Simulation of Optimum topology and Routing Algorithm for 3D Network on Chip

Project Report submitted in partial fulfilment of the requirement for the degree of

Bachelor of Technology

Computer Science & Engineering

under the Supervision of

Dr. Vivek Sehgal

By

GAURAV SINGH (ROLL NO . 111258)

to

Jaypee University of Information and Technology

Waknaghat, Solan – 173234, Himachal Pradesh

Certificate

This is to certify that project report entitled "**Simulation of Optimum topology and Routing Algorithm for 3D Network on Chip**", submitted by **Gaurav Singh (Roll Number. 111258)** in partial fulfillment for the award of degree of Bachelor of Technology in Computer Science & Engineering to Jaypee University of Information Technology, Waknaghat, Solan has been carried out under my supervision.

This work has not been submitted partially or fully to any other University or Institute for the award of this or any other degree or diploma.

Date: 13-05-2015 **Supervisor's Name : Dr. Vivek Sehgal**

Designation : Astt. Prof.

Acknowledgement

I have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organization. I would like to extend my sincere thanks to all of them.

I am highly indebted to my project guide **Professor. Vivek Sehgal** for his guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in completing the project.

I would like to express my gratitude towards my parents & member of Jaypee University of Information Technology (JUIT) for their kind co-operation and encouragement which help me in completion of this project.

I would like to express my special gratitude and thanks to faculty for giving me such attention and time.

My thanks and appreciations also go to my friends in developing the project and people who have willingly helped me out with their abilities.

Gaurav Singh (Roll Number 111258)

Date: 13-05-2015 Name of the student

Table of Content

List of Figures

Abstract

Network-on-Chip (NoC) inter connection scheme is proposed as a unified solution for the design problems faced in SoC. NoC is an on-chip communication methodology proposed to resolve the increased interconnection problems in SoC. Network on Chip (NoC) is an appropriate candidate to implement interconnections in SoCs. Increase in number of IP blocks in 2D NoC will lead to increase in chip area, global interconnect, length of the communication channel, number of hops transversed by a packet, latency and difficulty in clock distribution. Despite the higher scalability and parallelism integration offered by Network-on-Chip.(NoC) over the traditional shared-bus based systems, it is still not an ideal solution for future large scale Systems-on-Chip (SoCs), due to limitations such as high power consumption, high cost communication, and low throughput. Recently, extending 2D-NoC to the third dimension (3D-NoC) has been proposed to deal with these problems. Topology, switching mechanism and routing algorithm are major area of 3D NoC research. In this report, I have discussed three topologies 3d Mesh Topology, 3d Star Topology, 3d Recursive Network Topology(3D-MT, 3D-ST and 3D-RNT) which are derived from their 2d versions and their corresponding routing algorithm for 3D NoC are presented. As 3D-NoC systems are exposed to a variety of manufacturing and design factors making them vulnerable to different faults that cause corrupted message transfer or even catastrophic system failures. Therefore,a 3D-NoC system should be fault tolerant to transient malfunctions or permanent physical damages. Therefore I have also discussed low latency, high throughput and fault tolerant routing algorithm named Look Ahead Fault Tolerant (LAFT). I have also discussed an efficient 3-D Asymmetric Torus routing algorithm for NoC. The 3-D torus has constant node degree, recursive structure, simple communication algorithms, and good scalability. A Quadrant-XYZ dimension order routing algorithm is proposed to build 3-D Asymmetric Torus NoC router. All the algorithms are simulated using my very own simulator and the algorithms are compared on the basis of latency, number of hops traversed , energy dissipation and other important factors.

CHAPTER 1

1. Introduction

According to Moore’s law, number of transistors per chip is doubled every two years that enables Integrated Circuit (IC) manufactures to provide more powerful electronic gadgets that derive multiple applications. Starting with 0.25 µm CMOS technology, wire delay dominates gate delay and the gap between wire delay and gate delay becomes wider as process technology improves, thus wires, not transistors are determining the performance of chips. Increase in number of transistors in a chip permits chip designers to integrate various components of an electronic system on a single IC to implement a complete System on a Chip (SoC) in which various components are named as cores or Intellectual Property (IP) blocks which include microprocessor, DSP, memory unit, I/O controller, analog signal or Radio Frequency module. The constraints like very short time to test, exploit reuse and market, force the designers to design SoCs with IP blocks which are designed by different IP vendors. Major challenge in SoC is interconnecting more number of IP blocks. Nowadays, on chip communications in SoCs are realized by direct cross bar interconnections and shared buses that are inefficient on performance, cost and reliability.

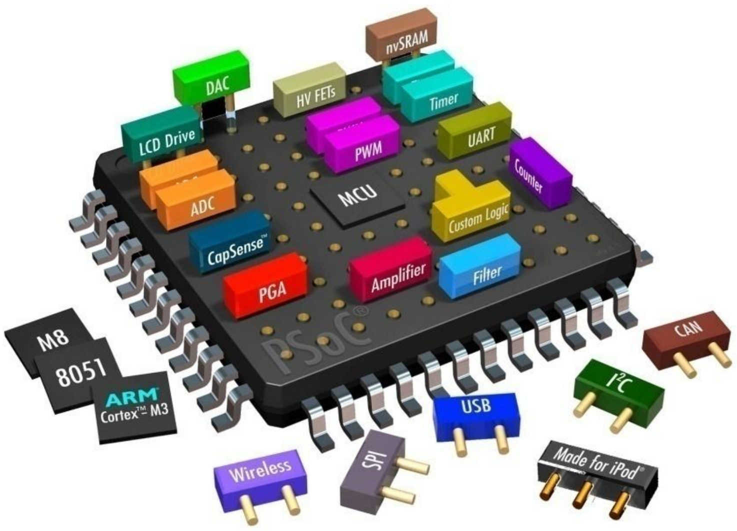


Fig 1: System on Chip (SOC)

Technology scaling has allowed **Systems-on-Chip** (SoCs) designs to grow continuously in component count and complexity. This significantly led to some very challenging problems, such as power dissipation and resource management. In particular, the interconnection network starts to play an important role in determining the performance and power of the entire chip .These challenges have led conventional bus-based-systems to no longer be reliable architectures for SoC, due to their lack of scalability and parallelism integration, high latency and power consumption, in addition to their low throughput.

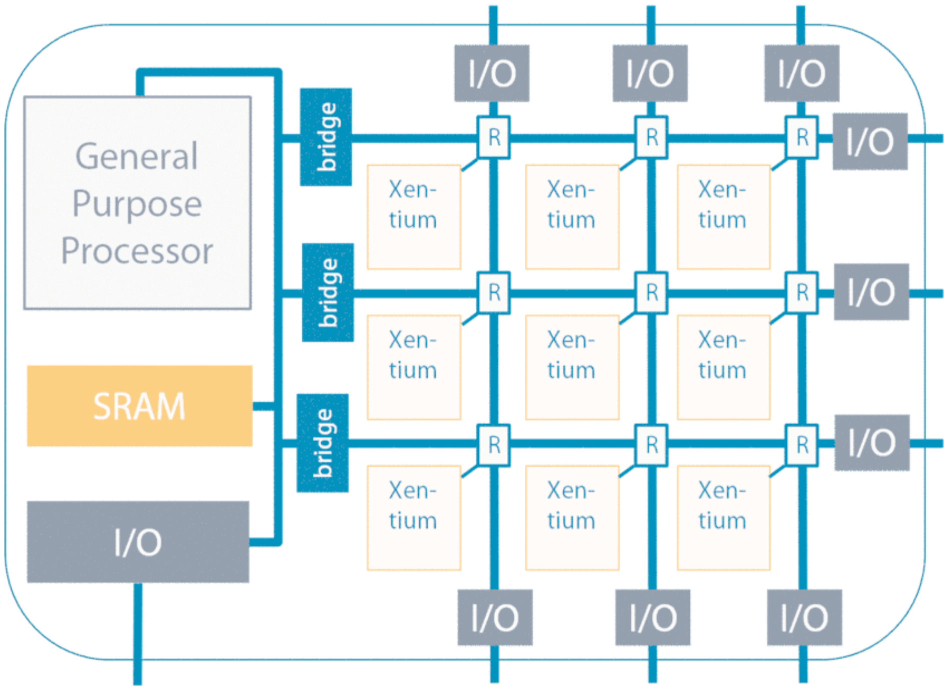


Fig 2 : Network on Chip (NOC)

Network-on-Chip (NoC) was introduced as a promising paradigm that can respond to these issues. Based on a simple and scalable architecture platform, NoC connects processors, memories and other custom designs together using switches to distribute packets on a hop-by-hop basis to increase the bandwidth and performance. At the same time, future applications are getting more and more complex, demanding a scalable architecture to ensure a sufficient bandwidth for any transaction between memories and cores, as well as communication between different cores on the same chip. This has made conventional 2D-NoC not suitable enough for future large-scale systems. One of the main limitations is the high diameter that 2D-NoC suffers from especially with a large network size. The diameter is an important parameter for NoC systems, since a large network diameter incurs a negative impact on the worst case routing latency in the network.

Considering all these facts, optimizing NoC-based architecture becomes extremely necessary, and several works have been conducted to achieve this goal. One of the proposed solutions is extending the 2D-Network-on-Chip to the third dimension. In the past few years, three dimensional integrated circuits (3D-ICs) have attracted a lot of attention as a potential solution to resolve the interconnect bottleneck. Thanks to the reduced average interconnect length, 3D-ICs can achieve higher performance and significantly lower interconnect power consumption. 3D ICs also make circuitry more resistant to noise ,and enable the realization of mixed technology. Combining the NoC structure with the benefits of the 3D integration offers a promising 3D-NoC architecture. This combination provides a new horizon for NoC designs to satisfy the high requirements of future large scale applications.

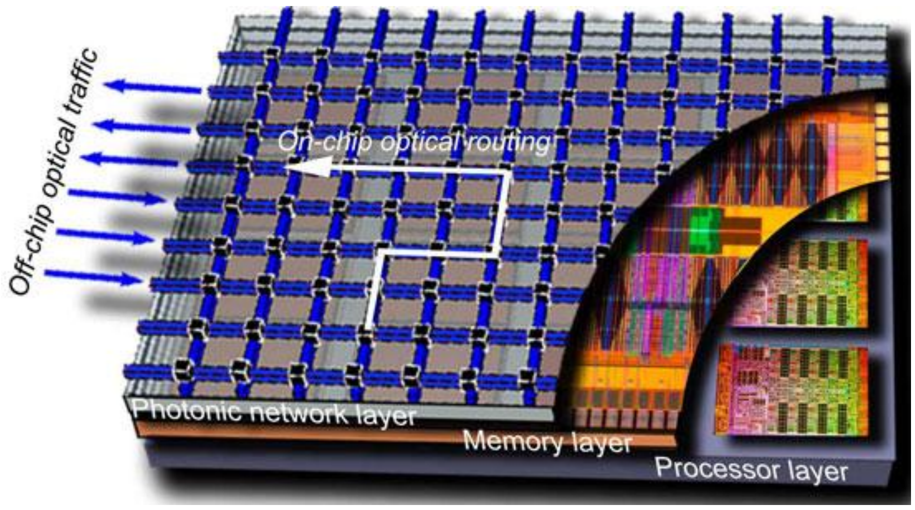


Fig 3 : (3D silicon processor chip with optical IO layer featuring on-chip nanophotonic network by IBM)

3D-Network-on-Chip architectures demand many trade-offs in order to meet some specific performance requirements. This raises various design challenges that have attracted a lot of attention, and extensive studies have been dedicated to tackle these challenges in different ways.

One of the challenges that can arise when designing a 3D-NoC is how to verify the system’s correctness and integrity. Most of the proposed 3D-NoC systems’ evaluations are based upon high-level simulation, where some used only synthetic traffic patterns and others used both synthetic and real workloads; however, the 3D-NoC architecture includes many trade-offs between topology, routing, flow control, buffer size, packet size, and other optimization techniques. It is difficult to analyze these trade-offs using only high-level simulations; therefore, prototyping is an essential design phase for evaluating the performance of 3D-NoC architectures under real applications.

Due to the complex nature of 3D-IC fabrics and the continuing shrinkage of semiconductor components, 3D-NoC systems are becoming increasingly vulnerable to failures caused by physical defects (permanent faults) and transient faults caused by some component failures. 3D-NoC systems are susceptible to many kinds of faults such as in routers, IPs, links etc. the majority of failures (80%) are caused by transient faults, while the rest of them originate mainly in permanent and intermittent faults. These kinds of faults should not cause a complete system failure as a safety requirement and 3D-NoCs should be able to run and deliver correct messages to their corresponding destination nodes, even with degraded performance. This can be done by either employing a fault tolerant mechanism that avoids or deactivates the faulty components or by reconfiguring the system without causing any important performance drop.

Many works have been undertaken to tackle the link failure problem especially by adopting fault tolerant techniques in the routing calculation phase. However, and as it is explained in the next section, all the already existing routing schemes suffer either from an unacceptable area overhead, or from additional latency due to the non-minimal routing approach adopted.

In order to boost the 3D-NoC systems while guaranteeing fault tolerance, routing algorithms should be minimal, congestion aware and with an acceptable extra hardware. Furthermore, some techniques to reduce the routing delay can be used for further performance improvement. One of these techniques that has shown great performance in 2D-NoC is look-ahead routing. Taking advantage of the high performance of look-ahead routing for fault tolerance can be a very promising solution for fault tolerant 3D-NoC architectures. It is discussed Later.

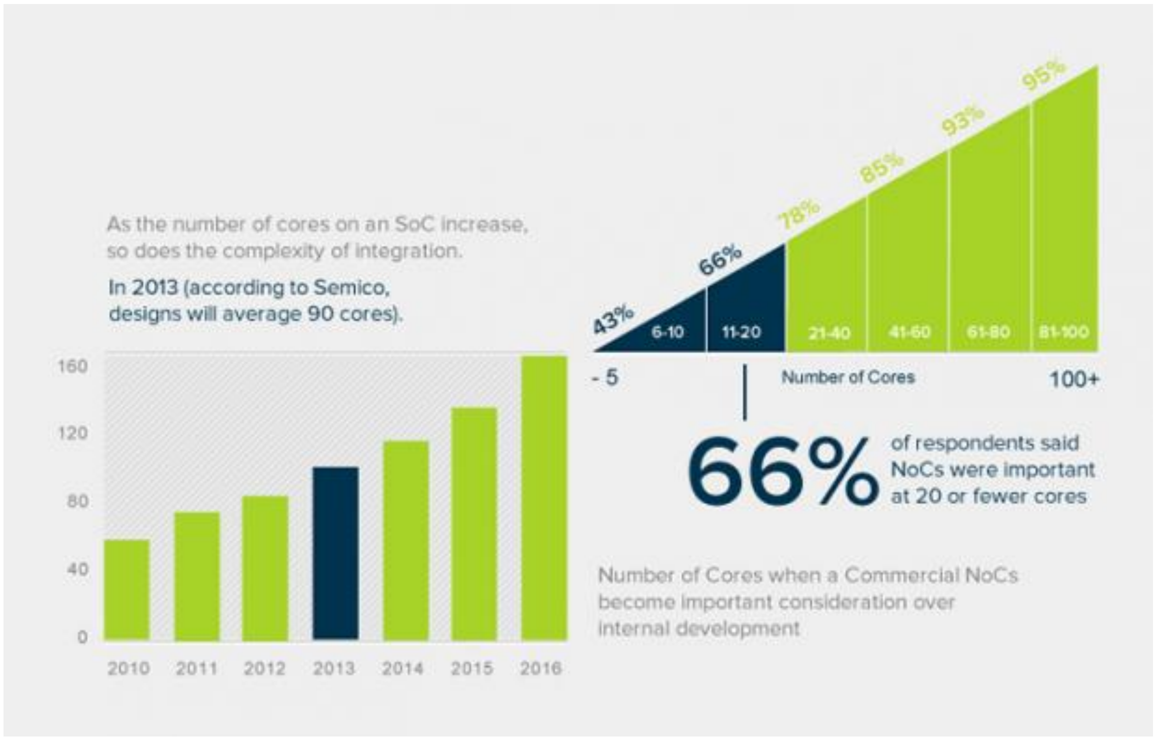


Fig 4: Noc with respect to number of Chores.

