RELIABLE RECONFIGURABLE STRUCTURES FOR ARRAY ARCHITECTURES

Edwin Hsing-Mean Sha
Kenneth Steiglitz

CS-TR-316-91

April 1991

Reliable Reconfigurable Structures for Array Architectures †

Edwin Hsing-Mean Sha hms@cs.princeton.edu

Kenneth Steiglitz ken@cs.princeton.edu

Dept. of Computer Science

Princeton University

Princeton, NJ 08544

ABSTRACT

This paper describes some explicit constructions for reconfigurable array architectures. Given a working architecture (*application graph*), we add redundant hardware to increase reliability. The *degree of reconfigurability*, DR, of a redundant graph is a measure of the cost of reconfiguration after failures. When DR is independent of the size of the application graph, we say the graph is *finitely reconfigurable*, FR. We present a class of simple layered graphs with a logarithmic number of redundant edges, which can maintain both finite reconfigurability and a fixed level of reliability for a wide class of application graphs. By sacrificing finite reconfigurability, we show that by using expanders we can construct highly reliable structures with the asymptotically optimal number of edges for one-dimensional and tree-like array architectures.

†This work was supported in part by NSF Grant MIP-8912100, and U. S. Army Research Office-Durham Grant DAAL03-89-K-0074.

1 Introduction

The need for very high throughput in regular scientific computations has led to the study of highly parallel pipelined structures, such as linear arrays, trees, meshes, and multiple pipelines (see, for example, [HKu, SKu, ASAP]). As the degree of parallelism gets large, and the number of processors grows, the reliability problem becomes increasingly important. When high throughput applications involve real-time signal processing, the reliability issue becomes especially critical.

In most of the literature on fault tolerance [ChLeRo, KuLa, GrGa, SaSt], faults are confined to processing elements (PE's). When the number of switches and connections becomes large, this assumption becomes invalid, and in this paper we use a graph model that takes into account failures of switches and interconnection wires as well. PE's and switches will be represented by nodes of the graph in the obvious way, and a connection between two elements in the computational structure will be represented by a node inserted in the edge between the appropriate two nodes in the graph model. Each node of the graph will have associated with it a (uniform) probability of failure ε, and nodes fail independently.

To achieve fault tolerance, we add redundancy to the system. After a failure the original working architecture (*application graph*) is reconfigured by replacing some nodes that were being used by redundant nodes. We adopt the same model and definitions as we did in [ShSt], using the notion of *degree of reconfigurability*, DR, a measure of the cost of reconfiguration after failures have occurred. When DR is independent of the size of the system, it represents the situation when the amount of change necessary to repair the system depends only on the number of failed nodes, but not on the size of the system. In this case, we say the graph is *finitely reconfigurable.*

We will emphasize on-line reconfiguration of run-time errors, as in [KuJeCh], instead of fabrication-time reconfiguration [LeLe, GrGa]. This is accomplished by using a simple

layered graph. Nodes will be replaced only by nodes in the same layer. After definitions in section 2, we present in section 3 a practical structure based on layered graphs which can maintain both finite reconfigurability and any fixed level of reliability for any application graphs with bounded degree. This structure uses a linear number of nodes and a logarithmic number of redundant edges.

An interesting question is how much redundant hardware is needed to maintain reliable operation in the presence of failures. Lower and upper bounds are given in [DoCr1, DoCr2, Pi, PiStTs] for computing a Boolean function with noisy gates. They show that $\Theta(n \log n)$ noisy gates are necessary and sufficient for computing a Boolean function that can be computed with n noiseless gates. The previous work by Alon, Chung [AlCh], and Friedman and Pippenger [FrPi], gives strong theoretical results for maintaining reliability of linear arrays and trees, without giving explicit construction or reconfiguration algorithms after failures. Leighton and Leiserson [LeLe] also show how to embed linear arrays and two-dimensional meshes in two-dimensional redundant meshes at the fabrication level. We consider here the problem of designing reliable and general architectures which have the asymptotically optimal number of edges, and can at the same time be easily reconfigured.

In section 4, by sacrificing finite reconfigurability, we present fault-tolerant structures for one-dimensional pipelines and tree-like array architectures which are optimal in the sense of using only a linear number of nodes and edges. This follows the work of Leighton and Maggs [LeMa], which uses expanders on butterfly networks. Since our fault-tolerant structure is based on layered graphs, the construction and reconfiguration is simpler. Our results apply to a class of graphs with bounded *tree weight* and bounded degree, which includes tree-like structures and one-dimensional dynamic graphs [Or].

2 Definitions and Preliminaries

A VLSI/WSI array architecture is represented by a graph $G = (V, E)$; each node of G is regarded as a processor, and each edge as a connection between two processors. We assume that nodes fail independently, each with (uniform) probability ϵ. As mentioned above, a node in our graph model can represent a PE, a switch, or interprocessor connection.

Real working architectures are considered to be a family of graphs, $\mathcal{G}_a$, called *application graphs*; $G_a^i = (V_a^i, E_a^i)$ denotes the ith application graph of $\mathcal{G}_a$. In this paper the index i will always be the number of nodes in G_a^i. For example, $\mathcal{G}_a$ can be a family of n-node linear arrays. We always assume that each G_a^i is connected. Since we need to add redundant nodes or edges to increase reliability, the embedding structures, $\mathcal{G}_r$, called *redundant graphs*, are also represented as a family of graphs; $G_r^i = (V_r^i, E_r^i)$ denotes the ith redundant graph of $\mathcal{G}_r$.

Given two isomorphic graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, define the isomorphism function μ: $V_1 \rightarrow V_2$ such that $(v_i, v_j) \in E_1$ if and only if $(\mu(v_i), \mu(v_j)) \in E_2$. Let $\mu(V_1)$ be the image of V_1. Given an isomorphism function $\mu : V_1 \rightarrow V_2$, let the mapping set $S(\mu)$ be the set of pairs, $\{(v, \mu(v)) | \, v \in V_1\}$. Thus, $S(\mu) - S(\mu')$ represents the difference between two isomorphism functions μ and μ'.

Given $\mathcal{G}_a$ and $\mathcal{G}_r$, the following function determines which graph in $\mathcal{G}_r$ will be the redundant graph that corresponds to the ith application graph.

