Catastrophic faults in reconfigurable systolic linear arrays

Roberto De Prisco, Alfredo De Santis

MIT Laboratory for Computer Science, 545 NE Technology Square 43-368, Cambridge, MA 02139, USA

Dip. di Informatica ed Applicazioni, Università di Salerno, 84081 Baronissi (SA), Italy

Received 22 November 1994; revised 19 July 1996

Abstract

In regular architectures of identical processing elements, a widely used technique to improve the reconfigurability of the system consists of providing redundant processing elements and connections together with mechanisms of reconfiguration.

In this paper we consider linear arrays of processing elements, with unidirectional bypass links of length $g$. We study those sets of faulty processing elements, called catastrophic, which prevent the reconfiguration. We show that the number of catastrophic faults of $g$ elements is equal to the $(g-1)$th Catalan number. We also provide algorithms to rank and unrank all catastrophic sets of $g$ faults. Finally, we describe a linear-time algorithm that generates all such sets of faults. Our results are useful to provide reliability estimates of linear arrays and for testing the behavior of reconfiguration strategies in the presence of catastrophic faults.

Keywords: Algorithms; Catalan numbers; Catastrophic fault patterns; Fault tolerance; Reliability analysis; Systolic linear arrays

1. Introduction

Systolic systems have been widely used as parallel models of computation. 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 then 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. The processing elements operate in parallel and are synchronized by a global clock. Systolic models allow to perform computations concurrently and several instances of the same problem can be pipelined into the system [15, 17].
The simplest systolic model is the linear array. In such a system the processing elements are connected in a linear fashion: processing elements are arranged in linear order and each element is connected to the previous and the following element. Despite their simplicity, systolic linear arrays have been used to solve several problems. It is well-known how to use a systolic linear array for the matrix-vector multiplication; several other numerical problems (e.g. convolutions, triangular linear systems) have been solved using systolic linear arrays (see, e.g., [15]). The use of systolic linear arrays is not limited to numerical problems. For example, various algorithms that solve the longest common subsequence problem on a systolic array have been devised [14, 28].

Fault-tolerant techniques are very important to systolic systems. Indeed, since the number of processing elements is very large, the probability that a set of processing elements becomes faulty is not small. Thus, fault-tolerant mechanisms must be provided in order to avoid that faulty processing elements take part in the computation. Fault-tolerant techniques for several architectures have been widely studied (we refer the reader to the bibliography). A widely used technique to achieve reconfigurability consists of providing redundancy to the desired architecture (e.g., [2, 5, 9, 10, 19, 13, 30]). In systolic linear arrays the redundancy consists of additional processing elements, called spares, and additional connections, called bypass links. Bypass links are links that connect each processor with another processor at a fixed distance greater than 1. A reconfiguration algorithm has to avoid faulty processing elements using spares and additional connections. Once a systolic linear array has been constructed, when faults occur, it is possible to correct the array using a laser beam to change connections in order to bypass defective processing elements. However, there are sets of faulty processing elements for which no reconfiguration strategy is possible. Such sets are called catastrophic. If we have to reconfigure a system when a faulty set occurs, it is necessary to know if the set is catastrophic or not. Therefore, it is important to study the properties of catastrophic sets.

Nayak et al. [23] proved that a catastrophic set must contain a number of faulty processing elements which is greater than or equal to the length of the longest bypass link. They analyze catastrophic sets having the minimal number of faults and describe algorithms for constructing a catastrophic set. Nayak et al. [21] describe algorithms for testing whether a set of faults is catastrophic or not.

Given a linear array with a set of bypass links, an important problem is to count the number of catastrophic sets. The knowledge of the number of catastrophic sets enables us to estimate the probability that the system operates correctly: assuming that all fault sets are equally likely to appear, dividing the number of catastrophic sets by the total number of fault sets, we obtain the probability that a given fault set is catastrophic. Pagli and Pucci [25] proved tight upper and lower bounds to the number $F^B(g)$ of catastrophic sets of size $g$ for a linear array with one bidirectional bypass link of length $g$. In particular, they proved that $F^B(g) = \Theta(3^g/g^{3/2})$. They also proved that $F^U(g) = O(10^g/g^{3/2})$, where $F^U(g)$ is the number of catastrophic sets of size $g$ for a linear array with one unidirectional bypass link of length $g$. 
In this paper we consider linear arrays with bypass unidirectional links of length $g$. We compute the exact number of catastrophic sets of size $g$. We prove that $F^U(g)$ is equal to the $(g - 1)$th Catalan number. This enables us to prove that $F^U(g) = \Theta(4^g/g^{3/2})$. In order to characterize these catastrophic sets, we also give a classification of all the catastrophic sets: we rank and unrank all catastrophic sets and we provide an efficient algorithm that generates such sets. The ranking of catastrophic sets is useful, for example, to generate catastrophic fault patterns at random to test the behavior of reconfiguration strategies (that should be able to recognize the impossibility of reconfiguring the array).

This paper is organized as follows. Section 2 provides definitions and some known results. In Section 3 we count the number of particular fault sets. As a special case we obtain that the number of catastrophic sets for a linear array with unidirectional bypass link of length $g$ is the $(g - 1)$th Catalan number. In Section 4 we rank and unrank all such catastrophic sets. In Section 5 we describe and analyze a linear-time (in the number of catastrophic sets) algorithm that generates all the catastrophic sets.

2. Preliminaries

In this section we give preliminaries, definitions and some known results. We refer the reader to [20, 21, 23, 25], for a justification of the definitions and for proofs of the results.

