# A Torus Assignment for an Interconnection Network Recursive Diagonal Torus

Qin Fan[†] Yulu Yang[†]
† Dept. of Computer and System Science, Nankai University,
94,Weijin Road,Tianjin,300071,China
Tel:+86-22-2350-3547, Fax:+86-22-2350-9054
E-mail:{yangyulu,parallel}@public.tpt.tj.cn

Akira Funahashi[‡] Hideharu Amano[‡]
‡ Dept. of Computer Science, Keio University,
3-14-1 Hiyoshi, Kouhoku-ku, Yokohama, 223 Japan
Tel:+81-45-560-1063, Fax:+81-45-560-1064
E-mail:{funa,hunga}@aa.cs.keio.ac.jp

## Abstract

*Recursive Diagonal Torus (RDT) is a class of interconnection network consisting of recursively overlaid two-dimensional square diagonal tori for massively parallel computers with up to* $2^{16}$ *nodes. Connection structures of the RDT vary according to the assignment of upper rank diagonal tori into a node. Although traditional simple assignment called RDT(2,4,1)/*$\alpha$ *shows enough performance under the uniform traffic, the congestion of low rank tori degrades the performance when local communication is dominant.*

*In this paper, RDT(2,4,1)/*$\beta$ *torus assignment is proposed, focusing on improving the performance for local communication. With a simplified simulation algorithm, result shows that RDT(2,4,1)/*$\beta$ *improves the average distance compared with RDT(2,4,1)/*$\alpha$ *assignment when considering local area.*

**Keywords:** interconnection network, routing algorithm

## 1. Introduction

Highly parallel computers are commonly organized as a set of nodes consisting of a processor and memory that communicate over an interconnection network. Thus the communication network topology plays a critical role in such an architecture.

Through the history of multicomputers, the mesh network, featured for its small degree and fitness for various scientific calculations including flow dynamics, QCD, and structural analysis, has been widely used in this area. Moreover, many parallel computation algorithms have been accumulated and refined on the current machines with mesh structure.

Recursive Diagonal Torus (RDT) [5] is a class of networks, which carries on the advantages of mesh structure and greatly improves the performance of mesh when the number of nodes reaches to ten thousand nodes.

The RDT consists of recursively structured mesh (torus) connection. It supports smaller diameter and degree than that of the hypercube if the number of nodes is between 1000 and 10000. Since the native network called the Perfect RDT is unrealistic for its large degree, the concept of torus assignment is introduced to make a practical extension of RDT.

A simple assignment called RDT(2,4,1)/$\alpha$ has been proposed[5], and the router chip based on the assignment was implemented[1] and used in a real machine called JUMP-1[2]. Although this assignment shows enough performance under uniform traffic, the congestion of low rank tori degrades the performance when local communication is dominant. In this paper, RDT(2,4,1)/$\beta$ torus assignment is proposed, focusing on improving the performance for local communication.

In Section 2, the interconnection structure of the RDT is briefly introduced. A new torus assignment RDT(2,4,1)/$\beta$ is proposed in Section 3, and the routing algorithms are presented in Section 4. In Section 5, a comparison is made between RDT(2,4,1)/$\beta$ and RDT(2,4,1)/$\alpha$ on local area message transfer over a large size of RDT network.

## 2. Interconnection Network: RDT

### 2.1. Definition of the RDT

The name Recursive Diagonal Torus (RDT) itself expresses clearly its characteristics. In other words, this novel class of network is composed of a series of recursively structured mesh (torus) connections with increasing size in the diagonal directions.

First, a two-dimensional square mesh (torus) will serve as the basis of RDT.

