



## Logical Fault Detection Approach for Mixed Control Flipping Faults in Reversible Circuits

Mousum Handique & Amrit Prasad

**To cite this article:** Mousum Handique & Amrit Prasad (19 May 2025): Logical Fault Detection Approach for Mixed Control Flipping Faults in Reversible Circuits, IETE Journal of Research, DOI: [10.1080/03772063.2025.2503340](https://doi.org/10.1080/03772063.2025.2503340)

**To link to this article:** <https://doi.org/10.1080/03772063.2025.2503340>



[View supplementary material](#)



Published online: 19 May 2025.



[Submit your article to this journal](#)



Article views: 16



[View related articles](#)



[View Crossmark data](#)

# Logical Fault Detection Approach for Mixed Control Flipping Faults in Reversible Circuits

Mousum Handique<sup>1</sup> and Amrit Prasad<sup>2</sup>

<sup>1</sup>Assam University, Silchar 788011, India; <sup>2</sup>Rapawalk Fashion Technologies, Bengaluru, Karnataka 560025, India

## ABSTRACT

Due to the rapid development of computing machines in terms of micro-architectural designs, millions of transistors are involved per chip, which leads to complex digital circuit design and meets the demands of more computational power. It indicates that a single transistor's size is advancing at the atomic scale in future computing technologies. The gain of loss-less bit information in reversible logic computation is highly acceptable in modern computing devices. Therefore, motivated by these, researchers have explored the reversible logic concept for applying newer technologies such as quantum computing, optical computing, nanotechnology, DNA computing, digital communication systems, low-power CMOS design, etc. In the current literature, numerous developments on synthesis and optimization in reversible circuits have been proposed. Simultaneously, in order to observe the correct behavior of reversible circuits in terms of performing the correct functionality and integrity performance, testing plays an important role. In this article, we consider a fault model labeled the Mixed Control Flipping Fault (MixCFF) model that is applied in the reversible circuit, which is also relevant to the quantum circuits for representing the faults. The proposed work presents an automatic test pattern generation (ATPG) algorithm to detect the MixCFF. The proposed MixCFF model enables the correlation between the existing fault models in the reversible circuit, which are also presented in the proposed work. Experimental results are evaluated based on the MixCFF detection and the MixCFF fault coverage range with the help of different benchmark circuits using the suggested ATPG algorithm.

## KEYWORDS

Complete test set; fault coverage; fault detection; mixed control flipping faults; reversible logic circuit

## 1. INTRODUCTION

The modern digital system comprises a complex digital circuit due to the rapid progress of the fabrication technology and more digital components are included onto the silicon chip. According to Moore's law [1], the density of the integrated structure of the circuit exponentially improves, *i.e.* the number of transistors doubles every 18 months. The truth of Moore's law still exists in modern semiconductor electronic technologies. Due to the newly improved fabrication process, modern electronic devices can shrink transistor dimensions and increase the computational speed. As a result, the digital circuit leads to a complex design with smaller interconnection features that cause high power dissipation and increase the probability of error rate in the circuit. Moreover, the computation cannot be reverted in conventional logic circuits, *i.e.* the input information of the irreversible gate is lost or erased when the gate operation is executed [6]. Therefore, the operation of the conventional logic computation is in an irreversible manner, where the output is not uniquely retrieved from the inputs [42].

In classical OR gate operation, the input logic values are only uniquely retrievable when the output is produced with logic value 0. However, the input values of an OR gate operation are not determined when the output logic value is 1. More precisely, conventional logic computing follows logical irreversibility such that the previous computation state is not guaranteed to reconstruct the current state. The logically irreversible gate operation dissipates some energy during the computation because of the one-bit information loss or erase. In 1961, Rolf Landauer [3] postulated that logical irreversibility is associated with physical irreversibility that requires minimal heat generation per machine cycle and  $kT \approx 3 \times 10^{-21}$  amount of energy is needed for each irreversible function, where  $k$  is the Boltzmann's constant and  $T$  is the absolute temperature of the system in degrees Kelvin. The required amount of energy dissipation  $kT$  is distributed for standardizing signals of the system and these signals are independent of their exact logical history [6]. Therefore, when the logically irreversible operations are performed in a physical device, every one-bit of information is lost

during computation, which leads to dissipating at least  $kT\ln 2$  Joules of energy and generates a corresponding amount of entropy [2,43].

Reversible logic computing is defined in contrast to classical logic computing in that the mapping from one state to its successor must be one-to-one. In general, reversible logic computation uses operations that are easily and precisely reversible or undone to carry out the computation. To satisfy the reversibility of the reversible logic computation in any deterministic device, the reversible function is executed where  $n$ -input and  $n$ -output are bijective transformations from each other. Then, it is called logical reversibility, which ensures the loss-less information during computation. In another way, physical reversibility exists when the device is capable of running backward to forward propagation, which leads to energy efficiency due to the generation of less amount of new physical entropy. In 1973, C. H. Bennett [4] showed that reversible logic computation can reduce or minimize power dissipation when the technologies are accurately implemented by the reversible logic operation. Therefore, the researchers are motivated to explore reversible logic design to effectively utilize reversible logic in current CMOS technology. Moreover, the implementation framework of the newer technologies such as the trapped-ion technology [5], quantum computing [6,7], DNA computing [8,9], optical computing [10], etc., are used the concept of reversible computation.

A reversible logic function is computed by the reversible operations and the execution of these operations is performed by the set of reversible gates. A circuit is said to be reversible if it is designed and implemented only by the reversible gates [11]. The properties of the reversible circuit allow the realization of  $n$ -input and  $n$ -output functions, where each input generates a unique output and vice versa. These properties are included (i) the input of the reversible circuit always produces the unique output and the output of the circuit is also capable of retrieving its corresponding input (ii) the number of inputs is equal to the number of outputs, (iii) the circuit is free from the fanout and feedback connections [6]. The synthesis of the reversible circuit depends on the various reversible gate libraries, where some of the gate libraries are commonly used, like NOT-CNOT-Toffoli (NCT), NOT-CNOT-Toffoli-Fredkin (NCTF), Multiple Controlled Toffoli (MCT) or  $k$ -CNOT and Multiple Controlled Fredkin (MCF). The optimal design and efficient synthesis process of the reversible circuit are the challenging tasks for evaluating various performance parameters such as gate count, quantum cost, garbage outputs and ancilla inputs [12]. As a result, various approaches for

the synthesis [11,13,14] and optimization [15,16] of the reversible circuits have been introduced.

Owing to the swift advancement in synthesizing and optimizing reversible circuits, accuracy in functionality behavior and integrity performance are also crucial. Any type of defect in a circuit caused by physical disturbances in manual design and environmental factors [17] is referred to as a fault and might result in incorrect functional behavior of the circuit. For this purpose, there is a need to represent an abstract model that is capable of modeling many different physical faults that occur in the circuit [18]. In the circuit testing domain, an abstract model applied to represent the physical faults is called a logical fault model, which helps reduce the complexity of fault analysis and is also applicable for technology independent [18]. For this reason, conventional and reversible fault models such as stuck-at fault [19,20], bridging fault [20,21] and cell fault [19] have been used to detect defects. In addition to these classic fault models, some fault models [5,22,23] have been studied specifically for reversible circuits.

In this context, testing is a process that examines circuits in order to identify erroneous functioning and also analyzes the circuits to obtain the desired performance. In common, the circuit preparation for the testing process includes two stages: fault detection and fault localization. The fault detection stage is dependable for identifying the faults within the circuit, and the fault localization stage evaluates the precise location of these faults. The testing is performed on the circuit using two different approaches: online and offline. An online testing is defined in the circuit when the fault(s) is recognized during the circuit's operation. In contrast, offline testing is executed by applying the sequence of inputs and observing the sequence of outputs to distinguish the fault-free and faulty circuits. In offline testing, the sequence of inputs is applied to detect the faults represented as input test vectors, and the formulation of several test vectors is denoted as a test set. When the circuit is under offline testability, the test set is called a complete test set if it successfully distinguishes all the occurrence faults; it is referred to as a minimal complete test set if it has a minimal number of test vectors. The perfection of the fault analysis for a given fault model in the circuits depends on generating an efficient complete test set that is relatively straightforward in reversible circuits compared to the conventional circuits due to reversible properties controllability and observability [19]. However, testing a reversible circuit can be challenging in achieving quality performance measures, like complete test set size, number of inputs, number of gates, execution time, fault coverage range, and the

correlation of other fault models. Thus, various methodologies in offline testing of the reversible circuits have been presented in the current literature; some of these approaches are based on the deterministic and randomized algorithms [5,19,24,25,26,27], while other belongs to the design for testability (DFT) [28,29].

In this article, we look at a complete test set generation problem in order to find the Mixed control flipping faults (MixCFF) under the control flipping fault model (CFF) in the mixed-controlled based  $k$ -CNOT circuits. To solve this problem, we develop an ATPG algorithm that generates a complete test set. This test set detects the MixCFFs in mixed-controlled based  $k$ -CNOT circuits. In this work, we compare the fault coverage range of the generated test set of the proposed method with other types of fault models such as single missing gate fault (SMGF), multiple missing gate fault (MMGF), repeated-gate fault (RGF) and partial missing gate fault (PMGF) in the reversible circuit.

The specific contributions of this work are as follows:

- (1) The complete test set is constructed using an ATPG algorithm for identifying the MixCFFs in the mixed-controlled based  $k$ -CNOT circuits.
- (2) The fault coverage range with other fault models is determined by the complete test set for MixCFFs.

The remaining parts of the article are organized as follows. Section 2 presents the basic fundamentals related to reversible circuits and a short overview of the fault models. The discussion of the existing works for testing in reversible circuits is also discussed in this section. Section 3 provides a brief introduction to the MixCFF model. The proposed methodology for generating the complete test set for detecting the MixCFFs and also addressing the correlation between the MixCFF and existing fault models are explained in Section 4. In Section 5, demonstrates the experimental results based on the various benchmark circuits. Finally, the conclusion of the article is reported in Section 6.