Let $A = \{p_0, p_1, \ldots, p_{n-1}\}$ be a linear array of processing elements, which are connected by regular links $(p_i, p_{i+1})$ and by bypass links $(p_i, p_{i+g})$ of fixed length $g \geq 2$, both unidirectional. We refer to this structure as a linear array with link redundancy $g$ or simply as a linear array, when $g$ is clear from the context or immaterial. Fig. 1 shows a linear array of 15 processing elements.

We assume the presence of an external input processor, which we call $I$, connected to $p_0, p_1, \ldots, p_{g-1}$, and an external output processor, which we call $O$, connected to $p_{n-g}, p_{n-g+1}, \ldots, p_{n-1}$. These special connections of $I$ and $O$ give the same degree of reconfigurability to all processing elements, and enable us to focus our attention on that part of $A$ beginning at the first faulty processor and ending at the last faulty processor, assuming that there are more than $g$ processors before the first fault and after the last fault. $I$ and $O$ always operate correctly. In other words, we can assume that $A$ is an infinite array, no matter how many processors are there before the first fault and after the last fault. The connections with $I$ and $O$, except the regular ones, are not drawn in the previous picture.

![Fig. 1. A linear array of 15 processing elements.](image-url)
For a linear array of size $N$ and any link redundancy, a fault pattern $F$ starting at a fixed $p_{f_0}$ is a set of integers $F = \{f_0, f_1, \ldots, f_{m-1}\}$, where $f_{i-1} < f_i$ for $1 \leq i \leq m$ and $f_{m-1} \leq N$. Processor $p_{f_i}$ is faulty if and only if $f_i \in F$. The cardinality of $F$ is $m$.

Given a linear array $A$, a fault pattern $F$ is catastrophic for $A$ if and only if no path exists between $I$ and $O$, once the faulty processors $p_i, i \in F$, and their links are removed.

We denote a fault pattern by FP and a catastrophic fault pattern by CFP.

Because of the special connections of $I$ and $O$, any translation of a fault pattern does not affect the property of catastrophe of the pattern. Therefore, we assume, without loss of generality, that the first fault of any pattern is $p_0$.

A catastrophic fault pattern $F$ for $A$ must contain at least $g$ fault processors. As done in [23, 25], we consider only fault patterns of cardinality $g$, so, in general, $F = \{0, f_1, \ldots, f_{g-1}\}$.

The width $w_F$ of a fault pattern $F$ is defined to be the number of processors between and including the first and the last fault processor in $F$, i.e. $w_F = f_{g-1} - f_0 + 1$.

A necessary condition for a fault pattern $F$ to be catastrophic is $g \leq w_F \leq (g - 1)^2 + 1$ [23].

A convenient way to represent a fault pattern $F$, starting at the fixed processor $p_0$, is the matrix representation [21]. The fault pattern $F$ is represented as a boolean matrix $W$ of size $(g - 1) \times g$, defined, for $0 \leq i, j \leq g - 1$, by

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

**Example 1.** Consider the case $g = 6$ and $F = \{0, 5, 10, 14, 15, 19\}$. The matrix representation of $F$ is shown in Fig. 2.

Observe that in matrix $W$ each regular link corresponds either to two consecutive elements in the same row or to the last element in a row with the first element in the following row, whereas each bypass link corresponds to two consecutive elements in the same column. For a CFP $F$, the matrix $W$ contains only one entry filled by 1 for each column. Indeed, if there were a column of $W$ with two 1, then there would be a column of $W$ with only 0 entries, as $F$ has cardinality $g$. Using the bypass links of this column we can pass over the fault zone, contradicting the hypothesis that $F$ is catastrophic. Therefore, a CFP can be represented by the set of row indices corresponding.

![Fig. 2. Matrix representation of the fault pattern of Example 1.](image-url)
to the entry 1 in columns. Formally, the row representation of a CFP $F$ is the $g$-upla $(r_0, r_1, \ldots, r_{g-1})$, where each $r_i$ is the unique integer such that $W[r_i, i] = 1$. Another convenient way to represent a CFP is the catastrophic sequence [25]. A catastrophic fault pattern is represented as a sequence of $g-1$ integer moves $(m_1, m_2, \ldots, m_{g-1})$, where $m_i$ represents the distance from the row index of the element set to 1 in column $i-1$, to the one in column $i$. Formally, we have that $m_i = r_{i-1} - r_i$.

Example 2. Let $g=6$ and $F=(0,5,10,14,15,19)$. Its catastrophic sequence is $(-3,1,0,1,1)$ and its row representation is $(0,3,2,2,1,0)$.

3. Counting catastrophic faults

In this section we count the number of sets of faulty processors, starting at the fixed processor $p_0$, that satisfy particular conditions. We will use this counting in order to rank and unrank all the CFPs (Section 4) and to design an algorithm that generates all the CFPs (Section 5). The counting gives us the number of catastrophic fault patterns, of size $g$, which turns out to be the $(g-1)$th Catalan number. An alternative and more simple proof of this fact is also provided.

To prove our results we need the following theorem.

Theorem 1 (Nayak et al. [23]). Necessary and sufficient conditions for a fault pattern $F$ of cardinality $g$ to be catastrophic for a unidirectional array with link redundancy $g$ are:

1. $W[0,0] = W[0,g-1] = 1$;
2. for $1 \leq k \leq g-2$, if $W[h,k-1] = 1$ then only one among $W[h-1,k], W[h,k], \ldots$, $W[y-2,k]$ is equal to 1;
3. for $1 \leq k < g-2$, if $W[h,k+1] = 1$ then only one among $W[0,k], W[1,k], \ldots$, $W[h+1,k]$ is equal to 1.

