

## Design of 3D NOC Interconnection Router for Efficient Multicast Communication

<sup>1</sup>Kantha Bheemireddygar, <sup>2</sup>Sailaja Vemula

<sup>1</sup>M.Tech, Vishwa Bharati College of Engineering, India

<sup>2</sup>Asst. Professors, Vishwa Bharati College of Engineering, India

### ABSTRACT

In the NOC technology three dimensional integrated circuits offer greater device integration. In the network on chip technology the communication can be done by node routers. As well as the number of hops should be very less in order to reduce the signal delay and power. They also provide greater design flexibility by allowing heterogeneous integration. In order to exploit the intrinsic capability of reducing the wire length in 3D ICs, 3D NOC- Bus Hybrid mesh architecture was proposed. This architecture provides a seemingly significant platform to implement efficient multicast routings for 3D networks-on-chip. In this paper, we propose hybrid architecture using hamiltonian path based algorithm it finds shortest path between source nodes to the destination node. As well as implementing the DMA controller to improve the speed of the transaction. This is suitable for multicast communication in the networks.

**Keywords:** 3D NOC-Bus Hybrid Architecture; 3D ICs; Wormhole Routing; Multicast Communication; Hamiltonian Model.

### INTRODUCTION

As technology scales down in today's many-core chip multiprocessors (CMPs), providing efficient communication in a single die have become one of the critical factors for modern system designs. To get full benefits of parallel processing containing tens to hundreds of processors, a multiprocessor system needs efficient on-chip communication architecture [1]. One such emerging paradigm for communications with large-scale VLSI systems, which would meet the requirements of future cloud based applications, is network-on-chip (NoC). It is a general concept, proposed for complex on-chip communications and has better scalability, throughput and reduced power consumption [2]. However, increasing the number of cores over a 2D plane is not efficient enough due to long interconnects [3][4]. The 3D integration of such NOC's offers us immense benefits of parallel processing and efficient on-chip communication architecture. In order to overcome the problems associated with the interconnects and the limits posed by the traditional CMOS scaling, three-dimensional (3D) integrated circuits have been proposed [5].

Three-dimensional integrated circuit is emerging as attractive options for new on-chip interconnect design innovations. In 3D integration technologies, multiple layers of active devices are stacked above each other and vertically interconnected using Through-Silicon VIA's (TSVs) [6]. It permits a large degree of freedom in choosing an on-chip network topology. Thus a wide range of on-chip network structures that were not explored earlier are being considered. The 3D NoC is a cutting edge technology by integrating NoC systems in a 3D fashion [7]. It delivers better system performance

*\*Address for correspondence:*

Kantha.ece15@gmail.com

with significantly lower energy per packet, as compared to the 2D implementations due to increased package density and shorter wires [8].

The 3D Symmetric NOC architecture can be simply created by adding two additional physical ports to each router for interlayer communication [3]. Despite simplicity, this architecture does not exploit the beneficial feature of a negligible inter-wafer distance in 3D chips, because in this architecture, inter-layer and intra-layer hops are indistinguishable. Also, due to two extra ports, a considerably larger crossbar is required [9]. The 3D Stacked (Hybrid NoC-Bus) mesh architecture presented in [10] is hybrid architecture between the packet switched network and the bus architecture to overcome the above mentioned 3D Symmetric NoC challenges. It takes advantage of the short inter-layer distances that are characteristic of 3D ICs. In stacked mesh architecture, a six-port router is required instead of a seven port one for typical 3D NoC router, and vertical communication is just one hop away to any destination layer [11]. The single-hop and broadcast based interlayer communication offers a unique and hitherto previously unexplored way to implement an efficient multicast routing for 3D networks-on-chip.

The performance of networks-on-chip is greatly dependent on the efficiency of the message routing mechanism. Looking to the modern applications, supporting multicast traffic is becoming more common [12]. Multicast (one-to-many) is defined as delivering of the same message from a source node to an arbitrary number of destination nodes. Many cache-coherent shared memory systems utilize broadcast or multicast based protocols such as Intel’s QPI, AMD Opteron, Uncorq, Multicast Snooping and Destination Set Prediction protocols [13][14]. The multicasting communication can be implemented using existing unicast-based routing schemes; however unicast routing cannot support multicast traffic efficiently due to significant degradation of network performance and considerable increase in power consumption. In [13], *Jerger* et al. showed that for the uniform random traffic pattern if 1% of the total traffic is replaced with multicast traffic destined for a random number of destinations, the saturation point in the average packet latency curve will drop by about 38% being significant latency degradation. To cope with this issue, several recent research proposals and industry Trends lend themselves to support multicasting in hardware implementation of networks-on-chip.

