Configurable Low-Latency Interconnect for Multi-core Clusters

Giulia Beanato, Igor Loi, Giovanni Micheli, Yusuf Leblebici, Luca Benini

Configurable Low-Latency Interconnect for Multi-Core Clusters

Giulia Beanato[1], Igor Loi[2],
Giovanni De Micheli[1], Yusuf Leblebici[1], and Luca Benini[2]

[1] EPFL, Lausanne, Switzerland,
[2] DEIS, University of Bologna, Bologna, Italy
{giulia.beanato,giovanni.demicheli,yusuf.leblebici}@epfl.ch
{igor.loi,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 chapter 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, evaluating the improvement in both form factor and latency.

Keywords: 3D integration, multi-core processor, shared memory, interconnection network.

1 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 avoiding slow off-chip connections.

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.[13], 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 order to connect memory and logic placed on different layers, several groups already explored a methodology to extend NoC design into a 3D setting. The simple extension of traditional NoC fabrics to the third dimension adding routers at each layer (Symmetric NoC), does not pay in performance due to the different delay between fast vertical TSV and the horizontal interconnects. A first proposal has been done by Li et al. [13], with a network architecture embedded into the L2 cache memory. The use of Time-Division Multiple Access (dTDMA) buses as Communication Pillars between the wafers is proposed in order to have single-hop communication amongst the layers. The 3D Dimensionally-Decomposed(DimDe) Router [14], focus on optimizing of the inter-strata com-

munication with single hop connection between any two layers. Park et al. [15] propose a Multi-layered on-chip Interconnect Router Architecture (MIRA) divides the NoC between the multiple layers optimizing the micro-architecture for Non Uniform Cache Architecture (NUCA)-based CMP. A Low-Radix Low-Diameter 3D Interconnection Network is proposed by Xu et al. [16] which adopts long wires to connect remote intra-layer nodes and results in a 3 hops diameter network. More recently, Xue et al. [17] uses long range links to replace multiple short links in order to build a 5 hops 3D interconnection network for many core processors that exploits the DimDe router. While Ben Ahmed et al.[18] focus on overcoming the limitations in power, communication cost and throughput of their 2D OASIS-NoC by extending it to 3D.
This chapter aims to propose a fully synthesizable *3D Logarithmic Interconnection Network (3D-LIN)* 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. 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.

2 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. These features makes LIN an interesting option for interfacing multi-processors to a shared Tight Coupled Data Memory (TCDM) constituted by multiple equal memory banks. This topology permits to avoid data replication providing also a simple and fast way for inter-processors communication and multi-core synchronization. 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. The key property of this soft IP is the reconfigurability. The user has control on a number of parameters:

- Number of masters, N, that is a power of two;
- Number of memory cuts, M, that is a power of two. With a number of MMs at least double the numbre of PEs, access collision can be drastically reduced;
- Size of the memory cuts, all the banks should have the same size;
- Data and Address width;

– Enable for word level interleaving, for spreading transactions among all banks drastically reducing access collision.
– Test and Set bit. This bit act as enable for a test-and-set instruction used to write to a memory location and return the old value as a single atomic operation.

