iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** VI **Month of publication:** June 2025

DOI: https://doi.org/10.22214/ijraset.2025.72366

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue VI June 2025- Available at www.ijraset.com

Design and FPGA Implementation of a 4-Bit ALU Using Reversible Logic Gates

Yug Milind Thosar[1], Param Patel[1], Sunil Patel[2]
[1]*Bachelor of Engineering (Electronics) student,* [2]*Assistant professor, Electrical Engineering Department, Faculty of Technology and Engineering, The Maharaja Sayajirao University of Baroda*

Abstract: ***Reversible computing offers a promising approach to mitigate the power dissipation challenges inherent in conventional digital systems. In this work, we present the design and analysis of a novel 4-bit reversible Arithmetic Logic Unit (ALU) that leverages reversible logic gates—including Fredkin, Feynman, Universal Reversible Gate (URG), and Thapliyal-Srinivas Gate (TSG)—to perform a suite of fundamental arithmetic and logical operations. The ALU is architected to execute operations such as binary addition, subtraction, logical AND, OR, and NOT under a unified reversible computing framework, thereby ensuring a one-one mapping between inputs and outputs and minimizing energy loss due to information erasure. To validate the design, the complete system was modeled using MATLAB Simulink and further synthesized to VHDL code, with implementation on a BASYS-3 FPGA featuring a Xilinx ARTIX-7 device. Comprehensive simulation results demonstrate the circuit's functional correctness and favorable performance in terms of resource utilization and power consumption. The study also includes performance analysis highlighting the low usage of Look-Up Tables (LUTs) and flip-flops and identifies potential avenues for further optimization, including reductions in quantum cost and propagation delay. The proposed reversible ALU underscores the practicality of energy-efficient, reversible logic in real-world VLSI and quantum-aware architectures, thus opening new possibilities for sustainable computing technology.***
Keywords: Arithmetic Logic Unit, Feynman, Fredkin, Reversible Computing, Quantum Cost, VLSI

I. INTRODUCTION

In modern digital systems, Arithmetic Logic Units (ALUs) serve as a fundamental building block for performing arithmetic and logical operations [1]. Energy bottlenecks are now a matter of concern in traditional computing systems and hence reversible computing may be a potential remedy to overcome this bottleneck [1,2]. The concept of reversible computing ideally focuses on reducing power consumption by ensuring that no information is lost during the computation process, making it particularly suitable for energy-conscious applications [1].

This article explores the design and analysis of a Reversible 4-bit ALU, an essential component of computing systems . The 4-bit ALU is designed to perform basic arithmetic and logical operations, such as addition, subtraction, AND, OR and 1's complement. By leveraging reversible gates and logic circuits, the proposed ALU enhances computational efficiency while ensuring that no energy is wasted due to information erasure.

The primary objectives of this study include the development of the ALU's architecture using reversible logic gates, followed by a detailed analysis of its performance in terms of power consumption, speed, and functional correctness [8,9]. Through this analysis, we aim to demonstrate the potential advantages of reversible ALUs in both theoretical and practical applications, setting the stage for more energy-efficient computational systems in the future.

II. REVERSIBLE COMBINATIONAL CIRCUIT MODULES

The reversible logic gates form the fundamental element of every circuit designed for the purpose of reversibility [1,2]. There are several reversible logic gates whose explanation is beyond the scope of this paper. However the concept of reversibility can be briefly explained using the Fredkin gate as an example [1].

- Figure 1 shows a 3*3 Fredkin gate. The input vector is I (A, B, C) and the output vector is O (P, Q, R).
- The output is defined by (P=A, Q=A′B ⊕ AC and R=A′C ⊕ AB).
- Quantum cost of a Fredkin gate is 5.

