Journal of VLSI circuits and systems

RESEARCH ARTICLE

Journal of VLSI Circuits and Systems, ISSN: 2582-1458 Vol. 7, No. 1, 2025 (pp. 198-209)
WWW.VLSIJOURNAL.COM

Energy-Efficient High Performance 64-bit ALU using Reversible Logic Based on Self Error Detection and Correction Technique

Veeresh K.[1]*, Vilaskumar Patil[2]

[1]*Sharnbasva University, Kalaburagi, Department of Electronics and communication Engineering, Veerappa Nisty Engineering College, Karnataka, India.*

[2]*Department of Electronics and Communication Engineering, Sharnbasva University, Kalaburagi, Karnataka, India.*

KEYWORDS:
FPGA,
ALU,
Reversible Logic,
Majority Gate,
Array Multiplier,
Processor Design,
Parity Checking, ECC.

ARTICLE HISTORY:
Received: 12.06.2025
Revised: 21.07.2025
Accepted: 28.07.2025

DOI:
https://doi.org/10.31838/JVCS/07.01.21

Abstract

The aim of this research proposal is to design a self-error detection-based arithmetic operations unit, which is widely required in today's AI technology for performing high-speed calculations of arithmetic and logical operations, and is a top priority in deep learning algorithms. In the proposed method, a high-speed and double precision arithmetic and logic unit (ALU) is designed, which employs reversible majority gate technology along with parity checking and error-correcting codes (ECC) to improve fault tolerance for low-power computing systems. Reversible logic gates are given high priority in larger circuit designs and the use of more FPGA resources, as they reduce garbage signals and critical path delay in combinational design architecture. This reversible logic reduces the signal path and increases energy efficiency. Furthermore, to simplify complex logic operations, the novelty architecture designed a full adder using the majority logic (ML) gate, which uses reversible methodology that is specially used for Feynman and Toffoli gates. For self-error correction, parity checking is utilized to identify single-bit mistakes, whereas Hamming code-based error correction was also included for multibit mistakes, and these hybrid implementations are helpful to detect and rectify errors. Furthermore, the proposed ALU facilitates fundamental operations such as addition, subtraction, and multiplication, which are designed with the proposed ML full adder architecture and finally evaluated for decreased power consumption and enhanced error resistance. The proposed design, synthesized and tested on the Xilinx Vivado Artix-7 FPGA platform, exhibits up to 52% reduced power consumption and utilizes 7.5% fewer LUTs than traditional designs.

Authors' e-mail ID: veeresh.kumasagi@gmail.com, mpvilaskumar@gmail.com

Authors' Orcid ID: 0009-0009-8407-1845, 0000-0003-3038-9283

How to cite this article: Veeresh, K. and Patil, V. Energy-Efficient High Performance 64-bit ALU using Reversible Logic Based on Self Error Detection and Correction Technique, Journal of VLSI circuits and systems, Vol. 7, No.1, 2025 (pp. 198-209).

Introduction

The demands for high-speed, fault-tolerant arithmetic operations and energy-efficient systems for computing has increased in the ever-changing technological environment of today.[4] It is mostly prioritized in aerospace systems, where common issues of radiation error, signal disturbance, vibrations, highly sensitive noise, and environmental interference arise.[8] In embedded applications, fault-tolerant issues are also caused by voltage fluctuations, temperature changes, electromagnetic interference, data corruption, or system failure.[21] The proposed methodology addressing these issues is crucial to maintain consistent performance and to get advanced technology without data losses because of the error.[23] In today's all applications environment, arithmetic operations are most common one from the core of data processing in aerospace system, quantum computing and low power computing devices.[2] As for results, all these

applications require self-error correction-based arithmetic operations. Thus, the proposed method investigates, with low power environment platform and find out to resolve error resilient, critical path and low power circuit components.[6] As for this study, reversible logic and ML are the most important methods to explore for formulating a method with high performance and reliability.[1],[16] By using reversible logic, the design ensures computations, lower energy, and improved efficiency.[25] The ML design simplifies complex arithmetic operations and also helps to enhance the fault tolerance circuit, making it suitable for mission critical applications.[10] Hence, the proposed architecture designs a novel ALU design using this reversible and ML design.[7] Moreover, the self-error correction system is the goal of this design. In this regard, for basic operations, the research investigated the parity checking method, and for larger bit size of arithmetic operations, the design required forward error correction (FEC) methods of AN codes, BCH code, LDPC codes, hamming codes, turbo codes, and so on.[23] An analysis of the hamming codes will require very less logic sizes and also provide reliable error corrections.[23] For that, the proposed architecture design will involve hamming error correction and detection-based arithmetic self-testing method.[23] For timing and complexity reduction, the single-bit error detection solves with parity-based error correction, and for multibit operation, the error is detected and corrected with the hamming method.[21] The proposed architecture of error detection and correction method is shown in Figure 1, and the overall proposed ALU architecture is aimed at providing low-power error-resilient architecture for modern computing applications.[18]

Furthermore, in the implementation of Figure 1, the basic key component is a single-bit adder design. This adder design will occupy more logic size in all of the digital circuits, even multiplier design architecture is

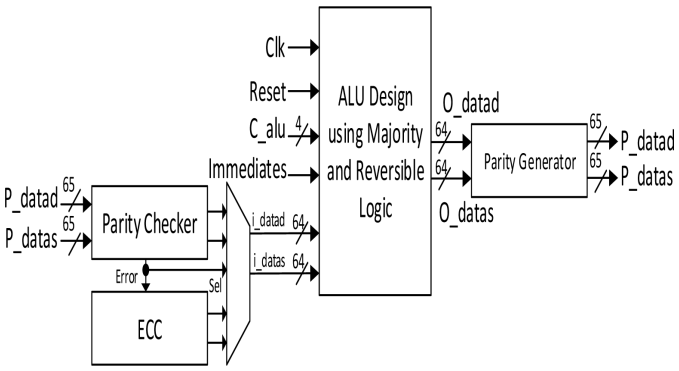


Fig. 1: The architecture of the proposed ALU with error detection and correction method.

constructed with this adder.[18],[19] To analyze the fundamental operations of this ALU, it will present with addition, subtraction, multiplication, division, shifting, and logical method of operations.[1] For the future study of this work, this proposed ALU is integrated with self-error detection-based RISC processor design architecture.[24] For the processor design, this ALU is also prioritized with the opcode method. The C_alu in Figure 1 architecture controls the ALU operations including parity checker and ECC.[4] The input signal P_datad and P_datas is initially given to the parity checker, which identifies the single-bit error and corrects the error.[23] The output operations of Q_datad and Q_datas again generates the parity generations and then provides the results in F_datad and F_datas.[21] As of this functionality, the processor-based ALU design architecture is required for addition, subtraction, increment, decrement, negative conversion, XOR operations, multiplication, and divisions; these operations will have prioritized based on C_alu, which input we need to process is an immediate input or register input and which type of registers.[5] By analysis of this processor based ALU operations, the adder is mostly used in all the design, while multiplications and increment.[9] For the novelty aim of this proposed research to solve with low power consumption, error resilient, critical path reduction and less logic utilization based a novelty adder required for further research.[18] Similarly, this research also proves with Xilinx Vivado FPGA platform and prove the simulation analysis using Modelsim.[24]

To achieve this novel, adder architecture, the proposed design utilized the Feynman and Toffoli gates, two well-known reversible logic gates, to create innovative ML operations of ab + bc + ca.[13],[12],[14] Using these reversible ML gates, the proposed full adder design was developed.[19] This novel, full adder design architecture focused on reducing logical complexity and eliminating unnecessary outputs.[18],[28] This design makes an efficient reversible logic full adder design, and it's suitable for addition, multiplication, and other significant operations.[1] Moreover, the architecture of complete reversible ML-based ALU design significantly reduces the amount of logical complexity and garbage outputs, and it's helpful for error-resilient applications.[7] The remaining part of this work presents the literature review in Section II. The evaluation and effectiveness of the proposed reversible full adder design is discussed in Section III, and the proposed reversible multiplier design architecture is described in Section IV. The error correction design of parity and hamming design with ALU is described in Section V. The conclusion of this research is presented in Section VI.

