Efficient VLSI Layouts of Hypercubic Networks

Chi-Hsiang Yeh, Emmanouel A. Varvarigos, and Behrooz Parhami
Department of Electrical and Computer Engineering
University of California, Santa Barbara, CA 93106-9560, USA

Abstract

In this paper, we present efficient VLSI layouts of several hypercubic networks. We show that an N-node hypercube and an N-node cube-connected cycles (CCC) graph can be laid out in $4N^2/9 + o(N^2)$ *and* $4N^2/(9\log_2^2 N) + o(N^2/\log^2 N)$ *areas, respectively, both of which are optimal within a factor of* $1.\bar{7} + o(1)$. *We introduce the multilayer grid model, and present efficient layouts of hypercubes that use more than 2 layers of wires. We derive efficient layouts for butterfly networks, generalized hypercubes, hierarchical swapped networks, and indirect swapped networks, that are optimal within a factor of* $1 + o(1)$. *We also present efficient layouts for folded hypercubes, reduced hypercubes, recursive hierarchical swapped networks, and enhanced-cubes, which are the best results reported for these networks thus far.*

1. Introduction

The derivation of efficient VLSI layouts for interconnection networks is important, since it improves the cost-performance of the resulting parallel architecture, both by reducing its cost (fewer chips, boards, and assemblies) and by lowering various performance hindrances, such as signal propagation delay, drive power, and fraction of data transfers to off-chip destinations. Efficient layouts for several interconnection networks can be found in [5, 6, 8, 12].

Hypercubes, butterfly networks [13], and cube-connected cycles (CCC) [16] are among the most important interconnection networks. In [6], a collinear layout of an N-node hypercube that requires $N - \log_2 N$ tracks was proposed. In this paper, we show that the collinear layout of a hypercube can be considerably improved to one that uses $\lfloor 2N/3 \rfloor$ tracks, which is within a factor of $1.\bar{3} + o(1)$ from a trivial lower bound. We also show that an N-node hypercube can be laid out in $4N^2/9 + o(N^2)$ area, which is within a factor of $1.\bar{7} + o(1)$ from a trivial lower bound and improves the layout area given in [6] by a factor of $2.25 + o(1)$. We also show that an N-node CCC can be laid out in $4N^2/(9\log_2^2 N) + o(N^2/\log^2 N)$ area, which is smaller than the layout area given in [7] by a factor of $1.125 + o(1)$, and is within a factor of $1.\bar{7} + o(1)$ from a lower bound. The layouts for the hypercube and CCC given in this paper are the best results reported thus far for these networks.

We introduce the multilayer 2-D grid and multilayer 3-D grid models for VLSI layouts of networks. The motivations for using the multilayer grid model include significant reduction in layout area, volume, and maximum wire length. In particular, we show that an N-node hypercube can be laid out in $\frac{16N^2}{9L^2} + o\left(\frac{N^2}{L^2}\right)$ area, $\frac{16N^2}{9L} + o\left(\frac{N^2}{L}\right)$ volume, and $\frac{2N}{3L} + o\left(\frac{N}{L}\right)$ maximum wire length when we use L layers of wires, L is even, and $L = o(\sqrt{N}/\log N)$.

We derive tight bounds on the VLSI area of generalized hypercubes, hierarchical swapped networks (HSNs) [23, 25], and indirect swapped networks (ISNs) [22], which are optimal within a factor of $1 + o(1)$. Moreover, we present efficient layouts for butterfly networks [13], folded hypercubes [1], reduced hypercubes (RH) [29], recursive hierarchical swapped networks (RHSNs) [23, 25], and enhanced-cubes [21], which are the best results reported for these networks so far in the literature. Our layout method and lower bound techniques can also be extended to a variety of other networks [25, 28].

The organization of the remainder of the paper is the following. In Section 2, we present efficient layouts for hypercubes, folded hypercubes, CCC, and reduced hypercubes. In Section 3, we introduce the multilayer grid model and present multilayer layouts of hypercubes. In Section 4, we present efficient layouts for butterfly networks, generalized hypercubes, HSNs, RHSNs, and ISNs. In Section 5, we show that several of the layouts given in Section 4 have areas that are optimal within a factor of $1 + o(1)$. In Section 6 we present our conclusions.

2. VLSI layouts for hypercubes, CCCs, and related networks

In this section, we present a method for laying out hypercubes, folded hypercubes, CCC, and reduced hypercubes.

We use the extended version [8, 17, 25] of the grid model, also called Thompson's model [18], for the VLSI layout of networks whose node degrees may be larger than 4. In this model, a network is viewed as a graph whose nodes correspond to processing elements and edges correspond to wires. The graph is then embedded in a 2-D grid, where wires have unit width and a node of degree d occupies a square of side d. The wires can run either horizontally or vertically along grid lines.

The area of a layout is defined as the area of the smallest rectangle that contains all the nodes and wires. When there are two layers of wires, it is guaranteed that we can lay out the network within the area. In Section 3, we modify layouts derived in this section to obtain layouts that use more than two layers of wires and have smaller area

2.1. Efficient layouts for hypercubes and several variants

In this subsection, we first derive a collinear layout for the hypercube and then use it to obtain efficient 2-D layouts for hypercubes, folded hypercubes, and their variants.

