ELSEVIER

INTEGRATION, the VLSI Journal 20 (1996) 327–342

INTEGRATION
the VLSI journal

On testing for catastrophic faults in reconfigurable arrays with arbitrary link redundancy[1]

Amiya Nayak[a,*], Linda Pagli[b], Nicola Santoro[a]

[a]*Center for Parallel and Distributed Computing, School of Computer Science, Carleton University, Ottawa, Canada K1S 5B6*

[b]*Dipartimento di Scienze dell'Informazione, University of Pisa, Corso Italia 40, 56100 Pisa, Italy*

Received 20 December 1994

Abstract

Fault tolerance through the incorporation of redundancy and reconfiguration is quite common. The distribution of faults can have severe impact on the effectiveness of any reconfiguration scheme; in fact, patterns of faults occurring at strategic locations may render an entire system unusable regardless of its component redundancy and its reconfiguration capabilities. Testing of catastrophic faults was given for reconfigurable arrays with 2-link redundancy; i.e., a bypass link of fixed length is provided to each element of the array in addition to the regular link.

In this paper, we study the more general case of arbitrary (but regular) link redundancy. In particular, we focus on the problem of deciding whether a pattern of k faults is catastrophic for a k-link redundant system; i.e., in addition to the regular link of length $g_1 = 1$, each element of the array is provided with $k-1$ bypass links of length $g_2, g_3, \ldots, g_k$, respectively.

We study this problem and prove some fundamental properties which any catastrophic fault pattern must satisfy. We then show that these properties together constitute a necessary and sufficient condition for a fault pattern to be catastrophic for a k-link redundant system. As a consequence, we derive a provably correct testing algorithm whose worst-case time complexity is $O(kg_k)$; this also improves on the previous algorithm for $k = 2$.

Keywords: Fault tolerance; Systolic arrays; Catastrophic fault patterns; Testing

1. Introduction

Faults can occur in all systems at all levels. Therefore, a proper fault-tolerance mechanism must be in place to cope with possible failures. A common and practical approach for achieving fault-tolerance in VLSI-based regular architectures is by incorporating component redundancy

*Corresponding author. Tel.: 613 788 4333; fax: 613 788 4334; e-mail: nayak@scs.carleton.ca.

[1] A preliminary version of this paper has appeared in the Proceedings of the International Workshop on Defect and Fault Tolerance in VLSI Systems, 1992.

0167-9260/96/$15.00
PII S0167-9260(96)00005-3

and mechanisms for reconfiguration of the architecture. 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 [1–17] 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 main objective of all the reconfiguration algorithms is to map faulty elements to spares (using bypass links) while preserving the high degree of regularity and locality of reference required by the system to perform correctly.

The effectiveness of using redundancy to increase fault tolerance clearly depends on both the amount of redundancy and the reconfiguration capability of the system. It does however depend also on the distribution of the faults in the system. In fact, faults occurring at strategic locations in a regular architecture may have catastrophic effects on the entire structure and cannot be overcome by any amount of clever design. Patterns of faults which can have catastrophic effects are denoted as *catastrophic fault patterns*. From a network prospective, such fault patterns can cause network disconnection. For a given design, it is not difficult to identify a set of elements whose failure will have catastrophic consequences. For example, in a linear array of PEs with no link redundancy, a single PE fault in any location is sufficient to stop the flow of information from one side to the other. Similarly, the same array with $k-1$ bypass links $\{g_2, g_3, \ldots, g_k\}$ cannot tolerate g_k (not k) 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 (random distribution) of g_k faults, not in a block, which can be fatal for the system. Therefore, the characterization of such fault patterns is obviously crucial for the identification, testing and dectection of such catastrophic events.

The fault patterns that are catastrophic have been extensively studied for classes of regular architectures [18]. The knowledge about these catastrophic fault patterns can be used in many ways to improve the reliability of regular systems. In fact, the characterization of these fault patterns has been utilized in [19] to assess the reliability of redundant VLSI arrays. 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. The patterns of faults can very well be the distribution of faults, frequently observed in the post-manufacturing phase or in an application domain.

In this paper, we are concerned with the development of efficient testing schemes; that is, efficient mechanisms which automatically determine whether or not an observed/detected pattern of faults will have catastrophic consequences. The availability of such testing schemes can have many practical consequences; e.g., they can be used after the fault detection/location phase to determine whether reconfiguration is possible, well ahead of time before the system is used in a critical operation. Similarly, the testing schemes can be used when generating fault patterns to test for reconfigurability.

A fault pattern is a *cut-set* of the graph corresponding to the architecture under consideration; this correspondence was first observed in [20]. Testing if a fault pattern is catastrophic for the regular architecture is equivalent to checking if the fault pattern is a *cut-set* of the corresponding graph [21]. For an array A of N processors and links $\{g_1, g_2, \ldots, g_k\}$ for each processor, a standard cut-set algorithm for testing connectivity would have time complexity $O(Nk)$. In the case of redundant arrays, N is much greater than k and g_k. Noticing that generic algorithms for

finding cut-sets [22] are inefficient for our special case, our goal is to find an efficient algorithm for testing catastrophic fault patterns in redundant arrays.

The problem introduced here can be modelled within the framework of network reliability, especially the reliability of communication networks. Reliability is an important issue in the design of a communication network. Many communication networks provide fault-tolerant routing mechanisms for bypassing any faulty node making use of bypass links. In such networks, successive failures can lead to a state of network disconnection (a catastrophe) whereby one or more non-faulty nodes are cut off from the rest of the network.

The problem of testing whether or not a fault pattern is catastrophic for a redundant array with a given level of link redundancy has been addressed only for specific cases. In particular, an $O(g^2)$ testing algorithm was derived in [18] for 2-link redundant systems, where g is the length of the bypass link.

