OPTIMAL LAYOUTS FOR SMALL SHUFFLE-EXCHANGE GRAPHS

F. Thomson Leighton and Gary L. Miller

Mathematics Department
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139, USA

1. INTRODUCTION

The shuffle-exchange graph is one of the best structures known for parallel computation. Among other things, it can be used to compute discrete Fourier transforms, multiply matrices, evaluate polynomials, perform permutations and sort lists [P80, S80, St71]. The algorithms needed for these operations are extremely simple and for the most part, require no more than logarithmic time and constant space per processor. The only exceptions are sorting lists (for which the best algorithm known requires $\Omega(log^2n)$ time) and performing arbitrary permutations (which may require $\Omega(logn)$ space per processor).

With the development of integrated circuit technology, it has become possible to place large numbers of very simple processors on a single chip. Thus the question of how best to lay out the shuffle-exchange graph on a grid (using as little area as possible) has gained practical as well as theoretical importance. Thompson was the first to address the issue in the context of VLSI. In his thesis [T80], he showed that any layout of the n-node shuffle-exchange graph requires at least $\Omega(n^2/log^2n)$ area. In addition, he described a layout requiring only $O(n^2/log^{1/2}n)$ area. Shortly thereafter, Hoey and Leiserson [HL80] improved the upper bound by finding an $O(n^2/logn)$-area layout. By combining the techniques of Thompson, Hoey and Leiserson, both Rodeh and Steinberg [RS80] and Leighton, Lepley and Miller [LLM81] independently found $O(n^2/log^{3/2}n)$-area layouts. The area layout question was finally settled by Kleitman, Leighton, Lepley and Miller [KLLM81] who employed entirely new methods to find an $O(n^2/log^2n)$ area-layout for the n-node shuffle-exchange graph, thus achieving Thompson's lower bound.

Although the $O(n^2/log^2n)$-area layout for the shuffle-exchange graph described in [KLLM81] is (up to a constant) *asymptotically* optimal, it is not optimal for small values of n (e.g., $n=2^7$). In fact, none of the general layout procedures thus far discovered [HL80, KLLM81, LLM81, RS80, T80] 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 paper, we describe techniques for finding good layouts for small shuffle-exchange graphs. Although the techniques 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 $n = 8,16,32,64$ and 128. 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.

2. PRELIMINARIES

(a) The shuffle-exchange graph

The *shuffle-exchange graph* consists of $n=2^k$ nodes and $3n/2$ edges. Each node is associated with a unique k-bit binary string $a_{k-1}\cdots a_0$. Two nodes ω and ω' are linked via a *shuffle edge* if ω' is a left or right cyclic shift of ω (i.e., if $\omega=a_{k-1}\cdots a_0$ and $\omega'=a_{k-2}\cdots a_0 a_{k-1}$ or $\omega'=a_0 a_{k-1}\cdots a_1$). Two nodes ω and ω' are linked via an *exchange edge* if ω and ω' differ only in the last bit (i.e., if $\omega=a_{k-1}\cdots a_1 0$ and $\omega'=a_{k-1}\cdots a_1 1$ or vice-versa).

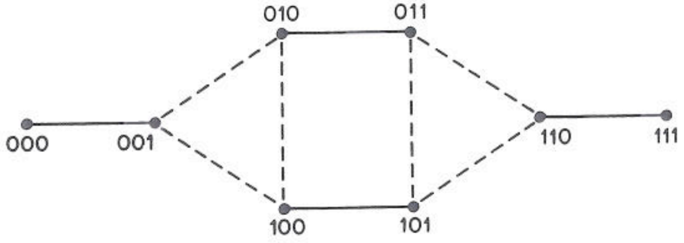

Fig. 1. The 8-node shuffle-exchange graph.

For example, we have drawn the *8*-node shuffle-exchange graph in *Figure 1*. In this figure, as well as throughout the rest of the paper, we have drawn the shuffle edges with *dashed lines* and the exchange edges with *solid lines*.

