International Journal of Information Technology and Knowledge Management
January June 2009, Volume 2, No. 1, pp. 87-90

DESIGN AND COMPARISON OF TORUS EMBEDDED HYPERCUBE WITH MESH EMBEDDED HYPERCUBE INTERCONNECTION NETWORK

N. Gopalakrishna Kini*, M. Sathish Kumar & Mruthyunjaya H. S.*****

This paper analyzes a product network generated from torus and hypercube networks known as Torus embedded hypercube scalable interconnection network suitable for parallel architecture. It is shown here that how good a Torus embedded Hypercube interconnection network can be compared to the existing Mesh embedded hypercube interconnection network with minor modifications in its architecture. A complete design analysis and comparison is given using various network parameters. The analysis and computational results show that the Torus embedded hypercube interconnection network is highly scalable and more efficient in terms of communication.

Keywords: Hypercube network, Torus network, Mesh embedded hypercube network, Scalability, Network parameters.

1. Introduction

The dominating features of hypercube network are small network diameter, high connectivity and simple routing [1]. On the other hand, Mesh is a network with constant node degree in its internal nodes where as Torus network has constant node degree with all its nodes [1]. The advantages of these networks can be combined by embedding torus with hypercube to give rise to embedded architecture known as Torus embedded hypercube scalable interconnection network. In this paper, we discuss about the torus embedded hypercube network. More details on this can be found in [2]-[4]. Also it has been proved how efficient a Torus embedded Hypercube interconnection network compared to the existing Mesh embedded hypercube interconnection network.

2. Theoretical Consideration

While combining the torus and the hypercube network, several concurrent torus networks are used in the architecture [2]-[5]. Let $l \times m$ be the size of several concurrent torus networks and N be the number of nodes connected in the hypercube. Nodes with identical positions in the torus networks will form a group of N number of nodes and hence the resultant torus embedded hypercube network having a size of (l, m, N). The nodes in the network can be addressed with three components; row number i and column number j of torus appended with the address of node k of hypercube. Hence, a (l, m, N)–torus embedded hypercube network will have $l \times m \times N$ number of nodes and a node will be addressed as (i, j, k) where $0 \le i < l$, $0 \le j < m$ and $0 \le k < N$.

Combining the data routing functions of torus and hypercube will provide with the routing functions of the torus embedded hypercube [2], [4] as in (1)-(5).

$$T_{h1}(i, j, k) = (i, (j + 1) \bmod m, k) \quad (1)$$

$$T_{h2}(i, j, k) = (i, (m + j - 1) \bmod m, k) \quad (2)$$

$$T_{h3}(i, j, k) = ((i + 1) \bmod l, j, k) \quad (3)$$

$$T_{h4}(i, j, k) = ((l + i - 1) \bmod l, j, k) \quad (4)$$

$$T_{C_d}(k_{n-1} \ldots k_{d+1}\, k_d\, k_{d-1} \ldots k_0) = (k_{n-1} \ldots k_{d+1}\, \overline{k_d}\, k_{d-1} \ldots k_0) \quad (5)$$

for $d = 0, 1, \ldots n-1$ where k_j for ($j = 0$ to $n-1$) is the binary representation of node address k and $n = log_2(N)$ where N is the total number of nodes in the hypercube.

The address of individual node is represented by n-bit binary vector. A link will exist between two nodes where the addresses of these two nodes differ exactly by one bit [6]. For a (2, 2, 8)-torus embedded hypercube network, a node with a five bit address has its left most bit representing row number, the next bit representing column number and the remaining least significant bits representing the address of a node in the hypercube as shown in Fig. 1. In the diagram the ring connections of row/column of each torus are not shown for simplicity and without that the network will be a (2, 2, 8)-mesh embedded hypercube network. A wraparound connection is done along each row/column of the mesh if they have same label in the diagram to deduce it to (2, 2, 8)-torus embedded hypercube network.

