Design of High-Performance Massively Parallel Architectures Under Pin Limitations and Non-Uniform Propagation Delay

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

Inter-module bandwidth is one of the major constraints on the performance of current and future parallel systems. In this paper, we propose and evaluate several high-performance bus-based parallel architectures, including bus-based cyclic networks (BCNs) and quotient cyclic networks (BQCNs), which are particularly efficient in view of their respective inter-module communication patterns. The inter-cluster connection in a BCN is defined on a set of nodes whose addresses are cyclic shifts of one another. The node degree of a basic BCN is 3; while those of BQCNs and enhanced BCNs can vary from a small constant (e.g., 2) to as large as required, thus providing flexibility and effective tradeoff between cost and performance. A variety of algorithms can be performed efficiently on these networks, thus proving the versatility of BCNs and BQCNs.

1 Introduction

The design of the interconnection architecture is one of the most important and difficult tasks for a high-performance multiprocessor system. The choice of the interconnection topology may affect several characteristics of the final system, such as performance, ease of programming, reliability, scalability, and complexity of physical layout. Hypercubes, star graphs [1] and generalized hypercubes [4] have desirable 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 I/O bounded variants or alternatives, such as the cube-connected cycles (CCC) [21], shuffle-exchange, de Bruijn graph, butterfly networks [18], periodically regular chordal rings [19], and star connected cycles (SCC) [2, 17] have been introduced and shown to have some desirable properties.

In addition to network latency, inter-module bandwidth is one of the major factors limiting the performance of current and future parallel systems. One of the main advantages of hypernets, a class of communication-efficient interconnection architectures, is that the required transmissions tends to be confined within basic modules [13]; that is, the required numbers of transmissions between basic modules are considerably fewer than those of other topologies, such as hypercubes and star graphs, for many communication problems. Other interconnection topologies that have such a localized communication property include SCC, WK-recursive networks [23], hierarchical shuffle-exchange (HSE) networks [5], hierarchical cubic networks (HCN) [9], hierarchical folded-hypercube networks (HFN) [7], recursively connected complete (RCC) networks [12], and swapped networks (SN) [25].

Pin minimization is another important issue for high-performance parallel architectures since the number of processors that can be placed on a chip or board is often limited by off-chip or off-board connections for large systems [3, 6, 8]. Several bus-based parallel architectures have been proposed in order to minimize the number of pins required per processor and/or to enhance their performance [8].

In this paper, we propose several parallel architectures, including *cyclic networks (CNs)* and *quotient cyclic networks (QCNs)*, which are particularly efficient in view of the required inter-module transmissions for many problems. The node degree of a basic BCN is 3; while those of BQCNs and enhanced BCNs can vary from a small constant (e.g., 2) to as large as required, thus providing flexibility and effective tradeoff between cost and performance. The diameters of BCNs and BQCNs and the required inter-module traffic for many problems are asymptotically optimal within a small constant factor from their lower bounds. A BCN 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 movements when performing many important algorithms on BCNs are largely confined within basic modules, thus leading to small network delay when the delay associated with transporting a packet through an on-module link is small.

The remainder of this paper is organized as follows. In

0-8186-7870-4/96 $10.00 © 1997 IEEE

Section 2, we define basic bus-based cyclic networks, derive their parameters, and present a simple and fast routing algorithm. In section 3, we show that BCNs and quotient BCNs are efficient in terms of inter-module communication. In Section 4, we generalize the construction to enhanced BCNs, and show that the results for basic BCNs can be easily generalized to the entire family. In Section 5, we compare and summarize the hardware properties of BCNs and some competing parallel architectures.

2 Basic Bus-Based Cyclic Networks

In this section, we give the definition of basic cyclic networks (basic CNs), also called ring-cyclic networks (Ring-CNs), explore some of the properties for basic bus-based CNs, and introduce the needed notation. A network is represented as a graph, whose nodes and edges represent processors and links, respectively. 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.

Definition 2.1 (Ring-Cyclic Network, Ring-CN(l,G)) Let the nucleus be $G = (\mathcal{V}_{\mathcal{G}}, \mathcal{E}_{\mathcal{G}})$. 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 $\mathcal{V} = \{V_{l:1} | V_i \in \mathcal{V}_{\mathcal{G}}, i = 1,...,l\}$ is the set of vertices, and $\mathcal{E} = \{(U_{l:1}, V_{l:1}) | U_i, V_i \in \mathcal{V}_{\mathcal{G}}, i = 1,2,...,l$, satisfying $U_{l:2} = V_{l:2}$ and $(U_1, V_1) \in \mathcal{E}_{\mathcal{G}}$, or $U_i = V_{(i \bmod l)+1}$, or $V_i = U_{(i \bmod l)+1}$, for $1 \leq i \leq l\}$ is the set of edges.