Definition 2.1 *An Embedding Strategy for $\mathcal{G}_a$ and $\mathcal{G}_r$ is a function $ES : \mathcal{G}_a \rightarrow \mathcal{G}_r$, i.e., if $ES(G_a^i) = G_r^j$, G_r^j is the redundant graph for G_a^i.*

If $ES(G_a^i) = G_r^j$, and k nodes of G_r^j have failed, the failed nodes and all the edges incident to them will be removed and G_r^j becomes a new subgraph $\hat{G_r^j} = (\hat{V_r^j}, \hat{E_r^j})$. The procedure for finding a new isomorphism function μ_k^i: $V_a^i \rightarrow \hat{V_r^j}$ is called *reconfiguration.*

Definition 2.2 *Given $\mathcal{G}_a$, $\mathcal{G}_r$ and ES, the maximum fault-tolerance of G_a^i, $MFT(G_a^i)$, is the maximum number of nodes that are allowed to fail arbitrarily in $ES(G_a^i)$ such that $ES(G_a^i)$ can still find a subgraph isomorphic to G_a^i. In addition, $FT(G_a^i)$ is given which is some fixed number $\leq MFT(G_a^i)$ for each i.*

Definition 2.3 *Given $\mathcal{G}_a$, $\mathcal{G}_r$, ES and Fault Tolerance $FT(G_a^i) \leq MFT(G_a^i)$ for each i, the quadruple $(\mathcal{G}_a, \mathcal{G}_r, ES, FT)$ is called an Embedding Architecture, EA.*

For simplicity, if the context is clear, we will always assume the ith application graph maps to the ith redundant graph, i.e., $ES(G_a^i) = G_r^i$. Let $\mu_0^i : G_a^i \rightarrow G_r^i$, be the initial isomorphism function for the ith application graph G_a^i.

Definition 2.4 *Given an Embedding Architecture, define the Initial Embedding, IE, to be a set of μ_0^i for all G_a^i in the family.*

Given an *embedding architecture* for a G_a^i, after k nodes have failed, obviously there may be many different isomorphism functions μ_k's. But, the difference between $S(\mu_0^i)$ and $S(\mu_k^i)$ should be as small as possible for the purpose of real-time fault-tolerance.

Suppose that the number of nodes in G_a^i is n. Given EA, IE and that $k \leq FT(G_a^i)$ nodes have failed, let the *cost of reconfiguration* of G_a^i, $\triangle(k,n)$, be the minimum of $|S(\mu_0^i) - S(\mu_k^i)|$ over all the possible isomorphism functions μ_k^i, i.e.,

$$\triangle(k,n) = \min_{\mu_k^i} |S(\mu_0^i) - S(\mu_k^i)|.$$

When there is no μ_k^i, $\triangle(k,n) = \infty$. Under a given EA and IE, let $DR(k,\ n)$, the *Degree of Reconfigurability for G_a^i* , be the maximum of $\triangle(k,n)$ over all possible k failures in G_r^i, $k \leq FT(G_a^i)$; i.e.,

$$DR(k,\ n) = \max_{\substack{\text{failures of k nodes} \\ k \leq FT(G_a^i)}} \triangle(k,n).$$

Definition 2.5 *An Embedding Architecture, EA is finitely reconfigurable, if there exists an Initial Embedding, IE, such that for all the* $G_a^i \in \mathcal{G}_a$, $DR(k,n)$, *can be bounded from above by a function of* k *but not* n.

3 A General Construction for Graphs with Bounded Degree

In this section we restrict attention to *application graphs* with bounded degree, graphs where the node degrees cannot grow indefinitely as the number of nodes increases. This class includes common array architectures, such as pipelines or meshes. The constructed *redundant graphs* will be layer-like. The problem will be how to add redundancy to achieve a fixed level of reliability, which is equivalent to the problem of choosing a good *embedding architecture*. We say a graph G_r can *simultaneously realize* n G_a's if there are n edge- and node-disjoint isomorphic G_a's in G_r. The following is the main result of this section. Actually this theorem can be easily generalized to other classes of graphs by modifying the number of redundant edges allowed.

Theorem 3.1 For every n-node application graph G_a with bounded degree, we can construct an $O(n \log n)$-node redundant graph with degree $O(\log n)$, that can simultaneously realize $O(\log n)$ G_a's with a given fixed level of reliability, while at the same time being finitely reconfigurable. □

To illustrate the construction, we first consider the case when $\mathcal{G}_a$ is a family of linear arrays. A *layered graph* is one where the nodes are partitioned into levels, and edges can only connect nodes in two consecutive levels. The redundant graphs, $\mathcal{G}_r$ are layered graphs, and the N-th redundant graph G_r^N has N levels, each with m nodes (see figure 1). (In this paper N will always be the number of levels, and m the number of nodes in each level). Each level will be considered one stage in the pipeline, and the interconnection between levels in G_r^N is complete. That is, each node in level i is connected to every

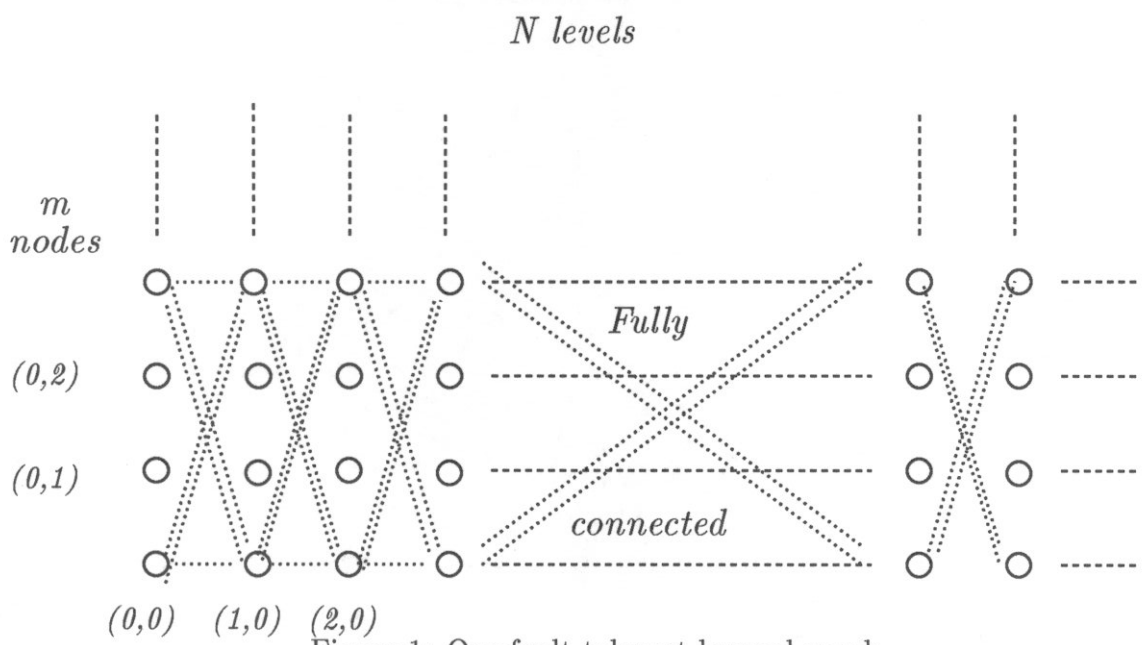

Figure 1: Our fault-tolerant layered graph

node in level $i+1, i = 0 \ldots N-2$. Our initial embedding is to embed the node at stage i of the pipeline G_a^N into an arbitrary node in level i of G_r^N. At any point, after possible failures, the node at stage i of G_a^N will be mapped to one of the m nodes in level i of G_r^N. Finally, we will let the fault tolerance $FT(G_a^N)$ be m.

