International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 11 Issue: 11 | Nov 2024 www.irjet.net p-ISSN: 2395-0072

PERFORMANCE OPTIMIZATION OF 32-BIT ALU IMPLEMENTED WITH REVERSIBLE LOGIC USING PIPELINING AND CLOCK GATING ON FPGA

Chaithra K[1], Dr. K N Rajanikanth[2]

[1]Post Graduate Student, Dept. of ECE, BMS College of Engineering, Bengaluru, India
[2]Associate Professor, Dept. of ECE, BMS College of Engineering, Bengaluru, India

---

Abstract - *Modern electronic devices face challenges in managing power dissipation, making efficient design crucial. This paper presents a high-performance ALU that uses reversible logic gates and Vedic multipliers, implemented with Xilinx Vivado. Reversible logic gates help reduce power consumption and heat by allowing bidirectional data processing, while Vedic multipliers improve arithmetic speed using traditional techniques. The ALU also features clock gating for energy savings and input-based pipelining to boost processing efficiency. ALU is capable of handling various operations, this design offers better power efficiency and speed than conventional ALUs, making it ideal for mobile and embedded applications.*

Key Words: ALU (Arithmetic and logical unit), FPGA (Field programmable gate array), Reversible logic gates, Urdhva Tiryagbhyam, Vedic multiplier.

1. INTRODUCTION

In the rapidly evolving technology of today, numerous devices are created with incredibly tiny sizes, often measured in nanometers, and the Arithmetic and Logic Unit (ALU) is essential in these systems. The ALU is in charge of carrying out mathematical and logical tasks on binary data, made up of zeroes and ones, the basic computer language. Functioning as the primary component of the central processing unit (CPU) to handle calculations, the arithmetic logic unit (ALU) decodes CPU commands and performs operations like addition, subtraction, multiplication, and comparisons to facilitate effective data processing. After getting binary inputs, the ALU performs tasks such as adding numbers and then transmits the outcomes to the CPU for additional purposes. The ALU also carries out logical operations such as AND, OR, XOR, and NOT, in addition to arithmetic functions, which are essential for data comparison and decision-making. The ALU, as a key element in computer design, plays a crucial role in performing a wide range of functions, from basic math operations to complex data handling, and is vital in today's computer systems.

When designing an ALU, it's crucial to prioritize speed and energy efficiency, making sure it operates quickly while consuming minimal power. Certain methods, such as employing reversible gates and Vedic multiplication techniques, can assist in attaining these objectives. Reversible gates are unique as they can handle data while preserving information, enabling energy recovery that is typically wasted. The Vedic multiplier method, rooted in ancient Indian mathematics, enhances multiplication efficiency and simplifies calculations.

Lately, engineers have been exploring novel methods to develop quicker and more energy-efficient ALUs utilizing reversible gates and Vedic multipliers techniques. These sophisticated techniques have considerable promise for enhancing both the efficiency and performance of ALU designs. Utilizing reversible gates that reduce energy waste and Vedic multipliers that streamline intricate calculations, these methods provide significant benefits for contemporary computing systems. This paper will extensively examine the design process utilizing these methods, along with simulated outcomes to confirm their efficacy and showcase their improved performance.

2. CONCEPT

Reversible gates are essential in modern circuit design since they allow computations to move in both directions while maintaining all information. This makes them a unique and efficient tool for managing information. Unlike traditional logic gates that can lead to information loss, reversible gates allow for the retrieval of the original input from the output, making them highly beneficial in various applications. In reversible computing, preserving information is a core concept. Each combination of inputs in a reversible gate must correspond uniquely and reversibly to the output, ensuring that no information is lost during processing. This trait distinguishes reversible gates from irreversible ones, offering the possibility of reduced power use and improved efficiency. In some cases, extra inputs or outputs are needed to guarantee that a circuit can be reversed. The total number of inputs, including fixed inputs, must equal the overall number of outputs, even if certain outputs, known as "garbage" outputs, are not essential for the function. These "garbage" results are crucial for preserving a one-to-one link between inputs and outputs. Figure 1 depicts a reversible logic gate of size n*n with n inputs and n outputs, where every input (In) is connected to its matching output (On).

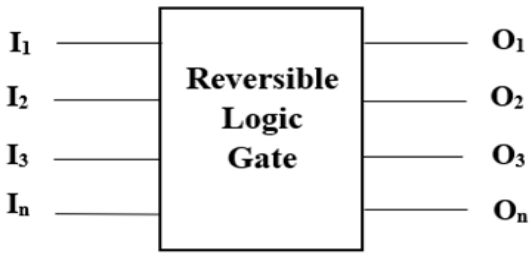


Fig-1: Block diagram of Reversible Logic Gate

A Vedic multiplier is a digital multiplier rooted in ancient Indian Vedic mathematics, utilizing formulas from historical texts. It simplifies multiplication in digital circuits by breaking the task into smaller phases that use particular numerical sequences. The method called "Urdhva Tiryagbhyam" (Vertically and Crosswise) is often employed to reduce partial products, enhancing both speed and efficiency. This technique is employed in modern digital circuit design for quick multiplication in applications like digital signal processing and mathematical computations. The Urdhva Tiryagbhyam method involves generating partial products vertically and merging them diagonally to achieve the final result. Figure 2 depicts the Urdhva Tiryagbhyam method for performing multiplication with binary numbers.

