Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 56 (2015) 409 – 414

Procedia Computer Science

The 2nd International Workshop on Design and Performance of Networks on Chip (DPNoC 2015)

Shortest Path Routing Algorithm for Hierarchical Interconnection Network-on-Chip

Omair Inam[a,*], Sharifa Al Khanjari[b], Wim Vanderbauwhede[b]

[a]*Department of Computer Science, COMSATS Institute of Information Technology, Islamabad 45550, Pakistan*
[b]*School of Computing Science, University of Glasgow, Glasgow G128 QQ, UK*

Abstract

Interconnection networks play a significant role in efficient on-chip communication for multicore systems. This paper introduces a new interconnection topology called the Hierarchical Cross Connected Recursive network (HCCR) and a shortest path routing algorithm for the HCCR. Proposed topology offers a high degree of regularity, scalability, and symmetry with a reduced number of links and node degree. A unique address encoding scheme is proposed for hierarchical graphical representation of HCCR networks, and based on this scheme a shortest path routing algorithm is devised. The algorithm requires $5(k-1)$ time where $k = \log_4^n - 2$ and $k > 0$, in worst case to determine the next node along the shortest path.

Peer-review under responsibility of the Conference Program Chairs

Keywords: Shortest path routing algorithm; Network-on-Chip; Hierarchical interconnection network; Hierarchical graph representation;

1. Introduction

Over the last decade, on-chip communication has become a major bottleneck due to trend in integrating thousands of cores on single chip. Network on Chip (NoC) has emerged as a promising candidate for efficient on-chip communication, providing high throughput, better scalability and reusability in multicore systems. The two

* Corresponding author. Tel.: +92-51-5462642; fax: +92-51-9247006.
E-mail address: omair_inam@comsats.edu.pk

Peer-review under responsibility of the Conference Program Chairs
doi:10.1016/j.procs.2015.07.228

fundamental design challenges for any interconnection network in terms of overall system cost and performance are topology and routing algorithm.

There have been many architectural and theoretical studies on NOCs topologies. Direct topologies[1] such as rings are cost-effective, but deliver relatively poor performance, especially as the number of connected cores increases. A reconfigurable system[2] was proposed based on a hierarchical mesh interconnection network consisting of nearest neighbor connectivity at the lowest hierarchy level, together with horizontal and vertical buses for global connectivity. WK-recursive hierarchical networks[3] offer high degree of regularity, scalability, and symmetry, which make them suitable for manufacture using VLSI technology. On the other hand, among higher connectivity topologies, 2-D mesh[4] has been the most popular topology. Small degree topologies such as Octagon[5] and Spidergon[6] are proposed by ST Microelectronics. Triple-based Hierarchical Interconnection Network[7] (THIN) is presented in literature, focusing on decreasing node degree, reducing links and shortening diameter.

An extensive research has been conducted on the shortest path routing algorithm for hierarchical interconnection networks. Yu, X & Li presented an algorithm for shortest path routing on crossed cube connected ring network[8], with time complexity $O(n^2)$. Optimal dynamic[9] two terminal message routing algorithm was presented for k-circulant ($k \geq 2$) networks for the restricted shortest path in $O(\log n)$ time. In another work, shortest path routing was proposed for WK-recursive[10] incomplete networks with $O\ (d.t)$ time preprocessing where $d > 1$ is the size of basic building block and $t \geq 1$ is the level of expansion. This algorithm takes $O\ (t)$ time for each intermediate node to determine the next node along the shortest path. Qiu presented a routing algorithm that finds n disjoint shortest paths in n-dimensional hypercube[11] in $O(n^3 \log n)$. Jha & Jana presented shortest path routing, which requires 12 log n + 1time in the worst case, is proposed for Multi-Mesh of Trees on n^4 processor network[12].

