Fault Tolerance and Node Mapping Algorithm Hierarchical Graph HCN

Mihyeon Park [1], Hyeongok Lee [1]*

[1] *National Sunchon University, Suncheon, South Korea.* hangeuli@nate.com

[2] *National Sunchon University, Suncheon, South Korea.* oklee@scnu.ac.kr

Abstract: Embedding in an interconnection network maps the interconnection network G to H and analyzes the relationship between G and H. Hypercubes and toruses are widely known as interconnection networks, and various algorithms have been developed. The HCN graph is a network with a hierarchical structure to improve the network cost of hypercube. In this study, we analyze the fault tolerance and embedding properties of HCN graphs. As a result of the research, the HCN graph has the same node connectivity and degree, so it has the maximum fault tolerance property. In addition, an algorithm that can one-to-one node map the HCN structure to the torus structure was proposed. The embedding results showed that it was possible to embed the torus structure in the HCN graph at an extension rate of 3 and dilation of 1. The results of the embedding study mean that various algorithms developed in Torus can be efficiently used in the HCN structure.

Keywords: Fault tolerance, Node mapping Algorithm, HCN Graph, Embedding, Interconnection Network

1. INTRODUCTION

Today, large-capacity data processing and fast computing speed are required in the process of solving many problems in the fields of weather observation, nuclear energy, aerospace, advanced medicine, big data, and AI. Among many methods to improve computing performance, there is supercomputing that applies parallel processing techniques. A parallel processing computer connects a large number of processors through an interconnection network, enabling fast calculations and processing of large amounts of data. Research on interconnection networks has been conducted as a way to improve processing speed in parallel computer structures. An interconnection network is a structure in which a large number of processing devices are connected through communication lines. Graphs are used as mathematical models to analyze interconnection networks.

Graph theoretical properties of interconnected networks include degree, symmetry, scalability, bipartite graph, interconnected network coloring, and independent sets. Application fields utilizing interconnection networks include message transmission routing algorithms, routing paths without node duplication, fault tolerance, fault tolerant routing, one-to-many broadcasting, Hamilton cycle, VLSI layout, and fault diagnosis [1].

The interconnection networks proposed so far are classified based on the number of nodes: Mesh, Torus, Honeycomb Mesh, Hypercube, Folded Hypercube, Star graph, Transposition Graph, HCN(n,n), HFN(n,n) and Hyper-star [1]. Embedding of an interconnection network is to analyze whether the algorithm developed in an interconnection network G can be utilized in another interconnection network H. Embedding analysis is a method to efficiently utilize algorithms developed in interconnection networks. Parallel processing techniques to improve computer performance model the interconnection network structure as a graph and analyze it from a graph theory perspective. The method of modeling an interconnection network as a graph represents the processors of the interconnection network as nodes in the graph, and the communication lines connecting the processors as edges of the graph. The analysis method is to map the nodes and edges of graph G to the nodes and edges of another graph H one-to-one. The evaluation criteria for embedding include dilation, density rate, and expansion rate. Dilation refers to the routing distance from node u' to v' in graph H when two adjacent nodes u and v in graph G are mapped to nodes u' and v' in H [2-4].

The mesh structure is a flat graph that is used in the fields of VLSI circuit design, mobile communication systems, graphics, animation, and 3D modeling. Parallel computer structures utilizing mesh structures have been commercialized by MPP (Goodyear Aerospace), MP-I (MASPAR), Victor (IBM), Paragon (Intel), and T3D (Cray) [5].

There is a torus structure as an interconnection network in which the diameter of the mesh structure has been improved to approximately 1/2. A torus is a network constructed by connecting two nodes at both ends of a mesh row (column) with wraparound edges. The torus structure has advantages such as node symmetry, edge symmetry, short diameter, Hamilton cycle, and maximum fault tolerance [6][7][16].

Hypercube has node symmetry, edge symmetry, simple routing algorithm, and maximum fault tolerance. Hypercube has the disadvantage of increasing network costs because the degree increase rate is *O(n)* as the number of nodes increases [8][9]. To improve the network cost of hypercube, folded hypercube, HCN(n,n), HFN(n,n), etc. have been proposed [10][11][12].

