ARTICLE IN PRESS

Available online at www.sciencedirect.com

ScienceDirect

INTEGRATION, the VLSI journal ▮ (▮▮▮▮) ▮▮▮–▮▮▮

INTEGRATION the VLSI journal

www.elsevier.com/locate/vlsi

Characterization, testing and reconfiguration of faults in mesh networks

Soumen Maity[a], Amiya Nayak[b,*], S. Ramsundar[c]

[a]Department of Mathematics, Indian Institute of Technology, Guwahati 39, Assam, India
[b]School of Information Technology and Engineering (SITE), University of Ottawa, 800 King Edward Avenue, Ottawa, Ont., Canada K1N 6N5
[c]Computer Science & Engineering, Indian Institute of Technology, Guwahati 39, Assam, India

Received 9 March 2006; received in revised form 7 August 2006; accepted 4 November 2006

Abstract

Achieving fault-tolerance through incorporation of redundancy and reconfiguration is quite common. The distribution of faults can have several impacts on the effectiveness of any reconfiguration scheme; in fact, patterns of faults occurring at strategic locations may render an entire VLSI system unusable regardless of its component redundancy and its reconfiguration capabilities. Such fault patterns are called catastrophic fault patterns (CFPs). In this paper, we characterize catastrophic fault patterns in mesh networks when the links are bidirectional or unidirectional. We determine the minimum number of faults required for a fault pattern to be catastrophic. We consider the problem of testing whether a fault pattern is catastrophic. When a fault pattern is not catastrophic we study the problem of finding *optimal* reconfiguration strategies, where optimality is with respect to either the number of processing elements in the reconfigured network (the reconfiguration is optimal if such a number is maximized) or the number of bypass links to activate in order to reconfigure the array (the reconfiguration is optimal if such a number is minimized). The problem of finding a reconfiguration strategy that is optimal with respect to the size of the reconfigured network is NP-complete, when the links are bidirectional, while it can be solved in polynomial time, when the links are unidirectional. Considering optimality with respect to the number of bypass links to activate, we provide algorithms which efficiently find an optimal reconfiguration.

Keywords: VLSI arrays; Catastrophic faults; Array reconfiguration

1. Introduction

VLSI systems have been widely used as parallel models of computation [1,2]. These systems consist of a large number of identical and elementary processing elements locally connected in a regular fashion. Each element receives data from its neighbors, computes and sends the results again to its neighbors. Few particular elements located at the extremes of the systems (these extremes depend on the particular system) are allowed to communicate with the external world. In this paper, we will focus on VLSI mesh networks.

Recently, a number of two-dimensional mesh-based networks have been proposed, owing to their advantages of scalability, modularity, expendability, and degree boundedness. Commercial multiprocessor products based on the mesh have been announced from Ametek and Intel Scientific Computers. Mesh-based designs have been used in the ILLIAC IV computer, Intel Paragon, Cray T3D, and the Goodyear MPP massively parallel computer.

Fault tolerant techniques are very important to VLSI systems. Here we assume that only processors can fail. The likelihood of failure increases with the increase in the number of processing elements. Without the provision of fault-tolerance capabilities, the yield of VLSI chips for such an architecture would be so poor that it would be unacceptable. Thus, fault-tolerant mechanisms must be provided in order to avoid faulty processing elements taking part in the computation. A widely used technique to achieve fault tolerance consists of providing redundancy to the desired architecture [3,4]. In VLSI systems the redundancy consists of additional processing elements, called spares, and additional connections, called bypass

*Corresponding author.
E-mail addresses: soumen@iitg.ernet.in (S. Maity), nayak@uottawa.ca, anayak@site.uottawa.ca (A. Nayak), sundark@iitg.ernet.in (S. Ramsundar).

0167-9260/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.vlsi.2006.11.002

links. Bypass links are links that connect each processor with another processor at a fixed distance greater than 1. The redundant processing elements are used to replace any faulty processing element; the redundant links are used to bypass the faulty processing elements and reach others. 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 faults in the system. There are sets of faulty processing elements for which no reconfiguration strategy is possible. Such sets are called catastrophic fault patterns (CFPs). From a network perspective, such fault patterns can cause network disconnection.

If we have to reconfigure a system when a fault pattern occurs, it is necessary to know if the fault pattern is catastrophic or not. Therefore, it is important to study the properties of catastrophic fault patterns. Till today, the characterization of CFPs is known for linear arrays with the following results. The characterization has been used to obtain efficient testing algorithms both for unidirectional and bidirectional cases [5] with order of magnitude improvement over [6,7]. Efficient techniques has been obtained for constructing CFPs [8]. Using random walk as a tool, a closed form solution for the number of CFPs for uni- and bidirectional links has been provided in [9], an improvement over [10]. Recently, Maity et al. [11] characterize catastrophic fault patterns for two-dimensional arrays.

The main contribution of this paper is complete characterization of CFPs for mesh networks. We determine the minimum number of faults required for a fault pattern to be catastrophic. From a practical viewpoint, above result allow to prove some answers to the question about the guaranteed level of fault tolerance of a design. Guaranteed fault tolerance indicates positive answer to the question as: will the system withstand up to k faults always regardless of how and where they occur? We analyze catastrophic sets having the minimal number of faults. The paper also describes algorithm for testing whether a set of faults is catastrophic or not. In addition, when a fault pattern is not catastrophic, we consider the problem of finding optimal reconfiguration strategies for both unidirectional and bidirectional networks. Where the optimality is with respect to the number of processors in the reconfigured network or with respect to the number of bypass links. The reconfiguration is optimal if number of processing elements is maximized in the former case, while the number of bypass links are to be minimum in the latter case.

The results for arrays apply to a large variety of commercially available array processors such as geometric arithmetic parallel processor (GAPP) [12] of NCR, distributed array processor (DAP) [13] of ICL, England, NASA's massively parallel processor (MPP) [14], and connection machines [15] of Thinking Machines Corporation. The results presented in this paper also apply to a large number of processor arrays which include the systolic arrays [1], reconfigurable array of processors ELSA (European Large SIMD Array) [16], and a variety of special-purpose VLSI and experimental WSI devices for applications such as signal processing, image processing, and numerical computations. Furthermore, the results are equally applicable to the memory chips. Memory chips are the most obvious candidates since the underlying architecture is highly regular and has a large number of identical cells.

2. Preliminaries

In this paper, we will focus on *mesh network*s. The basic components of such a network are the processing elements(PEs) indicated by circles in Fig. 1. There are two kinds of links: *regular* and *bypass*. Regular links connect neighboring (either horizontal or vertical) PEs while bypass links connect non-neighbors. The bypass links are used strictly for reconfiguration purposes when a fault is detected, otherwise they are considered to be the redundant links. We now introduce the following definitions:

Definition 2.1. A mesh network $\mathscr{M} = (V, E)$ consists of a set V of PEs and a set E of links (where a link joins a pair of distinct PEs) satisfying the conditions listed below.

V is the union of five disjoint sets: the set $ICUL = \{ICUL_1, ICUL_2, \ldots, ICUL_{N_1}\}$ of left interface control units, the set $ICUR = \{ICUR_1, ICUR_2, \ldots, ICUR_{N_1}\}$ of right interface control units, the set $ICUT = \{ICUT_1, ICUT_2, \ldots, ICUT_{N_2}\}$ of top interface control units, the set $ICUB = \{ICUB_1, ICUB_2, \ldots, ICUB_{N_2}\}$ of bottom interface control units, and a two-dimensional array $A = \{p_{ij}: 1 \leqslant i \leqslant N_1, 1 \leqslant j \leqslant N_2\}$ of PEs. We sometimes refer to the processing element p_{ij} as (i,j).

