Fault-Tolerant Meshes and Hypercubes with Minimal Numbers of Spares

Jehoshua Bruck, Robert Cypher, and Ching-Tien Ho, *Member, IEEE*

Abstract—**Many parallel computers consist of processors connected in the form of a d-dimensional mesh or hypercube. Two- and three-dimensional meshes have been shown to be efficient in manipulating images and dense matrices, whereas hypercubes have been shown to be well suited to divide-and-conquer algorithms requiring global communication. However, even a single faulty processor or communication link can seriously affect the performance of these machines.**

This paper presents several techniques for tolerating faults in d-dimensional mesh and hypercube architectures. Our approach consists of adding spare processors and communication links so that the resulting architecture will contain a fault-free mesh or hypercube in the presence of faults. We optimize the cost of the fault-tolerant architecture by adding exactly k spare processors (while tolerating up to k processor and/or link faults) and minimizing the maximum number of links per processor. For example, when the desired architecture is a d-dimensional mesh and $k = 1$, we present a fault-tolerant architecture that has the same maximum degree as the desired architecture (namely, $2d$) and has only one spare processor. We also present efficient layouts for fault-tolerant two- and three-dimensional meshes, and show how multiplexers and buses can be used to reduce the degree of fault-tolerant architectures. Finally, we give constructions for fault-tolerant tori, eight-connected meshes, and hexagonal meshes.

I. Introduction

MANY existing parallel machines have a mesh or hypercube topology. Examples of hypercube computers include the Cosmic Cube (from Caltech), the iPSC/860 (from Intel), the NCUBE (from NCUBE Inc.), and the CM-2 (from Thinking Machines). Examples of two-dimensional mesh computers include the MPP (from Goodyear Aerospace) [3], the MP-1 (from MASPAR), VICTOR (from IBM), and DELTA (from Intel and Caltech). The J-Machine, which is under development at MIT, and the GC series from Parsytec [20] are three-dimensional meshes. In addition, memory chips are also organized in the form of a two-dimensional mesh [16], [27].

As improvements in technology lead to the creation of larger parallel computers, it becomes essential to consider the issue of computing in the presence of faults. In particular, the ability to tolerate even a small number of faults could allow the machine to be used between the time a failure is first detected and the time the machine is repaired. As a result, several existing parallel machines contain spare processors and are designed to tolerate a limited number of faults [3], [20].

A large amount of research has been devoted to creating fault-tolerant parallel architectures. The techniques used in this research can be divided into two main classes. The first class consists of techniques that *do not* add redundancy to the desired architecture. Instead, these techniques attempt to mask the effects of faults by using the healthy part of the architecture to simulate the entire machine [1], [6], [12], [14]. The hope with this approach is to obtain the same functionality with a reasonable slowdown factor. Although this approach yields interesting theoretical results, even a constant factor slowdown in performance can be very significant in practice. Furthermore, this approach requires that some healthy processors simulate several processors. As a result, each simulated processor can have only a fraction of the memory present in a healthy processor.

The second class consists of techniques that *do* add redundancy to the desired architecture. These techniques attempt to isolate the faults, usually by disabling certain links or disallowing certain switch settings, while maintaining the complete desired architecture [2], [3], [5], [8]–[10], [13], [15]–[17], [19], [21]–[24], [26], [28]. Many of these techniques require either a nonminimal number of spare processors [2], [3], [5], [15], [16], [24], [26] or a switching mechanism assumed to be immune to faults [3], [15], [16], [21], [22], [24], [26]. In contrast, the results presented herein use only the minimal number of spare processors and can tolerate faults in any of the components. Finally, we assume a worst case distribution of faults, whereas many of the preceding approaches do not work in a worst case scenario.

Our approach is based on a graph model. In this model a distributed memory parallel computer is viewed as being a graph in which the nodes represent the processors and the edges represent the communication links. A target graph with n nodes is selected first. Then a fault-tolerant graph with $n + k$ nodes is defined with the property that, given any set of k or fewer faulty nodes, the remaining graph is guaranteed to contain the target graph as a subgraph. This approach guarantees that any algorithm designed for the target graph will run *with no slowdown* in the presence of k or fewer node faults, regardless of their distribution. Note that in our approach the spare nodes are fully utilized. Hence, minimizing the cost in this model amounts to constructing a fault-tolerant graph with

Manuscript received July 2, 1991; revised November 12, 1991, and August 28, 1992. This work is based on "Fault-Tolerant Meshes with Minimal Numbers of Spares," by J. Bruck, R. Cypher, and C. T. Ho, which appeared in the *Proceedings of the 3rd IEEE Symposium on Parallel and Distributed Processing,* Dallas, TX, Dec. 2–5, 1991, pp. 288–295, and "Efficient Fault-Tolerant Mesh and Hypercube Architectures," which appeared in the *Proceedings of the 22nd Annual International Symposium on Fault-Tolerant Computing,* Boston, MA, July 8–10, 1992, pp. 162–169. ©1991, 1992 IEEE.
The authors are with IBM, Almaden Research Center, San Jose, CA 95120.
IEEE Log Number 9208568.

0018–9340/93$03.00 © 1993 IEEE

a small maximum degree. Although our results are stated for node faults, it should be noted that they can also be used to tolerate edge faults by viewing a node incident with each faulty edge as being faulty.

This graph model of fault tolerance has been used by several other researchers. Hayes [13] has used this model with target graphs of cycles, linear arrays, and trees. Rosenberg examined fault-tolerant graphs for linear arrays [23]. The work by Wong and Wong [28] and Paoli *et al.* [19] relates to cycles. The more recent work by Dutt and Hayes uses trees [8], hypercubes [9], circulant and nearly circulant graphs [9], and arbitrary graphs [10] as target graphs.

The main contribution of this paper is the creation of efficient fault-tolerant graphs for several important target graphs. Specifically, we give four different constructions for creating fault-tolerant two-dimensional meshes, as well as constructions for creating fault-tolerant d -dimensional meshes, hypercubes, tori, eight-connected meshes, and hexagonal meshes. In all cases, our fault-tolerant graphs have a smaller degree than any previously known graphs with the same properties. In particular, we present a construction for fault-tolerant d-dimensional meshes that can tolerate k faults and has degree $(k+1)d$ when k is odd and $(k+2)d$ when k is even. Thus when $k=1$ this construction has degree $2d$, which is no larger than the degree of the target graph. Our fault-tolerant graph for the d-dimensional hypercube has degree $(k+2)d-(k+2)\log k+2k-3$ when k is a power of 2. This is approximately a factor of 2 improvement over the result obtained by Dutt and Hayes, which has degree $2((k+1)d-(k+1)\log k-3)$ when k is a power of 2 [9]. We also show how multiplexers and buses can be used to reduce the degree of the fault-tolerant architectures.

The rest of this paper is organized as follows. Definitions that will be used throughout the paper are given in Section II. In Section III, we present several fault-tolerant two-dimensional meshes, all of which are based on a family of graphs known as "circulant graphs." In Section IV, we introduce another family of graphs, called "diagonal graphs," and show how they can be used to create fault-tolerant d-dimensional meshes and hypercubes. We also present efficient implementations for many of these fault-tolerant architectures. Section V shows how the same techniques can be used to create fault-tolerant graphs for target graphs that are related to the mesh. Conclusions are presented in Section VI.

II. Definitions

The following definitions will be used throughout this paper.

Definition: Let k be a nonnegative integer and let $G=(V,E)$ be a graph. We say that the graph $G'=(V',E')$ is (k,G)-tolerant if the subgraph of G' induced by any set of $|V'|-k$ nodes contains G as a subgraph. We note here that throughout this paper the number of spare nodes is minimal, so $|V'|=|V|+k$.

Definition: Given two graphs G_1 and G_2, a function of ϕ that maps the vertices of G_1 to the vertices of G_2 is called an *embedding of* G_1 *into* G_2 if for any pair of distinct nodes i and j in $G_1, \phi(i)\neq\phi(j)$, and for any edge (i,j) in $G_1, (\phi(i),\phi(j))$ is an edge in G_2.

Definition: For any positive integer n, the set $\{0,1,\cdots,n-1\}$ will be denoted $[n]$.

Definition: Let $n_0,n_1,\cdots,n_{d-1}$ be integers all of which are greater than or equal to 2. The $n_0\times n_1\times\cdots n_{d-1}$ *d-dimensional mesh* M consists of $\Pi_{i=0}^{d-1} n_i$ nodes. Each node in M has a unique label of the form $(a_0,a_1,\cdots,a_{d-1})$ where for all $i\in[d], a_i\in[n_i]$. Each node $(a_0,a_1,\cdots,a_{d-1})$ is connected to the $2d$ other nodes of the form $(a_0,\cdots,a_{i-1},a\pm 1,a_{i+1},\cdots,a_{d-1})$ provided they exist.

Definition: The d-dimensional hypercube, denoted Q_d, is a $2\times 2\times\cdots\times 2$ d-dimensional mesh with $n=2^d$ nodes. Note that hypercubes are simply special types of d-dimensional meshes, so that results presented for d-dimensional meshes will apply to d-dimensional hypercubes as well.

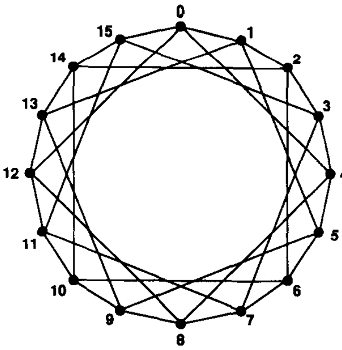


Fig. 1. Circulant graph with 16 nodes and offsets 1 and 4.

III. Circulant Graphs

This section discusses a class of graphs known as "circulant graphs" [11] and shows how they can be used to create fault-tolerant two-dimensional meshes. We begin by defining circulant graphs and reviewing some of their known properties.

Definition: Let p be a positive integer and let S be a set of integers in the range 1 through $p-1$. The p *-node circulant graph with connection set* S, denoted $C_{p,S}$, consists of p nodes. Each node in $C_{p,S}$ has a unique label in the range 0 through $p-1$. Each node i is connected to all nodes of the form $(i\pm s)$ mod p where $s\in S$. The values in the connection set S will be referred to as "jumps" or "offsets." A simple example of a circulant graph is a cycle, where there is only one offset and the value of that offset is 1. Fig. 1 shows an example of a circulant graph.

Definition: Let p be a positive integer and let S be a set of integers in the range 1 through $p-1$. The *closure of* S *by* p, denoted *close*(S,p), is the set

$$T=\{t|t\in S \text{ or } (p-t)\in S\}.$$

Note that the degree of $C_{p,S}$ is $|close(S,p)|$. In addition, note that $|S|\leq|close(S,p)|\leq 2|S|$.

Definition: Let S be a set of integers and let k be a nonnegative integer. The *expansion of* S *by* k, denoted $expand(S,k)$, is the set T, where

$$T = \bigcup_{s \in S} \{s, s+1, \cdots, s+k\}.$$

Note that $|expand(S,k)| \leq (k+1)|S|$.

The following theorem is an immediate consequence of a result proven by Dutt and Hayes [9].

Theorem 3.1 [9]: Let n be a positive integer, let S be a set of integers in the range 1 through $n-1$, let k be a nonnegative integer, and let $T = expand(S,k)$. The circulant graph $C_{n+k,T}$ is $(k, C_{n,S})$-tolerant.

The idea behind Theorem 3.1 is that given any set of k faulty nodes in $C_{n+k,T}$, we can embed the target graph $C_{n,S}$ into the healthy nodes of the fault-tolerant graph $C_{n+k,T}$ by mapping each node i in the target graph to the ith healthy node in the fault-tolerant graph. It is clear that any pair of nodes that are x apart in the target graph are mapped to nodes in the fault-tolerant graph that are at least x apart and at most $x+k$ apart (because there are between 0 and k faulty nodes between them). Consider any edge that connects nodes that are x apart in the target graph, where $x \in S$. This edge will be mapped to nodes that are x' apart in the fault-tolerant graph, where $x' \in T$, so it will be mapped to an edge in the fault-tolerant graph.

Theorem 3.1 gives a general technique for creating a fault-tolerant graph when the target graph is circulant. We will use Theorem 3.1 to obtain four different constructions for fault-tolerant two-dimensional meshes. Each construction first defines a circulant graph, which is a supergraph of the desired two-dimensional mesh. Then Theorem 3.1 is used to add fault tolerance to the supergraph. It is interesting to note that circulant graphs that contain two-dimensional meshes as subgraphs have been studied in a context unrelated to fault tolerance [4].

Throughout the remainder of this section, let r and c be integers greater than or equal to 2, and let k be a nonnegative integer. Additional constraints on these parameters will be added as needed. In addition, let $M_{r,c}$ denote the two-dimensional mesh with r rows and c columns. The four different constructions and their degrees are given in Theorems 3.3, 3.5, 3.7, and 3.14. Another construction for fault-tolerant two-dimensional meshes is given in Corollary 4.7 in Section IV. This final construction has the smallest degree when the number of faults that must be tolerated is small.

A. Mesh Construction 1

The first fault-tolerant mesh construction is based on the fact that, when the nodes in $M_{r,c}$ are labeled in row-major order, the labels of adjacent nodes differ by either 1 or c [see Fig. 2(a)].