In other words, two nodes U and V are connected by an undirected link if the l-symbol addresses of nodes U and V are cyclic shifts of one another (see Fig. 1); nodes are also connected as the graph G if they are within the same nucleus. We denote the address obtained from X by i right shifts as $X^{(i)}$. That is, $X^{(0)} = X$ and $X^{(i)} = X_{i:1}X_{l:i+1}$ for $1 \leq i < l$, where $X = X_{l:1}$. Note that $X^{(i)} = X^{(i \bmod l)}$.

As in other hierarchical networks such as HCN, HFN, hypernets, RCC, and CCC, the choice of basic module is crucial to the performance of the resultant CNs. By using a bus-based nucleus, one can obtain the following three advantages for such networks: 1) smallest node degree, 2) smallest diameter, and 3) balanced traffic. Buslet is suitable for CNs since a routed packet tends to traverse from one node to any other node in the same nucleus with comparable probability. Balanced traffic for a BCN can be achieved by appropriately selecting the bandwidth of the buses within the nucleus. In this paper, we assume that a nucleus buslet is implemented on a single chip in order to increase the processor-memory and intra-nucleus bandwidths, or more generally, on a single module (e.g., a board). Similar assumptions are used for several networks, such as hypercubes and several parallel architectures proposed in [22, 15].

An l-level basic bus-based cyclic network (basic BCN(l,M) or Ring-BCN(l,M)), is obtained by replacing the nucleus G with M processors connected to a common bus, where M is the number of nodes in G. Thus, a Ring-BCN(l,M) has M^l nodes of degree at most equal to 3, and its nucleus is called an M-node buslet. Each node is connected to other nodes within the same nucleus via the bus and to two other neighbors via the *left- (right-) shift links*, where the *left- (right-) shift neighbors* of node X are $X^{(-1)}(X^{(1)})$.

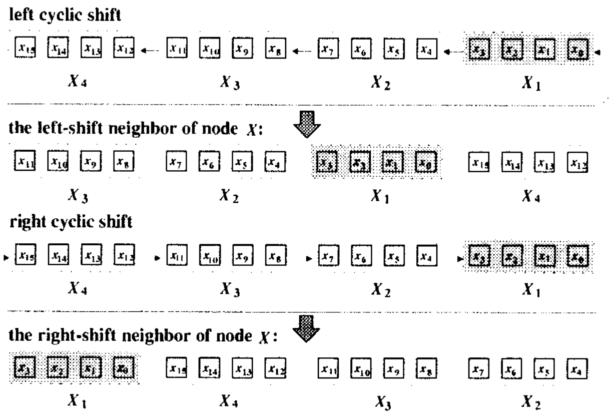
left cyclic shift
x_{15} x_{14} x_{13} x_{12}
x_{11} x_{10} x_9 x_8
x_7 x_6 x_5 x_4
x_3 x_2 x_1 x_0
X_4
X_3
X_2
X_1
the left-shift neighbor of node X:
x_{11} x_{10} x_9 x_8
x_7 x_6 x_5 x_4
x_3 x_2 x_1 x_0
x_{15} x_{14} x_{13} x_{12}
X_3
X_2
X_1
X_4
right cyclic shift
x_{15} x_{14} x_{13} x_{12}
x_{11} x_{10} x_9 x_8
x_7 x_6 x_5 x_4
x_3 x_2 x_1 x_0
X_4
X_3
X_2
X_1
the right-shift neighbor of node X:
x_3 x_2 x_1 x_0
x_{15} x_{14} x_{13} x_{12}
x_{11} x_{10} x_9 x_8
x_7 x_6 x_5 x_4
X_1
X_4
X_3
X_2

Figure 1. The derivation of shift links (neighbors) of a node $X = X_{4:1}$ in a basic CN$(4,G)$, where the nucleus G has M nodes with $8 < M \leq 16$. Symbols $X_i \in [0, M-1]$, $i = 1,2,3,4$, are represented by 4-bit binary numbers $x_{4i-1:4(i-1)}$.

Note that a node with l identical symbols in its address has no shift links (or, alternatively, has shift links connecting to itself) and is called a *leader*. 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 node degree of the network. If leader $X_{l:1}$ connects to leader $Y_{l:1}$ where $X_i = X_j, Y_i = Y_j$ and $X_i = M - Y_i - 1$, $i, j = 1,2,...,l$, the average distance between nodes and, in most cases, the diameter of the network will be reduced. This type of Ring-CN is called *Ring-CN with diameter links*. Varying the connectivity between leaders results in other classes of Ring-CNs.

2.1 Rings within Basic BCNs

Let $X = X_{l:1}$ be a node in a Ring-BCN(l,M), where $X_i \in [0, M-1]$, $X \neq X^{(i)}$ for $i = 1,2,...,l-1$. It can be seen that by the definition of Ring-BCNs, nodes X, $X^{(1)}$, $X^{(2)}$, ..., $X^{(l-1)}$ form an l-node ring, connected through shift links. In general, the majority of rings formed by the shift links are of this type.

