



# An Efficient Test Set Construction Scheme for Multiple Missing-Gate Faults in Reversible Circuits

Mousum Handique<sup>1</sup> · Jantindra Kumar Deka<sup>1</sup> · Santosh Biswas<sup>2</sup>

Received: 28 July 2019 / Accepted: 12 January 2020 / Published online: 4 February 2020  
© Springer Science+Business Media, LLC, part of Springer Nature 2020

## Abstract

Several fault models are introduced for efficiently identifying the faults in the reversible circuits, where some of the fault models are borrowed from the conventional circuits. In this work, we consider the Missing Gate Fault Model (MGF) which is specifically used for reversible circuits. The proposed work provides a scheme for generating the complete test set for detecting the single and any number of consecutive multiple missing gate faults in  $k$ -CNOT based reversible circuits. The complete test set generation method is twofold. First, a local test pattern is applied to each level of  $k$ -CNOT gates and the reverse simulation method is used for identifying all the possible Single Missing Gate Faults (SMGFs). Second, using the complete test set for SMGFs and based on the structure of the  $k$ -CNOT based circuit, a test set is formulated. The generated test set is capable of detecting all the MMGFs and as well as the SMGFs in reversible circuits. However, the generated complete test set is not minimal. For achieving the minimality, a table is constructed covering row and column faults and an integer linear programming (ILP) problem is formulated to achieve the minimality of the test set. The experimental results demonstrate that the size of the generated minimized test set is smaller or similar as compared to the existing methods and attains 100% fault coverage.

**Keywords** Reversible logic · Single missing gate fault · Multiple missing gate fault · Complete test set · Minimal complete test set

## 1 Introduction

In conventional logic, the operations are logically irreversible, i.e., the input of logic circuits can not be determined uniquely from the output. This is because of loss of information during logic computation. On the other hand, the reversible logic performs the lossless computation and due to that the input can be uniquely derived from the given

output. In 1961, Rolf Landauer [2] showed that if the computation is irreversible, then the energy dissipated is  $kT \ln 2$  Joules due to the loss of every bit of information, where  $k$  is the Boltzmann constant and  $T$  is the temperature of the system in Kelvins. According to Moore's law [1], the number of transistors doubles in every 18 months and circuits are becoming larger. As a result, today's technologies have more power dissipation due to information loss and with that heat generation is becoming severe. It has also been postulated that the information is lossless if the operations are performed in a reversible way [3]. It has motivated the researchers to explore reversible logic as a circuit design alternative.

In every computing device, there may be a possibility that an incorrect state is reached during the computation. This incorrect state can be considered as a fault and the occurrence of the fault(s) causes an effect on the functional behavior of a system. The fault models represent the physical description of these faults. Several fault models have been introduced for the reversible circuit such as stuck-at faults [4], bridging faults [5, 6], missing gate faults [7] and crosspoint faults [9]. Some of the fault models are

---

Responsible Editor: B. B. Bhattacharya

✉ Mousum Handique  
mousum@iitg.ac.in

Jantindra Kumar Deka  
jatin@iitg.ac.in

Santosh Biswas  
santoshbiswas402@yahoo.com

<sup>1</sup> Department of CSE, Indian Institute of Technology, Guwahati (IITG), Guwahati, Assam 781039, India

<sup>2</sup> Department of EECS, Indian Institute of Technology, Bhilai, India

common to the conventional logic circuits. More precisely, a fault model is generally used to abstract the effects of physical failures and also helps to simplify the complexity to detect the faults [10]. The detail discussion and connection between different fault models on reversible circuits can be found in [11].

The test patterns generation for a reversible circuit are relatively simple as compared to a conventional circuit because the property of reversibility ensures high controllability and observability [4]. The property of controllability ensures that any test vector produces a unique test pattern at a particular level in the reversible circuit using the reversible gate operation. Due to the property of observability, if a fault occurs at an intermediate level in the reversible circuit, then it also affects the primary output of the circuit. These two properties make the backtracking process simple for reversible circuits, which is considered to be difficult for conventional circuits [12]. The properties of controllability and observability of reversible circuit help to generate the test set to detect faults and we propose methods to generate the minimal test set to detect single missing gate faults and consecutive multiple missing gate faults in such circuits. A test set is a collection of input test vectors to apply in a given reversible circuit to observe and detect the faults. The goal of the test pattern generation process is to generate a set of test vectors to detect all possible faults in a given reversible circuit. The test set is called a minimal complete test set if it contains a minimum number of test vectors to detect all possible faults.

In this work, an ATPG (Automatic Test Pattern Generation) method is proposed to generate the complete test set for detecting the single and consecutive multiple missing gate faults (MMGF) in the reversible circuits. To extract the minimal complete test set an ILP (Integer Linear Programming) formulation is used. The proposed method is designed for the  $k$ -CNOT based reversible circuit structure and it gives the minimal complete test set with nearly 100% fault coverage. Moreover, the correlation with other fault models like stuck-at faults, appearance crosspoint faults and partial missing gate faults to the MMGF model is also established.

The main contributions of this work are as follows:

1. ATPG algorithms are proposed to generate the complete test set for SMGFs and MMGFs, and extract the minimal complete test set for detecting these faults in the reversible circuits.
2. Correlate the fault coverage range with other fault models such as stuck-at faults, partial missing gate faults, and appearance crosspoint faults.

The rest of this paper is organized as follows: Section 2 provides a general discussion on reversible logic function and reversible gates,  $k$ -CNOT based reversible logic

circuits, and an overview of the relevant fault models. The related works on fault detection in reversible circuits are also discussed in this section. Section 3 describes the proposed ATPG algorithms for generating the complete and minimal test set to detect the SMGFs and MMGFs in reversible circuits. The comparison with other fault models is also included in this section. The experimental results of proposed test set generation method and comparison with other fault models are reported in Section 4. Finally, concluding remarks are presented in Section 5 with some possible directions for future works.

## 2 Background

### 2.1 Reversible Logic Function and Gates

A logic function  $f(x_1, x_2, \dots, x_n)$  of  $n$  Boolean variables is called a reversible function if it realizes bijective functions, i.e.,  $f: \mathbb{B}^n \Rightarrow \mathbb{B}^m$  that maps each possible input pattern to a unique output pattern, where the number of input patterns should be equal to a number of output patterns (i.e.,  $n = m$ ). The bijective function allows a permutation on the set of input patterns to produce an output pattern such that each possible input pattern can uniquely determine an output pattern [13]. Also, an output pattern can uniquely restore the input pattern.

A logic gate is reversible if it realizes a reversible logic function. A necessary condition of the reversible gate is that it has same number of inputs and outputs and every single input gives a unique output. The formulation of a reversible gate consists of  $k$ -input and  $k$ -output wires called as a reversible  $k \times k$  gate [4]. Therefore, the reversible  $k \times k$  gate has the capability to reconstruct the input states from the output states. For the construction of the reversible circuit, several gates have been proposed over the past decades. Some of the basic reversible gates are NOT gate, CNOT gate [14], TOFFOLI gate [15], FREDKIN gate [16], PERES gate [17], and SWAP gate [18]. The most commonly used gate libraries are NOT-CNOT-TOFFOLI (NCT) library and Multiple Controlled TOFFOLI (MCT) or  $k$ -CNOT library which can be used to synthesize a reversible circuit.

### 2.2 $k$ -CNOT Reversible Circuits

The structure of the reversible circuit is a linear cascade structure [19]. A reversible circuit  $C$  consists of a gate set  $G$ , where  $G = \{g_1, g_2, \dots, g_N\}$  and  $N$  is the number of gates in the circuit  $C$ . The gate  $g_i$  is the  $i^{\text{th}}$  gate for the circuit  $C$  where  $1 \leq i \leq N$ . Gate  $g_i$  is only active when the gate  $g_{i-1}$  has produced an output. Each gate of the reversible circuit is categorized by level, and the number of levels always depends on the number of gates present in the circuit.

If the reversible circuit contains only the  $k$ -CNOT gates, then it is called a  $k$ -CNOT based reversible circuit. The logical extension of a TOFFOLI gate is represented as a  $k$ -CNOT gate, where  $k$  is the number of input control lines  $x_1, x_2, \dots, x_k$  and one input target line  $t$ . Therefore,  $k$ -CNOT gate has  $(k + 1)$ -inputs and  $(k + 1)$ -outputs wires. The reversible logic function of this gate is expressed as  $(x_1, x_2, \dots, x_k, t) \rightarrow (x_1, x_2, \dots, x_k, t \oplus x_1, x_2, \dots, x_k)$ . Figure 1 shows the structure of the  $k$ -CNOT circuit, where four gates process the four signals (“wires”) from gate  $g_1$  to  $g_4$ . The gate  $g_1$  to  $g_4$  are 1-CNOT gate (also called as CNOT gate), 2-CNOT gate, 3-CNOT gate, and 1-CNOT gate, respectively. Figure 1, illustrates the propagation of input vector “1010” to the primary output vector “1100”. Since all the  $k$ -CNOT gates are reversible, so each gate produces a unique output in each level during the gate operation. Here, the output vector “1110” is generated in gate  $g_3$  which lies between levels 3 and 4, when the input vector “1010” is applied. Furthermore, if any gate produces different vector at their corresponding level, then a different output vector is produced in place of “1100” at the primary output level in the circuit.

### 2.3 Fault Models in Reversible Circuit

In general, a fault model is a mathematical model that describes the different levels of abstraction of physical faults in a system. The level of abstraction can be defined as behavioral, functional, structural, and geometric [10]. The fault models help to evaluate faults and reduce the complexity of generating test vectors [11]. Apart from the traditional fault models, few more fault models need to be considered for describing the faults that occur in the  $k$ -CNOT based reversible circuit and these faults are Single Missing Gate Fault (SMGF) [7], Multiple Missing Gate Fault (MMGF) [7], Repeated Gate Fault (RGF) [8], and Partial Missing Gate Fault (PMGF) [8] under the Missing Gate Fault model. In this work, the main emphasis is given

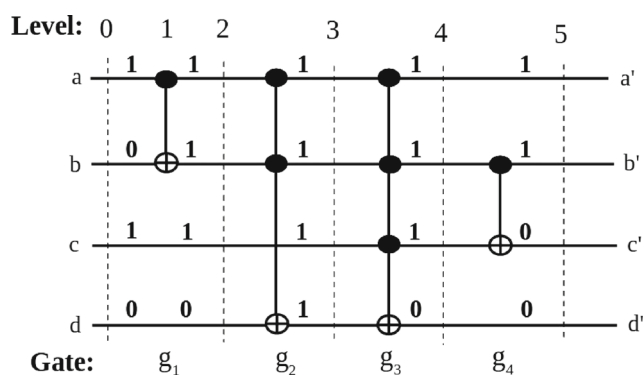


Fig. 1  $k$ -CNOT reversible circuit comprising of two 1-CNOT, one 2-CNOT, and one 3-CNOT gate

on SMGF and MMGF, which are structural fault models in the reversible circuit.

