Cyclic Networks:
A Family of Versatile Fixed-Degree Interconnection Architectures

Chi-Hsiang Yeh and Behrooz Parhami
Department of Electrical and Computer Engineering
University of California, Santa Barbara, CA 93106-9560, USA
{yeh@engineering| parhami@ece}.ucsb.edu

Abstract

In this paper, we propose a new family of interconnection networks, called cyclic networks (CNs), *in which an inter-cluster connection is defined on a set of nodes whose addresses are cyclic shifts of one another. The node degrees of basic CNs are independent of system size, but can vary from a small constant (e.g., 3) to as large as required, thus providing flexibility and effective tradeoff between cost and performance. The diameters of suitably constructed CNs can be asymptotically optimal within their lower bounds, given the degrees. We show that packet routing and ascend/descend algorithms can be performed in $\Theta(\log_d N)$ communication steps on some CNs with N nodes of degree $\Theta(d)$. Moreover, CNs can also efficiently emulate homogeneous product networks (e.g., hypercubes and high dimensional meshes). As a consequence, we obtain a variety of efficient algorithms on such networks, thus proving the versatility of CNs.*

1 Introduction

The design of interconnection architectures may significantly affect several characteristics of the final parallel systems, such as performance, ease of programming, reliability, scalability, complexity of physical layout, and hardware cost. In general, high performance, fixed node degree, and simplicity of algorithms are some of the desirable, but often conflicting, requirements. Hypercubes, star graphs [1] and generalized hypercubes [3] have appealing topological, algorithmic, and fault tolerance properties, but they tend to have high node degrees for large system sizes.

To overcome the problem of unbounded node complexity in large hypercube or star networks, some constant-degree variants or alternatives, such as the cube-connected cycles (CCC) [14], shuffle-exchange, de Bruijn graph, butterfly networks [13], and star connected cycles (SCC) [12] have been introduced and shown to have some desirable properties. Other interconnection networks, such as hypernets [10], hierarchical shuffle-exchange (HSE) networks [5], and WK-recursive networks [15] may also use nodes of small (constant) degrees. When high performance and moderate cost are required, networks that have better topological, algorithmic, and fault tolerance properties than constant-degree networks and have intermediate node degrees (e.g., higher than 3, but lower than $\log_2 N$) may become attractive. Several such interconnection schemes for parallel architectures have been proposed in recent years [4, 6, 8, 9, 16, 17].

In this paper, we propose a new family of fixed-degree interconnection networks called *cyclic networks (CNs)*, which compare favorably with the above interconnection networks in terms of implementation cost and performance of many important algorithms [18]. A CN can use identical copies of any small network as its basic modules, connected through a set of nodes whose addresses are cyclic shifts of one another. The required data movement when performing many important algorithms on CNs is largely confined within basic modules, thus leading to small network delay and high throughput when the delay associated with transporting a packet through an intra-module (e.g., chip or board) link is small. To sum up, the proposed topologies possess many advantages, including 1) small (fixed) node degree, 2) small (optimal) diameter, 3) simple and efficient routing and ascend/descend algorithms, 4) efficient emulation (embedding) of popular topologies, 5) effective tradeoffs between cost and performance, 6) balanced traffic, adaptive to various communication patterns, and 7) suitability for VLSI implementation.

2 Basic cyclic networks

In this section, we define basic cyclic networks, also called ring-cyclic networks (Ring-CNs), explore some of their basic properties, and introduce the needed notation. For convenience, for any $j_1 \geq j_2$, we let $Z_{j_1:j_2}$ denote $Z_{j_1}Z_{j_1-1}\cdots Z_{j_2}$, where Z can be any symbol, such as U, V or X.

1063-7133/97 $10.00 © 1997 IEEE

Definition 2.1 (Ring-Cyclic Network): Let the nucleus graph be $G = (\mathcal{V}_G, \mathcal{E}_G)$. An l-level ring-cyclic network based on the nucleus G is defined as the graph Ring-CN$(l,G) = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{V = V_{l:1} | V_i \in \mathcal{V}_G, i = 1,...,l,\}$ is the set of vertices, and $\mathcal{E} = \{(U = U_{l:1}, V = V_{l:1}) | U_i, V_i \in \mathcal{V}_G, i = 1,2,...,l,$ satisfying $U_{l:2} = V_{l:2}$ and $(U_1, V_1) \in \mathcal{E}_G$, or $U_i = V_{(i \bmod l)+1}$, or $V_i = U_{(i \bmod l)+1}$, for $1 \le i \le l\}$ is the set of edges.

