OPEN ACCESS
Journal of Multidisciplinary Applied Natural Science Vol. 6 No. 1 (2026) https://doi.org/10.47352/jmans.2774-3047.331
Check for updates
Research Article

Photoresist Thickness Variation-Aware Design of a Power-Efficient 4-Bit Arithmetic Logic Unit using PD-RBB and 2SMWO Technique

Shylaja Veerabhadraiah*, N Kannan, and T Y Satheesha

Received : May 20, 2025	Revised : September 4, 2025	Accepted : September 15, 2025	Online : December 24, 2025

Abstract
The Arithmetic Logic Unit (ALU) is a fundamental component for executing arithmetic and logical operations in digital systems. However, existing ALU designs often overlook variations in photoresist thickness during fabrication, leading to increased gate leakage and reduced reliability. This paper proposes a power-efficient 4-bit ALU design that is aware of photoresist thickness fluctuations, leveraging Pareto Distribution Reverse Body Biasing (PD-RBB) and Singer Spider Map Wasp Optimization (2SMWO). Fluctuation detection and mitigation are achieved through a Bernstein Polynomial Sigmoid Fuzzy Inference System (BPS-FIS), while leakage current and transistor failures are addressed using PD-RBB and Wishart Distribution-based Triple Modular Redundancy (Wd-TMR), respectively. To enhance reversibility and reduce power loss, the Golay Ternary Reversible Gate (GTRG) is integrated. Division complexity is managed via Sigmoid-based Correction Non-Restoring Division (SbC-NRD), and a Carry Look-Ahead Adder (CLAA) supports parallel processing. Experimental results demonstrate a 92% reduction in leakage current, confirming the effectiveness of the proposed approach in improving power efficiency and reliability.

Keywords: arithmetic logic unit, carry look-ahead adder, leakage power optimization, photoresist thickness variation, transistor reliability

1. INTRODUCTION

A universal logic combinational device and a crucial computational unit, making it an important component of powerful processors is ALU [1]. The processor employs the ALU for performing arithmetic operations like addition, subtraction, multiplication, and division, along with logical operations like OR, AND, XOR, inversion, and transformation operations. Likewise, ALU helps to design full-adder circuits along with low-order multiplexer circuits of different sizes [2]. Usually, the ALU takes two input operands (i.e., A and B), and processes them according to predefined operations to produce the output. Generally, for higher-speed and lower-power applications, including image processing, digital signal processing, and so on, Arithmetic operations are important [3][4]. Yet, in designing embedded processors [5], power consumption is a serious issue. The key deciding factors of power dissipation are frequency of operation, fabrication technology, along with switching per clock cycle [6].

Technology scaling's outcome has increased power consumption as well as delay. Thus, the main problem in any computational unit's design is having a power-efficient ALU [7]. The increasing market demand for low power, faster, and more compact devices is a key factor in designing devices that satisfy the growing market demand. Thus, designing a low-power and high-performance 4-bit ALU is an essential aspect of modern digital circuit design, especially for applications needing energy efficiency [8]. Designing low-power and high-performance ALU is difficult owing to the trade-offs betwixt speed, power, and area [9]. Recently, because of the rising number of portable devices, many techniques have been developed for designing a low power with outstanding performance ALU [10]-[14]. A 16-bit ripple carry adder/subtractor with XOR gates, a 16-bit pipelined vedic multiplier, and a 16-bit logic unit using operand isolation were employed for designing power-efficient ALU in existing studies [15]. Similarly, some existing works employed QCA-constructed full adder logic circuits centered on '5' input majority gates for designing high-performance ALU [16][17].

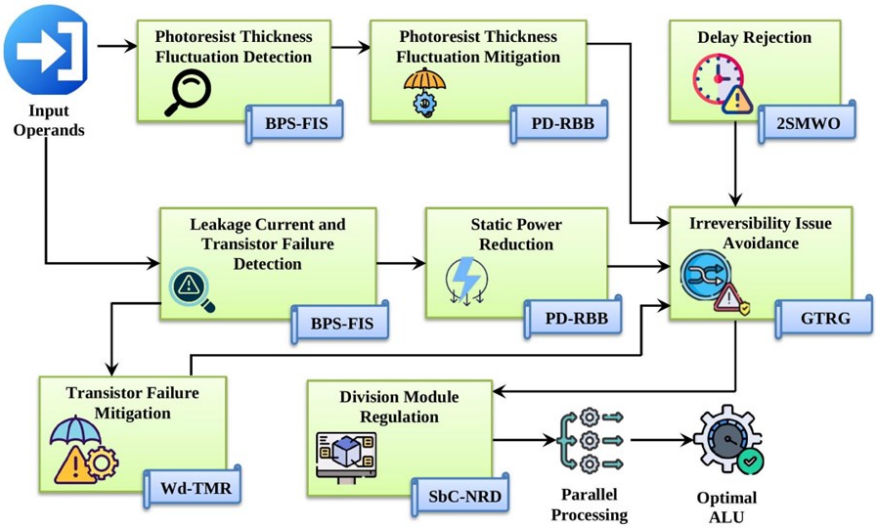

Figure 1. Structural representation of the proposed model.

Similarly, dual-mode pass transistor logic was used in conventional studies for implementing energy-efficient and high-performance ALU [18]. Also, for designing a power-efficient Reversible ALU circuit, certain prevailing works utilized a Hybrid Nano-Gate and three Fredkin gates [19][20]. Also, for implementing high-performance ALU, a reversible Three-Sub-Gate (TSG) with a reversible AND-OR gate was used [21]. Furthermore, a microring-resonator–centric all-optical switch using a pump–probe configuration was employed to design a power-efficient ALU [22]. Also, for performing multiple operations, some conventional works used Quantum Fourier Transform (QFT)-enabled quantum ALU [23]. Yet, the prevailing studies failed to mitigate the photoresist thickness fluctuations present in the design of 4-bit ALU, thus leading to increased gate leakage in ALU operations. To address these problems, a novel photoresist thickness fluctuation-aware optimal design of power-efficient 4-bit ALU using PD-RBB and 2SMWO is proposed in this article.

The problems statement for this current research can be explored with conventional works regarding optimal designing of 4-bit ALU have some limitations, which are described as follows. At first, none of the existing works concentrated on photoresist thickness fluctuations present in the design of 4-bit ALU, thereby leading to increased gate leakage, higher power consumption, and thermal instability in ALU operations. Second, the conventional, failed to prevent the risk of transistor failures and soft errors, thus diminishing the system's resilience [24]. Third, significant input information loss might happen due to the irreversibility issue presented in the existing [25]. Also, performing high-speed computations was difficult. Fourth, the delay arose in the sequential execution of arithmetic along with logic operations owing to the intricate interconnections amongst different circuit elements in prevailing [14]. Fifth, the divider module was added; but, special considerations were not laid on this module, thus leading to slower computation and bottleneck issues in existing [26]. Sixth, in most of the prevailing works, 4-bit ALU was designed as a single-cycle combinational circuit, which lacked parallel execution mechanisms. Seventh, in prevailing works, an idle state caused current leakage in the transistor when the switching operation was not performed.

The key objectives of the proposed model are described as follows. BPS-FIS is introduced to detect the photoresist thickness fluctuations, and PD -RBB is employed to mitigate the photoresist thickness fluctuations. BPS-FIS is used to detect transistor failures, and Wd-TMR is established to mitigate the transistor failures. GTRG is utilized to avoid the irreversibility issue, thus effectively performing high-speed computations. 2SMWO is established to reject the processing delay that occurs due to the sequential execution of operations. SbC-NRD is employed to regulate the complex division modules, thus avoiding slower computation. CLAA is utilized for performing parallel processing. PD-RBB is used for static

power reduction in the transistor.

Safaizadeh et al. [24] offered a framework named Design and Simulation of Reversible ALU. Here, for designing the reversible ALU, the Quantum-dot Cellular Automata (QCA) technology was utilized. Here, reversible BS1 blocks were deployed by the reversible ALU. Therefore, the model excellently improved the quantum cost, occupied area, and the number of cells. Yet, it didn't prevent the risk of transistor failures and soft errors, thus reducing the system's resilience. Liu et al. [25] introduced an effective machine learning-based Design Technology Co-Optimization (DTCO) for circuits. A ridge regression technique was employed for bypassing the simulation, compact model extraction, and cell library characterization. The model excellently reduced the time consumption, errors of delay, internal power consumption, along with leakage power. However, the research had an irreversibility issue, which might lead to significant input information loss. Tiwari [14] recommended an implementation of generic Vedic ALU. The Vedic mathematics principles were utilized for deriving excellent computational methods. Likewise, in the ALU, minimal power dissipation and reduced heat

Table 1. The pseudocode for BPS-FIS.

