3D-LIN: A Configurable Low-Latency Interconnect for Multi-Core Clusters with 3D Stacked L1 Memory

Giulia Beanato*, Igor Loi†, Giovanni De Micheli*, Yusuf Leblebici* and Luca Benini†
*EPFL, Lausanne, Switzerland
†DEIS, University of Bologna, Bologna, Italy
giulia.beanato@epfl.ch, igor.loi@unibo.it, giovanni.demicheli@epfl.ch, yusuf.leblebici@epfl.ch, and luca.benini@unibo.it

***Abstract*—Shared L1 memories are of interest for tightly-coupled processor clusters in programmable accelerators as they provide a convenient shared memory abstraction while avoiding cache coherence overheads. The performance of a shared-L1 memory critically depends on the architecture of the low-latency interconnect between processors and memory banks, which needs to provide ultra-fast access to the largest possible L1 working set. The advent of 3D technology provides new opportunities to improve the interconnect delay and the form factor. In this paper we propose a network architecture, 3D-LIN, based on 3D integration technology. The network can be configured based on user specifications and technology constraints to provide fast access to L1 memories on multiple stacked dies. The extracted results from the physical synthesis of 3D-LIN permit to explore trade-offs between memory size and network latency from a planar design to multiple memory layers stacked on top of logic. In the case where the system memory requirements lead to a memory area that occupies 60% of the chip, the form factor can be reduced by more than 60% by stacking 2 memory layers on the logic. Latency reduction is also promising: the network itself, configured for connecting 16 processing elements to 128 memory banks on 2 memory layers is 24% faster than the planar system.**

I. Introduction

Following Moore's law, the scaling to nanometer technologies has led to a transition from single-core to multi-core processors, and is now moving towards many-cores architectures [1]. Whereas hundreds of millions of transistors can now be placed on a single chip leading to increased computing power, they cannot be fully exploited due to interconnect latency. In nanometer-scale technologies, interconnect latency and power do not scale as much as device geometries, thus becoming a performance bottleneck. These limiting factors need to be overcome at the architectural level. For many applications, the exploitation of customized accelerators will be the way to obtain the highest performance, together with more efficient types of interconnect and memory hierarchies [2].

For this reason, new interconnect architectures have already been envisaged. For instance, *Network-on-chip (NoC)* [3] has been adopted to substitute conventional bus-based systems when high bandwidth and high speed are required. When ultra-low latency processor to memory interconnection is requested for parallel computing, novel fast interconnect topologies are imperative to guarantee the access to the memory in few clock cycles. Several research efforts are already focused on low-latency, high-bandwidth connection between the processing elements and multi-banked on-chip memories. The *Mesh-of-Trees (MoT)* Interconnection Network proposed in [4], the Hyper-core architecture [5] and the single-cycle interconnection network presented in [6] are just few examples of low-latency networks. Nevertheless, future generations of *Chip Multi-Processor (CMP)* require a major innovation in both integration technology and on-chip communication infrastructure.

A promising option to overcome the barrier in interconnect scaling is the 3D integration of integrated circuits (3D ICs) [7]. Stacking multiple chips and connecting them by *Through Silicon Vias (TSVs)* has the potential to reduce the interconnect wire length while offering high vertical connect density. Multi-cores and many-cores processors can benefit from several characteristics of 3D devices: (a) Wire length reduction improves the latency of core to memory interconnect; (b) High TSV density and their small length can be exploited for improving memory bandwidth when stacking memory layers on top of logic layers; (c) The smaller form factor due to the addition of a third dimension is essential for moving on-chip the memory required by the processing elements (PC) avoiding slow off-chip connections.

