IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received July 26, 2020, accepted August 30, 2020, date of publication September 7, 2020, date of current version September 18, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3022027

Fine-Grained Defect Diagnosis for CMOL FPGA Circuits

JIHYE KIM, HAYOUNG LEE, (Graduate Student Member, IEEE), SEOKJUN JANG, AND SUNGHO KANG, (Senior Member, IEEE)
Department of Electrical and Electronic Engineering, Yonsei University, Seoul 03722, South Korea

Corresponding author: Sungho Kang (shkang@yonsei.ac.kr)

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2019R1A2C3011079).

ABSTRACT Nanotechnology is an important technological alternative to overcome the limitations of complementary metal-oxide-semiconductor (CMOS) technology. Various circuit implementation methods based on nanotechnology have been proposed, and their most important characteristics are a high defect ratio and defect tolerance through circuit reconfiguration. CMOS-nanowire-MOLecular (CMOL) field-programmable gate array (FPGA) circuits are advanced logic circuit structures that combine the advantages of CMOS and nanotechnology. However, researches on defect diagnosis methods for the reconfiguration of CMOL FPGA circuits are barely conducted. In this paper, efficient circuit configuration methods for defect diagnosis of the CMOL FPGA circuits are proposed to address the problem. Also, diagnosis algorithms for both stuck-at open and stuck-at close defects are introduced. Experimental results show that with the proposed methods, diagnosis is possible for CMOL fabrics with up to 20% stuck-at open defects and 0.02% or more stuck-at close defects.

INDEX TERMS CMOL, defect tolerance, diagnosis, nano-crossbar, nanotechnology.

I. INTRODUCTION

Lithography-based complementary metal-oxide-semiconductor (CMOS) technology is facing physical limitations owing to the increase in the complexity of electronic devices and the decrease in supply power [1]. In the past few years, nanotechnology based on bottom-up self-assembly has been extensively researched as an alternative to the next generation electronic systems [2]–[6]. Nanoscale components based on crossbars of nanowires or nanotubes can be assembled into arrays of low-power, high-density nanofabrics and can be implemented as functional circuits through post-manufacturing configurations. Although nanotechnology has the advantages of high density and low power, its defect rate (10^{-3} to 10^{-1}) is much higher than that of CMOS systems (10^{-12} to 10^{-9}) [7]. Therefore, an effective defect-tolerant approach for implementing circuits using nanotechnology is very important and essential. Among the various researches related to the nanoscale circuits, nano-programmable logic arrays (nano-PLAs) [8]–[10] and CMOS-nanowire-MOLecular (CMOL) hybrid circuits [11]–[13] are attracting considerable attention. Consequently, various defect-tolerant methodologies for nano-PLAs [9], [14], [15] and CMOL circuits [16]–[18] have been proposed. Accordingly, defect maps must be generated based on effective and accurate test and diagnosis for successful defect-tolerant logical mapping of nanoscale circuits.

The reconfigurable computer system Teramac [19] has shown the possibility of reliable computing systems, despite a considerable amount of defects through LFSR based diagnosis method. In [20], [21], defect testing with built-in self-test (BIST) methods has been proposed for nanofabrics. These methods and traditional CMOS-based programmable logic array (PLA) test and diagnosis techniques [22]–[24] can be modified and applied to nano-PLAs. However, researches on test and diagnostic methods considering the structural characteristics of CMOL circuits do not exist. The nano-crossbar architectures used in CMOL circuits and nano-PLAs have different structures. The nanowires used in nano-PLAs are connected from the input to the output. In contrast, the nanowires of CMOL circuits are cut off periodically. This characteristic is an advantage of CMOL circuits; however, test and diagnosis become complicated owing to the connection with limited adjacent cells. In the case

The associate editor coordinating the review of this manuscript and approving it for publication was Baoping Cai.

of CMOL memories having similar structures, defects can be directly tested and diagnosed on the nano-layer through write and read operations. However, in the CMOL-cell-based field-programmable gate array (FPGA)-like integrated circuits (CMOL FPGAs) proposed in [11], defects in nanodevices or nanowires should be observed through connected inverters in CMOL basic cells. If specific functional circuits such as LFSR are configured, only some parts of the CMOL FPGAs are used for test and diagnosis. Therefore, configurations that make overall crosspoints operated are necessary. The BIST methods can also be applied to CMOL FPGAs by dividing the blocks. However, the configurations and the test and diagnostic patterns for block under test (BUT) should be considered based on the structural characteristics of CMOL circuits. Therefore, all the defect types that can occur in the nano-crosspoint (hereinafter referred to as crosspoint), nano-crossbar (hereinafter referred to as crossbar), and CMOS layer should be modeled, and accurate diagnostic algorithms to find defect types and locations in the overall CMOL FPGAs are required. This is the first research to find the location of defects through circuit configurations that reflect the structural characteristics of CMOL FPGAs.

In this paper, fine-grained diagnostic methods for cell-based CMOL FPGAs are proposed to diagnose the type and location of defects in crosspoints accurately using efficient diagnostic logic mapping. The rest of this paper is organized as follows. Section 2 provides the relevant background, such as the structure of CMOL FPGAs and fault models. Section 3 presents efficient circuit configuration methods for test and diagnosis. Section 4 proposes effective test and diagnostic algorithms with the circuit configurations described in Section 3. Section 5 describes the experiments performed. Finally, Section 6 concludes this paper.

II. BACKGROUND

A. STRUCTURE OF CMOL FPGAS

As shown in Figure 1(a), a CMOL basic cell consists of the CMOS bottom layer including four transistors that constituting one inverter and two pass transistors, several metal layers, and a nanoelectronic add-on layer. Two directional nanowires (crossbars) are placed orthogonally to each other, and the crosspoints existing at their intersections are used to determine the connections between the inverters through the configuration process. During the configuration, Vdd is set to ground to disable the inverters [5]. When the appropriate voltages are applied through the pass transistors, the ON–OFF states of each crosspoint are set. After the configuration, the inverters are enabled, and the connected inverters operate as a specific functional circuit according to the configuration. In general, the CMOL basic cell can be represented by a simple symbol as shown in Figure 1(b) [11]. The blue and red dots indicate the inverter input and output pins connected to the crossbar, respectively. Only one crossbar is directly connected to each input (blue dot) and output (red dot) of the inverter. When the crosspoints on the crossbars are ON, the inverter is connected to the corresponding inverters through the crosspoints. As shown in Figure 1(c), 12 crosspoints on the crossbar are connected to the output pin (red dot) of the inverter cell in the dark gray box, and 11 connectable inverters (light gray boxes) correspond to each crosspoint. “Connectivity domain” [5], [11] of the inverter cell (dark grey box) is the area indicated by the red lines including the connectable inverters. Another connectivity domain is represented by the area indicated by the blue lines based on the input pin (blue dot) of the inverter. The shape and size of the connectivity domain are determined by the number of the crosspoints. In this paper, the connectivity domain fits within a square of dimensions 4 × 4, and only the crosspoint numbering based on the output pin (red dot) is used. The structure of the CMOL fabric is determined by the CMOS and nanotechnology process constraints. The number of crosspoints (M) on the crossbar is determined by the formula $M = 2r(r\text{-}1)$. Here, r is a positive integer and is typically large [5], [11]. However, for better understanding, simple cases with $r = 3$ are used as examples and experiments in this paper.

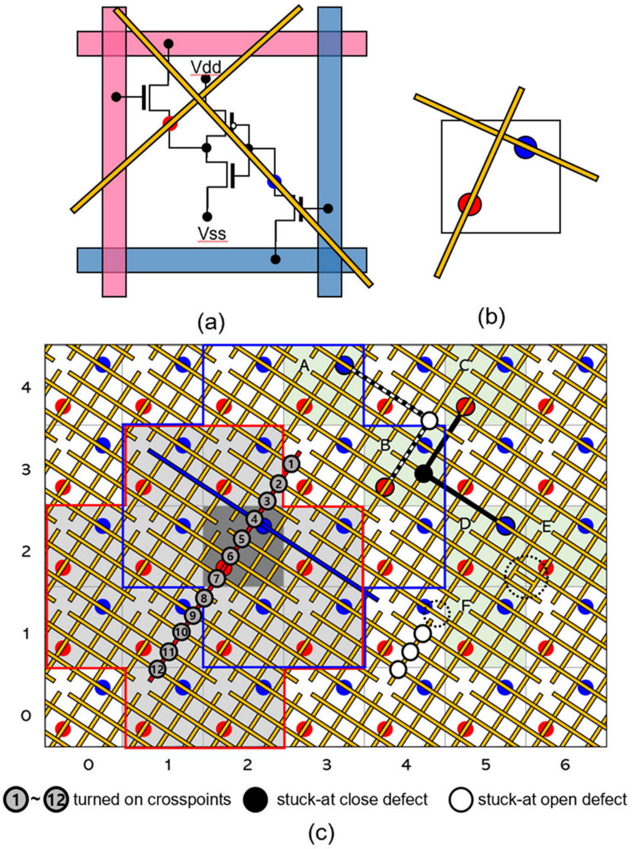


FIGURE 1. (a) Structure of the CMOL basic cell. (b) Symbol of the CMOL basic cell. (c) Structure of the CMOL FPGA, connectivity domain, and defects.

