# A High-throughput and Low-Latency Interconnection Network for Multi-Core Clusters with 3-D Stacked L2 Tightly-Coupled Data Memory

Kyungsu Kang[*], Luca Benini[§], and Giovanni De Micheli[*]
[*]LSI, EPFL, Lausanne, Switzerland, {kyungsu.kang, giovanni.demicheli}@epfl.ch
[§]DEIS, University of Bologna, Bologna, Italy, luca.benini@unibo.it

***Abstract*— The performance of most digital systems today is limited by the interconnect latency between logic and memory, rather than by the performance of logic or memory itself. Three-dimensional (3-D) integration using through-silicon-vias (TSVs) may provide a solution to overcome the scaling limitations by stacking multiple memory dies on top of a many-core die. In this paper, we propose a Mesh-of-Trees (MoT) network to support high-throughput and low-latency communication between processing cores and 3-D stacked multi-banked shared L2 data memory. Compared to conventional MoT network [5] that is straightforwardly adapted to 3-D integration, the experimental results show that the proposed network significantly improves the number of operations per second. We also investigate the architecture parameters of 3-D memory stacking (e.g., number of tiers to be stacked, TSV sharing, etc.) that affect the interconnection network as well as the system performance and fabrication cost, which permits to explore trade-offs among different 3-D memory stacking architectures.**

## I. Introduction

Nowadays, the increasing focus on energy-efficient architecture coupled with a slowdown in clock speed improvement have brought a growing interest in parallel computing where a large number of simple cores are integrated onto the same die. GP-GPUs such as NVIDA Fermi [1], HyperCore [2], and STMicroelectronics Platform 2012 [3] are the most visible examples in this trend. All of the cited architectures share a common trait: a multi-core cluster consisting of many simple cores and an on-chip shared tightly-coupled data memory (TCDM). The shared TCDM enables parallel threads to cooperate with each other, facilitates extensive reuse of on-chip memory data, and greatly reduces the off-chip memory accesses. However, the design of low-latency and high-bandwidth on-chip interconnection network is crucial for such multi-core clusters having shared TCDM for parallel processing [4, 5].

3-D integration by using through-silicon vias (TSVs) is a promising option to overcome the scaling limitations of 2-D integrated circuit (IC) including the well-known memory-wall problem [6]. However, in such integration, there is an inherent asymmetry in the delays between the fast vertical interconnections and the horizontal interconnections due to the differences in wire lengths (few tens of µm in the vertical direction as compared to few thousand µm in the horizontal direction). Vertical interconnections also impose a larger area overhead than corresponding horizontal wires due to the requirement for bonding pads and can compete with device area as the TSVs punch through the wafer when Face-to-Back bonding is used. Therefore, the design of 3-D interconnection network brings new constraints and opportunities as compared to that of 2-D interconnection network.

In this work, we focus on the communication between multiple cores and a shared multi-banked L2 TCDM which is multiply stacked on top of the multi-core die. By avoiding cache coherence overheads as well as cache indeterminacy, the shared L2 TCDM can be used as a frame buffer for video processing which should deal with a large amount of data within a tightly bounded time [7, 8]. The fully combinational Mesh-of-Trees (MoT) interconnection network proposed in [5] is suitable for this shared multi-banked L2 TCDM with high throughput and low memory access latency. However, straightforward extension of a traditional MoT network to the third dimension by simply inserting TSVs at every connection from the interconnection network to the memory banks (which we call *plain MoT*) is not a good option, when considering the large TSV overhead as well as the inherent asymmetry of wire delay in 3-D ICs.

This paper provides two contributions. First, we propose a new 3-D MoT network that is suitable for a multi-core cluster with a shared multi-banked L2 TCDM stacked onto the multi-core cluster. By inserting sequential routing switches on the critical paths instead of combinational ones, interconnection network fully exploits the delay asymmetry in 3-D ICs, which increases clock speed of the interconnection network and, thus, the system performance. As far as the authors know, this is the first work that considers MoT network for shared L2 TCDM in 3-D IC. Second, we investigate various parameters of 3-D stacked L2 TCDM architecture, such as the number of memory tiers stacked and TSV sharing, while taking into account the communication contention at the shared TSVs. This investigation allows us to exploit the 3-D MoT properties with different 3-D memory structures in the view of footprints, interconnect latency, system performance, and fabrication cost.

