DIAGNOSING PROGRAMMABLE INTERCONNECT SYSTEMS FOR FPGAs[1]

Fabrizio Lombardi David Ashen Xiaotao Chen
Deptment of Computer Science
Texas A&M University
College Station, TX 77843

Wei Kang Huang
Dept. of Electronic Engineering
Fudan University
Shanghai 200433, P.R. China

Abstract

This paper deals with the diagnosis of field programmable interconnect systems (FPIS) in which nets are connected through programmable switches arranged in grids. A hierarchical approach to diagnosis is proposed. The conditions by which such process yields full diagnosis and the characteristics of the programming sequence, are fully proved. For a FPIS consisting of a $k \times k$ grid array, the number of tests is given by $4 + 4kn^2$, while the number of programming steps is $4nk + 1$, where n is the dimension of a grid The application of this technique to commercially available FPIS in FPGAs, is discussed.

1 Introduction

The programmable nature of today's digital circuits such as FPGAs (field programmable gate arrays), PLDs (programmable logic devices) and FPICs (field programmable interconnect chips) has made possible the manufacturing of complex digital systems with a substantial reliance on sophisticated interconnect resources [8,9,10]. Diagnosis consists of fault detection and location [6]. A different scenario arises in the diagnosis of field programmable interconnect systems (FPIS); these systems can be either stand alone chips (such as the FPIC of [11]), or an integral part of a chip (such as the PIA of the PLD of [9] or the interconnect of a FPGA) [8]. The presence of switches in the interconnect structure however, prevents the application of a traditional diagnostic approach (such as those found in [1,2,3,4,5] as well as continuity tests for antifuse based technology [10]), because the feature of programmability can not be accounted due to the rather static analysis of the diagnostic process (as provided by these techniques [4,5]) and the lack of internal probing.

The objective of this paper is to analyze and propose a hierarchical approach for diagnosing FPIS with no aliasing and confounding. Diagnosis (detection and location) of multiple faults in the switches and nets of a FPIS is considered.

[1]This research supported by grants from the Texas ARP, ATP and the Ministry of Education and Science of Japan.

2 Review

In this section, a brief review of behavioral and structural testing approaches for interconnects as related to the proposed approach will be presented. There are three types of faults commonly associated with nets [2]: stuck-at faults, open faults and bridge (short) faults. These faults can be tested by using either a *behavioral testing* or a *structural testing* strategy. The *Counting Sequence Algorithm* of [3] can be used to detect all bridge faults. This is accomplished using the so-called *Sequential Test Vectors (STV).* If both the STVs consisting of all 0 and all 1 are included, then a *Modified Counting Sequence* [5] is obtained. These test vectors The *Sequential Response Vector (SRV)* of a net to a STV is then used to detect and/or diagnosis shorts between nets. If the SRV of a net differs from its STV, then this vector is referred to as a *fault syndrome.* If a syndrome in the presence of a fault is the same as the fault-free response of a net, then it is impossible to determine whether or not this net is also part of the short. The response in this case is referred to as an *aliasing syndrome.* Similarly, the bridge fault between a pair of nets may produce the same syndrome as between another another net pair; it is impossible to determine whether or not there is a short between which pair of nets. The response is called a *confounding syndrome.* The *walking-1* test set proposed by [2] can avoid the aliasing and confounding problems,

All of the above approaches however, can be solely used for diagnosing interconnects in which no programmable device is present. A diagnosis method which partially addresses the issue of programmable chips, has been presented in [7]. This paper introduces a diagnosis method for EEPLAs (electrically erasable programming logic arrays). However, the approach of [7] is not applicable to the diagnosis of a FPIS as programmable devices in a FPIS have different and more complex programming modes.

