Compact Layouts of Banyan/FFT Networks*

David S. Wise
Indiana University
Computer Science Department
Bloomington, Indiana 47405

ABSTRACT A two-layer pattern is presented for the crossover pattern that appears as the FFT signal flow graph and in many switching networks like the banyan, delta, or omega nets. It is characterized by constant path length and a regular pattern, providing uniform propagation delay and capacitance, and ameliorating design problems for VLSI implementation. These are important issues since path length grows linearly with the number of inputs to such networks, even though switching delay seems to grow only logarithmically.

Independent of that pattern, an arrangement for stacking planes of such planar crossover patterns in three dimensions is described. Whereas a planar crossover pattern of $\Theta(m)$ inputs and outputs has at best $O(m)$ path length, the stacked pattern allows $O(\sqrt{m})$ path length. The scheme provides for stacking $2k$ planar nets (perhaps VLSI chips), each with k inputs/outputs into a network of k^2 inputs/outputs. Using this pattern, all such paths would have length (propagation delays) = $O(k)$.

Key Words and Phrases: VLSI, switching networks, delta networks, omega networks, multiprocessing.

CR Categories: 6.1, 4.32, 3.81, 3.83.

I. INTRODUCTION

This paper offers two results that can both be described as pictures. They are Figures 1 and 3. The perceptive reader may stop here, since the remainder of this paper only describes them.

---

*This research was supported in part by the National Science Foundation under grant number MCS 77-22325.

II. NOTATION

The logarithm-base-2 of n is written $\lg x$. For real functions f and g, we write $f(x) = O(g(x))$ if there is a constant k and some value x_0 such that $f(x) \leq k \cdot g(x)$ for all $x > x_0$. This notation expresses a proportional, asymptotic upper bound of g for f. If $f(x) = O(g(x))$ and $g(x) = O(f(x))$ then we write $f(x) = \Theta(g(x))$. It is only necessary to express g as a single term (e.g. lg x, 2^x, x^2) in such contexts.

The abbreviatios FFT and VLSI refer to the Fast Fourier Transform and Very Large Scale Integration circuit technology, respectively.

III. AN APPLICATION

This work is immediately motivated by the need for a switching network between processors and memory in a multiprocessor system of 100 or 1000 processors [4]. In order to increase bandwidth to memory, reducing contention among the processors, a banked memory is envisioned. Its access is through a fast, parallel switching network.

A suitable model for such a network is a banyan network [7] whose elementary functional unit is a 2x2 crossbar switch. It may be perceived as a router [2], a store-and-forward unit with two input lines and two output lines. Figure 1 might be interpreted as such a network from $2n = 16$ processor to $2n$ memories.

Memory fetch and store instructions are transmitted as packets through the network; duplicate networks pass information in the reverse direction. (Say, honoring a fetch instruction or allocating free nodes from a heap [5].) Each packet initially contains a binary (destination) memory-address followed by a message. Upon arrival at each router, its high-order address list determines along which path it is to be forwarded. The entire message is shifted left one bit, displacing that address bit for transmission to the next stage. In Figure 1, a zero bit would send the modified packet from a router on a leftward (northwest) line; a one bit would send the remainder of the packet to the right (northeast). It is possible to insert into the vacated low-order bit a value identifying the input line by which a packet entered each router. After a packet has passed through the network to its destination, its destination address will have been shifted out. In its stead (at the end of the packet) would be the address of its source processor when the vacant bits are so used.

This describes a network which uses crossover pattern to allow as many as n messages to arrive at some of 2n different destinations simultaneously, each over a path of at most lg n routers. As many as $n \cdot \lg n$ messages might be already in the switch, pipelined behind the arriving wave.

On the other hand, contention is still possible (for instance, all messages going to one destination) since each router can only handle one message at a time.

IV. PLANAR LAYOUT WITH CONSTANT PATH-LENGTH.

Figure 1 presents an alternative to the wiring diagram which is the essential component of many switching networks, such as the banyan network, the delta network, the omega network, and the Benes network (as either a circuit switch or a packet switch) [6,9]. It also appears as the broadcast pattern for intermediate results in the FFT [1].

The functional boxes, pictured as shaded boxes in the figure, are of little importance here. Their definitions vary with the various applications of this crossover network. The only assumption we make is that they are all of uniform design, perhaps mirror images of one another. Their size affects the aggregate size of the network.

