The Recursive Grid Layout Scheme for VLSI Layout of Hierarchical Networks

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

Abstract

We propose the recursive grid layout scheme for deriving efficient layouts of a variety of hierarchical networks and computing upper bounds on the VLSI area of general hierarchical networks. In particular, we construct optimal VLSI layouts for butterfly networks, generalized hypercubes, and star graphs that have areas within a factor of $1+o(1)$ *from their lower bounds. We also derive efficient layouts for a number of other important networks, such as cube-connected cycles (CCC) and hypernets, which are the best results reported for these networks thus far.*

1. Introduction

The layout of interconnection networks has important cost and performance implications. It is self-evident that a more compact layout leads to lower cost, since reducing the per-processor layout area directly translates to fewer chips, boards, and assemblies. Smaller physical size also leads to shorter wires, thereby improving the signal propagation delay and power requirements. Thus, the impact of efficient VLSI layout on cost-performance is amplified by the lower cost and higher performance. Efficient layouts for several interconnection networks can be found in [5, 8, 15, 18, 23].

We propose the *recursive grid layout scheme* for efficient VLSI layout of hierarchical networks. The proposed scheme is generally applicable to a very wide variety of interconnection networks. Based on this scheme, we derive upper bounds on the VLSI areas of general hierarchical networks. We also derive layouts of butterfly networks [16], generalized hypercubes [4, 12], hierarchical cubic networks (HCNs) [9], hierarchical folded-hypercube networks (HFNs) [7], transposition networks [14], hierarchical swapped networks (HSNs) [22, 23], and indirect swapped networks (ISNs) [21], which have areas optimal within a factor of $1+o(1)$. Moreover, we present efficient layouts for cube-connected cycles (CCC) [17], folded hypercubes [1], hypernets [11], pancake graphs [2], bubble-sort graphs [2], reduced hypercubes [27], recursively connected complete (RCC) networks [10], hierarchical hypercube networks (HHNs) [26], star-connected cycles (SCC) [13], recursive hierarchical swapped networks (RHSNs) [22], and enhanced cubes [20], which are the best results reported for these networks thus far.

2. The recursive grid layout scheme

In this section, we present a generally applicable scheme for laying out hierarchical networks. We use the *extended grid model*, which is an extended version [8, 18, 23] of Thompson's grid model [19], for the VLSI layout of networks with arbitrary node degree. 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 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 that area.

2.1. Describing the layout scheme

Suppose we are given a degree-d network that is viewed as having l levels of hierarchy. Each link of a node is assigned a distinct label i, which is called the *dimension* of the link. We assume that the network can be partitioned into M_l disjoint subgraphs, each of which has (at most) N_l nodes and is called a *level-l cluster* so that every dimension-i link, $i \leq d - p_l$, is confined within a level-l cluster. Moreover, we assume that for $h = l, l-1, l-2, \ldots, 3$, a level-$h$ cluster can be partitioned into M_{h-1} level-$(h-1)$ clusters, each having (at most) N_{h-1} nodes, so that every dimension-i link, $i \leq d - \sum_{j=h}^{l} p_j$, is confined within a level-$(h-1)$ cluster. Level-2 clusters are the basic building modules of the network and are called *nuclei* in this paper. For example, a d-dimensional hypercube, or d-cube, is a d-level hierarchical network with $p_i = 1$, $M_i = 2$, and $N_i = 2^{i-1}$ for $i = 2, 3, 4, \ldots, d$, whose nucleus consists of two connected nodes (i.e. a 1-cube). A $(d+1)$-dimensional star graph, or $(d+1)$-star, is another d-level hierarchical network with $p_i = 1$, $M_i = i+1$, and $N_i = i!$ for $i = 2, 3, 4, \ldots, d$, whose nucleus consists of two connected nodes (i.e. a 2-star). A d-dimensional hypercube can also be viewed as a $\lceil d/2 \rceil$-level hierarchical network with $p_i = 2$ and $M_i = 4$ for $i = 2, 3, 4, \ldots, \lceil d/2 \rceil$, whose nucleus is a 1-cube when d is odd and is a 2-cube otherwise. In general, an l-level hierarchical network can be characterized (not in a unique way) by the set of integers $\{p_i, M_i, N_i\}, i = 2, 3, \ldots, l$, and its nucleus.

