# Effective Scheme of Parity-Preserving Reversible Floating-Point Divider

Mohammad Talebi[1], Mohammad Mosleh[1,*], Majid Haghparast[1,2], and Mohsen Chekin[1]

[1] Department of Computer Engineering, Dezful Branch, Islamic Azad University, Dezful, Iran

[2] Faculty of Information Technology, University of Jyväskylä, P.O. Box 35, FI-40014 University of Jyväskylä, Jyväskylä, Finland

* Corresponding Author: Mosleh@iaud.ac.ir

**Abstract.** Most recently, there has been a growing need for developing very large-scale integration (VLSI) circuits with low energy consumption and high speed for use in fast transmission systems. In addition, the main challenge in designing irreversible integrated circuits is heat generation due to data loss. Thus, in recent years, reversible design has been preferred for low-power VLSI circuits because the data is not lost. In this article, a new design of parity preserving reversible (PPR) floating-point divider is suggested. A floating-point divider structure includes parallel adder, multiplexer, register, and left-shift register. To optimize these circuits, first, we propose a 5×5 PPR block and a PPR D-latch. Second, using the proposed circuits, a ripple carry adder (RCA), a register, and an efficient parallel input parallel output (PIPO) left-shift register, rounding register, and normalization register circuits are introduced in PPR logic. The comparisons illustrate that the suggested circuits are preferable to the circuits presented in previous works in terms of various criteria such as quantum cost, constant inputs, and garbage outputs.

**Keywords:** Quantum-computing; reversible logic; parity-preserving (PP); floating-point divider; quantum-cost.

## 1. Introduction

Currently, the construction of Nano-scaled logic circuits and transistors is particularly important. Moreover, most designs are playing a vital role in the current society. One of the significant advantages of such designs is their low power consumption. Thus, if these circuits can be designed using reversible logic, the power consumption, heat generation in the transistor, and the complexity and density of the circuits will be significantly reduced. Besides, in future designs, logic circuits will not be feasible without the use of nanotechnology, quantum computing, and reversible logic, because the current designs, according to Moore's Law, cause high power consumption and heat production, thus leading to power dissipation.

Furthermore, the reliable scheme of Parity-Preserving Reversible (PPR) circuits reduces the possibility of computational errors due to the compactness of the circuits. Landauer (1961) showed that in irreversible logic computation, *KTLn2* Joules of thermal energy is released for all bits of data lost, where *K* is the Boltzmann constant and *T* is the absolute temperature at which the computations are performed [1]. Given that the losing heat of a data bit would be ($2.9 \times 10^{-21}$) Joules, which is a minimal amount. Moor (1965) has indicated that the number of elements will be approximately twice every eighteen months in digital circuits. So, power dissipation will become a significant challenge in the structures of VLSI circuits in the future[2]. Bannet (1973) demonstrated that reversible gates in circuits would prevent energy dissipation and reduce power consumption in circuits[3]. Moreover, the application of reversible logic in diverse fields like low-power CMOS circuit structure [4], quantum processing[5], and nanotechnology[6] has led to the design of complex circuits considered by researchers. One solution to optimizing power consumption in low-power CMOS is to apply quantum computing and reversible logic [7]. Computations in current computer systems are all irreversible. Thus, deleted data is lost and cannot be restored. A circuit becomes reversible when a one-to-one mapping among the outputs and inputs of each of its logic gates[8]. The main challenges in designing irreversible circuits are heat generation and high energy consumption due to the loss of data. Thus, reversible design is used for low-power VLSI circuits because the data is not lost. Therefore, it is expected that due to the unique characteristics of reversible circuits, in the near future we will see their use in the design and implementation of special purpose hardware in the field of computational intelligence, neural networks and

robotics[9-13]. Reversible circuits can be designed with parity-preserving capacity. PPR gates are important elements in the design of these models of circuits. Designing a reversible circuit using Parity-Preserving (PP) gates can detect computational errors in the circuit. Thus, PP capacity can be incorporated into the designs of reversible gates to reduce the possibility of computational errors. The division operation has been counted as one of the basic operations in the arithmetic logic unit(ALU)[14]. The divider is one of the most complex computational circuits implemented in digital systems, and the floating-point divider is of particularly importance for the design of microprocessors. A floating-point divider circuit consists of multiplexer, RCA adder, register, and PIPO left-shift register. The first n-bit reversible divider scheme for positive integers using the non-restoring division algorithm was introduced in 2009 and did not have the PP ability [15]. Two different approaches to reversible dividers were proposed in 2011 with the structure of a PPR-PIPO left-shift register. The first and second approaches have slight differences in architecture[16]. In 2016, a fault-tolerant reversible floating-point divider was designed. This scheme was presented using PPR divider components, containing: RCA adder, normalization register, PIPO left-shift register, register, rounding register, and a reversible divider[17].

In this article, we have proposed an efficient PPR floating-point divider, which can be used in digital systems to improve efficiency significantly. The scientific of this article will be as follows:

- Introducing a novel PPR block
- Proposing a RCA based on the suggested block
- Introducing a PPR D-latch
- Designing register, PIPO left-shift register, rounding register, and normalization register circuits using the proposed D-latch
- Designing an efficient PPR floating-point divider using the proposed circuits

This manuscript is organized as follows: Sec.2 presents an introduction to reversibility logic and division techniques. Sec.3 discusses the procedure used to design the proposed divider. Sec.4 presents the results of evaluations and comparisons. Finally, the manuscript is finished with a conclusion in Sec.5

## 2. Preliminaries

In this section, at first, the fundamental concepts of reversible logic are reviewed, and then division algorithms are discussed.

### 2.1 Reversible logic

A circuit will be reversible when there is a single-to-single mapping among the inputs and outputs of each of its logic gates, and each output can be retrieved from its unique input, and data will be lost if it is not possible to retrieve inputs from outputs. As a result, due to the single-to-single mapping among inputs and outputs in reversible circuits, data is not lost. Thus, using reversible computations, circuits with zero power loss can be developed. If the input vector in the reversible gate is $I_v = \{I_1 . I_2 \dots I_n\}$ and if the output vector is $O_v = \{O_1 . O_2 \dots O_n\}$, Eq. (1) is permanently established [8, 18]:

$$\mathrm{I_v} = \mathrm{O_v} \tag{1}$$

Besides, a reversible circuit will be named PP when the Exclusive-OR (XOR) of the input vector must be the same with the XOR of outputs (Eq.2). Thus, any error occurring in the output can be found out if the reversible circuit is made only of PPR blocks [19].

$$\mathrm{I_1} \oplus \mathrm{I_2} \oplus \dots \oplus \mathrm{I_{n-1}} \oplus \mathrm{I_n} = \mathrm{O_1} \oplus \mathrm{O_2} \oplus \dots \oplus \mathrm{O_{n-1}} \oplus \mathrm{O_n} \tag{2}$$

The main challenges to the design of reversible circuits are the optimization of criteria such as Constant Inputs (CI), Garbage Outputs (GO), and Quantum Cost (QC) [20, 21].