Algorithm 1: Pseudo code for BPS-FIS

Input: Photoresist thickness variations *(pth)*, Gate leakage current *(G)*, Power consumption *(PC)*, and Thermal instability *(I)*

Output: Photoresist thickness fluctuation detection outcomes *(PTO)*

Begin

Initialize *(pth)*, *(G)*, *(PC)*, and *(I)*
Consider *(pth)*, *(G)*, *(PC)*, and *(I)* as J_{ab}

For J_{ab}

Estimate fuzzy rules $(\Im)$

Find Bernstein polynomial sigmoid membership function

$\beta = \sum_{z=0}^{m} \frac{b_z^m(J_{ab})}{1+\exp^{-v\left(\frac{f}{m}-j\right)}}$

Perform decision making unit

$\partial m \overset{\text{operator}}{\rightarrow} \rho\varepsilon$

Implement fuzzification process

$Fp = J_{ab} \overset{\beta}{\rightarrow} \Im$

Compute defuzzification process

$Dp = (\Im \rightarrow J_{ab})$

End For

Obtain Photoresist thickness fluctuation detection outcomes $PTO = [\ddot{y}, \vartheta]$

End

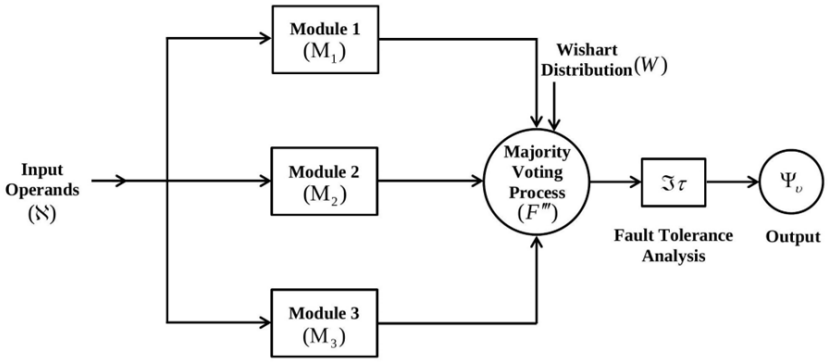

Figure 2. Architectural diagram of the proposed Wd-TMR.

generation were achieved by utilizing reversible logic. The model excellently performed fundamental arithmetic operations and bitwise logical operations and improved the overall performance. The delay arose in the sequential execution of arithmetic and logic operations due to the intricate interconnections among different circuit elements.

Jujjavarapu and Poulose [26] explored a design of Inter of Things (IoT)-enabled arithmetic logic and compression unit. Here, a compression module and a fast multiplier were integrated into the design of the 16-bit ALU according to the Vedic algorithm. The ALU was synthesized under 32-nanometer High-Voltage Threshold (HVT) cells from the Synopsys database for creating the overview of logic levels and areal efficiency. The model aided in quick computational processes. However, special considerations were not laid on the division module, leading to slower computation and bottleneck issues. Jadhav et al. [27] elucidated the designing of arithmetic and logical units. Here, the Micro-Ring Resonator (MRR) was used for performing one-bit addition and one-bit comparison, which made the model easier to integrate with Very Large Scale Integrated Optics (VLSIO). Here, for attaining low-intensity losses, the beam splitter was replaced with MRR. Therefore, an optimized design of ALU was obtained. Nevertheless, the model was sensitive to temperature fluctuations, leading to computation errors. Razmkhah and Bozbey [28] considered an ALU for ultra-fast computing. Here, for designing a minimal fan in/fan out, optimal latency, and high-efficiency ALU module, the model employed superconductor technology and bit-parallel architecture. Likewise, by employing a standard commercial process, the ALU was fabricated. Therefore, the model achieved impressive clock frequency and reduced power consumption. Besides, it superiorly performed arithmetic and logic operations. However, the research introduced additional design and power delivery challenges.

Vanlalchaka et al. [29] advocated a model for designing a 16-bit ALU with low power. Here, the ALU module was designed based on the Metal–Oxide-Semiconductor Field Effect Transistor (MOSFET) and Fin Field-Effect Transistor (FinFET) technologies. Likewise, the ALU module incorporated the adiabatic switching principle. Here, adiabatic logic families, including Quasi-Static Energy Recovery Logic (QSERL), two Phase drive Adiabatic Dynamic Complementary Metal Oxide Semiconductor (CMOS) Logic (2PADCL), and Adiabatic Dynamic CMOS Logic (ADCL) were used. The model excellently saved the power. Yet, adiabatic logic circuits increased the transistor count, leading to a larger chip area. Roy et al. [30] recognized a model for designing a reversible ALU. Here, the reversible gate (i.e., modified 3:3 TSG gate) with QCA technology was employed for optimally designing the ALU with low delay and power dissipation. Therefore, the model reduced the occupied area, delay, complexity, and power but the model was prone to fabrication defects and charge leakage, thus introducing errors in computation.

Subbulakshmi et al. [31] advanced a framework named implementation of low power ALU-enabled sliced processor. Here, the Gated Diffusion Input (GDI) logic and modified GDI logic were employed for the effective designing of ALU. The power consumption of the ALU was diminished based on a recursive parallel self-timed adder. The model excellently diminished the number of transistors. However, while performing some logic operations,

it introduced higher propagation delays. Abdi and Shahoveisi [32], designed a fault-tolerant ALU for critical embedded systems. Here, simple shift and swap operators were used to avoid the high time overhead. Here, the fault tolerance approach [33] passed the replicated and varied outcomes to a weighted voter designed regarding reward/ punishment strategy. The model proficiently improved the system's reliability but it had some latency, thus degrading the ALU's processing speed.

Despite various advancements in ALU design, several critical research gaps remain unaddressed. The Safaiezadeh's work [24] utilized QCA for reversible ALU design but failed to mitigate transistor failures and soft errors, compromising system resilience. Similarly, the machine learning-based DTCO approach in Liu's work [25] efficiently reduced simulation overhead but suffered from irreversibility, which risks significant input information loss. The Vedic ALU proposed in Tiwari's work [14] minimized power dissipation using reversible logic; however, it introduced delays due to sequential execution and complex circuit interconnections. In Jujjavarapu and Poulose' model [26], the integration of a compression unit and Vedic algorithm improved processing but neglected division module optimization, causing computational bottlenecks. The model reported by Jadhav et al. [27] based on MRR enabled compact optical design but was highly sensitive to temperature fluctuations, affecting reliability. Although the ultra-fast ALU using superconductors by Ramzkhah and Bozbey

Table 2. Pseudocode for Wd-TMR.

Algorithm 2: Pseudocode for Wd-TMR

Input: Input operands $(\aleph)$

Output: Transistor failure mitigation outcomes (ψ_v)

Begin

Initialize $(\aleph)$

For $(\aleph)$

Estimate three modules

$M_1 = fn(\aleph)$
$M_2 = fn(\aleph)$
$M_3 = fn(\aleph)$

Perform majority voting process

$F''' = \left((M_1 \wedge M_2) \vee (M_2 \wedge M_3) \vee (M_3 \wedge M_1)\right) * W$

Compute Wishart distribution function

$$W = \frac{|\aleph|^{\frac{fd-h-1}{2}} \exp^{-\frac{1}{2}\mathrm{tra}}(\sum^{-1}\aleph)}{2^{\frac{fdh}{2}}|\sum|^{\frac{fd}{2}} Gm_h\left(\frac{fd}{2}\right)}$$

Implement fault tolerant analysis

$\Im\tau = 3Pb_y^2 - 2Pb_y^3$

End For

Obtain (ψ_v)

End

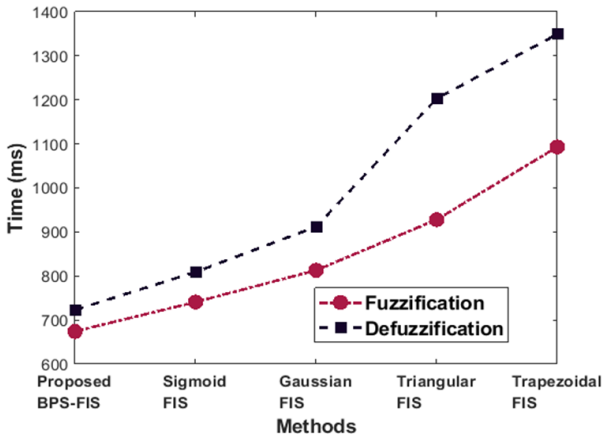

Figure 3. Graphical representation regarding fuzzification and defuzzification time.

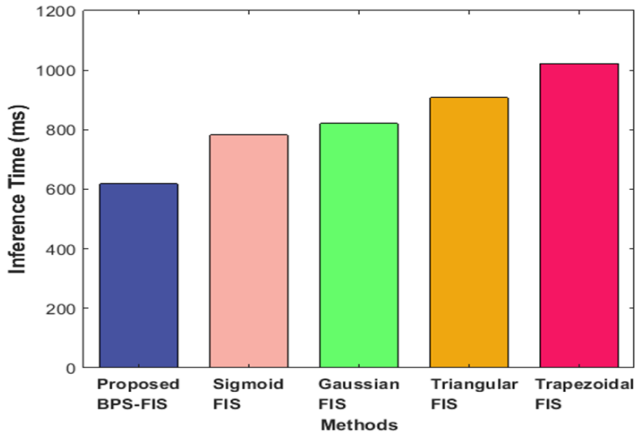

Figure 4. Inference time analysis.

[28] offered excellent speed and energy efficiency, it introduced complex design and power delivery challenges. The adiabatic logic-based approach by Vanlalchaka et al. [29] conserved power but at the cost of increased transistor count and chip area. QCA-based reversible gates reduced delay and power yet remained prone to fabrication defects and charge leakage [30]. The sliced processor ALU achieved low power through GDI logic but encountered high propagation delays in certain logic operations [31]. Lastly, the fault-tolerant ALU for embedded systems improved reliability through a weighted voting mechanism, but latency issues persisted, potentially impacting performance in real -time applications [32].