## II. Target 3-D Architecture

Figure 1 shows an example of MoT interconnect consisting of four cores and eight stacked L2 TCDM 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. During a read/write operation, data and control signals are asserted in the form of packet by the cores. When a packet needs to be arbitrated among the other simultaneous packets forward to the same memory bank, the round-robin algorithm is used to provide a starvation-free arbitration.

Figure 2 shows a schematic of our target 3-D multi-core cluster architecture. As shown in Figure 2 (a), the MoT interconnect (i.e., routing and arbitration switches shown in Figure 1) 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

This research presented in this paper was supported by the NanoSys project, the program ERC-AdG-246810

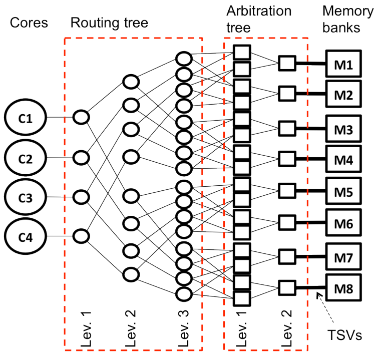

Figure 1. Plain 4x8 mesh of trees (MoT): empty circles represent routing switches and empty squares represent arbitration switches.

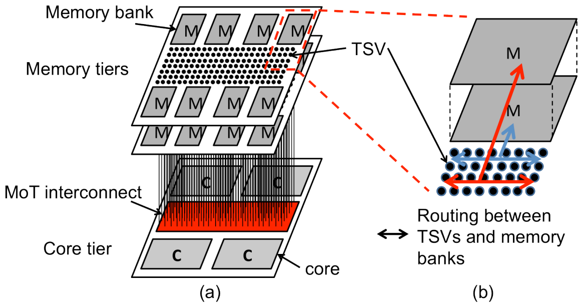

Figure 2. (a) Illustration of 3-D multi-core cluster with stacked L2 TCDM banks, (b) TSVs allocation to each stacked memory bank.

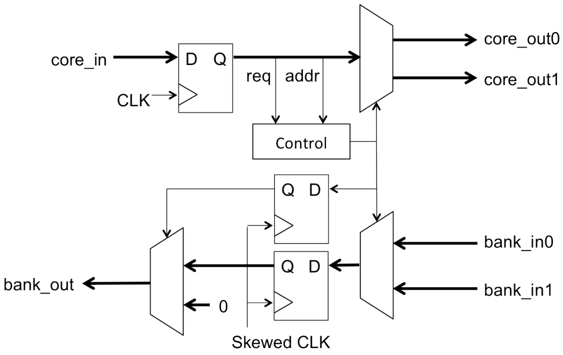

Figure 3. Proposed sequential routing switch for 3-D MoT interconnect

of the arbitration tree are directly connected to each memory bank through TSVs, which are distributed in the middle of the memory die [9]. TSVs are allocated per bank and each bank is connected to neighboring TSVs as shown in Figure 2 (b). Note that the size of stacked L2 TCDM dies does not need to be the same with that of the multi-core die while assuming that all the memory dies have an identical layout for the fabrication cost.

## III. 3-D MoT Interconnection Network

### A. *Sequential Routing Switches*

As mentioned before, in 3-D integration, there is an inherent asymmetry in the delays between the fast vertical interconnections and the horizontal interconnections due to the differences in wire lengths. The latency difference between the two directions is even larger as the delay of TSVs is getting smaller. To eliminate the wide disparity, we propose a sequential routing switch as shown in Figure 3, which can replace some of combinational routing switches without any network configuration change. The sequential routing switch has two directions: forward (core ports) which sends out the incoming packet from its input port at core side to one of its output ports at memory side; backward which rolls packet back from memory side to core side (bank ports). The flip-flops are inserted at each packet direction in order to buffer the incoming packets as well as the memory control signal. The flip-flop on the forward direction is clocked with the main clock CLK, whereas the others on the backward direction are clocked with a skewed CLK which is able to transfer data on both the rising and falling edges of the clock signal [5].