©IJRASET: All Rights are Reserved

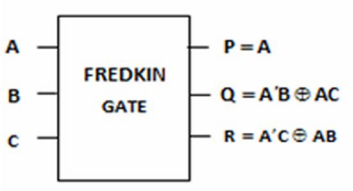


Figure 1. Fredkin gate [5]

A	B	C	P	Q	R
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	0	0
1	0	1	1	1	0
1	1	0	1	0	1
1	1	1	1	1	1

Table 1. Truth-table of Fredkin Gate [5]

From the two highlighted terms of Table 1, it can be observed that for initial inputs (A,B,C) as '101' the output received is (P,Q,R) as '110' [1]. As this output is applied as input in (A,B,C) as '110' the previous input is obtained at (P,Q,R) as '101' [1]. This illustration proves the reversibility of Fredkin Gate and allows us to use in other circuits to meet the requirements.

II.A. Multiplexers:

Fredkin Gate can be used for designing of multiplexer:

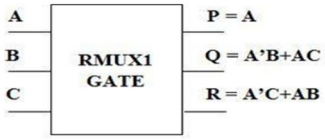


Figure 2. 2x1 MUX by Fredkin Gate [9]

Here, in Figure 2, A is a select line whereas B and C are inputs of the multiplexer. When A=0 then B is forwarded at output Q and vice versa C is transferred at Q when A=1 [9]. This characteristic of Fredkin Gate allows us to implement 2x1 multiplexer [9]. This multiplexer is integral component of proposed ALU design as it enables user to select respective operation as per choice [8].

A. Adders/ Subtractors

Adders/ Subtractor form an essential part of ALU [8]. Besides performing addition or subtraction operation these circuits are also capable to perform certain logical operations on the event of certain modifications in their input signals [5,8]. Hence an ALU can be constructed using a full adder just by manipulating its input signals for the desired output response [8].

B. Half Adder/Subtractor

Below Figure 3 is the tested and verified design of a half adder/subtractor and gates used are Feynman Gate (Quantum cost of 1) and a Peres Gate (Quantum cost of 4) [5]. There are two garbage outputs as g1 and g2. S/D signal stands for the sum or difference output, whereas C/B stands for carry/Borrow [5]. The bit assigned on the Ctrl signal denotes which operation will be performed.

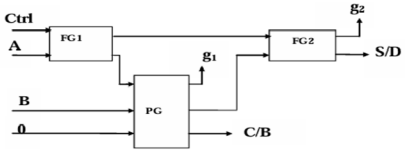


Figure 3. Half Adder/Subtractor [5]

C. Full Adder/ Subtractor

Following Figure 4 is the tested and verified design for full adder/subtractor constructed using two Feynman gates and two Peres gates [5]. Overall Quantum cost stands at (2*1) + (2*4) = 10. The number of garbage outputs are 3 [5]. For Ctrl value zero the circuit performs addition and Subtraction for Ctrl value one [5].

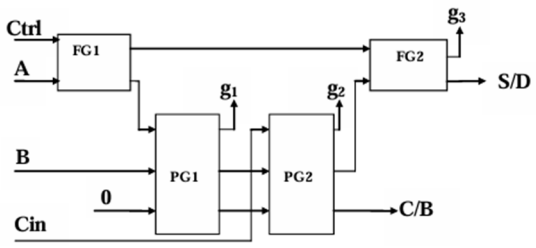


Figure 4. Full Adder/Subtractor [5]

III. REVERSIBLE SEQUENTIAL CIRCUIT MODULES

Sequential components have great importance in meeting timing requirements of computing systems [2]. These components can act as buffers, storage registers, counters etc. for transferring necessary information during computation [4]. SR and D Latches, D- and T- Flip-flops and reversible counter are realized using reversible logic gates [4,7]. For any sequential circuit to function properly, internal delay has to meet timing requirements or else corrupted output is obtained [2]. Firstly, we realized the working of SR Latch through two different designs [4]. Both the designs consisted of internal feedback paths and therefore maintaining time delay for these paths was crucial [2]. The respective values of time delay varies as per clock provided to circuit.:

A. SR Latch

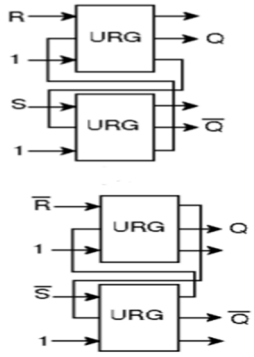


Figure 5. SR Latch [2]

The First design in Figure 5 is of active high SR Latch where Q is high when S is high and R is low [2]. URG gates are used for carrying out this functionality. Similarly, Second design is of active low SR latch where Q is high when S is low and R is high [2].

