Journal of Computer and System Sciences 79 (2013) 475–491

Contents lists available at SciVerse ScienceDirect

Journal of Computer and System Sciences

www.elsevier.com/locate/jcss

Cluster-based topologies for 3D Networks-on-Chip using advanced inter-layer bus architecture

Masoumeh Ebrahimi, Masoud Daneshtalab *, Pasi Liljeberg, Juha Plosila, Hannu Tenhunen

Department of Information Technology, University of Turku, Finland

ARTICLE INFO

Article history:
Received 30 December 2010
Received in revised form 30 April 2011
Accepted 26 September 2012
Available online 9 October 2012

Keywords:
Three-dimensional Networks-on-Chip
Bus
Network topology
Inter-layer communication

ABSTRACT

Three-dimensional integrated circuits (3D ICs) have emerged as a viable candidate to achieve better performance and packaging density as compared to traditional two-dimensional (2D) ICs. In addition, combining the benefits of 3D ICs and Networks-on-Chip (NoCs) schemes provides a significant performance gain for 3D architectures. In recent years, through-silicon-via (TSV), employed for inter-layer connectivity (vertical channel), has attracted a lot of interest since it enables faster and more power efficient inter-layer communication across multiple stacked layers. The router-based and bus-based organizations are the two dominant architectures for utilizing TSVs as inter-layer communication channel in 3D architectures. Both approaches have some disadvantages. The former suffers from poor scalability and deteriorates the performance at high injection rates, and the latter consumes more area and power. The area overhead of TSVs reduces wafer utilization and yield, which can impact designing 3D architectures with a large number of TSVs. In this paper, two mesh-based topologies for 3D architectures are introduced to mitigate TSV footprint and power dissipation on each layer with a small performance penalty. On top of that, we propose a novel pipeline bus structure for inter-layer communication to improve the performance by reducing the delay and complexity of traditional bus arbitration.

1. Introduction

As technology geometries have shrunk to the deep submicron, the communication delay and power consumption of global interconnections of high performance Multi-Processor Systems-on-Chip (MPSoCs) are becoming a major bottleneck [1,2]. The Network-on-Chip (NoC) architecture paradigm, based on a modular packet-switched mechanism, can address many of the on-chip communication design issues such as performance limitations of long interconnects, and integration of a large number of intellectual property (IP) cores in a chip [3–5]. However, two-dimensional (2D) chip fabrication technology is facing several challenges in the deep submicron regime even when utilizing NoC architectures [6,7], e.g. designing the clock-tree network for a large chip, having limited floor-planning choices, increasing the wire delay and power consumption, integrating diversity components that are digital, analog, MEMS RF, etc.

The three-dimensional (3D) integration has emerged as a potent solution to address these problems and the design complexity of MPSoC in 2D Integrated Circuits (IC). 3D ICs reduce the interconnect delay by stacking vertically active silicon layers as well as offering a number of advantages over the traditional 2D chip [7–10]: (1) shorter global interconnects; (2) higher performance; (3) lower interconnect power consumption due to wire-length reduction; (4) higher packing density

* Corresponding author.
E-mail addresses: masebr@utu.fi (M. Ebrahimi), masdan@utu.fi (M. Daneshtalab), pakrli@utu.fi (P. Liljeberg), juplos@utu.fi (J. Plosila), hanten@utu.fi (H. Tenhunen).

0022-0000/$ – see front matter
http://dx.doi.org/10.1016/j.jcss.2012.09.005

and smaller footprint; and (5) support for the implementation of mixed-technology chips. In this paper we focused on wafer stacking technology. In wafer-to-wafer bonding technology, which is one of the popular options for 3D integrations, dies are vertically stacked. Short, fat, and vertical Through Silicon Vias (TSVs) are exploited for inter-layer communication. The distance between wafers can range from 5 μm to 50 μm [10,12], which is much shorter than the wire length between cores on a tier, and the pitches of TSVs can range from 1 μm to 10 μm square [10,12]. That is, the wire delay, power consumption and chip forming factor are significantly reduced [13,14,16].

3D NoC topologies not only enable scalable networks to provide communication requirements in 3D ICs [7–10], but also are a crucial factor of 3D chips in terms of performance, cost, and energy consumption [7]. Various on-chip network topologies have been studied for 3D NoCs [7–11,13,15,17]. Mesh-based structures are popularly used in 3D systems, because their grid-based regular architecture is intuitively considered to be matched to the 2D VLSI layout for each stack layer [7–10,13]. Nevertheless, if the number of IP-cores and memories increases in each layer, more TSVs are necessitated to handle the inter-layer communication. Inasmuch as each TSV employs a pad for bonding, the area footprint of TSVs in each layer is augmented significantly [10,17]. The main contributions of this work are twofold.

First, we propose two novel stacked mesh topologies to reduce the area overhead of TSVs and power dissipation with a small performance penalty. The proposed stacked mesh topologies, named Clustered Mesh Inter-layer Topology (CMIT) and Concentrated Inter-layer Topology (CIT), benefit of clustering the mesh topology for each layer. Each cluster of the presented topologies has its dedicated vertical channel, composed of a set of TSVs. CMIT and CIT preserve the advantages of the clustered mesh topology and mitigates both power density and TSV area footprint on each layer. However, since in both CMIT and CIT topologies, multiple IPs/routers in each layer can be connected to the same vertical channel, conventional bus architectures are not efficient for inter-layer connections due to the following reasons. Conventional buses require a centralized arbiter to control access requests from different routers to the bus necessitating many control wires between each layer and the central arbitration module. Also, they suffer from the limited parallel accesses to a vertical bus which can degrade the inter-layer communication performance.

The second contribution of the paper is to introduce a novel pipeline bus structure to solve the problems imposed by traditional bus architectures for inter-layer communication in 3D ICs. The proposed bus structure allows simultaneous transmissions over the bus without using centralized arbitration unit and overcomes the drawbacks of previously presented bus structures for vertical channels. This distributed arbitration mechanism improves the performance by reducing the delay and complexity of traditional bus arbitration while reducing the wiring overhead. Moreover, the presented bus architecture takes advantage of a fair forwarding policy along with a non-blocking scheme to share the inter-layer bandwidth among different layers efficiently.

Finally, the presented pipeline bus structure can utilize bi-synchronous FIFO for synchronization between stacked layers, if each layer is fabricated by different technologies.

The rest of the paper is organized as follows. In Section 2, the background is discussed and a brief review of related works is presented in Section 3. The proposed cluster architectures and the pipeline bus architecture are presented in Sections 4 and 5, respectively. The experimental results are discussed in Section 6 while the summary and conclusion are given in the last section.

2. Background

2.1. 3D IC technology overview

There are many technologies for die stacking being pursued by industry and academia. Wafer-Bonding [19,20] and Multi-Layer Buried Structures (MLBS) [21,22] are the most promising ones. The details of these processes are described in [6]. Wafer-to-wafer bonding appears to be the leading contender in industry and many recent academic studies have assumed this type of 3D stacking technology [6–11,23].

Wafers can be stacked either Face-to-Face (F2F) or Face-to-Back (F2B) and both have pros and cons. While the former provides the greatest layer-to-layer via density, it is suitable only for two-layers; and additional layers would have to employ back-to-back placement using larger and longer vias. On the other hand, Face-to-Back provides uniform scalability to an arbitrary number of layers, despite a reduced inter-layer via density [15–24]. Layers, stacked on top of each other, are connected via vertical interconnects tunneling through them. Wire bonding, micro-bump, contactless, and TSV are some of the vertical interconnect technologies that have been used in stacked structures [21]. The TSV interconnection has the potential to offer the greatest vertical interconnect density and therefore is the most promising one among these vertical interconnect technologies [15–24]. In this work, we assumed the F2B method with TSV interconnects to provide more scalability when more than two layers are employed.

2.2. 3D NoC architecture

3D-Symmetric NoC and 3D NoC-Bus Hybrid (stacked mesh) structures are popularly used in 3D systems, because their grid-based regular structure is intuitively considered to match the 2D VLSI layout for each layer [7–10,13]. The 3D-symmetric NoC structure, shown in Fig. 1(a), is an extension of 2D mesh by adding two additional physical ports to each baseline-router (one for up and one for down) in the popular 2D mesh-based system [7,10]. Adding two additional ports requires

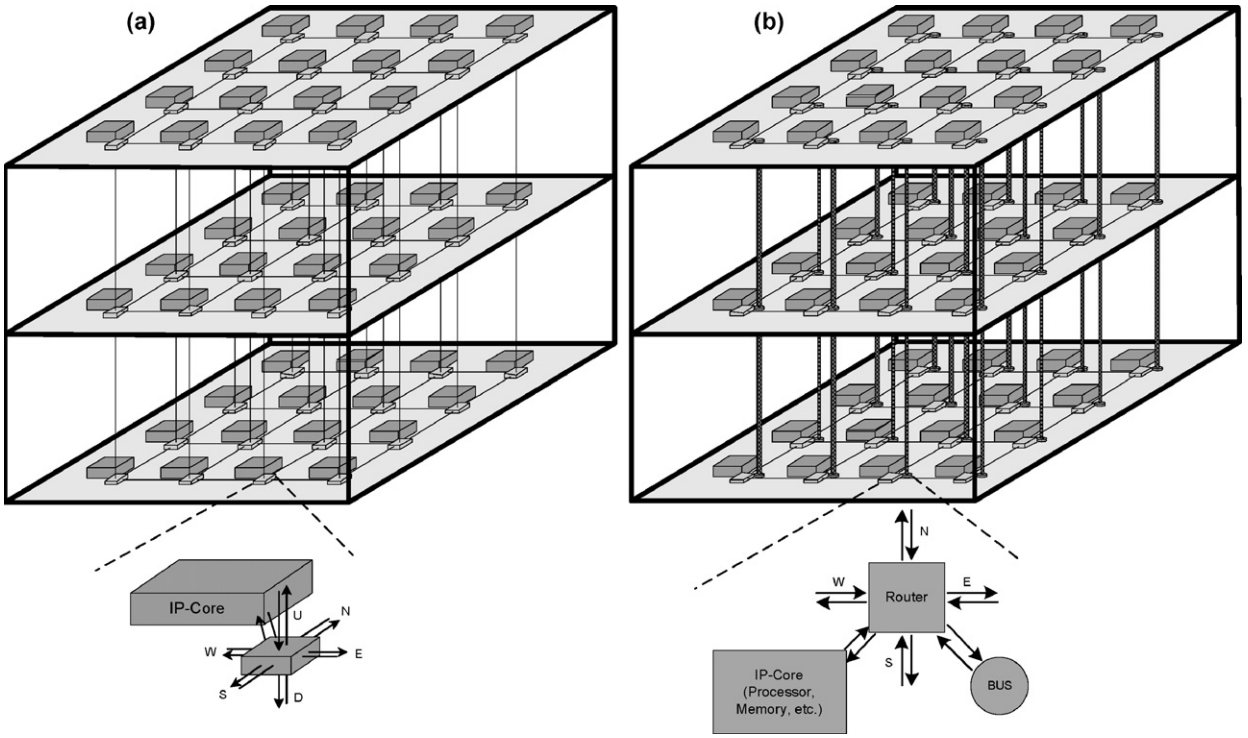

Fig. 1. Mesh-based NoC architectures: (a) 3D-symmetric NoC, (b) 3D NoC-Bus Hybrid structures.

larger crossbar incurring significant area and power overhead and increases the blocking probability occurring inside the router.

