Explorations of Honeycomb Topologies for Network-on-Chip

Alexander Wei Yin, Thomas Canhao Xu, Pasi Liljeberg, Hannu Tenhunen
Department of Information Technology
University of Turku
Turku, Finland
{yinwei, canxu, pakrli, hatenhu}@utu.fi

***Abstract*—Rectangular mesh and torus are the mostly used topologies in network-on-chip (NoC) based systems. In this paper, we quantitatively illustrate that the honeycomb topology is an advantageous design alternative in terms of network cost which is one of the most important parameters that reflects both network performance and implementation cost. Comparing with the rectangular mesh and torus, honeycomb mesh and torus topologies lead to 40% decrease of the network cost. Then we explore the NoC related topological properties of both honeycomb mesh and torus topologies. By transforming the honeycomb topologies into rectangular brick shapes, we demonstrate that the honeycomb topologies are feasible to be implemented with rectangular devices. We also propose a 3D honeycomb topology since 3D IC has become an emerging and promising technique. Another contribution of this paper is the proposal of deadlock free routing algorithms. Based on either the concept of turn model or the logical network, deadlock free routing for all the discussed honeycomb topologies can be achieved.**

I. Introduction

Over the last forty years, the integrated circuit (IC) technology has provided the ability of integrating increasing number of elements in a planar form. As an emerging approach to large scale system-on-chip design, NoC borrows the concepts and techniques from the well established computer networks. This is due to the fact that in today's nanometer CMOS technology, interconnect dominates both the performance and power consumption since signal propagation in the wires often takes several clock cycle. The regular and well controlled network structure allows the NoC to reduce the design complexity for communication scheme while enhance the system predictability and reliability.

The key contribution of NoC lies in that it provides a communication infrastructure for the resources, i.e., it separates the communication from computation phase. Resources are connected to each other via the communication infrastructure which is an on-chip interconnect network.

The origin of topology in NoC system comes from the research field of parallel computing. However, not all the topologies in parallel computing are suitable for the NoC paradigm. Although high throughput and low latency are still desirable characteristics, other factors such as power consumption constraints and implementation feasibility have to be considered with equal, if not more, efforts. During the past years, a number of different NoC topologies have been proposed such as mesh, torus, spidergon, spin, ring, octagon, fat-tree, butter-fat-tree and other irregular topologies.

The network cost parameter for measuring the implementation feasibility of topologies is introduced in [1] and [2], as is shown in Equation 1. Degree refers to the number of channels entering and leaving each node in the network. If the degrees of nodes in a network are not identical, the network degree is determined by the maximum node degree. The diameter is the largest, minimal hop count over all pairs of terminal nodes in the network. Thus, in designing a topology, there is always a trade-off between degree, which relates to the hardware cost, and diameter, which relates to the message transmission time.

$$NetworkCost = Degree * Diameter \tag{1}$$

Mesh and torus are the mainstream topologies which constitute over 60% of the cases in NoC systems. This is because of their high regularity, symmetry and scalability. In this paper, we examine the honeycomb topology and its implementation feasibility in NoC based systems. By adopting the honeycomb topology, a huge proportion of network cost in NoC systems can be reduced, while maintaining the regularity, symmetry and scalability. We also propose several deadlock free routing algorithms for the honeycomb mesh and torus topologies.

During the recent years, three-dimensional (3D) IC has become a hot topic in both academia [3] and industry [4]. The 3D NoC is a cutting edge technology by integrating NoC systems in a 3D fashion. Containing multiple layers of active devices, 3D NoCs have the potential to enhance system performance. Beside this clear benefit, in 3D NoCs, package density can be increased significantly, power can be reduced via shorter communication links, and circuitry is more immune to noise [5].

In this paper, we extend the honeycomb topology into a 3D paradigm. Deadlock free routing algorithms for 3D honeycomb are also proposed.

The rest of the paper is organized as follows. Section II presents the related works. In section III and IV, 2D and 3D honeycomb networks and their properties are discussed respectively. The deadlock free routing algorithms are presented in section V. Section VI concludes the paper and presents the ongoing work.