#### 2.3.1 Single Missing Gate Fault Model

A single missing gate fault (SMGF) causes removal of one  $k$ -CNOT gate from the circuit. The occurrence of an SMGF is that the generated signal(s) for operating the gate is (are) short, missing, misaligned or mistuned [7]. Consider the circuit of Fig. 2a, if we apply  $x_1=1, x_2=1$  and  $x_3=0$  at the input of the circuit, the normal output would be 1, 0, 1 for  $y_1, y_2$  and  $y_3$ , respectively. Consider the SMGF, where the second CNOT gate is missing, due to the presence of SMGF in the circuit; the output would be  $y_1=1, y_2=1$ , and  $y_3=1$ . The behavior of this fault model shows that the maximum number of SMGF faults in a circuit is equal to the total number of gates  $N$  present in the circuit.

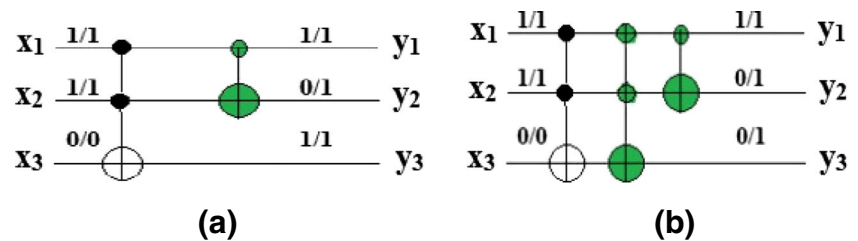
#### 2.3.2 Multiple Missing Gate Fault Model

A multiple missing gate fault (MMGF) causes the removal of two or more consecutive  $k$ -CNOT gates [8]. An example involving two missing gates is shown in Fig. 2b. The second and third  $k$ -CNOT gates are missing and due to this, the output would be  $y_1=1, y_2=1$ , and  $y_3=1$  instead of  $y_1=1, y_2=0$ , and  $y_3=0$ . In 2005 Polian et al. [8] showed that the complete test set of SMGFs is not capable of detecting all the MMGFs. Furthermore, based on the characteristic of SMGF, every SMGF is a subset of MMGF. Therefore, the total number of consecutive MMGFs in an  $N$ -gate circuit is  $N(N + 1)/2$ .

### 2.4 Related Work

Some of the existing research works that are compatible with our work are briefly reviewed in this section. In 2004, Hayes et al. [7] showed that  $\lceil N/2 \rceil$  test vectors detect all Missing Gate Faults (MGFs) in an  $N$ -gate  $k$ -CNOT circuit. Here, the MGF considers one gate missing at a time. Also, this method proposed that a single vector is capable of detecting all the MGFs of a given  $k$ -CNOT circuit by adding one wire and several 1-CNOT gates. The proposed a design for testability (DFT) method inverts the values that correspond to the undetectable faults such that all detection conditions can be satisfied simultaneously. Based on the concept of the previous work, the authors in [8] proposed a different type of fault that occurs under the Missing-Gate Fault model. This work presented a method to generate the optimal test sets computed by integer linear programming (ILP) to detect the various types of MGFs such as Single Missing Gate Fault (SMGF), Repeated Gate Fault (RGF), Multiple Missing Gate Fault (MMGF), and Partial Missing Gate Fault (PMGF). Also, this work showed that the

**Fig. 2** **a** Single missing gate fault **b** Multiple missing gate fault



complete test set of SMGFs is not capable of detecting all the MMGFs. The total number of consecutive MMGFs in an  $N$ -gate circuit is  $N(N + 1)/2$ . In 2008, authors in [20] proposed a scheme that divides the circuit into sub-circuits to get the complete test. The division of the circuit is based on the dominant and independent relationship of the gates. If the consecutive  $k$ -CNOT gates are dominant, then they are divided into the same sub-circuit; otherwise, these gates belong to different sub-circuits. This work generated the test vectors for each sub-circuits to obtain the complete test set. However, the generated complete test set by dividing the sub-circuit method is not minimal. The authors proposed the set covering method to get the minimal test set for detecting the SMGFs and MMGFs in  $k$ -CNOT circuits. The methodology proposed by them does not cover all the MMGFs and several additional test vectors are required to detect all the MMGFs. In 2008 and 2011, Rahaman et al. [21, 24] proposed a DFT method for an  $(n \times n)$  reversible circuit by adding only one extra line along with duplication of each  $k$ -CNOT gate to get a universal test set of size  $(n + 1)$ , which is sufficient to detect all SMGFs along with other faults like RGFs and PMGFs. In 2010, Kole et al. [22] proposed an algorithm for detecting SMGFs, MMGFs and RGFs. The proposed algorithm derived an optimal test set (OTS), where each gate is represented by a gate  $Id$  and each  $Id$  is used as a key to represent the permutation produced by the  $k$ -CNOT corresponding gate. Here, all permutations of size  $n$  for each gate  $Id$  are generated in a given an  $n$ -input reversible circuit and  $N(N+1)/2$  sets are constructed where circuit depth is  $N$ . The minimal set cover is used to derive an OTS from these sets. It is observed that if the circuit size is large in terms of a number of gates, then the construction of permutations for each gate  $Id$  is complex. In 2010, Zhang et al. [25] proposed an ATPG method using the concept of Boolean satisfiability (SAT) for generating the complete test set, which can detect the single missing control fault (SMCF), single additional control fault (SACF), and single missing-gate fault (SMGF) in reversible circuits. However, this method does not provide any guarantee to generate a minimal complete test set. In 2011, the work presented by Wille et. al [26] proposed an ATPG method using a simulation-based technique, Boolean satisfiability (SAT) based, and pseudo-Boolean optimization (PBO) based approach to test reversible circuits. These approaches are used for detecting the single missing control fault (SMCF)

along with the single missing-gate fault (SMGF), and a single additional control fault (SACF). The authors in [26] have mentioned that the PBO-based approach is more effective in terms of the size of the test set as compared to the simulation-based and SAT-based approaches. In 2011, Zhang et al. [27] proposed a SAT-based algorithm for determining the minimal complete test set to detect SMCF and SMGF in reversible circuits. This work is basically an improvement over the existing works presented in [25], and [26]. In 2011, Zamani et al. [23] proposed a technique named as Ping-Pong testing that provided a test vector to the circuit and generated output is considered as the next vector to detect the SMGFs and RGFs. This technique showed 100% fault coverage for SMGF as well as single RGF. But, for the multiple MGF (MMGFs) fault coverage is 86% on an average. In 2013, Mondal et al. [28] proposed a DFT technique for detecting all faults PMGFs, SMGFs, and detectable RGFs in a  $(n \times n)$   $k$ -CNOT based reversible circuit. The proposed DFT method can be implemented by the addition of only two extra input lines along with the insertion of a few  $k$ -CNOT gate to generate a universal test set of size  $(n+1)$  to detect all the faults. However, the quantum cost of the testable circuit increases when the circuit size is large. In 2014, Mondal et al. [31] proposed a Boolean generator which is developed by the Boolean difference method to derive the test set. The derived test is in the form of Boolean expression only, and it is capable of detecting all the SMGF in  $k$ -CNOT based reversible circuits. In 2016, Nagamani et al. [32] proposed an ATPG algorithm using the exact approach for generating the complete test set which can detect the single and multiple stuck-at faults, single and multiple missing gate faults, repeated gate faults, and partial missing gate faults. These algorithms provided an optimal solution, but the complexity of these algorithms is exponential to the gate count and the number of lines in the circuit, which is not suitable for large circuits. In 2017, Kole et al. [29] discussed the effort of evaluation for generating the test pattern of testing reversible circuits. For this purpose, this work presented two ATPG approaches (i) naive test generator and (ii) SAT-based test generator to test the SMGF along with SACF, and PMGF in reversible circuits. In this work, the SAT-based exact approach produces a smaller number of test patterns as compared to the naive test generator approach, but it requires an extensive run time and is not scalable.

In 2017, Prakash Surhonne et al. [33] provided a method to generate Automatic test patterns for MMGFs detection (considered only two gates are missing). It is based on the generated SMGFs which are stored in a Binary Decision Diagram (BDD) and test patterns to detect all MMGFs are generated by dependency analysis between the two gates. In 2018, Nagamani et al. [30] proposed a genetic algorithm-based heuristic test generation method for detecting the complete missing-gate fault (CMGF), PMGF, bridging fault and stuck-at fault in Toffoli-based, Peres-based and Fredkin-based reversible circuits. Based on the concept of a genetic-based heuristic method, the authors provide two approaches, namely as a random search approach and a directed search approach. The random search approach does not achieve high fault coverage for every test. The directed search approach is an improvisation over a random search approach.

It is observed that many of the approaches considered the DFT methodology for generating the test vectors to cover all the faults. In DFT methodology, there is an additional circuit overhead which is incorporated by additional input lines or control lines and additional gates. Moreover, by using an exact approach the minimal complete test set can be generated, but the computational complexity grows exponentially for large reversible circuits.

In this work, we propose an ATPG method to determine the complete test set for detecting all SMGFs and MMGFs using the reversible circuit properties (controllability and observability) without changing the structure of the circuit and the time complexity of the method is  $O(N^2)$  with  $N$  number of gates in a  $k$ -CNOT circuit. Moreover, the proposed complete test set generation method that can be applied to any large  $k$ -CNOT based reversible circuit.

### 3 Proposed Method

In this section, a method is proposed for  $k$ -CNOT reversible circuit to generate the complete test set which can detect the SMGFs and MMGFs. The proposed method starts with the generation of the complete test set for detecting all the SMGFs. A local test pattern is applied to each gate of the circuit and by using the reverse simulation technique, the complete test set is generated to detect the SMGFs of the given  $k$ -CNOT based reversible circuits. After analysis of the complete test set for SMGF, it is observed that the generated complete test set for SMGF is unable to detect all the possible MMGFs in a given reversible circuit. Therefore, a solution is formulated to generate the complete test set for detecting all the MMGFs considering the structure of the  $k$ -CNOT circuit and the complete test set for detecting SMGFs. The generated complete test set is able to detect all the SMGFs and MMGFs. The generated test set to

detect SMGF and MMGF is not minimal. To obtain the minimality, a table is constructed by using fault simulation with the generated test set. Integer Linear Programming (ILP) is formulated by using the fault simulation table and the minimal test set is obtained for a given circuit by the Branch and Bound technique of ILP.

