International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. 10, No.1 (Dec-2021)

http://dx.doi.org/10.12785/ijcds/1001126

New Board-Level Interconnect Fault Diagnosis Approach in Industrial Applications

Tamer Sayed Abdel Aziz[1] and Mohamed H. El-Mahlawy[2]

[1]Egyptian Armed Forces, Cairo, Egypt
[2]Electrical Engineering Department, Faculty of Engineering and Technology, Future University in Egypt, Cairo, Egypt

Received 15 Feb. 2021, Revised 14 Oct. 2021, Accepted 13 Nov. 2021, Published 28 Dec. 2021

Abstract: In this paper, the new BIST approach to test interconnect faults, based on the boundary scan architecture, is presented. The new algorithm is implemented by the MATLAB code, whose analysis is based on the random manner to generate the required test pattern set that detects interconnect faults without aliasing and confounding syndromes. The test pattern set complies with all requirements to detect two and three short-circuits from seven and ten terminals of ICs (boards). Different test responses of each short-circuit between different terminals are achieved, considered the basis of the presented fault diagnosis approach. In addition, this paper presents two generative approaches that generate the target test set. It is found that one generative approach using a linear feedback shift-register (LFSR) and a decoder reduces the test application time and suffers from aliasing and confounding syndromes due to the multi-input shift-register (MISR) with high hardware overhead. However, the other generative approach using an LFSR only has large test application time and is not suffering from aliasing and confounding syndromes with low hardware overhead. The new algorithm is compared with several previously published algorithms. The simulation results of the new algorithm have best results comparing to the existing algorithms in terms of the fault coverage and the applicability of the BIST scheme. The new algorithm is the most efficient algorithm to diagnose interconnect faults, based on two and three short-circuits from seven and ten terminals of ICs with accepted test application time and without aliasing and confounding syndromes.

Keywords: Testing of Interconnect Faults, Testing of Digital Circuits on the PCB, Fault Disgnosis of Digital Circuits, Testing Based on Boundary Scan

1. INTRODUCTION

The enhancement quality level of the industry production is the main issue in the board-level testing. Nowadays, most efforts are directed to enhance the testing cost and the test application time. The importance of printed circuit board (PCB) testing adds a role with the design for testability (DFT) techniques, merged from the beginning phase of design to reduce the testing cost and test application time. The main aim of testing is to detect the target faults and locate the place of their occurrence [1-4].

Faults in a circuit may occur due to defective components, assembled in the PCB, breaks in signal lines, lines shorted to ground or power supply rail, short-circuits between signal lines, excessive delays, etc. In general, the effect of a fault is represented by means of a model which represents the change in circuit signals. The fault models in digital circuits, used today are stuck-at faults [1, 5], bridging faults [6, 7], stuck-open faults, and delay faults [6, 8]. Different types of faults take place in the PCB during the assembly process and the soldering of components. The major faults are open and short circuits, which can appear between elements of the PCB. Components of the PCB are connected with the huge network of wiring connections and most of short and open circuits, occurred between terminals of PCB interconnections. Single-net and multi-net faults are considered the main faults that happens between PCB interconnections. Different algorithms were developed to detect these faults [7, 9-14]. They are either short-circuiting between any two or three wires, denoted as wired-AND (WAND) and wired-OR (WOR) (known bridging faults) or stuck-at faults in a single wire.

Traditionally, the automatic test equipment (ATE) is the main tool for the testing of a PCB. Their disadvantages are the testing cost, and high test application time. However, the DFT, and the built-in self-test (BIST) techniques [1, 5] are added to the PCB design

E-mail address: tilki101@gmail.com, mohamed.elmahlawy@fue.edu.eg

to reduce and eliminate the disadvantages of the ATE and to enhance both the testing cost, and test application time. The merging of the BIST to the PCB circuitry on the same board to fully test all PCB components is effective approach.

2. BACKGROUND AND PROBLEM STATEMENT

All state-of-the-art algorithms, presented in the literature to detect interconnect faults [7, 9-14], can generate the test pattern set that complies with all requirements to detect two short-circuits from seven or ten terminals of integrated circuits (boards). However, they are suffering from different problems either in detection of faults or the implementation of the test pattern set. The problem of the aliasing syndrome, referred to the situation of having the same signature from the test response compactor (TRC) for different test response of data sets. This problem leads to undetected faults. In addition, the problem of the confounding syndrome, referred to the situation of having two or three short-circuit nodes, produces the same test response of another two or three short-circuit nodes on the same board or the same integrated circuit (IC) chip.

The counting sequence algorithm (CSA) is considered the earliest works to detect the interconnect faults [9]. All possible binary combinations generate the parallel test vectors (PTVs) to be $\lceil \log_2 N \rceil$ for that algorithm, where N is the number of the tested terminal nodes. The drawbacks of the CSA are suffering from the problem of aliasing and confounding syndromes, and the detection of the stuck-at faults [9, 15]. The modified counting sequence algorithm (MCSA) was found in [10, 15]. Table I shows the test pattern set of the MCSA, modified to extend the PTVs to be $\lceil \log_2 (N + 2) \rceil$ instead of $\lceil \log_2 N \rceil$. It eliminates all logic '0' and all logic '1' sequential test vectors (STVs) so that every STVs has at least logic '0' and logic '1' to detect stuck-at faults [15]. Unfortunately, the MCSA is suffering from the problems of the aliasing syndrome and the confounding syndrome. The test set of the MCSA in Table I has the (7×4) matrix, whose row is the STV (= 7), and column is the PTV (= 4).

The true/complement test and diagnosis algorithm (TCTDA) was found by Paul Wagner [11]. In this algorithm, the test vectors were doubled to be $2\lceil \log_2 (N + 2) \rceil$ instead of $\lceil \log_2 (N + 2) \rceil$ as in the MCSA. The additional $\lceil \log_2 (N + 2) \rceil$ PTVs are obtained by complimenting the first set of test vectors, shown in Table I. The advantage of this algorithm is to remove the problem of aliasing syndrome and it can detect the stuck-at faults. The disadvantage of this algorithm still has the problem of the confounding syndrome. The walking one's / walking zero's sequence algorithm (WSA) was found in [12]. Its sequence, shown in Table I, was designed in such a way that any short-circuit between two nodes only are precisely identified in an output pattern. The advantage of the WSA is to remove the problem of aliasing and confounding syndromes. However, this algorithm requires more PTVs to test interconnect faults and detects two short-circuits only.

TABLE I. TEST PATTERN SET OF THE IMPORTANT ALGORITHMS

Nodes	PTVs of the MCSA				STVs	PTVs of the TCTDA								PTVs of the WSA		PTVs of the BSBTA			
	V1	*V2*	*V3*	*V4*		*True vectors*				*Complement vectors*				*One's Sequence*	*zero's Sequence*	*V1*	*V2*	*V3*	*V4*
N1	0	0	0	1	T1	0	0	0	1	1	1	1	0	1000000	0111111	1	0	0	0
N2	0	0	1	0	T2	0	0	1	0	1	1	0	1	0100000	1011111	0	1	0	0
N3	0	0	1	1	T3	0	0	1	1	1	1	0	0	0010000	1101111	1	0	1	0
N4	0	1	0	0	T4	0	1	0	0	1	0	1	1	0001000	1110111	1	0	0	1
N5	0	1	0	1	T5	0	1	0	1	1	0	1	0	0000100	1111011	0	0	0	1
N6	0	1	1	0	T6	0	1	1	0	1	0	0	1	0000010	1111101	1	1	0	1
N7	0	1	1	1	T7	0	1	1	1	1	0	0	0	0000001	1111110	0	1	1	1
N8					T8	1	0	0	0	0	1	1	1						
N9					T9	1	0	0	1	0	1	1	0						
N10					T10	1	0	1	0	0	1	0	1						

The w-test adaptive algorithm (WTAA) was found by the authors in [13] to limit the Walking One's algorithm. It was proposed in a two-step algorithm with a combination of the MCSA and the Walking One's sequence algorithm, whose disadvantage is high test application time to detect interconnect faults with only two short-circuits in two steps using two different algorithms. The Boundary-scan based testing algorithm (BSBTA) was proposed to test dominant-1 WOR, and dominant-0 WAND between interconnection of two nodes only and stuck-at faults [7, 14]. It can diagnose two short-circuit wire only and determine the specific position of target faults with four PTVs only, shown in Table I. In the BSBTA, the PTVs of all nodes (N1-N7) are simultaneously passed through the input boundary cell (BSC) in the boundary scan architecture [15-17]. The output of each cell is simultaneously compared with STVs of remaining nodes. The BSBTA detects interconnect faults (two short-circuits only) for seven nodes. The authors in [7, 14] proposed that, for any node pair, if it is logic '1' at the same bit, this bit position of STVs for both nodes is replaced by the state 'X'. Unfortunately, the BSBTA cannot be properly implemented in the real hardware to replace each state '1' by the state 'X'.

A. *Problem definition*

The main challenge in this problem is to develop the new algorithm to generate the required test pattern set for interconnection fault detection without aliasing and confounding syndromes. To achieve this objective for large number of IC outputs, the computations to get the required test pattern set must comply with certain requirements.