To lay out an l-level hierarchical network, we first place nodes belonging to the same level-l cluster within a block, which we call a *level-l block*. We arrange the blocks as a 2-D grid, with neighboring rows (or columns) separated by sufficient horizontal tracks (or vertical tracks, respectively) (see Fig. 1). We then lay out dimension-i links, $i = d, d-1, \ldots, d - p_l + 1$, which are collectively called *level-l inter-cluster links*, outside the blocks. Note that we will eventually connect each of the level-l inter-cluster links incident to a level-l block to a certain node within the block. We can then continue to lay out each level-l cluster, including the M_{l-1} level-$(l-1)$ blocks within it and the links connecting these level-$(h-1)$ blocks, within a level-l block. This pro-

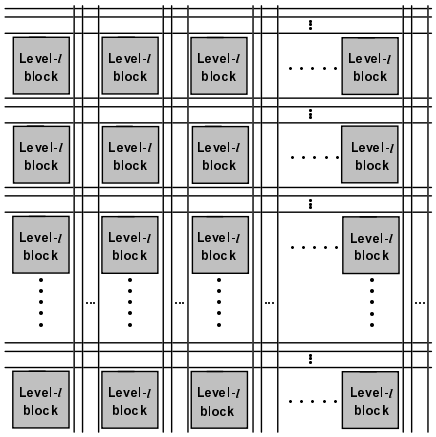

Figure 1. Top-view of a layout based on the recursive grid layout scheme. Level-l blocks are arranged as a 2-D grid.

cess is repeated recursively until each block contains a nucleus, or until the number of nodes within a block to be laid out is small. Then we use any viable method to lay out all the nuclei or small clusters.

Note that we can use a block of side $p_l N_l$ to accommodate the wires connecting level-l inter-cluster links to nodes within the block. However, we may need extra space to accommodate intra-cluster links connecting nodes within a level-l cluster. This can be easily done by expanding the blocks to the required size. All the blocks remain aligned as a 2-D grid and all the tracks outside level-l blocks are moved accordingly. Except for the increased width and height for these blocks, the numbers of vertical and horizontal tracks required outside these blocks are not changed. Similarly, we can use blocks of side $\sum_{j=h}^{l} p_j N_h$ to accommodate the wires from outside a level-h block, $h = l-1, l-2, \ldots, 3$, before laying out the links within the block. If such a square is not large enough to accommodate the wires from outside the block, the level-$(h-1)$ blocks within it, and the links connecting these level-$(h-1)$ blocks, we simply expand the level-h blocks and maybe the blocks of level $h+1$, $h+2$, and so on, to which they belong, if necessary. Sometimes we may lay out a level-h cluster and connect its nodes to inter-cluster links within an area smaller than that of the original block. In such a case, we simply shrink these blocks and keep them aligned as a 2-D grid (see Fig. 1).

This top-down layout method is quite simple, and can lead to the best layout areas for a variety of networks, such as butterfly, CCC, star graphs, generalized hypercubes, hypernets, HCNs, and transposition networks. As shown in [23], many of the resulting layouts are optimal within a factor of $1+o(1)$.

2.2. Deriving area upper bounds for general hierarchical networks

In this subsection, we derive upper bounds on the VLSI areas of general hierarchical networks based on the recursive grid layout scheme.

Lemma 2.1 *An N-node network can be laid out in a square of side at most $p_l N/2 + S_l \lceil \sqrt{M_l} \rceil$, where p_l is the maximum number of top-level inter-cluster links per node, S_l is the side required for a top-level block, and M_l is the number of top-level clusters in the network.*

Proof: To lay out a link, we need at most one vertical and one horizontal track, in addition to the two ending segments connecting the link to (at most) two level-l blocks. Since there are at most $N/2$ links of dimension i for each $i \in [d - p_l + 1, d]$, where d is the degree of the network, we need at most $p_l N/2$ vertical and horizontal tracks to accommodate all the level-l inter-cluster links. If we arrange the level-l blocks as a square 2-D grid, the increased width or height required to accommodate these blocks is $S_l \lceil \sqrt{M_l} \rceil$ and the result follows. □

From Lemma 2.1, it can be seen that the area required for laying out the top-level inter-cluster links is approximately proportional to the square of the number of nodes in the network, when the proposed recursive layout scheme is used. If each cluster at the same level has the same size, which is the usual case for hierarchical networks, then we can obtain the following theorems.

