Wafer-Scale Integration of Systolic Arrays

TOM LEIGHTON, MEMBER, IEEE, AND CHARLES E. LEISERSON, MEMBER, IEEE

Abstract — **VLSI technologists are fast developing *wafer-scale integration*. Rather than partitioning a silicon wafer into chips as is usually done, the idea behind wafer-scale integration is to assemble an entire system (or network of chips) on a single wafer, thus avoiding the costs and performance loss associated with individual packaging of chips. A major problem with assembling a large system of microprocessors on a single wafer, however, is that some of the processors, or *cells*, on the wafer are likely to be defective. In the paper, we describe practical procedures for integrating "around" such faults. The procedures are designed to minimize the length of the longest wire in the system, thus minimizing the communication time between cells. Although the underlying network problems are NP-complete, we prove that the procedures are reliable by assuming a probabilistic model of cell failure. We also discuss applications of the work to problems in VLSI layout theory, graph theory, fault-tolerant systems, planar geometry, and the probabilistic analysis of algorithms.**

Index Terms — **Channel width, fault-tolerant systems, matching, probabilistic analysis, spanning tree, systolic arrays, traveling salesman problem, tree of meshes, VLSI, wafer-scale integration, wire length.**

I. Introduction

VLSI technologists are fast developing *wafer-scale integration* [32]. Rather than partitioning a silicon wafer into chips as is usually done, the idea behind wafer-scale integration is to assemble an entire system (or network of chips) on a single wafer, thus avoiding the costs and performance loss associated with individual packaging of chips. A major problem with assembling a large system of microprocessors on a single wafer, however, is that some of the processors, or *cells*, on the wafer are likely to be defective. Thus, a practical procedure for integrating wafer-scale systems must have the ability to configure networks "around" such faults.

This paper considers a variety of problems involving the construction of systolic arrays [18]. Systolic arrays are a desirable architecture for VLSI because all communication is between nearest neighbors. In a wafer-scale system, however, all the nearest neighbors of a processor may be *dead*, and thus the prime advantage of adopting a systolic array architecture may be lost if a long wire connects electrically adjacent processors. In general, the longest interconnection between processors will be a communication bottleneck in the system. Of the many possible ways in which the *live* cells on a wafer can be connected to form a systolic array, therefore, the one that minimizes the length of the longest wire is most desirable.

To illustrate the subtleties inherent in configuring systolic arrays, consider the problem of constructing a linear (i.e., one-dimensional) array using all of the live cells in an N-cell wafer. Unfortunately, if we wish to minimize the length of the longest wire, the problem is NP-complete [13]. Even more discouraging is that there are some arrangements of live and dead cells for which even the optimal linear array has unacceptably long wires. Thus optimal solutions — even if they could be found quickly — are not always practical.

By assuming a probabilistic model of cell failure, however, many positive results can be proved. For example, Fig. 1 illustrates a possible solution to the problem of connecting the live cells of a wafer into a linear systolic array. The live cells, which are denoted by small squares, are connected together, one after another, in a snake-like pattern. Dead cells, denoted by ×'s, are skipped over. With probability $1 - O(1/N)$, the length of the longest wire is $O(\lg N)$, where N is the number of cells in the wafer and where each cell independently has a 50 percent chance of failure.[1]

This bound comes from the observation that the length of the longest wire that connects two cells in the array is just the length of the longest sequence of dead cells in the snake-like string. For a given set of k cells, the probability that all are dead is $1/2^k$, and thus the probability that any set of $2 \lg N$ cells are dead is $1/N^2$. Since there are less than N sets of $2 \lg N$ consecutive cells, the chances are thus less than one in N of having to skip more than $2 \lg N$ cells in the entire snake-like path of length N. Hence, the maximum wire length is $O(\lg N)$ with probability $1 - O(1/N)$.

To say that "with probability $1 - O(1/N)$ the maximum wire length is $O(\lg N)$" is a substantially stronger statement than saying that the expected maximum wire length is $O(\lg N)$. This is because no wire can ever have length greater than $O(\sqrt{N})$, even in the worst case. Hence, the expected maximum wire length is at most

$$(1 - O(1/N)) \times O(\lg N) + O(1/N) \times O(\sqrt{N}) = O(\lg N).$$

Manuscript received May 16, 1983; revised February 4, 1984. This work was supported in part by the Defense Advanced Research Projects Agency under Contract N00014-80-C-0622, in part by the U.S. Air Force under Contract OSR-82-0326, and in part by a Bantrell Fellowship. A preliminary version of this paper was presented at the 1982 IEEE Foundations of Computer Science Conference.

T. Leighton is with the Department of Mathematics and the Laboratory for Computer Science, Massachusetts Institute of Technology, Cambridge, MA 02139.

C. E. Leiserson is with the Laboratory for Computer Science, Massachusetts Institute of Technology, Cambridge, MA 02139.

[1] Here and throughout the paper, we use $O(f(N))$ to denote a function that is bounded above by $cf(N)$ for a fixed constant c and all sufficiently large N. We also use $\Omega(f(N))$ to denote a function that is bounded below by $cf(N)$, and $\Theta(f(N))$ to denote a function that is bounded above by $c_1 f(N)$ and below by $c_2 f(N)$ for some fixed constants c, c_1, and c_2, and all sufficiently large N. We also use $\lg N$ to denote $\log_2 N$, $\lg^2 N$ to denote $(\lg N)^2$, and $\lg \lg^2 N$ to denote $(\lg \lg N)^2$. Lastly, $\lfloor x \rfloor$ denotes the largest integer less than or equal to x, and $\lceil x \rceil$ denotes the smallest integer greater than or equal to x.

0018-9340/85/0500-0448$01.00

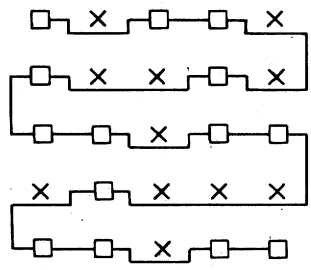


Fig. 1. A simple means of constructing a linear systolic array from the live cells on a wafer.

Moreover, the chances that the maximum wire length is much greater than $O(\lg N)$ are minuscule. In particular, the probability of having to skip more than $k \lg N$ dead cells at a fixed point in the snake-like path is less than one in N^k. Hence, every wire has length at most $k \lg N$ with probability $1 - 1/N^{k-1}$.

Not surprisingly, there are algorithms which, under similar assumptions of cell failure, produce far better results than the algorithm illustrated in Fig. 1. For example, we will describe in Section III another simple procedure which, with high probability, constructs a linear array using wires of length $O(\sqrt{\lg N})$. We will also show that, up to the leading constant, the algorithm is the best possible of its kind. By relaxing the constraint that *all* live cells be connected into the linear array, however, we can do much better. In fact, we will also show in Section III that with high probability a linear array containing any constant fraction (less than one) of the live cells on an N-cell wafer can be constructed using wires of constant length.

Although there are numerous uses for linear systolic arrays [24], two-dimensional systolic arrays are also important. Not only can the two-dimensional array be used as a powerful communications structure for parallel computation [18], but it can also serve as an all-purpose structure in which arbitrary networks can be embedded [3], [23], [45], [47]. As one might expect, the problem of constructing a two-dimensional array from the live cells of a wafer is more difficult than the corresponding problem for linear arrays. Specifically, Section IV contains a proof that with high probability a two-dimensional array that uses any constant fraction of the live cells must have wires of length $\Omega(\sqrt{\lg N})$.

Although we do not know how to construct two-dimensional arrays from most of the live cells using wires of length $O(\sqrt{\lg N})$ or channels of constant width, we can come close. We show in Section VI that with high probability, a two-dimensional array can be constructed on an N-cell wafer using:

1) all the live cells with wires of length $O(\lg N \lg \lg N)$ and channels of width $O(\lg \lg N)$,

2) any constant fraction less than one of the live cells with wires of length $O(\sqrt{\lg N} \lg \lg N)$ and channels of width $O(\lg \lg N)$, and

3) at least $\Omega(1/\lg \lg^2 N)$ of the live cells with wires of length $O(\sqrt{\lg N})$ and channels of width 1.

The remainder of the paper is divided into seven sections. Section II more formally describes our model for wafer-scale integration and discusses the practicality of the modeling assumptions. The algorithms for constructing linearly connected systolic arrays are presented in Section III. Section IV contains the lower bound result for wire length in two-dimensional systolic arrays. In Section V we present a worst case (nonprobabilistic) upper bound on the channel width necessary to configure a two-dimensional array. This result also has application to the fault-tolerant encoding of two-dimensional arrays in complete binary trees [34]. Section VI gives algorithms for constructing two-dimensional arrays in the probabilistic model. In Section VII, we mention some related problems in geometric complexity, graph theory, and the probabilistic analysis of algorithms. The related problems are nice theoretically in that some of them have tight upper and lower bounds. They also suggest a wealth of interesting questions concerning the design of fault-tolerant systems. We conclude the paper with some additional remarks in Section VIII.

II. The Wafer-Scale Model

Laser programming the interconnect of a wafer is a promising means of achieving wafer-scale integration. This technology was pioneered at IBM [26] and pursued in the direction of wafer-scale integration at M.I.T. Lincoln Laboratory [32], [33]. Fig. 2 shows a scanning electron microscope photograph of a portion of a wafer with programmable interconnect. Laser welds can be made between two layers of metal, and by using the beam at somewhat higher power, wires can be cut. Defective components can thus be avoided by programming connections among only the good components.

Fig. 3 shows a typical organization of a wafer-scale system with programmable interconnections. The components are organized as a matrix of cells, and between the cells are channels through which the interconnect runs. Fig. 4 is a close-up of the channel structure. At the intersection of a horizontal and vertical channel, laser-programmable connections can make a horizontal and a vertical wire electrically equivalent. Between two cells, connections can be made from the wires in the channel to the inputs and outputs of the two cells. Given that the interconnect is programmable, we shall adopt a usage of the term "wire" to mean an electrically equivalent portion of the programmable interconnect.

The preassignment of wire segments to layers such that wires in one layer run horizontally and in the other run vertically is commonly referred to as *Manhattan* wiring [19]. This wiring model has been studied extensively, but in this paper the details of the wiring are not the central issue. It will be sufficient to understand one fact about Manhattan wiring: the width of a channel need only be a constant factor times the maximum number of wires that occupy any portion of the channel.