B. D Latch

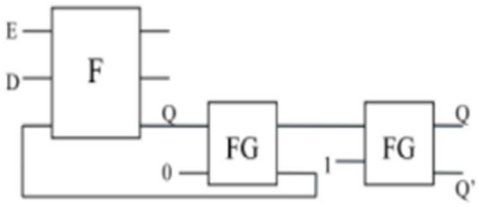


Figure 6. D Latch [2]

The given Figure 6 shows a D Latch where E (enable) signal activates circuit and output Q follows D input. Fredkin and Feynman gates are used to perform this D Latch functionality [4]. This design exhibits internal feedbacks which need to be handled for smooth functioning of D Latch operation as per enable signal [2,4].

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue VI June 2025- Available at www.ijraset.com

C. Flip Flop

T flip flops have great usage in computing systems to divide frequency, govern controlling mechanisms and more [7]. To ensure timing requirements of the CPU, a proposed T Flip-flop made from reversible logic gates is shown in the figure below. Two Sayen gates and one Feynman gate are required to implement reversible T Flip-flop [1].

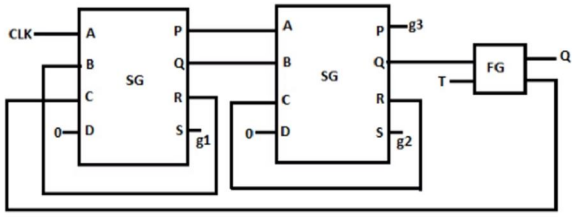


Figure 7. T Flip-flop [1]

$$\text{Quantum cost} : 2 * 9 + 1 = 19 \quad (1)$$

Here, CLK input at A for Sayem-1 in Figure 7 acts as clock reference for T Flip-flop [1]. Also, constant signals need to be given to respective ports of gates. There are 3 feedbacks associated to ensure proper functioning and hence interconnection delay has to be calibrated as per given clock frequency [1]. T Flip-flops form an integral part of counter design and maintain control flow and synchronization between timing cycles [2].

D. Up/Down Counter

This sequential component is an integral part of any real time computing system [1, 2, 7]. It helps to manage the flow of program execution and adds flexibility to operations. Figure 8 shows the design circuit of a 4-bit reversible counter which incorporates 4 T Flip flops (Sayem based) [1]. Here, the mode bit determines state of counter while CLK signal acts as clock reference for counter.

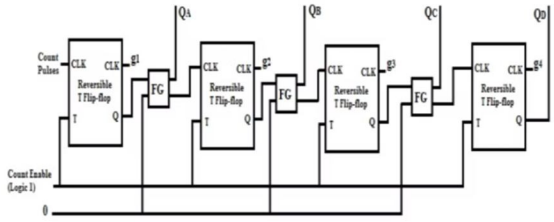


Figure 8. Up/Down Counter circuit [1]
Quantum cost = 4*19 + 3 = 78

The given counter in Figure 8 uses T flip-flop made from Sayem gate and reversibility of this circuit can be examined using providing set of garbage outputs for one state to input and observe previous input at output side [7]. The counter is in up mode when mode bit is high and in down mode when mode bit is low [1]. Counter functioning toggles at the instant at which mode bit flips to either state instead of reaching boundary value of counter range [1], This counter circuit was realized using VHDL code. The simulation of this up/down counter was verified on EDA playground web-tool and waveforms are provided in supplementary material for reference.

