Received 24 May 2023; revised 16 July 2023 and 2 August 2023; accepted 13 August 2023. Date of publication 15 August 2023;
date of current version 1 September 2023.

*Digital Object Identifier 10.1109/OJCAS.2023.3305557*

# Reversible Gates: A Paradigm Shift in Computing

**SYED FARAH NAZ (Student Member, IEEE), AND AMBIKA PRASAD SHAH (Senior Member, IEEE)**

IC-ResQ Lab., Department of Electrical Engineering, Indian Institute of Technology Jammu, Jammu and Kashmir 181221, India

This article was recommended by Associate Editor Y. Li.

CORRESPONDING AUTHOR: A. P. SHAH (e-mail: ambika.shah@iitjammu.ac.in)

This work was supported in part by the Prime Minister's Research Fellows (PMRF) Scheme, Government of India and in part by the FIST Scheme of DST, Government of India under Grant SR/FST/ET-II/2021/814.

**ABSTRACT** The reversible gate has been one of the emerging research areas that ensure a continual process of innovation trends that explore and utilizes the resources. This review paper provides a comprehensive overview of reversible gates, including their fundamental principles, design methodologies, and various applications. It also analyzes the reversible gates, comparing them based on metrics such as Quantum Cost, Complexity, and other performance evaluation measures. The analysis of several reversible gates is presented in this paper and provides a comprehensive overview of reversible gates, encompassing their fundamental principles, design methodologies, and diverse applications. Reversible logic circuits allow for the production of both unique outputs and distinct input combinations. The majority of the findings about the reversible gates from previous research papers are discussed and contrasted. All the reversible gates that have been proposed till now are presented in tabular form and the parameters are discussed to help the researchers to find every detail related to the reversible gates. To highlight our understanding, we have ended most of the sections with questions. The inclusion of questions is likely intended to stimulate further discussion and promote a deeper understanding of the material presented in this paper. These questions can serve as prompts for readers to reflect on the content and potentially explore related research directions or areas of improvement.

**INDEX TERMS** Reversible logic, quantum computing, mapping, energy loss, fault-tolerant gates, parity preserving gates.

## I. INTRODUCTION

QUANTUM computation is a rapidly growing field of study that explores the use of quantum mechanical phenomena, such as superposition and entanglement, to perform computations that would be difficult or impossible with classical computers. One important aspect of quantum computation is the use of quantum logic gates, which are the building blocks of quantum circuits. Quantum logic gates operate on qubits, which are quantum bits that can exist in a superposition of states, unlike classical bits which are either 0 or 1. Quantum logic gates can perform operations on these superposed qubits, allowing for the simultaneous computation of multiple possibilities. For example, the Hadamard gate can create a superposition of two possible states, while the CNOT gate can entangle two qubits, creating complex quantum states. There are several types of quantum logic gates, including single-qubit gates and multi-qubit gates, which are used to create more complex quantum circuits. These gates are designed to manipulate the states of the qubits, performing operations that are reversible and unitary [1].

In terms of logic, quantum computation uses a different approach than classical computing. In classical computing, Boolean logic is used to represent logical states as either 0 or 1, while in quantum computing, qubits can exist in a superposition of states. In quantum computers, the information is switched with the help of change in quantum states. These states are defined via linear combinations, called as superposition for a single qubit system as described in Eq. (1).

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \tag{1}$$

where $\alpha$ and $\beta$ are the two complex numbers and $|0\rangle$, $|1\rangle$ indicates the Dirac notations of the two states. There are only two quantum states $|0\rangle$ and —$|1\rangle$ that can be readily

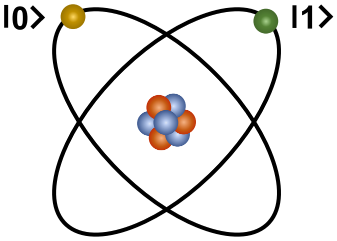


FIGURE 1. Qubit representation as two electronic levels [2].

distinguished for carrying out any operation out of the limitless quantum information space (Hilbert Space) in the superposition. As seen in Fig. 1, these states refer to the ground state and excited state of electrons in an atom, respectively. The qubit once propagated from the input will not be taken back and cannot be taken as inputs to numerous stages in a quantum system. This means that quantum computing is capable of performing computations on multiple possibilities simultaneously, leading to an exponential speedup in certain types of problems. In conclusion, quantum computation and logic are exciting and rapidly growing fields of study with a wide range of potential applications, from cryptography and optimization to simulation and drug discovery. While the principles of quantum mechanics bring unique challenges to quantum computing, the potential for exponential speedup in certain types of problems makes it an area of active research and development. According to Landauer [3], for irreversible logic computations, every bit of information lost results in the production of KTln2 Joules of heat energy, where k is the Boltzmann constant and T is the absolute temperature at which the computation is performed. As a result, the number of bits lost during processing directly correlates with the quantity of energy lost in a system. Bennett [4] further demonstrated that reversible computation prevents the kTln2 energy dissipation from happening. Circuits that are reversible and don't lose information can be used to perform reversible computations. Logic gates which are reversible that generate distinct output vectors from each input vector and vice versa, are used to design reversible circuits. In other words, there is a one-to-one mapping between the input and the output vectors. As there is no information loss in the reversible logic gates, they are used for low-power VLSI circuits working even beyond the thermodynamic limitations of computing [5]. Reversible logic has also shown promise in quantum computing [6], [7], [8].

Reversible computing is a computing paradigm that aims to minimize energy consumption by ensuring that all operations performed by a computer can be undone without any loss of information. In conventional computing, many operations are inherently irreversible, such as erasing a bit or dissipating heat as a result of a computation. These irreversibilities lead to energy losses and ultimately limit the energy efficiency of computing systems. In reversible computing, computations are performed using reversible logic gates, which can be implemented using various physical systems such as quantum systems, classical digital circuits, or analog electronic circuits. These reversible gates are designed to ensure that the inputs of the gate can be uniquely determined from its outputs, and vice versa, which enables all computations to be fully reversible [7], [9]. One of the key advantages of reversible computing is its potential to reduce energy consumption. By avoiding the dissipation of energy during computation, reversible computing systems can in theory achieve near-zero energy consumption. This makes reversible computing an attractive option for energy-constrained systems, such as mobile devices and sensor networks, as well as for large-scale computing systems where energy consumption is a significant concern. Researchers are currently investigating the potential applications of certain dominant technologies in physical foregrounds. These applications include superconductors, nuclear magnetic resonance, photonics, quantum dots, trapped Ions, DNA computation, etc. [2].

However, reversible computing also has some limitations. One major challenge is the design and implementation of reversible logic gates, which can be more complex and require additional resources compared to their irreversible counterparts [10]. In addition, reversible computing is still an emerging field, and there are many practical challenges that need to be addressed before reversible computing can become a mainstream technology. Despite these challenges, reversible computing is an important area of research with the potential to revolutionize the field of computing. As we continue to explore and develop new reversible computing technologies, we may be able to create more energy-efficient computing systems that have a smaller environmental impact and can be used in a wide range of applications.

In reversible computing, the aim is to minimize the quantum cost, delay, and quantity of garbage outputs (GO), as these factors have a direct impact on the overall performance of the reversible logic circuit. The quantum cost metric represents the number of quantum gates required to implement a reversible circuit, while the delay metric represents the time it takes for the circuit to produce an output. The GO, as mentioned above, are unutilized outputs that do not carry out any meaningful operations, but are necessary to preserve the reversibility of the circuit [11]. Additionally, the restriction on feedback in reversible computing limits the design of reversible logic circuits to combinational circuits. This means that sequential circuits and feedback loops, which are commonly used in classical computing, are not permitted in reversible computing. However, this restriction has motivated researchers to explore new techniques for designing reversible logic circuits that are efficient and have low quantum costs, delays, and GO. The restriction on feedback in [12] was first proposed by Edward Fredkin, who argued that it was necessary to preserve reversibility in computing. However, Toffoli's work showed that feedback can be permitted in reversible computing, as long as the combinational part of the circuit is reversible [12].

Toffoli's work challenged the notion that feedback is not permitted in reversible computing and opened up new avenues for research in the design of reversible logic circuits [12]. His results showed that by carefully designing the combinational part of a circuit, it is possible to introduce feedback and still maintain reversibility. This has led to the development of more advanced reversible logic circuits that are capable of implementing complex operations while still maintaining low quantum costs, delays, and GO. Overall, Toffoli's work has had a significant impact on the field of reversible computing and has helped to push the boundaries of what is possible in this area.

Due to the requirement that quantum evolution be reversible, reversible circuits can be thought of as a specific example of quantum circuits [6]. The "circuit rules" for classical (non-quantum) reversible gates are the same whether they operate on classical bits or quantum states. Universal gate libraries for classical reversible computation are frequently subsets of prominent universal gate libraries for quantum computation. Although pure quantum gates are necessary for the speedups that make quantum computing appealing, logic synthesis for classical reversible circuits is the first step toward the synthesis of quantum circuits. To highlight our understanding or misunderstanding we have ended most sections with questions.

The organization of this paper is as follows: Section II discusses the motivation regarding the reversible gates, Section III discusses the parameters and features of Reversible Gates including logical primitives, delay calculation, reversibility and invertibility, universality followed by maximum reversibility depth and implementation of 13 Standard Boolean Functions using reversible gates. Section IV discusses the implementation methodology of reversible gates. Section V discusses the reversible gates and their classifications including Quantum Cost and Complexity of Reversible Gates and Parity Preserving Reversible Gates. Section VI discusses the future scope of the reversible gates and Section VII concludes the paper.

## II. MOTIVATION

A quantum network or family of quantum networks made up of quantum logic gates will be considered as the quantum computer. Each gate will perform an elementary unitary operation on one, two, or more qubits, that are two-state quantum systems. Each qubit, which corresponds to the traditional bit values 0 and 1, represents a basic unit of information. Since all unitary operations are reversible, quantum networks must be constructed using reversible logic elements [7]. The ability to execute processing at the atomic level using quantum computers, which is seen to be one of the most promising computing paradigms, makes it possible to push above the semiconductor industry's current boundaries. Potentially intractable non-deterministic polynomial time problems that cannot be solved by conventional computers may be solved using quantum computer methods. This could imply that once quantum computers are used in crucial disciplines like biotechnology, nanomedicine, secure computing, etc., hitherto unsolvable problems could be resolved. As a result, the viability of reversible logic circuits may have a crucial impact on the development of quantum computing.

The von Neumann-Landauer (VNL) principle, named after its discoverers, states that when a calculation is carried out irreversibly, 1 bit's worth of lost logical information always results in at least KTln2 of physical energy dissipation [13]. Bennett demonstrated that KTln2 energy dissipation would not happen if a calculation is done in a reversible method [4] from a thermodynamic perspective to escape this restriction. Therefore, even if reliability difficulties could be disregarded, a firm lower limit on $\mathrm{E_{diss}} = \mathrm{KTln2}$ ($\sim$ 18meV) dissipation (in a room-temperature environment) is required for conventional (irreversible) logic. If the logic is also physically reversible in its physical implementation, reversible logic can be advantageous in the construction of non-dissipative circuits. Since CMOS is not physically reversible, it cannot be regarded as a platform for practical application. Voltage-coded logic signals in contemporary CMOS technology contain an energy of $\mathrm{E} = \frac{1}{2}\mathrm{CV}^2$, where C is the capacitance and V is the node voltage, which dissipates whenever the node voltage changes and is orders of magnitude larger than the KTln2 factor.

Reversible logic can significantly reduce energy dissipation in new nanotechnologies such as Quantum Dot Cellular Automata (QCA) computing, Optical Computing, and Superconductor Flux Logic (SFL) family, among others [14], [15], [16], [17], [18] by ensuring that no information is lost during computation, thereby avoiding the waste of energy that would otherwise contribute to heat dissipation. In conventional logically irreversible designs, the energy dissipation per logic operation is close to a few KTln2. However, by using reversible logic techniques, the energy dissipation can be reduced significantly below this limit. This will not only make these nanotechnologies more efficient but also more environmentally friendly.

Furthermore, reversible logic can provide a solution for building ultra-low power circuits, which is a critical requirement for the development of wearable and portable devices. These devices typically have limited battery life and are required to operate for extended periods without charging. Reversible logic can enable the design of circuits that consume less power, thereby increasing the battery life of these devices. Thus, reversible logic is an important technique that can significantly reduce energy dissipation and enable the development of more efficient and environmentally friendly nanotechnologies. The key difference between classical logic gates and reversible logic gates is that classical logic gates are irreversible, while reversible logic gates are bijective. This means that, in classical logic gates, the input vector states cannot be uniquely reconstructed from the output vector states [7].

This property of bijectivity is crucial for reversible computing because it allows for the preservation of information during computation, which is essential for quantum computing and other areas of computing where the conservation of information is critical. In addition, by preserving information, reversible computing can minimize the quantum cost, delay, and GO, making it an attractive approach for the design of efficient and low-power computing systems.

*Question*: *What are some of the main challenges associated with designing and implementing reversible gates, and how can they be addressed?*

### A. RELATIONSHIP BETWEEN QUANTUM COMPUTING AND REVERSIBLE COMPUTING

Quantum computing and reversible computing are closely related concepts, as reversible computing forms the foundation for quantum computing. There are key connections between the two. Both quantum computing and reversible computing rely on reversible logic gates. Reversible gates are fundamental building blocks in reversible circuits, where each gate operation is invertible, allowing for information to be recovered and computation to be undone [7]. In quantum computing, reversible gates play a crucial role in implementing quantum operations on qubits.

Quantum computing takes advantage of a fundamental property of quantum mechanics called superposition. Superposition allows qubits in a quantum computer to be in a state of both 0 and 1 simultaneously, enabling parallel computations. Reversible computing also makes use of superposition, although on a more limited scale compared to quantum computing. Quantum computing introduces specific gates called quantum gates that operate on qubits and enable the manipulation of quantum information. Quantum gates, similar to reversible gates, are required to be reversible and unitary to preserve the reversible nature of quantum computation [19]. Both reversible computing and quantum computing emphasize the conservation of information during computation. Reversible computing strives to minimize information loss, allowing computations to be undone and preserving the bijectivity property. Quantum computing takes this concept further, where information is conserved and entanglement between qubits allows for the reversible transformation of quantum states.

