LAYOUTS FOR THE SHUFFLE-EXCHANGE GRAPH AND LOWER BOUND TECHNIQUES FOR VLSI

FRANK THOMSON LEIGHTON

Laboratory for Computer Science
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

June 1982

This report was prepared with the assistance of a National Science Foundation Graduate Fellowship and grant #N00014-80-C-0622 from the Defense Advanced Research Projects Agency.

This empty page was substituted for a blank page in the original document.

LAYOUTS FOR THE SHUFFLE-EXCHANGE GRAPH AND LOWER BOUND TECHNIQUES FOR VLSI

by

Frank Thomson Leighton

Submitted to the Department of Mathematics on August 7, 1981
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

ABSTRACT

The thesis is divided into two parts. In the first part, we describe and analyze several new VLSI layouts for the shuffle-exchange graph. These include

1) an asymptotically optimal, $\Theta(N^2/\log^2 N)$-area layout for the N-node shuffle-exchange graph, and
2) several practical layouts for small shuffle-exchange graphs.

The new layouts require substantially less area than previously known layouts and can serve as the basis for designing large scale shuffle-exchange chips.

In the second part of the thesis, we develop general methods for proving lower bounds on the layout area, crossing number, bisection width and maximum edge length of VLSI networks. Among other things, we use these methods to find

1) an N-node planar graph which has layout area $\Theta(N\log N)$ and maximum edge length $\Theta(N^{1/2}/\log^{1/2} N)$,
2) an N-node graph with an $O(N^{1/2})$-separator which has layout area $\Theta(N\log^2 N)$ and maximum edge length $\Theta(N^{1/2}\log N/\log\log N)$, and
3) an N-node graph with an $O(N^{\alpha})$-separator (for $\alpha > 1/2$) which has maximum edge length $\Theta(N^{\alpha})$.

The area results indicate that some graphs with $O(N^{1/2})$-separators (and, in particular, some planar graphs) do *not* have linear-area layouts, thus disproving a popular conjecture. The edge length bounds indicate that the layouts of some networks must have very long wires (possibly as long as the width of the layout).

Key words: crossing number, graph separator, layout area, maximum wire length, parallel computation, shuffle-exchange graph, Thompson model, Very Large Scale Integration (VLSI)

Thesis supervisor: Gary L. Miller
Title: Associate Professor of Mathematics

ACKNOWLEDGEMENT

In what follows, I give thanks to the many individuals who have contributed so much, both professionally and personally, to the completion of this thesis. First, I would like to thank Dan Kleitman, Gary Miller and Ron Rivest for serving on my thesis committee. They have each been of great help in preparing this document. In particular, I would like to thank Gary Miller for the extraordinary amount of time he has devoted to me during my tenure at M.I.T. He has been a great source of inspiration and encouragement, and has always made himself available for discussion of my work.

During the course of the last year, several other people have also contributed to the content and writing of the thesis. In particular, I would like to thank Sandeep Bhatt, Dan Hoey, Paris Kanellakis, Margaret Lepley, Charles Leiserson, John Reif, Michael Rodeh, Arnold Rosenberg, Jim Shearer, Clark Thompson and Les Valiant for their helpful remarks and suggestions.

This research was primarily funded through a graduate fellowship from the National Science Foundation. I am deeply indebted to them for their support during the last three years. Additional support was provided by the Defense Advanced Research Projects Agency under Contract #N00014-80-C-0622.

I would also like to take this opportunity to thank some of the many people who have, in the past, gone out of their way to further my educational development. In particular, I would like to recognize the efforts of Forman Acton, Len Adleman, Elva Aucklund, Alan Goldman, Charles Johnson, Phoebe Knipling, Steve Maurer, Dorothy Nelson, Frank Peterson and Dorothy Shriver. I would especially like to thank Carl Hammer, David Labovitz and Morris Newman for sacrificing many hours of their valuable time in order to advise, teach and encourage me.

Lastly, I would like to thank my wife Joanne for her support during the writing of the thesis and my entire family for the lifetime of sacrifices that they have made in order to insure that I obtained the best education available. Without their endless love and support, this thesis would not have been possible.

CONTENTS

Introduction 2

PART I: LAYOUTS FOR THE SHUFFLE-EXCHANGE GRAPH 7

Chapter 1: Review of Known Layouts 8

1.1 Thompson's Layout 8
1.2 Hoey and Leiserson's Complex Plane Diagram 9
1.2.1 Definition 9
1.2.2 Properties 10
1.2.3 An $O(N^2/logN)$-Area Layout 11

Chapter 2: Layouts Based on the Complex Plane Diagram 12

2.1 A Straightforward $O(N^2/logN)$-Area Layout 12
2.2 An Improved $O(N^2/log^{3/2}N)$-Area Layout 15
2.3 Other Layouts 18

Chapter 3: More Sophisticated Layouts 19

3.1 Preliminaries 19
3.2 A Near-Optimal Layout 22
3.2.1 Location of the Nodes 22
3.2.2 Insertion of the Edges 23
(a) Exchange Edges Which Link Nodes in Different Rows 24
(b) Exchange Edges Which Link Nodes in the Same Row 24
3.3 An Optimal $O(N^2/log^2N)$-Area Layout 26
3.3.1 More Definitions 26
3.3.2 Location of the Nodes 27
3.3.3 Insertion of the Edges 28
(a) Exchange Edges Which Link Nodes in Different Rows 28
(b) Exchange Edges Which Link Nodes in the Same Row 29
3.3.4 Comments 30
3.4 Layouts With Additional Edges 30
3.4.1 Shift Edges 30
3.4.2 Reverse Edges 31
3.4.3 Transpose Edges 32
Appendix: Proofs of Lemmas 3-1 Through 3-4 33

Chapter 4: Practical Layouts 40

4.1 Preliminaries 40
4.1.1 A Closer Look at the Thompson Model 41

4.1.2 A Class of Practical Layouts 42
4.2 Optimization Techniques 42
4.2.1 Ordering the Necklaces 43
4.2.2 Inserting the Exchange Edges 44
4.2.3 Additional Savings 46
4.3 Optimal Layouts 46
4.4 Other Layouts 49

PART II: LOWER BOUND TECHNIQUES FOR VLSI 50

Chapter 5: Review of Known Techniques 51

5.1 Area Bounds 51
5.2 Edge Length Bounds 55

Chapter 6: Network Constructions 59

6.1 The *2*-Dimensional Mesh of Trees 59
6.1.1 Definition 59
6.1.2 Properties 60
6.1.3 Applications 61
(a) Matrix-Vector Multiplication 61
(b) Sorting 62
(c) Switching 63
6.2 The *r*-Dimensional Mesh of Trees 63
6.2.1 Definition 63
6.2.2 Properties 64
6.2.3 Application to Matrix Multiplication 64
6.2.4 A Further Generalization 65
6.3 The Tree of Meshes 65
6.3.1 Definition 65
6.3.2 Properties 66
6.3.3 Applications 67
(a) Embeddings of Planar Graphs 67
(b) Embedding of $M_{2,n}$ in T_n 70
6.4 The Augmented Mesh of Trees 72

Chapter 7: Crossing Number Arguments 74

7.1 The Relationship Between Crossing Number and Layout Area 74
7.2 A General Method for Proving Lower Bounds 75
7.3 Applications 77
7.3.1 Lower Bounds for the Shuffle-Exchange Graph 77
7.3.2 Lower Bounds for the *2*-Dimensional Mesh of Trees 79
7.3.3 Lower Bounds for the *r*-Dimensional Mesh of Trees 82

7.4 Further Methods 83
7.4.1 A Combinatorial Lower Bound for Crossing Numbers 83
7.4.2 Applications 85
7.4.3 An Upper Bound for Crossing Numbers 87

Chapter 8: Wire Area Arguments 89

8.1 Lower Bounds for the *2*-Dimensional Mesh of Trees 89
8.2 Lower Bounds for the Tree of Meshes 93
8.3 Lower Bounds for a Restricted Class of Binary Tree Layouts 94

Conclusion 97

Index 98

References 100

Addendum 104

This empty page was substituted for a blank page in the original document.

LAYOUTS FOR THE SHUFFLE-EXCHANGE GRAPH

AND

LOWER BOUND TECHNIQUES FOR VLSI

This empty page was substituted for a blank page in the original document.

INTRODUCTION

The recent engineering advances in Very Large Scale Integrated (VLSI) circuitry have made it possible to wire tens of thousands of transistors onto a single chip. In the near future, it is expected that fabrication of chips containing *millions* of transistors will be commonplace [MC80]. In order that this massive computational resource be efficiently utilized, theoretical researchers have been actively trying to answer such questions such as:

1) "What is a good model for VLSI chip design and computation?,"

2) "What communications networks can best perform important operations such as sorting, matrix multiplication and discrete Fourier transform?" and

3) "What is the best method of laying out a network on a chip?."

Several models have been proposed for VLSI computation [T80,LS81,CM81]. The most widely accepted is due to Thompson and is known as the *Thompson model* [T79,T80]. Thompson's model of a VLSI chip is quite simple. The chip is presumed to consist of a grid of vertical and horizontal *tracks* which are spaced apart by unit intervals. Processors are viewed as points and are located only at the intersection of grid tracks. Wires are routed through the tracks in order to connect pairs of processors. Although a wire in a horizontal track is allowed to cross a wire in a vertical track, pairs of wires are not allowed to overlap for any distance (i.e., in they cannot overlap in the same track). Further, wires are not allowed to overlap processors to which they are not linked. As an example, we have drawn a Thompson model layout of a *4*-processor network in Figure 1.

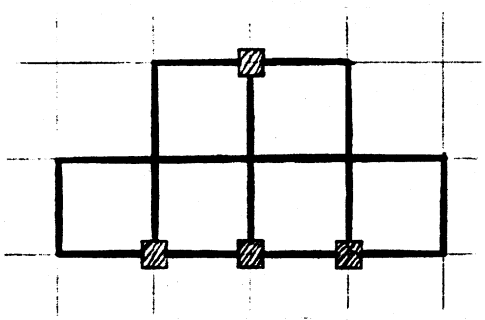

Figure 1: *A Thompson model layout of a 4-processor network in which each processor is linked to every other processor.*

Much has also been accomplished in the way of finding good communications networks for VLSI. For example, the complete binary tree [MC80], the *2*-dimensional mesh [TK77,KL78,MC80], the cube-connected-cycles graph [PV79] and the shuffle-exchange graph [S71,L75,L76,NS79,P80,S80,SR80a,T79,T80] are all known to be capable of performing a wide range of operations. The shuffle-exchange graph, in particular, is an incredibly powerful and efficient communications network. Among other things, it can be used to compute discrete Fourier transforms, multiply matrices, sort lists and evaluate polynomials. Except for sorting (which requires $O(log^2 N)$ time), these operations require no more than logarithmic time and constant space per processor. This is *exponentially* faster than the running times of the corresponding sequential algorithms and the corresponding parallel algorithms on networks such as the *2*-dimensional mesh. As, in addition, the processors required for these operations are quite simple, the shuffle-exchange network is very well suited for VLSI implementation on a chip.

The *shuffle-exchange graph* comes in various sizes. In particular, there is an N-node shuffle-exchange graph for every N which is a power of two. Each node of the $(N=2^k)$-node shuffle-exchange graph is associated with a unique k-bit binary string $a_{k-1}\cdots a_0$. Two nodes w and w' are linked via a *shuffle edge* if w' is a left or right cyclic shift of w (i.e., if $w = a_{k-1}\cdots a_0$ and $w' = a_{k-2}\cdots a_0 a_{k-1}$ or $w' = a_0\cdots a_{k-1}a_1$, respectively). Two nodes w and w' are linked via an *exchange edge* if w and w' differ only in the last bit (i.e., if $w = a_{k-1}\cdots a_1 0$ and $w' = a_{k-1}\cdots a_1 1$ or vice-versa). As an example, we have drawn the *8*-node shuffle-exchange graph in Figure 2. Note that the shuffle edges are drawn with solid lines while the exchange edges are drawn with dashed lines. We shall follow this convention throughout the thesis.

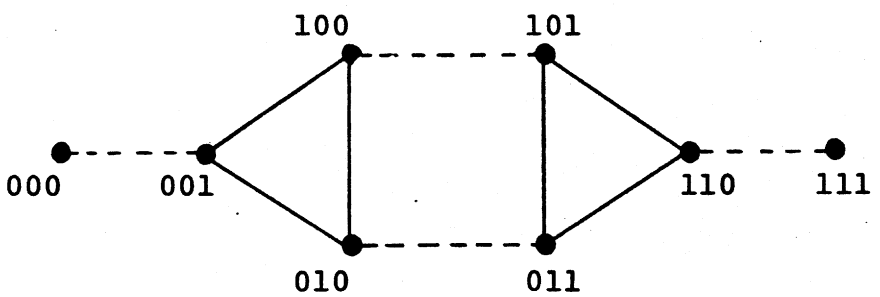

Figure 2: *The 8-node shuffle-exchange graph.*

The third question of interest to VLSI researchers ("What is the best method of laying out a network on a chip?") has proved to be, by far, the most difficult. It is also the subject of this thesis. In order to answer the question for a particular network, we must do the following three things:

1) decide what it means for a layout to be "good,"

2) find a "good" layout for the network, and

3) prove that the layout is as "good" as possible.

Most people agree that a "good" layout is one which does not require much area. This is quite reasonable since small layouts are easier to wire on a chip, cost less and have far higher yields than layouts with larger amounts of area. Recently, there has also been interest in designing layouts with short wires. Although wire length considerations are not as important as area considerations, it is possible that layouts with long wires may be less efficient and run slower (due to longer transmission times) than layouts with shorter wires. Both quantities are easily expressed in terms of the Thompson model, which is nice from a mathematical point of view. For example, the *layout area* of a network is the minimum amount of area required to lay out the network in the Thompson model. (The *area of a layout* in the Thompson model is defined to be the product of the number of vertical tracks and the number of horizontal tracks which contain a processor or wire segment of the layout.) Similarly, the *maximum edge length* of a network is the minimum amount of wire which is needed to embed the longest *edge* in any Thompson model layout of the network.

Good layouts are known for several communications networks; including the complete binary tree [MR79,PRS81,BL81], the 2-dimensional mesh and the cube-connected-cycles graph [PV79]. The known layouts for the shuffle-exchange graph, however, are not very good. Thompson [T80] was the first to find a nontrivial layout for the shuffle-exchange graph. In particular, he found an $O(N^2/\log^{1/2}N)$-area layout of the N-node shuffle-exchange graph. He also showed that *any* layout for the N-node shuffle-exchange graph must have at least $\Omega(N^2/\log^2 N)$ area. Hoey and Leiserson [HL80] improved the upper bound by finding an $O(N^2/\log N)$-area layout for the N-node shuffle-exchange graph. Neither Thompson's nor Hoey and Leiserson's layouts are practical, however, and neither meets Thompson's asymptotic lower bound.

In Part I of the thesis, we find good layouts for the shuffle-exchange graph. In particular, we describe an asymptotically optimal $O(N^2/\log^2 N)$-area layout for the N-node shuffle-exchange graph. Although the layout is not optimal for small values of N, we show how it can be modified in order to produce good layouts for small shuffle-exchange graphs. As these layouts *are* practical, it should now be possible to build a shuffle-exchange chip.

Finally, we are left with the task of proving that a layout which appears to be good is, in fact, optimal. Although Thompson [T79,T80], Vuillemin [V80] and Lipton and Sedgewick [LS81] have all shown how to prove area lower bounds for certain computationally useful networks (such as the shuffle-exchange graph), it is not known how to prove such lower bounds in general. For example, no nontrivial lower bounds have been found for the class of graphs which have $O(N^{1/2})$-separators. (This class includes the very important class of planar graphs.) Nor have any methods been discovered for proving nontrivial lower bounds on the maximum edge length of a network.

In Part II of the thesis, we describe several techniques for proving good layout area and maximum edge length lower bounds. In particular, we concentrate on finding good lower bounds for the crossing number, wire area and maximum edge crossing of a network. The *crossing number* of a graph is the minimum number of pairs of edges which must cross in any drawing of the graph in the plane. The *maximum edge crossing* of a graph is the largest number of edges which *must* be crossed by some edge in any drawing of the graph. The *wire area* of a network is simply the minimum amount of wire which must be used to embed the network in the Thompson model. It is clear that for any network,

$$\textit{crossing number} \leq \textit{wire area} \leq \textit{layout area}$$

and also that

$$\textit{maximum edge crossing} \leq \textit{maximum edge length}.$$

In addition, the crossing number, wire area and maximum edge crossing are worth minimizing independent of layout area and maximum edge length considerations. This is due to the fact that

1) chips with a large number of wire crossings (and, in particular, those with wires which cross many other wires) have substantially more problems with

capacitive coupling (i.e., interference between overlapping wires) than do chips with fewer crossings, and

2) chips with high wire area cost more and experience lower yields than do chips with lesser wire area.

Unfortunately, the results of Part II indicate that the crossing number and wire area are usually as large (up to a constant factor) as the layout area. In addition, the maximum edge crossing is often nearly as large as the side length of the chip. More importantly, however, crossing number and wire area arguments can be used to prove better lower bounds on the layout area and maximum edge length than were possible with existing techniques. In particular, we will use such arguments to find

1) an N-node planar graph which has layout area $\Theta(N \log N)$ and maximum edge length $\Theta(N^{1/2}/\log^{1/2} N)$,

2) an N-node graph with an $O(N^{1/2})$-separator which has layout area $\Theta(N \log^2 N)$ and maximum edge length $\Theta(N^{1/2} \log N / \log\log N)$, and

3) an N-node graph with an $O(N^{\alpha})$-separator (for $\alpha > 1/2$) which has maximum edge length $\Theta(N^{\alpha})$.

The area results indicate that not all graphs with $O(N^{1/2})$-separators (and, in particular, not all planar graphs) can be laid out in linear area, thus disproving a popular conjecture. The edge length bounds indicate that layouts of certain networks must have some very long wires (possibly even as long as the side length of the layout). Taken together, these results answer all of the previously open questions concerning layout area and maximum edge length of VLSI networks with known separators.

This empty page was substituted for a blank page in the original document.

PART I

LAYOUTS FOR THE SHUFFLE-EXCHANGE GRAPH

CHAPTER 1

REVIEW OF KNOWN LAYOUTS

In this chapter, we review the known layouts of the shuffle-exchange graph. In section 1.1, we describe Thompson's [T80] straightforward $O(N^2/\log^{1/2}N)$-area layout. This is followed in section 1.2 by a detailed description of Hoey and Leiserson's complex plane diagram. The complex plane diagram is very helpful in finding good layouts for the shuffle-exchange graph. For example, Hoey and Leiserson [HL80] have used the diagram to find an $O(N^2/\log N)$-area layout for the N-node shuffle-exchange graph. In Chapter 2, we will use the diagram to find a variety of layouts for the N-node shuffle-exchange graph including one which requires only $O(N^2/\log^{3/2}N)$ area. (Such a layout has also recently been found independently by Steinberg and Rodeh [SR80b].) The complex plane diagram will also be used in Chapter 4 as an aide in the construction of good practical layouts for small shuffle-exchange graphs.

1.1 Thompson's Layout

Thompson was the first to investigate VLSI layouts for the shuffle-exchange graph. In his thesis [T80], he showed that *any* layout for the N-node shuffle-exchange graph requires at least $\Omega(N^2/\log^2 N)$ area. (We reprove this fact using crossing number arguments in Part II of the thesis.) In addition, he described a layout requiring only $O(N^2/\log^{1/2}N)$ area. In what follows, we present Thompson's layout and give a simple proof that it does, in fact, require just $O(N^2/\log^{1/2}N)$ area.

Given any k-bit string w, define the *size* of w to be the number of *1*-bits it contains. For example, the size of *10110* is *3*. Thompson's idea was to lay out the $N=2^k$ nodes of the shuffle-exchange graph on a straight line in order of nondecreasing size. It is easily seen that shuffle edges link nodes which have the same size and that exchange edges link nodes which have sizes differing by one. Thus the edges of such a layout are relatively short. In particular, the number of horizontal tracks needed to embed all of the edges is at most $O(\max_{1 \le s \le k} B_s)$ where

B_s is the number of nodes of size s. This is due to the fact that at most $O(B_{s-1}+B_s+B_{s+1})$ edges can cross any vertical cut of the layout which is located between a pair of nodes of size s.

It is easy to show that $B_s = C(k,s)$ for each s where

$$C(k,s) = k!/[s!(k-s)!]$$

is the well-known function for binomial coefficients. It is also well-known that $C(k,s)$ achieves its maximum value at $s=k/2$ for any k. Using standard asymptotic analysis, it is easily shown that $C(k,k/2) \sim \Theta(2^k/k^{1/2})$ for large k. (For a good review of such techniques, see Bender and Orszag's book [BO78].) Thus Thompson's layout requires only $O(N/log^{1/2}N)$ horizontal tracks. Since at most *3* vertical tracks are needed to embed the vertical portions of the edges incident to any given node, we can conclude that Thompson's layout has area $\Theta(N^2/log^{1/2}N)$.

1.2 Hoey and Leiserson's Complex Plane Diagram

In [HL80], Hoey and Leiserson observed that there is a very natural embedding of the shuffle-exchange graph in the complex plane. In what follows, we describe this embedding (henceforth referred to as the *complex plane diagram*) and point out some of its more important properties. In addition, we give a brief description of the method used by Hoey and Leiserson to transform the diagram into an $O(N^2/logN)$-area layout for the N-node shuffle-exchange graph.

1.2.1 Definition

Let $\delta_k = e^{2\pi i/k}$ denote the *kth* primitive root of unity. Given any k-bit binary string $w = a_{k-1} \cdots a_0$, let $p(w)$ be the map which sends w to the point

$$p(w) = a_{k-1}\delta_k^{k-1} + \cdots + a_1\delta_k + a_0$$

in the complex plane. As each node of the $(N=2^k)$-node shuffle-exchange graph corresponds to a k-bit binary string, it is possible to use the map to embed the shuffle-exchange graph in the complex plane. For example, we have done this for the *32*-node shuffle-exchange graph (whence $k=5$) in Figure 1-1. As is usual, we have drawn the shuffle edges with solid lines and the exchange edges with dashed lines. For simplicity, each node is labeled with its value instead of its *5*-bit binary string. (By the *value* of a node, we mean the numerical value of the associated k-bit binary string.)

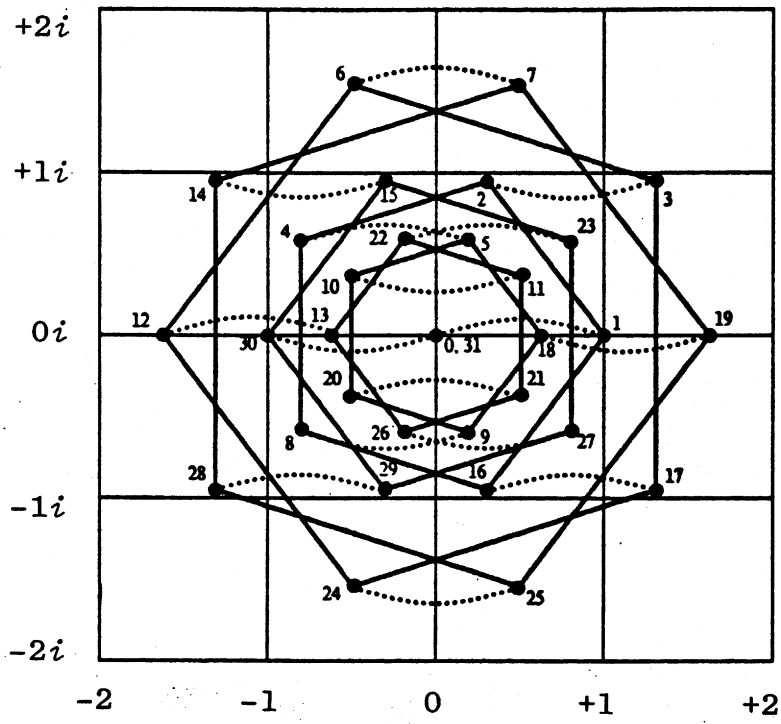

Figure 1-1: *The complex plane diagram for the 32-node shuffle-exchange graph. (Taken from* [HL80].)

1.2.2 Properties

Examination of Figure 1-1 indicates that the complex plane diagram has some very interesting properties. First, it is apparent that the shuffle edges occur in cycles (which we call *necklaces*) which are symmetrically placed about the origin. This phenomenon is easily explained by the following identity:

$$\begin{aligned}\delta_k p(a_{k-1}\cdots a_0) &= a_{k-1}\delta_k{}^k + a_{k-2}\delta_k{}^{k-1} + \cdots + a_1\delta_k{}^2 + a_0\delta_k \\ &= a_{k-2}\delta_k{}^{k-1} + \cdots + a_0\delta_k + a_{k-1} \\ &= p(a_{k-2}\cdots a_0 a_{k-1}).\end{aligned}$$

Thus traversal of a shuffle edge corresponds to a $2\pi/k$ rotation in the complex plane.

Except for degenerate cases, the preceding identity also indicates that each necklace is composed of k nodes, each a cyclic shift of the other. Such necklaces are called *full necklaces. Degenerate necklaces* contain fewer than k nodes and, because they must have some symmetry, are mapped entirely to the origin of the

complex plane diagram. For example, {*00000*} and {*0101, 1010*} are degenerate necklaces while both {*101, 011, 110*} and {*11100, 11001, 10011, 00111, 01110*} are full.

It will often be convenient to refer to a necklace by one of its nodes. In particular, we will use the notation $\langle w \rangle$ to indicate the *necklace generated by w.* This is simply the collection of cyclic shifts if *w.* For example, the necklace generated by *101* is $\langle 101 \rangle = \{101, 011, 110\}$.

Exchange edges are also embedded in a very regular fashion by the complex plane diagram. In fact, each exchange edge is embedded as a horizontal line segment of unit length. This phenomenon is explained by the identity

$$p(a_{k-1} \ldots a_1 0) + 1 = a_{k-1}\delta_k^{k-1} + \ldots + a_1\delta_k + 1$$
$$= p(a_{k-1} \ldots a_1 1).$$

In some cases, several exchange edges are contained in the same horizontal line of the diagram. Such lines are called *levels.* For example, there are *9* levels in the diagram of the *32*-node shuffle-exchange graph shown in Figure 1-1. We will use the properties of levels in Chapter 2 to find an $O(N^2/\log^{3/2}N)$-area layout for the *N*-node shuffle-exchange graph. They will also be used in Chapter 4 to find good practical layouts for small shuffle-exchange graphs.

1.2.3 An $O(N^2/\log N)$-Area Layout

In [HL80], Hoey and Leiserson showed how to use the complex plane diagram to construct an $O(N^2/\log N)$-area layout for the *N*-node shuffle-exchange graph. Their method was very involved, however, and we have chosen not to include it here. The basic idea is to use the structural properties of the complex plane diagram to find an $O(N/\log^{1/2}N)$-separator for the *N*-node shuffle-exchange graph whenever *N* is of the form 2^{2^r} for some $r \geq 0$. The separator can then used to construct an $O(N^2/\log N)$-area layout by using Leiserson's general layout technique for graphs with known separators [L80a].

Shortly after writing [HL80], Hoey and Leiserson found a far simpler $O(N^2/\log N)$-area layout for the *N*-node shuffle exchange graph which was, in addition, valid for all *N*. By the that time, however, we (as well as several others) had also observed that the complex plane diagram could be used to find a simple layout for the shuffle-exchange graph. This layout is described in Chapter 2.

CHAPTER 2

LAYOUTS BASED ON THE COMPLEX PLANE DIAGRAM

In this chapter, we present several layouts of the shuffle-exchange graph which are based on Hoey and Leiserson's complex plane diagram. We commence in section 2.1 with a straightforward $O(N^2/logN)$-area layout of the N-node shuffle-exchange graph. As we mentioned in Chapter 1, this layout has also been discovered by many others (including Hoey and Leiserson). In section 2.2, we show how the layout can be modified so as to require only $O(N^2/log^{3/2}N)$ area. The latter layout was also discovered independently by Steinberg and Rodeh [SR80b]. We conclude the chapter by mentioning an additional $O(N^2/log^{3/2}N)$-area layout as well as a layout which might require even less area.