**Base torus**
The base torus is a two-dimensional square array of nodes each of which is numbered with a two-dimensional number as follows:

$$\begin{array}{ccccc}
(0,0) & (1,0) & (2,0) & \cdots & (\mathrm{N}-1,0) \\
(0,1) & (1,1) & (2,1) & \cdots & (\mathrm{N}-1,1) \\
(0,2) & (1,2) & (2,2) & \cdots & (\mathrm{N}-1,2) \\
\vdots & & & & \\
(0,\mathrm{N}-1) & (1,\mathrm{N}-1) & (2,\mathrm{N}-1) & \cdots & (\mathrm{N}-1,\mathrm{N}-1)
\end{array}$$

where $N = n^k$. The $n$ and $k$ are natural numbers. The torus network is formed with four links between node $(x, y)$ and neighboring four nodes:

$$(\text{mod}(x \pm 1, N),\, y) \textit{ and } (x,\, \text{mod}(y \pm 1, N))$$

This base torus is also called the rank-0 torus.

**Upper rank tori**
The upper tori are formed in a recursive way. Four links can be added between node $(x, y)$ and nodes $(x \pm n, y \pm n)$. Thus, these four new links form a new torus network on the basis of rank-0 torus with a direction of 45 degrees to the original torus. Then on what is called rank-1 torus, another torus network is formed in the same manner, and yet another. In one word, the rank-($r$+1) torus is formatted on the rank-$r$ torus to provide bypass links in the diagonal direction.

Figure 1 shows rank-1 and rank-2 tori when $n$ is set to be 2.

The rank-($r$+1) torus is called an upper rank torus based on the rank-$r$. $n$ is called **cardinal number**.

**RDT**
Recursive Diagonal Torus RDT($n, R, m$) is a class of networks in which each node has links to form base torus(rank-0) and $m$ upper tori (the maximum rank is R) with the cardinal number $n$. According to this definition, the degree of the RDT($n, R, m$) can be shown as: $4(m + 1)$.

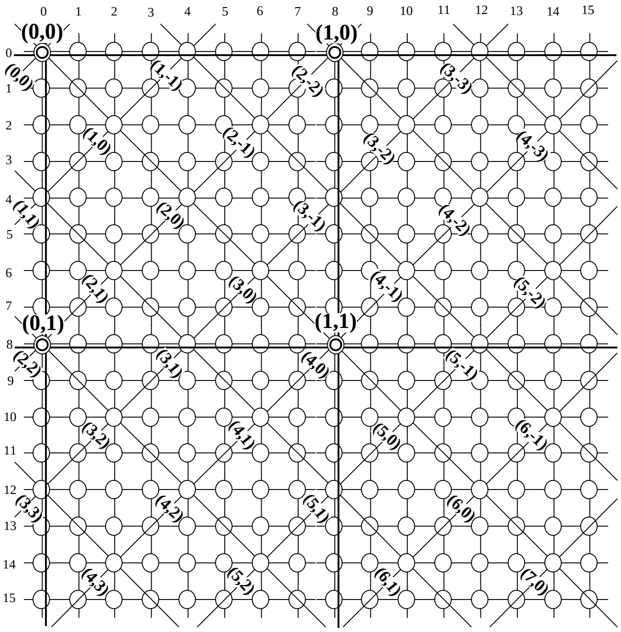

**Figure 1. Upper rank tori**

**Perfect RDT**
A network in which every node has links to form all possible upper rank tori is called the Perfect RDT (**PRDT(**$n, R$**)**) where $n$ is the cardinal number and $R$ is the maximum rank.

Although the PRDT is unrealistic because of its large degree ($4(R + 1)$), it is an important basis for establishing message routing algorithms of the RDT theoretically.

### 2.2. Torus assignment

The change of the parameters of RDT ($n$ or $m$) could provide a variety of RDT structures which will apply to different applications or provide various characteristics. For easy inheritance of the previous accomplished routing algorithms based on binary tree or cube, the cardinal number $n$ is set to be 2.