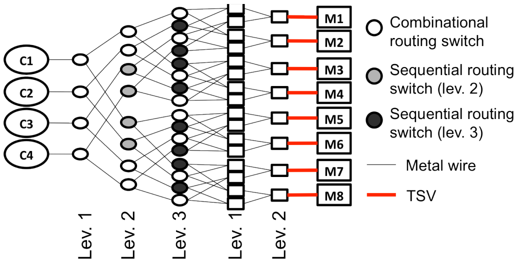
C1
C2
C3
C4
M1
M2
M3
M4
M5
M6
M7
M8
Lev. 1
Lev. 2
Lev. 3
Lev. 1
Lev. 2
○ Combinational routing switch
● Sequential routing switch (lev. 2)
● Sequential routing switch (lev. 3)
Metal wire
TSV

Figure 4. Inserting sequential routing switches for four cores with eight-bank L2 TCDM

Inserting such sequential logic on the combinational paths of MoT interconnect reduces the longer horizontal wire delay and, thus, allows the MoT interconnect to run at higher clock frequency, while sacrificing the number of clock cycles to be consumed. For this reason, it is important to consider how to insert the sequential routing switches in the fully combinational MoT network. Figure 4 shows an example of inserting sequential routing switches for four cores with eight-banked L2 TCDM, where those are connected with a 3-D 4x8 MoT interconnection network. Thanks to the inherent characteristics of a binary tree, inserting $N_{core}\cdot(N_{lev}–1)$ sequential routing switches at each routing level makes half of the memory banks to be closer and the rest to be farther, where $N_{lev}$ is the current level at the routing tree. When assuming that $N_{seq}$ is the number of routing levels where $N_{core}\cdot(N_{lev}–1)$ sequential routing switches are inserted at each routing level, the number of the closest banks for each core is to be $N_{bank}/2^{Nseq}$. Also, the farthest banks are accessed from a core by passing $N_{seq}$ sequential routing switches, which means the number of clock cycles for the farthest bank access is $2\cdot N_{seq}+1$. Note that the non-uniform memory access latency with appropriate memory data placement policies, such as thread-affinity-based memory data placement [10], gives significant performance improvement, as shown in the experimental results.

### B. *TSV Sharing Effects*

TSVs connect multiple stacked dies with good electrical characteristics, but their area footprint is much bigger with respect to the on-chip metal lines. Sharing TSVs among L2 TCDM banks which are directly stacked on each other (which we called a *bank stack*) is the most straightforward method to reduce TSVs [9]. The total number of TSVs is reduced with respect to $N_{tier}$ as follows.

$$N_{tsv} = N_c + 1 + \frac{N_{bank}}{N_{tier}} \cdot (\log_2 N_{tier} + Nb_{addr} + Nb_{data}) \qquad (1)$$

where $N_c$ is the number of clock TSVs added to one reset TSV. The rest of the equation presents the number of TSVs needed per bank stack. $\log_2 N_{tier}$ is the number of bits needed for the tier ID. $Nb_{addr}$ and $Nb_{data}$ are the number of bits for memory address and data, respectively.

Sharing TSVs among banks in a bank stack gives the possibility to reduce the interconnect latency as $N_{tier}$ increases owing to the reduction in the area occupied by TSVs as well as the routing switches to be passed. However, it may cause collision at the shared TSVs when cores access different memory banks in the same bank stack. The amount of collision strongly depends on the rate of L2 data memory accesses of applications to be run on the cores as well as the number of banks sharing the same TSVs (which is the same of $N_{tier}$). To estimate memory contention at the shared TSVs with respect to $N_{tier}$, we adopt the M/M/1 queuing model where the arrival and service rate are assumed to be Poisson distribution [11]. In Section IV, we show the experiment results that allow us to find the performance trade-offs, in terms of operations per second, for different values of $N_{tier}$.