In this paper, we propose and analyze a torus structure with the same number of nodes, 2^{2n}, and a node mapping algorithm for HCN(n,n). The structure of this paper is as follows. In Chapter 2, we learn about the torus and HCN(n,n) structures, and in Chapter 3, we analyze the fault tolerance of HCN(n,n) and the embedding algorithm between the $2^{2n-k} \times 2^k$ torus and the HCN(n,n) graph. Analyze. Finally, conclusions are drawn in Chapter 4.

2. RELATED RESEARCH

A common method of expressing node addresses in a torus structure is to express them according to the row and column where the node is located.

In this paper, the node address of a $2^{2n-k} \times 2^k$ torus will be expressed using gray code. In the torus $2^{2n-k} \times 2^k$ structure, 2^{2n-k} represents the row position and 2^k represents the column. The address of each node is expressed as a continuous 2n bit string by concatenating the row address followed by the column address.

Gray code is a coding method in which only one bit changes in a continuous binary bit string [13]. Gray code has self-conservativeness and periodicity, and (n+1)-bit gray code $g_n \cdots g_1 g_0$ is a random (n+1)-bit binary number. It can be derived from $b_n \cdots b_1 b_0$ as follows.

$g_i = b_i \oplus b_{i+1}$, 0≤i≤n-1, $g_n = b_n$.

Also, let B_i be a binary bit string of integer i, and B'_i be a bit string obtained by shifting B_i one bit to the right. B'_i is a binary bit string in which the B_i bit string is shifted by one bit and 0 is inserted on the left. The i-th gray code G in the bit string of B'_i can be defined as follows (0≤i≤ 2^n-1).

$G_i = B_i \oplus B'_i$

The symbol ⊕ in the formula represents exclusive-OR operation [9]. For example, 2-bit gray code is expressed as 00-01-11-10. 3-bit gray code can be expressed as 000-001-011-010-110-111-101-100 using 2-bit gray code.

A. Constant degree interconnection network

The mesh class is used as an interconnection network for multiple computers. Node addresses constituting an n-dimensional mesh can be expressed as an n-dimensional vector, and an edge exists when the addresses of two nodes differ by 1 in one dimension. Low-dimensional meshes have the advantage of being easy to design and highly scalable, but have the disadvantage of being large in diameter. High-dimensional meshes have smaller diameters and have efficient characteristics for parallel algorithms, but have the disadvantage of being expensive to implement. A torus structure was proposed to improve the shortcomings of the mesh structure. A torus has a regular network structure by adding wraparound edges to connect nodes at both ends of rows (columns) in a two-dimensional mesh structure.

When the torus structure is expressed *k* × n, there are *k* × n nodes, 2*k*n edges, degree 4, and diameter $\left[\frac{k}{n}\right] + \left[\frac{k}{n}\right]$. In this paper, the node address of the 2^{2n-k} x 2^k torus is expressed as $H = (h_1 h_2 \dots h_n h_{n+1} \dots . h_{2n})$. Node address expressed as 2n bit strings The set of nodes with the same bit string from the 1st to the nth node will be called a cluster $(h_1 h_2 \dots h_n)$. Therefore, there are 2^n clusters in the 2^{2n-k} x 2^k torus.

For example, a 4×4 torus has cluster(00), cluster(01), cluster(10), and cluster(11). The cluster and node addresses of the 4×4 torus structure are expressed in gray code as shown in Figure same. Four nodes in one row (column) are the nodes that make up a cluster. In this paper, the 2^{2n-k} x 2^k torus is simply referred to as a torus. In Figure 1, nodes constituting the cluster 00 are {0000, 0001, 0011, 0010}.

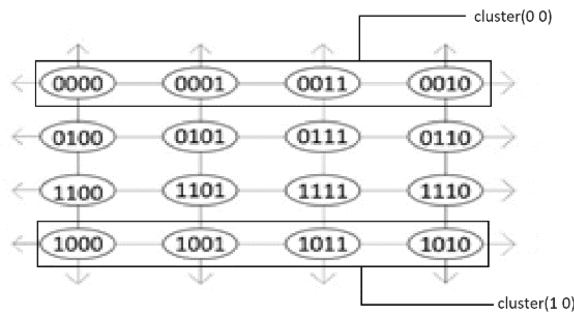

Figure 1

Various methods have been proposed to improve the diameter of the torus structure. For example, there are ways to reduce the number of degree of the interconnection network, change the node edge definition, use a structure with a short diameter, and layer the interconnection network structure. Toroidal mesh[19], Diagonal mesh[19], Honeycomb mesh[18], Petersen-torus[21], Hyper-torus[20], Semi-diagonal torus[17] are networks that improve the diameter of the constant degree graph have been proposed[25].