* Dept. of Computer Science and Engineering, Manipal Institute of Technology (Manipal University), Manipal, Karnataka, India, E-mail: *ng.kini@manipal.edu*

* School of Eecs, Seoul National University, Seoul, South Korea, E-mail: *mskuin@yahoo.com*

*** Dept. of Electronics and Communication, Manipal Institute of Technology (Manipal University), Manipal, Karnataka, India, E-mail: *mruthyu.hs@manipal.edu*

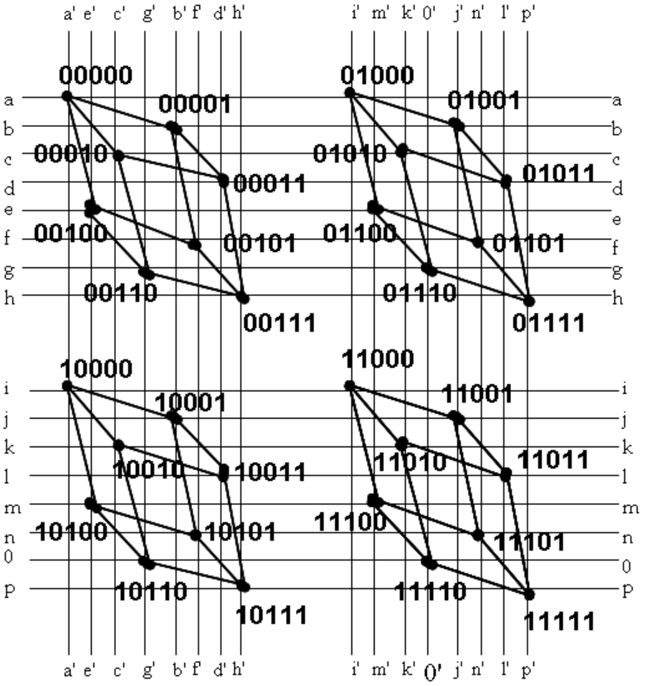

Figure 1: A (2, 2, 8) - Torus Embedded Hypercube Network

3. SCALABILITY

We define scalability of a network as the property by which the size of the system can be expanded with nominal changes in the existing configuration provided that system expansion results with improvement in performance.

Scalability can be achieved in two ways [2]. Firstly, the dimension of the hypercube can be increased by keeping the size of concurrent mesh/torus same but increasing the number of concurrent meshes/torus accordingly. Secondly, dimension of mesh/torus is expanded by keeping the size of the hypercube constant. Since node configuration is not required, scaling up the system using the latter method is preferable over the former.

4. COMPARISON RESULTS OF MESH EMBEDDED HYPERCUBE WITH TORUS EMBEDDED HYPERCUBE NETWORK

The performance evaluation of the torus embedded hypercube network is done using network parameters such as node degree, network diameter, total number of links in the network and topological network cost. The definitions of these network parameters can be found in [2]-[4].

4.1 Node Degree Analysis

It may be noted that the node degree has to be as least as possible because if the number of links to a node is increased, the number of I/O ports also increase. The increase in the node degree can result in a larger penalty on the network cost.

Table 1 and Fig. 2 give the comparison of *node degree* of Mesh Embedded Hypercube with Torus Embedded Hypercube Network. There is no difference observed with these two networks with respect to node degree analysis. It should be noted that in mesh embedded hypercube network a node with maximum link complexity is considered.

Table 1
Comparison of Node Degree

No. of processors / *Network type*	*512*	*1024*	*2048*	*4096*	*8192*	*16384*
(16,16,N)– Mesh embedded Hypercube)	5 N = 2	6 N = 4	7 N = 8	8 N = 16	9 N = 32	10 N = 64
(*l*, *m*, 16) – Mesh embedded Hypercube)	8	8	8	8	8	8
(16, 16, N) – Torus embedded Hypercube	5 N = 2	6 N = 4	7 N = 8	8 N = 16	9 N = 32	10 N = 64
(*l*, *m*, 16) – Torus embedded Hypercube)	8	8	8	8	8	8

