Efficient Construction of Catastrophic Patterns for VLSI Reconfigurable Arrays with Bidirectional Links

Amiya Nayak†, Linda Pagli‡, Nicola Santoro†

† Center for Parallel & Distributed Computing, School of Computer Science
Carleton University
Ottawa, K1S 5B6, Canada

‡ Dipartimento di Scienze dell'Informazione
University of Pisa
Corso Italia 40, 56100 Pisa, Italy.

Abstract

Patterns of faults that are catastrophic for regular architectures, particularly the systolic arrays, have been studied. For a given link configuration, there are many fault patterns which are catastrophic. Among those, there is a particular fault pattern, called the *reference fault pattern*, which is crucial for the development of testing techniques; furthermore, the efficiency of any testing algorithm can be further improved in the presence of efficient algorithms for constructing the reference fault pattern.

In this paper, we develop a new algorithm for the construction of the reference fault pattern for VLSI reconfigurable arrays in which the links are bidirectional. The complexity of the new algorithm is $O(kN)$ which is a significant improvement over the existing $O(N^2)$ algorithm, where k is the number of bypass links, and N is the length of the largest bypass link.

1 Introduction

Fault tolerance by means of component redundancy and mechanisms for reconfiguration is common in VLSI-based regular architectures. The redundant processing elements (PEs) are used to replace any faulty PE(s); the redundant links are used to bypass the faulty PEs and reach the redundant PEs used as a replacement. In the literature, many algorithms [2-4,9-11] have been proposed which take into account the built-in redundancy and reconfigure the system in the presence of faulty PEs and faulty links. The reconfiguration approaches work quite well and achieve good yield, but are prone to failures when confronted with specific patterns of faults in the system. In fact, faults occuring at strategic locations may have catastrophic effect on the entire structure and cannot be overcome by any clever reconfiguration process.

In a linear array of PEs, with no link redundancy, a single PE fault in any location is sufficient to block the flow of information from input to output. Similarly, it is easy to see that the same array with $k-1$ bypass links $\{g_2, g_3, \ldots, g_k\}$ cannot tolerate g_k (not k, the total number of links) PE faults if they occur in a block (or cluster). The probability of block faults of size g_k or higher is relatively small; however, there exist many patterns of g_k faults, not in a block, for which any reconfiguration algorithm will fail, and the structure will still be unusable.

The fault patterns that are catastrophic have been studied for systolic arrays [6-8]. The knowledge about the catastrophic fault patterns can be used in many ways to improve reliability of regular systems. The knowledge about the catastrophic fault patterns can be applied to test for the likelihood of a catastrophe in regular systems. It is also possible to evaluate a design, using the characterization of catastrophic fault patterns, to verify if specific patterns of faults are catastrophic; should this be the case, any future design can be upgraded by incorporating appropriate redundancy structure into the design to minimize catastrophe.

For a given link configuration, there are many fault patterns which are catastrophic. Among those, there is a particular fault pattern, called the *reference fault* *pattern*, which has special properties. These properties have been used in [5] in 1) the identification of necessary and sufficient conditions for an arbitrary fault pattern to be catastrophic and 2) the development of efficient testing technique. The reference fault pattern is an essential part of the testing algorithm. The efficiency of any testing algorithm can be further improved in the presence of efficient algorithms for constructing reference fault pattern.

Algorithm for constructing the reference fault pattern in the case of bidirectional links is given in [8]. The algorithm has complexity $O(N^2)$, where $N = g_k$ is the length of the largest bypass link. In this paper, we develop an improved algorithm for the construction of reference fault pattern. The new algorithm has time complexity $O(kN)$ where k is the number of bypass links in the VLSI reconfigurable array.

The organization of the paper is as follows. Basic concepts and terminologies are introduced in Section 2. The improved algorithm is described in Section 3 followed by a conclusion in Section 4.