B. Hierarchical interconnection network

A layered structure of basic modules was proposed as a way to improve the diameter of the hypercube. Hierarchical structures that expand the number of nodes using basic modules include HCN(n,n) [23], HFN(n,n) [24], and Hierarchical Petersen Network [22].

HCN(n,n) is a network expanded to have a hierarchical structure using the n-dimensional hypercube Q_n as a basic module. HCN(n,n) has 2^{2n} nodes, $(n+1)2^{2n-1}$ edges, and degree n+1 [14-15]. The node address of HCN(n,n) is expressed as (I,J). In the node address (I,J), I represents the basic module of the hypercube, and J identifies the nodes inside the module. HCN(n,n) is divided into an internal edge and an external edge.

Internal edges connect nodes in a module, and external edges connect nodes in different modules. The external edge is divided into diameter edge and non-diameter edge according to the node address (I,J). The diameter edge connects node (I,I) and node (J,J), and I and J are complementary to each other. An external edge that is not a diameter edge is called a non-diameter edge, and connects node (I,J) with node (J,I).

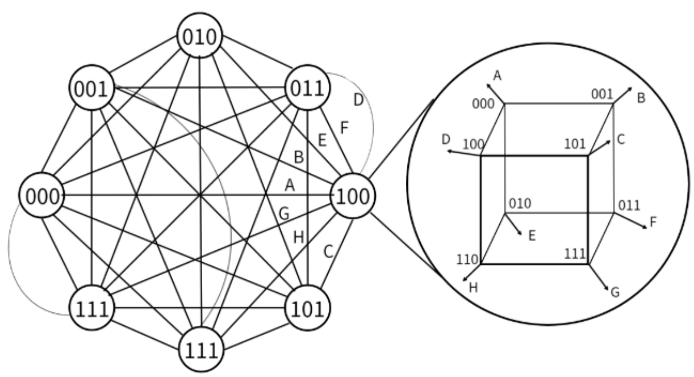

Figure 2. HCN (3, 3)

3. FAULT TOLERANCE AND NODE MAPPING ALGORITHM

A. HCN (n,n) Fault Tolerance

Fault tolerance is a measure that evaluates the severity of node failure in an interconnection network. Node (edge) connectivity is the minimum number of nodes (edges) that must be removed to divide the interconnection network into two or more graphs without overlapping nodes (edges). When k-1 or less nodes are removed from the interconnection network, the network is in a connected graph state, and when the appropriate k nodes are removed and the interconnection network is separated, the degree of connectivity of the interconnection network is called k. An interconnection network with the same node connectivity and degree is said to have maximum fault tolerance [1].

Let the node connectivity, edge connectivity, and degree of the interconnection network G be $_k(G)$, $\lambda(G)$, and δ(G), respectively. It is known that $_k(G) \leq \lambda(G) \leq \delta(G)$between the connectivity degree and the degree[16]. To analyze the maximum fault tolerance properties of the HCN(n,n) graph, we prove that the node connectivity and degree are the same. Additionally, using the degree and node connectivity results, it is shown that HCN(n,n) has the maximum fault tolerance.

Theorem 1 The node connectivity of HCN(n,n) is n+1 $_k\big(HCN(n,n)\big) = \mathrm{n} + 1, (\mathrm{n} \geq 2)$.

Proof The degree of HCN(n,n) is n+1, and the basic module of the hierarchy is the n-dimensional hypercube Q_n. It shows that HCN(n,n) is not divided even if n random nodes are removed from HCN(n,n). In (I,J), which represents a node of HCN(n,n), I is the address of the basic module itself, and J is the address of the node inside the basic module. The basic module of HCN(n,n) consists of 2^n hypercubes Q_n, and the node (I,J) inside the basic module is connected to the node (J,I) of another module by a non-diameter link. $(1 \leq I, J \leq 2^n, I \neq J)$. Additionally, the node (I, J) inside the basic module is connected to the node $(\bar{I}, \bar{J})$ of another module by a diameter link $(1 \leq I = J \leq 2^n)$.

