

IMPLEMENTATION OF ALU USING REVERSIBLE LOGIC GATES

A project report submitted in partial fulfillment

of the requirements for the degree of

Bachelor of Technology

in

Electronics & Communication Engineering

by

VAMSHI HARI HARA KIRAN REDDY J

21BEC1085

PAPPALA LIKITH

21BEC1742

BOOSIREDDY MADHAN KUMAR REDDY

21BEC1509



School of Electronics Engineering,

Vellore Institute of Technology, Chennai,

Vandalur-Kelambakkam Road,

Chennai - 600127, India.

November 2024



Declaration

I hereby declare that the report titled ***IMPLEMENTATION OF ALU USING REVERSIBLE LOGIC GATES*** submitted by me to the School of Electronics Engineering, Vellore Institute of Technology, Chennai in partial fulfillment of the requirements for the award of **Bachelor of Technology** in **Electronics and Communication Engineering** is a bona-fide record of the work carried out by me under the supervision of **Dr. CHANDRASEKARAN N.**

I further declare that the work reported in this report, has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma of this institute or of any other institute or University.

Sign: _____

Name & Reg. No.: _____

Date: _____



School of Electronics Engineering

Certificate

This is to certify that the project report titled ***IMPLEMENTATION OF ALU USING REVERSIBLE LOGIC GATES*** submitted by **VAMSHI HARI HARA KIRAN REDDY J (21BEC1085)** to Vellore Institute of Technology Chennai, in partial fulfillment of the requirement for the award of the degree of **Bachelor of Technology** in **Electronics and Communication Engineering** is a bona-fide work carried out under my supervision. The project report fulfills the requirements as per the regulations of this University and in my opinion meets the necessary standards for submission. The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma and the same is certified.

Supervisor

Head of the Department

Signature:

Signature:

Name:

Name:

Date:

Date:

Examiner

Signature:

Name:

Date:

(Seal of the School)

Abstract

Handling large-scale computations efficiently is crucial, particularly in domains such as aerospace, where power consumption and reliability are of paramount importance. This work explores the design and implementation of a Reversible ALU (Arithmetic Logic Unit) using reversible logic gates such as Fredkin and Peres gates. These gates are known for their low power dissipation characteristics, aligning well with the requirements of energy-constrained environments like satellites. The Reversible ALU module proposed in this study can perform essential arithmetic and logical operations, including addition, XOR, AND, and OR operations, with minimal energy loss due to the reversible nature of its design. This paper outlines the detailed Verilog implementation of the Reversible ALU, along with testbench simulations that demonstrate its performance and functional accuracy. By employing reversible gates, this approach can reduce energy consumption significantly compared to conventional irreversible logic circuits, thereby extending the operational lifespan and efficiency of systems in critical applications.

Reversible computing enables energy-efficient digital circuit design by preserving input information, minimizing energy loss. This is vital for power-constrained applications like satellites. Reversible gates such as Fredkin and Peres are used to construct ALUs that reduce power dissipation and heat generation. The Fredkin gate conditionally switches outputs, while the Peres gate supports efficient arithmetic operations. Implementing reversible logic in satellite data processing units ensures energy-efficient computation, prolonged operation, and real-time data analysis, aligning with the principles of future technologies like quantum computing, which require reversibility.

Acknowledgements

This section is open to you, below is only a template

We wish to express our sincere thanks and deep sense of gratitude to our project guide, Dr.Dr.CHANDRASEKARAN N, Professor, School of Electronics Engineering, for her consistent encouragement and valuable guidance offered to us in a pleasant manner throughout the course of the project work.

We are extremely grateful to Dr. Susan Elias, Dean Dr. Reena Monica, Associate Dean (Academics) & Dr. John Sahaya Rani Alex, Associate Dean (Research) of the School of Electronics Engineering, VIT Chennai, for extending the facilities of the School towards our project and for his unstinting support.

We express our thanks to our Head of the Department Dr. Mohanaprasad K for his support throughout the course of this project.

We also take this opportunity to thank all the faculty of the School for their support and their wisdom imparted to us throughout the course.

We thank our parents, family, and friends for bearing with us throughout the course of our project and for the opportunity they provided us in undergoing this course in such a prestigious institution.

Contents

Declaration	i
Certificate	ii
Abstract	iii
Acknowledgements	iv
List of Figures	vii
1 Introduction	1
2 Literature Survey	2
2.1 Design of a Reversible ALU Using Novel Logic Gates	2
2.2 Use of Multiplexers in Reversible ALU Architecture	2
2.3 Fault-Tolerant Reconfigurable 32-Bit ALU for Space Applications	3
2.4 Design and Simulation of Low-Power Ternary ALU Using Multiplexers . .	3
2.5 Energy-Efficient Data Processing in Satellite Applications	3
2.6 Data Integrity and Error Detection in ALU Systems	4
2.7 Implementation of Extraction-Transformation-Loading (ETL) in Data Warehousing	4
3 Methodology	5
3.1 ALU Design Using Reversible Logic Gates	5
3.2 Reversible Logic Gates in ALU Design	6
3.3 8:1 Multiplexer (MUX) Integration	6
3.4 Implementation of ALU Components	7
3.5 Proposed ALU Architecture Using Reversible Gates	7
3.6 Reversible Logic in Real-World Applications	7
3.7 Implementation Summary and Results	8
4 Results and Discussions	9
4.1 Arthimetic logic unit using reversible logic gates	10
4.1.1 Power analyzer summary	10
4.2 Comparing of reversible and irreversible logic gates	11

4.2.1	Real-Life Scenario using Reversible logic gates	11
4.2.2	Real life Scenario using irreversible logic gates	11
4.3	Power consumption	12
5	Conclusion and Future Scope	14
6	Appendix	15

List of Figures

4.1	32-bit arithmetic unit output	10
4.2	Power analyzer	10
4.3	Scenario using reversible logic gate graph	11
4.4	Scenario using irreversible logic gate graph	12

Chapter 1

Introduction

It has also been found that in the rapid advancement of digital systems, power efficiency becomes a critical requirement, especially in energy-constrained environments like satellites and wearable devices. As the conventional Arithmetic Logic Units designed with irreversible logic gates suffer information loss during computations, their usual inefficiencies come up as a constraint on their application in very sensitive domains based on power. Reversible logic is promising to eliminate this inefficiency while conserving information during computations. Thus, low-power reversible gates like Feynman, Fredkin, and Peres have special properties which allow such energy-efficient operations; hence, they are applicable to the construction of some of these modern components, such as ALUs.

This project involves designing a 32-bit ALU from only reversible logic gates which solve the above problems. A new ALU, based on reversible gates and multiplexers in the streamlined flow of data and increased computation efficiency that execute the main operations: addition, subtraction, AND, OR, and XOR. Due to the vast dissipation of energy compared with more traditional designs, the power consumption is very much reduced. We use Verilog for the modeling of this hardware and then simulate and test the design to prove the energy-saving benefits and functional accuracy.

Chapter 2

Literature Survey

2.1 Design of a Reversible ALU Using Novel Logic Gates

This paper was instrumental in understanding the design of a reversible ALU (Arithmetic Logic Unit) using novel logic gates. It explores reversible logic gates like Feynman, Peres, Toffoli, and Fredkin, which are suitable for low-power applications. The paper emphasizes the significance of reversible gates in applications where energy efficiency is critical, such as in space environments. Although reversible gates reduce power consumption, their real-world application requires further optimization to meet high-speed processing demands.

