

A Unified Formulation of Honeycomb and Diamond Networks

Behrooz Parhami, *Fellow, IEEE*, and Ding-Ming Kwai

Abstract—Honeycomb and diamond networks have been proposed as alternatives to mesh and torus architectures for parallel processing. When wraparound links are included in honeycomb and diamond networks, the resulting structures can be viewed as having been derived via a systematic pruning scheme applied to the links of 2D and 3D tori, respectively. The removal of links, which is performed along a diagonal pruning direction, preserves the network's node-symmetry and diameter, while reducing its implementation complexity and VLSI layout area. In this paper, we prove that honeycomb and diamond networks are special subgraphs of complete 2D and 3D tori, respectively, and show this viewpoint to hold important implications for their physical layouts and routing schemes. Because pruning reduces the node degree without increasing the network diameter, the pruned networks have an advantage when the degree-diameter product is used as a figure of merit. Additionally, if the reduced node degree is used as an opportunity to increase the link bandwidths to equalize the costs of pruned and unpruned networks, a gain in communication performance may result.

Index Terms—Cayley graph, k -ary n -cube, network topology, processor array, pruned torus network, VLSI layout.

1 INTRODUCTION

A distributed-memory parallel architecture is characterized by a graph, with vertices representing processing nodes and edges corresponding to communication links. The chosen graph for a particular parallel machine, known as its interconnection network (topology), is a determining performance factor [15]. Interconnection networks are notoriously hard to compare in abstract terms [10]; the relative merits of various networks change with hardware implementation technology, data routing scheme, computational workload, data/task distribution, and many other architectural, system, or application parameters. Researchers in parallel processing are thus motivated to propose new or improved interconnection networks, arguing the benefits and offering performance evaluations in different contexts.

Some interconnection network topologies borrow from nature. Mesh, honeycomb, and diamond networks, for instance, bear resemblance to atomic or molecular lattice structures. Mesh networks are well-known and extensively studied [9], [15]. The term "mesh" commonly refers to a gridlike topology in which each node is connected to its four nearest neighbors (four-neighbor mesh), although eight-neighbor and hex or six-neighbor meshes are also of some interest [15]. A honeycomb network (Fig. 1a, without the wraparound links) is formed by tiling the plane with regular hexagons and placing a degree-3 node at each vertex in the natural way [11], [19], [20]. A diamond network [3], [12] is a 3D lattice structure (Fig. 3, without the

wraparound links). In the diamond network, connectivity of a node (x, y, z) along dimensions X and Y depends on whether $x + y + z$ is even (connections exist in positive direction) or odd (connections are in negative direction). Connections along the Z dimension are as in a 3D mesh.

Interest in honeycomb, diamond, and similar networks stems from the fact that they have smaller node degrees than mesh or torus networks of the same dimension, thus leading to lower implementation cost. If these simpler networks offer the same topological properties and communication performance as meshes or tori, then they would in fact be of considerable practical importance. In some adaptations, wraparound links have been added to eliminate the boundary effect, making the nodes regular in degree and the networks symmetric in structure. In the case of meshes, the addition of wraparound links in each dimension results in torus networks that have been quite popular in recent parallel machines [4], [13], [14]. The primary advantage of honeycomb and diamond networks, in their original forms or with wraparound links, is reduced node degree compared to 2D and 3D mesh/torus networks, respectively. The smaller node degree translates directly to lower hardware implementation cost and also implies greater cost-effectiveness under the commonly used degree-diameter-product figure of merit.

A honeycomb network can be built in different ways, depending on the number and positioning of the hexagons used in its construction. A particularly attractive variant has a rectangular exterior shape and is known as a honeycomb rectangular mesh (HReM) or torus (HReT), depending on the absence or presence of wraparound links [20]. One of our contributions in this paper is to note that HReT can be derived by systematically pruning certain links from a 2D torus. As an example, Fig. 1 depicts the honeycomb rectangular torus HReT(6, 3) and its redrawing as a pruned 6×6 torus, where dotted lines represent pruned links along the horizontal dimension. This observation, combined with

• B. Parhami is with the Department of Electrical and Computer Engineering, University of California, Santa Barbara, CA 93106-9560. E-mail: parhami@ece.ucsb.edu.

• D.-M. Kwai is with Worldwide Semiconductor Corporation, Hsinchu, Taiwan.

Manuscript received 15 Mar. 1999; revised 17 July 2000; accepted 19 July 2000.

For information on obtaining reprints of this article, please send e-mail to: tpds@computer.org, and reference IEEECS Log Number 109426.