- **GO:** The number of additional outputs that have not been used as input to other gates or blocks, making a gate function reversible.
- **CI:** The number of inputs that have been kept constant with values of '0' or '1'.
- **QC:** The total quantum cost of reversible gates or blocks is considered [22].

### 2.2 Reversible gates

Reversible gates are classified as follows:

**NOT gate:** A NOT gate is a 1×1 quantum gate with QC equal to one, as illustrated in Fig. (1) [23]:

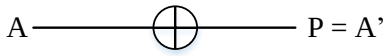



**Fig. (1):** Quantum circuit of a NOT gate[23]

**CNOT gate:** This gate, also called the Feynman Gate (FG), is a 2×2 reversible gate with control input A and target input B. The output generated by P = A and Q = A⊕B. As illustrated in Fig. (2), when the control input is A = '1', the inverse of the target input ($\bar{B}$) will be at the output. In other respects, the target B will be moved to Q unchanged [24].

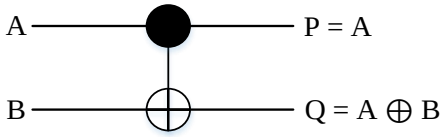



**Fig. (2):** Quantum circuit of a CNOT gate[24]

**Control gates V and V⁺:** Control gates V and $V^+$ are 2×2 quantum gates[23, 25-27]. When control line is A='0', the input is sent to the output unchanged. Besides, if control input A is '1', the controlled V and controlled -$V^+$ gates are sent to the output as V(B) and $V^+$(B) outputs, in order. Their QC is equal to one. The quantum representations of these gates are illustrated in Fig. (3):

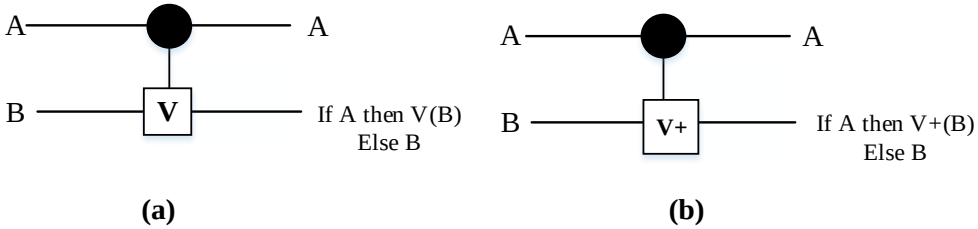



**Fig. (3):** Quantum circuits of control gates (a) V, (b)$V^+$ [23, 25-27]

The matrices V and $V^+$ are derived from Eqs. (3) and (4), as displayed below:

$$V = \frac{1+i}{2}\begin{pmatrix} 1 & -i \\ -i & 1 \end{pmatrix} \tag{3}$$

$$V^{+} = \frac{1}{i+1}\begin{pmatrix} 1 & -1/i \\ i & 1 \end{pmatrix} \quad (4)$$

The properties of the matrices are displayed using Eqs. (5), (6), and (7), as follows[28, 29]:

$$V \times V = NOT \quad (5)$$
$$V^{+} \times V = V \times V^{+} = I \quad (6)$$
$$V^{+} \times V^{+} = NOT \quad (7)$$

In order to detect bit errors of reversible circuits, we can design them with PP ability [30-34]. In the following section, we introduce three PPR logic blocks, including Double Feynman Gate (DFG), FRedkin Gate (FRG), and Bolhassani Haghparast Parity-Preserving Full-adder (BHPF) gates.

**DFG:** It is a well-known reversible gate, which is a 3×3 PPR gate. Its QC is equal to two [19]. Also, quantum circuit of the DFG gate is illustrated in Fig. (4).

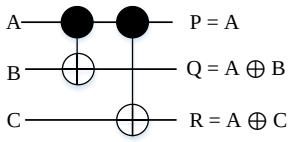



**Fig. (4):** Quantum circuit of the DFG gate [19]

**FRG:** This is a famous gate, which is a 3× 3 PPR gate illustrated in Fig. (5)[35]. Its outputs are P = A, Q = A′B⊕AC, and R = A′C⊕AB. Besides, its QC is equal to five.

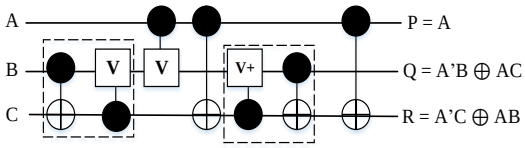



**Fig. (5):**Quantum circuit of the FRG gate [35]

**BHPF gate**: The BHPF gate is a 4×4 PPR block with output equations P = A,Q = A⊕B, R = B⊕C, and S = A⊕B⊕D[32]. The quantum circuit of the BHPF gate is illustrated in Fig. (6). Its QC is three.

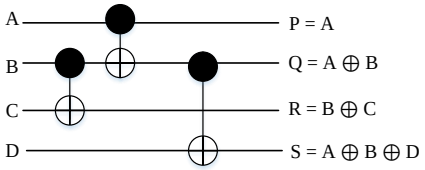



**Fig. (6):** Quantum circuit of the BHPF gate[32]

### 2.3 Division techniques

The division is considered as one of the most important operations in arithmetic computing. According to Eq. (8), a division algorithm is a method which, given dividend (D) and divisor (V), calculate their quotient (Q) and the remainder (R).

$$D = (Q \times V) + R \quad (8)$$

In the following, two-division algorithms will be examined.

### 2.3.1 Restoring division algorithm

One type of division method is the restoring division algorithm. The division is performed in each step based on the operation $R_{i+1} = 2R_i - V$. If the subtraction outcome is negative, the partial remainder value is calculated with the true value and restored in the addition operation $R_{i+1} = 2R_i + V$. Fig. (7) shows the flowchart of restoring division algorithm [36].