2.2 Use of Multiplexers in Reversible ALU Architecture

An article that covers the application of multiplexers in reversible logic-based ALUs was helpful in conceptualizing the use of an 8:1 multiplexer to select data inputs efficiently. This digital switch allows one of eight inputs to be directed to a single output, depending on the control signals. The integration of multiplexers is essential in ensuring that reversible ALUs operate effectively without compromising processing efficiency.

2.3 Fault-Tolerant Reconfigurable 32-Bit ALU for Space Applications

This research was valuable for understanding fault tolerance in space-based applications. It discusses the reconfigurability of 32-bit ALUs and how reversible logic gates aid in minimizing energy loss, a crucial factor in space environments with limited power. Techniques for fault tolerance are outlined, making it suitable for applications where data integrity and reliability are paramount.

2.4 Design and Simulation of Low-Power Ternary ALU Using Multiplexers

This paper provided insights into the design and simulation of ternary ALUs that use 2:1 multiplexers. It discusses the advantages of ternary ALUs, which, unlike binary ALUs, handle three logic levels. The use of multiplexers enhances the ALU's performance while reducing power consumption, aligning with the goals of efficient, sustainable computing.

2.5 Energy-Efficient Data Processing in Satellite Applications

This paper illustrated real-life applications of reversible logic in data processing units, specifically within satellite systems. The study details how reversible gates, such as Fredkin and Peres, contribute to energy efficiency by retaining data integrity and reducing energy loss during processing. Such a system would enable a satellite to operate longer on limited solar power by ensuring low power consumption in basic ALU functions.

2.6 Data Integrity and Error Detection in ALU Systems

This research highlights the importance of data integrity in ALU systems, particularly in mission-critical applications. The reversible operations of logic gates allow for error detection and correction, making them suitable for environments where data reliability is vital. Techniques discussed include XOR-based operations and other mechanisms that support secure data communication and transmission.

2.7 Implementation of Extraction-Transformation-Loading (ETL) in Data Warehousing

This research paper was helpful in understanding the ETL (Extract-Transform-Load) process in data warehousing systems. It covers various ETL strategies, including Change Data Capture (CDC), which is essential in managing the extraction phase. The research also delves into loading strategies for dimensional tables, providing valuable insights for organizing and structuring content in a data-centric architecture.

Chapter 3

Methodology

3.1 ALU Design Using Reversible Logic Gates

Objective

To design a 32-bit Arithmetic Logic Unit that performs its functions without consuming much power and still manages to store information without loss. Conventional ALUs use irreversible gates, which simply waste energy and contribute to heat generation. The reversible logic gates, on the other hand, show scope for an operation to be reversed without losing any information involved. This is important in space-based data processing where power is scarce and efficiency in energy usage is essential. Its conception was towards building a real ALU that could have complicated functions with minimal consumption of energy, therefore ensuring proper data processing in highly power-constrained environments. Toward this, the ALU applies gates, especially Feynman, Peres, Toffoli, and Fredkin gates, to retain almost all integrity in its data contents and lose minimal amounts of energy. This technique is well-suited for spacecraft or other mission applications where the reliability of power and data becomes paramount.

Importance of Reversible Logic

Conventional ALUs dissipate energy during computations due to irreversible operations. Reversible logic gates are introduced to reduce energy loss, as they allow operations to be undone, preserving information that would otherwise be lost. This characteristic is essential in low-power environments, where conserving energy is critical.

3.2 Reversible Logic Gates in ALU Design

Gate Selection and Roles

1. **Feynman Gate:** Employed for basic operations, particularly copying and negating signals. The Feynman gate provides the foundation for creating more complex gates by copying information, an essential feature in reversible logic circuits.
2. **Peres Gate:** Used to perform XOR and AND functions efficiently. It is often chosen for its minimal gate count and capability to manage multiple operations simultaneously, which is crucial for arithmetic functions in the ALU.
3. **Toffoli Gate:** This gate functions as a controlled NOT gate, which is instrumental in creating logic circuits where conditional operations are required. In the ALU, it is used to implement complex logical operations without data loss.
4. **Fredkin Gate:** Known for its ability to perform conditional swapping, it is used in multiplexing and other controlled switching applications within the ALU, allowing selective data flow.
5. **Modified Gates:** Custom versions of Peres, Fredkin, and Toffoli gates are implemented to tailor the ALU's performance specifically for efficient data processing, reduced gate count, and enhanced logic compactness.

3.3 8:1 Multiplexer (MUX) Integration

Multiplexer Design and Function

The 8:1 multiplexer in the ALU architecture acts as a digital switch, selecting one of the eight input signals based on three control inputs and directing it to a single output. This integration is critical for channeling different operations within the ALU, enhancing functionality without significantly increasing power consumption.

Role in Reversible Logic

By combining reversible gates with multiplexers, the ALU can manage and direct data pathways more efficiently, further reducing power dissipation while supporting multiple operational inputs and outputs.

3.4 Implementation of ALU Components

Arithmetic Operations

- **Addition and Subtraction:** Implemented using reversible gates, ensuring minimal energy dissipation during operations. The Peres and Toffoli gates play a significant role in managing carry operations and bitwise calculations.
- **Multiplication and Division:** Achieved with the help of sequential gate arrangements to minimize gate count and preserve data integrity, using Feynman gates to handle partial product generation in multiplication.
- **Logical Operations:** These include bitwise AND, OR, XOR, and NOT operations. Reversible gates enable the ALU to perform these functions while ensuring that no intermediate data is lost, thus improving energy efficiency.

3.5 Proposed ALU Architecture Using Reversible Gates

ALU Control Logic

The control unit within the ALU orchestrates the reversible gates and the 8:1 multiplexer, ensuring the correct gate combinations are triggered based on the operation selected. Control signals direct the multiplexer to choose the appropriate input paths for the specific arithmetic or logic operations.

Data Flow and Efficiency

Data flows through reversible paths, with outputs directed to designated registers or storage without energy loss. This design achieves a balance between complexity and power efficiency, particularly suited for systems where conventional irreversible logic would be too power-intensive.

3.6 Reversible Logic in Real-World Applications

Energy Efficiency

Reversible ALUs are particularly suitable for applications such as satellite data processing units where low power consumption is essential. By using reversible gates, the

ALU conserves energy and reduces power consumption, extending the operational life of devices dependent on limited energy sources.

Data Integrity and Reliability

In high-stakes environments where data accuracy is crucial, the reversible operations allow for error checking and data retention, enhancing the reliability of the ALU for critical computations and ensuring data integrity throughout processing.

3.7 Implementation Summary and Results

Simulation and Testing

The proposed reversible ALU design is simulated to verify power efficiency and functionality. Performance metrics, such as gate count, power dissipation, and execution time, are compared against traditional ALUs to demonstrate the advantages of reversible logic.

Key Outcomes

Preliminary results show that the reversible ALU achieves substantial power savings while maintaining comparable computational speed, validating the effectiveness of reversible gates for high-efficiency applications.