Reversible computing and quantum computing share a common interest in energy efficiency. Reversible computing aims to minimize energy consumption by ensuring that all operations can be reversed, thereby reducing energy dissipation due to information loss [20]. Quantum computing also seeks energy efficiency by taking advantage of reversible gates and the inherent conservation of information in quantum operations. While quantum computing extends the principles of reversible computing into the realm of quantum mechanics, it also introduces additional concepts such as superposition, entanglement, and quantum algorithms. Nonetheless, reversible computing provides the foundational framework for designing and understanding the reversible and information-conserving aspects of quantum computing.

*Question*: *What is the impact of reversible computing principles on the development and implementation of quantum computing systems?*

## III. PARAMETERS AND FEATURES OF REVERSIBLE GATES

Reversible gates are a type of logic gate in which every input combination produces a unique output combination, and every output combination corresponds to a unique input combination. This property makes reversible gates important in quantum computing, where the reversible nature of the gates is necessary to prevent the loss of quantum information. Other parameters of the reversible gates are discussed in the following subsections.

### A. LOGICAL PRIMITIVES

A logic element is a building block used to create the logic circuits that power computing systems. There are two different kinds of logic components: one without memory, known as a logic gate, and one with memory. Reversible logic elements with memory (RLEMs) play a critical role in the design of reversible computing systems. Unlike traditional logic gates, RLEMs combine both memory and computational capabilities, making them a versatile building block for a wide range of reversible computing systems [19]. These include reversible Turing machines, reversible sequential machines, and other types of reversible computing systems. The study of RLEMs has shown that even the most basic RLEMs have universality, meaning that they can perform any computational task that can be performed by a traditional computer. This makes RLEMs an important component in the development of new and efficient computing systems. The construction of reversible computing systems using RLEMs is also different from traditional logic circuit construction. In traditional circuit design, logic gates like AND, OR, NOT, NAND, etc. are treated as separate components from memory components like flip-flops. However, in the construction of reversible computing systems, RLEMs serve as the primary building blocks, combining both computational and memory capabilities into a single component.

In conclusion, RLEMs play a crucial role in the design of reversible computing systems and have shown to be versatile and efficient building blocks for a wide range of reversible computing systems. The study of RLEMs and their use in the construction of reversible computing systems is an ongoing and exciting area of research [19]. Reversible computing is a field of study that aims to find effective techniques for building reversible machines using reversible logic elements and to realize reversible logic elements using physically reversible phenomena. Reversible logic is defined as an aspect of logic that operates according to an injection, meaning that the mapping between inputs and outputs is one-to-one. Reversible logic components, like NOT gates, are considered to be reversible because they realize injective logical functions. On the other hand, irreversible logic components, like NAND gates and flip-flops, realize non-injective logical functions and cannot be used to build reversible computers directly [7].

However, reversible Turing machines and other reversible computer models can be built from irreversible logic components, but this is considered essentially pointless as reversible computing aims to identify a critical component between the abstract level of machines and the physical level of systems. Thus, reversible computing aims to find a critical component between abstract machines and physical systems and to develop techniques for building reversible machines using reversible logic elements. The study of RLEMs is an important part of this field of research and has the potential to lead to the development of more efficient and versatile computing systems.

*Question*: *What are some of the main challenges associated with implementing logical primitives, such as arithmetic and comparison operations, in reversible gates, and what approaches are used to address these challenges?*

### B. ANCILLA INPUT

An ancilla input (AI) refers to an additional constant input that is introduced into a circuit to convert an irreversible circuit into a reversible one. To convert an irreversible circuit into a reversible one, ancilla inputs are used. The ancilla inputs typically provide additional control signals or carry extra information about the state of the circuit. It's important to note that the introduction of ancilla inputs typically increases the gate count and can impact the overall complexity and cost of the circuit [4]. Reversible circuit design involves finding a balance between achieving reversibility and managing the associated overhead introduced by ancilla inputs.

### C. INPUT

The number of inputs, including ancilla inputs, in a reversible circuit directly impacts the number of qubits required to implement that circuit. Each input wire corresponds to a qubit in a quantum computing context. In reversible computing, a common metric used to evaluate the performance of a design methodology is the number of wires or inputs (n) in the circuit. This metric provides an indication of the circuit's complexity and resource requirements. A higher number of inputs generally implies a larger circuit with more qubits needed to represent and manipulate the input data.

The size of the circuit, in terms of the number of qubits, is an important consideration in quantum computing due to the challenges associated with scaling up quantum systems. The number of qubits affects factors such as computational power, circuit depth, gate count, and the potential for errors and noise.

### D. GARBAGE OUTPUT

In the context of reversible computing, the bijectivity property ensures that each unique input to a circuit corresponds to a unique output, and vice versa. However, in practice, achieving perfect bijectivity in reversible circuits can be challenging due to physical constraints and the nature of reversible logic operations [20]. To maintain the bijectivity property in reversible circuits, it is often necessary to introduce additional output bits that are left unused or discarded after the computation. These unused outputs are referred to as GO. The purpose of GO is to ensure that the mapping between inputs and outputs remains one-to-one, even if the circuit contains irreversible or non-invertible operations. By introducing these unused outputs, the circuit is extended to create a reversible mapping, preserving the bijectivity property.

Garbage outputs carry information that is irrelevant to the desired computation and can be safely ignored or discarded. They can be considered as temporary storage for intermediate results or as placeholders to maintain the consistency of the circuit. After the computation, these GO can be disregarded, as they do not contribute to the desired output of the circuit. The presence of GO may increase the overall circuit size, gate count, and complexity. However, they are a necessary trade-off to ensure reversibility and maintain the bijectivity property in the face of irreversible operations.

Efficient reversible circuit design methodologies often aim to minimize the number of GO while achieving the desired functionality. Minimizing the number of GO helps to reduce resource utilization, gate count, and potential circuit overhead.

### E. GATE COST (GC)

The gate count or gate cost of a circuit refers to the number of logic gates required to construct that circuit. It is a measure of the complexity and resource utilization of the circuit design [21]. The gate count is often used as an indirect measure of the cost of a circuit, as it correlates with factors such as manufacturing costs, power consumption, and performance.

The gate count depends on the complexity of the logic operations performed by the circuit. Different types of logic gates, such as AND gates, OR gates, NAND gates, NOR gates, XOR gates, etc., have varying gate counts. A higher gate count generally implies more complex circuitry, which can result in increased manufacturing costs, higher power consumption, and longer propagation delays.

### F. DELAY CALCULATION

The delay of a reversible gate refers to the time it takes for the gate to produce its output, given its input. The delay of a reversible gate is typically measured in terms of the number of gate delays, which refers to the time it takes for a signal to propagate through a gate. In general, the delay of a reversible gate is dependent on several factors, including the gate's physical implementation, its size and complexity, and the technology used to implement the gate:

$$\text{Total Delay} = f(\mathrm{N}) \tag{2}$$

where N is the number of individual gate delays.

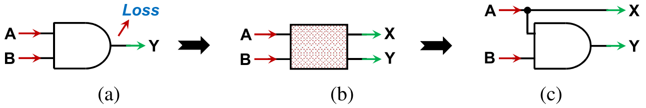


FIGURE 2. Irreversible computation (a) 2-input AND gate (b) 2×2 module (c) 2×2 AND gate [2].

TABLE 1. Truth table for I/O of AND gate and 2×2 AND gate.

| Input | | | Output of AND | | Output of 2×2 AND | | |
|---|---|---|---|---|---|---|---|
| A | B | Permutation | Y | Permutation | X | Y | Permutation |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 2 | 0 | 0 | 1 | 0 | 2 |
| 1 | 1 | 3 | 1 | 1 | 1 | 1 | 3 |

The delay of 2×2, 3×3, and 4×4 reversible gates are often believed to be of unit delay in prior publications [22], [23], regardless of their computing complexity. The logical depth is the unit of measurement for the delay in computations [1]. Each 1×1 and 2×2 reversible gate is considered to have a delay of one unit. Any reversible gate, such as the CNOT gate, Controlled-V, and Controlled-V+ gates, can be created from 1×1 reversible gates and 2×2 reversible gates.

*Question*: *How do the various factors that influence the delay of a reversible gate or circuit interact with each other, and how can designers balance these factors to optimize the performance of their circuits? Furthermore, what are some of the most effective techniques for reducing delay in reversible gates, such as gate-level optimization, circuit-level optimization, and architectural optimization, and how can these techniques be applied in practice to achieve the best possible performance? Finally, what are some of the key trade-offs that designers must consider when optimizing for delay, such as power consumption, gate count, and error rate, and how can these trade-offs be managed to ensure that the overall performance of the circuit is maximized?*

### G. REVERSIBILITY AND INVERTIBILITY

Reversibility refers to the ability of a process or a system to be reversed or undone, resulting in the original state being restored [19]. In other words, if a process or system is reversible, it can be undone and returned to its original state without any net loss or gain of energy, matter, or information. Reversibility is an important concept in many fields, including physics, chemistry, and thermodynamics. In thermodynamics, a reversible process is one that occurs so slowly and smoothly that it can be reversed at any point, with no change in the system or its surroundings. This is in contrast to an irreversible process, which cannot be reversed without some loss or degradation of energy.

Since digital logic circuits are made of irreversible gates, they are physically irreversible. The source's energy is eventually turned into heat with every bit lost. For instance, a two-input AND gate, as shown in Fig. 2 (a) suffers a bit of loss every clock. Additionally, after all logical operations are finished, irreversible logic does not oppositely traverse the state sequences to return to the initial state. The output permutation for the three input permutations, 0, 1, and 2, is 0 in the I/O of the AND gate illustrated in Table 1. The output cannot be unique because there are only two logic states (0 and 1) where at least two outputs will be identical. This is because the reverse calculation is not possible. The statement is false since there are more inputs than possible permutations. Therefore, the number of inputs and outputs should be equal. As seen in Fig. 2 (b), there will be two outputs for two inputs. However, expecting two outputs won't lead to a solution. Think about the 2 × 2 AND gate example circuit in Fig. 2 (c), whose I/O is listed in Table 1. For two input permutations of 0 and 1, the output permutation is 0, more than once. Logic needs to be developed to resolve this scenario and preserve output bits. Thus, such cases are the challenges to the design of reversible gates from irreversible gates.

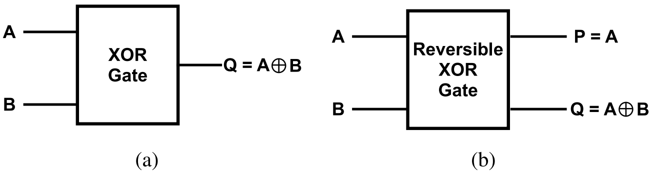


FIGURE 3. Conventional XOR and Reversible XOR gates.

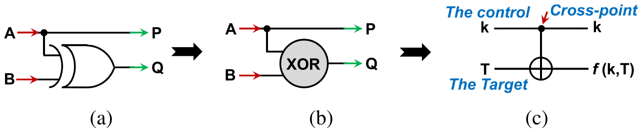


FIGURE 4. 2×2 XOR gate (a) Logic circuit (b) Mathematical representation (c) Reversible/quantum representation [2].

TABLE 2. Truth table for I/O of conventional and reversible XOR gates.

| Input | | | Conventional XOR | | Reversible XOR | | |
|---|---|---|---|---|---|---|---|
| A | B | Permutation | Q | Permutation | P | Q | Permutation |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |
| 1 | 0 | 2 | 1 | 1 | 1 | 1 | 3 |
| 1 | 1 | 3 | 0 | 0 | 1 | 0 | 2 |

To further understand the reversibility operation, let us consider two XOR gates, conventional and reversible XOR gates as shown in Fig. 3. The 2 × 2 XOR gate mathematical equivalent circuit is seen in Fig. 4 (a)-(b). Fig. 4 (c) depicts its related reversible/quantum representation, the reversible Controlled NOT (CNOT) gate. It has a target output T and a control input, k. The input k controls the output function f (k, T). As depicted in Table 2, output 0 is produced by two input vectors, AB = (00, 11), in the case of the conventional irreversible XOR gate. On the other hand, reversible logic gates have the property that a unique input vector can be produced for each output vector, as demonstrated by the truth table in Table 2. This truth table depicts the difference between the reversible and non-reversible gates wherein an extra output (also called GO) is added to the non-reversible gate to make it reversible.

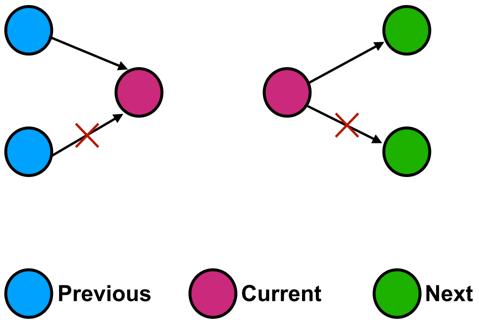


**FIGURE 5.** Forward and Backward Determinism.

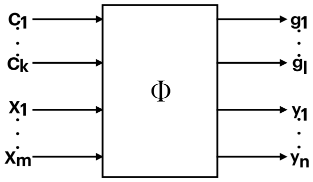


**FIGURE 6.** A combinatorial circuit $\phi$ having function $f$: $(x_1, \ldots, x_m) \rightarrow (y_1, \ldots, y_n)$ embedded in it having constant inputs $c_1, \ldots, c_k$, and GO $g_1, \ldots, g_l$, besides actual inputs $x_1, \ldots, x_m$, and actual outputs $y_1, \ldots, y_n$.

It is common to present reversibility in computer processes as an operational characteristic based on forward and backward determinism. Fig. 5 is used to demonstrate these ideas. Every computation state in a program has a distinct next computation state if it is locally forward deterministic (or simply deterministic). The idea of local backward determinism, which mandates that each computing state has a distinct preceding state, is more esoteric.