In a collinear layout all nodes are placed on the same line. A collinear layout that requires $N - \log_2 N$ tracks was presented in [6] for an N-node hypercube. In what follows, we improve on their result by finding a collinear layout that uses only $\lfloor 2N/3 \rfloor$ tracks.

To describe the hypercube layout we use a bottom-up approach, starting with a 2-dimensional hypercube, and inductively moving to hypercubes of higher dimensions. A collinear layout of a 2-dimensional hypercube can be obtained by placing the 4 nodes along a row, connecting node 0 with node 1, and node 1 with node 3, through wires in the first track, and then connecting node 0 with node 2, and node 2 with node 3, through wires in the second track (see Fig. 1a). Clearly, this layout requires 2 tracks.

Assume that we have a collinear layout for an n-cube that requires $f(n)$ tracks, where n is even. To obtain the collinear layout of an $(n+1)$-cube, we start with the layouts of two n-cubes. By doubling the horizontal space, we can place the i^{th} node of the second layout adjacent (from the right) to the i^{th} node of the first layout. We also double the number of tracks (i.e., vertical space) to accommodate the $2f(n)$ tracks of the two layouts. Moreover, to connect the two n-cubes into an $(n+1)$-cube, we need an extra track which contains paths connecting adjacent nodes (i.e., the i^{th} nodes of the two layouts). Therefore, the number of tracks required for the collinear layout of the $(n+1)$-dimensional hypercube is $f(n+1) = 2f(n)+1$, assuming that n is even.

To obtain the collinear layout of an $(n+2)$-cube, we start with the layouts of four n-cubes. By increasing the horizontal space by a factor of 4, we can place the nodes with the same ID of the four layouts adjacent to each other. We also have to increase the number of tracks by a factor of 4 to accommodate the $4f(n)$ tracks of the four layouts. Finally, to connect the four n-cubes into an $(n+2)$-cube, we need two extra tracks for laying out the paths that connects each set of 4 nodes of the n-cubes that have the same ID as a 2-cube (see Fig. 1b). Therefore, we have $f(n+2) = 4f(n)+2$ when n is even. Since $f(2) = 2$, we obtain the following theorem.

Theorem 2.1 *The number of tracks required for the collinear layout of an N-node hypercube is* $\lfloor \frac{2N}{3} \rfloor$.

Proof: When n is even, we have

$$f(n) = 4f(n-2)+2$$

and $f(0) = 0$, where $n = \log_2 N$. Therefore,

$$f(n) = 2^2 f(n-2) + 2^1 = 2^4 f(n-4) + 2^3 + 2^1$$
$$= 2^n f(0) + 2^{n-1} + 2^{n-3} + \cdots + 2^1$$
$$= \frac{N}{2}\left(1 + \frac{1}{4} + \frac{1}{16} + \cdots + \frac{4}{N}\right) = \frac{2}{3}(N-1) = \left\lfloor \frac{2N}{3} \right\rfloor$$

when $n = \log_2 N$ is even. For the case where n is odd, we have

$$f(n) = 2f(n-1)+1 = \frac{4}{3}\left(\frac{N}{2} - 1\right) + 1 = \frac{2}{3}N - \frac{1}{3} = \left\lfloor \frac{2N}{3} \right\rfloor.$$

□

To lay out an n-cube on a 2-D grid, we let $n = n_1 + n_2$, and use 2^{n_1} copies of the collinear layout of an n_2-cube, each placed along a row. We then connect the 2^{n_1} nodes that belong to the same column (i.e., nodes that have the same ID within each of the n_2-cubes) vertically according to the collinear layout of an n_1-cube (see Fig. 1c). Note that when n_2 (and/or n_1) is odd, we can eliminate 2^{n_1} horizontal tracks (and/or 2^{n_1} vertical tracks, respectively) by moving the wires connecting neighboring nodes to horizontal tracks (and/or vertical tracks, respectively) between nodes. When n_2 (and/or n_1) is even, we can also remove 2^{n_1} horizontal tracks (and/or 2^{n_1} vertical tracks, respectively) after some minor modifications at the expense of longer wires. Since in the VLSI model a node of degree $\log_2 N$ requires a square of side $\log_2 N$, we need an extra $O(N\sqrt{N}\log N)$ area to accommodate the nodes. By choosing $n_1 = \Theta(n_2)$, we obtain the following theorem.