Some terms are defined formally which are used to describe the proposed solution.

**Definition 1** A test vector  $TV_i$  is a combination of binary inputs that are applied to a reversible circuit for testing. The binary inputs  $\langle b_1 b_2 \dots b_n \rangle$  are assigned to the input lines, where  $b_i$  is the  $i^{th}$  bit that refers to the  $i^{th}$  line of the reversible circuit.

**Definition 2** The test set  $TS$  is the set of test vectors that are required to test all the possible faults (SMGFs and MMGFs) in the reversible circuit. Let the test set be  $TS = \{TV_1, TV_2, \dots, TV_k\}$ , for  $1 \leq i \leq k$ , then the test vector is  $TV_i = \langle b_{1i} b_{2i} \dots b_{ni} \rangle$ , where  $b_{ji}$  is the  $j^{th}$  bit of  $i^{th}$  test vector and  $b_{ji} \in \{0, 1\}$ .

**Definition 3** A local test pattern  $TV_{lp}$  is a combination of binary inputs that are applied to each  $k$ -CNOT gate of the reversible circuit to activate the gate for detecting any faults therein. Let the local test pattern be  $TV_{lp} = \langle b_1 b_2 \dots b_n \rangle$ , where  $b_i$  is  $i^{th}$  bit that refers to the  $i^{th}$  line for  $1 \leq i \leq n$  and  $b_i = 1$ .

**Definition 4** The test set  $TS_{SMGF}$  is the complete test set to detect all the single missing gate faults (SMGFs) in a given reversible circuit. In other words, the test set  $TS_{SMGF}$  is capable of detecting all the SMGFs, which occur at any level in the reversible circuit and  $TS_{SMGF} \subset TS$ .

**Definition 5** The test set  $TS_{MIN}$  is the minimal complete test set to detect all the single and multiple missing gate faults in a given reversible circuit.

**Definition 6**  $F_{SMGF}$  and  $F_{MMGF}$  are the sets consisting of all single and multiple missing gate faults, respectively in a given  $n$ -input reversible circuit.

The notation  $f_{g_i}$  for SMGF is used to denote a single missing gate fault of the  $i^{th}$  gate for  $1 \leq i \leq N$ . The multiple missing gate fault for missing of  $q$  number of gates is denoted as  $f_{g_1, g_2 \dots g_q}$ .

#### 3.1 Detection Technique for Single Missing Gate Fault

Consider a reversible circuit consisting of  $N$  gates  $\{g_1, g_2, \dots, g_N\}$ . For every gate  $g_i$ ,  $i=1$  to  $N$ , there are some control line(s) (denoted as  $\bullet$ ), some unconnected



line(s) and a target line (denoted as  $\oplus$ ). To generate the test set for SMGFs, the gates are scanned from left to right. To activate the gate  $g_i$  for detecting any faults therein, a local test vector is applied to the gate  $g_i$  by assigning the logic value 1 on the lines where control connections are present, and randomly we set the logic value to either 0 or 1 on all other lines (unconnected and target lines). In this work, the logic value 1's is considered for all lines in a given  $k$ -CNOT based circuit, and a local test pattern  $TV_{lp}$  represents it. With the applied local test pattern  $TV_{lp}$  to the gate  $g_i$ , the back propagation toward the input side is used to obtain the required test vector ( $TV_i$ ) to detect the missing gate fault for gate  $g_i$ . We carry out fault simulation with test vector  $TV_i$  to detect SMGFs, and then we remove the faults that are detected by  $TV_i$ . These detected faults need not be considered for further iteration and we repeat this process until all the faults are covered for the fault set  $F_{SMGF}$ .

### 3.2 Complete Test Set Generation for Single Missing Gate Fault

In this section, the proposed method to generate a complete test set for detecting all single missing gate faults in a given reversible circuit is described. The complete test set generation method for single missing gate faults is presented in Algorithm 1.

---

**Algorithm 1** Complete test set  $TS_{SMGF}$  generation for detecting the SMGFs.

---

**Input:** Reversible Circuit  $C$  with  $n$  input-lines and  $N$ -gates.

**Output:** The complete test set  $TS_{SMGF}$  for detecting the SMGFs.

- 1 Extract the required parameters  $n$  and  $N$  from circuit  $C$  comprising of  $N$  gates  $g_1, g_2, \dots, g_N$  with  $n$  input lines. Construct the local test vector  $TV_{lp} = \langle b_1 b_2 \dots b_n \rangle$ ,  $b_i = 1$  and  $i = 1$  to  $n$
  - 2 Generate the fault set  $F_{SMGF}$  consisting of all SMGFs in a given  $n$ -input reversible circuit.
  - 3 Back propagate  $TV_{lp}$  for each gate  $g_i$  to obtain the corresponding input test vector  $TV_i$  (occur at input level) and  $TS_{SMGF} \leftarrow TV_i$ . Since every gate  $g_i$  is reversible, for a given  $TV_{lp}$ , the test vector  $TV_i$  is unique.
  - 4 Carry out fault simulation with test vector  $TV_i$  to determine the faults in fault set  $F_{SMGF}$  that get detected.
  - 5 Construct the complete test set  $TS_{SMGF}$  after eliminating the all duplicate test vector  $TV_i$ .
- 

*Example 1* The complete flow of Algorithm 1 for generating the test set  $TS_{SMGF}$  of the reversible benchmark circuit  $rd32-v0.66$  is represented in Fig. 3.

Let the benchmark circuit  $rd32-v0.66$  be provided as input to Algorithm 1 having  $n$ -lines and  $N$ -gates. In Step 1, the required parameters  $n = 4$  and  $N = 4$  are extracted from the given circuit. Now, we construct the local test pattern  $TV_{lp} = \langle b_1 b_2 b_3 b_4 \rangle$ , where bit  $b_i$  is assigned to logic value 1, where  $i = 1$  to 4. In Step 2, the fault set  $F_{SMGF} = \{f_{g1}, f_{g2}, f_{g3}, f_{g4}\}$  is generated based on the connections of  $N$  gates. In Step 3, we apply and back propagate  $TV_{lp} = \langle 1 1 1 1 \rangle$  to the gate  $g_4$  and obtain the corresponding test vector  $TV_4 = \langle 1 0 0 1 \rangle$  at the input level. Similarly, the same process is repeated for the remaining gates  $g_3, g_2$ , and  $g_1$  and the corresponding test vectors are  $TV_3 = \langle 1 0 1 0 \rangle$ ,  $TV_2 = \langle 1 0 1 1 \rangle$ , and  $TV_1 = \langle 1 1 1 0 \rangle$ , respectively. In Step 4, the fault simulation process for each fault  $f_{g_i}$  in the fault set  $F_{SMGF}$  is carried out with the help of each test vector  $TV_i$  at the input level. For this example, the faults  $f_{g1}, f_{g2}, f_{g3}$ , and  $f_{g4}$  are identified by test vectors  $TV_1 = \langle 1 1 1 0 \rangle$ ,  $TV_2 = \langle 1 0 1 1 \rangle$ ,  $TV_3 = \langle 1 0 1 0 \rangle$ , and  $TV_4 = \langle 1 0 0 1 \rangle$ , respectively. Finally, complete test set  $TS_{SMGF} = \{1110, 1011, 1010, 1001\}$  is constructed for the benchmark circuit  $rd32-v0.66$  as mentioned in Step 5.

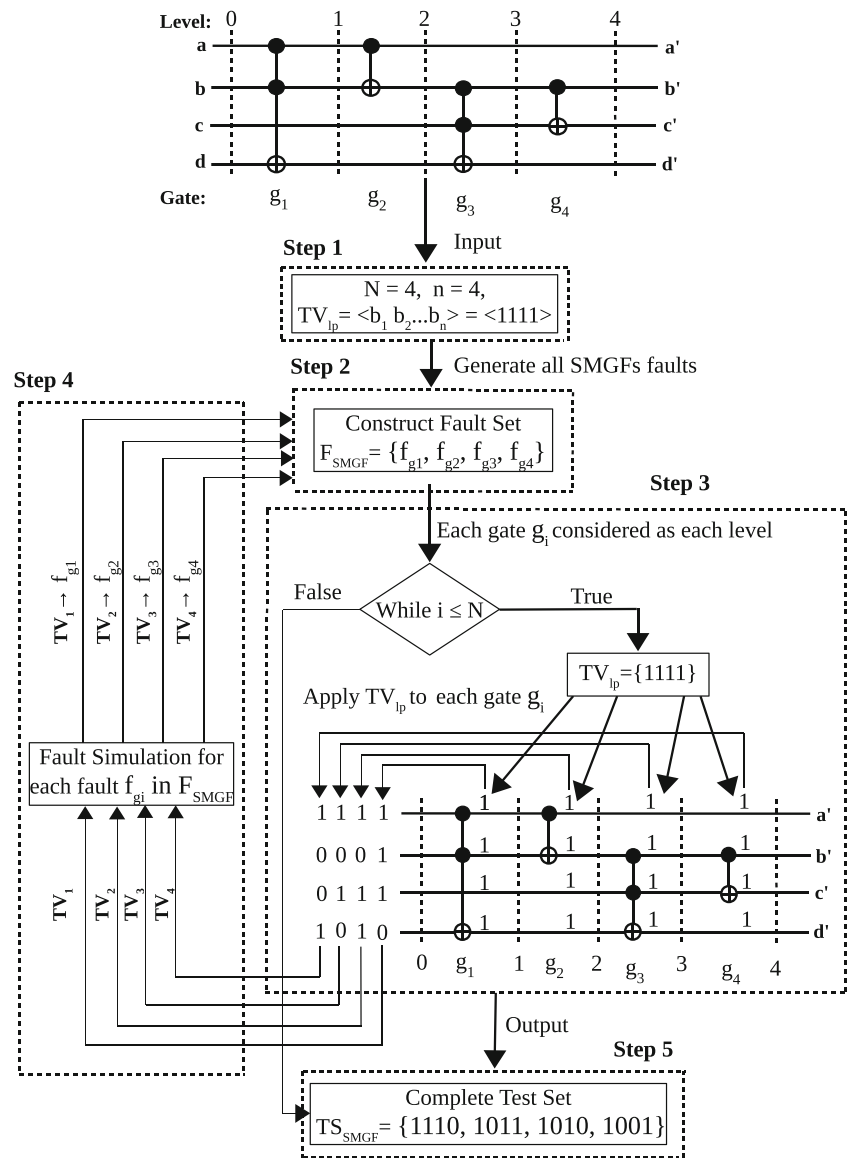
**Lemma 1** The test set,  $TS_{SMGF}$ , generated using the proposed method detects all single missing gate faults in a given reversible circuit with  $N$ -gates.