For example, to get the test pattern set with seven PTVs to test short-circuit fault between any two terminals of seven tested nodes, the (7×7) matrix is generated from all possible combinations of the binary sequence with 128 (2^7) and the number of all possible candidate matrices, generated from this binary sequence, is calculated from C_7^{128}, which equals 94,525,795,200 candidate matrices. For each (7×7) matrix, all combinations are checked for aliasing and confounding syndromes according to C_2^7, which equals 21. For the (10×10) matrix, all possible combinations of the binary sequence are 1024 (2^{10}) and the number of candidate matrices is calculated from C_{10}^{1024}, which equals 334,265,867,498,622,000,000,000 candidate matrices. For each (10×10) matrix, all combinations are checked aliasing and confounding syndromes according to C_2^{10}, which equals 45.

To get the test pattern set with n PTVs to test short-circuit fault between any two terminals of n tested nodes, the ($n \times n$) matrix is generated from all possible combinations of the binary sequence with $C_n^{2^n}$ candidate matrices. For each ($n \times n$) matrix, all combinations are checked for aliasing and confounding syndromes according to C_2^n. Therefore, the required time complexity is $O(C_2^n \times C_n^{2^n})$. From these candidate matrices, the applicable matrix must comply with our requirements to detect interconnect faults for two short-circuit terminals from seven nodes or ten nodes (terminal wire of IC outputs) without aliasing and confounding syndromes. By increasing the dimension of the matrices and increasing the number of short-circuit nodes, the computations become more and more hard. Therefore, the high-speed computer is required to finish the required calculations on reasonable time. For example, if the test process of a (7x7) matrix needs 1 msec, the whole process calculations need about three years.

B. *Motivation of the proposed methodology*

The objective of this paper is to develop the new algorithm that generates the applicable test pattern set to detect interconnect faults between two or three terminals from seven terminals or ten terminals and to eliminate the problem of aliasing and confounding syndromes. In addition, it is required to develop a new algorithm able to be implemented using the standard known test pattern generator.

Due to the massive computations required to generate the applicable test pattern set, the random search based on the MATLAB code is used to select the candidate test pattern set from the candidate matrix, and to check if it is the applicable test pattern set that detects the target faults without the problem of aliasing and confounding syndromes. Any short-circuit fault between any two or three terminals of PCB interconnections will be detected using the new algorithm and precisely determines its occurrence. In addition, the design of the required generator to generate these test pattern sets is required to be adapted with the BIST scheme. Two generative approaches are achieved in this paper. It is mainly based on the linear feedback register (LFSR) as the test pattern generator (TPG) and the multi-input shift-register (MISR) as the test response compactor (TRC).

This paper is divided into seven sections. The first section gives a brief introduction to the area of the presented work. The second section is motivated to the background of the presented work, problem definition, and the motivation of the proposed methodology. The third section presents the new algorithm to detect interconnects faults. The fourth section presents the new test pattern set to detect interconnect faults. The fifth section presents the test pattern generative approach. The sixth section presents the comparison between the presented approach in this paper and the other all approaches in the literature. The seventh section presents the conclusion and the summary of the presented work in this paper.

3. NEW ALGORITHM TO DETECT INTERCONNECT FAULTS

In this section, the presented algorithm is generated from ($2^n \times m$) main matrix of the binary sequence. The applicable test pattern set forms the ($n \times m$) matrix from the ($2^n \times m$) main matrix. Therefore, the total number of all possible candidate ($n \times m$) matrices from the ($2^n \times m$) main matrix is calculated from the following equation:

$$C_n^{2^n} = \frac{2^n!}{n! \times (2^n - n)!} \qquad (1)$$

Therefore, it is required to determine ($n \times m$) applicable matrix, represented by the applicable test pattern set. These test patterns are non-zeros and are not repeated to be fault-free from stuck-at faults. These m PTVs test n tested nodes of IC output terminals ($n = m$ is special case). To avoid the problems of the aliasing and confounding syndrome, the applicable test pattern set for interconnect fault detection is generated based on two test checks. In the generated ($n \times m$) matrix and $m \geq n$, when any two (three in the case of three short-circuits) rows in the matrix is logically performed OR operation with any other row in the matrix. The resulted rows are not like any other row in the matrix. If all rows in the matrix are logically performed OR operation together and the resulted rows are not like any row in the matrix, the generated test pattern set is fault-free from the problem of the aliasing syndrome. In addition, when any two (three in the case of three short-circuits) rows in the matrix is logically performed OR operation together, the resulted

row is not like the resulted row from other two (three in the case of three short-circuits) rows, performed by the same OR operation. If all possible resulted rows, performed OR operation, are not similar, the generated test pattern set is fault-free of the confounding syndrome.

Previous two test checks are followed to test the candidate test pattern set in the case of Dominant-1 fault detection. In case of Dominnat-0 fault detection, the previous two test checks should be repeated after replacing logic OR operation by logic AND operation. The code was written using the MATLAB software. The developed algorithm code is based on the following steps and two test checks.

1. The developer should enter the dimension of the targeted candidate binary matrix, so the number of rows (n) and columns (m) should be entered.
2. Using the MATLAB binary random function, a binary non-repeated sequence matrix is generated.
3. The first check for the generated candidate test pattern set is applied to check the existence of the aliasing syndrome. Each two (three) rows combination in the matrix are performed OR operation together. In this step, the first check takes place, based on the comparison between the resulted rows from the OR operations in the third step with each row in the matrix. The decision must be taken here if there is similarity between any rows (it means the resulted matrix is singular), the algorithm returns to the second step. On the other hand, if the similarity does not exist (it means the resulted matrix is non-singular), the algorithm continues to the fourth step.
4. The second check for the generated candidate test pattern set is applied to check the problem existence of confounding syndrome. Each two (three) rows in the matrix are performed OR operation together. In this step, the second check takes place, based on the comparison between the resulted rows from the OR operations in the fourth step together with other resulted rows. The decision must be taken here if there is similarity between any resulted rows, the algorithm returns to the second step. On the other hand, if the similarity does not exist the algorithm continues to the fifth step.
5. After applying the two previous checks for the aliasing syndrome and confounding syndrome, the generated candidate test pattern set is passed from both checks. Therefore, the applicable test pattern set is generated for interconnect fault detection.
6. All steps from 1 to 5 are repeated for dominant-0 test pattern set with replacing OR operations by AND operations.
7. By using the proposed algorithm and after generating the code, it is found that there are several applicable matrices can achieve the previous checks.

Since all previously published papers used the matrices with dimensions (7×7) and (10×10) to detect and locate two short-circuits, therefore, for the sake of comparison, the authors in this paper focus on the matrices with dimensions (7×7) and (10×10). However, the presented algorithm in this paper can be easily extended to select any dimension. In addition, no published papers speak about the detection of three short-circuits. Therefore, it is not possible to detect three short-circuits based on the matrices with dimensions (7×7) and (10×10). Therefore, the dimensions are extended to (7×11) matrix and (10×20) matrix to detect and locate three short-circuits without aliasing and confounding syndrome.

According to the previous search, there is no test pattern set generated from (7×4) matrix according to the proposal, presented in [7, 14]. Therefore, the presented algorithm in this paper generates new test pattern set for $m \geq n$ to detect interconnect faults between two (three) terminals from seven or ten terminals and to precisely determine its occurrence without the problem of aliasing and confounding syndromes. The number of all possible ($n \times m$) candidate matrices is generated from equation (1). From these candidate matrices, the applicable matrix must comply with our requirements. For each ($7 \times m$) matrix and $m \geq 7$, all test checks are $C_2^7 = 21$ for two short-circuits and $C_3^7 = 35$ for three short-circuits. For each ($10 \times m$) matrix and $m \geq 10$, all test checks are $C_2^{10} = 45$ for two short-circuits and $C_3^{10} = 120$ for three short-circuits.

By increasing the dimension of the matrix and increasing the number of short-circuit terminals, the computations become more difficult. Therefore, high-speed computer is required to finish the required calculations. After the analysis of all available solutions is achieved, high-speed devices run the program code to get all possible applicable matrices that verify the previous test checks. Unfortunately, it is difficult to achieve the optimum test pattern set that verifies the previous requirements and can be applied with minimum test application time.

4. NEW TEST PATTERN SET TO DETECT INTERCONNECT FAULTS

In this section, several applicable test pattern sets are generated from previous section for both two and three short-circuits. They overcome the problem of aliasing and confounding syndromes. Two applicable test pattern sets are presented. The first test pattern set is based on ($7 \times m$) matrix and $m \geq 7$. The second test pattern set is based on ($10 \times m$) matrix and $m \geq 10$. The following sub-sections will demonstrate these applicable test pattern sets for both dominant-1 and dominant-0.

A. Dominant-1 and dominant-0 of the applicable test pattern set for ($7 \times m$) matrix

After the program code is run, the dominant-1 and dominant-0 of the test pattern set, based on (7×7)

applicable matrix, are generated to detect interconnect faults of two short-circuits, shown in Table II. By the similar way, the generation of the new applicable test pattern set will be achieved. Computation challenge becomes more and more hard in the case of the detection of three short-circuits. Therefore, the value of m is increased to 11 instead of 7. After the program code is run, the dominant-1 and dominant-0 of the test pattern set, based on (7×11) applicable matrix, are generated to detect interconnect faults of three short-circuits, shown in Table. II.