Lemma 3.2: Let $S = \{1, c\}$. The mesh $M_{r,c}$ is a subgraph of the circulant graph $C_{rc,S}$.

Proof: Let $\phi(i,j) = ic + j$. It is straightforward to verify that ϕ defines an embedding of $M_{r,c}$ into $C_{rc,S}$. □

Theorem 3.3: Let $S = \{1, c\}$ and let $T = expand(S,k)$. The circulant graph $C_{rc+k,T}$ is $(k, M_{r,c})$-tolerant and has degree at most $4k+4$.

0	1	2	3	4	5	6	7
8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23
24	25	26	27	28	29	30	31
32	33	34	35	36	37	38	39

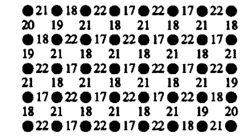
(a)

0	9	18	27	36	5	14	23
8	17	26	35	4	13	22	31
16	25	34	3	12	21	30	39
24	33	2	11	20	29	38	7
32	1	10	19	28	37	6	15

(b)

0	21	3	25	8	30	13	35
20	2	24	7	29	12	34	17
1	23	6	28	11	33	16	38
22	5	27	10	32	15	37	19
4	26	9	31	14	36	18	39

(c)



(d)

Fig. 2. Three orderings of mesh nodes.

Proof: From Theorem 3.1, the graph $C_{rc+k,T}$ is $(k, C_{rc,S})$-tolerant. From Lemma 3.2, the graph $M_{r,c}$ is a subgraph of $C_{rc,S}$. As a result, the graph $C_{rc+k,T}$ is $(k, M_{r,c})$-tolerant. Because $|S| \leq 2$ and $T = expand(S,k)$, $|T| \leq 2k+2$ and the degree of $C_{rc+k,T}$ is at most $4k+4$. □

B. Mesh Construction 2

Whereas Construction 1 is a very natural application of Theorem 3.1, Theorem 3.1 can also be used to obtain more efficient constructions. We will now give a construction for obtaining a graph that tolerates k faults and has degree only $2k+4$. This construction is based on an ordering of the nodes in the mesh that we call the *antidiagonal-major order* [see Fig. 2(b)]. The advantage of antidiagonal-major order is that it leads to a circulant graph that has a connection set consisting of two consecutive integers. As a result, fault tolerance can be added to the circulant graph in an efficient manner.

Lemma 3.4: Let $S = \{c, c+1\}$. The mesh $M_{r,c}$ is a subgraph of the circulant graph $C_{rc,S}$.

Proof: Let $\phi(i,j) = ((i+j) \bmod r)c + j$. It is straightforward to verify that ϕ defines an embedding of $M_{r,c}$ into $C_{rc,S}$ □

Theorem 3.5: Let $S = \{c, c+1\}$ and let $T = expand(S,k)$. The circulant graph $C_{rc+k,T}$ is $(k, M_{r,c})$-tolerant and has degree at most $2k+4$.

Proof: From Theorem 3.1, the graph $C_{rc+k,T}$ is $(k, C_{rc,S})$-tolerant. From Lemma 3.4, the graph $M_{r,c}$ is a subgraph of $C_{rc,S}$. As a result, the graph $C_{rc+k,T}$ is $(k, M_{r,c})$-tolerant. Because $T = \{c, c+1, \cdots, c+k+1\}$, $|T| \leq k+2$ and the degree of $C_{rc+k,T}$ is at most $2k+4$. □

C. Mesh Construction 3

The fault-tolerant meshes produced by Construction 2 require two additional edges per node for each additional fault that is tolerated. We will now give a construction that requires only one additional edge per node for each additional fault that is tolerated. However, this reduced rate of growth in the degree requires a larger initial degree.

The construction is based on an ordering of the nodes in the mesh that we call the *interleaved antidiagonal-major order* [see Fig. 2(c)]. The interleaved antidiagonal-major order assigns the numbers 0 through $rc-1$ to the nodes in $M_{r,c}$. Node (0, 0) (the upper left corner) is assigned the value 0, and successive values are assigned to the nodes in every other antidiagonal. Then node (1, 0) (the node immediately below the upper left corner) is assigned the value $\lceil rc/2\rceil$, and successive values are assigned to the nodes in the remaining antidiagonals. The advantage of interleaved antidiagonal-major order is that it leads to a circulant graph with rc nodes that has a connection set clustered about the value $rc/2$.

Lemma 3.6: Let r and c be integers greater than 2, let $a=\lceil rc/2\rceil-\lceil r/2\rceil$, let $b=\lceil rc/2\rceil+\lfloor r/2\rfloor$, and let S be the set of integers in the range a through b. The mesh $M_{r,c}$ is a subgraph of the circulant graph $C_{rc,S}$.

Proof: Let $\phi(i,j)$ be the value assigned to node (i,j) when $M_{r,c}$ is labeled in interleaved antidiagonal-major order. It will be shown that for any nodes (i_1,j_1) and (i_2,j_2) that are adjacent in $M_{r,c}, a\le|\phi(i_1,j_1)-\phi(i_2,j_2)|\le b$.

For all integers i and j where $0\le i\le r-1$ and $0\le j<c-1$, let $h_{i,j}=|\phi(i,j)-\phi(i,j+1)|$. For all integers i and j where $0\le i<r-1$ and $0\le j\le c-1$, let $v_{i,j}=|\phi(i,j)-\phi(i+1,j)|$. We will call the $h_{i,j}$ values *horizontal differences* and the $v_{i,j}$ values *vertical differences* [see Fig. 2(d)]. First, we will show that if there is a vertical difference that is not in the range a through b, then there is also a horizontal difference that is not in the range a through b. Let $v_{i,j}$ be any vertical difference. If $j=0$ then either $h_{i,j}<v_{i,j}<h_{i+1,j}$ or $h_{i+1,j}<v_{i,j}<h_{i,j}$. Conversely, if $j>0$ then either $h_{i,j-1}<v_{i,j}<h_{i+1,j-1}$ or $h_{i+1,j-1}<v_{i,j}<h_{i,j-1}$. Therefore, if the $v_{i,j}$ is not in the range a through b, there also exists an $h_{i',j'}$ that is not in the range a through b.

It is clear that for all i and j where $1\le i\le r-1$ and $0\le j\le c-3, h_{i,j}=h_{i-1,j+1}$, so horizontal differences that are in the same antidiagonal are equal. It will be helpful to divide the horizontal differences into two sets according to their parity. We will say that horizontal difference $h_{i,j}$ is *even* if $i+j$ is even, and it is *odd* otherwise. It is straightforward to show that all even horizontal differences are greater than $rc/2$ and all odd horizontal differences are less than $rc/2$. Furthermore, it is straightforward to show that the largest horizontal difference is $h_{r-2,0}$ if r is even and $h_{r-1,0}$ if r is odd, whereas the smallest horizontal difference is $h_{r-1,0}$ if r is even and $h_{r-2,0}$ if r is odd. Therefore, it suffices to show that $h_{r-2,0}$ and $h_{r-1,0}$ are in the range a through b. There are two cases:

*Case 1) **r** is even:* In this case, $\phi(r-2,0)=r^2/4-r+1, \phi(r-2,1)=rc/2+r^2/4-r/2+1, \phi(r-1,0)=rc/2+r^2/4-r/2$, and $\phi(r-1,1)=r^2/4$. Thus, $h_{r-2,0}=rc/2+r/2=b$ and $h_{r-1,0}=rc/2-r/2=a$.

*Case 2) **r** is odd:* In this case, $\phi(r-2,0)=\lceil rc/2\rceil+\lfloor r/2\rfloor^2-\lfloor r/2\rfloor, \phi(r-2,1)=\lfloor r/2\rfloor^2+1, \phi(r-1,0)=\lfloor r/2\rfloor^2, \phi(r-1,1)=\lceil rc/2\rceil+\lfloor r/2\rfloor^2+\lfloor r/2\rfloor$. Thus, $h_{r-2,0}=\lceil rc/2\rceil-\lceil r/2\rceil=a$ and $h_{r-1,0}=\lceil rc/2\rceil+\lfloor r/2\rfloor=b$.

It has been shown that all horizontal and vertical differences are in the range a through b. Furthermore, ϕ assigns a unique value in the range 0 through $rc-1$ to each node in $M_{r,c}$. As a result, it follows that ϕ defines an embedding of $M_{r,c}$ into $C_{rc,S}$. □

Theorem 3.7: Let r and c be integers greater than 2, let $a=\lceil rc/2\rceil-\lceil r/2\rceil$, let $b=\lceil rc/2\rceil+\lfloor r/2\rfloor$, let S be the set of integers in the range a through b, and let $T=expand(S,k)$. The circulant graph $C_{rc+k,T}$ is $(k,M_{r,c})$-tolerant and has degree at most $k+r+1$ when r is odd and c is even, and at most $k+r$ otherwise.

Proof: From Theorem 3.1, the graph $C_{rc+k,T}$ is $(k,C_{rc,S})$-tolerant. From Lemma 3.6, the graph $M_{r,c}$ is a subgraph of $C_{rc,S}$. As a result, the graph $C_{rc+k,T}$ is $(k,M_{r,c})$-tolerant. Note that $T=\{a,a+1,\cdots,b+k\}$ and that $b-a=r$. If r is odd and c is even, then $close(T,rc+k)=\{a,a+1,\cdots,b+k+1\}$ and the degree of $C_{rc+k,T}$ is at most $k+r+1$. Otherwise, $close(T,rc+k)=\{a,a+1,\cdots,b+k\}$ and the degree of $C_{rc+k,T}$ is at most $k+r$. □

Theorem 3.7 is based on Lemma 3.6, which showed that the mesh $M_{r,c}$ is a subgraph of a circulant graph with rc nodes and a connection set that has values near $rc/2$. Specifically, when r is odd and c is even, all of the values in the connection set are within $(r+1)/2$ of $rc/2$, and in all other cases all of the values in the connection set are within $r/2$ of $rc/2$. If Lemma 3.6 could be improved by finding a circulant graph with a connection set that is more tightly clustered around $rc/2$, the degree of the construction in Theorem 3.7 could be reduced. However, as we will see in Theorem 3.8, no such improvement in Lemma 3.6 is possible. The proof of Theorem 3.8 is given in the Appendix.

Theorem 3.8: Let r and c be integers where $4\le r\le c$ and let $C_{rc,S}$ be a circulant graph that contains the mesh $M_{r,c}$ as a subgraph. There exists an $s\in S$ such that $|s-(rc/2)|\ge(r+1)/2$ if r is odd and c is even, and such that $|s-(rc/2)|\ge r/2$ otherwise.

D. Mesh Construction 4

We will now present constructions of $(k,M_{r,c})$-tolerant graphs that combine the advantages of Constructions 2 and 3. More precisely, the degree of the construction given here increases at the rate of two per fault up to some number of faults, at which point it increases at the rate of one per fault. The cut-off point at which the rate of growth in the degree slows depends on a value called the *gap*, which will be defined later.

Lemma 3.9: The following properties hold. i) If r is odd then r and $(r-1)/2$ are relatively prime. ii) If $r \bmod 4=0$ then r and $(r/2)-1$ are relatively prime. iii) If $r \bmod 4=2$ then r and $(r/2)-2$ are relatively prime.

Proof: First, we prove i). Let $r=2x+1$ where x is an integer. Thus $\gcd(r,(r-1)/2)=\gcd(2x+1,x)=\gcd(1,x)=1$, so r and $(r-1)/2$ are relatively prime. To prove ii), let $r=4x$ where x is an integer. Thus $\gcd(r,(r/2)-1)=\gcd(4x,2x-1)=\gcd(2,2x-1)=1$. To prove iii), let $r=4x+2$ where x is an integer. Thus, $\gcd(r,(r/2)-2)=\gcd(4x+2,2x-1)=\gcd(4,2x-1)=1$. □

Lemma 3.10: Let r be odd and let $S=\{(r-1)c/2,(r-1)c/2+1\}$. The mesh $M_{r,c}$ is a subgraph of the circulant graph $C_{rc,S}$.

Proof: Let $f(i)=(i(r-1)/2) \bmod r$ and let $\phi(i,j)=f(i+j)c+j$. First, we will show that ϕ maps distinct nodes to

(a)

0	13	26	11
12	25	10	23
24	9	22	7
8	21	6	19
20	5	18	3
4	17	2	15
16	1	14	27

(b)

0	13	26	7
12	25	6	19
24	5	18	31
4	17	30	11
16	29	10	23
28	9	22	3
8	21	2	15
20	1	14	27

(c)

0	13	26	39
12	25	38	11
24	37	10	23
36	9	22	35
8	21	34	7
20	33	6	19
32	5	18	31
4	17	30	3
16	29	2	15
28	1	14	27

Fig. 3. Three orderings of mesh nodes for the mesh construction 4: (a) r is odd, (b) $r \bmod 4 = 0$, and (c) $r \bmod 4 = 2$.

distinct values. From Lemma 3.9, r and $(r-1)/2$ are relatively prime, so for any integers x and x', if $f(x) = f(x')$ it follows that $x \bmod r = x' \bmod r$. Let (i,j) and (i',j') be any nodes in $M_{r,c}$. Note that $f(i+j)c$ and $f(i'+j')c$ are multiples of c, and that $j \in [c]$ and $j' \in [c]$. Therefore, if $\phi(i,j) = \phi(i',j')$, it follows that $j = j'$, which implies that $f(i+j) = f(i'+j')$, so $i \bmod r = i' \bmod r$ and $i = i'$.