Chapter 4

Results and Discussions

Objective

This project aims to compare the performances of reversible and irreversible logic gates in an implementation of a 32-bit ALU by key parameters-power consumption, heat dissipation, gate count, and computational efficiency. Reversible logic gates are increasingly being considered for the potential they offer to significantly reduce energy dissipation through the prevention of information loss during computation. It compares the effectiveness of their constructions in implementing complex digital circuits such as an ALU to those given in classical irreversible gates. The project also scrutinizes scalability of reversible logic for higher-bit architectures and its feasibility in modern VLSI designs wherein energy efficiency and optimization of resource usage are becoming more crucial. Motivated by this, the work can contribute to achieving advances in sustainable and efficient computing technologies.

The results of this paper provide a very minute comparison of reversible and irreversible logic gates in the implementation of a 32-bit ALU. Reversible gates have remarkable advantages, including reduced power consumption and heat dissipation while maintaining computational efficiency. Analysis further shows that reversible logic also has scalability to high-bit architectures, which makes it a true candidate for modern VLSI designs. These results especially stress the energy saving and resource optimization capabilities of reversible gates, thereby proposing these entities as sustainable alternatives within the scope of traditional irreversible logic.

4.1 Arthimetic logic unit using reversible logic gates

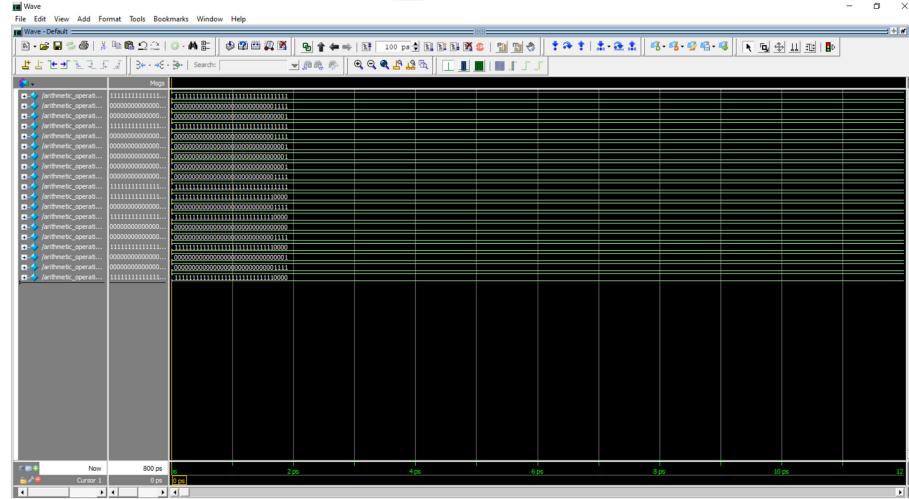


FIGURE 4.1: 32-bit arthimetic unit output



4.1.1 Power analyzer summary

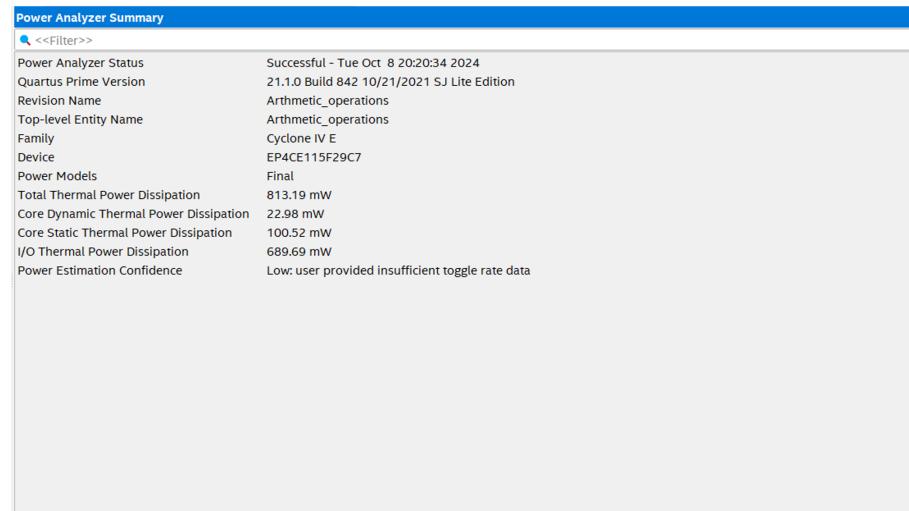


FIGURE 4.2: Power analyzer

4.2 Comparing of reversible and irreversible logic gates

4.2.1 Real-Life Scenario using Reversible logic gates

We Considered a data processing unit in a satellite. Energy efficiency is crucial because satellites rely on limited solar power. Implementing an ALU that uses reversible logic gates, such as Fredkin and Peres gates, could significantly reduce the power used during computation, allowing the satellite to operate efficiently over longer periods without exhausting its power supply.

