Memory-Efficient Logic Layer Communication Platform for 3D-Stacked 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 (3D) chip stacking technology is emerging as a viable candidate to address the memory bandwidth problem by stacking multiple DRAM layers on top of a multiprocessor layer (logic layer) to reduce wire delay and energy consumption between them. In addition, combining the benefits of 3D memory-on-processor stacking architecture and on-chip networks provides a significant performance gain. To fully exploit the benefits of the 3D stacked memory-on-processor architectures, an efficient on-chip communication platform is required to be integrated in the logic layer. In this paper, we present an on-chip communication platform for the logic layer to exploit the potential bandwidth of stacked memory-on-processor architectures. This platform guarantees low-latency access to the stacked DRAM layers by employing an adaptive memory controller. Experimental results demonstrate that the proposed platform mitigates the average memory access latency (34%) and average memory utilization (31%) considerably and the overall performance gain is about 20%.

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 high memory bandwidth is the three-dimensional (3D) stacking, enabling the construction of circuits using multiple layers of active silicon bonded with low-latency, high-bandwidth and very dense vertical interconnects [6]-[21]. 3D stacking reduces the interconnect delay problem 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 process 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 [11][21]-[35]. 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 good scalability in number of layers.

As 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 [36][37].

In this paper, we present an efficient on-chip communication platform for the logic layer in the realm of 3D stacked memory-on-processor architectures where the key ideas are threefold. The first idea is to design a modular multiprocessor communication platform for the logic layer to scale the bandwidth among the processors. The second idea is to equip the on-chip network 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 proposed network interface also supports the out-of-order handling for memory transactions. The last idea is to introduce an adaptive memory controller that lies between the presented network interface and a 3D DRAM. The advantage of this memory controller is to improve memory utilization and reduce the memory latency. Also, the presented memory controller is compatible with different 3D DRAM organizations. 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. PRELIMINARIES

A. DRAM Structure and Memory Controller

The DRAM memory is divided into ranks, typically with one or two ranks per DRAM chip. Fig. 1(a) shows a simplified architecture of an DRAM memory chip where each rank is subdivided into banks [2][3][4]. Memory requests to different banks can be serviced in parallel. That is, a benefit of a multibank architecture is that commands to different banks can be pipelined. A complete DRAM access may require three commands: bank precharge, row activation, and column access (read/write). A bank precharge charges and prepares the bank, while a row-activation command is used to copy all data in the selected row into the row buffer (sense amplifier). Once a row is in the row buffer, then column commands can be issued to read/write data from/into the memory addresses (columns) contained in the row. To prepare the bank for a next row activation after completing the column accesses, the cached row must be written back to the bank memory array by the precharge command [2].

