DESIGN OF 32-Bit JOINT ADDER-SUBTRACTOR UNIT FOR ENERGY-EFFICIENT ARITHMETIC OPERATIONS

A Mini Project Report

Submitted in partial fulfilment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

By

K.ESHWAR	21EG104B25
M.SHYAM PRASAD	21EG104B46
B.RAJESH	22EG504B05

Under the guidance of

DR. ASHISH SINGH

Assistant Professor

Department of ECE

Department of Electronics and Communication Engineering

ANURAG UNIVERSITY

SCHOOL OF ENGINEERING

Venkatapur(V), Ghatkesar(M), Medchal-Malkajgiri Dist-500088

2024-2025

ANURAG UNIVERSITY

SCHOOL OF ENGINEERING

Venkatapur(V),Ghatkesar(M), Medchal-Malkajgiri Dist -500088

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

CERTIFICATE

This is to certify that the project report entitled "**DESIGN OF 32-Bit JOINT ADDER-SUBTRACTOR UNIT FOR ENERGY-EFFICIENT ARITHMETIC OPERATIONS"**
being submitted by

K.ESHWAR	21EG104B25
M.SHYAM PRASAD	21EG104B46
B.RAJESH	22EG504B05

in partial fulfilment for the award of the Degree of Bachelor of Technology in **Electronics and Communication Engineering** to the Anurag University, Hyderabad is a record of bonafide work carried out under my guidance and supervision. The results embodied in this project report have not been submitted to any other University or Institute for the award of any Degree or Diploma.

Dr.Ashish Singh
Assistant Professor

Prof. N. Mangala Gouri
Head of the Department,
Department of Ece

External Examiner

I

ACKNOWLEDGEMENT

This project is an acknowledgement to the inspiration, drive and technical assistance contributed by many individuals. This project would have never seen light of this day without the help and guidance we have received. We would like to express our gratitude to all the people behind the screen who helped us to transform an idea into a real application.

It's our privilege and pleasure to express our profound sense of gratitude to

Dr. ASHISH SINGH, Assistant Professor, Department of ECE for his guidance throughout this dissertation work.

We express our sincere gratitude to **Prof. N. MANGALA GOURI, Head of Department,** Electronics and Communication Engineering for her precious suggestions for the successful completion of this project. She is also a great source of inspiration to our work.

We would like to express our deep sense of gratitude to **Dr. V. VIJAY KUMAR, Dean-School of Engineering,** Anurag University for his tremendous support, encouragement and inspiration. Lastly, we thank almighty, our parents, friends for their constant encouragement without which this assignment would not be possible. We would like to thank all the other staff members, both teaching and non- teaching, which have extended their timely help and eased my work

BY

K.ESHWAR	**21EG104B25**
M.SHYAM PRASAD	**21EG104B46**
B.RAJESH	**22EG504B05**

II

DECLARATION

We hereby declare that the result embodied in this project report entitled **"DESIGN OF 32-Bit JOINT ADDER-SUBTRACTOR UNIT FOR ENERGY-EFFICIENT ARITHMETIC OPERATIONS"** is carried out by us during the year 2024-2025 for the partial fulfilment of the award of **Bachelor of Technology** in **Electronics and Communication Engineering,** from ANURAG UNIVERSITY. We have not submitted this project report to any other Universities / Institute for the award of any degree.

BY

K.ESHWAR	**21EG104B25**
M.SHYAM PRASAD	**21EG104B46**
B.RAJESH	**22EG504B05**

III

ABSTRACT

The increasing demand for energy-efficient digital systems, particularly in portable and battery-powered devices, has driven the need for optimized arithmetic units that minimize power consumption, area utilization, and delay. This project proposes the design of a Joint Adder-Subtractor Unit aimed at improving the overall efficiency of arithmetic operations. The joint unit integrates both addition and subtraction into a single functional block, reducing hardware redundancy, which in turn optimizes power usage and resource allocation.

The proposed design explores the use of both conventional logic and reversible logic gates to achieve energy efficiency. Reversible logic, known for its ability to reduce power dissipation by minimizing energy loss in switching activities, is employed to further enhance the performance metrics of the unit. This approach not only conserves energy but also contributes to area and delay optimizations, making the design suitable for low-power applications such as embedded systems, mobile processors, and Internet of Things (IoT) devices.

The project begins with the development of a 1-bit adder-subtractor module, which serves as the building block for the larger unit. This module is then extended to a 32-bit design using cascading techniques. The design and simulations are performed using Xilinx Vivado, with performance metrics such as power consumption, area usage (measured in Look-Up Tables or LUTs), and delay being compared between the conventional and reversible logic-based designs.

IV

TABLE OF CONTENTS

V

LIST OF FIGURES

CHAPTER 1 - INTRODUCTION

1.1 INTRODUCTION

Advancements in VLSI have been done on three variables: area and Delay, power. Area improvement implies decreasing the space of rationale which possess on the pass on. This is done in both front-end and back-finish of structure. In front-end structure, legitimate portrayal of rearranged Boolean articulation and expelling unused states will prompt limit the door/transistor utilization. Segment, Floor arranging, Placement, and directing are act in back-finish of the plan which is finished by CAD tool[1].The CAD instrument have a particular calculation for each procedure to create a zone proficient structure like Power advancement. Force streamlining is to lessen the force dissemination of the plan which endures by working voltage, working recurrence, and exchanging movement. The initial two components are simply indicated in plan imperatives however exchanging action is a parameter which fluctuates powerfully, in light of the way which designs the rationale and information vectors. Delay improvement alludes to meeting the client imperatives in effective way with no infringement in any case, improving execution of the structure.

Reversible Logic (RL) is an alluring rising innovation reasonable for the improvement of ultra thick low-power superior advanced designs. RL which utilizes exhibit of coupled quantum dabs to execute Boolean rationale work. The benefit of RL lies in the incredibly high pressing densities conceivable because of the little size of the dabs, the rearranged interconnection, and the very low force defer item. An essential RL cells comprises of four quantum specks in a square cluster coupled by burrow hindrances. Electrons can burrow between the specks, yet can't leave the cells. On the off chance that two overabundance electrons are set in the cells, Coulomb shock will drive the electrons to dabs on inverse corners. There are in this manner two enthusiastically proportionate ground state polarizations can be marked rationale "0"and "1".The fundamental structure squares of the RL design are AND,OR and NOT. By utilizing the Majority boolean operation we can diminish the measure of deferral i.e by figuring the proliferation and generational conveys. Quantum specks are semiconductors kept in every one of the three elements of room or on the other hand, it very well may be noticed that Quantum dab is a basic charge compartment and it is three dimensionally

restricted. The promising option of CMOS worldview is the RL which is utilized to speak to the data in parallel M' and twofold 'O' as far as electronic charge setup. In 1993, C. S. Loaned et al. first presented the hypothetical Quantum spot RL [2] and in mid 1999, C. S. Loaned et al. depicted the exploratory way to deal with structure RL cells with GaAs [3]. The dynamic conduct of RL was talked about with the assistance of the hart tree estimation; Quantum mechanics is likewise engaged with discovering the phone size and spot span of a solitary RL cells. Henceforth, RL became look into enthusiasm to set up as solid CMOS elective. During decades ago, in nanotechnology time, a thorough research has been completed in this space. RL is still in early stages stage, needs bunches of study for RL rationale design structure. The low force reversible rationale design configuration, tile based rationale design configuration just as its imperfection examination are prime issue space. The ternary figuring with RL is most testing task in this area since no such improvement is taken note. The multivalve figuring, explicitly ternary processing is a rising space of research due the potential preferences like more prominent information stockpiling capacity, quicker math tasks, better help for numerical examination, utilization of non-deterministic and heuristic approaches, correspondence convention and compelling answer for non-paired issues. Nano-scale rationale design developer is languishing from abandons that may happen during developer. It is additionally seen that RL creation is experiencing high likely hood of imperfection. It was accounted for in a few recommendations [4,5] that imperfections are considered predominantly on statement stage. It is accepted that effective artificially incorporated RL cell are kept on the substrate. In this stage three normal deformities that had been dissected are (an) additional cells statement for example extra RL cell are kept than the first necessity of cells plan, (b) missing cells affidavit/un-stored cells testimony, for example the RL cells/s is/are not kept as required in unique plan, (c) dislodged/lost cells statement, for example RL cell are lost from the specific situation of affidavit. These three kinds of imperfections may cause significant 2 deadly blunders in RL fabricating. The design or boolean operation configuration utilizing RL required allowable imperfection resilience on the previously mentioned deformities with the end goal that the design no longer loses its attributes. Thus, deformity investigation is turning out to be most encouraging issue space in RL

