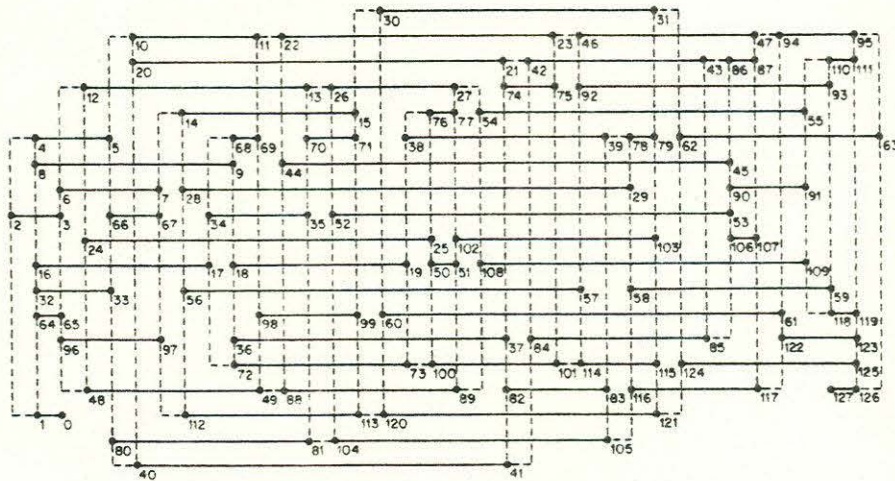# AN ASYMPTOTICALLY OPTIMAL
LAYOUT FOR
THE SHUFFLE - EXCHANGE GRAPH

Daniel Kleitman
Frank Thomson Leighton
Margaret Lepley
Gary L. Miller

October 1982

AN ASYMPTOTICALLY OPTIMAL LAYOUT

FOR THE SHUFFLE - EXCHANGE GRAPH

Daniel Kleitman

Frank Thomson Leighton

Margaret Lepley

Gary L. Miller

Mathematics Department

and

Laboratory for Computer Science

Massachusetts Institute of Technology

Cambridge, Massachusetts 02139

Abstract: The shuffle-exchange graph is one of the best structures known for parallel computation. Among other things, a shuffle-exchange computer can be used to compute discrete Fourier transforms, multiply matrices, evaluate polynomials, perform permutations and sort lists. The algorithms needed for these operations are quite simple and many require no more than logarithmic time and constant space per processor. In this paper, we describe an $O(N^2/log^2 N)$-area layout for the shuffle-exchange graph on a two-dimensional grid. The layout is the first which is known to achieve Thompson's asymptotic lower bound.

Key words: area-efficient chip layouts, cube-connected-cycles graph, graph embedding, shuffle-exchange graph, Thompson grid model, Very Large Scale Integration (VLSI)

Cover design: A 19 x 36 Thompson model layout for the 128-node shuffle-exchange graph.

1. Introduction

The shuffle-exchange graph has long been recognized as one of the best structures known for parallel computation. Among its many applications, a shuffle-exchange computer can be used to compute discrete Fourier transforms, multiply matrices, evaluate polynomials, perform permutations and sort lists [S71, P80, S80]. The algorithms needed for these operations are very simple and many require no more than logarithmic time and constant space per processor.

Recent developments in Very Large Scale Integration (VLSI) circuit technology have made it possible to fabricate large numbers of simple processors on a single chip. As most of the processors contained in a shuffle-exchange computer are simple, the shuffle-exchange graph serves as an excellent basis upon which to design and build chip-sized microcomputers. One of the main difficulties with such an architecture, however, is the problem of routing the wires which link the processors together in a shuffle-exchange network. Current fabrication technology limits the designer to two or three layers of insulated wiring on a chip and demands that he make the chip as small in area as possible.

Abstracted, the designer's problem becomes the mathematical question of how to embed the shuffle-exchange graph in the smallest possible two-dimensional grid. Thompson was the first to formalize the question mathematically. In his thesis [T80], he showed that any *layout* (i.e., embedding in a two-dimensional grid) of the N-node shuffle-exchange graph requires at least $\Omega(N^2/log^2 N)$ area. In addition, he described a layout requiring only $O(N^2/log^{1/2} N)$ area. Shortly thereafter, Hoey and Leiserson [HL80] described an embedding for the shuffle-exchange graph in the complex plane (called the *complex plane diagram*) and showed how the diagram could be used to find an $O(N^2/logN)$-area layout for the N-node shuffle-exchange graph. Subsequently, Leighton, Lepley and Miller [LLM82], and (independently) Steinberg and Rodeh [SR81] showed how the complex plane diagram could be used to find an $O(N^2/log^{3/2} N)$-area layout.

