



International Journal of Artificial Intelligence and Machine Learning

Publisher's Home Page: <https://www.svedbergopen.com/>



Research Paper

Open Access

Fault-Tolerant Arithmetic Logic Unit Design Using Quantum Dot Cellular Automata for Reliable Nanoelectronics Systems

Smita C. Chetti^{1*}, Veena M. B.²

^{1*}Department of Electronics and Communication Engineering, BMS College of Engineering, Bengaluru, Visvesvaraya Technological University, Belagavi, India. ORCID: 0000-0002-4142-2539, E-mail: smita.chetti@gmail.com.

²Department of Electronics and Communication Engineering, BMS College of Engineering, Bengaluru, Visvesvaraya Technological University, Belagavi, India. ORCID: 0000-0001-9171-2625, E-mail: veenamb.ece@bmsce.ac.in.

ABSTRACT

Quantum-Dot Cellular Automata (QCA) offer a promising alternative to conventional Complementary Metal-Oxide-Semiconductor (CMOS) technology for compact and energy-efficient nanoelectronics circuit design. However, previous QCA-based Arithmetic Logic Unit (ALU) designs have mainly emphasized optimization, reversible computation, or individual fault-tolerant circuits, with limitations in simultaneously achieving compactness, low delay, and reliable operation under multiple fault conditions. This study aimed to develop a fault-tolerant QCA-based ALU addressing these limitations. The proposed methodology involved designing a majority gate, full adder, 4:1 multiplexer, logic unit, and arithmetic unit, followed by their integration into a complete ALU. Functional verification, fault injection analysis, performance evaluation, and statistical validation were subsequently performed using QCADesigner. The proposed ALU achieved 206 cells, an occupied area of $0.455 \mu\text{m}^2$, a 0.5 clock-cycle delay, and 0.0730 eV energy dissipation, with an overall 94.25% fault reliability. The comparative analysis demonstrated improved compactness and competitive delay relative to the considered existing designs. The study concludes that the proposed architecture provides a reliable and compact solution for QCA-based nanoelectronics. Overall, the results demonstrate the feasibility of fault-tolerant QCA-based ALU design for emerging reliable and scalable nanoelectronic systems, supporting their potential integration into future high-density computational architectures.

Keywords: Arithmetic Logic Unit, Fault-Tolerant Design, Nanoelectronics, Quantum-Dot Cellular Automata, QCADesigner.

1. Introduction

The continual scaling of CMOS technology has resulted in tremendous improvements in the performance of digital circuits over the last few decades [1-3]. Nevertheless, at the nanoscale, the performance of CMOS is limited due to factors such as leakage current, excessive power consumption, short-channel effect, and difficulty in manufacturing [4-6]. Such difficulties have led to increased research into post-CMOS computing architectures that could enable future developments in nanoelectronics [7-9]. There are several novel technologies that have drawn much interest. Still, among them, the most notable one is the QCA due to its non-transistor structure, low power dissipation, high density, and fast data processing [10-12]. Rather than depending on electric currents, QCA technology computes by manipulating electron spin polarization, thus making it a potential technology for nanocircuits [13-16].

ALUs are some of the most essential computation units for digital devices, which can perform several arithmetic and logic operations that are needed by processors and other embedded systems [17-19]. As a result, various QCA ALUs have been suggested to minimize hardware complexity, area usage, delay, and power dissipation [20-22]. In spite of all these developments, realization of QCA circuits is still difficult due to some imperfections during the manufacturing process [23-25]. Cell omission, cell displacement, cell misalignment, cell overdeposition, and cell rotation are some of the common faults that exist in QCA structures and can be seen in Figure 1. These faults cause errors in cell polarization, affect signal transmission and result in incorrect output from the circuits. This means that building a fault-tolerant ALU structure in QCA has become a significant research area [26-29].

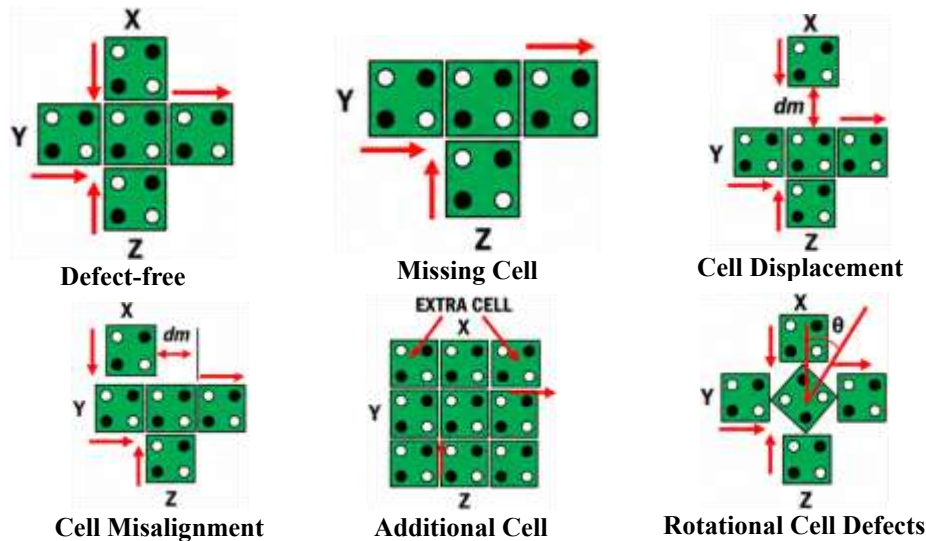


Figure 1: Types of manufacturing defects considered in the proposed QCA-based ALU [30].

Despite the recent advancements in QCA-based ALUs and associated digital circuits, the majority of existing designs mainly emphasize improvement in performance-related parameters, including number of cells, area occupied, latency time, and energy consumption [31,32]. Relatively little effort has been put into improving fault tolerance for several manufacturing defects without affecting circuit efficiency. Consequently, the majority of existing ALUs face poor performance and reliability in the presence of physical faults, thus making them impractical in reliable nanoelectronic devices [33,34]. Moreover, integrating fault tolerance methods with efficient circuit design poses a huge challenge because of the reliability versus design complexity tradeoff. Therefore, there is a need for a robust QCA-based ALU that can effectively tolerate manufacturing defects while maintaining high computational efficiency and compact circuit implementation [35,36].

In order to overcome the mentioned challenges, this study aims to present an error-tolerant ALU design utilizing QCA technology. The methodology involved in this study involves designing an error-tolerant majority gate and then designing an adder, 4:1 multiplexer, and respective logic and arithmetic blocks. All of these components are used to design the presented ALU and are implemented using QCA physical design and simulated in QCADesigner. Finally, the proposed design is verified and validated through functional verification, fault injection tests, performance evaluation, and statistical validation.

The novelty of the proposed research is in the development of the complete fault-tolerant QCA-based ALU using fault-tolerant circuits for the design process. The difference from many research works, where the main focus was on hardware efficiency and optimizations, is that the proposed research involves systematic circuit design, functional verification, fault injection simulation, performance evaluation, and statistical validation under one design approach. This allows the comprehensive evaluation of the proposed ALU in terms of its reliability, hardware efficiency, and performance.

The main goal of this study is to design an efficient, fault-tolerant ALU based on QCA. This research focuses on achieving better reliability, reduced hardware complexity, lower latency and power consumption, and increased robustness with respect to fabrication errors. The significant contributions made by this research are as follows:

- Proposed a fault-tolerant QCA-based ALU through the design and integration of a majority gate, full adder, 4:1 multiplexer, logic unit, and arithmetic unit.
- Implemented the proposed ALU using a QCA physical layout and simulated its functionality in QCADesigner.
- Validated the proposed design through functional verification and fault injection analysis under multiple manufacturing defect scenarios.
- Established a systematic fault analysis framework to investigate the robustness of the anticipated ALU under multiple manufacturing defects and validate its operational stability.

The remainder of this research is structured as follows. Section 2 presents the literature review, Section 3 describes the recommended methodology, Section 4 discusses the results and analysis, and Section 5 concludes the study and presents future scope.

2. Literature Review

This section discusses a review of current studies on QCA-based digital circuit design and related architecture.

2.1 Fundamentals of Quantum-dot Cellular Automata

QCA is a new nanotechnology technique that has been suggested as one of the potential alternatives to existing CMOS technology. QCA technology differs from CMOS technology by representing binary data through electron placement in the quantum-dot cells instead of representing the data through current flow, thus resulting in low power consumption, higher density of devices, fast switching, and smaller circuit area. The simplest QCA cell comprises four quantum dots and two movable electrons, as illustrated in Figure 2. Because of the Coulombic repulsion, the two electrons would be at the diagonally opposite corners of the cell, giving rise to two stable polarization states; the value of polarization would be +1 for binary 1 and -1 for binary 0. Data would be transferred through Coulombic interactions between neighboring cells, thus making it possible to create gates, arithmetic circuits, and ALUs through QCA technology. There have been several proposals of QCA circuits in the literature, and some of the recent contributions to this field are discussed below.

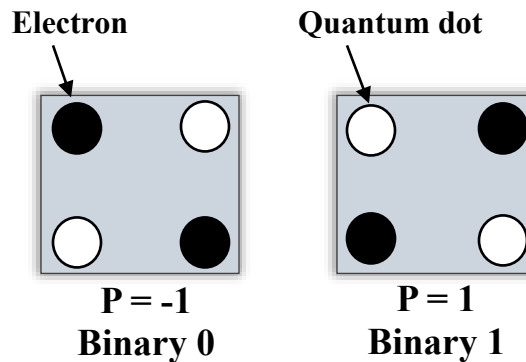


Figure 2: Basic structure of a QCA cell [37]

Based on these fundamental concepts, several QCA-based circuit designs have been suggested in the literature. Some of the recent research contributions are discussed below.