Literature Review

The recent advancement in fault-tolerant arithmetic operations focuses on error resilience and power efficiency, especially in the application of signal processing, embedded computing, and AI technology.[4],[21] FEC is also focused on rectifying the problems in digital systems to detect and correct the errors automatically, without requiring the data to be resent.[23] This method has a lot of correction and rectification systems, such as parity codes, Hamming codes, BCH codes, Reed–Solomon codes, Viterbi, LDPC, Turbo, and so on.[4] Moreover, these FEC codes were explored in earlier works.[23] This proposed approach has motivated ongoing research into integrating error correction codes (ECC) with modern logic style to further enhance the system and minimize hardware overhead.[21] In all digital systems, the processor is the core of the architecture, and it's evaluated with the number of arithmetic operations, the number of recent technologies of logic size reduction and power consumption, the number of research projects focused on approximate computing, and also focuses on error minimization.[10] If the approximate data are given into the accurate circuit input means, it also operates with approximate functionality, but the logic size can't be reduced. However, the processor is a high-precision one; no one gadget is present with an approximate processor design. Here, error correction is enormously important.[23] Similarly, errors in processor arithmetic operations can happen because of several reasons, such as (1) hardware fault: it may happen because of transistor defects or interconnection of logic circuits.[6] (2) Noise and interference: there may be spikes, disturbances, and electromagnetic interference.[21] (3) Radiation effect: it may happen in high-altitude environments with cosmic rays.[8] (4) Timing error: it may happen because of clock slew, critical paths, and setup and hold time issues.[22] (5) Overheating error: It may happen in high temperatures. Therefore, the proposed configurability processor arithmetic operations are required to be validated with proper error corrections and also optimized for resource usage and power consumption.[1] In particular, the introduction of reversible logic circuits integrated with arithmetic design and also with ECC addresses the problem of energy dissipation linked to irreversible operations.[25]

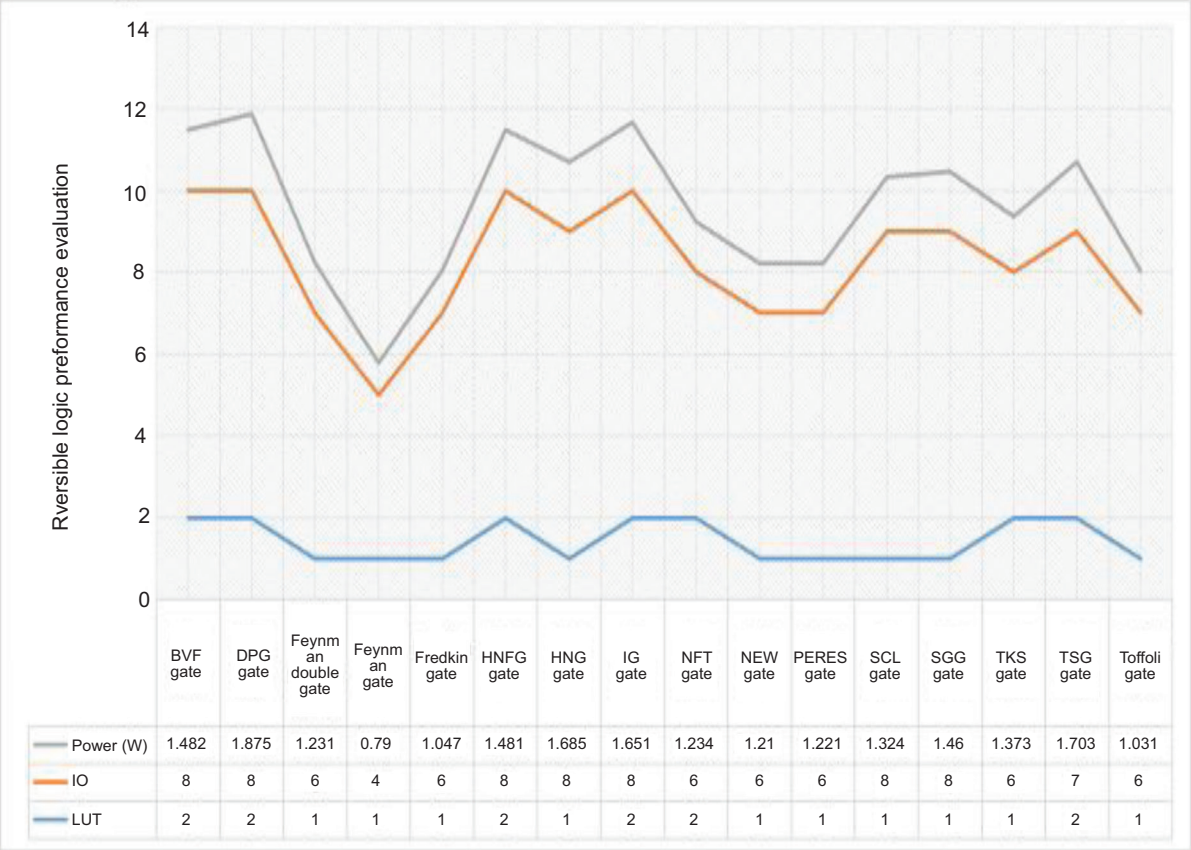
Integrating this parity checking with reversible circuits has increased the efficiency of single-bit fault corrections and detection, and this enables the generation of real-time error signals.[23] Such a technique reduced the quantum cost and unnecessary garbage signals for scalability and practical hardware realization.[14] The evolution of programmable and logical, flexible, and reversible gates, such as BVF Gate, DPG Gate, Feynman Double Gate, Feynman Gate, Fredkin Gate, HNFG Gate, HNG Gate, IG Gate, NFT Gate, NEW Gate, PERES Gate, SCL Gate, SGG Gate, TKS Gate, TSG Gate, and Toffoli Gate, further supports the construction of ALU with lower quantum cost, improved speed, and output configurations.[12],[13],[14] Another notable complex arithmetic circuit is the multiplier design, which takes more partial products, logic size, and power consumption compared to all traditional design multipliers.[9] Here, this integration of reversible full adder and majority gate has shown substantial and efficient implementation in the multiplication design.[17] The introduction of ML with reversible technique in multiplier design has further optimized circuit performance for error-tolerant applications, offering a balance between computational accuracy, speed, and hardware cost.[16],[20]

Performance of the Proposed Reversible Full Adder Design

Reversible Logic Gates

Reversible logic gates are exceptional types of gates where every input and output produces a unique value; compared to the traditional logic gates, this reversible gate also provides the same functionality along with some additional outputs. It passes through bypassing the original values to the output.[13] It will help minimize the power loss by preserving information throughout the circuit.[25] The priority of reversible logic gates is increasing, especially in the quantum computing method, Thereafter, in this reversible logic common gate is Feynman gates, which can XOR the signal, and Toffoli gate, which is used for universal logic operations, and the Fredkin gates, which are used for control and swapping the bits.[14],[12] Especially, the bypassing input will be used for another subsequent gate in digital circuits; it will help minimize the logic connections and increase the efficiency and reliability.[14]