The nucleus G can be a mesh, torus, hypercube, complete graph, generalized hypercube, ring, Petersen graph, a group of nodes connected to a common bus, or any other connected graph/hypergraph (of more than one node).

In other words, two nodes U and V are connected by an undirected link if nodes U and V are neighbors within the same nucleus G, or the addresses of nodes U and V are cyclic shifts of the l-symbol address of one another. We denote the address obtained by i right shifts as $X^{(i)}$. That is, $X^{(0)} = X$ and $X^{(i)} = X_{i:1}X_{l:i+1}$ for $1 \le i < l$, where $X = X_{l:1}$. Note that $X^{(i)} = X^{(i \bmod l)}$. Each node is connected to its neighbors in the same nucleus by *nucleus links* and to two others via the *left- (right-) shift links.*

Note that a node with the same l symbols in its address has no shift links (or, alternatively, has shift links connecting to itself) and are called the *leaders.* Leaders can be used as I/O ports or be connected to other leaders via their unused ports to provide better fault tolerance or to improve the performance and reduce the diameter of Ring-CNs without increasing the (maximum) node degree of the network.

2.1 Routing in CNs

In this subsection, we present a routing algorithm to route a packet from node X to node Y in a Ring-CN(l,G) using left- (or right-) shift links and nucleus links.

Suppose that a routing algorithm for the nucleus G is known. Let the addresses of nodes X and Y within the Ring-CN(l,G) be $X_{l:1}$ and $Y_{l:1}$, respectively, where $X_i, Y_i \in \mathcal{V}_G$.

```
Route(X,Y)
  for i = l downto 1 (or i = 1 to l) do
    begin
      Route the packet to node Y_i (or Y_((i mod l)+1))
        within the nucleus in which the packet
        currently resides.
      if i ≠ 1 (or i ≠ l) then send the packet through
        the left-shift (or right-shift) link.
    end
```

Routing from node $X = X_l \cdots X_3X_2X_1$ to $Y = Y_l \cdots Y_3Y_2Y_1$ within a CN(l,G) using left-shift links and nucleus links can be expressed as follows:

$$X_{l:2}X_1 \xrightarrow{\text{nucleus}} X_{l:2}Y_l \overset{\text{left-shift}}{\underset{}{\xrightarrow{\text{link}}}} X_{l-1:2}Y_lX_l \xrightarrow{\text{nucleus}}$$

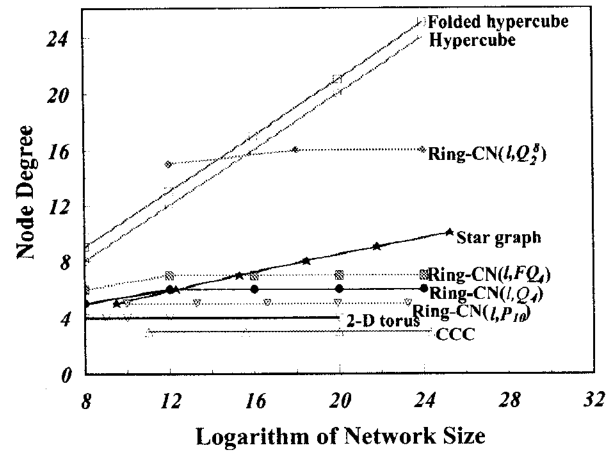

Figure 1. The node degree of several interconnection networks.

$$X_{l-1:2}Y_lY_{l-1} \overset{\text{left-shift}}{\xrightarrow{\text{link}}} X_{l-2:2}Y_{l:l-1}X_{l-1} \longrightarrow \cdots \longrightarrow Y_{l:1}.$$

If the routing algorithm on the nucleus G requires at most $T_R(1,G)$ time, the routing algorithm on Ring-CN(l,G) requires time at most $T_R(l,G) = lT_R(1,G) + l - 1$.

