# 在漸增式超立方體圖形上的嵌置研究
# Embedding Graphs onto IEH Graphs

張弘毅
Hung-Yi Chang

交通大學資訊工程系
Department of Computer Science and Information EngineeringNational Chiao Tung University, 1001 TA Hsueh Road Hsinchu, 30050, Taiwan, ROC
leorean@csie.nctu.edu.tw

陳榮傑
Rong-Jaye Chen

交通大學資訊工程系
Department of Computer Science and Information Engineering, National Chiao Tung University, 1001 TA Hsueh Road, Hsinchu, 30050, Taiwan, ROC
rjchen@csie.nctu.edu.tw

**摘要**

*爲了克服在超立方體架構上節點數必須爲二的冪次這個缺點，可以有多節點的漸增式超立方體架構被設計出來。在此論文中，我們首先證明不完全超立方體架構是漸增式超立方體架構的子圖；其次，我們確定了可以容納完全二元樹的最少節點的漸增式超立方體；我們再嵌置環框網狀結構到此架構。*

**Abstract**

*In order to overcome the drawback of the hypercube that the number of nodes is limited as power of two, the incrementally extensible hypercube (IEH) graph is derived for arbitrary number of nodes [14]. In this paper, we first prove that the incomplete hypercube (IH) is a spanning subgraph of IEH. Next, we determine the minimum size of 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.*

*關鍵字:* 超立方體架構；嵌置；二元樹；漸增式超立方體；連結網路

***Keywords*:** Hypercubes; Embedding; Binary Trees; Incrementally Extensible Hypercubes; Interconnection Networks

## 1. Introduction

*Hypercube* graphs are one class of the most popular topologies for implementing massively parallel machines. It has many advantages: regularity, symmetry, low diameter, optimally fault tolerance, and so on [12]. However, the hypercube has one major drawback that it is not incrementally extensible. The number of nodes for hypercubes must be power of two, which considerably limit the choice of the number of nodes in the graphs. To overcome this drawback, a few papers have so far been written to improve this drawback [2,7,13-14] but cause new problems as described briefly as follows. Bhuyan and Agrawal [2] proposed *Generalized Hypercubes*, which have two drawbacks: (1) the network reduces to a complete graph when the number of nodes is prime and (2) it changes significantly when a new node is added. Katseff [7] proposed *incomplete hypercubes* (IHs), which suffer from the problem of fault tolerance -- failure of a single node will cause the entire network disconnected. Sen [13] proposed *Supercubes*, which become more irregular as the size of the networks grows. Recently, Sur and Srimani [14] 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, simple routing algorithm, and almost regularity.

Many papers have studied the ability of various network topologies to execute parallel algorithms by using embedding techniques [1, 3-4, 6, 8-12, 15]. However, embedding 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($N$) for $N$ is the number of nodes, Next, we determine the minimum size of IEH that contains a complete binary tree of height $h$. We then embed a torus(with a side length as $2^n$ ) 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 IH is a spanning subgraph of IEH. In Section 4 and 5, we embed trees and tori into IEH graphs.

## 2. Preliminaries

In the research on interconnection networks, systems are often 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 connects 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 which 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 of $N$ nodes. This graph is constructed by the following algorithm [14].

**Algorithm CONSTR.**
**Input** : a positive integer $N$
**Output** : IEH($N$)

1. Express $N$ as a binary number $(c_n, \ldots, c_1, c_0)_2$ 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 $H_i$, 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 $H_i$ :
   $\overbrace{11\ldots1}^{n-i}0\overbrace{11\ldots1}^{i-j-1}b_jb_{j-1}\ldots b_0$,
   $\overbrace{11\ldots1}^{n-i}00\overbrace{1\ldots1}^{i-j-1}b_jb_{j-1}\ldots b_0$,
   $\ldots$,
   $\overbrace{11\ldots1}^{n-i}0\overbrace{11\ldots0}^{i-j-1}b_jb_{j-1}\ldots b_0$.
   Set $j = i$ and $G_i$ be the composed graph obtained in this step. /* $G_i$ is the graph which is composed of $H_k$s' for $k \leq i$.*/
   **endif**
   $i = i + 1$.
   **endwhile**

