Hindawi
Journal of Nanomaterials
Volume 2022, Article ID 3487853, 17 pages
https://doi.org/10.1155/2022/3487853

Research Article

Design and Implementation of ALU Using Graphene Nanoribbon Field-Effect Transistor and Fin Field-Effect Transistor

D. Rebecca Florance,[1] **B. Prabhakar**,[2] **and Manoj Kumar Mishra**[3]

[1]*Research Scholar, Department of Electronics and Communication Engineering, JNTU College of Engineering, Jagtial, Telangana, India*
[2]*Associate Professor, Department of Electronics and Communication Engineering, JNTU College of Engineering, Jagtial, Telangana, India*
[3]*Professor, Salale University, Fitche, Ethiopia*

Correspondence should be addressed to D. Rebecca Florance; rebeccaflorance25@gmail.com
and Manoj Kumar Mishra; mkmishra@slu.edu.et

Received 18 May 2022; Revised 7 June 2022; Accepted 16 June 2022; Published 1 July 2022

Academic Editor: Samson Jerold Samuel Chelladurai

Arithmetic and logical unit (ALU) are the core operational programmable logic block in microprocessors, microcontrollers, and real-time-integrated circuits. The conventional ALUs were developed using complementary metal oxide semiconductor (CMOS) technology, which resulted in excessive power consumptions, path delays, and number of transistors. Therefore, this article focuses on the design and development of hybrid delay-controlled reconfigurable ALU (DCR-ALU) using field-effect transistor (FinFET) and graphene nanoribbon field-effect transistor (GnrFET) technologies. Initially, a novel carry output predictable full adder (COPFA) and carry input selectable full adders (CISFA) are developed using multiplexer selection logic; then, delay-controlled hybrid adder (DCHA) and delay-controlled hybrid subtractor (DCHS) are developed using these full adders. In addition, a unified delay-controlled hybrid adder and subtractor (DCHAS) is developed by combining these DCHA and DCHS. Further, a delay controller array multiplier (DCAM) also developed using DCHA modules. Finally, DCR-ALU is developed by adopting the DCHAS, DCAM, and logical operations. The obtained simulation results shows that the proposed nanotechnology-based models outperformed the conventional adders and subtractors in terms of area, power, and delay reduction.

1. Introduction

Recent years, the digital circuitry needs the decrease of area and power by optimizing the time with improving performance in speed. The fundamental digital circuits consist of adders and multipliers as their building blocks. Further, these are generally utilized as various hardware blocks in DSP and ALU frameworks [1]. In DSP, it incorporates refinement and digital filtering like digital communications, spectral analysis, FFT, and DFT. In view of the activity of adder, the power utilization execution relies upon it. In VLSI industry [2], the segments of convenient electronics have quick development for the circuits of low control math. The proficient structure of the power and area utilization with an information path high speed model is a research area to decorate. Through the adder, the speed is constrained according to the time necessity for carry propagation in the digital adders [3]. The situations of every bit are created successively in a steady progression in an adder basic and proliferate a carry into the following position. Many high-performance systems, such as microprocessors, DSP, digital communications, and spectrum analysis, rely on ALUs. The performance of a system is governed by the multiplier's performance, as it is the system's most space-consuming and slowest component. As a result, maximizing the multiplier's [4] speed and area is a crucial design consideration. Since the area and speed are opposing restrictions, several methods and designs have been created to provide a better trade-off

between the two. Furthermore, due to the scenario of many DSP applications are aimed at portable, both battery-operated systems and power dissipation are also a significant design limitation. The ALUs are the major building block in every IC, which is utilized widely in various VLSI systems such as specific application of microprocessors and DSP architectures [5]. Further, the ALU performs the logical operations, additions, and multiplications, respectively. The key task of addition is adding the two binary numbers; it is the basis of several useful functions such as division, multiplication, and subtraction and addresses calculation [6], etc. In critical path, adder is a part of most of these systems which regulates the systems' complete performance. The performance enhancing of the ALU cell (the binary adder building block) is an important goal. Recently, VLSI systems are developing with low power and have occurred as extremely in request due to the development of technologies in computation and communication of mobile. But there is a restricted power offered for the mobile systems. Full adder (FA) is the simple building blocks of various digital VLSI circuits. Some modifications have been done in concern to its structure from its invention. In submicron [7] and deep submicron technologies, the motivation of the VLSI designers is to reduce the static and dynamic power consumption in VLSI ICs. When aiming about the energy efficient arithmetic circuits inside the processors, there is a need for obtaining an energy efficient ALU architecture. To design an energy efficient ALU, which is to be utilized in higher-order adders, one of the necessary parameters is the reduction in total power consumption. This will increase the switching of the logic level signal in all the internal as well as external loads of each subcircuits in arithmetic units which leads to large dynamic power consumption [8]. Hence, the reduction in dynamic power consumption is the most significant factor in improving the energy efficiency in arithmetic building blocks. The rise in leakage current inside the discrete devices fabricated in VLSI IC leads to considerable increase in static power consumption [9]. In order to design an energy efficient ALU cell, one has to concentrate in achieving the reduction in both static and dynamic power consumption. This is evident with the following discussion on power dissipation in CMOS-based circuits.

Recently, FinFET, GnrFET, carbon nanotube FET (CntFET), and RibbonFET are widely using in variety applications for area, power, and delay improvement performance. These advanced FET models are developed by modifying the processing parameters (such as size, thickness, temperature, current, voltage levels, materials of fabrication, gate oxidation, source, and drain divisions, introducing extra terminals) of different layers. Thus, this work considered the FinFET and GnrFET models for development of various arithmetical operations. The major contributions of this work are illustrated as follows:

Initially, modified full adder (MFA) is developed with 10 transistors using multiplexer selection logics. Then, CISFA and COPFA are developed by changing the carry input and carry out operations of MFA.

Further, 4-bit DCHA, DCHS, and DCHAS are developed using MFA, CISFA, and COPFA modules, where DCHS is developed by using two complementation addition properties with DCHA as building block.

Then, 4-bit DCHA, DCHS, and DCHAS blocks are extended with reconfigurable properties, which formed the N-bit implementations of delay-controlled reconfigurable hybrid adder (DCRHA), delay-controlled reconfigurable hybrid subtractor (DCRHS), and delay-controlled reconfigurable hybrid adder subtractor (DCRHAS).

In addition, 4-bit DCAM is developed by using DCHA modules and extended with N-bit delay-controlled reconfigurable array (DCRAM).

Finally. DCR-ALU is developed by using DCRHAS, DCRAM, and logical operations using FinFET and GnrFET technologies. The simulation results show that the proposed models are outperformed in terms of area, power, and delay reduction as compared to conventional adders and subtractors.

Rest of the paper is contributed as follows: Section 2 deals with the related work with problems. Section 3 deals with preliminaries of the proposed models. Section 4 deals with the proposed method. Section 5 deals with simulation results and discussions. Section 6 concludes with future scope.

2. Related Work

This section gives the detailed survey of various adders developed over the past few years. The survey is mainly focusing on analysis of hybrid adders. In [9], the authors developed the approximate adders exhibits with slow speed of compact design but carry look ahead performed faster and consumed more area. Radiation-hardened majority-based magnetic full adder (RHMFA) [10] is designed and performed in this addition process of increasing speed. In analysis of approximate adders, utilizing FinFET shows that the speed of the approximate adders is almost double than the conventional RCA.

In [11], the authors developed the Hybrid FinFET Full adder (HFFA) and Hybrid FinFET adder (HFA) design for video and image processing applications. It exhibits the gate depth in the structure of adder and implemented dual RCA with the input 0 and 1, respectively. Further, Imprecise Minority-Based CNFET Full Adder (IMC-FA) and Imprecise Minority-Based CNFET Adder (IMCA) work with efficient and simple process of significantly modification of gate level and the parameter reduction in conventional HFFA [12]. Further, IMCA is an optimization process in the constraints of VLSI designs, respectively. To overcome the area, power, and delay consumption issues in 4-bit, 8-bit CNTFET Ternary Full adder (CTFA) and CNTFET Ternary adder (CTA) without utilizing Mux and utilizing approximate adders is developed in [13] and simulations resulted in better performance for modified designs as compared to state of art approaches. The ternary half adder and 1-trit multiplier architecture have been developed [14] and evaluated the performances of the design in terms of area, power, and delay.

In [15], the authors developed the PTL-based subtractor (PTLS) with the gate-level modifications, which required less

gates to perform the operation in the proposed work. It provides area reduction and the total power. The result analysis shows the better performances of the circuit and faster than the others. In this way, MTCMOS subtractor (MTCMOSS) [16] makes efficient and simple way to process the VLSI hardware implementations. The mobile industry is growing rapidly not only because of arithmetic unit but also with the arithmetic units of less power and area. A simple and efficient modification of gate level makes the reduction in power, area, and delay in Trit Unbalanced Ternary Subtractor (TUTS) [17]. Based on the modifications of CSLA, the performances were compared with other adders. By the BEC modification instead of MFA [18] chain, the logic converter provides the circuit with slight changes of delay. The fast process performances of TUTS are utilized for arithmetic functions in data processing processors.

In [19], the authors developed the gating-aware energy adders and subtractors (GAEAS) for power utilization that has transformation. For the reduction of circuit consumption, BEC is utilized in the modified quantum adders instead of CSLA and RCA with increasing the delay slightly. In [20], the authors proposed the low error efficient approximate FinFET-based Hybrid Adder/Subtractor Circuit (FHAS), which is linear proportional of N delay performed with N -bit, so highest delay process is performed by these adders. Normally, it provides faster results with more delay process than the other adders. It provided because of large number of logic gates and fan-in. In [21], the authors proposed CNTFET-based adder and subtractor (CNTFET-AS), which exhibits high speed with compact design but consumes more area. Also, CNTFET is accessible with low-power multipliers. The simulation results shown that it resulted in better performance as compared to the FinFET-based adders [22]. In [23], the authors developed the multioperative reversible adder and subtractor (MRAS), which can be used in the design of high-performance modules [24] like multiple bit adders, multipliers, multiplexers, subtractors, comparators, and registers. The advancement in fabrication nanotechnology [25] with the shrinking device sizes has allowed for placement of nearly two billion transistors on Intel's advanced processor.