Most analyses of such networks emphasize these functional boxes as the critical cost of such networks. That is, any path from the n input boxes (at the bottom) through the n output boxes (at the top) of such a graph passes through exactly lg n such boxes. If such a network is implemented in VLSI technology, however, n can become locally large and the apparently low O(log n) delay may be unattainable when timing considerations due to line-delay are included [3]. While Figure 1 has the property that all lines at any stage are of equal length, it still exhibits this line delay-problem. Drawn with uniform inter-wire spacing, it shows that the spacing between the stages, between functional boxes along any path, grows exponentially (top-to-bottom). Thus, although there are only lg n stages, every path length is $\Theta(n)$.

For any planar graph with n inputs/outputs, the longest path length is kn, for some constant k. This is easily proved, since the upper-left output must have a path from the lower-right input, and the network has width n. Figure 1 exhibits the desirable property that all paths are of length kn, so that special considerations due to line delay, attenuation, and capacitance are lessened. Functional boxes may be defined more uniformly, a desideratum in VLSI design.

Also important for planar fabrication technology is the fact that the crossover network is exactly two layers thick. While it exhibits more wire crossings than the common planar patterns (e.g. the cover of [1]), the others require three or more layers. Here one layer is composed of all diagonal wires running "northeast" (lower-left to upper-right) and the other layer is composed of "northwest" wires (lower-right to upper-left). The northeast wires might be in the metal layer of a VLSI chip and the northwest wires in the diffusion or (better) a second metal

sible (for
since each

ing diagram
switching
a network,
s either a
lso appears
ilts in the

oxes in the
efinitions
crossover
ney are all
ne another.
work.
size these
networks.
he bottom)
ch a graph
a network
can become
elay may be
line-delay
ty that all
l exhibits
inter-wire
he stages,
h, grows
there are

puts, the
c. This is
have a path
as width n.
l paths are
lue to line
lessened.
formly, a

hnology is
two layers
s than the
the others
s composed
er-left to
"northwest"
east wires
p and the
econd metal