This paper aims to propose a fully synthesizable *3D Logarithmic Interconnection Network (3D-LIN)*. The network is configurable in both 2D and 3D-domains and is automatically split between the chosen number of memory layers. In order to reduce the chip cost, regardless of the number of memory layers needed, they all have the same layout and can all be produced exploiting the same mask. Design automation and configuration of the network allow us to experiment with different 3D structures, in the search for the trade-off points between speed, footprint and number of layers. Thus, the main contribution of this paper is the exploration of various 3D structures for multi-processing, while taking into account the interconnect properties. In the following section, related research efforts are presented. The 2D-LIN is presented in Section III, while Section IV describes the 3D implementation. In Section V, experimental results are shown. Finally, Section VI concludes the presented work.

978-1-4673-2658-2/12/$31.00 ©2012 IEEE

II. Related Work

In the last few years, several studies have been published exploring 3D integration technology in order to address the high area overhead of SRAM. A proposal from Li et al. [8], focuses on the L2 cache design and management in a 3D chip. They propose a network architecture embedded into the L2 NUCA cache memory for connecting it to a collection of cores. A different approach is followed by Loh, that in [9] considers 3D-DRAM stacked on top of multi-processors and revises the memory system organization in a 3D context. More recently, also Woo et al. [10], have explored a memory architecture that exploits TSVs for connecting the last level cache to the 3D stacked DRAM. The work of Madan et al. [11] instead, takes in consideration a 3D system composed by a DRAM layer and an SRAM cache banks layer on top of a processing layer. Considering emerging memory technologies, Mishra et al. [12] study the integration of STT-RAM in a multi-core system, together with a network level solution for decreasing the write latency associated with these novel memories.

In this paper, we propose a 3D structure for connecting a cluster of processing elements, placed on a logic layer, to multiple layers of SRAM modules. These modules constitute a single shared L1 memory that can enable fast communication among the tightly coupled processing elements avoiding cache coherence overheads.

III. 2D Network

The basic 2D-LIN is a low-latency and flexible crossbar that connects multiple *processing elements (PEs)* to multiple SRAM *memory modules (MMs)*. The IP is designed and optimized for sustaining full bandwidth and supporting non-blocking communication within a single clock cycle. It also provides simple and fast inter-processors communication and multi-core synchronization. The key property of this soft IP is the reconfigurability: the user has control on a number of parameters, such as number of master ports and slave ports, type of address decoding and several others. In order for the design to be simple and efficient, the interconnect is built following the Mesh Of Trees approach, where the network is created combining binary trees. Each tree provides a unique combinational path between the processing element cluster and one memory module, and viceversa. Aiming to sustain non blocking communication, the request and the response path must be decoupled, hence 2D-LIN features independent request and response network.

A. Network Architecture Protocol

A memory access starts with a request issued by a PE through a master port, then, the master is kept updated on the status of the request by a simple and lean protocol based on a credit based flow control. Each clock cycle, all the requests made from PEs are propagated through the binary trees. Collisions due to multiple requests directed to the same memory bank are avoided by Round Robin arbitration performed at each node. The processors losing the arbitration are stalled. The PE winning the arbitration concludes the transfer in a single clock cycle in case of a store, while, in case of a load, the read data is returned the next clock cycle.