In [26], the authors developed the 14 nm FinFET technology-based 8-bit Dadda multiplier (FDM) using approximate 4 : 2 compressors. The carry propagate adders are designed with a 4 : 2 compressor to reduce the height of the partial product rows. Further, CNTFET-based vedic mathematics processor (CVMP) [27] is developed to reduce the power complexities presented in the FDM, which performs additions, multipliers, multiply and accumulations, and DWT operations. The maximum height of the partial product was reduced by one unit. This was achieved through the pipelining of the multiplier by optimization of the array reduction stage. In [28], the authors developed the Majority-Based Imprecise Multiplier (MBIM) with 7 nm FinFET. The MBIM adopted the advanced quantum dot cellular automata for generalization of majority logic formulations. The partial product reductions were achieved through the hybrid compressors with majority logic gates. Further, FinFET-based Energy Efficient Pass Transistor Adiabatic Logic (EEPAL) [29] was introduced to develop the Carry Save Multiplier. But this method consumes the high computational complexity in terms of delay. In [30], the authors developed the area-efficient FinFET-based approximate multiplier methodology with low voltage with least current consumption circuits. These were implemented for a current feed operational EETM and the current conveyor. Then, FinFET-based discrete wavelet transform (FDWT) [31] was developed by using reversable logic gates. The comparison has revealed that this method of FinFET design could reduce the area by almost 50% of the smallest area.

In [32], the authors developed high-speed 8-bit ALU (HS-ALU) using 18 nm FinFET technology which is based on the kogge stone additions and Dadda multiplier. It is one of the most vividly encountered operations across a majority of the real-time DSP solutions but consumed higher power. Further, N-bit ALU [33] is developed by using 18 nm FinFET technology, which mainly implies function of additions and designs relied the considerable speed as it is one of the performance metrics and digital circuits with high-speed performance has always gained significant importance. This method is suffering with the high computational complexity. In [34], the authors developed the 32 nm-FinFET-based 4-bit ALU (FALU-32) developed using array multipliers, carry save additions. The power consumption in the case of the FALU-32 circuit is to be mitigated either for two distinct reasons as to mitigate the heat dissipation to support significant volumes of functions that are integral to handling the IC. Further, sub-10 nm gate length-based Schottky-barrier FinFET (SB-FinFET) [35] was developed to reduce the power consumption problems presented in basic FinFET technology. Further, 1-bit ALU is developed by using SB-FinFET technology; however, this method is suffering with the fabrication issues. Then, hybrid pin-transfer torque-magnetic tunnel junction (STT-MTJ) and hybrid tunnel FET (TFET) [36] are developed to reduce the mutual problems presented in conventional GnrFET, FinFET, and CNTFET technologies. Then, 1-bit magnetic ALU (MALU) is developed by using STT-MTJ and TFET technologies, where STT-MTJ resulted in reduced area and delay metrics.

3. Preliminaries

The design and development of full adders are critical because they are the fundamental building blocks of all integrated circuits. Therefore, the design of full adders must be done with care because of their importance. Basic CMOS technology-based full adder is utilizing the greater energy, area, delay, and power consumption. So, the GnrFET and FinFET technologies serve as an alternate technology to the basic CMOS technology, which is characterized by low resource consumption. This work implemented novel three different types of full adders for performing addition and subtraction operations. They are MFA with 10 transistors, COPFA with 12 transistors, and CISFA with 12 transistors, respectively. All the three designs contain equal number of PMOS and NMOS transistors, which allows for the introduction of the equilibrium state in three models, allowing

for the control and synchronization of the unbiased electron-hole pair. Furthermore, this equilibrium condition ensures that the ideal power consumption is maintained while simultaneously improving energy efficiency.

3.1. MFA. Figure 1 shows the block diagram of MFA, which contains two multiplexers for sum and carry out generations. Here, C_{in} is applied as selection input to the MUX21, and it generates the sum output based on data selection logic between XOR and XNOR outcomes. Finally, sum output of MFA is generated as follows:

$$S = \begin{cases} A \oplus B \oplus C_{\text{in}} = A \oplus B, C_{\text{in}} = 0, \\ A \oplus B \oplus C_{\text{in}} = A \odot B, C_{\text{in}} = 1. \end{cases} \quad (1)$$

Further, XNOR of inputs A, B is applied as selection input to the MUX21-A, and it generates the carry output based on data selection logic. Finally, carry output of MFA is generated as follows:

$$C_{\text{out}} = \begin{cases} C_{\text{in}}, (A \odot B) = 0, \\ B, (A \odot B) = 1. \end{cases} \quad (2)$$

Figure 2 shows the FinFET-based transistor level architecture of the proposed MFA, which contains five PMOS-FinFETs and five NMOS-FinFETs. Initially, input B is applied to the gate terminal of P1 and N1 FinFETs, which functions as inverter and generates the outcome $\bar{B}$. Further, $\bar{B}$ and input A are applied to gate terminal of P2 and N2 FinFETs, which functions as XOR gate and generates output as $A \oplus B$.. Further, $A \oplus B$ is also applied to the gate terminal of P3 and N3 FinFETs, which act as inverter and generates the output as $A \odot B$. Then, C_{in} is applied as input to gate terminal to the gate terminal of P4 and N4 FinFETs, which acts as MUX21 with $A \oplus B$ and $A \odot B$ as data inputs, so sum output is generated. Finally, $A \odot B$ was generated from MUX21 applied as input to gate terminal of P5 and N5 FinFETs, which acts as MUX21, so carry output is generated.

3.2. CISFA. The major advantage of CISFA is that it has the potential capability to select the carry inputs. Generally, the complexity of addition process purely depends on carry generation and forwarding nature. So, the fast operation of full adders can be effectively performed by minimizing the computations through different carry inputs. The CISFA module considers the selectable extra carry input (C_{in_s}), which is different from original carry input (C_{in}).

Figure 3 shows the block diagram of CISFA, which is implemented from the fundamentals of MFA by introducing the extra OR gate. Initially, C_{in_s} and C_{in} applied as input to MUX21, which is used to select the two carry inputs. Here, if both inputs A and B are equal, then MUX21-A generates the output as C_{in_s}, and if inputs A and B are not equal, then

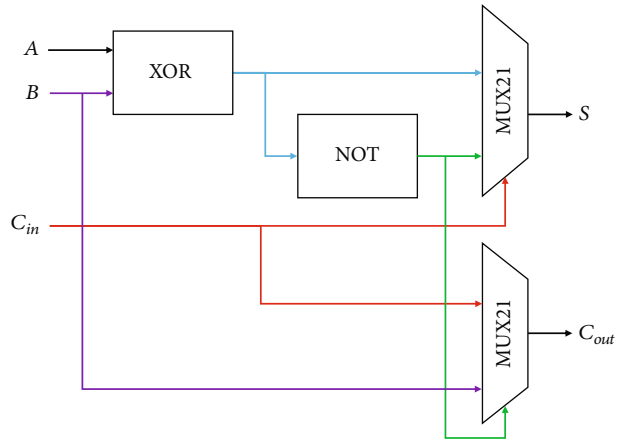

Figure 1: Block diagram of MFA.

MUX21-A generates the output as C_{in}. The operation of MUX21-A is illustrated as follows:

$$\text{OUT}_A = \begin{cases} C_{\text{in}}, A = B, \\ C_{\text{in}_s}, A \neq B. \end{cases} \quad (3)$$

Then, output of MUX21-A (OUT_A) is applied as selection input to the MUX21-A, which generates the sum output based on XOR-XNOR selection logic. Finally, sum of CISFA is generated as follows:

$$S = \begin{cases} A \oplus B, \text{OUT}_A = 0, \\ A \odot B, \text{OUT}_A = 1. \end{cases} \quad (4)$$

Further, XNOR of inputs A, B is applied as selection input to the MUX21-A, and it generates the carry output based on data selection logic. Finally, carry output of CISFA is generated as follows:

$$C_{\text{out}} = \begin{cases} \text{OUT}_A, (A \odot B) = 0, \\ B, (A \odot B) = 1. \end{cases} \quad (5)$$

Figure 4 shows the FinFET layout of CISFA, which contains 6 number of PMOS-FinFETs and 6 number of NMOS-FinFETs. Initially, input B is applied to the gate terminal of P1 and N1 FinFETs, which functions as inverter and generates the outcome $\bar{B}$. Further, $\bar{B}$ and input A are applied to gate terminal of P2 and N2 FinFETs, which functions as XOR gate and generates output as $A \oplus B$. In addition, $A \oplus B$ is applied to gate terminal of P3 and N3 FinFETs, which acts as MUX21-A with C_{in_s} and C_{in} as data inputs. Further, $A \oplus B$ is also applied to the gate terminal of P4 and N4 FinFETs, which act as inverter and generates the output as $A \odot B$. Then, OUT_A generated from MUX21-A is applied as input to gate terminal of P5 and N5 FinFETs, which acts as MUX21-B with $A \oplus B$ and $A \odot B$ as data inputs, so sum output is generated. Finally, $A \odot B$ generated from MUX21-B is applied as input to gate terminal of P6

