A Two-Phase Reconfiguration Strategy for Extracting Linear Arrays out of Two-Dimensional Architectures

Hussain Al-Asaad and Elias S. Manolakos

Electrical and Computer Engineering Department
Northeastern University, Boston, MA 02115

Abstract

In order to maintain constant interconnection wire lengths between logically adjacent cells and avoid introducing additional tracks of busses and switches when linear arrays are extracted out of two-dimensional architectures with faulty processing elements, the "spiral" reconfiguration approach has been introduced. Its main drawback, relative to the tree and patching approaches, is that it leads to low harvesting. In this paper we introduce a two-phase reconfiguration strategy that drastically increases the harvesting ratio. The algorithm of the first phase achieves comparable harvesting to the previously proposed schemes [5,6], while it is simpler and can be implemented by on-chip logic. The algorithm of the second phase may complement any other scheme used during the first phase, and raises the harvesting ratio to levels that could be achieved by the much more involved tree approach.

1 Introduction–Motivation

Linear arrays are used in signal/image processing, associative string processing, iterative cellular arithmetic, serial memories, etc. Due to their wide use many researchers have investigated the design of fault tolerant linear arrays. A possible solution is offered by the *Diogenes* array (DIOG) [1,2], where a faulty Processing Element (PE) is simply bypassed. Although it achieves 100% utilization of good PEs (perfect *harvesting*) the physical distance between two logical neighbors depends on the fault distribution and failures in the supporting logic (switches and interconnections) cannot be tolerated.

In order to obtain probabilistic bounds on the maximum wire length in the physical two-dimensional architecture and also cope with interconnection failures, additional tracks of busses and switches can be inserted between PE rows and columns. The *patching* and the *tree* approaches have been developed along these lines (see [3] for a comprehensive treatment and references). Perfect harvesting is possible using patching, but only if some long wires are acceptable. On the other hand, the tree approach leads to time consuming reconfiguration and is more suitable for off-line end-of-production restructuring.

The *spiral* approach was introduced to keep *constant* the length of wires between logical neighbors in the array and eliminate the need for additional tracks of busses and switches. Using this approach a good cell is first selected (the *head* of the spiral), either at the border or towards the center of the physical 2-D architecture. The spiral then gets extended using an algorithm that explores the fault-free neighbors of the current head. Several published proposals fall into this category. Koren's algorithm [4] is very simple but leads to usually unacceptable harvesting. Aubusson and Catt's algorithm [5] is still

0-8186-3502-9/93 $3.00

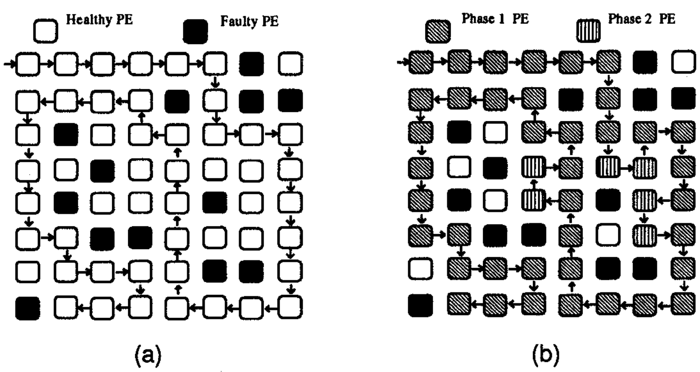

Figure 1: (a) The longest linear array extracted by Algorithm 1. (b) The longest linear array after applying Algorithm 2; the PEs added to the spiral during phase 2 are shaded differently.

simple and achieves better harvesting. Manning's algorithm [6], which has "memory" in the sense that it takes into account the current position of the head relatively to the borders of the physical array, achieves the best harvesting ratio.

In this paper we introduce an alternative two-phase spiral approach reconfiguration strategy for extracting maximum length linear arrays out of two-dimensional VLSI or WSI array architectures with faults. Both phases are simple (memoryless) and can be implemented efficiently using on-chip logic. The second phase algorithm can complement any one of the previously reported schemes in order to boost the harvesting ratio.