As for parameter $m$, the RDT with a large $m$ would require too much hardware, since each upper torus requires four links. If we consider a system with ten thousand nodes(for example, an array of $128 \times 128$ or $256 \times 256$ nodes), and $m$ is set to be 1(degree=8), the maximum rank of upper tori is 4. Therefore, the RDT(2,4,1) is mainly treated here.

In RDT(2,4,1), the structure varies with different assignments for upper rank tori to each node. This assignment is called the *torus assignment*. In order to assign tori clearly, we need to define the method of identifying the upper rank tori.

**Identification of upper torus**

An upper torus is represented with its node number following the $2n \times n$ array fraction ($n$ is the cardinal number) of the base torus.

A rank-($r$+1) torus is represented with the node number of the above array in the rank-r torus.

For example, the rank-1 torus in Figure 2 is called (1,0) torus. The rank-2 torus in Figure 2 is (1,0) torus formed on (0,0) torus, and thus, called ((0,0)(1,0)) torus. ((0,0)(*,*)) represents all rank-2 tori on the rank-1 torus (0,0).

Various torus assignment strategies can be selected considering the traffic of the network. If the local traffic is large, the number of nodes which have low ranks should be increased.

A relatively simple torus assignment is proposed first.

**RDT(2,4,1)/$\alpha$**

Here, RDT(2,4,1) with the following torus assignment is called the RDT(2,4,1)/$\alpha$[14].

- $rank$-1: (1,0) , (3,1)
- $rank$-2: ((0,0)(*,*)) , ((2,1)(*,*))
- $rank$-3: ((1,1)(*,*)(*,*)) , ((3,0)(*,*)(*,*))
- $rank$-4: ((0,1)(*,*)(*,*)(*,*)) , ((2,0)(*,*)(*,*)(*,*))

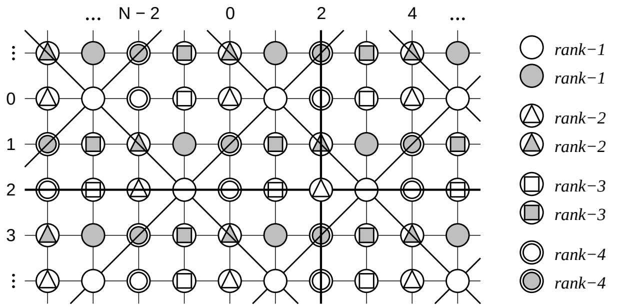

**Figure 2. Torus assignment for the RDT(2,4,1)/$\alpha$**

In the above assignment, as what is previously defined, each node has eight links altogether, four of which are for the base torus(rank-0), and the rest for upper torus(rank-1 to rank-4). Since the cardinal number $n$ is set to 2, 8 upper independent rank-1 tori can be formed on the base torus.

Among these 8 tori, two of them$(1,0)(3,1)$ are used just as the rank-1 tori. Two others $(0,0)(2,1)$ are used to form rank-2 tori, still two others for rank-3 tori, and the rest for rank-4 tori. In this way, each node with an upper rank torus would be attached to the neighboring nodes that have other three upper ranks. Such features will be advantageous to the routing algorithm by reducing the diameter and average steps between nodes, since any packet can be passed to any rank torus within a single message transfer between neighboring nodes.

## 3. A new torus assignment RDT(2,4,1)/$\beta$

RDT is featured for its good performance in a MPC with more than ten thousands nodes when there is an uniform access probability to local and remote nodes. However, for many practical applications, local message communication happens much more often than remote ones. That is to say, one concerns more about how soon the message could be sent to his near neighbors than to a point that is far away. Consequently, when local message communication is dominant in such machines, the small number of low level rank tori in RDT(2,4,1)/$\alpha$ assignment does not fit.

In order to adjust to such application requirements, and not to increase much of the diameter, rank-1 and rank-2 links are increased in number with corresponding decrement of rank-3 and rank-4 links. That is because more rank-1 and rank-2 links would enable more adaptable routing of local message transfer. A collection of torus assignments meet the above description. One of them is chosen and named RDT(2,4,1)/$\beta$, since it could better implement the floating vector algorithm and minimize rank-0 steps.