In this research analysis, the logical analysis of reversible logic gates, including the BVF Gate, DPG Gate, Feynman Double Gate, Feynman Gate,[12] Fredkin Gate, HNFG Gate, HNG Gate,[13] IG Gate, NFT Gate, NEW Gate, PERES Gate, SCL Gate, SGG Gate, TKS Gate, TSG Gate, and Toffoli Gate,[14] is the fundamental aspect of these circuits. Each gate is designed in Verilog HDL and synthesized in Xilinx FPGA. They possess distinct logic sizes and power consumption levels. The proposed technique employs the analysis of several reversible logic gates and evaluates their enhanced performance, especially in the creation of an innovative reversible full adder. Figure 2 presents the



	BVF gate	DPG gate	Feynman double gate	Feynman gate	Fredkin gate	HNFG gate	HNG gate	IG gate	NFT gate	NEW gate	PERES gate	SCL gate	SGG gate	TKS gate	TSG gate	Toffoli gate
Power (W)	1.482	1.875	1.231	0.79	1.047	1.481	1.685	1.651	1.234	1.21	1.221	1.324	1.46	1.373	1.703	1.031
IO	8	8	6	4	6	8	8	8	6	6	6	8	8	6	7	6
LUT	2	2	1	1	1	2	1	2	2	1	1	1	1	1	2	1

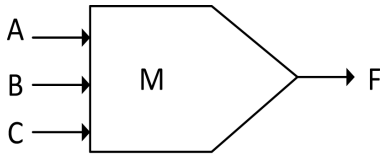
Fig. 2: The comparative evaluations of 16 distinct reversible logic gates, namely, Feynman, Fredkin, and Toffoli, will determine the least logic size and power usage.

comparative assessments of 16 distinct reversible logic gates. This methodology seeks to illustrate the benefits of reversible logic gates in developing more efficient and effective computing architecture.

Majority Gate Design Using Reversible Logic Gates

ML gates are a priority in recent technology like quantum computing and spintronics.[15],[26] These gates are mostly required in complex arithmetic operations. They lead to smaller, faster, and more energy-efficient circuits, and these majority gates with three input logic gates are especially helpful in designing strong, fault-tolerant circuits, as they can even find a faulty bit.[16]

The proposed approach of this design is useful in implementing a reversible logic gate, specifically Feynman and Toffoli gates. It enables lossless information in VLSI design circuits.[12],[13] The advantages of this novel approach are no loss of information, which ensures reversibility, minimizes power consumption, and prevents heat generation.[25] It's scalable, which reduces hardware complexity and enhances the natural process of fault tolerance, which can help in error correction



$$F = M(A, B, C) = AB + BC + AC$$

Fig. 3: Three-input majority logic gate with logical equation.

and detection methods.[17] Applications like low-power CMOS technology, nanotechnology, and quantum computing will benefit greatly from it.[15], [16] When it comes to reversible quantum processes, this approach is seen as crucial. When designing an ML gate, using reversible logic yields better outcomes in mathematical processes.[20] The three-input ML gate and its corresponding equation are seen in Figure 3. We need to use high-performance reversible logic gates to create this ML gate utilizing reversible logic. This requires creating a number of different reversible logic gates and then picking the most effective one to act as the majority gate. Our recommendation for the three-input ML gate is based on the comparison study shown in Figure 2,

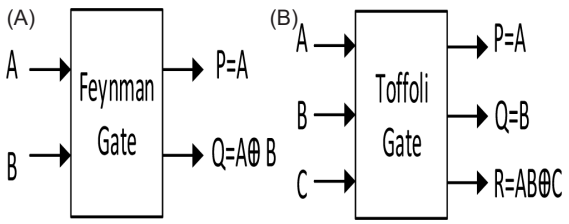


Fig. 4: (A) Feynman gate and (B) Toffoli gate.

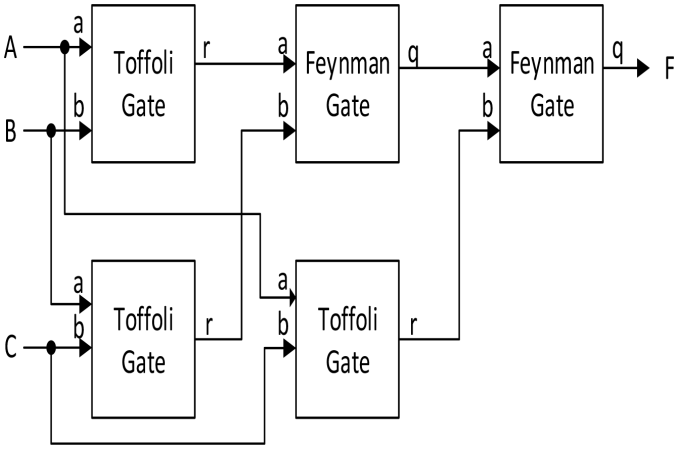


Fig. 5: Majority logic gate design using Feynman and Toffoli gate.

which suggests the Feynman Gate and the Toffoli Gate designs. Feynman and Toffoli Gate architecture is shown in Figure 4. This design builds a more efficient and effective configuration of three-Input ML gates by incorporating these gates into the design.[17] The Feynman Gate and the Toffoli Gate were selected because they outperformed other gates in reversible logic comparisons with regard to logic size, power consumption, and overall efficiency.

An architecture that uses an ML design based on Feynman and Toffoli gates, with three Toffoli gates and two Feynman gates, is shown in Figure 5. When it comes to logic size and power consumption, this architecture is meant to maximize performance. With respect to the three-input ML gate, Table 1 shows the results of the binary truth table. In this case, the inputs to the reversible majority gate are A, B, and C, and the output is F. To better understand how the ML gate functions in various input situations, this table shows the logical link between the inputs and the outputs.

Full Adder Design Using Reversible ML gate

The implementation of a full adder using an ML gate is more to reduce the circuit complexity, especially in

Table 1: Truth table for three-input majority logic gate.

A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

emerging technologies like the QCA.[17] The traditional full adder design will have five logic gates, with each taking two input logic gates only. Compared to this, the novel approach here will take three input levels to reduce the complexity in the full adder design. In Figure 6, the design shows the full adder design using three-input ML gates. This design occupies only three ML gates instead of five traditional logic gates. This ML gate with reversible functionality will also reduce the number of interconnections in quantum computing.[19]

The comparative analysis of the proposed novel full adder design compare to the traditional reversible full adder design, in this analysis with conventional majority full adder, PERES gate full adder design, Fredkin Feynman full adder design, HNG gate full adder design. As of analysis this design in FPGA Xilinx verifications, the table graph update in Figure 7. Here, LUT will take very less in proposed reversible majority full adder it occupies only 6485 LUT, it compared to very less in PERES FA, HNG FA, both of this PERES and HNG full adder occupy 16,193 LUT, it attains 33.33% of enhancement in efficiency. The power analysis also occupied with very less in proposed method of 0.401 W, but in traditional it will occupy 0.947 W power consumption.[18]