II. Related Work

The two dimensional honeycomb network topologies were explored for the first time in [6]. The author proposed an efficient coordinate system for the honeycomb mesh and torus networks. Some topological properties of the honeycomb mesh were presented in this paper. The author also proposed a routing algorithm based on the broadcasting technique. However, this routing algorithm is not deadlock free and thus is infeasible for the on-chip networks.

The turn model was proposed in [7] in early 1990s as an efficient concept for deadlock free routing in rectangular mesh topology. The eight turns in the mesh topology can form two cyclical dependent circles; one is clockwise and the other is counterclockwise. The authors proved that deadlocks can be avoided by preventing the cyclical dependent circles. This is achieved by eliminating one of the four turns in each circle. In this paper, we adapt the turn model idea and use it in the honeycomb topologies.

Another method to prevent deadlock in NoC is based on the logical networks (LNs) which was proposed in [8] and implemented in [9]. The concept of LNs refers to an infinite set of associated (time slot, buffer) pairs with respect to a buffer on a given virtual circuit (VC). Each LN is logically separate from others and thus deadlock can be avoided.

The 3D honeycomb topology has been explored in [10] [11]. By adopting different complicated 3D structures, the authors proposed several honeycomb topologies which remained the degree of 3 even in 3D manner. However, these topologies are too sophisticated to be implemented in today's IC technology. Therefore, in this paper, we use a straight forward 3D topology by stacking the 2D honeycombs into different layers. Although it sacrifices the network degree from 3 to 4, its implementation feasibility is obvious.

III. Topology of 2D Honeycomb Network

A. Topology Overview

The two mostly used topologies in today's NoC research are the two-dimensional mesh (Figure 1(a))and torus (Figure 1(b)) network. A mesh topology is defined as a rectangularly aligned field of routers. Each router except for the ones on the boundary of the mesh is connected in four directions, namely, north, south, west and east. The torus topology is basically the same as a regular mesh [12]. The only difference lies in that the routers at the boundary are connected to the ones at the opposite boundary through wrap-around links.

Similarly, in honeycomb topologies, we can also define the mesh (Figure 1(c)) and torus (Figure 1(d)) alternatives. A honeycomb topology is composed of a number of hexagons. According to [6], the size of a honeycomb is defined as follows. The size of a single hexagon is 1 for a honeycomb mesh topology, which is denoted as HM_1. The HM_2 of size 2 is obtained by adding 6 hexagons on the boundary of HM_1. Inductively, HM_t is obtained by adding a ring of hexagons on the boundary of HM_{t-1}. Like in rectangular topologies, by adding a number of wrap-around links on the boundary, a honeycomb mesh can be turned into a honeycomb torus topology.

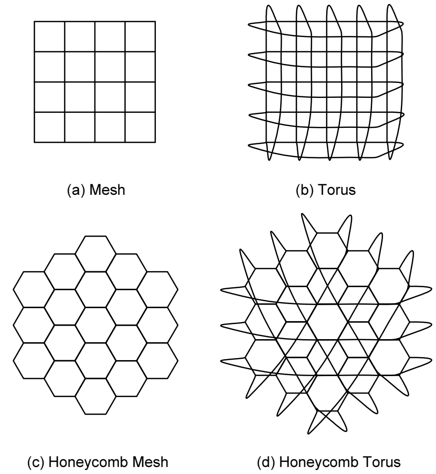

Fig. 1. Rectangular and Honeycomb Topologies

B. Topology Properties

The coordinate system used in this paper was proposed in [6]. Figure 2 shows the coordination axes on the honeycomb mesh topology. The X, Y and Z axes start from the center of the honeycomb and evenly divide the topology into three regions. The nodes with the same coordinate in a certain direction (X, Y or Z) form a zigzag chain which is vertical to the axis. Take the Z coordinate as an example for illustration. All nodes on each of the bold zigzag chains in Figure 2 have the same Z coordinate. Let the two central zigzag chains have the Z coordinate of 0 and 1 respectively, the six chains with respect to Z axis are z = {-2, -1, 0, 1, 2, 3}. The Z coordinate can be changed only by moving along the links which are parallel to the Z axis.