E consists of the links obtained as follows. Fix integers $1 = g_1 < g_2 < \cdots < g_k \leqslant N_2 - 1$ and $1 = v_1 < v_2 < \cdots < v_l \leqslant N_1 - 1$. Join p_{ij} to $p_{i'j'}$ by a link if and only if (i) $i = i'$ and $j' - j$ is one of $g_1, g_2, \ldots, g_k$ or (ii) $j = j'$ and $i' - i$ is one of $v_1, v_2, \ldots, v_l$. Also join $ICUL_i$ to $p_{i1}, p_{i2}, \ldots, p_{ig_k}$ and join $p_{i,N_2-g_k+1}, p_{i,N_2-g_k+2}, \ldots, p_{iN_2}$ to $ICUR_i$ by links, for $i = 1, 2, \ldots, N_1$. Similarly join $ICUT_j$ to $p_{1j}, p_{2j}, \ldots, p_{v_lj}$ and join $p_{N_1-v_l+1,j}, p_{N_1-v_l+2,j}, \ldots, p_{N_1j}$ to $ICUB_j$ by links, for $j = 1, 2, \ldots, N_2$.

We assume that $N_2 > g_k$ and $N_1 > v_l$. We also assume N_1 and N_2 are multiples of v_l and g_k, respectively.

Definition 2.2. We call $g_1, g_2, \ldots, g_k$ the *horizontal link redundancies* of $\mathscr{M}$ and $v_1, v_2, \ldots, v_l$ the *vertical link redundancies* of $\mathscr{M}$. We refer to $G = (g_1, g_2, \ldots, g_k | v_1, v_2, \ldots, v_l)$ as the *link redundancy* of $\mathscr{M}$.

Fig. 1 shows a mesh network with $N_1 = 6$, $N_2 = 9$ and $G = (1, 3 \mid 1, 2)$. According to the above assumption, note that $N_1 = 6$ and $N_2 = 9$ are chosen to be multiple of $v_2 = 2$ and $g_2 = 3$, respectively. A link joining two PEs of the type p_{ij} and $p_{i,j+1}$ is called a *horizontal direct link* and a link

Please cite this article as: S. Maity, et al., Characterization, testing and reconfiguration of faults in mesh networks, INTEGRATION, the VLSI journal (2007), doi:10.1016/j.vlsi.2006.11.002

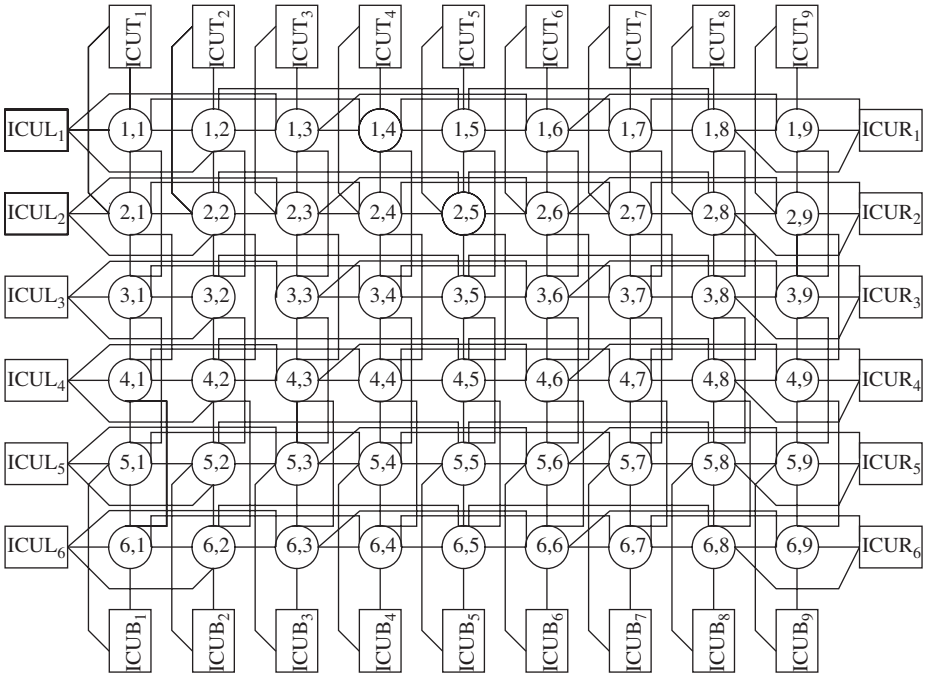

Fig. 1. Mesh network of 54 PEs.

joining two PEs of the type p_{ij} and $p_{i+1,j}$ is called a *vertical direct link*. Direct links are also called *regular links*. Links joining p_{ij} and $p_{i,j+g}$ with $g>1$ are called *horizontal bypass links* and links joining p_{ij} and $p_{i+v,j}$ with $v>1$ are called *vertical bypass links*.

The *length* of the horizontal bypass link joining p_{ij} to $p_{i,j+g}$ is g and the *length* of the vertical bypass link joining p_{ij} to $p_{i+v,j}$ is v.

Note that no links exist in the network $\mathscr{M}$ except the ones specified by G as in Definition 2.1. It is assumed that $ICUL$, $ICUR$, $ICUT$ and $ICUB$ always operate correctly and we are considering information flow from $ICUL \cup ICUT$ to $ICUR \cup ICUB$.

Definition 2.3. Given a two-dimensional array A, a *fault pattern* F for A is the set of faulty processors which can be any non-empty subset of A. An assignment of a fault pattern F to A means that every processing element belonging to F is faulty (and the others operate correctly).

Definition 2.4. A fault pattern is *catastrophic* for the mesh network $\mathscr{M}$ if $ICUL \cup ICUT$ is not connected to $ICUR \cup ICUB$ when the fault pattern F is assigned to A.

Example 2.1. Consider the fault pattern $F = \{(1,7),(1,8),(1,9),(2,7),(2,8),(2,9),(3,3),(3,4),(3,5),(3,6),(4,2),(4,4),(4,5),(4,6),(5,1),(5,2),(5,3),(6,1),(6,2),(6,3)\}$ with link redundancy $G = (1,3\,|\,1,2)$. We see from Fig. 2, that the removal of the processing elements belonging to F along with their incident links disconnects $ICUL \cup ICUT$ and $ICUR \cup ICUB$. Hence F is catastrophic. It is easy to check that F is not catastrophic with respect to link redundancy $G = (1,3\,|\,1,3)$.

Definition 2.5. Let F be a fault pattern in a mesh network $\mathscr{M}$ with link redundancy $G = (1, g_2, \ldots, g_k | 1, v_2, \ldots, v_l)$. If we remove all faulty PEs, their adjacent links and all bypass links from $\mathscr{M}$ then a component of $\mathscr{M}$ will be called a *chunk*. Let $C_0, C_1, \ldots, C_n$ be the chunks of F where C_0 is connected with $ICUL \cup ICUT$, C_n is connected with $ICUR \cup ICUB$ and other C_i's are labeled arbitrarily.

Example 2.2. Consider the fault pattern F of Example 2.1. There are three chunks: $C_0 = \{(1,1),(1,2),(1,3),(1,4),(1,5),(1,6),(2,1),(2,2),(2,3),(2,4),(2,5),(2,6),(3,1),(3,2),(4,1),\}$, $C_1 = \{(4,3)\}$, and $C_3 = \{(3,7),(3,8),(3,9),(4,7),(4,8),(4,9),(5,4),(5,5),(5,6),(5,7),(5,8),(5,9),(6,4),(6,5),(6,6),(6,7)(6,8),(6,9)\}$. Chunks are shown in Fig. 3.

Definition 2.6. A *path* from a working processor (i_0,j_0) to a possibly faulty processor (i_{s+1},j_{s+1}) is a sequence of processors $(i_0,j_0),(i_1,j_1),\ldots,(i_s,j_s),(i_{s+1},j_{s+1})$ such that, for each $k = 0,1,\ldots,s$, processor (i_k,j_k) is a working processor connected by a link to processor (i_{k+1},j_{k+1}) and a processor is used only once. The length of the path is $s+1$. An *escape path* is a path form $ICUL \cup ICUT$ to $ICUR \cup ICUB$.