IV. DESIGN OF ALU

The design of the Arithmetic Logic Unit (ALU) presented in this work focuses on the implementation of reversible logic to achieve energy-efficient computation, with applicability in low-power VLSI and quantum-aware architectures [8,9]. The proposed reversible ALU is configured to perform five fundamental operations: logical AND, logical OR, logical NOT, binary addition, and binary subtraction. These operations cover a representative subset of typical ALU functionalities while ensuring compatibility with reversible computing principles [8]. A 3-bit operation select line governs the selection of the active operation, and all modules are designed to preserve reversibility by maintaining a bijective mapping between inputs and outputs, thus avoiding information loss.
The ALU logic is constructed using a combination of Feynman, Universal Reversible Gate (URG), and Thapliyal-Srinivas Gate (TSG). The Feynman gate is utilized for signal duplication and bitwise XOR operations, which are essential for implementing the NOT function and supporting intermediate logic in arithmetic units [5].

©IJRASET: All Rights are Reserved

The URG is employed for flexible synthesis of logical operations, including AND and OR, within the constraints of reversibility [2]. For arithmetic computation, the TSG gate serves as a compact and efficient building block for full-adder and subtractor units, offering reduced garbage outputs and lower quantum cost compared to conventional designs [5,6]. Cascading of three URG gates is done as borrow expression in SUB operation is quite complex [2]. The selection logic, responsible for routing the appropriate operation output based on the control lines, is realized using pyramidal arrangement of Fredkin gates, which act as 2x1 reversible multiplexers and maintain the structural integrity required by reversible logic design [9].

The complete design is made in MATLAB Simulink and VHDL code is generated by HDL Coder app available in MATLAB. Further, this VHDL code was simulated, synthesized and implemented on the BASYS-3 development board, featuring a Xilinx ARTIX-7 FPGA. Standard HDL-based modeling is used in conjunction with Xilinx Vivado for synthesis, placement, and timing analysis. The design prioritizes reduction of garbage outputs, minimization of gate count, and control of propagation delay, ensuring suitability for real-time applications. Simulation results validate the functional correctness of all five operations, and synthesis metrics confirm the feasibility of deploying reversible logic-based ALUs on FPGA platforms. This implementation demonstrates a practical pathway toward integrating reversible logic into mainstream hardware systems, bridging the gap between theoretical low-power computing and real-world digital design. Table 2 depicts respective Select Lines orientation for corresponding functionality.

SELECT LINES			ARITHMETIC AND LOGICAL OPERATION	
S2	S1	S0	FUNCTION	BOOLEAN EXPRESSION
0	0	0	OR	A | B
0	0	1	AND	A & B
0	1	0	NOT	~ A
0	1	1	ADD	A + B
1	0	0	SUB	A - B

Table 2. Function/Opcode Table of ALU

A. Single BIT ALU

- Single Bit ALU is capable of 1 Bit operations.
- It takes input as select lines S0,S1,S2,A,B and Cin while outputs Desired Operation Result, Carry/Borrow out and Garbage Outputs.

B. Scaling of Single BIT ALU

- For a 2-bit ALU, single bit ALU has been transformed with inputs {A and A1} and B and B1}, where A1 and B1 are MSB.
- Carry out of LSB is supplied as carry in for the MSB.
- For a 4-bit ALU, single bit alu has been transformed to inputs A(A3 - A) and B(B3 - B) with A3 and B3 as MSB.

V. PERFORMANCE ANALYSIS

The efficiency of proposed design is evaluated in terms of resource utilization and power consumption. However, this design is solely for study purpose and unoptimized paving a way for future upgrades.

Resource	Utilization	Available	Utilization %
LUT	20	20800	0.10
FF	16	41600	0.04
IO	30	106	28.30
BUFG	1	32	3.12

Figure 9. Resource Utilization

- LUTs (Look Up Tables): This design uses 20 LUTs, accounting for 0.10% of total available LUTs.
- Flip Flops (FF): The design uses minimum of sequential elements counting for 16 FFs, which contributes to lower dynamic power consumption.
- I/O Pins: ALU uses 30 pins, indicating multiple control signals, operand lines and output flags.
- Global Clock Buffers (BUFG): One clock buffer is used highlighting efficient clock management.