Algorithm for Urdhva Tiryagbhyam Multiplier

1. Divide the numbers: Begin by dividing the two numbers you wish to multiply into smaller components, usually pairs of digits.
2. Vertical multiplication: Multiply each pair of digits vertically and record the partial products one below the other.
3. Diagonal multiplication: Perform crosswise (diagonal) multiplication of each pair of digits and place the results diagonally below the vertical partial products.
4. Sum the partial products: Combine all the partial products obtained from the vertical and crosswise multiplications.
5. Handle carries and finalize: If any carries are generated during addition, carry them to the appropriate positions and calculate the final sum to obtain the result.

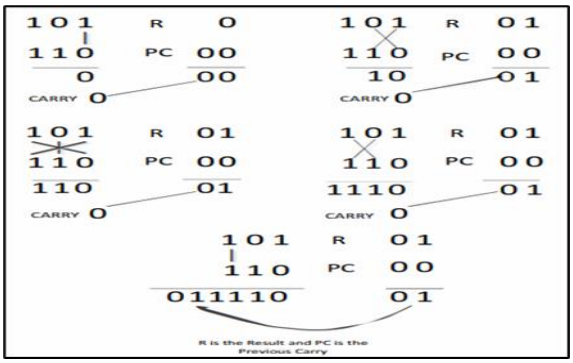


Fig-2: Urdhva Tiryagbhyam Algorithm for binary multiplication

3. LITERATURE SURVEY

Article in reference [1] introduces a reversible ALU design specifically designed for DSP applications, leading to enhancements in power and area efficiency. By incorporating reversible gates such as Fredkin, Toffoli, and Peres, the implementation decreases power consumption by 1.45% and reduces the area needed by 1.79% when compared to conventional ALUs. Created using Verilog and processed with Xilinx ISE 14.7, it showcases improved efficiency and power utilization for systems with limited resources.

Article in reference [2], a comparison is made between a 32-bit ALU constructed using reversible and irreversible logic, with a focus on low-power uses. Reversible gates, which avoid information loss, greatly decrease power consumption. The reversible ALU proposition demonstrates a reduction in delay of up to 48% and a decrease in area of 34% compared to conventional CMOS designs, showcasing its aptness for systems with limited resources and a focus on energy efficiency.

Article in reference [3] investigates the VLSI design and testing of an ALU with Xilinx ISE 14.7, with an emphasis on gate and chip level simulations. The ALU is capable of performing nine different tasks, such as addition and multiplication for arithmetic operations, and AND, OR, and XOR for logical functions. Showing a delay of 125.711 ns, the design highlights the importance of simulation tools in enhancing ALU performance for contemporary digital systems with effectiveness and competitiveness.

Article in reference [4] describes the creation of a 16-bit ALU utilizing reversible logic gates, programmed in Verilog on Xilinx ISE 14.7, and trialed on a Spartan 6 FPGA. The study evaluates how it performs in comparison to a typical ALU, emphasizing the benefits of reversible gates in minimizing power consumption and avoiding data loss. A collection of reversible gates was created to perform different arithmetic and logic functions, highlighting the benefits of reversible logic in enhancing efficiency and dependability in low-power digital systems.

Article in reference [5] introduces a design for a 64-bit ALU that integrates Vedic mathematics principles, employing Sutras such as Urdhva Tiryakbhyam and Nikhilam for arithmetic functions. The Vedic multiplier, running on a Spartan FPGA in Verilog HDL, achieves a delay of only 4.014 ns. Simulation findings exhibited an 85% decrease in delay when compared to traditional methods, emphasizing the capability of Vedic practices to improve effectiveness in high-speed digital circuits.

Article in reference [6] investigates the construction of a 32-bit ALU utilizing reversible logic, contrasting it with a conventional ALU constructed using regular gates. The reversible ALU was tested using ModelSim SE 6.4c and implemented in Xilinx ISE 14.5, resulting in a 5.1% decrease in power usage and utilizing only 7% of FPGA resources. It results in a decrease in delay from 2.266 ns to 1.907 ns and a decrease in power dissipation from 0.312 mW to 0.261 mW, demonstrating its effectiveness for contemporary digital uses.

Article in reference [7] presents a technique for encoding integers in Modified Booth (MB) form for ALU creation with Verilog HDL and Xilinx ISE 14.7, but it does not offer substantial enhancements in area, power, or delay. By integrating an n-bit ALU into a 32-bit framework, the design enhances flexibility and includes a power and area-optimized multiplier, particularly beneficial for neural networks. It also recommends distributing RTL models and extending the ALU capabilities to accommodate different bit widths, putting a focus on efficiency and flexibility in digital circuits.