This paper proposes a new on-chip interconnection architecture i.e. Hierarchical Cross Connected Recursive network (HCCR) along with efficient shortest path routing algorithm. HCCR is derived from the WK-recursive topology[3], with small degree and reduced number of links. The proposed network is simple, hierarchical, symmetrical and scalable interconnection architecture which can make it suitable for communication in parallel/distributed networks. Our contribution to this paper is to give unified representation of HCCR with unique address mapping. On the basis address mapping, the shortest path routing algorithm is proposed. In worst case it takes 5(k-1) time, where $k = \log_4^n - 2$ and $k > 0$, to determine the next node along shortest path.

2. Topology and Construction of HCCR

HCCR is a hierarchical, symmetrical and scalable interconnection architecture with small node degree, reduced number of links and short diameter. Level k HCCR (L_k) network ($k = \log_4^n - 2$ where n denotes the number of nodes), is constructed using recursive L_{K-1} HCCR networks as shown in figure 1.d. Let L_0 denotes *local block* in HCCR with $k = 0$. L_0 consists of 4 cross connected basic modules as shown in figure 1.a and 1.b. Each node in L_k is associated with three types of links labeled as 'X' (along x-axis), 'Y' (along y-axis) and 'B' as shown in the figure 1.c. B is called *bridge link* which joins one basic module to another basic module in L_K HCCR. Table1 compares the network properties of the proposed HCCR with 2-D mesh and Hypercube.

Table 1. Comparison of topology properties

Types of network	Network degree	Number of links	Diameter
HCCR	3	$\frac{1}{2}(3N-4)$	$(2^{\log_4^N - 1} + \sqrt{N} - 1)$
2-D mesh	4	$2(N-\sqrt{N})$	$2(\sqrt{N}-1)$
Hypercube	$\log_2^N$	$\frac{N \log_2^N}{2}$	$\log_2^N$

Network cost is usually determined by the number of links, number of routers, wire density, number of interfaces and VLSI layout area complexity models. Networks with lower node degree are likely to have less communication complexity of the router, including wire congestion in the physical layout. Moreover with constant nodes degree: independent of the network size, the network is more modular. For all switching strategies, small network diameter improves network contention (in buffer and edges) and reduces propagation delay in point-to-point communication.

Figure 2.a and 2.b, compares the links and communication complexity of three different networks having different sizes, node degree and diameters. A major drawback of hypercube-based networks is that they lack expansibility. As their size grows, number of ports gives an upper bound on their expansion. On the other hand HCCR has the fewest number of links in different scales when compared to 2-D Mesh and Hypercube. As shown in figure 2.a and 2.b that despite having low diameter over large size networks, Hypercube becomes complex structure due to more number of links and variable node degree. The diameter of HCCR is shorter than 2-D mesh even for the large network size. This property of HCCR can be exploited to get high performance communication at low cost by utilizing the locality that exists in parallel/distributed network.

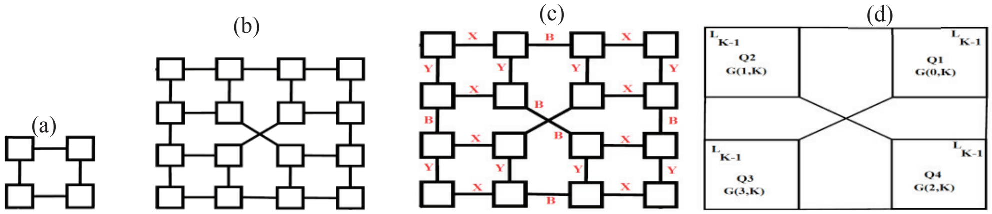

Fig. 1. (a)Basic Module of HCC; (b) Local Block. $L_0(K = 0)$; (c) Link description in HCC; (d) HCCR L_K network