2.1 A Straightforward $O(N^2/logN)$-Area Layout

In this section, we describe a straightforward layout of the shuffle-exchange graph which requires only $O(N^2/logN)$ area. The layout is formed from a grid of levels and necklaces which we refer to as the *level-necklace grid.* Each row of the grid corresponds to a level of the complex plane diagram. The columns are divided into consecutive column pairs, each pair corresponding to a necklace. In particular, the leftmost column of each column pair corresponds to that part of the necklace which is contained in the left half of the complex plane. Similarly, the rightmost column corresponds to the part of the necklace contained in the right half of the complex plane. We assume that the rows are ordered from top to bottom so as to be consistent with the natural ordering of the levels in the complex plane but (for the time being) place no restrictions on the left-to-right order of the necklaces.

Each node of the shuffle-exchange graph is placed at the intersection of the row and column of the grid which correspond to the level and part of the necklace (left half or right half) to which it belongs in the complex plane diagram. For example, we have done this for a random ordering of the necklaces of the *32*-node shuffle-exchange graph in Figure 2-1.

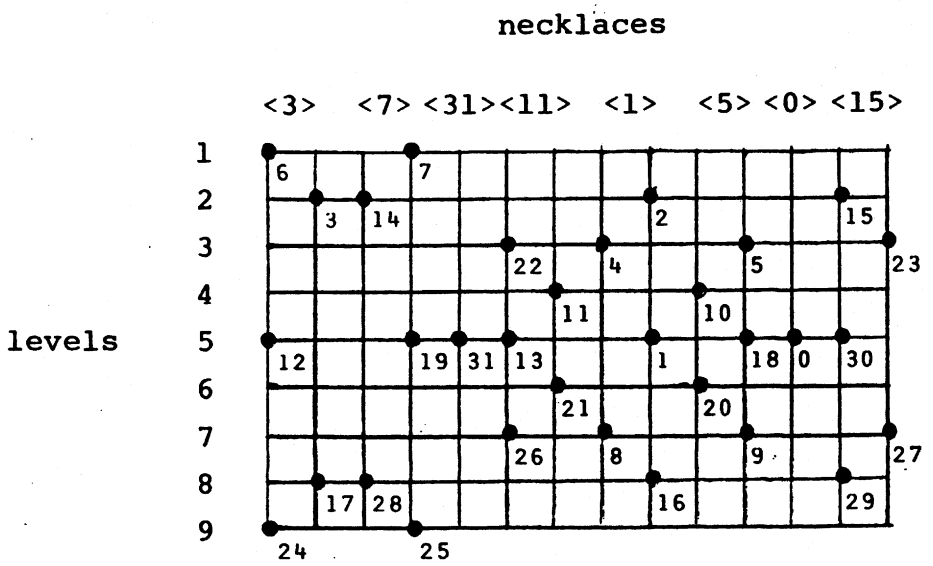

Figure 2-1: *A level-necklace grid for the 32-node shuffle-exchange graph.*

Notice that we used just one vertical track to embed the necklaces *<0>* and *<31>* in the grid. As each necklace contains just one node, it is clear that this is sufficient. In general, necklaces which are mapped to the origin by the complex plane diagram are a nuisance since they become lumped together in a single point of the level-necklace grid. Fortunately, there are relatively few such nodes. In particular, Hoey and Leiserson showed the following.

Lemma 2-1 (Hoey and Leiserson [HL80]): *At most* $O(N/\log N)$ *nodes of the N-node shuffle-exchange graph are mapped to the origin of the complex plane diagram.*

Proof: Every node which is mapped to the origin of the complex plane diagram is adjacent (via an exchange edge) to a node at position *(1,0)* or *(-1,0)*. Any node which is not mapped to the origin is contained in some full necklace, at most two nodes of which are contained in positions *(1,0)* or *(-1,0)*. Thus for every pair of nodes which are mapped to the origin, there are at least $k = \log N$ nodes which are not mapped to the origin. Thus at most $O(N/k) = O(N/\log N)$ nodes can be mapped to the origin □

Since at most $O(N/\log N)$ nodes are mapped to the origin, we can (for the time being) ignore them. They can always be inserted later at a cost of at most $O(N/\log N)$ additional vertical and horizontal tracks. Since any layout of the shuffle-exchange graph which we will consider will have at least $\Omega(N/\log N)$ vertical

and horizontal tracks, the added tracks can increase the area of the final layout by at most a constant factor. We will also use this strategy in Chapter 3 when we ignore several $O(N/logN)$-sized sets of nodes.

Since each full necklace contains at most $k = logN$ nodes, it is easy to see that the N-node shuffle-exchange graph has at most $O(N/logN)$ full necklaces. Thus at most $O(N/logN)$ vertical tracks are needed to embed all of the shuffle edges in the level-necklace grid. It is also easy to show that at most N horizontal tracks are needed to embed all of the exchange edges (one track is used for each exchange edge). Thus the total area of the layout for the N-node shuffle-exchange graph is $O(N^2/logN)$. As an example, we have added the edges of the *32*-node shuffle-exchange graph to the level-necklace grid in Figure 2-1 to produce the layout shown in Figure 2-2. Note that we have omitted *<0>* and *<31>* in this layout since they are mapped to the origin of the complex plane diagram.

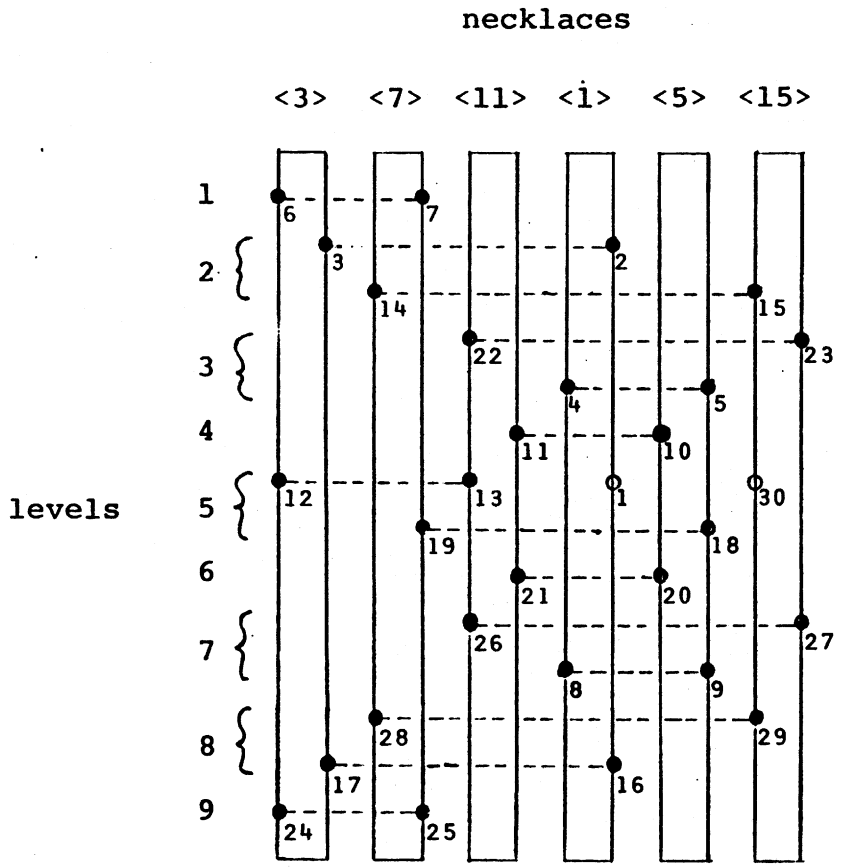

Figure 2-2: *Layout produced from the level-necklace grid shown in Figure 2-1.*

2.2 An Improved $O(N^2/\log^{3/2}N)$-Area Layout

It is possible to improve the layout described in section 2.1 by reducing the number of horizontal tracks needed to embed the exchange edges. This can be done in two ways. First, exchange edges which are in the same level of the complex plane diagram but which do not overlap in the level-necklace grid can be inserted on the same horizontal track. As more exchange edges are inserted on the same track, fewer total tracks will be needed to embed all of the exchange edges. Secondly, the necklaces can be re-ordered so as to increase the average number of exchange exchange edges which can be inserted on each horizontal track.

Although we do not know how to best order the necklaces in general, we have found several orderings which yield $O(N^2/\log^{3/2}N)$-area layouts for the N-node shuffle-exchange graph. For instance, we will show in what follows that such a layout can be constructed by arranging the necklaces from left to right in order of nondecreasing size. (The *size of a necklace* is simply defined to be the size of any of its nodes.) This observation has also been made by Steinberg and Rodeh in [SR80b].

In order to bound the number of horizontal tracks needed to insert the exchange edges, we will show that the maximum overlap of exchange edges on each level occurs in between necklaces of size $k/2$. Since the maximum overlap of exchange edges on each level is an upper bound on the number of horizontal tracks needed to insert the exchange edges on that level, we can thus conclude that the total number of horizontal tracks needed to insert all of the exchange edges is at most

$$O(B_{k/2}) \sim O(N/\log^{1/2}N) .$$

Thus the resulting layout will have area at most $O(N^2/\log^{3/2}N)$.

It is not immediately clear why the maximum overlap on each level occurs between nodes of size $k/2$, however. In what follows, we break up each level into sublevels (for which the analysis is easier) and show that the maximum overlap on each sublevel occurs between necklaces of size $k/2$. Before doing this, however, we must introduce some further notation.

Consider a node of the form $a_{k-1}\cdots a_1 0$ for which either $a_{k-i}=0$ or $a_i=0$ or both for each $i\leq k$. We will refer to such a node as *basis node*. A node $b_{k-1}\cdots b_0$ is said to be *generated* by the basis node $a_{k-1}\cdots a_0$ if

1) $b_{k-i}=a_{k-i}$ and $b_i=a_i$ whenever $a_{k-i}=a_i$ for $1 \leq i \leq k$, and

2) $b_{k-i}=b_i$ whenever $a_{k-i}=a_i=0$ for $1 \leq i \leq k$.

For example, *10000* generates *10001*, *11100* and *11101* but not *11111*.

It is not difficult to show that if u generates v, then both u and v are on the same level of the complex plane diagram. For example, let $u = a_{k-1} \cdots a_0$ and $v = b_{k-1} \cdots b_0$ and observe that

$$\begin{aligned} p(v) - p(u) &= (b_{k-1} - a_{k-1})\,\delta_k^{k-1} + \cdots + (b_1 - a_1)\,\delta_k + (b_0 - a_0) \\ &= c_{k-1}\delta_k^{k-1} + \cdots + c_1\delta_k + c_0 \end{aligned}$$

where $c_{k-i}=c_i$ for each i, $1 \leq i \leq k$. Since δ_k^{k-i} is the complex conjugate of δ_k^{i} for $1 \leq i \leq k$, we can conclude that $p(v) - p(u)$ is a real number and thus that u and v are in the same level of the complex plane diagram.

It is also easy to show that each node of the shuffle-exchange graph is generated by a unique basis node. In particular, the node which generates $b_{k-1} \cdots b_0$ can be found by

1) setting $b_0=0$ and (if k is even) setting $b_{k/2}=0$, and

2) setting $b_i=b_{k-i}=0$ for each i such that (originally) $b_i=b_{k-i}=1$.

Since exchange edges link nodes which are in the same sublevel, we can conclude from the preceding arguments that it is possible to partition each level of the complex plane diagram into sublevels so that the nodes in each sublevel are precisely the nodes generated by some basis node. We will now show that the maximum overlap at each sublevel occurs between necklaces of size $k/2$.

Since the necklaces have been arranged from left to right in order of nondecreasing size, we can use arguments similar to those of section 1.1 to conclude that the overlap of exchange edges between two nodes of size s in any sublevel is at most $O(\max_{1 \leq s \leq k} B_s')$ where B_s' is the number of nodes in that sublevel with size s. A straightforward counting argument shows that each basis node of size r generates

1) $C(k/2 - r, i)$ nodes of size $s=r+2i$ for any $i \leq k/2 - r$, and

2) $C(k/2 - r, i)$ nodes of size $s=r+2i+1$ for any $i \leq k/2 - r$

when k is odd, and

1) $C(k/2 - r - 1, i) + C(k/2 - r - 1, i - 1) = C(k/2 - r, i)$ nodes of size $s=r+2i$ for any $i \leq k/2 - r$, and

2) $2C(k/2 - r - 1, i)$ nodes of size $s=r+2i+1$ for any $i \leq k/2 - r - 1$

when k is even. We can therefore conclude that in *all cases*, the maximum value of B_s' occurs when $i = (k - 2r)/4$ and thus when $s=k/2$. This concludes the proof.

As an example, we have drawn such a layout for the *32*-node shuffle-exchange graph in Figure 2-3. Note that far fewer horizontal tracks are needed for this layout than are used for the layout in Figure 2-2. For completeness, we have included the necklaces *<0>* and *<31>* even though they are degenerate.

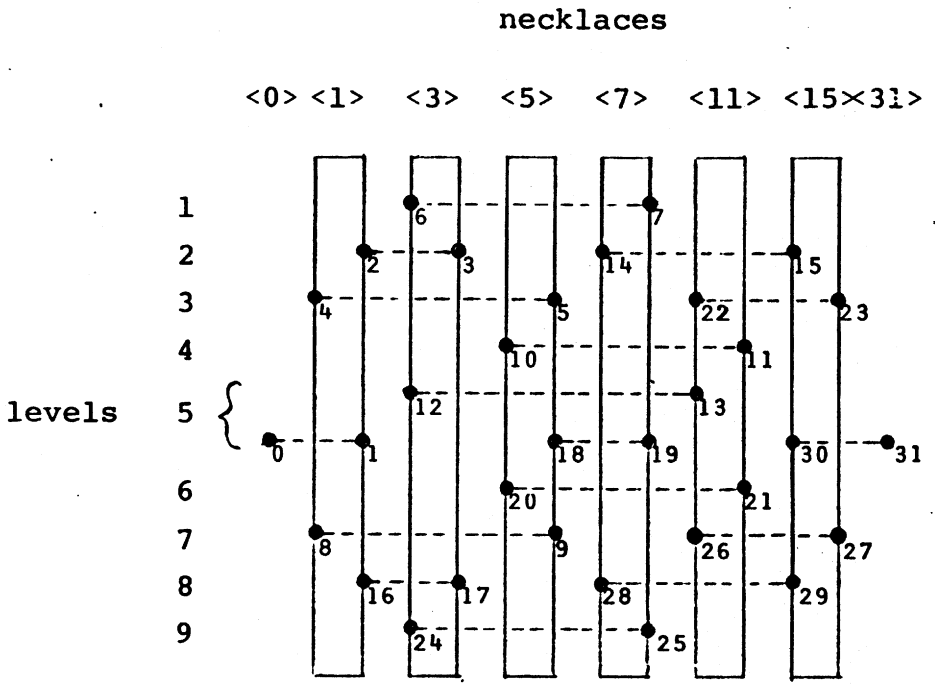

Figure 2-3: *An improved layout for the 32-node shuffle-exchange graph.*

2.3 Other Layouts

It is not difficult to find other orderings of the necklaces which produce $O(N^2/log^{3/2}N)$-area layouts for the N-node shuffle-exchange graph. For example, Lepley [LLM81] used standard statistical methods to show that the arrangement of necklaces from left to right in order of nondecreasing radius produces such a layout. (By the *radius of a necklace*, we mean the radius of the circle in the complex plane which contains the necklace.) The proof is similar to the one in section 2.2. In particular, it is shown that the maximum overlap in most levels occurs in the same place and that the total overlap of all of the levels at that point is $\Theta(N/log^{1/2}N)$.

Although we consider it likely that better orderings of the necklaces exist, we do not know of any ordering which (provably) results in a layout with less than $o(N^2/log^{3/2}N)$ area. There is another ordering of interest, however. That is the ordering of the necklaces according to the minimal number represented by each necklace. (The *minimum number represented* by a necklace is simply the smallest value of any node in the necklace.) Coincidentally, the layout displayed in Figure 2-3 has such an ordering. Using techniques which are developed in Chapter 3, it is possible to show that the combined maximum overlap of exchange edges in all levels is at most $O(NloglogN/logN)$ for this ordering. This is substantially better than the $O(N/log^{1/2}N)$ overlap found in previous orderings and also very close to the lower bound of $\Omega(N/logN)$. Unfortunately, we do not know how to show that the maximum overlap at each level occurs in the same place. In fact, it appears that this may not be the case. (We are deeply indebted to Kleitman for pointing out the possibility of such an improvement. Although we were not able use his idea in the context of complex plane diagram layouts, it was crucial to the development of the asymptotically optimal layout described in Chapter 3.)

For orderings which have a small combined maximal overlap but for which the maximal overlap at each level is difficult to compute (such as the ordering by minimal value represented), it may be possible to improve the situation by altering the level structure. As Miller pointed out to us, there are many possible levelings of the exchange edges. (By a *leveling*, we mean any arrangement of the exchange edges in levels which is consistent with the necklace structure of the complex plane diagram.) Although we have investigated several levelings, we have not found any (provably) better layouts for the shuffle-exchange graph by this method.

CHAPTER 3

MORE SOPHISTICATED LAYOUTS

In section 3.3 of this chapter, we describe an asymptotically optimal $O(N^2/log^2N)$-area layout for the N-node shuffle-exchange graph. Unlike the previously described layouts, the optimal layout is fairly sophisticated and requires a substantial amount of preliminary machinery. Most of the necessary definitions and lemmas are included in section 3.1. In section 3.2, we describe and analyze a near-optimal preliminary version of the optimal layout. The optimal layout is then described in section 3.3. In section 3.4, we extend the methods developed in earlier sections in order to show that certain useful supergraphs of the N-node shuffle-exchange graph can also be laid out in $O(N^2/log^2N)$ area. We have also included an appendix to the chapter in which we prove Lemmas 3-1 through 3-4.

3.1 Preliminaries

The layouts described in this chapter are based on some important combinatorial properties of strings which contain long blocks of consecutive zeros. Before describing the layouts, however, it is useful to review some of these properties. In this section, we mention several combinatorial lemmas and definitions which will be heavily used in the analysis which follows later. As the proofs of the lemmas are somewhat complicated, they have been included in the appendix.

In what follows, we will be particularly interested in the size and location of the longest block of consecutive *0*-bits in the k-bit binary string associated with each node. In order that the size of this block be the same for all nodes within a necklace, we allow blocks to begin at the end and end at the beginning of a string. For example, the longest block of zeros in the string *01010* starts at the fifth bit and has length two.

Let $\Psi_k(t)$ denote the number of k-bit strings for which the longest block of consecutive zeros has length t. For example, $\Psi_3(2)=3$. The following combinatorial lemma provides a good asymptotic bound on the growth of $\Psi_k(t)$.

Lemma 3-1: *For* $(\log k)/2 + \log\ln k \leq t << k$ and $k \to \infty$,

$$\Psi_k(t) \sim 2^k (e^{-k2^{-(t+2)}} - e^{-k2^{-(t+1)}}).$$

In order to illustrate the important features of the function in Lemma 3-1, we have sketched a graph of $2^{-k}\Psi_k(t)$ versus t in Figure 3-1. The maximum of $2^{-k}\Psi_k(t)$ occurs at $t = \log k - 1$ whence

$$2^{-k}\Psi_k(t) = (e^{1/2} - 1)/e$$

$$\approx .23865.$$

For $t > \log k - 1$, $2^{-k}\Psi_k(t)$ decreases *exponentially* as t increases. For $t \leq \log k - 1$, $2^{-k}\Psi_k(t)$ decreases *doubly exponentially* as t decreases.

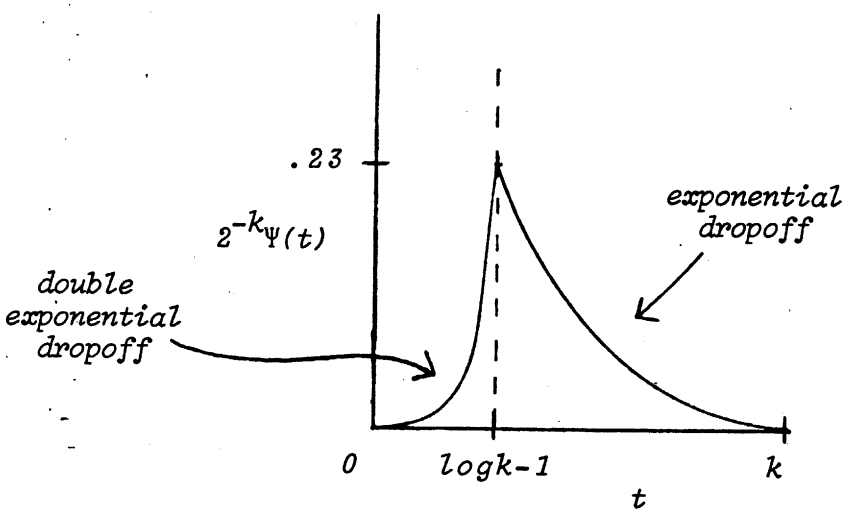

Figure 3-1: *Density of k-bit binary strings for which the longest block of consecutive zeros has length t.*

Roughly speaking, Lemma 3-1 states that the longest block of consecutive zeros in nearly *1/4* of all k-bit strings has length precisely $\log k - 1$. Further, there are not many strings of length k with substantially more than $\log k$ consecutive zeros and even fewer strings for which the longest block of consecutive zeros has length substantially less than $\log k$. This information is further quantified in the following lemma.

Lemma 3-2: *The number of k-bit strings for which the longest block of consecutive zeros has length less than* $\log k - \log\ln k - 1$ *or length greater than* $2\log k$

is at most $O(2^k/k) = O(N/\log N)$.

As we mentioned in Chapter 2, we may ignore $O(N/\log N)$-sized sets of nodes which have undesirable properties. As such nodes can be inserted with the addition of at most $O(N/\log N)$ vertical and horizontal tracks, we can always add them later without increasing the total area by more than a constant factor. By Lemma 3-2, we can thus henceforth consider only those nodes for which the longest block of zeros has length between *logk - loglnk - 1* and *2logk.*

We will also be interested in the size of the second longest block of consecutive zeros in each string. Usually, the size of the second longest block of zeros will be very close to the size of the longest block of zeros. We state this observation more precisely in the following lemma.

Lemma 3-3: *The sum over all necklaces of the difference in length between the longest and second longest blocks of consecutive zeros is at most* $O(N/\log N)$.

Using information about the size and location of blocks of zeros within the necklace, it is possible to distinguish one particular node in the necklace. More precisely, we define the *distinguished node of a necklace* to be the node containing the longest leading block of zeros. For example, *00101* is the distinguished node of *<01010>*. Should two or more nodes of a necklace begin with equal and maximal length blocks of zeros, then each node of the necklace contains at least two blocks of zeros of maximal length. In such cases, we distinguish that node for which the leading block of zeros is maximal and for which the second occurence of a maximal length block of zeros is as near as possible to the beginning of the string. For example, *01011* (not *01101*) is the distinguished node of the necklace *<10101>*. For some necklaces, such as *<111>* and *<1010101>*, there is no uniquely distinguished node. As we show in the following lemma, such necklaces are sufficiently rare that we need not consider them further.

Lemma 3-4: *At most* $O(N/\log N)$ *nodes are contained in necklaces which fail to have a uniquely distinguished node.*

We refer to the leading block of zeros of a distinguished node as the *primary block of zeros.* If a distinguished node has two or more maximal length blocks of zeros, then the maximal length block following the primary block is referrred to as the *secondary block of zeros.* These definitions can be easily extended to any node contained in a necklace which has a uniquely distinguished node. For example,

the primary block of zeros of *01010* starts in the fifth bit and has length two. Note that this string does *not* have a secondary block of zeros. As another example, we note that the secondary block of zeros in the string *11010* consists solely of the fifth bit. Note that the secondary block of zeros (if it exists) always has the same length as the primary block of zeros.

If the last bit of a node occurs in the primary block of zeros, we call that node a *primary node.* Similarly, if the last bit of a node occurs in the secondary block of zeros, we call the node a *secondary node.* For example, *10110* is a primary node, *11010* is a secondary node and *10010* is neither primary nor secondary.

Note that all primary and secondary nodes are necessarily even. (We say that a node is *even* if its last bit is *0* and *odd* if its last bit is *1.*) Note also that, by Lemma 3-2, we need only consider necklaces which contain between *logk - loglnk - 1* and *2logk* primary nodes. Such necklaces will also have at most *2logk* secondary nodes.

In what follows, we will represent nodes in terms of their corresponding distinguished nodes. More precisely, we use the notation $a_{k-1}\cdots a_{i+1}\overline{a_i}a_{i-1}\cdots a_0$ to denote the node $a_{i-1}\cdots a_0 a_{k-1}\cdots a_i$. For example, $00\overline{1}01$ denotes the node *10010.* Using this notation, a primary node has the form $0\cdots\overline{0}\cdots 0w$ while a secondary node has the form $0\cdots 0w'0\cdots\overline{0}\cdots 0w''$ where $0\cdots 0w$ and $0\cdots 0w'0\cdots 0w''$ are assumed to be distinguished nodes.

3.2 A Near-Optimal Layout

We are now prepared to describe a near-optimal preliminary version of the optimal layout. In section 3.3, we will show how to modify this layout in order to construct an optimal $O(N^2/\log^2 N)$-area layout for the N-node shuffle-exchange graph.

3.2.1 Location of the Nodes

The near-optimal layout is constructed from a $\log N \times O(N/\log N)$ grid of nodes. Each column of the grid corresponds to a necklace of the shuffle-exchange graph. The nodes of each necklace are ordered from top to bottom so that the *ith* node is a left cyclic shift of the *(i-1)st* node for each *i* and so that the distinguished node is placed in the bottom row. The necklaces are ordered from left to right so that the values of the distinguished nodes form an increasing sequence. For

example, we have constructed such a grid for the *32*-node shuffle-exchange graph in Figure 3-2. In the figure, we have represented each node in terms of the associated distinguished node. This representation readily illustrates the fact that the last bit of any node in the *ith* row corresponds to the *ith* bit of the associated distinguished node. Note that the necklaces *<00000>* and *<11111>* have not been included since they are degenerate.

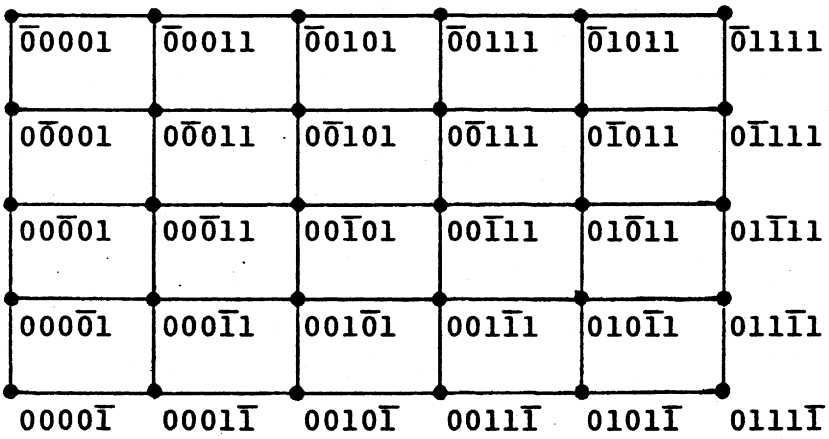

Figure 3-2: *The grid of nodes for the 32-node shuffle-exchange graph.*

3.2.2 Insertion of the Edges

It is easily observed that the shuffle edges can be inserted in the grid with the addition of $O(N/logN)$ vertical and *2* horizontal tracks. In the following, we will show that the exchange edges can be inserted with the addition of $O(NloglogN/logN)$ vertical and horizontal tracks. Thus the total area of the layout is $O(N^2(loglogN)^2/log^2N)$. This is only a factor of $O((loglogN)^2)$ off from the lower bound of $O(N^2/log^2N)$.