In this paper, we study the more general case of testing catastrophic fault patterns in reconfigurable arrays with arbitrary link redundancy; i.e., $k > 1$. For these, we prove some fundamental properties which any catastrophic fault pattern must satisfy. We then show that these properties together constitute a necessary and sufficient condition for a fault pattern to be catastrophic for k-link redundant systems. As a consequence, we derive a provably correct testing algorithm whose worst-case time complexity is $O(kg_k)$; thus, we also improve on the previous algorithm for 2-link redundant systems. Since the number of processors N in the redundant array is much greater than k and g_k, the proposed $O(kg_k)$ algorithm is more efficient than any known $O(Nk)$ cut-set algorithm for testing catastrophic faults in regular redundant arrays.

The remaining of this paper is organized as follows. Basic concepts that provide basis for further analysis are introduced in Section 2. In Section 3, a representation for fault patterns based on Boolean matrices is given, and the existing testing algorithm is described. A special fault pattern, called the reference fault pattern, is introduced and its properties are described in Section 4. Necessary and sufficient conditions for a pattern to be catastrophic are given in Section 5. An improved testing scheme is presented in Section 6 followed by a conclusion in Section 7.

2. Preliminaries

In this paper, we will focus on one-dimensional (or linear) arrays. The basic component of such an array is the processing element (PE) as shown in Fig. 1(a). The links can be either unidirectional or bidirectional. There are two kinds of links in redundant arrays: *regular* or *bypass*. Regular links exist between neighboring PEs while bypass links are assumed to exist between non-neighbors. 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. Bypass links are shown in bold in Fig. 1(a).

Let $\mathcal{S} = \{\mathrm{ICU}_l, A, \mathrm{ICU}_r\}$, as shown in Fig. 1(b), represent a systolic system in which ICU_l and ICU_r denote the left and right interface control unit (ICU) respectively and $A = \{p_1, p_2, \ldots, p_N\}$ denotes a one-dimensional array of PEs. The ICUs which interface with the array A are responsible for all I/O functions. In reality, there may be just one ICU looking after both I/O ports. Each $p \in A$ represents a processing element and there exists a direct link between p_i and p_{i+1}, $1 \leqslant i < N$. Any

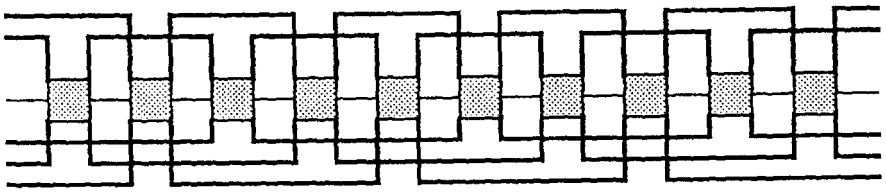


Fig. 1(a). One-dimensional array of processors.

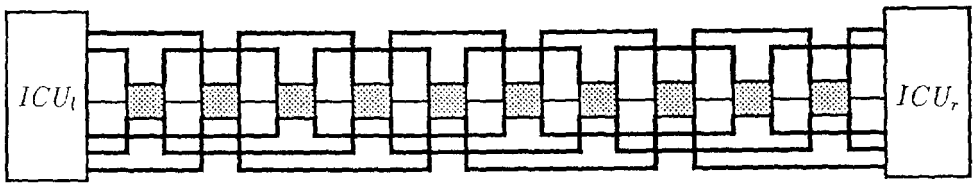


Fig. 1(b). Model systolic system.

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., $|j - i|$.

Definition 1. Given an integer $g \in [1, N-1]$ and an array A of size N, A is said to have link redundancy g, if for every $p_i \in A$ with $i \leqslant N - g$ there exists a link between p_i and p_{i+g}; if $g > 1$, such a link will be called a bypass link.

Note that for each $p_i \in A$ with $i > N - g$ there exists a link between p_i and ICU_r. Similarly, ICU_l is connected to p_g.

The above definition can be extended to a set of bypass links as follows:

Definition 2. The array A has *link redundancy* $G = \{g_1, g_2, \ldots, g_k\}$ where $g_j < g_{j+1}$ and $g_j \in [1, N-1]$, if A has link redundancies $g_1, g_2, \ldots, g_k$.

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*. Notice that $g_1 = 1$ is the regular link, while all other g_i's correspond to bypass links. Every p_i has in-degree (also out-degree) k; ICU_l has out-degree g_k; and ICU_r has in-degree g_k. It is also assumed that the array size is much larger than the length of the largest bypass link, i.e., $N \gg g_k$.

Given a linear array A of size N, a *fault pattern* for A is a set of integers $F = \{f_1, f_2, \ldots, f_m\}$ where $m \leqslant N$, $f_j < f_{j-1}$ and $f_j \in [1, N]$. An assignment of a fault pattern F to A means that for every $f \in F$, p_f is faulty.

Definition 3. The *window* W_F of a fault pattern F is a subset of A that starts with f_1 and finishes with f_m.

Definition 4. The *width* ω_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 $F = \{f_1, \ldots, f_m\}$ then $\| W_F \| = \omega_F = f_m - f_1 + 1$.

Definition 5. A fault pattern F is *catastrophic* for an array A with link redundancy G if ICU_l and ICU_r are not connected in the presence of such an assignment of faults.

In other words, F is catastrophic if the removal of the faulty elements and their incident links will cause the I/Os to become disconnected. A characterization of catastrophic fault patterns was given in [18]. 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 fall within precise bounds for the pattern to be catastrophic; these bounds were established on the width ω_F of the fault pattern for different link configurations. In this paper, we will consider *minimal* catastrophic fault patterns; that is, fault patterns which have exactly g_k faulty PEs.