2 The New Reconfiguration Strategy

The logical array is linear and the physical array is two-dimensional. Every PE of the physical array is connected to four near neighbors via four communication ports facing up (north), right (east), down (south) and left (west). We will denote these directions by $D[1]$, $D[2]$, $D[3]$ and $D[4]$ respectively. Faults can occur in the PEs as well as in the communication ports and their locations can be either randomly distributed or clustered. A faulty PE cannot serve as a connecting element between two good PEs.

Our reconfiguration strategy is two-phase. The objective of the first phase is to find the longest possible linear array using a simple algorithm that can be implemented using on-chip logic. The second phase will attempt to further improve the harvesting by reclaiming good PEs left unused by the first phase.

2.1 The first phase

A border cell is chosen as the spiral *head*; let us call it $C(1)$. Assuming that a spiral of length h has been built let us call $C(h)$ the cell at the current head, using the notation in [3].

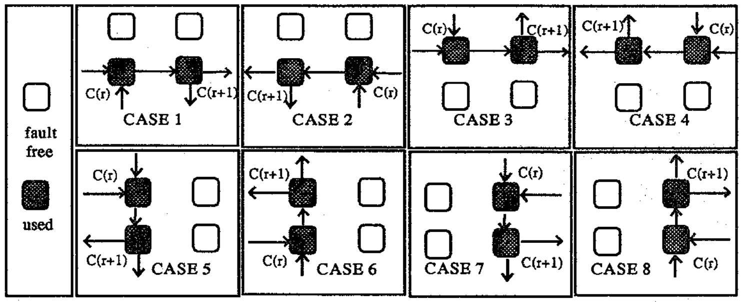

Figure 2: The cases that can arise during phase 2

ALGORITHM 1

1. Let $x = 1$
2. Check the cell in direction $D[x]$ relatively to $C(h)$; if fault-free and not used go to step 5.
3. Increment x; if $x < 5$ go to step 2.
4. Record length of the spiral; back track to $C(h-1)$. If $C(h-1) = C(1)$ go to step 6, else go to step 1.
5. Add the cell to the spiral as a new head. Record that the cell is used, go to step 1.
6. Select the longest linear array.

As described above Algorithm 1 can be implemented under host control. However it is not difficult to implement it under distributed local control, using simple on-chip logic as well (due to space limitations the distributed implementation will not be discussed here). In an attempt to extend the spiral the current head $C(h)$ checks its neighbors in the following *fixed* order: up, right, down, and left. This order of searching for good neighbors is *independent* of the $C(h-1) \rightarrow C(h)$ segment orientation (as opposed to the algorithms in [4,5]) and of the head's position in the physical array (as opposed to the algorithm in [6]). Algorithm 1 has comparable complexity with Aubusson-Catt's algorithm [5] and achieves harvesting comparable to the more involved algorithm by Manning [6]. Details on the results of our comparative evaluation study based on extensive Monte Carlo simulations will be discussed in section 3.

2.2 The second phase

Several good PEs are not included in the longest linear array of length L built during the first phase. However some of them can be easily inserted during phase 2 by the following simple algorithm. Let $C(r)$ and $C(r+1)$ be two neighboring PEs in the spiral and $C(r).[x]$ be the neighbor $C(r)$ in the direction $D[x]$, $x \in \{1,2,3,4\}$. Also let $C(r).[x] = 0$ denote that this neighbor is fault-free and has not been included in the spiral so far.

