

VLSI layouts of complete graphs and star graphs

Chi-Hsiang Yeh, Behrooz Parhami*

Department of Electrical and Computer Engineering, University of California, Santa Barbara, CA 93106-9560, USA

Received 18 December 1997

Communicated by F. Dehne

Abstract

In this paper, we present efficient layouts for complete graphs and star graphs. We show that an N -node complete graph can be optimally laid out using $\lfloor N^2/4 \rfloor$ tracks for a colinear layout, and can be laid out in $N^4/16 + o(N^4)$ area for a 2D layout. We also show that an N -node star graph can be laid out in $N^2/16 + o(N^2)$ area, which is smaller than any possible layout of a similar-size hypercube. This solves an open question posed by Akers and Krishnamurthy in 1986. Both the layouts of complete graphs and star graphs are optimal within a factor $1 + o(1)$. © 1998 Published by Elsevier Science B.V. All rights reserved.

Keywords: Complete graphs; Star graphs; VLSI layout; Parallel processing

1. Introduction

Deriving an efficient VLSI layout for an interconnection network 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. Parallel processing interconnection networks are often characterized by their graph theoretic and topological parameters such as node degree, diameter, average inter-node distance, and bisection width. While lower node degree directly translates to lower cost, the cost implications of the other parameters, as well as the effect on performance, depend on other factors that are not easily quantified. For example, a small diameter can potentially lead to higher performance by reducing the data transmission latency measured by the number of hops. However, if

the improved diameter necessitates the use of longer wires, the associated increase in signal propagation delays and message contentions might nullify the gain and even lead to lower overall performance. Efficient layouts for several interconnection networks can be found in [7,11,17,19,22].

It has been previously shown that a particular class of Cayley graphs known as star graphs [2] possess desirable properties such as symmetry and strongly hierarchical structure, as well as smaller diameters, average distance, and node degrees compared to similar-size hypercubes. The development of many efficient algorithms for the star graphs [3,4,20,21,24] and investigation of their various properties [8,14] has confirmed that the above topological superiority leads to a smaller number of communication steps. Our contribution in this paper is to show that star graphs also have more efficient layouts than the hypercube. Hence the lower graph-theoretic density and algorithmic step

* Corresponding author. Email: parhami@ece.ucsb.edu.

count do indeed translate to higher performance at the hardware level.

In [1,2], several open questions about the layout of star graphs were posed. In particular, the following question is stated in [1, p. 223]: “Can the star graph be laid out at least as efficient as the hypercube?” This question is partially answered by Sýkora and Vrt’o [22], who provide a layout for an N -node star graph that has an area of $4.5N^2$, which is of the same order of magnitude as that of a similar-size hypercube. However, its leading constant is considerably larger than that of a hypercube [5]. In this paper, we improve the answer to this question by providing an optimal layout for the star graph that has an area of $N^2/16 + o(N^2)$, which is 4 times smaller than the lower bound $N^2/4$ on the area of a similar-size hypercube. The upper bound given in this paper is 72 times smaller than the one in [22] and is within a factor $1 + o(1)$ from the lower bound.

We show that the number of tracks required for the colinear layout of an N -node complete graph and, as a result, any N -node simple graph is no more than $\lfloor N^2/4 \rfloor$. This result exactly matches the lower bound for the colinear layout of complete graphs and improves the upper bound given in [7] by 25%. We also show that an N -node complete graph can be laid out in $N^4/16 + o(N^4)$ area, which is optimal within a factor $1 + o(1)$ from its lower bound.

2. Efficient layouts for complete graphs

In this section, we present several optimal layouts for complete graphs.

We use the grid model for VLSI layout of networks [23], extended for nonconstant node degree [7, 22]. 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 2D 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. More details concerning the VLSI model can be found in [5,7,17,22].

2.1. Optimal colinear layouts of complete graphs

In [7], a layout that requires $4(4^{\log_2 N - 1} - 1)/3 \approx N^2/3$ tracks is presented for mapping an N -node complete graph, K_N , onto a linear array. In what follows, we show that such a mapping, called a *colinear layout*, can be considerably improved to one that uses $\lfloor N^2/4 \rfloor$ tracks, which exactly matches the bisection-based lower bound.

