A Fault Detection Method for Missing Gate Faults in Reversible Circuits using Binary to Gray Code Conversion

1st Dimpimoni Kalita
Department of CSE, TSSOT
Assam University
Silchar-788011, Assam, India
monidimpi5@gmail.com

2nd Mousum Handique
Department of CSE, TSSOT
Assam University
Silchar-788011, Assam, India
mousum.smit@gmail.com

***Abstract*—Reversible computing has a remarkable ability to reduce heat dissipation in computing machinery. It can be broadly applied in various fields, which include Digital signal processing, Cryptography, DNA computing, Network congestion, Database transactions, Quantum computing, etc. Fault detection is a complicated and demanding problem in reversible circuits. Fault detection is an essential process in the field of testing to ensure the reliability and integrity of the circuit. This paper proposes a straightforward approach for Single Missing Gate Fault (SMGF), Multiple Missing Gate Fault (MMGF), Repeated Gate Fault (RGF) and Partial Missing Gate Fault (PMGF) under the Missing Gate Fault (MGF) model. The method includes the process of binary to gray code conversion in order to determine the total number of test vectors to detect the respective MGFs. Experimental results are performed on reversible benchmark circuits to evaluate the number of test vectors required to recognize all the MGFs. The comparative analysis of the proposed work with the existing work is also presented.**

***Index Terms*—Reversible circuit, test vector, gray code, fault detection**

I. Introduction

In today's technological world, computing requirements and applications of favorable speed are growing day by day. Traditional or classical computing is difficult to deal with such increasing demand because of these native constraints, such as dissipation of heat, low packaging density, restrictions on speed of light and many other factors [1]. The traditional computation process functions as an irreversible computation process, where the output cannot be determined by the input. Due to these limitations, reversible computation came into existence. It can be implemented on reversible circuits, which are formed by the combination of reversible gates which follows the two essential reversible logic functions, i.e., an equal number of inputs and outputs and one-to-one correspondence of all the inputs with the outputs. The operation to be reversible is needed to be performed by maintaining three conditions (a) there must be unique relation between the inputs and outputs of the reversible circuits (b) it should be possible to get the inputs that are derived from the corresponding outputs (c) the circuit must assure the low-power consumption with less heat dissipation [2].

A fault is an error or any undesirable feature that occurs in a circuit, which affects the utility and performance of the circuit that may be intended to last for an indefinite period of time or for a limited interval of time. Thus, testing is one of the essential procedures to identify the imperfections or errors that may occur in the circuit to maintain the reliability and steady functioning of the circuit [3].

In this paper, the identification of MGFs by the fault detection method is considered for the reversible circuits consisting of gates under the Multiple-Controlled-Toffoli (MCT) library. The method we have put forward here identifies all the SMGF, MMGF, RGF, and PMGF present in the circuit. The technique starts with the simple conversion of test vectors from binary to gray code. The level-wise computation is performed in every gate of the circuit using the generated gray codes. The most substantial contribution of this work is that the proposed technique is suitable for determining all the possible missing gate faults.

The remaining portion of the paper includes a brief background that comprises reversible logic functions, reversible gates, reversible circuits, missing gate fault models, and some existing works in Section II. The proposed fault detection method is discussed in Section III. In Section IV and Section V, the experimental results for some reversible benchmark circuits and conclusions are put forward, respectively.

II. Background

This section includes the preliminaries of reversible logic functions, reversible gates, reversible circuits, and missing gate fault models. Also, we provide certain existing works that are relevant to our proposed work.

A. Reversible Logic Function

A function is termed as a reversible function if it is bijective in nature, i.e., it includes both injective and surjective operations. The law of reversible logic function constitutes a reversible circuit. Thus, a reversible circuit consists of an equal number of inputs and outputs and there is a one-to-one relation between the inputs and the outputs. A reversible function is realized by reversible gates of the circuit that contain equal input and output variables. Let f be a function and define as f: $X \rightarrow Y$. The function f is surjective (onto) if f (X)=Y and the function f is injective (one-to-one) if $X \neq X'$ implies $f(X) \neq f(X')$, then X=X'.