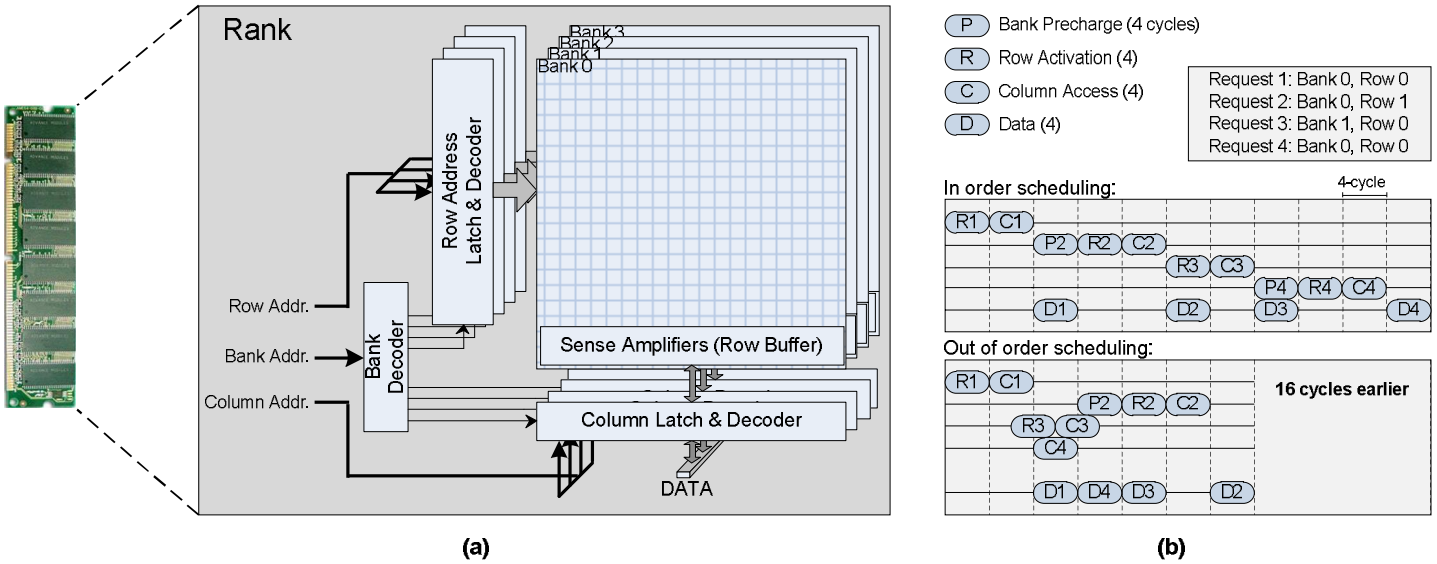

Fig. 1. (a) DRAM architecture and (b) the memory scheduling of four memory requests.

Since the latency of a memory request depends on whether the requested row is in the row buffer of the bank or not, a memory request could be a row hit, row conflict or row empty with different latencies [2][3][4]. A row hit occurs when a request is accessing the row currently in the row buffer and only a read or a write command is needed. It has the lowest bank access latency as only a column access is required. The contents of the row buffer first need to be written back into the memory array using the precharge command. Afterward, the required row should to be opened and accessed using the activation and read/write commands. The row conflict has the highest bank access latency. The memory controller which lies between processor(s) and the DRAM chip is responsible for interfacing with the actual DRAM chips. Indeed, it generates the required commands for each request from a processor and schedules them on the DRAM buses. Among all pending memory requests, based on the state of the DRAM banks and the timing constraints of the DRAM, the scheduler of memory controller decides which DRAM command should be issued. The average memory access latency and memory bandwidth utilization can be reduced and improved, respectively if an efficient memory scheduler is employed [2][3][4]. Fig. 1(b) reveals how the memory access scheduling affects the performance where the sequence of four memory requests is considered. Request 1 and 3 are row empties, while request 2 and 4 are row conflicts based on the timing constraints of a DDR2-512MB used as example throughout this paper, 4-4-4 (t_{RP}-t_{RCD}-t_{CL}) [5]. As depicted in Fig. 1(b), if requests are scheduled out of order: request 4 is scheduled before request 2 and 3 to turn request 4 from a row conflict to a row hit, request 3 is pipelined after request 1 for bank interleaving. As a result, the out of order scheduler needs only 28 memory cycles to complete 4 requests while in order scheduler requires 44 cycles.

B. 3D IC Technology Overview

3D integration is a promising technology allowing multiple silicon layers to be stacked on top of each other. There are many technologies for die stacking being pursued by industry and academia. Wafer-Bonding [16][17] and Multi-Layer Buried Structures (MLBS) [18][19] are the most promising ones. The details of these processes are described in [6]. Wafer-to-wafer bonding appears to be the leading contender in industry and many recent academic studies have assumed this type of 3D stacking technology [6]-[11][20]. Wafers can be stacked either Face-to-Face (F2F) or Face-to-Back (F2B) and both have pros and cons. While the former provides the greatest layer-to-layer via density, it is suitable for two-layers; and additional layers would have to employ back-to-back placement using larger and longer vias. On the other hand, Face-To-Back provides uniform scalability to an arbitrary number of layers, despite a reduced inter-layer via density [12]-[21]. Each layer consists of bulk silicon, active devices, and multiple layers of metal routing while after fabrication each wafer is thinned 10-100µm in thickness and the distance between wafers can range from 5µm to 50µm which is much shorter than the wire length between cores on a tier [16]-[21]. Wire bonding, micro-bump, contactless, and Through Silicon Via (TSV) are some of the vertical interconnect technologies that have been used to connect stacked silicon layers [12]-[21]. The TSV interconnections are short, fast, and dense allowing very high inter-layer bandwidth that cannot be provided by other technologies, therefore, are the most promising one among these vertical interconnect technologies [12]-[21]. The pitches of a TSV can range from 1µm to 10µm square [10][29][34] while state-of-the-art TSV manufacturing will be able to produce approximately a pitch of 4mm in 2011 [22]. 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 [11][12][21]-[27]. In this work, we assume the F2B method with TSV interconnects to provide more scalability when more than two layers are employed.

C. 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 [21]-[24][34]. Since the storage density of DRAM is much higher than SRAM, multiple on-chip DRAMs can be implemented on top of the processors in addition to the

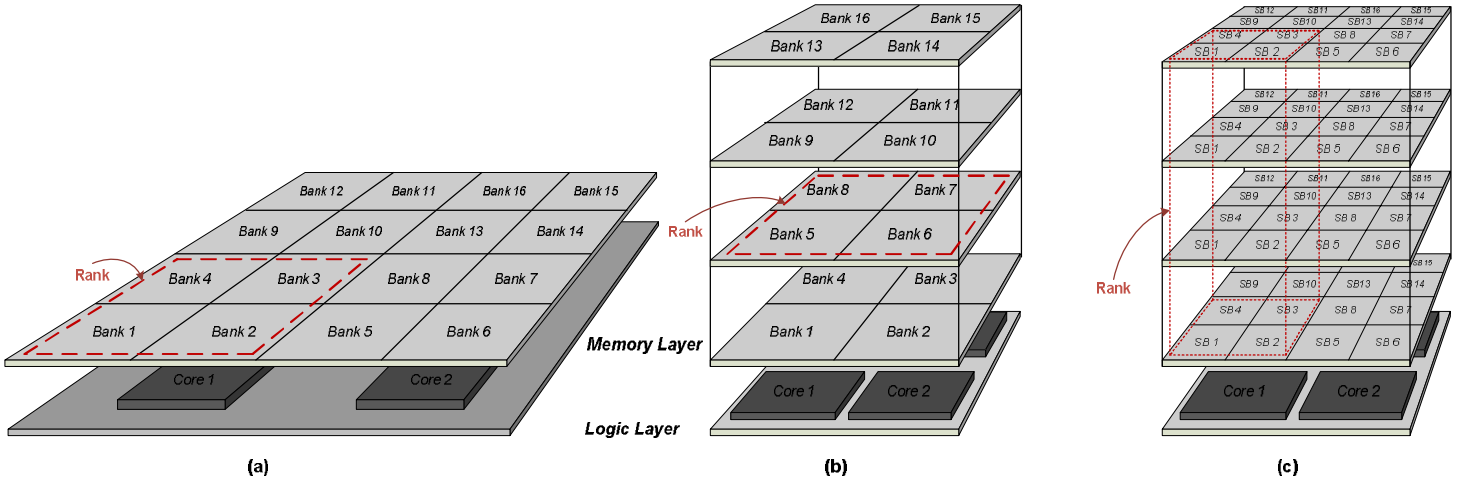

Fig. 2. 3D-stacked DRAM on logic layer with (a) the whole memory on one layer (b) one rank per layer and (c) ranks split across multiple layers.

main memory present on the board. Fig. 2(a) shows a typical DRAM stacking where all DRAM memories (on one layer) are stacked on the logic layer. This structure has advantage if the size of the memories is small. As depicted in Fig. 2(b), one possible memory stacking, named *planar* 3D DRAM, is simply using TSVs to implement a vertical bus across multiple DRAM layers to link them to the logic layer [25][29]-[31]. Although the long memory access latency of such memory organization is reduced, the individual memory architecture in each layer is still based on the traditional two-dimensional off-chip memories and does not provide much bandwidth benefits [21][23][26]. Intuitively, inasmuch as the *planar* 3D DRAM may not be able to exploit the potential benefit of 3D stacking to its full extent, *true* 3D DRAM is introduced by tezzaron [21][34]. In a *true* 3D memory, each flat memory bank of a rank can be divided into several subbanks which are stacked together to form a 3D bank (Fig. 2(c)). This stacked memory structure reduces the length of internal buses, wordlines and bitlines, which in turn reduces the access latency of the memory; and TSVs are implemented as part of the bitlines and wordlines, leading to significant increase of memory bandwidth. Unlike the conventional *planar* 3D memory, all peripheral logic of the *true* 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 (subbanks) through TSVs. Separating the logic layer from the memory layers allows mixing heterogeneous integration of 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. This reduces the memory access time dramatically (32% improvement).

According to thermal analysis carried out in [21][25][28][31], the eight memory layers cause the worst case temperature to increase 5~10°C, meaning that memory layers are passive heat source while the logic layer will be the most active layer which contributes most of the power consumption. Thus, placing a heat sink close to the logic layer is a streamlined method for the purpose of power dissipation and thermal efficiency.

III. Related Work

Most proposed architectures simply consider the 3D memory to be another level in the cache hierarchy [20][24], but some recent studies have already started exploring the benefits of using 3D integration to stack main memory on top of a processor layer [21][23][25][30][31][32][33]. These studies report impressive performance speedups for 3D-stacked memories. In [21] 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 [21]. They presented a 3D internal DRAM architecture (based on *true*-3D memory organization) to better exploit the possibilities enabled by 3D technologies. In this study, each DRAM layer contained 1GB of memory for a total of 16GB in the stack, more than enough storage to act as the entire main memory for netbook, laptop, and desktop systems. 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. FaStack creates 3D memory devices contain many layers of memory cells on top of one control and interface layer, greatly increasing memory density up to 300% as compared to current commercial DRAM [34]. 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. The memory access time is 4ns, which is near SRAM speed. 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 [29] 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 waste due to duplication of circuit components. A memory interface for 3D stacked DRAM is presented in [11]. The memory interface is integrated inside the processor core to reduce the latency of requests for local stacked memory.

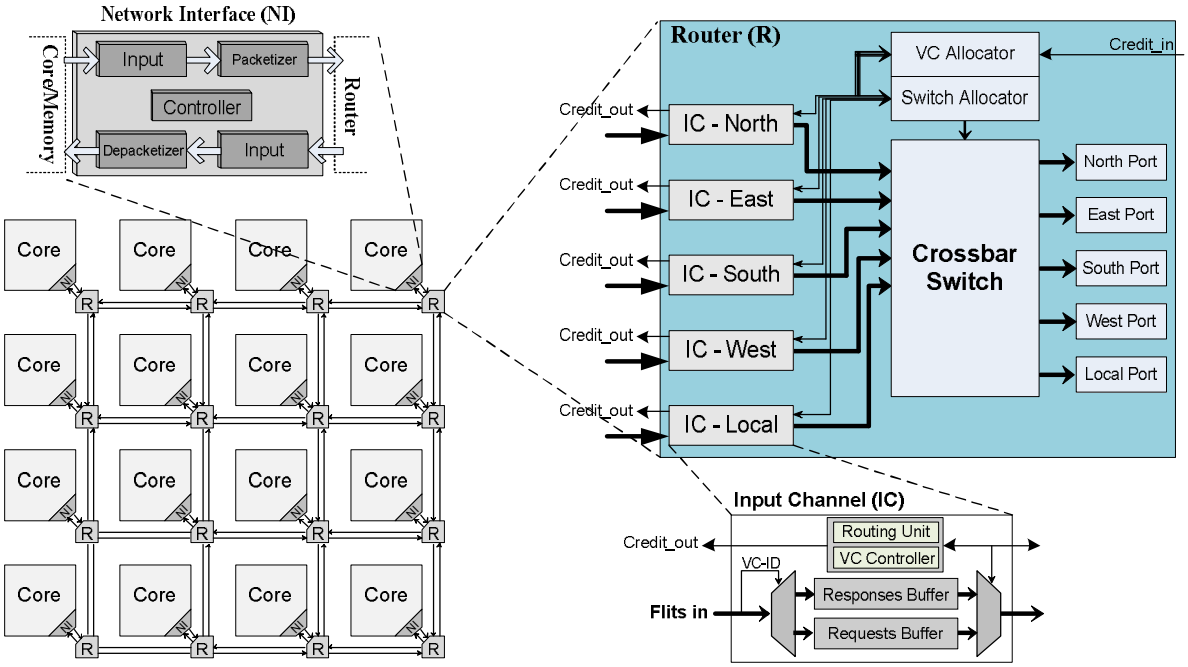

Fig. 3. On-chip inter-core communication platform for the logic layer.

The presented memory interface not only degrades the non-local requests but also it does not support any adaptive memory scheduling to improve the memory utilization which can affect the overall network performance.

Regarding the memory controller, several memory scheduling mechanisms were presented to improve the memory utilization and to reduce the memory latency. The key idea of these mechanisms is on the scheduler for reordering memory accesses. The memory access scheduler proposed in [2] reorders memory accesses to achieve high bandwidth and low average latency.In this scheme, called bank-first scheduling, memory accesses to different banks are issued before those to the same bank. Shao et al. [42] proposed the burst scheduling mechanism based on the row-first scheduling scheme. In this scheme, memory requests that might access the same row within a bank are formed as a group to be issued sequentially, i.e. as a burst. Increasing the row hit rate and maximizing the memory data bus utilization are the major design goals of burst scheduling. The core-aware memory scheduler revealed that it is reasonable to schedule the requests by taking into consideration the source of the requests because the requests from the same source exhibit better locality [4]. In [3] the authors introduced an SDRAM-aware router to send one of the competing packets toward an SDRAM using a priority-based arbitration. As network-on-chips are strongly emerging as a communication platform for chip-multiprocessors, the major limitation of presented memory scheduling mechanisms is that none of them did take the order of the memory requests into consideration. Order sensitive is an adaptive memory controller which based on the order of the memory requests introduced in [40]. This memory controller is integrated inside the network interface. The major contribution of this paper is to propose an efficient on-chip communication platform for the logic layer of the 3D memory-on-processor architecture. The presented platform exploits of a novel adaptive memory controller which increases the memory utilization and reduce the 3D stacked memory latency considerably.

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 the interconnect 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 [35][24]. A 2D-mesh NoC based system is shown in Fig. 3. As mentioned earlier, 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 [38]. 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) [39][40]. 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. 3, has a typical state-of-

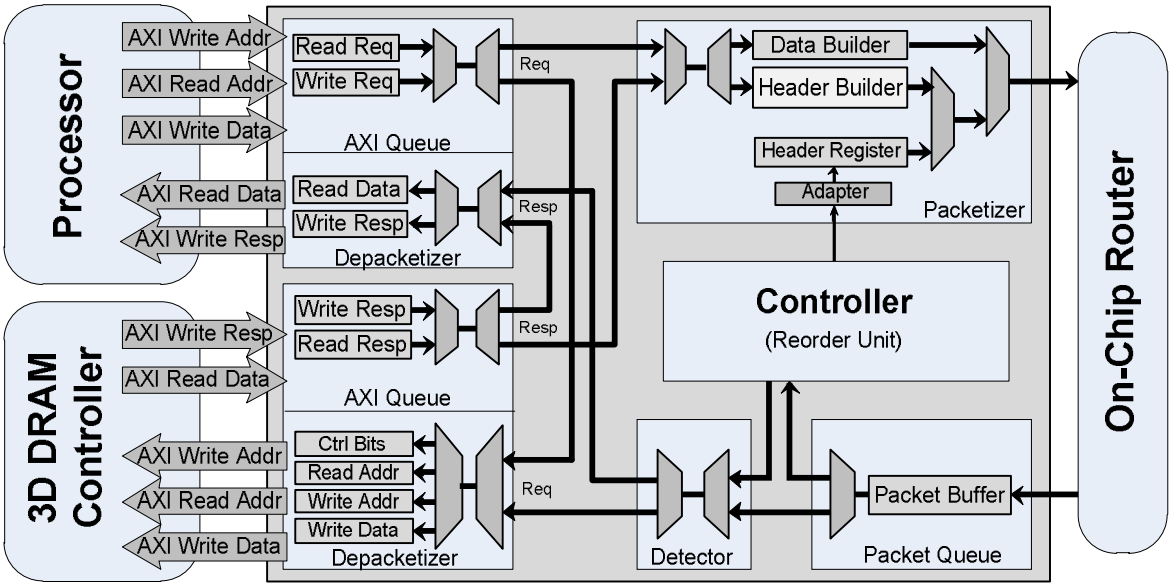

Fig. 4. Logic layer network interface.

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 [41]. 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. 4 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. 4 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. 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.

C. Adaptive Memory Controller

To exploit the potential improvement in latency and bandwidth of 3D DRAM stacking, we propose an efficient memory controller, attached to the network interface. Typically the memory controller lies between processors and the DRAM to generate the required commands for each request and to schedule them on the DRAM buses. Fig. 5 shows the architecture of the proposed adaptive memory controller which efficiently connects the network interface to the stacked 3D DRAM. It consists of AXI inverters, a controller unit, and a physical interface. Due to the fact that the number of banks (subbanks) for each rank in *true* 3D DRAM increased substantially, dedicating a separate request queue for each bank is not worthwhile. Therefore, the controller unit is equipped with a request table, shared read/write request and data buffers, and a memory scheduler. A request table is used to store the state of each memory request, e.g. valid, address, read/write, age, header pointer to the data buffer and any additional state necessary for memory scheduling. The data of outstanding requests are stored in read and write buffers. The read and write buffers (request buffers) are implemented as linked lists. Each memory request (read and write) allocates an entry in its respective buffer until the request is completely serviced. The physical interface generates the data strobes for write operation (through a DLL or delay chain), forwards the data to the memory at double data rate and collects the incoming chunks of the read data.

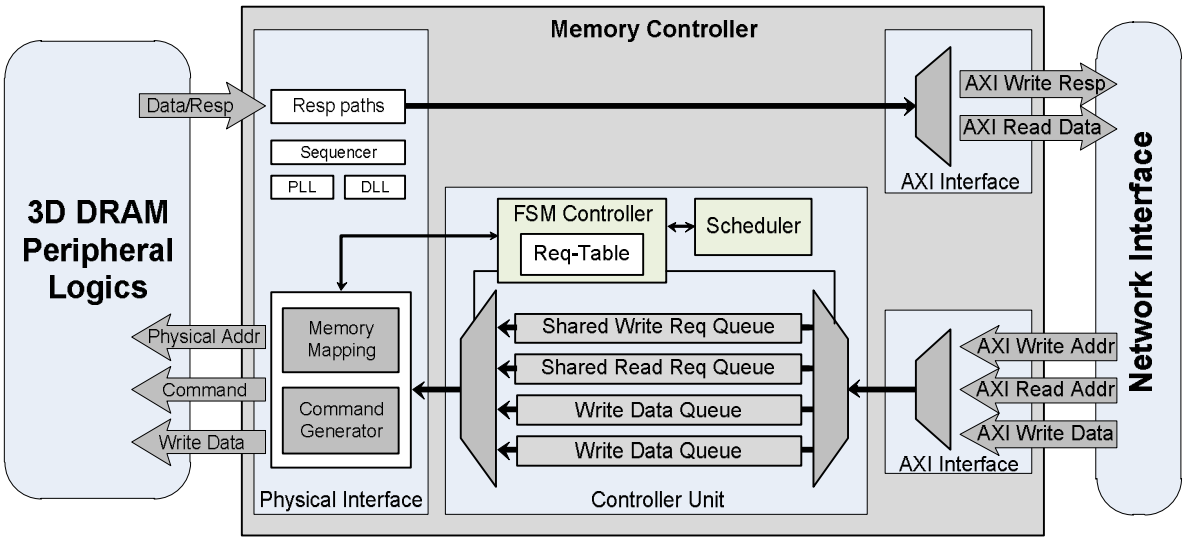

Fig. 5. The proposed adaptive memory controller.

Inasmuch as the *row-first* and *bank-first* policies have better memory utilization [2][4][21], the scheduler of the presented memory controller is based on both schemes. To prevent starvation, aging mechanism is employed so that once a new request enters a queue the age of each request in the queue is increased. At first the scheduler tries to find a request which is a row hit with the highest age. If there are not any row hits, the scheduler selects the oldest request addressing different banks (subbanks) to provide the *bank interleaving* scheme. Otherwise, just the oldest request will be chosen if there are not any row hits or bank interleaving requests in the queue.

V. EXPERIMENTAL RESULTS

In this section, we assess the proposed logic layer communication platform with different 3D DRAM organizations by measuring the average network latency under different traffic patterns. Hence, a cycle-accurate simulator is implemented to assess the efficiency of the proposed method. The simulator models all major components of the logic layer and memory layers.

Table 1. memory parameteres.

Memory type	Time (ns)
2D & planar 3D	t_{RAS}=36ns, t_{RCD}, t_{CAS}, t_{WR}, t_{RP}=12ns each
True 3D	t_{RAS}=24.3ns, t_{RCD}, t_{CAS}, t_{WR}, t_{RP}=8.1ns each

A. System Configuration

In this work, 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-256MB (32b, 4 banks), is stacked on top of each processor so that in total we require a 4GB-stacked memory containing 16 ranks in four layers. For this 3D memory we consider two different organizations, *planar* 3D DRAM (Fig. 2(b)) and *true* 3D DRAM (Fig. 2(c)). Inasmuch as the memory is now stacked on 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) [21][25][31], while the memory access latency for the *true* 3D DRAM improves by 32.5% because of the combination of reducing bitline capacitance, using high-speed logic, and exploiting high-speed TSVs [21][34]. The exact timings are listed in Table 1 for a memory with 4GB, 16 ranks each with 4 banks, and 64-bit FSB [5][21]. We adopt a commercial memory controller and memory physical interface, DDR2SPA module from Gaisler ip-cores [44] and develop two different memory controllers for the experimental results: Typical Memory Controller (TMC) and Adaptive Memory Controller (AMP). 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 [37][43]. 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 configured 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, 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. The request rate is defined as the ratio of the successful read/write request injections into the network interface over the total number of injection attempts. All the cores and routers are assumed to operate at 1.2GHz. For fair comparison, we keep the bisection bandwidth constant in all configurations. 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 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.

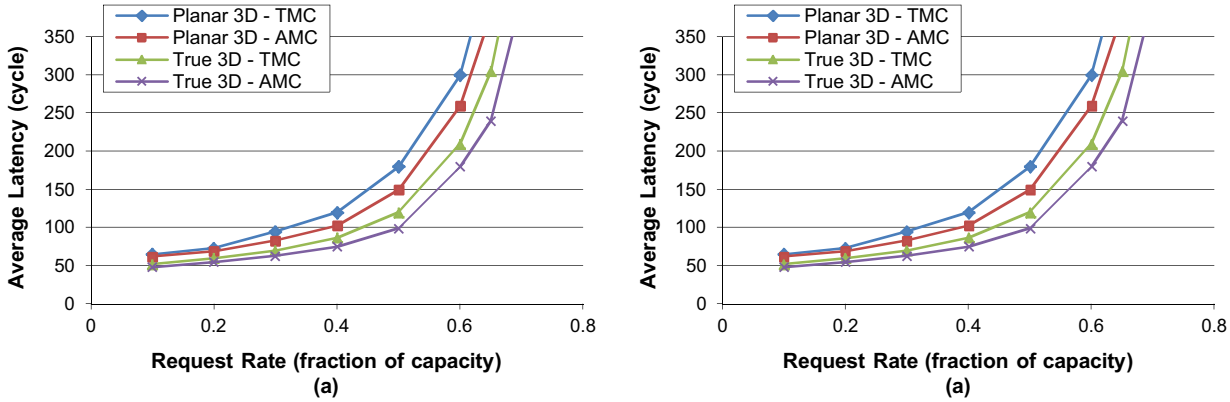

Fig. 6. Performance results under (a) uniform and (b) non-uniform traffic profiles.

These workloads provide insight into the strengths and weaknesses of the different buffer management mechanisms in the interconnection networks, and we expect applications stand between these two synthetic traffic patterns. 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. Eight burst sizes, from 1 to 8, are stochastically chosen according to the data length of the request. 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. 6(a) and (b), respectively. As demonstrated in figures, for both memory organizations the adaptive memory controller reduces the average latency when the request rate increases under both traffic profiles. In addition, the average utilization and latency of memories of each memory organization have been computed near the saturation points (0.5) under the uniform profile. As a result, compared with the baseline architecture (TMC), the average utilization and average latency of the *True* 3D memories are improved by 34% and 31% whereas for the *planar* 3D memories are improved by 32% and 30%. That is, the higher the memory utilization is, the better memory performance is achieved. It is clear that reducing the memory latency improves the network performance in such memory-based systems. As the *true* 3D memory organization provides faster access speed, the average network latency of a logic layer configured with *true* 3D DRAM is much less than a logic layer configured with *planar* 3D DRAM. Moreover, the overall performance gain (included the memory and network latencies) of *true* 3D and *planar* 3D is about 20%, and 18%, respectively.

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, adaptive memory controllers, and peripheral logics is synthesized by Synopsys D.C. using the 65nm LP CMOS technology from STMicroelectronics. Depending on the technology and manufacturing process, the pitches of TSVs can range from 1µm to 10µm square [10][29][34]. In this work, the pad size for TSVs is assumed to be 5µm square with pitch of around 8µ. In addition to the aforementioned configuration of the network interface, the T-ID and S-N were set to 4-bit and 3-bit respectively. The layout areas and power consumptions of every component of the presented communication platform are shown in Fig. 7. The total area cost of this platform is about 16 mm^2 while the area of processors and DRAMs are not included in the chart. The most expensive components are 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 [23][26][27].

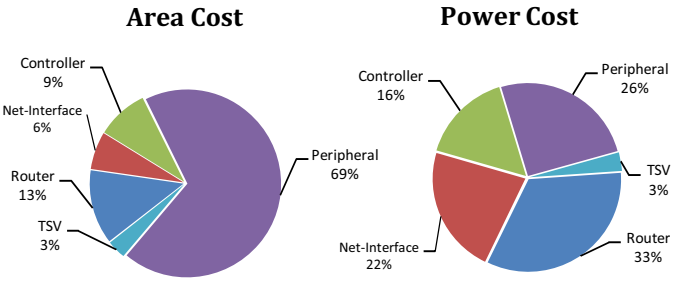

Fig. 7. Area and power consumption cost of the presented on-chip communication platform for the logic layer.

The relative power consumption of the proposed communication platform is also illustrated in Fig. 7. The power consumption is computed near the saturation point (0.6) under the non-uniform traffic profile using Orion library [45]. 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, subbanks can be accessed in parallel and all remote requests are travel through the routers.

VI. Conclusion

Previous studies have already demonstrated that 3D stacking of

memory on processors can provide significant memory bandwidth. In this work, we have demonstrated a streamlined on-chip communication platform for the logic layer of 3D stacked memory-on-processor architectures. This platform takes advantage of a novel memory controller which lies between the network interface and the memory peripheral logic. The memory controller benefits from an adaptive scheduling mechanism which improves the memory utilization and memory latency noticeably. According to the hardware implementation, because the area overhead of peripheral logic is considerably large, it has been suggested to be placed on the interface layer on top of the processor 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] S. Rixner, et al. "Memory access scheduling," In Proc. of ISCA'00, pp. 128-138, US, 2000.

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

[4] Z. Fang, X. H. Sun, Y. Chen, S. Byna, "Core-aware memory access scheduling schemes," In Proc. of IEEE International Symposium on Parallel & Distributed Processing (IPDPS'09), pp. 1-12, Italy, 2009.

[5] Samsung, 512Mb: x4, x8, x16 DDR2 SDRAM Datasheet, 2007.

[6] 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.

[7] B. S. Feero, P. P. Pande, "Networks-on-Chip in a Three-Dimensional Environment: A Performance Evaluation," IEEE Transactions on Computers, vol. 58, no. 1, pp. 32-45, Jan. 2009.

[8] D. Park, et al., "MIRA: A Multi-Layered On-Chip Interconnect Router Architecture", ISCA 2008, pp. 251-261, Pennsylvania State, USA.

[9] J. Kim, C. Nicopoulos, D. Park, R. Das, Y. Xie, V. Narayanan, and C. R. Das, "A novel dimensionally-decomposed router for on chip communication in 3D architectures," in Proc. of the ISCA, pp. 138-149, Boston, USA, 2007.

[10] F. Li, et al., "Design and Management of 3D Chip Multiprocessors Using Network-in-Memory," In 33rd International Symposium on Computer Architecture (ISCA), pp. 130–141, 2006.

[11] 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.

[12] A. Y. Weldezion, M. Grange, D. Pamuwa, Z. Lu, A. Jantsch, R. Weerasekera, and H. Tenhunen. Scalability of the Network-on-Chip communication architecture for 3D meshes. In International Symposium on Networks-on-Chip (NoCS), pp. 114-123,2009.

[13] I. Savidis, S. M. Alam, A. Jain, S. Pozder, R. E. Jones, R. Chatterjee, "Electrical modeling and characterization of through-silicon vias (TSVs) for 3D integrated circuits," Microelectronics Journal, Vol. 41(1), pp. 9-16, 2010.

[14] S. Pasricha, "Exploring Serial Vertical Interconnects for 3D ICs," in Proc. IEEE/ACM DAC, pp. 581-586, 2009.

[15] Y. Lee, Y. J. Kim, G. Huang, M. Bakir, Y. Joshi, A. Fedorov, and S. K. Lim, "Co-Design of Signal, Power, and Thermal Distribution Networks for 3D ICs," Design, Automation & Test in Europe Conference & Exhibition (DATE), 2009, pp. 610-615.

[16] S. Das, A. Chandrakasan, R. Reif, "Three Dimensional Integrated Circuits: Performance, Design, Methodology and CAD tools," in proc. of ISVLSI, pp. 13–18, 2003.

[17] S. Das et al., "Technology, Performance, and Computer Aided Design of Three-Dimensional Integrated Circuits," In Proc. International Symposium on Physical Design, USA, pp. 108-115, 2004.

[18] Y. Xie, G. H. Loh, B. Black, K. Bernstein, "Design Space Exploration for 3D Architectures," ACM Journal on Emerging Technologies in Computing Systems 2, pp. 65–103, 2006.

[19] S.-M. Jung et al., "The revolutionary and truly 3-dimensional 25F^2 SRAM technology with the smallest S^3 cell, 0.16um^2, and SSTFT for ultra high density SRAM," in IEEE Symp. VLSI Tech., pp. 228-229, 2004.

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

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

[22] International Technology Roadmap for Semiconductors, 2007.

[23] 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.

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

[25] T. Kgil, et al., "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.

[26] 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.

[27] 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.

[28] P. Jacob et al., "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.

[29] 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.

[30] C. C. Liu, I. Ganusov, M. Burtscher, and S. Tiwari. Bridging the Processor-Memory Performance Gap with 3D IC Technology. IEEE Design and Test of Computers, 22(6):556–564, 2005.

[31] 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.

[32] M. Ebrahimi, 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.

[33] M. Daneshtalab, M. Ebrahimi, P. Liljeberg, Juha Plosila, H. Tenhunen, "CMIT- A Novel Cluster-based Topology for 3D Stacked Architectures," in Proc. Of 2nd IEEE International 3D System Integration Conference (3DIC), pp. 1-5, Nov 2010, Germany.

[34] Tezzaron Semiconductors, FaStack Memory, http://www.tezzaron.com/memory/FaStack memory.html.

[35] 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.

[36] L.Benini and G.De Micheli, "Networks on chips: a new SoC paradigm", IEEE Computer, January 2002.

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

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

[39] W. Kwon, S. Yoo, S. Hong, B. Min, K. Choi, and S. Eo, "A Practical Approach of Memory Access Parallelization to Exploit Multiple Off-chip DDR Memories", Proc. DAC, 2008.

[40] M. Daneshtalab, 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.

[41] 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.

[42] Jun Shao and Brian T. Davis, "A Burst Scheduling Access Reordering Mechanism", Proceedings of the 13th International Symposium on High-Performance Computer Architecture, 2007.

[43] P. Lotfi-Kamran, 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.

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

[45] 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.