Theorem 2.2 *An N-node hypercube can be laid out in* $\frac{4}{9}N^2 + o(N^2)$ *area.*

The layout area given in Theorem 2.2 for the hypercube improves the corresponding area given in [6] by a factor of $2.25 + o(1)$, and is the best result reported thus far for hypercubes. The area is within a factor of $1.\bar{7} + o(1)$ from a trivial

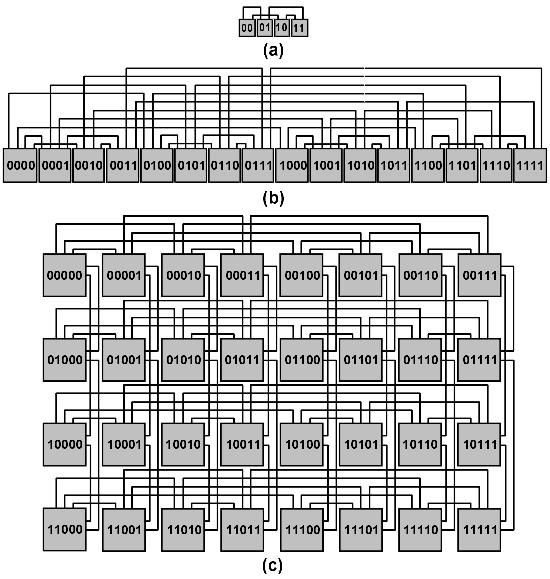

Figure 1. collinear layouts of (a) a 2-cube and (b) a 4-cube. (c) A 2-D layout of a 5-cube.

lower bound $N^2/4$ [which follows from the fact that the area of a graph is at least equal to B^2 [19, 25], where $B = N/2$ is the bisection width of a hypercube, or from Lemma 5.1 of Section 5 since a total exchange task in a hypercube requires $N/2$ steps [20]. The proposed hypercube layout has maximum wire length $N/3 + o(N)$, which is (slightly) shorter than the best previous result [10] for hypercubes of (currently) practical sizes (e.g., $N \leq 2^{14} = 16K$), and has smaller area by a factor of $2.25 + o(1)$ at the same time. Note that we can move the longer wires (and other wires belong to the same tracks) to the first 2^{n_2-1} horizontal tracks (or the first 2^{n_1-1} vertical tracks, respectively) in order to (slightly) reduce the maximum wire length.

An enhanced-cube is a hypercube with one additional outgoing link per node leading to a random node [21]. A folded hypercube [1] is a hypercube with one additional link per node, where each node S has a link connecting it to the node whose label is the bitwise complement of S. By adding additional links to the hypercube layout of Theorem 2.2, we can lay out a folded hypercube in $\frac{49}{36}N^2 + o(N^2)$ area, and an enhanced-cube in $\frac{25}{9}N^2 + o(N^2)$ area. More precisely, we first lay out an N-node hypercube in a square of side $\frac{2}{3}N + o(N)$. To lay out an additional link, we need at most an additional vertical track and an additional horizontal track, in addition to the two ending segments connecting the link to two nodes.

Since there are $N/2$ additional links in a folded hypercube, we need at most $N/2$ extra vertical and horizontal tracks to accommodate all the diameter links. Therefore, the area for the layout of a folded hypercube is

$$\left(\frac{7}{6}N + o(N)\right) \times \left(\frac{7}{6}N + o(N)\right) = \frac{49}{36}N^2 + o(N^2).$$

Since there are N additional links in an enhanced-cube, we need at most N vertical and horizontal tracks to accommodate all the additional links. Therefore, the area for the layout of an enhanced-cube is $\frac{25}{9}N^2 + o(N^2)$. □

The preceding layouts for folded hypercubes and enhanced-cubes improve the areas of the corresponding layouts given in [21] by constant factors.

2.2. Efficient layouts for CCC and reduced hypercubes

An n-dimensional cube-connected cycles (CCC) graph is obtained by replacing each node in an n-cube with an n-node cycle [16]. A reduced hypercube, $\mathrm{RH}(\log_2 n, \log_2 n)$ [29], can be obtained by replacing each n-node cycle in a CCC with a $\log_2 n$-dimensional hypercube.

Theorem 2.3 *An N-node CCC or $RH(\log_2 n, \log_2 n)$ can be laid out in*

$$\frac{4N^2}{9\log_2^2 N} + o\left(\frac{N^2}{\log^2 N}\right)$$

area.

Proof: We first lay out an n-cube using the 2-D layout introduced in Subsection 2.1, and then lay out the n-node cycles within each of the hypercube nodes. Since the size of an n-dimensional CCC is $N = n2^n$ and its area is dominated by its hypercube links, which requires $2^{n+2}/9 + o(2^n)$ area, an N-node CCC can be laid out in

$$\frac{4N^2}{9\log_2^2 N} + o\left(\frac{N^2}{\log^2 N}\right)$$

area. Using the same layout method, the reduced hypercube can be laid out in asymptotically the same area. □

In [16], layouts of area $\frac{2N^2}{\log_2^2 N} + o\left(\frac{N^2}{\log^2 N}\right)$ and $\frac{4N^2}{3\log_2^2 N} + o\left(\frac{N^2}{\log^2 N}\right)$ were proposed for the CCC graph. Our layout has area smaller than that of the layouts given in [16] by a factor of at least $3 + o(1)$, and smaller than that of the more recent layouts given in [7] by a factor of $1.125 + o(1)$. The layout area given in Theorem 2.3 is within a factor of $1.\bar{7} + o(1)$ from the lower bound given in [7] and is the best result reported thus far for the CCC network.

3. Layouts under the multilayer grid model

In this section, we introduce the *multilayer grid model* for VLSI layouts of networks. We then derive efficient multilayer layouts for hypercubes.