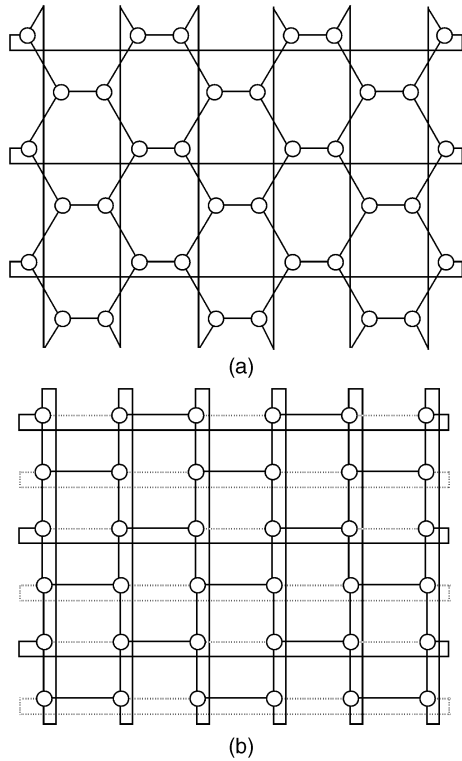


Fig. 1. Honeycomb rectangular torus (a) is isomorphic to pruned 2D torus (b). The removed links of the pruned torus are shown as dotted lines.

a proof that the resulting pruned network is a Cayley graph, and thus node-symmetric, allows us to derive an efficient routing algorithm and a compact VLSI layout for a honeycomb rectangular torus network.

Our second contribution is to note that the pruning scheme used to derive HReT networks from 2D torus can be generalized to 3D torus, leading to diamond networks [3], [12]. Again, simpler routing algorithms and efficient VLSI layouts are biproducts of this observation. Thus, honeycomb and diamond networks are intimately related; they belong to a family of graphs obtained by pruning tori of various dimensions in a certain way. Other members of this large family as yet have no specific names and have not been studied. The unified framework provided in this paper can be viewed as our third contribution. It allows for the entire family of networks mentioned above to be studied and evaluated for use in future parallel computers.

Our presentation of honeycomb and diamond networks with wraparound links in the rest of this paper is organized as follows. Section 2 contains Cayley graph constructions leading to these networks, thus establishing their node-symmetry property. Section 3 deals with the derivation of network diameter and, in the process, presents shortest-path routing algorithms for the two networks. Average internode distances for the two networks are discussed in Section 4. Section 5 is devoted to the advantages of such pruned networks, including aspects of their VLSI layout and cost-performance. Section 6 contains our conclusions.

2 SYMMETRY PROPERTIES

One way to prove that an interconnection network is node-symmetric, thereby establishing that it enjoys the advantages that come with this property, is to show that it is a Cayley graph. We thus proceed to define Cayley graphs and then prove that honeycomb and diamond networks are Cayley graphs.

Given a (nonempty) finite group Γ to be used as the node set, we identify a subset Ω of Γ that generates Γ under the group operation \otimes . The binary operator \otimes is associative but not necessarily commutative. Because we consider only graphs with no self-loop, the identity element ι does not belong to the generator set Ω . In the graph, an undirected edge connects node α to node β whenever $\beta = \alpha \otimes \omega$ for some $\omega \in \Omega$. Lack of direction on edges implies that the inverse of ω is also in the generator set Ω . By the definition above, Cayley graphs are easily seen to be node-symmetric with a node degree equal to the cardinality of Ω [18], [8].

Let us denote a node in an $l \times k$ torus as (x, y) , where $0 \leq x \leq l-1$ and $0 \leq y \leq k-1$. Consider a pruning scheme where for each node (x, y) , the connection to $(x+1, y)$ is removed if $x+y$ is odd and the connection to $(x-1, y)$ is removed if $x+y$ is even. Clearly, such a pruned $l \times k$ torus will be regular of degree 3 only when l is even. It is possible to construct a proof similar to that in [6] to show that the network defined above is a Cayley graph of cyclic groups \mathbb{Z}/l and \mathbb{Z}/k (modulo- l and modulo- k integers). Here and throughout this paper, it is understood that modulo arithmetic applies to all node-index expressions.