To check the problem of aliasing and confounding syndromes, each two rows in the test pattern set are logically performed OR (in the case of the dominant-1) and AND (in the case of the dominant-0) operation together without repetition. All twenty-one rows (C_2^7) are written down their values in Table III. By comparing all resulted rows in Table III with each row in Table II, it is discovered that there is no similarity between any resulted rows in Table III and the rows in Table II. In addition, there is no similarity between any resulted rows in Table III. Therefore, it is found that the test pattern set in Table II, based on (7×7) applicable matrix, avoids the problem of aliasing and confounding syndromes for dominant-1 and dominant-0 interconnect faults of two short-circuits.

To check the problem of aliasing and confounding syndromes using the case of three short-circuits, each three rows in the test pattern set, shown in Table II, are logically performed OR (dominant-1) and AND (dominant-0) operation together without repetition. All thirty-five rows (C_3^7) are written down their values in Table IV. By comparing all resulted rows in Table IV with each row in Table II, it is discovered that there is no similarity between any resulted row in Table IV and the rows in Table II. In addition, there is no similarity between any resulted rows in Table IV. Therefore, it is found that the test pattern set in Table II, based on (7×11) applicable matrix, avoids the problem of aliasing and confounding syndromes for dominant-1 and dominant-0 interconnect faults of three short-circuits.

TABLE II. NEW TEST PATTERN SET OF (7×M) MATRIX FOR TWO SHORT-CIRCUITS AND THREE SHORT-CIRCUITS

Node	*m* = 7 for two short-circuits		*m* = 11 for three short-circuits	
	Dominant-1	*Dominant-0*	*Dominant-1*	*Dominant-0*
N1	0 1 1 0 0 0 0	1 0 0 1 1 1 1	0 0 0 1 0 1 0 0 0 1 0	1 1 1 0 1 0 1 1 1 0 1
N2	1 0 0 1 0 1 0	0 1 1 0 1 0 1	1 0 0 1 0 1 0 1 0 0 1	0 1 1 0 1 0 1 0 1 1 0
N3	0 0 0 1 1 1 0	1 1 1 0 0 0 1	0 1 1 1 0 0 0 0 0 0 0	1 0 0 0 1 1 1 1 1 1 1
N4	0 1 0 0 1 0 0	1 0 1 1 0 1 1	0 1 0 0 0 1 0 0 1 0 0	1 0 1 1 1 0 1 1 0 1 1
N5	0 0 1 0 0 1 0	1 1 0 1 1 0 1	1 0 0 0 1 1 0 0 1 0 0	0 1 1 1 0 0 1 1 0 1 1
N6	1 0 1 0 0 0 1	0 1 0 1 1 1 0	0 0 0 1 0 0 1 0 0 0 0	1 1 1 0 1 1 0 1 1 1 1
N7	0 0 0 0 1 1 1	1 1 1 1 0 0 0	0 0 0 0 0 0 0 1 0 0 0	1 1 1 1 1 1 1 1 0 1 1

TABLE III. THE RESULTED ROWS FROM OR (AND) OPERATION OF EACH ROW IN TABLE II (M = 7).

No	Short-Circuit	Dominant-1 Resulted Row	Dominant-0 Resulted Row
1	N1N2	1111010	0000101
2	N1N3	0111110	1000001
3	N1N4	0110100	1001011
4	N1N5	0110010	1001101
5	N1N6	1110001	0001110
6	N1N7	0110111	1001000
7	N2N3	1001110	0110001
8	N2N4	1101110	0010001
9	N2N5	1011010	0100101
10	N2N6	1011011	0100100
11	N2N7	1001111	0110000
12	N3N4	0101110	1010001
13	N3N5	0011110	1100001
14	N3N6	1011111	0100000
15	N3N7	0001111	1110000
16	N4N5	0110110	1001001
17	N4N6	1110101	0001010
18	N4N7	0100111	1011000
19	N5N6	1010011	0101100
20	N5N7	0010111	1101000
21	N6N7	1010111	0101000

TABLE IV. THE RESULTED ROWS FROM OR (AND) OPERATION OF EACH ROW IN TABLE II (M = 11).

No	Short-Circuit	Dominant-1 Resulted Row	Dominant-0 Resulted Row
1	N1N2N3	11110101011	00001010100
2	N1N2N4	11010101111	00101010000
3	N1N2N5	10011101111	01100010000
4	N1N2N6	01001011011	01101000100
5	N1N2N7	10010101011	01101010100
6	N1N3N4	01110100110	10001011001
7	N1N3N5	11111100110	00000011001
8	N1N3N6	01110110010	10001001101
9	N1N3N7	01110101010	10001010101
10	N1N4N5	11011100110	00100011001
11	N1N4N6	1010110110	10101001001
12	N1N4N7	1010101110	10101010001
13	N1N5N6	10011110110	01100001001
14	N1N5N7	10011101110	01100010001
15	N1N6N7	00010111010	11101000101
16	N2N3N4	11110101101	00001010010
17	N2N3N5	11111101101	00000010010
18	N2N3N6	11110111001	00001000110
19	N2N3N7	11110101001	00001010110
20	N2N4N5	11011101101	00100010010
21	N2N4N6	11010111101	00101000010
22	N2N4N7	11010101101	00101010010
23	N2N5N6	10011111101	01100000010
24	N2N5N7	10011101101	01100010010
25	N2N6N7	10010111001	01101000110
26	N3N4N5	11111100100	00000011011
27	N3N4N6	01110110100	10001001011
28	N3N4N7	01110101100	10001010011
29	N3N5N6	11111110100	00000001011

No	Short-Circuit	Dominant-1 Resulted Row	Dominant-0 Resulted Row
30	N3N5N7	11111101100	00000010011
31	N3N6N7	01110011000	10001100111
32	N4N5N6	11011110100	00100001011
33	N4N5N7	11001101100	00110010011
34	N4N6N7	01010111100	10101000011
35	N5N6N7	10011111100	01100000011

B. Dominant-1 and dominant-0 of the applicable test pattern set for (10 × m) matrix

After the program code is run, the dominant-1 and dominant-0 of the test pattern set, based on (10×10) applicable matrix, are generated to detect interconnect faults of two short-circuits, shown in Table V. Computation challenge of the new test pattern set becomes more and more hard in the case of the detection of three short-circuits especially when increasing the values of *n*. Therefore, the value of *m* is increased to 20 instead of 10. After the program code is run, the dominant-1 and dominant-0 of the test pattern set, based on (10×20) applicable matrix, are generated to detect interconnect faults of three short-circuits, shown in Table V.

To check the problem of aliasing and confounding syndromes, each two rows in the test pattern set, shown in Table V, are logically performed OR (dominant-1) or AND (dominant-0) operation together without repetition. All forty-five rows (C_2^{10}) are written down their values in Table VI. By comparing all resulted rows in Table VI with each row in Table V, it is discovered that there is no similarity between any resulted row in Table VI and the rows in Table V. In addition, there is no similarity between any resulted row in Table VI. Therefore, the test pattern set in Table V, based on (10×10) applicable matrix, avoids the problem of aliasing and confounding syndromes for dominant-1 and dominant-0 interconnect faults of two short-circuits.

By the similar way using interconnect fault detection of three short-circuits, each three rows in the test pattern set, shown in Table V, are logically performed OR (dominant-1) or AND (dominant-0) operation together without repetition. All one-hundred and twenty rows (C_3^{10}) are written down their values in Table VII. By comparing all resulted rows in Table VII with each row in Table V, it is discovered that there is no similarity between any resulted row in Table VII and the rows in Table V. In addition, there is no similarity between any resulted row in Table VII. Therefore, the test pattern set in Table V, based on (10×20) applicable matrix, avoids the problem of aliasing and confounding syndromes for dominant-1 and dominant-0 interconnect faults of three short-circuits.

TABLE V. NEW TEST PATTERN SET OF (10×M) MATRIX FOR TWO SHORT-CIRCUITS AND THREE SHORT-CIRCUITS

Node	*m* = 10 for two short-circuits		*m* = 20 for three short-circuits	
	Dominant-1	*Dominant-0*	*Dominant-1*	*Dominant-0*
N1	1 0 0 0 0 0 1 1 0 0	0 1 1 1 1 1 0 0 1 1	11001100010001000001	00110011101110111110
N2	0 0 1 0 1 1 0 1 1 1	1 1 0 1 0 0 1 0 0 0	01100111000011000000	10011000111100111111
N3	0 0 0 0 0 1 1 0 1 0	1 1 1 1 1 0 0 1 0 1	01000101010100110010	10111010101011001101
N4	0 1 0 1 0 0 1 0 0 1	1 0 1 0 1 1 0 1 1 0	10101100011100100000	01010011100011011111
N5	0 0 0 1 1 0 1 1 0 0	1 1 1 0 0 1 0 0 1 1	11100000001101011000	00011111110010100111
N6	0 1 1 0 0 0 1 0 1 1	1 0 0 1 1 1 0 1 0 0	01010100110010000010	10101011001101111101
N7	0 0 1 1 0 1 0 0 0 1	1 1 0 0 1 0 1 1 1 0	10010100101101100100	01101011010010011011
N8	0 1 0 1 0 1 0 1 1 0	1 0 1 0 1 0 1 0 0 1	10001000101000001100	01110111010111110011
N9	1 1 0 0 0 1 0 1 0 1	0 0 1 1 1 0 1 0 1 0	10010001000000000000	01101110111111111111
N10	1 0 0 0 1 1 0 0 1 1	0 1 1 1 0 0 1 1 0 0	00000001100011001001	11111110011100110110

TABLE VI. THE RESULTED ROWS FROM OR (AND) OPERATION OF EACH ROW IN TABLE V (M = 10).