In an imperative language, for instance, a program that executes a destructive assignment like x := 5 is not backward deterministic because, in most cases, there is no way to restore the state of x before it was given the value 5. Similar to how conditionals and case expressions are prominent causes of forward nondeterminism in functional programming languages, two or more branches may result in identical outputs for different inputs [19]. A key component of what is often amusingly referred to as the Copenhagen interpretation of reversibility is the locality of forward and backward determinism, which is sometimes emphasized by the phrase "reversibility is a local phenomenon". It specifically demands that every computing step along the route (regardless of whether it is a simple instruction or a complex loop structure) operate in a fashion that is forward and backward-predictable (i.e., inputs uniquely determine outputs and vice-versa). Reversibility is a characteristic of the program rather than the function it computes, to put it another way (i.e., it is an intentional property).

Reversible logic gates are an important aspect of reversible computing, where the goal is to find efficient techniques for building reversible machines using reversible logic elements and to realize reversible logic elements using physically reversible phenomena. The study of reversible logic gates and the development of techniques for building reversible machines from these components is an ongoing area of research with the potential to lead to the development of more efficient and versatile computing systems. Some brief definitions of the reversible gates are discussed below.

*Definition 1:* In the context of Boolean logic, a function $f : B^m \rightarrow B^n$ is known as a logical function or a Boolean function, where B = 0, 1 represents the truth values "false" and "true," respectively, and m, n are positive integers. A logic gate with m inputs and n outputs is referred to as an m-input, n-output logic gate. If a logic gate realizes the function $f$, it is considered to be a reversible logic gate if the function is injective, meaning that the mapping between inputs and outputs is one-to-one. In this case, it is also necessary that m $\leq$ n must hold.

*Definition 2:* Assume S is a small collection of reversible logic gates. A system made up of a limited number of copies of reversible logic gates drawn from S and connected to one another with the following restriction is known as a reversible combinatorial logic circuit over S [19].

1) Fan-out of output is restricted because each output of a gate can only be connected to one input of another gate.
2) Merging of outputs is prohibited when two or more outputs are coupled to the same input port of another gate.
3) A closed path, or feedback loop, is not permitted in the circuit.

It should be noted that in the definition above, fan-out (copying) of output is prohibited in 1. There are many justifications for placing this restriction. The initial explanation is as follows. Fan-out is used in the conventional method (irreversible) for logic circuits at the expense of supplying some energy. The energy consumption issue and the no-cloning theorem in quantum physics both play a role in the inhibition of fan-out. Additionally, the ability to use the inverse circuit of a certain reversible circuit adds another reason to avoid fan-out [24], [25]. Overall, reversible computing and quantum computing rely on preserving the state of signals, making fan-out a challenge in these circuit models. The inverse circuit will have a merging of two or more outputs if the original circuit had fan-out, which makes it difficult to characterize its behavior. Let the circuit of combinatorial logic be reversible. The input ports for the complete circuit are the input ports of gates that are not connected to any other gates. It goes without saying that if we send truth values to the output ports, it will produce values at those output ports. Thus, some injective logical function is implemented over the entire circuit. Here, however, we "embed" a non-injective logical function in using the following method. A set $X = x_1, \ldots, x_m$ of actual inputs and a set $C = c_1, \ldots, c_k$ of constant inputs make up the set of input ports. A set $Y = y_1, \ldots, y_n$ of actual outputs and a set $G = g_1, \ldots, g_l$ of GO make up the set of output ports of Fig. 6. To indicate the truth value at the appropriate ports, we also employ the symbols $x_1, \ldots, x_m, c_1, \ldots, c_k, y_1, \ldots, y_n,$ and $g_1, \ldots, g_l$.

Additionally, the same symbol is used to represent the logical function that the entire circuit performs. In light of this, $(c_1, \ldots, c_k, x_1, \ldots, x_m) = (g_1, \ldots, g_l, y_1, \ldots, y_n)$ [26].

*Definition 3:* Let $s : \{0-1\}^p \rightarrow \{0-1\}^q$ be a logical function and let $\phi$ be an injective logical function with the sets of real inputs $X = \{0-1\}^p$, constant inputs $C = c_1, \ldots, c_k$, real outputs $Y = y_1, \ldots, y_q$, and GO $G = g_1, \ldots, g_l$. If the following is true, we can state that $s$ is embedded in $\phi$.

1) $\exists\ (c_1, \ldots, c_k)\ \varepsilon\ \{0, 1\}^k$,
2) $\forall\ (x_1, \ldots, x_p)\ \varepsilon\ \{0, 1\}^p$, $\forall\ (y_1, \ldots, y_q)\ \varepsilon\ \{0, 1\}^q$,
3) $\exists\ (g_1, \ldots, g_l)\ \varepsilon\ \{0, 1\}^l$
$(f(x_1, \ldots, x_p) = (y_1, \ldots, y_q) \Rightarrow \phi(c_1, \ldots, c_k, x_1, \ldots, x_p) = (g_1, \ldots, g_l, y_1, \ldots, y_q))$

This condition states that the circuit computes the logical function $f$ by allowing the provision of appropriate constant values to $c_1, \ldots, c_k$ and the generation of GO from $g_1, \ldots, g_l$ (even if it is non-injective) [19].

*Definition 4:* Assume that $f$ consists of a few reversible logic gates. Any logical function $p$ that is present in a reversible combinatorial logic circuit over the set $f$ is referred to as being logically universal or functionally complete. Since $f = p$ for some reversible logic gate $p$, the logic gate s is often referred to as being logically universal if the set $f$ is logically universal and $f$ is a singleton.

*Question*: *What is the relationship between reversibility and invertibility in reversible gates, and how does this relationship impact the design and implementation of reversible circuits?*

### H. UNIVERSALITY

To prove the universality of reversible gates, we need to show that they can be used to implement any Boolean function [20]. In other words, we need to demonstrate that any Boolean function can be expressed using a combination of reversible gates. It has been shown that a set of reversible gates is universal if and only if it includes a set of gates that can generate all possible Boolean functions on n bits. The standard set of reversible gates that are used to demonstrate universality includes the Toffoli gate and Hadamard gate, as well as some other gates that can be constructed from them.

The universality of reversible gates refers to the ability of a particular set of reversible logic gates to be used to implement any arbitrary reversible Boolean function. The universality of reversible gates is important because it means that any reversible function can be implemented using a fixed set of gates, which can simplify the design and analysis of reversible circuits. Additionally, having a universal set of reversible gates allows for the efficient implementation of many different reversible functions using the same hardware, which can reduce the overall complexity and cost of a system.

For understanding, we are considering Fredkin gate [27] to understand the universality of the reversible gates. Fredkin gate is a 3×3 reversible gate with outputs as $\mathrm{P} = \mathrm{A}$, $\mathrm{Q} = \overline{\mathrm{A}}\mathrm{B} + \mathrm{AC}$ and $\mathrm{R} = \mathrm{AB} + \overline{\mathrm{A}}\mathrm{C}$. This section provides a detailed explanation of how the Fredkin gate can be used to implement seven core Boolean functions, including NOT, AND, OR, NAND, NOR, XOR, and XNOR. The Fredkin gate is shown to be a universal reversible gate and its versatility is demonstrated through the various input manipulations needed to implement the different Boolean functions. By applying different inputs to the Fredkin gate, it can be used to perform NOT, AND, OR, NAND, NOR, XOR, and XNOR operations. The output of the Fredkin gate, as well as the input manipulations needed to perform each operation, is clearly outlined in Table 3. For understanding, let's take the example of the XOR operation. When a complemented value of A (i.e., $\overline{\mathrm{A}}$) is applied at input A; the complemented value of B (i.e., $\overline{\mathrm{B}}$) is applied at input B and 'B' is applied at input C, XOR output can be ensured at output Q. Similarly, other functionalities can be obtained. Overall, this section highlights the versatility and effectiveness of the Fredkin gate in reversible computing.

**TABLE 3. Setting different logics to the inputs of Fredkin gate to implement seven Boolean functions.**

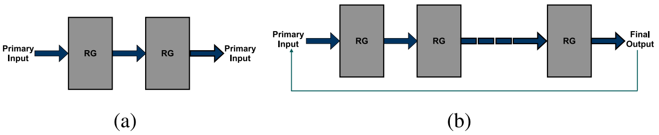
| S. No. | Logic gate | Fredkin Functions | Output |
|---|---|---|---|
| 1 | NOT | Fredkin ($\overline{\mathrm{A}}$, B, C) | $\mathrm{P} = \overline{\mathrm{A}}$ |
| 2 | AND | Fredkin (A, B, 0) | R = A.B |
| 3 | OR | Fredkin (A, B, 1) | Q = A+B |
| 4 | NAND | Fredkin ($\overline{\mathrm{A}}$, $\overline{\mathrm{B}}$, 1) | $\mathrm{Q} = \overline{\mathrm{AB}}$ |
| 5 | NOR | Fredkin ($\overline{\mathrm{A}}$, $\overline{\mathrm{B}}$, 0) | $\mathrm{R} = \overline{\mathrm{A} + \mathrm{B}}$ |
| 6 | XOR | Fredkin ($\overline{\mathrm{A}}$, $\overline{\mathrm{B}}$, B) | $\mathrm{Q} = \mathrm{A} \oplus \mathrm{B}$ |
| 7 | XNOR | Fredkin (A, B, $\overline{\mathrm{B}}$) | $\mathrm{R} = \mathrm{A} \odot \mathrm{B}$ |

*Question*: *What is the concept of universality in reversible gates, and how does it relate to the ability to implement arbitrary quantum circuits? Moreover, what are the fundamental requirements that must be met in order for a set of reversible gates to be considered universal, and how do these requirements impact the design and implementation of reversible circuits?*

### I. MAXIMUM REVERSIBILITY DEPTH (MRD)

Logical reversibility refers to the ability to determine the input operator from the gate's output, while physical reversibility refers to the ability to do the reverse computation. It is noted that not all cognitively reversible systems are physically reversible and vice-versa and that a gate may be logically irreversible but not physically reversible if there is no regeneration of the original input permutations when the corresponding output is applied to the input [4]. The importance of preserving information in reversible computing is emphasized through the concept of logical and physical reversibility, which helps to reduce energy consumption and prevent information loss [28]. When a gate is capable of doing the reverse computation, that is, when the primary input is acquired when the matching output is applied to the gate's input as depicted in Fig. 7 (a), where RG is a reversible gate and is described by the term "Physical Reversibility".

The gate may be conceptually irreversible but not physically reversible if there is no regeneration of the principal input permutations when the appropriate output is applied to



(a) (b)

FIGURE 7. **Reversible Gate (a) Concept of Physical Reversibility, (b) Illustration of Reversibility Depth.**

the input. However, in [29], authors found that the permutations may be produced by employing multiple stages of the same gate. The example is shown in Fig. 7 (b), which compares the primary input and final output after employing various stages of RG. This phenomenon is known as the Maximum Reversibility Depth (MRD) [30].

To put it another way, it is the number of gates that are used to apply the principal input permutations to the matching output. It is also possible to say that an increase in MRD results in a rise in the number of gate stages required to provide main input data, which raises GC.

*Question*: *What is the concept of maximum reversibility depth in reversible gates, and how does it affect the design and implementation of large-scale reversible circuits? Furthermore, how do the fundamental principles of quantum mechanics, such as unitary evolution and the no-cloning theorem, impose constraints on the maximum reversibility depth of a reversible gate or circuit, and what techniques are used to overcome these constraints?*

## IV. IMPLEMENTATION METHODOLOGY OF REVERSIBLE GATES

The reversible gates are designed such that each input combination maps to a unique output combination, ensuring reversibility. These implementations are relatively simple and straightforward, making them suitable for small-scale reversible circuits. Quantum circuit implementations utilize quantum gates to realize reversible operations. Quantum gates, such as the Controlled-NOT (CNOT) gate and Toffoli gate, inherently possess reversibility. These gates can be combined and controlled to construct larger reversible circuits. Quantum circuit implementations are particularly relevant in the context of quantum computing, where quantum gates are the building blocks for quantum algorithms. Another approach includes the reversible logic synthesis implementation which is a systematic approach to design reversible circuits. It involves the transformation of irreversible Boolean functions or logic circuits into their reversible equivalents. Various algorithms and techniques are employed in reversible logic synthesis, such as using additional ancillary bits, exploiting symmetry properties, and applying gate-level optimizations. Reversible logic synthesis aims to minimize the number of gates, quantum cost, or other metrics while ensuring reversibility.

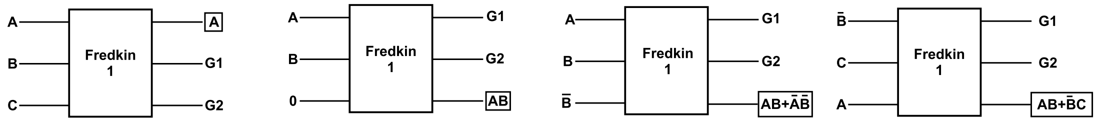
These basic implementation methods can be combined and adapted to suit specific design requirements. Furthermore, there are advanced techniques and specialized reversible gate libraries available for more complex reversible circuit designs [19]. The choice of implementation method depends on factors such as the size and complexity of the circuit, available resources, and the target application or computing paradigm.