A natural question to ask about the use of programmable interconnections to avoid defective cells is, *"If cells are unreliable, why might not the interconnect fail also?"* The answer is that, indeed, interconnect does fail. But the reliability of the interconnect is much higher than the reliability of the cells. The interconnect in the M.I.T. Lincoln Laboratories project [33], for example, takes three masking steps to fabricate, but manufacturing the active devices requires well over a dozen steps. This project targets yields of 30 to 50 percent for cells and over 95 percent for wires. And even

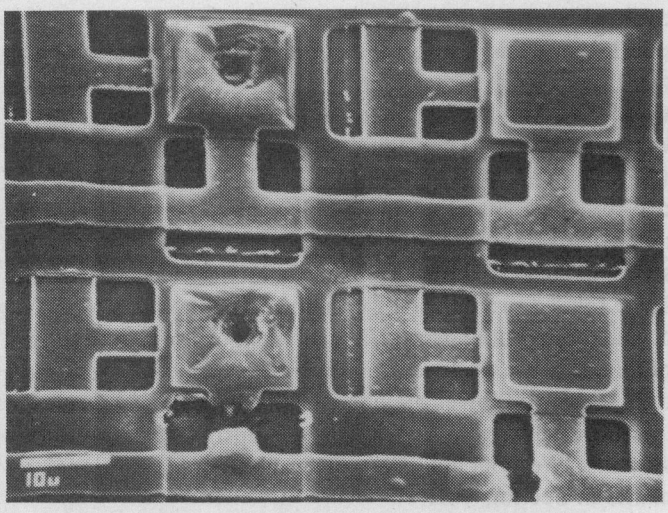


Fig. 2. A close-up of laser-programmable interconnect.

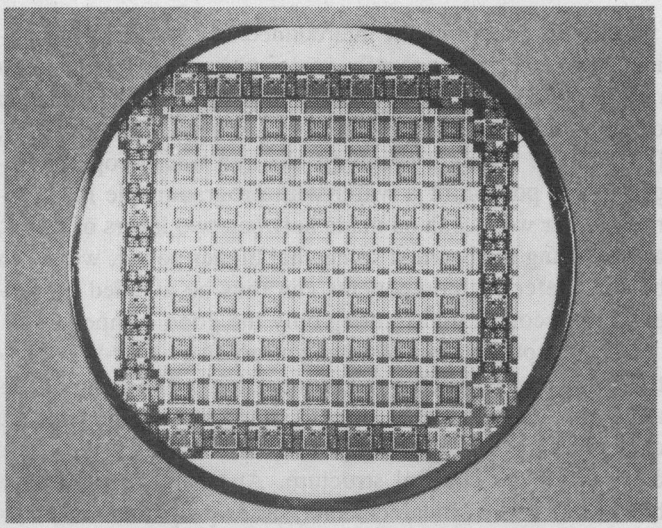


Fig. 3. A wafer-scale system of cells and programmable interconnect.

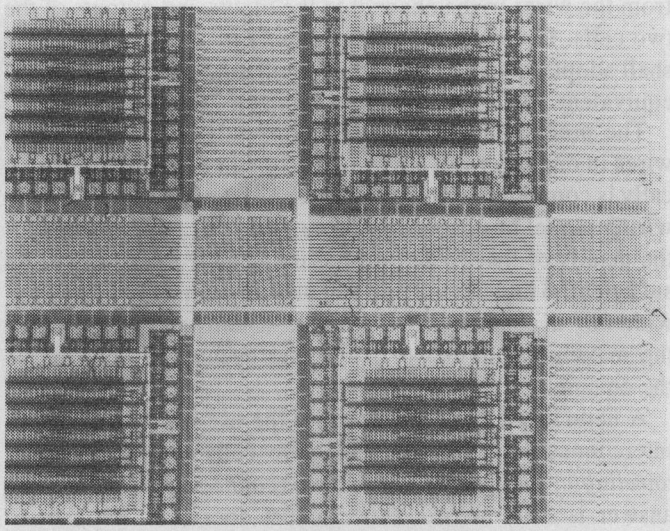


Fig. 4. The channel structure of a wafer-scale system.

if a wire fails at one point, it is often possible to break it into two useable pieces.

In this paper we shall assume that the interconnect has sufficient redundancy so that the inability to interconnect cells arbitrarily is a rare phenomenon. In this sense, we are making the same assumption that is used to substantiate redundancy in any fault-tolerant system. The idea is not that the system will be completely reliable, but that its failure will depend on the failure of the most reliable component instead of the least reliable component.

Another assumption that must be examined more closely is that the probability of cell failure is independent and the same for all cells. Failures can be attributed to one of several causes. Here we consider two material defects during manufacturing and mask misalignment. Material defects are spread uniformly, but the size of the region affected by a defect is a separate random variable. This means that if one point on the wafer is flawed, neighboring points are also likely to be flawed. Nevertheless, independence of cell failures is quite a reasonable assumption because the area of a cell is substantially larger than the expected area of a defect.

Mask misalignment is a somewhat more serious problem with respect to our modeling assumptions. The reason is that misalignment is a global failure mode. Misalignment due to translation of the axes of one mask relative to the others poses no real problem in terms of the modeling assumptions, however, because the effect is the same for all cells. The real problem is misalignment due to angular rotation of one mask with respect to the others. Those cells near the center of rotation are much more likely to be good than those far from the center. Experimental evidence indicates, however, that the effects from angular rotation that cannot be accounted for by our model are minimal.

The two cost functions we shall examine in this paper are *channel width* and *maximum wire length*. Minimizing channel width is important because the available wafer area is essentially fixed. If the channel width is large, the size of the system, and hence its functionality, is reduced. In addition, larger channel widths require longer wires to cross them, and minimizing the length of the longest wire is our other cost criterion.

Minimizing the length of the longest wire in a wafer-scale system is important because communication delays can be the limiting factor of the performance of the system. Since both resistance and capacitance increase with the length of wire, the time required to drive a wire might grow as fast as the square of the length of the wire [29]. (See [5] for a discussion of propagation delays through wires.) In particular, a designer who chooses a two-dimensional systolic array architecture is counting on low overhead for communication, and will not want communication down a long wire to degrade the performance of the system. Moreover, for reasons of electrical correctness, cells must be designed with signal buffers capable of driving the maximum length wire. Assuming the speed of a circuit is to be maximized, the size of a buffer must vary with the size of the load being driven. Thus, a substantial amount of area in a cell can be saved if the maximum length wire is known to be short. As was argued previously, this saving in area translates to larger systems with greater functionality.

Throughout the paper, we will consider cells that occupy an $s \times s$ square region on the wafer and that have (independently) a probability p of failure. Unless specifically stated to the contrary, we will assume for simplicity that $s = 1$ and

$p = 1/2$. As we will later observe, these restrictions have little bearing on the analysis. In addition, we will use the term *"high probability"* to mean *"with probability at least* $1 - O(1/N)$" where N is the number of cells on the wafer.

We conclude this section with a simple result that places the rest of this paper in a proper context. Given a circuit of a given area, composed of active components and wires, it is possible to construct a wafer of not much more area (asymptotically) which is fault tolerant. If there are N active components, expand the layout of the circuit in each dimension by $c\sqrt{\lg N}$ where c is a constant chosen large enough that $2 \lg N$ copies of a given active component fit in the space designated to that component in the original circuit. The probability that every one of the $2 \lg N$ copies is bad is $1/N^2$, and thus with high probability, one of the copies of every component is good. It only remains to hook them up in the space left for wires.

This scheme works even if components are different, but is not very practical since only one in every $2 \lg N$ cells is used. For typical values of $N \geq 100$, this is grossly inefficient. The results in this paper are much better because we can utilize substantially more of the live cells at less cost; some of our algorithms use all of the live cells, and others use a considerable proportion.

III. Wafer-Scale Integration of Linearly Connected Systolic Arrays

With high probability, the snake-like scheme described in the introduction connects all the live cells on an N-cell wafer into a linear array with wires of length at most $O(\lg N)$. This section substantially improves and generalizes this result. We commence by showing that this bound can be improved to $O(\sqrt{\lg N})$, which is optimal to within a constant factor.

Theorem 1: With probability $1 - O(1/N)$*, the live cells on an N-cell wafer can be connected in a linear array using wires of length* $O(\sqrt{\lg N})$*. Up to the leading constant, this bound is the best possible.*

Proof: We first show how to construct a linear array using wires of length $O(\sqrt{\lg N})$. Partition the wafer into square regions containing $2 \lg N$ cells each as is shown by the dashed lines in Fig. 5. The probability that each of the $2 \lg N$ cells is dead in one or more of the squares is at most

$$\frac{N}{2 \lg N} 2^{-2 \lg N} = \frac{1}{2N \lg N},$$

which is less than $1/N$. Thus, with probability $1 - O(1/N)$, each of the squares contains at least one live cell.

Construct a linear array out of the live cells in each square using the "transpose" of the algorithm from Section I, except that when an empty column is encountered, the column is skipped. In Fig. 5, these connections are shown with solid lines. Since any pair of cells in the same square can be linked with a wire of length at most $2\sqrt{2 \lg N}$, the wires in each array have length $O(\sqrt{\lg N})$. Next, add wires, shown by dotted lines in the figure, which connect the small arrays into one large array. Because each region contains at least one live cell, these connections can be made with wires of length at most $3\sqrt{2 \lg N}$. Thus, every wire in the completed linear array has length $O(\sqrt{\lg N})$ with high probability.

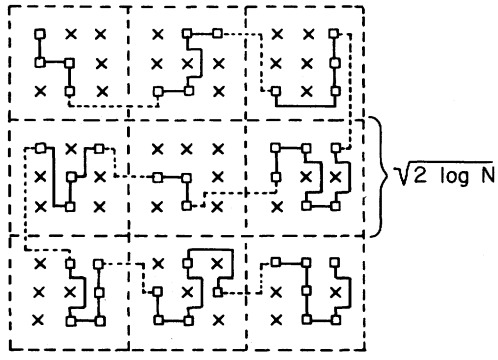


Fig. 5. A scheme for constructing linear arrays from all live cells on a wafer with wires of length $O(\sqrt{\lg N})$ and constant channel widths.

That the bound cannot be improved by more than a constant factor is due to the observation that with high probability some live cell will be at the center of a region of $\Omega(\lg N)$ dead cells. Thus, a wire of length $\Omega(\sqrt{\lg N})$ will be required to link the isolated live cell to any other live cell. To demonstrate this bound more formally, we again partition the wafer into square regions, but this time the squares are rotated by 45° in the plane to form diamond-shaped regions containing $\lg N - 2 \lg \lg N$ cells each, as is shown in Fig. 6.