**Definition of RDT(2,4,1)/$\beta$**

According to the previous definition of RDT(2,4,1)/$\alpha$, the following torus assignment shows the RDT(2,4,1)/$\beta$:

- $rank$-1: (1,0),(2,1),(0,1)
- $rank$-2: ((0,0)(*,*)),((3,1)(*,*)),((1,1)(*,*))
- $rank$-3: ((3,0)(*,*)(*,*))
- $rank$-4: ((2,0)(*,*)(*,*)(*,*))

Just as in RDT(2,4,1)/$\alpha$, each node has eight links, four of which are for the base(rank-0) torus, and the rest are for rank-1 to rank-4 torus. However, unlike RDT(2,4,1)/$\alpha$, more nodes are connected to form lower ranks including rank-1 and rank-2. This assignment improves the performance of communication between nodes that are near to each other, without much effect to those that are far between.

In this torus assignment, one could see that not all the ranks are connected to each node as what is shown in RDT(2,4,1)/$\alpha$. The neighboring nodes that are directly connected to each node only cover three ranks. However, from the viewpoint of probability, most of the cases lacked are

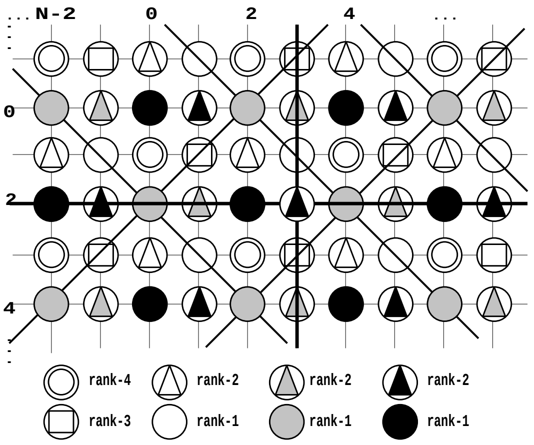

**Figure 3. Torus assignment for the RDT(2,4,1)/$\beta$**

links that have remote connections, such as rank-3, and rank-4.

On the other hand, the number of links that have local connections is increased. To be more specific, each node has connections to three links that has rank-1 or rank-2 connections. Obviously, this would improve the message routing when considering local area.

## 4. Message Routing Algorithms

For RDT(n,R,m), the routing algorithm can be divided into two parts:

- Vector routing for the Perfect RDT that is assignment independent[5]
- Floating vector routing to fit different torus assignments

### 4.1. Revised floating vector routing for RDT(2,4,1)/$\beta$

The vector routing algorithm for the PRDT must be modified so as to enable message routing on different torus assignments of RDT(2,4,1), since only one upper torus is connected to each node in RDT(2,4,1)/$\alpha$ and RDT(2,4,1)/$\beta$.

Additional steps must be taken in order to route the message through ranks that are different from what is connected to the source node, and are required by the pre-determined vectors for the PRDT. For this purpose, the floating vector routing is proposed. In this method, packets are routed in a way to minimize the routing with rank-0 torus, according to the vector routing on the PRDT.

The vector reduction described above is done in the source node. As a result, both the diameter and average distance are minimized. The algorithm for the floating vector routing is as follows:

1. If the node $i$ $(i_x, i_y)$ has the rank-$r$ torus and $(v_{rh}, v_{rv}) \neq (0,0)$, send the packet according to $(v_{rh}, v_{rv})$. Otherwise, choose rank $p$ whose $(v_{ph}, v_{pv}) \neq (0,0)$. If there are no upper ranks to be routed, goto (4).

2. Choose the node $j$ $(j_x, j_y)$ which has the rank-$p$ torus and minimizes $|(j_x - i_x) - v_{0h}| + |(j_y - i_y) - v_{0v}|$. This node selection is done by the local table reference.

