Cost-Effective Design of Mesh-of-Tree Interconnect for Multicore Clusters With 3-D Stacked L2 Scratchpad Memory

Kyungsu Kang, *Member, IEEE*, Luca Benini, *Fellow, IEEE*, and Giovanni De Micheli, *Fellow, IEEE*

***Abstract*— 3-D integrated circuits (3-D ICs) offer a promising solution to overcome the scaling limitations of 2-D ICs. However, using too many through-silicon-vias (TSVs) pose a negative impact on 3-D ICs due to the large overhead of TSV (e.g., large footprint and low yield). In this paper, we propose a new TSV sharing method for a circuit-switched 3-D mesh-of-tree (MoT) interconnect, which supports high-throughput and low-latency communication between processing cores and 3-D stacked multibanked L2 scratchpad memory. The proposed method supports traffic balancing and TSV-failure tolerant routing. The proposed method advocates a modular design strategy to allow stacking multiple identical memory dies without the need for different masks for dies at different levels in the memory stack. We also investigate various parameters of 3-D memory stacking (e.g., fabrication technology, TSV bonding technique, number of memory tiers, and TSV sharing scheme) that affect interconnect latency, system performance, and fabrication cost. Compared to conventional MoT interconnect [6] that is straightforwardly adapted to 3-D integration, the proposed method yields up to $\times 2.11$ and $\times 1.11$ improvements in terms of cost efficiency (i.e., performance/cost) for microbump TSV bonding and direct Cu–Cu TSV bonding techniques, respectively.**

***Index Terms*— 3-D integration, multicore, networks-on-chip (NoC), scratchpad memory (SPM).**

I. Introduction

NOWADAYS, the increasingfocus on energy-efficient architecture coupled with a slowdown in clock speed improvement has brought a growing interest in parallel computing. General purpose graphics processing units (GP-GPUs), such as NVIDIA Fermi [1], HyperCore [2], and STMicroelectronics Platform 2012 [3], are visible examples in this trend. All of the cited architectures share a common trait: a multicore cluster consisting of many simple cores with a private or shared L1 cache, a shared L2 cache, and a shared scratchpad memory (SPM). SPM, which is also termed tightly coupled memory, is an on-chip static RAM (SRAM) array with only decoding and column circuits. SPM yields much higher storage density per unit area, lower power consumption, and lower access latency than cache memory [4], [5].

3-D integration is a promising option to overcome the scaling limitations of 2-D integrated circuit (2-D IC) [7]. The main benefits of 3-D integration rely on the fact that long global wires are shortened owing to the additional vertical routing paths as well as the reduced die size as the number of stacked tiers increases [8]. However, 3-D IC technology also faces challenges due to the larger pitch of through-silicon-vias (TSVs) that takes up space in the active layers and has at least an order of magnitude greater footprint than regular vias in the metal layers. These TSVs are spread out (uniformly or non-uniformly) in each tier, which will make floorplanning and routing extremely challenging [10]–[13]. When considering the large footprint and parasitic capacitance of a real TSV, the global wire delay does not decrease substantially and continuously with the increased number of stacked tiers [14]. In addition, TSVs are usually etched or drilled through device layers by special techniques and are costly to fabricate. Large numbers of TSVs degrade fabrication yield of the final chip, resulting in high fabrication cost [15].

In this paper, we focus on interconnection networks within a multicore cluster consisting of multiple cores and a shared multibanked L2 SPM, where L2 SPM banks are stacked on top of the multicore die. The high density and the low latency of L2 SPM make it a very interesting option for 3-D integration of multicore clusters as 3-D integration enables a massive increase of SPM relatively close to the cores. Not only application processors, but also almost all mobile system-on-chips (SoCs) feature a pretty large L2 on-chip memory, which is shared by multiple cores. TI OMAP 5 Platform [16], STE NOVATHOR Platform [17], and Snapdragon S4 Processors [18] are just a few representative examples. However, the main issue with these architectures is that the on-chip memory has a limited number of ports (i.e., 1 or 2 ports) and the limited bandwidth in input/output (I/O) due to the long on-chip planar interconnect. On the other hand, we are proposing an architecture, which gives a bandwidth proportional to N, where $N(\leq 32)$ is the number of cores within a multicore cluster, and a latency that is just a few clock cycles

Manuscript received December 6, 2012; revised July 14, 2013 and March 6, 2014; accepted July 16, 2014. Date of publication September 4, 2014; date of current version August 21, 2015. This work was supported in part by the NanoSys Project under Grant ERC-AdG-246810 and in part by the European Research Council (ERC) Multitherman Project under Grant ERC-AdG-291125.

K. Kang is with the Memory Business, Samsung Electronics, Hwaseong 445-330, Korea (e-mail: kyungsu.kang@gmail.com).

L. Benini is with the Department of Information Technology and Electrical Engineering, Swiss Federal Institute of Technology Zurich, Zurich 8092, Switzerland, and also with the Department of Electrical, Electronic and Information Engineering, University of Bologna, Bologna 40136, Italy (e-mail: lbenini@iis.ee.ethz.ch).

G. De Micheli is with the Integrated Systems Laboratory, École Polytechnique Fédérale de Lausanne, Lausanne 1015, Switzerland (e-mail: giovanni.demicheli@epfl.ch).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TVLSI.2014.2346032

(i.e., <5 cycles).[1] The fully combinational mesh-of-tree (MoT) interconnect proposed in [6] is suitable for this architecture that needs an interconnect with high throughput and low latency. In [6], the fully combinational circuit-switched MoT interconnect was fabricated in a 65-nm technology node, featuring single-cycle transfer from core to on-chip memory and vice versa. The MoT interconnect provides distributed round-robin arbitration for fair access to memory banks as well as fine-grained address interleaving to reduce memory bank conflicts. However, a straightforward extension of the traditional MoT interconnect to the third dimension by simply inserting TSVs at every connection to the memory banks (which we call plain MoT) is not a good option, since it requires too many TSVs.

Even though traditional packet-switched on-chip interconnects provide bandwidth scalability, the latency is not adequate [29]. For reducing the latency, some researchers proposed high-radix on-chip interconnects, such as Clos [73] and flattened butterfly [74], which can decrease the diameter of the network and, thus, reduce the overall latency. However, the long wires of the high-radix interconnects in conjunction with buffers in routers make them inappropriate. In addition, unlike high-radix interconnects, a binary tree of MoT interconnect (i.e., small number of ports of circuit routers) makes it simple, lean, and lightweight to be implemented.

The contributions of this paper are as follows. First, we propose a new circuit-switched 3-D MoT interconnect that supports sharing TSV bus (i.e., a set of TSVs for address, data, and control signals to a memory bank) among multiple memory banks in a congestion-aware and fault-tolerant manner. As far as we know, this is the first work that considers TSV sharing for 3-D MoT interconnect. In addition, the proposed method allows the number of TSV buses not to be a power of two. When considering the large area occupied by TSVs, putting an arbitrary number of TSVs enables a fine control of TSV overhead to improve the system performance. The proposed method also supports a modular design strategy that allows stacking multiple identical memory dies with the same identical mask and, thus, reduces the fabrication cost. Second, we investigate various parameters for 3-D memory stacking, such as fabrication technology, TSV bonding schemes, number of memory tiers, and TSV sharing structures, that affect the interconnect latency, the system performance, and the fabrication cost. This investigation allows us to find the best 3-D MoT configuration in view of cost efficiency (i.e., performance/cost). For the TSV bonding schemes, we considered the two most widespread and intensively studied technology options: state of the art microbumps with electrostatic discharge (ESD) protection circuits [20]–[22] and high-density Cu–Cu direct bonding [23].

The rest of this paper is organized as follows. Section II discusses the related works. Section III provides details of our target architecture with a plain MoT interconnect. Section IV shows the effects of TSV sharing on a 3-D MoT interconnect. Section V explains the proposed TSV sharing scheme. In Sections VI and VII, experimental setup and results are, respectively, shown, followed by a conclusion in Section VIII.

II. Related Work

General purpose 3-D network-on-chip (NoC), which delivers packets between homogeneous nodes, is one of the well-known interconnection network architectures for a large 3-D SoC. Li *et al.* [24] proposed a hybrid 3-D NoC-bus interconnect. The hybrid interconnect takes advantage of the short vertical interconnections by connecting multiple layers of 2-D mesh network with a bus architecture spanning the entire vertical distance of the chip. Seiculescu *et al.* [28] proposed a design tool to synthesize application-specific 3-D NoC topology, which assigns the network components on to the 3-D tiers and performs a placement of them in each tier, while considering power and latency cost. In [27] and [29], various 3-D NoC topologies have been evaluated and summarized in terms of throughput, latency, and energy dissipation. Several 3-D NoC prototypes have been published [30], [31]. Even though all these works are scalable and provide lots of bandwidth, they are not optimized for low-latency processor to memory traffic, but provide generic intercomputing-node services. For reducing the latency, decompositions of NoC router architecture have been proposed [25], [26], [80], [81]. It breaks up router architecture into a set of smaller components, thereby avoiding the use of a large crossbar (i.e., 7 × 7 crossbar) and reducing the latency within the router itself. However, despite the reduced latency, the large number of hops to be passed makes it inappropriate for on-chip interconnects with tightly coupled cores and memory banks.

Prior studies of 3-D on-chip interconnect for stacked memory architectures can be grouped into three categories: 1) 3-D stacked cache with wide I/O interfaces; 2) 3-D stacked nonuniform cache architecture (NUCA); and 3) 3-D stacked SPM. In [32]–[34], the authors have demonstrated that implementing memory bus between a L2 cache and an on-chip main memory as wide as a cache line that operates at core's clock frequency can provide the maximum bandwidth that the L2 cache can consume and contribute to a larger gain in the system performance. In [35] and [36], 3-D stacked DRAM and SRAM caches with a vertical wide I/O interconnect have been fabricated at 50-nm and 0.18-μm technology nodes, respectively. In [37], a 3-D memory stacked system with 64 ARM Cortex-M3 cores has been fabricated at a 130-nm technology node. It is designed to be expandable to four tiers of core and cache with three tiers of stacked DRAM. Eight DRAM controllers are connected to the cores with a 128-bit bus, providing 2.23 GB/s. However, in spite of all advantages of 3-D stacked caches with wide I/O interfaces, a centralized shared memory still lacks in scalability [38], [39]; on the other hand, NoC-based 3-D stacked NUCA brings a scalable and modular communication infrastructure.

[1] Note that the proposed architecture is only for a limited number of processing cores (equal or <32) because as the number of both cores and memory banks increases, the latency of the combinational interconnect increases (>10 ns, which is out of our scope). Scaling to larger number of processing cores (e.g., >32) requires building a multicluster fabric connected through a scalable NoC [19]. Scaling beyond that will require hierarchical multicluster schemes, and for extremely scaled architectures (1000 cores), hierarchical clustered multihop networks will have to be used.

In 3-D stacked NUCAs, the stacked cache is divided into multiple banks with different access latencies according to their locations to cores. Each core and bank is connected to each other through a mesh interconnect [40]–[42], a tree interconnect [43], or a ring interconnect [44]. Despite the high bandwidth and parallel communications between cores and stacked cache banks with flexible access routines, the inherent large access latency resulting from multiple routers and buffers is not adequate for multicore clusters with shared L2 multibanked SPM. These 3-D stacked NUCAs have higher latency (i.e., order of ten or more clock cycles) than the required performance in our target architecture (i.e., less than five clock cycles). Even though some techniques, such as dynamic data migration [42] and cache partitioning [41], help to reduce the average hop distance for memory data access, they increase hardware (HW) complexity and lead to complex network architectures.