### A. DEMONSTRATION OF 13 STANDARD BOOLEAN FUNCTIONS USING FREDKIN GATE

To demonstrate the universality of the reversible gates, we need to show that any Boolean function can be expressed as a combination of Toffoli gates, Hadamard gates, and other gates that can be constructed from them. This can be done using Boolean algebra, which involves manipulating Boolean expressions to obtain the desired result. The reason we need 13 Boolean expressions to prove the universality of reversible gates is that these expressions are used to represent all possible Boolean functions on three bits, which is the minimum number of bits required to demonstrate universality [20]. By showing that we can express any Boolean function on three bits using a combination of Toffoli gates, Hadamard gates, and other gates, we can demonstrate that this set of gates is universal. The 13 Boolean expressions are given below:

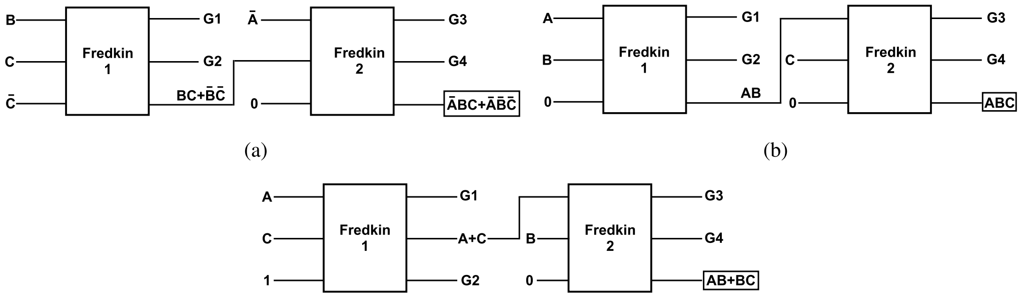
1) $A$
2) $AB$
3) $AB + BC + CA$
4) $\overline{A}BC + \overline{A}\,\overline{B}\,\overline{C}$
5) $ABC + \overline{A}\,\overline{B}C + \overline{A}B\overline{C} + A\overline{B}\,\overline{C}$
6) $ABC + A\overline{B}\,\overline{C}$
7) $ABC$
8) $AB + BC$
9) $AB + \overline{A}\,\overline{B}C$
10) $AB\overline{C} + \overline{A}BC + \overline{A}\,\overline{B}\,\overline{C}$
11) $AB + BC + \overline{A}\,\overline{B}\,\overline{C}$
12) $AB + \overline{B}C$
13) $AB + \overline{A}\,\overline{B}$

To fulfill the condition of the multi-operation nature of any reversible gate, it must satisfy the thirteen standard Boolean expressions and to demonstrate that, we have taken Fredkin gate to demonstrate the implementation of those equations. We have designed the block diagrams for all 13 equations for the Fredkin gate. The same procedure can be followed for other reversible gates to obtain these 13 Boolean equations. The block schematics for implementing each of the 13 functions utilizing a Fredkin gate are shown in Fig. 8 to Fig. 12. The figures demonstrate how the 13 equations need to prove the reversibility of the Fredkin gate and are derived by giving different inputs to the gate. For instance, in order to derive the first equation F=A, we give the Fredkin inputs (A, B, 0), meaning that the first input is given as "A", the second input is fixed at "B" and the third input is given as "0". As a result, we got A at output P, while the other two outputs are GO. This method is used to assign the different inputs to the gates and thus obtain 13 different Boolean equations. The same procedure can be followed by other reversible gates to prove their reversibility and universality.



(a) (b) (c) (d)

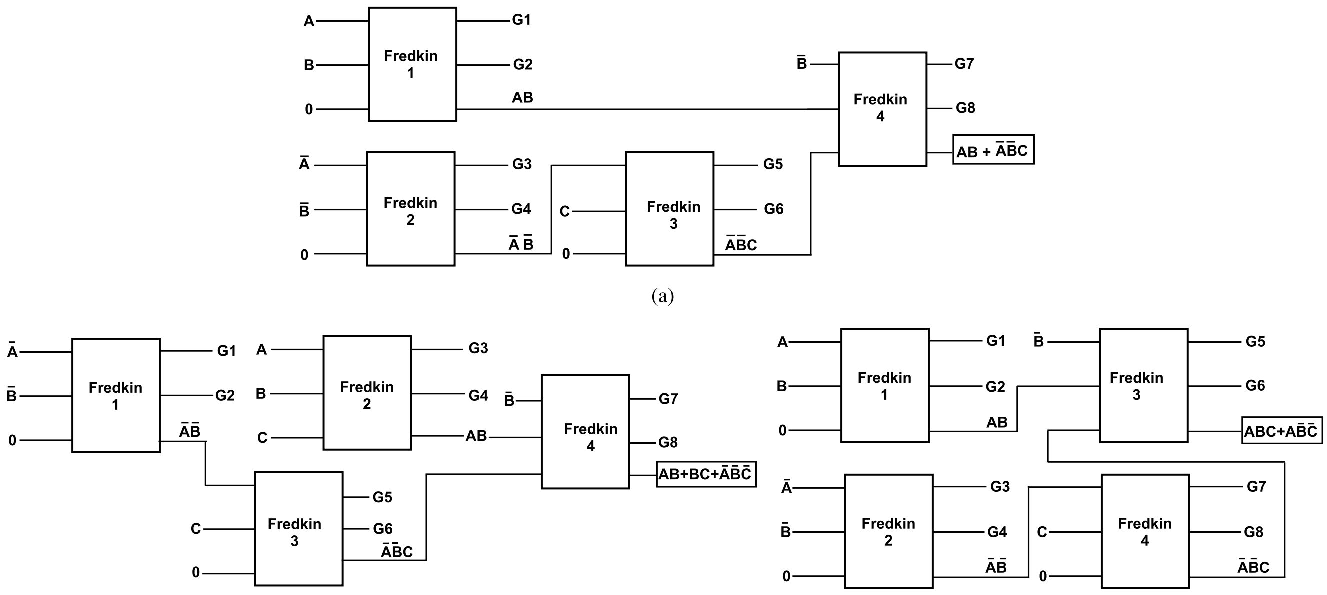
**FIGURE 8.** Implementation of expressions that use only one Fredkin gate (a) A, (b) AB, (c) $AB + \bar{A}\,\bar{B}$ and (d) $AB + \bar{B}C$.



(a) (b)

(c)

**FIGURE 9.** Implementation of expressions that use two Fredkin gates (a) $\bar{A}BC + \bar{A}\,\bar{B}\,\bar{C}$, (b) ABC, and (c) AB + BC.



(a)

(b) (c)

**FIGURE 10.** Implementation of expressions that use four Fredkin gates (a) $AB + \bar{A}\,\bar{B}C$, (b) $AB + BC + \bar{A}\,\bar{B}\,\bar{C}$, and (c) $ABC + A\bar{B}\,\bar{C}$.

**TABLE 4.** 2×2 reversible gates with A, B as inputs and P, Q as outputs.

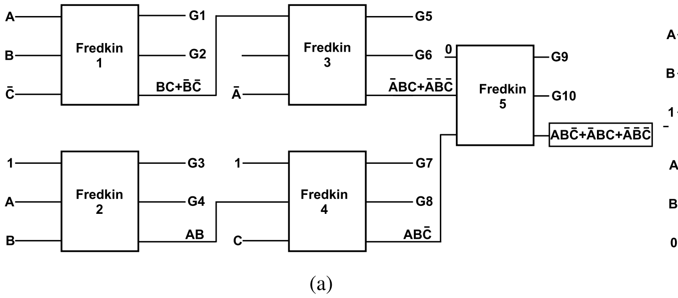
| Gate | P | Q |
|---|---|---|
| Feynman/CNOT [31] | A | A⊕B |
| Swap [32] | B | A |

## V. ANALYSIS OF VARIOUS REVERSIBLE GATES

There are a number of reversible logic gates that exist in present literature as given in [100], [103], [104], [36], [105]. In this study, we attempt to present all the reversible gates that are not covered by previous review papers which may be helpful to researchers. We have categorized all the reversible gates present so far on the basis of the number of input-output vectors. So, depending on that, the reversible gates are divided into four categories. 2×2, 3×3, 4×4 and 5×5 reversible gates as depicted in Tables 4, 5, 6 and 7 respectively.

### A. QUANTUM COST OF REVERSIBLE GATES

The quantum cost of a reversible gate is a measure of the resources required to implement the gate in a quantum circuit. It is defined as the number of elementary quantum gates (i.e., single-qubit gates and CNOT gates) needed to implement the reversible gate. The quantum cost of a reversible



(a) (b)

**FIGURE 11.** Implementation of expressions that use five Fredkin gates (a) $AB\overline{C} + \overline{A}BC + \overline{A}\,\overline{B}\,\overline{C}$, (b) $AB + BC + CA$.

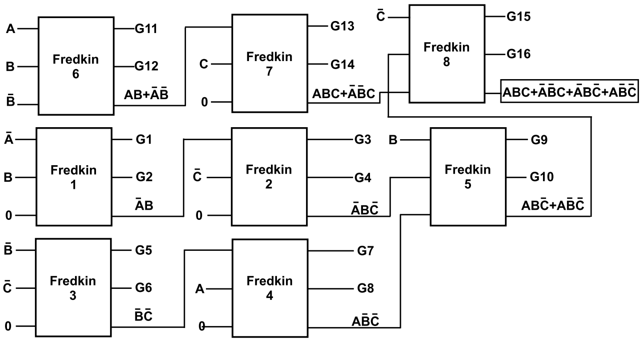


**FIGURE 12.** Implementation of expression, $ABC + \overline{A}\,\overline{B}C + \overline{A}B\overline{C} + A\overline{B}\,\overline{C}$ that use eight Fredkin gates.

gate can depend on the particular set of elementary gates that are used to implement it. For example, some sets of elementary gates may be more efficient for certain gates than others [38].

The quantum cost can also depend on the size of the gate. Generally, the quantum cost of a gate increases as the number of qubits it acts on increases. This is because more resources are required to implement a gate that acts on multiple qubits than a gate that acts on only one or two qubits. There is no single formula or algorithm for calculating the quantum cost of a reversible gate, but it can be estimated through analysis of the gate's structure. The actual implementation of a reversible gate in a particular quantum architecture may also affect the quantum cost, as different architectures have different capabilities and limitations. The Quantum cost of some of the reversible gates is tabulated in Table 8.

*Question*: *What are the potential applications and implications of minimizing quantum cost in various fields, such as quantum cryptography, quantum error correction, and quantum algorithms?*

### B. COMPLEXITY OF SOME REVERSIBLE GATES

The total number of reversible gates that are needed to obtain the 13 standard Boolean equations determines its complexity [20]. The lesser the number of gates used less is the complexity. The complexity of some of the reversible gates is given in Table 9. For instance, the sum of gates used in the case of Fredkin gate is 42 by adding the total number of gates that are used to implement all 13 Boolean equations. The same procedure can be followed by other reversible gates to calculate the total number of gates used to implement the 13 Boolean expressions as given in Table 9.

*Question: How the complexity is related to the output function of reversible gates?*

### C. QUANTUM LOGIC DIAGRAMS OF SOME REVERSIBLE GATES

Quantum diagrams, also known as quantum circuits, are a visual representation of the operations performed in quantum computation. These circuits are made up of gates, which are the basic building blocks of quantum algorithms. In a quantum circuit diagram, the qubits are represented as horizontal lines, and the gates are represented as boxes that act on one or more qubits. The input state of the qubits is represented on the left-hand side, and the output state is represented on the right-hand side. Quantum circuits are diagrams that represent quantum operations as boxes with inputs and outputs. Reversible gates can be represented as quantum circuits that have the property that they can be run backward to recover the input state.

Some common reversible gates that are used in quantum circuits include the NOT gate, the Controlled-NOT (CNOT) gate, the Toffoli gate, and the Fredkin gate. Quantum equivalent realization of some of the common reversible gates [106] is shown in Fig. 13 where A, B, C, and D (P, Q, R, and S) are inputs (outputs) for every gate.

The NOT gate is a single-qubit gate that flips the state of a qubit and is represented in a quantum circuit diagram as a box with a single input and output line. The CNOT gate is a two-qubit gate that flips the second qubit if the first qubit is in the state $|1\rangle$. It is represented in a quantum circuit diagram as a box with two input lines (the first qubit is the control qubit and the second qubit is the target qubit) and two output lines.

The Toffoli gate is a three-qubit gate that flips the third qubit if the first two qubits are in the state $|1\rangle$. It is represented in a quantum circuit diagram as a box with three input lines (the first two qubits are the control qubits and the third qubit is the target qubit) and three output lines. The Fredkin gate is a three-qubit gate that swaps the second and third qubits if the first qubit is in the state $|1\rangle$. It is represented in a quantum circuit diagram as a box with three input lines