Suppose a linear array can be constructed using wires of length at most $\sqrt{1/2 \lg N - \lg \lg N}$. Then in any given diamond, the center cell is not the only live cell in the diamond. The probability that every diamond avoids this condition is at most

$$\begin{aligned}(1 - 2^{-\lg N + 2 \lg \lg N})^{N/(\lg N - 2 \lg \lg N)} &= \left(1 - \frac{\lg^2 N}{N}\right)^{N/(\lg N - 2 \lg \lg N)} \\ &\leq e^{(-\lg^2 N)/N \times N/(\lg N - 2 \lg \lg N)} \\ &= e^{-(\lg^2 N)/(\lg N - 2 \lg \lg N)} \\ &\leq e^{-\lg N} \\ &\leq \frac{1}{N}.\end{aligned}$$

Thus, the probability that the optimal linear array has a wire of length $\Omega(\sqrt{\lg N})$ is at least $1 - O(1/N)$. ∎

If all the cells are incorporated in a linear array, then the maximum wire length is $\Theta(\sqrt{\lg N})$ with high probability. But the proof of the lower bound suggests that isolated cells induce the long wires. Instead of insisting that *all* live cells be incorporated in the linear array, suppose we only require that *most* of the live cells be included. A linear array that incorporates most of the live cells can be constructed with constant-length wires. The proof is indirect, and depends on the following lemma. (The lemma is essentially equivalent to the result of Sekanina [39], which states that the cube of a nontrivial connected graph always has a Hamiltonian circuit. This result was later reproved by Karaganis [14] and Rosenberg and Snyder [38].)

Lemma 2: A spanning tree T with maximum wire length L can be transformed into a linear array with maximum wire length 6L.

Proof: We show that, without regard for wire widths, the linear array can be constructed using wires of length $3L$

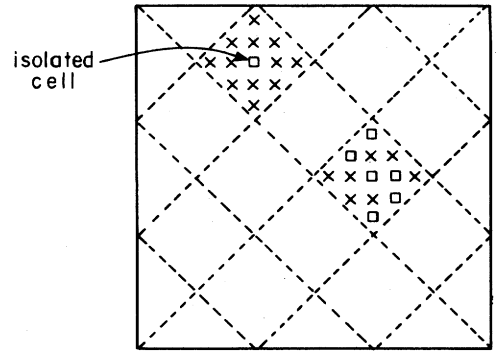


Fig. 6. An example of an isolated cell.

by tracing over wires in T no more than twice each. Since every wire is traced over at most twice, the channel widths could (at worst) double in the resulting wiring, thus increasing the maximum wire length from $3L$ to $6L$ when wire widths are accounted for.

Choose a node v to be the root of T, and let $T_1, T_2, \cdots, T_m$ be the subtrees of v as is shown in Fig. 7. (Degenerate cases not like Fig. 7 are easily handled, but we do not include the details here.)

Assume as an inductive hypothesis that we have constructed linear arrays on the nodes of $T_1, T_2, \cdots, T_m$ such that no wire has length greater than $3L$, and so that the endpoints of the array in T_i are v_i and u_{i1} for $1 \leq i \leq m$. Join the arrays in the subtrees by adding the following wires: (v, u_{11}), $(v_1, u_{21}), (v_2, u_{31}), \cdots, (v_{m-1}, u_{m1})$. (These wires are shown as dashed lines in Fig. 7.) Each of these wires has length at most $3L$, and the resulting network is a linear array on the nodes of T with endpoints v and v_m, which completes the induction. For completeness, we remark that the basis of the induction is easily verified. ∎

The problem of constructing a linear array with constant maximum wire length that contains most of the live cells has now been reduced to the problem of constructing a spanning tree with constant maximum wire length that contains most of the live cells. The next lemma shows that such a spanning tree can be formed with high probability.

Lemma 3: There exists a positive constant c such that for any d (which might be a function of N), with probability $1 - O(1/N)$, at least $1 - O(2^{-cd^2})$ of the live cells on an N-cell wafer can be connected in a spanning tree using wires of length at most d. Up to the constant factors, this is the best possible bound.

Proof: We first show that up to constant factors, the bound is the best that one could hope for. In fact, we show something stronger—that for any constant $\hat{c} > 2$ with probability $1 - O(1/N)$, no more than $\lceil N - O(2^{-\hat{c}d^2}N)\rceil$ of the live cells on an N-cell wafer can be connected in *any* network using wires of length at most d. The proof is based on showing that with high probability, there are $\Omega(N/d^2 2^{2d^2})$ live cells, each of which is located at the center of a region of dead cells whose radius is at least d for any d such that $d^2 2^{2d^2} \leq N/(16 \lg N)$. For larger d and sufficiently large N, $\lceil N - O(2^{-\hat{c}d^2}N)\rceil = N$, and the claim is trivially true.

Partition the wafer into diamond-shaped regions, as was done in Fig. 6 to prove the lower bound of Theorem 1, except make the size of each region be $2d^2$ cells. The probability that

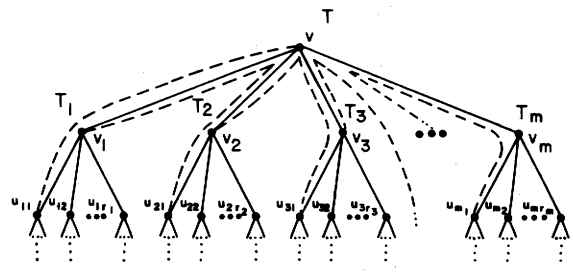


Fig. 7. Constructing a linear array from a spanning tree.

any particular region consists of an isolated live cell at the center of $2d^2 - 1$ dead cells is 2^{-2d^2}. The probability that T or fewer of the $N/2d^2$ regions are like this is thus

$$\sum_{x=0}^{T} \binom{N/2d^2}{x} (2^{-2d^2})^x (1 - 2^{-2d^2})^{(N/2d^2)-x}$$

$$\leq (1 - 2^{-2d^2})^{N/2d^2} \sum_{x=0}^{T} \frac{\left(\frac{N}{2d^2}\right)^x 2^{-2d^2 x}}{(1 - 2^{-2d^2})^x x!}$$

$$\leq e^{-N/2d^2 2^{2d^2}} \sum_{x=0}^{T} \frac{N^x}{[2d^2(2^{2d^2} - 1)]^x x!}.$$

These calculations are made using the standard inequalities $\binom{n}{r} \leq n^r/r!$ and $1 - \epsilon \leq e^{-\epsilon}$. When T assumes the value $N/8d^2 2^{2d^2}$, the largest term in the series occurs for $x = T$, and thus the preceding expression can be bounded above by

$$\frac{(T+1)}{T!} e^{-N/2d^2 2^{2d^2}} \left(\frac{N}{2d^2(2^{2d^2} - 1)}\right)^T = O(1/N)$$

whenever $N/(8d^2 2^{2d^2}) \geq 2 \lg N$ by a simple application of Stirling's formula for $T!$.

In order to prove the upper bound, consider the graph whose vertices are live cells on the wafer and whose edges connect cells which are within distance d of each other on the wafer. In what follows, we will show that there is one *main connected component* in this graph, and that the total size of all other *isolated components* is a small fraction of N. More specifically, we will show that there exist constants c and c' such that the probability that more than $c' 2^{-cd^2} N$ live cells are isolated is $O(1/N)$.

This means that, without regard for wire widths, $1 - O(2^{-cd^2})$ of the live cells can be connected in a spanning tree with wires of length d or less. As in the proof of Lemma 2, extending the result to wires of unit width only changes the overall wire lengths by a constant factor, which can be accounted for by choosing c appropriately.

First, partition the wafer into $d/8 \times d/8$ blocks of cells. (By adjusting the value of c, we can always assume that d is a multiple of 8.) Call a block *live* if it contains at least one live cell and *dead* otherwise. By definition, every isolated region of live cells must be surrounded by a contiguous rectilinear path of dead blocks. Define the *outer boundary* of an isolated region to be the rectilinear path of dead blocks that encompasses the fewest blocks overall. Then it is easily checked that the outer boundaries for distinct isolated regions do not overlap one another. For example, see Fig. 8.

Our approach will be to find a crude upper bound on the number of rectilinear paths of blocks that can define the outer

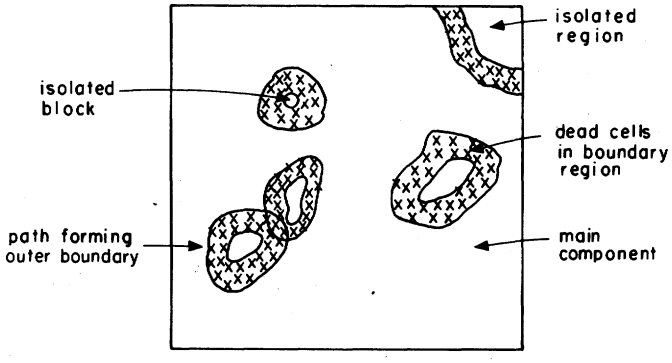


Fig. 8. Examples of isolated regions.

boundary of an isolated region. For any given path which defines the outer boundary of a potentially isolated region, the probability is very small that all the blocks are actually dead in the path. In particular, the longer the path that defines a potentially isolated region, the smaller the probability that the region is actually isolated. Hence, we will be able to conclude that very few (and only small) regions are isolated overall.

Because there are $64N/d^2 \leq N$ blocks at which a path can start and at most four ways it can continue at each step, there are at most $N4^r$ paths consisting of r consecutive blocks. Thus, there are at most

$$\binom{N4^r}{k} \leq \frac{N^k 4^{rk}}{k!}$$

sets of k different paths of length r.

The number of paths of length r is quite a formidable number, and at first glance it seems unlikely that our approach will work. The probability is quite small, however, that each of k given paths actually defines a region which is both isolated and contains at least one live block. For a region to be isolated, its outer boundary must consist of at least $rd^2/64$ dead cells where $r \geq 3$. The probability that all $krd^2/64$ cells are dead in the outer boundaries of k potentially isolated regions with a boundary of length r is $2^{-krd^2/64}$. Thus, the probability that there are actually k isolated regions, each containing one or more live blocks, with outer boundaries of length r, is at most

$$\frac{N^k 2^{2rk - krd^2/64}}{k!},$$

which for $k \geq eN/2^{rd^2/512}$ and $d \geq 32$ is less than $1/N^2$.

Observe that a region with an outer boundary of length r contains $O(d^2r^2)$ live cells. Thus, for $d \geq 32$, with probability $1 - O(1/N)$, at most

$$\sum_{r=3}^{N} \frac{eN}{2^{rd^2/512}} O(d^2r^2) = O\left(\frac{d^2N}{2^{d^2/512}}\right)$$

live cells are isolated from the largest component on the wafer, which implies that for a sufficiently small constant $c > 0$, at most $O(2^{-cd^2}N)$ live cells are isolated. For $d < 32$ the same result holds by simply adjusting the constant factors. ∎

By choosing d to be a sufficiently large constant, Lemma 3 ensures that with high probability a constant fraction of the live cells on the wafer can be connected into a spanning tree with constant wire length. Because we know all wires will be constant length, Prim's minimum spanning tree algorithm [30] can be modified to construct a spanning tree in linear time instead of the normal $O(N^2)$.