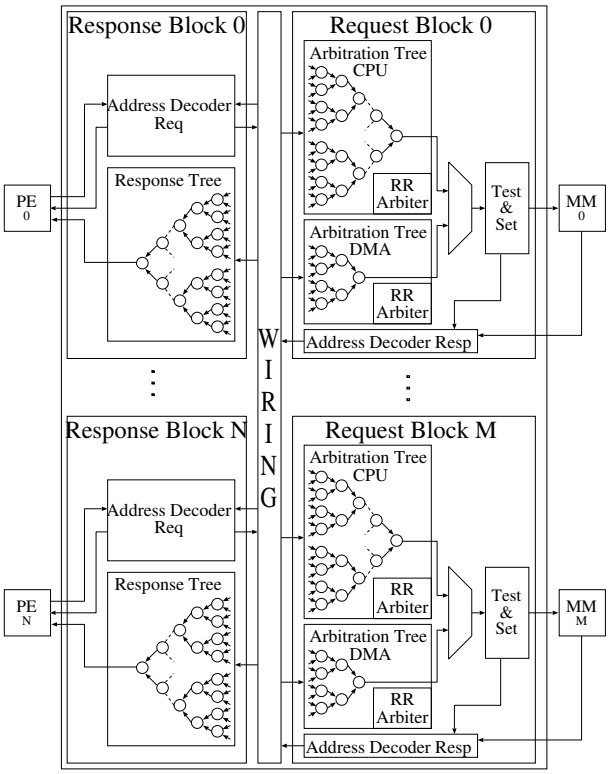

Fig. 1. Block schematic of the 2D-LIN

2.1 Network Architecture Protocol

The network is created by independent and decoupled Request and Response channel. 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.

2.2 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: $\mathrm{NUM}_{stage}=\log_2(\mathrm{N})$, 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 supported. 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.

2.3 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.

3 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 the previous section, 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, Figure 3. 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:

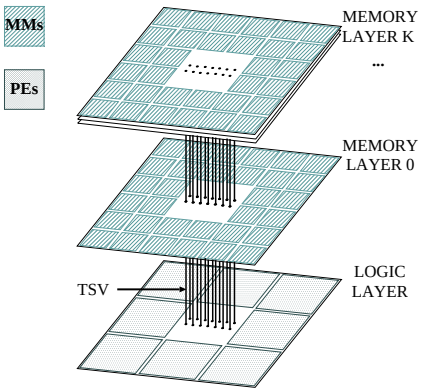

Fig. 2. 3D chip architecture.

- 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.

Table 1 summarize the main parameters of 3D-LIN versus 2D-LIN. We can notice that in terms of 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. The number of levels directly affects the latencies of the request network path (PE to MM), and the response path (MM to PE). 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. The number of primitives per layer and in the system give an estimation on how the area of the network can be reduced by moving to 3D. The main reduction is encountered for the primitives of the Response Tree, but also the Arbitration Tree diminish.

3.1 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, it is possible to place the minimum number of TSVs dedicated to signal propagation:

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

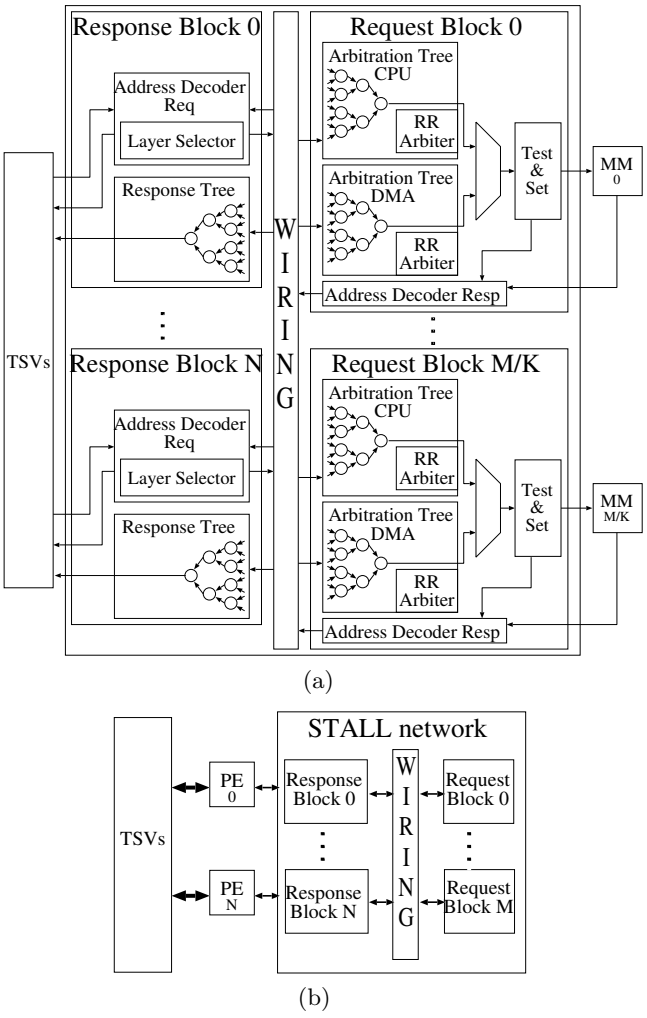

Fig. 3. Block schematic of the 3D-LIN: (a) Logic layer block diagram; (b) Single memory layer block diagram.

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:

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

Hence, the PEs and the small Network for the stall (see Figure 3(b)) are placed on the logic layer, while each memory layer has the same layout and contains a Network of cardinality $\text{N}\times\frac{M}{K}$ and $\frac{M}{K}$ memory banks (see Figure 3(a)). This configuration that minimize the number of TSVs needed for 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.

Table 1. 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	$\sum_{i=1}^{log_2 N} M \times \frac{N}{2^i}$	$\sum_{j=1}^{K} \sum_{i=1}^{log_2 N} \frac{M}{K} \times \frac{N}{2^i}$

3.2 Network Operation

During a read/write operation, the master asserts data and control signals that are sent as a packet. Some control signals go 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 operations and Test and Set operations, the read data is propagated back to the related PE in the next clock cycle.

4 Experimental results

This section provides the evaluation of 3D-LIN in terms of area, power and delay. The Network is implemented in System-Verilog 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 [19]. 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 [20]. According to the chosen dimensions, the TSV's parasitic capacitance have been obtained through the analytical model proposed by Kim,[21]. 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.

4.1 Physical Analysis

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 that communicates with the memory banks is divided in Nx$\frac{M}{K}$ smaller networks distributed on each memory layer. We first explore the impact of the 3D partitioning on the network area, measured as equivalent kgates (nand2), for several systems:

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

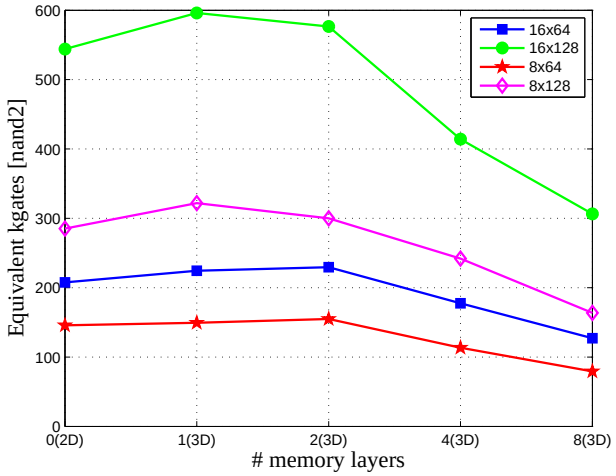

Fig. 4. Area occupied by the network in the 3D system.

Figure 4 depicts the trend of the total area, that is the sum of the area occupied by the partitioned network on each layer, for different network cardinalities. We can notice that for 3D-systems composed of 1 memory layer, the total area has a slight increase. This is due to the fact that moving from a 2D-system to a 3D-system, the small stall network is added on the logic layer. Once we reach 3 or more layers, even if the network is replicated on each memory layer, the area reduction per layer dominates. Since the total number of primitives constituting

3D-LIN is equal to $\sum_{j=1}^{K} \sum_{i=1}^{log_2 \frac{M}{K}} \frac{\frac{M}{K}}{2^i} \times N + \sum_{j=1}^{K} \sum_{i=1}^{log_2 N} \frac{M}{K} \times \frac{N}{2^i}$, is expected that the area reduction is more accentuated for networks connecting a higher number of MMs.