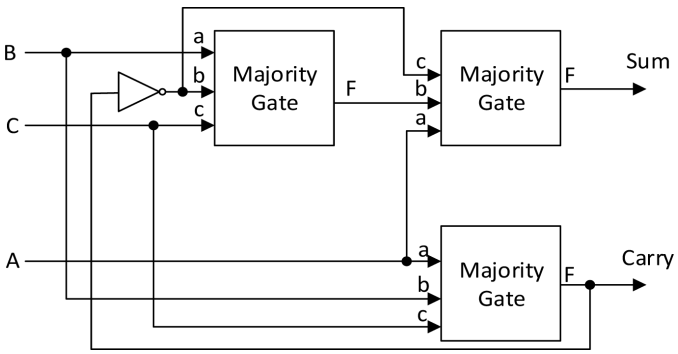
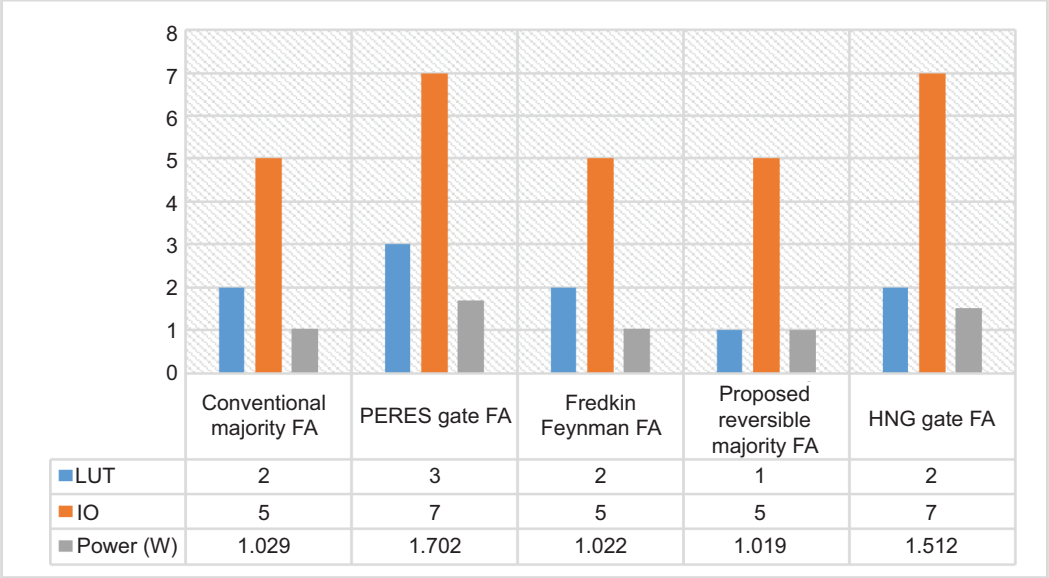


Fig. 6: Full adder design using the majority logic gate.



	Conventional majority FA	PERES gate FA	Fredkin Feynman FA	Proposed reversible majority FA	HNG gate FA
LUT	2	3	2	1	2
IO	5	7	5	5	7
Power (W)	1.029	1.702	1.022	1.019	1.512

Fig. 7: Comparative analysis of the proposed reversible majority full adder.

Performance of the Proposed Reversible 64-Bit Multiplier

The valuation of the proposed 64-bit array multiplier design with the novel reversible ML full adder architecture, its reduces the number of interconnections, and logic size and power consumptions.[20] Moreover, the design's intrinsic capacity to reduce garbage interconnections outputs guarantees accurate computational results, it's essential for complicated processes such as multiplication. The multiplier effectively many partial product reductions simultaneously, and its highlighting scalability and parallelism. The proposed 64-bit array multiplier concurrently processes 4096 partial products, from a0b0 to a63b63. This parallel systolic array method decreases total computing time relative to sequential multiplication techniques. The proposed methodology pursues to minimize the logic size of array multipliers in extensive multiplication contexts by utilizing reversible ML techniques.[17],[27], Figure 8 depicts the architecture of the parallel 64 × 64 array multiplier design. A comprehensive comparative analysis was performed on this array multiplier design, its utilizing different full adder configurations: the traditional majority gate full adder, the Fredkin Feynman gate full adder, the PERES gate full adder, the HNG gate full adder, and the proposed reversible ML full adder. The results of this investigation are encapsulated and revised in Figure 9.

The comparative analysis of proposed reversible ML array multiplier design given in Figure 9. In this comparative

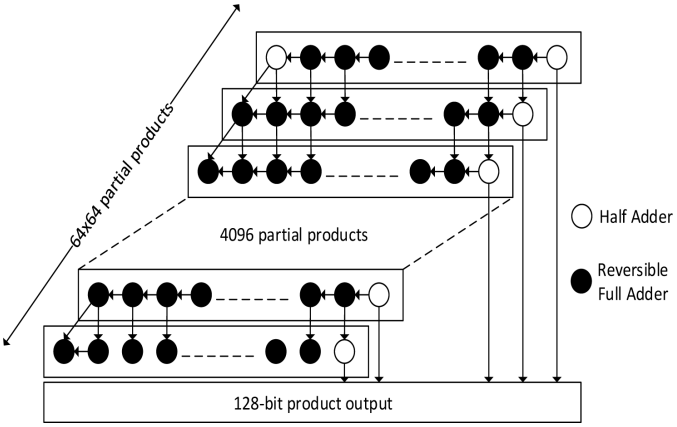


Fig. 8: Parallel 64 × 64 array multiplier architecture.

method analysis with number of LUT and power consumptions, the proposed multiplier design has realized very low LUT consumptions from 61.31% to 59.24% when compared to the PERES Gate, Fredkin, Feynman and HNG gate based multiplier design. Similarly, the proposed design exhibits very less power consumptions ranging from 60.78% to 60.68%, when compare the traditional design. These reductions of LUT and power analysis will be computing high efficiency in speed of processor and ALU design.[20]

Parity and Hamming-Based Error Correction Technique

Parity checking is a method for detecting an error in digital data transmission and reception; it is a

