On Self-Similarity and Hamiltonicity of Dual-Cubes

Changfu Wu and Jie Wu
Department of Computer Science and Engineering
Florida Atlantic University
Boca Raton, FL 33431
{cwu02588@fau.edu, jie@cse.fau.edu}

Abstract

The dual-cube is a newly proposed topology for interconnection networks, which uses low dimensional hypercubes as building blocks. The primary advantages of the dual-cube over the hypercube are that, with the same node degree n, the dual-cube contains 2^{n-1} times more nodes than the hypercube and, with the same amount of nodes, the dual-cube has approximately 50% less links than the hypercube. This paper was focused on the investigations of the structural self-similarity and the Hamiltonian property of the dual-cube. It was shown that a dual-cube can be recursively constructed from lower dimensional dual-cubes and, conversely, a dual-cube can be recursively decomposed into lower dimensional dual-cubes. It was also proved that all dual-cubes are Hamiltonian. There exist multiple Hamiltonian cycles on a dual-cube, among which, $\lfloor (n-1)/2 \rfloor$ cycles are edge-disjoint. It was illustrated that rings and linear arrays can be effectively emulated on dual-cubes. Some strategies for track sharing in efficient VLSI layout design were also discussed.

1. Introduction

In the past two decades, a variety of interconnection networks have been suggested for parallel computing architectures, many of which are extensions or variations of the hypercube [15]. Hypercubes possess many appealing properties [6, 10, 17], such as regularity, rich connectivity, high bisection width, logarithmic node degree and diameter, ease to embed other common structures, and efficient routing. Several hypercube-based machines have been built [1, 18], from Caltech's Cosmic Cube (1983), Mark II (1984), and Intel's iPSC/1 (1985) to more recent Silicon Graphics' Origin 2000 (1996) and Origin 3000 (2000). The Origin 2000 and 3000 series servers utilize the so-called cc-NUMA (cache coherent non-uniform memory) architecture to link up multitudes of routers and nodes by emulating a multidimensional hypercube.

However, a primary disadvantage of the hypercube architecture is the cost of increasing size. The number of nodes 2^n in an n-dimensional hypercube grows rapidly as n increases, making VLSI/WSI implementation difficult for large systems. To overcome such a shortcoming, many hypercube-like architectures have been proposed. Among them, to name a few, there are incomplete hypercubes [8], reduced hypercubes [22], cube-connected cycles [13], hierarchical cube networks [5], folded hypercubes [4], unidirectional hypercubes [3], spanning bus connected hypercubes [12], augmented binary hypercubes [9], balanced hypercubes [21], Fibonacci cubes [7], extended Fibonacci cubes [20], and enhanced Fibonacci cubes [14].

Recently, Li et al [11] proposed an interconnection network called dual-cubes. An n-dimensional dual-cube is comprised of 2^n $(n-1)$-dimensional hypercubes as building blocks, also referred to as clusters. The clusters are further grouped into two classes. Each node in a cluster has only one external link to a node in another cluster of different class.

The dual-cube was found to share many properties of the hypercube and increase tremendously the total number of nodes in the systems with a limited node degree. For a fixed node degree of n, a dual-cube contains nodes as much as 2^{n-1} times that of a hypercube. Since the integrated circuit (IC) technology limits the number of links per node, the effort is very meaningful to increase the total number of nodes while keeping the node degree small. This appreciable property motivated the current study on the dual-cube as a variation and an improvement of the hypercube, which is still used by the industry in building high performance supercomputers, such as the most recent SGI Origin 3000 series.

Some basic topological properties of the dual-cube can be found in [11], such as degree, diameter, cost, average node distance, and bisection width. This paper is focused on the dual-cubes structural self-similarity, i.e., recursive construction and decomposition, and embeddability of other

0-7695-1926-1/03/$17.00 (C) 2003 IEEE

structures. Some primitive considerations on optimal VLSI layout design of the dual-cube are also presented.

The rest of the paper is organized as follows: Section 2 introduces a formal definition of the dual-cube and the notations used. Section 3 investigates topological self-similarity of the dual-cube. Section 4 studies the Hamiltonian property of the dual-cube. Section 5 discusses some considerations on efficient VLSI layout design. Finally, Section 6 concludes the paper and describes some future work.

2. Definition of the dual-cube

The most basic and perhaps the most important property that the dual-cube inherits from the hypercube is the binary coding scheme of the nodes, which is beneficial for the implementation of the communication primitives. A node in an n-dimensional ($n \geq 2$) dual-cube, abbreviated as n-dual-cube and denoted as DC_n hereafter, can be expressed as

$$0 \wedge C_n^i \wedge N_n^j$$

or

$$1 \wedge N_n^j \wedge C_n^i \qquad (1 \leq i, j \leq 2^{n-1})$$

where $\wedge$ denotes the concatenation of binary strings. For simplicity, we will omit the symbol $\wedge$ in the following context. In the above notation, the first bit, namely 0 and 1, represents class ID and C_n^i and N_n^j refer to cluster ID and node ID, respectively.