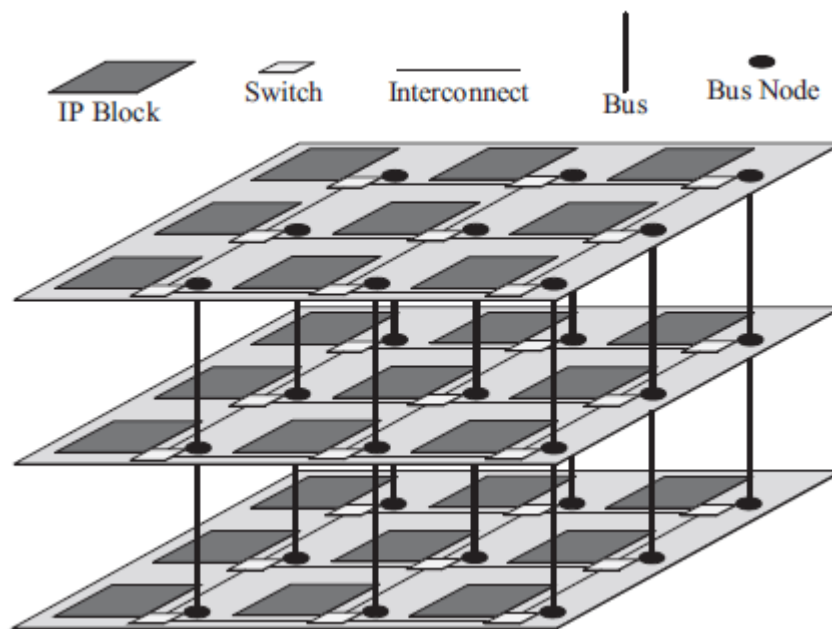
In this paper, we propose a novel multicast partitioning and routing strategy for the 3D NoC-Bus Hybrid mesh architectures to enhance the overall system performance and reduce the power consumption. The proposed architecture exploits the beneficial attribute of a single-hop (bus-based) interlayer communication of the 3D stacked mesh architecture to provide high-performance hardware multicast support. To this end, to the best of our knowledge for the first time, a customized partitioning method and routing algorithm for the 3D NOC Bus Hybrid mesh architecture are presented.

The remainder of the paper is organized as follows. Section II elaborates the demands for supporting hardware multicast in 3D NoC architectures and motivates this work. Section III presents the proposed implementation of multicast routing. The simulation methodology and results are shown in Section IV and finally, Section V concludes this paper.

## **RELATED WORK AND CONTRIBUTION**

The 3D Symmetric NoC is an extension to the 2D meshes based NoC architecture. For each NoC router of mesh topology, two extra ports is needed resulting a  $7 \times 7$  crossbar instead of  $5 \times 5$  crossbar for the 2D mesh architecture. Since, the crossbar power increases quadratically with number of ports, the power consumption for a 3D router is much higher than a 2D router [8]. The solution to the power

consumption for a 3D router has been proposed by Li *et al.* [10]. The proposed architecture is 3D NoC-Bus Hybrid mesh architecture. It takes advantage of the short inter-layer distances, around  $20\mu\text{m}$ , that are characteristics of 3D ICs [4] [15]. It integrates the multiple layers of 2D mesh networks by connecting them with a bus spanning the entire vertical distance of the chip as shown in Fig. 1. As the inter-layer distance for 3D ICs is small, the bus length will also be smaller. This makes the bus suitable for inter-layer communication in vertical direction. Due to one hop vertical communication and  $6\times 6$  routers, proposed architecture is efficient enough in terms of power consumption and latency. Since the bus is a shared medium it does not allow concurrent communication in the third dimension, however it has been shown that the dTDMA bus outperforms an NoC for the vertical communication as long as the number of 2D planes is less than 9 [10].