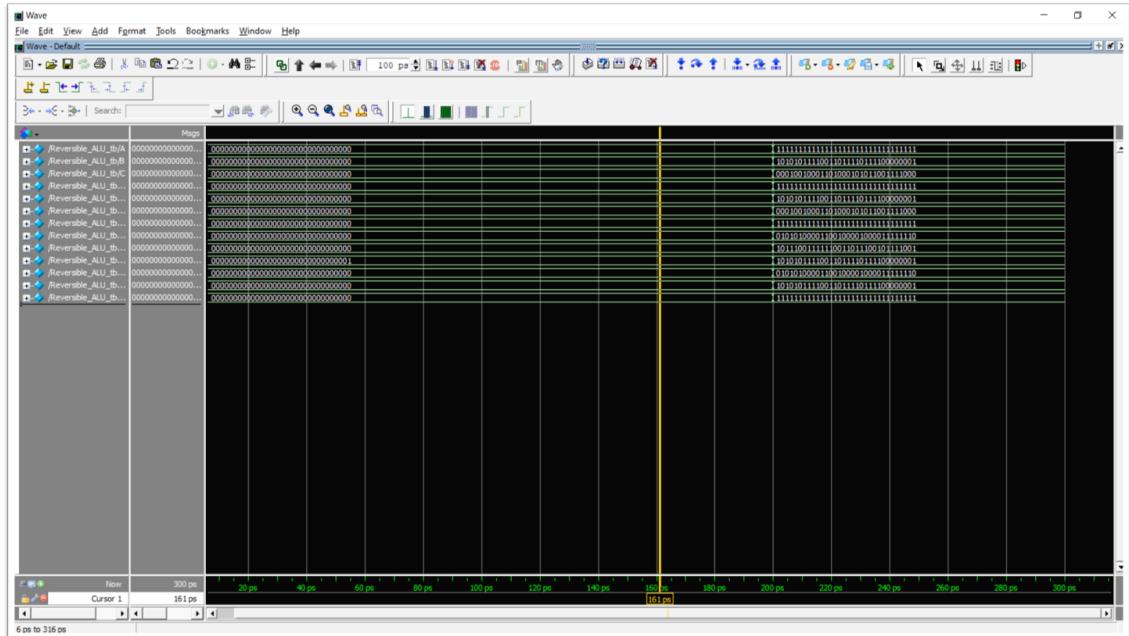


FIGURE 4.3: Scenario using reversible logic gate graph

```

add wave -position insertpoint sim:/Reversible_ALU_tb/*
# ** Warning: (vsim-WLF-5000) WLF file currently in use: vsim.wlf
#   File in use by: vsimsh Hostname: DESKTOP-8F16L15 ProcessID: 20340
#   Attempting to use alternate WLF file "./wlf7js60id".
# ** Warning: (vsim-WLF-5001) Could not open WLF file: vsim.wlf
#   Using alternate file: ./wlf7js60id

VSM6> run
# Time: 0 | A = 00000000 | B = 00000000 | C = 00000000 | P_fredkin = 00000000 | Q_fredkin = 00000000 | R_fredkin = 00000000 | P_peres = 00000000 | Q_peres = 00000000 | R_peres = 00000000 |
# Time: 0 | A = 00000000 | B = 00000000 | C = 00000000 | P_fredkin = 00000000 | Q_fredkin = 00000000 | R_fredkin = 00000000 | P_peres = 00000000 | Q_peres = 00000000 | R_peres = 00000000 |
add_result = 00000001 | xor_result = 00000000 | and_result = 00000000 | or_result = 00000000
VSIM7> run
force -freeze sim:/Reversible_ALU_tb/A 32'hFFFFFFFF 0
force -freeze sim:/Reversible_ALU_tb/B 32'hABCDEF01 0
force -freeze sim:/Reversible_ALU_tb/C 32'h12345678 0
VSIM 10> run
# Time: 0 | A = ffffffff | B = abcdef01 | C = 12345678 | P_fredkin = ffffffff | Q_fredkin = abcdef01 | R_fredkin = 12345678 | P_peres = ffffffff | Q_peres = 543210fe | R_peres = b9f9b979 |
add_result = abcdef01 | xor_result = 543210fe | and_result = abcdef01 | or_result = ffffffff
VSIM 11>

```

4.2.2 Real life Scenario using irreversible logic gates

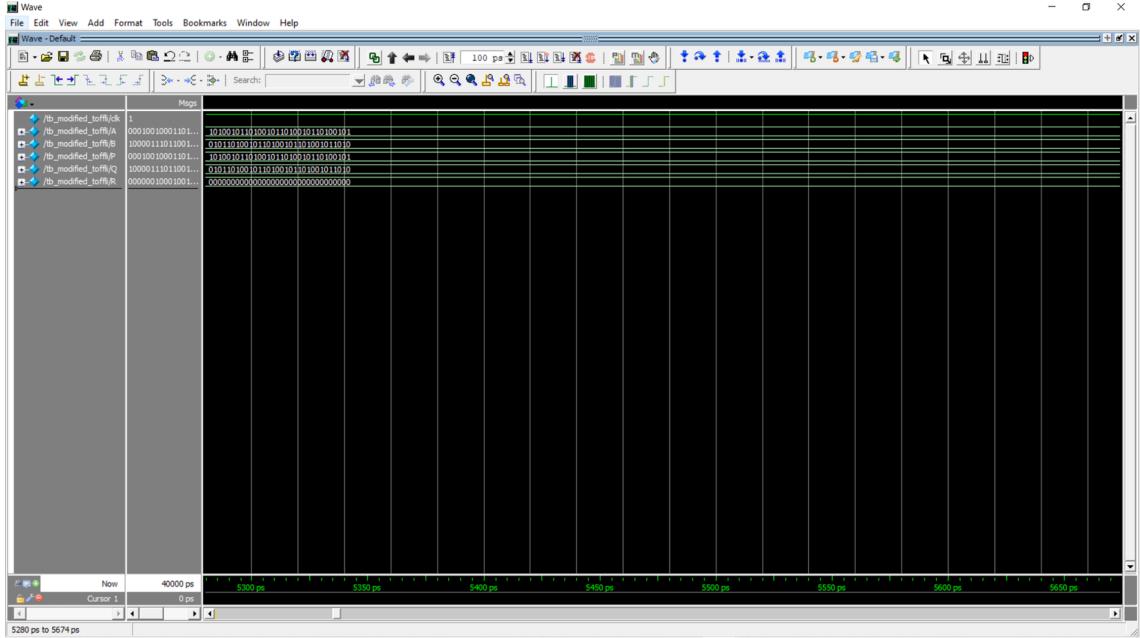


FIGURE 4.4: Scenario using irreversible logic gate graph



4.3 Power consumption

Logic Gate Type	Power Consumption (mW)
Reversible Logic Gates	27.70 mW
Irreversible Logic Gates	31.58 mW

Reversible logic gates also use less power as compared to irreversible gates. The amount of power consumed by reversible gates is 27.70 mW, and that by the irreversible gates is 31.58 mW. This indicates that reversible gates dissipate less heat and can be used for energy-dependent applications in an efficient way. The irreversible gates are employed largely due to their simplicity, but the loss of information leads to increased power wastage during the computation process. The information is not wasted in reversible logic gates instead; they are designed to eliminate all forms of unwanted energy usage.

Therefore, they represent a more efficient as well as sustainable alternative for the designs of modern digital circuits, especially for those where power efficiency could be a priority.