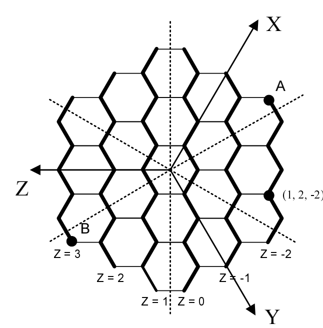

Fig. 2. Coordinate and Coding of Honeycomb Network

LEMMA 1. Nodes of HM_t can be coded by integer triples (x, y, z) such that $-t+1 \leq x, y, z \leq t$, and $1 \leq x+y+z \leq 2$ [6].

Proof for this lemma can be found in [6].

LEMMA 2. The distance between a pair of nodes (x, y, z) and (x', y', z') is $|x - x'| + |y - y'| + |z - z'|$.

According to Lemma 1, the honeycomb network is coded in three axes, X, Y and Z. Hops on each axis can affect the *coordinate only on this axis. Therefore, the Hamming distance is suitable to be used to define distance between two nodes in the honeycomb mesh topology.*

LEMMA 3. The diameter of HM_t is 4t-1.

According to the definition, diameter is the largest, minimal hop count over all pairs of terminal nodes in the network. In a synchronous on-chip interconnect, the distance between two nodes determines the minimal hop count. Thus, it can be observed from Figure 2 that the pairs of nodes which are on the opposite outermost border of the honeycomb, e.g. A and B, have the largest minimal hop count. The generalized coordinates of nodes A and B are $(t, 0, -t+1)$ and $(-t+1, 1, t)$ respectively. Therefore, based on lemma 2, the diameter H_{max} is $|t-(-t+1)|+|0-1|+|(-t+1)-t| = 4t-1$.

LEMMA 4. The number of terminal nodes in HM_t $N = 6t^2$. [6]

As is shown in Figure 2, a honeycomb mesh topology HM_t can be partitioned into six regions by the dotted lines based on the edges and vertexes of HM_1. It can be observed that in each region, the outermost border of HM_t has 2t-1 vertexes. Therefore, the total number of terminal nodes in HM_t is $\sum_{n=1}^{t} 6*(2t-1) = 6t^2$.

THEOREM 1. The network cost of honeycomb mesh topology which has N terminal nodes is $12\sqrt{\frac{N}{6}} - 3$.

As is mentioned before, the network cost is calculated as the product of network degree and diameter. In the honeycomb mesh topology, the network degree is 3. From lemma 3 and 4, the diameter is $4\sqrt{\frac{N}{6}} - 1$.

Similarly, the topology properties of honeycomb torus network are studied and analyzed.

THEOREM 2. The diameter of honeycomb torus topology HT_t which has N terminal nodes is $2t = 2\sqrt{\frac{N}{6}}$ and the network cost is $6\sqrt{\frac{N}{6}}$.

According to [2], the network costs of two dimensional mesh and torus topology are $8\sqrt{N} - 8$ and $4\sqrt{N}$ respectively. From the graphical impression shown in Figure 3, it is obvious that by using the honeycomb topology, the network costs can be reduced by approximately 40%.

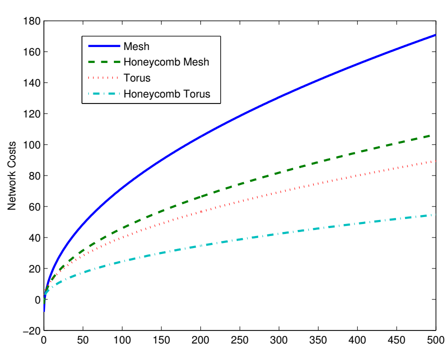

Fig. 3. Comparison of the Network Costs between rectangular and honeycomb topologies

C. Implementation Feasibility

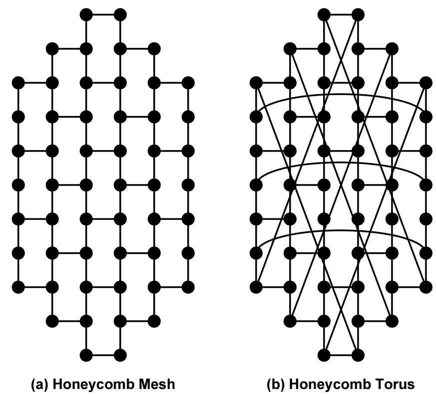

Fig. 4. Brick Transformations of Honeycomb Mesh and Torus

In today's IC technology, devices of rectangular shapes are dominating the fabrication industry. Therefore, NoCs which are of honeycomb shapes may not be the most efficient interconnect implementation in terms of placement, wire routing, layout and fabrication.