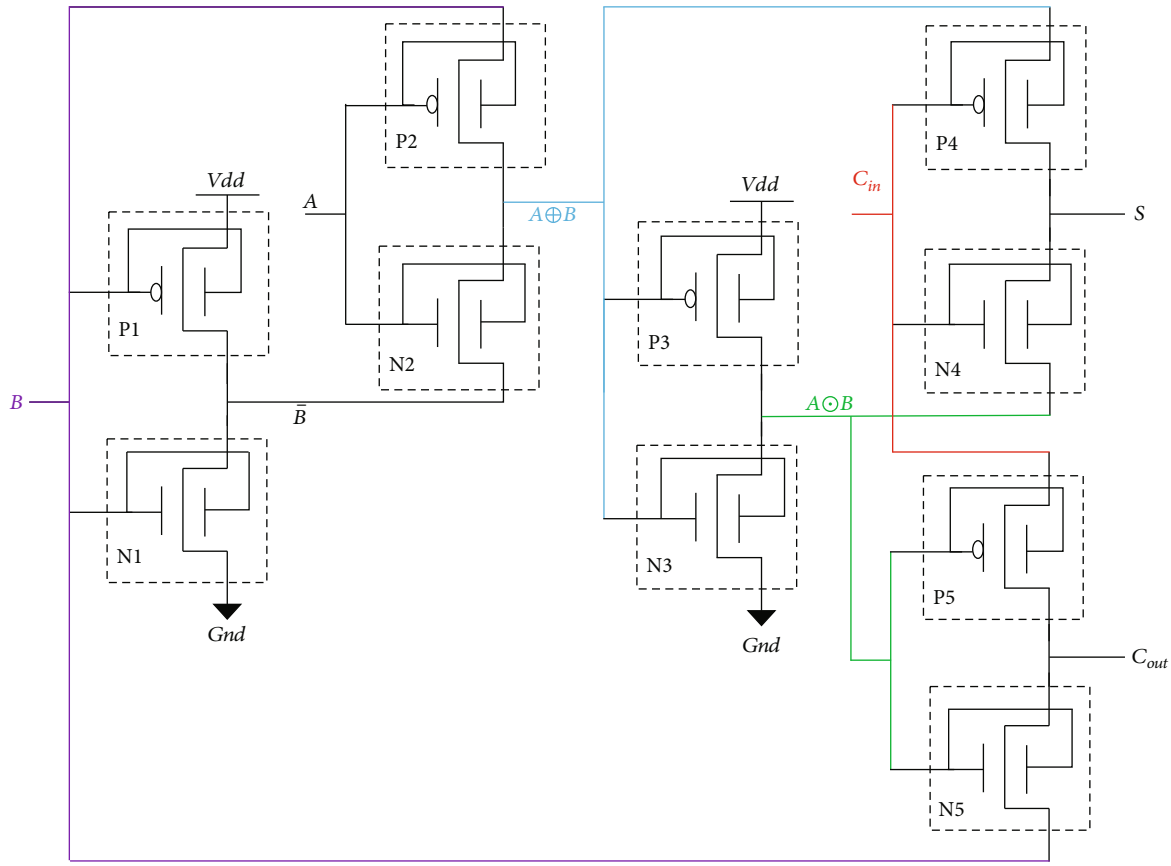

Figure 2: FinFET modelling of MFA.

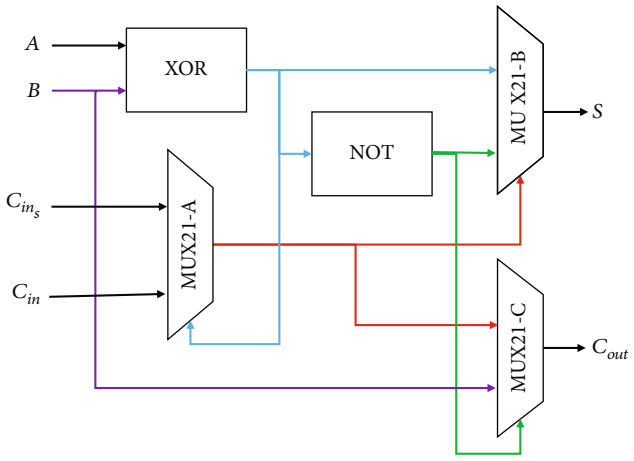

Figure 3: Block diagram of CISFA.

and N6 FinFETs, which acts as MUX21-C, so carry output is generated.

3.3. COPFA. The major advantage of COPFA is that it predicts two carry outputs such as carry output (C_{out}) and predictable carry output (C_{out_p}). These carry outputs are used as inputs for the CISFA module, which can be used to fast switching of data. Figure 5 shows the block diagram of COPFA, which is implemented by replacing XOR gate with XOR-AN module. This module generates both XOR and AND outputs, where AND output is the temporary carry output.

In most scenarios, C_{out} is generated when two or more numbers of ones available in input data, so this temporary carry out will check the availability of ones and generates the C_{out_p},respectively. Figure 6 shows the FinFET layout of COPFA, which contains 6 number of PMOS-FinFETs and 6 number of NMOS-FinFETs. It is also acts similar like MFA with extra AND operation.

4. Proposed DCR-ALU Method

ALUs are the basic building in every integrated circuit, and they decide the performance of various applications. Thus, the optimal design of adders and subtractors will improve the performance of microprocessors and microcontrollers. This section deals with the detailed analysis of implementation DCR-ALU, which is developed by using DCRHA, DCRHS, DCRHAS, and DCRAM, respectively. Further, these adders and subtractors are developed by using-by-using MFA, COPFA, and CISFA modules (mentioned in Section 3). Figure 7 shows the N-bit implementation of the proposed DCR-ALU, which consumes the low area, power, and delay properties as they are implemented by using advanced full adders. Here, A, B are N-bit inputs, S is the selection line with three inputs, and OUT is the final ALU

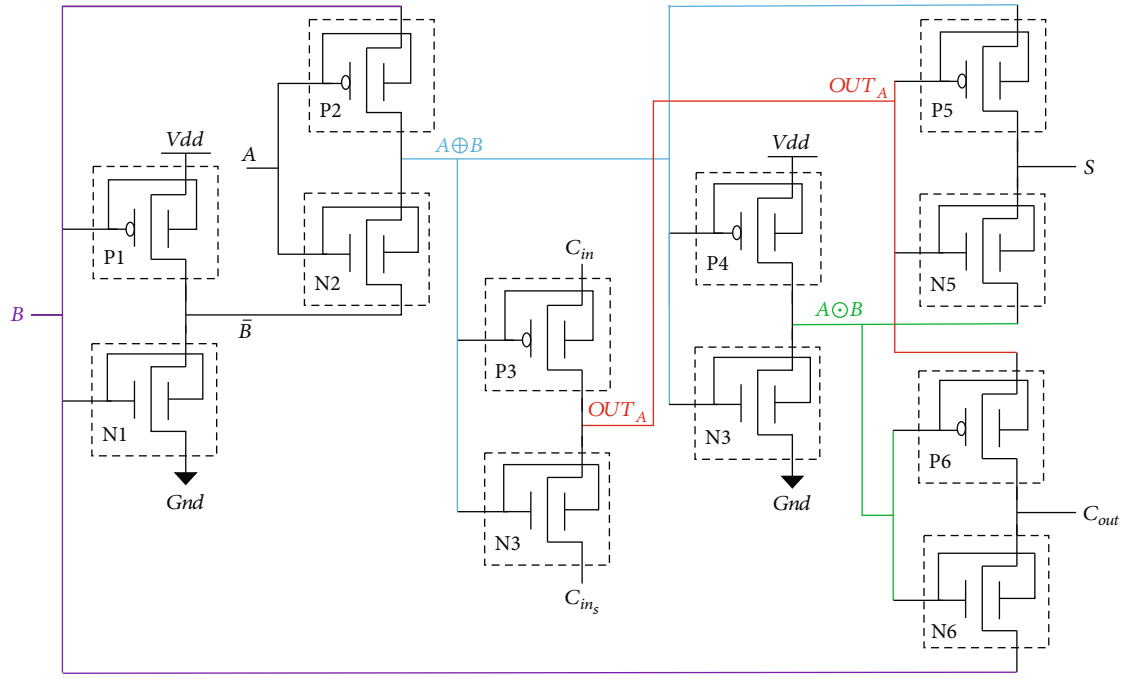

Figure 4: FinFET modelling of CISFA.

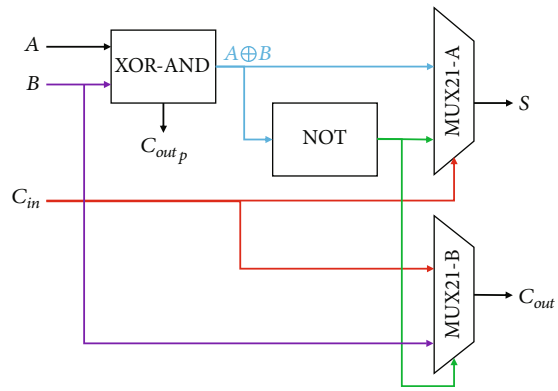

Figure 5: Block diagram of COPFA.

output. Further, based on selection combinations, ALU will perform the different operations. Table 1 presents the detailed operations for each selection combination of the proposed DCR-ALU.

4.1. DCRHA. The delay problems generated in the conventional adders are overcome by introducing CISFA-COPFA combination. The MFA and MUX21 of basic adders are replaced by CISFA-COPFA combination, which significantly reduces the area, power, and delay as compared to other approaches. Figure 8 shows the block diagram of DCHA. It reduces the problems generated in the SRHA by employing the high-speed carry propagation (HSCP) and regular-speed carry propagation (RSCP) paths. Here, HSCP is the low delay consumption path, RSCP is the high delay consumption path. Further, the DCHA is developed by replacing the intermediate MFA stages of SRHA with COPFA and CISFA modules.

The 4-bit DCHA totally contains 44 number of FinFETs. The detailed operation of 4-bit DCHA is illustrated as follows:

Step 1. Initially, A_0, B_0 and C_{in} are applied as input to the MFA-1 and it generates S_0, C_{out1} as output signals.

Step 2. Apply A_1, B_1 and C_{out1} variables as inputs to COPFA, which generates the C_{out2} as regular carry out and C_{out2p} as predicated carry out. Here, C_{out2p} is generated fast by analyzing the intermediate carries as compared to C_{out2}.

Step 3. Apply A_2, B_2, C_{out2}, and C_{out2p} variables as inputs to CISFA, and it generates S_2, C_{out3} as output signals. This module has the capacity to generate carry output by selecting high-speed input data levels. Based on the accurate sum generation situations, accurate path is selected. Usually, the RSCP path contains the generalized carry propagations, and HSCP path contains propagation carry from predicted outcome to selectable input.

Step 4. Finally, A_3, B_3 and C_{out3} are applied as input to the MFA, and it generates final outputs S_3, C_{out}.

