

# Reconfiguration in 3D Meshes

Anuj Chandra                      and  
Department of Electrical Engineering  
University of Pittsburgh  
Pittsburgh, PA 15261

Rami Melhem  
Department of Computer Science  
University of Pittsburgh  
Pittsburgh, PA 15260

## Abstract

*The  $1\frac{1}{2}$  track model for fault tolerant 2D processor arrays is extended to 3D mesh architectures. Non-intersecting, continuous, straight and non-near miss compensation paths are considered. It is shown that when six directions in the 3D mesh are allowed for compensation paths, then switches with 13 states are needed to preserve the 3D mesh topology after faults. It is also shown that switch reconfiguration after faults is local in the sense that the state of each switch is uniquely determined by the state of the 2 processors connected to it.*

## 1: Introduction

3D mesh architectures offer a greater degree of interconnection as compared to 2D meshes. In addition, for practical size arrays, the diameter of a 3D mesh is smaller than the diameter of 2D meshes since, the diameter of a 2D array with  $N$  processors is  $2N^{\frac{1}{2}}$  and the diameter of a 3D mesh with  $N$  processors is  $3N^{\frac{1}{3}}$ . In addition the 3D mesh has a larger bisection width than the 2D array. Moreover, due to the rich connectivity that the 3D mesh offers, it seems to be ideal for number crunching and image processing tasks like 3D Magnetic Resonance image processing and 3D connected component labeling [5]. Cray's recent massively parallel processing machine, CRAY T3D, has a 3D torus interconnect [4]. It is therefore desirable to extend 2D fault tolerance techniques into 3D.

Several techniques for tolerating faulty processing elements (PE's) in 2D by replacing them with spare PE's have been studied [6, 7, 8, 9, 11, 12, 13]. This paper extends the  $1\frac{1}{2}$  track model used in 2D [9, 14] to 3D and studies reconfigurability in the 3D  $1\frac{1}{2}$  track model.

## 2: A $1\frac{1}{2}$ track, 3D mesh model

The 3D mesh architecture can be considered as layers (in the  $xy$  plane) of 2D meshes stacked in the  $z$  direction. A  $3 \times 3 \times 3$  mesh is shown in Figure 1. Each PE is connected to its six nearest neighbors. The six neighbors are in the North (N), South (S), East (E), West (W),  $Z_+$  and  $Z_-$  directions.

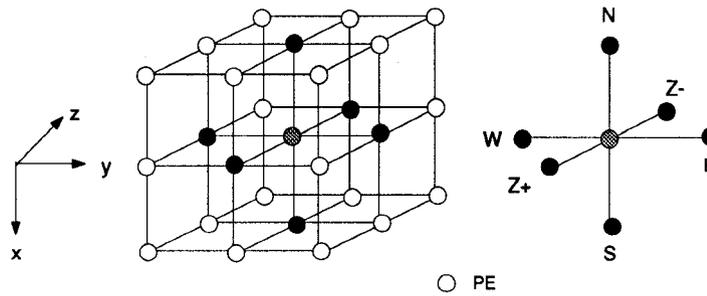


Figure 1: A  $3 \times 3 \times 3$  mesh. The six neighbors of the grey PE are shown in black.

To augment a 3D mesh into the  $3D\ 1\frac{1}{2}$  track model, a six port switch is inserted between a PE and each of its six neighbors. The entire 3D array is covered in each of the six sides by a  $2D\ n \times n$  layer of spare PE's. Figure 2(a) shows a  $2 \times 2 \times 2$  mesh with the corresponding switches in the  $1\frac{1}{2}$  track model. Three types of switches are shown depending on where