However, as is shown in Figure 4, both the honeycomb mesh and torus topologies can be transformed into rectangular shapes. By horizontally squeezing the honeycomb topologies towards the center and vertically drawing the zigzag chains into straight lines, the honeycomb mesh HM_3 (Figure 1 (c)) and honeycomb torus HT_3 (Figure 1 (d)) are transformed into the rectangular brick shapes which are shown in Figure 4 (a) and (b) respectively. These brick shapes not only have the implementation feasibility, but also keep all the topological properties and advantages of the honeycomb architectures.

IV. 3D Honeycomb Network

This section delves into the exploration of possible architectural frameworks for a three-dimensional honeycomb NoC topology. The basic 2D layer can be either a honeycomb mesh or a honeycomb torus architecture. A 3D honeycomb NoC is formed by vertically stacking a number of 2D honeycomb NoCs. Therefore, the bandwidth between processing elements in different layers can be significantly increased and the critical path can be shortened [13].

Figure 5 demonstrates a feasible and intuitive implementation of 3D honeycomb mesh network. Besides the X, Y and Z coordinates in each 2D layer, a vertical axis V is used to indicate that on which layer a processing element is located. Thus, each processing element can be labeled with an integer quadruple address (x, y, z, v). The V axis starts from the center of the bottom layer and hence this layer has $v = 0$. In this architecture, each router except for the ones on the top and bottom layers has two vertical communication links which are connected to its upper and lower immediate adjacent routers.

However, this architecture suffers from the high implementation cost due to the large number of vertical communication links. This architecture has the network degree of 5 which

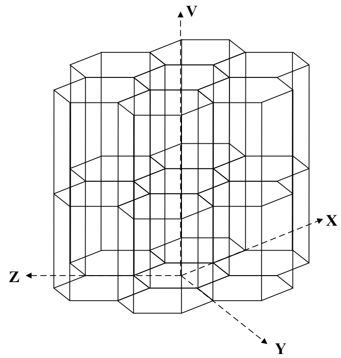

Fig. 5. Architecture of 3D honeycomb mesh network

leads to much higher network cost than that of the 2D honeycomb NoC. Moreover, in terms of the fabrication feasibility, the vertical links are much more expensive and area consuming comparing with the horizontal links [13]. Therefore, as is shown in Figure 6, an improved four-degree 3D honeycomb mesh architecture is proposed to reduce the number of vertical links.

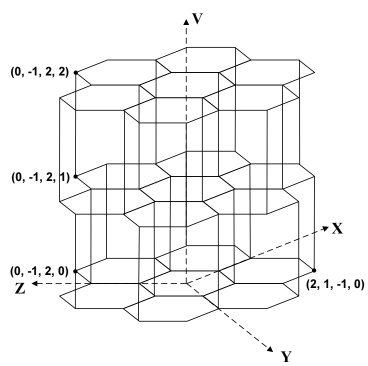

Fig. 6. Four-degree architecture of 3D honeycomb mesh network

In the four-degree 3D architecture, the network degree is reduced to 4 and half of the vertical links are removed by the bipartition of the routers into odd and even groups. A router belongs to the odd (even) group if the sum of its X, Y, Z and V coordinates is odd (even). The even nodes have vertical links to communicate with their upper layers and the odd nodes have vertical links to communicate with their lower layers.

V. Deadlock Free Routing Algorithms for Honeycomb Networks

In this section, we demonstrate the deadlock free routing algorithms for the honeycomb mesh, torus and 3D topologies. The deadlock freedom is achieved either via the turn model or via the logical networks. The variables and functions used in this section are listed in Table I.

Variable/Function	Description
t	size of the honeycomb
f.pos	current position of a flit
f.src	source address
f.dst	destination address
f.dir	next routing direction
f.LN	the LN a flit belongs to
f.cdir	direction to cross wraparound edge
minRoute(f)	send f to the next node according to minimal routing algorithm

A. Routing Algorithms for Honeycomb Mesh Topology

The turn model is one of the most common bases for deadlock free routing algorithms in rectangular network topologies. During our research, we have found that the turn model can be extended to the honeycomb topologies by eliminating one of the six turns in each cyclical dependent circle.