The analysis is divided into two parts. In part (a), we show that only $O(NloglogN/logN)$ exchange edges link nodes which are in *different* rows of the grid. Thus such edges can be inserted with the addition of at most $O(NloglogN/logN)$ vertical and horizontal tracks. In part (b), we conclude the analysis by showing that at most $O(N/logN)$ horizontal tracks are needed to insert the exchange edges which link two nodes in the *same* row.

(a) Exchange Edges Which Link Nodes in Different Rows

Consider an exchange edge which links two nodes that are in different rows of the grid. In particular, assume that the edge is incident to an even node in the *ith* row for some *i*. By definition, the even node can be represented as $w\bar{0}w'$ where $|w|=i-1$ and $w0w'$ is the distinguished node of $\langle w0w' \rangle$. The exchange edge is also incident to the odd node $w\bar{1}w'$. By assumption, $w\bar{1}w'$ is not located in the *ith* row and thus $w1w'$ is *not* a distinguished node. Since $w0w'$ *is* a distinguished node, we know that the *ith* bit of $w0w'$ (the bit that was changed in order to produce $w\bar{1}w'$) must be in the primary or secondary block of zeros of $w0w'$. Otherwise, the primary and (if it exists) secondary blocks of zeros of $w1w'$ would be identical in location and size to the primary and secondary blocks of $w0w'$. This would imply that $w1w'$ is also distinguished, a contradiction. Thus $w\bar{0}w'$ must be a primary or secondary node. As was previously mentioned, we can assume that each necklace has at most $2\log k = 2\log\log N$ primary and $2\log\log N$ secondary nodes. Thus at most $4\log\log N$ nodes in each necklace are both even and incident to an exchange edge which links nodes in different rows. Since every exchange edge is incident to an even node and since there are $O(N/\log N)$ necklaces, we can conclude that there are at most $O(N\log\log N/\log N)$ exchange edges which link nodes in different rows.

(b) Exchange Edges Which Link Nodes in the Same Row

We next show that those exchange edges which link two nodes that are in the same row can be inserted with the addition of at most $O(N/\log N)$ horizontal tracks. Once again, the analysis is divided into two parts. In the first part, we show that at most $O(N/\log N)$ exchange edges are contained in the first $\log k$ rows. Such edges can be trivially inserted with the addition of $O(N/\log N)$ horizontal tracks. In the second part, we show that only 2^{k-i} horizontal tracks are needed to insert the exchange edges in the *ith* row for any $i > \log k$. Since $\sum_{i=\log k+1}^{k} 2^{k-i} \leq 2^k/k = N/\log N$, this will be sufficient to show that at most $O(N/\log N)$ additional horizontal tracks are necessary to insert the remaining exchange edges.

Consider a necklace which has t primary nodes for some $t \leq \log k$. By definition, the nodes in the first t rows of such a necklace are all even. Thus, such a necklace can have at most $r = \log k - t$ odd nodes in the first $\log k$ rows. By Lemma 3-1, we know that there are

$$\Psi_k(t)/k \sim (2^k/k)(e^{-k2^{-t-2}} - e^{-k2^{-t-1}})$$

such necklaces for $(logk)/2 + loglnk \leq t << k$. By Lemma 3-2, we can assume that $t \geq logk - loglnk - 1$ and thus the total number of odd nodes occurring in the first $logk$ rows is at most

$$\begin{aligned}
&\sim \sum_{t = logk - loglnk - 1}^{\log k} (logk - t)(2^k/k)(e^{-k2^{-t-2}} - e^{-k2^{-t-1}}) \\
&= (2^k/k) \sum_{r=0}^{\log\ln k + 1} r(e^{-k2^{r-2-logk}} - e^{-k2^{r-1-logk}}) \\
&= (2^k/k) \sum_{r=0}^{\log\ln k + 1} r(e^{-2^{r-2}} - e^{-2^{r-1}}) \\
&= (2^k/k) \sum_{r=0}^{\log\ln k + 1} e^{-2^{r-2}} \\
&\leq (2^k/k) \sum_{r=0}^{\infty} e^{-2^{r-2}} \\
&= O(N/logN).
\end{aligned}$$

Since every exchange edge is incident to an odd node, the above bound implies that at most $O(N/logN)$ exchange edges are contained in the first $logk$ rows.

We next consider the number of horizontal tracks necessary to insert the exhange edges contained in the *ith* row for $i > logk$. This number is identical to the maximum number of exchange edges that can overlap each other at a single point of the *ith* row. In Figure 3-3, we illustrate the necessary conditions for two exchange edges to overlap in the *ith* row. All representations are in terms of distinguished nodes.

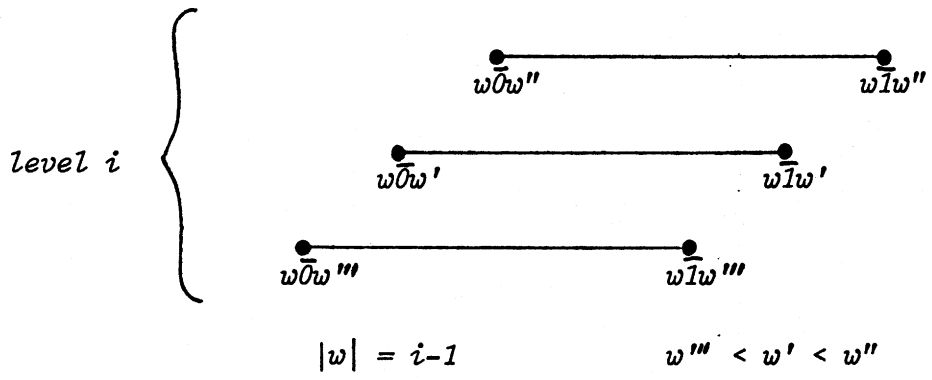

Figure 3-3: *Necessary conditions for exchange edges to overlap in the ith row.*

Note that the even end of an exchange edge is always to the left of the odd end. Also note that any node which occurs between $w\bar{0}w'$ and $w\bar{1}w'$ must be represented as $w\bar{0}w''$ where $w'' > w'$ or as $w\bar{1}w'''$ where $w''' < w'$. In either case, the exchange edge incident to the overlapped node extends beyond the exchange edge linking $w\bar{0}w'$ to $w\bar{1}w'$. Since there are at most $2^{k-i} - 1$ nodes between $w\bar{0}w'$ and $w\bar{1}w'$, these facts imply that at most 2^{k-i} exchange edges can overlap at any point of the *ith* row. This observation completes the argument that the near optimal layout requires only $O(N^2(loglogN)^2/log^2N)$ area.

3.3 An Optimal $O(N^2/log^2N)$-Area Layout

In this section, we will modify the layout described in section 3.2 in order to produce an optimal $O(N^2/log^2N)$-area layout for the N-node shuffle-exchange graph. In particular, we will relocate the primary and secondary nodes of each necklace so that they are closer to and in the same row as the nodes to which they are linked via an exchange edge. Before going into the details of this relocation, however, it is necessary to introduce some additional terminology.

3.3.1 More Definitions

In order to construct an optimal layout for the shuffle-exchange graph, we have found it necessary to break up each necklace into two or, possibly, three pieces. The *basic piece* of each necklace consists of all those nodes which are neither primary nor secondary. The *primary piece* of each necklace consists of the primary nodes while the *secondary piece* consists of the secondary nodes (if there are any). For example, the basic piece of $\langle 01011 \rangle$ is $\{01011, 01011, 01011\}$, the primary piece is $\{01011\}$, and the secondary piece is $\{01011\}$.

It is also necessary to extend the notion of a distinguished node to include pieces of necklaces. The *distinguished node of a basic piece* is the same as the distinguished node of the associated necklace. The *distinguished node of a primary piece* of a necklace is that node of the necklace which becomes distinguished when we ignore the primary block of zeros (i.e., when we temporarily replace the primary block of zeros in each node of the necklace with an equal-length block of ones). Similarly, the *distinguished node of a secondary piece* of a necklace is that node which becomes distinguished when we ignore the secondary block of zeros. For example, *010110111* is the distinguished node of the basic piece of $\langle 010110111 \rangle$, *011011101* is the distinguished node of the primary piece, and

011101011 is the distinguished node of the secondary piece. Note that the distinguished nodes of the primary and secondary pieces of any necklaces are necessarily odd nodes and thus are contained in the basic piece of the necklace.

It is important to note that some necklaces (such as *<01111>*) have a distinguished node but do not have a distinguished node for the primary or secondary piece of the necklace. Fortunately, arguments such as those used to prove Lemmas 3-3 and 3-4 can be used to show that at most O(*N/logN*) nodes are contained in such necklaces. Thus, we can assume henceforth that every piece of every necklace has an associated distinguished node.

3.3.2 Location of the Nodes

As in section 3.2, the layout is constructed from a *logN* x O(*N/logN*) grid of nodes. Each column of the grid corresponds to a piece of a necklace. The nodes of each piece are arranged within a column so that a node of the form $a_{k-1}\cdots\overline{a}_{k-i}\cdots a_0$ (where $a_{k-1}\cdots a_0$ is assumed to be the distinguished node of the associated piece) is placed in the *ith* row of the grid. Note that nodes in the basic piece of any necklace (these include all odd nodes) are in the same row as they were in the near-optimal layout described in section 3.2. The columns are ordered from left to right so that the values of the distinguished nodes of the associated pieces form a nondecreasing sequence. For example, we have constructed such a grid for $k=5$ in Figure 3-4.

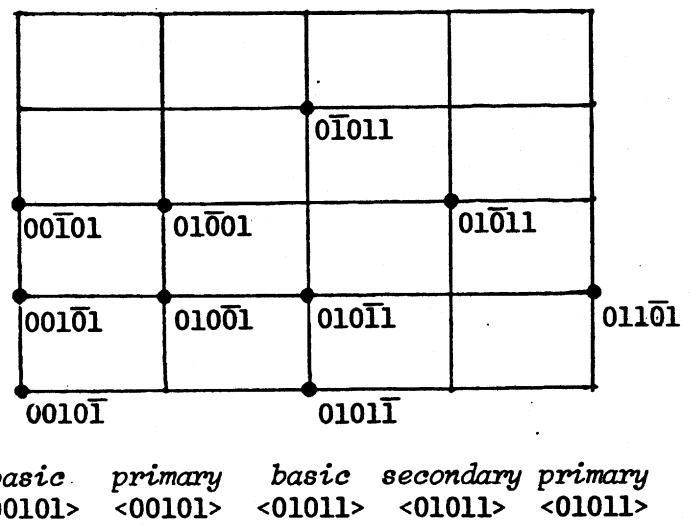

Figure 3-4: *Relocated nodes for the 32-node shuffle-exchange graph.*

Note that the necklaces $\langle 00001 \rangle$, $\langle 00011 \rangle$, $\langle 00111 \rangle$, and $\langle 01111 \rangle$ have not been included in Figure 3-4 since their associated primary pieces do not have distinguished nodes.

3.3.3 Insertion of the Edges

As each necklace is broken up into at most four *contiguous* pieces in the modified grid (the basic piece may have been broken up into two contiguous pieces), the shuffle edges can be inserted with the addition of at most $O(N/\log N)$ vertical and horizontal tracks. In what follows, we will show that at most $O(N/\log N)$ vertical and horizontal tracks are needed to insert all of the exchange edges as well. Thus the area of the layout will be $O(N^2/\log^2 N)$, which is optimal.

As before, we divide the analysis of the exchange edges into two parts. We first show that at most $O(N/\log N)$ exchange edges link nodes which are in different rows of the grid. Such edges can thus be trivially inserted with the addition of at most $O(N/\log N)$ vertical and horizontal tracks. We then show that those exchange edges which link two nodes in the same row can be inserted with the addition of only $O(N/\log N)$ horizontal tracks. The arguments will be very similar to those in section 3.2.2.

(a) Exchange Edges Which Link Nodes in Different Rows

Consider an exchange edge which links two nodes which are in different rows of the grid. Since only primary and secondary nodes have been relocated, we can conclude from the arguments of section 3.2.2a that the even node which is incident to the edge is either a primary or secondary node. In what follows, we will show that the even node is, in fact, a primary node.

Assume for the purposes of contradiction that the even node is a secondary node. Then this node can be represented as $w\bar{0}w'$ where $w0w'$ is the distinguished node of the secondary piece of $\langle w0w' \rangle$ and $|w| = i-1$ for some i. By definition, $w\bar{0}w'$ is located in the i*th* row of the grid and is linked to $w\bar{1}w'$ via the exchange edge. Since $w\bar{1}w'$ is odd, it is contained in the basic piece of $\langle w1w' \rangle$. By assumption, $w\bar{1}w'$ is not also in the i*th* row and thus $w1w'$ cannot be the distinguished node of $\langle w1w' \rangle$. Since the lengths of the two blocks of zeros in $w1w'$ created by switching the i*th* bit from *0* to *1* are less than the length of the primary block of zeros (in fact, the sum of their lengths is precisely one less than the length of the primary block), $w1w'$ will be the distinguished node of $\langle w1w' \rangle$

precisely when $w0w'$ is the node distinguished in $\langle w0w' \rangle$ by ignoring the secondary block of zeros. By definition, this is the case precisely when $w0w'$ is the distinguished node of the secondary piece of $\langle w0w' \rangle$. By assumption, $w0w'$ *is* the distinguished node of the secondary piece of $\langle w0w' \rangle$ and thus we can conclude that $w1w'$ is the distinguished node of $\langle w1w' \rangle$, a contradiction.

Next consider a *primary* node which is incident to an exchange edge linking two nodes in different rows of the grid. By the preceding arguments, this node must be of the form $w1\overbrace{0\cdots0}^{t_1}\overbrace{0\cdots01}^{t_2}w'$ where $w10\cdots01w'$ is the distinguished node of the primary piece of $\langle w10\cdots01w' \rangle$ and either t_1 or t_2 is larger than or equal to the length of the longest block of zeros in $w11w'$. Otherwise, $w1\overbrace{0\cdots01}^{t_1}\overbrace{0\cdots01}^{t_2}w'$ would (by definition) be the distinguished node of $\langle w1\overbrace{0\cdots01}^{t_1}\overbrace{0\cdots01}^{t_2}w' \rangle$ and thus $w1\overbrace{0\cdots01}^{t_1}\overbrace{0\cdots01}^{t_2}w'$ would be on the same row as $w1\overbrace{0\cdots00}^{t_1}\overbrace{0\cdots01}^{t_2}w'$, a contradiction. Each necklace contains at most $2r$ such primary nodes where r is the difference between the lengths of the longest and second longest block of zeros in any string of the necklace. By Lemma 2-3, we can conclude that there are at most $O(N/logN)$ such primary nodes in the entire shuffle-exchange graph. Thus, at most $O(N/logN)$ exchange edges link nodes which are in different rows.

(b) Exchange Edges Which Link Nodes in the Same Row

Using the analysis developed in section 3.2.2b, it is not difficult to show that at most $O(N/logN)$ horizontal tracks are needed to insert the exchange edges which link two nodes that are in the same row. In particular, there are still only $O(N/logN)$ odd nodes in the top $logk$ rows of the grid and thus at most $O(N/logN)$ exchange edges are contained in the top $logk$ rows. These can be trivially inserted with the addition of just $O(N/logN)$ horizontal tracks.

Again following the methods of section 3.2.2b, it is not difficult to show that two exchange edges overlap on the ith row only if the first i bits of the associated nodes are identical. Thus at most 2^{k-i} tracks are needed to insert all of the exchange edges in the ith row for all $i>logk$. Summing, we can again conclude that at most $O(N/logN)$ additional horizontal tracks are needed to insert the remaining exchange edges.

3.3.4 Comments

The methods developed in this chapter can be used to find several other optimal layouts for the shuffle-exchange graph. The key variant is the method by which a node is distinguished. In particular, this method must be impervious to small alterations in the necklace. (This is so that most exchange edges will link nodes which are in the same row of the grid.) Only by changing the value of a bit in a small segment of the necklace (such as in the primary or secondary block of zeros) should we be able to globally change the distinguished node.

Another method of distinguishing a node is to select that node in the necklace which has the minimal value. Although the proof is very difficult, it can be shown that the layout for the N-node shuffle-exchange graph constructed in this manner has at most $O(N^2/log^2N)$ area. In the following section we will desribe additional methods of distinguishing nodes.

At this point, we should also note that the layout just described is *not known* to have optimal maximum edge length. In Part II of the thesis, we show that every layout of the N-node shuffle-exchange graph must have some edge of length at least $\Omega(N/log^2N)$. All the layouts we have considered thus far contain wires of length $\Theta(N/logN)$.

3.4 Layouts With Additional Edges

For some applications (such as the calculation of the discrete Fourier transform), it is useful to consider networks which have more than just shuffle and exchange edges. In particular, we will be interested in layouts for the shuffle-exchange graph which also include shift, reverse and transpose edges. In what follows, we will show how to modify the optimal layout for the shuffle-exchange graph so that these additional edges can be inserted without increasing the total area by more than a constant factor.

3.4.1 Shift Edges

Shift edges link the *ith* node to the *(i+1)st* node for all *odd i*. When combined with the exchange edges, the resulting network will have links between the *ith* and the *(i+1)st* nodes for all *i*. The inclusion of such edges facilitates the computation of discrete Fourier transforms at sequential intervals of a continuous signal. In

such applications, the input data contained in the *ith* processor is shifted to the $(i+1)st$ processor for each i after each computation of a discrete Fourier transform. The graph consisting of shuffle, exchange and shift edges is known as the *shuffle-shift graph.*

Using the methods developed in section 3.3, it is not difficult to show that the N-node shuffle-exchange graph can be laid out using only $O(N^2/log^2N)$ area. As before, the necklaces are broken into two or three pieces and placed in a grid according to the value of the associated distinguished node. Thus the shuffle edges can be inserted as before using only $O(N/logN)$ vertical and horizontal tracks.

For most odd nodes, adding a *1* to the value of the node changes only a relatively small number of bits at the end of the string. Thus it can be shown that at most $O(N/logN)$ shift edges link nodes which are in different rows. These can be easily inserted using only $O(N/logN)$ vertical and horizontal tracks. Of those edges which link nodes in the same row, at most $O(N/logN)$ are contained in the first *logk* rows. For $i>logk$, at most 2^{k-i} shift edges overlap at any point of the *ith* row. By introducing an extra vertical track for each necklace piece, it is possible to separate the layout of the shift edges on each level from that of the exchange edges. Thus both can be inserted simultaneously in the *ith* row using only $O(2^{k-i})$ total horizontal tracks. By the arguments of section 3.3, this means that at most $O(N/logN)$ additional horizontal tracks are needed to embed all of the remaining shift and exchange edges, thus completing the argument.

3.4.2 Reverse Edges

Reverse edges link pairs of nodes that are associated with binary strings which are reverses of each other. For example, $a_{k-1}\cdots a_0$ is linked to $a_0\cdots a_{k-1}$ via a reverse edge. Since the algorithm which computes discrete Fourier transforms on the shuffle-exchange network leaves the output for node $a_{k-1}\cdots a_0$ in node $a_0\cdots a_{k-1}$, reverse edges provide a fast and convenient way of straightening out the solution. The graph consisting of shuffle, exchange, shift and reverse edges will be referrred to as the *shuffle-shift-reverse graph.*

Using the techniques developed in section 3.3, it is also possible to show that the N-node shuffle-shift-reverse graph can be laid out in $O(N^2/log^2N)$ area. The basic idea is to modify the layout described in section 3.4.1 so that

1) pieces of necklaces which are reverses of each other are paired together in the left-to-right ordering, and

2) pieces of necklaces are folded in half.

The first constraint insures that the maximal overlaps of the reverse edges in each row will be small while the second constraint insures that most reverse edges link nodes which are in the same row. Although it is not immediately obvious, it can be checked that these modifications do not substantially change the procedure for inserting the shuffle, shift and exchange edges which was described in section 3.4.1. Thus all of the edges can be inserted using at most $O(N/logN)$ vertical and horizontal tracks.

3.4.3 Transpose Edges

Transpose edges link the *ith* node to the *(N-1-i)th* node for each *i*. Viewed in terms of binary strings, transpose edges link each node to its complement. Although we do not know of any specific applications of transpose edges, they would be useful for problems that require frequent transposition of the data.

By further modifying the optimal layout for the shuffle-shift-reverse graph, it is possible to add transpose edges without increasing the total area by more than a constant factor. In particular, the layout should be modified so that

1) pieces of necklaces which are complements of each other are paired together in the left-to-right ordering, and

2) the distinguished node is selected on the basis of the location of the longest block of consecutive identical bits (be they zeros *or ones*).

The first constraint insures that the maximal overlaps of the transpose edges in each row are small while the second constraint insures that most transpose edges link nodes which are on the same row. Although we do not present the details here, it is possible to show that such a layout can be constructed using only $O(N^2/log^2N)$ area, the least possible.

Appendix: Proofs of Lemmas 3-1 Through 3-4

We now present the proofs of Lemmas 3-1 through 3-4. Such results can also be found in the recent work of Guibas and Odlyzko [GO81a,GO81b]. We are deeply indebted to Kleitman for suggesting the proof of Theorem 3-1.

In what follows, we will write $\overline{\Psi}_k(t)$ to denote the number of k-bit strings which do *not* contain t-1 consecutive zeros. Except for the string of all zeros (which we ignore), these are precisely the strings which do not contain the substring $v_t = \overbrace{10\cdots0}^{t}$. The proofs of Lemmas 3-1 through 3-4 depend heavily on the following combinatorial result.

Theorem 3-1: *For large t and k,*

$$\overline{\Psi}_k(t) = 2^k e^{-k2^{-t}} e^{O(t2^{-t}, kt2^{-2t})}.$$

Proof: We first count the number $\overline{\Psi}_k'(t)$ of k-bit strings which do not contain an occurrence of v_t between the beginning and end of the string (i.e., for the time being we ignore the occurrences of v_t which begin at the end and end at the beginning of a string).

Fix t and let f_i denote the number of i-bit strings ending with v_t but which do not contain any other occurrences of v_t in the string. Set $F(x) = \sum_{i=0}^{\infty} f_i x^i$. Note that $\overline{\Psi}_k'(t)$ is the $(k+t)th$ coefficient of $F(x)$. Let $f_i^{(j)}$ denote the number of i-bit strings ending in v_t which contain precisely j occurrences of v_t and set

$$F^{(j)}(x) = \sum_{i=0}^{\infty} f_i^{(j)} x^i .$$

Since occurrences of v_t cannot overlap, it is not difficult to show that $F^{(j)}(x)$ is identical to $F(x)^j$ for all $j > 1$.

Let g_i be the number of i-bit strings which end in v_t (regardless of the number of other occurrences of v_t which appear in the string) and set $G(x) = \sum_{i=0}^{\infty} g_i x^i$. Since $g_i = 2^{i-t}$ for all $i \geq t$, it is easily seen that $G(x) = x^t/(1-2x)$. Also note that

$$G(x) = \sum_{j=1}^{\infty} F^{(j)}(x)$$

$$= \sum_{j=1}^{\infty} F(x)^j$$

$$= [1/(1 - F(x))] - 1$$

and thus that

$$F(x) = G(x)/(G(x) + 1)$$

$$= x^t/(1 - 2x + x^t) .$$

Thus $\overline{\Psi}_k'(t)$ is simply the kth coefficient of $1/(1 - 2x+x^t)$. For example, $\overline{\Psi}_4'(2)=5$ which is the coefficient of x^4 in the expansion of $1/(1 - 2x+x^2)$.

Let $p(x) = 1 - 2x+x^t$. It is easily observed that $gcd(p(x), dp(x)/dx) = 1$ and thus that $p(x)$ does not have any multiple roots for $t > 2$. Thus we can expand

$$p(x)^{-1} = \sum_{i=1}^{t} A_i/(x-r_i)$$

where $\{r_i \mid 1 \le i \le t\}$ is the set of distinct (and possibly complex) roots of $p(x)$ and

$$A_i = [(x-r_i)/p(x)]_{r_i}$$

$$= 1/[dp(x)/dx]_{r_i}$$

for $1 \le i \le t$. Once the roots of $p(x)$ are known, we can calculate $\overline{\Psi}_k'(t)$ from the formula

$$\overline{\Psi}_k'(t) = -\sum_{i=1}^{t} A_i r_i^{-(k+1)} .$$

Although we do not know how to find the roots of $p(x)$ explicitly for large t, we can describe them asymptotically. First observe that as $t\to\infty$, the absolute value of every root must approach either $1/2$ or 1. Otherwise the absolute value of one term of $p(x)$ will dominate the sum of the absolute values of the other two terms. For example, if $|r| \le c < 1/2$ as $t\to\infty$ for some root r and constant c, then $1 > |2r|+|r^t|$ for large t.

If there are to be any roots r such that $|r|\to 1/2$, it is essential that $r\to 1/2$. Otherwise, the real part of $p(r)$ cannot vanish for large t. By substituting $(1/2)e^{s(t)}$ for r where $s(t)\to 0$ as $t\to\infty$, we find that

$$1 - e^{s(t)} + 2^{-t} e^{ts(t)} = 0$$

and thus that

$$1 - (1 + s(t) + O(s(t)^2)) + 2^{-t}(1 + O(ts(t))) = 0$$

Thus $s(t) = 2^{-t} + q(t)$ where $|q(t)| << 2^{-t}$ as $t \to \infty$. Another iteration of this process reveals that $q(t) = O(t2^{-2t})$ and thus that

$$r = (1/2)\, e^{2^{-t}} e^{O(t2^{-2t})} \quad \text{as } t \to \infty .$$

In fact, there is precisely one root, say r_1, which approaches $1/2$ as $t \to \infty$. The absolute values of the remaining roots approach 1. In particular, the absolute values of these roots must be greater than or equal to 1 for large t. Otherwise there would be a root r and a function $\varepsilon(t) \to 0^+$ such that $|r| = 1 - \varepsilon(t)$. But then

$$\begin{aligned} |2r| &= 2 - 2\varepsilon(t) \\ &> 1 + |1 - \varepsilon(t)|^t \\ &= 1 + |r^t| \end{aligned}$$

for $t > 2$ and it would be impossible for $p(r)$ to vanish for large t, a contradiction.

It remains to compute the A_i. Since $dp(x)/dx = tx^{t-1} - 2$, we find that $A_1 = -(1/2) + O(t2^{-t})$ and that $A_i = O(1/t)$ for $2 \le i \le t$. Thus

$$\overline{\Psi}_k'(t) = O(1) - [-1/2 + O(t2^{-t})]\, 2^{k+1} e^{-(k+1)2^{-t}} e^{O(kt2^{-2t})} .$$

Replacing $1 + O(t2^{-t})$ with $e^{O(t2^{-t})}$ and simplifying, we conclude that

$$\overline{\Psi}_k'(t) = 2^k e^{-k2^{-t}} e^{O(t2^{-t},\, kt2^{-2t})}$$

for large t and k.

The only strings which are included in the count of $\overline{\Psi}_k'(t)$ but not in that of $\overline{\Psi}_k(t)$ are those of the form $\overbrace{0 \ldots 0}^{i} w \overbrace{10 \ldots 0}^{t-i}$ where $1 \le i \le t-1$ and w is a string which is included in the count of $\overline{\Psi}_{k-t}'(t)$. Thus

$$\overline{\Psi}_k(t) = \overline{\Psi}_k'(t) - (t-1)\overline{\Psi}_{k-t}'(t)$$

$$= 2^k e^{-k2^{-t}} e^{O(t2^{-t}, kt2^{-2t})} - (t-1) 2^{k-t} e^{-(k-t)2^{-t}} e^{O(t2^{-t}, kt2^{-2t})}$$

$$= 2^k e^{-k2^{-t}} e^{O(t2^{-t}, kt2^{-2t})}$$

for large t and k. This completes the proof of the theorem □

We can now prove Lemmas 3-1 and 3-2.