Premananda et al. (2026) [38] presented a Static Random Access Memory (SRAM) architecture based on QCA using QCADesigner and QCADesigner-E for efficient Internet of Things (IoT) memory. The architecture showed a reduction in cell number by 48.19%, Area by 50.00%, clocking by 20.00%, and an improvement in timing by 30.00%. Similarly, a low-cost QCA-based approximate multiplier based on majority logic was presented by **Fayazi & Abdoli (2026) [39]**. This design had a 69% reduction in the number of cells, an 86.7% reduction in Area, and more than a 79% reduction in energy dissipation compared to existing QCA-based multipliers.

Additionally, **Javaran et al. (2025) [40]** designed a fault-tolerant QCA-based 7-input majority gate with the help of QCADesigner and QCAPro, with which they were able to achieve a fault-tolerant design with 66 cells, $0.03 \mu\text{m}^2$ area, 0.25-clock delay, and 64.5% average fault tolerance. Similarly, **Mohammad Abdullah-Al-Shaf (2025) [41]** proposed the QCA-based architecture of compact Random Access Memory (RAM), Demultiplexer (DEMUX), and ALU using simulations in QCADesigner 2.0.3. The proposed designs provided a 67% reduction in the number of RAM cells, a 66% reduction in Area, a 92% reduction in quantum cost, and a 20% reduction in latency, thereby making them more efficient. In another study, **Seyedi and Abdoli (2025) [42]** designed a QCA-based Fault-Tolerant Carry Skip Adder (CSA) design for IoT applications through simulation using QCADesigner-E. Their design had 85% fault tolerance, took up $4.59 \mu\text{m}^2$ of area, used 1542 cells and needed 4.75 clock phases. Additionally, **Zohaib et al. (2025) [37]** implemented a QCA-based ALU using a coplanar non-rotated architecture in QCADesigner to reduce Area and energy consumption. The design achieved 42.48% lower cell count and 64.28% smaller occupied Area than previous designs.

Earlier, **Kalyan and Singh (2024) [30]** presented a QCA-based fault-tolerant Linear Feedback Shift Register (LFSR) using a novel five-input majority gate, XOR gate, and D flip-flop simulated in QCA Designer. The proposed LFSR achieved 435 QCA cells, $0.53 \mu\text{m}^2$ area, 1-clock latency, and 0.12 eV energy dissipation, demonstrating improved fault tolerance. Likewise, **Alharbi et al. (2024) [43]** developed a QCA hybrid half-adder using reversible, irreversible, and partially reversible majority gates with QCADesigner-E 2.2 to optimize power, delay, and Area. The design achieved 90 QCA cells, $0.12 \mu\text{m}^2$ area, 1 clock-cycle latency, and 60% speed improvement. Moreover, **Kassa et al. (2024) [44]** introduced a QCA-based cryptographic nanorouter using a majority-gate DEMUX architecture implemented in QCADesigner to enhance secure nano-communication. The design achieved 8–10% fewer cells, 50–60% lower latency, and 3–5% smaller Area.

Additionally, **Alharbi et al. (2023) [45]** designed a QCA-based reversible ALU using reversible majority gates with QCADesigner-E simulation to improve energy efficiency. The design achieved 88.8% higher energy efficiency, 51% fewer QCA cells, and 47% less Area. Finally, **N and MB (2022) [46]** developed an ALU functional block using QCA and QCADesigner 2.0.3. The design achieved 35.40% lower cell count, 15.88% smaller Area, and 48.21% reduced clock delay.

3. Proposed Methodology

This section describes the software tools used for circuit simulation and statistical analysis, followed by the research methodology adopted for the design, simulation, and evaluation of the suggested fault-tolerant ALU based on QCA.

3.1 Tools Used

The recommended research makes use of QCADesigner 2.0.3 [47] and STATA [48] to facilitate the development, simulation, and validation of the fault-tolerant ALU. QCADesigner 2.0.3 is utilized in developing the QCA layout for the majority gate, full adder, multiplexer, logic unit, and arithmetic unit, as well as performing simulation studies to verify the effectiveness of the design under different conditions. The output generated from the simulation studies done using QCADesigner is then analyzed using STATA, where the software tool is utilized to analyze the simulation results using descriptive statistical analysis, Analysis of Variance (ANOVA), sensitivity analysis, confidence interval analysis, and Wilcoxon Signed-Rank Test.

3.2 Research Design

The suggested methodology provides a systematic framework for designing and evaluating the fault-tolerant QCA-based ALU. It includes circuit design, functional simulation, fault analysis, performance evaluation, and statistical validation to assess the functionality, reliability, and efficiency of the suggested design. The overall workflow of the suggested methodology is illustrated in Figure 3.

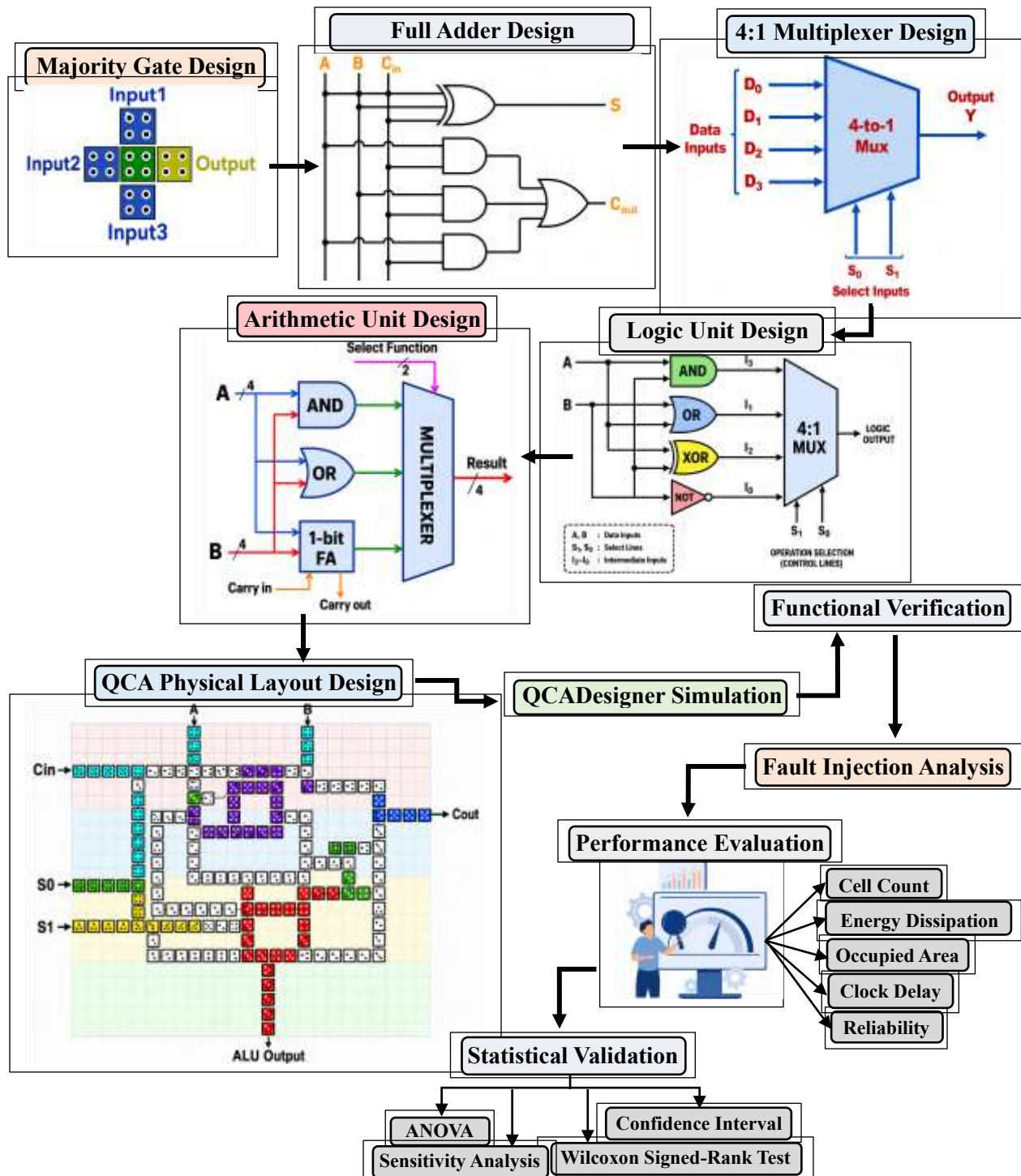


Figure 3: Proposed methodology

3.2.1 Majority Gate Design

The methodology that is proposed starts with designing a fault-tolerant majority gate that has three inputs. It is the basic structure of the proposed QCA-based ALU. The QCA cells have been placed in such a way that they can perform the majority voting function while transferring polarity in a stable manner among the adjacent cells. The placement of cells has been done in an optimal way in order to make it more tolerant against various faults, such as missing, displacement, rotation, and extra cell deposition. The logical function of the majority gate is written as:

$$M(A, B, C) = AB + AC + BC \quad (1)$$

Where A , B , and C represent the three binary input signals, $M(A, B, C)$ denotes the majority gate output, and AB , AC , and BC represent the pairwise logical AND operations whose logical OR produces the majority output.

The resulting gate provides reliable majority logic operation under defect conditions and serves as the foundation of the proposed fault-tolerant QCA circuit design.

3.2.2 Full Adder Design

Building upon the suggested fault-tolerant majority gate, a fault-tolerant full adder is designed to perform single-bit binary addition in the suggested QCA -based ALU. The full adder receives three binary inputs, namely A , B , and the carry-in (C_{in}), and generates the Sum (S) and Carry-out (C_{out}) outputs. The QCA cells are arranged to ensure reliable signal propagation and stable arithmetic operation while improving tolerance against manufacturing defects. The logical functions of the full adder are expressed as:

$$S = A \oplus B \oplus C_{in} \quad (2)$$

$$C_{out} = AB + AC_{in} + BC_{in} \quad (3)$$

Where A and B represent the binary input bits, C_{in} denotes the input carry, S is the sum output, and C_{out} represents the carry output. The suggested fault-tolerant full adder provides reliable binary addition while maintaining correct functionality under common QCA manufacturing defect conditions.