Article in reference [8] presents a Vedic multiplier that is reversible and designed to reduce power consumption, making it ideal for low-power applications. It includes a compact squarer circuit and a Vedic ALU Controller (VAC) that boosts data processing efficiency, providing better performance compared to conventional designs. The research improves ALU design to enhance resource utilization and processing efficiency.

Article in reference [9] utilizes the Gate Diffusion Input (GDI) technique to decrease power usage and number of transistors, resulting in a smaller chip footprint. The ALU conducts eight functions and consists of a 3T XOR full adder and a fresh 1-to-8 demultiplexer. Experiments conducted using Xilinx ISE 14.7 validate the efficiency of the GDI method in enhancing power and area for VLSI designs.

Article in reference [10] investigates techniques for creating reversible logic circuits with the goal of decreasing expenses through decreasing the number of gates and implementation costs. It contrasts various algorithms, emphasizing their strengths, weaknesses, and uses in green computing and quantum computing. The research suggests the need for more investigation to enhance algorithms for reversible logic synthesis in advanced, low-power systems.

Article in reference [11], a novel ALU design is introduced that relies on reversible computing and incorporates Fredkin and CNOT gates to decrease power usage and maintain data accuracy. The design allows for error detection with parity checks, surpassing traditional ALUs in gate count, quantum cost, and fault detection capabilities. The research emphasizes the benefits of reversible computing for enhancing power efficiency and error detection in digital systems.

Article in reference [12] introduces an optimized design of a single-precision floating-point ALU, prioritizing speed and power efficiency, which are essential for AI, machine learning, and signal processing tasks. Significant enhancements consist of a carry look-ahead adder, altered Booth encoder, and Goldschmidt algorithm to expedite operations. Developed with Verilog HDL and deployed on a Spartan 7 FPGA, the project demonstrated a 36% enhancement in speed, decreased power usage, and improved hardware utilization.

Article in reference [13] explains how an ALU is designed using an 8-bit Programmable Ring Counter, which includes both straight and twisted ring counters. The Arithmetic Logic Unit processes 4-bit binary inputs and executes different arithmetic and logical operations, using ring counter setups to manage the calculations. Utilizing Xilinx ISE 14.2, the goal of the design is to improve computational flexibility and efficiency within digital systems.

Article in reference [14] is focuses on minimizing power usage in digital circuits, specifically in microprocessors found in portable gadgets. The research introduces a 4-bit ALU built for high speed and low power, fine-tuned with 45nm technology and Verilog HDL. It emphasizes how energy-efficient ALUs are crucial for enhancing the efficiency of mobile computing devices.

Article in reference [15], an ALU tailored for microcontrollers is discussed, with a particular emphasis on area efficiency and necessary functions. The design includes Boolean principles, DSP-based multiplier and divider units, and a barrel shifter to enhance performance. Utilizing Xilinx 9.1 ISE and confirmed with ModelSim, the ALU improves resource efficiency and sets the stage for additional microcontroller enhancements.

4. PROPOSED METHODOLOGY

The ALU is a crucial part of a computer's CPU that handles 32-bit data and performs different arithmetic operations like addition, subtraction, and multiplication, along with logical functions such as AND, OR, and XOR. Its activities are controlled by signals from the CPU commands that indicate the function needed. In reversible ALUs, every operation is based on the principle of bijection, ensuring that each unique input combination results in a distinct output that can be reversed to obtain the original inputs. This is achieved by

utilizing reversible gates, which maintain information, reduce power loss, and facilitate energy-efficient computation. The ALUs are equipped with a set of instructions that incorporate reversible arithmetic and logical operations, leveraging reversible gates to enhance performance and energy efficiency in digital systems.

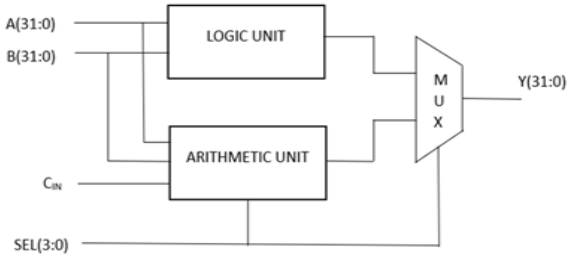


Fig-3: Block diagram of 32-bit ALU

4.1 REVERSIBLE GATES

i. NOT GATE

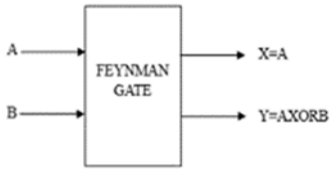
A reversible NOT gate, essential in reversible computing, differs from a traditional NOT gate by allowing the input to be reconstructed from the output, ensuring the process is fully reversible.

A	Y
0	1
1	0

Fig-4: NOT Gate and its truth table

ii. FEYNMAN GATE (FG)

The Feynman gate is a reversible logic gate with two inputs, A and B, generating two outputs, X and Y. The output X equals A, whereas Y is computed as A XOR B. The truth table shows the outputs for all possible input combinations.