When considering the high manufacturing cost of 3-D integration due to high TSV failure rate as well as reduced yield, the reduction in both the die area and number of TSVs resulted from TSV sharing makes the fabrication yield higher and reduces the fabrication cost compared to 3-D stacked TCDM without TSV sharing. The stacking yield for $N_{tier}$ dies can be modeled as follows [12].

$$Y_{stacking} = \left\{ Y_{bonding} \cdot (1 - f_{tsv})^{N_{tsv}} \right\}^{N_{tier}-1} \qquad (2)$$

where $Y_{bonding}$ captures the yield loss of the chip due to the faults in the bonding process and $f_{tsv}$ is the TSV failure rate.

## IV. Experimental Results

We performed experiments using a 3-D multi-core cluster with shared multi-banked L2 data memory stacked on top of the multi-core cluster. The cluster consists of 32 cores and 64 memory banks. The core is considered to be ARM Cortex-A5 with 16KB/16KB instruction and data caches. The core estimated area is 1.183mm x 1.183mm for 65nm technology [16]. The core operating clock frequency ($f_{core}$) is assumed to be 1GHz. Each L2 TCDM bank has a capacity of 64 KB and a size of 0.867mm x 0.624mm, which are estimated for 65nm technology [17]. The access time of a bank itself (i.e., delay due to row decode, sense amplifier, and multiplexer in the bank) is assumed to be 1.062 ns. The number of stacked memory tiers ($N_{tier}$) used in the experiments varies from 1 to 8. For the simulation, an in-house simulator is used, whose details are explained below.

In order to estimate MoT network performance, the latency for the longest possible link between cores and memory banks is estimated using Elmore distributed resistance-capacitance (RC) delay model for 65nm technology [13][14]. We assumed that TSV pitches of 10μm x 10μm, TSV diameter of 5μm, and TSV height of 20μm were used. To evaluate the many-core system performance, we used the metric of operations per second (OPS) presented in [15]. The average memory access time (i.e., the sum of average access times of L1 data cache, L2 TCDM, and off-cluster memory) is presented as follows.

$$t_{mem} = p_{TCDM} t_{TCDM} + (1 - p_{TCDM})\{p_{hit} t_{cache} + (1 - p_{hit}) \cdot t_{off-cluster}\} \quad (3)$$

where $p_{hit}$ and $p_{TCDM}$ are, respectively, the average hit ratio of L1 data cache and average access ratio of L2 TCDM for each thread. $t_{cache}$ and $t_{off\text{-}cluster}$ is the latency (in terms of cycles) for the L1 data cache hit and off-cluster memory access. $t_{TCDM}$ is the latency for the L2 TCDM access (i.e., sum of interconnect delay from core to a target memory bank and the access time of the bank itself). All the values related to the system performance evaluation are shown in Table I. Note that, in Table I, $p_0$ and $p_1$ represents the probabilities to access TCDM bank regions divided by sequential routing switches (when $N_{seq}$ is 1), which can be general when static or dynamic data mapping methods are used in parallel processing [10]. For the fabrication cost estimation, we used analytical models proposed in [12] [23] assuming that wafer-to-wafer (W2W) and face-to-back 3-D bonding is performed. We assumed that $Y_{bonding}$ and $f_{tsv}$ in Equation (2) to be 0.99 and 0.00001, respectively.

TABLE I. Parameter values for performance evaluation

| Parameter | Value | Parameter | Value |
|---|---|---|---|
| $t_{cache}$ | 1 cycle | $N_{tpc}$ | 2 |
| $t_{off\text{-}cluster}$ | 100 cycles | $IPC$ | 2.0 |
| $r$ | 0.3 | $p_{TCDM}$ | 0.2 |
| $p_{hit}$ | 0.9 | $p_0/p_1$ | 0.8 / 0.2 |

For architecture comparisons, we evaluated four candidates;

**2-D MoT**: All the cores and memory banks are placed on 2-D planar structure.

**Plain MoT**: Multiple memory tiers are stacked on the multi-core tier and memory banks are connected to cores through a plain MoT (presented in Section II).

**SRS**: Multiple memory tiers are stacked on the multi-core tier and memory banks are connected to cores through 3-D MoT with sequential routing switches (presented in Section III.A).

**TSVshare**: Multiple memory tiers are stacked on the multi-core tier and memory banks are connected to cores through 3-D MoT with TSV sharing (presented in Section III.B).

