

**A Tight Layout of the
Butterfly Network**

Aythan Avior, Tiziana Calamoneri,
Shimon Even, Ami Litman & Arnold L. Rosenberg
CMPSCI Technical Report 96-05
January, 1996

A Tight Layout of the Butterfly Network

Aythan Avior¹ Tiziana Calamoneri² Shimon Even³
Ami Litman⁴ Arnold L. Rosenberg⁵

Abstract. We establish upper and lower bounds on the layout area of the butterfly network, which differ only in low-order terms. Specifically, the N -input, N -output butterfly network can be laid out in area $(1 + o(1))N^2$, while no layout of the network can have area smaller than $(1 - o(1))N^2$. These results improve both the known upper bound and the known lower bound on the area of butterfly network layouts.

¹ Computer Science Department, Technion, Haifa 3200, Israel [aythan@cs.technion.ac.il]

² Computer Science Department, University of Rome “La Sapienza”, Via Salaria 113, 00198 Roma, Italy [calamo@dsi.uniroma1.it]

³ Computer Science Department, Technion, Haifa 3200, Israel [even@cs.technion.ac.il]

Supported by the Fund for the Promotion of Research at the Technion.

Supported by the United States-Israel Binational Science Foundation, Grant no. 94-00266.

⁴ Computer Science Department, Technion, Haifa 3200, Israel [litman@cs.technion.ac.il]

Supported by the United States-Israel Binational Science Foundation, Grant no. 94-00266.

⁵ Computer Science Department, University of Massachusetts, Amherst, MA 01003, USA [rsnrbg@cs.umass.edu]

A portion of this research was done while visiting the Computer Science Department at the Technion.

Supported in part by NSF Grant CCR-92-21785 and in part by the United States-Israel Binational Science Foundation Grant no. 94-00266.

1 Introduction

1.1 Overview

Layouts of graphs on rectilinear grids are of wide interest in the study of the VLSI layout problem for integrated circuits [13], as well as in the study of algorithms for drawing graphs. Further, each such layout is a restricted form of embedding of a graph in the grid, hence contributes to the study of the mapping problem for parallel architectures [2, 5], particularly the problem of mapping parallel programs onto mesh-structured parallel architectures; cf. [12].

The fields of graph embedding and VLSI layout have developed powerful techniques which produce embeddings and layouts which are quite efficient—often within constant factors of optimal; cf. [3, 4]. However, even a modest constant factor may render an asymptotically optimal layout or embedding unacceptably inefficient in practice. This observation motivates the current paper.

The aim of this paper is to find a grid-layout of the *butterfly network* [3] whose deviation from optimality is of lower order than a constant factor. We achieve this goal by presenting, in Section 2, a layout of the N -input, N -output butterfly network whose area is $(1 + o(1))N^2$, and by proving, in Section 3, that no layout of this network can have area smaller than $(1 - o(1))N^2$. Thus, our upper and lower bounds coincide, up to a low-order additive term.

Both the upper- and lower-bound components of our result improve prior bounds for butterfly network layouts. The previously known lower bound for the layout area of the N -input, N -output butterfly network was $\frac{1}{4}N^2$ [13, 14]. The 1981 upper bound of $2N^2$ for the same problem [15] was improved only in 1992, to $\frac{11}{6}N^2$ [6].

1.2 The Formal Setup

A. The Graphs of Interest

We define several graphs that are germane to our study. We begin with the two graphs that are of primary interest, butterfly networks and grids.

Butterfly networks. For each integer n , the n -level *butterfly network* \mathcal{B}_n has node-set¹

$$\{0, 1, \dots, n\} \times \{0, 1\}^n.$$

¹ $\{0, 1\}^n$ denotes the set of length- n binary strings.

For each $0 \leq \ell \leq n$, the set $\{\ell\} \times \{0, 1\}^n$ is the ℓ th level of \mathcal{B}_n . The nodes at level 0 of \mathcal{B}_n are called *inputs*, and those at level n are called *outputs*.² The string $x \in \{0, 1\}^n$ is the *position-within-level string* (*PWL string*, for short) of node $\langle \ell, x \rangle$. Each node

$$\langle \ell, \beta_0\beta_1 \cdots \beta_{\ell-1}\beta_\ell\beta_{\ell+1} \cdots \beta_{n-1} \rangle$$

on level ℓ ($0 \leq \ell < n$; each $\beta_i \in \{0, 1\}$) of \mathcal{B}_n is connected by a *level- ℓ straight-edge* with node

$$\langle \ell + 1, \beta_0\beta_1 \cdots \beta_{\ell-1}\beta_\ell\beta_{\ell+1} \cdots \beta_{n-1} \rangle$$

on level $\ell + 1$, and by a *level- ℓ cross-edge* with node

$$\langle \ell + 1, \beta_0\beta_1 \cdots \beta_{\ell-1}\bar{\beta}_\ell\beta_{\ell+1} \cdots \beta_{n-1} \rangle$$

on level $\ell + 1$. When we draw \mathcal{B}_n level by level, in such a way that, at each level, the PWL strings are the reversals of the binary representations of the integers $0, 1, \dots, 2^n - 1$, in that order³, we get the familiar drawing of \mathcal{B}_n shown in Figure 1(a).