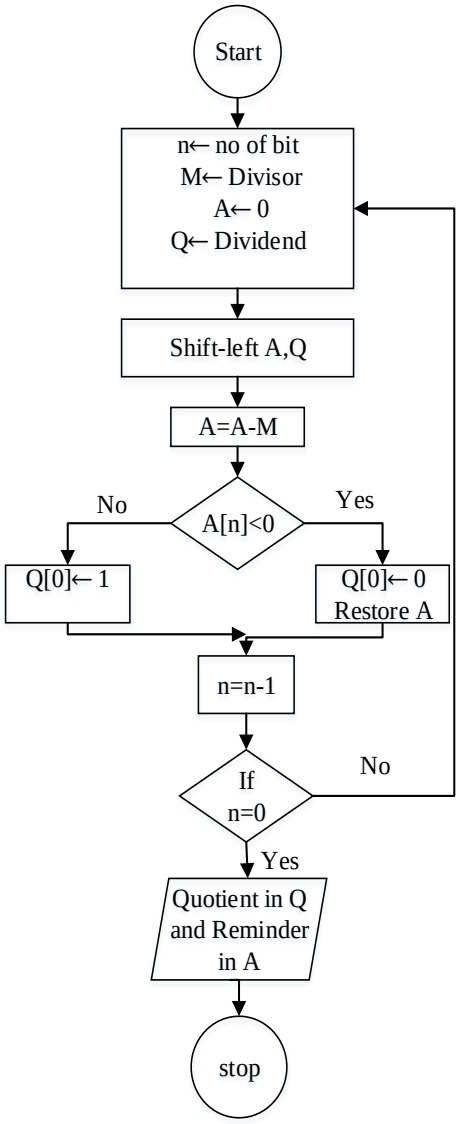



**Fig. (7):** Flowchart for restoring division algorithm [36]

The steps taken in the restoring algorithm are as follows [36]:

1. Save count = 0 and zero in register A.
2. Start repeating loops n times.
3. Shift to left A & Q.
4. Subtract M from A and store the answer in A.

5. If A <0, Set $Q_0 = 0$ and add M to A (restore A).
6. Else $Q_0 = 1$.
7. If count = n-1, then stop the loop.

### 2.3.2 Non-restoring algorithm

Another division method is the non-restoring division algorithm [15]. In this division method, if the subtraction outcome is negative, the partial remainder will not be stored immediately, and the continuation of the operation depends on the possibility of combining the computational step $R_i = R_i + V$. Partial remainder $R_{i+1} = 2R_i - V$ followed by calculating the partial remainder in the next step $R_{i+1} = 2R_i + V$ can be integrated into a single operation. Thus, if the quotient is $q_i = 1$, the next remainder is calculated by subtraction. If the quotient is $q_i = 0$, instead of storing the partial remainder value, the next step is performed by adding the divisor to the partial remainder. If the quotient is equal to 0, then the partial remainder will be negative due to subtraction, so an inherent correction is made by adding the divisor and the remainder [36]. Fig. (8) shows the flowchart for the non-restoring algorithm.

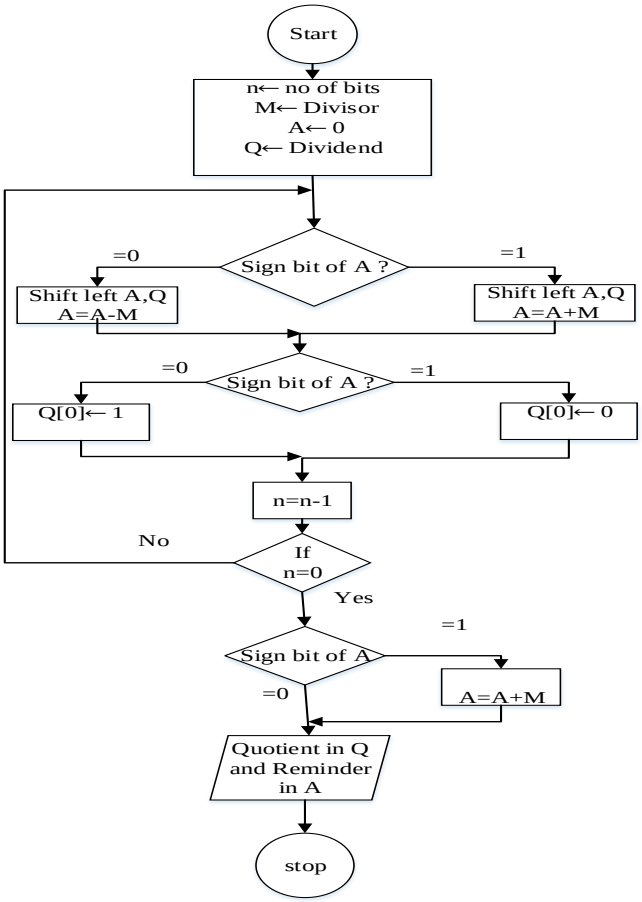



**Fig. (8):** Flowchart for non-restoring division algorithm [37]

## 3. The suggested PPR floating-point divider

This section describes the implementation of the proposed components of a PPR divider circuit-based reversible gates and blocks as follows. The ingredients of the reversible floating point divider circuit include

an RCA adder, reversible multiplexer, reversible register, PIPO left-shift register, rounding register, and normalization register.

### 3.1 The suggested PPR logic block

In order to design the circuits that are used in the proposed PPR divider. First, we propose a novel PPR block called TMB1. Its quantum representation is illustrated in Fig. (9). The suggested PPR block is a 5×5 block. The outputs are resulted via Eqs (9) to (13). The QC of the TMB1 block is eight.

$$P=A' \quad (9)$$
$$Q=(A\oplus B)' \quad (10)$$
$$R=A\oplus B\oplus C\oplus D \quad (11)$$
$$S= (A\oplus B)(C\oplus D)\oplus(AB\oplus D) \quad (12)$$
$$T=(A\oplus B)(C\oplus D)\oplus(A'B\oplus D)\oplus E \quad (13)$$

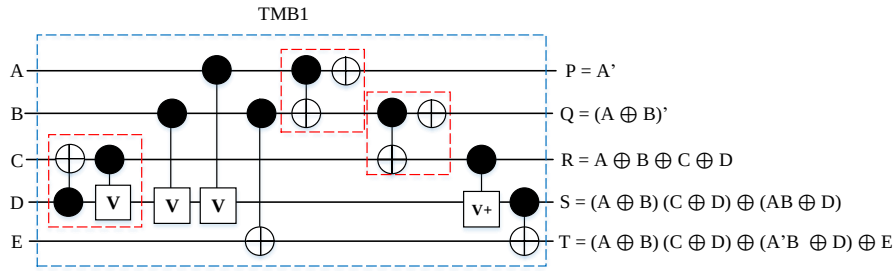



**Fig. (9):** The quantum circuit of the TMB1block

### 3.2 The suggested PPR adder

As mentioned, one of the consisting components of the proposed PPR divider is adder circuit. Here, we introduce an efficient PPR adder using the proposed TMB1 block. For this purpose, inputs D and E of the proposed TMB1 logic block are tuned to '0'; so the proposed block will turn into one-bit PPR full-adder. As can be seen in Fig. (10), the outputs Sum and Carry are resulted from the outputs R and S, respectively, and the remaining outputs of the circuit are garbage outputs. The QC of the PPR full-adder circuit is eight.