Proof of Lemma 3-1: From the definition, we know that

$$\Psi_k(t) = \overline{\Psi}_k(t+2) - \overline{\Psi}_k(t+1)$$

$$= 2^k e^{-k2^{-(t+2)}} e^{O(t2^{-t}, kt2^{-2t})} - 2^k e^{-k2^{-(t+1)}} e^{O(t2^{-t}, kt2^{-2t})}$$

for large t and k. For $t \geq (logk)/2 + loglogk$, both $t2^{-t}$ and $kt2^{-2t}$ vanish as $k \to \infty$. In what follows, we will show that if $t << k$, then

$$e^{-k2^{-(t+2)}} - e^{-k2^{-(t+1)}} >> O(t2^{-t}, kt2^{-2t})$$

and thus that

$$\Psi_k(t) \sim 2^k (e^{-k2^{-(t+2)}} - e^{-k2^{-(t+1)}}) .$$

Assume for the purposes of contradiction that

$$e^{-k2^{-(t+2)}} - e^{-k2^{-(t+1)}} \leq O(t2^{-t}, kt2^{-2t}) .$$

Then, $e^{-k2^{-(t+2)}} \sim e^{-k2^{-(t+1)}}$ which means that $e^{-k2^{-(t+2)} + k2^{-(t+1)}} \sim 1$ and thus that $k2^{-(t+2)} \to 0$. Thus we can use a Taylor series expansion of the exponentials to find that

$$e^{-k2^{-(t+2)}} - e^{-k2^{-(t+1)}} \sim (1 - k2^{-(t+2)}) - (1 - k2^{-(t+1)})$$

$$= k2^{-(t+2)}$$

$$>> O(t2^{-t}, kt2^{-2t})$$

provided that $t << k$, a contradiction □

Proof of Lemma 3-2: The number of k-bit strings which do not contain a block of $logk - loglnk - 1$ consecutive zeros is

$$\overline{\Psi}_k(logk - loglnk) \sim 2^k e^{-k2^{-logk+loglnk}}$$

$$= 2^k/k$$

$$= O(N/logN) .$$

The number of k-bit strings which contain a block of $2logk+1$ consecutive zeros is

$$2^k - \overline{\Psi}_k(2logk+2) \sim 2^k - 2^k e^{-k2^{-2logk-2}} e^{O((logk)/k^2)}$$

$$= 2^k - 2^k[1 - 1/(4k) + O((logk)/k^2)]$$

$$\sim 2^k/4k$$

$$= O(N/logN) \square$$

The proofs of Lemmas 3-3 and 3-4 depend on the following corollary to Theorem 3-1.

Corollary 3-1: *For bounded m and p and large k and t,*

$$\sum_{t=1}^{\frac{k+c}{m}} \overline{\Psi}_{k-mt+p}(t) = O(2^k/k^m) .$$

Proof: We first observe that for $t < 2logk/3$,

$$\overline{\Psi}_{k-mt+p}(t) \leq \overline{\Psi}_k(2logk/3)$$

$$\sim 2^k e^{-k2^{-(2logk)/3}}$$

$$= 2^k e^{-k^{1/3}}$$

and thus that

$$\sum_{t=1}^{\frac{2logk}{3}} \overline{\Psi}_{k-mt+p}(t) \leq (2/3)\, logk\, 2^k e^{-k^{1/3}}$$

$$\ll 2^k/k^m$$

for any finite m and p as $k \to \infty$.

For larger values of t,

$$\overline{\Psi}_{k-mt+p}(t) \sim 2^{k-mt+p} e^{-k2^{-t}}$$

and thus

$$\sum_{t=\frac{2\log k}{3}}^{\frac{k+p}{m}} \overline{\Psi}_{k-mt+p}(t) \sim \sum_{t=\frac{2\log k}{3}}^{\frac{k+p}{m}} 2^{k-mt+p} e^{-k2^{-t}} .$$

By making the change of variables $r = t - \log k$, we can see that the preceding sum is at most

$$(2^{k+p}/k^m) \sum_{r=-\infty}^{\infty} 2^{-mr} e^{-2^{-r}}$$

and thus at most $O(2^k/k^m) = O(N/\log N)$ □

Proof of Lemma 3-3: A string whose longest block of zeros has length t and whose second longest block of zeros has length $s \leq t$ is of the form $w\overbrace{10\cdots0}^{t+1}w'$, where the longest block of zeros in ww' has length s. By definition, there are at most $k\Psi_{k-t-1}(s)$ such strings. Thus the sum over all *necklaces* of the difference between the sizes of the longest block and second longest block of zeros is at most

$$\begin{aligned}
&\leq (1/k) \sum_{t=0}^{k} \sum_{s=0}^{t} (t-s)\, k\, \Psi_{k-t-1}(s) \\
&= \sum_{t=0}^{k} \sum_{s=0}^{t} (t-s) \left[\overline{\Psi}_{k-t-1}(s+2) - \overline{\Psi}_{k-t-1}(s+1)\right] \\
&= \sum_{s=1}^{k} \sum_{t=s}^{k} \overline{\Psi}_{k-t}(s) \\
&= \sum_{s=1}^{k} \left(2^k e^{-k2^{-s}} e^{O(s2^{-s},\, ks2^{-2s})} \sum_{t=s}^{k} 2^{-t} e^{t2^{-s}} \right) \\
&\leq \sum_{s=1}^{k} \left(2^k e^{-k2^{-s}} e^{O(s2^{-s},\, ks2^{-2s})}\ 2^{-s} e^{O(s2^{-s})} \right) \\
&= \sum_{s=1}^{k} 2^{k-s} e^{-k2^{-s}} e^{O(s2^{-s},\, ks2^{-2s})} \\
&\leq \sum_{s=1}^{k} \overline{\Psi}_{k-s}(s) \\
&= O(N/\log N)
\end{aligned}$$

by Corollary 3-1 □

Proof of Lemma 3-4: Consider a necklace which fails to have a uniquely distinguished node. Each node in such a necklace must have one of the following three forms:

1) $w_1\overbrace{0\cdots0}^{t}w_2\overbrace{0\cdots0}^{t}w_3$,

2) $w_1\overbrace{0\cdots0}^{t}w_2\overbrace{0\cdots0}^{t}w_3\overbrace{0\cdots0}^{t}w_4$ or

3) $w_1\overbrace{0\cdots0}^{t}w_2\overbrace{0\cdots0}^{t}w_3\overbrace{0\cdots0}^{t}w_4\overbrace{0\cdots0}^{t}w_5$

where t is the length of the longest block of zeros in any of the strings. It is easily seen that there are at most

1) $k\sum_{t=1}^{k/2}\overline{\Psi}_{k-2t}(t+2)$ nodes of the first type,

2) $k^2\sum_{t=1}^{k/3}\overline{\Psi}_{k-3t}(t+2)$ nodes of the second type and

3) $k^3\sum_{t=1}^{k/4}\overline{\Psi}_{k-4t}(t+2)$ nodes of the third type.

By Corollary 3-1, we can thus conclude that there are at most $O(N/\log N)$ such nodes altogether □

CHAPTER 4

PRACTICAL LAYOUTS

Although the $O(N^2/\log^2 N)$-area layout for the shuffle-exchange graph described in Chapter 3 is (up to a constant) *asymptotically* optimal, it is not optimal for small values of N (e.g., $N=128$). In fact, none of the general layout procedures thus far discussed provide good layouts for small shuffle-exchange graphs. For practical applications, however, these are precisely the shuffle-exchange graphs for which we need good layouts.

In this chapter, we descibe techniques for finding good layouts for small shuffle-exchange graphs. Although the techniques (which are described in section 4.2) do not yet constitute a general procedure for finding truly optimal layouts for all shuffle-exchange graphs, they can be used to find "very nice" layouts for "small" shuffle-exchange graphs. As examples, we have included layouts for the *8*-node, *16*-node, *32*-node, *64*-node and *128*-node shuffle-exchange graphs in section 4.3. The layouts are "very nice" in the sense that:

1) they require much less area than previously discovered layouts,

2) they have a certain natural structure which facilitates efficient layout description, chip manufacture and I/O management, and

3) they require the minimal amount of area for layouts with such structure.

4.1 Preliminaries

We have chosen to use the Thompson grid model [T80] to illustrate our techniques because of its widespread acceptance and its simplicity. For practical layouts, however, the assumption that processors can be represented by points is clearly false. Nontheless, we show in section 4.1.1 that good Thompson model layouts can still be used to find good practical layouts. Thus we will be able to rest assured that the Thompson model is, in fact, an acceptable means for describing practical layouts of the shuffle-exchange graph.

We must also be sure that the *layouts* we design can be effectively used in practice. For example, it is important that the layouts have a suitable input/output structure so that data can be put on and taken off the chip efficiently. In section 4.1.2, we describe a general class of layouts for the shuffle-exchange graph which appear to satisfy such constraints. The remainder of the chapter will then be devoted to finding optimal layouts within this class.

4.1.1 A Closer Look at the Thompson Model

The manner in which the Thompson model is useful for describing practical layouts varies with the size of the processors involved. For example, if one desires to use the shuffle-exchange graph as a permuter, then each processor need only contain k storage registers and some I/O hardware. Such a processor can be easily hardwired in a kxk square. In order to achieve maximum parallelism, each wire of the Thompson model layout is reproduced k times so that an entire k-bit word can be transmitted in one time step. For example, the optimal *2x6* Thompson model layout for the *8*-node shuffle-exchange graph (which is shown in Figure 4-3 in section 4.3) can be transformed into the more realistic *6x18* layout shown in Figure 4-1 by tripling the grid lines and replacing the point processors by *3x3* boxes (into which the guts of each processor can later be wired).

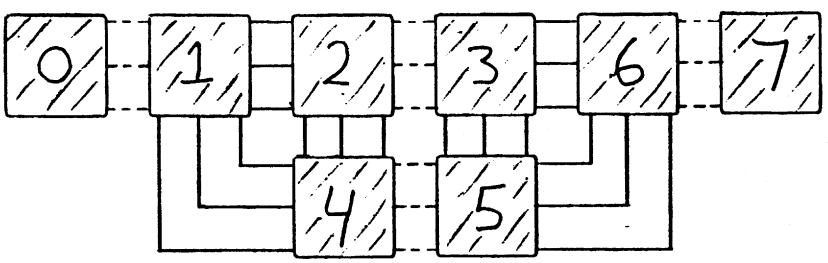

Figure 4-1: *A transformed Thompson model layout for the 8-node shuffle-exchange graph.*

For some applications, the processors themselves require an entire chip. For example, every processor of a shuffle-exchange graph used to compute discrete Fourier transforms must be equipped with a floating point multiplier. Using the best technology currently available, only a few floating point multipliers can be wired onto a single chip. In this case, a Thompson model layout can be used to

design an efficient *layout of chips* where each chip contains a single processor. (Such a device is currently under development at IBM.) The wires, as before, are replicated to achieve maximum parallelism but now serve as links between chips. Since the wires must be much wider in such a device, the side length of a processor (the chip) is about the same as the combined width of all the wires (pins) attached to it. By following an expansion procedure similar to the one described in the previous example, a good Thompson model layout can thus be used to design a good practical layout.

4.1.2 A Class of Practical Layouts

In this chapter, we will consider layouts for the shuffle-exchange graph for which:

1) each necklace appears as a rectangle consisiting of arbitrarily long segments of two vertical tracks and unit length segments of two horizontal tracks,

2) the horizontal tracks are divided into pairs, each pair containing at most one full necklace and any number of degenerate necklaces, and

3) each exchange edge appears as a horizontal line segment.

For example, the layouts described in Chapter 2 have this form.

Such layouts are particularly well suited for practical implementation since their structure facilitates efficient description, chip manufacture and data management. For example, by attaching a pin to each of the $\Theta(N/logN)$ necklaces (this *is* feasible for small N), it is possible to load N input values into an N-processor shuffle-exchange chip in just $O(logN)$ steps.

Even more importantly, we will show in the following section how to find layouts with the above form which require very small amounts of area. Thus very little is lost by restricting our attention to such layouts.

4.2 Optimization Techniques

In this section, we explain how to find layouts for small shuffle-exchange graphs which are optimal up to the constraints described in section 4.1.2. For the most part, our methods are comprised of common sense, heuristics and exhaustive searches.

4.2.1 Ordering the Necklaces

The first step in finding optimal layouts of the form described in section 4.1.2 is to order the necklaces from left to right so that the number of exchange edges which overlap at each point of the ordering is kept small. More precisely, we wish to find an ordering of the necklaces for which the maximum number of exchange edges overlapping at any point is minimized. For example, no more than *6* exchange edges overlap at any point of the ordering used to produce the layout for the *32*-node shuffle-exchange graph shown in Figure 4-2. If we switched the necklace ⟨*5*⟩ with ⟨*11*⟩, however, *9* exchange edges would overlap in the gap between ⟨*7*⟩ and ⟨*5*⟩. Since the maximum overlap is a lower bound on the number of horizontal tracks necessary to insert the exchange edges, we can easily see that the latter ordering is inferior since any layout it produces must have at least *9* horizontal tracks. Note that the layout in Figure 4-2 has just *6* horizontal tracks.

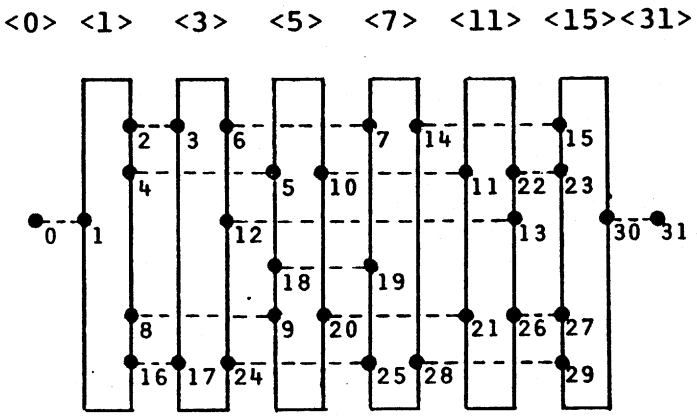

Figure 4-2: *A good ordering of the necklaces for the 32-node shuffle-exchange graph.*

As we mentioned in Chapter 3, it is not known how best to order the necklaces in general. For small shuffle-exchange graphs, however, there are several simple heuristics which produce optimal orderings. For example, arrangements of the necklaces from left to right in order of nondecreasing size or, alternatively, in order of increasing minimal number represented are usually quite close to optimal for small shuffle-exchange graphs. In fact, such orderings are within a necklace swap of optimal for $N \leq 256$ ($k \leq 8$). Note the the ordering displayed in Figure 4-2 could

have been produced by either of these methods.

Probably the most difficult task is proving that a good ordering is, in fact, optimal. The techniques we have used to prove optimality depend heavily on exhaustive searches. For $k \leq 8$, the techniques have suceeded in proving the optimality of good orderings. For $9 \leq k \leq 13$, we have found good orderings but have been unable to prove that they are optimal. We have summarized the results in Table 4-1. Note that for each k, the maximum overlap of the best known ordering serves only as a *lower* bound for the number of horizontal tracks that will be required for any layout with that ordering. In some cases, additional horizontal tracks may be required.

Table 4-1

Maximum Overlap of Best Known Orderings

k	N	maximum overlap of best known ordering	optimal?
3	8	2	yes
4	16	3	yes
5	32	6	yes
6	64	10	yes
7	128	18	yes
8	256	33	yes
9	512	62	?
10	1024	115	?
11	2048	214	?
12	4096	388	?
13	8192	754	?

4.2.2 Inserting the Exchange Edges

The second step in constructing optimal layouts for small shuffle-exchange graphs is to insert the exchange edges using as few horizontal tracks as possible.

Recall that in Chapter 2, we showed how to use the complex plane diagram as one method of inserting the exchange edges. Although this method is theoretically nice, it is not very practical since it uses an excessive number of horizontal tracks to insert the exchange edges. For example, *10* horizontal tracks were used to insert the exchange edges in the layout shown in Figure 2-3 whereas only *6* tracks were required in the layout shown in Figure 4-2 (even though the same necklace orderings were used for both layouts).

The complex plane diagram can still be of use when inserting exchange edges, however. For example, notice that the top-to-bottom orderings of the exchange edges across most of the vertical cuts which are located between necklaces in the layout in Figure 4-2 are the same as the orderings for the corresponding cuts in Figure 2-3. In general, knowledge of the level structure of the complex plane diagram is very helpful in optimizing the insertion of the exchange edges. In fact, we relied heavily on such knowledge when constructing the optimal layouts displayed in section 4.3.

For very small shuffle-exchange graphs (e.g., for $k \leq 5$), it is possible to find optimal embeddings of the exchange edges by trying all reasonable possibilities. For somewhat larger shuffle-exchange graphs (e.g., $k=6,7$), however, the task is substantially more difficult. In order to find the optimal layouts shown in section 4.3, we

1) first located the center of the region of maximum overlap and (using the complex plane diagram as a guide) inserted the exchange edges which crossed the region (one edge on each horizontal track),

2) next inserted the exchange edges located in neighboring regions without (if possible) introducing any additional tracks, and

3) lastly inserted the remaining exchange edges (again without adding any new horizontal tracks).

Steps 1 and 3 are easy but step 2 can be difficult. In some cases it is necessary to interchange the left and right parts of some necklaces or to slide a node around from one part of a necklace to the other. For $k = 6$ and 7, it is also necessary to introduce an extra horizontal track at step 2. For larger shuffle-exchange graphs, it would probably be necessary to introduce even larger numbers of horizontal tracks.

4.2.3 Additional Savings

All of the practical layouts we have considered thus far have two horizontal tracks which are used solely for the purpose of connecting the left part of each necklace to the right part. It is not difficult to show that these tracks can be eliminated without affecting the rest of the layout. As an example of how this can be accomplished, we suggest that the reader compare the layout of the *32*-node shuffle-exchange graph shown in Figure 4-2 with that in Figure 4-5.

Even larger savings can be had for some shuffle-exchange graphs by doubling up the degenerate necklaces with full necklaces in the same pair of vertical tracks, thus reducing the number of vertical tracks used. Of course, it is necessary to rearrange the exchange edges somewhat but, as degenerate necklaces have very few nodes in small shuffle-exchange graphs, this can usually be done without introducing any additional horizontal tracks. For example, substantial savings can be achieved in this manner for the *16*-node and *64*-node shuffle-exchange graphs.

4.3 Optimal Layouts

In the following figures, we exhibit layouts for the *8*-node, *16*-node, *32*-node, *64*-node and *128*-node shuffle-exchange graphs which are optimal up to the constraints described in section 4.1.2. The layouts were found via the techniques described in section 4.2.

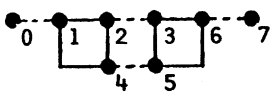

Figure 4-3: *A 2x6 layout for the 8-node shuffle-exchange graph.*

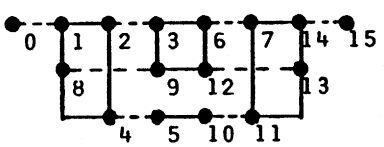

Figure 4-4: *A 3x8 layout for the 16-node shuffle-exchange graph.*

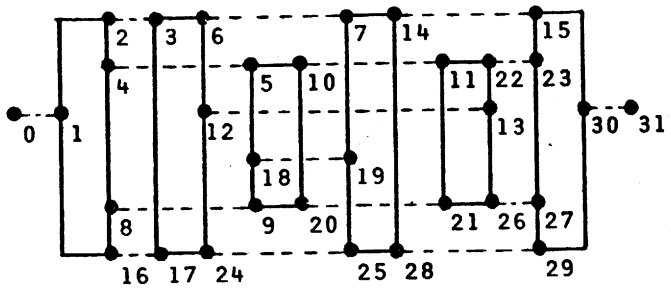

Figure 4-5: *A 6x14 layout for the 32-node shuffle-exchange graph.*

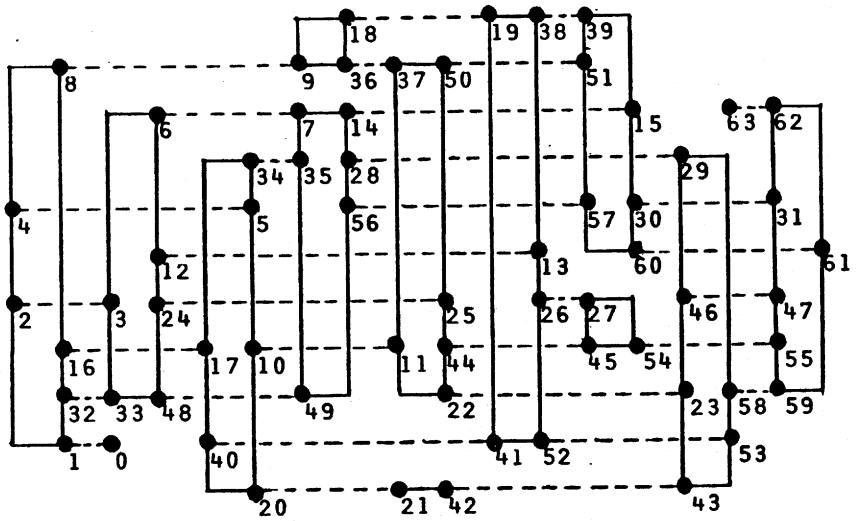

Figure 4-6: *An 11x18 layout for the 64-node shuffle-exchange graph.*

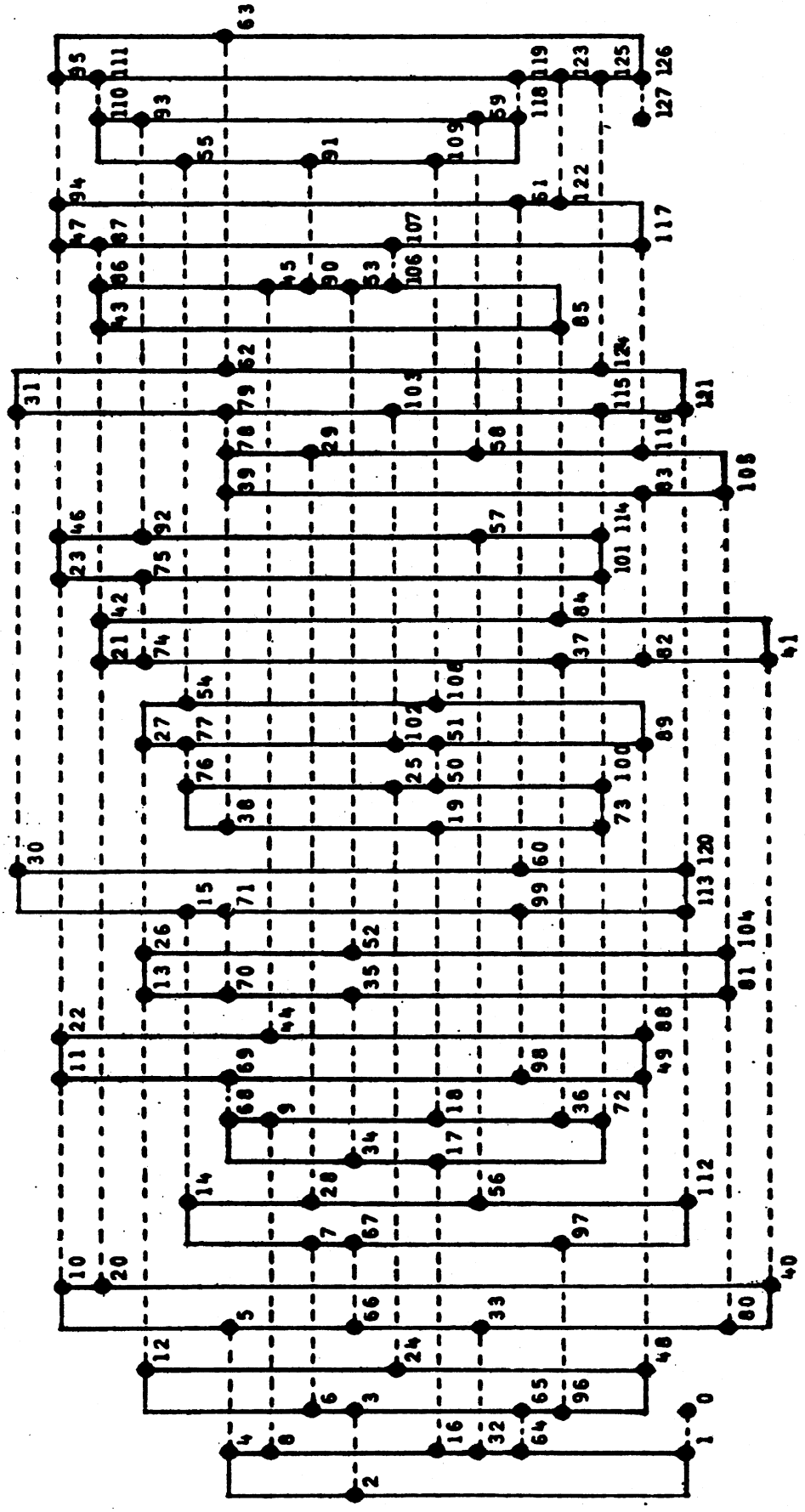

Figure 4-7: *A 19x36 layout for the 128-node shuffle-exchange graph.*

4.4 Other Layouts

To this point, we have considered only a specific class of layouts for the shuffle-exchange graph. As these layouts are quite good, it is not clear that we need to consider others. Nevertheless, it is worth pointing out that slightly better layouts do exist for some shuffle-exchange graphs. For example, by considering layouts in which the exchange edges are allowed to bend and in which two or more full necklaces can occupy the same pair of vertical tracks, it is possible to construct the layout for the *32*-node shuffle-exchange graph shown in Figure 4-8.

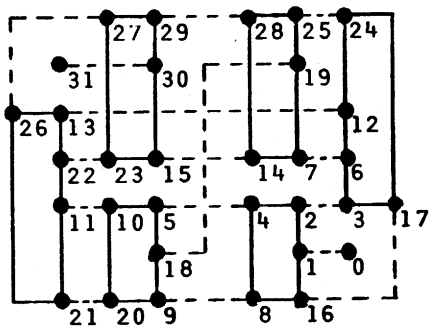

Figure 4-8: *An improved 7x9 layout for the 32-node shuffle-exchange graph.*

It is likely that slight improvements can also be made for larger shuffle-exchange graphs. At this point, however, we feel that research efforts should be directed more towards implementation of the good layouts already discovered. Once this is done, it will be much clearer whether or not the effort necessary to further reduce the layout area is justified.

This empty page was substituted for a blank page in the original document.

PART II

LOWER BOUND TECHNIQUES FOR VLSI

CHAPTER 5

REVIEW OF KNOWN TECHNIQUES

In this chapter, we review the known techniques for determining the layout area and maximum edge length of an arbitrary VLSI network. We also preview the results we will prove in Chapters 6 through 8 of the thesis. A comparison of our lower bounds with the previously known upper and lower bounds can be found in Tables 5-2 and 5-4.

5.1 Area Bounds

One of the most important problems in the theory of VLSI is the determination of the minimum amount of area required to lay out a network on a chip. Given an arbitrary graph, this problem has two parts; namely,

1) finding a good layout for the graph, and

2) showing that the layout is optimal.

There are a variety of techniques known for finding good layouts for specific graphs [MR79, PV79, S79, HL80, MC80, PV80, SR80b, T80, BL81, KLLM81, LLM81, LM81, PRS81, T81], but the only known general technique is due to Leiserson [L80a,L80b]. In particular, he showed how to construct a good layout for any graph for which a good separator is known. (An N-node graph is said to have an *$f(N)$-separator* if it can be partitioned into two equal-sized subgraphs G_1 and G_2 such that at most $f(N)$ edges link G_1 to G_2 and both G_1 and G_2 have $f(N/2)$-separators.) We have summarized Leiserson's results in Table 5-1.