Having described the embedding architecture, we would like to show that we can maintain any given fixed level of reliability with a value of m that is not too large. We can show:

Lemma 3.2 Let the application graph G_a^N be an N-node linear array, let the redundant graph G_r^N be as shown in figure 1, and let the embedding strategy be as described above. Then we can maintain both finite reconfigurability and any fixed level of reliability if $m = O(\log N)$ nodes in each level of G_r^N.

Proof: Because two consecutive levels are fully connected, only one node needs to be changed after a node fails, $DR(k,\ n) = k$, and the graph is finitely reconfigurable.

Let ε be the probability of failure of one node. We know that the reliability of each stage = Prob [there exists at least one good node] $= 1 - \varepsilon^m$. Therefore, the reliability of

N levels $= (1-\varepsilon^m)^N \geq \exp(-\varepsilon^m Nc)$, where c is a constant. We want this to be bounded from below by the desired reliability β, leading to the condition

$$-\varepsilon^m Nc \geq \log \beta$$

or

$$m \geq \frac{\log N + \log c - \log(-\log \beta)}{-\log \varepsilon}$$

So, $m = O(\log N)$ and the total number of nodes is $O(N \log N)$. □

Actually, we can improve this result by embedding more than one pipeline in G_r^N. In the following lemma, we allow $O(m)$ nodes in G_r^N to be working simultaneously.

Lemma 3.3 The redundant graph G_r^N in figure 1 can maintain finite reconfigurability, any fixed level of reliability, and can simultaneously realize at least $(1-\alpha)m$ pipelines, if $m = O(\log N)$ and $\alpha > \varepsilon$.

Proof: Let the random variable X be the number of faulty nodes in any one level. The reliability of one level = Prob[there are at least $(1-\alpha)m$ good nodes] = Prob[$X < \alpha m$], which is explicitly

$$= \sum_{i=0}^{\alpha m} C_i^m \varepsilon^i (1-\varepsilon)^{m-i}$$

This can be approximated by the integral of the normal distribution [Me]

$$\frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\frac{m(\alpha-\varepsilon)}{\sqrt{m\varepsilon(1-\varepsilon)}}} e^{-t^2/2}\, dt$$

Bounding the tail of the normal distribution from above [Fe], we can write

$$1 - \text{Prob}[X < \alpha m] < \frac{1}{\sqrt{2\pi}} \frac{\sqrt{m\varepsilon(1-\varepsilon)}}{m(\alpha-\varepsilon)} \exp\left(\frac{-1}{2} \frac{(\alpha-\varepsilon)^2}{\varepsilon(1-\varepsilon)}\, m\right)$$

Since $\alpha > \varepsilon$, the upper bound on the right-hand-side is positive. Letting $A = \frac{1}{\sqrt{2\pi}} \frac{\sqrt{\varepsilon(1-\varepsilon)}}{\sqrt{m}(\alpha-\varepsilon)}$ and $B = \frac{(\alpha-\varepsilon)^2}{2\varepsilon(1-\varepsilon)}$, we can write this more compactly as

$$1 - \text{Prob}[X < \alpha m] < \frac{A}{\sqrt{m}} \exp(-Bm)$$

The reliability of N levels is

$$\geq (1 - \frac{A}{\sqrt{m}} \exp(-Bm))^N \geq \exp\left(-\frac{A}{\sqrt{m}} \exp(-Bm)N\right)$$

This can be lower-bounded by β by choosing $m = (\ln A + \ln N - \ln(\ln \frac{1}{\beta}))/B$, which results in $m = O(\log N)$. □

Consider a concrete example, with a desired reliability of β. To satisfy the conditions of the preceding lemma, it turns out to be sufficient to choose the number of nodes in each level to be $m = (\ln N + \ln A - \ln(\ln \frac{1}{\beta}))/B$, where $A = \frac{1}{\sqrt{2\pi}} \frac{\sqrt{\varepsilon(1-\varepsilon)}}{\sqrt{m}(\alpha - \varepsilon)}$ and $B = \frac{(\alpha - \varepsilon)^2}{2\varepsilon(1-\varepsilon)}$. If $\varepsilon = 0.1$, $\alpha = 0.3$, $N = 2^{16}$, and $\beta = 1 - 10^{-8}$, then $m = 131$ is sufficient to satisfy the given reliability criterion with finite reconfigurability. This means we have $0.7 \times 131 > 91$ parallel 2^{16}-node pipelines. If $\varepsilon = 0.1$, $\alpha = 0.5$, $N = 2^{10}$, and $\beta = 1 - 10^{-8}$, then m can be as low as 28, with 14 parallel pipelines.

We can connect many simultaneously realized pipelines together into one long pipeline, in a snake-ike fashion, and if we do that we get the following result.

Theorem 3.4 There exists a two-dimensional layered graph which can maintain an $O(N \log N)$-node linear array with a fixed level of reliability, and local reconfigurability, by using $O(N \log N)$ nodes of degree $O(\log N)$.

Proof: From the previous lemma, we know the graph G_r^N in figure 1 realizes $O(\log N)$ parallel linear arrays with any fixed level reliability. Add two more levels, one before the first level and one after the last, and completely connect these two extra levels with their adjacent levels. This preserves the property of finite reconfigurability. These two levels can be used to stitch together those $O(\log N)$ parallel linear arrays into one linear array of length $O(N \log N)$. □

For example, if $\varepsilon = 0.1$, $\alpha = 0.5$, $N = 2^{10}$, $\beta = 1 - 10^{-8}$, then from lemma 3.3, m can be set to 28. We can therefore construct a linear array of length of 14×2^{10} nodes with the desired reliability and the property of finite reconfigurability. If we include the