Since TSVs are shorter and wider than intra-layer interconnects, they have lower resistance and can support higher signaling speeds [10,17]. As router latencies may dominate the fast vertical interconnects, this has led the researchers to propose 3D NoC-Bus Hybrid structures using a bus with a centralized arbiter for each vertical channel, which allows single hop latency for packets between any layers [7–10]. As depicted in Fig. 1(b), on-chip routers in this structure have at most 6 ports, one to the IP-core, one to the bus, and four for cardinal directions. According to [10] the 3D-hybrid structure was observed to be better than the 3D-symmetric for the vertical interconnection as long as the number of device layers was less than 9. This has motivated us to present an efficient pipeline bus to overcome drawbacks of the conventional bus that has been employed for inter-layer communication.

2.3. Constraint on the number of TSVs

A relatively high area penalty due to via blockage may impose limitations on the number of TSVs that can be utilized for inter-layer communication due to the following reasons. First, the move from 2D to 3D architecture could accentuate the thermal concerns due to the increased power densities resulting from placing one logic block over another in the multilayered 3D stack. An efficient solution for cooling 3D ICs is to employ either thermal TSVs [17,18], establishing a thermal path from the core of a chip to the heat sink or liquid cooling based on fluidic TSVs [18]. This can take at least 10–20% of total chip area to create thermal efficient 3D ICs [17,18,25,26]. Second, conventionally in 3D ICs the input clock signal, at the center of the clock tree, is fed to each layer via TSV and each layer has its own clock tree with associated clock buffers implemented in the corresponding active layer. However, the obvious disadvantage of this scheme is the design overhead, both in terms of resources and design efforts required for the layer customization. Moreover, because of the separate customization of the different layers, the skew between terminals in different layers may be high even if the skew is low in the same layer. The alternative scheme, named via topology, implements the clock tree with the clock buffers on a single layer and using TSVs the clock signals from the terminals of the clock tree are passed to all other layers [27]. This scheme provides uniform skew compensation across layers since the same terminal clock signals are transmitted across layers with less design overhead, i.e. approximately N times less area and power than the conventional scheme, where N is the number of layers in the 3D clock tree. The only shortcoming of this scheme is the high number of TSVs required for passing the clock signals from terminals to all other layers. Third, if we consider signal TSVs and power/ground TSVs separately, since each signal TSV uses the minimal allowable TSV size and microprocessors typically require a few hundred I/O signals, signal TSVs occupy a very small area on stacked dies. On the other hand, microprocessors dies typically need tens or few hundreds of amperes current, which causes power consumption overhead for power TSVs due to the resistance

of TSVs. Thus, we have to increase the aggregate size of those power TSVs so that the area overhead induced through power TSV network is considerably high [28]. However, not only the area overhead of TSVs is quite high, but also floor-planning and routing is extremely challenging since TSVs are distributed in each layer. In this work, we present two area-efficient stacked architectures to reduce the TSV footprint with a small performance overhead though the presented bus architecture diminished the TSVs overhead in compared to conventional buses.

3. Related work

Design techniques and methodologies for 3D architectures have been investigated to efficiently exploit the benefits of 3D technologies. Several NoC topologies for 3D systems have been exhaustively investigated in [7–10,15,30,31]. The authors in [7] demonstrate that besides reducing the footprint in a fabricated design, 3D systems provide a better performance compared to traditional 2D systems. They have also demonstrated that both mesh and tree topologies for 3D systems achieve better performance compared to traditional 2D systems. However, the mesh topology shows significant performance gains in terms of throughput, average latency, and energy dissipation with a small area overhead [7]. In [30] different 3D mesh-based architectures have been compared in the zero-load latency, but the performance of the network with different traffic patterns and loads is also necessary to be evaluated.

To construct an optimistic 3D mesh-based system, several 3D structures have been presented. Baseline-routers in 2D mesh-based systems have 5 ports, i.e. 4 ports are connected to adjacent routers and one for the resource node. The straightforward extension for 3D mesh-based systems (3D-*symmetric NoC*) is to utilize routers with two additional inter-layer links by adding two physical ports to baseline-routers (one for up and one for down) [7,10,13,15,32]. As mentioned earlier, the 3D structure using such routers, not only increases the area and power overhead of the routers but also contention in the routers may arise. The electrical behavior of the relatively short and wide TSV, i.e. the low resistance, and supporting much higher signaling speeds led the authors of [10] to propose the 3D-hybrid structure. This 3D structure exploits the Dynamic Time Division Multiple Access (dTDMA) bus [33] with a centralized arbiter for the vertical communication link. Thus, moving from one layer to any of the other layers takes only one hop. However, contention issues in the bus limit the attainable performance gains [7]. That is, such structures inherently suffer from the limitation of buses since only one transmission is allowed each time over a vertical bus.

In [9], the DimDe router for 3D architectures has been proposed. The presented router uses a full 3D crossbar and a simple bus structure spanning all layers of the chip and fusing them into a single router entity. This router can minimize vertical traversal to one hop between any layers, but requires huge number of vertical connections and significantly complicates the control and arbiter of the router.

A multilayered 3D router architecture, named MIRA, is introduced for 3D systems by D. Park et al. [8]. The router components are classified as separable components (buffers, crossbar, and inter-router links) and non-separable components (arbiter and routing modules). The separable components are laid out across multiple layers to save chip area and reduce power by dynamically shutting down some inactive layers. However, such routers are too aggressive in the current technology [34].

To reduce the area footprint of TSVs, a serialization scheme for vertical channels has been presented in [17], but this scheme is only applicable with 3D-hybrid structures where each node has a dedicated vertical channel.

Due to the above concerns, in this paper, we have focused on both the 3D-symmetric structure (7-port switch design) and the 3D-hybrid structure (bus-based vertical interconnect). As described in [15], the 3D-hybrid structure is shown to perform the worst compared to the other structures in terms of scalability under local traffic. Although shown to be weak in [15,34], the bus may be appropriated for hot spot traffic injection where many packets may need to be sent through several layers to a hot spot frequently. This may be akin to a processor on one layer, and a memory stack directly above it. Hence, in 3D architectures, the 3D-hybrid structure performance degrades as the number of layers and number of processing nodes increase [15], thereby the 3D-symmetric structure is more feasible, mature, and more efficient than the 3D-hybrid structure as network size increases [35]. However, our proposed stacked architectures are applicable for both 3D-hybrid and 3D-symmetric structures where a group of nodes can share a vertical channel as an inter-layer interconnection. We also introduce a novel bus architecture which is more efficient than the conventional buses utilized for inter-layer communications in terms of scalability and performance.

4. Cluster-based topologies

As mentioned earlier, both 3D-symmetric and 3D-hybrid structures require a large number of TSV interconnections for inter-layer communication. In addition, each TSV requires a pad (around 5 $\mu \times 5$ μ) with the pitch of around 8 μ for bonding to a wafer, thereby, the area overhead of TSVs impose constraints on the number of TSVs [10,17,29]. In order to reduce vertical channels, we present two novel topologies, named CIT (Concentrated Inter-layer Topology) and CMIT (Clustered Mesh Inter-layer Topology). Although both of the presented topologies can be implemented as the 3D-symmetric (7-port router with vertical packet switched interconnection) and 3D-hybrid (vertical bus with an interface at each 6-port router) structures, we describe these topologies based on the 3D-hybrid scheme which is more efficient than the 3D-symmetric structure [10,17].

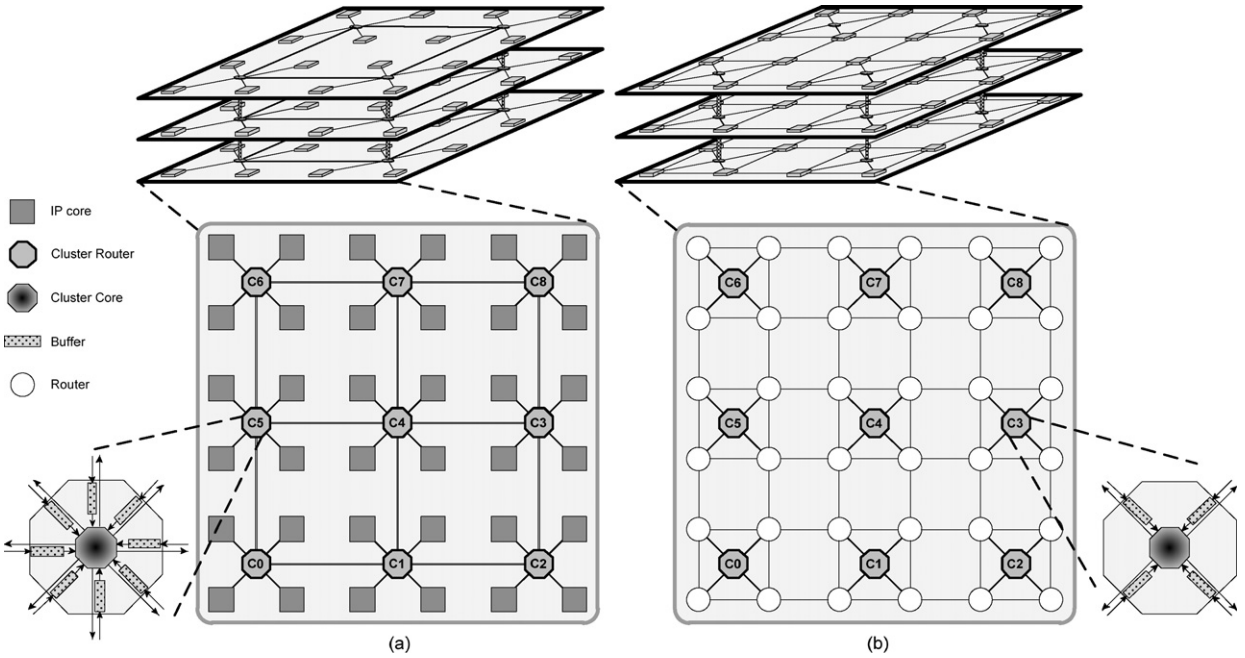

Fig. 2. Clustering approaches: (a) CIT and (b) CMIT.

