JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 17, 23-33 (2001)

Graph Embedding Aspect of IEH Graphs

HUNG-YI CHANG AND RONG-JAYE CHEN
Department of Computer Science and Information Engineering
National Chiao Tung University
Hsinchu, Taiwan 300, R.O.C.
E-mail: leorean@csie.nctu.edu.tw

In order to overcome the drawback of the hypercube that the number of nodes is limited to a power of two, the incrementally extensible hypercube (IEH) graph is derived for an arbitrary number of nodes [12]. In this paper, we first prove that the incomplete hypercube (IH) is a spanning subgraph of IEH. Next, we present a new method to construct an IEH from an IH. From the aspect of graph embedding, we determine the minimum size of the IEH that contains a complete binary tree. We then embed a torus (with a side length as power of two) into an IEH with dilation 1 and expansion 1.

Keywords: hypercubes, embedding, binary trees, meshes, incrementally extensible hypercubes, interconnection networks

1. INTRODUCTION

Hypercube graphs are one class among the most popular topologies for implementing massively parallel machines. It has many advantages: regularity, symmetry, low diameter, optimally fault tolerance, and so on [10]. However, the hypercube has one major drawback that it is not incrementally extensible. The number of nodes for hypercubes must be a power of two, which considerably limits the choice of the number of nodes in the graphs. To overcome this drawback, a few studies have so far tried to improve this situation but have caused new problems described briefly in the following. Bhuyan and Agrawal [2] proposed *generalized hypercubes*, which have two drawbacks: (1) the networks reduce to complete graphs when their numbers of nodes are prime, and (2) they change significantly when a new node is added. Katseff [5] proposed *incomplete hypercubes* (IHs), which suffer from the problem of fault tolerance: failure of a single node will cause the entire network to become disconnected. Sen [11] proposed *Supercubes*, which become more irregular as the size of the networks grows; for a supercube with N nodes, $2^n < N < 2^{n+1}$, the difference between the maximum and the minimum degrees of nodes can be $n - 2$. Recently, Sur and Srimani [12] have proposed a new generalization class of hypercube graphs: *incrementally extensible hypercubes* (IEHs). This topology can be defined for an arbitrary number of nodes and still reserves several advantages, such as optimal fault tolerance, low diameter, a simple routing algorithm, and near regularity.

Received March 25, 1998; revised May 11 & July 6, 1998; accepted July 27, 1998.
Communicated by Wen-Lian Hsu.

Graph embedding has been used to model the problem of simulating a parallel algorithm in a parallel machine. It is a mapping *M* of a guest graph *G* onto a host graph *H*. The cost of an embedding is measured in terms of *dilation, congestion*, and *expansion* [1, 3, 4, 6-10, 13-15]. The dilation of an embedding is the maximum distance of all edges of *G* in *H*. The congestion of an embedding is the maximum number of edges of *G* that share an edge of *H*. The expansion of an embedding is the ratio of the size of *H* to the size of *G*. Intuitively, dilation measures communication performance, congestion measures queuing delay, and expansion measures processor utilization. If *G* can be embedded into *H* with dilation 1 and expansion 1, then we say the embedding is optimal [15].

However, embedding of trees and tori into IEH graphs has never been studied. In this paper, we focus on IEH graphs and obtain the following results. First, we prove that IH(N) is a spanning subgraph of IEH(N), where N is the number of nodes. Next, we present a new method to construct an IEH from an IH. From the view point of graph embedding, we determine that the minimum size of IEH is $2^{h+1} + 1$, which contains a complete binary tree of height h as a subgraph. We then embed a torus (with a side length of 2^n) into an IEH graph with dilation 1 and expansion 1.

The rest of this paper is organized as follows. In Section 2, we introduce basic terminology for hypercubes, IHs, and IEHs. In Section 3, we show the relation between IHs and IEHs. In Sections 4 and 5, we embed binary trees and tori into IEH graphs. Finally, in Section 6, we present some conclusions.

2. PRELIMINARIES

In the research on interconnection networks, systems are modeled as graphs. In these graphs, nodes represent processors, and edges represent communication channels. A hypercube H_n is a graph $G(V, E)$, where V is the set of 2^n nodes, which are labeled as binary numbers of length n; E is the set of edges that connect two nodes if and only if they differ in exact one bit of their labels. An IH is a graph with N nodes that are labeled as binary numbers of length $\lceil \log_2 N \rceil$. Each edge joins two nodes that differ in exact one bit of their labels. An IEH graph, a generalized hypercube graph, is composed of several hypercubes of different sizes. These hypercubes are connected with *Inter-Cube* (IC) edges. Let IEH(N) be an IEH graph of N nodes. This graph is constructed by the following algorithm [12].