To obtain the colinear layout, we first place the N nodes, labeled 1 through N , along a row. Let a link be *type- i* if it connects two nodes whose addresses differ by i . Then the $N(N-1)/2$ links in K_N can be classified into types $1, 2, 3, \dots, N-1$, and there are $N-i$ type- i links. To lay out K_N , we place all the type-1 links in one track, all the type-2 links in two tracks, where links connecting nodes with odd addresses are put in one track and links connecting nodes with even addresses in the other, and then all the type- i links in $\min(i, N-i)$ tracks for $i = 3, 4, 5, \dots, N-1$. More precisely, type- i links are placed in i tracks if $i \leq N/2$, where links are put in the same track if the remainders of their node-addresses modulo i are the same, and each of the $N-i$ type- i links is placed in a different track if $i > N/2$. Clearly, such an arrangement will not result in overlapped links within a track. The resulting layout for a K_9 is illustrated in Fig. 1.

The total number of tracks in the layout described above is equal to

$$\begin{aligned} \sum_{i=1}^{N-1} \min(i, N-1) &= \sum_{i=1}^{\lfloor N/2 \rfloor} i + \sum_{i=\lfloor N/2 \rfloor + 1}^{N-1} (N-i) \\ &= \sum_{i=1}^{\lfloor N/2 \rfloor} i + \sum_{i=1}^{\lfloor N/2 \rfloor - 1} i = \lfloor N^2/4 \rfloor. \end{aligned}$$

Since the bisection width of K_N is equal to $N^2/4$ when N is even and $(N^2-1)/4$ when N is odd, this layout is strictly optimal in terms of the number of tracks for colinear layouts of complete graphs.

A *simple graph* is a graph that has no self-cycles at any node and no multiple links between any pair of nodes. Since any simple graph can be embedded into a complete graph of the same size, we obtain the following theorem.

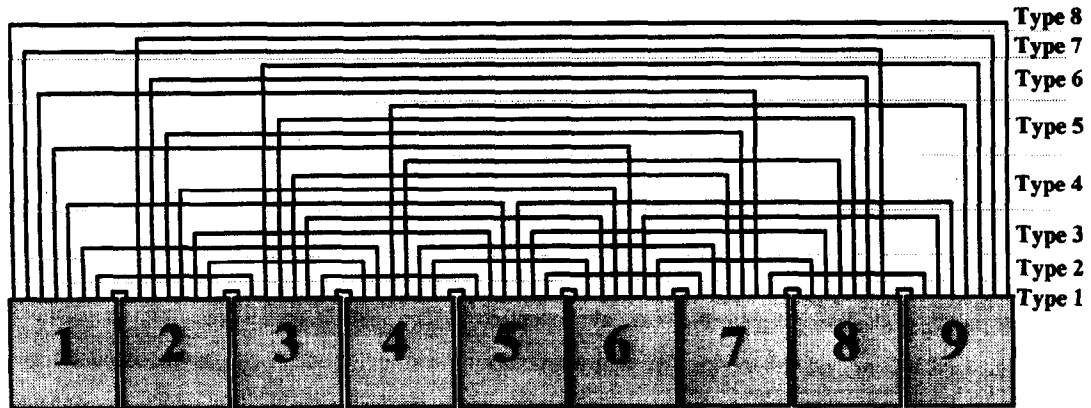


Fig. 1. A colinear layout for the 9-node complete graph K_9 .

Theorem 2.1. *The number of tracks required for the colinear layout of an N -node simple graph is at most $\lfloor N^2/4 \rfloor$.*

This upper bound is 25% smaller than the one given in [7, Theorem 1]. The above number of tracks leads to an area of $N(N-1)\lfloor N^2/4 \rfloor \approx N^4/4$.

2.2. Optimal 2D layouts of complete graphs

Although the method introduced in the previous subsection leads to the smallest possible number of tracks for the colinear layout of a complete graph, layouts with smaller area can be obtained using 2D layouts. Based on the previous colinear layout, we first derive an area-efficient layout for directed complete graphs, where each pair of nodes are connected by two directed edges. Without loss of generality, we assume that $N = m_1 \times m_2$ for some pair of integers $m_1, m_2 = \Theta(\sqrt{N})$.