3 Preliminaries

As outlined previously, a *homogeneous* FPIS is assumed; in this FPIS, the bidirectional nets run horizontally and vertically through a two-dimensional array of equally like programmable grids. The following definitions are applicable to a programmable grid: **(1)** *Switch:* a generic name for a programmable device to connect nets by switching [8,9]. The status of a switch is either closed (connected, on) or open (disconnected, off). **(2)** *Input Endpoint:* the origin point of a net. **(3)** *Output Endpoint:* the final (or destination) point of a net.

As in previous papers [4,7], the programmable grid is modeled as follows: it is assumed that the input endpoints are placed at a side (or port) of the programmable grid (initially in the horizontal direction and denoted by the input net set $I = I_j, j = 1, ..., n$ as shown in Figure (1)) and the output endpoints are located at a different port (initially in the vertical direction and denoted by the output net set $O = O_j, j = 1, ..., p$. Note that in this paper only the case of square grid is considered (i.e. $n = p$) as occurring in practice [8,11]. For a switch, the same model as in [8] is assumed (as shown in Figure (1)); a switch in the programmable grid is identified by a pair of coordinates (x, y) which corresponds to the position (x) of the input endpoint and the position (y) of the output endpoint to be connected. This switch is denoted as $S(x, y)$. A generalization is used for the placement of the switches on the grid, i.e. it is assumed that a switch is placed at every intersection of horizontal and vertical nets. Hence, the total number of switches is given by $n \times p$.

The programming process assumed in this paper consists of a reversible process in the switches. Prior to programming, all switches are assumed to be disconnected, i.e. off. The programming process consists of selectively turning on and off the switches by specifying their coordinates; the number of times that the grid is programmed, is referred to as the *number of programming steps.*

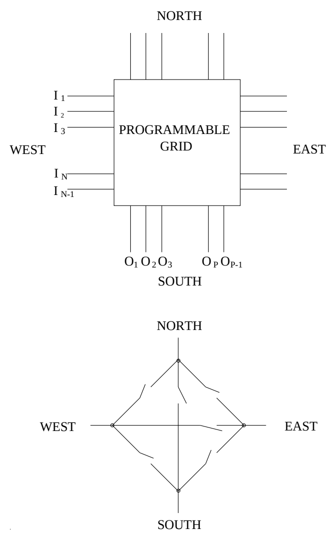


Figure 1: Programmable Grid and Switch Models

The grid is specified by four *ports* placed on the directions of the nets: East (E), West (W), South (S) and North (N). A programmed switch accomplishes a *one-to-one* connection between a single pair of endpoints (one input and one output) located on two different ports. The modes of a switch are as follows: NS, EW, NE, NW, ES, WS. If *all* switches in a grid are in a mode, then the grid is said to be in that mode, i.e. the mode of a grid is denoted by the pair of ports to which all nets are connected. The *status* of a grid (as for the programming of the switches) can be distinguished as follows: *(a) Uncommitted:* the grid is in the NS and EW modes (Figure (2)). *(b) Committed:* the grid is in either the SW, or SE or NE or NW mode. *(c) Unprogrammed:* the grid is off (Figure (3)). Equivalently, it is assumed that a switch can have the following *modes:* *(1)* OFF: disconnected mode. *(2)* ON: connected mode denoted by the ports of the input and output endpoints, i.e. either EW or NS or SW or SE or NE or NW. Note that using the model of [8], the physical nets for the EW and NS modes in a switch must be located on two different planes in the layout; the nets for the SW, SE, NE and NW modes are on the same plane.

In the analysis, diagnosis is based on a fault model made of two parts: *(A)* The fault model for the nets in the interconnect; *(B)* The fault model for the switches and their programming process.