3.1. The multilayer grid model

In the multilayer grid model, a network is viewed as a graph whose nodes correspond to processing elements and edges correspond to wires. The nodes and edges of the graph are then embedded in a 3-D grid, where edges have unit width, can run along grid lines, but cannot cross each other (i.e., the paths for embedding these edges must be edge- and node-disjoint). The area A of a layout is defined as the area of the smallest rectangle along the x-y directions that contains all the nodes and wires. The volume of a layout is equal to the number L of layers times its area A.

In the multilayer 2-D grid model, the nodes of the graph are embedded in the 2-D grid of the first layer (i.e., $z = 1$), where a node of degree d occupies a square of side d. Note that a network with area A under the extended grid model can be laid out with area no larger than A under the multilayer 2-D grid model with $L = 2$ layers. In the multilayer 3-D grid model, the nodes of the graph are embedded in L_A layers of the 3-D grid, where a node of degree d occupies a $d/h \times d/h \times h$ cuboid and $1 \leq h \leq L_A \leq L$. These L_A layers are called "active layers." The multilayer 2-D grid model is a special case of the multilayer 3-D grid model with $L_A = 1$ active layer. Note that a $d/h \times d/h \times h$ cuboid node requires h active layers for its implementation, while a $d \times d \times 1$ cuboid node requires only 1 active layer. Layouts designed for these models can be easily modified to obtain layouts with different assumptions on the size of nodes. The cost of a layout under the multilayer grid model is a function of A, L, and L_A.

The motivations for using the multilayer grid model are three folds: (1) some technologies can lay out wires using more than 2 layers, leading to significant reduction in layout area; (2) the volume of the layouts of many networks may be reduced by a factor of approximately $L/2$ compared to their layouts under the grid model; and (3) the maximum length of wires in many networks may be reduced by a factor of approximately $L/2$. When we use L layers, the number of tracks in x and y directions may both be reduced by a factor of about $L/2$ in many networks, for a factor of $L^2/4$ reduction in its area compared to the layout under the grid model, while the number of layers is only increased by a factor of $L/2$. This leads to items (1) and (2) and, therefore, the cost of the resultant layout can be significantly reduced. As a comparison, if we fold a layout derived for the grid model in order to use all the available layers, the area can be reduced by a factor of only $L/2$ and the volume cannot be reduced; if we extended the collinear layout model to its multilayer version, the volume cannot be reduced either since the area can only be reduced by a factor of at most $L/2$ when L layers are used. The maximum wire lengths in many networks are approximately proportional to the number of tracks in x or y direction (or their sum). Therefore, if the numbers of tracks in x and y directions are both reduced by a factor of about $L/2$, the maximum wire length can also be reduced by a factor of approximately $L/2$, leading to significant improvement in performance [item (3)]. As a comparison, the maximum wire length in a collinear layout using L layers or in a layout obtained by folding the layout derived using the grid model remains similar in most cases. These arguments will become clear by looking at the multilayer layouts derived in the following subsections.

We can extend the multilayer grid model to the *multilayer layout model* by allowing nodes and edges to run in other specified directions. Layouts under this model may have smaller area and volume compared with layouts under its multilayer grid model counterpart. Moreover, wires in this model may have different width and cross area, depending on the technology used. For example, wires along the z direction may have larger cross area in PCB. In what follows, we focus on the multilayer 2-D grid model. Layouts under other models will be reported in the near future.

3.2. The layout area and volume of hypercubes under the multilayer grid model

In this subsection we present efficient multilayer layouts for hypercubes.

We first derive hypercube layouts with an even number L of layers. The multilayer 2-D grid layout of a hypercube can be obtained from its 2-D grid layout by partitioning all the horizontal (resp., vertical) tracks above each row (resp., to the right of each column) of nodes into $L/2$ groups, each of which has at most $k_1 = \lceil \frac{2\lfloor 2^{n_2+1}/3 \rfloor}{L} \rceil$ (or $\lceil \frac{2\lfloor 2^{n_2+1}/3 \rfloor - 1}{L} \rceil$ if n_2 is odd) horizontal tracks [resp., $k_2 = \lceil \frac{2\lfloor 2^{n_1+1}/3 \rfloor}{L} \rceil$ (or $\lceil \frac{2\lfloor 2^{n_1+1}/3 \rfloor - 1}{L} \rceil$ if n_1 is odd) vertical tracks] and is wired using two layers. More precisely, the vertical segments connecting the horizontal tracks of groups i (above each row) and the vertical tracks of groups i (to the right of each column) are wired using layer $2i-1$, and the horizontal tracks of groups i and the horizontal segments connecting the vertical tracks of groups i are wired using layer $2i$, for $i = 1, 2, \ldots, L/2$. When a link makes a turn in the 2-D grid layout, its vertical and horizontal segments, wired in neighboring layers $i-1$ and i in the multilayer layout, should be connected by a wire (or via) along the z direction.