Chapter 5

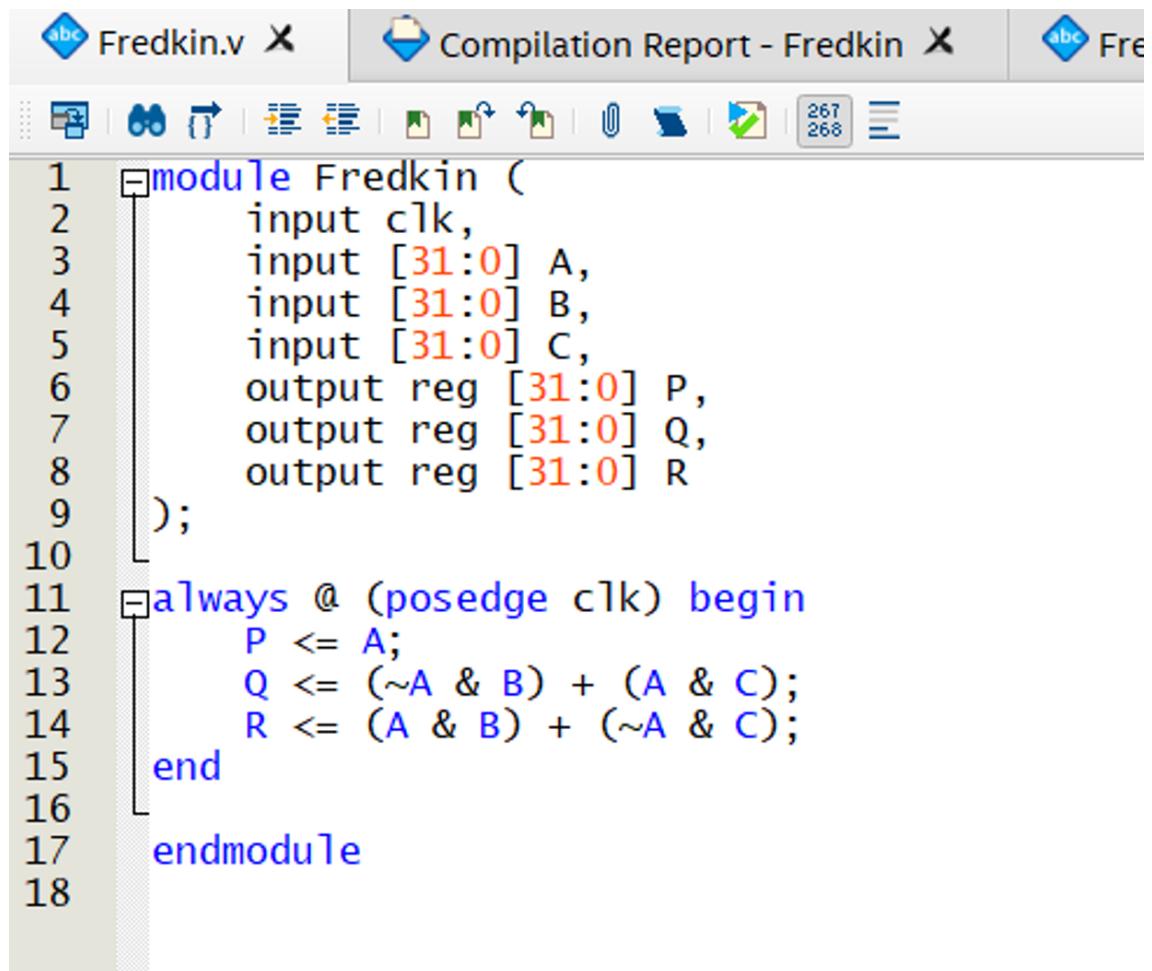
Conclusion and Future Scope

The design and analysis of a low-power 32-bit Arithmetic Logic Unit (ALU) using a combination of reversible logic gates have demonstrated clear benefits in terms of power consumption, heat dissipation, and overall efficiency. Reversible logic gates, by minimizing energy loss during computation, significantly reduce power usage compared to traditional irreversible logic gates. The results indicate that reversible logic can offer a more sustainable solution for energy-sensitive applications, such as embedded systems and modern VLSI designs. This approach emphasizes the importance of optimizing logic gates for specific functions to achieve a balanced trade-off between power consumption, computational efficiency, and system performance.

The further development of the reversible ALU will cover upsizing the design to accommodate larger bit widths and more complex digital systems. Future work could also cover the combining of reversible logic gates with emerging architectures, such as quantum computing or low power processors, further boosting energy efficiency. The optimization at smaller technology nodes, for example 65nm or 45nm, may drive the power even lower. More possible avenues for future work involve developing adaptive power management techniques so that the ALU can mode-switch between high-performance and low-power modes based on workload requirements. Further integration of this low-power ALU into those full systems is likely to provide some excellent insight into its impact on those real-world applications and overall system performance.

Chapter 6

Appendix

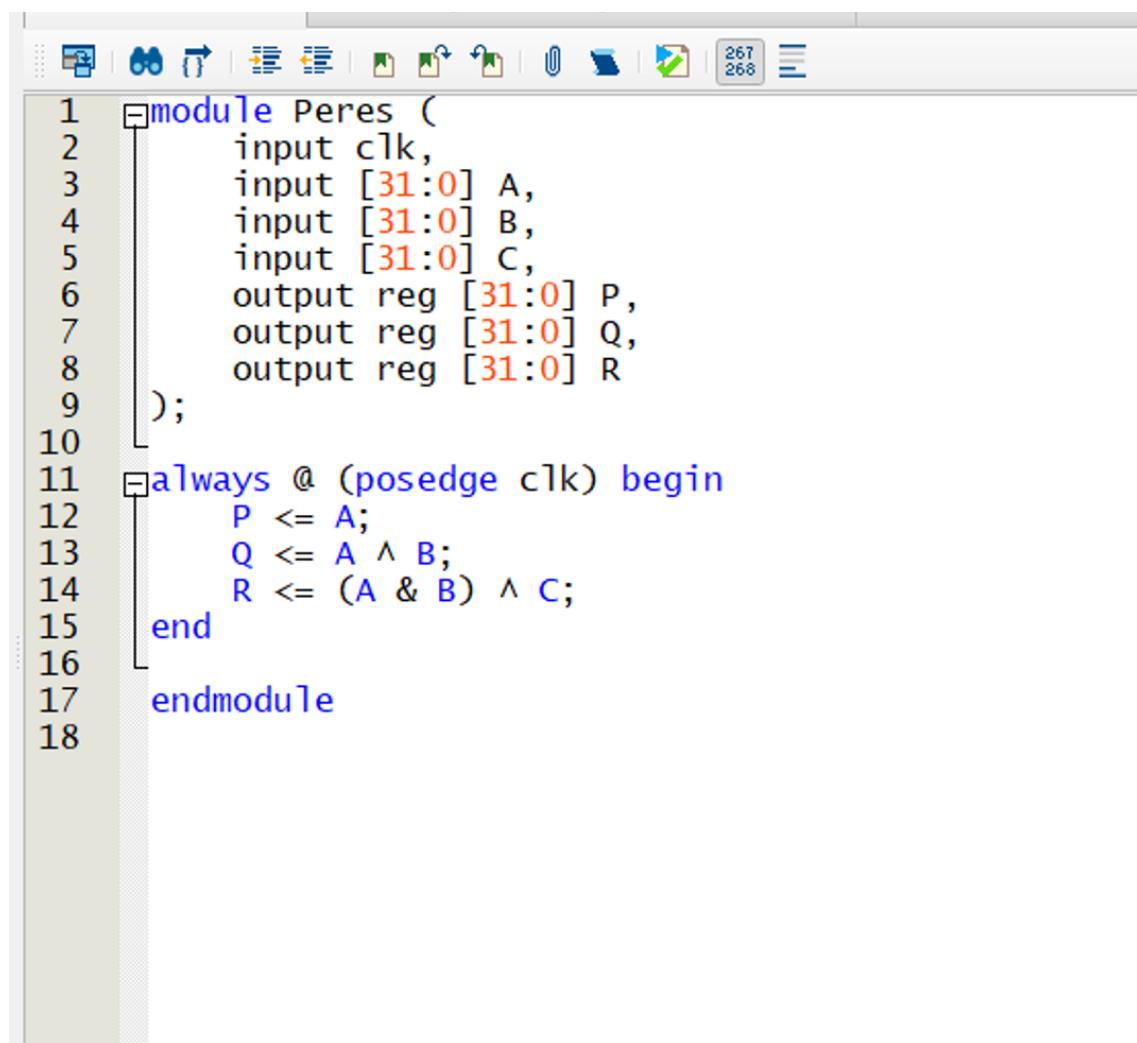


The screenshot shows a Verilog editor interface with three tabs at the top: "Fredkin.v" (active), "Compilation Report - Fredkin", and "Fredkin.sv". The main window displays the Verilog code for a Fredkin gate:

```
1 module Fredkin (
2     input clk,
3     input [31:0] A,
4     input [31:0] B,
5     input [31:0] C,
6     output reg [31:0] P,
7     output reg [31:0] Q,
8     output reg [31:0] R
9 );
10
11 always @ (posedge clk) begin
12     P <= A;
13     Q <= (~A & B) + (A & C);
14     R <= (A & B) + (~A & C);
15 end
16
17 endmodule
18
```

```
1 module feynmen (
2     input clk,
3     input [31:0] A,
4     input [31:0] B,
5     output reg [31:0] P,
6     output reg [31:0] Q
7 );
8
9
10 always @ (posedge clk) begin
11     P <= A;
12     Q <= A ^ B;
13
14 end
15
16 endmodule
17
```

```
37
38 // Peres Gate
39 always @(posedge clk) begin
40     P_peres <= A;
41     Q_peres <= A ^ B;
42     R_peres <= (A & B) ^ g;
43 end
44
45 // Selected Operations
46 always @(posedge clk) begin
47     xor_result <= Q_peres;
48     xnor_result <= ~Q_peres;
49     not_result1 <= ~A;
50     and_result <= R_peres;
51     nand_result <= ~R_peres;
52     a_b_one_result <= Q_fredkin2;
53     or_result <= Q_fredkin1;
54     nor_result <= ~Q_fredkin1;
55 end
56
57 endmodule
58
```