Theorem 4: With probability $1 - O(1/N)$, any constant fraction (less than 1) of the live cells on an N-cell wafer can be connected in a linear array with constant-length wires.

Proof: Straightforward from Lemmas 2 and 3. ∎

To conclude this section, we provide a theorem which states our results on constructing linear arrays in their fullest generality. The proof is similar to that of Lemma 3, and is not included here.

Theorem 5: With probability $1 - O(1/N)$, at least $1 - \epsilon$ of the live cells on an N-cell wafer can be connected in a linear array using wires of length $O(s\sqrt{\log_p \epsilon})$ and channels of width 2, where p is the probability of a particular cell dying, s is the side length of each cell, and $1/N \leq \epsilon \leq p < 1$. This bound cannot be improved by more than a constant factor for any p, ϵ, or s.

IV. A Lower Bound for Wafer-Scale Integration of Two-Dimensional Systolic Arrays

The problem of linking the live cells on a wafer to form a square two-dimensional array is substantially more difficult than the corresponding problem for linear arrays. The main difficulty with constructing two-dimensional arrays is that constant length wires no longer suffice when we throw away some of the live cells. In this section we provide a lower bound on the length of the longest wire required by a two-dimensional array. This bound was first discovered by Greene and Gamal [10]. Our proof (which is similar to but more general than that in [10]) was obtained independently from an idea due to Spencer [44].

Theorem 6: With probability $1 - O(1/N)$ every realization of any m-cell two-dimensional array on an N-cell wafer has a wire of length $\Omega(\sqrt{\lg m})$, for all $m = \Omega(\lg^2 N)$.

Proof: The proof consists of two parts. In the first, we show that with high probability, the wafer contains a large number of regularly spaced square regions of $1/4 \lg m$ cells, each of which is dead. In the second part of the proof, we show that any realization of an m-cell two-dimensional array must contain a cycle of four cells that surrounds the center of one of these dead regions. Thus, one of the wires in the four-cycle will have length at least $1/2\sqrt{\lg m}$.

First, partition the N-cell wafer into square regions with $m/32$ cells each, and then partition each of these regions into square subregions with $1/4 \lg m$ cells each. We claim that with high probability, every $m/32$-cell region contains a $1/4 \lg m$-cell subregion in which every cell is dead, as is illustrated in Fig. 9.

The probability that any particular $1/4 \lg m$-cell subregion contains at least one live cell is $1 - m^{-1/4}$. Thus, the probability that each of the $1/4 \lg m$-cell subregions in a particular $m/32$-cell region contains at least one live cell is

$$(1 - m^{-1/4})^{m/8 \lg m} \leq e^{-m^{3/4}/8 \lg m}$$

since $1 + x \leq e^x$ for all x. The probability that one or more of the $32N/m$ $m/32$-cell regions fail to contain a totally dead

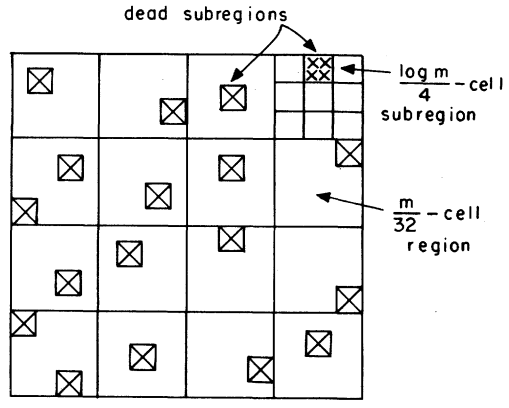


Fig. 9. The distribution of dead 1/4 lg m-cell subregions.

1/4 lg m-cell subregion is at most

$$\frac{32N}{m} e^{-m^{3/4}/8 \lg m} = O(1/N)$$

for $m = \Omega(\lg^2 N)$, which completes the first half of the proof.

If we can show that a four-cycle of the two-dimensional array encloses the center of one of the 1/4 lg m-cell dead regions, the proof will be complete because one of the wires of the four-cycle will have length at least $1/2\sqrt{\lg m}$. More generally, however, if *any* cycle in the array surrounds the center of a dead subregion, then *some* wire in the array must have length $1/2\sqrt{\lg m}$. This observation follows because

1) every directed cycle in a two-dimensional array can be decomposed into the sum of directed four-cycles, and

2) the number of times a cycle "wraps" around a point in the plane is equal to the sum of the number of wraps for each four-cycle in its decomposition.

Thus, a two-dimensional array with a cycle that encloses the center of a dead region must also contain a four-cycle that surrounds the center of the dead region.

We must now show that with high probability, every realization of an m-cell two-dimensional array contains a cycle that encloses the center of a square region of 1/4 lg m dead cells. We already know that with high probability a wafer contains a dead subregion of this size in every square region of $m/32$ cells. Assume for the purposes of contradiction that an m-cell two-dimensional array can be realized on such a wafer so that no cycle of the array surrounds the center of one of the dead regions. Consider a line drawn between the centers of two dead regions. If any wires cross this line, their removal will disconnect the two-dimensional array into two or more components, as is shown in Fig. 10.

Among all pairs of neighboring dead regions (i.e., pairs contained in $m/32$-cell regions that share an edge or corner), there is at least one pair for which removal of the wires passing between them disconnects the array into two pieces, each with at least $m/3$ cells. Since at most $4\sqrt{m/32} = \sqrt{m/2}$ wires can cross the line between the centers of two neighboring dead regions, by removing only $\sqrt{m/2}$ wires, we can disconnect an m-cell two-dimensional array into two pieces, each with at least $m/3$ cells. But it is well known that any such disconnection requires $\sqrt{m}$ wires to be removed, and we have obtained the contradiction that completes the proof. ∎

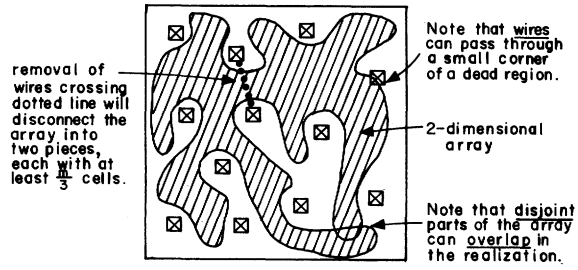


Fig. 10. Disconnecting a two-dimensional array.

The most interesting case of Theorem 6 is when the two-dimensional array to be constructed has $m = \Theta(N)$ cells.

Corollary 7: With probability $1 - O(1/N)$ every realization of any two-dimensional array that utilizes any constant fraction of the live cells on an N-cell wafer has a wire of length $\Omega(\sqrt{\lg N})$.

It is worth noting that Theorem 6 and Corollary 7 are true even if wires have no width. To prove this claim, we break the wafer into square regions with $m/\lg^3 m$ cells each and then into subregions with 1/4 lg m cells each. Otherwise, the proof is as before (there will still be a totally dead (1/4 lg m)-cell subregion in every $(m/\lg^3 m)$-cell region with high probability, for example) except that we must reargue that fewer than $\sqrt{m}$ wires can cross the line between any pair of neighboring dead subregions. This is not difficult since only $O(d^2\sqrt{m}/\lg^{3/2} m)$ length d wires can cross such a line. This is because crossing wires must link two cells (which do have width) that are located within distance d of the line. There are at most $O(d^2\sqrt{m}/\lg^{3/2} m)$ such cells, each with degree at most four. Since $d = O(\sqrt{\lg m})$, the number of wires crossing the line is $O(\sqrt{m/\lg m})$, which is much less than $\sqrt{m}$ for all sufficiently large m.

V. A Divide-and-Conquer Method for Constructing Two-Dimensional Systolic Arrays

The principal focus of this paper is the construction of systolic arrays on wafers such that the maximum wire length is minimized. In this section we ignore maximum wire length as a cost measure and look at the problem of constructing systolic arrays when only channel width is at issue. In doing so, we shall extend the general VLSI layout results of [3] and [23] to the wafer-scale situation where some of the cells may be faulty. Furthermore, the analysis of this section is *worst case* and not probabilistic, and thus all possible configurations of live and dead cells, however unlikely, can be handled. We have included this material in the paper because we use it heavily in Section VI when we need to bound the worst case behavior of our algorithm on very small portions of the wafer.

The basic result of this section is that a two-dimensional array can always be constructed from all the live cells of an N-cell wafer if the channels have width $\Omega(\lg N)$. The proof technique is based on the divide-and-conquer paradigm and is a special case of Theorem 15 in [3].

The proof consists of two parts. First, we show how to encode (à la Rosenberg [34]) any two-dimensional array (square or otherwise) into a complete binary tree where some of the leaves may be dead. We then show how this encoding

can be used to obtain the desired embedding of the array on a wafer.

An *encoding* of a graph $G = (V, E)$ in a tree T is a one-to-one mapping f from the vertices V to the leaves T. In our case, f must map V to live leaves of T. Such a mapping can be extended naturally to map E to the paths of T, where f maps (u, v) to the unique simple path connecting $f(u)$ to $f(v)$.

Lemma 8: Let T be a complete binary tree with each of its N leaves labeled as either "live" or "dead," and let M be the number of live leaves. Then for any M-element two-dimensional array G, there exists an encoding f of G in T such that only $O(\sqrt{k})$ edges of E are mapped by f to an edge of T that has k descendant leaves.

Proof: We consider first the special case when all the leaves of T are live and G is a square array. The encoding is constructed by a simple induction on the number of cells in an array of the form shown in Fig. 11 for which the length is at most twice the width.

Assume for the inductive hypothesis that such an array with $N/2$ cells can be encoded in an $N/2$-leaf binary tree so that only $O(\sqrt{k})$ edges of the array are mapped to an edge of the tree with k descendant leaves, and so that the "cut-wires" incident to the border cells of the array are routed to the root of the tree. Then consider an N-cell array of the same form. By cutting at most $O(\sqrt{N})$ wires, we can partition the N-cell array along its shorter dimension into two $(N/2)$-cell arrays of the desired form. Using the induction hypothesis to encode the subarrays and cut-wires in the left and right subtrees of T, we can complete the encoding by connecting the $O(\sqrt{N})$ cut-wires through the root of T. The inductive hypothesis is thus established.

The previous result is easily extended to nonsquare arrays by partitioning subarrays along their shorter dimension during each inductive step. Extension to the case when $N > M$ is a bit more difficult and is explained in the following.