There are two difficulties with Leiserson's method. First, it is not always possible to find a good separator for a graph. For instance, a minimal $O(N/\log N)$-separator was not found for the shuffle-exchange graph until after an optimal $O(N^2/\log^2 N)$-area layout was discovered. Secondly, the layouts produced by Leiserson's technique are not always optimal – even if a minimal separator is known. For example, Leiserson's technique requires $\Theta(N\log^2 N)$ area to lay out the N-node mesh, substantially more than is really needed. For the most part

Table 5-1

Upper Bounds on the Layout Area of N-Node Graphs With Specified Separators

separator	upper bound on layout area
N^α, $\alpha < 1/2$	N
N^α, $\alpha = 1/2$	$N\log^2 N$
N^α, $\alpha > 1/2$	$N^{2\alpha}$

however, Leiserson's method is a good one and certainly the most general technique currently available.

Once a good layout for a network has been found, it remains to show that the layout is optimal. This is accomplished by proving a good *lower* bound on the layout area of the network. The only known methods for proving such lower bounds are due to Thompson [T79,T80], Vuillemin [V80] and Lipton and Sedgewick [LS81]. They have concentrated on the related problem of proving lower bounds for the bisection width of a graph. (The *bisection width* of a graph is the minimum number of edges which must be removed in order to separate the graph into two disjoint and equal-sized subgraphs.)

Thompson was the first to notice the relationship between bisection width and layout area. In particular, he showed that the *wire* area of a graph with bisection width b is at least $\Omega(b^2)$. In what follows, we prove the slightly weaker (and simpler) result for layout area.

Theorem 5-1 (Thompson [T79]): *The layout area of a graph with bisection width b is at least $\Omega(b^2)$.*

Proof: Consider an optimal layout of a graph G with bisection width b. Cut the layout horizontally so that precisely $1/2$ of the nodes of G are above the cut. (For an example, see the diagram in Figure 5-1). Since at least b edges must cross the cut, the layout must contain at least $b-1$ vertical tracks. A similar argument reveals that the layout must also have at least $b-1$ horizontal tracks. Thus the area of the layout is at least $(b-1)^2 = \Omega(b^2)$ □

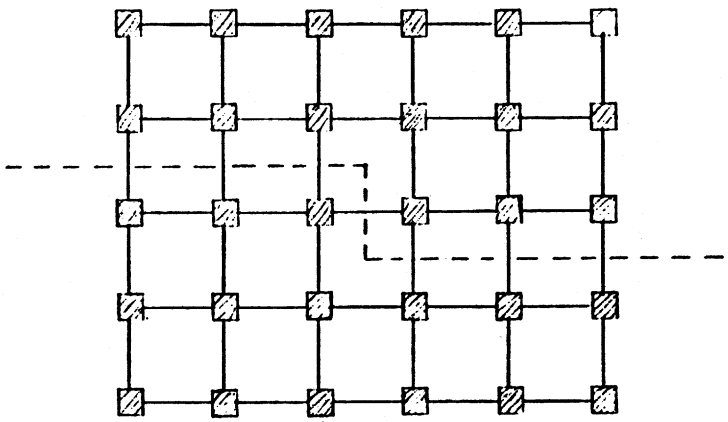

Figure 5-1: *A horizontal bisection of a layout.*

Although the task of finding a good lower bound on the bisection width of a graph is difficult in general, Thompson [T79] was succesful in finding good bisection width lower bounds for a variety of computationally useful networks. For example, he used information transfer arguments to show that any network which is capable of computing the discrete Fourier transform on N elements in T steps must have bisection width at least $\Omega(N/T)$. Among other things, he was thus able to conclude that at least $\Omega(N^2/\log^2 N)$ area is required to lay out the N-node shuffle-exchange graph.

Thompson's work has recently been extended; first by Vuillemin [V80] and then by Lipton and Sedgewick [LS81]. Vuillemin characterized a broad class of graphs for which Thompson's lower bound arguments can be applied while Lipton and Sedgewick showed how to use crossing sequence arguments to prove lower bounds for an even larger class of graphs.

Although the methods of Thompson, Vuillemin, Lipton and Sedgewick are quite elegant and useful in establishing good bisection width lower bounds for certain graphs, their applicability is inherently limited to graphs for which the layout area is no more than a constant times as large as the square of the bisection width. Thus they have not been of use in resolving two of the key open questions in VLSI theory; namely,

1) "How much area is needed to lay out a planar graph?" and

2) "How much area is needed to lay out a graph which has an $O(N^{1/2})$-separator?."

The planar graph question is particularly important since, as we will show in Chapter 7, the layout problem of an arbitrary graph can be reduced to that for a planar graph. No nontrivial lower bounds have been found for either problem, however. As we mentioned previously, the best procedure known requires $O(N\log^2 N)$ area to lay out an arbitrary N-node graph with an $O(N^{1/2})$-separator. As Lipton and Tarjan [LT77] have shown that every N-node planar graph has an $O(N^{1/2})$-separator, the $O(N\log^2 N)$-area layout procedure also works for planar graphs. Although it is suspected that better layout procedures exist for planar graphs, none have yet been found.

In the thesis, we pursue an entirely different strategy in developing new lower bound techniques for VLSI. Whereas previous researchers have been concerned primarily with the bisection width of a network, we shall be concerned with its *crossing number* and *wire area*. Both are lower bounds on the layout area of any graph. In fact, we will show in Chapter 7 that

$$\Omega(b^2) \leq c+N \leq w \leq A$$

for any N-node graph with bisection width b, crossing number c, wire area w and layout area A.

The preceding inequality implies that every lower bound technique for the bisection width of a graph is also a lower bound technique for its crossing number and wire area. Thus nothing is lost by forgetting about bisection width and concentrating ones efforts on finding good lower bounds for the crossing number and wire area of a graph. In fact, much can be gained. For example, we will use such techniques to find

1) an N-node planar graph which has layout area $\Theta(N\log N)$, and

2) an N-node (nonplanar) graph with an $O(N^{1/2})$-separator which has layout area $\Theta(N\log^2 N)$.

The first result demonstrates that not all planar graphs can be laid out in linear area, thus disproving a conjecture thought by many to be true. The second result indicates that Leiserson's $O(N\log^2 N)$-area layout technique for graphs with $O(N^{1/2})$-separators is optimal at least some of the time and thus cannot, in general, be improved.

For easy reference, we have summarized our results along with the previously

known upper and lower bounds in the following table. The upper bounds are due to Leiserson [L80a] and represent the maximal amount of area needed to lay out any graph with the designated property. The lower bounds, on the other hand, represent the minimal amount of area required to lay out a *specific* class of graphs with the designated property. The previously known lower bounds are, for the most part, trivial. The only exception is the $N^{2\alpha}$ bound which, as a corollary of Theorem 5-1, is due to Thompson [T79].

Table 5-2

Area Bounds

separator	previous lower bound	our lower bound	upper bound
N^{α}, $\alpha < 1/2$	N		N
N^{α}, $\alpha = 1/2$	N	$N\log^2 N$	$N\log^2 N$
N^{α}, $\alpha > 1/2$	$N^{2\alpha}$		$N^{2\alpha}$
(planar)	N	$N\log N$	$N\log^2 N$

5.2 Edge Length Bounds

There has been a great deal of interest lately in the problem of minimizing the length of the longest wire in VLSI layouts [BL81,CM81,PRS81]. It is not difficult to show that the length of the longest wire in any reasonable, area-optimal VLSI layout is at most a constant times the square root of the layout area. (Otherwise, some wire would be longer than the perimeter of the layout, which is unreasonable.) Bhatt and Leiserson [BL81] recently found better layouts for graphs with small separators. We have summarized their results in Table 5-3. (For completeness, we have also included the trivial bound for graphs with large separators.)

It is worth noting that the layouts which achieve the bounds in Table 5-3 simultaneously achieve the best known bounds for layout area. Thus no *layout area/maximum edge length* tradeoffs are apparent.

Table 5-3

Upper Bounds on the Maximum Edge Length of N-Node Graphs With Specified Separators

separator	upper bound on maximum edge length
N^{α}, $\alpha < 1/2$	$N^{1/2}/logN$
N^{α}, $\alpha = 1/2$	$N^{1/2}logN/loglogN$
N^{α}, $\alpha > 1/2$	N^{α}

Very little has been accomplished in the way of lower bounds, however, since bisection width arguments do not seem to be applicable to edge length considerations. In fact, the only known lower bound for maximum edge length is the trivial lower bound derived from the diameter of a graph. (The *diameter* of a graph is the greatest distance between any pair of nodes in the graph where *distance* is defined to be the length of the shortest path linking the pair of nodes.) The precise lower bound is stated in the following theorem.

Theorem 5-2: *Any layout of a graph G with diameter d and layout area A has some edge of length at least* $A^{1/2}/3d$.

Proof: Let Γ be any layout of G and q be the length of the longest wire in Γ. We will use Γ to construct another layout Γ' of G which has at most $9d^2q^2$ area. Since any layout for G has at least A area, this will be sufficient to show that $q \geq A^{1/2}/3d$.

Since every pair of nodes in G is linked by a path of length d or less, we can conclude that every pair of nodes are within distance dq of each other in Γ. (Otherwise, some edge would have length greater than q in Γ, a contradiction.) Thus, all of the nodes are contained in some $dq \times dq$ square in Γ. Since every wire which leaves the square must re-enter at some other point, we can conclude that at most $2dq$ wires can cross the boundary of the square at any point. By rewiring the portion of Γ which is outside the square, it is possible to produce a second layout Γ' for G which has at most $2dq$ additional horizontal tracks and $2dq$ additional vertical tracks. (One additional horizontal track and one additional

vertical track are needed to replace each wire.) Thus the total area of Γ' is at most $9d^2q^2$. (As an example of how the rewiring should be done, we have included Figure 5-2.) □

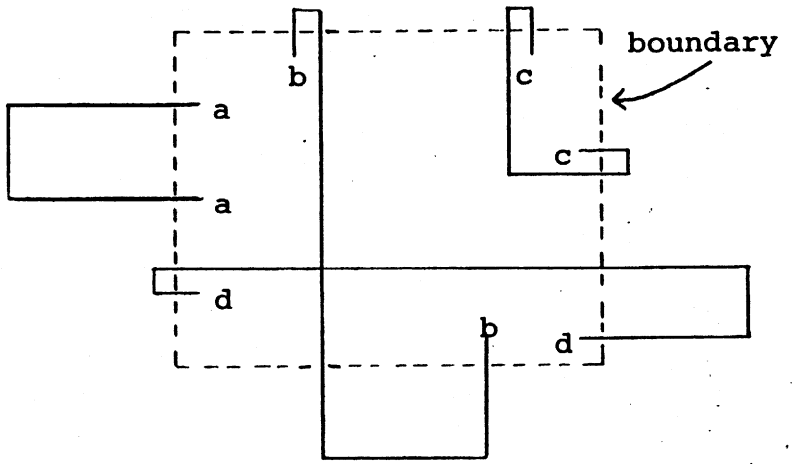

Figure 5-2: *Rewiring the outer portion of a layout.*

It is not difficult to construct N-node graphs with $f(N)$-separators which have $\log N$ diameter for any $f(N)$. By Theorem 5-2, any layout of such a graph must have a wire of length $\Omega(f(N)/\log N)$. Using crossing number and wire area arguments, however, we will find examples of graphs which must contain even longer wires. In particular, we will describe

1) an N-node planar graph for which any layout must have a wire of length $\Theta(N^{1/2}/\log^{1/2}N)$,

2) an N-node graph with an $O(N^{1/2})$-separator for which any layout must have a wire of length $\Theta(N^{1/2}\log N/\log\log N)$, and

3) an N-node graph with an $O(N^{1-1/r})$-separator for which any layout must have a wire of length $\Theta(N^{1-1/r})$ for any $r \geq 3$.

The latter two results achieve the known upper bounds for maximum wire length. They also indicate that some wires in some layouts must be very long (possibly as long as the length of the entire layout).

For convenience, we have summarized our edge length results along with the

previously known upper and lower bounds in Table 5-4. The upper bounds are due to Bhatt and Leiserson [BL81] while the lower bounds are all easy corollaries of Theorem 5-2.

Table 5-4

Maximum Edge Length Bounds

separator	previous lower bound	our lower bound	upper bound
N^α, $\alpha < 1/2$	$N^{1/2}/logN$		$N^{1/2}/logN$
N^α, $\alpha = 1/2$	$N^{1/2}/logN$	$N^{1/2}logN/loglogN$	$N^{1/2}logN/loglogN$
N^α, $\alpha > 1/2$	$N^\alpha/logN$	N^α	N^α
(planar)	$N^{1/2}/logN$	$N^{1/2}/log^{1/2}N$	$N^{1/2}logN/loglogN$

CHAPTER 6

NETWORK CONSTRUCTIONS

In this chapter, we will describe the networks for which we will later establish layout area and maximum edge length lower bounds. As the networks are new and interesting in their own right, we will discuss each at some length.

6.1 The 2-Dimensional Mesh of Trees

The N-node 2-dimensional mesh of trees will be the first example of a graph with an $O(N^{1/2})$-separator known to have layout area $\Theta(N\log^2 N)$ and maximum edge length $\Theta(N^{1/2}\log N/\log\log N)$.

6.1.1. Definition

The *2-dimensional nxn mesh of trees* $M_{2,n}$ (where n is assumed to be a power of 2) is defined as follows. Starting with an nxn matrix of nodes and adding nodes wherever necessary, construct a complete binary tree in each row and column of the matrix. The trees should be constructed so that

1) the leaves in each tree are precisely the nodes in the corresponding row or column of the original matrix, and

2) the subgraph induced on the nodes in each quadrant is $M_{2,n/2}$.

For example, we have drawn $M_{2,4}$ in Figure 6-1. The nodes in the original $4x4$ matrix are represented by dots. The nodes which were added in order to form row trees are drawn as small triangles while those added to form column trees are shown as small squares. The row tree edges are drawn with solid lines while dashed lines represent edges of column trees. Notice that if we were to remove the roots of the row and column trees of $M_{2,4}$ and the edges incident to them, we would be left with 4 copies of $M_{2,2}$, one in each quadrant. In general, if we remove the nodes and edges in the top k levels of the binary trees in $M_{2,n}$, we will be left with 2^{2k} copies of $M_{2,n/2^k}$. This important property of meshes of trees is used extensively throughout Chapters 7 and 8.

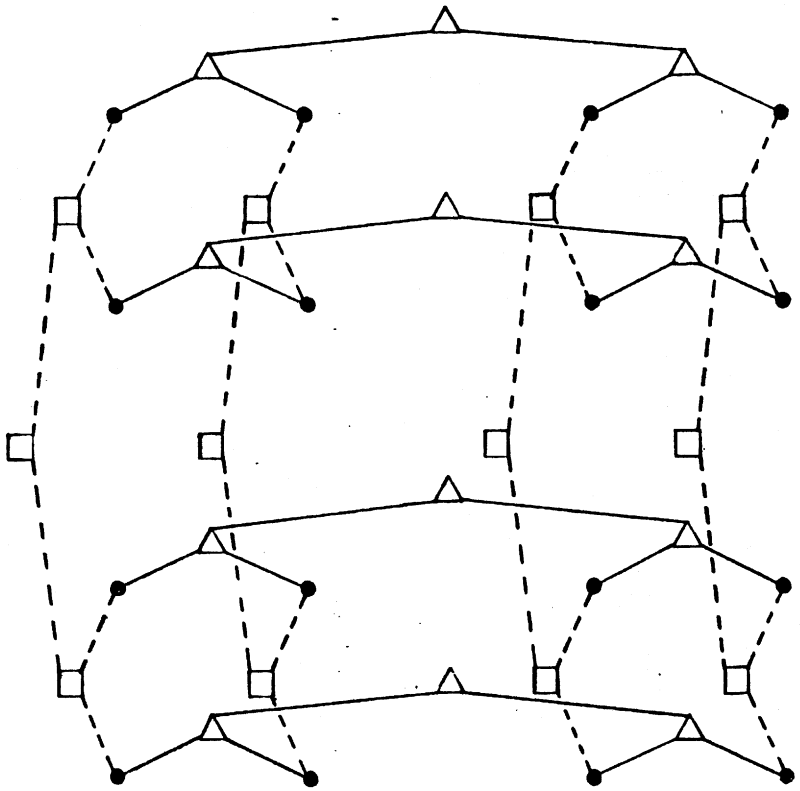

Figure 6-1: *The 4x4 mesh of trees* $M_{2,4}$.

6.1.2. Properties

It is not difficult to show that the *nxn* mesh of trees $M_{2,n}$ has

1) $N = 3n^2-2n = \Theta(n^2)$ nodes,

2) bisection width $n = \Theta(N^{1/2})$,

3) diameter $4\log n = \Theta(\log N)$, and

4) an $O(N^{1/2})$-separator.

By applying the methods discussed in Chapter 5, we can thus conclude that the *N*-node *2*-dimensional mesh of trees has

1) crossing number at most $O(N\log^2 N)$,

2) layout area between $\Omega(N)$ and $O(N\log^2 N)$, and

3) maximum edge length between $\Omega(N^{1/2}/\log N)$ and $O(N^{1/2}\log N/\log\log N)$.

In fact, we will show in Chapters 7 and 8 that the N-node 2-dimensional mesh of trees has

1) crossing number $\Theta(NlogN)$,

2) layout area $\Theta(Nlog^2N)$, and

3) maximum edge length $\Theta(N^{1/2}logN/loglogN)$.

Thus the 2-dimensional mesh of trees is the first graph with an $O(N^{1/2})$-separator known to acheive the upper bound for layout area discovered by Leiserson [L80a] and the upper bound for maximum edge length discovered by Bhatt and Leiserson [BL81].

6.1.3 Applications

Computationally, the nxn mesh of trees is a very powerful network. Among other things, it can be used to

1) multiply a fixed nxn matrix by m different n-vectors in $m+2logn$ (word) steps,

2) sort a list of n m-bit words in $2m+5logn$ (bit) steps, and

3) link n input terminals to n output terminals in any order in $logn$ (bit) steps.

The algorithms and processors needed for these operations are quite simple. For example, the processors needed for sorting and switching need only contain a few *and* and *or* gates while those for matrix-vector multiplication need only contain a word multiplier or adder. We describe the algorithms needed for these operations in the following three subsections.

(a) Matrix-Vector Multiplication

Given any fixed nxn matrix $S=(s_{ij})$, we will show how to program $M_{2,n}$ to compute the product of S and any m input n-vectors in $m+2logn$ (word) steps. As S is fixed, it is not considered to be part of the on-line input. Rather, it is considered to be part of the program (in the form of off-line input) and thus we assume that the value of s_{ij} is initially stored in the (i,j) leaf of $M_{2,n}$ for each i and j. The algorithm proceeds as follows.

Given any input vector $v=(v_j)$, input the jth entry v_j into the root of the jth

column tree for each j, $1 \leq j \leq n$. Pass the entries of v down the column trees so that after $logn$ steps, each leaf in the jth column tree has received the value of v_j. Computation of the n^2 products $\{s_{ij}v_j \mid 1 \leq i, j \leq n\}$ can now take place simultaneously. Afterwards, we can find the entries of the product vector Sv by summing the values of the leaves in each row tree. This operation takes an additional $logn$ steps.

The total running time of the algorithm just described is $1+2logn$. By pipelining the input vectors through the column trees and the output sums through the row trees, it is not difficult to see that m such products can be calculated in $m+2logn$ steps.

(b) Sorting

The algorithm for sorting proceeds as follows. Starting at the roots, input (bit by bit) the ith word to be sorted into the ith row and column trees for each i, $1 \leq i \leq n$. Pass the bits down each tree so that after $logn$ steps the leading bit of the ith word has reached each leaf of the ith row and column trees. Comparison of the ith and jth words for all i and j can now proceed simultaneously. After at most m additional steps, the (i,j) leaf has decided whether the ith word is smaller or larger than the jth word. Ties are broken arbitrarily (e.g., depending on the values of i and j). Once this is done, each leaf transmits a 0 or a 1 to its column tree father depending on whether its column tree word was smaller or larger than its row tree word. Each column tree then sums these values in order to determine the position of its word in the final ordering. (If the sum is carried out bit by bit starting with the least significant bit, this process takes $2logn$ steps.) This information is then used to mark a path in each column tree from the root to that leaf which is also in the appropriate row tree (again taking $2logn$ steps). It is now a simple matter to transmit the bits of the ith word along the unique path from the ith column tree root to the appropriate row root for each i. As the paths are all pairwise disjoint, this process takes only $m+2logn$ steps.

The algorithm just described sorts a list of n m-bit numbers in $2m+7logn$ steps. It is a simple exercise to speed up the alogorithm to obtain the $2m+5logn$ step bound. We should also point out that this algorithm is similar to the one described by Muller and Preparata in [MP75]. The VLSI implementation of the algorithm is new, however, and far superior to many of the VLSI sorting algorithms discussed by Thompson in his recent survey paper [T81].

(c) Switching

Given the algorithm just described for sorting, it is clear how to program $M_{2,n}$ to serve as a switching network for n input and output lines. For example, assume that the *ith* input line is to be connected to the *jth* output line for some i and j. In order to do this, we first hook up the *ith* input line to the *ith* column root. We next establish a path from the root of the *ith* cloumn tree to that leaf in the tree which is also in the *jth* row tree. This can be done by inspection of the binary representation $b_1 \cdots b_{logn}$ of the number j. More precisely, at the *kth* level of the binary tree, we branch left or right depending on whether b_k is *0* or *1* (respectively). Lastly, we link the appropriate leaf of the *jth* row tree to the root of the *jth* row tree and then to the *jth* output line (again taking *logn* steps).

The algorithm just described takes *2logn* steps to link n input lines to n output lines in any order. It is not difficult to show that if the row tree connections are hardwired in advance (i.e., by linking the root of each row tree to all of its leaves), then the input-output connections can be properly made in just *logn* steps.

6.2 The r-Dimensional Mesh of Trees

The N-node r-dimensional mesh of trees (for $r>2$) will be the first example of a graph with an $O(N^\alpha)$-separator (for $\alpha>1/2$) known to have maximum edge length $\Theta(N^\alpha)$.

6.2.1 Definition

The *2*-dimensional mesh of trees can be easily generalized to higher dimensions. For example, the *3-dimensional nxnxn mesh of trees* $M_{3,n}$ can be constructed as follows. Starting with an *nxnxn* cube of nodes and adding nodes wherever necessary, construct a set of n^2 complete binary trees in each of the three dimensions of the cube. As before, the trees should be constructed so that the leaves are precisely the nodes of the original cube and so that the subgraph induced on each octant of nodes is $M_{3,n/2}$. The general *r-dimensional mesh of trees* $M_{r,n}$ is formed from an $\overbrace{nxnx \cdots xn}^{r}$ hypercube in a similar manner. In general, removal of the roots and edges which are in the top level of the binary trees will leave 2^r copies of $M_{r,n/2}$.

6.2.2 Properties

It is easily observed that the r-dimensional $\overbrace{nxnx\ldots xn}^{r}$ mesh of trees $M_{r,n}$ has (for bounded r)

1) $N = (r+1)n^r - rn^{r-1} = \Theta(n^r)$ nodes,

2) bisection width $n^{r-1} = \Theta(N^{1-1/r})$,

3) diameter $2r\log n = \Theta(\log N)$, and

4) an $O(N^{1-1/r})$-separator.

Thus we can easily infer that the N-node r-dimensional mesh of trees has (for bounded r)

1) crossing number at most $O(N^{2-2/r})$,

2) layout area $\Theta(N^{2-2/r})$, and

3) maximum edge length between $\Omega(N^{1-1/r}/\log N)$ and $O(N^{1-1/r})$.

In fact, we will show in Chapter 7 that the graph has

1) crossing number $\Theta(N^{2-2/r})$, and

2) maximum edge length $\Theta(N^{1-1/r})$.

Thus the r-dimensional mesh of trees is the first graph with an $O(N^{\alpha})$-separator (for $\alpha > 1/2$) known to achieve the trivial upper bound on maximum edge length.

6.2.3 Application to Matrix Multiplication

Computationally, the r-dimensional mesh of trees is a very powerful network. For example, $M_{r,n}$ can be used to multiply m pairs of nxn matrices in $m+2\log n$ (word) steps. The algorithm is very similar to the one used by $M_{2,n}$ to compute matrix-vector products. It procceds as follows.

At each time step, a pair of matrices is entered into the network via the roots of the trees in two of the dimensions (one dimension for each matrix). The entries are passed down through the trees so that after $\log n$ steps, the leaf in the (r,s,t) position of the cube contains the (r,s) entry of the first matrix and the (s,t) entry of the second matrix for each r,s and t. All n^3 multiplications can then be performed simultaneously. The entries of the product matrix are then calculated by summing

the values of the leaves of each tree in the third (previously unused) dimension. This process takes an additional *logn* steps. As the network is easily pipelined, it is clear that the total computation time is just $m+2logn$ (word) steps.

6.2.4 A Further Generalization

The r-dimensional mesh of trees was defined as a natural generalization of the computationally powerful 2-dimensional mesh of trees. $M_{r,n}$ can also be viewed as a generalization of the r-cube, also a very powerful communications network. For example, $M_{r,2}$ is an r-cube with every edge replaced by a path of length 2. Viewed in this light, the r-dimensional mesh of trees motivates the definition of a *shuffle-tree graph* in the same way that the r-cube motivates the definition of the shuffle-exchange graph. Although we have yet to investigate this graph in detail, it is quite possible that it has important applications.

(As an aside, we should caution the reader that the asymptotic estimates given in section 6.2.2 do not necessarily apply to $M_{r,2}$ since r was assumed to be bounded. The correct estimates are not difficult to work out, however.)

6.3 The Tree of Meshes

The N-node tree of meshes will be the first example of a planar graph known to have $\Theta(NlogN)$ layout area.

6.3.1 Definition

The *tree of meshes* is similar to the 2-dimensional mesh of trees in that it combines the structure of a mesh with that of a complete binary tree in a natural way. Unlike the 2-dimensional mesh of trees, however, the tree of meshes is a planar graph. It is formed by replacing each node of a complete binary tree with a mesh and each edge by several edges which link the meshes together. More precisely, the root of the binary tree is replaced by an nxn mesh (where n is assumed to be a power of 2), its sons are replaced by $n/2 \times n$ meshes, their sons are replaced by $n/2 \times n/2$ meshes, and so on until the leaves are replaced by $1x1$ meshes. In the place of each *right edge* of the binary tree (i.e., one which links a node to its right son), we link the rightmost column of nodes in the mesh corresponding to the father to the topmost row of nodes in the mesh corresponding to the right son. Similar replacements are made for *left edges* of the binary tree. In both cases, the connections are made so as to preserve the column and row order

of the nodes and to insure that the resulting graph will be planar. The resulting graph is refered to as the *nxn* tree of meshes and will be denoted by T_n. For example, we have drawn T_4 in Figure 6-2.