B. DEFECT MODELING

The following types of defects can occur in the CMOL FPGA structure: stuck-at open and close defects of crosspoints, broken and shorted crossbars, defective interface pins between the CMOS and the nano-layer, and transistor defects of the CMOS layer [18].

1) STUCK-AT OPEN DEFECTS

When stuck-at open defects exist at the crosspoints, the crosspoints are permanently turned off. As shown in Figure 1 (c), if a stuck-at open defect exists in the crosspoint connecting cells A and B, the output of cell B and the input of cell A can never be connected. If cells A and B are configured to be connected, the input of A cell becomes tied to 0, since the pass transistors of cell A are used as the pull-down resistors. In spite of the different types of defects in the CMOL fabric, there are cases where the defect phenomenon is similar to the stuck-at open defects at the crosspoints. If the inverters in the CMOS layer have stuck-at faults or some crosspoints are not configured due to the defects in the interface pins, the CMOL cell behaves as if it were fixed at the value 0 or 1. Moreover, when the crossbar connected to the output of cell D in Figure 1 (c) is broken, it can be interpreted as if stuck-at open defects exist at the continuous crosspoints. Like this, as various types of defects can be interpreted as stuck-at open defects at the crosspoints, and the defect ratio is very high (i.e., approximately up to 20% [25]), the location of the stuck-at open defects needs to be analyzed accurately for effective CMOL FPGA re-mapping. In addition, defect types can be classified in detail based on the results of the analysis.

2) STUCK-AT CLOSE DEFECTS

When stuck-at close defects exist at the crosspoints, the crosspoints are permanently turned on. As shown in Figure 1 (c), if a stuck-at close defect exists at the crosspoint connecting cells C and D, the output of cell C and the input of cell D can never be disconnected. Therefore, the input of cell D is always affected by the output of cell C. If the output crossbars of cells E and F are shorted to each other, the outputs of cells E and F will affect each other, similar to the stuck-at close defects at the crosspoints. As the defect rate of stuck-at close defects is relatively low (i.e., approximately 0.02% [26], [27]), some studies have focused only on open defects. However, in the case of a stuck-at close defect, the activation possibility of the defects is much higher than that of stuck-at open defects in a circuit. Moreover, stuck-at close defects directly affect the circuit structure and operation; hence, they need to be located accurately and then handled.

III. CONFIGURATION MAP GENERATION FOR EFFECTIVE TEST AND DIAGNOSIS

For effective re-mapping of CMOL FPGAs with high defect ratio, accurate defect maps must be generated. Therefore, effective test and diagnostic algorithms are required to generate defect maps. Accordingly, new circuits that facilitate test and diagnosis regardless of the functional circuit can be constructed by using the reconfigurable capabilities of CMOL FPGAs. In this section, methods for generating circuit configurations that improve the quality of test and diagnosis are proposed.

For testing, it is efficient to drive as many cells as possible at the same time. However, activating the crosspoints individually is effective to distinguish defects. If the appropriate configurations are applied to turn on only one crosspoint for each crossbar, inverter chains driving all the CMOL cells can be connected in various combinations.

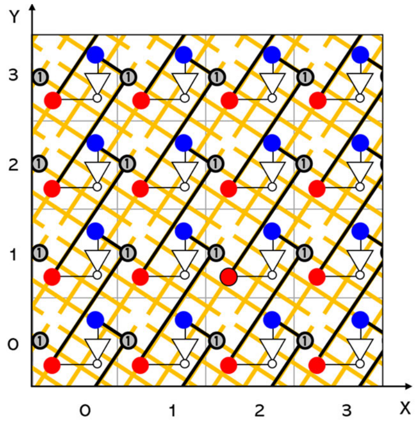


FIGURE 2. Inverter chain connections when only all the first crosspoints are ON.

A. SINGLE-CROSSPOINT ITERATIVE MAPS

Each CMOL cell can be expressed using coordinate information such as $C(X, Y)$. The turned on crosspoint number on the output crossbar of $C(X, Y)$ can be expressed as $XP(X, Y) = N$ (where N is a positive integer less than or equal to 12). For example, if only the first crosspoints among the 12 crosspoints on the output crossbar of all the cells are configured to be ON ($\forall\{XP(X, Y)\} = 1$), all $C(X, Y)$ is connected to $C(X, Y+1)$, as shown in Figure 2. Thus, the inverter chains in this case are formed in the y-axis direction. Note that the inverters of the basic cells are drawn for better understanding in Figure 2. As only one crosspoint among the 12 crosspoints on the output crossbar of each cell is turned on for the CMOL cell diagnosis in the proposed method, the configurations of the diagnostic circuit (inverter chains) can be simply expressed by the coordinates of each cell and the turned on crosspoint number of the cell. Therefore, the connection of the circuit (Figure 3) can be expressed using the configuration "MAP," which describes the crosspoint number to be turned on at the coordinate position of each cell. When the same crosspoint number is turned on for all the cells excluding crosspoint number 4, inverter chains are constructed in various combinations without any unconnected CMOL cell, as shown in Figure 3(a). When $XP(X, Y)$ is 4, as the output of the inverter is connected to its own input, the chain is not constructed; moreover, such a net connection is rarely used in most functional circuit configurations. Therefore, crosspoint number 4 is not considered in this research. Consequently, 11 types of single-crosspoint iterative MAPs can be obtained. However, more configurations with various

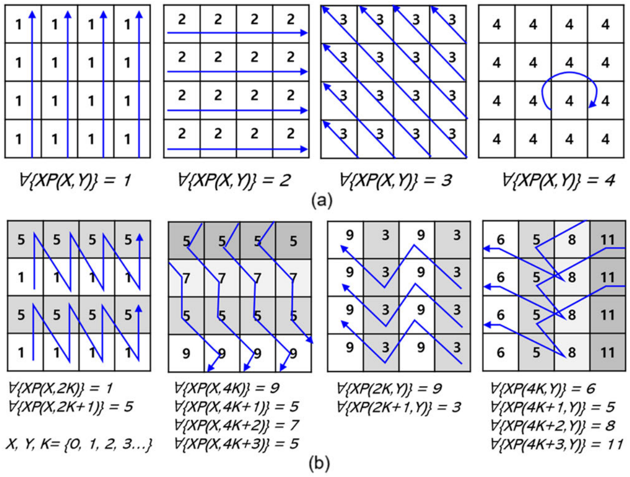


FIGURE 3. Various configuration MAP examples and inverter chain directions. (a) Single crosspoint iterative MAPs. (b) Multiple crosspoints iterative MAPs.

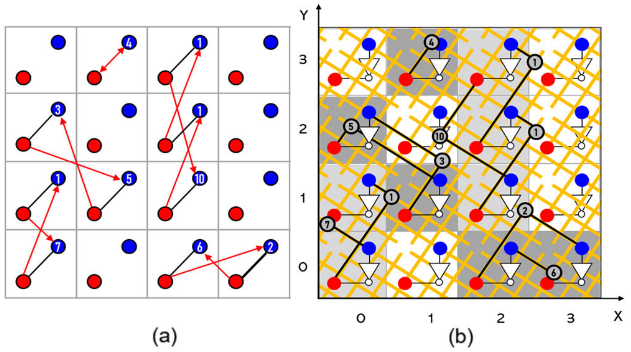
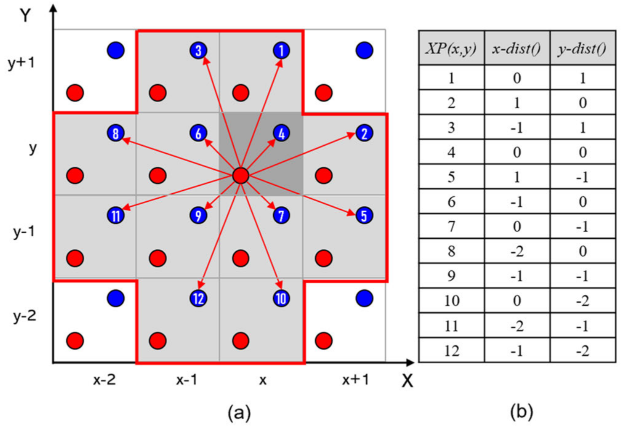


FIGURE 4. (a) Brief examples and (b) detailed examples of cell connections forming loops.



