

Triangle Hyper Hexa-cell Interconnection Network A Novel Interconnection Network

Asmaa Aljawawdeh¹, Esraa Emriziq², Saher Manaseer³

¹Computer Science Department
The University of Jordan

²Business Information Technology Department
The University of Jordan

³Computer Science Department
The University of Jordan

Abstract—The interconnection networks play the main role in many applications, because it has a direct influence on it. Nowadays; the challenge is to find suitable topology that can deal with fewer requirements and min-cost. One of the most famous interconnection structures is the cube; it is used to build many interconnection networks such as the cube Hyper hexa cell topology. This work proposes a new topology; a hybrid topology between hyper hexa cell topology and triangle topology. In the propose interconnection network, the focus on the diameter which is less than the cube hyper hexa cell within one in any dimension and this effect on many parameters such as execution time. In the simulation environment, the radix sort being applied on the suggested Interconnection network using dimension number two on both; Triangle Hyper Hexa-cell and Cube Hyper Hexa-cell. Depending on the comparison between both topologies; in Theoretical and practical. The result shows the best performance for Triangle Hyper Hexa-cell. Practically; the measured parameter was the execution time in the simulation environment. Theoretically, the topological properties for both have been measured and got the equations for both, such as: number of nodes in every dimension, the number of edges, the network degree, and the diameter. This architecture will promise to be useful, more powerful for new-generation parallel architectures, and more effective for many applications and can be applied in different fields.

Keywords—Interconnection network; hyper hexa-cell; parallel system; radix sort; triangle hyper hexa-cell; triangle topology

I. INTRODUCTION

Parallel computing is one of the most trending area in computer architecture nowadays. The parallel computing means to compute simultaneously the jobs, tasks or instructions which reflects on performance criteria. The principle of parallel is to divide job into small jobs and start sub-jobs simultaneously[1]. Interconnection network provide ways to send and receive data between nodes processing. The parallel structures interconnections are divided into two well-known classifications which are [2][3]: 1- Static (direct connection). 2- Dynamic (intermediate stages). One of the most significant application that depends on the topology design is the Processors. The Processors are designed as groups, wherein each group contains processors connected along short distance networks such as the hypercube, Bus and Ring [4][5]. The main key in any interconnection network performance is the topology that used to build it [4][2][6]. One of the most famous interconnection networks is the cube-based architectures, that

being widely used according to its benefits in structure, such as low diameter, low cost and less complexity [6]. Some other familiar and well-known interconnection networks are: Mesh, Torus and hyper cube that used specially in digital communications systems[5][7]. Parallel systems are being used widely nowadays, this led to introduce new interconnection networks to support parallel processing. In this research paper we will introduce anew interconnection network by using a combination between two interconnection network.

II. RELATED WORKS

In [4] the authors used all-reduce communication operations to evaluated and analyzed on three interconnection network (the single port OTIS-Mesh, all port OTIS-Mesh and all-port-EDN-OTIS-Mesh) on different number of EDN (Extended Dominating Node), all OTIS-Mesh achieve better results for max and min number of communication than both single-port and all-port-OTIS-Mesh, the worst results in their proposed technique show performs about 146 and 142 better than other techniques. One of the disadvantage in this research appears on poor latency and the authors didn't include various collective communication operations such as scatter, reduction or another type of OTIS approaches like Hyper Hexa-cell and OTIS-Hyper cube.

In [2], the authors designed and implemented new topology OTIS Hyper Hexa Cell (OHHC) over OTIS-Mesh to integrate strength point for both HHC and OTIS characteristic, then evaluated for various terms such as diameter, minimum and maximum node degree, optical cost and bisection width. Their proposed topology showed improvement in terms and effectiveness for large parallel systems. One of the weakness in their approach that was no clear problem to solving such as load balancing, TSP and sorting algorithms.

In [1], the authors designed embedded hex-cell algorithm for n number of nodes in a tree-hyper cube network, which includes multiprocessor system. The proposed algorithm on tree-hyper cube can perform for irregular shape, but this study didn't perform for regular shape and show some weakness such as increase cost, dilation one, and congestion one and expansion. The authors suggested that their proposed algorithm on various topologies will reducing weakness for future works.

III. TOPOLOGICAL PROPERTIES OF TRIANGLE HYPER HEXA-CELL

This section first presents the important topological properties theoretical for both Triangle hyper Hexa Cell Topology and default Triangle Topology. Then a simulation was performed in order to measure the performance for both Triangle hyper Hexa Cell Topology and Cube Hyper Hexa Cell through applying the radix sort algorithm.