However, when l is not a prime number, there will also be shorter rings with l_f nodes, where l_f divides l. For example, let $l = 8$, node $X = X_{8:5}X_{4:1}$ with $X_{8:5} = X_{4:1}$ is identical to node $X^{(4)}$; X thus forms a ring with nodes $X^{(1)}, X^{(2)}$ and $X^{(3)}$. Since the addresses of shift neighbors are obtained by performing cyclic shift on the address of a node, and these derived neighbors form a ring, we call such networks "ring-cyclic" networks. The rings with l nodes are called the *cyclic-shift (CS) graphs* of the BCN; the rings with l_f nodes are called the *degenerate CS graphs* of the BCN.

2.2 Routing and Broadcasting Algorithms

In this subsection, we present an algorithm to route a packet from node X to node Y in a Ring-BCN(l,M) using left- (right-) shift links and buses. We also show that broadcasting can be performed using a similar method.

Let the addresses of nodes X and Y within the Ring-BCN(l,M) be $X_{l:1}$ and $Y_{l:1}$, respectively, where $X_i, Y_i \in [0, M-1]$.

```
Route($X,Y$)
    for $i = l$ downto 1 (or $i = 1$ to $l$) do
        begin
            Send the packet to node $Y_i$ (or $Y_{(i \bmod l)+1}$)
                within the nucleus in which the packet
                currently resides.
            If $i \neq 1$ (or $i \neq l$) then
                send the packet through the left-shift
                (right-shift) link.
        end
```

It can be seen that only left-shift or right-shift operation and nucleus transmissions suffice for packet routing in a BCN. In fact, this property also holds for many other useful algorithms, such as ascend/descend algorithms. As a result, we can use directed links to implement BCNs, called directed BCNs, whose left-shift and right-shift links are implemented using output and input links, respectively, in order to reduce the number of off-module pins required by a factor of 2. The routing algorithm on Ring-BCN(l,M) requires time at most $T_R(l) = 2l - 1$.

Broadcasting can also be performed with a simple and fast algorithm on Ring-BCNs. To execute non-overlapping broadcasting in optimal time, we simply replace the step "send the packet within the nucleus" with "broadcast within the nucleus."

2.3 Basic Topological Properties

Let M be the number of nodes in a nucleus buslet. The number of nodes N of a BCN(l,M) is $N = M^l$. The level of Ring-BCN(l,M) of N nodes is $l = \frac{\log_2 N}{\log_2 M}$. The node degree of a Ring-BCN$(2,M)$ is 2, and the node degree of a Ring-BCN(l,M) with $l > 2$ is 3. The diameter of a Ring-BCN(l,M) is obtainable from the routing algorithm given in Subsection 2.2.

Theorem 2.1 *The diameter of a Ring-BCN(l,M) is $2l-1$.*

Proof: It can be shown that the routing algorithm presented in Subsection 2.2 is optimal for routing from node $X = \underbrace{X'X'\cdots X'}_{l}$ to node $Y = \underbrace{Y'Y'\cdots Y'}_{l}$, where $X' \neq Y'$. Thus, the time complexity of the algorithm provides both an upper bound and a lower bound for the diameter of the Ring-BCN(l,M). □

Using proofs similar to those in [25], we can show that any network with both the maximum number of processors connecting to a bus and the node degree not exceeding $O(M)$ have diameter at least in the same order as that of a similar sized basic BCN. In other words, the diameter of a basic BCN is asymptotically optimal.

3 Minimizing the Inter-Module Bandwidth Requirement

The locality of data access patterns for applications and the success of cache systems not only hide or eliminate the network latency but also help reduce the network bandwidth requirements (and thus inter-module traffic) in many current parallel computers. However, with the rapid advances in VLSI technologies, the number of processors and the computation capacity (e.g., MIPS) per chip are expected to grow significantly, making the above techniques alone inadequate for future parallel computing environments. As a consequence, the inter-module bandwidth is expected to become one of the major constraints on the performance of parallel computers.

In this section, we will first analyze the communication characteristics of BCNs. We will then propose new CN variant topologies that can take full advantage of the inter-module bandwidth available. Compared with other popular topologies, BCNs and their variants require significantly fewer inter-module transmissions for the same communication problems.

3.1 Communication Characteristics of BCNs

Assume that a 16-node nucleus buslet is implemented on a single chip as the basic building module. To build a 64K-node BCN multicomputer, 4K identical chips, each with 32 off-chip undirected links, are employed and interconnected using the cyclic shift rule (Definition 2.1). To perform packet routing for arbitrary uniformly distributed source and destination nodes, 2.9 inter-chip transmissions on the average and 3 inter-chip transmissions in the worst case are required.

For comparison, consider a 64K-node hypercube multicomputer built with 4K identical chips, each with a 16-node 4-cube and 192 off-chip undirected links, required by the hypercube connectivity rule. To perform packet routing for any source and destination, 12 inter-chip transmissions are required in the worst case and 6 on the average.

