An Abstract of the Thesis of

Younggeun Kwon for the degree of Doctor of Philosophy in
Computer Science presented on May 4, 1993.

Title: Embeddings In Parallel Systems

Redacted for Privacy

Abstract approved: ____________________

Bella Bose

An embedding f of a graph $G = (V_g, E_g)$ into a graph $H = (V_h, E_h)$ is a one-to-one mapping from V_g to V_h. For a mapping function f, the dilation cost of an edge $(v, w) \in E_g$ is the distance between f(v) and f(w). The average dilation of the mapping f is the total sum of the dilation cost of each edge in E_g divided by the number of edges, $|E_g|$. Dilation 1 embedding means G is a subgraph of H.

By developing a mapping function from one interconnection topology to another, one can simulate the algorithms designed for the former topology on a parallel machine that uses the latter topology.

First, the embedding of butterfly-like graphs into Banyan-Hypercube networks is studied. The butterfly-like structures, considered here are the FFT, butterfly(wrap-around FFT), and the CCC (cube-connected cycle). Our embedding finds that the FFT graph, and CCC are the subgraphs of the smallest Banyan-Hypercubes which are big enough to hold them.

Further, embedding of ring structured networks on Banyan-Hypercubes network is studied. The ring structures, considered here are the regular rings, X-trees, Chordal rings, and Torus. In many cases, it is shown that the dilation of these embeddings is one. In the case where dilation is two, we show that the embedding is optimal in terms of the average dilation.

The topological properties of k-ary n-cube such as diameter, average distance, connectivity, recursive decomposibility, node-symmetry, and the number of node-disjoint paths between two nodes are also investigated.

In addition, the embedding of rings in the k-ary n-cube is investigated. Our embedding finds that a ring with n nodes, $k^{p-1} < n \le k^p$, can be embedded into a k-ary p-cube with dilation 1 between any two adjacent nodes if k is odd. In the case of k being even, a ring with n nodes, $k^{p-1} < n \le k^p$ can be embedded into a k-ary p-cube with dilation 2 if n is odd, and with dilation 1 if n is even.

The embedding of a Hamiltonian cycle in the presence of edge faults in the k-ary n-cube is also studied. Our embedding shows that there exists a Hamiltonian cycle in any direction in the k-ary n-cube with 2n - 2 edge faults, such that the Hamiltonian cycle includes any particular nonfaulty edge in that direction; A Hamiltonian cycle in the k-ary n-cube is said to be dominant in the dimension i, $0 \le i \le n - 1$ if the number of edges, not of dimension i in the k-ary n-cube used in the Hamiltonian cycle is less than or equal to 2 $(k^{n-1} - 1)$. It is also shown that there exists a dominant Hamiltonian cycle in the k-ary n-cube with 4n - 5 edge faults, provided that each node is incident to at least two nonfaulty links. These results are shown to be optimal in the sense that if more than this number of edge faults occur, it may not be possible to construct a Hamiltonian cycle.

The problem of allocating processors in a banyan-hypercube multiprocessor to the arriving task is also considered. We have shown that the subbanyan allocation problem is NP-complete, and that there does not exist any statically or dynamically optimal algorithm for recognizing subbanyan. We have presented an allocation algorithm that can recognize any available subbanyan.

Embeddings In Parallel Systems

By

Younggeun Kwon

A Thesis

Submitted to

Oregon State University

in partial fulfillment of the

requirements for the degree of

Doctor of Philosophy

Completed May 4 , 1993

Commencement June 1993

Approved:

Redacted for Privacy

Professor of Computer Scinece, in charge of major

Redacted for Privacy

Head of Department of Computer Science

Redacted for Privacy

Dean of Graduate School

Date thesis is presented May 4, 1993

Typed by Younggeun Kwon for Younggeun Kwon

To my parents

Acknowledgements

I wish to express my sincere gratitude to my advisor, Bella Bose, who was a constant source of ideas and encouragement throughout the research period. I would also like to thank other members in my committee: Prof. Toshi Minoura, Prof. Tim Budd, Prof. Vikram Saletore, and Prof. John Bolte for their support, and encouragement.

I am also grateful to Air Force Headquarters of Republic of Korea for its financial support and Oregon State University for giving me the opportunity to pursue my ph.D work.

I am deeply indebted to Prof. Sangbae Lee of Yonsei University and to Prof. Ducksoo Chang of ROK Air Force Academy for their sincere guide, and support.

I would also like to express my appreciation to Robert Broeg, Tallini Luca, Ramanathan Gowri , Seungjin Park, Eunbae Kong, Myungmoon Bae, Suckjun Lee, and Major Dokyung Ok for the good friendship and to Robert Rowley for many exciting ,and fruitful discussions during the years.

My biggest debt is to my wife and our children Yul and Haeyoung for putting up with the occasional pressures that come with my research and for providing the warm and loving environment that is important for such an endeavor.

Table of Contents

Chapter 1 Introduction 1

1.1 What is embedding ? 1

1.2 Why is embedding important ? 2

1.3 Why Banyan-Hypercube ? 3

1.4 Why K-ary n-cube ? 8

Chapter 2 Optimal Embedding of Butterfly-like Graphs into Banyan-Hypercube 11

2.1 Introduction 11

2.2 Preliminaries 12

2.3 Embeddings 15

2.3.1 Embedding an FFT into a Hypercube 15

2.3.2 Embedding an FFT into a Banyan-Hypercube 16

2.3.3 Embedding a CCC into a Banyan-Hypercube 25

2.3.4 Embedding a Butterfly network into a Banyan-Hypercube 27

Chapter 3 Embedding Ring-structured Graphs into Banyan-Hypercube 28

3.1 Introduction 28

3.2 Preliminaries 28

3.3 Embeddings 31

3.3.1 Embedding rings into a Banyan-Hypercube 31

3.3.2 Embedding Chordal Rings into a Banyan-Hypercube 34

3.3.3 Embedding X-Trees into a Banyan-Hypercube 42

3.3.4 Embedding a Torus into a Banyan-Hypercube 48

3.3.5 Embedding an Illiac into a Banyan-Hypercube 51

Chapter 4 Topological Properties of k-ary n-cube. 53

4.1 Introduction 53

4.2 Preliminaries 54

4.3 Topological Properties 55

Chapter 5 Embedding Rings into k-ary n-cube. 65

5.1 Introduction 65

5.2 Preliminaries 65

5.3 Ring Embeddings 67

Chapter 6 Embedding a Hamiltonian Cycle into Faulty k-ary n-cube 75

6.1 Introduction 75

6.2 Embedding a Hamiltonian cycle into faulty k-ary n-cube 75

Chapter 7 Processor Allocation In Banyan-Hypercube 88

7.1 Introduction 88

7.2 Preliminaries 89

7.3 Subbanyan Allocation Problem 90

7.4 Allocation Strategy 93

7.5 Algorithm Analysis 100

Chapter 8 Conclusion 101

8.1 Summary 101

8.2 Future Work 105

Bibliography 106

List of Figures

Figure

1.1	Rectangular Banyan	3
1.2	3-cube	3
1.3	Banyan-Hypercubes : BH(4, 3, 2)	5
1.4	5-ary 2-cube	9
2.1	3-dimensional FFT	13
2.2	3-dimensional CCC	13
2.3	3-dimensional butterfly	14
2.4	Embedding F(2) into relabeled BH(2, 3, 2)	20
2.5	Embedding F(4) into BH(3, 5, 2)	20
2.6	Embedding of 3-dimensional CCC into BH(3, 4, 2)	25
3.1	Banyan-Hypercubes: BH(4, 3, 2)	29
3.2	Embedding of rings into BH(4, 3, 2)	33
3.3	Two layout of n nodes chordal ring with w = n /2	34
3.4	Embedding Chordal ring of size 16 into BH(4, 3, 2)	37
3.5	The embedding sequence and the Hamiltonian cycle for n = 4	39
3.6	Embedding chordal ring	41
3.7	4-level X-tree	42
3.8	Embedding X-tree into BH	44
3.9	(5,5) torus and (5,5) Illiac network	48
4.1	4-ary 2-cube as a Cayley graph	59
5.1	Hamiltonian cycle in 5-ary 2-cube	68
5.2	Embedding cycles into 5-ary 2-cube	70
6.1	Combining two cycles into one	76

7.1	**No dynamically optimal algorithm exists for BH**	**92**
7.2	**The complete binary tree for the cube allocation using the buddy strategy**	**94**
7.3	**Buddy tree for the level 1 of BH(4, 3, 2)**	**97**
7.4	**Data structure for BH(3, 2, 2)**	**98**

Embeddings in Parallel Systems

Chapter 1

Introduction

1.1 What is embedding ?

A parallel computer can be represented by a graph, in which the nodes represent the processors and the edges represent the communication links among the processors. Similarily, a parallel algorithm can be represented by a graph, in which the nodes represent the processes, and the edges represent the communications among the processes[Ber 89]. To execute a parallel algorithm efficiently on a parallel computer, one attempts to allocate communicating processes to adjacent processors insofar as possible, so that the communication overhead is minimized. This problem is known as the embedding problem.

By an embedding of a guest graph G into the host graph H, we mean a mapping f : G ---> H that takes the nodes of G to the nodes of H and the edges of G to paths in H. The maximum amount that we must stretch any edge to achieve the embedding is called the dilation of the embedding. By expansion, we mean the ratio of the number of nodes in the host graph, H to the number of nodes in the guest graph, G. The congestion of an embedding is the maximum number of edges of the guest graph that are embedded using any single edge of the host graph. The load of an embedding is the maximum number of nodes of the guest graph that are embedded in any single node of the host graph[Lei 92].

1.2 Why is embedding important ?

Not surprisingly, the best embeddings are those for which the dilation, expansion, congestion, and load are all small. This is because these four measures bound the speed and efficiency with which a host graph can simulate the guest graph. If all four measures, dilation, congestion, expansion, and load are constant, then the host graph will be able to simulate the guest graph with constant slowdown[Lei 92]. Hence by developing a good mapping function from one interconnection topology to another, one can simulate the algorithms designed for the former topology on a parallel machine that uses the latter topology without much loss of efficiency.

When commercial hypercube-based parallel computers were introduced in 1985, the message passing strategy used was store-and-forward. Therefore, message transmission time was proportional to the path length. Hence it was important to obtain a low dilation mapping.

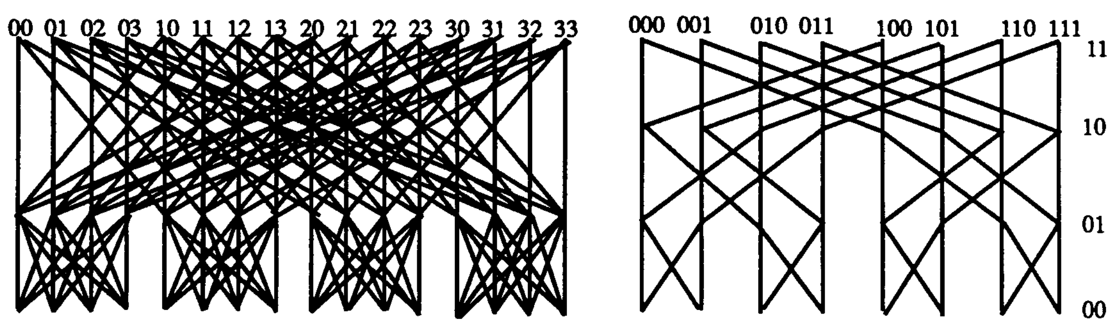
Beginning in 1987, the second generation of commercial hypercubes used more sophisticated "virtual cut-through" routing networks, such as the Intel iPSC/2's Direct-Connect routing, breaking the linear relationship between the path length and the transmission time. The iPSC/2 is able to route a message to the most distant processor in a 128-node network in only 10 % more time than it takes to reach an adjacent node[Int 87]. With the new generation of machines, the mapping problem seemed not to be important.

The problem of interconnection networks in second-generation hypercubes is that they have more communication capacity than their processors are capable of utilizing. Routing times are comparatively uniform only as long as the networks are uncongested. When a network becomes congested, delays grow with increasing path length. The experience in building hypercube based parallel machine has shown that the mapping problem becomes important whenever we try to make effective use of the communication channels in the interconnection networks of the parallel machine[Woe89].

In this thesis, several embedding algorithms for two types of interconnection network topologies - Banyan Hypercube and k-ary n-cube are investigated.

1.3 Why Banyan-Hypercube ?

A banyan is a hasse diagram of a partial ordering in which there is one and only one path from any base to any apex. A base is defined as any node with no edges incident into it, an apex is defined as any node with no edges incident out of it, and all other nodes are called intermediates [Gok78]. A L-level banyan is a banyan in which every path from any base to any apex is of length L.



(a) (b)

Figure 1.1 : Rectangular Banyan

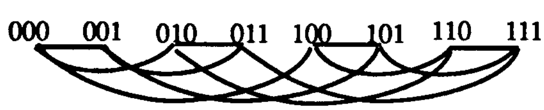


Figure 1.2 : 3-cube

A regular banyan is a L-level banyan in which the indegree of every node except the bases is s and the outdegree of every node except the apexes is f. A rectangular banyan is a regular banyan in which $s = f = d$. A (d, L) rectangular banyan has d^L nodes at each level and d^{L+1} edges from each level to the next [Pre81]. Figure 1.1 shows rectangular banyans.

In a n-dimension hypercube Q_n, each node i, $0 \leq i \leq 2^n-1$ is represented by a n-bit binary number. Two nodes are adjacent iff they differ exactly in one bit. Figure 1.2 shows a Q_3. An edge of a hypercube between two nodes that differ in the i-th bit is said to lie in the i-th dimension.

A banyan-hypercube BH(h, k, s), where $h \leq k+1$ and s is a power of two, has hs^k nodes, distributed in h levels with s^k nodes per level. All the nodes in each level are connected such that they form a hypercube with s^k nodes. Between the different levels, the nodes are connected in rectangular banyan structure. A formal definition of banyan-hypercube is given below.

Definition 1.1 :

Let BH(h, k, s) be banyan-hypercube of h levels with s^k nodes in each level. Each node is uniquely identified by a pair (L, X) of its level number L and its cube label X, where $X = x_{k-1} \dots x_1 x_0$ in base system s is adjacent to the following s nodes in level L+1 : $(L+1, x_{k-1} \dots x_{L+1}\ a\ x_{L-1} \dots x_1 x_0)$, for a = 0, 1,... s-1 if $L < h$, and to the following s nodes in level L-1: $(L-1, x_{k-1} \dots x_L\ a\ x_{L-2} \dots x_1 x_0)$, for a = 0, 1,... s-1 if $L > 0$. It is also adjacent to the k log s nodes (L, Y) in the same level where Y differs from X in exactly one bit when both X and Y are expressed in binary [You90].

Some of the desirable features of parallel machines are partitionability, the support of some common topological structures, and small diameter. The hypercube is among the networks that have been proposed and studied. It is partitionable, has a small diameter, and embeds rings, meshes, trees, butterfly-like graphs(butterfly, wrap-around butterfly, CCC), and mesh of trees efficiently.

Banyan-hypercube BH(h, k, s) can be viewed as recursive structures. BH(1, 0, s) is the smallest banyan-hypercube and consists of one node only. BH(h, k, s) can be viewed as constructed from s copies of BH(h, k - 1, s) in the following way if h < K+ 1.

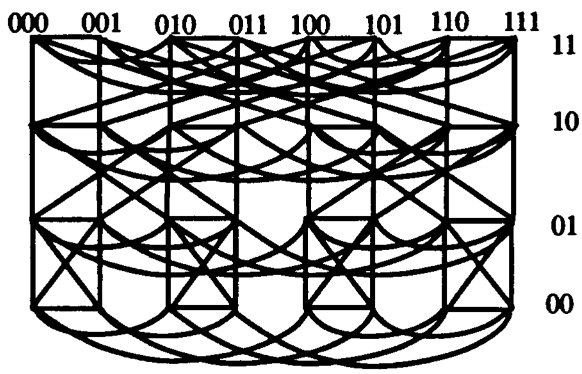


Figure 1.3 : Banyan-Hypercubes : BH(4, 3, 2)

Label these networks by 0, 1, ,...., s-1 in s-ary and transform the label of each node (L, X) in the ith network to (L, iX) (iX is the concatenation of the s-ary digit and the s-ary label X). Two nodes (L, iX) and (L, jX) in network i and j, respectively, are called siblings. These two siblings are afterwards interconnected in an s-cube, that is, any two nodes (L, iX) and (L, jX) become adjacent if i and j differ in only one bit when expressed in binary. This is done at every level.

Similarily, BH(h, k, s) can be constructed from BH(h - 1, k, s) by adding a new level (labeled h) of s^k nodes, and then making every node (h - 1, x_{k-1}, ...x_0) of level h - 1 adjacent to all the s nodes (h, x_{k-1}, ..$x_{L+1}ax_{L-1}..x_0$) of the new level a = 0, 1,...,s - 1. Therefore, BH(k + 1, k, s) can for example be constructed from BH(k, k, s) which in turn is constructed from s BH(k, k - 1, s),[You 90].

Let us compare the banyan-hypercube with the hypercube in terms of their density; a measure that relates the size of the graph to two parameters - degree and diameter. The degree of BH(h, k, s) can be easily seen to be 2s + k log s, the sum of the degree 2s of the banyan of spread s and the degree k log s of the hypercube of s^k nodes. The degree of the hypercube of hs^k nodes is log hs^k, which is equal to h + k log s. Therefore, the degree of the hypercube is asymptotically larger than that of the banyan-hypercube, assuming fixed s and increasing k and h. However, for practical values of s (2 or 4), the degree of the BH is slightly larger.

The diameter of BH(h, k, s) is k log s for s = 2, 4 [You 90]. The diameter of the hypercube of the same number of nodes hs^k is log hs^k = log h + k log s. Therefore, BH(h, k, s) has a smaller diameter than the hypercube of the same size.

	Banyan-Hypercube			Hypercube		
Net-Size	Diameter	Avg. Dist	Degree	Diameter	Avg. Dist	Degree
4	1	0.75	3	2	1.0	2
32	3	1.41	7	5	2.5	5
1024	7	4.67	11	10	5.0	10
2**19	15	10.15	19	19	9.5	19

Table 1.1: Diameter, Average distance, and Degree of Hypercubes and Banyan-Hypercubes of The Same Size.

As the diameter reflects only the worst case communication time, the average distance conveys better in practice the actual performance of the network. The average distance of BH(h, k, 2) is $k/2 + (h^2 - 1)/6h$, and that of BH(h, k, 4) is k [You 90].

The average distance of the hypercube of the same number of nodes hs^k as BH(h, k, s) is $(\log hs^k)/2 = (\log h + k \log s)/2$. Therefore, the average distance k of BH(h, k, 4) is always smaller than the average distance $(k \log s + \log h)/2$ of the hypercube of the same size. For the case s = 2, Table 1.1 presents some actual values of the diameter, average distance and degree of BH(k + 1, k, 2)'s and the hypercubes of the same number of nodes. The table clearly shows the improvement in diameter and average distance of BH's over hypercubes for practical sizes, and also shows the somewhat larger node degree of the BH.

From an embedding viewpoint, Banyan-hypercube has much to recommend it as a general purpose architecture. BH(h, k, s) embeds a ring of any size with dilation cost 1. If a $2^{i1} \times \ldots\ldots \times 2^{it}$ mesh is embedded at every level of BH(h, k, s), a new $h \times 2^{i1} \times \ldots\ldots \times 2^{it}$ mesh can be embedded in BH(h, k, s) with dilation 1. Every two consecutive levels of BH(h, k, s) embed a complete binary tree of $2s^k - 1$ nodes with dilation 1. When $(h \geq 4)$ is a power of two, BH(h, k, s) embeds a complete binary tree of $hs^k - 1$ nodes with dilation h/4. A pyramid with a $2^k \times 2^k$ base is a subgraph of BH(k + 1, k, 4)[You 90]. The regular rings, X-trees, chordal rings, torus, and butterfly like graphs such as CCC, butterfly and wrap-around butterfly can be embedded into BH(h, k, s) with dilation 1 or 2 as shown in this thesis.

1.4 Why k-ary n-cube ?

In the case of the hypercube, the Hamming distance is an appropriate parameter to explain the interconnection topology. In a similar way, the Lee distance[Pet 72], which is defined below, is an appropriate parameter to explain the k-ary n-cube topology.

Let $A = a_n\, a_{n-1} \ldots\, a_1$ be an n-bit vector with each $a_i \in \{0, 1, 2, \ldots, (k-1)\}$, where $1 \le i \le n$. The Lee weight of a vector A is defined as $W_L(A) = \Sigma \mid a_i \mid$, for $i = 0 \ldots, k-1$ where

$$\begin{aligned} \mid a_i \mid &= a_i & \text{if } 0 \le a_i \le k/2 \\ &= k - a_i & \text{if } k/2 < a_i \le k-1. \end{aligned}$$

The Lee distance between two vectors A and B, denoted by $D_L(A, B)$ is $W_L(A - B)$, the Lee weight of their difference, where - is the bitwise mod k subtraction operation. For example, when $k = 4$, the Lee weight of 321is $|4 - 3| + 2 + 1 = 4$, and the Lee distance between two vectors, 123, and 321 is $D_L(123, 321) = W_L(123 - 321) = W_L(202) = 4$.

When $k = 2$ or 3, the Lee distance between two vectors A and B is same as the Hamming distance between them. For $k > 3$, $D_L(A, B) \ge D_H(A, B)$, where $D_H(A, B)$ is the Hamming distance between two vectors A and B.

Each node in a k-ary n-cube is labelled by a distinct n-bit k-ary vector, $a_n\, a_{n-1} \ldots a_1$. Two nodes, U, and V in a k-ary n-cube are linked by an edge if and only if $W_L(U-V) = 1$, i. e, the Lee distance between them is 1. Further, the distance between any two nodes, u, and v in the k-ary n-cube is equal to $D_L(u, v)$, the Lee distance between them. Since k-ary n-cube is just the n-dimensional hypercube, when $k = 2$, here, we will assume that $k \ge 3$.

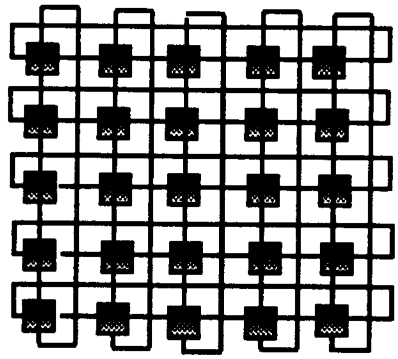


Figure 1. 4: 5-ary 2-cube

Most of the results on the embeddings related to parallel computations deal with the problem of embedding different types of graphs into hypercubes or grids, as these are the most commonly used large parallel architectures. However, the k-ary n-cube graph has been successfully used in the design of several concurrent computers including the Ametek 2010[Sei1 88], the J-machine[Dal 89], [Dal 91], and the Mosaic [Sei2 88]. It is thus of practical interest to consider embeddings of graphs into the k-ary n-cube.

A Cayley graph is a group action graph, in which the vertices correspond to the elements of the finite group G and the edges correspond to the action of the generators. That is, there is an edge from an element a to an element b if and only if there is a generator g such that ag = b is in the group[Ann 90].

The cited sources argue that the interconnection networks based on Cayley graphs endow an architecture with substantive advantages, in terms of algorithmic efficiency and fault tolerance. They support their case by noting that many interconnection networks of algorithmic and commercial importance are Cayley graphs, including the hypercube, butterfly (with wraparound), cube-connected cycles, and star networks.

It can be easily shown that the Cayley graph generated by n nodes (1, 0, 0.....0), (0, 1, 0,....0), (0, 0, 1,...., 0),......., (0, 0,....,1) with mod k bitwise plus operation is isomorphic to the k-ary n-cube. Hence k-ary n-cube is also a Cayley graph. One of the properties of Cayley graph is that it is node-symmetric[Ake 86]. Therefore, given any two vertices in the k-ary n-cube, there exists an automorphism of the graph that maps one vertex into the other (An automorphism of a graph is a one-to-one mapping of the nodes to the nodes such that edges are mapped to edges. A graph is said to be node-symmetric if for every pair of nodes, u and v, there exists an automorphism of the graph that maps a into b.).

Furthermore, k-ary n-cube is strongly hierarchical under any ordering of the set of generators. (A Cayley graph is said to be hierarchical, if its generators can be ordered as g1, g2,..., gd, such that for each i, $1 \leq i \leq d$, gi is outside the subgroup generated by the first i - 1 generators. If the Cayley graph is hierarchical under any ordering of the set of generators, we call such a Cayley graph strongly hierarchical[Ake 86].). A strongly hierarchical graph has the property that it can be recursively decomposed using the generators in any order[Ake 86]. Thus, the k-ary n-cube can be decomposed into n k-ary (n -1)-cubes along any one of its n dimensions.

Another important property of interconnection networks is their fault tolerance. The fault tolerance of a graph is better defined through the graph theoretic property, called connectivity. The connectivity of a graph is the minimum number of vertices that need to be removed to disconnect the graph. The fault tolerance is then one less than the connectivity and indicates the maximum number of vertices that can be removed and still have the graph remain connected. Clearly, any graph can be disconnected by removing all the vertices adjacent to a given vertex. Thus its connectivity can be at most its degree. It has been shown that hierarchical Cayley graphs are maximally fault tolerant[Ake 84]. That is, their fault tolerance is exactly one less than their degree. Since the degree of k-ary n-cube is 2n, k-ary n-cube is connected even in the presence of 2n -1 node faults.

Chapter 2

Optimal Embedding of Butterfly-like Graphs into Banyan-Hypercube

2.1 Introduction

One of the important features affecting the performance of parallel machines is the capability of embedding common topological structures . These structures have emerged in many applications, and include variations of binary trees, arrays, butterflys, and shuffle-exchange graphs. This occurs when the inherent underlying structure of the algorithm is a tree (divide and conquer problem) or a mesh (as is the case for many problems in numerical analysis and linear algebra)[Ber 89] or a butterfly or shuffle-exchange graph (as is the case for Fourier Transform and data manipulation problems)[Lei 92].

By developing a mapping function from one interconnection topology to another, one can simulate the algorithms designed for the former topology machine on a parallel machine that uses the latter topology. For this reason, the problem of embedding one interconnection topology into another is well studied. In particular, Youssef and Narahar [You90] proposed embeddings of Hamiltonian cycle, mesh, binary tree and pyramid topologies into the banyan-hypercube.

In this chapter, we consider further results on embedding networks with butterfly-like structures such as the FFT, butterfly(wrap-around FFT) and CCC (cube connected cycle) into the BH network.

The rest of the chapter is organized as follows. Section 2.2 introduces the necessary definitions and notations. Embedding of butterfly-like networks into the banyan hypercube will be discussed in section 2.3.

2.2 Preliminaries

In this section, we give some definitions and preliminary results useful for the subsequent sections.

In an n-dimensional hypercube Q_n, each node(vertex) i, $0 \leq i \leq 2^n-1$ is represented by a n-bit binary number. Two nodes are adjacent if and only if they differ exactly in one bit. A hypercube edge is said to lie in the ith dimension if it connects two nodes that differ in the ith bit.