Algorithm 1.
Input : a positive integer N
Output : IEH(N)

1. Express N as a binary number $(c_n, \ldots, c_1, c_2)_2$, where $c_n = 1$. For each c_i, with $c_i \neq 0$, construct a hypercube H_i. The edges constructed in this step are called *regular* edges.
2. For all H_is', label each node with a dedicated binary number $11\ldots10b_{i-1}\ldots b_0$, where the length of leading 1s is $n - i$, and $b_{i-1}\ldots b_0$ is the label of this node in the regular hypercube of dimension i.
3. Find minimum i, where $c_i = 1$, set $G_j = H_i$, and set $j = i$.

$i = i + 1$.
While $i \leq n$
if $c_i \neq 0$ **then**
Connect the node $11\ldots1b_jb_{j-1}\ldots b_0$ in G_j to the following $i - j$ nodes in H_i :

$\overbrace{11...1}^{n-i}0\overbrace{11...1}^{i-j-1}b_jb_{j-1}\ldots b_0$,
$\overbrace{11...1}^{n-i}00\overbrace{01...1}^{i-j-1}b_jb_{j-1}\ldots b_0$,
$\ldots$,
$\overbrace{11...1}^{n-i}0\overbrace{11...0}^{i-j-1}b_jb_{j-1}\ldots b_0$.

Set $j = i$ and G_i be the composed graph obtained in this step. /* G_i is the graph which is composed of H_ks' for $k \leq i$.*/
endif
$i = i + 1$.
endwhile

Thus, obtain G_n as the output.#

In Algorithm 1, we observe two useful properties. First, G_i is the IEH($\sum_{k=0}^{i} c_k 2^k$) graph. Second, any two nodes that are joined by IC edges differ in one or two bits of their labels. To illustrate, Fig. 1 shows the IEH(11) graph. Note that solid lines represent regular edges, and that dotted lines represent IC edges.

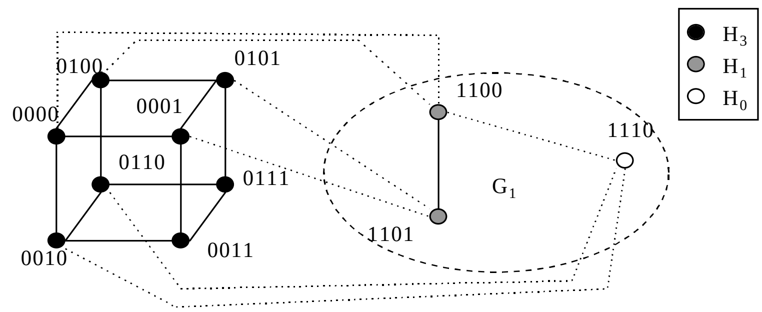

Fig. 1. IEH(11) graph.

For convenience of discussion, we divide IC edges into two classes: 1-*IC* edges and 2-*IC* edges. A 1-*IC* edge connects nodes that differ in exactly one bit of their labels; and a 2-*IC* edge connects nodes that differ in exact two bits. Let (u, v) be an IC edge, u be in H_i, and v be in H_j for $i \neq j$. We call (u, v) a *forward IC* edge of u if $i < j$; otherwise, it is called a *backward* one. Fig. 1 shows that (1100, 1110) is a forward 1-IC edge of node 1110 and (0000,1100) is a backward 2-IC edge of node 0000. Note that node u, which has forward 2-IC edges joining some nodes in H_k for $k > i$, has exactly one forward 1-IC edge to a dedicated node in H_k.

3. RELATION BETWEEN IH AND IEH

In [7], an IH was decomposed into several hypercubes of different sizes. Any pair of distinct subcubes H_k and H_j, where $k > j$, are only connected through links along dimension k. Applying this idea, we have the following algorithm, similar to Algorithm 1, to construct an IH.