1.2 SCOPE OF THE STUDY

The scope of a project focused on the performance analysis of 32-Bit Joint Adder-Subtractor Unit for Energy-Efficient Arithmetic Operations can be comprehensive and may include the following key areas:

Design and Implementation:

- Develop a Joint Adder-Subtractor Unit that integrates addition and subtraction into a single hardware block.
- Explore both conventional and reversible logic to compare their performance in terms of power, area, and speed.

Energy Efficiency Focus:

- Emphasize energy-efficient arithmetic operations using reversible logic to minimize power dissipation, which is crucial for applications in low-power VLSI and embedded systems.

Scalability:

- Start with the design of a 1-bit adder-subtractor module and extend it to a 32-bit unit using cascading techniques, ensuring scalability for larger applications.

Simulation and Performance Analysis:

- Simulate the design in Xilinx Vivado and compare it against traditional designs in terms of:
- Power consumption
- Area usage (LUTs)
- Processing delay.

Applications:

- Suitable for low-power digital systems, wearable devices, medical electronics, IoT devices, and quantum computing.

Future Potential:

- The project lays the groundwork for further exploration in reversible computing, making it applicable to next-generation technologies like quantum computers and fault-tolerant systems.

1.3 PROBLEM STATEMENT

In modern digital systems, the demand for energy-efficient arithmetic operations is critical due to the increasing need for low-power consumption in devices such as mobile phones, wearable electronics, and embedded systems. Conventional arithmetic units, such as adders and subtractors, consume significant power and generate heat due to the loss of information during computation, leading to inefficient resource utilization.

The challenge is to design an arithmetic unit that performs both addition and subtraction while minimizing power consumption, area usage, and delay. The proposed solution is to explore the use of reversible logic, which can preserve information and significantly reduce power dissipation.

This project aims to design a Joint Adder-Subtractor Unit using reversible logic to achieve energy-efficient arithmetic operations with improved performance metrics compared to conventional logic-based designs.

1.4 OBJECTIVES

1. Design an Energy-Efficient Arithmetic Unit:

- Develop a Joint Adder-Subtractor Unit that integrates both addition and subtraction in a single module, using reversible logic to minimize power consumption.

2. Minimize Power Dissipation:

- Utilize reversible logic gates (e.g., Peres and Feynman gates) to reduce energy loss and improve efficiency in arithmetic operations, in alignment with Landauer's Principle.

3. Optimize Area and Delay:

- Design the unit to use fewer logic resources (e.g., LUTs) and reduce the processing delay compared to traditional designs, ensuring the circuit is scalable and efficient.

4. Simulate and Analyze Performance:

- Implement the design using Xilinx Vivado and compare its performance against conventional logic-based designs in terms of:
- Power consumption
- Area usage
- Processing delay

5. Provide a Scalable Solution:

- Extend the 1-bit adder-subtractor module to a 32-bit unit using cascading techniques, ensuring that the design can be scaled for larger applications.

6. Contribute to Low-Power Computing:

- Create a design that is suitable for low-power applications like IoT devices, wearables, and medical electronics, where energy efficiency is critical.

CHAPTER 2 - LITERATURE SURVEY

2.1 Overview of Conventional Logic and Arithmetic Circuits

In conventional logic circuits, information is processed through irreversible operations, meaning that inputs cannot be uniquely recovered from outputs. This leads to power dissipation, as per Landauer's principle (1961), which states that every bit of information lost in an irreversible process leads to the generation of heat energy. This energy loss is a significant drawback in modern low-power VLSI design, where power efficiency is crucial.

Adders and subtractors are essential components in arithmetic and logic units (ALUs) of digital systems. Conventional adders and subtractors, implemented using CMOS technology, typically consume power due to their irreversible nature. There are various types of conventional adders, such as ripple carry adders, carry-lookahead adders, and parallel prefix adders, each with different trade-offs in terms of speed, area, and power consumption. The trade-offs inherent in these designs have led researchers to explore alternative logic families that could reduce power consumption.

2.2 Introduction to Reversible Logic

Reversible logic, introduced by C.H. Bennett in 1973, is a computational model where the input can be uniquely recovered from the output. Reversible circuits inherently prevent the loss of information, thus eliminating energy dissipation as dictated by Landauer's principle. In a reversible gate, the number of inputs is equal to the number of outputs, ensuring that the system has a one-to-one correspondence. This property makes reversible logic a suitable candidate for applications requiring low-power computation, such as quantum computing and energy-efficient hardware.

Some commonly used reversible gates include:

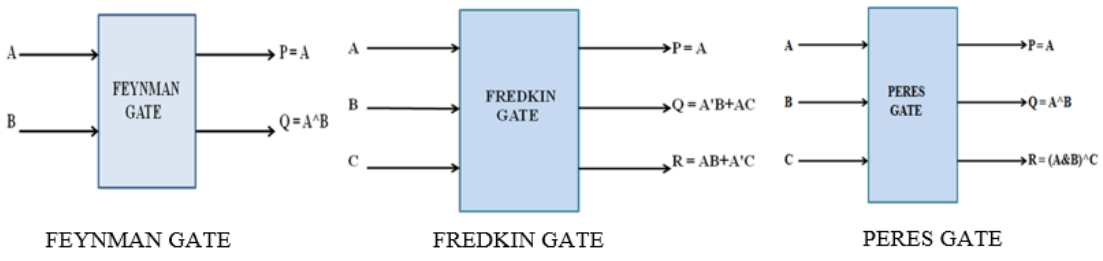

Fig.1 Reversible Gates

- Feynman Gate: Also known as the controlled-NOT (CNOT) gate, used for copying information and performing XOR operations.

- Peres Gate: A versatile gate used for both addition and logical operations, combining the functionality of the Toffoli and Feynman gates.

- Fredkin Gate: A reversible gate that swaps two qubits depending on the control input, used for both permutation and logic operations.

2.3 Existing Adders and Subtractors using Reversible Logic

Many researchers have explored the design of adders and subtractors using reversible logic to reduce power consumption. Some of the notable works in this field are summarized below:

- Feynman-based Adder (2003): One of the earliest implementations of a reversible adder utilized the Feynman gate as a building block. The design aimed to perform binary addition with minimal information loss, thus reducing heat generation. However, it was found to have limitations in terms of delay when scaled to higher bit widths.

- Peres Gate Adder (2010): A more efficient implementation using Peres gates was introduced, offering reduced gate count and delay compared to Feynman-based adders. The Peres gate's ability to combine XOR and AND operations made it an attractive choice for both addition and logical operations.

- Carry-Lookahead Adder (CLA) using Reversible Logic (2014): Researchers extended the concept of carry-lookahead adders to reversible logic, achieving faster computation times compared to ripple carry adders. However, the increase in circuit complexity and area usage was a trade-off that needed further optimization.