Our main contribution here is a complete characterization of catastrophic fault patterns for mesh networks. Let $\mathscr{M}$ be a mesh network with link redundancy $G = (g_1, g_2, \ldots, g_k \,|\, v_1, v_2, \ldots, v_l)$, and let F be a fault pattern. Then we prove that, F is catastrophic with respect to $\mathscr{M}$ implies that the cardinality of F, $|F| \geqslant \max\{\frac{N_1}{v_l}, \frac{N_2}{g_k}\} v_l g_k$. We give necessary and sufficient conditions for a fault pattern F to be catastrophic with respect to link redundancy

Please cite this article as: S. Maity, et al., Characterization, testing and reconfiguration of faults in mesh networks, INTEGRATION, the VLSI journal (2007), doi:10.1016/j.vlsi.2006.11.002

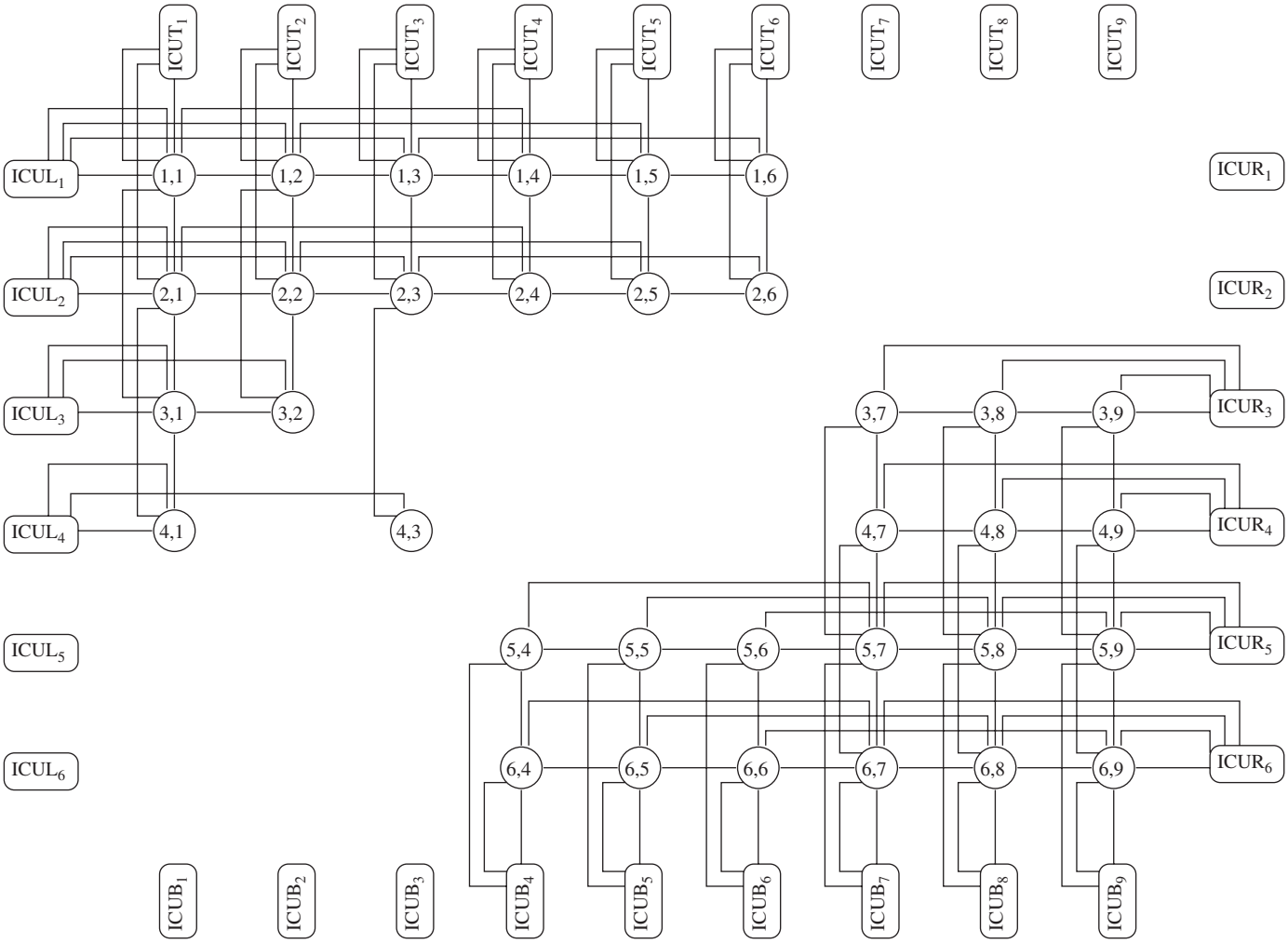

Fig. 2. Network $\mathscr{M}$ after the removal of F and their incident links.

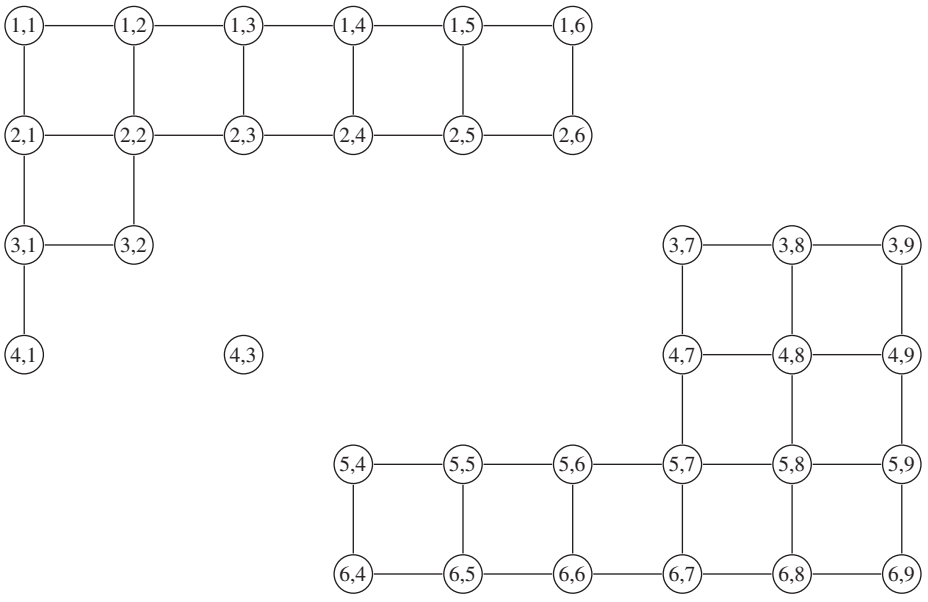

Fig. 3. Chunks of the fault pattern F.

Please cite this article as: S. Maity, et al., Characterization, testing and reconfiguration of faults in mesh networks, INTEGRATION, the VLSI journal (2007), doi:10.1016/j.vlsi.2006.11.002

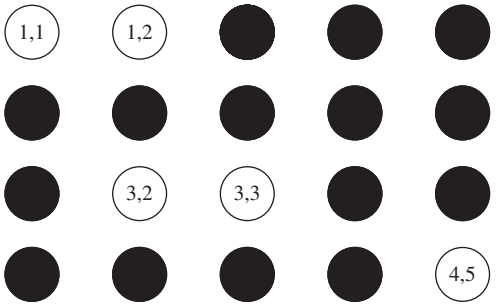

Fig. 4. A mesh with a fault pattern.

$G = (g_1, g_2, \ldots, g_k \mid v_1, v_2, \ldots, v_l)$. We provide an algorithm to test whether a given F is catastrophic with respect to link redundancy $G = (g_1, g_2, \ldots, g_k \mid v_1, v_2, \ldots, v_l)$.

When a fault pattern is not catastrophic, we are interested in finding escape paths. Depending on the fault pattern there can exist several escape paths. We are interested in finding those escape paths that are *optimal* with respect to the size of the reconfigured network or the number of redundant links to be activated to reconfigure the network.