We will now show that ϕ maps edges in $M_{r,c}$ to edges in $C_{rc,S}$. We will show this by proving that for any integers i and j, i) $|\phi(i+1,j) - \phi(i,j)| \in close(S, rc)$, and ii) $|\phi(i,j+1) - \phi(i,j)| \in close(S, rc)$. Let $s = (r-1)c/2$ and let $s' = (r+1)c/2$, and note that $close(S, rc) = \{s, s+1, s'-1, s'\}$. In addition, note that for any integer x, $f(x+1) - f(x)$ equals either $(r-1)/2$ or $-(r+1)/2$. Therefore, $|\phi(i+1,j) - \phi(i,j)| = |f(i+j+1) - f(i+j)|c$, which equals either s or s', and property i holds. Let $\phi(i,j) = y$. Clearly, $\phi(i,j+1) = \phi(i+1,j)+1$, which equals either $y+s+1$ or $y-s'+1$, so property ii holds as well. As a result, ϕ is an embedding of $M_{r,c}$ into $C_{rc,S}$. □

Lemma 3.11: Let $r \bmod 4 = 0$ and let $S = \{(r/2-1)c, (r/2-1)c+1\}$. The mesh $M_{r,c}$ is a subgraph of the circulant graph $C_{rc,S}$.

Proof: Let $f(i) = ((r/2-1)i) \bmod r$ and let $\phi(i,j) = f(i+j)c + j$. The proof is analogous to that of Lemma 3.10 and will not be repeated here. □

Lemma 3.12: Let $r \bmod 4 = 2$ and let $S = \{(r/2-2)c, (r/2-2)c+1\}$. The mesh $M_{r,c}$ is a subgraph of the circulant graph $C_{rc,S}$.

Proof: Let $f(i) = ((r/2-2)i) \bmod r$ and let $\phi(i,j) = f(i+j)c + j$. The proof is analogous to that of Lemma 3.10 and will not be repeated here. □

Fig. 3 shows examples of the mapping ϕ for three meshes, representing the cases of Lemmas 3.10 through 3.12, respectively.

Definition: Let n, s, and x be integers where $0 \le s < \lfloor n/2 \rfloor$ and $0 \le x \le s$, and let $S = \{s-x, s-x+1, \cdots, s\}$. Then $gap(S,n) = n - 1 - 2s$.

Intuitively, $gap(S,n)$ is the length of the "gap" between the two consecutive groups of offsets in $close(S,n)$. For instance, if $S = \{5,6\}, close(S,16) = \{5,6,10,11\}$ and $gap(S,16) = 3$. The following lemma is similar to one proven by Dutt and Hayes [9].

Lemma 3.13: Let n, s, and x be integers where $0 \le s < \lfloor n/2 \rfloor$ and $0 \le x \le s$, let $S = \{s-x, s-x+1, \cdots, s\}$ and let $T = expand(S,k)$. The circulant graph $C_{n+k,T}$ has degree at most

$$d = \begin{cases} 2k + 2|S|, & \text{if } k \le gap(S,n), \\ k + 2|S| + gap(S,n), & \text{if } k > gap(S,n). \end{cases}$$

Proof: The fact that $d \le 2k + 2|S|$ follows immediately from the definitions of expansion and closure. Now consider the case where $k > gap(S,n)$. Note that $T = \{s-x, s-x+1, \cdots, s+k\}$ so we will compare the values of $s+k$ and $n+k-(s+k) = n-s$ to see if there is a gap between the two groups of offsets in $close(T, n+k)$. Note that $(n-s)-(s+k) = n-2s-k$, which is less than $n-2s-(n-1-2s) = 1$ because $k > gap(S,n) = n-1-2s$. Therefore, there is no gap between the two groups of offsets in $close(T, n+k)$, so $close(T, n+k) = \{s-x, s-x+1, \cdots, n+k-(s-x)\}$ and $|close(T, n+k)| = n+k-(s-x)-(s-x-1) = n+k-2s+2x+1$. Since $k + 2|S| + gap(S,n) = k + 2(x+1) + n - 1 - 2s$, we have $|close(T, n+k)| = k + 2|S| + gap(S,n)$. □

Theorem 3.14: Let

$$S = \begin{cases} \left\{\dfrac{(r-1)c}{2}, \dfrac{(r-1)c}{2}+1\right\} & \text{if } r \text{ is odd}, \\ \left\{\left(\dfrac{r}{2}-1\right)c, \left(\dfrac{r}{2}-1\right)c+1\right\} & \text{if } r \bmod 4 = 0, \\ \left\{\left(\dfrac{r}{2}-2\right)c, \left(\dfrac{r}{2}-2\right)c+1\right\} & \text{if } r \bmod 4 = 2, \end{cases}$$

and let $T = expand(S,k)$. Then the circulant graph $C_{rc+k,T}$ is $(k, M_{r,c})$-tolerant and has degree at most

$$d(k,r,c) = \begin{cases} 2k+4, & r \text{ is odd and } k \le c-3, \\ c+k+1, & r \text{ is odd and } k > c-3, \\ 2k+4, & r \bmod 4 = 0 \text{ and } k \le 2c-3, \\ 2c+k+1, & r \bmod 4 = 0 \text{ and } k > 2c-3, \\ 2k+4, & r \bmod 4 = 2 \text{ and } k \le 4c-3, \\ 4c+k+1, & r \bmod 4 = 2 \text{ and } k > 4c-3. \end{cases}$$

Proof: From Theorem 3.1, the graph $C_{rc+k,T}$ is $(k, C_{rc,S})$-tolerant. From Lemmas 3.10 to 3.12, the graph $M_{r,c}$ is a subgraph of $C_{rc,S}$. Thus, the graph $C_{rc+k,T}$ is $(k, M_{r,c})$-tolerant. The degree follows directly from Lemma 3.13 and the fact that $gap(S, rc) = c-3$ if r is odd, $2c-3$ if $r \bmod 4 = 0$, and $4c-3$ if $r \bmod 4 = 2$. □

Note that if the numbers $0, 1, \cdots, rc-1$ are assigned to meshes in the diagonal manner instead of the antidiagonal manner, then the new $gap(S, rc)$ is greater than the original $gap(S, rc)$ by 2.

IV. Diagonal Graphs

In Section III, we studied the family of circulant graphs. Another important class of graphs consists of what we call "diagonal graphs." In this section we will show that diagonal graphs can be used to create fault-tolerant d-dimensional meshes and hypercubes with small degree. We will also present efficient implementations for many of these fault-tolerant graphs. The definition of diagonal graphs and a general technique for adding fault tolerance to diagonal graphs are given next.

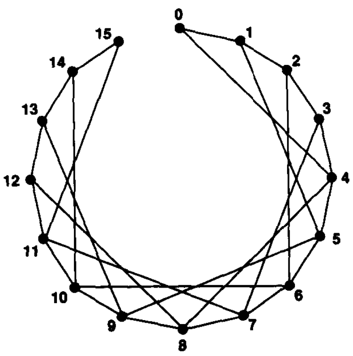


Fig. 4. Diagonal graph with 16 nodes and offsets 1 and 4.

Definition: Let n be a positive integer and let S be a set of integers in the range 1 through $n-1$. The ***n*-node diagonal graph with connection set *S***, denoted $D_{n,S}$, consists of n nodes. Each node in $D_{n,S}$ has a unique label in the range 0 through $n-1$. Each node i is connected to all nodes of the form $i \pm s$ where $s \in S$. Thus diagonal graphs are similar to circulant graphs, except they do not have the "wraparound" connections from high numbered nodes to low numbered nodes. Fig. 4 shows an example of a diagonal graph. The name "diagonal graph" refers to the structure of the adjacency matrix of such a graph.

Given the target graph $D_{n,S}$ (with the restriction that $S \subseteq \{1, 2, \cdots, \lceil n/3 \rceil\}$), we will use the circulant graph $C_{n+k,T}$, where $T = expand(S, \lfloor k/2 \rfloor)$, as the fault-tolerant graph. The idea is similar to the technique for adding fault tolerance to circulant graphs given in Theorem 3.1. Recall that given the circulant target graph $C_{n,S}$, the fault-tolerant graph has the connection set $T = expand(S, k)$. The reason that we have to expand S by k is that an edge in the target graph may have to "jump over" as many as k faults in the fault-tolerant graph. In contrast, given the diagonal target graph $D_{n,S}$, the fault-tolerant graph requires only the connection set $T = expand(S, \lfloor k/2 \rfloor)$. The reason that we can expand S by $\lfloor k/2 \rfloor$ rather than by k is that the lowest and highest numbered nodes in $D_{n,S}$ have smaller degree than the other nodes in $D_{n,S}$. Thus if the fault-tolerant graph has a cluster of faults that are near one another (and thus could require an edge to jump over a large number of faults), we can choose to map the lowest and highest numbered nodes in $D_{n,S}$ to the healthy nodes near that cluster of faults. Using this mapping none of the edges has to jump over the cluster of faults, and the expansion by $\lfloor k/2 \rfloor$ is sufficient. This argument is formalized subsequently.

Definitions: Let n and k be positive integers, let $y = \lceil n/3 \rceil$, let $P = [n+k]$, let $F \subset P$ be any set of k elements in P, and let $H = P \backslash F$. The set P will represent the $n+k$ processors in the fault-tolerant graph, the set H will represent the n healthy processors, and the set F will represent the k faulty processors. The elements of F and H will be denoted $f_0, f_1, \cdots, f_{k-1}$ and $h_0, h_1, \cdots, h_{n-1}$, respectively, where $f_i < f_j$ if $i < j$ and $h_i < h_j$ if $i < j$. Given any healthy node $h_i \in H$ and any integer x where $1 \le x \le n-1, jumps(h_i, x)$ is defined to be $\{f_j | h_i < f_j < h_{i+x}\}$ if $i + x < n$ and $\{f_j | h_i < f_j \text{ or } f_j < h_{i+x-n}\}$ otherwise. In additior, $marked(x) = \{h_i | \, |jumps(h_i, x)| > k/2\}$ and $tail(h_i, x) = \{h_i, h_{(i-1) \bmod n}, h_{(i-2) \bmod n}, \cdots, h_{(i-x+1) \bmod n}\}$. Thus, $jumps(h_i, x)$ is the set of faults that must be jumped over when connecting h_i to the xth healthy node following $h_i, marked(x)$ is the set of healthy nodes that jump over a majority of the faults when they are connected to the xth healthy node following them, and $tail(h_i, x)$ is the set of x consecutive healthy nodes (in cyclic order) ending with h_i. Finally, for any healthy nodes h_i and h_j, let $dist(h_i, h_j) = \min((i-j) \bmod n, (j-i) \bmod n)$. Note that $dist(h_i, h_j)$ denotes the distance (in cyclic order) between h_i and h_j, where only healthy nodes are considered to contribute to the distance.

Lemma 4.1: Given n, k, y, P, F, and H as defined earlier, and given any nonempty set $X \subseteq H$, if for all h_i and h_j in $X, dist(h_i, h_j) \le y - 1$, then there exists some $h_c \in X$ such that $X \subseteq tail(h_c, y)$.

Proof: Let h_b be an arbitrary element in X. Let $h_a = h_{(b-y+1) \bmod n}$ and let $h_d = h_{(b+y-1) \bmod n}$. Note that every member of X appears in the sequence

$$W = h_a, (h_a + 1) \bmod (n+k). \\ (h_a + 2) \bmod (n+k), \cdots, h_d$$

because for each $h_i \in X, dist(h_b, h_i) \le y = 1$. Let h_c be the last element in X to appear in the sequence W. Because W contains $2y - 1$ members of H, there must be at least $n - 2y + 1 \ge y - 1$ members of H that do not appear in W. Furthermore, for each $h_i \in X, dist(h_c, h_i) \le y - 1$, so $h_i \in tail(h_c, y)$. As a result, $X \subseteq tail(h_c, y)$. □

Lemma 4.2: Given n, k, y, P, F, and H as defined earlier, there exists some element $h_c \in H$ such that for all x, where $1 \le x \le y, marked(x) \subseteq tail(h_c, x)$.

Proof: We will assume that $marked(y)$ is not empty, because the lemma is trivially true otherwise. First, we will show that there exists an element $h_c \in marked(y)$ such that $marked(y) \subseteq tail(h_c, y)$. We will then show that for all x, where $1 \le x \le y - 1, marked(x) \subseteq tail(h_c, x)$.

Let h_i and h_j be arbitrary members of $marked(y)$. Because $jumps(h_i, y)$ and $jumps(h_j, y)$ both contain a majority of the elements in F, there must be some $f_a \in F$ such that $f_a \in jumps(h_i, y)$ and $f_a \in jumps(h_j, y)$. As a result, $dist(h_i, h_j) \le y - 1$. Therefore, it follows from Lemma 4.1 that there is some $h_c \in marked(y)$ such that $marked(y) \subseteq tail(h_c, y)$.