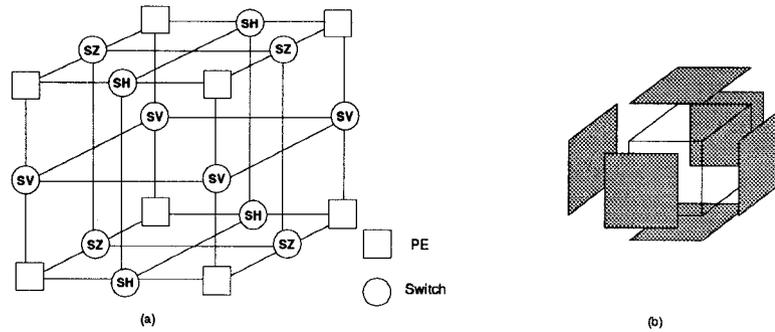


Figure 2: The  $1\frac{1}{2}$  track model in 3D.

they are located: an SV switch is located between two PE's that are N and S neighbors, an SH switch is located between two PE's that are E and W neighbors and an SZ switch is located between two PE's that are neighbors in the Z direction. In addition to being connected to the two PE's that a switch lies between, an SV switch is connected to four other SV switches. Namely, its four SV neighboring switches in the  $yz$  plane. Similarly, an SZ switch is connected to 4 other SZ switches and an SH switch is connected to 4 other SH switches.

Given an  $n \times n \times n$  3D mesh, six  $2D\ n \times n$  meshes of spares are added such that one is added to each side (face) of the 3D mesh. In this model which is shown in Figure 2(b),  $6n^2$  spares are added to the  $n^3$  non-spare PE's.

The connection in a switch of port  $i$  to port  $j$  will be denoted by  $C\langle i, j \rangle$  where  $C$  is a connection operator,  $\{i, j\} \in \{N, S, E, W, Z_+, Z_-\}$  and  $i \neq j$ . Obviously the operator  $C$  is symmetric, that is  $C\langle i, j \rangle \Rightarrow C\langle j, i \rangle$ . The state of a switch will be denoted by the set of ports that are connected together. For example the state in which N and W are connected and E and S are connected will be denoted by  $\{C\langle N, W \rangle, C\langle E, S \rangle\}$ .

We assume that a faulty PE shorts its north and south ports, its east and west ports, and its  $Z_+$  and  $Z_-$  ports. For full reconfigurability, a switch should be able to connect any two of its ports. In general, for a switch with  $n$  ports, the number of states connecting every port to a different port are  $(n-1)(n-3)(n-5)\dots(n-(n-1))$  or  $\prod_{i \text{ odd}}^{n-1} (n-i)$ . This implies that for a reconfigurable switch with  $n=6$  there will be 15 states. Figure 3 shows the 15 states labeled A, B, ..., O.

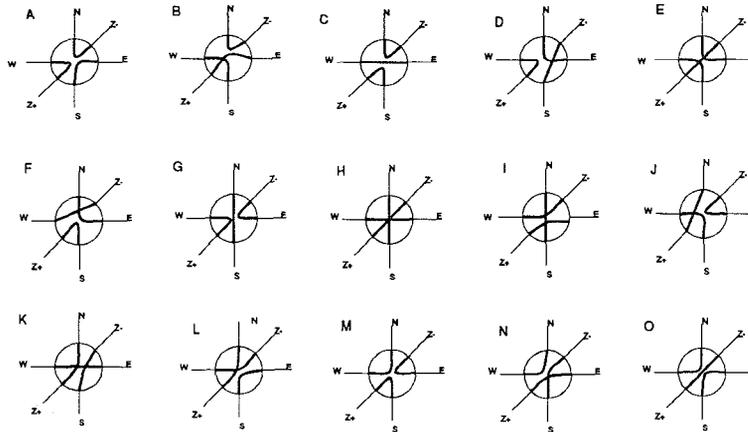


Figure 3: Switch states A through O.