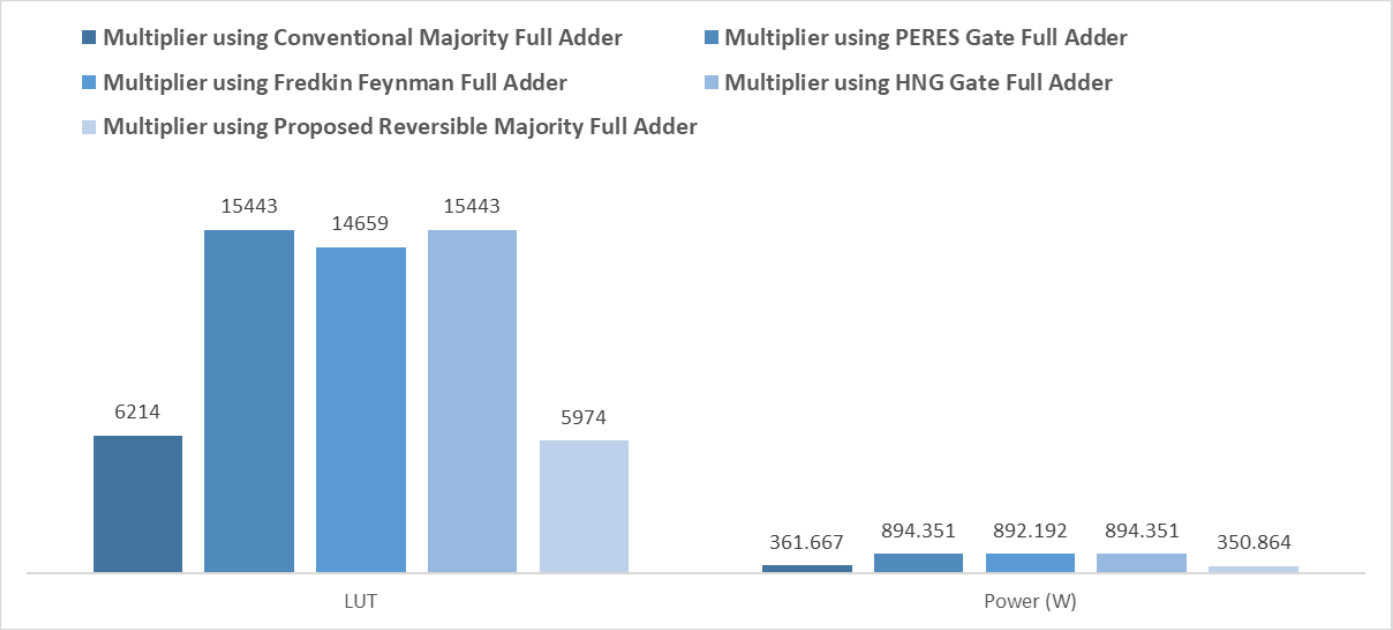


Fig. 9: Comparative analysis of 64-bit array multiplier using conventional and reversible full adders.

single-bit ECC. it simply tests the parity which using even or odd parity, even means calculating total number of 1s in the vector, and odd means calculating total number of 0s in the vector.[23] However, the Hamming code is also an ECC by Richard Hamming. It is specifically used for more than one-bit error corrections, which means it's a single ECC.[21] As per this approach, a combination of parity and Hamming code works in error detection and correction in all digital gadgets and applications. Advantages of this approach are that it is useful in single-bit error correction and also suitable for double-bit error correction. its more optimized the memory.[23] In the proposed ALU design, parity checkers are placed at both the input and output stages. The analysis of Figure 10 indicates that the proposed architecture has two input data bits, namely, P_datad and P_datas. These two signals are calculated with parity. if its produces single error correction the parity checker will take the responsibilities, if multibit error hamming will have the single ECC, it will take responsibilities, here key of the selection automatically decided by error predictions.[21] The procedure of parity generation is computed as Equation (1):

$$P = d_1 \oplus d_2 \oplus .... \oplus d_n \quad (1)$$

Thus, the analysis in Equation (1), the parity bit calculated as XOR combinations, its ensuring the code word maintaining with parity conditions, for analysis of data word length k, the hamming codes adds the parity bits "r", where 2r>k+r+1.[23] By the following the reference to single-error correction, the parity bit is arranged in positions that are power of 2 (i.e., 1,2,4,8). Each parity bit position is calculated as XOR logic. In the bit vector mode, the data bit comes with four positioning levels of D=d1, d2, d3, and d4, and the parity bits p1, p2, and p3 are computed as Equation (2):

$$\begin{aligned} P_1 &= d_1 \oplus d_2 \oplus d_3 \\ P_2 &= d_1 \oplus d_2 \oplus d_4 \\ P_3 &= d_1 \oplus d_3 \oplus d_4 \end{aligned} \quad (2)$$

Following the parity computations, the word CC is used for interleaving the data bit and parity bit. Whenever the data are received from the parity computations, the Hamming design needs to calculate a syndrome and analyze the error. Then, syndrome vectors indicate the positions of the error bits, and it allows the systems to flip and interconnect bits and restore the original data.[21] For example, the above syndrome vector S equal to S1, S2, and S3 is computed as:

$$\begin{aligned} S_1 &= P_1 \oplus d_1 \oplus d_2 \oplus d_3 \\ S_2 &= P_2 \oplus d_1 \oplus d_2 \oplus d_4 \\ S_3 &= P_3 \oplus d_1 \oplus d_3 \oplus d_4 \end{aligned} \quad (3)$$

Following Equation 3, the syndrome vector is used to identify the error positions of the data. For instance, S = 011. It indicates that the bit position of the third

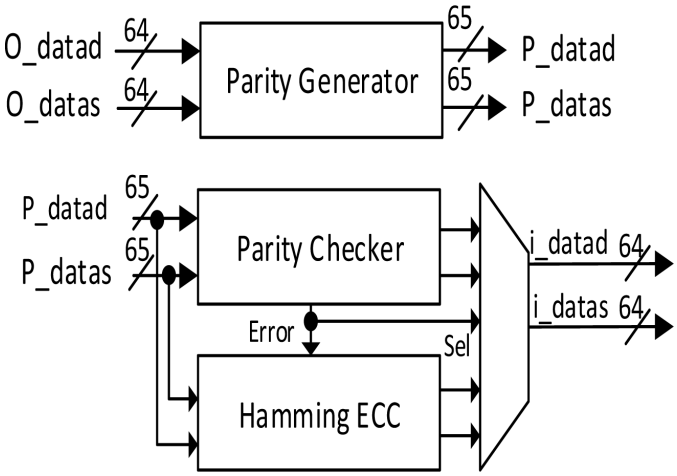


Fig. 10: Combination of parity checking and hamming error correction.

vector is an error, and it needs to be corrected. From Figure 10, which shows the combinations of this, the proposed architecture, design of parity checking, and the Hamming ECC, it offers several advantages, and it is tried with a two number of approaches, one is a robust protection against single-bit error correction, and the other is ensuring the data integrity through the computation process, of hamming code.[21] The parity checker of this hamming code, it is imposing a multiple resources and power overhead.

The error detection and correction of hamming and parity proposed fault detection system was developed in Verilog HDL and simulation and synthesis done in Xilinx Vivado Artix-7 FPGA. For analysis of single bit correction will occupy approximately 0.7nm in delay and 0.03 W in power consumption. This proposed architecture is more reliable to ensure all the ALU arithmetic operations and the performance of key metrics such as error detection rate, correction accuracy, resource utilization, and computational efficiency.[21] The simulation also provided 99.9% of accurate evaluation in fault-tolerant scenarios, and the ECC will occupy minimal logic utilization of 7.5% in LUTs.

Operation and Performance Analysis of the Proposed Reversible ALU Design with ECC

The integration of this proposed ECC with parity checking in ALU design architecture is especially required for all the digital platforms of arithmetic operations. The comprehensive examination of the operational flow of the ALU design also comes with low latency, reduced hardware complexity, and very low garbage outputs.[1]

The internal architecture of this ALU presents with a 64-bit architecture design, and it performs addition, subtraction, XOR, array multiplications, and division. This design selection key done within a four-bit size is named C_alu. In this four-bit size, there are 16 scenarios; it is designed as per operations mentioned in Table 2. In this ALU design architecture, there are three input sequences, i_datad, i_datas, and Immediate, each with a 64-bit size. Internal operations of addition and subtraction are also present with 64-bit input and 64-bit output, but the multiplication of n × n produces 2n output bits, as per this array multiplication, which generates a 128-bit output size. This multiplication output will be divided as MSB and LSB. The C_alu operation of "0111" decides the O_datad to generate the MSB of the multiplication, and O_datas generates the LSB of the multiplication.

Demonstrating the operations of ALU design, as per architecture given in Figure 11, the C_alu selection point guides two multiplexers and generates the outputs of o_datad and o_datas. The outputs are subsequently recorded with the system clock to get the final result. A comparative investigation of the proposed ALU design, employing both reversible and conventional full adder architectures.