We will now show that for any x, where $1 \le x \le y - 1, marked(x) \subseteq tail(h_c, x)$. Clearly, $marked(x) \subseteq marked(y)$, so $marked(x) \subseteq tail(h_c, y)$. Thus all that remains to be shown is that $tail(h_c, y) \backslash tail(h_c, x)$ does not contain any elements in $marked(x)$. Let h_i be an arbitrary member of $tail(h_c, y) \backslash tail(h_c, x)$. Note that $jumps(h_i, x) \cap jumps(h_c, y) = \emptyset$ and $|jumps(h_c, y)| > k/2$. so $|jumps(h_i, x)| < k/2$ and $h_i \notin marked(x)$. Therefore, $tail(h_c, y) \backslash tail(h_c, x)$ does not contain any members of $marked(x)$, and $marked(x) \subseteq tail(h_c, x)$. □

Theorem 4.3: Let n be a positive integer, let $y = \lceil n/3 \rceil$, let S be a set of integers in the range 1 through y, let k be a positive integer, and let $T = expand(S, \lfloor k/2 \rfloor)$. The circulant graph $C_{n+k,T}$ is $(k, D_{n,S})$-tolerant.

Proof: We will show the existence of an embedding ϕ that maps the nodes of $D_{n,S}$ to the healthy nodes in $C_{n+k,T}$. Let P, F, and H be as defined earlier, with P representing the nodes in $C_{n+k,T}$, F representing an arbitrary set of k faulty nodes, and H representing the remaining n healthy nodes. From Lemma 4.2, there exists a node $h_c \in H$ such that for all x, where $1 \leq x \leq y, marked(x) \subseteq tail(h_c, x)$. Let $h_z = (h_c + 1) \bmod (n+k)$. Define the function ϕ that maps from $[n]$ to $[n+k]$ such that for any $i \in [n], \phi(i) = h_{(z+i) \bmod n}$.

We will show that ϕ is an embedding of $D_{n,S}$ into the healthy nodes in $C_{n+k,T}$. It is clear that for any node i in $D_{n,S}, \phi(i)$ is a healthy node in $C_{n+k,T}$. In addition, it is clear that for any distinct nodes i and j in $D_{n,S}, \phi(i) \neq \phi(j)$. Thus all that remains to be shown is that for any edge (i, j) in $D_{n,S}, (\phi(i), \phi(j))$ is an edge in $C_{n+k,T}$. Every edge in $D_{n,S}$ is of the form $(a, a+x)$, where $a \in [n-x]$ and $x \in S$. We will show that $(\phi(a), \phi(a+x))$ is an edge in $C_{n+k,T}$. Note that $\phi(a) = h_{(z+a) \bmod n}$, and because $a \in [n-x], \phi(a) \notin tail(h_c, x)$. However, h_c was selected so that $marked(x) \subseteq tail(h_c, x)$, so it follows that $|jumps(\phi(a), x)| \leq \lfloor k/2 \rfloor$ and $x + |jumps(\phi(a), x)| \in T$. Therefore, $(\phi(a), \phi(a+x))$ is an edge in $C_{n+k,T}$. □

A. d-Dimensional Meshes and Hypercubes

The previous theorem on diagonal graphs can be used to construct efficient fault-tolerant d-dimensional meshes and hypercubes.

Lemma 4.4: Let M be an $n_0 \times n_1 \times \cdots \times n_{d-1}$ d-dimensional mesh, let $n = \Pi_{i=0}^{d-1} n_i$, and for all $i \in [d]$ let $s_i = \Pi_{j=0}^{i-1} n_j$ (thus $s_0 = 1$). The graph M is a subgraph of the diagonal graph $D_{n,S}$, where $S = \{s_0, s_1, \cdots, s_{d-1}\}$.

Proof: Let $\phi(a_0, a_1, \cdots, a_{d-1}) = \Sigma_{i=0}^{d-1}(a_i)(s_i)$. It is straightforward to verify that ϕ defines an embedding of M into $D_{n,S}$. □

Theorem 4.5: Let M be an $n_0 \times n_1 \times \cdots \times n_{d-1}$ d-dimensional mesh, let $n = \Pi_{i=0}^{d-1} n_i$, for all $i \in [d]$ let $s_i = \Pi_{j=0}^{i-1} n_j$, let $S = \{s_0, s_1, \cdots, s_{d-1}\}$, let k be a positive integer, and let $T = expand(S, \lfloor k/2 \rfloor)$. The circulant graph $C_{n+k,T}$ is (k, M)-tolerant and has degree at most $(k+2)d$ if k is even, and at most $(k+1)d$ if k is odd.

Proof: From Lemma 4.4, the graph M is a subgraph of $D_{n,S}$. We will consider two cases based on the value of n_{d-1}. If $n_{d-1} \geq 3$, then $S \subseteq \{1, 2, \cdots, \lfloor n/3 \rfloor\}$. Therefore, it follows from Theorem 4.3 that $C_{n+k,T}$ is $(k, D_{n,S})$-tolerant, which implies that $C_{n+k,T}$ is (k, M)-tolerant.

Conversely, if $n_{d-1} = 2$ let $S' = S \backslash \{n/2\}$ and note that $S' \subseteq \{1, 2, \cdots, \lfloor n/3 \rfloor\}$. Therefore, it follows from Theorem 4.3 that $C_{n+k,T}$ is $(k, D_{n,S'})$-tolerant, so there exists a function ϕ that is an embedding of $D_{n,S'}$ into the healthy nodes in $C_{n+k,T}$. We will show that ϕ is also an embedding of $D_{n,S}$ into the healthy nodes in $C_{n+k,T}$. It is clear that each edge in $D_{n,S}$ of the form $(i, i+x)$, where $x \in S'$, is mapped to an edge $(\phi(i), \phi(i+x))$ in $C_{n+k,T}$. Thus all that remains to be shown is that each edge in $D_{n,S}$ of the form $(i, i+n/2)$ is mapped to an edge $(\phi(i), \phi(i+n/2))$ in $C_{n+k,T}$. But $\{n/2, n/2+1, \cdots, n/2+\lfloor k/2 \rfloor\} \subseteq T$, so $\{n/2, n/2+1, \cdots, n/2+k\} \subseteq close(T, n+k)$, and no matter how many faulty nodes are located between $\phi(i)$ and $\phi(i+n/2)$, the edge $(\phi(i), \phi(i+n/2))$ is in $C_{n+k,T}$.

Finally, note that $|S| \leq d$ and $T = expand(S, \lfloor k/2 \rfloor)$, so $|T| \leq d(1 + \lfloor k/2 \rfloor)$. Therefore, the degree of $C_{n+k,T}$ is at most $(k+2)d$ if k is even and at most $(k+1)d$ if k is odd. □

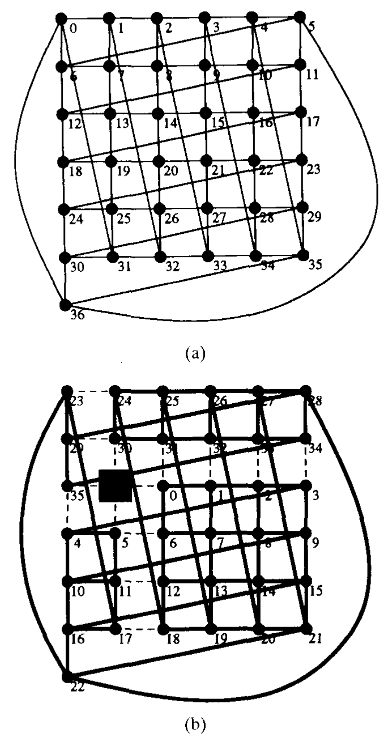


Fig. 5. (a) A 6×6 one-fault-tolerant two-dimensional mesh. (b) New labeling of the fault-tolerant mesh when the node with original label 13 is faulty.

Corollary 4.6: Let Q_d be a d-dimensional hypercube, let $n = 2^d$, for all $i \in [d]$ let $s_i = 2^i$, let $S = \{s_0, s_1, \cdots, s_{d-1}\}$, and let $T = expand(S, \lfloor k/2 \rfloor)$. The circulant graph $C_{n+k,T}$ is (k, Q_d)-tolerant and has degree at most $(k+2)d - k - 1$ if k is even, and at most $(k+1)d - k + 1$ if k is odd. In particular, the circulant graph $C_{n+k,T}$ has degree at most $(k+2)d - (k+2)\log k + 2k - 3$ when k is a power of 2.

Corollary 4.7: Let r and c be integers greater than or equal to 2, let $M_{r,c}$ be a two-dimensional mesh with r rows and c columns, let $n = rc$, let $S = \{1, c\}$, let k be a positive integer, and let $T = expand(S, \lfloor k/2 \rfloor)$. The circulant graph $C_{n+k,T}$ is $(k, M_{r,c})$-tolerant and has degree at most $2k+4$ if k is even, and at most $2k+2$ if k is odd.

It is helpful to examine some specific examples of the preceding general construction. First, consider the important case where the target graph M is an $m \times m$ two-dimensional mesh and $k = 1$ fault must be tolerated. In this case the fault-tolerant graph is a circulant graph with $m^2 + 1$ nodes and offsets 1 and m. An example is shown in Fig. 5(a), where it is assumed that $m = 6$. The graph has 37 nodes, the offsets are 1 and 6, and the connections are calculated using modulo-37

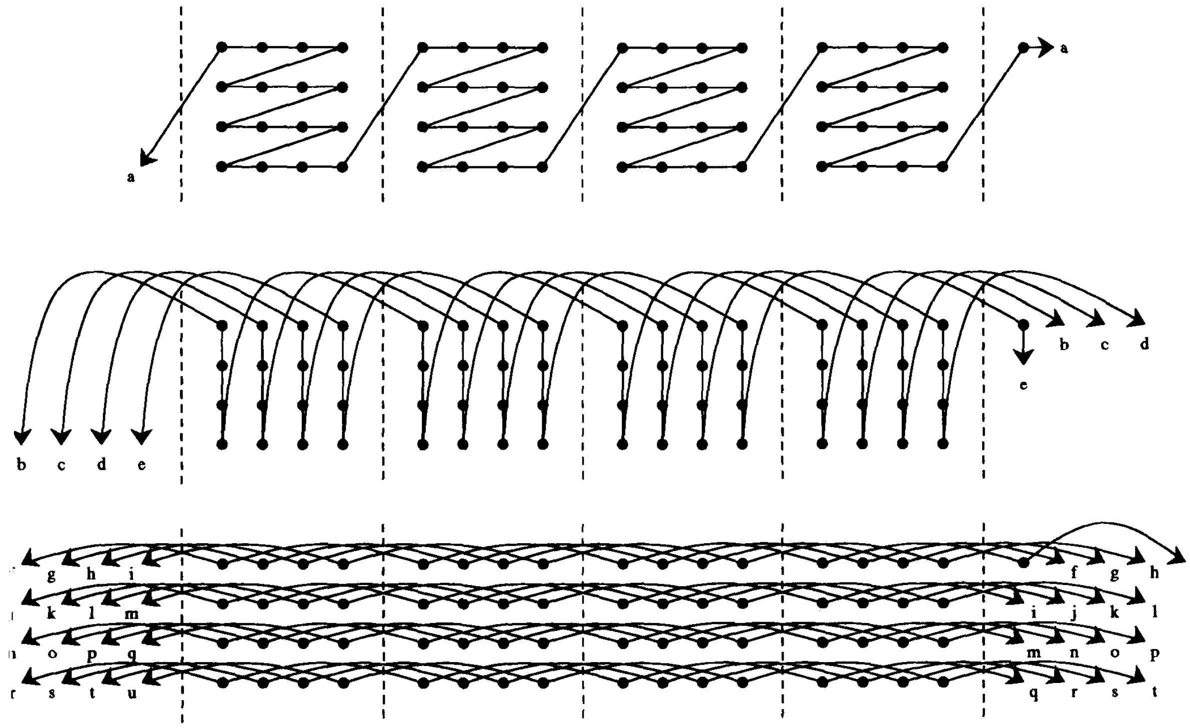


Fig. 6. A $4 \times 4 \times 4$ one-fault-tolerant three-dimensional mesh partitioned according to three types of edges.

arithmetic. As another example, consider the case where M is an $m \times m \times m$ three-dimensional mesh and $k = 1$ fault must be tolerated. In this case, the fault-tolerant graph is a circulant graph with $m^3 + 1$ nodes and offsets $1, m$, and m^2 (see Fig. 6, in which the edges corresponding to the three different offsets are shown separately for clarity). It will be shown in Section IV-C-1 that fault-tolerant two- and three-dimensional meshes can be laid out efficiently in two and three dimensions, respectively.

Another interesting example is where M is a d -dimensional hypercube and $k = 1$ fault must be tolerated. In this case, the fault-tolerant graph is a circulant graph with $2^d + 1$ nodes and offsets $1, 2, 4, \cdots, 2^{d-1}$. Finally, consider the case where M is an $m \times m$ two-dimensional mesh and $k = 3$ faults must be tolerated. In this case, the fault-tolerant graph is a circulant graph with $m^2 + 3$ nodes and offsets $1, 2, m$, and $m + 1$. Although both of these last two constructions yield graphs that require twice the degree of the target graph, it will be shown in Sections IV-C-2 and IV-C-3 that the actual implementations can have a much smaller degree.