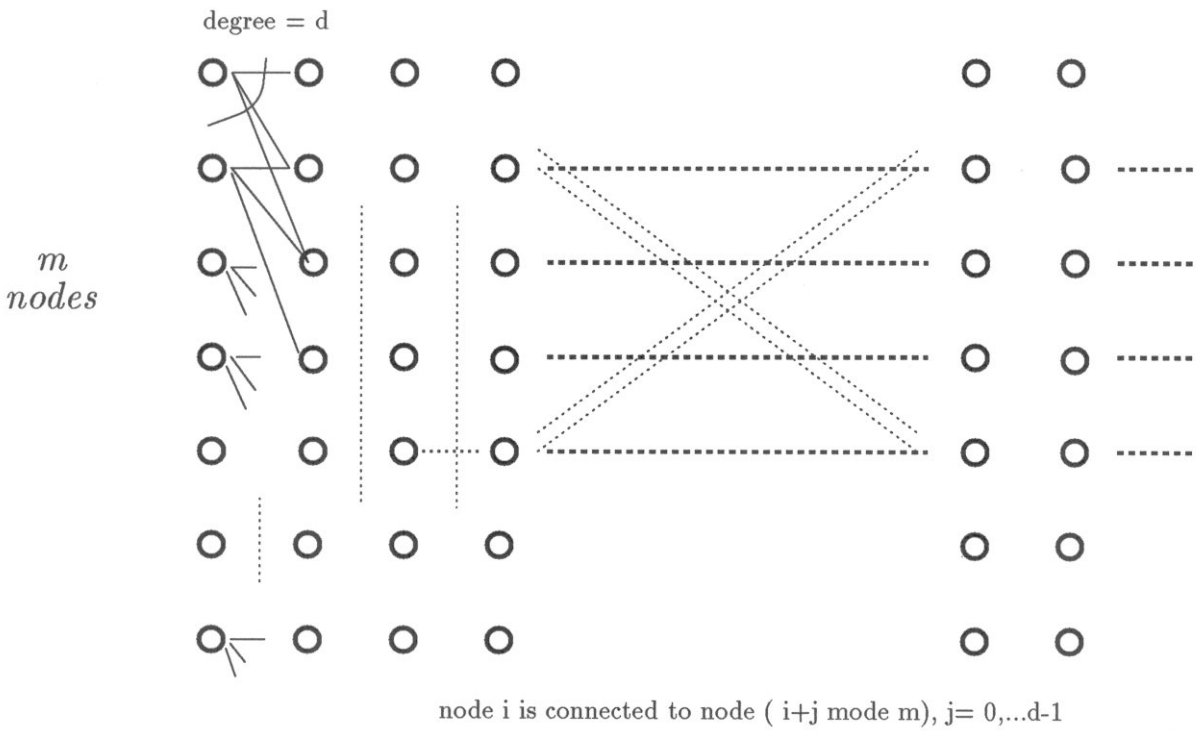

Figure 2: A graph with reduced degree

number of edges in the calculation of cost, the total hardware cost is $O(N \log^2 N)$ for constructing an $O(N \log N)$-node linear array.

Actually, we can easily decrease the degree of each node from m to αm, where $\alpha > \varepsilon$, and since ε is usually small, this gives us a relatively better result, even though αm is still $O(\log N)$. We can do this by constructing a layered graph in which node i is only connected to nodes $(i + j) \, mod \, m$, $j = 0, ..., d - 1$ in the next level, as shown in figure 2. The following lemma shows that there exists a bipartite matching between surviving good nodes in adjacent levels, provided that fewer than d nodes have failed in each level, where d is the degree of each node.

Lemma 3.5 Let A and B be two adjacent levels of the redundant graph G_r^N. If fewer than d nodes have failed in each of A and B, there still exists a matching between those

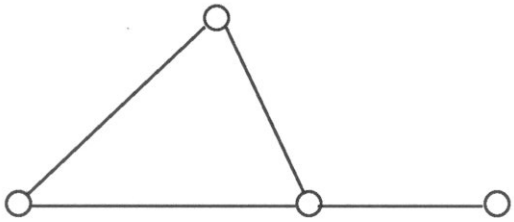

Figure 3: An example of an application graph

two levels with $m - d + 1$ edges.

Proof: Denote by $\Gamma(X)$ the set of nodes in B which are incident to at least one of the nodes in X, $X \subset A$. We consider only the case of matchings with $m - d + 1$ edges (not perfect matchings). We know from [HaVa, LoPl], that $|\Gamma(X)| \geq |X|$ for all subsets $X \subset A$ with $|X| \leq m - d + 1$, if and only if there is a matching with $m - d + 1$ edges. Therefore, if there does not exist a matching of size at least $m - d + 1$, there must be a subset X of good nodes in A, with $|\Gamma(X)| < |X|$. According to the connection pattern, we know that $|\Gamma(X)| \geq |X| + d - 1$ before any nodes have failed. But by hypothesis at most $d - 1$ nodes fail in B, so after failures $|\Gamma(X)| \geq |X|$, which is a contradiction. □

This shows we can use the lemma above and set $d = \alpha m$ to construct a layered graph that maintains a given level of reliability. But now the degree of reconfigurability is not finite, since our pipeline is connected by bipartite matchings between adjacent levels, and we may need to change the entire structure to reconfigure after a failure. We further exploit this idea for maintaining reliable structures using bipartite matchings in the next section.

Lemma 3.3 is generalized to more general structures as follows. Given an N-node graph G_a^N with bounded degree, we construct a new graph G_r^N by substituting a column of $O(\log N)$ nodes for each node of G_a^N and fully connecting this column with the adjacent columns. Figures 3 and 4 show an example. This gives the main theorem at the beginning of this section.

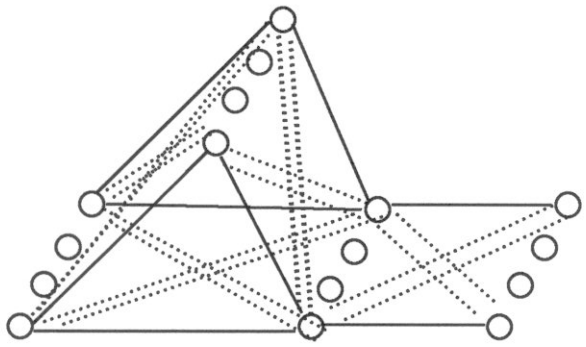

Figure 4: The redundant graph for the previous graph

Proof of Theorem 3.1 : The calculation of the reliability of G_r is exactly the same as in the proof of lemma 3.3. Thus, both finite reconfigurability and any fixed level of reliability can be maintained by choosing columns of size $O(\log N)$. If d is the maximum degree of any node in the original application graph G_a, the degree of each node is then at most $d\times\ O(\log N) = O(\log N)$. □

When the application graph is a k-dimensional dynamic graph [Or], the reliable redundant graph is a $(k+1)$-dimensional graph that realizes $O(logN)$ parallel dynamic graphs, where N is the number of nodes. As in the one-dimensional case (see the proof of theorem 3.4) we can add two k-dimensional hyperplanes at the front and back along any one of the $(k+1)$ dimensions, and use those new levels to connect the $O(logN)$ parallel application graphs together. For example, in the proof of 3.4, the application graph is a linear array, and two columns of nodes are added. In this way, we get the asymptotically minimum number of nodes in the reliable system, but the degree of each node is still not a constant. In the next section, we present optimal redundant graphs for tree-like application graphs which do achieve a constant degree.