For general values of N and M, assign labels $1, 2, \cdots, M$ to the live cells of T in order from left to right. Similarly label the cells of G so that were $N = M$, the ith cell of G would be mapped to the ith cell of T by the encoding algorithm for the special case when $N = M$. The encoding of G into T for general N and M is then given by simply mapping the ith cell of G into the ith live cell of T.

We now verify that at most $O(\sqrt{k})$ edges of G traverse through any edge of T with k descendant leaves. Let e be an edge of T with k descendant leaves, $k' \leq k$ of which are live. Consider the encoding of G in an all-live M-leaf tree T', and let S be the interval of leaves in T' corresponding to the live leaves under e in T. By removing at most two edges from each of the bottom lg k' levels of T', S can be disconnected from the rest of T'. (For example, see Fig. 12.) Hence, at most

$$O(\sqrt{k'} + \sqrt{k'/2} + \sqrt{k'/4} + \cdots + 1) = O(\sqrt{k'})$$

edges link cells below e in T to cells in the rest of T. Thus, at most $O(\sqrt{k})$ paths cross e in T, and the proof is complete. ∎

The encoding of a two-dimensional array in an N-leaf complete binary tree corresponds naturally to an embedding of the array in an $O(N)$-leaf *tree of meshes* [3], [20], [21]. The tree of meshes is constructed from a complete binary tree by replacing nodes with meshes and single edges with groups of

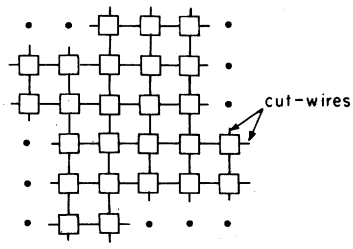


Fig. 11. A 6 × 6 array that is missing some border cells.

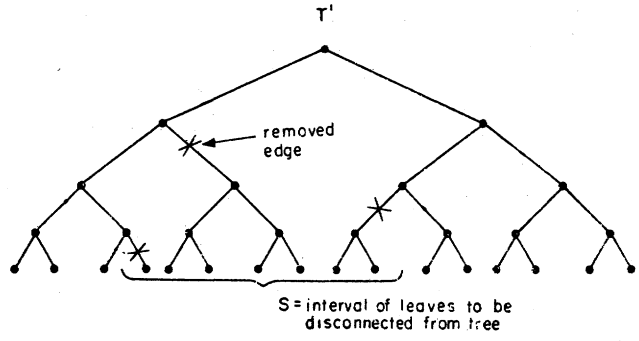


Fig. 12. Removing an interval of leaves from a tree in the proof of Lemma 8.

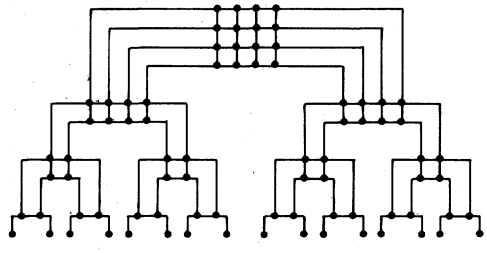


Fig. 13. The 16-leaf tree of meshes.

edges linking the meshes. For example, Fig. 13 shows a 16-leaf tree of meshes.

The root of the complete binary tree in the encoding of Lemma 8 has $O(\sqrt{N})$ connections passing through it from one side to the other. In the corresponding tree of meshes, the switching of these connections is accomplished by a $\Theta(\sqrt{N}) \times \Theta(\sqrt{N})$ mesh at the root. The two subtrees of the root of the complete binary tree correspond recursively to the two subtrees of the root of the tree of meshes. The leaves of the complete binary tree correspond to small meshes at a fixed constant distance from the leaves of the tree of meshes.

The complete binary tree and the tree of meshes differ in depth by an additive constant so that the tree of meshes can be made large enough to accommodate the wiring of the array. In particular, the upper level meshes of the tree of meshes contain only wires, the bottom level meshes are empty, and the small meshes near the bottom contain the cells of the two-dimensional array. If we chop off the unused lower level meshes, we obtain a *truncated tree of meshes* whose leaves correspond to the cells of the two-dimensional array. The next lemma shows that a truncated tree of meshes can be embedded on a wafer with channels of width $O(\lg N)$.

Lemma 9: An N-leaf truncated tree of meshes can be constructed on an N-cell wafer that has a uniform channel width of $\Theta(\lg N)$ so that the leaves of the truncated tree of meshes correspond in a one-to-one manner with the cells of the wafer.

Proof: The first step is to construct a $\Theta(\lg N)$-layer three-dimensional layout [22], [35] of the truncated tree of meshes. Fold the connections between the root of the truncated tree of meshes and each of its two sons so that the sons fit naturally on a second layer over the root. Fold the connections to each of the grandsons so that they fit naturally over the sons on a third layer, and so forth. This generates a $\Theta(\lg N)$-layer three-dimensional layout where each layer has linear area. By projecting the three-dimensional layout onto a single layer in the manner of [46, pp. 36–38], channels with a uniform width of $\Theta(\lg N)$ are obtained. ■

The next theorem is the major result of this section.

Theorem 10: Any M-cell two-dimensional array can be constructed from any M-subset of the live cells on an N-cell wafer using wires of length $O(\sqrt{N} \lg N)$ and channels of width $O(\lg N)$.

Proof: Immediate from Lemmas 8 and 9. ■

VI. Upper Bounds for Wafer-Scale Integration of Two-Dimensional Systolic Arrays

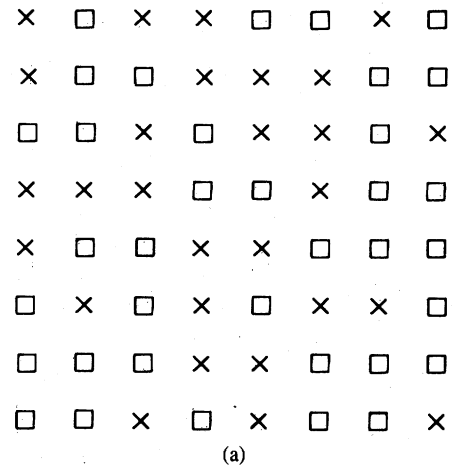
Theorem 6 from Section IV gives a lower bound of $\Omega(\sqrt{\lg N})$ on the length of a wire in any realization of a two-dimensional systolic array that utilizes all or most of the live cells of an N-cell wafer. We do not know how to achieve this lower bound, but we can come close. This section gives three nontrivial upper bounds for wire length and channel width. Of the three methods, however, only the algorithm in the proof of Theorem 13 achieves the lower bound of Theorem 6. Unfortunately, this algorithm utilizes only $m = \Theta(N/\lg \lg^2 N)$ of the live cells.

We first present a divide-and-conquer algorithm that constructs a square two-dimensional array using all the live cells on a wafer. In the first stage, the wafer is recursively bisected, and the number of live cells in each half is counted. Based on the count of live cells in each half of the wafer, the algorithm computes the dimensions of the two subarrays that must be constructed, and then recursively constructs the subarrays. The two subarrays are then linked together to form the complete array.

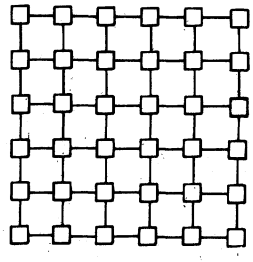
The algorithm remains in the first stage until subproblems with $\Theta(\lg N)$ cells are encountered. At this point, the techniques used in Theorem 10 are used to complete the wiring of a $\Theta(\lg N)$-cell subarray. The exact crossover point between the first and second stages can be set at subproblems of size $c \lg N$, where c is any constant sufficiently large to ensure that with high probability every $c \lg N$-cell region contains $\Omega(\lg N)$ live cells. (For example, a choice of $c = 2$ will suffice.)

Figs. 14 through 17 illustrate the divide-and-conquer procedure. Fig. 14(a) shows a 64-cell wafer which contains 36 live cells. In what follows, we step through the algorithm as it constructs a 6×6 array, which is identified as the "overall target" in Fig. 14(b).

The first step is to bisect the wafer vertically, which gives 19 live cells in the left half and 17 in the right. Therefore, we wish to construct a 19-cell subarray in the left half of the wafer and a 17-cell subarray in the right half of the wafer.



(a)



(b)

Fig. 14. (a) A 64-cell wafer that contains 36 live cells. (b) The target: a 6×6 systolic array.

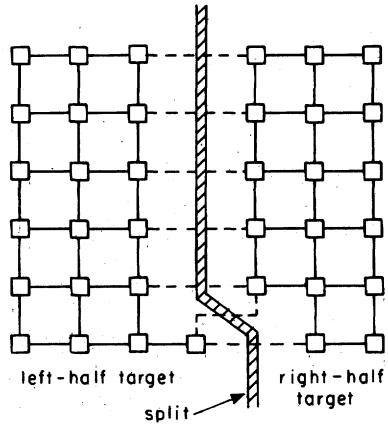


Fig. 15. Partitioning the target.

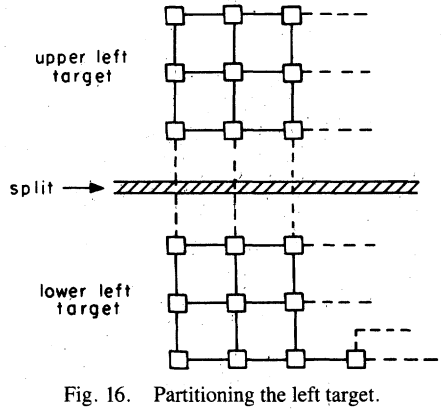


Fig. 16. Partitioning the left target.

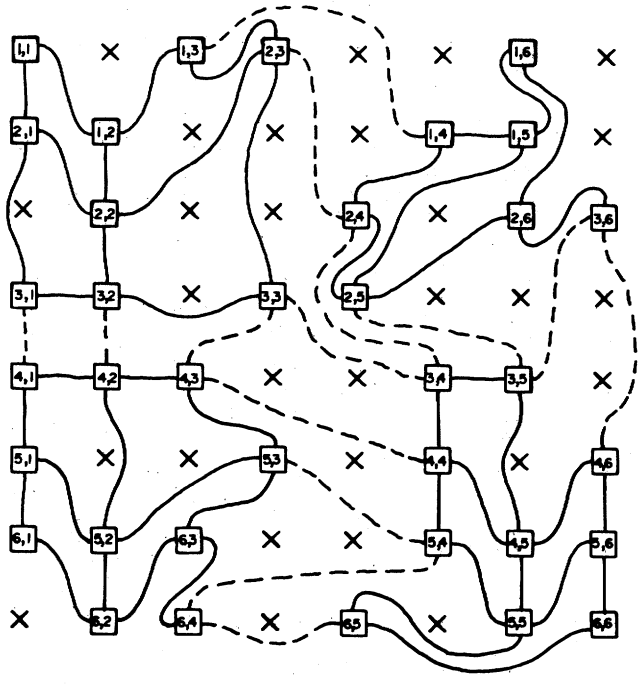


Fig. 17. Completed cell assignment and wiring of the 6 × 6 array.