Observe that Theorem 1 is equivalent to the following proposition.

Proposition 1. Necessary and sufficient conditions for a fault pattern $F$ of cardinality $g$ to be catastrophic for a unidirectional array with link redundancy $g$ are:

1. $W[0,0] = W[0,g-1] = 1$;
2. for $0 \leq k \leq g-2$, if $W[h,k+1] = 1$ then only one among $W[0,k], W[1,k], \ldots$, $W[h+1,k]$ is equal to 1.

Making use of the concepts of sequence of moves and of row representation, Proposition 1 can be rewritten as follows.

Proposition 2 (Pagli and Pucci [25]). Necessary and sufficient conditions to have that $(m_1, \ldots, m_{g-1})$ is the catastrophic sequence of a CFP for a unidirectional linear array
with link redundancy $g$ are:
1. $m_i \leq 1$ for $i = 1, \ldots, g - 1$,
2. $\sum_{i=1}^{k} m_i \leq 0$ for $k = 1, \ldots, g - 2$,
3. $\sum_{i=1}^{g-1} m_i = 0$.

**Proposition 3.** Necessary and sufficient conditions to have that $(r_0, r_1, \ldots, r_{g-1})$ is the row representation of a CFP for a unidirectional linear array with link redundancy $g$
are:
1. $r_0 = r_{g-1} = 0$,
2. for $0 \leq c \leq g - 2$, $r_c \leq r_{c+1} + 1$.

Now, we introduce the notion of $(i, j)$-fault pattern.

**Definition 1.** An $(i, j)$-fault pattern, for $i \geq 0$ and $j \geq 1$ such that $i + j \leq g - 1$, is a fault pattern of cardinality $j + 1$, whose matrix representation satisfies
1. $W[0, 0] = 1$;
2. for $0 \leq k \leq j - 1$, if $W[h, k + 1] = 1$ then only one among $W[0, k], W[1, k], \ldots, W[h + 1, k]$ is 1;

Roughly speaking, an $(i, j)$-fault pattern, $(i, j)$-FP for short, is a piece of a CFP, characterized by a matrix representation equal to that of the CFP up to the $j$th column, and filled by zeroes from column $j + 1$ to column $g - 1$ (remember that for an $(i, j)$-fault pattern $i + j \leq g - 1$). Notice that the definition of $(i, j)$-FP is independent of $g$.

**Example 3.** Consider the fault pattern $F = \{0, 14, 19\}$, with link redundancy $g = 6$. The matrix representing $F$ is shown in Fig. 3. $F$ is a $(2, 2)$-FP. If we add to $F$ the set $\{5, 10, 15\}$, $F$ becomes catastrophic, for a linear array with link redundancy $g$. Other $(2, 2)$-FPs are $\{0, 7, 14\}, \{0, 1, 14\}, \{0, 13, 14\}, \{0, 14, 25\}$.

Define $N_{i,j}$, for $i \geq 0$, $j \geq 0$, as the number of $(i, j)$-FPs. Next, we derive and solve a recurrence relation for $N_{i,j}$. 

![Matrix representation of the fault pattern of Example 3.](image)
Lemma 1. **Integers** $N_{i,j}$ **satisfy the relation**

$$N_{i,j} = N_{0,j-1} + \cdots + N_{i,j-1} + N_{i+1,j-1} \quad \text{for } i \geq 0, \ j \geq 2$$  

and

$$N_{i,1} = 1 \quad \text{for } i \geq 0.$$  

**Proof.** An $(i, 1)$-FP has two fault processors, namely $p_0$ and $p_{i+1}$. Thus, there is a unique $(i, 1)$-FP, which implies $N_{i,1} = 1$.

Condition 2 in Proposition 1 tells us that by adding the processor $p_{i+j}$ to a $(k,j-1)$-FP, where $0 \leq k \leq i+1$, we obtain an $(i,j)$-FP and that any $(i,j)$-FP can be constructed in this way. Thus, the number of $(i,j)$-FPs is the sum of the number of $(k,j-1)$-FPs for $0 \leq k \leq i+1$. $\square$

For the degenerate cases $N_{i,0}$ we assume $N_{0,0} = 1$ and $N_{i,0} = 0$ for $i > 0$, so that (1) is true for $j = 1$, too.

Lemma 2. **The solution of the recurrence** (1), (2) **is**

$$N_{i,j} = (i + 2) \frac{(2j + i - 1)!}{(j + i + 1)! (j - 1)!}$$

for $i \geq 0$ and $j \geq 1$.