No	Short Circuit	Dominant-1 Resulted Row	Dominant-0 Resulted Row	No	Short Circuit	Dominant-1 Resulted Row	Dominant-0 Resulted Row	No	Short Circuit	Dominant-1 Resulted Row	Dominant-0 Resulted Row
1	N1N2	1010111111	0101000000	16	N2N9	1110110111	1111101000	31	N5N6	0111101111	1000010000
2	N1N3	1000011110	0111100001	17	N2N10	1010110111	0101001000	32	N5N7	0011111101	1100000010
3	N1N4	1101001101	0010110010	18	N3N4	0101011011	1010100100	33	N5N8	0101111110	1010000001
4	N1N5	1001101100	0110010011	19	N3N5	0001111110	1110000001	34	N5N9	1101111101	0010000010
5	N1N6	1110001111	0001110000	20	N3N6	0110011011	1001100100	35	N5N10	1001111111	0110000000
6	N1N7	1011011101	0100100010	21	N3N7	0011011011	1100100100	36	N6N7	0111011011	1000100100
7	N1N8	1101011110	0010100001	22	N3N8	0101011110	1010100001	37	N6N8	0111011111	1000100000
8	N1N9	1100011101	0011100010	23	N3N9	1100011111	0011100000	38	N6N9	1110011111	0001100000
9	N1N10	1000111111	0111000000	24	N3N10	1000111011	0111000100	39	N6N10	1110111011	0001000100
10	N2N3	0010111111	1101000000	25	N4N5	0101101101	1010010010	40	N7N8	0111010111	1000101000
11	N2N4	0111111111	1000000000	26	N4N6	0111001011	1000110100	41	N7N9	1111010101	0000101010
12	N2N5	0011111111	1100000000	27	N4N7	0111011001	1000100110	42	N7N10	1011110011	0100001100
13	N2N6	0110111111	1001000000	28	N4N8	0101011111	1010100000	43	N8N9	1101010111	0010101000
14	N2N7	0011110111	1100001000	29	N4N9	1101011101	0010100010	44	N8N10	1101110111	0010001000
15	N2N8	0111110111	1000001000	30	N4N10	1101111011	0010000100	45	N9N10	0111101111	0011001000

TABLE VII. THE RESULTED ROWS FROM OR (AND) OPERATION OF EACH ROW IN TABLE V (M = 20).

No	Short-Circuit	Dominant-1 Resulted Row	Dominant-0 Resulted Row	No	Short Circuit	Dominant-1 Resulted Row	Dominant-0 Resulted Row
1	N1N2N3	11101111010011000001	00010000101000001100	61	N2N7N10	11110111101111101101	00001000010000010010
2	N1N2N4	11101111011111100001	00010000100000011110	62	N2N8N9	11111111101011001100	00000000010100110011
3	N1N2N5	11101111011111011001	00010000100000100110	63	N2N8N10	11101111101011001101	00010000010100110010
4	N1N2N6	11111111110011000011	00000000001100111100	64	N2N9N10	11110111100011001001	00001000011100110110
5	N1N2N7	111111111111 11100101	00000000000000011010	65	N3N4N5	11101101011101111010	00010010100010000101
6	N1N2N8	11101111010111001101	00010000000100110010	66	N3N4N6	11111101111110110010	00000010000001001101
7	N1N2N9	11111111010011000001	00000000101100111110	67	N3N4N7	11111101111101110110	00000010000010001001
8	N1N2N10	11101111110011001001	00010000001100110110	68	N3N4N8	11101101111100111110	00010010000011000001
9	N1N3N4	11101101011101110011	00010010100010001100	69	N3N4N9	11111101011100110010	00000010100011001101
10	N1N3N5	11101101011101111011	00010010100010000100	70	N3N4N10	11101101111111111011	00010010000000000100
11	N1N3N6	11011101110111110011	00100010001000001100	71	N3N5N6	11110101111111111010	00001010000000000101
12	N1N3N7	11011101111101110111	00100010000010001000	72	N3N5N7	11110101111101111110	00001010000010000001
13	N1N3N8	11001101111101111111	00110010000010000000	73	N3N5N8	11101101111101111110	00010010000010000001
14	N1N3N9	11011101010101110011	00100010101010001100	74	N3N5N9	11110101011101111010	00001010100010000101
15	N1N3N10	11001101110111111011	00110010001000000100	75	N3N5N10	11100101111111111011	00011010000000000100
16	N1N4N5	11101100011101111001	00010011100010000110	76	N3N6N7	11010101111111110110	00101010000000001001
17	N1N4N6	11111100111111100011	00000011000000011100	77	N3N6N8	11011101111110111110	00100010000001000001
18	N1N4N7	11111100111101100101	00000011000010011010	78	N3N6N9	11010101110110110010	00101010001001001101
19	N1N4N8	11101100111101101101	00010011000010010010	79	N3N6N10	01010101110111111011	10101010001000000100
20	N1N4N9	11111101011101100001	00000010100010011110	80	N3N7N8	11011101111101111110	00100010000010000001
21	N1N4N10	11101101111111101001	00010010000000010110	81	N3N7N9	11010101111101110110	00101010000010001001
22	N1N5N6	11111100111111011011	00000011000000100100	82	N3N7N10	11010101111111111111	00101010000000000000
23	N1N5N7	11111100111101111101	00000011000010000010	83	N3N8N9	11011101111100111110	00100010000011000001
24	N1N5N8	11101100111101011101	00010011000010100010	84	N3N8N10	11001101111111111111	00110010000000000000
25	N1N5N9	11111101011101011001	00000010100010100110	85	N3N9N10	11010101110111111011	00101010001000000100
26	N1N5N10	11101101111111011001	00010010000000100110	86	N4N5N6	11111100111111111010	00000011000000000101
27	N1N6N7	11011100111111100111	00100011000000011000	87	N4N5N7	11111100111101111100	00000011000010000011
28	N1N6N8	11011100110111001111	00100011000100110000	88	N4N5N8	11101100111101111100	00010011000010000011
29	N1N6N9	11011101110011000011	00100010001100111100	89	N4N5N9	11111101011101111000	00000010100010000111
30	N1N6N10	11011101110011001011	00100010001100110100	90	N4N5N10	11101101111111111001	00010010000000000110
31	N1N7N8	11011100111101101101	00100011000010010010	91	N4N6N7	11111100111111100110	00000011000000011001
32	N1N7N9	11011101111101100101	00100010000010011010	92	N4N6N8	11111100111110101110	00000011000001010001
33	N1N7N10	11011101111111101101	00100010000000010010	93	N4N6N9	11111101111110100010	00000010000001011101
34	N1N8N9	11011101111001001101	00100010000110110010	94	N4N6N10	11111101111111101011	00000010000000010100
35	N1N8N10	11001101111011001101	00110010000100110010	95	N4N7N8	10111100111101101100	01000011000010010011
36	N1N9N10	11011101110011001001	00100010001100110110	96	N4N7N9	10111101111101100100	01000010000010011011
37	N2N3N4	11101111011111110010	00010000100000001101	97	N4N7N10	10111101111111101101	01000010000000010010
38	N2N3N5	11100011011111111010	00011000100000000101	98	N4N8N9	10111101111100101100	01000010000011010011
39	N2N3N6	01110111110111110010	10001000001000001101	99	N4N8N10	10101101111111101101	01010010000000010010
40	N2N3N7	11110111111111110110	00001000000000001001	100	N4N9N10	10111101111111101001	01000010000000010110
41	N2N3N8	11101111111111111110	00010000000000000001	101	N5N6N7	11110100111111111110	00001011000000000001
42	N2N3N9	11110111010111110010	00001000101000001101	102	N5N6N8	11111100111111011110	00000011000000100001
43	N2N3N10	01100111110111111011	10011000001000000100	103	N5N6N9	11110101111111011010	00001010000000100101
44	N2N4N5	11101111011111111000	00010000100000000111	104	N5N6N10	11110101111111011011	00001010000000100100
45	N2N4N6	11111111111111100010	00000000000000011101	105	N5N7N8	11111100101101111100	00000011010010000011
46	N2N4N7	11111111111111100100	00000000000000011011	106	N5N7N9	11110101101101111100	00001010010010000011
47	N2N4N8	11101111111111101100	01110111010111110011	107	N5N7N10	11110101101111111101	00001010010000000010
48	N2N4N9	11111111011111100000	01101110111111111111	108	N5N8N9	11111001101101011100	00000110010010100011
49	N2N4N10	11101111111111101001	11111110011100110110	109	N5N8N10	11101001101111011101	00010110010000100010
50	N2N5N6	11110111111111011010	00001000000000100101	110	N5N9N10	11110001101111011001	00001110010000100110
51	N2N5N7	11110111101111111100	00001000010000000011	111	N6N7N8	11011100111111101110	00100011000000010001
52	N2N5N8	11101111101111011100	00010000010000100011	112	N6N7N9	11010101111111100110	00101010000000011001
53	N2N5N9	11110111001111011000	00001000110000100111	113	N6N7N10	11010101111111101111	00101010000000010000
54	N2N5N10	11100111101111011001	00011000010000100110	114	N6N8N9	11011101111010001110	00100010000101110001
55	N2N6N7	11110111111111100110	00001000000000011001	115	N6N8N10	11011101111011001111	00100010000100110000
56	N2N6N8	11111111111011001110	00000000000100110001	116	N6N9N10	11010101110011001011	00101010001100110100
57	N2N6N9	11110111100111000010	00001000001100111101	117	N7N8N9	10011101101101101100	01100010010010010011
58	N2N6N10	01110111110011001011	10001000001100110100	118	N7N8N10	10011101101111101101	01100010010000010010
59	N2N7N8	11111111101111101100	00000000010000010011	119	N7N9N10	10010101101111101101	01101010010000010010
60	N2N7N9	11110111101111100100	00001000010000011011	120	N8N9N10	10011001101011001101	01100110010100110010