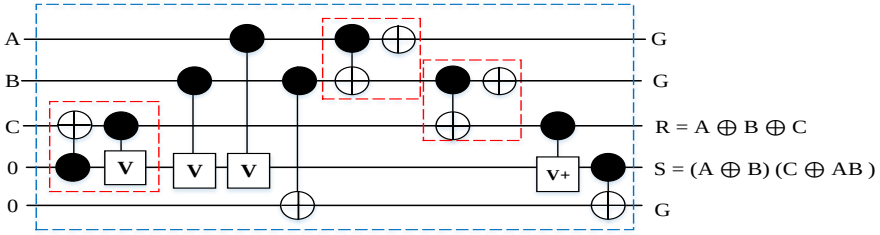



**Fig. (10):** The quantum circuit of the suggested one-bit PPR full-adder circuit using the TMB1 block

Fig. (11) shows an (n + 1)-bit PPR-RCA-adder using the suggested full-adder and the BHPF gate. The proposed circuit has (n + 1) gates, (n + 4) constant inputs, (3n + 2) garbage outputs, and a QC equal to (8n + 3).

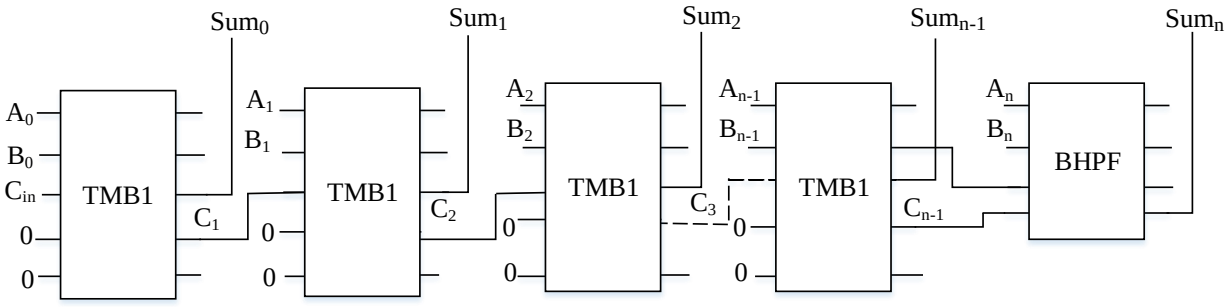



**Fig. (11):** The proposed (n + 1)-bit PPR-RCA-adder circuit

### 3.3 The suggested PPR D-latch

One of the consisting components of the divider circuits, in order to store intermediate results, is the latch circuit[38]. In the following, we first introduce PPR E1 block [39] and then apply it to design an effective D-Latch. Quantum circuit of the E1 block is illustrated in Fig. (12), with QC =6.

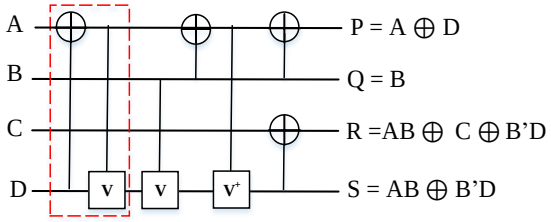



**Fig. (12):** The quantum circuit of the PPR E1 block [39]

As shown in Fig. (13), in order to design an effective PPR D-Latch, we set the inputs A, B, and C of the E1 block to D, CLK, and '0', respectively. Moreover, the output S is entered into the input D. The suggested E1 D-Latch has CI=1, GO=2, and QC=6.

$$Q_{t+1} = D \cdot CLK + \overline{CLK} \cdot Q_t \tag{14}$$

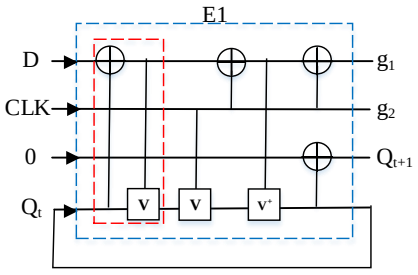



**Fig. (13):** The quantum circuit of the proposed E1 D-Latch

By connecting a cascade of *n* E1 D-Latches, as illustrated in Fig. (14), an n-bit PPR register can be resulted with criteria including CI=n, GO=n, and QC= 6n.

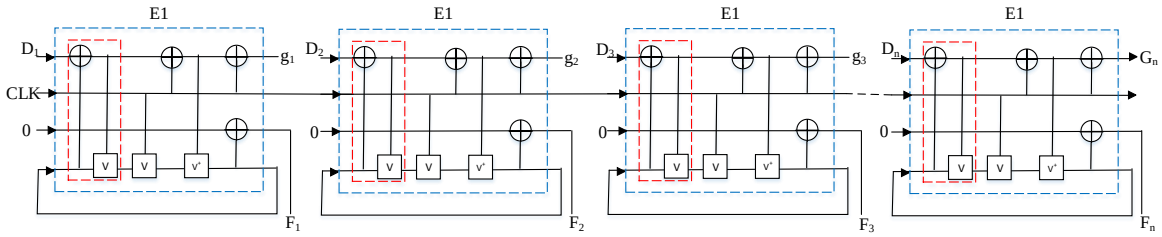



**Fig. (14):** The quantum circuit of the proposed n-bit register using the proposed E1 D-Latch

Moreover, we suggest a novel PPR rounding register using the proposed E1 D-Latch and the DFG gate (Fig. (15)). As illustrated, the proposed circuit has CI=2, GO=2, and QC=8.

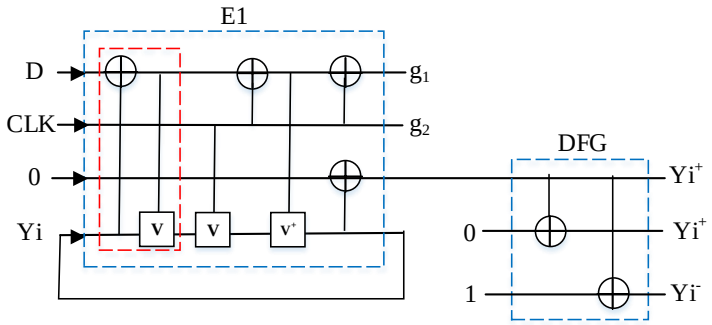



**Fig. (15):** The quantum circuit of the proposed PPR rounding register

Also, the proposed PPR normalization register designed using the E1 D-Latch is illustrated in Fig. (16). As can be seen, the proposed circuit has CI=1, GO=1, and QC=6.