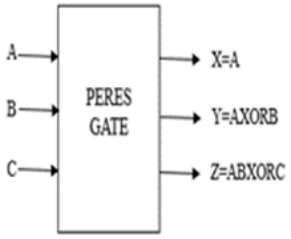


A	B	X	Y
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	0

Fig-5: FG and its truth table

iii. PERES GATE (PG)

The Peres gate is a reversible logic gate with three inputs: A, B, and C, and it generates three outputs: X, Y, and Z. The output X equals A, Y is determined by A XOR B, and Z is produced by XORing the multiplication of A and B with C. The truth table illustrates the outputs for every possible set of input combinations.

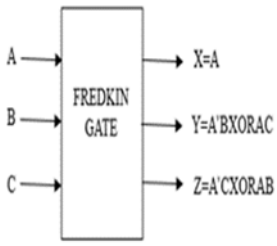


A	B	C	X	Y	Z
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	1	0
1	0	1	1	1	1
1	1	0	1	0	1
1	1	1	1	0	0

Fig-6: PG and its truth table

iv. FREDKIN GATE (FDG)

The Fredkin gate is a reversible logic gate that has three inputs: A, B, and C, along with three outputs: X, Y, and Z. The output X is equal to A, whereas Y is calculated using the formula (Y = (A' B) XOR (AC)). Z is obtained from the formula (Z = (A'C) XOR (AB)). The truth table displays the output values for every possible combination of inputs.

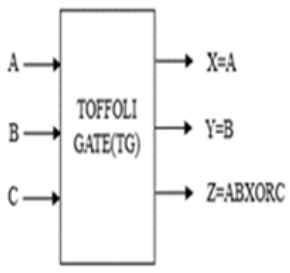


A	B	C	X	Y	Z
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	1	0	0
1	0	1	1	1	0
1	1	0	1	0	1
1	1	1	1	1	1

Fig-7: FDG and its truth table

v. TOFFOLI GATE(TG)

The Toffoli gate, often referred to as the CCNOT gate or controlled-controlled-NOT gate, is a fundamental reversible logic gate used in classical computing. It works with three inputs: A, B, and C. The gate toggles the target bit A only when both control bits B and C are equal to 1; if not, A remains unchanged.



A	B	C	X	Y	Z
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	0	1
1	1	0	1	1	1
1	1	1	1	1	0

Fig-8: TG and its truth table

4.2 Reversible 32-bit ALU

The ALU carries out various functions based on control signals. Its features comprise a 4-bit control signal, two 32-bit digital data inputs, and one 32-bit digital data output. According to the control signals, various arithmetic and logical functions are executed, as outlined in Table 1. The 32-bit reversible ALU can perform 13 distinct operations, all directed by these control signals. The ALU's combinational circuits are constructed with reversible logic gates. Figure 8 illustrates the block diagram of a 32-bit ALU.

Table-1: Functional Description of 32-Bit ALU

Sl. No.	Selection of operation	Sel
1	Addition	0
2	Subtraction	1
3	Multiplication	2
4	Increment	3
5	Decrement	4
6	Logical AND	5
7	Logical OR	6
8	Logical NOR	7
9	Logical NAND	8
10	Logical XOR	9
11	Logical XNOR	10
12	Logical Negation	11

4.2.1 Combinational and Basic gates

Arithmetic and logic units of reversible ALU are designed by using following combinational and basic gates.

1. Reversible Full Adder/Subtractor

A reversible full adder/subtractor circuit can be constructed using Feynman gates (FG) and Peres gates (PG), as illustrated in Figure 8. This circuit enables binary addition and subtraction while maintaining reversibility. It has three inputs: A, B, and Cin (carry in), with a Control Line (CL) that specifies the operation mode. In addition mode (CL=0), PG1 calculates the Sum (S) of A and B, while PG2 generates the Carry (C) for any overflow. In subtraction mode (CL=1), PG1 computes the Difference (D), and PG2 provides the Borrow (B) when A is less than B.

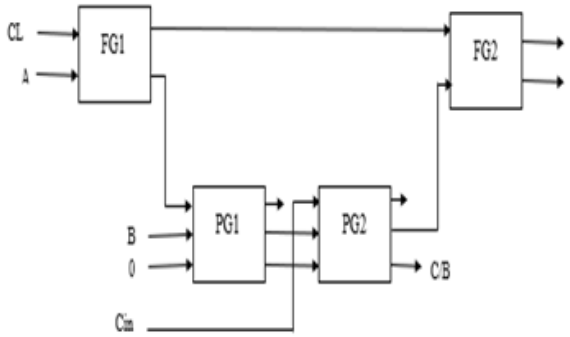


Fig-9: Full adder/Subtractor

2. Reversible multiplier

A Vedic multiplier module implements efficient hardware multiplication using the Urdhva Tiryagbhyam sutra. It multiplies two 32-bit numbers by breaking them into 16-bit halves and using four 16x16 Vedic multipliers to produce 32-bit partial products. 32-bit adders combine these partial products to yield a 64-bit result, divided into four 16-bit sections. This approach streamlines large-number multiplication by leveraging parallel processing and hierarchical module design.