### 3: Fault Tolerance in $1\frac{1}{2}$ track 3D meshes

When a fault occurs in the 3D mesh, the faulty PE is replaced by one of its six neighboring PE's, which in turn is replaced by one of its six neighboring PE's. This successive replacement of PE's continues until a spare PE is reached. The path from the faulty PE to the spare PE is called the compensation path. Like in 2D [9], we will consider compensation paths in the 3D mesh that are straight, continuous, non-intersecting, and non-overlapping. Straight compensation path imply that a faulty PE can only be replaced by PE's that have *direct* links to it and that the path does not change direction. This restricts the compensation path types to at most 6. By continuous we mean that the compensation path terminates at the spare layer surrounding the 3D mesh. Non-intersecting paths implies that no two or more paths in different directions can pass through the same PE. Paths are overlapping if they are in the same or opposite directions and are passing through the

same PE. Non-overlapping does not allow this. The non-intersecting and non-overlapping conditions together imply that only a *single* compensation path can pass through any PE.

In order to generalize the requirement that compensation paths are not in near-miss, we recall that with  $1\frac{1}{2}$  tracks, near-miss occurs in 2D when the projection of one path on the adjacent row (column) overlaps with a path in the opposite directions such that there is more than one PE in common. In 3D, near miss paths can be understood as a combination of 2D near-miss paths in two different slices. To check if a compensation path passing through a PE is in near-miss, the path through this PE is 'projected' onto 4 rows (columns) to check for near miss. For example a PE that has a East or West path (i.e. a horizontal path) passing through it, will check its adjacent rows to the north and south in its  $xy$  slice, and two adjacent rows in the  $z$  direction in its  $yz$  slice. Similarly, adjacent columns in the  $xy$  and  $xz$  plane have to be checked for near miss in vertical compensation paths (paths along the  $x$  direction) and adjacent rows in the  $xz$  and  $yz$  planes need to be checked for near miss in compensation paths in the  $Z$  direction. To clarify the above kinds of compensation paths in 3D, Figure 4 illustrates specific examples.

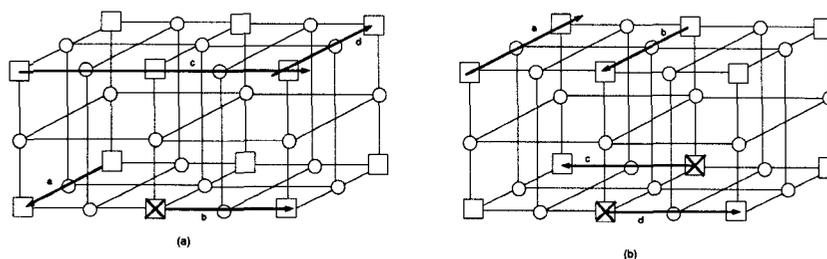


Figure 4: (a) Paths c and d are intersecting. Paths a and b are non-intersecting. (b) Paths a and b are in near-miss. Paths c and d are not in near-miss.

The following notation will be used to refer to a particular kind of compensation path passing through a particular PE: NC denotes that no compensation path passes through the PE. A compensation path passing through a PE will be denoted by the direction of the path. When referring to the state of a PE, we will append to its faulty/healthy state, the compensation path, if any, that passes through it. For example the state of a PE that is not faulty and has no compensation path passing through it will be denoted by NF-NC. Table 1 defines the thirteen possible states of a PE.

We now define the logical array as the desired mesh topology. This is denoted by  $\mathcal{L} = \{\mathcal{L}(i, j, k) \mid i, j, k = 0, \dots, n\}$ . The physical array is the array which is available to us. It may or may not contain faulty PE's. The physical array is denoted by  $\mathcal{P} = \{\mathcal{P}(i, j, k) \mid i, j, k = 0, \dots, n+1\}$ . Fault tolerance is accomplished by mapping each  $\mathcal{L}(i, j, k)$  either to  $\mathcal{P}(i, j, k)$  (if  $\mathcal{P}(i, j, k)$  is NF-NC), to  $\mathcal{P}(i \pm 1, j, k)$ , to  $\mathcal{P}(i, j \pm 1, k)$ , or to  $\mathcal{P}(i, j, k \pm 1)$ . In order to preserve the connectivity among the logical nodes in the 3D mesh, switches have to be set appropriately. The following two theorems will help in determining the states of the switches.