Theorem 2.2 *An N-node hierarchical network can be laid out in $O(N^2)$ area if the number of level-i inter-cluster links $p_i = O(1)$ for all i and the area required for all the nuclei is $O(N^2)$.*

Proof: If all clusters at the same level have the same size, then the size of a cluster is no more than 1/4 of that of a cluster that is two levels higher. Therefore, we can view the network as having l levels of hierarchy with $p_i = O(1)$, $M_i \geq 4$, and $N_i/N_{i+1} \leq 1/4$, by merging two levels when necessary. The overall increase in width or height required for the expansion of the blocks in order to accommodate the level-i inter-cluster links, $i = 2, 3, \ldots, l-1$, is

$$O\left(\sum_{i=2}^{l-1} N_{i+1} \prod_{j=i+1}^{l} \lceil \sqrt{M_j} \rceil\right) = O(N).$$

Thus, the overall width or height is $O(N)$ from Lemma 2.1 and the area is $O(N^2)$. □

When the top-level clusters of a hierarchical network are not large, the upper bound on its area can be improved.

Theorem 2.3 *An N-node l-level hierarchical network can be laid out in $N^2/4 + o(N^2)$ area, if $p_l = 1$, $p_i = O(1)$ for all $i \leq l-1$, M_l is not a constant, and the area required for all the nuclei is $o(N^2)$.*

Proof: Similar to the proof of Theorem 2.2, the increase in width or height required for inter-cluster links at all levels i, $i = 2, 3, \ldots, l-1$, is $O(N_l) = o(N)$. Thus, the overall width or height is $N/2 + o(N)$ from Lemma 2.1, and the area is $N^2/4 + o(N^2)$. □

As can be seen from the previous proofs, the recursive layout scheme allows us to derive tight layouts for many hierarchical networks easily by focusing on the layout of the top-level inter-cluster links. We present some examples in the following section.

3. Efficient layouts for several networks

3.1. Layouts for certain Cayley graphs

In this subsection we present efficient layouts for several Cayley graphs [2], including star, pancake, and bubble-sort graphs [2], star-connected cycles (SCC) [13], and transposition networks [14].

Theorem 3.1 *An N-node star graph, pancake graph, or bubble-sort graph can be laid out in $N^2/16 + o(N^2)$ area.*

Proof: An n-star contains n disjoint $(n-1)$-stars as subgraphs, each pair of which are connected by $(n-2)!$ links. If we view each $(n-1)$-star subgraph as a supernode, the n-star becomes a complete graph with n supernodes and multiple edges. Therefore, all the dimension-n links can be laid out based on the layout of an n-node complete graph K_n with $(n-2)!$ edges between each pair of nodes. In [23, 24] we have shown that the 2-D layout for a K_n with 2 edges between each pair of nodes requires $n^4/4 + o(n^4)$ area. Similarly, a K_n with $(n-2)!$ edges between each pair of nodes can be laid out in

$$(n^2(n-2)!)^2/16 + o(n^2(n-2)!)^2 = N^2/16 + o(N^2)$$

area, where $N = n!$. This can be easily done be expanding each side-$(2n-2)$ node in a directed K_n into a side-$(n-1)!$ node and replicating each link into $(n-2)!/2$ links. When we continue to lay out level-$(n-1)$ clusters, which are $(n-1)$-stars, the level-l blocks may need to be expanded. The maximum height or width increase due to such expansion is no more than $O(N/\sqrt{n})$. As a result, the layout area for an n-star is $N^2/16 + o(N^2)$.

An n-dimensional pancake graph (or bubble-sort graph) also has n pancake graphs (or bubble-sort graphs) of dimension $n-1$ as subgraphs, each pair of which are connected by $(n-2)!$ links. Therefore, they can be laid out using the preceding method, and the required area is asymptotically identical to that of an n-star. □

The layout area upper bounds for the star graph and pancake graph, given in Theorem 3.1, are 72 times smaller than the ones in [18]. By using the following lemma and theorem [23], we can show that the preceding area for the star graph is optimal within a factor of $1 + o(1)$.