4 An Optimal Construction for Tree-like Graphs

In the past few years, the use of expanders has led to new theoretical results on the existence and construction of linear-sized redundant networks for maintaining reliable

linear arrays and trees. Alon and Chung [AlCh] proved that we can maintain reliable n-node linear-array application graphs using fault tolerant graphs with only $O(n)$ edges. Later, Friedman and Pippenger [FrPi] generalized this to a similar result for trees, also using linear-sized fault-tolerant networks. Their results are stronger, since they are not based on a probabilistic model, and allow any portion of nodes to be deleted. But their results do not show how to find an isomorphic application graph in polynomial time, nor how to do reconfiguration after nodes have failed.

Leighton and Leiserson [LeLe] show that under a probabilistic model, with high probability $1 - O(1/n)$, we can connect any constant fraction (less than 1) of the nonfaulty nodes of an $O(n)$-sized two-dimensional mesh to be a linear array. This shows the existence of a small two-dimensional dynamic graph that can maintain reliable linear arrays.

In this section we follow Leighton and Maggs [LeMa] and use expanders to give explicit constructions of fault-tolerant structures for a variety of application graphs, including linear arrays, multiple pipelines, and tree-like structures. These structures are also easily reconfigurable on-line. As before, we first describe the case when $\mathcal{G}_a$ is a family of linear arrays. We need the following fact from probability theory.

Lemma 4.1 The probability that at least $k\varepsilon m$ nodes fail in one level is exponentially small, that is $< \left(\frac{e}{2^k}\right)^{\varepsilon m}$, where k is a constant, and $k > (1/ln2)$.

Proof: Let X be the number of faulty nodes in one level, and $P(Z)$ the probability generating function

$$P(Z) = \sum_{i=0}^{\infty} Prob\,(X = i) Z^i$$

The probability that X exceeds r is

$$\begin{aligned} Prob\,(X \geq r) &= \sum_{i=r}^{\infty} Prob\,(X = i) \\ &\leq \sum_{i=0}^{r-1} Prob\,(X = i)\frac{Z^i}{Z^r} + \sum_{i=r}^{\infty} Prob\,(X = i)\frac{Z^i}{Z^r} \\ &= Z^{-r} P(Z) \end{aligned}$$

where $Z > 1$ is a real number. Since $Prob\,(X = i)$ is the binomial distribution $C_i^m \varepsilon^i (1 - \varepsilon)^{m-i}$, the generating function $P(Z) = (\varepsilon Z + 1 - \varepsilon)^m$. If we set $r = k\varepsilon m$, and $Z = 2$,

$$\begin{aligned} Prob\,(X \geq k\varepsilon m) &\leq 2^{-k\varepsilon m}(1+\varepsilon)^m \\ &\leq 2^{-k\varepsilon m} e^{\varepsilon m} = \left(\frac{e}{2^k}\right)^{\varepsilon m} \end{aligned}$$

Since $k > (1/\ln 2)$, it follows that $\frac{e}{2^k} < 1$. □

We first review some definitions and theorems about expanders [LuPhSa, AlMi, Up, LeMa]. Consider a bipartite graph $G = (A, B, E)$, and let $\Gamma(X)$, $X \subset A$, be the set of nodes in B which are adjacent to X. We say that G has the *expansion property* (α, β, d, m), if $|A| = |B| = m$, the degree of each node in G is d, and for every subset $X \subset A$, $|X| \leq \alpha m$, $|\Gamma(X)|$ is at least $\beta|X|$, where $\alpha\beta \leq 1$, $\beta > 1$. From [Up], we know that there exists an (α, β, d, m) expander such that $\alpha\beta < 1$, and $d < \beta + 1 + \frac{\beta + 1 + \ln \beta}{-\ln \alpha\beta}$. This expander can be found with high probability by randomly generating a regular degree-d bipartite graph. The expansion property can be checked by calculating the second largest eigenvalue of its adjacency matrix [AlMi]. There is also an explicit construction of an expander graph, called the *Ramanujan Graph* [LuPhSa].

Our construction for the N-th redundant graph is similar to the graph in figure 1, except the connection between two adjacent levels is now an expander with expansion property (α, β, d, m), where A corresponds to the previous level, and B to the next level. The matching theorem [HaVa, LoPl] then implies a desirable property: there always exists a matching from every subset $X \subset A$, $|X| = \alpha m$ to the next level B. This gives us αm N-node pipelines. If we can design a reconfiguration method such that with high probability there are still αm N-node pipelines after failures have occurred, this will be a highly reliable structure with a linear number of edges, since the degree of each node is a constant d.

We first describe a key part of our reconfiguration method, called *backward propagation*, which preserves the (α, β, d, m) expansion property after the reconfiguration process.

Backward Propagation: Let a constant $\varepsilon_1 < 1$ be given, and examine each node in each level, starting from level $N-1$, and progressing down to level 0. If a node in level i is adjacent to $\geq \varepsilon_1 d$ nodes in level $i+1$ which are nonfaulty, then the node is declared to be *good*; otherwise, it is declared *faulty*. We call nodes which are declared faulty in this process *propagated faulty nodes*. Thus we have two kinds of faulty nodes — physically faulty nodes and propagated faulty nodes.

Theorem 4.2 After backward propagation, an (α, β, d, m) expander retains the expansion property (α, β', d, m) if $1 < \beta' \leq \beta - (1-\varepsilon_1)d$.

Proof: Consider a subset $X \subset A$ where $|X| \leq \alpha m$ and all nodes in X are good. Let $\Gamma_f(X)$ be the set of faulty neighbors in the succeeding level, and $\Gamma_g(X)$ be the corresponding set of good neighbors. The total number of neighbors is $|\Gamma(X)| = |\Gamma_f(X)| + |\Gamma_g(X)| \geq \beta|X|$. Since each node in X is good, the number of faulty connecting nodes to each must be less than $(1-\varepsilon_1)d$. Therefore

$$|\Gamma_f(X)| \leq (1-\varepsilon_1)d|X|$$

and

$$|\Gamma_g(X)| \geq (\beta - (1-\varepsilon_1)d)|X| = \beta'|X|$$

□

The following lemma shows that not too many nodes at any one level become faulty by propagation.

