Design and Implementation of 4-bit ALU using Reversible Logic Gates

Tanya[1], Ashwini Gowda H.B.[2], Utsav Anand[3], and Prateek Kumar[4]
[1]*Tanya, Dayananda Sagar College of Engg,* [2] *Ashwini Gowda H.B., Dayananda Sagar College of Engg*
[3]*Utsav Anand, Dayananda Sagar College of Engg,* [4] *Prateek Kumar, Dayananda Sagar College of Engg*

Abstract: ***Modern computing architectures are evolving toward systems where computational reversibility plays a fundamental role. A key innovation in this domain is the development of a novel arithmetic logic unit (ALU) that maintains complete bidirectional operation capabilities. This advanced ALU architecture employs sophisticated multiplexer configurations and precise control signaling to achieve reversible computation. As a critical component within the central processing unit, this reversible ALU design represents a significant step toward programmable quantum computing systems. The architecture leverages multiplexer-based operation selection, enabling flexible computational pathways while maintaining information preservation. Through implementation of programmable reversible logic gates, this design transcends traditional AND/OR gate limitations. The proposed 4-bit ALU configuration demonstrates enhanced efficiency by utilizing inverted data references, significantly reducing power consumption in logic circuits. The implementation, validated through comprehensive simulation using industry-standard tools including Verilog HDL, ModelSim Altera, and Quartus Prime, confirms the design's viability for next-generation computing applications. This innovative approach represents a crucial advancement in developing energy-efficient, quantum-compatible processing units.***

Keywords - Reversible logic gate, reversible ALU, Quantum Computing

I. INTRODUCTION

Emerging breakthroughs in computational engineering have unveiled innovative approaches to quantum computing architectures through advanced reverse engineering techniques. Reversible logic represents a cutting-edge structural design in nanotechnology that minimizes physical entropy while unprecedented computational efficiency. The intricate information derived from these studies provides critical insights into developing sophisticated reverse logic gate patterns and codes. Despite the potential, researchers have historically overlooked the comprehensive design of reversible arithmetic logic units (ALUs). Traditional binary logic circuits constructed with irreversible gates inherently generate significant power dissipation, creating substantial challenges in heat management that currently limit the scalability of complementary metal-oxide-semiconductor (CMOS) technologies.

Leveraging principles from Landauer's theorem, computational power consumption is fundamentally linked to logical operations when component specifications remain undefined. Conventional logic gates, typically featuring dual inputs and singular outputs, inevitably lose computational information during signal transmission. Each data bit's energy dissipation can be quantified as kT ln2 joules, where k represents the Boltzmann constant and T signifies absolute temperature. This inherent energy loss in digital circuitry generates unavoidable thermal byproducts. Theoretical frameworks suggest that implementing circuits with inverted gates could potentially mitigate energy consumption, offering a promising alternative that operates with minimal signal power. In very-large-scale integration (VLSI) circuitry, this approach typically involves intricate configurations of logical gates that can preserve and recycle computational states, presenting an attractive solution for emerging technological domains including nanotechnology, quantum computing, and optical computational systems.

II. THE REVERSIBLE LOGIC GATE

Computational reversibility represents a groundbreaking paradigm in circuit design, transcending traditional computational boundaries by establishing a mathematically precise information preservation mechanism. Unlike conventional circuit architectures, reversible systems introduce a transformative approach where every computational state maintains an intrinsic ability to retrace its originating configuration. This bidirectional information mapping creates a cryptographic-like

precision in circuit behavior, where input and output vectors share an unambiguous, deterministic relationship. If there is a one-to-one communication, the gate logic is invertible. The expression is as follows:

Iv = (I1, I2,….,In)

Ov=(O1,O2…..On)

where Iv is the input vector, Ov is the output vector.

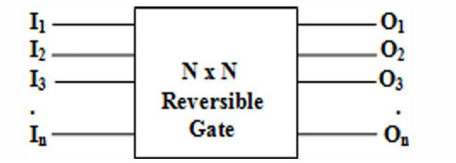

Fig. 2.1: N x N Reversible gate

Here the input I (A, B, C) and the output O (P, Q, R). The output is defined by P=A, Q=A'B^C and R=A′C^AB. Quantum cost of a Fredkin gate is 5.