Theorem 1. *The honeycomb (pruned 2D torus) network is a Cayley graph.*

Proof. Consider an $l \times k$ pruned torus. Take $\Gamma = \{[x \ y]^T \mid 0 \leq x \leq l-1, 0 \leq y \leq k-1\}$ and $\iota = [0 \ 0]^T$. Define the group operator as follows:

$$\begin{bmatrix} x \\ y \end{bmatrix} \otimes \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}^{x+y} \begin{bmatrix} u \\ v \end{bmatrix} + \begin{bmatrix} x \\ y \end{bmatrix}.$$

Then, it is easily verified that the generator set

$$\Omega = \left\{ \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ -1 \end{bmatrix} \right\}$$

is closed under inverse, with $[1 \ 0]^T$ being its own inverse and the other two generators being each other's inverse. That the preceding Cayley graph construction produces an $l \times k$ honeycomb or pruned torus network is evident from the fact that the square of the 2×2 matrix A used in defining \otimes is the identity matrix I_2 , causing successive powers of A to alternate between A and I_2 . When $A^{x+y} = A$, neighbors of $[x \ y]^T$ are $[x-1 \ y]^T$, $[x \ y+1]^T$, and $[x \ y-1]^T$; when $A^{x+y} = I_2$, the neighbors are $[x+1 \ y]^T$, $[x \ y+1]^T$, and $[x \ y-1]^T$. Note that the operator \otimes defined above is associative but not commutative; associativity is easily verified and noncommutativity is deduced by noting that traversing a dimension- X link followed by a dimension- Y link does not lead to the same destination node as traversal of links in the reverse order. \square

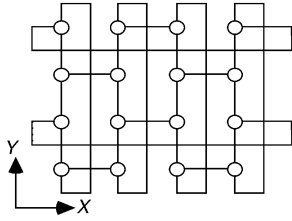


Fig. 2. A square torus pruned along the direction $x + y$ is not edge-symmetric.

The pruned torus network is node-symmetric by Theorem 1, but it is not edge symmetric, even for $l = k$. Fig. 2 shows an example with $l = k = 4$, where lack of edge-symmetry is obvious in view of the fact that a dimension- X link does not belong to any cycle of length four, whereas dimension- Y links do form four-cycles.

In an $m \times l \times k$ torus, with both m and l even, alternately removing the dimension- X and dimension- Y links along $x + y + z$ leads to a pruned $m \times l \times k$ torus that can similarly be proven to be a Cayley graph of the cyclic groups \mathbf{Z}/m , \mathbf{Z}/l , and \mathbf{Z}/k (see Theorem 2 for the general nD case and its proof). The resulting pruned 3D torus network is isomorphic to the diamond lattice [3], [12], with wraparound links added to make it regular of degree 4.

Fig. 3 depicts an example diamond lattice with $m = l = k = 4$. In this pruning scheme, for each node (x, y, z) , the connections to node $(x + 1, y, z)$ and $(x, y + 1, z)$ are removed if $x + y + z$ is odd and the links to nodes $(x - 1, y, z)$ and $(x, y - 1, z)$ are removed if $x + y + z$ is even. All dimension- Z links are kept intact. The node degree is reduced from 6 in the case of 3D torus to 4 for the pruned version. More generally, this type of pruning reduces the node degree from $2n$ to $n + 1$.

Based on the discussion above, generalizing honeycomb and diamond networks to n dimensions is straightforward. In the nD case, a $k_0 \times k_1 \times \dots \times k_{n-1}$ torus, where all dimensions except possibly for k_{n-1} are even, is pruned

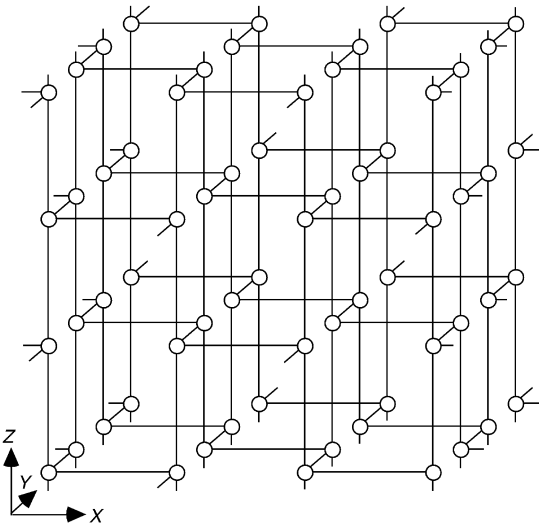


Fig. 3. A diamond lattice is isomorphic to pruned 3D torus when wraparound links (shown as short line segments in the diagram to avoid clutter) are added. The pruned 2D torus of Fig. 3 is a slice of this network cut along dimension X or Y .

along the diagonal direction $x_0 + x_1 + \dots + x_{n-1}$. Links of dimension X_{n-1} are kept intact, while half of the links belonging to the other dimensions are removed.