For fair comparison, we assume that the off-chip bandwidth per chip is the same for both cases, or equivalently, that the bandwidth of an off-chip link for the BCN is 6 times larger than that for the hypercube. The BCN can execute more routing tasks by a factor of 2.1 on the average and 4 in the worse case of both systems, assuming that the inter-chip (rather than intra-chip) bandwidth is the limiting factor. Note that the hardware costs for both systems will be approximately the same. On the other hand, this analysis also implies that the cost for a BCN can be considerably smaller than that of a hypercube system with similar performance.

In what follows, we will introduce CN variants that are even more efficient in terms of inter-chip communication.

3.2 Bus-Based Quotient CNs

In this subsection, we show that a new class of inter-processor topologies, called *quotient cyclic networks (QCNs)*, are highly communication-efficient. In particular, when the source and destination nodes are uniformly distributed over the network, the required inter-chip traffic is asymptotically optimal within a constant factor 1. Hypernets and basic cyclic networks are the only known topologies that have this desirable property, but they require that the number of processors per chip be comparable to the maximum number of pins per chip, which is not always possible. QCNs are proposed mainly for this reason. This strategy can also be applied to derive quotient hypernets to achieve similar characteristics. Note that the traffic on inter-chip links is not uniform for hypernets; while it is uniform for QCNs and CNs.

A QCN is obtained by "merging" several nodes within the same nucleus in a CN; a BQCN is derived from a BCN using the same method. For example, we can use one node in a QCN to replace 4 Ring-CN nodes in the same nucleus so that each node has 8 shift links connecting it to the 4 right- and 4 left-shift neighbors of the merged nodes. The QCN nodes in the same nucleus remain connected to a bus. In what follows, we analyze an example of BQCN and compare its inter-chip traffic requirement with a hypercube system of the same size using the same number of processors and pins per chip. We will focus on BQCN derived from directed Ring-BCN in order to minimize the required inter-chip transmissions.

Suppose that we want to build a 128K-node parallel computer. Assume that a nucleus buslet or hypercube with 8 processors can be accommodated on a single chip as the basic building module, whose maximum number of off-chip pins for connecting links is 256. To build a 128K-node BQCN multicomputer, 16K identical buslet chips are required. Since 16K is equal to $2^{14} = 128^2$, a directed Ring-BCN$(3, 128)$ seems suitable for the network size since the BCN has 16K nuclei in it. Since we need only 8 processors per chip, we can view each physical processor as representation 16 virtual nodes in the base 128^3-node topology (e.g., by merging nodes whose first 3 bits in the 7-bit addresses within the nucleus are the same. Since the number of required off-chip pins is $128 \cdot 2 = 256$, this construction is feasible under the assumption. To build a 128K-node hypercube multicomputer, 16K 3-cube chips are required. Such construction requires 224 off-chip pins (i.e., for 112 serial-in/-out links) is also feasible under the same assumption.

To perform packet routing for any source and destination, 1.84 inter-chip transmissions on the average (for uniformly distributed source and destination) and 2 inter-chip transmissions in the worst case are required in the BQCN multicomputer; while 7 inter-chip transmissions on the average and 14 inter-chip transmissions in the worst case are required in the hypercube multicomputer. The BCN can execute more routing tasks by an average factor of 3.8 and by a factor of 7 in the worse case of both systems, assuming that the inter-chip bandwidth, rather than the intra-chip bandwidth, is the limiting factor.

Note that BQCNs can use more processors per chip in order to obtain larger peak MIPS (or FLOPs) and to obtain larger processor-to-memory bandwidth per chip if the VLSI area per chip permits; while hypercubes cannot since the number of pins required per chip will exceed its limit. Also note that if the number of available pins per chip can be significantly increased as conjectured by the SIA projection [22], the superiority of BQCN over hypercube can be even more pronounced. Since the diameter of a 16K-node network with in-/out-degree 128 can be shown to be at least 2, it can be shown using similar proof that the required inter-chip traffic (in the worst case) for routing in the above BQCN is strictly optimal. Note that the average case is also close to optimal since 1.84 is close to the minimum possible average distance in such a 128K-node network. We formally present the result in the following lemma.

Lemma 3.1 *The average number of inter-nucleus transmissions required for routing in an N-node BQCN is* $\log_P N + o(\log_P N)$, *assuming uniformly distributed source and destination nodes, where P is the number of links per nucleus module.*

The technique used to derive QCNs from CNs (i.e., merging several appropriate nodes into one) can be applied to other networks, such as hypercubes, star graphs, CCC and SCC, in order to fully utilize the pins available, if the number of processors per chip is limited by area rather than by pin count. To eliminate the "number of parts" problem, we assume that a parallel system uses at most several types of modules as building blocks, and one or several cycles in the CCC or SCC can be built on the same module. Then, the average numbers of inter-module transmissions required for routing in N-node (quotient) hypercubes and (quotient) star graphs will be approximately $\frac{1}{2}\log_2 N - \frac{1}{2}\log_2 P$ and

$\frac{\log_2 N}{\log_2 \log_2 N} + o\left(\frac{\log N}{\log \log N} - \log P\right)$, respectively, assuming uniformly distributed source and destination nodes, where P is the number of links per module and $P > \frac{\log_2 N}{\log_2 \log_2 N}$ (so that at least one node in the N-node hypercube or star graph can be put onto a module). The corresponding parameters of (quotient) CCC and (quotient) SCC are close to those of similar-sized hypercubes and star graphs, respectively, when P is large. It is clear that BQCNs outperform the above networks and their quotient variants under this criterion.