Lemma 3.2 *d TE tasks can be executed in $(N-1)D_{ave}$ communication time in a vertex- and edge-symmetric network under the all-port communication model, where d is the degree of the network, D_{ave} is the average distance of the network, and N is the size of the network.*

Lemma 3.2 leads to the following universal lower bound on the VLSI area of any vertex- and edge-symmetric network.

Theorem 3.3
The VLSI area of a vertex- and edge-symmetric network is at least

$$\frac{d^2 \lfloor N/2 \rfloor^2 \times \lceil N/2 \rceil^2}{D_{ave}^2 (N-1)^2} \approx \frac{d^2 N^2}{16 D_{ave}^2} \ ,$$

where d is the degree of the network, D_{ave} is the average distance of the network, and N is the size of the network.

An SCC can be viewed as a 2-level hierarchical network with $p_2 = 1$, $M_2 = n!$, and $N_2 = (n-1)$, whose nucleus is an $(n-1)$-node ring. The layout of the SCC can be obtained by expanding each node in the layout of an n-star into a block containing an $(n-1)$-node ring, leading to the following theorem.

Theorem 3.4 *An N-node SCC can be laid out in area*

$$\frac{N^2(\log_2 \log_2 N)^2}{16 \log_2^2 N} + o\left(\frac{N^2(\log\log N)^2}{\log^2 N}\right).$$

An n-dimensional transposition network can be viewed as an $(n-1)$-level hierarchical network with $p_i = i$, $M_i = i+1$, and $N_i = i!$ for $i = 2, 3, 4, \ldots, n$, whose nucleus consists of two connected nodes. An n-dimensional transposition network has n transposition networks of dimension $n-1$ as subgraphs, each pair of which are connected by $(n-1)!$ links and can be laid out using a method similar to that for an n-star by replicating a wire connecting i-star supernodes in the layout i times, $i = 3, 4, \ldots, n-1$, leading to the following theorem.

Theorem 3.5 *An N-node transposition network can be laid out in area*

$$\frac{N^2 \log_2^2 N}{16(\log_2 \log_2 N)^2} + o\left(\frac{N^2 \log^2 N}{(\log\log N)^2}\right).$$

This layout for transposition network is optimal within a factor of $1 + o(1)$ from the lower bound given in [23].

3.2. Layouts for generalized hypercubes and related networks

In this subsection we present efficient layouts for several networks that are recursively constructed by connecting the clusters as generalized hypercubes [4, 12].

If we view each level-l cluster of an l-level hierarchical swapped network, HSN(l, G), as a supernode, the HSN(l, G) becomes a complete graph with M supernodes and N/M^2 edges connecting each pair of supernodes, where M is the size of its nucleus G. Similar to the proofs for Theorems 3.1 and 3.5, we can show that if the top-level clusters are connected as a complete graph with single or multiple edges and there are at most p_l inter-cluster link(s) per node (where $p_l = 1$ for HSNs), the top-level inter-cluster links can be laid out in $p_l^2 N^2/16 + o(p_l^2 N^2)$ area. This leads to the following theorems.

Theorem 3.6 *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 a nucleus G, is not a constant.

The layouts for HSNs are optimal within a factor of $1+o(1)$ from the lower bound given in [23, 25] if the nucleus G is dense enough (i.e., the nucleus G can execute l TE tasks in M steps under the all-port communication model [23]).

A hierarchical hypercube network (HHN) [26] is an HSN whose nucleus is a hypercube. A hierarchical cubic network (HCN) [9] without diameter links (or a hierarchical folded-hypercube network (HFN) [7]) is a 2-level HSN that uses a $\sqrt{N}$-node hypercube (or a folded hypercube, respectively) as the nucleus. Their layout areas are given in the following corollary.

Corollary 3.7 *An N-node HCN, HFN, or HHN can be laid out using $N^2/16+o(N^2)$ area.*

The layouts for HCNs and HFNs are optimal within a factor of $1+o(1)$.