For the nets, a *strictly physical* characterization consisting of stuck-at and bridge faults, is used in the fault model. The fault model for a switch is based on a structural model; this model utilizes a *functional* characterization in which faults are defined as follows: **(1)** *Programming fault:* a fault causes the net to be erroneous such that the wrong (input or output) end-

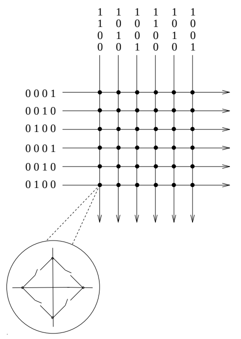


Figure 2: Uncommitted Grid

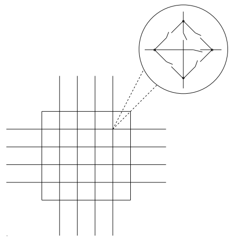


Figure 3: Unprogrammed Grid

point is connected. In practice, this means that either a switch $S(k,h)$ is programmed instead of $S(i,j)$ ($S(k,h)$ $S(i,j)$ is referred to as the extra (missing) switch), or for a given mode the switch is not programmed. *(2) Stuck:* within a mode a switch is either stuck-on, or stuck-off.

Hereafter, the following assumptions are valid in the analysis. (1) The OR (permanent) bridge is assumed for simplicity in the nets. (2) There is no restriction on the number of faulty nets and switches of the interconnect.

4 Uncommitted Grid Diagnosis

In this section, the diagnosis of an uncommitted grid (i.e. a grid with all switches in the NS and EW modes) is considered. An uncommitted grid can be modeled as made of two disjoint sets of parallel nets: one set of parallel nets runs in the horizontal direction, while the second set of parallel nets runs in the vertical direction (Figure (2)). In absence of faults, these sets must be disjoint, because all the switches are assumed to be in the EW and NS modes. Physically, this corresponds to an implementation in which the nets for the EW and NS modes in each switch are located on different planes, i.e. physically, these sets of nets are located on different planes too. This section partially utilizes the following fault model: **(1)** *Adjacency assumption:* any bridge (short) fault may happen between two nets only if these two nets are physically adjacent in the same plane. **(2)** *Continuous assumption:* given two non-adjacent nets n_i, n_j and a subset of nets (denoted by B) between n_i and n_j on a plane, if ni and n_j are shorted, then all nets in B between them are also shorted together. **(3)** *Planar assumption:* a short between nets on two different planes occurs strictly perpendicular in the vertical direction.

The proposed method for testing an uncommitted grid is based on the following two steps: *(1)* Find the local adjacent nets in the nets under test on every plane and generate the local test (diagnostic) vectors for each plane. *(2)* Generate the test vectors for testing the absence of any adjacencies between planes, where the *local adjacent nets under test (LANUT)* are the nets on a plane of the uncommitted grid as well as connecting the other grids. The *local test vectors* can test the faults which occur within the LANUT and diagnose them.

Observation 1: Adjacencies of nets (on a plane) can be tested by using only three vectors under the adjacency and the continuous assumptions.

Each set of parallel nets can be tested under the adjacency, continuous and planar assumptions; this corresponds to the diagnosis of the very popular planar block layout for FPIS as well as MCM systems, commonly referred to as the *bus structure.* In absence of a fault, there is no crossing between adjacent nets. For testing a bus structure, three STVs are defined by the following vectors, thus yielding the next observation: $STV_1 = 1001001...,; STV_2 = 0100100...,: STV_3 = 0010010....$

Observation 2: Three STVs are sufficient to diagnose any bridge faults (as well as stuck-at and open faults) among an arbitrary width of regular adjacent nets in a bus structure under the continuous and adjacency assumptions, provided the widest bridge fault is within five consecutive nets.

Observation 3. A fault in each switch $S(x,y)$ of an uncommitted grid can be detected by using a further test for each plane: $STV_4^h = 000000...,: STV_4^v = \overline{STV_4^h}$

where the upperscript identifies the set of nets to which the test is applied, i.e. either the horizontal (h) or vertical (v) set.