Note that these desirable properties of BCNs and BQCNs can also be extended to the case when a nucleus is built with several chips. This is in fact another advantage of bus-based CNs since only one off-chip transmission is required for routing in a multi-chip buslet. In addition to their minimal inter-nucleus traffic, the off-chip bandwidth of BCNs and BQCNs can therefore be fully utilized. Similarly, if we replace the cycle in a SCC with a buslet, the off-chip bandwidth utilization can be improved in this case as well.

4 Enhanced BCNs and Variant Topologies

In this section, we present the constructions and properties of enhanced CNs, which can be generalized from the definition of basic CNs. Enhanced CNs have the advantage of better emulation capability compared with basic CNs. They have performance comparable to that of swapped networks and have choices of degrees more flexible than those of swapped networks [24, 25]. We will then define some subclasses of enhanced BCNs and explore their specific properties.

An l-level CN based on G, CN(l, G), is obtained by removing all shift links (rings) from a Ring-CN(l, G) and reconnecting nodes $X, X^{(1)}, X^{(2)}, ..., X^{(l-1)}$ in some other way. The new links used between nodes $X, X^{(1)}, X^{(2)}, ..., X^{(l-1)}$ have to form a connected graph (or hypergraph), for each node X in the CN. If $X^{(i)} \neq X^{(j)}$ for $i \neq j$, where $i, j = 0, 1, 2, ..., l-1$, the resultant connected graph is called the CS (cyclic-shift) graph; otherwise, it is called the degenerate CS graph. Note that there may be multiple links connecting two nodes in a degenerate CS graph. A CN using G_{CS} as its CS graphs is called G_{CS}-cyclic network and denoted by G_{CS}-CN.

4.1 Complete-BCNs

Complete-BCNs are obtained by replacing the rings in Ring-BCNs with complete graphs, as the (degenerate) CS graphs of the Complete-BCNs. Although the node degrees of Complete-BCNs are increased, they can emulate many networks efficiently. We formally define them as follows.

Definition 4.1 (Complete-CN(l, G)): Let the nucleus network 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 $\mathcal{V} = \{V_{l:1} | V_i \in \mathcal{V}_G, i = 1, ..., l, \}$ is the set of vertices, and $\mathcal{E} = \{(U_{l:1}, V_{l:1}) | U_i, V_i \in \mathcal{V}_G, i = 1, 2, ..., l,$ satisfying $U_{l:2} = V_{l:2}$ and $(U_1, V_1) \in \mathcal{E}_G$, or $V_{l:1} = U_{l:1}^{(j)}$, for some integer $j, 1 \leq j < l\}$ is the set of edges.

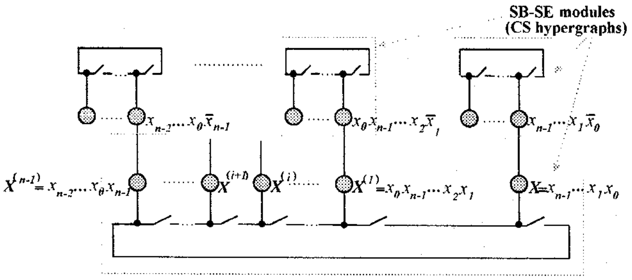
SB-SE modules
(CS hypergraphs)
$x_{n-2} \cdots x_0 \bar{x}_{n-1}$
$x_0 x_{n-1} \cdots x_2 \bar{x}_1$
$x_{n-1} \cdots x_1 \bar{x}_0$
$X^{(n-1)} = x_{n-2} \cdots x_0 x_{n-1}$
$X^{(i+1)}$
$X^{(i)}$
$X^{(1)} = x_0 x_{n-1} \cdots x_2 x_1$
$X = x_{n-1} \cdots x_1 x_0$

Figure 2. Structure of an n-dimensional SB-SE module and its off-module connections, where $X^{(i)} \neq X^{(j)}$ for $i \neq j$, $0 \leq i, j \leq n-1$ and $X = x_{n-1} x_{n-2} \cdots x_1 x_0$.

Note that the above definition implies the existence of multiple links and self loops in the degenerate CS graphs.

An l-level Complete-BCN is obtained by replacing the nucleus G with an M-node buslet. It can be seen that the Ring-BCN(l, M) is a subgraph of Complete-BCN(l, M) since a ring is a subgraph of a complete graph. A Complete-BCN(l, M) has the same size and diameter as a Ring-BCN(l, M), but it has a larger node degree l.