A. Topological Properties

The fundamental properties of Triangle Hyper Hexa-cell including the total number of nodes and edges, diameter, and cost factor.

- 1) Nodes:
Nodes defined as processing elements .The number of nodes in a Triangle Hyper Hexa-cell is given in Table I comparing with default Triangle topology.
- 2) Edges:
The edge is defined as the connection between two nodes.The number of edges for every node in a Triangle Hyper Hexa-cell is given in Table I comparing with triangle topology.
- 3) Degree of Network:
The degree of a node is defined as the total number of edges connected to that node. The degree of a network is defined as the largest degree of all the vertices that being used in the graph representation.
- 4) Diameter:
The diameter of a topology is defined as the shortest distance between the farthest two nodes in the topology.

Table I shows eight equations which measure and calculate the metric values for both topologies. The letter “d” in the below equations is representing the number of dimensions.

TABLE I. METRICS (RULES) FOR BOTH TOPOLGIES

Metrics	Triangle
No. of edges in every node	= d
No. of nodes	= (3 * (2d-1))/2
Degree of network	= (No. of edges * No. of nodes)/2 = d*((3*(2d-1))/2) /2 = d*((3*(2d-1)) /4
Diameter	= d-1
Metrics	Tria-HHC
No. of edges in every node	= d+3
No. of nodes	= ((3 * (2d-1) /2) *6 = 9 * (2d-1)
Degree of network	= (No. of edges * No. of nodes)/2 = ((d+3)* 9* ((2d-1)) /2
Diameter	= d+1

B. Topology Structure and Design

After defined the equations in the previous table, now we will determine the main properties for the proposed topology.

By using previous figures for the structure and rules in order to create and build the dimension number two structure for the Triangle Hyper Hexa cell (Tria-HHC). In this paper research, we use “Tria-HHC” name as an abbreviation for Triangle Hyper Hexa cell which is the proposed interconnection . For Triangle topology, Fig. 1(a), (b), (c) shows the structure and number of nodes in each dimension (two, three and four), respectively.

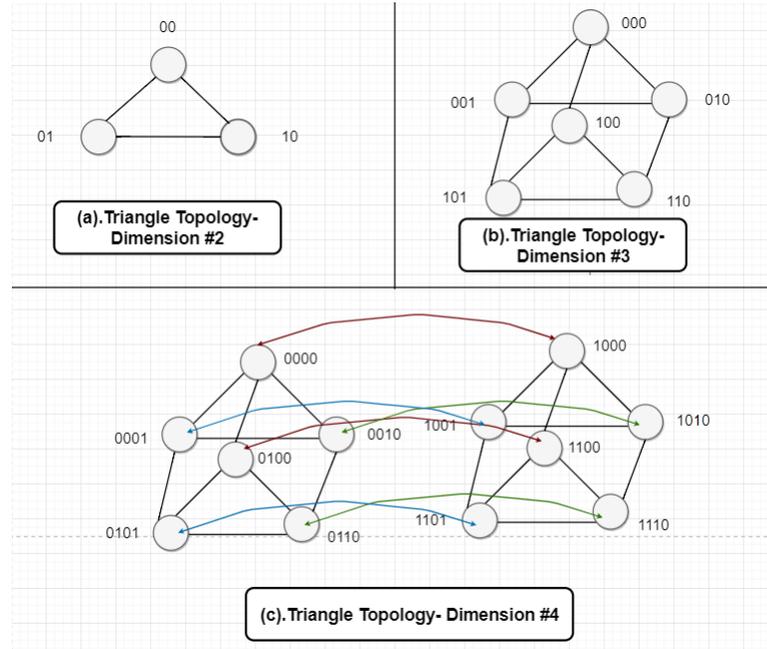


Fig. 1. Triangle topology.

The Triangle Hyper Hexa cell consists of two parts (Hyper Hexa cell and Triangle) as shown in Fig. 2 , which are being combined for building the new structure.