*Proof* Let us consider the  $k$ -CNOT circuit  $C$  consisting of gates  $\{g_1, g_2, \dots, g_N\}$ . To activate the SMGF at gate  $g_i$  ( $i=1$  to  $N$ ), the local test pattern  $TV_{lp} = \langle b_1 b_2 \dots b_n \rangle = \langle 1 1 \dots 1 \rangle$  is applied at the output level of gate  $g_i$ . The controllability property ensures that on backtracking from any gate  $g_i$  with local test pattern  $TV_{lp}$  it is always possible to generate a unique vector (say  $TV_i$ ) at the input gate-level. The observability property ensures that the generated test vector  $TV_i$  for each gate  $g_i$  produces fault-free output and the same test vector  $TV_i$  produces faulty output at the primary gate-level, if a fault occurs. Therefore, each test vector  $TV_i \in TS_{SMGF}$  detects all the individual faults in the fault set  $F_{SMGF}$ . Hence, the test set  $TS_{SMGF}$  is the complete test set for detecting all the single missing gate faults in a given reversible circuit.  $\square$

### 3.3 Complexity Analysis of Complete Test Set Generation

For an  $n$ -input reversible circuit with  $N$  number of gates, the local test pattern  $TV_{lp}$  is applied for each gate  $g_i$  for detecting the SMGFs in the circuit. Construction of the local test pattern  $TV$  requires constant time (say  $C_1$ ) due to assigning the logic value 1 to all the lines  $n$  in a given reversible circuit with  $N$  number of gates. The back propagation for each gate  $g_i$  requires  $N$  steps to obtain the

**Fig. 3** Demonstration of Algorithm 1 for *rd32\_v0\_66* benchmark circuit



corresponding input vector  $TV_i$  at the input level. Now, recursively we simulate all the faults in the fault set  $F_{SMGF}$  with the test vector  $TV_i$ , where  $|F_{SMGF}|=N$ . For checking the presence of a test vector in the fault set  $F_{SMGF}$ , it requires constant time, say,  $C_2N$ . Therefore, the overall time complexity of the algorithm is  $C_1+N+C_2N=O(N)$ .

### 3.4 Complete Test Set Generation for Multiple Missing Gate Fault

The generated test set  $TS_{SMGF}$  obtained by Algorithm 1 is sufficient for detecting all the single missing gate faults (SMGFs) in a reversible circuit, but the test vector  $TV_i \in TS_{SMGF}$  is not capable of detecting all the multiple missing gate faults (MMGFs). Moreover, a complete test set for SMGFs does not cover all the MMGFs [8]. For every gate

$g_i$  of a reversible circuit, there are some control line(s) (denoted as  $\bullet$ ), some unconnected line(s) and a target line (denoted as  $\oplus$ ).

In  $k$ -CNOT circuit structure, some of the lines contain only target connections, and in some lines both target and control connections are available. If only the target connections are present in a line, then some of the multiple missing gate faults can not be detected by the test set for detecting SMGFs. Consider the case that two consecutive gates are missing whose targets are in the same line, then the single missing gate fault of the first gate is nullified by the missing of the second gate and so missing of two gates cannot be detected by the test set of SMGFs. Therefore, for this category of circuits, the test vector  $TV_i \in TS_{SMGF}$  cannot detect all the MMGFs in a given  $k$ -CNOT based circuit. For detecting all the MMGFs, some additional test

vectors are included in the test set  $TS_{SMGF}$ . For this purpose, the test set  $TS$  is constructed as:  $TS = \bigcup_{i=1}^N S(g_i) \cup TS_{SMGF}$ .  $S(g_i)$  represents all the possible test vectors  $TV_i$  for the corresponding SMGFs in gate  $g_i$ . In the other case, target and control connections are present on the same line. Consider a case where a control point is followed by a target point is a line. That is, gate  $g_i$  is connected as target point and gate  $g_j$  is connected as a control point in the line  $l$ . So, the missing of gate  $g_i$  effects the control connections of gate  $g_j$  which eventually effects the output of gate  $g_j$ . Missing of both the gates  $g_i$  and  $g_j$  effect the control connects of the next gate. The absence of the control and target in these gates directly effect the control and target connection of the next consecutive gates, and as a result, the primary output of fault-free circuit gets effected. Therefore, the generated test set  $TS_{SMGF}$  for detecting SMGFs is capable of detecting all the MMGFs in a given  $k$ -CNOT circuit for this category of circuits. The construction of the complete test set  $TS$  for detecting all MMGFs is given in Algorithm 2.

**Algorithm 2** COMPLETE test set generation for MMGFs.

**Input:**  $k$ -CNOT based reversible Circuit.  
 $N$ : Number of gates of the circuit.  
 $n$ : Number of input lines of the circuit.  
 $TS_{SMGF}$ : The complete test set for SMGFs.  
 $S(g_i)$ : All the possible test vectors  $TV_i$  for SMGFs in gate  $g_i$ .

**Output:** The complete test set  $TS$  for detecting MMGFs

```

1  $TS = TS_{SMGF}$ 
2  $flag = false$ 
3 for  $i \leftarrow 1$  to  $n$  do
4   if line  $l_i$  contains only target connection then
5      $flag = true$ 
6 if ( $flag$ ) then
7   for  $j \leftarrow 1$  to  $N$  do
8      $TS = TS \cup S(g_j)$ 
9 return  $TS$ 

```

**Example 2** Consider the benchmark circuit Toffoli\_double\_4 as illustrated in Fig. 4. The circuit consists of two  $k$ -CNOT gates (i.e.,  $N=2$ ) and four input lines (i.e.,  $n=4$ ). Based on the circuit structure as shown in Fig. 4, there is an occurrence of two consecutive target connections on the same line 'd'. Hence, the generation of the complete test set  $TS$  is according to the Step:4 of Algorithm 2. The complete test set for SMGFs is  $TS_{SMGF} = \{1110, 1111\}$  for the reversible circuit Toffoli\_double\_4 according to Algorithm 1. In the circuit of Fig. 4, if the gate  $g_1$  is missing then all possible test vectors to detect the missing gate  $g_1$  is

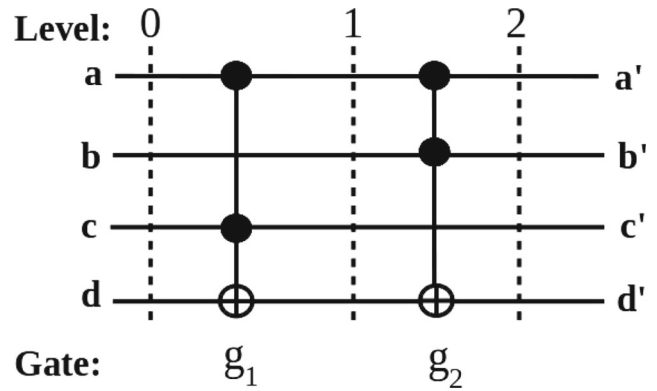


Fig. 4 Reversible benchmark circuit: Toffoli\_double\_4 circuit

$S(g_1) = \{1010, 1011, 1110, 1111\}$  and similarly all possible test vectors to detect the missing gate  $g_2$  is  $g_2 = \{1100, 1101, 1110, 1111\}$ . Now, the required test set to detect all MMGFs is  $TS = \{S(g_1) \cup S(g_2)\} \cup TS_{SMGF} = \{1010, 1011, 1110, 1100, 1101, 1111\}$ . The obtained test set  $TS$  is capable of detecting all the faults in fault sets  $F_{SMGF}$  and  $F_{MMGF}$ . Thus, the test set  $TS$  is complete but not minimal to detect all MMGFs in the reversible benchmark circuit Toffoli\_double\_4.

**Lemma 2** The test set  $TS$  generated using the proposed method detects all single and multiple missing gate faults in a given  $n$ -input  $k$ -CNOT based reversible circuit.

**Proof** Let us consider that the target connections of the gates  $g_i$  and  $g_{i+1}$  which are in the same line. It means that the operation of the gate  $g_i$  does not effect the gate  $g_{i+1}$  if they are missing together, i.e., fault-free and faulty output are indistinguishable, since the missing of two consecutive target connections does not effect the functional behavior of the circuit. If the generated test vector  $TV_i \in TS_{SMGF}$  is applied, then missing of even number of consecutive gates shows the similar functional effect as compared to the fault-free circuit. Due to this, the test set  $TS_{SMGF}$  is unable to detect all the MMGFs. In this case, all the possible test vectors  $S(g_i)$  for all possible occurrence of SMGFs are considered. However, each single missing gate fault is also a multiple missing gate fault [8]. Thus, any test vector in  $S(g_i)$  involves detecting two or more multiple missing gate faults where the target connection is in the same line for each gate. Hence, the generated test set  $TS = \bigcup_{i=1}^N S(g_i) \cup TS_{SMGF}$  is capable of detecting all multiple missing gate faults in a given  $k$ -CNOT based circuit. In another case, the control connection of gate  $g_i$  is the target connection of the gate  $g_{i+1}$  and vice versa; then control connection of gate  $g_i$  directly effects the gate  $g_{i+1}$ . The complete removal of two gates  $g_i$  and  $g_{i+1}$  generate the faulty responses at the



primary output of the circuit, which is distinguishable with fault-free primary output responses. The multiple missing gate faults in this scenario are detected by the test vector  $TV_i \in TS_{SMGF}$ . Since the back propagation of each gate produces different vectors in subsequent levels, thus, at least one generated test vector  $TV_i \in TS_{SMGF}$  can detect the multiple missing gates  $g_i$  and  $g_{i+1}$ . In the similar analogy, it can be stated that the test set  $TS$  ( $TS = TS_{SMGF}$ ) is capable of detecting more than two multiple missing gate faults.  $\square$

### 3.5 Complexity Analysis of Complete Test Set $TS$ Generation

The generation of the complete test set  $TS$  for all possible SMGFs and MMGFs is dependent on the number of input lines  $n$  and the number of gates  $N$  in the circuit. All the lines in the circuit are scanned to identify the type of connections (target and control) present in a line for each gate, and for checking the type of connection in a line requires constant time ( $C_1$ ). The time complexity for checking the type of connections is  $N \times n \times C_1$ . Evaluation of the test vectors for each gate also requires constant time ( $C_2$ ) and time complexity for evaluation of the test set for all the gates is  $N \times C_2$ . Hence, the time complexity for generating the test set  $TS$  is  $N \times n \times C_1 + N \times C_2 = O(N \times n)$ . The time complexity in the worst case is  $O(N^2)$ , where  $n = N$ .