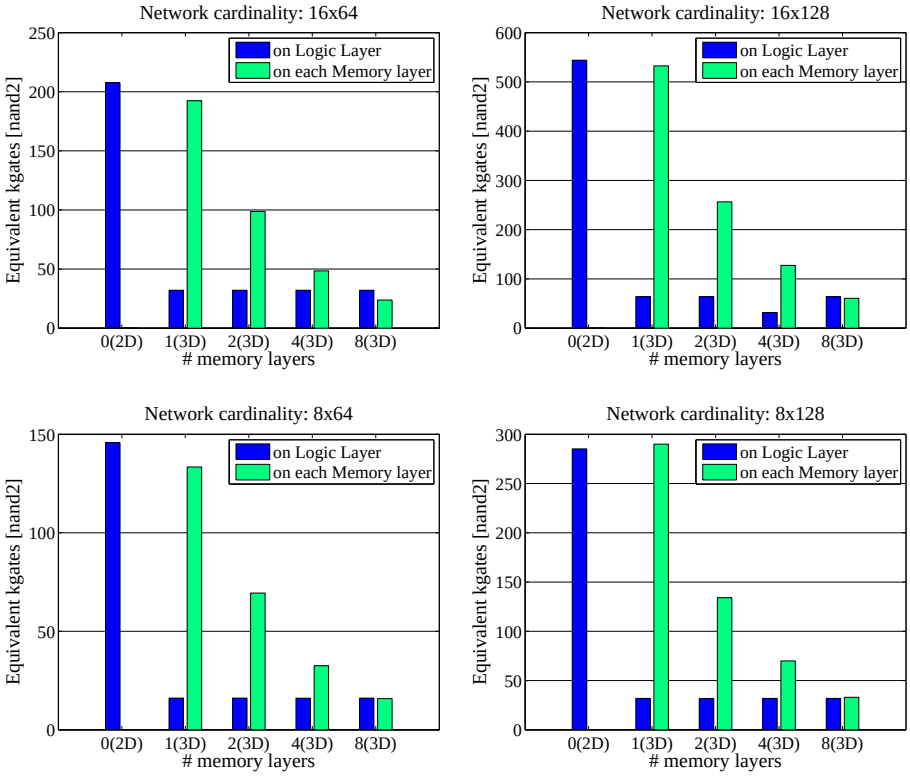

Fig. 5. Area of the Stall/Valid Network on the logic layer (blue) and area of the data Network on each memory layer (green) for different number memory layers stacked on top of the logic layer.

In a 3D system, however, is important to consider the per-layer reduction, since the form factor is influenced by the single layers dimension. The area occupied by the network on the logic layer and the ones on each memory layer is shown in Figure 4.1. Once adding more memory layers, there is a strong decrease in the per-layer network area.
Figure 6 shows the trend of the ratio between the network area and the memory area both per layer and in the full 3D system composed of 16 PEs interfaced to 64 MMs. When moving from a planar design to a stacked system, the sum of

the ne twork areas on each layer is higher than the 2D counterpart, nevertheless

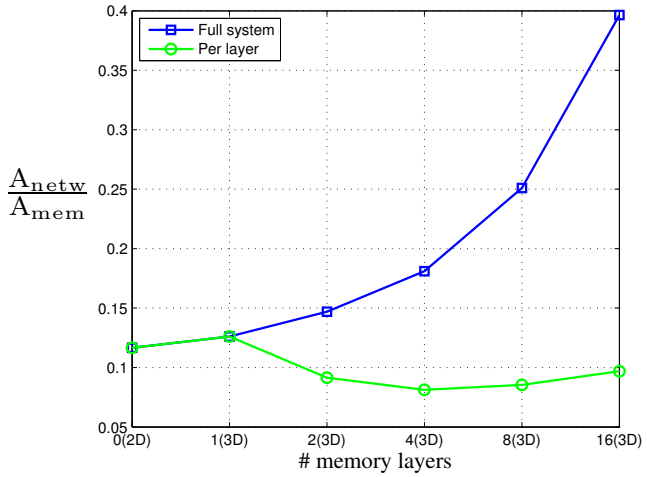

Fig. 6. Area of the network over the area of the memory for each memory layer(green), and for the whole system(blue)

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 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% .