This ALU design is the most prioritized design in the processor from the direct input, and it has priority in computational speed, power consumption, and overall efficiency. As the processor design, it also handles more complex tasks in digital gadgets, AI, signal processing, image procession, and so on. Similarly, this ALU must support all of these operations internally with high speed and lower power consumption. This proposed architecture of the ALU design will consume very less power, and it's very suitable for high speed and highly

Table 2: Instruction of ALU operation for reversible processor.

Operation	C_alu	Operation
ADD reg_d reg_s	0000	$R(reg_d) \leftarrow R(reg_d) + n\ R(reg_s)$
SUB reg_d reg_s	0001	$R(reg_d) \leftarrow R(reg_d) - n\ R(reg_s)$
ADD1 reg_d	0010	$R(reg_d) \leftarrow R(reg_d) + n\ 1$
SUB1 reg_d	0011	$R(reg_d) \leftarrow R(reg_d) - n\ 1$
NEG reg_d	0100	$R(reg_d) \leftarrow 0 - n\ R(reg_d)$
XOR reg_d reg_s	0101	$R(reg_d) \leftarrow R(reg_d) \oplus R(reg_s)$
XORI reg_d imm	0110	$R(reg_d) \leftarrow R(reg_d) \oplus imm$
MUL reg_d reg_s	0111	$R(reg_d) \leftarrow mul2n\ R(reg_d * reg_s)$
DIV2 reg_d	1000	$R(reg_d) \leftarrow div2n\ (R(reg_d))$

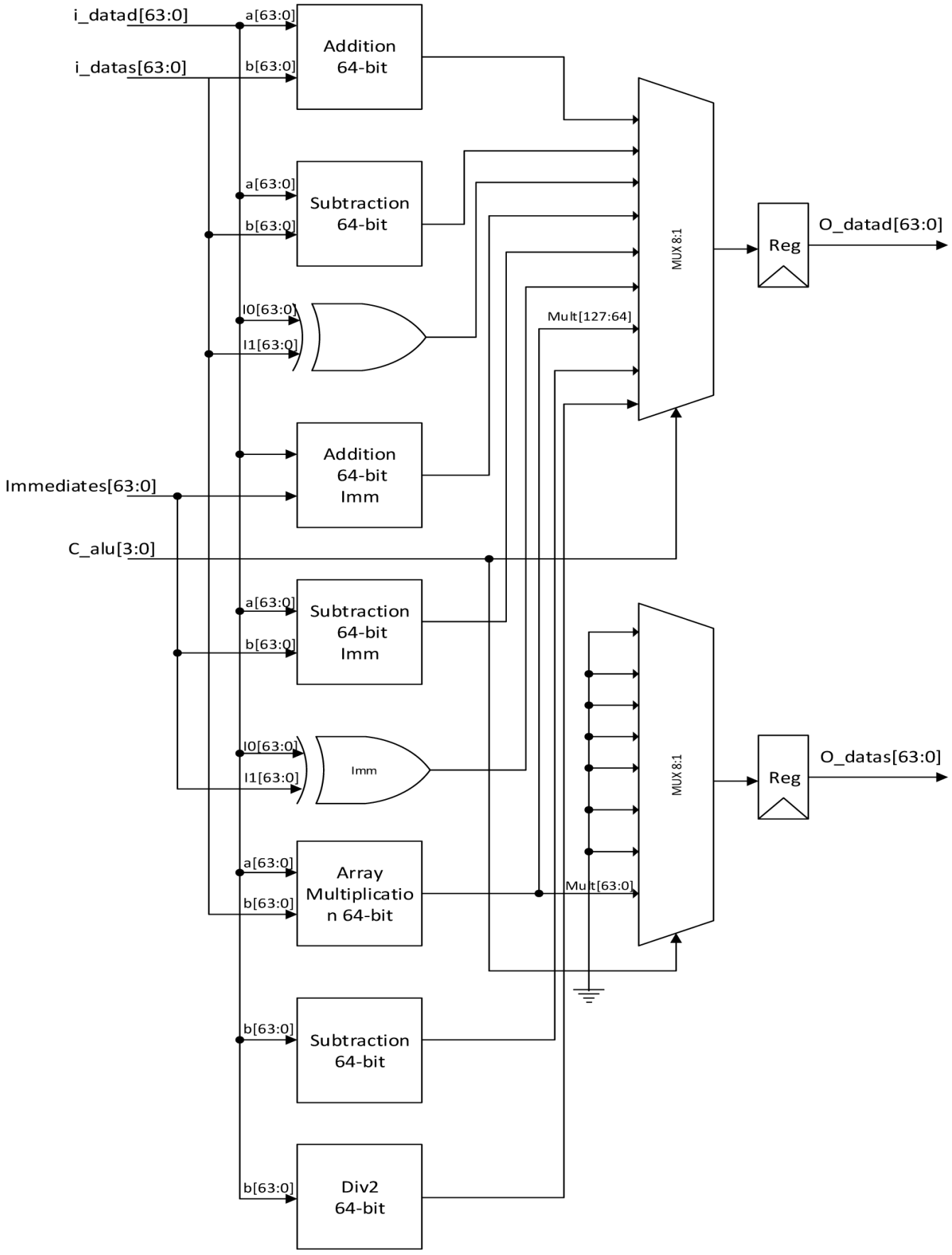


Fig. 11: Architecture of 64-bit ALU for reversible processor.