XP(x,y)	*x-dist()*	*y-dist()*
1	0	1
2	1	0
3	-1	1
4	0	0
5	1	-1
6	-1	0
7	0	-1
8	-2	0
9	-1	-1
10	0	-2
11	-2	-1
12	-1	-2

FIGURE 5. (a) Cell connections for each crosspoint in the X-Y coordinate domain. (b) X-Y axis distances for each crosspoint.

combinations of inverter chains are required for better diagnosis resolution.

B. MULTIPLE-CROSSPOINT ITERATIVE MAPS

The configuration MAPs that change the turned on crosspoint number in row, column, or diagonal units can be considered herein, as shown in Figure 3(b). Configurations having the same turned on crosspoints iteratively in row, column, or diagonal units have two advantages. First, each crosspoint has regular diagnostic accuracy, as all the operations. Second, the configuration time can be reduced as the rows, columns, or diagonals having the same turned on crosspoints and no overlapped connectivity domains can be programmed simultaneously. However, not all combinations of crosspoints that are turned on in row, column, or diagonal units form inverter chains. If the turned on crosspoint combinations that break the configuration of the inverter chains are excluded, the combinations of crosspoint numbers that can be used together in a configuration MAP are obtained. The configuration of the inverter chains is broken by two cases of turned on crosspoint combinations: when several cells are connected to form a loop and when the outputs of multiple inverters are connected to the input of the same cell.

1) LOOP CONDITION

As shown in Figure 4, certain cell combinations form a loop on their own. Figure 4(a) shows a brief and intuitive schematic of the crosspoint combinations forming a loop, and Figure 4(b) shows the same loop examples expressed in the CMOL structure. A loop is formed when the output of one cell is connected to its own input through other cells. "Distance" is the coordinate difference between the base cell $C(X, Y)$ and the connected cell $C(X', Y')$. When the crosspoint number to be turned on is determined, the relative position coordinates of the cell to be connected with the current cell can also be identified. The distance in x coordinates, X'-X, is expressed by *x-dist(XP(X, Y))*, and the distance in y coordinates, Y'-Y, is expressed by *y-dist(XP(X, Y))*. Figure 5 shows the cell connections in the output connectivity domain and the distances in x and y coordinates for each crosspoint. Based on the table in Figure 5, in the case of the crosspoint combination {3, 5} in Figure 4, *x-dist(3)* is -1, *x-dist(5)* is 1, and the summation of these distances is 0. The summation of the distances in y coordinates is also 0. For all other combinations in Figure 4, the summations of the *x* and *y* distances are 0. In summary, when *L* cells form a loop, the summations of all *x-dist* and *y-dist* of all the cells are 0.

Therefore, the loop condition is expressed as follows:

$$SUM\{x - dist(XP(x_{1,}y_1)), \ldots, x - dist(XP(x_{L,}y_L))\} = 0 \ \&$$
$$SUM\{y - dist(XP(x_{1,}y_1)), \ldots, y - dist(XP(x_{L,}y_L))\} = 0$$

For row crosspoint iterative MAPs, the crosspoint combinations for which the summation of *y-dist* is 0 should be excluded. For column crosspoint iterative MAPs, the crosspoint combinations for which the summation of *x-dist* is 0 should be excluded. For diagonal iterative crosspoint MAPs, the crosspoint combination for which the summations of both *x-dist* and *y-dist* are 0 should be excluded.

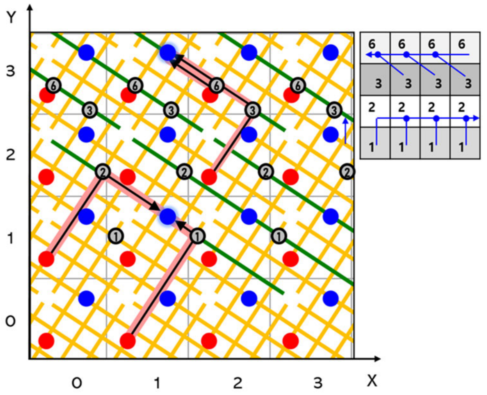
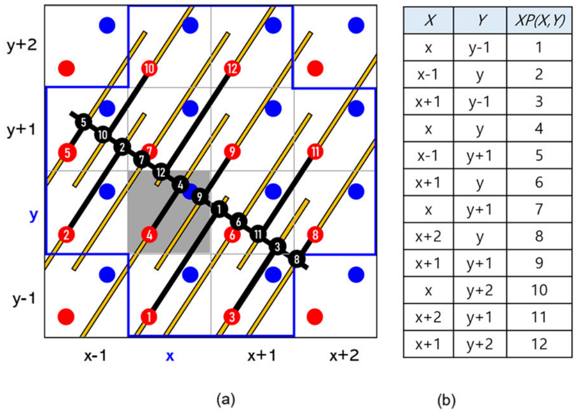


FIGURE 6. Example of multiple fan-in occurrence.



X	*Y*	*XP(X,Y)*
x	y-1	1
x-1	y	2
x+1	y-1	3
x	y	4
x-1	y+1	5
x+1	y	6
x	y+1	7
x+2	y	8
x+1	y+1	9
x	y+2	10
x+2	y+1	11
x+1	y+2	12

FIGURE 7. (a) Relationship between the crosspoint numbers located on the input crossbar of *C(X,Y)* and the neighboring cells and (b) the coordinate and crosspoint numbers of the neighboring cells that can be connected to the input of *C(X,Y)*.

2) MULTIPLE FAN-IN CONDITION

In order to form inverter chains, the output of one cell must be connected to the input of another one cell. As shown in Figure 6, when a row crosspoint iterative MAP exists in which {1, 2, 3, 6} is repeated, crosspoint 1 of the cells in the first row and crosspoint 2 of the cells in the second row are turned on so that the outputs of two cells are simultaneously connected to the input of one cell. And crosspoint 3 and crosspoint 6 are turned on so that the outputs of two cells are simultaneously connected to the input of one cell. Two crosspoints on the green input crossbars are turned on simultaneously in Figure 6. The inverter chains are not constructed in these cases. Combinations that cannot construct inverter chains can be determined based on the crosspoint numbers that can be turned on simultaneously in one input crossbar and the positional relationship of neighboring cells that can be connected through the crosspoints. Figure 7 shows the coordinates of the neighboring cells and the crosspoint numbers of the cells to be connected to the reference cell, *C(X, Y)*, within the input connectivity domain. The row crosspoint iterative MAPs should be generated considering the y-coordinate relationships and crosspoint numbers in the table of Figure 7. For example, if the crosspoint number of the first row is {1}, the crosspoint number of the second row, third row, and fourth row cannot be {2, 4, 6, 8}, {5, 7, 9, 11}, and {10, 12}, respectively. The column iterative crosspoint MAPs should be generated considering the x-coordinate relationships and crosspoint numbers. Further, the diagonal crosspoint iterative MAPs should be generated considering crosspoint numbers with x-coordinate and y-coordinate relationships.

If the combinations that form the loops and multiple fan-in connections are excluded crosspoint numbers from the multiple-crosspoint iterative combinations, MAPs with various inverter chain configurations can be generated. As the size of the connectivity domain is 4 × 4 in this paper, MAPs iterated with two to four crosspoints can be generated to configure various connections of the crosspoints within one connectivity domain. In the diagonal cases, MAPs iterated with up to 5 crosspoints can be generated.

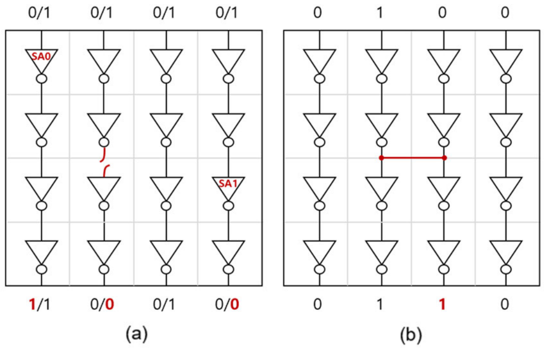


FIGURE 8. Applying (a) all-zero/-one pattern and (b) one-hot pattern for testing inverter chains.