When we use a hypercube or folded hypercube as the nucleus graph of a CN, the expected traffic on the network links will be approximately uniform. For other nucleus graphs, such as a complete graph or higher-radix hypercube, we can also find a corresponding enhanced CN, introduced in Section 4, that has approximately uniform traffic, assuming unit link capacity.

2.2 Basic topological properties

Let the nucleus G be a graph with M nodes of degree d_G. The number of nodes in a Ring-CN is increased by a factor of M when the level is increased by 1. Thus, the number N of nodes of a Ring-CN(l,G) is $N = M^l$, and the level of Ring-CN(l,G) of N nodes is $l = \log_M N$. The (maximum) node degree of a Ring-CN$(2,G)$ is $d = d_G + 1$, and the degree of a Ring-CN(l,G) with $l \ge 2$ is $d = d_G + 2$.

The diameter of a Ring-CN(l,G) is obtainable from the routing algorithm given in Subsection 2.1.

Theorem 2.1 *The diameter of a Ring-CN(l,G) is $l(D_G + 1) - 1$, where D_G is the diameter of the nucleus G.*

Note that if leaders are connected (e.g., via diameter links [8, 18]), the diameter may be reduced without increasing the maximum node degree.

CNs are compared to several popular networks with respect to node degree and network diameter in Figs. 1 and 2. CNs provide effective tradeoffs between node degree and network diameter, while remaining highly competitive with

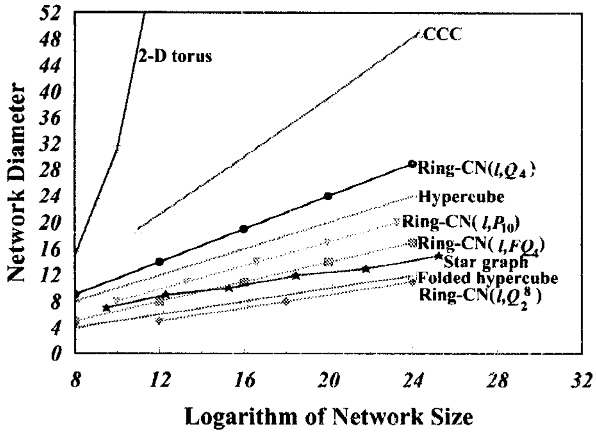

Figure 2. The diameter of several interconnection networks.

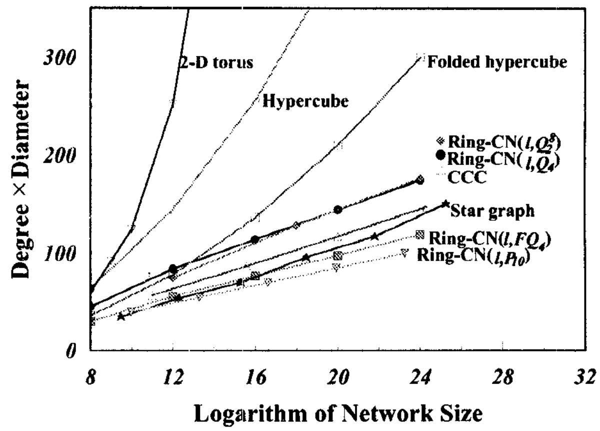

Figure 3. The product of node degree and network diameter of several interconnection networks.

regard to a composite cost measure [3], defined as the product of node degree and diameter (see Fig. 3).

It can be shown that the diameter of Ring-CNs based on a nucleus G is asymptotically optimal within a constant factor from the lower bound $\log_d N$, given its network size N and node degree d, if the nucleus has M nodes of degree d_G and has diameter $O(\log_{d_G} M)$ [18]. In particular, when the nucleus is an n-dimensional radix-r hypercube Q_n^r [3, 11], the ratio of the diameter to its lower-bound for the $\mathrm{CN}(l, Q_n^r)$ is optimal asymptotically with a constant factor 1 if $\log n = o(\log r)$ and n is not a constant in N [18].

3 Ascend/descend algorithms

Ascend/descend algorithms [14] require successive operations on data items that are separated by a distance equal to a power of 2. Many applications, such as Fast Fourier Transform (FFT), bitonic sort, matrix multiplication, and convolution, can be formulated using algorithms in this general category. In this section, we show that algorithms in this category can be performed efficiently on a Ring-CN as long as ascend/descend algorithms can be performed efficiently on its nucleus graph.