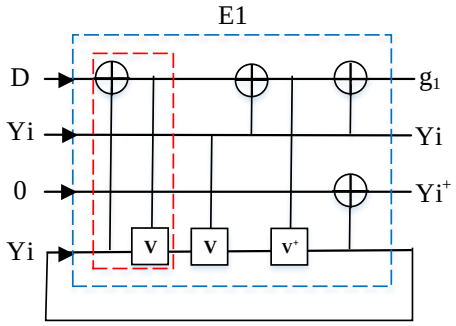
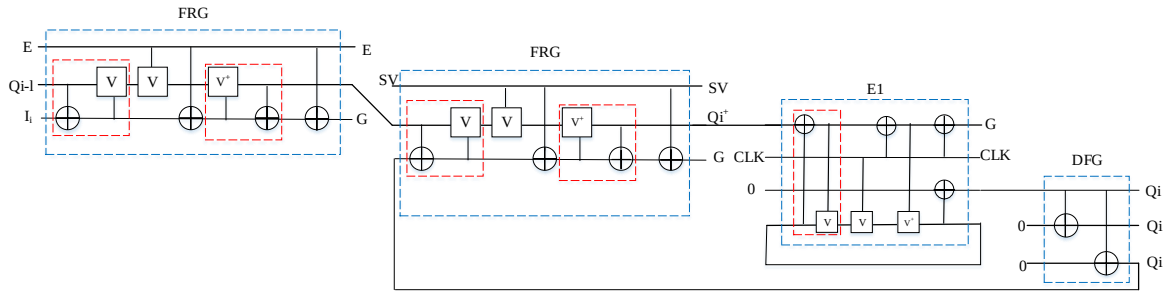


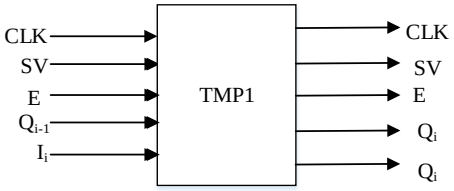

**Fig. (16):** The quantum circuit of the proposed PPR normalization register

### 3.4 The suggested PPR-PIPO left-shift register

In this section, we are going to propose a novel PPR-PIPO left-shift register using the proposed E1 D-Latch, called TMP1. In the proposed circuit, all bits will be loaded with one clock in the register, and after the shift operation, all bits can be sent to the output at once. The TMP1 consists of one D-latch, two FRG gates, and one DFG gate, as illustrated in Fig. (17a). The proposed design has CI=3, GO=4, and QC=18. The logical schema of the TMP1 cell is illustrated in Fig. (17b). By sequencing TMP1 cells, an n-bit PPR-PIPO left-shift register circuit can be designed and realized, as illustrated in Fig. (18).




(a)




(b)

**Fig. (17):** The proposed PPR-PIPO left-shift register, (a) quantum circuit, and (b) logical circuit of the TMP1 cell

As illustrated in Table 1, in the TMP1, the control values are determined based on SV and E, and implemented according to Eq. (15).

$$Q_i = SV' \cdot E \cdot I_i + SV' \cdot E' \cdot Q_{i-1} + SV \cdot Q_i \quad (15)$$

So that when the clock pulse is applied [15]:

1. If SV = 0 and E = 0, then the left-shift operation will be performed.
2. If SV = 0 and E = 1, the input bits will be sent parallel to the output.
3. If SV = 1, the PIPO register stores the current value, and there will be no change.

**Table 1:** The control functions of PIPO left-shift register inputs

| **Operations** | **Output $Q_i$** | **Control mode** | |
|---|---|---|---|
| | | **E** | **SV** |
| **Left-Shift** | $Q_{i-1}$ | 0 | 0 |
| **Parallel load** | $I_i$ | 1 | 0 |
| **No change** | $Q_i$ | × | 1 |

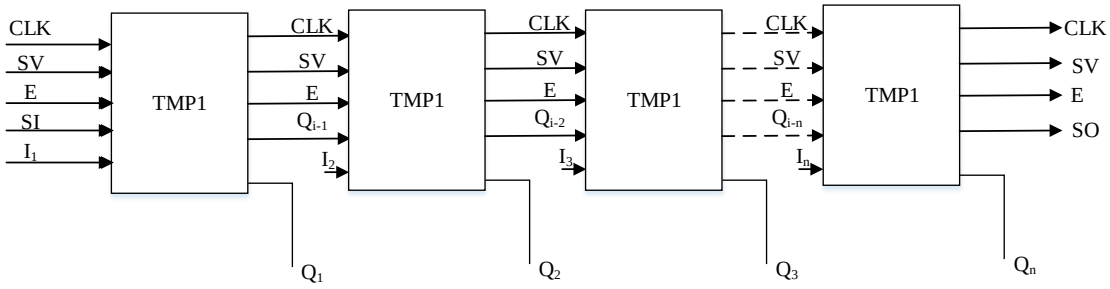



**Fig. (18):** The logical circuit of the proposed n-bit PIPO left-shift register circuit using TMP1 cell

### 3.5 PPR floating-point division circuit

In this section, the details of the proposed design of non-restoring reversible divider will be expressed. A logical block diagram of PPR floating-point divider based on the existing design in [17] is provided. Our proposed design in terms of Parametric and optimizing the divider circuit components effectively has been improved, which is shown in Fig. (19). As observed, it consists of following main modules:

- Two PPR multiplexers (MUXs)
- Two PPR-PIPO left-shift registers (TMP1)
- An n-bit PPR reversible register
- A PPR D-Latch (E1)
- A rounding register
- A normalization register for the divisor at each step (normalization means placing data into a domain when they are not in one domain).
- PPR RCA adder (TMB1)

