calculations, such as portfolio optimization, risk assessment, and option pricing, are performed, enabling faster and more accurate financial

decision-making.

These are just a few examples of the wide range of applications where an adder with a single LUT can be beneficial. Its compact design, low power consumption, and efficient arithmetic capabilities make it a valuable component in various domains that require high-performance and energy-efficient computing..

References:

[1] YaswanthD , S. Nagaraj , R.VishnuVijeth-Design and analysis of high speed and using carry select adder 2020.low area vedic multiplier

[2] Shamim Akhter, Saurabh Chaturvedi, KilariPardhasardi - CMOS Implementation of Efficient 16-Bit Square Root Carry-Select Adder 2015.

[3] Junqi H, T.Nandha Kumar, Haider AbbasFabrizio Lombardi- Simulation-Based Evaluation of Frequency Upscaled Operation of Exact/Approximate Ripple Carry Adders 2017.

[4] YajuanHe,Chip-Hong Chang- A Power-Delay Efficient Hybrid Carry Lookahead/Carry-Select Based Redundant Binary to Two's Complement Converter 2008.

[5] Leonardo B, Morgana M, Claudio M, Eduardo A-Exploring Power-Performance-Quality Tradeoff of Approximate Adders for Energy Efficient Sobel Filtering 2018.

[6] Alish Kanani,Jigar Mehta, Neeraj Goel - ACA-CSU: A Carry Selection Based Accuracy Configurable Approximate Adder Design 2020.

[7] Morgana Macedo, Leonardo Soares , Bianca Silveira , Claudio M. Diniz , Eduardo A. C. da Costa - Exploring the Use of Parallel Prefix Adder Topologies into Approximate Adder Circuits 2017.

[8] Pallavi Saxena - Design of Low Power and High Speed Carry Select Adder Using Brent Kung Adder 2015.

[9] Sana Mazahir, Osman Hasan, Rehan Hafiz, Muhammad Shafique, and Jorg Henkel – Probabilistic Error Modeling for Approximate Adders 2016.

[10] Mary Christina Joy , Ansa Jimmy ,Tony C.Thomas, Manju I.Kollannur-Modified 16 bit Carry Select and Carry Bypass Adder Architectures for High Speed Operations 2020.

[11] Leonardo Bandeira Soares, Sergio Bampi-Approximate Adder Synthesis for Area- and EnergyEfficient FIR Filters in CMOS VLSI 2015.

[12] Vaibhav Gupta, Debabrata Mohapatra, Anand Raghunathan,- Low-Power Digital Signal Processing Using Approximate Adders 2013.

[13] TUAHA NOMANI , MUJAHID MOHSIN ,ZAHID PERVAIZ,AND MUHAMMAD SHAFIQUE- xUAVs: Towards Efficient Approximate Computing for UAVs—Low Power Approximate Adders With Single LUT Delay for FPGA-Based Aerial Imaging Optimization 2020.

[14] Farhad Ebrahimi-Azandaryani, Omid Akbari , Mehdi Kamal , Ali Afzali-Kusha , and Massoud Pedram-Block-Based Carry Speculative Approximate Adder for Energy-Efficient Applications 2020.

[15] Vincent Camus , Student Member, IEEE, Mattia Cacciotti , Jeremy Schlachter and Christian Enz-Design of Approximate Circuits by Fabrication of False Timing Paths: The Carry Cut-Back Adder 2018.

.