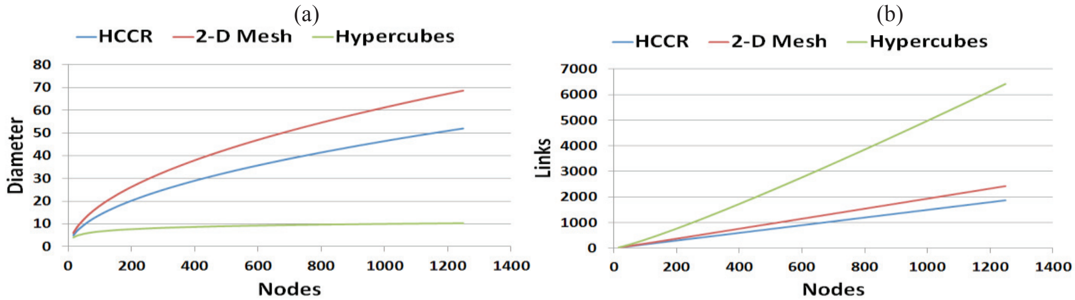

Fig. 2. (a) Diameter vs Network size; (b) Links vs Network size.

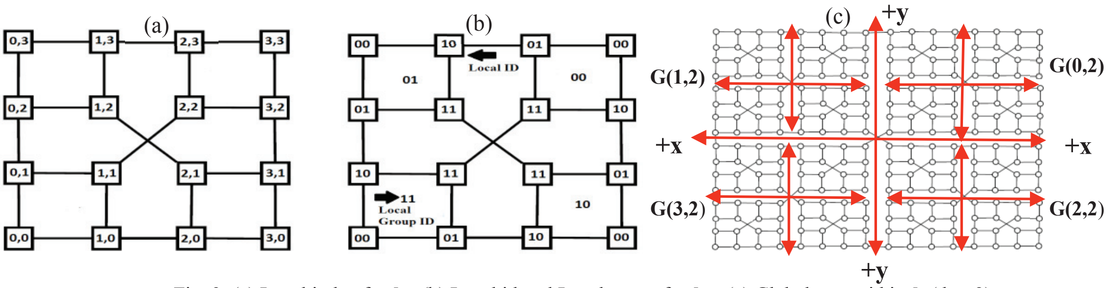

Fig. 3. (a) Local index for L_0; (b) Local id and Local group for L_0; (c) Global group id in L_2(k = 2)

3. Hierarchical Address-Encoding scheme for HCCR

In this section hierarchical address encoding scheme for L_k HCCR is discussed. This scheme has its own significance in realization of the shortest path routing algorithm. Let $n(lx, ly)$ denote the position vector of node n in L_0 where $0 \leq lx \leq 3$ and $0 \leq ly \leq 3$. As shown in figure 3.a, lx and ly are referred as *local index* of node n in x and y dimension respectively. All the nodes in L_0 are represented by unique ID's with 2-bits symbol as shown in the figure 3.b. If N_L defines the set of ID's in L_K HCCR then $N_L = \{N_{L0}, N_{L1}, N_{L2}, N_{L3}\} = \{00, 01, 10, 11\}$. N_{Li} denotes *local id* of node i in L_0 where $0 \leq i \leq 3$. All nodes in L_0 are divided into 4 groups with 2-bits symbol as shown in the figure 3.b. If G_L defines the set of 4 groups in L_0 then $G_L = \{G_{L0}, G_{L1}, G_{L2}, G_{L3}\} = \{00, 01, 10, 11\}$. Each group consists of 4 nodes with unique *local id's* as shown in figure 3.b. All the groups are linked with each other via B links. G_{Li} denotes *local group id* of node i in L_0 where $0 \leq i \leq 3$. For global representation of any node in the HCCR $k > 0$, 4 symmetrical quadrants (Q1, Q2, Q3, and Q4) are defined for each recursive level as shown in figure 3.c. All 4 quadrants have only positive x and y axis. *Global group ids* $(\mathrm{G}(\mathrm{i},\mathrm{k}))$ where $0 \leq \mathrm{i} < 4$, are assigned to all the nodes in the $\mathrm{L_k}$ network according to their respective quadrants. See figure 3.c where $\mathrm{G}(0,2)$ represents that all the nodes of $\mathrm{L_2}$ HCCR in Q1 have global group id = 0. In any quadrant, nodes are addressed by *Global index* $\mathrm{n_g}$ $(\mathrm{gx},\mathrm{gy})$ where gx , gy are the coordinates of $\mathrm{n_g}$ in x and y dimension with $0 \leq \mathrm{gx} \leq 2^{k+1} - 1$, $0 \leq \mathrm{gy} \leq 2^{k+1} - 1$. Corners nodes of $\mathrm{L_0}$ are defined as top right corner $(\mathrm{n_{lx}} = 3, \mathrm{n_{lx}} = 3)$, top left corner $(\mathrm{n_{lx}} = 0, \mathrm{n_{lx}} = 3)$, bottom right corner $(\mathrm{n_{lx}} = 3, \mathrm{n_{lx}} = 0)$ and bottom left corner $(\mathrm{n_{lx}} = 0, \mathrm{n_{lx}} = 0)$.

