Microprocessors and Microsystems 49 (2017) 44–53

Contents lists available at ScienceDirect

Microprocessors and Microsystems

journal homepage: www.elsevier.com/locate/micpro

Optimizing power efficiency for 3D stacked GPU-in-memory architecture

Wen Wen[a,*], Jun Yang[a], Youtao Zhang[b]

[a] Department of Electrical and Computer Engineering, University of Pittsburgh, Pittsburgh, USA
[b] Department of Computer Science, University of Pittsburgh, Pittsburgh, USA

ARTICLE INFO

Article history:
Received 1 February 2016
Revised 22 September 2016
Accepted 17 January 2017
Available online 21 January 2017

Keywords:
GPU
Stacked memory
NoC
Power efficiency

ABSTRACT

With the prevalence of data-centric computing, the key to achieving energy efficiency is to reduce the latency and energy cost of data movement. Near data processing (NDP) is a such technique which, instead of moving data around, moves computing closer to where data is stored. The emerging 3D stacked memory brings such opportunities for achieving both high power-efficiency as well as less data movement overheads. In this paper, we exploit power efficient NDP architectures using the 3D stacked memory. We integrate the programmable GPU streaming multiprocessors into the NDP architectures, in order to fully exploit the bandwidth provided by 3D stacked memory. In addition, we study the tradeoffs between area, performance and power of the NDP components, especially the NoC designs. Our experimental results show that, compared to traditional architectures, the proposed GPU based NDP architectures can achieve up to 43.8% reduction in EDP and 41.9% improvement in power efficiency in terms of performance-per-Watt.

1. Introduction

In today's computing systems, one emerging challenge arises from data movement between processors and its main memory. Traditional computer systems follow Von-Neumann architectures, in which processors communicate with memory through a hierarchy. Such an architecture inherently presents latency problem in walking through the hierarchy. Moreover, as data-centric computing becomes more prevalent, the volume and resulted energy consumption for data movement well exceeds the energy consumed in actual computing, according to recent studies [1–3].

In recent years, there have been many studies on near data processing (NDP) architectures that aim to achieve higher performance with better power efficiency [4–8]. The basic idea is simply to move data processing close to the memory, instead of transferring data between them via off-chip buses. This could reduce the bandwidth pressure on the memory channel while reducing the memory access latency.

Even though the concept of NDP was proposed over a decade ago, its resurgence in recent years is mainly driven by the innovation in 3D stacked memory thanks to the advances in semiconductor manufacturing and chip packaging [4]. One typical product is the Hybrid Memory Cube (HMC) [9], which stacks multiple layers of DRAM dies on top of a logic die and integrate them into a single package. The logic die communicates with memory layers using through-silicon vias (TSVs). It also hosts other necessary logics such as memory controllers and I/O interfaces to host processors [10–14]. Such an architecture brings new opportunities for NDP because the logic layer provides real estate for compute units to perform data processing. In addition, large amount of TSVs and their high frequency provide significantly higher bandwidth than with traditional CPU-memory channel which is mainly limited by the processor pin count. As a result, implementing NDP in such an architecture can greatly leverage the inherent large bandwidth to benefit data intensive workloads.

In this paper, we propose to use GPU-based NDP architecture, with streaming multiprocessors (SM) as compute units on the logic layer of an HMC. The rationale is that SMs are naturally designed for massive data parallel processing, which is entailed by data intensive workloads. Such compute units can well leverage the large memory bandwidth and high bank-level parallelism provided by HMC to deliver high instruction throughput. Also, the widely adopted programming languages for GPUs such as CUDA and OpenCL make programming the NDP architecture much easier than using custom designed specialized hardware [7,8,12,15] as the computing units. Finally, previous work has also demonstrated the advantages of GPU-based NDP architecture [5] based on analytical analysis. However, the architecture within the logic die, es-

[*] Corresponding author.
E-mail addresses: wew55@pitt.edu (W. Wen), juy9@pitt.edu (J. Yang), zhangyt@cs.pitt.edu (Y. Zhang).

http://dx.doi.org/10.1016/j.micpro.2017.01.005
0141-9331/© 2017 Elsevier B.V. All rights reserved.