**SRS+TSVshare**: Multiple memory tiers are stacked on the multi-core tier and memory banks are connected to cores through 3-D MoT with both sequential routing switches and TSV sharing.

Figure 5 shows the results of MoT network clock frequency (i.e., the reciprocal of MoT network latency) with respect to the number of L2 TCDM tiers, i.e., $N_{tier}$. In case of *SRS* and *SRS+TSVshare*, we assumed that the number of sequential routing level, i.e., $N_{seq}$, is 1. As shown in Figure 5, most of the memory-stacked architectures show the increase of MoT

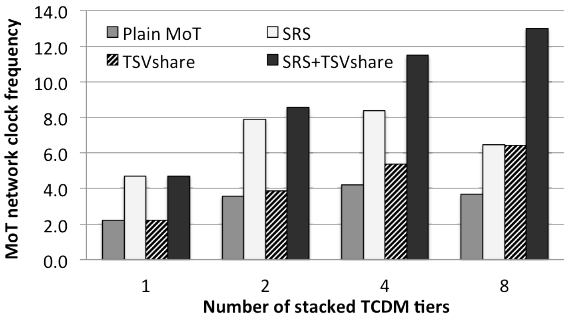

Figure 5. Results of MoT network clock frequency. All the values are normalized with respect to the clock frequency of *2-D MoT*.

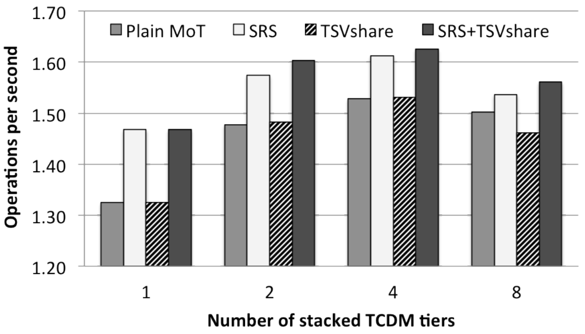

Figure 6. Results of operations per second (OPS). All the values are normalized with respect to the OPS of *2-D MoT*.

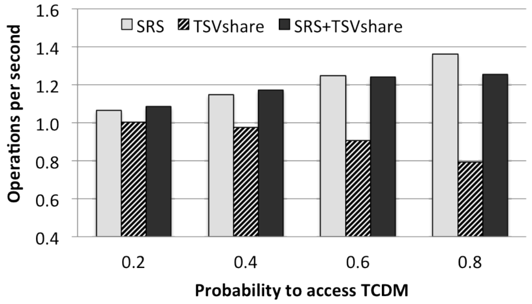

Figure 7. Results of OPS with respect to the probability to access TCDM ($p_{TCDM}$) when $N_{tier}$ is 2. All the values are normalized with respect to the OPS of *Plain MoT*.

network clock frequency as $N_{tier}$ increases, since the on-chip global wires decreases as the floorplan size of TCDM tiers decreases with $N_{tier}$. However, in case of *Plain MoT* and *SRS*, the clock frequency does not increase consistently with $N_{tier}$ because of the large area of TSVs. In these two architectures, the number of TSVs and, thus, the area occupied by TSVs is linearly increasing with $N_{tier}$, while the footprint of TCDM tier is decreasing. So, as the area of occupied by the TSVs is becoming dominant, the interconnect delay may increase with $N_{tier}$ (e.g., $N_{tie\text{r}}= 8$ in Figure 5). Sharing TSVs among banks in a bank stack largely affects the system performance due to the reduction in both the area occupied by the TSVs and the level of the routing tree. These reductions allow us to reduce the horizontal wire delay and, thus, increase the maximum available clock frequency. However, the memory contention occurring at the shared TSVs may degrade the system performance despite of the increase in clock frequency. As can be seen in Figure 6, *TSVshare* and *SRS+TSVshare* do not give steady performance improvement with respect to $N_{tier}$ even though *SRS+TSVshare* yields the best system performance. In Figure 7, we varied the L2 TCDM access probability from 0.2 to 0.8 when $N_{tier}$ is 2. TSV sharing makes the system performance worse as the probability to access TCDM, i.e., $p_{TCDM}$, increases because of the heavy contentions at the shared TSVs. Note that inserting sequential routing switches gives better performance improvement as $p_{TCDM}$ increases.