4. Address Mapping for Shortest Routing Algorithm

Based on address-encoding presented in section 3, this section discuses address mapping scheme for HCCR $\mathrm{L_K}$ where $\mathrm{k} > 0$. This scheme maps every source node against a fixed number. These fixed numbers are used as the reference point to find the shortest path in proposed algorithm and remain fixed for $\mathrm{k} > 0$. Figure 4.a illustrates the two basic communication patterns; *up-down communication* and *diagonal communication*, between any two nodes communicating from one global group to another. All other patterns exist in the network are the mirror of above stated communication patterns along x- axis or y-axis. In proposed algorithm only basic patterns are used as reference for all other patterns existing in the network. Generation of fixed number for any source depends on the basic communication pattern between source and destination. Following section explains the two basic communication patterns and generation of fixed numbers.

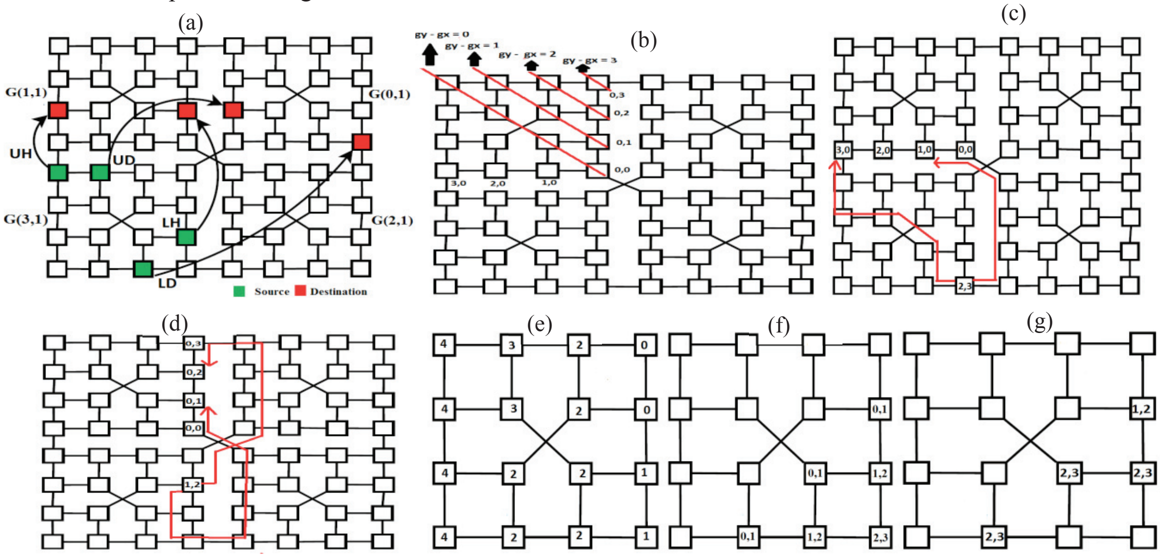

Fig. 4. (a) Up-Down and Diagonal communication; (b) Generation of fixed numbers for UH based on intercepts of lines; (c) LH communication from top left corner; (d) LH communication from bottom right corner; e) mapping for UH in G(3,1); (f) and (g) mapping for LH in G(3,1)