Figure 7 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

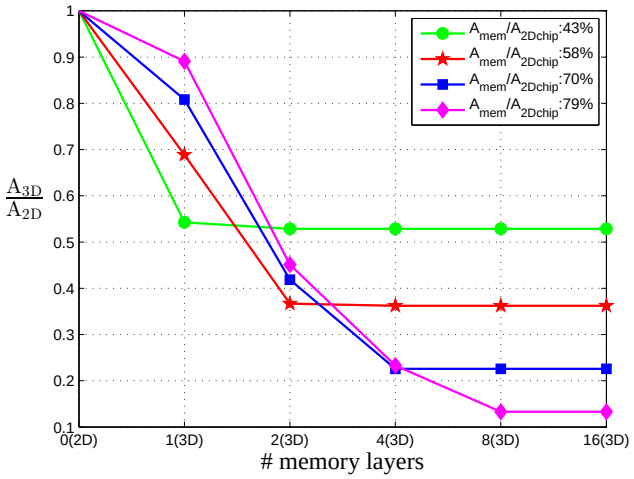

Fig. 7. Area of the 3D chip normalized to the area of the 2D implementation

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.

4.2 Power Analysis

The power consumption is an important parameter to be considered. For 3D-ICs, it is even more important: stacking more layers arise new challenges due to an increased power density per footprint, which may cause temperature to increase beyond the limits that guarantees reliability. At the design level, careful floorplan definition and thermal management techniques such as *dynamic voltage and frequency scaling (DVFS)* can help, but are not sufficient. There is a significant research effort to tackle the power issue at different levels. At the software level thermal-aware task scheduling policies [23] can be implemented, while at the fabrication level, cooling techniques such as inter-layer micro-channel liquid cooling [22] and *Thermal-TSVs(TTSV)* [24], [25] can be exploited to remove the excessive heat.

In this chapter, we do not propose any cooling or thermal management techniques, but we focus on exploring the power dissipation of 3D-LIN to ensure reliability. The total dynamic power consumed by the network is depicted in figure 8. We can observe how the trend for power is correlated to the network area. As the number of blocks to be interconnected increases, the size of the die

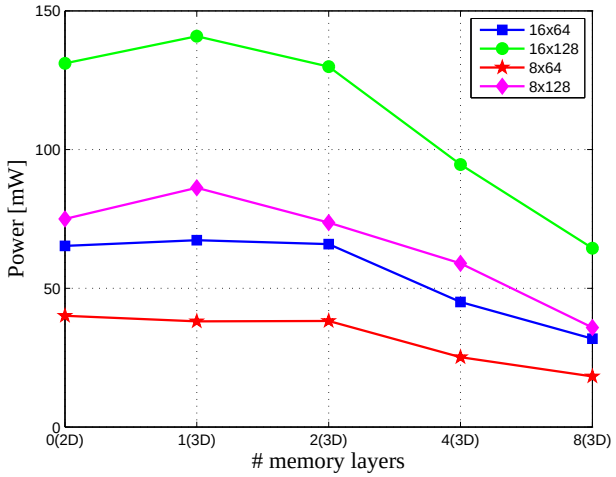

Fig. 8. Total dynamic power consumption of the network in the 3D system.

affect the wire length and the power related to wiring start dominating the cell internal power. Hence, the gain in power consumption is more pronounced for systems with higher cardinality and appears once stacking more memory layers which reduces both the per-layer network cardinality, and the single layer size.

The power contribution of the different single layers is shown in figure 8. The power consumed by the stall network on the logic layer is small compared to the consumption of the network on each memory layer, which is the dominant contribution. As the number of stacked memory layers increases, the cardinality of the network on each layer is reduced, leading to a significant gain in power.