In this paper, we pursue an entirely different strategy in order to find an $O(N^2/log^2 N)$-area layout for the N-node shuffle-exchange graph, thus acheiving Thompson's asymptotic lower bound. In addition, we will describe how the new techniques can be used to find $O(N^2/log^2 N)$-area layouts for more general graphs (such as the shuffle-shift-reverse graph).

1

As is the case with much of the previous work on this problem, our results are more theoretical than practical. This is due, in part, to the fact that the layout procedure described in the paper is designed to produce good layouts for *large* shuffle-exchange graphs. Unfortunately, it produces poor layouts for *small* shuffle-exchange graphs and, for the time being, these are the networks of practical interest. Nevertheless, the *methods* developed in the paper do appear to have some practical value. For example, Leighton and Miller [LM81] have constructed a 19-by-36 layout for the 128-node shuffle-exchange graph by extending the methods described in this paper.

The remainder of the paper is divided into nine sections. In section 2, we define the shuffle-exchange graph and the grid model of a chip. Section 3 contains more definitions and some useful combinatorial lemmas. The proofs of these lemmas are included in Section 4. In Section 5, we describe a near-optimal, $O(N^2 loglog^2 N/log^2 N)$-area layout for the N-node shuffle-exchange graph. In Section 6, we show how to modify the near-optimal layout in order to produce an asymptotically optimal layout. In Section 7, we show how to lay out several supergraphs of the shuffle-exchange graph. We conclude in Sections 8-10 with some remarks, acknowledgements and references.

2. Preliminaries

2a) The shuffle-exchange graph

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 $w' = a_0 a_{k-1} \cdots a_1$, 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 1. Note that the shuffle edges are drawn with solid lines while the exchange edges are drawn with dashed lines.
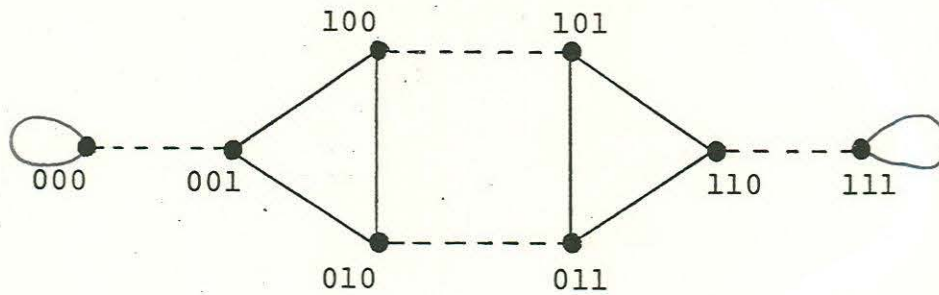
Figure 1: *The 8-node shuffle-exchange graph.*

By replacing the nodes and edges of the shuffle-exchange graph by processors and wires (respectively), the shuffle-exchange graph can be transformed into a very powerful parallel computer (which we call the *shuffle-exchange computer*). The computational power of the shuffle-exchange computer is partly derived from the fact that every pair of nodes in an N-node shuffle-exchange graph is linked by a path containing at most $2logN$ edges and thus the communication time between any pair of processors is short.

More importantly, however, the shuffle-exchange computer is capable of performing a perfect shuffle on a set of data in a single parallel operation. For example, consider a deck of 8 cards distributed among the 8 processors of the 8-node shuffle-exchange graph so that processor 000 initially has card 0, processor 001 initially has card 1, processor 010 initially has card 2, and so forth. Next, consider a (parallel) operation of the shuffle-exchange computer in which each processor $a_2a_1a_0$ sends its card across a shuffle edge to the neighboring processor $a_1a_0a_2$. It is easily verified that, after completion of the operation, processor 000 contains card 0 (the top card in the shuffled deck), processor 001 contains card 4 (the second card in the shuffled deck), and so forth.

The power of card shuffling and its mathematical abstractions is well known to magicians and mathematicians [DGK81] as well as to computer scientists [S71, S80]. For a good survey of the computational power of the shuffle-exchange graph, we recommend Schwartz' paper on ultracomputers [S80]. In addition, Stone's paper [S71] contains a nice description of some important parallel algorithms based on the shuffle-exchange graph.