Algorithm 2.
Input : a positive integer N
Output : IH(N)

1. Express N as a binary number $(c_n, \ldots, c_1, c_2)_2$, where $c_n = 1$. This vector is called *cube vector*. For each $c_i \neq 0$, construct a hypercube H_i.
2. For all H_is', label each node with a dedicated binary number $c_n \ldots c_{i+1} 0 b_{i-1} \ldots b_0$, where $b_{i-1} \ldots b_0$ is the label of this node in the regular hypercube of dimension i.
3. Find minimum i where $c_i = 1$, set $G_j = H_i$, and set $j = i$.

$i = i + 1$.
While $i \leq n$
  if $c_i \neq 0$ **then**
  Connect the node $c_n \ldots c_{j+1} b_j b_{j-1} \ldots b_0$ in G_j to the node in H_i :
    $\overbrace{c_n \ldots c_{i+1}}^{n-i} 0 \overbrace{c_{i-1} \ldots c_{j+1}}^{i-j-1} b_j b_{j-1} \ldots b_0$.
    Set $j = i$ and G_i be the composed graph obtained in this step. /* G_i is the graph which is composed of H_ks' for $k \leq i$.*/
  endif
  $i = i + 1$.
endwhile

Thus, obtain G_n as the output.#

Observe Algorithm 1 and 2. We find that they both use hypercubes of the same size as subcubes. Further, let lab(x) denote node x's label, and let (u, v) be an arbitrary edge connecting subcubes in IH(N). By relabeling IEH(N) with Step 1 and 2 of Algorithm 2, we can find a 1-IC edge (u', v') in IEH(N) such that lab(u) = lab(u') and lab(v) = lab(v'). Thus, we have the following corollary.

Corollary 1. IEH(N) contains IH(N) as a subgraph.

***Proof*:** This corollary is proved by the above argument.#

Since IHs are subgraphs of IEHs, many good results for IHs are immediately available in IEHs. For example, there is a deadlock-free routing algorithm for IHs [5]; thus, this result can be used to implement a wormhole routing algorithm for IEHs. Moreover, many parallel algorithms for IHs [3, 6, 9, 13, 15] will adapt to IEHs with slight modification.

In another topological view, we can construct IEH(N) from IH(2^n-1), where $2^{n-1} \leq N \leq 2^n - 1$. Observe Algorithm 1; in each iteration, we find that by means of 2-IC edges, a node v in G_j connects nodes in H_i that are different in two bits from v; one is the ith bit, and the other is the kth bit, where $j < k < i$. Thus, IEH(N) graphs can be obtained as follows. First, construct IH(2^n-1). Second, let $N = (c_n, \ldots, c_1, c_2)_2$, where $c_n = 1$. Consider each node u in H_l, where $c_l = 0$, and its backward IC edge from $H_{k'}$ for $k' < l$ and $c_{k'} = 1$. Connect u's backward IC-edge to its forward IC-edge with respect to H_k, where k is the minimum integer for $c_k = 1$ and $n \geq k > l$. Third, delete u but keep the edges constructed in the second step left. For example, Fig. 2 shows how to construct IEH(9) from IH(15). In this figure, gray cycles represent exist nodes, and dashed lines represent IC edges in IEH(9). Note that two forward 2-IC edges, (1110, 0100) and (1110, 0010), are composed of paths as 1110-1100-0100 and 1110-1010-0010 in IH(15), respectively.

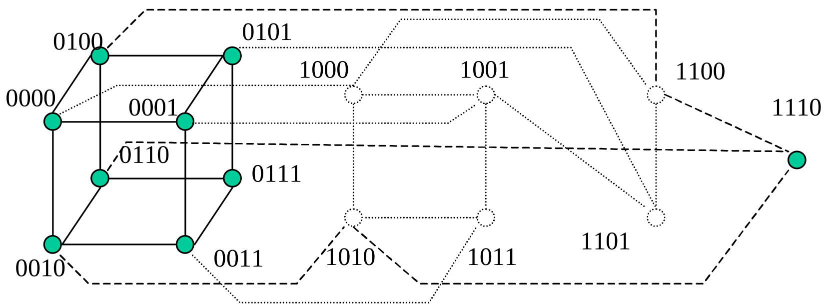

Fig. 2. Construct IEH(9) from IH(15).

4. EMBEDDING COMPLETE BINARY TREES INTO IEHS

In this section, we will show how to optimally embed complete binary trees in IEHs. We will now give some necessary definitions and explain our work.