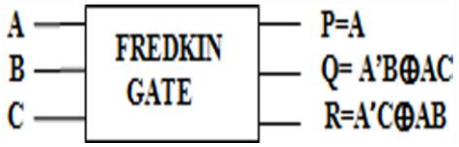

Figure 2.2 it shows a 3×3 Fredkin gate

Commonly used reversible gates are NOT gate, CNOT gate, Toffoli gate (CCNOT gate), Fredkin gate, Peres gate Feynman gate, etc. Following Fig 2.3 shows XOR implementation using irreversible logic gate, here there is no unique mapping as input has 2 vectors whereas the output has only 1 vector, this unequal number of inputs and output leads to loss of information.

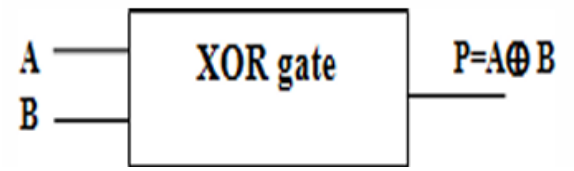

Figure 2.3 XOR Gate

In the Reversible XOR gate there is lossless information signals. As it maps the input with output which gives the number of equal inputs and equal output .This is shown in Fig2.4.

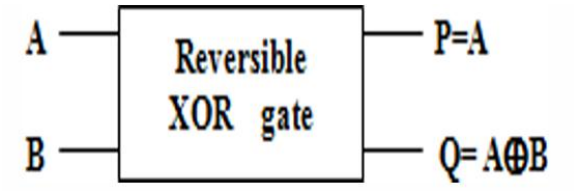

Figure 2.4 Reversible XOR (CNOT) gate

Peres gate 3x3 is shown below in Figure 2.5. In the design, Peres gate has lowest quantum cost that's why we have used Peres gate. The input vector includes I (A, B, C) and the output vector includes O(P,Q,R) The output is defined by P= A, Q = A^B and R =(A.B) ^ C. Quantum cost of a Peres gate is 4.

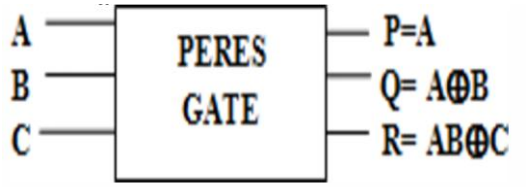

Figure 2.5 3x3 Peres gate.

The input vector of Feynman gate is I (A, B) and the output vector is O (P, Q) as we have considered a 2x2. The outputs are defined by P=A, Q=A ^B and it is shown in Figure 2.6. Quantum cost of a Feynman gate is 1.

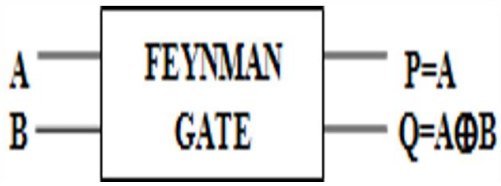

Figure 2.6 2x2 Feynman gate.

Following Figure 2.7 shows 3x3 Toffoli Gate also known as CCNOT gate. The input vector is represented by I(A, B, C) and the output vector is represented by O(P,Q,R). The outputs are defined by P=A, Q=B, R=AB^C.

Quantum cost of a Toffoli gate is 5.

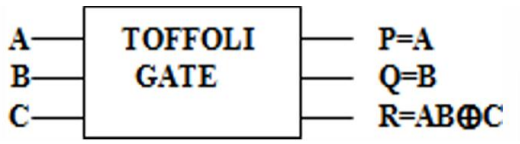

Figure 2.7 3×3 Toffoli (CCNOT) gate.

III. DESIGN OF ARITHMETIC LOGIC UNIT

Arithmetic and logic unit (ALU) is a data processing unit, which is a part of CPU. Among various types of CPUs which are available every CPU contains an ALU.