The set of nodes in DC_n can be expressed as

$$DC_n = \{0C_n^i N_n^j, 1N_n^j C_n^i, 1 \leq i, j \leq 2^{n-1}\}$$

For $n = 2$, we have $C_2^1 = 0$,$C_2^2 = 1$,$N_2^1 = 0$, and $N_2^2 = 1$. Thus,

$$DC_2 = \{000, 001, 010, 011, 100, 110, 101, 111\}$$

Two nodes in DC_n are connected if and only if they differ in exactly one bit and either of the following two conditions is satisfied: (1) The difference occurs in node ID, or (2) The difference occurs in class ID. Links satisfying the first condition are intra-cluster ones, whereas those satisfying the second condition are inter-cluster ones. The 2-dual-cube is shown in Figure 1.

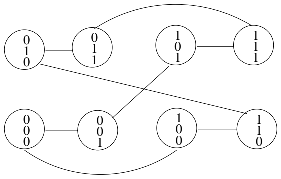

Figure 1. A 2-dual-cube, DC_2.

We can also see that the nodes connected by intra-cluster links constitute hypercubes, that is, each cluster in an n-dual-cube is an $(n-1)$-dimensional hypercube, denoted as HC_{n-1}.

3. Topological self-similarity of the dual-cube

Self-similarity of an interconnection network is desirable when deriving substructures and designing VLSI/WSI layouts. It is also the basis of some algorithms, such as divide-and-conquer. An important property of an n-dual-cube is that it can be constructed recursively from lower dimensional cubes. The set of nodes of DC_n ($n \geq 3$) can be written as

$$\begin{aligned} DC_n = \{ & 00C_{n-1}^i 0N_{n-1}^i, 00C_{n-1}^i 1N_{n-1}^i, 01C_{n-1}^i 0N_{n-1}^i, \\ & 01C_{n-1}^i 1N_{n-1}^i, 10C_{n-1}^i 0N_{n-1}^i, 10C_{n-1}^i 1N_{n-1}^i, \\ & 11C_{n-1}^i 0N_{n-1}^i, 11C_{n-1}^i 1N_{n-1}^i, 1 \leq i \leq \ 2^{n-1}\} \end{aligned}$$

with $C_2^1 = 0$,$C_2^2 = 1$, $N_2^1 = 0$, and $N_2^2 = 1$.

For example, DC_3={00000, 00001, 00100, 00101, 00010, 00011, 00110, 00111, 01000, 01001, 01100, 01101, 01010, 01010, 01110, 01111, 10000, 10001, 10100, 10101, 10010, 10011, 10110, 10111, 11000, 11001, 11100, 11101, 11010, 11011, 11110, 11111}.

Separating each cluster into two parts, one with the first bit of the node ID being 0 and the other with the first bit of the node ID being 1, an n-dual-cube can be divided into four subgraphs. An example is given in Figure 2 for the case of $n = 3$, in which a subgraph is enclosed by the dotted lines.

Obviously, if we remove the links between the four subgraphs and remove the first bits of the cluster ID and the node ID, we will obtain four identical disjoint $(n-1)$-dual-cubes. This property is summarized in the following propositions:

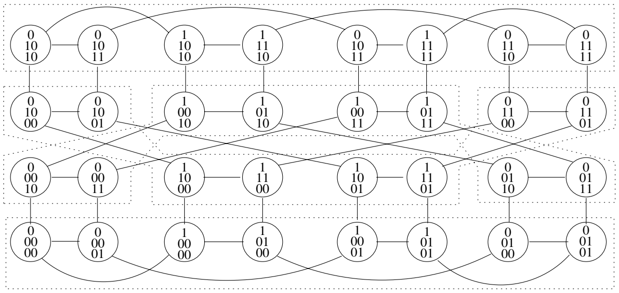

Figure 2. Separation of a DC_3 into four subgraphs.