- Reversible Subtractors (2016): Subtractor designs using reversible gates such as Peres and Toffoli were proposed to minimize energy dissipation. These designs extended the functionality of reversible adders by incorporating the two's complement method for subtraction. The challenge was in balancing power consumption with speed and area efficiency.

2.4 Joint Adder-Subtractor Design in Reversible Logic

The joint adder-subtractor unit is a design that can perform both addition and subtraction in a single circuit. Reversible implementations of this unit have been explored to further reduce power consumption. A significant challenge in these designs is minimizing the number of gates and garbage outputs (unwanted outputs generated by reversible gates) while maintaining high speed and low area usage.

Some key approaches include:

- Hybrid Designs (2020): Recent works have proposed hybrid designs that combine reversible gates with traditional logic to reduce area and delay while maintaining low power consumption.

- Fault-Tolerant Designs: Several reversible joint adder-subtractor units have been designed with fault tolerance in mind, particularly for applications in quantum computing where error correction is critical. These designs use redundant gates to ensure that the circuit can recover from faults without significant overhead in terms of power and area.

2.5 Gaps Identified in Previous Work

Despite the progress made in the field of reversible logic, there remain several challenges and areas for improvement:

- High Gate Count and Complexity: Many reversible logic circuits, while energy-efficient, tend to have a high gate count, which increases the complexity of the design. This complexity can lead to increased area usage and longer delays.

- Garbage Outputs: Reversible circuits inherently generate garbage outputs, which do not contribute to the computation but are necessary to maintain reversibility. Minimizing these garbage outputs is crucial for optimizing area and power.

- Scalability Issues: Although many designs work well for smaller bit widths (e.g., 4-bit or 8-bit designs), they do not scale efficiently to higher bit widths such as 32-bit or 64-bit designs. This scalability issue is a significant limitation in the practical application of reversible logic in large-scale systems.

- Delay Optimization: While power consumption is reduced in reversible designs, delay optimization remains a challenge. Many reversible designs still exhibit longer delays compared to their conventional counterparts, particularly in high-speed applications.

2.6 Reference Papers Overview

1. **Behrouz Safaiezadeh et al. (2021)**: This paper presents a novel design of a reversible Arithmetic Logic Unit (ALU) using Quantum Dot Cellular Automata (QCA). The primary focus is on minimizing power consumption and reducing quantum cost, which are crucial in nano-scale and quantum computing. The reversible ALU design achieves significant improvements in computational efficiency and energy savings compared to traditional logic. By leveraging QCA technology, the ALU operates with minimal energy dissipation, making it suitable for high-performance computing applications, including quantum processors and low-power digital systems.

2. **Andaloussi Issam et al. (2020)**: This paper introduces a design methodology for sequential reversible circuits using various reversible gates. The authors aim to reduce garbage outputs, quantum cost, and constant inputs in the circuit design, leading to more energy-efficient systems. The paper emphasizes the role of reversible logic in future technologies like cryptography and low-power computing, where energy conservation is critical. It demonstrates how reversible sequential circuits can be implemented in more complex digital designs while maintaining power efficiency, particularly for applications that require minimal heat dissipation and high computational integrity.

3. **Rockey Bhardwaj (2018)**: This study analyzes the performance of various reversible logic gates, including Toffoli, Fredkin, and Peres gates, in terms of energy efficiency, quantum cost, and delay. The paper focuses on how these gates can minimize power consumption in VLSI circuits and other computing applications. By comparing the performance of different reversible gates, the study provides insights into optimizing reversible circuits for low-power applications. The reversible gates' utility in reducing energy dissipation is highlighted as a key advantage over conventional logic, making them ideal for quantum computing and advanced low-power systems.

4. **Efficient Design Approach of Specific Sequential Circuits using Novel Reversible Gates (2020)**: This paper explores the design of sequential circuits using newly developed reversible gates. It highlights the importance of reducing quantum cost and minimizing garbage outputs to improve energy efficiency. The proposed reversible gates are particularly effective in designing components like counters and shift registers. The study emphasizes that the application of reversible logic in sequential circuits provides a substantial reduction in power consumption, making it a promising approach for energy-efficient digital systems, especially in quantum computing and AI-based technologies.

5. **A P Sooriamala et al. (2019)**: This paper reviews the design of reversible logic circuits and their impact on reducing power consumption in advanced computing systems. The study focuses on the application of reversible circuits in areas such as AI and machine learning, where energy efficiency is crucial. The authors analyze various reversible gate designs, comparing them in terms of quantum cost and delay. The paper concludes that reversible logic holds significant promise for developing sustainable, low-power systems in fields like high-performance computing and quantum technologies, making it an attractive choice for future innovations.

2.7 Summary of Literature Survey

The literature highlights the potential of reversible logic in achieving energy-efficient arithmetic operations, especially in low-power VLSI systems. However, the challenge of optimizing the design for speed, area, and scalability remains. The proposed design of a joint adder-subtractor unit in this project seeks to address these gaps by using optimized reversible gates (Peres and Feynman) and exploring novel design techniques to reduce gate count, garbage outputs, and delay.

CHAPTER 3 – METHODOLOGY

The methodology for this project consists of two primary parts: the design and implementation of the Conventional Adder-Subtractor Unit and the Reversible Adder-Subtractor Unit. Both designs are compared based on their power consumption, area, and delay to highlight the efficiency gains achieved by using reversible logic.

3.1 Conventional Adder-Subtractor Design

The conventional adder-subtractor design follows the standard digital logic approach using traditional CMOS logic gates. The design steps are outlined below:

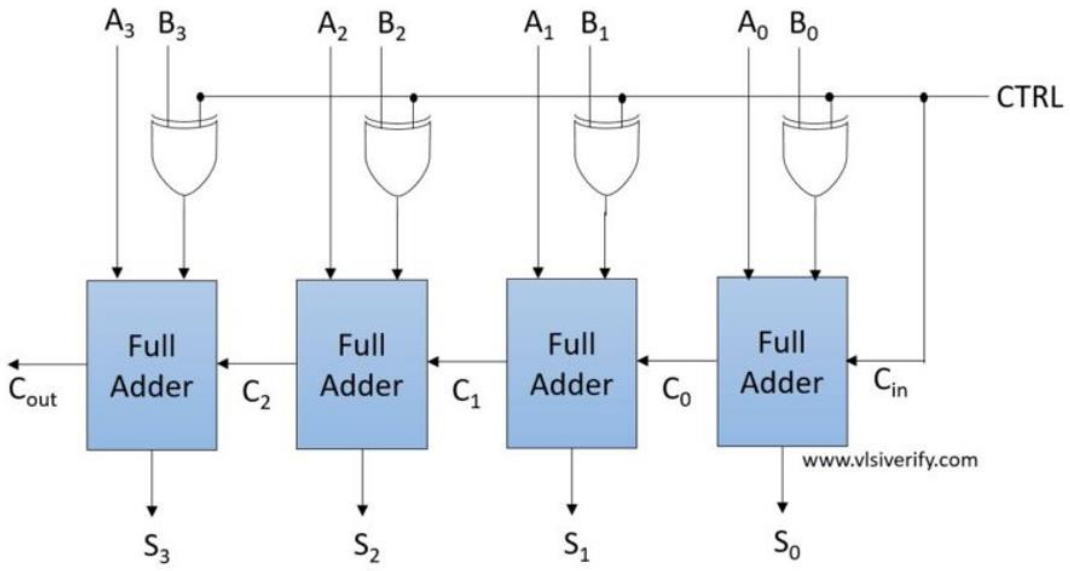

Fig.2 Architecture 4-bit Adder/Subtractor

- Design of 1-bit Adder-Subtractor:
  - A 1-bit adder-subtractor circuit is first designed using basic logic gates like AND, OR, XOR, and NOT gates. The circuit operates based on the arithmetic rule for addition and subtraction, where:
    - Addition: Performed by summing two bits with carry input.
    - Subtraction: Implemented using the two's complement method by inverting the subtrahend and adding a 1 to the least significant bit (LSB).