Here optimality is achieved when the size of the reconfigured network is maximized, that is, when the number of processors in the escape path that reconfigured the network is maximized. In this case, an optimal escape path is called a *maximum escape path*, and a reconfiguration set that achieves a maximum escape path is called a *maximum reconfiguration set*.

Example 2.3. Consider the fault pattern $F = \{(1,3),(1,4),(1,5),(2,1),(2,2),(2,3),(2,4),(2,5),(3,1),(3,4),(3,5),(4,1),(4,2),(4,3),(4,4)\}$ in a 4×5 mesh network $\mathscr{M}$ with link redundancy $G = (1,2,3 \mid 1,2)$ as shown in Fig. 4. The maximum escape path, having 4 processing elements, is given by $ICUL_1$ or $ICUT_1$, (1,1), (1,2), (3,2), (3,3) $ICUR_3$ or $ICUB_3$ for both bidirectional or unidirectional case. However maximum escape path is not unique always.

In the latter case, optimality is achieved when the number of redundant links that we have to activate in order to reconfigure the network is minimized. In this case, an optimal escape path is called *minimum escape path*, and a reconfiguration set that achieves a minimum escape path is called a *minimum reconfiguration set*.

Example 2.4. Consider the fault pattern F given in Example 2.3. The escape path $ICUL_3$, (3,2), (3,3), $ICUR_3$ is a minimum escape path since it uses only two bypass links and there are no escape paths that use only 1 bypass link.

3. Characterization of catastrophic fault patterns

In this section, we will characterize the catastrophic fault patterns for mesh networks and prove that the minimum number of faults in a catastrophic fault pattern is a function of N_1, N_2, the length of the longest horizontal bypass link and the length of the longest vertical bypass link.

Theorem 3.1. *Suppose v_l divides N_1 and g_k divides N_2, then F is catastrophic with respect to $\mathscr{M}$ implies that the cardinality of F, $|F| \geqslant \max\{\frac{N_1}{v_l}, \frac{N_2}{g_k}\} v_l g_k$.*

Proof. Suppose to the contrary that $|F| < \max\{\frac{N_1}{v_l}, \frac{N_2}{g_k}\} v_l g_k$. Then partition the two-dimensional array A of PEs into blocks of v_l rows as $A = \begin{pmatrix} A_1 \\ A_2 \\ \vdots \\ A_{N_1/v_l} \end{pmatrix}$ and again partition each block A_i into sub-blocks of g_k columns as $A_i = (A_{i1} \vdots A_{i2} \vdots \ldots \vdots A_{iN_2/g_k})$. For example, sub-block $A_{12} = \{(1,4),(1,5),(1,6),(2,4),(2,5),(2,6)\}$in Fig. 5.

We place the sub-blocks $A_{1j}, A_{2j}, \ldots, A_{N_1/v_l j}$ as consecutive floors to form tower j as shown in Fig. 5. On the other hand, the sub-blocks $A_{i1}, A_{i2}, \ldots, A_{iN_2/g_k}$ form bridge i in between the towers. We shall refer the sub-blocks as floors later. Observe that, each horizontal bypass link of the maximum length joins two consecutive elements in the same *bar* of a bridge. On the other hand, each vertical link of the maximum length joins two consecutive elements in the same *pillar* of a tower. For example, Fig. 5 shows the bar $[(6,3),(6,6),(6,9)]$ and the pillar $[(2,9),(4,9),(6,9)]$. So, in this representation, going up along a pillar corresponds to using the longest vertical bypass links and going right along a bar corresponds to using the longest horizontal bypass links. Note that, there are $N_1 g_k$ bars and $N_2 v_l$ pillars. Now we consider two cases:

Case 1: $\frac{N_1}{v_l} > \frac{N_2}{g_k}$: That is $|F| < N_1 g_k$. Since the number of faulty elements $|F|$ is less than the number of bars, there must be a bar with no faulty element, regardless of the distribution of the fault pattern. Since the left and right of each bar are linked to ICUL and ICUR, respectively, F cannot be catastrophic since we can use the horizontal bypass links of length g_k to avoid the faulty PEs, a contradiction.

Case 2: $\frac{N_1}{v_l} < \frac{N_2}{g_k}$: That is $|F| < N_2 v_l$. Since the number of faulty elements $|F|$ is less than the number of pillars, there must be a pillar with no faulty element, regardless of the distribution of the fault pattern. Since the top and bottom of each pillar are linked to ICUT and ICUB, respectively, F cannot be catastrophic since we can use the vertical bypass links of length g_k to avoid the faulty PEs, a contradiction which proves the theorem. □

This theorem gives us a necessary condition on the minimum number of faults required for blocking a mesh network when $v_l | N_1$ and $g_k | N_2$. In general, we have the following result:

Theorem 3.2. *F is catastrophic with respect to $\mathscr{M}$ implies that the cardinality of F, $|F| \geqslant \max\{N_1 g_k, N_2 v_l\}$.*

Please cite this article as: S. Maity, et al., Characterization, testing and reconfiguration of faults in mesh networks, INTEGRATION, the VLSI journal (2007), doi:10.1016/j.vlsi.2006.11.002

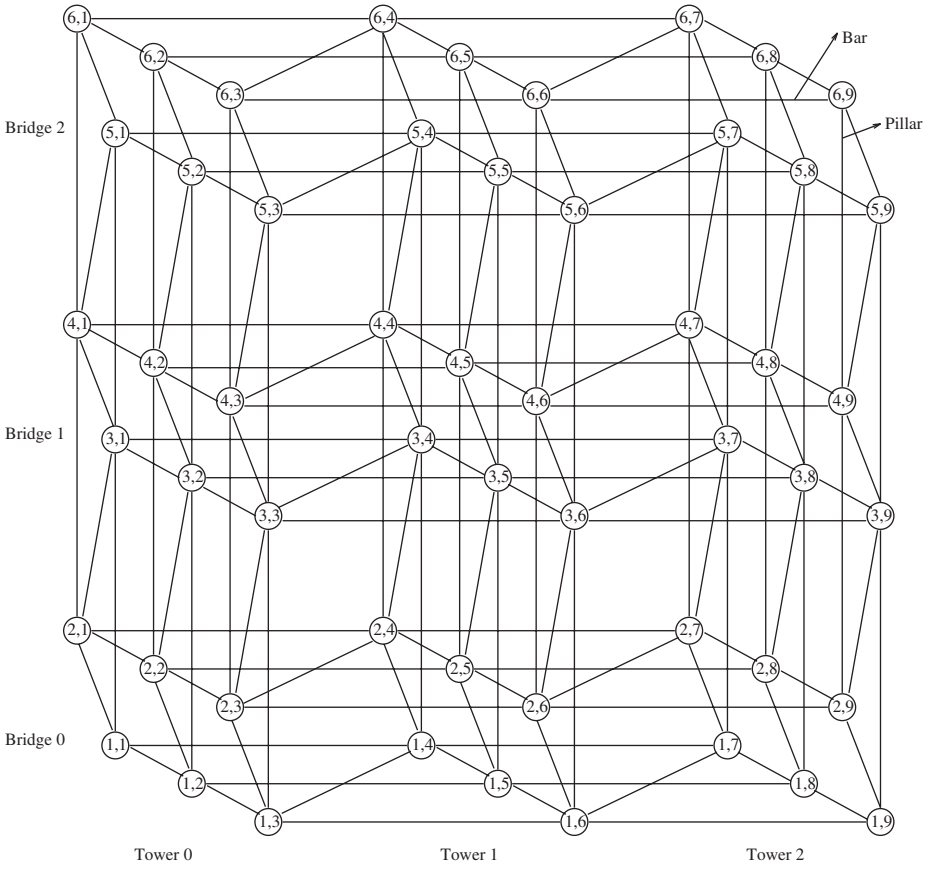

Fig. 5. Tower-Bridge representation of mesh networks of Fig. 1.