3.2.3 Design of 4:1 Multiplexer

Based on the discussed fault-tolerant full adder circuit, a fault-tolerant 4:1 multiplexer has been developed to reliably select the correct one among several outputs of an arithmetic logic function performed by the suggested QCA-based ALU. To achieve the correct operation of the multiplexer, QCA cells are connected in such a way that it ensure stable propagation of signals while reducing the influence of manufacturing faults. At the stage of operation, the proposed multiplexer accepts four input signals and two selection lines that ensure that the desired signal can propagate to the output, whereas other inputs are blocked.

The functional structure of the suggested 4:1 multiplexer is illustrated in Figure 4, showing the four data inputs (I_0, I_1, I_2 and I_3), the two selection inputs (S_1 and S_0), and the selected output (Y).

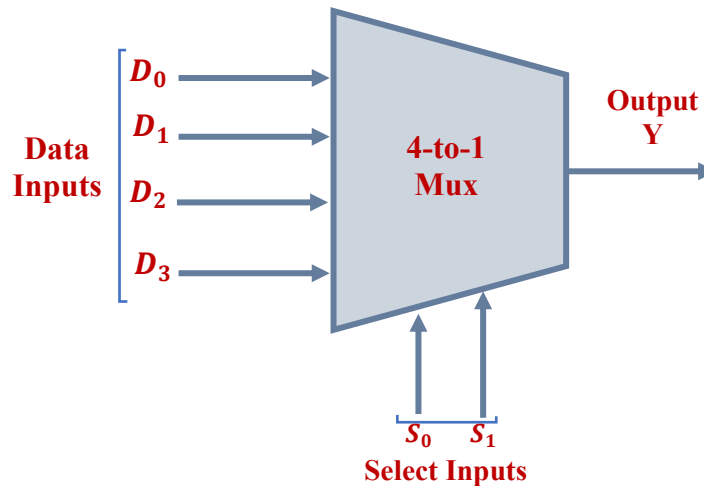


Figure 4: Functional block diagram of the 4:1 multiplexer [Source: Author's own]

The logical function of the 4:1 multiplexer is expressed as

$$Y = \bar{S}_1 \bar{S}_0 I_0 + \bar{S}_1 S_0 I_1 + S_1 \bar{S}_0 I_2 + S_1 S_0 I_3 \quad (4)$$

Where I_0, I_1, I_2 , and I_3 represent the four data inputs, S_1 and S_0 denote the selection inputs, and Y represents the selected output. The fault-tolerant QCA cell arrangement enhances reliable data selection while maintaining correct functionality under common manufacturing defect conditions.

3.2.4 Logic Unit Design

Based on the design of the fault-tolerant 4:1 multiplexer, the logic unit is designed to carry out logical operations in the suggested QCA-based ALU. The logic unit has been designed using QCA-based logic gates to carry out various logical operations such as AND, OR, XOR, and NOT. The logical operation to be carried out is chosen through the control signals, and the output is produced by the circuit while ensuring stable propagation of signals. The arrangement of QCA cells ensures greater tolerance to any manufacturing errors that might occur during the process. The output of the logic unit is represented by:

$$L = f(A, B, S) \quad (5)$$

Where A and B denote the binary input signals, S represents the control or selection signal that determines the required logical operation, and L is the logic unit output. The proposed fault-tolerant design ensures accurate

execution of logical operations while improving circuit reliability under common QCA manufacturing defect conditions.

3.2.5 Arithmetic Unit Design

Based on the fault-tolerant logic unit developed, the fault-tolerant arithmetic unit can be created to carry out basic arithmetic calculations in the proposed QCA-based ALU. The arithmetic unit can be designed using the integration of the fault-tolerant full adder and logic unit through the multiplexer structure to perform both arithmetic and logical functions under the control of the selection signals. The QCA cells are positioned in such a way that there is stable polarization transfer and signal transmission with little effect from any production defects.

The functional architecture of the anticipated arithmetic unit is illustrated in Figure 5. As shown in the figure, the arithmetic circuit, logic circuit, and multiplexer are integrated into a unified architecture. The binary inputs (A and B) and the carry input (C_{in}) are processed by the arithmetic and logic circuits, while the selection function controls the multiplexer to choose the required operation and generate the final ALU output.

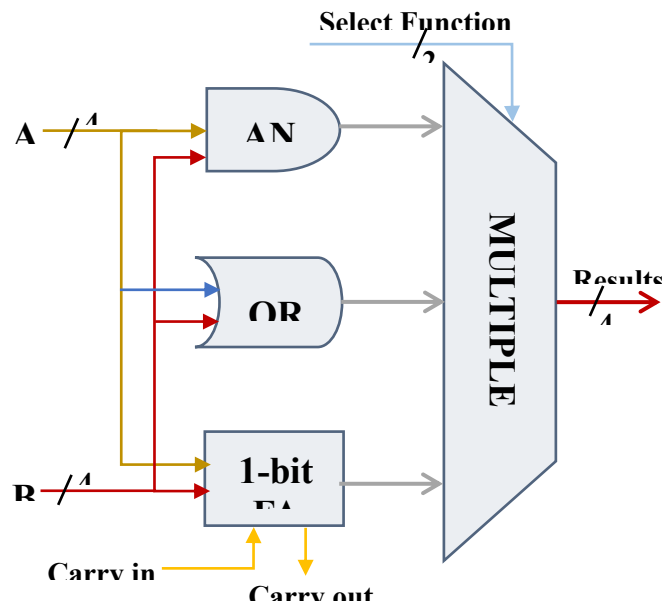


Figure 5: Functional architecture of the ALU [Source: Author's own]

The arithmetic output is represented as

$$A_{out} = f(A, B, C_{in}, S) \quad (6)$$

Where A and B represent the binary input operands, C_{in} denotes the carry input, S represents the control signal that selects the required arithmetic or logical operation, and A_{out} represents the final output of the integrated ALU architecture. The suggested fault-tolerant design enables reliable execution of multiple operations while maintaining correct functionality under common QCA manufacturing defect conditions.

3.2.6 QCA Physical Layout Design

Based on the design of the suggested fault-tolerant ALU, a physical design of QCA is carried out to implement the design at the nanometer scale. The physical design is carried out through the placement of QCA cells, majority gates, interconnects, and clock zones according to the functionality required of the ALU and ensuring proper signal propagation through the circuit. Placement of the cells in the layout is done in such a way as to minimize the area occupied, minimize signal delay, and increase the robustness of the circuit to any manufacturing defects. Special care is taken to ensure proper interaction of cells and clocking to ensure proper polarization propagation and logic execution.

3.2.7 QCADesigner Simulation

Based on the QCA physical design, the fault-tolerant ALU is simulated through the use of QCADesigner, and its operation is evaluated through the study of its behavior. This simulation process entails the setting of various QCA engine parameters like cell dimension, clocking strategy, radius of effect, tolerance, and simulation algorithm, which help in accurately simulating the polarization behavior of QCA cells. During the simulation process, various binary inputs are supplied to the circuit, and the resulting output polarization states are observed to check how signals are propagating within the circuit.

3.2.8 Functional Verification

Based on the simulation outcomes, functional verification is done to make sure that the fault-tolerant ALU design functions based on the intended logic and arithmetic operations. The different input values are provided to the circuit, and the output values generated are compared with the respective outputs based on the ALU truth table. In the process of verification, it is ensured that each arithmetic and logical operation generates the right output while sustaining signal stability within the QCA circuit. The comparison of the outputs generated during the simulation process and the expected outputs helps in validating the functionality of the suggested design.

3.2.9 Fault Injection Analysis

The fault injection analysis is conducted on the basis of the verified functionality of the suggested ALU for assessing its reliability under defect conditions. Defect conditions, such as omission of a cell, displacement of a cell, rotation of a cell, and the addition of a cell, are deliberately applied to the QCA layout in order to study their effect on the operation of the circuit. The effect of a defect condition on the circuit operation is assessed through the comparison of the generated output results with the expected ones in order to check whether the suggested ALU functions properly.

3.2.10 Performance Evaluation

The performance of the suggested fault-tolerant ALU is evaluated using key design metrics, including cell count, occupied Area, clock delay, energy dissipation, and reliability. These metrics provide a comprehensive assessment of the efficiency and reliability of the suggested QCA-based ALU.

The cell count is determined to measure the hardware complexity of the suggested QCA-based ALU and is expressed as:

$$N_{cell} = N \quad (7)$$

Where N_{cell} is the total number of QCA cells used in the proposed ALU layout, and N represents the total number of cells employed in the design.

The occupied Area is calculated to evaluate the physical layout size required for implementing the proposed ALU and is given by:

$$A = L \times W \quad (8)$$

Where A denotes the occupied layout area, L is the layout length, and W is the layout width.

The clock delay is computed to determine the time required for signal propagation through the proposed ALU and is expressed as:

$$D = N_{clk} \times T_{clk} \quad (9)$$

Where D represents the total clock delay, N_{clk} is the number of QCA clock phases required for signal propagation, and T_{clk} denotes the duration of one QCA clock cycle.

The energy dissipation is estimated to evaluate the energy consumed during the operation of the proposed ALU and is calculated as:

$$E = P \times T \quad (10)$$

Where E represents the energy dissipation, P denotes the average power consumption, and T is the circuit operating time.

The reliability is evaluated to quantify the ability of the proposed ALU to produce correct outputs under normal and fault conditions and is computed as:

$$R = \frac{N_c}{N_t} \times 100 \quad (11)$$

Where R is the reliability expressed as a percentage, N_c denotes the number of successful output responses under all evaluated conditions, and N_t represents the total number of test cases or fault scenarios considered.

3.2.11 Statistical Validation

Statistical validation is performed to assess the significance, consistency, and reliability of the performance results of the recommended fault-tolerant ALU.