## 2. PRELIMINARIES

### 2.1 Reversible Logic Function and Reversible Gates

A function  $f(x_1, x_2, \dots, x_n)$  that contains  $n$  Boolean variables is considered a reversible logic function if the function  $f$  realizes both injective (one-to-one) and surjective (onto) functions. More precisely, the function  $f: \mathbb{B}^n \Rightarrow \mathbb{B}^m$  permits a permutation over the set of input vectors

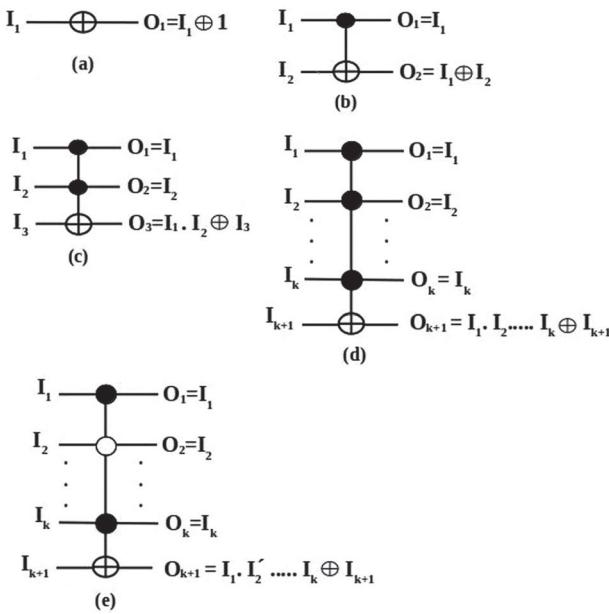
**Table 1: Illustration of 3-input, 3-output Reversible function  $f$**

Inputs			Outputs		
$I_1$	$I_2$	$I_3$	$O_1$	$O_2$	$O_3$
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	1
0	1	1	0	1	0
1	0	0	1	0	0
1	0	1	1	0	1
1	1	0	1	1	1
1	1	1	1	1	0

$I = I_1, I_2, \dots, I_n$  and the set of output vectors  $O = O_1, O_2, \dots, O_m$  in such a way that one-to-one and onto mapping are established between input and output vectors, and the number of input and output vectors of the function  $f$  is equal (i.e.  $n = m$ ). These above two properties allow to make a function  $f$  to be reversible. The reversible function  $f$  is demonstrated with the help of Table 1. Here,  $f$  is 3-input, 3-output reversible function that is defined  $(I_1, I_2, I_3) \Rightarrow (O_1, O_2, O_3)$ , where  $O_1 = I_1$ ,  $O_2 = I_2$  and  $O_3 = I_2 \oplus I_3$ . Thus, the reversible function  $f$  satisfies an equal number of input and output vectors. Also, Table 2 shows the input-output mappings are  $(000 \Rightarrow 000, 001 \Rightarrow 001, 010 \Rightarrow 011, 011 \Rightarrow 010, 100 \Rightarrow 100, 101 \Rightarrow 101, 110 \Rightarrow 111, 111 \Rightarrow 110)$ . It indicates that one-to-one and onto mappings are established from the input vectors to their corresponding output vectors in reversible function  $f$ .

To realize the reversible function, the reversible circuit is used that is comprised of the linear cascade of reversible gates. In general, the structure of a reversible gate consists of  $k$ -inputs and  $k$ -outputs, which is denoted by the  $k \times k$  reversible gate [12]. The reversible gate follows the two properties to ensure the characteristics of the reversible function, which are (i) function/mapping is bijective and (ii) inputs and outputs are equal. In the current literature, several reversible gates are introduced, but we focus only on those relevant to these proposed works, which are explained below.

- (i) The NOT gate of the conventional circuit is only the reversible gate that satisfies the reversible logic function. Similar to the conventional NOT gate is formulated with 1-input and 1-output ( $1 \times 1$ ). In Figure 1 (a), the input is defined as  $I = I_1$  and the output is defined as  $O = O_1$ , where  $O_1 = I_1 \oplus 1$ .
- (ii) The CNOT (Controlled-NOT) gate is also called as Feynman gate [7] and it is formulated with 2-inputs and 2-outputs ( $2 \times 2$ ). Here, the input  $I = I_1, I_2$  represents the output  $O = O_1, O_2$ , where  $O_1 = I_1$  and  $O_2 = I_1 \oplus I_2$ , as shown in Figure 1 (b). The input  $I_1$  is connected to positive control connection



**Figure 1:** Structure and Gate operation of classical reversible gates: (a) NOT gate [48] (b) CNOT gate [48] (c) Toffoli gate [48] (d)  $k$ -CNOT gate [48] (e) Mixed-polarity based  $k$ -CNOT gate

(•) and the input  $I_2$  is connected to target connection ( $\oplus$ ). The input  $I_1$  remains unchanged at output  $O_1$  and the output vector  $O_2$  of the target connection is the complement form of the input vector  $I_2$ , when only the first input vector is assigned as logic value 1.

- (iii) The Toffoli gate is a  $3 \times 3$  gate and it is also known as Controlled-Controlled-NOT gate or 2-CNOT gate [6,23,30,32,45,46]. The Toffoli gate is formulated with 3-input and 3-output, where two input lines are control lines and one is a target line. The output of the control lines remains unchanged from their corresponding input of the control lines after the Toffoli gate operation. The output of the target line is only inverted based on the corresponding input of the target line when both inputs of the control lines are logic value 1; otherwise, the target output remains unchanged from their corresponding input. The Toffoli gate maps  $(I_1, I_2, I_3) \rightarrow (I_1, I_2, I_1 \cdot I_2 \oplus I_3)$ , which is the bijective function. Figure 1 (c) illustrates inputs  $(I_1, I_2, I_3)$  are mapping to the outputs  $(O_1, O_2, O_3)$ , where  $O_1 = I_1$ ,  $O_2 = I_2$  and  $O_3 = I_1 \cdot I_2 \oplus I_3$ .
- (iv) A  $k$ -CNOT gate is a  $(k+1) \times (k+1)$  gate, where  $k$  denotes as control inputs  $c_1, c_2, \dots, c_k$  and  $t$  is a target input [5,47]. The  $k$ -CNOT gate is formulated by  $(k+1)$ -input and  $(k+1)$ -output [22], where the output of the  $k$  control lines propagates the same respective input of the  $k$  control lines and output of target  $t$  at the  $(k+1)^{\text{th}}$

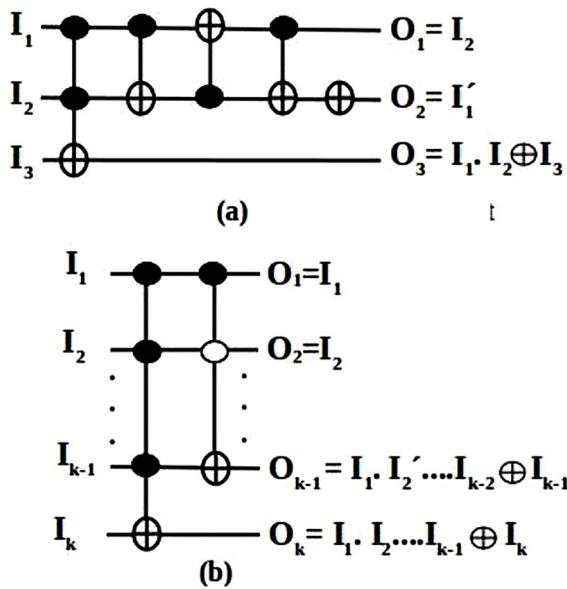
line inverts if and only if all the  $k$  control inputs are assigned by logic value 1. The  $k$ -CNOT gate maps  $(c_1, c_2, \dots, c_k, t) \rightarrow (c_1, c_2, \dots, c_k, t \oplus (c_1 \cdot c_2 \dots c_k))$ . Figure 1 (d) is demonstrated as mapping inputs  $(I_1, I_2, \dots, I_k, I_{k+1})$  to outputs  $(O_1, O_2, \dots, O_k, O_{k+1})$ , where  $O_1 = I_1$ ,  $O_2 = I_2, \dots, O_k = I_k$ ,  $O_{k+1} = (I_1 \cdot I_2 \dots I_k) \oplus I_{k+1}$ . Based on the  $k$ -CNOT gate structure, it is also comprised of mixed polarity-based positive/negative (•/○) control lines and a target line  $t$ . Suppose,  $k$  control inputs  $c_1, c_2, \dots, c_k$  and  $t$  is a target input that lie on a  $(k+1)^{\text{th}}$  line, where  $c_1, c_2, \dots, c_k$  are connected to the positive (•) control lines and  $c_2$  is the negative (○) control line. In this case, the mixed polarity based  $k$ -CNOT gate maps  $(c_1, c_2, \dots, c_k, t) \rightarrow (c_1, c_2, \dots, c_k, t \oplus (c_1 \cdot c_2 \dots c_k))$ . The input and output of the  $k$  control lines remain unchanged. The output of the target line  $t$  is inverted if and only if all the positive (•) and negative (○) control input lines are assigned by the logic value 1 and 0, respectively; otherwise, the target output remains unchanged of their corresponding target input. Figure 1 (e) illustrates as mapping inputs  $(I_1, I_2, \dots, I_k, I_{k+1})$  to outputs  $(O_1, O_2, \dots, O_k, O_{k+1})$ , where  $O_1 = I_1$ ,  $O_2 = I_2, \dots, O_k = I_k$ ,  $O_{k+1} = (I_1 \cdot I_2 \dots I_k) \oplus I_{k+1}$ .

## 2.2 Structure of Reversible Circuits

A reversible circuit is a cascade structure of the reversible gates  $G_1, G_2, \dots, G_N$ , where  $N$  is the total number of gates. The output of the gate  $G_i$ , for  $1 \leq i \leq N$ , behaves as an input of the gate  $G_{i+1}$  and the operations of the gate  $G_{i+1}$  is completed only after the operations of the gate  $G_i$ . Therefore, the logical design of the reversible circuit does not permits the fan-out and feedback connections [6]. The efficient synthesis of the reversible circuit reduces the complexity measure that can be gained by the minimum number of gates used (*i.e.*, minimum gate count), minimum number of unused outputs (*i.e.*, garbage output), and also minimum delay. Several gate libraries are used to formulate the reversible circuit, where NCT (NOT, CNOT, Toffoli) [30], GT (Generalized Toffoli) [30,31] and NCTF (NOT, CNOT, Toffoli, Fredkin) [31] libraries. The structure and the gate operations of the NCT and GT library based reversible circuits are shown in Figure 2 (a) and (b), respectively.

## 2.3 Fault Model

Similar to conventional logic circuits, several fault models are applied to reversible logic circuits to perform the testing process for analyzing the faults. A fault model



**Figure 2:** Reversible circuit: (a) NCT based library (b) GT based library

is defined by the mathematical model, and it represents the abstract level of physical imperfections in the circuit. These abstract levels of physical imperfections are mathematically modeled by the different categories, such as structural, behavioral, functional, and geometric [17]. Therefore, a fault model determines the accurate interpretation of the faults that occur in the circuit. Moreover, the number of input lines and gate count are associated with the area/size of the circuit design and are measures of the circuit complexity [44]. The time complexity for testing in terms of generating the test patterns to detect the faults depends on the circuit design. In this connection, a fault model relates to the circuit design that describes the various fault possibilities that occur during the operation or manufacturing of the circuit. Hence, the complexity of generating test patterns for detecting faults is reduced by applying the fault model [23]. In the context of the reversible circuit, most of the existing fault models are related to the gate level and gate interconnections, which are covered by the structural fault model. The proposed work in this article discusses explicitly the MixCFF model in reversible circuits, which is also considered the structural fault model. In Section 3, the detailed structure and occurrence of the MixCFFs are explained.