To obtain an area-optimal layout, we put the N nodes of the complete graph, labeled (i, j) for $i = 1, 2, \dots, m_1$, $j = 1, 2, \dots, m_2$, on an $m_1 \times m_2$ grid. Two neighboring rows are separated by $2m_1\lfloor m_2^2/4 \rfloor$ tracks while two neighboring columns are separated by $2m_2\lfloor m_1^2/4 \rfloor$ tracks. We call a link from the source node (i_1, j_1) to the destination node (i_2, j_2) a *type-* $(i_1, j_1, j_2 - j_1)$ link. If $i_1 = i_2$ or $j_1 = j_2$, we can route the link as in the colinear layout. Otherwise, we first route it from the source node to the vicinity of the upper right corner of the *turning node* (i_1, j_2) along

a horizontal track, and from there to the destination node (i_2, j_2) along a vertical track. Recall that we need $\min(k, m_2 - k)$ tracks for all the $m_2 - k$ type- k links in the colinear layout of an undirected K_{m_2} . Since m_1 links go from the node (i_1, j_1) to node (i_1, j_2) as the turning or destination node, and vice versa, we can expand a track in the colinear layout of a K_{m_2} to $2m_1$ tracks to accommodate the horizontal segments of the $2m_1$ directed links, leading to $2m_1\lfloor m_2^2/4 \rfloor$ tracks above each row of nodes.

We next show that the vertical segments of all the links to the immediate right of a column can be placed in $2m_2\lfloor m_1^2/4 \rfloor$ tracks. We present a possible arrangement as follows. We place all the type- (x, y, z) links within the *bundle* (i_1, k) , if $y = j_1$ and $z = k$ or $-k$ for some positive integers i_1 and k , for all integers $x = 1, 2, 3, \dots, m_1$. In other words, links are put within the same bundle if their source nodes belong to the same column i_1 and the difference between their row numbers of the source and destination is the same (i.e., equal to k or $-k$). There are $m_2(m_1 - 1)$ bundles between a pair of columns. Bundle (i_1, k) can be laid out using $2\min(k, m_1 - k)$ successive tracks, which is similar to the layout for two groups of type- k links in the colinear layout of a K_{m_1} . More precisely, a link is placed in the first half of the bundle to which it belongs if $\lfloor (i_1 - 1)/k \rfloor$ is even and placed in the second half otherwise. Within the half of bundle, a link is placed in the l th track if $i_1 \bmod k = l$. Note that we place the vertical segments of links of type- (i_1, j_1, k) and type- $(i_1, j_1, -k)$ alternately along a vertical track when

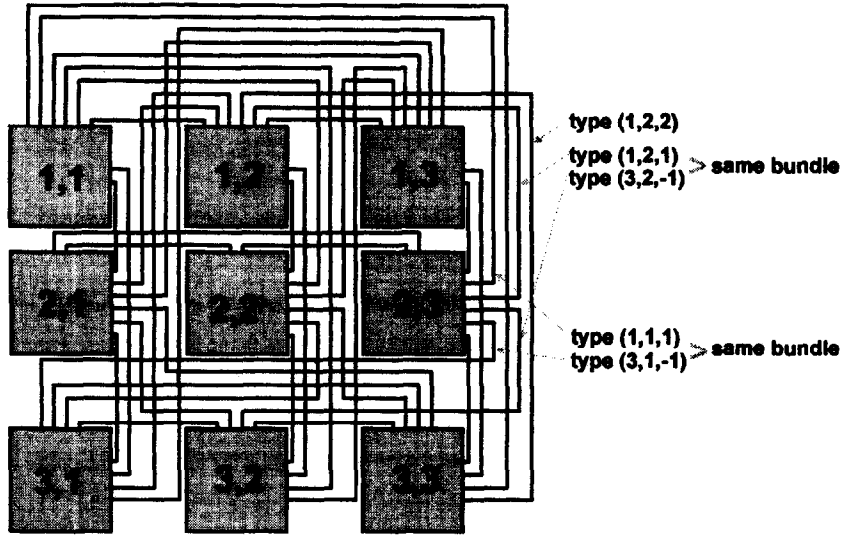


Fig. 2. A 2D layout for an undirected complete graph K_9 .