In the initialization of the divider circuit, the initial values are zero. These values include B ($B_{n-1}$, $B_{n-2}$, …, $B_0$) = 0, S = 0, Dividend = D ($D_{n-1}$, $D_{n-2}$, …, $D_0$), Divisor = Y ($Y_{n-1}$, $Y_{n-2}$, …, $Y_0$), and Control = 0. Besides, the registers are the quotient = X ($Q_{n-1}$, $Q_{n-2}$, …, $Q_0$) and the remainder = Z ($B_{n-1}$, $B_{n-2}$, …, $B_0$). When the division operation starts, if select = 1, the inputs of the (n + 1)-bit multiplexer will be S = 0 and B ($B_{n-1}$, $B_{n-2}$, …, $B_0$) = 0, and the n-bit multiplexer will be equal to Dividend = D ($D_{n-1}$, $D_{n-2}$, …, $D_0$). When the clock pulse is applied, if SV2 = 0 and E = 1, the input is S1 = 1 and the output of the (n + 1)-bit multiplexer is placed in parallel in the (n + 1)-bit PIPO left-shift register. If E = 1 and SV1 = 0, the output of the n-bit multiplexer is placed in parallel in the n-bit PIPO left-shift register. If E = 0, both PIPO left-shift registers perform the shift operation. The SO output of the n-bit register X is connected to the SI of the (n + 1)-bit register Z. After the shift operation, the value S is transferred to S1 to detect the subtraction or addition operations that will be performed on Z and Y. If S1 = 1, then the operation Z + Y is calculated; else if S1 = 0, the operation Z-Y will be calculated, and the result of the addition or subtraction operation will be sent to the (n + 1)-bit RCA-adder. Then, when select = 0, the complement of the Most Significant Bit (MSB) of the adder is placed in bit $Q_0$ of register X, and with the next clock, the total computation of the adder shifts to register Z. The n-bit divisor is inserted into the rounding register via the F2G gate. In the next clock, the divisor bits are rounded against Y and placed in the normalization register through the other DFG gate. At each clock, the remainder of the division is obtained from the normalization register and placed in the (n + 1)-bit TMB1-adder. Thus, X stores the quotient value. If value S of Z register is equal to zero, then the remainder does not need to be restored, and B saves the remainder. If S= 1, the remainder needs to be restored. With the next clock pulse, if E = 1, the remaining value is stored in the (n + 1)-bit Z register. After restoring the remainder, S must be equal to zero. As a result, SV2 = 1, and B will store the remainder.

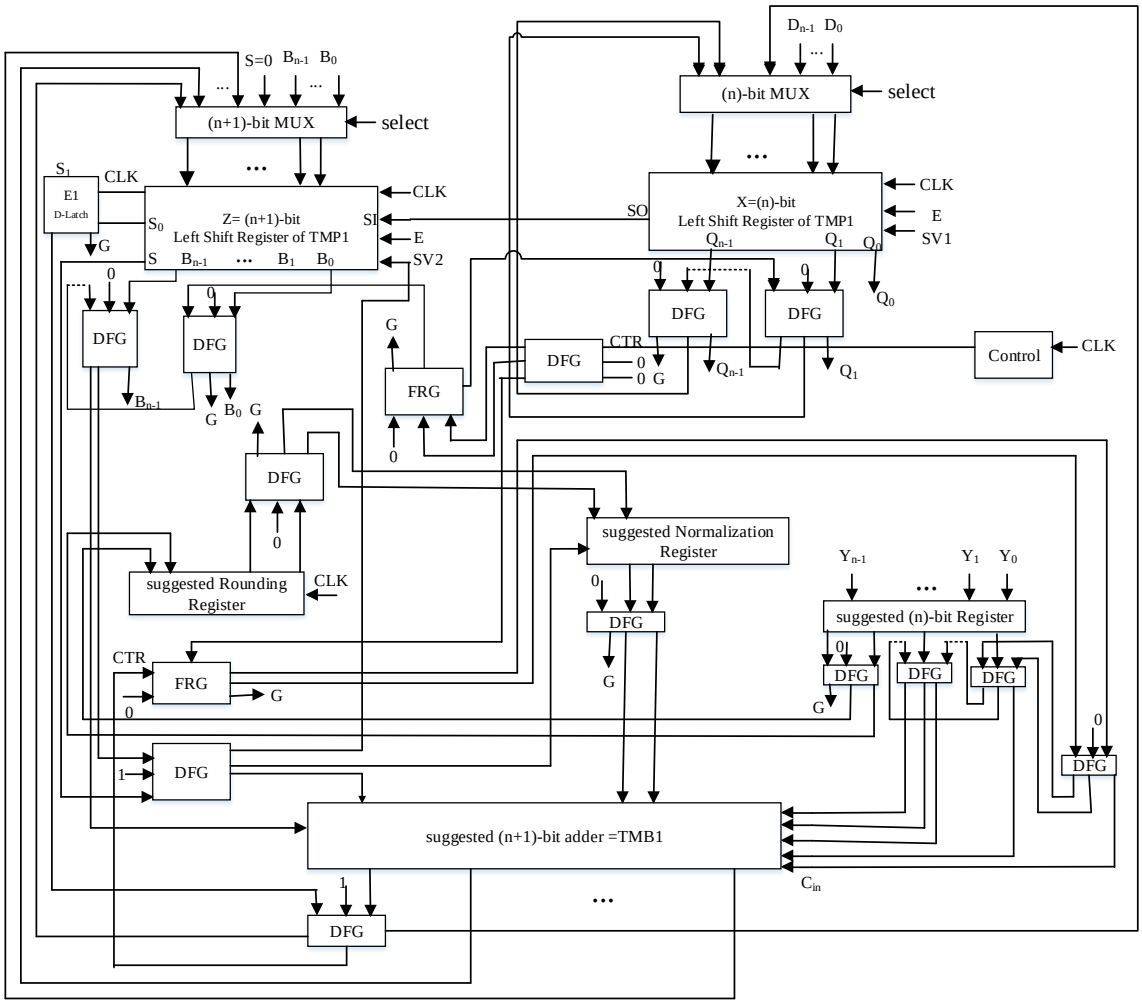



**Fig. (19)**: The n-bit PPR floating-point division circuit

## 4. Evaluations and comparisons

This section examines the details of the proposed designs. All the presented criteria were evaluated in terms of implementation, independent of technology.

### Evaluating PPR registers

As mentioned, the PPR register includes n E1 D-Latches sequenced in a cascade to develop an n-bit PPR register. Table 2 presents a comparison of the proposed n-bit PPR register with the existing design in [40] in terms of evaluation metrics.

**Table 2:** Comparison results of n-bit PPR registers

| Designs | CI | GO | QC |
|---|---|---|---|
| Proposed design | n | n | 6n |
| Design in [40] | n | n+1 | 7n |

As can be seen, although the proposed design is similar to the design [40] in terms of constant input, it is better from the point of view of garbage output and quantum cost.

**Evaluating PPR-PIPO left-shift registers (TMP1)**

This section compared the n-bit TMP1 with the two existing n-bit reversible designs developed in previous studies [16, 17]. Table 3 compares the proposed n-bit PPR-PIPO register with previous designs in terms of evaluation metrics.

**Table 3:** Comparison of PPR-PIPO left-shift registers

| Designs | CI | GO | QC |
|---|---|---|---|
| Proposed design | 3n | 3n + 2 | 18n |
| Design in [16] | 3n | 3n + 2 | 19n |
| Design in [17] | 3n | 3n + 1 | 22n |

As can be seen from Table 3, the suggested design is superior to the designs developed in [16, 17] in terms of quantum cost.

**Evaluating PPR-RCA-adder**

Comparison results of the proposed (n+1)-bit PPR-RCA-adder with previous designs in [16, 17, 41] are given in Table 4.