**TABLE 5.** 3×3 reversible gates with A, B, and C as inputs and P, Q, and R as outputs.

| Gate | P | Q | R |
|---|---|---|---|
| Fredkin [27] | $A$ | $\overline{A}B+AC$ | $AB+\overline{A}C$ |
| SFV [20] | $A\oplus B$ | $B\oplus C$ | $\overline{A}B+BC+\overline{A}C$ |
| TS-3 [33] | $A$ | $B$ | $A\oplus B\oplus C$ |
| RM [34] | $A\oplus BC$ | $\overline{A}B+AC$ | $\overline{A}C+AB$ |
| RG [35] | $A\oplus B$ | $AB+BC+AC$ | $\overline{B}C+A(\overline{B\oplus C})$ |
| RG1 [21] | $\overline{B}$ | $A\overline{B}+BC$ | $A\oplus C$ |
| G1 [36] | $\overline{B}$ | $\overline{A}\ \overline{B}+AB$ | $\overline{A}\ \overline{B}C+A\overline{C}+B\overline{C}$ |
| RG2 [21] | $\overline{A}\ \overline{B}\oplus C$ | $\overline{A}\oplus\overline{B}$ | $A$ |
| URLG [37] | $AB+BC+AC$ | $\overline{A}B+B\overline{C}+\overline{A}\ \overline{C}$ | $A\overline{B}+\overline{B}\ \overline{C}+\overline{A}\ \overline{C}$ |
| MPG [38] | $A$ | $A\oplus B$ | $A\overline{B}\oplus C$ |
| SAM [38] | $\overline{A}$ | $\overline{A}B\oplus A\overline{C}$ | $\overline{A}C\oplus AB$ |
| R [34] | $A\oplus B$ | $A$ | $\overline{C}\oplus(AB)$ |
| QCA1 [39] | $AB+BC+AC$ | $AB+B\overline{C}+A\overline{C}$ | $\overline{A}B+BC+\overline{A}C$ |
| QCA2 [39] | $AB+BC+AC$ | $AB+B\overline{C}+A\overline{C}$ | $\overline{A}B+B\overline{C}+\overline{A}\ \overline{C}$ |
| RQG [40] | $AB+BC+AC$ | $\overline{A}B+BC+\overline{A}C$ | $A\oplus C$ |
| Peres [41] | $A$ | $A\oplus B$ | $AB\oplus C$ |
| URG [42] | $(A+B)\oplus C$ | $B$ | $AB\oplus C$ |
| SSG [43] | $B\oplus C$ | $A\oplus B$ | $AB+BC+AC$ |
| MNFT [44] | $(A\oplus B)C\oplus A$ | $A\oplus B$ | $(A\oplus B)C\oplus A\oplus C$ |
| SRK [45] | $\overline{A}$ | $A\oplus B\oplus C$ | $\overline{A}C\oplus AB$ |
| SSG-1 [46] | $\overline{A\oplus B}$ | $\overline{B\oplus C}$ | $AB+BC+AC$ |
| NNG [47] | $C$ | $AB+B\overline{C}+\overline{A}\ \overline{B}\ \overline{C}$ | $AB+A\overline{C}+\overline{A}\ \overline{B}C$ |
| IMG [48] | $B$ | $\overline{A}C+A(BC+\overline{B}C)$ | $\overline{A}$ |
| TR [49] | $A$ | $A\oplus B$ | $A\overline{B}\oplus C$ |
| RUG [50] | $AB+BC+AC$ | $AB+\overline{A}\ \overline{C}$ | $B\oplus C$ |
| NG [51] | $A$ | $AB\oplus C$ | $\overline{A}\ \overline{C}\oplus\overline{B}$ |
| DG [51] | $A$ | $\overline{A\oplus B}$ | $\overline{A}B\oplus C$ |
| RMG [30] | $A\oplus BC$ | $\overline{A}B+AC$ | $\overline{A}C+AB$ |
| Toffoli [12] | $A$ | $B$ | $AB\oplus C$ |
| MCL [52] | $\overline{B}\ \overline{C}$ | $\overline{A}\ \overline{B}$ | $A$ |
| GI [36] | $\overline{B}$ | $\overline{A}\ \overline{B}+AB$ | $\overline{A}\ \overline{B}C+A\overline{C}+B\overline{C}$ |
| BJN [53] | $A$ | $B$ | $(A+B)\oplus C$ |
| HAS [54] | $A$ | $A\oplus B\oplus C$ | $AB+\overline{A}C$ |
| New Gate [55] | $A$ | $AB\oplus C$ | $\overline{A}\ \overline{C}\oplus\overline{B}$ |
| ORG-I [56] | $AB+(A\oplus B)C$ | $A\oplus B$ | $A\overline{B}+(A\oplus\overline{B})$ |
| ORG-II [56] | $A\overline{B}+BC$ | $\overline{A}B+\overline{B}C$ | $AB+\overline{B}C$ |
| MF [57] | $A$ | $\overline{A}B+A\overline{C}$ | $AB+\overline{A}C$ |
| BG-1 [58] | $A\oplus B$ | $B\oplus C$ | $C$ |
| GB-1 [58] | $A\oplus B\oplus C$ | $B\oplus C$ | $C$ |
| NG-R2 [59] | $A$ | $A\oplus B$ | $(A+B)\oplus C$ |
| DG [60] | $A$ | $\overline{A\oplus B}$ | $A\overline{B}\oplus C$ |
| RSG [61] | $A\oplus B$ | $AB\oplus C$ | $\overline{A}B\oplus C$ |
| TKS [62] | $A\overline{C}+BC$ | $A\oplus B\oplus C$ | $AC+B\overline{C}$ |
| NCT [63] | $A$ | $B$ | $\overline{AB}\oplus C$ |
| NRLG [64] | $A$ | $A\odot B$ | $A\overline{B}\oplus C$ |
| $S_1G$ [65] | $\overline{A}B\oplus C$ | $A\odot B$ | $\overline{AB}\odot C$ |
| $S_2G$ [65] | $AB\oplus C$ | $A\oplus B$ | $AB\oplus C$ |
| FRSG-1[66] | $A$ | $A\odot B$ | $AB\oplus C$ |
| JTF1[66] | $A$ | $A\oplus B$ | $A\oplus B\oplus C$ |
| MG[67] | $A$ | $A\oplus B\oplus C$ | $\overline{A}C\oplus AB$ |
| PRG[68] | $A\oplus B\oplus C$ | $AC+\overline{B}\ \overline{C}$ | $AC+B\overline{C}$ |
| NFT[69] | $A$ | $\overline{B}C\oplus A\overline{C}$ | $BC\oplus A\overline{C}$ |

(the first qubit is the control qubit and the second and third qubits are the target qubits) and three output lines.

In quantum circuit diagrams, the order in which the gates are applied is important, as quantum gates do not necessarily commute with each other. This means that the output of one gate depends on the input to the next gate, and the order in which the gates are applied can affect the final output state.

### D. PARITY PRESERVING REVERSIBLE GATES

Parity-preserving reversible gates are important in quantum computing because they preserve the parity of the qubits in the quantum state being operated on. The parity of a qubit state is the number of qubits in the state that are in the state $|1\rangle$, modulo 2. For example, the parity of the state $|001\rangle$ is 1, the parity of the state $|011\rangle$ is 0, and the parity of the state $|101\rangle$ is 1.

Parity-preserving reversible gates are important in quantum error correction, where the ability to detect and correct errors in a quantum state is crucial. Quantum error correction schemes typically rely on the parity of the qubits in the state to detect errors, and so any operation that changes the parity of the qubits can introduce errors that may not be detectable. Parity-preserving reversible gates are also important in quantum algorithms that rely on maintaining the parity of the qubits in the state being operated on.

In addition, parity-preserving reversible gates have some theoretical advantages, such as being able to implement arbitrary unitary operations on a set of qubits with a smaller number of gates that would be required without parity preservation. Overall, parity-preserving reversible gates are an important tool in quantum computing, with applications in error correction, quantum algorithms, and circuit optimization.

Parity-preserving reversible gates belong to a type of reversible logic gates with the extra characteristic that their input and output parity are identical. If the Exclusive-OR operation of the inputs and the outputs are the same, the parity of the input and the output will be preserved in a reversible logic gate. This is verified by using the Feynman gate as depicted in Table 10.

An N×N reversible gate will be parity-preserving if the inputs $I_1, I_2, \ldots, I_N$ and outputs of the gate $O_1, O_2, \ldots, O_N$ satisfy the following condition:

$$I_1 \oplus I_2, \ldots, \oplus I_N = O_1 \oplus O_2, \ldots, \oplus O_N. \tag{3}$$

Different parity-preserving reversible gates that are present in the literature so far are tabulated in Table 11.

*Question*: *What are the fundamental principles and mathematical underpinnings behind designing parity-preserving reversible gates, and how can such gates be efficiently implemented in hardware to minimize gate count and computational complexity while maintaining a high degree of gate fidelity and error-correction capability?*

## VI. FUTURE SCOPE

Researchers can explore innovative circuit architectures and logic gate configurations that offer improved performance, reduced quantum cost, lower complexity, and other desirable characteristics. The review paper has discussed performance evaluation measures for reversible gates. Future work can delve deeper into optimization techniques for reversible logic circuits, aiming to minimize quantum cost, gate count, or circuit delay. Researchers can explore algorithms, heuristics, or machine-learning approaches to optimize gate-level or circuit-level designs. They can investigate physical implementations of reversible gates using emerging technologies such as quantum computing, adiabatic circuits, or

**TABLE 6.** 4×4 reversible gates having A, B, C, and D as inputs and P, Q, R, and S as outputs.

| Gate | P | Q | R | S |
|---|---|---|---|---|
| DFSCL [70] | A | $A\oplus B$ | $A\oplus C$ | $A(B+C)\oplus D$ |
| BME [71] | A | $AB\oplus C$ | $AD\oplus C$ | $\overline{A}B\oplus C\oplus D$ |
| Pareek [72] | A | $\overline{A}B\oplus AD$ | $\overline{A}B\oplus AD\oplus C$ | $B\oplus D$ |
| SCL [73] | A | B | C | $A(B+C)\oplus D$ |
| PPHCG [74] | $B\oplus C\oplus D$ | $A\oplus B\oplus C$ | $A\oplus B\oplus D$ | $A\oplus C\oplus D$ |
| TSG [62] | A | $\overline{A}\,\overline{C}\oplus\overline{B}$ | $(\overline{A}\,\overline{C}\oplus\overline{B})\oplus D$ | $(\overline{A}\,\overline{C}\oplus\overline{B})D\oplus(AB\oplus D)$ |
| PFAG [75] | A | $A\oplus B$ | $A\oplus B\oplus C$ | $(A\oplus B)C\oplus AB\oplus D$ |
| HNG [76] | A | B | $A\oplus B\oplus C$ | $(A\oplus B)C\oplus AB\oplus D$ |
| PAOG [77] | A | $A\oplus B$ | $AB\oplus D$ | $((A\oplus B)\oplus D)\oplus(AB\oplus C)$ |
| MRG [77] | A | $A\oplus B$ | $(A\oplus B)\oplus C$ | $(AB\oplus D)\oplus((A\oplus B)\oplus C)$ |
| SMS [78] | $A\oplus C\oplus D$ | $D\oplus BC$ | C | $D\oplus B\oplus C\oplus BC$ |
| RI [79] | $A\oplus C$ | $B\oplus C\oplus AB\oplus BC$ | $A\oplus B\oplus C$ | $D\oplus C\oplus AB\oplus BC$ |
| OTG [80] | A | $A\oplus B$ | $A\oplus B\oplus D$ | $(A\oplus B)D\oplus(AB\oplus C)$ |
| MCT [63] | A | B | C | $ABC\oplus D$ |
| DKG [81] | B | $\overline{A}C+A\overline{D}$ | $(A\oplus B)(C\oplus D)\oplus CD$ | $B\oplus C\oplus D$ |
| SG [82] | A | $\overline{A}B\oplus AC$ | $\overline{A}B\oplus AC\oplus D$ | $AB\oplus\overline{A}C\oplus D$ |
| MRLG [83] | A | $AB\oplus\overline{A}C$ | $B\oplus AC$ | $B\oplus AC\oplus D$ |
| RAM [84] | A | $A\oplus B$ | $A\oplus B\oplus C$ | $A\oplus B\oplus C\oplus D$ |
| HNFG [85] | A | $A\oplus C$ | B | $B\oplus D$ |
| FAS [54] | $A\oplus B\oplus C$ | $(B\oplus C\oplus D)A\oplus(C\oplus D)B$ | C | $B\oplus C\oplus D$ |
| BVF [86] | A | $A\oplus B$ | C | $C\oplus D$ |
| MKG [87] | A | C | $(\overline{A}\,\overline{D}\oplus\overline{B})\oplus C$ | $(\overline{A}\,\overline{D}\oplus\overline{B})C\oplus(AB\oplus D)$ |
| DPG [86] | A | $A\oplus B$ | $A\oplus B\oplus D$ | $(A\oplus B)D\oplus AB\oplus C$ |
| Sayem [88] | A | $\overline{A}B\oplus AC$ | $\overline{A}B\oplus AC\oplus D$ | $AB\oplus\overline{A}C\oplus D$ |
| RR [89] | A | $A\oplus B$ | $AB\oplus C$ | $A\overline{B}\oplus D$ |
| RDFF [90] | A | $\overline{A}C\oplus AB$ | $\overline{A}C\oplus AB\oplus D$ | $AC\oplus\overline{A}B$ |
| MTSG [69] | A | $A\oplus B$ | $A\oplus B\oplus C$ | $(A\oplus B)C\oplus AB\oplus D$ |
| MHNG [91] | A | D | $A\oplus B\oplus C$ | $(A\oplus B)C\oplus AB\oplus D$ |
| R2D [76] | A | $A\oplus B$ | $AB\oplus D$ | AB |
| RSJ [92] | A | D | $AB\oplus C$ | $AB\oplus D$ |
| SVS [93] | A | $B\overline{C}\oplus\overline{A}B\oplus A\overline{B}C$ | $B\overline{C}\oplus\overline{A}B\oplus A\overline{B}C\oplus D$ | $\overline{A}C\oplus AB$ |
| Inventive 0 [93] | C | $A\oplus B\oplus C$ | $((A\oplus B)C+AB)\oplus D$ | $((A\oplus B)C+\overline{A}B)\oplus\overline{D}$ |
| BE [59] | $A\oplus B\oplus D$ | $B\oplus D$ | $(A+B)\oplus D$ | D |
| BG-2 [59] | A | $A\oplus B$ | $B\oplus C$ | $C\oplus D$ |
| RMF [59] | A | $A\oplus B$ | $A\oplus B\oplus C$ | $AB\oplus D$ |
| GB-2 [59] | A | $A\oplus B$ | $A\oplus B\oplus C$ | $A\oplus B\oplus C\oplus D$ |
| NS [94] | $\overline{A}$ | $B\oplus C\oplus D$ | $[(A\oplus B)\oplus D]C+B(A\oplus D)$ | $(\overline{A}\,\overline{B}\,\overline{D}+ABD)+$<br>$(AB\overline{C}+\overline{A}\,\overline{B}C)+(A\overline{C}D+\overline{A}C\overline{D})$ |