<u>ALGORITHM 2</u>

```
r = 1;
Repeat
        x := 1;
        found := false;
        Repeat
                If ((C(r).[x] = 0) AND (C(r + 1).[x] = 0))
                  then          {if both PEs are fault-free and not used}
                        found:=true
                  else
                        Increment(x);
        Until{ found or (x=5)};
        If (found) then         { prolong the spiral }
                Begin
                  C'(1 : r) := C(1 : r);           { C' is a temporary list }
                  C'(r + 1) := C(r).[x];           { insert two good PEs }
                  C'(r + 2) := C(r + 1).[x];
                  C'(r + 3 : L + 2) := C(r + 1 : L);
                  C(1 : L + 2) := C'(1 : L + 2)
                  L := L + 2;
                End
        else Increment(r);
Until {r=L};
```

To better understand the reconfiguration strategy consider the 8x8 physical array shown in Figure 1(a), where 13 PEs are assumed to be faulty (yield 79.7%). After using Algorithm 1, a linear array of length $L = 39$ is extracted corresponding to a harvesting ratio of 60.9%. After applying also Algorithm 2, the linear array of Figure 1(b) is obtained where $L = 45$ (70.3% harvesting). Notice that although Algorithm 1 requires backtracking in order to find the longest linear array (also true for the schemes in [4,5,6]), Algorithm 2 will traverse only once the array generated by Algorithm 1. As it will be shown in the next section phase 2 may raise the harvesting ratio of phase 1 by as much as 40%, depending on the fault distribution and the yield.

Let $C(r)$ and $C(r+1)$ be two consecutive PEs in the spiral. When $C(r)$ is reached during phase 2 the configurations shown in Figure 2 can be encountered, where each case represents four different configurations. For example, in CASE 1, $C(r-1)$ can be west or south of $C(r)$ and $C(r+2)$ can be east or south of $C(r+1)$. It is clear that checking for pairs of good neighbors is required towards at most two directions, depending on the orientation (horizontal vs. vertical) of the segment $C(r) \rightarrow C(r+1)$. By extensive Monte Carlo simulations on arrays of size 20×20 we have found the following probabilities of occurrence for each case.

Horizontal $C(r) \rightarrow C(r+1)$	Probability	Vertical $C(r) \rightarrow C(r+1)$	Probability
CASE 1	0.0100	CASE 5	0.0001
CASE 2	0.0100	CASE 6	0.0942
CASE 3	**0.3808**	**CASE 7**	**0.4074**
CASE 4	0.0092	CASE 8	0.0883

It is observed that:

$$\Pr(\text{CASE3}) \gg \Pr(\text{CASE1}) \qquad \Pr(\text{CASE2}) \approx \Pr(\text{CASE4}) \tag{1}$$

$$\Pr(\text{CASE7}) \gg \Pr(\text{CASE5}) \qquad \Pr(\text{CASE6}) \approx \Pr(\text{CASE8}) \tag{2}$$

Hence, if the segment $C(r) \rightarrow C(r+1)$ is horizontal (vertical), it is advantageous to check first downwards (leftwards).

3 Simulation Results–Discussion

Extensive Monte Carlo simulations have been performed in order to compare the new two-phase strategy with the spiral approach techniques reported in the literature by Koren (KOR) [4], Aubusson-Catt (AUB) [5], and Manning (MAN) [6]. A 20x20 physical array was used in all simulations. For each percentage value of fault free PEs (yield) examined, 500 different fault maps have been generated randomly. The head of the spiral was placed at the upper left corner PE in every simulation run. If this PE is faulty its right neighbor is used as head and so on. The *mean* length of the longest array extracted by phase 1, normalized by the number of PEs, is plotted in Figure 3(a). It is observed that Algorithm 1 outperforms KOR and AUB (in most cases) and gives comparable average harvesting to MAN, which is a more complicated algorithm.

By running phase 2, in conjunction with *all* the above mentioned algorithms for phase 1, the harvesting is raised substantially as shown in Figure 3(b), where we plot the average percentage of harvesting improvement vs. the yield. As we can see from Figure 3(c) all three algorithms (except KOR) *achieve approximately the same harvesting across a wide range of yield values* after phase 2. This *saturation* of performance is a strong indication that our two-phase reconfiguration strategy gets very close to the maximum harvesting that can be achieved by *any* spiral approach technique. Since spiral approach schemes trade harvesting for interconnection complexity and implementation simplicity, our two-phase strategy offers: (i) a simpler first phase algorithm compared to AUB and MAN and (ii) an overall near optimal harvesting ratio. Also note that applying the proposed two-phase reconfiguration strategy leads to better harvesting relative to any other algorithm used alone, as shown by Figure 3(d).