Proposition 3.1 *Any n-dual-cube ($n \geq 3$) can be divided into four $(n-1)$-dual-cubes.*

Proof: An n-dual-cube consists of 2^{2n-1} nodes. An $(n-1)$-dual-cube consists of 2^{2n-3} nodes. Obviously, the size of an n-dual-cube is four times that of an $(n-1)$-dual-cube. Therefore, we only need to prove that each subgraph obtained by the process described above is a $(n-1)$-dual-cube. The nodes in an n-dual-cube are labeled by the set of $(2n-1)$-bit binary numbers. By removing the first bit of the cluster ID, we get two sets of $(2n-2)$-bit binary numbers. Further removing the first bit of the node ID, we will get four sets of $(2n-3)$-bit binary numbers. Furthermore, while removing the intra-cluster links between nodes with distinct first-bit node ID, all the inter-cluster links remain unchanged. The adjacency of nodes in each subgraph still satisfies the two linking conditions. Therefore, each of the subgraph is a $(n-1)$-dual-cube. □

In the above discussion, we give privilege to the first bits of the cluster ID and the node ID, i.e., we insert one bit before the first bits of the cluster ID and the node ID of DC_{n-1} to construct DC_n, and divide DC_n into four DC_{n-1}'s by removing the intra-cluster links between nodes with identical binary number in the first bit of the node ID. As a matter of fact, this is not a necessary condition. We can operate on any bit of the cluster ID or node ID, which leads to the following two propositions.

Proposition 3.2 *There are $(n-1)^2$ different ways in which we can construct an n-dual-cube from an $(n-1)$-dual-cube.*

Proof: Both the cluster ID and the node ID have $n-1$ bits. We can insert one bit in front of the ith bit $(1 \leq i \leq n-1$) of the cluster ID and the jth bit $(1 \leq i \leq r-1$) of the node ID of DC_{n-1}, as long as the same rule is followed in all nodes. Altogether, we have $(n-1) \times (n-1) = (n-1)^2$ different ways. The n-dual-cube constructed this way is unique. □

Proposition 3.3 *There are $n-1$ different ways in which we can divide an n-dual-cube into four $(n-1)$-dual-cubes.*

Proof: Each node ID has $n-1$ bits. We can remove the intra-cluster links between nodes with identical ith-bit $(1 \leq i \leq n-1)$ in the node ID. □

As an example, another way of separating DC_3 is illustrated in Figure 3.

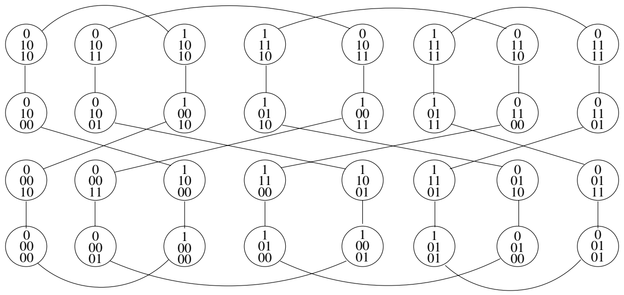

Figure 3. Separation of a DC_3 into four DC_2's.

4. Hamiltonicity of the dual-cube

The Hamiltonian property is one of the major requirements in designing the topology of networks. It enables a network to embed a ring structure, to implement the popular and new routing technique of ring routing, and to use ring-based multicast algorithms. Here we will first prove all dual-cubes are Hamiltonian and then show how to emulate rings and linear arrays on the dual-cube.

Theorem 4.1 *For any $n \geq 2$, a DC_n is a Hamiltonian graph.*

Proof: Let $\{B_1, B_2, B_3, \cdots, B_{2^k}\}$ be a sequence of k-bit Gray codes with $k = n-1$ and $B_1 = 0^{(k)}$. The superscript k in the parenthesis designates k consecutive 0's. We know that B_1 and B_{2^k} differ in only one bit according to the property of the Gray code.

A node in DC_n is represented by either $0B_iB_j$ or $1B_iB_j$. Without loss of generality, we start from node $0B_1B_1$ to construct a Hamiltonian path in DC_n. By the definition of a dual-cube, we can construct the following path:

$$\begin{array}{llll}
\{0B_1B_1, & 0B_1B_2, & \cdots, & 0B_1B_{2^k}, \\
1B_1B_{2^k}, & 1B_2B_{2^k}, & \cdots, & 1B_{2^k}B_{2^k}, \\
0B_{2^k}B_{2^k}, & 0B_{2^k}B_1, & \cdots, & 0B_{2^k}B_{2^k-1}, \\
1B_{2^k}B_{2^k-1}, & 1B_1B_{2^k-1}, & \cdots, & 1B_{2^k-1}B_{2^k-1}, \\
0B_{2^k-1}B_{2^k-1}, & 0B_{2^k-1}B_{2^k}, & \cdots, & 0B_{2^k-1}B_{2^k-2}, \\
1B_{2^k-1}B_{2^k-2}, & 1B_{2^k}B_{2^k-2}, & \cdots, & 1B_{2^k-2}B_{2^k-2}, \\
\cdots, & \cdots, & \cdots, & \cdots, \\
0B_{2^k-2}B_{2^k-2}, & 0B_{2^k-2}B_{2^k-1}, & \cdots, & 0B_{2^k-2}B_{2^k-3}, \\
1B_{2^k-2}B_{2^k-3}, & 1B_{2^k-1}B_{2^k-3}, & \cdots, & 1B_{2^k-3}B_{2^k-3}, \\
0B_2B_2, & 0B_2B_3, & \cdots, & 0B_2B_1, \\
1B_2B_1, & 1B_3B_1, & \cdots, & 1B_1B_1\}
\end{array}$$

where links within each row are intra-cluster ones and links between two rows are inter-cluster ones. It is obvious to see that the above path traverses every node in DC_n and there is a link between nodes $0B_1B_1$ and $1B_1B_1$. Thus, the path is a Hamiltonian circle. □