3. Send the packet to node $j$ through links for the rank 0 torus.
   $(v_{0h}, v_{0v}) \leftarrow (v_{0h} + (j_x - i_x), v_{0v} + (j_y - i_y))$. When the packet reaches to the node $j$, replace $i$ by $j$ $(i = j)$ and goto (1).

4. Send the packet according to $(v_{0h}, v_{0v})$ through links of the rank 0 torus.

This routing method is torus assignment independent. In RDT(2,4,1)/$\alpha$, only one single step of message transfer to a neighboring node is required for every node to use any rank of torus. In the newly proposed torus assignment RDT(2,4,1)/$\beta$, additional one step is required when the neighboring nodes do not have tori required in the predetermined routing. However, the possibility of such cases is small when local message transfer without using upper tori is dominant.

## 5. Performance Evaluation of RDT(2,4,1)/$\beta$

With up to $2^{16}$ nodes, RDT(2,4,1)/$\beta$ torus assignment inherits the merits of RDT(2,4, 1)/$\alpha$ with reasonable degree and comparatively small diameter, when it focuses on improving local communication performance.

### 5.1. Performance evaluation with other direct network.

First, the diameter and degree of the RDT(2,4,1)/$\beta$ are compared with other direct networks.

From Table 1, one could see that RDT(2,4,1)/$\beta$ supports smaller diameter than most of the direct networks considered. To be more specific, RDT(2,4,1)/$\beta$ provides a smaller diameter than that of the hypercube when the system reaches to 65536 nodes, a much easier way to emulate the mesh than that of De Bruijn, Kautz, Pladhan, and a recursive structure compared with Diagonal Mesh[12].

**Table 1. Diameter (degree) of direct networks**

| Number of nodes | 4096 | 65536 |
|---|---|---|
| 2D Torus | 64 (4) | 256 (4) |
| 3D Torus | 24 (6) | 48 (6) |
| Hypercube | 12 (12) | 16 (16) |
| De Bruijn[3] | 12 (4) | 16 (4) |
| Kautz[6] | 11 (4) | 15 (4) |
| Pladhan[7] | 12 (5) | 16 (5) |
| Circular omega[8] | 20 (4) | 26 (4) |
| n-Star graph[4] | 7 (6) | 8 (7) |
| CCC[9] | 21 (3) | 29 (3) |
| Hypernet[10] | 19 (5) | 17 (6) |
| Crossed Cube [13] | 7 (12) | 9 (16) |
| Midimew [11] | 46 (4) | 181 (4) |
| RDT(2,4,1)/$\alpha$ | 10 (8) | 14 (8) |
| RDT(2,4,1)/$\beta$ | 10 (8) | 14 (8) |

### 5.2. RDT(2,4,1)/$\beta$ compared with RDT(2,4,1)/$\alpha$

Diameter and average distance of RDT(2,4,1)/$\alpha$ and RDT(2,4,1)/$\beta$ are shown in the following table:

**Table 2. Diameter (average distances) of RDT(2,4,1)$\alpha$ and $\beta$**

| Number of nodes | 1024 | 4096 | 16384 |
|---|---|---|---|
| RDT(2,4,1)/$\alpha$ | 9(4.73) | 10(5.81) | 12(7.16) |
| RDT(2,4,1)/$\beta$ | 9(4.57) | 10(5.85) | 12(7.35) |

In RDT(2,4,1)/$\alpha$, the torus assignment covers every rank tori. That is, nodes that have different rank connections are equally distributed. Therefore, with floating vector routing method, diameter and average distance do not increase much.

Considering that some applications have requirement mainly on local message transfer, and that at the same time not to reduce the number of links over which messages can be sent through upper rank tori, RDT(2,4,1)/$\beta$ is proposed to enhance local communication.