IV. TEST PATTERN GENERATION AND DIAGNOSTIC ALGORITHMS

A. TEST PATTERN GENERATION

If configurations that constitute various combinations of inverter chains are prepared, then appropriate test patterns that can test the circuits composed of inverter chains need to be generated. The hard defects existing in the inverter chains are tested by measuring the outputs by applying 0 or 1 to the input of the chains. As shown in Figure 8 (a), stuck-at faults in the inverter cells and open defects of the crosspoints that cause broken chains can be detected by all-zero and all-one patterns, as they affect only the chains in which they exist. However, stuck-at close defects that cause shorts between neighboring chains are not sufficiently tested with all-zero and all-one patterns. The defect phenomena of the stuck-at close defects may or may not appear depending on the states (values) of neighboring cells. Hence, the outputs should be observed by changing the input states (values) independently to ensure that the defect phenomena can be revealed. Therefore, as shown in Figure 8(b), if only one difference value is applied to the target chain sequentially, such as walking-one and walking-zero patterns, the test time increases but all the hard defects occurring in a circuit can be detected. In addition, this case is advantageous for diagnosis by reducing the possibility of several defect phenomena which occurs at the same time as the changes of the cell states inside the circuit are reduced when compared with the case where the inputs of all the inverter chains are changed simultaneously. Nevertheless, as multiple defects exist in the circuit, aliasing may occur in some test patterns, but accurate test and diagnosis are possible if sufficient circuit configurations are applied. Note that the input values of the chains whose connectivity domains are not overlapped, can be changed simultaneously to reduce the test time.

B. CMOL CELL DIAGNOSIS

In this section, the diagnostic algorithms for multiple crosspoint defects are introduced. The diagnostic algorithms are premised on the use of the configuration MAPs introduced in Section 3 and the walking-one and walking-zero test patterns described in Section 4.1.

In general, as the defect rate is low in CMOS circuits, a single fault is assumed for efficient test and diagnosis. Therefore, the failure response is regarded as a phenomenon caused by one failure, and the failure location is analyzed by comparing the results of good simulation and test failure results. However, it is natural that multiple defects exist in CMOL FPGAs, and the failure output response can be assumed to be the result of a mixture of various failure phenomena. Therefore, to diagnose CMOL FPGAs, using the output responses obtained by applying specific input patterns such as all-zero or all-one patterns to the circuit with defects as the reference outputs is more effective than using the good simulation results. Then, the output responses obtained by changing the input values sequentially are observed and compared with the reference outputs to analyze the defect locations. The diagnostic algorithms for stuck-at open and stuck-at close defects are different; hence, each algorithm is described separately. Diagnosis method for the case where the two types of defects exist simultaneously is discussed in Section 5.

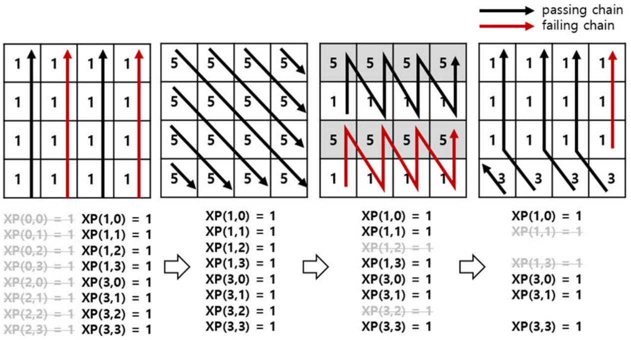


FIGURE 9. Stuck-at open defect candidate reduction with various circuit configurations.

Algorithm 1 Proposed Stuck-At Open Defect Diagnosis

```
Specify the number of configuration MAPs, $N_{MAP}$
for $i = 1$ to $N_{MAP}$ do
  Specify the number of inverter chains, $N_{chain}$,
      for the current configuration
  for $j = 1$ to $N_{cahin}$ do
    $I[j] = 0$            // $I[j]$: input of the chain $j$
  end for
  for $j = 1$ to $Nchain$ do
    $O_{ref}[j] = O[j]$        // $O_{ref}$: reference outputs
                               // $O[j]$: output of the chain $j$
  end for
  for $j = 1$ to $N_{chain}$ do
    if $j! = 1$ do $I[j-1] = 0$ endif
    $I[j] = 1$
    if $O_{ref}[j]! = O[j]$ do
      for $XP(X,Y)$ used for constructing target chain $j$ do
        $C(X,Y) \rightarrow > XP(X,Y) \rightarrow SCOREopen+1$
      end for
    end if
  end for
end for
for all $XP(X,Y)$ in the CMOL FPGAs do
  inv_$sort(C(X,Y)->XP(X,Y)->SCOREopen$
end for
```

1) DIAGNOSIS OF STUCK-AT OPEN DEFECTS

A stuck-at open defect of crosspoints is the most frequently occurring defect in CMOL structures. When multiple defects exist in the circuit, it is difficult to accurately diagnose the location due to the mixture of defect phenomena. However, stuck-at open defects affect only the inverter chain where they exist, and the failure phenomenon is not propagated to other chains. Therefore, even if a large number of stuck-at open defects exist, accurate fault diagnosis is possible if sufficient circuit configurations are provided to distinguish them. If no stuck-at open defect exists in an inverter chain, the output value is changed when the input of the chain is changed. However, if a stuck-at open defect exists at a specific crosspoint, the inverter chains formed through the crosspoint cannot transfer the value from input to output as the inverter chains are cut off in the middle and the output values are not altered even if the input values are changed. Therefore, if the output of the inverter chain is altered by the input transition, it can be regarded that there is no hard defect in the inverter chain regardless of whether the output responses are pass or fail. This indicates that the crosspoints used for connecting the inverter cells to construct the chain have no stuck-at open defect. When crosspoints without defects are excluded from the candidate list (Figure 9), substantial crosspoint candidates with stuck-at open defects can be obtained. Algorithm 1 is the proposed diagnosis algorithm for stuck-at open defects. This algorithm is applied with walking-one and walking-zero test patterns to the inverter chains formed according to the configuration MAPs. And each crosspoint is scored depending on whether the output is altered. Finally, crosspoints with a score of 0 indicate that no output of inverter chains through the crosspoints has ever been altered, and these crosspoints are the most likely defect candidates.

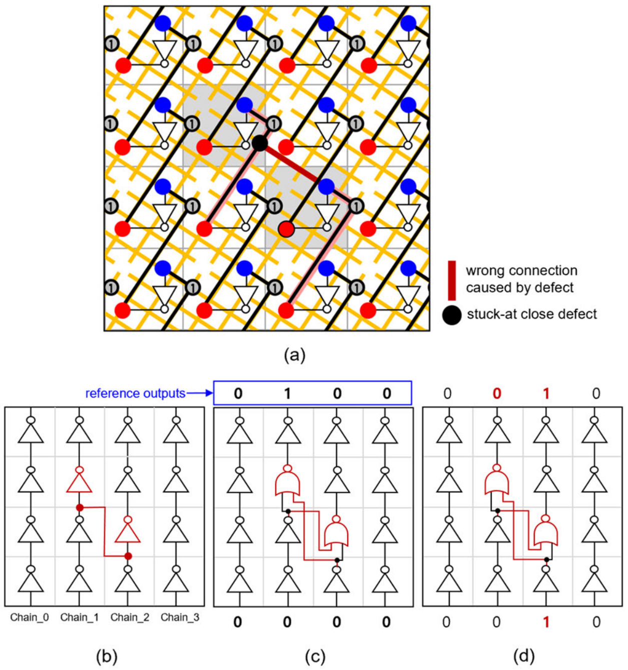


FIGURE 10. (a) Circuit constructed using ∀{$XP(X,Y)$) = 1 configuration MAP with stuck-at close defect and (b) its symbolic schematic. (c) Reference output values of all-zero inputs to the circuit modified under the influence of a stuck-at close defect. (d) Output transition monitoring with one-hot input pattern.

2) DIAGNOSIS OF STUCK-AT CLOSE DEFECTS

If stuck-at close defects exist at the crosspoints of a cell, they affect not only the inverter chain containing the cell but also other adjacent chains. Therefore, if several stuck-at close defects exist in the circuit, diagnosing the exact location may be difficult because of the mixed defect phenomena. However, the occurrence frequency of stuck-at close defects is significantly lower than that of stuck-at open defects. Therefore, accurate fault diagnosis can be performed by using test results for various circuit configurations and the proposed diagnosis algorithm.