The proposed 4-bit DCHA contains the "reconfiguration block," which is formed by the combination of CISFA and COPFA modules. This reconfigurable block is used to develop the N-bit DCRHA by repeating the reconfiguration blocks multiple times. Further, the number of reconfiguration blocks is decided by length of N-bit DCRHA. For N-bit length of DCRHA, $M(=(N-2)/2)$ numbers of reconfiguration blocks are used. Further, the number of transistors in the N-bit DCRHA is $20 + 24 * ((N-2)/2)$, respectively.

Figure 9 shows the block diagram of N-bit DCRHA, and its operation is illustrated as follows:

Step 1. Initially, A_0, B_0 and C_{in} are applied as input to the MFA-1, and it generates S_o, C_{out1}-based output signals.

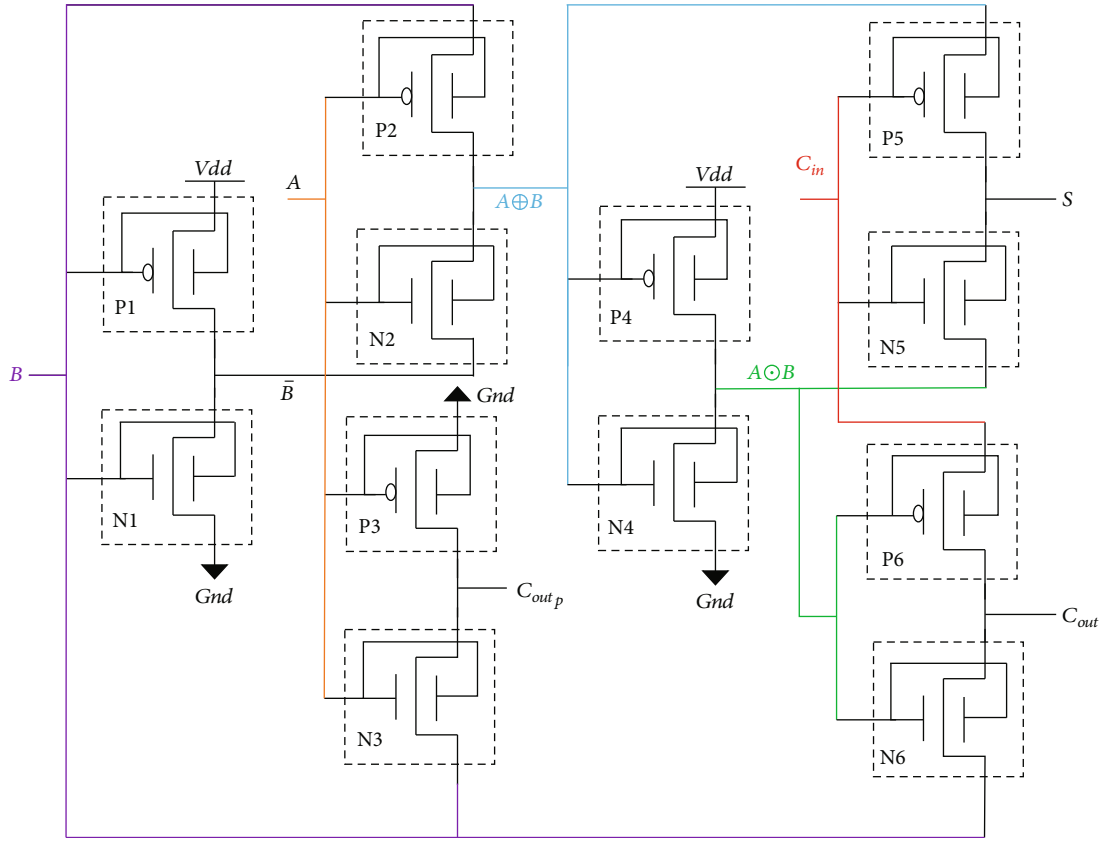

Figure 6: FinFET modelling of COPFA.

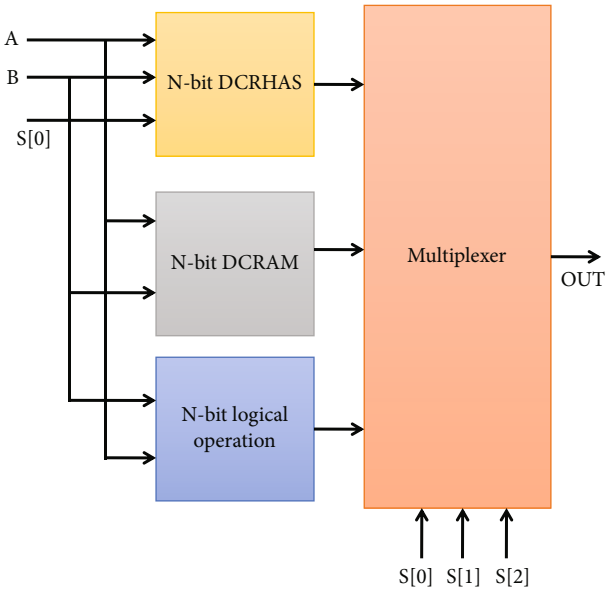

Figure 7: Architecture of N-bit DCR-ALU.

Step 2. Apply A_1, B_1 and C_{out1} variables as inputs to COPFA, which generates the $S_1, C_{out2} C_{out2p}$. Apply A_2, B_2, C_{out2}, and C_{out2p} variables as inputs to CISFA, and it generates S_2, C_{out3} as output signals.

Step 3. The process of reconfigurable blocks is repeated for other inputs, and the number of reconfigurable blocks is depending on factor M.

Step 4. Finally, A_{N-1}, B_{N-1}, and $C_{out_{N-1}}$ are applied as inputs to the MFA, and it generates final outputs S_{N-1}, C_{out}.

4.2. DCRHS. This section gives the detailed analysis of hybrid subtractors, which are developed by performing the twos complement addition process. Therefore, there is no special requirement of full subtractors for performing subtraction operation.

Figure 10 shows the block diagram of DCHS, and it is developed by performing the twos complement operation on DCHA. The 4-bit DCHS totally contains 52 (= 44 + 8) number of FinFETs; here 8 extra FinFETs are used for four inverters. The detailed operation of 4-bit DCHS is illustrated as follows. Initially, input B is applied to the inverter, which generates the $\bar{B}$ as output. Then, $A_0, \overline{B_0}$ and binary − 1 are applied as input to the MFA, but it acts as full subtractor and generates D_0 and Br_{out1} as output signals. Apply $A_1, \overline{B_1}$ and Br_{out1} variables as inputs to COPFA, which generates the Br_{out2} as regular borrow out and Br_{out2p} as predicated borrow out. Here, Br_{out2p} is generated fast by analyzing the intermediate borrows as compared to Br_{out2}. Apply $A_2, \overline{B_2}$, Br_{out2}, and Br_{out2p} variables as inputs to CISFA, and it generates D_2, Br_{out3} as output signals. Finally, $A_3, \overline{B_3}$ and Br_{out3} are

Table 1: N-bit DCT-ALU operations.

Selection line (S)	Output operation	Comments
000	OUT = DCRHAS $(A, B, S[0])$	As $S[0] = 0$, DCRHAS acts as DCRHA.
001	OUT = DCRHAS $(A, B, S[0])$	As $S[0] = 1$, DCRHAS acts as DCRHS.
010	OUT = DCRAM $(A, B, S[0])$	Multiplication
100	OUT = A	Buffer
101	OUT = XOR (A, B)	EX-OR logical operation
110	OUT = XNOR (A, B)	EX-NOR logical operation
110	OUT = AND (A, B)	AND logical operation
111	OUT = OR (A, B)	OR logical operation

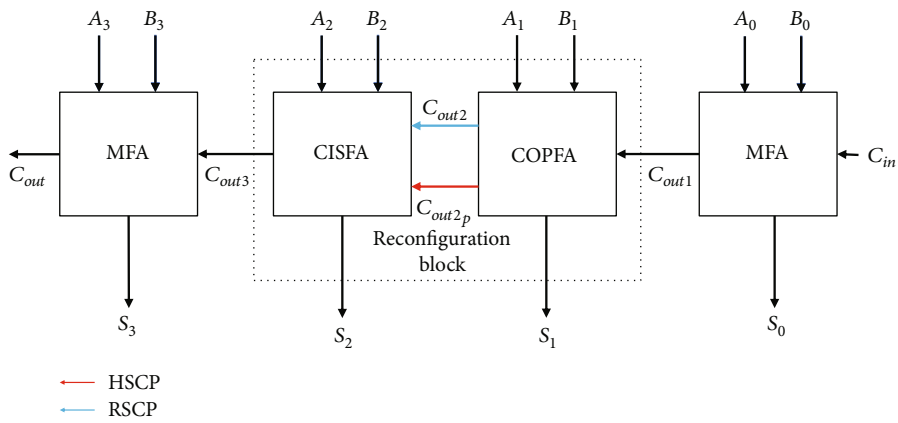

Figure 8: Proposed 4-bit DCHA.

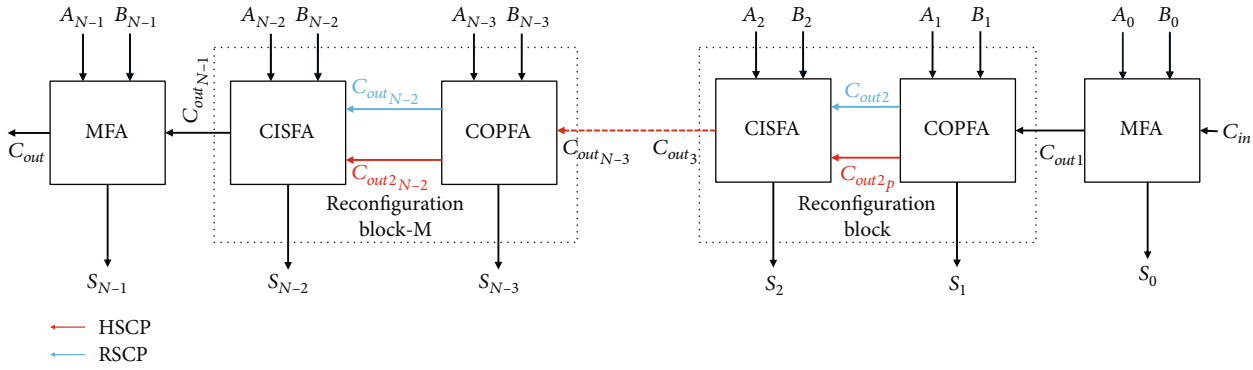

Figure 9: Proposed N-bit DCRHA.

applied as input to the MFA, and it generates final outputs S_3, $\mathrm{Br}_{\mathrm{out}}$.