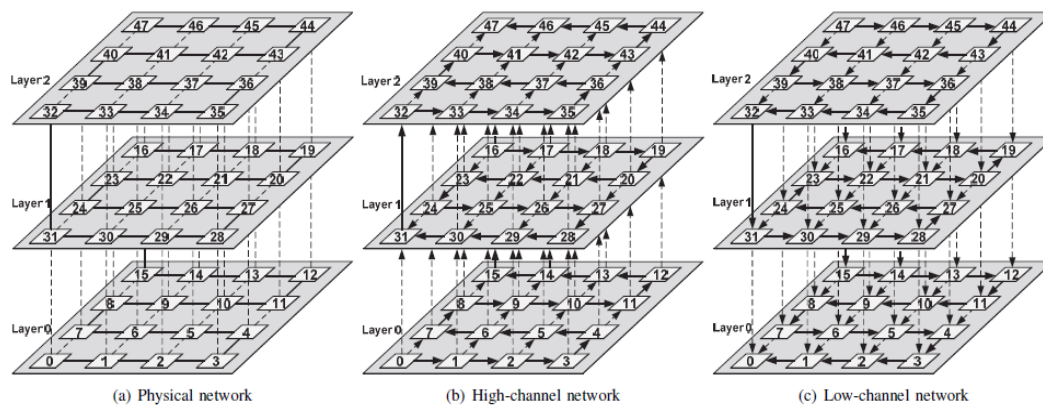


**Figure1.** 3D Hybrid NOC-Bus mesh architecture

A growing number of parallel and shared memory applications for CMPs necessitate the support for multicast communication. In multicast routing, the path selection procedure is crucial to maximize the efficiency of the message delivery process. Path selection refers to the process to determine an appropriate path to deliver a multicast message to all destination nodes. Various path selection techniques have been proposed in the literature. In general, they can be classified into four main categories: unicast-based, broadcast-based, tree based and path-based routing. In unicast-based routing identical messages are sent to the destination nodes recursively. As mentioned before, this technique suffers from performance inefficiency as well as excessive power consumption. In broadcast-based routing, which is used in BLBDR technique [16], multicasting is performed based on broadcasting in a small domain. The issues with this type of multicast routing is that if the destination nodes are located in the different part of the network and distributed not close to each other, the multicasting process will be complex. The reason is that in such cases, forming suitable broadcasting domains to cover all the destinations is a sophisticated process and may result to generating redundant traffic. The tree-based multicasting tries to construct a tree rooted at the source node in order to deliver a multicast message to destination nodes along the paths on the formed tree. Virtual Circuit Tree Multicasting (VCTM) [13] and Recursive Partitioning Multicast (RPM) [14] are two examples of the most recent tree-based multicast routing in NOCs. Tree-based routing can result in early network saturation when wormhole switching is used due to need for sending setup packet to

construct a tree as well as latency overhead for tree reconstruction (in the case of changing in the location of multicast source node) [17]. In addition, the area overhead to store tree information is not negligible. Path-based multicasting is proposed to cope with the early saturation problem associated with tree-based routing [18]. In path-based routing, an ordered sequence of destination addresses that must be delivered in the specific order is stored in the header of a packet. When a multicast message is received by an intermediate destination, the top address is removed from the header and it can be copied to the local memory. The message is routed to the next destination specified in the sorted list. The last destination node removes the packet from the network.

Node labeling is commonly used in path-based multicast routing. Hamiltonian path-based routing [19] is one the deadlock-free multicast routing in which node labeling is used. The label assignment algorithm for an  $m \times n \times r$  3D mesh has been explained in [20].



**Figure 2.** The Hamiltonian Labeling Of A  $4 \times 4 \times 3$  Mesh-Based Symmetric NOC Architecture (Bold Solid Lines: Hamiltonian Path, Equidistant Lines: Auxiliary Links)

In the Hamiltonian model, each node in the path is visited once. As shown in Fig. 2(a), the Hamiltonian path starts at the node labeled 0 and goes in ascending order to the last node with label  $N-1$ , which  $N$  is the number of nodes in the network. Based on the assigned label, partitioning methods are used to divide the network into two sub networks: high channel sub network and low channel sub network. The high channel sub network (Fig. 2(b)) contains all of the directional common channels with the nodes labeled from 0 to  $N-1$ . The low channel sub network (Fig. 2(c)) contains all of the directional common channels with the nodes labeled from  $N-1$  to 0.

In [21], Ebrahimi *et al.* present a set of partitioning method to enhance the efficiency of the Hamiltonian path based multicast routing in 3D Symmetric NoC architectures. In addition, they present a Minimal Adaptive Routing (MAR) algorithm for their proposed partitioning methods to further improve the performance characteristics of the network. However, as mentioned before, 3D Symmetric NoC architecture does not efficiently use the benefits of 3D ICs and suffers from large  $7 \times 7$  routers. In addition, in this architecture, the length of Hamiltonian path grows as the number of layers increases which results in considerable performance degradation. The multi-hop interlayer communication characteristic of this architecture has also a negative impact on the worst-case delay of the multicasting. The reason is that when a source node is located in or close to the top (bottom) layer of the NoC, the Hamiltonian-based partitioning is not efficient any more due to the unbalanced partitions. In this work, we cope with these issues by exploiting the benefits associated with the vertical buses used in interlayer communication of the 3D NoC-Bus Hybrid mesh architecture. Due to the single hop interlayer communication, the latency of our customized multicast routing is

independent of the layer in which a source node is located. As will be shown in the following section, this architecture bounds the length of the Hamiltonian path and consequently maintains the scalability of the Hamiltonian path based multicast routing.

### **PROPOSED HAMILTONIAN PATH BASED MULTICASTING STRATEGY**

As discussed before, the 3D NoC-Bus Hybrid architecture takes advantage of vertical buses offering single-hop and broadcast-based interlayer communication. These advantages necessitate an efficient and customized multicast labeling, partitioning and routing for this architecture. One of the main drawbacks of using Hamiltonian path for the 3D Symmetric NoC architecture is that depending on the number of layers and number of nodes per layer, the length of Hamiltonian path can dramatically increase. We address this issue in our proposed architecture in such a way that the Hamiltonian path is formed for each layer separately. As a result, the path length is bounded to number of nodes per layer. Fig. 3(a) illustrates an example of the proposed Hamiltonian labeling for a 4×4×3 mesh-based 3D NoC-Bus Hybrid architecture. As can be seen from the figure, the node labels are projected in all the layers identically and nodes with the same labels are differentiated using their layer IDs. Our labeling function  $L(\mu)$  for each node in an  $m \times n$  mesh-based layer can be expressed in terms of layer ID ( $i$ ) and x- and y- coordinates of nodes as follows:

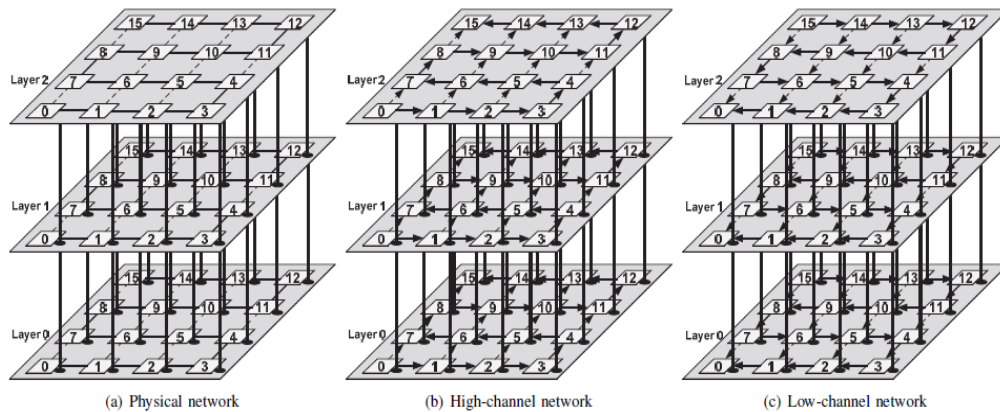
$$L(x, y)_i = \begin{cases} \{y \times n + x\}_i & \text{if } y \text{ is even} \\ \{(y + 1) \times n - (x + 1)\}_i & \text{if } y \text{ is odd} \end{cases} \quad (1)$$

Based on the hybrid architecture and the proposed labeling function, the high channel and the low channel sub networks are formed as shown in the Fig. 3(b) and Fig. 3(c), respectively. For this architecture, the high channel sub network for each layer contains all of the directional common channels with the nodes labeled from 0 to  $K-1$  and the low channel sub network contains all of the directional common channels with the nodes labeled from  $K-1$  to 0, where  $K$  is the number of nodes per layer. In our representation style, for instance,  $D_2(7)$  indicates the destination node labeled 7 at the layer 2. Similarly,  $7_2$  represent the node labeled 7 at the layer 2.

#### **Partitioning Algorithm**

The partitioning method plays a key role when implementing a multicast routing algorithm. The partitioning is the process of preparing a multicast message for being sent from a source node to a set of destination nodes. In Hamiltonian model, depending on the location of the source node, the partitioning method splits the network into subnetworks and prepares a well-structured multicast message for each partition. Partitioning policy can considerably reduce the network latency by dividing the network into balanced partitions as well as reducing the number of identical injected packets from a source node (startup latency). Our proposed partitioning algorithm for the 3D NoC-Bus Hybrid architecture is based on dividing the destination set ( $D$ ) into two disjoint groups ( $DU$  and  $DL$ ) depending upon the location of the source node ( $S$ ) in accordance with Algorithm 1.  $DU$  contains the destination nodes, which have a higher label than that of the source node. Similarly,  $DL$  consists of the destination nodes, which have a lower label than that of the source node. Then, the destination nodes in  $DU$  and  $DL$  are sorted in ascending and descending orders, respectively. Finally, two messages, one including the sorted  $DU$  and one including the sorted  $DL$  in their headers, will be sent from the source node to the destination nodes using the high channel and low channel sub networks, respectively. Since in our labeling method, the labels as such are not unique, if





**Figure3.** The proposed Hamiltonian labeling of a  $4 \times 4 \times 3$  mesh-based 3D NoC-Bus Hybrid architecture (horizontal solid lines: Hamiltonian path, vertical solid lines: interlayer buses, equidistant lines: auxiliary links)

Some destination nodes have the same labels, and then they are locally sorted based on their layer IDs. In the local sorting process, there is an exception that if the destination node is located in the same layer with the source node, it will come first. The reason is that a destination node with the same layer ID as a source node is visited before a multicast message is transmitted to the connected vertical bus. As will be shown in the following, our customized labeling and partitioning methods for the 3D NoC-Bus Hybrid architecture can help achieve significant performance improvements for multicasting over the conventional 3D NoC architectures.