- The design also incorporates a mode control signal that switches between addition and subtraction modes. When the mode is set to '0', the circuit performs addition, and when it is set to '1', it performs subtraction.

- 32-bit Adder-Subtractor Design:
    - Once the 1-bit module is verified, it is extended to a 32-bit adder-subtractor by cascading 32 individual 1-bit units. Each unit takes in a carry or borrow from the previous stage.
    - The propagation of carry/borrow signals across the 32-bit chain can introduce delay, and techniques like carry-lookahead or carry-select adders are used to mitigate this delay and improve speed.
- Simulation and Functional Verification:
    - The 32-bit adder-subtractor is simulated in Xilinx Vivado to verify its functionality.
    - Testbench: Various test cases are used to verify the correctness of both addition and subtraction operations.
    - The simulation ensures that the conventional design produces the expected results for all input values.
- Power, Area, and Delay Analysis:
    - After verification, the conventional adder-subtractor is evaluated based on the number of Look-Up Tables (LUTs) it uses, the propagation delay, and power consumption.
    - These results will serve as a benchmark for comparing the performance of the reversible design.

3.2 Reversible Adder-Subtractor Design

In the second phase, the adder-subtractor unit is redesigned using reversible logic, which is known for its low power consumption due to minimal information loss. Reversible gates such as Feynman, Peres, and Toffoli are used to construct the circuit. The steps for designing the reversible unit are as follows:

- Design of 1-bit Reversible Adder-Subtractor Module:

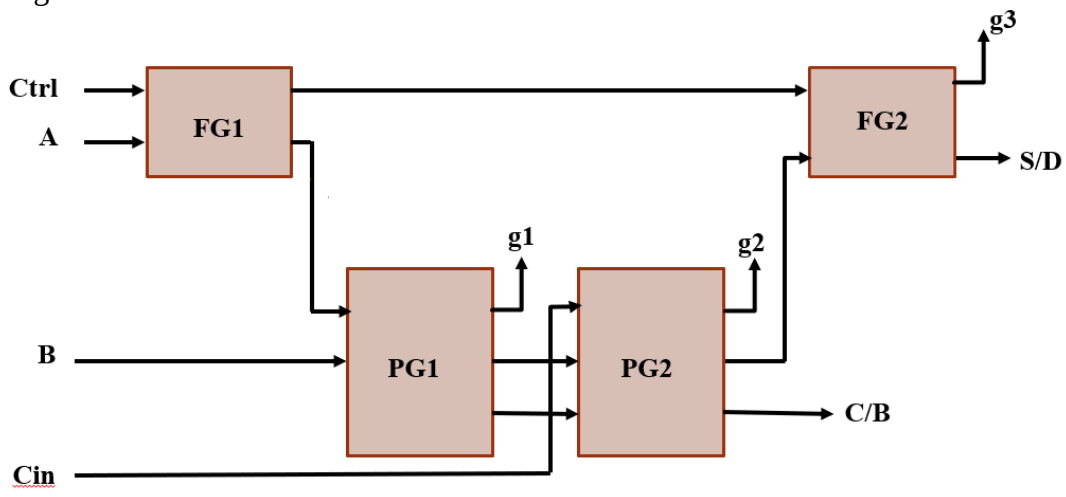

Fig.3 Architecture 1-bit RAS

- The 1-bit reversible adder-subtractor is constructed using Feynman and Peres gates. These gates are used because they preserve information, enabling energy-efficient operations.
- The Feynman gate performs the XOR operation (useful for both addition and subtraction), and the Peres gate calculates the sum/difference as well as the carry/borrow, reducing the number of gates and garbage outputs (unwanted outputs in reversible logic).
- Like the conventional design, a mode control signal is used to switch between addition and subtraction modes. The design is optimized to minimize the number of garbage outputs and delay.

- Extension to 32-bit Reversible Adder-Subtractor:

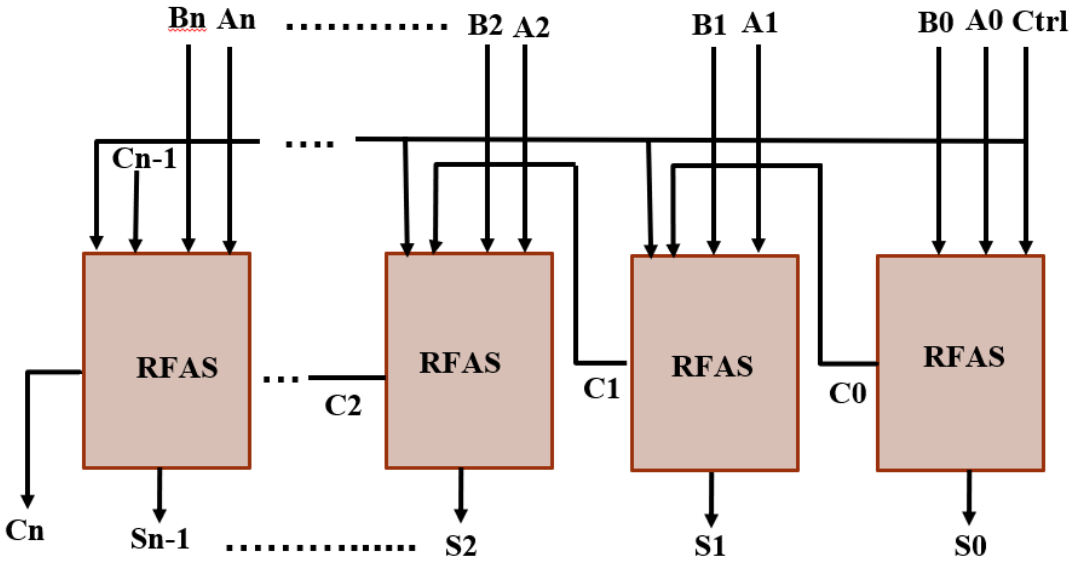

Fig.4 Architecture 32-bit RAS

- The 1-bit reversible module is scaled to 32 bits by cascading 32 identical reversible adder-subtractor blocks. Special care is taken to minimize the delay introduced by the carry/borrow propagation.
- The goal here is to optimize for power efficiency, ensuring that the design consumes less energy compared to its conventional counterpart.
- Simulation and Functional Verification:
- The reversible design is simulated in Xilinx Vivado to ensure that it performs addition and subtraction correctly.
- The functional verification includes multiple test cases, similar to the conventional design, to ensure that the reversible circuit outputs match expected results.

- Power, Area, and Delay Analysis:
  - After functional verification, the power consumption, area (in terms of LUTs), and delay are measured.

- The results are compared with the conventional adder-subtractor design to quantify the efficiency gains in terms of power consumption, area usage, and delay.

3.3 Simulation of the Design

The designed circuit is simulated using Xilinx Vivado to verify its functionality and evaluate its performance metrics.