Figure 11 shows the block diagram of N-bit DCRHS, and its operation is illustrated as follows, initially, $A_0, \overline{B_9}$ and 1 are applied as input to the MFA and it generates D_0, $\mathrm{Br}_{\mathrm{out1}}$ based output signals. Apply $A_1, \overline{B_1}$ and $\mathrm{Br}_{\mathrm{out1}}$ variables as inputs to COPFA, which generates the D_1, $\mathrm{Br}_{\mathrm{out2}}\mathrm{Br}_{\mathrm{out2}p}$. Apply $A_2, \overline{B_2}$, $\mathrm{Br}_{\mathrm{out2}}$, and $\mathrm{Br}_{\mathrm{out2}p}$ variables as inputs to CISFA, and it generates D_2, $\mathrm{Br}_{\mathrm{out3}}$ as output signals. The process of reconfigurable blocks is repeated for M number of times and generates multiple difference, borrow outputs. Finally, $A_{N-1}, \overline{B}_{N-1}$ and $C_{\mathrm{out}_{N-1}}$ are applied as input to the MFA and it generates final outputs D_{N-1}, $\mathrm{Br}_{\mathrm{out}}$.

4.3. DCRHAS. Figure 12 shows the block diagram of 4-bit DCHAS, and it is developed by performing the twos complement operation with mode selection properties. Figure 13 shows the block diagram of N-bit DCRHAS. The DCHAS

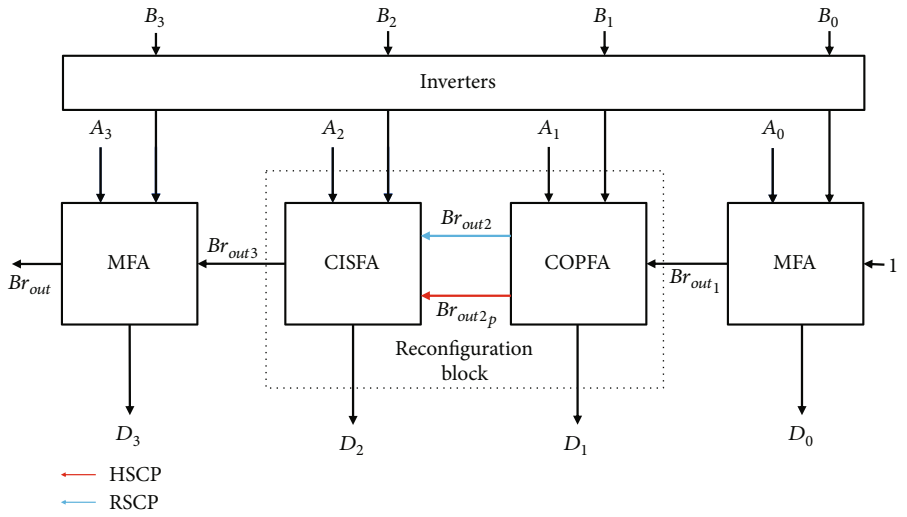

Figure 10: Proposed 4-bit DCHS

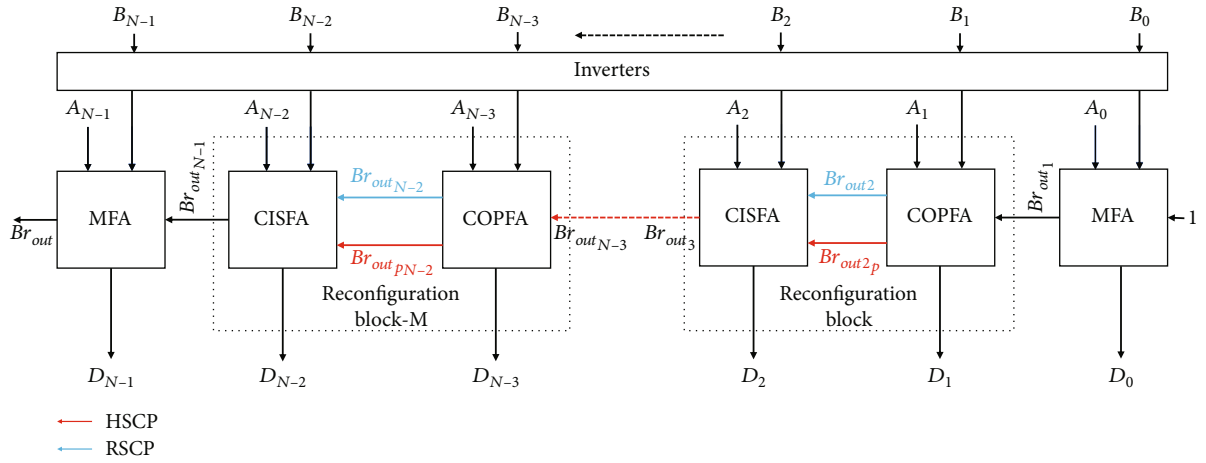

Figure 11: Proposed N-bit DCRHS

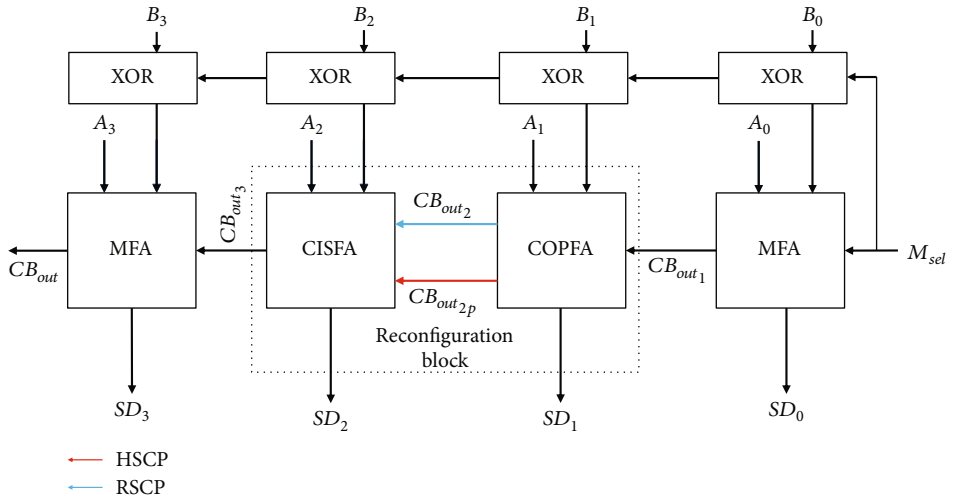

Figure 12: Proposed 4-bit DCHAS.

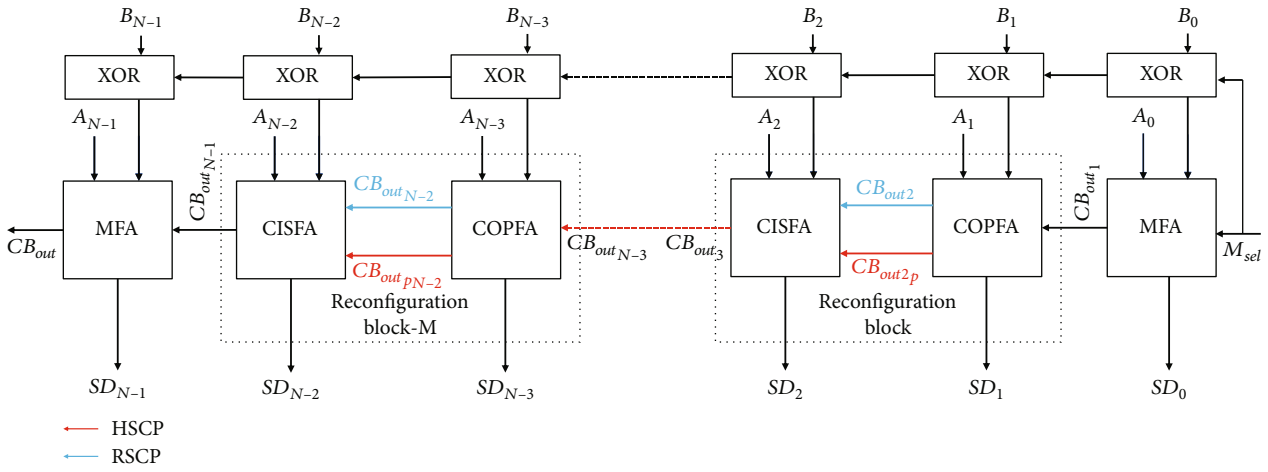

Figure 13: Proposed N-bit DCHAS.