Assume that the nucleus G has $M = 2^k$ nodes. Let $i_a, i_b \in [0, lk-1]$, $a_1 = \lfloor i_a/k \rfloor$, $a_0 = i_a \bmod k$, $b_1 = \lfloor i_b/k \rfloor$, and $b_0 = i_b \bmod k$. We present the ascend algorithm $\mathrm{Asc}(i_a, i_b, l, G)$ (for operations on data separated by a distance 2^j, $j = i_a, i_a+1, ..., i_b$) on Ring-CN(l, G) as follows:

Asc(i_a, i_b, l, G)
begin
  if $a_1 \le l/2$ **then** Each node repeatedly sends data via its right-shift link a_1 times,
  else Each node repeatedly sends data via its left-shift link $l - a_1$ times.
  for $i := a_1$ to b_1 **do**
    begin
      $c_0 :=$ **if** $i = a_1$ **then** a_0 **else** 0.
      $d_0 :=$ **if** $i = b_1$ **then** b_0 **else** $k-1$.
      Perform $\mathrm{ASC}(c_0, d_0, 1, G)$.
      Each node sends data via its right-shift link.
    end
  if $b_1 \ge l/2$ **then** Each node repeatedly sends data via its right-shift link $l - b_1$ times,
  else Each node repeatedly sends data via its left-shift link b_1 times.
end

Note that throughout the paper, if a node does not have the specified link, the corresponding operation is simply skipped for that node.

By performing the sending step via right-shift (or left-shift) link i times, node X will hold the data item from node $X^{(i)}$ (or $X^{(-i)}$). In essence, this moves data items separated by a distance of 2^j, $j = ki, ki+1, ..., ki+k-1$, into the same nucleus, such that they are now separated by a distance of 2^{j-ki}. Thus, we can use the ascend algorithms on the nucleus G (i.e., Ring-CN$(1, G)$) to emulate the step needed in dimensions j, $j = ki, ki+1, ..., ki+k-1$.

It can be easily verified that ascend/descend algorithms (for all the $\log_2 N$ possible operations) can be performed in $(1 + 1/k)\log_2 N$ time on a Ring-CN based on hypercubes, and can be performed using $\frac{2\log_2 N}{k}$ communication steps on a Ring-CN based on complete graphs, where N is the number of nodes in Ring-CNs and and 2^k is number of nodes in the nucleus graph. We can also trade the number of communication steps with computation steps for Ring-CNs based on complete graphs. For example, FFT can be performed using $\frac{(1+t)\log_2 N}{k}$ communication steps and $\frac{t(2^{k/t}-1)\log_2 N}{k}$ computation steps on an N-node Ring-CN(l, K_{2^k}) or an N-node Ring-CN$(l, Q_{\frac{t}{v}}^{2^{kv/t}})$, assuming that t divides k and v divides t, where K_{2^k} is a 2^k-node complete graph. Detailed results for various important algorithms and comparisons with other popular networks can be found in [18].

4 Enhancements and variants

In this section, we present the constructions of enhanced CNs and several variant constructions for CNs.

An l-level CN based on G, CN(l,G), is obtained by removing all shift links from a Ring-CN(l,G) and reconnecting nodes $X, X^{(1)}, X^{(2)}, \ldots, X^{(l-1)}$. The only rule required for the connection is that nodes $X, X^{(1)}, X^{(2)}, \ldots, X^{(l-1)}$, which is a cyclic-shift (CS) graph of the CNs, have to form a connected graph (or hypergraph), for each node X in the CN.

4.1 Complete-cyclic networks

Complete-cyclic networks (Complete-CNs) are obtained by replacing the rings in Ring-CNs with complete graphs, which are the CS graphs of the Complete-CNs.