2. MATERIALS AND METHODS

Here, to detect the photoresist thickness fluctuation, leakage current, and transistor failure, the proposed BPS-FIS is introduced. The proposed PD-RBB is employed for mitigating the photoresist thickness fluctuation and reducing the static power. Likewise, to mitigate the transistor failure, the proposed Wd-TMR is established. The proposed GTRG is utilized to avoid the irreversibility issue. Similarly, to reject the processing delay that occurs due to the sequential execution of operations, the proposed 2SMWO is used. The proposed SbC-NRD is used for regulating the complex division modules. In Figure 1, the proposed model's structural representation is displayed. The proposed methodology excellently mitigated the photoresist thickness fluctuations, transistor failures, and leakage power for optimal designing of ALU.

2.1. Input Operands

Initially, for performing the ALU operation, two input operands are given as input. The photoresist thickness fluctuation is detected while processing these operands, which is explained in the upcoming section. It is expressed as Eq. 1.

$$\aleph \xrightarrow{\text{operands}} (\alpha, \gamma) \quad (1)$$

where, $\aleph$ states the input and (α , γ) signifies the two input operands.

2.2. Photoresist Thickness Fluctuation Detection

Here, regarding photoresist thickness variations ρth , gate leakage current (G), power consumption (PC), and thermal instability (I), the photoresist thickness fluctuation is detected while processing $\aleph$. BPS-FIS is used for detecting the photoresist thickness fluctuation, thus reducing the gate leakage, power consumption, and thermal instability in ALU operations. Usually, the photoresist thickness variations are not always sharp or binary. The FIS can effectively deal with these gradual deviations. The system may miss the real defects if the membership function is not tuned correctly in FIS. To address this problem, the Bernstein polynomial sigmoid membership function is employed in FIS. The working process of BPS-FIS is explained as follows,

2.2.1. Rules Formation

Here, the inputs ρth, (G) , (PC), and (I) are considered as J_{ab}. Primarily, based on if-then conditions, the fuzzy rules ($\Im$) are estimated and are given as Eq. 2.

$$\Im \xrightarrow{J_{ab}} \begin{cases} \text{if } \rho\text{th} = 0 - 5nm \;\&\&\; G = 0 - 10nA \;\&\& & \text{Then } \ell \\ PC = 0 - 5mW \;\&\&\; I = 0 - 2^{\circ}\text{ Celcius Variation} & \\ \text{if } \rho\text{th} > 5nm \;\&\&\; G > 10nA \;\&\& & \text{Then } \hbar \\ PC > 5mW \;\&\&\; I > 2^{\circ}\text{ Celcius Variation} & \\ \text{if } \rho\text{th} = \ell \;\&\&\; G = \ell \;\&\& & \text{Then } \because \\ PC = \ell \;\&\&\; I = \ell & \\ \text{if } \rho\text{th} = \hbar \;\&\&\; G = \hbar \;\&\& & \text{Then } \vartheta \\ PC = \hbar \;\&\&\; I = \hbar & \end{cases} \quad (2)$$

where, ℓ indicates the low range, $\hbar$ signifies the high range, nm denotes the nanometer, nA indicates nanoampere, and mW signifies megawatt.

2.2.2. Bernstein Polynomial Sigmoid Membership Function

The Bernstein polynomial sigmoid membership function (β) is utilized for improving the performance of photoresist thickness fluctuation detection. This is mathematically expressed as Eq. 3.

$$\beta = \sum_{z=0}^{m} \frac{b_z^m(J_{ab})}{1+\exp^{-v\left(\frac{z}{m}-j\right)}} \quad (3)$$

where, b_z^m signifies the Bernstein basis polynomial, exp designates the exponential function, v controls the slope, j finds the center of the sigmoid function, z depicts the index of the basis polynomial, and m defines the polynomial order.

2.2.3. Decision Making Unit

The decision-making unit (∂m) executes interference operations, such as fuzzification and defuzzification in BPS-FIS. It is given as Eq. 4.

$$\partial m \xrightarrow{\text{operator}} \rho\varepsilon \quad (4)$$

where, $\rho\varepsilon$ signifies the operation done by the decision-making unit.

2.2.4. Fuzzification Interference Unit

By utilizing (β), the crisp data (j_{ab}) is converted into fuzzy data ($\Im$). It is equated as Eq. 5.

$$Fp = J_{ab} \xrightarrow{\beta} \Im \quad (5)$$

where, Fp denotes the fuzzification process.

2.2.5. Defuzzification Interference Unit

The fuzzy data ($\Im$) is converted into crisp data (J_{ab}) in the defuzzification interference unit (Dp) and is specified as Eq. 6.

Table 3. Memory utilization evaluation.

Techniques	Memory Utilization (kB)
Proposed BPS-FIS	391
Sigmoid FIS	510
Gaussian FIS	612
Triangular FIS	721
Trapezoidal FIS	819

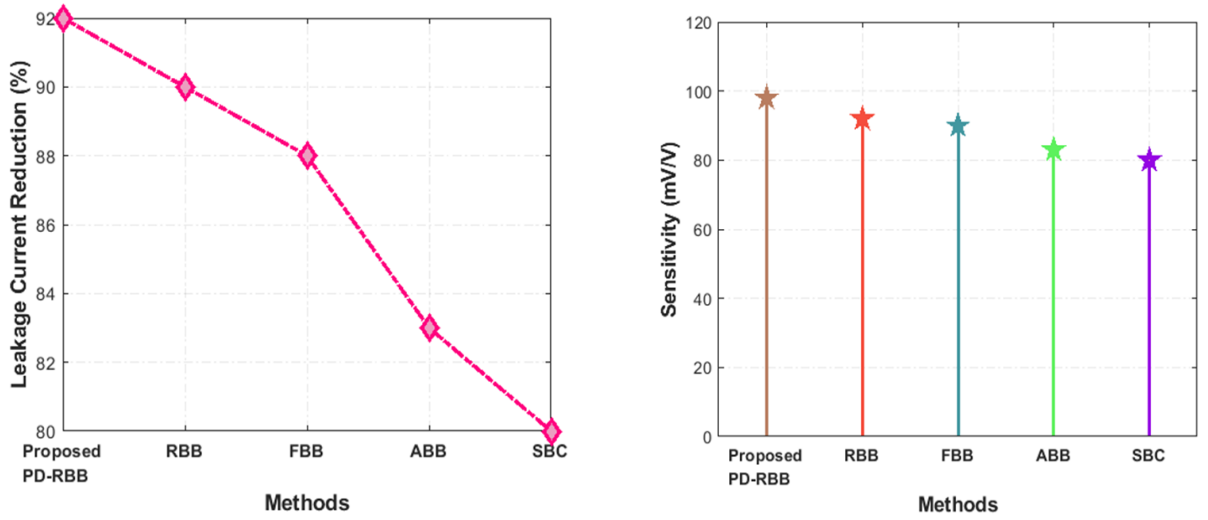

Figure 5. Performance validation regarding (a) leakage current reduction and (b) sensitivity.

$$Dp = (\Im \rightarrow J_{ab}) \quad \textbf{(6)}$$

Next, the photoresist thickness fluctuation detection outcomes (*PTO*) are provided as Eq. 7.

$$PTO = [\ddot{\gamma}, \vartheta] \quad \textbf{(7)}$$

where, $\ddot{\gamma}$ implies the absence of photoresist thickness fluctuation and ϑ depicts the presence of photoresist thickness fluctuation. The pseudocode for BPS-FIS is given in Table 1. Therefore, the proposed BPS-FIS excellently detected the photoresist thickness fluctuation.

2.3. Photoresist Thickness Fluctuation Mitigation

If the photoresist thickness fluctuation is present (ϑ), then it is mitigated by using the Pareto Distribution Reverse Body Biasing (PD-RBB) technique. Reverse Body Biasing (RBB) rapidly diminishes the subthreshold leakage by increasing threshold voltage $(V_{\tau\hbar})$, which is worsened by fabrication variations. Likewise, RBB effectively stabilizes the leakage currents in transistors that are affected by photoresist non-uniformities but the high reverse bias may induce undesired body effects in RBB, leading to parasitic leakage paths. To address this problem, Pareto distribution is employed instead of applying a fixed RBB dynamically, thus preventing the parasitic leakage paths. The step-by-step working of PD-RBB is described as follows Eq. 8–10. Initially, the threshold voltage $(V_{\tau\hbar})$ is expressed as,

$$V_{\tau\hbar} = V_{\tau\hbar 0} + b\varepsilon\left(\sqrt{|V_{sr} + 2F_{pot}|} - \sqrt{|2F_{pot}|}\right) \quad \textbf{(8)}$$

$$b\varepsilon = \frac{\sqrt{2elpD_{sub}}}{Ox} \quad \textbf{(9)}$$

$$F_{pot} = \frac{\kappa Tmp}{el} \ln\left(\frac{D_{sub}}{inr}\right) \quad \textbf{(10)}$$

where,$V_{\tau\hbar 0}$specifies threshold voltage at $V_{sr} = 0, b\varepsilon$, describes body effect coefficient, V_{sr} indicates source -to-body voltage, F_{pot} signifies Fermi potential, *el* states the electron charge, *p* indicates the permittivity of silicon, D_{sub} denotes the doping concentration of the substrate, *Ox* specifies the oxide capacitance per unit area, *k* indicates the Boltzmann constant, *Tmp* indicates the absolute temperature, ln indicates the natural logarithm, and *inr* signifies the intrinsic carrier concentration of silicon. Then, RBB is applied. Here, to enhance the threshold voltage, a negative reverse bias voltage (V_{bias}) or a positive reverse bias voltage (V_{bias}) is applied. Here, to avoid the parasitic leakage paths, a Pareto distribution (*pD*) is employed. It is given as Eq. 11 and 12.