## 2.4 Related Work

The correctness and integrity performance of the traditional circuit and the reversible circuit are evaluated by the testing approaches, like offline and online testing. In this section, we focus only on offline testing approaches of

reversible circuits that are relevant to our work. In 2011, the authors in [32] proposed a testable design approach for  $k$ -CNOT circuits, in which the copy of each gate is included in the design and only one additional control line is added to create the universal test set(UTS). The generated UTS of size  $(n + 1)$  for the  $n$  number of input lines is ability to identify all the SMGFs, detectable RGFs and PMGFs. In addition, one test vector of the UTS is applied to determine the fault location of the SMGF model. In 2012, Zamani *et al.* [24] proposed a compact test generation method that achieves 100% fault coverage for SMGF and RGF in reversible circuits. The proposed method is implemented by the Circuit Self-Test Path (CSTP) that is applied to reversible circuits, where generated outputs are used as the inputs of the circuit to produce the test vector with limited test cycles for the purpose of fault coverage. In 2013, the authors in [33] proposed a fault ordering scheme for ATPG of reversible circuits. The proposed fault ordering scheme is categorized as the “easy” and “hard” faults, which depend on the number of control lines in a gate with the faults, like a single missing control fault (SMCF) and SMGF in a given reversible circuit. The proposed fault ordering scheme showed that the size of the test set is reduced by up to 65% while considering the “hardness” of a fault that is achieved by targeting the fault of a gate containing a large number of control lines. In 2014, a Boolean difference method to generate the test set has been proposed by author in [25] for identifying the SMGF in  $k$ -CNOT circuits. In 2015, the authors in [34] proposed a DFT technique for the reversible circuits to detect all possible detectable missing gate faults in  $k$ -CNOT circuits. The proposed DFT is designed with the help of Fredkin gates using the bit-swapping concept. The generated universal test set size  $(n + k + 2)$  from the DFT could detect all the missing gate faults that occurred in the original circuit, where  $n$  is the input lines and  $k$  is a maximum number of control connections in  $k$ -CNOT gates. In 2016, the authors in [35] proposed a fault detection scheme for detecting the SMGF and PMGF containing wholly positive or negative or mixed controlled  $k$ -CNOT gates. In 2017, the authors in [36] introduced an ATPG approach to generate SMGFs, initially stored in a BDD (Binary Decision Diagram), and to test for two consecutive missing gate failures (MMGFs) by generating test patterns using two gates dependency analysis. In 2018, the authors in [37] proposed a heuristic test generation method based on the genetic algorithm for identifying the various types of faults, like SMGF, PMGF, stuck-at and bridging faults in reversible circuits. The proposed heuristic method involves two approaches: one is considered the random search, and the second is based on the direct search for selecting the needful test vectors

to escape an exhaustive search for the selection of test vectors. In 2019, in a paper [21], the authors proposed an ATPG method that uses the path level expression defined by the circuit structure of the  $k$ -CNOT circuit to define the minimum complete test set that can be used to detect the complete bridging types with a fault coverage of 100%. In 2020, the authors in [27] investigated a scheme to detect and localize the SMGF, PMGF and MMGF in  $k$ -CNOT circuits. The proposed scheme is deployed to compute the test vectors initially. Next, a unique test set is generated by applying the Boolean difference method. The generated unique test set is applied to detect and locate the faults. In 2021, the authors in [38] proposed a DFT technique to detect the SMGF, PMGF and MMGF in reversible circuits, where the different sets are formulated based on the cluster of gates. Each prepared cluster of gates connects to an additional input line that acts as an extra control connection to the respective gates. Now, the test set is generated using the fault detection technique of the SMGF and PMGF and applied to the DFT circuit to detect the faults. In 2023, the authors in [39] introduced a testable design approach for diagnosing the positive control flipping faults (PCFFs) using a  $k$ -CNOT circuit. The testable design approach proposed by [39] showed that only one test vector can detect all the potential PCFFs in the  $k$ -CNOT circuits. The parity bit operation is additionally extended to the tested design circuit to determine the location of the faulty gate.

Based on the above literature review, we can observe that the offline testing of reversible circuits to determine the faults is carried out using various techniques such as heuristic testing, deterministic testing, and DFT. The main challenging tasks of the offline testing approaches are generating efficient test vectors, optimal solutions, less computation costs, and low circuitry overhead costs. The heuristic approaches have been targeted to create an efficient test set for determining the detection of faults but do not provide any assurance for the optimal solution. In the deterministic approach, the minimal solution is achieved for generating the complete test set with the expansive computation costs for the large circuits. In contrast, the DFT approaches are expansive due to the additional overhead cost of circuitry. The generation of the mixed test set capable of covering a maximum number of faults under the different fault models in reversible circuits is limited.

The study of the proposed work elaborates on the MixCFF model in mixed-controlled based  $k$ -CNOT circuits. The proposed work establishes the correlation between the MixCFF and existing fault models in terms of fault coverage range by applying the generated complete test

set for MixCFF. In this work, we propose an ATPG algorithm for generating the complete test set for the MixCFF and verifying the connection with the existing fault models in reversible circuits.

### 3. MIXED CONTROL FLIPPING FAULT (MIXCFF) MODEL

As per our earlier discussion, the  $k$ -CNOT circuit is constructed in the linear form of  $k$ -CNOT gates. The  $k$ -CNOT gate operation is performed by the control connection(s), unconnected connection(s) and the target connection. The output of the control connection(s) and unconnected connection(s) of the gate are propagated the same as the input. The functional logic operation is different in the output of the target connection, which is dependent on the control connection(s) associated with the target connection. Two types of control connections are used in the mixed-polarity based  $k$ -CNOT gate; one is represented as a positive control connection (marked as  $\bullet$ ), and the other is defined as a negative control connection (marked as  $\circ$ ). The logical operation for both ( $\bullet/\circ$ ) the control connections are distinguished in the target connection. The control connection may be flipping from positive control to negative control or vice versa due to the wrong interconnection within the gate. It is also prescribed by the design fault from the designer and considered a structural fault related to the gate level interconnection. The control flipping occurs in the form of different variants, such as positive to negative controls flipping and negative to positive controls flipping, and both are present based on mixed-controlled  $k$ -CNOT circuit structure. In this work, both the control flipping (*i.e.*, positive to negative controls and vice versa) are considered within the mixed-controlled based  $k$ -CNOT circuit labeled as Mixed Control Flipping Fault (MixCFF).

The mixed control flipping fault (MixCFF) model is considered when both positive and negative control connections are flipped to negative and positive control connections, respectively, as shown in Figure 3. It is noted that the MixCFF occurs when the circuit contains both positive and negative control connections in  $k$ -CNOT gates.

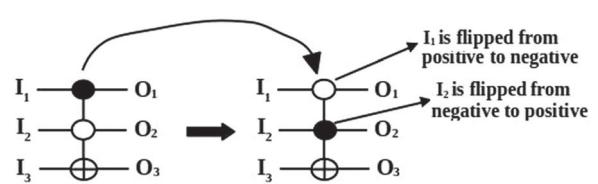
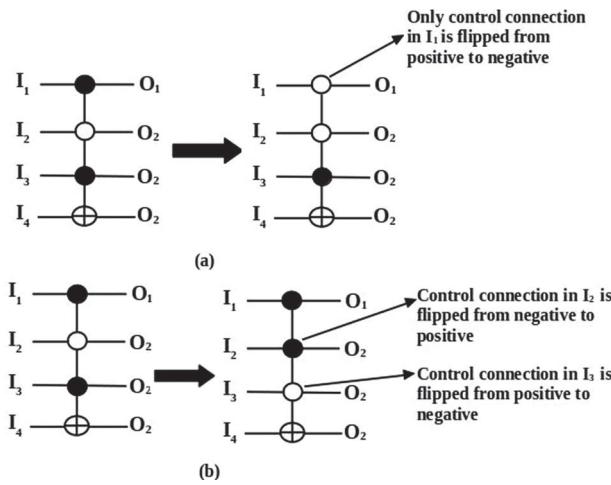


Figure 3: Illustration of MixCFF in Toffoli gate



**Figure 4:** Demonstration of (a) m-SCFF of MixCFF (b) m-MCFF of MixCFF

The MixCFF is divided into two types-mixed single control flipping fault (*m*-SCFF) and mixed multiple control flipping fault (*m*-MCFF). An *m*-SCFF occurs in a gate when a single control connection is flipped either from negative to positive or vice-versa, as shown in Figure 4 (a). In contrast, the *m*-MCFF defines when more than one control connection is flipped either from a negative to a positive control connection or vice-versa, as shown in Figure 4 (b).

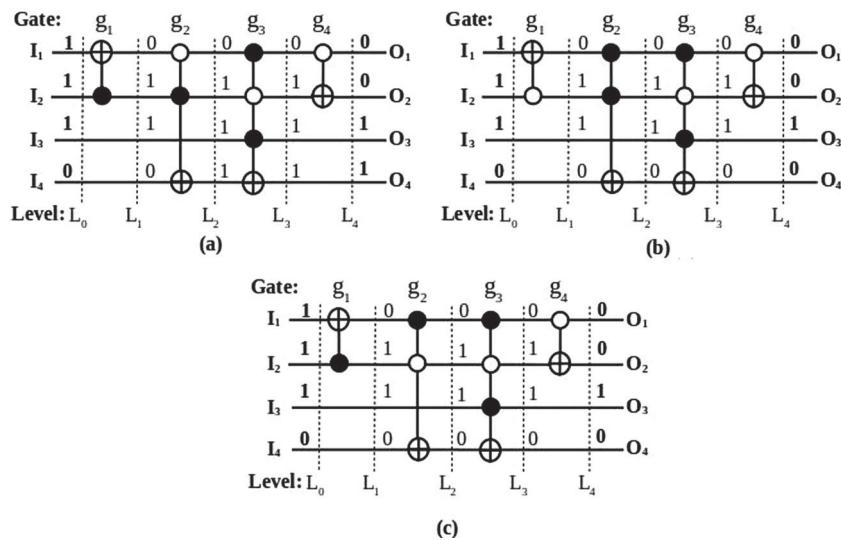
**Example 3.1:** In this example, we consider the reversible circuit that contains mixed controlled connection(s) in k-CNOT gates, as shown in Figure 5 (a). Let us consider the initial input test vector  $<1110>$  at level  $L_0$  applies to the fault-free circuit and the primary output would be

$<0011>$  at level  $L_4$ , as shown in Figure 5 (a). In Figure 5 (b), the negative control connection flips to the positive control connection at input line  $I_1$  of  $g_2$  gate, due to the effect of *m*-SCFF. Therefore, the initial input test vector  $<1110>$  produces the faulty primary output test vector  $<0010>$  at level  $L_4$  instead of fault-free primary output vector  $<0011>$ . Figure 5 (c) shows the *m*-MCFF occurs at the control connections in the input lines  $I_1$  and  $I_2$  of the gate  $g_2$ , where negative control connection flips to the positive control connection at the input line  $I_1$  and positive control connection flips to the negative control connection at the input line  $I_2$ . Due to the *m*-MCFF effects, the faulty primary output test vector  $<0010>$  is generated at level  $L_4$ .