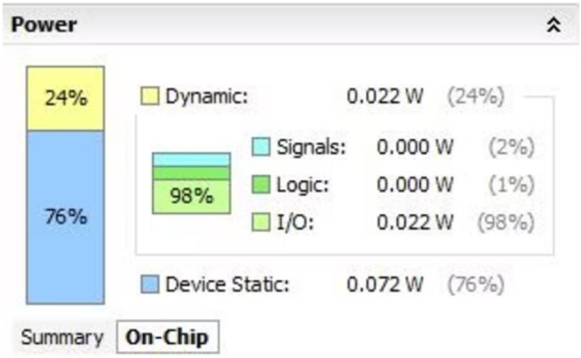


Figure 10. Power Report

Power report in Figure 10 provides brief insight of overall power utilization in proposed design. The static power comes out to be 0.072 W while dynamic power is found to be 0.022 W and most of dynamic power is consumed due to I/O blocks used.

VI. RESULTS

Implementation of 4-Bit Reversible ALU is done as
Inputs A3-A0, B3-B0, S2-S0, Cin are entered through manually operated switches whereas outputs R3-R0, Zero, Cout are located at LEDs' as marked.
Additionally, there are 4 seven segment displays on the FPGA board, which are configured as A, B, Cout and Result respectively. (all values on 7 segment display in Hexadecimal).

1) OR Operation (S2 S1 S0 := 0 0 0)

- Inputs: A3-A0 : 0101 ; B3-B0 : 1010;
- Outputs: Result: 1111; Cout: NA;

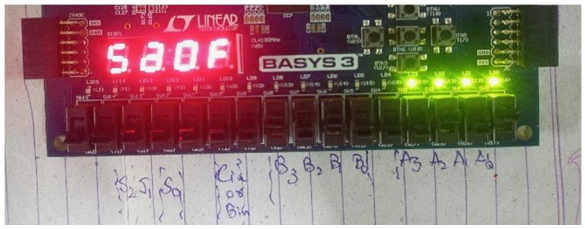


Figure 11. OR operation

2) AND (S2 S1 S0 := 0 0 1)

- Inputs: A3-A0 : 0111 ; B3-B0 : 1010;
- Outputs: Result: 0010; Cout: NA;



Figure 12. AND operation

3) NOT (S2 S1 S0 := 0 1 0)

- Inputs: A3-A0 : 1101 ; B3-B0 : NA;
- Outputs: Result: 0010; Cout: NA;

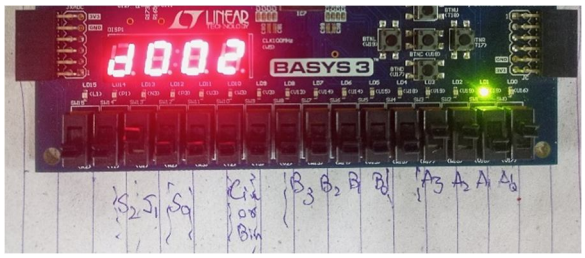


Figure 13. NOT operation

4) ADD (S2 S1 S0 := 0 1 1)

- Inputs: A3-A0 : 1101 ; B3-B0 : 1010;
- Outputs: Result: 0111; Cout: 1;

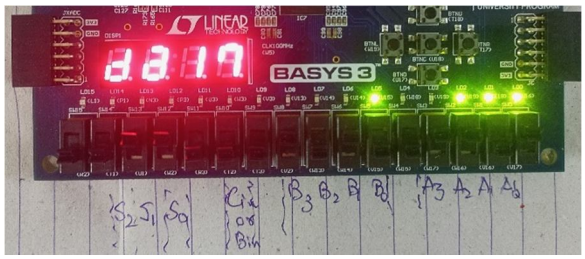


Figure 14. ADD operation

5) SUB (S2 S1 S0 := 1 0 0)

- Inputs: A3-A0 : 1101 ; B3-B0 : 1010;
- Outputs: Result: 0011; Cout: 0;