The same kind of comparative evaluation was performed assuming a *clustered* fault distribution. In every simulation run the position of the center and the radius of each cluster have been generated randomly. The setup was the same as before (500 runs/yield value, on 20×20 physical arrays) The average harvesting achieved during phase 1 is shown in Figure 4(a). After applying phase 2, the harvesting is substantially improved as shown in Figures 4(b) and 4(c). By comparing the simulation results for random and clustered faults for the same yield value, we observe that longer linear arrays are extracted on the

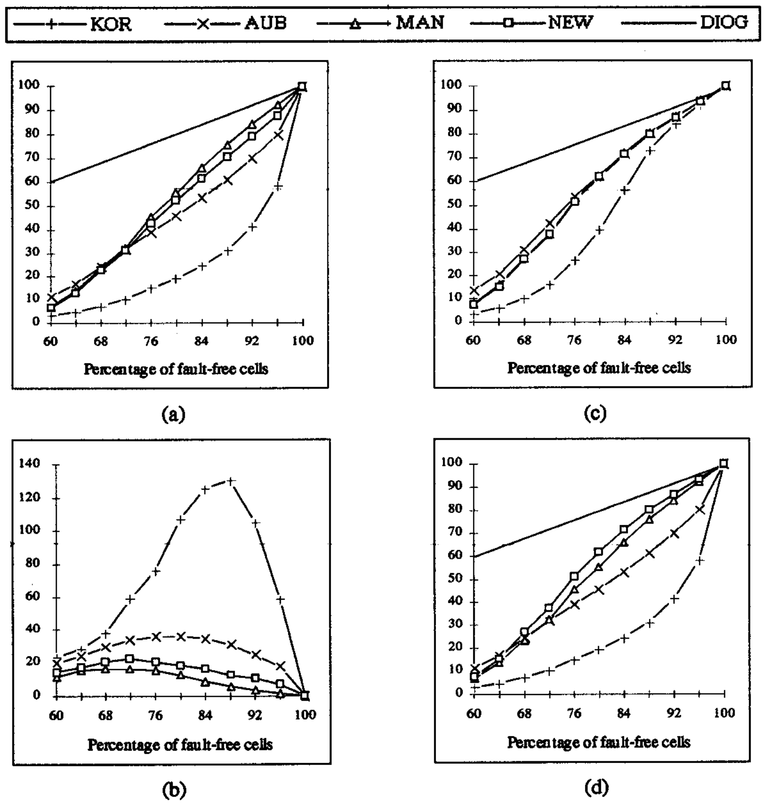

Figure 3: Simulation results on a 20 × 20 physical arrays with random faults (500 runs/yield). (a) Phase 1 harvesting (mean length of the longest linear array normalized by the number of PEs vs. the yield); (b) the percentage of harvesting improvement after applying phase 2 to all algorithms; (c) the harvesting after applying phase 2 in conjunction with all other algorithms used in phase 1; (d) the harvesting achieved by the proposed two-phase strategy relative to the other algorithms used alone.