5. TEST PATTERN GENERATIVE APPROACHES

The previous published approaches in the literature that detect interconnect faults proposed the way of the implementation, based on the boundary scan architecture [6, 9-14]. They are based on the serial test scheme [14]. Therefore, the required test application time is high due to the scan of each PTV and the test response of the previous PTV through the boundary scan input-output cells. The authors in [17] proposed the incorporation of the BIST circuitry and the boundary scan circuitry into one test architecture. This test scheme consumes less test application time. Therefore, in this paper, two test pattern generative approaches to generate the applicable test pattern sets are achieved based on the parallel BIST

scheme [1, 17]. The first test pattern generative approach is based on either 7-bit LFSR for (7×*m*) matrix in Table II or 10-bit LFSR for (10×*m*) matrix in Table V as the TPG and the MISR as the TRC. The second test pattern generative approach is based on 7-bit LFSR (or 10-bit LFSR) with a decoder as the TPG and the MISR as the TRC. The simulation for the short circuits between any two or three terminals and the compaction of their test responses are achieved in the following sub-sections.

A. First test pattern generative approach

The applicable test pattern sets, required to be implemented based on the (7×*m*) matrix in Table II and the (10×*m*) matrix in Table V, are included in the test sequence of 7-bit LFSR and 10-bit LFSR, respectively. Therefore, the LFSR can generate the desired applicable test pattern set through its output states. The total length of the 7-bit LFSR output sequence without repetition is $(2^7 - 1)$ output states, and $(2^{10} - 1)$ output states for the 10-bit LFSR. It is required to choose the proper primitive polynomial to generate the applicable test pattern set with minimal clock number.

The applicable test pattern set, based on the (7×*m*) matrix, is generated using 7-bit LFSR and its primitive polynomial of $(1 + x^6 + x^7)$ [18-21]. Based on the simulation results, Table VIII shows that the applicable PTVs of dominant-1 (7×7) matrix are generated between the test sequence 12 and 113 to achieve 102 clocks, and between the test sequence 51 and 127 to achieve 77 clocks for the applicable PTVs of dominant-0 (7×7) matrix. In addition, Table VIII shows that the applicable PTVs of dominant-1 (7×11) matrix are generated between the test sequence 4 and 90 to achieve 87 clocks, and between the test sequence 41 and 124 to achieve 84 clocks for the applicable PTVs of dominant-0 (7×11) matrix.

By the same way, the test pattern set based on the (10×*m*) matrix is generated using 10-bit LFSR and its primitive polynomial of $(1 + x + x^3 + x^4 + x^{10})$ [18-21]. The applicable PTVs of dominant-1 (10×10) matrix are generated between the test sequence 19 and 1008 to achieve 990 clocks, and between the test sequence 96 and 813 to achieve 718 clocks for the applicable PTVs of dominant-0 (10×10) matrix. In addition, the applicable PTVs of dominant-1 (10×20) matrix are generated between the test sequence 38 and 1003 to achieve 966 clocks, and between the test sequence 2 and 927 to achieve 926 clocks for the applicable PTVs of dominant-0 (10×20) matrix. In all cases, the applicable test pattern vectors are included within output states of the 7-bit LFSR, and the 10-bit LFSR.

The test response of the generated PTVs is compacted by the MISR. Therefore, the parallel test scheme, based on the 7-bit LFSR and the 7-bit MISR, is used to generate the target PTVs and to compact their test responses, respectively. The Xilinx ISE Design Suite simulator of the Field Programmable Gate Array (FPGA) technology is used to simulate this test scheme under the effect of all expected two (three) short-circuits for dominant-1 and dominant-0 faults between any two (three) terminals. The logic OR (AND) simulates dominant -1 (dominant -0) short circuits between target terminals. The LFSR outputs are directly applied to the MISR inputs for test response compaction, except the simulated short-circuit terminals that pass-through logic OR (AND) to be compacted by the MISR. Finally, the required clocks to generate the target PTVs are the summation of the required clocks to generate the required output states of the LFSR besides the required clocks to apply the initial seed and to extract the generated signature from the MISR.

TABLE VIII. APPLICABLE TEST PATTERNS OF (7×M) MATRIX

Applicable test patterns of (7×7) matrix		**Applicable test patterns of (7×11) matrix**	
Test sequence number	***Applicable test vector***	***Test sequence number***	***Applicable test vector***
12	1100000	4	0000001
16	0000110	6	0100000
32	0001001	7	0010000
34	0100010	9	0000100
79	1010110	10	0000010
94	0110001	14	0011000
113	1001100	15	0001100
51	1111001	30	0100111
84	1110110	59	1000010
104	1011101	70	0010010
109	1001110	90	0011011
111	0110011	41	1110011
116	0101001	54	1011111
127	0011111	72	1100100
		83	1101101
		85	1111011
		86	0111101
		88	1101111
		95	1011000
		110	1100111
		123	1111101
		124	1111110

The TRC, based on the MISR, detects errors in stream data bits, caused by interconnect faults. It generates a signature by the MISR of each tested node. When the fault-free signature and measured signature are differed, a fault is detected. The aliasing probability of an *n*-stage MISR approaches 2^{-n} [22]. Therefore, it is required to detect a fault and to locate its place of occurrence between short-circuit of two (or three) terminals. All expected short-circuit faults and their test responses are compacted by the MISR. Based on the FPGA simulation results, the signatures generated from the MISR of the dominant-1 and dominant-0 test pattern generation of the (7×7) matrix for two short-circuits and (7×11) matrix for three short-circuits are presented in Table IX. The signatures ($SIGN_{(n\times m)}$), shown in Table IX, are in the hexadecimal number format. In addition, the signatures, generated from the MISR of the dominant-1 and dominant-0 test pattern generation of the (10×10) matrix for two short-circuits and (10×20) matrix for three short-circuits, are presented in Table IX. From Table IX, different signatures for each

short-circuit indicates the ability of this approach to detect a fault and to locate its occurrence without neither the aliasing nor the confounding syndromes.