An r-deep recursive hierarchical swapped network (RHSN) [22] is defined as RHSN$(l_r,l_{r-1},\ldots,l_1,G)$ = HSN$(l_r,$ RHSN$(l_{r-1},l_{r-2},\ldots,l_1,G))$. Clearly, RHSN can be laid out by recursively laying out HSNs.

Theorem 3.8 *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, $l_r=o(logN)$.*

An l-level recursively connected complete (RCC) graph [10] is equivalent to an RHSN$(\underbrace{2,2,\ldots,2}_{l-1},G)$, leading to:

Corollary 3.9 *An N-node l-level RCC can be laid out using $N^2/16+o(N^2)$ area if*

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

(b) *$l\geq 3$,*

where M is the size of the nucleus.

By viewing each nucleus of an HSN as a supernode, we obtain a generalized hypercube with radix-M [4, 12]. Therefore, the layout of Theorem 3.6 leads to the following theorem for the layout of high-radix hypercubes.

Theorem 3.10 *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.*

Since a radix-M generalized hypercube is vertex- and edge-symmetric, we can show that the layout for generalized hypercubes is optimal within a factor of $1+o(1)$ from Theorem 3.3. The above layout can be easily extended to mixed-radix generalized hypercubes [4].

Hypernets are constructed by recursively connecting identical networks using complete graphs [11]. A hypernet is an l-level hierarchical network with $M_l=\sqrt{N/2^{l-1}}$ and $N_l=\sqrt{N2^{l-1}}$, whose nucleus is a cubelet, treelet, or buslet.

Theorem 3.11 *An l-level hypernet can be laid out using $N^2/2^{2l+2}+o(N^2/2^{2l})$ area, where N is the number of nodes in the network.*

Proof: The top-level inter-cluster links of an l-level hypernet are connected as a $\sqrt{N/2^{l-1}}$-node complete graph, which requires $N^2/2^{2l+2}+o(N^2/2^{2l})$ area. The additional area required to accommodate all the level-i inter-cluster links, $i=2,3,4,\ldots,l-1$, diameter links, and all the nuclei is of a smaller order of magnitude. □

3.3. Layouts for some hypercubic networks

Hypercubic networks are among the most important networks for parallel processing and have been intensely studied in the literature [1, 16, 17, 20, 25].

An enhanced-cube is a hypercube that has an additional outgoing link per node leading to a random node [20]. In [23, 25] we have shown that an N-node hypercube can be laid out in $\frac{4}{9}N^2+o(N^2)$ area, leading to the following theorems.

Theorem 3.12 *An N-node folded hypercube can be laid out in $\frac{49}{36}N^2+o(N^2)$ area and an N-node enhanced-cube can be laid out in $\frac{25}{9}N^2+o(N^2)$ area.*

Proof: 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 a vertical track and a horizontal track, in addition to the two ending segments connecting the link to two nodes. Since there are $N/2$ diameter 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 $\frac{25}{9}N^2+o(N^2)$. □

Note that by arranging these additional links appropriately so that a track may be shared by two or more links, the areas of the above layouts may be considerably improved.

We can view an n-dimensional CCC as a 2-level hierarchical network with $p_2=1$ and $M_2=2^n$, whose nucleus is an n-node ring. We can lay out all the $N/2$ inter-cluster links of an n-dimensional CCC using the layout for an n-cube, which requires $2^{n+2}/9+o(2^n)$ area [23, 25]. A reduced hypercube, RH$(\log_2 n,\log_2 n)$ [27], can be obtained by replacing each n-node cycle in a CCC with a $\log_2 n$-dimensional hypercube and can be laid out in asymptotically the same area.

Theorem 3.13 *An N-node CCC or RH$(\log_2 n,\log_2 n)$ can be laid out in area*

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

The area of our layout is smaller than the area of the layout given in [6] by a factor of 1.125 and is within a factor of $1.7+o(1)$ from the lower bound given in [6].

A indirect swapped network (ISN) (also called unfolded swapped network (USN) [21]) is a multistage network obtained by unfolding the structure of a swapped network [22, 23]. If we place every M_l rows of the ISN into the same top-level block, then each pair of the blocks are connected by 2 links, where M_l is the number of top-level clusters in the corresponding swapped network unfolded to generate the ISN.