B. Reversible Gates

A reversible gate under the MCT library contains two essential elements, a control point (•) and a target point ($\oplus$), as illustrated in Fig. 1. The operation of each gate is based on reversible logic function, which includes an identical number of inputs and outputs; it also maintains a distinct one-to-one relation. In our proposed work, we have considered the circuits belonging to the MCT or k-CNOT library, which is one of the most widely used reversible gate libraries that are composed of the three basic reversible gates NOT gate, Feynman gate and Toffoli gate, as shown in Fig. 1.

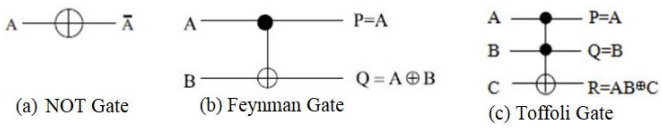

Fig. 1. Structure of basic reversible gates

A simple and basic reversible gate is the NOT gate that consists of one input and one output (1×1). The function of NOT gate is A=A' and the depiction of NOT gate is shown in Fig. 1 (a). The 2×2 Feynman gate [4] as shown in Fig. 1 (b), also called controlled-not (CNOT). It is a 2-inputs and 2-outputs reversible gate that realizes as P=A, Q=$A \oplus B$, where A and B are the inputs of the gate and P and Q are the outputs of the gate. The 3×3 Toffoli gate [5] is also known as 3×3 Feynman gate or controlled-controlled-not (CCNOT) gate, which is shown in Fig. 1 (c). It is described by the three equations P=A, Q=B, R= $AB \oplus C$, where A, B and C are the inputs of the gate and P, Q and R are the outputs of the gate.

C. Reversible Circuits

A circuit is considered as a reversible if it is based on a bijective Boolean function and consists of reversible gates. The reversible circuit follows cascade rules, i.e., the output of one gate will act as an input to the next consecutive gate. Thus the operation of the reversible circuit depends only on the primary input and the primary output [6]. Moreover, the reversible circuit does not follow fanout and feedback connections [7].

D. Missing Gate Fault (MGF) Model in Reversible Circuits

A fault is an imperfection that occurs in the circuit, which hampers the circuit from operating the functional behavior normally. The complexity of testing is rationalized by introducing the fault model that determines the presence of faults in the circuit. A fault model recognizes the testing objective and evaluates the type of fault that is appeared in the circuit [8]. Various missing gate faults can be classified as SMGF [8], MMGF [8], RGF [9], and PMGF [9]. The illustration of these faults is described with the help of Example 1.

Example 1. *Let us consider the reversible benchmark circuit $Peres$, which consists of 3-input and 3-output and 2 k-CNOT gates, as shown in Fig. 2. Here, all the SMGF, MMGF, RGF, and PMGF are demonstrated. When one k-CNOT gate disappears from the circuit completely, it is called an SMGF. In Fig. 2 (a), the first gate of the $Peres$ reversible benchmark circuit disappears, causing a faulty output as $\langle$ 0 0 1$\rangle$ for the test vector $\langle$ 0 1 1$\rangle$ instead of fault free output $\langle$ 1 0 1$\rangle$. The gate count for a reversible circuit is the total number of SMGFs [8], [9]. When a set of consecutive gates in a reversible circuit get vanishes, it is known as an MMGF. For a circuit with N number of gates, there are N (N-1)/2 possible number of MMGFs [9]. Hence, there is only one MMGF for reversible circuit Peres and it is shown in Fig. 2 (b), where the test vector $\langle$ 0 1 1$\rangle$ results in a faulty output $\langle$ 0 1 1$\rangle$ instead of fault free output $\langle$ 1 0 1$\rangle$. The RGF is the unwanted repetition of a gate in a reversible circuit. The number of RGFs can be infinite. Any reversible gate in the circuit can repeat for any number of times [9]. The RGF is shown in Fig. 2 (c), where the first gate of the circuit repeats for one more time, due to which it results in a faulty output $\langle$ 0 0 1$\rangle$ for the test vector $\langle$ 0 1 1$\rangle$ instead of $\langle$ 1 0 1$\rangle$. The PMGF corresponds to the disappearance of one or more control points from a gate in a circuit. The number of PMGFs refers to the number of controls which are disappeared [9]. The occurrence of a PMGF is shown in Fig. 2(d), where the control point of the second gate of the circuit gets disappears, which results in a faulty output $\langle$ 0 1 0$\rangle$ for the test vector $\langle$ 0 0 0$\rangle$ on the contrary of fault free output $\langle$ 0 0 0$\rangle$.*