The screenshot shows a Verilog code editor window with the following code:

```
1 module Peres (
2     input clk,
3     input [31:0] A,
4     input [31:0] B,
5     input [31:0] C,
6     output reg [31:0] P,
7     output reg [31:0] Q,
8     output reg [31:0] R
9 );
10
11 always @ (posedge clk) begin
12     P <= A;
13     Q <= A & B;
14     R <= (A & B) & C;
15 end
16
17 endmodule
18
```

```
1 module Toffli (
2     input clk,
3     input [31:0] A,
4     input [31:0] B,
5     input [31:0] C,
6     output reg [31:0] P,
7     output reg [31:0] Q,
8     output reg [31:0] R
9 );
10
11    always @ (posedge clk) begin
12        P <= A;
13        Q <= B;
14        R <= (A & B) ^ C;
15    end
16
17 endmodule
18
```

```
1 module modified_Fredkin (
2     input clk,
3     input [31:0] A,
4     input [31:0] B,
5     output reg [31:0] P,
6     output reg [31:0] Q,
7     output reg [31:0] R
8 );
9 parameter C=1'b1;
10
11 always @ (posedge clk) begin
12     P <= A;
13     Q <= (~A & B) + (A & {32{C}});
14     R <= (A & B) + (~A & {32{C}});
15 end
16 endmodule
```

```
1 module modified_Pers (
2     input clk,
3     input [31:0] A,
4     input [31:0] B,
5     output reg [31:0] P,
6     output reg [31:0] Q,
7     output reg [31:0] R
8 );
9
10 parameter C=1'b0;
11
12 always @ (posedge clk) begin
13     P <= A;
14     Q <= A & B;
15     R <= (A & B) & {32{C}};
16 end
17
18 endmodule
19
```

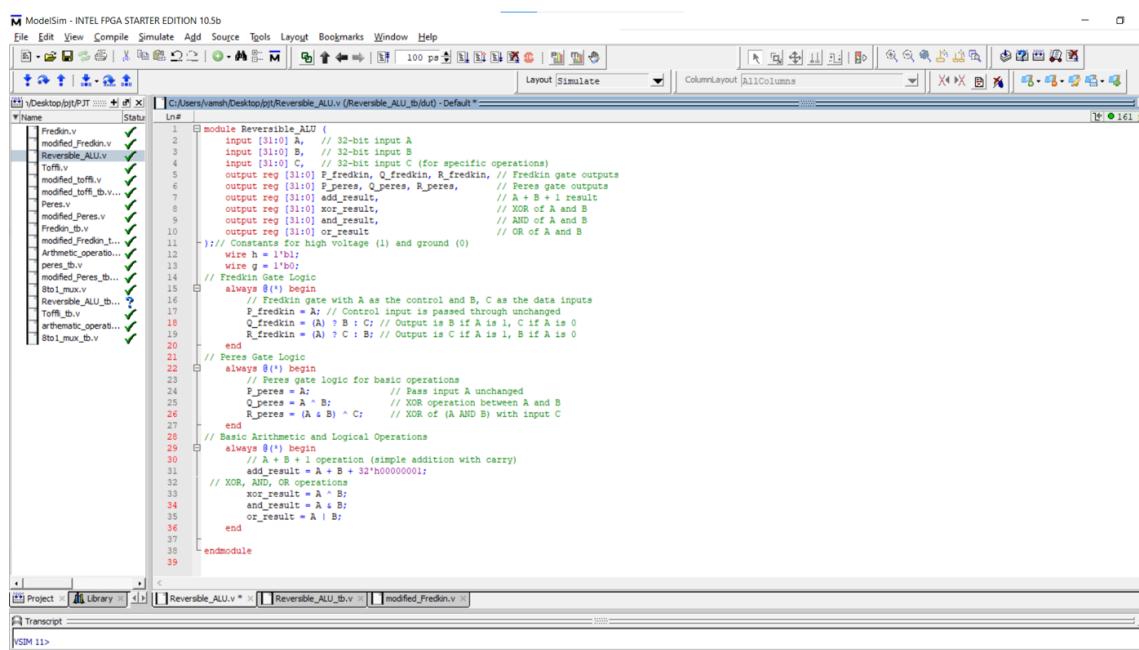
```
1 module modified_toffli (
2     input clk,
3     input [31:0] A,
4     input [31:0] B,
5     output reg [31:0] P,
6     output reg [31:0] Q,
7     output reg [31:0] R
8 );
9 parameter C=1'b0;
10
11 always @ (posedge clk) begin
12     P <= A;
13     Q <= B;
14     R <= (A & B)^{32{C}};
15 end
16 endmodule
17
```

```
1  module mux_8to1 (
2    input clk,
3    input [7:0] data_in,
4    input [2:0] sel,
5    output reg data_out
6  );
7
8  always @(posedge clk) begin
9    case (sel)
10      3'b000: data_out <= data_in[0];
11      3'b001: data_out <= data_in[1];
12      3'b010: data_out <= data_in[2];
13      3'b011: data_out <= data_in[3];
14      3'b100: data_out <= data_in[4];
15      3'b101: data_out <= data_in[5];
16      3'b110: data_out <= data_in[6];
17      3'b111: data_out <= data_in[7];
18      default: data_out <= 1'b0;
19    endcase
20  end
21
22 endmodule
23
```

```

1  module Arithmetic_operations(
2      input clk, // Clock signal
3      input [31:0] A,
4      input [31:0] B,
5      input [31:0] C,
6      output reg [31:0] xor_result,
7      output reg [31:0] xnor_result,
8      output reg [31:0] not_result1,
9      output reg [31:0] and_result,
10     output reg [31:0] nand_result,
11     output reg [31:0] a_b_one_result,
12     output reg [31:0] or_result,
13     output reg [31:0] nor_result
14 );
15
16 wire h = 1'b1;
17 wire g = 1'b0;
18
19 // Internal signals (not assigned to outputs, just for computation)
20 reg [31:0] P_fredkin1, Q_fredkin1, R_fredkin1;
21 reg [31:0] P_fredkin2, Q_fredkin2, R_fredkin2;
22 reg [31:0] P_peres, Q_peres, R_peres;
23
24 // Fredkin Gate 1
25 always @(posedge clk) begin
26     P_fredkin1 <= A;
27     Q_fredkin1 <= (A & B) | (~A & h);
28     R_fredkin1 <= (A & h) | (~A & B);
29 end
30
31 // Fredkin Gate 2
32 always @(posedge clk) begin
33     P_fredkin2 <= h;
34     Q_fredkin2 <= (h & Q_fredkin1) | (~h & h);
35     R_fredkin2 <= (h & h) | (~h & Q_fredkin1);
36 end
37