2b) Necklaces

As can be seen in Figure 1, there is a natural partitioning of the shuffle edges into cycles. These cycles are called *necklaces*. A necklace is simply the collection of all cyclic shifts of some node of the graph. In particular, the necklace which contains the node *w* is called the *necklace generated by w* and is denoted by ⟨*w*⟩. For example, the necklace generated by *001* is ⟨*001*⟩ = {*001, 010, 100*}.

If a necklace contains precisely *k* nodes, it is said to be *full*. Otherwise the necklace contains less than *k* nodes and is said to be *degenerate*. As Leighton, Lepley and Miller show in [LLM82], the number of degenerate necklaces is quite small compared to the number of full necklaces. In particular, they prove the following.

Lemma 1: *At most* $O(N^{1/2})$ *nodes are contained in the degenerate necklaces of an N-node shuffle-exchange graph. The remaining nodes are contained in* $N/logN$ - $O(N^{1/2}/logN)$ *full necklaces.*

2c) The grid model

Among the many mathematical models that have been proposed for VLSI computation, the most widely accepted is due to Thompson and is known as the *Thompson grid model* [T79, T80]. The grid 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 (without making an electrical connection), pairs of wires are not allowed to overlap for any distance or to overlap at corners (i.e., they cannot overlap in the same track). Further, wires are not allowed to overlap processors to which they are not linked. (The routing of wires in this fashion is also known as *layer per direction routing* and *Manhattan routing*.)

As an example, we have included a grid layout for the 8-node shuffle-exchange graph in Figure 2. As before, the shuffle edges are drawn with solid lines while the exchange edges are drawn with dashed lines. Notice that we have omitted the self-loops in Figure 2 since they are electrically redundant. In general, the embedding will not be planar (as it is in this example).
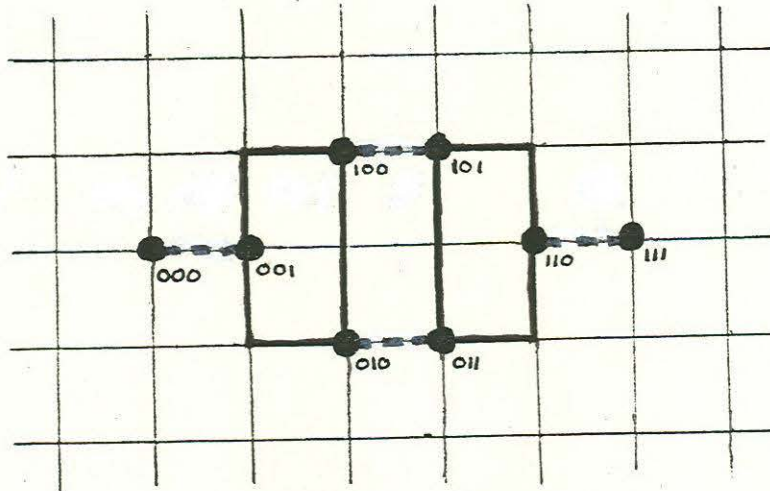
4

Figure 2: *A grid model layout of the 8-node shuffle-exchange graph.*

Practical considerations dictate that the area of a VLSI layout be as small as possible. The *area of a layout* in the grid model is defined to be the product of the number of horizontal tracks and the number of vertical tracks which contain a processor or wire segment of the layout. For example, the layout in Figure 2 has area 18. As Leighton and Miller have shown in [LM82], this layout is suboptimal.

Leiserson observed in [L80a] that any M wires can be added to a layout by inserting at most $2M$ vertical and $2M$ horizontal tracks. Hence M wires can be added to a $\Omega(M)$-by-$\Omega(M)$ layout without increasing the area by more than a constant factor. As any layout for the N-node shuffle-exchange graph must have $\Omega(N/logN)$ vertical and $\Omega(N/logN)$ horizontal tracks, the preceding observation means that a nearly complete layout for the shuffle-exchange graph with area A can be extended to a complete layout with area $O(A)$. This result will be used at several points in the paper and is stated formally in the following lemma.

Lemma 2: *Any area A layout which contains all but O(N/logN) nodes and edges of the N-node shuffle-exchange graph can be extended to form a complete layout for the N-node shuffle-exchange graph with area O(A).*

As an immediate application of Lemma 2, we can henceforth ignore nodes which are contained in degenerate necklaces. This is because at most $O(N^{1/2})$ nodes are contained in degenerate necklaces (Lemma 1) and thus they can be inserted into any layout of full necklaces without increasing the total area by more than a constant factor (Lemma 2).