A few works on 3-D stacked SPM with configurable on-chip interconnect have been presented in [45]– [47]. In [45], the authors proposed a configurable memory tier that consists of many uniform memory elements (containing a micromemory-core RAM, an I/O, a configuration register, and a routing switch). Each memory element is connected to each other with a switch-based 3-D mesh interconnect. Instead of using a crossbar switch, AND logic is used in order to reduce the latency. In [46], customizable redistribution layer (RDL) routing was proposed. The RDL, which is a plated metal layer with superior electrical characteristics than typical metal wire, enables connecting each core and memory cell without any switch connection. In [47], a prototype of 3-D stacked SPM has been published. It is a two-tier 3-D IC, where the logic die consists of 64 general purpose processor cores running at 277 MHz, and the memory die contains 256-KB SRAM. Each processor core directly connected to each 4 KB of SRAM. However, despite the extremely low access latency, all these works described above were simply focusing on the use of private memory while our work proposes a solution for sharing L2 memory.

A few works with TSV disconnection in 3-D NoCs have been addressed to enhance the reliability of interconnection network. Such works dealt with this challenge by suggesting redundant TSV links [19], reliable routings [48]–[51], error detection/correction HW modules [52], or TSV-variation-aware synthesis [53]. One approach for reliable routings, which is suitable for both 2-D and 3-D NoC interconnection network, is leveraging reconfigurable routing table to keep fault-tolerant routing paths [48], [49]. This method is highly resilient but suffers from poor scalability due to the area required for the tables. The other approach is applying ZXY routing, but dynamically determining when a packet moves vertically, so that incoming packets never use the faulty TSVs [50], [51]. Those are, therefore, designed to tackle faults on vertical links since fault rate on TSVs is much larger than conventional horizontal ones (i.e., metal wire).

Recently, there have been some works of vertical interconnect serialization as one way to reduce the number of TSVs [9], [54]–[57]. Such serialization schemes reduce the number of TSVs, resulting in more efficient core layout across multiple layers due to the reduced routing congestion and increase the fabrication yield with a small impact on the latency. Pasricha [9] has shown that 4:1 serialization of TSV interconnects saves >70% of TSV footprint with only 1.86% performance degradation on an average at a 65-nm technology node, and proposed a framework for TSV-serialization-aware synthesis of 3-D NoC [58]. However, vertical interconnect serialization is made feasible only in packet-switched 3-D NoC interconnects, where either horizontal propagation delay or vertical propagation delay dominates the performance of interconnect (i.e., network clock frequency) rather than the sum of horizontal and vertical propagation delays. In [59], a configurable serialization scheme of TSV interconnects is proposed to ensure fault-tolerant data transmission.

TABLE I
ZERO-LOAD LATENCY COMPARISONS OF 3-D ON-CHIP INTERCONNECTS

Network Topology	Longest Wire Length (mm)	Delay (ns)	Avg. Hops	Latency (cycle)
3-D Mesh	1.103	0.419	5.0	17.00
3-D Hybrid Bus-Mesh [24]	1.103	0.419	3.5	15.50
3-D Hybrid Bus-Tree [43]	2.206	0.882	2.7	14.66
3-D Flattened Butterfly [74]	3.309	1.391	2.5	13.75
3-D MoT	6.658	3.481	1.0	5.00

In this paper, we propose a new circuit-switched 3-D MoT interconnect for a multicore cluster to connect multiple processing cores, placed on a logic tier, with multiple tiers of multibanked SRAM modules. These SRAM modules constitute a single shared L2 SPM that enables fast communication with the tightly coupled cores for parallel processing. For brief comparisons of the most published 3-D NoC interconnects and our MoT interconnect, Table I shows zero-load latencies of the interconnects for 4 × 4 × 2 3-D NoC architectures. The zero-load latency is the latency where only one packet traverses the network. Although such a latency does not consider contention among packets, it can be used to describe effects of an interconnect topology on the performance [82]. In Table I, the second column shows the length of the longest physical wire in each on-chip interconnect. The longest wire of each interconnect is determined by the longest Manhattan distance between routers connected to each other. The third column shows the resistance–capacitance (RC) delay of the longest wire for each interconnect. The wire delay is estimated using the first-order Elmore model for a 32-nm technology node. The fourth presents the average number of hop counts during a packet traversal. The last column presents the results of zero-load latency. On-chip interconnect assumed to be run on 1 GHz. In 3-D mesh, 3-D hybrid bus-mesh, and 3-D hybrid bus-tree, the latencies of a router and a vertical bus (including arbitration delay and TSV delay) assumed to be, respectively, 2 and 3 cycles [42]. In 3-D flattened butterfly, the router latency assumed to be 3 cycles. The latency to access 64-KB L2 memory bank itself is 1.004 ns [64]. As shown in Table I, compared with our MoT interconnect, the 3-D NoC interconnects have much higher latency that is not appropriate for a multicore cluster, but for a multicluster fabric.

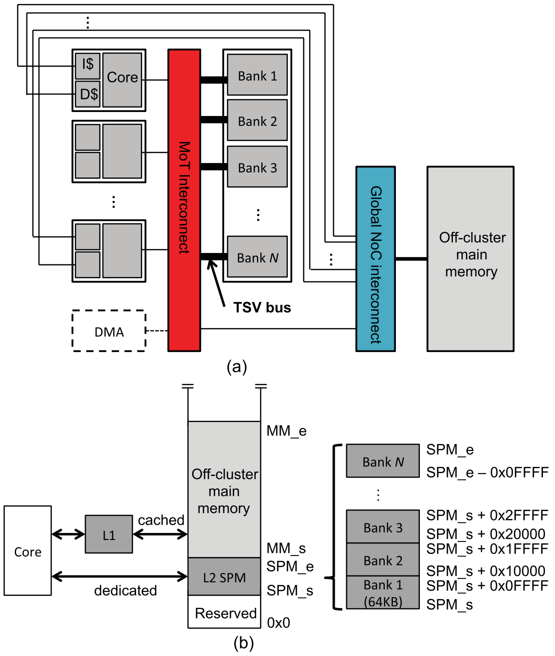


Fig. 1. Memory hierarchy of target 3-D multicore cluster. (a) Block diagram representing the interconnection network between cores and hierarchical memories. (b) Memory map showing a range of global memory addresses allocated to the shared L2 SPM.

III. Target 3-D Multicore Cluster

Fig. 1 shows the memory hierarchy of the target multicore cluster. As shown in Fig. 1(a), the cluster consists of simple cores each of which has its own private L1 instruction and data caches. When designing a multicore cluster using commercial core IPs (e.g., ARM Cortex MPCore), this assumption is reasonable since L1 instruction and data caches are often deeply entrenched in the core subsystem. The multibanked stacked L2 SPM consists of multiple SRAM banks connected with the cores through a MoT interconnect. Each stacked SRAM bank is connected with the MoT interconnect through a TSV bus (i.e., a set of TSVs for address, data, and control signals to a memory bank). Access to an off-cluster large main memory is coordinated by the global NoC interconnect. An optional direct memory access (DMA) engine can be used to carry out data transfers from the off-cluster main memory to the L2 SPM. Fig. 1(b) shows the memory map of our target architecture in view of one core. Note that, as shown in Fig. 1(b), a range of dedicated global addresses is allocated to the shared L2 SPM, while the rest (i.e., off-cluster main memory) is cached by L1 cache. The shared L2 SPM can be used to store the following data: 1) shared local data: maintains variables explicitly defined to be shared at compile time; 2) shared stack: maintains the parameters for passing among cores; and 3) heap: used for dynamically allocated structures.

Fig. 2 shows an example of MoT interconnect consisting of four cores and eight stacked L2 SPM banks. When a core accesses its target memory bank, a combinational path is created through the two kinds of binary trees, i.e., routing tree and arbitration tree. This combinational path is able to support low-latency and nonblocking communication between cores and memory banks [6]. In Fig. 2, during a read/write operation, data and control signals are asserted in the form of packet by a core. This packet is routed through routing switches until it reaches the last level of the routing tree. In order to reach the target memory bank, the packet must be arbitrated among the other simultaneous packets heading for the same memory bank. The round-robin algorithm is used for a starvation-free arbitration. If a request from one core loses the arbitration in the current clock cycle, it is supposed to be granted in the next clock cycle. The arbitration switches first arbitrate the requests following the round-robin policy and, then, route the request in a combinational way.

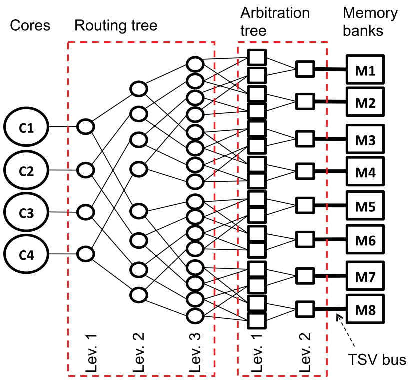


Fig. 2. 4 × 8 MoT interconnect. Empty circles: routing switches. Empty squares: arbitration switches.

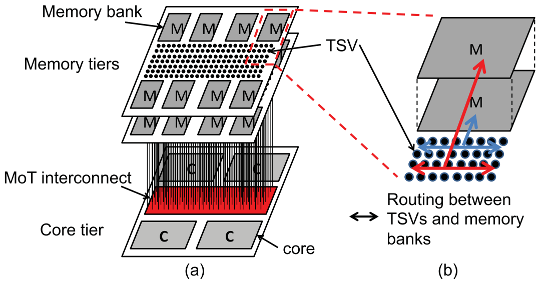


Fig. 3. Geometry view of 3-D multicore cluster with stacked L2 SPM banks. (a) 3-D multicore cluster with TSVs array. (b) TSVs allocation to each stacked memory bank.

Fig. 3 shows a geometry view of our target architecture. As shown in Fig. 3(a), MoT interconnect (i.e., the routing and arbitration switches shown in Fig. 2) is placed in the middle of core tier, which makes it easier that memory access latency from each core is well balanced. Output ports of arbitration switches at the last level of arbitration tree are directly connected to each memory bank through TSVs (also shown in Figs. 1 and 2), which are distributed in the middle of the memory die [60]. Each stacked memory bank is connected to neighboring TSVs as shown in Fig. 3(b). Note that the stacked L2 SPM dies do not need to be the same as the size of the multicore die.

The major advantage of 3-D memory stacking is that the overall wire length of the on-chip interconnect is reduced owing to the reduced memory form factor as well as the additional vertical routing paths. However, this straightforward

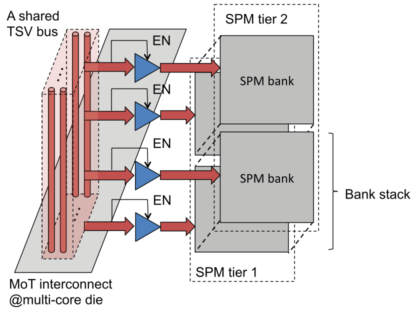


Fig. 4. Sharing a TSV bus among L2 SPM banks that are the closest to each other. The memory banks shared by one TSV bus might be placed in a bank stack or multiple bank stacks, where a bank stack consists of multiple SPM banks, which are directly stacked on each other.

extension of plain MoT to 3-D integration (i.e., per-bank TSVs allocation) requires a considerably larger number of TSVs. When considering the large footprint and low fabrication yield of TSV, it is important to share TSVs among multiple memory banks in order to reduce the number of TSVs with less performance degradation.