3D ICs allow for performance enhancements even in the absence of scaling. This is the result of the reduction in interconnect length. Besides this clear benefit, package density is increased significantly, power is reduced from shorter wires, and circuitry is more immune to noise . The performance improvement arising from the architectural advantages of NoCs will be significantly enhanced if 3D ICs are adopted as the basic fabrication methodology. The amalgamation of two emerging paradigms, NoC and 3D IC, allows for the creation of new structures that enable significant performance enhancements over more traditional solutions. With freedom in the third dimension, architectures that were impossible or prohibitive due to wiring constraints in planar ICs are now possible, and many 3D implementations can outperform their 2D counterparts.

1.1 Advantages of NoC over conventional crossbar interconnections and shared buses

• Wire segmentation and wire sharing design techniques are used to resolve the performance bottleneck caused by wire delay

• It uses a distributed control mechanism, resulting in a scalable interconnection network architecture Flexible and user-defined network topology

• Point-to-point connections and a Globally Asynchronous Locally Synchronous (GALS) implementation decouple the IP blocks

• Creating derivative products by easily adding and removing IP blocks from network

Research in 3D NoC is now emerging to realize on chip communications in 3D ICs. 3D integration is achieved by stacking a number of 2D layers. Interconnection of two neighboring 2D layers is accomplished using Through-Silicon-Vias (TSVs) which provide vertical channel through vertical interconnect links. This way, everything remains in 2D, except for the vertical links. These links can be integrated in the communication system by so-called 3D or vertical routers. Number of TSVs in an 3D architecture should be minimized as it has alignment problem and occupies a considerable chip area.

CHAPTER 2

2. 3D NOC ARCHITECTURE

Enabling design in the vertical dimension permits a large degree of freedom in choosing an on-chip network topology. Due to wire-length constraints and layout complications, the more conventional 2D ICs have placed limitations on the types of network structures that are possible. With the advent of 3D ICs, a wide range of on-chip network structures that were not explored earlier are being considered. It is challenging to design mixed signal chips which combine analog processing IP blocks, such as antenna or pixel arrays, with digital IP blocks, such as microprocessors and memories, in conventional planar chip-making processes. To overcome the challenge, analog IP blocks are kept on one layer, the digital IP blocks are placed on one or two other layers and combine them in a chip which is termed as 3-D IC.

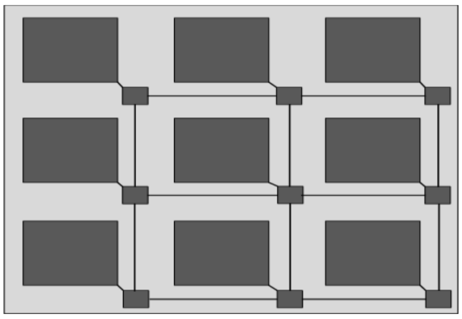


Fig 5: 2d mesh

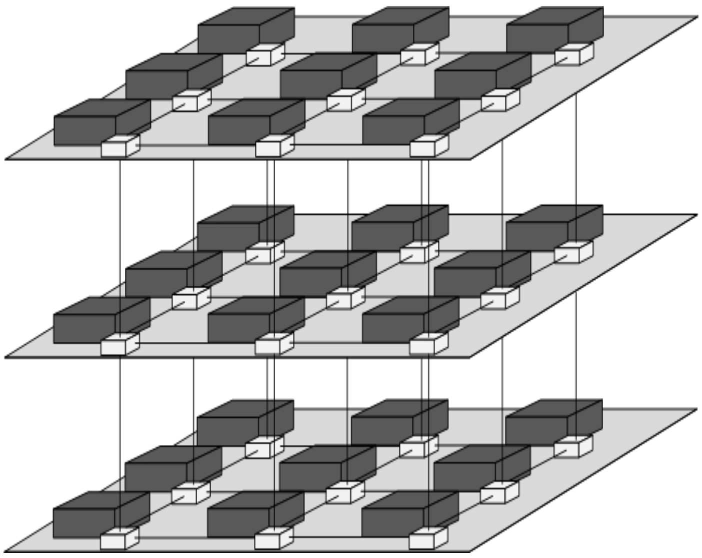


Fig 6: 3d Mesh

Advantages of 3d Noc.

2.1 Chip area

Minimization of chip area is important as the yield is in general increased. Not all circuits that are manufactured function properly. The yield is the percentage of correct circuits. Causes of failure, like crystal defects, defects in the masks, defects due to contact with dust particles are less likely to affect a chip when its area is smaller. The major advantage of 3-D IC is considerable reduction in chip length, resulting in a decrease in the chip area.

Total chip area = x^2 and network area = y^2, where x-chip length, y-network length. It is assumed that length x of 2D chip is 68µm, y is 64 µm and length of constant a is 2 µm. From Fig. 1, the following equations can be derived:

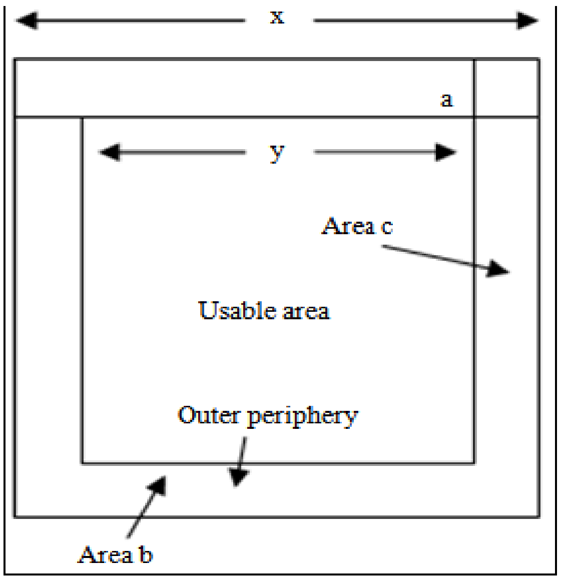


Fig. 7: 2D single layer chip area

$$\text{Area } b = x(x-y/2) \tag{1}$$

$$\text{Area } c = y\,(x-y/2) \tag{2}$$

$$\begin{aligned} P &= x^2 - y^2 = 2(b+c) \\ &= 2 * x(x-y/2) + 2 * y\,(x-y/2), \\ &\text{But } a = x - y/2 \\ P &= 2xa + 2ya \end{aligned} \tag{3}$$

$$x^2 = 2xa + 2ya + y^2 \tag{4}$$

Using Eq. 1 and 2, the area b and c can be calculated. Total outer periphery of the chip P can be calculated using Eq. 3 and 4 gives the total chip area. In order to reduce chip area, single layer is divided into multiple layers. Both length x and y are decreased as number of layers is increased. Chip length x reduction is not 50%, but it is only 47% as length a is constant when IP blocks are placed in two layers in lieu of placing in single layer. Similarly, there is no proportionate area reduction in 3D IC when number of layers is increased. It is concluded from Fig. 7 that length x is decreased as number of layers is increased. Trade off must be made between number of layers and chip area reduction in 3D IC as there is no proportionate reduction in length x.

2.2 Hop count

Pavlidis and Friedman (2007) have shown that average number of hops a packet transverses from source to destination node in 3D IC is:

$$\text{Hops} = \frac{n_1 n_2 n_3 (n_1 + n_2 + n_3) - n_1 n_2}{3(n_1 n_2 n_3 - 1)} \tag{5}$$

where, n1×n 2 is dimension of a layer and n 3 is number of layer. Multiplication of n1 and n2 gives number of nodes in a layer. Figure below shows that number of hops a packet transverses to reach destination from source node reduces when number of layers is increased.

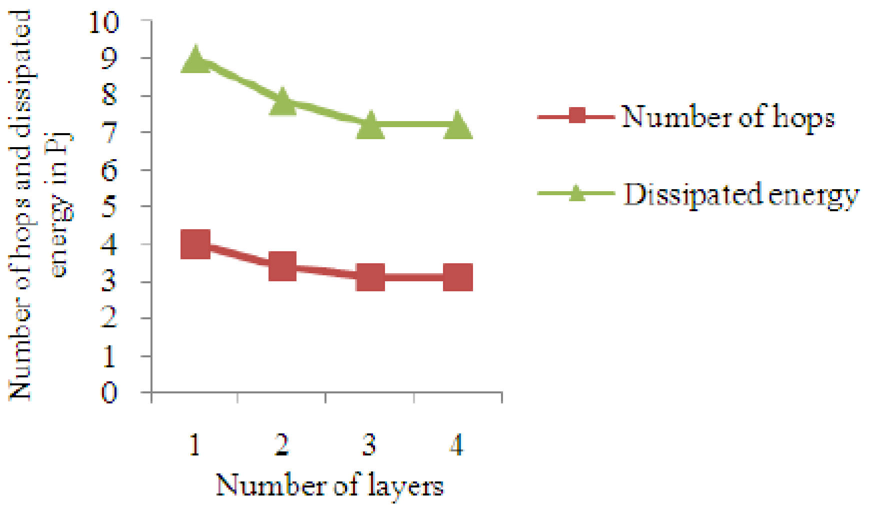


Fig 8: Number of hops and dissipated energy in 3D IC at different number of layers.

2.3 Energy dissipation

In the NoC paradigm, energy dissipation for interconnection of IP blocks depends on two independent parameters:

• Injected traffic load

• Energy dissipated in the switches and interswitch wire segments

Energy dissipation in switches and interswitch wire segments is considered here.

The following assumptions are made

• Uniform traffic patterns are used for message

• Length of wire segment between two switches is fixed

• Each switch consumes 1 Pico-joule (Pj) energy to process a packet

• Each interconnect wire segment consumes 1 Pico- joule (Pj) energy to transfer a packet

Energy dissipated by a packet for 1 hop:

$$E_{packet} = \sum_{1}^{n} E_{switch} + \sum_{1}^{n-1} E_{wiresegment}$$

$$\text{Where } E_{switch} = E_{wiresegment} + 1$$

$$E_{Packet} = (2 \times D) + 1$$

where, D is distance between source and designation node which is expressed in terms of hops:

$$\text{For n packets, } E_{packets} = \sum_{i=1}^{n} (2 \times D) + 1$$

where, total energy consumed to transverse a packet from source to destination node is represented by Epacket. Eswitch represents the energy consumed by both buffering and switching activities of a router and Ewire segment represents the energy consumed by charging and discharging of link capacitance. Each packet transverses (n+1) switches through n wire segments, thus hop count is n. It is considered that number of IP blocks to be placed is 36 and maximum number of layers is 4. From the fig – 8, it is concluded that average energy dissipation is reduced as number of layers is increased.

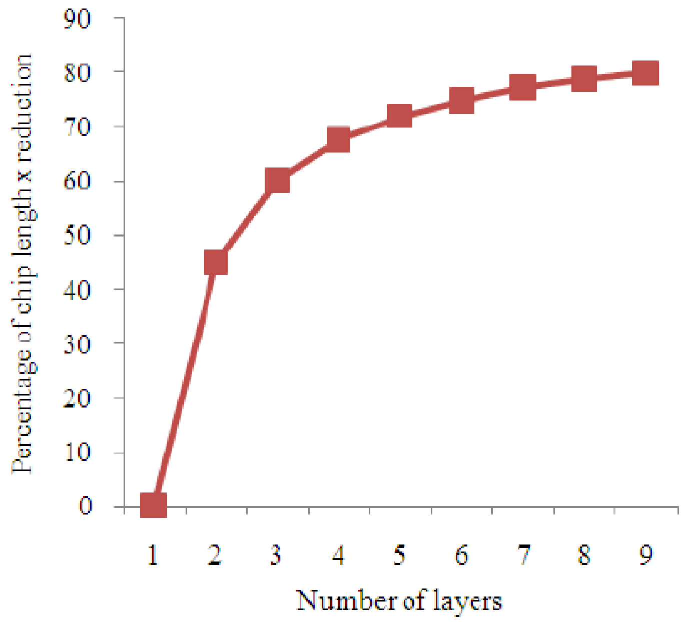


Fig 9: Chip length Reduction

In addition to chip area, hop count and energy dissipation, 3-D ICs have following advantages:

1) Layer yield decreases exponentially with increases in layer size, so splitting a single layer design into two or more can save money in the end
2) Increasing the number of transistors that are within one clock cycle of each other
3) Maximum global-interconnect length and the average global-interconnect length both decrease by a factor equal to the square root of the number of layers being stacked
4) Higher packing density

CHAPTER 3

3. Topology and Rounting Algorithm for 3D NOC

Components :

- IP blocks : includes microprocessor, DSP, memory unit, I/O controller, analog signal or Radio Frequency module.

- Router : that forwards data packets between networks.
- Node : A node in NoC comprises of an IP block and a router.
- Edges(Wire) : connection between nodes

Mesh, Star and WK recursive network topologies are very familiar topologies in 2D NoC. Three 3D topologies, Mesh Topology (3D-MT), Star Topology (3D-ST) and Recursive Network Topology (3D-RNT) are derived by modifying the 2D topologies. Three layers are considered in each topology in which IP blocks may be either homogeneous or heterogeneous as the topologies have more than one layer. In the topologies, cluster is formed by grouping four nodes with one node is identified as Cluster Head (CH) which can act as CH as well as node. A layer has four clusters, thus total number of nodes in a layer is sixteen (Feero and Pande, 2009; Loh, 2008).

IP blocks are connected to routers, in turn routers are interconnected using horizontal interconnect links. Vertical interconnect links (TSVs) are used to interconnect neighboring layer routers to form 3D network. Interlayer communications are realized only through CHs. CHs and other nodes can be identified by an ID of three digits XYZ. First digit X of the ID represents a layer, second digit Y represents a cluster and third digit Z represents either a node or CH. In 3DST, CHs in a layer are interconnected to communicate each other in single hop.

Intercluster nodes cannot communicate each other, they will communicate each other only through CHs. In 3D MT and 3D-RNT, CHs will communicate each other only through intercluster nodes that are allowed to communicate each other in single hop.

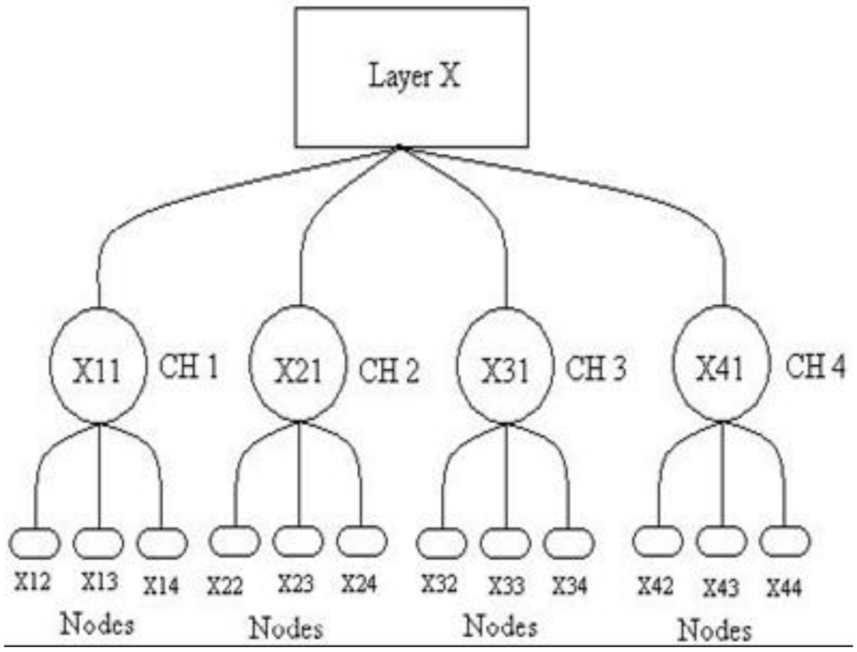


Fig 10: Tree representation of the topologies

Fig-11 shows packet format used to transfer message in the 3D network. The header flits have n bits in which first bit is End of Message (EOM) and the second bit is Start of Message (SOM). The rest of the bits (n-2) indicates ID of the source and destination nodes. Payload flits contain variable length message of maximum length 400-n bits as packet size is 50 bytes.