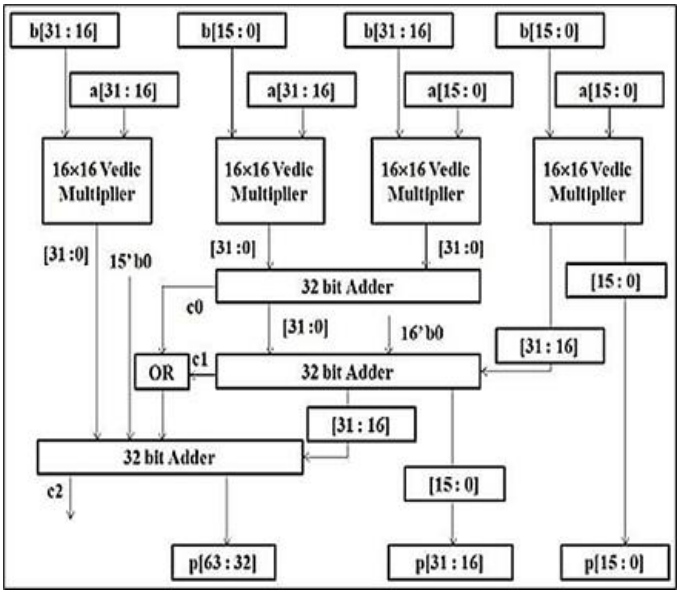


Fig-10: 32-bit Vedic multiplier

3. Increment and Decrement

32-bit increment and decrement operations adjust a 32-bit binary number by one. For incrementing, the operation starts from the rightmost bit, flipping 0 to 1 and stopping, or flipping 1 to 0 and continuing left. An overflow occurs if all bits are 1. For decrementing, the operation starts similarly but flips 1 to 0 and stops, or flips 0 to 1 and continues left. An underflow occurs if all bits are 0.

4. Logical AND

An AND gate constructed with Fredkin gates, also known as controlled-SWAP gates, executes the AND operation by guaranteeing that both output bits are 1 only when both input bits are 1. If either input is 0, the output will also be 0, mirroring the functionality of a standard AND gate.

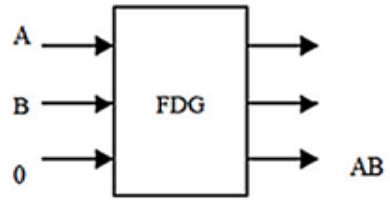


Fig-11: AND Gate using Fredkin

5. Logical OR

An OR gate constructed using Fredkin gates, a form of reversible logic circuit, functions by setting the output bits to 1 if at least one input bit is 1. If both inputs are 0, the output will also be 0, which aligns with the behavior of a conventional OR gate.

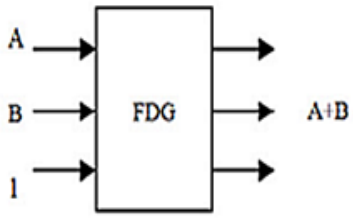


Fig-12: OR Gate using Fredkin

6. Logical NOR

A NOR gate built with Fredkin gates and a NOT gate executes the NOR operation in reversible logic. The Fredkin gate manages the NOR function, while the NOT gate inverts the output to yield the opposite of an OR operation.

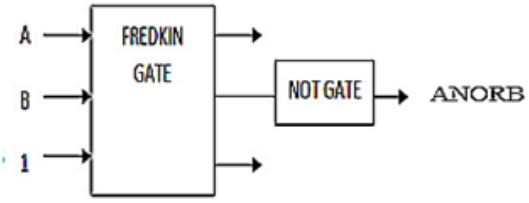


Fig-13: NOR gate using Fredkin and NOT gate

7. Logical NAND

A NAND gate constructed using Toffoli gates is a reversible logic circuit that produces a true output unless both inputs are true. It utilizes two Toffoli gates to toggle input bits and manage the output. The output is false when both inputs are true; if either input is false, the output becomes true, which aligns with the function of a NAND gate.

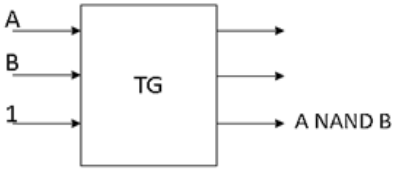


Fig-14: NAND gate using Toffoli

8. Logical XOR

An XOR gate constructed with Feynman gates is a reversible circuit that carries out the exclusive OR function. Feynman gates, which perform controlled-NOT operations, generate an output of 1 when the input bits are different and 0 when they are the same, which reflects the behavior of a standard XOR gate.