B. Renaming Algorithm

When the target graph is circulant and Theorem 3.1 is used to create a fault-tolerant graph, the problem of locating a healthy target graph in the fault-tolerant graph is relatively simple. Any healthy node can be selected to play the role of node 0, and the ith healthy node following the selected node plays the role of node i. However, when the target graph is diagonal and Theorem 4.3 is used to create a fault-tolerant graph, the location of a healthy target graph is more involved. In particular, only certain healthy nodes can be selected to play the role of node 0. We will now present an efficient algorithm for locating a fault-free d -dimensional mesh contained in the fault-tolerant graph defined in Theorem 4.5. We will need the following definition to present the algorithm.

Definition: An ordering of the nodes in a d-dimensional mesh M is a *row-major order* if for each pair of nodes $a = (a_0, a_1, \cdots, a_{d-1})$ and $b = (b_0, b_1, \cdots, b_{d-1})$, a precedes b whenever there is some j such that $a_i = b_i$ for $0 \le i < j$ and $a_j < b_j$. Thus a row-major order is simply a lexicographic order of the nodes according to their positions. A *row-major labeling* of the nodes in an n-node mesh assigns the labels $0, 1, \cdots, n-1$ to the nodes in row-major order.

Let M be the desired healthy mesh and let $\hat{M}_k$ denote the fault-tolerant circulant graph $C_{n+k,T}$ as defined in Theorem 4.5. The Renaming Algorithm assigns the labels $0, 1, \cdots, n-1$ to the n healthy nodes in the fault-tolerant graph $\hat{M}_k$. These labels correspond to the row-major labeling of a healthy mesh M contained in $\hat{M}_k$.

Renaming Algorithm: The input to the Renaming Algorithm is a fault-tolerant graph $\hat{M}_k$ as defined earlier, with a set of at most k faulty nodes. We will assume that exactly k of the nodes are faulty, because if there are $x < k$ faults we can arbitrarily select any $k - x$ healthy nodes and consider them to be faulty. Recall that the nodes in $\hat{M}_k$ are numbered 0 through $n+k-1$. These nodes will be viewed as being ordered cyclically, with nodes $n + k - 1$ and 0 being adjacent. Thus, when the nodes are traversed in ascending order node 0 follows node $n+k-1$, and when they are traversed in descending order node $n+k-1$ follows node 0. In the following description, let $y = \lceil n/3 \rceil$. The Renaming Algorithm consists of three steps.

1) The first step uses two counters, one to count faulty nodes and one to count healthy nodes. The following routine is performed for all values of i where $0 \leq i \leq n+k-1$. First, both counters are set to 0. Then the nodes are visited in ascending order, starting with node i. As each node is visited, the appropriate counter is incremented. That is, if the visited node is faulty the counter for faulty nodes is incremented, and if the visited node is healthy the counter for healthy nodes is incremented. The counter for healthy nodes is checked after it is incremented. If this counter is greater than y, the process of visiting the nodes in ascending order is terminated, and the counter for faulty nodes is checked. If the counter for faulty nodes is greater than $k/2$, node i is designated as being "marked," whereas if it is less than or equal to $k/2$, node i is designated as being "unmarked."
2) The second step figures out which healthy node should play the role of node 0 in the healthy mesh. The second step uses a single counter and it consists of three phases. Phase 1 begins by setting the counter to 0. Then the nodes are visited in descending order, starting with any arbitrarily selected node. As each node is visited, the node is checked to see whether it is faulty and whether it is marked. There are three cases that are possible:
   a) If the node is healthy and unmarked, the counter is incremented.
   b) If the node is healthy and marked, the counter is reset to 0.
   c) If the node is faulty, the counter is left unchanged.
3) Next, the counter is checked and Phase 1 is terminated if the counter is greater than or equal to y. We will call the node that is being visited when the counter reaches y node d. Phase 2 then visits the nodes in descending order beginning with node d. It terminates when it encounters a healthy node that is marked. This healthy marked node will be called node c. Phase 3 then visits the nodes in ascending order beginning with node c. It terminates when it encounters an unmarked healthy node, which will be called node z.
4) The third step then assigns numbers to the healthy nodes. The nodes are visited in ascending order, starting with node z, and the healthy nodes are assigned the values $0, 1, \cdots, n-1$ in order. Thus node z is assigned 0, the next healthy node that is visited is assigned 1, and the last healthy node that is visited is assigned $n-1$. These numbers correspond to the row-major labels of a healthy mesh.

Notice that in the case of a single fault, the preceding algorithm will result in a new labeling that starts immediately after the fault. For example, consider the fault-tolerant mesh in Fig. 5(a) and assume that node 13 is faulty. Fig. 5(b) presents the new labeling of the mesh. The edges of the new mesh are highlighted with thick lines and the unused edges are represented by dashed lines.

Theorem 4.8: The Renaming Algorithm presented earlier correctly labels the healthy nodes in the fault-tolerant graph according to a row-major labeling of the nodes in the target mesh.

Proof: It is straightforward to verify that the first step of the Renaming Algorithm marks exactly those healthy nodes that are in the set $marked(y)$ (recall that $y = \lceil n/3 \rceil$). Then Phase 1 of the second step finds a block of y consecutive (ignoring faulty nodes) unmarked healthy nodes. Such a block is guaranteed to exist, because from Lemma 4.2 all of the marked healthy nodes are located in a block of y consecutive (ignoring faulty nodes) healthy nodes. Phase 2 of the second step then finds a marked healthy node followed by at least y unmarked healthy nodes and labels this node c. It is clear that this node c must correspond to the node h_c defined in the proof of Theorem 4.3, because all other marked healthy nodes have another marked healthy node within the following y healthy nodes. Phase 3 of the second step then labels a node z that must correspond to the node h_z defined in the proof of Theorem 4.3. Finally, the third step labels the healthy nodes in order from node z. It was shown in the proof of Theorem 4.5 that this labeling corresponds to a row-major labeling of the nodes in the target mesh. □

It is easy to verify that the Renaming Algorithm, as presented earlier, requires $\Theta(n(n+k))$ time. However, note that steps 2 and 3 require only $O(n+k)$ time. In addition, note that step 1 simply calculates the number of faulty nodes between each node i and the yth healthy node following i. This calculation could be performed in $O(n+k)$ time as well by noting that the calculations for successive healthy nodes i and j differ only by the number of faults between i and j and by the number of faults between the yth healthy node following i and the yth healthy node following j. Thus, the Renaming Algorithm can be modified to run in $O(n+k)$ time, which is optimal (as every node may require a new label).

C. Efficient Implementations

Many of the constructions for fault-tolerant meshes and hypercubes given by Theorem 4.5 can be implemented efficiently. First, we will show how the fault-tolerant two- and three-dimensional meshes can be laid out in two and three dimensions, respectively, using only short wires. We will then show how multiplexers and buses can be used to reduce the degree of the fault-tolerant graphs.

1) Layouts with Short Wires: When considering layouts with short wires, we will assume that the processors are arranged in a two- or three-dimensional array, and we will consider lengths in terms of the Manhattan distance between processors (ignoring the area or volume required by the wires). However, these constructions also yield efficient layouts in terms of area and volume.

It is clear from Fig. 5(a) that the fault-tolerant construction for two-dimensional meshes given by Corollary 4.7 are very closely related to torus networks. In fact, when $k = 0$ the construction yields what is known as a "singly twisted torus"

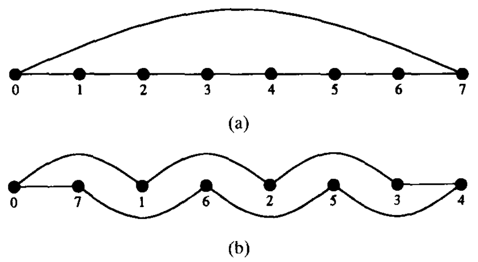


Fig. 7. Layout of a cycle using (a) normal ordering and (b) interleaved ordering.

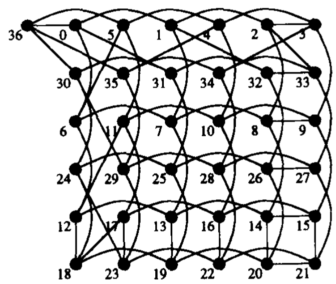


Fig. 8. Layout of 6 × 6 one-fault-tolerant two-dimensional mesh.

[18]. As a result, known techniques for laying out torus networks with short wires can be used for the fault-tolerant constructions.

The key idea for obtaining these short connections is to interleave the first and second halves of each row and column. This idea can be exemplified by considering a cycle of m nodes that are labeled from 0 to $m-1$. Clearly, if we lay out the nodes on a line according to this labeling, we will have $m-1$ short edges (of length 1) and one long edge (of length $m-1$) between node 0 and node $m-1$ [see Fig. 7(a)]. However, if we place the nodes in the order $0, m-1, 1, m-2, 2, \cdots, \lfloor m/2 \rfloor$, then each edge is of length at most 2 [see Fig. 7(b)]. Formally, we can define a function

$$\phi_1(i,m) = \begin{cases} 2i, & \text{if } i \le \lfloor \frac{m}{2} \rfloor, \\ 2(m-i)-1, & \text{otherwise.} \end{cases}$$

Then node i in a cycle of m nodes is mapped to position $\phi_1(i,m)$. We will call this ordering of the indices the *interleaved ordering*. It is easy to verify the following two properties:

Property 1: $mdist(\phi_1(i,m), \phi_1(i+1,m)) \le 2$ for all $0 \le i \le m-2$, and

Property 2: $mdist(\phi_1(m-1,m), \phi_1(0,m)) = 1$,

where $mdist(i,j)$ returns the Manhattan distance between two grid points i and j.

Layout for one-fault-tolerant two-dimensional mesh: Assume that we have an $m \times m$ mesh with one additional spare node.

1) Label the elements in each row in the $m \times m$ mesh from 0 to $m-1$. Label the rows from 0 to $m-1$.
2) Lay out the elements in each row according to the interleaved ordering.
3) Lay out the rows according to the interleaved ordering.
4) Place the spare node next to the upper left corner of the mesh layout.

See Fig. 8 for an example of a layout of a 6 × 6 one-fault-tolerant mesh. Formally, let $x(i,m) = i \bmod m$ and let $y(i,m) = \lfloor i/m \rfloor$. Then, node i, where $0 \le i \le m^2$, is mapped to a position

$$\phi_2(i,m) = \begin{cases} (\phi_1(x(i,m),m)+1, \phi_1(y(i,m),m)), \\ \qquad \text{if } i < m^2, \\ (0,0), \qquad \text{if } i = m^2, \end{cases}$$

in a two-dimensional layout. In the figure, the first axis corresponds to the horizontal direction and the second axis corresponds to the vertical direction with the upper leftmost position being (0, 0). We now prove the following lemma.

Lemma 4.9: The layout for the one-fault-tolerant two-dimensional mesh has edges of length at most 3 (using Manhattan distance).

Proof: First, it is easy to prove that the four edges of the last node (i.e., node m^2) are of length at most 3 in the layout. Thus, we now consider lengths (with respect to the layout) of the edges in the remaining m^2 node graph. For convenience, we refer to X-edges as the edges with offset 1 and Y-edges as those with offset m in the graph. Further partition X-edges into internal edges and wraparound edges with the latter edges being of the form $(i, i+1)$ where $(i+1) \bmod m = 0$. [For instance, edges (5, 6), (11, 12), $\cdots$, (29, 30) in Fig. 5(a) are wraparound X-edges.] Similarly, partition the Y-edges into internal edges and wraparound edges with the latter edges being of the form $(i, (i+m) \bmod (m^2+1))$ where $m(m-1) < i < m^2$. [For instance, edges (31, 0), (32, 1), $\cdots$, (35, 4) in Fig. 5(a) are wraparound Y-edges.] It is easy to see that all internal X-edges and internal Y-edges are of length at most 2 in the layout (by Property 1). The length of a wraparound X-edge $(i, i+1)$ is no more than the sum of the lengths of edges $(i, i+1-m)$ and $(i+1-m, i+1)$. Since the former is of distance 1 (by Property 2) and the latter is of distance at most 2 (by Property 1), the total distance is at most 3. The length of wraparound Y-edges can be similarly derived to be at most 3. □

For three-dimensional meshes we present an efficient three-dimensional layout.

Layout for one-fault-tolerant three-dimensional mesh: Assume that we have an $m \times m \times m$ mesh with one additional spare node.

1) Label the elements in each row from 0 to $m-1$. Label the rows in each $m \times m$ plane from 0 to $m-1$. Label the two-dimensional $m \times m$ planes from 0 to $m-1$.
2) Lay out the elements in each row according to the interleaved ordering.
3) Lay out the rows in each plane according to the interleaved ordering.
4) Lay out the planes according to the interleaved ordering.
5) Place the spare node next to the corner of the mesh layout that is the node labeled zero according to row-major ordering.