IV. Effects of TSV Sharing on 3-D MoT

Fig. 4 shows the most straightforward method for TSV sharing to reduce the number of TSVs while allowing a modular design strategy using the same identical mask for every stacked memory tier [36], [60]. In Fig. 4, a TSV bus is shared by multiple memory banks. The memory banks can be placed in one bank stack[2] or multiple bank stacks so that all the memory banks shared by a TSV bus are the closest to each other. For example, if we assume that two SPM tiers are stacked on a multicore die and four memory banks are shared by one TSV bus, then all the memory banks in every two bank stacks are to be shared by one TSV bus, as shown in Fig. 4. Tristate buffers are inserted between each memory bank and shared TSV bus to make sure that only one memory bank is connected to the MoT interconnect at a time for packet transmission. In this TSV sharing scheme, the total number of TSVs is reduced with respect to the number of memory banks shared by one TSV bus, i.e., N_{share}, which is shown in Table II.

The reduced number of TSVs resulting from TSV sharing makes the fabrication yield of 3-D ICs higher. When assuming a wafer-to-wafer (W2W) bonding technique for the fabrication, the yield of 3-D SPM is estimated as follows [61]:

$$Y_{\text{SPM}} = (Y_{\text{die}})^{N_{\text{tier}}} \cdot (Y_{\text{stacking}})^{N_{\text{tier}}-1} \quad (1)$$

$$Y_{\text{stacking}} = Y_{\text{bonding}} \cdot (1 - f_{\text{tsv}})^{N_{\text{tsv}}} \quad (2)$$

where N_{tier} is the number of SPM tiers stacked on a multicore die, Y_{die} yield of single SPM die, Y_{bonding} yield of 3-D bonding process, f_{tsv} TSV failure rate, and N_{tsv} the total number of TSVs. As shown in (1) and (2), the reduction in the number of TSVs increases the fabrication yield because it decrease N_{tsv} while increasing Y_{die} owing to the reduced area occupied by TSVs.

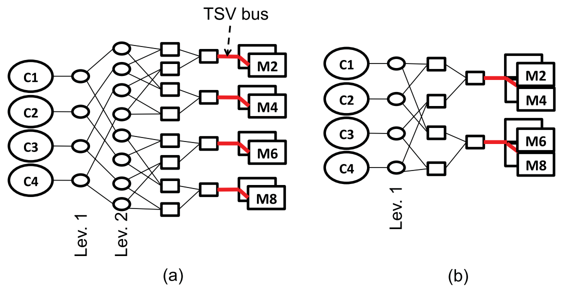


Fig. 5. TSV bus sharing and its effect on the number of routing switch levels in 3-D MoT interconnect. (a) When N_{share} is 2. (b) When N_{share} is 4.

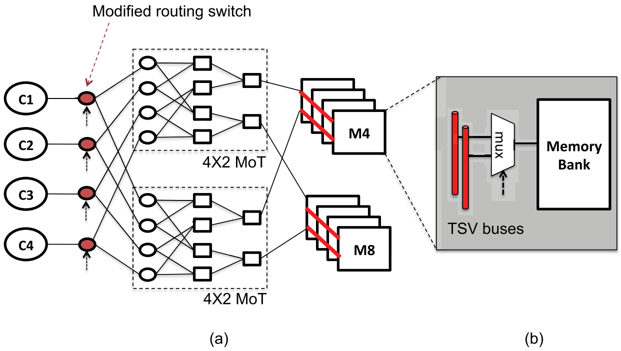


Fig. 6. Example of the proposed TSV sharing method. (a) Two 4 × 2 MoT interconnects with double TSV buses. (b) Memory bank connected with double TSV buses through a multiplexer.

Sharing TSVs also gives a possibility to reduce the MoT interconnect latency. The reduced area occupied by TSVs decreases the wire length in the critical path. In addition, the number of routing switches being passed decreases since the routing tree strongly depends on the number of bank groups, each of which consists of multiple memory banks shared by one TSV bus. Fig. 5 shows two examples of 3-D MoT interconnect when N_{share} is 2 and 4, respectively. As shown in Fig. 5, the number of MoT routing levels and, thus, the number of routing switches being passed decreases as N_{share} increases. Table II shows details about the effects of TSV bus sharing on 3-D MoT interconnects.

Despite all the benefits mentioned above, this straightforward TSV sharing scheme causes traffic collision at shared TSV buses even when each of processing cores accesses different memory banks, if the memory banks accessed by different cores are in the same bank group and accessed at the same time. The amount of collision at shared TSV buses strongly depends on the rate of L2 SPM accesses of applications executed on the multiple cores as well as the number of banks in a bank group, i.e., N_{share}. To control and alleviate the collision at shared TSV buses while keeping all the benefits of TSV sharing, we propose a new TSV sharing method, which is able to balance packet traffics among memory banks.

V. Congestion-Aware TSV Sharing for 3-D MoT

The main idea of the proposed method is to use multiple small 3-D MoT interconnects for multiple paths from cores to

[2] As shown in Fig. 4, a bank stack consists of multiple SPM banks that are directly stacked on each other.

TABLE II

COMPARISONS OF 3-D MOT INTERCONNECTS WITHOUT/WITH TSV BUS SHARING

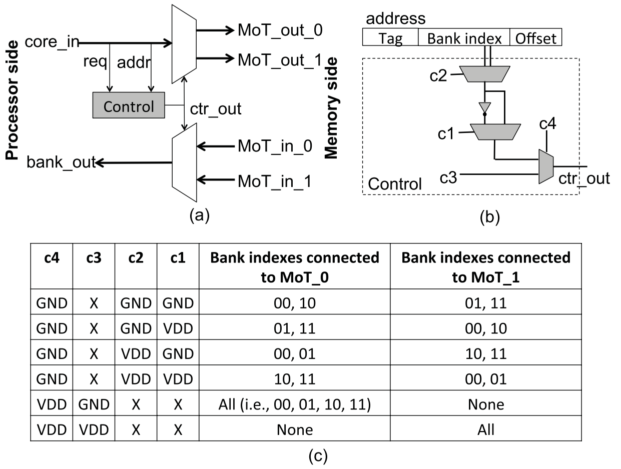
	Without sharing TSV bus	With sharing TSV bus
Number of levels for routing tree	$log_2 N_{bank}$	$log_2(N_{bank}/N_{share})$
Number of levels for arbitration tree	$log_2 N_{core}$	$log_2 N_{core}$
Number of routing switches	$\sum_{i=1}^{log_2 N_{bank}} 2^{i-1} \cdot N_{core}$	$\sum_{i=1}^{log_2(N_{bank}/N_{share})} 2^{i-1} \cdot N_{core}$
Number of arbitration switches	$\sum_{i=1}^{log_2 N_{core}} 2^{i-1} \cdot N_{bank}$	$\sum_{i=1}^{log_2 N_{core}} 2^{i-1} \cdot (N_{bank}/N_{share})$
Number of TSVs	$N_c + 1 + N_{bank} \cdot (Nb_{addr} + Nb_{data})$	$N_c + 1 + (N_{bank}/N_{share}) \cdot (log_2 N_{share} + Nb_{addr} + Nb_{data})$

TABLE III

COMPARISONS BETWEEN CONVENTIONAL TSV SHARING SCHEME AND THE PROPOSED METHOD

	Conventional scheme	Proposed scheme
Type of MoT	4×4 MoT	2 × (4×2) MoT
Ratio of shared banks and TSV buses	2:1	4:2
Required HW	12 routing sw. 12 arbitration sw.	8 routing sw. 12 arbitration sw. 4 modified routing sw. 8 mux
Routing type	Static routing	Dynamic routing

memory banks instead of using one large 3-D MoT interconnect. Fig. 6 shows an example of the proposed TSV sharing method with double 3-D MoT interconnects. In Fig. 6(a), compared with the 3-D MoT interconnect in Fig. 5(a), two 4 × 2 MoT interconnects are used instead of single 4 × 4 MoT interconnect with the same number of TSV buses. Thus, memory banks in a bank group [i.e., four memory banks as shown in Fig. 6(a)] are shared by two TSV buses, which are the same as the number of MoT interconnects. As shown in Fig. 6(b), a multiplexer is inserted between each memory bank and its TSV buses in order to choose one of the two TSV buses for packet communication. To determine a MoT interconnect (and finally its TSV bus) for packet communication, a modified routing switch shown in Fig. 7 is proposed to be added between each core and two MoT interconnects as shown in Fig. 6(a). Fig. 7(a) shows the proposed circuit of the modified routing switch, which is exactly the same circuit as the routing switch within the conventional MoT interconnect except for the control logic. The control logic shown in Fig. 7(b) determines a MoT interconnect (i.e., either MoT_0 or MoT_1) based on the control signals [i.e., c1, c2, c3, and c4 shown in Fig. 7(c)] and bank indexes (i.e., the last two digits in the address of bank index). By setting the control signals to VDD or GND, the routing path (including MoT interconnect and TSV bus) for each bank index is determined as shown in Fig. 7(c). Table III shows comparisons between the conventional TSV sharing scheme [shown in Fig. 5(a)] and the proposed method [shown in Fig. 6(a)]. As shown in Table III, with a negligible HW overhead, the proposed TSV sharing method gives the following three advantages: 1) traffic balancing; 2) TSV fault tolerance; and 3) implementation of unconstrained number of TSV buses, while keeping a modular design strategy using the same identical mask for every stacked memory tiers.



c4	c3	c2	c1	Bank indexes connected to MoT_0	Bank indexes connected to MoT_1
GND	X	GND	GND	00, 10	01, 11
GND	X	GND	VDD	01, 11	00, 10
GND	X	VDD	GND	00, 01	10, 11
GND	X	VDD	VDD	10, 11	00, 01
VDD	GND	X	X	All (i.e., 00, 01, 10, 11)	None
VDD	VDD	X	X	None	All

Fig. 7. Proposed modified routing switch and its control scheme. (a) Modified routing switch circuit. (b) Control logic in the modified routing switch. (c) Memory bank mapping to each MoT interconnect.