The proof of this theorem is similar to that of Theorem 3.1. This tells us that fewer than $\max\{N_1g_k, N_2v_l\}$ faults occurring in A will not be catastrophic. *In the following we will restrict ourselves to the case where there are at least* $\max\{N_1g_k, N_2v_l\}$ *faults, and we will characterize the blocking fault patterns containing exactly* $\max\{N_1g_k, N_2v_l\}$ *faults.*

4. Necessary and sufficient conditions for a pattern to be catastrophic

In this section, we consider the particular case $\frac{N_1}{v_l} = \frac{N_2}{g_k}$. Suppose we are given a fault pattern F with N_1g_k faults in a mesh network with link redundancy $G = (g_1, g_2, \ldots, g_k \mid v_1, v_2, \ldots, v_l)$. We now consider the Tower-Bridge representation of $\mathscr{M}$ used in the proof of Theorem 3.1. We label the v_l rows of a sub-block or floor with $0, 1, \ldots, v_l - 1$ and the g_k columns with $0, 1, \ldots, g_k - 1$. For example consider the sub-block or floor with PEs $(4,1)$, $(4,2)$, $(4,3)$ $(3,1)$, $(3,2)$, $(3,3)$ in Fig. 5. We label the two rows $[(4,1), (4,2), (4,3)]$ and $[(3,1), (3,2), (3,3)]$ with 0 and 1, respectively, and the three columns $[(4,1),(3,1)]$, $[(4,2),(3,2)]$ and $[(4,3),(3,3)]$ with 0, 1 and 2, respectively. The towers and bridges are labeled using $0, 1, 2, \ldots$. With every PE (i,j) we can uniquely associate the 4-tuple (x, y, z, w) where x, y, z and w are the tower label, bridge label, row label and column label of the position (i,j) occupies in the Tower-Bridge representation. We will write

$$W(x,y,z,w) = \begin{cases} 1 & \text{if } (i,j) \in F, \\ 0 & \text{otherwise.} \end{cases}$$

We will some time refer to (x, y, z, w) as the *location of the PE* (i,j).

Suppose now F is a fault pattern such that there is exactly one faulty PE in each pillar and in each bar. Consider the tower x. Then for each row–column combination (z, w), there is exactly one y for which $W(x, y, z, w) = 1$. We then denote this y by h^x_{zw}. For example, $h^0_{10} = 1$ and $h^0_{00} = 2$. This is because PE $(3, 1)$ and $(6, 1)$ are defective PEs. Similarly, consider the bridge y. Then for each row–column combination (z, w), there is exactly one x for which $W(x, y, z, w) = 1$. We then denote this x by r^y_{zw}.

Note that every minimal CFP (i.e., CFP with N_1g_k or N_2v_l faulty PEs) satisfies the conditions stated at the beginning of the preceding paragraph. We now define the interior, exterior and border elements in the Tower-Bridge representation of a minimal CFP.

Definition 4.1. Let F be a minimal CFP. Then the PE of A corresponding to the location (x, y, z, w) is said to be *interior*, *border* or *exterior* element of F, with respect to vertical links, accordingly as $y < h^x_{zw}$, $y = h^x_{zw}$ or $y > h^x_{zw}$. Similarly, the PE of A corresponding to the location (x, y, z, w) is said to be *interior*, *border* or *exterior* element

Please cite this article as: S. Maity, et al., Characterization, testing and reconfiguration of faults in mesh networks, INTEGRATION, the VLSI journal (2007), doi:10.1016/j.vlsi.2006.11.002

of F, with respect to horizontal links, accordingly as $x < r^y_{zw}$, $x = r^y_{zw}$ or $x > r^y_{zw}$. Note that the set of interior (resp. exterior) elements of F with respect to vertical links is not same as the set of interior (resp. exterior) elements with respect to horizontal links.

Example 4.1. Consider the fault pattern $F = \{(1,7),(1,8),(1,9),(2,7),(2,8),(2,9),(3,1),(3,2),(3,6),(4,2),(4,4),(4,6),(5,3),(5,4),(5,5),(6,1),(6,3),(6,5)\}$ for mesh network with link redundancy $G = (1,3|1,2)$. The interior, border, and exterior elements of F with respect to vertical and horizontal links are shown in Figs. 6 and 7, respectively. Note that, PEs $(5,1),(5,2)$ and $(6,2)$ are exterior processors with respect to vertical links but they are interior processors with respect to horizontal links.

Lemma 4.1. *A fault pattern F is catastrophic for a mesh network $\mathscr{M}$ with bidirectional link redundancy G iff it is not possible to reach any exterior element from any interior element and also any interior element from any exterior element, with respect toboth vertical and horizontal links, using the links in $\mathscr{M}$.*

Proof. It is easy to see that all interior elements, with respect to vertical links (resp. horizontal links), are reachable from *ICUB* (resp. *ICUL*) and all exterior elements, with respect to vertical links (resp. for horizontal links), are reachable from *ICUT* (resp. *ICUR*). The lemma follows from Definition 2.4. □

In the case when the network has unidirectional links, we have the following lemma:

Lemma 4.2. *A fault pattern F is catastrophic for a mesh network $\mathscr{M}$ with unidirectional link redundancy G iff it is not possible to reach any exterior element from any interior element, with respect to both vertical and horizontal links, using the links in $\mathscr{M}$.*

5. An algorithm to test weather a fault pattern is catastrophic

5.1. Bidirectional mesh

Let $\mathscr{M}$ be a bidirectional mesh network of N_1N_2 processors with link redundancy $G = (1, g_2, \ldots, g_k \mid 1, v_2, \ldots, v_l)$, and let F be a fault pattern with m faults. A simple way to test if F is catastrophic for $\mathscr{M}$ is to consider a graph whose set of vertices is given by the chunks of working processors. More formally, we construct a graph $H = (V, E)$ as follows: The set V of vertices is $\{C_0, C_1, \ldots, C_n\}$, where C_i's represent chunks of F and $(C_i, C_j) \in E$ if and only if there are two processors, $p_{xy} \in C_i$ and $p_{x'y'} \in C_j$ such that $y = y'$ and $|x - x'| \in \{v_1, v_2, \ldots, v_l\}$ or $x = x'$ and $|y - y'| \in \{g_1, g_2, \ldots, g_k\}$, that is, such that these two processors are connected in $\mathscr{M}$ by a bypass link.

Fact 1. A fault pattern F is not catastrophic for a network $\mathscr{M}$, if and only if C_0 and C_n are connected in the graph H.

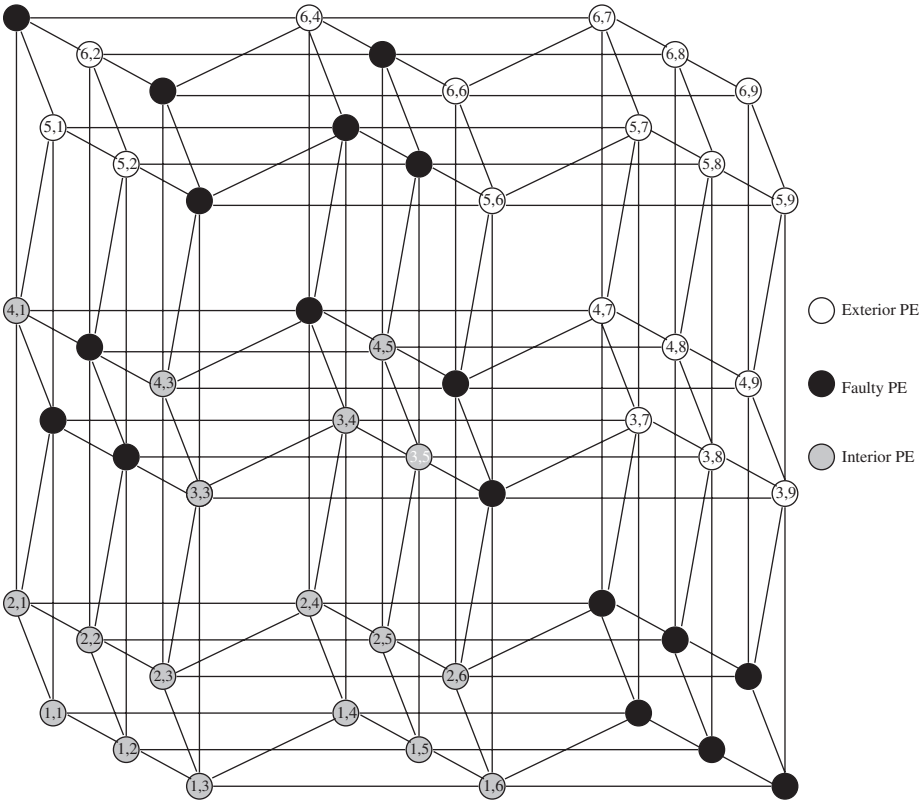

Fig. 6. Interior, exterior and border with respect to the fault pattern given in Example 4.1 for vertical bypass links.