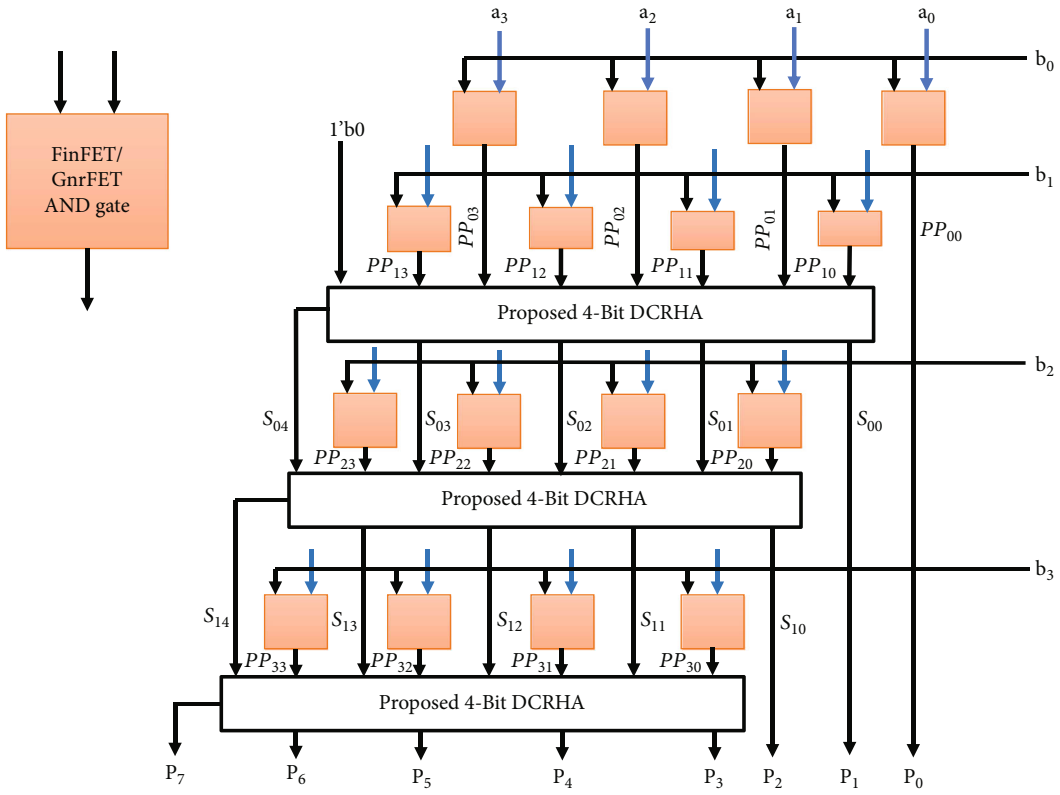

Figure 14: Architecture of 4-bit DCAM.

contains M_{sel} pin, which is used to select addition and subtraction operations. Further, if M_{sel} is zero, then DCHAS acts as DCHA, because XOR operation between M_{sel},B results output the same as B only. In this case, sum-difference output (SD) signal acts as S and carry-borrow (CB) output signal acts as C_{out}. In addition, if M_{sel} is one, then DCHAS act as DCHS, because XOR operation between M_{sel},B results output as $\bar{B}$, respectively. This complemented output is useful for twos complement-based addition process to generate the end difference outcome. In this case, sum-difference output (SD) signal acts as D and carry-borrow (CB) output signal acts as Br_{out}.

$$SD = \begin{cases} A + B, M_{sel} = 0, \\ A - B, M_{sel} = 1. \end{cases} \quad (6)$$

4.4. DCRAM. This section gives the detailed analysis of delay controlled reconfigurable array multiplier (DCRAM), which is developed by using DCRHA. Figure 14 shows the

Table 2: Operation of 4-bit DCAM.

Operation							Comments
			a_3	a_2	a_1	a_0	Input a
			b_3	b_2	b_1	b_0	Input b
		1'b0	PP_{03}	PP_{02}	PP_{01}	PP_{00}	Partial products generated and inputs to 1st stage 4-bit DCRHA
		PP_{13}	PP_{12}	PP_{11}	PP_{10}		
	S_{04}	S_{03}	S_{02}	S_{01}	S_{00}		Sum generated from first stage 4-bit DCRHA and applied as input to second stage 4-bit DCRHA
	PP_{23}	PP_{22}	PP_{21}	PP_{20}			Partial products
S_{14}	S_{13}	S_{12}	S_{11}	S_{10}			Sum generated from second stage 4-bit DCRHA and applied as input to third stage 4-bit DCRHA
PP_{33}	PP_{32}	PP_{31}	PP_{30}				Partial products
S_{24}	S_{23}	S_{22}	S_{21}	S_{20}			Sum generated from third stage 4-bit DCRHA
P_7	P_6	P_5	P_4	P_3	P_2	P_1	P_0	Final product bits

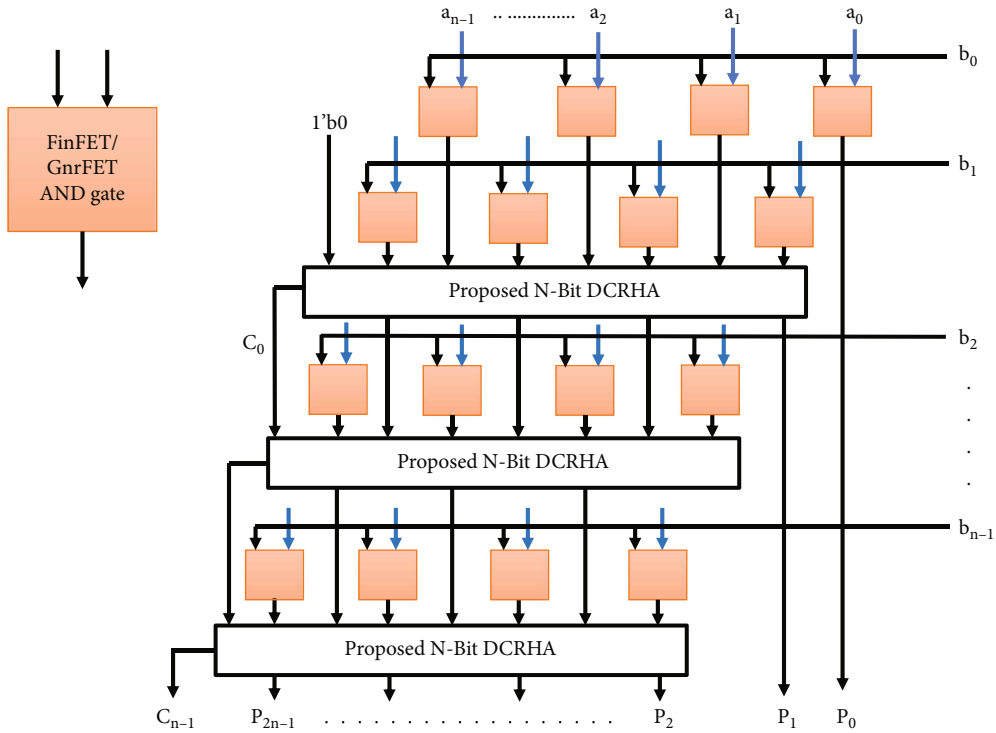

Figure 15: Architecture of N-bit DCRAM.

architecture of DCAM, which is developed by using FinFET/GnrFET-based AND gates and 4-bit DCRHAs. Initially, 4-bit inputs such as $a[3:0] = (a_3, a_2, a_1, a_0)$ and $b[3:0] = (b_3, b_2, b_1, b_0)$ are applied to AND gates, which generates the partial products.

$$PP_{mn} = \text{FinFET}_{\text{AND}(a_m, b_m)}. \tag{7}$$

Here, PP_{mn} represents the partial products, m represents the bit position of input a, and n represents the bit position of input b. In the scenario of 4-bit DCAM, values of m and n are ranged from 0 to 3, respectively. For an instance, PP_{00} represents the FinFET_AND operation between a_0 and b_0.Further, these partial products are applied to multistage 4-bit DCRHA, which generates the final multiplier output bits. Table 2 presents the detailed operation of 4-bit DCAM.

The operation of 4-bit DCAM is extended with N-bits with reconfigurable properties and generated as N-bit DCRAM. Figure 15 presents the architecture of N-bit DCRAM, which is developed by using FinFET/GnrFET-based AND gates and N-bit DCRHAs. Here, inputs a, b are ranged from 0 to $(n-1)$ such as $\{a_{n-1}, a_{n-2}, \cdots \cdots .a_2, a_1, a_0\}$

Table 3: Operation of 4-bit DCRAM.

Operation								Comments
		a_{n-1}	·····	a_3	a_2	a_1	a_0	Input a
		b_{n-1}	·····	b_3	b_2	b_1	b_0	Input b
	1'b0	$PP_{0,n-1}$	·····	PP_{03}	PP_{02}	PP_{01}	PP_{00}	Partial products generated and inputs to 1st stage 4-bit DCRHA
	$PP_{1,n-1}$	$PP_{1,n-1}$	·····	PP_{12}	PP_{11}	PP_{10}		
$S_{0,n}$	$S_{0,n-1}$	$S_{0,n-2}$	·····	S_{02}	S_{01}	S_{00}		Sum generated from first stage 4-bit DCRHA and applied as input to second stage 4-bit DCRHA
$PP_{2,n-1}$	$PP_{2,n-2}$	$PP_{2,n-3}$	·····	PP_{21}	PP_{20}			Partial products
								Repeat the process until (N-1) number of addition stages are completed.
P_{2n-1}	P_{2n-2}	P_{2n-3}	·····	P_3	P_2	P_1	P_0	Final product bits

Table 4: Device parameters for FinFET.

Parameter	Value
FinFET technologies	7 nm, 10 nm, 14 nm, 16 nm, and 20 nm
Length of the channel (L)	7 nm, 10 nm, 14 nm, 16 nm, and 20 nm
Thickness of front/back gate oxide	1.2 nm
Thickness of the fin (Si)	4 nm
Height of the fin (h_{fin})	7 nm, 10 nm, 14 nm, 16 nm, and 20 nm
Work function	4.5 eV/4.9 eV
Power supply (VDD)	1 V
Channel doping	2×10^{20} cm^{-3}
Source/drain doping	2×10^{20} cm^{-3}

and $\{b_{n-1}, b_{n-2}, \cdots \cdots ., b_2, b_1, b_0\}$, respectively. Further, develop the partial products using FinFET/GnrFET-based AND gates. Here, N-bit DCAM is capable of developing the N^2 number of partial products. Further, The N-bit DCAM contains (N-1) number of N-bit DCRHAs, where the output of one adder is applied is input to next stage. Then, apply these partial products to N-bit DCRHAs and reduce the partial products with stage-by-stage multiplication, which resulted in final product bits. Table 3 presents the detailed operation of N-bit DCRAM.

5. Results and Discussion

This section provides a full examination of simulation results obtained using a variety of parameters, which may be used to demonstrate the efficiency of the proposed research. On top of that, all of the planned tasks have been implemented and simulated, and their parameters have been determined using the HSpice tool on both the FinFET and GnrFET technology platforms. Also included is a full analysis of the many simulation parameters that were utilized in the design and development of the proposed work using the H-SPICE tool, which is shown in Table 4.