TABLE IX. ALL REQUIRED SIGNATURES GENERATED FROM THE MISR BASED ON THE FIRST TEST PATTERN GENERATIVE APPROACH

Short-circuit No.	Dominant-1		Dominant-0		Dominant-1		Dominant-0		Short-circuit No.	Dominant-1	Dominant-0
	$SIGN_{(7\times7)}$	$SIGN_{(7\times11)}$	$SIGN_{(7\times7)}$	$SIGN_{(7\times11)}$	$SIGN_{(10\times10)}$	$SIGN_{(10\times20)}$	$SIGN_{(10\times10)}$	$SIGN_{(10\times20)}$		$SIGN_{(10\times20)}$	$SIGN_{(10\times20)}$
Fault-free	00	33	7f	32	1f3	33e	33c	17f			
1	56	06	0a	5a	3e6	2a8	1f2	1f9	61	32e	1ea
2	43	3d	55	4c	3b2	339	1f1	1fd	62	314	11a
3	01	6a	1e	30	313	329	1f7	1fc	63	254	1da
4	46	6c	65	34	055	21b	1fb	1f7	64	077	1bf
5	7a	61	5b	60	3db	12e	1e3	1ef	65	1fe	078
6	2d	74	76	7a	14e	352	1d3	1df	66	2c0	074
7	63	5b	7d	0c	3b4	25e	1ba	1ba	67	103	06c
8	28	7b	5d	77	3e8	2f7	17c	17f	68	13f	05c
9	17	20	03	1a	379	393	1e8	0f1	69	005	039
10	70	48	52	61	1bd	345	0f0	0fa	70	37f	0f9
11	64	7e	63	63	149	127	0f6	06f	71	00c	072
12	3d	52	02	2e	2ae	1e5	0fa	0ec	72	125	06a
13	3b	18	37	4f	366	38f	0e2	0dc	73	38c	05a
14	24	67	43	0d	2f5	24d	0d2	0ab	74	036	04f
15	6b	07	1d	45	1da	071	2b3	07b	75	034	0ff
16	10	46	1b	03	194	351	279	0f8	76	354	066
17	22	2a	08	54	341	28e	1f8	0f5	77	32c	056
18	25	2e	73	35	11e	21a	0f5	0ed	78	275	033
19	30	59	3d	71	0ed	156	0f9	0dd	79	07f	0f3
20	2a	35	39	36	1c4	2fe	0e1	0b8	80	3b2	04e
21	61	69	2a	00	111	24c	0d1	087	81	24e	02b
22		1e		58	2b0	27a	0d1	0f4	82	055	0eb
23		0e		43	1e3	081	276	0e6	83	0a5	01b
24		3f		55	146	189	3ff	0dd	84	27c	0db
25		68		28	019	23b	0ff	0be	85	312	1b9
26		0a		21	324	1c0	39c	07e	86	0ca	071
27		5a		25	310	325	3ac	0e7	87	37b	069
28		6d		1f	1f7	257	0be	0d7	88	20e	059
29		2d		7e	22a	2f1	078	0b2	89	033	03c
30		0b		27	192	3d3	3f5	073	90	06d	0fc
31		04		65	2d7	10a	0eb	0cf	91	15e	065
32		4b		31	0f7	17e	0db	0aa	92	16c	055
33		63		75	2b9	3ae	0b2	06d	93	060	030
34		41		0f	06a	20c	074	09a	94	001	0f0
35		51		1e	31e	35d	1f3	05b	95	238	04d
36					3ca	28c	0c3	03f	96	23a	028
37					3d8	270	0aa	17c	97	02f	0e8
38					1ea	2e0	0c2	17a	98	205	018
39					3d2	220	1e9	176	99	355	0d8
40					1f8	0d2	09a	16e	100	3f6	0bd
41					196	128	05c	15e	101	2a7	063
42					34f	304	1d9	13b	102	250	053
43					184	274	039	1fb	103	22e	036
44					10b	2d5	1b0	179	104	004	0f6
45					16f	10f	17a	175	105	05f	04b
46						17c		16d	106	28d	02e
47						3a4		15d	107	37c	0ee
48						340		138	108	1b0	01e
49						282		1f8	109	247	0de
50						2d0		173	110	277	0bb
51						100		16b	111	3ad	047
52						323		15b	112	22f	022
53						258		13e	113	040	0e2
54						135		1fe	114	3a3	012
55						154		167	115	3f3	0d2
56						357		157	116	27e	0b7
57						114		132	117	341	00a

Short-circuit No.	Dominant-1		Dominant-0		Dominant-1		Dominant-0		Short-circuit No.	Dominant-1	Dominant-0
	$SIGN_{(7\times7)}$	$SIGN_{(7\times11)}$	$SIGN_{(7\times7)}$	$SIGN_{(7\times11)}$	$SIGN_{(10\times10)}$	$SIGN_{(10\times20)}$	$SIGN_{(10\times10)}$	$SIGN_{(10\times20)}$		$SIGN_{(10\times20)}$	$SIGN_{(10\times20)}$
58						325		1f2	118	024	0ca
59						10c		14f	119	1ae	0af
60						130		12a	120	0b4	09f

B. The second test pattern generative approach

The second approach for the implementation of the test pattern generation depends on the 7-bit LFSR with a decoder as the TPG and the 7-bit MISR as the TRC. The idea is to use 7-bit LFSR with a decoder to generate the desired applicable PTVs. The decoder converts the output sequence of the LFSR to the required PTVs in order to minimize the required number of clocks. Table X and Table XI represent the conversion tables of the LFSR and a decoder for (7×7) matrix, (7×11) matrix, (10×10) matrix, and (10×20) matrix, respectively. The inputs of the decoder are applied by the output sequence of the LFSR, and the output sequence of the decoder is used as the applicable dominant-1 (dominat-0) PTVs. This test generation approach needs only seven clocks for (7×7) matrix and eleven clocks for (7×11) matrix to generate the domanint-1 (domanint-0) PTVs. In addition, it needs ten clocks for (10×10) matrix and twenty clocks for (10×20) matrix to generate the domanint-1 (domanint-0) PTVs. The hardware overhead of this approach focuses on the hardware required to design the decoder besides the required hardware of the LFSR.

All expected short-circuit faults and their test responses are compacted by the MISR. Based on the FPGA simulation results, the signatures, generated from the MISR for the dominant-1 and dominant-0 test pattern generation of all four matrices are shown in Table XII. The signatures in Table XII are in the hexadecimal number format, collected from the FPGA simulation. It is found that there is the aliasing and the confounding syndromes. Therefore, interconnect faults cannot precisely be detected, based on the second approach. For example, short-circuit 2 (N1N3) of the dominant-1 (7×7) matrix has the same signature of short-circuit 16 (N4N5), and short-circuit 20 (N5N7) of the dominant-0 (7×7) matrix has the same signature of short-circuit 21 (N6N7). In addition, short-circuit 65 (N3N4N5) of the dominant-1 (10×20) matrix has the same signature of short-circuit 66 (N3N4N6), and short-circuit 7 (N1N2N9) of the dominant-0 (10×20) matrix has the same signature of short-circuit 9 (N1N3N4). All shadow cells in Table XII cause the similarly in their signatures.

TABLE X. LFSR/DECODER CONVERSION TABLE FOR DOMINANT-1 AND DOMINAT-0 (7×M) MATRIX

No	(7×7) matrix				(7×11) matrix			
	Dominant-1		*Dominant-0*		*Dominant-1*		*Dominant-0*	
	LFSR Outputs	*Decoder Outputs*	*LFSR Outputs*	*Decoder Outputs*	*LFSR Outputs*	*Decoder Outputs*	*LFSR Outputs*	*Decoder Outputs*
1	0001111	0100010	0001000	1011101	0001111	0100100	1100000	1011011
2	0000111	1001000	0000100	0110111	0000111	0011000	0110000	1100111
3	0000011	1000110	0000010	0111001	0000011	0010000	0011000	1101111
4	0000001	0110000	1000001	1001111	0000001	1110010	0001100	0001101
5	1000000	0011001	1100000	1100110	1000000	0000100	0000110	1111011
6	0100000	0110101	0110000	1001010	0100000	1101100	1000011	0010011
7	0010000	0000011	0011000	1111100	0010000	0000010	0100001	1111101
8					0001000	0100001	1010000	1011110
9					0000100	0001100	0101000	1110011
10					0000010	1000000	0010100	0111111
11					1000001	0100000	0001010	1011111

TABLE XI. LFSR/DECODER CONVERSION TABLE FOR DOMINANT-1 AND DOMINAT-0 (10×M) MATRIX

No	(10×10) matrix				(10×20) matrix			
	Dominant-1		*Dominant-0*		*Dominant-1*		*Dominant-0*	
	LFSR Outputs	*Decoder Outputs*	*LFSR Outputs*	*Decoder Outputs*	*LFSR Outputs*	*Decoder Outputs*	*LFSR Outputs*	*Decoder Outputs*
1	1000000000	1000000011	1011100011	0111111100	1000000000	1001101110	0110000100	0110010001
2	1100000000	0001010110	0101110001	1110101001	1100000000	1110110000	1011000010	0001001111
3	1110000000	0100011000	0010111000	1011100111	1110000000	0101100000	1101100001	1010011111
4	0111000000	0001101100	1001011100	1110010011	0111000000	0000011010	1110110000	1111100101
5	0011100000	0100100001	0100101110	1011011110	0011100000	1001000100	0111011000	0110111011
6	0001110000	0110001111	0010010111	1001110000	0001110000	1111011000	0011101100	0000100111
7	1000111000	1011110000	0001001011	0100001111	1000111000	0100000000	0001110110	1011111111
8	1100011100	1100100110	0000100101	0011011001	1100011100	0110000011	1000111011	1001111100

No	(10×10) matrix				(10×20) matrix			
	Dominant-1		*Dominant-0*		*Dominant-1*		*Dominant-0*	
	LFSR Outputs	*Decoder Outputs*	*LFSR Outputs*	*Decoder Outputs*	*LFSR Outputs*	*Decoder Outputs*	*LFSR Outputs*	*Decoder Outputs*
9	1110001110	0110010101	1000010010	1001101010	1110001110	0000011101	0100011101	1111100010
10	0111000111	0101011011	1100001001	1010100100	0111000111	1011010000	1010001110	0100101111
11					1011100011	0001101100	0101000111	1110010011
12					0101110001	0011101000	0010100011	1100010111
13					0010111000	0100010001	0001010001	1011101110
14					1001011100	1100101001	0000101000	0011010110
15					0100101110	0011001000	0000010100	1100110111
16					0010010111	0010100000	0000001010	1101011111
17					0001001011	0000100101	0000000101	1111011010
18					0000100101	0000001100	1000000010	1111110011
19					1000010010	0010010000	1100000001	1101101111
20					1100001001	1000000001	0110000000	0111111110