**Theorem 1** *In the  $1\frac{1}{2}$  track 3D model, only SV switches are used to establish vertical connectivity, only SH switches are used to establish horizontal connectivity and only SZ switches are used to establish Z-direction connectivity.*

Table 1: Possible states of a PE

Index	Notation	Fault State	Compensation path through PE
1	NF-NC	Not Faulty	no compensation paths
2	NF-E	Not Faulty	East Compensation path
3	F-E	Faulty	East Compensation path
4	NF-W	Not Faulty	West Compensation path
5	F-W	Faulty	West Compensation path
6	NF-N	Not Faulty	North Compensation path
7	F-N	Faulty	North Compensation path
8	NF-S	Not Faulty	South Compensation path
9	F-S	Faulty	South Compensation path
10	NF-Z <sub>-</sub>	Not Faulty	Z <sub>-</sub> Compensation path
11	F-Z <sub>-</sub>	Faulty	Z <sub>-</sub> Compensation path
12	NF-Z <sub>+</sub>	Not Faulty	Z <sub>+</sub> Compensation path
13	F-Z <sub>+</sub>	Faulty	Z <sub>+</sub> Compensation path

**Proof:** Suppose we want to determine the state of a vertical switch, call it  $SV_O$ , that lies between two PE's,  $\mathcal{P}(i, j, k)$  and  $\mathcal{P}(i+1, j, k)$ . For vertical connectivity, the south port of  $\mathcal{L}(i, j, k)$  should be connected to the north port of  $\mathcal{L}(i+1, j, k)$ . But the south port of any processor is connected only to a SV switch and so is the north port of any processor. Moreover by the construction of the network of switches, it is observed that SV switches are *only* connected to other SV switches. The only way to change directions, i.e. try using a SH or a SZ switch would be if a SV switch is connected to a faulty PE. But as stated earlier, a faulty PE shorts its north and south ports, which would connect the SV via the faulty PE to another SV. There is *no* way to change directions and to use a SH or a SZ switch to maintain vertical neighbors.

Hence only SV switches will be in the routing path that maintains vertical connectivity between PE's. Similarly, the theorem can be proved for SZ and SH switches. •

**Corollary 1** *Switch routing for any particular kind of switches is independent of the routing for the other two kinds of switches.*

**Proof:** Follows directly from Theorem 1. •

**Theorem 2** *For preserving the 3D mesh topology, the state of each switch is determined only by the states of the two PE's connected to it.*

**Proof:** Because of Theorem 1 we need to only prove Theorem 3 for any one kind of a switches. Let us consider  $SV_O$ . Let the PE north of  $SV_O$  be  $\mathcal{P}(i, j, k)$  and the PE south of  $SV_O$  be  $\mathcal{P}(i+1, j, k)$ . If both  $\mathcal{P}(i, j, k)$  and  $\mathcal{P}(i+1, j, k)$  are NF-NC then  $SV_O$  is obviously in state **H**. If a compensation path passes through either  $\mathcal{P}(i, j, k)$  or through  $\mathcal{P}(i+1, j, k)$  then that PE is being replaced by another PE in the direction of the path. This direction cannot change since only non-intersecting compensation paths are being considered. Hence we do not require any information other than the states of the two PE's connected to the switch since this information, namely, the direction in which the replacement PE lies in, cannot

change. Therefore all the information needed to determine the state of the switch lies locally in the two PE's. Looking globally will not provide any additional useful information. •

The local information, that is the states of the two PE's that lie on either side of a switch, can be used to index a look up table that stores the switch states. For SV switches, a table, call it SVT, may be used to store switch states corresponding to all possible combination of states for the two PE's connected to the switch. Similary for SH there will be an SHT and for SZ there will be an SZT. Table 2 shows SVT. The table is in the form of a matrix.