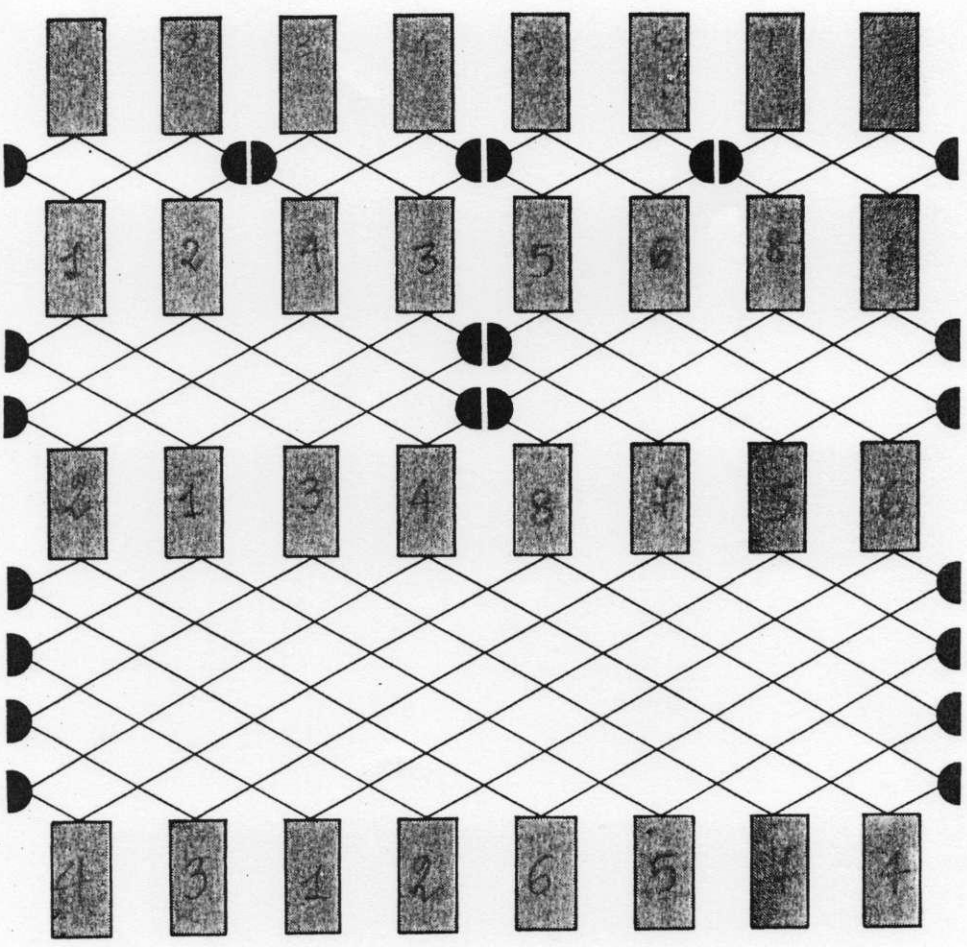

Figure 1. Crossover Net for n=8

layer. Alternatively the northeast wires might be on one side of a printed circuit board and the northwest wires on the other. The D-shaped blobs in the figure indicate contact cuts through the insulator connecting the layers.

The size of the contact-cut is one constraint on the height of the network. According to Mead and Conway, [7] their diameter, including insulating spacing, is 7λ in VLSI, larger than any insulated wire. Let that wire diameter be w, and let us measure the width of one functional box, including insulation as a multiple of $w = qw$.* As Figure 2 illustrates, the angle of either diagonal is then arcsin $1/q$. This is another constraint on the height of the network.

In fact, the height of the n input/output crossover net of Figure 1 can then be calculated accurately as

$$\sum_{0 \le i < \lg n} 2^i \cdot qw \cdot \tan\theta$$

$$= w(n-1)q/\sqrt{q^2-1} \ .$$

To this must be added the height of $(\lg n + 1)$ functional boxes.

The path length along any path is the height of the network times $q = \operatorname{cosec}\theta$ plus the path within $(\lg n + 1)$ functional boxes. This is not quite accurate, since additional width of the crossover net is necessary to prevent contact cuts from overlapping. Over all, the width must be increased by $wn/4$ to allow for the $n/4$ pairs of cuts across the second level of Figure 1; the top level can be wired without contact cuts.

It is then possible to derive the exact area of the crossover network. The width is $nw(q+0.25)$ and the area of the wiring is $w^2q(q+0.25)(q^2-1)^{-0.5}(n^2-n) = \boxed{\Theta(n^2)}$. To this must be added $w(q+0.25)n(\lg n+1)$ times the height of one functional box. This can be compared with the asymptotic area of a shuffle-exchange network [6] of $\Theta((n/\log n)^2)$; that network performs unpipelined FFT or packet switching (not circuit switching) with n functional boxes in $(\lg n+1)$ iterations, accounting for one factor of log n in its compression of area.

V. RACKING NETWORKS IN CUBIC SPACE.

Regardless of the technology or design of a planar crossover network of the sort presented above, extending it to large n -- say 10^4 or 10^6 -- requires a large plane. As we have seen, and as Franklin [3] points out, both width

---

*This does not allow horizontal spacing for immediately adjacent contact cuts -- the facing D's in Figure 1 -- but we are measuring vertical distance.

Embedding Interconnection Networks in Grids via the Layered Cross Product

Guy Even
Electrical Engineering—Systems Department, Tel-Aviv University, Tel-Aviv 69978, Israel

Shimon Even
Computer Science Department, Technion—Israel Institute of Technology, Haifa, 3200, Israel

A technique for automatically producing a rectilinear planar drawings of interconnection networks is described. It is based on the layered cross product suggested by Even and Litman. The technique is demonstrated on the butterfly network, the binary tree, and the mesh-of-trees of Leighton. © 2000 John Wiley & Sons, Inc.

Keywords: grid layout; layered cross product; graph drawing; network area; butterfly network; mesh-of-trees

1. THE LAYERED CROSS PRODUCT

Many layered graphs are known to be useful as interconnection networks. Examples are the butterfly network, including all its different looking representations, meshes of trees, fat-trees, the Beneš network, and multi-butterflies.