$k \leq m_1/2$ to avoid overlapping. By arranging links according to the above rules, the vertical segments of all the $2m_2(m_1 - k)$ type- (x, y, k) and type- $(x, y, -k)$ links, $x = 1, 2, 3, \dots, m_1$, $y = 1, 2, 3, \dots, m_2$, can be placed in $2m_2 \min(k, m_1 - k)$ tracks. As a result, the total number of vertical tracks required is equal to $2m_2$ times the required number of tracks $\lfloor m_1^2/4 \rfloor$ for the colinear layout of a K_{m_1} ; that is $2m_2 \lfloor m_1^2/4 \rfloor$.

Since a node occupies a square of side $m_1 m_2 - 1$, the area required for the above 2D layout of the directed K_N is given by

$$\begin{aligned} & m_1(2m_1 \lfloor m_2^2/4 \rfloor + m_1 m_2 - 1) \\ & \times m_2(2m_2 \lfloor m_1^2/4 \rfloor + m_1 m_2 - 1) \\ & = N^4/4 + O(N^{3.5}). \end{aligned}$$

For an undirected K_N , where each pair of nodes are connected by an edge only, the required area can be reduced to $N^4/16 + O(N^{3.5})$ by properly removing half of the tracks in both horizontal and vertical directions. One of the possible methods is to remove the links within the second half of each of the bundles and their horizontal segments as well as half of the links whose sources and destinations have the same row or column numbers. Fig. 2 shows a resultant 2D layout for an undirected K_9 . Note that there are 12 tracks between 2 neighboring rows or columns in

the layout for a directed K_9 ; while after the removal of the second halves of bundles, there are only 6 vertical tracks left between two neighboring columns, and there are 10, 2, and 6 horizontal tracks left above the 1st, 2nd, and 3rd rows, respectively, in the layout for an undirected K_9 .

Theorem 2.2. *An N -node complete graph can be laid out in $N^4/16 + o(N^4)$ area.*

These layout areas are larger than their respective lower bounds by a factor of $1 + o(1)$ and are thus quite close to being strictly optimal. In the following subsections, we will show that the optimal layouts for complete graphs can be used to derive efficient layouts for star graphs.

3. Optimal layouts for star graphs

An n -dimensional star graph, n -star, is a symmetric graph that has $N = n!$ nodes of degree $n - 1$ [2]. Each node in an n -star is assigned a label, which is a distinct permutation of the set of n symbols $\{1, 2, 3, \dots, n\}$. Two nodes are connected with a dimension- i link, $2 \leq i \leq n$, if and only if the label of one node can be obtained from the other by interchanging the first

