

HEX-REPAIR: A NEW APPROACH TO HEXAGONAL ARRAY RECONFIGURATION

R. Venkateswaran P. Mazumder Kang G. Shin
The University of Michigan
Ann Arbor, Michigan 48109

Abstract

The goal of this paper is to present a new approach to the reconfiguration problem for regular hexagonal arrays that find numerous applications in VLSI and WSI designs such as multipliers, signal processing circuits, etc. Efficient heuristics are used to obtain a fast solution that has excellent fault-coverage characteristics even in the presence of multiple faults.

Introduction

Fault reconfiguration for regular processor-arrays is an important problem since it is often the only solution capable of giving acceptable production yields. In order to be viable, however, such a solution should have a low overhead in terms of extra switching circuitry, be able to route communication paths between non-faulty cells so as to maintain *transparency* and at the same time minimize the path-length so as to suffer the least possible performance degradation. The topology of the processor array also has a considerable impact on the reconfiguration strategy. Much of the previous work in array reconfiguration [1, 2, 3, 4] have dealt with rectangular or square arrays and do not work well for hexagonal arrays. In [5], a reconfiguration scheme is presented for hexagonal arrays. The drawback is that it is intended when only one or two faults occur at a time.

In this paper, we suggest a new reconfiguration algorithm called HEX-REPAIR which is more robust in the presence of multiple faults and has fault coverage rates comparable to index-mapped schemes proposed for square arrays. HEX-REPAIR can be divided into two phases: (a) *Covering Phase*: wherein all faults are covered by either horizontal or vertical fault-cover lines, and (b) *Configuration Phase*: wherein the switching devices are appropriately set to ensure correct operation of the array.

Preliminaries

Each cell of the array is represented by a pair of *physical* indices (p_x, p_y) , and a pair of *logical* indices (l_x, l_y) , that indicate the indices of the function of each cell at run-time. The two are the same in the absence of faults, but can differ when faults occur. We let \mathcal{F} denote the fault set comprising of the physical indices (X_i, Y_i) of all the faulty elements in the array. We assume the presence of R spare rows and C spare columns which are available to replace the faulty cells.

Two cells are said to be *horizontally* connected if they lie on the same physical row. Two cells are said to be *row (column)* connected if they are either horizontally (vertically) or diagonally connected. An *H-line* is any collection of n' row-connected PEs, where n' is the number of columns in the physical grid. Thus, each H-line starts at column 1 and ends at the last column. Definitions for vertically connected cells and V-lines proceed analogously.

The H- and V-lines are cover-lines, i.e. they determine the cells that are to be bypassed in the final solution. These bypassed cells will be referred to as *switching-elements* (SE) to indicate the fact that they cease to perform processing per se, but rather behave like switches between pairs of neighboring cells. A lookup table is used to configure the SEs so as to maintain proper interconnections. Fig. 1 shows the solution for an 8×8 physical grid

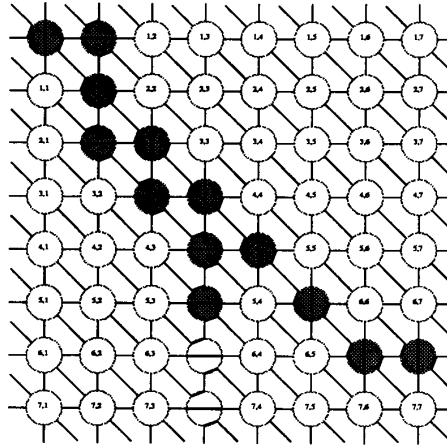


Figure 1: Example of a reconfiguration problem

with 13 faulty processing elements using just one spare row and one spare column of cells.

Details of HEX-REPAIR

The *covering* phase comprises of the first four steps while the latter two constitute the *configuration* phase.

(i) **Fault Mapping:** This step assigns a number between 1 and $|\mathcal{F}|$ (the number of faults) to each fault in the array based on scanning the cells from right to left, in successive diagonal sets D_j , which consists of all cells with physical indices (x, y) such that $j = x + y - 1$.

(ii) **Horizontal and Vertical Cover Graphs:** The Horizontal and Vertical Cover Graphs aid in identifying the maximal sets of faults, such that all the faults in one set can be simultaneously covered with a single spare row (HCG) or column (VCG). They impose a partial order on the set \mathcal{F} thereby limiting the search space considerably.

The Horizontal Cover Graph (HCG) for a given \mathcal{F} consists of $|\mathcal{F}| + 1$ nodes, one for each fault and a special *sink* node. There exists a directed edge from node i to node j provided i is not already connected to a node that is an ancestor of j and also provided either $(Y_i > Y_j, X_i \geq X_j, Y_i - Y_j \geq X_i - X_j)$, or j is the *sink* node. The Vertical Cover Graph (VCG) is defined similarly by interchanging the X and the Y . The HCG and VCG, as defined above, can be constructed in $O(|\mathcal{F}|^2)$ time.

(iii) **Graph Traversal:** Once the graphs are constructed, we determine a collection of paths S_h (S_v) in the HCG (VCG respectively) starting from a *root* node, *i.e.* one with no ancestors, and terminating at the *sink* node. New paths are generated using a depth-first search strategy.