The Hamiltonian circle in DC_3 corresponding to the so-called binary reflected Gray code $\{00, 01, 11, 10\}$ is given as follows: $\{00000$, 00001, 00011, 00010, 10010, 10110, 11110, 11010, 01010, 01000, 01001, 01011, 11011, 10011, 10111, 11111, 01111, 01110, 01100, 01101, 11101, 11001, 10001, 10101, 00101, 00111, 00110, 00100, 10100, 11100, 11000, $10000\}$

Since there are many different ways to generate the gray codes, the Hamiltonian circles in DC_n are not unique. For instance, we can construct another Hamiltonian circle in DC_3 based on the Gray code $\{00, 10, 11, 01\}$: $\{00000$, 00010, 00011, 00001, 10001, 11001, 11101, 10101, 00101, 00100, 00110, 00111, 10111, 10011, 11011, 11111, 01111, 01101, 01100, 01110, 11110, 10110, 10010, 11010, 01010, 01011, 01001, 01000, 11000, 11100, 10100, $10000\}$.

For a large system of hundreds of nodes and links, the possibility exists for the occurrence of faulty links. Therefore, it is desirable for applications based on Hamiltonian cycles to build alternative cycles. Furthermore, if there exist edge-disjoint Hamiltonian cycles in the dual-cube, the effectiveness and fault-tolerability will undoubtably improve.

Theorem 4.2 *A dual-cube DC_n can be decomposed into $\lfloor \frac{n-1}{2} \rfloor$ edge-disjoint Hamiltonian cycles.*

Proof: We know that each cluster in a dual-cube DC_n is a hypercube HC_{n-1}. It has been shown that there exist $\lfloor \frac{n}{2} \rfloor$ edge-disjoint Hamiltonian cycles on a hypercube HC_n [16]. For example, two edge-disjoint Hamiltonian cycles on HC_4 are illustrated in Figure 4. From the proof of Theorem 5.1, we can observe that the Hamiltonian cycle on a dual-cube is actually the sequential head-tail interconnection of the Hamiltonian subcycles on hypercubic clusters. Now, the claim of theorem 2 becomes obvious. □

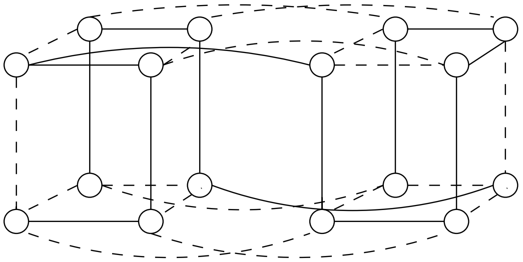

Figure 4. Edge-disjoint decomposition of a HC_4 into two Hamiltonian cycles.

Theorem 4.3 *A ring of even length $4 \leq k \leq 2^{2n-1}$ can be embedded into a DC_n, except for the case $k = 6$ and $n = 3$, with dilation, congestion, load all one and expansion $\frac{2^{2n-1}}{k}$.*

Proof: It is easy to observe that, as in a hypercube, a ring of odd length cannot be embedded into any dual-cube.

Since each cluster in a DC_n is an HC_{n-1}, a ring of even length $4 \leq k \leq 2^{n-1}$ ($n \geq 3$) can be embedded within any cluster of DC_n.

We now prove the case where the embeddings contain inter-cluster links. According to the definition of a dual-cube, there are no direct inter-cluster links between nodes of the same class and there is only one inter-cluster link between two nodes of different classes. Therefore, we need at least 4 clusters (2 clusters of class 0 and 2 clusters of class 1) to form a ring. It is then easy to verify the exceptional case.