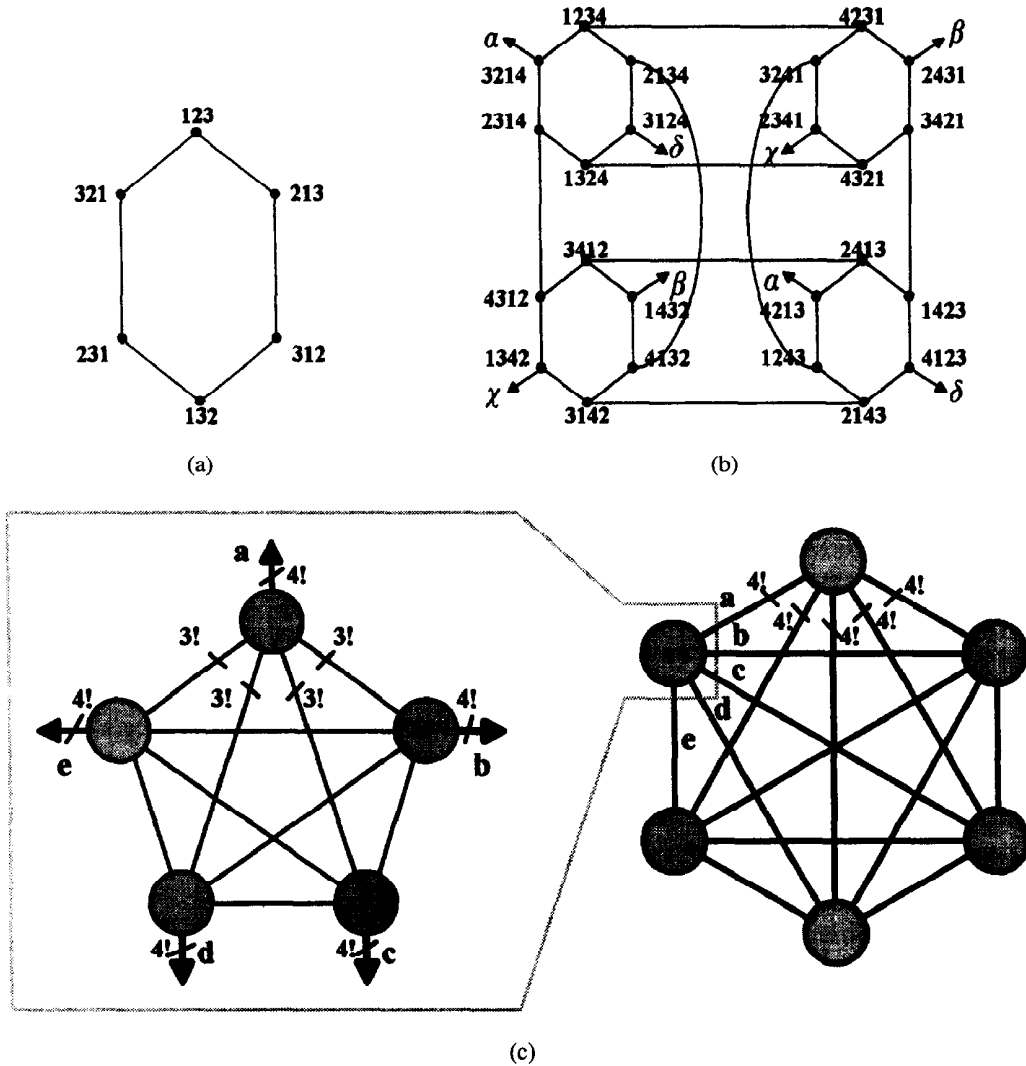


Fig. 3. Structure of a (a) 3-star, (b) 4-star and (c) top view of a 6-star and one of its 5-star subgraphs. Each pair of the 5-star subgraphs within the 6-star is connected by $4!$ dimension-6 links.

symbol and the i th symbol. An n -star contains n disjoint $(n - 1)$ -stars as subgraphs, each pair of which are connected by $(n - 2)!$ links. We can view an $(n - 1)$ -star as a supernode, and then the n -star becomes an n -supernode complete graph with multiple edges. Top-level views for a 6-star and its 5-star supernodes and the complete structures of a 4-star and a 3-star are illustrated in Fig. 3.

Let $n_1 = \lceil \sqrt{n} \rceil$ and $n_2 = \lceil n/n_1 \rceil$. Recall that the grid layout for a $K_{n_1 n_2}$ 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. This can be easily done by expanding each side- $(2n - 2)$ node into a side- $(n - 1)!$ node and replicating each link into $(n - 2)!/2$ links.

To lay out an n -star, we first place nodes belonging to the same $(n - 1)$ -star subgraph within a block of side $(n - 1)!$, which we call an $(n - 1)$ -block, and lay out the dimension- n links using the previous layout

for a $K_{n_1 n_2}$ with multiple edges. We will eventually connect each of the links incident to the $(n - 1)$ -block to a certain node within the block. From the layout of the $K_{n_1 n_2}$, we can see that about $(n!)^2/16$ area is required for such wiring. We can then continue to lay out the $(n - 1)$ -star within each of the $(n - 1)$ -blocks. This process can be done recursively until the number of nodes within a block to be laid out is small. Then we use any viable method to lay out these small substars.

Theorem 3.1. *An n -star can be laid out in $N^2/16 + o(N^2)$ area, where $N = n!$.*