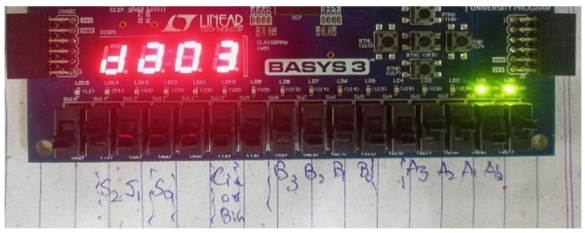


Figure 15. SUB operation

VII. CONCLUSION AND FUTURE SCOPE

From this process of designing an ALU consisting of reversible logic gates, we have known the capability of design environment in VLSI domain to enhance the chip performance many-folds. We conclude that our designed ALU can be essential component in a novel reversible computer architecture and it holds the virtue to be optimized in terms of power consumption, propagation delay, quantum cost and area. The proposed ALU can also serve in day-to-day applications if incorporated with conventional computer architectures to optimize speed of calculation. This research is an attempt to understand and experimentally cater the advancements and capacity of reversible logic in modern-day applications. The re-configurability of FPGA made this research possible. We have understood that mainly three paths of advancements exist for proposed ALU design, which can be stated as:

1) Power-reduction and Optimization is one of the important paradigm one can look into for enhancing the performance of proposed ALU design. This will also help to steer through Landauer's principle to reduce power consumption and make it viable for commercial usage.

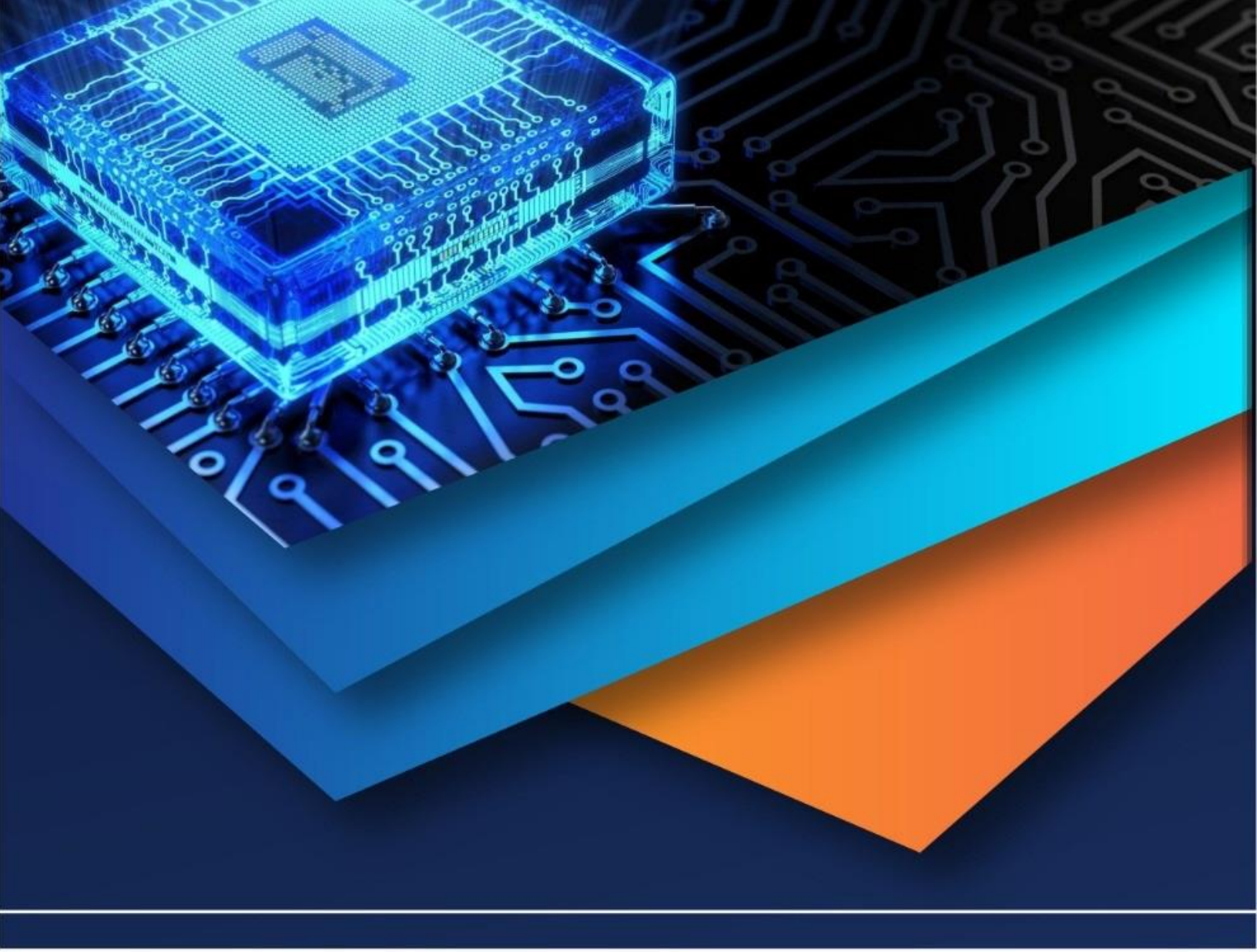
2) Insertion of more operations like shifting, multiplication and more would make this ALU design more acceptable in recent computer architectures. While modified the design, complexity will increase but balance has to be maintained between compactness and ALU functionalities.