In HCN(n,n), let |X| be the failed nodes and be a subset of V(HCN(n,n)) where It is shown that $_k(HCN(n,n)) \geq n + 1$by showing that the graph from which the faulty node set X is removed from HCN(n,n) is a connected graph.

Let the nodes of HCN(n,n) be S, and the graph obtained by removing the set of failed nodes X from HCN(n,n) is denoted as HCN(n,n)-X. It is divided into two types according to the location of the node X to be removed from HCN(n,n), showing that HCN(n,n)-X is always a connected graph.

Case 1. When the faulty node set X is located in one basic module of HCN(n,n)

When the address of node S of HCN(n,n) is (I,J), from the degree n+1 of HCN(n,n) to the node in hypercube Q_n with the address of basic module I including node S The number of connected degree is n. If n nodes adjacent to node S are the same as the node X to be removed, the basic module I including node S is divided into two components. That is, it is divided into a hypercube Q_n-X graph and node S. However, all nodes of basic module I, including node S, are connected to nodes (J,I) in other basic modules J by non-diameter links if $I \neq J$. Also, when I = J, there is one edge connecting the nodes $(\bar{I}, \bar{J})$ in the basic module $\bar{I}$ by a diameter link.

And the node (J,I) and the node $(\bar{I}, \bar{J})$ are connected to the inner edge of another basic module by a non-diameter link or diameter link. Therefore, in HCN(n,n), if n faulty node sets X are located within one basic module, HCN(n,n)-X is always connected. If only α out of n nodes adjacent to a certain node S of basic module I belong to the failed node set X, and n-α failed nodes belonging to the failed node set It is easy to see from the above case that HCN(n,n) is a connected graph.

Case 2. When X is located in two or more basic modules

In HCN(n,n), assuming that failed nodes are distributed across two or more basic modules I and J, the number of nodes that fail in one basic module I or J is at most n-1. In HCN(n,n), the number of degree for each node in the basic module is n, so even if n-1 adjacent nodes are removed from node S of basic module I or J, node S remains 1 of basic module I including node S. It can be seen from case 1 that it is connected to the dog node. If the remaining node to be removed is a node of basic module J that does not include S, it can be seen that HCN(n,n)-X is connected.

Therefore, it can be seen that HCN(n,n) is always connected even if the faulty node set X at any position is removed from HCN(n,n).

The node connectivity of HCN(n, n) is $_k\big(HCN(n,n)\big) \geq n + 1$, which is the same as the number of degree, and HCN(n, n) is a regular network with a number of degree n+1, $_k\big(HCN(n,n)\big) \leq n + 1$. Therefore, $_k\big(HCN(n,n)\big) = n + 1$, and HCN(n, n) has the maximum fault tolerance.

B. HCN (n,n) Node mapping Algorithm

2^{2n-k} x 2^k torus and HCN(n,n) have the same 2^{2n} nodes. The nodes of the torus are mapped one-to-one to those with the same node address in HCN(n,n). Because the torus cluster has the same structure as the basic module of HCN, the torus cluster node is mapped to the basic module node of HCN(n,n). And the edge connecting the cluster of the torus is mapped to the external edge connecting the basic module of HCN(n,n).

Theorem 2 2^{2n-k} x 2^k torus structure is HCN(n,n), which allows one-to-one node mapping with a dilation of 3 and an expansion rate of 1.

Proof Let us classify two adjacent nodes in a torus structure as $H = (h_1h_2 \dots h_nh_{n+1} \dots h_{2n})$ and $H' = (h'_1h'_2 \dots h'_nh'_{n+1} \dots h'_{2n})$. A random node in HCN is $S = (s_1s_2 \dots s_ns_{n+1} \dots . s_{2n})$, $S' = (s'_1s'_2 \dots s'_ns'_{n+1} \dots s'_{2n})$. Map node H of the torus into node S of the HCN, and map node H' of the torus into node S' of the HCN. The dilation is analyzed by analyzing the path length from node S of HCN to node S'. The cases are divided according to the bit string conditions of the node H of the torus and the adjacent H'.