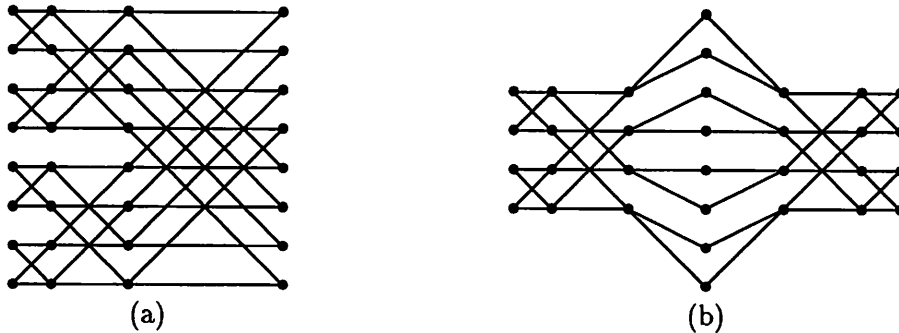


Figure 1: (a) \mathcal{B}_3 ; (b) another view of \mathcal{B}_3 .

The following folklore lemma will be useful in our layout of \mathcal{B}_n [7].

Lemma 1.1 *For any nonnegative integers j, k the subgraph of \mathcal{B}_n induced by the nodes of levels $j, j + 1, \dots, j + k$ is the disjoint sum⁴ of 2^{n-k} copies of \mathcal{B}_k .*

²The terms “input” and “output” derive from the fact that the $(n + 1)$ -level butterfly network is the data-dependency graph of the 2^n -input Fast Fourier Transform algorithm [1].

³We call this the *natural ordering* of the levels of \mathcal{B}_n .

⁴We use the term “sum” here, rather than “union” to emphasize that the constituent graphs share neither nodes nor edges.

Grids. For integers m and n , the $m \times n$ grid (or, mesh) $\mathcal{M}_{m,n}$ has node-set

$$\{1, 2, \dots, m\} \times \{1, 2, \dots, n\}.$$

The edges of $\mathcal{M}_{m,n}$ connect nodes $\langle i, j \rangle$ and $\langle i', j' \rangle$ just when $|i - i'| + |j - j'| = 1$. The path induced by the set of nodes $\{i\} \times \{1, 2, \dots, n\}$ (resp., the set $\{1, 2, \dots, m\} \times \{j\}$) is the i th row (resp., the j th column) of $\mathcal{M}_{m,n}$. Row 1 is the top side of $\mathcal{M}_{m,n}$; row m is the bottom side; column 1 is the left side of $\mathcal{M}_{m,n}$; column n is the right side. Finally, we call the product mn the area of grid $\mathcal{M}_{m,n}$.

Three auxiliary graphs will be used in our study. Two augmented versions of the butterfly network will be useful in constructing our layout of \mathcal{B}_n in Section 2.

Augmented butterfly networks. We denote by \mathcal{B}'_n the network obtained by appending two new nodes, called *output terminals* to each output of \mathcal{B}_n ; see Figure 2(a). We denote by \mathcal{B}''_n the network obtained by appending two new nodes, called *input terminals* to each input of \mathcal{B}'_n ; see Figure 2(b). We refer to the input terminals collectively and to the output terminals collectively as a *terminal group*.



Figure 2: (a) \mathcal{B}'_2 ; (b) \mathcal{B}''_2 .

The following folklore result, a proof of which can be found in [7], indicates that the designations “input” and “output” in our definitions of \mathcal{B}_n and \mathcal{B}''_n are artificial and are useful only as an aid in visualizing the networks. This fact is crucial in our layout strategy.

Lemma 1.2 *There is an automorphism of \mathcal{B}_n which maps each level $0 \leq \ell \leq n$ of the networks onto level $n - \ell$. There is a similar automorphism of \mathcal{B}''_n , which also maps the input terminals onto the output terminals and vice versa.*

Our lower bound argument in Section 3 uses the following well-known auxiliary graph.

Complete bipartite graphs. The $N \times N$ complete bipartite graph $\mathcal{K}_{N,N}$ has N input nodes $V^{(i)}$ and N output nodes $V^{(o)}$; its edges connect every input $u \in V^{(i)}$ with every output $v \in V^{(o)}$.

B. Graph Embeddings and Grid Layouts

Graph embeddings. An *embedding of graph \mathcal{G} into graph \mathcal{H}* (which has at least as many nodes as \mathcal{G}) comprises a one-to-one association α of the nodes of \mathcal{G} with nodes of \mathcal{H} , plus a *routing* ρ which associates each edge (u, v) of \mathcal{G} with a path in \mathcal{H} that connects nodes $\alpha(u)$ and $\alpha(v)$. The *congestion* of embedding $\langle \alpha, \rho \rangle$ is the maximum, over all edges e of \mathcal{H} , of the number of edges of \mathcal{G} whose ρ -routing paths contain edge e .