When comparing the fabrication cost between *SRS* and *SRS+TSVshare* with respect to $N_{tier}$, *SRS+TSVshare* gives lower fabrication cost than *SRS* and the disparity increases with $N_{tier}$ (up to 47% of the disparity when $N_{tier}$ is 8) because of the large area overhead as well as the high failure of TSV itself in case of *SRS*. Without TSV sharing, the large number of TSVs causes large die area and reduces stacking yield, which results in high fabrication cost.

## V. CONCLUSION

In this paper, we presented a MoT interconnection network that can be integrated in a multi-core cluster where 3-D multi-banked shared L2 TCDM is stacked on the multi-core die. To exploit the fast vertical interconnections in 3-D integration, we proposed a sequential routing switch that can be adapted to the plain MoT interconnect without any network configuration changes. The experimental results show that the proposed sequential routing switch significantly improves the system performance. The architecture parameters of 3-D stacked memory also have been explored with TSV sharing. TSV sharing reduces fabrication cost as well as gives the highest MoT network clock frequency owing to the reduced form factor. However, since the system performance deeply depends on the memory contention at the shared TSVs, new solutions such as adding additional paths using redundant TSVs are needed in order to compensate the memory contentions, which will be our future work.

## REFERENCES

[1] NVDIA, The next generation CUDA architecture, code named Fermi, [online]. Available: www.nvidia.com/object/ fermi_architecture.htm.

[2] Plurality Ltd. “The hypercore architecture,” in white paper, Jan. 2010.

[3] ST Microelectronics and CEA, “Platform 2012: A many-core programmable accelerator for ultra-efficient embedded computing in nanometer technology,” in white paper, 2010.

[4] A. O. Balkan, G. Qu, and U. vishkin, “A mesh-of-trees interconnection network for single-chip parallel processing,” ASAP, 2006, pp. 73 – 80.

[5] A. Rahimi et al., “A fully-synthesizable single-cycle interconnection network for shared-L1 processor clusters,” DATE, 2011, pp. 1 – 6.

[6] G. Loh, “3D-stacked memory architectures for multi-core processors,” ISCA, 2008, pp. 453 – 464.

[7] J. H. Ahn et al., “Evaluating the imagine stream architecture,” ISCA, 2004, pp. 14 – 25.

[8] S. Kyo, S. Okazaki, and T. Arai, “An integrated memory array processor architecture for embedded image recognition systems,” ISCA, 2005, pp. 134 – 145.

[9] U. Kang et al., “8 Gb 3-D DDR3 DRAM using through-silicon-via technology,” IEEE JSSC, vol. 45, no. 1, Jan. 2010.

[10] A. Marongiu, M. Ruggiero, and L. Benini, “Efficient OpenMP data mapping for multicore platforms with vertically stacked memory,” DATE, 2010, pp. 105 – 110.

[11] E. Gelenbe and G. Pujolle, “Introduction to queueing networks,” 2nd Ed., Wiley Publisher, July 1998.

[12] Y. Chen et al., “Cost-effective integration of three-dimensional (3D) ICs emphasizing testing cost analysis,” ICCAD, 2010, pp. 471 – 476.

[13] T. Sakurai, “Closed-form expressions for interconnection delay, coupling, and crosstalk in VLSI’s,” IEEE Trans. Electron Devices, vol. 40, no. 1, pp. 118 – 124, Jan. 1993.

[14] G. Katti et al., “Electrical modeling and characterization of through silicon via for three-dimensional ICs,” IEEE Trans. on. Electron Devices, Jan. 2010.

[15] Z. Guz et al., “Many-core vs. many-thread machines: stay away from the valley,” IEEE Computer Architecture Letters, vol. 8, no. 1, Jan 2009.

[16] ARM Cortex-M3 Processor, [online]. Available: http://www.arm.com/products/processors/cortex-m/index.php

[17] D. Tarjan, S. Thoziyoor, and N. P. Jouppi, “CACTI 4.0,” HP Laboratories, Palo Alto, CA, Tech. Rep. HPL-2006-86, Jun. 2006.