**Table 4:** Comparison of (n+1)-bit PPR-RCA-adder circuits

| Designs | CI | GO | QC |
|---|---|---|---|
| Proposed design | n + 4 | 3n + 2 | 8n + 3 |
| Existing in [16] | 2n | 3n + 2 | 14n + 4 |
| Existing in [17] | n + 2 | 3n + 2 | 9n + 3 |
| Existing in [41] | 2n | 3n + 3 | 12n + 12 |

As can be observed, the proposed design is better than previous designs in [16] and [41] from quantum-cost metrics viewpoints. Moreover, although the proposed design is close to the existing design in [17] in terms of constant inputs and garbage outputs, it is better than them from quantum cost.

**Evaluating n-bit PPR floating-point divider**

This section evaluates the suggested n-bit PPR floating-point divider. It can be claimed that the suggested structure in this study is more efficient and cost-effective compared to other designs in [16, 17] that have been implemented using a similar approach. The evaluation parameters for the proposed scheme are QC = 66n + 68, GO = 15n + 18 and CI = 13n + 18. The evaluation of the circuit parameters is equal to the sum of the evaluation of all the components of the divider circuit as follows:

1. The n-bit PPR MUX: CI = n, GO=n, and QC = 5n
2. The (n+1)-bit PPR MUX: CI = n, GO=n, and QC = 5n + 5
3. The n-bit PPR register: CI = n, GO = n, and QC = 6n
4. The n-bit PPR-PIPO left-shift register: CI = 3n, GO=3n+2, and QC = 18n
5. The (n +1)-bit PPR-PIPO left-shift register: CI = 3n, GO = 3n+5, and QC = 18n + 18
6. The PPR E1 D-Latch: CI = 1, GO = 2, and QC = 6
7. The PPR rounding register: CI = 3, GO=2, and QC = 8

8. The PPR normalization register: CI = 1, GO=1, and QC = 6
9. The n-bit PPR-RCA-adder: CI = n + 4, GO = 3n + 2, and QC = 8n + 3

Other gates/blocks:

10. FRG: CI = 2, GO=2, and QC = 10
11. DFG: CI = 3n + 7, GO=3n+2, and QC = 6n + 12

Also, the components specifications of the proposed divider are summarized in Table 5.

**Table 5:** Component characteristics of the suggested n-bit PPR divider

| Designs | CI | GO | QC |
|---|---|---|---|
| (n)-bit MUX | n | n | 5n |
| (n+1)-bit MUX | n | n | 5n + 5 |
| (n)-bit register | n | n | 6n |
| (n)-bit PIPO left-shift register | 3n | 3n + 2 | 18n |
| (n+1)-bit PIPO left-shift register | 3n | 3n + 5 | 18n + 18 |
| D-Latch | 1 | 2 | 6 |
| Round register | 3 | 2 | 8 |
| Normalization register | 1 | 1 | 6 |
| (n+1)-bit PPR-RCA-adder | n + 4 | 3n + 2 | 8n + 3 |
| Other interface gates | 3n + 9 | 3n + 4 | 6n + 22 |

Table 6 compares the suggested PPR floating–point divider with the schemes introduced in [16, 17].

**Table 6:** Comparison of different- n-bit PPR divider circuits

| Designs | CI | GO | QC |
|---|---|---|---|
| Proposed design | 13n + 18 | 15n + 18 | 66n + 68 |
| Existing design in [16] | 11n + 14 | 12n + 20 | 75n + 60 |
| Existing design in [17] | 10n + 17 | 12n + 17 | 67n + 69 |

As seen, although the proposed scheme is close to previous designs in terms of constant inputs and garbage outputs viewpoints, it is superior to them in terms of quantum cost.

## 5. Conclusion

Divider circuits play an important role in developing computational units such as arithmetic logic units. The floating-point divider is one of the most complex computational circuits implemented in digital systems and the design of microprocessors. We proposed an efficient PPR floating-point divider, which can be used in digital systems to improve efficiency significantly. This paper introduced new designs for various components of a PPR floating-point divider circuit using the proposed TMB1 block and the proposed PPR E1 D-Latch. The proposed circuits were utilized to design various circuits, including an (n+1)-bit PPR-RCA-adder, an (n +1)-bit PPR register, an (n+1)-bit PPR-PIPO left-shift register, a PPR rounding register, and a PPR normalization register. Finally, using the proposed components, a PPR floating-point divider was introduced with the evaluation parameters, including QC=66n+68, GO=15n+18, and CI=13n+18. The evaluation results indicated that the proposed design has improved compared to the existing counterparts.

**Authors contributions.** All authors contributed equally to the study conception and design. All authors read and approved the final manuscript.

**Data Availability**. Data sharing not applicable to this article as no datasets were generated or analyzed during the current study.

**Conflicts of interest/Competing interest.** The authors have no financial or proprietary interests in any material discussed in this article.

**References**


[1] R. Landauer, "Irreversibility and heat generation in the computing process," *IBM journal of research and development,* vol. 5, no. 3, pp. 183-191, 1961.

[2] G. E. Moore, "Cramming more components onto integrated circuits," ed: McGraw-Hill New York, NY, USA:, 1965.

[3] C. H. Bennett, "Logical reversibility of computation," *IBM journal of Research and Development,* vol. 17, no. 6, pp. 525-532, 1973.

[4] E. P. Blair, E. Yost, and C. S. Lent, "Power dissipation in clocking wires for clocked molecular quantum-dot cellular automata," *Journal of computational electronics,* vol. 9, no. 1, pp. 49-55, 2010.

[5] M. A. Nielsen and I. L. Chuang, "Quantum computing and quantum information," ed: Cambridge University Press, Cambridge, 2000.

[6] R. C. Merkle, "Two types of mechanical reversible logic," *Nanotechnology,* vol. 4, no. 2, p. 114, 1993.

[7] M. Haghparast, M. Mohammadi, K. Navi, and M. Eshghi, "Optimized reversible multiplier circuit," *Journal of Circuits, Systems, and Computers,* vol. 18, no. 02, pp. 311-323, 2009.

[8] M. Perkowski *et al.*, "A general decomposition for reversible logic," 2001.

[9] A. Banan, A. Nasiri, and A. Taheri-Garavand, "Deep learning-based appearance features extraction for automated carp species identification," *Aquacultural Engineering,* vol. 89, p. 102053, 2020.

[10] Y. Fan, K. Xu, H. Wu, Y. Zheng, and B. Tao, "Spatiotemporal modeling for nonlinear distributed thermal processes based on KL decomposition, MLP and LSTM network," *IEEE Access,* vol. 8, pp. 25111-25121, 2020.

[11] H. A. Afan *et al.*, "Modeling the fluctuations of groundwater level by employing ensemble deep learning techniques," *Engineering Applications of Computational Fluid Mechanics,* vol. 15, no. 1, pp. 1420-1439, 2021.