A ring of length 2^{i+2}, $i = 1, 2, 3, \cdots, n-1$, can be embedded as follows:

For $i < n-1$,

$$\begin{array}{ll}
\{0B_1B_1, & 0B_1B_2, \\
\ 1B_1B_2, & 1B_2B_2, \\
\ \vdots & \vdots \\
\ 0B_{2^{i-1}}B_{2^{i-1}}, & 0B_{2^{i-1}}B_{2^{i-1}+1}, \\
\ 1B_{2^{i-1}}B_{2^{i-1}+1}, & 1B_{2^{i-1}+1}B_{2^{i-1}+1}, \\
\ 0B_{2^{i-1}+1}B_{2^{i-1}+1}, & 0B_{2^{i-1}+1}B_{2^{i-1}}, \\
\ 1B_{2^{i-1}+1}B_{2^{i-1}}, & 1B_{2^{i-1}+2}B_{2^{i-1}}, \\
\ \vdots & \vdots \\
\ 0B_{2^{i-2}}B_{2^{i-2}}, & 0B_{2^{i-2}}B_1, \\
\ 1B_{2^{i-2}}B_1, & 1B_1B_1\}
\end{array}$$

For $i = n-1$,

$$\begin{array}{ll}
\{0B_1B_1, & 0B_1B_2, \\
\ 1B_1B_2, & 1B_2B_2, \\
\ 0B_2B_2, & 0B_2B_3, \\
\ 1B_2B_3, & 1B_3B_3, \\
\ \vdots & \vdots \\
\ 0B_{2^i-1}B_{2^i-1}, & 0B_{2^i-1}B_{2^i}, \\
\ 1B_{2^i-1}B_{2^i}, & 1B_{2^i}B_{2^i}, \\
\ 0B_{2^i}B_{2^i}, & 0B_{2^i}B_1, \\
\ 1B_{2^i}B_1, & 1B_1B_1\}
\end{array}$$

where links within each row are intra-cluster ones and links between two rows are inter-cluster ones.

In the above construction, the ring only traverses one intra-cluster link of two adjacent nodes. From the topology of a hypercube, there exist 1 to $2^n - 1$ links between two adjacent nodes. Therefore, by replacing the intra-cluster link with more intra-cluster links up to $2^n - 1$, we can expand the above rings to contain up to 2^{n+i} links. Obviously, the intervals $[2^{i+2}, 2^{n+i}]$ ($i = 1, 2, 3, \cdots, n-1$) contain all even numbers from 8 to 2^{2n-1}.

It is further noticed that, when $n \geq 3$, the intervals $[2^{i+2}, 2^{n+i}]$ ($i = 1, 2, 3, \cdots, n-1$) intersect each other, which implies that the embedding of a ring into DC_n is not unique. □

Theorem 4.4 *A linear array of arbitrary size $1 \leq k \leq 2^{2n+1}$ can be embedded into DC_n with dilation, congestion, load all one and expansion $\frac{2^{2n-1}}{k}$.*

Proof: It is directly implied by Theorem 4.1. □

5 VLSI layout of the dual-cube

In a wafer scale integration (WSI), all the processors of a multicomputer are placed on a silicon wafer. They are arranged in a specific pattern on the wafer surface and are connected by on-wafer wiring. On-wafers wires are allowed to run only horizontally and vertically. A track is defined as a continuous horizontal or vertical line on which wires are placed without overlapping each other. If all the processors (nodes) in a multiprocessor system are arranged in an array, then all the tracks are one-dimensional. To have an efficient layout, we need to minimize the number of tracks.

Before discussing the VLSI design for the dual-cube, we will first investigate the relation between the dual-cube and the hypercube. As mentioned earlier in Section 2, a cluster in an n-dual-cube is an $(n-1)$-hypercube. Looking into the structure of a dual-cube in more detail, we can obtain the following proposition:

Proposition 5.1 *A DC_n can be embedded into an HC_{2n-1}.*

Proof: A DC_n has 2^{2n-1} nodes, each of which is represented by a $(2n-1)$-bit binary number. Two nodes are adjacent if and only if they differ in exactly one bit in class ID or node ID. On the other hand, an HC_{2n-1} has 2^{2n-1} nodes, each of which is also represented by a $(2n-1)$-bit binary number. Two nodes are adjacent if and only if they differ in one bit. Obviously, a DC_n has the same amount of nodes as an HC_{2n-1}. But the former has fewer links. Therefore, a DC_n can be embedded into an HC_{2n-1}. □

For a node degree of n, there are a total number of $2^{n-1}n$ links in the hypercube and $2^{2(n-1)}n$ links in the dual-cube. Therefore, with the same number of 2^{2n-1} nodes, the dual-cube contains $2^{2(n-1)}n$ links, while the hypercube has $2^{2(n-1)}(2n-1)$ links. The total number of l inks decreases by nearly 50%. This proposition implies that an efficient VLSI layout of a DC_n will contain much less number of tracks than that of a hypercube HC_{2n-1}.