**Table 2:** Look Up Table for SV switches in the  $1\frac{1}{2}$  track 3D mesh model.

North \ South	1	2	3	4	5	6	7	8	9	10	11	12	13
1 NF-NC	H	E	E	O	O				H	C	C	K	K
2 NF-E	O	H	O						O	M	M	N	N
3 F-E	O	E	*		H				O	M	M	N	N
4 NF-W	E			H	E				E	F	F	D	D
5 F-W	E		H	O	*				E	F	F	D	D
6 NF-N						H	H						
7 F-N	H			O	O				*	C	C	K	K
8 NF-S								H					
9 F-S	H							H					
10 NF-Z <sub>-</sub>	K	J	J	L	L				K	H	K		
11 F-Z <sub>-</sub>	K	J	J	L	L				K	C	*		H
12 NF-Z <sub>+</sub>	C	B	B	A	A				C			H	C
13 F-Z <sub>+</sub>	C	B	B	A	A				C		H	K	*

The row entries correspond to the state of the PE north of SV while the column entries correspond to the state of the PE south of SV. The PE states were defined in Table 1.

To illustrate the construction of SVT, we consider  $SVT_{11,2}$ , the case corresponding to the north PE (say  $\mathcal{P}(i,j,k)$ ) being in state 11 and the south PE (say  $\mathcal{P}(i+1,j,k)$ ) being in state 2. In other words, as shown in Figure 5,  $\mathcal{P}(i,j,k)$  is F-Z<sub>-</sub> and  $\mathcal{P}(i+1,j,k)$  is NF-E and the state of SV<sub>O</sub> which lies between them is to be determined. In Figure 5,  $\mathcal{L}(i,j,k)$  is

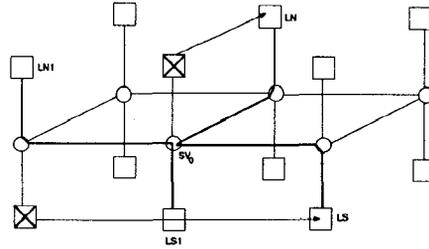


Figure 5: Example showing SV<sub>O</sub> switch setting.

denoted by LN,  $\mathcal{L}(i+1,j,k)$  is denoted by LS,  $\mathcal{L}(i,j-1,k)$  is denoted by LN1 and  $\mathcal{L}(i+1,j-1,k)$  is denoted by LS1. The SV switch north of LS will route the north link of LS to the West since it needs to send back the north link of LS in the direction opposite to direction of the compensation path that passes through  $\mathcal{P}(i+1,j+1,k)$ .  $SV_O$  expects the north link of LS on its East port. It will connect its East port to the  $Z_-$  port since LN is in the  $Z_-$  direction. The PE south of  $SV_O$  has an East path through it. Consequently,  $SV_O$  will connect its South port to its West port since LN1 is in the west direction. This state of  $SV_O$ ,  $\{C<E, Z_-, C<S, W>\}$ , is satisfied by J. Therefore entry  $SV_{T_{11,2}}$  is J. The other entries in SVT can be obtained by similar reasoning. Some entries in the SVT corresponds to compensation paths that are intersecting, overlapping, or in near-miss. These entries are left blank since they correspond to non-reconfigurable situations. Some entries in SVT are marked with a  $\star$  to indicate that the switch setting is not important since it is not used for establishing a connection.

From the above discussion, it is obvious that, if the compensation path (through the two PE's connected to SV) in any one of the three directions (vertical, horizontal and Z-direction) is missing, then the state assignment can be treated like in 2D and the two ports corresponding to the missing direction do not have to be connected. For example, if there is no Z-direction path through the two PE's flanking  $SV_O$  then the Z-ports of  $SV_O$  do not have to be connected.  $SV_O$  can be thought of lying in the  $xy$  plane requiring either  $\{C<N, W>, C<S, E>\}$  or  $\{C<N, E>, C<S, W>\}$  or  $\{C<N, S>, C<E, W>\}$ . This is reflected in the matrix on Table 2 by looking at the upper left-most  $9 \times 9$  sub-matrix which contains only three states, H, E and O. In all these three states the Z ports are not used and this is reflected in the connection  $\{C<Z_+, Z_-\}$  in all three states. From Table 2, it is seen that 13 switch states are sufficient to maintain connectivity.