We say that a graph $G = (V, E)$ is the cross product of graphs $G^1 = (V_1, E_1)$, $G^2 = (V_2, E_2)$,........, $G^k = (V_k, E_k)$ if $V = \{(v_1, v_2, \ldots\ldots v_k) \mid v_i \in V_i \text{ for } 1 \leq i \leq k\}$ and $E = \{((u_1, u_2, \ldots, u_k), (v_1, v_2, \ldots, v_k)) \mid \exists\, j \text{ such that } (u_j, v_j) \in E_j \text{ and } u_i = v_i \text{ for all } i \neq j\}$. Notationally, we represent the cross product as $G = G^1 \otimes G^2 \otimes \ldots\ldots\ldots \otimes G^k$.

Lemma 2.1[Lei92]

If $G = G^1 \otimes G^2 \otimes \ldots\ldots\ldots \otimes G^k$ and $G' = G'^1 \otimes G'^2 \otimes \ldots\ldots\ldots \otimes G'^k$

for some $k \geq 1$, and G^i is a subgraph of G'^i for $1 \leq i \leq k$. Then, G is a subgraph of G'.

The r-dimensional FFT, F(r) has $(r + 1)2^r$ nodes and $r2^{r+1}$ edges. The nodes correspond to pairs (i, w) where i is the level of the node ($0 \leq i \leq r$) and w is an r-bit binary number that denotes the row of the node. Two nodes (i, w) and (i', w') are adjacent if and only if i' = i + 1 and either :

a) w and w' are identical, or

b) w and w' differ in precisely the i'th bit.

If w and w' are identical, the edge is said to be a straight edge. Otherwise, the edge is a cross edge.

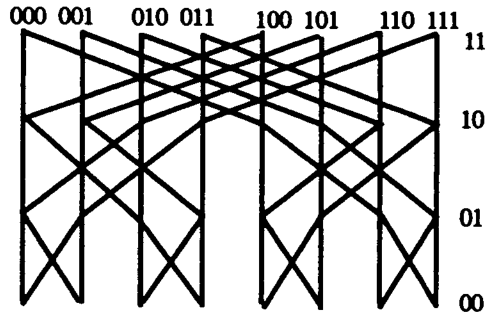


Figure 2.1: 3-dimensional FFT

Figure 2.1 shows the 3-dimensional FFT.

The r-dimensional cube-connected cycle (CCC), [Prep81] can be represented by a pair (i, w) where i ($1 \leq i \leq r$) is the position of the node within its cycle and w (r-bit binary string) is the label of the node in the hypercube that corresponds to the cycle.

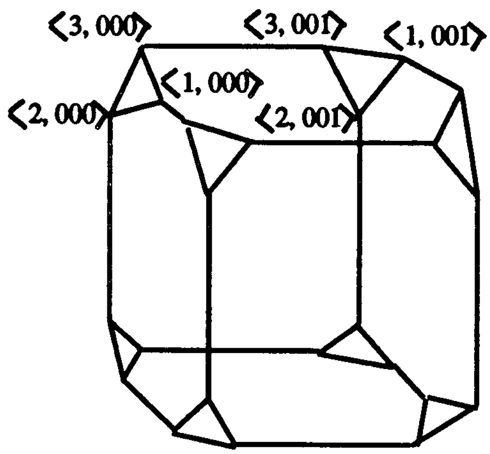


Figure 2.2 : 3-dimensional CCC

Two nodes (i, w) and (i', w') are adjacent in the CCC if and only if either

a) w = w' and | i - i' = mod r | = 1, or

b) i = i' and w differs from w' in precisely the ith bit .

Edges of the first type are called cycle edges, and edges of the second type are called hypercube edges. Figure 2.2 shows the 3-dimensional CCC.

When the first and last levels of the r-dimensional FFT are merged into a single level, the result is an r-level graph with $r\,2^r$ vertices, each of degree 4. We call this graph the butterfly network. In this case, two vertices (w, i) and (w', i') are linked by an edge if and only if i' = i + 1 mod r and either w = w' or w and w' differ in the i'th bit.

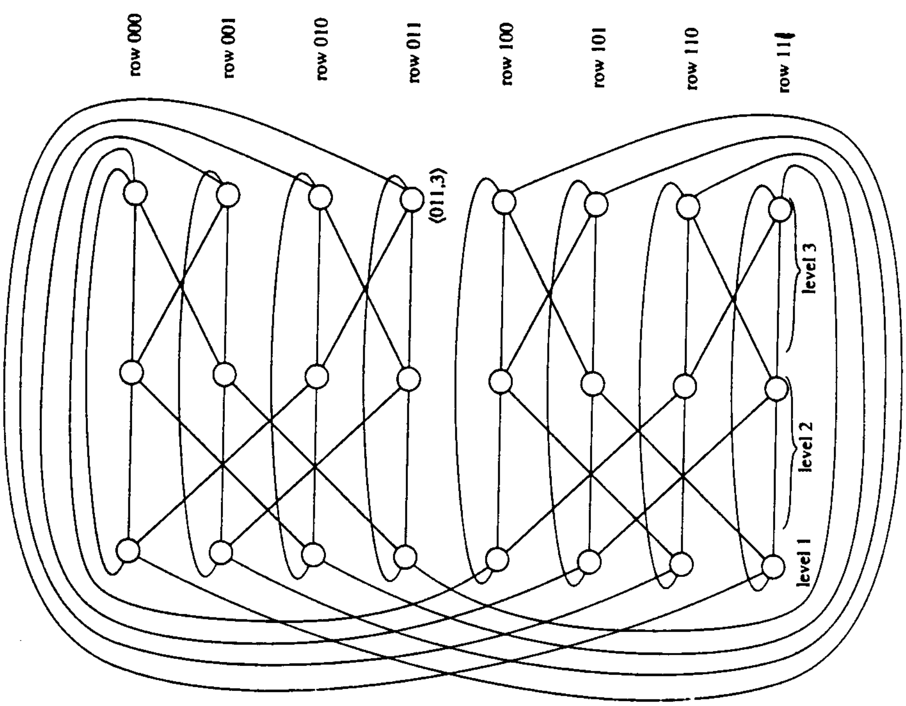


Figure 2.3 : 3-dimensional butterfly

Figure 2.3 shows the 3-dimensional butterfly.

Let $d_n = \{i_0, i_1, ..., i_{n-1}\}$ be an ordered sequence of n distinct integers. Also, let $d_n(p) = \{i_0, i_1, ..., i_p\}$, $0 \le p < n$. Then, the dimensional-representation of the n^{th} Gray code sequence G^{d_n} is defined recursively as follows.

$G^{d}_{n}(0) = i_0$

$G^{d}_{n}(k) = G^{d}_{n}(k-1) \,.\, i_k \,.\, G^{d}_{n}(k-1)$, where $1 \le k < n$; and

$G^{d}_{n} = G^{d}_{n}(n-1)$.

We denote by $G^{d}n\,(i)$ the i^{th} element of $G^{d}n$.

For example, if we let $d_4 = \{1, 2, 3, 4\}$, $G^{d}1 = 1$, $G^{d}2 = 1, 2, 1$, $G^{d}3 = 1, 2, 1, 3, 1, 2, 1$, and $G^{d}4 = 1, 2, 1, 3, 1, 2, 1, 4, 1, 2, 1, 3, 1, 2, 1$.

Node-representation of a binary reflected Gray code of all n-bit binary numbers, G_n is defined recursively as follows.

$G_1 = (0, 1)$.

$G_k = (\,0G_{k-1},\ 1(G_{k-1})^r\,)$ for $k > 1$

where $(G_{k-1})^r$ denote the sequence of binary strings obtained from G_{k-1} by reversing the order of the strings in G_{k-1}. For example, we know that $G_2 = (00, 01, 11, 10)$ and $G_3 = (000, 001, 011, 010, 110, 111, 101, 100)$. We denote by G_n (i) the ith element of G_n.

2.3 Embeddings

2.3.1 Embedding an FFT into a Hypercube.

In this section, we propose an algorithm for embedding an r-dimensional FFT, F(r) into the smallest subbanyan BH(h, k, 2) with at least $(r + 1)\, 2^r$ nodes, i.e, $k > r$ and $h\, 2^k \ge (r + 1)2^r > (h - 1)\, 2^k$. We begin by describing a technique for embedding F(m) into d_m dimensional hypercube Q(d_m) where $d_m = m + \lceil \log (m + 1) \rceil$.

Our algorithm for embedding an FFT into a subbanyan is related to this approach.

Embedding F(m) into Q(d_m), where $d_m = m + \lceil \log (m+1) \rceil$ is accomplished by labeling vertex $v_0 =_{def} (0, 00....0)$ of F(m) with the length-d_m string of 0's (thereby assigning it to vertex 00...0 of Q(d_m) in the embedding) and by using a single pair (s_i , c_i) of bit-positions, called a bp-pair (bit-position pair), for label assignments to edges between levels i - 1 and i of F(m), $1 \le i \le m$; in particular, all straight-edges between these levels flip bit-position s_i ,and all cross-edges between these levels flip bit-positions c_i .

A levelled bp-pair sequence (LBPS, for short), S (m) = (s_1, c_1), (s_2, c_2)..... (s_m, c_m) has been defined in [Gre 90]as follows.

- $s_L = G^{d_\lambda} (L)$

- c_L = the Lth largest integer in the set $d_{m+\lambda} - d_\lambda = \{\lambda + 1, \lambda + 2, ..., m + \lambda\}$

for all $L \in \{1, 2, 3,......,m\}$, where $\lambda = \lceil \log (m+1) \rceil$, $d_{m+\lambda} = \{1, 2, 3, 4,...., m + \lambda\}$and $d_\lambda = \{1, 2, 3, 4,...., \lambda\}$.

2.3.2 Embedding an FFT into a Banyan-Hypercube

Theorem 2.1.

For any r, the r-dimensional FFT graph, F(r) with $(r + 1)2^r$ nodes is a subgraph of the graph induced by all the nodes in any h adjacent levels of the BH(k + 1, k, 2), where k > r and $h\, 2^k \ge (r + 1)2^r > (h - 1)\, 2^k$. Thus every FFT graph is embeddable with unit dilation in an optimal subbanyan-hypercube.

Lemma 2.2[You2 90].
Any subgraph of h adjacent levels (of a full banyan-hypercube of k + 1 levels) is isomorphic to BH(h, k, s) .

Hence, it is sufficient to embed F(r) into the BH(h, k, s) with dilation 1.

The embeddings of F(r) into BH(h, k, 2), necessary to prove Theorem 2.1 are specified via three labelling schemes :

- We relabel each vertex $v = (L, x_{k-1} \ldots x_{L+1}\, x_L\, x_{L-1} \ldots\, x_1\, x_0)$ in BH(h, k, 2) as $G_p\,(L + 1) \mid x_{k-1} \ldots x_{L+1}\, x_L\, x_{L-1} \ldots\, x_1\, x_0$, where $p = \lceil \log h \rceil$, $0 \le L \le h - 1$, $x_i = 0$ or 1 for all $0 \le i \le k - 1$ and $\mid$ is the concatenation operation. For example, v = (2, 0100) in BH(3, 4, 2) is relabelled as $G_2\,(3) \mid 0100 = 110100$.

An edge between two nodes in a relabelled banyan - hypercube that differ in the i-th bit is said to lie in the i-th dimension. Also edge between two nodes $u = G_p\,(L) \mid x_{k-1} \ldots x_{L+1}\, x_L\, x_{L-1} \ldots\, x_1\, x_0$ and $v = G_p\,(L + 1) \mid x_{k-1} \ldots x_{L+1}\, x_L\, cx_{L-1} \ldots\, x_1\, x_0$, i. e, $(L - 1, x_{k-1} \ldots x_{L+1}\, x_L\, x_{L-1} \ldots\, x_1\, x_0)$, and $(L, x_{k-1} \ldots x_{L+1}\, x_L\, cx_{L-1} \ldots\, x_1\, x_0)$, where

cx_{L-1} is the complement of x_{L-1} is said to lie in dimensions t_L (Nodes u and v are adjacent along the cross edge in BH(h, k, 2)) ; Here nodes u and v differ in two bit positions, i.e, L and σ, where σ is the dimension between two nodes, $G_p\,(L) \mid x_{k-1} \ldots x_{L+1}\, x_L \ldots\, x_1\, x_0$, and $G_p\,(L + 1) \mid x_{k-1} \ldots x_{L+1}\, x_L \ldots\, x_1\, x_0$.

For example, traversing an edge of dimensions t_1 in BH(h, k, 2) flips the bit-positions 1, and 1 + k.

- We assign to each vertex v of F(r) a unique $\lceil \log h \rceil + k$ - bit label L(v), which is it's image vertex in the relabelled BH(h, k, 2).

- We assign to each edge (u, v) of F(r) a bit position label B(u, v) $\in$ { h, h + 1,......, $\lceil \log h \rceil + k$, t_1, t_2,........., t_{h-1} } such that L(u) and L(v) differ only in bit-position B(u, v).

That is, our embedding of F(r) into BH(h, k, 2) is specified by means of a LBPS

$$S(r) = (s_1, c_1), (s_2, c_2)..... (s_r, c_r)$$

as is the case of [Gre90].

We say that a dimension sequence among the set of dimensions, D generates a path from vertex w to vertex x in some graph, if we traverse the edges of each dimension in the dimension sequenece, vertex w reaches the vertex x. For example, the dimension sequence 0, 1, 2, 0 among the set D = {0, 1, 2, 3} generates a path from the node 0001 to the node, 0111 in the 4-dimension hypercube.

Lemma 2.3.

Given any two nodes u, and v in the relabelled BH(h, k, 2), where $h \le k + 1$, there is a dimension sequence among the set of dimensions, D ={ h, h + 1,.., $\lceil \log h \rceil + k$, t_1, t_2,.., t_{h-1} }.

Proof:

Trivially, for any two nodes u, and v in BH(h, k, 2), there is a dimension sequence among the set of dimensions, D_1 = {1, 2,... h,...., $\lceil \log h \rceil + k$, t_1, t_2,........., t_{h-1} }. But if we traverse edge of dimension d, $1 \le d \le h - 1$, from a given node, and arrive at some node w, then by just traversing an edge of dimension t_d and an edge of dimension $\sigma = k + d$, we can arrive at node w from the given node. Hence, there is also a dimension sequence among the set of dimensions, $D = \{ h, h + 1, \ldots, \lceil \log h \rceil + k, t_1, t_2, \ldots, t_{h-1} \}$.

An algorithm for embedding F(r) into BH(h, k, 2) with $k > r$ and $h\, 2^k \geq (r + 1)2^r > (h - 1)\, 2^k$ is as follows.

Algorithm

Let $p = k - r$, $d_\lambda = \{r + 1, r + 2, \ldots, r + p - 1, k, \ldots, k + \lceil \log h \rceil \}$, and G^{d_λ} be the dimensional-representation of the λth Gray-code.

- $c_1 = h$.

  $c_L = t_x$ if $L \bmod 2^p = 0$ and $L = x\, 2^p$

  $c_L = c_{(L-1)} + 1$ if $L \bmod 2^p \neq 0$ and $(L - 1) \bmod 2^p \neq 0$.

  $c_L = c_{(L-2)} + 1$ if $L \bmod 2^p \neq 0$ and $(L - 1) \bmod 2^p = 0$.

- $s_L = G^{d_\lambda}(L)$

for all $L \in \{1, 2, 3, \ldots, r\}$, where $\lambda = \lceil \log h \rceil + p = \lceil \log h \rceil + k - r$.

Example 2.1 : Let us try to embed a 2-dimensional FFT with $3 * 2^2 = 12$ vertices into BH(2, 3, 2) with $2 * 2^3 = 16$ vertices. By the algorithm $c_1 = 2$, $c_2 = t_1$, $s_1 = 3$, $s_2 = 4$.

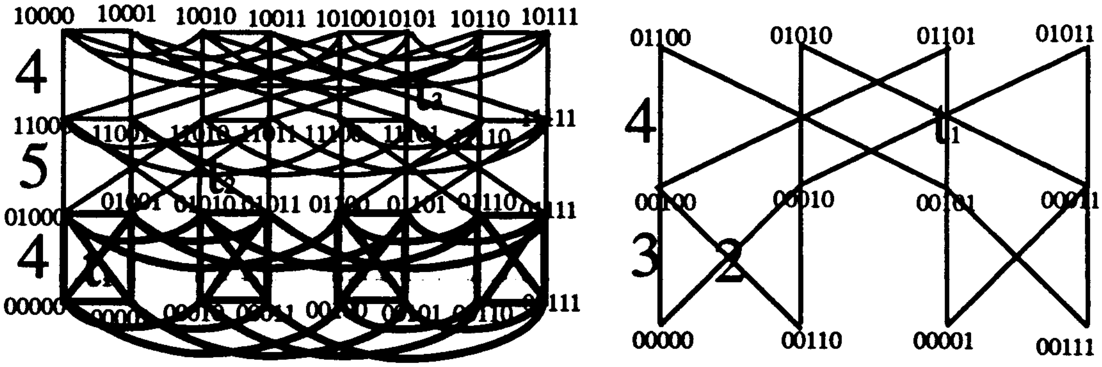


Figure 2.4 : Embedding F(2) into relabeled BH(2, 3, 2)

Example 2.2 : Let us try to embed 4-dimensional FFT with $5 * 2^4 = 80$ vertices into BH(3, 5, 2) with $3 * 2^5 = 96$ vertices.

By the algorithm, we get the LBPS as follows

$$c_1 = 3, c_2 = t_1, c_3 = 4, c_4 = t_2$$
$$s_1 = 5, s_2 = 6, \quad s_3 = 5, s_4 = 7.$$

with vertex v = (0, 0000) in FFT mapped into node L(v) =0000000 in BH(3, 5, 2).

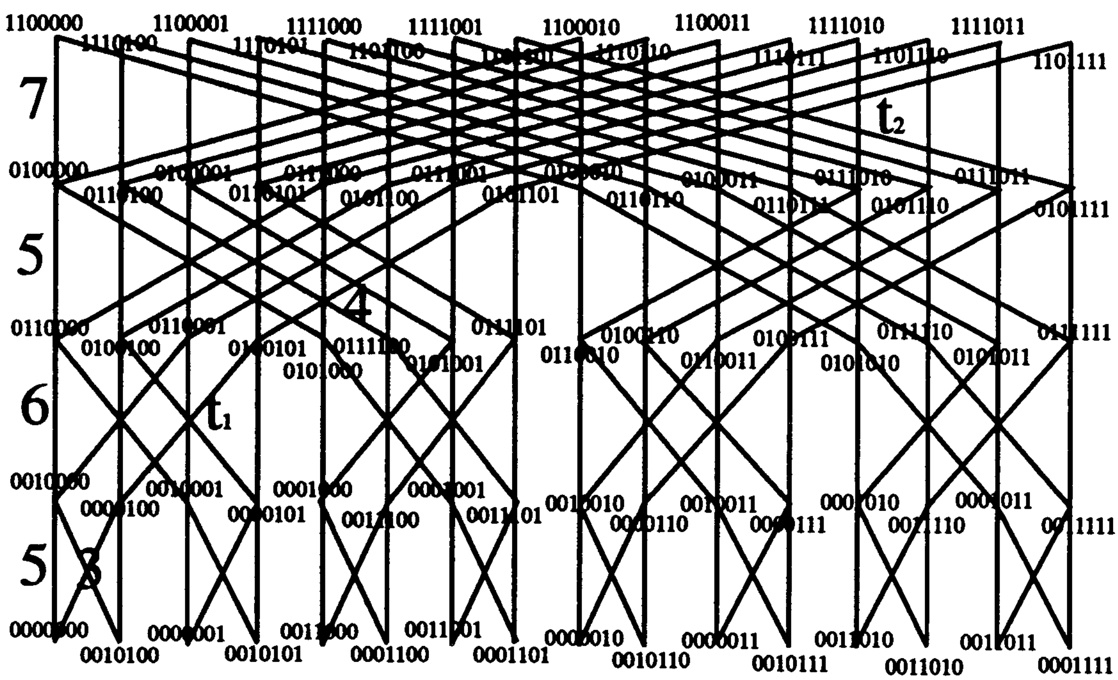


Figure 2.5: Embedding F(4) into BH(3, 5, 2)

Lemma 2.4.

Bit-pair sequence, (s_{L2P}, c_{L2P}) from level $L2^P - 1$ to level $L2^P$ of F(r), when embedding F(r) into BH(h, k, 2) with $p = k - r$ is $(\sigma, L \oplus \sigma)$, for $1 \le L \le h - 1$, where σ is the dimension between two nodes, $(L - 1, x_{k-1} \ldots x_{L+1}\ x_L\ x_{L-1} \ldots\ x_1\ x_0)$, and$(L, x_{k-1} \ldots x_{L+1}\ x_L\ x_{L-1} \ldots\ x_1\ x_0)$ in BH(h, k, 2), and $L \oplus \sigma$ means complementing bit positions L and σ.

Proof : By the algorithm, $c_{L2P} = t_L =_{def} L \oplus \sigma$.

we can prove $s_{L2P} = \sigma$, using the following basic facts.

- $s_{L2P} =_{def} G^{d_\lambda}\ (L2^P)$, where $d_\lambda = \{r + 1, r + 2, \ldots, r + p - 1, k, \ldots, k + \lceil \log h \rceil\}$.

- σ is $G^{d_\delta}\ (L)$, where $d_\delta = \{ k + 1, k + 2, \ldots, k + \lceil \log h \rceil \}$.

  By definition, σ is the dimension between two nodes, $G_n(L) \mid x_{k-1} \ldots x_{L+1}\ x_L\ x_{L-1} \ldots\ x_1\ x_0$, and $G_n(L + 1) \mid x_{k-1} \ldots x_{L+1}\ x_L\ x_{L-1} \ldots\ x_1\ x_0$, where $n = \lceil \log h \rceil$. This is nothing but $G^{d_\delta}\ (L)$.

- $G^{d_\lambda}\ (L2^P) = G^{d_\delta}\ (L)$.

  Trivially, $s_{2P} = G^{d_\lambda}\ (2^P) = k + 1 = G^{d_\delta}\ (1)$.

  Assume that $G^{d_\lambda}\ ((L - 1)2^P) = G^{d_\delta}\ (L - 1)$, for $L > 2$.

  $G^{d_\lambda}\ (L2^P) = G^{d_\lambda}\ ((L - 1)2^P + 2^P) = G^{d_\delta}\ (L)$ by inspection.

Lemma 2.5.

When we assign to $v_0 =_{def} (0, 00....0)$ of F(r) string of 0's of length $\lceil \log h \rceil + k$ in the relabelled BH(h, k, 2), where $(h-1)\, 2^k \leq (r + 1)\, 2^r \leq h\, 2^k$, the LBPS in the algorithm completely determines the labels of all remaining vertices within the given BH(h, k, 2).

Proof :

It is sufficient to show that any vertex $v \in F(r)$ is mapped to the vertex L(v) in BH(h, k, 2) by the LBPS. We shall show that all the labels of vertices from level 0 to level $2^p - 1$ in F(r), mapped by the LBPS come from the vertices of level 0 in BH(h, k, 2), where $p = k - r$. Similarily, all the labels of vertices from level $n * 2^p$ to level $n * 2^p + 2^p - 1$ in F(r) comes from the vertices of level n in BH(h, k, 2), where $1 \leq n \leq h - 1$.

Let us take the label i corresponding to vertex $(2^p - 1, 0...0)$ in level $2^p - 1$ of F(r). The label of this vertex comes from some node in level 0 of BH(h, k, 2), because label 00...00, corresponding to vertex (0, 00...0) in F(r) comes from level 0 in BH(h, k, 2), and by the algorithm, we can arrive at label i (vertex $(2^p - 1, 0...0)$ in F(r)) from label 00...0 (vertex (0, 00...0) in F(r)) by just traversing edges of dimension sequence among {r + 1,..., k}.

Let us take another label j corresponding to some vertex in level $2^p - 1$ of F(r). By Lemma 2.3, there exists a dimension sequence among $\{ h, h + 1, \ldots, \lceil \log h \rceil + k, t_1, t_2, \ldots, t_{h-1} \}$ that generates a path from label i to label j. By Lemma 2.4, the bit-pair sequence, (s_{L2^p}, c_{L2^p}) from level $L2^p - 1$ to level $L2^p$ in F(r) is $(\sigma, L \oplus \sigma)$, where

σ is the dimension between two nodes, $(L - 1, x_{k-1} \ldots x_{L+1}\, x_L\, x_{L-1} \ldots x_1\, x_0)$, and$(L, x_{k-1} \ldots x_{L+1}\, x_L\, x_{L-1} \ldots x_1\, x_0)$ for $1 \le L \le h - 1$. Hence $(s_{2^P}, c_{2^P}) = (k + 1, (1 + k) \oplus 1)$.

But the basic fact in F(r) is that for each level $L \in \{0, 1, 2, \ldots, r\}$, the number of level - L edges, i.e, edges from level L to level L + 1, appear an even number of times in the path from label i to label j, because label i and label j are in the same level of F(r). Therefore, for each dimension s_{m2^P}, in the dimension sequence from label i to label j with $1 \le m \le h - 1$, another corresponding s_{m2^P} or c_{m2^P} must also be in that dimension sequence. Similarily, for each dimension c_{m2^P}, in the dimension sequence from label i to label j with $1 \le m \le h - 1$, another corresponding c_{m2^P} or s_{m2^P} must also be in that dimension sequence.

Hence, traversing all the edges of dimension , s_{L2^P}, and c_{L2^P} for $1 \le L \le h - 1$ in the dimension sequence from label i to label j, just affects bit position L. Also by the algorithm, traversing the edges of remaining dimensions in the dimension sequence just changes bit position m, for $h \le m \le k$. Therefore, label j is also from some vertex in level 0 of BH(h, k, 2).

If all the labels in level $2^P - 1$ are from the vertices in level 0 of BH(h, k, 2), then trivially all the labels from level 0 to level $2^P - 2$ also come from the vertices in level 0 of BH(h, k, 2). In a similar way, we can show that all the labels of vertices from level $n * 2^P$ to level $n * 2^P + 2^P - 1$ in F(r) come from the vertices of level n in BH(h, k, 2), where $1 \le n \le h - 1$.

The proof of Lemma 2.6 is almost same as that of Proposition 1 in [Gre90].

Lemma 2.6.

Any mapping of the vertices of the FFT graph to the vertices of the BH(h, k, 2) induced by the LBPS in the algorithm is well-defined.

Proof:

The proof is based on the three basic facts about cycles in an m-dimensional FFT.

1. For each level $l \in \{0, 1, 2,.., m\}$, the number of level-l edges in any cycle in F(m) is even.

2. For each level $l \in \{0, 1, 2,.., m\}$, the number of level-l cross-edges in any cycle in F(m) is even.

3. For each level $l \in \{0, 1, 2,.., m\}$, the number of level-l straight-edges in any cycle in F(m) is even.

The proof of Lemma 2.7 is similar to that of validation in [Gre90].