In [5], Even and Litman introduced the concept of the *layered cross product* (LCP) of layered graphs and showed that several of the important and well-known networks are LCPs of simple layered graphs, such as trees. Following is the definition of LCP from [5]:

A *layered graph*, of $l+1$ layers, $G = (V_0, V_1, \ldots, V_l, E)$, consists of

- $l + 1$ *layers* of vertices; V_i is the (nonempty) set of vertices in layer i.
- E is a set of directed edges. Every edge $\langle u, v\rangle$ connects two vertices of two adjacent layers, that is, if $u \in V_i$, then $v \in V_{i+1}$.

Let G^1, G^2 be layered graphs of $l + 1$ layers each, that is, for $j \in \{1,2\}, G^j = (V_0^j, V_1^j, \ldots, V_l^j, E^j)$. Their LCP $G^1 \times G^2$ is a layered graph, $G = (V_0, V_1, \ldots, V_l, E)$, where

- For every $0 \le i \le l, V_i = V_i^1 \times V_i^2$.
- There is an edge $\langle u, v\rangle$ in G, connecting vertices $u = (u^1, u^2)$ and $v = (v^1, v^2)$, if and only if $\langle u^1, v^1\rangle$ and $\langle u^2, v^2\rangle$ are edges in G^1 and G^2, respectively.

In [5], no regard is paid to the labels or names assigned to vertices or to the order in which vertices on the same layer are drawn, that is, two-layered graphs are considered *equal* if they are isomorphic via a level-preserving mapping. Under this assumption, the LCP operation is commutative and associative. Thus, we may consider the LCP of more than two-layered graphs, all with the same number of layers, without regard to the order in which they are written or the order in which the binary operations are applied.

Two examples are depicted in Figure 1. Some of the results shown in [5] are

- The butterfly network is the LCP of two binary trees. The mesh of trees is the LCP of two binary trees with paths attached to their leaves. A particular fat-tree is the LCP of a binary tree and a quad-tree.
- Several of the important properties of these networks are shown to be trivial consequences, once a network is presented as an LCP of simpler graphs.

2. CREATING LAYOUTS VIA LCPs

2.1. *Objectives*

An abstraction of a VLSI layout problem is to model the circuit as a graph and to look for an embedding of the graph in a rectilinear grid, using the following rules:

- Vertices of the graph are mapped to grid-points, at most one vertex per grid-point.

Received February 1999; accepted March 2000
Correspondence to: G. Even; e-mail: guy@eng.tau.ac.il URL: http://www.eng.tau.ac.il/~guy/
Contract grant number: United States–Israel Binational Science Foundation
Contract grant number: 94-00266/1
Contract grant sponsor: Fund for the Promotion of Research at the Technion

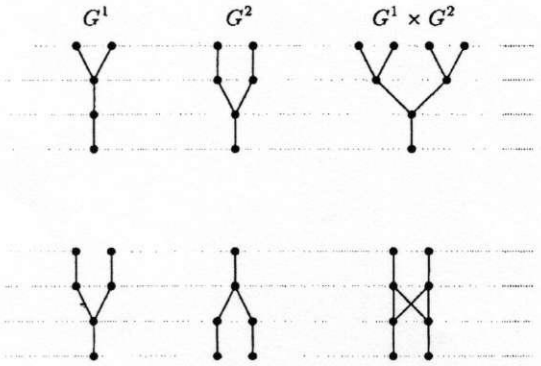

FIG. 1. Two examples of LCPs.

- Edges of the graph are to be routed along grid paths by an edge-disjoint mapping, namely, an edge of the grid may not belong to more than one routing path. Two paths may share an intermediate grid-point, but they must cross at this point, that is, no knock-knees are allowed.
- If a vertex is mapped to a grid-point, then all paths representing edges incident on this vertex must end at that grid-point, and no other path is allowed to pass through that point.

A layout which obeys these rules is called *rectilinear*. The *area* of a layout is defined to be the smallest area of a rectangle, sides parallel to the grid lines, which contains the entire layout. (The height of a rectangle is measured by the number of horizontal grid-lines in it and, similarly, for the width. The area is the product of the two.)

Another desirable property, when a grid layout represents a circuit layout, is to have the input/output (I/O) terminals on the boundary of the region of the grid that contains the layout [2, 6]. The terminals in the layouts proposed in this paper are not placed along the boundaries. Note, however, that this may not be an important constraint if the terminals denote the processors to be connected, and these are realized on the same chip or printed board.