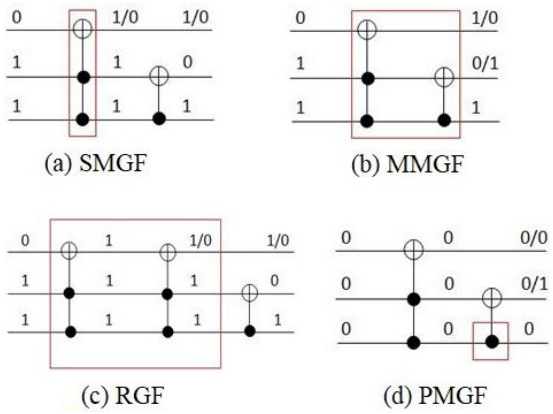

Fig. 2. Illustration of $Peres$ circuit for SMGF, MMGF, RGF and PMGF

E. Related Work

In 2004, the authors in [8] described the testing requirements of reversible circuits composed of k-CNOT gates with respect to the MGF model. This paper proposed an offline testing method using the Design For Testability (DFT) concept that can detect the MGFs. This method introduced the testing of reversible circuits by adding a single line that contains control points for one or more CNOT gates which require modification of the original reversible circuit. This technique requires one to $\lceil N/2 \rceil$ test vectors for detecting all possible MGFs. In 2008, the authors in [10] introduced a method to get a complete test set concerning SMGF and MMGF in reversible circuits. This fault detection method includes dividing the actual circuit into sub-circuits to get the complete test set. Next, the method used a set covering method using linear programming to get a minimal complete test set. In 2012, the authors in [11] presented a compact test vector generation method named as Ping Pong method for detecting SMGF and RGF in reversible circuits. This

method includes four steps; firstly, for each SMGF and RGF, the coverage list of faults is determined, then for simulating the faults a possible order is explored, followed by cycle dropping and selection of the best sequence. In 2014, the authors in [12] proposed a fault detection technique for detecting SMGF by using a Boolean generator computed by a function to get a Boolean expression for generating a test set to detect SMGFs of the reversible circuit. In 2016, Mondal et al. [13] presented a technique for detecting two faults, SMGF and PMGF, that comprises three phases of circuit augmentation, fault localization and fault detection. In the circuit augmentation process, a set of Toffoli gates and one extra line is added to make the circuit testable. The fault is detected after the circuit is augmented by applying test vectors. In 2017, the authors in [14] provided a method for generating test patterns for identifying the MMGFs where only two missing gates are considered. The method initiates the generation of SMGFs. Those are analyzed in the form of a Binary Decision Diagram (BDD) structure. Finally, for the identification of MMGFs, the test patterns are obtained where dependency analysis is done between the two gates. In 2021, the authors in [15] proposed a technique for fault location and detection of the MCT-based reversible circuits considering all the MGFs. Here, the primary circuit and its complement are considered to find the faults. A comparator is used to detect the faults where the input of the reversible circuit and the output of complemented reversible circuit are compared.

As per the above discussion, we have observed that there is limited work for evaluating all the missing gate faults by using only a single technique. Many existing works show a complex procedure to detect the faults, which requires re-construction or modification of the original circuit, increasing the overhead circuit cost. Thus, we are motivated to propose a straightforward technique to detect all the MGFs for the SMGF, MMGF, RGF and PMGF without any modification of the original reversible circuit.