3. Matrix representation for fault patterns and the existing testing algorithm

We now introduce a representation for fault patterns based on Boolean matrices. This representation will be instrumental in establishing new properties of catastrophic fault patterns and deriving an efficient testing algorithm. The purpose of this section is to define the representation and describe the existing testing algorithm.

Consider an arbitrary fault pattern $F = \{f_1, f_2, \ldots, f_{g_k}\}$, consisting of g_k faults for an arbitrary link configuration $G = \{g_1, g_2, \ldots, g_k\}$. Without loss of generality, assume that $f_1 = 1$. The links can be either unidirectional or bidirectional. We present F by a Boolean matrix W of size $(\omega_F^+ \times g_k)$, where $\omega_F^+ = \lceil \omega_F / g_k \rceil$, defined as follows:

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

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

Example 1. Consider the fault pattern F_1 (shown in Fig. 2) for an array of PEs with bidirectional links with link configuration $G = \{1, 4, 8\}$; $\| F_1 \| = 8$ and $\omega_F = 19$. The Boolean matrix representation of F_1 is shown in Fig. 3.

Let W be the matrix representation of a minimal fault pattern F. Notice that any minimal catastrophic fault pattern satisfies the necessary condition that $\forall j$, there is only one i for which $W[i,j] = 1$. Therefore, we are only interested in fault patterns whose corresponding matrix W has exactly one non-zero entry in every column (otherwise, the pattern is trivially non-catastrophic). Let $x_i = \lfloor f_i - 1/g_k \rfloor$ denote the row coordinate in W of the entry corresponding to $f_i \in F$; let $\{x_0, x_1, \ldots, x_{g_k - 1}\}$ be the ordered multiset of such row coordinates corresponding to F. In the example of Fig. 3, the multiset is $\{1, 1, 1, 2, 2, 2, 2, 3\}$. We now define the *interior*, *exterior*, and *border* elements in the matrix representation of a fault pattern.

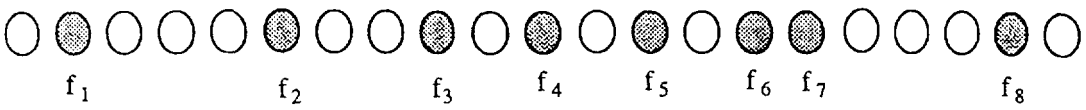


Fig. 2. A fault pattern F_1 for $G = \{1, 4, 8\}$.

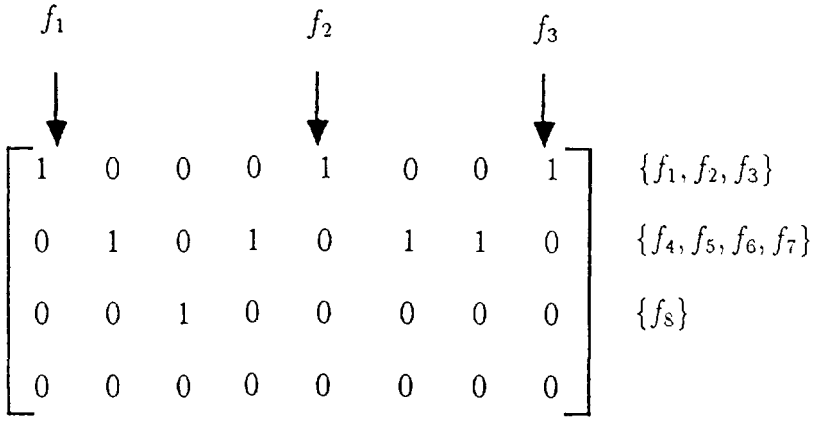


Fig. 3. The matrix representation for F_1 in Fig. 2.

Definition 6. 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 $i > x_l$.

The definition of interior, border, and exterior can now be extended from element to regions as follows:

Definition 7. 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.

Example 2. Consider the fault pattern $F = \{1, 6, 10, 12, 15, 17, 19, 23, 24, 28\}$ with 10 faults in an array with the link configuration $G = \{1, 5, 10\}$ in which all links are bidirectional. The interior, border, and exterior elements are shown in Fig. 4; the first and last rows in Fig. 4 correspond to elements in the array which are outside of W_F and not part of W.

Lemma 1. *A fault pattern F is* catastrophic *for an array A with link redundancy G iff it is not possible to reach any exterior element from any interior element using the links in G.*

Proof. It is easy to see that all interior elements are reachable from ICU_l and all exterior elements are reachable from ICU_r. The lemma follows from Definition 5. □

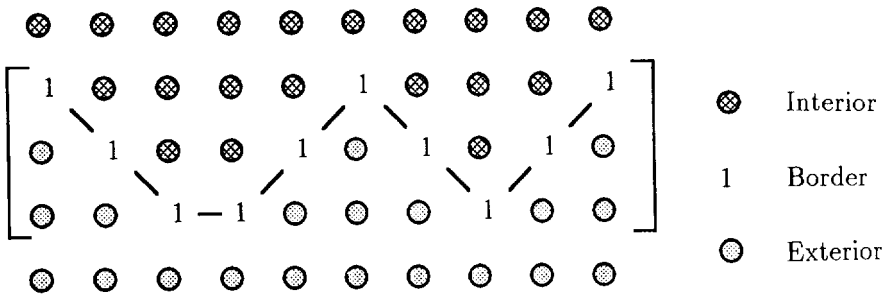


Fig. 4. *Interior*, *exterior* and *border* of a fault pattern.