Grid layouts. We formulate the notion of a *grid-layout* of a graph \mathcal{G} as a special kind of embedding of \mathcal{G} into a grid; there are alternative, equivalent formulations of the notion which make it a special kind of drawing of \mathcal{G} in the plane. Following [13], a *layout* of an N -node graph \mathcal{G} in a grid $\mathcal{M}_{m,n}$, where $N \leq mn$, is an embedding $\langle \alpha, \rho \rangle$ of \mathcal{G} into $\mathcal{M}_{m,n}$ whose routing paths collectively satisfy the following conditions.

- Distinct routing paths are edge-disjoint, so that the embedding that embodies a layout has unit congestion. It follows that at most two routing paths can “cross” at a node of $\mathcal{M}_{m,n}$, i.e., touch the node without terminating there.
- Routing paths that share an intermediate node of $\mathcal{M}_{m,n}$ must *cross* at that node; that is, one path enters the node from the left and leaves toward the right, while the other path enters the node from the bottom and leaves toward the top. Thus, we do not allow “knock-knee” routing [9].
- A routing path may touch no image node $\alpha(u)$, except at its endpoints.

We denote by $AREA(\mathcal{G})$ the area of the smallest grid in which \mathcal{G} can be laid out.

C. Graph Splicing

Our layout of \mathcal{B}_n in Section 2 proceeds by finding layouts of subgraphs of \mathcal{B}_n and “splicing” them together. We now define this intuitive operation formally.

Let us be given a graph \mathcal{G} , a graph \mathcal{H} , a sequence $\sigma = \langle u_1, u_2, \dots, u_k \rangle$ of distinct nodes of \mathcal{G} , and an equal-size sequence $\sigma' = \langle v_1, v_2, \dots, v_k \rangle$ of distinct nodes of \mathcal{H} . Say that \mathcal{G} has nodes $U \cup \{u_1, u_2, \dots, u_k\}$ and that \mathcal{H} has nodes $V \cup \{v_1, v_2, \dots, v_k\}$, where the sets U , V , $\{u_1, u_2, \dots, u_k\}$ and $\{v_1, v_2, \dots, v_k\}$ are pairwise disjoint. The operation of *splicing graphs \mathcal{G} and \mathcal{H} along sequences σ and σ'* produces the graph \mathcal{F} whose nodes comprise the set

$$U \cup V \cup \{\langle u_1, v_1 \rangle, \langle u_2, v_2 \rangle, \dots, \langle u_k, v_k \rangle\}$$

and whose edges connect node w_1 and w_2 just when:

- $\{w_1, w_2\} \subseteq U$ (resp., $\{w_1, w_2\} \subseteq V$), and w_1 and w_2 are adjacent in \mathcal{G} (resp., in \mathcal{H});

- $w_1 = \langle u_i, v_i \rangle$, $w_2 \in U$, and u_i and w_2 are adjacent in \mathcal{G} ;
- $w_1 = \langle u_i, v_i \rangle$, $w_2 \in V$, and v_i and w_2 are adjacent in \mathcal{H} ;
- $w_1 = \langle u_i, v_i \rangle$, $w_2 = \langle u_j, v_j \rangle$, and either u_i and u_j are adjacent in \mathcal{G} , or v_i and v_j are adjacent in \mathcal{H} (or both).

Examples of splicing. (a) One can splice one copy of $\mathcal{M}_{m,n}$ along its right side to another copy of $\mathcal{M}_{m,n}$ along its left side, to produce an instance of $\mathcal{M}_{m,2n-1}$. (b) One can produce \mathcal{B}_3 by appropriately splicing the disjoint sum of two copies of \mathcal{B}_2 with the disjoint sum of four copies of \mathcal{B}_1 ; the former sum produces the first two levels of \mathcal{B}_3 , the latter sum produces the last level of \mathcal{B}_3 , and the two sums combine to produce the third level of \mathcal{B}_3 ; see Figure 1(a).

Our final example, depicted in Figure 1(b), is so germane to our layout that we encapsulate it as a lemma (whose proof is left to the reader).

Lemma 1.3 *Let the sequence $\sigma = \langle u_1, u_2, \dots, u_{2^{n+1}} \rangle$ list all the output terminals of \mathcal{B}'_n , in an arbitrary order. Splicing \mathcal{B}'_n with a copy of itself along sequence σ produces \mathcal{B}_{n+1} .*

We finally have all the machinery we need to study grid layouts of butterfly networks. To simplify our exposition, let n henceforth be an arbitrary positive integer, and let $N = 2^n$.

2 The Upper Bound on Layout Area

This section is devoted to proving our upper bound on the layout area of \mathcal{B}_n .