$$V_{\tau\hbar} = \left(V_{\tau\hbar 0} + b\varepsilon\left(\sqrt{|V_{sr} + 2F_{pot}|} - \sqrt{|2F_{pot}|}\right)\right) * \rho D \quad \textbf{(11)}$$

$$\rho D = \frac{\varsigma (V_{\tau\hbar})_{\min}^{\varsigma}}{(V_{\tau\hbar})^{\varsigma+1}}, V_{\tau\hbar} \geq (V_{\tau\hbar})_{\min}, \varsigma > 0 \quad \textbf{(12)}$$

where, ς specifies the shape parameter and $(V_{th})_{\min}$ signifies the minimum value. The threshold

value is approximated for small and large reverse bias based on the first-order Taylor series approximation and is formulated as Eq. 13 and 14.

$$\tilde{V}_{\tau h} \approx V_{\tau h0} + \frac{b\varepsilon}{2}\frac{V_{sr}}{\sqrt{|2F_{pot}|}} \quad \text{(For small reverse bias)} \quad \textbf{(13)}$$

$$\ddot{V}_{\tau h} \approx V_{\tau h0} + b\varepsilon\sqrt{|V_{sr}|} \quad \text{(For large reverse bias)} \quad \textbf{(14)}$$

Where, $(\tilde{V}_{\tau h})$ and $\ddot{V}_{\tau h}$ implies the approximated threshold value for small and large reverse bias, respectively. Eventually, the increased threshold voltage is signified as V_{th}^{*}. After mitigating the photoresist thickness fluctuation, the leakage current with fewer input operands $(L\ell)$ is given for ALU operation.

2.4. Leakage Current and Transistor Failure Detection

Now, by employing BPS-FIS, leakage current and transistor failure while processing $\aleph$ are detected. The working process of BPS-FIS is explained in section 2.2. Here, the fuzzy rule is computed as Eq. 15.

$$\aleph = \begin{cases} \text{if} \quad \partial q < 10\mu A \quad \text{Then} \quad \lambda \\ \text{if} \quad \partial q > 50\mu A \quad \text{Then} \quad \varpi \\ \text{if} \quad \partial q > 100\mu A \quad \text{Then} \quad \eta \end{cases} \quad \textbf{(15)}$$

where, ∂q states Direct Drain Quiescent Current, μA designates microampere, λ implies the normal range, ϖ signifies the presence of leakage current, and η states the failure in the transistor. Finally, the leakage current and transistor failure detection outcomes (LT) are provided as Eq. 16.

$$LT = \{\lambda, \varpi, \eta\} \quad \textbf{(16)}$$

If leakage current and transistor failure are presented, then it is mitigated, which is explained in the following sections.

2.5. Static Power Reduction

Static power reduction is performed by utilizing PD-RBB if leakage current is present (ϖ). Here, the leakage current (ϖ) is reduced. The working process of PD-RBB is explained in section 3.3. Static power reduction outcomes are signified as S_n.

2.6. Transistor Failure Mitigation

If the transistor has a failure (η), then it is mitigated based on Wd-TMR. Generally, TMR superiorly identifies and corrects single-point failures and transient faults. The majority voter ensures that a single faulty module does not impact the final output in TMR. Yet, comparing three outputs before generating a final result generates latency during the majority voting process in TMR. To overcome this problem, the Wishart distribution function is employed in TMR. The proposed Wd-TMR's architectural diagram is shown in Figure 2.

The steps involved in the Wd-TMR are derived as follows, Initially, $\aleph$ is replicated three times. Here, three identical copies of the functions are operated in parallel. It is equated as Eq. 17.

$$\begin{aligned} M_1 &= fn(\aleph) \\ M_2 &= fn(\aleph) \\ M_3 &= fn(\aleph) \end{aligned} \quad \textbf{(17)}$$

were, M_1, M_2, and M_3 implies the three modules and $fn(\aleph)$ represents the function of $\aleph$. Then, a majority voting system is performed. Here, three redundant outputs are compared to select the correct output. The majority vote generates the correct output if any of the three modules fails. Here, to avoid latency during the majority voting process, the Wishart distribution function is used. It is defined as Eq. 18 and 19.

Table 4. Comparative evaluation based on performance metrics.

Methods	Sleep State Leakage Reduction (%)	Total Power Consumption (µW)
Proposed PD-RBB	94	458
RBB	90	598
FBB	88	613
ABB	85	747
SBC	81	801

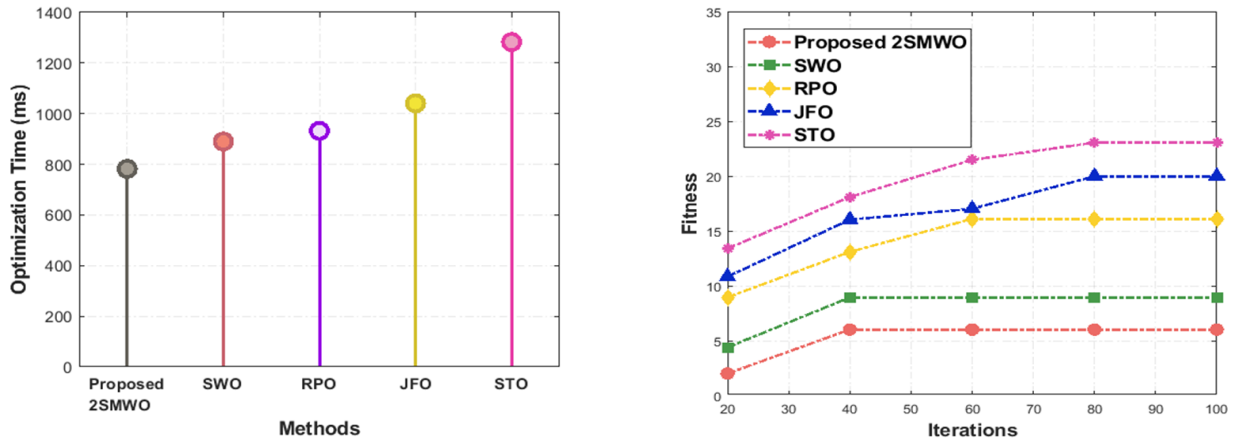

Figure 6. Performance analysis in terms of (a) optimization time and (b) fitness vs iterations.

$$F''' = ((M_1 \wedge M_2) \vee (M_2 \wedge M_3) \vee (M_3 \wedge M_1)) * W \quad \textbf{(18)}$$

$$W = \frac{|\aleph|^{\frac{fd-h-1}{2}} \exp^{-\frac{1}{2}\mathrm{tra}(\Sigma^{-1}\aleph)}}{2^{\frac{fdh}{2}} |\Sigma|^{\frac{fd}{2}} Gm_h\left(\frac{fd}{2}\right)} \quad \textbf{(19)}$$

where, F''' indicates the final outcome of the majority voting process, W states the Wishart distribution function, $\wedge$ exhibits the logical AND operation, $\vee$ represents the logical OR operation, Gm_h indicates the multivariate gamma function, fd denotes the degrees of freedom, Σ outlines the h x h the scale matrix, and tra indicates the trace operator. Then, to make Wd-TMR a reliable fault-tolerant technique, fault tolerance analysis $(\Im\tau)$ is done (Eq. 20).

$$\Im\tau = 3Pb_y^2 - 2Pb_y^3 \quad \textbf{(20)}$$

Here, Pb_y signifies the probability of failure for a single module, $3Pb_y^2$ accounts for cases where two modules fail correctly, and $2Pb_y^3$ corrects overcounting for the case when all modules fail. The transistor failure mitigation outcomes are denoted as ψ_v. The pseudocode for Wd-TMR is depicted as follows Table 2. Next, the irreversibility issue is avoided and is explained in the next section.