From the proof of Theorem 4.1, we can see that, in a Hamiltonian cycle of an n-dual-cube, clusters of class 0 and class 1 are arranged consecutively in an alternating pattern, which divide the Hamiltonian cycle into 2^n sections. Each section corresponds to a Hamiltonian cycle of an $(n-1)$-hypercube. Intuitively, an efficient arrangement of both the inter-cluster links in each section and the intra-cluster links crossing sections will lead to an overall efficient VLSI layout for a dual-cube.

Chen et al. [2] proposed an efficient layout of hypercubes using one-dimensional tracks and calculated the number of one-dimensional tracks required for a given N-node hypercube, where $N = 2^n$ and n is the dimension of the hypercube. In their approach, a Hamiltonian path is constructed first, where nodes are arranged in a linear order using the binary reflected Gray code. Then, the N-node hypercube is divided two $\frac{N}{2}$-node subcubes. The addresses of two connected nodes, one from each of these two subcubes, differ in the most significant bit. In the above sequence the first eight nodes belong to one subcube and the remaining eight nodes are in the other subcube. The number of tracks required in connecting nodes among two subcubes is measured by the bisection width of this division, i.e. $\frac{N}{2}$. The above process can be repeated at each $\frac{N}{2}$-node subcube utill all the subcubes are divided into 1-node subcubes. The total number of tracks required is the sum of the bisection width at each division. It was shown that the exact number of tracks used is $N - \log N$.

Later, Wu [19] showed that, for the same layout, the number of tracks used can be reduced to $\frac{2N-1}{3}$ if n is odd or $\frac{2N-2}{3}$ if n is even. The basic idea used in Wu's approach is to share wires in bisections $2k-1$ and $2k$.

Figure 5 shows an example of wire sharing among connections in bisections 1 and 2 (where $k = 1$). Because nodes are arranged in a linear order using the binary reflected Gray code, all connections are symmetric with respect to their corresponding bisection lines. It is clear from Figure 5 that half of the connections in bisection 1 (between point A and point B in the figure) can share tracks used for connections in bisection 2. More specifically, one connection in bisection 1 shares the same track with two connections in division 2 (see Figure 5). Similar arguments can be applied to connections in bisections $2k-1$ and $2k$.

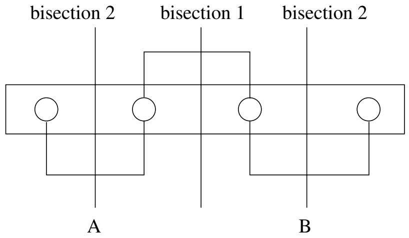

Figure 5. Wire sharing among connections in bisections 1 and 2.

Theorem 5.1 *The number of tracks (T) for connecting an array of nodes of a hypercube of size $N = 2^n$ is:*

$$T = \begin{cases} \frac{2N-1}{3} & n \text{ is odd} \\ \frac{2N-2}{3} & n \text{ is even} \end{cases}$$

Proof: Based on the above arguments, half of connections in bisection (division) $2k-1$ can share tracks used for connections in bisection $2k$; therefore, T tracks consist of

- $2^{n-1} - 2^{n-2} = 2^{n-2}$ tracks for bisection 1,
- 2^{n-2} tracks for bisection 2,
- $2^{n-3} - 2^{n-4} = 2^{n-4}$ tracks for bisection 3,
- 2^{n-4} tracks for bisection 4,
- $\vdots$
- $2^{n-(2k-1)} - 2^{n-2k} = 2^{n-2k}$ tracks for bisection $2k-1$,
- 2^{n-2k} tracks for bisection $2k$,
- $\vdots$
- 2^0 track for bisection n.

We have

$$\begin{aligned} T &= 2^{n-2} + 2^{n-2} + 2^{n-4} + 2^{n-4} + ... \\ &\quad +2^{2k} + 2^{2k} + ... \\ &= \frac{2 \times 2^n - 1 - (-1)^{n+1}}{3} \\ &= \begin{cases} \frac{2N-1}{3} & \text{n is odd} \\ \frac{2N-2}{3} & \text{n is even} \end{cases} \end{aligned}$$ □

Figures 6 and 7 show a layout of a 4-dimensional hypercube where 12 tracks are used using Chen et al.'s approach and 10 tracks are used using Wu's approach, respectively.

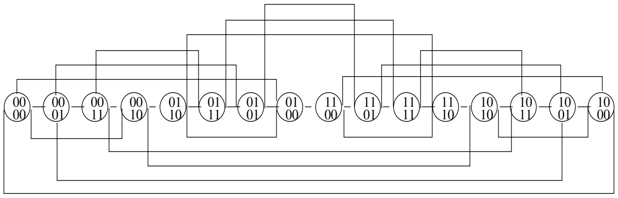

Figure 6. Linear layout of a 16-node hypercube using 12 tracks.