Theorem 2.1 *For all positive integers n , there is a grid-layout \mathcal{L}_n of \mathcal{B}_n with AREA $(N + o(N))^2$. Hence,*

$$\text{AREA}(\mathcal{B}_n) \leq (1 + o(1))N^2.$$

We prove Theorem 2.1 via a sequence of reductions.

2.1 The First Reduction

We show how to construct the desired layout of \mathcal{B}_{n+2} from four copies of a suitable layout of \mathcal{B}_n'' .

Claim 2.1 *One can construct a grid-layout \mathcal{L}_{n+2} of \mathcal{B}_{n+2} with the area indicated in Theorem 2.1, from four copies of a grid-layout \mathcal{L}_n'' of \mathcal{B}_n'' that has the following properties.*

- \mathcal{L}_n'' places \mathcal{B}_n'' in a $(2N + o(N)) \times (2N + o(N))$ grid \mathcal{M} ;
- \mathcal{L}_n'' places one terminal group of \mathcal{B}_n'' on a vertical side of \mathcal{M} and the other terminal group on a horizontal side of \mathcal{M} .

Proof: Assume, with no loss of generality, that the given layout \mathcal{L}_n'' places the terminal groups on the bottom and right sides; see Figure 3(a). Flip \mathcal{L}_n'' around its right side to produce layout $\widetilde{\mathcal{L}}_n''$ of \mathcal{B}_n'' . Splice layouts \mathcal{L}_n'' and $\widetilde{\mathcal{L}}_n''$ along the pivot line, as depicted in Figure 3(b). By Lemma 1.3, the resulting layout, call it \mathcal{L}'_{n+1} , is a layout of \mathcal{B}'_{n+1} .

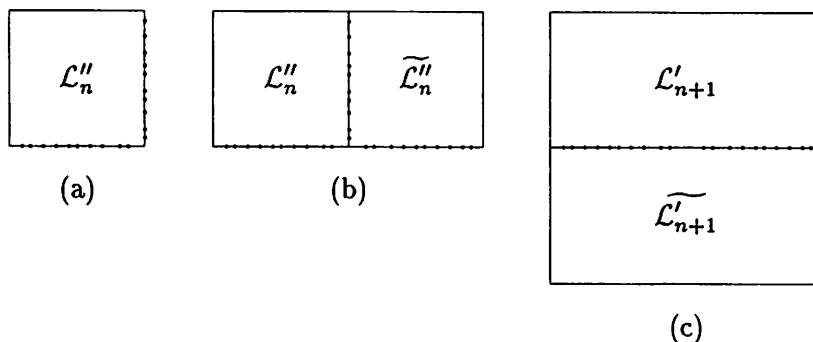


Figure 3: (a) \mathcal{L}_n'' : a layout of \mathcal{B}_n'' ; (b) \mathcal{L}'_{n+1} : a layout of \mathcal{B}'_{n+1} ; (c) \mathcal{L}_{n+2} : a layout of \mathcal{B}_{n+2} .

Next, flip layout \mathcal{L}'_{n+1} around its bottom to produce layout $\widetilde{\mathcal{L}}'_{n+1}$ of \mathcal{B}'_{n+1} . Splice layouts \mathcal{L}'_{n+1} and $\widetilde{\mathcal{L}}'_{n+1}$ along the pivot line to produce the layout \mathcal{L}_{n+2} ; see Figure 3(c). By Lemma 1.3, \mathcal{L}_{n+2} is a layout of \mathcal{B}_{n+2} . (Note the implicit use of Lemma 1.2 here.) Clearly, layout \mathcal{L}_{n+2} resides in a $(4N + o(N)) \times (4N + o(N))$ grid, hence satisfies the conditions of the Theorem. \square

We have thus reduced our layout problem to one of producing a layout \mathcal{L}_n'' , as used in Claim 2.1.

2.2 The Second Reduction

We now show how to construct the layout \mathcal{L}_n'' of Claim 2.1.

Claim 2.2 *Say that we can lay any \mathcal{B}_m'' out in a $(2^{m+1} + o(2^m)) \times o(4^m)$ grid, in such a way that each terminal group of \mathcal{B}_m'' resides on one of the length- $(2^{m+1} + o(2^m))$ (vertical) sides of the grid. Then one can construct the grid-layout \mathcal{L}_n'' of \mathcal{B}_n'' described in Claim 2.1.*

Proof: Let the levels of \mathcal{B}_n'' be numbered $-1, 0, \dots, n, n+1$, where levels (-1) and $(n+1)$ are the terminal groups. We create our layout of \mathcal{B}_n'' in stages.

First, pick any $k \in \{0, 1, \dots, n-1\}$, and construct the graph $\mathcal{B}_n^{(k)}$ from \mathcal{B}_n'' by placing a new node—called a *token*—on every edge connecting levels k and $k+1$ of \mathcal{B}_n'' (or, equivalently, by replacing every such edge by a length-2 path). Note that $\mathcal{B}_n^{(k)}$ has $n+4$ levels, numbered $-1, 0, 1, \dots, n, n+1, n+2$, with the tokens residing in level $k+1$. Clearly, any layout of $\mathcal{B}_n^{(k)}$ is also a layout of \mathcal{B}_n'' .