4.3 Timing Analysis

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

- 16 PEs and 32 MMs;
- 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 4.3 and 4.3, 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

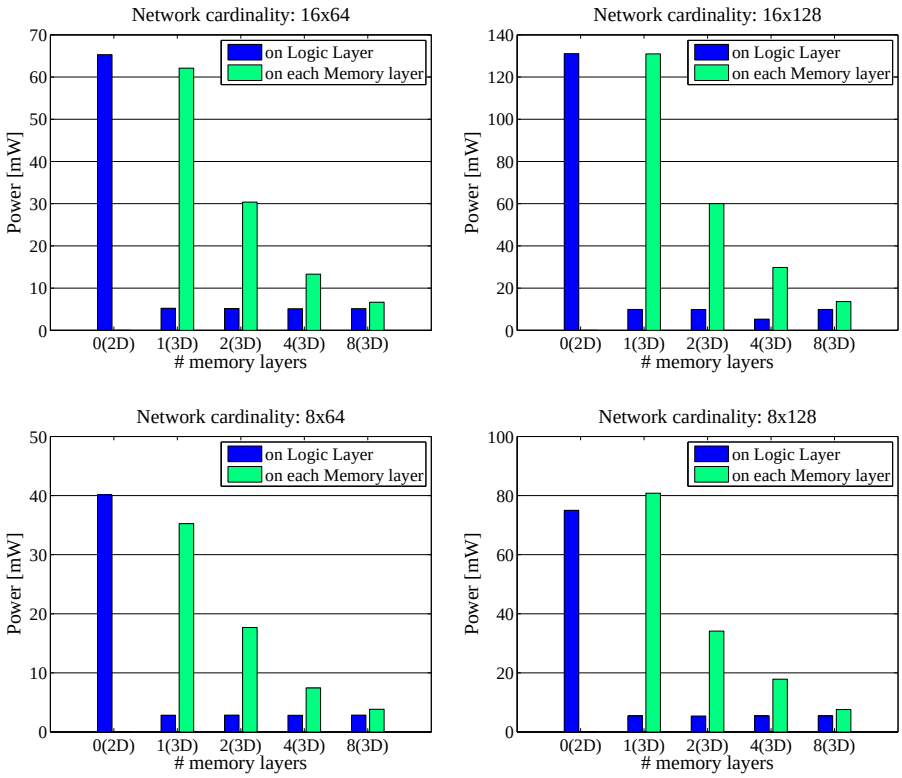

Fig. 9. Dynamic power consumed by the Stall/Valid Network on the logic layer (blue) and dynamic power consumed by the data Network on each memory layer (green) for different number memory layers stacked on top of the logic layer.

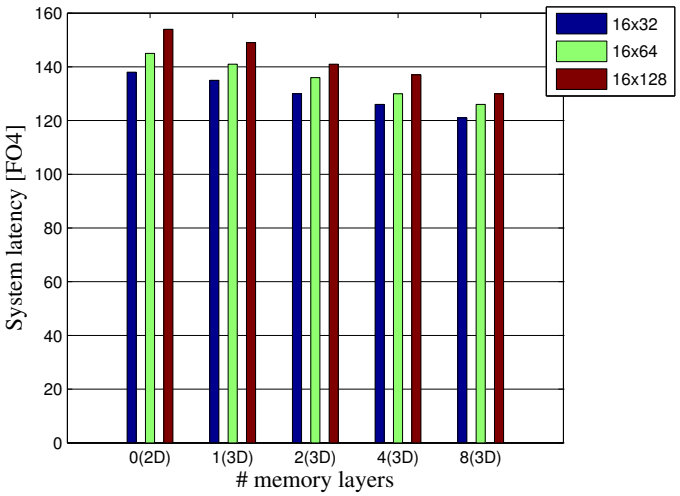

Fig. 10. System latency: Network delay plus memory access time.

or more memory layers, due to the changes in the network topology. The reduction in delay is more evident in Figure 4.3 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 2 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.