Definition 1. [8] A *double-rooted binary tree* $DRBT_d$, where d is the height of the tree, is a complete binary tree with the root replaced by a path of length two.#

Definition 2. A *twin binary tree* TBT_d, where d is the height of the tree, is a complete binary tree with the root removed and the two level-one nodes are joined.#

To illustrate, Fig. 3 (a) shows $DRBT_3$, and Fig. 2 (b) shows TBT_2. We still need the following two lemmas for ease of reference.

Lemma 1. [8] A double-rooted tree of height h can be embedded into a (h+1) -dimensional hypercube with edge adjacency reserved.#

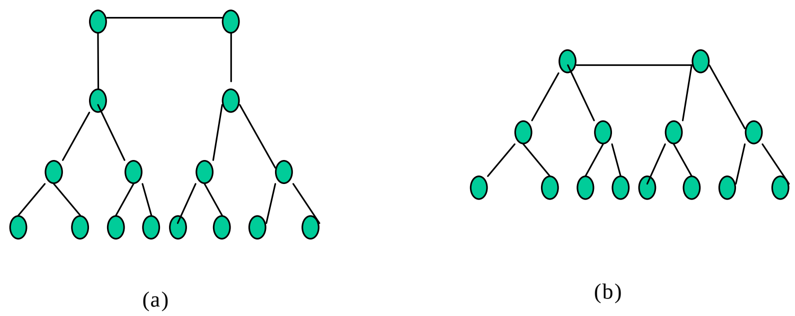

Fig. 3. $DRBT_3$ and TBT_2.

It seems that we can easily embed a TBT_2 from a $DRBT_3$ into H_4 by removing the edge of roots and joining the two nodes in the second level. However, by this method, it is impossible to embed TBT_1 from $DRBT_2$ in H_3 with edge adjacency since every node's degree is three. Thus, the following lemma is necessary.

Lemma 2. A twin binary tree of height h can be embedded into a (h+2)-dimensional hypercube with edge adjacency reserved.

Proof. It is trivial that TBT_1 can be embedded in H_3 as Fig. 4 (a) shows. Consider the embedding of TBT_2 in H_4. H_4 is divided into two H_3: one contains TBT_1 and the other contains $DRBT_2$ as Fig. 4 (b) shows. Obviously, TBT_2 can be embedded in H_4. By way of induction, we assume TBT_k can be embedded into H_{k+2} for $k > 2$. Consider the case of k+1. By Lemma 1 and the above hypothesis, we partition H_{k+3} into two H_{k+2}: one contains a $DRBT_{k+1}$ as a subgraph and the other contains a TBT_k as a subgraph as Fig. 5 shows. By adding necessary edges (i.e., the dot lines) and deleting the redundant one (i.e., the dash line), this lemma is proved.#

Observe that a complete binary tree CBT_d has $2^{d+1} - 1$ nodes. Under the condition of expansion 1, we have the following theorem for embedding a complete binary tree into an IEH with the same size.

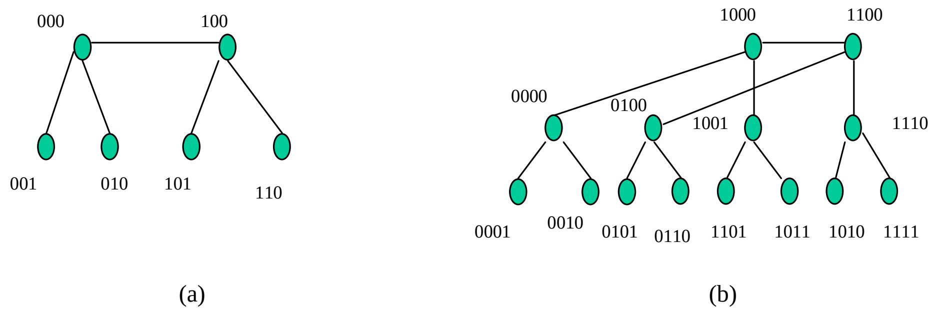

Fig. 4. Embed TBT_1 and TBT_2 into H_3 and H_4.