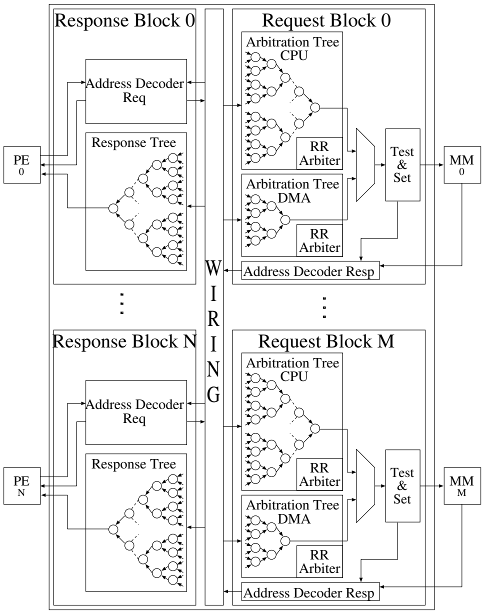


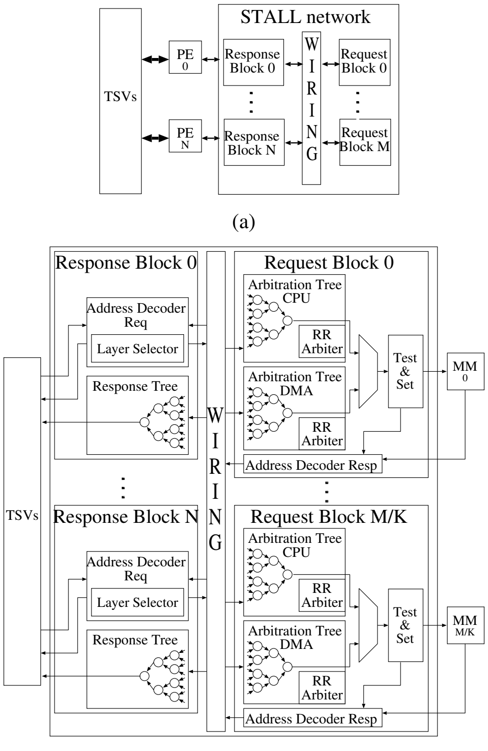
Fig. 1. Block schematic of the 2D-LIN

B. Request block

The request block is in charge of collecting all the PE's requests directed to a specific memory module (see Figure 1). In the simplest case of two PEs, the block is built out of a single binary tree where the request block is composed of 1 node, being a routing-arbitration primitive. The number of stages of the Arbitration Tree is a function of the number of masters attached to it: NUM_{stage}=$\log_2$(N), N being the number of PEs. Combining several binary trees, the network can support both generic number of ports and different priorities. Hence, a high priority channel for the processors and a low priority channel for eventual peripherals can be implemented. The primitives composing the request block first arbitrate among eventual requests through a Round Robin policy, then the winning one is routed to the MM in a combinational way. At the same time, the flow control signals traveling from MMs to PEs, are also managed. Both normal read/write operation and atomic test and set are supported.

C. Response block

The response block (see Figure 1) is in charge of collecting all the responses from memory modules which are directed to a specific processing element, therefore, it can be considered as a specular version of the request block. Nevertheless, since the response network is only used for read operations and the read latency is deterministic (1 cycle), no response collisions are possible. Hence, the response path does not need any arbitration, and it can be simplified replacing round robin arbiters with simpler decoders.



(a)

(b)

Fig. 2. Block schematic of the 3D-LIN: (a) Logic layer block diagram; (b) Single memory layer block diagram.

IV. 3D Interconnection Network

Within a standard planar(2D) architecture, when more storage capability or more processing power are needed, the network size increases, and the single-cycle communication becomes the limiting factor for the maximum achievable operating frequency. 3D-LIN is the extension of the 2D structure presented in Section III, to be integrated in a 3D-stacked CMP. This network topology allows designers to overcome the limitation in frequency by automatically splitting the 2D floorplan into one logic layer and several memory layers and stacking them one on top of the other. All the power-hungry processing elements are placed on logic layer, close to the heat sink, while the memory banks, are divided among the memory layers. The network is partitioned among the layers in an automated way following the assumption that all the memory layers should have the same identical layout:

- Each layer automatically auto-configures during runtime. This permits to reduce the chip cost and the design effort.
- TSVs from the bottom layer are connected to the lowest metal layer, while the TSVs to the upper layer are connected to the top metal layer.
- The M memory banks are equally divided among K memory layers, where K is a power of 2. Each memory layer contains M_L=M/K memory banks.