Fig. 4 shows an example of multicasting using a same scenario for 3D Symmetric NoC architecture using conventional two-phase multicasting and 3D NoC-Bus Hybrid mesh architecture using proposed two-phase multicast partitioning strategy. In this example, it is assumed that for 3D Symmetric NoC architecture, the source node is at node 7 where the destination set is  $D=\{3,13,23,28,36,40,46\}$  and for 3D NoC-Bus Hybrid mesh architecture using our proposed multicasting method, the source node is at node 70 where the destination set is  $D=\{30, 31, 42, 81, 82, 130, 142\}$ . In addition, it is supposed that a minimal Hamiltonian path based static routing has been used [20]. For the conventional 3D NoC shown in the Fig. 4(a), two packets are formed for the high channel and low channel sub networks where  $DL=\{3\}$  and  $DU=\{13,23,28,36,40,46\}$ . The paths to cover all the destination nodes in the high channel and low channel sub networks are  $\{7,8,9,10,13,18,21,22,23,24,25,26,27,28,35,36,37,38,39,40,41,46\}$  and  $\{7,6,5,4,3\}$ , respectively. In this case, the maximum latency is equal to 22 hops. For the 3D NoC-Bus Hybrid mesh architecture using the proposed multicasting strategy, there is a dramatic improvement in terms of packet latency for the same scenario. As illustrated in Fig. 4(b), similar to the previous example, two packets are formed for the high channel and low channel sub network. In this case, the formation of the low channel and high channel sub network is based on our proposed partitioning method which leads to more efficient and balanced divisions where  $DL=\{4, 2, 30, 31\}$  and  $DU=\{81, 82, 130, 142\}$ . The paths to cover all the destination nodes in the high channel and low channel sub networks are  $\{70, 80\{81, 82\}, 90, 100, 130, 140\{142\}\}$  and  $\{60, 50, 40\{42\}, 30\{31\}\}$ , respectively. In this case, the maximum latency is equal to 7 hops (the intermediate bus transactions are not counted since they are performed in parallel with the packet traversal through the Hamiltonian path). The reason for such a significant improvement is threefold. First, due to the single-hop interlayer communication, the maximum latency is independent of the distance between a source node and a destination node located in the farthest layer. The second reason is that, the broadcast nature of the bus enables a multicast message to be delivered to the multiple destination nodes on the same bus at the same time. Finally, our

proposed novel labeling and partitioning method exploits the mentioned aspects of the bus to balance the sub networks.

**Algorithm 1** Two-Phase 3D Multicast Partitioning Algorithm

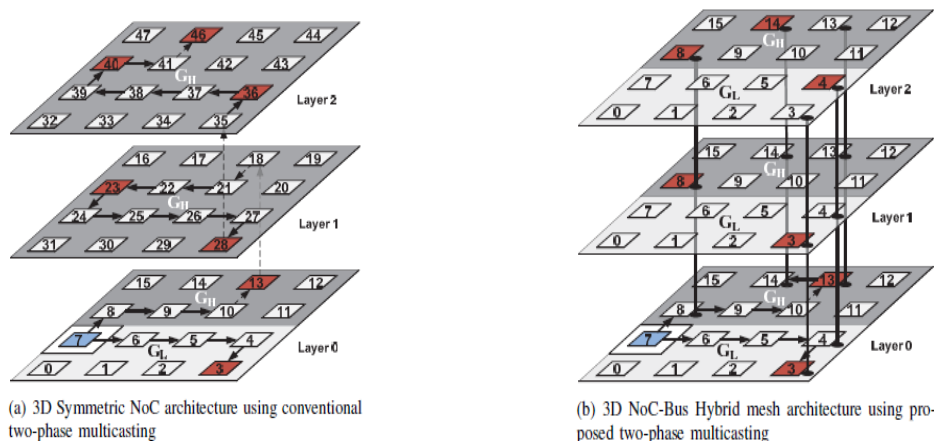
```

Input:  $(L, K, S, D) - \{L: \text{Number of layers, } K: \text{Number of nodes per layer, } S: \text{Source node, } D: \text{Destination set}\}$ 
Output:  $(D_L, D_U) - \{D_L \text{ and } D_U: \text{Destination set in low-channel and high-channel subnetwork, respectively}\}$ 

1: for  $i = 0 \rightarrow L - 1$  do
2:   for  $j = 0 \rightarrow K - 1$  do
3:     if  $\text{Label}(D_i(j)) < \text{Label}(S)$  then
4:       Add  $D_i(j)$  to  $D_U$ ;
5:     else
6:       Add  $D_i(j)$  to  $D_L$ ;
7:     end if
8:   end for
9: end for
10: Sort  $D_U$  in ascending order according to the label numbers. If some nodes have same label, sort them locally in ascending order using layer numbers (the node with the same layer ID as the source node comes first);
11: Sort  $D_L$  in descending order according to the label numbers. If some nodes have same label, sort them locally in descending order using layer numbers (the node with the same layer ID as the source node comes first);
12: Construct two messages for two disjoint subnetworks using  $D_U$ , and  $D_L$  in their headers;
    
```