**TABLE 7.** 5×5 reversible gates having A, B, C, D, and E as inputs and P, Q, R, S, and T as outputs.

| Gate | P | Q | R | S | T |
|---|---|---|---|---|---|
| FTRG [95] | $A\oplus B\oplus C$ | $BC+(B+C)(A\oplus D)$ | $AB+AD$ | $\overline{B}C+(B+\overline{C})(A\oplus B)$ | $\overline{A}\,\overline{E}+A(D\oplus E\oplus B)$ |
| LCG [96] | A | $A\oplus B$ | $A\oplus B\oplus C$ | $D\oplus(A\oplus B)C\oplus AB$ | $B\oplus E\oplus D\oplus(A\oplus B)C\oplus AB$ |
| NG-R1 [59] | A | B | C | $AB\oplus D$ | $AC\oplus E$ |
| PPPG [97] | A | $\overline{A}\,\overline{C}\oplus\overline{B}$ | $(\overline{A}\,\overline{C}\oplus\overline{B})\oplus D$ | $(\overline{A}\,\overline{C}\oplus\overline{B})D\oplus(AB\oplus C)$ | $BE(A+D)+\overline{A}D(C\oplus E)+\overline{B}D(A+E)$ |
| BVPPG [98] | A | B | $AB\oplus C$ | D | $AD\oplus E$ |
| BBCDC [99] | A | $B\overline{D}\,\overline{E}\oplus\overline{B}(CD\oplus E)$ | $\overline{B}E\oplus C(B\oplus\overline{D})$ | $BE\oplus\overline{B}\,\overline{C}D$ | $E\oplus D(B\oplus C)$ |
| MPS [100] | $A\overline{B}+\overline{A}BC+B\overline{C}(A\oplus D)$ | $A(C+B)+\overline{C}(A\overline{B}D+\overline{A}B\overline{D})$ | $\overline{A}C(\overline{B}+BE)+A\overline{C}\,\overline{D}+AC(B+D)$ | $BC\overline{(A\oplus D)}+\overline{B}(A\oplus D)+AB\overline{C}$ | E |
| HASFT [101] | A | $D\oplus B(A\oplus C)$ | $B\oplus C$ | $(D\oplus B)\overline{(A\oplus C)}$ | $B\oplus D\oplus E$ |
| FASFT [101] | A | B | $A\oplus B\oplus C$ | $A(B\oplus C)\oplus BC\oplus D$ | $A\overline{(B\oplus C)}\oplus B\overline{C}\oplus E$ |
| ZPLG [96] | $A\oplus B$ | $A\oplus B\oplus D$ | $A\oplus B\oplus C\oplus D$ | $(A\oplus D)(B\oplus C)\oplus BC\oplus D$ | $(A\oplus D)(B\oplus C)\oplus B\overline{C}C\oplus D\oplus E$ |
| OD [54] | $B+C\oplus D\oplus A$ | B | C | D | $(A\oplus D)(B+C)D\oplus E\oplus AD$ |
| NG-PP [102] | A | B | C | $A\oplus B\oplus C\oplus D$ | $A\oplus B\oplus C\oplus E$ |
| HG-PP [102] | $A\oplus C\oplus E$ | $B\oplus C$ | C | $D\oplus E$ | E |

nanoscale/molecular electronics. Realizing reversible gates in hardware and analyzing their performance in practical scenarios would be valuable. Researchers can investigate the development of domain-specific reversible gates optimized for applications in areas such as cryptography, quantum computing, low-power design, or bioinformatics. The future of reversible gates looks promising, especially in the context of emerging technologies such as quantum computing and low-power computing. Reversible gates have the potential to reduce energy dissipation and power consumption, which is particularly important in mobile and IoT devices. As reversible gates can also preserve input and output information, they can help to reduce errors and improve the reliability of electronic systems. In summary, reversible gates have a promising future in both classical and quantum computing, and their continued development and optimization will likely have a significant impact on the next generation of electronic devices and systems.

In the realm of quantum computing, reversible gates are essential for the implementation of quantum algorithms, where information is encoded in quantum states that must be carefully preserved throughout the computation. Reversible gates also offer the possibility of building large-scale quantum systems with lower error rates, making them more feasible for practical applications.

**TABLE 8.** Quantum cost of reversible gates.

| S. No | Quantum Cost | Reversible Gates |
|---|---|---|
| 1 | 1 | Feynman |
| 2 | 2 | Double-Feynman, TS-3, HNFG, BVF |
| 3 | 3 | MCF |
| 4 | 4 | Peres, HG-PP, Fredkin, TR, TSG |
| 5 | 5 | BME, MNFT, SAM, SRK, SAM, NFT, BJN, BME HAS, RMF, Toffoli, NG-PP, NCT, R, MCT |
| 6 | 6 | HNG, MTSG, RR, PPHCG, RCQCA, URG |
| 7 | 7 | IG, MIG, PAREEK |
| 8 | 8 | FAS, PFAG |
| 9 | 9 | OTG |
| 10 | 10 | OD, RSG, BVPPG, HASFT, FASFT |
| 11 | 11 | New Gate |
| 12 | 15 | MKG |
| 13 | 19 | PPRG |
| 14 | 20 | RMG, RUG |
| 15 | 41 | CQCA |
| 16 | 42 | MX-QCA |

Some of the major open problems include the following:

- Given a reversible function, the problem of synthesizing a reversible circuit that implements the

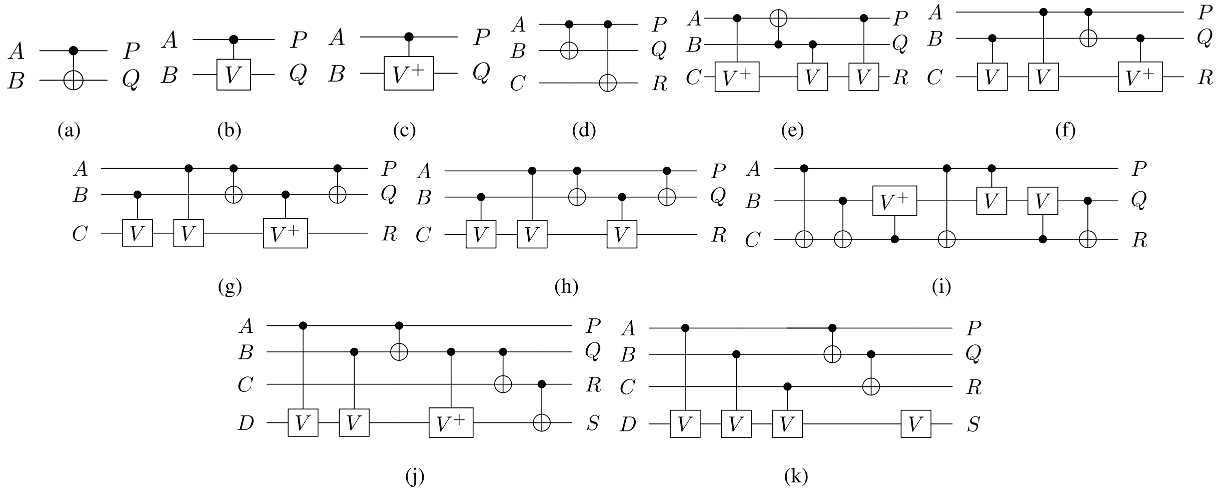


**FIGURE 13.** Quantum Equivalent Realization of some of the reversible gates where A, B, C, and D (P, Q, R, and S) are inputs (outputs) for every gate. (a) Feynman Gate (b) Controlled-V (c) Controlled-$V^+$ (d) Double-Feynman Gate (e) TR Gate (f) Peres Gate (g) Tofolli Gate (h) BJN Gate (i) Fredkin Gate (j) MRG Gate (k) HNG Gate.

**TABLE 9.** Complexity of reversible gates.

| Boolean Equations | CQCA | Toffoli | Fredkin | RG-QCA | RUG | SFV-QCA | SSG-QCA | t-QCA | Peres | QCA1 | RM | RC-QCA | PRG |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 2 | 1 | 1 | 1 | 1 | 1 |
| AB | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| ABC | 2 | 2 | 2 | 2 | 3 | 2 | 1 | 1 | 2 | 2 | 2 | 2 | 2 |
| AB+BC | 2 | 2 | 2 | 2 | 3 | 2 | 1 | 1 | 2 | 2 | 2 | 2 | 2 |
| $AB+\overline{B}C$ | 3 | 3 | 1 | 3 | 3 | 3 | 1 | 1 | 3 | 3 | 1 | 1 | 1 |
| $AB+\overline{A}\,\overline{B}$ | 4 | 2 | 2 | 1 | 2 | 1 | 2 | 8 | 1 | 2 | 2 | 2 | 2 |
| $AB+\overline{A}\,\overline{B}\,C$ | 5 | 5 | 5 | 5 | 3 | 4 | 3 | 4 | 3 | 3 | 2 | 3 | 2 |
| $ABC+A\overline{B}\,\overline{C}$ | 6 | 5 | 4 | 5 | 3 | 2 | 2 | 2 | 3 | 3 | 3 | 2 | 2 |
| AB+BC+AC | 1 | 5 | 5 | 1 | 1 | 1 | 5 | 6 | 4 | 1 | 5 | 3 | 2 |
| $\overline{A}BC+\overline{A}\,\overline{B}\,\overline{C}$ | 3 | 3 | 3 | 2 | 2 | 2 | 2 | 2 | 3 | 2 | 2 | 3 | 3 |
| $AB+BC+\overline{A}\,\overline{B}\,\overline{C}$ | 6 | 5 | 6 | 5 | 4 | 4 | 2 | 2 | 1 | 4 | 2 | 3 | 2 |
| $AB\overline{C}+\overline{A}BC+\overline{A}\,\overline{B}\,\overline{C}$ | 6 | 6 | 6 | 5 | 3 | 5 | 4 | 2 | 4 | 3 | 3 | 3 | 3 |
| $ABC+\overline{A}\,\overline{B}C+\overline{A}B\overline{C}+A\overline{B}\,\overline{C}$ | 3 | 2 | 3 | 1 | 2 | 2 | 4 | 6 | 3 | 2 | 2 | 2 | 1 |
| Total | 43 | 42 | 42 | 34 | 31 | 30 | 30 | 38 | 32 | 29 | 28 | 28 | 24 |

**TABLE 10.** Parity preserving nature of Feynman gate.

| A | B | P | Q | $A \oplus B$ | $P \oplus Q$ |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 1 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 | 1 |
|  |  |  |  | XOR (A, B) = 0 | XOR (P, Q) = 0 |

function with the minimum number of gates is an open problem. Currently, the best-known algorithms for reversible circuit synthesis rely on heuristic approaches and do not guarantee optimal solutions in all cases.

- Quantum circuits can be constructed using reversible gates, and the problem of synthesizing optimal quantum circuits is also an open problem. Specifically, given a quantum circuit that implements a unitary operation, the problem is to find the shortest quantum circuit that implements the same operation using a given set of reversible gates.
- The automation of reversible logic design is another open problem. Specifically, the challenge is to develop automated methods for designing efficient and scalable reversible circuits that can handle large and complex reversible functions.
- Reversible computation is energy-efficient, and the development of reversible circuits is a promising approach to reducing energy consumption in computing systems. However, the optimization of energy consumption in reversible circuits is an open problem that requires the development of new design methods and algorithms.

## VII. CONCLUSION

The broad knowledge about reversible gates and their uses that we researched in this paper is compiled from several academic works and research publications. Future designers will be assisted by this survey in creating complicated computer systems with reversible gates. Low-power digital circuit design has taken into account reversible gates. There are numerous uses for reversible gates in cutting-edge technologies, including nanotechnology, quantum computing, optical computing, spacecraft, smart tags on inventory, low-power CMOS, biomolecular computation, communication, and FPGAs. Several important questions are still unanswered which we are confident will be answered in the coming years. These questions can serve as prompts for readers to reflect on the content and potentially explore related research directions or areas of improvement. Hence, the researchers can

**TABLE 11.** Parity preserving reversible gates having A, B, C, D, and E as inputs and P, Q, R, S, and T as outputs.

| Gate | P | Q | R | S | T |
|---|---|---|---|---|---|
| MX [29] | AB | $A\overline{B}\oplus BC$ | B+C | - | - |
| CQCA [107] | A | AB+BC+AC | $\overline{A}B+BC+\overline{A}C$ | - | - |
| NFT [108] | $A\oplus B$ | $B\overline{C}\ \oplus A\overline{C}$ | $BC+A\overline{C}$ | - | - |
| PPRG [109] | $A\oplus B$ | $AC+\overline{B}\ \overline{C}$ | $\overline{A}C+\overline{B}\ \overline{C}$ | - | - |
| DFG [110] | A | $A\oplus B$ | $A\oplus C$ | - | - |
| Fredkin [27] | A | $\overline{A}B+AC$ | $\overline{A}C+AB$ | - | - |
| MCF [27] | A | $\overline{A}B+AC$ | $AB+\overline{A}C$ | - | - |
| KMD Gate1 [111] | A | $\overline{A}C\ \oplus A\overline{B}$ | $\overline{A}B\ \oplus A\overline{C}$ | - | - |
| KMD Gate2 [111] | A | $\overline{A}\ \overline{B}\ \oplus A\overline{B}$ | $\overline{A}\ \overline{C}\ \oplus AB$ | - | - |
| RQCA-PP [112] | $\overline{A}$ | $A\oplus B\oplus C$ | $\overline{A}B+AC$ | $(\overline{A}\ \overline{B}+AC)\oplus D$ | - |
| RCQCA [113] | $A\overline{B}+BD$ | B | $(A\overline{B}+BD)\oplus C$ | $A\oplus D$ | - |
| CRG [35] | $A\oplus B$ | AB+BC+AC | $\overline{B}C+A(\overline{B\ \oplus\ C})$ | $A\oplus D$ | - |
| MIG [114] | A | $A\oplus B$ | $AB\oplus C$ | $A\overline{B}\oplus D$ | - |
| KMD Gate 3 [111] | $A\oplus D$ | $B\oplus C\oplus D$ | $\overline{A}C\oplus AB$ | $\overline{A}C\oplus AB\oplus D$ | - |
| IG [115] | A | $A\oplus B$ | $AB\oplus C$ | $BD\oplus\overline{B}(A\oplus D)$ | - |
| KMD Gate 4 [111] | A | $\overline{A}B\oplus A\overline{C}$ | $A\oplus B\oplus C$ | $A(B\oplus C)\oplus BC\oplus D$ | $B\oplus E\oplus D\oplus(A\oplus B)C\oplus AB$ |
| F2PG [44] | $AC+B\overline{C}$ | $A\oplus B$ | $A\oplus B\oplus C$ | $(A\oplus B)C\ \oplus\ (AB\oplus D)$ | $A\overline{B}\oplus E$ |

explore innovative circuit architectures and logic gate configurations that offer improved performance, reduced quantum cost, lower complexity, and other desirable characteristics.