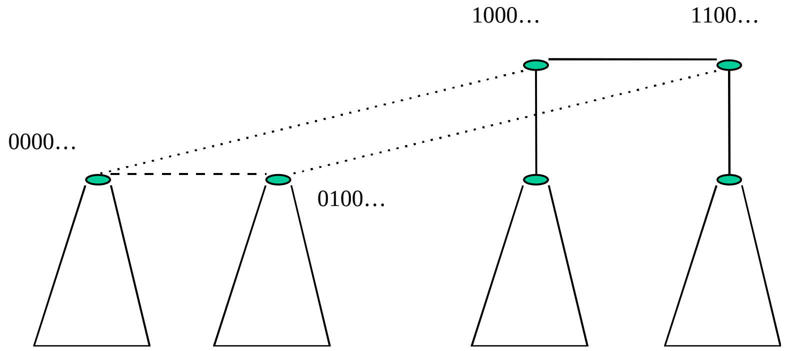

Fig. 5. Embed TBT'$_{k+1}$ into H_{k+3}.

Theorem 1. A complete binary tree CBT_d can be embedded into IEH(2^{d+1}-1) with dilation two, congestion one, and expansion one.

***Proof*.** By Corollary 1, IEH(2^{d+1} − 1) is an IH(2^{d+1} − 1) as well as H_{d+1}\(11…1). Consider the base case for d is one or two. As Fig. 6 shows, CBT_1 and CBT_2 can be embedded into IEH(3) and IEH(7) with dilation one and two, respectively. By way of induction, we assume CBT_k, where $k > 2$, can be embedded into IEH(2^{k+1}-1) with dilation two. Consider IEH(2^{k+2} − 1) is composed of H_{k+1} and IEH(2^{k+1}-1) by Algorithm 1. Further, IEH(2^{k+1}-1) is isomorphic to H_{k+1}\(011…10). Thus, by the hypothesis we can embedded CBT_{k+1} into IEH(2^{k+2} − 1) by Fglocating the root at (011…10). And the root has (11…10) and (11…10) as its sons. (For illustration, Fig. 7 shows how to embed CBT_3 into IEH(15).)#

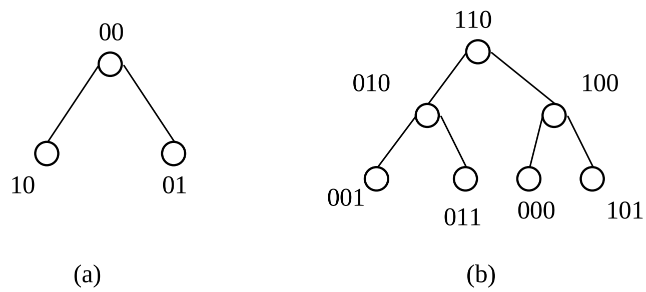

Fig. 6. Embed CBT_1 and CBT_2 into IEH(3) and IEH(7).

Under the condition of congestion 1, Tzeng et al. [13] presented an embedding of CBT_n into IH($2^n + 2^{n-1}$) with dilation 1 and expansion about 3/2. They also showed that no embedding of CBT_n into IH(2^n+2^i) with dilation 1 where $i < n - 1$. However, for IEHs, we show an optimal embedding of CBT_n into the IEH(2^{n+1} + 1) with dilation 1 and expansion 1+2/(2^{n+1} − 1). This result is superior to that of IH since we have better processor utilization in IEH.

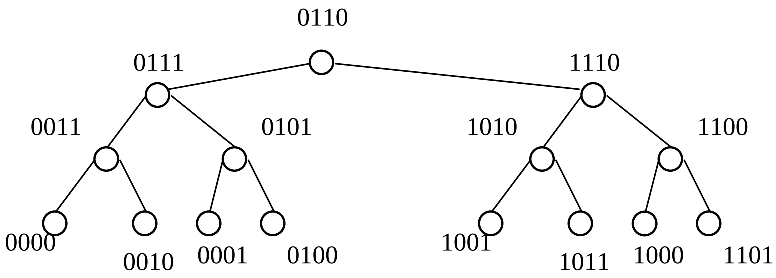

Fig. 7. Embed CBT3 into IEH(15).

Theorem 2. The minimal size of IEHs that contains a CBT_d as a subgraph is $2^{d+1} + 1$ for $d > 0$.