The ANOVA is performed to determine whether there are statistically significant differences among the performance metrics of the proposed and existing QCA-based ALU designs. The ANOVA F-statistic is calculated as:

$$F = \frac{MS_B}{MS_W} \quad (12)$$

Where F is the ANOVA test statistic, MS_B symbolizes the mean square between groups, and MS_W indicates the mean square within groups.

The Sensitivity Analysis is carried out to quantify the influence of manufacturing defect variations on the performance of the proposed ALU and is expressed as:

$$S = \frac{\Delta O}{\Delta I} \quad (13)$$

Where S denotes the sensitivity, ΔO represents the change in the output performance metric, and ΔI represents the change in the input parameter or defect condition.

The Confidence Interval (CI) is computed to estimate the range within which the true mean of the performance metric is expected to lie and is given by:

$$CI = \bar{x} \pm z \left(\frac{\sigma}{\sqrt{n}} \right) \quad (14)$$

Where CI denotes the confidence interval, $\bar{x}$ is the sample mean, z is the standard normal critical value corresponding to the selected confidence level, σ stands for the standard deviation, and n is the sample size. The Wilcoxon Signed-Rank Test is employed to determine whether the proposed ALU exhibits statistically significant performance improvements over existing QCA-based ALU designs. The test statistic is computed as:

$$W = \sum R_i \quad (15)$$

Where W is the Wilcoxon signed-rank statistic and R_i denotes the signed rank associated with the i^{th} paired observation.

3.2.12 Pseudo code

Algorithm 1: Proposed Fault-Tolerant QCA-Based ALU Design

Input:

- Binary operands A and B
- Carry input Cin
- Operation select signals S0 and S1

Output:

- Fault-tolerant ALU output Y

Begin

Step 1:

- MajorityGate $\leftarrow$ Design a fault-tolerant three-input majority gate.

Step 2:

- FullAdder $\leftarrow$ Construct a fault-tolerant full adder using the majority gate.

Step 3:

- Multiplexer $\leftarrow$ Design a fault-tolerant 4:1 multiplexer for output selection.

Step 4:

- Develop the logic unit to perform
 - AND,
 - OR,
 - XOR, and
 - NOT operations.

Step 5:

- ArithmeticUnit $\leftarrow$ Develop the arithmetic unit using the fault-tolerant full adder and multiplexer.

Step 6:

- ALU $\leftarrow$ Integrate the arithmetic unit and logic unit.

Step 7:

- Design the physical QCA layout of the proposed ALU.

Step 8:

- Simulate the QCA layout using QCADesigner.

Step 9:

- Apply all input combinations.
- Verify that the simulated outputs match the expected outputs.

Step 10:

- FOR each fault type $\in \{\text{Cell Omission, Cell Displacement, Cell Rotation, Additional Cell Deposition}\}$ DO
 - Inject the fault into the QCA layout.
 - Evaluate the circuit functionality.
- END FOR

Step 11:

- Measure the performance metrics including
 - Cell Count,
 - Occupied Area,
 - Clock Delay,
 - Energy Dissipation, and
 - Reliability.

Step 12:

Perform statistical validation using
ANOVA,
Sensitivity Analysis,
Confidence Interval, and
Wilcoxon Signed-Rank Test.

$Y \leftarrow$ Verified fault-tolerant ALU output

Return Y

End

4. Results Analysis

This section illustrates the simulation and performance analysis of the suggested fault-tolerant QCA-based ALU. The study includes the functional analysis of the majority gate, full adder, 4:1 multiplexer, logic unit, arithmetic unit, and the complete ALU, along with the fault injection analysis, performance analysis, and statistical analysis. The obtained results are analyzed to check the functional and reliability characteristics of the suggested design at the given operational conditions. The configuration of the system used for designing, simulating, and analyzing the suggested design is shown in Table 1.

Table 1: System configuration

Category	Component / Software	Specification / Version
Hardware	System	Lenovo 90X0004PIN, x64 PC
	CPU	Intel Core i5-14400F
	Cores / Threads	10 cores / 16 logical processors
	RAM	~16 GB
	GPU	NVIDIA GeForce RTX 3050, ~4 GB VRAM
	Storage	C: 1 TB
Software	Ubuntu	22.04 LTS
	QCADesigner	2.0.3

The suggested design approach started off with designing a fault-tolerant majority gate, which was the first circuit element used in the evaluation. Figure 6 illustrates the QCA layout of the suggested majority gate that consists of three signals A, B, and C, and the output M_{out} . The QCA cells were designed in such a way as to provide polarization-based signal transmission for the majority voting function. The final layout provides an implementation of the majority function, $M(A, B, C) = AB + BC + AC$, where the output of the gate depends on the majority state of the three input signals. The obtained layout demonstrates the physical realization of the suggested fault-tolerant majority-gate structure and its ability to establish the intended majority logic operation.

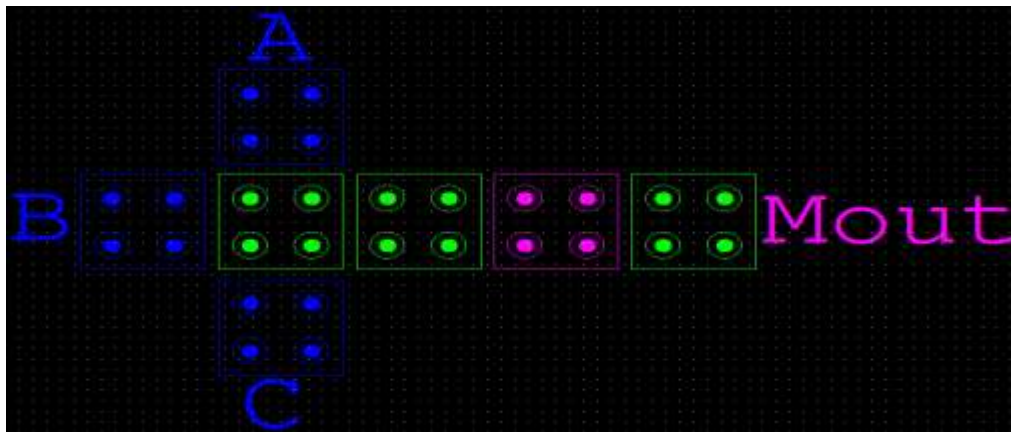


Figure 6: QCA majority gate layout.

By exploiting the physical implementation of the designed majority gate, the functionality of the designed majority gate was tested using the QCADesigner simulator. Figure 7 shows the input/output waveforms of the majority gate in terms of the simulation period taken into account. As can be seen from the figure, the output

M_{out} changes between about -0.984 and +0.984, which corresponds to the two polarization states representing logic 0 and logic 1. It is clear that the output changes based on the majority rule of inputs A, B, and C.

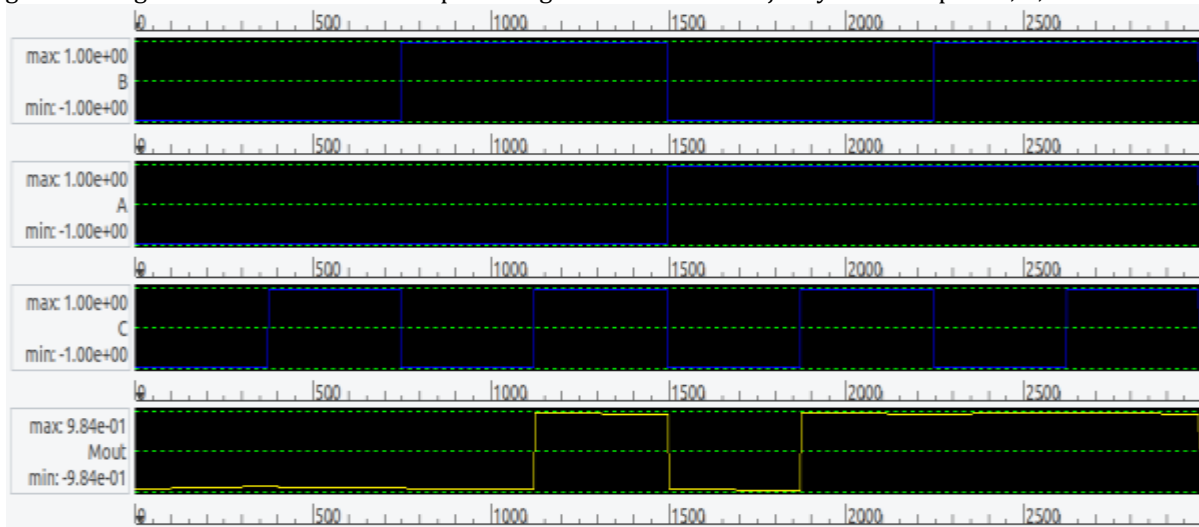


Figure 7: Majority gate simulation waveform.

Based on the design of the majority gate, a fault-tolerant full adder has been implemented for carrying out the binary addition based on three inputs, A , B , and C_{in} . Figure 8 illustrates the QCA design of the implemented full adder, including the three input signals and two output signals, SUM and C_{out} . The QCA cells have been organized in an effective way for the propagation of polarization of the input signals as well as for realizing the necessary Sum and carry functions. The SUM output indicates the result of the binary addition, while C_{out} is the carry generated due to the three input signals. The resulting QCA structure provides the required full-adder functionality while maintaining the fault-tolerant design characteristics considered in the suggested circuit.



Figure 8: QCA full adder layout.

The functional behavior of the designed full adder was determined using QCADesigner simulation by applying various combinations of the input signals A , B , and C_{in} . Figure 9 depicts the waveforms of the three input signals as well as their outputs SUM and C_{out} . The SUM signal depends on the parity of the three inputs, whereas C_{out} is activated when the majority of the inputs is 1. The polarizations of the simulated SUM signal were found to be +0.954 and -0.954 for logic 1 and 0, respectively. The same holds for the polarizations of the C_{out} signal, which are equal to +0.980 and -0.980 for logic 1 and 0, respectively. The output transitions correspond correctly to the applied input combinations, confirming the functional operation of the suggested full-adder circuit.