Theorem 3.14 *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 M_l is not a constant.

The previous layout area improves the result given in [21] by a factor of 4 and is optimal within a factor of $1+o(1)$ from the lower bound given in [23, 25].

Theorem 3.15 *An N-node butterfly network can be laid out in an area equal to*

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

Proof: By unfolding 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 an $(\log_2 N)$-dimensional butterfly. Therefore, the area of the butterfly is approximately 4 times that of an ISN; that is

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

Recently, Avior et al proposed an area-optimal VLSI layout for butterfly networks [3] under Thompson's grid model [19], assuming that the width of a network node is equal to 1 (i.e., the same as the width of a wire). The area of the layout proposed in [3], however, becomes $\frac{W^2N^2}{\log_2^2 N} + o\left(\frac{W^2N^2}{\log^2 N}\right)$ when the width of network nodes is W. As a comparison, our layout is the only butterfly layout reported in the literature that has area optimal within a factor of $1+o(1)$ under the extended grid model ($W=4$).

4. Conclusion

We proposed the recursive grid layout scheme for efficient VLSI layout of hierarchical networks. The proposed scheme is generally applicable to a very wide variety of networks as well as general hierarchical networks. Many of our layouts are optimal within a factor of $1+o(1)$; others are the best results reported in the literature thus far.

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] Akers, S.B. and B. Krishnamurthy, "A group-theoretic model for symmetric interconnection networks," *IEEE Trans. Comput.*, vol. 38, Apr. 1989, pp. 555-565.

[3] Avior, A., T. Calamoneri, S. Even, A. Litman, and A. Rosenberg, "A tight layout of the butterfly network," *Theory Comput. Sys.*, vol. 31, no. 4, 1998, pp. 475-488.

[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] 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.

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

[7] Duh, D., G. Chen, and J. Fang, "Algorithms and properties of a new two-level network with folded hypercubes as basic modules," *IEEE Trans. Parallel Distrib. Sys.*, vol. 6, no. 7, Jul. 1995, pp. 714-723.

[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] Ghose, K. and R. Desai, "Hierarchical cubic networks," *IEEE Trans. Parallel Distrib. Sys.*, vol. 6, no. 4, Apr. 1995, pp. 427-435.

[10] Hamdi, M., "A class of recursive interconnection networks: architectural characteristics and hardware cost," *IEEE Trans. Circuits and Sys.–I: Fundamental Theory and Applications*, vol. 41, no. 12, Dec. 1994, pp. 805-816.

[11] Hwang, K. and J. Ghosh, "Hypernet: a communication efficient architecture for constructing massively parallel computers," *IEEE Trans. Comput.*, vol. 36, no. 12, Dec. 1987, pp. 1450-1466.

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

[13] Latifi, S., M. de Azevedo, and N. Bagherzadeh, "The star connected cycles: a fixed-degree network for parallel processing," *Proc. Int'l Conf. Parallel Processing*, vol. I, 1993, pp. 91-95.

[14] Latifi, S. and P.K. Srimani, "Transposition networks as a class of fault-tolerant robust networks," *IEEE Trans. Parallel Distrib. Sys.*, vol. 45, no. 2, Feb. 1996, pp. 230-238.

[15] Leiserson, C.E., "Fat-trees: universal networks for hardware-efficient supercomputing," *IEEE Trans. Computers*, vol. C-34, no. 10, Oct. 1985, pp. 892-901.

[16] Parhami, B., *Introduction to Parallel Processing: Algorithms and Architectures*, Plenum Press, 1999.

[17] 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.

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

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

[21] 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.

[22] 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.

[23] 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.

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

[25] Yeh, C.-H., E.A. Varvarigos, and B. Parhami, "Efficient VLSI layouts of hypercubic networks," *Proc. Symp. Frontiers of Massively Parallel Computation*, Feb. 1999, to appear.

[26] Yun, S.-K. and K.H. Park, "Hierarchical hypercube networks (HHN) for massively parallel computers," *J. Parallel Distrib. Comput.*, vol. 37, no. 2, Sep. 1996, pp. 194-199.

[27] 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.