Definition 4.1 (Complete-Cyclic Network): Let the nucleus graph be $G = (\mathcal{V}_G, \mathcal{E}_G)$. An l-level complete-cyclic network based on the nucleus G is defined as the graph Complete-CN$(l,G) = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{V = V_{l:1} | V_i \in \mathcal{V}_G, i = 1, \ldots, l,\}$ is the set of vertices, and $\mathcal{E} = \{(U = U_{l:1}, V = V_{l:1}) | U_i, V_i \in \mathcal{V}_G, i = 1, 2, \ldots, l$, satisfying $U_{l:2} = V_{l:2}$ and $(U_1, V_1) \in \mathcal{E}_G$, or $V = U^{(i)}$, for some integer i, $1 \leq i < l\}$ is the set of edges.

Clearly, a Ring-CN(l,G) is a subgraph of a Complete-CN(l,G) since a ring is a subgraph of a complete graph.

Homogeneous product networks (HPN) form a subclass of product networks with identical component networks [7]. More precisely, an HPN is the iterated Cartesian product of the same graph, and HPN(p,G) denotes the p^{th} *power of* G; that is, HPN$(p,G) = \prod_{i=1}^{p} G = \underbrace{G \times G \times \cdots \times G \times G}_{p}$. A dimension-$n$ radix-M (generalized) hypercube is an HPN(n, K_M) and an M-ary n-cube is an HPN(n, R_M), where R_M is an M-node ring. Although Complete-CNs have higher node degrees than Ring-CNs, they can emulate HPNs with optimal slowdown under both the single-dimension [16, 17] and all-port communication models.

Theorem 4.1 *Any HPN(l,G) algorithm that only uses links of the same dimension at a time can be emulated on a Complete-CN(l,G) with a slowdown factor of 3.*

If the algorithm can be emulated using dynamic emulation [18], the slowdown factor can be reduced to 2.

Theorem 4.2 *Any step of an N-node hypercube algorithm with all-port communication can be optimally emulated on a Complete-CN(l, Q_k) that uses N nodes of degree $\Theta(\sqrt{\log N})$ in $\Theta(\sqrt{\log N})$ steps.*

As a result of Theorem 4.2, many Complete-CN algorithms can be performed in optimal time by simply emulating hypercube algorithms.

Corollary 4.3 *The total exchange task can be optimally executed on a Complete-CN(l, Q_k) that uses N nodes of degree $\Theta(\sqrt{\log N})$ in $\Theta(N\sqrt{\log N})$ time, under the all-port communication model.*

Corollary 4.4 *The multiple-node broadcast task can be optimally executed on a Complete-CN(l, Q_k) with N nodes of degree $\Theta(\sqrt{\log N})$ in $\Theta(N/\sqrt{\log N})$ time, under the all-port communication model.*

We can also use certain loop-based topologies [2], any other previously proposed network with proper size, or a specially designed small graph to replace the CS graphs (e.g., rings (or complete graphs) consisting of nodes $X, X^{(1)}, X^{(2)}, \ldots, X^{(l-1)}$ in a Ring-CN(l,G) (or Complete-CN(l,G))). Some examples and the proofs for the above theorem and corollaries can be found in [18].

4.2 Incomplete CNs

To obtain variants of CNs with smaller step-size, we can use $M_l M^{l-1}$ rather than M^l nodes to construct l-level incomplete CNs, where M_l divides M, the number of nodes in a nucleus. A node $X = X_l X_{l-1:1}$ in the incomplete variant is assigned a pseudo-address $X = X'_l X_{l-1:1}$ such that X'_l ranges from 0 to $M-1$ (rather than 0 to $M_l - 1$ for X_l).

For example, node $X_l X_{l-1:1}$ can be assigned the pseudo-address $X'_l X_{l-1:1}$, with $X'_l = X_Q M_l + X_l$, where $X_Q = X_1 \bmod M_l$. Given the pseudo-address, we can then construct incomplete CNs using previous definitions. We can also use methods similar to those used for HSE [5]. Most results derived in this paper can be applied to such CN variants either directly or with minor modifications.

4.3 Recursive CNs and related networks

Another way to obtain a CN with node degree and performance similar to CNs that use loop topologies as their CS graphs is to recursively construct the CN based on smaller CNs (e.g., Ring-CN, Complete-CN) as the nuclei. The formal definition is given as follows.

Definition 4.2 (Recursive Cyclic Networks): An r-deep Recursive-CN$(l_r, l_{r-1}, \ldots, l_1, G)$ is recursively defined as CN$(l_r$, Recursive-CN$(l_{r-1}, l_{r-2}, \ldots, l_1, G))$, with Recursive-CN$(l_1, G)$ = CN(l_1, G).