```



```

ModelSim - INTEL FPGA STARTER EDITION 10.5b
File Edit View Compile Simulate Add Source Tools Layout Bookmarks Window Help
Layout | Simulate | ColumnLayout | AllColumns | X4 X2 X1 X0 X1 X2 X4 X8 X16 X32 X64 X128 X256 X512 X1024 X2048 X4096 X8192 X16384 X32768 X65536 X131072 X262144 X524288 X1048576 X2097152 X4194304 X8388608 X16777216 X33554432 X67108864 X134217728 X268435456 X536870912 X107374184 X214748368 X429496736 X858993472 X1717986944 X3435973888 X6871947776 X13743895520 X27487791040 X54975582080 X109951164160 X219902328320 X439804656640 X879609313280 X1759218626560 X3518437253120 X7036874506240 X14073749012480 X28147498024960 X56294996049920 X112589992099840 X225179984199680 X450359968399360 X900719936798720 X1801439873597440 X3602879747194880 X7205759494389760 X14411518988779520 X28823037977559040 X57646075955118080 X115292151910236160 X230584303820472320 X461168607640944640 X922337215281889280 X1844674430563778560 X3689348861127557120 X7378697722255114240 X14757395444510228480 X29514790889020456960 X59029581778040913920 X118059163556081827840 X236118327112163655680 X472236654224327311360 X944473308448654622720 X1888946616897309245440 X3777893233794618490880 X7555786467589236981760 X15111572935178473963520 X30223145870356947927040 X60446291740713895854080 X120892583481427791708160 X241785166962855583416320 X483570333925711166832640 X967140667851422333665280 X1934281335702844667330560 X3868562671405689334661120 X7737125342811378669322240 X15474250685622757338644480 X30948501371245514677288960 X61897002742491029354577920 X123794005484982058709155840 X247588010969964117418311680 X495176021939928234836623360 X990352043879856469673246720 X1980704087759712939346493440 X3961408175519425878692986880 X7922816351038851757385973760 X15845632702077703514771947520 X31691265404154407029543895040 X63382530808308814059087790080 X126765061616617628118175580160 X253530123233235256236351160320 X507060246466470512472702320640 X1014120492932941024945404641280 X2028240985865882049890809282560 X4056481971731764099781618565120 X8112963943463528199563237130240 X16225927886927556399126474260480 X32451855773855112798252948520960 X64903711547710225596505897041920 X129807423095420451193011794083840 X259614846190840902386023588167680 X519229692381681804772047176353440 X103845938476336360954409435270720 X207691876952672721908818870541440 X415383753905345443817637741082880 X830767507810688887635275482165760 X166153501562137777527055096433120 X332307003124275555054110192866240 X66461400624855110010822038572480 X13292280124971022002044077154960 X26584560249942044004088154309320 X53169120499840088008166308618640 X106338240998801760016326617317280 X212676481997603520032653234634560 X425352963995207040065266668673120 X850705927990414080130533337346240 X170141185598082816026066667489280 X340282371196165632052133334978560 X680564742392321264104266669957120 X1361129484784624528208533339914240 X2722258969569249056416533379828480 X54445179391384981128320666596568960 X108890357826769762256641331193137760 X217780715653539524513282662286275520 X43556143130707904902656533457355040 X8711228626141580980531306691470080 X1742245725228316196106261338294160 X3484491450456632392212522676588320 X6968982900913264784425045353176640 X13937965801826525568850090706353280 X27875931603653051137700091412706560 X55751863207306102275400092825413120 X11150372641461204454800095650826240 X22300745282924408909600091301652480 X44601490565848817819200090603304960 X89202981131697635638400090306609920 X178405962263395271276800090153319840 X356811924526785272553600089976639680 X713623849053570545107200089953379360 X1427247698107141090214400089926758720 X2854495396214282180428800089913517440 X5708990792428564360857600089906754880 X11417981584571328717153600089903509760 X22835963168542657434307200089901754880 X45671926337085314868554400089900875440 X91343852674170629737108800089900437760 X182687705348341259474017600089900215520 X365375410696682518948035200089900107760 X730750821393365037896070400089900053840 X1461501642786730077892140800089900026960 X2923003285573460155784281600089900013480 X5846006571146920311568573232000899000067440 X1169201314229840622313706464000899000033760 X2338402628459681244627412928000899000016880 X4676805256919362489254825856000899000008440 X93536105138387249785096517120008990000042200 X1870722102767744955704930342400089900000211000 X37414442055354899114098606848000899000001055000 X748288841107097982281972136960008990000005275000 X1496577682214195964543944273920008990000002638000 X2993155364428391929087888547840008990000001319000 X59863107288567838581757770956800089900000006595000 X1197262145713356771635155419136000899000000032975000 X2394524291426713543270307838272000899000000016488000 X4789048582853427086540615665544000899000000008244000 X9578097165706854173081231331088000899000000004122000 X1915619433141370834616262665544000899000000002061000 X3831238866282741669232525331088000899000000001030500 X76624777325654833384650506655440008990000000005152500 X1532495546513096667693010133108800089900000000025762500 X30649910930261933353860202665544000899000000000128812500 X612998218605238667077204053310880008990000000000644062500 X1225996437210477334154080266554400089900000000003220312500 X24519928744209546682801605331088000899000000000016101562500 X490398574884190933656032026655440008990000000000080507812500 X9807971497683818673120640533108800089900000000000402539062500 X196159429953676373462412810665544000899000000000002012790312500 X3923188599073527469248256213310880008990000000000010063951562500 X7846377198147054938496512466554400089900000000000050319757812500 X15692754396294109877930249331088000899000000000000251598789062500 X31385508792588219755860496665544000899000000000000125799394512500 X62771017585176439511720893310880008990000000000000628996972562500 X1255420351703528790234417866554400089900000000000003144984862812500 X25108407034070575804688357331088000899000000000000015724924314062500 X502168140681411516093767146655440008990000000000000078624621570312500 X10043362813628223208755342933108800089900000000000000393123107851562500 X200867256272564464175106858665544000899000000000000001965615539257812500 X4017345125451289283521337173310880008990000000000000009828077696289062500 X8034690250902578567042674346655440008990000000000000004914038848154512500 X160693805018051571340853486933108800089900000000000000024570194240772562500 X3213876100361031426817069738665544000899000000000000000122850971203862500 X64277522007220628536341394773310880008990000000000000000614254856019312500 X128555044014401357072678895466554400089900000000000000003071274280096562500 X2571100880288027141453577809331088000899000000000000000015356371400482812500 X51422017605760542828715556186655440008990000000000000000076781857002414062500 X102844035211521085654311113433108800089900000000000000000383909285012070312500 X205688070423042171308622226866554400089900000000000000000191954642506035062500 X411376140846084342617244453733108800089900000000000000000095977321253017512500 X8227522816921686852344889074665544000899000000000000000000479886606265087562500 X16455045633843373704697778093310880008990000000000000000002399433031325437812500 X329100912676867474093955561866554400089900000000000000000011977165156627189062500 X658201825353734948187911123433108800089900000000000000000005988582578313594512500 X13164036507074698163782222468665544000899000000000000000000029942912891567972562500 X26328073014149396327564444937310880008990000000000000000000149714564457839862500 X526561460282987926551288898746655440008990000000000000000000748572822289199312500 X1053122920565975853035777974933108800089900000000000000000003742864111445976562500 X21062458411319517060715559498665544000899000000000000000000018714320557229882562500 X42124916822639034121431118973310880008990000000000000000000093571602786149412500 X842498336452780682428622379466554400089900000000000000000000467858013930747062500 X1684996672905561364573446789331088000899000000000000000000002339290069653735312500 X33699933458111227291468935786655440008990000000000000000000011696450348268677562500 X67399866916222454582937871573310880008990000000000000000000058482251741343387562500 X134799733832448591655855531146655440008990000000000000000000029241125870671693812500 X26959946766489718331171106293310880008990000000000000000000014620562935335346562500 X5391989353297943666234221258665544000899000000000000000000000731028147766771312500 X10783978706598867332464425173310880008990000000000000000000003655140748833856562500 X215679574131977346649288503466554400089900000000000000000000018275703744169282562500 X43135914826395469329777706893310880008990000000000000000000009137851872084744512500 X862718296527909386595554037866554400089900000000000000000000045689259360423722562500 X17254365930558187731911080757331088000899000000000000000000000228446296802118612500 X345087318611163754638221615146655440008990000000000000000000001142231484010593062500 X6901746372223275092764432302933108800089900000000000000000000005711157420052965312500 X13803492744446501855328666058665544000899000000000000000000000028555787100264826562500 X2760698548889300371065733211733108800089900000000000000000000001427789355013241312500 X55213970977786007421314666429331088000899000000000000000000000007138946775066221562500 X110427941955572014842629332586655440008990000000000000000000000035694733875331112500 X2208558839111440296852586655440008990000000000000000000017847366937665562500 X4417117678222880593705173310880008990000000000000000000000089236834688330312500 X8834235356445760187410346655440008990000000000000000000000044618417343665262500 X1766847071289152037482069331088000899000000000000000000000002230920867183301562500 X35336941425783040749641386655440008990000000000000000000000011154604359066507562500 X70673882851566081499282773310880008990000000000000000000000005577302179533251562500 X14134776570313216299565546655440008990000000000000000000000002788651089766625312500 X28269553140626432599131093310880008990000000000000000000000001394325544883312562500 X5653910628125286519826218665544000899000000000000000000000000069716277244165627562500 X1130782125625141303652437331088000899000000000000000000000000034858115494062813812500 X2261564251252806607304874665544000899000000000000000000000000017429057747031407562500 X4523128502505613214609749331088000899000000000000000000000000008714528874401571562500 X9046257005011226429219498665544000899000000000000000000000000004357264437303135312500 X180925140100224528543899773310880008990000000000000000000000000021786322186506676562500 X36185028020044905708779954665544000899000000000000000000000000001089316109325333812500 X72370056040089811417559909331088000899000000000000000000000000005446580546627669312500 X144740112080179622835198186655440008990000000000000000000000000027232902733138346562500 X28948022416035924567039637331088000899000000000000000000000000001361645136656917312500 X5789604483207184913407927533108800089900000000000000000000000000068082256833134866562500 X115792089664043698268158550665544000899000000000000000000000000003404112841656743312500 X2315841793280873965363171013310880008990000000000000000000000000017020564208283716562500 X463168358656174793072634202665544000899000000000000000000000000000851028210414185812500 X9263367173123495861452684053310880008990000000000000000000000000004255141052070929062500 X1852673434
```