**Proof.** We prove the formula by induction. Let $j = 1$; we get from (3) that $N_{i,1} = 1$. Fix a row $r \geq 0$ and a column $c \geq 2$, and suppose that (3) is true for every $N_{i,j}$ in the previous column, i.e., for $i \geq 0$ and $j < c$, and for all previous elements on the column $c$, i.e., for $i < r$ and $j = c$. For the induction step we distinguish between two cases: $r = 0$ and $r > 0$. If $r = 0$, from (1), we have

$$N_{0,c} = N_{0,c-1} + N_{1,c-1}$$

$$= 2 \frac{(2c - 3)!}{c!(c - 2)!} + 3 \frac{(2c - 2)!}{(c + 1)!(c - 2)!}$$

$$= 2 \frac{(2c - 1)!}{(c + 1)!(c - 1)!}$$

whereas if $r > 0$, from (1) we have

$$N_{r,c} = N_{0,c-1} + \cdots + N_{r,c-1} + N_{r+1,c-1}$$

$$= N_{r-1,c} + N_{r+1,c-1}$$

$$= (r + 1) \frac{(2c + r - 2)!}{(r + c)!(c - 1)!} + (r + 3) \frac{(2c + r - 2)!}{(r + c + 1)!(c - 2)!}$$

$$= (r + 2) \frac{(2c + r - 1)!}{(r + c + 1)!(c - 1)!}. \quad \square$$
Observe that there is a bijective mapping between CFPs of a linear array with link redundancy \( g \) and \((0, g - 1)\)-FPs. This is straightforward from the definition of \((i, j)\)-FP and from Proposition 1. Thus, \( N_{0, g-1} \) is equal to the number \( F^U(g) \) of unidirectional CFPs for a linear array with link redundancy \( g \). Hence,

\[
F^U(g) = N_{0, g-1}
\]

and

\[
F^U(g) = 2^g \frac{(2g - 3)!}{g!(g - 2)!} = \frac{1}{g} \binom{2g - 2}{g - 1}.
\]

The next theorem provides a more simple proof of this fact.

**Theorem 2.** The number of CFPs for a linear array with unidirectional bypass links of length \( g \) is the \((g - 1)\)th Catalan number, i.e.,

\[
F^U(g) = \frac{1}{g} \binom{2g - 2}{g - 1}.
\]

**Proof.** It is well-known that the \((g - 1)\)th Catalan number represents the number of well-formed expressions of length \( 2g - 2 \) over the alphabet \( \{(, )\} \) (see e.g. [16]). Recall that a well-formed expression of length \( 2k \) is a sequence of \( k \) \( ( \) and \( k \) \( ) \) that satisfies the following property: for each \( i, 1 < i < 2k \), the number of \( ( \) among the first \( i \) letters of the sequence is greater than or equal to the number of \( ) \). In order to prove the theorem it is sufficient to show the existence of a bijective mapping between the set of CFPs and the set of well-formed expressions of length \( 2g - 2 \). Let \( F \) be a CFP and \((m_1, \ldots, m_{g-1})\) its catastrophic sequence. To each integer \( m_i \) we associate the string \( s(m_i) = ((\ldots(0, \text{ consisting of } 1 - r_{m_i}) \) followed by a single \( ) \). To the CFP \( F \) we associate the string \( s(F) \) obtained by concatenating \( s(m_1)s(m_2)\ldots s(m_{g-1}) \). As an example, the CFP \( F \) considered in Example 2, whose catastrophic sequence is \((-3,1,0,1,1)\), has \( s(F) = (((0))()) \). From Proposition 2 we have that \( s(F) \) is a well-formed expression. On the other hand, \( s(F) \) contains exactly \( g - 1 \) \( ) \), so it is a well-formed expression of length \( 2g - 2 \). Conversely, every well-formed expression of length \( 2g - 2 \) can be viewed as a concatenation of \( g - 1 \) strings \( s(m'_i), i = 1, \ldots, g - 1 \). From the definition of well-formed expression we have that integers \( m'_i, i = 1, \ldots, g - 1 \), satisfy Proposition 2. \( \square \)

Using the well-known Stirling approximation [16]

\[
n! = \sqrt{2\pi n} \left(\frac{n}{e}\right)^n \left(1 + O\left(\frac{1}{12n}\right)\right),
\]

we obtain the following asymptotic estimate of the number of CFPs as function of \( g \):

\[
F^U(g) = \frac{4^g}{\sqrt{\pi g}^{3/2}} \left(1 + O\left(\frac{1}{g}\right)\right).
\]
A concept which will turn out to be useful in Sections 4 and 5 is the complement of an \((i,j)\)-FP.

**Definition 2.** A complement of an \((i,j)\)-fault pattern, for a linear array with link redundancy \(g\), for \(0 \leq i \leq j \leq g - 1\), is a fault pattern of cardinality \(g - j\), whose matrix representation satisfies

1. \(W[0,g - 1] = 1\);
2. for \(g - 2 \geq k \geq j + 1\), if \(W[h,k + 1] = 1\) then only one among \(W[0,k], W[1,k], \ldots, W[h + 1,k]\) is 1;
3. \(W[i,j] = 1\).

Informally, a complement of an \((i,j)\)-FP is a piece of a CFP, characterized by a matrix representation filled by zeros from the first column up to the column \(j - 1\) and equal to that of a CFP from the \(j\)th column to column \(g - 1\). Notice that the definition of a complement of an \((i,j)\)-FP depends on \(g\).

**Example 4.** Consider the fault pattern \(F = \{5,10,14,15\}\), with bypass links of length \(g = 6\). The matrix representing \(F\) is shown in Fig. 4. \(F\) is a complement of a \((2,2)\)-FP. If we add to \(F\) the set \(\{0,19\}\), \(F\) becomes catastrophic.

We denote a complement of an \((i,j)\)-FP by \((i,j)^c\)-FP. Let \(M_{i,j}\) be the number of \((i,j)^c\)-FPs. Next we evaluate \(M_{i,j}\).

**Lemma 3.** **Integers \(M_{i,j}\)** satisfy the relation

\[
M_{i,j} = M_{i-1,j+1} + M_{i,j+1} + \cdots + M_{g-j-2,j+1} \quad \text{for } i \geq 0, \ j \geq 0
\]

where \(M_{-1,k}\), for \(k \geq 0\), is assumed to be 0. Moreover,

\[
M_{0,g-1} = 1,
\]

\[
M_{i,g-1} = 0 \quad \text{for } i > 0.
\]

**Proof.** The proof is similar to the proof of Lemma 1. \(\square\)

The next corollary follows immediately from Lemma 3.