III. Proposed Fault Dectection Method

The method that we have proposed initiates with the generation of a binary encoded pattern based on the count of the input lines present in the reversible circuit. The process flow diagram of the proposed method is depicted in Fig. 3. At first, we assume the circuit of any fault's inexistence. For this purpose, a straightforward technique is implemented for generating the test vectors. All the patterns present in binary form are converted to respective gray codes at the next level. Then, the obtained gray codes are fed as input to the circuit's first gate, and the level-wise computation process continues for all the gates present in a given circuit. Finally, the output is obtained in gray code at the last level of the circuit. The entire missing gate faults (SMGF, MMGF, PMGF, and RGF) are considered for the respective reversible circuits. After inserting the fault, the same conversion process is applied to the circuit consisting of faults, and by performing level-wise computation, the final output is noted after the primary gate of the circuit. The primary output responses are compared with the output obtained from the circuit without fault, which helps to determine the faults in the circuit. For all the respective MGFs, the test vectors that are capable of identifying all the MGFs are noted and counted separately. The significance of converting binary code to gray code is considered a well-defined conversion approach based on the XORed operation. Moreover, the reversible gate under the MCT library also operates on the XORed operation for the target connection. Due to this reason, we have applied the binary to the gray code conversion process in our proposed method. Each component of the process flow diagram is explained below:

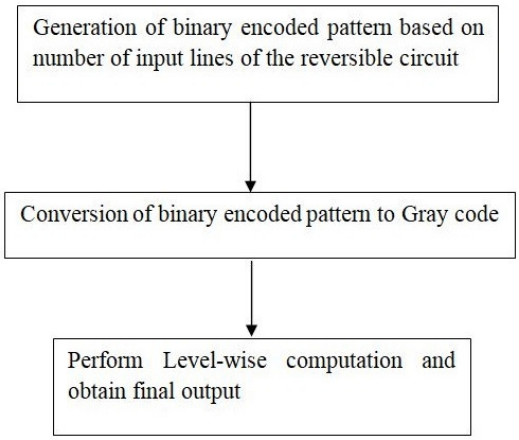

Fig. 3. Process flow diagram of fault detection method

1) **Generation of Binary Encoded Pattern based on number of lines of the reversible circuit:** All the possible binary encoded patterns are generated based on the number of input lines in the circuit. If n is the number of input lines present in the circuit, then the total number of binary encoded patterns for the circuit will be 2^n.
2) **Conversion process of Binary Encoded Pattern to Gray code:** The generated test patterns are converted from binary encoded patterns to gray code by the binary to gray code conversion technique. This technique is implemented as there is a unique relation between binary and gray code, which helps in maintaining the unique relation between the input and the output. A binary code can be converted to a gray code by a straightforward method. Firstly, the Most Significant Bit (MSB) of the binary code will be exactly similar to the MSB of the gray code. The second bit of gray code can be obtained from the binary code by performing the XOR operation of MSB of binary code and the second bit of binary code. Thus, all the other bits of gray code can be obtained by performing an XOR operation on the current and previous index bit of binary code. For example, if a binary code is 1001, its gray code will be 1101. The leftmost bit, i.e., MSB of the binary code would be precisely similar to the gray code. Therefore it is 1 and the second bit of gray code is obtained by the XOR operation of the first and second bit of binary code; thus, it is 1. The same process continues for all the other bits and thus, a binary code is converted to respective gray codes, as shown in Fig. 4.
3) **Perform Level-wise Computation and obtain final output:** The converted gray code is first fed into the

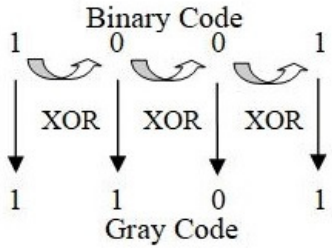

Fig. 4. Illustration of Binary to Gray Conversion process

first gate of the reversible circuit. Then, the second gate uses the output of the first gate as an input to perform the level-wise computation. The same process continues until the final gate of the reversible circuit that provides the primary output. This process is performed level-wise in order to achieve the primary output for each gate in the circuit.