3) Lowering the Quantum Cost is a key advancement which can be done in proposed ALU design as it will not only reduce the area consumed but also reduce complexity in terms of time and space simultaneously.

REFERENCES

[1] Singh, Vivek & Ramesh, K.. (2022). Introduction to Reversible Logic Gates and its Operations. International Journal for Research in Applied Science and Engineering Technology. 10. 1148-1156. 10.22214/ijraset.2022.40249.
[2] S. K. S. Hari, S. Shroff, S. N. Mahammad and V. Kamakoti, "Efficient Building Blocks for Reversible Sequential Circuit Design," *2006 49th IEEE International Midwest Symposium on Circuits and Systems*, San Juan, PR, USA, 2006, pp. 437-441, doi: 10.1109/MWSCAS.2006.382092
[3] H.R.BHAGYALAKSHMI, & Venkatesha, M.. (2010). An improved design of a multiplier using reversible logic gates. International Journal on Computer Science and Engineering. 2.
[4] H. Thapliyal and N. Ranganathan, "Design of Reversible Latches Optimized for Quantum Cost, Delay and Garbage Outputs," 2010 23rd International Conference on VLSI Design, Bangalore, India, 2010, pp. 235-240, doi: 10.1109/VLSI.Design.2010.74.
[5] H G, Rangaraju & Venugopal, U & N, Muralidhara & K B, Raja. (2010). Low Power Reversible Parallel Binary Adder/Subtractor. International Journal of VLSI Design & Communication Systems. 1. 10.5121/vlsic.2010.1303.
[6] Thapliyal, Himanshu & Srinivas, M.. (2006). A New Reversible TSG Gate and Its Application For Designing Efficient Adder Circuits.https://doi.org/10.48550/arXiv.cs/0603091
[7] Rahman, Md & Mandal, Indrani & Mamun, Md. (2015). An Optimized Design of Counter Using Reversible Logic. International Journal of Advanced Research in Computer and Communication Engineering. 4. 7-10. 10.17148/IJARCCE.2015.4802.
[8] V. Priyanka, N. S. Reddy, G. Jeevana and M. A. Arab, "Design of Arithmetic Logic Unit Using Reversible Logic Gates," 2024 2nd World Conference on Communication & Computing (WCONF), RAIPUR, India, 2024, pp. 1-6, doi: 10.1109/WCONF61366.2024.10692088
[9] Sharma, Naman & Sachdeva, Rajat & Saraswat, Upanshu & Yadav, Rajat & Kaur, Gunjeet. (2015). Power Efficient Arithmetic Logic Unit Design using Reversible Logic. International Journal of Computer Applications. 128. 36-41. 10.5120/ijca2015906591.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)