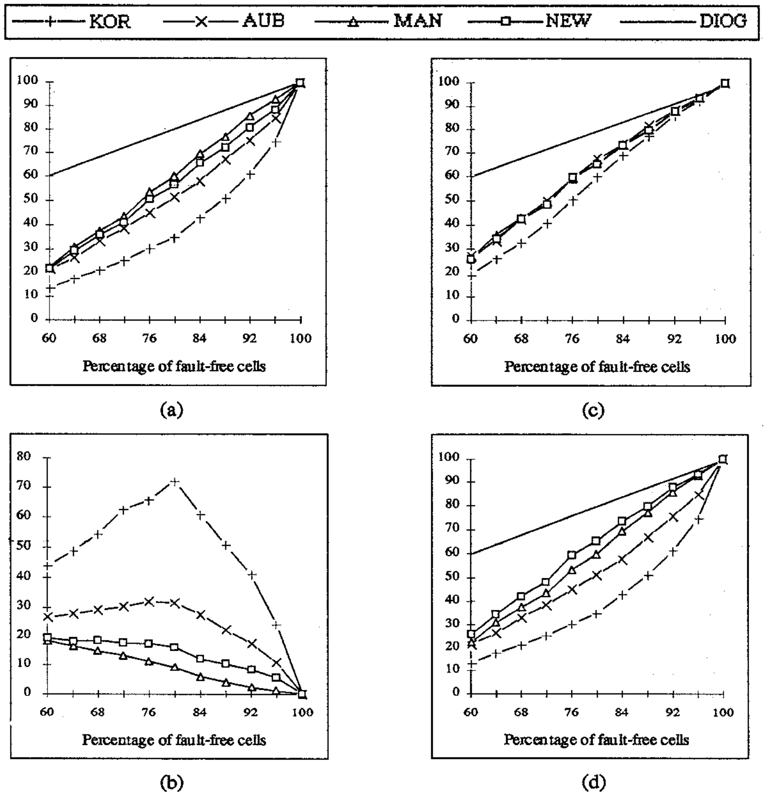

Figure 4: Simulation results on a 20 × 20 physical arrays with clustered faults (500 runs/yield). (a) Phase 1 harvesting (mean length of the longest linear array normalized by the number of PEs vs. the yield); (b) the percentage of harvesting improvement after applying phase 2 to all algorithms; (c) the harvesting after applying phase 2 in conjunction with all other algorithms used in phase 1; (d) the harvesting achieved by the proposed two-phase strategy relative to the other algorithms used alone.

average in the latter case. This is so because the probability that the head of the spiral will be initially poised at an isolated set with a small number of good cells is lower when the faults are clustered.

4 Conclusions

A new two-phase reconfiguration strategy following the spiral approach for extracting maximum length linear arrays out of two-dimensional physical arrays with faults has been introduced. If only the first phase is used (Algorithm 1) the harvesting is in most cases higher than that obtained by Aubusson-Catt's algorithm and comparable to Manning's algorithm, even though Algorithm 1 is much simpler due to the fixed order memoryless exploration for good neighbors of the current spiral head. Our most important contribution is the introduction of a second phase (Algorithm 2) that can complement any other algorithm used during the first phase and boost the harvesting ratio to levels that are comparable to those achieved by the much more complicated tree approach (the interested reader could compare our figures with those of the study reported in [3], pp. 119). Our two-phase strategy can also be implemented under locally distributed control using a moderate amount of additional on-chip logic. This part has not been included here due to space limitations. We are currently investigating extensions that will allow *both* the head and the tail of the spiral to be placed at the borders of the 2D physical architecture, to facilitate input and output to the linear array.

References

[1] A. Rosenberg, "The Diogenes approach to Testable Fault-tolerant Arrays of Processors", *IEEE Transactions on Computers*, C-32(10):902-910, 1983.

[2] K. Belkhale, P. Banerjee, "Reconfiguration Strategies for VLSI Processor Arrays and Trees Using a Modified Diogenes Approach", *IEEE Transactions on Computers*, C-41(1):83-96, 1992.

[3] R. Negrini, M. Sami, R. Stefanelli, *Fault Tolerance Through Reconfiguration in VLSI and WSI Arrays*, MIT press, 1989.

[4] I. Koren, "A reconfigurable and Fault Tolerant VLSI multiprocessor Array", in the Proc. of the *8th IEEE Computer Architecture Symposium* pp. 426-441, May 1981.

[5] R. Aubusson, I. Catt, "Wafer-Scale Integration: a Fault Tolerant Procedure", *IEEE Journal of Solid-State Circuits*, SC-13(3)339-344, 1978.

[6] F. Manning, "An Approach to Highly Integrated, Computer-Maintained Cellular Arrays", *IEEE Transactions on Computers*, C-26(6)536-552, 1977.