## REFERENCES

[1] M. Mohammadi and M. Eshghi, "On figures of merit in reversible and quantum logic designs," *Quant. Inf. Process.*, vol. 8, no. 4, pp. 297–318, 2009.

[2] H. Gaur, T. Sasamal, A. Singh, A. Mohan, and D. Pradhan, "Reversible logic: An introduction," *Design Testing of Reversible Logic*. Singapore: Springer, 2020, pp. 3–18.

[3] R. Landauer, "Irreversibility and heat generation in the computing process," *IBM J. Res. Develop.*, vol. 5, no. 3, pp. 183–191, Jul. 1961.

[4] C. H. Bennett, "Logical reversibility of computation," *IBM J. Res. Develop.*, vol. 17, no. 6, pp. 525–532, Nov. 1973.

[5] M. P. Frank, "Approaching the physical limits of computing," in *Proc. 35th Int. Symp. Multiple-Valued Logic (ISMVL)*, 2005, pp. 168–185.

[6] M. A. Nielsen and I. Chuang, *Quantum Computation and Quantum Information*. Cambridge, MA, USA, Massachusetts Inst. Technol., 2002.

[7] V. Vedral, A. Barenco, and A. Ekert, "Quantum networks for elementary arithmetic operations," *Phys. Rev. A*, vol. 54, no. 1, p. 147, 1996.

[8] H. A. Bhat, F. A. Khanday, B. K. Kaushik, and K. A. Shah, "Design and analysis of 3× 3 reversible quantum gates," *J. Comput. Electron.*, vol. 22, no. 1, pp. 266–275, 2023.

[9] V. K. Sharma, "Reversible logic gates using quantum dot cellular automata (QCA) Nanotechnology," in *AI For Big Data-Based Engineering Applications From Security Perspectives*. Boca Raton, FL, USA: CRC Press, 2023, pp. 215–239.

[10] C. S. Cheng and A. K. Singh, "Heuristic synthesis of reversible logic–A comparative study," *Adv. Elect. Electron. Eng.*, vol. 12, no. 3, pp. 210–225, 2014.

[11] A. D. Vos and Y. V. Rentergem, "Power consumption in reversible logic addressed by a ramp voltage," in *Proc. Int. Workshop Power Timing Model., Optim. Simulat.*, 2005, pp. 207–216.

[12] T. Toffoli, "Reversible computing," in *Proc. Int. Colloq. Autom. Lang. Program.*, 1980, pp. 632–644.

[13] R. Laundauer, "Irreversibility and heat generation in the computing process," *IBM J. Res. Develop.*, vol. 5, no. 3, pp. 183–191, 1961.

[14] V. K. Semenov, G. V. Danilov, and D. V. Averin, "Classical and quantum operation modes of the reversible Josephson-junction logic circuits," *IEEE Trans. Appl. Supercond.*, vol. 17, no. 2, pp. 455–461, Jun. 2007.

[15] J. Ren, V. K. Semenov, Y. A. Polyakov, D. V. Averin, and J.-S. Tsai, "Progress towards reversible computing with nSQUID arrays," *IEEE Trans. Appl. Supercond.*, vol. 19, no. 3, pp. 961–967, Jun. 2009.

[16] J. Ren and V. K. Semenov, "Progress with physically and logically reversible superconducting digital circuits," *IEEE Trans. Appl. Supercond.*, vol. 21, no. 3, pp. 780–786, Jun. 2011.

[17] N. Kostinski, M. P. Fok, and P. R. Prucnal, "Experimental demonstration of an all-optical fiber-based Fredkin gate," *Opt. Lett.*, vol. 34, no. 18, pp. 2766–2768, 2009.

[18] S. Das and S. Ghosh, "Randomized reversible gate-based obfuscation for secured compilation of quantum circuit," 2023, *arXiv:2305.01133*.

[19] K. Morita, *Theory of Reversible Computing*. Tokyo, Japan, Springer, 2017.

[20] 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," *Int. J. Circuit Theory Appl.*, vol. 49, no. 4, pp. 990–1012, 2021.

[21] A. A. Lakshmi and G. Sudha, "Design of a reversible single precision floating point subtractor," *Springer Plus*, vol. 3, no. 1, pp. 1–20, 2014.

[22] A. K. Biswas, M. M. Hasan, A. R. Chowdhury, and H. M. H. Babu, "Efficient approaches for designing reversible binary coded decimal adders," *Microelectron. J.*, vol. 39, no. 12, pp. 1693–1703, 2008.

[23] H. M. H. Babu, M. R. Islam, S. M. A. Chowdhury, and A. R. Chowdhury, "Synthesis of full-adder circuit using reversible logic," in *Proc. 17th Int. Conf. VLSI Design*, 2004, pp. 757–760.

[24] D. Dieks, "Communication by EPR devices," *Phys. Lett. A*, vol. 92, no. 6, pp. 271–272, 1982.

[25] W. K. Wootters and W. H. Zurek, "A single quantum cannot be cloned," *Nature*, vol. 299, no. 5886, pp. 802–803, 1982.

[26] W. D. Pan and M. Nalasani, "Reversible logic," *IEEE Potentials*, vol. 24, no. 1, pp. 38–41, Feb./Mar. 2005.

[27] E. Fredkin and T. Toffoli, "Conservative logic," *Int. J. Theor. Phys.*, vol. 21, no. 3, pp. 219–253, 1982.

[28] M. Ueda, "Logical reversibility and physical reversibility in quantum measurement," 1997, *quant-ph/9709045*.

[29] H. Thapliyal and N. Ranganathan, "Design, synthesis and test of reversible circuits for emerging nanotechnologies," in *Proc. IEEE Comput. Soc. Annu. Symp. VLSI*, 2012, pp. 5–6.

[30] H. Gaur, A. Singh, A. Mohan, and D. Pradhan, "Computational analysis and comparison of reversible gates for design and test of logic circuits," *Int. J. Electron.*, vol. 106, no. 11, pp. 1679–1693, 2019.

[31] R. P. Feynman, "Quantum mechanical computers," *Opt. News*, vol. 11, no. 2, pp. 11–20, 1985.

[32] N. D. Mermin, "From classical state swapping to quantum teleportation," *Phys. Rev. A*, vol. 65, no. 1, 2001, Art. no. 12320.

[33] H. Thapliyal, S. Kotiyal, and M. Srinivas, "Novel BCD adders and their reversible logic implementation for IEEE 754r format," in *Proc. 19th Int. Conf. VLSI Design Held Jointly 5th Int. Conf. Embedded Syst. Design (VLSID)*, 2006, pp. 387–392.

[34] B. Sen, M. Dutta, M. Goswami, and B. K. Sikdar, "Modular design of testable reversible ALU by QCA multiplexer with increase in programmability," *Microelectron. J.*, vol. 45, no. 11, pp. 1522–1532, 2014.

[35] B. Bilal, S. Ahmed, and V. Kakkar, "Modular adder designs using optimal reversible and fault tolerant gates in field-coupled QCA nanocomputing," *Int. J. Theor. Phys.*, vol. 57, no. 5, pp. 1356–1375, 2018.

[36] M. K. Singh and R. Nakkeeran, "Design of novel reversible logic gate with enhanced traits," in *Proc. Int. Conf. Invent. Comput. Inform. (ICICI)*, 2017, pp. 202–205.

[37] R. Sen et al., "Priority encoder using reversible logic gates in QCA," in *Proc. 8th IEEE Annu. Inf. Technol. Electron. Mobile Commun. Conf. (IEMCON)*, 2017, pp. 319–323.

[38] M. Mamun, S. Al, and D. Menville, "Quantum cost optimization for reversible sequential circuit," 2014, *arXiv:1407.7098*.

[39] X. Ma, J. Huang, C. Metra, and F. Lombardi, "Testing reversible 1D arrays for molecular QCA," in *Proc. 21st IEEE Int. Symp. Defect Fault Tolerance VLSI Syst.*, 2006, pp. 71–79.

[40] E. Taherkhani, M. H. Moaiyeri, and S. Angizi, "Design of an ultra-efficient reversible full adder-subtractor in quantum-dot cellular automata," *Optik*, vol. 142, pp. 557–563, Aug. 2017.

[41] A. Peres, "Reversible logic and quantum computers," *Phys. Rev. A*, vol. 32, no. 6, p. 3266, 1985.

[42] S. Mamataj, D. Saha, and N. Banu, "A review of reversible gates and its application in logic design," *Amer. J. Eng. Res.*, vol. 3, no. 4, pp. 151–161, 2014.

[43] S. M. Bhat and S. Ahmed, "Design of ultra-efficient reversible gate based 1-bit full adder in QCA with power dissipation analysis," *Int. J. Theor. Phys.*, vol. 58, no. 12, pp. 4042–4063, 2019.

[44] Q. Xuemei, C. Fulong, Z. Kaizhong, G. Liangmin, L. Yonglong, and H. Min, "Design of fast fault tolerant reversible signed multiplier," *Int. J. Phys. Sci.*, vol. 7, no. 17, pp. 2506–2514, 2012.

[45] I. Gassoumi, L. Touil, and B. Ouni, "Division circuit using reversible logic gates," in *Proc. Int. Conf. Adv. Syst. Electric Technol. (IC_ASET)*, 2018, pp. 60–65.

[46] S. M. Bhat and V. Kakkar, "Design and modeling of an ultra-efficient 3x3 SSG-1 reversible gate for nanoscale applications," in *Proc. Int. Conf. Emerg. Smart Comput. Inform. (ESCI)*, 2021, pp. 720–723.

[47] N. Nafees, I. Manzoor, M. I. Baba, S. M. Bhat, V. Puri, and S. Ahmed, "Modeling and logic synthesis of multifunctional and universal 3× 3 reversible gate for nanoscale applications," in *Proc. Int. Conf. Intell. Comput. Smart Commun.*, 2020, pp. 1423–1431.

[48] I. Manzoor, N. Nafees, M. I. Baba, S. M. Bhat, V. Puri, and S. Ahmed, "Logic design and modeling of an ultraefficient 3× 3 reversible gate for nanoscale applications," in *Proc. Int. Conf. Intell. Comput. Smart Commun.*, 2020, pp. 1433–1442.

[49] H. Thapliyal and N. Ranganathan, "Design of efficient reversible binary subtractors based on a new reversible gate," in *Proc. IEEE Comput. Soc. Annu. Symp. VLSI*, 2009, pp. 229–234.

[50] T. N. Sasamal, A. K. Singh, and A. Mohan, "Efficient design of reversible alu in quantum-dot cellular automata," *Optik*, vol. 127, no. 15, pp. 6172–6182, 2016.

[51] B. Dehghan, A. Roozbeh, and J. Zare, "Design of low power comparator using DG gate," *Circuits Syst.*, vol. 5, no. 1, 2014, Art. no. 41935.

[52] S. Farzana, A. N. Bahar, and S. S. Islam, "Area efficient layout design of multiply complements logic (MCL) gate using QCA technology," *Global J. Res. Eng.*, vol. 14, no. J4, pp. 7–10, 2014.

[53] A. N. Bahar, S. Waheed, and M. A. Habib, "A novel presentation of reversible logic gate in quantum-dot cellular automata (QCA)," in *Proc. Int. Conf. Elect. Eng. Inf. Commun. Technol.*, 2014, pp. 1–6.

[54] N. K. Misra, S. Wairya, and V. K. Singh, "Frame of reversible BCD adder and carry skip BCD adder and optimization using new reversible logic gates for quantum-dot cellular automata," *Aust. J. Basic Appl. Sci.*, vol. 9, no. 31, pp. 286–298, 2015.

[55] M. Mohammadi, M. Eshghi, M. Haghparast, and A. Bahrololoom, "Design and optimization of reversible bcd adder/subtractor circuit for quantum and nanotechnology based systems," *World Appl. Sci. J.*, vol. 4, no. 6, pp. 787–792, 2008.

[56] S. Kotiyal, H. Thapliyal, and N. Ranganathan, "Mach-Zehnder interferometer based design of all optical reversible binary adder," in *Proc. Design Autom. Test Europe Conf. Exhibit. (DATE)*, 2012, pp. 721–726.

[57] P. Singh, A. Majumder, B. Chowdhury, A. Mondal, and T. Shekhawat, "Reducing delay and quantum cost in the novel design of reversible memory elements," *Procedia Comput. Sci.*, vol. 57, pp. 189–198, Dec. 2015.

[58] M. Sarvaghad-Moghaddam and A. A. Orouji, "New symmetric and planar designs of reversible full-adders/subtractors in quantum-dot cellular automata," *Eur. Phys. J. D*, vol. 73, no. 6, pp. 1–12, 2019.

[59] N. K. Misra, B. Sen, and S. Wairya, "Towards designing efficient reversible binary code converters and a dual-rail checker for emerging nanocircuits," *J. Comput. Electron.*, vol. 16, no. 2, pp. 442–458, 2017.

[60] J. C. Das and D. De, "Reversible binary subtractor design using quantum dot-cellular automata," *Front. Inf. Technol. Electron. Eng.*, vol. 18, no. 9, pp. 1416–1429, 2017.

[61] U. Garg and R. Jain, "Design and performance analysis of reversible RSG gate using QCA," *Int. J. Comput. Appl.*, vol. 139, no. 12, pp. 37–41, 2016.

[62] H. Thapliyal and M. Srinivas, "A novel reversible TSG gate and its application for designing reversible carry look-ahead and other adder architectures," in *Proc. Asia-Pac. Conf. Adv. Comput. Syst. Architect.*, 2005, pp. 805–817.