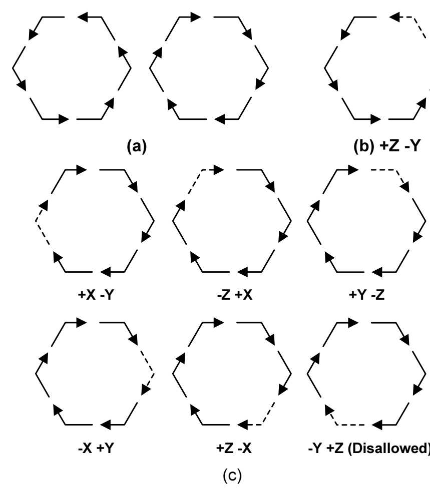

Fig. 7. Turn model based routing algorithm on honeycomb mesh topology. (a) Two cyclical dependent circles. (b) eliminate one turn in the counterclockwise cycle. (c) Possible turn eliminations in the clockwise cycle.

As is illustrated in Figure 7 (a), there are two cyclical dependent circles in a honeycomb mesh network; one is counterclockwise and the other is clockwise. Each circle is formed by six turns. The two sides of each turn are in the directions of X+, X-, Y+, Y-, Z+, Z- and the measure of the included angle is 120°. According to the turn model theory, deadlock free routing can be achieved if there is no cyclical dependent circles in the routing graph. Thus, one of the six turns in each circle need to be prohibited to break the cyclical dependency. Figure 7 (b) shows one of the prohibition possibilities in the counterclockwise circle. The "ring" formed by the other five turns is named as a "+Z -Y" turn since it is equivalent to a turn which starts from the +Z direction and ends in the -Y direction. Figure 7 (c) presents the six possible turn prohibitions in the clockwise circle. However, if we take the "+Z -Y" turn in Figure 7 (b) and "-Y +Z" turn in Figure 7 (c), as is shown in Figure 8, a new cyclical dependent circle which is of "∞" shape will be formed. But all other five

Algorithm 1: -X +Z first routing in honeycomb mesh

```
procedure mXpZ(mesh, flit):
for each flit f in honeycomb mesh:
    for(f.pos.X>f.dst.X or f.pos.Z<f.dst.Z)
        f.dir := (-X,+Z)
        do send f to next node
            if f reaches honeycomb border
                f.dir := Y
        until no (-X,+Z)
    minRoute(f)
    if f.pos == f.dst
        f.dir := NI
```

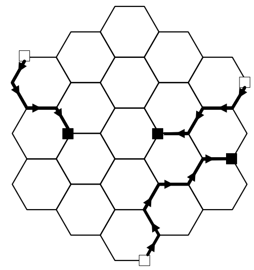

Fig. 10. Examples of the -X +Z first routing in a honeycomb mesh network

possibilities in Figure 7 (c) can be combined with the one in Figure 7 (b) to avoid the deadlock.

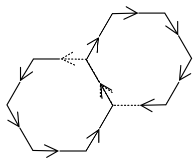

Fig. 8. The combinatorial cyclical dependent circle formed by two non-cyclical dependent rings

To illustrate the routing algorithm, we take the “+Z -Y” and “-X +Y” turns which are shown in Figure 9. This routing algorithm is named as -X +Z first routing. In a wormhole routing scheme, a flit firstly checks if the destination is on the -X or +Z side of its current position. If so, it is routed in the -X +Z zigzag chain until there is no more -X and +Z. Then it is delivered according to the minimal routing algorithm until it reaches the destination. In case a flit reaches the border before finishing all the -X and +Z, it will be routed in the Y dimension for 1 hop so that the routing in -X and +Z can be continued. If the destination is neither on the -X nor +Z side of its source position, the flit is routed by minimal routing algorithm from the beginning. When the flit reaches the destination, it is routed to the network interface (NI) and then forwarded to the processing element. Three example paths for the -X +Z first routing algorithm are shown in Figure 10 where the white squares represent source nodes, black squares represent destination nodes and the arrowed bold lines represent the routing paths. The pseudo code of -X +Z routing algorithm is shown in Algorithm 1.