Let $T = STV_i, i = 1, ..., 4$ and the grid be uncommitted. The following statements are derived from the previous observations. *(1)* An uncommitted grid is fully tested by T. *(2)* No aliasing and/or confounding exist using T for testing all possible (vertical and horizontal) adjacencies in the nets through an uncommitted grid. *(3)* Under the adjacency, continuous and planar assumptions, T is an optimal test set for $n \geq 4$.

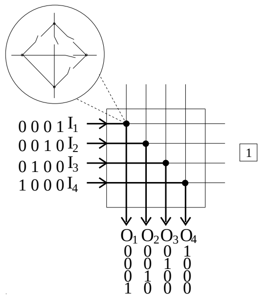


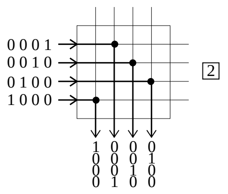
Figure 4: First Programming Step (Mode WS)

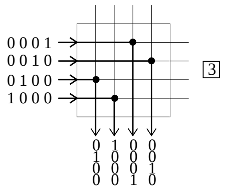
5 Programmable Grid Diagnosis

The cases in which a grid and its switches are all programmed in a mode other than EW and NS, require a different model. This is related to the function of an on-switch: an on-switch $S(i,j)$ connects I_i to O_j, while at the same time it disables the horizontal (vertical) net to proceed along the row (column) of the grid as specified by the mode.

An example will be given to show the basic principles of the proposed approach for a programmable grid in which all nets connect the input endpoints (located West) to the output endpoint (located South), i.e. for the mode WS. Consider the programmable square grid shown in Figure (3), i.e. $n = 4 = p$. The proposed sequence in the programming of the switches is performed according to a toroidal diagonal strategy. In this method, the set of switches along a diagonal is programmed (turned on) and tested using multiple test vectors for a walking-1 test set. Initially, all the switches along the major diagonal are programmed in the WS mode, i.e. $S(1,1), S(2,2), S(3,3)$ and $S(4,4)$ are turned on, while keeping all $S(i,j)$ (for $i \neq j$) in the off mode (Figure (4)). This process is continued by programming and testing next the switches located one position to the right of the main diagonal. This means that the diagonal is effectively shifted with wrap-around in the horizontal direction by one position to the right (as shown in Figure (5)). The programming steps and the grid configurations for all steps are shown in Figure (5) too.

For each configuration of the grid in a programming step, n tests are applied to the input endpoints of the programmed grid. These vectors make up a walking-1 set test (denoted as T_s): if there is no fault, then there is only one output net with a value of one, i.e. for this set, the j output endpoint (at net O_j) is equal to 1 (all other endpoints at nets $O_k \neq O_j$ are 0) provided the on-switch is $S(i,j)$ and the only input endpoint with a 1 is I_i (all $I_h = 0$ for $h \neq i$). The test set as well as the





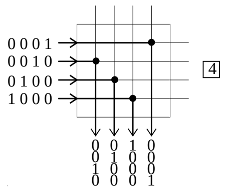


Figure 5: Programming Steps (continued) (Mode WS)

values of the output endpoints for each programming step are also given in Figures (4) and (5).

For a square grid (i.e. $n = p$), then the programming procedure is given as follows (note that $mod_n(m)$ denotes the modulo operation from 1 to n on m).

Algorithm 1.

```
DO for $h = 1$ to $max$
    DO for $k = 1$ to $max$
    $i = mod_{max}(k + h - 1)$
    $j = k$
    Program the switch $S(i,j)$ as ON
    END DO
    Apply $T_s$ and compare the values at the output endpoints with the expected values.
    If they are different, at least a fault has occurred.
END DO
```

where $max = n$. The number of programming steps is $n = p$; the number of tests is given by the number of applications of T_s; as T_s is applied p times and $|T_s| = n$, then the total number of tests is $n \times p = n^2$. Note that Algorithm (1) is only applicable for a grid in the WS mode; however, similar algorithms can be deviced for the other three remaining modes, i.e. ES, NE and NW.