TABLE XII. ALL REQUIRED SIGNATURES GENERATED FROM THE MISR BASED ON THE SECOND TEST PATTERN GENERATIVE APPROACH

Short-circuit No.	Dominant-1		Dominant-0		Dominant-1		Dominant-0		Short-circuit No.	Dominant-1	Dominant-0
	$SIGN_{(7\times7)}$	$SIGN_{(7\times11)}$	$SIGN_{(7\times7)}$	$SIGN_{(7\times11)}$	$SIGN_{(10\times10)}$	$SIGN_{(10\times20)}$	$SIGN_{(10\times10)}$	$SIGN_{(10\times20)}$		$SIGN_{(10\times10)}$	$SIGN_{(10\times20)}$
Fault-free	36	59	36	4f	1e9	17f	1e9	17f			
1	5a	04	73	3b	2a4	3a0	07b	205	61	00b	134
2	07	07	22	6f	1d4	04a	3bc	332	62	155	3f1
3	05	38	6b	01	1d9	1b6	101	0a7	63	310	1c9
4	61	48	61	4f	38f	09f	0db	2f4	64	32a	246
5	0a	1d	45	03	3d6	137	00f	293	65	2e7	035
6	5f	20	5d	16	26c	26b	02a	3fc	66	2e7	1e4
7	4f	70	5a	61	2d8	386	290	170	67	165	220
8	7d	46	44	27	1ff	033	1d8	330	68	066	142
9	13	47	7e	28	29f	301	3b7	170	69	315	3ce
10	34	01	4b	2b	0c2	256	2db	1b9	70	1b3	1f6
11	48	46	2d	7f	2aa	2ff	0a1	1f0	71	343	3e0
12	26	5e	23	6f	1ba	2f4	3be	1ea	72	0d3	33e
13	00	38	69	18	162	165	260	2dd	73	157	322
14	40	69	19	08	273	314	39b	051	74	0f6	1ae
15	5d	4d	08	09	04c	1f1	2be	269	75	001	396
16	07	38	3c	04	200	240	388	2b6	76	253	13f
17	02	78	00	70	26c	17e	351	1c3	77	319	28c
18	68	08	51	34	0e1	1c3	0de	04f	78	0d9	000
19	39	5f	4d	3b	0e1	16f	105	340	79	1b2	238
20	5d	0a	43	7a	25f	316	00c	164	80	176	2c9
21	3f	38	43	6d	06c	397	173	370	81	12e	045
22		75		3d	0de	003	268	16c	82	0b5	27d
23		7e		44	311	089	1c3	38c	83	362	2d0
24		7b		12	362	3d2	01a	3e8	84	237	0e8
25		3b		5d	29c	1ad	05a	104	85	219	2e7
26		7f		0a	28c	2fb	3f6	161	86	23c	0eb
27		33		11	13c	06a	297	319	87	1bc	3ea
28		6c		0a	1ce	317	10b	00a	88	0a1	357
29		1c		20	1f8	153	024	286	89	21a	29f
30		4b		3b	158	095	192	0be	90	061	28f
31		6c		6e	062	16c	3bc	083	91	1fe	016
32		7c		73	36a	30f	1b4	26f	92	31c	395
33		30		73	345	23d	2ac	28b	93	15e	119
34		45		73	218	1d7	3bf	03a	94	1c2	321
35		2c		4d	37a	3b7	1ea	26e	95	2b0	17c
36					36e	005	046	0e2	96	343	1f4
37					0ed	2af	354	114	97	248	364
38					2b7	3bd	1c6	212	98	125	3a1
39					19e	0dc	1ca	0d8	99	3da	1f1
40					3e1	34a	09f	28d	100	352	37d
41					3ea	21c	017	17a	101	094	360

Short-circuit No.	Dominant-1		Dominant-0		Dominant-1		Dominant-0		Short-circuit No.	Dominant-1	Dominant-0
	$SIGN_{(7\times7)}$	$SIGN_{(7\times11)}$	$SIGN_{(7\times7)}$	$SIGN_{(7\times11)}$	$SIGN_{(10\times10)}$	$SIGN_{(10\times20)}$	$SIGN_{(10\times10)}$	$SIGN_{(10\times20)}$		$SIGN_{(10\times10)}$	$SIGN_{(10\times20)}$
42					308	0ed	178	274	102	087	1f5
43					0a6	378	30c	04f	103	187	379
44					189	21c	181	336	104	12d	141
45					347	083	01c	3e7	105	30b	21c
46						3b3		3a4	106	3d7	194
47						265		063	107	3a3	365
48						1eb		2ef	108	07e	3c1
49						3c2		0d7	109	3c8	098
50						198		2c7	110	12b	11d
51						20f		2bf	111	100	027
52						04b		203	112	3b2	3d3
53						1df		08f	113	05f	099
54						3ec		2de	114	264	007
55						2e8		07e	115	39f	214
56						2ca		3ad	116	0c3	0b3
57						0f8		121	117	303	0aa
58						325		12d	118	357	261
59						039		3e8	119	123	0f6
60						36c		164	120	27d	263

Discussion: From simulation results, it is found that by using a decoder with an LFSR as the TPG instead of only LFSR in the first approach, the number of required clocks is minimized and the test application time is reduced. On the other hand, the hardware overhead problem is increased and the aliasing and the confounding syndromes are occurred due to the MISR as the TRC. However, the first approach based on an LFSR only as the TPG needs larger clocks with less hardware overhead and without the aliasing and the confounding syndrome due to the MISR as the TRC. Therefore, the fault coverage according to the presented approach in this paper to detect interconnect faults are greater than all previously published approaches [7, 9-14]. However, the test application time according to the presented approach in this paper is considered greater than the other approaches. Finally, the requirements to increase the fault coverage of interconnection faults without aliasing and confounding syndromes have higher priority than the accepted complexity of the testing system.

6. COMPARISON BETWEEN THE PRESENTED ALGORITHM AND THE OTHER ALGORITHMS

The presented new approach is used to generate the new PTVs to detect interconnect faults for the two and three short-circuits. The experiments were conducted on the seven nodes and ten nodes. Table XIII shows the comparison between the presented approach and the other previously published approaches [7, 9-14] in terms of the fault coverage (aliasing syndrome, confounding syndrome, and stuck-at fault detection), the generation of the PTVs for seven and ten tested nodes for two and three short-circuits, and the applicability of the BIST test scheme.

The results of the CSA [9] and the MCSA [10] have the poor fault coverage with low test application time. However, the MCSA can detect stuck-at faults. The results of the TCTDA [11] have poor fault coverage with low test application time. However, the TCTDA can detect stuck-at faults and there is no aliasing syndrome. The results of the WSA [12] have good fault coverage with the same test application time compared to the presented approach. It is applicable for both serial and parallel test scheme. However, the WSA cannot detect three short-circuits. The results of the WTAA [13] have good fault coverage with high test application time. However, the WTAA cannot detect three short-circuits. The results of the BSBTA [7, 14] have poor fault coverage with low test application time compared to the presented approach. However, the BSBTA can detect stuck-at faults. In addition, the authors in [7, 14] proposed certain condition. This condition cannot be properly implemented in the real hardware to replace each state '1' by the state 'X'. Without replacing “1” by “X” in [7, 14], the resulted aliasing syndromes and confounding syndromes are determined from Table XIV.

The *aliasing syndromes*: The resulted row of short-circuit N1N3 in Table XIV has the same STV of N3 in Table I. The resulted rows of the short-circuits N1N4, N1N5, and N4N5 in Table XIV have the same STV of N4 in Table I. The resulted rows of the short-circuits N1N6, N2N4, N2N6, N4N6, and N5N6 in Table XIV have the same STV of N6 in Table I. Finally, the resulted rows of the short-circuits N2N7, and N5N7 in Table XIV have the same STV of N7 in Table I.

The *confounding syndrome*: The resulted rows of short-circuits N1N4, N1N5, N4N5 in Table XIV have the same value "1001". The resulted rows of short-circuits N1N6, N2N4, N2N6, N4N6, and N5N6 in Table XIV have the same value "1101". The resulted rows of the short-circuits N1N7, N3N6, N3N7, N4N7, and N6N7 in Table XIV have the same value "1111". Finally, the

resulted rows of the short-circuits N3N4, and N3N5 in Table XIV have the same value "1011".

Finally, the presented results, shown in Table XIII, demonstrate the superiority of the presented approach in this paper, compared to the other approaches. Especially, the new approach can detect three short-circuits and locate the faults without any aliasing and confounding syndromes. In addition, it is applicable for both serial and parallel test scheme. Therefore, it has several improvements over all previously published approaches, related to interconnection fault detection and location of the PCB.

TABLE XIII. COMPARISON BETWEEN THE NEW TEST PATTERN SETS AND THE PERVIOUSLY PUBLISHED ALGORITHMS

Comparison Issues	New test pattern sets	CSA [9]	MCSA [10]	TCTDA [11]	WSA [12]	WTAA [13]	BSBTA [7, 14]
Aliasing syndrome	√	×	×	√	√	√	×
Confounding syndrome	√	×	×	×	√	√	×
Stuck-at fault detection	√	×	√	√	√	√	√
PTV for 7 tested nodes (two short-circuits)	7	3	4	8	7	11	4
PTV for 7 tested nodes (three short-circuits)	11	×	×	×	×	×	×
PTV for 10 tested nodes (two short-circuits)	10	4	4	8	10	14	×
PTV for 10 tested nodes (three short-circuits)	20	×	×	×	×	×	×
Applicability of the serial BIST test scheme	√	√	√	√	√	√	√
Applicability of the parallel BIST test scheme	√	×	×	×	√	×	×

TABLE XIV. THE RESULTED ROWS FROM OR OPERATION OF EACH ROW IN THE BSBTA, LISTED IN TABLE I.