3. Some Combinatorial Lemmas

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 $\Psi_k(t)$. The proof of the lemma (as well as of Lemmas 4-6) is included in Section 4.

Lemma 3: *For* $(logk)/2 + loglnk \leq t \ll k$ *and* $k \to \infty$,

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

In order to illustrate the important features of the function in Lemma 3, we have sketched a graph of $2^{-k}\Psi_k(t)$ versus t in Figure 3. The maximum of $2^{-k}\Psi_k(t)$ occurs at $t = logk-1$ where $2^{-k}\Psi_k(t) \sim (e^{1/2} - 1)/e \approx .23865$. For $t > logk - 1$, $2^{-k}\Psi_k(t)$ decreases *exponentially* as t increases. For $t \leq logk - 1$, $2^{-k}\Psi_k(t)$ decreases *doubly exponentially* as t decreases.
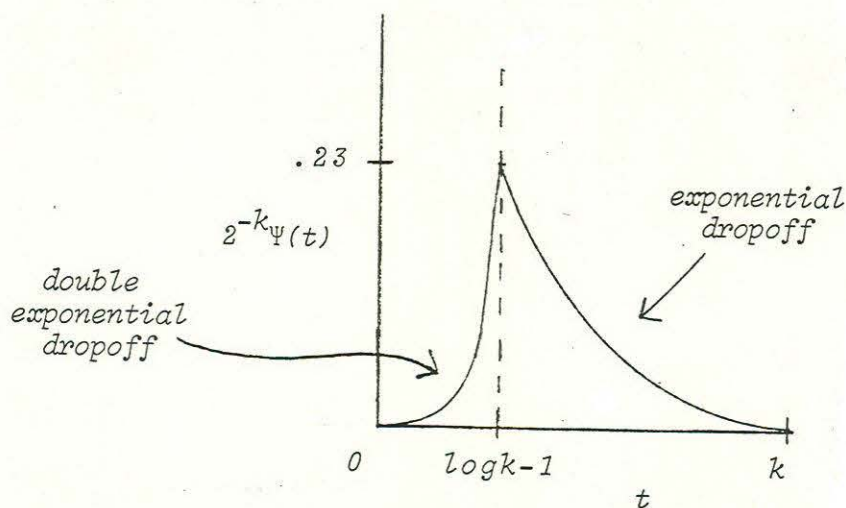

Figure 3: *Density of k-bit binary strings for which the longest block of consecutive zeros has length t.*

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

Lemma 4: *The number of k-bit strings for which the longest block of consecutive zeros has length less than $logk - loglnk - 1$ or length greater than $2logk$ is at most $O(2^k/k) = O(N/logN)$.*

By Lemma 2, we may ignore $O(N/logN)$-sized sets of nodes which have undesirable properties. As such nodes can be inserted with the addition of at most $O(N/logN)$ vertical and horizontal tracks, we can always add them later without increasing the total area by more than a constant factor. By Lemma 4, 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 5: *The sum over all necklaces of the difference in length between the longest and second longest blocks of consecutive zeros is at most $O(N/logN)$.*

Using information about the size and location of blocks of zeros within the necklace, it is possible to distinguish one particular node in most necklaces. 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 6: *At most $O(N/logN)$ 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 referred 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 Lemmas 2 and 4, 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, $001\overline{0}1$ 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.

4. Proofs of Lemmas 3 - 6

We now present the proofs of Lemmas 3 through 6. Such results can also be found in the recent work of Guibas and Odlyzko [GO81a,GO81b]. As the proofs are fairly technical, many readers may wish to skip this section and proceed directly to Section 5.

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 $v_t = \overbrace{10\cdots 0}^{t}$. The proofs of Lemmas 3 through 6 depend heavily on the following combinatorial result.

Theorem 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

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

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

$$\approx\; [1/(1-F(x))] - 1$$

and thus that

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

9

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

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

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

for $1 \leq i \leq t$. 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 $|r| \leq c < 1/2$ 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 $|q(t)| \ll 2^{-t}$ as $t \to \infty$. Another iteration of this process reveals that $q(t) = O(t2^{-2t})$ and thus that

10

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

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

$$|2r| = 2 - 2\varepsilon(t)$$

$$> 1 + |1 - \varepsilon(t)|^t$$

$$= 1 + |r^t|$$

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 $\overbrace{0\cdots0}^{i}w\overbrace{10\cdots0}^{t-i}$ 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 $\square$

We can now prove Lemmas 3 and 4.