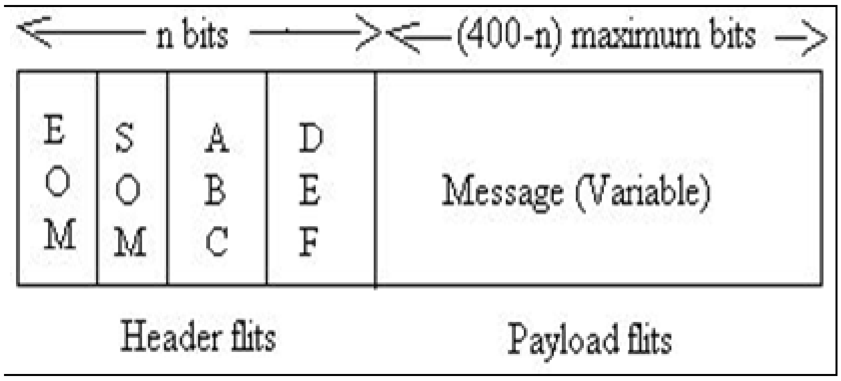


Fig 11 : Packet format

Hierarchy and clustering are two efficient Network techniques as they providescalability, Higher performance, easy maintainability, manageability and resource reusability, are applied in developing 3D routing algorithm.

3.1 3D Star Topology (3D ST)

The 2d version consists of a central node, to which all other nodes are connected; this central node provides a common connection point for all nodes through a hub. In **star topology**, every node (computer workstation or any other peripheral) is connected to a central node called a hub or switch.

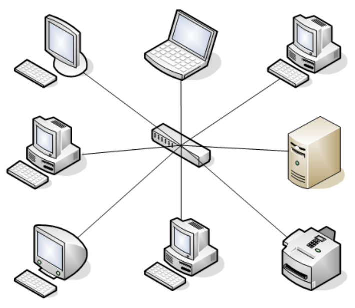


Fig 12: 2d Star Topology

Modified version of 2d star Topology and Each cluster is connected with other one using cluster heads only.

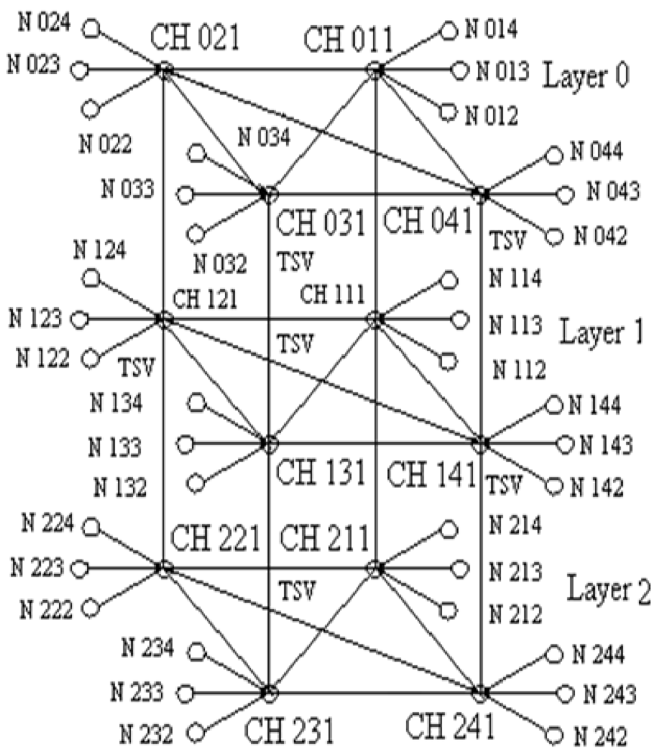
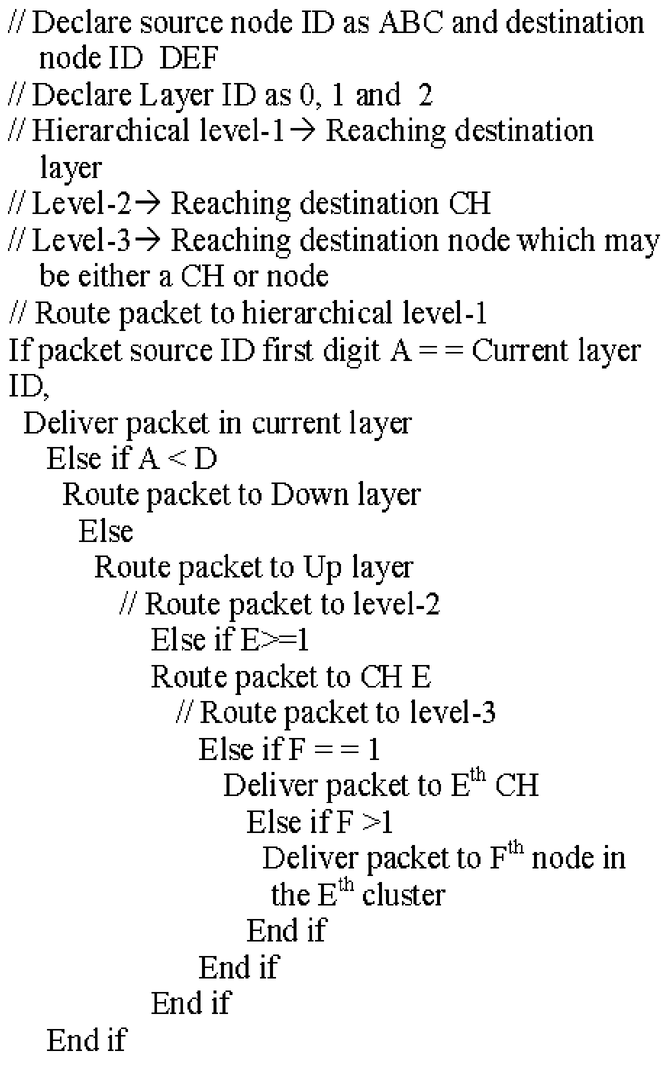


Fig 13 : 3D Star Topology With Clusters

PseudoCode for 3D ST routing Algorithm



```
// Declare source node ID as ABC and destination
   node ID  DEF
// Declare Layer ID as 0, 1 and  2
// Hierarchical level-1→ Reaching destination
   layer
// Level-2→ Reaching destination CH
// Level-3→ Reaching destination node which may
   be either a CH or node
// Route packet to hierarchical level-1
If packet source ID first digit A = = Current layer
ID,
 Deliver packet in current layer
   Else if A < D
     Route packet to Down layer
       Else
         Route packet to Up layer
            // Route packet to level-2
               Else if E>=1
               Route packet to CH E
                  // Route packet to level-3
                     Else if F = = 1
                        Deliver packet to $E^{th}$ CH
                           Else if F >1
                             Deliver packet to $F^{th}$ node in
                              the $E^{th}$ cluster
                           End if
                     End if
               End if
   End if
```

Fig 14 : Routing Algorithm for 3D – Star Topology

For 3D-ST, hierarchical shortest path is established between source node of ID 242 and destination node of ID 032.

- Source node ID → 242→ layer 2, cluster 4, node 2
- 241→ layer 2, cluster 4, node 1
- 141→ layer1, cluster 4, node 1
- 041→ layer 0, cluster 4, node 1
- 031→ layer 0, cluster 3, node 1
- 032→ layer 0, cluster 3, node 2

3.2 3D Mesh Topology (3D MT)

One of the well-known 2D NoC architectures is the 2D Mesh. This architecture consists of an m x n mesh of switches interconnecting IP blocks placed along with them. It is known for its regular structure and short interswitch wires. From this structure, a variety of 3D topologies can be derived. The straightforward extension of this popular planar structure is the 3D Mesh. Fig. 15b shows an example of 3D Mesh-based NoC. It employs seven-port switches: one port to the IP block, one each to switches above and below, and one in each cardinal direction (North, South, East, and West), as shown in Fig. 16a.

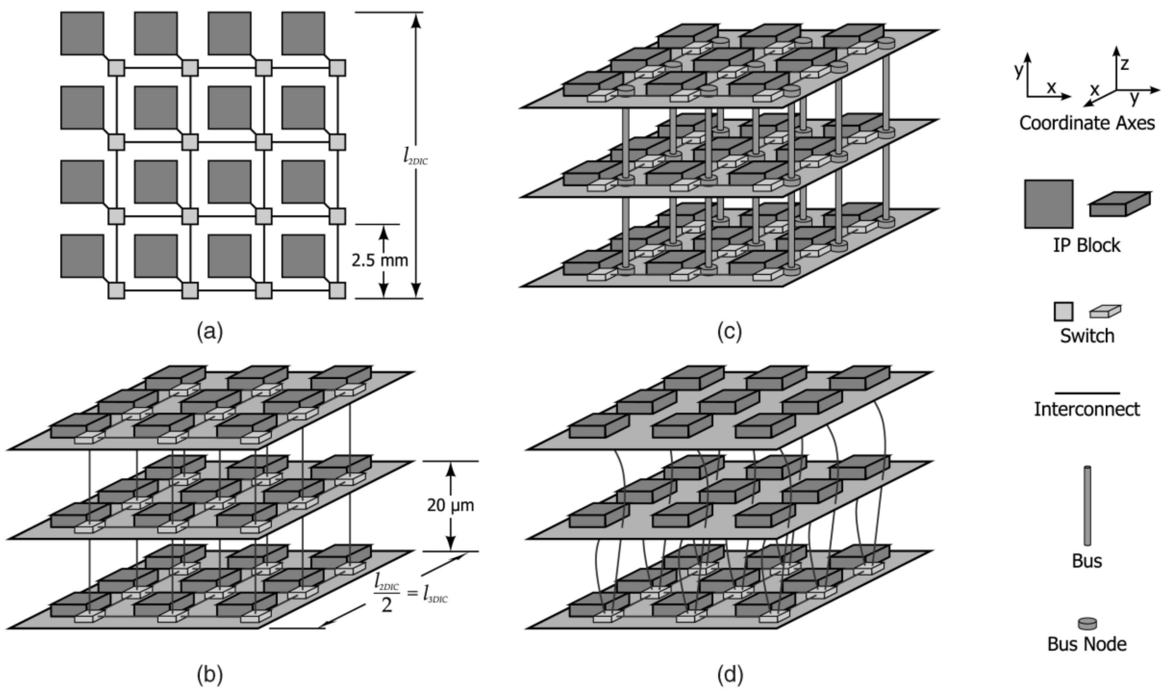


Fig 15 : Mesh-based NoC architectures. (a) Two-dimensional mesh.

(b) Three-dimensional mesh. (c) Stacked mesh. (d) Ciliated 3D mesh.

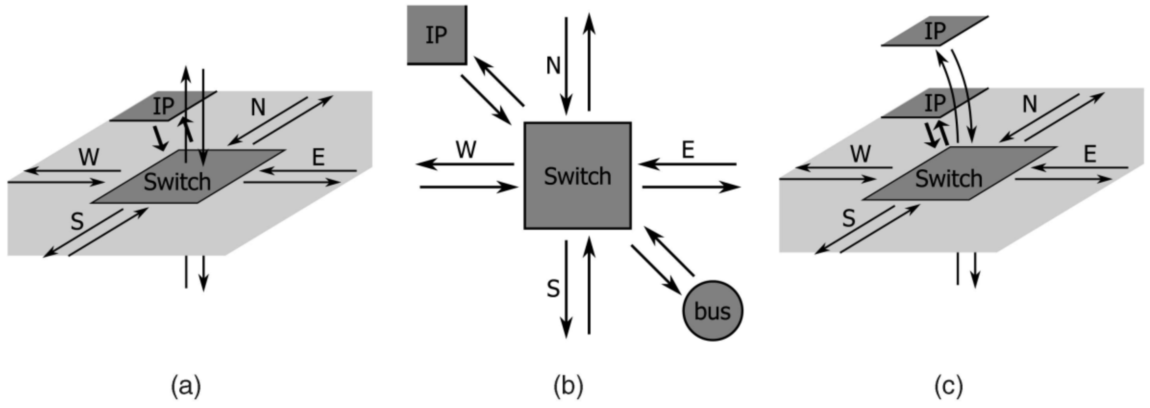


Fig 16: Switches for mesh-based NoCs. (a) Three-dimensional mesh. (b) Stacked mesh. (c) Ciliated 3D mesh.

A second derivation, Stacked Mesh (Fig. 15c), takes advantage of the short interlayer distances that are characteristics of a 3D IC. Stacked Mesh is a hybrid between a packet-switched network and a bus. It integrates multiple layers of 2D Mesh networks by connecting them with a bus spanning the entire vertical distance of the chip. As the distance between the individual 2D layers in 3D IC is extremely small, the overall length of the bus is also small, making it a suitable choice for communicating in the z-dimension. Furthermore, each bus has only a small number of nodes (i.e., equal to the number of layers of silicon), keeping the overall capacitance on the bus small and greatly simplifying bus arbitration. A switch in a Stacked Mesh network has, at most, six ports: one to the IP, one to the bus, and four for the cardinal directions (Fig. 16b).
A third method of constructing a 3D NoC is by adding layers of functional IP blocks and restricting the switches to one layer or a small number of layers, and we get ciliated 3D Mesh. This structure is essentially a 3D Mesh network with multiple IP blocks per switch. For the ciliated 3D Mesh. This is shown in Fig. 15d. In a ciliated 3D Mesh network, each switch contains seven ports (one for each cardinal direction, one either up or down, and one to each of the two IP blocks), as shown in Fig. 16c. This architecture will clearly exhibit lower overall bandwidth than a complete 3D Mesh due to multiple IP blocks per switch and reduced connectivity; however, this shows that this type of network offers an advantage in terms of energy dissipation, especially in the presence of specific traffic patterns.