2 Preliminaries

Let $A = \{p_0, p_2, \ldots, p_N\}$ denote a one-dimensional array of PEs, where each $p \in A$ represents a processing element and there exists a direct link (*regular link*) between p_i and $p_{i+1}, 0 \leq i < N$. Any link connecting p_i and p_j where $j > i + 1$ is said to be a *bypass link*. The length of a bypass link, connecting p_i and p_j, is the distance in the array between p_i and p_j; i.e., $|j - i|$. Regular links exist between neighbouring PEs while the bypass links are assumed to exist between non-neighbours. The bypass links are used strictly for reconfiguration purposes when a fault is detected. For all other purpose, the bypass links are considered to be the redundant links. The links can be either unidirectional or bidirectional, but the focus will be only on bidirectional links in this paper.

Given an integer $g \in [1, N]$ and an array A of size N, A is said to have *link redundancy g*, if for every $p_i \in A$ with $i \leq N - g$ there exists a link between p_i and p_{i+g}; if $g > 1$, such a link will be called a bypass link. Let $G = \{g_1, g_2, \ldots, g_k\}$, where $g_j < g_{j+1}$ and $g_j \in [1, N]$, be the set of all links for A. The array A is said to have link redundancy G if A has link redundancy $g_1, g_2, \ldots, g_k$. In the following, it will be assumed that no other links exist in the array except the ones specified by G. Thus, G totally defines the *link structure* of A, and A will be called a *k−redundant system*.

Given a linear array A of size N, a *fault pattern* for A is a set of integers $F = \{f_0, f_1, \ldots, f_m\}$ where $m \leq N, f_j < f_{j+1}$ and $f_j \in [0, N]$. An assignment of a fault pattern F to A means that for every $f \in F$, p_f is faulty.

Definition 1 *A fault pattern F is catastrophic for an array A with link redundancy G if the array cannot be reconfigured in the presence of such an assignment of faults.*

In other words, F is a cut-set of the graph corresponding to A; that is, the removal of the faulty elements and their incident links will cause the array to become disconnected.

Definition 2 *The width W_F of a fault pattern F is the number of PEs between and including the first and the last fault in F. That is, if $F = \{f_0, \ldots, f_m\}$ then $W_F = f_m - f_0 + 1$.*

A characterization of catastrophic fault patterns was given in [8]. It was shown that a catastrophic fault pattern for a link configuration $G = \{g_1, g_2, \ldots, g_k\}$ must have at least g_k number of faults. Also, the width of a fault pattern must be bounded for the pattern to be catastrophic. Bounds were established on the width of the fault window W_F for different link configurations. In this paper, we will consider *minimal* catastrophic fault patterns; that is, a fault pattern which has exactly g_k faulty PEs.

Let $F = \{f_0, f_1, \ldots, f_{g_k - 1}\}$ an arbitrary fault pattern, consisting of g_k faults for an arbitrary link configuration $G = \{g_1, g_2, \ldots, g_k\}$. Without loss of generality, assume that $f_0 = 0$. We represent F by a Boolean matrix W of size $(W_F^+ \times g_k)$, where $W_F^+ = \lceil W_F / g_k \rceil$, defined as follows:

$$W[i,j] = \begin{cases} 1 & \text{if } (ig_k + j) \in F \\ 0 & \text{otherwise} \end{cases}$$

In the matrix representation, each $f_l \in F$ is mapped into $W[x_l, y_l]$ where $x_l = \lfloor f_l / g_k \rfloor$ and $y_l = f_l \bmod g_k$. Notice that $W[0,0] = 1$ which indicates the location of the first fault.

Example 1: Consider a fault pattern F_1 with 8 faults for an array of PEs with bidirectional links with link configuration $G = \{1, 4, 8\}$ which has $W_F = 19$ as shown in Figure 1. The Boolean matrix representation of F_1 is shown Figure 2.

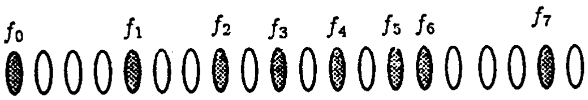

Figure 1 : A fault pattern F_1 for $G = \{1, 4, 8\}$

$$\begin{bmatrix} \overset{f_0}{1} & 0 & 0 & 0 & \overset{f_1}{1} & 0 & 0 & \overset{f_2}{1} \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & \underset{f_7}{1} & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Figure 2 : The matrix representation for F_1

Notice that any minimal catastrophic fault pattern satisfies the necessary condition that $\forall j$, there is only one i for which $W[i, j] = 1$.