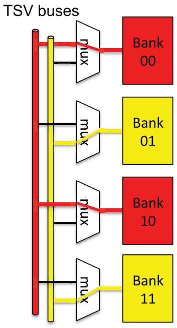
1) *Traffic Balancing:* The logical memory space for an application comprises several memory blocks for data, instructions, heap, and stack. Each memory block has different access frequency (i.e., the number of accesses divided by the total clock cycle counts). Even in the same memory block, each memory segment has quite different access frequencies because different loops and functions are accessed with different frequencies in each memory segment. When multiple applications are loaded onto a multicore system, the different behavior of each application may intensify the disparity of memory access frequency even more. In conclusion, different memory access behavior among memory segments, memory blocks, and applications makes the access frequency of each memory bank quite different. In the proposed TSV sharing method, because of the multiple routing paths from cores to each memory bank, each memory bank can be allocated to one of the multiple TSV buses based on the profiled information so that the traffic to each TSV bus is to be balanced. Fig. 8 shows an example of traffic balancing using the proposed TSV sharing method. Let us assume that a program is executed with four threads, which are run in parallel as shown in Fig. 8(a), and the access frequency of each memory bank is given in Fig. 8(b). As shown in Fig. 8(c), two TSV buses with a multiplexer make the traffic balancing possible

```
parallel program ( )
{
   thread_A ( );  // access memory bank 00
   thread_B ( );  // access memory bank 01
   thread_C ( );  // access memory bank 10
   thread_D ( );  // access memory bank 11
}
```

(a)

Bank index	Access frequency
00	70%
01	25%
10	30%
11	80%

(b)



(c)

Fig. 8. Example of traffic balancing using the proposed TSV sharing scheme. (a) Program with four threads run in parallel. (b) Access frequency of each memory bank resulting from the four threads executed. (c) Connection of two TSV buses with four memory banks in order to balance memory traffic at each TSV bus.

by connecting Bank 00 and Bank 10 to one TSV bus (i.e., left TSV bus) and Bank 01 and Bank 11 to the other TSV bus (i.e., right TSV bus).[3]

2) *TSV Fault Tolerance:* Fabrication and bonding of TSVs can fail, which causes a number of stacked known-good-dies to be discarded and increases the fabrication cost. Even after the fabrication process, wear-out mechanisms, such as a resistance, increase due to electromigration at a TSV may increase the delay at the TSV and eventually lead to an open circuit [62]. As shown in Fig. 6, double TSV buses shared by memory banks in a bank group guarantee a tolerance for single TSV bus failure by rerouting the packet routing path to avoid the faulty TSV bus, as shown in the last two rows of Fig. 7(c).

3) *Unconstrained Number of TSV Buses:* Since the number of memory banks as well as capacity of one memory bank are determined by a range of address bits each of which consists of binary digits, the number of banks must be a power of two. In addition, when using TSV sharing scheme, the number of memory banks shared by one TSV bus must be a power of two, since the destination memory bank is also activated based on the memory address. Thus, in conventional TSV sharing schemes, the number of TSV buses is $N_{\text{bank}}/N_{\text{share}}$, which must be a power of two, since both N_{bank} and N_{share} are numbers of the form 2^n, where n is an integer. When considering large TSV overhead (e.g., large footprint, low yield, and so forth), putting TSV buses with a number of a power of two makes a huge disparities in the view of performance and fabrication cost among TSV sharing schemes with different values of N_{share}. However, the proposed TSV sharing method allows unconstrained number of TSVs insertion using heterogeneous multiple MoT interconnects, which makes it possible to finely control TSV overhead in order to improve the system performance. Fig. 9 shows an example of the proposed TSV sharing method using two heterogeneous MoT interconnects (i.e., 4 × 2 MoT and 4 × 1 MoT). In Fig. 9, each TSV bus from 4 × 2 MoT interconnect (i.e., TSV buses with red color) is shared by four memory banks while a TSV bus from 4 × 1 MoT interconnect (i.e., a TSV bus with blue color) is shared by eight memory banks. In addition, for packet communication, each bank can choose a TSV bus connected to either 4 × 2 MoT interconnect or 4 × 1 MoT interconnect so that the packet traffic is evenly distributed among TSV buses.

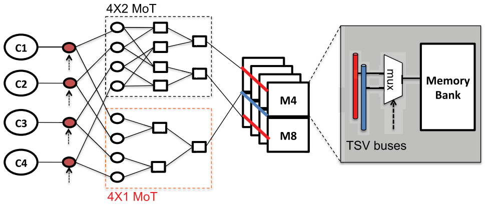


Fig. 9. Heterogeneous 3-D MoT interconnects to support unconstrained number of TSV buses.

4) *Discussion:* In our 3-D MoT, the control signals need to be dynamically set based on the profiled information in order to reduce memory traffic congestion and/or to avoid faulty TSVs. For that, a HW monitor and a software (SW) algorithm need to be implemented. Since most commercial processors already employ HW performance monitors and error-collection code, they can trace both memory access frequency and faulty TSVs with negligible overhead. We assume that the SW algorithm used for traffic balancing is also implemented in processing cores. Note that the complexity of the traffic-balancing algorithm does not increase as the numbers of cores and banks increase, but depends on the ratio of shared memory banks and TSV buses. For example, if we consider four shared memory banks with two TSV buses (as shown in Fig. 8), what the traffic balancing would do is grouping the memory banks into two groups (same as the number of TSV buses) based on the profiled information (i.e., memory bank access frequency) and mapping each group of the memory banks to each TSV bus in order to evenly distribute memory traffic on TSV buses. After determining the mapping, the control signals of 3-D MoT are automatically determined due to the physical connections between the 3-D MoT and TSV buses. For the mapping, the memory access frequency of each memory bank is sorted. Then, the banks with the highest and the lowest memory access frequencies are connected to one TSV bus and the rest of banks are connected to the other TSV bus. This traffic balancing method is not optimal, but simple and effective enough as shown in the experimental results in Section VII.

VI. Experimental Setup

We performed experiments using a 3-D multicore cluster with a multibanked shared L2 SPM stacked on top of the multicore die. The 32 processing cores are integrated in the

[3] The configuration is made by the control signals such that c1, c2, and c4 are GND, GND, and GND, respectively, as shown in the second row of Fig. 7(c).

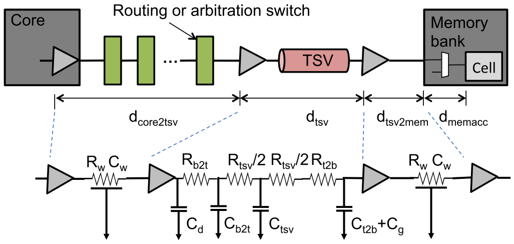


Fig. 10. Latency estimation of 3-D MoT interconnect. Elmore distributed *RC* delay model to estimate delay from a core to a SPM bank.

multicore cluster and each core is considered to be ARM Cortex-A5 with 16-/16-KB instruction and data caches [63]. The operating core clock frequency is assumed to be 1 GHz. The L2 stacked SPM consists of 64 SRAM banks. Each memory bank has a capacity of 64 KB. The size of a memory bank and the propagation delay from memory bank IO to memory core cell within a memory bank, i.e., d_{memacc} shown in Fig. 10, are estimated from CACTI [64]. The number of stacked SPM tiers used in the experiments varies from 1 to 8, which means the number of memory banks per memory tier varies from 64 to 8 (= 64/8). For TSV bonding, we tested our solutions with two different bonding techniques: 1) microbumps [20]; and 2) Cu–Cu direct bonding [23]. For the microbumps, a minimum pitch of 40 μm × 50 μm, and for the direct bonding a more dense pitch of 10 μm × 10 μm are assumed, respectively. The interdie signal interfaces in 3-D ICs are vulnerable to electrical stress induced during stacking or testing steps. To cope with these issues, I/O interconnects passing through TSVs will be protected by ESD protection circuits [22]. As an optimistic corner case, the ESD protection circuits are not considered in the direct bonding technique.

In order to estimate the latency of MoT interconnect, the delay for the longest possible link between cores and memory banks is estimated using Elmore distributed *RC* delay model [23], [65], since on-chip global signal wires are highly resistive while the inductance is negligible. Thus, in a planar 2-D system, the delay between two diagonal corners is considered, while in a 3-D IC, the delay from a corner on the bottom chip to a diagonally opposite corner on the top die is considered. Fig. 10 shows how a delay from a core to a stacked SPM bank is composed [i.e., delay for a global metal wire in core tier (d_{core2tsv}), a TSV (d_{tsv}), a global metal wire in memory tier (d_{tsv2mem}), and a memory bank itself (d_{memacc})]. In Fig. 10, size of each buffer has been determined so that the delays of the on-chip metal wires and TSV are to be minimal. The delay of routing and arbitration switches is assumed to be four times as long as the one of a minimum sized buffer. A parasitic capacitance of 35 fF and a resistance of 18 mΩ are used to model a TSV [23]. For a microbump [20] and an ESD protection circuit [22], capacitive loads of 10 and 20 fF are assumed, respectively. For an on-chip metal wire and a buffer, all the parasitic capacitances and resistances data at a 65-nm technology node are obtained from [66] and scaled to each fabrication technology node used in the experiments (i.e., 65-, 45-, and 32-nm technology nodes). The method of using Elmore delay model is suitable to estimate interconnect performance at the early stages of design flow where all the logics are not synthesized yet, because the design tools for the TSV and the 3-D technology we used are not publicly available. The same method was already used and verified in [66] and [75] for estimating interconnect performance of 3-D ICs. Especially in [75], delay estimation using the first-order Elmore model for 3-D cache memory has been validated by the Cadence Spectre [76] simulation of a four-way 18-Mb Intel SRAM cache at the 180-nm technology node, achieving an accuracy within 10% of the Cadence simulation result. To estimate interconnect power, we used analytical models proposed in [77]. To calibrate the leakage power model for a 32-nm technology node, we used McPAT [78], which is a simulator for timing, area, and dynamic, short-circuit, and leakage power of multicore systems, including interconnects. At a nominal temperature of 300 K, the leakage power for interconnect is 5% of total power. A similar calibration approach has been introduced in [79]. In this paper, we did not consider the routing of interconnects on the 3-D die. Because the routing congestion of interconnects on the 3-D die increases with the number of TSVs [10], the increased sharing of TSV buses reduces the routing congestion and, thus, decreases the packet latency. Thus, ignoring the effect of the routing congestion on TSV sharing means that the experimental results of our solution represent at-least performance improvement.

TABLE IV

ARCHITECTURE CONFIGURATIONS FOR GRAPHITE SIMULATOR

Feature	Discription
Core	1GHz, 32 cores, in-order execution
L1 inst. and data cache	Private, 16KB (per-core), 4-way associative, LRU replacement
L2 SPM	Shared, 64 banks, 64KB (per bank)
Off-cluster DRAM	One controller, 100ns latency

TABLE V

BENCHMARK PROGRAMS IN THREE TEST PROGRAM SUITS

Program suit	SPM Utilization	Programs
Low	< 0.01	water-spatial, fmm, ocean-non-contiguous, scan
Mid	0.01 - 0.1	2d-jacobi, lu-contiguous, radix, water-nsquared
High	0.1 - 0.9	fft, lu-non-contiguous

For the performance evaluation of 3-D multicore cluster, we employed graphite [67], which is an open-source parallel multicore simulator and 64 SPM banks with multiple TSV buses are added into the architectural model of graphite simulator. Table IV shows the details of configuration in graphite simulator. For simulation benchmarks, SPLASH-2 benchmark suite [68], 2-D Jacobi [69], and scan [70] were used, all of which are appropriate to parallel machines with shared memory. The benchmark programs are classified into three test program suits with different SPM utilization ratio (i.e., number of SPM accesses per memory instruction) as shown in Table V.

For the fabrication cost estimation, we assumed that W2W and face-to-back 3-D bonding are performed. W2W bonding does not need any test before bonding and it is easy for

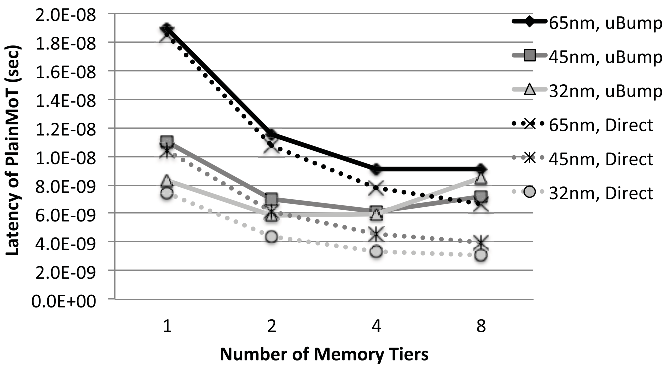


Fig. 11. Results of MoT interconnect latency for PlainMoT with respect to the number of stacked L2 SPM tiers, TSV bonding techniques, and fabrication technology nodes.