(b) Necklaces

The collection of all cyclic shifts of a node ω is called a *necklace* and is denoted by <ω>. For example, the necklace generated by *001* is <*001*> = {*001*, *010*, *100*}. Note that each necklace corresponds to a cycle in the shuffle-exchange graph (see *Figure 1*) and that shuffle edges always link nodes which are in the same necklace.

If a necklace contains precisely k nodes, then it is said to be *full*. Otherwise a necklace contains less than k nodes and is said to be *degenerate*. For example, <*001*> is full while <*000*> is degenerate.

The partition of the shuffle edges into necklaces is a key part of the layout technique described in *Section 3*.

(c) The Thompson model

In what follows, we will descibed layouts for the shuffle-exchange graph in terms of the *grid model* developed by Thompson [T80]. In this model, processors are represented by points which are located at the intersection of grid lines. Wires connect pairs of processors and must follow along grid lines. Two wires can cross each other but only at the intersection of grid lines (i.e., two wires cannot overlap for any distance). In addition, wires are not allowed to overlap processors. The *area* of a layout is defined to be the product of the number of vertical tracks and the number of horizontal tracks containing a processor or wire of the network.

As an example, we have included a Thompson model layout of the *8*-node shuffle-exchange graph in *Figure 2*. This layout requires *2* horizontal tracks and *6* vertical tracks, thus having area *12* (which is optimal). For simplicity, we have replaced the *3*-bit binary string associated with each node by its *numeric value*.

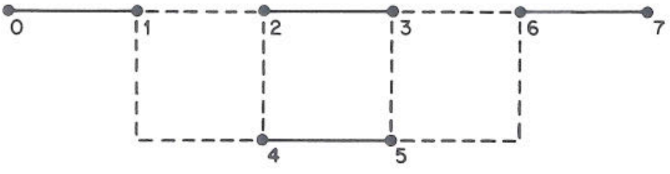

Fig. 2. An optimal Thompson model layout for the 8-node shuffle-exchange graph.

(d) Rational for using the Thompson model

We have chosen to use the Thompson model to illustrate our techniques because of its widespread acceptance and its simplicity. Although the assumption that processors can be represented by points is clearly false in practice, good Thompson model layouts can still be used to develop good practical layouts. The manner in which a Thompson model is useful 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 easily be hardwired in a k by k 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 each time step. For example, the optimal Thompson model layout in *Figure 2* (where $k=3$) can be transformed into the more realistic *6x18* layout shown in *Figure 3* by tripling the grid lines and replacing point processors by *3x3* boxes (into which the guts of each processor will later be wired).

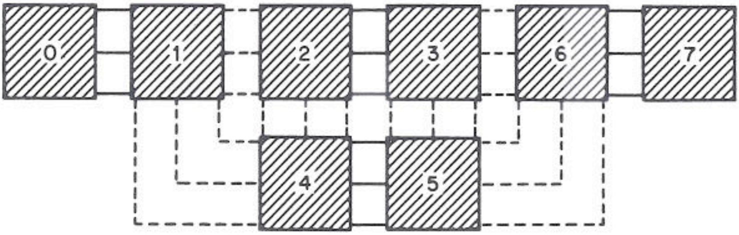

Fig. 3. A transformed Thompson model layout of the 8-node shuffle-exchange graph.

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 equiped 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.

3. LAYOUT TECHNIQUES

(a) A general class of layouts

In what follows, we will consider layouts of the shuffle-exchange graph for which:

1) each necklace appears as a rectangle consisting 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 in *Figures 2* and *4-7* have this form. In [LLM81], Leighton, Lepley and Miller show that such a layout exists for every shuffle-exchange graph. In fact, by rearranging the left-to-right ordering of the necklaces and/or the top-to-bottom ordering of the exchange edges, it is easy to produce a large class of such layouts for any shuffle-exchange graph. In particular, we will be interested in those layouts which use the smallest number of vertical and horizontal tracks. *The layouts in Figures 2 and 4-7 are optimal in this respect.*

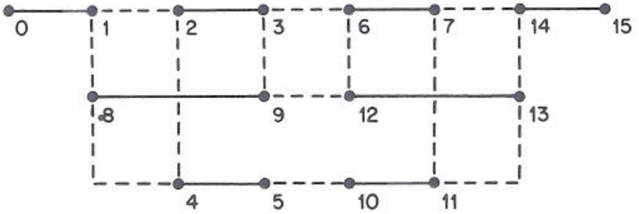

Fig. 4. A 3x8 Thompson model layout for the 16-node shuffle-exchange graph.