In CMOL FPGAs, when the outputs of two inverter cells are connected to the input of one inverter cell, the input connected inverter cell becomes having the functionality of wired-NOR [11]. Therefore, if an additional connection is generated to the existing chain connection due to the stuck-at close defect, the connected inverter cells behave similarly to the NOR cells as shown in Figure 10. When the circuit configured by the configuration MAP, ∀{$XP(X, Y)$) = 1, has a stuck-at close defect, as shown in Figure 10(a), the two inverter chains are incorrectly connected. This can be symbolically represented as inverter chains with a wrong connection between chain_1 and chain_2 as shown in Figure 10(b). Also, the incorrectly connected inverter cells in the circuit behave like NOR gates and the output responses to all-zero inputs can be set as reference outputs regardless of whether they are pass or fail, as shown in Figure 10(c). In this case, when input of chain_2 is changed from 0 to 1, not only output of chain_2 but also output of chain_1 are changed at the same time as shown in Figure 10(d). In this situation, it can be checked whether an incorrect connection exists between chain_1 and chain_2. At this time, all the crosspoints that can connect the two chains become the candidates for stuck-at close defects, and defect scores are added to each candidate. When these test results are analyzed for various circuit configurations, the scores of the crosspoints with real defects are gradually increased, and finally substantial candidates are obtained. Algorithm 2 is the proposed algorithm for diagnosing stuck-at close defects. For diagnosing the stuck-at open defects, the output transition of the chain whose input is altered is measured. In contrast, for diagnosing the stuck-at close defects, the output transition of other chains whose inputs are not altered is measured. The score of defective crosspoints becomes very high as the fault phenomenon always occurs except for the configuration in which the crosspoint with the stuck-at close defect is turned on. Therefore, the real defects have higher possibility to exist among candidates with the highest score.

Algorithm 2 Proposed Stuck-At Close Defect Diagnosis

```
Specify the number of configuration MAPs, $N_{MAP}$
for $i = 1$ to $N_{MAP}$ do
  Specify the number of inverter chains, $N_{chain}$, for the current configuration
  // Applying walking one test patterns
  for $j = 1$ to $N_{cahin}$ do
    $I[j] = 0$                  // I[j]: input of the chain j
  end for
  for $j = 1$ to $Nchain$ do
    $O_{ref}[j] = O[j]$             // $O_{ref}$: reference outputs
                                    // $O[j]$: output of the chain j
  end for
  for $m = 1$ to $N_{chain}$ do
    if $m! = 1$ do $I[m\text{-}1] = 0$ endif
    $I[m] = 1$
    for $n = 1$ to $N_{chain}$ do
      if $m! = n$ & $Oref[n]! = O[n]$ do
        for $C(X,Y)$ in target chain m do
          for $C(X',Y')$ in connected chain n do
            if $C(X',Y')$ is in connectivity domain of $C(X,Y)$ do
              $XP(X',Y') = find\_XP(C(X,Y),C(X',Y'))$
              $C(X',Y') \rightarrow XP(X',Y') \rightarrow SCOREclose{+}1$
            end if
          end for
        end for
      end if
    end for
  end for
end for
for all $XP(X,Y)$ in the CMOL FPGAs do
  $sort(C(X,Y) \rightarrow XP(X,Y) \rightarrow SCOREclose$
end for
```

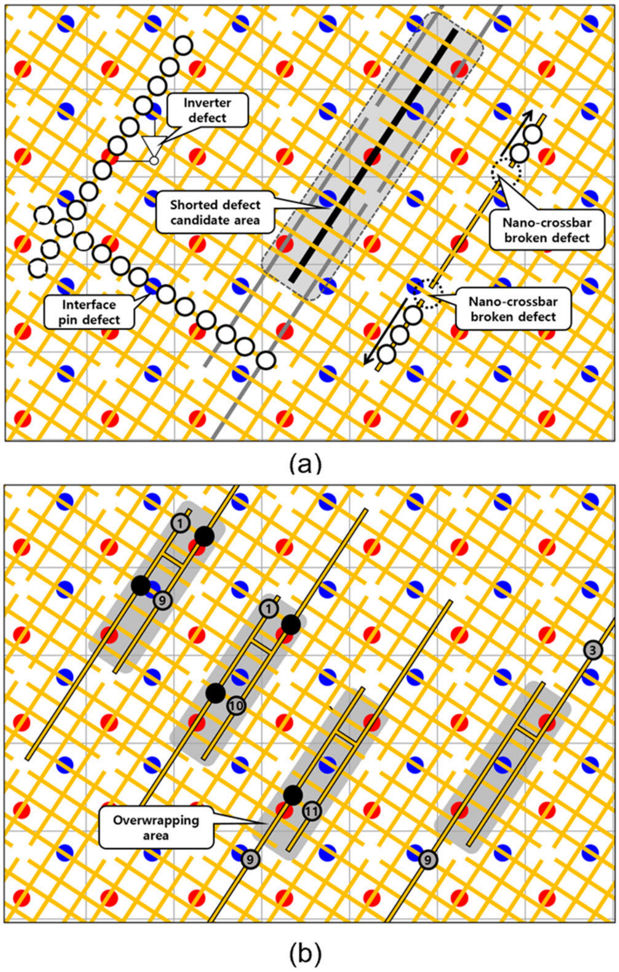

Inverter defect
Shorted defect candidate area
Nano-crossbar broken defect
Interface pin defect
Nano-crossbar broken defect
(a)
1
9
1
10
3
Overwrapping area
11
9
9
(b)

FIGURE 11. (a) Types of defect phenomena. (b) Defect phenomena of the crossbar shorted by the circuit configuration.

C. POST PROCESSING FOR DEFECT TYPE CLASSIFICATION

When algorithms for diagnosing stuck-at open and close defects are applied, scores for open and close defects are given to each crosspoint of the cells on the CMOL fabric. As shown in Figure 11(a), even if defects exist in the CMOS layer, interface pins, or crossbar, the defect effects are expressed during the test process and scored. Therefore, further defect categorization can be possible via post-processing of the diagnosis results. Meanwhile, if a hard defect exists in the inverter cell, the data propagation of the inverter chain is stopped at the cell even if any crosspoint of the cell is turned on. Therefore, all the crosspoints of inverter cells with defects have an open defect score of zero. If the interface pin is defective, ON-OFF control of the cross points, which must be programmed via the interface pin, is not performed, and the connection between the inverter cell and the nano-layer is broken. Therefore, if a defect exists in the input (output) side interface pin, it is observed that the open defect scores for the crosspoints on the input (output) crossbar are all zeros. In the case of crossbars, a broken defect and a shorted defect between the two crossbars can exist. The interface pin is generally connected to the middle of the crossbar. Therefore, when the crossbar is broken, the crosspoints existing on the opposite side of the interface pin from the broken spot behave as if stuck-at open defects exist to ensure that all the open defect scores of the crosspoints become zero. Meanwhile, it is assumed that shorted defects occur only in adjacent parallel crossbars on the same layer (shorted defect candidate area). Two crossbars are shorted to each other, and a similar phenomenon as stuck-at close defects is observed. However, the locations at which the defect phenomenon occurs are changed depending on the circuit configurations, as shown in Figure 11(b). If real stuck-at close defects exist, failure phenomena are observed at the same locations for most configurations. Therefore, the close defect scores of the crosspoints with the real close defect become very high. However, the close defect scores for shorted defects of crossbars may or may not be added depending on the configuration. Therefore, the close defect scores of each crosspoint are not higher than that of the crosspoint where the real stuck-at close defect exists. However, when the scores of the crosspoints located in the area where the shorted crossbars are overlapped are added up, the summation result is higher than that of the area without shorted defects. Therefore, the shorted defect can be estimated by the summation of the close defect scores of the specific crosspoints based on the applied circuit configurations during the test.

V. EXPERIMENTAL RESULTS

Figure 12 shows the experimental process for evaluating the performance of the proposed method. This method constructs inverter chains according to various generated MAPs, inserts defects, performs fault simulation, and performs diagnosis according to defect types to score each crosspoint and to select the final defect candidates. Experiments were

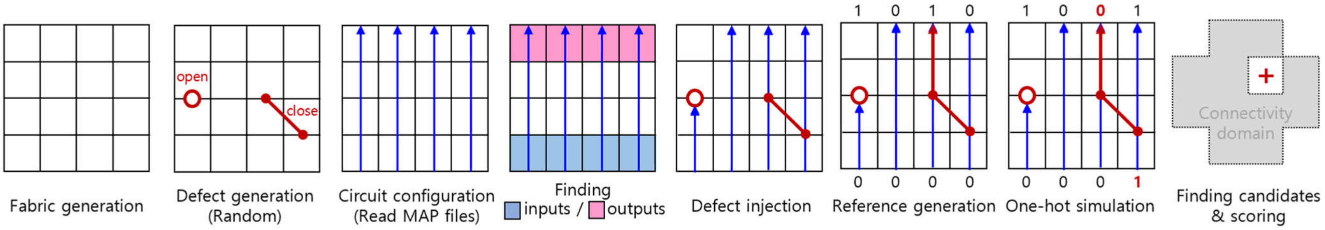


FIGURE 12. Experimental process for diagnosing stuck-at open and close defects of the crosspoints in CMOL FPGAs.

conducted by changing the CMOL fabric size from 5 × 5 to 30 × 30 depends on cases. Defects were randomly generated for the cases of stuck-at open defects only, stuck-at close defects only, and a mixture of stuck-at open and close defects. The experimental results are the average values obtained over 100 repetitions for each case.