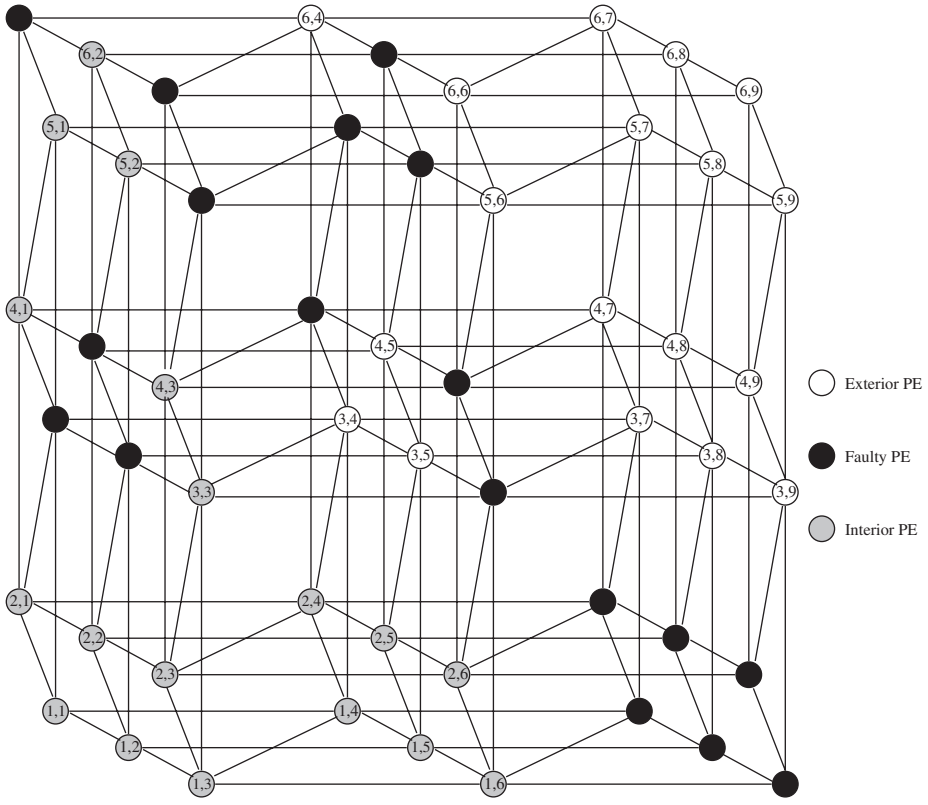

Fig. 7. Interior, exterior and border with respect to the fault pattern given in Example 4.1 for horizontal bypass links.

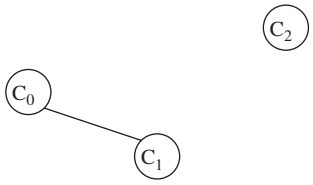

Fig. 8. Graph H.

Example 5.1. Consider the fault pattern F of Example 2.1. Fig. 8 shows the graph H for the fault pattern F and link redundancy $G = (1, 3 \mid 1, 2)$. To see that $(C_0, C_1) \in E$ one may use, for example, p_{23} and p_{43}. Note that, C_0 and C_2 are not connected in H. Hence F is catastrophic with respect to link redundancy $G = (1, 3 \mid 1, 2)$.

5.2. Unidirectional mesh

Given a mesh network $\mathscr{M}$ and a fault pattern F, we construct a graph $G_0 = (V, E)$ as follows: The set V of vertices is the set of working processors and *ICU*s, and the set E of edges are the links between two working processors and the links between working processors and *ICU*s. In addition, there are two more vertices *source* and *sink*. We join *source* to each of the *ICUT*s and *ICUL*s and join each of the *ICUR*s and *ICUB*s to *sink*. If $\mathscr{M}$ is unidirectional then G_0 is directed and undirected otherwise. For the directed graph we preserve the direction of the link, in the edge representing it. We call the graph G_0 the *auxiliary graph* of $\mathscr{M}$ for the fault pattern F.

Example 5.2. Consider the fault pattern $F = \{(1,3),(1,4),(2,1),(2,2),(2,3),(2,4),(3,2),(3,4),(3,5),(4,1),(4,2),(4,4)\}$ in a 4×5 unidirectional mesh network $\mathscr{M}$ with link redundancy $G = (1, 2, 3 | 1, 2)$. Fig. 9 shows the auxiliary graph.

Note that in the auxiliary graph $G_0 = (V, E)$, $|V| = 2N_1 + 2N_2 + w + 2$ and $|E| \leqslant 2N_1 + 2N_2 + w|G|$, where w is the number of working processing elements in the mesh. Also note that the auxiliary graph resulting from unidirectional mesh is directed acyclic in nature (See [17]).

Time required to construct an auxiliary graph $G_0 = (V, E)$ is $O(2N_1 + 2N_2 + w|G|)$, that is, $O(N_1 + N_2 + w|G|)$.

To find out if the fault pattern is catastrophic or not, we assign weight 1 to all the edges in the auxiliary graph. Now, we can run the shortest path algorithm given in [17] for directed acyclic graphs and find out if there exists a path from *source* to *sink*. In a graph (V, E), the shortest path algorithm takes $O(|V| + |E|)$ time. A fault pattern F is not catastrophic for mesh $\mathscr{M}$, if and only if *source* is connected with *sink* in the auxiliary graph. Thus the problem of testing whether a fault pattern is catastrophic for unidirectional mesh network requires $O(N_1 + N_2 + w|G|)$ time.

Please cite this article as: S. Maity, et al., Characterization, testing and reconfiguration of faults in mesh networks, INTEGRATION, the VLSI journal (2007), doi:10.1016/j.vlsi.2006.11.002