Proof of Lemma 3: 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 $t \geq (\log k)/2 + \log\log k$, both $t2^{-t}$ and $kt2^{-2t}$ vanish as $k \to \infty$. In what follows, we will show that if $t \ll k$, then

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

and thus that

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

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

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

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

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

provided that $t \ll k$, a contradiction $\square$

Proof of Lemma 4: The number of k-bit strings which do not contain a block of $\log k - \log\ln k - 1$ consecutive zeros is

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

$$= 2^k / k$$

$$= O(N/\log N) .$$

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

12

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

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

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

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

The proofs of Lemmas 5 and 6 depend on the following corollary to Theorem 1.

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

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

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

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

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

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

and thus that

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

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

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

For larger values of t,

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

and thus

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

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

13

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

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

Proof of Lemma 5: A string whose longest block of zeros has length t and whose second longest block of zeros has length $s \le t$ is of the form $w1\overbrace{0\cdots0}^{t+1}w'$, 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

$$\le (1/k) \sum_{t=0}^{k} \sum_{s=0}^{t} (t-s)\, k\, \Psi_{k-t-1}(s)$$

$$= \sum_{t=0}^{k} \sum_{s=0}^{t} (t-s)\, [\overline{\Psi}_{k-t-1}(s+2) - \overline{\Psi}_{k-t-1}(s+1)]$$

$$= \sum_{s=1}^{k} \sum_{t=s}^{k} \overline{\Psi}_{k-t}(s)$$

$$= \sum_{s=1}^{k} \left(2^k e^{-k2^{-s}} e^{O(s2^{-s},\, ks2^{-2s})} \sum_{t=s}^{k} 2^{-t} e^{t2^{-s}} \right)$$

$$\le \sum_{s=1}^{k} \left(2^k e^{-k2^{-s}} e^{O(s2^{-s},\, ks2^{-2s})} \, 2^{-s} e^{O(s2^{-s})} \right)$$

$$= \sum_{s=1}^{k} 2^{k-s} e^{-k2^{-s}} e^{O(s2^{-s},\, ks2^{-2s})}$$

$$\le \sum_{s=1}^{k} \overline{\Psi}_{k-s}(s)$$

$$= O(N/logN)$$

by Corollary 1 □

Proof of Lemma 6: 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) $w_1\overbrace{0\cdots0}^{\substack{k/2 \\ t}}w_2\underbrace{0\cdots0}_{t}w_3,$

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

14

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

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 1, we can thus conclude that there are at most $O(N/logN)$ such nodes altogether □

5. A Near-Optimal Layout

We are now prepared to describe a near-optimal, preliminary version of the optimal layout. In Section 6, we will show how to modify this layout in order to construct an optimal $O(N^2/log^2N)$-area layout for the N-node shuffle-exchange graph.

5a) Location of the nodes

The near-optimal layout is constructed from a $logN$ x $O(N/logN)$ 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. (The *value* of a node is simply the numerical value of the associated k-bit binary string.) For example, we have constructed such a grid for the *32*-node shuffle-exchange graph in Figure 4. 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.
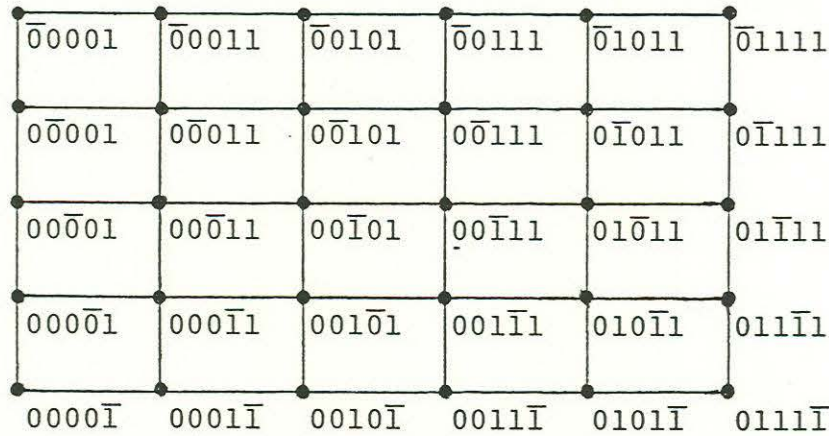
Figure 4: *The grid of nodes for the 32-node shuffle-exchange graph.*