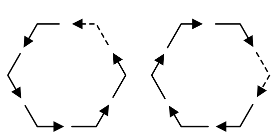

Fig. 9. The “+Z -Y” and “-X +Y” turns

Another method to achieve the deadlock free routing is based on the concept of LN. The LNs are created by using time-division-multiplexing (TDM) VCs. For each TDM VC, the link is virtually allocated for certain flits by reserving the related buffers in the routers at certain time slots. In the honeycomb mesh topology, we create two LNs on the X and Y dimensions; one is used to transfer flits towards the +Z direction while the other towards -Z. Flits are not allowed to “jump” from one LN to the other. Because there are two LNs for the links in X and Y dimension, time slots of the buffers in the routers are divided into odd and even categories. At the odd time slots, the links can be only used by the flits in one LN and at the even slots, they only serve the flits in the other. Apparently, the two LNs are logically separate and do not overlap with each other, thus the routing is deadlock free. Besides the deadlock freedom for the routing, LNs also enable NoC to provide guaranteed latency and bandwidth [8]. However, these benefits are not free. The adoption of LNs increases the complexities of routers by inserting more buffers and functionalities. Therefore, it is an application dependent trade-off whether to use LNs or not.

Algorithm 2 shows the routing procedure for the LN based routing algorithm. A flit first determines to which LN it belongs by comparing the Z coordinates of its source and destination addresses. If the Z coordinate of the destination address is not greater than that of the source address, the flit is routed in LN0. Otherwise, it is routed in LN1. The flits in LN0 can be sent in X and Y direction only at the odd time slots while those in LN1 can be sent only at even time slots. Take LN0 for example, a flit is routed in the negative direction in Z axis, but for X and Y axes, it can be routed in both positive and negative directions as long as the routing is executed at odd time slots.

B. Routing Algorithms for Honeycomb Torus Topology

For the honeycomb torus topology, similar to its mesh counterpart, both the turn model and LN based algorithms can be used to provide deadlock free routing. One of the key observations in honeycomb torus topology HT_t is that the coordination difference between two nodes on the wraparound edge is either $\pm(2t-1,-t,-t)$, $\pm(-t,2t-1,-t)$ or $\pm(-t,-t,2t-1)$.

The -X +Z first turn model based algorithm in honeycomb torus network is almost the same as that in honeycomb mesh except when a flit reaches the border before finishing all the -X and +Z. In a honeycomb torus network, the flit will be routed to the other end of the wraparound edge and continue the

Algorithm 2: LN based routing in honeycomb mesh

```
procedure LN_Routing(mesh, flit, LN):
for each flit f in honeycomb mesh:
    if f.dst.z <= f.src.z
        f.LN := 0
    else
        f.LN := 1
    for f.pos != f.dst
        if f.LN == 0
            f.dir := (X0,Y0,-Z)
            if time slot = odd number
                minRoute(f)
        if f.LN == 1
            f.dir := (X1,Y1,+Z)
            if time slot = even number
                minRoute(f)
    f.dir := NI
```

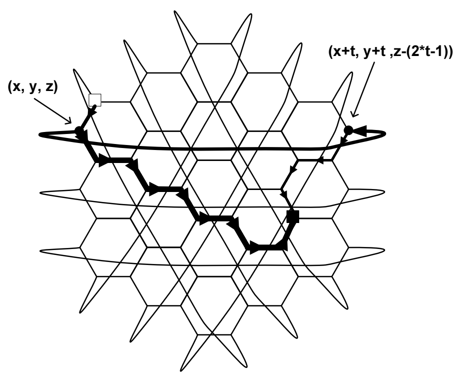

Fig. 11. An example of the -X +Z first routing in a honeycomb torus network

routing in -X and +Z directions. An example for this algorithm is shown in Figure 11 where the white square represents the source node, the black square represents the destination node and the arrowed bold line represents the routing path. A flit firstly reaches the border with coordinate (x, y, z), then it is routed to the node on other side of the wraparound edge with coordinate (x+t, y+t, z-(2t-1)) and continues the routing in the -X +Z directions until it reaches its destination. To distinguish the algorithms in honeycomb torus and mesh, the routing path which is in accordance with the -X +Z first routing algorithm in honeycomb mesh is highlighted by the bolder line in Figure 11. In this study case, the torus based routing algorithm also decreases the hop counts from ten to six. Algorithm 3 shows the pseudo code of -X +Z first routing in honeycomb torus network.