3.1 The ALU Based on Reversible logic:

The multi-function ALU based on reversible logic gates mainly contains the two 4 bit operands A and B on which Addition, Subtraction, Multiplication and Division operations are performed as a part of Arithmetic operations, for Logical operations AND, OR, XOR and XNOR operations are performed.

Along with this, Cyclic shift modules are created with the functionality of both left and right cyclic shift operation.

3.2 The Realization of Reversible ALU:

Figure-3.1 RTL design showcases a complex 4-bit arithmetic logic unit (ALU) implemented within the FPGA development environment. The model introduces two primary functions: 4x4 integer multiplication and 4-bit division, both leveraging reversible logic to enhance power efficiency and maintain data integrity throughout computations. The system operates on two 4-bit data bus inputs (a[3..0] and b[3..0]) and incorporates a mode selector for controlling functions. The combiner processes the 4-bit inputs to generate an 8-bit output (product [7...0]) for multiplication, while the divider calculates the division result and residual values. A multiplexer governs the final output, dynamically switching between multiplication and division based on the input type. The reversibility of the design makes it well-suited for applications requiring low-power systems, quantum computing prototypes, and educational platforms focused on understanding reversible logic.

The 4-bit implementation achieves a balance between simplicity and performance, providing efficient arithmetic operations while maintaining computational integrity.

Figure 3.1: RTL view of MUL and DIV. Figure 3.2: The RTL design represents a 4-bit reversible logic operations unit implemented within an FPGA environment. This architecture integrates various reversible logic gates and comparators to execute different logical operations. It takes three main inputs: choice[1:0] for operation selection and two 4-bit inputs, A[3:0] and B[3:0].

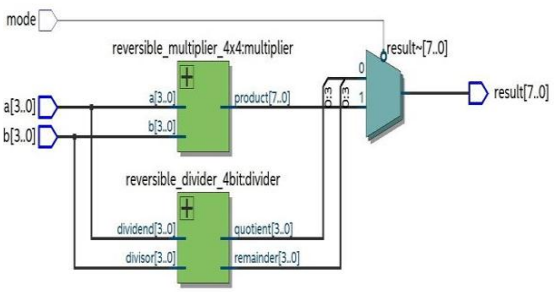

Figure 3.1 RTL view of MUL and DIV

The circuit includes several key components:

Two reversible 4-bit XNOR/XOR gates (indicated in green). A reversible 4-bit OR Toffoli gate. A reversible 4-bit AND Toffoli gate. Three equality comparators (Equal0, Equal1, and Equal2), which regulate the operation of multiplexers. The data flow is managed through multiple multiplexers that handle different bit ranges (Y~[3:0], Y~[7:4], and Y~[11:8]), ultimately combining to generate the 4-bit output Y[3:0]. This reversible design ensures information conservation throughout the computation, enhancing energy efficiency and making it suitable for quantum computing applications.

The hierarchical architecture emphasizes thoughtful signal routing and operation sequencing. The green blocks signify modular and reusable components, highlighting the design's adaptability. The systematic arrangement of the logic gates and comparators enables advanced logical operations while adhering to the principles of reversibility. This makes it an ideal choice for low-power and quantum computing environments.