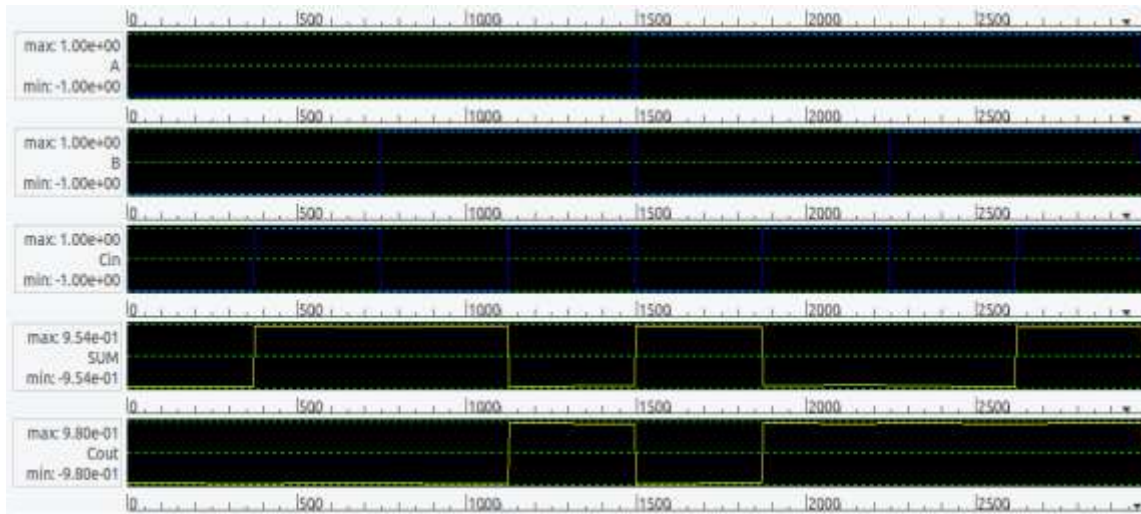


Figure 9: Full adder simulation waveform.

The suggested fault-tolerant 4:1 MUX was designed following the structure of the full adder circuit to enable selection from among four data inputs. Figure 10 shows the QCA layout of the multiplexer with four data inputs I_0 , I_1 , I_2 , and I_3 , along with two selection inputs S_0 and S_1 . This layout is based on the selection signals $M01$ and $M23$, representing the selection paths of their corresponding data inputs, as well as the final selected output represented by Y_{MUX4} . The QCA cells were configured in order to carry out the necessary selection operation through polarization-based signal propagation. The resulting layout establishes the physical realization of the 4:1 multiplexer within the suggested fault-tolerant QCA architecture.

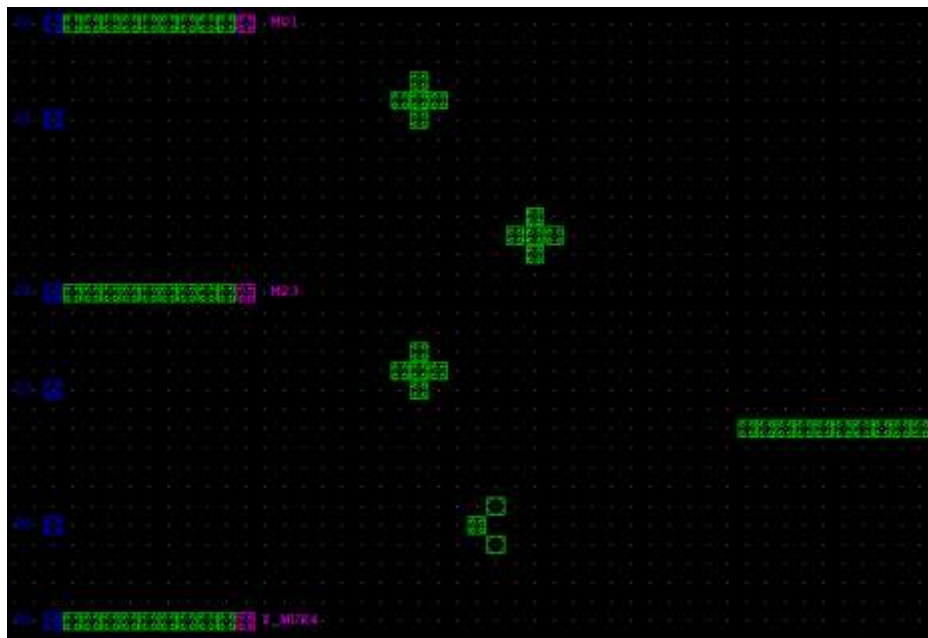


Figure 10: QCA 4:1 multiplexer layout.

The functional behavior of the designed 4:1 multiplexer was checked using the QCADesigner simulation tool, where several combinations of the data inputs I_0 , I_1 , I_2 , and I_3 together with selection inputs S_0 and S_1 were used. The input and output waveforms for this case are shown in Figure 11, and it is clear that Y_{MUX4} varies depending on the selected data input. The value of the simulated output varies between approximately +0.940 for logic 1 and -0.940 for logic 0. It can be seen that the output values vary with the selection inputs and corresponding data inputs, which means that the multiplexer functions properly.

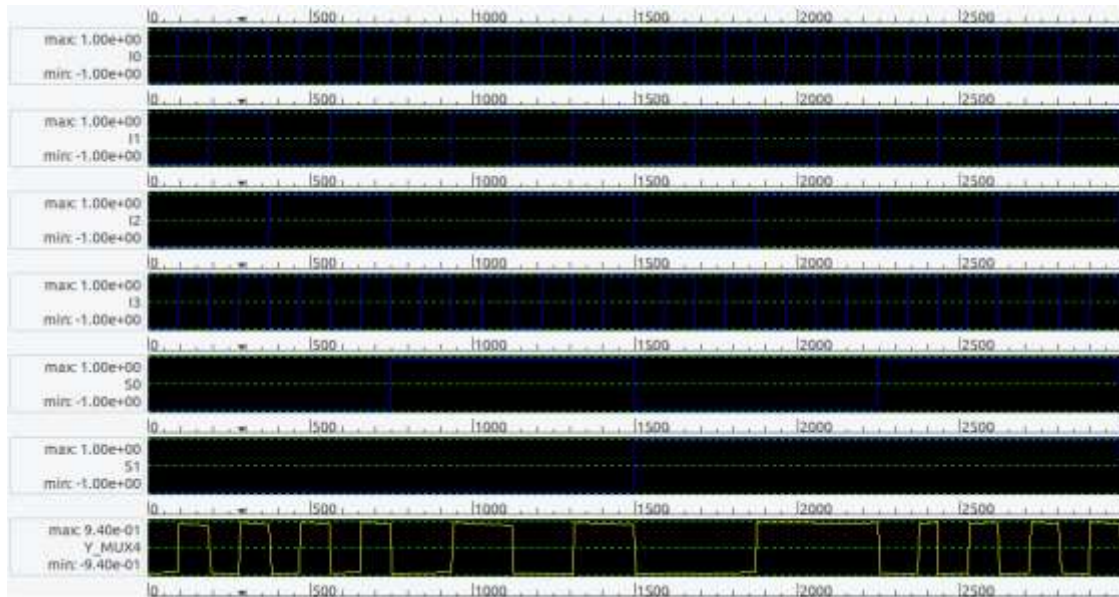


Figure 11: 4:1 multiplexer simulation waveform.

After the design of the fault-tolerant 4:1 multiplexer was verified, the fault-tolerant logic unit was designed to carry out the desired logical operations through the QCA-based circuitry. Figure 12 shows the QCA design of the logic unit with the input signals A and B , control signals S_0 and S_1 and logic-operation paths. The design uses the AND, OR, and NOT functions, along with the multiplexer selection circuitry, to produce the final output Y_{LOGIC} . The internal signal M23 is used as part of the selection path to combine the logic operation outputs produced. The arrangement of the QCA cells ensures that the operation is done through controlled polarization signal operation. The resulting layout establishes the physical implementation of the suggested logic unit for integration into the overall QCA-based ALU.

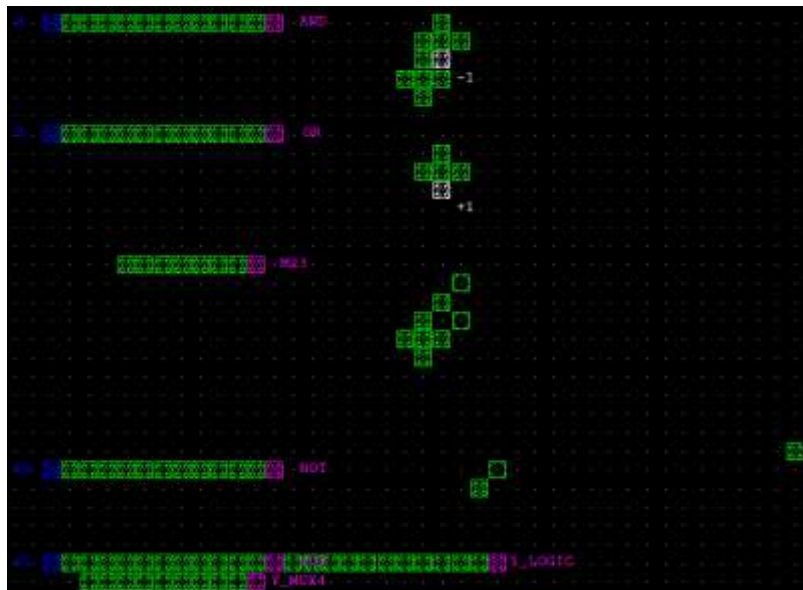


Figure 12: QCA logic unit layout.