When $L = o(\sqrt{N}/\log N)$, the area of the resultant L-layer layout can be reduced from $4N^2/9 + o(N^2)$ under the grid model to

$$\frac{16N^2}{9L^2} + o\left(\frac{N^2}{L^2}\right);$$

the maximum wire length of the L-layer layout can be reduced from $N/3 + o(N)$ to

$$\frac{2N}{3L} + o\left(\frac{N}{L}\right);$$

the total wire length of a routing path is $1.\bar{3}N/L + o(N/L)$; and the volume of the L-layer layout can be reduced from $8N^2/9 + o(N^2)$ (assuming wires cannot cross each other) to

$$\frac{16N^2}{9L} + o\left(\frac{N^2}{L}\right).$$

When L is odd, we simply partition horizontal tracks into $(L+1)/2$ groups, wired on layers $1, 3, \ldots, L$, and partition vertical tracks into $(L-1)/2$ groups, wired on layers $2, 4, \ldots, L-1$. We can also partition and wire them the other way around. The area of the resultant layout is

$$\frac{16N^2}{9(L^2-1)} + o\left(\frac{N^2}{L^2}\right)$$

when L is odd and $L = o(\sqrt{N}/\log N)$.

These multilayer hypercube layouts are the best results reported in the literature thus far for $L = 2, 3, \ldots, o(\sqrt{N}/\log N)$ in terms of area and volume. Since we have obtained area-efficient L-layer layouts for hypercubes, $L = 2, 3, \ldots, o(\sqrt{N}/\log N)$, we can optimize the cost for implementation by minimizing the cost function $f(A, L, L_A = 1)$.

If a large number $L = \Omega(\sqrt{N}/\log N)$ of layers are available and more than one active layer is available, we can design hypercube layouts under the multilayer 3-D grid model to further reduce the layout area, maximum wire length, and volume. To obtain multilayer 3-D layouts for an $(n_1 + n_2 + n_3)$-cube, we simply use 2^{n_3} copies of a multilayer 2-D $(n_1 + n_2)$-cube layout, and connect nodes belonging to the same grid point in a way similar to a collinear layout of an n_3-cube. More details will be reported in the near future.

4. VLSI layouts for several networks

In this section, we present efficient layouts for several interconnection networks under the grid model.

4.1. Efficient layouts for generalized hypercubes, HSNs and RHSNs

An l-level hierarchical swapped network, denoted by HSN(l, G) [23, 25], is an l-level network consisting of M level-l clusters, each of which is an HSN$(l-1, G)$ network, where M is the number of nodes in the nucleus G. Each of the M^{l-2} nuclei of a level-l cluster has a link connecting it to each of the other $M-1$ level-l clusters. If we view a level-l cluster as a supernode, the HSN(l, G) becomes an M-supernode complete graph with N/M^2 edges between each pair of supernodes.

Theorem 4.1 *An N-node HSN(l, G) can be laid out using $N^2/16 + o(N^2)$ area if*

(a) *$l = 2$ and the nucleus G can be laid out in a square of side $o(M^{\frac{3}{2}})$, or*
(b) *$l = 3$ and the nucleus G can be laid out in a square of side $o(M^2)$, or*
(c) *$l \geq 4$,*

assuming that M, the size of the nucleus G, is not a constant.

Proof: The inter-cluster links between top-level clusters can be laid out in $N^2/16 + o(N^2)$ area using the layout of an M-node complete graph [25, 27] with multiple edges. When one of the conditions holds, the area for all nuclei does not affect the leading constant of the layout area and the required area is dominated by the top-level inter-cluster links. □

An r-deep recursive hierarchical swapped network (abbreviated RHSN) [23, 25] is obtained by recursively replacing the nucleus of an HSN with an $(r-1)$-deep RHSN. More precisely, RHSN$(l_r, l_{r-1}, \ldots, l_1, G)$ = HSN$(l_r,$ RHSN$(l_{r-1}, l_{r-2}, \ldots, l_1, G))$. Therefore, RHSN can be laid out by recursively laying out HSNs.

Theorem 4.2 *An N-node RHSN$(l_r, l_{r-1}, \ldots, l_1, G)$ can be laid out using $N^2/16 + o(N^2)$ area, assuming that the depth r is at least 2 and the number of nodes in an RHSN$(l_{r-1}, l_{r-2}, \ldots, l_1, G)$ is not a constant. (In other words, at least one of the parameters r, M, and l_i for any $i \leq r-1$ is not constant, where M is the size of the nucleus G.)*

By shrinking all the nuclei of an HSN into a node, we obtain a radix-M generalized hypercube [4, 11]. This combined with Theorem 4.1 leads to the following theorem for the layout of high-radix hypercubes.

Theorem 4.3 *A radix-M generalized hypercube can be laid out using $M^2N^2/16 + o(M^2N^2)$ area, assuming that M is not a constant.*

The above layout can be easily extended to general mixed-radix generalized hypercubes [4]. As will be shown in Section 5, the proposed layouts for generalized hypercubes and HSNs are optimal within a factor of $1 + o(1)$.

4.2. Optimal layouts for butterfly networks and indirect swapped networks (ISNs)

In this subsection we present efficient layouts for butterfly networks and indirect swapped networks (ISNs) [22]). A butterfly network [13] is obtained by unfolding the structure of a hypercube along routing paths, while an indirect swapped network (ISN) (also called an unfolded swapped network (USN) [22]) is a multistage network obtained by unfolding the structure of a swapped network [23, 25]. We first present optimal layouts for ISNs and then use them to derive optimal layouts for butterfly networks.