2.7. Irreversibility Issue Avoidance

Then, for performing ALU operation, ψ_v, S_n , and $(L\ell)$ are given. Here, by employing the GTRG, the irreversibility issue is avoided. Usually, the TRG has the ability to store more information per digit. Likewise, it reduces the number of needed logic gates and power dissipation. But, error identification in TRG is more difficult than binary logic. To address this problem, Golay Code is utilized instead of processing raw ternary input. Here, the circuit processes Golay-encoded data and retrieves the original data at the end by adding a Golay Decoder. The working process of GTRG is given, Here, the inputs , ψ_v, S_n and ($L\ell$) are considered as H_σ . Initially, by using Golay Code, the $H_\sigma$is encoded for performing error identification. The Golay-encoded data (ξn) is provided as Eq. 21 and 22.

$$\xi n = H_\sigma Gn \quad \textbf{(21)}$$

$$Gn = [Id_{12} \quad Pt] \quad \textbf{(22)}$$

where, Id_{12} states the 12 x 12 identity matrix, Pt states the 12 x 11 parity matrix, and Gn states the generator matrix. In GTRG, each input (ξn) and output takes values from {0, 1, 2}. Likewise, this operation employs modulo-3 arithmetic. The 3x3 GTRG maps three inputs ($\xi n_1, \xi n_2, \xi n_3$) to three outputs (O_1, O_2, O_3) to ensure the unique inverse. Therefore, the simple GTRG is given as Eq. 23.

$$\begin{aligned} O_1 &= \xi n_1 \\ O_2 &= \xi n_1 \oplus \xi n_2 \bmod 3 \\ O_3 &= \xi n_3 \oplus (\xi n_1 \cdot \xi n_2) \bmod 3 \end{aligned} \quad \textbf{(23)}$$

where, $\oplus$ states the ternary addition modulo 3 and $\cdot$ designates the ternary multiplication modulo 3. Then, the irreversibility issue is avoided. For that,

the gate must be invertible and is expressed as Eq. 24.

$$\begin{aligned} O_1 &= \xi n_1 \\ O_2 &= \xi n_2(-)\xi n_1 \text{mod} 3 \\ O_3 &= \xi n_3(-)(\xi n_1 \cdot \xi n_2)\text{mod} 3 \end{aligned} \quad \textbf{(24)}$$

where, (-) denotes the ternary subtraction mod 3. Equation (24) makes sure that the gate is reversible by one-to-one mapping. Finally, the outcomes (Θ) obtained from Equation (24) are decoded by the Golay Code for retrieving the original irreversibility avoided data ($\hat{H}_\sigma$). It is determined as Eq. 25.

$$\text{Syn} = U\hat{H}_\sigma \quad \textbf{(25)}$$

where, Syn implies the computed syndrome and U depicts the 11 x 23 parity-check matrix. If $Syn = 0$, then no error is identified. If $Syn \neq 0$, then the error is corrected. Afterward, the original 12-bit message (Ct_{mes}) is extracted.

2.7.1. Delay Rejection

Now, based on Singer Spider Map Wasp Optimization (2SMWO), the processing delay ($\partial\ell$) that occurs owing to the sequential execution of operations is rejected. Usually, Spider Wasp Optimization (SWO) ensures a balanced approach to exploring new solutions and dynamically adjusts search strategies regarding problem complexity. Yet, SWO may slow down in highly complex or multimodal landscapes, causing a slow convergence rate. To solve this problem, the Singer map is employed in SWO. The working principle of 2SMWO is explained, initially, the population of the spider wasp is initialized with respect to its population. Here, processing delay($\partial\ell$) is considered as the initialized population (R). It is expressed as Eq. 26 and 27.

$$R \xrightarrow{\partial\ell} \begin{bmatrix} R_1 \\ \vdots \\ R_f \\ \vdots \\ R_Z \end{bmatrix}_{Z\times v} = \begin{bmatrix} r_{1,1} & \cdots & r_{1,g} & \cdots & r_{1,v} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ r_{f,1} & \cdots & r_{f,g} & \cdots & r_{f,v} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ r_{Z,1} & \cdots & r_{Z,g} & \cdots & r_{Z,v} \end{bmatrix}_{Z\times v} \quad \textbf{(26)}$$

$$r_{f,g} = \text{low} + \text{rd} \cdot (\text{upper-low}) \quad \textbf{(27)}$$

where, R_f indicates the f^{th} Spider Wasp member in which f = (1, 2, ..., Z), $r_{f,g}$ symbolises the g^{th} dimension in the search space in which $g = (1,2,\ldots,v)$, rd represents the random number, and low and upper depicts the lower and upper bounds, accordingly. Afterward, the fitness function (A) is computed. Here, minimum power loss (min (PL)) is assumed as the fitness function. It is mathematically expressed as Eq. 28.

$$A = \min(PL) * \begin{bmatrix} R_1 \\ \vdots \\ R_f \\ \vdots \\ R_Z \end{bmatrix}_{Z\times 1} = \begin{bmatrix} A(R_1) \\ \vdots \\ A(R_f) \\ \vdots \\ A(R_Z) \end{bmatrix}_{Z\times 1} \quad \textbf{(28)}$$

Searching behavior: Here, the female spider wasp discovers the relevant spiders for feeding the larvae. The new position in the searching stage

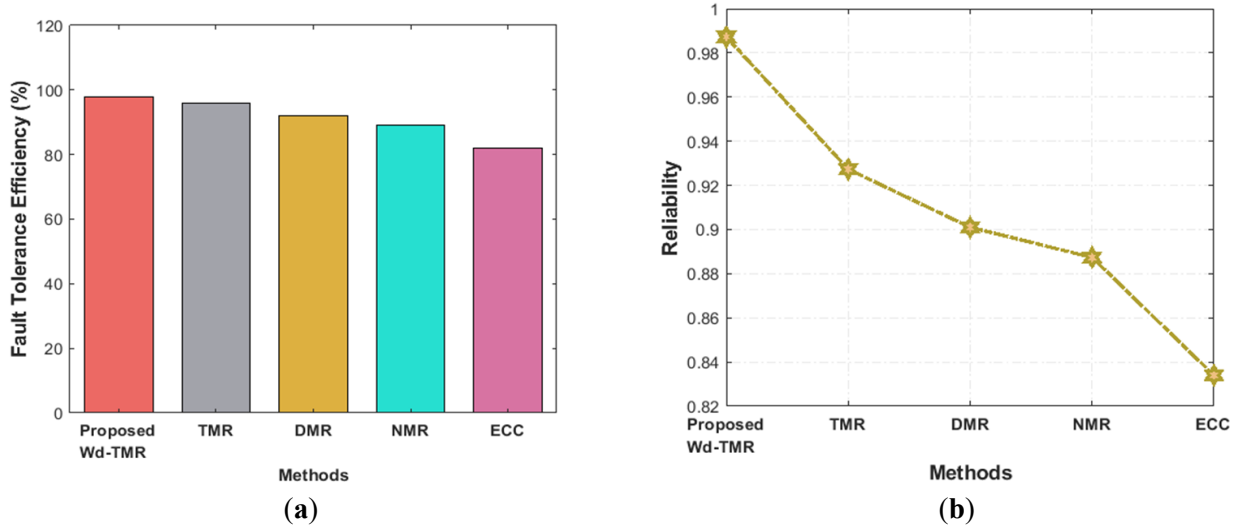

Figure 7. Graphical representation with respect to (a) fault tolerance efficiency and (b) reliability.

Table 5. Computational delay assessment.

Techniques	Computational Delay (ms)
Proposed SbC-NRD	323
NRD	475
RD	578
SRTD	691
GDA	789

(R_f^{newl}) is defined as Eq. 29.

$$R_f^{\text{new }1} = R_f + dr_1 \cdot \left(R_f^{\ddot{\lambda}1} - R_f^{\ddot{\lambda}2}\right) \quad \textbf{(29)}$$

were, dr finds the direction of the search and $R_f^{\ddot{\lambda}1}$ and $R_f^{\ddot{\lambda}2}$ are the two random individuals. Sometimes, the female spider wasp searches the area around the location, where the spider falls, and the new position ($R_f^{newl\,2}$) is given as Eq. 30.

$$R_f^{\text{new }2} = R_f^{\ddot{\lambda}3} + dr_2 \cdot (\text{low} + \text{rd} \cdot (\text{upper} - \text{low})) \quad \textbf{(30)}$$

where, $R_f^{\ddot{\lambda}3}$ depicts the random individual. Next, the search stage (Rs_f) is modeled as Eq. 31.

$$Rs_f = \begin{cases} R_f^{\text{new }1} & rd_2 < rd_3 \\ R_f^{\text{new }2} & \text{otherwise} \end{cases} \quad \textbf{(31)}$$

where, rd_2 and rd_3 are random numbers.

Following and escaping behavior: Here, the female spider wasp hunts the prey after locating it. Likewise, the prey attempts to escape when it is attacked. Here, to avoid the slow convergence rate, the singer map (ςg) function is employed (Eq. 32 – 34).

$$R_f^{\text{new }3} = R_f + Clr \cdot \left|2 \cdot sg \cdot R_f^{\ddot{\lambda}1} - R_f\right| \quad \textbf{(32)}$$

$$sg = \omega\left(sR_f - tR_f^2 + kR_f^3 - b'R_f^4\right) \quad \textbf{(33)}$$

$$R_f^{\text{new }4} = R_f \cdot \Phi \quad \textbf{(34)}$$

where, Φ denotes the D-dimensional vector, $R_f^{new\,3}$ represents the new position in the hunting behavior, Clr signifies the distance control factor, ω signifies the parameter, s, t, k, and b' signifies the constant values, and $R_f^{new\,4}$ signifies the new position during escaping. The following and escaping behavior (Rf_f) is signified as Eq. 35.

$$f_f = \begin{cases} R_f^{\text{new3}} & \varsigma g_2 < \varsigma g_3 \\ R_f^{\text{new4}} & \text{otherwise} \end{cases} \quad \textbf{(35)}$$

Then, the searching behavior and following and escaping behavior are weighed.