In the following, using the terminologies just introduced, we describe the existing algorithm for testing whether or not a fault pattern $F = \{f_1, f_2, \ldots, f_{g_k}\}$ with g_k PE faults is catastrophic for a link configuration $G = \{g_1, g_2, \ldots, g_k\}$.

Algorithm 1: Testing if a fault pattern is catastrophic

```
Begin
  TEST := TRUE;
  for every element x ∈ I(F) do
    for every link g ∈ G do
      if x + g ∈ E(F) then TEST := FALSE;
    endfor
  endfor
End.
```

This algorithm considers all elements in $I(F)$ and all links in $G = \{g_1, g_2, \ldots, g_k\}$ and verifies, for each interior element, whether it is possible to reach an exterior element using any link in G. The algorithm can obviously be rephrased so that it terminates as soon as TEST becomes FALSE. In any case, the complexity of the algorithm is bounded above by $\| I(F) \| k$, where k is the number of links in G; the bound is exact if, for example, the pattern is catastrophic. Since $\| I(F) \| = O(g_k^2)$, the worst-case complexity of the algorithm is $O(kg_k^2)$. This is basically the testing algorithm used for 2-link redundant systems [18]; in this case, $k = 2$ and the complexity is $O(g_k^2)$.

The rest of this paper is dedicated to the formulation of a more efficient testing algorithm. We are able to achieve this by exploiting several inherent properties, not previously studied, of the catastrophic fault patterns.

4. Properties of the reference fault pattern

In this section, we will consider a special fault pattern, called the *reference fault pattern*, for a link configuration and describe some of its properties. These properties will be instrumental in the development of an efficient testing algorithm.

The *area* of a fault pattern in its matrix representation can be defined as follows.

Definition 8. The *area* α_F of a fault pattern F is the number of interior and border elements; that is,

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

Recall that the width ω_F (see Definition 4) of a fault pattern $F = \{f_1, f_2, \ldots, f_{g_k}\}$ is $\omega_F = f_{g_k} - f_1 + 1$. A reference fault pattern can now be defined in terms of its width and area as follows:

Definition 9. Given a link configuration G, a *reference fault pattern* (RFP) is a catastrophic fault pattern for G which has largest width ω_F and maximum area α_F.

Algorithms to construct RFPs for arrays with unidirectional and bidirectional links, respectively, were given in [23]. In both cases, the algorithms construct a reference fault pattern $\mathcal{F}$ with time complexity $O(\omega_F + kg_k)$ and space complexity $O(\omega_F)$, where $k = \| G \|$ is the number of links.

We will now establish some properties of the RFPs using the matrix representation described in Section 3. Since any catastrophic fault pattern is invariant with respect to translation (i.e., if F is catastrophic then $F + c$ is also catastrophic for any arbitrary integer c and vice versa), we can assume without loss of generality that $f_1 = 1$.

Let $\mathcal{F}$ be a reference fault pattern. By definition of reference fault pattern, $\omega_{\mathcal{F}}$ is maximal and $\alpha_{\mathcal{F}}$ is maximal.

Consider two fault patterns $F_\alpha = \{f'_1, f'_2, \ldots, f'_{g_k}\}$ and $F_\beta = \{f''_1, f''_2, \ldots, f''_{g_k}\}$ for a given link configuration G. We define the concatenation of F_α and F_β as follows:

Definition 10. Let $\{x'_0, x'_1, \ldots, x'_{g_k - 1}\}$ and $\{x''_0, x''_1, \ldots, x''_{g_k - 1}\}$ be the row coordinates of F_α and F_β respectively in their respective matrix representation. The *concatenation* of F_α and F_β (denoted by $F_\alpha \| F_\beta$) is a fault pattern F whose row coordinates are $\{x_0, x_1, \ldots, x_{g_k - 1}\}$, where $x_i = \{\max(x'_i, x''_i)\}$ for $0 \leqslant i \leqslant g_k - 1$.

An interesting property of the concatenation operation is the following:

Property 1. *Let F_α and F_β be catastrophic for G. Then, their concatenation $F_\alpha \| F_\beta$ is also catastrophic for G.*

Proof. Let $F = F_\alpha \| F_\beta$. We must show that no exterior element of F is reachable from any interior element of F. $F = F_\alpha \| F_\beta$ implies $I(F_\alpha) \cup B(F_\alpha) \subseteq I(F) \cup B(F)$ and $I(F_\beta) \cup B(F_\beta) \subseteq I(F) \cup B(F)$. Let x be any arbitrary element in $I(F)$ and g be an arbitrary link in G. We will now show that $x + g \in I(F) \cup B(F)$.

Case 1: $x \in I(F_\alpha)$. Since F_α is catastrophic, $x \in I(F_\alpha)$ implies $x + g \in I(F_\alpha) \cup B(F_\alpha) \subseteq I(F) \cup B(F)$.

Case 2: $x \in I(F_\beta)$. Similarly, using the fact that F_β is catastrophic, $x \in I(F_\beta)$ implies $x + g \in I(F_\beta) \cup B(F_\beta) \subseteq I(F) \cup B(F)$.

Case 3: $x \in B(F_\alpha)$. In this case, since $x \in I(F)$, $x \in I(F_\beta)$. Therefore, by Case 2, $x + g \in I(F_\beta) \cup B(F_\beta) \subseteq I(F) \cup B(F)$.

Case 4: $x \in B(F_\beta)$. $x \in I(F)$ implies that $x \in I(F_\alpha)$. Therefore, by Case 1, $x + g \in I(F_\alpha) \cup B(F_\alpha) \subseteq I(F) \cup B(F)$.