Grid labels (top four rows with labels above, bottom row with labels below):

$\overline{0}0001$	$\overline{0}0011$	$\overline{0}0101$	$\overline{0}0111$	$\overline{0}1011$	$\overline{0}1111$
$0\overline{0}001$	$0\overline{0}011$	$0\overline{0}101$	$0\overline{0}111$	$0\overline{1}011$	$0\overline{1}111$
$00\overline{0}01$	$00\overline{0}11$	$00\overline{1}01$	$00\overline{1}11$	$01\overline{0}11$	$01\overline{1}11$
$000\overline{0}1$	$000\overline{1}1$	$001\overline{0}1$	$001\overline{1}1$	$010\overline{1}1$	$011\overline{1}1$
$0000\overline{1}$	$0001\overline{1}$	$0010\overline{1}$	$0011\overline{1}$	$0101\overline{1}$	$0111\overline{1}$

5b) 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 $O(N^2 loglog^2 N/log^2 N)$. This is only a factor of $O(loglog^2 N)$ off from the lower bound of $O(N^2/log^2 N)$.

The analysis is divided into two parts. First we show that only $O(NloglogN/logN)$ exchange edges link nodes which are in *different* rows of the grid. Such edges can be inserted with the addition of at most $O(NloglogN/logN)$ vertical and horizontal tracks. We then 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.

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 $w\overline{0}w'$ where $|w| = i\text{-}1$ and $w0w'$ is the distinguished node of $\langle w0w'\rangle$. The exchange edge is also incident to the odd node $w\overline{1}w'$. By assumption, $w\overline{1}w'$ 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 $w\overline{1}w'$) must be in the primary or secondary block of zeros of $w0w'$.

16

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 $w\overline{0}w'$ must be a primary or secondary node. As was previously mentioned, we can assume that each necklace has at most $2logk = 2loglogN$ primary and $2loglogN$ secondary nodes. Thus at most $4loglogN$ 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 $O(N/logN)$ necklaces, we can conclude that there are at most $O(NloglogN/logN)$ exchange edges which link nodes in different rows.

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 $O(N/logN)$ horizontal tracks. Once again, the analysis is divided into two parts. In the first part, we show that at most $O(N/logN)$ exchange edges are contained in the first $logk$ rows. Such edges can be trivially inserted with the addition of $O(N/logN)$ 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 $i > logk$. Since $\sum_{t=logk+1}^{K} 2^{k-i} \leq 2^k/k = N/logN$, this will be sufficient to show that at most $O(N/logN)$ additional horizontal tracks are necessary to insert the remaining exchange edges.

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

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

such necklaces for $(logk)/2 + loglnk \leq t \ll k$. By Lemma 4, 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

$$\sim \sum_{t=logk-loglnk-1}^{log\,k} (logk - t)(2^k/k)(e^{-k2^{-t-2}} - e^{-k2^{-t-1}})$$

$$= (2^k/k)\sum_{r=0}^{loglnk+1} r(e^{-k2^{r-2-logk}} - e^{-k2^{r-1-logk}})$$

$$= (2^k/k)\sum_{r=0}^{loglnk+1} r(e^{-2^{r-2}} - e^{-2^{r-1}})$$

$$= (2^k/k) \sum_{r=0}^{\log \ln k + 1} e^{-2^{r-2}}$$

$$\leq (2^k/k) \sum_{r=0}^{\infty} e^{-2^{r-2}}$$

$$= O(N/\log N).$$

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

We next consider the number of horizontal tracks necessary to insert the exhange edges contained in the ith row for $i > \log k$. 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 5, we illustrate the necessary conditions for two exchange edges to overlap in the ith row. All representations are in terms of distinguished nodes.
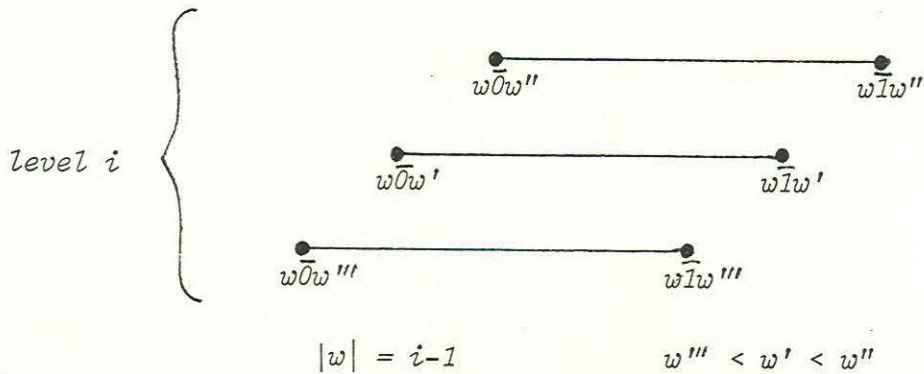
Figure 5: *Necessary conditions for exchange edges to overlap in the ith row.*