The process of forming all the feasible test vectors for detecting all MGFs is given in Algorithm 1. Also, the complete flow of Algorithm 1 is demonstrated with the help of Example 2

Algorithm 1: COMPLETE TEST SET GENERATION FOR MGFS USING BINARY TO GRAY CODE CONVERSION PROCESS

Input: A reversible circuit consisting of N number of reversible k-CNOT gates $G_0, G_1, \ldots, G_{N-1}$ and n number of input lines.
$C_{fault-free}$: Store the structure of the circuit without consisting any fault
C_{faulty}: Store the structure of the faulty circuit after injecting all MGFs.
Output: Generation of all feasible test vectors for detecting all possible MGFs in a given circuit.

1 Consider the circuit $C_{fault-free}$
2 **for** $i \leftarrow 0$ ***to*** $(n-1)$ **do**
3 Generate all the possible binary encoded patterns
4 Convert the generated pattern to Gray code
5 **for** $j \leftarrow 0$ ***to*** $(N-1)$ **do**
6 Perform computation on $G_0, G_1, \ldots, G_{N-1}$
7 Obtain final output after G_{N-1} gate operation
8 Insert MGFs to the circuit
9 Repeat Step 2
10 Fault simulation with each test vector from the initial level of the circuit and compare the primary output of $C_{fault-free}$ and C_{faulty}
11 Obtain the required test vectors for all the MGFs from Step 10

Example 2. *To explain the complete flow in Algorithm 1, we consider the benchmark circuit $Fredkin$-6, as shown in Fig. 5. In **Step 1**, consider a fault-free circuit $C_{fault-free}$ for the $Fredkin$-6 that comprises of three input lines (i.e., n=3) and three k-CNOT gates (i.e., N=3). In **Step 2**, we generate all the possible binary encoded pattern 0 to (2^n-1) for the $Fredkin$-6 circuit. Thus, 8 binary encoded patterns*

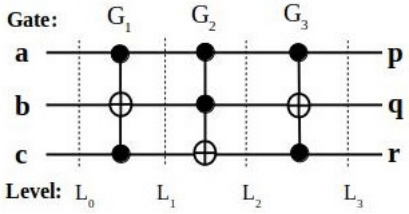

Fig. 5. Reversible circuit $Fredkin$-6

TABLE I
LEVEL-WISE COMPUTATION FOR THE FAULT-FREE $Fredkin$-6 CIRCUIT

L_0	$Gray$	L_1	L_2	L_3
000	000	000	000	000
001	001	001	001	001
010	011	011	011	011
011	010	010	010	010
100	110	110	111	101
101	111	101	101	111
110	101	111	110	110
111	100	100	100	100

*are generated for the circuit, which is shown in the first level L_0 of Table I. The generated binary pattern conversion to their respective gray code is shown in the second column and the first gate of the circuit is computed using the gray code, which is shown in the third column (i.e., level L_1) of Table I. Similarly, after computing the second and third gate, the level L_2 and L_3 are obtained, respectively, as shown in the fourth and fifth columns of Table I. The final output is obtained after computing the last gate, i.e., the last level L_3 of the circuit, which is shown in column 5 of Table I. This process continues from the **Step 3** to **Step 7**. In **Step 8**, all the faults SMGF, MMGF, RGF, and PMGF under the MGF model are injected into the circuit $Fredkin$-6. Each type of fault is illustrated in Fig. 6. Since there are three gates in the $Fredkin$-6 circuit. Therefore, 3 numbers of SMGFs occurred that are enumerated as f_{G1}, f_{G2}, and f_{G3}. As per the definition of MMGF, there is total 3 numbers of MMGFs occur in the $Fredkin$-6 circuit, which is enumerated as f_{G1G2}, f_{G2G3} and f_{G1G2G3}. Here, we assume that the repetition of gates occurs one more time after existing the actual gates; then, there is one more level is created for each k-CNOT gate, as shown in Fig. 6 (c). Therefore, the total number of RGFs is*