Since both x and g are arbitrary, it follows that it is not possible to reach any exterior element from any interior element in $F_\alpha \| F_\beta$ using any link in G. Hence, $F_\alpha \| F_\beta$ is catastrophic for G. □

We use this property to prove that, for a link configuration G, there is one and only one reference fault pattern.

Property 2. *For any link configuration G, the reference fault pattern is unique.*

Proof. By contradiction, let $F_\alpha \neq F_\beta$ be two reference fault patterns for G. By definition, $\alpha_{F_\alpha} = \alpha_{F_\beta} = \alpha_{\max}$. By Property 1, $F = F_\alpha \| F_\beta$ is also catastrophic for G; on the other hand, $\alpha_F > \alpha_{F_\alpha} = \alpha_{\max}$, contradicting the definition of maximal area. □

This property implies that $\mathscr{F}$, constructed by the algorithms in [23], is the unique reference fault pattern. We are now in a position to prove the necessary and sufficient conditions for a fault pattern to be catastrophic for a given link configuration.

5. Necessary and sufficient conditions for catastrophe

The first necessary condition establishes an important relationship between a fault pattern F and the reference fault pattern $\mathscr{F}$.

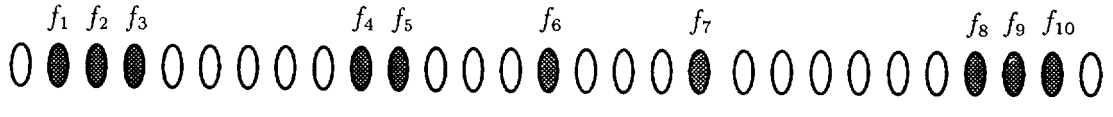
Definition 11. For any two fault patterns F_α and F_β, F_α and F_β *cross* if $I(F_\alpha) \not\subseteq I(F_\beta)$ and $I(F_\beta) \not\subseteq I(F_\alpha)$.

Lemma 2. *Given G, let $\mathscr{F}$ be the reference fault pattern and F be any fault pattern for G. If F and $\mathscr{F}$ cross, then F is not catastrophic for G.*

Proof. We will prove the lemma by contradiction. Suppose F is catastrophic. Since F crosses the reference fault pattern $\mathscr{F}$, it implies that $I(F) \not\subseteq I(\mathscr{F})$. Consider the new fault pattern, $\mathscr{F} \| F$, which is the concatenation of F and $\mathscr{F}$. By Property 1, $\mathscr{F} \| F$ is catastrophic. Furthermore, $\alpha_{\mathscr{F}\|F} > \alpha_{\mathscr{F}}$, contradicting the fact that the reference fault pattern $\mathscr{F}$ has the largest area. Therefore, the lemma follows. □

Example 3. Fig. 5 shows the matrix representation of the reference fault pattern $\mathscr{F}$ and a fault pattern F_2 with 10 faults for $G = \{1, 5, 10\}$. The links in this case are bidirectional. The solid line and the dashed line indicate the border of $\mathscr{F}$ and F_2 respectively. Notice that F_2 crosses $\mathscr{F}$ and, therefore, is not catastrophic. The escape path is shown in the figure.

Lemma 2 expresses a necessary condition for a fault pattern to be catastrophic. However, not crossing $\mathscr{F}$ is not sufficient for a fault pattern to be catastrophic. The following example illustrates such a case.



A fault pattern F_2

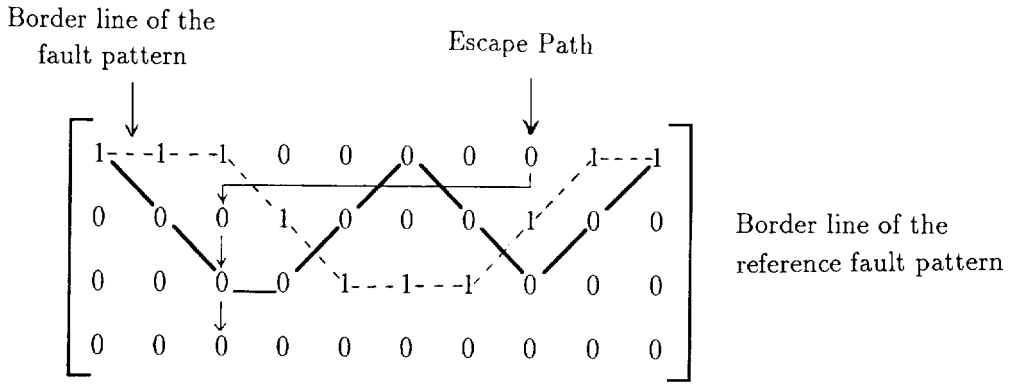


Fig. 5. A fault pattern F_2 which crosses the reference fault pattern for $G = \{1, 5, 10\}$.

Example 4. Fig. 6 shows the matrix representation of $\mathcal{F}$ and a fault pattern F_3 with 10 faults for $G = \{1, 10\}$ when the links are bidirectional. The solid line and the dashed line indicate the border of $\mathcal{F}$ and F_3 respectively. Note that although F_3 does not cross $\mathcal{F}$, it is still not catastrophic. The escape path is shown in the figure.

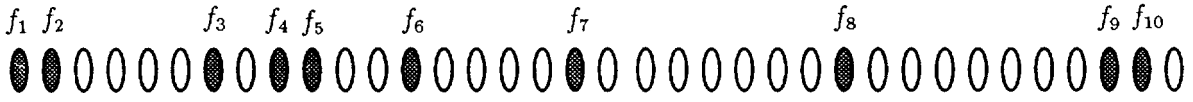
There exist additional conditions which a fault pattern must satisfy to be catastrophic. Such conditions are expressed in this section; based on these conditions, the improved testing algorithm is presented.