Lemma 4.3 Let $\varepsilon_1 < \frac{1}{d}(\beta - \frac{\alpha + 3\varepsilon}{\alpha})$, $m = \frac{k}{\varepsilon}\ln N$, and $k > 1$. Then the number of propagated faulty nodes at each level is less than αm with probability at least $1 - \frac{1}{N^{k-1}}$.

Proof: We first prove that if the number of physically faulty nodes is less than $3\varepsilon m$ and the number of propagated nodes is less than αm at level $l+1$, then the number of propagated nodes is less then αm at level l, where $l = N-2, \ldots, 0$. We then prove that with probability $> 1 - \frac{1}{N^{k-1}}$ there are in fact fewer than $3\varepsilon m$ physically faulty nodes

at each level. Thus, with probability $> 1 - \frac{1}{N^{k-1}}$, there are fewer than αm propagated faulty nodes.

The first part is proved by induction. The basis is true trivially because there are no propagated faulty nodes at the last level. Suppose that at level $l+1$, the number of propagated faulty nodes is less than αm, the number of physically faulty nodes is less than $3\varepsilon m$, and at level l, the number of propagated faulty nodes is greater than αm. Choose X to be a set of propagated faulty nodes at level l with $|X| = \alpha m$. Let F be the set of faulty nodes at level $l+1$. From the induction hypothesis, we know that $|F| < \alpha m + 3\varepsilon m$.

Let $\Gamma_f(X)$ be the set of faulty neighbors of X at level $l+1$, and let $\Gamma_g(X)$ be the corresponding set of good neighbors. Since $x \in X$ is a propagated faulty node at level l, the number of connecting good nodes must be less than $\varepsilon_1 d$. Thus, $|\Gamma_g(X)| < \varepsilon_1 d\alpha m$. Since $|\Gamma(X)| \geq \beta\alpha m$, $|\Gamma_f(X)| > \alpha m(\beta - \varepsilon_1 d)$. We know that $\varepsilon_1 < \frac{1}{d}(\beta - \frac{\alpha + 3\varepsilon}{\alpha})$, so $\alpha m(\beta - \varepsilon_1 d) > \alpha m + 3\varepsilon m$. Thus, from $|F| \geq |\Gamma_f(X)|$ and $|\Gamma_f(X)| > \alpha m(\beta - \varepsilon_1 d)$, we know that $|F| > \alpha m + 3\varepsilon m$, which is a contradiction, finishing the induction.

From Lemma 4.1, we know that the probability that the number of physically faulty nodes at any one level exceeds $3\varepsilon m$ is exponentially small, being upper bounded by $< \left(\frac{e}{8}\right)^{\varepsilon m}$. Thus, the probability P that there exists one level with more than $3\varepsilon m$ physically faulty nodes is $< N\left(\frac{e}{8}\right)^{\varepsilon m}$. Since $m = \frac{k}{\varepsilon}\ln N$,

$$\begin{aligned} P &= N\left(\frac{e}{8}\right)^{k\ln N} \\ &= N^{1+k-k\ln 8} \\ &< \frac{1}{N^{k-1}} \end{aligned}$$

Therefore, the probability that there are fewer than $3\varepsilon m$ physicaly faulty nodes at every level is greater than $1 - \frac{1}{N^{k-1}}$. Thus, the conclusion of the induction part is true with probability greater than $1 - \frac{1}{N^{k-1}}$. □

Appropriate choice of α, β, d and ϵ_1 in the two lemmas above yields the following

theorem.

Theorem 4.4 An n-node linear array can be maintained at any fixed level of reliability using a simple layered structure with $O(n)$ edges, provided $\varepsilon < \frac{1-2\alpha}{3}$.

Proof: From lemma 4.1, we know that with probability greater than or equal to $1 - \left(\frac{e}{8}\right)^{\varepsilon m}$, there are fewer than $3\varepsilon m$ physically faulty nodes at any one level. Thus, as in the proof of lemma 4.3, with probability greater than $1 - \frac{1}{N^{k-1}}$, there are fewer than $3\varepsilon m$ physically faulty nodes at every level. After backward propagation, by lemma 4.3, at most αm nodes are declared to be faulty at each level with probability greater than $1 - \frac{1}{N^{k-1}}$, where m can be set to $\frac{k}{\varepsilon} \ln N$. Therefore, with probability greater than $1 - \frac{2}{N^{k-1}}$, there are fewer than $\alpha m + 3\varepsilon m$ total faulty nodes. We can always make k large enough to satisfy the desired reliability.

Also, if we choose $\beta' = (\beta - (1 - \varepsilon_1)d) > 1$, the structure will still have the expansion property after finishing the backward propagation. Since $\varepsilon < \frac{1-2\alpha}{3}$, we know $\alpha m < m - (\alpha + 3\varepsilon)m$. We can find αm pipelines in the whole system, and can add two special levels, as in the proof of theorem 3.4, one before the first level and one after the last, to stitch the pipelines together. This will make an $O(N \log N)$-node linear array, with total number of edges $2m^2 + mN = O(N \log N)$, since $m = O(\log N)$. □

Note that even when the given error probability ϵ is more than $(1 - 2\alpha)/3$, we can reduce the error probability to satisfy the upper bound constraint of theorem 4.4 by using multiple modular redundancy and a majority voter for each node. From [Pi], we know that a k-argument majority function can be computed by a noisy network of size $O(k^7)$. Since $(1 - 2\alpha)/3$ is a constant, the number of multiple modules and the size of the majority voter for each node are also constants. Thus, the upper bound constraint of theorem 4.4 can be satisfied with only a constant factor more hardware.

By using the same techniques, we can get similar results for trees with bounded degree δ. The construction replaces each node of the tree with a column of m nodes. The

connection between two adjacent columns is also an expander with expansion property (α, β, d, m). To establish the result we generalize backward propagation as follows: Apply the old backward propagation method for each path from the root to a leaf. A node is *good* if it is declared good in every path it is in; otherwise, it is *faulty.*

The expansion property will be guaranteed after this generalized reconfiguration, provided we choose ε_1, d and β to make the new $\beta' > 1$, as we did in theorem 4.2. The number of propagated faulty nodes is also upper-bounded by $(\delta - 1)\alpha m$ with high probability, by an analysis similar to that establishing lemma 4.3. We therefore get the following corollary.