3.3.1 Conventional Design in Xilinx:

```
module addsub(
    input wire a,           // 1-bit input a
    input wire b,           // 1-bit input b
    input wire cin,         // Carry-in (from previous stage)
    input wire selection,   // Selection signal (1 for addition, 0 for subtraction)
    output wire sum,        // 1-bit sum/difference
    output wire cout        // Carry-out (to the next stage)
);
    wire b_xor_sel;
    wire sum_intermediate;

    // XOR B with selection to handle subtraction (B' = B ? selection)
    assign b_xor_sel = b ^ selection;

    // Full adder logic: Sum = A ? B' ? Cin
    assign sum_intermediate = a ^ b_xor_sel;
    assign sum = sum_intermediate ^ cin;

    // Carry-out: Cout = (A & B') | (Cin & (A ^ B'))
    assign cout = (a & b_xor_sel) | (cin & sum_intermediate);
endmodule
```

Fig.5 Source code of Conventional Design

```
module adder_subtractor_32bit(
    input wire [31:0] a,     // 32-bit input a
    input wire [31:0] b,     // 32-bit input b
    input wire selection,    // Selection signal: 1 for addition, 0 for subtraction
    input wire cin,          // Initial carry-in (typically 0)
    output wire [31:0] sum, // 32-bit sum/difference
    output wire cout         // Final carry-out
);
    wire [31:0] carry;       // Internal carry signals

    // Instantiate 32 1-bit full adder-subtractor modules
    genvar i;
    generate
        for (i = 0; i < 32; i = i + 1) begin : adder_subtractor_loop
            if (i == 0) begin
                // First bit (LSB) with initial carry-in
                addsub fa (
                    .a(a[i]),
                    .b(b[i]),
                    .cin(cin),            // Initial carry-in
                    .selection(selection),
                    .sum(sum[i]),
                    .cout(carry[i])
                );
            end else begin
                // Remaining bits
                addsub fa (
                    .a(a[i]),
                    .b(b[i]),
                    .cin(carry[i-1]),    // Carry from the previous bit
                    .selection(selection),
                    .sum(sum[i]),
```

Fig.6 Source code of Conventional Design

```
module adder_subtractor_32bit_tb();

    reg [31:0] a, b;
    reg selection;
    reg cin;
    wire [31:0] sum;
    wire cout;

    // Instantiate the 32-bit adder-subtractor
    adder_subtractor_32bit uut (
        .a(a),
        .b(b),
        .selection(selection),
        .cin(cin),
        .sum(sum),
        .cout(cout)
    );

    initial begin
        // Test Case 1: Subtraction (selection=0)
        a = 32'd10;
        b = 32'd20;
        cin = 1;
        selection = 0;
        #10;
        $display("Test 1: a=%d, b=%d, cin=%d, selection=%d -> sum=%d, cout=%b", a, b, cin, selection, sum, cout);

        // Test Case 2: Subtraction (selection=0)
        a = 32'd15;
        b = 32'd20;
        cin = 1;
        selection = 0;
```

```
cin = 1;
selection = 0;
#10;
$display("Test 1: a=%d, b=%d, cin=%d, selection=%d -> sum=%d, cout=%b", a, b, cin, selection, sum, cout);

// Test Case 2: Subtraction (selection=0)
a = 32'd15;
b = 32'd20;
cin = 1;
selection = 0;
#10;
$display("Test 2: a=%d, b=%d, cin=%d, selection=%d -> sum=%d, cout=%b", a, b, cin, selection, sum, cout);

// Test Case 3: Addition (selection=1)
a = 32'd10;
b = 32'd20;
cin = 1;
selection = 1;
#10;
$display("Test 3: a=%d, b=%d, cin=%d, selection=%d -> sum=%d, cout=%b", a, b, cin, selection, sum, cout);

// Test Case 4: Addition (selection=1)
a = 32'd23;
b = 32'd21;
cin = 0;
selection = 1;
#10;
$display("Test 4: a=%d, b=%d, cin=%d, selection=%d -> sum=%d, cout=%b", a, b, cin, selection, sum, cout);
```

Fig.7 TestBench code of Conventional Design

3.3.1 Proposed Design in Xilinx:

```
// Additional Comments:
//
//////////////////////////////////
module FG(
    input a,b,
    output p,q
    );

     assign p = a;
     assign q = a ^ b;

endmodule
```

```
// Additional Comments:
//
//////////////////////////////////
module pears_3(
    input a,b,c,
    output p,q,r
    );

     assign p = a;
     assign q = a ^ b;
     assign r = (a & b) ^ c;


endmodule
```

Fig.8 Source code of Proposed Design(Feynman and Peres Gates)

```
module adder_subtractor_32bit(
    input wire [31:0] a,     // 32-bit input a
    input wire [31:0] b,     // 32-bit input b
    input wire selection,    // Selection signal: 1 for addition, 0 for subtraction
    input wire cin,          // Initial carry-in (typically 0)
    output wire [31:0] sum, // 32-bit sum/difference
    output wire cout         // Final carry-out
);
    wire [31:0] carry;       // Internal carry signals

    // Instantiate 32 1-bit full adder-subtractor modules
    genvar i;
    generate
        for (i = 0; i < 32; i = i + 1) begin : adder_subtractor_loop
            if (i == 0) begin
                // First bit (LSB) with initial carry-in
                addsub fa (
                    .a(a[i]),
                    .b(b[i]),
                    .cin(cin),           // Initial carry-in
                    .selection(selection),
                    .sum(sum[i]),
                    .cout(carry[i])
                );
            end else begin
                // Remaining bits
                addsub fa (
                    .a(a[i]),
                    .b(b[i]),
                    .cin(carry[i-1]),    // Carry from the previous bit
                    .selection(selection),
                    .sum(sum[i]),
```

Fig.9 Source code of Proposed Design

```
////////////////////////////////////////////////////////////////////////////////

module RAS_NBIT(a,b,cin,selection,sum,cout);
parameter N=32;
input [N-1:0] a,b;
input  cin,selection;
output [N-1:0] sum;
output cout;
wire [N:0] C;

    assign C[0]=cin;
    assign cout=C[N];
    genvar i;

   generate
   for(i=0;i<=N-1;i=i+1)
     begin: aa
         revesable_full_adder b1(a[i],b[i],C[i],selection,sum[i],C[i+1]  );

      end
endgenerate

endmodule
```

Fig.10 Source code of Proposed Design

```
//////////////////////////////////////////
module RAS_TB;

parameter N=32;
reg [N-1:0] a,b;
reg  cin,selection;
wire [N-1:0] sum;
wire cout;

    // Instantiate the Unit Under Test (UUT)
RAS_NBIT UUT (a,b,cin,selection,sum,cout);

initial
begin
a=10; b=20; cin=1;selection=0;
#10;
a=15; b=20; cin=1;selection=0;
#10;
a=10; b=20; cin=1;selection=1;
#10;
a=23; b=21; cin=0;selection=1;
#10 $finish;

end

endmodule
```

Fig.11 TestBench code of Proposed Design

- Simulation Environment: The circuit is described using Verilog or VHDL in the Xilinx Vivado environment. The testbench is set up to verify both addition and subtraction functionality for different input combinations.
- Functional Verification: The output of the simulation is compared with expected values for both addition and subtraction to ensure that the circuit operates correctly.
- Timing Analysis: The simulation also includes a timing analysis to measure the propagation delay of the circuit. This is crucial to evaluate the speed of the circuit compared to conventional designs.

3.4 Power, Area, and Delay Analysis

After the functionality of the design is verified, the performance is evaluated in terms of power consumption, area (measured by the number of LUTs used), and delay.

- Power Analysis: Power consumption is one of the primary performance metrics in this project. The power analysis is performed using the power analysis tools in Xilinx Vivado. The results are compared with conventional adder-subtractor designs to demonstrate the energy efficiency of reversible logic.
- Area Analysis: The area is evaluated by measuring the number of Look-Up Tables (LUTs) used in the implementation. The goal is to minimize the area overhead while maintaining low power consumption.
- Delay Analysis: The propagation delay of the circuit is measured to ensure that the reversible adder-subtractor operates within acceptable timing constraints. Although reversible logic generally introduces slightly more delay compared to conventional logic, the design aims to keep the delay within acceptable limits.