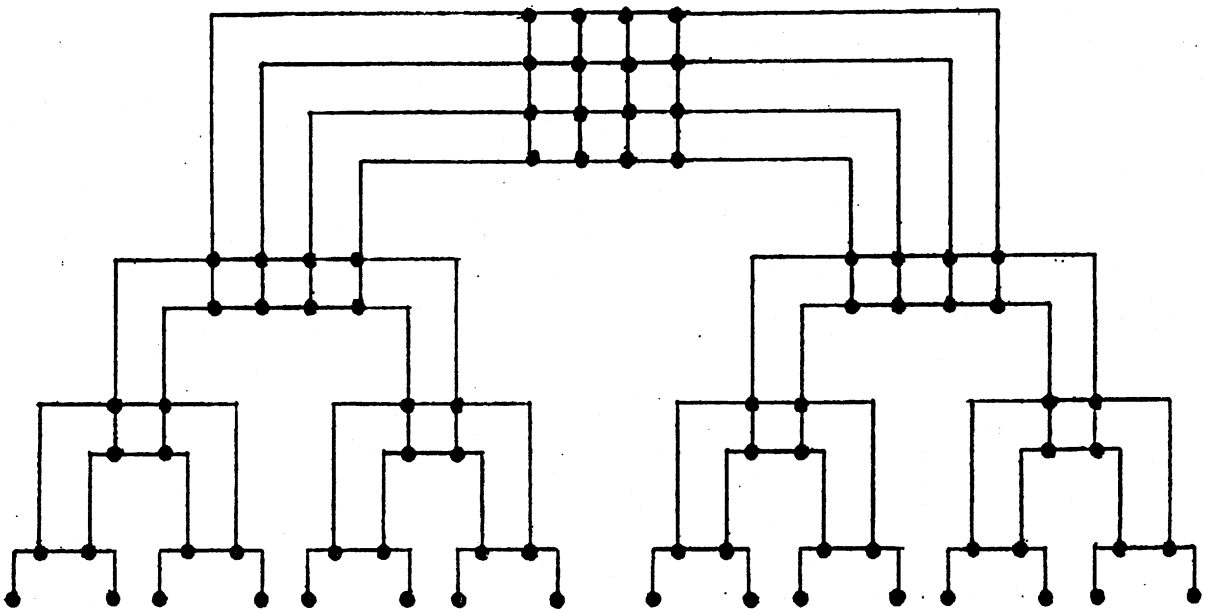

Figure 6-2: *The 4x4 tree of meshes* T_4.

6.3.2 Properties

It is easily seen that the *nxn* tree of meshes T_n has

1) $N = 2n^2 \log n + n^2 = \Theta(n^2 \log n)$ nodes,

2) bisection width $n = \Theta(N^{1/2}/\log^{1/2} N)$,

3) diameter $8n = \Theta(N^{1/2}/\log^{1/2} N)$, and

4) an $O(N^{1/2}/\log^{1/2} N)$-separator.

Thus we can easily infer that the N-node tree of meshes has

1) layout area between $\Omega(N)$ and $O(N \log N)$, and

2) maximum edge length between $\Omega(\log^{1/2} N)$ and $O(N^{1/2} \log^{1/2} N)$.

In fact, we will show that the graph has

1) layout area $\Theta(NlogN)$ and

2) maximum edge length $\Theta(logN)$.

The maximum edge length bound is fairly straightforward. We will show in Chapter 8 that the wire area of the N-node tree of meshes is $\Theta(NlogN)$. As the graph has $\Theta(N)$ wires, we can conclude that some of them must have length at least $\Omega(logN)$. The lower bound can, in fact, be achieved by a straightforward modification of the H-tree layout for binary trees [MR79].

In section 6.4, we will show how to augment the N-node tree of meshes so that any layout will have to contain a wire of length at least $\Omega(N^{1/2}/log^{1/2}N)$.

6.3.3 Applications

The tree of meshes is a particularly interesting planar graph since it can embed arbitrary planar graphs much more efficiently than can the ordinary mesh. For example, it is not known how to embed an arbitrary planar graph in less than an $\Theta(Nlog^2N)$-node mesh. As we show in part (a) of this section, however, any N-node planar graph can be embedded in an $O(NlogN)$-node tree of meshes.

The tree of meshes can also be used to embed many nonplanar graphs which have $O(N^{1/2})$-separators. For example, we will show in part (b) of this section how to embed $M_{2,n}$ in T_{2n} for any n. This result will later allow us to give a simple proof that the N-node tree of meshes has wire area at least $\Omega(NlogN)$.

(a) Embeddings of Planar Graphs

In [LT77], Lipton and Tarjan prove an $O(N^{1/2})$-separator theorem for the class of planar graphs. Recently, Bhatt and Leiserson [BL81] generalized this result by showing that the class of planar graphs has an $O(N^{1/2})$-simultaneous separator. (An N-node graph G is said to have an *$f(N)$-simultaneous separator* if for any *2*-coloring (say, black and white) of the nodes of G, there are disjoint subgraphs G_1 and G_2 of G such that G_1 and G_2 each contain *1/2* of the black nodes and *1/2* of the white nodes of G, at most $f(N)$ edges link G_1 to G_2, and both G_1 and G_2 have $f(N/2)$-simultaneous separators.) In the following theorem, we show that any N-node graph with an $O(N^{1/2})$-simultaneous separator can be embedded in an $O(NlogN)$-node tree of meshes. As a corollary, we will thus be able to conclude that any N-node planar graph can be embedded in an $O(NlogN)$-node tree of meshes.

Theorem 6-1: *Every N-node graph with an* $O(N^{1/2})$*-simultaneous separator can be embedded in an* $O(N\log N)$*-node tree of meshes.*

Proof: Let G be an N-node graph with an $f(N)$-simultaneous separator ($f(N)$ will later be chosen to be $O(N^{1/2})$). Partition G into two subgraphs G_1 and G_2 in accordance with the usual separator theorem. Color the nodes of G_1 (G_2) white or black according to whether or not they are linked to a node in G_2 (G_1). (To be precise, we should also weight each node according to the number of nodes in the other subgraph to which it is adjacent.) Now use the simultaneous separator to partition G_1 and G_2. Proceed in this manner until only isolated nodes remain. At each step, color the nodes in the subgraph white if they are adjacent to some node outside of the subgraph and black if they are adjacent only to nodes within the subgraph.

After the first step, at most $f(N)$ edges will link each $(N/2)$-node subgraph to the other. After the second step, at most $f(N)/2+f(N/2)$ edges will link each $(N/4)$-node subgraph to any other. Using induction, it is not difficult to show that after k steps, at most

$$f(N)/2^{k-1} + f(N/2)/2^{k-2} + f(N/4)/2^{k-3} + \cdots + f(N/2^{k-2})/2 + f(N/2^{k-1})$$

edges will link each $(N/2^k)$-node subgraph to any other. In particular, for $f(N) = O(N^{1/2})$, we can conclude that at most $O(m^{1/2})$ edges will link any m-node subgraph produced by this process to any other subgraph.

Each subgraph produced by the above procedure corresponds in a natural way to a mesh of the tree of meshes. For example, G corresponds to the root mesh, G_1 and G_2 correspond to the second level meshes, and so on. In general, each m-node subgraph corresponds to an $\Theta(m)$-node mesh. Thus each mesh can be used as a switching network to embed the $O(m^{1/2})$ edges which link the corresponding subgraph to other subgraphs. As an example of how this is done, we have included Figure 6-3. In each switching network, the edges entering from the top are linked to the edges entering from the sides. The nodes of G are embedded in the bottom levels of the tree of meshes □

Corollary 6-1: *Every N-node planar graph can be embedded in an* $O(N\log N)$*-node tree of meshes.*

Proof: Obvious □

(b) **Embedding of $M_{2,n}$ in T_{2n}**

Although we have not worked out the details, it appears likely that *any* N-node graph with an $O(N^{1/2})$-separator can be embedded in an $O(NlogN)$-node tree of meshes. In section 7.4.3, we prove a slightly weaker result; namely that every N-node graph with an $O(N^{1/2})$-separator can be embedded in *some* $O(NlogN)$-node planar graph.

Of particular importance, however, is the fact that $M_{2,n}$ can be embedded in T_{2n} for any n. For example, consider the embedding of $M_{2,4}$ in T_8 displayed in Figure 6-4. The embedding has been drawn as though it were construted as part of a larger embedding (say of $M_{2,8}$) in order to illustrate the recursive nature of the general embedding procedure. In addition, the nodes and edges of $M_{2,4}$ have been drawn as they appear in Figure 6-1. For clarity, we have represented the nodes of T_8 as pinpoints and omitted its edges altogether. Also notice that we have not included the bottom two levels of T_8 since they are not needed for the embedding.

The embedding of $M_{2,n}$ in T_{2n} for arbitrary $n \geq 4$ proceeds as follows.

step 1: Remove the roots of the row and column trees of $M_{2,n}$ and all the edges incident to them.

step 2: Embed the four copies of $M_{2,n/2}$ obtained from step 1 in four separate copies of T_n by calling this procedure recursively.

step 3: Embed the $2n$ roots of the row and column trees in the $2n \times 2n$ mesh so that

1) the column roots are located at positions (i,i) for $1 \leq i \leq n/2$ and $3n/2 < i \leq 2n$, and

2) the row roots are located at positions $(2i-1,2i-1)$ and $(2i-1,2i)$ for $n/4 < i \leq 3n/4$.

step 4: Draw left and right horizontal edges from each column root to the left and right outer columns of the $2n \times 2n$ mesh and then to the appropriate node in the top row of the corresponding $n \times 2n$ mesh. Similarly draw two left edges from each row root with position $(2i-1,2i-1)$ for some i and two right edges from each row root with position $(2i-1,2i)$ for some i.

step 5: The $n \times 2n$ meshes are used as switching networks. In particular, we

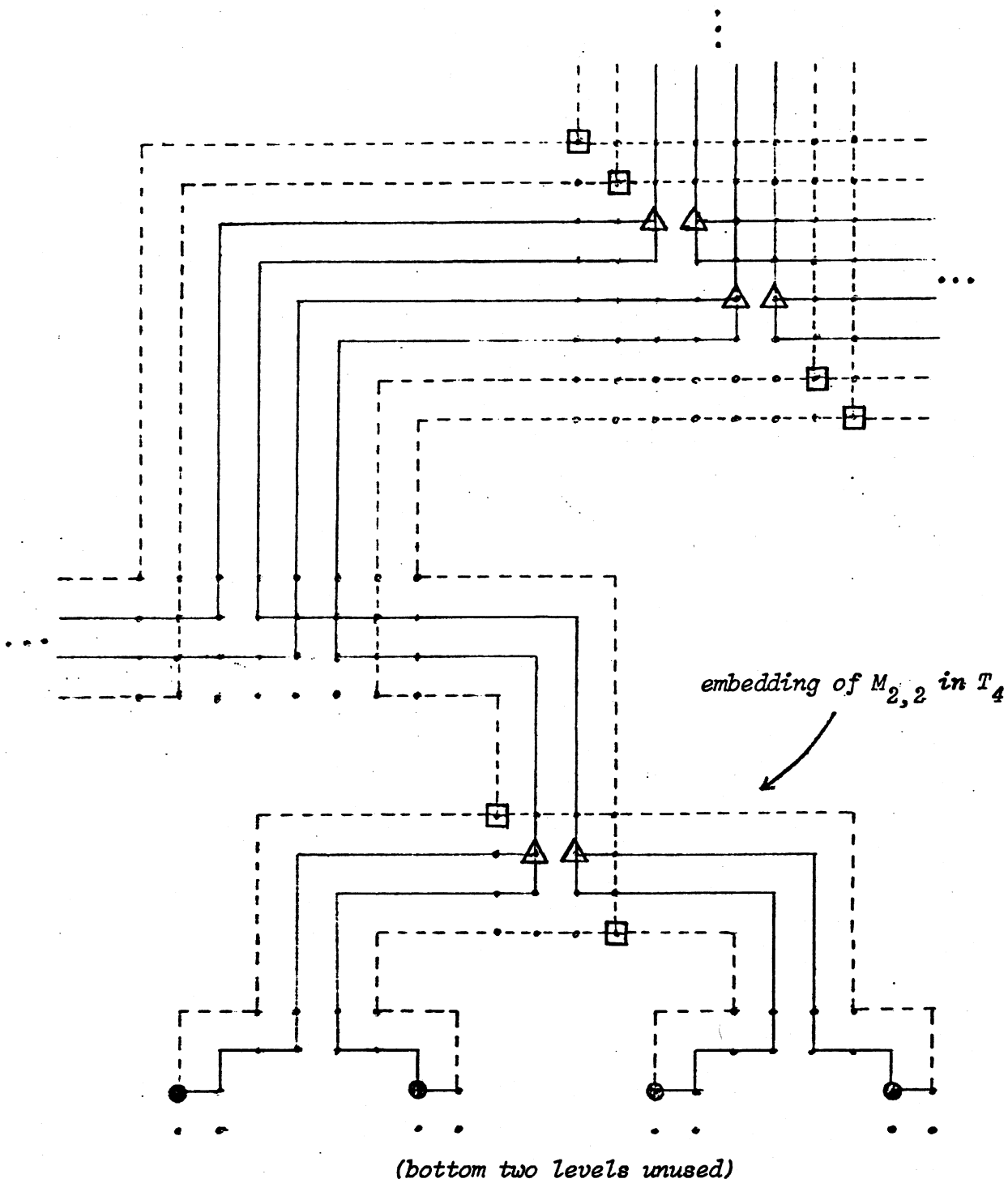

Figure 6-4: *The embedding of $M_{2,4}$ in T_8.*

use them to make the following connections:

1) $(1,i)$ to $(i,1)$ for $1 \leq i \leq n/4$ (column tree connection)

2) $(1,i)$ to $(i+n/2, 1)$ for $n/4 < i \leq n/2$ (column tree connection)

3) $(1, 2i-1)$ to $(i,1)$ for $n/4 < i \leq 3n/4$ (row tree connection)

4) $(1, 2i)$ to $(i, 2n)$ for $n/4 < i \leq 3n/4$ (row tree connection)

5) $(1,i)$ to $(5n/2 - i+1, 2n)$ for $3n/2 < i \leq 7n/4$ (column tree connection)

6) $(1,i)$ to $(2n - i+1, 2n)$ for $7n/4 < i \leq 2n$ (column tree connection)

step 6: Each $n \times 2n$ mesh can be easily linked to two copies of T_n, each of which contains an embedding of $M_{2,n/2}$ produced by this procedure. In particular, attach the wire leaving via the *ith* row of the $n \times 2n$ mesh to the node in the *ith* column of the appropriate nxn mesh of T_n for each n. (Note that the nodes in the nxn meshes are roots of $M_{2,n/2}$ and will become second level nodes of $M_{2,n}$).

6.4 The Augmented Tree of Meshes

As we mentioned in section 6.3.2, the N-node tree of meshes can be laid out so that every wire has length at most $O(\log N)$. By slightly modifying the graph, however, it is possible to increase the maximum edge length dramatically. The basic idea is to add a complete binary tree with n^2 leaves to the nxn tree of meshes so that the leaves of one are linked in a one-to-one fashion with the leaves of the other. It is important that the attachments between the two graphs be made so that the resulting graph (which we call the *nxn augmented tree of meshes* T_n') is planar. For example, we have drawn the *4x4* augmented tree of meshes in Figure 6-5.

It is easily seen that the augmented tree of meshes has, up to a constant, the same bisection width, diameter, separator, layout area and number of nodes as does the original tree of meshes. By adding the binary tree, we have simply decreased the distance between any two *leaves* of the tree of meshes. In Chapter 8, we will show that any layout of the N-node tree of meshes has two leaves which are spaced at least $\Omega(N^{1/2}\log^{1/2}N)$ apart. We will thus be able to conclude that the maximum edge length of T_n' is at least $\Omega(n\log n) = \Omega(N^{1/2}/\log^{1/2}N)$. Using the techniques developed by Bhatt and Leiserson in [BL81], it is not difficult to show that the lower bound is attainable.

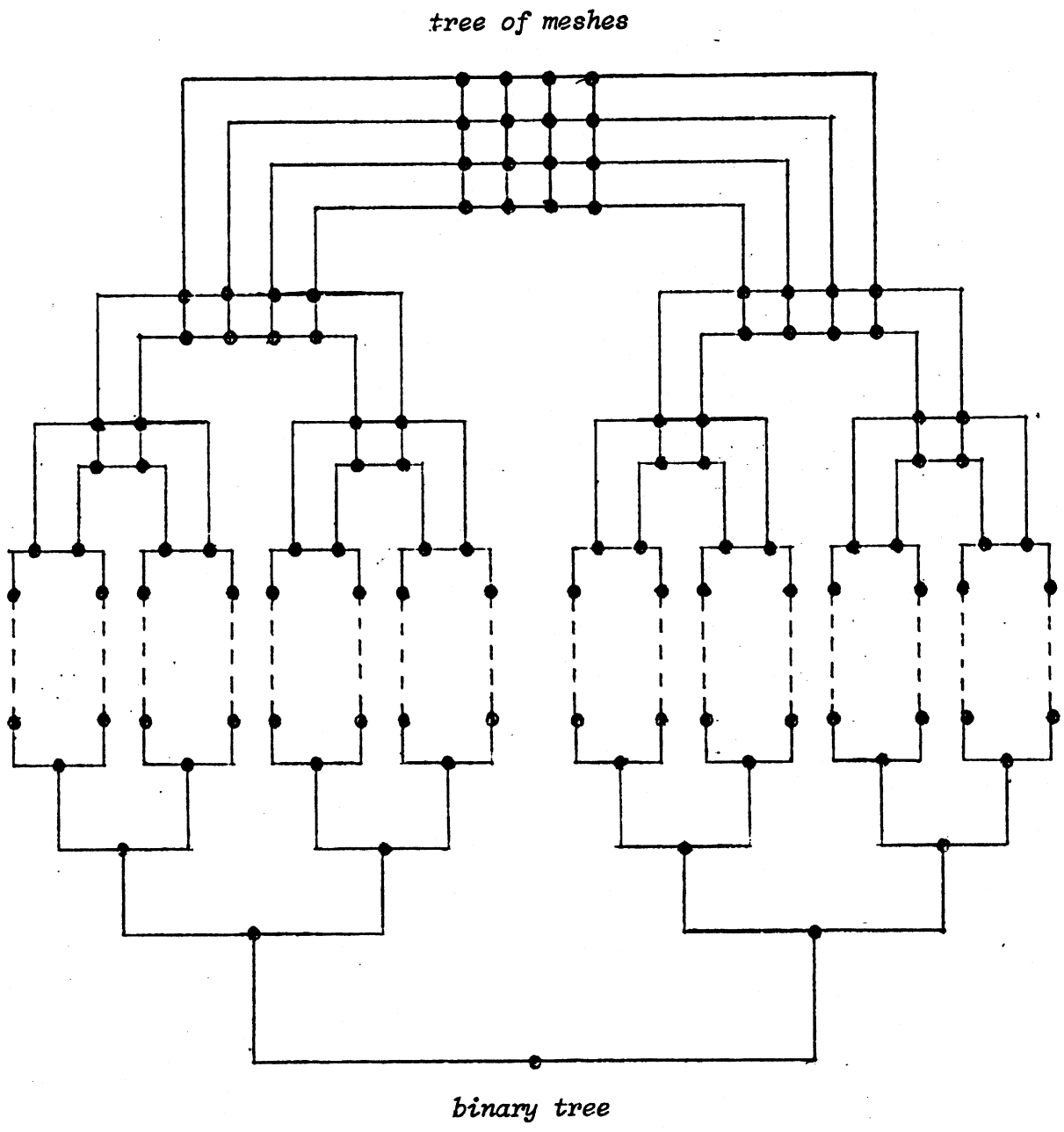

Figure 6-5: *The 4x4 augmented tree of meshes T_4'.*

CHAPTER 7

CROSSING NUMBER ARGUMENTS

In this chapter, we demonstrate the power of the crossing number as a lower bound technique for VLSI. We commence by showing that the crossing number is at least as large (up to a constant) as the square of the bisection width. In section 7.2, we describe a powerful method for finding crossing number lower bounds. This method is then used in section 7.3 to find tight lower bounds on the crossing numbers of a variety of networks. We conclude in section 7.4 with a collection of miscellaneous results. Included are additional upper and lower bounds for the crossing number of a network as well as a procedure for embedding an arbitrary N-node graph with an $O(N^{1/2})$-separator in an $O(N\log N)$-node planar graph.

7.1 The Relationship Between Crossing Number and Layout Area

We first show that crossing number arguments are at least as powerful as bisection width arguments in establishing lower bounds for layout area.

Theorem 7-1: *If G is an N-node graph with crossing number c and bisection width b, then* $c+N \geq \Omega(b^2)$.

Proof: Let D be a drawing of G in the plane with c crossings. Replace each crossing of D with an *artificial node.* Call the resulting graph G' and note that it has precisely $c+N$ nodes. Using the weighted version of the Lipton-Tarjan planar separator theorem [LT77], it is possible to bisect the real nodes of G' (by assigning weight *1* to the real nodes and weight *0* to the artificial nodes) without cutting more than $O((c+N)^{1/2})$ edges. After replacing the artificial nodes with their original edge crossings, it becomes apparent that we have, in fact, constructed an $O((c+N)^{1/2})$ bisection for G. Squaring, we find that $c+N \geq \Omega(b^2)$ □

Using a similar proof technique, we can show that the crossing number is also close to an upper bound for the layout area of a graph. In fact, should a really good layout algorithm for planar graphs be found, then the following result could become useful in laying out arbitrary graphs.

Theorem 7-2: *Given an optimal drawing D for an N-node graph G with crossing-number c, it is possible to construct a layout for G with area at most* $O((c+N)log^2(c+N))$. *Should a procedure be found which lays out an arbitrary N-node planar graph in* $A(N)$ *area, then we could construct a layout for G with area at most* $O(A(c+N))$.

Proof: As in the proof of Theorem 7-1, we replace each edge crossing of D with an artificial node. The resulting graph G' has $c+N$ nodes and is planar. Using the methods developed by Lipton and Tarjan [LT77] and Leiserson [L80a], G' can be laid out in $O((c+N)log^2(c+N))$ area. It is then a simple matter to replace the artificial nodes with their original edge crossings to obtain the desired layout for G. Alternatively, should an $A(N)$-area planar graph layout procedure be discovered, we could construct an $O(A(c+N))$-area layout for G □

As we have just seen, the idea of replacing edge crossings with artificial nodes is simple but powerful. Jai-Wei and Rosenberg have also employed this strategy in their work with embeddings of graphs in binary trees [JR81].

7.2 A General Method for Proving Lower Bounds

In this section, we will describe a general method for proving crossing number lower bounds. A variant of this method will later be used to prove lower bounds for bisection width and wire area. The basic idea is as follows.

Given a drawing D for an N-node graph G, we will construct a drawing D' for the complete graph on N nodes K_N by tracing over the edges of D. For example, we have done this for the 4-node graph shown in Figure 7-1. The edges of the original graph are drawn with dashed lines while solid lines indicate edges of K_4.

If we are careful not to trace over each edge of D too many times during the construction of D', it may be possible to infer something about the number of crossings in D by counting the number of crossings in D'. This is due to the fact that the number of crossings in D is closely related to the number of crossings in D'. For example, if e_1 and e_2 are edges of G which cross in D and e_1 is traced over s_1 times while e_2 is traced over s_2 times, then the crossing of e_1 with e_2 will appear s_1s_2 times in D'. Such a crossing of D' is called a *crossing of the first kind.* For example, there are four crossings of the first kind in the drawing of K_4 in Figure 7-1.

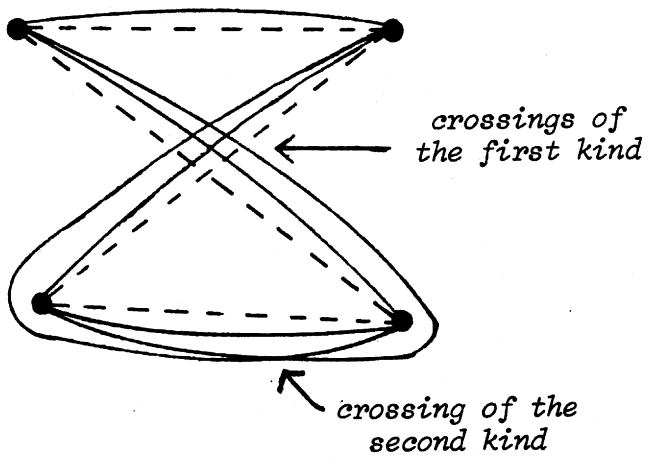

Figure 7-1: *Construction of K_4 from the drawing of a 4-node graph.*

Sometimes, it is necessary for two edges of D' to cross while traversing the same edge of D. Such a crossing is called a *crossing of the second kind.* Note that there is only one crossing of the second kind in the drawing of K_4 in Figure 7-1. Since D' can easily be drawn so that no pair of edges cross each other more than once, there are usually not very many crossings of the second kind. More precisely, if G has edges $e_1, \ldots, e_k$ and if edge e_i is traced over s_i times for each i during the construction of D', then D' can have at most $\sum_{i=1}^{k} s_i^2/2$ crossings of the second kind. For most applications of the method, this number is substantially smaller than the number of crossings of the first kind in D' and thus we usually do not have to worry about crossings of the second kind.

By showing that the number of crossings in D' is large, we can conclude that there must be a large number of crossings in D. For example, if each edge of D is traced over at most s times during the construction of D' and D' is found to have y crossings, then we can conclude that D has at least y/s^2 crossings. This follows from the fact that each crossing of D is replicated at most s^2 times in D'. (Note that we have neglected crossings of the second kind in this argument.)

Fortunately, it is easy to find a good lower bound on the number of crossings in any drawing of K_N. We state the result formally in the following lemma. The proof can also be found in Kleitman's work [K70] but is generally regarded as folklore.

Lemma 7-1 (Kleitman [K70]): *The crossing number of K_N, the complete graph on N nodes, is at least $N(N-1)(N-2)(N-3)/120$ for $N \geq 5$.*

Proof: Let D be a drawing of K_N in the plane with the smallest possible number of crossings $c(N)$. We may assume that no pair of edges which cross in D are incident to a common node. Otherwise, it would be possible to produce a drawing D' for K_N with $c(N)-1$ crossings by exchanging the parts of the crossing edges which lie between the common node and the point of crossing. This would contradict the minimality of $c(N)$.

Consider the N subdrawings of D obtained by deleting one of the nodes and all of the edges incident to it. Note that each crossing of D appears in precisely $N-4$ of the subdrawings. (A crossing does not appear in any of the 4 subdrawings which correspond to the deletion of a node incident to an edge of the crossing.) Since each of the subdrawings is a drawing of K_{N-1}, each must have at least $c(N-1)$ crossings. Thus $(N-4)c(N) \geq Nc(N-1)$. Applying the inequality recursively and noting that $c(5)=1$, we can conclude that

$$c(N) \geq [N/(N-4)]\,[(N-1)/(N-5)] \cdots [6/2]$$

$$= N(N-1)(N-2)(N-3)/120 \quad \text{for } N \geq 5 \quad \square$$

7.3 Applications

Using the technique described in the previous section, it is possible to prove crossing number lower bounds for a variety of networks. In particular, we will prove lower bounds for the shuffle-exchange graph, the 2-dimensional mesh of trees and the r-dimensional mesh of trees. We commence with the shuffle-exchange graph.

7.3.1 Lower Bounds for the Shuffle-Exchange Graph

Our main result in this section is the following.

Theorem 7-3: *The crossing number of the N-node shuffle-exchange graph is $\Theta(N^2/\log^2 N)$.*

Proof: As we showed in Part I of the thesis, the N-node shuffle-exhange graph has layout area $\Theta(N^2/\log^2 N)$. Thus $O(N^2/\log^2 N)$ is an upper bound for the crossing number. In what follows, we will use the method of section 7.2 in order to

show that the crossing number of the N-node shuffle-exchange graph is at least $\Omega(N^2/\log^2 N)$.

Let D be any drawing of the N-node shuffle-exchange graph G where $N=2^k$. We first show how to construct a drawing D' of K_N on the nodes of G without tracing over any edge of D more than $N\log N$ times.

Given any pair of nodes $a_k \cdots a_1$ and $b_k \cdots b_1$, draw the edge from $a_k \cdots a_1$ to $b_k \cdots b_1$ along the path

$$a_k \cdots a_3 a_2 a_1 \longrightarrow a_k \cdots a_3 a_2 b_1 \longrightarrow b_1 a_k \cdots a_3 a_2 \longrightarrow b_1 a_k \cdots a_3 b_2 \longrightarrow$$
$$b_2 b_1 a_k \cdots a_3 \longrightarrow \cdots \longrightarrow b_{k-1} \cdots b_2 b_1 b_k \longrightarrow b_k b_{k-1} \cdots b_2 b_1 .$$

(In order that every edge of K_N not be drawn twice, we should assume that the value of $a_k \cdots a_1$ is less than that of $b_k \cdots b_1$ but this has no bearing on the argument.)