Case 1. $h_1h_2 \dots h_n = h'_1h'_2 \dots h'_n$

The bit string of node S of HCN to which node H of torus is mapped is $(s_1s_2 \dots s_i \dots s_n, s_{n+1} \dots s_j \dots s_{2n})$, and the bit string of node S' to which node H' is mapped is $(s_1s_2 \dots s_i \dots s_n, s_{n+1} \dots s_j \dots s_{2n})$. In the bit strings of nodes S and S', only the j-th bit is complementary to each other, so it can be seen that nodes S and S' are nodes within the same module of HCN. By the definition of HCN, nodes S and S' are nodes adjacent to each other. Therefore, it can be seen that embedding is possible at a dilation of 1 when mapping the nodes H and H' of the torus to the nodes S and S' of the HCN, respectively.

Figure 3. Case1

Case 2. $h_1h_2 \dots h_n \neq h'_1h'_2 \dots h'_n$

The bit string of node S of HCN to which node H' of the torus is mapped is $(s_1s_2 \dots s_i \dots s_n, s_{n+1} \dots s_j \dots s_{2n})$, and the node S' to which node H' is mapped is $(s_1s_2 \dots s_i \dots s_n, s_{n+1} \dots s_j \dots s_{2n})$. Since only the i-th bit in the bit strings of nodes S and S' is complement, it can be seen that S and S' are nodes in different modules of HCN. The nodes of the mapped HCN $(s_1s_2 \dots s_i \dots s_n, s_{n+1} \dots s_j \dots s_{2n})$are converted to $(s_{n+1} \dots s_j \dots s_{2n}, s_1s_2 \dots s_i \dots s_n)$by non-diameter edges. Connect to Connect the connected node $(s_{n+1} \dots s_j \dots s_{2n}, s_1s_2 \dots s_i \dots s_n)$ to the node $(s_{n+1} \dots s_j \dots s_{2n}, s_1s_2 \dots s_i \dots s_n)$ inside the module.

Connect the connected $(s_{n+1} \dots s_j \dots s_{2n}, s_1s_2 \dots s_i \dots s_n)$ to $(s_{n+1} \dots s_j \dots s_{2n}, s_1s_2 \dots s_i \dots s_n)$ by a non-diameter edge. Therefore, it can be seen that embedding is possible at a dilation of 3 when mapping the nodes H and H' of the torus to the nodes S and S' of the HCN, respectively.

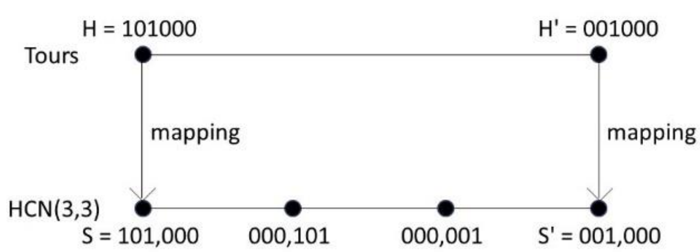

Figure 4. Case 2

Case 3. $h_1h_2 \dots h_n \neq h'_1h'_2 \dots h'_n$ and $h_1h_2 \dots h_n = h_{n+1} \dots h_{2n}$

The bit string of node S of HCN to which node H of the torus is mapped is $(s_1s_2 \dots s_i \dots s_n, s_{n+1} \dots s_j \dots s_{2n})$, and the bit string of node S' to which node H' is mapped is $(s_1s_2 \dots s_i \dots s_n, s_{n+1} \dots s_j \dots s_{2n})$. Among the mapped bit strings of node S, since $s_1s_2 \dots s_i \dots s_n = s_{n+1} \dots s_j \dots s_{2n}$, there is no need to use the first connected non-diameter edge in case 2. Therefore, in this case, dilation is 2.

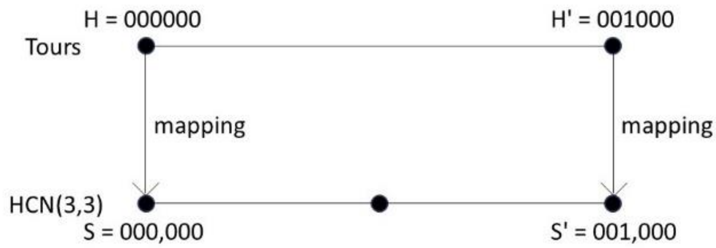

Figure 5. Case 3

As proven in the three cases above, the dilation required to embed a torus in HCN is 3 or less.