### 3.6 Determination of Minimal Complete Test Set

The complete test set  $TS$  generated by Algorithm 2 to detect all MMGFs is not a minimal one. A method is proposed to derive the minimal complete test set  $TS_{Min}$ . For this purpose, firstly, a row and column fault covering table is constructed with the help of the complete test set  $TS$  and all possible faults present in  $F_{SMGF}$  and  $F_{MMGF}$  in a given reversible circuit. Secondly, Integer Linear Programming (ILP) Problem is formulated from the constructed row and column fault covering table. Finally, using Branch and Bound technique of ILP, the test set  $TS_{MIN}$  is obtained for detecting all SMGFs and MMGFs. The following steps are carried out to generate  $TS_{MIN}$ :

- (i) Using fault simulation with each test vector  $TV_i \in TS$ , the corresponding faults in  $F_{SMGF}$  and  $F_{MMGF}$  are determined. The row and column fault covering table is constructed in the form of a matrix  $\mathbf{M}_{r \times c}$ , where  $r$  is the number of test vectors present in  $TS$  and  $c$  is the number of all possible faults in a given reversible circuit. The value  $\mathbf{m}_{(i,j)}=1$  denotes that the test vector  $TV_i$  in the  $i^{th}$  row detects the corresponding  $j^{th}$  fault; otherwise, the fault is undetectable by the test vector  $TV_i$  of the  $i^{th}$  row.

- (ii) Formulate the ILP model with binary decision variables  $t_i$  associated with each test vector  $TV_i$ . Let us consider  $|TS| = d$ , then there are  $d$  variables  $t_i$ , where  $i = 0$  to  $d - 1$  and  $t_i \in \{0, 1\}$ . The variable  $t_i$  is represented as  $i^{th}$  row of  $\mathbf{M}_{r \times c}$  and,  $\bar{T} = [t_0, t_1, \dots, t_{d-1}]^T$ . Here, we define:

$$\mathbf{M}_{r \times c} \cdot \bar{T} = \left[ \sum_{i=0}^{d-1} \mathbf{m}_{i1} t_i, \sum_{i=0}^{d-1} \mathbf{m}_{i2} t_i, \dots, \sum_{i=0}^{d-1} \mathbf{m}_{iN(N+1)/2} t_i \right]^T$$

where,  $\sum_{i=0}^{d-1} m_{ij}$  for all  $1 \leq j \leq N(N + 1)/2$  and if  $t_i=1$ , then corresponding test vector  $TV_i$  is able to detect the faults of  $j^{th}$  row of  $\mathbf{M}_{r \times c}$ .

The Objective function for ILP is formulated as follows:

$$\begin{aligned} \min f(t_0, t_1, \dots, t_{d-1}) &= t_0 + t_1 + \dots + t_{d-1} \\ \text{subject to the constraints } \mathbf{M}_{r \times c} \cdot \bar{T} &\geq [\mathbf{1}, \mathbf{1}, \dots, \mathbf{1}]^T \end{aligned}$$

- (iii) All the constraints  $\sum_{i=0}^{d-1} m_{ij} t_i \geq 1$  are applied in LINGO 17.0 [34] and shows that at least one test vector  $TV_i \in TS$  is able to detect any fault in a given circuit.

*Example 3* Consider the benchmark circuit *ham3tc* as shown in Fig. 5. The total number of faults in the circuit *ham3tc* is 15. According to Algorithm 2, the extracted complete test set  $TS=\{011, 101, 110, 100\}$  for the reversible benchmark circuit *ham3tc*. Using the fault simulation with each test vector  $TV_i \in TS$  with the fault-free and faulty circuit, the corresponding faults are extracted for the circuit *ham3tc*. Based on this information, the row and column fault covering table is constructed according to the Step (i) to generate  $TS_{MIN}$ . Table 1 shows the fault coverage by each test vector  $TV_i$ . Next, we formulate the ILP model as mentioned in Step (ii):  $\min f(t_0, t_1, t_2, t_3)=t_0+t_1+t_2+t_3$ , subject to the generated constraints as indicated in Table 1. In Step (iii), all the constraints generated by the ILP formulation are applied to the LINGO 17.0, and it gives  $t_0=1$  and  $t_1=1$ . The respective test vectors for the binary decision variables  $t_0$  and  $t_1$  are 011 and 101, respectively. Therefore, the minimal test set is  $TS_{MIN}=\{011, 101\}$  for the reversible benchmark circuit *ham3tc*.

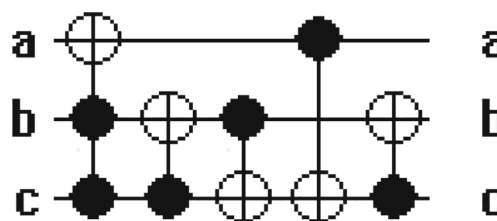


Fig. 5 Reversible benchmark circuit: *ham3tc* circuit

**Table 1** Row and column fault coverage table of the *ham3tc* reversible benchmark circuit

$TV_i$	$F_{SMGF}$					$F_{MMGF}$									
	$f_{g1}$	$f_{g2}$	$f_{g3}$	$f_{g4}$	$f_{g5}$	$f_{g1,g2}$	$f_{g2,g3}$	$f_{g3,g4}$	$f_{g4,g5}$	$f_{g1,g2,g3}$	$f_{g2,g3,g4}$	$f_{g3,g4,g5}$	$f_{g1,g2,g3,g4}$	$f_{g2,g3,g4,g5}$	$f_{g1,g2,g3,g4,g5}$
011	1	1	0	1	0	1	1	1	1	1	1	1	1	1	1
101	0	1	1	1	1	1	1	0	1	1	1	1	1	0	0
110	0	0	1	1	0	0	1	0	1	1	0	0	0	0	0
100	0	0	0	1	1	0	0	1	1	0	1	1	1	1	1

following constraints are generated from the fault coverage table

$$\sum_{i=0}^{d-1} m_{ij}t_i \geq 1 \implies M = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} t_0 \\ t_1 \\ t_2 \\ t_3 \end{bmatrix}$$

$$\geq \begin{bmatrix} 1 \\ 1 \end{bmatrix} \implies \begin{matrix} t_0 \geq 1 \\ t_0 + t_1 \geq 1 \\ t_1 + t_2 \geq 1 \\ t_0 + t_1 + t_2 + t_3 \geq 1 \\ t_1 + t_3 \geq 1 \\ t_0 + t_1 + t_2 \geq 1 \\ t_0 + t_3 \geq 1 \\ t_0 + t_1 + t_3 \geq 1 \end{matrix}$$

**3.6.1 Complexity Analysis of ILP Formulation**

For the determination of a complete minimal test set  $TS_{Min}$ , we consider 0-1 ILP, where each binary decision variable  $t_0, t_1, \dots, t_{d-1}$  can assume binary value either 0 or 1. Suppose, consider the number of constraints is  $k$ , and all the constraints are lower bound constraints. For the state space, let us consider all possible  $2^d$  assignments of binary

decision variables  $t_0, t_1, \dots, t_{d-1}$  in a non-deterministic manner. Based on the ILP formulation, the time taken for checking the feasible solution for each assignment is  $O(d \times k)$ . The minimized objective function is solved using the Branch and Bound technique of ILP and computing the value of the objective function for each feasible assignment needs  $O(d)$  time.

**Lemma 3** *The test set  $TS_{MIN}$  is a minimal complete test set for detecting all the single and multiple missing gate faults in a given  $n$ -input  $k$ -CNOT based reversible circuit.*

*Proof* In Lemma 2, it is established that the test set  $TS$  can detect all single and multiple missing gate faults of a given reversible circuit. Moreover, each possible fault can be detected by one of the test vector  $TV_i \in TS$  because each  $TV_i$  satisfies the condition  $M_{r \times c} \cdot \bar{T} \geq [1, 1, \dots, 1]^T$ . Based on the ILP formulation, the objective function  $\min f(t_0, t_1, \dots, t_{d-1})$  provides the smallest number of binary variable(s) which is associated with their corresponding test vector  $TV_i$  in the test set  $TS$  under the condition  $M_{r \times c} \cdot \bar{T} \geq [1, 1, \dots, 1]^T$ . Thus, all the constraints of a given condition are applied to optimal software that gives the least number of variables, which are assigned to  $TS_{MIN}$ . Hence,  $TS_{MIN}$  is a minimal complete test set for detecting all single and multiple missing gate faults. □

**3.7 Fault Coverage Evaluation with Other Faults Models**

The fault coverage is defined as the ratio of the actual number of detected faults to the total number of faults that occur in a circuit. Several fault models are used to detect different kinds of faults. In this work, methods are proposed to generate the complete test set to detect SMGFs and MMGFs in a reversible circuit. There exists a correlation between different fault models and it is also a good exercise to check the fault coverage of other fault models by the generated test set of another fault model. Therefore, the minimal test set generated by our proposed method to detect SMGFs and MMGFs is applied for detecting the faults in other fault models such as stuck-at fault (SAF), partial

missing gate fault (PMGF), and appearance crosspoint fault. The correlations between different fault models are explained briefly.