4.1. Address Mapping for Up-Down Communication and Diagonal Communication

For Up-Down communication from $\mathrm{G}(3,1)$ to $\mathrm{G}(1,1)$ illustrated in Figure 4.a, any sender node (S) can either belong to Upper Half (UH) with $\mathrm{Sgy} \geq \mathrm{Sgx}$, or Lower Half (LH) with $\mathrm{Sgy} < \mathrm{Sgx}$. For UH communication $\mathrm{MapForUH}(\,)$ assigns fixed number to the sender node in $\mathrm{G}(3,1)$ according to its local index in $\mathrm{L_0}$. In this case fixed number represents the line intercepts in $\mathrm{G}(1,1)$ as shown in Figure 4.b and 4.e. For communication in UH, function $\mathrm{UpperHalf}(\,)$ compares fixed numbers to select the appropriate corner of local block $\mathrm{L_0}$, between top left and top right corner, for the shortest path communication and $\mathrm{getdirection}(\,)$ returns appropriate direction ('X', 'Y' or 'B') for communication with in local block ($\mathrm{L_0}$). For communication from LH of $\mathrm{G}(3,1)$ to $\mathrm{G}(1,1)$, $\mathrm{MapForLH\_TR\_TL}(\,)$ and $\mathrm{MapForLH\_BR\_TR}(\,)$ map the fixed number to sender node in $\mathrm{G}(3,1)$ according to sender's local index in $\mathrm{L_0}$ as shown in the figure 4.f and 4.g. In this case fixed numbers represent global index of nodes in $\mathrm{G}(1,1)$ which are at equal distance from the sender node. These fixed numbers provide reference to select one corner among top right, top left or bottom right of L_0. Inrange( ) checks the possible corner nodes on the basis of fixed numbers and LowerHalf( ) selects the appropriate corner of local block L_0 for the shortest path communication as shown in figure 4.c and 4.d. For diagonal communication Upper Half Diagonal (UD) and lower Half Diagonal (LD) are defined in the same way as UH and LH. For UD and LD communication MapForUD( ) and MapForLD( )assign the fixed numbers to the sender nodes. UpperHalfDiagonal( )and LowerHalfDiagonal( ) select an appropriate corner of local block L_0 for the shortest path communication and getdirection( ) returns the direction ('X', 'Y' or 'B'). Pseudo codes for up-down and diagonal communication are given in figure 5.