Next, decompose $\mathcal{B}_n^{(k)}$ along the token-level into

- $\mathcal{G}_{k,1}$: the induced subgraph of $\mathcal{B}_n^{(k)}$ on levels $-1, \dots, k+1$
- $\mathcal{G}_{k,2}$: the induced subgraph of $\mathcal{B}_n^{(k)}$ on levels $k+1, \dots, n+2$.

Easily, one can obtain $\mathcal{B}_n^{(k)}$ by splicing $\mathcal{G}_{k,1}$ and $\mathcal{G}_{k,2}$ along the replicated level. Importantly, by Lemma 1.1, $\mathcal{G}_{k,1}$ is the disjoint sum of 2^{n-k} copies of \mathcal{B}_k'' , while $\mathcal{G}_{k,2}$ is the disjoint sum of 2^{k+1} copies of \mathcal{B}_{n-k-1}'' .

For definiteness, let us now assume that n is odd, and let us consider the graphs $\mathcal{B}_n^{(k)}$, $\mathcal{G}_{k,1}$, and $\mathcal{G}_{k,2}$ when $k = (n-1)/2$. When n is even, we must adjust the details of our layout and its analysis, but only in ways that affect just low-order terms; details are left to the reader. In the case at hand, both $\mathcal{G}_{k,1}$ and $\mathcal{G}_{k,2}$ are disjoint sums of $2^{(n+1)/2}$ disjoint copies of $\mathcal{B}_{(n-1)/2}''$.

Now we are ready to construct the desired layout of \mathcal{B}_n'' from the claimed layout of $\mathcal{B}_{(n-1)/2}''$. To this end, let \mathcal{L} be a layout of $\mathcal{B}_{(n-1)/2}''$ in a $(2^{(n+1)/2} + o(2^{n/2})) \times o(2^n)$ grid, in which the terminal groups reside on opposing vertical sides (of size $2^{(n+1)/2} + o(2^{n/2})$).

We first construct a layout $\mathcal{L}^{(1)}$ of $\mathcal{G}_{(n-1)/2,1}$, by abutting⁵ $2^{(n+1)/2}$ copies of \mathcal{L} , with its token level on the left side, along their (long) horizontal sides. Layout $\mathcal{L}^{(1)}$ resides in a $(2N + o(N)) \times o(N)$ grid.

⁵Note that we are *not* splicing these grids: abutting two copies of the $m \times n$ grid along vertical sides creates a copy of the $m \times 2n$ grid.

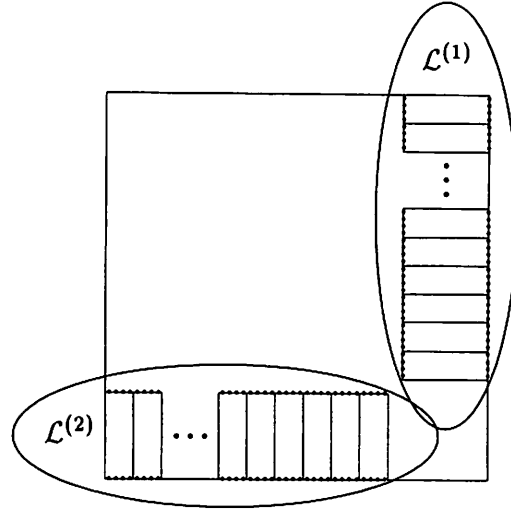


Figure 4: Resplicing $\mathcal{L}^{(1)}$ and $\mathcal{L}^{(2)}$.

Next, we rotate layout $\mathcal{L}^{(1)}$ by 90 degrees to produce $\mathcal{L}^{(2)}$, a layout of $\mathcal{G}_{(n-1)/2,2}$ with the token level on the top side.

As the next to last step, we place layouts $\mathcal{L}^{(1)}$ and $\mathcal{L}^{(2)}$ in the smallest grid \mathcal{M} which will hold them in the following non-overlapping configuration. We position layout $\mathcal{L}^{(1)}$ flush with the top and right sides of \mathcal{M} , and we position layout $\mathcal{L}^{(2)}$ flush with the left and bottom sides of \mathcal{M} ; see Figure 4. One verifies easily that a $(2N + o(N)) \times (2N + o(N))$ grid is large enough to accommodate this placement of layouts $\mathcal{L}^{(1)}$ and $\mathcal{L}^{(2)}$.