Wherever $a_i = b_i$ for some i, the preceding path will have a loop. When actually drawing the edges of D', we ignore such loops. For example, the edge from *01100* to *11101* is drawn along the path

$$01100 \xrightarrow{e} 01101 \xrightarrow{s} 10110 \xrightarrow{s} 01011 \xrightarrow{s} 10101 \xrightarrow{s} 11010 \xrightarrow{e}$$
$$11011 \xrightarrow{s} 11101 .$$

For convenience, we have labeled the shuffle edges with an $\xrightarrow{s}$ and the exchange edges with an $\xrightarrow{e}$. Note also that we have omitted loops at *10110*, *01011* and *10101*.

It is not difficult to show that every edge of D is traced over at most $N\log N$ times during the construction of D'. For example, consider the shuffle edge linking $a_k \cdots a_2 a_1$ to $a_1 a_k \cdots a_2$. It is traced over during the construction of edges of D' which link a node of the form

$$a_{k-i+1} \cdots a_2 \overbrace{* \cdots *}^{k-i}$$

to a node of the form

$$\overbrace{* \cdots *}^{i} a_1 a_k \cdots a_{k-i+2}$$

for any i, $1 \le i \le k$ (where $*$ indicates either a *0*-bit or a *1*-bit). It is easily seen that there are at most $k2^k$ such edges in D' and thus each shuffle edge is traced over at most $N\log N$ times. A similar argument shows that each exchange edge is also

traced over at most $NlogN$ times.

Since each edge is traced over at most $NlogN$ times, there can be at most

$$(3N/2)\,[(NlogN)^2/2] \;=\; 3N^3/(4log^2N)$$

crossings of the second kind in D'. This is substantially less than total number $\Omega(N^4)$ of crossings in D'. Thus D' must have $\Omega(N^4)$ crossings of the first kind. As each edge of D is traced over at most $NlogN$ times, this means that D has at least $\Omega(N^4/(NlogN)^2) = \Omega(N^2/log^2N)$ crossings □

As the N-node shuffle-exchange graph has $\Theta(N)$ edges, we can conclude from Theorem 7-1 that some edge of any layout for the graph must cross at least $\Omega(N/log^2N)$ other edges. We do not know whether or not this bound can be achieved, however. The only known layouts for the N-node shuffle-exchange graph have edges which cross at least $\Omega(N/logN)$ other edges.

It is also worth pointing out that the preceding argument can be used to prove that the N-node shuffle-exchange graph has bisection width at least $\Omega(N/logN)$. The result follows from the observation that K_N has bisection width $\Theta(N^2)$ and the fact that every edge of D was traced over at most $NlogN$ times during the construction of D'. This means that the bisection width of the N-node shuffle-exchange graph is at least $\Omega(N^2/(NlogN)) = \Omega(N/logN)$, as claimed.

In fact, a similar modification of the method described in section 7.2 can be used to find tight bisection width lower bounds for *all* of the networks we have investigated. For most of these networks, however, it is much more useful to study the corresponding crossing number and wire area bounds.

7.3.2 Lower Bounds for the 2-Dimensional Mesh of Trees

In this section, we use a more sophisticated version of the method of section 7.2 to prove a nontrivial lower bound on the crossing number of the 2-dimensional mesh of trees.

Theorem 7-4: *The crossing number of the N-node 2-dimensional mesh of trees is at least* $\Omega(NlogN)$.

Proof: As before, let $M_{2,n}$ denote the 2-dimensional mesh of trees (where n is a power of 2). We will show that the crossing number of $M_{2,n}$ is at least

$(n^2 \log n - 121n^2 + 121n)/40$ for all $n \geq 1$.

Since $M_{2,n}$ has $N = \Theta(n^2)$ nodes, this will be sufficient to prove the desired result.

The proof consists of two steps. In the first, we show how to construct a drawing of K_{n^2} from any drawing of $M_{2,n}$ by tracing over the edges of $M_{2,n}$. We then apply Lemma 7-1 to conclude that there are a large number of crossings among the edges in the top levels of the binary trees of $M_{2,n}$. In the second step, we complete the proof by inductively applying the result of the first step.

step 1: Let D be any drawing of $M_{2,n}$ in the plane. From this drawing, we can construct a drawing D' of K_{n^2} in the following way. First locate the n^2 leaves of the binary trees of D. They will serve as the nodes for K_{n^2}. Given any pair (i,j) and (k,l) of these nodes, draw an edge from (i,j) to (k,l) along the unique path from (i,j) to (i,l) in the *ith* row tree of D and then from (i,l) to (k,l) in the *lth* column tree of D. (In order that each edge not be drawn twice, we shall assume that $i \leq k$ and, when $i = k$, that $j < l$.) As usual, we assume that the edges of D' are drawn so that no pair cross each other more than once.

We next count the number of crossings of the second kind in D'. In order to do this, we need to count the number of times each edge of D is traeed over during the construction of D'. It is not difficult to show that each edge in the *ith* level of a binary tree of $M_{2,n}$ (henceforth, referred to as a *type i* edge) is traced over at most

$$n2^{-i}(n^2 - n^2 2^{-i}) \leq n^3 2^{-i}$$

times for any $i \leq \log n$ during the construction of D'. Thus at most $n^6 2^{-2i-1}$ crosses of the second kind can occur at any type i edge of D. Since there are $2^{i+1}n$ type i edges in $M_{2,n}$, we can conclude that the total number of crosses of the second kind in D' is at most

$$\sum_{i=1}^{\log n} (2^{i+1}n)(n^6 2^{-2i-1}) = n^7 \sum_{i=1}^{\log n} 2^{-i} \leq n^7.$$

We next count the number of crossings of the first kind (i.e., those corresponding to crosses in D). We say that a crossing of D is *type i-j* if it is the crossing of a type i edge and a type j edge. Let t_{ij} denote the number of type i-j crossings in D and set $t_i = \sum_{j=i}^{\log n} t_{ij}$. Since each type i edge is traced over at most $n^3 2^{-i}$ times, each type i-j crossing of D produces at most $(n^3 2^{-i})(n^3 2^{-j}) = n^6 2^{-i-j}$ crosses of the first kind in D'. Thus the total number of crossings of the first kind in D'

is at most

$$\sum_{i=1}^{\log n}\sum_{j=i}^{\log n} n^6 2^{-i-j} t_{ij} \le n^6 \sum_{i=1}^{\log n} 2^{-2i} t_i .$$

Summing, we find that the total number of crossings of either kind in D' is at most $n^7 + n^6 \sum_{i=1}^{\log n} 2^{-2i} t_i$. By Lemma 7-1, this number must be at least $n^2(n^2-1)(n^2-2)(n^2-3)/120$ for $n^2 \ge 5$. Simplifying, we can conclude that

$$\sum_{i=1}^{\log n} 2^{-2i} t_i \ge (n^2-121n)/120 \quad \text{for } n \ge 6.$$

Let $s_k = \sum_{i=1}^{k} t_i$ be the number of crossings involving at least one edge from the top k levels of some binary tree of $M_{2,n}$. In what follows, we will use the preceding inequality to show that $s_k \ge (n^2-121n)k/40$ for at least some value of $k \ge 1$. Assume otherwise and observe that

$$\sum_{i=1}^{\log n} 2^{-2i} t_i = \sum_{i=1}^{\log n} 2^{-2i} (s_i - s_{i-1})$$

where s_0 is defined to be 0. The coefficient of each s_i $(i=0)$ in this sum is $2^{-2i} - 2^{-2i-2}$ which is positive so for each i we may substitute $(n^2-121n)i/40$ as an upper bound for s_i in order to see that

$$\sum_{i=1}^{\log n} 2^{-2i} t_i < [(n^2-121n)/40] \sum_{i=1}^{\log n} 2^{-2i}[i-(i-1)]$$
$$= [(n^2-121n)/40] \sum_{i=1}^{\log n} 4^{-i} .$$

Since $\sum_{i=1}^{\log n} 4^{-i} \le 1/3$ for all n, we can conclude that

$$\sum_{i=1}^{\log n} 2^{-2i} t_i < (n^2-121n)/120 \quad \text{for all } n > 121,$$

a contradiction. Thus for all $n \ge 121$, there is a $k \ge 1$ such that $s_k \ge (n^2-121n)k/40$.

step 2: Let $c(n)$ denote the crossing number of $M_{2,n}$. Using the result of step 1, we will now show by induction on n that $c(n) \ge (n^2 \log n - 121n^2 + 121n)/40$ for all $n \ge 1$.

As $(n^2 \log n - 121n^2 + 121n)/40$ is nonpositive for small n, the lower bound trivially holds for all $n < 128$. Assume that the lower bound holds for all $m < n$ where $n \ge 128$ and let D be any drawing for $M_{2,n}$. By counting the crossings of D in two groups according to whether or not at least one edge of the crossing is contained in the top k levels of the binary trees of $M_{2,n}$, we can observe that

$$c(n) \geq 2^{2k}c(n2^{-k}) + s_k .$$

(Recall the definition of s_k and the structure of $M_{2,n}$.) By choosing k as in step 1 so that $s_k \geq (n^2\text{-}121n)k/40$ and applying the inductive hypothesis for $c(n2^{-k})$, we obtain

$$\begin{aligned} c(n) &\geq 2^{2k}[n^2 2^{-2k}(\log n - k)/40 - 121n^2 2^{-2k}/40 + 121n2^{-k}/40] + n^2k/40 - 121nk/40 \\ &\geq n^2\log n/40 - 121n^2/40 + 121n/40 + 121n(2^k - k - 1)/40 \\ &\geq (n^2\log n - 121n^2 + 121n)/40 . \end{aligned}$$

Thus the inductive hypothesis is established and we can conclude that the crossing number of $M_{2,n}$ is at least $\Omega(n^2\log n) = \Omega(N\log N)$ □

In section 7.4.3, we will show that the crossing number of any N-node graph with an $O(N^{1/2})$-separator is at most $O(N\log N)$. Thus, we will be able to conclude that the crossing number of the N-node 2-dimensional mesh of trees is precisely $\Theta(N\log N)$.

7.3.3 Lower Bounds for the r-Dimensional Mesh of Trees

By modifying the proof of Theorem 7-4, it can be shown that any layout of the r-dimensional mesh of trees must have very long wires. In particular, they must be as long as the width of any optimal layout for the graph. We state this result more precisely in the following theorem.

Theorem 7-5: *Any drawing of the N-node r-dimensional mesh of trees contains an edge which crosses at least $\Omega(N^{1-1/r})$ other edges.*

Proof: The r-dimensional $\overbrace{nxnx\cdots xn}^{r}$ mesh of trees $M_{r,n}$ has $N = (r+1)n^r - rn^{r-1} = \Theta(n^r)$ nodes for bounded r. We will show that any layout D of $M_{r,n}$ contains an edge which crosses at least $\Omega(n^{r-1}) = \Omega(N^{1-1/r})$ other edges, thus proving the theorem. The method used is very similar to that of Theorem 7-4.

As we did for the case of $r=2$ in Theorem 7-4, we first construct a drawing D' of the complete graph on the n^r leaves of $M_{r,n}$. Each type i edge of D is traced over at most $n^{r+1}2^{-i}$ times by this procedure. Thus the total number of crossings in D' is at most

$$(rn^{3r+1})/2 + n^{2r+2}\sum_{i=1}^{\log n} 2^{-2i}t_i$$

where, as before, $t_i=\sum_{j=1}^{\log n} t_{ij}$ and t_{ij} is the number of type i-j crossings in D. Applying Lemma 7-1, we can conclude that $\sum_{i=1}^{\log n} 2^{-2i} t_i \geq \Omega(n^{2r-2})$.

Let $s_k=\sum_{i=1}^{k} t_i$ be the total number of crossings of D involving an edge from the top k levels of the binary trees in $M_{r,n}$. Using arguments similar to those used to prove Theorem 7-4, it is not difficult to show that for large n, there exists a k such that $s_k \geq \Omega(n^{2r-2}2^k)$. As there are only $rn^{r-1}(2^{k+1}-2)$ edges in the top k levels of $M_{r,n}$ for any k, we can conclude that at least one of them crosses at least $\Omega(n^{r-1})$ other edges □

It is worth pointing out that the preceding arguments can also be used to show that the crossing number of the N-node r-dimensional mesh of trees is $\Theta(N^{2-2/r})$ for bounded $r>2$.

7.4 Further Methods

In this section, we describe some additional methods for proving crossing number bounds. We first generalize Lemma 7-1 to prove a combinatorial lower bound on the crossing number of any N-node graph with at least $4N$ edges. This result is then used in section 7.4.2 to prove crossing number lower bounds for a class of graphs which are similar to the 2-dimensional mesh of trees. We conclude by proving a nontrivial upper bound on the crossing number of graphs which have $O(N^{1/2})$-separators. As a corollary, we wiil show that any N-node graph with an $O(N^{1/2})$-separator can be embedded in some $O(N\log N)$-node planar graph, thus generalizing Theorem 6-1.

7.4.1 A Combinatorial Lower Bound for Crossing Numbers

In this section, we substantially generalize the result of Lemma 7-1. Throughout, we assume that G is a *simple* graph (i.e., that it has no loops or multiple edges).

Theorem 7-6: *If G is a graph with E edges and N nodes where $E\geq 4N$, then the crossing number of G is at least $E^3/375N^2$.*

Proof: The proof is by induction on N. For $N=1$, the result is vacuously true. Assume that the result is true for all $N'<N$ where $N>1$ and let G be a graph with N nodes and E edges where $E\geq 4N$. We will show that the crossing number c of G is at least $E^3/375N^2$, thus proving the theorem. There are two cases to

consider.

case 1: $4N \leq E < 5N$.

We first use Euler's formula [BLW76] in order to show that the genus of G is large. Euler's formula states that

$$E + 2 = N + f + 2g$$

where f is the number of faces of any proper embedding of G on a surface of genus g. Since G has no loops or multiple edges, every face contains at least 3 edges and thus $3f \leq 2E$. Substituting, we find that

$$\begin{aligned} 2g &= E + 2 - N - f \\ &\geq E + 2 - N - (2E/3) \\ &= E/3 + 2 - N \end{aligned}$$

and thus that $g > (E-3N)/6$. For $4N \leq E < 5N$, it is not difficult to show that $(E-3N)/6 \geq E^3/375N^2$ and thus that $g \geq E^3/375N^2$.

Given any graph with crossing number c, it is possible to find a proper embedding of the graph on a surface with genus c. We can do this by drawing the graph on a sphere so that only c pairs of edges cross and then putting a "handle" in the region immediately surrounding each crossing. The edges of the crossing can then be redrawn through the handle so that they no longer cross. As the resulting surface has genus c, we can conclude that $g \leq c$ for any graph with genus g and crossing number c. In particular, we can conclude that $c \geq E^3/375N^2$ for G.

case 2: $E \geq 5N$.

Let $d_1, \ldots, d_N$ be the degrees of the N nodes of G and let D be an optimal drawing of G. As usual, we can assume that no pair of edges which cross in D are incident to the same node of G. Consider the subdrawing D_i of D obtained by deleting the *ith* node of G and all the edges incident to it. This subdrawing is also a drawing of a graph with $N-1$ nodes and $E-d_i$ edges. Since $E \geq 5N$ and $d_i \leq N-1$, we can conclude that

$$E - d_i \geq 4N+1 > 4(N-1).$$

Thus we can apply the inductive hypothesis to D_i in order to conclude that it has at

least $(E-d_i)^3/[375(N-1)^2]$ crossings.

Each crossing of D will appear in precisely $N-4$ of the N subdrawings of D produced by the above procedure. Applying the technique used to prove Lemma 7-1, we can thus conclude that

$$\begin{aligned} c &\geq [1/(N-4)] \sum_{i=1}^{N} (E-d_i)^3/[375(N-1)^2] \\ &= [1/375(N-4)(N-1)^2] \sum_{i=1}^{N} (E^3 - 3E^2 d_i + 3E d_i^2 - d_i^3) \\ &= [1/375(N-4)(N-1)^2] \; [E^3 N - 3E^2(2E) + \sum_{i=1}^{N} (3E d_i^2 - d_i^3)] \; . \end{aligned}$$

Since $2E = \sum_{i=1}^{N} d_i$, it is not difficult to show that $\sum_{i=1}^{N} (3E d_i^2 - d_i^3)$ attains its minimal value when $d_i = 2E/N$ for $1 \leq i \leq N$. At this point,

$$\sum_{i=1}^{N} (3E d_i^2 - d_i^3) \geq 12E^3/N - 8E^3/N^2$$

and thus

$$c \geq (E^3 N - 6E^3 + 12E^3/N - 8E^3/N^2) / [375(N^3 - 6N^2 + 9N - 4)] \; .$$

For $N \geq 2$, this expression can easily be reduced to show that $c \geq E^3/375N^2$ □

It is interesting to note that the lower bound proved in Theorem 7-6 is (up to a constant) tight. For example, the N-node graph consisting of N^2/E disjoint copies of $K_{E/N}$ has $\Theta(E)$ edges and crossing number at most $O(E^3/N^2)$ for any $E \geq 4N$.

7.4.2 Applications

When defining the 2-dimensional mesh of trees, we required that the binary trees be interconnected so that $M_{2,n}$ contain 2^{2k} disjoint copies of $M_{2,n/2^k}$ as subgraphs for any k. Not only is this definition the most natural, but it also allows us to use induction in the lower bound proofs for the network. Surprisingly, however, the constraint is not necessary in order to show that $M_{2,n}$ can perform matrix-vector multiplication, sorting or switching in $O(\log n)$ time. In fact, any network consisting of n row trees and n column trees which share the same set of leaves can do these operations quickly. Thus it is conceivable that some other arrangement of the tree interconnections might lead to a network with a smaller crossing number. In what follows, we use Theorem 7-6 to show that this is not the case.

Theorem 7-7: *If G is an N-node graph formed in the same way as the nxn mesh of trees except that arbitrary interconnections are allowed between the leaves of the binary trees, then G must have crossing number at least* $\Omega(N\log N)$.

Proof: Let G_k denote the subgraph of G obtained by deleting the nodes and edges in the top k levels of the binary trees of G for $0 \leq k < \log n$. For example, if $G \simeq M_{2,n}$, then G_k consists of 2^{2k} disjointcopies of $M_{2,n2^{-k}}$. Otherwise, G_k is a graph for which each node of the original nxn matrix of nodes is a leaf of a horizontal complete binary tree of depth $\log n - k$ and a leaf of a vertical complete binary tree of depth $\log n - k$. For each k, let H_k denote the graph whose nodes are the n^2 leaves of G_k and whose edges are the paths in G_k of the form

leaf – path in horizontal binary tree – leaf – path in vertical binary tree – leaf.

Note that if $G \simeq M_{2,n}$, then H_k consists of 2^{2k} disjoint copies of $K_{n^2 2^{-2k}}$. In any case, H_k is a regular graph for which each node has degree $n^2 2^{-2k} - 1$.

Given any drawing D_k of G_k, it is easy to construct a drawing D_k' for H_k by tracing over the edges of G_k in the natural way. It is not difficult to see that each type i edge of G is traced over at most $(2^{\log n - k})3 2^{-(i-k)} = n^3 2^{-2k-i}$ times by this procedure for $i > k$. Thus each type i-j crossing is reproduced at most $n^6 2^{-4k-i-j} \leq n^6 2^{-4k-2i}$ times for $j \geq i > k$.

Given any drawing D of G, construct 2^{6k} separate drawings D_k' of H_k for each $k \geq 0$. Each type i-j crossing of D will appear a total of

$$\sum_{k=0}^{i-1} (n^6 2^{-4k-2i})(2^{6k}) = n^6 2^{-2i} \sum_{k=0}^{i-1} 2^{2k} \leq O(n^6)$$

times in these drawings. In what follows, we will show that there are at least $\Omega(n^8 \log n)$ total crossings of the first kind in these drawings. We will thus be able to conclude that the crossing number of G is at least $\Omega(n^2 \log n)$.

As H_k has $E_k = O(n^4 2^{-2k})$ edges and $N_k = n^2$ nodes, we can apply Theorem 7-6 to conclude that D_k' has at least $\Omega(E_k^3 / N_k^2) = \Omega(n^8 2^{-6k})$ crossings. Thus there are at least $\Omega(n^8)$ crossings among the 2^{6k} drawings D_k'. Summing over k for $0 \leq k \leq \log n$, we find that there are at least $\Omega(n^8 \log n)$ total crossings among all of the drawings $\{D_k' \mid 0 \leq k \leq \log n\}$. It is not difficult to check that there are at most $O(n^7 2^{-5k})$ crossings of the second kind in each drawing of H_k. As there are 2^{6k}

such drawings for each k, we can conclude that there are at most

$$\sum_{k=1}^{\log n}(n^7 2^{-5k})2^{6k} \leq O(n^8)$$

total crossings of the second kind in all the drawings $\{D_k' \mid 0 \leq k \leq \log n\}$. Thus there are at least $\Omega(n^8 \log n)$ total crossings of the first kind and the crossing number of G is at least $\Omega((n^8 \log n)/n^6) = \Omega(n^2 \log n) = \Omega(N \log N)$ □

As a corollary, we can see once again that the crossing number of $M_{2,n}$ is at least $\Omega(N \log^2 N)$.

7.4.3 An Upper Bound for Crossing Numbers

Since any N-node graph with an $O(N^\alpha)$-separator for some $\alpha > 1/2$ has an $O(N^{2\alpha})$-area layout, we can easily see that it also has crossing number at most $O(N^{2\alpha})$. By Theorem 7-1, we can conclude that this bound is tight since many such graphs also have bisection width at least $\Omega(N^\alpha)$.

The situation is not as clear for graphs with $O(N^{1/2})$-separators, however. For example, the best known upper bound on the layout area of an N-node graph with an $O(N^{1/2})$-separator is $O(N \log^2 N)$ yet no such graph is known to have a crossing number greater than $\Omega(N \log N)$. In what follows, we prove a tight upper bound on the crossing number of any such graph.

Theorem 7-8: *The crossing number of any N-node graph with an* $O(N^{1/2})$*-separator is at most* $O(N \log N)$.

Proof: Given such a graph G, we will construct a drawing for G with at most $O(N \log N)$ crossings. In order to construct the drawing, we will

1) decompose G into subgraphs according to the separator theorem,

2) draw the subgraphs by recursively calling the procedure, and

3) draw the edges which link the subgraphs together without introducing too many crossings and so that every node remains "close" to the exterior of the drawing.

In order to illustrate the procedure, we will describe in detail how drawings D_1 and D_2 of two m-node subgraphs are used to construct a drawing D of the combined $2m$-node subgraph. Let $c(m)$ denote number of crossings in D_1 or D_2, whichever is larger. Further let $d(m)$ denote the maximum number of edges which

must be crossed in order to draw an edge from any node in D_1 or D_2 to the exterior of D_1 and D_2. Construct D from the drawings of D_1 and D_2 by drawing in the $O(m^{1/2})$ edges which link them together in the best way possible. Now let $c(2m)$ and $d(2m)$ be the obvious values for the constructed drawing D. It is not difficult to show that

$$c(2m) \leq 2c(m) + O(m) + O(m^{1/2}d(m))$$

and that

$$d(2m) \leq d(m) + O(m^{1/2}).$$

Solving the recurrences in general, we find that $d(m) \leq O(m^{1/2})$ and thus that $c(m) \leq O(mlogm)$. Thus the above procedure can be used to find a drawing for G with at most $O(NlogN)$ crossings □

Using the preceding result, we can substantially generalize Theorem 6-1.

Theorem 7-9: *Any N-node graph with an* $O(N^{1/2})$*-separator can be embedded in an* $O(NlogN)$*-node planar graph.*

Proof: Construct a drawing of the graph with $O(NlogN)$ crossings according to the method described in the proof of Theorem 7-8. Replace each edge crossing in the drawing with an artificial node. The resulting graph has $O(NlogN)$ nodes, is planar and embeds the original graph □

CHAPTER 8

WIRE AREA ARGUMENTS

In this chapter, we extend the method of section 7.2 to prove lower bounds on the wire area of a variety of networks. In each proof, we will use a layout of a network to produce a layout for the complete graph. By showing that the nodes of the layout are widely spread out, we will be able to conclude that the wire area of the layout for the complete graph is very large. Provided that the edges of the original network were not traced over too many times, we can then reason that the wire area of the original network is also large.

8.1 Lower Bounds for the 2-Dimensional Mesh of Trees

In this section, we find tight lower bounds for the layout area and maximum edge length of the 2-dimensional mesh of trees.

Theorem 8-1: *The wire area of the N-node 2-dimensional mesh of trees is at least* $\Omega(N\log^2 N)$.

Proof: As usual, we denote the nxn mesh of trees by $M_{2,n}$. In addition, let $w(n)$ denote the wire area of $M_{2,n}$ and let α be a positive constant such that

(*) $\alpha \leq n/(4\log^2 n)$ for all $n \geq 2$, and

(**) $\alpha \leq 2^{2i-20}/(\beta^2 i^6)$ for all $i \geq 1$

where $\beta = \sum_{j=1}^{\infty} j^{-2}$, also a constant. Clearly such a constant exists ($\alpha = 2^{-30}$ should suffice) and clearly $w(n) \geq \alpha n^2 \log^2 n$ for $n=1$ and 2. Consider a value of $n \geq 4$ which is a power of 2 and assume that for all values of $m < n$ which are powers 2 that $w(m) \geq \alpha m^2 \log^2 m$. We will use induction to show that $w(n) \geq \alpha n^2 \log^2 n$. Since $M_{2,n}$ has $N = \Theta(n^2)$ nodes, this will be sufficient to prove the theorem.

Consider any layout for $M_{2,n}$ which uses $w(n)$ wire. Partition the layout into three vertical strips V_0, V_1 and V_2 so that the center strip contains $3n^2/4$ leaves and each outer strip contains $n^2/8$ leaves. Similarly partition the layout into three

horizontal strips H_0, H_1 and H_2 so that the middle strip contains $3n^2/4$ leaves and each outer strip contains $n^2/8$ leaves. For example, see Figure 8-1.

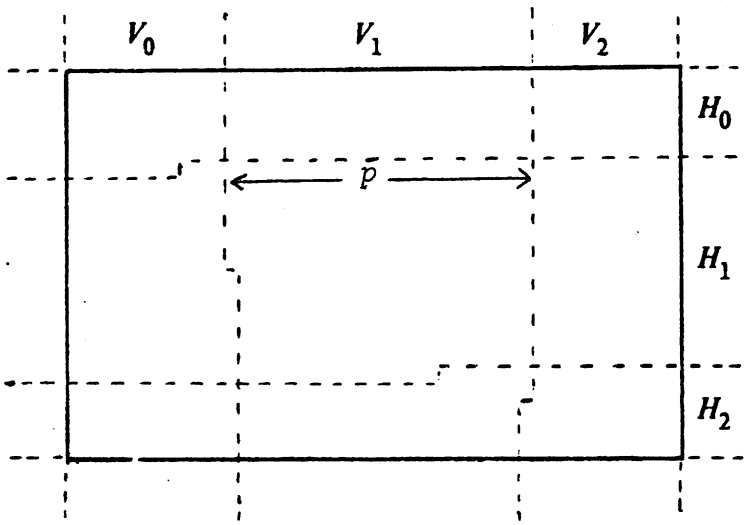

Figure 8-1: *Partitioning of the layout for* $M_{2,n}$.

Let p denote the length of the longest side of the center block formed by the intersection of V_1 and H_1. Without loss of generality, we assume that the longest side is horizontal. In what follows, we will show that $p \geq (\alpha^{1/2} n \log n)/8$.