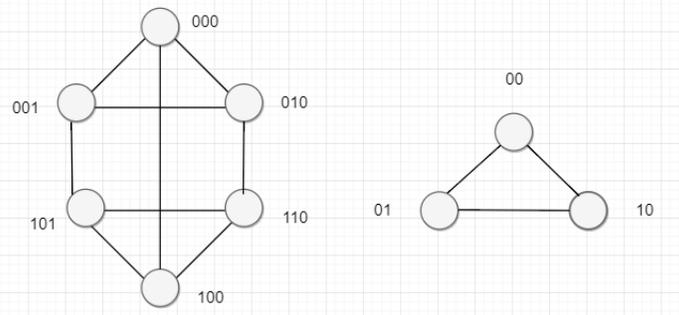


Fig. 2. Tria-HHC Parts

The Connection between nodes can be noticed in Fig. 3. Also, the metrics equations for dimension two achieved the real numbers by comparing the equations output with Fig. 3.

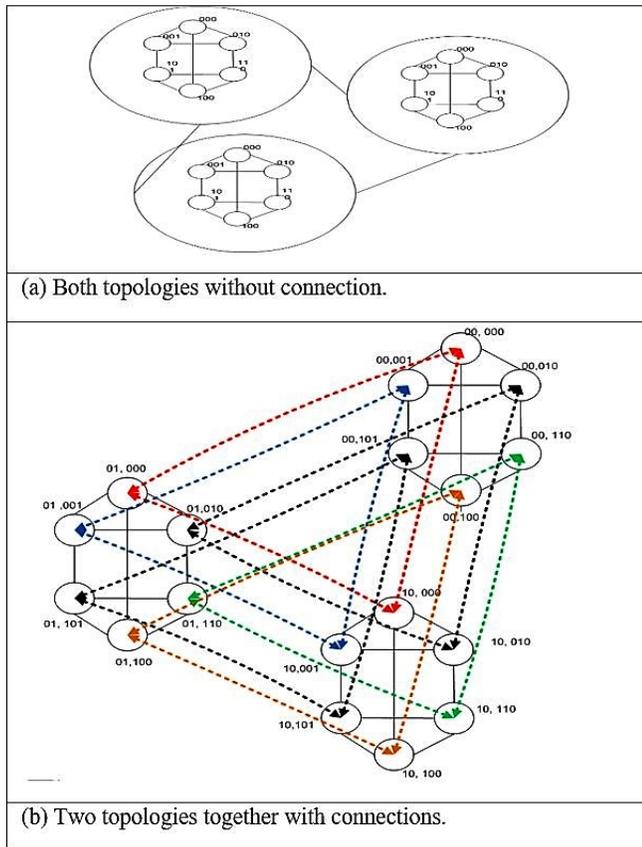


Fig. 3. Triangle Hyper Hexa cell (Tria-HHC)

The connection between nodes inside the hyper Hexa cell will be as it is, and between every two hyper hexa cell will be a connection for same nodes position(opposite) which basically depends on Triangle node's connections. Even though we must be careful about the number labeling.

C. Labeling and Naming

The naming rule for every node in the new hybrid interconnection is to make two parts of labeling separated by a comma. The first part of the name label to determine the External connection node name which is on the left side of a comma and it's defined as the group name. The other one to determine the internal connection, which is the same like default Hyper Hexa Cell and its existence on the right side. The two parts separated by a comma. The left part will change in every dimension. The number of bits will be the same number for dimension number, for example, if we have dimension two, in order to label the external part for the nodes we will add two bits (which will be the value of the Triangle label in fact) to the left plus a comma with its internal label as shown in Fig. 3 and so on for other dimensions.

D. Radix Sort

Sorting is one of the basic algorithms of computer science field and it is the most important part in many computation problems. Sorting plays a great role in many large-scale applications. Among successful sorting algorithms proposed

in the last few decades, the radix sort is best suitable for many applications and highly effective for parallel topologies [8]. A Sorting algorithm takes input a sequence of elements, and it outputs a permutation of the sequence in order. Radix sort is an integer sort, which is a sorting algorithm that sorts all elements numbers based on a single digit at a time[9]. There are two types of radix sorting: Most significant digit (MSD) radix sort starts sorting from the beginning of strings (most significant digit). LSD radix sort starts sorting from the end of strings (least significant digit) [10]. The Radix sort type which used in this simulation is the Most significant digit (MSD) version.

E. Simulation Environment

1) Materials used :

The materials that being used as mentioned in Table II.