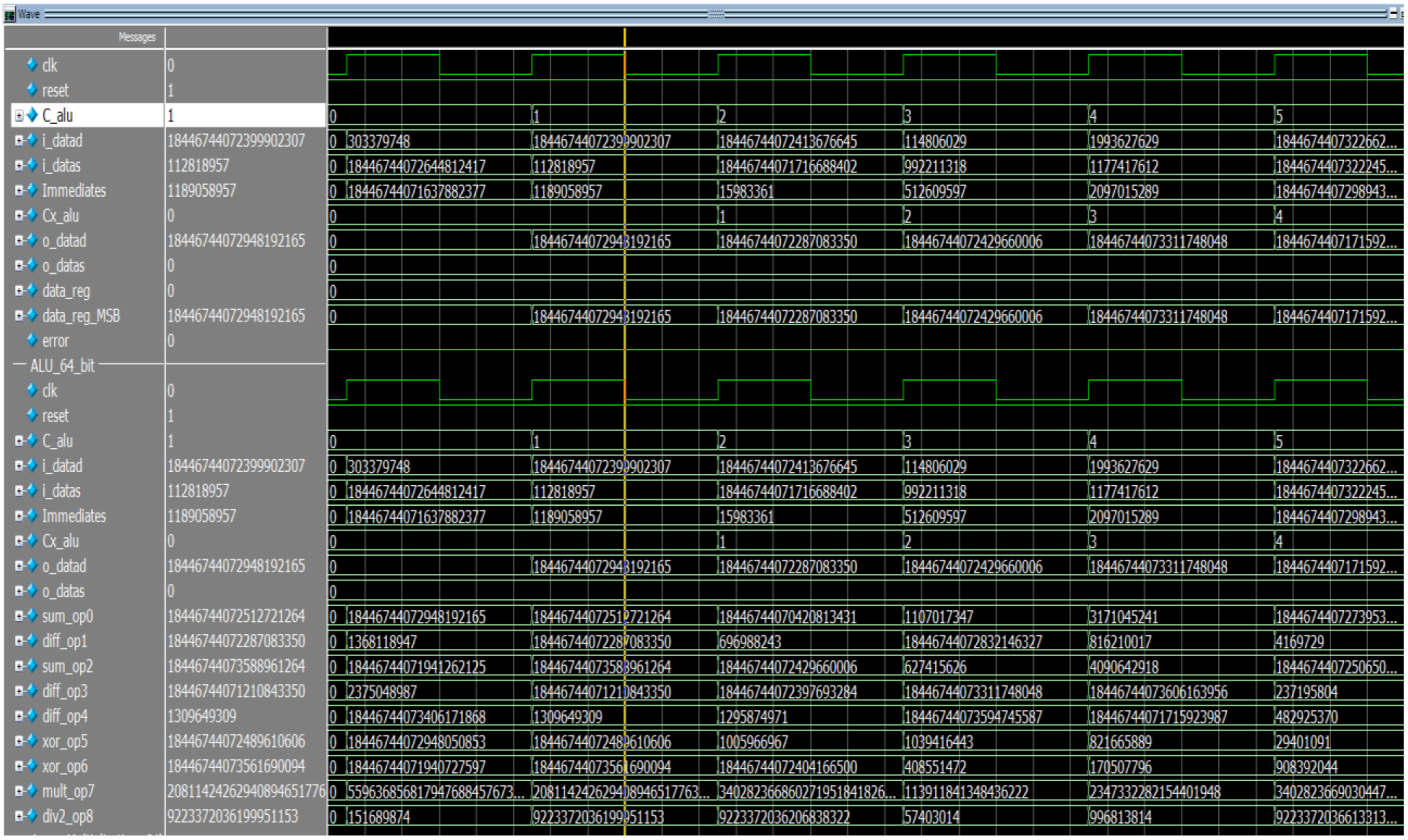


Fig. 12: Simulation results of 64-bit reversible ALU design.

Table 3: Comparative analysis of 64-bit ALU design with ECC using four different reversible and conventional majority full adder.

	LUT	FF	IO	BUFG	Delay (ns)	Power (W)
Conventional Majority FA[17]	7013	132	330	1	4.5	0.415
PERES Gate Full Adder[13]	16193	132	330	1	4.5	1.702
Fredkin Feynman Full Adder[14]	15493	132	330	1	4.5	0.947
HNG Gate Full Adder[14]	16193	132	330	1	4.5	0.94
Proposed without ECC	6485	132	330	1	4.5	0.401
Proposed with ECC	7100	132	330	1	5.2	0.43

reliable processor design. This novel architecture consumes only 0.401 W of power, 6485 in LUT, and 132 FFs. Compared to the traditional design architecture analysis, this proposed architecture will take lower utilization only.[1] The whole comparison analysis is shown in Table 3, the simulation results are shown in Figure 12, and the RTL analysis is done in Figure 13. These are generated using the Xilinx Vivado software. The results highlight the efficiency and efficacy of the proposed ALU architecture, demonstrating its potential for improved performance and diminished power usage in contemporary computing applications.

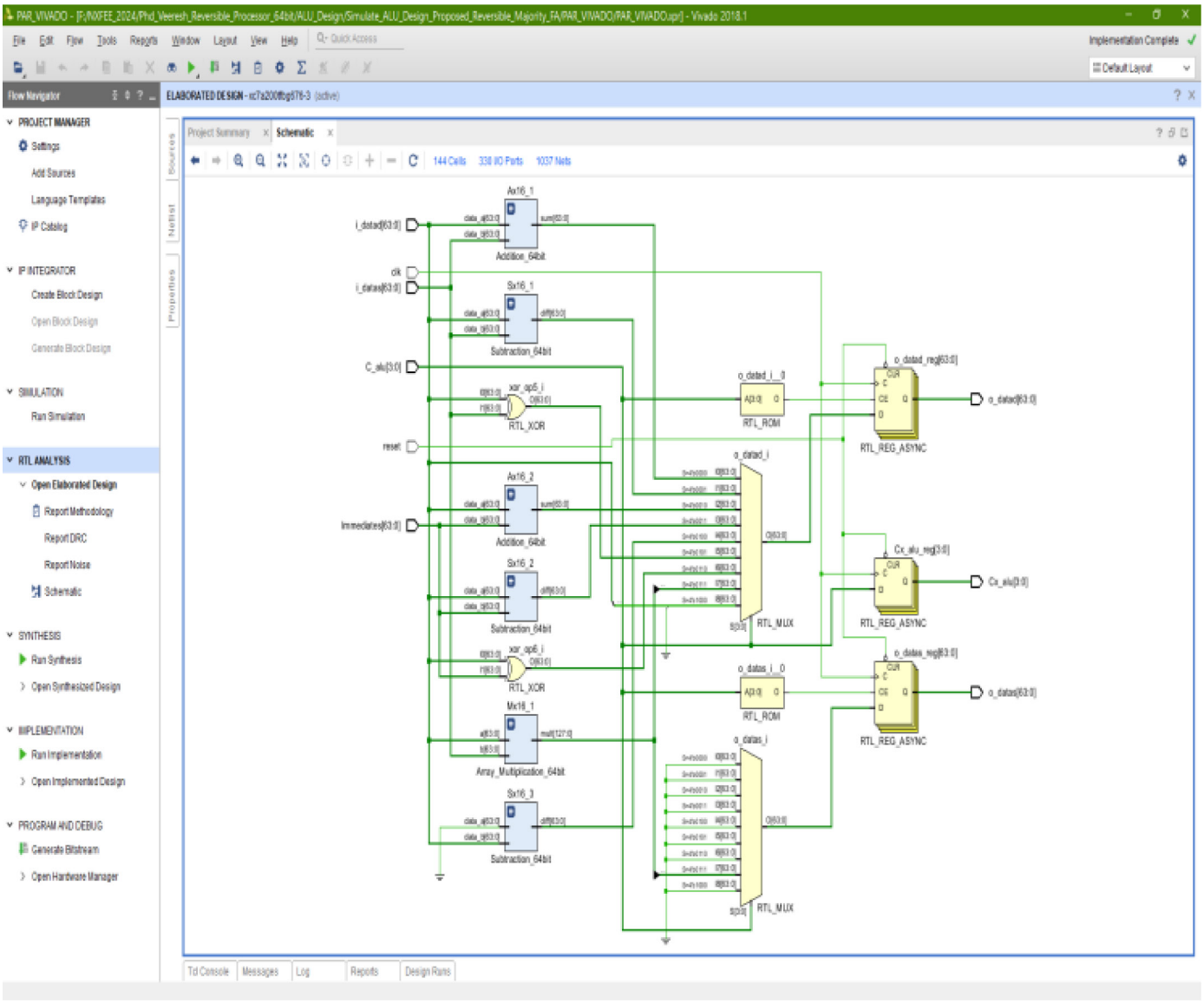


Fig. 13: RTL schematic of the proposed ALU design using reversible logic.

Conclusion

In conclusion, the proposed architecture of self-error detection and correction-based ALU design successfully achieves significant improvement of power efficiency, logic size utilization, and latency performance, especially for fault-tolerant arithmetic operations in processor design architecture. This integration was successfully proved with reversible majority gates, especially using the Feynman and Toffoli gate design, and has proven to effectively reduce circuit complexity, garbage output, and critical path delay reductions. By computing this proposed parity and Hamming architecture, the design ensures reliable detection and correction of both single- and multibit error corrections and greatly enhances the dependability of operations. The experimental results obtained from the Xilinx Artix-7 Vivado FPGA and their evaluation proved less power consumption of 52% and 7.5% by LUT compared to traditional design architecture. Further, the future design of this ALU is highly suitable for RISC processor design architecture. By utilizing the reversible ML gate in the core functional block of the ALU and RISC processor, it is especially suitable for mission-critical and low-power applications, such as AI, medical devices, aerospace applications, and communication systems, where dependable and self-correcting integration in RISC architecture is highly suitable for real-time error correction in a unified, FPGA-proven framework.