No	Short-Circuit	Resulted Row	No	Short-Circuit	Resulted Row
1	N1N2	1100	12	N3N4	1011
2	N1N3	1010	13	N3N5	1011
3	N1N4	1001	14	N3N6	1111
4	N1N5	1001	15	N3N7	1111
5	N1N6	1101	16	N4N5	1001
6	N1N7	1111	17	N4N6	1101
7	N2N3	1110	18	N4N7	1111
8	N2N4	1101	19	N5N6	1101
9	N2N5	0101	20	N5N7	0111
10	N2N6	1101	21	N6N7	1111
11	N2N7	0111			

7. CONCLUSION

In this paper, the state-of-art algorithmic approach that generates the test pattern sets for interconnect fault detection is achieved without aliasing and confounding syndromes. It consists of two parts. The first part is the algorithm, implemented by the MATLAB code, to generate the applicable test pattern sets. It detects the target faults between any two (or three) terminals without aliasing and confounding syndromes. Due to huge computations of the exhaustive search, the analysis is achieved in the random search to generate the applicable test pattern sets, and to reduce the huge computation time of the exhaustive search. The computations became more and more difficult as the dimensions of the applicable test pattern sets are increased without aliasing and confounding syndromes. The applicable test pattern sets based on (7×7) matrix and (10×10) matrix for two short-circuits and (7×11) matrix and (10×20) matrix for three short-circuits are developed without aliasing and confounding syndromes. From the simulation results, the new test pattern sets can perform interconnect fault detection and locate the faults, occurred between any two (or three) terminals without aliasing and confounding syndromes.

The second part is the simulation and the implementation of the TPG using the Xilinx ISE Design Suite of the FPGA software to generate the applicable test pattern set based on the LFSR as the TPG and to compact the test response based on the MISR as the TRC. Therefore, different generators for the target applicable test pattern sets are developed for the interconnect fault detection of two (or three) terminals on the PCB. There are two approaches, discussed implementing the target applicable test sets. The first approach depends on the LFSR as the TPG and the MISR as the TRC. The second approach depends on the LFSR with a decoder as the TPG and the MISR as the TRC.

Based on the FPGA simulation results of both generative approaches, the first approach minimizes the hardware overhead of the required TPG circuitry but in the same time it needs more test application time for test set generation without aliasing and confounding syndromes. On the other hand, the second approach minimizes the test application time, but it increases the hardware overhead of the TPG circuitry and also it suffers from the problem of the aliasing and the confounding syndrome due to the MISR. From all previously published works stated in this paper, we can conclude the following:

- The stated applicable test pattern sets in this paper can be considered the most efficient test pattern vectors in the field of interconnect fault detection of the PCB.
- The presented approach to generate all applicable test pattern sets that detect interconnect faults without

aliasing and confounding syndromes with accepted hardware implementation and test application time. The problem of aliasing and confounding syndrome does not exist anymore in interconnect fault detection.

- The TPG, based on an LFSR with a decoder, improves the generation of the PTVs from test application time point of view but, on the other hand, it increases the hardware overhead and causes the problem of aliasing and confounding syndromes in the MISR.

REFERENCES

[1] Mohamed H. El-Mahlawy, Sherif Hussein, and Gouda I. Mohamed, "New Hybrid-Based Self-Test Strategy for Faulty Modules of Complex Microcontroller Systems", *Electronics Journal*, Vol. 22, No. 2, Dec. 2018.

[2] Mohamed S. Saleh, Mohamed H. El-Mahlawy, and Hossam E. Abou-Bakr Hassan, "Digital Signature Based Test of Analogueue Circuits Using Amplitude Modulated Multi-Tone Signals," 28th IEEE International Conference on Microelectronics (ICM 2016), pp. 117-120, Dec. 2016.

[3] Hui Luo, Wei Lu, Youren Wang, and Ling Wang, " A New Test Point Selection Method for Analog Continuous Parameter Fault", Journal of Electronic Testing, Vol. (33): pp. 339–352, May 2017.

[4] Shailaja Mantha, and Dr. M. Asha Rani, "Analog Testing Methodologies - A Survey, International Journal of Electronics and Communication Engineering and Technology (IJECET), 7(6): pp. 78–84, Nov.-Dec. 2016.

[5] R. Srinivasan, S. K. Gupta, and M. A. Breuer, "Novel test pattern generators for pseudoexhaustive testing," *IEEE Transactions on Computers*, vol. 49, pp. 1228-1240, 2000.

[6] Mahmoud S. Ragab, Mohamed H. El-Mahlawy and Emad A. El-Samahy, "Efficient Microcontroller System to Test an SRAM Chip Using Signature Analysis," 13th *International Computer Engineering Conference* (*ICENCO*), pp. 388-392, Dec 27-28, 2017.

[7] D.K. Sharma, Meerut, R.K. Sharma, B.K. Kaushik, and Pankaj Kumar, "Boundary scan based testing algorithm to detect interconnect faults in printed circuit boards," *Circuit World*, vol. 37, no. 3, pp. 27–34, 2011.

[8] Irith Pomeranz, "Functional Constraints in the Selection of Two-Cycle Gate-Exhaustive Faults for Test Generation," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 29, no. 7, pp. 1500-1504, May 2021.

[9] W. H. Kautz, "Testing for Faults in Wiring Networks," *IEEE Transactions on computers*, Vol. c-23, No. 4, pp. 358-363, April 1974.

[10] P. Goel and M. T. McMahon, "Electronic Chip-in-Place Test," 19th *IEEE Design Automation conference*, pp. 482-488, June 1982.

[11] Paul Wagner, ""Interconnect Testing with Boundary Scan", *IEEE International Test Conference*, Washington, DC, USA, 1987.

[12] Abu Hassan, Janusz Rajski and Vinod K.Agarwa, "Testing and Diagnosis of Interconnects using Boundary Scan Architecture," *IEEE International Test Conference*, Washington, DC, USA, 1988.

[13] Gordon D. Robinson and John G. Deshayes, "Interconnect Testing of Boards with Partial Boundary Scan," *IEEE International Test Conformance*, Washington, DC, USA, Sept. 1990.

[14] Pankaj Kumar, R. K. Sharma, D. K. Sharma and B. K. Kaushik, "A Novel Method for Diagnosis of Board Level Interconnect Faults Using Boundary Scan," *International Conference on Computer & Communication Technology*, (ICCCT'10), 2010.

[15] Artur Jutman, "At-Speed On-Chip Diagnosis of Board-Level Interconnect Faults," Proceedings of IEEE 9th European Test Symposium, May 2004.

[16] K. P. Parker, *the boundary-scan handbook*, 4th edition, Springer International Publishing, 2016.

[17] Mohamed H. El-Mahlawy, Ehab A. El-Sehely, Al-Emam S. Ragab, and Sherif Anas, "Design and Implementation of a New Built-In Self-Test Boundary Scan Architecture." *IEEE 15th International conference on Microelectronics*, pp. 27-31, 9-11 Dec. 2003.

[18] Paul H. Bardell, Willian H. McAnney, Jacob Savir, Built-In test for VLSI: pseudorandom techniques, John Wiley and Sons, 1987.

[19] S. W. Golomb, Shift Register Sequences, Laguna Hills, CA: Aegean Park Press, 1982.

[20] Mohamed H. El-Mahlawy, "Signature-Based Self-Test Approach for Single-Shot Circuits on the Circuit Board Level," 4th International Japan-Egypt Conference on Electronics, Communications and Computers (JEC-ECC 2016), pp. 38-42, May 31 - June 2, 2016.

[21] Mohamed H. El-Mahlawy, "Signature Multi-Mode Hardware-Based Self-Test Architecture for Digital Integrated Circuits," *IEEE International Conference on Electronics, Circuits, & Systems*, pp. 437-441, 6-9 Dec. 2015.

[22] Thomas W. Williams, Wilfried Daehn, Matthias gruetzner, and Cordt W. Starke, "Bounds and analysis of aliasing errors in linear feedback shift register," IEEE transactions on computer-aided design, vol. 7, no. 1, pp 75-83, Jan. 1988.

Tamer Sayed Abdel Aziz received a B.S. in Engineering from the military technical college, Cairo in 2005. He received a M.Sc. in the military technical college, Cairo in 2019. He is serving with the Egyptian armed forced for more than 20 years. His career works in the testing engineering.

Mohamed H. El-Mahlawy has been associate professor of electronics with the faculty of Engineering and Technology, Future university in Egypt since 2016 and has over 16 years in the military technical college as the assistant professor and associate professor. He is retired from the military technical college, where he was chairman of the biomedical engineering department. Prior to the military technical college, he received a Ph.D. in the design for testability of the VLSI circuits at University of Kent at Canterbury, UK in 2000. He received a M.Sc. in testing engineering from the military technical college in 1995. In addition, he received a B.Sc. in electronics and communication engineering from military technical college in 1989.