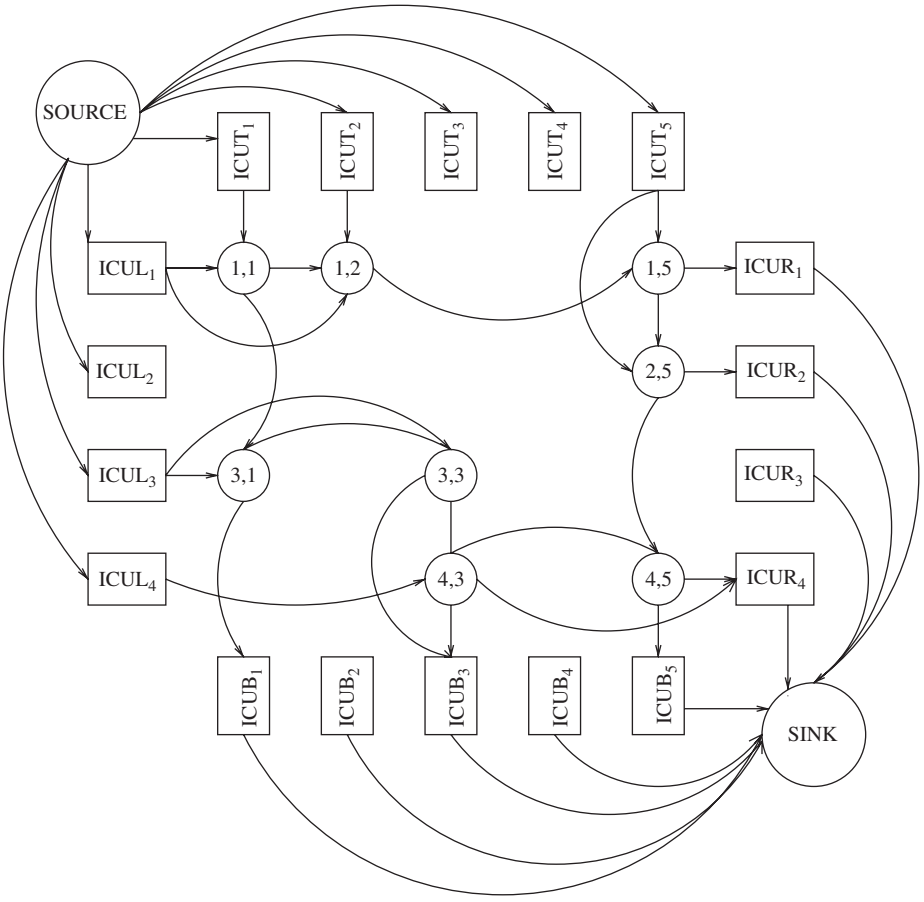

Fig. 9. Auxiliary graph G_0.

6. Maximum escape paths

In this section we consider the problem of finding maximum escape paths. We prove that the problem is NP-complete for a bidirectional mesh network, while for unidirectional mesh we provide an algorithm that finds a maximum escape path in $O(N_1 + N_2 + w|G|)$ time.

6.1. MRL problem in the case of bidirectional links

Consider the following *maximum reconfiguration length* (MRL) problem:

Definition 6.1 (*MRL problem*). Given a bidirectional redundant mesh $\mathscr{M}$, with link redundancy G, a fault pattern F and a positive integer K, is there an escape path of length at least K?

Theorem 6.1. *The MRL problem is NP-complete.*

Proof. We reduce the problem of testing whether there exists a Hamiltonian path (HP) between two given vertices of a graph, known to be NP-complete (see [18]), to MRL problem. Since it is easy to give a non-deterministic polynomial time algorithm that solves MRL problem we conclude that MRL problem belongs to the class NP.

Let $H = (V, E)$ be the input graph of the HP problem. Without loss of generality, assume that $V = \{1, 2, \ldots, n\}$ and that 1 and n are the vertices to be tested.

Consider the following instance of our problem. The mesh $\mathscr{M}$ has $N_1 = 3$ rows and $N_2 = \frac{(6n^3-3n^2-9n+8)}{2}$ columns. For $i = 1, 2, \ldots, n$ define $a_i = (n + i - 2)n^2 + \frac{(n-1)(n-2)+(i-1)(i-2)}{2} + 1$. The horizontal link redundancies are for each edge $(i, j) \in E$, a horizontal bypass link of length $|a_i - a_j|$. Moreover, there is an additional bypass link of length $g = a_1$ (it is easy to see that this is the longest horizontal link connecting $ICUL_2$ and PE $(2, a_1)$, and also PE $(2, a_n)$ and $ICUR_2$). The vertical link redundancy will be singleton $\{1\}$. The fault pattern F consists of all the processing elements in the first and the third rows and all the processing elements in the second row except those in the a_ith column, i.e., in the whole mesh only the processing elements $(2, a_i)$, $1 \leqslant i \leqslant n$, are working. Finally $K = n$.

Notice that the above MRL instance can be constructed in time polynomial in the size of the graph and all the integers occurring in the description of the instance are polynomially related to n.

We will prove that H has a HP if and only if the above instance of the MRL problem admits a solution, i.e., if there is an escape path of size n. In order to prove this, we first need the following four facts.

Please cite this article as: S. Maity, et al., Characterization, testing and reconfiguration of faults in mesh networks, INTEGRATION, the VLSI journal (2007), doi:10.1016/j.vlsi.2006.11.002

(a) The only possible escape path connects $ICUL_2$ and $ICUR_2$. It is easy to notice that all other pairs of ICU's are already disconnected.
(b) Any escape path must traverse $(2,a_1)$ and $(2,a_n)$. Indeed, as the first and the last $g-1$ processing elements are faulty and the length of the longest link is g.
(c) If $(2,a_i)$, with $1<i<n$, is traversed by an escape path, then it must be traversed after $(2,a_1)$ and before $(2,a_n)$. Indeed, let d_{ij}, $1\leqslant i\neq j\leqslant n$, be the distance between $(2,a_i)$ and $(2,a_j)$, i.e., $d_{ij}=|a_i-a_j|=n^2|i-j|+|\frac{(j-1)(j-2)}{2}-\frac{(i-1)(i-2)}{2}|$. Since for $(i\neq j)$ it holds that $n^2|i-j|<d_{ij}<n^2|i-j|+n^2$, then (for $1\leqslant i\neq j,u\neq v\leqslant n$), we have $d_{ij}=d_{uv}$ if and only if $\{i,j\}=\{u,v\}$.
(d) Graph H is isomorphic to the graph consisting of the non-faulty elements $(2,a_i)$, $i=1,2,\ldots,n$ and their incident horizontal links. Indeed, since $d_{ij}<d_{1n}<g$, $1\leqslant i\neq j\leqslant n$, processors $(2,a_i)$ and $(2,a_j)$ are connected by a bypass link, if and only if vertices i and j are connected by an edge in graph H. Moreover, since no other two working processors are at a distance d_{ij}, this bypass link connects only $(2,a_i)$ and $(2,a_j)$.

Now we can prove that there is an escape path of length at least $K=n$ if and only if there is a HP between vertices 1 and n in the graph H. Assume that there is an escape path of size $K=n$. Since in $\mathscr{M}$ there is exactly K working processors, each processor is involved in the escape path. Since all the working processors are traversed, by (a), (b), (c), (d), we conclude that there exists a HP between vertices 1 and n in H (recall that by the definition of path each processor can be traversed at most once).

Conversely, given a HP between vertices 1 and n in H, by (d), it corresponds to a path from $(2,a_1)$ to $(2,a_n)$, which traverses once all the non-faulty processing elements of $\mathscr{M}$. This path can be easily extended to an escape path of size $K=n$ connecting $ICUL_2$ to $(2,a_1)$ and $ICUR_2$ to $(2,a_n)$, respectively, by means of the longest bypass link.

Therefore, we can test if there exists a HP between vertices 1 and n in H by testing if there exists an escape path of size at least K for the array $\mathscr{M}$. □

6.2. MRL problem in the case of unidirectional links

When the mesh is unidirectional, the problem of finding a maximum escape path can be solved in $O(N_1+N_2+w|G|)$ time.

Definition 6.2. An *auxiliary escape path* is a path from *source* vertex to *sink* vertex in an auxiliary graph.

It is easy to note that if there are p edges in an auxiliary escape path then the corresponding escape path contains $p-3$ processing elements.

Given a fault pattern F in an unidirectional mesh network, we first construct the corresponding auxiliary graph $G_0=(V,E)$. To get the escape path with maximum processors, we assign weight -1 to all the edges in E. Now we run the single source shortest path algorithm for directed acyclic graphs (DAG) given in [17] on G_0 with the source vertex as the *source*. The shortest auxiliary path, if any, from *source* to *sink* is obtained. By shortest auxiliary path we mean that the sum of the edge weights in the path is least. This algorithm takes $O(|V|+|E|)$ time. The construction of the graph takes $O(N_1+N_2+w|G|)$ time. It takes $O(|E|)$ time to assign the weight -1 to each edge. So the overall time required to find out an escape path with maximum processing elements, if any, is $O(N_1+N_2+w|G|)$.

7. Minimum escape paths

In this section we consider the problem of finding minimum escape paths. We prove that the problem can be solved in $O(w|G|+(N_1+N_2+w)\log(N_1+N_2+w))$ time if the bypass links are bidirectional, and in $O(N_1+N_2+w|G|)$ time if the bypass links are unidirectional. First we consider the case of bidirectional links.