**Packet Format**

In our multicast routing, once a destination node receives a multicast message it checks whether its address matches that of the first destination node in the message header. If so the address is removed from the message header, the message is copied and sent together with its header to the neighboring node in accordance with our proposed routing algorithm which will be shown in the following subsection. Since the 3D NoC-Bus Hybrid mesh architecture is a hybridization of two different communication media, it necessitates a new customized packet format to support multicasting. The packet format that is being used in our Hamiltonian path based multicasting strategy is shown in Fig. 5. It can be seen that the packet has a header flit and a number of payload flits. Each flit is  $p$  bits wide. The first bit in the flits is reserved for the *hop* (begin-of-packet) flag and the second bit for the *EOP* (end-of-packet) flag respectively. In the header flit,  $s$  bits are reserved for the routing information (RI). The remaining bits are allocated to implement some higher-level protocols (HLP). Although such a protocol is not defined in this paper, it is reserved for our derivative and future work. As shown in the Fig. 5, the RI field consists of four fields:



**Figure4.** Example of multicast partitioning and routing when the source node is at node 7 (GH: high channel sub network, GL: low channel sub network,)

Routing Mode (RM), Communication Mode (CM), Number of Destinations on the Bus (NDB), and Destination Address (es) (DA). The *RM* bit indicates whether the routing mode is unicast or multicast ('0'/'1'). In the case of multicast routing, the *CM* bit defines whether the communication mode is intra layer or interlayer ('0'/'1'). If the mode is intra layer, then it means that the current target destination node is located in the same layer (no interlayer communication is needed) and the first *DA* field (*DA0*) is just used. Otherwise, it describes that the current target destination node(s) are located in the layer(s) other than the current layer. In the case of interlayer multicast mode ( $RM=CM='1'$ ), there is a possibility that more than one destination node is connected to the same bus. The *NDB* field is used for such cases to indicate the number of destination nodes on the bus. In this format, the maximum number of supported destination nodes on a bus is  $2r$ . Based on our labeling and partitioning method, each *DA* field is divided into two subfields, *label* and *layer ID* (LID), to uniquely specify the position of each node.

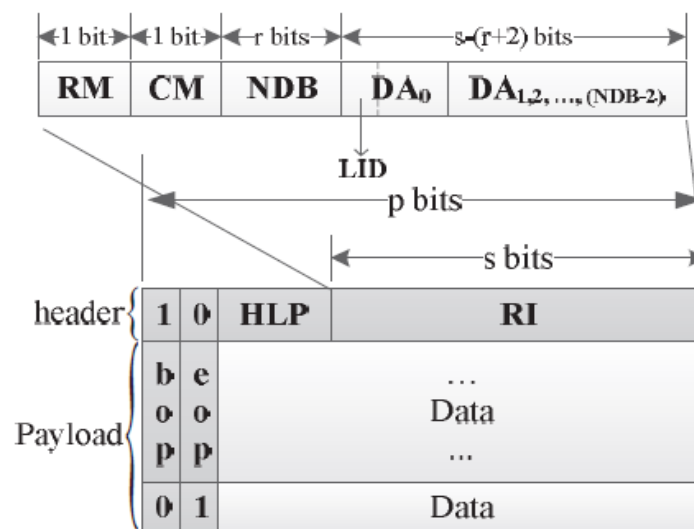


Figure 5. Packet format supporting our proposed multicasting mechanism

### Multicast Routing Algorithm

The multicast routing of packets that takes place in the 3D NoC-Bus Hybrid mesh architecture is based on our *Two-Phase 3D Multicast Routing Algorithm* routing algorithm, which has been elaborated in Algorithm 2. In this routing algorithm, the direction the data packet has to traverse, depends on the location of the current node (*Node C*), the location of the current target destination node (*Node D*), the location of the eastern neighboring node (*Node Eastern Neighbor*), the location of the western neighboring node (*Node Western Neighbor*), as well as the x- coordinate of the current (*X Node C*) and current target distinction (*X Node D*) nodes. The detailed description of the proposed multicast routing algorithm has been presented in Algorithm 2. It should be noted at this point that when a router sends a multicast packet to its *Up/Down* port, a bus architecture being able to support multicasting is needed to deliver the packet to the router(s) of connected destination node(s). For this purposes, we have utilized the bus architecture proposed in [22] which offers a multicast protocol for packet-based transactions. The proposed routing algorithm is a simple Hamiltonian path based static routing customized for the 3D NoC-Bus Hybrid mesh architecture. This algorithm is deadlock-free because it restrictively follows the Hamiltonian path model. If the deadlocks can be avoided then the proposed algorithm can be used to support adaptivity which will form part of our future work.