4.1. CIT (Concentrated Inter-layer Topology)

Unlike the mesh topology where each IP-core is connected to a router, CIT forms a scalable architecture by sharing a router between multiple nodes (IP-cores and memories). CIT reduces the number of routers decreasing the number of vertical channels and hop counts. A 3×3 CIT with 36 nodes is shown in Fig. 2(a), where four nodes are grouped into a cluster, thereby forming 9 clusters in the network. Each cluster has a router with at most 9 ports (10 ports for 3D-symmetric structure): one connected to the bus, four to IP-cores/memories and the other four ports to neighbor routers. Communication channels in CIT can be classified as intra-layer channels (horizontal channels) and inter-layer channels (vertical channels). As illustrated in Fig. 2, the inter-layer communication is achieved by the cluster nodes. Each cluster node has a cluster core to establish the vertical connection via an interface to the vertical bus.

Due to the fact that each CIT router has larger number of input ports than the symmetric and hybrid routers, it consumes more area and power in comparison with conventional routers in the two other structures. In addition, the larger number of input ports becomes a performance bottleneck in terms of increased router complexity and contention probability inside the router (i.e. there are more input ports competing for an output port) [10,15]. Nonetheless, as the number of routers is decreased by the clustering approach in CIT, it not only reduces the area and power dissipation of the network but also the TSV area footprint is considerably diminished on each layer. Besides, the distance between two nodes of the same cluster in CIT is only one router so that the data transmission between nodes of the same cluster can be very fast. That is, the latency in CIT for distant nodes is more than that in the mesh-based 3D structures, but for nearby nodes the latency in CIT is smaller.

4.2. CMIT (Cluster Mesh Inter-layer Topology)

The structure of CMIT, depicted in Fig. 2(b), is basically similar to that of the mesh topology, except that for every layer the number of vertical channels has been reduced by sharing a vertical bus among routers of each cluster. That is, even preserving the advantage of the mesh on each layer, CMIT diminishes the number of inter-layer interconnection to meet constraints on the number of TSVs.

In CMIT, each router has at most 6 ports: one to the node (IP-core/memory), one to the bus (cluster router), and four for neighbors. Fig. 2(b) exhibits CMIT with 64 nodes, in which every four routers are grouped into a cluster on each layer. Even though CMIT achieves better area and power efficiency than the typical 3D mesh structure due to reducing the number of vertical channels, since several routers are connected to a shared vertical bus, the performance may be degraded when the inter-layer traffic is augmented. The specification of the three described architectures has been summarized in Table 1. The arbiter should be placed in the middle layer of the chip to keep wire lengths as uniform as possible. The number of control wires of each arbiter increases with the number of nodes attached to the vertical channel (bus). As a result, the presence of a centralized arbiter is the reason why the number of vertical channels in the chip should be kept low [10,17]. We believe that, the proposed topology can keep the number of vertical channels low with a negligible performance penalty.

Table 1
Number of routers, cluster routers and vertical channels of the described topologies in a $(4 \times 4 \times 4)$ 3D architecture.

Topology	# of routers	# of cluster routers	# of vertical channels
3D-hybrid	64	0	64
CIT	0	16	16
CMIT	64	16	16

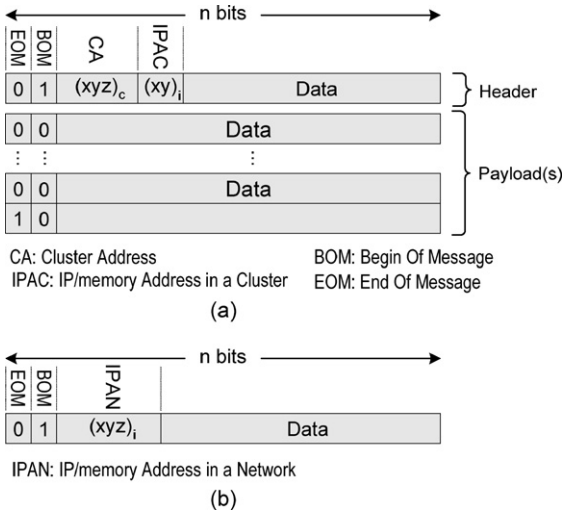

Fig. 3. (a) The packet format in CIT and (b) the header format in CMIT.

4.3. Routing algorithm

For the presented CIT, we employ the dimension order routing (DOR) algorithm which guarantees the network is deadlock free. DOR is a minimal deterministic routing scheme in which the message is first forwarded along the X-dimension, then along the Y-dimension, and, finally, along the Z-dimension. Fig. 3(a) shows the packet format of the CIT network. The header flit is n-bit wide and the nth bit is the EOM (End Of Message) sign and the $(n-1)$th bit is the BOM (Begin Of Message) sign. The third field indicates the address of the destination cluster and the next one is used for the IP-core/memory address inside the cluster. The content of the message is located in the rest of the flits (Payload). As can be seen from Fig. 3(b), the packet format of the CMIT network only has one destination address field (i.e. IPAN) similar to the mesh network. In CIT, the cluster routers perform the routing mechanism, while in CMIT the routing is performed by typical routers. That is, cluster routers are only employed for the purpose of inter-layer communication in CMIT.

5. Pipeline bus architecture

Traditionally, a bus is described as a shared link which can be owned by one attached subsystem at a time. Parallelism can be added to the structure by partitioning the bus into segments with bridges and allowing these segments to operate concurrently [36]. However, on one side, the overall system performance in such designs is still limited by the lack of parallel bus transactions, and on the other hand, it is not a suitable approach for vertical bus in 3D ICs because of requiring many control wires between each layer and the central arbitration module. This problem gets worse when the number of layers increases. Also, as the number of vertical channels has been reduced in both CIT and CMIT topologies, the performance can be considerably degraded when employing traditional bus structures for inter-layer communication. Our solution for these bottlenecks for vertical buses is to regard the system bus as a bidirectional pipeline which is capable of transferring data concurrently from one or more sources to several destinations. As the proposed architecture is illustrated in Fig. 4(a), the system is partitioned into a set of modules each of which is used to connect the corresponding layer to the pipeline bus. As the system is based on GALS design paradigm, the layers can internally operate at different clock frequencies. The layers are independent of each other, in case there are some inter-layer transactions, the layers exchange data synchronously or asynchronously through the pipelined system bus, a segmented communication link which allows simultaneous transfer in both directions. The layers can concurrently access the bus without waiting for any grant signals, because of the pipeline structure of the proposed bus architecture. The interface module acts as a synchronizer between the router and the pipeline bus. To construct the pipeline bus, the physical wires that implement the bus are divided into a set of segments separated from each other by Transfer Stages (TS), one attached to each layer (Fig. 4(a)). Each transfer stage contains internal FIFO

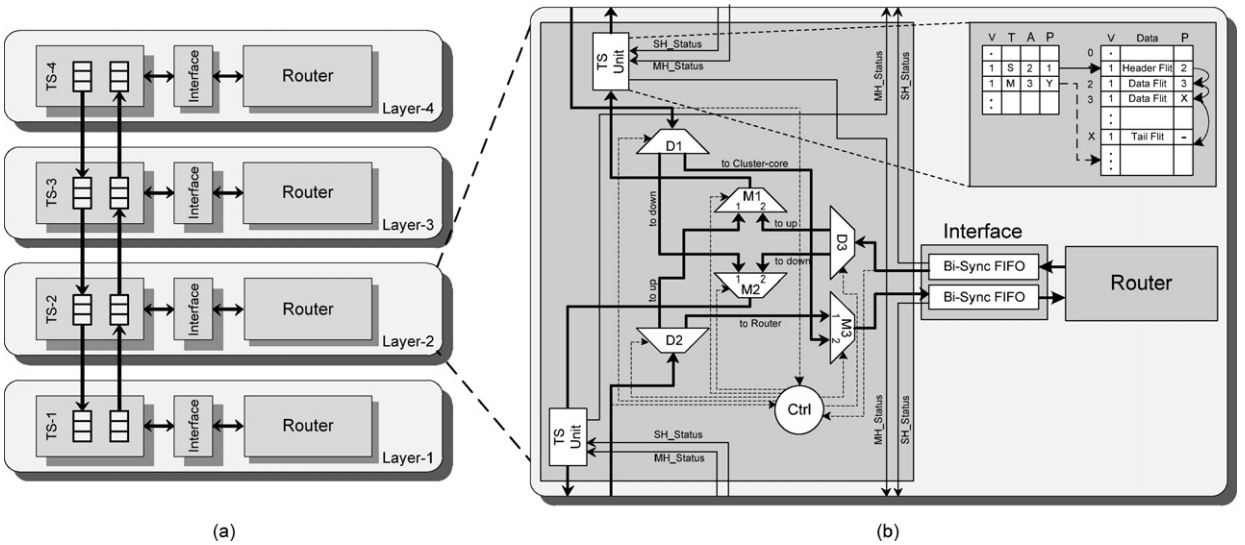

Fig. 4. (a) Proposed bus architecture and (b) the micro-architecture of the transfer stage.

queues for pipelining the data flow, and a bus segment between adjacent stages consists of two separate unidirectional point-to-point interconnects which transfer data synchronously (or asynchronously) between the stages in opposite directions. These two links of a segment can operate in parallel, and due to pipelining, all segments of the bus can transfer data simultaneously. Each layer has a unique address for inter-layer communication. Furthermore, each IP-core/memory in a layer has its own address which makes addressing of a specific module in a given layer possible. Hence, a datagram propagating along the bus has a header containing both the layer address and the IP-core/memory address. The former is analyzed at each transfer stage, and the latter is decoded by routers in each layer.