A. MAP SELECTION

In Section 3, the method to generate various MAPs was introduced. When MAPs are generated using this method, 11 cases of single-crosspoint iterative MAPs can be generated. Furthermore, multiple-crosspoint iterative MAPs can be generated for four cases, namely, row, column, diagonal, and orthogonal diagonal. For two-, three-, and four-crosspoint iterative MAPs, 134, 636, and 4218 MAPs can be generated, respectively. Although the MAPs to be used can be optimized in several ways, the performance of the MAPs was evaluated by applying the MAPs cumulatively by type, starting from single-crosspoint iterative MAPs. In the experiments, the fabric size was 10 × 10, and stuck-at open defects were applied as representatives in the range 5%–20%. For example, an insertion of 20% defects indicates that 240 defects were inserted among 1,200 crosspoints. As the open defect scores of the crosspoints with open defects will be 0, the performances were normalized by dividing the candidates with a score of 0 by the actual number of defects. If all the defects are found correctly, the normalized performance index becomes 1. In Figure 13, single-XP represents the case in which only single-iterative MAPs are applied. 2-XP represents the case where both single- and two-crosspoint iterative MAPs are applied, and 3-XP (4-XP) represents the case where 2-XP (3-XP) MAPs and three (four)-crosspoint iterative MAPs are applied. The numbers in parentheses are the numbers of MAPs applied in each case. As shown in Figure 13, the diagnostic resolution is improved as more MAPs are applied. However, the resolution of 3-XP is similar to that of 4-XP, but the number of applied MAPs is less than one-sixth that of the latter. Therefore, in the experiments, 3-XP MAPs (781) were applied and additional MAPs were applied if required.

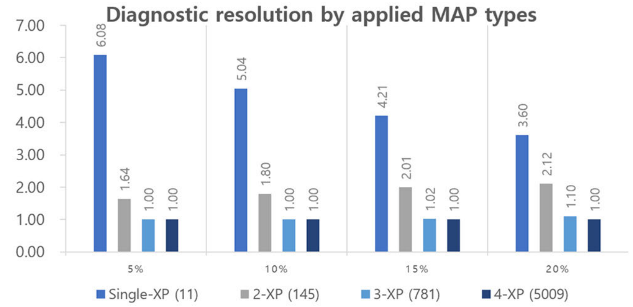


FIGURE 13. Diagnostic resolution by applied MAP types.

B. EXPERIMENTAL RESULTS FOR STUCK-AT OPEN DEFECTS

The diagnostic resolution for the stuck-at open defects was evaluated with the CMOL fabric sizes ranging from 5 × 5 to 30 × 30. Stuck-at open defects were randomly generated with the defect ratio ranging from 5% to 20% of the total number of crosspoints. The criteria may vary depending on the process maturity such as defect ratio and yield. However, in this paper, a normalized performance index up to approximately 1.25 is considered acceptable. Figure 14 shows the normalized performance indexes of the stuck-at open defect diagnostic algorithm. For example, the performance index of the 10% in a fabric size 15 × 15 is 1.03. This means, when stuck-at open defects are inserted in 270 crosspoints (10%) among 2,700 crosspoints in a fabric of size 15 × 15, all the defects are found within an average of 278 candidates. As shown in Figure 14, despite the insertion of numerous defects, the diagnostic algorithm for stuck-at open defects shows excellent performance in cases with a defect ratio ranging from 5% to 10% or a fabric size ranging from 5 × 5 to 15 × 15. To improve the resolution with a defect ratio of 20%, additional four-crosspoint iterative MAPs are applied and analyzed. Consequently, for a size of 20 × 20, the performance index is slightly higher than 1.25, but the performance indexes are reduced as shown in Figure 15.

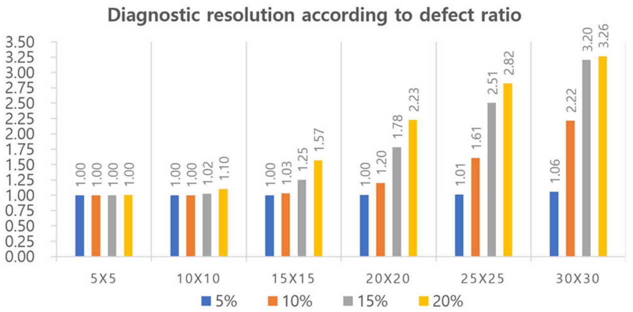


FIGURE 14. Normalized diagnostic performance indexes for stuck-at open defects.

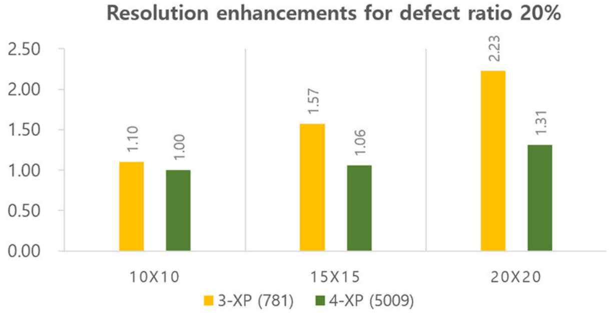


FIGURE 15. Resolution enhancements with additional configuration MAPs.

In this paper, for better understanding, cases having a very small connectivity domain of 4 × 4 with 12 crosspoints ($M = 2r(r\text{-}1)$, $r = 3$) on one crossbar are assumed. If the size of the connectivity domain is small, the length of the chains becomes longer when the inverter chain is configured. Figure 16 shows the average length of all the inverter chains constructed using the configuration MAPs for each CMOL fabric size. The x-axis is the length of one side of the square

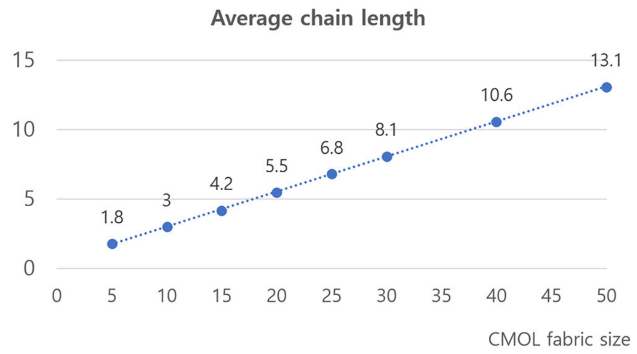


FIGURE 16. Average inverter chain length by CMOL fabric size.

fabric. This is similar to the value obtained by adding the constant (in this case, 0.5) to the length of one side of the fabric divided by the length of one side of the connectivity domain. The average length of the inverter chain of the CMOL fabric of size 20 × 20 is 5.5. When a defect ratio of 20% is assumed, consequently, it can be considered that approximately one defect exists in an inverter chain on average. If multiple defects exist in one inverter chain, the diagnostic resolution decreases. Therefore, in the experiments, because the connectivity domain of the CMOL fabric was small, it could be diagnosed up to the size of 20 × 20. However, in the case of a practical CMOL structure, r is greater than 12 [5], and the connectivity domain is configured to be larger than 16 × 16 in size. Therefore, configuration MAPs capable of diagnosing a fabric of size up to 80 × 80 can be generated. Moreover, if the defect ratio is reduced with process enhancements, a larger-sized CMOL fabric can be diagnosed. In addition, as the CMOL fabric has a regular structure, it can be divided into a size suitable to perform the test and diagnosis of the CMOL structure.