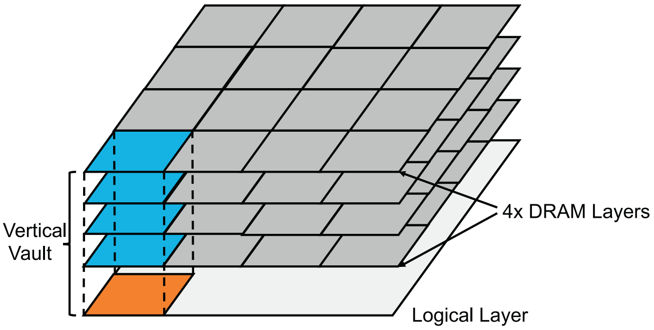

Fig. 1. An HMC with 8 layers of vertically stacked DRAM dies.

pecially the interconnection among SMs have been overlooked. In this work, we use cycle-accurate simulation to evaluate and understand the performance, area and power consumption of our proposed NDP architecture. Further, we propose a Concentrated Diagonally-linked Mesh NoC for connecting SMs, memory controllers and I/O interfaces on the logic die, and study its tradeoffs against a fully-connected crossbar in area, performance and power. Specifically, this paper makes following contributions:

- We propose an area-efficient concentrated diagonally linked mesh NoC for the logic die. We carefully analyze the possible trade-offs between feasible NoCs, number of SMs on the logical layer, in order to produce the most power efficient design.
- We use a cycle-accurate architectural simulator, with models for power and area estimations to perform a design space exploration for our proposed NDP architecture. With those tools, we demonstrate that our proposed GPU-based NDP architecture improve the energy efficiency over the traditional host-GPU based execution model without incurring thermal threat. This is achieved by cutting down the number of SMs and moving the actual computing closer to the memory with optimized NoC design.

The remainder of this paper is organized as follows. Section 2 presents previous related work and backgrounds for 3D stacked memory and GPU architecture. Section 3 introduces our proposed GPU-based NDP architecture and its design space exploration. Section 4 demonstrates our experimental setups, methodology and results. Finally, Section 5 concludes the whole paper.

2. Background and related work

2.1. 3D stacked memory

The emerging 3D stacked memory technology brings new opportunities to implementing NDP. First, multiple DRAM dies are integrated in the same package, with or without a logic layer. The former can be custom designed to integrate with a target complex processor die, e.g. the High Bandwidth Memory [16], while the latter comes with on its logic layer basic components such as memory controllers and I/O interfaces that can form scalable network of stacked memory, e.g. HMC [9]. Both designs use through silicon vias (TSVs) for data transmission across memory layers and the logic layer. TSVs provides critical benefits including much higher bandwidth than traditional off-chip memory bus, lower memory access latency and energy consumption, all of which are important to achieving better energy efficiency in NDP. We use the design of HMC in this work and extend its logic layer to incorporate compute units. Fig. 1 sketches the architecture of an HMC with 8 layers of DRAM dies and 16 memory vaults connected by TSVs.

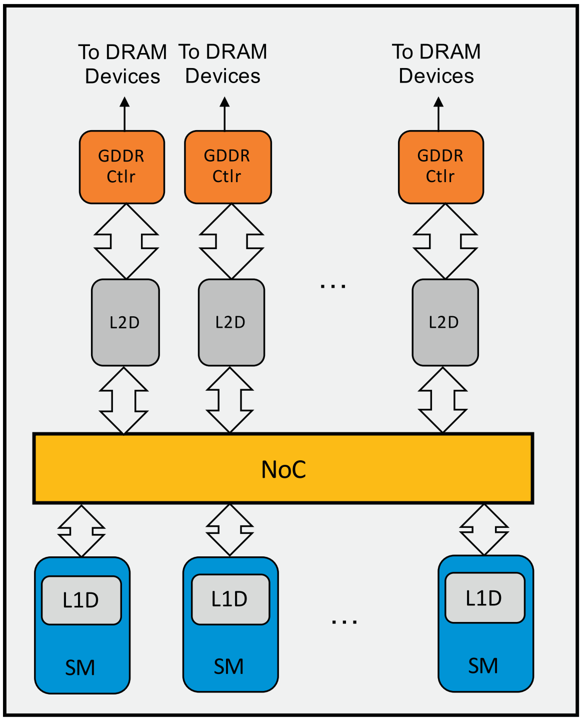

Fig. 2. GPU architecture.

2.2. GPU architecture

A GPU typically consists of a group of single instruction and multiple thread (SIMT) processing units which can exploit significant thread level parallelism (TLP). In this paper, we mainly focus on nVidia GPU Fermi architecture, as shown in Fig. 2. There are 15 SIMT processing units, known as streaming multiprocessors (SM). They are connected to L2 caches via NoC. L2 caches are further coupled with memory controllers using GDDR5 interface. Each SM has 32 compute cores, enabling thousands of threads executing in parallel. Although memory access latency can be hidden under computing cycles due to multithreading, the high amount of thread level parallelism generates high volume of memory requests. Such a high pressure on memory bandwidth can be naturally accommodated by HMC. Hence, we opt for SM as the compute units in our NDP design.