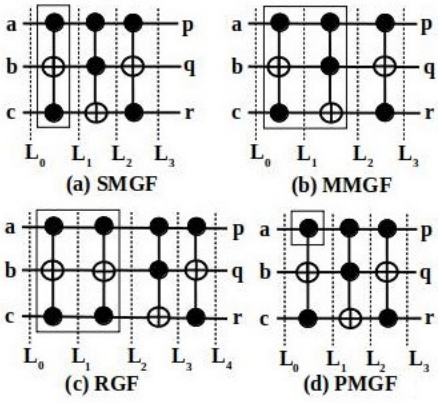

Fig. 6. Illustration of $Fredkin$-6 for SMGF, MMGF, RGF and PMGF

TABLE II
LEVEL-WISE COMPUTATION OF THE FAULTY $Fredkin$-6 CIRCUIT FOR DIFFERENT MGFS

Initial Level (L_0)	Gray Code	SMGF			MMGF			RGF				PMGF		
		L_1	L_2	L_3	L_1	L_2	L_3	L_1	L_2	L_3	L_4	L_1	L_2	L_3
000	000	000	000	000	000	000	000	000	000	000	000	000	000	000
001	**001**	001	001	001	001	001	001	001	001	001	001	**011**	**011**	**011**
010	**011**	011	011	011	011	011	011	011	011	011	011	**001**	**001**	**001**
011	010	010	010	010	010	010	010	010	010	010	010	010	010	010
100	**110**	110	111	101	**110**	**110**	**110**	**110**	**110**	**110**	**110**	110	111	101
101	**111**	**111**	**110**	**110**	**111**	**111**	**101**	101	111	101	111	101	101	111
110	**101**	**101**	**101**	**111**	**101**	**101**	**111**	**111**	**101**	**111**	**101**	111	110	110
111	100	100	100	100	100	100	100	100	100	100	100	100	100	100

TABLE III
ILLUSTRATION OF FAULT COVERAGE FOR THE $Fredkin$-6 CIRCUIT

Test Vectors	SMGF			MMGF			RGF			PMGF					
	f_{G1}	f_{G2}	f_{G3}	f_{G1G2}	f_{G2G3}	f_{G1G2G3}	f_{RG1}	f_{RG2}	f_{RG3}	f_{C1}	f_{C2}	f_{C3}	f_{C4}	f_{C5}	f_{C6}
000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
001	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0
010	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0
011	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
100	0	1	1	1	1	1	1	1	1	0	1	0	0	0	0
101	1	0	1	1	1	0	0	0	1	0	0	0	1	0	0
110	1	1	0	1	1	1	1	1	0	0	0	0	0	0	1
111	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1

*3 for the total k-CNOT gates present in $Fredkin$-6 circuit and the RGFs are enumerated as f_{RG1}, f_{RG2} and f_{RG3}. For evaluating the number of PMGFs, we consider each control point is missing for one time (i.e., first-order PMGF), then 6 number of PMGFs occur, which are enumerated as f_{C1}, f_{C2}, f_{C3}, f_{C4}, f_{C5} and f_{C6}. In **Step 9**, after injecting all the MGFs to the circuit $Fredkin$-6 (i.e., C_{faulty}), the similar process is repeated from the Step 2, i.e., the faulty circuit is computed level-wise till the last gate, as shown in Table II. In **Step 10**, the fault simulation process is performed for each test vector at the initial level of the circuit and compared with the primary output between the $C_{fault-free}$ and C_{faulty} circuits. In **Step 11**, all the test vectors are extracted to construct the complete test set for MGFs, which are derived from their corresponding distinct primary outputs of the $C_{fault-free}$ and C_{faulty} circuits. The complete test sets are* $\{\langle 1\ 0\ 1\rangle, \langle 1\ 1\ 0\rangle, \langle 1\ 0\ 0\rangle\}$, $\{\langle 1\ 0\ 0\rangle, \langle 1\ 0\ 1\rangle, \langle 1\ 1\ 0\rangle\}$, $\{\langle 1\ 0\ 0\rangle, \langle 1\ 1\ 0\rangle, \langle 1\ 0\ 1\rangle\}$ *and* $\{\langle 0\ 0\ 1\rangle, \langle 0\ 1\ 0\rangle, \langle 1\ 0\ 0\rangle, \langle 1\ 1\ 1\rangle, \langle 0\ 1\ 1\rangle, \langle 1\ 0\ 1\rangle, \langle 1\ 1\ 0\rangle\}$ *for detecting all the MGFs in $Fredkin$-6 circuit. The detailed illustration of the fault coverage table for all SMGFs, MMGFs, RGFs, and PMGFs under the MGF fault model using the Binary to Gray conversion process for the $Fredkin$-6 circuit is shown in Table III.*