The behavior and functional performance of the logic block design were analyzed through the QCADesigner simulation with different combinations of inputs A , B and the control signals S_0 and S_1 . Figure 13 demonstrates the simulation results of the input signals, control signals, individual output signals of each logic operation and the final Y_{LOGIC} signal. Simulation results of AND, OR, NOT, and XOR operations show the expected logic states, with polarization states varying around $+0.940/-0.940$ for AND, OR and NOT operations and $+0.982/-0.982$

for the XOR operation. In addition, the output of the final Y_{LOGIC} shows clearly polarization variation from about +0.982 to -0.982, which corresponds to the selected logic operations.

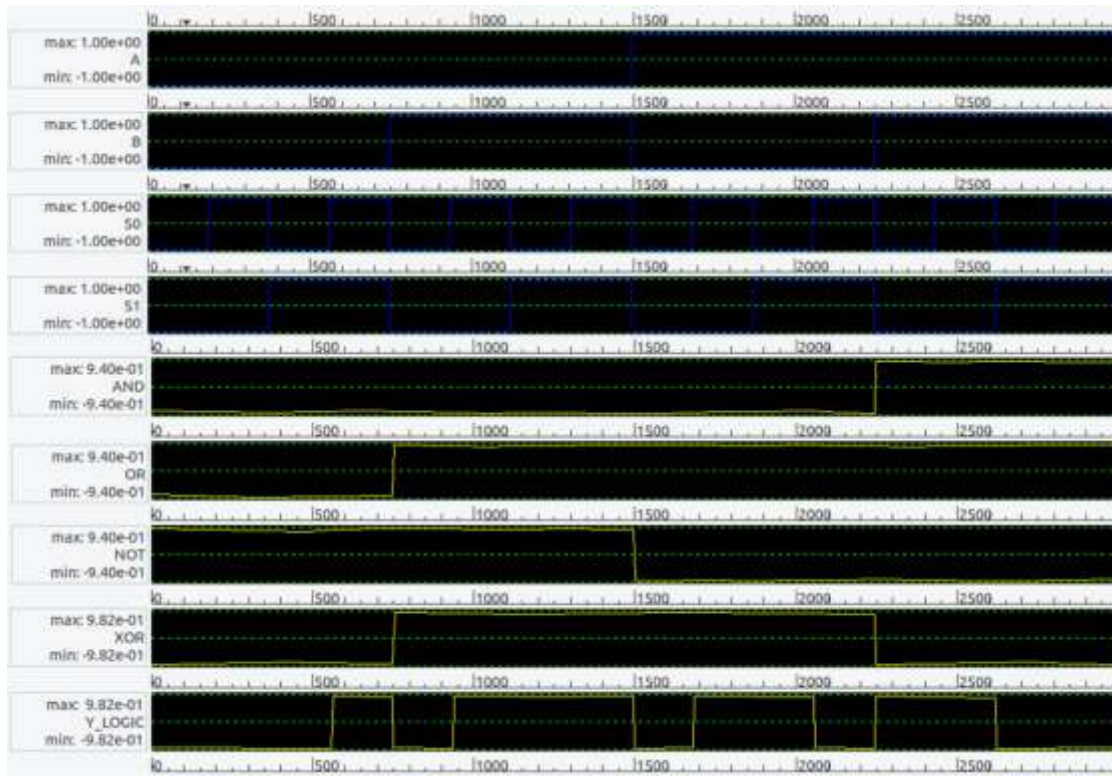


Figure 13: Logic unit simulation waveform.

Following the design of the logic unit, the fault-tolerant arithmetic unit was designed to execute arithmetic operations using the QCA-based full-adder and multiplexer architectures. The QCA layout of the arithmetic unit is illustrated in Figure 14, and this layout comprises the inputs A , B , and C_{in} , control signals S_0 and S_1 , and output signals SUM and $Cout$ produced by the full-adder architecture. The layout also comprises the intermediate output $M23$ and the output of the multiplexer Y_{MUX4} , which are employed in the arithmetic selection process. The combination of these signals using the QCA cell arrangement generates the output of the arithmetic unit Y_{ARITH} . The resulting structure integrates the required arithmetic and selection components into a unified QCA-based arithmetic unit while maintaining the fault-tolerant design approach.



Figure 14: QCA arithmetic unit layout.

The functionality of the arithmetic unit designed was verified through QCA Designer simulation by considering various combinations of the input signals A , B , and C_{in} , along with the control signals S_0 and S_1 . Figure 15 shows the simulated waveforms of the input, control, intermediate arithmetic signals, and the output Y_{ARITH} signal. The SUM signal has polarization values of around +0.940 and -0.940, whereas the C_{out} signal has polarization values of around +0.943 and -0.943, which correspond to the two logic states. The output signal Y_{ARITH} also

changes in response to the input and control signal combination from +0.940 to -0.940. The transition of the output signals verifies the functioning of the arithmetic unit through QCA simulation.



Figure 15: Arithmetic unit simulation waveform.

After the design of the arithmetic unit, the individual functional blocks based on QCA have been combined to form the suggested fault-tolerant ALU. Figure 16 shows the QCA implementation of the designed ALU, including the required input signals, control signals, paths for performing logic operations, paths for performing arithmetic operations, and their corresponding output ports. The designed layout includes the earlier designed full adder and multiplexer circuits along with logic and arithmetic units to form a single ALU circuit. The QCA cells and their connections have been designed in such a way that they allow signal propagation among the functional blocks in the correct operational flow. The resulting integrated layout provides the complete physical realization of the suggested fault-tolerant QCA-based ALU.



Figure 16: Suggested fault-tolerant QCA ALU layout.

The integrated ALU was tested using QCADesigner simulation by testing various combinations of input signals A , B , and C_{in} along with control signals S_0 and S_1 . Figure 17 shows the waveforms of input and control signals, logic output $L_{Y_{LOGIC}}$, arithmetic output $Expected_{Y_{ARITH}}$, and the outputs of the ALU from the simulation. The $L_{Y_{LOGIC}}$ and $Expected_{Y_{ARITH}}$ outputs have polarities of approximately +0.940 and -0.940, which are the two logic levels. The ALU_OUT output from the simulated waveforms has transitions of approximately +0.940 and -0.940 based on the applied combination of inputs and controls. The close match between the expected and simulated outputs indicates that the integrated ALU performs the required operations correctly.

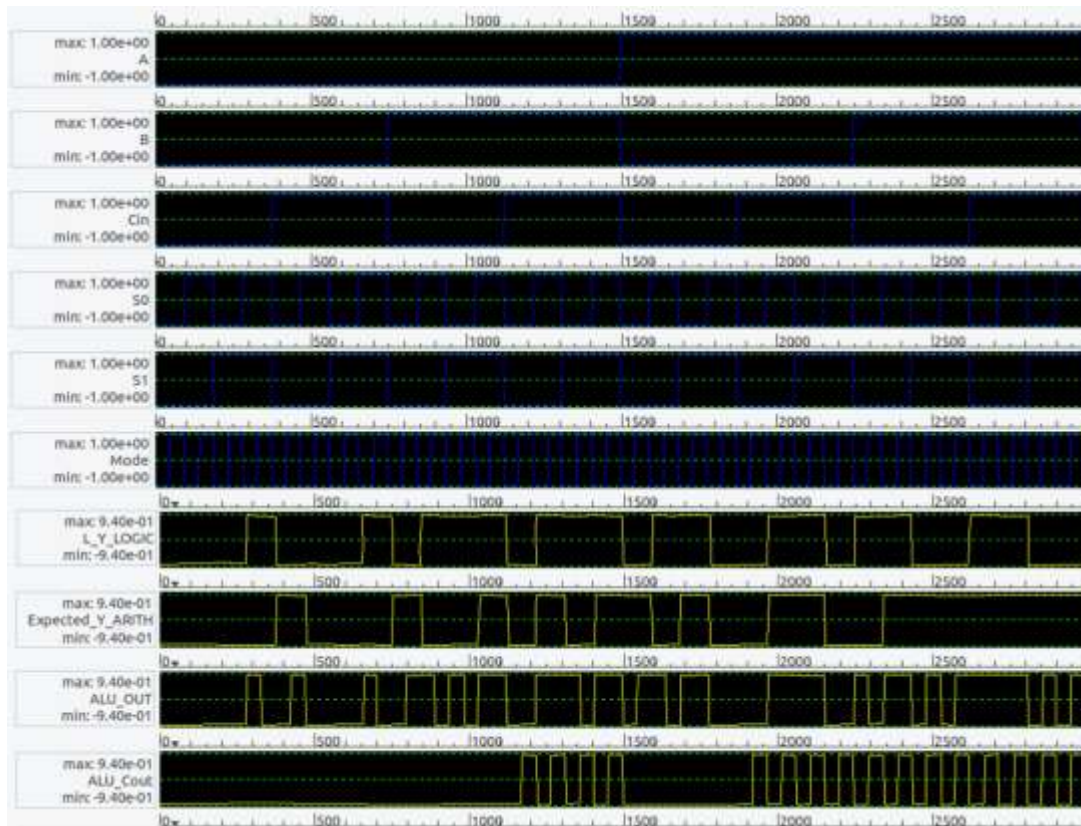


Figure 17: Complete ALU simulation waveform.

Functional verification was done, followed by fault injection analysis to determine the reliability of the ALU that was suggested under different QCA defects. Table 2 shows the findings for the selected fault types and their respective reliability. The outcomes show the reliability of the suggested ALU under the fault conditions.

Table 2: Fault-injection test-vector analysis.

Fault type	Injected cases	Correct outputs	Failed outputs	Reliability (%)
Cell omission	100	96	4	96.00
Cell displacement	100	92	8	92.00
Cell rotation	100	94	6	94.00
Additional-cell deposition	100	95	5	95.00
Overall	400	377	23	94.25

Table 3 shows the details about the test vector, fault position, and output value that can be achieved correctly and incorrectly for individual components and the suggested ALU. The performance of the suggested ALU was checked using 400 test vectors.

Table 3: Fault reliability analysis.