5. Shortest Path Routing Algorithm for HCCR

As shown in the figure 5, algorithm first determines the level of communication between source-destination using their addresses. In case of local communication (i.e. source and destination belong to same local block (L_0)), callLocalBlockcom( ) finds the direction in which packet needs to move forward for the shortest path inside L_0 . For local communication in L_0, all intermediate nodes between source and destination call the stated function to find valid direction for shortest path communication. But in case of communication outside (L_0), the algorithm finds one of the two communication patterns described above. After determining pattern, index conversions are done in the sender and receiver addresses with respect to the reference communication patterns; $G(3,k) \rightarrow G(1,k)$ reference for updown-communciation and $G(3,k) \rightarrow G(0,k)$ for diagonal communication. FindRoutingParam_up_down_comm( ) and FindRoutingParam_diagonal_comm( ) are performing all conversions in the sender and receiver addresses according to the above defined references. In case where sender and receiver do not belong to L_0, algorithm selects corner node for an exit from L_0. The dimension of selected corner node is provided to the getdirection( ) which works similar to the callLocalBlock( ) with only difference in the context. Both are responsible for the shortest path communication inside L_0 . Every intermediate node sends packet in appropriate corner of L_0 until the packet finally reaches its destination block. If k denotes the level of HCCR network then FindRoutingParam_up_down_comm( ) and FindRoutingParam_diagonal_comm( ) each require 3(k-1) time for index conversions. For local communication (k = 0), callLocalBlockcom( ) and getaddress( ) involves single call with time complexity $O(1)$.LowerHalf( ) require 2(k-1) time whereas UpperHalfDiaognal( ) and LowerHalfDiagonal( ) each take k-1 time. UpperHalf( ) involves single call to getadress( ) requires $O(1)$ time. As shown in Fig. 5 Up-down communication has the worst case where 5(k-1) times is required to determine the next node along the shortest path.

```
Direction UpperHalf(arg)
{FixNumber = MapForUH()
if(Rgy-Rgx < FixNumber) corner = topleft
else corner = topright
direction = getdirection(cornernode)
return direction
}//end UpperHalf()

Direction LowerHalf(arg)
{FixNumberTR_TL = MapForLH_TR_TL()
FixNumberBR_TR = MapForLH_BR_TR()
rangex = Inrange(FixNumberTR_TL)
rangey = Inrange(FixNumberBR_TR)
 if (rangex == true && rangey == false)
 { corner = bottomright
   direction = getdirection(cornernode)
 }
else direction = UpperHalf(arg)
return direction
}//end LowerHalf()
```

```
Direction UpperHalfDiagonal(arg)
{FixNumber = MapForUD()
rangey = Inrange(FixNumber)
 if (rangey == fasle) corner = topleft
else corner = topright
direction = getdirection(cornernode)
return direction
}//end UpperHalfDiagonal()

Direction LowerHalfDiagonal(arg)
{FixNumber = MapForUD()
rangey = Inrange(FixNumber)
 if (rangey == fasle) corner = bottomright
else corner = topright
direction = getdirection(cornernode)
return direction
}//end LowerHalfDiagonal()
```

```
Direction ShortestPathRouting_HCCR()
{if (levelOfCommuniation == 0)direction = callLocalBlockCom()
 else{
  if (up_down_comm)
  {  para = FindRoutingParam_up_down_comm(arg) // index conversions
     if (upperhalf) direction = UpperHalf(para)
     else direction = LowerHalf(para) }
  else { para = FindRoutingParam_diagonal_comm(arg) // index conversions
    if (low_half_diag) direction = UpperHalfDiagonal(para)
    else if (upper_half_diag) direction = LowerHalfDiagonal(para)
    else {
    corner = topright
    direction = getdirection(cornernode)
    }}}
return direction
}// end ShortestPathRouting_HCCR
```

Fig. 5. Pseudo Code of up-down and diagonal communication functions and shortest path routing algorithm for HCCR

6. Experiments and result

The proposed algorithm is verified by finding average distance using different network sizes of HCCR (k = 0 to k = 3) and comparing the results with Dijkstra's algorithm. The results plotted with different networks size up to 1024 nodes are shown in Figure 6 a, there is 7.7% increase in the average distance of HCCR (L_3) in comparison to 2-D Mesh. Whereas L_3 HCCR has fewer longer paths in comparison to 2-D Mesh (32 x 32) as shown in Figure 6.b. This property of HCCR can be scope of future work to get high performance communication at low cost by exploring the locality based traffic for HCCR networks.

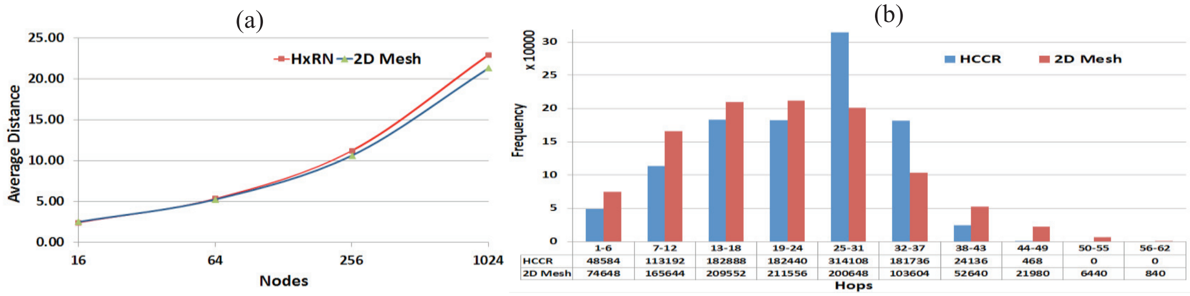

Fig. 6. (a)Comparison of average distance; (b) HCCR (L_3) vs 2-D Mesh (32x32)