See Fig. 9 for an example of a layout of a 4 × 4 × 4 one-fault-tolerant mesh. Formally, let $x(i,m) = i \bmod m$, let

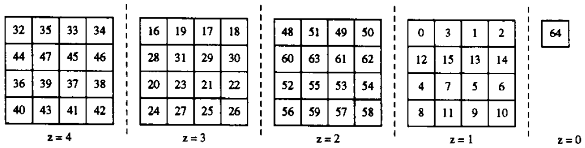


Fig. 9. Layout of $4 \times 4 \times 4$ one-fault-tolerant three-dimensional mesh.

$y(i,m) = (\lfloor i/m \rfloor) \bmod m$, and let $z(i,m) = \lfloor i/m^2 \rfloor$. Then, node i, where $0 \le i \le m^3$, is mapped to a position

$$\phi_3(i,m) = \begin{cases} (\phi_1(x(i,m),m), \phi_1(y(i,m),m), \\ \quad \phi_1(z(i,m)+1,m)), & \text{if } i < m^3, \\ (0,0,0), & \text{if } i = m^3, \end{cases}$$

in a three-dimensional layout. We now prove the following lemma.

Lemma 4.10: The three-dimensional layout for the one-fault-tolerant three-dimensional mesh has edges of length at most 4 (using Manhattan distance).

Proof: First, we list a few properties related to the two-dimensional layout using interleaved ordering, which can be derived from Properties 1 and 2 and the definition of $\phi_2(i,m)$. For brevity, we omit the second argument of ϕ_2 in the following, which is always m. *Property 3:* $mdist(\phi_2(0), \phi_2(m^2-1)) = 2$; *Property 4:* $mdist(\phi_2(0), \phi_2(m(m-1))) = 1$; *Property 5:* $mdist(\phi_2(i), \phi_2(i+m(m-1)+1)) \le 3$ for all $0 \le i \le m-1$; and *Property 6:* $mdist(\phi_2(i), \phi_2(i+1)) \le 3$ (from Lemma 4.9).

It is easy to prove that the six edges of the last node (i.e., node m^3) are of length at most 4. We now consider the remaining edges in the graph according to three different offsets: X-edges (offset 1), Y-edges (offset m), and Z-edges (offset m^2). There are two types of X-edges: internal edges are those within the same plane, and external edges are those between different planes. Clearly, internal edges are of length at most 3 (Property 6). External edges are of lengths at most 2 + 2 (Property 3 and Property 1, respectively).

There are three types of Y-edges: internal edges are those within the same plane, external edges are those between planes except for that between the last and the first planes, and wraparound edges are those between the last and first planes. Internal Y-edges are of length at most 3 (Property 6). External Y-edges are of length at most 1 + 2 (Property 4 for the former and Property 1 for the latter). Wraparound Y-edges are of length at most 1 + 3 (Property 2 for the former and Property 5 for the latter).

There are two types of Z-edges: wraparound edges are those between the last and first planes, and all other edges are internal edges. Internal Z-edges are of length at most 2 (Property 1). Wraparound Z-edges are of length at most 1 + 3 (Property 2 for the former and Property 6 for the latter). □

Although we have considered only layouts for the case where $k = 1$, it is possible to generalize these layouts for larger values of k. In particular, a k-fault-tolerant $m \times m$ mesh can be layed out in two dimensions as follows. First, label a row of m processors using the interleaved ordering. Next, take the processors from this row, in order from left to right, and lay them in a $\sqrt{k} \times m/\sqrt{k}$ array in column-major order. Thus, the first $\sqrt{k}$ processors are placed in the first column, the next $\sqrt{k}$ processors are placed in the second column, etc. This completes the layout of processors 0 through $m-1$. Now repeat this process until m such $\sqrt{k} \times m/\sqrt{k}$ arrays have been formed, and place these arrays one above the other in interleaved order. This completes the layout of processors 0 through m^2-1. Finally, place the k spare processors in a $\sqrt{k} \times \sqrt{k}$ square next to the upper left corner of the mesh layout. It is straightforward to verify that all of the wires are of length $O(\sqrt{k})$ (using the Manhattan distance). A similar approach can be used to obtain three-dimensional layouts of k-fault-tolerant three-dimensional meshes with wires of length $O(k^{1/3})$. Thus the fault-tolerant constructions retain many of the favorable properties of meshes.

2) Implementations with Multiplexers: The one-fault-tolerant graph for the d-dimensional hypercube presented in Section IV has degree $2d$. This is because each node j is connected to both node $(j+2^i) \bmod (n+1)$ and node $(j-2^i) \bmod (n+1)$, where $0 \le i < d$. However, only one out of each of these pairs of connections actually will be used once a healthy hypercube has been found. As a result, a 2-to-1 multiplexer can be used to connect processor j to the pair of processors $(j+2^i) \bmod (n+1)$ and $(j-2^i) \bmod (n+1)$. This reduces the degree of the fault-tolerant architecture to d, which is equal to the degree of the target graph. Note that the multiplexers do not have to be assumed to be immune to faults, as a faulty multiplexer can be avoided by treating the processor to which it is attached as being faulty. In addition, a similar technique of using 2-to-1 multiplexers to connect each processor j to pairs of processors of the form $(j+x) \bmod (n+k)$ and $(j-x) \bmod (n+k)$ can be used to reduce the degree of k-fault-tolerant hypercubes. However, it should be noted that the use of multiplexers does not reduce the number of wires.

3) Implementations with Buses: Finally, we can also use buses to reduce the degree of the fault-tolerant architectures. For example, when k is odd and the target graph is a two-dimensional mesh M with r rows and c columns, the fault-tolerant graph $\hat{M}_k$ given by Corollary 4.7 is a circulant graph with offsets $1, 2, \cdots, (k+1)/2$ and $c, c+1, \cdots, c+(k-1)/2$. However, for each healthy node i in $\hat{M}_k$, the target mesh M will use only at most one of the edges of the form $(i, (i+x) \bmod (n+k))$, where $1 \le x \le (k+1)/2$, and at most one of the edges of the form $(i, (i+x) \bmod (n+k))$, where $c \le x \le c+(k-1)/2$. As a result, we can use a bus to connect each node i in $\hat{M}_k$ to all of the nodes of the form $(i+x) \bmod (n+k)$, where $1 \le x \le (k+1)/2$, and we can use a separate bus to connect node i to all of the nodes of the form $(i+x) \bmod (n+k)$, where $1 \le x \le c+(k-1)/2$. An example of the case $k = 3$ is shown in Fig. 10. Note that the layout is quite compact and that most of the connections are short. Although a few of the connections appear to be relatively long in the figure, they are, in fact, of constant length and need not be made longer to create larger meshes.

A similar approach can also be used when k is even and when $d > 2$. In general, the use of buses results in a fault-tolerant architecture with degree at most $d(k+3)/2$ if k is odd and $d(k+4)/2$ if k is even. Note that if the bus connecting node i to some set of consecutive nodes of the

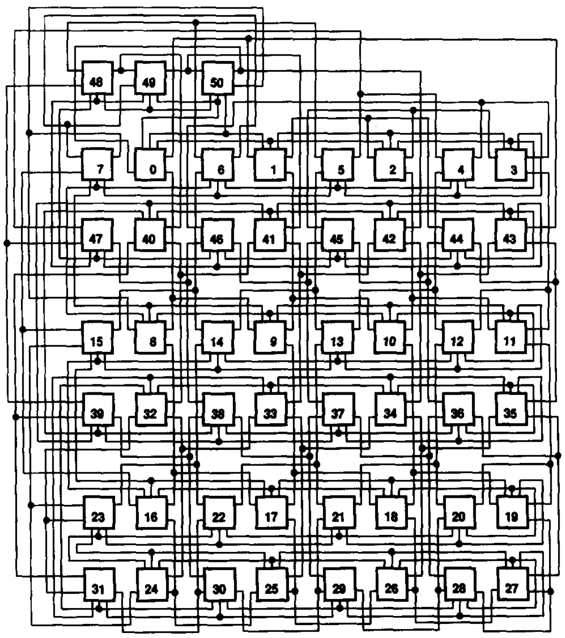


Fig. 10. Detailed layout of three-fault-tolerant two-dimensional mesh with 6 rows and 8 columns using buses.

form $(i + x) \bmod (n + k)$, where $x > 0$, is faulty, we can avoid using the bus by viewing node i as being faulty. Thus, even bus faults can be tolerated with this architecture.

V. Other Graphs

In this section we will present fault-tolerant graphs for target graphs that are tori, eight-connected meshes and hexagonal meshes.

A. Torus Construction

A torus with r rows and c columns, denoted $M^T_{r,c}$, is a mesh $M_{r,c}$ to which "wraparound" connections have been added that connect the first and last nodes in each row and the top and bottom nodes in each column. We will show that given any torus $M^T_{r,c}$, we can construct a $(k, M^T_{r,c})$-tolerant graph with $rc + k$ nodes and degree at most

$$\begin{cases} 2k+4, & \text{if } r \text{ and } c \text{ are relatively prime,} \\ 2k+6, & \text{if at least one of } r \text{ and } c \text{ is odd,} \\ 4k+6, & \text{if both } r \text{ and } c \text{ are even.} \end{cases}$$

1) Case 1—r and c are Relatively Prime: The construction of a fault-tolerant torus $M^T_{r,c}$ for which r and c are relatively prime is based on the *wraparound diagonal-major order*.

Lemma 5.1: Let r and c be relatively prime, let x be the integer satisfying $xc \bmod r = 1$ and $1 \le x < r$, and let $S = \{xc-1, xc\}$. The torus $M^T_{r,c}$ is a subgraph of the circulant graph $C_{rc,S}$.

Proof: Let $f(i) = xi \bmod r$ and let $\phi(i,j) = f(i-j)c + j$. The proof is analogous to that of Lemma 3.10 and will not be repeated here. □

An example of the numbering given by the function ϕ in the previous lemma is shown in Fig. 11(a).

0	21	7	28	14
15	1	22	8	29
30	16	2	23	9
10	31	17	3	24
25	11	32	18	4
5	26	12	33	19
20	6	27	13	34

0	19	2	21	4	23
24	1	26	3	28	5
6	25	8	27	10	29
30	7	32	9	34	11
12	31	14	33	16	35
36	13	38	15	40	17
18	37	20	39	22	41

0	15	2	17	4	19	34
35	1	37	3	39	5	20
21	36	23	38	25	40	6
7	22	9	24	11	26	41
42	8	44	10	46	12	27
28	43	30	45	32	47	13
14	29	16	31	18	33	48

Fig. 11. Three orderings of torus nodes: (a) $\gcd(r, c) = 1$, (b) r is odd and c is even, and (c) both r and c are odd and $r \le c$.

Theorem 5.2: Let r and c be relatively prime, let x be the integer satisfying $xc \bmod r = 1$ and $1 \le x < r$, let $S = \{xc-1, xc\}$, and let $T = expand(S, k)$. The circulant graph $C_{rc+k,T}$ is $(k, M^T_{r,c})$-tolerant and has degree at most $2k + 4$.

Proof: Follows from Theorem 3.1 and Lemma 5.1. □

Note that $gap(S, rc) = (r - 2x)c - 1$ if $x < r/2$, and $(2x - r)c - 3$ otherwise. It is possible to reduce the *gap* by two for the former case by doing an antidiagonal traversal instead of a diagonal traversal. As before, for $k \ge gap(S, rc)$, the additional edge per node is one for each additional fault (i.e., for each increment in k).

2) Case 2—r is Odd and c is Even: The construction of fault-tolerant torus $M^T_{r,c}$ for which r is odd and c is even is based on the *interleaved zigzag-major order* [see Fig. 11(b)].

Lemma 5.3: Let r be odd, let c be even, and let $S = \{(r-1)c/2 - 1, (r-1)c/2, (r-1)c/2 + 1\}$. The torus $M^T_{r,c}$ is a subgraph of the circulant graph $C_{rc,S}$.

Proof: Let $\phi(i,j) = [(([i - (j \bmod 2)]^{(r+1)/2}) \bmod r]c + j$. The proof is analogous to that of Lemma 3.10 and will not be repeated here. □

Theorem 5.4: Let r be odd, let c be even, let $S = \{(r-1)c/2 - 1, (r-1)c/2, (r-1)c/2 + 1\}$, and let $T = expand(S, k)$. The circulant graph $C_{rc+k,T}$ is $(k, M^T_{r,c})$-tolerant and has degree at most

$$\begin{cases} 2k+6, & \text{if } k \le c-3, \\ k+c+3, & \text{otherwise.} \end{cases}$$

Proof: The fact that the circulant graph $C_{rc+k,T}$ is $(k, M^T_{r,c})$-tolerant follows from Theorem 3.1 and Lemma 5.3. The degree follows from the fact that $gap(S, rc) = c-3$, $|S| = 3$, and Lemma 3.13. □

Note that if the same zigzag-style ordering is applied without interleaving between successive zigzag rows, the degree is still $2k + 6$ but the gap is much larger.

3) Case 3—Both r and c are Odd: Assume without loss of generality that $r \le c$. The construction is based on a hybrid method combining the wraparound diagonal-major order (of Case 1) with the zigzag-major order (similar to Case 2). See Fig. 11(c) as an example of labeling a 7×7 torus.

Lemma 5.5: Let r and c be odd integers with $r \le c$ and let $S = \{2c-1, 2c, 2c+1\}$. The torus $M^T_{r,c}$ is a subgraph of the circulant graph $C_{rc,S}$.