TABLE II. SIMULATION MATERIALS

	Properties	Other Properties
Laptop used	Lenovo ThinkPad L50	OS: Win 7 Ultimate - 32bit service pack 1, Memory: 2G CPU: Intel Core I5 Model: 2520M - 2.5GHz.
Programming Language	Asp.net - C # (Console)	Microsoft Visual Studio Professional 2010 /Version : 10.0.40219.
Dimension Used	Two	used for both topologies (HHC and Tria-HHC)
Algorithm	Radix Sort algorithm	

2) The dataset used:

Random numbers were being used and created by a Method that creates a range between 1 to 999. Every integer number in the array equals 4 bytes (at least using 600,000 in the array). For example: 600,000 = 2,400,000 bytes = 2.28882 MB.

Table III shows the size of the dataset that has been used in the simulation. The data size that has been used in the experiment were converted using a website online converter.

TABLE III. DATA SIZE

Data size for Integer array	Data size in Bytes	Data size in MB
600,100	2,400,400	2.2892
1,000,000	4,000,000	3.8147
1,500,000	6,000,000	5.72205
2,000,000	8,000,000	7.62939
2,500,000	10,000,000	9.53674

3) Data Distribution and Collected Data between Nodes:

The same concept rule for data distribution and data gathering being applied in both topologies (Cube hyper hexa cell and Triangle Hyper Hexa Cell). In every hyper hexa cell (Internal) we used a two nodes for distributing data, then the same two nodes (in reverse) gathering the result from nodes in reverse method of the data distribution method. The remaining four nodes are used for applying the radix sort algorithm on its partitioned data. In every

Hyper Hexa Cell, there will be two nodes for data distribution and the same for data gathering, those two nodes work between inside nodes and outside nodes. The remaining four nodes will be used for Sorting. Fig. 4 and Fig. 5 illustrated this process in both Internal and External Cube Hyper Hexa cell nodes. The same methods applied for Triangle Hyper Hexa-cell.

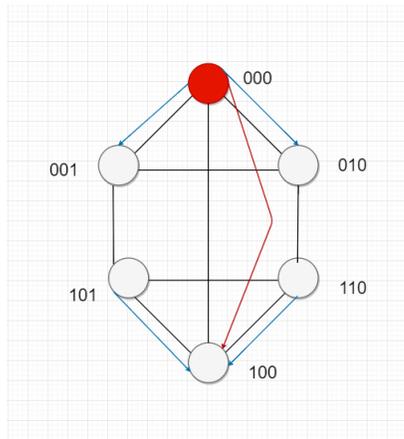


Fig. 4. Hyper Hexa cell -Distribution Data Internal

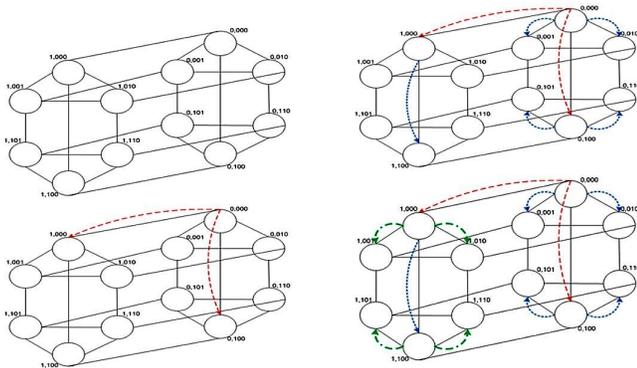


Fig. 5. Cube Hyper Hexa cell -Distribution Data External.

IV. RESULTS AND DISCUSSIONS

The collected results from the simulation after applying the radix sort on different structures are shown in Table IV which followed by a chart. The results show the time measurements (execution time) for the three structures: Triangle HHC, Cube HHC, and series (not parallel). The time execution is the computation time added to communication time. The time unit that is used in this simulation is Milliseconds. For validation, every experiment is repeated for 5 times and then calculated the average for those five experiments. It is clear that the series without parallel takes a lot of time, but by comparing the other topologies (Tria-HHC and Cube HHC), we can notice from Fig. 6 there is a difference in the Execution time. The performance for Tria-HHC is better than the Cube HHC depends on time measurements. Although there are some cases for time convergence, which could be explained through the increasing communication time between

nodes. Theoretically, depending on equations the Diameter for Tria-HHC is always less than one in every dimension comparing to Cube HHC. The best performance depending on equations, theoretical and simulation is the Triangle Hyper Hexa Cell (Tria-HHC).