Emulation of an l-dimensional base-M generalized hypercube [4, 16] on a Complete-BCN(l, M) is simple and fast. For example, to emulate the transmission from node $X_4 X_3 X_2 X_1$ to node $X_4 X_3' X_2 X_1$ in a 4-dimensional generalized hypercube, we first send the packet from node $X = X_4 X_3 X_2 X_1$ to node $X^{(2)} = X_2 X_1 X_4 X_3$ using the appropriate inter-nucleus link, then to node $X' = X_2 X_1 X_4 X_3'$ within the nucleus, and finally to node $X'^{(2)} = X_4 X_3' X_2 X_1$ using the appropriate inter-nucleus link. The result is formalized in the following theorem.

Theorem 4.1 *A Complete-BCN(l, M) can emulate an l-dimensional base-M generalized hypercube with dilation of at most 3.*

4.2 Segmented-Bus-CNs

Although Complete-BCNs can emulate generalized hypercubes efficiently, their node degrees grow with l. A more cost-effective enhanced CN is to replace the rings in Ring-CNs with segmented buses [20], which results in a Segmented-Bus-CN (SB-CN). The main reason we are using such segmented buses is that they can emulate rings without slowdown, thus supporting emulation of the Ring-CN and efficient routing scheme.

An SB-BCN(l, M) can embed an l-dimensional base-M generalized hypercube of the same size with dilation 3. Note that the node degree of such an embedded generalized hypercube is lM while that of the SB-BCN is 2.

An attractive SB-CN subclass results if we choose the nucleus to be an edge (i.e., a 1-cube). This results in a SB-CN(l, Q_1) which can also be called l-dimensional SB-shuffle-exchange (SB-SE) network (see Fig. 2). To implement an SB-SE, we can put each CS graph (or a couple of them if they are small), which is a segmented buslet, onto a chip (or a module), where each node has at most one off-chip (or off-module) link connecting it to its nucleus neighbor. Such an arrangement requires only short segmented buses and one off-module link per node, leading to lower implementation cost. Note that the default implementation of other classes of CNs is to put a nucleus onto a single module.

4.3 Incomplete BCNs

To obtain variants of BCNs with smaller step-size, we can use $M_l M^{l-1}$ rather than M^l nodes to construct l-level incomplete BCNs, 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 $X_l X_{l-1:1}$ is assigned $X'_l X_{l-1:1}$, with $X'_l = X_Q M_l + X_l$, where $X_1 = X_Q M_l + X_R$, and X_Q, X_R are positive integers with $X_R < M_l$. Given the pseudo-address, we can then construct incomplete BCNs using previous definitions. Most results derived in this paper can be applied to such BCN variants either directly or with minor modifications.

4.4 Recursive BCNs

Another way to obtain a CN is to recursively construct the CN based on smaller CNs (e.g., Ring-BCN, Complete-BCN) as the nuclei. The formal definition is given as follows.

Definition 4.2 (Recursive-CN$(l_r, l_{r-1}, \ldots, l_1, G)$) *For $r > 1$, 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 BCNs can be recursively applied to recursive BCNs with minor modifications.

We have introduced several classes of BCNs and briefly shown their advantages. We can use the techniques introduced in [24] to transform the constructions of BCNs in order to obtain regular and symmetric variants. We can also obtain unfolded variants of CNs by using the techniques given in [24].

5 Cost-Performance Comparisons

In this section, we compare the hardware costs and network delays of BCNs with those of several other interconnection networks.

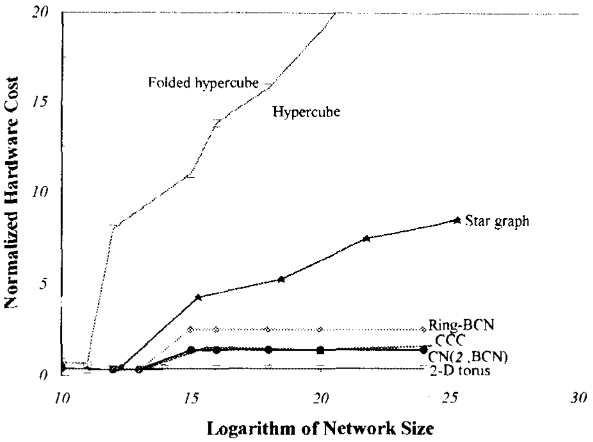

Figure 3. The normalized hardware cost, defined as the ratio of hardware cost over network size, for several interconnection networks.

An important advantage of BCNs, hypernets, and WK-recursive networks is that they can be hierarchically constructed using identical modules, with a fixed number of I/O ports independent of the network size. Each node in a hypernet or a WK-recursive network requires one external I/O port, and each node in a Ring-BCN requires one or two external I/O ports, connecting it to nodes in other modules. On the contrary, the number of required external I/O ports per node in the hypercube, folded hypercube, star graph, HCN, HFN, RCC or SN increases with the network size.