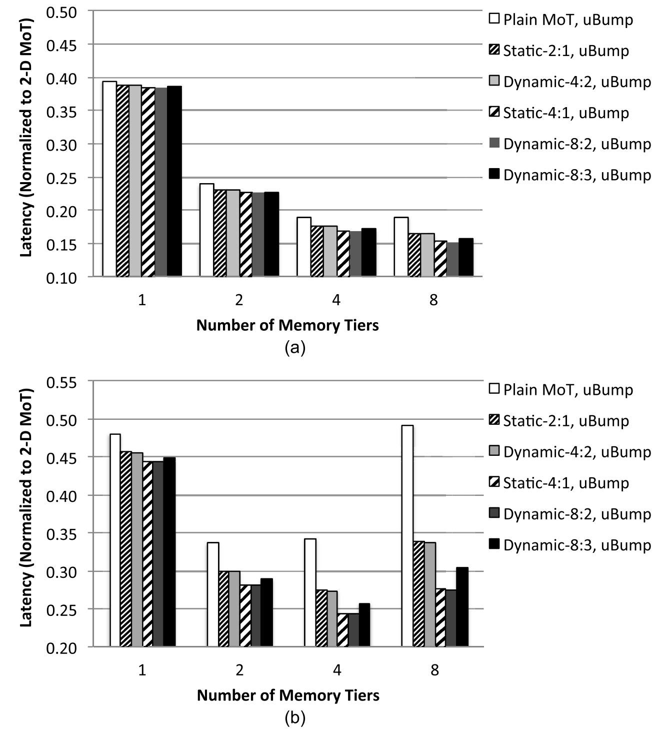


Fig. 12. Results of MoT interconnect latency with respect to the number of stacked L2 SPM tiers and TSV bonding techniques. (a) When 65-nm technology node is used. (b) When 32-nm technology node is used. All the values are normalized with respect to the latency results of 2-D MoT (i.e., 48.05 ns for 65-nm technology node and 17.31 ns for 32-nm technology node).

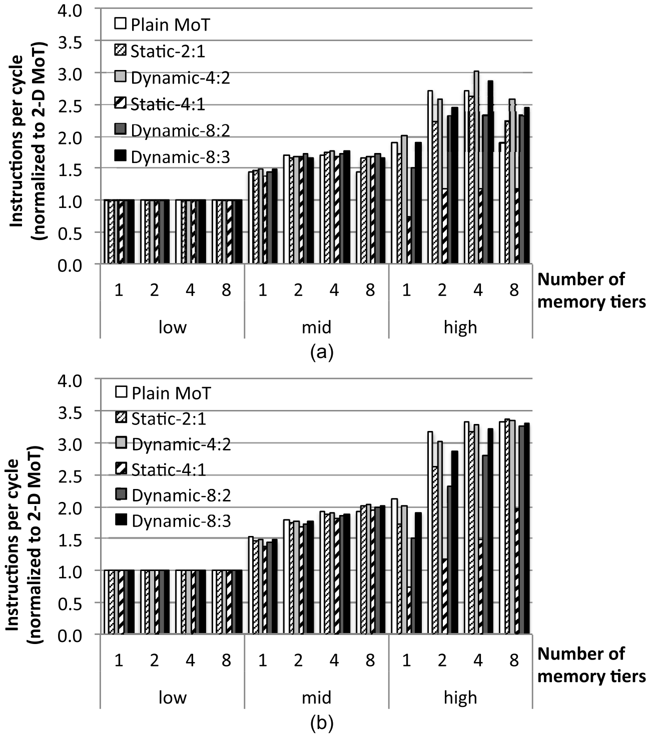


Fig. 13. Results of IPC for the three test program suits (i.e., low, mid, and high) with respect to the number of memory tiers in 32-nm technology node. (a) When microbump TSV bonding is used. (b) When direct TSV bonding is used. All the values are normalized with respect to the IPC results of 2-D MoT. The IPC results of 2-D MoT are, respectively, 5.524, 3.085, and 2.322 for low, mid, and high test program suite.

die alignments with higher throughput, at the expense of yield loss. To estimate the fabrication cost of L2 SPM, we used analytical models proposed in [66] and [71], which are presented as follows:

$$C_{\text{SPM}} = \frac{N_{\text{tier}} \cdot C_{\text{die}} + (N_{\text{tier}} - 1) \cdot C_{\text{stacking}}}{Y_{\text{SPM}}} \tag{3}$$

where C_{die} $(= C_{\text{wafer}}/N_{\text{die}})$ is a fabrication cost for a memory die and C_{stacking} $(= C_{\text{tsv}} \cdot N_{\text{tsv}})$ is a fabrication cost to stack one memory tier. The yield of L2 SPM, i.e., Y_{SPM}, is presented in (1). All the parameters related to the cost estimation are presented in [66]. We assumed that Y_{bonding} and f_{tsv} in (2) are 0.98 and 1E-06, respectively. Note that the absolute values of fabrication yield and cost from the analytical models are not accurate as much as those from the real fabrication, since they depend on many circumstances such as fabrication foundry, market demand, and so forth. However, the analysis of relative values for each TSV sharing scheme based on the analytical models helps to choose the best scheme of TSV sharing for the target system at the early stages of design flow. This paper focuses only on evaluating the effect of TSV footprint on the chip fabrication yield. Thus, some parameters, such as TSV density (i.e., the number of TSVs per unit area), that may impact on TSV yield itself are not considered, and the TSV yield is modeled as a constant. Precise estimation of TSV yield will be our future work.

VII. Experimental Results

We performed the experiment using three different versions of the proposed methods (Dynamic-4:2, Dynamic-8:2, and Dynamic-8:3) and four conventional ones (2-D MoT, PlainMoT, Static-2:1, and Static-4:1) as follows.

1) *2-D MoT:* All the cores and L2 SPM banks are placed on a 2-D planar structure.
2) *PlainMoT:* Multiple L2 SPM tiers are stacked on the multicore die and all the memory banks are connected to the cores through a plain MoT interconnect (presented in Section III), where TSV sharing scheme is not applied.
3) *Static-2:1 (Static-4:1):* Multiple L2 SPM tiers are stacked on the multicore die and all the memory banks are connected to the cores through single 3-D MoT

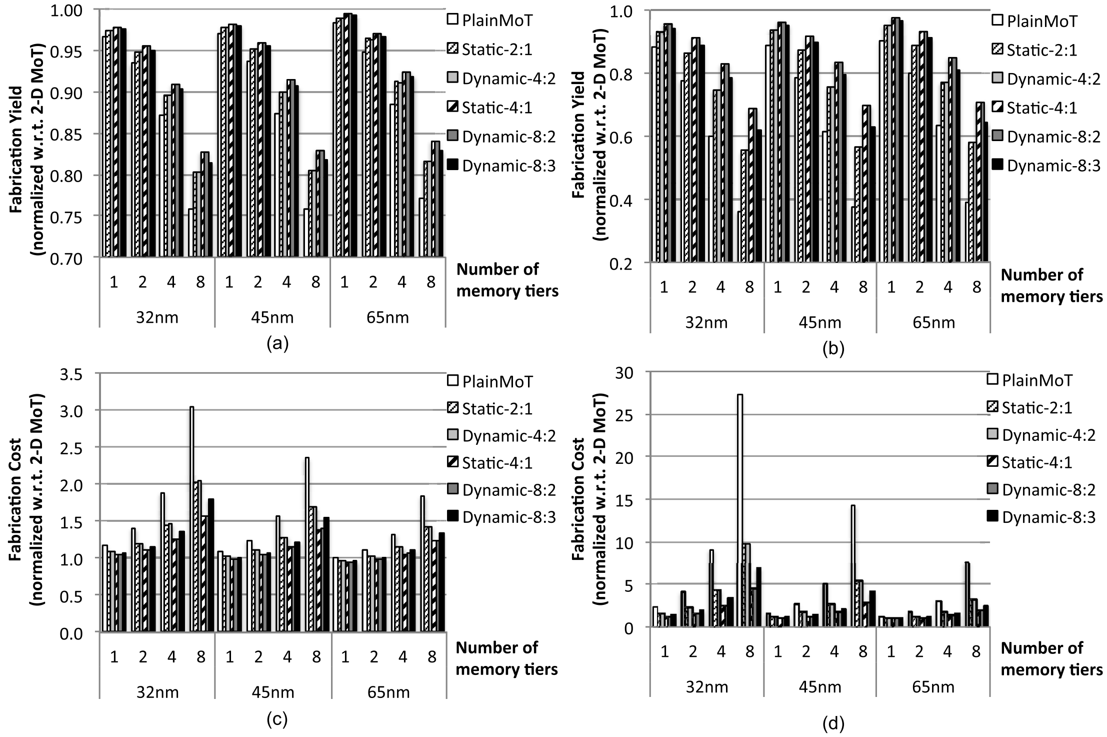


Fig. 14. Results of fabrication yield and cost of a stacked L2 SPM with respect to the number of stacked SPM tiers and fabrication technology nodes. (a) Fabrication yield when direct TSV bonding is used. (b) Fabrication yield when microbump TSV bonding is used. (c) Fabrication cost when direct TSV bonding is used. (d) Fabrication cost when microbump TSV bonding is used. All the values are normalized with respect to the results of 2-D MoT.

interconnect with static TSV sharing scheme (presented in Section IV), where the ratio of shared memory banks and TSV buses is 2:1 (4:1).

4) *Dynamic-4:2 (Dynamic-8:2 or Dynamic-8:3):* Multiple L2 SPM tiers are stacked on the multicore die and all the memory banks are connected to the cores through double 3-D MoT interconnects with the proposed TSV sharing scheme (presented in Section V), where the ratio of shared memory banks and TSV buses is 4:2 (8:2 or 8:3).

Our first experiment results show the impact of changing the number of stacked memory tiers on the latency of MoT interconnect. When moving from a 2-D planar structure to two or more stacked structures, we notice a decrease in the form factor, which reduces the interconnect wire delay. Fig. 11 shows the latency of 3-D MoT interconnect with respect to the number of L2 SPM tiers, i.e., N_{tier}, in PlainMoT with different TSV bonding techniques and fabrication technology nodes. It can be seen that for direct TSV bonding technique in a 65-nm technology node, the MoT interconnect latency decreases as the number of stacked memory tiers increases, as we expected. However, the reduction of the latency begins to saturate as the fabrication technology improves (e.g., in a 32-nm technology node) because shrinking logic device with the fabrication technology makes the area occupied by TSVs getting dominant. For microbump (with ESD protection circuits) bonding technique, the decrease of the MoT interconnect latency starts saturating even in a 65-nm technology node because TSVs with microbumps occupy significant silicon area. Such large TSVs not only have the stacked memory banks spread out so that the critical-path distance between cores and memory banks does not decrease as much as expected, but also make the TSV placing and routing harder. Thus, in 45- and 32-nm technology nodes, the MoT interconnect latency even increases when the numbers of stacked memory tiers are more than four and two, respectively.