![Matrix representation of the fault pattern of Example 4.](image)
Table 1
Values of the $N_{i,j}$'s for $g = 8$

<table>
<thead>
<tr>
<th>$i$</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>14</td>
<td>42</td>
<td>132</td>
<td>429</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>9</td>
<td>28</td>
<td>90</td>
<td>297</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
<td>4</td>
<td>14</td>
<td>48</td>
<td>165</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>1</td>
<td>5</td>
<td>20</td>
<td>75</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>1</td>
<td>6</td>
<td>27</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>1</td>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2
Values of the $M_{i,j}$'s for $g = 8$

<table>
<thead>
<tr>
<th>$i$</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>429</td>
<td>132</td>
<td>42</td>
<td>14</td>
<td>5</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>429</td>
<td>132</td>
<td>42</td>
<td>14</td>
<td>5</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>297</td>
<td>90</td>
<td>28</td>
<td>9</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>165</td>
<td>48</td>
<td>14</td>
<td>4</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>75</td>
<td>20</td>
<td>5</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>27</td>
<td>6</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>7</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

**Corollary 1.** Integers $M_{i,j}$, for $0 \leq i, j \leq g - 1$, satisfy the relations

\[ M_{i,j} = M_{i+1,j} + M_{i-1,j+1} \]  \hspace{1cm} (5)

and

\[ M_{g-1-i} = 1. \]  \hspace{1cm} (6)

Observe that $M_{0,j} = M_{1,j}$ for $j = 0, 1, \ldots, g - 2$, since the number of $(0,j)^{C}$-FPs is equal to the number of $(1,j)^{C}$-FPs.

**Lemma 4.** The number of $(i,j)^{C}$-FPs and the number of $(i,j)$-FPs, for $0 \leq i \leq j \leq g - 1$, are related by

\[ M_{i,j} = N_{i-1,g-i-j}. \]

**Proof.** Define $D_{i,j}$, for $i \geq 0$ and $j > 0$, as follows:

\[ D_{i,j} = M_{i+1,g-1-(i+j)}, \]
Then one gets $M_{i,j} = D_{i-1,g-(i+j)}$. Using this fact, (5), and (6) one can easily show that the $D_{i,j}$ satisfy the same recurrence relation satisfied by the $N_{i,j}$ in Lemma 1. Hence, we have that $D_{i,j} = N_{i,j}$. □

From Lemmas 2 and 4 it follows that, for $0 \leq i \leq j \leq g - 1$,

$$M_{i,j} = N_{i+1,g-i-j}$$

$$= (i + 1) \frac{(2g - 2i - 2j + i - 1 - 1)!}{(g - i - j + i - 1 + 1)!(g - i - j - 1)!}$$

$$= (i + 1) \frac{(2g - i - 2j - 2)!}{(g - j)!(g - i - j - 1)!}$$

Notice that fixing an entry $(i, j)$ of the matrix $W$, the number of CFPs which contain the processor represented by $(i, j)$, i.e., the processor $p_{i+j}$, is $N_{i,j} M_{i,j}$. Since any CFP must contain one and only one of the processors represented by the elements of a fixed column $c$ of $W$, with $1 \leq c \leq g - 2$, we have that

$$F^U(g) = \sum_{i=0}^{g-1-c} N_{i,c} M_{i,c}.$$ 

Tables 1 and 2 show the $N_{i,j}$'s and $M_{i,j}$'s for $g = 8$, respectively.

### 4. Ranking and unranking catastrophic faults

In this section we provide an invertible mapping defined over the set of CFPs for a linear array with unidirectional link redundancy $g$, which assumes values in the set of integers $0, 1, \ldots, F^U(g) - 1$. This function enables us to rank all CFPs. The inverse of the ranking function is given as an algorithm. In the following we consider the row representation of a CFP.

The rank of a CFP $F$ whose row representation is $(r_0, r_1, \ldots, r_{g-1})$ is the integer given by the sum of the $N_{i,j}$'s that satisfy $i < r_j$, for $j = 0, \ldots, g - 1$. Formally, the rank of $F$ is

$$R(F) = \sum_{c=0}^{g-1} R_c(F),$$

where

$$R_c(F) = \begin{cases} \sum_{r=0}^{r_c-1} N_{r,c} & \text{if } r_c > 0, \\ 0 & \text{if } r_c = 0. \end{cases}$$

Observe that $R(0) = R_{g-1}(F) = 0$ since for any CFP $F$ we have $r_0 = r_{g-1} = 0$. In order to prove that $R$ is a bijective mapping between the set of CFPs for a linear array
with unidirectional bypass links of length $g$ and the set of integers $\{0, 1, \ldots, F^U(g) - 1\}$, we need the following lemma.