Nesting Behavior: Here, the female spider wasp pulls the spider towards the most appropriate area for the female spider wasp. Secondly, the nest is constructed according to the location of the selected female spider wasp (Eq. 36 and 37).

$$R_f^{\text{new }5} = R_f^* + \cos(2\pi) \cdot \left(R_f^* - R_f\right) \quad \textbf{(36)}$$

$$R_f^{\text{new }6} = R_f^{\ddot{x}1} + rd_1 \cdot |f| \cdot \left(R_f^{\ddot{x}1} - R_f\right) + (1 - rd_2) \cdot B \cdot \left(R_f^{\ddot{x}2} - R_f^{\ddot{x}3}\right) \quad \textbf{(37)}$$

where, $R_f^{new\ 5}$ identifies the new position while pulling the spider, $R_f^{new\ 5}$ identifies the new position while constructing the nest, lf illustrates the random number produced by levy flight, and B indicates the binary vector. Then, the nesting behavior is modeled. Finally, the searching, following and escaping, and nesting behaviors are weighed.

Mating Behavior: Here, a female spider wasp creates a new spider wasp. The generated male spider wasp (R_{male}^{new}) is given as Eq. 38.

$$R_{\text{male}}^{\text{new }7} = R_f + \exp \cdot |\ddot{\alpha}| \cdot \tilde{v}_1 + (1 - \exp) \cdot |\ddot{\alpha}_2| \cdot \tilde{v}_2 \quad \textbf{(38)}$$

where, $\ddot{\alpha}$ and $\ddot{\alpha}_2$ indicate random numbers created regarding normal distribution and $\tilde{v}_1$ and $\tilde{v}_2$ are created from the random individuals. The hunting, nesting, and mating behaviors are weighed based on the trade-off rate. Finally, the female spider wasps are optimized for superior outcomes. This updation is continued until convergence is achieved. Thus, the rejected delay is denoted as $\Re_{delay}$.

2.8. Division Module Regulation

If any complex division modules $(\wp, \Omega)$ are present while performing ALU operations, then they are regulated based on the Sigmoid-based Correction Non-Restoring Division (SbC-NRD). Usually, the NRD effectively reduces the number of operations per cycle to boost the execution speed. An extra correction step is required if the remainder becomes negative at the end, thus increasing computation time. To address this problem, the sigmoid-based correction is utilized in NRD. The steps involved in the SbC-NRD are derived. Initially, the quotient (Qt) and remainder (P) are expressed as Eq. 39.

$$Qt = \frac{\wp}{\Omega}, P = \wp \bmod \Omega \quad \textbf{(39)}$$

where, $\wp$ indicates the numerator and Ω specifies the denominator. Next, the (P) is set to zero, and the numerator $(\wp)$ is copied to the quotient register (Qt). For each bit (i.e., from Most Significant Bit (MSB) to Least Significant Bit (LSB)), the left shift and non-restoring subtraction are performed. It is expressed as Eq. 40 and 41.

$$\begin{aligned} &P = P \ll 1 \ \text{(shift by1bit)} \\ &P = P + MSB \ \text{of} \ Qt \\ &Qt = Qt \ll 1 \end{aligned} \quad \textbf{(40)}$$

$$P = P - \Omega \quad \textbf{(41)}$$

Later, the sign of (P) is checked by using Eq. 42.

$$SC = \begin{cases} \text{if} & P \geq 0 \ \text{Then} \ \text{set LSB of} \ Qt \ \text{to 1} \\ \text{if} & P < 0 \quad \text{Then restoration} \end{cases} \quad \textbf{(42)}$$

where, SC designates the sign checking outcomes. Then, for correcting the final remainder $P < 0$, sigmoid-based correction is utilized. It is expressed as Eq. 43.

$$P_{new} = P + \Omega \times \text{sig}(P) \quad \textbf{(43)}$$

where, sig (P) signifies the sigmoid-based correction. Finally, the obtained remainder and quotient are indicated as φ and Ξ.

2.9. Parallel Processing

Similarly, by using CLAA, parallel processing is done to perform multiple operations simultaneously. CLAA excellently removes the sequential carry propagation delay. The steps involved in the CLAA are explained as follows, Initially, the generate and propagate for each bit in the (α,γ) are computed. Here, if both bits are 1, then the carry is generated. If at least one bit is 1, then the carry is propagated. It is written as Eq. 44.

$$\begin{aligned} Q_x &= \alpha_x \cdot \gamma_x \\ K_x &= \alpha_x \oplus \gamma_x \end{aligned} \quad \textbf{(44)}$$

where, x signifies the position of the bit (i.e., 0, 1, 2, 3). Next, all carry values (Y) are estimated rather than waiting for the previous carry. It is formulated as Eq. 45.

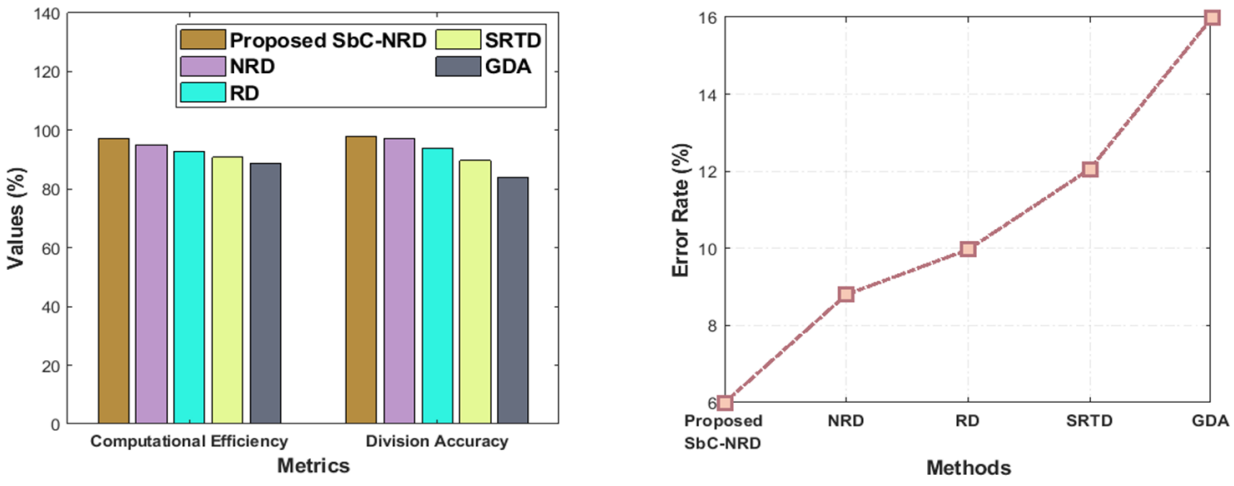

Figure 7. Graphical representation with respect to (a) fault tolerance efficiency and (b) reliability.

Table 6. Comparative evaluation of the proposed and related works.

Ref.	Objective	Techniques	Pros	Cons
Proposed Model	Optimal designing of power-efficient 4-bit ALU	PD-RBB and 2SMWO	The proposed model excellently detected and mitigated the photoresist thickness fluctuations in the designing of 4-bit ALU.	However, it failed to consider the defect density correlation with resist thickness fluctuations in the design of ALU.
[34]	Designing signal-gating aware energy-efficient 8-bit ALU	HFA and ASG	The model consumed less power.	It might increase the overall chip area.
[35]	Implementing ternary ALU	CNTFET	The research enabled compact circuit realization.	The model increased the design complexity.
[36]	Designing of fully non-volatile reconfigurable magnetic ALU	Majority logic technique	The model had better feasibility.	Yet, it had higher switching delays.
[37]	Implementing 64-bit ALU	ATCM	It had reduced time complexity.	The research introduced dynamic power consumption challenges.
[38]	Designing of reversible ALU	QCA	The model increased the energy efficiency.	But, it had increased propagation delay.

$$\begin{aligned} Y_1 &= Q_0 + K_0 Y_0 \\ Y_2 &= Q_1 + K_1 Y_1 \\ Y_3 &= Q_2 + K_2 Y_2 \\ Y_4 &= Q_3 + K_3 Y_3 \end{aligned} \quad (45)$$

The Equation 45 ensures parallel processing, thus making the operation faster. Lastly, the operation is computed. Therefore, an optimal design of ALU is obtained by the proposed methodology.

3. RESULTS AND DISCUSSIONS

Here, the performance and comparative analysis of the proposed and existing techniques are provided to prove the proposed model's effectiveness. Likewise, the proposed model is implemented in the working platform of MATLAB.

3.1. Performance Assessment

Here, the performance of the proposed model is weighed against prevailing techniques to show the dependability of the proposed model.

The graphical representation of the proposed BPS-FIS and prevailing techniques are depicted in Figure 3. Here, the proposed BPS-FIS took less fuzzification and defuzzification times of 674 and 723 ms, respectively. However, the prevailing techniques, such as Sigmoid FIS, Gaussian FIS, Triangular FIS, and Trapezoidal FIS took high average fuzzification and defuzzification times of 893.4 and 1068.25 ms, correspondingly. Here, to improve the detection of photoresist thickness fluctuations, the Bernstein polynomial sigmoid membership function is modified with FIS.