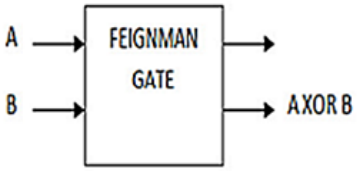


Fig-15: XOR gate using Feynman

9. Logical XNOR

An XNOR gate using Feynman gates and a NOT gate performs the exclusive NOR operation. The Feynman gates manage controlled-NOT functions, while the NOT gate inverts the output, yielding 1 when the input bits match and 0 when they differ, consistent with XNOR logic.

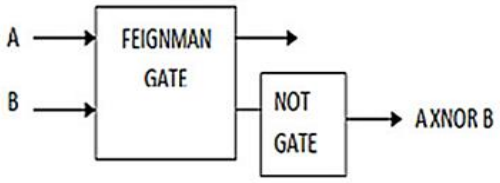


Fig-16: XNOR gate using Feynman and NOT gate

10. Pipelining

Pipelining is a technique that improves instruction processing by breaking it into sequential stages within a pipeline, allowing overlapping execution of multiple instructions. Each pipeline stage includes an input register and a combinational circuit, with the register storing data and the circuit performing operations. This setup increases overall throughput by processing instructions concurrently and reducing the critical path with strategically placed

latches, thus enabling higher speeds. However, it also leads to more latch usage and increased system latency.

11. Clock gating concept

In VLSI circuit design, reducing power dissipation is now a major focus. Previously, designers prioritized optimizing area, delay, and testability, but advancements in technology and shrinking chip sizes have brought power leakage and dissipation to the forefront. To tackle these challenges, techniques such as clock gating and voltage scaling are crucial. This work emphasizes minimizing dynamic power dissipation by reducing signal activity in the design.

Flip-flop-Based Clock Gating

In numerous applications, designs that originally utilized latches are now transitioning to flip-flop-based designs. By splitting a flip-flop, it effectively creates two latches, following the master-slave concept. This method involves using a D flip-flop in combination with an AND gate.

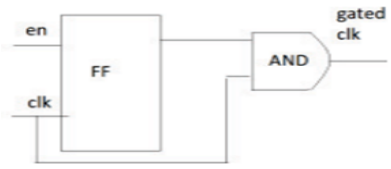


Fig-17: Flip-Flop-Based Clock Gating

As illustrated in the figure, the gated clock signal activates only when both the flip-flop output and the clock signal are high. If either of these is low, the gated clock remains low. Therefore, when the clock is in sleep mode, the gated clock also stays in the low state.

5. RESULTS AND DISCUSSION

The ALU is designed and simulated using Xilinx Vivado, and synthesized for parameter analysis. In Figure 18, inputs A[31:0] and B[31:0] perform different operations based on the selection line, showing results for both conventional and reversible ALUs.