The request network path (PE to MM), and the response path (MM to PE), have different latencies that depends on the number of levels of the trees. The first strongly depends on the number of PEs, while the second is related to the number of MMs (see Table I). When connecting the memory banks, the access time to read the data from the memory is added to the latency of the response path. 3D-LIN allows us to decrease the number of arbitration levels of the response tree when implemented on 2 or more memory layers, hence it allows the system to run at higher frequencies.

TABLE I
3D-LIN vs. 2D-LIN

	2D-LIN	**3D-LIN**
Number of levels Response Tree	$log_2 M$	$log_2 \frac{M}{K}$
Number of levels Arbitration Tree	$log_2 N$	$log_2 N$
Number of primitives on each memory layer - Response Tree	$\sum_{i=1}^{log_2 M} \frac{M}{2^i} \times N$	$\sum_{i=1}^{log_2 \frac{M}{K}} \frac{\frac{M}{K}}{2^i} \times N$
Number of primitives on each memory layer - Arbitration Tree	$\sum_{i=1}^{log_2 N} M \times \frac{N}{2^i}$	$\sum_{i=1}^{log_2 N} \frac{M}{K} \times \frac{N}{2^i}$
Number of primitives in the system - Response Tree	$\sum_{i=1}^{log_2 M} \frac{M}{2^i} \times N$	$\sum_{j=1}^{K} \sum_{i=1}^{log_2 \frac{M}{K}} \frac{\frac{M}{K}}{2^i} \times N$
Number of primitives in the system - Arbitration Tree	$\sum_{i=1}^{log_2 N} M \times \frac{N}{2^i}$	$\sum_{j=1}^{K} \sum_{i=1}^{log_2 N} \frac{M}{K} \times \frac{N}{2^i}$

A. Network Architecture

TSVs connecting the stacked dies have good electrical characteristics, but their area footprint is bigger compared to the on-chip metal lines. For this reason it is important to place the minimum number of TSVs, while still guaranteeing the maximum possible bandwidth. When the signals traversing the tiers are the direct input and output of the processor, is possible to place the minimum number of TSVs dedicated to signal propagation:

$$TSV = (Nc + 1 + log_2 K) + \\ N(Nb_{addr} + 2Nb_{data} + Nb_{byteEN} + 2) \quad (1)$$

where Nc is the number of TSVs for clock propagation, summed to one TSV for the reset signal, $\log_2$K is the number of bits needed for the layer ID. Nb_{addr}, Nb_{data} and Nb_{byteEN} are respectively the number of TSVs for propagating the address, the data and the byte enable signals. The maximum bandwidth of the 2D system is:

$$BW_{max} = f(\frac{Nb_{data}}{8})K \quad (2)$$

Hence, the PEs and the small Network for the stall (see Figure 2(a)) are placed on the logic layer, while each memory layer has the same layout and contains a Network of cardinality N×$\frac{M}{K}$ and $\frac{M}{K}$ memory banks (see Figure 2(b)). This configuration that minimize the number of TSVs needed for

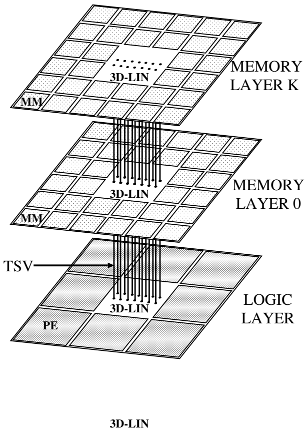


Fig. 3. 3D chip architecture.

the signals, still guarantee BW$_{max}$ also for the 3D implementation. The layerID signal is sent from the logic layer to identify each memory layer, so that the address space is equally divided between all the MMs. Each memory layer takes the incoming layerID as its own identifier, and send to the next mem layer the received signal incremented by one. In the TSV count, the Stall signal is not taken in account. In the 2D network, the Stall signal is critical, because it needs to be asserted much in advance with respect to the next clock rising edge. Hence, in order to optimize it, the logic that computes the Stall signals is detached from the main Network connecting PEs to MMs and placed on the logic layer as a small independent Network.