Lemma 2.7.

The mapping of the vertices of the FFT graph to the vertices of the BH(h, k, 2) that is induced by the LBPS in the algorithm is one-to-one.

Proof:

Let us take two arbitrary nodes, u, and v in F(m).

If they are from the same column, i.e, for u = (i, w), and v =(i', w'), w = w', the dimension sequence in the straight path from node u to node v, mapped into BH(h, k, 2) is a subsequence of the Gray code sequence, that forms a Hamiltonian cycle. Hence nodes, u and v are mapped into different nodes in BH(h, k, 2).

Now assume that nodes, u, and v are in different columns. Let u' be the vertex of F(m) in the bottom level of the same column as u, and let v' be the vertex of F(m) in the top level of the same column as v. Consider the path P in F(m) that starts at u, traverses all the straight-edges up to node u', follows the unique length-m path from u' to v' (there exists a unique length-m path from any node in the bottom level of F(m) to the other node in the top level of F(m)), and finally traverses straight-edges up to node v.

Here let us think about the unique length-m path P' from u' to v'. The dimension sequence from node u' to the node v' in the path P' , mapped into BH(h, k, 2) includes some dimension d, $d \in \{h, h+1, .., k, t_1, t_2, ..., t_{h-1}\}$ that is shared by no cross-edge, and no straight-edge at any other level of F(m), mapped into BH(h, k, 2). Hence in this case also, two nodes u' and v' are mapped into different nodes in BH(h, k, 2).

2.3.3 Embedding a CCC into a Banyan-Hypercube

Theorem 2.2

For any r, the r-dimensional cube-connected cycle (CCC) graph, C(r) with $r2^r$ vertice is a subgraph of the graph induced by all the nodes in any h adjacent levels of the BH(k + 1, k, 2), where k > r and $h\,2^k \geq r\,2^r > (h-1)\,2^k$.

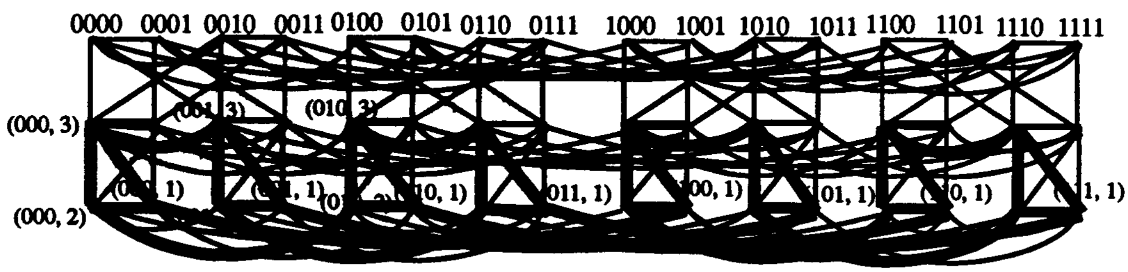


Figure 2.6 : Embedding of 3-dimensional CCC into BH(3, 4, 2).

By Lemma 2.2, it suffices to embed C(r) into BH(h, k, 2) with dilation 1, where $k > r$ and $h\,2^k \geq r\,2^r > (h-1)2^k$.

Lemma 2.8.

Let C_r be the ring with r vertice. Then $C_r \otimes Q_r$ is the subgraph of BH(h, k, 2), where Q_r denotes the r-dimensional hypercube, $k > r$ and $h\,2^k \geq r\,2^r > (h-1)\,2^k$.

Proof : Let L_h be the line of length h.

- $L_h \otimes Q_{r1} \otimes Q_r$ is the subgraph of BH(h, k, 2), where $r1 + r = k$ [Kwo1].
- Let S be the induced subgraph of BH(h, k, 2) by all the nodes in $L_h \otimes Q_{r1}$, then, $S \otimes Q_r$ is the subgraph of BH(h, k, 2).
- Let |S| be the number of vertices in the graph S, then $|S| \geq r$, and we can embed ring of length r in S as in the proof of Theorem 1 in [Kwo1].

Lemma 2.9. The r-dimensional CCC with $r * 2^r$ vertices is a subgraph of $C_r \otimes Q_r$

Proof: The proof follows from the definitions of the CCC and of $C_r \otimes Q_r$

2.3.4 Embedding a Butterfly network into a Banyan-hypercube.

Lemma 2.10[Lei92].

An r-dimensional butterfly can be embedded one-to-one with dilation 2 into the r-dimensional FFT.

Theorem 2.3.

The r-dimensional butterfly with $r\ 2^r$ nodes can be embedded into BH(h, k, 2) with dilation 2, where $h\ 2^k \geq (r + 1)\ 2^r > (h - 1)\ 2^k$

Proof : The proof follows directly from Theorem 2.1, and Lemma 2.10

Chapter 3

Embedding Ring-structured Graphs into Banyan-Hypercube

3.1 Introduction

In this chapter, we consider results on embedding networks with ring connections such as ring, chordal ring, X-tree, torus, and Illiac networks into BH network.

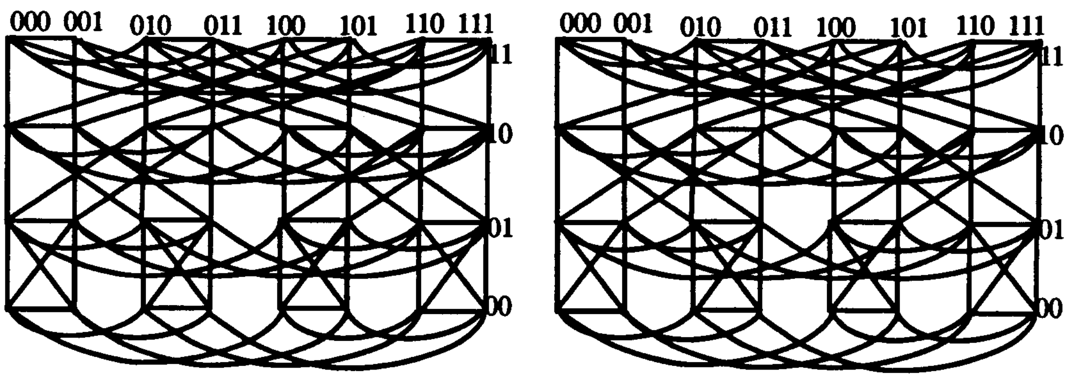
The rest of the chapter is organized as follows. Section 3.2 introduces the necessary definitions and notations. Embedding of ring structured networks into banyan hypercube will be discussed in section 3.3.

3.2. Preliminaries

In this section, we give some definitions and preliminary results useful for the subsequent sections.

An embedding f of a graph $G = (V_g, E_g)$ to a BH(h, k, s) = (V_h, E_h) is a one-to-one mapping from V_g to V_h. For a mapping function f, the dilation cost of an edge (v, w) $\in E_g$ is d(f(v), f(w)), i.e, the distance between f(v) and f(w). The average dilation of the mapping f is the total sum of the dilation cost of each edge in E_g divided by the number of edges, $|E_g|$. If we define $M_d = |\{ (a, b) \mid d(f(a), f(b)) = d, (a, b) \in E_g\}|$, then ΣM_d, for $d = 1 \ldots d_{bh}$ is directly related to average dilation, where d_{bh} is the diameter of the graph, BH(h, k, s).

We say that a graph $G = (V, E)$ is the cross product of graphs $G_1 = (V_1, E_1)$, $G_2 = (V_2, E_2)$,........, $G_k = (V_k, E_k)$ if $V = \{(v_1, v_2, \ldots v_k) \mid v_i \in V_i \text{ for } 1 \le i \le k\}$ and $E = \{((u_1, u_2, \ldots, u_k), (v_1, v_2, \ldots, v_k)) \mid \exists\, j \text{ such that } (u_j, v_j) \in E_j \text{ and } u_i = v_i \text{ for all } i \ne j\}$. Notationally, we represent the cross product as $G = G_1 \otimes G_2 \otimes \ldots\ldots \otimes G_k$.



(a) Node labels in the same level in increasing order

(b) Node labels in the same level in binary reflected Gray code order

Figure 3.1 : Banyan-Hypercubes : BH(4, 3, 2)

Figure 3.1a and 3.1b shows banyan-hyperecubes BH(4, 3,2), where nodes of each level in Figure 3.1a are in increasing order, and those in Figure 3.1b are in binary reflected Gray code order.

In the next section, the properties of the Gray codes are used in embedding. In a Gray code, every adjacent words differ in one bit. The binary reflected Gray code can be formally defined as follows.

Definition 3.1 : Let G_n denote a binary reflected Gray code of all n-bit binary numbers. Then, G_n is defined recursively as follows.

$$G_1 = (0, 1).$$

$G_k = (0G_{k-1}, 1(G_{k-1})^r)$ for $k > 1$ where $(G_{k-1})^r$ denote the sequence of binary strings obtained from G_{k-1} by reversing the order of the strings in G_{k-1}. For example, we know that $G_2 = (00, 01, 11, 10)$ and $G_3 = (000, 001, 011, 010, 110, 111, 101, 100)$.

Lemma 3.1 :
Let G_k be the binary reflected Gray code for the k-cube and $G_k(r)$ denote the (r+1)th element of G_k, where $r = 0, ...2^k-1$, then $G_k(i)$ and $G_k(j)$ are adjacent if $i + j = 2^k-1$.

Proof :

When $k = 1$, it is trivial. Hence it is sufficient to prove for $k > 1$.
Without loss of generality, assume that $i < j$.
Since $i + j = 2^k - 1$, trivially $G_k(i)$ and $G_k(j)$ belong to $0G_{k-1}$, and $1(G_{k-1})^r$ respectively. Let ip and jp be the relative positions of $G_k(i)$ and $G_k(j)$ in $0G_{k-1}$, and $1(G_{k-1})^r$ respectively ; then $ip = i$ and $jp = j - 2^{k-1}$. By the definition of reverse operation, $(G_{k-1})^r(x)$ is equal to $G_{k-1}(y)$ if y is $2^{k-1} - x - 1$, $x = 0,.... 2^{k-1} - 1$, that is, if $x + y = 2^{k-1} - 1$. Since $ip + jp = i + j - 2^{k-1} = 2^k - 1 - 2^{k-1} = 2^{k-1} - 1$, $G_{k-1}(ip) = (G_{k-1})^r(jp)$. Since $0G_{k-1}(ip)$ and $1(G_{k-1})^r(jp)$ differs one bit in G_k, $0G_{k-1}(ip)$ and $1(G_{k-1})^r(jp)$ are adjacent nodes in the k-cube. But ip and jp are the relative position of $G_k(i)$ and $G_k(j)$ in $0G_{k-1}$, and $1(G_{k-1})^r$ respectively. Hence $G_k(i)$ and $G_k(j)$ are adjacent in k-cube.

For example, in G_3, $G_3(1) = 001$, and $G_3(6) = 101$ are adjacent.

Lemma 3.2 [Lei92] For any $k \geq 1$ and $r = r1 + r2 + ... + rk$, the r-dimensional hypercube Q_r can be expressed as the cross product

$$Q_r = Q_{r1} \otimes Q_{r2} \otimes \dots\dots\dots \otimes Q_{rk}$$

where Q_{ri} denotes the ri-dimensional hypercube for $1 \leq i \leq k$.

Lemma 3.3 [Lei92] If $G = G_1 \otimes G_2 \otimes \ldots\ldots\ldots \otimes G_k$ and $G' = G'_1 \otimes G'_2 \otimes \ldots\ldots\ldots \otimes G'_k$ for some $k \geq 1$, and G_i is a subgraph of G_i' for $1 \leq i \leq k$, then G is a subgraph of G'.

3.3 Embeddings

3.3.1 Embedding rings into a Banyan-Hypercube.

In a bipartite graph, the nodes in the graph can be partitioned into two disjoint non-empty subsets such that all the edges are connected from nodes in one subset to the nodes in the other subset. In the case of hypercube, if we partition the set of nodes on the basis of the parity, it is easy to see that the connections are only from the odd parity nodes to the even parity nodes. Thus, hypercube is a bipartite graph.

One important result in graph theory is that a graph is bipartite iff all the cycles are of even length [Swa81]. Thus cycle of odd length can't be embedded in hypercube with dilation 1. However, in banyan-hypercube, using the edges in banyan structure and hypercube together, we can embed a cycle of any length into the banyan-hypercube with dilation 1 as shown in Theorem 3.1.

Theorem 3.1 : A ring with n nodes, $n \leq h2^k$, can be embedded into a BH(h, k, 2) with dilation 1.

Proof :

Let us assume that binary reflected Gray code G_i for i-cube in level L of banyan-hypercube is denoted as $[L, G_i]$, and (r + 1)th element of $[L, G_i]$ as $[L, G_i][r]$.

case 1: n : even and $n \leq 2^k$.

From Lemma 3.1, this even length cycle can be embedded in any level of banyan-hypercube .

case 2: n : even and $p2^k \leq n < (p + 1) 2^k$ where $2 \leq (p + 1) \leq h$.

Let $n - p2^k$ be t. At level 0, t elements will be embedded, using the binary reflected Gray code property as shown in Lemma 3.1. $p 2^k$ nodes are embedded in p adjacent levels of the BH.

The even length cycle is given by

$([0, G_k][0], [0, G_k][1], \ldots\ldots, [0, G_k][t/2 - 1], [0, G_k][2^k - t/2], [0, G_k][2^k - t/2 + 1], [0, G_k][2^k - t/2 + 2] \ldots\ldots [0, G_k][2^k - 1]) ([1, 1G_{k-1}], [2, 1(G_{k-1})^r], [3, 1G_{k-1}], \ldots\ldots, [p, 1G_{k-1}], [p, 0(G_{k-1})^r], [p-1, 0G_{k-1}] \ldots\ldots [1, 0(G_{k-1})^r])$

when p is odd.

and

$([0, G_k][0], [0, G_k][1], \ldots\ldots, [0, G_k][t/2 - 1], [0, G_k][2^k - t/2], [0, G_k][2^k - t/2 + 1], [0, G_k][2^k - t/2 + 2] \ldots\ldots [0, G_k][2^k - 1]) ([1, 1G_{k-1}], [2, 1(G_{k-1})^r], [3, 1G_{k-1}], \ldots\ldots, [p, 1(G_{k-1})^r], [p, 0G_{k-1}], [p-1, 0(G_{k-1})^r] \ldots\ldots [1, 0(G_{k-1})^r])$

when p is even.

case 3: n : odd and $2^{k-1} < n < 2^k$.

Let us embed the cycle in BH(2, k-1, 2). Let $t = n - 2^{k-1}$. Then the cycle is

$([0, G_{k-1}][0], [0, G_{k-1}][1], \ldots\ldots, [0, G_{k-1}][(t - 1)/2 - 1], [0, G_{k-1}][2^{k-1} - (t - 1)/2], [0, G_{k-1}][2^{k-1} - (t - 1)/2 + 1], [0, G_{k-1}][2^{k-1} - (t - 1)/2 + 2], \ldots\ldots [0, G_{k-1}][2^{k-1} - 2]) [1, 1(G_{k-2})], [1, 0(G_{k-2})^r]$

Note that node $[0, G_{k-1}][2^{k-1} - 2]$ is connected to the node $[1, 1(G_{k-2})][0]$ with the edge in banyan structure case 4: n : odd and $p2^k \le n < (p + 1)\, 2^k$ where $(p + 1) \le h$.

Let $n - p2^k$ be t. Then, the following sequence of nodes form a ring of size n .

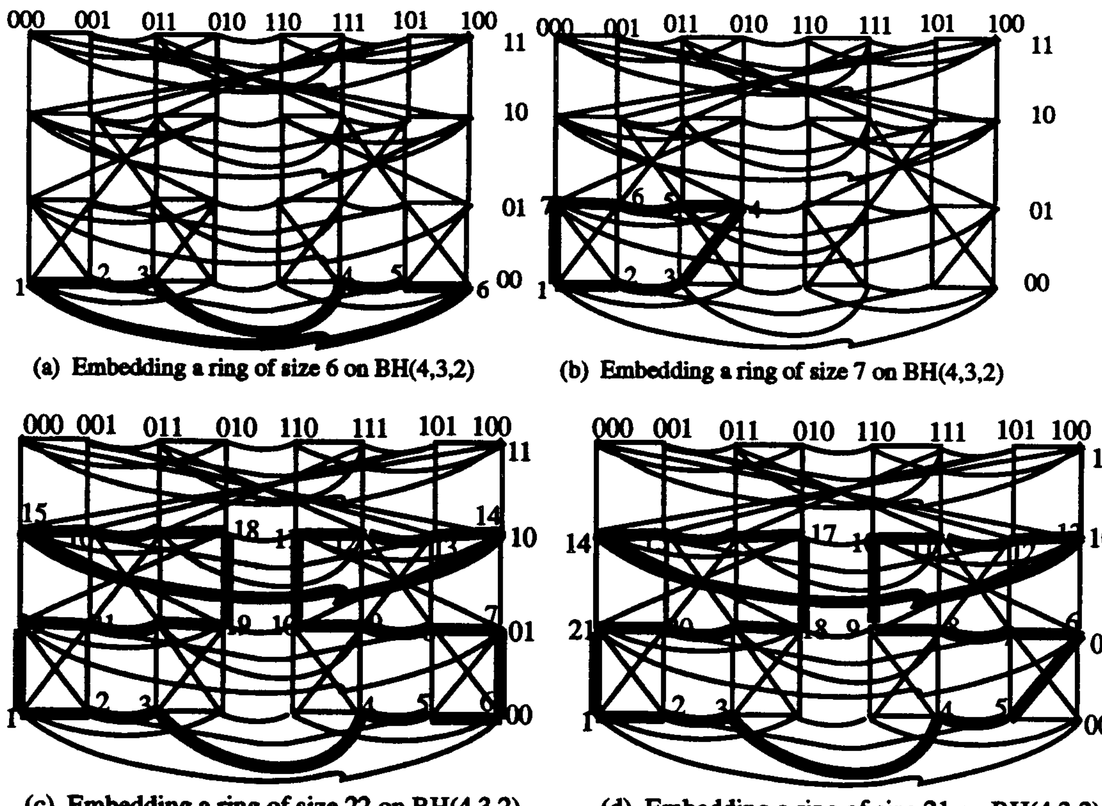
$([0, G_k][0], [0, G_k][1], \ldots\ldots, [0, G_k][(t-1)/2 - 1], [0, G_k][2^k - (t-1)/2]$

$[0, G_k][\, 2^k - (t-1)/2 + 1\,], [0, G_k][\, 2^k - (t-1)/2 + 2\,] \ldots\ldots [0, G_k][\, 2^k - 2\,]\,)$

$(\,[1, 1G_{k-1}], [2, 1(G_{k-1})^r], [3, 1G_{k-1}], \ldots\ldots\ldots, [p,\ 1G_{k-1}], [p, 0(G_{k-1})^r], [p-1, 0G_{k-1}] \ldots\ldots\ldots, [1,\ 0(G_{k-1})^r]\,)$

when p is odd

and



(a) Embedding a ring of size 6 on BH(4,3,2)

(b) Embedding a ring of size 7 on BH(4,3,2)

(c) Embedding a ring of size 22 on BH(4,3,2)

(d) Embedding a ring of size 21 on BH(4,3,2)

Figure 3.2 : Embedding of rings into BH(4, 3, 2)

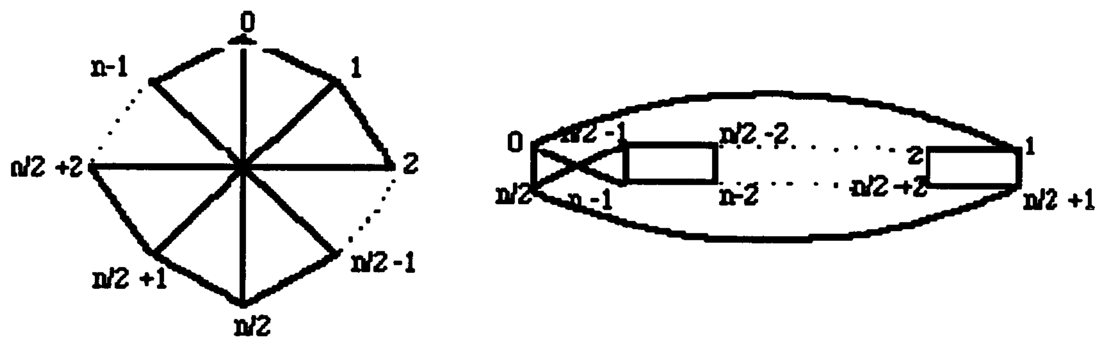
$([0, G_k][0], [0, G_k][1], \ldots\ldots, [0, G_k][(t-1)/2 - 1], [0, G_k][2^k - (t-1)/2],$
$[0, G_k][2^k - (t-1)/2 + 1], [0, G_k][2^k - (t-1)/2 + 2] \ldots\ldots [0, G_k][2^k - 2])$

$([1, 1G_{k-1}], [2, 1(G_{k-1})^r], [3, 1G_{k-1}], \ldots\ldots, [p, 1(G_{k-1})^r], [p, 0G_{k-1}], [p-1, 0(G_{k-1})^r] \ldots\ldots [1, 0(G_{k-1})^r])$

when p is even

Figure 3.2 illustrates the embedding of cycles of length 6, 7, 22, and 21 on BH(4, 3, 2).

3.3.2 Embedding Chordal Rings into a Banyan Hypercube.



[a] [b]

Figure 3.3 : Two layout of n nodes chordal ring with w = n/2

The Chordal Ring is a ring structured network, in which each node has an additional link, called a chord, to some other node across the network. The number of nodes in a chordal ring is assumed to be even and nodes are indexed 0, 1, 2, ,....,n - 1 around the ring. In the chordal ring, each odd-numbered node i (i = 1, 3,....,n - 1) is connected to a node (i + w) mod n. (This implies that each even-numbered node j (j = 0, 2,...n-2) is connected to a node (j - w) mode n.)

Here, w is called the chord length and is assumed to be positive [Ard81]. In this paper, we consider the case where w = n/2.

Lemma 3.4

A chordal ring with n nodes and w = n/2 (n > 5) can not be embedded with dilation 1 into any one level BH(h, k, 2), where $2^k > n$.

Proof :

Chordal ring of n nodes with w = n/2 is isomorphic to the Figure3.3 (b). Hence, it is sufficient to check whether Figure3.3 (b) can be embedded into any one level in BH(h, k, 2) with dilation 1. Since any level in BH(h, k, 2) is a hypercube of dimension k, let us embed into level 0 of BH(h, k, 2).

Label each edge in the the above chordal ring with a number that represents the dimension of the corresponding edge in hypercube of dimension k. Let's consider the edge between nodes 1, and (n + 2)/2 in the chordal ring. If the dimension of that edge is p, it can easily be checked that the dimension of all the vertical edges in Figure 3.3 (b) also must be p for an embedding of dilation 1 . That is, if we partition the above hypercube of dimension k into two hypercubes of dimension k - 1 with respect to dimension p, then all the nodes from 0,1,...up to n/2 - 1 in the above chordal ring appear in one hypercube of dimension k-1, and all the nodes from n/2,n/2 + 1,...up to n - 1 appear in another hypercube of dimension k-1. Here, node 0 in one hypercube of dimension k - 1 is connected to two nodes in another hypercube of dimension k - 1. This is impossible in hypercube.

Lemma 3.4 has also been proved differently in [Chu90]. The general concept ,used in proving Lemma 3.4 will be useful in proving Lemma 3.5.

Lemma 3.5 If n is not divisible by 4, a chordal ring with n nodes and w = n/2 (n > 5) can not be embedded with dilation 1 into the BH(2, k, 2), where $2*2^k > n$.

Proof :

By assumption, we can represent n as n = 4*p + 2, where p is some integer. Any subgraph of h adjacent levels (of a full banyan-hypercube of k + 1 levels) is isomorphic to BH(h, k, 2) [You290]. Hence, let's try to embed into levels 0 and 1 of BH(h, k, 2), i. e, BH(2, k, 2).

Here, BH(2, k, 2) has a hypercube of dimension k+1 as a subgraph . Let us denote all the edges between nodes, (0, x_k...... x_1 0) and (1, x_k...... x_1 1) as of dimension s, and all the edges between (1, x_k...... x_1 0) and (0, x_k...... x_1 1) as of dimension t, where x_i = 0 or 1, i = 1,..k. Label each edge in figure 3.3(b) with a number that represents the dimension of the corresponding edge in BH(2, k, 2).

If the embedding is of dilation 1, then it can be easily checked that edges of dimension s and those of dimension t must appear the same number of times in the embedding. Also if edges of dimension t or s appear r > 1 times in the embedding, then we can also embed in such a way that only one edge of dimension s, say edge (0, n-1) and only one edge of dimension t, say edge (n/2-1, n/2) appear.

Let's consider the edge between nodes 0, and n/2 in chordal ring. Then, the dimension of that edge in the BH(2, k, 2) must be either 1, or k+1. This can be proved similar to the proof of Lemma 3.4. Let us assume that the edge (0, n/2) is of dimension k+1 in BH(2, k, 2). Then, all the corresponding edges from node set (0, 1, 2,.....n/2-1) to another node set (n/2,.......,n-1) must be of dimension k+1.

Since level 0 of BH(2, k, 2) is a hypercube of dimension k, and the hypercube is a regular structure, without loss of generality, we can assume that node 0 in the chordal ring is mapped into (0, 0......0). Then, by assumption, nodes n/2-1, n/2, n-1 are mapped into nodes (0, 000...1), (1, 00....0), and (1, 00...1) respectively. That is, all the nodes 0, 1,....n/2-1 appear in the 0 level of BH(2, k, 2), and all the nodes n/2,...n-1 in the 1 level of BH(2, k, 2).

Here, closed path 0, 1, 2, 3,...,n/2-1, 0 form a cycle of odd length. This gives a contradiction because cycle of odd length can't appear in the hypercube.

In the case of edge (0, n/2) being of dimension 1, a similar proof can be given.

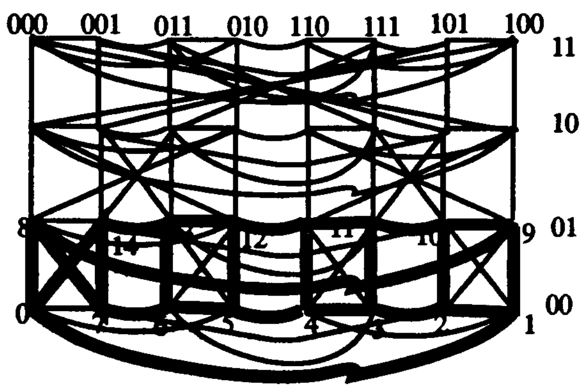


Figure 3.4 : Embedding Chordal ring of size 16 into BH(4, 3, 2)

Theorem 3.2 :

If n is divisible by 4, a chordal ring with n nodes and w = n/2 (n > 5) can be embedded with dilation 1 into the BH(2, k, 2), where $2*2^k \geq n$;

otherwise into BH(2, k, 4), where $2*4^k \geq n$.

Proof :

case 1: n is divisible by 4.