In RDT(2,4, 1)/$\beta$, the number of nodes that have lower rank tori connections (rank-1, rank-2) is increased, sacrificing some upper rank(rank-3, rank-4) connections. In this way, when applying floating vector routing, the message transfer will have more choice on direction so that messages could be floated to the neighboring nodes when it needs lower rank tori, thus to reduce diameter and average distance greatly in this aspect. With a computer simulator, we compare the average diameter of RDT(2,4,1)/$\alpha$ and RDT(2,4,1)/$\beta$ under such a condition.

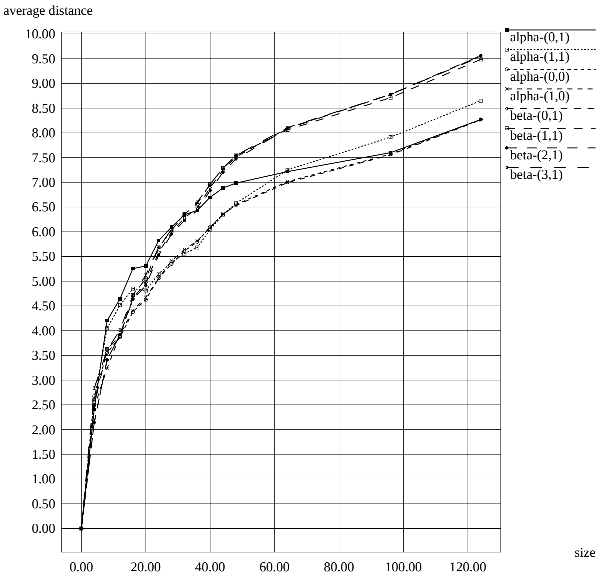

**Figure 4. A comparison between RDT(2,4,1)/$\alpha$ and RDT(2,4,1)/$\beta$**

In Figure 4, a square area with a small number of nodes, from $2 \times 2$ to $128 \times 128$, is randomly sampled from a fairly large size of RDT structure, the latter of which must be large enough to ensure the routing on the sampled square is independent. That is, the sampled square will not contain torus structure, but plain mesh instead. In the graph, the X-axis from $0$ to $128$ represents the number of nodes to make a sampling area, while the Y-axis shows the average distance calculated based on the corresponding size of area. Since the start position on RDT(2,4,1) with various ranks would end with quite different results considering local message transmission, the data are calculated on all the nodes in one assignment, i.e., altogether $8$ lines for RDT(2,4,1)/$\alpha$ and RDT(2,4,1)/$\beta$ are involved. To be more specific, in Figure 4 the line ”alpha-(0,1)” represents that the routing message starts from nodes ((0,1)(*,*)(*,*)(*,*) as what is previously defined in the definition of RDT(2,4,1)/$\alpha$, etc..

To evaluate the local performance on these two assignments respectively, the concept of weight for local or remote transmission should be added. However, the weight calculation would be different with various applications. Therefore the weight is supposed to be the same in this paper.

From Figure 4, one could see that between $4 \times 4$ and $16 \times 16$ nodes, RDT(2,4,1)/$\beta$ shows a better average performance than that of RDT(2,4,1)/$\alpha$. And from $16 \times 16$ to $32 \times 32$, some of the cases in RDT(2,4,1)/$\beta$ are better than RDT(2,4,1)/$\alpha$. Over $32 \times 32$ nodes, more upper ranks(rank-

3,rank-4) are required to route the packet, and there are less nodes that have direct connection to upper ranks. Therefore, the average distance of RDT(2,4,1)/$\beta$ within this quantity is a little bit more than that of RDT(2,4,1)/$\alpha$. But the difference in between would not be enlarged as the size of the sampled area increases. Instead, it will remain approximately a constant value. This shows that with the variation of torus assignment, routing for the local message transfer could be greatly improved.

## 6. Conclusion

A new torus assignment named RDT(2,4,1)/$\beta$ is proposed and discussed. Based on the floating vector routing algorithm, this torus assignment achieves better performance than the traditional RDT(2,4,1)/$\alpha$ assignment with same diameter and less average distance when local communication is dominant.