[63] N. Abdessaied, M. Amy, R. Drechsler, and M. Soeken, "Complexity of reversible circuits and their quantum implementations," *Theor. Comput. Sci.*, vol. 618, pp. 85–106, Mar. 2016.

[64] H. Maity, A. Biswas, A. K. Bhattacharjee, and A. Pal, "Design of reversible combinational circuits using new reversible logic gate," *J. Eng. Sci. Technol. Rev.*, vol. 11, no. 5, pp. 170–172, 2018.

[65] A. Slimani, A. Benslama, and N. K. Misra, "Optimal designs of reversible/quantum decoder circuit using new quantum gates," *Int. J. Theor. Phys.*, vol. 61, no. 3, pp. 1–19, 2022.

[66] F. A. Khanday and R. Akhtar, "Reversible stochastic computing," *Int. J. Numer. Model. Electron. Netw. Devices Fields*, vol. 33, no. 4, 2020, Art. no. e2711.

[67] S. K. Mitra, L. Jamal, M. Kaneko, and H. M. Hasan Babu, "An efficient approach for designing and minimizing reversible programmable logic arrays," in *Proc. Great Lakes Symp. VLSI*, 2012, pp. 215–220.

[68] T. N. Sasamal, A. K. Singh, and A. Mohan, "Design of cost-efficient QCA reversible circuits via clock-zone-based crossover," *Int. J. Theor. Phys.*, vol. 57, no. 10, pp. 3127–3140, 2018.

[69] M. Surekha, "Efficient approaches for designing quantum costs of various reversible gates," *Int. J. Eng.*, vol. 9, no. 1, pp. 57–78, 2017.

[70] S. Waheed, S. Aktar, and A. N. Bahar, "A novel design and implementation of new double Feynman and six-correction logic (DFSCL) gates in quantum-dot cellular automata (QCA)," *Eur. Sci. J.*, vol. 13, no. 15, p. 265, 2017.

[71] M. Mahfuzzreza, R. Islam, and M. B. Ali, "Optimized design of high performance reversible multiplier using BME and MHNG reversible gate," *Amer. Int. J. Res. Sci. Technol. Eng. Math. IASIR USA*, vol. 2, no. 2, pp. 227–232, 2013.

[72] V. Pareek, S. Gupta, S. C. Jain, and A. Kumar, "A novel realization of sequential reversible building blocks," in *Proc. Future Comput. 6th Int. Conf. Future Comput. Technol. Appl.*, 2014, pp. 1–6.

[73] M. A. Rahman, S. Waheed, M. A. Habib, and A. N. Bahar, "Six-correction logic (SCL) gates in quantum-dot cellular automata (QCA)," *Int. J. Sci. Eng.*, vol. 9, no. 1, pp. 9–12, 2015.

[74] M. Haghparast and K. Navi, "Novel reversible fault tolerant error coding and detection circuits," *Int. J. Quantum Inf.*, vol. 9, no. 2, pp. 723–738, 2011.

[75] M. S. Islam, M. M. Rahman, Z. Begum, and M. Z. Hafiz, "Low cost quantum realization of reversible multiplier circuit," *Inf. Technol. J.*, vol. 8, no. 2, pp. 208–213, 2009.

[76] M. Morrison and N. Ranganathan, "Design of a reversible ALU based on novel programmable reversible logic gate structures," in *Proc. IEEE Comput. Soc. Annu. Symp. VLSI*, 2011, pp. 126–131.

[77] S. Yadav and M. Saxena, "High speed time efficient reversible ALU based logic gate structure on vertex family," *Int. J. Eng. Res. Dev*, vol. 11, no. 10, pp. 72–77, 2015.

[78] T. N. Sasamal, A. Mohan, and A. K. Singh, "Design of parity preserving combinational circuits using reversible gate," in *Proc. 2nd Int. Conf. Next Gener. Comput. Technol. (NGCT)*, 2016, pp. 631–638.

[79] L. Gopal, A. K. Chowdhury, A. A. Gopalai, A. K. Singh, and B. Madon, "Reversible logic gate implementation as switch controlled reversible full adder/subtractor," in *Proc. IEEE Int. Conf. Control Syst. Comput. Eng. (ICCSCE)*, 2014, pp. 1–4.

[80] H. Thapliyal and A. P. Vinod, "Designing efficient online testable reversible adders with new reversible gate," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2007, pp. 1085–1088.

[81] D. Krishnaveni, M. G. Priya, and K. Baskaran, "Design of an efficient reversible 8x8 wallace tree multiplier," *World Appl. Sci. J.*, vol. 20, no. 8, pp. 1159–1165, 2012.

[82] P. Garg and S. Saini, "A novel design of compact reversible SG gate and its applications," in *Proc. 14th Int. Symp. Commun. Inf. Technol. (ISCIT)*, 2014, pp. 400–403.

[83] V. Kumar and D. Dhawan, "Design of reversible adder subtractor using multifunction reversible logic gate (MRLG)," *Int. J. Adv. Comput. Electron. Eng.*, vol. 1, no. 2, pp. 5–11, 2016.

[84] H. Rangaraju, A. B. Suresh, and K. Muralidhara, "Design and optimization of reversible multiplier circuit," *Int. J. Comput. Appl.*, vol. 52, no. 10, pp. 44–50, 2012.

[85] M. Haghparast and K. Navi, “A novel reversible BCD adder for nanotechnology based systems,” *Amer. J. Appl. Sci.*, vol. 5, no. 3, pp. 282–288, 2008.

[86] H. Bhagyalakshmi and M. Venkatesha, “An improved design of a multiplier using reversible logic gates,” *Int. J. Eng. Sci. Technol.*, vol. 2, no. 8, pp. 3838–3845, 2010.

[87] M. Haghparast and K. Navi, “A novel reversible full adder circuit for nanotechnology based systems,” *J. Appl. Sci.*, vol. 7, no. 24, pp. 3995–4000, 2007.

[88] A. S. M. Sayem and M. Ueda, “Optimization of reversible sequential circuits,” 2010, *arXiv:1006.4570*.

[89] T. Ravi, S. Ranjith, and V. Kannan, “A novel design of D-flip flop using new rr fault tolerant reversible logic gate,” *Int. J. Emerg. Technol. Adv. Eng. (IJETAE)*, vol. 3, no. 2, pp. 386–394, 2013.

[90] S. Mamataj, S. Chandran, and B. Das, “Implementation of sequence generator by the sequential elements (d-flip flop) of reversible gates,” *Int. J. Comput. Appl.*, vol. 975, no. 1, pp. 33–37, 2016.

[91] M. B. Ali, H. A. Rahman, and M. M. Rahman, “Design of a high performance reversible multiplier,” *Int. J. Comput. Sci.* , vol. 8, no. 6, pp. 134–141, 2011.

[92] V. Rajmohan and V. Ranganathan, “Design of counters using reversible logic,” in *Proc. 3rd Int. Conf. Electron. Comput. Technol.*, vol. 5, 2011, pp. 138–142.

[93] K. Harish Naik, G. Jyothi, K. Murulidhara, and M. Kurian, “Design and implementation of reversible sequential circuits,” *Int. J. Adv. Res. Comput. Eng. Technol.*, vol. 3, no. 4, pp. 1336–1344, 2014.

[94] N. K. Misra, S. Wairya, and V. K. Singh, “An inventive design of 4 4 bit reversible NS gate,” in *Proc. Int. Conf. Recent Adv. Innovat. Eng. (ICRAIE)*, 2014, pp. 1–6.

[95] A. B. Sudhakar and Veena M. B, “Reconfigurable single precision floating point multiplier using reversible logic,” *Transstellar J. Publ. Res. Consul.*, vol. 4, no. 6, pp. 21–28, 2014.

[96] M. Valinataj, M. Mirshekar, and H. Jazayeri, “Novel low-cost and fault-tolerant reversible logic adders,” *Comput. Elect. Eng.*, vol. 53, pp. 56–72, Jul. 2016.

[97] K. M. Murthy, G. Gayatri, and M. R. Kumar, “Design of efficient adder circuits using proposed parity preserving gate (PPPG),” *Int. J. VLSI Design Commun. Syst.*, vol. 3, no. 3, pp. 83–93, 2012.

[98] H. Bhagyalakshmi and M. Venkatesha, “Optimized multiplier using reversible multi-control input toffoli gates,” *Int. J. VLSI Design Commun. Syst.*, vol. 3, no. 6, pp. 27–40, 2012.

[99] S. Mamataj, B. Das, and A. Rahaman, “A more effective Realization of BCD adder by using a new reversible logic BBCDC,” *Int. J. Comput. Eng. Res.*, vol. 4, no. 2, pp. 13–19, 2014.

[100] M. S. Mia, M. Mukib, and M. Islam, “An extended review on reversible logic gates and their implementation,” *Int. J. Latest Eng. Res. Appl*, vol. 1, pp. 8–18, Nov. 2015.

[101] A. Sarker, A. Bose, and S. Gupta, “Design of a compact fault tolerant adder/subtractor circuits using parity preserving reversible gates,” in *Proc. 17th Int. Conf. Comput. Inf. Technol. (ICCIT)*, 2014, pp. 1–7.

[102] N. K. Misra, B. Sen, and S. Wairya, “Novel conservative reversible error control circuits based on molecular QCA,” *Int. J. Comput. Appl. Technol.*, vol. 56, no. 1, pp. 1–17, 2017.

[103] R. Khanam, A. Rahman, and Pushpam, “Review on reversible logic circuits and its application,” in *Proc. Int. Conf. Comput. Commun. Autom. (ICCCA)*, 2017, pp. 1537–1542.

[104] R. Garipelly, P. M. Kiran, and A. S. Kumar, “A review on reversible logic gates and their implementation,” *Int. J. Emerg. Technol. Adv. Eng.*, vol. 3, no. 3, pp. 417–423, 2013.

[105] V. V. Shende, A. K. Prasad, I. L. Markov, and J. P. Hayes, “Synthesis of reversible logic circuits,” *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 22, no. 6, pp. 710–722, Jun. 2003.

[106] S. Thakral and D. Bansal, “Novel reversible DS gate for reversible logic synthesis,” *Int. J. Modern Educ. Comput. Sci.*, vol. 8, no. 6, pp. 20–26, 2016.

[107] H. Thapliyal and N. Ranganathan, “Conservative QCA gate (CQCA) for designing concurrently testable molecular QCA circuits,” in *Proc. 22nd Int. Conf. VLSI Design*, 2009, pp. 511–516.

[108] S. Waheed and M. G. Rasel, “Design and implementation of new Feynman and Toffoli (NFT) gates in quantum-dot cellular automata (QCA),” *Circul. Comput. Sci.*, vol. 2, no. 4, pp. 64–67, 2017.

[109] A. Roohi, R. Zand, S. Angizi, and R. F. DeMara, “A parity-preserving reversible QCA gate with self-checking cascadable resiliency,” *IEEE Trans. Emerg. Topics Comput.*, vol. 6, no. 4, pp. 450–459, Oct.–Dec. 2018.

[110] B. Parhami, “Fault-tolerant reversible circuits,” in *Proc. 14th Asilomar Conf. Signals Syst. Comput.*, 2006, pp. 1726–1729.

[111] A. Kamaraj and P. Marichamy, “Design of integrated reversible fault-tolerant arithmetic and logic unit,” *Microprocess. Microsyst.*, vol. 69, pp. 16–23, Sep. 2019.

[112] B. Sen, M. Dutta, S. Some, and B. K. Sikdar, “Realizing reversible computing in QCA framework resulting in efficient design of testable ALU,” *ACM J. Emerg. Technol. Comput. Syst. (JETC)*, vol. 11, no. 3, pp. 1–22, 2014.

[113] N. K. Misra, S. Wairya, and B. Sen, “Design of conservative, reversible sequential logic for cost efficient emerging nano circuits with enhanced testability,” *Ain Shams Eng. J.*, vol. 9, no. 4, pp. 2027–2037, 2018.

[114] M. S. Islam, M. M. Rahman, Z. Begum, and M. Z. Hafiz, “Fault tolerant reversible logic synthesis: Carry look-ahead and carry-skip adders,” in *Proc. Int. Conf. Adv. Comput. Tools Eng. Appl.*, 2009, pp. 396–401.

[115] S. Shoaei and M. Haghparast, “Novel designs of nanometric parity preserving reversible compressor,” *Quant. Inf. Process.*, vol. 13, no. 8, pp. 1701–1714, 2014.

**SYED FARAH NAZ** (Student Member, IEEE) received the B.Tech. degree from IUST Kashmir in 2018, and the M.Tech. degree from SMVDU Katra, Jammu, in 2020. She is currently pursuing the Ph.D. degree with the Electrical Engineering Department, Indian Institute of Technology Jammu, Jammu and Kashmir, India.

She has more than 18 publications in various peer-reviewed journals and conferences. Her current research includes reliability analysis of digital circuits, fault tolerant circuits, QCA-based circuit designs, and design of security circuits. She has been awarded the prestigious “PMRF” Fellowship from August 2020.

**AMBIKA PRASAD SHAH** (Senior Member, IEEE) received the Ph.D. degree from the Electrical Engineering Department, Indian Institute of Technology (IIT) Indore, India, in 2019.

He is currently working as an Assistant Professor and leading the IC-ResQ Lab, Electrical Engineering Department, IIT Jammu, India. He is also a Visiting Professor with the University of Virginia, USA. Before joining IIT Jammu, he worked as a Postdoctoral Fellow with the Institute for Microelectronics, TU Vienna, Austria. He has authored/coauthored more than 75 research papers in peer-reviewed international journals and conferences. His current research interest includes reliability analysis of digital circuits, design for reliability, fault-tolerant circuits, reliability modeling, low-power high-performance circuit designs, and hardware security primitives. He is a Guest Editor of Springer and Elsevier journals. Moreover, he served as the Conference Chair and the Publication Chair for the VDAT-2022, the Fellowship Chair for the VLSID-2022, and the Publicity Chair for the iSES-203. He is also part of the technical program committees of various conferences (MWSCAS, VDAT, VLSID, and MICRO). He is a Fellow of IETE.