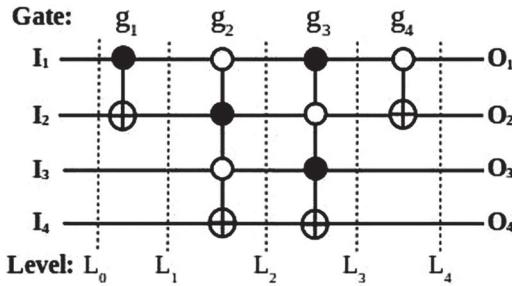
### 3.1 Generating the Total Number of MixCFFs

In this section, we provide a generalized formula to evaluate the total number of MixCFFs in a given  $k$ -CNOT circuit. The formula is given as:  $\sum_{i=0}^{n-1} k_i(2^i - 1)$ , where  $0 \leq i \leq n-1$ ,  $n$  represents the total number of input lines in a given circuit and  $k$  represents gates having same number of control connections. More precisely,  $k_i$  refers to the collective number of gates that are containing  $i^{\text{th}}$  number of control connection(s).

**Example 3.2:** Let us consider the mixed controlled based  $k$ -CNOT circuit having 4-input lines and 4 number of gates, as shown in Figure 6. The total number of MixCFFs is evaluated using the generalized formula as given below. Here,  $n = 4$ ,  $n-1 = 3$ ,  $k_0 = 0$ ,  $k_1 = 2$ ,  $k_2 = 0$  and  $k_3 = 2$ , as per the given circuit, as depicted in Figure 6.



**Figure 5:** Demonstration of (a) the fault-free circuit (b) *m*-SCFF occurs at line  $I_1$  of the gate  $g_2$  (c) *m*-MCFF occurs at both input lines  $I_1$  and  $I_2$  of the gate  $g_2$



**Figure 6:** Evaluating the total number of mixed controlled based  $k$ -CNOT circuit

Therefore,

$$\begin{aligned} & \sum_{i=0}^{n-1} k_i(2^i - 1) \\ &= k_0(2^0 - 1) + k_1(2^1 - 1) + k_2(2^2 - 1) + k_3(2^3 - 1) \\ &= 0 \times 1 - 1 + 2 \times (2 - 1) + 0 \times (4 - 1) + 2 \times (8 - 1) \\ &= 0 + 2 + 0 + 14 \\ &= 16 \end{aligned}$$

Hence, the total number of MixCFFs that can be generated in a given circuit is 16. It has been observed that the total number of m-SCFF can be given by the total number of controls ( $\bullet/\circ$ ) present in a given reversible circuit. The total number of m-MCFF can be obtained by subtracting total number of m-SCFF from the total number of MixCFFs.

Table 2 illustrates the total number of  $m$ -SCFFs and  $m$ -MCFFs are present in the mixed controlled based  $k$ -CNOT circuits, which are calculated by the generalized formula. Here, column 5 and column 6 are indicated the total number positive and negative control connections are involved, respectively, in various mixed controlled based  $k$ -CNOT circuits. The last column 7 indicates the total number of MixCFFs in a given circuit.

**Lemma 3.1:** The proposed generalized formula  $\sum_{i=0}^{n-1} k_i(2^i - 1)$  is applied to evaluate the total number of MixCFFs in a mixed-control based  $k$ -CNOT circuit.

*Proof.* Let us consider a mixed-control based  $k$ -CNOT reversible circuit, where all control connections are either positive or negative, or both are mixed in  $k$ -CNOT gates. Suppose the  $k$ -CNOT gate has only one control connection for the  $n$  number of input lines. In this case, a single control connection flips from positive to negative or vice-versa. Thus, the total number of MixCFF generates 1 in a given  $k$ -CNOT gate. Therefore, if we consider “ $i$ ” as the total number of control connections per  $k$ -CNOT gate, then the total number of MixCFFs on each gate is given by

$2^{i-1}$  for the input lines  $n$ . In the  $k$ -CNOT gate, the maximum number of control connections can occur as  $n-1$  for the  $n$ -input lines in the circuit. As a result, the total number of MixCFFs in each type of  $k$ -CNOT gate based on the  $i^{\text{th}}$  control connection that is formulated as  $\sum_{i=0}^{n-1} (2^i - 1)$ .

Now, the  $k_i$  indicates to the collective number of  $k$ -CNOT gates that are containing  $i^{\text{th}}$  number of control connection(s), which can be expressed as  $\sum_{i=0}^{n-1} k_i(2^i - 1)$ .

Hence, the generalized formula  $\sum_{i=0}^{n-1} k_i(2^i - 1)$  is capable of evaluating the total number of MixCFFs in a given mixed-control based  $k$ -CNOT circuit.

#### 4. PROPOSED METHODOLOGY

This Section discusses the proposed method to determine the MixCFFs in the  $k$ -CNOT circuits. According to the MixCFF, we considered mixed controlled  $k$ -CNOT circuits presenting both positive and negative control connections. Here, we introduce the fault detection logic to activate the MixCFF that is applied to the proposed ATPG algorithm to obtain the complete test set. Finally, the correlation between the MixCFF model and the existing fault models in reversible circuits is established. The following definitions are presented that are used in the complete test generation process.

**Definition 4.1:** A test pattern  $T = \langle b_1 b_2 \dots b_n \rangle$  is generated by the fault detection logic of MixCFF to enable the fault(s) for each gate of the circuit using the backpropagation of the gate operation. Here, each bit  $b_i \in T$  represents the  $i^{\text{th}}$  input line of the circuit, where  $b_i \in \{0, 1\}$ ,  $n$  is the total number of input lines for  $1 \leq i \leq n$  of the circuit.

**Definition 4.2:** The test set  $TS_{\text{mcf}}$  is the complete test set for detecting all the MixCFFs (both m-SCFF and m-MCFF) in a given mixed-controlled based  $k$ -CNOT circuit. Here, the test set  $TS_{\text{mcf}} = TV_1, TV_2, \dots, TV_l$ , for  $1 \leq j \leq l$ . Each test vector  $TV_j = \langle b_{1j} b_{2j} \dots b_{nj} \rangle$ , where  $b_{ij} \in \{0, 1\}$  represents the  $i^{\text{th}}$  bit position of the  $n$  number of input lines associated with the  $j^{\text{th}}$  test vector  $TV_j \in TS_{\text{mcf}}$ .

##### 4.1 Fault Detection Logic for MixCFF

The main objective of the fault detection logic for MixCFF is to activate each mixed-polarity based  $k$ -CNOT gate to evaluate the faults therein. As per our previous discussion, the MixCFF occurs when the circuit contains positive ( $\bullet$ ) and negative ( $\circ$ ) control connections

**Table 2: Evaluation of total number of MixCFFs in various benchmark reversible circuits**

Benchmark circuit	No. of input lines ( $n$ )	No. of $m$ -SCFFs	No. of $m$ -MCFFs	No. of positive control connections are flipping	No. of negative control connections are flipping	Total no. of MixCFFs
2of5d4-n7-gc12-qc31	7	17	5	4	18	22
2of5d5-n7-gc11-qc32	7	17	5	5	17	22
2of5d6-n6-gc10-qc118	6	25	8	15	18	33
2of5d7-n6-gc9-qc268	6	28	49	42	35	77
mod 4mod5-n5-gc4-qc13	5	6	1	4	3	7
mod 5mod5-n6-gc7-qc429	6	35	68	57	46	103
rd53d1-n7-gc11-qc96	6	24	13	6	31	37
rd53d1-n7-gc12-qc82	7	24	23	1	46	47
symmetric 6-n7-gc14-qc1308	7	66	258	162	162	324
symmetric 6-n7-gc15-qc825	7	70	190	113	147	260

in  $k$ -CNOT gates. The test pattern  $T$  (as mentioned in Definition 1) is generated using the fault detection logic for MixCFF in the mixed-controlled based  $k$ -CNOT circuit. The fault detection logic for MixCFF is applied for generating the test pattern  $T = \langle b_1 b_2 \dots b_n \rangle$ . The generated  $T = \langle b_1 b_2 \dots b_n \rangle$  for each gate is performed the backpropagation to extract the test vector  $TV_j$  at the initial input level of the circuit. The fault detection logic for MixCFF is that the positive control ( $\bullet$ ) connection is assigned by the logic value 1, the negative control ( $\circ$ ) connection is assigned by the logic value 0, and the target ( $\oplus$ ) connection and unconnected connection are assigned by don't-care ( $\times$ ). In our proposed method, we consider the logic value 1 for the don't-care ( $\times$ ) assignment. The following definition is applied to the fault detection logic for MixCFF in our proposed method.

**Definition 4.3:** The mixed-controlled based  $k$ -CNOT circuit consists of  $N$  number of gates, where each gate  $g_k$  (for  $1 \leq k \leq N$ ) comprises of positive controlled ( $\bullet$ ) connection(s), negative controlled ( $\circ$ ) connection(s), unconnected connection(s) and one target connection ( $\oplus$ ), which are associated with individual  $n$  number of input lines.

The notation  $I_i \sim C^{g_k}$  indicates that the positive control ( $\bullet$ ) connection of the  $k^{\text{th}}$  gate is connected to the input line  $I_i$ . Similarly, the notations  $I_j \sim NC^{g_k}$ ,  $I_k \sim U^{g_k}$ ,  $I_l \sim t^{g_k}$  are used to describe the negative control ( $\circ$ ) connection, unconnected connection and target connection ( $\oplus$ ) of the  $k^{\text{th}}$  gate, which are connected to the input lines  $I_j$ ,  $I_k$  and  $I_l$ , respectively.

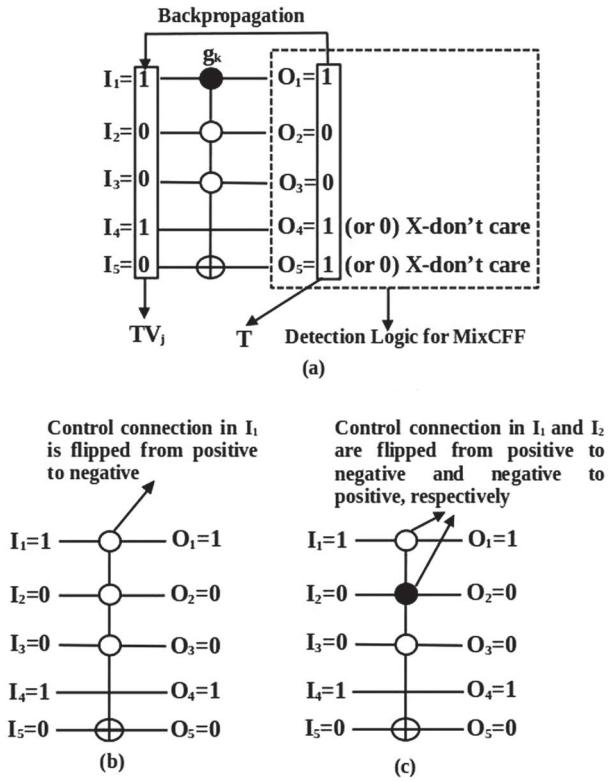
Suppose, the structure of the gate  $g_k$  defined as  $I_1 \sim C^{g_k}$ ,  $I_2 \sim C^{g_k}$ , ...,  $I_j \sim C^{g_k}$ , ...,  $I_k \sim NC^{g_k}$ , ...,  $I_{n-1} \sim U^{g_k}$ ,  $I_n \sim t^{g_k}$ , as per the Definition 3. Now, the fault detection logic for MixCFF is applied to input lines  $I_1$ ,  $I_2$ , ...,  $I_j$ , ...,  $I_k$ , ...,  $I_{n-1}$ ,  $I_n$  to obtain the test pattern  $T = \langle b_1 b_2 \dots b_j \dots b_k \dots b_{n-1} b_n \rangle = \langle 11 \dots 1 \dots 0 \dots 11 \dots \rangle$ , where  $\langle b_1 b_2 \dots b_j \rangle \in \{1\}$ ,  $\langle b_k \rangle \in \{0\}$ ,

$\langle b_{n-1} b_n \rangle \in \{1\}$ . The fault detection logic for MixCFF is illustrated with the help of an Example 3.