B. Network Operation

During a read/write operation, the master asserts data and control signals that are sent as a packet. Some control signals goes to the Stall Network that arbitrates possible collision and eventually sends the Stall signal to the PE within the same clock cycle. The full packet, data and control signals, are also sent through the TSVs to the memory layers. Each memory layer receives the packet and checks if the request is for a position in its address range. The layer containing the address lets the packet enter, while the other layers invalidate the request. When a packet accesses the memory layer containing the requested address, the network routes and arbitrates the packet among the other simultaneous requests, allowing the higher priority request to access the memory bank. Write operations are performed in the same clock cycle, while for Read operation and Test and Set operations, the read data is propagated back to the related PE in the next clock cycle.

V. Experimental results

This section provides the evaluation of 3D-LIN in terms of delay and area. The Network is implemented in SystemVerilog and synthesized with Synopsys Design Compiler in topographical mode using 65nm CMOS technology library from ST-Microelectronics. The physical synthesis has been chosen to extract the results because it allows the user to floorplan the design and accurately predict post-layout timing using real net capacitances during RTL synthesis [13]. The functionality has been verified using Mentor Graphics' Modelsim.

In this experiment we considered 5μm wide TSV with 10μm minimum pitch and a length of 50μm, which represents the state-of-the-art for high density through silicon vias [14]. According to the chosen dimensions, the TSV's parasitic capacitance have been obtained through the analytical model proposed by Kim, [15]. For the experiments, the parasitics values have been rounded to 20mΩ for the resistance and 30fF for the capacitance.

The memory size depends on the multi-core application. For the experiments, we chose a case study with memory modules chosen to be SRAM banks of 8kB, which timing and physical information are provided by the lib file and the Milkyway database. Each MM occupy 0.06mm^2. Regarding the processing elements, dummy hard macros are used in order to emulate their area occupation. Each PE is considered to be an ARM CortexM3, which the estimated area is around 0.07mm^2 for 65nm technology.

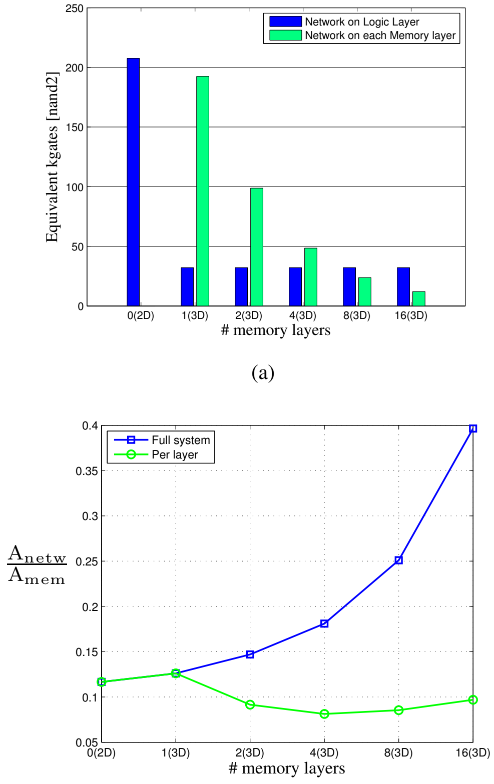
Unfortunately, the current version of Synopsys DC does not support TSV and 3D stacking, hence, in the absence of established design kits, the synthesis flow is performed in several main steps. Starting from the synthesizable RTL description of the network, already configured with the user constraints, the floorplanning of memory layer is performed, and the time and physical constraints are added to emulate the TSVs. After the physical synthesis of the memory layer, the back-annotated delays are used to perform the physical synthesis of the logic layer. After the floorplan definition, the logic layer is synthesized considering the latencies of the stacked dies. These steps are then iterated to meet the desired timing constraints for the complete 3D-stacked system.

Figure 3 depicts the chosen approach for the block placement.