Finally, we must splice $\mathcal{G}_{(n-1)/2,1}$ and $\mathcal{G}_{(n-1)/2,2}$ along the token level, to recreate $\mathcal{B}_n^{((n-1)/2)}$. Since all of the nodes to be “merged” have unit degree, we can accomplish the splicing by routing a specific bijection from nodes on the left side of $\mathcal{L}^{(1)}$ to nodes on the top side of $\mathcal{L}^{(2)}$. Our positioning of layouts $\mathcal{L}^{(1)}$ and $\mathcal{L}^{(2)}$ within \mathcal{M} has left a large unpopulated area (as one can see in Figure 4) in which any such bijection can be routed in a cross-bar fashion.

This completes the layout of $\mathcal{B}_n^{((n-1)/2)}$, hence of \mathcal{B}_n'' . □

2.3 The Third Reduction

Our final task is to construct the layouts of \mathcal{B}_n'' demanded in Claim 2.2. We construct these layouts implicitly, by appealing to a result of Pinter [10] on channel routing.

An (h, l, k) *cross-channel routing problem* involves an $h \times l$ grid (the *channel*) and a set of k *two-point nets*: each net is a pair of gridpoints which reside on opposite vertical sides of the grid. The problem is to construct k edge-disjoint paths which connect every net and which can simultaneously be *laid out* in the grid in the sense of Section 1.2.B. Such a layout may not be possible if the grid is too small; Pinter guarantees that the grid need not be too big.

Lemma 2.1 [10] *Any (h, l, k) cross-channel routing problem satisfying $h > k$ and $l > \frac{3}{2}k + 1$ can be routed within the given grid.*

Lemma 2.1 enables the desired layouts of \mathcal{B}_n'' in the following way.

Claim 2.3 *One can lay \mathcal{B}_n'' out in an $(2^{n+1} + 1) \times O(n2^n)$ grid, in such a way that each terminal group of \mathcal{B}_n'' resides on one of the vertical sides.*

Proof: We place each of the $n + 1$ internal levels of nodes of \mathcal{B}_n'' in a $(2^{n+1} + 1) \times (2^n + 2)$ grid, in the staggered fashion depicted in Figure 5. That is, the nodes of a level are placed on grid-points of the form $(2i, i + 1)$, where $i = 1, 2, \dots, 2^n$. (We shall see momentarily that Lemma 2.1 allows us to specify the exact mapping of butterfly nodes to these grid-points in any arbitrary way.)

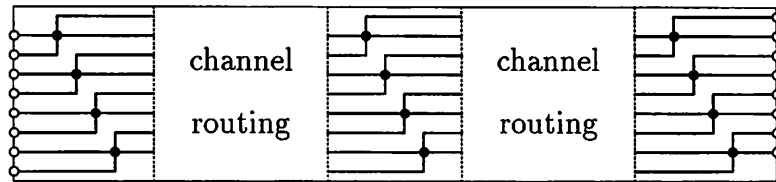


Figure 5: A layout of \mathcal{B}_2'' .

Now, route four edges out of each node to four “terminals,” two on each vertical side of the grid, as depicted in Figure 5. Align the layouts of the $n + 1$ internal levels of \mathcal{B}_n'' horizontally, keeping a space of $\frac{3}{2} 2^{n+1} + 2$ between them, so that we can apply Lemma 2.1. Easily, this produces the claimed layout of \mathcal{B}_n'' in a $(2^{n+1} + 1) \times O(n2^n)$ grid. \square

We now have provided all the machinery necessary to create the layout of Theorem 2.1. The overall layout of \mathcal{B}_n implicit in our proof is depicted in Figure 6; the inputs and outputs of the network are on the middle vertical and horizontal lines, respectively; the shadowed areas contain no butterfly nodes, being dedicated to routing butterfly edges. \square

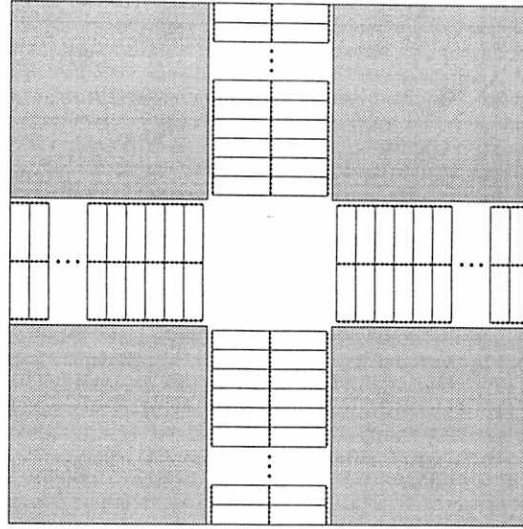


Figure 6: The overall layout of \mathcal{B}_n .

3 The Lower Bound on Layout Area

This section is devoted to proving our lower bound on the layout area of \mathcal{B}_n .

Theorem 3.1 *For all positive integers n , any grid-layout of \mathcal{B}_n has AREA at least $(2^n - 1)^2$. Hence,*

$$\text{AREA}(\mathcal{B}_n) \geq (1 - o(1))N^2.$$

The bounding strategy. We modify the basic lower-bound strategy invented in [13] via the nonstandard notion of a *special-bisection* of a graph.