---

**Algorithm 2** *Two-Phase 3D Multicast Routing Algorithm*

---

**Input:**  $Node_C, Node_D, X_{Node_C}, X_{Node_D}, Node_{Eastern\_Neighbour}, Node_{Western\_Neighbour}$

**Output:** *Next Hop (East, West, North, South, Local, Up/Down)*

```

1: if Label(NodeC) = Label(NodeD) then
2:   if Layer_ID(NodeC) = Layer_ID(NodeD) then
3:     Deliver the packet to the Local node and exit;
4:   else
5:     Send the packet to the Up/Down output port (connected bus)
      towards the destination(s);
6:   end if
7: else if Label(NodeC) < Label(NodeD) then
8:   if XNodeC < XNodeD then
9:     if Label(NodeC) < Label(NodeEastern_Neighbour) then
10:      Send the packet to the East output port towards the destina-
      tion(s);
11:    else
12:      Send the packet to the North output port towards the destina-
      tion(s);
13:    end if
14:  else if XNodeC > XNodeD then
15:    if Label(NodeC) < Label(NodeWestern_Neighbour) then
16:      Send the packet to the West output port towards the destina-
      tion(s);
17:    else
18:      Send the packet to the North output port towards the destina-
      tion(s);
19:    end if
20:  else
21:    Send the packet to the North output port towards the destination(s);
22:  end if

23: else
24:   if XNodeC < XNodeD then
25:     if Label(NodeC) > Label(NodeEastern_Neighbour) then
26:      Send the packet to the East output port towards the destina-
      tion(s);
27:     else
28:      Send the packet to the South output port towards the destina-
      tion(s);
29:     end if
30:   else if XNodeC > XNodeD then
31:     if Label(NodeC) > Label(NodeWestern_Neighbour) then
32:      Send the packet to the West output port towards the destina-
      tion(s);
33:     else
34:      Send the packet to the South output port towards the destina-
      tion(s);
35:     end if
36:   else
37:     Send the packet to the South output port towards the destination(s);
38:   end if
39: end if

```

---

## DMA Controller

Direct memory access (DMA) is a feature of modern computers that allows certain hardware subsystems within the computer to access system memory independently of the central processing unit (CPU). Without DMA, when the CPU is using programmed input/output, it is typically fully occupied for the entire duration of the read or writes operation, and is thus unavailable to perform other work. With DMA, the CPU initiates the transfer, does other operations while the transfer is in progress, and receives an interrupt from the DMA controller when the operation is done. This feature is useful any time the CPU cannot keep up with the rate of data transfer, or where the CPU needs to perform useful work while waiting for a relatively slow I/O data transfer. Many hardware systems use DMA, including disk drive controllers, graphics cards, network cards and sound cards. DMA is also



[19] for preparing the destination set of each multicast message. We used a 48-node (three layers of 4×4 mesh) network and it was assumed the buffer size of each FIFO was eight flits, and the data width was set to 64 bits.

## **CONCLUSION AND FUTURE WORK**

In this paper, an efficient multicast partitioning and routing strategy for the 3D NoC-Bus Hybrid mesh architecture was proposed to support multicasting, thereby improving the overall NoC performance. The proposed architecture exploits the beneficial attribute of a single-hop (bus-based) interlayer communication of the 3D stacked mesh architecture to provide high-performance hardware multicast support. To this end, we proposed a customized labeling and partitioning method to efficiently split the network into well-balanced sub networks and enhance the multicast routing function. In addition, we presented a Hamiltonian path-based multicast routing algorithm, which exhibits a high degree of parallelism and reduces the startup latency by generating only two messages for created sub networks. Our extensive simulations with different traffic profiles showed that our architecture using the proposed multicast routing strategy can help achieve significant performance improvements over the recently proposed 3D NoC architectures offering hardware multicasting. In the future, our work will be extended by introducing a congestion-aware adaptive routing for the proposed architecture and simulating our network using a set of realistic workloads. We will also measure power consumption and area characteristics of both typical and proposed designs.

## **REFERENCES**

- [1] L. Benini and G. De Micheli. Networks on chips: a new SoC paradigm. *Computer*, 35(1):70–78, 2002.
- [2] A. Jantsch and H. Tenhunen. *Networks on Chip*. Kluwer Academic Publishers, 2003.
- [3] L. P. Carloni, P. Pande, and Y. Xie. Networks-on-chip in emerging interconnect paradigms: Advantages and challenges. In *Proceedings of the 2009 3rd ACM/IEEE International Symposium on Networks-on-Chip*, pages 93–102, 2009.
- [4] A.-M. Rahmani, K. Latif, K.R. Vaddina, P. Liljeberg, J. Plosila, and H. Tenhunen. Congestion Aware, Fault Tolerant, and Thermally Efficient Inter-Layer Communication Scheme for Hybrid NoC-Bus 3D Architectures. In *Proceedings of the 5th ACM/IEEE International Symposium on Networks-on-Chip*, pages 65–72, 2011.
- [5] A. W. Topol, D. C. La Tulipe, Jr., L. Shi, D. J. Frank, K. Bernstein, S. E. Steen, A. Kumar, G. U. Singco, A. M. Young, K. W. Guarini, and M. Jeong. Three-dimensional integrated circuits. *IBM Journal of Research and Development*, 50(4/5):491–506, 2006.
- [6] V. F. Pavlidis and E. G. Friedman. *Three-dimensional Integrated Circuit Design*. Morgan Kaufmann, 2008.
- [7] A.-M. Rahmani, K. Latif, P. Liljeberg, J. Plosila, and H. Tenhunen. Research and practices on 3D networks-on-chip architectures. In *Proceedings of the IEEE International NORCHIP Conference*, pages 1–6, 2010.
- [8] B.S. Feero and P.P. Pande. Networks-on-Chip in a Three-Dimensional Environment: A Performance Evaluation. *IEEE Transactions on Computers*, 58(1):32–45, 2009.
- [9] A.-M. Rahmani, K. Latif, K.R. Vaddina, P. Liljeberg, J. Plosila, and H. Tenhunen. ARB-NET: A novel adaptive monitoring platform for stacked mesh 3D NoC architectures. In *Proceedings of the 17th Asia and South Pacific Design Automation Conference*, pages 413–418, 2012.
- [10] L. Feihui, C. Nicopoulos, T. Richardson, Yuan X., V. Narayanan, and M. Kandemir. Design and