Thus it further proves that RDT structure can be extended to different practical models to meet various needs from all kinds of applications. Through the computer simulation, we are now investigating an effective method to show the rule of torus assignment to adjust RDT to suit different application requirements.

## 7. Acknowledgement

This research was sponsored by the Scientific Research Foundation for the Returned Overseas Chinese Scholars, State Education Ministry.

## References

[1] H. Nishi, K. Anjo, T. Kudoh and H. Amano, “The RDT Router Chip: A Versatile Router for Supporting a Distributed Shared Memory,” *IEICE TRANS INF. and SYST.*,,VOL.E80-D, NO.9,pp. 854-862, Sep.1997.

[2] H. Tanaka, Editor-in-Chief, “The Massively Parallel Processing System JUMP-1”, *IOS Press,* 1996. (ISBN 90-5199-262-9)

[3] M. R. Samatham and D. K. Pladhan, “The De Bruijn Multiprocessor Network: Versatile Parallel Processing and Sorting Network for VLSI,” *IEEE Trans. Comput.*, Vol.38, No.4, pp. 567-581, Apr. 1989.

[4] S. B. Akers, D. Harel and B. Krishnamurthy, “The Star Graph: An Attractive Alternative to the n-cube,” *Proc. ICPP ’87,* pp. 393-400, Aug. 1987.

[5] Y. Yang, H. Amano, H. Shibamura and T. Sueyoshi, “Recursive Diagonal Torus: An Interconnection Network for Massively Parallel Computers,” *Proc. the 5th IEEE Symposium on Parallel and Distributed Processing,* pp. 591-594, Dec. 1993.

[6] J. C. Bermond and C. Peyrat, “De Bruijn and Kautz Networks: A Competitor for the Hypercube,” *Hypercube and Distributed Computers*, F. Andre and J. P. Verjus, eds., North-Holland, pp. 279-293, 1989.

[7] D. K. Pladhan, “Fault-Tolerant Multiprocessor Link and Bus Network Architectures,” *IEEE Trans. on Comput.*, Vol. 34, No. 1, Jan. 1985, pp. 35-45.

[8] S. Sakai, Y. Yamaguchi, K. Hiraki, Y. Kodama and T. Yuba, “An Architecture of a Dataflow Single Chip Process,” *Proc. of the 16th International Symposium on Computer Architecture*, pp. 46-53, 1989.

[9] F. P. Preparata and J. Vuillemin, “The Cube-Connected Cycles: A Versatile Network for Parallel Computation,” *Comm. ACM,* Vol.24, No. 5 pp. 300-309, May 1981.

[10] K. Hwang and J. Ghosh, “Hypernet: A Communication-Efficient Architecture for Constructing Massively Parallel Computer,” *IEEE Trans. on Comput.*, Vol. 36, No. 12 pp. 1450-1466, Dec. 1987.

[11] R. Beivide, R. Herrada, J. L. Balcazar and A. Arruabarrena, “Optimal Distance Networks of Low Degree for Parallel Computers,” *IEEE Trans. on Comput.*, Vol. 40, No. 10, pp. 1109-1124, Oct. 1991.

[12] K. W. Tang and S. A. Padubidri, “Routing and Diameter Analysis of Diagonal Mesh Networks,” *Proc. of ICPP ’92*, pp. I143-I150, Aug. 1992.

[13] K. Efe, “The Crossed Cube Architecture for Parallel Processing,” *IEEE Trans. on Comput.*, Vol. 3, No. 5, pp. 513-524, Sept. 1992.

[14] Y. Yang and H. Amano, “Message Transfer Algorithms on the Recursive Diagonal Torus,” *IEICE TRANS INF. and SYST.*,,VOL.E79-D, NO.2,pp. 107-116, Feb.1996.