2.2. *The Projection Methodology*

In this section, we describe a *projection methodology* (PM) for layout construction, which grows from the LCP.

Consider two layered graphs, G^1 and G^2 of $l+1$ layers each. Let $G = (V_0, V_1, \ldots, V_l, E)$ denote the LCP $G^1 \times G^2$. A layout of G is obtained in the PM as follows: Consider a cube and draw the graph G_1 on the x–y face of the cube so that (a) the y-coordinate of every vertex $u \in V_i^1$ equals i, and (b) the x-coordinate of every vertex is an integer. Similarly, draw the graph G_2 on the y–z face of a cube: The y-coordinate of every vertex $v \in V_i^2$ equals i, and the z-coordinate is an integer. A three-dimensional drawing of the LCP G is constructed in the cube as follows: If $u \in V_i^1$ is drawn in coordinates $(x_u, i, 0)$ and $v \in V_i^2$ is drawn in coordinates $(0, i, z_v)$, then the coordinates of vertex $(u, v) \in V_i$ are (x_u, i, z_v). A two-dimensional drawing of G is obtained by projecting the three-dimensional drawing to the x–z plane.

We now describe how rectilinear layouts of G can be obtained via the PM. First, we formalize necessary and sufficient conditions for the edges of the x–z projection of G to be along grid paths, for vertices to be mapped to different grid points, and for not using any grid edge more than once.

In Figure 2, we use standard techniques of descriptive geometry to represent the *vertical* projection on the x–y plane, the *horizontal* projection on the y–z plane, and the floor projection. Vertices belonging to layer V_0 of the multiplicand graphs and the product graphs are drawn as circles, whereas vertices belonging to layer V_1 are drawn as diamonds. All edges are drawn as straight-line segments.

Figure 2 depicts three types of edges in the product graph: (a) The product of the edges e_3 and e_2 (with their endpoints) yields the edge e_6. Note that when both multiplied edges are diagonal then their product edge is also diagonal. (b) The product of the edges e_3 and e_1 yields the edge e_5. Similarly, the product of the edges e_4 and e_2 yields the edge e_7. Note that if exactly one of the edges that we multiply is diagonal then the product edge is straight (coincides with a grid line). (c) The product of the straight edges e_4 and e_1 yields a product edge. However, the endpoints of this product edge are mapped to the same point in the floor projection, thus violating the layout condition that vertices must be mapped to different grid-vertices.

The implications of these observations are summarized in the following claims:

CLAIM 1. For every edge $e = \langle (u^1, u^2), (v^1, v^2) \rangle \in E$, where $e_i = \langle u^i, v^i \rangle \in E^i$, for $i = 1, 2$. The PM generates a layout of G in which the edges are grid lines if and only if the drawings of G_1 and G_2 on the faces of the cube satisfy the following condition: For every edge $e \in E$, exactly one of the edges e_1 and e_2 is drawn diagonally,

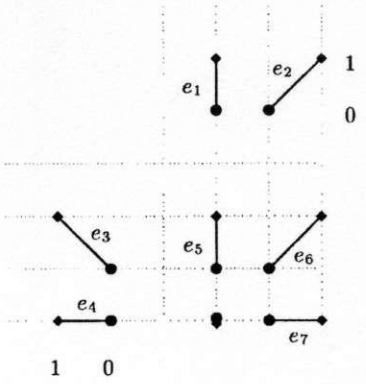

FIG. 2. An example of PM.

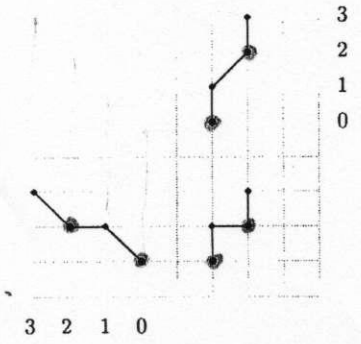

FIG. 3. A rectilinear path obtained by alternating diagonal levels.

namely, either the x-coordinates of the endpoints of e_1 are different or the z-coordinates of the endpoints of e_2 are different, but not both.

CLAIM 2. The PM generates a layout of G in which at most one vertex is mapped to each grid point if and only if the sets $\{(x_u, z_v)\colon u \in V_i^1 \text{ and } v \in V_i^2\}$ are disjoint, for $i = 0, \ldots, l$.