Routing Algorithm :

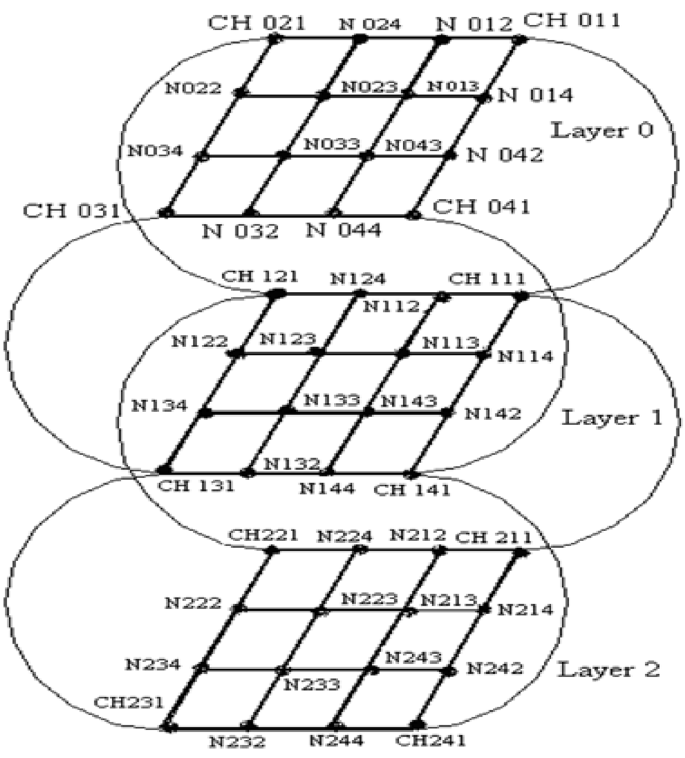


Fig 17 : 3D Mesh Topology

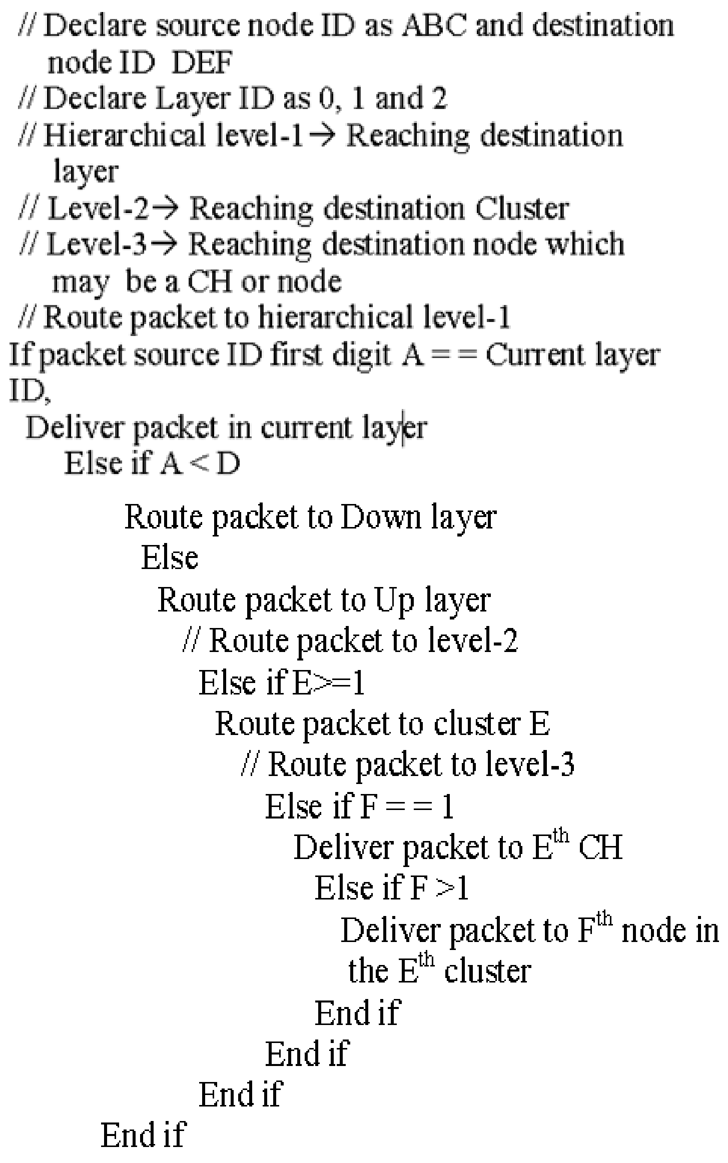
Above diagram show the 3d mesh topology and is used to show how routing algorithm works in this network.

Shortest path between nodes of ID 242 and 032 in 3D- MT is

242→241→141→041→044→032

- Source node ID → 242→ layer 2, cluster 4, node 2
- 241→ layer 2, cluster 4, node 1
- 141→ layer1, cluster 4, node 1
- 041→ layer 0, cluster 4, node 1
- 044→ layer 0, cluster 4, node 4
- 032→ layer 0, cluster 3, node 2

PseudoCode for 3D MT routing Algorithm



```
// Declare source node ID as ABC and destination
   node ID  DEF
// Declare Layer ID as 0, 1 and 2
// Hierarchical level-1→ Reaching destination
    layer
// Level-2→ Reaching destination Cluster
// Level-3→ Reaching destination node which
    may  be a CH or node
// Route packet to hierarchical level-1
If packet source ID first digit A = = Current layer
ID,
  Deliver packet in current layer
      Else if A < D
             Route packet to Down layer
               Else
                 Route packet to Up layer
                   // Route packet to level-2
                     Else if E>=1
                       Route packet to cluster E
                         // Route packet to level-3
                           Else if F = = 1
                             Deliver packet to E^th CH
                               Else if F >1
                                 Deliver packet to F^th node in
                                  the E^th cluster
                               End if
                           End if
                     End if
           End if
```

Fig 18 : Routing Algorithm for 3D –Mesh Topology

3.3 3D Recursvie Network Topology (3D RNT)

The 3D RNT is derived from WK recursive network topology of W (4, 2) where four additional flipping links are used in each layer except in the top and bottom layer of the topology and the topology has many favourable properties such as Hamiltonian connectedness, topology expandability without changing existing vertex degree, scalability, fault tolerance and symmetry. This network has a recursive definition quite similar to that of conventional pyramid Networks. Only 25% of the blocks in each layer are interconnected with the neighbouring layers using the vertical links.
It is a deadlock free routing algorithm to minimize congestion in the network.

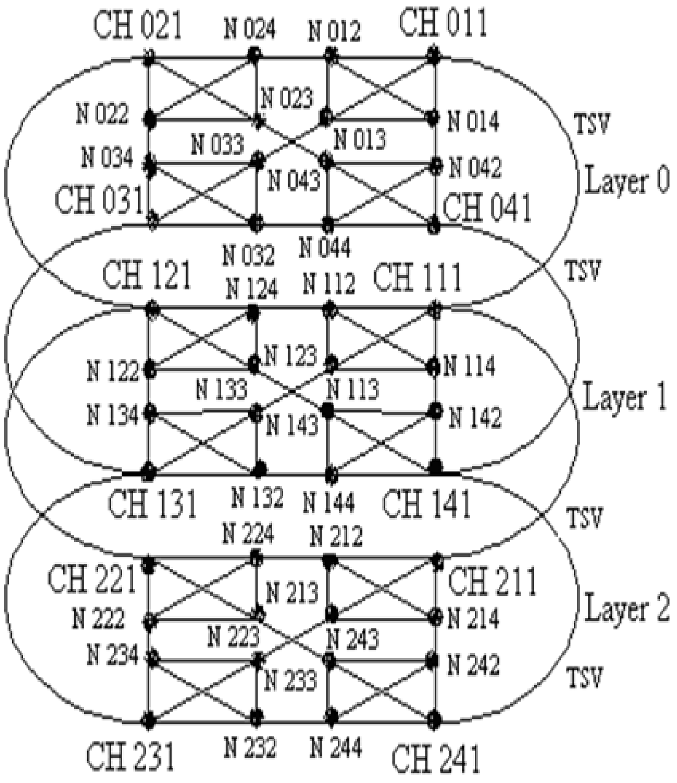


Fig 19: 3D Recursive Network Topology

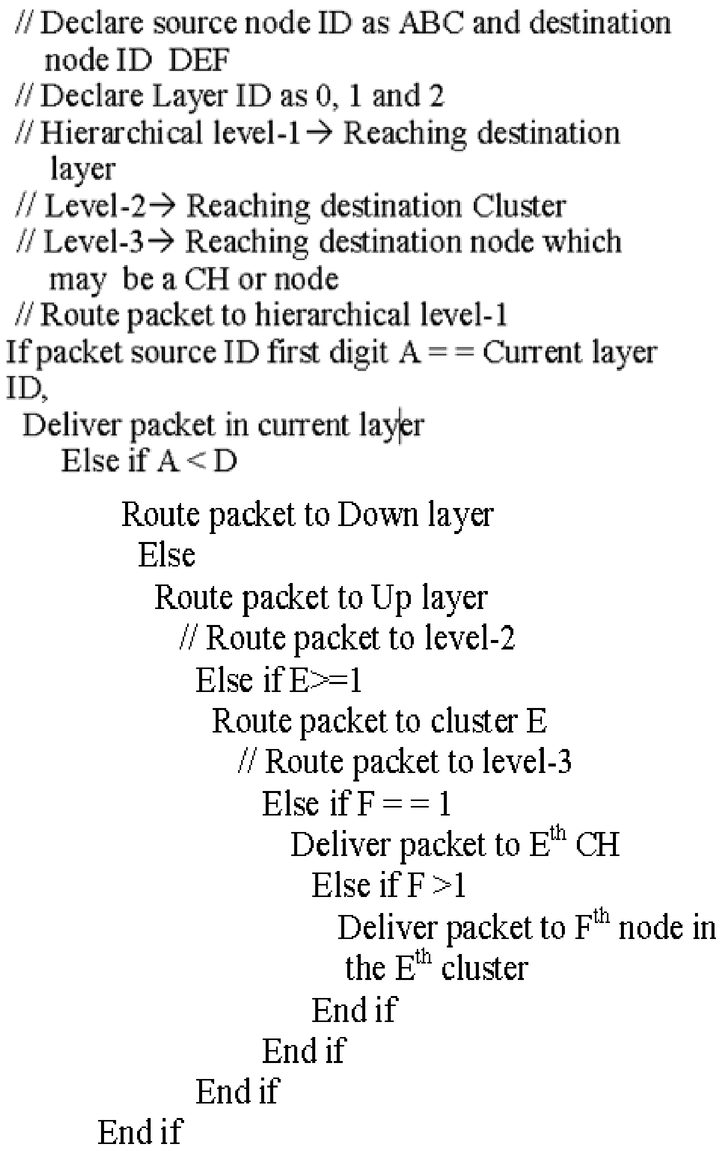
Shortest path between nodes of ID 242 and 032 in 3D- RNT is

242→241→141→041→044→032

- Source node ID → 242→ layer 2, cluster 4, node 2
- 241→ layer 2, cluster 4, node 1
- 141→ layer1, cluster 4, node 1

- 041→ layer 0, cluster 4, node 1
- 044→ layer 0, cluster 4, node 4
- 032→ layer 0, cluster 3, node 2

Routing Algorithm for 3D-RNT:



```
 // Declare source node ID as ABC and destination
    node ID  DEF
 // Declare Layer ID as 0, 1 and 2
 // Hierarchical level-1→ Reaching destination
    layer
 // Level-2→ Reaching destination Cluster
 // Level-3→ Reaching destination node which
    may  be a CH or node
 // Route packet to hierarchical level-1
If packet source ID first digit A = = Current layer
ID,
  Deliver packet in current layer
     Else if A < D
             Route packet to Down layer
               Else
                 Route packet to Up layer
                   // Route packet to level-2
                     Else if E>=1
                       Route packet to cluster E
                         // Route packet to level-3
                           Else if F = = 1
                             Deliver packet to E^th CH
                               Else if F >1
                                 Deliver packet to F^th node in
                                  the E^th cluster
                               End if
                           End if
                     End if
          End if
```

Fig 20 : Routing Algorithm for 3D – RNT Topology

3.4 Look-Ahead-XYZ Routing

Conventional XYZ routing compares the address of the processing node with the destination node's address to determine the Output-Port. The computed information is sent to the Switch-Arbiter requesting access of the selected output-port. Despite its simplicity, XYZ suffers from an inefficient pipeline stage usage, which can incur a high communication latency therefore degrading the system throughput [49].

Figure 21 (a) shows a conventional router pipeline design based on the XYZ scheme. Virtual Channels are not taken into consideration for improving the performance of best-effort traffic. This figure shows the router's four pipeline stages: Bu_er Writing (BW), Routing Calculation (RC), Switch Arbitration (SA), and the Crossbar Traversal stage (CT). This router's design increases the flit latency because any flit should go through all these stages at each hop while travelling from source to destination. This can introduce undesirable overall system performance
degradation, especially with a large network size where the network diameter scales. In these routing algorithms, the pipeline stages are dependent on each others, and each one of them cannot perform its computation unless it receives information from the previous stage. This dependency is especially seen between the RC and SA stages.

To solve this dependency problem, Look-Ahead-XYZ (LA-XYZ) parallelizes the RC and SA stages and eliminates the dependency between them. LA-XYZ pre-computes the Next- Port direction of the downstream router and then embeds it in the flit. When arriving to the downstream node, this hot encoded Next-Port identifier will be used by the Switch-Arbiter directly to request authorization for using the selected output-port and reaching the next neighbouring node. At the same time, when the grant is computed in SA, the RC calculates in parallel the direction of the Next-Port that will be used by the next downstream node. This parallel process optimizes the pipeline stages as shown in Fig.21 (b).

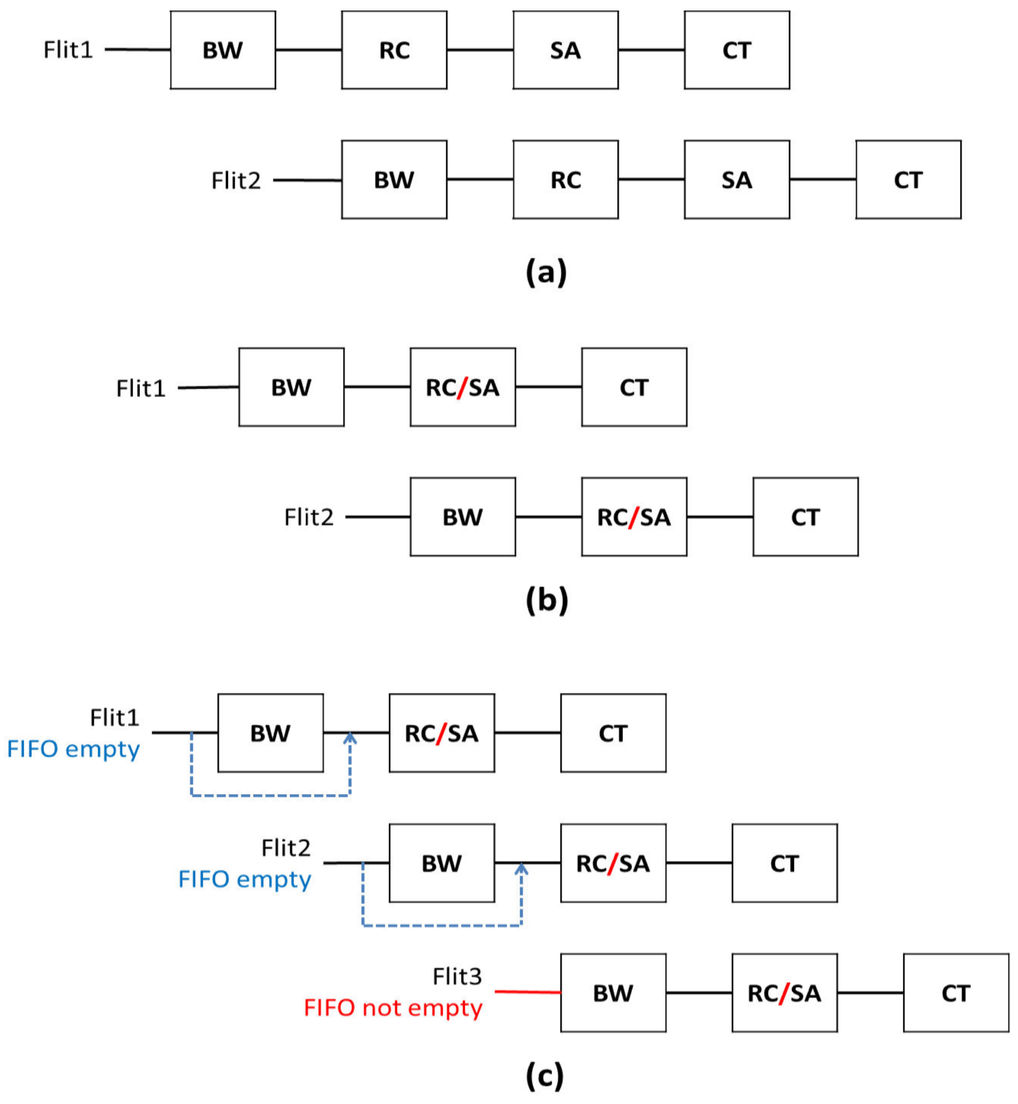


Fig 21: Router pipeline stages: (a) conventional XYZ (b) LA-XYZ (c) LA-XYZ with no-load bypass.