5.1. Transfer stage micro-architecture

The micro-architecture of the transfer stage is illustrated in Fig. 4(b) where it includes two identical pipelines transferring data to the opposite directions. Each pipeline contains multiple slots to pipeline packets between slots. Apart from the pipelines the interface contains FIFO queues used as input and output buffers of the host port. Their capacity has to be chosen according to the speed of the bus interface and the estimated data rate of the attached router. Each transfer stage also contains three multiplexers (M1, M2 and M3) and three de-multiplexers (D1, D2 and D3) to establish a communication between inputs and outputs of the transfer stage. On top of that, each transfer stage has the following functions:

1. It forwards incoming packets from the preceding stage to the next stage through a buffer, in both directions. That is, if the incoming packet from the upper stage (lower stage) is intended to be forwarded to the lower stage (upper stage), D1 and M2 (D2 and M1) will provide the required connections.
2. If an incoming packet from an adjacent transfer stage is intended to be processed by the router, the transfer stage delivers the packet to the interface module of the layer through a FIFO queue, i.e. D1 and M3 (D2 and M3) establishes the required connections to deliver the incoming packet from the upper stage (lower stage) to the router.
3. When a data is sent to another layer, the transfer stage operates as an output buffer. This means that it takes care of first receiving data from the interface module of the attached layer through a FIFO queue and then sending this data to one of the two adjacent transfer stages, depending on the direction in which, the target layer is located. Namely, D3 and M1 (D3 and M2) will be responsible for the required connections when the router decides to send a packet to upper layer (lower layer). When a packet arrives at a transfer stage, the header flit is sent to the controller unit to determine in which direction the packet should be sent. Based on the controller decision, it will be either forwarded to the next stage or transferred to the host router via the interface. Also, an arbitration in the controller module has to be performed to prevent the two parallel operating pipelines from writing simultaneously to the FIFO in the interface. In addition, because the electrical behavior of short and wide TSVs provides much higher signaling speeds, the credit-based flow control [4] has been implemented for the transmission protocol on a segment between transfer stages.

While the conventional bus architectures degrade the performance significantly, the presented bus is the enhanced approach of a 3-port router in a ring without the end-around connections in order to provide two important features: weight-based arbitration and a non-blocking scheme.

```
-- Multiplexers M1
w1= number of lower layers
w2= 1 (connected to one layer)
Process
Begin
   If input1='1' and counter ≤ w1 then
      service <= input1;
      counter <= counter + 1;
   Elsif input2='1' then
      service <= input2;
      counter <= 1;
   Else
      service <= input1;
      counter <= 1;
   End If;
End;
```

```
-- Multiplexers M2
w1= number of upper layers
w2= 1 (connected to one layer)
Process
Begin
   If input1='1' and counter ≤ w1 then
      service <= input1;
      counter <= counter + 1;
   Elsif input2='1' then
      service <= input2;
      counter <= 1;
   Else
      service <= input1;
      counter <= 1;
   End If;
End;
```

```
-- Multiplexers M3
w1= number of lower layers
w2= number of upper layers
Process
Begin
   If input1='1' and counter ≤ w1 then
      service <= input1;
      counter <= counter + 1;
   Elsif input2='1' then
      service <= input2;
      If counter =w1+w2 then
         counter <= 1;
      Else
         counter <= counter+1;
      End If;
   Else
      service <= input1;
      counter <= 1;
   End If;
End;
```

Fig. 5. Pseudo VHDL code of the weight-based arbitration for multiplexers: M1, M2, and M3.

5.2. Weight-based arbitration

Arbiters in the controller may use a round-robin policy to arbitrate between the inputs of the multiplexers. However by using round-robin arbitration policy in the transfer stage of the pipeline bus architecture, fairness can become a problem. Fairness is not an issue when the traffic load is low, but as the traffic load approaches saturation, fairness can become a bottleneck. Let us consider an example to illustrate the unfairness problem where three layers 1, 2 and 3 have large amounts of packets to be sent to the layer 4. Contention might occur in layers 2 and 3 as the packets must share the same channel resources. For instance, in the layer 3 there are two flows of packets competing for the bandwidth of the output. One flow is comprised of the packets being generated in the layer 3, and the packets in the other flow are arrived from the layer 1 and layer 2. In other words, the layer closest to the destination layer, layer 3, will get the most bandwidth, 1/2 of the available bandwidth. The remaining half of the bandwidth is allocated to the layers 1 and 2, so each can receive 1/4 of the total bandwidth. Therefore, the allocation of the available bandwidth to the competing flows is not fair if round-robin policy is used. To overcome this limitation, a weight-based arbitration is employed so that the weight of each input port is determined by the number of upstream layers connected to that input port through the pipeline bus. In the above example at the layer 3, the weight of one input port is two as it can accept packets from two lower layers ($w1 = 2$), and the weight of the other input port is one since it can receive data only from the current layer ($w2 = 1$). As a result, the arbiter transmits the maximum of $w1$ packets (if any) from the first flow and then allows the other flow to forward up to $w2$ packets (if any) to the output and the process is repeated for the rest of the packets. By this approach, in each layer the total bandwidth is equally shared among packets from different layers. This simple weighted round-robin arbitration achieves a fair forwarding policy with a very low hardware overhead. Fig. 5 is the pseudo code of the weight-based arbitration for multiplexers M1, M2 and M3.

5.3. Non-blocking scheme

The proposed pipeline structure allows simultaneous transmissions without using centralized bus arbitration, which considerably reduces arbitration complexity and improves bandwidth. However in the pipeline bus architecture, a single blocked packet might obstruct the subsequent packets so that increasing the communication latency. A blocking situation is shown in Fig. 6 where the packet A and packet B destined for the layer 3 and layer 4, respectively. When the packet A is blocked, the packet B can be obstructed behind the packet A. In order to prevent this blocking situation, we introduce two types of packet, Single-Hop (SH) packet and Multiple-Hop (MH) packet. SH packet destinations are located in one of the neighboring layers while MH packets require passing several layers. However, a MH packet changes its type to SH once the destination is one layer away.

In each transfer stage, the incoming packets are de-multiplexed into two separate paths: one path delivers the SH packets to the interface (SH path) while the other path forwards the MH packets to the next stage (MH path). That is, packets are de-multiplexed to either the TS unit or interface. The point is that if one of these two paths gets blocked, the remaining flows from the upstream stage cannot pass through the other path if it is idle. This blocking probability can be considerably reduced by considerably increasing the size of both the interface and TS unit buffers, which is an expensive solution for such systems. The idea is to reduce the blocking probability with a low hardware cost. Therefore, each stage

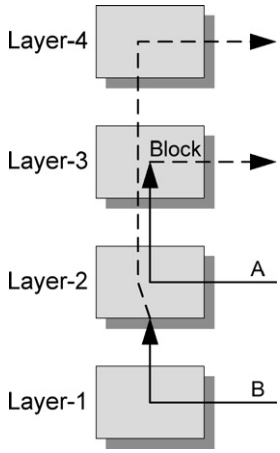

Fig. 6. A blocking situation.

adopts congestion condition of its downstream transfer stage buffers (interface and TS unit buffers) so that it can decide to deliver a packet to the less congested path of the next stage. The congestion condition of SH and MH paths, indicating the stress value of the interface and TS unit buffers, can be transmitted from one layer to another through two separate inter-layer signals (MH_Status and SH_Status). These signals are employed by the TS unit of the upstream transfer stage to forward a non-blocking packet, i.e. send a MH packet if the SH path is congested/blocked or send a SH packet if the MH path is congested/blocked.

As depicted in Fig. 4(b), each TS unit is composed of a table and a buffer. Each row of the table corresponds to a packet and includes a valid tag (v), a packet type (T), a packet age (A) and a header pointer (P). In the buffer, the flits of each packet are stored with a linked list structure providing high resource efficiency with a little hardware overhead. Fig. 4(b) exhibits a pointer field adopted to indicate the next flit position in the buffer. As multiple packets might be stored in the buffer, an arbitration mechanism is needed to determine which packet is allowed to be transmitted. The TS unit arbitration decision is based on the stress value of MH and SH paths. This arbiter selects the oldest packet (highest age) requesting an available path with the lowest stress value. Afterward, the age value of each packet having the same type as the selected one is increased by the arbiter to prevent starvation.

5.4. Synchronizing FIFO

Bi-synchronous (Bi-Sync) FIFOs are widely used in multi-clock systems to synchronize signals from different clock/frequency domains. Each domain is synchronous to its own clock signal but can be asynchronous with respect to others in either clock frequency or phase [37]. The challenges of designing Bi-Sync FIFOs include the enhancement of reliability and reducing latency and power/area cost. We identify the Bi-Sync FIFO structure presented in [38] as a suitable synchronizer to be used in the interfaces.

The structure of the Bi-Sync FIFO is depicted in Fig. 7. The FIFO implementation uses two pointers, one defining the next writing position and another defining the next reading position. The FIFO state is either full or empty when both pointers refer to the same address. Thus, it is necessary to compare the pointers. Although this procedure is trivial in synchronous circuits, it implies some complexity in the Bi-Sync FIFO, because the pointers are generated by different clocks. A common solution to this problem is to transfer and synchronize the writing pointer (reading pointer) with the receiver clock domain (the sender clock domain) which generates the empty signal (the full signal). Exchanging the pointers (write_ptr and read_ptr) via a handshake protocol implies additional latency. Therefore, two synchronizers are utilized for exchanging the pointers [38]. The addresses are converted to the gray code which guarantees that consecutive addresses are at a Hamming distance of 1. In this way, the metastability problem is confined to a single bit and synchronizers can be employed without handshake. Utilizing the Bi-Sync FIFO in the interfaces, allows each layer to work with its own clock source. Table 2 lists the input and output signals and their functionalities of the Bi-Sync FIFO.

6. Experimental results

In this section, we compare the presented topologies with the conventional structures in terms of latency, power consumption, and area cost. Also, the impact of using the novel pipeline bus has been explored. Hence, a cycle-accurate 3D NoC simulator is developed to assess the efficiency of the proposed architectures. The simulator models all major components of the NoC such as network interfaces [39], routers, and wires along with vertical channels.