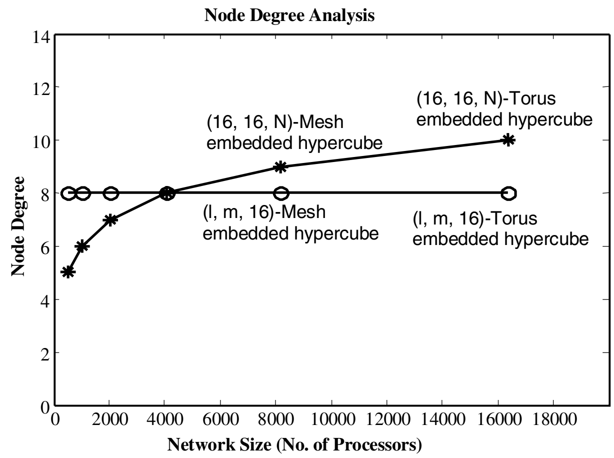

Figure 2: Node Degree Analysis

4.2 Network Diameter Analysis

The diameter of a network determines maximum number of hops an average message takes to reach to its destination. If the diameter is too large, it implies that a large number of nodes will have to be busy to get connected to the destination node. This in turn reduces the performance of the whole system. From computational analysis given in Table 2 and Fig. 3, as far as the network diameter is concerned, it is obvious that the torus embedded hypercube network is much superior than mesh embedded hypercube network as the torus embedded hypercube network needs lesser network diameter to get connected between a source node and a destination node.

4.3 Number of Links Analysis

Table 3 shows the number of links with respect to the scaling of the parallel architecture for the two networks considered.

Table 2
Comparison of Network Diameter

No. of processors / Network type	512	1024	2048	4096	8192	16384
(16,16,N)– Mesh embedded Hypercube)	31 N = 2	32 N = 4	33 N = 8	34 N = 16	35 N = 32	36 N = 64
(*l*, *m*, 16) – Mesh embedded Hypercube)	14	18	26	34	48	66
(16, 16, N) – Torus embedded Hypercube	17 N = 2	18 N = 4	19 N = 8	20 N = 16	21 N = 32	22 N = 64
(*l*, *m*, 16) – Torus embedded Hypercube)	10	12	16	20	26	36

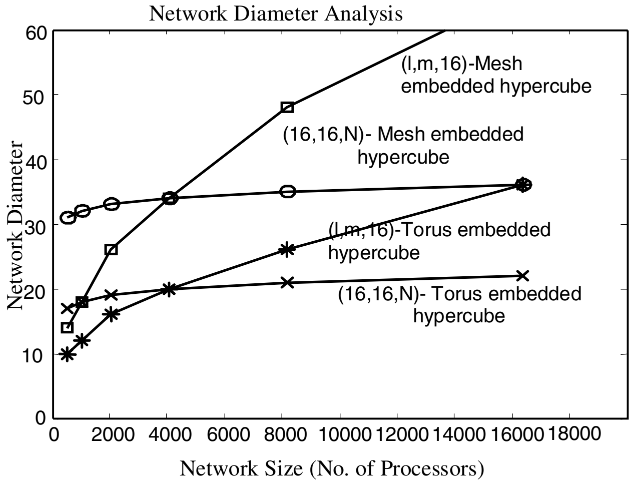

Figure 3: Network Diameter Analysis

As it is seen from the enhancement of architecture of mesh embedded hypercube, it is natural that the torus embedded hypercube will need more number of links for a system with specified number of processors. From Fig. 4, it can be observed that torus embedded hypercube offers larger number of links since every node of torus embedded hypercube configuration is with a link complexity of four.

Table 3
Comparison of Number of Links

No. of processors / Network type	512	1024	2048	4096	8192	16384
(16,16,N)– Mesh embedded Hypercube)	1216	2944	6912	15872	35840	79872
(*l*, *m*, 16) – Mesh embedded Hypercube)	1877	3840	7851	15872	32055	64512
(16, 16, N) – Torus embedded Hypercube	1280	3072	7168	16384	36864	81920
(*l*, *m*, 16) – Torus embedded Hypercube)	2048	4096	8192	16384	32768	65536

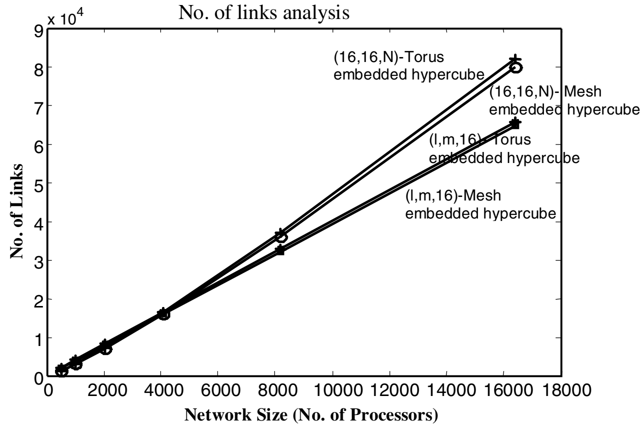

Figure 4: Number of Links Analysis

4.4 Network Cost

The topological network cost analysis result is given in Table 4 and Fig. 5. It is seen that the torus embedded hypercube network will have low network cost. Though the network diameter is found to be increasing in $(l, m, 16)$ –torus embedded hypercube network , it has to be noted that the torus embedded hypercube has better values for network cost as the system is scaled up.