An analysis of the diagnostic process for the programmable grid in the WS mode follows. Let $ptv(i)$ denote the input test vector (PTV) made of all 0's except for a 1 at I_i in the walking-1 test set, i.e $T_s = ptv(j) forj = 1, ..., n$. Grid diagnosis with no aliasing and confounding is possible due to the following characteristics. *(1)* If the test vector $ptv(i)$ is provided at the inputs of the grid and $S(i,j)$ is the only on-switch on I_i, then in the absence of faults $O_j = 1$

and $O_h = 0$ (for all $h \neq j$). *(2)* Every switch $S(i,j)$ is programmed in the on-state only once in Algorithm (1); in absence of faults, I_i is connected to O_j through $S(i,j)$ only for the tests in a single application of T_s. *(3)* As shorts between nets have already been diagnosed using the test set T of Section (4), then at least a switch fault must have occurred provided the values at the output port are different from the values at the input port.

Assume that $S(i,j)$ is faulty; the following cases are possible. *(1)* Stuck-on: then $O_j = 1$ for p times during the execution of Algorithm (1) (once per application of $ptv(i)$ as T_s is applied p times). *(2)* Stuck-off: then $O_j = 0$ for every $ptv(i)$. *(3)* Programming fault: this corresponds to $S(i,j)$ (extra switch) being turn on in place of another switch $S(h,k)$ (missing switch) for $h \neq i$ and $k \neq j$; this is detected by $ptv(i)$ ($O_j = 1$ and $O_k = 0$) and $ptv(h)$ ($O_j = 0$ and $O_k = 0$). All other cases of indices for missing/extra switches are detected too.

6 FPIS with Interconnected Programmable Grids

This Section extends the results of the previous sections to multiple grids connected together through busses. This FPIS is shown in Figure (6): each square is a $n \times n$ programmable grid and in the FPIS, there are $k \times k$ equally-like grids. Initially, the simplest configuration shown in Figure (7) must be considered; the output nets of Grid 1 are the input nets of Grid 2, i.e. Grid 1 is in the WS mode, while Grid 2 is in the NE mode. The scenario of interconnected grids inherently assumes that faulty switches in the grids are independent except for the nets extending from one grid to another. Consider Figure (7); the grids (G_1 and G_2) are two square grids. The programming steps are shown in Figures (7) and (8): this corresponds to the same technique as in Section (5) for both grids such that the I_k of G_1 is the same as the O_k out of G_2 in the absence of faults. This grid pair is denoted by [G_1,G_2]. So to test [G_1,G_2], the same testing requirements as for a single grid are applicable, i.e. n^2 tests and n programming steps.

The above process for testing the grid pair of Figure (7) can be considered as the basic step for testing the whole FPIS; this process is based on the following two rules in the programming sequence: *Rule 1:* every switch is programmed to be simultaneously in the NS and EW modes; *Rule 2:* all other modes are programmed individually such that all switches in the grid are in the same mode. Therefore, this hierarchical process consists of two phases. *Phase 1:* test the

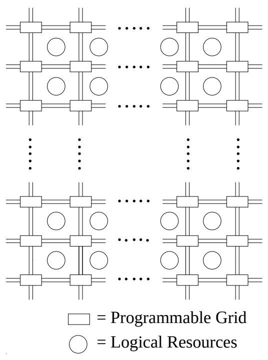


Figure 6: FPIS Model

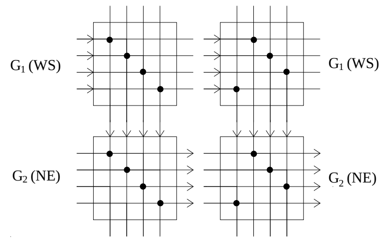


Figure 7: Grid Pair

uncommitted grids for the EW and NS modes. *Phase 2:* test the programmable grids in the FPIS for the NE, NW and ES and EW modes.