It is possible to derive SZT and SHT from SVT by considering rotational symmetries. By constructing the tables SVT, SHT and SZT, we thus demonstrate that the 3D mesh connectivity can be maintained by setting the switches appropriately. That is, we prove the following theorem.

**Theorem 3** *A given fault distribution in the  $1\frac{1}{2}$  track 3D mesh is reconfigurable if the set of compensation paths covering the faulty PE's are straight, continuous, non-intersecting, non-overlapping and not in near-miss. Moreover, the state of each switch depends only on the state of its two neighboring processors.*

This is a generalization of the 2D reconfigurability theorem in [9].

#### 4: Routing

In order to obtain a routing algorithm, i.e. the distribution of the compensation paths covering all the faulty processors such that the paths satisfy the conditions in Theorem 3, the maximum independent graph theory in [9] is now extended to 3D. A completely connected graph with 6 vertices is formed per faulty PE to represent the six possible compensation path directions from the PE. Then inter-graph edges are drawn between two vertices that belong to two different graphs. An inter-graph edge represents directions of compensation paths that violate the conditions in Theorem 3. For example, in Figure 6 let  $v_i$  be a vertex in the graph for a faulty PE denoted by  $PE_1$  and  $v_j$  be a vertex in the graph for another faulty PE,  $PE_2$ , then an edge  $\langle v_i, v_j \rangle$  represents the fact that a

compensation path passing in direction  $v_i$  through  $PE_1$  and a path passing in direction  $v_j$  through  $PE_2$  will violate a compensation path condition given in Theorem 3. Specifically, if  $PE_1$  has a north compensation path, then  $PE_2$  cannot have a south compensation path since they are in near-miss. Also  $\langle N, Z_+ \rangle$  are intersecting,  $\langle Z_-, Z_+ \rangle$  are in near-miss and  $\langle Z_-, S \rangle$  are intersecting.

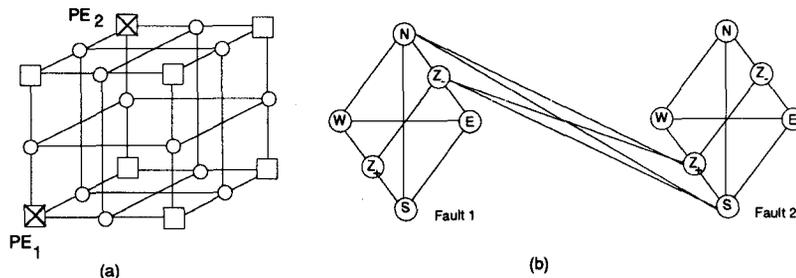


Figure 6: Compensation paths for  $PE_1$  and  $PE_2$ .

The maximal set of independent vertices, that is vertices that are not connected, is then found. Since each graph is completely connected, one vertex from each graph is chosen. If the number of vertices so found is equal to the number of faulty PE's then the fault distribution is said to be reconfigurable. The directions that should be assigned to a faulty PE is given by the direction represented by the vertex chosen from the graph of the faulty PE. For example in Figure 6 some of the paths allowed are  $\{S, E\}$ ,  $\{N, Z_-\}$  and  $\{Z_+, N\}$ .

It is well known that the solution to the maximum independent set problem is NP-complete. In [15, 14], the authors give a polynomial time algorithm in which the routing problem in the 2D  $1\frac{1}{2}$  model is treated as the problem of finding non-intersection straight lines from faulty PE's (vertices in a grid) to the the boundary of the grid. The extension of the polynomial routing algorithm to 3D is not obvious and is the subject of future research.

## 5: Conclusions

The contribution of this paper is not to compare the reliability and hardware complexity of 3D mesh models with other architectures. Its purpose is to show that the  $1\frac{1}{2}$  track model can be applied to 3D meshes as well as 2D meshes and to study the conditions and switch settings required for reconfiguration after faults in 3D meshes.