Theorem 2. *The nD torus network pruned along the diagonal direction $x_0 + x_1 + \dots + x_{n-1}$ is a Cayley graph.*

Proof. Consider pruning a $k_0 \times k_1 \times \dots \times k_{n-1}$ torus. Take $\Gamma = \{[a_0, a_1, \dots, a_{n-1}]^T \mid 0 \leq a_i \leq k_i - 1, 0 \leq i \leq n - 1\}$ and define $\iota = [0 \ 0 \ \dots \ 0]^T$ as the identity element. If node $\alpha = [a_0 \ a_1 \ \dots \ a_{n-1}]^T$ is connected to node $\beta = [b_0 \ b_1 \ \dots \ b_{n-1}]^T$ by a generator $\omega_i = [w_0 \ w_1 \ \dots \ w_{n-1}]^T \in \Omega$, their index vectors are related by a semidirect product

$$\begin{bmatrix} b_0 \\ b_1 \\ \dots \\ b_{n-1} \end{bmatrix} = \begin{bmatrix} -1 & 0 & \dots & 0 \\ 0 & -1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 \end{bmatrix}^{\sum a_i} \begin{bmatrix} w_0 \\ w_1 \\ \dots \\ w_{n-1} \end{bmatrix} + \begin{bmatrix} a_0 \\ a_1 \\ \dots \\ a_{n-1} \end{bmatrix}$$

and the generator set can be found to be:

$$\Omega = \left\{ \begin{bmatrix} 1 \\ 0 \\ \dots \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ \dots \\ 0 \end{bmatrix}, \dots, \begin{bmatrix} 0 \\ 0 \\ \dots \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ \dots \\ -1 \end{bmatrix} \right\}.$$

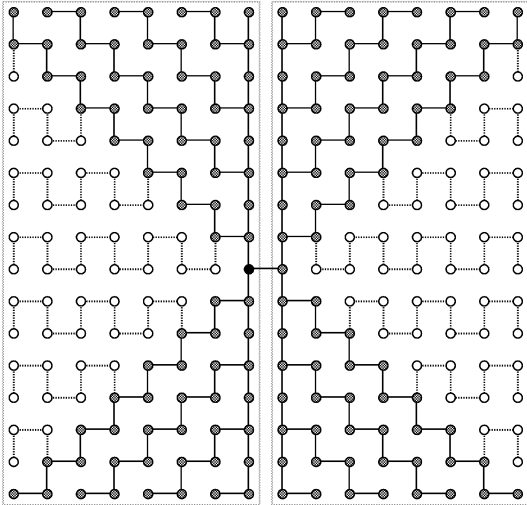
Note that the square of the $n \times n$ matrix A used in defining \otimes is the identity matrix I_n , causing successive powers of A to alternate between A and I_n as $\sum a_i = a_0 + a_1 + \dots + a_{n-1}$ assumes odd and even values. Each of the first $n - 1$ generators is its own inverse and the last two generators are each other's inverse. Because the generator set has $n + 1$ elements, the node degree of the pruned network is $n + 1$, whereas the original nD torus had node degree $2n$. \square

3 DIAMETER AND SHORTEST-PATH ROUTING

In addition to economy and ease of layout, to be discussed in Section 5, an advantage of treating the honeycomb network as a pruned 2D torus is that as in torus, we can base the routing algorithm on the offsets Δx and Δy in dimensions X and Y . The resulting algorithm is simpler than the one suggested in [20]. The latter algorithm is based on mapping the network to the Euclidean 3-space.

In the following, we consider only pruned square tori ($l = k$ or $m = l = k$), with the common side length k even. Results for other cases can be derived analogously. The diameter of the corresponding unpruned 2D or 3D torus is k or $3k/2$, respectively. In what follows, we show the diameters of the pruned versions to be the same as the unpruned networks. As part of the proof process, we obtain a simple shortest-path routing algorithm for honeycomb rectangular tori. Note that, despite a misguided tendency to dismiss the significance of network diameter, given the widespread use of wormhole switching whose latency is rather insensitive to hop distance, low network diameter is still quite important as it affects many aspects of performance when network cost or aggregate bandwidth is normalized [17].

Theorem 3. *The diameter of the pruned $k \times k$ torus is k .*


 Fig. 4. Shortest paths in the pruned 16×16 torus.

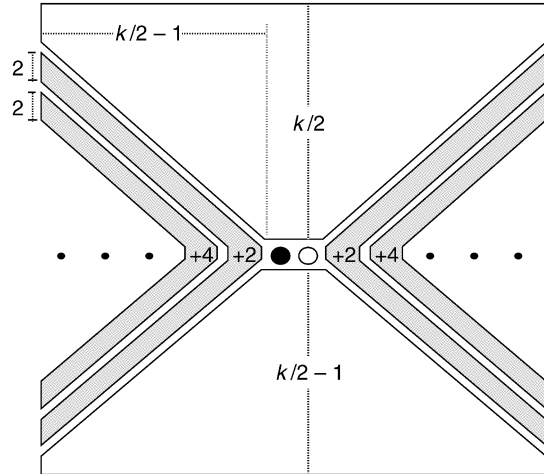
Proof. The proof is based on constructing a shortest-path routing scheme that never needs more than k hops. Given the network's node-symmetry, we can take the source node to be one of the four nodes near the center of the network's drawing (the black node in Fig. 4 or Fig. 5). The solid routing paths leading to the shaded nodes of Fig. 4 are shortest paths in the corresponding unpruned torus. The white nodes require extra hops, but never a routing distance greater than k .