1. **Missing gate fault and Stuck-at fault model:** According to Patel et al. [4], the stuck-at faults (SAFs) can be detected by test vector such that each wire at every level of the circuit can be set to both logic value 0 and 1, while SMGFs and MMGFs are detected by setting the control connections to logic value 1 and other connections (the target and unconnected) are set arbitrarily to logic values 0 and 1. In the proposed method, a local test pattern to each  $k$ -CNOT gate  $g_i$  is applied and we traverse back towards input to obtain the test vector  $TV_i$  at the input level. During back propagation through the gates present in the circuit, the operations performed in each gate changes the local test patterns at each level and are set to logic value 0 or 1, which are capable of detecting stuck-at 1 (SA1) and stuck-at 0 (SA0), respectively. Thus, the complete test set  $TS$  for SMGFs and MMGFs satisfies the requirement for detecting most of the SAFs in a given reversible circuit.
2. **Missing gate fault and PMGF model:** The detection criteria for a PMGFs is that the missing control input is set to logic value 0 and we assign the logic value 1 to all other control inputs. The remaining input lines are assigned arbitrarily to logic value 0 and 1. As per our proposed method, when we apply a local test pattern to a particular  $k$ -CNOT gate lying at a particular level in the circuit, then the test vectors get changed due to propagation through various gates at different levels. Therefore, there is a possibility to get some test vectors which satisfy the criteria for detecting the PMGFs. It may happen that all PMGFs can not be detected by the generated test set  $TS$  by our proposed method, then the test set can be reconstructed to cover all PMGFs.
3. **Missing gate fault and Crosspoint fault model:** As described above, to detect the SMGFs and MMGFs, we apply a logic value 1 where control connections are present in the gate and we randomly fill by logic value either 0 or 1 for all other lines. To detect for appearance crosspoint faults, we apply a logic value 1 to all control connections of the gate and set logic value 0 to all other lines. Therefore, it is observed that the test set for detecting both SMGFs and MMGFs is capable of covering appearance crosspoint faults. In our proposed method, the complete test set  $TS$  is generated by a local test pattern for each gate with the help of back-propagating through various levels in the circuit. Therefore, some test vectors  $TV_i$  in the test set  $TS$  satisfies the testing criteria for detecting the appearance crosspoint faults. Moreover, the PMGFs are same as the disappearance crosspoint faults.

## 4 Experimental Results and Discussions

The algorithms for the test set generation for single and multiple missing gate faults have been implemented in Python 3.4 and run on a Core-i5 machine with an Intel Pentium (R) CPU-8250U@ 1.60GHz  $\times$  8 system, running Ubuntu v16.04 (64-bit) with 8 GB RAM. The minimal test set generation method has been implemented in LINDO Extended 17.00 software and running on the same core-i5 machine. The reversible benchmark circuits based on the  $k$ -CNOT gates have been considered [35] to perform the experiments. The number of test vectors required for the detection of SMGFs and MMGFs before and after minimization along with the CPU time taken as per our proposed method are reported for the experimental results. The experimental results are compared with some of the previous works performed on SMGF and MMGF [7, 8, 26, 29–33]. The fault coverage range with other fault models such as stuck-at fault (SAF), partial Missing gate fault (PMGF) and crosspoint fault (appearance) in reversible circuits are also reported.

The experimental results are reported in Table 2. The first four columns in Table 2 provide the name of the benchmark circuit, number of input lines ( $n$ ), number of gates ( $N$ ), and the total number of faults for both SMGFs and MMGFs respectively. Columns 5 and 7 in Table 2 present the number of test vectors required for detecting both SMGFs and MMGFs before and after minimization, respectively. The CPU time (sec) to generate the complete test set  $TS$  and minimal test set  $TS_{MIN}$  for the benchmark circuits are presented in columns 6 and 8, respectively. Column 9 indicates the percentage of difference between the test set generated by the proposed method before and after minimization. From the reported results in Table 2, it is observed that the test set is minimized by more than or equal to 50% by the proposed minimization method for about 88% of the circuits, but still 100% fault coverage is retained. The maximum reduction of 95% of the test set for the reversible benchmark circuit 0410184-169 is achieved by our reduction technique. The minimum reduction of 33% of test set is observed for circuits *Fredkin-6*, *ex-1-166*, and *4gt11-84*. From the experiments performed on reversible benchmark circuits and the result reported in Table 2, it is evident that our proposed method is scalable to handle large circuits.

The results of our proposed method are compared with the work of [33] and the comparison is shown in Columns 4, 5 and 11 of Table 3. The value ‘-’ indicates that the results in the corresponding work [33] are not available. The authors in [33] proposed a greedy and BDD based covering method for generating the minimal test set for detecting all possible cases of two consecutive missing gate faults. The number of test patterns obtained in our proposed method

**Table 2** Complete and minimal test set for detection of SMGFs and MMGFs with CPU time (sec) for the benchmark circuits

Benchmarks circuit	$n$	$N$	Total No. Faults (SMGF+MMGF)	$TS$ (SMGF+MMGF)	Time (sec) $TS$	$TS_{MIN}$ (SMGF+MMGF)	Time (sec) $TS_{MIN}$	% Difference ( $TS$ & $TS_{MIN}$ )
Peres_9	3	2	3	2	0.00	1	0.02	50
Fredkin_6	3	3	6	3	0.00	2	0.02	33.33
nth_Prime	3	4	10	4	0.00	1	0.04	75
ex_1_166	3	4	10	3	0.00	2	0.04	33.33
ham_3	3	5	15	4	0.00	2	0.02	50
3_17_13	3	6	21	4	0.00	2	0.07	50
rd32_v0_66	4	4	10	4	0.00	2	0.06	50
mini_alu_167	4	6	21	5	0.00	3	0.04	40
mod10_171	4	10	55	6	0.00	3	0.11	50
hwb4_52	4	11	66	8	0.00	2	0.09	75
4-49d3	4	12	78	8	0.00	3	0.11	62.50
4_49_16	4	16	136	7	0.00	4	0.17	42.85
hwb4tc	4	17	153	7	0.00	3	0.13	57.14
4gt11_84	5	3	6	3	0.00	2	0.06	33.33
Xor5d1	5	4	10	4	0.00	1	0.06	75
mod5d4	5	5	15	4	0.00	2	0.03	50
alu-v0_26	5	6	21	5	0.00	2	0.07	60
mod5d1_63	5	7	28	5	0.00	2	0.08	60
mod5d1	5	8	36	28	0.00	4	0.05	85.71
mod5d2	5	9	45	8	0.00	2	0.10	75
mod8-1-_177	5	14	105	8	0.00	3	0.12	62.50
rd32_273	5	20	210	15	0.00	7	0.06	53.33
hwb5_55	5	24	300	14	0.00	4	0.15	71.42
hwb5_53	5	55	1540	19	0.00	5	0.19	73.68
hwb5tc	5	56	1596	19	0.00	5	0.26	73.68
graycode6_47	6	5	15	5	0.00	1	0.06	80
ex3_229	6	7	28	7	0.00	3	0.05	57.14
Xor5-254	6	7	28	6	0.00	2	0.04	66.67
majority-239	6	8	36	6	0.00	3	0.06	50
mod5adder-128	6	15	120	7	0.00	3	0.16	57.14
2of5d1	6	18	171	15	0.00	5	0.05	66.67
mod5adder	6	21	231	10	0.00	4	0.06	60
hwb6tc	6	126	8001	40	0.03	8	0.67	80
rd53d1	7	12	78	11	0.00	2	0.11	81.81
2of5d2	7	12	78	12	0.00	2	0.13	83.33
ham7_105	7	21	231	15	0.00	3	0.13	80
ham7tc	7	24	300	15	0.01	4	0.17	73.33
rd53rcmg	7	30	465	23	0.01	4	0.11	82.60
rd53d2	8	12	78	12	0.00	2	0.10	83.33
cm82a-208	8	22	253	16	0.01	4	0.09	75
rd73-140	10	20	210	20	0.04	3	0.14	85
6symd2	10	20	210	19	0.04	3	0.15	84.21
9symd2	12	28	406	28	0.23	3	0.15	89.28
adr4-197	13	55	1540	47	0.82	6	0.11	87.23
0410184-169	14	46	1081	40	1.11	2	0.19	95
rd84-142	15	28	406	28	1.57	3	0.17	89.28
ham15-108	15	70	2485	53	3.60	8	0.20	84.90
hwb5_131	28	88	3916	13	3.08	7	0.33	35.71

**Table 2** (continued)

Benchmarks circuit	$n$	$N$	Total No. Faults (SMGF+MMGF)	$TS$ (SMGF+MMGF)	Time (sec) $TS$	$TS_{MIN}$ (SMGF+MMGF)	Time (sec) $TS_{MIN}$	% Difference ( $TS$ & $TS_{MIN}$ )
rd84_313	34	104	5460	22	3.82	13	0.53	40.90
ham15_298	45	153	11781	30	4.12	10	0.67	66.66
bw_291	87	307	47278	14	5.31	9	0.88	50

is found to be less than the test patterns reported in [33]. Moreover, the method proposed in [33] can detect only two consecutive missing gate faults, whereas our proposed method can detect any number of consecutive missing gate faults. So, the fault coverage of our proposed method for multiple missing gate faults is more than that of [33]. The reduction of test set by our proposed test pattern generation method to detect MMGFs is more than or equal to 50% for more than 77% of the benchmark circuits that are used in the experiment. For the circuits *hwb6tc* and 0410184-169, the test sets are reduced by 81.39% and 77.77%, respectively. For the circuits, *ex3\_229* and *rd32-v0\_66*, the size of test sets are small [33] and so the scope of reducing the test set is less.

The authors in [32] proposed an ATPG algorithm to generate the complete test set using an exact approach for detecting missing gate faults along with single and multiple stuck-at faults, repeated gate faults, and partial missing gate faults. The exact approaches aim to provide the optimal solution, but these approaches are computationally expensive (exponential complexity) for large circuits. Our proposed method generates a minimal test for covering all the single and multiple missing gate faults. From a computational point of view, our method requires linear time for obtaining the minimal test set. The comparison of our results with the results of [32] is shown in Columns 6 and 11 of Table 3. It is observed that the size of the test set generated by our proposed method is same as the size of test set generated by the method of [32] for most of the circuits, but the computational complexity of our proposed method is less.

The authors of the work [7] proposed two approaches, the first one is the greedy heuristic and the second one is based on exact branch and bound algorithm for generating the test vectors to detect the SMGFs. The method used for generating the test set in [7] is a DFT based approach which requires incorporation of additional testing circuits, so there is extra overhead in the hardware. The experiment results are presented and compared in Columns 7, 8 and 11, respectively of Table 3. It is observed that due to the use of DFT method, several gates are added to each circuit to detect the faults. Also, the work reported in [7] can detect only SMGFs, but our proposed method can detect both SMGFs and MMGFs. For most of the circuits the size of

test set produced by our method is comparable to the size of test given by the method of [7].

The authors in [8] used an exact automatic test patterns generation method for detecting SMGFs and MMGFs based on the integer linear programming. The comparison of our result with the result of [8] is reported in Columns 9 and 11 of Table 3. It is observed that size of test set is same in both the methods for most of the circuits, but the computational complexity of an exact automatic test pattern generation method is always more.