LA-XYZ computation goes under two steps: Assign next address and Define new Next-port as illustrated in Algorithm 1 and Algorithm 2 respectively. The first step decodes the Next-Port identifier from the incoming flit. Depending on the direction of this identifier, the address of the next downstream node can be computed. This address is then used in the second step by comparing it with the destination address of the flit which is also fetched from the flit's head. At the end of this process, information about the Next-Port is issued and embedded again in the flit to be used as a source of information for the Switch-Arbiter in the downstream node.

For further optimization, the no-load bypass technique can be also associated with LAXYZ. In case where the input FIFO buffer is empty, the flit does not have to be stored in the input buffer, but it continues its path straight to the RC/SA stage where the computation of both stages are still done in parallel. As a result, the pipeline stages can be further optimized by overlapping the BW stage, as it is illustrated in Fig.21 (c).

As we previously mentioned, LA-XYZ’s lack of support of fault tolerance outweighs the performance merits obtained when compared to other 3D routings.In addition, it suffers from a low worst-case throughput since the network congestion is not considered in the routing decision process.

Algorithm 1: LA-XYZ 1st phase: Assign next address

```
// Current node address
Input: $X_{cur}, Y_{cur}, Z_{cur}$
// Next port identifier
Input: Next-port
// Next node address
Output: $X_{next}, Y_{next}, Z_{next}$
// Evaluate Next node x_address
if (Next-port is EAST) then
    $X_{next} \leftarrow X_{cur} + 1$;
else
    if (Next-port is WEST) then
        $X_{next} \leftarrow X_{cur} - 1$;
    else $X_{next} \leftarrow X_{cur}$;
end
// Evaluate Next node y_address
if (Next-port is NORTH) then
    $Y_{next} \leftarrow Y_{cur} + 1$;
else
    if (Next-port is SOUTH) then
        $Y_{next} \leftarrow Y_{cur} - 1$;
    else $Y_{next} \leftarrow Y_{cur}$;
end
// Evaluate Next node z_address
if (Next-port is UP) then
    $Z_{next} \leftarrow Z_{cur} + 1$;
else
    if (Next-port is DOWN) then
        $Z_{next} \leftarrow Z_{cur} - 1$;
    else $Z_{next} \leftarrow Z_{cur}$;
end
```

Fig 22: First Phase of Look Ahead – XYZ Routing Algorithms

Algorithm 2: LA-XYZ 2nd phase: Define new Next-port

```
// Destination address
Input: $X_{dest}$, $Y_{dest}$, $Z_{dest}$
// Next node address
Input: $X_{next}$, $Y_{next}$, $Z_{next}$
// New next port for next node
Output: New-next-port
if ($X_{next}$ is equal to $X_{dest}$) then
    if ($Y_{next}$ is equal to $Y_{dest}$) then
        if ($Z_{next}$ is equal to $Z_{dest}$) then
            New-next-port ← LOCAL;
        else
            if ($Z_{next}$ is smaller than $Z_{dest}$) then
                New-next-port ← UP;
            else New-next-port ← DOWN;
        end
    else
        if ($Y_{next}$ is smaller than $Y_{dest}$) then
            New-next-port ← NORTH;
        else New-next-port ← SOUTH;
    end
else
    if ($X_{next}$ is smaller than $X_{dest}$) then
        New-next-port ← EAST;
    else New-next-port ← WEST;
end
```

Fig 23: Second Phase of Look Ahead – XYZ Routing Algorithms

3.5 Look Ahead Fault Tolerant Routing Algorithm

To keep the benefits of the look-ahead routing, the proposed Look Ahead Fault Tolerant (LAFT) performs the routing decision for the next node taking into consideration its link status and selects the best minimal path. Before starting to explain LAFT, there are two important assumptions that should be mentioned. First, the links connecting the PE to the local input and output ports are always non-faulty. Second, we assume that there exists at least one minimal path between a (source, destination) pair. These assumptions are natural and necessary to deliver any flit from source to destination.

Simple fault detection mechanism based on a single multiplexer in each input port that reads the incoming flit and verifies whether is corrupted or not. Depending on this verification, the fault-control module sends a single bit signal to the upstream node that can be either 0 or 1, for valid or faulty respectively, as shown in Fig. 25. Each router sends the collected information corresponding to its own fault status to each one of the 6 neighbouring nodes and also to the Network-Interface of the attached PE (see Fig. 25). This information is represented in a 6 bits signal representing the router link status in each direction (North, East, Up, South, West and Down).

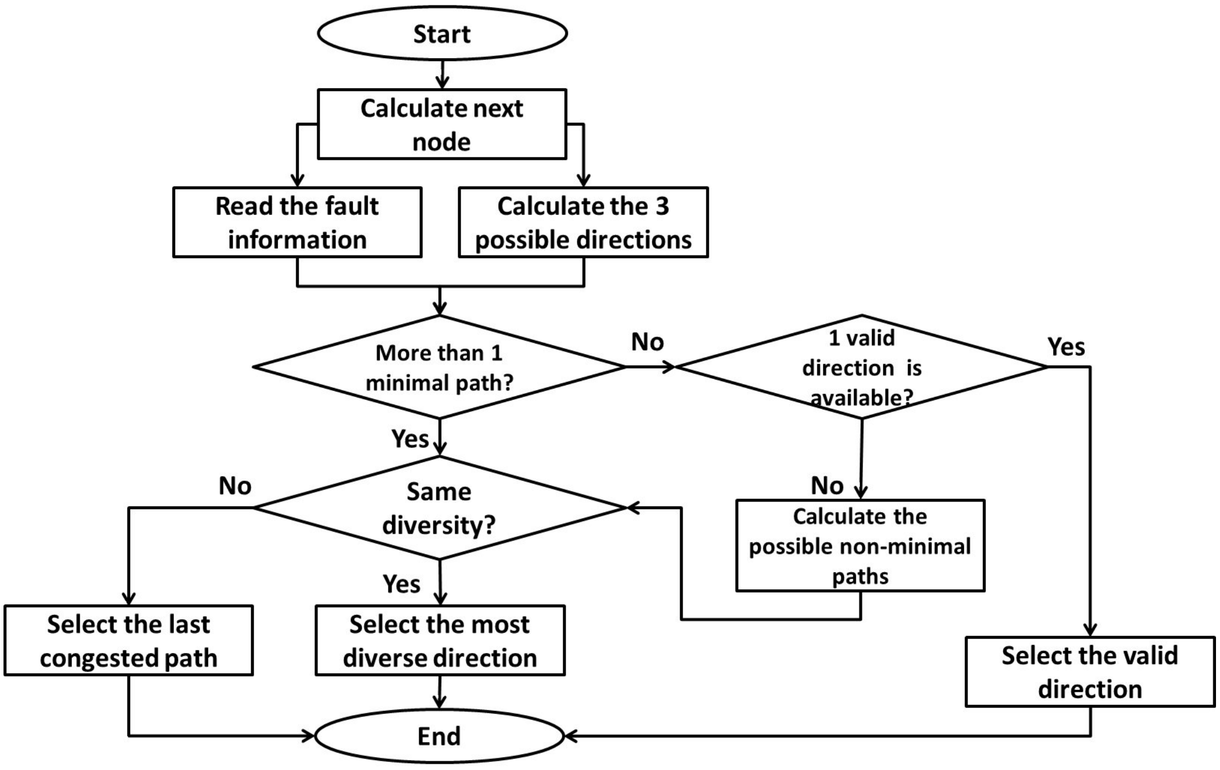

Start
Calculate next node
Read the fault information
Calculate the 3 possible directions
More than 1 minimal path?
No
1 valid direction is available?
Yes
Yes
No
Same diversity?
No
Calculate the possible non-minimal paths
Yes
Select the last congested path
Select the most diverse direction
Select the valid direction
End

Fig 24: Look Ahead Fault Tolerant routing algorithm flow chart.

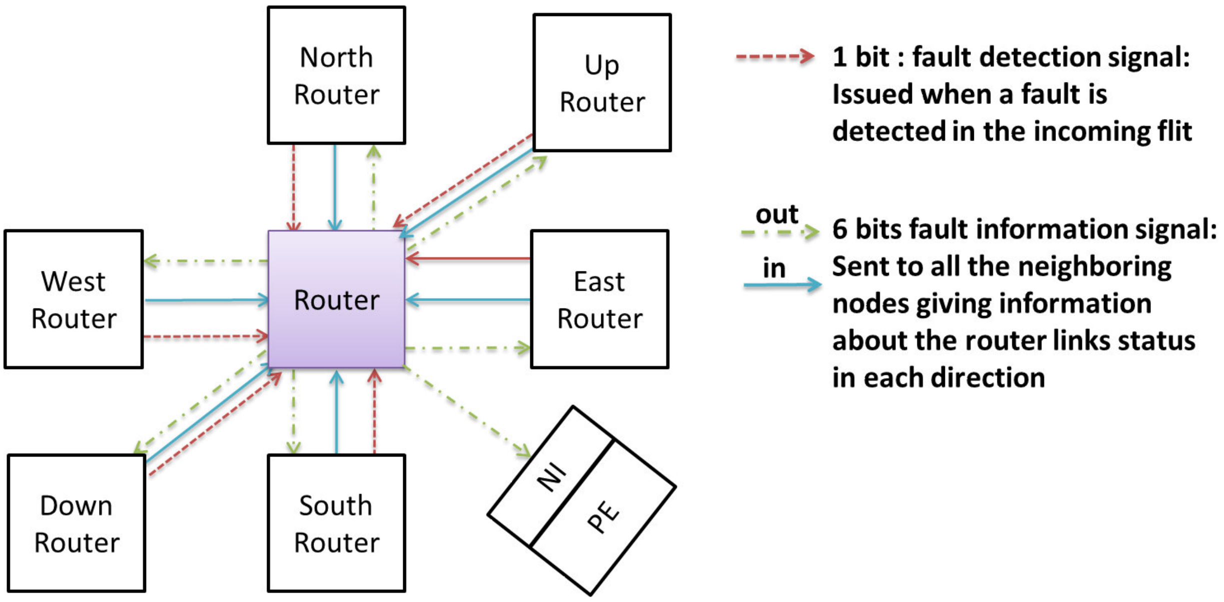

North Router
Up Router
West Router
Router
East Router
Down Router
South Router
NI
PE
1 bit : fault detection signal: Issued when a fault is detected in the incoming flit
out
in
6 bits fault information signal: Sent to all the neighboring nodes giving information about the router links status in each direction

Fig 25 : Fault information exchange.

It is important to mention that the choice of using control signals to transfer the faultinformation rather than using control flits is taken to enhance the performance. Using the latter approach will increase the congestion in the router where we may find data and control flits competing for the router resources. Also, we avoided using registers to store this information, and instead used signals to decrease as much area overhead as possible that might be caused by additional registers.

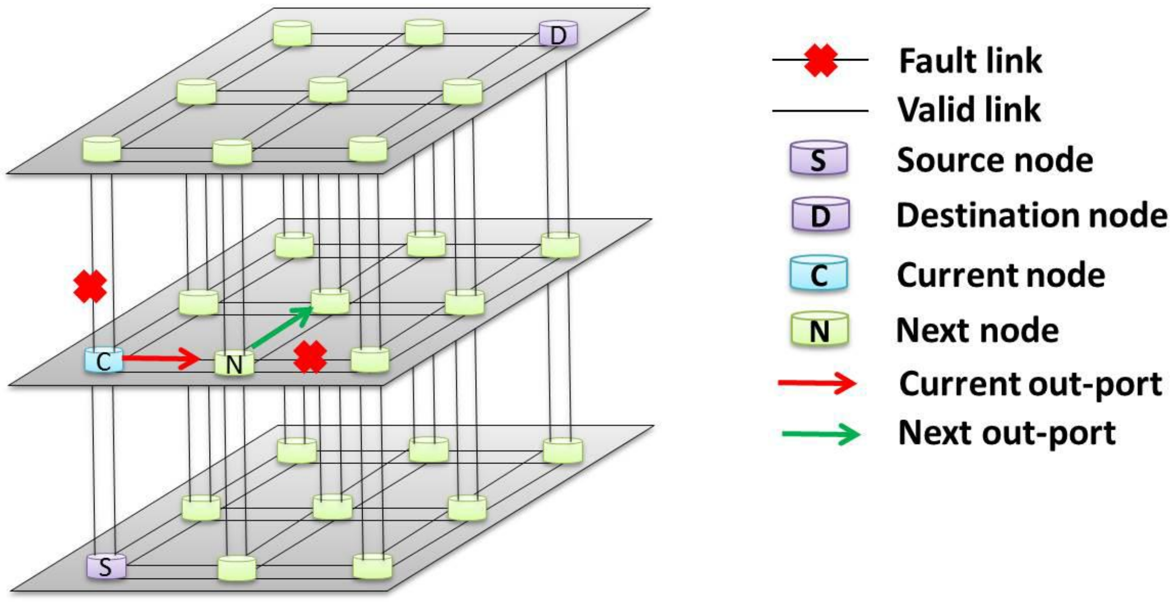


Fig 26 : Look Ahead Fault Tolerant routing algorithm example

The fault information is read by each input-port where LAFT is executed. Figure 24 illustrates the flow chart of this algorithm. The first phase of this algorithm calculates the next node address depending on the next-port identifier read from the flit. This phase is the same as in LA-XYZ previously illustrated in Algorithm 1. For a given node wishing to send a flit to a given destination, there exist at most 3 possible directions through X, Y, and Z dimensions respectively. In the second phase, LAFT performs the calculation of these 3 directions by comparing x, y and z coordinates of both current and destination nodes concurrently. At the same time, as these directions are being computed, the fault-control module reads the next-port identifier from the flit and sends the appropriate fault information to the corresponding input-port. By the end of this second phase, LAFT has information about the next node fault status and also the three possible directions for a minimal routing. In the next phase, the routing selection is performed. For this decision, a set of prioritized

Conditions are adopted to ensure fault tolerance and high performance either in the presence or absence of faults:

1. The selected direction should ensure a minimal path and it is given the highest priority in the routing selection.
2. We should select the direction with the largest next hope path diversity.
3. The congestion status is given the lowest priority.

Depending on these priorities, LAFT reads the fault status of the next node received from the fault-control module and checks the number of possible non-faulty minimal directions. As illustrated in Fig. 24, if only one non-faulty minimal direction is obtained, this direction will be selected as out-port for the next node. If more than one possible minimal direction is available, the algorithm selects the direction which leads to a node with higher path diversity. The diversity value for a given node is the number of possible directions leading to the destination through a minimal path. A node with high diversity results in more routing choices. This means that the probability of finding a non-faulty link is greater when considering faults. When no faults are detected in the system, selecting the direction with the highest diversity gives more choices to find the least congested direction. To obtain directions with high diversity, we should select those leading to nodes located in the centre of the mesh and avoid routing to the edges of the network.

When the three possible directions are minimal and have the same diversity, the routing selection is made depending on the congestion of each output port. This congestion information is obtained by the stop signal issued from the flow control used in our 3D-OASIS-NoC system.

When there is no valid minimal route available, LAFT chooses a non-minimal route while also considering the 2nd and 3rd priorities (path diversity and congestion) as illustrated in Fig. 3.