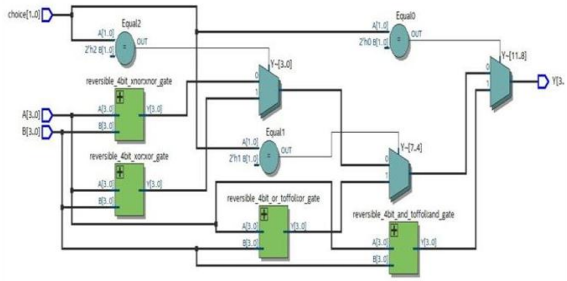

Figure 3.2 RTL view of Logical operation module

Figure 3.3 RTL viewer showing a digital circuit diagram of a "ReversibleShift" component. The design implements a reversible shift register that can shift d ata in either direction based on a shift direction control input. The circuit takes a 4-bit input (in[3..0]) along with a 2-bit control input (r[1..0]) and produces a 4-bit output (out[3..0]). The internal architecture consists of multiple "Less Than" comparator blocks for comparison operations, several multiplexers labeled as "temp" with different bit ranges for data routing, and logic gates marked as "always0=0", "always0=1", and "always0=2" for control logic. The overall structure is organized to enable bidirectional shifting of data through the register, with the various comparison blocks and multiplexers working together to control the data flow path and perform the shifting operation according to the control inputs.

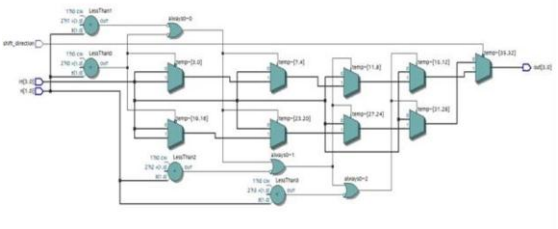

Figure 3.3 RTL view of Cyclic Shift Register

Figure 3.4 RTL design illustrates a dual-function 4-bit arithmetic unit that can perform both subtraction and addition operations using reversible logic principles. The architecture consists of two main components: a Reversible4BitSubtractor and a Reversible4BitAdder, both shown as green blocks in the diagram. The circuit accepts three primary inputs: a mode selector for choosing between addition and subtraction operations, and two 4-bit input buses A[3..0] and B[3..0]. The subtractor module generates a difference output (Diff[4..0]), while the adder module produces both a sum (Sum[3..0]) and a carry-out signal. A multiplexer (shown in turquoise) selects between the subtraction and addition results based on the mode input to produce the final Result[4..0]. The design's reversible nature ensures minimal energy dissipation and maintains information preservation throughout the computation process.

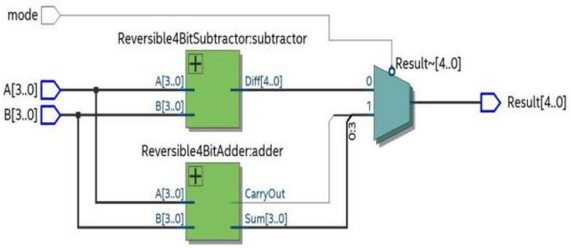

Figure 3.4 RTL view of Adder and Subtractor

The hierarchical structure demonstrates careful consideration of signal routing and operation sequencing, with the green blocks indicating modular, reusable components. The design's systematic arrangement of logic gates and comparators enables complex logical operations while maintaining reversibility principles, essential for low- power and quantum computing applications.

3.3 SIMULATION RESULTS:

The proposed reversible ALU architecture is implemented using structural modeling in Verilog HDL. Toffoli gates are packaged as modular components, instantiated as discrete objects within the design.Simulation and verification are conducted using ModelSim-Altera 10.1d, representing input and output signals through carefully defined variables. The modular design enables flexible circuit architecture and comprehensive signal management.Comparative analysis between simulated program outputs and actual circuit responses demonstrates remarkable consistency. This correlation establishes a robust validation framework, providing confidence in the design's accuracy and reliability.