The hierarchy abides to testing the uncommitted/committed status of the grids within the restriction that controllability and observability of the test vectors must be achieved in the whole FPIS. Phase (1) utilizes the test set of Section (4); this phase diagnoses the uncommitted status of all grids and the nets in the horizontal and vertical directions (which connect the grids) as bus structures. Phase (2) is executed after diagnosing the faults found in Phase (1). In the second phase, a more comprehensive analysis is required. In the FPIS, each programmable grid (denoted as $G_{h,l}$, for $h = 1,..,k$ and $l = 1,...,k$) is considered as the basic programming unit in a manner similar to the switch in the diagnosis approach of Section (5). The FPIS is tested in a pair-wise grid fashion (an example is shown in Figure (9)) along two diagonals in two further subphases; for example, along the main diagonal and the adjacent diagonal this process is given as follows (in the analysis for each pair, Grid 1 is in the WS mode, while Grid 2 is in the NE mode).

Subphase 2a: Test each grid pair $[G_h,h,G_{h+1,h}]$ for $h = 1,3,5....$ The input endpoints are on the horizontal nets into $G_{h,h}$ at port W, while the output endpoints are on the horizontal nets out of $G_{h+1,h}$ at port E. Note that every grid $G_{h,\alpha}$ (for $\alpha < h$) and $G_{h+1,\beta}$ (for $\beta > h$) are uncommitted in the EW mode.

Subphase 2b: Repeat Subphase 2a for the grids with

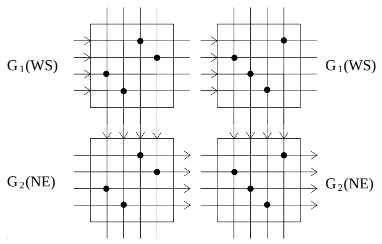


Figure 8: Grid Pair (continued)

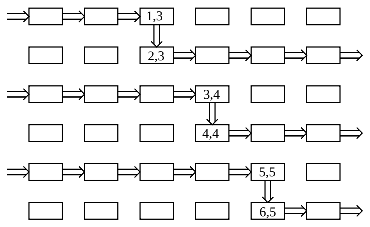


Figure 9: Example

values of $h = 2, 4, 6 ....$

The above subphases must be executed for all grids along every diagonal of the FPIS using the same testing technique as in Procedure (A). However, a grid can be used to connect nets along all four ports, i.e for the four modes WS, ES, NW and NE. This means that Phase (2) as described above must be executed for each mode (i.e. ES, WS, NW and NE). Figure (10) shows the four grid pairs used in testing a grid for the four modes ES, WS, NE and NW in the committed status.

The complexity of the proposed hierarchical diagnosis approach for FPIS can be calculated as follows. Each subphase requires n^2 tests and n programming steps for each mode; as there are k diagonals in the array, then the FPIS can be tested in each execution of Phase (2) using $2 \times k \times n^2$ tests and $k \times n$ programming steps for the grids located in two adjacent diagonals. Phase (2) must be executed four times for the committed status. Section (4) has proved that Phase (1) requires 4 tests; so the FPIS of Figure (6) can be tested in $4 + 4kn^2$ tests and the number of programming steps is $4kn + 1$. Note that the approach of [7] within the same assumptions and fault models as described in previous sections, requires $5 \times n^2k^2$ tests and $n^2 \times k^2$ programming steps.