Table 4
Comparison of Network Cost

No. of processors / Network type	512	1024	2048	4096	8192	16384
(16,16,N)– Mesh embedded Hypercube)	37696	94208	228096	539648	1254000	2875392
(*l*, *m*, 16) – Mesh embedded Hypercube)	26278	69120	204126	539648	1538640	4257792
(16, 16, N) – Torus embedded Hypercube	21760	55296	136192	327680	774144	1802240
(*l*, *m*, 16) – Torus embedded Hypercube)	20480	49150	131072	327680	851968	2359296

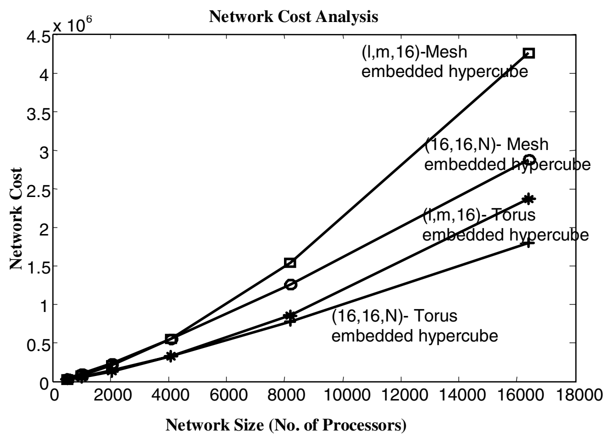

Figure 5: Network Cost Analysis

5. CONCLUSION AND FUTURE WORK

We have analyzed a torus embedded hypercube interconnection network and compared its network parameters with mesh embedded hypercube interconnection network for a parallel architecture. It is necessary to come up with a network that is scalable, constant node degree, minimum network diameter and a minimum topological cost. All afore mentioned requirements are met by the torus embedded network and hence it can supersede the mesh embedded network. The results show that torus embedded hypercube network is much faster than the mesh embedded hypercube in terms of communication. Hence this network could be chosen as interconnection network for parallel architecture.

The future work involved is to develop a parallel algorithm for torus embedded hypercube interconnection network. The principle of existing parallel algorithms for the mesh embedded hypercube network, with minor modifications, can be applied to the torus embedded hypercube network.

References

[1] K. Hwang, "Advanced Computer Architecture: Parallelism, Scalability, Programmability," New York McGraw-Hill, (1993).

[2] A. Louri and H. Sung, "An Optical Multi-Mesh Hypercube: A Scalable Optical Interconnection Network for Massively Parallel Computing," *IEEE J. Lightwave Technology*, **12**, (Apr. 1994), 704–716.

[3] N. Gopalakrishna Kini, M. Sathish Kumar and Mruthyunjaya H. S., "A Torus Embedded Hypercube Scalable Interconnection Network for Parallel Architecture," *E_proceedings of IEEE International Advance Computing Conference (IACC'09)*, Patiala, India, Mar 6-7, 2009.

[4] N. Gopalakrishna Kini, M. Sathish Kumar and Mruthyunjaya H. S., "Analysis and Comparison of Torus Embedded Hypercube Scalable Interconnection Network for Parallel Architecture," *International Journal of Computer Science and Network Security*, **9**, (1), (Jan. 2009), 242–247.

[5] J. F. Fang, J. Y. Hsiao and C. Y. Tang, "Embedding Meshes and TORUS Networks onto Degree-four Chordal Rings," *IEE Proc.-Comput. Digit. Tech.*, **145**, (2), (March 1998).

[6] Kim Jong-Seok, Lee Hyeong-Ok and Heo Yeong-Nam, "Embedding Among HCN(n, n), HFN(n, n) and Hypercube," *Proceedings of Eighth International Conference on Parallel and Distributed Systems (ICPADS)-2001*, (26-29 June 2001), 533–540.