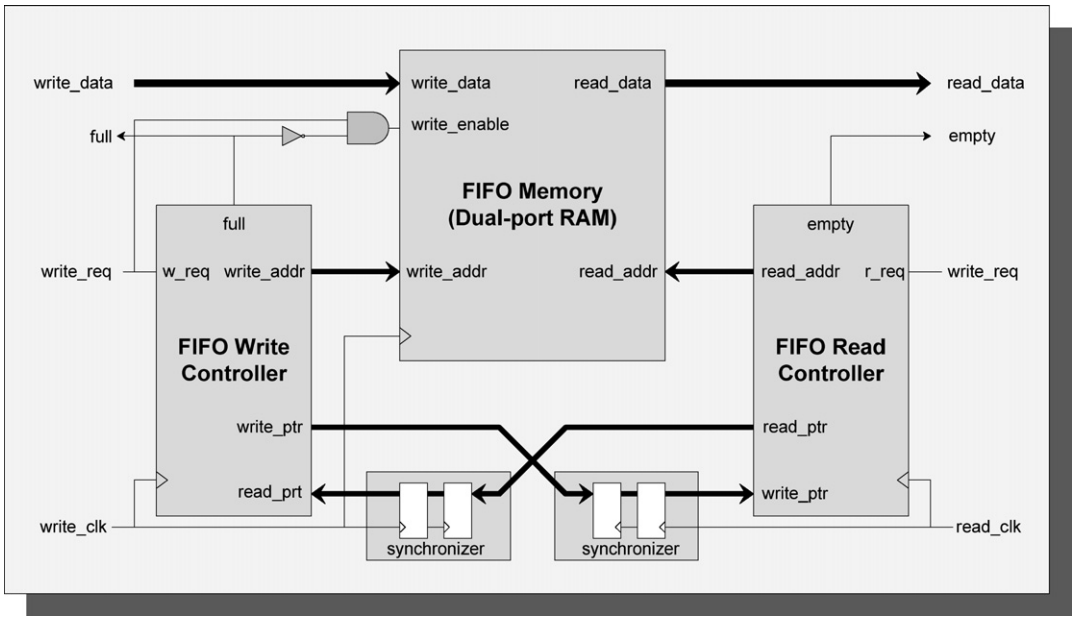

Fig. 7. Bi-Sync FIFO structure.

Table 2
Description of Bi-Sync FIFO signals.

Signal	Description
write_data	Data to be written in the FIFO
write_req	Write request
write_clk	Clock signal in the write domain
full	Signal to indicate the FIFO is full and no more data can be received
read_data	Data to be read from the FIFO
read_req	Read request
read_clk	Clock signal in the read domain
empty	Signal to indicate the FIFO is empty and henceno data can be read

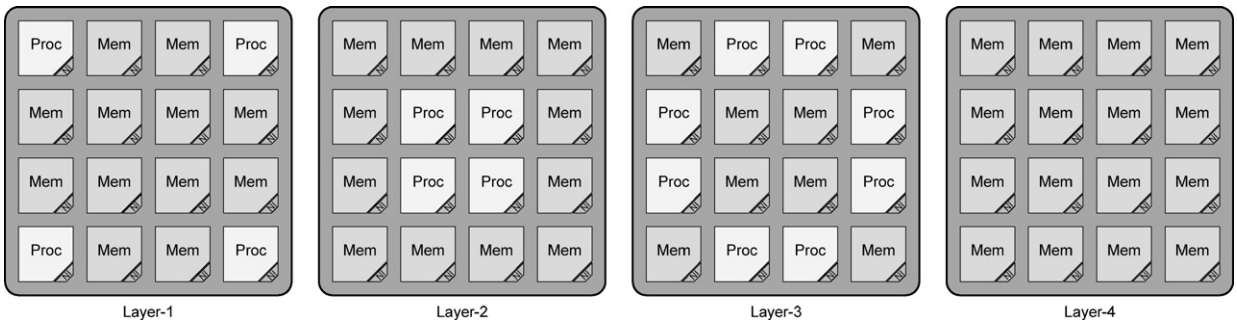

Fig. 8. $4 \times 4 \times 4$ stacked mesh layout.

6.1. System configuration

In this work, we configure a 64-node ($4 \times 4 \times 4$) 3D stacked architecture. In this configuration, illustrated in Fig. 8, out of 64 nodes, 16 nodes are assumed to be processors and other 48 nodes are memory blocks, i.e. DRAMs. The processors are 32b AXI compatible core and the memory blocks are DDR2-256MB (t_{RP}–t_{RCD}–$t_{CL} = 2$–2–2, 32b, 4 banks) [40]. Three different 3D on-chip network topologies are considered for experiment: 3D-hybrid structure, CIT, and CMIT. The 3D-hybrid and CMIT networks are formed by a typical state-of-the-art router structure including input buffers, a VC (Virtual Channel) allocator, a

routing unit, a switch allocator, and a crossbar as well as an interface unit connecting the router to either a vertical channel (bus) or a cluster router. Typical routers of 3D-hybrid and CMIT have at most six input/output ports. Every cluster router of CMIT has five input/output ports, i.e. four for local routers and one for the vertical channel interface, while cluster routers of CIT have at most nine input/output ports, i.e. four for local IP-cores/memories connections, at most four for neighboring cluster routers, and one for the vertical channel interface. Each input port of router has 2 VCs where packets of different message types (request and response) are assigned to corresponding VCs to avoid message dependency deadlock [41]. The arbitration scheme of the switch allocator in the typical router structure is round-robin. The array size, routing algorithm, link width, number of VCs, buffer depth of each VC, and traffic type are the other parameters which must be specified for the simulator. The routers adopt the DOR routing scheme and utilize wormhole switching [4,5,42]. For all routers, the data width (flit) was set to 32 bits, and the buffer depth of each VC is 5 flits. As the proposed pipeline bus is composed of two unidirectional pipeline channels, to be fair, conventional buses are also organized with two unidirectional channels in opposite directions to propagate the inter-layer communication data; so, each channel includes a separate arbiter along with control signals. Perhaps this approach imposes a considerable hardware cost for conventional buses but in turn not only improves the performance of conventional buses significantly but also facilitates a fair comparison between the proposed approach and the conventional buses. Thus, 32 bits of the channel are allocated to upward direction and the other 32 bits of the channel are employed for the downward direction. Each channel has its arbiter module and bus controller [10,33]. The depth of buffers in the transfer stage is 6 flits.

The presented configuration uses 1 flit for messages related to read requests and write responses, and the size of read request messages typically depends on the network size and memory capacity of the configured system. The message size of the read responses and write requests is variable and depends on the request/response length produced by a master/slave core (burst size 1:8). As for the performance metric, we use latency defined as the number of cycles between the initiation of a request operation issued by a master (processor) and the time when the response is completely delivered to the master from a slave (memory). The request rate is defined as the ratio of the successful read/write request injections into the network interface over the total number of injection attempts. All the cores and routers are assumed to operate at 1 GHz. For fair comparison, we keep the bisection bandwidth constant in all configurations. All memories (slave cores) can be accessed simultaneously by each master core with continuously generating memory requests. To estimate the power consumption of networks, we have used Orion [43] as well as the power and delay values of vertical links in [30].

6.2. Performance comparison

To assess the performance of the presented pipeline bus architecture, the uniform and non-uniform synthetic traffic patterns have been considered separately and we expect realistic applications stand between these two synthetic traffic patterns. The random traffic represents the most generic case, where each processor sends in-order read/write requests to memories with the uniform probability, and the memories and request type (read or write) are selected randomly. Eight burst sizes, from 1 to 3, are stochastically chosen according to the data length of the request. In the non-uniform mode, 70% of the traffic is local requests, where the destination memory is one hop away from the master core, and the rest 30% of the traffic is uniformly distributed to the non-local memory modules.

Here, we explore the average latency of using the presented pipeline bus architecture. Two conventional baseline buses, dTDMA [10,33] and SAMBA [36], have been considered to be used for vertical channels. Fig. 9(a)–(c) and Fig. 10(a)–(c) show the performance gain of employing the pipeline bus architecture for vertical channels in the 3D-hybrid, CMIT and CIT networks under uniform and non-uniform traffic profiles, respectively. This performance gain is achieved due to a small low-latency arbiter in each transfer stage, reducing the arbitration delay significantly. Fig. 9(d)–(e) and Fig. 10(a)–(c) present throughput, another key metric of network performance. These figures follow the same trend as the network latency experiments, where using the pipeline bus clearly outperforms the conventional bus schemes.

To explore the efficiency of the two presented topologies without considering the pipeline bus (employing dTDMA), the simulation results under the uniform, non-uniform, and hotspot traffic profiles are depicted in Fig. 11. In the hotspot traffic pattern, one or more nodes are designated as hotspot nodes receiving an extra portion of the traffic in addition to the regular uniform traffic. Newly generated packets are directed to each hotspot node with an additional H percent probability. We simulate hotspot traffic with four hotspot nodes. Four hotspot nodes are chosen at the center of each layer, $(2, 2, 1)$, $(3, 3, 2)$, $(2, 3, 3)$, $(3, 2, 4)$, with equal probability of $H = 20\%$.

As demonstrated in Fig. 11(a) and (c), CIT has the lowest average latency in the low traffic load (< 0.2), one of the foremost reasons for such an improvement is that CIT reduces the average hop count and improves load balance across the channels. But in high traffic load the performance of CIT degrades considerably since the network bandwidth in CIT is lower than that of mesh-based structures. That is, the number of links in CIT is much smaller than that of mesh-based structures. Therefore, in the high traffic load, the traffic in CIT links is much higher than in mesh-based structures. Another subtle point regarding clustered topology is that the latency in CIT for distant destinations is significantly larger than that of mesh-based structures due to the high router complexity and contention probability, while for nearby destinations the latency of CIT is smaller. Thus, CIT might have better performance in applications where most of requests are issued among neighboring nodes under low traffic load. This can be seen from the results in Fig. 11(b) where each processor sends requests to the memories based on the non-uniform traffic profile. CIT outperforms the others in terms of latency when the request rate is below the saturation point and most of the traffic is local. The average latency of each presented topology has been

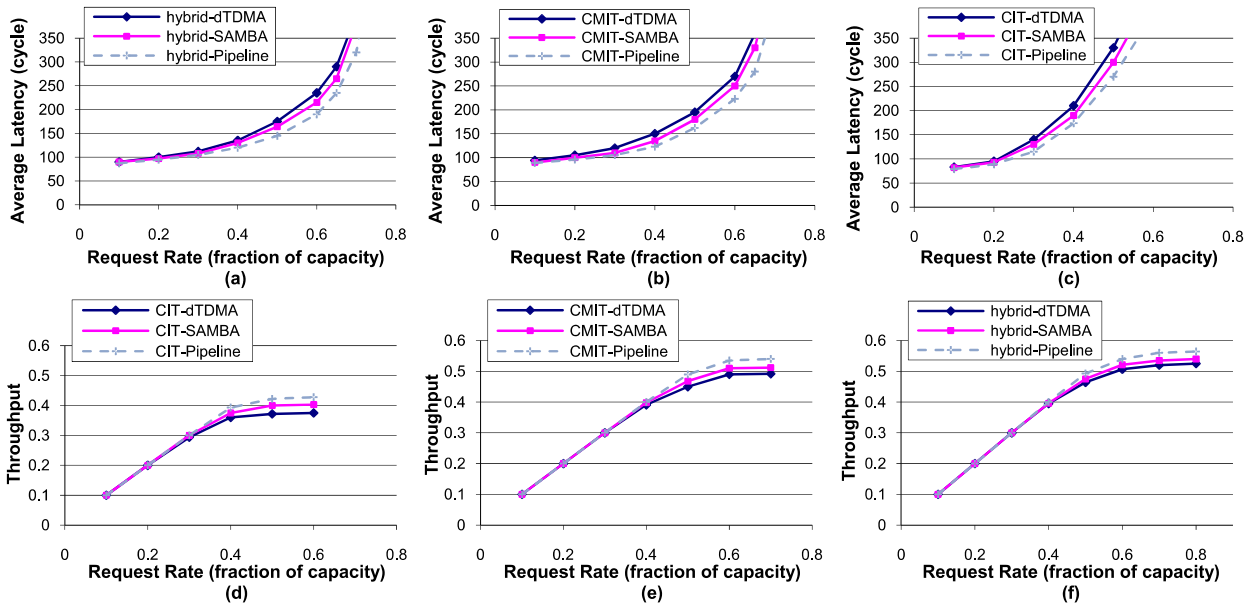

Fig. 9. Performance impact of using the presented bus in hybrid, CMIT, and CIT networks under uniform traffic profile.