7 Application : Diagnosis of FPIS in FPGAs

This section presents the application of the proposed approach to the diagnosis of a FPIS as part of a Field Programmable Gate Array chip [8]. The assumed

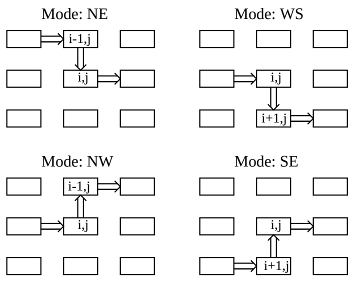


Figure 10: Phase 2 for Committed Status of Grid (i,j)

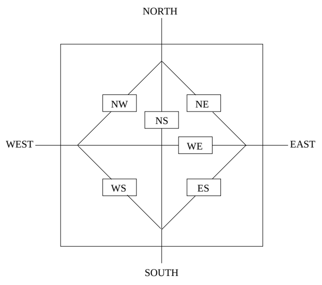


Figure 11: Switch Matrix Model

FPIS configuration consists of the Logic Cell Array (LCA) general purpose interconnect, as found in the 3000 FPGA family by Xilinx [8]. As shown in Figure (6), this FPIS consists of an array of $(g + 1) \times (g + 1)$ programmable grids (referred to as switch matrices) to connect $g \times g$ logic blocks (which are assumed to be not connected to the nets).

In [8], a switch matrix can connect five nets per port according to 20 different options for a total of 54 switching possibilities. The proposed model for the switch matrix of the XC3000 family is consistent with the characterization of a programmable switch used by Xilinx [8]: a switch matrix consists of 6 programmable grids corresponding to the modes (NW,WS,NS,SE,WE,NE) described in Section (5). It is then possible to define the six grids, each grid associated with a mode; they are shown in Figure (11). Therefore, a switch matrix is modeled functionally as no detailed information is available in [8].

Observation 4. The grids for the EW, NS, WS and NE modes have a major diagonal with all switches such that input and output endpoints follow the same order as described for the general FPIS model of Section (6).

Observation 5. The modes NS and EW can allow the order of the nets to be changed along either the vertical or horizontal direction.

Observation 6. For diagonals in each grid with only $w(w < 5)$ switches, diagnosis by a walking-1 test set still requires a T_s such that $|T_s| = w$.

Assume that nets are numbered at the ports of a grid as in [8], i.e. nets 1-5 at port N, nets 6-10 at port

E, 11-15 at port S and 16-20 at port W (in a clockwise fashion). Let $d_{i,j}(\beta,\gamma,\alpha)$ denote the diagonal starting at the switch which can connect net i of port β and net j of port γ of a grid $\beta\gamma$ in a matrix $M_\alpha(\alpha = 1, 2)$, i.e. in mode $\beta\gamma$. Diagnosis of the general purpose interconnect for the XC3000 family requires a slight modification to the procedure of Section (6). As for Observation (4), Phase (1) is now given as follows (for two grids in the switch matrices M_1 and M_2 in a pair as equivalent to the two grids of Figure (7)):

1a. Test $d_{5,11}(N,S,1)$ and $d_{5,11}(N,S,2)$. Test $d_{6,20}(E,W,1)$ and $d_{6,20}(E,W,2)$.

1b. Test $d_{7,20}(E,W,1)$ and $d_{7,20}(E,W,2)$. Test $d_{4,11}(N,S,1)$ and $d_{4,11}(N,S,2)$.

1c. Test $d_{5,12}(N,S,1)$ and $d_{5,12}(N,S,2)$. Test $d_{6,19}(E,W,1)$ and $d_{6,19}(E,W,2)$.

The number of tests for Phase (1) is given as follows: (1a) 5 tests, (1b) 4 tests, (1c) 4 tests.

To test the four remaining modes of M_2, Phase (2) is given by 4 substeps as follows:

2a. M_1 is in mode WS and M_2 is in mode NE. The tests are as follows:

(2aa) Test $d_{15,20}(W,S,1)$ and $d_{5,10}(N,E,2)$; (2ab) Test $d_{15,20}(W,S,1)$ and $d_{4,8}(N,E,2)$; (2ac) Test $d_{15,19}(W,S,1)$ and $d_{5,10}(N,E,2)$. The number of tests is 5+2+2=9.

2b. M_2 is in mode WS and M_3 is in mode NE. The tests are the same as in (2a) by exchanging M_2 with M_3 and M_1 with M_2. The same number of tests as in Substeps (2a) is applicable.