IV. EXPERIMENTAL RESULTS

The experimental work for the proposed method is performed on reversible benchmark circuits, which is elaborated in Table IV. The considered reversible benchmark circuits available in [16] that are given in the first column of Table IV. The second and third column of the table gives the number of gates (N) and the number of input lines (n) present in the circuit, respectively. The fourth column of Table IV shows the respective number of SMGFs, MMGFs, RGFs, and PMGFs. The SMGF count is the same as the number of gates present in the circuit (N) and the MMGF count can be calculated by $[N(N-1)/2]$. RGF count is the same as the number of gates, i.e., one gate is added in extra in the existing circuits. The PMGF count is determined based on the number of control points present in the circuit. Column 5 of Table IV indicates the count of test vectors necessary for identifying all considered MGFs after applying the proposed method to the respective circuits.

TABLE IV
EXPERIMENTAL RESULTS FOR VARIOUS BENCHMARK CIRCUITS FOR THE FAULT DETECTION OF MGFS

Benchmark Circuit	N	n	Total No. of MGFs				Total No. of Test Vectors
			SMGF	MMGF	RGF	PMGF	MGFs
fredkin-6	3	3	3	3	3	6	7
peres	2	3	2	1	2	3	4
ham3	5	3	5	10	5	6	3
nth_prime	4	3	4	6	4	5	6
exe	4	3	4	6	4	4	6
3_17_13	6	3	6	15	6	7	7
miller-11	5	3	5	10	5	8	5
rd-32	5	4	5	10	5	6	12
toffoli-double4	2	4	2	1	2	4	3
mod10-176	7	4	7	21	7	13	14
mod5d4	5	9	5	10	5	8	8
rd32d1	4	4	4	6	4	5	6
mod5d2	9	5	9	36	9	16	5
hwb4d1	17	4	17	136	17	27	7
4_49d3	12	4	12	66	12	15	4
mod5d1	8	5	8	28	8	12	4
xor5d1	4	5	4	6	4	4	5
Average							6.24

We have done our experimental work using moderate-sized reversible circuits. We obtain the test vectors that are responsible for detecting the MGFs using the binary to gray code conversion technique. After examining the circuits, we have observed that the average number of test vectors required is 6.24 for finding all MGFs in Table IV. The final outcome of the proposed method includes the complete fault coverage of all MGFs. The reversible benchmark circuit $mod10$-176 has a higher number of test vector (14 nos.) requirements for detecting all the MGFs from the proposed method, which could be improved after applying a minimization process.

TABLE V
COMPARISON RESULTS OF THE NUMBER OF FAULTS AND THE COMPLETE TEST SET WITH [12]

Benchmark Circuit	N	n	Total No. of Faults [12]	Total No. of Faults [Proposed]	Total No. of Test Vectors [12]	Total No. of Test Vectors [Proposed]
			SMGF	(SMGF+MMGF+RGF+PMGF)	SMGF	(SMGF+MMGF+RGF+PMGF)
ham3tc	5	3	5	26	4	3
rd32	4	4	4	26	6	12
xor5d1	4	5	4	18	2	5
3-17tc	6	3	6	34	2	7
mod5d1	8	5	8	56	2	4
4_49d3	12	4	12	105	4	4
hwb4d1	17	4	17	197	7	7
mod5d2	9	5	9	70	2	5
rd32d1	4	4	4	19	6	6
mod5d4	5	9	5	28	4	8
Average			7.40	57.90	3.90	6.10