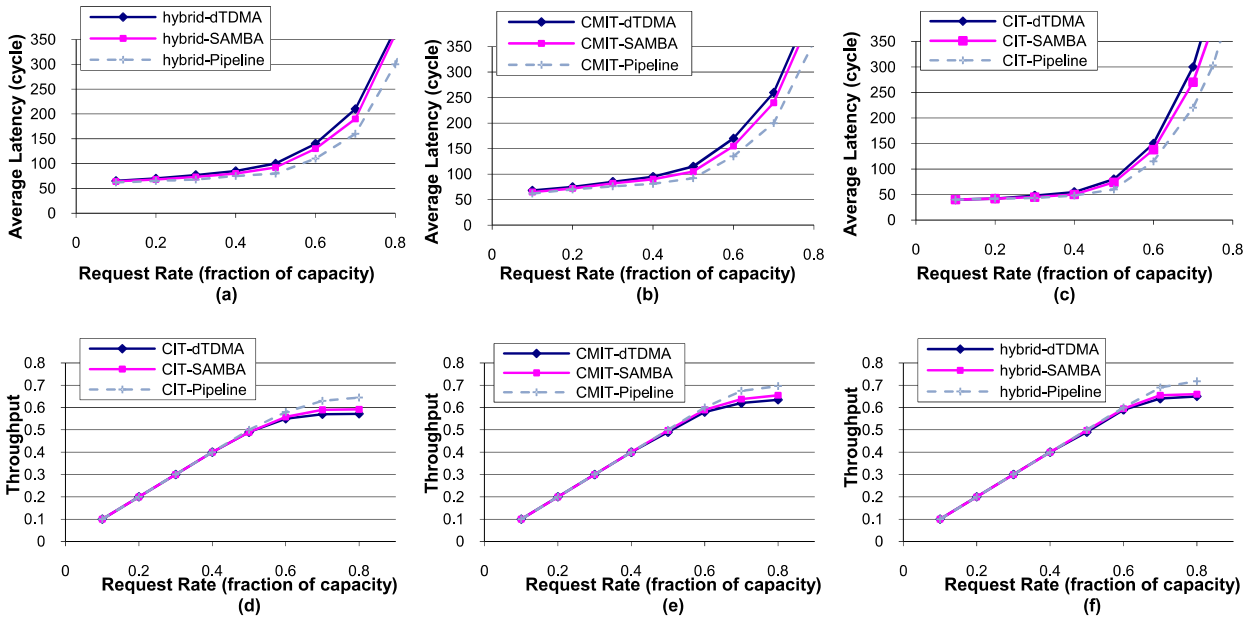

Fig. 10. Performance impact of using the presented bus in hybrid, CMIT, and CIT networks under non-uniform traffic profile.

computed near saturation point (0.5) under the non-uniform traffic profile. As a result, compared with the 3D-hybrid and CMIT, the average latency of CIT is reduced by 20% and 30%, respectively.

Fig. 12(a)–(c) demonstrates the performance impact of the pipeline bus on different topologies for uniform, non-uniform, and hotspot traffic profiles. As illustrated, employing the pipeline bus for the hybrid structure outperforms the 3D-symmetric structure where each router includes 7 ports. This performance gain is due to the fact that the pipeline scheme is inspired from typical router which is optimized for inter-layer communication and can benefit from the weight-based arbitration and the non-blocking scheme.

Additionally, to illustrate how local traffic under the non-uniform profile can affect the performance, we scale the amount of local traffic from 0 to 100%. The results obtained at rate 0.5 (near the saturation point) are shown in Fig. 13 where CIT, CMIT, and hybrid utilize the proposed pipeline bus. Like in the previous non-uniform experiments, CIT achieves significant latency reduction when the amount of local traffic is increased, particularly from 40%.

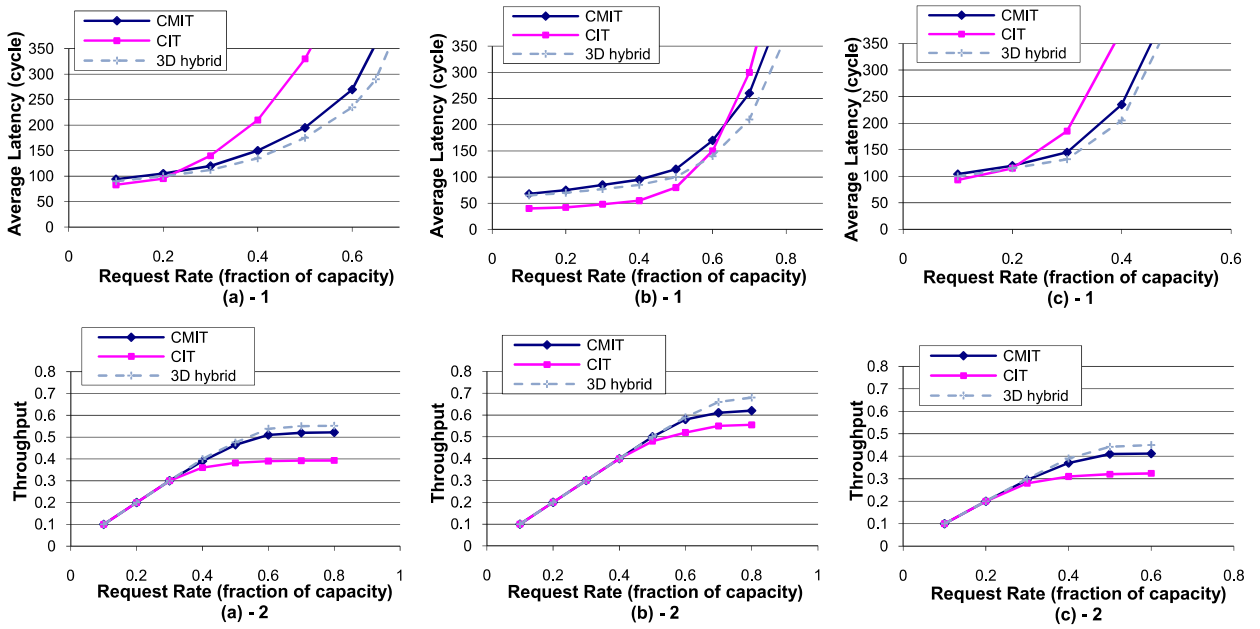

Fig. 11. Performance comparison of different 3D structures for (a) the uniform, (b) non-uniform, and (c) hotspot traffic profiles using dTDMA.

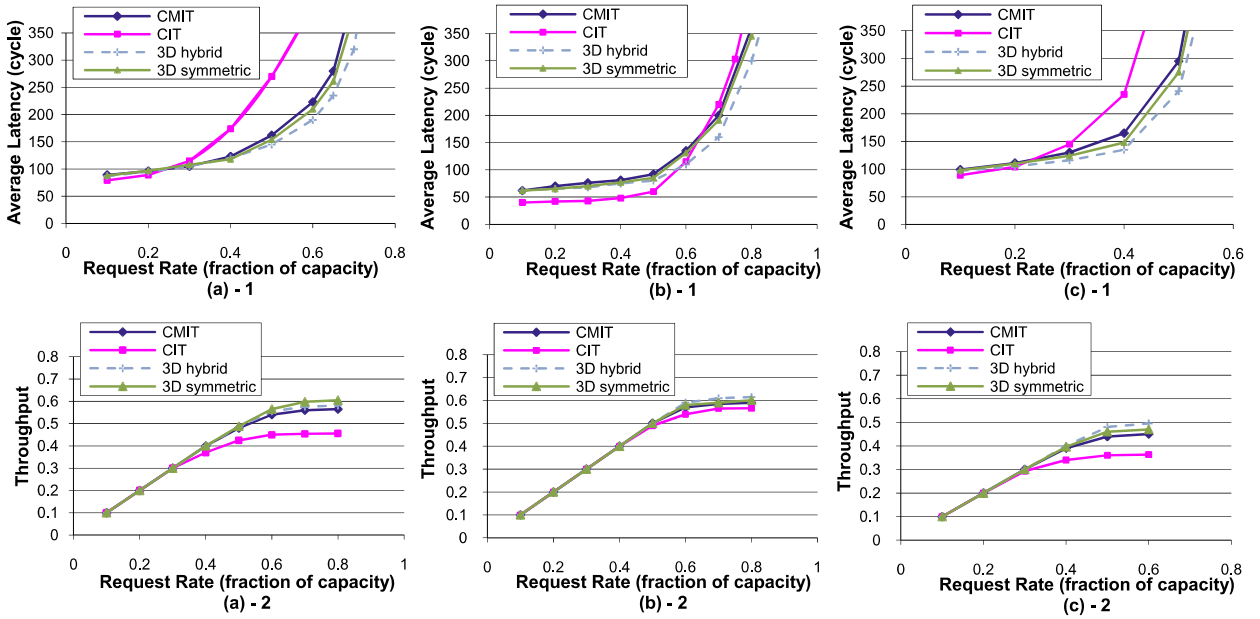

Fig. 12. Performance comparison of different 3D structures for (a) uniform, (b) non-uniform, and (c) hotspot traffic profiles using the presented bus.