Bibliography

- [1] S. M. Swamynathan and V. Banumathi, *Design and analysis of FPGA based 32 bit ALU using reversible gates*, 2017 IEEE International Conference on Electrical, Instrumentation and Communication Engineering (ICEICE), Karur, India, 2017, pp. 1-4, doi: 10.1109/ICEICE.2017.8191959.
- [2] M. Singh, M. Kumar and V. Nath, *Comparative Analysis of Reversible Arithmetic Logic Units (R-ALU) Using Verilog*, 2024 First International Conference on Electronics, Communication and Signal Processing (ICECSP), New Delhi, India, 2024, pp. 1-8, doi: 10.1109/ICECSP61809.2024.10698408.
- [3] M. Morrison and N. Ranganathan, *Design of a Reversible ALU Based on Novel Programmable Reversible Logic Gate Structures*, 2011 IEEE Computer Society Annual Symposium on VLSI, Chennai, India, 2011, pp. 126-131, doi: 10.1109/ISVLSI.2011.30.
- [4] B. Jana, A. Jana, S. Basak, J. K. Sing and S. K. Sarkar, *Design and performance analysis of reversible logic based ALU using hybrid single electron transistor*, 2014 Recent Advances in Engineering and Computational Sciences (RAECS), Chandigarh, India, 2014, pp. 1-4, doi: 10.1109/RAECS.2014.6799652.
- [5] S. Sen, P. Saha and S. Saha, *FPGA-Supported HDL Approach to Implement Reversible Logic Gate-Based ALU*, 2023 11th International Conference on Internet of Everything, Microwave Engineering, Communication and Networks (IEMECON), Jaipur, India, 2023, pp. 1-5, doi: 10.1109/IEMECON56962.2023.10092307.
- [6] S. Thakral and D. Bansal, *Comparative study and implementation of BCD adders for reversible logic based ALU*, 2017 2nd International Conference on Telecommunication and Networks (TEL-NET), Noida, India, 2017, pp. 1-5, doi: 10.1109/TEL-NET.2017.8343544.
- [7] A. Kamaraj and P. Marichamy, *Design and implementation of arithmetic and logic unit (ALU) using novel reversible gates in quantum cellular automata*, 2017 4th

- International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 2017, pp. 1-8, doi: 10.1109/ICACCS.2017.8014578.
- [8] J. F. Chaves, D. S. Silva, V. V. Camargos and O. P. Vilela Neto, *Towards reversible QCA computers: Reversible gates and ALU*, 2015 IEEE 6th Latin American Symposium on Circuits & Systems (LASCAS), Montevideo, Uruguay, 2015, pp. 1-4, doi: 10.1109/LASCAS.2015.7250458.
- [9] H. Thapliyal, S. Kotiyal and M. B. Srinivas, *Novel BCD adders and their reversible logic implementation for IEEE 754r format*, 19th International Conference on VLSI Design held jointly with 5th International Conference on Embedded Systems Design (VLSID'06), Hyderabad, India, 2006, pp. 6 pp.-, doi: 10.1109/VLSID.2006.122.
- [10] R. K. James, K. P. Jacob and S. Sasi, *Design of compact reversible decimal adder using RPS gates*, 2012 World Congress on Information and Communication Technologies, Trivandrum, India, 2012, pp. 344-349, doi: 10.1109/WICT.2012.6409100.
- [11] A. Banerjee, *Reversible cryptographic hardware with optimized quantum cost and delay*, 2010 Annual IEEE India Conference (INDICON), Kolkata, India, 2010, pp. 1-4, doi: 10.1109/INDCON.2010.5712605.
- [12] M. Morrison, M. Lewandowski, R. Meana and N. Ranganathan, *Design of a novel reversible ALU using an enhanced carry look-ahead adder*, 2011 11th IEEE International Conference on Nanotechnology, Portland, OR, USA, 2011, pp. 1436-1440, doi: 10.1109/NANO.2011.6144406.

Biodata

Overleaf is a great professional tool to edit online, share and backup your L^AT_EX projects. Also offers a rather large base of help documentation.



Name:

Mobile No.:

E-mail:

Permanent Address:

Overleaf is a great professional tool to edit online, share and backup your L^AT_EX projects. Also offers a rather large base of help documentation.



Name:

Mobile No.:

E-mail:

Permanent Address: