# High-Performance On-Chip Network Platform for Memory-on-Processor Architectures

Masoud Daneshtalab, Masoumeh Ebrahimi, Pasi Liljeberg, Juha Plosila, Hannu Tenhunen
*Department of Information Technology, University of Turku, Finland*
*{ masdan, masebr, pakrli, juplos, hanten }@utu.fi*

***Abstract*— Three Dimensional Integrated Circuits (3D ICs) are emerging to improve existing Two Dimensional (2D) designs by providing smaller chip areas, higher performance and lower power consumption. Stacking memory layers on top of a multiprocessor layer (logic layer) is a potential solution to reduce wire delay and increase the bandwidth. To fully employ this capability, an efficient on-chip communication platform is required to be integrated in the logic layer. In this paper, we present an on-chip network platform for the logic layer utilizing an efficient network interface to exploit the potential bandwidth of stacked memory-on-processor architectures. Experimental results demonstrate that the platform equipped with the presented network interface increases the performance considerably.**

***Keywords-Three Dimension Integrated Circuit; Network-on-Chip; Memory-on-Processor Architectures; Memory Wall;***

## I. INTRODUCTION

As the number of cores integrated onto a single die increases, the performance of applications will be limited by the memory bandwidth. This increasing number of cores shares the off-chip DRAM bandwidth which will continue to be restricted due to limited number of pins in the processor's package. On top of that, the increasing difference in speeds between the processor and the memory causes the processor to be starved of data [1]. This problem, called the memory wall, is typically relevant to the processor-main memory interface. Therefore, new architectural innovations must be discovered to overcome the memory bandwidth bottleneck.

One promising solution to satisfy the demand for high memory bandwidth is the three-dimensional (3D) stacking, enabling the construction of systems using multiple active silicon layers bonded with low-latency, high-bandwidth and very dense vertical interconnects [3]-[7]. 3D stacking reduces the interconnect delay by stacking vertically active silicon layers. Besides the benefits of interconnect performance, this scheme leads to increase packing density, smaller chip area, lower power dissipation, and provides means to integrate dissimilar technologies in the same chip, but on different active layers, e.g. high speed CMOS with high-density DRAM.

Among various 3D architectures, the 3D stacked memory-on-processor architecture where multiple DRAM layers (memory layer) are directly stacked on top of a multiprocessor layer (logic layer), can satisfy the high memory bandwidth demands that future multiprocessor architectures require [4][7]-[17]. This architecture has gained its popularity because of short processor to memory interconnect delay, best heat dissipation capability because the processor layer can be placed close to the heat sink, and a good scalability in number of layers.

Integrating a large number of cores onto the logic layer is looming as a major performance bottleneck. Networks-on-Chip (NoC) has emerged as a solution to address the communication demands of processors in the logic layer due to its reusability, scalability, and parallelism in communication infrastructure [18][19][20].

In this paper, we present an efficient on-chip network platform for the logic layer in the realm of 3D stacked memory-on-processor architectures. We design a modular communication platform for the logic layer to scale the bandwidth among the processors. In addition, the on-chip network is equipped with a streamlined network interface to provide an efficient communication between the processor and memories. Unlike the other network interfaces, requests for local memory will not have to travel through the on-chip network.

The paper is organized as follows. In Section II, the preliminaries are discussed. In Section III, a brief review of related works is presented while the logic layer architecture is presented in Section IV. The experimental results are discussed in Section V with the summary and conclusion given in the last section.

## II. 3D STACKED MEMORY-ON-PROCESSOR ARCHITECTURE

Stacking DRAM wafers (memory layers) on top of processors wafer (logic layer) is a promising approach to overcome the Memory Wall problem [7]-[10]. Since the storage density of DRAM is much higher than SRAM, it is reasonable to stack multiple on-chip DRAMs on top of the processors in addition to the main memory present on the board. Fig. 1(a) shows a conventional DRAM stacking where all DRAM memories (on one layer) are stacked on top of the

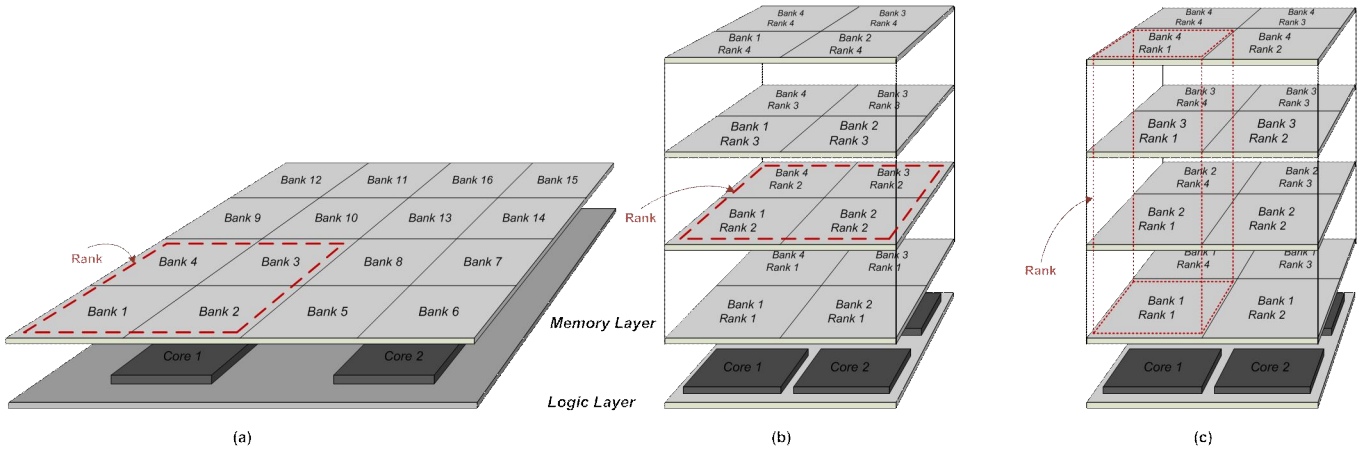


Fig. 1. 3D-stacked DRAM on logic layer with (a) the whole memory on one layer (b) one rank per layer and (c) ranks split across multiple layers.

logic layer. This structure has advantage if the size of the memories is small. As depicted in Fig. 1(b), one possible memory stacking, named *planar* 3D DRAM, is using Through Silicon Vias (TSVs) to implement a vertical bus across multiple DRAM layers to link them to the logic layer [11][15]-[16].

The TSV interconnections are short, fast, and dense allowing a very high inter-layer bandwidth that cannot be provided by other technologies (e.g. wire bonding, micro-bump, contactless), therefore, TSVs are the most promising one among these vertical interconnect technologies [4]-[7]. The pitches of a TSV can range from 1μm to 10μm square [15] while state-of-the-art TSV manufacturing will be able to produce approximately a pitch of 4mμ in 2011 [8]. That is, several millions of TSVs can be implemented in one square centimeter, thereby the size and latency of TSVs will not be a limiting factor [4][7]-[13].

Fig. 1(c) depicts *stacked* 3D DRAM where the main memory is divided into 4 banks. Each bank is aligned to one core slice. The four banks stacked directly atop a core compose a rank, which will be used most frequently by the core underneath. The peripheral circuits of banks are projected to adjoining positions on the interface layer. Therefore, on top of each core, there are 4 banks / columns, or 1 rank. The granularity of data array partition is limited by the area of interface layer: smaller banks lead to larger bank count, which means that more area needs to be reserved for peripherals on the interface layer resulting in a more complex layout and lower area efficiency. Although higher capacity can fit into our chip area, we use a rank capacity of 128MB (32MB per bank), for three reasons: (1) 128MB/rank leads to a 2GB main memory, which is enough for most of applications; (2) large memory capacity will result in large peripheral logic area which is constrained by the available area in the interface layer; (3) enough area needs to be reserved on each cell layer for wordline and bitline TSVs.

Unlike the conventional 3D memory, all peripheral logic of the *stacked* 3D memory, including the row decoder, sense amplifiers, row buffers and output drivers, are placed on the logic layer which connects with the memory layers (banks) through TSVs. Separating the logic layer from the memory layers allows mixing heterogeneous wafers from different process technologies. The memory layers are implemented using high-density NMOS process technology optimized to create high-quality capacitors and low-leakage transistors whereas the logic layer is implemented with CMOS process technology optimized for speed [7][9][12].

## III. Related Work

Many proposed architectures simply consider the 3D memory to be another level in the cache hierarchy [5][10], but some recent studies have already started exploring the benefits of using 3D integration to stack the main memory on top of a processor layer [7][9][11][16]. These studies report impressive performance speedups for 3D-stacked memories. In [7] it is demonstrated that the processor layer could have as many as sixteen layers of DRAM stacked on top of it without exceeding the maximum thermal limit [7]. With this amount of storage, the reduced access latency, and the increased memory bandwidth, 3D stacked DRAM is an excellent candidate for a main system memory in future-generation many-core processors.

Individual memory cell arrays are stacked in a 3D fashion, therefore reducing length of internal buses, wordlines and bitlines, which in turn reduces the access latency of the memory. Much of this speed is due to the process separation described above as well as using short vertical interconnects (TSVs) in place of long horizontal wires allows faster access to all the memory cells in a high-capacity chip. In [15] a 8Gb 3D DDR3 using TSVs to stack 4 DRAM dies is presented. The first die includes DRAM banks, R/W buffers and IO circuits. Read and write buses are independent, but row and column addresses are multiplexed as in the conventional DRAM. As the author remarks, the DRAM module are simply added on each tiers, therefore this results in increased power and area due to duplication of circuit components. A memory interface for 3D stacked DRAM is presented in [4]. The memory interface is integrated inside the processor core to reduce the latency of requests for local stacked memory. The presented

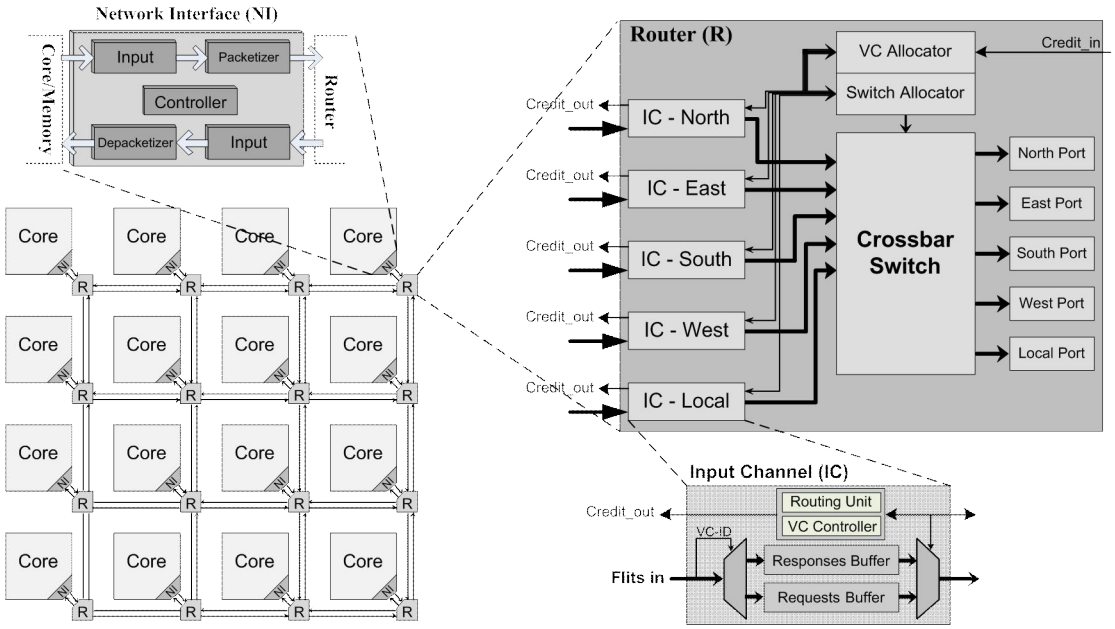


Fig. 2. On-chip inter-core communication platform for the logic layer.

memory interface degrades the performance of non-local requests.

In [2] the authors introduced an SDRAM-aware router to send one of the competing packets toward an SDRAM using a priority-based arbitration. One of the practical approaches of network interfaces is to translate the language between the PE and router based on a standard communication protocol (AXI [21] and OCP [22]) which is not supported by the presented network interface.

The major contribution of this paper is to propose an efficient on-chip communication platform, exploiting an efficient network interface, for the logic layer of the 3D memory-on-processor architecture.

## IV. Logic Layer Communication Platform

As discussed earlier, there are multiple architectural options to integrate memory banks on top of the logic layer in a 3D chip. In this section, we present our logic layer communication architecture for 3D stacked memory-on-processor configurations.

### *A. Communication Plantform*

The inter-core communication in the logic layer can lead to numerous implications for power, performance, and routing area. To minimize power consumed by interconnects we found a 2D mesh network-on-chip. 2D-mesh has many desirable properties for NoCs, including scalability, low cross-section bandwidth, and the fixed degree of nodes [17][10]. A 2D-mesh NoC based system is shown in Fig. 2. As described in literatures, e.g. [23], NoC consists of Routers (R), Cores, and Network Interfaces (NI). Each core is connected to the corresponding router port using the network interface. To be compatible with existing transaction-based cores, we use the AMBA AXI protocol. AMBA AXI is an interfacing protocol, having advanced functions such as a multiple outstanding address function and data interleaving function [21]. AXI can be implemented on NoCs as an interface protocol between each AXI-based core and router to avoid the structural limitations in SoCs due to the bus architecture. In the AXI transaction-based model, processing elements can be classified as master (processor) and slave (memory) [23]. Master cores initiate transactions by issuing read and write requests and one or more slaves (memories) receive and response to each request. The network interface lies between a core and the corresponding attached router. This unit prevents cores from directly interacting with the rest of the network components in the NoC. The architecture of the router, depicted in Fig. 2, has a typical state-of-the-art structure including input buffers, a VC (Virtual Channel) allocator, a routing unit, a switch allocator and a crossbar. Each router has 5 input/output ports, and each input port of the router has 2 VCs. Packets of different message types (request and response) are assigned to corresponding VCs to avoid message dependency deadlock [24]. The arbitration scheme of the switch allocator is round-robin. The round-robin is a fair policy when all packets have the same priority.

### *B. Logic-Layer Network Interface*

Fig. 3 depicts the proposed network interface of each node in the logic layer. It is partitioned into forward path and reverse path. The forward path transmits the AXI transactions received from a processor (or a memory) to a router; and the reverse path receives the packets from the router and converts them back to AXI transactions.

As shown in Fig. 3 the forward path is composed of an AXI-Queue, a Packetizer unit, and a Reorder unit, while the reverse path, receiving the responses from the network, is composed by a Packet-Queue, a Depacketizer unit, a Detector, and the Reorder unit. The Reorder unit is a shared module between the forward and reverse paths. AXI-Queue stores requests/responses in either write or read request/response buffer.

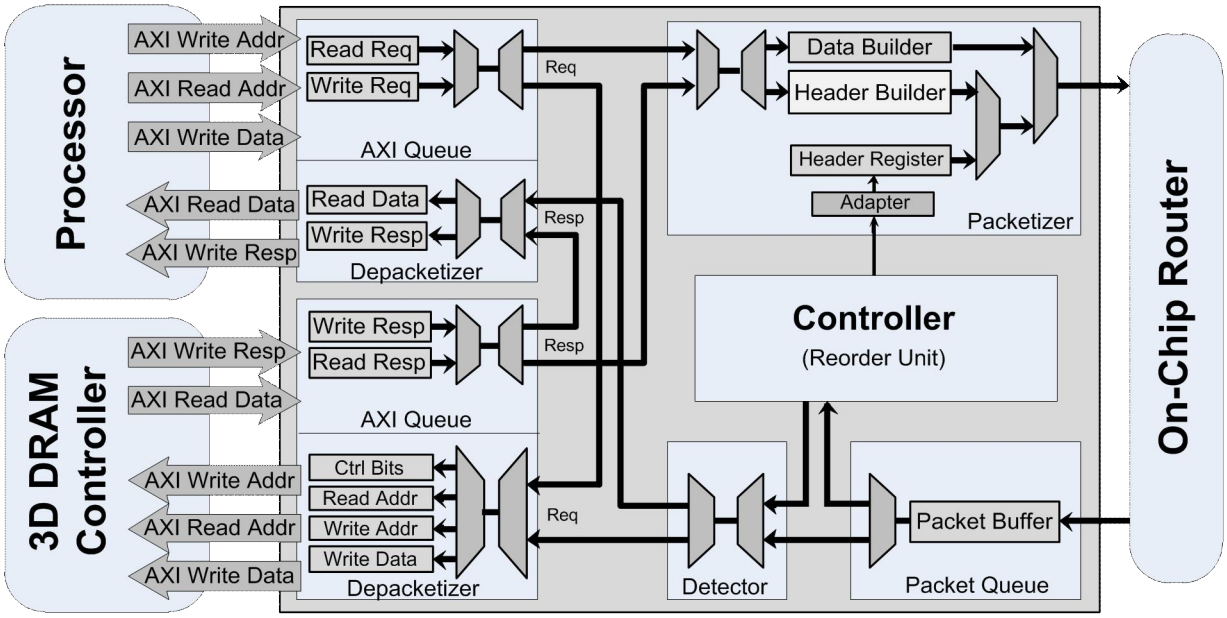


Fig. 3. Logic layer network interface.

The packetizer unit converts incoming messages from the AXI-Queue unit into header and data flits, and delivers the produced flits to the router. The header builder converts the AXI address into a network address by using an address decoder and after receiving the sequence number provided by the reorder unit the header of the packet can be assembled and delivered to the router. Packet-Queue receives packets from the router and according to the decision of the reorder unit a packet is delivered to the depacketizer unit or reorder buffer. In fact, when a new packet arrives, the sequence number and transaction ID of the packet will be sent to the reorder unit. Based on the decision of the reorder unit, if the packet is out of order, it is transmitted to the reorder buffer, and otherwise it will be delivered to the depacketizer unit directly. Based on the type of incoming packet (Req/Resp) the detector unit determines the target unit (memory-side depacketizer/processor-side depacketizer). Depacketizer restores packets coming from either the packet queue or reorder unit into the original data format of the processor or memory. The reorder unit in the forward path prepares a sequence number for corresponding transaction ID. In the reverse path, this unit determines where the outstanding packets from the packet queue should be transmitted (reorder unit or depacketizer), and when the packets in the reorder buffer could be released to the depacketizer unit. Unlike the other network interfaces, since there is a direct channel between the processor and the local memory, requests for local memory do not need to travel through the on-chip network.

## V. Experimental Results

To evaluate the proposed communication platform by measuring the average network latency under different traffic patterns, a cycle-accurate simulator is implemented. The presented on-chip network platform is evaluated with each memory organization.

### A. System Configuration

For the logic layer we use a 16-node (4×4) 2D mesh on-chip network where each node is considered to have a 32b-AXI processor (ARM11MP-16K L1). We assume a 4-layer 3D DRAM, DDR-128MB (32b, 4 banks), is stacked on top of each processor so that in total we have a 2GB-stacked memory containing 16 ranks in four layers. For this 3D memory we consider two different organizations, *planar* 3D DRAM (Fig. 1(b)) and *stacked* 3D DRAM (Fig. 1(c)). Inasmuch as the memory is now stacked on top of the logic layer, the front-side bus and memory controller run at the same speed as the processor. The timing of the *planar* 3D DRAM is still the same as a traditional 2D memory ($t_{CAS}$, $t_{RAS}$, etc. are unchanged) [7][11][16], while the memory access latency for the *stacked* 3D DRAM improves by 32.5% because of the combination of reducing bitline capacitance, using high-speed logic, and exploiting high-speed TSVs [7].

We adopt a commercial memory controller and memory physical interface, DDR2SPA module from Gaisler ip-cores [26] and develop two different network interfaces for the experimental results: Conventional Network Interface (CNI) [4][25] and Logic Layer Network Interface (LLNI). The former is based on the first-come-first-service policy while the latter was described in previous section. The array size, routing algorithm, link width, number of VCs, buffer depth of each VC, and traffic type are the other parameters which must be specified for the simulator. The routers adopt the XY routing algorithm and utilize wormhole switching [18]. For all routers, the data width (flit size) was set to 32 bits, and the buffer depth of each VC to 5 flits. The size of read request messages typically depends on the network size and memory capacity of the system. The message size of the proposed mechanism is variable and depends on the request/response length produced by either a processor or a memory. As the performance metric,

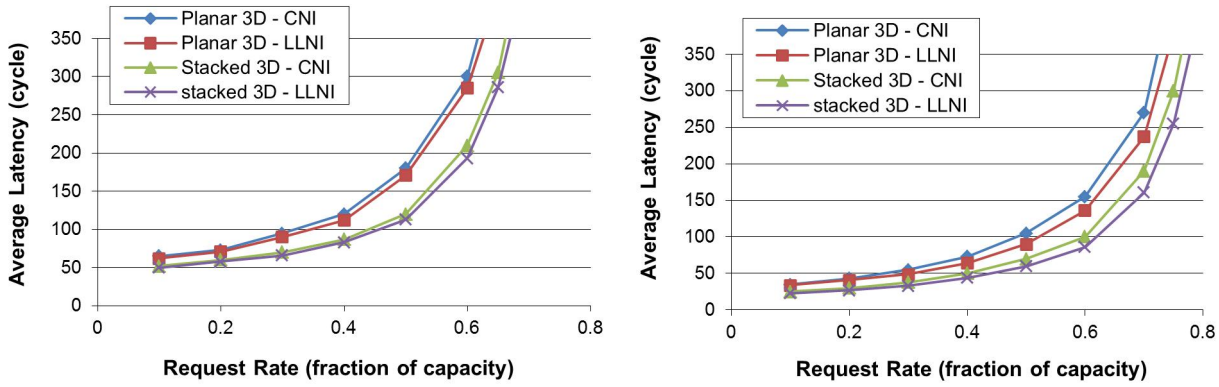


Fig. 4. Performance results under (a) uniform and (b) non-uniform traffic profiles.

we use latency defined as the number of cycles between the initiation of a request operation issued by a processor and the time when the response is completely delivered to the processor from a memory. All the cores and routers are assumed to operate at 1GHz. We also set the size of the reorder buffer to 48 words, able to embed 6 outstanding requests with the burst size of 8. All memories can be accessed simultaneously by each processor continuously generating memory requests. Furthermore, the size of each queue (and FIFO) in the network interface and memory controller is set to 8×32 bits.

### *B. Performance Analysis*

We have considered the uniform and non-uniform synthetic traffic patterns to evaluate the efficiency of the proposed logic layer communication platform. These workloads provide insight into the strengths and weaknesses of the different buffer management mechanisms in the interconnection networks. The random traffic represents the most generic case, where each processor sends in-order read/write requests to memories with the uniform probability. Hence, the memories and request type (read or write) are selected randomly. In the non-uniform mode, 70% of the traffic is local requests, where the destination memory is one hop away from the master core, and the rest 30% of the traffic is uniformly distributed to the non-local memory modules. The simulation results under the uniform and non-uniform traffic profiles are depicted in Fig. 4(a) and (b), respectively.

As demonstrated in figures, for both memory organizations the platform utilizing LLNI reduces the average latency under both traffic profiles. As mentioned earlier, unlike the conventional network interfaces, since there is a direct channel between the processor and the local memory, requests for local memory do not need to travel through the network. Thus, using the presented network interface for the logic layer reduces the network latency for the non-uniform traffic profile considerably, i.e. most of the requests in non-uniform are local. Fig. 4(b) reveals that the performance gain of the platform using LLNI under the non-uniform traffic is considerably higher.

### *C. Physical Analysis*

To assess the area overhead and power consumption of the proposed logic layer communication architecture, the whole platform including network interfaces, routers, and peripheral logics is synthesized by Synopsys D.C. using the 65nm LP CMOS technology. In this work, the pad size for TSVs is assumed to be 5µm square with pitch of around 10µ. The layout area and power consumption of every component of the presented communication platform are shown in Fig. 5. The total area of this platform is about 16 $mm^2$ while the area of processors and DRAMs are not included in the chart. The most expensive component is the peripheral logic, which consumes about 69% of the total area. This high cost is mainly due to the row buffers and decoders, thereby, peripheral logics are typically implemented on another high-speed logic layer on top of the processor logic layer [9][12][13]. The relative power consumption of the proposed communication platform is also illustrated in Fig. 5. The power consumption is computed near the saturation point (0.6) under the non-uniform traffic profile using Orion library [27]. The total power consumption is about 3.4W at 1.2GHz (i.e. the whole system operates in the same frequency). Routers and peripheral logics are the most power hungry components due to the large amount of switching activity, that is, banks can be accessed in parallel and all remote requests are traversed through the routers.

## VI. CONCLUSION

Previous research works have already demonstrated that 3D stacking of memory on processors can provide significant memory bandwidth. In this work, we have presented a streamlined on-chip communication platform for the logic layer of 3D stacked memory-on-processor architectures. This platform takes advantage of a novel network interface, in order to serve both local and non-local requests efficiently. The results revealed that using the proposed network interface increase the performance of non-local request considerably. Besides, according to the hardware implementation, because the area overhead of peripheral logic is considerably large, it has been reasonable to be placed on the interface layer on top of the processor layer.

## VII. ACKNOWLEDGMENTS

The authors wish to acknowledge the Academy of Finland and Nokia Foundation for the partial financial support during the course of this research.

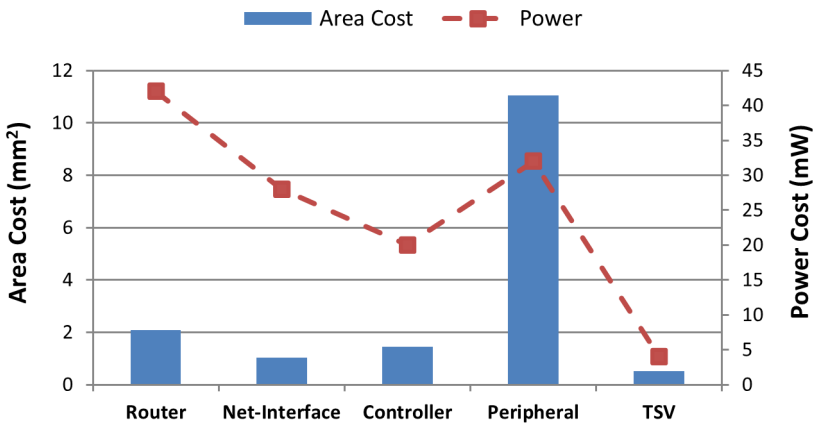


Fig. 5. Area and power consumption cost of the presented on-chip communication platform for the logic layer.

## References


[1] W. A. Wulf and S. A. McKee, "Hitting the memory wall: Implications of the obvious," ACM SIGARCH Comput. Architect. News, vol. 23, no. 1, pp. 20–24, Mar. 1995.

[2] W. Jang and D. Z. Pan, "An SDRAM-Aware Router for Networks-on-Chip," in proc. of DAC'09, pp. 800-805, US, 2009.

[3] K. Banerjee, S. J. Souri, P. Kapur, K. C. Saraswat, "3D ICs: A Novel Chip Design for Improving Deep-Submicrometer Interconnect Performance and Systems-on-Chip Integration", Proc. of the IEEE, 89(5):602-633, May 2001.

[4] I. Loi and L. Benini, "An Efficient Distributed Memory Interface for Many-Core Platform with 3D Stacked DRAM," in Proc. of the DATE Conference, Germany, pp. 99-104, 2010.

[5] B. Black et al., "Die-Stacking (3D) Microarchitecture," in Proceedings of MICRO-39, pp. 469-479, 2006.

[6] M. Ebrahimi and et al., "Exploring Partitioning Methods for 3D Networks-on-Chip Utilizing Adaptive Routing Model," in Proceedings of 5th ACM/IEEE International Symposium on Networks-on-Chip (NOCS), pp. 73-80, May 2011, USA.

[7] G. H. Loh, "3D-Stacked Memory Architectures for Multi-core Processors," in proc. of International Symposium on Computer Architecture (ISCA), pp. 453–464, 2008.

[8] International Technology Roadmap for Semiconductors, 2007.

[9] D. H. Woo, N. H. Seong, D. L. Lewis, and H. S. Lee, "An Optimized 3D-Stacked Memory Architecture by Exploiting Excessive, High-Density TSV Bandwidth," In Proc. of 16th International Symposium on High-Performance Computer Architecture, pp.429-440, India, 2010.

[10] M. B. Healy, and et al., "Design and Analysis of 3D-MAPS: A Many-Core 3D Processor with Stacked Memory," To appear in the IEEE Custom Integrated Circuits Conference, San Jose, California, September, 2010.

[11] T. Kgil, A. G. Saidi, N. L. Binkert, S. K. Reinhardt, K. Flautner, T. N. Mudge, "PicoServer: Using 3D stacking technology to build energy efficient servers," in ACM Journal on Emerging Technologies in Computing Systems (JETC), Vol. 4, No. 4, 2008.

[12] G. H. Loh, "Extending the Effectiveness of 3D-Stacked DRAM Caches with an Adaptive Multi-Queue Policy," in proc. of International Symposium on Microarchitecture (MICRO), pp. 201-212, 2009.

[13] B. Zhao, Y. Du, Y. Zhang, J. Yang, "Variation-Tolerant Non-Uniform 3D Cache Management in Die Stacked Multicore Processor Categories and Subject Descriptors," in proc. of International Symposium on Microarchitecture (MICRO), pp. 222-231, 2009.

[14] P. Jacob et al., A. Zia,M. Chu, JW. Kim, R. Kraft, JF. McDonald, K. Bernstein, "Mitigating memory wall effects in high clock rate and multi-core cmos 3D ICs: Processor memory stacks," in Proceedings of IEEE, 97(1), pp. 108-122, 2009.

[15] U. Kang et al., "8Gb 3-D DDR3 DRAM Using Through-Silicon-Via Technology," In Proc. International Solid State Circuits Conference (ISSCC), pp. 130–131, 2009.

[16] G. L. Loi, B. Agarwal, N. Srivastava, S.-C. Lin, and T. Sherwood, "A Thermally-Aware Performance Analysis of Vertically Integrated (3-D) Processor-Memory Hierarchy," In Proceedings of the 43rd Design Automation Conf., pp.991-996, 2006.

[17] J. Liang, S. Swaminathan, and R. Tessier, "aSOC: a scalable, single-chip communication architectures," in IEEE Int. Conf. on PACT, pp. 37–46, Oct. 2000.

[18] A. Jantsch and H. Tenhunen, "Networks on Chip," Kluwer Academic, 2003.

[19] M. Daneshtalab and et al., "A generic adaptive path-based routing method for MPSoCs," Journal of Systems Architecture (JSA-elsevier), Vol. 57, No. 1, pp. 109-120, 2011.

[20] P. Lotfi-Kamran, and et al., "BARP- A Dynamic Routing Protocol for Balanced Distribution of Traffic in NoCs to Avoid Congestion," in Proceedings of 11th IEEE/ACM Design, Automation, and Test in Europe (DATE), pp. 1408-1413, Mar 2008, Germany.

[21] ARM, AMBA AXI Protocol Specification, Mar. 2004.

[22] OCP International Partnership, Open Core Protocol Specification. 2.0 Release Candidate, 2003.

[23] M. Daneshtalab, and et al., "A Low-Latency and Memory-Efficient On-Chip Network," in Proc. of International Symposium on Network-on-Chip (NOCS), IEEE/ACM Press, pp. 99-106, France, 2010.

[24] S. Murali, and et al. "Designing message-dependent deadlock free networks on chips for application-specific systems on chips," In Proc. VLSI-SoC, pages 158-163, 2006.

[25] X. Yang, Z. Qing-li, F. Fang-fa, Y. Ming-yan, L. Cheng, "NISAR: An AXI compliant on-chip NI architecture offering transaction reordering processing", in Proc. ASICON, pp. 890-893, 2007, Greece.

[26] Gaisler IP Cores, http://www.gaisler.com/products/grlib/, 2009.

[27] A. Kahng B. Li, L. Peh, and K. Samadi, "Orion 2.0: A fast and accurate noc power and area model for early-stage design space exploration," in proc. of DATE, pp. 423-428, 2009.