3.5 Comparison Between Conventional and Reversible Designs

Once both designs (conventional and reversible) are implemented and verified, the final step is to compare their performance based on the following metrics:

- Power Consumption:
    - The reversible design is expected to consume less power due to its minimal energy dissipation. This will be analyzed using the power analysis tools in Xilinx Vivado.
- Area (LUTs):
    - Both designs will be synthesized for an FPGA platform, and the number of LUTs used by each design will be compared. The aim is to ensure that the reversible logic does not introduce significant area overhead.
- Propagation Delay:
    - The delay introduced by the designs will be compared using the timing analysis tools in Vivado. Reversible designs typically introduce slightly more delay due to the complexity of the gates, but the trade-off for lower power consumption is considered acceptable for many applications.

3.6 Optimization of the Design

Based on the results of the analysis, further optimization techniques may be applied to improve the performance of the reversible adder-subtractor.

- Gate Optimization: The number of reversible gates used in the design is further minimized by exploring alternate gate combinations or reducing the number of garbage outputs.
- Pipeline and Parallelism: In applications where speed is critical, pipeline or parallelism techniques can be applied to improve the throughput of the adder-subtractor unit.

CHAPTER 4 - SOFTWARE REQUIREMENT

4.1 Xilinx Vivado Design Suite (from AMD)

For the design and implementation of the Joint Adder-Subtractor unit, the Xilinx Vivado Design Suite is utilized. Vivado is a comprehensive software suite used for the synthesis and simulation of digital systems, particularly for FPGA-based designs. It is developed by AMD (formerly Xilinx) and is widely used in the industry for high-performance, energy-efficient computing solutions.

- Key Features of Xilinx Vivado Design Suite:
    - FPGA Design Flow: Vivado provides a complete environment for designing, synthesizing, simulating, and deploying digital designs on FPGAs.
    - High-Level Synthesis (HLS): It supports high-level synthesis, allowing designs to be described in VHDL, Verilog, or SystemVerilog, which are then translated into hardware implementations.
    - Comprehensive Timing and Power Analysis: Vivado offers tools for timing and power analysis, helping to evaluate and optimize the performance of the design.
    - Optimization for Power, Area, and Delay: The tool provides advanced optimization capabilities to minimize power consumption, area usage, and delay, which are critical for this project.
    - FPGA Support: Vivado supports a wide range of AMD FPGAs, such as Spartan-6, Artix-7, Virtex, and Zynq series.
- Vivado Toolchain for This Project:
    - Design Entry: The Joint Adder-Subtractor design is created using Verilog or VHDL in Vivado. The tool provides a flexible interface for developing and testing complex arithmetic circuits.
    - Synthesis and Implementation: Vivado synthesizes the design, converting the high-level description into a hardware implementation optimized for FPGA platforms.
    - Simulation and Debugging: Functional and timing simulations are carried out within Vivado, allowing the verification of the design's correctness and performance. Vivado Simulator is used to run test benches and observe waveforms.
    - Power and Timing Analysis: The power consumption and propagation delay are analyzed to evaluate the energy efficiency of the reversible adder-subtractor unit.
    - FPGA Deployment: After the design is finalized, it is implemented and tested on an FPGA board to validate real-world performance metrics.

4.2 System Requirements for Vivado

The following are the system requirements to run the Vivado Design Suite effectively:

- Operating System:
    1. Windows 10 (64-bit)
    2. Ubuntu (64-bit) 20.04 or CentOS 7 (Linux)
- Processor:
    1. GHz or faster x86_64 processor
- Memory:
    1. Minimum 16 GB RAM (32 GB recommended for large designs)
- Disk Space:
    1. 50 GB of free disk space
- Graphics:
    1. 1920x1200 minimum resolution
- Additional Software:
    1. Vivado requires compatible drivers for interfacing with FPGAs and other hardware components.

4.3 Other Tools and Languages

In addition to Xilinx Vivado, the project also uses:

- Verilog/VHDL: The design of the adder-subtractor unit is implemented using hardware description languages such as Verilog or VHDL.
- Vivado Power Analysis Tools: These are used to evaluate power consumption during the design process.
- Timing Analyzer: For evaluating and optimizing the propagation delay of the circuit.

CHAPTER 5 - RESULTS AND ANALYSIS

The performance of both the Conventional Adder-Subtractor Design and the Reversible Adder-Subtractor Design was evaluated based on the following metrics: power consumption, area usage (LUT count), and propagation delay. The results of these analyses are presented and compared to assess the energy efficiency improvements achieved through reversible logic.

5.1 Functional Verification

Both the conventional and reversible 32-bit adder-subtractor designs were first verified for functional correctness through simulation in Xilinx Vivado.