To see this, let (x, y) be the source node and $(\Delta x, \Delta y)$ be the offsets to the destination node along dimensions X and Y , respectively, where $-k/2 + 1 \leq \Delta x, \Delta y \leq k/2$. Positive and negative signs in the offsets represent the directions. We always start the routing along dimension X , unless the required link has been removed; in such a case, we have to route along dimension Y so as to gain access to the dimension- X link in that particular direction.

If $|\Delta y| \geq |\Delta x|$, we encounter all required dimension- X links as we move along dimension Y ; in this case, the number $|\Delta x| + |\Delta y|$ of steps is the same as that for the unpruned torus. For $|\Delta y| \leq |\Delta x| - 1$, extra steps may have to be taken during routing. We consider two cases for $|\Delta x|$ and show that in each case, the number of steps is at most k . Without loss of generality, we assume that $x + y$ is even. If $x + y$ is odd, we simply switch the two cases.

Case 1. $\Delta x > 0$ (right half of white nodes in Fig. 4). The route from (x, y) traverses dimension X first. Since the subsequent routing along dimension Y provides access to $|\Delta y|$ of the required dimension- X links, at most $2[(\Delta x - |\Delta y|)/2]$ extra steps are needed for gaining access to the remaining $\Delta x - |\Delta y|$ dimension- X links, going back and forth along dimension Y for traversing each dimension- X link. The total number of routing steps is thus $\Delta x + |\Delta y| + 2[(\Delta x - |\Delta y|)/2] \leq 2\Delta x \leq k$.

Case 2. $\Delta x < 0$ (left half of white nodes in Fig. 4). The route from (x, y) traverses dimension Y first. Since $|\Delta y| - 1$ of the required dimension- X links become accessible as we route along dimension Y , at most $2[(-\Delta x - |\Delta y| + 1)/2]$ additional steps are needed. Thus,


 Fig. 5. Distribution of extra steps in a pruned $k \times k$ torus.

the total number of routing steps in this case is $-\Delta x + |\Delta y| + 2[(-\Delta x - |\Delta y| + 1)/2] \leq -2\Delta x + 1 \leq k - 1$. \square

Nodes that are diametrically opposite to a given node (x, y) in a pruned torus can be easily found. From the conditions $|\Delta y| \geq |\Delta x|$ and $|\Delta x| + |\Delta y| = k$, we find the node $(x + k/2, y + k/2)$, which is the only diametrically opposite node in the unpruned $k \times k$ torus. Case 1 in the proof of Theorem 3 implies that all nodes $(x + k/2, y + i)$, with i even when $k/2$ is even or i odd when $k/2$ is odd, are also diametrically opposite to node (x, y) . From the larger number of diametral paths in the pruned torus, it is intuitively obvious that the average internode distance increases as a result of pruning (see Section 4).

Routing on the pruned $k \times k \times k$ torus can mimic that of the pruned $k \times k$ torus. Let the offsets to the destination node be $(\Delta x, \Delta y, \Delta z)$, where $-k/2 + 1 \leq \Delta x, \Delta y, \Delta z \leq k/2$. We start by comparing $|\Delta x|$ to $|\Delta y|$. If $|\Delta x| \geq |\Delta y|$, we follow the routing on the pruned $k \times k$ torus in dimensions X and Z , while traversing dimension- Y links when they become accessible. Otherwise, we follow the routing on the pruned $k \times k$ torus in dimensions Y and Z , while traversing dimension- X links whenever possible. Based on the analysis in the proof of Theorem 1, the number $\max(|\Delta x|, |\Delta y|)$ of extra steps does not lead to an increase in diameter. Hence, the diameter remains $3k/2$.

4 AVERAGE INTERNODE DISTANCE

Based on the shortest-path routing algorithm described as part of the proof of Theorem 3, we can derive the average internode distance of the pruned $k \times k$ torus.