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 $O(N^2(\log\log N)^2/\log^2 N)$ area.

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

In this section, we will modify the layout described in Section 5 in order to produce an optimal $O(N^2/log^2 N)$-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.

6a) 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 $\{0\bar{1}011, 010\bar{1}1, 0101\bar{1}\}$, the primary piece is $\{\bar{0}1011\}$, and the secondary piece is $\{01\bar{0}11\}$.

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 $\langle 01111 \rangle$) 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 5 and 6 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.

6b) Location of the Nodes

As in Section 5, the layout is constructed from a $logN \times O(N/logN)$ 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 $a_{k-1} \cdots \bar{a}_{k-i} \cdots a_0$ (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 5. 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 6.
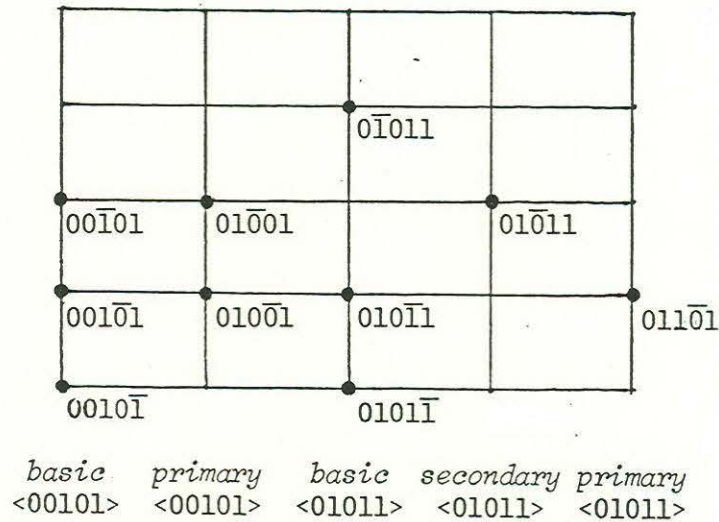

basic   primary   basic   secondary primary
<00101> <00101> <01011> <01011> <01011>

Figure 6: *Relocated nodes for the 32-node shuffle-exchange graph.*

Note that the necklaces *<00001>*, *<00011>*, *<00111>*, and *<01111>* have not been included in Figure 6 since their associated primary pieces do not have distinguished nodes.

6c) 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 $O(N/logN)$

vertical and horizontal tracks. In what follows, we will show that at most $O(N/logN)$ vertical and horizontal tracks are needed to insert all of the exchange edges as well. Thus the area of the layout will be $O(N^2/log^2N)$, which is optimal.

As before, we divide the analysis of the exchange edges into two parts. We first show that at most $O(N/logN)$ 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 $O(N/logN)$ 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 $O(N/logN)$ horizontal tracks. The arguments will be very similar to those in Section 5b.

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 5b 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 $|w| = i\text{-}1$ for some i. By definition, $w\bar{0}w'$ is located in the *ith* 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 *ith* 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 *ith* 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 $w1\overbrace{0\cdots0}^{i_1}\overbrace{00\cdots01}^{i_2}w'$ 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

$$\langle w1\overbrace{0\cdots01}^{t_1}\overbrace{0\cdots01}^{t_2}w'\rangle \text{ and thus } w1\overbrace{0\cdots0\bar{1}}^{t_1}\overbrace{0\cdots01}^{t_2}w'$$

would be on the same

row as $w1\overbrace{0\cdots0\bar{0}}^{t_1}\overbrace{0\cdots01}^{t_2}w'$, 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 5, 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.

Using the analysis developed in Section 5b, 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 5b, 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.

7. 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 might desire a layout for the shuffle-exchange graph which also includes shift, reverse and/or 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.

7a) 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 6, it is not difficult to show that the *N*-node shuffle-shift 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 6, 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.