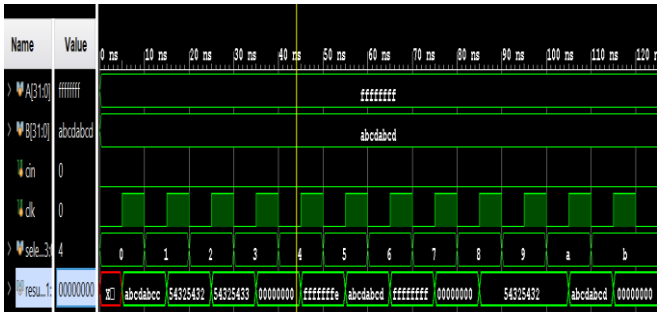


Fig-18: Simulation result of 32bit conventional and reversible ALU

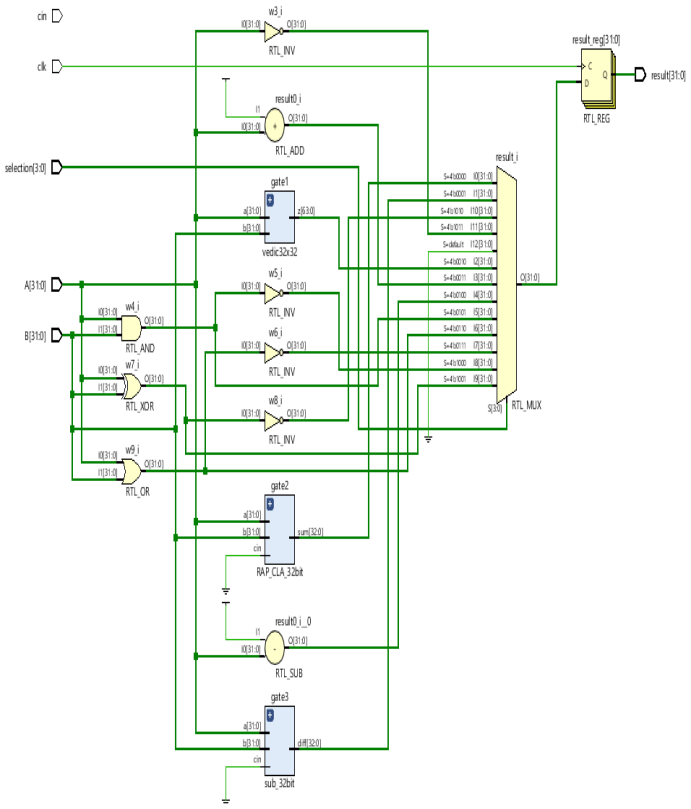


Fig-19: RTL schematic of 32bit conventional ALU

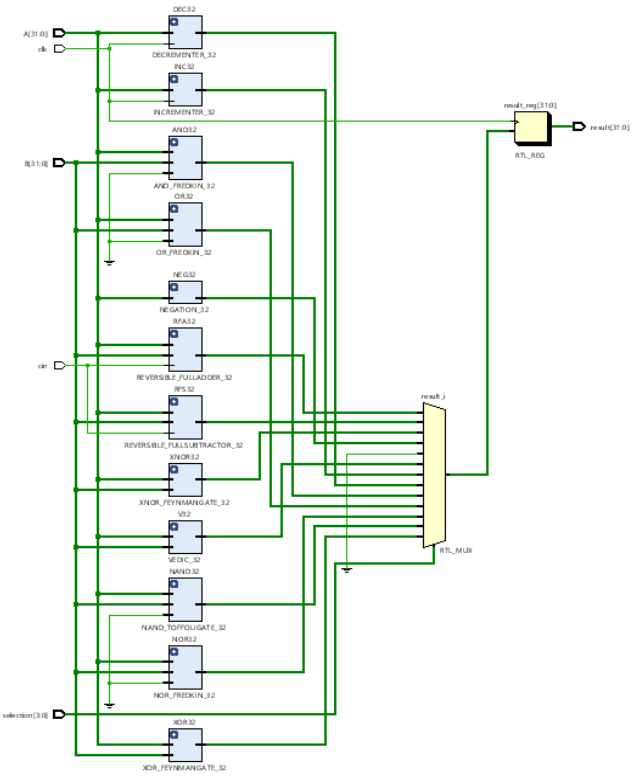


Fig-20: RTL schematic of 32bit reversible logic ALU

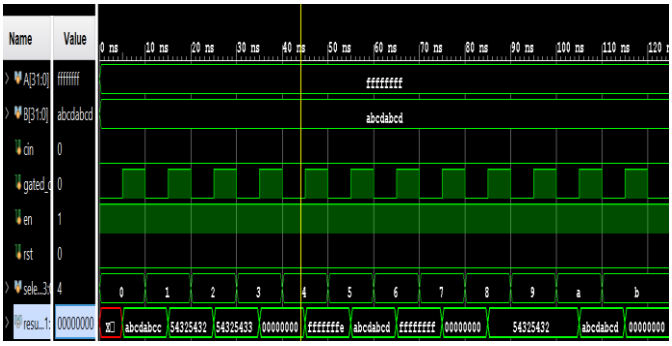


Fig-21: Simulation result of reversible ALU with advanced techniques

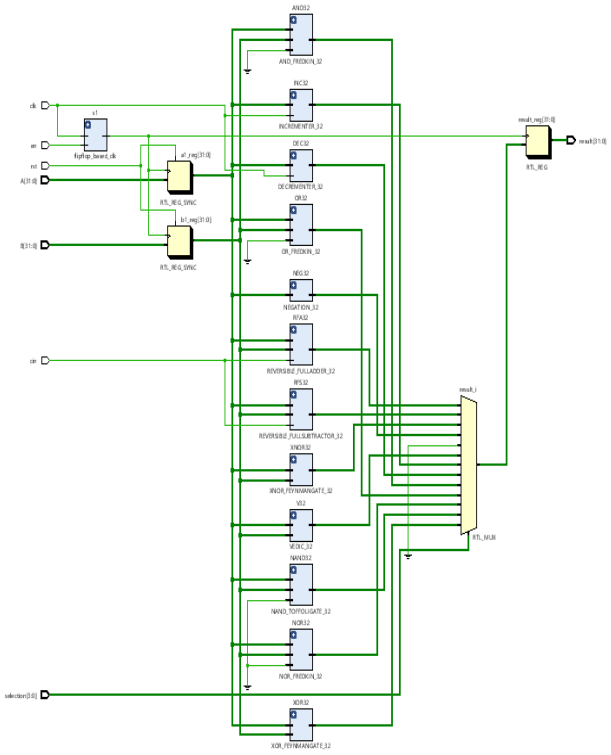


Fig-22: RTL schematic of 32bit reversible logic ALU with advanced techniques

Design	Power(W)	Delay(ns)	Area (LUTs)
Conventional ALU	44.923	13.330	1226
Reversible ALU	44.384	12.835	1166

Table-2: Comparison table of conventional and reversible ALU

Table 2 compares the results of Conventional and Reversible ALUs. Verification tests confirmed that the ALU accurately performs all intended arithmetic and logical operations. Integrating reversible logic gates into the ALU provided significant benefits over conventional designs, particularly in power consumption, area utilization, and delay. The use of clock gating and pipelining techniques further boosted ALU performance, resulting in a 90% reduction in power consumption and a processing speed of 12.511ns.