Theorem 4. *The average internode distance of the pruned $k \times k$ torus is $7k/12 - k^{-1/3}$.*

Proof. The average internode distance is obtained by summing the extra steps from a given node to all other nodes, adding the result to the sum $k^3/2$ of distances in an unpruned torus, and dividing by the number k^2 of

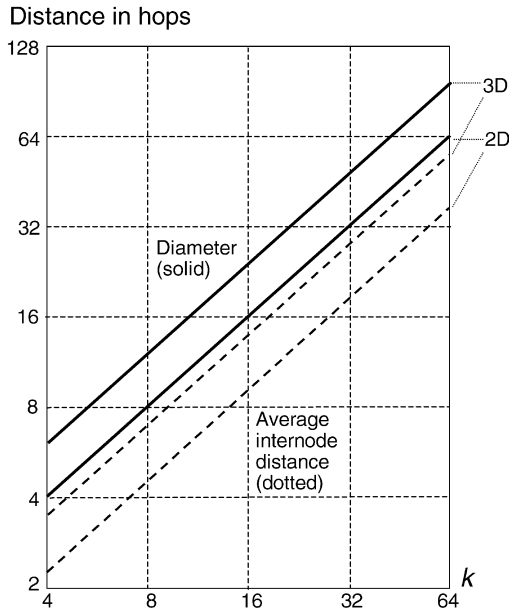


Fig. 6. Diameter and average internode distance for pruned 2D and 3D tori of side k .

nodes (we could divide by $k^2 - 1$, but we opt for a simpler expression). The distribution of extra steps is depicted in Fig. 5. Because the extra steps for the right and left halves of nodes are equal, we total the extra steps for the nodes in one half and then double the result. Recall that the number of extra steps is always an even number. If $k/2$ is even, we have one node requiring $k/2$ extra steps, eight nodes requiring $k/2 - 2$ extra steps, 16 nodes needing $k/2 - 4$ extra steps, etc. Thus, the total number of extra steps in this case can be written as:

$$E = 2 \left[k/2 + \sum_{i=1}^{k/4-1} 8i(k/2 - 2i) \right] = k^3/12 - k/3.$$

If $k/2$ is odd, we have four nodes requiring $k/2 - 1$ extra steps, 12 nodes needing $k/2 - 3$ extra steps, etc. Thus, the total number of extra steps in this case is:

$$E = 2 \left[\sum_{i=1}^{(k-2)/4} 4(2i-1)(k/2 - 2i + 1) \right] = k^3/12 - k/3.$$

In either case, the average internode distance of the pruned $k \times k$ torus is

$$(E + k^3/2)/k^2 = 7k/12 - k^{-1}/3 \approx 0.58k,$$

compared to the slightly lower average internode distance of $(k^3/2)/k^2 = 0.5k$ for the unpruned $k \times k$ torus. \square

Thus far, we have been unable to find a closed-form expression for the average internode distance of the pruned $k \times k \times k$ torus. Curve fitting on the results of numerical simulation with $4 \leq k \leq 64$, using a computer program to exhaustively enumerate all shortest paths, leads to a slope of $31k/36 \approx 0.86k$, compared to the average internode distance of $0.75k$ for the unpruned $k \times k \times k$ torus (Fig. 6).

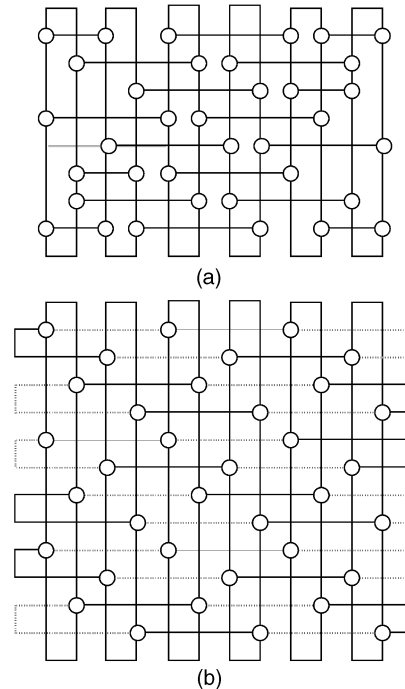


Fig. 7. Folded layout of the pruned 2D torus (a) is more compact than its unpruned counterpart (b). The removed links of the pruned torus are shown as dotted lines.

5 NETWORK ADVANTAGES

One consequence of treating the honeycomb rectangular torus as a pruned 2D torus is that its layout becomes straightforward. Any 2D torus layout can be converted to one for honeycomb torus by removing the redundant links and compacting the resulting layout to the extent possible. As is well-known for torus layouts, the long wraparound links can be avoided and wire lengths balanced by applying the standard technique of folding in both horizontal and vertical directions. When a row (column) is folded, the first half of its nodes are placed in one track and the second half in an adjacent track, in reverse order, so that the last node becomes adjacent to the first. Beginning with a folded layout for 2D torus (Fig. 7b), the space left from the removed links can be compacted to yield a smaller layout area for the desired honeycomb network (Fig. 7a).