Our second set of experimental results show the impact of TSV sharing on the latency of MoT interconnect. Sharing TSV buses among multiple memory banks can reduce the area occupied by TSVs as well as the number of routing-tree levels of MoT interconnect, which finally reduces the latency. Fig. 12 shows the MoT interconnect latency of all our candidates with respect to the number of L2 SPM tiers for microbump bonding technique in 65- and 32-nm technology nodes. In a 65-nm technology node [Fig. 12(a)], it can be seen that the latency improvement owing to TSV sharing schemes, compared with PlainMoT, increases with the number of stacked memory tiers because the area occupied by TSVs is getting dominant as the size of memory die decreases resulting from vertical stacking. As shown in Fig. 12(a), the maximum latency improvement owing to TSV sharing (i.e., the maximum difference among the results of all the candidates) is ~20% (e.g., Dynamic-8:2 when the number of tiers is eight). Note that the latency results of Static-2:1 and Static-4:1 are, respectively, very close with the ones of Dynamic-4:2 and Dynamic-8:2 because almost the same number of TSVs is used in each case (i.e., the case of Static-2:1 and Dynamic-4:2 and the case of Static-4:1 and Dynamic-8:2). The MoT interconnect latency of Dynamic-8:3 is always placed between the ones of Dynamic-4:2 and Dynamic-8:2. In Fig. 12(b), the disparity of the latency results between PlainMoT and the other TSV sharing schemes is

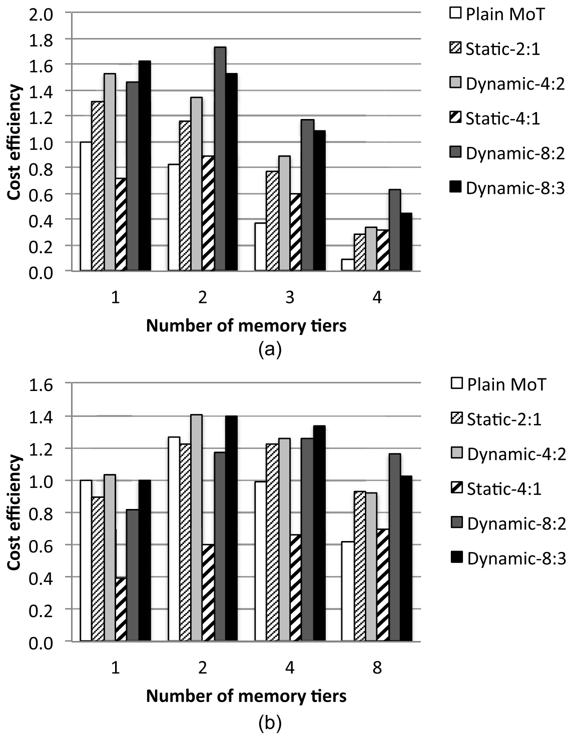


Fig. 15. Results of cost efficiency (i.e., IPC divided by fabrication cost) for high test program suit with respect to the number of stacked memory tiers in 32-nm technology node. (a) When microbump TSV bonding is used. (b) When direct TSV bonding is used. All the values are normalized with respect to the cost efficiency of PlainMoT when the number of memory tiers is one.

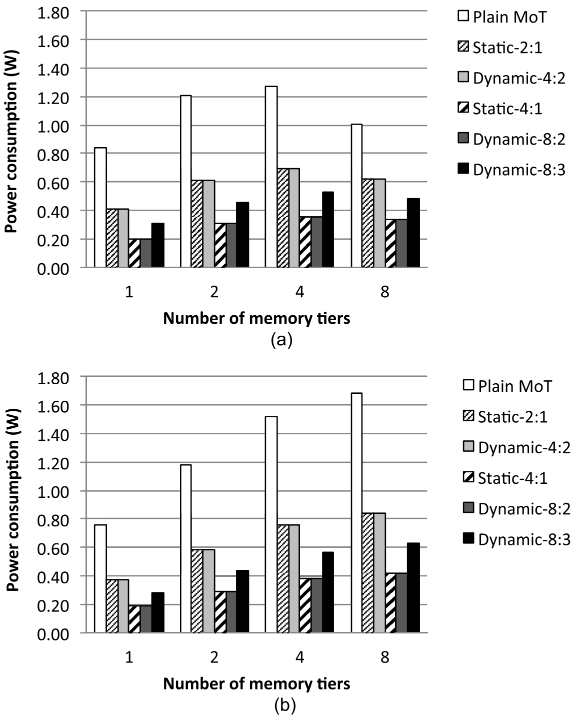


Fig. 16. Results of power consumption of MoT interconnect with respect to the number of stacked L2 SPM tiers. (a) When microbump TSV bonding is used. (b) When direct TSV bonding is used.

much bigger due to the smaller logic area with the advanced fabrication technology, i.e., 32 nm. Compared with PlainMoT, Dynamic-4:2, Dynamic-8:2, and Dynamic-8:3 yield up to 31%, 44%, and 38% latency improvements, respectively, when the number of tiers is eight.

The next experiment results show the effect of each TSV sharing scheme on the system performance, in terms of IPC. As mentioned above, a TSV sharing scheme helps to decrease the MoT interconnect latency. However, packet contention occurring at shared TSV buses may degrade the system performance despite of the reduced latency. Fig. 13 shows the IPC results of multicore clusters with respect to the number of L2 SPM tiers for the three test program suits (shown in Table V) when a 32-nm technology node with either microbump TSV bonding technique or direct TSV bonding technique is used. All the values are normalized with respect to the IPC results of 2-D MoT. As shown in Fig. 13(a) and (b), for low test program suit, all the candidates have similar IPC results because SPM utilization ratios of the applications in low test program suit are too low and, thus, difference in the MoT interconnect latencies among the candidates does not affect the system performance. However, for mid and high test program suits, it can be seen that the system performance increases as the SPM utilization increases, compared with IPC results of 2-D MoT. For microbump TSV bonding technique shown in Fig. 13(a), the IPC of PlainMoT decreases as the number of memory tiers is more than four because of an increase in the MoT interconnect latency (as shown in Fig. 11). Conventional TSV sharing schemes (i.e., Static-2:1 and Static-4:1) degrade the IPC results due to the packet congestion occurring at shared TSV buses while the proposed methods (i.e., Dynamic-4:2, Dynamic-8:2, and Dynamic-8:3) yield the best performance results when the number of memory tiers is more than four because congestion-aware TSV sharing reduces both the MoT interconnect latency and the congestion at shared TSV buses. Note that, in both TSV sharing schemes (i.e., the conventional ones and the proposed ones), the performance results degrade as the number of memory banks shared by one TSV bus increases due to the increase in packet congestion at shared TSV buses. As shown in Fig. 13(a), for high test program suits, when the number of memory tiers is eight, Dynamic-4:2 yields up to 14% and 27% IPC improvements compared with Static-2:1 and PlainMoT, respectively, and Dynamic-8:2 up to 49% and 18% IPC improvements compared with Static-4:1 and PlainMoT, respectively. For direct TSV bonding technique shown in Fig. 13(b), PlainMoT almost yields the best performance results except when the number of memory tiers is eight, because the dense TSV footprint in direct bonding technique allows the MoT interconnect latency to decrease continuously with respect to the number of stacked memory tiers without packet congestion, while TSV sharing schemes do not to reduce the latency enough but cause packet congestion at shared TSVs. However, when the number of memory tiers is eight, the latency improvement of PlainMoT is saturated (as shown in Fig. 11) and, thus, TSV sharing with proper packet congestion management schemes help to reduce the latency more and increase the system performance.

The next experimental results show the impact of TSV sharing on the fabrication yield and cost of 3-D stacked SPM. Because of high defect rates of 3-D die stacking processes with

TSVs, the reduced number of TSVs owing to TSV sharing increases the fabrication yield and reduces the cost. Fig. 14 shows the fabrication yield and cost with respect to the number of L2 SPM tiers and fabrication technology nodes for different TSV bonding techniques. As shown in Fig. 14(a) and (b), the fabrication yield increases as more TSVs are shared. However, in microbump TSV bonding technique, the disparity of fabrication yield with respect to the TSV sharing schemes (i.e., the number of shared TSVs) is much higher due to the large footprint of TSV. The disparity of fabrication yield also increases with respect to the number of SPM tiers because of the reduction in memory die footprint, which makes the area occupied by TSVs larger. The disparity of fabrication yields makes a big difference of fabrication costs as shown in Fig. 14(c) and (d). In a 32-nm technology node, Dynamic-8:2 yields up to 48% and 83% fabrication cost reductions compared with PlainMoT with direct TSV bonding and microbump TSV bonding, respectively. Note that the results of Static-2:1 and Static-4:1 are, respectively, almost the same as the ones of Dynamic-4:2 and Dynamic-8:2 due to almost the same number of TSVs used in each case.

The next experiment compares the TSV sharing schemes in terms of cost efficiency, i.e., performance/fabrication cost. Fig. 15 shows the results of cost efficiency with respect to the number of SPM tiers in a 32-nm technology node for high test program suit with different TSV bonding techniques. All the values are normalized with respect to the cost efficiency of PlainMoT when the number of memory tiers is one. For microbump TSV bonding technique shown in Fig. 15(a), PlainMoT yields almost the worst cost efficiency because many TSVs with large footprint have negative effects on 3-D MoT in the view of performance and fabrication cost as shown in Figs. 13 and 14, respectively. In Fig. 15(a), the maximum cost efficiency is achieved by Dynamic-8:2, which yields up to ×2.11 and ×1.96 improvements compared with PlainMoT and Static-4:1, respectively, when the number of memory tiers is two. For direct TSV bonding technique shown in Fig. 15(b), the maximum cost efficiency is achieved by Dynamic-4:2 when the number of memory tiers is two, and it yields up to ×1.11 and ×1.15 improvement compared with PlainMoT and Static-2:1, respectively. Static-4:1 almost yields the worst cost efficiency because of the large amount of congestion at shared TSV buses.

Fig. 16 shows the results of power consumption of MoT interconnect with respect to the number of stacked SPM tiers in a 32-nm technology node, which shows the impact of TSV sharing on the power consumption. As shown in Fig. 16(a) and (b), the power consumption of MoT interconnect decreases as more TSVs are shared due to the reduction in the number of interconnects (e.g., wires and TSVs) and the corresponding logics (e.g., routing and arbitration switches, buffers, and so forth). When the number of SPM tiers is one, power consumption of microbump TSV bonding is slightly larger than that of direct TSV bonding due to the additional capacitance of microbump and ESD circuits. However, compared with power consumption of microbump TSV bonding, power consumption of direct TSV bonding is larger as the number of SPM tiers goes higher because of the larger reduction in latency of MoT interconnect, which, in turns, increases the clock frequency in the MoT interconnect.

VIII. Conclusion

In this paper, we presented a new TSV sharing method for a cost-effective design of 3-D MoT interconnect that can be integrated in a multicore cluster where a 3-D multibanked shared L2 SPM is stacked on the multicore die. The proposed TSV sharing method gives better performance by supporting traffic balancing, TSV fault tolerance, and unconstrained number of TSVs insertion. Furthermore, the proposed method keeps a modular design strategy, which allows users to stack multiple memory dies with identical dies, without the need for different masks for dies at different levels in the stack. We also investigated architecture parameters of 3-D stacked memory (e.g., fabrication technology, TSV bonding technique, number of memory tiers, and TSV sharing scheme) in terms of latency, system performance, and fabrication cost. Compared with a plain TSV sharing scheme (i.e., PlainMoT), Dynamic-8:2 and Dynamic-4:2 offer up to ×2.11 and ×1.11 improvements for microbump TSV bonding and direct TSV bonding techniques, respectively. Compared with static TSV sharing schemes (i.e., Static-4:1 and Static-2:1), Dynamic-8:2 and Dynamic-4:2, respectively, offer up to ×1.96 and ×1.15 improvements for microbump TSV bonding and direct TSV bonding techniques.