Theorem 3 HCN(n,n) can be mapped to a 2^{2n-k} x 2^k Torus at a dilation of $O(2^n)$.

Proof The internal edge, non-diameter edge, and diameter edge are the edges that make up HCN. After HCN's nodes U= (I,J) and V= (J,I) are thought to be 1-to-1 respectively as Torus' nodes U'= (I,J) and V'= (J,I), the dilation is calculated by analyzing the path length.

Case 1. Internal edge

The n-dimensional hypercube, the basic module of HCN, is represented by an inner edge. The inner edge of HCN can be mapped 1-to-1 into a cluster of 2^{2n-k} x 2^k Torus, so the dilation is 1. For example, the edge (0000,0001) of HCN is mapped to (0000,0001) in the torus structure.

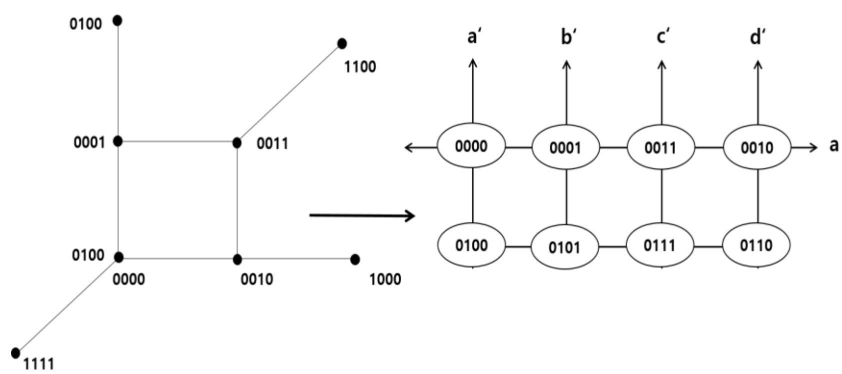

Figure 6. Internal Edge

Case 2. Non-diameter edge

The node address V= (J,I) is connected to the HCN node U= (I,J) by a non-diameter edge. The nodes U and V of HCN are mapped 1-to-1 into U'= (I,J) and V'= (J,I), which have the same node address of Torus. There are two possible movement paths from Torus node U'= (I,J) to V'= (J,I) as follows. Each cluster in Torus has 2^n nodes in each row and column, organized in a ring shape.

Path 1: U'= (I,J) → (I,I) → V'= (J,I)

Path 2: U'= (I,J) → (J,J) → V'= (J,J)

The path from U'=(I,J) → (I,I) of path 1 requires a maximum length of $\frac{2^n}{2}$ within one cluster consisting of a row-shaped ring. The path length from node (I,I) → V'=(J,I) requires a maximum length of $\frac{2^n}{2}$ within one cluster consisting of a column-shaped ring. Therefore, the dilation of path 1 is 2^n. If a similar method is applied to path 2, the dilation is 2^n. For example, the path of HCN's edge (0011, 1100) mapped from the torus is as follows.

Path: 0011 −> 0111−>1111−>1101−>1100

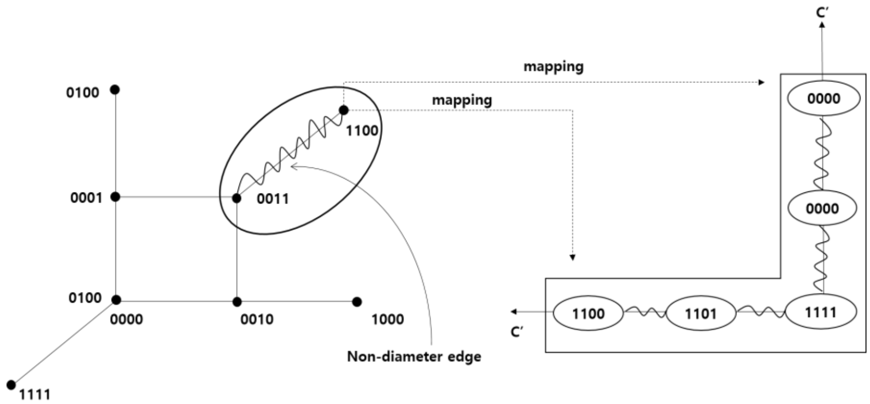

Figure 7. Non-diameter edge

Case 3. Diameter edge

In HCN, nodes U= (I,I) and V= (J,J) are connected by diameter edges. At this time, I=$\bar{J}$. The nodes U and V of HCN are mapped 1-to-1 to U'= (I,I) and V'= (J,J), which have the same node address of Torus. The same method as case 2 is applied to the path length from node U' to V'. Therefore, the dilation is 2^n.For example, the path of HCN's edge (0000,1111) mapped from the torus is as follows.