To understand better how LAFT works, we observe Fig. 26. Assuming that the current node (labeled C) received an incoming flit where the next port identifier, calculated in the previous node, indicates that the out-port for this flit is East (Red arrow). Applying Algorithm 1, the next node address is calculated (labeled N). Three minimal directions are possible for routing: East, North or Up. The East direction will not be selected since the link in this direction is faulty. Therefore, either North or Up can be selected, which both are minimal and non-faulty. In this case, the diversity priority is taken into consideration. If Up is selected, where the node in this direction is on one of the network edges, the diversity value is equal to 2 (2 minimal possible directions: (East or North). However if North is selected, its diversity value is equal to 3 (East, North or Up).

Having the highest priority, the North out-port (Green arrow) is selected for the next node and it is embedded in the flit to be used in the downstream node to allow the routing calculation and switch allocation to be performed both in parallel.

As is the case for every adaptive routing algorithm, the deadlock and live lock issues may rise. In our case, even with the absence of virtual channels, LAFT is able to avoid deadlock while making the routing minimal. In order to ensure this, we divided the rows of routers at each layer into 3 routing priorities: XYZ, YZX and ZXY. This division should be in a manner that each row [i,j] (where i and j are the layer and row identifiers respectively) has different priority from those of the neighbouring rows in each direction (row [i,j+1], row [i,j-1], row [i+1,j] and row [i-1,j]).

This priority is taken into consideration in two cases: The first one is when a non-minimal routing is selected. This selection should consider the priority of the row where the node is located.

For example, if the node is located in a row whose priority is YZX, the routing algorithm selects the direction through the Y dimension first if it is valid, and if it is not then it calculates the one in the Z dimension. The second case is when more than two minimal routes have the same diversity are computed. In this case also, the algorithm selects the direction depending on the priority of the row where the node is located. In this fashion, we eliminate the dependency between nodes by changing the routing dimension at each row if necessary and avoid routing through the same plane every time. This simple technique ensures deadlock freedom while keeping the route minimal as long as there exists one and without any considerable area overhead.

As long as the chosen route is minimal, the live lock problem does not exist either. However, it can be observed when a non-minimal direction is selected. For this reason, some restrictions are added when selecting the non-minimal route in addition to the one mentioned above. The first restriction forbids the flit to turn back to the same direction where it came from. The second one forbids selecting a path which is in the opposite direction of the faulty link (i.e. if ”East” is faulty then ”West” should not be selected). Adopting these restrictions guarantees the live lock freedom of LAFT, and the flits will continue to advance and search for a route until it finds a valid link.

LAFT is simple, minimal and ensures fault tolerance, and by using a low overhead solution, deadlock and livelock freedom can be guaranteed without any performance degradation. As it is shown later in the evaluation section, LAFT exhibits a good performance with a low area overhead.

CHAPTER 4

4.Simulation And Current Work

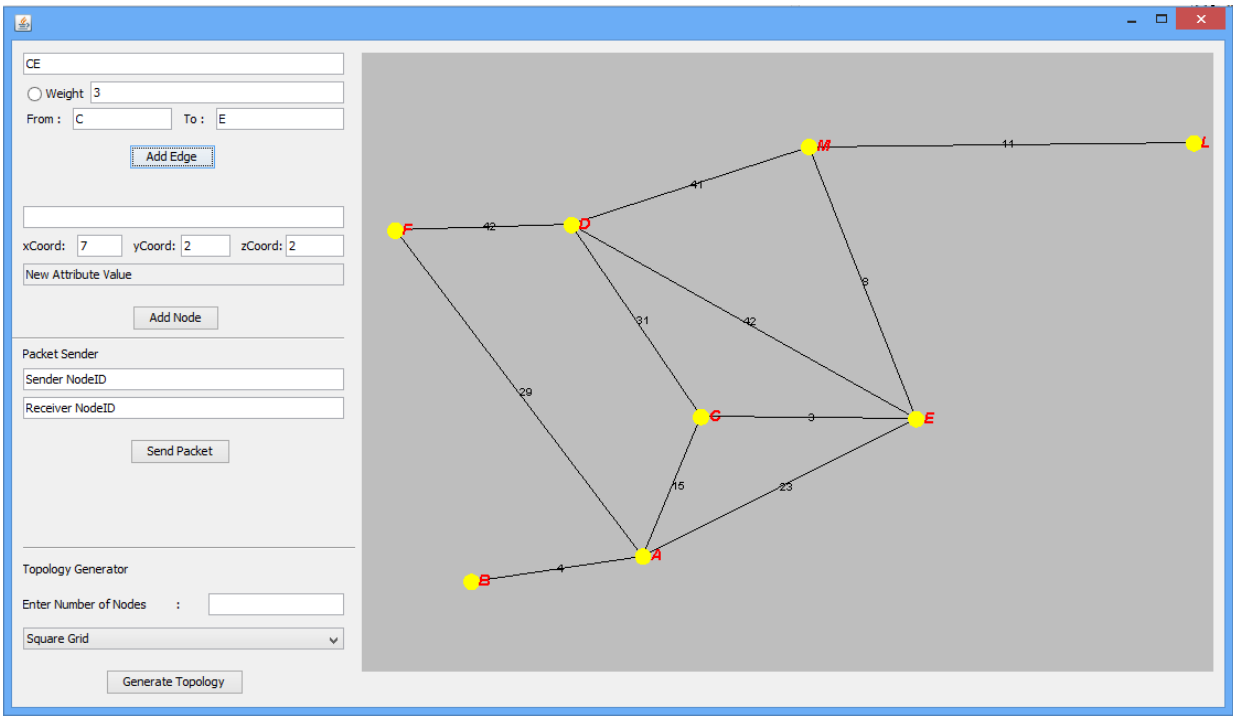
Following evaluation parameters are selected for performance evaluation of the topologies and routing algorithm

1)**Latency:** Latency is defined as the time taken by a packet to go through a communication path from its source to its intended sink.

2) **Drop probability:** Drop probability is calculated from the number of packets sent by the source and received by the sink.

3) **Network diameter:** An important parameter of any hierarchical network topology is network diameter which can be defined as the maximum internodes distance i.e. it is the maximum number of links that must be transversed to send a packet to any node along a shortest path. Time for sending a packet from one node to farthest away node in a network can be reduced as the network diameter is lower. Typically, to improve the performance and speed of network transmission, it needs to reduce the network diameter. To find total distance d of a network, one node is taken as source node and its distances to other nodes are calculated and finally number of nodes is multiplied by number of hops.

4) **Energy Dissipation** : Energy dissipation to transfer a single and multiple packets can be calculated
using (Eq.7 and 8) respectively


CE

Weight 3

From : C To : E

xCoord: 7 yCoord: 2 zCoord: 2

New Attribute Value

Packet Sender

Sender NodeID

Receiver NodeID

Topology Generator

Enter Number of Nodes :

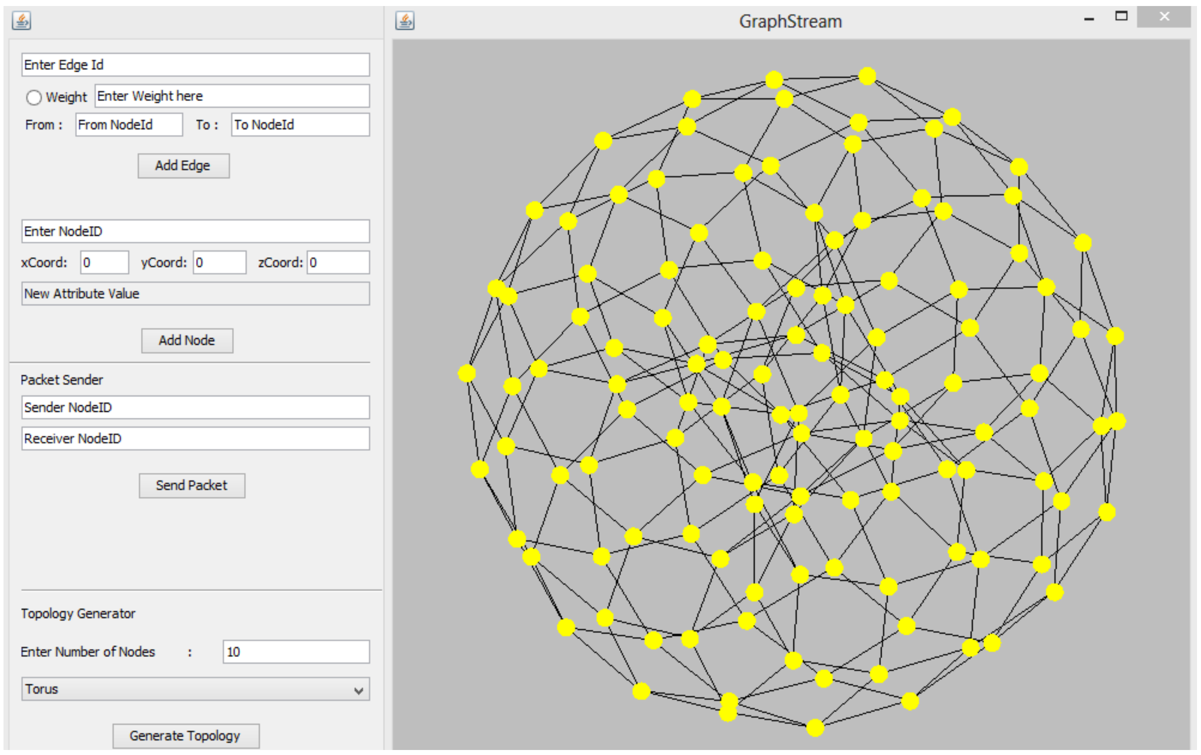
Square Grid

Fig 27 : Network generated using User Input

4.1 Torus Topology Generator :

A **torus interconnect** is a network topology for connecting processing nodes in a parallel computer system. It can be visualized as a mesh interconnect with nodes arranged in a rectilinear array of N = 2, 3, or more *dimensions*, with processors connected to their nearest neighbours, and corresponding processors on opposite edges of the array connected.The lattice has the topology of an N dimensional torus and each node has 2N connections.

A number of supercomputers on the TOP500 list use three-dimensional torus networks, e.g. IBM's Blue Gene/L and Blue Gene/P, and the CrayXT3. IBM's Blue Gene/Q uses a five-dimensional torus network. Fujitsu's K computer and the PRIMEHPC FX10 use a proprietary six-dimensional torus interconnect called Tofu.


Enter Edge Id

Weight Enter Weight here

From : From NodeId To : To NodeId

Add Edge

Enter NodeID

xCoord: 0 yCoord: 0 zCoord: 0

New Attribute Value

Add Node

Packet Sender

Sender NodeID

Receiver NodeID

Send Packet

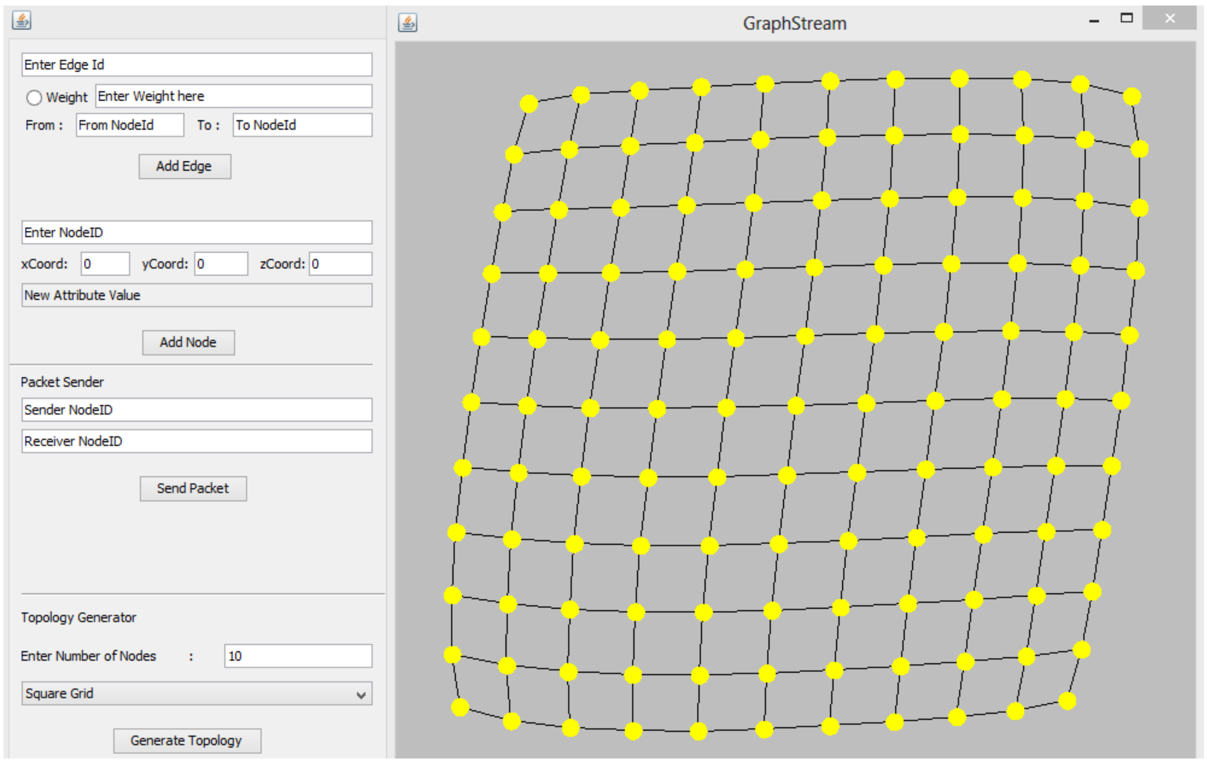
Topology Generator

Enter Number of Nodes : 10

Torus

Generate Topology

Fig 28: Torus Topology


Enter Edge Id

Weight Enter Weight here

From : From NodeId To : To NodeId

Add Edge

Enter NodeID

xCoord: 0 yCoord: 0 zCoord: 0

New Attribute Value

Add Node

Packet Sender

Sender NodeID

Receiver NodeID

Send Packet

Topology Generator

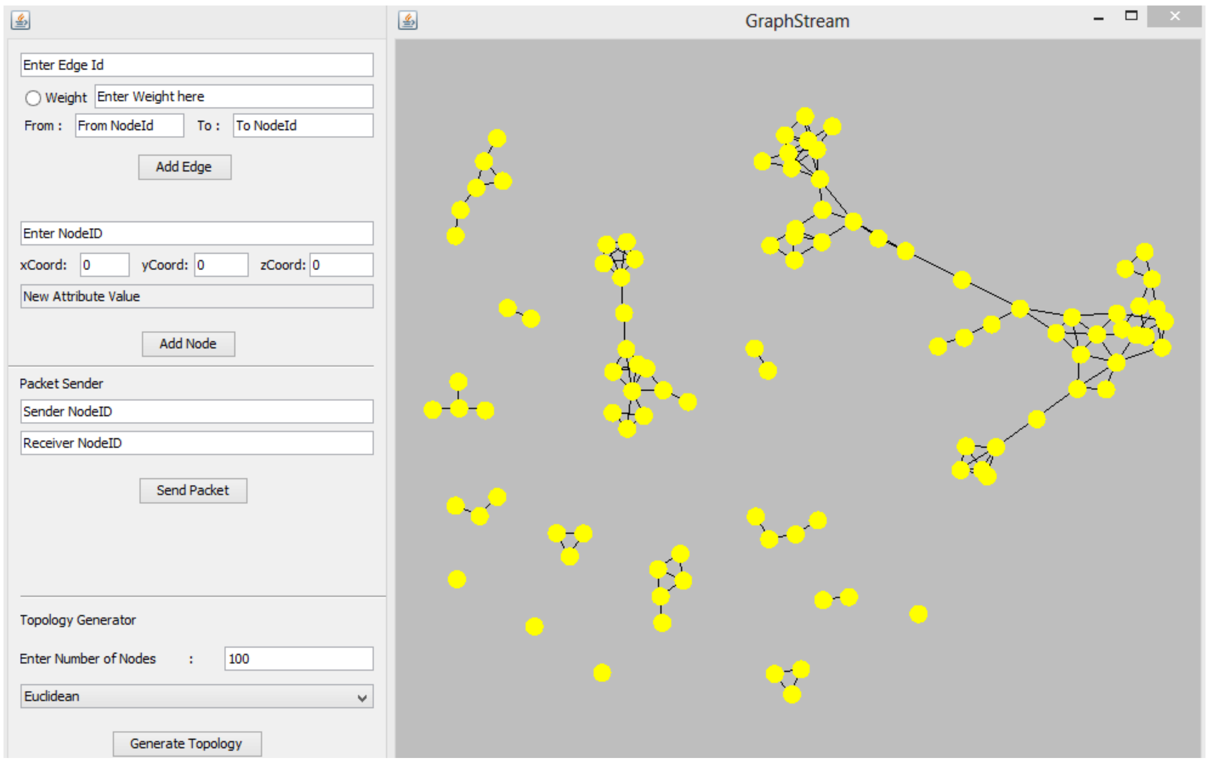
Enter Number of Nodes : 10

Square Grid

Generate Topology

Fig 29 : 2D Square Mesh

4.2 Euclidean Topology generator : This generator creates random graphs of any size. Links of such graphs are created according to a threshold. If the Euclidean distance between two nodes is less than a given threshold, then a link is created between those 2 nodes.