Corollary 4.5 For every N-node tree T with bounded degree, we can construct an $O(N \log N)$-node layer-like structure with bounded degree (and hence $O(N \log N)$ total hardware), which can simultaneously realize $O(\log N)$ T's with any given fixed reliability. □

Using similar techniques, we can generalize our result to the class of graphs with bounded tree weight, defined as follows:

Definition 4.1 Tree Weight: *Given a graph G, consider all ways in which its nodes can be contracted to components to yield a tree. A contraction whose largest component is smallest possible will be called a minimax contraction. The number of nodes in the largest component of a minimax contraction is defined to be the tree weight of G.*

We can get the following theorem for graphs which have bounded tree-weight and bounded degree. This class includes tree-like structures and one-dimensional dynamic graphs.

Theorem 4.6 For every N-node graph G with bounded degree and bounded tree weight, we can construct an $O(N \log N)$-node layer-like structure with bounded degree (and hence $O(N \log N)$ total hardware), which can simultaneously realize $O(\log N)$ G's with any given fixed reliability.

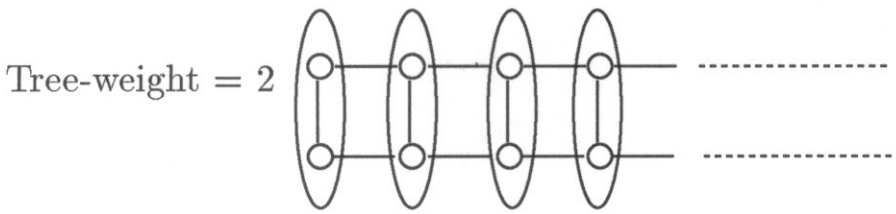

Figure 5: Example of application graph 1

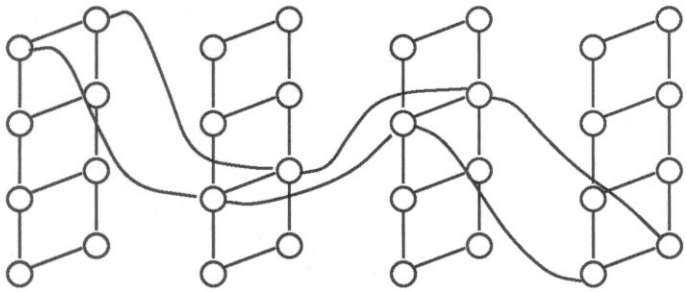

Figure 6: The reliable redundant version of graph 1

Proof: Let the *tree-weight* of G be k and let the corresponding tree be T. We can substitute a column of m components for each component of T (at most k nodes in G). We also connect two adjacent columns together to satisfy the expansion property (α, β, d, m). The error probability ε' of one component is at most $1-(1-\varepsilon)^k$, which is still a constant. From Corollary 4.5, we know that we maintain any fixed level of reliability by choosing parameters appropriately and setting m to $O(\log N)$. □

Simple algorithms can be used to implement both the backward propagation method, and reconfiguration of the bipartite matchings between adjacent columns.

Figures 5 – 8 show the constructions for two examples.

5 Conclusions

We have presented two explicit ways to construct reliable and easily reconfigurable structures for wide classes of application graphs. The reconfigurable structures are particularly

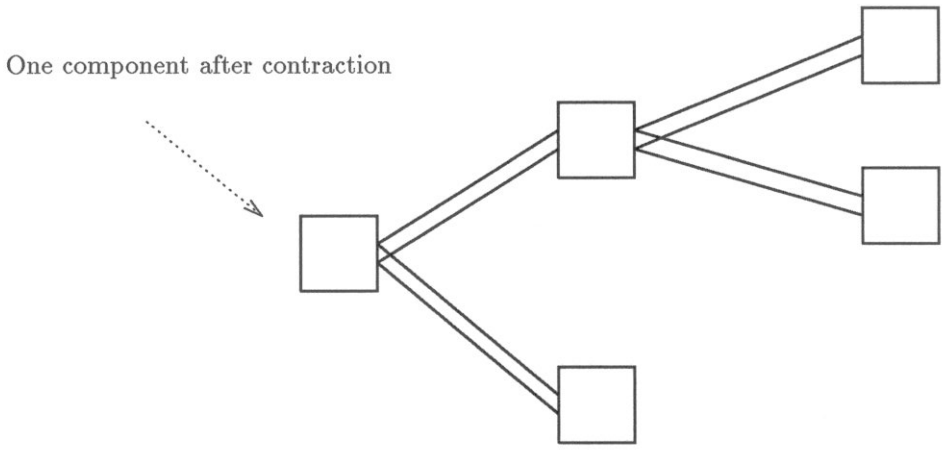

Figure 7: Example of application graph 2

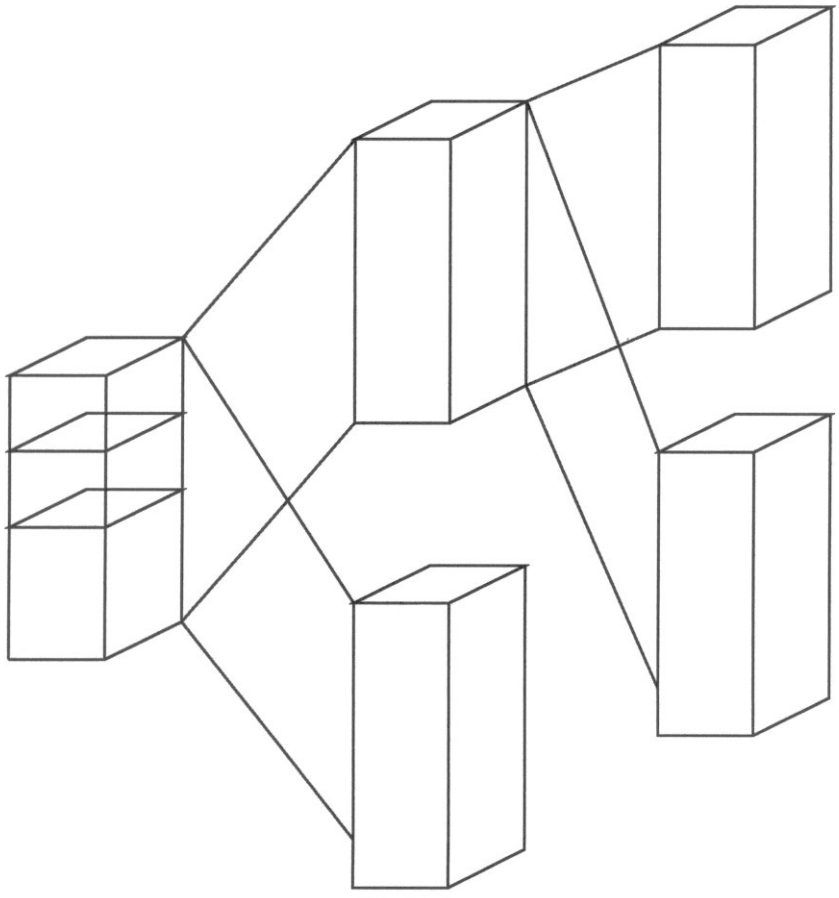

Figure 8: The reliable redundant version of graph 2

amenable to on-line reconfiguration. The first construction, described in section 3, applies to any application graph with bounded degree, and results in finitely reconfigurable graphs with the asymptotically minimum number of extra nodes. However, the asymptotic number of extra edges is not linear in the size of the original application graph.