Like in the honeycomb mesh topology, the LN based routing algorithm is achieved by adding VCs on two of the three axes, and thus the honeycomb torus is divided into two logically separate networks. Due to its similarity to the one in honeycomb mesh topology, no further illustration is needed here.

Algorithm 3: -X +Z first routing in honeycomb torus

```
procedure mXpZ_torus(torus, flit, t):
for each flit f in honeycomb torus:
    f.dir := (-X, +Z)
    do send f to next node
        if f.pos == honeycomb border
            switch (f.cdir)
                case +X:
f.dir := (x+(2*t-1),y-t,z-t)
                case -X:
f.dir := (x-(2*t-1),y+t,z+t)
                case +Y:
f.dir := (x-t,y+(2*t-1),z-t)
                case -Y:
f.dir := (x+t,y-(2*t-1),z+t)
                case +Z:
f.dir := (x-t,y-t,z+(2*t-1))
                case -Z:
f.dir := (x+t,y+t,z-(2*t-1))
    until no (-X, +Z)
    minRoute(f)
    if f.pos == f.dst
        f.dir := NI
```

C. Routing Algorithms for 3D Honeycomb Topology

In 3D honeycomb topology which is shown in Figure 6, all of the above mentioned routing algorithms can be extended into the 3D structure to achieve the 3D deadlock free routing. The most important concept in this extension is the *mapping node* (MN). Nodes are called the MNs of another node only if they have the same coordinates in X, Y and Z axes but with different V coordinates. For example, in Figure 6, the node (0, -1, 2, 1) has two MNs (0, -1, 2, 2) and (0, -1, 2, 0) which are on the upper and lower layers respectively.

The deadlock free routing algorithms in 3D honeycomb is presented in Algorithm 4. The flit first compares the V coordinate of its current position with that of the destination node. If the two V coordinates are identical, it means that the flit is already on the destination layer and do not need any vertical routing. Then the flit can be routed to the destination node according to any of the above mentioned 2D routing algorithms. If the V coordinate of the current position is greater than that of the destination position, the flit should be routed to the lower layers. Since only the odd nodes have vertical links to connect the lower layer, in case the MN of the destination node on the current layer is an even node, the flit should be routed to the neighbors of the MN and then take a vertically downward hop to its lower layer. Thereafter, the flit will again compare its current V coordinate with the destination node and repeat the above process. Similarly, if the V coordinate of the current position is less than that of the destination position, the flit is firstly routed to the MN or its neighbors which is an even node and then take a vertically upward hop to its upper layer. The flit then again checks whether it is already on the destination layer.

Figure 12 shows an example of a 3D routing algorithm based on the -X +Z first routing in a 3D honeycomb mesh

Algorithm 4: 3D honeycomb routing algorithm.

```
procedure 3D_honeycomb(honeycomb, flit):
for each flit f in 3D honeycomb:
Step1:
    if f.pos.v == f.dst.v
        2D route to the dst
        f.dir := NI
    else
        go to Step2
Step2:
    if f.pos.v > f.dst.v
        if MN is even
            2D route to a neighbor of MN
            f.dir := -V
            go to Step1
        else
            2D route to MN
            f.dir := -V
            go to Step1
    elsif f.pos.v < f.dst.v
        if MN is odd
            2D route to a neighbor of MN
            f.dir := +V
            go to Step1
        else
            2D route to MN
            f.dir := +V
            go to Step1
```

network. The white square stands for the source node and black square represents the destination. Since the MN of the destination node on the top layer where the source node lies is an even node, the flit is first routed to node A which is one of the neighbors of the MN according to the -X +Z algorithm. It is then routed downward to node B which is on the middle layer. The MN of the destination node on this layer is node C which is an odd node. Therefore the flit is routed to node C and then to the destination by taking a downward hop.