**Lemma 5.** The integers $N_{i,j}$’s, for $i, j \geq 0$, satisfy the following equality:

$$N_{i,j} = \sum_{c=0}^{j-1} \sum_{r=0}^{i+j-c-1} N_{r,c}. \quad (8)$$

**Proof.** From (1) we have that

$$N_{i,j} = \sum_{r=0}^{i} N_{r,j-1} + N_{i+1,j-1}$$

$$= \sum_{r=0}^{i} N_{r,j-1} + \sum_{r=0}^{i+1} N_{r,j-2} + N_{i+2,j-2}$$

$$\vdots$$

$$= \sum_{c=0}^{j-1} \sum_{r=0}^{i+j-c-1} N_{r,c}. \quad (8)$$

The maximum value of $\mathcal{R}$ is reached when each $r_c, c = 1, \ldots, g - 2$, assumes its maximum value. From Proposition 3 we have that $r_c \leq g - c$. Therefore, the maximum value of $\mathcal{R}$ is reached for the CFP whose row representation is $(0, g-2, g-3, \ldots, 2, 1, 0)$. From (4) and (8) one gets

$$F^U(g) = N_{0,g-1} = \sum_{c=0}^{g-2} \sum_{r=0}^{g-c-2} N_{r,c}. \quad (9)$$

From (7) and (9) and from the fact that $N_{0,0} = 1$ it follows that the maximum value of $\mathcal{R}$ is $F^U(g) - 1$. The function $\mathcal{R}$ is clearly non-negative. It is easy to see that it assumes the value 0 for the CFP whose row representation is $(0, 0, \ldots, 0)$.

The following lemma shows that different CFPs have different rank.

**Lemma 6.** Let $(r_0, r_1, \ldots, r_{g-1})$ and $(s_0, s_1, \ldots, s_{g-1})$ be two row representations of CFPs $F_1$ and $F_2$, respectively. If $F_1 \neq F_2$ then $\mathcal{R}(F_1) \neq \mathcal{R}(F_2)$.

**Proof.** Let $k$ be the greatest index for which $r_k \neq s_k$. Without loss of generality, we can assume that $r_k < s_k$. We have that $\mathcal{R}_j(F_1) - \mathcal{R}_j(F_2)$ for $j = k + 1, k + 2, \ldots, g - 1$, and $\mathcal{R}_k(F_2) - \mathcal{R}_k(F_1) \geq N_{r_k,k}$. By Proposition 3 we have that $r_c \leq r_k + k - c$, for $0 \leq c \leq k$. Hence, $\mathcal{R}_c(F_1) \leq \sum_{r=0}^{r+k-c-1} N_{r,c}$ which implies that $\sum_{r=0}^{k-1} \mathcal{R}_c(F_1) \leq \sum_{r=0}^{k-1} \sum_{r=0}^{r+k-c-1} N_{r,c}$ Since $N_{0,0} = 1$ and $N_{i,0} = 0$ for $i > 0$, by (8) we have that $\sum_{c=1}^{k-1} \mathcal{R}_c(F_1) \leq N_{r_k,k} - 1$. Since each term $\mathcal{R}_j(F_2)$ is non-negative we have that $\sum_{j=1}^{k-1} \mathcal{R}_j(F_2) - \sum_{j=1}^{k-1} \mathcal{R}_j(F_1) < N_{r_k,k}$. As $\sum_{j=k}^{g-1} \mathcal{R}_j(F_2) - \sum_{j=k}^{g-1} \mathcal{R}_j(F_1) \geq N_{r_k,k}$ we conclude that $\mathcal{R}(F_2) - \mathcal{R}(F_1)$ is greater than 0. □
Theorem 3. \( \mathcal{A} \) is a bijective mapping between the set of row representations of CFPs for a unidirectional linear array with link redundancy \( g \) and the set \( \{0, 1, \ldots, F^U(g) - 1\} \).

**Proof.** By Lemma 6 we have that \( \mathcal{A} \) is an injective function between the set of row representations of CFPs for a unidirectional linear array with link redundancy \( g \) and the set \( \{0, 1, \ldots, F^U(g) - 1\} \). On the other hand, these two sets have the same cardinality. Hence the theorem. \( \square \)

Fig. 5 shows algorithm \texttt{UNRANK} which takes in input an integer \( n, 0 \leq n \leq F^U(g) - 1 \), and gives as output the row representation of the CFP whose rank is \( n \). The next lemmas prove the correctness of the algorithm.

**Lemma 7.** \texttt{UNRANK}(\( n \)) is the row representation of a CFP.

**Proof.** Let \((r_0, r_1, \ldots, r_{g-1})\) be the list returned by \texttt{UNRANK}. Since we never change the initial value of \( r_0 \) and \( r_{g-1} \), we have that \( r_0 = r_{g-1} = 0 \). Fix \( s, 0 < s < g - 1 \). Consider the iteration of the second \texttt{for} with \( c = s + 1 \). Let \( q \) be the value assigned to \( r_{s+1} \) at the end of the \texttt{while}. Clearly, at this point we have that \( v < N_{q,s+1} \). Consider the iteration of the second \texttt{for} with \( c = s \). By contradiction suppose that at the end of the \texttt{while} the value assigned to \( r_s \) is greater than \( q + 1 \). During the \texttt{while} \( v \) has been decreased by a value greater than \( \sum_{j=1}^{q+1} N_{j,s} = N_{q,s+1} \). Since before the execution of the \texttt{while} \( v \) was less than \( N_{q,s+1} \), then, at the end of the \texttt{while} \( v \) will be less than zero. This is a contradiction because in the algorithm \( v \) is always non-negative. Hence, \texttt{UNRANK}(\( n \)) satisfies the conditions of Proposition 3. \( \square \)