Proof: The zigzag ordering is applied to the $(r + 1)/2$ leftmost columns if $r \bmod 4 = 3$, and to the $(r-1)/2$ leftmost columns if $r \bmod 4 = 1$. The wraparound diagonal ordering is applied to the rest of the columns on the right.

Formally, let $f_1(i) = i(r-2) \bmod r$, let $f_2(i) = 2i \bmod r$, and let

$$\phi(i,j) = \begin{cases} f_1(i - (j \bmod 2))c + j, & \text{if } r \bmod 4 = 3 \text{ and } j < \frac{r+1}{2}, \\ f_1\left(i - j + \frac{r+1}{2}\right)c + j, & \text{if } r \bmod 4 = 3 \text{ and } j \geq \frac{r+1}{2}, \\ f_2(i - (j \bmod 2))c + j, & \text{if } r \bmod 4 = 1 \text{ and } j < \frac{r-1}{2}, \\ f_2\left(i - j + \frac{r-1}{2}\right)c + j, & \text{if } r \bmod 4 = 1 \text{ and } j \geq \frac{r-1}{2}. \end{cases}$$

It is straightforward to verify that ϕ defines an embedding of $M^T_{r,c}$ into $C_{rc,S}$. □

Theorem 5.6: Let r and c be odd and $r \leq c$, let $S = \{2c-1, 2c, 2c+1\}$, and let $T = expand(S,k)$. The circulant graph $C_{rc+k,T}$ is $(k, M^T_{r,c})$-tolerant and has degree at most $2k+6$.

Proof: The fault tolerance of $C_{rc+k,T}$ follows from Theorem 3.1 and Lemma 5.5. Since $T = \{2c-1, 2c, \cdots, 2c+k+1\}$, the degree of $C_{rc+k,T}$ is $|close(T, rc+k)| = 2k+6$. □

4) Case 4—Both r and c are Even: In this case, we simply use *row-major order* as used in Section III-A for the first fault-tolerant mesh construction. The proof of the following lemma is analogous to that of Lemma 3.2 and will not be included.

Lemma 5.7: Let $S = \{1, c-1, c\}$. The torus $M^T_{r,c}$ is a subgraph of the circulant graph $C_{rc,S}$.

Theorem 5.8: Let $S = \{1, c-1, c\}$ and let $T = expand(S,k)$. The circulant graph $C_{rc+k,T}$ is $(k, M^T_{r,c})$-tolerant and has degree at most $4k+6$.

Proof: Follows from Theorem 3.1 and Lemma 5.7. □

Note that Theorem 5.8 does not require that r and c have any special properties. Finally, it should be noted that some of the constructions for fault-tolerant meshes can also be used to add fault tolerance to twisted torus networks [18]. For example, Mesh Construction 2 presented in Theorem 3.5 yields a degree $2k+4$ fault-tolerant singly twisted torus when $r = c$.

B. Eight-Connected Meshes

An eight-connected mesh with r rows and c columns, denoted $M^8_{r,c}$, is a mesh $M_{r,c}$ to which connections between nodes that are diagonal or antidiagonal neighbors have been added. We will use *row-major order* to construct its fault-tolerant graph. The proofs are analogous to those of the previous section and are omitted.

Lemma 5.9: Let $S = \{1, c-1, c, c+1\}$. The eight-connected mesh $M^8_{r,c}$ is isomorphic to the diagonal graph $D_{rc,S}$.

Theorem 5.10: Let $S = \{1, c-1, c, c+1\}$ and let $T = expand(S, \lfloor k/2 \rfloor)$. If $r > 3$, the circulant graph $C_{rc+k,T}$ is $(k, M^8_{r,c})$ tolerant and has degree at most $2k+6$ if k is odd and $2k+8$ if k is even.

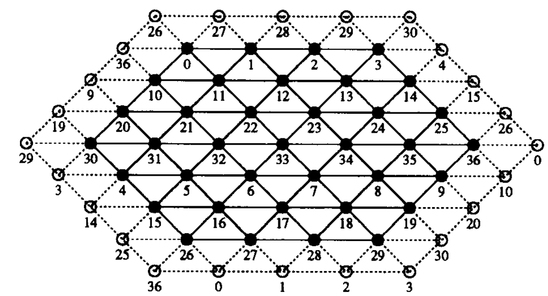


Fig. 12. Wraparound hexagonal mesh of order 4.

C. Hexagonal Meshes

A hexagonal mesh (H-mesh) of order c is a six-connected mesh with hexagonal boundary. Each node is connected to two horizontal neighbors, two diagonal neighbors, and two antidiagonal neighbors, if they exist. The order is the length of one coordinate. Chen *et al.* [7] defined the wraparound connection of H-meshes, termed C-type wrapping, such that they become node symmetric graphs. In the C-type wrapping, the rightmost node at row i, where $0 \leq i < 2c-1$, is connected to the leftmost node at row $(i+c) \bmod (2c-1)$. The same wrapping scheme is applied to two other coordinates after rotating the H-mesh. Fig. 12 shows the C-type wrapping H-mesh of order 4. Chen *et al.* [7] also showed the isomorphism between the C-type wrapping H-meshes and a family of circulant graphs (as described by the following lemma), which is useful in constructing the fault-tolerant graphs for H-meshes. In the following, we denote M^H_c the C-type H-mesh of order c and $N(c) = 3c^2 - 3c + 1$ the number of nodes in M^H_c.

Lemma 5.11 [7]: Let $S = \{1, 3c-2, 3c-1\}$ and let $N(c) = 3c^2 - 3c + 1$. The wraparound hexagonal mesh M^H_c is isomorphic to the circulant graph $C_{N(c),S}$.

Theorem 5.12: Let $S = \{1, 3c-2, 3c-1\}$, let $N(c) = 3c^2 - 3c + 1$, and let $T = expand(S,k)$. The circulant graph $C_{N(c)+k,T}$ is (k, M^H_c)-tolerant and has degree at most $4k+6$.

Note that an H-mesh of order c without wraparound is also a subgraph of a $(2c-1) \times (2c-1)$ eight-connected mesh. However, the latter has $c^2 - c$ more nodes than the former.

VI. Conclusion

We have presented new techniques for tolerating faults in d-dimensional meshes and hypercubes. The fault tolerance of the constructions relies on properties of circulant and diagonal graphs, many of which were derived herein. In particular, the construction given in Theorem 4.5 for a fault-tolerant d-dimensional mesh tolerates k faults and has degree at most $(k+1)d$ if k is odd and $(k+2)d$ if k is even. Thus this construction yields a one-fault-tolerant d-dimensional mesh that has only one spare node and degree $2d$. We also gave a renaming algorithm for locating a healthy mesh in the presence of faults, and efficient layouts (with very short edges) for fault-tolerant two- and three-dimensional meshes. In addition, we showed how multiplexers and buses can be used to reduce the degree of the fault-tolerant architectures. Finally, we showed how similar techniques can be used to obtain fault-tolerant tori, eight-connected meshes and hexagonal meshes.

APPENDIX

This appendix presents the proof of Theorem 3.8. The proof requires the following definitions and lemmas.

Definitions: Any node (i,j) is a mesh $M_{r,c}$ will be called *even* if $i+j$ is even, and *odd* otherwise. The graph $L_{r,c}$ consists of the $\lceil rc/2 \rceil$ even nodes in $M_{r,c}$. Any pair of nodes (i_1,j_1) and (i_2,j_2) in $L_{r,c}$ are adjacent iff $|i_1-i_2|+|j_1-j_2|=2$. Note that any pair of nodes adjacent in $L_{r,c}$ are connected by a path of length 2 in $M_{r,c}$. Given a set S of nodes in $L_{r,c}$, the *diagonal compression of* S, denoted $diag(S)$, is the set obtained by sliding the nodes in S upward along the diagonals until all gaps have been removed. Similarly, the *antidiagonal compression of* S, denoted $antidiag(S)$, is the set obtained by sliding the nodes in S upward along the antidiagonals until all gaps have been removed. More formal definitions of the diagonal and antidiagonal compression of S are as follows. For all i where $1-c \le i \le r-1$, the ith diagonal, denoted D_i, is the set $D_i = \{(x,y) \in M_{r,c} | x-y=i\}$. Any node (i,j) in $L_{r,c}$ is in $diag(S)$ if and only if either $i \le j$ and $i < |D_{i-j} \cap S|$ or $i > j$ and $j < |D_{i-j} \cap S|$. Similarly, for all i where $0 \le i \le r+c-2$, the ith antidiagonal, denoted A_i, is the set $A_i = \{(x,y) \in M_{r,c} | x+y=i\}$. Any node (i,j) in $L_{r,c}$ is in $antidiag(S)$ if and only if either $i+j<c$ and $i < |A_{i+j} \cap S|$ or $i+j \ge c$ and $c-1-j < |A_{i+j} \cap S|$. Given a graph G and a set S of nodes in G, the *neighbors of* S *in* G, denoted $neighb(S,G)$, is the set of nodes in G that are not in S but are adjacent to a node in S.

Lemma A1: Let r be any positive integer and let S be any set of nodes in $L_{r,r}$. Then $|neighb(diag(S),L_{r,r})| \le |neighb(S,L_{r,r})|$ and $|neighb(antidiag(S),L_{r,r})| \le |neighb(S,L_{r,r})|$.

Proof: It will be shown that $|neighb(diag(S),L_{r,r})| \le |neighb(S,L_{r,r})|$. The proof that $|neighb(antidiag(S,L_{r,r})| \le |neighb(S,L_{r,r})|$ is analogous and will not be given. The proof will first show that, within each diagonal, the diagonal compression operator can only decrease the number of neighbors. As a result, the overall number of neighbors can only decrease.

For notational simplicity, let $D_i = \emptyset$ for all i where either $i \le -r$ or $i \ge r$. For all i where $-r-1 \le i \le r+1$, let $S_i = S \cap D_i$ and let $S'_i = diag(S) \cap D_i$. Let a be any integer where $1-r \le a \le r-1$. Let $T_{-2} = D_a \cap neighb(S_{a-2},L_{r,r})$, let $T_0 = D_a \cap (neighb(S_a,L_{r,r}) \cup S_a)$, let $T_{+2} = D_a \cap neighb(S_{a+2},L_{r,r})$, and let $T = T_{-2} \cup T_0 \cup T_{+2}$. Similarly, let $T'_{-2} = D_a \cap neighb(S'_{a-2},L_{r,r})$, let $T'_0 = D_a \cap (neighb(S'_a,L_{r,r}) \cup S'_a)$, let $T'_{+2} = D_a \cap neighb(S'_{a+2},L_{r,r})$, and let $T' = T'_{-2} \cup T'_0 \cup T'_{+2}$.

First, it will be shown that $|T'_{-2}| \le |T_{-2}|$. There are three cases:

Case 1) $a \le 2-r$: In this case, $D_{a-2} = \emptyset$, so $T'_{-2} = T_{-2} = \emptyset$ and $|T'_{-2}| \le |T_{-2}|$.

Case 2) $3-r \le a \le 0$: In this case, $|T_{-2}| \ge |S_{a-2}|+2$ and $|T'_{-2}| = |S'_{a-2}|+2$. However, $|S_{a-2}| = |S'_{a-2}|$, so $|T'_{-2}| \le |T_{-2}|$.

Case 3) $2 \le a \le r-1$: In this case, if $|S_{a-2}| \ge |D_{a-2}|-1$ then $T_{-2} = D_a$ and $T'_{-2} \subseteq D_a$, so $|T'_{-2}| \le |T_{-2}|$. Conversely, if $|S_{a-2}| \le |D_{a-2}|-2$ then $|T_{-2}| \ge |S_{a-2}|$ and $|T'_{-2}| = |S'_{a-2}|$. However, $|S_{a-2}| = |S'_{a-2}|$, so $|T'_{-2}| \le |T_{-2}|$.

Next, we will show that $|T'_0| \le |T_0|$. If $S_a = D_a$ then $T'_0 = T_0 = D_a$ and $|T'_0| \le |T_0|$. Conversely, if $S_a \ne D_a$ then $|T_0| \ge |S_a|+1$ and $|T'_0| = |S'_a|+1$. However, $|S_a| = |S'_a|$, so $|T'_0| \le |T_0|$.

It can also be shown that $|T'_{+2}| \le |T_{+2}|$. The proof is analogous to the proof that $|T'_{-2}| \le |T_{-2}|$ and will not be repeated. Therefore, $|T'_{-2}| \le |T_{-2}|, |T'_0| \le |T_0|$, and $|T'_{+2}| \le |T_{+2}|$. Note that $T = T_{-2} \cup T_0 \cup T_{+2}$, so $|T| \ge \max(|T_{-2}|,|T_0|,|T_{+2}|)$. In addition, note that $|T'| = \max(|T'_{-2}|,|T_0|,|T'_{+2}|)$. As a result, $|T'| \le |T|$. Furthermore, $|D_a \cap neighb(S,L_{r,r})| \ge |T| - |S_a|$ and $|D_a \cap neighb(diag(S),L_{r,r})| = |T'| - |S'_a|$. However, $|S_a| = |S'_a|$, so $|D_a \cap neighb(diag(S),L_{r,r})| \le |D_a \cap neighb(S,L_{r,r})|$. Because a was chosen arbitrarily, the diagonal compression operation can only decrease the number of neighbors in each of the diagonals, and $|neighb(diag(S),L_{r,r})| \le |neighb(S,L_{r,r})|$. □

Lemma A2: Let r be any integer greater than 3 and let S be any set of $\lceil r^2/4 \rceil - 1$ nodes in $L_{r,r}$. Then $|neighb(S,L_{r,r})| \ge r$.