As is easily observed, these layouts require a surprisingly small amount of area. Further, the structure of the layouts facilitates efficient description, chip manufacture and data management. For example, for small values of n, it is feasible to attach a pin to each of the $\Theta(n/logn)$ necklaces, thus allowing n values to be loaded into an n-node shuffle-exchange graph in just $O(logn)$ steps.

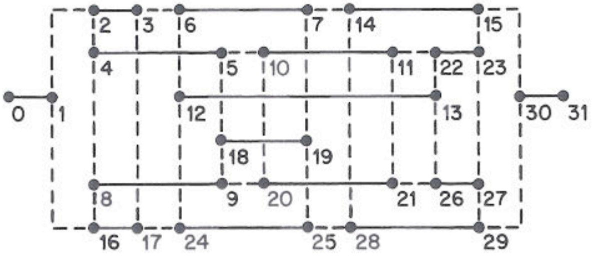

Fig. 5. A 6x14 Thompson model layout for the 32-node shuffle-exchange graph.

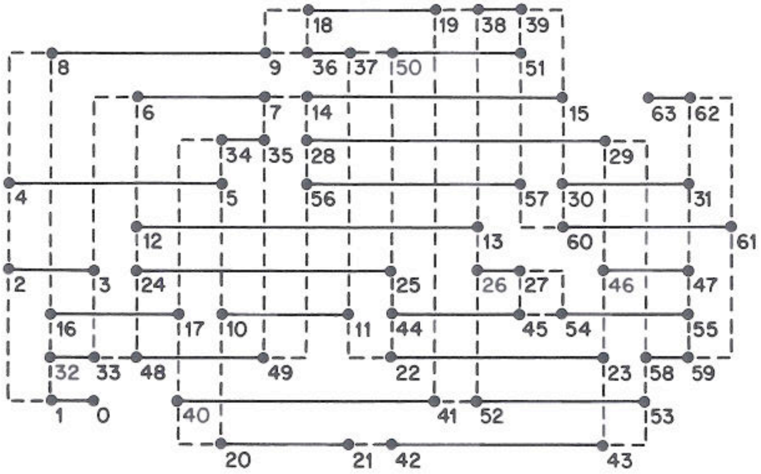

Fig. 6. An 11x18 Thompson model layout for the 64-node shuffle-exchange graph.