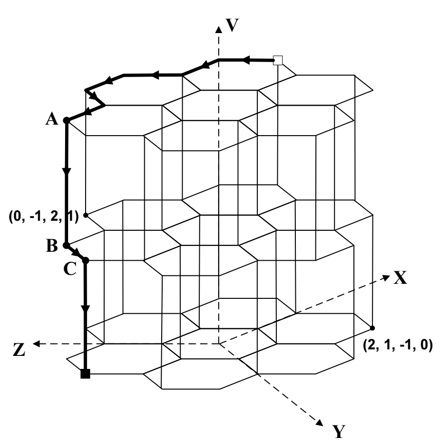

Fig. 12. Examples of the -X +Z first routing in a 3D honeycomb mesh network

VI. Conclusion and Ongoing Work

In this paper, we explore the honeycomb topology as an implementation alternative for NoC design. In two dimensional domain, comparing with the rectangular mesh and torus topologies, the honeycomb mesh and torus provide an approximate 40% reduction in terms of network cost. Therefore, we make an analytical topology exploration on the honeycomb topologies. Based on their characteristics, we propose deadlock free routing algorithms based on turn model and logical network. As three dimensional ICs become an emerging and promising technology in semiconductor field, we extend the 2D honeycomb topologies into a 3D structure by stacking them vertically layer by layer. The deadlock free routing algorithm for 3D honeycomb topology is also presented in the paper.

Compared with the rectangular topologies, the honeycomb mesh and torus have a smaller network degree which means a simpler router architecture. A router is more reliable and lower power consuming if it has a simple architecture. Therefore, our ongoing work of this paper includes the quantitative analysis, simulation and implementation of honeycomb topologies in respects of network performance, low power consumption and fault tolerance on different parallel computing applications.

References

[1] K. Efe, “A variation on the hypercube with lower diameter,” *IEEE Transaction of Computers*, vol. 40, no. 11, pp. 1312–1316, Nov. 1991.
[2] W. J. Dally and B. Towles, Eds., *Principles and Practices of Interconnection Networks*. Morgan Kaufmann Publishers, 2004.
[3] X. Dong and Y. Xie, “System-level cost analysis and design exploration for three-dimensional integrated circuits (3d ics),” in *Proc. Asia and South Pacific Design Automation Conference ASP-DAC 2009*, 19–22 Jan. 2009, pp. 234–241.
[4] A. W. Topol and et al., “Three-dimensional integrated circuits,” in *IBM Journal of Research and Development*, November 2006.
[5] B. S. Feero and P. P. Pande, “Networks-on-chip in a three-dimensional environment: A performance evaluation,” *IEEE Transaction on Computers*, vol. 58, no. 1, pp. 32–45, Jan. 2009.
[6] I. Stojmenovic, “Honeycomb networks: Topological properties and communication algorithms,” *IEEE Transaction on Parallel and Distributed Systems*, vol. 8, no. 10, pp. 1036–1042, Oct. 1997.
[7] C. Glass and L. Ni, “The turn model for adaptive routing,” in *Proc. 19th Annual International Symposium on Computer Architecture*, May 19–21, 1992, pp. 278–287.
[8] Z. Lu and A. Jantsch, “Slot allocation using logical networks for tdm virtual-circuit configuration for network-on-chip,” in *Proc. IEEE/ACM International Conference on Computer-Aided Design ICCAD 2007*, 4–8 Nov. 2007, pp. 18–25.
[9] A. W. Yin, *Generalization of slot table size for virtual circuits on Nostrum networks on chip*. Master Thesis in Royal Institute of Technology (KTH), Sweden, June 2008.
[10] C. Decayeux and D. Seme, “3d hexagonal network: modeling, topological properties, addressing scheme, and optimal routing algorithm,” *IEEE Transaction on Parallel and Distributed System*, vol. 16, no. 9, pp. 875–884, 2005.
[11] F. Garcia, J. Solano, I. Stojmenovic, and M. Stojmenovic, “Higher dimensional hexagonal networks,” *Journal of Parallel and Distributed Computing*, vol. 63, pp. 1164–1172, 2003.
[12] P. P. Pande, C. Grecu, M. Jones, A. Ivanov, and R. Saleh, “Performance evaluation and design trade-offs for network-on-chip interconnect architectures,” *IEEE Transaction on Computers*, vol. 54, no. 8, pp. 1025–1040, Aug. 2005.
[13] J. Kim, C. Nicopoulos, D. Park, R. Das, Y. Xie, N. Vijaykrishnan, M. Yousif, and C. Das, “A novel dimensionally-decomposed router for on-chip communication in 3d architectures,” in *Proceedings of ISCA-2007*, June 2007.