Proof. Since IEH($2^{d+1} - 1$) and IEH(2^{d+1}) are IH($2^{d+1} - 1$) and H_{d+1}, respectively, it is impossible to embed a CBT_d into them with edge adjacency reserved [13]. Observe that IEH(2^{d+1}+1) is a composition graph of H_{d+1} and H_0. By Lemma 2, H_{d+1} contains a TBT_{d-1} as a subgraph. Since H_{d+1} is symmetric, let two roots of this tree be $0\overbrace{11\ldots1}^{d}0$ and $0\overbrace{01\ldots1}^{d}0$. Adding H_0 and IC edges, a CBT_d is obtained for H_0 (i.e., $\overbrace{11\ldots1}^{d+1}0$) as the root, and $0\overbrace{11\ldots1}^{d}0$ and $0\overbrace{01\ldots1}^{d}0$ are its sons. Hence, the proof.#

In [1], Supercubes contained complete binary trees as spanning subgraphs. However, there is a drawback for supercubes that not all supercubes of size N, where $N > 2^{d+1}$-1 contains a CBT_d as a subgraph [1]. Without this drawback, IEH(N) contains a CBT_d as a subgraph when $N \geq 2^{d+1} + 1$.

Theorem 3. IEH(N) contains CBT_d as a subgraph when $N \geq 2^{d+1} + 1$.

Proof. Consider two cases.

Case 1. $2^{d+1} < N < 2^{d+1} + 2^d$

Because IEH(N) has H_{d+1} as a subcube, we have a TBT_{d-1} in this subcube by Lemma 2. Observe that a node v not in H_{d+1} will have 2-IC edges connecting to nodes in H_{d+1}. By adding v and its forward IC edges, our claim is found to be true in this case.

Case 2. $N \geq 2^{d+1} + 2^d$

Recall that IH is a spanning subgraph of IEH. Hence, in this case, our claim is found to be true [15].#

5. EMBEDDING MESHES AND TORI INTO IEHS

Linear arrays and rings are 1*n meshes and tori, respectively. Our previous work [4] proved that IEHs are *Hamiltonian* if the size of IEH is not $2^n - 1$ for all $n \geq 2$. Next, we showed that for an IEH of size N, an arbitrary cycle of even length N_e, where $3 < N_e < N$, is found. We also found an arbitrary cycle of odd length N_o, where $2 < N_e < N$, if and only if a node of this graph has at least one forward 2-IC edge. It would be interesting to know how many numbers we can choose for construction of IEHs such that they contain not only even cycles, but also odd cycles. Surprisingly, there are very few integers for constructing IEHs containing only even cycles as we will show in the following theorem.

Theorem 4. Let $M = \{N \mid \text{IEH}(N)$ contain only even cycles, where $2^n \leq N < 2^{n+1}\}$. Then, the size of set M, denoted by $|M|$, is $n + 1$.

Proof. Consider an IEH(N) which contains no odd cycles. Thus, this graph has no 2-IC edges from the above facts. Observe the only case in which $N = \sum_{i=j}^{n} 2^i$, where $j = 0,1, ..., n$. We obtain $|M| = n + 1$. Hence, the proof. #

In [1, 6], IHs and supercubes both contained 2^k*m meshes as spanning subgraphs where $k \geq 0$ and $m \geq 1$. Since IHs are spanning subgraphs of IEHs, a corollary is obtained immediately.

Corollary 2. IEH(N) contains 2^k*m meshes as a spanning subgraph.#

However, no embedding of tori in IHs and supercubes has been studied. In the following theorem, we will show that IEH(2^k*m) contains a 2^k*m tori as a subgraph if and only if $m \neq 2^n - 1$ for all $n \geq 2$.

Theorem 5. For all integers $k \geq 0$ and $m \geq 1$, IEH(2^k*m) contains a 2^k*m torus if and only if $m \neq 2^n - 1$ for all $n \geq 2$.

Proof. It is trivial to verify this assertion when m is one or two. For $m > 2$, recall that IEH(m) is *Hamiltonian* if and only if $m \neq 2^n$-1 for all $n \geq 2$ [4]. Further, observe that IEH(2^k*m) is a product graph of a k-dimension hypercube and an IEH(m) graph. Because a 2^k*m torus is isomorphic to a product graph of a 2^k ring and an m ring and a k-dimension hypercube contains a 2^k ring, this theorem is proved.#

6. CONCLUSIONS

In this paper, we have shown that IHs are spanning subgraphs of IEHs. Next, a complete binary tree of size N can be embedded into an IEH(N+2) graph with edge adjacency reserved and expansion near 1. We can then embed a torus of size 2^k*m into an IEH with dilation 1 and expansion 1 if and only if $m \neq 2^n - 1$ for all $n \geq 2$. Our main results are summarized in Table 1. These results support the assertion that the IEH graph is a good alternative to the hypercube for constructing an interconnection network.