In practice, when all else is equal, lowering of layout area potentially leads to superlinear gain in performance. A tighter layout means that larger chunks of the network can fit in a size-limited module such as a chip or a printed-circuit board; thus, there will be fewer off-chip or off-board connections, in turn leading to faster internode communication, given that on-chip links can be driven at higher speeds. Fewer chips and boards also mean lower implementation cost and power dissipation.

Smaller node degree often results in lower cost, whereas shorter diameter leads to better performance as long as the network does not operate close to saturation [16]. The pruned networks discussed in the preceding sections hold

advantages over their unpruned counterparts when the degree-diameter product is used as a figure of merit (to be minimized). This is true because the pruning scheme reduces the node degree from $2n$ to $n + 1$ without increasing the diameter. Similar advantages exist if the product of node degree and average internode distance is used for comparison.

The true test of a network's superiority is of course detailed simulation of its behavior under realistic workloads or accepted communication benchmarks. While we have not done such a study for honeycomb or diamond networks, our previous work on tori that are pruned in a different way show that performance is in fact gained. Specifically, when pruning is along y or z direction, as opposed to $x + y$ or $x + y + z$ as done here, significant performance gain can result under normalized implementation cost [7]. It remains to be seen if honeycomb and diamond networks offer comparable advantages.

6 CONCLUSION

We have presented a unified formulation of honeycomb and diamond networks with wraparound links as pruned 2D and 3D tori. Previous studies (see Section 1), focusing on several parameters such as diameter and node degree, have drawn the conclusion that these networks are attractive alternatives to complete tori. The obvious increase in routing complexity has not been dealt with.

In this paper, we have rectified what we view as misconceptions regarding routing and symmetry of these networks. It is often the case that the proof of one network topology being isomorphic to another leads to better understanding of their properties. Consolidation of algorithmic methods independently developed for the two networks is also beneficial in terms of simplification and improved efficiency. Our results serve to unify honeycomb and diamond networks with each other and with other forms of pruned tori. This unification has already simplified the layout and routing problems for such networks and may lead to other advantages, as well.

While pruned networks have received some attention from researchers in parallel processing [2], [5], [21] and a type of pruned 3D torus is used in a production parallel computer [1], more studies of the type reported in this paper appear to be warranted. Promising directions might include the development of new pruning schemes, applying known schemes to new networks, and relating various types of pruned networks to each other and to their unpruned counterparts in terms of implementation cost and communication performance.

That pruning of networks can lead to configurations with simpler 2D layouts and easier packaging, as in the two examples of this paper, is not surprising. It turns out that such pruned architectures may also outperform their

unpruned counterparts when the costs are normalized by making the communication channels of the pruned versions correspondingly wider [7]. This puts pruned networks in a unique position within the sea of interconnection networks [15]. Developers of tomorrow's massively parallel microchips and systems should consider such pruned networks as candidates for both on- and off-chip connectivity [16].