Theorem 4.4 *An N-node ISN can be laid out in*

$$\frac{N^2}{4\log_2^2 N} + o\left(\frac{N^2}{\log^2 N}\right)$$

area, assuming that the number M_l of top-level clusters in the corresponding swapped network (which is unfolded to generate the ISN) is not a constant.

Proof: If we place every M_l rows of the ISN into the same top-level block [25, 28], then each pair of the blocks are connected by 2 links. Therefore, we can lay out the inter-cluster links using the layout of an $(\frac{N}{M_l \log_2 N} + o(\frac{N}{M_l \log N}))$-node complete graph with multiple edges, which requires

$$\frac{N^2}{4\log_2^2 N} + o\left(\frac{N^2}{\log^2 N}\right)$$

area [25, 27]. $\square$

The layout area for the ISN improves the corresponding result given in [22] by a factor of $4 + o(1)$.

The following theorem gives a layout for the butterfly network that is optimal within a factor of $1 + o(1)$ from the lower bound given in [2].

Theorem 4.5 *An N-node butterfly network can be laid out in*

$$\frac{N^2}{\log_2^2 N} + o\left(\frac{N^2}{\log^2 N}\right)$$

area.

Proof: If we unfold an HSN(2, $\frac{\log_2 N}{2}$-cube), we obtain a $(\log_2 N + 2)$-stage ISN that uses $\frac{\log_2 N}{2}$-dimensional butterfly networks as the basic modules. If we double up the links connecting the middle two stages of the ISN, remove nodes in the $(\frac{\log_2 N}{2} + 2)$-th stage, and reconnect each of the replicated links to one of the two links between the $(\frac{\log_2 N}{2} + 2)$-th and the $(\frac{\log_2 N}{2} + 3)$-th stage through a removed node, we can obtain an automorphism of a $(\log_2 N)$-dimensional butterfly

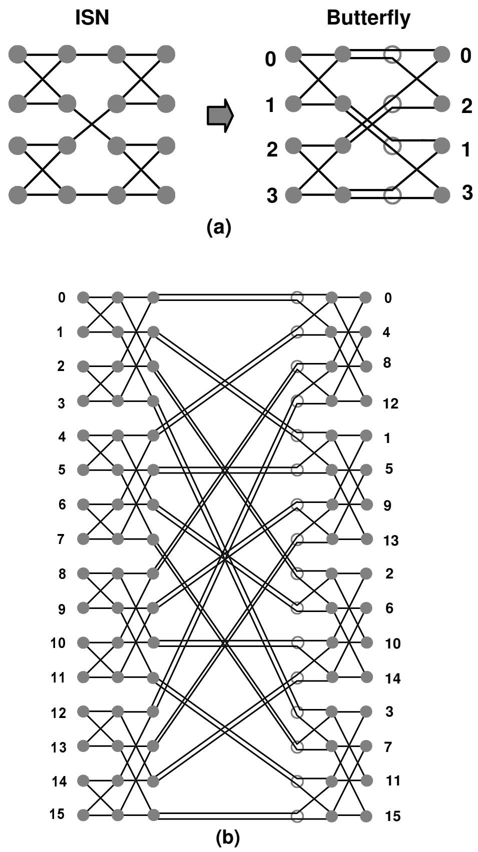

Figure 2. Deriving butterfly networks from indirect swapped networks. (a) Transforming a 4×4 ISN into a 4×4 butterfly network. (b) A resultant 16×16 butterfly network.

(see Fig. 2). Therefore, the area of the butterfly is approximately 4 times that of an ISN; that is

$$\frac{N^2}{\log_2^2 N} + o\left(\frac{N^2}{\log^2 N}\right).$$

$\square$

In [2] the same upper bound for the area of a butterfly network has been presented. The proof given in [2] is, however, considerably more complicated than our construction. It is also interesting that butterfly networks can be laid out based on the layout of a complete graph.

Generalized hypercubes, HSN(l, Q_n) with $l > 2$, and ISNs unfolded from such HSNs can be laid out based on the collinear layouts of complete graphs rather than the 2-D layouts of complete graphs. Similar to the multilayer layouts of hypercubes, the resultant layouts for these networks can be easily partitioned and wired using $L > 2$ layers. The resultant maximum wire lengths in these layouts can be reduced by a factor of approximately 2 compared to the layouts presented in this paper and the areas are also slightly reduced. Similar to Theorem 4.5 (see Fig. 2), multilayer layouts for butterfly networks can be obtained by modifying the multilayer layouts for ISNs. More details will be reported in the near future.

5. Tight bounds on the VLSI areas of several networks

In this section, we derive tight bounds on the areas of several networks under the grid model.

The total exchange (TE) task [3, 9] (also called all-to-all personalized communication) is a basic communication task that arises often in applications, where each node has to send a different (personalized) packet to every other node of the network. In [25], we have shown the following lemma concerning the relationship between the VLSI area of a network and the throughput for performing TE tasks in it.