Consider two diagonal edges $\langle a, b\rangle$ and $\langle c, d\rangle$ of one of the two multiplicands G^1, where the x–y coordinates of these four vertices are $(x_a, i, 0), (x_b, i+1, 0), (x_c, j, 0)$, and $(x_d, j+1, 0)$. We say that these two edges are *consistent* if the open intervals (x_a, x_b) and (x_c, x_d) are disjoint. Consistent edges are defined similarly for G^2.

CLAIM 3. The PM generates a layout of G in which no grid edge is used twice if and only if for every two inconsistent edges of one of the multiplicands the following condition holds: The two edges are not in the same level of the multiplicand, and on the two levels in which they appear, there are no (straight) edges of the other multiplicand which are collinear.

What do we do when the condition stated in Claim 1 is not satisfied? Our solution is to double the number of edge levels so that edges in the drawing of G^1 are diagonal in the odd layers and straight in the even layers, while the edges in the drawing of G^2 are straight in the odd layers and diagonal in the even layers. The doubling of the number of edge levels is achieved by stretching each edge of the two multiplicands to become a path of two edges. Note that if there is a path in G^1 which has straight and diagonal alternating edges, and a path in G^2 of a similar form, but the diagonal edges of G^2 are in the levels in which the edges of G^1 are straight then the path of their product, in the floor projection, is a rectilinear path, that is, only right angles are obtained. Figure 3 depicts a rectilinear embedding of a single-product path.

A simple way to guarantee that the condition of Claim 2 will hold is to make sure that no two vertices in the drawing of G^1 (G^2), except for the two endpoints of the same straight edge, share the x-coordinate (z-coordinate).

The condition of Claim 3 is harder to enforce and is a severe limitation on our technique. For this reason, the examples of the next section are limited to networks, each of which is the LCP of two trees.

3. EXAMPLES

In this section, we present how PM is used to construct rectilinear layouts of the butterfly network, a binary tree, and the mesh-of-trees.

3.1. *The Butterfly Network*

Consider the situation illustrated in Figure 4. The butterfly network is the LCP of two binary trees, one drawn upward and one drawn downward. (We dedicate a column to each vertex to prevent vertices of the layout from colliding.) Draw one tree next to the x–y plane (the north wall) and the other next to the y–z plane (the west wall). Construct their LCP in three dimensions inside the cube, in such a way that the two trees are the projections of the resulting butterfly on the x–y and y–z planes. The projection of this three-dimensional figure on the floor, depicted as the shadow cast by the butterfly, is a planar layout of the butterfly in area $4n^2$. This layout of the

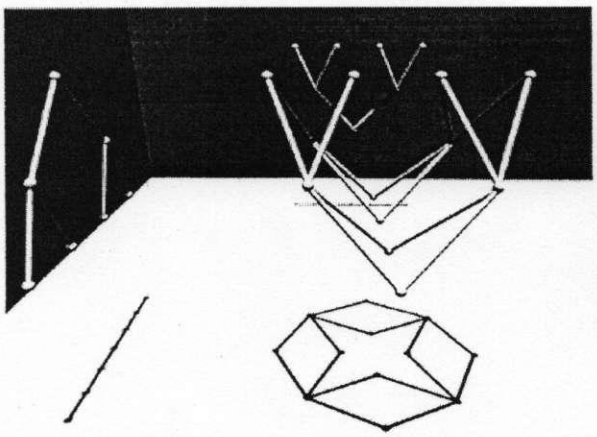

FIG. 4. Butterfly layout with diagonal lines.

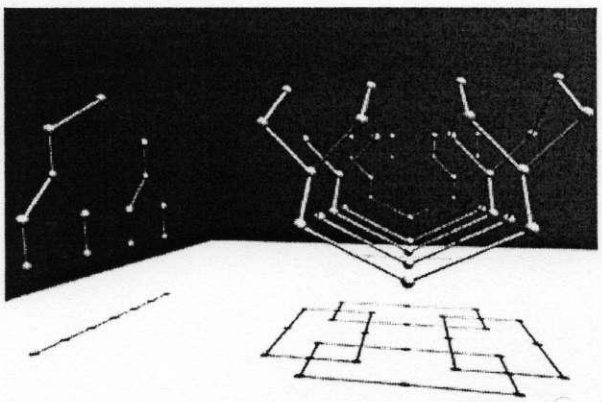

FIG. 5. A rectilinear layout of the butterfly.

butterfly is symmetric under rotation; however, it is not rectilinear since it uses diagonal lines. Note that the I/O terminals of the butterfly are on the central north–south and east–west symmetry axes.