Thus obtain the IEH($N$) graph to $G_n$.#

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

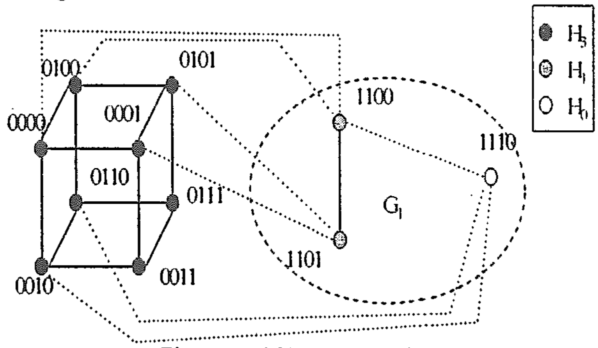


Figure 1. IEH(11) graph

For convenience of discussion, we divide IC edges into two classes: *1-IC* edges and *2-IC* edges. A 1-IC edge connects nodes which differ in exact one bit of their labels; and a 2-IC edge connects nodes which 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 a *backward* one. Figure 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, connected to some nodes in $H_k$ for $k > i$. It have exact one forward 1-IC edge to a dedicate node in $H_k$.

## 3. Relations between IH and IEH

In [9], an IH is decomposed into several hypercubes of different size. Any pair of distinct subcubes $H_k$ and $H_j$ where $k > j$ are only connected through links along dimension $k$. By applying this idea, we have the following algorithm, similar to algorithm CONSTR, to construct an IH($N$).

**Algorithm CONSTR-IH**
**Input** : a positive integer $N$
**Output** : IH($N$)

1. Express $N$ as a binary number $(c_n, \ldots, c_1, c_0)_2$ where $c_n = 1$. This vector is called *cube vector*. For each $c_i \neq 0$, construct a hypercube $H_i$.
2. For all $H_i$, label each node with a dedicated binary number $c_{n-1}\ldots c_i 0 b_{i-1}\ldots b_0$ 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** $i \le n$

**if** $c_i \ne 0$ **then**

Connect the node $c_{n-1} \ldots c_i b_j b_{j-1} \ldots b_0$ in $G_j$ to the node in $H_i$:

$$\overbrace{c_{n-1} \ldots c_{i+1}}^{n-i} 1 \overbrace{c_i \ldots c_{j+1}}^{i-j-1} b_j b_{j-1} \ldots b_0$$

Set $j = i$ and $G_i$ be the composed graph obtained in this step. /* $G_i$ is the graph which is composed of $H_k$s' for $k \le i$.*/

**endif**

$i = i + 1$.

**endwhile**

Thus obtain $G_n$, IH($N$), to output.#

Observe algorithm **CONSTR** and algorithm **CONSTR-IH**. We find that they both use the same hypercubes as subcubes. Further, these edges connecting subcubes in IH($N$) are the 1-IC edges in IEH($N$). Thus, we have the following corollary.

**Corollary 1.** IH($N$) is a subgraph of IEH($N$).

**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, IHs have a deadlock free routing[5] and this result can be used to implement wormhole routing in IEHs. Moreover, many parallel algorithms for IHs [4,9,11,15-16] will adapt to IEHs with a slight modification.

## 4. Embedding complete binary trees into IEHs

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

**Definition 1.**[10] 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 low-double-rooted binary tree $LDRBT_d$, 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.#

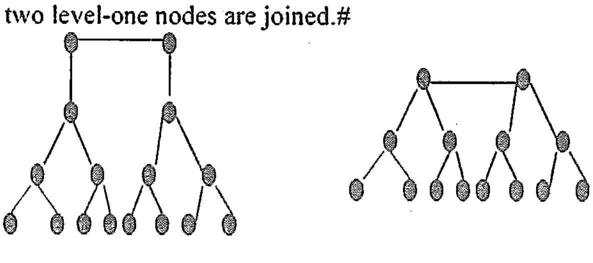


Figure 2. $DRBT_3$ and $LDRBT_2$.

For illustration, Figure 2 (a) shows $DRBT_3$ and Figure 2 (b) shows $LDRBT_2$. We still need the following lemmas for ease of reference.