In order to explore the real impact of the proposed inter-layer scheme, we use traces generated using the GEMS simulator [44] from SPLASH-2. We use the Radix, Ocean, and FFT applications from SPALSH-2 for our simulations. Table 3 summarizes our full system configuration where the cache coherence protocol is MESI [45] and access latency to the L2 cache is derived from the CACTI [46]. We configure a 64-node on-chip network ($4 \times 4 \times 4$) that four layers are stacked on top of each other, i.e. out of the 64 nodes, 16 nodes are processors and other 48 nodes are L2 caches. L2 caches are distributed in the bottom three layers, while all the processors are placed in the top layer close to a heat sink so that the best heat dissipation capability is achieved [8,11]. The simulator produces, as output, the communication latency for cache access. The CIT, CMIT, and hybrid configurations are equipped with the pipeline bus. Fig. 14 shows the average network latency of the real workload traces collected from the aforementioned system configurations. We can see that the hybrid configuration consistently reduces the average network latency across all tested benchmarks. It shows a steady reduction

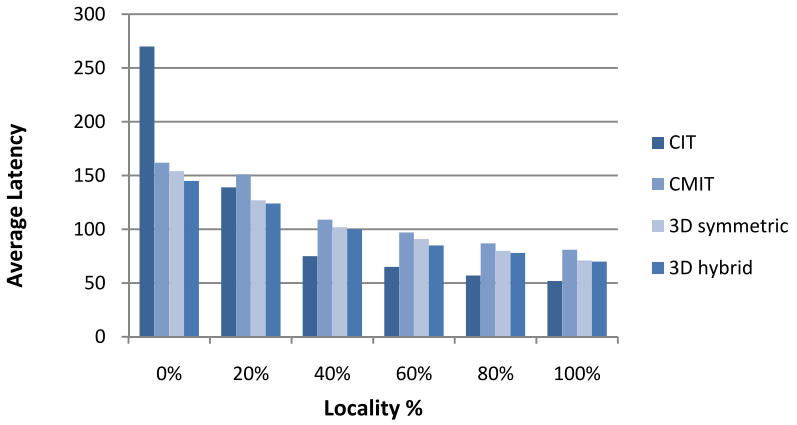

Fig. 13. Performance impact of different topologies using presented bus.

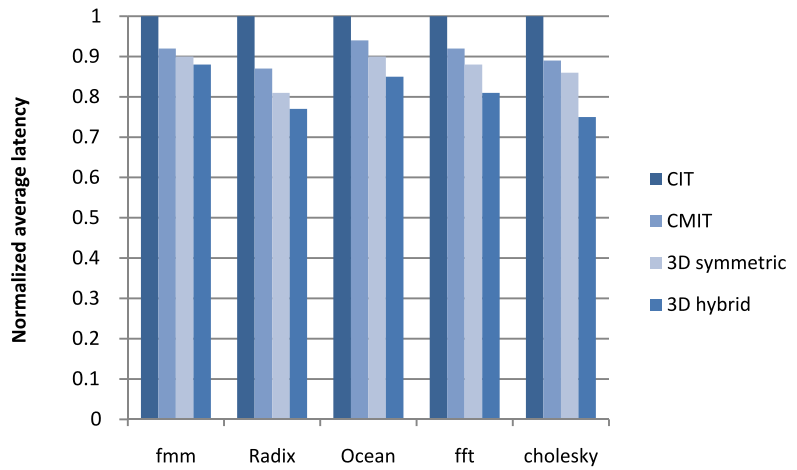

Fig. 14. Performance for application traces normalized to CIT.

Table 3
System configuration parameters.

16 SPARC cores, issue width = 1
L1 cache: private, split instruction and data cache, each cache is 16KB. 4-way associative, 64-bit line, 3-cycle access time
L2 cache: shared, distributed in 4 layers, unified 48MB (28 banks, each 1MB). 64-bit line, 6-cycle access time, SNUCA MESI protocol
Main memory: 4GB DRAM, 260 cycles, 16 outstanding
Router: wormhole, 2 cycles, 32-bit flit

amount: 12%–25% (hybrid/CIT), 4%–16% (hybrid/CMIT), and 3%–13% (hybrid/symmetric) with the average of 19%, 10%, and 7%, respectively.

6.3. Power analysis

Using the simulator, the average power consumption of the presented topologies was calculated and compared under uniform and non-uniform traffic patterns close to the saturation point. As mentioned earlier, to estimate the power consumption of each 3D network, the Orion [43] libraries (estimate both dynamic and static power) as well as the power and delay values of vertical links presented in [30] are used in the simulator. The results are shown in Fig. 15(a) and (b). According to Fig. 11 (a) and (b), the saturation points that have been considered for computing the average power values are 0.3 for uniform traffic and 0.5 for non-uniform traffic. As presented in Fig. 15(a), under the uniform traffic profile, the average power dissipation of the CIT scheme is 30%, 16% and 10% less than those of the 3D-symmetric, 3D-hybrid and CMIT schemes, respectively. Furthermore, the results in Fig. 15(b) indicate that the average power of CIT, under the non-uniform

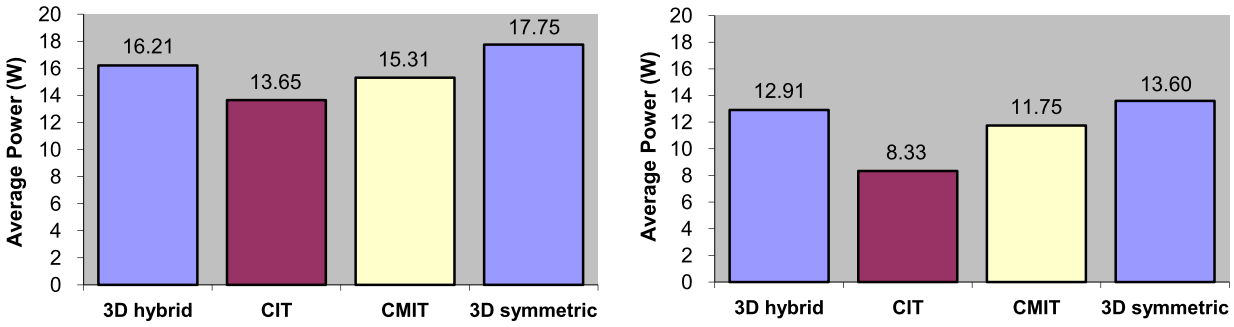

Fig. 15. Average power dissipation results under (a) uniform and (b) non-uniform traffic profiles.

traffic profile, is 39%, 35%, and 30% less than that of the 3D-symmetric, 3D-hybrid, and CMIT schemes, respectively. We can notice that although the power consumption of every cluster router is about 1.5 times more than the power consumption of a typical router, the average power in the CIT network, compared to other schemes, is considerably lower under non-uniform traffic profile since the average number of hops between two arbitrary nodes is less than in the other presented schemes. Also, to illustrate how the proposed pipeline bus affects the power dissipation, we compute the average power of each network close to its saturation point under the uniform traffic profile. Based on the achieved results, the average power consumption of the pipeline bus in the hybrid network is diminished by 10% and 12% compared with SAMBA and dTDMA, respectively. The average power reduction of using pipeline bus in the CMIT network is 6% and 9% compared with SAMBA and dTDMA, respectively, while in the CIT network, it is 5% and 10%. In fact, this power saving is obtained because of the following reasons. First, the hardware overhead of the pipeline bus is smaller than that of SAMBA and dTDMA. Second, central arbiters, employed in SAMBA and dTDMA, cause a lot of switching compared to small local arbiters used in pipeline bus so that the power dissipation of those two buses are higher than that of the pipeline bus.

6.4. Physical analysis

The number of routers and vertical channels in a chip affects the area and implementation cost. Thus, to compute the network area for each topology, we estimate the area of routers, cluster routers, and vertical channels. The network platform of each topology with the aforementioned configuration is synthesized using Synopsys Design Compiler with the UMC 0.09 µm technology, while the backend is performed with the Cadence Encounter tool. Depending on the technology and manufacturing process, the pitch of TSVs can range from 1 µm to 10 µm [10,11,16]. In this work, the pad size for TSVs is assumed to be 5 µm square with pitch of around 8 µ, the flit-width is set to 32 bits, and each vertical channel requires 3×14 control wires for arbitration in four-stacked layers [10]. Hence, after the TSV area is calculated with the given values, the TSV footprint has been reduced from 0.41 mm^2 in 3D-hybrid to 0.1 mm^2 in CIT and CMIT, resulting in about 75% area saving for the TSV footprint. 3D-hybrid occupies a larger network area than CIT and CMIT, because each router in 3D-hybrid has a transceiver module to interface with the vertical channel (bus) [10] and each bus should have its own arbiter module. Like 3D-hybrid, the transceiver and arbiter modules are only integrated in cluster routers of CIT and CMIT. On the other hand, the total network area used by CIT is significantly smaller than that of the other architectures (37% and 42% less than that of CMIT and 3D-hybrid respectively) since the network is formed only by cluster routers.

Using the pipeline bus reduces the TSV footprint for each vertical channel. As each vertical channel, i.e. two unidirectional 32-bit dTDMA buses, occupies 6400 μm^2 and the required area for each vertical channel using the pipeline bus is 4096 μm^2, the proposed bus scheme can save more than 35% of the TSV area footprint with a performance gain. Hence, after considering the TSV footprint area, the hardware overhead of the pipeline bus is approximately 10% and 8% less than that of the conventional dTDMA and SAMBA buses, respectively. In addition, comparing the cost of the network using the presented pipeline bus with the 3D-symmetric network reveals that the area overhead of the 3D-symmetric network is about 10%. This is because a 7-port router is larger than a 6-port router.

7. Conclusion

3D stacked architectures provide significant benefits in performance, footprint and yield. It has been demonstrated that combining 3D ICs and on-chip networks can be a promising option for designing large multiprocessor architectures. One critical issue in 3D design is that the vertical interconnections are very fast and fat such that the area overhead of TSVs impose constraints on the number of TSVs for existing 3D architectures. In this paper, two cluster-based topologies have been presented to deal with constraints on the number of TSVs. Also, a novel pipeline bus structure for vertical channels is introduced not only to mitigate the drawbacks of existing bus structures in terms of power and performance, but also to reduce the number of required inter-layer arbiter control signals. Experimental results revealed that the on-chip network formed by the two presented topologies (CIT and CMIT) reduces the number of TSVs significantly with low performance penalty under

uniform traffic, but under non-uniform traffic which is more realistic case, CIT outperformed the other network structures in terms of the average network latency.