Since we want the two subarrays to fit together nicely after they have been constructed, we choose the shapes of the two subarrays that are determined by the partition of the 6 × 6 array shown in Fig. 15.

We now invoke the procedure recursively on the two subarrays, but this time we bisect each of the halves horizontally. For example, when the left half of the wafer is bisected, the 19 live cells are divided into 9 cells above and 10 cells below, as displayed in Fig. 16. The algorithm continues in this fashion, alternating between horizontal and vertical divisions, until the wafer and the target have been partitioned into $\Theta(\lg N)$-cell regions, at which point the algorithm proceeds to the second stage, and the technique of Theorem 10 is applied.

In this example the number of cells is small enough that the second stage construction can be performed by inspection. The inspection strategy can also be used effectively in practice. Since the second stage operates on regions of size $\Theta(\lg N)$, the routings of this size can be precomputed. The second stage then consists of a single table lookup. At worst, this strategy costs polynomial time and space.

Fig. 17 shows the final solution to the problem in Fig. 14. For clarity the wires have not been routed within the channels of the wafer. Notice that each quadrant contains the specified targets for the second level of recursion. The dashed lines represent wires that connect cells in different quadrants of the wafer.

The next theorem describes how well the divide-and-conquer algorithm performs with respect to wire length and channel width.

Theorem 11: With probability $1 - O(1/N)$ a two-dimensional array can be constructed from all the live cells on an N-cell wafer using wires of length $O(\lg N \lg \lg N)$ and channels of width $O(\lg \lg N)$.

Proof: The divide-and-conquer algorithm just described provides the bounds in the theorem. The analysis is divided into two parts corresponding to the two stages of the algorithm.

We begin at the first level of recursion. Consider the wires that link a cell in the left subarray to a cell in the right subarray, as is illustrated by the two examples in Fig. 18. For the most part, the connecting wires can be routed in the channel that separates the left and right halves of the wafer. The length of the longest wire in the channel, as well as the width of the channel itself, is proportional to the longest vertical distance that a single wire must traverse.

The length of the longest wire in the center channel depends on the distribution of cells in each quadrant. For example, if we are extremely lucky and the live cells are regularly spaced, the longest wire may have constant length, as in Fig. 18(a). But if we are very unlucky, half the live cells might occur in the upper right quadrant and the other half in the lower left quadrant [Fig. 18(b)]. To connect the two halves, some wire will have length $\Omega(\sqrt{N})$.

The length of the longest wire in the center channel can also be influenced by the distribution of cells within a quadrant. For example, if the upper left quadrant contains $\sqrt{N/8}$ live cells (about the right number), but they are distributed as in Fig. 19, then the center channel still contains a wire of length $\Omega(\sqrt{N})$.

Most often, we are not so unlucky that a wire in the center channel has length $\Omega(\sqrt{N})$, but neither are we lucky enough that all wires have constant length. We now show that with high probability, we are more lucky than unlucky because the length of the longest wire in the center is $O(\lg N)$. To prove this claim, we first observe that for all positive r, with probability $1 - O(e^{-2r^2})$ the four quadrants in an ith subproblem each have $m/4 \pm O(r\sqrt{m})$ live cells, where $m = \Omega(\lg N)$ is the number of live cells in the subproblem overall. This can be proved by a standard analysis of binomial distributions. Since each row and column of the subarray to be constructed in the subregion contains $\Theta(\sqrt{m})$ cells, every variation of $\sqrt{m}$ cells in the four quadrants causes an $O(1)$ distortion in the corresponding channels. Thus, with probability $1 - O(e^{-2r^2})$, an $\Omega(\lg N)$-cell subproblem contributes $O(r)$ distortion of wires in these channels. There are $O(\lg N)$ $\Omega(\lg N)$-cell subproblems that can contribute to the distortion of a given wire in the center channel, and the distortion caused by each can be treated as an independent random variable.

Using standard combinatorial arguments involving sums of random variables, it is now possible to show that the sum of the distortions is $O(\lg N)$ with probability $1 - O(1/N)$. That is because the probability of having total distortion $r_1 + r_2 + \cdots + r_s = t$ is at most

$$\sum^{r_1+r_2+\cdots+r_s=t} e^{-2r_1^2-2r_2^2-\cdots-2r_s^2}$$

where $s = O(\lg N)$. Applying standard asymptotic inequalities, this can be seen to be at most

$$\binom{t+s}{s} e^{-2s(t/s)^2} \le \left(\frac{(t+s)e}{se^{2(t/s)^2}}\right)^s.$$

For $t \ge s$, this is at most $e^{-t^2/s} \le e^{-t}$. Thus, the probability

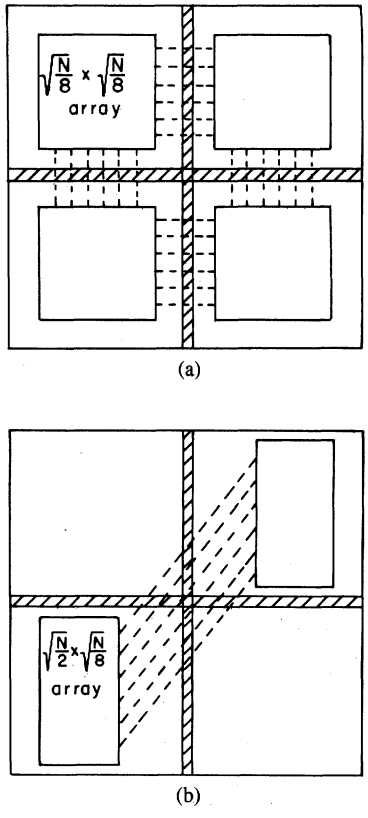


Fig. 18. (a) A distribution of live cells which might allow a narrow center channel. (b) A distribution of live cells which requires a wide center channel.

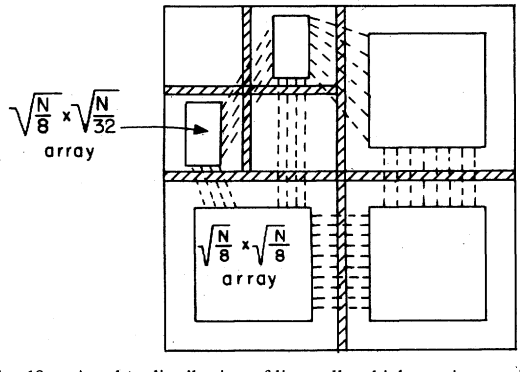


Fig. 19. Another distribution of live cells which requires a wide center channel.

that the distortion in a channel is greater than 2 lg N is very small.

The same analysis can be used to prove a high-probability bound of $O(\sqrt{\lg N \lg m})$ on the distortion of wires that connect subarrays of size $m = \Omega(\lg N)$. In this case, $s = O(\lg m)$, and $e^{-t^2/s}$ becomes very small when $t = 2\sqrt{s \lg N} = \Theta(\sqrt{\lg m \lg N})$. Thus, it is sufficient that the channels between subproblems with m cells have width $O(\sqrt{\lg N \lg m})$. By summing over all $\Omega(\lg N)$-sized subproblems, it can be checked that at this point the *average* channel width on the wafer is $O(1)$, which is because the channels inside $O(\lg N)$-sized subproblems have not been used at all. The constant average channel width can be achieved as a maximum without increasing the length of any wire by more than $O(\lg N)$. The idea is to distribute the $\Theta(\sqrt{\lg N \lg m})$-width channels across neighboring unused channels. As the details of this argument are somewhat tedious, we have omitted them. This concludes the first stage of the analysis.

The analysis of wires that link cells within a $\Theta(\lg N)$-cell subproblem differs substantially from the preceding analysis because live cells within a small region can have arbitrarily irregular distributions with high probability. The regions of irregularity are small enough, however, that the worst case distributions are not really all that bad. For example, if a $\Theta(\lg N)$-cell region has the structure shown in Figs. 18(b) or 19, then the maximum distortion of a wire at the top level of the recursion is just $O(\sqrt{\lg N})$.

In fact, the analysis of Section V ensures that the algorithm constructs a two-dimensional array in each $m = \Theta(\lg N)$-cell region using wires of length $O(\sqrt{m} \lg m) = O(\sqrt{\lg N} \lg \lg N)$ and channels of width $O(\lg m) = O(\lg \lg N)$. Thus, the entire two-dimensional array is constructed using wires of length $O(\lg N \lg \lg N)$ and channels of width $O(\lg \lg N)$. The extra lg lg N factor in the wire length bound comes about because a wire with $O(\lg N)$ distortion crosses $O(\lg N)$ channels, each of width $O(\lg \lg N)$. ■

The wire length analysis of the *algorithm* in Theorem 11 is fairly tight. For example, it can be shown that the algorithm requires wires of length $\Omega(\lg N)$ with high probability. Thus, if the lower bound in Theorem 6 is to be achieved, a different algorithm must be discovered. It may be possible to improve the channel width bound, however. Any improvement in Theorem 10 would directly lead to an improvement in the channel width bounds in both Theorem 11 and the next theorem, which shows how to construct a two-dimensional array from *most* of the live cells on a wafer.

Theorem 12: With probability $1 - O(1/N)$ a two-dimensional array can be constructed from any constant fraction (less than 1) of the live cells on an N-cell wafer using wires of length $O(\sqrt{\lg N \lg \lg N})$ and channels of width $O(\lg \lg N)$.

Proof: The key idea is to partition the wafer into $N/c \lg N$ square regions, each containing $m = c \lg N$ cells where c is a sufficiently large constant. We expect each such region to contain $1/2c \lg N$ live cells. Of course, there will be variations. Using standard techniques, however, it is not difficult to show that with probability $1 - O(1/N)$ each of the regions contains at least $m' = 1/2m - \sqrt{m}\sqrt{\lg N} = 1/2c(1 - 2/\sqrt{c}) \lg N$ live cells. Using the technique of Theorem 10, we can therefore construct an m'-cell two-dimensional array in each region using wires of length $O(\sqrt{m} \lg m) = O(\sqrt{\lg N} \lg \lg N)$ and channels of width $O(\lg m) = O(\lg \lg N)$. The $N/c \lg N$ two-dimensional arrays are then connected together into one large array with $1/2N(1 - 2/\sqrt{c})$ live cells. The added wires have length at most $O(\sqrt{\lg N} \lg \lg N)$, and the channel width is not substantially increased. ■

For each of the two previous results, the channels on the wafer have width $O(\lg \lg N)$. The next theorem shows that with high probability a two-dimensional array can be constructed from many of the live cells on a wafer using channels of unit width. Furthermore, the lower bound of $\Omega(\sqrt{\lg N})$ on wire length given in Theorem 6 is achieved by this construction.