C. EXPERIMENTAL RESULTS FOR STUCK-AT CLOSE DEFECTS

The performance of the diagnosis algorithm for stuck-at close defects is expressed as the ratio of candidate counts and the actual defect counts. For example, when three defects are inserted, the normalized performance index becomes 2 if six candidates with high close defect scores contain all the actual defects. As in the case of the stuck-at open defect, the index becomes 1 when the defect is diagnosed accurately. Stuck-at close defects were inserted at a rate more than or equal to 0.02%. Here, 0.02% is the general rate of stuck-at close defects, which is very small. Moreover, the number of inserted defects cannot be applied in decimal units. Therefore, a different rate more than or equal to 0.02% was applied depending on the total number of crosspoints. For the stuck-at close defects, experiments were performed on CMOL fabrics of sizes ranging from 10 × 10 to 30 × 30. In Figure 17, the number in parentheses indicates the inserted stuck-at close defect ratio. From the experiment, it is observed that the normalized diagnostic performance indexes are very close to 1 for all sizes of CMOL fabrics. As the defect ratio is very low and only stuck-at close defects exist, most defects are accurately found.

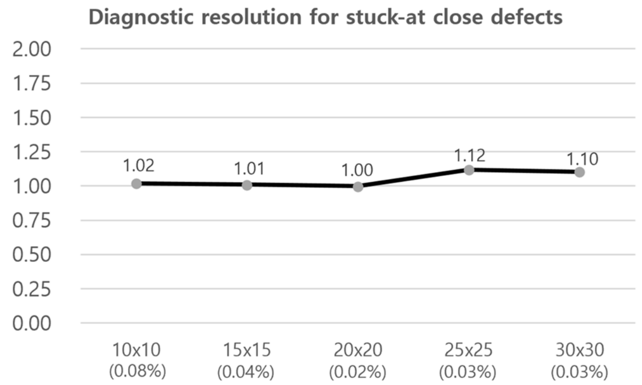


FIGURE 17. Normalized diagnostic performance indexes for stuck-at close defects.

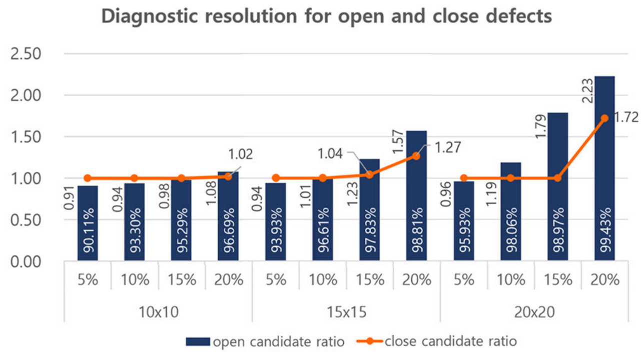


FIGURE 18. Diagnostic resolutions for the cases with a mixture of stuck-at open and close defects.

D. EXPERIMENTAL RESULTS FOR A MIXTURE OF STUCK-AT OPEN AND CLOSE DEFECTS

In the case with a mixture of defects, stuck-at close defects in the same ratio as in Section 5.3 were inserted. At the same time, stuck-at open defects were inserted by changing the ratio from 5% to 20%. The same fault simulation results were analyzed using the two diagnosis algorithms for the open and close defects, respectively. The experiments were performed for CMOL fabric sizes ranging from 10 × 10 to 20 × 20. In Figure 18, the navy bars represent the diagnosis results for open defects, and the orange line represents the diagnosis results for close defects. The value of the navy bars indicates the normalized performance index, which is the ratio of the number of candidates with an open defect score of 0 to the actual number of stuck-at open defects. The numbers inside the bars show the percentage of actual stuck-at open defects included in the candidates with a score of 0. More than 93% open defects were found in most cases. The best diagnostic result is a case where the performance index is 1 and the real defect ratio is 100%. The values on the orange line represent the normalized performance indexes of the stuck-at close defects used in Section 5.3. These values are in the range of 1 to 2 on average. In other words, twice as many

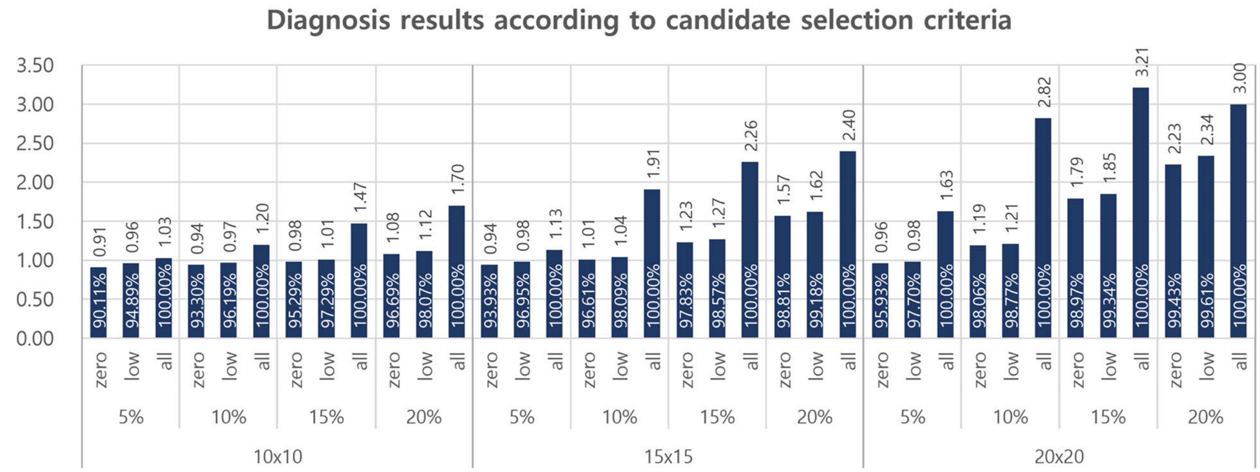


FIGURE 19. Comparison of the resolution of stuck-at open defect diagnosis according to candidate selection criteria.

candidates can be selected as the final candidates, considering the expected close defect ratio in the process. When close defects are surrounded by open defects, the close defect score can be very low. Therefore, if the overall close scores are lower than those of other samples, the final candidates can be safely selected with a sufficient margin. In this experiment, four outlier samples were excluded in the 20 × 20 cases.

When the stuck-at open and close defects are mixed, these two types of defects influence each other, and the resolution decreases. If the resolution should be increased, candidates with non-zero open defect scores should be included in the final candidates. In the experiment shown in Figure 19, the ratio of candidates and the actual defects in candidates were measured while relaxing the criteria for selecting the final open defect candidates. ‘‘Zero,’’ ‘‘low,’’ and ‘‘all’’ on the horizontal axis label indicate the final candidate selection criteria. ‘‘Zero’’ indicates the cases in which candidates with an open defect score of 0 are selected as the final candidates. ‘‘Low’’ indicates the cases in which candidates having scores ranging from 0 to the lowest non-zero value are selected. ‘‘All’’ indicates the cases in which the criteria have been relaxed to the extent that all the defects are included in the final candidates. When only stuck-at open defects exist, the open defect scores of the crosspoints with defects must be 0. However, in the cases with a mixture of defects, the output values of the inverter chains with stuck-at open defects may be altered due to the effects of stuck-at close defects depending on the configuration. Therefore, the open defect scores of crosspoints with stuck-at open defects may not be 0. If candidates with the open defect scores of 0 are selected as the final defect candidates, some stuck-at open defects are omitted. When comparing the values of ‘‘zero’’ and ‘‘low’’ cases, the number of candidates is increased a little, but the actual defect ratio is improved for the ‘‘low’’ case. If the score criterion is relaxed until all the actual open defects are found, a lot of candidates are added. When the ‘‘all’’ cases are compared with the cases with only stuck-at open defects shown in Figure 14, even though very few stuck-at close defects are inserted, the indexes of the ‘‘all’’ cases in Figure 19 are observed to increase significantly. In other words, even if the ratio is very small, stuck-at close defects can significantly affect the operation of the circuit. Hence, stuck-at open and close defects should be considered together during the diagnosis. Of course, due to the effects of stuck-at close defects, the diagnostic performance of stuck-at open defects may be degraded, and some defects may be omitted. However, most proportion of the crosspoints on the CMOL fabric become to the open state after a functional circuit configuration. Therefore, some missing stuck-at open defects may have no effect on the functional behavior of the circuit, or a slight yield drop may occur. Therefore, selection criteria for the final candidates should be adjusted considering the defect ratio and yield.