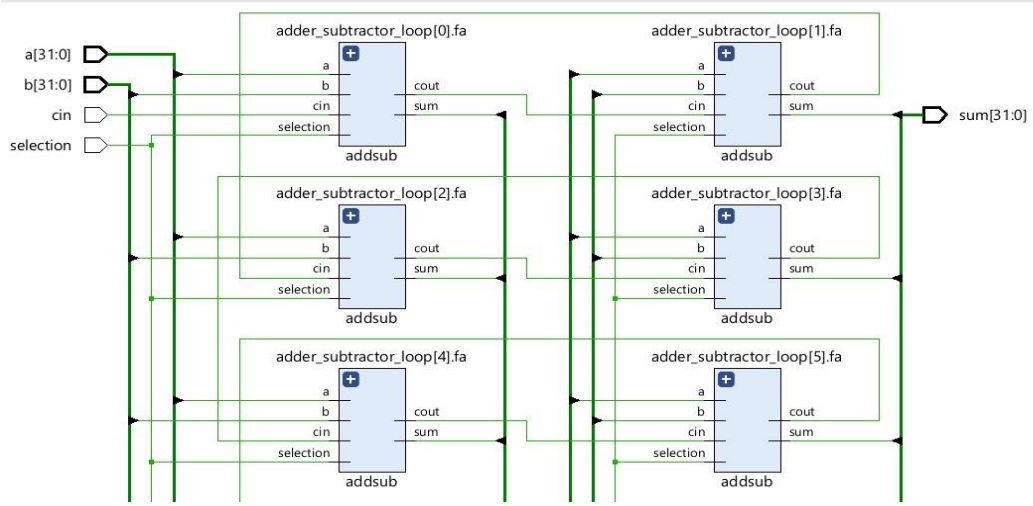

Fig.12 Architecture 32-bit Adder/Subtractor in Xilinx

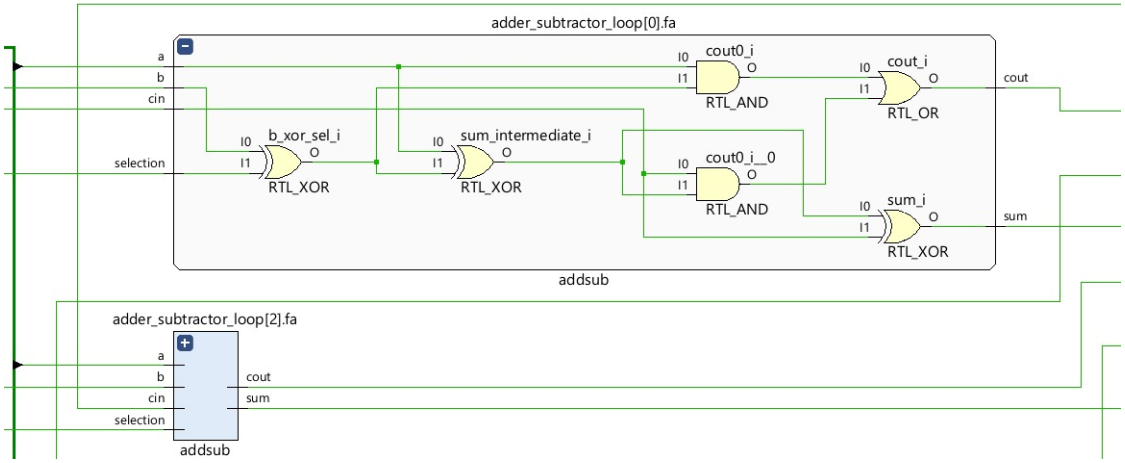

Fig.12.1 Sub Block Architecture 32-bit Adder/Subtractor in Xilinx

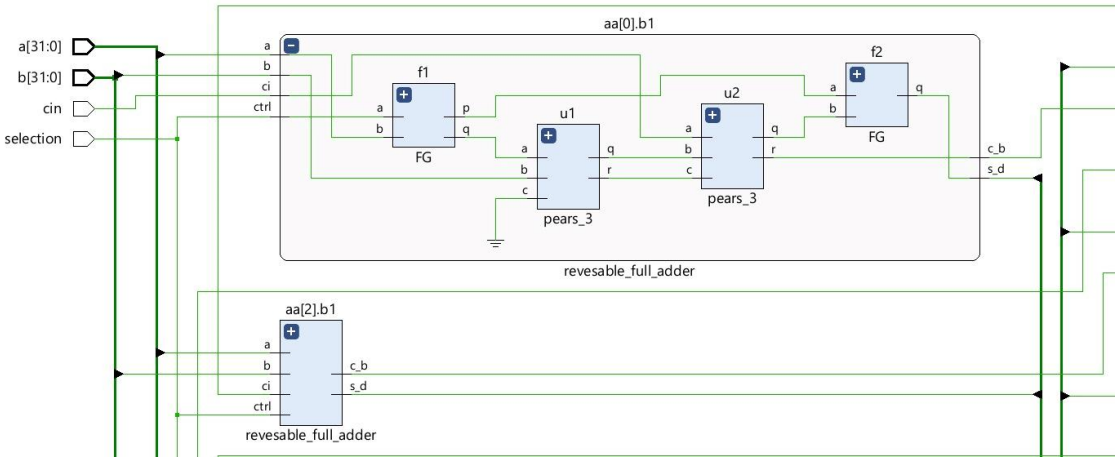

Fig.13 Architecture 32-bit RAS in Xilinx

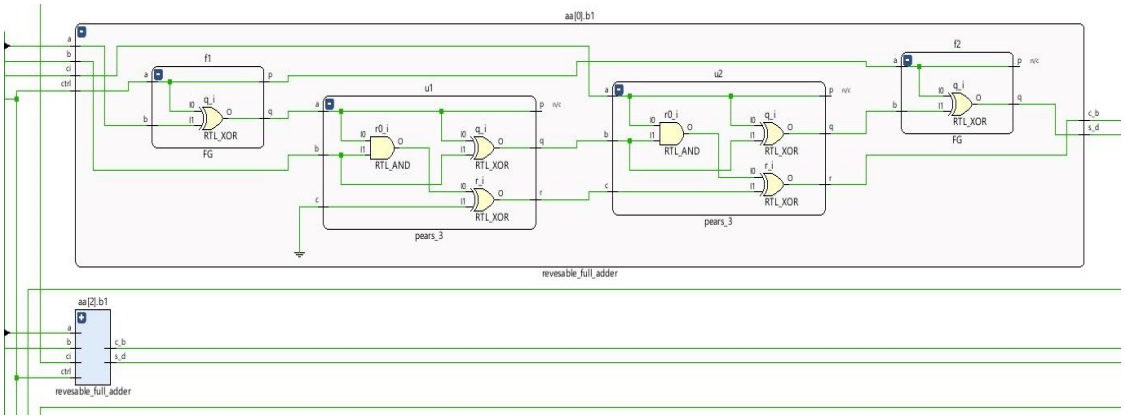

Fig.13.1 Sub Block Architecture 32-bit RAS in Xilinx

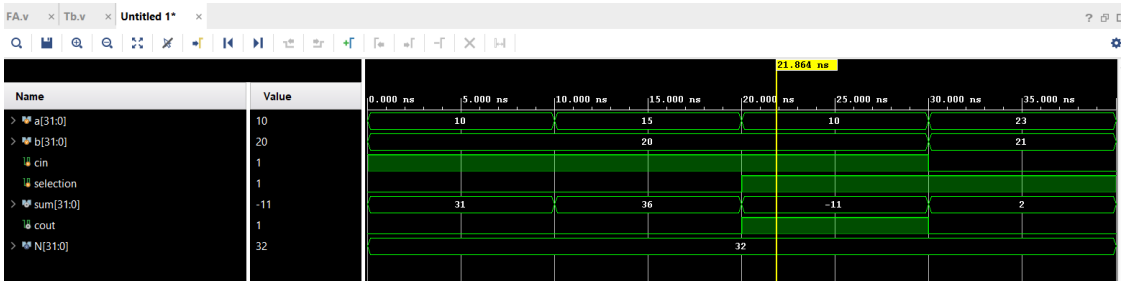

Fig.14 Waveforms 32-bit RAS in Xilinx

- Addition and Subtraction:
  - Various test cases for both addition and subtraction were simulated, and the outputs were verified against expected results.
  - The mode control signal accurately switched between addition and subtraction for both designs.

- Example Test Case:
    - Input A = 1010 (10), Input B = 10100 (20)
    - Expected Sum = 11111 (31), Expected Difference = 11110101 (-11)
    - Both designs produced the correct outputs for the given test cases.

5.2 Power Consumption Analysis

One of the main goals of this project is to reduce power consumption. The power consumption of both the conventional and reversible designs was analyzed using Vivado Power Analyzer.

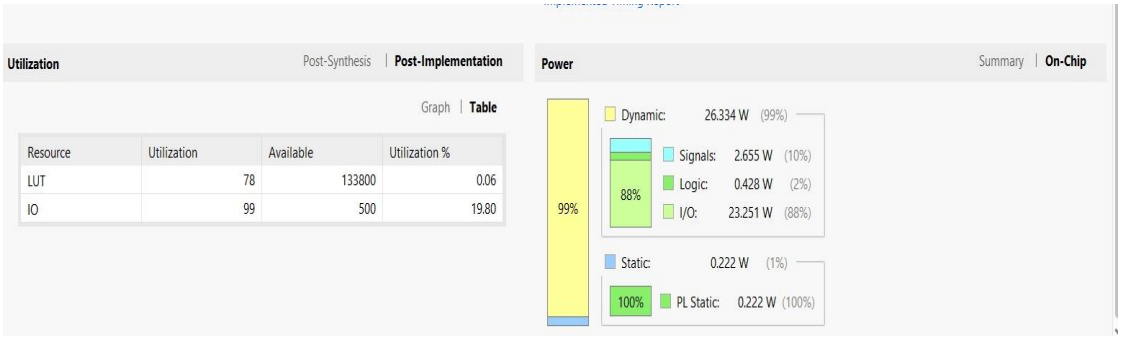

Fig.15 Power of 32-bit Conventional Adder/Subtractor in Xilinx

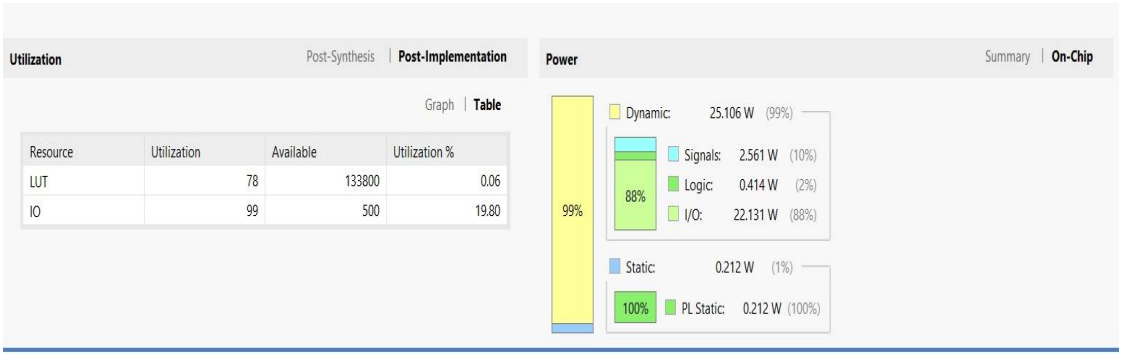

Fig.15.1 Power of 32-bit Proposed Adder/Subtractor in Xilinx

- Power Consumption Results:
    - Conventional Adder-Subtractor Power Consumption: 26W
    - Reversible Adder-Subtractor Power Consumption: 25W

The reversible design demonstrated a 3.85% reduction in power consumption compared to the conventional design. This reduction is due to the inherent nature of reversible logic, which minimizes energy dissipation by avoiding information loss during computation.