Proof. To lay out all the dimension- n links, $N^2/16 + o(N^2)$ area will be required based on the layout of a $K_{n_1 n_2}$ with $(n - 2)!$ edges between each pair of nodes. To lay out an i -block, $i = n - 1, n - 2, n - 3, \dots, l$, we first connect the wires outside the block to appropriate $(i - 1)$ -blocks within it, and then lay out the dimension- i links of the i -star within the i -block using the layout for a $K_{i_1 i_2}$, where $i_1 = \lceil \sqrt{i} \rceil$ and $i_2 = \lceil i/i_1 \rceil$. Note that $i_1 i_2 - i$ out of the $i_1 i_2$ i -blocks can be removed since there are only $i(i - 1)$ -stars within a block. The number l can be any integer smaller than n and greater than 3. In what follows, we assume $l = 5$. The area for a 5-block is $O(n^2)$ since there are $O(n)$ links incident to it. If a side- $(n - i)!$ square is not large enough to accommodate the wires from outside the i -block and the $i(i - 1)$ -blocks within it, we simply expand the i -blocks and maybe the outside $(i + 1)$ -blocks, $(i + 2)$ -blocks, and so on, if necessary. The maximum height or width increases due to such expansion is smaller than

$$\begin{aligned} & \left[\frac{N}{4\sqrt{n}} + o\left(\frac{N}{\sqrt{n}}\right) \right] \\ & + \left[\frac{N}{\sqrt{n(n-1)}} + \frac{N}{4\sqrt{n(n-1)}} + o\left(\frac{N}{\sqrt{n(n-1)}}\right) \right] \\ & + \dots = O\left(\frac{N}{\sqrt{n}}\right). \end{aligned}$$

As a result, the layout area for an n -star is $N^2/16 + o(N^2)$, which is mainly occupied by dimension- n links. \square

This layout area is smaller than the one given in [22] by a factor of 72. In [28] we have shown that the layout area of an N -node interconnection network is at least

equal to $\lfloor N/2 \rfloor^2 \times \lceil N/2 \rceil^2 / T_{TE}^2$ if f total exchange (TE) tasks can be executed in $f T_{TE}$ communication time and that $(n - 1)$ TE tasks can be executed in $nN + o(nN)$ communication time in an n -star under the all-port communication model. As a result, the lower bound on the VLSI area of an N -node star graph is $N^2/16 - o(N^2)$ and the upper bound we derived is within a factor $1 + o(1)$ from it.

There are a wide class of other interconnection networks, including pancake graphs [1], hierarchical cubic networks (HCNs) [12], hierarchical folded-hypercube networks (HFNs) [10], hypernets [15], transposition networks [16,18] recursively connected complete (RCC) networks [13], tightly connected networks (TCNs) [6], hierarchical swapped networks (HSNs) [26], multi-dimensional hypernets [25], and recursive hierarchical fully-connected (RHFC) networks [25], which are also recursively interconnected as complete graphs. By using techniques introduced in this paper, we can also obtain the best known VLSI layouts for the above networks. These techniques can also be applied to a variety of other hierarchical networks, such as WK recursive networks [9] and cyclic networks [27].

4. Conclusion

In this paper, we have presented efficient layouts for complete graphs and star graphs. We showed that an N -node complete graph could be optimally laid out using $\lfloor N^2/4 \rfloor$ tracks for a colinear layout, and could be laid out in $N^4/16 + o(N^4)$ area for a 2D layout. We also showed that an N -node star graph could be laid out in $N^2/16 + o(N^2)$ area. This solved an open question posed by Akers and Krishnamurthy. The areas of the layouts for complete graphs and star graphs are both optimal within a factor $1 + o(1)$. Our layout methods can also be extended to a variety of other hierarchical networks.

References

- [1] S.B. Akers, B. Krishnamurthy, A group theoretic model for symmetric interconnection networks, in: Proc. Internat. Conf. Parallel Processing, 1986, pp. 216–223.
- [2] S.B. Akers, D. Harel, B. Krishnamurthy, The star graph: an attractive alternative to the n -cube, in: Proc. Internat. Conf. Parallel Processing, 1987, pp. 393–400.