Lemma 5.1 *Assume that $f(N)$ total exchange (TE) tasks can be executed in $f(N)T_{TE}$ communication steps in an N-node interconnection network for some integer function $f(N)$, under the all-port communication model. Then the layout area of the network is at least equal to*

$$\frac{\lfloor N/2\rfloor^2 \times \lceil N/2\rceil^2}{T_{TE}^2} \approx \frac{N^4}{16T_{TE}^2}.$$

When performing the TE tasks we assume that links are bidirectional and nodes can have infinitely large routing tables and buffer space and perform infinitely many computation steps if required. (Links are still assumed to have unit width in the layout.)

In what follows we show that several of the layouts presented in Section 4 have areas that are optimal within a factor of $1+o(1)$.

Theorem 5.2 *The area of the minimal layout of a radix-M generalized hypercube is equal to $M^2N^2/16 \pm o(M^2N^2)$, assuming that M is not a constant, where N is the number of nodes in the network.*

Proof: The upper bound is given in Theorem 4.3. A lower bound on its VLSI area is given by

$$\frac{(M-1)^2n^2N^2}{16n^2} = \frac{M^2N^2}{16} - o(M^2N^2)$$

from Lemma 5.1 and the fact that n TE tasks can be performed in nN/M steps in an n-dimensional radix-M hypercube. □

The throughput for performing TE tasks in HSNs is given in the following lemma [25].

Lemma 5.3 *The throughput for executing TE tasks in an N-node HSN(l,G) can be arbitrarily close to $1/N$, provided that the nucleus G can execute l TE tasks in M time steps under the all-port communication model, where M is the number of nodes in G.*

By combining Lemma 5.3 with Theorems 4.1 and 5.1, we can prove that the layout for HSNs is also close to being strictly optimal.

Theorem 5.4 *The area of the minimal layout of an N-node HSN(l,G) is equal to $N^2/16+o(N^2)$ if the nucleus G can execute l TE tasks in M time steps under the all-port communication model and*

- (a) *$l=2$ and the nucleus G can be laid out in a square of side $o(M^{\frac{3}{2}})$, or*
- (b) *$l=3$ and the nucleus G can be laid out in a square of side $o(M^2)$, or*
- (c) *$l \geq 4$,*

assuming that M, the size of a nucleus G, is not a constant.

In Section 4 we derived optimal layouts for butterfly networks based on the layouts of ISNs. In what follows, we show that the lower bound on the VLSI area of a butterfly network given in [2] can be used to derive a lower bound on the area of an ISN.

Theorem 5.5 *The area of the minimal layout of an N-node ISN is equal to $\frac{N^2}{4\log_2^2 N} \pm o(\frac{N^2}{\log^2 N})$, assuming the nucleus of the ISN is a butterfly network.*

Proof: From Theorem 4.5, we can see that if it were possible to lay out an ISN in $\frac{(1-\varepsilon)N^2}{4\log_2^2 N}$ area, then it would also be possible to lay out a butterfly network in $\frac{(1-\varepsilon)N^2}{\log_2^2 N} + o(\frac{N^2}{\log^2 N})$ area, where ε is a positive constant. This contradicts the lower bound given in [2]. Therefore, the area of an ISN is at least $\frac{N^2}{4\log_2^2 N} - o(\frac{N^2}{\log^2 N})$. The upper bound is given in Theorem 4.4. □

Theorem 5.5 can be generalized to ISNs that are based on other nuclei that contain a butterfly network of the same size as a subgraph.

By using the techniques introduced, we can also obtain tight bounds on the bisection widths of the networks investigated in this paper and efficient layouts for many other networks, such as macro-star networks [26] periodically regular chordal rings [14, 15], and cyclic networks [24]. Some examples can be found in [25, 28].

6. Conclusion

We derived layouts for butterfly networks, generalized hypercubes, HSNs, and ISNs that are optimal within a factor of $1+o(1)$ under the grid model. We presented efficient layouts for hypercubes, CCCs, folded hypercubes, reduced hypercubes, RHSNs, and enhanced-cubes, which are the best results reported thus far under the grid model. In particular, the number of tracks of the collinear layout of hypercubes is optimal within a factor of $1.\bar{3}$, and the areas of proposed 2-D layouts for hypercubes and CCC are optimal within a factor of $1.\bar{7}$ under the grid model. We also derived efficient multilayer layouts for hypercubes, which are the best results reported thus far for the given numbers of layers. The techniques used in this paper can be used to obtain efficient layouts for a wide variety of other interconnection networks [25, 28].

References

[1] Adams, G.B. and H.G. Siegel, "The extra stage cube: a fault-tolerant interconnection network for supersystems," *IEEE Trans. Comput.,* vol. 31, no. 5, May. 1982, pp. 443-454.

[2] Avior, A., T. Calamoneri, S. Even, A. Litman, and A. Rosenberg, "A tight layout of the butterfly network," *Proc. ACM Symp. Parallel Algorithms and Architectures,* Jun. 1996, pp. 170-175.

[3] Bertsekas, D.P. and J. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods,* Athena Scientific, 1997.

[4] Bhuyan L.N. and D.P. Agrawal, "Generalized hypercube and hyperbus structures for a computer network," *IEEE Trans. Comput.,* vol. 33, no. 4, Apr. 1984, pp. 323-333.