Figure 5 depicts a rectilinear layout of the butterfly network obtained by doubling the number of edge-layers. (Smaller rectilinear layouts of the butterfly are known. Dinitz [3] showed that area $\frac{11}{6}n^2$ suffices. Avior et al. [1] showed that $n^2 + o(n^2)$ area is necessary and sufficient, if one measures the area by a rectangle whose sides are parallel to the grid axes. Dinitz, et al. [4] showed recently that if slanted rectangles are allowed to be used to measure the grid-area then the optimal area is $\frac{1}{2}n^2 + o(n^2)$. However, these layouts are not produced automatically through the LCP.)

3.2. *Binary Trees*

Let l be an even integer. The binary tree with $2^l = n$ leaves is the LCP of two stretched binary trees, each having $\sqrt{n}$ leaves. We can get a rectilinear drawing of the binary tree by a slightly modified PM "stretching" technique. In the vertical projection, put a stretched tree, with a vertical extra edge attached to its root, and trim the leaves to be of one edge, instead of the (stretched) two. In the horizontal projection, we use an unmodified stretched tree. Observe that here, too, a diagonal edge is only multiplied by a straight edge to obtain a rectilinear layout, as depicted in Figure 6. In fact, the result is the famous H-tree, of area $4n - 4\sqrt{n} + 1$, invented by Shiloach [8].

3.3. *Mesh-of-Trees*

Finally, let us demonstrate how PM can be used to generate a rectilinear layout of the mesh-of-trees (MOT), invented by Leighton [7], in area $(n(1 + \log n) - 1)^2$. This network has n input and n output terminals, and it is nonblocking in the strict sense, that is, if a pair of input and output terminals are free, then there is a unique path of length $2 \log n$ connecting them, and this path is vertex disjoint from any other path which may be in use. (In fact, we realize a slightly modified version of this network, in which n^2 vertices and edges are saved and the terminal connecting paths are of length $2 \log n - 1$.)

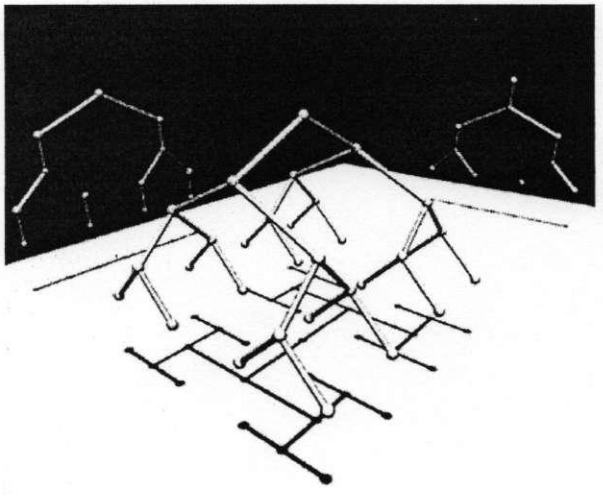

FIG. 6. H-tree: A rectilinear layout of a binary tree.

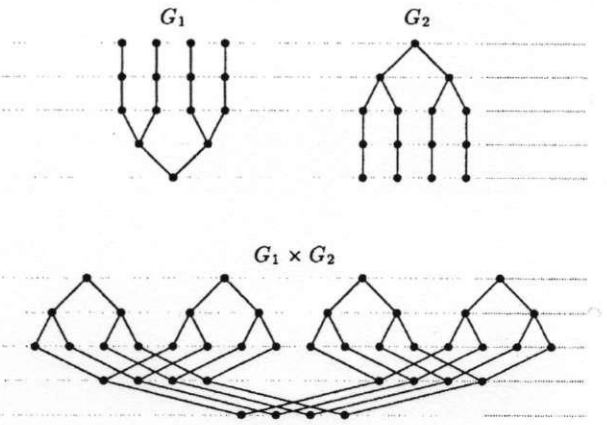

FIG. 7. An MOT is the LCP of two binary trees with attached paths.