Let t = n/2. Now, it can be easily seen that the following sequence of nodes form a chordal ring of size n.

$([0, G_k][0], [0, G_k][\,2^k - 1\,], [0, G_k][\,2^k - 2\,] ....., [0, G_k][\,2^k - t/2 + 1\,], [0, G_k][\,2^k - t/2\,], [0, G_k][t/2 - 1], [0, G_k][t/2 - 2], ... [0, G_k][1])$
$([1, G_k][0], [1, G_k][\,2^k - 1\,], [1, G_k][\,2^k - 2\,] ....., [1, G_k][\,2^k - t/2 + 1\,], [1, G_k][\,2^k - t/2\,], [1, G_k][t/2 - 1], [1, G_k][t/2 - 2], ... [1, G_k][1])$

case 2: n is not divisible by 4, i.e, $n = 4y + 2$, y : positive integer.

Let $t = (n - 2)/2$. The cycle is given by
$([0, G_{2k}][0], [0, G_{2k}][\,4^k - 1\,], [0, G_{2k}][\,4^k - 2\,] ....., [0, G_{2k}][\,4^k - t/2 + 1\,], [0, G_{2k}][\,4^k - t/2\,], [0, G_{2k}][\,4^k - t/2 - 1\,], [0, G_{2k}][t/2], [0, G_k][t/2 - 2], ..., [0, G_k][2])$ $([1, G_{2k}][0], [1, G_{2k}][\,4^k - 1\,], [1, G_{2k}][\,4^k - 2\,] ....., [1, G_{2k}][\,4^k - t/2 + 1\,], [1, G_{2k}][\,4^k - t/2\,], [1, G_{2k}][\,4^k - t/2 - 1\,], [1, G_{2k}][t/2], [1, G_k][t/2 - 2], ..., [1, G_k][2])$.

The embedding of a ring of length 2^n in an n-cube may be defined as a sequence of nodes, $R = (v_1, v_2, ..., v_{2^n})$, where each adjacent nodes v_i and $v_{(i+1) \bmod 2^n}$, for $i=1,...2^n$, are neighbors across some dimension d_i in the n dimensional hypercube. The same embedding can be represented in terms of the sequence of dimensions that the adjacent nodes go across, i.e, $S = (d_1, d_2, ...d_{2^n})$.

Lemma 3.6

Let Q_n be a hypercube of dimension n. Then, we can find an embedding sequence, S for the Hamiltonian cycle, in which dimension m, and dimension d appears as follows, $S = (d, m, d, s_1, d, m, d, s_2, d, m, d, s_3, ...d, m, d, s_{2^{(n-2)}})$, where s_i depends on m and d, $i = 1,...2^{n-2}$, and $m \neq d$.

Proof:

Let DM = (dm_1, dm_2,..........dm_n) be a permutation of (1, 2, 3,....., n), in which dm_1 = d, and dm_2 = m. Then embedding sequence, S produced by the following procedure satisfies the given condition.

S <- (dm_1)
For i = 2 to n Do
    S <- S | (dm_i) | S;
Return S;

where the vertical bar denotes the concatenation [Cha90, Dek81, Pro88].

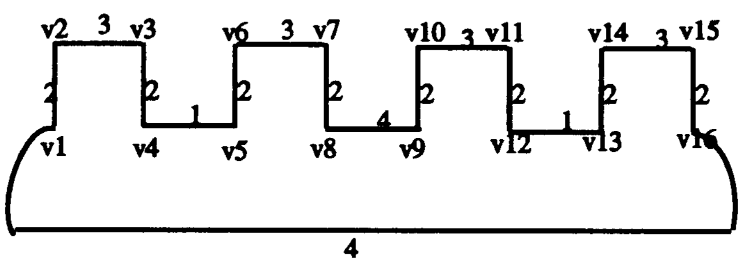


Figure 3.5 : The embedding sequence and the Hamiltonian cycle for n = 4

For example, let dm_1 = 2, dm_2 = 3, dm_3 = 1, and dm_4 = 4. Then DM = (2, 3, 1, 4). The embedding sequence for a 16-node cycle in a 4-cube resulting from DM = (2, 3, 1, 4) is S = (2, 3, 2, 1, 2, 3, 2, 4, 2, 3, 2, 1, 2, 3, 2, 4). The Hamiltonian cycle from the embedding sequence S is shown in Figure 3.5.

Theorem 3.3

A chordal ring with n nodes and w = n/2 (n > 6) can be embedded with dilation 1 into BH(m, k, 2) where $m*2^k \geq n \geq (m-1)*2^k$, and $3 \leq m < k + 2$.

Proof :

By assumption, we can represent n as $n = (m - 1)*2^k + 2p$, where p is an integer. Our claim is that we can find the structure given in Figure 3.3(b) of n nodes in

BH(m, k, 2). Let us assume that each edge from node $[L - 1, G_k][x]$ to node $[L, G_k][x]$ is said to be a straight upward edge, and that each edge from node $[L, G_k][y]$ to node $[L-1, G_k][y]$ a straight downward edge, where $0 \le x \le 2^k-1$, $0 \le y \le 2^k-1$ and $1 \le L \le m$.

case 1: p = 0 :

Start from node $[0, (G_{k-1})0][0]$, and go up to the node $[m - 2, (G_{k-1})0][0]$, traversing all the straight upward edges. From here, traverse the hypercube edge of dimension m-2, and arrive at node $[m - 2, (G_{k-1})0][x]$ for some x. Then, go down to the node $[0, (G_{k-1})0][x]$, traversing all the straight down edges. If we do the same procedures continously in the order as we do in Lemma 3.6, where DM $=(dm_1, dm_2, \ldots\ldots\ldots dm_k)$ is a permutation of (2, 3, 4...., k, k + 1), in which $dm_1 = k + 1$, and $dm_2 = m - 2$, we will find a cycle of length $(m - 1)2^{k-1}$. Let us call the above cycle as C_1. In a similar way, we can find another corresponding cycle C_2, such as $[0, (G_{k-1})1][0]$, $[1, (G_{k-1})1][0]$....... $[0, (G_{k-1})1][0]$.

Let e_1 be the edge between nodes $[0, (G_{k-1})0][0]$, and $[1, (G_{k-1})1][0]$, and e_2 be the edge between nodes $[1, (G_{k-1})0][0]$ and $[0, (G_{k-1})1][0]$. Let $C_1 =(V_1, E_1)$, and $S= \{ ([r, (G_{k-1})0][x], [r, (G_{k-1})1][x]) \mid [r, (G_{k-1})0][x] \in V_1 \}$. Then C_1, C_2, e_1, e_2, and S form a Figure 3.3(b) structure of n nodes.

case 2: $p = 2h$, $h \ge 1$ is integer.

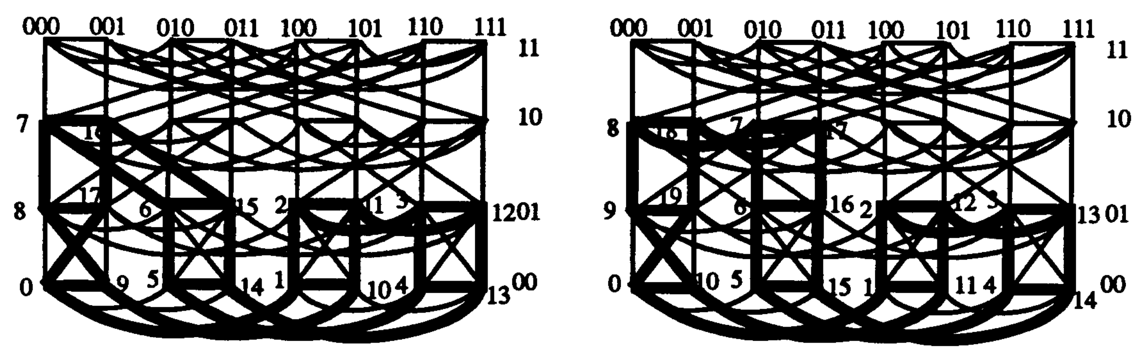
This is somewhat similar to case 1. Here, we create a cycle C_1' of length $(m - 1)*2^{k-1} + p$ by traversing straight upward edges up to level m - 1 in exactly h places. In other places, we need to traverse upward edges up to level m - 2. Whenever we arrive at $[m - 1, (G_{k-1})0][x]$, traversing straight upward edges, traverse the hypercube edge of dimension m-2, and go to the node $[m - 1, (G_{k-1})0][y]$ for some y .

From the node $[m - 1, (G_{k-1})0][y]$, go down to the node, $[0, (G_{k-1})0][y]$ by traversing all the straight downward edges. Another corresponding cycle, C_2' of length $(m - 1)*2^{k-1} + p$ can be constructed similarily.

Let $S' = \{([r, (G_{k-1})0][x], [r, (G_{k-1})1][x]) \mid [r, (G_{k-1})0][x] \in V_1' \}$, where $C_1' = (V_1', E_1')$. Then C_1', C_2', e_1, e_2, and S' form a Figure 3.3(b) structure of n nodes.

case 3: $p = 2h + 1$, $h > 0$ is integer.

Here, we create C_1'' by traversing straight upward edges up to level m - 1 in exactly h + 1 places. The difference between C_1' and C_1'' is that in C_1'', at exactly one place among the above h + 1 places, when we arrive at $[m - 1, (G_{k-1})0][x]$ by traversing straight upward edges, we traverse the banyan edge and go to the node $[m - 2, (G_{k-1})0][y]$, where $[m - 1, (G_{k-1})0][x]$ is adjacent to $[m - 1, (G_{k-1})0][y]$ along dimension m - 2.



(a) chordal ring with 18 nodes

(b) chordal ring with 20 nodes

Figure 3.6 : Embedding chordal rings

Figure 3.6 shows the embedding of chordal rings of length 18, and 20 on BH(4, 3, 2).

3.3.3 Embedding X-trees into a Banyan Hypercube

A L-level $(L \geq 1)$ X-tree is a complete binary tree of L levels with additional edges added to connect consecutive nodes on the same level of the tree [Des78]. An example of 4-level X-tree is shown in Figure 3.7. Since there are $(1 + 2 + 2^2 + \ldots\ldots + 2^{k-1}) = 2^k - 1$ nodes from level 1 to level k of L-level X-tree, $L \geq k$, we can label nodes at level $(h + 1)$ as $2^h, 2^h + 1, 2^h + 2, \ldots\ldots\ldots 2^{h+1} - 2, 2^{h+1} - 1$ in the order as they appear in the tree from left to right. The root of X-tree is then labeled as 1, and for each nonterminal node n of X-tree, its left and right children are labeled as 2n and 2n + 1.

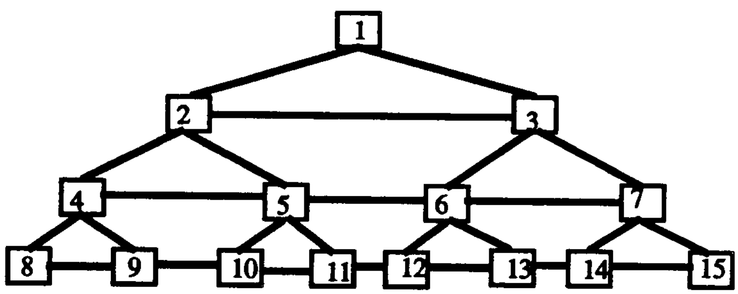


Figure 3.7 : 4-level X-tree

Lemma 3.7

In a L-level X-tree, there exists at least $2^{L-1} - 1$ edge disjoint cycles, each of which is of length 3.

Proof :

Let us prove by induction on L.

When L = 2, trivially there is $2^{2-1} - 1 = 1$ cycle of length 3.

Assume that there are $2^{L-2} - 1$ cycles with length 3 in (L - 1)-level X-tree.

In the L-level X-tree, there are 2^{L-2} nodes at level L-1, and all the nodes and edges up to level L - 1 in L-level X-tree form another (L - 1) level X-tree.

Here, each node at level L - 1 makes another one cycle of length 3 with the left and right child of it. Hence, there are at least $(2^{L-2} - 1) + 2^{L-2} = 2^{L-1} - 1$ of cycles with length 3 in a L-level X-tree.

Lemma 3.8

A L-level X-tree can't be embedded into BH(1, L, 2) with dilation 1.

Proof :

From Lemma 3.7, there are cycles of odd length in X-tree. Since cycle of odd length can't be embedded in BH(1,L,2) with dilation 1, a L-level X-tree can't be embedded into BH(1, L, 2) with dilation = 1.

Lemma 3.9

A L-level X-tree can't be embedded into BH(1, L, 2) with dilation 2 and $M_2 < 2^{L-1} - 1$.

(M_2 = number of edges, mapped with dilation 2)

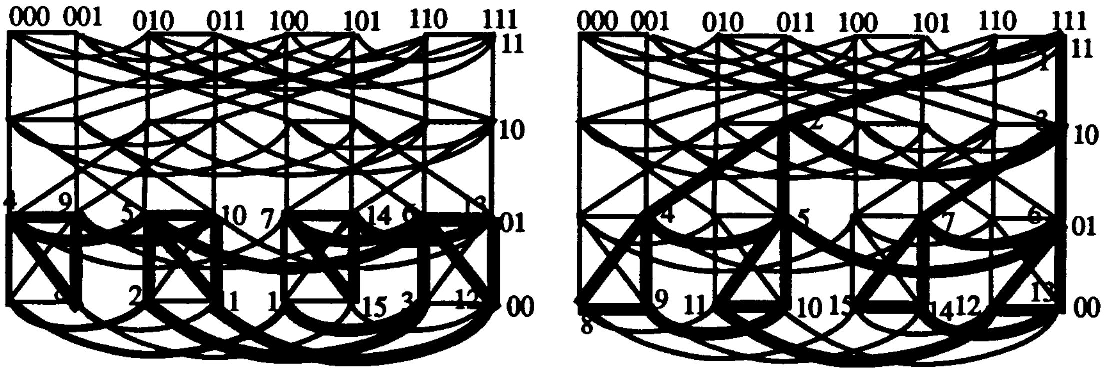
Proof :

Lemma 3.7 says that there are at least $2^{L-1} - 1$ edge-disjoint cycles of length 3.

Since cycle of length 3 can't be embedded with dilation 1 into BH(1, L, 2), each cycle of length 3 generates one pair of nodes, which are mapped with dilation 2.

Theorem 3.4 [Che90]

A L-level X-tree can be embedded into BH(1, L, 2) with dilation 2 and $M_2 = 2^{L-1}-1$.



(a) 4-level X-tree in BH(2, 3, 2) (b) 4-level X-tree in BH(4, 3, 2)

Figure 3.8 : Embedding X-tree into BH

Lemma 3.10

A L-level X-tree can't be embedded into BH(2, L - 1, 2) with dilation 2 and $M_2 < 2^{L-2} - 1$.

Proof :

Lemma 3.7 says that there are at least $2^{L-1} - 1$ edge-disjoint cycles of length 3 in L-level X-tree. But, it can be easily shown that there exist exactly 2^{L-2} edge-disjoint cycles of length 3 in BH(2, L-1, 2). Hence, $(2^{L-1} - 1) - 2^{L-2} = 2^{L-2} - 1$ cycles of length 3 must be mapped with dilation 2.

Theorem 3.5 A L-level X-tree can be embedded into BH(2, L - 1, 2) with dilation 2, expansion = $2^L/(2^L - 1)$, and $M_2 = 2^{L-2} - 1$.

Proof :

We prove by induction on L.

When $L = 2$, it is trivial. Actually in this case, $M_2 = 2^{L-2} - 1 = 0$. That is, dilation = 1. Assume that n-level X-tree can be embedded into BH(2, n - 1, 2) with dilation 2, expansion = $2^n/(2^n - 1)$, and $M_2 = 2^{n-2} - 1$ such that (0, 0.....0) is the idle node, and

the root node is at most of distance 1 from this idle node.

Any BH(2, n, 2) can be decomposed into $BH_1(2, n - 1, 2)$, and $BH_2(2, n - 1, 2)$, each of which is of BH(2, n - 1, 2). Let us assume that the most significiant bit of all the nodes in $BH_1(2, n - 1, 2)$ is 0, and the most significiant bit of all the nodes in $BH_2(2, n - 1, 2)$ is 1. By assumption, we can embed one n-level X-tree into $BH_1(2, n - 1, 2)$ with dilation 2, expansion = $2^n/(2^n - 1)$, and $M_2 = 2^{n-2} - 1$ such that (0, 00.....0) is the idle node and root node is at most of distance 1 from (0, 00....0).

Also, we can embed another n-level X-tree into $BH_2(2, n - 1, 2)$ with dilation 2, expansion = $2^n/(2^n - 1)$, and $M_2 = 2^{n-2} - 1$ such that (0, 10.....0) is the idle node and root node is at most of distance 1 from this idle node, (0, 10.....0). Make node (0, 10...0) the new root with the old roots as its children. Then, (0, 00...0) becomes the idle node, and the root node is of distance 1 from this idle node. Here in each step of the embedding, we can embed in such a way that all the consecutive nodes in the same level form a ring. Hence $M_2 = (2^{n-2} - 1) + (2^{n-2} - 1) + 1 = 2^{n-1} - 1$.

Lemma 3.11

A L-level X-tree can not be embedded into BH(h, h - 1, 2) with dilation 1, if $h < L$.

Proof :

Let us say that edges between the nodes in the same level as hypercube edges, and edges between the nodes in the adjacent level as banyan edges.

case 1: root node of L-level X-tree is mapped into some node in the top level of BH(h, h - 1,2).

Since $h < L$, there exists at least one node, $v = (m, x_{h-1} \ldots x_{m+1}\ x_m\ x_{m-1} \ldots\ x_1\ x_0)$ in X-tree, which is adjacent to its parent node along the banyan edge, and any of its

left or right child does not lie in the level lower than its level. Otherwise, we can easily prove that the X-tree of level L can't be embedded into BH(h, h - 1, 2) with dilation 1. Here, both the left and right children of v can't be in the same level of v. Otherwise, we will have a cycle of length 3 in the hypercube. Also left and right children of v can't be in level m + 1 simultaneously, because at most two nodes at level i + 1 are adjacent to a node at level i for each level, i = 0,..L - 1.

Hence, one of the two children of v must be in level m + 1 of BH(h, h - 1, 2), and the other child in level m. Without loss of generality, assume that the left child of v, lv is in level m + 1, such that $lv = (m + 1, x_{h-1} \ldots x_{m+1}\ x_m\ x_{m-1} \ldots\ x_1\ x_0)$. Then parent node, pv of v is $pv = (m + 1, x_{h-1} \ldots x_{m+1}\ cx_m\ x_{m-1} \ldots\ x_1\ x_0)$, and right child, rpv of pv is $rpv = (m, x_{h-1} \ldots x_{m+1}\ cx_m\ x_{m-1} \ldots\ x_1\ x_0)$, where cx_m is the complement of x_m. But right child, rv of v must also be $rv = (m + 1, x_{h-1} \ldots x_{m+1}\ cx_m\ x_{m-1} \ldots\ x_1\ x_0)$ or $(m, x_{h-1} \ldots x_{m+1}\ cx_m\ x_{m-1} \ldots\ x_1\ x_0)$. Otherwise, three nodes, v, rv and lv can't make a cycle of length three. Assume that $rv = (m + 1, x_{h-1} \ldots x_{m+1}\ cx_m\ x_{m-1} \ldots\ x_1\ x_0)$. This says $(m + 1, x_{h-1} \ldots x_{m+1}\ cx_m\ x_{m-1} \ldots\ x_1\ x_0)$ is the child of v and also the parent of v, which is impossible.

When $rv = (m, x_{h-1} \ldots x_{m+1}\ cx_m\ x_{m-1} \ldots\ x_1\ x_0)$, we can give a similar contradiction.

case 2: root node of L level X-tree is mapped into some node in level m, $0 < m < h$.

Proof :

Similar to the case 1.

case 3: root node of L level X-tree is mapped into some node in level 0 .

Proof :

Similar to the case 1.

Theorem 3.6 A L-level X-tree can be embedded into BH(L, L - 1, 2) with dilation 1.

Proof :

In X-tree, we can assign an integer label to each node in such a way that the root of X-tree is 1, and for each nonterminal node n of X-tree, its left and right children as 2n and 2n + 1.

Let f be a mapping from the labels in X-tree to the nodes in BH(L, L - 1, 2) as follows.

f(1) = (L - 1, 111......1111), that is, the rightmost node in the top level.

For each nonterminal node, n, where $f(n) = (m, x_{L-1}\ldots x_{m+1}\, x_m\, x_{m-1}\cdots\, x_1\, x_0)$,

$$f(2n) = (m - 1, x_{L-1}\cdots\, x_m\, cx_{m-1}\cdots\, x_1\, x_0),$$
$$f(2n+1) = (m - 1, x_{L-1}\cdots\, x_m\, x_{m-1}\cdots\, x_1\, x_0) \text{ if n is even.}$$

$$f(2n) = (m - 1, x_{L-1}\cdots\, x_m x_{m-1}\cdots\, x_1\, x_0)$$
$$f(2n+1) = (m - 1, x_{L-1}\cdots\, x_m cx_{m-1}\cdots\, x_1\, x_0) \text{ if n is odd.}$$

Then, it is easy to verify that f is an one-to-one mapping such that a given child and its parent in the X-tree are mapped into two adjacent nodes in BH(L, L - 1, 2).

It remains to show that any two consecutive nodes in the same level of the X-tree are adjacent in BH(L, L - 1, 2). We prove this by induction on label, n in X-tree. Trivially, f(2) and f(3) are adjacent.

Assume that f(k) and f(k + 1) are adjacent if $k \le p$, and they are in the same level of X-tree, and let us consider f(p + 1) and f(p + 2), where label p + 1 and p + 2 are in the same level. If p is odd, f(p + 1) and f(p + 2) are left and right children of f((p + 1)/2).

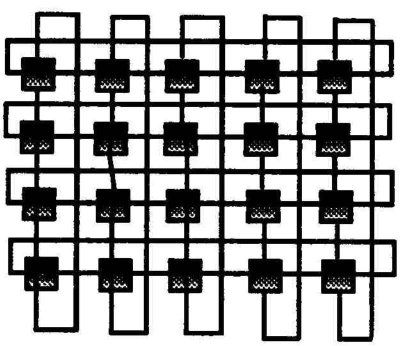
Hence, they are adjacent.

If p is even, it can be easily shown that label p/2 and p/2 + 1 are in the same level. Hence f(p/2) and f(p/2 +1) are adjacent by induction assumption.

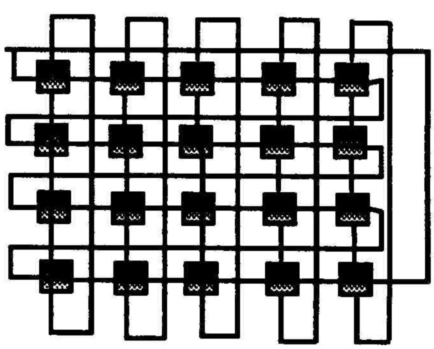
Also from the definition of the function, f, all the nodes in the same level of X-tree appear in the same level of BH(L, L - 1,2). Therefore, we can represent f(p/2) and f(p/2 + 1) as $f(p/2) = (r, x_{L-1}.....x_s... x_1 x_0)$, and $f(p/2 + 1) = (r, x_{L-1}.....cx_s... x_1 x_0)$ for some x_s, and r.

Here if p/2 is also even, then p/2 + 1 is odd. By the definition of the function , f, $f(2(p/2) + 1) = f(p + 1) = (r - 1, x_{L-1}.....x_s... x_1 x_0)$ and $f(2(p/2 + 1)) = f(p + 2) = (r - 1, x_{L-1}.....cx_s... x_1 x_0)$. Hence f(p + 1) and f(p + 2) are adjacent.

In the case of p/2 being odd, we can give a similar proof.



(a) Torus



(b) Illiac Network

Figure 3. 9 : (5, 5) torus and (5, 5) Illiac Network

3.3.4 Embedding a torus into a Banyan Hypercube.

A $(l_1, l_2,.........,l_{d-1}, l_d)$ d-dimensional mesh is a d-dimensional lattice of width l_i for dimension i, i = 1, 2,.....d., where l_i nodes in each dimension i, i = 1, 2,.....d., form a line of length l_i. The nodes are labelled as d-tuples $(i_1, i_2,.....,i_d)$, where $0 \leq i_j \leq l_j - 1$ for all $1 \leq j \leq d$. Two nodes are linked by an edge if they differ in precisely one coordinate and if the absolute value of the difference in that coordinate is 1. Torus is just a wrap-around mesh.

Lemma 3.12 [Ho87] If a $(l_1, l_2, \ldots\ldots\ldots, l_{d-1}, l_d)$ d-dimensional mesh is embedded in an n-cube with dilation one, then $n \geq \sum \lceil \log_2 l_i \rceil$, for i = 1, 2..., d.

Lemma 3.13 If a $(l_1, l_2, \ldots\ldots\ldots, l_{d-1}, l_d)$ d-dimensional torus is embedded in an n-cube with dilation one, then $n \geq \sum \lceil \log_2 l_i \rceil$, for i = 1, 2 ...., d.

Proof :

It follows from Lemma 3.12, and the definition of torus.

Theorem 3.7 A $(l_1, l_2, \ldots\ldots\ldots, l_{d-1}, l_d)$ d-dimensional torus can be embedded into BH(1, n, 2), where $n = \sum \lceil \log_2 l_i \rceil$, for i = 1, 2 ...., d.

(a) with dilation 1 if all l_i are even, for $1 \leq i \leq d$.

(b) otherwise, let l_{kj} for $1 \leq j \leq p$ be the given odd dimensions. Then the mapping can be done with dilation 2, and $M_2 = \sum D(l_{kj})$, for $1 \leq j \leq p$ where $D(l_{kj}) = (\prod l_i) / l_{kj}$, for i = 1,2 ....,d.

Proof :

(a) It follows from Lemma.3. 2, Lemma 3.3, and case 1 in Theorem 3.1.

(b) It follows from Lemma 3.2, Lemma 3.3 and the fact that a ring of odd length can be embedded with dilation 2, and $M_2 = 1$ into $\lceil \log_2 l \rceil$–cube.

Lemma 3.14 For r = r1 + r2, the BH(2, r, 2) can be expressed as the cross product

$$BH(2, r, 2) = BH(2, r1, 2) \otimes Q_{r2}$$

where Q_{r2} denotes the r2-dimensional hypercube .

Proof :

The proof is by induction on r2. When r2 = 1, the result is true by inspection. Assume that the above is true when r2 = n, and consider the case, when r2 = n+1.

That is, r = r1 + n + 1 = r' + 1, where r1 + n = r'. By induction assumption, BH(2, r', 2) = BH(2, r1, 2)$\otimes$ Q_n. But by inspection also, BH(2, r, 2) = BH(2, r', 2)$\otimes Q_1$ = BH(2, r1, 2)$\otimes$ $Q_n \otimes Q_1$ = BH(2, r1, 2)$\otimes$ Q_{n+1} (by Lemma 3.2). Hence the above is also true when r2 = n +1.