References

[1] L. Benini, G. De Micheli, Networks on chips: A new SoC paradigm, IEEE Comput. (January 2002) 70–78.
[2] P. Magarshack, P.G. Paulin, System-on-chip beyond the nanometer wall, in: Proceedings of 40th Design Automation Conf. (DAC 03), ACM Press, 2003, pp. 419–424.
[3] W.J. Dally, B. Towles, Route packets, not wires: On-chip interconnection networks, in: Proceedings of the 38th Design Automation Conference, June 2001, pp. 684–689.
[4] A. Jantsch, H. Tenhunen, Networks on Chip, Kluwer, New York, 2003.
[5] J. Duato, S. Yalamanchili, L.M. Ni, Interconnection Networks: An Engineering Approach, Morgan Kaufmann Publishers, 2003.
[6] K. Banerjee, S.J. Souri, P. Kapur, K.C. Saraswat, 3D ICs: A novel chip design for improving deep-submicrometer interconnect performance and systems-on-chip integration, Proc. IEEE 89 (5) (May 2001) 602–633.
[7] B.S. Feero, P.P. Pande, Networks-on-Chip in a three-dimensional environment: A performance evaluation, IEEE Trans. Comput. 58 (1) (January 2009) 32–45.
[8] D. Park, S. Eachempati, R. Das, A.K. Mishra, Y. Xie, N. Vijaykrishnan, C.R. Das, MIRA: A multi-layered on-chip interconnect router architecture, in: ISCA, Pennsylvania State, USA, 2008, pp. 251–261.
[9] M. Daneshtalab, M. Ebrahimi, J. Plosila, HIBS-novel inter-layer bus structure for stacked architectures, in: Proceedings of IEEE International 3D Systems Integration Conference (3DIC), January 2012, Japan, pp. 1–7.
[10] F. Li, C. Nicopoulos, T. Richardson, Y. Xie, V. Narayanan, M. Kandemir, Design and management of 3D chip multiprocessors using network-in-memory, in: 33rd International Symposium on Computer Architecture (ISCA), 2006, pp. 130–141.
[11] I. Loi, L. Benini, An efficient distributed memory interface for many-core platform with 3D stacked DRAM, in: Proc. of the DATE Conference, Germany, 2010, pp. 99–104.
[12] W.R. Davis, J. Wilson, S. Mick, J. Xu, H. Hua, C. Mineo, A.M. Sule, M. Steer, P.D. Franzon, Demystifying 3D ICs: The pros and cons of going vertical, IEEE Des. Test Comput. 22 (6) (November 2005) 498–510.
[13] I. Loi, F. Angiolini, L. Benini, Supporting vertical links for 3D networks on chip: Toward an automated design and analysis flow, in: Proc. Nanonets, 2007.
[14] K. Snoeckx, E. Beyne, B. Swinnen, Copper-nail TSV technology for 3D-stacked IC integration, Solid State Technol. 50 (5) (2007) 53.
[15] A.Y. Weldezion, M. Grange, D. Pamuwa, Z. Lu, A. Jantsch, R. Weerasekera, H. Tenhunen, Scalability of the Network-on-Chip communication architecture for 3D meshes, in: International Symposium on Networks-on-Chip (NoCS), 2009, pp. 114–123.
[16] I. Savidis, S.M. Alam, A. Jain, S. Pozder, R.E. Jones, R. Chatterjee, Electrical modeling and characterization of through-silicon vias (TSVs) for 3D integrated circuits, Microelectron. J. 41 (1) (2010) 9–16.
[17] S. Pasricha, Exploring serial vertical interconnects for 3D ICs, in: Proc. IEEE/ACM DAC, 2009, pp. 581–586.
[18] Young-Joon Lee, Yoon Jo Kim, Gang Huang, Muhannad Bakir, Yogendra Joshi, Andrei Fedorov, Sung Kyu Lim, Co-design of signal, power, and thermal distribution networks for 3D ICs, in: Design, Automation & Test in Europe Conference & Exhibition (DATE), 2009, pp. 610–615.
[19] S. Das, A. Chandrakasan, R. Reif, Three dimensional integrated circuits: Performance, design, methodology and CAD tools, in: Proc. of ISVLSI, 2003, pp. 13–18.
[20] S. Das, et al., Technology, performance, and computer aided design of three-dimensional integrated circuits, in: Proc. International Symposium on Physical Design, USA, 2004, pp. 108–115.
[21] Y. Xie, G.H. Loh, B. Black, K. Bernstein, Design space exploration for 3D architectures, ACM J. Emerg. Technol. Comput. Syst. 2 (2006) 65–103.
[22] S.-M. Jung, et al., The revolutionary and truly 3-dimensional $25F^2$ SRAM technology with the smallest S^3 cell, 0.16 um^2, and SSTFT for ultra high density SRAM, in: IEEE Symp. VLSI Tech., 2004, pp. 228–229.
[23] B. Black, et al., Die-stacking (3D) microarchitecture, in: Proceedings of MICRO-39, 2006, pp. 469–479.
[24] G.H. Loh, 3D-stacked memory architectures for multi-core processors, in: Proc. of International Symposium on Computer Architecture (ISCA), 2008, pp. 453–464.
[25] M. Daneshtalab, M. Ebrahimi, P. Liljeberg, J. Plosila, H. Tenhunen, Cluster-based topologies for 3D stacked architectures, in: Proceedings of ACM International Conference on Computing Frontiers (CF), May 2011, Italy.
[26] Z. Li, et al., Efficient thermal-oriented 3D floorplanning and thermal via planning for two-stacked-die integration, ACM TODAES 11 (2) (April 2006) 325–345.
[27] M. Mondal, A. Ricketts, S. Kirolos, T. Ragheb, G. Link, V. Narayanan, Y. Massoud, Thermally robust clocking schemes for 3D integrated circuits, in: Proceedings of the IEEE Design Automation and Test in Europe, (DATE), Nice, France, 2007.
[28] Q. Wu, K. Rose, J.-Q. Lu, T. Zhang, Impacts of though-DRAM vias in 3D processor-DRAM integrated systems, in: IEEE Int' 3D System Integration Conf., (3DIC), September 2009, pp. 1–6.
[29] M. Daneshtalab, M. Ebrahimi, P. Liljeberg, J. Plosila, H. Tenhunen, CMIT – A novel cluster-based topology for 3D stacked architectures, in: Proceedings of IEEE International 3D Systems Integration Conference (3DIC), November 2010, Germany, pp. 1–5.
[30] V.F. Pavlidis, E.G. Friedman, 3D topologies for Networks-on-Chip, IEEE Trans. Very Large Scale Integr. 15 (10) (2007) 1081.
[31] M. Ebrahimi, M. Daneshtalab, P. Liljeberg, J. Plosila, H. Tenhunen, Exploring partitioning methods for 3D Networks-on-Chip utilizing adaptive routing model, in: Proceedings of 5th ACM/IEEE International Symposium on Networks-on-Chip (NOCS), USA, May 2011, pp. 73–80.
[32] S. Murali, C. Seiculescu, L. Benini, G.D. Micheli, Synthesis of networks on chips for 3D systems on chips, in: Proceedings of the 14th Asia and South Pacific Design Automation Conference (ASPDAC'09), January 2009, pp. 242–247.
[33] T. Richardson, C. Nicopoulos, D. Park, V. Narayanan, Y. Xie, C. Das, V. Degalahal, A hybrid SoC interconnect with dynamic TDMA-based transaction-less buses and on-chip networks, in: Proc. VLSID, 2006, pp. 8–15.
[34] Y. Qian, Z. Lu, W. Dou, From 2D to 3D NoCs: A case study on worst-case communication performance, in: Proc. of the International Conference on Computer-Aided Design (ICCAD), 2009, pp. 555–562.
[35] A.Y. Weldezion, Z. Lu, R. Weerasekera, H. Tenhunen, 3D memory organization and performance analysis for multi-processor Network-on-Chip architecture, in: Proc. of IEEE International 3D System Integration Conference, San Francisco, USA, 2009.
[36] R. Lu, A. Cao, C. Koh, SAMBA-Bus: A high performance bus architecture for system-on-chips, IEEE Trans. Very Large Scale Integr. 15 (1) (January 2007) 69–79.
[37] T. Ono, M. Greenstreet, A modular synchronizing FIFO for NoCs, in: Proc. of the 3rd ACM/IEEE International Symposium on Networks-on-Chip, USA, 2009, pp. 224–233.
[38] C.E. Cumings, Simulation and synthesis techniques for asynchronous FIFO design, in: SNUG, 2002.
[39] M. Daneshtalab, M. Ebrahimi, P. Liljeberg, J. Plosila, H. Tenhunen, Memory-efficient on-chip network with adaptive interfaces, IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. (IEEE-TCAD) 31 (1) (January 2012).

[40] Micron Technology, Inc. Micron 256Mb: x4, x8, x16 DDR2 SDRAM Datasheet, 2006.
[41] S. Murali, et al., Designing message-dependent deadlock free networks on chips for application-specific systems on chips, in: Proc. VLSI-SoC, 2006, pp. 158–163.
[42] P. Lotfi-Kamran, M. Daneshtalab, Z. Navabi, C. Lucas, BARP – a dynamic routing protocol for balanced distribution of traffic in NoCs to avoid congestion, in: Proceedings of 11th ACM/IEEE Design, Automation, and Test in Europe Conference (DATE), Germany, March 2008, pp. 1408–1413.
[43] A. Kahng, B. Li, L.-S. Peh, K. Samadi, Orion 2.0: A fast and accurate NoC power and area model for early-stage design space exploration, in: Proc. of DATE, 2009, pp. 423–428.
[44] M.M.K. Martin, D.J. Sorin, B.M. Beckmann, M.R. Marty, M. Xu, A.R. Alameldeen, K.E. Moore, M.D. Hill, D.A. Wood, Multifacet's general executiondriven multiprocessor simulator (GEMS) toolset, SIGARCH Comput. Archit. News 33 (4) (2005) 92–99.
[45] A. Patel, K. Ghose, Energy-efficient MESI cache coherence with pro-active snoop filtering for multicore microprocessors, in: Proc. Low Power Electronics and Design, 2008, pp. 247–252.
[46] N. Muralimanohar, et al., Optimizing NUCA organizations and wiring alternatives for large caches with cacti 6.0, in: Proc. 40th IEEE/ACM International Symposium on MICRO, 1–5 December 2007, pp. 3–14.