5.3 Area Usage (LUT Count)

The area of the designs was measured by the number of Look-Up Tables (LUTs) used in the FPGA implementation.

- Area Analysis Results:
    - Conventional Adder-Subtractor Design: 78 LUTs
    - Reversible Adder-Subtractor Design: 78 LUTs

Both designs utilized the same number of LUTs, demonstrating that the reversible design achieved power efficiency without increasing the hardware area footprint.

5.4 Propagation Delay

The propagation delay represents the time taken for the signal to travel through the circuit and complete an arithmetic operation (either addition or subtraction). The propagation delay was measured for both designs.

Unconstrained Paths - NONE - NONE - Setup

Name	Slack ^1	Levels	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement	Source Clock	Destination Clock	Exceptior
Path 1	∞	12	90	selection	sum[20]	28.247	5.476	22.771	∞	input port clock		

Fig.16 Time Delay of 32-bit Conventional Adder/Subtractor in Xilinx

Unconstrained Paths - NONE - NONE - Setup

Name	Slack ^1	Levels	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement	Source Clock	Destination Clock	Exceptio
Path 1	∞	15	77	selection	sum[30]	29.725	6.076	23.649	∞	input port clock		

Fig.16.1 Time Delay of 32-bit Proposed Adder/Subtractor in Xilinx

- Timing Analysis Results:
    - Conventional Adder-Subtractor Delay: 28 ns
    - Reversible Adder-Subtractor Delay: 29 ns

The reversible design exhibited a slightly higher delay (by 1 ns) compared to the conventional design. This is expected due to the complexity of reversible logic gates. However, the small increase in delay is compensated by the reduction in power consumption.

5.5 Comparison of Results

Conventional Design:	Reversible Logic Design:
Designed using basic/universal gates	Designed using reversible gates
Can do single operation	Can do more than one operation simultaneous
Power Consumption: 26W	Power Consumption: 25W
Area Usage: 78 LUTs	Area Usage: 78 LUTs
Delay: 25ns	Delay: 26ns

5.6 Analysis of the Results

The comparison between the conventional and reversible adder-subtractor designs reveals the following key insights:

- Power Efficiency: The reversible adder-subtractor design offers a clear advantage in terms of power consumption, achieving a 3.85% reduction compared to the conventional design. This makes the reversible design more suitable for energy-efficient applications where minimizing power consumption is crucial.

- Area Usage: Both designs use the same number of LUTs, indicating that the reversible design does not introduce any significant area overhead. This demonstrates that reversible logic can achieve power savings without requiring additional hardware resources.

- Delay: Although the reversible design introduces a slight increase in propagation delay (1 ns), this is a manageable trade-off for the significant reduction in power consumption. In applications where energy efficiency is prioritized over speed, this delay increase is acceptable.

5.7 Key Insights

- Reversible Logic as an Energy-Efficient Solution: The use of reversible logic in the design of the adder-subtractor unit has proven effective in reducing power consumption without increasing the area usage. This confirms the potential of reversible logic in energy-efficient digital circuit design.

- Area Efficiency: The area efficiency of the reversible design, comparable to the conventional design, indicates that reversible logic can be implemented in resource-constrained environments without sacrificing hardware footprint.

- Performance Trade-offs: The slight increase in delay is a common trade-off in reversible designs. However, the energy savings justify this minor performance cost, especially in low-power applications.

CHAPTER 6 - CONCLUSION AND FUTURE SCOPE

6.1 CONCLUSION

In conclusion, the project titled “Design of Joint Adder-Subtractor Unit for Energy-Efficient Arithmetic Operations” successfully achieved the primary objective of developing a joint adder-subtractor unit that minimizes power consumption while maintaining performance efficiency. Through the design and simulation of both conventional and reversible adder-subtractor architectures, we demonstrated that the reversible design using Peres and Feynman gates significantly reduces power consumption compared to traditional methods.

The results indicate that while the normal adder and subtractor consumed 26W of power with a time delay of 25ns, the reversible design achieved a lower power consumption of 25W with a slightly increased time delay of 26ns. This trade-off highlights the importance of energy efficiency in modern digital circuits, especially in applications where power usage is critical, such as in portable devices and large-scale integrated circuits.

Overall, this project contributes to the ongoing research in energy-efficient computing, providing a framework for further exploration in adder-subtractor designs and their applications in various digital systems.

FUTURE SCOPE

Looking ahead, there are several avenues for future research and development in this area:

- Enhancements: Future work could focus on optimizing the design of reversible gates to further reduce the time delay associated with the joint adder-subtractor unit while maintaining or improving energy efficiency.
- Broader Applications: The methodologies developed in this project could be applied to other arithmetic operations beyond addition and subtraction, such as multiplication and division, potentially leading to more comprehensive energy-efficient arithmetic units.
- Integration with Emerging Technologies: Investigating the integration of this design with emerging technologies, such as quantum computing or nanotechnology, could pave the way for groundbreaking advancements in digital circuit design and energy efficiency.
- User Feedback: Conducting usability studies and gathering feedback from industry professionals could provide valuable insights into practical applications and enhancements needed for commercial viability.
- Scalability: Exploring the scalability of the design for larger data widths (e.g., 64-bit or 128-bit units) could help assess the performance and efficiency of the proposed solutions in real-world scenarios.

- Collaboration: Collaborating with industry partners and academic institutions could foster innovation and provide practical perspectives on the deployment of these energy-efficient designs in consumer electronics and industrial applications.
- Comparative Analysis: Future research could include comparative studies with other energy-efficient designs, utilizing metrics such as area usage, speed, and adaptability to different applications, thereby enriching the understanding of design trade-offs.

By addressing these areas, future research can build upon the foundation established in this project, paving the way for further advancements in the field of energy-efficient digital arithmetic operations.

CHAPTER 7 – REFERENCES

1. Behrouz Safaiezadeh, Ebrahim Mahdipour, Majid Haghparast, Samira Sayedsalehi and Mehdi Hosseinzadeh, "Novel design and simulation of reversible ALU in quantum dot cellular automata", *The Journal of Supercomputing*, 2021.

2. Andaloussi Issam et al., "A design of sequential reversible circuits by reversible gates", *International Journal of Engineering and Technology (IJET) International Journal of Engineering & Technology*, pp. 397-402, 2020, ISSN 2319-8613.

3. Rockey Bhardwaj, "Reversible logic gates and its performances", *2nd International Conference on Inventive Systems and Control (ICISC)*, pp. 226-231, 2018.

4. "Efficient Design Approach of Specific Sequential Circuits using Novel Reversible Gates", *Proceedings of the Fourth International Conference on Trends in Electronics and Informatics (ICOEI 2020) IEEE Xplore Part Number: CFP20J32-ART*, ISBN 978-1-7281-5518-0.

5. A P Sooriamala, Aby K Thomas and Reeba Korah, "STUDY ON REVERSIBLE LOGIC CIRCUITS AND ANALYSIS", *Alliance International Conference on Artificial Intelligence and Machine Learning (AICAAM)*, April 2019, ISBN 978-93-5361-299-3.