Circuit	Fault type	Fault location/magnitude	Test vectors	Correct outputs	Failed outputs
Majority gate	Omission	Voter/interconnection cells	8	8	0
Full adder	Displacement	5, 10, 15, 20 nm	32	30	2
Multiplexer	Rotation	Selected cells at 45° and 90°	32	30	2
Logic unit	Extra-cell deposition	Selected interconnection locations	64	61	3

Arithmetic unit	All fault types	Multiple critical locations	128	120	8
Suggested ALU	All fault types	Multiple locations across integrated layout	400	377	23

After performing the verification of functionality, the fault injection technique was used to study the reliability of the ALU suggested using QCA defects. Table 4 shows the results obtained from various faults considered, along with their reliability. It can be seen that the circuit works reliably under the faults applied.

Table 4: Performance evaluation of QCA circuits.

Design	Cell count	Occupied Area (μm^2)	Clock delay	Energy dissipation (eV)	Fault reliability (%)
FT majority gate	7	0.003364	1 phase = 0.25 cycle	0.0018	97.50
FT full adder	58	0.128	2 phases = 0.50 cycle	0.0185	96.50
4:1 multiplexer	64	0.142	2 phases = 0.50 cycle	0.0210	96.25
Logic unit	108	0.240	3 phases = 0.75 cycle	0.0355	95.75
Arithmetic unit	136	0.301	3 phases = 0.75 cycle	0.0440	95.25
Proposed ALU	206	0.455	2 phases = 0.50 cycle	0.0730	94.25

The statistical analysis of the proposed ALU was done by one-way ANOVA, sensitivity analysis, 95% confidence interval and Wilcoxon signed-rank test. As illustrated in Table 5 below, the analysis provides statistical evidence of the reliability of the proposed ALU.

Table 5: Statistical validation results.

Analysis	Statistic	p-value/interval
One-way ANOVA	F = 0.535	p = 0.659
Sensitivity analysis	S = -0.40% per 5 nm	—
95% confidence interval	94.25% reliability	[91.50%, 96.32%]
Wilcoxon signed-rank test	W = 2.00	p = 0.043

Table 6 illustrates a comparison of the considered QCA-based designs on the basis of QCA cell count, occupied Area and delay/latency. Chetti and Veena (2026) designed an optimized QCA-based ALU that realized 226 cells, $0.50 \mu\text{m}^2$ area and 0.5 clock cycle delay. Seyedi and Abdoli (2025) designed a fault-tolerant Carry-Skip Adder that yielded 1542 cells, $4.59 \mu\text{m}^2$ Area and 4.75 clock phases. Zohaib et al. (2025) designed a 1-bit QCA-based ALU that realized 528 cells, $0.60 \mu\text{m}^2$ Area and 5.5 clock cycles. Alharbi et al. (2023) designed a reversible QCA-based ALU that realized 1153 cells, $2.14 \mu\text{m}^2$ area and 6 clock cycles. The proposed study designed a fault-tolerant QCA-based ALU that realized 206 cells, $0.455 \mu\text{m}^2$ area and 0.5 clock cycles. As compared to the considered designs, the proposed ALU design achieved the lowest QCA cell count and occupied Area while maintaining the lowest delay level.

Table 6: Comparative analysis

Study	Circuit / Architecture	QCA Cell Count	Area (μm^2)	Delay / Latency
Chetti and Veena (2026) [49]	Optimised QCA-based ALU	226	0.50	0.5 clock cycles
Seyedi and Abdoli (2025) [42]	Fault-tolerant Carry-Skip Adder	1542	4.59	4.75 clock phases
Zohaib et al. (2025) [37]	1-bit QCA-based ALU	528	0.60	5.5 clock cycles

Alharbi et al. (2023) [43]	Reversible QCA-based ALU	1153	2.14	6 cycles
Proposed Study (2026)	Fault-Tolerant QCA- based ALU	206	0.455	0.5 clock cycles

5. Conclusion and Future Scope

QCA provides a promising alternative to conventional CMOS technology for compact and energy-efficient nanoelectronic circuits. However, previous QCA-based ALU designs have limitations related to circuit complexity, Area, delay, and reliability under manufacturing defects, creating a need for an integrated and fault-tolerant ALU architecture. The present study aimed to develop a fault-tolerant QCA-based ALU with improved compactness and reliable operation under considered fault conditions. The methodology involved designing a majority gate, full adder, 4:1 multiplexer, logic unit, and arithmetic unit, followed by their integration into the complete ALU and evaluation through QCA Designer-based functional verification, fault injection, performance analysis, and statistical validation. The proposed ALU achieved 206 cells, an occupied area of 0.455 μm^2 , 0.5 clock-cycle delay, 0.0730 eV energy dissipation, and 94.25% fault reliability. Comparative analysis demonstrated its compact architecture, low Area, and short delay relative to the considered existing designs. The findings confirm the feasibility of integrating fault-tolerant QCA components into a reliable ALU architecture for emerging nanoelectronic applications. Future work can focus on further reducing cell count, area, energy dissipation, and delay while enhancing fault tolerance and extending the architecture toward more complex and scalable QCA-based nanoelectronic systems.

References

1. Zhang, Qingzhu, Yongkui Zhang, Yanna Luo, and Huaxiang Yin. "New structure transistors for advanced technology node CMOS ICs." National Science Review 11, no. 3 (2024): nwae008. <https://doi.org/10.1093/nsr/nwae008>
2. Radamson, Henry H., Yuanhao Miao, Ziwei Zhou, Zhenhua Wu, Zhenzhen Kong, Jianfeng Gao, Hong Yang et al. "CMOS scaling for the 5 nm node and beyond: device, process and technology." Nanomaterials 14, no. 10 (2024): 837. <https://doi.org/10.3390/nano14100837>
3. Bespalov, V. A., N. A. Dyuzhev, and V. Yu Kireev. "Possibilities and limitations of CMOS Technology for the production of various Microelectronic systems and devices," Nanobiotechnology Reports 17, no. 1 (2022): 24-38. <https://doi.org/10.1134/S2635167622010037>
4. Cao, Wei, Huiming Bu, Maud Vinet, Min Cao, Shinichi Takagi, Sungwoo Hwang, Tahir Ghani, and Kaustav Banerjee. "The future transistors." Nature 620, no. 7974 (2023): 501-515. <https://doi.org/10.1038/s41586-023-06145-x>
5. Knobloch, Theresia, Siegfried Selberherr, and Tibor Grasser. "Challenges for nanoscale CMOS logic based on two-dimensional materials." Nanomaterials 12, no. 20 (2022): 3548. <https://doi.org/10.3390/nano12203548>
6. Zeng, Senfeng, Chunsen Liu, and Peng Zhou. "Transistor engineering based on 2D materials in the post-silicon era." Nature Reviews Electrical Engineering 1, no. 5 (2024): 335-348. <https://doi.org/10.1038/s44287-024-00045-6>
7. Kumar, Yash. "Theoretical Foundations and Future Directions of Nanoelectronic Computing: A Quantum-Dot Cellular Automata Perspective." Available at SSRN 5518800. <http://dx.doi.org/10.2139/ssrn.5518800>
8. Strasser, Evelien. "Microelectronics in Evolution: A Systematic Review of Enabling Technologies, Emerging Applications, and Prospective Research Challenges." Transactions on Computational and Scientific Methods 5, no. 4 (2025). <https://doi.org/10.5281/zenodo.15244692>
9. Srinivas, Mamidipaka BR, and Konguvel Elango. "Era of sentinel tech: Charting hardware security landscapes through post-silicon innovation, threat mitigation and future trajectories." IEEE Access 12 (2024): 68061-68108. <https://doi.org/10.1109/ACCESS.2024.3400624>
10. Memon, Qurban A., Mahmoud Al Ahmad, and Michael Pecht. "Quantum computing: Navigating the future of computation, challenges, and technological breakthroughs," Quantum Reports 6, no. 4 (2024): 627-663. <https://doi.org/10.3390/quantum6040039>
11. Raj, Marshal, Lakshminarayanan Gopalakrishnan, and Seok-Bum Ko. "Reliable SRAM using NAND-NOR gate in beyond-CMOS QCA technology." IET Computers & Digital Techniques 15, no. 3 (2021): 202-213. <https://doi.org/10.1080/03772063.2020.1831411>
12. Alharbi, Mohammed. "System Level Simulation of Quantum-Dot Cellular Automata Computer Circuits." PhD diss., Liverpool John Moores University (United Kingdom), 2024.