Lemma 3. *Let the links be unidirectional. If F does not satisfy the following property then F is not catastrophic for G: for any column y_i $(0 \leqslant y_i \leqslant g_k - 1)$ in W and for any link $g \in G = \{g_1, g_2, \ldots, g_k\}$*

$$x_i \leqslant \begin{cases} x_{i+g} + 1 & \text{if } i + g \leqslant g_k - 1, \\ x_j & \text{otherwise,} \end{cases}$$

where $j = (i + g) \bmod g_k$.

Proof. By contradiction, let F be catastrophic for G and let there exist i and g such that the property does not hold. Consider first the case where $i + g \leqslant g_k - 1$ (see Fig. 7(a)). If the property does not hold then $x_i > x_{i+g} + 1$. The element in position $(x_i - 1, y_i)$ is an interior one; on the other hand, the element in position $(x_i - 1, y_{i+g})$, which is reachable from $(x_i - 1, y_i)$ using link g, is



A fault pattern F_3

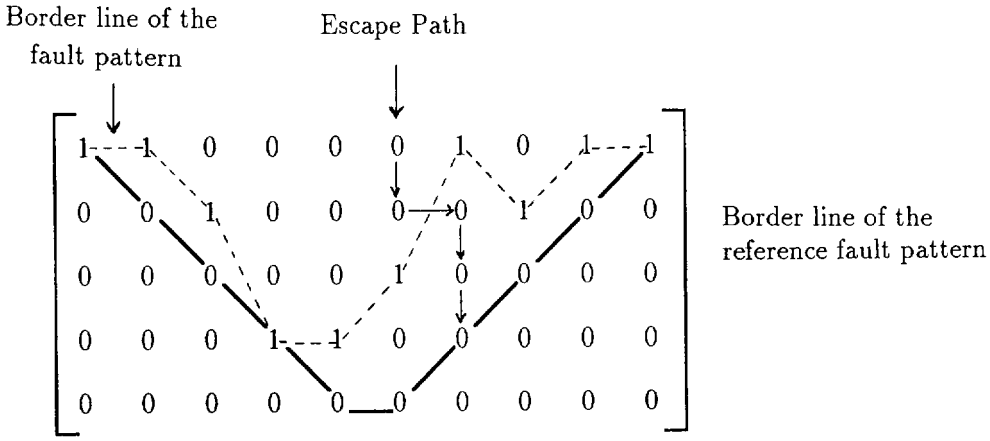


Fig. 6. A fault pattern F_3 which does not cross the reference fault pattern for $G = \{1, 10\}$.

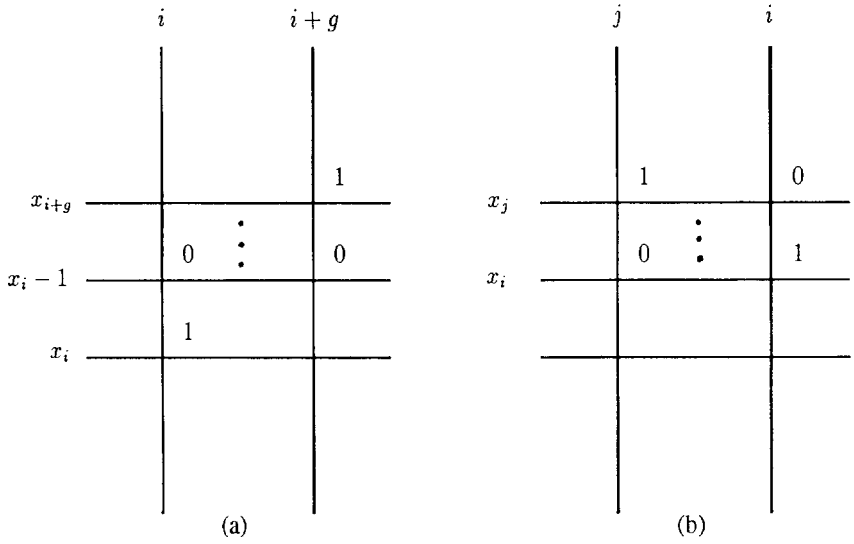


Fig. 7. Illustration of two cases for Lemma 2.

exterior, contradicting the fact that the pattern is catastrophic. Consider now the case $i + g > g_k - 1$ (see Fig. 7(b)). If the property does not hold then $x_j \leqslant x_i - 1$ where $j = (i + g) \bmod g_k$. In this case, the interior element in position $(x_i - 1, y_i)$ can reach the exterior element in position (x_i, y_i), contradicting the fact that the pattern is catastrophic. □

Lemma 3 can be extended to the case of bidirectional links as follows:

Lemma 4. *For bidirectional links, if F does not satisfy the following property then F is not catastrophic for G: for any column* y_i $(0 \leqslant y_i \leqslant g_k - 1)$ *in W and for any link* $g \in G = \{g_1, g_2, \ldots, g_k\}$

$$(1)\ x_i \leqslant \begin{cases} x_{i+g} + 1 & \text{if } i + g \leqslant g_k - 1, \\ x_j & \text{otherwise,} \end{cases}$$

$$(2)\ x_i \leqslant \begin{cases} x_{i-g} + 1 & \text{if } i - g \geqslant 0, \\ x_{\lfloor i + g_k - g \rfloor} + 2 & \text{otherwise,} \end{cases}$$

where $j = (i + g) \bmod g_k$.