Path: 0000−>0001−>0011−>0111−>1111

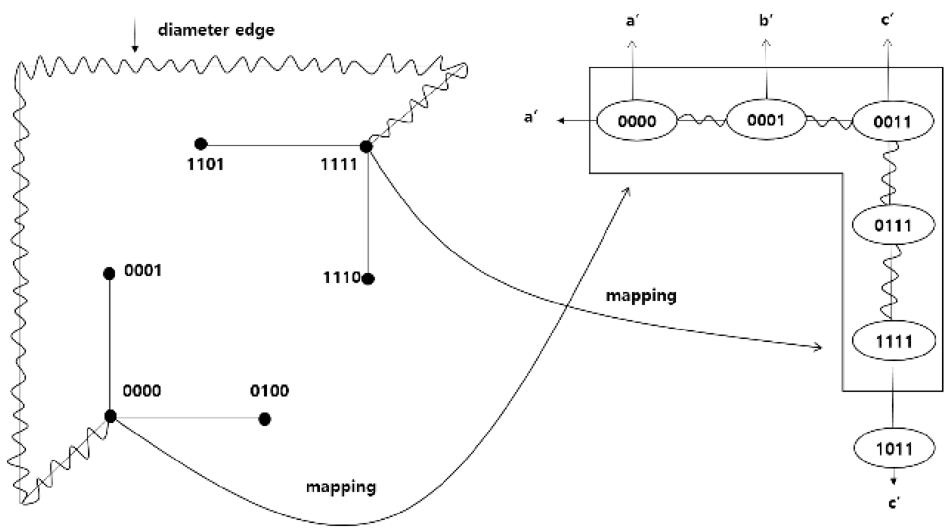

Figure 8

Therefore, HCN can map nodes one-to-one with a 2^{2n-k} x 2^k Torus with dilation of $O(2^n)$.

4. CONCLUSION

The demand for high-performance computing to solve many problems in the field of science and technology continues to increase. Embedding of interconnection networks is a research field aimed at efficiently utilizing algorithms developed in computer architectures for parallel processing. Embedding of an interconnection network is mapping a network G to a network H to find out how a network G is included in or related to another network H.

In this paper, through analysis of the fault tolerance of the HCN(n,n) graph, it was shown that the HCN(n,n) graph has the maximum fault tolerance. In addition, the embedding properties were analyzed to utilize the algorithm developed in the torus structure, which is widely used as an interconnection network, in the HCN(n,n) graph. It was shown that the torus structure can be embedded in HCN(n,n) with dilation of 3, and the HCN(n,n) structure can be mapped to the torus graph with an extension rate of $O(2^n)$. The research results of this paper mean that various algorithms developed in Torus can be efficiently used in HCN(n,n).

Acknowledge

This work was supported by Sunchon National University Research Fund in 2023.(Grant number: 2023-0321)

5. REFERENCES

[1] T. Feng, "A Survey of Interconnection Networks," IEEE computer, pp. 12-27, December 1981.

[2] S. B. Akers and B. Krishnamurthy, "A Group-Theoretic Model for Symmertric Interconnection Network," IEEE Trans. Comput., Vol. 38, No. 4, pp. 555-565, 1989.

[3] Y. Wu, "Embedding of Tree Networks into Hypercubes," J. Parallel and Distributed Computing, Vol. 2, pp. 238-249, 1985.

[4] Yang, Y.Y. Tang and J. Cao, "Embedding Torus in Hexagonal Honeycomb Torus," IET Computers & Digital Techniques, Vol. 2, pp 86-93, Mar 2008.

[5] J. Bruck, R Cypher and C.-R. Ho, "Wildcard Dimensions, Coding Theory and Fault-Tolerant Meshes and Hypercubes," IEEE Trans. on Computers, Vol. 44, No. 1, pp. 150-155, 1995.