- [3] S.G. Akl, K.A. Lyons, *Parallel Computational Geometry*, Prentice-Hall, Englewood Cliffs, NJ, 1993.
- [4] N. Bagherzadeh, M. Dowd, S. Latifi, A well-behaved enumeration of star graphs, *IEEE Trans. Parallel Distrib. Systems* 6 (5) (1995) 531–535.
- [5] G. Brebner, Relating routing graphs and two-dimensional grids, in: *VLSI: Algorithms and Architectures*, 1985, pp. 221–231.
- [6] P.T. Breznay, M.A. Lopez, Tightly connected hierarchical interconnection networks for parallel processors, in: *Proc. Internat. Conf. Parallel Processing*, Vol. I, 1993, pp. 307–310.
- [7] C. Chen, D.P. Agrawal, dBCube: a new class of hierarchical multiprocessor interconnection networks with area efficient layout, *IEEE Trans. Parallel Distrib. Systems* 4 (12) (1993) 1332–1344.
- [8] K. Day, A. Tripathi, A comparative study of topological properties of hypercubes and star graphs, *IEEE Trans. Parallel Distrib. Systems* 5 (1) (1994) 31–38.
- [9] G. Della Vecchia, C. Sanges, A recursively scalable network VLSI implementation, in: *Future Generations Comput. Systems*, 1988, pp. 235–243.
- [10] D. Duh, G. Chen, J. Fang, Algorithms and properties of a new two-level network with folded hypercubes as basic modules, *IEEE Trans. Parallel Distrib. Systems* 6 (7) (1995) 714–723.
- [11] A. Fernández, K. Efe, Efficient VLSI layouts for homogeneous product networks, *IEEE Trans. Comput.* 46 (10) (1997) 1070–1082.
- [12] K. Ghose, R. Desai, Hierarchical cubic networks, *IEEE Trans. Parallel Distrib. Systems* 6 (4) (1995) 427–435.
- [13] M. Hamdi, A class of recursive interconnection networks: architectural characteristics and hardware cost, *IEEE Trans. Circuits and Sys., I: Fundamental Theory and Applications* 41 (12) (1994) 805–816.
- [14] D.A. Hoelzeman, S. Bettayeb, On the genus of star graphs, *IEEE Trans. Comput.* 43 (6) (1994) 755–759.
- [15] K. Hwang, J. Ghosh, Hypernet: a communication efficient architecture for constructing massively parallel computers, *IEEE Trans. Comput.* 36 (12) (1987) 1450–1466.
- [16] S. Latifi, P.K. Srimani, Transposition networks as a class of fault-tolerant robust networks, *IEEE Trans. Parallel Distrib. Systems* 45 (2) (1996) 230–238.
- [17] F.T. Leighton, *Complexity Issues in VLSI: Optimal Layouts for the Shuffle-exchange Graph and Other Networks*, MIT Press, Cambridge, MA, 1983.
- [18] F.T. Leighton, *Introduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercubes*, Morgan-Kaufman, San Mateo, CA, 1992.
- [19] C.E. Leiserson, Fat-trees: universal networks for hardware-efficient supercomputing, *IEEE Trans. Comput.* C-34 (10) (1985) 892–901.
- [20] V.E. Mendia, D. Sarkar, Optimal broadcasting on the star graph, *IEEE Trans. Parallel Distrib. Systems* 3 (4) (1992) 389–396.
- [21] D.K. Saikia, R.K. Sen, Two ranking schemes for efficient computation on the star interconnection networks, *IEEE Trans. Parallel Distrib. Systems* 4 (1996) 321–327.
- [22] Šýkora, I. Vr'no, On VLSI layouts of the star graph and related networks, *Integration, VLSI J.* (1994) 83–93.
- [23] C.D. Thompson, Area-time complexity for VLSI, in: *Proc. ACM Symp. Theory of Computing*, 1979, pp. 81–88.
- [24] Y.-C. Tseng, J.-P. Sheu, Toward optimal broadcast in a star graph using multiple spanning trees, *IEEE Trans. Comput.* 46 (5) (1997) 593–599.
- [25] C.-H. Yeh, B. Parhami, Recursive hierarchical fully-connected networks: a class of low-degree small-diameter interconnection networks, in: *Proc. IEEE Internat. Conf. Algorithms and Architectures for Parallel Processing*, June 1996, pp. 163–170.
- [26] C.-H. Yeh, B. Parhami, Recursive hierarchical swapped networks: versatile interconnection architectures for highly parallel systems, in: *Proc. IEEE Symp. Parallel and Distributed Processing*, October 1996, pp. 453–460.
- [27] C.-H. Yeh, B. Parhami, Cyclic networks—a family of versatile fixed-degree interconnection architectures, in: *Proc. Internat. Parallel Processing Symp.*, April 1997, pp. 739–743.
- [28] C.-H. Yeh, B. Parhami, Tight bounds on the VLSI areas and bisection width of star graphs and HCNs, submitted.