**Example 4.1:** The fault detection logic for MixCFF is applied to the mixed-polarity based  $k$ -CNOT gate  $g_k$ , as shown in Figure 7 (a). The structure of the  $g_k$  gate represents all the control connection(s), negative connection(s), unconnected connection(s) and the target connection. Figure 7 (a) shows the all the connections of the gate  $g_k$ , which is defined as  $I_1 \sim C^{g_k}$ ,  $I_2 \sim NC^{g_k}$ ,  $I_3 \sim NC^{g_k}$ ,  $I_4 \sim U^{g_k}$ ,  $I_5 \sim t^{g_k}$ . Next, we apply the fault detection logic to the input lines  $I_1$ ,  $I_2$ ,  $I_3$ ,  $I_4$ ,  $I_5$  that is based on the structure of the gate  $g_k$  for obtaining the test pattern  $T = \langle b_1 b_2 b_3 b_4 b_5 \rangle$ , where  $\langle b_1 \rangle \in \{1\}$ ,  $\langle b_2 b_3 \rangle \in \{0\}$ ,  $\langle b_4 b_5 \rangle \in \{1\}$ . After applying the fault detection logic, the test pattern  $T = \langle 10011 \rangle$  is generated. Now, the generated test pattern  $T = \langle 10011 \rangle$  back propagates toward the input side to retrieve the initial input test vector  $TV_j = \langle 10010 \rangle$  of gate  $g_k$ , as illustrated in Figure 7 (a). However, the same test vector  $TV_i = \langle 10010 \rangle$  that extracts from the  $T = \langle 10011 \rangle$  is applied to the faulty  $k$ -CNOT gate, where an  $m$ -SCFF occurs at the input line  $I_1$  due to a flipping from  $I_1 \sim C^{g_k}$  to  $I_1 \sim NC^{g_k}$ , as shown in Figure 7 (b). It is observed that the faulty  $k$ -CNOT gate generates the faulty output vector  $\langle 10010 \rangle$  instead of the output vector  $\langle 10011 \rangle$  of fault-free  $k$ -CNOT gate. Figure 7 (c) shows the effect of an  $m$ -MCFF that occurs at input line  $I_1$  and  $I_2$  due to the flipping from  $I_1 \sim C^{g_k}$  to  $I_1 \sim NC^{g_k}$  and  $I_2 \sim NC^{g_k}$  to  $I_2 \sim C^{g_k}$ , respectively. Due to the occurrence of the  $m$ -MCFF, the faulty output vector  $\langle 10010 \rangle$  is generated by applying the same input test vector  $TV_i = \langle 10010 \rangle$ . Therefore, the generated  $T = \langle 10011 \rangle$  by the rule of fault detection logic for MixCFF can activate the faulty mixed-polarity based  $k$ -CNOT gate.

#### 4.2 Generation of Complete Test Set $TS_{mcf}$ for MixCFF

This section describes the generation of the complete test set  $TS_{mcf}$  for the MixCFF in the mixed controlled based



**Figure 7:** (a) Fault detection logic for MixCFF (b) Occurrence of an  $m$ -SCFF (c) Occurrence of an  $m$ -MCFF

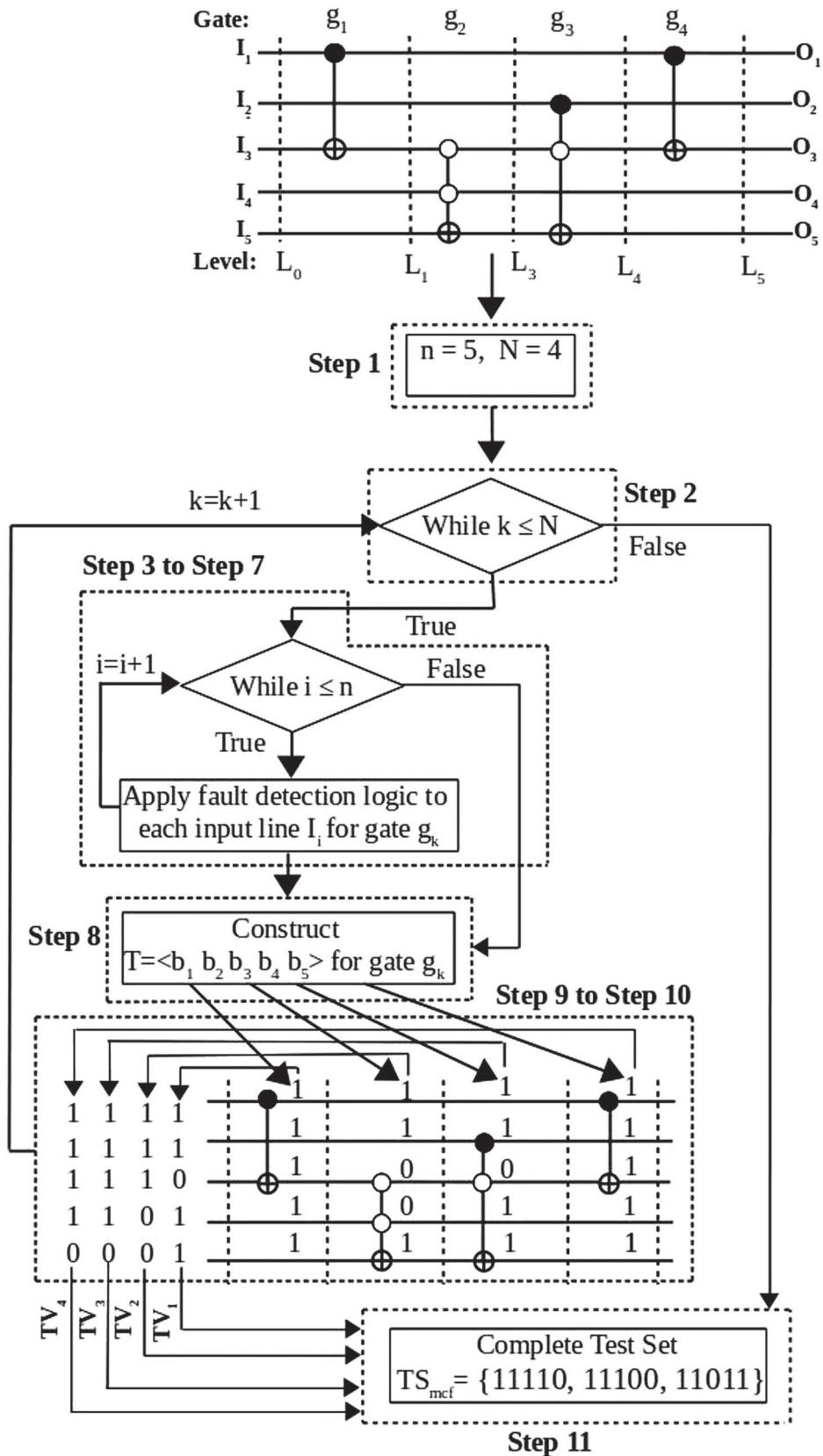
$k$ -CNOT circuits. Here, we consider the constant input and garbage output lines are assumed to be normal input and output lines, respectively, for testing purposes only. The construction of complete test set  $TS_{mcf}$  for MixCFF is presented in Algorithm 1.

**Example 4.2:** The mod\_4mod5\_n5\_gc4\_qc13 benchmark circuit is considered to explain the process flow of Algorithm 1 for generating the complete test set  $TS_{mcf}$ , which is shown in Figure 8. In Step 1, identifies the number of input lines  $n$  and gates  $N$  from a given reversible circuit. In mod\_4mod5\_n5\_gc4\_qc13 circuit,  $n = 5$  and  $N = 4$ . In Step 2, each gate  $g_1, g_2, g_3$  and  $g_4$  are considered for generating the complete test set  $TS_{mcf}$ . In Step 3 to Step 7, we are scanning all the input lines  $I_1, I_2, I_3, I_4$  and  $I_5$  of each gate  $g_k$  (for  $1 \leq k \leq 4$ ) for applying the fault detection logic. The fault detection logic for MixCFF ensures that if the input line  $I_i$  of each gate  $g_k$  is connected to the positive control connection (i.e.,  $I_i \sim C^{g_k}$ ), or unconnected control connection (i.e.  $I_i \sim U^{g_k}$ ), or target connection (i.e.  $I_i \sim t^{g_k}$ ), then  $I_i$  is assigned by the bit  $(b_i) = 1$ , otherwise, for the negative control connection (i.e.,  $I_i \sim NC^{g_k}$ ),  $I_i$  is assigned by the bit  $(b_i) = 0$ . In this circuit,  $g_1: I_1 \sim C^{g_1}, I_2 \sim U^{g_1}, I_3 \sim t^{g_1}, I_4 \sim U^{g_1}, I_5 \sim U^{g_1}$ ,  $g_2: I_1 \sim U^{g_2}, I_2 \sim U^{g_2}, I_3 \sim NC^{g_2}, I_4 \sim NC^{g_2}, I_5 \sim t^{g_2}$ ,  $g_3: I_1 \sim U^{g_3}, I_2 \sim C^{g_3}, I_3 \sim NC^{g_3}, I_4 \sim U^{g_3}, I_5 \sim U^{g_3}$ ,  $g_4: I_1 \sim t^{g_4}, I_2 \sim U^{g_4}, I_3 \sim U^{g_4}, I_4 \sim U^{g_4}, I_5 \sim U^{g_4}$ .

$\sim t^{g_3}$  and  $g_4: I_1 \sim C^{g_4}, I_2 \sim U^{g_4}, I_3 \sim t^{g_4}, I_4 \sim U^{g_5}, I_5 \sim U^{g_5}$ . In Step 8, the test pattern  $T = <11111>$ ,  $T = <11001>$ ,  $T = <11011>$  and  $T = <11111>$  for the gate  $g_1, g_2, g_3$  and  $g_4$ , respectively. In Step 9 to Step 10, the generated test pattern  $T = <11111>$  is applied to the gate  $g_4$  and back propagate to the initial input level  $L_0$  of the circuit to extract the corresponding test vector  $TV_4 = <11110>$ , as shown in Figure 8. Similarly, the test pattern  $T = <11011>$ ,  $T = <11001>$  and  $T = <11111>$  are applied to the gate  $g_3, g_2$  and  $g_1$ , respectively, and obtained the corresponding test vector  $TV_3 = <11110>$ ,  $TV_2 = <11100>$  and  $TV_1 = <11011>$  at the input level  $L_0$  of the circuit using the back propagation. In Step 11, remove the duplicate test vector and construct the complete test set  $TS_{mcf} = 11110, 11100, 11011$  for the mod\_4mod5\_n5\_gc4\_qc13 circuit.