When interconnection networks grow very large, pin limitations become a major constraint in their VLSI implementation [3, 6, 12, 13]. In what follows, we compare the required total counts of PC boards and inter-board connectors for the implementation of BCNs and hypercubes. The analysis and assumptions are similar to those given in [10, 11, 12, 14]. We assume that a multiprocessor system is built incrementally using VLSI chips and PC boards. A chip can house many processors, and a PC board houses many of these chips. A chip is mainly constrained by technology in terms of the number of processors that can be packed onto it, and the number of its I/O pins. A PC board is mainly constrained by the number of chips that can be housed within a board and the number of I/O ports per board. Suppose we want to build a multiprocessor system with 32K processors, using an 8-node buslet or 3-cube as the basic module within a chip. In view of current technology, the following limits are assumed: (1) maximum I/O pins per chip = 256 (pin-limited), (2) maximum chips per board = 256 (area-limited), (3) maximum I/O ports per board = 2400 (connector-limited). Under the above limitations, an 8-node buslet or 3-cube with at most 16 inter-module links per node (each having two I/O pins, namely serial-in and serial-out), can be implemented on a single chip. A 7-cube can be ac-

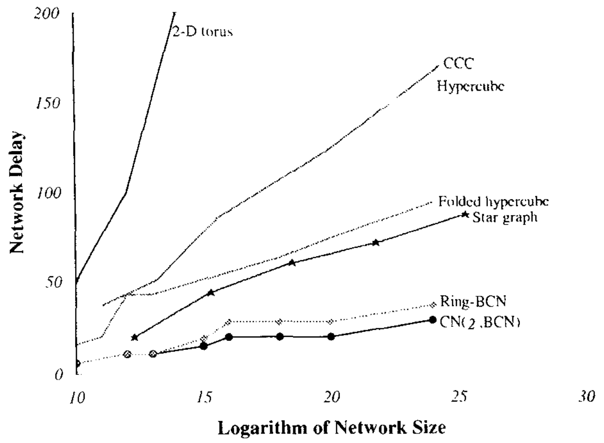

Figure 4. The network delays of several interconnection networks.

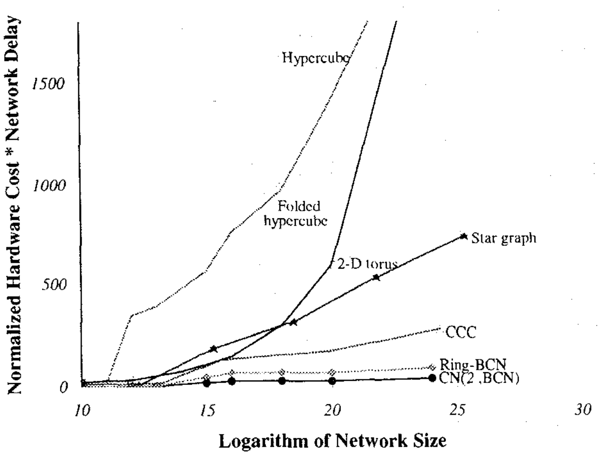

Figure 5. The product of normalized hardware cost and network delay for several interconnection networks.

commodated on a board in order to meet the I/O connector limit. Thus, 256 PC boards are required to implement a 15-cube. To optimize the performance and minimize the hardware cost of the desired BCN, we first construct two Ring-BCN(3,8) on a board. To implement a 32K-node BCN, we use 64 of the above Ring-BCNs to construct a 2-level incomplete CN. As a consequence, 32 PC boards are required for the incomplete CN(2,BCN(3,8)). The total number of boards and inter-board connectors required for the hypercube are 8 times those for the BCN.

The actual cost of a parallel architecture is known only after detailed designs for the various components are available and VLSI layout is attempted. However, there are simple rules of thumb, resulting from years of implementation experience, that can lead to reasonable preliminary estimates for comparison purposes. To provide such a comparison of the implementation costs of hypercube and BCN parallel systems of the same size, we assume that the costs of a board, a chip, and a connector are 100, 10, and 1, respectively [11, 12]. Under these assumptions, in the above example, the hardware cost for the hypercube is approximately 3.3 times that of the BCN. To compare the network delay, we assume that the delays associated with transporting a packet through an inter-board link, an inter-module link, and an intra-chip link are 8, 4, and 1, respectively [11, 12]. Under this assumption, the network delays for the hypercube and the BCN are 83 and 25, respectively. A packet in the BCN will go through an inter-board link and an inter-module link at most one and three times, respectively, along a shortest path. As a result, the product of hardware cost and network delay for the hypercube will be approximately 11 times larger than that of the BCN.