**Lemma 8.** Let \( F \) be the CFP whose row representation is \texttt{UNRANK}(\( n \)). Then \( \mathcal{A}(F) = n \).

**Proof.** Let \((r_0, r_1, \ldots, r_{g-1})\) be the list returned by \texttt{UNRANK}, which by Lemma 7 is a row representation of a CFP. Consider the last iteration of the second \texttt{for}, that is the

```
\texttt{UNRANK} \( (n) \)
\begin{align*}
v &= n \\
& \text{for } i = 0 \text{ to } g - 1 \ r_i = 0 \\
& \text{for } c = g - 2 \text{ to } 1 \text{ step } -1 \\
& \quad i = 0 \\
& \quad \text{while } v \geq N_{i,c} \text{ do} \\
& \quad \quad v = v - N_{i,c} \\
& \quad \quad i = i + 1 \\
& \quad r_c = i \\
& \text{return } (r_0, r_1, \ldots, r_{g-1})
\end{align*}
```

Fig. 5. Algorithm \texttt{UNRANK}. 

iteration for which $c = 1$. Since $N_{i,1} = 1$ for $i \geq 0$, the algorithm decreases $v$ by one until $v$ is 0. Therefore, at the end of the algorithm $v$ is equal to zero. Since the rank of the CFP represented by this list, is equal to the sum of the $N_{i,j}$'s used to decrease $v$ in the while, the lemma follows. □

5. Generation of the catastrophic faults

In this section we describe and analyze an algorithm for the generation of all the catastrophic fault patterns for a linear array with link redundancy $g$. The problem of the generation of the objects of a given set has been widely studied [27]. Our algorithm, as in many algorithms for the systematic generation of a set of objects, has three components: the initialization, the transformation from an object of the set to the next one, and the end condition telling when to stop. In our case, the set of objects is the set of all the catastrophic fault patterns. We want generate the CFPs according to the order established by the rank, i.e., we want to start with the CFP whose rank is 0, and then proceed by generating the CFPs in order of increasing rank. The initialization is the generation of the CFP which has the smallest rank. The transformation from a catastrophic fault pattern $F$, whose rank is $R(F)$, yields the CFP $G$ whose rank is $R(G) = R(F) + 1$. Let $(r_0, r_1, \ldots, r_{g-2})$ be the row representation of a catastrophic fault pattern $F$. We remark that $r_0$ must be 0. The CFP with rank $R(F) + 1$ is obtained by increasing the row index $r_c$, $1 \leq c \leq g-2$, such that $r_c \leq r_{c+1}$ and $r_j > r_{j+1}$, for $1 \leq j < c$, and by setting to 0 the row indexes $r_1, r_2, \ldots, r_{c-1}$. Observe that if such index $r_c$ does not exist, then the CFP has row representation $(0, g-2, g-3, \ldots, 2, 1, 0)$ and it has the biggest rank. Procedure NEXT uses the dummy row index $r_0 = 0$ to detect this situation which constitute the end condition. Procedure GENERATE uses procedure INIT to perform the initialization, and calls procedure NEXT until the end condition is reached, to obtain all the CFPs. Fig. 6 shows the algorithm.

Notice that the procedure NEXT, in order to obtain the next CFP, modifies only a subset of the row indexes $r_1, r_2, \ldots, r_{g-2}$, without rewriting those that remain unchanged. The correctness of procedures INIT and GENERATE is straightforward. Next lemma proves the correctness of NEXT.

**Lemma 9.** Given as input a catastrophic fault pattern $F$, with $R(F) < F^U(g)$, a call to NEXT returns the catastrophic fault pattern $G$ whose rank is $R(G) = R(F) + 1$.

**Proof.** Let $(0, r_1, \ldots, r_{c-1}, r_c, r_{c+1}, \ldots, r_{g-2}, 0)$ be the row representation of $F$. First, observe that the procedure yields always a CFP. Indeed, let $c$ be the value of $j$ at the end of the while. Then $c$ is the smallest index for which $r_c \leq r_{c+1}$. Procedure NEXT increases the row index $r_c$ and, if $c > 1$, it sets to zero the row indexes $r_1, r_2, \ldots, r_{c-1}$. Hence, the obtained row representation satisfies Proposition 3. If $c = 1$ it is easy to see that $R(G) = R(F) + 1$. Assume $c > 1$. Then $(0, 0, \ldots, 0, r_c + 1, r_{c+1}, \ldots, r_{g-2}, 0)$ is the row representation of $G$. Since $R_0(H) = R_{g-1}(H) = 0$ for any CFP $H$ for
a linear array with link redundancy $g$, from (7) we have that
\[
\mathcal{R}(F) = \mathcal{R}_1(F) + \cdots + \mathcal{R}_c(F) + \cdots + \mathcal{R}_{g-2}(F)
\]
\[
= \sum_{r=0}^{r_{c-1}} N_{r,1} + \cdots + \sum_{r=0}^{r_{c-1}} N_{r,c} + \mathcal{R}_{c+1}(F) + \cdots + \mathcal{R}_{g-2}(F),
\]
whereas
\[
\mathcal{R}(G) = \sum_{r=0}^{r_{c}} N_{r,c} + \cdots + \mathcal{R}_{g-2}(G).
\]
Since $\mathcal{R}_k(F) = \mathcal{R}_k(G)$, for $g - 2 > k > c$, from (8) we have that
\[
\mathcal{R}(G) - \mathcal{R}(F) = N_{r,c} \left( \sum_{r=0}^{r_c-1} N_{r,1} + \cdots + \sum_{r=0}^{r_c-1} N_{r,c-1} \right) = N_{0,0} - 1;
\]
hence the lemma. $\square$

Now, we analyze the complexity of \textsc{generate}. The execution of \textsc{generate} requires one call to \textsc{init} and exactly $F^G(g)$ calls to \textsc{next}. The complexity of \textsc{init} is clearly $\Theta(g)$. Procedure \textsc{next} yields the next catastrophic fault pattern by increasing an index $j$ from 1.
up to the first value \( c < g - 1 \) for which \( r_c \leq r_{c+1} \). Hence, a simple upper bound is \( O(g) \).