In Figure 4, inference time analysis of the proposed BPS-FIS and conventional methods is shown. Here, for improving the performance of transistor failure and leakage current detection, the BPS-FIS employs the Bernstein polynomial sigmoid membership function. The proposed BPS-FIS took a less inference time of 619ms, whereas the conventional Sigmoid FIS, Gaussian FIS, Triangular FIS, and Trapezoidal FIS took high inference times of 782, 821, 908, and 1023 ms, accordingly. Therefore, the effectiveness of the proposed model is proved.

The proposed BPS-FIS uses the Bernstein polynomial sigmoid membership function for

improving the identification of leakage current and transistor failure presented while processing the input operands. In Table 3, the memory utilization evaluation of the proposed BPS-FIS and prevailing methods is given. Here, low memory of 391 kB is deployed by the proposed BPS-FIS. But, the prevailing Sigmoid FIS, Gaussian FIS, Triangular FIS, and Trapezoidal FIS utilized an average memory of 665.5 kB, which was higher than the proposed technique.

The performance validation of the proposed PD-RBB and existing techniques regarding leakage current reduction and sensitivity is depicted in Figures 5 (a) and (b). Here, the proposed PD-RBB achieved a high leakage current reduction of 92% and sensitivity of 98 mV/V owing to the usage of Pareto distribution. Likewise, the existing RBB, Forward Body Biasing (FBB), Adaptive Body Biasing (ABB), and Substrate Bias Control (SBC) obtained low leakage current reduction of 90%, 88%, 83%, and 80%, respectively. Likewise, the existing techniques attained low sensitivity. Therefore, the proposed model's reliability is proved.

The comparative evaluation of the proposed PD-RBB and conventional techniques with respect to sleep state leakage reduction and total power consumption are depicted in Table 4. Here, to prevent the parasitic leakage paths, the proposed PD -RBB employs Pareto distribution. The proposed PD-RBB achieved a high sleep state leakage reduction of 94% and consumed a low total power of 458 µW. Yet, the conventional RBB and FBB obtained low sleep state leakage reduction of 90% and 88%, respectively. Likewise, the conventional ABB and SBC consumed a maximum total power of 747 and 801 µW, correspondingly. Therefore, the proposed model is better than the conventional techniques.

In Figures 6 (a) and (b), performance analysis regarding optimization time and fitness vs iterations of the proposed 2SMWO and existing techniques is displayed. Here, the proposed 2SMWO took a less optimization time of 782 ms, whereas the existing techniques, such as SWO, Red Panda Optimization (RPO), Jelly Fish Optimization (JFO), and Siberian Tiger Optimization (STO) took high optimization times of 890, 932, 1041, and 1283 ms, correspondingly. Similarly, the proposed 2SMWO achieved a low fitness of 6.01378 µW for 100 number of iterations. Yet, the existing techniques obtained a high average fitness of 17.0145 µW for 100 numbers of iterations. Here, to avoid the slow convergence rate, the Singer Map function is modified with the existing SWO.

The graphical representation of the proposed Wd -TMR and prevailing methods with respect to fault tolerance efficiency and reliability is depicted in Figures 7 (a) and (b). Here, for predicting the errors, the proposed Wd-TMR utilized Wishart distribution, thus effectively mitigating the transistor failure. The proposed Wd-TMR achieved a higher fault tolerance efficiency (98%) and reliability (0.98723) when analogized to the prevailing methods. Similarly, the prevailing methods, such as TMR, Double Modular Redundancy (DMR), N-Modular Redundancy (NMR), and Error Correction Code (ECC) obtained a low average fault tolerance efficiency (89.75%) and reliability (0.88741).

The computational delay assessment of the proposed SbC-NRD and existing methods are depicted in Table 5. Here, owing to the inclusion of sigmoid-based correction, the proposed SbC-NRD achieved a low computational delay of 323 ms, whereas the existing techniques, such as NRD, Restoring Division (RD), SRT Division (SRTD), and Goldschmidt's Division Algorithm (GDA) attained high computational delay of 475, 578, 691, and 789 ms, respectively. Therefore, the efficacy of the proposed model is demonstrated.

In Figures 8 (a) and (b), performance validation of the proposed SbC-NRD and conventional techniques is displayed. Here, the proposed SbC-NRD achieved a high computational efficiency and division accuracy of 97.36% and 98.0129%, correspondingly. Likewise, the proposed SbC-NRD attained a low error rate of 6%. Yet, the conventional RD and GDA obtained low computational efficiency of 92.8475% and 88.743%, respectively. Similarly, the conventional NRD and SRTD attained high error rates of 8.8% and 12.06%, respectively. Likewise, the conventional techniques attained poor division accuracy. Here, to diminish the computation time in some cases, the proposed SbC-NRD employs sigmoid-based correction.

3.2. Comparative Analysis

Here, a comparative analysis of the proposed and related works is provided. The comparative evaluation of the proposed along with related works are depicted in Table 6. Here, in the designing of 4-Bit ALU, the proposed PD-RBB and 2SMWO superiorly detected and mitigated the photoresist thickness fluctuations. Yet, the overall chip area is augmented by the existing Hybrid Full Adder (HFA) and Adaptive Signal Gating (ASG). Likewise, the conventional Carbon Nanotube Field Effect Transistor (CNTFET) had design complexity. Similarly, the prevailing Majority logic technique and QCA introduced high switching delay and propagation delay, respectively. Likewise, the existing Automated Timing Characterization Methodology (ATCM) had dynamic power consumption challenges. Therefore, the proposed model is better than the existing works.

4. CONCLUSIONS

An effective framework, known as PD-RBB and 2SMWO, is introduced for the optimal design of a power-efficient 4-bit ALU, with a focus on mitigating photoresist thickness fluctuations. This methodology encompasses crucial processes such as detecting and mitigating photoresist thickness fluctuations, identifying leakage current and transistor failures, reducing static power, addressing transistor failures, avoiding irreversibility issues, rejecting delays, regulating the division module, and enabling parallel processing. To address photoresist thickness fluctuations, the proposed PD-RBB achieved a significant leakage current reduction of 92% and a sensitivity of 98 mV/V. Similarly, the proposed Wd-TMR demonstrated a high fault tolerance efficiency of 98%, underscoring the model's effectiveness. Additionally, the proposed BPS-FIS detected leakage current and transistor failures with a reduced inference time of 619 ms, thereby enhancing the model's reliability and trustworthiness. However, the proposed model did not account for the correlation between defect density and resist thickness fluctuations in the ALU design, despite effectively detecting and mitigating these fluctuations in the 4-bit ALU design. Consequently, future advancements will focus on developing techniques to address the defect density correlation with resist thickness fluctuations for further optimization in ALU design.

AUTHOR INFORMATION

Corresponding Author

Shylaja Veerabhadraiah — Department of ECE, CMR University, Bangalore-562149 (India); Department of ECE, Bangalore Institute of Technology, Bangalore-560004 (India);
orcid.org/0009-0001-3911-8060
Email: shylaja.v@cmr.edu.in

Authors

N Kannan — Department of ECE, CMR University, Bangalore-562149 (India);
orcid.org/0000-0001-7688-7662

T Y Satheesha — School of Computer Science and Engineering, REVA University, Bengaluru-560064 (India);
orcid.org/0000-0003-3869-0839

Author Contributions

Conceptualization, Methodology, Software, Writing – Original Draft Preparation: S. V.; Validation, Formal Analysis, Investigation, Writing – Review & Editing: N. K. and T. Y. S.

Conflicts of Interest

The authors declare no conflict of interest.

ACKNOWLEDGEMENT

The authors acknowledge the support from CMR University, BIT, Bengaluru, REVA University, India for the facilities provided to carry out the research.

DECLARATION OF GENERATIVE AI

Not applicable.

REFERENCES

[1] S. Yu, J. Fu, Z. Lin, C. Peng, and X. Wu. (2022). "In-Memory Calculation with Embedded Arithmetic and Logic Units for

Deep Neural Network". *Electronics Letters.* **58** (17): 639-641. 10.1049/ell2.12549.

[2] N. A. Adela and D. K. Alamen. (2022). "Design and Implementation of Single Precision Floating-Point Arithmetic Logic Unit for RISC Processor on FPGA". University of Tripoli. 10.1109/MI-STA57575.2023.10169623

[3] Q. Xie, M. Yuan, J. Niroula, B. Sikder, S. Luo, K. Fu, N. S. Rajput, A. B. Pranta, P. Yadav, Y. Zhao, N. Chowdhury, and T. Palacios. (2023). "Towards DTCO in High Temperature GaN-on-Si Technology: Arithmetic Logic Unit at 300 °C and CAD Framework up to 500 °C". 1-3. 10.23919/VLSITechnologyandCir57934.2023.10185364

[4] J. A. Khatokar, N. Vinay, A. S. Bale, M. A. Nayana, R. Harini, V. S. Reddy, N. Soundarya, T. Y. Satheesha, and A. S. Huddar. (2021). "A Study on Improved Methods in Micro-Electromechanical Systems Technology". *Materials Today: Proceedings.* **43** : 3784-3790. 10.1016/j.matpr.2020.10.993.