5.1. Performance Evaluation. This work considers the multiple metrics to evaluate the performance of the proposed models and comparison with state of art approaches. They are average power consumption (APC), sum rise delay (SRD), sum fall delay (SFD), carry output rise delay (CORD), carry output fall delay (COFD), static noise margin (SNM), total energy consumption (TEC), static power consumption (SPC), total current (TC), and propagation delay (PD). Table 5 compares the 7 nm based FinFET and 22 nm based GnrFET performances for various proposed modules such as DCHA, DCHS, DCHAS, DCAM, and DCR-ALU. Further, the FinFET technology resulted in superior performance as compared GnrFET technology for all the proposed models.

Table 6 compares the performance of the proposed CISFA with various state of art models like RHMFA [10], HFFA [11], IMC-FA [12], CTFA [13], and MFA [18]. From the simulations, it is observed that the proposed CISFA resulted in superior performance, because the CISFA utilizes the advanced path selection mechanisms. Further, the conventional methods suffering with the more power consumptions due to a greater number of transistors.

Table 7 compares the performance of the proposed DCHA with various state of art adders like HFA [11], IMCA [12], CTA [13], and RCA [18]. From the simulations, it is observed that the proposed DCHA resulted in superior performance, because the CPHA utilizes the COPFA-based path forwarding properties. Table 8 compares the performance of the proposed DCHS with various state of art subtractors like PTLS [15], MTCMOSS [16], and TUTS [17]. From the simulations, it is observed that the proposed DCHS resulted in superior performance, because it performs subtraction using twos complement addition process. Table 9 compares the performance of the proposed DCHAS with various state of art combined adders and subtractors like GAEAS [19], FHAS [20], CNTFET-AS [21], and MRAS [23]. From the simulations, it is observed that the proposed DCHAS resulted in superior performance, because it performs subtraction using twos complement addition process.

Table 10 compares the performance of the proposed DCAM with various state of art multipliers like FDM [26], MBIM [28], EEPAL [29], and CVMP [27]. From the

Table 5: FinFET and GnrFET comparisons of the proposed methods.

Method	DCHA		DCHS		DCHAS		DCAM		DCR-ALU	
Technology	FinFET-7 nm	GnrFET-22 nm	FinFET-7 nm	GnrFET-22 nm	FinFET-7 nm	GnrFET-22 nm	FinFET-7 nm	GnrFET-22 nm	FinFET-7 nm	GnrFET-22 nm
APC (nw)	2.192	4.14328	2.24344	4.25016	2.3512	4.4568	4.93752	9.35928	7.0771	13.41497
SRD (ns)	15.848	29.9432	16.5576	31.7864	13.3424	25.2336	28.019	52.99056	40.160	75.95314
SFD (ns)	20.024	37.8824	20.828	39.692	27.0656	51.15424	56.837	107.4239	81.467	153.9743
CORD (ps)	1.9776	3.75448	2.0572	3.8508	3.46304	6.545056	7.2723	13.74462	10.423	19.70062
COFD (ps)	15.824	29.8464	16.4584	30.6376	3.3084	6.25356	6.94764	13.13248	9.9582	18.82322
PD (ns)	1.976	3.71928	2.02376	3.77064	13.7424	25.97136	28.859	54.53986	41.364	78.17379
TC (nA)	2.536	4.81552	2.64784	5.00176	9.3608	17.69512	19.657	37.15975	28.176	53.26231
SPC (nw)	2.008	3.38968	2.27424	4.31136	4.32288	8.1824	9.0780	17.18304	13.011	24.62902
TEC (nJ)	0.072	0.11096	0.05296	0.10064	1.64344	3.0616	3.4512	6.42936	4.9467	9.215416
SNM	5.8722	10.6472	5.92344	11.1944	6.24776	11.7864	13.1203	24.75144	18.8057	35.47706

Table 6: Performance comparison of the proposed CISFA with existing full adders.

Method	RHMFA [10]	HFFA [11]	IMC-FA [12]	CTFA [13]	MFA [18]	CISFA
APC (nw)	12.00003	10.3798	8.384	2.262218	1.714	1.05472
SRD (ns)	45.57444	42.19374	42.02087	41.32643	9.7537	6.00208
SFD (ns)	92.32847	84.65102	84.50273	83.82905	19.785	12.1784
CORD (ps)	71.17313	53.90735	53.51331	10.72597	2.5315	1.54464
COFD (ps)	52.85658	50.23387	46.05195	10.24718	2.4185	0.8308
PD (ns)	57.09781	50.79739	43.00979	42.5649	10.046	5.82336
TC (nA)	41.72174	34.15361	30.00008	29.00142	6.8448	4.17848
SPC (nw)	31.52455	24.90975	21.06001	13.38892	3.16	1.94952
TEC (nJ)	38.33638	31.93173	25.45166	5.089908	1.2013	0.72616
SNM	35.93823	29.27089	29.06582	19.35038	4.567	2.64616

Table 7: Performance comparison of DCHA with existing 4-bit adders.

Method	HFA [11]	IMCA [12]	CTA [13]	RCA [18]	DCHA
APC (nw)	18.47604	14.92352	4.026748	3.864	2.192
SRD (ns)	75.10486	74.79715	73.56105	27.907	15.848
SFD (ns)	150.6788	150.4149	149.2157	35.268	20.024
CORD (ps)	95.95508	95.25369	19.09223	3.486	1.9776
COFD (ps)	89.41629	81.97247	18.23998	27.897	15.824
PD (ns)	90.41935	76.55743	75.76552	3.457	1.976
TC (nA)	60.79343	53.40014	51.62253	4.555	2.536
SPC (nw)	44.33936	37.48682	23.83228	3.6015	2.008
TEC (nJ)	56.83848	45.30395	9.060036	0.134	0.072
SNM	52.10218	51.73716	34.44368	10.35	5.8722

Table 8: Performance comparison of DCHS with existing 4-bit subtractors.

Method	PTLS [15]	MTCMOSS [16]	TUTS [17]	DCHS
APC (nw)	19.21508	15.52046	4.1818	2.24344
SRD (ns)	78.10905	77.78904	76.349	16.5576
SFD (ns)	156.706	156.4315	155.143	20.828
CORD (ps)	99.79328	99.06384	19.8592	2.0572
COFD (ps)	92.99294	85.25137	18.9658	16.4584
PD (ns)	94.03612	79.61973	78.7914	2.02376
TC (nA)	63.22517	55.53615	53.6843	2.64784
SPC (nw)	46.11293	38.98629	24.7857	2.27424
TEC (nJ)	59.11202	47.11611	9.42237	0.05296
SNM	54.18627	53.80665	35.8243	5.92344

simulations, it is observed that the proposed DCAM resulted in superior performance, because it utilizes the delay controller environment for data transfer from one module to another module.

Table 11 compares the performance of the proposed DCR-ALU with various existing ALUs such as HS-ALU [32], FALU-32 [34], SB-FinFET [35], and STT-MTJ [36]. The conventional ALUs are developed by using basic adders, multiplier prototypes, which caused to increase area, delay, and power consumptions. Further, the proposed DCR-ALU utilizes delay intensive paths, which caused to improve the performance.

Table 9: Performance comparison of DCHAS with existing adders and subtractors.

Method	GAEAS [19]	FHAS [20]	CNTFET-AS [21]	MRAS [23]	DCHAS
APC (nw)	20.52005	17.74946	14.33664	3.868393	2.3512
SRD (ns)	77.93229	72.1513	71.85569	70.6682	13.3424
SFD (ns)	157.8817	144.7532	144.4997	143.3477	27.0656
CORD (ps)	121.7061	92.18157	91.50776	18.34141	3.46304
COFD (ps)	90.38475	85.89992	78.74883	17.52268	3.3084
PD (ns)	97.63726	86.86354	73.54674	72.78598	13.7424
TC (nA)	71.34418	58.40267	51.30014	49.59243	9.3608
SPC (nw)	53.90698	42.59567	36.01262	22.89505	4.32288
TEC (nJ)	65.55521	54.60326	43.52234	8.703743	1.64344
SNM	61.45437	50.05322	49.70255	33.08915	6.24776

Table 10: Performance comparison of DCAM with existing multipliers.

Method	FDM [26]	MBIM [28]	EEPAL [29]	CVMP [27]	DCAM
APC (nw)	22.57206	19.52441	15.7703	4.255232	4.93752
SRD (ns)	85.72552	79.36643	79.04126	77.73502	28.019
SFD (ns)	173.6699	159.2285	158.9497	157.6825	56.837
CORD (ps)	133.8767	101.3997	100.6585	20.17555	7.2723
COFD (ps)	99.42323	94.48991	86.62371	19.27495	6.94764
PD (ns)	107.401	95.54989	80.90141	80.06458	28.859
TC (nA)	78.4786	64.24294	56.43015	54.55167	19.657
SPC (nw)	59.29768	46.85524	39.61388	25.18456	9.0780
TEC (nJ)	72.11073	60.06359	47.87457	9.574117	3.4512
SNM	67.59981	55.05854	54.67281	36.39807	13.1203

Table 11: Performance comparison of DCR-ALU with existing ALUs.

Method	HS-ALU [32]	FALU-32 [34]	SB-FinFET [35]	STT-MTJ [36]	DCR-ALU
APC (nw)	24.62406	21.29935	17.20397	4.642072	7.0771
SRD (ns)	93.51875	86.58156	86.22683	84.80184	40.160
SFD (ns)	189.458	173.7038	173.3996	172.0172	81.467
CORD (ps)	146.0473	110.6179	109.8093	22.00969	10.423
COFD (ps)	108.4617	103.0799	94.4986	21.02722	9.9582
PD (ns)	117.1647	104.2362	88.25609	87.34318	41.364
TC (nA)	85.61302	70.0832	61.56017	59.51092	28.176
SPC (nw)	64.68838	51.1148	43.21514	27.47406	13.011
TEC (nJ)	78.66625	65.52391	52.22681	10.44449	4.9467
SNM	73.74524	60.06386	59.64306	39.70698	18.8057

6. Conclusion

This article implemented the FinFET and GnrFET technology-based DCR-ALU by adopting the DCHA, DCHS, DCHAS, and DCAM modules. Initially, a unified 12 transistor COPFA and CISFA are developed by using multiplexer selection logic with low delay paths. Then, DCHA, DCHS, and DCHAS modules are developed by using COPFA and CISFA, which has the potential capacity to select the low-speed and high-speed carry propagation paths. In addition, DCAM developed with low area properties by using DCHA and FinFET/GnrFET technology-based AND gates. Moreover, all these DCHA, DCHS, DCHAS, and DCAM modules are extended with reconfigurable properties, which are formed as N-bit DCRHA, DCRHS, DCRHAS, and DCFRAM modules. The simulation results demonstrate that the proposed approach outperforms conventional adders and subtractors and consumed a smaller number of transistors, reduced power consumption with path delays. In future, the present work can be extended to

compute multiplication and accumulator operations with the proposed hybrid architectures.