Since each of the regions $V_0 \cap H_1$ and $V_2 \cap H_1$ can contain at most $n^2/8$ leaves, it is clear that $V_1 \cap H_1$ contains at least $n^2/2$ leaves. Consider the $n^{3/2}$ subgraphs of $M_{2,n}$ produced by eliminating the top $(3 \log n)/4$ levels of the row and column binary trees of $M_{2,n}$. Each of these subgraphs is isomorphic to $M_{2,n^{1/4}}$. By the pigeonhole principle, at least $1/2$ of these subgraphs have at least one leaf in $V_1 \cap H_1$. If $p < (\alpha^{1/2} n \log n)/8$ (otherwise we are done), then at most $4p < (\alpha^{1/2} n \log n)/2$ edges can cross the boundary of $V_1 \cap H_1$. Thus at most $(\alpha^{1/2} n \log n)/2$ of the subproblems which have at least one leaf in $V_1 \cap H_1$ can have some node or part of an edge outside $V_1 \cap H_1$. This means that at least $(n^{3/2} - \alpha^{1/2} n \log n)/2$ copies of $M_{2,n^{1/4}}$ are wholly contained in $V_1 \cap H_1$. Applying the inductive hypothesis, we conclude that $V_1 \cap H_1$ contains at least

$$(n^{3/2} - \alpha^{1/2} n \log n)\, w(n^{1/4})/2 \geq (\alpha n^2 \log^2 n - \alpha^{3/2} n^{3/2} \log^3 n)/32$$

$$\geq (\alpha n^2 \log^2 n)/64 \quad \text{wire.}$$

(The last inequality follows trivially from (*).) Thus $V_1 \cap H_1$ has at least $(\alpha n^2 \log^2 n)/64$ area and $p \geq (\alpha^{1/2} n \log n)/8$, as claimed.

We next use the layout for $M_{2,n}$ to construct a drawing for the complete graph on n^2 nodes (namely, the n^2 leaves of $M_{2,n}$). No matter how the edges of the complete graph are drawn in the plane (e.g., they may cross or overlap), it is clear from Figure 8-1 that the sum of the lengths of all the edges (as measured in Euclidean space) is at least $n^4 p/64 \geq (\alpha^{1/2} n^5 \log n)/2^9$. This is due to the fact that $n^4/64$ edges pass from region V_0 to region V_2 and that these regions are separated by a distance p.

Let L_i denote the sum of the lengths of the edges in the ith levels of the binary trees of $M_{2,n}$. Since every level i edge is traced over at most $n^3 2^{-i}$ times in the drawing of the complete graph, we can conclude that

$$\sum_{i=1}^{\log n} L_i 2^{-i} n^3 \geq (\alpha^{1/2} n^5 \log n)/2^9$$

and thus that

$$\sum_{i=1}^{\log n} L_i 2^{-i} \geq (\alpha^{1/2} n^2 \log n)/2^9.$$

In particular, this means that

$$L_i \geq (\alpha^{1/2} n^2 \log n 2^i)/(2^9 \beta i^2)$$

for some $i < \log n$. (Recall that $\beta = \sum_{j=1}^{\infty} j^{-2}$.) Otherwise,

$$L_i < (\alpha^{1/2} n^2 \log n 2^i)/(2^9 \beta i^2)$$

for $1 \leq i \leq \log n$ and thus

$$\sum_{i=1}^{\log n} L_i 2^{-i} < \sum_{i=1}^{\log n} (\alpha^{1/2} n^2 \log n)/(2^9 \beta i^2)$$
$$\leq (\alpha^{1/2} n^2 \log n)/2^9, \text{ a contradiction.}$$

Using the straightforward relation

$$w(n) \geq 2^{2i} w(n 2^{-i}) + L_i$$

where i has been chosen so that

$$L_i \geq (\alpha^{1/2} n^2 \log n 2^i)/(2^9 \beta i^2),$$

we can conclude that

$$w(n) \geq 2^{2i}\alpha(n2^{-i})^2(\log n - i)^2 + (\alpha^{1/2}n^2\log n 2^i)/(2^9\beta i^2)$$
$$\geq \alpha n^2\log^2 n - 2\alpha i n^2 \log n + (\alpha^{1/2}n^2\log n 2^i)/(2^9\beta i^2)$$
$$\geq \alpha n^2 \log^2 n \ .$$

(The last inequality follows trivially from (**).) Thus $w(n) \geq \Omega(n^2\log^2 n)$ for all n □

Theorem 8-2: *Any layout of the N-node 2-dimensional mesh of trees contains a wire of length at least* $\Omega(N^{1/2}\log N/\log\log N)$.

Proof: It is sufficient to show that any layout for $M_{2,n}$ contains a wire of length at least $\Omega(n\log n/\log\log n)$. Assume for the purposes of contradiction that this is not the case and consider a layout of $M_{2,n}$ for which the longest wire has length $q << O(n\log n/\log\log n)$. Using arguments similar to those used to prove Theorem 5-2, we first show that (without loss of generality) the area of such a layout is at most $O(q^2\log^2 n) << O(n^2\log^4 n)$.

Since every pair of nodes of $M_{2,n}$ is linked by a path of length at most $4\log n$, all of the nodes in the layout are contained in a $4q\log n \times 4q\log n$ square. At most $16q\log n$ wires may leave and re-enter the square at various points along its boundary. Without increasing the lengths of any of these wires, it is possible to rewire the segments outside the square using at most $O(q^2\log^2 n)$ additional area. Thus, the resulting layout for $M_{2,n}$ will have maximum edge length q and area at most $O(q^2\log^2 n)$.

The proof is completed by observing that any layout of $M_{2,n}$ with area less than $O(n^2\log^4 n)$ must have a wire of length at least $\Omega(n\log n/\log\log n)$. From the proof of Theorem 8-1, we know that $\sum_{i=1}^{\log n} L_i 2^{-i} \geq (\alpha^{1/2}n^2\log n)/2^9$. Thus either

1) there is an $i \leq 4\log\log n$ such that $L_i \geq (\alpha^{1/2}n^2\log n 2^i)/(2^{12}\log\log n)$, or

2) there is an $i > 4\log\log n$ such that $L_i \geq (\alpha^{1/2}n^2\log n 2^i)/(2^{10}\beta i^2)$

where, as before, the constant $\beta = \sum_{j=1}^{\infty} j^{-2}$. Otherwise,

$$\sum_{i=1}^{\log n} L_i 2^{-i} = \sum_{i=1}^{4\log\log n} L_i 2^{-i} + \sum_{i=4\log\log n+1}^{\log n} L_i 2^{-i}$$
$$< (\alpha^{1/2}n^2\log n)/2^{10} + [(\alpha^{1/2}n^2\log n)/2^{10}\beta] \sum_{i=4\log\log n+1}^{\log n} i^{-2}$$
$$\leq (\alpha^{1/2}n^2\log n)/2^9 \ , \quad \text{a contradiction.}$$

The second condition cannot possibly be true, however. If it were, the area of the layout would be at least

$$L_i \geq \Omega(n^2 \log n / i^2)$$

$$\geq \Omega(n^2 \log^5 n / (\log\log n)^2)$$

$$> \Omega(n^2 \log^4 n) \ , \quad \text{a contradiction.}$$

Thus the first condition must be true and there is an i such that $L_i \geq \Omega(n^2 \log n 2^i / \log\log n)$. Since there are $n2^{i+1}$ type i edges in $M_{2,n}$, we can conclude that at least one of them has length at least $\Omega(n \log n / \log\log n)$ □

8.2 Lower Bounds for the Tree of Meshes

Using the results of the previous section, it is easy to demonstrate the existence of planar graphs which *cannot* be laid out in linear area and which *must* have long wires. In particular, we can conclude the following.

Theorem 8-3: *The wire area of the N-node tree of meshes is at least* $\Omega(N \log N)$.

Proof: As we showed in section 6.3.3b, the N-node 2-dimensional mesh of trees can be embedded in an $O(N \log N)$-node tree of meshes. By Theorem 8-1, we can thus conclude that the wire area of the $N \log N$-node tree of meshes is at least $\Omega(N \log^2 N)$. Equivalently, the wire area of the N-node tree of meshes is at least $\Omega(N \log N)$. □

Theorem 8-4: *Any layout of the N-node augmented tree of meshes has a wire of length at least* $\Omega(N^{1/2} / \log^{1/2} N)$.

Proof: In the proof of Theorem 8-1, we showed that any layout of $M_{2,n}$ has two leaves which are spaced at least $\Omega(n \log n)$ distance apart. Since (as we showed in section 6.3.3b) $M_{2,n}$ can be embedded in T_{2n}' so that the leaves of $M_{2,n}$ are embedded in the leaves of T_{2n}, we can observe that any layout of T_{2n} also has two leaves which are spaced at least $\Omega(n \log n)$ distance apart. Since every pair of leaves in T_{2n} are linked by a path of length at most $O(\log n)$ in T_{2n}', we can conclude that some edge of T_{2n}' has length at least $\Omega(n) = \Omega(N^{1/2} / \log^{1/2} N)$ □

Had we so desired, we could have proved both results directly, using arguments identical to the ones used to prove Theorem 8-1.

8.3 Lower Bounds for a Restricted Class of Binary Tree Layouts

In [BK80], Brent and Kung considered layouts of N-node complete binary trees for which every leaf is located on the boundary of some convex region. In particular, they showed that the wire area of any such layout is at least $\Omega(NlogN)$. Recently, Patterson, Ruzzo and Snyder [PRS81] extended this result by showing that any such layout with area A must have some edge of length $\Omega(N/log(A/N))$. In particular, this means that if $A = O(NlogN)$, then there must be some edge of length $\Omega(N/loglogN)$ but that if $A = \Theta(N^{1+\varepsilon})$ for some $\varepsilon>0$, then there must only be an edge of length $\Omega(N/logN)$. In what follows, we show how to use the techniques developed in this chapter to give short proofs of these facts.

Theorem 8-5 (Brent and Kung [BK80]): *Any layout of the N-node complete binary tree in which every leaf is on the boundary of some convex region requires* $\Omega(NlogN)$ *area.*

Proof: Given any such layout, we first use the methods of section 8.1 to construct a layout of the complete graph on the $n=\Theta(N)$ leaves of the tree. Since the leaves are on the boundary of some convex region, it is easily shown that the layout of K_n uses at least $\Omega(n^3)$ wire.

Let L_i denote the sum of the lengths of the edges in the *ith* level of the tree. As each *ith* level edge is traced over at most $n^2 2^{-i}$ times, we know that

$$\sum_{i=1}^{\log n} n^3 2^{-i} L_i \leq \Omega(n^3)$$

and thus that $\sum_{i=1}^{\log n} L_i 2^{-i} \geq \Omega(n)$. Using arguments similar to those in the proof of Theorem 8-1, we can conclude that $L_i \geq \Omega(n2^i/i^2)$ for at least one value of i. Letting $w(n)$ denote the wire area of the binary tree layout, we can see that

$$w(n) \geq 2^i w(n2^{-i}) + \Omega(n2^i/i^2).$$

Solving the recurrence, we find that $w(n) \geq \Omega(nlogn) = \Omega(NlogN)$ □

Theorem 8-6 (Patterson, Ruzzo and Snyder [PRS81]): *Any A-area layout of the N-node complete binary tree in which every leaf is on the boundary of a convex region has some edge of length* $\Omega(N/log(A/N))$.

Proof: The proof follows that of the preceding theorem until it is concluded that $\sum_{i=1}^{\log n} L_i 2^{-i} \geq \Omega(n)$. Using methods similar to those used to prove Theorem 8-2, we

can then observe that one of the following conditions must be satisfied:

1) there is an $i \leq 2log(A/n)$ such that $L_i \geq \Omega(n2^i/log(A/n))$, or

2) there is an $i \geq 2log(A/n)$ such that $L_i \geq \Omega(n2^i/i^2)$.

The second condition cannot possibly hold since, if it did, the layout area would be at least $L_i \geq \Omega(n2^i/i^2)$ which, for $i \geq 2log(A/n)$, means that

$$A \geq \Omega(A^2/nlog^2(A/n))$$

$$> \Omega(A) \text{ , a contradiction.}$$

Thus the first condition holds and we can conclude that there is an i such that $L_i \geq \Omega(n2^i/log(A/n))$. As there are only 2^{i+1} edges in the *ith* level, at least one of them must have length at least $\Omega(n/log(A/n)) = \Omega(N/log(A/N))$ □

CONCLUSION INDEX REFERENCES

and ADDENDUM

This empty page was substituted for a blank page in the original document.

CONCLUSION

In Part I of the thesis, we described several new layouts for the shuffle-exchange graph. In particular, we found

1) an asymptotically optimal $O(N^2/log^2N)$-area layout of the N-node shuffle-exchange graph, and
2) practical layouts for small shuffle-exchange graphs.

As a result, it should now be possible to construct large scale shuffle-exchange chips. The only remaining question is whether or not there is a layout of the N-node shuffle-exchange graph for which every wire has length at most $O(N/log^2N)$. All known layouts have wires of length at least $\Omega(N/logN)$.

In Part II of the thesis, we descibed techniques for finding good lower bounds on the crossing number, wire area, maximum edge crossing and maximum edge length of a variety of VLSI networks. In particular, we applied these techniques to find

1) an N-node planar graph which has layout area $\Theta(NlogN)$ and maximum edge length $\Theta(N^{1/2}/log^{1/2}N)$,
2) an N-node graph with an $O(N^{1/2})$-separator which has layout area $\Theta(Nlog^2N)$ and maximum edge length $\Theta(N^{1/2}logN/loglogN)$, and
3) an N-node graph with an $O(N^{\alpha})$-separator (for $\alpha>1/2$) which has maximum edge length $\Theta(N^{\alpha})$.

Thus we have answered all the open questions concerning bounds for layout area and maximum edge length of networks with known separators. We have only partially answered the corresponding questions for planar graphs, however. In particular, it would be of great interest to know whether or not every N-node planar graph can be laid out in $O(NlogN)$ area.

INDEX

area of a layout *4*
artificial node *74*
augmented tree of meshes *72*

basic piece of a necklace *26*
basis node *15*
bisection width *52*

complex plane diagram *9*
crossing of the first kind *75*
crossing of the second kind *76*
crossing number *5*

degenerate necklace *10*
diameter *56*
distance in a graph *56*
distinguished node of a basic piece *26*
distinguished node of a necklace *21*
distinguished node of a primary piece *26*
distinguished node of a secondary piece *26*

even node *22*
exchange edge *3*

full necklace *10*

layout area *4*
left edge *65*
level *11*
leveling *18*
level-necklace grid *12*

maximum edge crossing *5*
maximum edge length *4*
mesh of trees *59, 63*
minimum number represented *18*

necklace *10*

odd node *22*

primary block of zeros *21*
primary node *22*
primary piece of a necklace *26*

radius of a necklace *18*
reverse edge *31*
right edge *65*

secondary block of zeros *21*
secondary node *22*
secondary piece of a necklace *26*
separator *51*
shift edge *30*
shuffle edge *3*
shuffle-exchange graph *3*
shuffle-shift graph *31*
shuffle-shift-reverse graph *31*
shuffle-tree graph *65*
simple graph *83*
simultaneous separator *67*
size of a necklace *15'*
size of a node *8*

Thompson model *2*
track *2*
transpose edge *32*
tree of meshes *65*
type *i* edge *80*
type *i-j* crossing *80*

value of a node *9*

wire area *5*

REFERENCES

[BK80] R. P. Brent and H. T. Kung, "On the area of binary tree layouts," *Information Processing Letters*, No. 11, 1980, pp. 44-46.

[BL81] S. N. Bhatt and C. E. Leiserson, "Minimizing the longest edge in a VLSI layout," Laboratory for Computer Science, Massachusetts Institute of Technology, preprint, 1981.

[BLW76] N. L. Biggs, E. K. Lloyd and R. J. Wilson, *Graph Theory 1736-1936*, Clarendon Press, Oxford, 1976.

[BO78] C. M. Bender and S. A. Orszag, *Advanced Mathematical Methods for Scientists and Engineers*, McGraw-Hill Book Company, New York, 1978.

[CM81]. B. Chazelle and L. Monier, "A model of computation for VLSI with related complexity results," *Proceedings of the 13th Annual ACM Symposium on Theory of Computation*, May 1981, pp. 318-325.

[GO81a] L. J. Guibas and A. M. Odlyzko, "Periods in strings," *Journal of Combinatorial Theory (Series A)*, Vol. 30, No. 1, January 1981, pp. 19-42.

[GO81b] L. J. Guibas and A. M. Odlyzko, "String overlaps, pattern matching and nontransitive games," *Journal of Combinatorial Theory (Section A)*, Vol. 30, 1981, pp. 183-208.

[HL80] D. Hoey and C. E. Leiserson, A layout for the shuffle-exchange network," *Proceedings of the 1980 IEEE International Conference on Parallel Processing*, August 1980.

[JR81] H. Jai-Wei and A. L. Rosenberg, "Graphs that are similar to binary trees," *Proccedings of the 13th Annual ACM Symposium on Theory of Computation*, May 1981, pp. 334-341.

[K70] D. J. Kleitman, "The crossing number of $K_{5,n}$," *Journal of Combinatorial Theory*, Vol. 9, No. 4, December 1970, pp. 315-323.

[KL78] H. T. Kung and C. E. Leiserson, "Algorithms for VLSI processor arrays," *Symposium on Sparse Matrix Computations*, Knoxville, Tennessee, November 1978.

[KLLM81] D. Kleitman, F. T. Leighton, M. Lepley and G. L. Miller, "New layouts for the shuffle-exchange graph," *Proceedings of the 13th Annual ACM Symposium on Theory of Computing*, May 1981, pp. 278-292.

[L75] D. Lawrie, "Access and alignment of data in an array processor," *IEEE Transactions on Computers*, Vol. C-24, December 1975, pp. 1145-1155.

[L76] T. Lang, "Interconnection between processing and memory modules using the shuffle-exchange network," *IEEE Transactions on Computers*, Vol. C-25, January 1976, pp. 55-66.

[L80a] C. E. Leiserson, "Area-efficient graph layouts (for VLSI)," *Proceedings of the 21st Annual IEEE Symposium on Foundations of Computer Science*, October 1980, pp. 270-281.

[L80b] C. E. Leiserson, *Area Efficient VLSI Computation*, Ph.D. dissertation, Department of Computer Science, Carnegie-Mellon University, November 1980.

[L81] F. T. Leighton, "New lower bound techniques for VLSI," *Proceedings of the 22nd Annual IEEE Symposium on Foundations of Computer Science*, October 1981, pp. 1-12.

[LLM81] F. T. Leighton, M. Lepley and G. L. Miller, "Layouts for the shuffle-exchange graph based on the complex plane diagram," unpublished notes, Applied Mathematics Department, Massachusetts Institute of Technology, Cambridge Massachusetts, August 1981.

[LM81] F. T. Leighton and G. L. Miller, "Optimal layouts for small shuffle-exchange graphs," *VLSI 81 - Very Large Scale Integration*, edited by John P. Gray, Academic Press, London, August 1981, pp. 289-299.

[LS81] R. J. Lipton and R. Sedgewick, "Lower bounds for VLSI," *Proceedings of the 13th Annual ACM Symposium on Theory of Computing*, May 1981, pp. 300-307.

[LT77] R. J. Lipton and R. E. Tarjan, "A separator theorem for planar graphs," *A Conference on Theoretical Computer Science*, University of Waterloo, August 1977.

[MC80] C. Mead and L. Conway, *Introduction to VLSI Systems*, Addison-Wesley Publishing Company, Reading, Massachusetts, October 1980.

[MP75] D. E. Muller and F. P. Preparata, "Bounds to complexities of networks for sorting and for switching," *Journal of the ACM*, Vol. 22, No. 2, April 1975, pp. 195-201.

[MR79] C. Mead and M. Rem, "Cost performance of VLSI computing structures," *IEEE Journal of Solid State Circuits*, Vol. SC-14, No. 2, April 1979, pp. 455-462.

[NS79] D. Nassimi and S. Sahni, "A self-routing Benes network and parallel permutation alogorithms," Technical report 79-13, University of Minnesota, May 1979.

[P80] D. S. Parker, "Notes on shuffle/exchange-type switching networks," *IEEE Transactions on Computers*, Vol. C-29, No. 3, March 1980, pp. 213-222.

[PRS81] M. S. Patterson, W. L. Ruzzo and L. Snyder, "Bounds on minimax edge length for complete binary trees," *Proceedings of the 13th Annual ACM Symposium on Theory of Computing*, May 1981, pp. 293-299.

[PV79] F. P. Preparata and J. E. Vuillemin, "The cube-connected-cycles: a versatile network for parallel computation," *Proceedings of the 20th Annual IEEE Symposium on Foundations of Computer Science*, October 1979, pp. 140-147.

[PV80] F. P. Preparata and J. E. Vuillemin, "Area-time optimal VLSI networks for matrix multiplication," *Proceedings of the 14th Princeton Conference on Information Science and Systems*, 1980.

[S71] H. S. Stone, "Parallel processing with the perfect shuffle," *IEEE Transactions on Computers*, Vol. C-20, No. 2, February 1971, pp. 153-161.

[S79] J. E. Savage, "Area-time tradeoffs for matrix multiplication and related problems in VLSI models," *Proceedings of the 17th Annual Allerton Conference on Communications, Control and Computing*, October 1979, pp. 670-676.

[S80] J. T. Schwartz, "Ultracomputers," *ACM Transactions on Programming Languages and Systems*, Vol. 2, No. 4, October 1980, pp. 484-521.

[SR80a] D. Steinberg and M. Rodeh, "Notes on the shuffle-exchange network," Technical report 083, IBM Israel Scientific Center, July 1980.

[SR80b] D. Steinberg and M. Rodeh, "A layout for the shuffle-exchange network with $\Theta(N^2/log^{3/2}N)$ area," Technical report 088, IBM Israel Scientific Center, September 1980.

[T79] C. D. Thompson, "Area-time complexity for VLSI," *Proceedings of the 11th Annual ACM Symposium on Theory of Computing*, May 1979, pp. 81-88.

[T80] C. D. Thompson, *A Complexity Theory for VLSI*, Ph.D. dissertation, Department of Computer Science, Carnegie-Mellon University, 1980.

[T81] C. D. Thompson, "The VLSI complexity of sorting," preprint, Computer Science Department, University of California at Berkely, 1981.

[TK77] C. D. Thompson and H. T. Kung, "Sorting on a mesh-connected parallel computer," *Communications of the ACM*, Vol. 20, 1977, pp. 263-271.

[V80] J. E. Vuillemin, "A combinatorial limit to the computing power of VLSI circuits," *Proceedings of the 21st Annual Symposium on Foundations of Computer Science*, IEEE Computer Society, November 1980, pp. 294-300.

ADDENDUM

Much has been accomplished during the period of time between the submission of this thesis to the MIT math department in August of 1981 and the publication of the thesis as a technical report in June of 1982. In fact, so much has been discovered in the interim that it would be possible to write several additional thesis on the subject. As an aide to those who wish to know more about the new material, we have included below a brief bibliography of some of the recent work on layout strategies for VLSI.

Of particular importance is the work contained in [V81], [CS81], [NMP81] and [L82]. In [V81], Valiant independently proves many of the separator-based results which are attributed to Leiserson in the thesis. The mesh of trees described in Chapter 6 of the thesis is independently discovered in [CS81] and [NMP81] where it is used to support a wide variety of fast parallel algorithms. Finally, the work reported in [L82] significantly extends the separator-based work of Leiserson and Valiant as well as the material in this thesis.

[BL82] S. N. Bhatt and C. E. Leiserson, "How to assemble tree machines," *Proceedings of the 14th Annual ACM Symposium on the Theory of Computing*, May 1982, pp. 77-84.

[BPP81] G. Bilardi, M. Pracchi and F. P. Preparata, "A critique and appraisal of VLSI models of computation," *Proceedings of the CMU Conference on VLSI Systems and Computations*, edited by H. T. Kung, B. Sproull and G. Steele, Computer Science Press, Rockville, Maryland, May 1981, pp. 81-88.

[CS81] P. R. Cappello and K. Steiglitz, "Area-efficient VLSI structures for multiplying at clock rate," Technical Report #289, Department of EECS, Princeton University, September 1981.

[GJ81] M. Garey and D. Johnson, "Crossing number is NP-complete," unpublished manuscript, December 1981.

[L82] F. T. Leighton, "A layout strategy for VLSI which is provably good," *Proceedings of the 14th Annual ACM Symposium on the Theory of Computing*, May 1982, pp. 85-98.

[LL82] F. T. Leighton and C. E. Leiserson, "Probabilistic algorithms for constructing networks with short wires," unpublished manuscript, April 1982.

[LR82] F. T. Leighton and A. L. Rosenberg, "Three-dimensional circuit layouts," unpublished manuscript, April 1982.

[NMP81] D. Nath, S. N. Maheshwari and P. C. P. Bhatt, "Efficient VLSI networks for parallel processing based on orthogonal trees," unpublished manuscript, 1981.

[P81] F. Preparata, "Optimal three-dimensional VLSI layouts," unpublished manuscript, 1981.

[R81] A. L. Rosenberg, "Routing with permuters: toward reconfigurable and fault-tolerant networks," Technical Report CS-1981-13, Duke University, 1981.

[S81] L. Snyder, "Overview of the CHiP computer," *VLSI 81 - Very Large Scale Integration*, edited by J. Gray, Academic Press, London, August 1981, pp. 237-246.

[V81] L. G. Valiant, "Universality considerations in VLSI circuits," *IEEE Transactions on Computers*, Vol. V-30, No. 2, February 1981, pp. 135-140.

This blank page was inserted to preserve pagination.

CS-TR Scanning Project
Document Control Form

Date : 8/18/95

Report # LCS-TR-274

Each of the following should be identified by a checkmark:
Originating Department:

☐ Artificial Intellegence Laboratory (AI)
☒ Laboratory for Computer Science (LCS)

Document Type:

☒ Technical Report (TR) ☐ Technical Memo (TM)
☐ Other:_____

Document Information

Number of pages: 117 (122-IMAGES)
Not to include DOD forms, printer instructions, etc... original pages only.

Originals are:
☒ Single-sided or
☐ Double-sided

Intended to be printed as :
☐ Single-sided or
☒ Double-sided

Print type:
☐ Typewriter ☐ Offset Press ☐ Laser Print
☐ InkJet Printer ☒ Unknown ☐ Other:_____

Check each if included with document:

☐ DOD Form ☐ Funding Agent Form ☐ Cover Page
☐ Spine ☒ Printers Notes ☐ Photo negatives
☐ Other: _____

Page Data:

Blank Pages(by page number):_____

Photographs/Tonal Material (by page number):_____

Other (note description/page number):

Description : Page Number:

IMAGE MAP: (1-10) UNN'D TITLE & BLANK PAGES, i-v, UNN BLANK, [redacted] PAGE, BLANK
(11-117) PAGES #'D 2-6, UNN BLANK, UNN [redacted] PAGE, 8-49, UNN BLANK,
UNN'D [redacted] PAGE, PAGE, 51-95, UNN PAGE, UNN BLANK, 97-105
(118-122) SCANCONTROL, PRINTER'S NOTES, TARGETS (3)

Scanning Agent Signoff:
Date Received: 8/18/95 Date Scanned: 9/7/95 Date Returned: 9/15/95

Scanning Agent Signature: Michael W. Cook

Scanning Agent Identification Target

Scanning of this document was supported in part by the **Corporation for National Research Initiatives,** using funds from the **Advanced Research Projects Agency** of the **United states Government** under Grant: **MDA972-92-J1029.**

The scanning agent for this project was the **Document Services** department of the **M.I.T Libraries.** Technical support for this project was also provided by the **M.I.T. Laboratory for Computer Sciences.**

Scanned

Date: 9/7/1995

M.I.T. Libraries
Document Services