Table 2. 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%

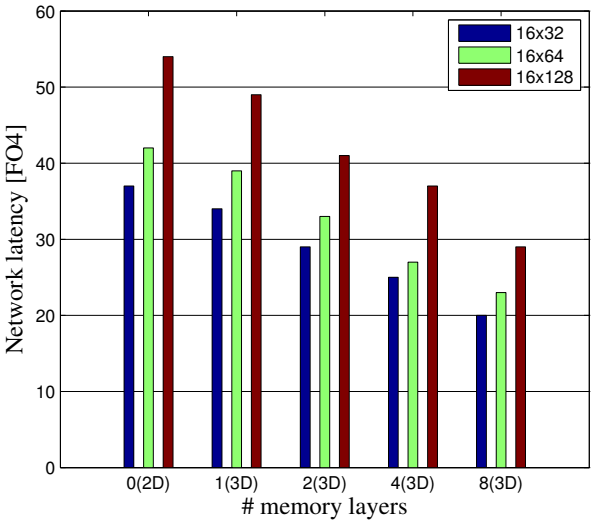

Fig. 11. Network latency.

5 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, power and latency. The benefits obtained by exploiting 3D integration are evaluated. Moreover, the study also focus on exploring the performances for different 3D structures, studying the effects of stacking different number of layers. 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. Latency and area improvements come without a worsening in terms of power. Stacking 2 or 3 layers, the power consumption is kept almost the same as for the 2D implementation, while starts improving as the number of layer increases.

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

References

1. Owens, J.D., Dally, W.J., Ho, R., Jayasimha, D.N., Keckler, S.W., Peh., L.-S. : Research challenges for on-chip interconnection networks. J. IEEE Micro 27, 96 108 (2007)
2. Borkar, S., Chien, A. A. : The Future of Microprocessors. J. Commun. ACM 54, 67-77 (2011)
3. Benini, L., De Micheli, G. : Networks on Chips: a New SoC Paradigm. J. Computer 35, 70 -78 (2002)
4. Balkan, A., Qu, G., Vishkin, U. : A Mesh-of-Trees Interconnection Network for Single-Chip Parallel Processing Application-Specific Systems. In: International Conference on Architectures and Processors, pp. 73 -80 (2006)
5. Plurality, Ltd. : The hyperCore architecture. White Paper (2010)
6. Rahimi, A., Loi, I., Kakoee, M., Benini, L.: A fully-synthesizable single-cycle interconnection network for Shared-L1 processor clusters Design. In: Automation Test in Europe Conference, pp. 1 -6 (2011)
7. Xie, Y. : Processor Architecture Design Using 3D Integration Technology. 23rd International Conference on VLSI Design, pp. 446 -451 (2010)
8. Li, F., Nicopoulos, C., Richardson, T., Xie, Y., Narayanan, V., Kandemir, M. : Design and management of 3D chip multiprocessors using network-in-memory. J. SIGARCH Comput. Archit. News 34, 130-141 (2006)
9. Loh, G. : 3D-Stacked memory architectures for multi-core processors. In: Proceedings of the 35th Annual International Symposium on Computer Architecture, pp. 453-464 (2008)
10. Woo, D. H., Seong, N. H., Lewis, D., Lee, H.-H: An Optimized 3D-Stacked Memory Architecture by Exploiting Excessive, High-Density TSV Bandwidth. In: 16th International Symposium on High Performance Computer Architecture, pp. 1 -12 (2010)
11. Madan, N., Zhao, L., Muralimanohar, N., Udipi, A., Balasubramonian, R., Iyer, R., Makineni, S., Newell, D. : Optimizing communication and capacity in a 3D stacked reconfigurable cache hierarchy. In: 15th International Symposium on High Performance Computer Architecture, pp. 262 -274 (2009)
12. Mishra, A., Dong, X., Sun, G., Xie, Y., Vijaykrishnan, N., Das, C.: Architecting on-chip interconnects for stacked 3D STT-RAM caches in CMPs. J. SIGARCH Comput. Archit. News 39, 69-80 (2011)
13. Li, F., Nicopoulos, C., Richardson, T., Xie, Y., Narayanan, V., Kandemir, M.: Design and Management of 3D Chip Multiprocessors Using Network-in-Memory. J. SIGARCH Comput. Archit. News 34, 130-141 (2006)
14. Kim, J., Nicopoulos, C., Park, D., Das, R., Xie, Y., Narayanan, V., Yousif, M., Das, C.: A novel dimensionally-decomposed router for on-chip communication in 3D architectures. In: 34th International symposium on Computer architecture, pp. 138-149 (2007)
15. Park, D., Eachempati, S., Das, R., Mishra, A., Xie, Y., Vijaykrishnan, N., Das, C.: MIRA: A Multi-layered On-Chip Interconnect Router Architecture. In: 35th Annual International Symposium on Computer Architecture, pp. 251-261 (2008)
16. Xu, Y., Du, Y., Zhao, B., Zhou, X., Zhang, Y., Yang, J.: A Low-Radix and Low-Diameter 3D Interconnection Network Design. In: 15th International Symposium on High Performance Computer Architecture, pp. 30-42 (2009)
17. Xue, L., Gao, Y., Fu, J.: A High Performance 3D Interconnection Network for Many-Core Processors. In: 2nd International Conference on Computer Engineering and Technology, pp. 383-389 (2010)