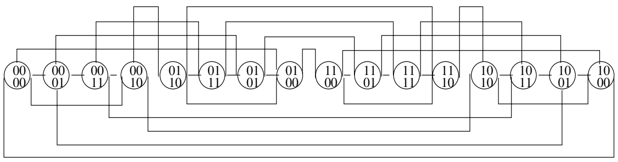

Figure 7. Linear layout of a 16-node hypercube using 10 tracks.

Wu's approach can also be used in laying out inter-cluster links in each hypercubic substructure of a dual-cube. Intra-cluster links can share tracks with each other or with inter-cluster links. Figure 8 shows a possible track sharing for the case where the end nodes of two links are adjacent to each other along the Hamiltonian path. As an example, an efficient VLSI layout for a 3-dual-cube is illustrated in Figure 9 where 9 tracks are used. On the other hand, a hypercube of the same number of nodes, 32 in this case, will need 21 tracks based on Theorem 5.1, as shown in Figure 10. It should be noted that the nodes in the Hamiltonian paths of the dual-cube and the hypercube are not in the same order.

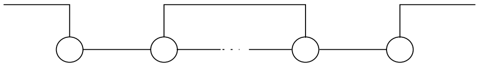

Figure 8. Track sharing among intra-cluster links and inter-cluster links.

6. Conclusions and future work

An n-dimensional dual-cube can also be viewed as an incomplete $(2n-1)$-dimensional hypercube with faulty links between nodes whose binary codes differ in only one bit and the difference occurs at positions 2 to n if the codes start with 0 and at positions $n+1$ to $2n-1$ if the codes start with 1. It was shown that the dual-cubes have recurrent structures. A higher dimensional dual-cube can be recursively constructed from or decomposed into lower dimensional dual-cubes. It was also proved that all dual-cubes are Hamiltonian. Rings and linear arrays can be embedded into a dual-cube. Compared with the hypercube of the same node degree n or of the same amount of nodes, the dual-cube has the appealing property of containing 2^{n-1} times more nodes or approximately 50% fewer links. Consequently, an efficient VLSI layout of the dual cube has significantly less number of tracks. All these desirable topological properties presented in this paper, together with many others, make the dual-cube a promising interconnection network for potential application in building large parallel computing systems.

Future work on dual-cubes includes the development of detailed strategy for designing efficient VLSI layout, the evaluation of the architectural complexity, and the investigation of some application issues in parallel computing.

References

[1] J. Ammon. Hypercube connectivity within ccnuma architecture. *URL: http://www.sgi.com/products/remarketed/origin /pdf/hypercube.pdf*, 1998.

[2] C. Chen, D. P. Agrawal, and J. R. Burke. dBCube: a new class of hierarchical multiprocessor interconnection networks with area efficient layout. *IEEE Transactions on Parallel and Distributed Systems*, 4(12):1332–1344, December 1993.

[3] C.-H. Chou and D. H. C. Du. Uni-directional hypercubes. *Proceedings of Supercomputing '90*, pages 254–263, November 1990.

[4] A. El-Amawy and S. Latifi. Properties and performance of folded hypercubes. *IEEE Transactions on Parallel and Distributed Systems*, 2(1):31–42, January 1991.

[5] K. Ghose and K. R. Desai. Hierarchical cubic networks. *IEEE Transactions on Parallel and Distributed Systems*, 6(4):427–435, April 1995.

[6] J. P. Hayes and T. Mudge. Hypercube supercomputers. *Proceedings of the IEEE*, 77(12):1829–1841, December 1989.

[7] W.-J. Hsu. Fibonacci cubes-a new interconnection technology. *IEEE Transactions on Parallel and Distributed Systems*, 4(1):3–12, January 1993.

[8] H. P. Katseff. Incomplete hypercubes. *IEEE Transactions on Computers*, 37(5):604–608, May 1988.

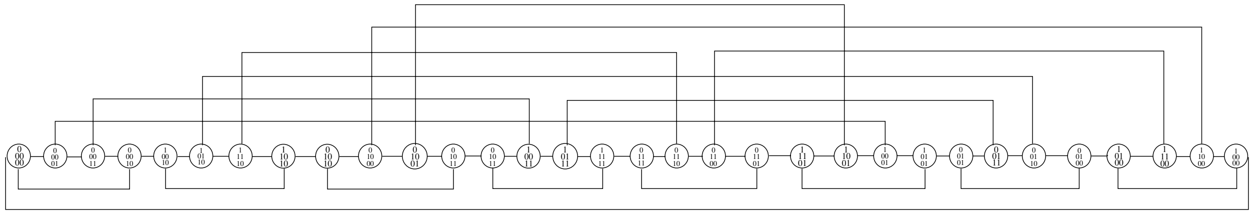

Figure 9. Linear layout of a 3-dual-cube (32 nodes) using 9 tracks.