Table 1. Main results.

	The minimum size to contain CBT_d as a subgraph.	If a 2^k*m mesh is a spanning subgraph.	If a 2^k*m tori is a spanning subgraph.
IH	$2^{d+1}+2^d$	Yes	No
Supercube	$2^{d+1}-1$	Yes	NA(still open)
IEH	$2^{d+1}+1$	Yes	Yes, when $m \neq 2^n-1$

REFERENCES

1. V. Auletta, A. A. Rescigno, and V. Scarano, "Embedding graphs onto the supercube," *IEEE Transactions on Computer*, Vol. 44**,** No. 4, 1995, pp. 593-597.
2. L. Bhuyan and D. P. Agrawal, "Generalized hypercube and hyperbus structure for a computer network," *IEEE Transactions on Computer*, Vol. 33, No. 3, 1984, pp. 323-333.
3. C. C. Chen and R. J. Chen, "Compact embedding of binary trees into hypercubes," *Information Processing Letter*, Vol. 54, No. 2, 1995, pp. 69-72.
4. H. Y. Chang and R. J. Chen, "Embedding cycles into IEH graphs," *Information Processing Letter*, Vol. 64, No. 1, 1997, pp. 23-27.
5. H. P. Katseff, "Incomplete hypercubes," *IEEE Transactions on Computer*, Vol. 37, No. 5, 1988, pp. 604-607.
6. A. Gupta, A. Boals, and N. Sherwani, "On optimal embeddings into incomplete hypercubes," in *Proceedings of the Fifth International Parallel Processing Symposium*, 1991, pp. 416-423.
7. S. Latifi and S. Q. Zeng, "Determination of Hamiltonian cycles in cube-based networks using generalized gray codes," *Computers Electric Engineering*, Vol. 21, No. 3, 1995, pp. 189-199.
8. T. Leighton, *Introduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercubes*, Morgan Kaufmann, reading, MA, 1992, pp. 406-408.
9. S. Ohring and S. K. Das, "Incomplete hypercubes: embeddings of tree-related networks," *Journal of Parallel and Distributed Computing*, Vol. 26, 1995, pp. 36-47.
10. Y. Saad and M. H. Schultz, "Topological properties of hypercubes," *IEEE Transactions on Computer*, Vol. 37, No. 7, 1988, pp. 867-872.
11. A. Sen, "Supercube: An optimal fault tolerant network architecture," *ACTA Informatica*, Vol. 26, No. 8, 1989, pp. 741-748.
12. S. Sur and P. K. Srimani, "IEH graphs: A novel generalization of hypercube graphs," *ACTA Informatica*, Vol. 32, No. 6, 1995, pp. 597-609.
13. N. F. Tzeng and H. L. Chen, "Structure and tree embedding aspects of incomplete hypercubes," *IEEE Transaction on Computer*, Vol. 43, No. 12, 1994, pp. 1434-1439.
14. A. S. Wagner, "Embedding the complete tree in the hypercube," *Journal of Parallel and Distributed Computing*, Vol. 20, No. 2, 1994, pp. 241-247.
15. Y. M. Yeh and Y. C. Shyu, "Efficient distributed schemes for embedding binary trees

into imcomplete hypercubes," in *Proceedings of 1994 IEEE Region Ninth Annual International Conference Theme: Frontiers of Computer Technology*, Vol. 1, 1994, pp. 182-186.

Hung-Yi Chang (張弘毅) was born in Taiwan in 1970. He received his B.S. and M.S. degrees in Computer Science and Information Engineering from National Chiao-Tung University, Hsinchu, Taiwan, Taiwan, in 1992 and 1994, respectively. He is currently a Ph.D. candidate in the Department of Computer Science and Information Engineering in National Chiao-Tung University, Hsinchu, Taiwan. His research interests include interconnection networks, parallel architecture, graph theory, and reliability.

Rong-Jaye Chen (陳榮傑) was born in Taiwan in 1952. He received his B.S. degree in Mathematics from National Tsing-Hua University, Taiwan, in 1977, his Ph.D. degree in Computer Science from the University of Wisconsin-Madison in 1987. He is now a professor in the Department of Computer Science and Information Engineering at National Chiao-Tung University, Hsinchu, Taiwan. Professor Chen is a member of IEEE. His research interests include algorithm design, theory of computation, DNA computing, interconnection networks, mobile computing, network optimization, and combinatorial optimization.