Most algorithms developed for CNs can be recursively applied to Recursive-CNs with minor modifications. It is worth noting that when the nucleus graph is a hypercube, a $\mathrm{CN}(2, Q_n)$ becomes a hierarchical cubic network $\mathrm{HCN}(n,n)$ without diameter links [8]; when the nucleus graph is a folded hypercube, a $\mathrm{CN}(2, FQ_n)$ becomes a hierarchical folded-hypercube network $\mathrm{HFN}(n,n)$ [6]; when the nucleus graph is a 1-cube (i.e., two connected nodes), a Ring-$\mathrm{CN}(l, Q_1)$ becomes an l-dimensional shuffle-exchange (SE) network; and when the nucleus graph is a shuffle-exchange network, a Ring-$\mathrm{CN}(l, SE)$ becomes an HSE [5]. Moreover, recursively connected complete (RCC) networks [9] also form a subclass of Recursive-CNs that are recursively constructed using 2-level CNs. That is, (M,r)-RCC is equivalent to $CN(\underbrace{2,...,2}_{r}, G) = \underbrace{\mathrm{CN}(2,\mathrm{CN}(2,...,\mathrm{CN}}_{r}(2,G)...))$, where M is the number of nodes in the nucleus G.

5 Conclusion

In this paper, we have proposed CNs as a new family of parallel interconnection architectures. CNs not only combine some desirable properties of both the hypercube (e.g., a wealth of fast, elegant algorithms) and the star graph (e.g., optimal diameter), but also use nodes of small (constant) degree, making them less expensive to implement and easier to expand. Some basic topological properties were presented and compared to other popular topologies. We also developed efficient routing, ascend/descend, and emulation algorithms on CNs. In addition to these advantages, we have shown in [18] that CNs and their variants significantly outperform other popular interconnection networks when off-module bandwidth is the limiting factor on system performance. Further research can be carried out in many directions, including the performance of CNs based on various routing schemes, their fault-tolerance properties, development and refinement of important algorithms, VLSI layout issues, and packaging considerations.

References

[1] Akers, S.B., D. Harel, and B. Krishnamurthy, "The star graph: an attractive alternative to the n-cube," *Proc. Int'l Conf. Parallel Processing*, 1987, pp. 393-400.

[2] Arden, B.W. and H. Lee, "Analysis of chordal ring networks," *IEEE Trans. Comput.*, vol. C-30, Apr. 1981, pp. 291-296.

[3] 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.

[4] 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.

[5] Cypher, R. and J.L.C. Sanz, "Hierarchical shuffle-exchange and de Bruijn networks," *Proc. IEEE Symp. Parallel and Distributed Processing*, 1992, pp. 491-496.

[6] 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.

[7] Efe, K. and A. Fernandez, "Products of networks with logarithmic diameter and fixed degree," *IEEE Trans. Parallel Distrib. Sys.*, vol. 6, no. 9, Sep. 1995, pp. 963-975.

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

[9] 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.

[10] 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.

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

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

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

[15] Vecchia, G.D. and C. Sanges, "Recursively scalable networks for message passing architectures," *Proc. Conf. Parallel Processing and Applications*, 1987, pp. 33-40.

[16] Yeh, C.-H. and B. Parhami, "Recursive hierarchical swapped networks: versatile interconnection architectures for highly parallel systems," *Proc. IEEE Symp. Parallel and Distributed Processing*, 1996, pp. 453-460[1].

[17] Yeh, C.-H. and E.A. Varvarigos, "Macro-star networks: efficient low-degree alternatives to star graphs for large-scale parallel architectures," *Proc. Symp. Frontiers of Massively Parallel Computation*, 1996, pp. 290-297[1].

[18] Yeh, C.-H. and B. Parhami, "Cyclic networks: a class of scalable communication-efficient interconnection architectures," Dept. Elec. & Comput. Engr., Univ. California, Santa Barbara, CA, Tech. Rep. no. 97-06, 1997 [1].

[1]References [16] through [18] are currently available at http://www.engineering.ucsb.edu/~yeh.