References

[1] NVIDIA. *The Next Generation CUDA Architecture, Code Named Fermi*. [Online]. Availabe: http://www.nvidia.com/object/fermi_architecture.htm, accessed Aug. 21, 2014.

[2] *The Hypercore Architecture*, Plurality Ltd., San Jose, CA, USA, Jan. 2010.

[3] D. Melpignano *et al.*, "Platform 2012, a many-core computing accelerator for embedded SoCs: Performance evaluation of visual analytics applications," in *Proc. 49th Annu. DAC*, 2012, pp. 1137–1142.

[4] R. Banakar, S. Steinke, B.-S. Lee, M. Balakrishnan, and P. Marwedel, "Scratchpad memory: A design alternative for cache on-chip memory in embedded systems," in *Proc. 10th Int. Symp. Hardw./Softw. Codesign (CODES)*, 2002, pp. 73–78.

[5] P. Marwedel, *Embedded System Design: Embedded Systems Foundations of Cyber-Physical Systems*, 2nd ed. New York, NY, USA: Springer-Verlag, Dec. 2010.

[6] A. Rahimi, I. Loi, M. R. Kakoee, and L. Benini, "A fully-synthesizable single-cycle interconnection network for shared-L1 processor clusters," in *Proc. DATE*, Mar. 2011, pp. 1–6.

[7] G. H. Loh, "3D-stacked memory architectures for multi-core processors," in *Proc. 35th ISCA*, Jun. 2008, pp. 453–464.

[8] J. W. Joyner, P. Zarkesh-Ha, and J. D. Meindl, "A stochastic global net-length distribution for a three-dimensional system-on-a-chip (3D-SoC)," in *Proc. 14th Annu. IEEE Int. ASIC/SOC Conf.*, Sep. 2001, pp. 147–151.

[9] S. Pasricha, "Exploring serial vertical interconnects for 3D ICs," in *Proc. 46th ACM/IEEE DAC*, Jul. 2009, pp. 581–586.

[10] K.-R. Dai, W.-H. Liu, and Y.-L. Li, "NCTU-GR: Efficient simulated evolution-based rerouting and congestion-relaxed layer assignment on 3-D global routing," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 20, no. 3, pp. 459–472, Mar. 2012.

[11] M.-C. Tsai, T.-C. Wang, and T. Hwang, "Through-silicon via planning in 3-D floorplanning," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 19, no. 8, pp. 1448–1457, Aug. 2011.

[12] D. H. Kim, K. Athikulwongse, and S. K. Lim, "A study of through-silicon-via impact on the 3D stacked IC layout," in *Proc. IEEE/ACM ICCAD*, Nov. 2009, pp. 674–680.

[13] J. Cong, G. Luo, J. Wei, and Y. Zhang, "Thermal-aware 3D IC placement via transformation," in *Proc. ASP-DAC*, Jan. 2007, pp. 780–785.

[14] W.-K. Mak and C. Chu, "Rethinking the wirelength benefit of 3-D integration," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 20, no. 12, pp. 2346–2351, Dec. 2012.

[15] A.-C. Hsieh and T. Hwang, "TSV redundancy: Architecture and design issues in 3-D IC," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 20, no. 4, pp. 711–722, Apr. 2012.

[16] Texas Instruments. (2011). *OMAP 5 Platform*. [Online]. Available: http://www.ti.com/ww/en/omap/omap5/omap5-OMAP5430.html

[17] ST-Ericcson. (2012). *NovaThor Platform*. [Online]. Available: http://www.stericsson.com/products/L9540-novathor.jsp

[18] QUALCOMM. (2012). *Snapdragon S4 Processors*. [Online]. Available: http://www.qualcomm.com/snapdragon/processors/s4

[19] I. Loi, F. Angiolini, S. Fujita, S. Mitra, and L. Benini, "Characterization and implementation of fault-tolerant vertical links for 3-D networks-on-chip," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 30, no. 1, pp. 124–134, Jan. 2011.

[20] E. J. Marinissen *et al.*, "Wafer probing on fine-pitch micro-bumps for 2.5D- and 3D-SICs," in *Proc. Southwest Test Workshop*, San Diego, CA, USA, 2011.

[21] X. Wu *et al.*, "Electrical characterization for intertier connections and timing analysis for 3-D ICs," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 20, no. 1, pp. 186–191, Jan. 2012.

[22] E. Rosenbaum, V. Shukla, and M.-S. Keel, "ESD protection networks for 3D integrated circuits," in *Proc. IEEE Int. 3DIC*, Jan./Feb. 2011, pp. 1–7.

[23] G. Katti, M. Stucchi, K. de Meyer, and W. Dehaene, "Electrical modeling and characterization of through silicon via for three-dimensional ICs," *IEEE Trans. Electron Devices*, vol. 57, no. 1, pp. 256–262, Jan. 2010.

[24] F. Li, C. Nicopoulos, T. Richardson, Y. Xie, V. Narayanan, and M. Kandemir, "Design and management of 3D chip multiprocessors using network-in-memory," in *Proc. 33rd ISCA*, 2006, pp. 130–141.

[25] J. Kim *et al.*, "A novel dimensionally-decomposed router for on-chip communication in 3D architectures," in *Proc. 34th Annu. ISCA*, 2007, pp. 138–149.

[26] D. Park *et al.*, "MIRA: A multi-layered on-chip interconnect router architecture," in *Proc. 35th ISCA*, Jun. 2008, pp. 251–261.

[27] V. F. Pavlidis and E. G. Friedman, "3-D topologies for networks-on-chip," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 15, no. 10, pp. 1081–1090, Oct. 2007.

[28] C. Seiculescu, S. Murali, L. Benini, and G. De Micheli, "SunFloor 3D: A tool for networks on chip topology synthesis for 3-D systems on chips," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 29, no. 12, pp. 1987–2000, Dec. 2010.

[29] B. S. Feero and P. P. Pande, "Networks-on-chip in a three-dimensional environment: A performance evaluation," *IEEE Trans. Comput.*, vol. 58, no. 1, pp. 32–45, Jan. 2009.

[30] I. Loi, P. Marchal, A. Pullini, and L. Benini, "3D NoCs—Unifying inter & intra chip communication," in *Proc. IEEE ISCAS*, May/Jun. 2010, pp. 3337–3340.

[31] M. H. Jabbar, D. Houzet, and O. Hammami, "3D multiprocessor with 3D NoC architecture based on Tezzaron technology," in *Proc. IEEE 3DIC*, Jan./Feb. 2011, pp. 1–5.

[32] G. L. Loi, B. Agrawal, N. Srivastava, S.-C. Lin, T. Sherwood, and K. Banerjee, "A thermally-aware performance analysis of vertically integrated (3-D) processor-memory hierarchy," in *Proc. 43rd ACM/IEEE DAC*, Jan. 2006, pp. 991–996.

[33] B. Black *et al.*, "Die stacking (3D) microarchitecture," in *Proc. 39th Annu. IEEE/ACM Int. Symp. MICRO*, Dec. 2006, pp. 469–479.

[34] D. H. Woo, N. H. Seong, D. L. Lewis, and H.-H. S. Lee, "An optimized 3D-stacked memory architecture by exploiting excessive, high-density TSV bandwidth," in *Proc. IEEE 16th Int. Symp. HPCA*, Jan. 2010, pp. 1–12.

[35] J.-S. Kim *et al.*, "A 1.2 V 12.8 GB/s 2 Gb mobile wide-I/O DRAM with 4×128 I/Os using TSV-based stacking," in *Proc. IEEE ISSCC*, Feb. 2011, pp. 496–498.

[36] A. Zia, P. Jacob, J.-W. Kim, M. Chu, R. P. Kraft, and J. F. McDonald, "A 3-D cache with ultra-wide data bus for 3-D processor-memory integration," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 18, no. 6, pp. 967–977, Jun. 2010.

[37] D. Fick *et al.*, "Centip3De: A 3930DMIPS/W configurable near-threshold 3D stacked system with 64 ARM Cortex-M3 cores," in *Proc. IEEE ISSCC*, Feb. 2012, pp. 190–192.

[38] S. Foroutan, A. Sheibanyrad, and F. Pétrot, "Cost-efficient buffer sizing in shared-memory 3D-MPSoCs using wide I/O interfaces," in *Proc. 49th Annu. DAC*, 2012, pp. 366–375.

[39] J. Meng, K. Kawakami, and A. K. Coskun, "Optimizing energy efficiency of 3-D multicore systems with stacked DRAM under power and thermal constraints," in *Proc. 49th ACM/EDAC/IEEE DAC*, Jun. 2012, pp. 648–655.

[40] Y. Xu, Y. Du, B. Zhao, X. Zhou, Y. Zhang, and J. Yang, "A low-radix and low-diameter 3D interconnection network design," in *Proc. IEEE 15th Int. Symp. HPCA*, Feb. 2009, pp. 30–42.

[41] J. Jung, K. Kang, and C.-M. Kyung, "Design and management of 3D-stacked NUCA cache for chip multiprocessors," in *Proc. 21st GLSVLSI*, 2011, pp. 91–96.

[42] G. Sun, H. Yang, and Y. Xie, "Performance/thermal-aware design of 3D-stacked L2 caches for CMPs," *ACM Trans. Design Autom. Electron. Syst.*, vol. 17, no. 2, p. 13, Apr. 2012.

[43] N. Madan *et al.*, "Optimizing communication and capacity in a 3D stacked reconfigurable cache hierarchy," in *Proc. IEEE 15th Int. Symp. HPCA*, Feb. 2009, pp. 262–274.

[44] S. Chou *et al.*, "No cache-coherence: A single-cycle ring interconnection for multi-core L1-NUCA sharing on 3D chips," in *Proc. 46th ACM/IEEE DAC*, Jul. 2009, pp. 587–592.

[45] H. Saito *et al.*, "A chip-stacked memory for on-chip SRAM-Rich SoCs and processors," *IEEE J. Solid-State Circuits*, vol. 45, no. 1, pp. 15–22, Jan. 2010.

[46] M. Facchini, P. Marchal, F. Catthoor, and W. Dehaene, "An RDL-configurable 3D memory tier to replace on-chip SRAM," in *Proc. DATE*, Mar. 2010, pp. 291–294.

[47] D. H. Kim *et al.*, "3D-MAPS: 3D Massively parallel processor with stacked memory," in *Proc. IEEE ISSCC*, Feb. 2012, pp. 188–190.

[48] D. Fick, A. DeOrio, G. Chen, V. Bertacco, D. Sylvester, and D. Blaauw, "A highly resilient routing algorithm for fault-tolerant NoCs," in *Proc. DATE*, Apr. 2009, pp. 21–26.

[49] C. Feng, M. Zhang, J. Li, J. Jiang, Z. Lu, and A. Jantsch, "A low-overhead fault-aware deflection routing algorithm for 3D network-on-chip," in *Proc. ISVLSI*, Jul. 2011, pp. 19–24.

[50] A.-M. Rahmani, P. Liljeberg, K. Latif, J. Plosila, K. R. Vaddina, and H. Tenhunen, "Congestion aware, fault tolerant, and thermally efficient inter-layer communication scheme for hybrid NoC-bus 3D architectures," in *Proc. 5th IEEE/ACM Int. Symp. NoCS*, May 2011, pp. 65–72.