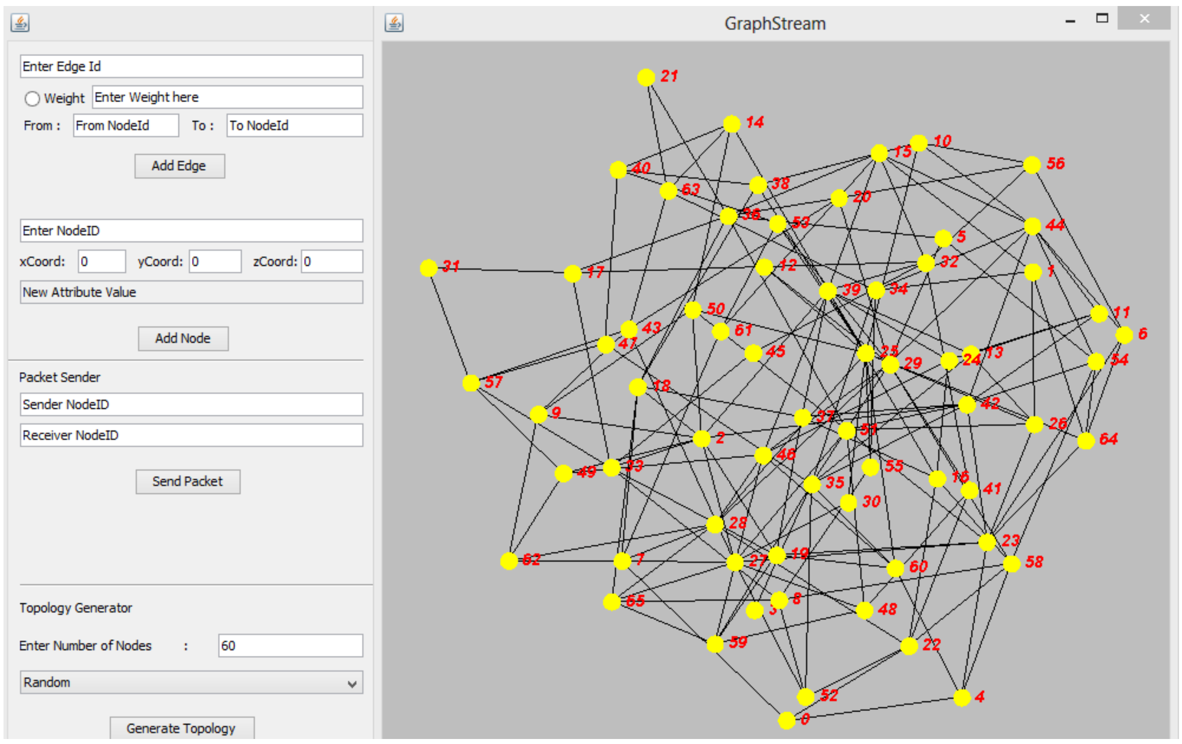
Enter Number of Nodes : 100

Euclidean

Fig 30 : Euclidean Network

4.3 Random Topology Generator : The generator tries to generate nodes with random connections, with each node having in average a given degree. The law in a Poisson law, however, the way this generator works, adding node after node, perturbs this process. We should first allocate all the needed nodes, then create edges. However, we create nodes at the same rate as edges. The more nodes are added the more the degree distribution curve is shifted toward the right.

This generator has the ability to add randomly chosen numerical values on arbitrary attributes on edges or nodes of the graph, and to randomly choose a direction for edges.



xCoord: 0 yCoord: 0 zCoord: 0

New Attribute Value

Enter Number of Nodes : 60

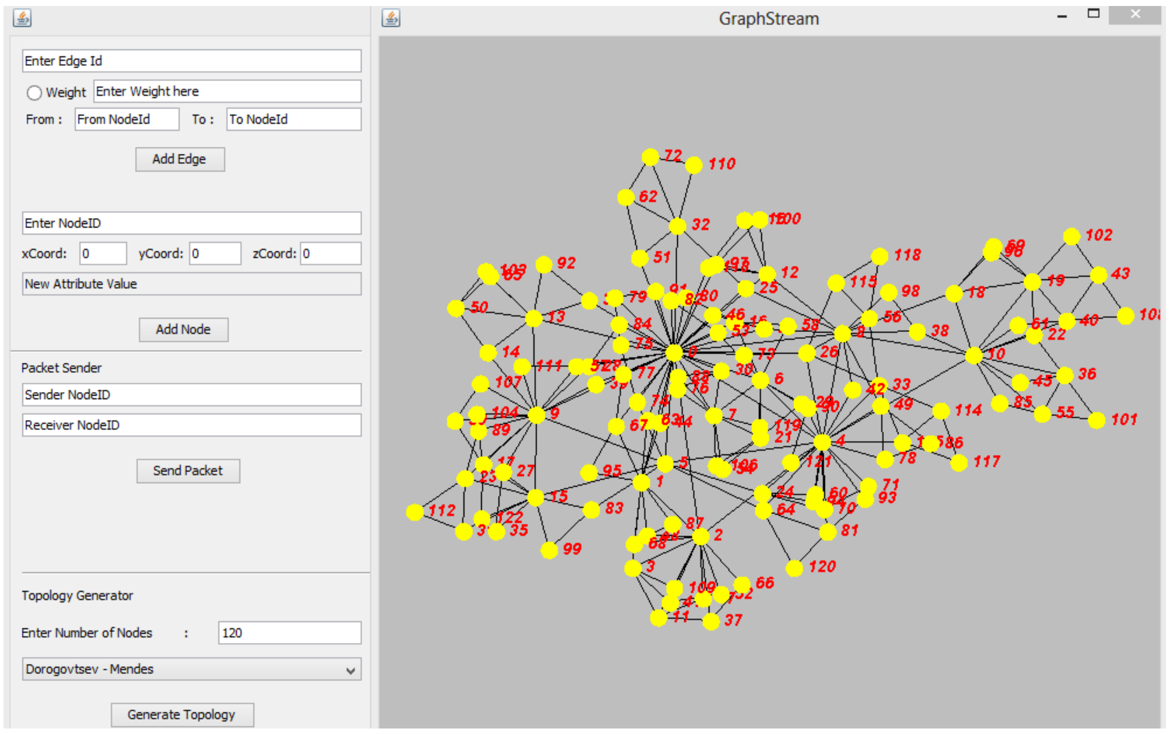
Random

Fig 31: Random Graph Generation With Average Degree of Node = 5

4.4 Dorogovtsev - Mendes Topology generator : This generator creates graph using the *Dorogovtsev and Mendes* algorithm. This starts by creating three nodes and tree edges, making a triangle, and then add one node at a time. Each time a node is added, an edge is chosen randomly and the node is connected via two new edges to the two extremities of the chosen edge.

This process generates a power-low degree distribution, as nodes that have more edges have more chances to be selected since their edges are more represented in the edge set.

The Dorogovtsev - Mendes algorithm always produce planar graphs.



Enter Edge Id
Weight: Enter Weight here
From : From NodeId To : To NodeId

Enter NodeID
xCoord: 0 yCoord: 0 zCoord: 0
New Attribute Value

Packet Sender
Sender NodeID
Receiver NodeID

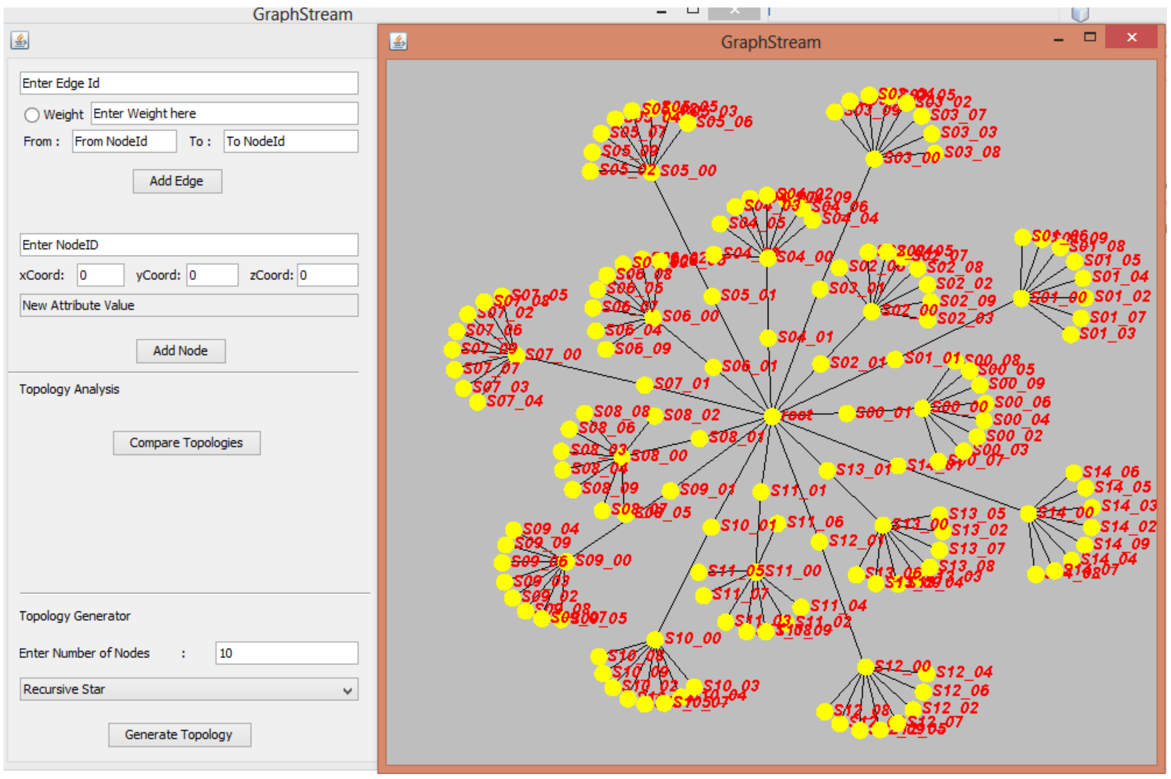
Topology Generator
Enter Number of Nodes : 120
Dorogovtsev - Mendes

Fig 32: Dorogovtsev – Mendes Network\

4.5 Recursive Star Topology : An (n, k) – recursive star network is a graph obtained by connecting one leaf of each of n copies of an k - star graph with a single root vertex that is distinct from all the stars.

Every vertex is connected to every other vertex by some path through root or through child nodes.

Network is 3-D because nodes have coordinates in all of the three dimensions x, y and z. 3D Star topology prevents the passing of data packets through an excessive number of nodes.


GraphStream

Enter Edge Id

Weight: Enter Weight here

From : From NodeId To : To NodeId

Add Edge

Enter NodeID

xCoord: 0 yCoord: 0 zCoord: 0

New Attribute Value

Add Node

Topology Analysis

Compare Topologies

Topology Generator

Enter Number of Nodes : 10

Recursive Star

Generate Topology

Fig 33 – Recursive Star Network

It has a central node which is called a root node.

4.6 Shortest Path Routing : the **shortest path problem** is the problem of finding a path between two vertices (or nodes) in a graph such that the sum of the weights (any parameter associated with links connecting vertices) of its constituent edges is minimized.

I have used Dijkstra’s algorithm (It is used for single source shortest path)if the weight of edges are all positive otherwise Bellman Ford algorithm is used (because for –ve edge weights the Dijkstra algorithm would not work).

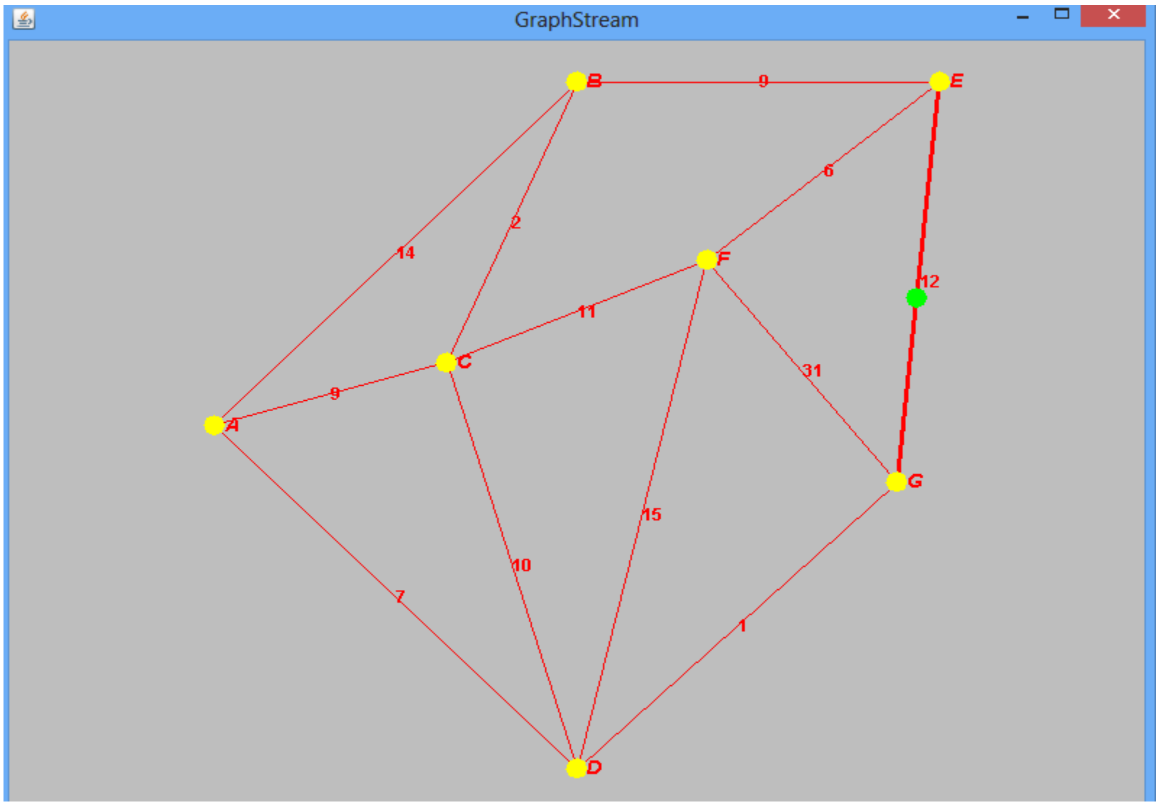
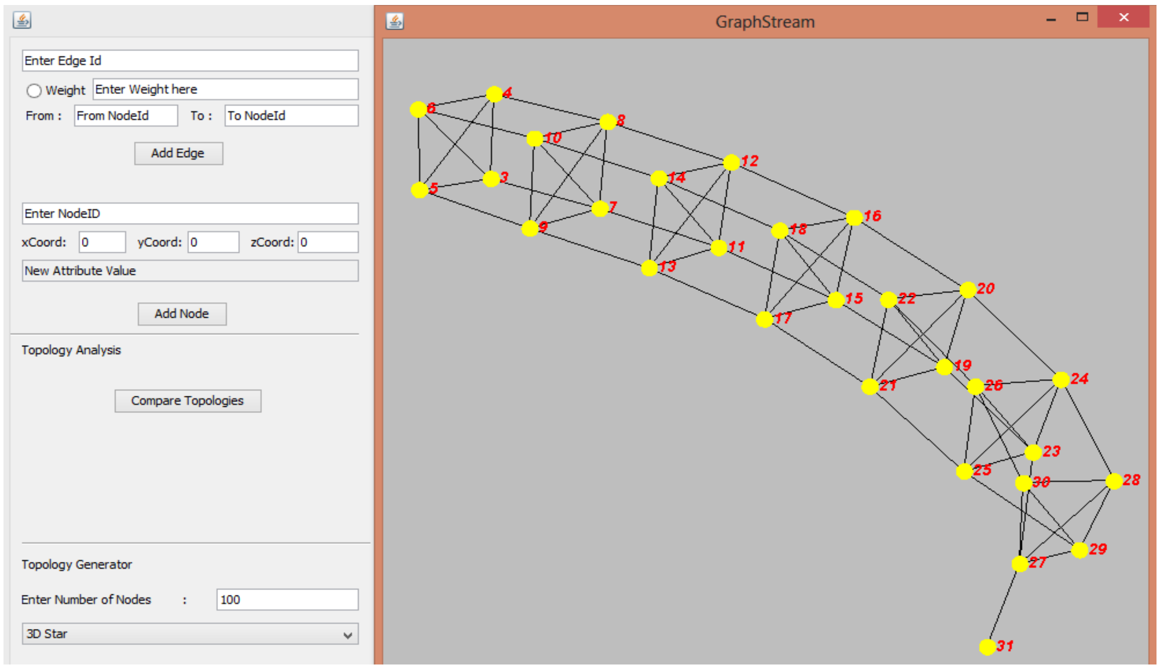