Lemma 3.15 For any k $\geq$ 1 and r = r1 + r2 +.....+ rk , the banyan-hypercube BH(2, r, 2) can be expressed as the cross product

$$BH(2, r, 2) = BH(2, r1, 2) \otimes Q_{r2} \otimes \ldots\ldots\ldots \otimes Q_{rk}$$

where Q_{ri} denotes the ri-dimensional hypercube for $2 \leq i \leq k$.

Proof :

It follows by Lemma 3.2, and Lemma 3.14.

Theorem 3.8 A $(l_1, l_2, \ldots\ldots\ldots, l_{d-1}, l_d)$ d-dimensional torus can be embedded into BH(2, n, 2), where $n + 1 = \sum \lceil \log_2 l_i \rceil$, for i = 1,2 ....,d.

(a) with dilation 1 if either all l_i, $1 \leq i \leq d$, is even; or only one l_i odd and the others even .

(b) otherwise, let l_{kj} for $1 \leq j \leq p$ be the given odd dimensions. Then the mapping can be done with dilation 2, and $M_2 = \sum D(l_{kj})$, for $1 \leq j \leq p$ where $D(l_{kj}) = (\prod l_i) / l_{kj}$, for i = 1,2 ....,d.

Proof :

(a) By Lemma 3.15, Lemma 3.2, Lemma 3.3, and Theorem 3.1.

(b) It follows from Lemma 3.2, Lemma 3.3, case 3 in Theorem 3.1 and the fact that a ring of odd length p can be embedded with dilation 2, and $M_2 = 1$ into $\lceil \log_2 p \rceil$–cube.

Lemma 3.16 $C_{2h} \otimes Q_{n-1}$ is a subgraph of BH(h, n, 2), where $h \leq n + 1$, and C_{2h} is a cycle of length 2h.

Proof :

It follows from Lemma 3.2, the fact that $L_h \otimes Q_n$ is a subgraph of BH(h, n, 2), and that $L_h \otimes Q_1$ is C_{2h}.

Theorem 3.9 A $(2h, l_2, \ldots, l_{d-1}, l_d)$ d-dimensional torus can be embedded into BH(h, n, 2), where $n - 1 \geq \sum \lceil \log_2 l_i \rceil$, for i = 2,...d.

(a) with dilation 1 if for all $2 \leq i \leq d$, l_i is even.

(b) otherwise, let l_{kj} for $1 \leq j \leq p$ be the given odd dimensions. Then the mapping can be done with dilation 2, and $M_2 = 2h \sum D(l_{kj})$, for $1 \leq j \leq p$ where $D(l_{kj}) = (\prod l_i) / l_{kj}$, for i = 1,2 ....,d.

Proof :

(a) It follows from (a) in Theorem 3.7, and Lemma 3.16.

(b) It follows from Lemma 3.2, Lemma 3.3, case 3 in Theorem 3.1 and the fact that a ring of odd length p can be embedded with dilation 2, and $M_2 = 1$ into $\lceil \log_2 p \rceil$–cube.

3.3.5 Embedding an Illiac network into a Banyan Hypercube.

An Illiac network is a (l_1, l_2) 2-dimensional grid, in which each column is a ring of length l_1, and each row is a linear array such that last node of each row is connected to the first node of its next row, where the next row of row i is defined as $(i + 1) \bmod l_1$, $i = 1, 2, \ldots, l_1$ [Bar20].

Theorem 3.10 A (l_1, l_2) Illiac network can be embedded into BH(2, (k + l), 2)

(a) with dilation 2, expansion = $2 * 2^{(k+l)}/(l_1 * l_2)$, and $M_2 = l_1$ if l_1 or l_2 is even, where $2^k < l_1 \leq 2^{k+1}$, and $2^{l-1} < l_2 \leq 2^l$.

(b) with dilation 2, and $M_2 = l_1 + l_2$ if l_1 and l_2 are odd.

Proof :

(a) Without loss of generality, assume that l_2 is even.

(1) By Theorem 3.8, torus (l_1, l_2) can be embedded into BH(2, (k + l), 2) with dilation one.

(2) A (l_1, l_2) Illiac network can be embedded into torus (l_1, l_2) with dilation 2, and $M_2 = l_1$.

(b) It follows from Theorem 3.8 and the fact that A (l_1, l_2) Illiac network can be embedded into torus (l_1, l_2) with dilation 2, and $M_2 = l_1$.

Theorem 3.11 A (l_1, l_2) Illiac network can be embedded into BH (h, n, 2)

(a) with dilation 2, expansion = $h * 2^n/(l_1 * l_2)$, and $M_2 = l_1$ if l_2 is even, where $l_1 = 2h$ or $2h - 1$, and $\lceil \log_2 l_2 \rceil = n - 1$.

(b) with dilation 2, and $M_2 = l_1 + l_2$ if l_1 and l_2 are odd.

Proof :

(a) It follows from (a) in Theorem 3.9 and the fact that a (l_1, l_2) Illiac network can be embedded into torus (l_1, l_2) with dilation 2, and $M_2 = l_1$.

(b) It follows from (b) in Theorem 3.9 and the fact that a (l_1, l_2) Illiac network can be embedded into torus (l_1, l_2) with dilation 2, and $M_2 = l_1$.

Chapter 4

Topological Properties of K-ary n-cube.

4.1 Introduction

The hypercube topology for interconnection networks has been studied extensively by many authors, because it has many interesting features such as node symmetry , edge symmetry, logarithmic diameter, recursively definable structure, and maximal fault tolerance. In addition, many interesting topologies such as tree[Wu85, Bha85, Bha86, Wag89, Lei89,Aie90, Bha91], mesh[Ale82, Cha88, Cha91, Ho87, Ho90, Lie91, Mel90], mesh of tree[Efa91], ring[Saa88, Pro88, Cha91], and butterfly-like graph[Gre90] topologies can be embedded efficiently.

However, the hypercube has the major drawback that the number of connections to each processor grows logarithmically with the size of the network. While this is not a problem for small hypercubes, it can present some difficulties for very large machines[Lei91].

The k-ary n-cube has many of the desirable properties of the hypercube. In addition, the number of connections is independent of the size of the network. Furthermore, it has the mesh topology as a subgraph. Hence many linear algebra computations, or partial differential equations can be performed effectively on the k-ary n-cube based topology. For this reason, several parallel machines based on the k-ary n-cube topology have appeared, including the Ametek 2010[Sei188], the J-machine[Dal89], [Dal91], and the Mosaic [Sei288].

The k-ary n-cube parallel machine consists of k^n identical processors, each provided with its own sizable memory, and connected to 2n neighbors.

The purpose of this chapter is to study the topological properties of the k-ary n-cube. The rest of the chapter is organized as follows. Section 4.2 introduces the necessary definitions and notation. In Section 4.3, we will derive some simple graph properties of the k-ary n-cube.

4.2 Preliminaries

In this section, we give some definitions useful for the subsequent sections.

In the k-ary n-cube, any k-ary p-cube, $p < n$ is denoted by (k + 1)-ary strings in $\{0, 1, 2, \ldots, (k-1), *\}^n$, where * is the don't care symbol, which can be a 0, 1, 2,...(k - 2), or (k - 1). For example, 3-ary 2-cube formed by nodes, 000, 001, 002, 010, 011, 012, 020, 021, and 022 in the 3-ary 3-cube is denoted by 0**.

Any edge joining two nodes that differ in the ith bit position in the k-ary n-cube, where $0 \le i \le n - 1$ is said to be of dimension i. For example, (000, 001), and (001, 002) are edges of dimension 0 in 3-ary 3-cube.

A nonempty set of elements G with a binary operation *, is called a group if the following conditions are valid.[Her75]

1. $a, b \in G$ implies $a * b \in G$ (closed).
2. $a, b, c \in G$ implies that $a * (b * c) = (a * b) * c$ (associative law).
3. There exists an element $e \in G$ such that $a * e = e * a = a$ for all $a \in G$

   (the existence of an identity element in G).
4. For every $a \in G$ there exists an element $a^{-1} \in G$ such that $a * a^{-1} = a^{-1} * a = e$

   (the existence of inverses in G).

For example, let G consist of the integers where * denotes addition, i.e, a * b is a + b for a, b $\in$ G. Then G is a group in which 0 plays the role of e and -a that of a^{-1}. When the number of elements in the group G is finite, we say that G is a finite group. For example, let G ={1, - 1} and let * denote multiplication. Then G is then a group with 2 elements.

A group action graph (GAG) is defined by a set V of vertices and a set Π of permutations of V: For each v $\in$ V and each $\pi \in \Pi$, there is an edge labeled π from vertex v to vertex $v\pi$. A Cayley graph is a GAG(V, Π), where V is the group Gr(Π) generated by Π and where $\pi \in \Pi$ acts on each g $\in$ Gr(Π) by right multiplication [Ann90].

4.3 Topological Properties

Lemma 4.1 : The k-ary n-cube is a Cayley graph.

Proof:

It can be easily shown that Cayley graph generated by n vectors, (1, 0, 0.....0), (0, 1, 0,....0), (0, 0, 1,...., 0),........, (0, 0,....,1) with mod k bitwise addition is isomorphic to the k-ary n-cube.

For example, the Cayley graph, generated by (1, 0), and (0, 1) with mod 4 addition becomes the 4-ary 2-cube.

An automorphism of a graph is a one-to-one mapping of the nodes to the nodes such that edges are mapped to edges. A graph is said to be node-symmetric if for every pair of nodes, u and v, there exists an automorphism of the graph that maps a into b. A graph is said to be edge-symmetric if for every pair of edges, e_1 and e_2, there exists an automorphism of the graph that maps e_1 into e_2.

The symmetric interconnection networks have the interesting property that the network viewed from any node of the network looks the same. For this reason, identical processors in each node can use identical routing algorithms. Furthermore, congestion problems in this kind of network can be minimized if the load in the network is distributed uniformly through all the nodes.

Lemma 4.2 : K-ary n-cube is node-symmetric.

Proof : The proof follows from Lemma 4.1 and the fact that all Cayley graphs are node symmetric[Ake 81] .

Alternatively, we can show explicitly that given any two nodes in the k-ary n-cube, there exists an automorphism of the graph that maps one vertex into the other. Let u and v be the two nodes in the k-ary n-cube. Let w = u - v, where - is the bitwise mod k subtraction operation. Then $\rho(a) = a + w$, where - is the bitwise mod k subtraction operation, is a one to one mapping from vertices to vertices. Clearly, this maps u to v. Further, ρ is an automorphism . For, if two nodes x and y are connected by an edge, i. e, the Lee distance between them is 1, then Lee distance between $\rho(x)$ and $\rho(y)$ is also 1. Hence $\rho(x)$ and $\rho(y)$ are connected by an edge.

A Cayley graph is said to be hierarchical, if its generators can be ordered g_1, g_2, ..., g_d, such that for each i, $1 \le i \le d$, g_i is not in the subgroup generated by the first i - 1 generators. If the Cayley graph is hierarchical under any ordering of the set of generators, it is said to be strongly hierarchical[Ake87].). A strongly hierarchical graph has the property that it can be recursively decomposed using the generators in any order[Ake87].

Another important measure of interconnection networks is their fault tolerance. The fault tolerance of a graph is better defined through the graph theoretic property, called connectivity. The connectivity of a graph is the minimum number of vertices that need to be removed to disconnect the graph. The fault tolerance is then one less than the connectivity and indicates the maximum number of vertices that can be removed and still have the graph remain connected. Clearly, any graph can be disconnected by removing all the vertices adjacent to a given vertex. Thus its connectivity can be at most its degree. It has been shown that hierarchical Cayley graphs are maximally fault tolerant[Ake84]. That is, their fault tolerance is exactly one less than their degree.

Lemma 4.3 : The k-ary n-cube is strongly hierarchical.

Proof :

Let us take any arbitrary (n - 1) vectors, g_1, g_1, ..., $g_{(n-1)}$ from the n vectors, (1, 0, 0.....0), (0, 1, 0,....0), (0, 0, 1,...., 0),........, (0, 0,....,1). Let us think about the subgroup G generated by g_1, g_1, ..., $g_{(n-1)}$ with mod k bitwise addition operation. Then the remaining vector is not in G.

For example, Let us think about the group, G = {(1, 0, 0), (1, 1, 0), (1, 2, 0), (2, 0, 0), (2, 1, 0), (2, 2, 0), (0, 0, 0), (0, 1, 0), (0, 2, 0,) } generated by (1, 0, 0), and (0, 1, 0) with mod 3 bitwise plus operation. Then, (0, 0, 1) $\notin$ G.

Lemma 4.4 : The k-ary n-cube is maximally fault tolerant.

Proof : The proof follows from Lemma 4.3 and the fault tolerance of the strongly hierarchical Cayley graphs.

We can also prove Lemma 4. 4 using induction on n.

When n = 1, i.e, in the case of ring with length k, 2 node faults are needed for the network to be disconnected. Let us assume that 2r node faults are needed for the network to be disconnected, when n = r. Suppose that n = r + 1, and that 2r + 1 node faults are present. Then we can decompose the k-ary (r + 1)-cube along some dimension, say, 1 into k k-ary r-cubes in such a way that in each k-ary r-cubes, at most 2r -1 node faults exist. By the induction assumption, each k-ary r-cube 0***..*, 1***..*,....(k -1)***..* is the connected graph. $k^r > 2\ (2r + 1)$ implies that there exists two nodes, 0A in 0***..*, and 1A in 1***..* , which are nonfaulty. Hence connected graphs in 0***..*, and 1***..* are connected by the edge (0A, 1A). In a similar way, between any two k-ary r-cubes, there exist two nonfaulty corresponding nodes, such that connected graphs in those two k-ary r-cubes are connected by the edge between those two nodes.

Since the degree of k-ary n-cube is 2n, the k-ary n-cube is connected even in the presence of 2n - 1 node faults.

Lemma 4.5 : The k-ary n-cube can be recursively decomposed along any dimension.

Proof:

The proof follows from Lemma 4.3 and the recursive structure of the strongly hierarchical Cayley graphs.

By the property of the recursive decomposibility, any k-ary (n + 1)-cube can be decomposed into k k-ary n-cubes along any dimension. For example, the k-ary (n + 1)-cube can be decomposed along dimension 0 into k k-ary n-cubes, **..*0, **..*1, **..*2, ..., and **..*(k - 1).

Also, any k-ary n-cube can be constructed recursively as follows.

1) A k-ary 1-cube is a ring with k nodes;

2) A k-ary n-cube consists of k k-ary (n - 1)-cubes, each one having k^{n-1} nodes.

Further, the corresponding nodes in each k-ary (n - 1)-cube form a ring of length k. That is, k-ary n-cube, $G_{k,\, n}$ is the cross product of a k-ary (n - 1)-cube, $G_{k,\, (n-1)}$ and a ring of length k, C_k.

Notationally, $G_{k,\, n} = G_{k,\, (n-1)} \otimes C_k$.

Where $V(G_{k,\, n}) = \{\, uv \mid u \in V(C_k) \text{ and } v \in V(G_{k,\, (n-1)}) \,\}$ and

$$E(G_{k,\, n}) = \{\, (u_1v_1,\ u_2v_2) \mid (\, (u_1, u_2) \in E(C_k) \text{ and } v_1 = v_2 \,) \text{ or } (\, u_1 = u_2 \text{ and } (v_1, v_2) \in E(G_{k,\, (n-1)}) \,) \,\}$$

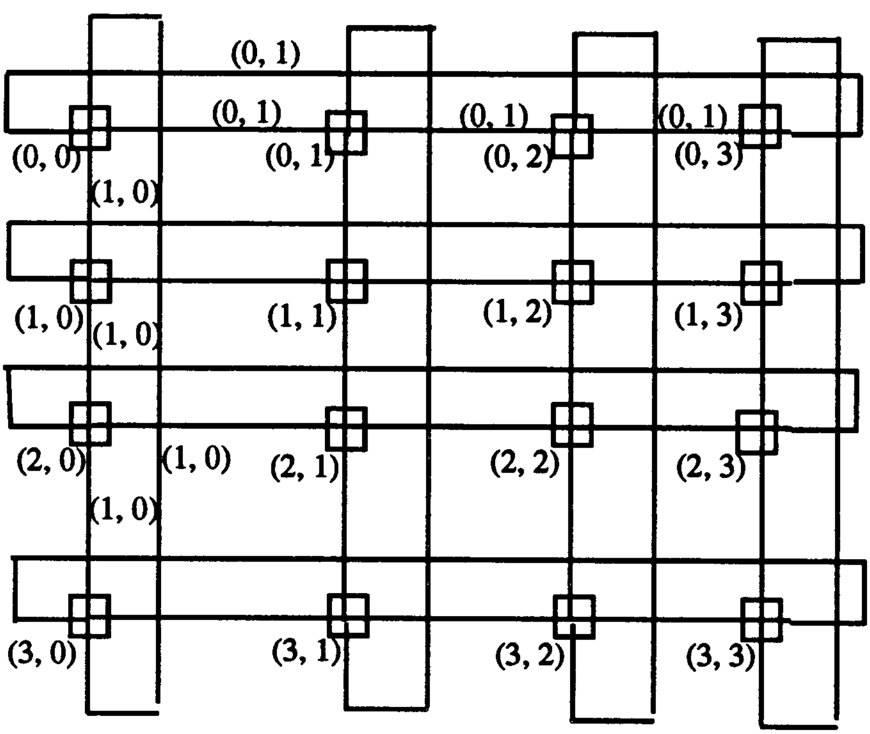


Figure 4.1 : 4-ary 2-cube as a Cayley graph.

If vertices u and v are connected in graph G, the distance between u and v in G, denoted by d(u, v) is the length of a shortest path from u to v in G. The diameter of G is the maximum distance between two vertices of G. Let us assume D(G) and $D_a(G)$ denote the diameter and average distance of the graph G respectively. When G is a product graph of two graphs, G_1, and G_2, i. e, $G = G_1 \otimes G_2$, then $D(G) = D(G_1) + D(G_2)$, and $D_a(G) = D_a(G_1) + D_a(G_2)$ [You3 90].

Lemma 4.6 : The diameter of the k-ary n-cube is $\lfloor k/2 \rfloor * n$.

Proof: Let u and v be two nodes in the k-ary n-cube. Then from the definition of the Lee distance, we have

$$D_L(u, v) = W_L(u - v) = \Sigma \mid a_i \mid$$

However, $$\mid a_i \mid \leq \lfloor k/2 \rfloor$$

This results in $$D_L(u, v) \leq \lfloor k/2 \rfloor * n$$

We can also prove Lemma 4. 6 by induction on n.

When n = 1, i. e, in the case of cycle with length k, the diameter is $\lfloor k/2 \rfloor$.

Assume that the diameter of the k-ary p-cube is $\lfloor k/2 \rfloor * p$, where $p \leq n - 1$.

By the property of the product graph, the diameter of the k-ary (p + 1)-cube = the diameter of the k-ary p-cube + the diameter of the k-ary 1-cube, i. e, $\lfloor k/2 \rfloor * p + \lfloor k/2 \rfloor = \lfloor k/2 \rfloor * (p + 1)$.

Lemma 4.7: The average distance of the k-ary n-cube is n*adist, where adist is the average distance in the k-ary 1-cube.

Proof: It follows from the following facts, using induction on n.

1) : The k-ary n-cube is a product graph.

2): If graph G is the product graph of G_1 and G_2, then $D_a(G) = D_a(G_1) + D_a(G_2)$.

Lemma 4.8[Bos 93]:

Let $u = u_n u_{n-1} \dots u_1$, $v = v_n v_{n-1} \dots v_1$.

Let $l = D_L(u, v)$, $h = D_H(u, v)$ and $w_i = D_L(u_i, v_i)$. Then, in a k-ary n-cube, there are a total of 2n node-disjoint parallel paths between u and v such that

1) h paths are of length l,

2) 2(n - h) paths are of length l + 2, and

3) for each i, $w_i > 0$, there is a path of length $l + k - 2w_i$ (a total of h paths)

Lemma 4.9 : There exists ${}_nC_r * k^{n-r}$ k-ary r-cubes in the k-ary n-cube.

Proof:

Let us take r positions from n bits, and put the "don't care" symbols in those positions. There exists ${}_nC_r$ cases of selecting r positions from n bits. For each case, we can fix the remaining n - r positions in k^{n-r}.

Lemma 4.10: There exists k^{n-r} node-disjoint k-ary r-cubes in the k-ary n-cube.

Proof:

Let p be the maximal number of node-disjoint k-ary r-cubes in the k-ary n-cube.

Then p * (number of nodes in the k-ary r-cube) = p $*k^r \leq n *k^n$.

Hence $p \leq k^{n-r}$.....(1).

Let S be the set of k-ary r-cubes in the k-ary n-cube such as S = {**..*$a_{r+1} a_{r+2} \ldots a_n$| * is the don't care symbol, and $a_i \in$ [0, 1, 2,..., k - 1] for all i = r+1,...., n}. Then all the k-ary r-cubes in S are mutually node-disjoint.

Also, the number of elements in S is k^{n-r}.....(2). $p = k^{n-r}$ follows from (1) and (2).

Lemma 4.11: There exists $\lfloor n/r \rfloor *k^{n-r}$ edge-disjoint k-ary r-cubes in the k-ary n-cube.

Proof:

Let N be the maximal number of edges-disjoint k-ary r-cubes in the k-ary n-cube. Then N * (number of edges in the k-ary r-cube) = N * r $*k^r \leq n *k^n$. Hence $N \leq \lfloor n/r\, k^{n-r} \rfloor$.....(1).

Let us denote the k-ary n-cube with 0, 1,... (k - 1) and * (don't care symbol). For example, 0** with 9 nodes 000, 001, 002, 010, 011, 012, 020, 021, and 022 is a 3-ary 2-cube.

Let us say that two k-ary r-cubes overlap completely, if all the bit positions of don't care symbols in the two k-ary r-cubes are same. For example, **00, and **01 overlap completely. We can easily verify that any two k-ary r-cubes that overlap completely, are edge-disjoint.

Let us say that two k-ary r-cubes do not overlap if all the bit positions of don't care symbols in the two k-ary r-cubes are different. For example, **11, and 00** do not overlap. We can easily verify that any two k-ary r-cubes not overlapping, are edge-disjoint.

Let us say that two k-ary r-cubes overlap partially if overlapping here is neither "completely" nor "not overlap", and if constants in the same bit positions are equal. For example, **11, and 0**1 overlap partially in the 4-cube. We can easily verify that any two k-ary r-cubes that overlap partially, are not edge-disjoint.

Let S be the set of maximal k-ary r-cubes in the k-ary n-cube such that any pair of k-ary r-cubes do not overlap. Trivially, $|S| = \lfloor r/s \rfloor$. For a given k-ary r-cube, the number of k-ary r-cubes that overlap completely with that k-ary r-cube is 2^{r-s}. Let T be the edge-disjoint k-ary r-cubes in the k-ary n-cube obtained by the previous procedure. $|T| = \lfloor r/s \rfloor * 2^{r-s}$. Let U be the set of all the k-ary r-cubes in the k-ary n-cube. Let us take some arbitrary set of edge-disjoint k-ary r-cubes from U - T, and call it V. Then for any $u \in V$, there exists, a unique $u' \in T$ such that u and u' are partially overlapped, i.e, u and u' are not edge disjoint. We can check this fact!. Hence T is the maximal edge-disjoint k-ary r-cubes in the k-ary n-cube.

A graph G = (V, E) is bipartite if its vertex set V can be partitioned into two subsets V_1 and V_2 such that each edge of E connects a vertex in V_1 with a vertex in V_2.

Lemma 4.12: When k is even, the k-ary n-cube is a bipartite graph.

Proof:

Let us say that a k-ary n-cube node is even if the sum of all the n digits in its k-ary address is even, and that it is odd otherwise. In the case of k being even, any edge in the k-ary n-cube connects an even node with an odd node.

For example, when k = 4, and n = 2, the edges connect nodes in {(0, 0), (0, 2), (1, 1), (1, 3), (2, 0), (2, 2), (3, 1), (3, 3)} to nodes in {(0, 1), (0, 3), (1, 0), (1, 2), (2, 1), (2, 3), (3, 0), (3, 2)}.

Lemma 4.13 : No cycles of odd length exist as a subgraph of the k-ary n-cube when k is even.

Proof:

When k is even, any cycle in the k-ary n-cube will have to alternate between visiting even nodes and odd nodes. Hence, a cycle of odd length can't be a subgraph of the k-ary n-cube, when k is even.

Lemma 4.14 : There exists n edge-disjoint Hamiltonian cycles in the k-ary n-cube.

Proof:

A k-ary n-cube can be represented as the cross product of n cycles, where the length of each cycle is k. Hence n edge-disjoint Hamiltonian cycles are in the k-ary n-cube[Als90].

The degree of each node in the k-ary n-cube is 2n. Since there are k^n nodes, and each edge contributes 2 degrees in the graph, the total number of edges in the k-ary n-cube is $2n * k^n/2 = n * k^n$. But in each Hamiltonian cycle of k-ary n-cube, k^n edges are included. That means $n * k^n$ different edges are included in the n edge-disjoint Hamiltonian cycles. Hence every edge in the k-ary n-cube is included in one of the n edge-disjoint Hamiltonian cycles.

Chapter 5

Embedding Rings into K-ary n-cube.

5.1 Introduction

The k-ary n-cube graph has been successfully used in the design of several concurrent computers including the Ametek 2010 [Sei1 88], the J-machine [Dal89], [Dal91], and the Mosaic [Sei2 88]. Furthermore, it has many interesting topological properties as an interconnection network for a parallel machine. For example, it is node-symmetric, recursively decomposable, and maximally fault-tolerant [Ann90].

Embedding of rings into a given parallel machine has been studied by many authors[Saa88, Pro88, Cha2 91, Yan, Wan]. In this chapter, we show that a ring with n nodes, $k^{p-1} < n \leq k^p$ can be embedded into a k-ary p-cube with dilation 1 between any two adjacent nodes if k is odd. We also show that in the case of k being even, a ring with n nodes, $k^{p-1} < n \leq k^p$ can be embedded into a k-ary p-cube with dilation 2 if n is odd, and with dilation 1 if n is even.

The rest of the chapter is organized as follows. Section 5.2 introduces the necessary definitions and notation. Embedding of a ring network into a k-ary n-cube will be discussed in Section 5.3.

5.2 Preliminaries

In this section, we give some definitions useful for the subsequent sections.

Let us denote the nodes and edges of the graph G as V(G), and E(G) respectively.

G - e denotes the graph obtained after deleting the edge e from the graph G, for $e \in E(G)$. Similarily, G + e denotes the graph obtained after adding the edge e = (u, v) to the graph G, for $u, v \in V(G)$. The union of two graphs G_1 and G_2, denoted by $G_1 \cup G_2$ is the graph, formed by the set of nodes $V(G_1) \cup V(G_2)$ and edges $E(G_1) \cup E(G_2)$ such that $e \in E(G_1 \cup G_2)$ if $e \in E(G_1)$ or $e \in E(G_2)$.