[51] S. Akbari, A. Shafiee, M. Fathy, and R. Berangi, "AFRA: A low cost high performance reliable routing for 3D mesh NoCs," in *Proc. DATE*, Mar. 2012, pp. 332–337.

[52] V. Pasca, L. Anghel, C. Rusu, R. Locatelli, and M. Coppola, "Error resilience of intra-die and inter-die communication with 3D spidergon STNoC," in *Proc. DATE*, Mar. 2010, pp. 275–278.

[53] W. Jang, O. He, J.-S. Yang, and D. Z. Pan, "Chemical-mechanical polishing aware application-specific 3D NoC design," in *Proc. IEEE/ACM ICCAD*, Nov. 2001, pp. 207–212.

[54] F. Sun, A. Cevrero, P. Athanasopoulos, and Y. Leblebici, "Design and feasibility of multi-Gb/s quasi-serial vertical interconnects based on TSVs for 3D ICs," in *Proc. 18th IEEE/IFIP VLSISoC*, Sep. 2010, pp. 149–154.

[55] F. Darve, A. Sheibanyrad, P. Vivet, and F. Petrot, "Physical implementation of an asynchronous 3D-NoC router using serial vertical links," in *Proc. ISVLSI*, Jul. 2011, pp. 25–30.

[56] P. Vivet, D. Dutoit, Y. Thonnart, and F. Clermidy, "3D NoC using through silicon via: An asynchronous implementation," in *Proc. IEEE/IFIP 19th Int. Conf. VLSISoC*, Oct. 2011, pp. 232–237.

[57] F. Clermidy, F. Darve, D. Dutoit, W. Lafi, and P. Vivet, "3D embedded multi-core: Some perspectives," in *Proc. DATE*, Mar. 2011, pp. 1–6.

[58] S. Pasricha, "A framework for TSV serialization-aware synthesis of application specific 3D networks-on-chip," in *Proc. 25th Int. Conf. VLSI Design*, Jan. 2012, pp. 268–273.

[59] V. Pasca, L. Anghel, C. Rusu, and M. Benabdenbi, "Configurable serial fault-tolerant link for communication in 3D integrated systems," in *Proc. IEEE 16th IOLTS*, Jul. 2010, pp. 115–120.

[60] U. Kang *et al.*, "8 Gb 3-D DDR3 DRAM using through-silicon-via technology," *IEEE J. Solid-State Circuit*, vol. 45, no. 1, pp. 111–119, Jan. 2010.

[61] X. Dong and Y. Xie, "System-level cost analysis and design exploration for three-dimensional integrated circuits (3D ICs)," in *Proc. ASP-DAC*, Jan. 2009, pp. 234–241.

[62] T. Frank *et al.*, "Resistance increase due to electromigration induced depletion under TSV," in *Proc. IEEE IRPS*, Apr. 2011, pp. 3F.4.1–3F.4.6.

[63] ARM Cortex-M3 Processor. *Cortex-M Series*. [Online]. Available: http://www.arm.com/products/processors/cortex-m/index.php, accessed Aug. 21, 2014.

[64] D. Tarjan, S. Thoziyoor, and N. P. Jouppi, "CACTI 4.0," HP Lab., Palo Alto, CA, USA, Tech. Rep. HPL-2006-86, Jun. 2006.

[65] T. Sakurai, "Closed-form expressions for interconnection delay, coupling, and crosstalk in VLSIs," *IEEE Trans. Electron Devices*, vol. 40, no. 1, pp. 118–124, Jan. 1993.

[66] R. Weerasekera, D. Pamunuwa, L.-R. Zheng, and H. Tenhunen, "Two-dimensional and three-dimensional integration of heterogeneous electronic systems under cost, performance, and technological constraints," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 28, no. 8, pp. 1237–1250, Aug. 2009.

[67] J. E. Miller *et al.*, "Graphite: A distributed parallel simulator for multicores," in *Proc. IEEE 16th Int. Symp. HPCA*, Jan. 2010, pp. 1–12.

[68] S. C. Woo, M. Ohara, E. Torrie, J. P. Singh, and A. Gupta, "The SPLASH-2 programs: Characterization and methodological considerations," in *Proc. 22nd Annu. ISCA*, Jun. 1995, pp. 24–36.

[69] J. M. Cecilia, J. M. García, and M. Ujaldón, "CUDA 2D stencil computations for the Jacobi method," in *Applied Parallel and Scientific Computing*, K. Jonasson, Ed. Berlin, Germany: Springer-Verlag, 2012, pp. 173–183.

[70] B. Bilgic, B. K. P. Horn, and I. Masaki, "Efficient integral image computation on the GPU," in *Proc. IEEE IVS*, Mar. 2010, pp. 528–533.

[71] Y. Chen, D. Niu, Y. Xie, and K. Chakrabarty, "Cost-effective integration of three-dimensional (3D) ICs emphasizing testing cost analysis," in *Proc. IEEE/ACM ICCAD*, Nov. 2010, pp. 471–476.

[72] K. Kang, L. Benini, and G. D. Micheli, "A high-throughput and low-latency interconnection network for multi-core clusters with 3-D stacked L2 tightly-coupled data memory," in *Proc. 20th Int. Conf. VLSI-SoC*, Oct. 2012, pp. 283–286.

[73] Y.-H. Kao, M. Yang, N. S. Artan, and H. J. Chao, "CNoC: High-radix clos network-on-chip," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 30, no. 12, pp. 1897–1910, Dec. 2011.

[74] J. Kim, J. Balfour, and W. J. Dally, "Flattened butterfly topology for on-chip networks," in *Proc. 40th Annu. IEEE/ACM Int. Symp. Microarchitecture*, Nov. 2007, pp. 172–182.

[75] A. Zeng, J. Lu, K. Rose, and R. J. Gutmann, "First-order performance prediction of cache memory with wafer-level 3D integration," *IEEE Des. Test. Comput.*, vol. 22, no. 6, pp. 548–555, Nov./Dec. 2005.

[76] *CADENCE Virtuoso Spectre Circuit Simulator*. [Online]. Available: http://www.cadence.com/products/cic/spectre_circuit/pages/default.aspx, accessed Aug. 21, 2014.

[77] W. Liao and L. He, "Full-chip interconnect power estimation and simulation considering concurrent repeater and flip-flop insertion," in *Proc. ICCAD*, Nov. 2003, pp. 574–580.

[78] S. Li, J. H. Ahn, R. D. Strong, J. B. Brockman, D. M. Tullsen, and N. P. Jouppi, "McPAT: An integrated power, area, and timing modeling framework for multicore and manycore architectures," in *Proc. 42nd Annu. IEEE/ACM Int. Symp. Microarchit.*, Dec. 2009, pp. 469–480.

[79] R. Kumar, K. I. Farkas, N. P. Jouppi, P. Ranganathan, and D. M. Tullsen, "Single-ISA heterogeneous multi-core architectures: The potential for processor power reduction," in *Proc. 36th Annu. IEEE/ACM Int. Symp. Microarchit.*, Dec. 2003, pp. 81–92.

[80] W. Lafi, D. Lattard, and A. Jerraya, "An efficient hierarchical router for large 3D NoCs," in *Proc. 21st IEEE Int. Symp. RSP*, Jun. 2010, pp. 1–5.

[81] M. Salas and S. Pasricha, "The roce-bush router: A case for routing-centric dimensional decomposition for low-latency 3D noC routers," in *Proc. CODES ISSS*, 2012, pp. 171–180.

[82] W. J. Dally and B. P. Towles, *Principles and Practices of Interconnection Networks*. New York, NY, USA: Elsevier, 2004.

Kyungsu Kang (S'06–M'10) received the B.S. degree from the Department of Electrical and Electronic Engineering, Kyungpook National University, Daegu, Korea, in 2003, and the M.S. and Ph.D. degrees from the Department of Electrical Engineering and Computer Science, Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea, in 2010.

He was a Post-Doctoral Fellow with the Smart Sensor Architecture Laboratory, KAIST, from 2010 to 2011, and Integrated Systems Laboratory, École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland, from 2011 to 2013. He has been with Memory Business, Samsung Electronics, Suwon, Korea, since 2013. His current research interests include 3-D integration, networks-on-chip, dynamic power/thermal management, and memory/storage system solutions.

Luca Benini (F'07) is currently a Full Professor with the University of Bologna, Bologna, Italy, and also the Chair of Digital Integrated Circuits and Systems with ETH Zurich, Zurich, Switzerland. He has served as a Chief Architect of the Platform 2012/STHORM Project with: STMicroelectronics, Grenoble, France, from 2009 to 2013. He has held Visiting and Consulting Researcher positions with: École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland; imec, Leuven, Belgium; the Hewlett-Packard Laboratories, Palo Alto, CA, USA; and Stanford University, Stanford, CA, USA. He is involved in energy-efficient smart sensors and sensor networks for biomedical and ambient intelligence applications. He has authored more than 700 papers in peer-reviewed international journals and conferences, four books, and several book chapters. His current research interests include energy-efficient system design and multicore SoC design.

Dr. Benini is a member of the Academia Europaea.

Giovanni De Micheli (S'79–M'79–SM'80–F'94) received the Nuclear Engineer degree from the Politecnico di Milano, Milan, Italy, in 1979, and the M.S. and Ph.D. degrees in electrical engineering and computer science from the University of California at Berkeley, Berkeley, CA, USA, in 1980 and 1983, respectively.

He was a Professor of Electrical Engineering with Stanford University, Stanford, CA, USA. He is currently a Professor and the Director of the Institute of Electrical Engineering with the Integrated Systems Centre, École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland, where he is also the Program Leader of the Nano-Tera.ch Program. He has authored or co-authored over 600 papers in journals and conferences, and a book entitled *Synthesis and Optimization of Digital Circuits* (McGraw-Hill, 1994), and co-authored and co-edited eight other books. His citation H-index is 84 according to Google Scholar. His current research interests include design technologies for integrated circuits and systems, such as synthesis for emerging technologies, networks on chips, 3-D integration, and heterogeneous platform design, including electrical components and biosensors, and data processing of biomedical information.

Prof. De Micheli is a fellow of the Association for Computing Machinery and a member of the Academia Europaea and Scientific Advisory Board of imec and STMicroelectronics. He has been with IEEE in several capacities, including as the Division 1 Director from 2008 to 2009, Co-Founder and President Elect of the IEEE Council on Electronic Design Automation from 2005 to 2007, President of the IEEE Circuits and Systems Society in 2003, and Editor-in-Chief of the IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN (CAD)/INTEGRATED CIRCUITS AND SYSTEMS (ICAS) from 1987 to 2001. He has been the Chair of several conferences, including DATE since 2010, pHealth since 2006, International Conference on Very Large Scale Integration since 2006, Design Automation Conference (DAC) since 2000, and International Conference on Computer Design since 1989. He was a recipient of the 2012 IEEE/Circuits and Systems Society (CAS) Mac Van Valkenburg Award for contributions to theory, practice, and experimentation in design methods and tools, the 2003 IEEE Emanuel Piore Award for contributions to computer-aided synthesis of digital systems, the Golden Jubilee Medal for outstanding contributions to the IEEE CAS Society in 2000, and the D. Pederson Award for the best paper in the IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN/INTEGRATED CIRCUITS AND SYSTEMS in 1987, and several best paper awards, including the DAC in 1983 and 1993, the DATE in 2005, and the Nanoarch in 2010 and 2012.