[6] J. G. Peters and M. Syska, "Circuit-Switched Broadcasting in Torus Networks," IEEE Trans. Parallel Distributed syst., Vol. 7, No. 3, pp. 246-255, March 1996.

[7] J. Jwo, S. Lakshmivarahan, and S. Dhall, "Embedding of Cycles and Grids in Star Graphs," Proc. of IEEE, Vol. 5, pp. 540-547, 1990.

[8] Y. Saad and M. H. Schutltz, "Topological Properties of Hypercubes," IEEE Trans. Comput., Vol. 37, pp. 867-872, 1988.

[9] S. Vaidya, P. S. N. Rao and S. R. Shankar, "A Class of Hypercube-like Networks," Proc. of the 5th IEEE Symposium on Parallel and Distributed Processing, pp. 800-803, Dec. 1993.

[10] K. Ghose and K. R. Desai, "Hierarchical Cubic Networks," IEEE Trans. Parallel Distributed syst., Vol. 6, No. 4, pp. 427-436, 1995.

[11] S-K. Yun and K-H. Park, "Comments on Hierarchical Cubic Networks," IEEE Trans. Parallel and Distributed syst., Vol. 9, No. 4, pp. 410-414, 1998.

[12] D-R. Duh, G-H. Chen and J-F. Fang, "Algorithms and Properties of a New Two-Level Network with Folded Hypercubes as Basic Modules," IEEE Trans. Parallel and Distributed Systems, Vol. 6, No. 7, pp. 714-723, July 1995.

[13] H. Mehta, R.M. Owens and M.J. Irwin, "Some issues in Gray code addressing," Proceedings of the Sixth Great Lakes Symposium on VLSI, March 1996.

[14] S.-K. Yun and K.-H. Park, "Comments on Hierarchical Cubic Networks," IEEE Trans. Parallel Distributed syst., Vol. 9, No. 4, pp. 410-414, 1998.

[15] S-L. Zhao, R-X. Hao and J. Wu, "The generalized 4-connectivity of hierarchical cubic networks," Discrete Applied Mathematics, Vol. 289, No. 31, pp. 194-206, 2021.

[16] T. Hasunuma and C. Morisaka, "Completely independent spanning trees in torus networks," Networks, Vol. 60, No. 1, pp. 59-69, 2012.

[17] Y-G. Wang, H-M Du and X-B. Shen, "Topological properties and routing algorithm for semi-diagonal torus networks," The Journal of China Universities of Posts and Telecommunications, Vol. 18, No. 5, pp. 64-70, 2011.

[18] I. Stojmenovic, "Honeycomb Network: Topological Properties and Communication Algorithms," IEEE Trans. Parallel and Distributed Systems, Vol. 8, No. 10, pp. 1036-1042, 1997.

[19] K-W. Tang and S-A. Padubidri, "Diagonal and Toroidal Mesh Networks," IEEE Trans. Comput., Vol. 43, No. 7, pp. 815-826, 1994.

[20] J-S. Kim, S-W. Kim, K. Qiu, and H-O. Lee, "Some Properties and Algorithms for the Hyper-torus Network," Journal of Supercomputing, Vol. 69, pp. 121-138, 2014.

[21] J-S. Kim, H-O. Lee, M-H. Kim, and S-W. Kim, "The New Petersen-torus Networks," Journal of Supercomputing, Vol. 71, pp. 894-908, 2015.

[22] J-H. Seo, J-S. Kim, H-J. Jang, and H-O. Lee, “The Hierarchical Petersen Network: A New Interconnection Network with Fixed Degree," Journal of Supercomputing, Vol. 74, pp. 1636-1654, 2018.

[23] K. Ghose and K-R. Desai, “Hierarchical Cubic Networks," IEEE Trans. Parallel Distributed systs., Vol. 6, No. 4, pp. 427-436, 1995.

[24] D-R. Duh, G-H. Chen and J-F. Fang, “Algorithms and Properties of a New Two-Level Network with Folded Hypercubes as Basic Modules," IEEE Trans. Parallel Distributed systs., Vol. 6, No.7, pp. 714-723, 1995.

[25] H-M. Lee and D-H. Seo, “FedLC: Optimizing Federated Learning in Non-IID Data via Label-wise Clustering," IEEE Access, Vol. 11, pp. 42082-42095, 2023.

DOI: https://doi.org/10.15379/ijmst.v10i2.2781