A. Physical Analysis

As explained in Section IV, when moving to a 3D configuration, the original NxM network is divided among the layers: a small NxM network for the Stall signal is placed on the logic layer, while the rest of the network to communicate with the memory banks is distributed on each memory layer as Nx$\frac{M}{K}$ networks. Keeping the number of memory banks fixed to 64, the total cell area of the network on each layer is shown in Figure 4(a) for 16 PEs. Figure 4(b) shows the trend of the ratio between the network area and the memory area both per layer and in the full 3D system. When moving from a planar design to a stacked system, the sum of the network areas on each layer is higher than the 2D counterpart, nevertheless the area per layer decreases.

The configurability of the Network gives the possibility to explore the form-factor trend for the 3D multi-core systems with shared L1 memory on top of logic. Given the specification of the system, the best trade-off can be found in terms of number of layers. In particular, we chose to analyze the area of the chip(A$_{3D}$) normalized to the area of the same chip



(a)

(b)

Fig. 4. (a) Area of the Stall/Valid Network on the logic layer (blue) and area of the data Network on each memory layer (green) for different number memory layers stacked on top of the logic layer; (b) Area of the network over the area of the memory for each memory layer(green), and for the whole system(blue)

implemented on a single silicon layer(A_{2D}) for the following configurations and area occupation of the memory(A_{mem}) over the area of the planar chip(A_{2Dchip}):

- 16 PEs and 16 MMs : $\frac{A_{mem}}{A_{2Dchip}}$=43% ;
- 16 PEs and 32 MMs : $\frac{A_{mem}}{A_{2Dchip}}$=58%;
- 16 PEs and 64 MMs : $\frac{A_{mem}}{A_{2Dchip}}$=70% ;
- 16 PEs and 128 MMs : $\frac{A_{mem}}{A_{2Dchip}}$=79% .

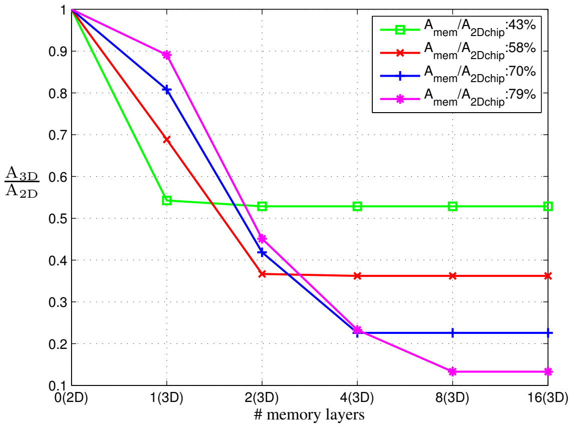
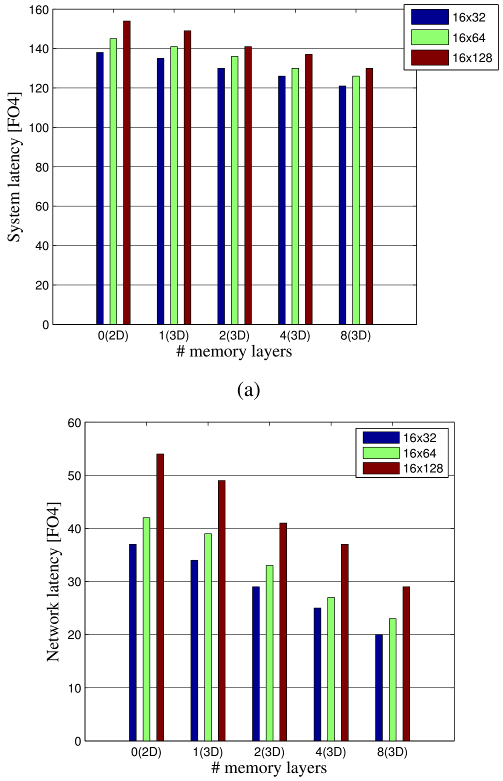


Fig. 5. Area of the 3D chip normalized to the area of the 2D implementation



(a)

(b)

Fig. 6. (a) System latency: Network delay plus memory access time; (b) Network latency.