Data Availability

The data used to support the findings of this study are included within the article.

Disclosure

The study was performed as a part of the Employment of Salale University, Ethiopia.

Conflicts of Interest

The authors declared that there is no conflict of interest in publication.

References

[1] T. Watanabe, "All-digital VCO-ADC TAD using 4CKES-type in 16-nm FinFET CMOS for technology scaling with stochastic-ADC method," in *2021 28th IEEE International Conference on Electronics, Circuits, and Systems (ICECS)*IEEE.

[2] S. Seyedi and N. J. Navimipour, "Designing a new 4: 2 compressor using an efficient multi-layer full-adder based on nanoscale quantum-dot cellular automata," *International Journal of Theoretical Physics*, vol. 60, no. 7, pp. 2613–2626, 2021.

[3] A. A. Sujata and D. Lalitha, "Survey on latest FinFET technology and its challenges in digital circuits and IC's," *International Journal of Electronics Engineering*, vol. 10, no. 2, pp. 577–585, 2018.

[4] P. Saritha, J. Vinitha, S. Sravya, V. Vijay, and E.. Mahesh, "4-Bit Vedic Multiplier with 18nm FinFET Technology," in *2020 International Conference on Electronics and Sustainable Communication Systems (ICESC)*, pp. 1079–1084, Coimbatore, India, 2020.

[5] M. T. Fakheri, K. Navi, and M. Tehrani, "A novel device to implement full set of three-input logic gates using a naphthalene-based single-molecule field-effect transistor," *IEEE Transactions on Electron Devices*, vol. 68, no. 2, pp. 733–738, 2021.

[6] Ayushi and Er. K. Chandra, "Low power and high speed mutliplier deisgns using FinFET technology," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 7, no. 2, pp. 143–146, 2018.

[7] W.-X. You, S. Pin, and H. Chenming, "Evaluation of NC-FinFET based subsystem-level logic circuits," *IEEE Transactions on Electron Devices*, vol. 66, no. 4, pp. 2004–2009, 2019.

[8] D. K. Nandhaiahgari, R. P. Somineni, and C. R. Kumari, "Design and analysis of different full adder cells using new technologies," *International Journal of Reconfigurable and Embedded Systems*, vol. 9, no. 2, p. 116, 2020.

[9] M. Nayeri, P. Keshavarzian, and M. Nayeri, "High-speed pen-ternary inverter gate using GNRFET," *Journal of Advances in Computer Research*, vol. 10, no. 2, pp. 53–59, 2019.

[10] S. B. Mamaghani, M. H. Moaiyeri, and G. Jaberipur, "Design of an efficient fully nonvolatile and radiation-hardened majority-based magnetic full adder using FinFET/MTJ," *Microelectronics Journal*, vol. 103, p. 104864, 2020.

[11] M. A. Vendhan and S. Deepa, "Investigations on performance metrics of FINFET based 8-bit low power adder architectures implemented using various logic styles," *Indian Journal of Science and Technology*, vol. 11, no. 24, pp. 1–22, 2018.

[12] N. Hajizadeh Bastani and K. Navi, "Imprecise minority-based full adder for approximate computing using CNFETs," *International Journal of Nanoscience and Nanotechnology*, vol. 15, no. 4, pp. 239–248, 2019.

[13] M. Mahjoubi, M. Dadashi, K. Manochehri, and S. Pourmozafari, "Two efficient ternary adder designs based on CNFET technology," *computer and knowledge Engineering*, vol. 3, no. 1, pp. 97–106, 2020.

[14] F. M. Sardroudi, M. Habibi, and M. H. Moaiyeri, "CNFET-based design of efficient ternary half adder and 1-trit multiplier circuits using dynamic logic," *Microelectronics Journal*, vol. 113, p. 105105, 2021.

[15] A. Rajput, T. Dua, R. Kumawat, and A. Srinivasulu, "Novel CMOS and PTL based half subtractor designs," in *2021 IEEE International Symposium on Smart Electronic Systems (iSES)(-Formerly iNiS)*IEEE.

[16] A. Sharma, N. Saxena, and A. S. Rajput, "Efficient and high-performance MTCMOS employed 34T-full subtractor," *Journal of Active & Passive Electronic Devices*, vol. 13, p. 1, 2018.

[17] Y. Shrivastava and T. K. Gupta, "Designing of low-power high-speed noise immune CNTFET 1-Trit unbalanced ternary subtractor," *Journal of Circuits, Systems and Computers*, p. 2250082, 2022.

[18] D. R. Florance and B. Prabhakar, "Design of FinFET and FET based full adder cell using multiplexer selection logic," in *2022 4th International Conference on Smart Systems and Inventive Technology (ICSSIT)*IEEE.

[19] D. Ajitha and M. C. S. Reddy, "Intelligent signal gating-aware energy-efficient 8-bit FinFET arithmetic and logic unit," *Circuits, Systems, and Signal Processing*, vol. 41, no. 1, pp. 299–320, 2022.

[20] E. Ramkumar, D. Gracin, P. Rajkamal, B. P. Bhuvana, and V. K. Bhaaskaran, "Design and analysis of low power and high speed FinFET based hybrid full adder/subtractor circuit (FHAS)," in *2020 IEEE International Symposium on Smart Electronic Systems (iSES)(Formerly iNiS)*IEEE.

[21] T. Sharma and L. Kumre, "CNTFET-based design of ternary arithmetic modules," *Circuits, Systems, and Signal Processing*, vol. 38, no. 10, pp. 4640–4666, 2019.

[22] E. Taka, K. Maragos, G. Lentaris, and D. Soudris, "Process variability analysis in interconnect, logic, and arithmetic blocks of 16-nm FinFET FPGAs," *ACM Transactions on Reconfigurable Technology and Systems (TRETS)*, vol. 14, no. 3, pp. 1–30, 2021.

[23] S. Riyaz, S. F. Naz, and V. K. Sharma, "Multioperative reversible gate design with implementation of 1-bit full adder and subtractor along with energy dissipation analysis," *International Journal of Circuit Theory and Applications*, vol. 49, no. 4, pp. 990–1012, 2021.

[24] Z. Zhang, Y. Luo, Y. Cui et al., "A Polarization-Switching, Charge-Trapping, Modulated arithmetic logic unit for in-memory computing based on ferroelectric fin field-effect transistors," *ACS Applied Materials & Interfaces*, vol. 14, no. 5, pp. 6967–6976, 2022.

[25] Z. T. Sandhie, F. U. Ahmed, and M. H. Chowdhury, "Design of ternary logic and arithmetic circuits using GNRFET," *IEEE Open Journal of Nanotechnology*, vol. 1, pp. 77–87, 2020.

[26] P. Zakian and R. N. Asli, "An efficient design of low-power and high-speed approximate compressor in FinFET technology," *Computers & Electrical Engineering*, vol. 86, p. 106651, 2020.

[27] V. M. Senthilkumar, S. Ravindrakumar, D. Nithya, and N. V. Kousik, "A vedic mathematics based processor core for discrete wavelet transform using FinFET and CNTFET technology for biomedical signal processing," *Microprocessors and Microsystems*, vol. 71, p. 102875, 2019.

[28] F. Sabetzadeh, M. H. Moaiyeri, and M. Ahmadinejad, "A majority-based imprecise multiplier for ultra-efficient approximate image multiplication," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 66, no. 11, pp. 4200–4208, 2019.

[29] B. P. Bhuvana and V. K. Bhaaskaran, "Design of FinFET-based energy efficient pass-transistor adiabatic logic for ultra-low power applications," *Microelectronics Journal*, vol. 92, p. 104601, 2019.

[30] V. M. Senthil Kumar and S. Ravindrakumar, "Design of an Area-Efficient Finfet-Based Approximate Multiplier in 32-Nm Technology for Low-Power Application," in *Soft Computing and Signal Processing*, pp. 505–513, Springer, Singapore, 2019.

[31] S. Palanisamy, Kesavan, and R. Ramachandran, "FinFET-based power-efficient, low leakage, and area-efficient DWT lifting architecture using power gating and reversible logic," *International Journal of Circuit Theory and Applications*, vol. 48, no. 8, pp. 1304–1318, 2020.

[32] N. Shylashree, B. Venkatesh, T. M. Saurab, T. Srinivasan, and V. Nath, "Design and analysis of high-speed 8-bit ALU using 18 nm FinFET technology," *Microsystem Technologies*, vol. 25, no. 6, pp. 2349–2359, 2019.

[33] R. Patel, Y. Agrawal, and R. Parekh, "A novel slice-based high-performance ALU design using prospective single electron transistor," *IETE Journal of Research*, vol. 68, no. 2, pp. 1–10, 2022.

[34] J. Rubia and R. B. Lincy, "Design of low power 4 bit ALU using 32nm FinFET technology," *International Journal of Pure and Applied Mathematics*, vol. 120, no. 6, pp. 8089–8100, 2018.

[35] T. F. Canan, S. Kaya, H. Chenji, and A. Karanth, "Fine-grain reconfigurable logic circuits for adaptive and secure computing via work-function engineered Schottky barrier FinFETs," *IEEE Journal on Exploratory Solid-State Computational Devices and Circuits*, vol. 7, no. 2, pp. 150–158, 2021.

[36] S. V. Yamani, N. Usha Rani, and R. Vaddi, "Design and performance benchmarking of hybrid tunnel FET/STT-MTJ-based logic in-memory designs for energy efficiency," *IEEE Transactions on Magnetics*, vol. 58, no. 4, pp. 1–11, 2022.