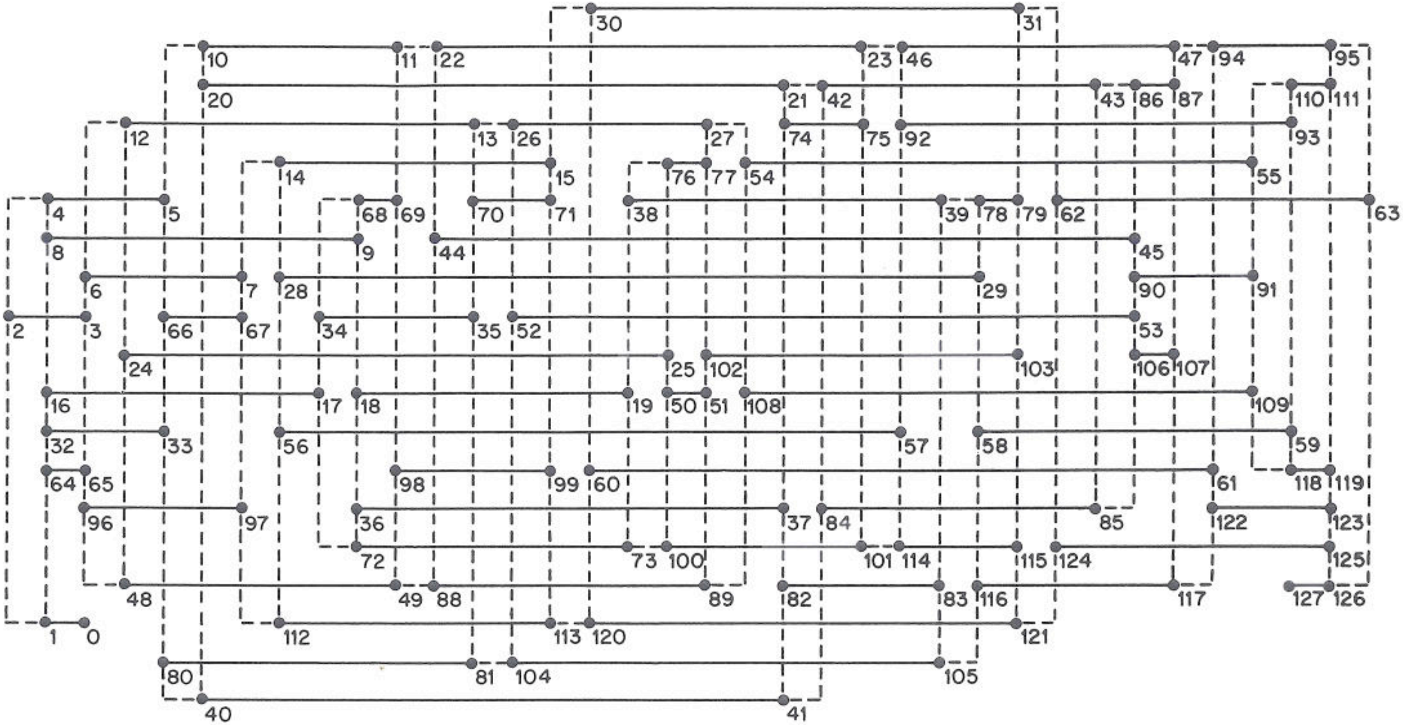

Fig. 7. A 19x36 Thompson model layout for the 128-node shuffle-exchange graph.

(b) Optimization Techniques

In order to find the layouts in *Figures 2* and *4-7*, we employed a combination of heuristics and exhaustive searches. The basic techniques are summarized in the following three steps.

Step 1: Partition the shuffle edges into necklaces. Order the necklaces linearly from left to right so that the number of exchange edges that overlap at each point of the ordering is kept small. More precisely, 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 5*. If we switch the necklace <*5*> with <*11*>, then *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 5* has just *6* horizontal tracks.

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, if we initially order the necklaces from left to right so that nodes with binary numbers containing r zeros are placed to the left of nodes containing $r-1$ zeros for $1\le r\le k$, then an optimal ordering can be produced by making at most one or two switches. The optimal ordering for $k=5$ ($n=32$) shown in *Figure 5* was produced by this method. Note that no switches were needed in this case. (See [LLM81] for a detaile theoretical discussion of such layouts).

A different but equally useful heuristic orders the necklaces from left to right so that the minimum value of the nodes in each necklace form an increasing sequence. The ordering in *Figure 5* could also be produced by this method. In this case, the minimal values are *0, 1, 3, 5, 7, 11, 15* and *31*. Layouts produced by this method are discussed from an asymptotic point of view in [KLLM81].

Probably the most difficult task is proving that a good ordering is optimal. The techniques we have used to prove optimality depend heavily on exhaustive searches. For $k\le 8$, the techniques have succeeded in proving the optimality of good orderings. For $9\le k\le 13$, we have found good orderings but have not been able to determine whether or not they are optimal. We have summarized the results in *Table 1*. Note that for each k, the maximum overlap of the best known ordering serves only as a lower bound on the number of horizontal tracks that will be required for any layout with that ordering. In some cases (e.g., $k=6,7$), additional horizontal tracks may be required.