[5] A. S. Bale, S. Saranya, S. B. C, T. T. S, T. Y. Satheesha, and S. Ponnuru. (2024). "Revolutionizing Chip Design Using Generative AI in Semiconductors: A Case Study Approach". 1-6. 10.1109/ICACCM61117.2024.11059162

[6] P. Srivastava, R. Yadav, and R. Srivastava. (2021). "Robust Circuit Implementation of 4-Bit 4-Tube CNFET-Based ALU at 16-nm Technology Node". *Analog Integrated Circuits and Signal Processing.* **109** (1): 127-134. 10.1007/s10470-021-01825-y.

[7] N. Ravi and M. B. Veena. (2022). "Design of an Efficient ALU Blocks in Quantum Dot Cellular Automata (QCA)". *Global Transitions Proceedings.* **3** (1): 157-168. 10.1016/j.gltp.2022.03.004.

[8] K. Singh and S. Mandal. (2021). "Z-Domain Mathematical Modeling and Performance Analysis of Ripple Ring Resonator (RRR) with Design of All-Optical Arithmetic Logic Unit (ALU)". *Optik.* **232** : 166532. 10.1016/j.ijleo.2021.166532.

[9] H. R. Archana, T. Sanjana, H. T. Bhavana, and S. V. Sunil. (2021). "System Verification and Analysis of ALU for RISC Processor". 1785-1789. 10.1109/ICACCS51430.2021.9442045

[10] X. Fei. (2025). "Optimized Design and Applications of Arithmetic Logic Units: Addressing Power Efficiency and Performance in Diverse Computing Applications". *Advances in Computer Engineering.* **128** (1): 132-137. 10.54254/2755-2721/2025.20216.

[11] V. Sharmila, P. Mallikarjuna, S. S. Reddy, and P. Arun. (2025). "Optimized 4-Bit ALU Architecture for Enhanced Speed". 966-971. 10.1109/ESIC64052.2025.10962763

[12] V. Priyanka, N. S. Reddy, G. Jeevana, and M. A. Arab. (2024). "Design of Arithmetic Logic Unit Using Reversible Logic Gates". 1 -6. 10.1109/WCONF61366.2024.10692088

[13] A. B and S. S. R. N. (2024). "Design of High -Speed Low-Power 4-Bit ALU Using CNTFET". *SSRG International Journal of Electrical and Electronics Engineering.* **11** (8): 36-49. 10.14445/23488379/IJEEE-V11I8P104.

[14] K. S. Tiwari. (2025). "Design of Generic Vedic ALU Using Reversible Logic". *Memories - Materials, Devices, Circuits and Systems.* **9** : 100121. 10.1016/j.memori.2025.100121.

[15] T. Mendez and S. G. Nayak. (2023). "Design of a Low-Power Computational Unit Using a Pipelined Vedic Multiplier". 1-7. 10.1109/ICONAT57137.2023.10080520

[16] K. Swetha, K. L. Krishna, J. V. S. Sowmya, D. S. Reddy, G. Pravallika, and G. A. Kumar. (2021). "Area-Efficient Multilayer Arithmetic Logic Unit Implementation in Quantum-Dot Cellular Automata". 1-6. 10.1109/ICICV50876.2021.9388584

[17] H. S. Poornima, P. Janani, and A. Manasa. (2024). "Next-Gen 32-Bit Arithmetic Logic Unit: Harnessing Cellular Automata Using Quantum Dots (QCA) Technology". 1-5. 10.1109/ICRISST59181.2024.10921840

[18] A. Chauhan, K. K. Saini, N. Rajput, and R. Domah. (2021). "Implementation of High Performance 4-Bit ALU Using Dual Mode

Pass Transistor Logic". 1-5. 10.1109/CONIT51480.2021.9498553

[19] R. Faraji and A. Rezai. (2024). "Design of a Multilayer Reversible ALU in QCA Technology". *The Journal of Supercomputing.* **80** (12): 17135-17158. 10.1007/s11227-024-06102-z.

[20] A. S. Bale, J. A. Khatokar, S. Singh, G. Bharath, M. S. Kiran Mohan, S. V. Reddy, T. Y. Satheesha, and S. A. Huddar. (2021). "Nanosciences Fostering Cross-Domain Engineering Applications". *Materials Today: Proceedings.* **43** : 3428-3431. 10.1016/j.matpr.2020.09.076.

[21] R. Roy and S. Sarkar. (2023). "Physical Design and Verification of 3D Reversible ALU by QCA Technology". *Materials Today: Proceedings.* **80** : 1-12. 10.1016/j.matpr.2021.05.621.

[22] S. Kundu, M. Hossain, and S. Mandal. (2023). "Modeling of Silicon Microring Resonator-Based Programmable Logic Device for Various Arithmetic and Logic Operation in Z-Domain". *Optical and Quantum Electronics.* **55** (2). 10.1007/s11082-022-04378-0.

[23] S. Çakmak, M. Kurt, and A. Gençten. (2023). "QFT-Based Quantum Arithmetic Logic Unit on IBM Quantum Computer". *arXiv.*

[24] B. Safaiezadeh, E. Mahdipour, M. Haghparast, S. Sayedsalehi, and M. Hosseinzadeh. (2021). "Novel Design and Simulation of Reversible ALU in Quantum Dot Cellular Automata". *The Journal of Supercomputing.* **78** (1): 1-15. 10.1007/s11227-021-03860-y.

[25] M. Liu, Z. Tang, H. You, C. Li, G. Guo, Z. Wang, L. Zhang, X. Liu, Y. Wang, Y. Dai, G. Bai, and X. Lin. (2025). "An Efficient Machine Learning-Enhanced DTCO Framework for Low-Power and High-Performance Circuit Design". *Journal of Information and Intelligence.* 1-17. 10.1016/j.jiixd.2025.02.001.

[26] R. M. Jujjavarapu and A. Poulose. (2022). "Verilog Design, Synthesis, and Netlisting of IoT-Based Arithmetic Logic and Compression Unit for 32 nm HVT Cells". *Signals.* **3** : 620-641. 10.3390/signals3030038.

[27] N. B. Jadhav, R. Bhagat, S. Paranjpe, S. Dahitule, S. Madke, and S. Jadhav. (2021). "Micro-Ring Resonator-Based All-Optical Arithmetic and Logical Unit". *Optik.* **244** : 167622. 10.1016/j.ijleo.2021.167622.

[28] S. Razmkhah and A. Bozbey. (2023). "Efficient Superconductor Arithmetic Logic Unit for Ultra-Fast Computing". *arXiv.* 1-11.

[29] R. H. Vanlalchaka, R. Maity, and N. P. Maity. (2023). "A Low-Power Design Using FinFET-Based Adiabatic Switching Principle: Application to 16-Bit Arithmetic Logic Unit". *Ain Shams Engineering Journal.* **14** (4). 10.1016/j.asej.2022.101948.

[30] R. Roy, S. Sarkar, and S. Dhar. (2021). "Design and Testing of a Reversible ALU by Quantum Cells Automata Electro-Spin Technology". *The Journal of Supercomputing.* **77** (12): 13601-13628. 10.1007/s11227-021-03767-8.

[31] N. Subbulakshmi, R. Sravanthi, M. S. Stalin, T. Swapna, T. Rajesh, and Y. Greeshma. (2023). "ALUSGDI: Low-Power Arithmetic Logic Unit-Based Sliced Processor Using GDI and MGDI". *Measurement: Sensors.* **28** : 100842. 10.1016/j.measen.2023.100842.

[32] A. Abdi and S. Shahoveisi. (2022). "FT-EALU: Fault-Tolerant Arithmetic and Logic Unit for Critical Embedded and Real-Time Systems". *The Journal of Supercomputing.* 10.1007/s11227-022-04698-8.

[33] A. S. Bale, S. Saranya, T. S. Arulananth, V. Y. Shanthakumar, T. Y. Satheesha, and G. Murtugudde. (2024). "Improving the 8T SRAM Cell Design for Fault Tolerance and Power Efficiency". 1-6. 10.1109/ACOIT62457.2024.10939847

[34] D. Ajitha and M. Chandra Sekhar Reddy. (2021). "Intelligent Signal Gating-Aware Energy-Efficient 8-Bit FinFET Arithmetic and Logic Unit". *Circuits, Systems, and Signal Processing.* 1-22. 10.1007/s00034-021-01775-w.

[35] F. Zahoor, F. A. Hussin, F. A. Khanday, M. R. Ahmad, and I. M. Nawi. (2021). "Ternary

Arithmetic Logic Unit Design Utilizing Carbon Nanotube Field Effect Transistor and Resistive Random Access Memory". *Micromachines.* **12** (11). 10.3390/mi12111288.

[36] S. Rangaprasad and V. K. Joshi. (2023). "A Fully Non-Volatile Reconfigurable Magnetic Arithmetic Logic Unit Based on Majority Logic". *IEEE Access.* **11** : 118944-118961. 10.1109/ACCESS.2023.3327261.

[37] A. Inamdar, J. Ravi, S. Miller, S. S. Meher, M. E. Celik, and D. Gupta. (2021). "Design of 64-Bit Arithmetic Logic Unit Using Improved Timing Characterization Methodology for RSFQ Cell Library". *IEEE Transactions on Applied Superconductivity.* **31** (5): 1-7. 10.1109/TASC.2021.3061639.

[38] M. Alharbi, G. Edwards, and R. Stocker. (2023). "Reversible Quantum-Dot Cellular Automata-Based Arithmetic Logic Unit". *Nanomaterials.* **13** (17). 10.3390/nano13172445.