References

1. Priyanka, V., Narla, S. R., Gogineni, J., & Mohd. A. A. (July 2024). Design of Arithmetic Logic Unit Using Reversible Logic Gates, IEEE, 2024, 2nd World Conference on Communication & Computing (WCONF), 12-14.https://doi.org/10.1109/WCONF61366.2024.10692088
2. Shahram, B., Ali, S., & Ali, N. B. (June 2019). Design of an Efficient Multilayer Arithmetic Logic Unit in Quantum Dot Cellular Automata (QCA), IEEE Transaction in Circuit and System II: Express Briefs, Vol. 66, Issue 6.https://doi.org/10.1109/TCSII.2018.2873797
3. Sithara, R., Pranose, J. E., Nithin, Y. B. K., & Vasantha, M.H. (6th October 2020). An Approximate Low Power Lifting Scheme Using Reversible Logic, IEEE Access, Vol. 8.https://doi.org/10.1109/ACCESS.2020.3029149
4. Gao, Z., Reviriego, P., Pan, W., Xu, Z., Zhao, M., Wang, J., & Maestro, J. A. (2014). Fault Tolerant Parallel Filters Based on Error Correction Codes. IEEE Transactions on Very Large Scale Integration (VLSI) Systems. Retrieved from IEEE Xplore.https://doi.org/10.1109/TVLSI.2014.2308322
5. Gschwind, M., Salapura, V., & Maurer, D. (2001). FPGA Prototyping of a RISC Processor Core for Embedded Applications. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 9(2), 241-250.https://doi.org/10.1109/92.924027
6. Gaur, H. M., Singh, A. K., Mohan, A., Fujita, M., & Pradhan, D. K. (2020). Design of Single Bit Fault Tolerant Reversible Circuits. IEEE Design & Test.https://doi.org/10.1109/MDAT.2020.3006808
7. Morrison, M., & Ranganathan, N. (2011). Design of a Reversible ALU Based on Novel Programmable Reversible Logic Gate Structures. IEEE Computer Society Annual Symposium on VLSI.https://doi.org/10.1109/ISVLSI.2011.30
8. Shukla, S., & Ray, K. C. (2019). Design and ASIC Implementation of a Reconfigurable Fault-Tolerant ALU for Space Applications. IEEE International Symposium on Smart Electronic Systems (iSES).https://doi.org/10.1109/iSES47678.2019.00042
9. Yugandhar, K., Raja, V. G., Tejkumar, M., & Siva, D. (2018). High Performance Array Multiplier using Reversible Logic Structure. 2018 IEEE International Conference on Current Trends towards Converging Technologies.https://doi.org/10.1109/ICCTCT.2018.8550872
10. Liu, W., Zhang, T., McLarnon, E., O'Neill, M., Montuschi, P., & Lombardi, F. (2019). Design and Analysis of Majority Logic Based Approximate Adders and Multipliers. IEEE Transactions on Emerging Topics in Computing.https://doi.org/10.1109/TC.2018.2890222
11. Patel, R. B., Ho, J., Ferreyrol, F., Ralph, T. C., & Pryde, G. J. (2016). A quantum Fredkin gate. Science Advances, 2(3), e1501531.https://doi.org/10.1126/sciadv.1501531
12. Maslov, D. (2016). Advantages of using relative-phase Toffoli gates with an application to multiple control Toffoli optimization. Physical Review A, 93(2), 022311. https://doi.org/10.1103/PhysRevA.93.022311
13. Peres, A. (1985). Reversible logic and quantum computers. Physical Review A, 32(6), 3266-3276.https://doi.org/10.1103/PhysRevA.32.3266
14. Thapliyal, H., & Srinivas, M. B. (2006). Novel reversible TSG gate and its application for designing components of primitive reversible/quantum ALU. arXiv preprint arXiv:cs/0609023.
15. Sapatnekar, S. S., Gupta, S. K., & Raghunathan, A. (2018). Performance characterization and majority gate design for MESO Logic. IEEE Journal on Exploratory Solid-State Computational Devices and Circuits, 4, 29-37.https://doi.org/10.1109/JXCDC.2018.2874805
16. Islam, M. S., Hossain, M. A., & Hasan, M. M. (2022). Design a low-cost majority gate in QCA nanotechnology. IEEE Access, 10, 41791-41804.
17. Ottavi, M., Liu, M. L., & Lombardi, F. (2004). Circuit design based on majority gates for applications with quantum-dot cellular automata. IEEE Transactions on Nanotechnology, 3(1), 49-55.
18. Bhookya, R. V. P., & Ravindra, J. V. R. (2024). Design of high speed hybrid full adder using reversible logic gates. AIP Conference Proceedings, 2942(1), 020017.https://doi.org/10.1063/5.0196570
19. Sarvaghad-Moghaddam, M. & Orouji, A. A. (2019). New symmetric and planar designs of reversible full-adders/subtractors in quantum-dot cellular automata, The European Physical Journal D, 73(8), 1-12.https://doi.org/10.1140/epjd/e2019-90315-x
20. Weiqiang, L., Tingting, Z., Emma, M., Maire, O'Neill, Paolo, M., & Fabrizio, L. (July-Sept, 2021). Design and Analysis of Majority Logic Based Approximate Adders and Multiplier, IEEE Transactions on Emerging Topics in Computing, Vol. 9.https://doi.org/10.1109/TETC.2019.2929100
21. Mehdi, S., Reza, I., & Olli, S. (February 2022). Algorithm Level Error Detection in Low Voltage Systolic Array, IEEE Transactions on Circuit and System II: Express Briefs, Vol. 69.https://doi.org/10.1109/TCSII.2021.3094923
22. Zhijun, H., Ercegovac, M. D. (March 2005). High Performance Low Power left to right array multiplier design, IEEE Transactions on Computers, Issue 3.
23. Zhen, G., Pedro, R., Wen, P., Zhan, X., Ming, Z., Jing, W., & Juan, A. M. (February 2015). Fault Tolerant Parallel Filters Based on Error Correction Codes, IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 23, Issue 2.https://doi.org/10.1109/TVLSI.2014.2308322
24. Gschwind, M., Salapura, V., Maurer, D. (April, 2001). FPGA Prototyping of a RISC Processor core for embedded applications, IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 9, Issue 2.https://doi.org/10.1109/92.924027
25. Md., S. I., Md., H., & Md., S. A., Design and Synthesis of Reversible Arithmetic and Logic Unit (ALU), 2014 International Conference on Electrical Engineering and Information & Communication Technology, Dhaka, Bangladesh, 2014, pp.
26. Wilamowski, G. J. (2025). Embedded system architectures optimization for high-performance edge computing. SCCTS Journal of Embedded Systems Design and Applications, 2(2), 47-55.
27. Usikalu, M. R., Alabi, D., & Ezeh, G. N. (2025). Exploring emerging memory technologies in modern electronics. Progress in Electronics and Communication Engineering, 2(2), 31-40. https://doi.org/10.31838/PECE/02.02.04
28. Prasath, C. A. (2025). Adaptive filtering techniques for real-time audio signal enhancement in noisy environments. National Journal of Signal and Image Processing, 1(1), 26-33.