Theorem 13: With probability $1 - O(1/N)$ at least a fraction $\Omega(1/\lg \lg^2 N)$ of the live cells on an N-cell wafer can be connected into a two-dimensional array using wires of length $O(\sqrt{\lg N})$ and channels of unit width.

Proof: The proof is similar to that of Theorem 12. As before, we partition the wafer into square regions with $c \lg N$ cells each. The constant c must be chosen large enough to ensure that with high probability each region contains $\lg N$ live cells. We next partition each $c \lg N$-cell region into square subregions with $c' \lg \lg^2 N$ cells each. Consider all pairs of indexes i and j in the range $1 \leq i, j \leq \sqrt{c'} \lg \lg N$. For a given region of $c \lg N$ cells, at least one pair (i, j) satisfies the condition that at least $1/c$ of the cells in the (i, j) positions of the subregions are alive. (Otherwise, it is impossible for $\lg N$ of the cells in the region to be alive.) Notice that by ignoring those cells not in the (i, j) positions of a subregion, the (i, j)-positioned cells, together with all of the channels of the region, form a "subwafer" with

1) $m = c \lg N/c' \lg \lg^2 N$ cells total,
2) at least $m/c = \lg N/c' \lg \lg^2 N$ live cells, and
3) channels of width $\sqrt{c'} \lg \lg N = O(\lg m)$.

By choosing c' large enough, the technique of Theorem 10 can be applied to construct within each $c \lg N$-cell region a two-dimensional array with $\lg N/c' \lg \lg^2 N$ cells using wires of length $O(\sqrt{m} \lg m) = O(\sqrt{\lg N})$. These arrays can then be easily connected together to form a two-dimensional array with $N/cc' \lg \lg^2 N$ cells and wires of length $O(\sqrt{\lg N})$. ■

By setting $m = \Omega(N/\lg \lg^2 N)$, it can be checked that Theorem 13 achieves the lower bounds for wire length proved in Theorem 6. The cell utilization, however, leaves something to be desired.

We have summarized the results of this section in Table I. Each bound is achieved with probability $1 - O(1/N)$, where N is the number of cells on the wafer, p is the probability that a particular cell is dead, and s is the side length of each cell. (Wires are assumed to have width one.)

TABLE I
BOUNDS ON WIRE LENGTH AND CHANNEL WIDTH FOR TWO-DIMENSIONAL ARRAYS

Portion of Live Cells Used	Wire Length	Channel Width
All	$O(\log_{1/p} N(s + \log_2 \log_{1/p} N))$	$O(\log_2 \log_{1/p} N)$
Constant fraction (less than one)	$O(\sqrt{\log_{1/p} N}\,(s + \log_2 \log_{1/p} N))$	$O(\log_2 \log_{1/p} N)$
$\Omega(1/(\log_2 \log_{1/p} N)^2)$	$O(\sqrt{\log_{1/p} N})$	1

VII. Related Models and Problems

The problem of incorporating all the live cells of a wafer into a linear array so that the maximum wire length is minimized has been studied in more standard graph-theoretic models and has come to be known as the *bottleneck traveling salesman* problem [8]. In addition, the wafer-scale model of N cells which fail independently with probability $1/2$ is essentially equivalent to the well-studied geometric model in which N points are thrown down randomly in a unit square [2], [11], [15], [31], [40], [48]. Thus, the algorithms for constructing linear arrays described in Section III can also be applied to the bottleneck traveling salesman problem in the geometric unit-square model. For example, our results can be modified to show that with high probability, all of the points in the unit square can be joined into a Hamiltonian path using wires of length $O(\sqrt{\lg N}/\sqrt{N})$, the least possible. In addition, *most* of the points can be joined in a linear array using wires of length $O(1/\sqrt{N})$, again the least possible. Although neither of these results have been explicitly stated in the literature, the first result is really just a minor extension to the prior work of Karp [15] and Bentley *et al.* [2]. The latter result of joining most of the points differs substantially from previous work, however. To the best of our knowledge, the only result of a similar nature is due to Erdös and Renyi [6], who showed that most graphs with N vertices and N edges have large connected components.

The analysis used to prove Theorem 11 has recently been found to have applications to a variety of matching and bin packing problems. For example, the same analysis can be used to bound the expected minimum maximum edge length in a matching of N random points in a unit square to N fixed points arranged in a square grid [1]. (In this model, wires have no width and the $\lg \lg N$ factors disappear.) This problem in turn has important applications in the analysis of the expected behavior of a variety of bin packing algorithms [16], [41].

The techniques developed in this paper can also be extended to prove results about average wire length. For linear arrays, it is easily argued that the average wire length is constant. This is because the overall area is linear in the number of wires. By similar reasoning, it can be seen that the square of the average channel width is an upper bound on the average wire length. Hence, the two-dimensional arrays constructed in Section VI have average wire length $O(\lg \lg^2 N)$ with high probability. In models where wires have no width, an $O(\lg \lg N)$ average wire length can be achieved for two-dimensional arrays.

The problems considered heretofore in this paper also have an interpretation in a purely graph-theoretic model. Suppose we are given a two-dimensional grid graph, and assume that each node in the grid has independently a probability p of being *bad*. We wish to find a subgraph of the grid that contains only *good* nodes and that forms a smaller two-dimensional grid. For example, Fig. 20 illustrates the embedding of a good 3×3 grid in a partially bad 4×4 grid.

The objectives we might choose to optimize in such a problem are

1) maximizing the size of the good grid,
2) minimizing the maximum distance between neighbors,
3) minimizing the total distance between all pairs of neighbors, and
4) minimizing the maximum number of times an edge in the partially bad grid is utilized.

These parameters roughly correspond in the wafer-scale model to the usage of live cells, maximum wire length, total wire length, and maximum channel width, respectively.

The beauty of the graph-theoretic model, however, is that it generalizes naturally to broader classes of graphs. For example, the same kinds of questions can be reasonably asked about the class of k-dimensional grids for any k, the class of complete binary trees, or the class of hypercubes. In

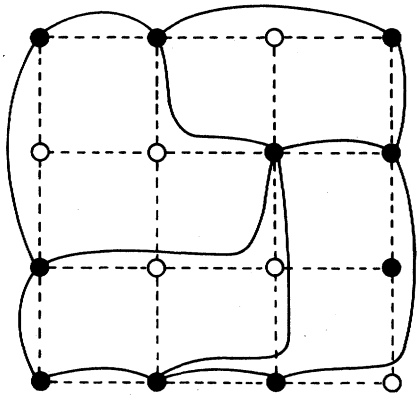


Fig. 20. A good 3 × 3 grid formed in a partially bad 4 × 4 grid. Good nodes are denoted by black dots.

each case, the appropriate problem might be

"A network in the class is given, but some portion of the nodes fail. How do we use the edges and good nodes of the network to construct a somewhat smaller network of the same type?"

For linear graphs the answer to the question is straightforward. This paper provides a starting point for two-dimensional grids. For other classes, the answers are as yet unknown. Also of interest is the problem of embedding a graph from one class in a partially bad graph from a different class. Research in this area should lead to a greater understanding of the fault tolerance of networks.

Despite the extended period of time since the research reported in this paper was completed, two central questions have remained unresolved. The first concerns the maximum channel width needed to embed an N-cell two-dimensional array in a worst case wafer with N live cells. In Section V, we prove an $O(\lg N)$ upper bound for this problem, but we know of no nontrivial lower bound. The second question concerns the maximum edge length needed for a random instance of the same problem. The bounds for this problem are $O(\lg N \lg \lg N)$ and $\Omega(\sqrt{\lg N})$, although the upper bound can be improved to $O(\lg N)$ if wire widths are ignored. Improvements to any of these bounds could well have applications to random matching [1] and bin packing [41] problems, in addition to wafer-scale integration of systolic arrays.

VIII. Concluding Remarks

For all the theoretical analysis in this paper, some of the algorithms described are quite practical. Not only are they fast, but they produce good results because the constants are small. For example, the methods of Section III can be used to show that there is a simple, linear-time algorithm to connect most of the live cells on an N-cell wafer into a linear array using wires of length 1, 2, or 3 and channels of width at most 2. In addition, the method from Section VI for connecting all the live cells into a two-dimensional array, modified to do table lookup on small subproblems, appears to be substantially better than what has been used in practice [42]. The same techniques should also work well for other natural structures, such as trees and hypercube-based networks.

Some of the problems mentioned in this paper have been studied independently by Greene and Gamal. In their recent paper [10], they prove most of the results found in Section III as well as the lower bound in Section IV. More recently, Greene [9] has shown how to remove the $O(\lg \lg N)$ terms in the upper bound for two-dimensional systolic arrays containing a constant fraction of the live cells. The result is difficult, and is solved using a clever probabilistic analysis of certain network flow problems.

Manning [27], [28], Hedlund [12], Koren [17], and Fussell and Varman [7] have also looked at the basic problem of constructing functioning arrays from defective arrays. Each gives algorithms but little theoretical or statistical analysis. Rosenberg [36], [37] has investigated issues of fault tolerance, and Bhatt and Leighton [3] recently extended the results in Section V to general graphs.

Acknowledgment

The authors would like to thank the members of the M.I.T. Lincoln Laboratory Restructurable VLSI project headed by J. Raffel for acquainting them with the details of their work and for providing the photographs in Figs. 2, 3, and 4. Special thanks in particular to J. Siskind and J. Southard of Lincoln Lab for exploring the practical aspects of their algorithms with them. They would like to thank J. Spencer, who suggested the idea behind the proof of Theorem 6. S. Bhatt, B. Chor, F. Chung, M. Fischer, A. El Gamal, J. Greene, and G. Miller were helpful in discussing the technical content of the paper. They would also like to thank J. Bentley, P. Kanellakis, D. Karp, C. Papadimitriou, A. Rosenberg, M. Steele, and L. Valiant for pointing out relevant literature. Lastly, they would like to thank the referee for making many helpful suggestions and criticisms.

References

[1] M. Ajtai, J. Komlos, and M. Tusnady, "On optimal matchings," unpublished manuscript, 1982.

[2] J. Bentley, B. Weide, and A. Yao, "Optimal expected-time algorithms for closest point problems," *ACM Trans. Math. Software*, vol. 6, pp. 563–580, Dec. 1980.

[3] S. N. Bhatt and F. T. Leighton, "A framework for solving VLSI graph layout problems," *J. Comput. Syst. Sci.*, vol. 28, pp. 300–343, Apr. 1984.