Table V shows the comparative results of the proposed method with the existing work of [12]. The first, second and third columns of Table V are the same presentation, as mentioned in Table IV. Columns 4 and 5 indicate the average count of the total faults for the existing work [12] and the proposed method, respectively. Columns 6 and 7 in Table V present the entire count of test vectors required for detecting the faults by the existing work [12] and the proposed work. The authors in [12] developed a Boolean expression based generator with the help of the Boolean difference technique. The complete test set is constructed for detecting the SMGFs using the Boolean expression only. However, the length of the Boolean expression is more for large-sized circuits. Due to this reason, the derived Boolean generator needs to be simplified. In our proposed method, we use the binary to gray code conversion technique and perform the level-wise computation to detect SMGFs along with MMGFs, RGFs and PMGFs with 100% fault coverage. Thus, the method that we have used covers faults to a great extent than the work of [12]. The average number of total faults is 57.90, covered by our proposed method, as shown in Column 5 of Table V. Whereas the work in [12] covered only SMGFs and the average number of faults is 7.40, as shown in Column 4 of Table V. The highest increase number of test vectors is 71.43% for the circuit 3-17tc by our proposed method, as analyzed with existing work [12]. An equal number of test vectors is occurred to detect the faults for the circuits 4_49d3, $hwb4d1$ and $rd32d1$ by both proposed and existing work [12]. As per the results reported in Table V, though the test vector requirement of our proposed method is found to be more, as compared to the work in [12], but proposed method is capable of covering all the possible faults under the MGF model.

V. CONCLUSION

In this work, we have investigated fault testing methods for the SMGF, MMGF, RGF and MMGF under the fault model MGF in reversible circuits. We have developed a straightforward fault detection method that converts binary to gray codes to detect all the MGFs in the reversible circuits by generating the test vectors. We have performed experimental work on small and medium reversible circuits. The method will be extended in the future to minimize the number of test vectors that are obtained from the proposed method. The experimental result will also be extended to large reversible circuits.

REFERENCES

[1] M. P. Frank, "The physical limits of computing," Computing in Science & Engineering, vol. 4, no. 3, pp. 16–26, 2002

[2] M. B. Plenio and V. Vitelli, "The physics of forgetting: Landauer's erasure principle and information theory," Contemporary Physics, vol. 42, no. 1, 2001, pp. 25–60.

[3] H. M. Gaur, A. K. Singh and U. Ghanekar, "Offline testing of reversible logic circuits: an analysis," Integration, 62, 2018, pp. 50–67, 2018.

[4] R. P. Feynman, "Quantum mechanical computers," Optics news, Vol. 11, no. 2, pp. 11–20, 1985.

[5] T. Toffoli, "Reversible computing," in J. Bakker, J. Leeuwen, (eds.) ICALP 1980, LNCS, Springer, Heidelberg, vol. 85, pp. 632–644, 1980.

[6] B. Mondal, P. Das, P. Sarkar and S. Chakraborty, "A comprehensive fault diagnosis technique for reversible logic circuits," Computers & Electrical Engineering, vol. 40, no. 7, pp. 2259–2272, 2014.

[7] M. A. Nielson and I. L. Chuang, Quantum Computation and Quantum Information. Monograph Collection (Matt - Pseudo), 2000.

[8] J. P. Hayes, I. Polian, and B. Becker, "Testing for missing-gate faults in reversible circuits," IEEE Asian Test Symposium, pp. 100–105, 2004.

[9] I. Polian, J. P. Hayes, T. Fienn, and B. Becker,"A family of logical fault models for reversible circuits," IEEE Asian Test Symposium, pp 422–427, 2005.

[10] X. Fang-ying, C. Han-wu, L. Wen-jie and L. Zhi-giang, "Fault detection for single and multiple missing-gate faults in reversible circuits," in 2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence), pp. 131–135, 2008.

[11] 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, pp. 164–169, 2012.

[12] 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, pp. 68–73, 2014.

[13] 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), pp. 96–100, 2016.

[14] 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, Kolkata, India, Proceedings 9, Springer International Publishing, pp. 6–7, July 6-7, 2017.

[15] D. Kheirandish, M. Haghparast, M. Reshadi and M. Hosseinzadeh, "Efficient techniques for fault detection and location of multiple controlled Toffoli-based reversible circuit," Quantum Information Processing, 20(11), 370, 2021.

[16] D. Maslov, "Reversible logic synthesis benchmarks page, 2011," URL http://webhome.cs.uvic.ca/ dmaslov, 2020.