The experimental result in Columns 10 and 11 of Table 3 shows the comparison of the proposed work with the work done in [31]. The authors in [31] targeted for generating the test set for detecting only SMGFs using the method of Boolean difference generator. It is observed that the size of the generated test set is less in our proposed method for most of the benchmark circuits as compared to the method of [31]. The maximum reduction of 66.67% for the size of test set is found for the circuit *rd32d1*.

For comparison purpose, we calculate the average number of test vectors generated by each method. From Table 3, it is observed that the average number of test vectors generated by our proposed method is reduced to half in comparison to the work of [33] and so the improvement in performance is 2X. The average number of test vectors generated by our proposed method is similar to the works reported in [7, 8, 31, 32].

The performance of our proposed method is compared with another set of works which are based on Boolean satisfiability and genetic algorithm. The comparison results are reported in Table 4. The circuits reported in [29, 30], or [26] are considered for comparison. If the result of a circuit is not reported in a particular paper, then the corresponding values are shown by ‘-’ in the comparison table.

The authors in [30] have introduced a genetic algorithm-based test generation method to detect the complete missing-gate fault (CMGF), partial missing-gate fault (PMGF), bridging fault, and stuck-at fault. CMGF is nothing but SMGF, i.e., complete gate is missing in the circuit. However, this method does not consider multiple missing-gate fault (MMGF). They have proposed two methods based on random and directed approached, and reported that directed approach gives better result than



**Table 3** Comparison of the complete test set with [7, 8, 31–33]

Circuit	$n$	$N$	Number of test patterns							
			Gr. [33]	BDD [33]	[32]	Gr. [7]	B&B [7]	[8]	[31]	Proposed
			MMGF	MMGF	MMGF	SMGF	SMGF	MMGF	SMGF	SMGF+MMGF
ham3_102	3	5	4	4	2	2	2	2	3 or 4	2
3_17_13	3	6	4	4	2	2	2	2	2	2
rd32_v0_66	4	4	2	2	2	2	2	2	6	2
mini_alu_167	4	6	5	5	—	—	—	—	—	3
mod10_171	4	10	6	6	—	—	—	—	—	3
hwb4_52	4	11	6	6	—	—	—	—	—	2
4-49d3	4	12	—	—	—	—	—	—	4	3
4-49tc1	4	16	—	—	3	3	3	3	—	4
hwb4tc	4	17	—	—	4	2	2	4	7	3
Xor5d1	5	4	—	—	1	1	1	1	2	1
mod5d4	5	5	—	—	—	—	—	—	4	2
mod5d1-63	5	7	4	4	—	—	—	—	—	2
mod5d1	5	8	—	—	4	1	1	4	4	4
mod5d2	5	9	—	—	2	1	1	2	2	2
mod18-10-177	5	14	6	6	—	—	—	—	—	3
hwb5-55	5	24	12	11	—	—	—	—	—	4
hwb5-53	5	55	21	21	—	—	—	—	—	5
hwb5tc	5	56	—	—	5	5	5	5	—	5
graycode6_47	6	5	4	3	—	—	—	—	—	1
ex3_229	6	7	3	3	—	—	—	—	—	3
Xor5-254	6	7	3	3	—	—	—	—	—	2
majority_239	6	8	4	4	—	—	—	—	—	3
mod5adder_128	6	15	8	7	—	—	—	—	—	3
5mod5tc	6	17	—	—	6	1	1	6	—	6
2of5d1	6	18	—	—	5	4	4	5	—	5
mod5adders	6	21	—	—	4	3	3	4	—	4
hwb6tc	6	126	43	43	9	9	8	8	—	8
2of5d2	7	12	—	—	2	2	2	2	—	2
rd53d1	7	12	—	—	3	2	2	3	—	2
rd53_137	7	16	9	8	—	—	—	—	—	3
ham7_105	7	21	9	5	—	—	—	—	—	3
ham7tc	7	24	—	—	4	4	4	4	—	4
rd53rcmg	7	30	—	—	4	4	3	4	—	4
hwb7tc	7	291	—	—	14	15	4	14	—	12
rd53d2	8	12	—	—	2	2	2	2	—	2
cma82a_208	8	22	9	8	—	—	—	—	—	4
6symd2	10	20	—	—	3	2	2	2	—	3
rd73d2	10	20	6	—	3	3	3	3	—	3
9symd2	12	28	—	—	3	3	3	3	—	3
addr_197	13	55	13	—	—	—	—	—	—	6
0410184_169	14	46	9	—	—	—	—	—	—	2
rd84-142	15	28	8	—	3	3	3	3	—	3
ham15_108	15	70	26	—	8	—	—	—	—	8
Average	7	28	9	8	4	3	3	4	4	4

**Table 4** Comparison of the complete test set and CPU time with [26, 29, 30]

Benchmark circuit	<i>n</i>	<i>N</i>	Genetic algorithm [30]			SAT based ATPG [29]			PBO based ATPG [26]			Proposed		
			Directed											
			CMGF			SMGF+SMCF			SMCF			SMGF+MMGF		
No. of faults	<i>TS</i>	Time (sec)	No. of faults	<i>TS</i>	Time (sec)	No. of faults	<i>TS</i>	Time (sec)	No. of faults	<i>TS<sub>MIN</sub></i>	Time (sec)			
one-two														
-three-v0_97	5	11	11	3	0.01	–	–	–	23	3	0.02	66	3	0.11
4gt12-v0_86	5	14	14	3	0.11	34	5	0.01	20	5	0.02	105	3	0.04
rd53_131	7	28	28	7	0.062	52	10	0.04	24	11	0.11	406	7	0.13
miller_5	8	16	–	–	–	40	5	0.02	24	4	0.03	136	4	0.01
mini-alu_84	10	20	–	–	–	47	3	0.03	27	4	0.03	210	3	0.02
sym9_148	10	210	210	1	0.29	–	–	–	756	14	1.79	22155	4	0.27
0410184_169	14	46	46	5	0.062	–	–	–	49	3	0.06	1081	2	0.19
rd84_142	15	28	28	5	0.046	77	5	0.09	49	8	0.14	406	3	0.17
4_49_7	15	42	–	–	–	103	5	0.06	61	5	0.10	903	5	0.12
ham15_108	15	70	–	–	–	195	11	0.17	125	9	0.21	2485	8	0.20
ham15_107	15	132	132	14	4.851	484	16	78.59	352	16	0.82	8778	14	0.57
hwb5_13	28	88	88	5	0.578	219	5	0.03	131	7	0.24	3916	7	0.38
rd84_313	34	104	143	19	1.33	–	–	–	–	–	–	5460	13	0.73
ham15_298	45	153	157	1	0.09	–	–	–	–	–	–	11781	10	0.93
bw_291	87	307	432	4	3.5	–	–	–	–	–	–	47278	9	1.27
Average	21	85	117	6	0.9935	139	7	8.7822	137	7	0.2975	7011	6	0.3426

random approach, so we compare our result with the directed approach.

The authors in [29] proposed two ATPG approaches to generate the complete test set using the naive-based and SAT-based approach for detecting the SMGF along with SMCF (Single Missing Control Faults), and PMGF in reversible circuits. It is reported that SAT-based provides better results in terms of number of test vectors, so SAT-based method is considered for comparison.

The authors in [26] proposed an ATPG method using a simulation-based technique, Boolean satisfiability (SAT) based, and pseudo-Boolean optimization (PBO) based approach to detect the SMCF, SMGF, and SACF in reversible circuits. The authors also mentioned that PBO based method gives better result than other two approaches, so the comparison is made with PBO based method. The approach in [26] has been considered for generating the complete test set to detect the individual fault model, whereas in our proposed work, both SMGF and MMGF are considered together.

For comparison purpose, we calculate the average number of test vectors generated, number of faults, time for test pattern generation, etc. by each method. It is observed from Table 4, that the performance of our proposed method

is better than the results reported in [29, 30], and [26], which are summarized as follows:

1. Single missing gate faults and consecutive multiple missing gate faults are considered in our proposed method, but mainly single missing gate faults are considered in the works of [29, 30], and [26].
2. Number of faults covered in our proposed method is much more than the faults covered in the works reported in [29, 30], and [26]. On an average, the number of faults considered in our proposed method is around 50X more than that of the faults considered in these three works.
3. The average number of test patterns generated by all the methods are almost similar, but our proposed method considers more number of faults.
4. Though we have considered more faults, still the average time taken to generate the test patterns by our method is less than the methods proposed in [30] and [29].

Finally, experiments are performed to check the fault coverage range of other fault models such as stuck-at faults (SAFs), partial missing gate faults (PMGFs) and crosspoint faults using the complete test set *TS* generated by our proposed method.

**Table 5** Fault coverage range with SAF, PMGF and crosspoint fault models by the proposed complete test set  $TS$ 

Benchmark circuit	$n$	$N$	No. of	No. of	No. of	Faults coverage	Faults coverage	Faults coverage
			faults	faults	faults	by $TS$ [Proposed]	by $TS$ [Proposed]	by $TS$ [Proposed]
			SAF	PMGF	Crosspoint	SAF	PMGF	Crosspoint
4gt11.84	5	3	24	4	8	79.16%	50%	12.5%
xor5d1	5	4	26	4	12	76.92%	50%	58.33%
graycode6_47	6	5	32	5	20	87.50%	80%	50%
2of5d2	7	12	76	19	53	96.05%	89.47%	58.49%
3_17_13	3	6	32	7	5	100%	100%	100%
3_17_14	3	6	32	7	5	100%	71.42%	100%
fredkin_6	3	3	24	6	0	83.33%	50%	0
rd32_v0_66	4	4	28	6	6	89.28%	66.67%	50%
decode24_v0_38	4	6	36	8	10	94.44%	87.50%	40%
miller_11	3	5	32	8	2	100%	100%	100%
ham3_102	3	5	28	6	4	92.85%	83.30%	25%
ex_1_166	3	4	22	4	4	90.90%	75%	25%
nth_Prime3	3	4	24	5	3	100%	100%	100%
peres_9	3	2	16	3	1	50%	33.33%	0
decode24_v2_43	4	6	34	7	11	100%	100%	90.90%
mini_alu_167	4	6	52	16	2	100%	93.75%	100%
mod10_171	4	10	64	18	12	100%	94.40%	91.67%
4gt11-v1.85	5	4	26	4	12	80.76%	50%	16.67%
4mod5v1_24	5	5	32	6	14	87.50%	50%	42.85%
alu-v0_26	5	6	38	8	16	86.84%	50%	50%
Average	4	5	34	6	10	89.77%	73.74%	55.57

The total number of SAFs (SA0 and SA1) in a reversible circuit is given by  $2(n + \sum_{i=1}^N g_i)$  [4], where  $g_i$  is the size of  $N$  gate of the circuit, and  $n$  is the total number of input wires. For determining the fault coverage range of PMGF fault model, the first-order PMGF is considered, i.e., only one control connection is missing at a time. The appearance crosspoint faults are considered in our experiments for crosspoint fault model. Table 5 presents the experiment results. The total number of SAFs, PMGFs and crosspoint faults are shown in Columns 4, 5 and 6, respectively. In Table 5, it is observed that SAFs, PMGFs and crosspoint faults coverage is 100% for the reversible benchmark circuits 3\_17\_13, miller\_11, and nth\_prime3 by the test set  $TS$  generated by our proposed method.