Figure 5 depicts the reduction of the area when the chip is designed to stack different numbers of memory layers on top of the logic layer. When moving from the planar structure, to a 2-layer structure, the memories and the network are moved to the upper layer, and we can notice a decrease in the form factor. However, this reduction is still limited due to the size of the network that, as explained before, does not shrink effectively. In additions, the TSV area occupation increases the size of both layers. Considering the stacking of two or more layers on top of the logic, the network cardinality start changing depending on the number of memory layers, leading to a decrease in its area occupation, while the TSV occupation remains the same as for the 3D, single memory layer, case. The best trade-off point can be found when the area of the memory layer is almost equal to the area of the logic layer. When reaching the best trade-off, the stacking of any more memory layers does not affect the form factor that is now defined from the area of the logic layer.

B. Timing Analysis

Exploring 3D-LIN in term of latency the following configurations are considered:

- 16 PEs and 32 MMs;

TABLE II
LATENCY IMPROVEMENT

	16x32		**16x64**		**16x128**	
	system	network	system	network	system	network
1 memory layer	2%	9%	2%	7%	3%	10%
2 memory layers	6%	22%	6%	20%	8%	24%
4 memory layers	8%	32%	10%	35%	11%	31%
8 memory layers	12%	46%	13%	44%	16%	46%

- 16 PEs and 64 MMs;
- 16 PEs and 128 MMs.

As previously discussed, the frequency of the network is limited by the response path that includes the access time to read a data from the memory bank. However, depending on the size of the memory module, this access time changes. In our experiments, we explored the latency of the network when connecting memory banks of 8kB. In Figure 6(a) and 6(b), both system latency and network latency are shown. We can notice that moving from the planar system to one stacked memory layer, the latency slightly decreases due to the shorter interconnect. The reduction in delay is more evident for the systems with two or more memory layers, due to the changes in the network topology. The reduction in delay is more evident in Figure 6(b) considering the network itself, independently from the attached memory banks. The latency of the network shows significant improvement, in the case of 16PEs connected to 64MMs, the 2D latency of ˜42FO4 is reduce down to ˜23FO4 .
Table II shows the latency improvements in percentage. The results show that stacking a single memory layer, the memory access time dominates the decreased latency of the interconnect and the improvement is only a few percents. However, when we move to two memory layers, we can obtain already around 8% improvement, reaching 11% with four memory layers for a network cardinality of 16x128. Independently from the attached memory, considering the network alone, the benefits are more evident, with 35% improvements for four memory layers stacked on top of the logic layer.

VI. CONCLUSION

In this paper, we present a configurable network architecture that can be integrated in 3D stacked CMP. The network enable the connection of multiple processing elements to a shared multi-banked memory guaranteeing low-latency connection. The network and the multi processor system has been explored in terms of area, form factor and latency. The physical synthesis results show the best trade off point between the amount of memory needed in the system and the number of stacked layers. In case of a memory occupation of 60% of the planar chip, by moving to a system that integrates two memory layers on top of a logic layer, the form factor is improved more than 60%. In terms of latency, the 16x128 configuration of the network can be improved up to around 24% in case of 2 memory layers, and 31% in case of four memory layers, leading to a latency reduction for accessing 8kB memory banks of 8% and 11% respectively.

ACKNOWLEDGMENT

This work has been partially supported by the EU project grant PRO3D FP7-ICT-248776

REFERENCES

[1] J. Owens, W. Dally, R. Ho, D. Jayasimha, S. Keckler, and L.-S. Peh, "Research challenges for on-chip interconnection networks," *Micro, IEEE*, vol. 27, no. 5, pp. 96 –108, sept.-oct. 2007.