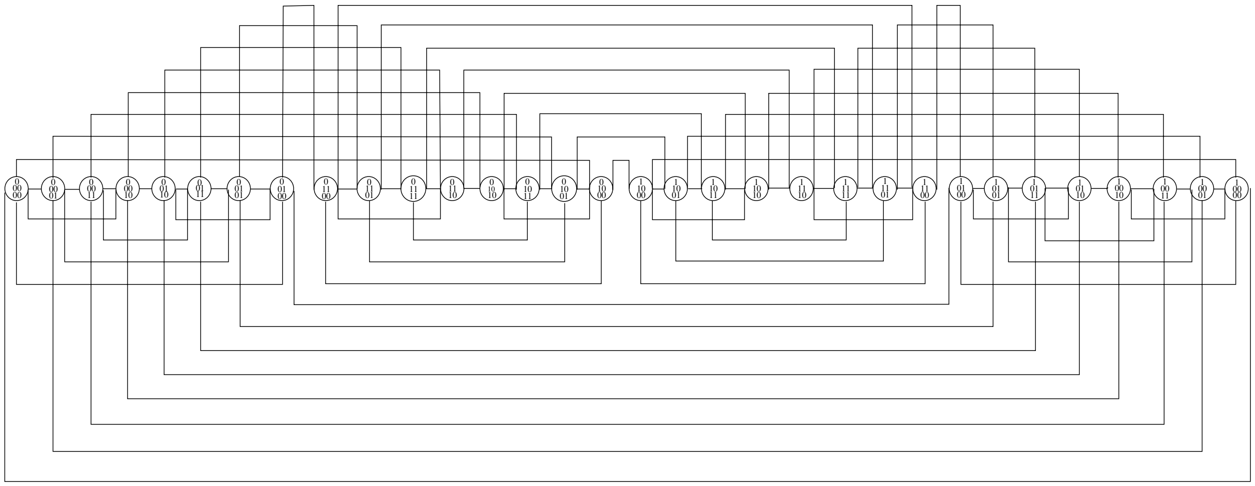

Figure 10. Linear layout of a 5-hypercube (32 nodes) using 21 tracks.

[9] H. Lalgudi, L. F. Akyildiz, and S. Yalamanchili. Augmented binary hypercube: a new architecture for processor management. *IEEE Transactions on Computers*, 45(8):980–984, July 1996.

[10] T. Leighton. *Introduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercubes*. Morgan Kaufmann Publishers, San Mateo, CA, 1992.

[11] Y. Li, S. Peng, and W. Chu. Efficient collective communications in dual-cube. *Proceedings of the 13th IASTED International Conference on Parallel and Distributed Computing and Systems (PDCS-2001)*, pages 266–271, August 2001.

[12] A. Louri and C. Neocleous. A spanning bus connected hypercube: a new scalable optical interconnection network for multiprocessors and massively parallel systems. *Journal of Lightwave Technology*, 15(7):1241–1253, July 1997.

[13] F. P. Preparata and J. E. Vuillemin. The cube-connected cycles: a versatile network for parallel computation. *Communications of the ACM*, 24(5):300–309, 1981.

[14] H. Qian and J. Wu. Unicast, multicast, and broadcast in enhanced Fibonacci cubes. *Proceedings of the Fourth International Conference on Computer Communications and Networks*, page 158161, September 1995.

[15] Y. Saad and M. H. Schultz. Topological properties of hypercubes. *IEEE Transactions on Computers*, 37(7):867–872, March 1988.

[16] S. W. Song. Towards a simple construction method for hamiltonian decomposition of the hypercube. *Discrete Mathematics and Theoretical Computer Science, DIMACS Series, American Mathematical Society*, pages 297–306, 1995.

[17] Q. F. Stout. Special issue on algorithms for hypercubes computers: Guest editor's introduction. *Journal of Parallel and Distributed Computing*, 8(4):301–302, 1990.

[18] G. V. Wilson. The history of the development of parallel computing. *URL: http://ei.cs.vt.edu/ history/Parallel.html*, 1994.

[19] J. Wu. *Comments on "dBCube: a new class of hierarchical multiprocessor networks with area efficient layout"*. Technical Report, TR-CSE-94-30, Florida Atlantic University, 1994.

[20] J. Wu. Extended Fibonacci cubes. *IEEE Transactions on Parallel and Distributed Systems*, 8(12):1203–1210, December 1997.

[21] J. Wu and K. Huang. The balanced hypercube: a cube-based system for fault-tolerant applications. *IEEE Transactions on Computers*, 46(4):484–490, April 1997.

[22] S. G. Ziavras. RH: a versatile family of reduced hypercube interconnection networks. *IEEE Transactions on Parallel and Distributed Systems*, 5(11):1210–1220, November 1994.