(iv) **Finding the Solution Set \mathcal{S} :** This consists of solving an integer program for choosing n_h paths from S_h and n_v paths from S_v , which cover all the faults subject to spare

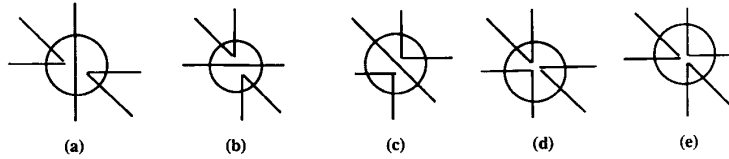


Figure 2: SE configurations.

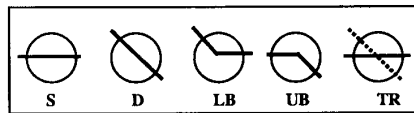


Figure 3a: H-line types

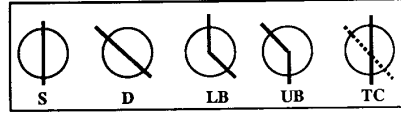


Figure 3b: V-line types

availability. The formulation is as follows:

Find a set of values for m integer variables x_1, x_2, \dots, x_m and the n integer variables y_1, y_2, \dots, y_n , which minimize the objective function $R \cdot \sum_{i=1}^m x_i + C \cdot \sum_{i=1}^n y_i$ subject to the $|\mathcal{F}| + 2$ constraints of the form

$$\sum_{i=1}^m a_{i,j} x_i + \sum_{i=1}^n a_{(i+m),j} y_i \geq 1 \quad j = 1, 2, \dots, |\mathcal{F}| ; \quad \sum_{i=1}^m x_i \leq R ; \quad \sum_{i=1}^n y_i \leq C$$

Here, $a_{i,j} = 1$ if fault j is in path i and 0 otherwise. The solution set \mathcal{S} includes all paths that correspond to a non-zero x_i or y_i value at the end of the optimization.

(v) **Configuring the Switching Elements:** For each horizontal(vertical) path p , a H/V line is constructed as follows: "Let p consist of P faults $\{f_1, f_2, \dots, f_P\}$ arranged in decreasing order of their fault-indices. The fault f_1 is horizontally (vertically) connected to the PE in the first column (row) of the row(column) of f_1 . Similarly, f_P is horizontally (vertically) connected to the PE in the last column(row) of the row(column) of f_P . For any other two faults f_i and f_{i+1} , we find a (unique) PE k which can be horizontally (vertically) connected to f_i and diagonally connected to f_{i+1} ."

In the presence of faults occurring in more than one line, this procedure can lead to some overlap between H/V lines which is eliminated by removing the common faults from all but one line and reconstructing the other lines as before. There are 6 H-line and V-line types for each SE (see Fig. 2). Table 1 is the SE configuration table. Entries marked with a '-' indicate invalid H-type/V-type combinations. The 5 possible SE configurations, "a", "b", "c", "d", and "e" are shown in Fig. 3. The 5 SE configurations can be easily implemented in hardware by a collection of 15 ON/OFF devices per processor, each of which can be as basic as a pass gate. No extra data paths are needed. Thus the additional area overheads are quite minimal.

(vi) **Assignment of Logical Indices:** All SEs are assigned logical indices $(0, 0)$. The logical row index of a working PE is its physical row index less the number of H lines that traverse the column above the physical row under consideration. The logical column index is similarly defined. If two non-faulty cells (p_{1x}, p_{1y}) and (p_{2x}, p_{2y}) get assigned the same logical indices, then the first one is treated as a SE of type a if $|p_{2x} - p_{1x}| \geq |p_{2y} - p_{1y}|$, or as type b if otherwise.

Table 1: Lookup table for SE settings

H-type → V-type ↓	U	S	D	LB	UB	TR
U	-	a	a	a	a	a
S	b	c	a	e	d	c
D	b	b	-	-	-	-
LB	b	e	-	e	-	-
UB	b	d	-	-	d	-
TC	b	c	-	-	-	-

Conclusions

An efficient reconfiguration algorithm for hexagonal arrays has been presented here. The individual processors are assumed to be simple in nature, consequently the approach uses simple switches only for rerouting. It can be proved that the maximum distance between two logical neighbors after rerouting is $\max(2R + C + 1, 2C + R + 1)$. It can also be analytically shown that the algorithm guarantees a solution when either all the faults can be trivially covered, one by each spare row or column, or when the probability of failure of an individual processing element in an N -cell array $\leq 1/N$.

Acknowledgments

This research was partially supported by the ONR grant number 00014-85-K-0122, Digital Equipment Corp faculty development grant, University of Michigan faculty grant, and the Research Initiation Program of the National Science Foundation under the grant number MIP-8808978.

References

- [1] F. Lombardi, M. G. Sami, and R. Stefanelli, "Reconfiguration of VLSI arrays by covering," *IEEE Transactions on Computer-Aided Design*, vol. 8, no. 9, pp. 952-965, September 1989.
- [2] F. Lombardi, D. Sciuto, and R. Stefanelli, "A technique for reconfiguring two-dimensional VLSI arrays," in *Real Time Symposium*, pp. 44-53, 1987.
- [3] R. Negrini, M. Sami, and R. Stefanelli, "Fault tolerance techniques for array structures used in supercomputing," *Computer*, pp. 78-87, 1986.
- [4] M. Sami and R. Stefanelli, "Reconfigurable architectures for VLSI processing arrays," in *Proceedings of the IEEE*, volume 74, pp. 712-722, May 1986.
- [5] D. Gordon, I. Koren, and G. M. Silberman, "Restructuring hexagonal arrays of processors in the presence of faults," *Journal of VLSI and Computer Systems*, pp. 23-35, 1987.