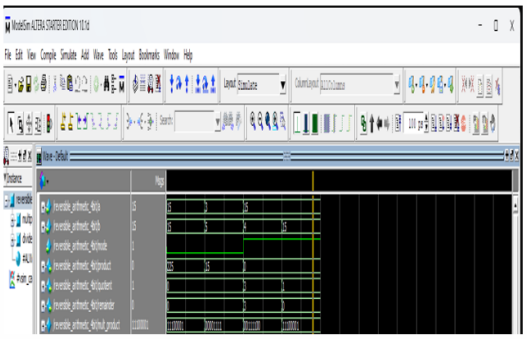

Figure 3.5 Simulation Result of Arithmetic Multiplication and Division

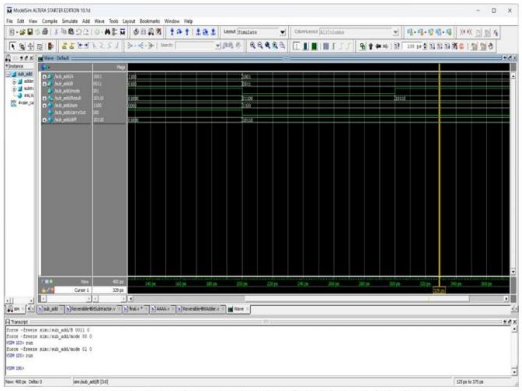

Figure 3.6 Simulation Result of Adder and Subtractor

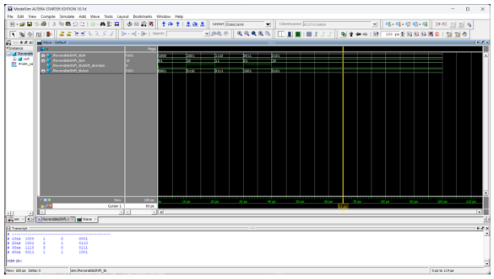

Figure 3.7 Simulation Result of Logical Shift Register

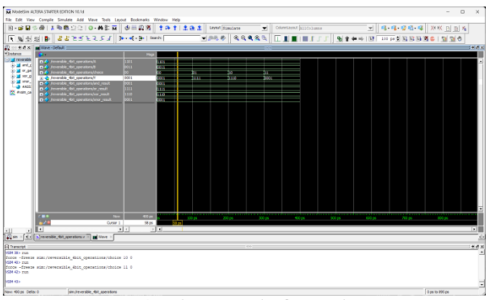

Figure 3.8 Simulation Result of Logical operations

IV. COMPILATION REPORTS

The Verilog code is compiled using intel Quartus Prime software for power dissipation analysis.

Below table depicts the power dissipation of 4-bit reversible ALUs. Also, as the number of bits is directly proportional to power dissipation.

	Reversible ALU (4 Bit)	Conventional ALU (4 Bit)
Power (mW)	98.47	206
Time Delay (ns)	1.071	2.041

Table 1. Analysis of Power dissipation 4 bit conventional and reversible ALU

	Utilized	Available	Utilization %
Logic Elements	2300	49,760	4.62 %
LUTs	2000	24,880	8.04 %
Flip-Flop	700	49,760	1.41 %
DSP Blocks	8	144	5.56 %