TABLE IV. THE COLLECT RESULTS

DATA	600,100	1,000,000	1,500,000	2,000,000	2,500,000
Series	188.264	318.369	477.205	643.911	785.356
Tria-HHC	112.582	140.574	185.741	281.455	305.163
Cube-HHC	131.565	184.494	222.635	286.829	318.518

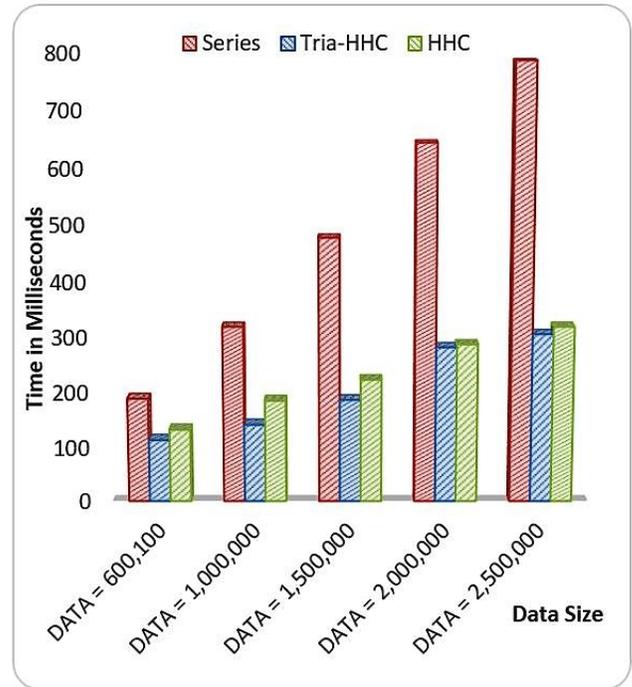


Fig. 6. Time measurement for Series, Cube HHC and Triangle HHC (Tria-HHC)

V. CONCLUSION AND FUTURE WORK

This paper presents a new interconnection topology called Triangle Hyper HexaCell. It consists of two topologies: Triangle and HyperHexaCell. The proposed topology is being useful in performance, according to its unique structure with a less diameter than the default Cube Hyper Hexa Cell. This will decrease the cost and the collected results show less time for Tria-HHC when applying the radix sort algorithm on the dimension number two into both cube HHC and Tria-HHC. There are routing algorithms which could be applied to this structure and other applications that depend on large-scale parallel computing systems will be efficient more using this topology. Future work, this topology can be applied in different fields and could be more useful. It is possible to combine three topologies together not only two.

REFERENCES

- [1] Qatawneh, A. (2013). Embedding Hex-Cells into Tree-Hypercube Networks. International Journal of Computer Science Issues (IJCSI), 10(3), 136-143.

- [2] Mahafzah, B. A., Sleit, A., Hamad, N. A., Ahmad, E. F., & Abu-Kabeer, T. M. (2012). The OTIS hyper hexa-cell optoelectronic architecture. *Computing*,94(5), 411-432.
- [3] Akhtar, A. H. & Lucas, K. T. (2014). Comparison of Communication Algorithms on OTIS-HHC and OTIS-Ring Parallel Architectures.
- [4] MAHAFAZH, B. A. and SERHAN, S. I. & TAHBOUB, R. Y. All-Reduce Communication Operation in OTIS-Mesh Interconnection Network.
- [5] Mohammad, Q. and Alamoush, A. and Basem, S. and Al Assaf, M. M. & Daoud, M. S. (2015). Embedding bus and ring into hex-cell interconnection network. *International Journal of Computer Networks & Communications (IJCNC)*, 7(3).
- [6] Alam, M. & Varshney, A. K. (2015). A Comparative Study of Interconnection Network. *International Journal of Computers and Applications*, 127(4), 37-43.
- [7] Yadav, S. & Krishna, C. R. (2014, August). CCTorus: A New Torus Topology for Interconnection Networks. In *Int'l Conference on Advanced Computational Technologies & Creative Media (ICACTCM' 2014)* (pp. 8-14).
- [8] Liu, X. & Deng, Y. (2014, December). Fast radix: A scalable hardware accelerator for parallel radix sort. In *Frontiers of Information Technology (FIT)*, 2014 12th International Conference on (pp. 214-219). IEEE.
- [9] Horsmalahi, P. (2012). Comparison of bucket sort and radix sort. arXiv preprint arXiv:1206.3511.
- [10] Polychroniou, O. & Ross, K. A. (2014, June). A comprehensive study of main-memory partitioning and its application to large-scale comparison- and radix-sort. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data* (pp. 755-766). ACM.