Proof. Condition (1) is the same as the condition in Lemma 3, and it must hold when links are used in the forward direction. Condition (2) refers to the case when the links can be used in the backward direction. In the case $i - g \geqslant 0$, the elements in column i can reach the elements in column $i - g$, lying in the same line; hence, the condition is the same as in case (1). For $i - g < 0$, the elements in column i can reach the elements in column $i + g_k - g$, lying one row above. Using the same reasoning as in Lemma 3, the properties can be easily proved. □

The above lemmas state the necessary conditions for a fault pattern to be catastrophic. We will now show that the combination of the conditions expressed by Lemmas 2, 3, and 4 constitute a necessary and sufficient condition.

Theorem 1. *A fault pattern F is catastrophic for a link configuration G if and only if*

(i) *it does not cross the reference fault pattern corresponding to G, and*

(ii) *it satisfies Lemma 3 in the case of unidirectional links and Lemma 4 in the case of bidirectional links.*

Proof. In this proof, we will consider the links to be unidirectional. The proof is similar for the case of bidirectional links. The necessity part has already been shown in Lemmas 2 and 3. To prove the theorem, we must show that if F does not cross the reference fault pattern and satisfies Lemma 3 then the pattern is catastrophic. Consider an arbitrary column i in W and a link g in G. Let $i + g \geqslant g_k - 1$. Since the property of Lemma 3 holds, $x_{i+g} + 1 \geqslant x_i$. Thus, every interior element in column i reaches, using link g, an element in column $i + g$ which is either an interior or a border element. Let $i + g > g_k - 1$. Since the property in Lemma 3 holds, $x_j \geqslant x_i$ where $j = (i + g) \bmod g_k$. Thus, every interior element in column i reaches, using link g, the element in column $i + g$ which is either interior or border.

Since g is arbitrary, it follows that interior elements in column i can only reach elements which are either interior or border. Since i is arbitrary, the proof is completed. □

6. An improved testing scheme

In this section, we use the preceding results to construct an efficient testing algorithm. In particular, the algorithm will verify whether the necessary and sufficient conditions expressed by Theorem 1 are met. The algorithm actually includes a pre-testing phase, ensuring that the width and area of F are not greater than the ones of the reference fault pattern $\mathscr{F}$; recall (from Definition 8) that $\omega_{\mathscr{F}}$ and $\alpha_{\mathscr{F}}$ are maximal.

Algorithm 2: Testing if F is catastrophic for G

```
Begin
  TEST := True;
  Test for violation of maximal area and width;
  if TEST then
    Test for crossing;
    if TEST then Test for property;
  endif
End.
```

Test for violation of maximal area and width (pre-testing)

```
Begin
  if ω_F < ω_F or α_F < α_F then
    TEST := false
  endif
End;
```

Test for crossing

```
Begin
  Let {x_i} and {x̄_i} be the row coordinates of F and F, respectively.
  i := 0;
  repeat
    if x_i > x̄_i then
      TEST := False
    endif
    i := i + 1;
  until i > g_k or not(TEST)
End;
```

Test for property

```
Begin
  i := 0;
  repeat
    j := 1;
    repeat
      if i + g_j ≤ g_k − 1 then x_p := x_{i+g_j} + 1 else x_p := x_{(i+g_j) mod g_k};
      if i − g_j ≥ 0 then x_q := x_{i−g_j} + 1 else x_q := x_{⌊i+g_k−g_j⌋} + 2;
      Case link orientation of
      unidirectional:
        if x_i > x_p then
          TEST := False
        endif
      bidirectional:
        if (x_i > x_p) and (x_i > x_q) then
          TEST := False
        endif
      endcase
      j := j + 1;
    until j > k or not(TEST)
    i := i + 1;
  until i > g_k or not(TEST)
End;
```

The major steps of Algorithm 2 are: Test for Crossing and Test for Property. Test for Crossing requires only the determination of the maximal row coordinate of F and $\mathscr{F}$; thus, it can be done in time $O(g_k)$. For each row coordinate and for each link, Test for Property requires either one or two tests depending on whether the links are unidirectional or bidirectional, respectively; hence, the entire process can be completed in time $O(kg_k)$.

Notice that the complexity of this algorithm represents an improvement on the $O(g_k^2)$ complexity of the existing algorithm for 2-link redundant systems (i.e., $G = \{1, g_k\}$ and $k = 2$); in fact, in this case, the proposed algorithm requires only $O(g_k)$ time.

Finally, observe that it is possible, for some F, that $\|I(F)\| < g_k$. Should this be the case, Algorithm 1, described in Section 3, becomes more efficient than Algorithm 2. Since the value $\|I(F)\|$ can be computed in $O(g_k)$ time, we can integrate the two techniques obtaining a recognition algorithm which has an overall time complexity $O(g_k + \min\{\|I(F)\|, g_k\}k)$.

7. Applications and conclusions

Regular systems are being designed with massive redundancy built into them. These systems also make use of the redundancy to reconfigure in the event of failure in one or more components; normally, a reconfiguration process is triggered as soon as a fault is detected. The success of the

reconfiguration process depends mainly on two factors: the availability of redundant components (level of redundancy) and the distribution of component faults. It is possible to provide a large number of redundant components with the current technology. With the incorporation of massive redundancy into a system, comes the increased likelihood of component failures. The reconfiguration process will now encounter not only more faults but also a variety of fault patterns. This raises the following questions:

- How effective is the reconfiguration process?
- Should the device or system be used in an application which cannot afford reconfiguration failure?

The results of the paper provide some answers to these questions. The proposed scheme can be used to determine the likelihood of a catastrophe in the system or device when some of its components fail; that is, the scheme allows the designer to test efficiently and effectively if the occurrence of specific patterns of faults will pose a problem and cannot be reconfigured. No such other mechanism exists to our knowledge. To be able to recognize such patterns is useful not only to test the effectiveness of the employed reconfiguration scheme but also to prevent a total system shutdown.U

The results of the paper provide a set of tools which can be employed in

1. assessing the fault-tolerance effectiveness of a design; this can be done by specifying the minimum number of faults which the design cannot be guaranteed to withstand,
2. testing whether a design meets the specified fault-tolerance requirements; this can be achieved by comparing the requirements with the ones derived using the properties of the catastrophic fault patterns, and
3. determining the redundancy requirement for the designer to meet a desired level of fault tolerance; this can be done by determining the minimal link configuration for which no catastrophic fault patterns exist below the specified amount of failure.