Given a fault pattern F in a bidirectional mesh network, we construct the auxiliary graph $G_0=(V,E)$ and assign weight 1 to all the edges representing the bypass links (horizontal or vertical) in G_0 and 0 to all the remaining edges. Now run the Dijkstra's single source shortest path algorithm on G_0 with source vertex as the *source*. The shortest auxiliary escape path, if any, is then obtained. This algorithm requires $O(|V|\log|V|+|E|)$ time (using Fibonacci heap data structure, see [17]). The construction of the auxiliary graph takes $O(N_1+N_2+w|G|)$ time and it takes $O(|E|)$ time to assign weight to each edge. Thus the problem of finding minimum escape path for bidirectional mesh network requires $O(w|G|+(N_1+N_2+w)\log(N_1+N_2+w))$ time.

Now we consider the case of unidirectional links. Given a fault pattern F in a unidirectional mesh network, we construct the auxiliary graph $G_0=(V,E)$ and assign weight 1 to all the edges representing the bypass links (horizontal or vertical) in G_0 and 0 to all the remaining edges. Now, repeat the shortest path algorithm for DAG, as in the Section 6.2. The shortest auxiliary escape path, if any, is obtained. Thus the problem of finding minimum escape path for unidirectional mesh network requires $O(N_1+N_2+w|G|)$ time.

8. Conclusions and open problems

In this paper we have completely characterized catastrophic fault patterns for mesh networks. We proved that, F is catastrophic with respect to $\mathscr{M}$ implies that the cardinality of F, $|F|\geqslant\max\{\frac{N_1}{v_l},\frac{N_2}{g_k}\}v_lg_k$ when v_l divides N_1 and g_k divides N_2. Necessary and sufficient conditions are given for a fault pattern to be catastrophic with respect to link redundancy $G=(g_1,g_2,\ldots,g_k\,|\,v_1,v_2,\ldots,v_l)$. Given a mesh network with a link redundancy G, an important problem is to count the number of catastrophic fault

Please cite this article as: S. Maity, et al., Characterization, testing and reconfiguration of faults in mesh networks, INTEGRATION, the VLSI journal (2007), doi:10.1016/j.vlsi.2006.11.002

patterns. The knowledge of this number enables us to estimate the probability that the system operates correctly. The number of catastrophic fault patterns is not known even for the link redundancy $G = (1, g|1, v)$.

Before attempting any reconfiguration it is important to test whether the set of faults is catastrophic. We have provided testing algorithms to test whether a given fault pattern is catastrophic with respect to link redundancy $G = (g_1, g_2, \ldots, g_k \mid v_1, v_2, \ldots, v_l)$. We have considered both the cases when the network has bidirectional links and when the network has unidirectional links.

We have also considered the problem of finding optimal reconfiguration when the set of faults is not catastrophic. Optimality is considered either with respect to the size of the reconfigured network or with the amount of changes needed to reconfigure the network. We proved that when the links are bidirectional, the problem of finding optimal reconfiguration with respect to the size of the reconfigured network is NP-complete. In all the other three cases we provided algorithms which efficiently find an optimal reconfiguration.

References

[1] H.T. Kung, Why systolic architecture?, IEEE Comput. 15 (1982) 37–46.
[2] T. Leighton, C.E. Leiserson, Wafer-scale integration of systolic arrays, IEEE Trans. Comput. C-34 (1985) 448–461.
[3] V. Balasubramanian, P. Banerjee, A fault tolerant massively parallel processing architecture, J. Parallel Distributed Comput. 4 (1987) 363–383.
[4] J. Bruck, R. Cypher, C.T. Ho, Fault-tolerant meshes with minimal number of spares, Proceedings of Third IEEE Symposium on Parallel and Distributed Processing, 1991, pp. 288–295.
[5] A. Nayak, L. Pagli, N. Santoro, On testing of catastrophic faults in reconfigurable arrays with arbitrary link redundancy, Integration VLSI J. 20 (1996) 327–342.
[6] R. De Prisco, A. Monti, L. Pagli, Efficient testing and reconfiguration of VLSI linear arrays, Theor. Comput. Sci. 197 (1998) 105–129.
[7] R. De Prisco, A. De Santis, Catastrophic faults in reconfigurable systolic linear arrays, Discrete Appl. Math. 75 (1997) 105–123.
[8] A. Nayak, L. Pagli, N. Santoro, Efficient construction of catastrophic patterns for VLSI reconfigurable arrays, Integration VLSI J. 15 (1993) 133–150.
[9] S. Maity, B. Roy, A. Nayak, Enumerating catastrophic fault patterns in VLSI arrays with both uni- and bidirectional links, Integration VLSI J. 30 (2001) 157–168.
[10] L. Pagli, G. Pucci, Counting the number of fault patterns in redundant VlSI arrays, Inf. Process. Lett. 50 (1994) 337–342.
[11] S. Maity, A. Nayak, B. Roy, On characterization of catastrophic faults in two-dimensional VLSI arrays, Intergration VLSI J. 38 (2004) 267–281.
[12] E.L. Cloud, The geometric arithmetic parallel processor, in: Proceedings of the Second Symposium on the Frontiers of Massively Parallel Processing, Fairfax, VA, October 1989.
[13] S.F. Reddaway, DAP—a distributed array processor, Proceedings of the Symposium on Computer Architecture, 1973, pp. 61–65.
[14] K.E. Batcher, Design of a massively parallel processor, IEEE Trans. Comput. C-29 (9) (1980) 836–840.
[15] Thinking Machines Corporation, Connection machine model CM-2 technical summary, Thinking Machines Corporation, Technical Report Series HA87-4, 1987.
[16] G. Saucier, J.L. Patry, E.F. Kouka, A reconfigurable wafer scale array for image processing, Proceedings of the International Conference on Wafer Scale Integration, 1989, pp. 277–288.
[17] T.H. Cormen, C.E. Lierson, R.L. Rivest, C. Stein, Introduction to Algorithms, MIT Press, Cambridge, MA, 1990.
[18] M. Garey, D. Johnson, Computers and Intractability, Freeman, New York, 1979.

Soumen Maity is currently an Assistant Professor in the Department of Mathematics of Indian Institute of Technology, Guwahati, India. He received his Ph.D. degree in Combinatorics from Indian Statistical Institute, Kolkata, India, in 2002. His research interests include Combinatorial methods in Statistics and VLSI Designs, and Cryptography.

Amiya Nayak received his B.Math. degree in Computer Science and Combinatorics & Optimization from University of Waterloo in 1981, and Ph.D. in Systems and Computer Engineering from Carleton University in 1991. He has over 17 years of industrial experience, working at CMC Electronics (formerly known as Canadian Marconi Company), Defence Research Establishment Ottawa (DREO), EER Systems and Nortel Networks, in software engineering, avionics and navigation systems, simulation and system level performance analysis. He has been an Adjunct Research Professor in the School of Computer Science at Carleton University since 1994. He had been the Book Review and Canadian Editor of VLSI Design from 1996 till 2002. He is in the Editorial Board of International Journal of Parallel, Emergent and Distributed Systems, International Journal of Computers, Information Technology & Engineering, and the Associate Editor of International Journal of Computing and Information Science. Currently, he is a Full Professor at the School of Information Technology and Engineering (SITE) at the University of Ottawa. His research interests are in the area of fault tolerance, distributed systems/algorithms, and mobile ad hoc networks with over 100 publications in refereed journals and conference proceedings.

Sundarkumar Ramsundar is currently pursuing his B. Tech Degree in Computer Science and Engineering at Indian Institute of Technology, Guwahati, India. His research interests include Fault -Tolerant Computing, Robust System Design for Emerging Nanotechnologies, and Design and Analysis of Algorithms.

Please cite this article as: S. Maity, et al., Characterization, testing and reconfiguration of faults in mesh networks, INTEGRATION, the VLSI journal (2007), doi:10.1016/j.vlsi.2006.11.002