Let W be the matrix representation of a minimal fault pattern F. The *row coordinates* of F is the ordered set $\{x_0, x_1, \ldots, x_{g_k-1}\}$ of the row indices of W corresponding to the faults f_i $(0 \leq i \leq g_k - 1)$.

Let $W[x_l, y_l]$ be the location of fault f_l. The location $W[i, y_l]$, with respect to f_l, is *interior* if $i < x_l$, *border* if $i = x_l$, and *exterior* if $i > x_l$. Similarly, for a given fault pattern F, $I(F)$ (i.e., *interior of* F) is the set of all interior elements, $B(F)$ (i.e., *border of* F) is the set of all border elements, and $E(F)$ (i.e., *exterior of* F) is the set of all exterior elements.

Now with respect to the matrix representation of F, a fault pattern F is said to be *catastrophic* for an array A with link redundancy G if it is not possible to reach any exterior element from any interior element using the links in G.

Definition 3 *The area A_F of a fault pattern F is the number of interior and border elements; that is,*

$$A_F = \|I(F) \cup B(F)\| = \sum_{j=0}^{g_k-1} (x_j - 1).$$

For a given link configuration, there are many fault patterns which are catastrophic. Among those, there is a particular fault pattern, called the *reference fault pattern*.

Definition 4 *Given a link configuration G, a reference fault pattern (RFP) is a catastrophic fault pattern for G which has largest width W_F and maximum area.*

3 Construction of Reference Fault Pattern

The algorithm for the construction of the reference fault pattern has been given in [8] for bidirectional links. The algorithm has complexity $O(W_F + kg_k)$. Since W_F can be as large as $O(g_k^2)$ in the worst case, the complexity of the algorithm can be $O(g_k^2)$. In the following, we present improved algorithm for constructing the reference fault pattern $\mathcal{F}$ in the case of bidirectional links.

The new algorithm practically constructs the reference fault pattern by determining the position of the border elements in the corresponding matrix representation. This is achieved without actually constructing the Boolean matrix unlike the case in the existing algorithm; note that to construct the matrix, $O(g_k^2)$ operations would be required.

The placement of a fault in a specific position in the array may "force" the placement of other faults. The forced placement of other faults by a fault f will be called the bidirectional effect of the fault f.

Definition 5 *Let $f \in F$ and let $W[i, j]$ be the corresponding entry in the Boolean representation of F. The (bidirectional) effect of f is the set $\psi(f) = \{f + g_k \pm g \ : \ g \in G\}$.*

Theorem 1 *F is catastrophic for G if and only if $\psi(f) \subseteq E(F) \cup B(F)$ for every $f \in F$.*

Proof: (if part) By contradiction, let $f \in F$ be such that $\psi(f) \cap I(F) \neq \emptyset$. Let $W[i, j]$ be the border element in W corresponding to f. Also let $x = f + g_k - g$ be an arbitrary element in $\psi(f) \cap I(F)$ for some $g \in G$. Then $x + g = (f + g_k - g) + g = f + g_k$; the corresponding element is $W[i + 1, j]$ which is exterior since $W[i, j]$ is the border element corresponding to f. Hence, it is possible to reach the exterior element $x + g$ from the interior element $x \in \psi(f) \cap I(F)$, contradicting the fact that F is catastrophic.

(only if part) Let $\psi(f) \subseteq E(F) \cup B(F)$ for every $f \in F$. To prove that F is catastrophic, it suffices to show that $\forall g \in G$ and $\forall x \in I(F)$, $x + g \in I(F) \cup B(F)$. Consider an arbitrary column in i in W, let x_i be the row index such that $W[x_i, i] = 1$ and let $f \in F$ be the corresponding fault mapped into $W[x_i, i]$. Given an arbitrary link $g \in G$, $f+g_k-g \in \psi(f) \subseteq E(F) \cup B(F)$. Now consider the following two cases:

- Case 1 ($g \geq i$): In this case, $f + g_k - g$ corresponds to element $W[x_i, i+g_k-g]$ which is either exterior or border since $\psi(f) \subseteq E(F) \cup B(F)$. In other words, $x_{i+g_k-g} \leq x_i$; this implies that every interior element in column $i+g_k-g$ reaches only an interior element in column i using link g.