7b) 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 6, it is also possible to show that the N-node shuffle-shift-reverse graph can be laid out in $O(N^2/log^2 N)$ area. The basic idea is to modify the layout described in Section 7a 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 7a. Thus all of the edges can be inserted using at most $O(N/logN)$ vertical and horizontal tracks.

7c) 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 $O(N^2/log^2 N)$ area, the least possible.

24

8. Remarks

For some applications it is useful to consider optimizing measures other than area. For example, we might wish to minimize the number of wire crossings in the layout, the length of the longest wire in the layout, and/or the sum of the lengths of all the wires in the layout [L82a]. In [L81], Leighton shows that any layout for the N-node shuffle-exchange graph must have $\Omega(N^2/log^2N)$ wire crossings. Since a layout with area A can have at most A wire crossings, the wire crossing lower bound is achieved by the layout described in this paper.

As a consequence of the wire crossing lower bound, it is also shown in [L81] that any layout for the N-node shuffle-exchange graph must have an edge of length $\Omega(N/log^2N)$ and total wire length $\Omega(N^2/log^2N)$. The layout described in this paper clearly acheives the latter bound. It does not acheive the maximum wire length lower bound, however. In fact, we do not know of any layout for the N-node shuffle-exchange graph for which every wire has length $o(N/logN)$. The layout described in this paper has wires of length $\Theta(N/logN)$.

The methods developed in this paper 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, the 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 the primary or secondary block of zeros) should we be able to globally change the distinguished node.

One such method of distinguishing a node is to select that node in the necklace which has the minimal value. Although the proof is fairly 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.

As it previously was not known whether or not the N-node shuffle-exchange graph could be laid out in $O(N^2/log^2N)$ area, several researchers have tried to develop alternate networks which can compute discrete Fourier transforms in $O(logN)$ steps and which can be laid out in $O(N^2/log^2N)$ area. The only other network discovered which has these properties is the *cube-connected-cycles graph* of Preparata and Vuillemin [PV79]. At this point, it is not clear which network best serves as the basis for practical parallel computation. Whereas the shuffle-

exchange graph appears to be simpler to program than the cube-connected-cycles, the cube-connected-cycles has somewhat simpler layouts. And although the shuffle-exchange graph appears to have smaller layouts for small values of N (see [LM81]), the cube-connected-cycles layouts are more regular and nicely structured.

Lastly, we would like to mention that the problem of finding a 3-dimensional layout for the shuffle-exchange graph with minimum volume remains unsolved. In fact, the shuffle-exchange graph is one of the few natural structures for which optimal 3-dimensional layouts are not known. (See [L82b] and [LR82] for a discussion of general layout strategies in 2 and 3 dimensions.)

9. Acknowledgements

10. References

[DGK81]   P. Diaconis, R. L. Graham and W. M. Kantor, "The mathematics of perfect shuffles," preprint, 1981.

[GO81a]   L. J. Guibas and A. M. Odlyzko, "Periods in strings," *Journal of Combinatorial Theory (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 (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.

[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.

[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. Thesis, Department of Computer Science, Carnegie-Mellon University, November 1980.

[L81]   F. T. Leighton, *Layouts for the Shuffle-Exchange Graph and Lower Bound Techniques for VLSI*, Ph.D. Thesis, Mathematics Department, Massachusetts ·Institute of Technology, Cambridge Massachusetts, September 1981.

[L82a]   F. T. Leighton, "New lower bound techniques for VLSI," *Math Systems Theory*, to appear.

[L82b]   F. T. Leighton, "A layout strategy for VLSI which is provably good," *Journal of Computer and System Sciences*, to appear.

[LLM82]   F. T. Leighton, M. Lepley, and G. L. Miller, "Layouts for the shuffle-exchange graph based on the complex plane diagram," submitted to *SIAM Journal of Algebraic and Discrete Methods*, June 1982.

[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.

[LR82]   F. T. Leighton and A. L. Rosenberg, "Three-dimensional circuit layouts," MIT-VLSI Technical Memo 82-102.

[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.

[P81]   F. P. Preparata, "Optimal three-dimensional VLSI layouts," *Math Systems Theory*, to appear.

[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 the Foundations of Computer Science*, October 1979, pp. 140-147.

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

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

[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.

[SR81]   D. Steinberg and M. Rodeh, "A layout for the shuffle-exchange network with $\Theta(N^2/log^{3/2}N)$ area," submitted to *IEEE Transactions on Computers*.

[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.