In the k-ary n-cube, any k-ary p-cube, p < n is denoted by (k + 1)-ary strings in $\{0, 1, 2, \ldots, (k-1), *\}^n$, where * is the don't care symbol, which can be a 0, 1, 2,...(k - 2), or (k - 1). For example, 3-ary 2-cube formed by nodes, 000, 001, 002, 010, 011, 012, 020, 021, and 022 in the 3-ary 3-cube is denoted by 0**.

Any edge joining two nodes that differ in the ith bit position in the k-ary n-cube, where $0 \le i \le n - 1$ is said to be of dimension i. For example, (000, 001), and (001, 002) are edges of dimension 0 in 3-ary 3-cube.

Let G be a subgraph of k-ary r-cube, induced by some set of nodes, $V = \{v_1, v_2, \ldots v_p\}$. xG, for some $x \in \{0, 1, \ldots, k-1\}$, denotes a subgraph of the k-ary (r + 1)-cube, induced by nodes, $V_1 = \{xv_1, xv_2, \ldots, xv_p\}$ such that two nodes xv_i and xv_j, for $i, j \in \{1, \ldots, p\}$ are adjacent in the graph xG if v_i and v_j are adjacent in the graph G. In a similar way, if Y is a node in the k-ary s-cube, YG represents another subgraph of the k-ary (r + s)-cube.

For example, Let Y be 01, and G be the graph with V(G) = {000, 001, 020, 010}, and E(G) = {(000, 001), (000, 010), (020, 010)}, which is a subgraph of the 3-ary 3-cube. Then YG is the graph with V(YG) = {01000, 01001, 01020, 01010}, and E(YG) = {(01000, 01001), (01000, 01010), (01020, 01010)}, which is a subgraph of the 3-ary 5-cube.

Two edges e_1 and e_2 between two cycles R and S with length r and s respectively are said to be connecting edges, if there exists a cycle of length r + s in the graph $R \cup S + e_1 + e_2$. The basic idea of the proofs in this chapter uses the concept of connecting edges.

By the property of recursive decomposibility, any k-ary n-cube can be constructed recursively as follows.

1) A k-ary 1-cube is a ring with k nodes;

2) A k-ary n-cube consists of k k-ary (n - 1)-cubes, each one having k^{n-1} nodes. Further, the corresponding nodes in each k-ary (n - 1)-cube form the ring of length k, respectively.

By the property of the recursive decomposibility, any k-ary (n + 1)-cube can be decomposed into k k-ary n-cubes along any dimension. For example, along the dimension 0, k-ary (n + 1)- cube can be decomposed into k k-ary n-cubes, **..*0, **..*1, **..*2, ..., and **..*(k - 1).

5.3. Ring Embeddings

The existence of a Hamiltonian cycle in the k-ary n-cube, using a mapping function is shown in [Ma87]. The existence of a Hamiltonian cycle has also been shown, using the gray code in the k-ary n-cube[Bos 93]. In Lemma 5.1, and Lemma 5.2, the existence of a Hamiltonian cycle between any two adjacent nodes in the k-ary n-cube is proven, using induction.

Lemma 5.1 : In the k-ary 2-cube, there exists a Hamiltonian cycle between any two adjacent nodes.

Proof :

By the property of node symmetry, it is sufficient to show that there exists a Hamiltonian cycle between nodes, A = 01, and B = 02 in the k-ary 2-cube, G_{k2}.
Let C_i be the cycle formed by the sequence of nodes, i0, i1,....., i(k - 2), i(k - 1), and i0 in G_{k2}, i = 0.., k - 1. Then C_i, i = 0.., k - 1 is the subgraph of G_{k2}.
Also, for any v, $v \in V(G_{k2})$, there exists some i, $i \in \{ 0.., k - 1\}$ such that $v \in V(C_i)$. Hence, it is sufficient to prove that there exists a cycle, H_p, p = 0,...., k -1, which traverse all the nodes in C_j, j = 0..,p, such that H_p starts at node 01 and ends at node 02.

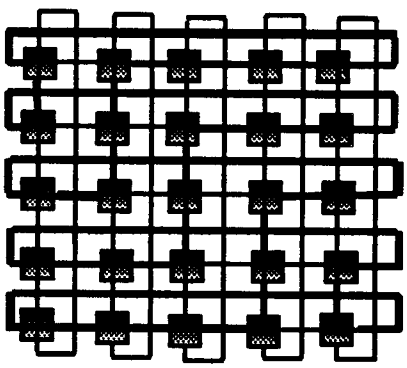


Figure 5.1: Hamiltonian cycle in 5-ary 2-cube

Trivially, $H_0 = C_0$, and H_1 is $H_0 \cup C_1$ - (00, 01) - (10, 11) + (00, 10) + (01, 11).

Let us assume that there exists a cycle, H_r, $1 \leq r < k - 1$, which traverse all the nodes in C_j, j = 0.., r starting at node 01 and ending at node 02.

Then $H_{r+1} = H_r \cup C_{r+1}$ - (rr, r(r + 1)) - ((r + 1)r, (r + 1)(r + 1)) + (rr, (r + 1)r) + (r(r + 1), (r + 1)(r + 1)) is a cycle, traversing all the nodes in C_m, m = 0.., r, r + 1 .

Here, H_{r+1} begins at node 01 and ends at node 02.

Lemma 5.2 : In the k-ary n-cube, there exists a Hamiltonian cycle between any two adjacent nodes.

Proof :

Again, the lemma is proven by induction.
When n = 1, it is trivial. When n = 2, it follows from Lemma 5.1.

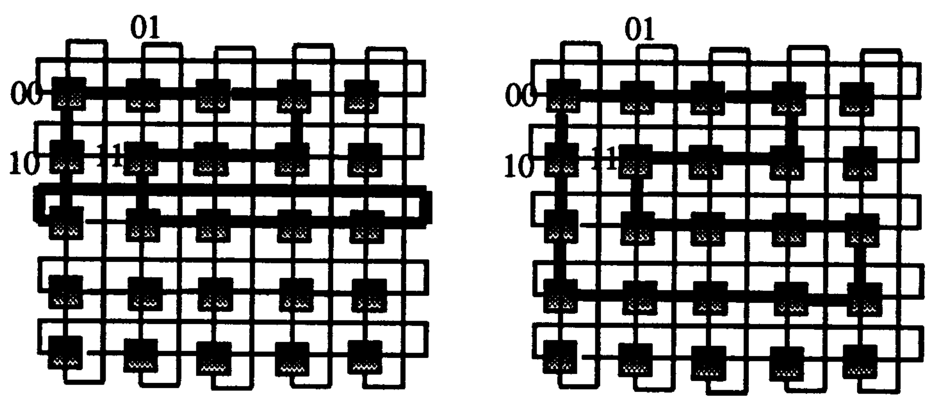
Let us assume that there exists a Hamiltonian cycle between any two adjacent nodes in the k-ary r-cube, where $r \geq 2$. As in the proof of Lemma 5.1, it will be shown that there exists a Hamiltonian cycle between two nodes, A = 00.....01, and B = 00 ......, 02 in k-ary (r + 1)-cube.

Let us partition the k-ary (r + 1)-cube in terms of r-th bit, and make k k-ary r-cubes, 0**....*, 1**..*,.....(k -2)**..*, and (k - 1)**..*.
By the induction assumption, there exists a Hamiltonian cycle C_0 between nodes, 00.....1, and 00...2 in the k-ary r-cube, 0**..* . Let us take two adjacent nodes, 0α and 0β, in C_0, where α, and β are r-bit of k-ary vectors, and (00...1, 00...2) $\neq$ (0α, 0β). By the induction assumption, there exists a Hamiltonian cycle C_1, between nodes 1α, and 1β in the k-ary r-cube, 1**..* . Then $H_1 = C_0 \cup C_1 - (0\alpha, 0\beta) - (1\alpha, 1\beta) + (0\alpha, 1\alpha) + (0\beta, 1\beta)$ is the cycle between two nodes, 00.....1, and 00...2, traversing all the nodes in the two k-ary r-cubes, 0**....*, and 1**..*.

Let us again assume that there exists a cycle, H_p, traversing all the nodes in k-ary r-cubes, 0**....*, 1**..*,....., and p**..*, which begins at node A = 00.....01 and ends at node B = 00.....02, where $p \geq 1$. Let us complete the proof by showing that there exists a cycle, traversing all the nodes in k-ary r-cubes, 0**....*, 1**..*,....., p**..*, and (p + 1)**..*, which begins at node A and ends at node B .

Let us take two adjacent nodes, $p\sigma$, and $p\delta$ in H_p, where σ, and δ are r-bit of k-ary vectors . By the induction assumption, there exists a Hamiltonian cycle, C_{p+1} between two nodes, $(p + 1)\sigma$, and $(p + 1)\delta$ in (p + 1)**...**.

Then, $H_{p+1} = H_p + C_{p+1} - (p\sigma, p\delta) - ((p + 1)\sigma, (p + 1)\delta) + (p\sigma, (p + 1)\sigma) + (p\delta, (p + 1)\delta)$ is the cycle between two nodes, 00.....1, and 00...2, traversing all the nodes in the k-ary r-cubes, 0**....*, 1**..*, ..., and (p + 1)**..* .



(a) Cycle of length 13 (b) Cycle of length 18

Figure 5.2 : Embedding cycles into 5-ary 2-cube

Lemma 5.3 : Between any two adjacent nodes in the k-ary 2-cube, a ring of length n, $k \le n \le k^2$ can be embedded with dilation 1, if k is odd.

Proof :

Case 1 : n is odd

Since $n \ge k$, and n is odd, there exists integers p and r such that $n = p * k + r$, where $r < k$.

1) When r is even, k and p must be odd.

a) Let us make a cycle C of length r, using 0*, and 1*.

Without loss of generality, let us assume that the adjacent nodes are 00 and 01. The procedure here is to make a line of length r/2 in 0*. That is, line, L_1 starts at node 00, and ends at node 0(r/2 - 1). Then make a corresponding line, L_2 in 1*, that begins at node 10 and ends at node 1(r/2 - 1). Then, $L_1 \cup L_2$ + (00, 10) + (0(r/2 - 1), 1(r/2 - 1)) is a cycle of length r.

b) Make a cyle of length n = p * k + r.

Let us make a cycle, C' of length p * k, using 2*, 3*,...., (p + 1) *, that begins at node 20 and ends at node 21. We can find the cycle, C' as is done in the proof of Lemma 5.1. Then $C \cup C'$ - (10, 11) - (20, 21) + (10, 20) + (11, 21) is a cycle of length n.

2) When r is odd, p must be even.

a) Let us make a cycle C of length r + k , using 0* and 1*.

The procedure here is to make a line of length (r + k)/2 in 0*. That is, a line, that starts at node 00 and ends at node 0((r + k)/2 - 1). Then make a corresponding line in 1*, that begins from node 10, and ends at node 1((r + k)/2 - 1).

b) Make a cycle of length n = p * k + r.

Let us make a cycle, C' of length (p - 1) * k, using 2*, 3*,...., and p *, that begins at node 20 and ends at node 21. We can find the cycle, C' as is done in the proof of Lemma 5.1. Then $C \cup C'$ - (10, 11) - (20, 21) + (10, 20) + (11, 21) is a the cycle of length n.

Case 2 : n is even

1) When $n < k$, we can embed cycle of length n in 0*, and 1*, as in the proof of Case 1.

Otherwise, since $n \geq k$, and n is even, there exists integers p, and r such that $n = p * k + r$, where $r < k$.

ıen r, and p are even, embed as in the proof of 1) in Case 1.

ʰ⁻ⁿ r, and p are odd, embed as in the proof of 2) in Case 1.

5.4 : A ring with odd length can't be embedded with dilation 1 into a k-ary n-
ıen k is even.

ıt us say that a k-ary n-cube node is even if the sum of all the n digits in its k-ary
is even, and that it is odd otherwise. Then, k-ary n-cube is a bipartite graph if k
Thus when k is even, any dilation 1 embedding of a ring into that k-ary n-cube
ve to alternate between visiting even nodes and odd nodes.
ring of odd length can't be embedded with dilation 1 into any k-ary n-cube,
: is even.

...ıa 5.5: A ring with n nodes between any two adjacent nodes, $n \leq k^2$ with k being even can be embedded into a k-ary 2-cube with dilation 2 if n is odd, and with dilation 1 if n is even.

Proof :

Case 1 : n is even

Since $n \geq k$ and n is even, there exists integers p and r such that $n = p * k + r$, where $r < k$. Here r must be even.

a) Let us make a cycle C of length r, using 0*, and 1*.

The procedure here is to make a line of length r/2 in 0*. That is, line L_1 starts at node 00 and ends at node 0(r/2 - 1). Then make a corresponding line, L_2 in 1*, that begins at node 10 and ends at node 1(r/2 - 1). Then, $L_1 \cup L_2$ + (00, 10) + (0(r/2 - 1), 1(r/2 - 1)) is a cycle of length r.

b) Make a cycle of length $n = p * k + r$.

Let us make a cycle, C' of lenth p * k, using 2*, 3*,...., (p + 1) *, that begins at node 20 and ends at node 21. We can find the cycle, C' as is done in the proof of Lemma 5.1. Then $C \cup C'$ - (10, 11) - (20, 21) + (10, 20) + (11, 21) the cycle of length n.

Case 2 : n is odd

Just embed the cycle, C of length n + 1 on the k-ary 2-cube. Then, line, L of length n that is in cycle C is the embedding of length n, cycle with dilation 2.

Theorem 5.1 : Between any two adjacent nodes in a k-ary p-cube, a ring with n nodes, $k^{p-1} < n \leq k^p$ can be embedded with dilation 1 if k is odd.

Proof :

Let us prove by induction on p. When p = 2, by lemma 5.3.
Assume that the above is true when p = q, $q \geq 2$, and suppose that p = q + 1.

By assumption, there exists integers r, and s such that $n = s * k^q + r$, where $0 \leq r < k^q$.

1) Make a cycle of length r.

Without loss of generality, let us take the adjacent nodes to be 000...01 and 000...02. By the induction assumption, we can embed a cycle C of length r in a k-ary q-cube, that is, in 0**..** between two nodes 000...01, and 000,...02.

2) Make a cycle of length $s * k^q + r$

By Lemma 5.2, there exists a Hamiltonian cycle, H_1 between two nodes, 100..01, and 100...02. $C \cup H_1$ - (000...01, 000..02) - (100..01, 100..02) + (000...01, 100..01) + (000..02, 100..02) is the cycle of length $k^q + r$.

In a recursive way, we can make a cycle of length $s * k^q + r$ between any two adjacent nodes.

Theorem 5.2 : A ring with n nodes, $k^{p-1} < n \leq k^p$ can be embedded into a k-ary p-cube with dilation 2 if n is odd, and with dilation 1 if n is even.

Proof :

Case 1 : n is even

We can prove this using lemma 5.5 , and the same method as is used in the proof of Theorem 5.1.

Case 2 : n is odd

Similar to case 1 in Theorem 5.1.

Chapter 6

Embedding a Hamiltonian Cycle into Faulty K-ary n-cube

6.1 Introduction

A Hamiltonian cycle in the k-ary n-cube is a cycle containing all nodes. Many communication algorithms can be efficiently implemented in parallel systems using Hamiltonian cycles [Ber91, John88]. In this chapter, we consider the problem of finding a Hamiltonian cycle in a faulty k-ary n-cube. In [Cha 91], the embedding of a Hamiltonian cycle in an n-dimensional binary hypercube in the presence of 2n - 5 edge faults is given. In the case of the k-ary n-cube, since there exists n edge disjoint Hamiltonian cycles [Als90], there is a fault free Hamiltonian cycle even in the presence of up to n - 1 edge faults. The results presented in this chapter improve upon this.

6.2 Embedding a Hamiltonian cycle into faulty k-ary n-cube.

The main interest here is the existence of a Hamiltonian cycle in the k-ary n-cube with the edge faults. The necessary condition for the existence of a Hamiltonian cycle in a graph is that each node has to be incident to at least two nodes.

Most of the lemmas and theorems in this chapter can be proven using the following concept. Let R and S be two cycles of length r and s and let e_1 and e_2 be two edges connecting two adjacent nodes of R and S as shown in Figure 6.1[a]. Then a cycle of length r + s can be constructed as shown in Figure 6.1[b]. The edges e_1 and e_2 are called the connecting edges of R and S.

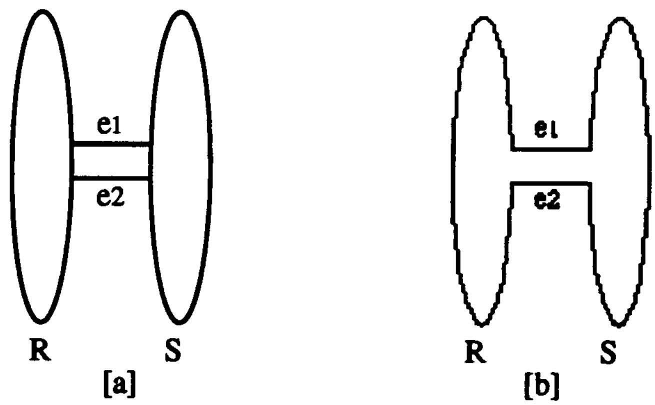


Figure 6.1: Combining two cycles into one

Any edge between two nodes that differ in the ith digit position is said to be of dimension i, $0 \leq i \leq n - 1$. A Hamiltonian cycle in the k-ary n-cube is said to be dominant in dimension i, $0 \leq i \leq n - 1$, if the number of edges not of dimension i used in the Hamiltonian cycle is less than or equal to $2 (k^{n-1} - 1)$.

Lemma 6.1 shows that there exists a dominant Hamiltonian cycle along any dimension in the k-ary n-cube. Since there are k^n edges in the Hamiltonian cycle, a dominant Hamiltonian cycle in dimension i, $0 \leq i \leq n - 1$, uses at least $k^n - 2 (k^{n-1} - 1) = k^{n-1} (k - 2) + 2$ edges of dimension i.

Let us take any p consecutive k-ary n-cubes, **..*j, **..*(j + 1), ..., and **..*(j + p -1) from the k k-ary n-cubes, obtained by decomposing the k-ary (n + 1)-cube in terms of dimension, 0, where $j \geq 0$, and $(j + p - 1) \leq k - 1$. By extending the definition of dominant Hamiltonian cycle for the k-ary n-cube, we can get the definition of dominant Hamiltonian cycle for the graph, **..*j ∪ **..*(j + 1) ∪,....., ∪**..*(j + p -1) along dimension i, $0 < i \leq k - 1$ as the Hamiltonian cycle, that includes at least $p * (k^{n-1} (k - 2) + 2) - 2(p - 1)$ edges of dimension i.

When p = k, then $p * (k^{n-1} (k - 2) + 2) - 2(p - 1) = k^n (k - 2) + 2$. When the k-ary (n + 1)-cube is decomposed into k k-ary n-cubes along any other dimension, A similar definition of a dominant Hamiltonian cycle for the p consecutive k-ary n-cubes can be given.

Lemma 6.1 : There exists a dominant Hamiltonian cycle along any dimension in the k-ary n-cube.

Proof :

The embedding used in the proof of Lemma 5.2 is the dominant embedding. When n = 2, just 2(k - 1) edges of dimensions that differ from i, $0 \le i \le n - 1$ are used. Assume that $2(k^{p-1} - 1)$ edges of dimensions that differ from i, $0 \le i \le n - 1$, are used in the embedding of Hamiltonian cycle for the k-ary p-cube. Then in the construction of k-ary (p + 1)-cube, at most $k * 2(k^{p-1} - 1) + 2(k - 1) = 2(k^p - 1)$ edges of dimension different from i, $0 \le i \le n - 1$ is used in the embedding of Hamiltonian cycle for the k-ary (p + 1)-cube.

The existence of a dominant Hamiltonian cycle along any dimension in the presence of 2n - 2 edge faults for k-ary n-cube is shown in Lemma 6.2, Lemma 6.3, and Theorem 6.1.

Lemma 6.2 : There exists a dominant Hamiltonian cycle along any dimension in the k-ary 2-cube with 2 edge faults.

Proof :

Let us say that the dimension along which the dominant Hamiltonian cycle exists is the interested dimension.

Case 1: No faulty edges are in the interested dimension.

Without loss of generality, let us assume that the interested dimension is 0. Then, each of k-ary 1-cubes, 0*, 1*, .., and (k - 1)* forms a cycle of length, $k \geq 3$. $k \geq 3$ and just two edge faults means that 0* includes one edge, (0u, 0v) such that there exists one corresponding edge, e in 1* or (k - 1)*, say, 1* with all the three edges,e = (1u, 1v), (0u, 1u), and (0v, 1v) being nonfaculty. $0^* \cup 1^* - (0u, 0v) - (1u, 1v) + (0u, 1u) + (0v, 1v)$ is the dominant Hamiltonian cycle for $0^* \cup 1^*$. Using recursion, the remaining part of Case 1 can be proved.

Case 2 : Just one faulty edge is in the interested dimension.

Without loss of generality, let us assume that the interested dimension is 0, and that one faulty edge is (00, 01). There exists one corresponding edge, e in 1* or (k - 1)*, say 1* with all the three edges, e = (10, 11), (01, 11), and (00, 10) being nonfaulty.

$0^* \cup 1^* - (00, 01) - (10, 11) + (00, 10) + (01, 11)$ is the dominant Hamiltonian cycle for $0^* \cup 1^*$. Using recursion, the remaining part of case 2 can be proved.

Case 3 : Two faulty edges are in the interested dimension.

Let us assume that the interested dimension is 0, and that the two faulty edges are in 0*. Let us denote the two faulty edges as (0u, 0v), and (0w, 0x). Then, (1u, 1v), (0u, 1u), (0v, 1v), ((k - 1)w, (k - 1)x), (0w, (k - 1)w), and (0x, (k - 1)x) are nonfaculty. $0^* \cup 1^* - (0u, 0v) - (1u, 1v) + (0u, 1u) + (0v, 1v) \cup (k-1)^* - (0w, 0x) - ((k - 1)w, (k - 1)x) + (0w, (k - 1)w) + (0x, (k - 1)x)$ is the dominant Hamiltonian cycle for $0^* \cup 1^* \cup (k - 1)^*$. Using recursion, the remaining part of case 3 can be proved. If one faulty edge is in say, 0* and another faulty edge is in say, 3*, a similar proof can be given.

Lemma 6.3 : There exists a dominant Hamiltoninan cycle along any dimension in the 3-ary n-cube with the 2n - 2 edge faults, such that the Hamiltonian cycle includes any given nonfaulty edge e in that direction.

Proof :

We shall prove the lemma by induction on n. When n = 2, Lemma 6.2 applies. Assume that the lemma is true in the case n = p - 1 where $p \geq 3$, and suppose that n = p. Without loss of generality, let us try to show that there exists a dominant Hamiltoninan cycle in dimension 0. From the fact that there are 2p - 2 edge faults in the 3-ary p-cube, we know there exists a dimension $i \neq 0$, $0 \leq i \leq p - 1$, such that at most 1 edge fault occurs in that dimension. We can also assume, without loss of generality, that the dimension i is p - 1. Let us decompose the 3-ary p-cube in terms of dimension p - 1, and make 3 3-ary (p - 1)-cubes, 0***..*, 1***..*, and 2***..* . Without loss of generality, assume that the interesting nonfaulty edge, e is in 0***..*.

Case1: 2p - 2 edge faults are in 0***..*, 1***..*, or 2***..*. Suppose they are in 0***..*.

Let us denote the 2p - 2 faulty edges by e_1, e_2, ..., $e_{(2p - 2)}$. Here, we know that the nodes, $0v_{11}$, $0v_{12}$, $0v_{21}$, and $0v_{22}$ are not incident to any faulty edge in the dimension p - 1, where $e_1 = (0v_{11}, 0v_{12})$ and $e_2 = (0v_{21}, 0v_{22})$. Let us temporarily assume that $e_1 = (0v_{11}, 0v_{12})$ and $e_2 = (0v_{21}, 0v_{22})$ are connected. Then, by the induction assumption, there exists a dominant Hamiltonian cycle, H_0 in 0***..*, that includes the particular interesting edge. If e_1 and e_2 are not included in H_0, then we can easily prove that there exists a dominant Hamiltonian cycle in the 3-ary p-cube that includes that particular interesting edge. If e_1 and e_2 are included in H_0, we can construct each of the corresponding dominant Hamiltonian cycles, H_1 and H_2 in 1***..*, and 2***..* in such a way that $(0u, 0v) \in H_0$ means $(1u, 1v) \in H_1$ and $(2u, 2v) \in H_2$.

Then $H = H_0 \cup H_1$ - $(0v_{11}, 0v_{12})$ - $(1v_{11}, 1v_{12})$ + $(0v_{11}, 1v_{11})$+ $(0v_{12}, 1v_{12}) \cup H_2$ − $(0v_{21}, 0v_{22})$ - $(2v_{21}, 2v_{22})$ + $(0v_{21}, 2v_{21})$+ $(0v_{22}, 2v_{22})$ is the dominant Hamiltonian cycle that includes the particular interesting edge.

A similar proof can be given for the case where just one edge from e_1 and e_2 is included in H_0.

Case 2 : 2p - 3 edge faults are in 0***..*, 1***..*, or 2***..*. Suppose they are in 0***..*.

The proof is similar to the proof of case 1.

Case 3: Number of edge faults in each of 0***..*, 1***..*, and 2***.. is at most 2p - 4.

By the induction assumption, there exists a dominant Hamiltonian cycle, H_0 with dimension 0 in 0***..*, that includes the particular interesting edge. Since H_0 is a dominant Hamiltonian cycle, H_0 includes $k^{p-1}(k - 2) + 2$ edges of dimension 0.

$k^{p-1}(k - 2) + 2 - (2p - 4) = k^{p-1}(k - 2) + 6 - 2p > 4$ for each $k \geq 3$, and at most one edge fault in dimension p - 1 implies that H_0 includes two edges, $e_1 = (0v_{11}, 0v_{12})$ and $e_2 = (0v_{21}, 0v_{22})$ of dimension 0 such that $e \neq e_1$, $e \neq e_2$, and all the edges, $(1v_{11}, 1v_{12})$, $(2v_{21}, 2v_{22})$, $(0v_{11}, 1v_{11})$, $(0v_{12}, 1v_{12})$, $(0v_{21}, 2v_{21})$, and $(0v_{22}, 2v_{22})$ are nonfaulty. By the induction assumption, there exists a dominant Hamiltonian cycle H_1 with dimension 0 in 1***..*, that includes edge, $(1v_{11}, 1v_{12})$. Similarily, there exists a dominant Hamiltonian cycle H_2 with dimension 0 in 2***..* that includes edge, $(2v_{11}, 2v_{12})$. Then $H = H_0 \cup H_1$ - $(0v_{11}, 0v_{12})$ - $(1v_{11}, 1v_{12})$ + $(0v_{11}, 1v_{11})$+ $(0v_{12}, 1v_{12}) \cup H_2$ − $(0v_{21}, 0v_{22})$ - $(2v_{21}, 2v_{22})$ + $(0v_{21}, 2v_{21})$+ $(0v_{22}, 2v_{22})$ is the dominant Hamiltonian cycle, that includes the particular interesting edge e.