The second construction, described in section 4, applies to tree-like application graphs, yields an explicit reconfiguration procedure, and results in redundant graphs with a linear (optimal) number of nodes and edges. References [AlCh, FrPi] also give linear-sized reliable structures for linear arrays and trees, but do not give reconfiguration procedures. Although the redundant graphs that result from the second construction are not finitely reconfigurable, the reconfiguration procedure — backward propagation and bipartite matching — can be implemented easily.

We mention here some problems that deserve further study. The construction in section 3 produces a redundant graph with $O(N \log^2 N)$ edges for an application graph which is an $(N \log N)$-node mesh. To the authors' knowledge it is unknown how to construct a linear-sized redundant structure that maintains a fixed level of reliability for even a two-dimensional mesh.

The redundant graphs resulting from our constructions are not dynamic graphs. It would be interesting to consider the construction of redundant graphs that are restricted to be dynamic graphs, which are more easily implemented than other, less regular graphs. We do know from [ShSt] that if we wish to use a dynamic graph as a redundant structure to maintain both finite reconfigurability, and a fixed level of reliability, for an application graph that is an array, the dynamic graph must be of dimension at least one greater than that of the array. Nontrivial lower and upper bounds on the size of reliable and finitely reconfigurable redundant structures that are restricted to be dynamic graphs are not known.

References

[AlCh] N. Alon and Fan R.K. Chung, "Explicit construction of linear sized tolerant networks," *Discrete Math.*, 72, 1988, pp. 15-19.

[AlMi] N. Alon and V.D. Milman, "λ_1, isoperimetric inequalities for graphs and superconcentrators," *J. Comb. Theory*, B 38, 1985, pp. 73-88.

[ASAP] *Proc. 1990 International Conf. on Application Specific Array Processors,* S.Y.Kung, Earl Swartzlander, Jose Fortes, and Wojtek Przytula eds., Princeton, NJ.

[ChLeRo] Fan R.K. Chung, F.T. Leighton, and A.L. Rosenberg, "Diogenes: A methodology for designing fault-tolerant VLSI processing arrays," *Proc. IEEE FTCS,* Milano, 1983, pp. 26-32.

[DoCr1] R.L. Dobrushin and S.I. Crtyukov, "Lower bound for the redundancy of self-correcting arrangement of unreliable functional elements," *Prob. of Info. Transm.*, vol. 13, 1977, pp. 59-65.

[DoCr2] R.L. Dobrushin and S.I. Crtyukov, "Upper bound for the redundancy of self-correcting arrangement of unreliable functional elements," *Prob. of Info. Transm.*, vol. 13 1977, pp. 203-218.

[Fe] W. Feller, *An Introduction to Probability Theory and its Applications,* John Wiley & Sons, Inc, NY, 1968.

[FrPi] J. Friedman and N. Pippenger, "Expanding graphs contain all small trees," *Combinatorica*, vol. 7, 1987, pp. 71-76.

[GrGa] J.W.Greene and A.E. Gamal, "Configuration of VLSI arrays in the presence of defects," *J. Asso. Comp. Mach.,* vol. 31, Oct. 1984, pp. 694-717.

[HaVa] P.R. Halmos and H.E. Vaughan, "The marriage problem," *Amer. J. Math.*, vol. 72, 1950, pp214-215.

[HKu] H.T. Kung, " Why systolic architectures?" *Computer Magizine*, vol. 15 , no. 1, January 1982, pp. 37-46.

[KuLa] H.T. Kung and M.S.Lam, "Fault tolerant VLSI systolic arrays and two-level pipelines," *J. Parall. and Distr. Proc.,* vol. 8, 1984, pp. 32-63.

[KuJeCh] S.Y. Kung, S.N. Jean and C.W. Chang, "Fault-tolerant array processors using single track switches," *IEEE Transactions on Computers,* vol. C-38, no. 4, April 1989, pp. 501-514.

[LeLe] T. Leighton and C. E. Leiserson, "Wafer-scale integration of systolic arrays," *IEEE Transactions on Computers,* vol. C-34, no. 5, 1989, pp. 448-461.

[LeMa] T. Leighton and B. Maggs, "Expanders might be practical: fast algorithms for routing around faults on multibutterflies," *Proc. FOCS*, 1989 pp. 384-389.

[LoPl] L. Lovasz and M.D. Plummer, *Matching theory*, North-Holland, New York, 1986.

[LuPhSa] A. Lubotzky, R. Phillips, and P. Sarnak, "Ramanujan graphs," *Combinatorica*, vol. 8, 1988, pp. 261-277.

[Me] P. Meyer, *Introductory probability and statistical applications,* Addison-Wesley, 1970.

[Or] Orlin, J., "Some problems on dynamic/periodic graphs," *Progress in Combinatorial Optimization*, W. R. Pulleyblank (ed.), Academic Press, Orlando, Florida, 1984, pp. 273-293.

[Pi] N. Pippenger, "On Networks of Noisy Gates," *Proc. FOCS*, 1985, pp. 30-38.

[PiStTs] N. Pippenger, G. Stamoulis, and J. Tsitsiklis "On a lower bound for the redundancy of reliable networks with noisy gates," Tech. Report, LIDS-P-1942, MIT, 1990.

[SaSt] M. Sami and R. Stefenelli, "Reconfiguration architecture for VLSI processing arrays," *Proc. IEEE FTCS*, 1986, pp. 712-722.

[ShSt] E. H.-M. Sha and K. Steiglitz, "Reconfigurability and reliability of systolic/wavefront arrays," to appear in *Proc. 1991 IEEE Int. Conf. on Acoustic, Speech, and Signal Processing*, Canada. (A Full version is available as Tech. Report CS-TR-280-90, Dept. of Computer Science, Princeton University, Aug., 1990.)

[SKu] S.Y. Kung, *VLSI Array Processors,* Prentice Hall, Englewood Cliffs, NJ, 1988.

[Up] Eli Upfal, "An O(log N) deterministic packet routing scheme," *Proc. STOC*, 1989, pp. 241-250.