Even and Litman [5] showed that the MOT is the LCP of two binary trees with paths attached to their leaves, one with the root up and one with the root down. For $2^l = n$, each tree has n leaves on the l-th level, and to each leaf, a path of length l is attached. Thus, the total number of layers (of vertices) in each tree is $2l + 1$. The trees and their LCP, for the case $l = 2$, are shown in Figure 7.

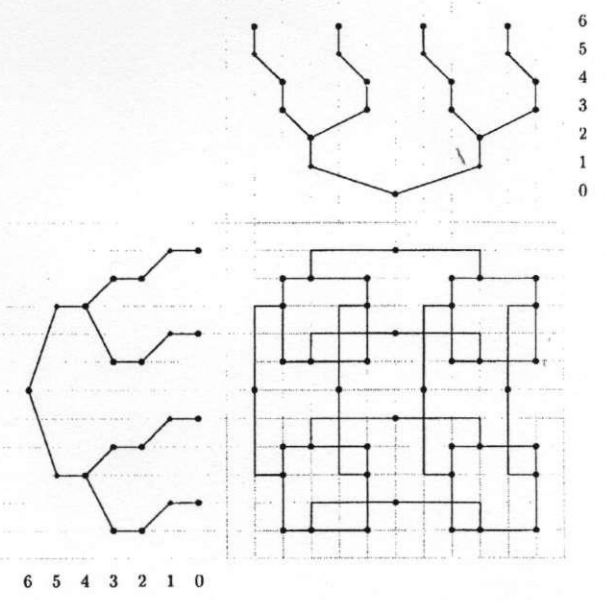

FIG. 8. A rectified layout of the MOT for $l = 2$. The product of the vertices, which belong to layer 3, constitutes the "mesh-vertices."

Observe that the middle layer consists of n^2 vertices of degree 2. These vertices can easily be dispensed with: Modify the multiplicand trees to have paths of length $l - 1$ attached to their leaves, instead of length l. Now the number of layers is $2l$, instead of $2l + 1$.

We generate a rectilinear layout of the modified MOT using PM as follows: We stretch the binary tree part in each multiplicand so that starting from the root; we alternate between a diagonal edge and a rectified edge. The paths part of each multiplicand is stretched as well. To make sure that vertices of the floor projection will not coincide, for each vertex on an even indexed layer of the vertical (horizontal) projection, we use a dedicated column (row). Thus, the number of columns (rows) used is $(l + 1) \cdot n - 1$, and, therefore, the area of the floor projection is $((1 + \log n)n - 1)^2$. The resulting layout for $l = 2$ is shown in Figure 8.

Acknowledgments

We thank Volker Hofmeyer for his invaluable help with the production of Figures 4, 5, and 6.

REFERENCES

[1] A. Avior, T. Calamoneri, S. Even, A. Litman, and A.L. Rosenberg, A tight layout of the Butterfly network, Theory Comput Syst 31 (1998), 475–488.

[2] R.P. Brent and H.T. Kung, On the area of binary tree layouts, Inf Proc Lett 11 (1980), 44–46.

[3] Ye. Dinitz, A compact layout of butterfly on the square grid, Technical Report No. 873, Dept. of Computer Science, Technion, 1995, 7 pp.

[4] Ye. Dinitz, S. Even, and M. Zapolotsky, An optimal layout of the butterfly network, Manuscript, 1999, 14 pp, presented at the Dagstuhl Seminar (No. 219) on Graph Algorithms and Applications, July 1998.

[5] S. Even and A. Litman, Layered cross product—A technique to construct interconnection networks, Networks 29 (1992), 219–223.

[6] M.J. Fischer and M.S. Paterson, Optimal tree layout, 12th ACM Symp on Theory of Computing, 1980, pp. 177–189.

[7] T. Leighton, Parallel computation using meshes of trees, Proc WG'83, Int Workshop on Graph Theoretic Concepts in Computer Science, M. Nagl and J. Perl (Editors), Trauner Verlag, Linz, 1984, pp. 200–218.

[8] Y. Shiloach, Linear and planar arrangements of graphs, Doctoral Thesis, Dept. of Applied Mathematics, The Weizmann Institute of Science, Rehovot, Israel, 1976.