- Case 2 ($g < i$): In this case, $f+g_k-g$ corresponds to element $W[x_i+1, i+g_k-g]$ which is either exterior or border since $\psi(f) \subseteq E(F) \cup B(F)$. In other words, $x_{i+g_k-g} \leq x_i + 1$; this implies that every interior element in column $i+g_k-g$ reaches only an interior or border element in column i using link g.

Since g is arbitrary, every interior element in every column which can reach column i must reach either an interior or a border element in column i. Since i is arbitrary, it follows that the exterior of F is not reachable from the interior of F; thus, F is catastrophic. □

The algorithm generates the faults f_i sequentially. Starting with $f_0 = 0$ (i.e., $W[0,0] = 1$), it determines the location of the elements of $\psi(f_i)$ in W. Because of Theorem 1, any such element must be either exterior or border; the algorithm checks any such location: if it is already an exterior element (i.e., the border element for that column has already been found), it is ignored; if it is not exterior, it must become a border element, and the corresponding new fault is added to the pattern. To determine whether or not a location corresponding to an element in $\psi(f_i)$ is exterior, the algorithm uses a Boolean array V of size g_k: $V[j] = 1$ indicates that the border element for column j has already been determined. Whenever a new border (i.e., new fault) is found, the corresponding entry in V is set to 1.

For the algorithm to operate correctly, the faults f_i's must be considered sequentially. To ensure sequential processing of the faults, a *heap* structure is employed. A *heap* [1] is an implicit data structure (i.e., implementable without pointers in an array) which supports the following operation: *Insert(x)* and *Extract-Min*. The *Extract-Min* operation will return the smallest element stored in the structure. Both operations can be performed in time $O(\log N)$ where N is the number of elements in the *heap*. The elements we insert are couples (i,j) denoting the indices of the entries in W corresponding to the faults in F so far determined. The total ordering enforced on the couples is a lexicographic one: that is, $(i,j) < (i',j')$ if and only if either $i < i'$ or $i = i'$ & $j < j'$.

The Algorithm:

```
Begin
    Let V[i] := 0  for  1 ≤ i ≤ g_k − 1;
    Let V[0] := 1; l := 0; f_l := 0;
    Insert(HEAP,[0,0]);
    while NonEmpty(HEAP) do
        [i,j] := Extract-MIN(HEAP);
        for r := k − 1 downto 1 do
            if j > g_r then
                ī := i + 1;
                j̄ := j − g_r;
            else
                ī := i;
                j̄ := j + g_k − g_r;
            endif
            Test-and-Insert;
            ī := i + 1;
            j̄ := j + g_r;
            if j̄ > g_k then
                ī := ī + 1;
                j̄ := j̄ mod g_k;
            endif
            Test-and-Insert;
    enddo
End.
```

```
Procedure Test-and-Insert;
Begin
    if V[j̄] = 0 then
        l := l + 1;
        f_l := ī g_k + j̄ + 1;
        V[j̄] := 1;
        Insert(HEAP,[ī,j̄]);
    endif
End;
```

Let $\mathcal{F}_b$ denote the fault pattern constructed by the improved algorithm for bidirectional links.

Example 2: $\mathcal{F}_b = \{0,5,9,11,14,16,18,22,23,27\}$ is the reference fault pattern with 10 faults, obtained by the algorithm, for an array of PEs with bidirectional links with link configuration $G = \{1,5,10\}$.

Property 1 *$\mathcal{F}_b$ is catastrophic for bidirectional links.*

Proof: By construction, for every $f \in \mathcal{F}_b$, $\psi(f) \in E(\mathcal{F}_b) \cup B(\mathcal{F}_b)$. Hence, by Theorem 1, $\mathcal{F}_b$ is catastrophic. □

Property 2 *$\mathcal{F}_b$ has maximum width and maximum area.*

Since $\mathcal{F}_b$ has the largest width and area, it follows (by Definition 4) that $\mathcal{F}_b$ is the reference fault pattern.