Fig 34 : Simulation of Shortest Path routing

4.7 3D Star Topology :



Enter Edge Id

Weight Enter Weight here

From : From NodeId To : To NodeId

Enter NodeID

xCoord: 0 yCoord: 0 zCoord: 0

New Attribute Value

Topology Analysis

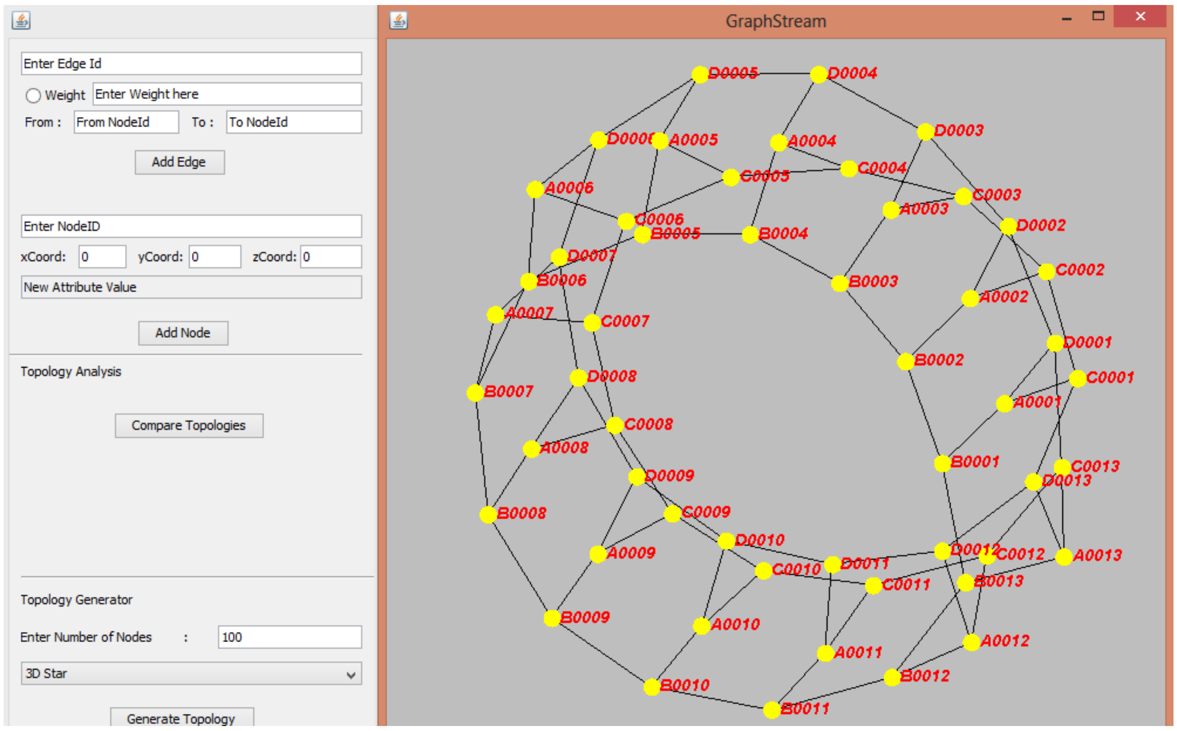
Topology Generator

Enter Number of Nodes : 100

3D Star

Fig 35 : 3D Star Network

4.8 3D Ring Topology : In 3d ring topology the coordinates of vertices lie in 3 dimensions x, y and z . The four group of vertices, only three out of them connected to the adjacent three form a 3D ring .


Enter Edge Id

Weight: Enter Weight here

From : From NodeId To : To NodeId

Add Edge

Enter NodeID

xCoord: 0 yCoord: 0 zCoord: 0

New Attribute Value

Add Node

Topology Analysis

Compare Topologies

Topology Generator

Enter Number of Nodes : 100

3D Star

Generate Topology

Fig 36 : 3D Ring Network

4.9 Comparison On the basic of Diameter of Graph

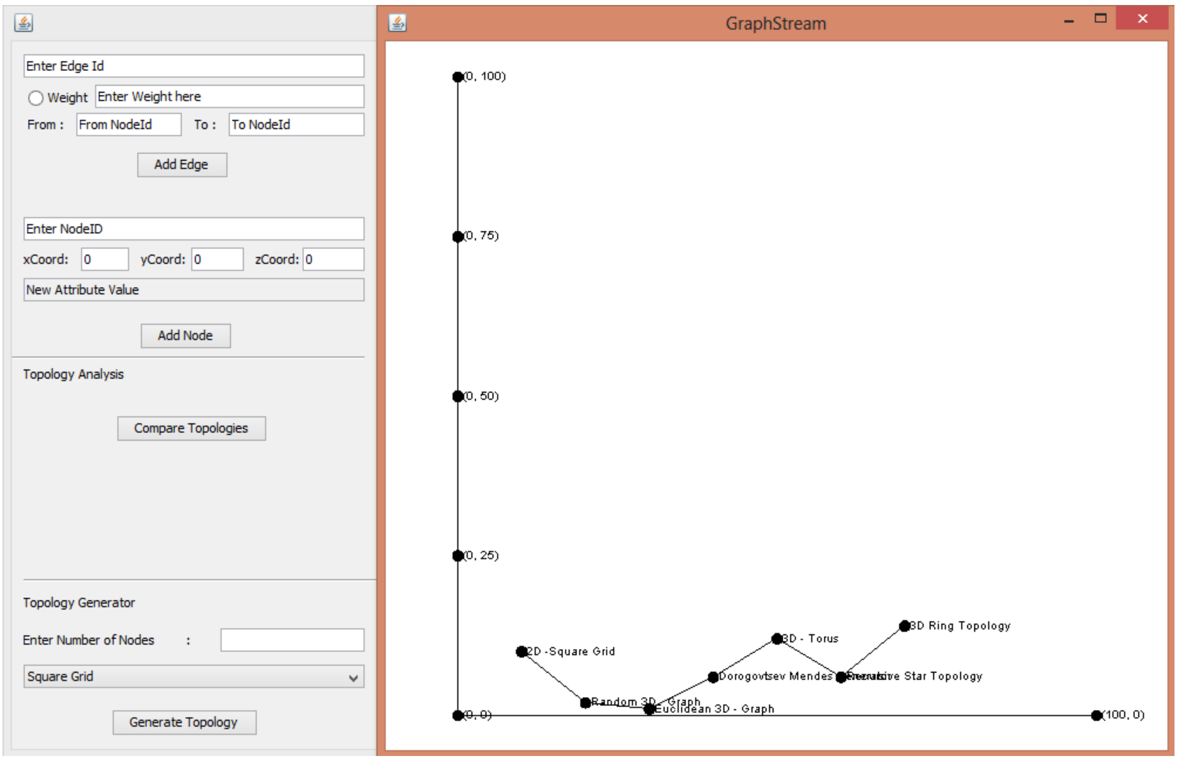


Fig 37 : Comparision of Topologies on the basic of diameter of Network

Diameter of graph is an important factor in deciding the quality or network performance , if the diameter of network is less that means nodes are mode connected or more closer to each other and if the diameter is more, then the network is sparse and connection between nodes is large i.e distance between two nodes large as compared to normal networks.

Conclusion And Future Work

Three-dimensional NoCs are natural extensions of 2D designs. In this paper, I have demonstrated that besides reducing the footprint in a fabricated design, 3D network structures provide a better performance compared to traditional, 2D NoC architectures. We have demonstrated that both mesh and tree-based NoCs are capable of achieving better performance when instantiated in a 3D IC environment compared to more traditional 2D implementations. The mesh-based architectures show significant performance gains in terms of throughput, latency, and energy dissipation with a small area overhead. On the other hand, the 3D tree-based NoCs achieve significant gain in energy dissipation and area overhead without any change in throughput and latency. However, if the NoC switches are designed to be as fast as the interconnect, even the 3D tree-based NoCs will exhibit performance benefits in terms of latency and bandwidth. The NoC paradigm continues to attract significant research attention in both academia and industry. With the advent of 3D ICs, the achievable performance benefits from NoC methodology will be more pronounced as shown in this paper. Consequently, this will facilitate adoption of the NoC model as a mainstream design solution for larger multicore system chips. we presented a novel low latency, high throughput and fault tolerant 3D-Networkon- Chip routing algorithm named Look Ahead Fault Tolerant routing algorithm (LAFT).

Future Work: Most of the topologies have been implemented (like, Torus 3D, 3D mesh, Random 3D topology, Planar Topologies like Dorogovtsev – Mendes and 3D start topology) and the left one are following :

1) Tree Base 3D topologies
2) 3D Recursive Network Topology (3D- RNT)

After Implementation of these technologies, One can implement all the left out routing algorithms, as shortest path routing has already been implemented following are algorithms will be implemented :

1) **K – Shortest path Routing**: The **K shortest path routing** algorithm is an extension algorithm of the shortest path routing algorithm in a given network.

   It is sometimes crucial to have more than one path between two nodes in a given network. In the event there are additional constraints, other paths different from the shortest path can be computed. To find the shortest path one can use shortest path algorithms such as Dijkstra's algorithm or Bellman Ford algorithm and extend them to find more than one path. The K Shortest path routing algorithm is a generalization of the shortest path problem. The algorithm not only finds the shortest path, but also K other paths in order of increasing cost. K is the number of shortest paths to find. The problem can be restricted to have the K shortest path without loops (loopless K shortest path) or with loop.

2. **Look Ahead – XYZ Routing** : already discussed earlier.

3) **Look Ahead - Fault Tolerant Routing** : already discussed earlier.

4) Routing Algorithm for 3D Start Network.

5) Routing Algorithm for 3D Mesh Network.

6) Routing Algorithm for 3D Recursive Network.

REFERENCES

[1] Y. Deng et al., “2.5D System Integration: A Design Driven System Implementation Schema,” Proc. Asia and South Pacific Design Automation Conf. (ASP-DAC), 2004.

[2] C. Grecu et al., “A Scalable Communication-Centric SoC Interconnect Architecture,” Proc. Fifth Int’l Symp. Quality Electronic Design (ISQED ’04), pp. 343-348, 2004.

[3] F. Li et al., “Design and Management of 3D Chip Multiprocessors Using Network-in-Memory,” Proc. 33rd Int’l Symp. Computer Architecture (ISCA ’06), pp. 130-141, June 2005.

[4] M. Ieong et al., “Three Dimensional CMOS Devices and Integrated Circuits,” Proc. IEEE Custom Integrated Circuits Conf. (CICC), 2003.

[5] P. Jacob et al., “Predicting the Performance of a 3D Processor- Memory Stack,” IEEE Design and Test of Computers, vol. 22, no. 6, pp. 540-547, Nov./Dec. 2005.

[6] C. Addo-Quaye, “Thermal-Aware Mapping and Placement for 3D NoC Designs,” Proc. IEEE Int’l System on Chip Conf. (SOCC ’05), pp. 25-28, 2005.

[7] C. Grecu, P.P. Pande, A. Ivanov, and R. Saleh, “Timing Analysis of Network on Chip Architectures for MP-SoC Platforms,” Microelectronics J., vol. 36, no. 9, pp. 833-845, Sept. 2005.

[8] W.R. Davis et al., “Demystifying 3D ICS: The Pros and Cons of Going Vertical,” IEEE Design and Test of Computers, vol. 22, no. 6, Nov./Dec. 2005.

[9] A.W. Topol et al., “Three-Dimensional Integrated Circuits,” IBM J. Research and Development, vol. 50, nos. 4/5, July-Sept. 2006.

[10] H.G. Lee et al., “On-Chip Communication Architecture Exploration: A Quantitative Evaluation of Point-to-Point, Bus, and Network-on-Chip Approaches,” ACM Trans. Design Automation
of Electronic Systems, vol. 12, no. 3, pp. 1-20, Aug. 2007.

[11] V.F. Pavlidis and E.G. Friedman, “3-D Topologies for Networkson- Chip,” IEEE Trans. Very Large Scale Integration (VLSI ’07), pp. 1081-1090, Oct. 2007.

[12] S. Vangal et al., “An 80-Tile 1.28TFLOPS Network-on-Chip in 65 nm CMOS,” Proc. IEEE Int’l Solid-State Circuits Conf. (ISSCC ’07), pp. 98-99, 2007.

[13] 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 Trans. Computers, vol. 54,
no. 8, pp. 1025-1040, Aug. 2005.

[14] Circuits Multi-Projects, http://cmp.imag.fr/, 2008.

[15] P. Chan, K. Dai, D. Wu, J. Rao and X Zou. The Parallel Algorithm Implementation ofMatrix Multiplication Based on ESCA. IEEE ASIA Pacific Conference on Circuits and Systems, pages 1091-1094, December 2010.

[16] Vitor de Paulo , CristinelAbabei, “3D Network-on-Chip Architectures Using Homogeneous Meshes and Heterogeneous Floorplans”, International Journal of Reconfigurable Computing, Hindawi, (2010)

[17] L. Torres, P. Benoit, G. Sassatelli, M. Robert, F. Clermidy and D. Puschini. An Introduction to Multi-Core System on Chip Trends and Challenges. Multiprocessor System-on-Chip: Hardware Design and Tool Integration, pages 1-21, January 2011

[18]A. DeOrio, D. Fick, V. Bertacco, D. Sylvester, D. Blaauw, J. Hu, and G. Chen. A Reliable Routing Architecture and Algorithm for NoCs. IEEE Trans. on CAD of Integrated Circuits and Systems, 31(5):726-739, May 2012.

[19] American Journal of Applied Sciences;2012, Vol. 9 Issue 3, p300