[2] S. Borkar and A. A. Chien, "The future of microprocessors," *Commun. ACM*, vol. 54, pp. 67–77, May 2011.

[3] L. Benini and G. De Micheli, "Networks on chips: a new soc paradigm," *Computer*, vol. 35, no. 1, pp. 70 –78, jan 2002.

[4] A. Balkan, G. Qu, and U. Vishkin, "A mesh-of-trees interconnection network for single-chip parallel processing," in *Application-specific Systems, Architectures and Processors, 2006. ASAP '06. International Conference on*, sept. 2006, pp. 73 –80.

[5] P. Ltd., "The hypercore architecture," in *White Paper*, January 2010.

[6] A. Rahimi, I. Loi, M. Kakoee, and L. Benini, "A fully-synthesizable single-cycle interconnection network for shared-l1 processor clusters," in *Design, Automation Test in Europe Conference Exhibition (DATE), 2011*, march 2011, pp. 1 –6.

[7] Y. Xie, "Processor architecture design using 3d integration technology," *VLSI Design, International Conference on*, vol. 0, pp. 446–451, 2010.

[8] F. Li, C. Nicopoulos, T. Richardson, Y. Xie, V. Narayanan, and M. Kandemir, "Design and management of 3d chip multiprocessors using network-in-memory," *SIGARCH Comput. Archit. News*, vol. 34, pp. 130–141, May 2006.

[9] G. H. Loh, "3d-stacked memory architectures for multi-core processors," *SIGARCH Comput. Archit. News*, vol. 36, pp. 453–464, June 2008.

[10] D. H. Woo, N. H. Seong, D. Lewis, and H.-H. Lee, "An optimized 3d-stacked memory architecture by exploiting excessive, high-density tsv bandwidth," in *High Performance Computer Architecture (HPCA), 2010 IEEE 16th International Symposium on*, jan. 2010, pp. 1 –12.

[11] N. Madan, L. Zhao, N. Muralimanohar, A. Udipi, R. Balasubramonian, R. Iyer, S. Makineni, and D. Newell, "Optimizing communication and capacity in a 3d stacked reconfigurable cache hierarchy," in *High Performance Computer Architecture, 2009. HPCA 2009. IEEE 15th International Symposium on*, feb. 2009, pp. 262 –274.

[12] A. K. Mishra, X. Dong, G. Sun, Y. Xie, N. Vijaykrishnan, and C. R. Das, "Architecting on-chip interconnects for stacked 3d stt-ram caches in cmps," in *Proceeding of the 38th annual international symposium on Computer architecture*, ser. ISCA '11. New York, NY, USA: ACM, 2011, pp. 69–80.

[13] *Design Compiler®User Guide*, Synopsys, December 2011, version F-2011.09-SP2.

[14] G. Van der Plas, P. Limaye, A. Mercha, H. Oprins, C. Torregiani, S. Thijs, D. Linten, M. Stucchi, K. Guruprasad, D. Velenis, D. Shinichi, V. Cherman, B. Vandevelde, V. Simons, I. De Wolf, R. Labie, D. Perry, S. Bronckers, N. Minas, M. Cupac, W. Ruythooren, J. Van Olmen, A. Phommahaxay, M. de Potter de ten Broeck, A. Opdebeeck, M. Rakowski, B. De Wachter, M. Dehan, M. Nelis, R. Agarwal, W. Dehaene, Y. Travaly, P. Marchal, and E. Beyne, "Design issues and considerations for low-cost 3d tsv ic technology," in *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2010 IEEE International*, feb. 2010, pp. 148 –149.

[15] D. H. Kim, S. Mukhopadhyay, and S. K. Lim, "Fast and accurate analytical modeling of through-silicon-via capacitive coupling," *Components, Packaging and Manufacturing Technology, IEEE Transactions on*, vol. 1, no. 2, pp. 168 –180, feb. 2011.