Theorem 6.1 : There exists a dominant Hamiltonian cycle in any direction in the k-ary n-cube with 2n - 2 edge faults, such that the Hamiltonian cycle includes any particular nonfaulty edge in that direction.

Proof :

For the case when k = 3, we can apply Lemma 6.3. The proof for the case when k ≥ 4 is similar to the proof of Lemma 6.3.

The existence of a dominant Hamiltonian cycle in the k-ary n-cube with 4n - 5 edge faults, in which each node is incident to at least two nonfaulty links, is shown in the following lemmas, and Theorem 6.2.

Lemma 6.4 : There exists a dominant Hamiltonian cycle in the k-ary 2-cube with 3 edge faults, provided that each node is incident to at least two nonfaulty links.

Proof :

Case 1: All the edge faults are in one dimension, say 0.

Embed a dominant Hamiltonian cycle in the direction of dimension 1 as in the proof of Lemma 5.1.

Case 2: Exactly two edge faults are in one dimension, say, 0.

We can easily show that there exists a dominant Hamiltonian cycle in the direction of dimension 0.

Lemma 6.5 : There exists a dominant Hamiltonian cycle in the k-ary 3-cube with 7 edge faults, provided that each node is incident to at least two nonfaulty links.

Proof :

7 edge faults in the k-ary 3-cube means that there exists a dimension, say 2 in which at least 3 edge faults occur. Let us decompose the k-ary 3-cube into k k-ary 2-cubes, 0**, 1**, .., and (k - 1)** with respect to dimension 2.

Case 1: At least 4 edge faults occur in dimension 2.
This means at most three faulty edges occur in the dimension 0 or 1.

Case 1.1: Three edge faults in dimensions 0, and 1 are incident on one particular node.

Without loss of generality, let us assume that three edge faults, e_1, e_2, and e_3 in dimension 0 and 1 are in 1**, and denoted by e_1= (111, 110), e_2= (111, 101), and e_3= (111, 121).

Case 1.1.1: At most, three edges among the four faulty edges in dimension 2 are incident on the nodes, 111, 110, 101, and 121.

In this case, there exists one edge, say e_1= (111, 110) such that in 0**, or 2**, say, 0**, e_4= (011, 010) is nonfaulty, and two edges, e_5= (011, 111), e_6= (010, 110) are also nonfaulty. If we temporily assume that e_1 is nonfaulty, there exists a dominant Hamiltonian cycle, H_0 in 1** in the direction of dimension 0 that includes the edge e_1 by Lemma 6.4. In a similar way, there exists a dominant Hamiltonian cycle H_1 in 0** in the direction of dimension 0, that includes the edge e_4. $H_0 \cup H_1$ - (111, 110) - (011, 010) + (011, 111) + (010, 110) is the dominant Hamiltonian cycle for 0** ∪1** . The remaining part of the proof is by recursion.

Case 1.1.2 : All the four faulty edges in the dimension 2 are incident on the nodes, 111, 110, 101, and 121.

In this case, if there exists one edge, say, e_1= (111, 110) such that in 0**, or 2**, say, 0**, e_4= (011, 010) is nonfaulty, and two edges, e_5= (011, 111), e_6= (010, 110) are also nonfaulty, prove as for the case 111. If not, we can decompose the k-ary 3-cube into k k-ary 2-cubes, **0, **1, ..., **(k - 1) along dimension, 0 such that one of k-ary 2-cubes, 0**, 1**,..., and (k - 1)**, contains three faulty edges, and each of the other k-ary 2-cubes, contains at most one faulty edge . The remaining part of proof is almost similar to the proof of Case 1.1.1.

Case 1.2: At most two edge faults in dimensions 0, and 1 are incident on some particular node.

Let us take the k-ary 2-cube with the most edge faults from 0**, 1** ..., and (k - 1)**. Without loss of generality, let us assume that the chosen k-ary 2-cube is 0**. By Lemma 6.4, there exists a dominant Hamiltonian cycle, H_0 in 0** in the direction of dimension 0, or 1. Assume it is in dimension 0. The existence of at least, $k^2 \geq 3^2 = 9$ edges in H_0, and at most 7 edge faults in the dimension 2 mean that H_0 includes one edge, $e_1 = (0v_{11}, 0v_{12})$ such that in 1*, or (k - 1)*, say, 1**, there exists one edge, e with all the edges, $e = (1v_{11}, 1v_{12})$, $(0v_{11}, 1v_{11})$, and $(0v_{12}, 1v_{12})$ being unfaulty. At most, one faulty edge in 1* means that there exists a dominant Hamiltonian cycle, H_1in the direction of dimension 0 that includes edge, e. $H_0 \cup H_1 - (0v_{11}, 0v_{12}) - (1v_{11}, 1v_{12}) + (0v_{11}, 1v_{11}) + (0v_{12}, 1v_{12})$ is the dominant Hamiltonian cycle for $0^{**} \cup 1^{**}$. The remainder of the proof is by recursion.

Case 2: Exactly, three edge faults are in dimension 2.

If in each of 0**, 1**, .., and (k - 1)**, there exists at most three edge faults, we can prove as in the proof of case 1. Hence let us assume that all the four edge faults occur in 0**, 1**, .., or (k - 1)**, say, 0**.

Case 2.1 : At most three faulty edges in dimension 0, and 1 are incident on one particular node.

The proof here is similar to the proof of Case1.1.1.

Case 2.2 : All four faulty edges in dimension 0, and 1are incident on one particular node.

Without loss of generality, let us assume that the four faulty edges , e_1, e_2, e_3, e_4 in dimensions 0, and 1 are denoted by e_1= (111, 110), e_2= (111, 101), e_3= (111, 121), e_4 = (111, 112). Then, there exists, say, e_1= (111, 110), e_2= (111, 101) such that (011, 010) in 0** , (011, 111), and (010, 110) are nonfaulty, also (211, 201) in 2** , (111, 211), and (101, 201) are nonfaulty. If we temporarily assume that e_1 and e_2 are nonfaulty, the there exists a dominant Hamiltonian cycle H_1 in 1**. Let us assume that H_1 includes the edges, e_1, and e_2 . By Theorem 6.1, there exists a dominant Hamiltonian cycle, H_0 in 0**, that includes the edge (011, 010). Also,there exists a dominant Hamiltonian cycle, H_2 in 2**, that includes the edge, (211, 201). Then $H_1 \cup H_0$ - (111, 110) - (011, 010) + (011, 111) + (010, 110) $\cup H_2$ - (111, 101) - (211, 201) + (111, 211) + (101, 201) is the Hamiltonian cycle for 0** $\cup$ 1** $\cup$ 2**. The remaining part of the proof is by recursion. In the case of H_1 including just e_1 or e_2 , we can give a similar proof. If H_1 includes neither e_1 nor e_2 , we can also give a similar proof.

Lemma 6.6 : There exists a dominant Hamiltonian cycle in the k-ary 4-cube with 11 edge faults, provided that each node is incident to at least two nonfaulty links.

Proof: 11 edge faults in the k-ary 4-cube means there exists a dimension, say, 3, in which at least 3 edge faults occur. Let us decompose the k-ary 4-cube into k k-ary 3-cubes, 0***, 1***, 2***, ..., (k - 1)*** with respect to dimension 3.

Case 1: At least, 4 edge faults occur in dimension, 3.

Case 1.1: Six edge faults in the 0, 1, and 2 dimension are incident on some particular node.

The proof here is similar to the proof of Case 2.2 in Lemma 6.5.

Case 1.2: Five edge faults in the 0, 1, and 2 dimension are incident on some particular node.

The proof here is similar to the proof of Case 1.1 in Lemma 6.5.

Case 1.3: At most, four edge faults in the 0, 1, and 2 dimension are incident on each node.

The proof follows from Theorem 6.1, Lemma 6.5, and the definition of a dominant Hamiltonian cycle.

Case 2: Exactly three faulty edges are in dimension, 3.

The proof here is similar to the proof of Case 1.

Lemma 6.7 : There exists a dominant Hamiltonian cycle in the k-ary 5-cube with 15 edge faults, provided that each node is incident to at least two nonfaulty links.

Proof: 15 edge faults in the k-ary 5-cube means that there exists a dimension, say, 4, in which at least 3 edge faults occur. Let us decompose the k-ary 4-cube into k k-ary 3-cubes, 0****, 1****, 2****, ..., (k - 1)**** with respect to dimension 4.

Case 1: At least , 4 edge faults occur in dimension, 4.

Case 1.1: 8 edge faults in the 0, 1, 2 and 3 dimension are incident on some particular node.

The proof here is similar to the proof of Case2.2 in Lemma 6.5.

Case 1.2: 7 edge faults in the 0, 1, 2 and 3 dimension are incident on some particular node.

The proof here is similar to the proof of Case1.1 in Lemma 6.5.

Case 1.3: At most, six edge faults in the 0, 1, and 2 dimension are incident on each node.

The proof follows from Theorem 6.1, Lemma 6.6, and the definition of a dominant Hamiltonian cycle.

Case 2: Exactly, 3 edge faults are in dimension, 4.

The proof here is similar to the proof of Case 1.

Theorem 6.2 : There exists a dominant Hamiltonian cycle in the k-ary n-cube with 4n - 5 edge faults, provided that each node is incident to at least two nonfaulty links, and this result is optimal.

Proof: We shall prove the theorem by induction on n.

When n = 2, 3, 4, and 5, the proof follows from Lemma 6.4, 6.5, 6.6, and 6.7 respectively. Let us assume that there exists a dominant Hamiltonian cycle in the k-ary p-cube with 4p - 5 edge faults, p > 5 in which each node is incident to at least two nonfaulty links and suppose that n = p + 1. In this case, there exists a dimension, say, p in which at least 4 edge faults occur. Let us decompose the k-ary (p + 1)-cube into k k-ary p-cubes, 0***...*, 1***..*, 2***...*, ..., (k - 1)***..* with respect to dimension p.

Case 1: 2p edge faults in dimension 0, 1, 2,.., (p -1) are incident on some particular node.

The proof here is by the induction assumption, and uses an approach similar to the proof of Case 2.2 in Lemma 6.5.

Case 2: (2p - 1) edge faults in dimension 0, 1, 2,.., (p -1) are incident on some particular node.

The proof here is by the induction assumption, and uses an approach similar to the proof of Case 2.1 in Lemma 6.5.

Case 3: At most, (2p - 2) edge faults in dimension 0, 1, 2,.., (p -1) are incident on each node.

The proof follows from Theorem 6.1 and the definition of dominant Hamiltonian cycle.

This result presented in Theorem 6. 2 is optimal because there exists k-ary n-cube with 4n - 4 edge faults, each node of which is incident to at least two nonfaulty links such that no Hamiltonian cycle exists. Consider the 4n edges incident on the two nodes, 000...00, and 110...00. Let us assume that among the above 4n edges, just (000..00, 010..00), (010..00, 110..00), (110..00, 100..00), and (100..00, 000..00) are nonfaulty. In this case, nodes 000..00, and 110..00 each have exactly two nonfaulty links incident to them. Those four edges form a cycle by themselves, making a Hamiltonian cycle impossible.

Chapter 7

Processor Allocation In Banyan-Hypercube

7.1 Introduction

In parallel machines, user programs are composed of a number of tasks, each of which can be executed by several processors in parallel. The processor allocation problem involves assigning available processors to arriving tasks, and releasing processors from finished tasks for later use . Efficient allocation and deallocation in the parallel machine is an important issue for achieving high performance in the parallel machine. An efficient processor allocation scheme maximizes the resource utilization, and reduces external and internal fragmentation.

An allocation method is called static if it accommodates incoming requests without considering processor relinquishment. If the allocation method considers processor relinquishment, when it accommodates incoming requests, the method is said to be dynamic.

An allocation method in which the operating system collects a sufficient number of requests before proceeding to allocate processors to them is said to be off-line. On the other hand, if each processor request is honored or denied immediately after it arrives, regardless of subsequent requests, The allocation method is said to be on-line.

The allocation of processors in a BH network consists of two steps.

1. Determination of the size of the incoming task in terms of the number of processors needed in order to accommodate it.

2. Recognizing and locating a subbanyan that can accommodate the incoming task.

Step 2 will be the main topic of this Chapter.

This Chapter is organized as follows. Section 7.2 introduces the necessary definitions and notation. Section 7.3 shows that the subbanyan allocation is an NP-complete problem. Section 7.4 presents the allocation strategy .

7.2 Preliminaries

Let us denote the nodes constituting BH(h, n, s) by V(BH(h,n,s)).
A banyan-hypercube BH(h1, n1, s) covers a banyan-hypercube BH(h2, n2, s), denoted by BH(h1, n1, s) $\supset$BH(h2, n2, s) if V(BH(h1, n1, s)) $\supset$ V(BH(h2, n2, s)).

The set of all free nodes of the banyan-hypercube BH(h, n, s) is denoted as V_f (BH(h, n, s)). The dimension of the banyan-hypercube BH(h, n, s) is n, and the depth of banyan-hypercube BH(h, n, s) is h.

A banyan-hypercube BH(h1, n1, s) is said to be prime if BH(h1, n1, s) $\supset$ BH(h, n, s) and there is no banyan-hypercube BH(h2, n2, s) such that BH(h2, n2, s) $\supset$ BH(h, n, s) and BH(h1, n1, s) $\supset$ BH(h2, n2, s).

BH(hh, kk, s) is said to be a subbanyan of BH(h, k, s) if and only if BH(hh, kk, s) is a banyan hypercube, where $hh \le h$, $kk \le k$, $hh \le kk + 1$, and BH(h, k, s) $\supset$ BH(hh, kk, s).

An allocation request set is denoted by $R = \{(h_1, n_1), (h_2, n_2), \ldots (h_k, n_k)\}$, where $k > 0$, each n_i is the dimension of banyan-hypercube, and each h_i is the depth of the corresponding banyan-hypercube. An allocation method is called statically optimal if a BH(h, n, s) can accommodate any input request sequence $\{BH(h_i, n_i, s)\}_{i=1.., k}$ if and only if $\Sigma (h_i) * (s)^{(n_i)} \le h * s^n$.

7.3 Subbanyan Allocation problem

It has been shown that the decision problem corresponding to the off-line allocation of the hypercube is NP-complete, an optimal on-line strategy is not computable [Dut91], and that there exists an optimal algorithm for allocating hypercube statically [Che87].

We will show that off-line allocation of the Banyan-Hypercube is also an NP-complete problem, an optimal on-line strategy is not computable for the Banyan-Hypercube, and that there does not exist a statically optimal algorithm for allocating a banyan-hypercube.

Definition 7.1 :

An allocation from BH(h, n, s) to an allocation request set R is a mapping $f : R \rightarrow 2^{V_f(BH(h, n, s))}$ such that for all i, $1 \le i \le |R|$, the graph induced in BH(h, n, s) by $f(h_i, n_i)$ is a banyan-hypercube with depth h_i, and dimension n_i. A feasible allocation from BH(h, n, s) to R is a mapping $f : R \rightarrow 2^{V_f(BH(h, n, s)}$ such that f is an allocation for R and for all $(h_i, n_i) \in R$, and $(h_j, n_j) \in R$, $f(h_i, n_i) \cap f(h_j, n_j) = \emptyset$, if $i \neq j$. An allocation request set for which there is a feasible allocation is a feasible allocation request set.

Problem 1 : Subcube Allocation

Given a positive integer n, the set P of all prime cubes of $V_f(Q_n)$, and an allocation request set $R = (n_1, \ldots, n_k)$, is there a feasible allocation from Q_n to R ?

Problem 2 : Subbanyan Allocation

Given a banyan-hypercube BH(h, n, s), the set P of all prime subbanyan of $V_f(BH(h, n, s))$, and an allocation request set $R = \{(h_1, n_1), (h_2, n_2), \ldots (h_k, n_k)\}$, is there a feasible allocation from BH(h, n, s) to R ?

Theorem 7.1[Dut91]: Subcube Allocation is NP-complete.

Theorem 7.2 : Subbanyan Allocation is NP-complete.

Proof:

Let us consider the case where only nodes in the top level of the banyan-hypercube are unavailable. Trivially, this case can happen.

If we assume allocation request $R = \{(h_1, n_1), (h_2, n_2), \ldots (h_k, n_k)\}$ to BH(h, n, s) is such that $h_1 = h_2 = h - 1$, $n_1 = n_2 = n - 1$, and $h_3 = h_4 = \ldots = h_k = 0$, then we have to allocate (h_1, n_1), (h_2, n_2) into two subbanyans $BH_1(h - 1, n, s)$, $BH_2(h - 1, n, s)$ of BH(h, n, s) respectively. Since the only available subbanyan of BH(h - 1, n, s) in BH(h, n, s) is the one whose nodes are all from level 0 to level n - 2 of BH(n, k, s), we have to check whether the allocation request $\{(h_3, n_3) \ldots (h_k, n_k)\}$ can be satisfied in the top level of BH(n, k, s). This problem is actually that of subcube allocation. Hence the subbanyan allocation problem is NP-complete.

Lemma 7.1 : There exists no statically optimal algorithm for the subbanyan allocation problem.

Proof:

Let us try to embed $BH_1(n - 1, k - 1, 2)$, $BH_2(n - 1, k - 1, 2)$, and $BH_3(2, k - 1, 2)$ into BH(n, k, 2).

Here, $(n - 1)*2^{(k - 1)} + (n - 1)*2^{(k - 1)} + 2^k = (n - 1)*2^k + 2^k = n*2^k$.

Hence, if there exists a statistically optimal algorithm, we must be able to embed $BH_1(n -1, k -1, 2)$, $BH_2(n -1,k - 1, 2)$, and $BH_3(2, k - 1, 2)$ into BH(n, k, 2).
But this obviously cannot be done .

Lemma 7.2 :

There exists no dynamically optimal algoritm for the subbanyan allocation problem.

Proof: This proof follows directly from Lemma 7.1.

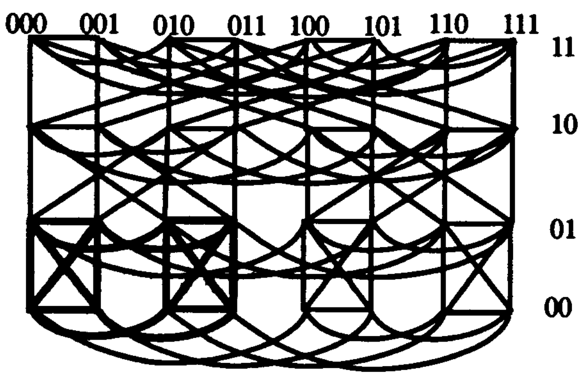


Figure 7.1 No dynamically optimal algorithm exists for BH

Lemma 7.3 :

Even in the case where the allocation sequence $R = \{(h_1, n_1), (h_2, n_2), \ldots (h_k, n_k)\}$ can be embedded into V_f (BH(h, n, s)) feasibly, there does not exist a dynamically optimal algorithm.

Proof:

Suppose that such an algorithm exists. Consider figure 7.1, which shows that BH(2,2,2) has been assigned. The two allocation request set R_1 = {(2, 2), (2, 3)} and R_2 = {(2, 2), (3, 2) } are both feasible. Suppose a request for a (2, 2) arrives first; then to grant the sequence of requests R_1, the algorithm will have to allocate the bottom right BH(2, 2, 2), while to grant the sequence of requests from R_2 , it should allocate the top left BH(2, 2, 2). Thus the algorithm can grant either R_1 or R_2 but not both, which contradcts the assumption that there exists a dynamically optimal algorithm that can grant all feasible allocation request sets.

7.4 Allocation Strategy

Our subbanyan recognition algorithm is based on the buddy strategy used to recognize subcubes.

The buddy strategy in hypercube recognition

Since there are 2^n nodes in an n-cube, 2^n allocation bits are used to keep track of the availability of the nodes. A value 0 (1) in the allocation bit indicates the availability (unavailability) of the corresponding node.

Allocation :

Step 1: Set k : = $| I_j |$, where $|I_j|$ is the dimension of a subcube required to accommodate the request I_j.

Step 2: Determine the least integer m such that all the allocation bits in the region

#[$m2^k$, $(m+1)2^k$-1] are 0's, and set all these allocation bits to 1's.

Step 3: Allocate nodes to the request I_j.

Relinquishment:

Step 1: Reset every pth allocation bit to 0, where p is the address of node that was released.

This strategy can be explained by the binary tree in Fig 7.2.
The level where the root node resides is numbered 0, and the nodes in level i are associated with subcubes of dimension n - i. A node in this binary tree is available only if all of its offspring are available. When an incoming request needs a Q_k, the buddy strategy searches level n - k of the tree from left to right and allocates the first available subcube to the request.

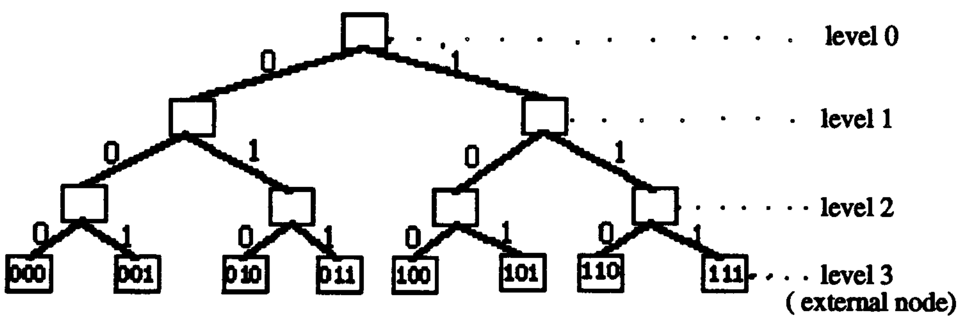


Figure 7.2 : The complete binary tree for the cube allocation using the buddy strategy

Definition 7.2

By rotation of a node, we mean a left cylic rotation of its address. If the address of i is $(i_{n-1}, i_{n-2},\ldots,i_1, i_0)$, then $Ro(i) = (i_{n-2},\ldots,i_1, i_0, i_{n-1})$.
By rotation of a graph G(V, E), we mean a graph Ro(G(V, E)) =G(Ro(V), Ro(E)), where Ro(V) = {Ro(i) | for all i belonging to V} and Ro(E) = { (Ro(i), Ro(j)) | for all (i, j) belonging to E}.

Lemma 7.4

The topology of a banyan-hypercube graph remains unchanged under rotation operation.

Proof:

Trivally, $Ro : (i_{n-1}, i_{n-2}, \ldots, i_1, i_0) \dashrightarrow (i_{n-2}, \ldots, i_1, i_0, i_{n-1})$ is a one to one onto and well defined function from V to Ro(v). Also by the definition of the rotation graph, There exists a one-to-one correspondence between their edge sets so that each corresponding edge of G and Ro(G) is incident on the corresponding nodes of V and Ro(v) respectively.

Lemma 7.5

Let BH(n, k, s) be one banyan-hypercube, S_1 the induced subgraph of BH(n, k, s), generated by all the nodes from level 0 to n - p of BH(n, k, s), and S_2 from level 1 to n - p + 1, ... S_l from level l - 1 to n - p+ l -1. Then S_1, S_2, ... S_l are topologically equivalent, where $n - p + l - 1 \leq n$.

Proof:

It is sufficient to show that S_1 is topologically equivalent to S_2, ... S_l respectively. Trivially $Ro(S_1) = S_2$ is . But by Lemma 7.4, we know that topology of S_1 is same as that of $Ro(S_1)$. Hence the topology of S_1 is equivalent to that of S_2.
In a similar way, we can show that S_1 is topologically equivalent to S_3, ...S_l respectively.

Lemma 7.6

Let BH(n, k, s) be one banyan-hypercube with n - 1 levels, and let BH(n_1, k_1, s) be a subbanyan such that $n_1 \leq n$, $k_1 \leq k$, and $n_1 \leq k_1 + 1$, whose lowest level lies at level 0 of BH(n, k, s). Then all the nodes existing at the lowest level of BH(n_1, k_1, s) form a buddy .

Proof

We shall prove lemma 7.6 by induction on k_1. When k_1 is 1, the claim is trivially true. Assume that the above is true when $k_1 = p$, and that $k_1 = p + 1$.
By the recursive structure of banyan-hypercube, a subbanyan of $k_1 = p + 1$ can be constructed from the s copies of subbanyans BH_1, and $BH_2 \ldots BH_s$, where k_1 of each subbanyan is p. By the induction assumption, each of the nodes existing at the lowest level of BH_i, for i = 1...s forms a buddy. Without loss of generality, assume that the labels of the nodes from BH_i, for i = 1..s, are in increasing order. That is, all the bits representing the nodes existing in BH_1, come before all the bits representing the nodes existing in BH_2, ...all the bits representing the nodes existing in BH_{s-1}, comes before all the bits representing the nodes existing in BH_s. Hence, there exists m such that all the nodes at the lowest level of BH_1 are in region $\#[ms^k, (m+1)s^k - 1]$, those of BH_2 are in region $\#[(m+1)s^k, (m+2)s^k - 1]$,.........and those of BH_s are in region $\#[(m+(s-1))\, s^k, (m+s)s^k - 1]$.

Here, our claim is that m is divisible by s. If not, nodes from address $\#[ms^k, (m+s)s^k - 1]$ can't form a Banyan-hypercube structure !!.
Since $\#[ms^k, (m+s)s^k - 1] = \#[(m/s)s^{k+1}, ((m/s)+1)s^{k+1} - 1]$, the proof follows.

By Lemmas 7.3, 7.4 and 7.5, we know that if we want to find a subbanyan, whose lowest level lies in level 1 of the original Banyan-hypercube, we have to rotate all the nodes from level 1 to the top level, and then proceed as in Lemma 7.6. Hence, the external nodes in the buddy tree for level i, $1 \leq i \leq k + 1$, of BH(h, k, s) can be generated by just rotating i times, each external node in the buddy tree for the level 0. For example, the buddy tree for level 1 of BH(4, 3, 2) comes from Figure 7.2, and is as follows.

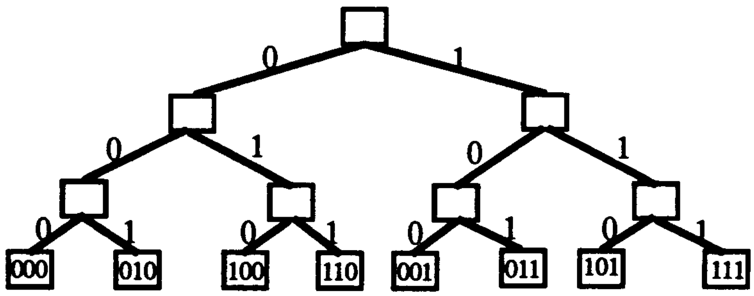


Figure 7.3 : Buddy tree for the level 1 of BH(4, 3, 2)

Lemma 7.7

Let BH(n, k, s) be one banyan-hypercube, and let BH(n_1, k_1, s) be the subbanyan such that $n_1 \le n$, $k_1 \le k$, and $n_1 \le k_1 + 1$.