- <https://www.proquest.com/openview/d434ce08eeba4661a6919ab436dc73a4/1?pq>
13. Jain, Vaibhav, Devendra Kumar Sharma, Hari Mohan Gaur, Ashutosh Kumar Singh, and Xiaoqing Wen. "Comprehensive and comparative analysis of QCA-based circuit designs for next-generation computation." *ACM Computing Surveys* 56, no. 5 (2023): 1-36. <https://doi.org/10.1145/3622932>
 14. Akter, Nasima, Md Abdullah-Al-Shafi, and Md Nasim Akhtar. "QCA Nanoarchitecture for Morphological Processes on Binary Images." *Nanosistemi, Nanomateriali, Nanotehnologii* 23, no. 2 (2025). <https://doi.org/10.15407/nnn.23.02.0333>
 15. Pal, Arnab, Shuo Zhang, Tanmay Chavan, Kunjesh Agashiwala, Chao-Hui Yeh, Wei Cao, and Kaustav Banerjee. "Quantum-engineered devices based on 2D materials for next-generation information processing and storage." *Advanced Materials* 35, no. 27 (2023): 2109894. <https://doi.org/10.1002/adma.202109894>
 16. Thayil, Ruchika, Saidi Reddy Parne, and C. V. Ramana. "2D MoS₂ for next-generation electronics and optoelectronics: from material properties to manufacturing challenges and prospects." *Small* 21, no. 14 (2025): 2412467. <https://doi.org/10.1002/sml.202412467>
 17. Abdi, Athena, and Sina Shahoveisi. "FT-EALU: fault-tolerant arithmetic and logic unit for critical embedded and real-time systems: A. Abdi, S. Shahoveisi." *The Journal of Supercomputing* 79, no. 1 (2023): 626-649. <https://doi.org/10.1007/s11227-022-04698-8>
 18. Birla, Shilpi, Neha Singh, Renu Kumawat, and Avireni Srinivasulu. "Unified Multimodal 64-Bit Arithmetic Logic Unit for High-Performance Computing Architectures." *Journal of VLSI Circuits and Systems* 7, no. 2 (2025): 1-8. <https://doi.org/10.31838/JCVS/07.02.01>
 19. Tiwari, Kanchan S. "Design of generic Vedic ALU using reversible logic." *Memories-Materials, Devices, Circuits and Systems* 9 (2025): 100121. <https://doi.org/10.1016/j.memori.2025.100121>
 20. Vahabi, Mohsen, Muhammad Zohaib, Seyed-Sajad Ahmadpour, and Osman Selvi. "An Area-Efficient QCA-Based Multiplier for High-Performance Nanoscale DSP and Embedded Computing." *Computers* 15, no. 6 (2026): 341. <https://doi.org/10.3390/computers15060341>
 21. Mohammed, Imran Ali. "Transitive Indexing for Main-Memory: Adaptive Performance under Massive Interleaved Queries and Updates." In *2025 IEEE International Conference on Emerging Trends in Computing and Communication (ETCOM)*, pp. 1-6. IEEE, 2025. <https://doi.org/10.1109/ETCOM66606.2025.11436538>
 22. Rostami, Amirhossein, Seyed Mohammad Ali Zeinolabedin, Liyuan Guo, Florian Kelber, Heiner Bauer, Andreas Dixius, Stefan Scholze et al. "NLU: An Adaptive, Small-Footprint, Low-Power Neural Learning Unit for Edge and IoT Applications." *IEEE Open Journal of Circuits and Systems* (2025). <https://doi.org/10.1109/OJCAS.2025.3546067>
 23. Patel, Ami, Vrushank Shah, and Dipti Khurge. "Design and Comparison of Low-Power Fault-Tolerant Logic Circuit Using QCA Nanotechnology." (2025). <https://doi.org/10.21203/rs.3.rs-6461225/v1>
 24. Pramanik, Amit Kumar, Jayanta Pal, and Bibhash Sen. "Analysis of fault tolerance capability of QCA circuit under regular clocking." *International Journal of Electronics* 112, no. 11 (2025): 2449-2471. <https://doi.org/10.1080/00207217.2025.2450738>
 25. Chetti, Smita C., and Omkar Yatgal. "QCA: A survey and design of logic circuits." *Global Transitions Proceedings* 3, no. 1 (2022): 142-148. <https://doi.org/10.1016/j.gltp.2022.04.012>
 26. Gelman, Len, and Raveendra H. Patil. "Design of Fault-Tolerant Nanoelectronic Architectures for Low-Power Computing Applications." In *ECCSUBMIT Conferences*, vol. 2, no. 1, pp. 107-114. 2024. <https://eccsubmit.com/index.php/congress/article/view/25>
 27. Rech, Paolo. "Artificial neural networks for space and safety-critical applications: Reliability issues and potential solutions." *IEEE Transactions on Nuclear Science* 71, no. 4 (2024): 377-404. <https://doi.org/10.1109/TNS.2024.3349956>
 28. Fassi, Youssef, Vincent Heiries, Jerome Boutet, and Sebastien Boisseau. "Toward physics-informed machine-learning-based predictive maintenance for power converters—a review." *IEEE Transactions on Power Electronics* 39, no. 2 (2023): 2692-2720. <https://doi.org/10.1109/TPEL.2023.3328438>
 29. Khan, Fatima Hameed, Muhammad Adeel Pasha, and Shahid Masud. "Advancements in microprocessor architecture for ubiquitous AI—An overview of history, evolution, and upcoming challenges in AI implementation." *Micromachines* 12, no. 6 (2021): 665. <https://doi.org/10.3390/mi12060665>
 30. Kalyan, Birinderjit Singh, and Balwinder Singh. "Fault-tolerant quantum-dot cellular automata linear feedback shift register for nano communication applications." *Results in Optics* 16 (2024): 100705. <https://doi.org/10.1016/j.rio.2024.100705>

31. Kandasamy, Nehru, Firdous Ahmad, Devi Ajitha, Balwinder Raj, and Nagarjuna Telagam. "Quantum dot cellular automata-based scan flip-flop and boundary scan register." *IETE Journal of Research* 69, no. 1 (2023): 535-548. <https://doi.org/10.1080/03772063.2020.1831411>
32. Chugh, Hemanshi, and Sonal Singh. "An energy- and cost-effective realization of high-performance single-layer Vedic multiplier in quantum-dot cellular automata." *Physica Scripta* 100, no. 6 (2025): 065112. <https://doi.org/10.1088/1402-4896/add2ae>
33. Singh, Ayushi Kirti, Subodh Wairya, and Divya Tripathi. "Cell Optimization and Realization of Vedic Multiplier Design in QCA." (2023). <http://dx.doi.org/10.12785/ijcds/1401116>
34. Mohammed, Imran Ali. "Optimizing Cross-Domain Assurance: A Signaling Approach to Reduce Accreditation Costs." In the 2026 International Conference on Smart Futuristic Technology, pp. 1-6. IEEE, 2026. <https://doi.org/10.1109/ICSFT66733.2026.11508043>
35. Varanasi, Anirudh, Robin Degraeve, Philippe Roussel, and Clement Merckling. "RTNinja: A generalized machine learning framework for analyzing random telegraph noise signals in nanoelectronic devices." *APL Machine Learning* 3, no. 4 (2025). <https://doi.org/10.1063/5.029545>
36. La Via, Francesco, Daniel Alquier, Filippo Giannazzo, Tsunenobu Kimoto, Philip Neudeck, Haiyan Ou, Alberto Roncaglia, Stephen E. Sadow, and Salvatore Tudisco. "Emerging SiC applications beyond power electronic devices." *Micromachines* 14, no. 6 (2023): 1200. <https://doi.org/10.3390/mi14061200>
37. Zohaib, Muhammad, Nima Jafari Navimipour, Mehmet Timur Aydemir, and Seyed-Sajad Ahmadpour. "A nanoscale design of arithmetic and logic unit for energy-efficient signal processing devices based on a quantum-based technology." *Cluster Computing* 28, no. 5 (2025): 340. <https://doi.org/10.1007/s10586-024-05073-3>
38. Premananda, B. S., Mohsen Vahabi, Muhammad Zohaib, Seyed-Sajad Ahmadpour, M. Barath, and K. R. Sreesha. "An Efficient Quantum-Dot Cellular Automata Memory Architecture for Internet of Things Systems." *Computers* 15, no. 5 (2026): 302. <https://doi.org/10.3390/computers15050302>
39. Fayazi, Shobeir, and Hatam Abdoli. "Low-cost approximate multipliers for quantum-dot cellular automata." *Scientific Reports* (2026). <https://doi.org/10.1038/s41598-026-62704-y>
40. Jahanshahi Javaran, Farzaneh, Somayyeh Jafarali Jassbi, Hossein Khademolhosseini, and Razieh Farazkish. "Quantum-Dot Cellular Automata-based and High-speed Design of a New Structure for Fault-Tolerant 7-Input Majority Gate." *International Journal of Industrial Electronics Control and Optimization* (2026). https://ieco.usb.ac.ir/article_9684_dd953eb8aa666612d465bc603ea8c6a6.pdf
41. Abdullah-Al-Shafi, Mohammad. "RAM, DEMUX and ALU in nanoscale: a quantum-dot cellular automata-based architecture." *Discover Electronics* 2, no. 1 (2025): 26. <https://doi.org/10.1007/s44291-025-00066-x>
42. Seyedi, Saeid, and Hatam Abdoli. "A fault-tolerant CSA in QCA technology for IoT devices." *Scientific Reports* 15, no. 1 (2025): 3396. <https://doi.org/10.1038/s41598-025-85933-z>
43. Alharbi, Mohammed, Gerard Edwards, and Richard Stocker. "Hybrid Quantum-Dot Cellular Automata Nanocomputing Circuits." *Electronics* 13, no. 14 (2024): 2760. <https://doi.org/10.3390/electronics13142760>
44. Kassa, Sankit, Jadav Chandra Das, Vijay Lamba, Debashis De, Bikash Debnath, Saurav Mallik, and Mohd Asif Shah. "Novel design of cryptographic architecture of nanorouter using quantum-dot cellular automata nanotechnology." *Scientific Reports* 14, no. 1 (2024): 10532. <https://doi.org/10.1038/s41598-024-61260-7>
45. Alharbi, Mohammed, Gerard Edwards, and Richard Stocker. "Reversible quantum-dot cellular automata-based arithmetic logic unit." *Nanomaterials* 13, no. 17 (2023): 2445. <https://doi.org/10.3390/nano13172445>
46. Ravi, N., and M. B. Veena. "Design of an efficient ALU block in quantum dot cellular automata (QCA)." *Global Transitions Proceedings* 3, no. 1 (2022): 157-168. <https://doi.org/10.1016/j.gltp.2022.03.004>
47. Galadima, Bahijja Yahaya, and G. S. Galadanci. "QCA-BASED design of reversible Hamming code encoding, Decoding and Correcting Circuits." https://www.researchgate.net/profile/Bahijja-Yahaya-Galadima/publication/382630069_pdf
48. Pant, Hira Ballabh, Tripura Batchu, Raashi Raj, A. Y. Nirupama, and Varun Agiwal. "Impact of hands-on training in STATA for data management and data analysis: How much knowledge is gained among health-care professionals? *IHOPE Journal of Ophthalmology* 2, no. 1 (2023): 4-8. https://doi.org/10.25259/IHOPEJO_30_2022
49. Chetti, Smita C. "Design of Optimised ALU using Quantum Dot Cellular Automata." *Journal of Intelligent Decision Making and Information Science* 3, no. 9s (2026): 973-984. <https://www.jidmis.org/index.php/jidmis/article/view/2455>