7. Conclusion

We have presented a new interconnection topology called the Hierarchical Cross Connected Recursive network (HCCR) and a shortest path routing algorithm for the HCCR. On comparing the 2-D Mesh and hypercube topologies, it is shown that the properties of HCCR make it suitable even for large network sizes due to better expansibility and small node degree. We proposed a shortest path routing algorithm to guarantee efficient communication in HCCR. Our results show that in case of shortest path communication in HCCR (L_3), there are fewer longer paths compared to a 2-D Mesh (32x32). We show that in worst case proposed algorithm takes a time proportional to 5(k-1) where $k = \log_4^n - 2$ and $k > 1$, to determine the next node along the shortest path. Future research will focus on simulating the shortest path routing algorithm under different network load conditions and comparing the performance of HCCR with other interconnection networks.

References

1. Bononi, L., & Concer, N. (2006, March). Simulation and analysis of network on chip architectures: ring, spidergon and 2D mesh. In *Proceedings of the conference on Design, automation and test in Europe: Designers' forum* (pp. 154-159). European Design and Automation Association.
2. Singh, H., Lee, M. H., Lu, G., Kurdahi, F. J., Bagherzadeh, N., & Chaves Filho, E. M. (2000). MorphoSys: an integrated reconfigurable system for data-parallel and computation-intensive applications. *Computers, IEEE Transactions on*,*49*(5), 465-481.
3. Duh, D. R., & Chen, G. H. (1994). Topological properties of WK-recursive networks. *Journal of Parallel and Distributed Computing*, *23*(3), 468-474.
4. Mirza-Aghatabar, M., Koohi, S., Hessabi, S., & Pedram, M. (2007, August). An empirical investigation of mesh and torus NoC topologies under different routing algorithms and traffic models. In *Digital System Design Architectures, Methods and Tools, 2007. DSD 2007. 10th Euromicro Conference on* (pp. 19-26). IEEE.
5. Karim, F., Nguyen, A., Dey, S., & Rao, R. (2001, June). On-chip communication architecture for OC-768 network processors. In *Proceedings of the 38th annual Design Automation Conference* (pp. 678-683). ACM.
6. Bononi, L., & Concer, N. (2006, March). Simulation and analysis of network on chip architectures: ring, spidergon and 2D mesh. In *Proceedings of the conference on Design, automation and test in Europe: Designers' forum* (pp. 154-159). European Design and Automation Association.
7. Xue, L., Shi, F., & Ji, W. (2011). 3D floorplanning of low-power and area-efficient Network-on-Chip architecture. *Microprocessors and Microsystems*,*35*(5), 484-495.
8. Yu, X., & Li, T. S. (2009, October). On shortest path routing algorithm in crossed cube-connected ring networks. In *Cyber-Enabled Distributed Computing and Knowledge Discovery, 2009. CyberC'09. International Conference on* (pp. 348-354). IEEE.
9. Dobravec, T., Žerovnik, J., & Robič, B. (2006). An optimal message routing algorithm for circulant networks. *Journal of Systems Architecture*, *52*(5), 298-306.
10. Su, M. Y., Chen, G. H., & Duh, D. R. (1997). A shortest path routing algorithm for incomplete WK-recursive networks. *Parallel and Distributed Systems, IEEE Transactions on*, *8*(4), 367-379.
11. Qiu, K. (2008, December). An Efficient Disjoint Shortest Paths Routing Algorithm for the Hypercube. In *ICPADS* (pp. 43-47).
12. Jha, S. K., & Jana, P. K. (2011). Shortest Path Routing on Multi-Mesh of Trees. In *Proceedings of the World Congress on Engineering* (Vol. 2).