[4] S. N. Bhatt and C. E. Leiserson, "How to assemble tree machines," in *Proc. 14th ACM Symp. Theory Comput.*, May 1982, pp. 77–84.

[5] G. Bilardi, M. Pracchi, and F. P. Preparata, "A critique and appraisal of VLSI models of computation," in *Proc. CMU Conf. VLSI Syst. Computations*, H. T. Kung, R. Sproull, and G. Steele, Eds., Oct. 1981, pp. 81–88.

[6] P. Erdös and A. Renyi, "On the evolution of random graphs," *Magyar Tud. Akad. Math. Kut. Int. Kozl.*, vol. 5, pp. 17–61, 1960.

[7] D. Fussell and P. Varman, "Fault-tolerant wafer-scale architectures for VLSI," in *Proc. 9th Annu. IEEE/ACM Symp. Comput. Architecture*, Apr. 1982, pp. 190–198.

[8] M. Garey and D. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. San Francisco, CA: Freeman, 1979.

[9] J. Greene, "Configuration of VLSI arrays in the presence of defects," Ph.D. dissertation, Stanford Univ., Stanford, CA, 1983.

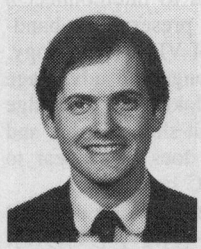
[10] J. Greene and A. Gamal, "Area and delay penalties for restructurable VLSI arrays," *J. Ass. Comput. Mach.*, to be published.

[11] J. Halton and R. Terada, "A fast algorithm for the Euclidean traveling salesman problem, optimal with probability one," *SIAM J. Comput.*, vol. 11, pp. 28–46, Feb. 1982.
[12] K. Hedlund, personal communication.
[13] A. Itai, C. H. Papadimitriou, and J. L. Szwarcfiter, "Hamiltonian paths in grid graphs," *SIAM J. Comput.*, vol. 11, pp. 676–686, Nov. 1982.
[14] J. J. Karaganis, "On the cube of a graph," *Canad. Math. Bull.*, no. 11, pp. 295–296, 1968.
[15] R. M. Karp, "Probabilistic analysis of partitioning algorithms for the traveling salesman problem in the plane," *Math. Oper. Res.*, vol. 2, pp. 209–244, 1977.
[16] R. M. Karp, M. Luby, and A. Marchetti-Spaccamela, "Probabilistic analysis of multidimensional bin packing algorithms," in *Proc. 16th ACM Symp. Theory Comput.*, 1983.
[17] I. Koren, "A reconfigurable and fault-tolerant VLSI multiprocessor array," in *Proc. 8th Annu. IEEE/ACM Symp. Comput. Architecture*, May 1981, pp. 425–431.
[18] H. T. Kung and C. E. Leiserson, "Systolic arrays (for VLSI)," in *Sparse Matrix Proceedings, 1978*, I. S. Duff and G. W. Stewart, Eds. Soc. Industr. Appl. Math., 1978, pp. 256–282.
[19] C. Y. Lee, "An algorithm for path connection and its applications," *IRE Trans. Electron. Comput.*, vol. EC-10, pp. 346–365, Sept. 1961.
[20] F. T. Leighton, *Complexity Issues in VLSI: Optimal Layouts for the Shuffle-Exchange Graph and Other Networks*. Cambridge, MA: M.I.T. Press, 1983.
[21] ——, "New lower bound techniques for VLSI," *Math. Syst. Theory*, vol. 17, pp. 47–70, 1984.
[22] F. T. Leighton and A. L. Rosenberg, "Three-dimensional circuit layouts," Mass. Inst. Technol., Cambridge, MA, VLSI Tech. Memo. 82-102, 1982; see also, ——, *SIAM J. Comput.*, to be published.
[23] C. E. Leiserson, "Area-efficient graph layouts (for VLSI)," in *Proc. 21st Annu. IEEE Symp. Foundations Comput. Sci.*, Oct. 1980, pp. 270–281.
[24] ——, *Area-Efficient VLSI Computation*. Cambridge, MA: M.I.T. Press, 1983.
[25] R. J. Lipton and R. E. Tarjan, "A separator theorem for planar graphs," presented at the Conf. Theoret. Comput. Sci., Univ. Waterloo, Aug. 1977.
[26] J. Logue, W. Kleinfelder, P. Lowy, J. Moulic, and W. Wu, "Techniques for improving engineering productivity of VLSI designs," in *Proc. IEEE Int. Conf. Circuits and Comput.*, 1980.
[27] F. Manning, "Automatic test, configuration, and repair of cellular arrays," Ph.D. dissertation, Mass. Inst. Technol., Cambridge, MA, M.I.T. Project MAC, June 1975.
[28] ——, "An approach to highly integrated, computer-maintained cellular arrays," *IEEE Trans. Comput.*, vol. C-26, June 1977.
[29] C. A. Mead and L. A. Conway, *Introduction to VLSI Systems*. Reading, MA: Addison-Wesley, Oct. 1980.
[30] R. C. Prim, "Shortest connection networks and some generalizations," *Bell Syst. Tech. J.*, vol. 36, pp. 1389–1401, 1957.
[31] M. O. Rabin, "Probabilistic algorithms," in *Algorithms and Complexity: New Directions and Recent Results*, J. F. Traub, Ed. New York: Academic, 1976, pp. 21–39.
[32] J. I. Raffel, "On the use of nonvolatile program links for restructurable VLSI," in *Proc. Caltech Conf. VLSI*, Jan. 1979, pp. 95–104.
[33] J. I. Raffel, A. H. Anderson, G. H. Chapman, K. H. Konkle, B. Mathur, A. M. Soares, and P. W. Wyatt, "A wafer-scale integrator using restructurable VLSI," Joint Special Issue on VLSI, *IEEE J. Solid-State Circuits* and *IEEE Trans. Electron Devices*, Feb. 1985.
[34] A. L. Rosenberg, "Encoding data structures in trees," *J. Ass. Comput. Mach.*, vol. 26, pp. 668–689, Oct. 1979.
[35] ——, "Three-dimensional integrated circuitry," in *Proc. CMU Conf. VLSI Syst. Computations*, H. T. Kung, R. Sproull, and G. Steele, Eds., Oct. 1981, pp. 69–80.
[36] ——, "The Diogenes approach to testable fault-tolerant networks of processors," Dep. Comput. Sci., Duke Univ., Durham, NC, Tech. Rep. CS-1982-6.1, May 1982.
[37] ——, "On designing fault-tolerant arrays of processors," Duke Univ., Durham, NC, Tech. Rep. CS-1982-14, 1982.
[38] A. L. Rosenberg and L. Snyder, personal communication, 1982.
[39] M. Sekanina, "On an ordering of the set of vertices of a connected graph," *Pub. Faculty of Sci., Univ. Brno*, Czechoslovakia, no. 412, pp. 137–142, 1960.
[40] M. I. Shamos, "Computational geometry," Ph.D. dissertation, Yale Univ., New Haven, CT, May 1978.
[41] P. Shor, "The average case analysis of some on-line algorithms for bin packing," in *Proc. 25th IEEE Conf. Foundations Comput. Sci.*, 1984.
[42] J. Siskind, private communication, Dec. 1981.
[43] L. Snyder, "Overview of the CHiP computer," in *VLSI 81*, J. Gray, Ed. New York: Academic, Aug. 1981, pp. 237–246.
[44] J. Spencer, personal communication, 1982.
[45] C. D. Thompson, "Area-time complexity for VLSI," in *Proc. 11th ACM Symp. Theory Comput.*, May 1979, pp. 81–88.
[46] ——, "A complexity theory for VLSI," Ph.D. dissertation, Dep. Comput. Sci., Carnegie-Mellon Univ., Pittsburgh, PA, 1980.
[47] L. G. Valiant, "Universality considerations in VLSI circuits," *IEEE Trans. Comput.*, vol. C-30, pp. 135–140, Feb. 1981.
[48] B. W. Weide, "Statistical methods in algorithm design and analysis," Ph.D. dissertation, Dep. Comput. Sci., Carnegie-Mellon Univ., Pittsburgh, PA, Aug. 1978.

Tom Leighton (M'81) was born in Washington, DC, in 1956. He received the B.S.E. degree in electrical engineering and computer science from Princeton University, Princeton, NJ, in 1978 and the Ph.D. degree in mathematics from the Massachusetts Institute of Technology, Cambridge, in 1981.

From 1981 to 1983, he was a Bantrell Postdoctoral Fellow at the Laboratory for Computer Science, Massachusetts Institute of Technology, and in 1982 he joined the faculty at the Massachusetts Institute of Technology as an Assistant Professor of Mathematics. He is currently an Associate Professor of Mathematics and a member of the Laboratory for Computer Science. He has published numerous papers on VLSI design, parallel algorithms and architectures, sequential algorithms, graph theory, and combinatorial methods. He is currently most interested in mathematical problems arising in the areas of VLSI design, parallel computation on fixed-connection networks, and the probabilistic analysis of algorithms.

Prof. Leighton has served on several conference technical committees and is currently an editor for *Algorithmica, Combinatorica,* and the *SIAM Journal of Computing*. He is a member of the ACM Special Interest Group on Automata and Computability Theory, the IEEE Computer Society, and the SIAM Special Interest Group on Discrete Mathematics. He was recently appointed a Presidential Young Investigator by the National Science Foundation.



Charles E. Leiserson (M'83) received the B.S. degree in computer science and mathematics from Yale University, New Haven, CT, in 1975 and the Ph.D. degree in computer science from Carnegie-Mellon University, Pittsburgh, PA, in 1981.

He is currently an Associate Professor of Computer Science and Engineering at the Massachusetts Institute of Technology, Cambridge. As a graduate student at Carnegie-Mellon, he wrote the first paper on systolic architectures with his advisor H. T. Kung, for which they received a U.S. patent. His Ph.D. dissertation "Area-Efficient VLSI Computation," which dealt with the design of systolic systems and with the problem of determining the VLSI area of a graph, won the first ACM Doctoral Dissertation Award. In 1981, he joined the faculty of the Theory of Computation Group in the Laboratory for Computer Science, Massachusetts Institute of Technology. He has authored or coauthored papers on systolic architectures, graph layout, digital circuit optimization, analysis of algorithms, computer-aided design, placement and routing, computer architecture, and parallel computation. His principal interest is in the theoretical foundation of parallel computation, especially as it relates to engineering reality.

Prof. Leiserson is a member of the Association for Computing Machinery, and he serves on the ACM General Technical Achievement Award Committee, which selects the Turing Award winner. In 1985 he received a Presidential Young Investigator Award from the National Science Foundation.