Furthermore, the results presented here can help to usefully incorporate knowledge of the application field into the design process as feedbacks to the designer. In particular, knowledge of the type and distribution of faults occurring in the application field can be used to determine for which designs those patterns are catastrophic; thus, the designer can remove those designs from further consideration (even though, without that knowledge, they might have been viable choices).

The technical details of this paper can be summarized as follows. We have studied the problem of testing whether a pattern of k faults is catastrophic for a k-link redundant system. We have proved some fundamental properties which any catastrophic fault pattern must satisfy. We have shown that these properties together constitute a necessary and sufficient condition for a fault pattern to be catastrophic for k-link redundant systems. As a consequence, we have derived a provably correct testing algorithm whose worst-case time complexity is $O(kg_k)$; thus, we have improved the previous algorithm for $k = 2$.

Acknowledgements

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

References

[1] K.P. Belkhale and P. Banerjee, Reconfiguration strategies in VLSI processor arrays, *Proc. Int. Conf. on Computer Design* (1988) 418–421.
[2] M. Chean and J.A.B. Fortes, A texanomy of reconfiguration techniques for fault-tolerant processor arrays, *IEEE Comput.* **23** (1990) 55–69.
[3] F. Distante, F. Lombardi and D. Sciuto, Array partitioning: a methodology for reconfigurability and reconfiguration problems, *Proc. Int. Conf. on Computer Design* (1988) 564–567.
[4] J.W. Greene and A.E. Gamal, Configuration of VLSI arrays in the presence of defects, *J. ACM* **31** (1984) 694–717.
[5] S.H. Hosseini, On fault-tolerant structure, distributed fault-diagnosis, reconfiguration and recovery of the array processors, *IEEE Trans. Comput.* **C-38** (1989) 932–942.
[6] J.H. Kim and S.M. Reddy, On the design of fault tolerant two-dimensional systolic arrays for yield enhancement, *IEEE Trans. Comput.* **C-38** (1989) 515–525.
[7] I. Koren and D.K. Pradhan, Introducing redundancy into VLSI designs for yield and performance enhancement, *Proc. 15th Int. Conf. on Fault-Tolerant Computers* (1985) 330–335.
[8] H.T. Kung and M.S. Lam, Fault-tolerance and two-level pipelining in VLSI systolic arrays, *Proc. M.I.T. Conf. on advanced Research in VLSI* (1984) 74–83.
[9] S. Kuo and W.K. Fuchs, Efficient spare allocation for reconfigurable arrays, *IEEE Design Test* (1987) 24–31.
[10] T. Leighton and C.E. Leiserson, Wafer-scale integration of systolic arrays, *IEEE Trans. Comput.* **C-34** (1985) 448–461.
[11] H.F. Li, R. Jayakumar and C. Lam, Restructuring for fault-tolerant systolic arrays, *IEEE Trans. Comput.* **C-38** (1989) 307–311.
[12] R. Negrini, M.G. Sami and R. Stefanelli, Fault-tolerance techniques for array structures used in supercomputing, *IEEE Comput.* **19** (1986) 78–87.
[13] S.P. Popli and M.A. Bayoumi, Fault diagnosis and reconfiguration for reliable VLSI arrays, *Proc. Conf. on Computers and Communications*, Phoenix (1988) 69–73.
[14] A.L. Rosenberg, The Diogenes approach to testable fault-tolerant arrays of processors, *IEEE Trans. Comput.* **C-32** (1983) 902–910.
[15] V.P. Roychowdhury, J. Bruck and T. Kailath, Efficient algorithms for reconfiguration in VLSI/WSI arrays, *IEEE Trans. Comput.* **C-39** (1990) 480–489.
[16] L.A. Shombert and D.P. Siewiorek, Using redundancy for concurrent testing and repairing of systolic arrays, *Proc. 17th Int. Conf. on Fault-Tolerant Computers* (1987) 244–249.
[17] H.Y. Youn and A.D. Singh, A highly efficient design for reconfiguring the processor array in VLSI, *Proc. Int. Conf. on Parallel Processing* (1988) 375–382.
[18] A. Nayak, N. Santoro and R. Tan, Fault-intolerance of reconfigurable systolic arrays, *20th Int. Symp. on Fault-Tolerant Computers*, Newcastle upon Tyne (1990) 202–209.
[19] L. Pagli and G. Pucci, Reliability analysis of redundant VLSI arrays, *Inform. Process. Lett.* **50** (1994) 337–342.
[20] J.P. Hayes, A graph model for fault-tolerant computing system, *IEEE Trans. Comput.* **C-25** (1976) 875–884.
[21] A. Nayak, L. Pagli and N. Santoro, Combinatorial and graph problems arising in the analysis of catastrophic fault patterns, *Proc. 23rd Southeastern Int. Conf. on Combinatorics, Graph Theory and Computing* (1992); *Cong. Numer.* **88** (Utilitas Math.) (1992) 7–20.
[22] F. Harary, *Graph Theory* (Addison-Wesley, Reading, MA, 1969).
[23] A. Nayak, L. Pagli and N. Santaro, Efficient construction of catastrophic patterns for VLSI reconfigurable arrays, *INTEGRATION, VLSI J.* **15** (1993) 133–150.