**Lemma 4.1:** The test set  $TS_{mcf}$  is capable of detecting all the MixCFFs in a given mixed-controlled based  $k$ -CNOT reversible circuit.

*Proof.* Let us consider the mixed-controlled based circuit comprises of  $g_1, g_2, \dots, g_N$ , where  $g_k$  (for  $1 \leq k \leq N$ ) formulates with  $I_1 \sim C^{g_k}, I_2 \sim C^{g_k}, \dots, I_j \sim C^{g_k}, \dots, I_k \sim NC^{g_k}, \dots, I_{n-1} \sim U^{g_k}, I_n \sim t^{g_k}$ . The fault detection logic for MixCFF is applied to input lines  $I_1, I_2, \dots, I_j, \dots, I_k, \dots, I_{n-1}, I_n$  to obtain the test pattern  $T = <b_1 b_2 \dots b_j \dots b_k \dots b_{n-1} b_n>$ , where  $b_k = 0$  that is associated with input line  $I_k$ . At first, we consider that the gate  $g_k$  is not affected by the MixCFF, i.e., no fault (s) is occurred in the  $g_k$  gate. Now, the  $T = <b_1 b_2 \dots b_j \dots b_k \dots b_{n-1} b_n>$  is applied to the gate  $g_k$  and the input test vector  $TV_k = <b_1 b_2 \dots b_j \dots b_k \dots b_{n-1} (b_1 b_2 \dots b_j \dots b'_k \dots \oplus b_n)> = <b_1 b_2 \dots b_j \dots b_k \dots b_{n-1} (1 \oplus b_n)> = <b_1 b_2 \dots b_j \dots b_k \dots b_{n-1} b'_n>$  is generated at the input level of the gate  $g_k$  with the help of back propagation. Thus, the input test vector  $TV_k = <b_1 b_2 \dots b_j \dots b_k \dots b_{n-1} b'_n>$  generates the output vector  $<b_1 b_2 \dots b_j \dots b_k \dots b_n>$  of the fault-free gate  $g_k$ . In contrast, we assume that the MixCFF occurs in gate  $g_k$  (i.e., an  $m$ -SCFF or  $m$ -MCFF) at line  $I_k$ , i.e.  $I_k$  is flipped negative to positive control connection. If we apply the same test vector  $TV_k = <b_1 b_2 \dots b_j \dots b_k \dots b_{n-1} b'_n>$  to the faulty gate  $g_k$ , the output vector  $<b_1 b_2 \dots b_j \dots b_k \dots b_{n-1} (b_1 b_2 \dots b_j \dots b_k \dots b_{n-1} \oplus b'_n)> = <b_1 b_2 \dots b_j \dots b_k \dots b_{n-1} (0 \oplus b'_n)> = <b_1 b_2 \dots b_j \dots b_k \dots b_{n-1} b'_n>$ . It is observed that the faulty gate  $g_k$  produces the output test vector  $<b_1 b_2 \dots b_j \dots b_k \dots b_{n-1} b'_n>$  instead of the output vector  $<b_1 b_2 \dots b_j \dots b_k \dots b_n>$  of the fault-free gate  $g_k$ . The same



**Figure 8:** Demonstration of Algorithm 1 for MixCFF with the help of *mod\_4mod5\_n5\_gc4\_qc13* circuit.

test vector  $TV_k$  generates the different outputs at the primary-gate level for the fault-free and faulty condition, which is established by the controllability property [19] of the reversible circuit. Hence, the test set  $TS_{mcf}$  is the complete test set for the MixCFFs in a given mixed-controlled based  $k$ -CNOT circuit.

---

**Algorithm 1:** Complete Test set TSMCF Generation for MMGFs

---

**Input:** Mixed-controlled based  $k$ -CNOT reversible Circuit

$n$ : Total number of input lines in a given circuit.

$N$ : Total number of gates in a given circuit.

$g_k$ :  $k^{\text{th}}$  number of gate.

$l_i$ :  $i^{\text{th}}$  number of input lines.

$TV_k$ :  $k^{\text{th}}$  number of input test vector of  $g_k$  gate.

$L_0$ : Initial input level of the circuit.

$C^{g_k}$ : Positive control connection in  $g_k$  gate.

$NC^{g_k}$ : Negative control connection in  $g_k$  gate.

$U^{g_k}$ : Unconnected connection in  $g_k$  gate.

$t^{g_k}$ : Target control connection in  $g_k$  gate.

**Output:** Complete test set  $TS_{mcf}$  for all MixCFFs

1 Identify the parameters  $n$  and  $N$  from the circuit

2 **for**  $k \leftarrow 1$  to  $N$  **do**

3   **for**  $i \leftarrow 1$  to  $n$  **do**

4     **if** Input line  $l_i$  is connected  $C^{g_k}$  or  $U^{g_k}$  or  $t^{g_k}$  **then**

5        $(b_i) \leftarrow$  logic value 1

6     **if** Input line  $l_i$  is connected  $NC^{g_k}$  **then**

7        $(b_i) \leftarrow$  logic value 0

8     Construct test pattern  $T = (b_1 b_2 \dots b_n)$  for each gate  $g_k$ .

9     Back propagate  $T$  for  $g_k$  to extract the test vector  $TV_k$  at level  $L_0$ .

10     $TS_{mcf} \leftarrow TV_k$

11   Generate complete test set  $TS_{mcf}$

---

the unconnected and the target connections for activating the SMGF of a particular gate. In our proposed work, the fault detection logic for MixCFF is the same for an SMGF in the mixed controlled-based  $k$ -CNOT circuits. Thus, the test set  $TS_{mcf}$  for MixCFF is also a complete test set for SMGF in a given circuit.

#### 4.3.2 MixCFF and MMGF

An MMGF occurs when two or more consecutive numbers of  $k$ -CNOT gates disappear [5,22]. Based on the property of MMGF, the SMGFs are a subset of the MMGFs [5]. However, it is optional that the complete test set for SMGF covers all the MMGFs [5]. We know that the detection logic of MixCFF covers the SMGF in the mixed-controlled  $k$ -CNOT circuits. This analogy indicates that some test vectors  $TV_i$  in  $TS_{mcf}$  have the capability to satisfy the fault detection criteria for MMGF. There is a possibility that all the MMGFs are not covered by the  $TS_{mcf}$ .

#### 4.3.3 MixCFF and PMGF

If any number of gate control disappears from the  $k$ -CNOT gate, it is referred to as a PMGF [5]. In the mixed control-based  $k$ -CNOT circuit, the logic value 0 and 1 are allocated to the missing positive and negative control connections, respectively; the unmissed positive and negative control connections are assigned by the logic value 1 and 0, respectively, and unconnected and target connections are allocated by the don't-care condition to activate the PMGF fault in each  $k$ -CNOT gate. However, the detection logic for the MixCFF in our proposed work does not match the fault detection logic of the PMGF for the missing control connections (positive or negative). Therefore, the test vector  $TV_i$  in  $TS_{mcf}$  for the MixCFF satisfies the fault detection logic for PMGF in some of the  $k$ -CNOT gates. As a result, the test set  $TS_{mcf}$  has not achieved 100% fault coverage in PMGF.

#### 4.3.4 MixCFF and RGF

An RGF occurs when an unwanted occurrence of a gate by several instantiations of the same gate [5,26,41]. The authors in [5] provided the physical justification for an RGF is the occurrence of long or duplicated pulses. The properties of the RGF [5] ensure that the gate is replaced by duplicate instances of  $k$  of the same gate, where the occurrence of  $k$  is even or odd. If the occurrence of instances  $k$  of the same gate is in even numbers, the effect of both RGF and SMGF is similar in terms of test set generation for detecting the faults. Thus, the generated  $TS_{mcf}$  for MixCFF that is applied for determining the fault coverage of SMGFs in our proposed work is sufficient for detecting all the detectable RGFs. In another way, the occurrence of instances  $k$  of the same gate is in

### 4.3 Correlation of MixCFF with the Existing Fault Models in Reversible Circuits

In circuit testing, the correlation of one particular fault model to the existing fault models is essential to evaluate the fault detection performance efficiency. Moreover, it is good practice to apply the generated complete test set for one particular fault model to the other fault models for checking the fault coverage range. The fault coverage range is defined by the ratio of the total number of faults detected and the total number of faults that appear in the circuit [17]. In this section, we correlate the MixCFF fault model with existing fault models in reversible circuits, like SMGF, MMGF and PMGF fault model to establish the correlation and determine the fault coverage range. The evaluation results of the fault coverage range are presented in Section 5.

#### 4.3.1 MixCFF and SMGF

The complete removal or disappearance of the single reversible gate from the circuit causes the SMGF [5]. The fault detection logic of the mixed controlled based  $k$ -CNOT circuit is that the logic value 1 is allocated to the positive control connection, logic value 0 is allocated to the negative control connection and the arbitrary logic value 0 or 1 (*i.e.* don't-care condition) are assigned to

**Table 3: Generation of complete test set  $TS_{mcf}$  for detecting the MixCFFs with CPU time (sec)**

Benchmark circuit	$n$	$N$	Total no. of faults	Total no. of faults	Total no. of faults	Size of $TS_{mcf}$	CPU time (sec)
			$m$ -SCFFs	$m$ -MCFFs	$(m$ -SCFFs + $m$ -MCFFs)		
mod 4mod5-n5-gc4-qc13	5	4	6	2	8	3	0.01562
mod 5mod5-n6-gc7-qc429	6	7	35	182	217	7	0.03124
2of5d7-n6-gc9-qc268	6	9	28	109	137	8	0.03652
2of5d6-n6-gc10-qc118	6	10	25	39	64	9	0.03125
2of5d5-n7-gc11-qc32	7	11	17	6	23	10	0.4687
2of5dn4-n7-gc12-qc31	7	12	17	6	23	9	0.05088
rd53d1-n7-gc11-qc96	7	11	24	13	37	11	0.04887
rd53d1-n7-gc12-qc82	7	12	48	6	54	12	0.05338
symmetric6-n7-gc14-qc1308	7	14	70	592	662	14	0.07813
symmetric6-n7-gc15-qc825	7	15	85	892	947	13	0.06250
symmetric6-n7-gc41-qc206	7	41	101	192	293	14	0.1022
nth_prime07_inc-n7-gc440-qc10989	7	440	1329	10399	11728	116	1.0487
nth_prime07_inc-n7-gc489-qc10898	7	489	1378	10437	11815	104	1.0416
nth_prime07_inc-n7-gc839-qc2284	7	839	1898	2043	3941	118	1.5397
nth_prime08_inc-n8-gc1692-qc6339	8	1692	3968	6092	10060	222	6.4840
rd73d2-n9-gc835-qc4069	9	835	2068	5081	7149	225	6.7764
rd73d2-n9-gc296-qc3421	9	296	1277	44195	45472	99	3.3957
symmetric9-n10-gc347-qc1975	10	347	875	2408	3283	131	5.9495
symmetric9-n10-gc73-qc61928	10	73	735	61756	62491	65	2.8694
symmetric9-n10-gc74-qc31819	10	74	682	45384	46066	60	2.8712
rd84d1-n11-gc2560-qc12397	11	2560	6033	17757	23790	550	85.517
rd84d1-n11-gc679-qc359384	11	679	3416	376108	379524	226	34.781

odd numbers; the functionality effect of the RGF behaves as the functionality of the fault-free circuit. Therefore, the test set is not required for the odd number of duplicate instances  $k$  of the same gate in RGF.