Note that the comparison given in [14] uses 5-cube and 3-cube as basic modules to build 64K-node hypernet and hypercube, respectively, and the comparison in [11, 12] uses 5-cube and 2-cube as basic modules to construct 1024K-node RCC and hypercube, respectively. If we use assumptions identical to those used in [11, 12, 14] (that is, the chip area limit allows a maximum of 32 processors per chip), we can use chips each with a 32-node buslet to build an 32K-node 2-deep incomplete recursive CN, incomplete CN(2,BCN(2,32)), which has smaller hardware cost and network delay than the previous incomplete CN(2,BCN(3,8)). The hardware cost of the 32K-node hypercube will be larger than the resultant incomplete CN(2,BCN(2,32)) by a factor of approximately 6.7, the network delay for the incomplete CN(2,BCN(2,32)) will be reduced to 15, and thus the product of hardware cost and network delay for the hypercube will be approximately 37 times larger than that of the incomplete CN(2,BCN(2,32)). The product of hardware cost and average delay for the hypercube is approximately 18 times larger. Note that the 15-dimensional hypercube cannot use a small hypercube with dimension larger than 3 within a basic VLSI module since the required number of pins per chip will exceed its limits. Note also that when the network size is increased, the differences of these parameters will be even more pronounced.

In Figs. 3,4, and 5, we compare the normalized hardware cost, network delay, and their product for several interconnection architectures using assumptions identical to those used in [11, 12]. Note that some of the Ring-BCN(l,32) and Recursive CN(l_2,BCN(l_1,32)) in these figures are constructed as their incomplete variants. Also note that all nodes belonging to the same cluster in a network are kept as physically close as possible. It is clear that BCNs outperform the torus, hypercube, folded hypercube, star graph, and CCC on this hardware cost × delay measure. Note that if we assume the all-port communication model, the performance of these hypercubes, folded hypercubes, and star graphs may be comparable to or somewhat better than that of the BCNs. However, the BCNs still compare favorably with the above networks in terms of the product of hardware cost and network performance.

6 Conclusion

In this paper, we have proposed BCNs and BQCNs as a new family of parallel interconnection architectures. BCNs and BQCNs not only combine some desirable properties of both the hypercube (e.g., a wealth of fast, elegant algorithms) and the star graph (e.g., small diameter), but also use nodes of small degree, making them less expensive to implement and easier to expand. We presented simple and efficient routing and broadcasting algorithms on BCNs. Moreover, we showed that BCNs and BQCNs are particularly efficient in view of their inter-module communication bandwidth requirements.

We also compared the hardware costs and network delays of BCNs with some competing networks. VLSI implementation and packaging constraints are quite pragmatic and do not lend themselves to a general treatment and universal conclusions. We do hope, however, that our examples and associated quantitative estimates have convinced the reader that BCNs are attractive candidates for high-performance parallel processing at reasonable cost.

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] Azevedo, M.M., N. Bagherzadeh, and S. Latifi, "Broadcasting algorithms for the star-connected cycles interconnection network," *J. Parallel Distrib. Comput.,* vol. 25, no. 2, Mar. 1995, pp. 209-222.

[3] Bakoglu, H.B., *Circuits, Interconnections, and Packaging for VLSI,* Addison-Wesley, Menlo Park, CA, 1990.

[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] 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] Cypher, R., "Theoretical aspects of VLSI pin limitations," *SIAM J. Comput.,* vol. 22, no. 2, Apr. 1993, pp. 356-378.

[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] Fiduccia, C.M. "Bused hypercubes and other pin-optimal networks," *IEEE Trans. Parallel Distrib. Sys.,* vol. 3, no. 1, Jan. 1992, pp. 14-24.

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

[10] Ghosh, J. and K. Hwang, "Mapping neural networks onto message-passing multicomputers," *J. Parallel Distrib. Comput.,* vol. 6, Apr. 1989, pp. 291-330.

[11] Hamdi, M., "Communication-efficient interconnection networks for parallel computations," Ph.D. dissertation, Dept. Electrical Eng., Univ. of Pittsburgh, PA, 1991.

[12] 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.

[13] 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.

[14] Hwang, K., *Parallel Processing for Supercomputers and Artificial Intelligence,* New York, McGraw-Hill, 1989.

[15] Kogge, P.M., "EXECUBE - a new architecture for scalable MPPs," *Proc. Int'l Conf. Parallel Processing,* vol. I, 1994, pp. 77-84.

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

[17] Latifi, S., M.M. 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.

[18] Leighton, F.T., *Introduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercubes,* Morgan-Kaufman, San Mateo, CA, 1994.

[19] Parhami B., "Periodically regular chordal ring networks for massively parallel architectures," *Proc. IEEE Symp. Frontiers of Massively Parallel Computation,* 1995, pp. 315-322.

[20] Parhami B. and C.-Y. Hung, "Robust shearsort on incomplete bypass meshes," *Proc. Int'l Parallel Processing Symp.,* 1995, pp 304-311.

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

[22] Sterling, T., P. Messina, and P. Smith, *Enabling Technologies for Peta(FL)ops Computing,* Cal Tech Report CCSF-45, http://nhse.npac.syr.edu/roadmap/petaflops/peta.html, Jul. 1994.

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

[24] Yeh, C.-H. and B. Parhami, "Cyclic Petersen networks: efficient fixed-degree interconnection networks for large-scale multicomputer systems," *Proc. Int'l Conf. Parallel and Distributed Processing: Techniques and Applications,* 1996, pp. 549-560.

[25] 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.