18. Ben Ahmed, A.;, Ben Abdallah, A., Kuroda, K.: Architecture and Design of Efficient 3D Network-on-Chip (3D NoC) for Custom Multicore SoC. In: Broadband, Wireless Computing, Communication and Applications, pp. 67 -73 (2010)
19. Design Compiler User Guide, Synopsys, version F-2011.09-SP2 (2011)
20. Van der Plas, G., Limaye, P., Loi, I., Mercha, A., Oprins, H., Torregiani, C., Thijs, S., Linten, D., Stucchi, M., Katti, G., Velenis, D., Cherman, V., Vandevelde, B., Simons, V., De Wolf, I., Labie, R., Perry, D., Bronckers, S., Minas, N., Cupac, M.; Ruythooren, W., Van Olmen, J., Phommahaxay, A., de Potter de ten Broeck, M., Opdebeeck, A., Rakowski, M., De Wachter, B., Dehan, M., Nelis, M., Agarwal, R., Pullini, A., Angiolini, F., Benini, L., Dehaene, W., Travaly, Y., Beyne, E., Marchal, P.: Design issues and considerations for low-cost 3-D TSV IC technology. J. of Solid-State Circuits 46, 293 -30 (2011)
21. Kim, D. H., Mukhopadhyay, S., Lim, S. K.: Fast and Accurate Analytical Modeling of Through-Silicon-Via Capacitive Coupling. J. IEEE Transactions on Components Packaging and Manufacturing Technology 1, 168180 (2011)
22. Bing Shi; Srivastava, A.: Liquid Cooling for 3D-ICs. In:International Green Computing Conference and Workshops, pp.1,6, 25-28 (2011)
23. Zhou, X., Yang, J., Xu, Y., Zhang, Y., Zhao, J.: Thermal-aware Task Scheduling for 3D Multicore Processors. J. IEEE Trans. Parallel Distrib. Syst. 21, 60-71 (2010)
24. Goplen, B., Sapatnekar, S.: Thermal Via Placement in 3D ICs. In: International Symposium on Physical Design, pp. 167-174 (2005)
25. Yu, H., He, L.: Dynamic Power and Thermal Integrity in 3D Integration. In: Communications, Circuits and Systems, pp. 1108 -1112 (2009)