## 5. EXPERIMENTAL RESULTS

The implementation of the proposed method for evaluating the MixCFF and determining the fault coverage range with already established fault models in reversible circuits are presented in this section. Various benchmark circuits [40] are considered to perform the experimental results. Here, each benchmark circuit is converted to the machine-readable version in *tfc* file format and the structure of the *tfc* file format is explored in [31]. The proposed ATPG algorithm is effectuated in Python 3.4, and system configurations consist of an Intel Pentium (R) Core-i5 machine with CPU-2020 @ 2.40 GHz  $\times$  2 system, 8GB RAM on Window 10 (64-bit).

The experimental results for generating the complete test set  $TS_{mcf}$  for the detection of MixCFFs in various mixed-controlled based  $k$ -CNOT circuits are presented in Table 3. The first six columns of Table 3 are represented the mixed-controlled based  $k$ -CNOT benchmark circuits, the total number of input lines ( $n$ ), the total number of gates ( $N$ ), the total number of  $m$ -SCFFs, the total number of  $m$ -MCFFs and the total number of MixCFFs, *i.e.* for both  $m$ -SCFFs and  $m$ -MCFFs, respectively. Column 7 provides the total number of test vectors in  $TS_{mcf}$  that is required to detect the MixCFFs. The last column 8 indicates the CPU time in seconds for completing the complete test set  $TS_{mcf}$  generation process

for the MixCFF of the corresponding circuit. Based on the experimental results as reported in Table 3, it is observed that the size of the complete test set  $TS_{mcf}$  increases when the total number of MixCFFs grows. For the circuits, *2of5d4-n7-gc12-qc31* and *rd53d1-n7-gc12-qc82*, the number of inputs ( $n$ ) and the number of gates ( $N$ ) are 7 and 12, respectively. However, the total number of MixCFFs (both  $p$ -SCFF and  $p$ -MCFF) is 23 and 54 for the circuits *2of5d4-n7-gc12-qc31* and *rd53d1-n7-gc12-qc82*, respectively. Thus, the size of the  $TS_{mcf}$  for the circuit *rd53d1-n7-gc12-qc82* is 12 that is larger as compared to the  $TS_{mcf}$  size (*i.e.* 9) for the circuit *2of5d4-n7-gc12-qc31*. In addition, some specific circuits have higher faults even though the total number of test vectors in  $TS_{mcf}$  is smaller. This is because there may be a higher number of gates in that particular circuit. For example, as presented in Table 3, the total number of MixCFFs is 11815 for the circuit *nth\_prime07\_inc-n7-gc489-qc10898*, which is relatively larger as compared to the number of MixCFFs for the circuits *nth\_prime07\_inc-n7-gc440-qc10989* (*i.e.* 11728) and *nth\_prime07\_inc-n7-gc839-qc2284* (*i.e.* 3941). But, the size of the  $TS_{mcf}$  is 104 for the circuit *nth\_prime07\_inc-n7-gc489-qc10898* that is less than the size of the  $TS_{mcf}$  for the circuits *nth\_prime07\_inc-n7-gc440-qc10989* (*i.e.* 116) and *nth\_prime07\_inc-n7-gc839-qc2284* (*i.e.* 118). Therefore, the number of gates in a given mixed-controlled based  $k$ -CNOT circuit also significantly impacts the complete test set  $TS_{mcf}$  and the CPU time.

In Table 4, the test set  $TS_{mcf}$  for MixCFF is applied to different existing fault models, like SMGF, MMGF, PMGF and RGF in mixed-control based  $k$ -CNOT circuit for

**Table 4: Experimental results for the fault coverage range of SMGF, MMGF, PMGF and RGF by the  $TS_{mcf}$** 

Benchmark circuit	$n$	$N$	Total no. of faults				Evaluation of fault coverage by $TS_{mcf}$ [Proposed]			
			SMGF	MMGF	PMGF	RGF ( $k = 2$ )	SMGF	MMGF	PMGF	RGF
mod 4mod5-n5-gc4-qc13	5	4	4	6	6	8	100%	100%	33.35%	100%
mod 5mod5-n6-gc7-qc429	6	7	7	21	35	14	100%	100%	0%	100%
2of5d7-n6-gc9-qc268	6	9	9	36	28	18	100%	100%	25%	100%
2of5d6-n6-gc10-qc118	6	10	10	45	25	20	100%	100%	48.32%	100%
2of5d5-n7-gc11-qc32	7	11	11	55	17	22	100%	100%	81.25%	100%
2of5dn4-n7-gc12-qc31	7	12	12	66	17	24	100%	100%	94.11%	100%
rd53d1-n7-gc11-qc96	7	11	11	55	24	22	100%	87.17%	62.55%	100%
rd53d1-n7-gc12-qc82	7	12	12	66	48	24	100%	100%	84%	100%
symmetric6-n7-gc14-qc1308	7	14	14	91	70	28	100%	63.34%	45.07%	100%
symmetric6-n7-gc15-qc825	7	15	15	105	85	30	100%	82.27%	50%	100%
symmetric6-n7-gc41-qc206	7	41	41	820	101	82	100%	56.19%	87.14%	100%
nth prime07 inc-n7-gc440-qc10989	7	440	440	96580	1329	880	100%	84.53%	62.41%	100%
nth prime07 inc-n7-gc489-qc10898	7	489	489	119072	1378	978	100%	92.14%	74.11%	100%
nth prime07 inc-n7-gc839-qc2284	7	839	839	351541	1898	1678	100%	78.77%	61.39%	100%
nth prime08 inc-n8-gc1692-qc6339	8	1692	1692	1430586	3968	3384	100%	100%	76.28%	100%
rd73d2-n9-gc835-qc4069	9	835	835	348195	2068	1670	100%	85.34%	44.09%	100%
rd73d2-n9-gc296-qc43421	9	296	296	43660	1277	592	100%	72.87%	34.16%	100%
symmetric 9-n10-gc347-qc1975	10	347	347	60031	875	694	100%	100%	81.21%	100%
symmetric 9-n10-gc73-qc61928	10	73	73	2628	735	146	100%	87.18%	96.11%	100%
symmetric 9-n10-gc74-qc31819	10	74	74	2701	682	148	100%	69.15%	47.15%	100%
rd84d1-n11-gc2560-qc12397	11	2560	2560	3275520	6033	5120	100%	57.26%	63.19%	100%
rd84d1-n11-gc679-qc359384	11	679	679	230181	3416	1358	100%	86.39%	87.18%	100%
<b>Average</b>							100%	86.48%	60.82%	100%

evaluating the fault coverage range. Here, columns 4, 5, 6, and 7 indicate the total number of faults in SMGF, MMGF, PMGF, and RGF, respectively. To determine the total number of faults in RGFs, we consider detectable RGFs in mixed-control based reversible circuits, where instances  $k$  of the repeated gate occur in even numbers. For this purpose, we consider  $k = 2$  for an RGF, as shown in column 7 of Table 4. The fault coverage for each fault model is determined by the  $TS_{mcf}$  is reported in columns 8, 9, 10, and 11. It is observed that the complete test set  $TS_{mcf}$  for MixCFF covers 100% fault coverage of both SMGF and RGF, as presented in Columns 8 and 11, respectively, due to the same test set exists for the detection of faults in SMGF and RGF. On average, fault coverage is 86.48% for the MMGF, which is higher as compared to the fault coverage on average of 60.82% for the PMGF.

## 6. CONCLUSION

In this article, we have considered the fault model MixCFF in the mixed-controlled based  $k$ -CNOT reversible circuit, which is considered a structural fault model. The detailed structure and occurrence of the MixCFF are presented in this article. An ATPG algorithm is reported to generate the complete test set  $TS_{mcf}$  for identifying all the possible MixCFFs. Moreover, we have presented the correlation of MixCFF with other existing fault models, and the evaluation of the fault coverage range for the existing fault models by the generated complete test set  $TS_{mcf}$  is also addressed. As per the experimental results, we have observed that the developed complete test set

for the MixCFF is effectively used for the other fault models, specifically for the SMGF. There is a possibility to reconstruct the  $TS_{mcf}$  to achieve the complete fault coverage of MMGF and PMGF by the  $TS_{mcf}$ . However, the  $TS_{mcf}$  is not a minimal complete test set in the proposed work; therefore, it can be extended to formulate a minimal complete test set for detecting the MixCFF.

The proposed fault detection logic for the MixCFFs is developed based on the reversibility in classical implementation, *i.e.*, the circuit computation is implemented with classical bits. Therefore, we restrict the proposed fault detection logic for MixCFFs as the work in the paper is for reversible circuits. Hence, our future work includes verifying the performance analysis of the fault-free and faulty circuit's behaviors using the test set  $TS_{mcf}$  for the MixCFF in IBM Quantum hardware. Also, our future work includes a detailed analysis of the probability distribution on the observations of all probabilistic output states in a quantum circuit simulator.

## DISCLOSURE STATEMENT

No potential conflict of interest was reported by the author(s).

## AUTHORSHIP CONTRIBUTION STATEMENT

**Mousum Handique** was involved in designing the proposed algorithm and evaluating and analyzing the results, apart from writing the manuscript and the overall supervision of the work. **Amit Prasad** was involved in the design of the proposed

algorithm and implementation of the proposed algorithm with several benchmark circuits.

## SUPPLEMENTAL DATA

Supplemental data for this article can be accessed online at <https://doi.org/10.1080/03772063.2025.2503340>.

## AVAILABILITY OF DATA AND MATERIAL

Several benchmark circuits are associated with this research, which is considered a dataset to verify and correctness of the proposed work (sources are mentioned in the reference section).

## CODE AVAILABILITY

Computer Program written in Python 3.4 is available with the authors and may be made available on request.