To allocate such subbanyan as BH(n_1, k_1, s) in BH(n, k, s), including at least one node from level 0, it is necessary and sufficient that we find the available region #[ms^{k_1}, $(m + 1)s^{k_1} - 1$] with s^{k_1} nodes at level 0, using the buddy strategy such that for all h = 0..n_1, and for all v ∈ #[ms^{k_1}, $(m + 1)s^{k_1}-1$], (h, v) is available.

Proof:

The proof follows directly from Lemma 7.6.

Similarily, we can find a subbanyan, whose lowest level lies at level p, $0 \le p \le n - n_1 + 1$, of the original Banyan-hypercube.

DataStructure

Let T_i be the buddy tree of level i in BH(k + 1, k, 2), i = 0.., k, and $T_i(x_{k-1} \ldots x_{L+1}$ a $x_{L-1} \ldots x_1 x_0)$ be the external node $x_{k-1} \ldots\ldots x_1 x_0$ in the ith buddy tree, where $x_p = 0$, or 1 for each p = 0,...,k - 1. Then, make a directed path from T_0 to T_k in the order of increasing subscripts for each corresponding node. For example, the data structure for BH(3, 2, 2) is as follows.

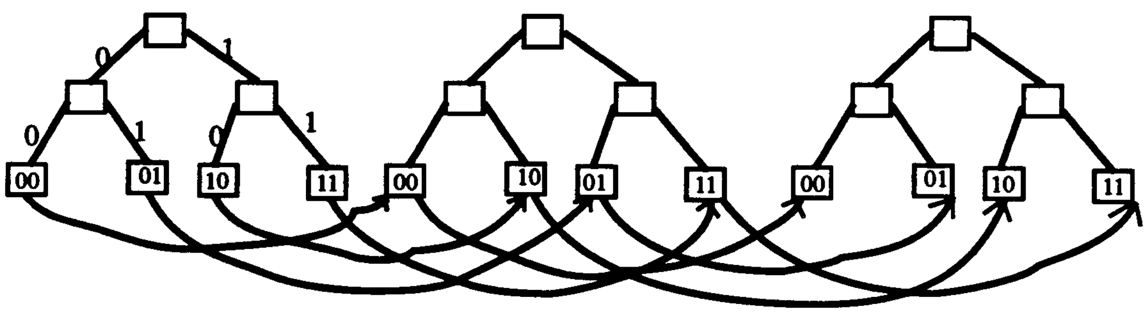


Figure 7.4 : Data structure for BH(3, 2, 2)

Algorithm

Let x[0, 2^k - 1] denote the region #[0, 2^k - 1] of level x in BH(h, k, 2) when all the nodes in level x of BH(h, k, 2) is represented by the buddy tree, and xn_1[m 2^{k_1}, (m + 1) 2^{k_1} - 1] denotes the nodes of BH(h, k, 2) which induce the subbanyan BH(n_1, k_1, 2), where nodes in level 0 of BH(n_1, k_1, 2) is x[m 2^{k_1}, (m + 1) 2^{k_1} - 1].

Allocation: Let us allocate BH(n_1, k_1, s) in BH(n, k, s), where $n_1 \leq n$, $k_1 \leq k$, and $n_1 \leq k_1 + 1$.

Step 1: Determine p such as $n - p + 1 = n_1$.

Step 2 : fact = TRUE;

x =0;

while (n - p + 1+ x ≤ n and fact) do

begin

{ determine the least integer m such that all the allocation bits in the region x[$m2^{k}1$, $(m + 1)2^{k}1$ - 1] are 0, and all the allocation bits in the region xh_1[$m2^{k}1$, $(m + 1)2^{k}1$ - 1] are 0}.

if such an m exist then fact = false

Else x = x + 1;

end

Step 3: If fact is false then

begin

Set all the bits in xh_1[$m2^{k}1$, $(m + 1)2^{k}1$ - 1] to 1's.

Allocate nodes to the request BH(n_1, k_1, 2).

end;

Else put the request into the waiting queue.

Relinquishment :

Release every bit p∈ xh_1[$m2^{k}1$, $(m + 1)2^{k}1$ - 1] , where xh_1[$m2^{k}1$, $(m + 1)2^{k}1$ - 1] is the region for the released subbanyan BH(n_1, k_1, 2).

Theorem 7.3 The above allocation strategy can recognize any available subbanyan.

Proof:

The proof follows from Lemma 7.7, and the correctness of the algorithm.

7.5 Algorithm Analysis

Let us analyze the algorithm's time complexity. The total number of subbanyan, $BH(h_1, k_1, 2)$ in $BH(h, k, 2)$, where $h_1 \leq h$, and $k_1 \leq k$ is $(h - h_1 + 1)\, 2^{(k - k_1)}$. The worst case time needed to determine the least integer m such that all the allocation bits in the region $x[m2^{k_1}, (m + 1)2^{k_1} - 1]$ are 0, and all the allocation bits in the region $xh_1[m2^{k_1}, (m + 1)2^{k_1} - 1]$ are 0 is $h_1\, 2^{k_1}$. Hence, the worst case complexity of allocation is $(h - h_1 + 1)\, 2^{(k - k_1)}\, h_1\, 2^{k_1} = (h - h_1 + 1)\, h_1\, 2^k$. The time complexity of deallocation is $h_1\, 2^{k_1}$. Hence, the overall time complexity of the algorithm is $(h - h_1 + 1)\, h_1\, 2^k$.

Chapter 8

Conclusion

8.1 Summary

In Chapter 2, the embedding of butterfly-like graphs into a banyan-hypercube network is studied. The butterfly-like structures considered here are the FFT, butterfly(wrap-around FFT), and the CCC (cube-connected cycle). Our embedding finds that the FFT graph, and CCC are the subgraphs of the smallest Banyan-Hypercubes which are big enough to hold them.

That is, the n-level FFT graph, with $(n + 1)\, 2^n$ vertices can be embedded into the BH(h, k, 2) with dilation 1, where $k > n$, and $h\, 2^k \geq (n + 1)\, 2^n \geq (h - 1)\, 2^k$, and the n-level CCC graph with $n\, 2^n$ in the BH(h', k', 2) with dilation 1, where $k' > n$, and $h'\, 2^{k'} \geq n\, 2^n \geq (h' - 1)\, 2^{k'}$. A butterfly network with $n\, 2^n$ nodes can be embedded with dilation 2 into the smallest banyan-hypercube.

In Chapter 3, the embedding of ring structured networks into a banyan-hypercube network is studied. The ring structures considered here are regular rings, X-trees, chordal rings, and the torus. We have obtained the following results.

1) A ring with n nodes, $n \leq h2^k$, can be embedded into BH(h, k, 2) with dilation 1.
2) A chordal ring with n nodes and $w = n/2$ $(n > 5)$ can't be embedded with dilation 1 into any one level BH(h, k, 2), where $2^k > n$.

3) If n is not divisible by 4, a chordal ring with n nodes and $w = n/2$ ($n > 5$) can't be embedded with dilation 1 into the BH(2, k, 2), where $2*2^k > n$.

4) If n is divisible by 4, a chordal ring with n nodes and $w = n/2$ ($n > 5$) can be embedded with dilation 1 into the BH(2, k, 2), where $2*2^k \geq n$ otherwise into BH(2, k, 4), where $2*4^k \geq n$.

5) A chordal ring with n nodes and $w = n/2$ ($n > 6$) can be embedded with dilation 1 into BH(m, k, 2) where $m*2^k \geq n \geq (m - 1)*2^k$, and $3 \leq m < k + 2$.

6) An L-level X-tree can't be embedded into BH(1, L, 2) with dilation 1.

7) A L-level X-tree can't be embedded into BH(1, L, 2) with dilation 2 and $M_2 < 2^{L-1} - 1$. (M_2 = number of edges, mapped with dilation 2)

8) A L-level X-tree can't be embedded into BH(2, L - 1, 2) with dilation 2 and $M_2 < 2^{L-2} - 1$.

9) A L-level X-tree can be embedded into BH(1, L, 2) with dilation 2 and $M_2 = 2^{L-1}-1$.

10)A L-level X-tree can be embedded into BH(2, L - 1, 2) with dilation 2,
expansion = $2^L/(2^L - 1)$, and $M_2 = 2^{L-2} - 1$.

11)A L-level X-tree can not be embedded into BH(h, h - 1, 2) with dilation 1,
if $h < L$.

12)A L-level X-tree can be embedded into BH(L, L - 1, 2) with dilation 1.

13) A $(l_1, l_2, \ldots\ldots\ldots, l_{d-1}, l_d)$ d-dimensional torus can be embedded into BH(1, n, 2), where $n = \sum \lceil \log_2 l_i \rceil$, for i = 1, 2 ....., d.

(a) with dilation 1 if l_i is even, for $1 \le i \le d$.

(b) otherwise, let l_{kj} for $1 \le j \le p$ be the given odd dimensions. Then the mapping can be done with dilation 2, and $M_2 = \sum D(l_{kj})$, for $1 \le j \le p$

where $D(l_{kj}) = (\prod l_i) / l_{kj}$, for i = 1,2 .....,d.

14) A $(l_1, l_2, \ldots\ldots\ldots, l_{d-1}, l_d)$ d-dimensional torus can be embedded into BH(2, n, 2), where $n + 1 = \sum \lceil \log_2 l_i \rceil$, for i = 1,2 .....,d.

(a) with dilation 1 if either all l_i, $1 \le i \le d$, is even; or only one l_i odd and the others even.

(b) otherwise, let l_{kj} for $1 \le j \le p$ be the given odd dimensions. Then the mapping can be done with dilation 2, and $M_2 = \sum D(l_{kj})$, for $1 \le j \le p$ where $D(l_{kj}) = (\prod l_i) / l_{kj}$, for i = 1,2 .....,d.

15) A $(2h, l_2, \ldots\ldots\ldots, l_{d-1}, l_d)$ d-dimensional torus can be embedded into BH(h, n, 2), where $n - 1 \ge \sum \lceil \log_2 l_i \rceil$, for i = 2,...d.

(a) with dilation 1 if for all $2 \le i \le d$, l_i is even.

(b) otherwise, let l_{kj} for $1 \le j \le p$ be the given odd dimensions. Then the mapping can be done with dilation 2, and $M_2 = 2h \sum D(l_{kj})$, for $1 \le j \le p$ where $D(l_{kj}) = (\prod l_i) / l_{kj}$, for i = 1,2 .....,d.

In Chapter 4, we have shown that the k-ary n-cube is a hierarchical Cayley graph as well as a product graph. From those facts, we can easily deduce some topological properties of k-ary n-cube such as diameter, average distance, connectivity, recursive decomposibility, and node-symmetry.

In Chapter 5, embedding of rings into the k-ary n-cube is investigated. Our embedding finds that a ring with n nodes, $k^{p-1} < n \leq k^p$, can be embedded into a k-ary p-cube with dilation 1 between any two adjacent nodes if k is odd. In the case of k being even, a ring with n nodes, $k^{p-1} < n \leq k^p$ can be embedded into a k-ary p-cube with dilation 2 if n is odd, and with dilation 1 if n is even.

In Chapter 6, we try to embed a Hamiltonian cycle in the presence of edge faults. Our embedding shows that there exists a dominant Hamiltoninan cycle in any direction in the k-ary n-cube with the 2n - 2 edge faults, such that the Hamiltonian cycle includes any particular nonfaulty edge in that direction; further it is shown that there exists a dominant Hamiltonian cycle in the k-ary n-cube with the 4n - 5 edge faults, provided that each node is incident to at least two nonfaulty links, and this result is optimal.

In Chapter 7, we consider the problem of allocating processors in banyan-hypercube multiprocessor to the arriving task. We have shown that subbanyan allocation problem is NP-complete, and that there does not exist any statically or dynamically optimal algorithm for recognizing subbanyans. We have presented an allocation algorithm that can recognize any available subbanyans.

8.2 Future Work

For future research, we have the following problems to explore :

From Chapter 2, try to determine whether the wrap-around butterfly is a subgraph of the optimal banyan-hypercube. If the wrap-around butterfly is not a subgraph of the optimal banyan-hypercube, finding an embedding with better average dilation will be interesting.

From Chapter 3 :

Try to embed a chordal-ring with $w \neq n/2$ into the Banyan-hypercube.

From Chapter 5, and 6:

Try to find out the maximal size of a ring in the k-ary n-cube with faulty nodes.

In addition, try to embed a complete binary tree, and mesh of trees in the k-ary n-cube.

From Chapter 7:

Try to allocate processors in the k-ary n-cube.

Bibliography

[Aie 91] B. Aiello, and T. Leighton, "Coding theory, hypercube embeddings, and fault tolerance", Proc. of 3rd annual ACM symposiun on parallel algorithms and architectures, pp125-135, July. 1991.

[Ake 84] S. B. Akers, and B. Krishnamutthy, "Group Graphs as interconnection networks", Proc. 14th Intl. Conf. Fault Tolerant Computinng, pp. 422-427, 1984.

[Ake 87] S. B. Akers, and B. Krishnamutthy, "A group-theoretic model for symmetric interconnection networks", Technical Report No. Cr-86-29, Dept of ECE, University of Massachusettes, June. 1987.

[Ale 82] R. Aleliunas, and A. Rosenberg, "On embedding Rectangular Grids in Square Grids", IEEE Trans. on Computers, Vol.31, No.9, pp.907-913, Sep.1982.

[Als 90] B.Alspach, J.-C. Bermond, and D. Sotteau, "Decomposition Into Cycles I : Hamiltonian Decompositions", Cycles and Rays, pp9 - 18, 1990.

[Ann 90] F. Annexstein, M. Baumslag, and A. L. Rosenberg, "Group action graphs and parallel architectures ", SIAM J.Computing, Vol. 19, No. 3, PP544-569, June. 1990.

[Ard 81] B. W. Arden and H. Lee, " Analysis of Chordal Ring Network", IEEE Trans.on Computers, Vol c-30, No.4, pp.291-295, Apr.1981.

[Ata 84] M. Atallah, and R. Kosaraju, "Graph problems on a mesh-connected processor array", Journal of the ACM, vol. 31, No. 3, PP649-667, July 1984.

[Ata 86] M. Atallah, and S. Hambrusch, "Solving tree problems on a mesh-connected processor array", Information and computation, Vol. 69, No. (1-3), pp168-187, 1986.

[Ata 88] M. Atallah, "On multidimensional arrays of processors", IEEE Trans. on Computers, Vol.37, No.10, pp.1306-1309, Oct. 1988.

[Bar68] G. H. Barnes et al, "The Illiac IV Computer", IEEE Trans.on Computers, Vol C-17, pp.749-757, 1968.

[Bau 91] M. Baumslag, M. C. Heydemann, J. Opatrny, and D. Scotteau, "Embeddings of schuffle-like graphs in hypercubes", Proc. of parallel architectures and languages in Europe, pp179-190.

[Ber 89] D. P. Bertsekas, and J. Tsitsiklis, "Parallel and Distributed Computation", Prentice Hall, Englewood Cliffs, New Jersey, 1989.

[Ber91] D. P. Bertsekas, C. Ozveren, G. D. Stamoulis, P. Tseng, and J. N. Tsitsiklis, " Optimal Communication Algorithms for Hypercubes", J of Parallel and Distributed Computing, Vol.11, pp. 263-275, 1991.

[Bha 86] S. Bhatt, F. Chung, T. Leighton, and A. Rosenberg, "Optimal simulations of tree machines", 27th annual symposium on foundations of computer science, pp274-282. IEEE, Oct. 1986.

[Bha 88] S. N. Bhatt, Fan R. K. Chung, and J. W. Hong,"Optimal simulations by butterfly networks", Proc. of the twenties annual ACM symposium on theory of computing, pp192-204, May 1988.

[Bha 91] S. Bhatt, F. Chung, T. Leighton, and A. Rosenberg, "Efficient embeddings of trees in hypercubes", SIAM Journal on computing, 1991.

[Bos 93] B. Bose, B. Broeg. Y. Kwon, and Y. Ashir, " Lee distance and Topological properties of k-ary n-cubes", submitted to Supercomputing 93.

[Cha 88] M. Y. Chan, "Dilation-2 embeddings of grids into hypercubes",Proc. of Int. Conference on Parallel Processing, pp.295-298, Aug.1988.

[Cha 89] M. Y. Chan, "Embedding of d-dimensional grids into optimal hypercubes", Proc. of the 1989 ACM symposium on parallel algorithms and architectures, pp52-57, June. 1989.

[Cha90] Mee yee Chan and Shiang-Jen Lee, "Falult Tolerant Embedding of Rings in the Hypercube", Proc. of 5th Distributed memory computing, pp.834-838, 1990.

[Cha1 91] M. Y.Chan, "Embedding of grids into optimal hypercubes", SIAM journal of computing, Vol. 20, No. 5, pp834-864, Oct. 1991.

[Cha91] M. Y. Chan ,and S-J Lee, " On the existence of Hamiltonian Circuits in Faulty Hypercubes", Siam J. Disc. Math. Vol. 4, No. 4, pp511-527, Nov 1991.

[Cha2 91] M. Y. Chan, and S. J. Lee, " Distributed fault-tolerant embeddings of rings in hypercubes", Journal of Parallel and Distributed Computing, Vol. 11, pp 63-71, 1991.

[Che87]M. S. Chen and K. G. Shin, " Processor Allocation in an N-Cube Multiprocessor Using Gray Code", IEEE Trans on Computers,Vol.C-36, No.12, Dec 1987.

[Che 90] C. Y. Roger Chen and Yeh-Ching Chung, "Embedding Networks With Ring Connections in Hypercube Machines", Proc. of. Int Conference on Parallel Processing, pp.327-334, 1990.

[Dal 89] W. J. Dally et al., "The J-Machine: A fine-grain concurrent computer", Proc. IFIP Congress, 1989.

[Dal 91] W. J. Dally et al., "The J-Machine: System support for actors", in Actors: Knowledge-Based Concurrent Computing, Hewitt and Agha, Eds. Cambridge, MA: MIT Press, 1991.

[Dek81] E. Dekel, D. Nassimi, and S. Sahni, " Parallel Matrix and Graph Algorithms", SIAM J.Computing, pp.657-675, Nov. 1981.

[Des 78] A. M. Despain and D. A. Patterson, " X-TREE : A Tree Structured Multiprocessor Computer Architecture", Proc.of the Fifth Annual Symposium on Computer Architecture, Vol.6, No.7, pp.144-151, 1978.

[Dut91] S. Dutt and J. P. Hayes, "Subcube Allocation in Hypercube Computers", IEEE Trans. on Computers, Vol. 40, No. 3, March 1991.

[Efe 91] K. Efe, "Embedding mesh of trees in the hypercube", Jounral of parallel and distributed computing, Vol. 11, pp222-230, 1991.

[Fel 91] R. Feldmann and W. Unger, "The cube connected cycle network is a subgraph of the butterfly network", Technical report, University of Paderborn, 1991.

[Gok78] L. R. Goke and G. L. Lipovski, " Banyan Networks for Partitioning Multiprocessor Systems", Proc. of Fifth Annual Symposium on Computer Architecture, pp.21-28, Dec.1978.

[Goo81] J. R. Goodman and C. H. Sequine, " Hypertree: A Multiprocessor Interconnection Topology", IEEE Trans. on Computers, Vol. C-30, No.12, pp.923-933, Dec.1981.

[Gre 90] D. S. Greenberg, L. S. Heath, and A. L. Rosenberg, " Optimal embeddings of the butterfly-like graphs in the hypercube", Mathematical systems theory, Vol. 23, No. 1, pp61-77, 1990.

[Gre 90] D. S. Greenberg, and S. n. Bhatt, "Routing multiple paths in hypercubes",Proc. of annual ACM symposium on parallel algorithms, and architectures", pp.45-54, 1991.

[Gup 88] A. K. Gupta, and S. E. Hambrusch " Embedding complete binary trees into butterfly networks", Dept of computer science, Purdue University, Nov. 1988.

[Her75] I. N. Herstein, "Topics in Algebra", John Wiley & Sons, Inc. 1975

[Ho 87] C. T. Ho and S. L. Johnsson. " On the embedding of arbitrary meshes in boolean cubes with expansion two dilation two", Proc. of Int. Conference on Parallel Processing, pp.188-191, 1987.

[Ho 90] C. T. Ho, and S. L. Johnsson, "Embedding three-dimensional meshes in boolean cubes by graph decomposition", Proc. of Int. Conference on Parallel Processing, pp.319-326, 1990.

[Ho 90] C. T. Ho, and S. L. Johnsson, "Embedding meshes in boolean cubes by graph decomposition", Jounral of parallel and distributed computing, Vol. 8, pp325-339, 1990.

[Int 87] INTEL scientific computers, Direct connect^{tm} routing solves node communication challenge, iSCurrents, pp. 5-6 , 1987.

[Joh 88] S. L. Johnsson, and C.T Ho, " Optimum Broadcasting and Personalized Communication in Hypercubes", IEEE Trans on Computers, Vol.38, No 9, pp. 1249-1268, 1988.

[Kos 88] R. Kosaraju, and M. Atallah, "Optimal simulations between mesh-connected arrays of processors", Journal of the ACM, Vol. 35, No. 3, pp635-650, July. 1988.

[Lei 89] T. Leighton, M. Newman, A. Ranade, and E. Schwabe, "Dynamic tree embeddings in butterflies and hypercubes",Proc. of the 1989 ACM symposium on parallel algorithms and architectures, pp224-234, June. 1989.

[Lei 92] T. Leighton, "Introduction to Parallel algorithms and Architectures : Arrays. Trees. Hypercubes", Morgan Kaufmann Publishers, 1992.

[Liu 89] W. Liu, T. H. Hildebrant, and R. Cavin, "Hamiltonian cycles in the shuffle-exchange network", IEEE Trans. on Computers, Vol.38, No.5, pp.745-750, May.1989.

[Liu 91] H. Liu, and S. H. Huang, " Dilation-6 embeddings of 3-dimensional grids into optimal hypercubes", Proc. of Int. Conference on Parallel Processing, pp.250-254, 1991.

[Ma 87] Y. E. Ma, and L. Tao, "Embeddings among toruses and meshes", Proc. of Int. Conference on Parallel Processing, pp.178-187, 1987.

[Mad 90] M. Nigam, S. Sahni, and B. Krishnamurthy, "Embedding hamiltonians and hypercubes in star intercinnection graphs", Proc. of Int. Conference on Parallel Processing, pp.III340-343, 1990.

[Mel 90] R. G. Melhem, G. Y. Hwang, "Embedding rectangular grids into square grids with dilation two",IEEE Trans. on Computers, Vol.39, No.12, pp.1446-1455, Dec.1990.

[Mon 88] B. Monien, and H. Sudborough, "Simulating binary trees on hypercubes", `Proc. of 3rd Aegean workshop on computing: VLSI algorithms and architectures, vol 319 of lecture notes in computer science, pp170-180, Springer-Verlag, july. 1988.

[Pet 72] W. W. Peterson, and E. J. Weldon, "Error Correcting Codes",MIT Press, pp41, 1972.

[Pre81] U. Premkumar, " A theoretical Basis for the Analysis and Partitioning of Regular SW Banyans", Ph.D. Thesis, University of Texas at Austin, Aug.1981.

[Pre 81] F.Preparata and J. Vuillemin, "The cube-connected cycles: a versatile network for parallel computation", Communication of the ACM, Vol 24, PP 300-309, May 1981.

[Pro 88] F. J. Provost, and R. Melhem, "Distributed Fault Tolerant Embeddings of Binary Trees and Rings in hypercubes", Proc. International Workshop on Defect and Fault Tolerance in VLSI Systems, 1988.

[Saa 88] Y. Saad and M. H. Schultz, "Topological Properties of Hypercubes", IEEE Trans. on Computers, pp. 867-872, July 1988

[Sch 90] E. Schwabe, " On the computational equivalence of hypercube-derived networks", Proc. of the 2nd annual ACM Symposium on parallel algorithms and architectures, pp388-397, July. 1990.

[Sch 91] E. Schwabe, "Efficient embeddings and simulations for hypercube networks", phD thesis, Massachusetts Institute of Technology, June. 1991.

[Seil 88] C.L. Seitz et al., "The architecture and programming of the Ametek Series 2010 Multicomputer", Proc. Third Conf. Hypercube Concurrent Comput. Appl., ACM, PP. 33-37, Jan. 1988.

[Sei2 88] C.L. Seitz et al., "Submicron systems architecture project semiannual technical report", Caltec Computer Science Tech. Rep., Caltes-CS-TR-88-18, p. 2, PP. 11-12, Nov. 1988.

[Seq79] C. H. Sequin. "Single-chip computers, the new VLSI building blocks.", Proc. of VLSI conf., California Inst. of Technol., Pasadena, CA, Jan. 22-23, 1979.

[Swa 81] M. N. S. Swamy and K. Thulasiraman, " Graphs, Networks, and Algorithms", Wiley Interscience Publication, 1981.

[Tuc 88] L. W. Tucker and G. G. Robertson, "Architecture and applications of the Connection machine", IEEE computer, Vol 21, pp 26-28, 1988

[Tze 90] N. F. Tzeng, H. L. Chen, and P. J. Chuang, "Embeddings in incomplete hypercubes", Proc. of Int. Conference on Parallel Processing, pp.325-339, 1990.

[Var 91] R. Varadarajan, "Embedding shuffle networks in hypercubes", Jounral of parallel and distributed computing, Vol. 11, pp252-256, 1991.

[Wag 89] A. Wagner, " Embedding arbitrary binary trees in a hypercube", Jounral of parallel and distributed computing, Vol. 7, pp503-520, 1989.

[Wan] A. Wang, and R. Cypher, "Fault-Tolerant embeddings of rings, meshes, and tori in hypercubes", Manuscript.

[Woe 89] W. K. Chen, F. M. Stallmann, and E. F. Gehringer, "Hypercube Embedding Heuristics: An Evaluation", International Journal of Parallel Programming, Vol. 18, No. 6, pp505-549, 1989

[Wu 85] A. Y. Wu, " Embedding of tree networks into hypercubes", Jounral of parallel and distributed computing, Vol. 2, pp238-249, 1985.

[Yan] P. J. Yang, S. B. Tien, and C. S. Raghavendra, " Embedding of rings and meshes onto faulty hypercubes", Msnuscript.

[Yaa 93] A. A. Yaagoub, " Properties and Communication Algorithms for K-ary n-cube Interconnection Networks", Oregon State University, 1993.

[You1 90] A. Youssef and B. Narahari , " The Banyan-Hypercube Networks", IEEE Trans. on Parallel and Distributed Systems, Vol.1, No.2, pp.160-169, April.1990.

[You2 90] A. Youssef and B. Narahari, "Topological Properties of Banyan-Hypercube Networks", Proc. of the 2th Symposium on the Frontiers of Massively Parallel Computation, pp.324- 332, Oct. 1990.

[You3 90] A. Youssef, "Product Networks: A Unified Theory of Fixed Interconnection Networks," Technical Report GWU-IIST-90-38, The George Washington University, Dec. 1990.