Proof: Define the infinite sequence of sets $S_0, S_1, \cdots$ such that $S = S_0$, for each $i, 0 \le i, S_{i+1} = antidiag(diag(S_i))$. Let h be the smallest nonnegative integer such that $S_h = S_{h+1}$ (such as h is guaranteed to exist, as both the diagonal and antidiagonal compression operators only move nodes upward in the mesh). Note that $S_h = diag(S_h) = antidiag(S_h)$. In addition, note that for any node $(i,j) \in S_h$, it follows that all nodes of the form (i',j'), where $i'-j' \le i-j$ and $i'+j' \le i+j$, are also in S_h.

Assume for the sake of contradiction that there exists a column j in $L_{r,r}$ such that all of the nodes in column j are in S_h. Let node (i,j) be the lowest node in column j of $L_{r,r}$ (thus, $i=r-1$ if $r+j$ is odd and $i=r-2$ if $r+j$ is even). Let T be the set of all nodes in $L_{r,r}$ of the form (i',j') where $i'-j' \le i-j$ and $i'+j' \le i+j$. Note that $T \subseteq S_h$. However, it is straightforward (but tedious) to show that $|T| > |S|$, which is a contradiction.

Next, assume for the sake of contradiction that there exists a pair of adjacent columns j and $j+1$ in $L_{r,r}$ such that none of the nodes in columns j and $j+1$ are in S_h. Let $a=j$ if j is even and let $a=j+1$ if j is odd. Let T be the set of all nodes in $L_{r,r}$ of the form (i',j'), where $i'-j' < -a$ or $i'+j' < a$. Because $(0,a) \in S_h$, it follows that $S_h \subseteq T$. However, it is straightforward (but tedious) to show that $|T| < |S|$, which is a contradiction.

Thus, there is no column in $L_{r,r}$ that is completely contained in S_h and there are no adjacent columns in $L_{r,r}$ that have no nodes in S_h. As a result, every column contains at least one node in $neighb(S_h,L_{r,r})$ and $|neighb(S_h,L_{r,r}) \ge r$. However, from Lemma A1, $|neighb(S_h,L_{r,r})| \le |neighb(S,L_{r,r})|$. As a result, $|neighb(S,L_{r,r})| \ge r$. □

Theorem 3.8: Let r and c be integers where $4 \le r \le c$ and let $C_{rc,S}$ be a circulant graph that contains the mesh $M_{r,c}$ as a subgraph. There exists an $s \in S$ such that $|s-(rc/2)| \ge (r+$

$1)/2$ if r is odd and c is even, and such that $|s-(rc/2)| \geq r/2$ otherwise.

Proof: First, we will show that in any case there exists an $s \in S$ such that $|s-(rc/2)| \geq r/2$. Assume for the sake of contradiction that for all $s \in S, |s-(rc/2)| < r/2$. Let $\phi(i,j)$ be an embedding of $M_{r,c}$ into $C_{rc,S}$. Now consider any two nodes (i_1,j_1) and (i_2,j_2) which are adjacent in $L_{r,r}$. Because there exists a path of length 2 in $M_{r,c}$ between (i_1,j_1) and $(i_2,j_2), dist(\phi(i_1,j_1),\phi(i_2,j_2),rc) \leq r-1$. Let $q = \lfloor r/2 \rfloor$ and assume without loss of generality that $\phi(q,q) = (r/2)(r-1)$ (no generality is lost because $C_{rc,S}$ is node-symmetric). Note that the distance in $L_{r,r}$ between (q,q) and any other node in $L_{r,r}$ is at most $r/2$. As a result, for any node (i,j) in $L_{r,r}, 0 \leq \phi(i,j) \leq 2(r/2)(r-1) = r^2 - r$. Therefore, given any two nodes (i_1,j_1) and (i_2,j_2) that are adjacent in $L_{r,r}, |\phi(i_1,j_1) - \phi(i_2,j_2)| \leq r-1$.

Let A be the set of $\lceil r^2/4 \rceil - 1$ nodes in $L_{r,r}$ that have the smallest ϕ values [formally, $|A| = \lceil r^2/4 \rceil - 1$ and for all $(i_1,j_1) \in A$ and $(i_2,j_2) \notin A, \phi(i_1,j_1) < \phi(i_2,j_2)$]. Let $B = neighb(A, L_{r,r})$. From Lemma A2, $|B| \geq r$, which implies that there exists a node $(i_1,j_1) \in A$ and a node $(i_2,j_2) \in B$ such that $|\phi(i_1,j_1) - \phi(i_2,j_2)| \geq r$, which is a contradiction. Thus, in all cases, there exists an $s \in S$ such that $|s-(rc/2)| \geq r/2$. Now consider the case in which r is odd and c is even. In this case, $(rc/2) - r/2$ and $(rc/2) + r/2$ are not integers, so there must exist an $s \in S$ such that $|s-(rc/2)| \geq (r+1)/2$. □

ACKNOWLEDGMENT

The authors would like to thank the anonymous referees for their helpful comments and suggestions.

REFERENCES

[1] F. Annexstein, "Fault tolerance in hypercube-derivative networks," in *Proc. 1st Annual ACM Symposium on Parallel Algorithms and Architectures,* 1989, pp. 179–188.

[2] V. Balasubramanian and P. Banerjee, "A fault-tolerant massively parallel processing architecture," *J. Parallel Distributed Computing,* vol. 4, pp. 363–383, 1987.

[3] K. E. Batcher, "Design of a massively parallel processor," *IEEE Trans. Comput.,* vol. C-29, no. 9, pp. 836–840, 1980.

[4] R. Beivide, E. Herrada, J. L. Balcázar, and J. Labarta, "Optimized mesh-connected networks for SIMD and MIMD architectures," in *Proc. 14th Annual Int. Symp. Computer Architecture,* 1987, pp. 163–170.

[5] J. Bruck, R. Cypher, and C. T. Ho, "On the construction of fault-tolerant cube-connected cycles networks," in *Proc. 1991 Int. Conf. on Parallel Processing,* vol. I, pp. 692–693.

[6] J. Bruck, R. Cypher, and D. Soroker, "Running algorithms efficiently on faulty hypercubes," in *Proc. 2nd Annual ACM Symp. on Parallel Algorithms and Architectures,* 1990, pp. 37–44.

[7] M.-S. Chen, K. G. Shin, and D. D. Kandlur, "Addressing, routing, and broadcasting in hexagonal mesh multiprocessors," *IEEE Trans. Comput.,* vol. C-39, no. 1, pp. 10–18, 1990.

[8] S. Dutt and J. P. Hayes, "On designing and reconfiguring k-fault-tolerant tree architectures," *IEEE Trans. Comput.,* vol. C-39, no. 4, pp. 490–503, 1990.

[9] S. Dutt and J. P. Hayes, "Designing fault-tolerant systems using automorphisms," *J. Parallel Distributed Computing,* vol. 12, pp. 249–268, 1991.

[10] S. Dutt and J. P. Hayes, "Some practical issues in the design of fault-tolerant multiprocessors," in *Proc. 21st Int. Symp. on Fault-Tolerant Computing,* 1991, pp. 292–299.

[11] B. Elspas and J. Turner, "Graphs with circulant adjacency matrices," *J. Combinatorial Theory,* no. 9, pp. 297–307, 1970.

[12] J. Hastad, F. T. Leighton, and M. Newman, "Fast computations using faulty hypercubes," in *Proc. 21st Annual ACM Symp. on Theory of Computing,* 1989, pp. 251–284.

[13] J. P. Hayes, "A graph model for fault-tolerant computing systems," *IEEE Trans. Comput.,* vol. C-25, no. 9, pp. 875–884, 1976.

[14] C. Kaklamanis, A. R. Karlin, F. T. Leighton, V. Milenkovic, P. Raghavan, S. Rao, C. Thomborson, and A. Tsantilas, "Asymptotically tight bounds for computing with faulty arrays of processors," in *Proc 31st Annual IEEE Symp. on Foundations of Computer Science,* pp 285–296, 1990.

[15] S. Y. Kung, S. N. Jean, and C. W. Chang, "Fault-tolerant array processors using single-track switches," *IEEE Trans. Comput.,* vol. C-38, no. 4, pp. 501–514, 1989.

[16] S-Y. Kuo and W. K. Fuchs, "Efficient spare allocation for reconfigurable arrays," *IEEE Design and Test,* pp. 24–31, 1987.

[17] F. T. Leighton and C. E. Leiserson, "Wafer scale integration of systolic arrays," *IEEE Trans. Computers,* vol. C-34, no. 5, pp. 448–461, 1985.

[18] A. J. Martin, "The torus: An exercise in constructing a processing surface," in *Proc. 2nd Caltech Conf. on VLSI,* 1981, pp. 527–537.

[19] M. Paoli, W. W. Wong, and C. K. Wong, "Minimum k-Hamiltonian graphs, II," *J. Graph Theory,* vol. 10, pp. 79–95, 1986.

[20] *Setting New Horizons.* Aachen, Germany: Parsytec Computer GmbH, 1991.

[21] A. L. Rosenberg, "The Diogenes approach to testable fault-tolerant VLSI processor arrays," *IEEE Trans. Comput.,* vol. C-32, no. 10, pp. 902–910, 1983.

[22] A. L. Rosenberg, "Routing with permuters: Toward reconfigurable and fault-tolerant networks," Tech. Rep. CS-1981-13, Dept. of Computer Science, Duke University, Durham, NC, 1981.

[23] A. L. Rosenberg, "On designing fault-tolerant VLSI processor arrays," *Advances in Computing Research,* vol. 2, pp. 181–204, 1984.

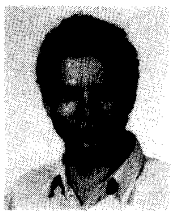
[24] V. P. Roychowdhury, J. Bruck, and T. Kailath, "Efficient algorithms for reconfiguration in VLSI/WSI arrays," *IEEE Trans. Comput.,* vol. C-39, no. 4, pp. 480–489, 1990.

[25] M. R. Samatham and D. K. Pradhan, "The de Bruijn multiprocessor network: A versatile network for parallel computation," *IEEE Trans. Comput.,* vol. 38, no. 4, pp. 567–581, 1989.

[26] M. Sami and R. Stefanelli, "Reconfigurable architectures for VLSI processing arrays," *Proc. IEEE,* vol. 74, no. 5, 1986.

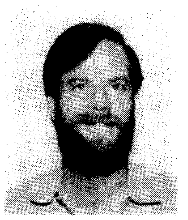
[27] Y. Ueoka, C. Minagawa, M. Oka, and A. Ishimoto, "A defect-tolerant design for full-wafer memory LSI," *IEEE J. Solid-State Circuits,* vol. SC-19, no. 3, 1984.

[28] W. W. Wong and C. K. Wong, "Minimum k-Hamiltonian graphs," *J. Graph Theory,* vol. 8, pp. 155–165, 1984.



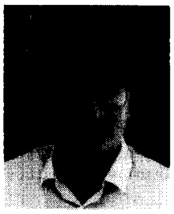
Jehoshua Bruck was born in Haifa, Israel, on April 19, 1956. He received the B.Sc. and M.Sc. degrees in electrical engineering from the Technion—Israel Institute of Technology, in 1982 and 1985, respectively, and the Ph.D. degree in electrical engineering from Stanford University in 1989.

From 1982 to 1985 he was with the IBM Haifa Scientific Center, Israel. In March 1989, he joined the IBM Research Division at the Almaden Research Center, San Jose, CA, where he currently manages the Foundations of Massively Parallel Computing Group. He is also a Consulting Assistant Professor in the Stanford University Electrical Engineering Department. Dr. Bruck's research interests include parallel computing, fault-tolerant computing, error-correcting codes, and neural networks.



Robert Cypher was born in Schenectady, NY, in 1959. He received the B.S. degree in mathematical sciences from Stanford University in 1982 and the M.S. and Ph.D. degrees in computer science from the University of Washington in 1987 and 1989, respectively.

He is currently a Research Staff Member of the IBM Almaden Research Center and a Consulting Assistant Professor in the Stanford University Computer Science Department. His research interests include parallel algorithms, VLSI, fault tolerance, and the design of interconnection networks.



Ching-Tien Ho received the B.S. degree in electrical engineering from National Taiwan University, Taiwan, in 1979 and the M.S., M.Phil., and Ph.D. degrees in computer science from Yale University, New Haven, CT, in 1985, 1986, and 1990, respectively.

Since August 1989, he has been a Research Staff Member in the Foundations of Massively Parallel Computing Group at IBM, Almaden Research Center. His primary research interests include communication issues for interconnection networks, graph embeddings, fault tolerance, parallel algorithms, and architectures.

Dr. Ho is a corecipient of the 1986 Outstanding Paper Award of the International Conference on Parallel Processing. He is a member of the Association for Computing Machinery and the IEEE Computer Society.