REFERENCES

- [1] R. Alverson, "The Tera Computer System," *Proc. ACM Int'l Conf. Supercomputing*, pp. 1-6, June 1990.
- [2] D.W. Bass and I.H. Sudborough, "Vertex-Symmetric Spanning Subnetworks of Hypercubes with Small Diameter," *Proc. Int'l Conf. Parallel and Distributed Computing and Systems*, pp. 7-12, Nov. 1999.
- [3] J. Gil and A. Wagner, "A New Technique for 3D Domain Decomposition on Multicomputers which Reduces Message Passing," *Proc. Int'l Parallel Processing Symp.*, pp. 831-835, Apr. 1996.
- [4] H. Ishihata, M. Takahashi, and H. Sato, "Hardware of AP3000 Scalable Parallel Server," *Fujitsu Science and Technologies J.*, vol. 33, pp. 24-30, June 1997.
- [5] O.H. Karam, "Thin Hypercubes for Parallel Computer Architectures," *Proc. Int'l Conf. Parallel and Distributed Computing and Systems*, pp. 66-71, Nov. 1999.
- [6] D.-M. Kwai and B. Parhami, "A Class of Fixed-Degree Cayley-Graph Interconnection Networks Derived by Pruning k -Ary n -Cubes" *Proc. Int'l Conf. Parallel Processing*, pp. 92-95, Aug. 1997.
- [7] D.-M. Kwai and B. Parhami, "Comparing Torus, Pruned Torus, and Manhattan Street Networks as Interconnection Architectures for Highly Parallel Computers," *Proc. Int'l Conf. Parallel and Distributed Computing and Systems*, pp. 19-22, Nov. 1999.
- [8] S. Lakshminarayanan, J.-S. Jwo, and S.K. Dahl, "Symmetry in Interconnection Networks Based on Cayley Graphs of Permutation Group: A Survey," *Parallel Computing*, vol. 19, pp. 361-401, 1993.
- [9] F.T. Leighton, *Introduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercubes*. Morgan Kaufmann, 1992.
- [10] K.J. Liszka, J.K. Antonio, and H.J. Siegel, "Problems with Comparing Interconnection Networks: Is an Alligator Better than an Armadillo?" *IEEE Concurrency*, vol. 5, no. 4, pp. 18-28, Oct.-Dec. 1997.
- [11] D. Milutinovic, V. Milutinovic, and B. Soucek, "The Honeycomb Architecture," *Computer*, vol. 20, pp. 81-83, Apr. 1987.
- [12] J. Nguyen, J. Pizaris, G. Pratt, and S. Ward, "Three-Dimensional Network Topologies," *Proc. Int'l Workshop Parallel Computer Routing and Comm.*, pp. 101-115, May 1994.
- [13] S. Oberlin, "The Cray T3D: Performance and Scalability," *Proc. ECMWF Workshop Use of Parallel Processors in Meteorology*, pp. 128-141, Nov. 1994.
- [14] E. Panizzi, "APEmille: A Parallel Processor in the Teraflopp Range," *Nuclear Physics*, vol. 53, pp. 1014-1016, Feb. 1997.
- [15] B. Parhami, *Introduction to Parallel Processing: Algorithms and Architectures*. Plenum Press, 1999.
- [16] B. Parhami and D.-M. Kwai, "Challenges in Interconnection Network Design in the Era of Multiprocessor and Massively Parallel Microchips," *Proc. Int'l Conf. Comm. Computing*, pp. 241-246, June 2000.
- [17] B. Parhami and C.-H. Yeh, "Why Network Diameter is Still Important," *Proc. Int'l Conf. Comm. Computing*, pp. 271-274, June 2000.
- [18] G. Sabidussi, "On a Class of Fixed-Point-Free Graphs," *Proc. American Mathematical Society*, vol. 9, pp. 800-804, 1958.
- [19] J.F. Sibeyn, "Routing on Triangles, Tori, and Honeycombs," *J. Foundations of Computer Science*, vol. 8, pp. 269-287, Sept. 1997.
- [20] I. Stojmenovic, "Honeycomb Networks: Topological Properties and Communication Algorithms" *IEEE Trans. Parallel and Distributed Systems*, vol. 8 pp. 1836-1842, Oct. 1997.
- [21] S.G. Ziarvas, "RH: A Versatile Family of Reduced Hypercube Interconnection Networks" *IEEE Trans. Parallel and Distributed Systems*, vol. 5, no. 11, pp. 1210-1220, Nov. 1994.



Behrooz Parhami received the PhD degree in computer science from the University of California, Los Angeles, in 1973. Presently, he is a professor in the Department of Electrical and Computer Engineering, University of California, Santa Barbara. His research deals with parallel architectures and algorithms, computer arithmetic, and reliable computing. In his previous position with Sharif University of Technology in Tehran, Iran (1974 to 1988), he was also

involved in the areas of educational planning, curriculum development, standardization efforts, technology transfer, and various editorial responsibilities, including a five-year term as editor of *Computer Report*, a Farsi-language computing periodical. His technical publications include more than 200 papers in journals and international conferences, a Farsi-language textbook, and an English/Farsi glossary of computing terms. His latest publications include two textbooks on computer arithmetic (Oxford, 2000) and parallel processing (Plenum, 1999).

Dr. Parhami is a fellow of both the IEEE and the British Computer Society, a member of the Association for Computing Machinery, and a distinguished member of the Informatics Society of Iran for which he served as a founding member and president during 1979-84. He also served as chairman of the IEEE Iran Section (1977-86) and received the IEEE Centennial Medal in 1984.



Ding-Ming Kwai received the BS and MS degrees in Taiwan from the National Cheng Kung University, Tainan, and the National Chiao Tung University, Hsinchu, in 1987 and 1989, respectively, and the PhD degree from the University of California, Santa Barbara, in 1997. He was with the Chung Cheng Institute of Technology, Taoyuan, Taiwan, as a reserve officer from 1989 to 1991 and with the Hualon Microelectronics Corporation, Hsinchu, Taiwan, as a design engineer from 1991 to 1993. He is now affiliated with the Worldwide Semiconductor Corporation, Hsinchu, Taiwan. His research interests include parallel processing, VLSI architectures, and fault-tolerant computing.