- Management of 3D Chip Multiprocessors Using Network-in-Memory. In Proceedings of the 33rd International Symposium on Computer Architecture, pages 130–141, 2006.
- [11] A.-M. Rahmani, K.R. Vaddina, K. Latif, P. Liljeberg, J. Plosila, and H. Tenhunen. Generic Monitoring and Management Infrastructure for 3D NoC-Bus Hybrid Architectures. In Proceedings of the 6th ACM/IEEE International Symposium on Networks-on-Chip, pages 177–184, 2012.
- [12] P. Abad, V. Puente, and J.-A. Gregorio. MRR: Enabling fully adaptive multicast routing for CMP interconnection networks. In Proceedings of the IEEE 15th International Symposium on High Performance Computer Architecture, pages 355–366, 2009.
- [13] N. E. Jerger, L.-S. Peh, and M. Lipasti. Virtual Circuit Tree Multicasting: A Case for On-Chip Hardware Multicast Support. In Proceedings of the 35th Annual International Symposium on Computer Architecture, pages 229–240, 2008.
- [14] L. Wang, Y. Jin, H. Kim, and E. J. Kim. Recursive partitioning multicast: A bandwidth-efficient routing for Networks-on-Chip. In Proceedings of the 3rd ACM/IEEE International Symposium on Networks-on-Chip, pages 64–73, 2009.
- [15] J. Kim, C. Nicopoulos, D. Park, R. Das, Y. Xie, V. Narayanan, M. S. Yousif, and C. R. Das. A novel dimensionally-decomposed router for on-chip communication in 3D architectures. In Proceedings of the ACM international symposium on Computer architecture, pages 138–149, 2007.
- [16] S. Rodrigo, J. Flich, J. Duato, and M. Hummel. Efficient unicast and multicast support for CMPs. In Proceedings of the 41st IEEE/ACM International Symposium on Microarchitecture, pages 364–375, 2008.
- [17] C. Yalamanchili, L. Ni, and J. Duato. Interconnection Networks: An Engineering Approach. Morgan Kaufmann, 2003.
- [18] X. Lin and L.M. Ni. Deadlock-free multicast wormhole routing in multicomputer networks. In Proceedings of the 18th annual international symposium on Computer architecture, pages 116–125, 1991.
- [19] X. Lin and L.M. Ni. Multicast communication in multicomputer networks. IEEE Transactions on Parallel and Distributed Systems, 4(10):1105–1117, 1993.
- [20] E.-O. Amnah and W.-L. Zuo. Hamiltonian Paths for Designing Deadlock-Free Multicasting Wormhole-Routing Algorithms in 3-D Meshes. Journal of Applied Science, 7(22):3410–3419, 2007.
- [21] M. Ebrahimi, M. Daneshtalab, P. Liljeberg, J. Plosila, and H. Tenhunen. Exploring partitioning methods for 3D Networks-on-Chip utilizing adaptive routing model. In Proceedings of the Fifth ACM/IEEE International Symposium on Networks-on-Chip, pages 73–80, 2011.
- [22] K. Latif, H. Tenhunen, and T. Seceleanu. MultiCast protocol for SegBus platform. In Proce pages 1–6, 2009.
- [23] A.-M. Rahmani, A. Afzali-Kusha, and M. Pedram. NED: A Novel Synthetic Traffic Pattern for Power/Performance Analysis of Network-on-Chips Using Negative Exponential Distribution. Journal of Low Power Electronics, 5(3):396–405, 2009.

### **AUTHORS' BIOGRAPHY**



**Kantha Bheemireddygari** has completed her B.Tech in Electronics and Communication Engineering from Maheshwara Institute of Technology , J.N.T.U.H affiliated college in 2012. She is pursuing her M.Tech in VLSI SYSTEM DESIGN from Vishwa Bharati College of Engineering, J.N.T.U.H affiliated college



**Sailaja Vemula** is an Assistant Professor at Vishwa Bharathi College of Engineering , Hyderabad. She received her Bachelor degree in Electronics and Communication Engineering from BVRIT Medak, Affiliated to J.N.T.U.H College, M.Tech, EMBEDDED SYSTEMS from RRS Institute of Technology, J.N.T.U.H affiliated college.