VI. CONCLUSION

In this paper, defect diagnosis methods for CMOL FPGA circuits were proposed. Accordingly, circuit configuration methods for a diagnosis independent of functional circuit configuration and diagnosis algorithms based thereon were proposed for an efficient defect diagnosis of CMOL FPGAs with a high defect ratio. Through the experiments, it was verified that CMOL fabrics of sizes up to 20 × 20 can be diagnosed. Furthermore, this method is applicable to CMOL fabrics of practical sizes with larger connectivity domains. For the diagnosis, the algorithms for stuck-at open defects and stuck-at close defects were proposed separately. The proposed defect diagnosis algorithms can be applied not only on stuck-at open and close defects separately but also on a mixture of the two types of defects. The diagnostic performance of the proposed algorithms was verified with CMOL fabrics having up to 20% stuck-at open defects and 0.02% or more stuck-at close defects.

REFERENCES

[1] *International Technology Roadmap for Semiconductors*, Semiconductor Industry Association, San Jose, CA, USA, 2004.

[2] S. Goldstein, A. Copen, and M. Budiu, "Nanofabrics: Spatial computing using molecular electronics," *ACM SIGARCH Comput. Archit. News* vol. 29, no. 2, pp. 178–191, 2001.

[3] M. R. Stan, P. D. Franzon, S. C. Goldstein, J. C. Lach, and M. M. Ziegler, "Molecular electronics: From devices and interconnect to circuits and architecture," *Proc. IEEE*, vol. 9, no. 11, pp. 1940–1957, Nov. 2003.

[4] A. Dehon, "Nanowire-based programmable architectures," *ACM J. Emerg. Technol. Comput. Syst.*, vol. 1, no. 2, pp. 109–162, Jul. 2005.

[5] K. K. Likharev and B. D. Strukov, "CMOL: Devices, circuits, and architectures," in *Introducing Molecular Electronics*. Berlin, Germany: Springer, 2006, 447-477.

[6] T. Wang, P. Narayanan, M. Leuchtenburg, and C. A. Moritz, "NASICs: A nanoscale fabric for nanoscale microprocessors," in *Proc. 2nd IEEE Int. Nanoelectron. Conf.*, 2008, pp. 1–5.

[7] *International Technology Roadmap for Semiconductors (ITRS)*, Semiconductor Industry Association, San Jose, CA, USA, 2009.

[8] H. Naeimi and A. DeHon, "A greedy algorithm for tolerating defective crosspoints in nanoPLA design," in *Proc. IEEE Int. Conf. Field-Program. Technol.*, Dec. 2004, pp. 49–56.

[9] W. Rao, A. Orailoglu, and R. Karri, "Logic level fault tolerance approaches targeting nanoelectronics PLAs," in *Proc. Des., Autom. Test Eur. Conf. Exhib.*, Apr. 2007, pp. 1–4.

[10] A. M. S. Shrestha, S. Tayu, and S. Ueno, "Orthogonal ray graphs and nano-PLA design," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2009, pp. 2930–2933.

[11] D. B. Strukov and K. K. Likharev, "CMOL FPGA: A reconfigurable architecture for hybrid digital circuits with two-terminal nanodevices," *Nanotechnology*, vol. 16, no. 6, p. 888, 2005.

[12] Z. Abid, M. Liu, and W. Wang, "3D integration of CMOL structures for FPGA applications," *IEEE Trans. Comput.*, vol. 60, no. 4, pp. 463–471, Apr. 2011.

[13] A. Madhavan, T. Sherwood, and D. B. Strukov, "High-throughput pattern matching with CMOL FPGA circuits: Case for logic-in-memory computing," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 26, no. 12, pp. 2759–2772, Dec. 2018.

[14] W. Rao, A. Orailoglu, and R. Karri, "Fault tolerant approaches to nanoelectronic programmable logic arrays," in *Proc. IEEE/IFIP Int. Conf. Dependable Syst. Netw. (DSN)*, Jun. 2007, pp. 216–224.

[15] I. Polian and W. Rao, "Selective hardening of NanoPLA circuits," in *Proc. IEEE Int. Symp. Defect Fault Tolerance VLSI Syst.*, Oct. 2008, pp. 1–8.

[16] W. N. N. Hung, C. Gao, X. Song, and D. Hammerstrom, "Defect-tolerant CMOL cell assignment via satisfiability," *IEEE Sensors J.*, vol. 8, no. 6, pp. 823–830, Jun. 2008.

[17] Z.-F. Chu, Y.-S. Xia, and L.-Y. Wang, "Cell mapping for nanohybrid circuit architecture using genetic algorithm," *J. Comput. Sci. Technol.*, vol. 27, no. 1, pp. 113–120, Jan. 2012.

[18] D. Chen, Y. Xia, and Z. Wang, "Stuck-at-close defect propagation and its blocking technique in CMOL cell mapping," *Microelectron. J.*, vol. 72, pp. 100–108, Feb. 2018.

[19] W. B. Culbertson, R. Amerson, R. J. Carter, P. Kuekes, and G. Snider, "Defect tolerance on the teramac custom computer," in *Proc. 5th Annu. IEEE Symp. Field-Program. Custom Comput. Mach.*, Dec. 1997, pp. 1–4.

[20] M. B. Tahoori and S. Mitra, "Fault detection and diagnosis techniques for molecular computing," in *Proc. Nanotech* vol. 3, 2004, pp. 57–60.

[21] Z. Wang and K. Chakrabarty, "Using built-in self-test and adaptive recovery for defect tolerance in molecular electronics-based nanofabrics," in *Proc. IEEE Int. Conf. Test*, 2005, pp. 1–7.

[22] J. Khakbaz and E. J. McCluskey, "Concurrent error detection and testing for large PLA's," *IEEE Trans. Electron Devices*, vol. 29, no. 4, pp. 756–764, Apr. 1982.

[23] S. J. Upadhyaya and K. K. Saluja, "A new approach to the design of built-in self-testing PLAs for high fault coverage," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 7, no. 1, pp. 60–67, Jan. 1988.

[24] B. D. Liu and J. J. Sheu, "A new low overhead design for testability of programmable logic arrays," in *Proc. IEEE Int. Symp. Circuits Syst.*, Dec. 1991, pp. 1972–1975.

[25] O. Tunali and M. Altun, "A survey of fault-tolerance algorithms for reconfigurable nano-crossbar arrays," *ACM Comput. Surv.*, vol. 50, no. 6, pp. 1–35, Jan. 2018.

[26] M. B. Tahoori, "Defects, yield, and design in sublithographic nanoelectronics," in *Proc. 20th IEEE Int. Symp. Defect Fault Tolerance VLSI Syst.*, 2005, pp. 2–11.

[27] S. M. Sait and A. M. Arafeh, "Reconfiguration-based defect-tolerant design automation for hybrid CMOS/Nanofabrics circuits using evolutionary and non-deterministic heuristics," *Arabian J. Sci. Eng.*, vol. 40, no. 9, pp. 2515–2529, Sep. 2015.

JIHYE KIM received the B.S. and M.S. degrees in electrical and electronic engineering from Yonsei University, Seoul, South Korea, in 2002 and 2004, respectively, where she is currently pursuing the Ph.D. degree with the Department of Electrical and Electronic Engineering. Since 2004, she has been a DFT Engineer with Samsung Electronics, Hwaseong, South Korea. Her current research interests include VLSI/SOC design and testing, design for testability, silicon failure analysis, fault tolerance, and nanotechnology.

HAYOUNG LEE (Graduate Student Member, IEEE) received the B.S. degree in electrical and electronic engineering from Yonsei University, Seoul, South Korea, in 2016, where he is currently pursuing the combined Ph.D. degree with the Department of Electrical and Electronics Engineering.

His current research interests include built-in self-repair, built-in self-testing, built-in redundancy analysis, redundancy analysis algorithms, reliability, and VLSI design.

SEOKJUN JANG received the B.S. degree in electrical and electronic engineering from Yonsei University, Seoul, South Korea, in 2018, where he is currently pursuing the combined M.S. degree with the Department of Electrical and Electronics Engineering. His current research interests include scan chain diagnosis, test pattern reordering, and design for testability.

SUNGHO KANG (Senior Member, IEEE) received the B.S. degree in control and instrumentation engineering from Seoul National University, Seoul, South Korea, in 1986, and the M.S. and Ph.D. degrees in electrical and computer engineering from The University of Texas at Austin, Austin, TX, USA, in 1988 and 1992, respectively.

He was a Research Scientist with the Schlumberger Laboratory for Computer Science, Schlumberger Inc., Austin, and a Senior Staff Engineer with Semiconductor Systems Design Technology, Motorola Inc., Austin. Since 1994, he has been a Professor with the Department of Electrical and Electronic Engineering, Yonsei University, Seoul. His current research interests include VLSI/SOC design and testing, design for testability, design for manufacturability, and fault tolerant computing.

• • •