A total of 20 benchmark circuits are considered, out of these, only for the circuit peres\_9 fault coverage of SAFs is 50%, and the fault coverage of SAFs is more than 75% for rest of the circuits. The fault coverage for PMGFs is 50% for 30% of the circuits and the fault coverage is more than 65% for rest of the circuits. In the crosspoint fault model, the highest deterioration of fault coverage is 12.50% for the circuit 4gt11.84, and for the circuit peres\_9, the test set  $TS$  is unable to detect any faults. The average

fault coverage for stuck-at faults, partial missing gate faults and appearance crosspoint faults are 89.77%, 73.74% and 55.57%, respectively.

## 5 Conclusion

This paper presents a scheme to generate the complete test set for detecting single and any number of consecutive multiple missing gate faults in the  $k$ -CNOT based reversible circuits. The complete test set generation method is twofold. First, the test vectors for SMGFs is generated by applying the local test pattern to all the gates of reversible circuits by using the reverse simulation method. Secondly, based on the complete test set  $TS_{SMGF}$  for SMGFs and the structure of the  $k$ -CNOT based circuit, the test set  $TS$  for detecting all the MMGFs is constructed. The generated complete test set is not a minimal one. For achieving the minimality, the row and column fault covering table is constructed, which is formulated as an ILP problem. Experimental results show that the size of test set  $TS_{MIN}$  generated by our proposed method is smaller or similar as compared to the methods available in the literature and covers more faults by maintaining 100% fault coverage. The fault coverage

ranges for the SAFs, PMGFs and appearance crosspoint faults with our generated complete test set are also analyzed and checked using experimental results. By looking into the correlation between different fault models, there is a possibility to reconstruct the generated complete test set to cover these fault models such that all the possible faults in reversible circuits can be detected by a single test set.

## References

- Schaller RR (1997) Moore's law: past, present, and future. *IEEE Spectr* 34(6):52–59. <https://doi.org/10.1109/6.591665>
- Landauer R (1961) Irreversibility and heat generation in the computing process. *IBM J Res Dev* 5(8):183–191
- Bennett C (1973) Logical reversibility of computation. *IBM J Res Dev* 17(6):525–532. <https://doi.org/10.1147/rd.176.0525>
- Patel KN, Hayes JP, Markov IL (2004) Fault testing for reversible circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 23(8):1220–1230
- Rahaman H, Kole DK, Das DK, Bhattacharya BB (2007) Optimum test set for bridging fault detection in reversible circuits. In: Proc. of 16th asian test symposium (ATS'07). IEEE, pp 125–128
- Handique M, Biswas S, Deka JK (2019) Test generation for bridging faults in reversible circuits using path-level expressions. *J Electron Test* 35(4):1–17
- Hayes JP, Polian I, Becker B (2004) Testing for missing-gate faults in reversible circuits. In: Proc. of 13th asian test symposium. IEEE, pp 100–105
- Polian I, Fiehn T, Becker B, Hayes JP (2005) A family of logical fault models for reversible circuits. In: Proc. of 14th asian test symposium. IEEE, pp 422–427
- Zhong J, Muzio JC (2006) Analyzing fault models for reversible logic circuits. In: Proc. of IEEE congress on evolutionary computation, CEC 2006, pp 2422–2427
- Jha NK, Gupta S (2003) Testing of digital systems. Cambridge University Press, Cambridge
- Rice J (2013) An overview of fault models and testing approaches for reversible logic. In: Proc. of IEEE pacific rim conference on communications, computers and signal processing (PacRim). IEEE, pp 125–130
- Xiaojun M, Huang J, Metra C, Lombardi F (2009) Detecting multiple faults in one-dimensional arrays of reversible qca gates quantum. *J Electron Test* 25:39–54
- Ma X, Huang J, Metra C, Lombardi F (2008) Reversible gates and testability of one dimensional arrays of molecular qca. *J Electron Test* 24(1-3):297–311
- Feynman RP (1961) Quantum mechanical computers. *Optics InfoBase, Optics News* 11(3):11–20
- Toffoli T (1980) Reversible computing. *Lecture Notes in Computer Science (LNCS)* 85. Springer, Berlin
- Fredkin E, Toffoli T (1982) Conservative logic. *Int J Theor Phys* 21:219–253
- Maslov D (2015) Reversible logic synthesis benchmarks page. Online: <http://webhome.cs.uvic.ca/dmaslov/>
- Loss D, DiVincenzo DP (1998) Quantum computation with quantum dots. *Phys Rev A* 57(1):120
- Maslov DA (2003) Reversible logic synthesis. Ph.D. thesis, Fredericton, N.B., Canada. AAINQ98874
- Fang-ying X, Han-wu C, Wen-jie L, Zhi-giang L (2008) Fault detection for single and multiple missing-gate faults in reversible circuits. In: Proc. of IEEE congress on evolutionary computation, (IEEE World Congress on Computational Intelligence), CEC 2008, pp 131–135
- Rahaman H, Kole DK, Das DK, Bhattacharya BB (2008) On the detection of missing-gate faults in reversible circuits by a universal test set. In: Proc. of 21st international conference on VLSI design, VLSID. IEEE, pp 163–168
- Kole DK, Rahaman H, Das DK, Bhattacharya BB (2010) Derivation of optimal test set for detection of multiple missing-gate faults in reversible circuits. In: Proc. of 19th asian test symposium. IEEE, pp 33–38
- Zamani M, Tahoori MB, Chakrabarty K (2012) Ping-pong test: compact test vector generation for reversible circuits. In: Proc. of 30th VLSI test symposium (VTS). IEEE, pp 164–169
- Rahaman H, Kole DK, Das DK, Bhattacharya BB (2011) Fault diagnosis in reversible circuits under missing-gate fault model. *Computers & Electrical Engineering* 37(4):475–485
- Zhang H, Wille R, Drechsler R (2010) SAT-based ATPG for reversible circuits. In: Proc. of 5th international design and test workshop. IEEE, pp 149–154
- Wille R, Zhang H, Drechsler R (2011) ATPG for reversible circuits using simulation, Boolean satisfiability, and pseudo Boolean optimization. In: Proc. of IEEE computer society annual symposium on VLSI. IEEE, pp 120–125
- Zhang H, Frehse S, Wille R, Drechsler R (2011) Determining minimal testsets for reversible circuits using Boolean satisfiability. In: Proc. of IEEE Africon'11. IEEE, pp 1–6
- Mondal J, Das DK, Kole DK, Rahaman H (2013) A design for testability technique for quantum reversible circuits. In: Proc. of East-West design & test symposium (EWDTS 2013). IEEE, pp 1–4
- Kole A, Wille R, Datta K, Sengupta I (2017) Test pattern generation effort evaluation of reversible circuits. In: Proc. of international conference on reversible computation. Springer, Cham, pp 162–175
- Nagamani AN, Anuktha SN, Nanditha N, Agrawal VK (2018) A genetic algorithm-based heuristic method for test set generation in reversible circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 37(2):324–336
- Mondal B, Kole DK, Das DK, Rahaman H (2014) Generator for test set construction of smgf in reversible circuit by boolean difference method. In: Proc. of 23rd asian test symposium. IEEE, pp 68–73
- Nagamani AN, Ashwin S, Abhishek B, Agrawal VK (2016) An exact approach for complete test set generation of toffoli-fredkin-peres based reversible circuits. *J Electron Test* 32(2): 175–196
- Surhonne AP, Chattopadhyay A, Wille R (2017) Automatic test pattern generation for multiple missing gate faults in reversible circuits. In: Proc. of international conference on reversible computation. Springer, Cham, pp 176–182
- Zhang B, Agrawal VD (2017) Three-stage optimization of pre-bond diagnosis of TSV defects. *J Electron Test* 33(5):573–589
- Wille R, Große D, Teuber L, Dueck GW, Drechsler R (2008) RevLib: an online resource for reversible functions and reversible circuits. In: Proc. of 38th international symposium on multiple valued logic (ismvl 2008). IEEE, pp 220–225

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Mousum Handique** is a PhD student in the Dept. of CSE, IIT Guwahati. His research interests include VLSI Testing and Reversible computing. He has published 6 research papers. He is the member of IEEE.

**Jantindra Kumar Deka** received the B.E. degree in Electronics from the Motilal Nehru National Institute of Technology Allahabad in the year 1988. He has completed his M.Tech in Computer Science and Information Technology from the Department of Computer Science and Engineering, Indian Institute of Technology Kharagpur in the year 1993. He obtained his PhD from the Department of Computer Science and Engineering, Indian Institute of Technology Kharagpur in the year 2001. He then joined the Department of Computer Science and Engineering, Indian Institute of Technology Guwahati and is currently a Professor. He is engaged with academic as well as industry-sponsored research related to VLSI Testing and Design for Testability. His research interests include Formal Modeling and Verification, CAD for VLSI and Embedded Systems (Design, Testing and Verification), Data Mining. He has published more than 40 research papers. He is a member of IEEE.

**Santosh Biswas** received the B.E. degree from the National Institute of Technology Durgapur in the year 2001. He has completed his MS from the Department of Electrical Engineering, Indian Institute of Technology Kharagpur with highest institute CGPA in the year 2004. He obtained his PhD from the Department of Computer Science and Engineering, Indian Institute of Technology Kharagpur in the year 2008. He joined the Department of Computer Science and Engineering, Indian Institute of Technology Guwahati in 2009 and is currently an Associate Professor. At present he is on deputation as HoD EECS dept. in IIT Bhilai. He has been involved in several Research Projects sponsored by Industry and Government agencies. He is engaged with academic as well as industry-sponsored research related to VLSI Testing and Design for Testability. His research interests include VLSI Testing and Design for Testability, Fault Tolerance, Network Security, Discrete-event systems and Embedded Systems. He has published about 150 research papers. He is a member of IEEE.