[12] W. Chen, D. Sharifrazi, G. Liang, S. S. Band, K. W. Chau, and A. Mosavi, "Accurate discharge coefficient prediction of streamlined weirs by coupling linear regression and deep convolutional gated recurrent unit," *Engineering Applications of Computational Fluid Mechanics,* vol. 16, no. 1, pp. 965-976, 2022.

[13] S. Shamshirband, T. Rabczuk, and K.-W. Chau, "A survey of deep learning techniques: application in wind and solar energy resources," *IEEE Access,* vol. 7, pp. 164650-164666, 2019.

[14] P. Safari, M. Haghparast, A. Azari, and A. Branch, "A design of fault tolerant reversible arithmetic logic unit," *Life Science Journal,* vol. 9, no. 3, pp. 643-646, 2012.

[15] N. M. Nayeem, A. Hossain, M. Haque, L. Jamal, and H. M. H. Babu, "Novel reversible division hardware," in *2009 52nd IEEE International Midwest Symposium on Circuits and Systems*, 2009: IEEE, pp. 1134-1138.

[16] F. Dastan and M. Haghparast, "A novel nanometric fault tolerant reversible divider," *International Journal of Physical Sciences,* vol. 6, no. 24, pp. 5671-5681, 2011.

[17] H. M. H. Babu and M. S. Mia, "Design of a compact reversible fault tolerant division circuit," *Microelectronics Journal,* vol. 51, pp. 15-29, 2016.

[18] A. Sarker, H. M. H. Babu, and S. M. M. Rashid, "Design of a DNA-based reversible arithmetic and logic unit," *IET nanobiotechnology,* vol. 9, no. 4, pp. 226-238, 2015.

[19] B. Parhami, "Fault-tolerant reversible circuits," in *2006 fortieth asilomar conference on signals, systems and computers*, 2006: IEEE, pp. 1726-1729.

[20] W. N. Hung, X. Song, G. Yang, J. Yang, and M. Perkowski, "Quantum logic synthesis by symbolic reachability analysis," in *Proceedings of the 41st annual Design Automation Conference*, 2004, pp. 838-841.

[21] M. Noorallahzadeh and M. Mosleh, "Parity-preserving reversible flip-flops with low quantum cost in nanoscale," *The Journal of Supercomputing,* pp. 1-33, 2019.

[22] A. Bolhassani and M. Haghparast, "Optimised reversible divider circuit," *International Journal of Innovative Computing and Applications,* vol. 7, no. 1, pp. 13-33, 2016.

[23] H. Thapliyal and N. Ranganathan, "Design of reversible sequential circuits optimizing quantum cost, delay, and garbage outputs," *ACM Journal on Emerging Technologies in Computing Systems (JETC),* vol. 6, no. 4, pp. 1-31, 2010.

[24] R. P. Feynman, "Quantum mechanical computers," *Optics news,* vol. 11, no. 2, pp. 11-20, 1985.

[25] A. Barenco *et al.*, "Elementary gates for quantum computation," *Physical review A,* vol. 52, no. 5, p. 3457, 1995.

[26] M. Morrison and N. Ranganathan, "Design of a reversible ALU based on novel programmable reversible logic gate structures," in *2011 IEEE computer society annual symposium on VLSI*, 2011: IEEE, pp. 126-131.

[27] M. Morrison and N. Ranganathan, "A novel optimization method for reversible logic circuit minimization," in *2013 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, 2013: IEEE, pp. 182-187.

[28] D. M. Miller, M. Soeken, and R. Drechsler, "Mapping NCV circuits to optimized Clifford+T circuits," in *International Conference on Reversible Computation*, 2014: Springer, pp. 163-175.

[29] M. Noorallahzadeh and M. Mosleh, "Efficient designs of reversible BCD to EX-3 Converter with low quantum cost in nanoscale," *International Journal of Quantum Information,* vol. 18, no. 05, p. 2050020, 2020.

[30] M. S. Gharajeh and M. Haghparast, "On design of a fault tolerant reversible 4-bit binary counter with parallel load," in *Australian Journal of Basic and Applied Sciences*, 2012: Citeseer.

[31] R.-G. Zhou, Y.-C. Li, and M.-Q. Zhang, "Novel designs for fault tolerant reversible binary coded decimal adders," *International Journal of Electronics,* vol. 101, no. 10, pp. 1336-1356, 2014.

[32] M. Haghparast and A. Bolhassani, "On design of parity preserving reversible adder circuits," *International Journal of Theoretical Physics,* vol. 55, no. 12, pp. 5118-5135, 2016.

[33] N. K. Misra, B. Sen, S. Wairya, and B. Bhoi, "Testable novel parity-preserving reversible gate and low-cost quantum decoder design in 1D molecular-QCA," *Journal of Circuits, Systems and Computers,* vol. 26, no. 09, p. 1750145, 2017.

[34] S. R. Arabani, M. R. Reshadinezhad, and M. Haghparast, "Design of a parity preserving reversible full adder/subtractor circuit," *International Journal of Computational Intelligence Studies,* vol. 7, no. 1, pp. 19-32, 2018.

[35] E. Fredkin and T. Toffoli, "Int'l J. of Theoretical Physics 21 (3/4)," *Conservative logic,* pp. 219-253, 1982.

[36] S. Kaur, M. Singh, and R. Agarwal, "Vhdl implementation of non restoring division algorithm using high speed adder/subtractor," *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering,* vol. 2, no. 7, pp. 3317-3324, 2013.

[37] B. K. Bhoi, N. K. Misra, and M. Pradhan, "Synthesis and simulation study of non-restoring cell architecture layout in perpendicular nano-magnetic logic," *Journal of Computational Electronics,* vol. 19, no. 1, pp. 407-418, 2020.

[38] M. Mohammadi, M. Eshghi, and M. Haghparast, "On design of multiple-valued sequential reversible circuits for nanotechnology based systems," in *TENCON 2008-2008 IEEE Region 10 Conference*, 2008: IEEE, pp. 1-6.

[39] E. PourAliAkbar, K. Navi, M. Haghparast, and M. Reshadi, "Novel Optimum Parity-Preserving Reversible Multiplier Circuits," *Circuits, Systems, and Signal Processing,* vol. 39, no. 10, pp. 5148-5168, 2020.

[40] M. Haghparast and K. Navi, "Novel reversible fault tolerant error coding and detection circuits," *International Journal of Quantum Information,* vol. 9, no. 02, pp. 723-738, 2011.

[41] A. Kamaraj and P. Marichamy, "Design of fault-tolerant reversible floating point division," *Informacije MIDEM,* vol. 48, no. 3, pp. 161-172, 2018.