Property 3 *The improved algorithm generates $\mathcal{F}_b$ for bidirectional links in time $O(kg_k)$.*

Proof: For each fault f_i, the algorithm considers $\psi(f_i)$; this requires $O(k)$ operations. Each f_i is inserted in the heap exactly once (since, once it is entered, the corresponding entry $V[j]$ is set to 1) and is extracted from the heap exactly once. At any time, there are at most k elements in the heap; since insert and extract-min operations in a heap of k elements require $O(\log k)$ time, the total complexity of heap operations is $O(k \log k)$. When f_i is extracted from the heap, the locations corresponding to $\psi(f_i)$ are considered, and for each such location, only a constant number of operations is performed for: 1) determining the corresponding $\bar{i}, \bar{j}$, 2) checking if $V[\bar{j}] = 0$. Since $||\psi(f_i)|| = k$ and $||\mathcal{F}_b|| = g_k$, the total number of these operations is $O(kg_k)$. In summary, the complexity of the algorithm is $O(k \log k + kg_k)$; since $\log k < g_k$, the total complexity of the algorithm is therefore $O(kg_k)$. □

4 Conclusions

In this paper, we have presented an alternate algorithm for the construction of the reference fault pattern for VLSI reconfigurable arrays in which the links are bidirectional. The complexity of the new algorithm is $O(kg_k)$ which is significantly better than $O(W_F + kg_k)$, the complexity of the existing algorithm since the window size W_F of a fault pattern can be as large as g_k^2.

ACKNOWLEDGEMENT

This work was supported in part by Natural Sciences and Engineering Research Council of Canada under Operating Grant A2415 and in part by Progetto Finalizzato Sistemi Informatici Calcolo Parallelo.

5 References

[1] Aho, A. V., Hopcraft, J. E. and Ullman, J. D., *The Design and Analysis of Computer Algorithms*, Addison–Wesley Publishing Company, 1974.

[2] Greene, J. W. and Gamal, A. E., "Configuration of VLSI Arrays in the Presence of Defects," *Journal of the ACM*, Vol. 31, No. 4, 1984, pp. 694-717.

[3] Koren, I. and Pradhan, D. K., "Introducing Redundancy into VLSI Designs for Yield and Performance Enhancement," in *Proc. 15th Int'l Conf. on Fault-Tolerant Computers*, 1985, pp. 330-335.

[4] Li, H. F., Jayakumar, R. and Lam, C., "Restructuring for Fault-Tolerant Systolic Arrays," *IEEE Trans. on Computers*, Vol. C-38, No. 2, 1989, pp. 307-311.

[5] Nayak, A., Pagli, L. and Santoro, N., "Recognition of Catastrophic Faults in Reconfigurable Arrays with Arbitrary Link Redundancy," submitted for publication.

[6] Nayak, A. and Santoro, N., "Bounds on Performance of VLSI Processor Arrays," in *5th Int'l Parallel Processing Symp.*, Anaheim, 1991, pp. 364-370.

[7] Nayak, A. and Santoro, N., "Performance Analysis of Redundant Ring Networks," in *Int'l Conf. on Parallel Computing*, London, 1991.

[8] Nayak, A., Santoro, N. and Tan, R., "Fault-Intolerance of Reconfigurable Systolic Arrays," in *20th Int'l Symp. on Fault-Tolerant Computers*, Newcastle upon Tyne, 1990, pp. 202-209.

[9] Negrini, R., Sami, M. G. and Stefanelli, R., "Fault-Tolerance Techniques for Array Structures used in Supercomputing," *IEEE Computer*, Vol. 19, No. 2, 1986, pp. 78-87.

[10] Popli, S. P. and Bayoumi, M. A., "Fault Diagnosis and Reconfiguration for Reliable VLSI Arrays," in *Proc. Conf. on Computers and Communications*, Phoenix, 1988, pp. 69-73.

[11] Roychowdhury, V. P., Bruck, J. and Kailath, T., "Efficient Algorithms for Reconfiguration in VLSI/WSI Arrays," *IEEE Trans. on Computers*, Vol. C-39, No. 4, 1990, pp. 480-489.