**Lemma 1.** [10] A double-rooted tree of height $h$ can be embedded into an ($h$+1)-dimensional hypercube with edge adjacency reserved.#

**Lemma 2.** A low-double-rooted binary tree of height $h$ can be embedded into an ($h$+2)-dimensional hypercube with edge adjacency reserved.

**Proof.** It is trivial that $LDRBT_1$ and $LDRBT_2$ can be embedded in $H_3$ and $H_4$ as Figure 3 shows. By way of induction, we assume $DRBT_k$ can be embedded into $H_{k+2}$ for $k > 2$. Consider the case of $k$+1. By Lemma 1 and above hypothesis, we partition $H_{k+3}$ into two $H_{k+2}$: one contains a $DRBT_{k+1}$ and the other contains a $LDRBT_k$ as Figure 4 shows. By adding necessary edges ( i.e., the dot lines) and deleting nodes ( i.e., the dash line), this lemma is proved.#

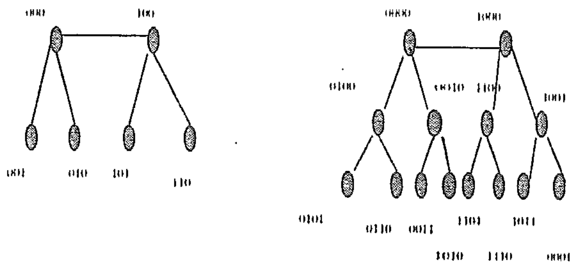

Figure 3. Embed $LDRBT_1$ and $LDRBT_2$ into $H_2$ and $H_3$.

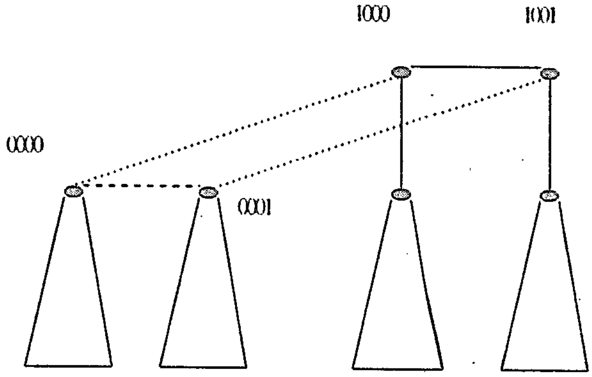


Figure 4. Embed $LDRBT'_{k+1}$ into $H_{k+3}$

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.

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

**Proof.** By Corollary 1, IEH($2^{d+1}$-1) is an IH($2^{d+1}$-1), a $V(H_{d+1}) \backslash \{11\ldots1\}$. Further, we know $H_{d+1}$ contains a $DRBT_d$ by Lemma 2. Since hypercubes are node and edge symmetric, we can assign these two roots labeled as (11...1) and (11...10). Thus there exists a path from (11...10) to the son of (11....1) without passing (11...1). Obviously, we embed a $DRBT_d$ into IEH rooted at (11...10) with dilation 2, congestion 2, and expansion 1. Hence the proof.#

Tzeng et al. [15] presented a subgraph

embedding of $CBT_n$ into $IH(2^n+2^{n-1})$ with dilation 1 and expansion 3/2. Yeh and Shyu [16] showed no embedding of $CBT_n$ into $IH(2^n+2^i)$ with dilation 1 where $i < n-1$. However, for IEHs, we show the optimal embedding of $CBT_n$, with expansion 1 and congestion 1 into the $IEH(2^{n+1}+1)$.

**Theorem 2.** The minimal size of IEH that contain $CBT_d$ is $2^{d+1}+1$ for $d > 0$.

**Proof.** 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 $LDRBT_{d-1}$. Since $H_{d+1}$ is symmetric, let two 'low' roots of this tree be $\overbrace{011\ldots10}^{d}$ and $\overbrace{001\ldots10}^{d}$. By adding $H_0$ and IC edges, A $CBT_d$ is obtained for $H_0$, $\overbrace{11\ldots10}^{d+1}$ is the root; $\overbrace{011\ldots10}^{d}$ and $\overbrace{001\ldots10}^{d}$ are its sons. Hence the proof.#