6. CONCLUSION

In conclusion, this paper presents an effective design strategy for a 32-bit reversible Arithmetic Logic Unit (ALU) that employs clock gating and pipelining techniques. The combination of these approaches leads to significant enhancements in the ALU's performance, achieving a 90% reduction in power consumption and a 2.52% increase in processing speed. Additionally, the incorporation of reversible logic gates boosts the ALU's efficiency, showcasing their potential to create energy-efficient, high-performance digital systems. This work emphasizes the promising future of reversible logic in low-power computing applications.

REFERENCES

[1] Jose, Cissy, T. D. Subash, and Simi P. Thomas. "FPGA implementation of dynamic power, area optimized reversible ALU for various DSP applications." Materials Today: Proceedings 24 (2020): 2044-2053.

[2] S. Nagaraj, B. V. Krishna, B. Chakradhar and D. Sarkar, "Comparison of 32-bit ALU for Reversible Logic and Irreversible Logic," 2021 Innovations in Power and Advanced Computing Technologies (i-PACT), Kuala Lumpur, Malaysia, 2021, pp. 1-5, doi: 10.1109/i-PACT52855.2021.9696935.

[3] G. Surekha, G. Madesh, M. P. Kumar and H. Sriramoju, "Design and Implementation of Arithmetic and Logic Unit (ALU)," 2023 2nd International Conference on Applied Artificial Intelligence and Computing (ICAAIC), Salem, India, 2023, pp. 1530-1536, doi: 10.1109/ICAAIC56838.2023.10140574.

[4] 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.

[5] N. Gadda and U. Eranna, "64-bit ALU Design using Vedic Mathematics," 2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE), Vellore, India, 2020, pp. 1-4, doi: 10.1109/ic-ETITE47903.2020.122.

[6] 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.

[7] V. Priyanka Brahmaiah, V. Kumar Gurrala, S. Teja Tuduru and K. Hari Kishore, "DESIGN OF AREA AND POWER-OPTIMIZED VLSI ARCHITECTURE OF ALUDESIGN USING SIGNED MULTIPLIER," 2022 International Conference on Recent Trends in Microelectronics, Automation, Computing and Communications Systems (ICMACC), Hyderabad, India, 2022, pp. 276-280,doi: 10.1109/ICMACC54824.2022.10093559.

[8] Rayudu, K. V. B. V., D. R. Jahagirdar, and P. Srihari Rao. "Modern Design and Testing of High Speed Vedic ALU Controller using Vedic Algorithms." Journal of The Institution of Engineers (India): Series B 104, no. 1 (2023): 221-230.

[9] S. Sarkar, H. Chatterjee, P. Saha and M. Biswas, "8-Bit ALU Design using m-GDI Technique," 2020 4th International Conference on Trends in Electronics and Informatics (ICOEI)(48184), Tirunelveli, India, 2020, pp. 17-22, doi: 10.1109/ICOEI48184.2020.9142881.

[10] B. Das, N. Vinutha, N. Zafar and M. Rajeshwar, "Synthesis of Reversible Logic Circuits-A Technical Review," 2023 International Conference on Advances in Electronics, Communication, Computing and Intelligent Information Systems (ICAECIS), Bangalore, India, 2023, pp. 309-313, doi:10.1109/ICAECIS58353.2023.10170496.

[11] M. Bhusal, R. Rohith and R. Sakthivel, "Single Bit Fault Detecting ALU Design using Reversible Gates," 2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE), Vellore, India, 2020, pp. 1-6, doi: 10.1109/ic-ETITE47903.2020.326.

[12] A. Y. N J and A. V R, "FPGA Implementation of a High-Speed Efficient Single Precision Floating Point ALU," 2023 International Conference on Control, Communication and Computing (ICCC), Thiruvananthapuram, India, 2023, pp. 1-5, doi: 10.1109/ICCC57789.2023.10165441.

[13] J. L. V. Ramana Kumari, Y. Varshitha, C. Gopi and M. Rakesh, "Design and Implementation of ALU Using Ring Counters," 2023 International Conference on Signal Processing, Computation, Electronics, Power and Telecommunication (IConSCEPT), Karaikal, India, 2023, pp. 1-5, doi: 10.1109/IConSCEPT57958.2023.10170295.

[14] J. L. V. Ramana Kumari, Y. Varshitha, C. Gopi and M. Rakesh, "Design and Implementation of ALU Using Ring Counters," 2023 International Conference on Signal Processing, Computation, Electronics, Power and Telecommunication (IConSCEPT), Karaikal, India, 2023, pp. 1-5, doi: 10.1109/IConSCEPT57958.2023.10170295.

[15] A. A. Purohit, M. R. Ahmed and R. V. S. Reddy, "Design of Area Optimized Arithmetic and Logical Unit for Microcontroller," 2020 IEEE VLSI DEVICE CIRCUIT AND SYSTEM (VLSI DCS), Kolkata, India, 2020, pp. 335-339, doi: 10.1109/VLSIDCS47293.2020.9179942.