Let \mathcal{G} be a graph having a designated set of $2c > 0$ *special* nodes. The *minimum special-bisection width* of \mathcal{G} , denoted $\text{MSBW}(\mathcal{G})$, is the smallest number of edges whose removal partitions \mathcal{G} into two disjoint subgraphs, each containing half of \mathcal{G} 's special nodes.

Lemma 3.1 *For any graph \mathcal{G} ,*

$$\text{AREA}(\mathcal{G}) \geq (\text{MSBW}(\mathcal{G}) - 1)^2.$$

Proof: We consider an arbitrary layout of \mathcal{G} in $\mathcal{M}_{m,n}$, where, with no loss of generality, $m \leq n$. As in [13], we find that there is a line L that has a single unit-length jog, which can be positioned on a drawing of $\mathcal{M}_{m,n}$ in the following way.

- L is aligned with the columns of $\mathcal{M}_{m,n}$ in such a way that the portion of L above the jog lies to the left of some column c of $\mathcal{M}_{m,n}$; the jog of L lies below some row of $\mathcal{M}_{m,n}$; the portion of L below the jog either lies outside of $\mathcal{M}_{m,n}$, or it lies to the right of column c .
- Removing the grid-edges crossed by L yields a special-bisection of \mathcal{G} .

By definition, at least $MSBW(\mathcal{G})$ edges of \mathcal{G} must cross line L . By construction, at most $m + 1$ edges of $\mathcal{M}_{m,n}$ cross line L . It follows that $m \geq MSBW(\mathcal{G}) - 1$, whence the lemma. \square

Our goal now is to show that, when we designate the input and output nodes of \mathcal{B}_n as special, we find that $MSBW(\mathcal{B}_n) \geq 2^n$. To this end, we employ $\mathcal{K}_{N,N}$ as an auxiliary graph. First, we note that, if we designate all nodes of $\mathcal{K}_{N,N}$ as special, we obtain the following lower bound on $MSBW(\mathcal{K}_{N,N})$.

Lemma 3.2 $MSBW(\mathcal{K}_{N,N}) = \frac{1}{2}N^2$ when all nodes of $\mathcal{K}_{N,N}$ are special.

Proof: Consider an arbitrary linearization of the nodes of $\mathcal{K}_{N,N}$. Cut the linearization in half. Say that this cut places K input nodes on one side of the cut, hence $N - K$ on the other. Clearly, the output nodes of $\mathcal{K}_{N,N}$ are cut in exactly complementary proportions. Since $\mathcal{K}_{N,N}$ has an edge between every input and every output, the K inputs and K outputs that are separated by the cut give rise to K^2 edges crossing the cut, while the $N - K$ inputs and $N - K$ outputs that are separated by the cut give rise to $(N - K)^2$ cut edges. We thus have $K^2 + (N - K)^2$ edges of $\mathcal{K}_{N,N}$ crossing the cut. This quantity is minimized when $K = \frac{1}{2}N$, in which case $\frac{1}{2}N^2$ edges cross the cut. \square

Congestion arguments. We employ a technique for bounding unknown $MSBW$'s from known ones, which derives from a technique originated in [8] and refined in [11].

The principle underlying congestion arguments is quite simple. Focus on a graph \mathcal{H} which has k special nodes, whose $MSBW$ we wish to bound from below. Say that we have an auxiliary graph \mathcal{G} which has k special nodes, whose $MSBW$ we know. Say further that we have an embedding ϵ of \mathcal{G} into \mathcal{H} , which maps the special nodes of \mathcal{G} onto the special nodes of \mathcal{H} , such that the congestion of ϵ does not exceed C . We claim that $MSBW(\mathcal{H}) \geq (1/C)MSBW(\mathcal{G})$. This inequality holds because the embedding ϵ allows us to view the act of partitioning \mathcal{H} into two disjoint subgraphs having equinumerous sets of special nodes as simultaneously partitioning \mathcal{G} into two disjoint subgraphs having the same partition of special nodes. With this view in mind, one can view the act of

removing any particular edge e of \mathcal{H} as effectively removing all edges of \mathcal{G} that are routed over e by the embedding ϵ . If we know that ϵ never routes more than C edges of \mathcal{G} over any edge of \mathcal{H} , which is what our upper bound on the congestion of ϵ means, then we know that cutting an edge of \mathcal{H} simultaneously cuts no more than C edges of \mathcal{G} . Since we also know that at least $MSBW(\mathcal{G})$ edges of \mathcal{G} must be cut in order to effect the desired partition of \mathcal{G} , we can infer that at least $MSBW(\mathcal{H}) \geq (1/C)MSBW(\mathcal{G})$ edges of \mathcal{H} must be cut in order to effect the desired partition of \mathcal{H} . This argument yields:

Lemma 3.3 (The Congestion Lemma)

Let \mathcal{G} and \mathcal{H} be graphs having equal numbers of special nodes. If there is an embedding of \mathcal{G} into \mathcal{H} which maps special nodes to special nodes and which has congestion $\leq C$, then

$$MSBW(\mathcal{H}) \geq (1/C)MSBW(\mathcal{G}).$$

Our lower bound. We obtain our bound via the congestion technique, by letting $N = 2^n$ and analyzing the “natural” embedding of $\mathcal{K}_{N,N}$ (which plays the role of the guest graph \mathcal{G}) into \mathcal{B}_n (which plays the role of the host graph \mathcal{H}).