Supercubes contain complete binary trees $CBT_d$ as spanning subgraphs. However, not all supercubes of size $N$, $N > 2^{d+1}-1$, contain $CBT_d$[1]. Without this drawback, $IEH(N')$ contains $CBT_d$ 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.** $N < 2^{d+1}+2^d$

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

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

In this case, our claim is true [15-16].#

## 5. Embedding meshes and tori into IEHs

Linear arrays and rings are $1*n$ meshes and tori, respectively. Our previous work [5] proved that IEHs are *Hamiltonian* except when they are of size $2^n-1$ 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 $2 < N_o < N$ if and only if a node of this graph has at least one forward 2-*Inter-Cube* (IC) edges. In [16], IH contains $2^k*m$ meshes as subgraphs. Supercubes [1] are also proved to contain $2^k*m$ mesh. Since IH are subgraphs of IEHs, a corollary is obtained immediately.

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

However, no embedding tori work has been studied in these two topologies. In the following theorem, we show that IEH contains $2^k*m$ tori if and only if it has 2-IC edges.

**Theorem 4.** IEH(N) contains $2^k*m$ tori if and only if it has 2-IC edges with expansion 1.

**Proof.** Note that a $2^k*m$ torus is isomorphic with a product graph of $2^k$ ring and $m$ one. Further, $IEH(2^k*m)$ is a product graph of $2^k$ cube and $m$ ring if and only if it contains odd cycles. Since $2^k$ cube contains $k$ ring, this theorem is proved.#

## References.


[1] V. Auletta, A. A. Rescigno, and V. Scarano, Embedding graphs onto the supercube, *IEEE Trans. Comput.* 44 (4) (1995) 593-597.

[2] L. Bhuyan and D. P. Agrawal, Generalized hypercube and hyperbus structure for a computer network, *IEEE Trans. Comput.* 33 (3) (1984) 323-333.

[3] K. Day and A. Tripathi, Embedding of cycles in arrangement graphs, *IEEE Trans. Comput.* 42 (8) (1993) 1002-1006.

[4] C. C. Chen and R. J. Chen, Compact embedding of binary trees into hypercubes, *Inform. Process. Lett.* 54 (1995) 69-72.

[5] H. Y. Chang and R. J. Chen, Embedding Cycles into IEH Graphs, *Inform. Process. Lett.*, to appear.

[6] A. Kanevsky and C. Feng, On the embedding of cycles in pancake graphs, *Parallel Computing* 21 (1995) 923-936.

[7] H. P. Katseff, Incomplete Hypercubes, *IEEE Trans. Comput.* 37 (5) (1988) 604-607.

[8] S. Latifi and N. Bagherzadeh, On the clustered-star graph and its properties, *Computer System Sci. and Eng.* 11 (3) (1996) 145-159.

[9] S. Latifi and S. Q. Zeng, Determination of Hamiltonian Cycles in Cube-Based Networks using generalized Gray Codes, *Computers Elect. Engng.*, 21(3) (1995) 189-199.

[10] T. Leighton, Introduction to Parallel algorithms and Architectures: Arrays, Trees, Hypercubes ( Morgan Kaufmann, reading, MA, 1992) 406-408.

[11] S. Ohring and S. K. Das, "Incomplete Hypercubes: Embeddings of Tree-Related Networks," 26 (1995) 36-47.

[12] Y. Saad and M. H. Schultz, Topological properties of hypercubes, *IEEE Trans. Comput.* 37 (7) (1988) 867-872.

[13] A. Sen, Supercube: An optimal fault tolerant network architecture, *ACTA Inform.* 26 (1989) 741-748.

[14] S. Sur and P. K. Srimani, IEH graphs: A novel generalization of hypercube graphs, *ACTA Inform.* 32 (1995) 597-609.

[15] N. F. Tzeng and H. L. Chen, Structure and Tree Embedding Aspects of Incomplete Hypercubes,

*IEEE Trans. on Comput.*, 43(1994) 1434-1439.

[16] Y. M. Yeh and Y. C. Shyu, Efficient Distributed Schemes for Embedding Binary Trees into Imcomplete Hypercubes, *Proceedings of 1994 IEEE Region Ninth Annual International Conference Theme: Frontiers of Computer Technology*, 1(1994), pp. 182-186.