[5] Brebner, G., "Relating routing graphs and two-dimensional grids," *VLSI: Algorithms and Architectures,* 1985, pp. 221-231.

[6] Chen, C. and D.P. Agrawal, "dBCube: a new class of hierarchical multiprocessor interconnection networks with area efficient layout," *IEEE Trans. Parallel Distrib. Sys.,* vol. 4, no. 12, Dec. 1993, pp. 1332-1344.

[7] Chen G. and F.C.M. Lau, "A compact layout of cube-connected cycles," *Proc. Int'l Conf. High Performance Computing,* Dec. 1997, 422-427.

[8] Fernández, A. and K. Efe, "Efficient VLSI layouts for homogeneous product networks," *IEEE Trans. Computer,* vol. 46, no. 10, Oct. 1997, pp. 1070-1082.

[9] Johnsson, S.L. and C.-T. Ho, "Optimum broadcasting and personalized communication in hypercubes," *IEEE Trans. Computers,* vol. 38, no. 9, Sep. 1989, pp. 1249-1268.

[10] Lai, T.-H. and A.P. Sprague, "Placement of the processors of a hypercube," *IEEE Trans. Computers,* vol. 40, no. 6, Jun. 1991, pp. 714-722.

[11] Lakshmivarahan, S. and S.K. Dhall, "A new hierarchy of hypercube interconnection schemes for parallel computers," *J. Supercomputing,* vol. 2, 1988, pp. 81-108.

[12] Leighton, F.T., *Complexity Issues in VLSI : Optimal Layouts for the Shuffle-exchange Graph and Other Networks* Cambridge, Mass., MIT Press, 1983.

[13] Leighton, F.T., *Introduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercubes,* Morgan-Kaufman, San Mateo, CA, 1992.

[14] Parhami B., *Introduction to Parallel Processing: Algorithms and Architectures*, Plenum Publishing Corp., 1999.

[15] Parhami B. and D.-M. Kwai, "Periodically regular chordal rings," *IEEE Trans. Parallel Distrib. Sys.,* to appear.

[16] Preparata, F.P. and J.E. Vuillemin, "The cube-connected cycles: a versatile network for parallel computation," *Commun. ACM,* vol. 24, No. 5, pp. 300-309, May 1981.

[17] Sýkora and I. Vrt'o, "On VLSI layouts of the star graph and related networks," *Integration, VLSI J.* 1994, pp. 83-93.

[18] Thompson, C.D., "Area-time complexity for VLSI," *Proc. ACM Symp. Theory of Computing*, 1979, pp. 81-88.

[19] Thompson, C.D., "A complexity theory for VLSI," Ph.D. dissertation, Dept. of Computer Science, Carnegie-Mellon Univ., Pittsburgh, PA, 1980.

[20] Varvarigos, E.A. and D.P. Bertsekas, "Communication algorithms for isotropic tasks in hypercubes and wraparound meshes," *Parallel Computing*, vol. 18, no. 11, Nov. 1992, pp. 1233-1257.

[21] Varvarigos, E.A., "Static and dynamic communication in parallel computing," Ph.D. dissertation, Dept. Electrical Engineering and Computer Science, Massachusetts Institute of Technology, 1992.

[22] Yeh, C.-H. and B. Parhami, "A class of parallel architectures for fast Fourier transform," *Proc. Midwest Symp. Circuits and Systems,* Aug. 1996, pp. 856-859.

[23] Yeh, C.-H. and B. Parhami, "Recursive hierarchical swapped networks: versatile interconnection architectures for highly parallel systems," *Proc. IEEE Symp. Parallel and Distributed Processing*, Oct. 1996, pp. 453-460.

[24] Yeh, C.-H. and B. Parhami, "Cyclic networks – a family of versatile fixed-degree interconnection architectures," *Proc. Int'l Parallel Processing Symp.*, Apr. 1997, pp. 739-743.

[25] Yeh, C.-H., "Efficient low-degree interconnection networks for parallel processing: topologies, algorithms, VLSI layouts, and fault tolerance," Ph.D. dissertation, Dept. Electrical & Computer Engineering, Univ. of California, Santa Barbara, Mar. 1998.

[26] Yeh, C.-H. and E.A. Varvarigos, "Macro-star networks: efficient low-degree alternatives to star graphs," *IEEE Trans. Parallel Distrib. Sys.,* Vol. 9, no. 10, Oct. 1998, pp. 987-1003.

[27] Yeh, C.-H. and B. Parhami, "VLSI layouts of complete graphs and star graphs," *Information Processing Letters,* Vol. 68, Oct. 1998, pp. 39-45.

[28] Yeh, C.-H., B. Parhami, and E.A. Varvarigos, "The recursive grid layout scheme for VLSI layout of hierarchical networks," *Proc. Merged Int'l Parallel Processing Symp. & Symp. Parallel and Distributed Processing*, Apr. 1999, to appear.

[29] Ziavras, S.G., "RH: a versatile family of reduced hypercube interconnection networks," *IEEE Trans. Parallel Distrib. Sys.,* vol. 5, no. 11, Nov. 1994, pp. 1210-1220.