Lemma 3.4 *One can embed $\mathcal{K}_{N,N}$ into \mathcal{B}_n with congestion $2^{n-1} = \frac{1}{2}N$, in such a way that the inputs and outputs of $\mathcal{K}_{N,N}$ map, respectively, to the inputs and outputs of \mathcal{B}_n .*

Proof: Note that \mathcal{B}_n has the *banyan* property: each input node u is connected to each output node v by exactly one path of length n . Consider any embedding ϵ of $\mathcal{K}_{N,N}$ into \mathcal{B}_n which assigns inputs of $\mathcal{K}_{N,N}$ to inputs of \mathcal{B}_n and outputs of $\mathcal{K}_{N,N}$ to outputs of \mathcal{B}_n , and which routes the edges of $\mathcal{K}_{N,N}$ “greedily”, via the unique length- n path which connects the two end-points in \mathcal{B}_n .

We now analyze the congestion of embedding ϵ . Call a path of \mathcal{B}_n *simple* if it does not visit any level twice. Let e be a level- k edge of \mathcal{B}_n . Since \mathcal{B}_n has the banyan property, one endpoint of e reaches precisely 2^{n-k-1} distinct output nodes via simple paths, while the other endpoint of e reaches precisely 2^k distinct input nodes via simple paths. Hence, edge e lies on precisely 2^{n-1} input-to-output simple paths; i.e., its congestion is precisely 2^{n-1} . \square

If we now designate all nodes of $\mathcal{K}_{N,N}$ as special and the input and output nodes of \mathcal{B}_n as special, we infer from Lemmas 3.2, 3.3, and 3.4 a lower bound on the $MSBW$ of \mathcal{B}_n .

Lemma 3.5 $MSBW(\mathcal{B}_n) \geq 2^n$.

Finally, Lemma 3.5 combines with Lemma 3.1 to yield the desired lower bound, Theorem 3.1, on the area of grid-layouts of \mathcal{B}_n . \square

References

- [1] A.V. Aho, J.E. Hopcroft, J.D. Ullman (1974): *The Design and Analysis of Computer Algorithms*. Addison-Wesley, Reading, Mass.
- [2] F. Berman and L. Snyder (1987): On mapping parallel algorithms into parallel architectures. *J. Parallel Distr. Comput.* 4, 439-458.
- [3] S.N. Bhatt, F.R.K. Chung, J.-W. Hong, F.T. Leighton, B. Obrenić, A.L. Rosenberg, E.J. Schwabe (1996): Optimal emulations by butterfly-like networks. *J. ACM*, to appear.
- [4] S.N. Bhatt and F.T. Leighton (1984): A framework for solving VLSI graph layout problems. *J. Comp. Syst. Scis.* 28, 300-343.
- [5] S.H. Bokhari (1981): On the mapping problem. *IEEE Trans. Comp.*, C-30, 207-214.
- [6] Ye. Dinitz (1995): A compact layout of butterfly on the square grid. Tech. Rpt. 873, TheTechnion.
- [7] S. Even and A. Litman (1992): Layered cross product — a technique to construct interconnection networks. *4th ACM Symp. on Parallel Algorithms and Architectures*, 60-69.
- [8] F.T. Leighton (1983): *Complexity Issues in VLSI: Optimal Layouts for the Shuffle-Exchange Graph and Other Networks*. MIT Press, Cambridge, Mass.
- [9] K. Mehlhorn, F.P. Preparata, M. Sarrafzadeh (1986): Channel routing in knock-knee mode: simplified algorithms and proofs. *Algorithmica* 1, 213-221.
- [10] R.Y. Pinter (1982): On routing two-point nets across a channel. *19th ACM-IEEE Design Automation Conf.*, 894-902.
- [11] A.L. Rosenberg and L.S. Heath (1996): *Graph Separators, with Applications*, in preparation.
- [12] L. Snyder (1986): Type architectures, shared memory, and the corollary of modest potential. *Ann. Rev. Computer Science* 1, 289-317.
- [13] C.D. Thompson (1980): *A Complexity Theory for VLSI*. Ph.D. Thesis, CMU.
- [14] C.D. Thompson (1983): Fourier transforms in VLSI. *IEEE Trans. Comp.*, C-32, 1047-1057.
- [15] D.S. Wise (1981): Compact layouts of banyan/FFT networks. *VLSI Systems and Computations* (H.T. Kung, B. Sproull, G. Steele, eds.) Computer Science Press, Rockville, Md., 186-195.