2c. M_1 is in mode ES and M_2 is in mode NW. The tests are as follows:

(2ca) Test $d_{6,14}(E,S,1)$ and $d_{2,20}(N,W,2)$. (2cb) Test $d_{7,15}(E,S,1)$ and $d_{1,19}(N,W,2)$. (2cc) Test $d_{7,14}(E,S,1)$ and $d_{2,20}(N,W,2)$. (2cd) Test $d_{7,14}(E,S,1)$ and $d_{1,20}(N,W,2)$. (2ce) Test $d_{6,14}(E,S,1)$ and $d_{1,20}(N,W.2)$. (2cf) Test $d_{7,14}(E,S,1)$ and $d_{1,20}(N,W,2)$. The number of tests is 2+2+3+3+2+3=15.

2d M_2 in mode ES and M_3 is in mode NW. The tests are the same as in (2c) by exchanging M_2 with M_3 and M_1 with M_2. The same number of tests as in Substep (2c) is applicable.

The number of tests for Phase (2) is therefore 9+9+15+15=48. Hence, to test the general purpose interconnect of a FPGA in the XC3000 family (arranged as $g \times g$ logic blocks), the number of tests is given by $T_t = 13 + 48 \times (g + 1)$; for example, if g is 8 (as for the XC3020), then $T_t = 445$. The number of programming steps T_p can be calculated as follows: for a switch matrix, every time a diagonal in each grid is tested, then this corresponds to a programming step; so Phase (1) requires 3 programming steps, while Phase (2) requires 18 programming steps.

Table 1:

Diagnosis	XC3020			XC3042			XC3195		
Approach	T_t	g	T_p	T_t	g	T_p	T_t	g	T_p
[7]	12150	8	2025	25350	12	4225	79350	22	13225
Proposed	445	8	189	637	12	273	1117	22	483

The general purpose interconnect therefore requires $T_p = 21 \times (g + 1)$ programming steps.

Table (1) compares T_t and T_p using the proposed method with the number of tests and programming steps required using the test set of [7] for different FPGAs of the XC3000 family. The proposed approach achieves a considerable reduction in the number of tests as well as programming steps, this reduction is more pronounced for the large FPGAs in this family.

References

[1] Lien, J.C. and M. A. Breuer, "Maximal Diagnosis for Wiring Networks," *Proc. IEEE ITC,* pp. 96-105, 1991.

[2] Hassan, A., J. Rajski and V.K. Agrawal, "Testing and Diagnosis of Interconnects using Boundary-scan," *International Test Conference,* pp.126-137, 1985.

[3] Kautz, W. H., "Testing for Fault in Wiring Networks," *IEEE Transactions on Computers,* Vol. C-23, No. 4, pp. 358 - 363, 1974.

[4] Liu, T., Lombardi, F. and J. Salinas, "Diagnosis of Interconnects and FPICs using a Structured Walking-1 Approach," *Proc. IEEE VLSI Test Symposium,* pp. 256-261, 1995.

[5] Wagner, P.T., "Interconnect Testing with Boundary Scan," *International Test Conference,* pp.52-57, 1987.

[6] Liu, T., Huang, W.K. and F. Lombardi, "Testing of Uncustomized Segmented Channel FPGAs," *Proc. ACM Int. Symp. on FPGAs,* pp. 125-131, 1995.

[7] Rajsuman, R., "A New Testing Method for EEPLA," *IEEE Trans. on CAD of ICAS,* Vol. CAD 13, No. 7, pp. 935-939, 1994.

[8] Xilinx, The Programmable Logic Data Book, San Jose, 1994.

[9] Altera, Data Book, San Jose, 1993.

[10] Actel, FPGA Data Book and Design Guide, Sunnyvale, 1994.

[11] Aptix Corporation, System Data Book, San Jose, 1993.