TABLE 1

Maximum Overlap of Best Known Ordering

k	n	maximum overlap	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	?

Step 2: The next step is to insert the exchange edges using as few horizontal tracks as possible. The techniques we used to find the layouts in *Figures 2* and *4-7* are extensions of the theoretical ideas developed in [LLM81]. In that paper, the authors use the structural properties of the shuffle-exchange graph to produce layouts of the desired form but with an excessive number of horizontal tracks.

In our layouts, we first insert those exchange edges which cross a region of maximum overlap. To do this, we (for the most part) follow the top-to-bottom ordering given in [LLM81], making sure that every (or, as in the case with $k=6$ and 7, nearly every) horizontal track contains an edge in the region of maximum overlap. We next insert the exchange edges which cross neighboring regions in such a way that no new horizontal tracks are required. This process continues until all of the exchange edges are inserted. If done carefully, the number of horizontal tracks used will be the same as or only slightly larger than the maximum overlap.

Step 3: The third and final step is to adjust (if possible) the exchange edges so that degenerate necklaces can be doubled up with full necklaces and inserted into the same pair of vertical tracks. Since degenerate necklaces have substantially fewer nodes than do full necklaces, it is usually possible to accomplish this task without increasing the number of horizontal tracks used. For example, the number of vertical tracks needed to lay out the *16*-node and *64*-node shuffle-exchange graphs can be substantially reduced by this procedure.

(c) 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 8*.

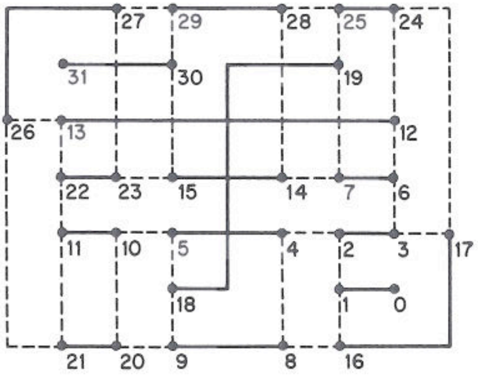

Fig. 8. A 7x9 Thompson model layout for the 32-node shuffle-exchange graph.

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.

4. ACKNOWLEDGEMENTS

This research was supported in part by the Advanced Research Projects Agency of the Department of Defense and monitered by the Office of Naval Research under contract number N00014-C-80-0622.

In addition, we would like to thank the following people for their helpful remarks and suggestions: D. Kleitman, C. Leiserson, M. Lepley, R. Rivest and R. Zippel.

5. REFERENCES

[HL80] D. Hoey and C.E. Leiserson, "A Layout for the Shuffle-Exchange Network," *Proceedings of the 1980 Conference on Parallel Processing*.

[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 the Theory of Computing*, Milwaukee, Wisconsin, 1981.

[LLM81] F.T. Leighton, M. Lepley and G.L. Miller, "New Layouts for the Shuffle-Exchange Graph Based on the Complex Plane Diagram," to appear as an MIT Tech Report.

[L80] C.E. Leiserson, "Area-Efficient Graph Layouts (for VLSI)," *Proceedings of the 21st Annual Symposium on Foundations of Computer Science*, Syracuse, New York, 1980.

[P80] D.S. Parker, "Notes on Shuffle-Exchange Type Switching Networks," *IEEE Transactions on Computers, C-29, 3 (March, 1980)*, pp. 213-222.

[RS80] M. Rodeh and D. Steinberg, "A Layout for the Shuffle-Exchange Graph with $O(n^2/log^{3/2}n)$ Area," Technical Report 088, IBM Israel Scientific Center, 1980.

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

[St71] H.S. Stone, "Parallel Processing with the Perfect Shuffle," *IEEE Transactions on Computers, C-20, 2 (February, 1971)*, pp. 153-161.

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