Table 2. Area Utilization on 4-bit reversible ALU

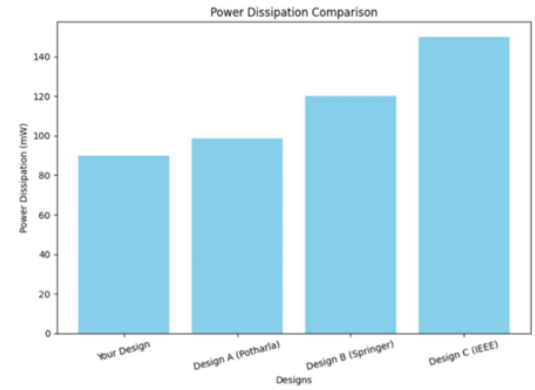

Figure 4.1 Power Dissipation

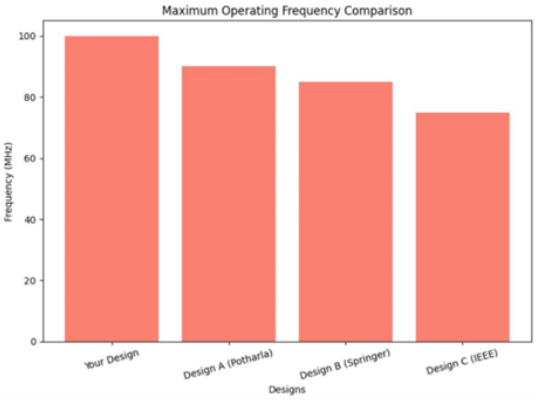

Figure 4.2 Maximum Operating Frequency

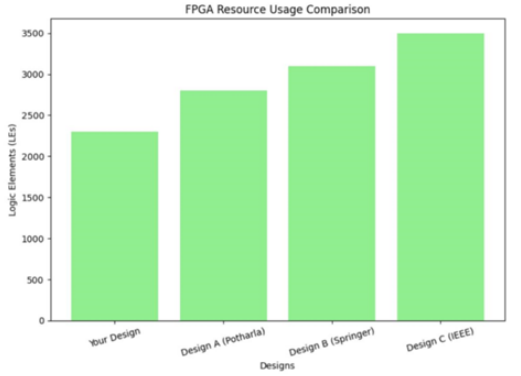

Figure 4.3 FPGA Resource Usage Comparison

The proposed design has lower power dissipation, higher operating frequency which shows most speed of the proposed design and lower resource usage which shows lowest area utilization among all the designs.

Design A -A Novel Design of a four-bit Reversible ALU using an emphasized CLAA.

Design B -Optimized Design of ALU Using Reversible Gate.

Design C -An Arithmtic Logic Unit design based on Reversible Logic Gates

V. CONCLUSION

This research introduces an innovative 4-bit arithmetic logic unit (ALU) design leveraging reversible logic gates as an alternative to conventional circuit architectures. By implementing quantum-inspired reversible logic principles, the proposed methodology demonstrates significant reduction in information bit loss and achieves optimal power consumption metrics.

The research systematically evaluates various computational modules using advanced electronic design automation tools, specifically Quartus Prime. Experimental findings reveal that circuits constructed through reversible logic gates exhibit substantially reduced power dissipation and computational delay compared to traditional design paradigms.

The theoretical framework emphasizes logical reversibility, highlighting the unique characteristic of input-output mappings where computational states can be precisely reconstructed. This approach ensures complete information preservation throughout the computational process, challenging conventional computational methodologies.

The proposed design not only demonstrates theoretical potential but also provides a pragmatic

pathway towards energy-efficient, information-conserving computational architectures.

VI. REFERENCES

[1] Khazamipour and K. Radecka, "Adiabatic Implementation of Reversible Gates", Proceedings of IEEE Midwest Symposium on Circuits and Systems (MSCAS), 2005, vol. 1, DOI: 10.1109/ MWSCAS.2005.1594096, pp.291-294.

[2] Matthew Morrison and Nagarajan Ranganathan, "Design of Reversible ALU based on Novel Programmable Reversible Logic Gate Structures", IEEE Computer Society Annual Symposium on VLSI, 2011, vol. 3, DOI: 10.1109/ISVLSI.2011 .30, pp.126-131.

[3] Khazamipour and Katarzyna Radecka "A New Architecture of Adiabatic Reversible Logic Gates", Proceedings of IEEE North East Workshop on Circuits and Systems (NEWCAS), 2006, DOI:10.1109/NEWCAS.2006.250900, pp.233-236.

[4] Zeljko, zilic, Radecka.K and Kzamipur.A "Reversible circuit technology mapping from non reversible specifications", Design, Automation and Test in Europe Conference and Exhibition, 2007, DOI: 10.1109/DATE.2007.364652, pp.1-6.

[5] R. Landauer, "Irreversibility and Heat Generation in the Computing Process," IBM Journal of Research and Development, 1961, vol: 5, DOI: 10.1147/rd.53.0183, pp.183-191.

[6] Frehse, S. Wille, and R. Drechsler, R. "Efficient Simulation- Based Debugging of Reversible Logic", IEEE International Symposium on Multiple Valued Logic (ISMVL), 2010, DOI: 10.1109/ISMVL.2010.37,pp.156-161.

[7] Thapliyal, and Vinod, A.P. "Design of Reversible Sequential Elements with Feasibility of Transistor Implementation", IEEE International Symposium on Circuits and Systems, 2007, DOI: 10.1109/ISCAS.2007.378815, pp.625-628.