The next lemma, by using an amortized analysis, characterizes the complexity of \( \text{NEXT} \).

**Lemma 10.** Let \( c \) be an integer, \( 1 \leq c \leq g - 2 \). During the \( F^U(g) \) calls, procedure \( \text{NEXT} \) ends the computation executing \( c \) iterations in the while statement exactly \( M_{2,c-1} \) times.

**Proof.** From Proposition 3, if in a CFP \( r_j > r_{j+1} \) then \( r_j = r_{j+1} + 1 \). Therefore if the algorithm ends the while with \( j = c \), then \( r_{k-1} = r_k + 1 \) for \( k = 2, 3, \ldots, c - 1 \) and \( r_c \leq r_{c+1} \). How many times does this situation occur? This situation occurs whenever the CFP input of \( \text{NEXT} \) has row representation \((0, r_c + c, \ldots, r_c + 1, r_c, \geq r_c, *, \ldots, *)\), where \( \geq r_c \) means an index unspecified but not smaller than \( r_c \) and * means an unspecified row index. For fixed values of \( c \) and \( r_c \), there are exactly \( M_{r_c,c+1} + M_{r_c,c+1} + \cdots + M_{r_c-2,c+1} + M_{r_c,c+1} \) such CFPs. To obtain the number of times in which \( \text{NEXT} \) ends the while with \( j = c \) we have to sum all previous quantities over all possible values of \( r_c \) for which \( M_{r_c,c} \) is not zero, i.e. for \( r_c = 0, 1, \ldots, g - c - 2 \). Hence, \( \text{NEXT} \) exits from the while with \( j = c \) exactly

\[
M_{1,c} + M_{2,c} + \cdots + M_{g-1-c,c} = M_{2,c-1}
\]
times. \( \Box \)

Lemma 10 enables us to estimate the total running time (i.e., the time needed to generate all the CFPs) of \( \text{GENERATE} \). We rewrite the sum \( \sum_{c=1}^{g-2} c M_{2,c-1} \) splitting the terms into \( g - 2 \) rows:

\[
\sum_{c=1}^{g-2} c M_{2,c-1} = M_{2,g-3} + M_{2,g-4} + \cdots + M_{2,2} + M_{2,1} + M_{2,0} + M_{2,g-3} + M_{2,g-4} + \cdots + M_{2,2} + M_{2,1} + \cdots + M_{2,2} + M_{2,1} + M_{2,g-3}
\]

By using (5) and the fact that, for any \( 0 \leq i \leq g - 2, M_{0,i} = M_{1,i} \), we have that for \( 0 \leq k \leq g - 3 \)

\[
M_{1,k} = M_{2,k} + M_{1,k+1} = M_{2,k} + M_{2,k+1} + M_{1,k+2} = M_{2,k} + M_{2,k+1} + \cdots + M_{2,g-3} + M_{1,g-2} = M_{2,k} + M_{2,k+1} + \cdots + M_{2,g-3} + 1.
\]
Thus, each row in the previous expression for \( \sum_{c=1}^{g-2} c \cdot M_{2,c-1} \) is equal to some \( M_{1,k} \) minus 1. In particular, we have that
\[
\sum_{c=1}^{g-2} c \cdot M_{2,c-1} = M_{1,0} - 1 + M_{1,1} - 1 + \cdots + M_{1,g-3} - 1 + M_{1,g-2} - 1.
\]
Notice that we added the term \( M_{1,g-2} - 1 \) which is equal to 0. By Lemma 4 we have that \( M_{1,0} = N_{0,g-1}, M_{1,1} = N_{0,g-2}, M_{1,2} = N_{0,g-3} \) and so on, thus
\[
\sum_{c=1}^{g-2} c \cdot M_{2,c-1} = N_{0,g-1} + N_{0,g-2} + \cdots + N_{0,1} - (g - 1).
\]
From (4), since \( F(k) = \Theta(4^g/g^{3/2}) \), the above expression is clearly \( \Theta(4^g/g^{3/2}) \). Therefore, the algorithm generates all the CFPs in linear time in the number of CFPs (actually the running time is equal to the number of CFPs).

6. Conclusion

In this paper we have analyzed catastrophic (i.e. that prevent reconfiguration) fault patterns for systolic linear arrays with unidirectional bypass links. We proved that the number of such catastrophic sets of faults is equal to the \((g - 1)\)th Catalan number \((1/g)\binom{2g-2}{g-1}\), where \( g \) is the length of the bypass link. The knowledge of the exact number of catastrophic fault patterns is useful to derive reliability measures for linear arrays (e.g. [25]). We also provided algorithms to rank, unrank and generate all the catastrophic fault patterns. This may be useful to generate catastrophic fault patterns at random to test the behavior of reconfiguration strategies, that should be able to recognize the impossibility of reconfiguring the array.

Acknowledgements

We thank L. Pagli for useful discussions and for pointing out the references on the subject. The first author would like to thank A. De Bonis for helpful suggestions. We also thank anonymous referees for useful comments.

References


D.G. Rogers, Pascal triangles, Catalan numbers and renewal arrays, Discrete Math. 22 (1278) 301–310.