The  $1\frac{1}{2}$  track model in [9] has been extended to 3D meshes. Straight, non-intersecting, non-overlapping and non-near miss compensation paths are characterized in 3D. Three kinds of switches in 3D are defined by considering the topology of the two PE's that each switch is connected to. It is shown that local information, that is knowledge about the compensation paths passing through the two PE's connected to a switch, is sufficient to determine the state of the switch needed to preserve the 3D mesh topology. It is also shown that vertical, horizontal and Z-direction routing are independent of one another. A table, SVT, is derived for the determination of the states of SV switches and the tables for the other two kinds of switches can be obtained from SVT by rotational symmetry. The number of unique states for switches required for the  $1\frac{1}{2}$  track 3D mesh model is shown to be 13.

## References

- [1] M.J.B. Duff and T.J. Fountain, eds. *Cellular Logic Image Processing*. Academic Press, New York, NY, 1986
- [2] V. Cantoni and S. Leviaidi. Multiprocessor computing for images. *Proc. IEEE*, 76:959-969, 1988.
- [3] F.T. Leighton. *Introduction to Parallel Algorithms and Architectures*, pages 222-232. Morgan Kaufmann, 1992.
- [4] R. E. Kessler and J. L. Schwarzmeier. CRAY T3D: A new dimension for Cray research. *Proc. of IEEE Computer Conference*, Spring 1993, pages 176-182.
- [5] A. Chandra. 3D Image component labeling in 3D mesh architectures. *M.S. Thesis*, Department of Electrical Engineering, University of Pittsburgh, PA 15261, August 1994.
- [6] A.D. Singh. Interstitial redundancy: An area efficient fault tolerance scheme for Large Area VLSI processor arrays. *IEEE Trans. Computers*, Vol. 37, No. 11, pp. 1398-1410, 1988.
- [7] R. Negrini, M. Sami and R. Stefanelli. Fault tolerance techniques for array structures used in supercomputing. *Computer*, Vol. 19, No. 2, pp. 78-87, February 1986.
- [8] S.Y. Kuo and W.K. Fuchs. Efficient spare allocation in reconfigurable arrays. *IEEE Design and Test of Computers*, Vol. 4, pp. 24-31, February 1987.
- [9] S.Y. Kung, S.N. Jean and C.W. Chang. Fault tolerant array processors using single-track switches. *IEEE Trans. Computers*, Vol. 38, pp. 501-514, April 1989.
- [10] R. Melhem and J. Ramirez. Meshes with flexible redundancy. In *Algorithms and Parallel VLSI Architectures II*, editors P. Quinton and Y. Robert. Elsevier Science Publishers, 1992.
- [11] R. Libeskind-Hadas, N. Shrivastava, R. Melhem and C.L. Liu. Efficient Bi-Level Reconfiguration Algorithms for Fault Tolerant Arrays. In *Proc. of the IEEE Int. Workshop on Defect and Fault Tolerance in VLSI Systems*, Nov. 1992, pp. 42-51. An extended version will appear in *IEEE Trans. on Parallel and Distributed Systems*.
- [12] J. Ramirez and R. Melhem. Computational arrays with flexible redundancy. *IEEE Trans. on Computers*, Vol 43, No 4, pp 413-430, 1994.
- [13] M. Wang, M. Cutler and S. Y. Su. Reconfiguration of VLSI/WSI mesh array processors with VLSI/WSI two-level redundancy. *IEEE Trans. Computers*, Vol. 38, No. 4, April 1989.
- [14] T.A. Varvarigou, V.R. Roychowdhury and T. Kailath. A polynomial time-algorithm for reconfiguring multiple-track models. *IEEE Trans. Computers*, Vol. 42, No. 4, April 1993.
- [15] V.P. Roychowdhury, J. Bruck and T. Kailath. Efficient Algorithms for reconfiguration in VLSI/WSI arrays. *IEEE Trans. Computers*, Vol. 39, No. 4, April 1990.