## REFERENCES

1. G. E. Moore, "Gramming more components onto integrated circuits," *Proc. IEEE*, Vol. 86, no. 1, pp. 82–5, **1998**.
2. C. H. Bennett, "Notes on Landauer's principle, reversible computation, and Maxwell's Demon," *Stud. Hist. Philos. Sci. Part B*, Vol. 34, no. 3, pp. 501–10, **2003**.
3. R. Landauer, "Irreversibility and heat generation in the computing process," *IBM J. Res. Dev.*, Vol. 5, no. 3, pp. 183–91, **1961**.
4. C. H. Bennett, "Logical reversibility of computation," *IBM J. Res. Dev.*, Vol. 17, no. 6, pp. 525–32, **Nov. 1973**.
5. I. Polian, T. Fiehn, B. Becker, and J. P. Hayes, "A family of logical fault models for reversible circuits," in 14th Asian Test Symposium (ATS'05), IEEE, 2005, pp. 422–427.
6. M. A. Nielsen, and I. L. Chuang. *Quantum Computation and Quantum Information*. New York, NY: Cambridge University Press, **2011**.
7. R. P. Feynman, "Quantum mechanical computers," *Found. Phys.*, Vol. 16, no. 6, pp. 507–31, **1986**.
8. M. Sarkar, P. Ghosal, and S. P. Mohanty, "Minimal reversible circuit synthesis on a DNA computer," *Nat. Comput.*, Vol. 16, no. 3, pp. 463–72, **2017**.
9. M. Rofail, and A. Younes, "Synthesis strategy of reversible circuits on DNA computers," *Symmetry*, Vol. 13, no. 7, pp. 1242, **2021**.
10. C. Taraphdar, T. Chattopadhyay, and J. N. Roy, "Mach-Zehnder interferometer-based all-optical reversible logic gate," *Opt. Laser Technol.*, Vol. 42, no. 2, pp. 249–59, **2010**.
11. V. V. Shende, A. K. Prasad, I. L. Markov, and J. P. Hayes, "Synthesis of reversible logic circuits," *IEEE Trans. Comput. Aided Des. Integr. Circ. Syst.*, Vol. 22, no. 6, pp. 710–22, **2003**.
12. A. Chattopadhyay, C. Chandak, and K. Chakraborty, "Complexity analysis of reversible logic synthesis," arXiv preprint arXiv:1402.0491, **2014**.
13. P. Gupta, A. Agrawal, and N. K. Jha, "An algorithm for synthesis of reversible logic circuits," *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.*, Vol. 25, no. 11, pp. 2317–30, **2006**.
14. M. Saeedi, M. S. Zamani, M. Sedighi, and Z. Sasanian, "Reversible circuit synthesis using a cycle-based approach," *ACM J. Emerg. Technol. Comp. Syst.*, Vol. 6, no. 4, pp. 1–26, **2010**.
15. M. Arabzadeh, M. Saeedi, and M. S. Zamani, "Rule-based optimization of reversible circuits," in *2010 15th Asia and South Pacific Design Automation Conference (ASP-DAC)*, IEEE, 2010, pp. 849–854.
16. Y. C. Chen, and F. J. Chao, "Optimization of reversible logic networks with gate sharing," in *Proceedings of the 28th Asia and South Pacific Design Automation Conference*, 2023, pp. 166–171.
17. N. K. Jha, and S. Gupta. *Testing of Digital Systems*. Cambridge: Cambridge University Press, **2003**.
18. M. Abramovici, M. A. Melvin, and A. D. Friedman. *Digital Systems Testing and Testable Design*. New York: Computer science press, **1990**.
19. K. N. Patel, J. P. Hayes, and I. L. Markov, "Fault testing for reversible circuits," *IEEE Trans. Comput. Aided Des. Integr. Circ. Syst.*, Vol. 23, no. 8, pp. 1220–30, **2004**.
20. M. Handique, J. K. Deka, and S. Biswas, "Detection of stuck-at and bridging fault in reversible circuits using an augmented circuit," in *2021 IEEE 30th Asian Test Symposium (ATS)*, IEEE, 2021, pp. 55–60.
21. M. Handique, S. Biswas, and J. K. Deka, "Test generation for bridging faults in reversible circuits using path-level expressions," *J. Electron. Test.*, Vol. 35, pp. 441–57, **2019**.
22. J. P. Hayes, I. Polian, and B. Becker, "Testing for missing-gate faults in reversible circuits," in *13th Asian test symposium*, IEEE, 2004, pp. 100–105.
23. J. E. Rice, "An overview of fault models and testing approaches for reversible logic," in *2013 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM)*, IEEE, 2013, pp. 125–130.
24. M. Zamani, M. B. Tahoori, and K. Chakrabarty, "Ping-pong test: Compact test vector generation for reversible circuits," in *2012 IEEE 30th VLSI Test Symposium (VTS)*, IEEE, 2012, pp. 164–169.

25. B. Mondal, D. K. Kole, D. K. Das, and H. Rahaman. “Generator for test set construction of smgf in reversible circuit by boolean difference method,” in 2014 IEEE 23rd Asian test symposium, IEEE, 2014, pp. 68–73.
26. A. N. Nagamani, S. Ashwin, B. Abhishek, and V. K. Agrawal, “An exact approach for complete test set generation of toffoli-fredkin-peres based reversible circuits,” *J. Electron. Test.*, Vol. 32, no. 2, pp. 175–96, 2016.
27. B. Mondal, C. Bandyopadhyay, A. Bhattacharjee, D. Roy, S. Parekh, and H. Rahaman, “An approach for detection and localization of missing gate faults in reversible circuit,” *IETE J. Res.*, Vol. 68, no. 5, pp. 3607–3627, 2022.
28. M. Bubna, N. Goyal, and I. Sengupta. “A DFT methodology for detecting bridging faults in reversible logic circuits,” in *TENCON 2007-2007 IEEE Region 10 Conference*, IEEE, 2007, pp. 1–4.
29. M. Ibrahim, A. R. Chowdhury, and H. M. H. Babu. “Minimization of CTS of k-CNOT circuits for SSF and MSF model,” in 2008 IEEE international symposium on defect and fault tolerance of VLSI systems, IEEE, 2008, pp. 290–298.
30. T. Toffoli. “Reversible computing”, in *International Colloquium on Automata, Languages, and Programming*. Springer, 1980, pp. 632–644.
31. D. Maslov. Reversible logic synthesis benchmarks page (2015). <http://webhome.cs.uvic.ca/dmaslov/>.
32. H. Rahaman, D. K. Kole, D. K. Das, and B. B. Bhattacharya, “Fault diagnosis in reversible circuits under missing-gate fault model,” *Comput. Electr. Eng.*, Vol. 37, no. 4, pp. 475–85, 2011.
33. R. Wille, H. Zhang, and R. Drechsler. “Fault ordering for automatic test pattern generation of reversible circuits,” in 2013 IEEE 43rd International Symposium on Multiple-Valued Logic, IEEE, 2013, pp. 29–34.
34. J. Mondal, D. K. Das, and B. B. Bhattacharya. “Design-for-testability in reversible logic circuits based on bit-swapping,” in 2015 IEEE 24th Asian Test Symposium (ATS), IEEE, 2015, pp. 217–222.
35. B. Mondal, C. Bandyopadhyay, and H. Rahaman. “A testing scheme for mixed-control based reversible circuits,” in 2016 Sixth International Symposium on Embedded Computing and System Design (ISED), IEEE, 2016, pp. 96–100.
36. A. P. Surhonne, A. Chattopadhyay, and R. Wille. “Automatic test pattern generation for multiple missing gate faults in reversible circuits: work in progress report,” in *Reversible Computation: 9th International Conference, RC 2017, Proceedings* 9, Springer International Publishing, 2017, pp. 176–182.
37. A. N. Nagamani, S. N. Anuktha, N. Nanditha, and V. K. Agrawal, “A genetic algorithm based heuristic method for test set generation in reversible circuits,” *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.*, Vol. 37, no. 2, pp. 324–36, 2018.
38. J. Mondal, A. Deb, and D. K. Das, “An efficient design for testability approach of reversible logic circuits,” *J. Circuits Syst. Comput.*, Vol. 30, no. 6, pp. 2150094, 2021.
39. M. Handique, and H. K. Sarma, “Testable design for positive control flipping faults in reversible circuits,” *Indon. J. Electr. Eng. Inform.*, Vol. 11, no. 2, pp. 416–30, 2023.
40. R. Wille, D. Große, L. Teuber, G. W. Dueck, and R. Drechsler. “Revlib: An online resource for reversible functions and reversible circuits,” in: 38th International Symposium on Multiple Valued Logic (ismvl 2008), IEEE, 2008, pp. 220–225.
41. H. Rahaman, D. K. Kole, D. K. Das, and B. B. Bhattacharya. “On the detection of missing-gate faults in reversible circuits by a universal test set,” in *21st International conference on VLSI design (VLSID 2008)*, 2008, pp. 163–168.
42. R. Wille, and R. Drechsler. *Towards a design flow for reversible logic*. Springer Dordrecht Heidelberg London New York: Springer Science & Business Media, 2010.
43. A. Leporati, C. Zandron, and G. Mauri. “Simulating the fredkin gate with energy-based p systems,” *Proceedings of the Second Brainstorming Week on Membrane Computing, Sevilla, ETS de Ingenieria Informática*, 2–7 December, Febrero. 2004, pp. 292–308.
44. H. M. Gaur, A. K. Singh, and U. Ghanekar, “A new dft methodology for k-cnot reversible circuits and its implementation using quantumdot cellular automata,” *Optik*, Vol. 127, no. 22, pp. 10593–601, 2016.
45. B. Mondal, P. Das, P. Sarkar, and S. Chakraborty, “A comprehensive fault diagnosis technique for reversible logic circuits,” *Comput. Electr. Eng.*, Vol. 40, no. 7, pp. 2259–72, 2014.
46. J. Mondal, and D. K. Das, “A new online testing technique for reversible circuits,” *IET Quant. Commun.*, Vol. 3, no. 1, pp. 50–9, 2022.
47. W. D. Pan, and M. Nalasani, “Reversible logic,” *IEEE Potent.*, Vol. 24, no. 1, pp. 38–41, 2005.
48. M. Handique, J. K. Deka, and S. Biswas, “Fault localization scheme for missing gate faults in reversible circuits,” *ACM Trans. Design Autom. Electr. Syst.*, Vol. 27, no. 4, pp. 1–29, 2022.

---

## AUTHORS



**Mousum Handique** received his bachelor of engineering (BE) in computer technology from Nagpur University in 2002 and a master of technology (Mtech) degree from Tezpur University, Tezpur, Assam, in 2005. He received his PhD degree in 2020 in the field of VLSI testing and reversible computing from the India Institute of Technology (IITG), Guwahati-781039, Assam, India. He was working as a lecturer in Sikkim Manipal Institute of Technology, Majitar, Sikkim, India (from September 2005 to November 2007). He is currently working as an associate professor at CSE, TSSOT, Assam University, Silchar, Assam, India. His current research interests include testing and synthesis of reversible and quantum circuits, formal system verification, machine intelligence, workflow automation, and queueing theory.

**Corresponding author. Email:** mousum.smit@gmail.com

---



**Amrit Prasad** holds a Btech degree in computer science and engineering from Assam University, Silchar, graduating in 2018. With over 4 years of professional experience, he is currently working as a software developer at Rapawalk Fashion Technologies. His expertise spans web development, data analytics, artificial intelligence, and machine learning, reflecting his passion for leveraging technology to solve complex problems and innovate in the digital space.

**Email:** amritpra94@gmail.com