2.3. Related work

Zhang, et al. [5] proposed the TOP-PIM architecture which integrates GPU compute units (CU in AMD terminology) on the logic die of an HMC, similar to our architecture. One contribution in that work is the analytical models for performance and energy evaluations. However, the work ignored the impact of NoC on the logic layer, and hence failed to exploit the tradeoff between area, performance and power in order to generate the most energy efficient architecture. In addition, we use a cycle-accurate simulator with area and power models for evaluation. This helps to better understand the dynamic behavior of different workloads, and better exploit a range of design alternatives with tighter estimations bounds.

Pugsley et al. [6] proposed to use NDP to accelerate the MapReduce algorithm in warehouse scale computing applications through integrating hundreds of wimpy cores into logic layer of the 3D stacked memory. This architecture also aims to achieve high level of parallelism but is less effective than a SIMT architecture such as SMs of a GPU. The latter also has less contention on the NoC as its size is significantly smaller.

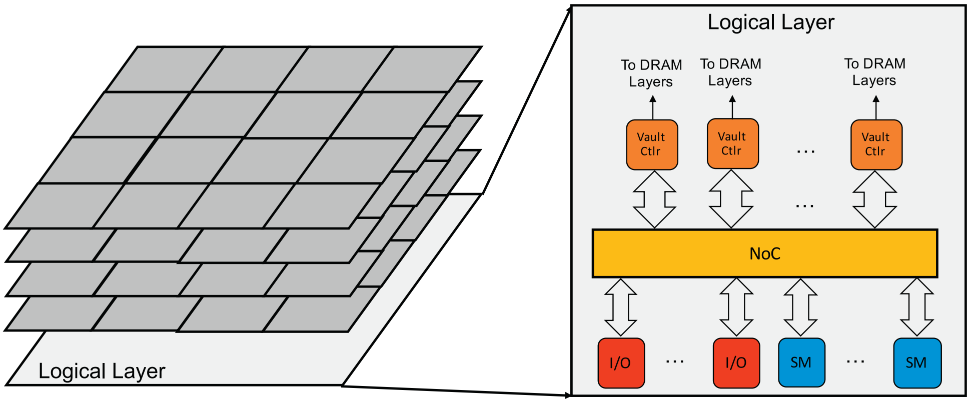

Fig. 3. System overview of the GPU-based NDP architecture.

Farmahini–Farahni et al. [7] proposed an NDP architecture called DRAMA. This architecture integrates a Coarse-Grained Reconfigurable Array (CGRA) on the logic layer and stacks it with a memory layer. However, the programmability is the weakness of the CGRA-based accelerators, which may greatly limit its uses to wide range of applications. In contrast, our design directly adopts SMs and hence can be easily programmed with mature GPU programming models.

Junwhan et al. [13] designed a novel PIM architecture, Tesseract, as a feasible solution for large-scale graphics applications. Leveraging 3D stacked memory technology and investigating the algorithms, Tesseract achieves high performance in large-scale graph processing. This paper also proposed a communication scheme between memory partitions, message-triggered prefetching as well as an efficient programming interface, which can further exploit the large bandwidth provided by 3D stacked DRAM. This work is different from ours since the Tesseract architecture is specifically optimized for graph application instead of general-purpose memory intensive applications.

Akin et al. [12] and Sadi et al. [15] presented similar 3D stacked application-specific in-memory accelerator for data intensive computing. However, application-specific logics can only execute predefined functions with limited configurations, and require highly optimized software or kernels that run on them. Hence, they are difficult to adapt to application evolutions and not as general-purpose as our design.

3. Design exploration of GPU-based near data processing

3.1. GPU-based NDP architecture

Our proposed NDP architecture is illustrated in Fig. 3. There are 16 memory controllers for 16 vaults and 4 I/O interfaces on the logic layer, all connected to the NoC. We place all SMs on this layer, connected to the NoC as well. We assume there are 4 layers of DRAM dies and a total of 256 banks [17]. Conceptually, the approach we build the GPU-based NDP architecture is shown in Fig 3. We place all the computing logics of SMs on logical layer, and all of them are connected to all 16 vault controllers and 4 I/Os via NoCs. With this design, the latency as well as the power consumption by moving data from off-chip links between GPU processor and main memory can be reduced. In our design, we use the HMC-like 3D stacked memory with 4 layers of DRAMs, 16 vaults and totally 256 banks.

One of the most important reason we use GPU's SMs as the computing units for our NDP architecture is its programmability. We can simply port the legacy CUDA codes onto our GPU-based NDP architecture with little modifications. In this initial work, we do not partition tasks between the host processor and near memory accelerator. Instead we load the entire task onto the SMs on the logic layer. All data used for computing are initialized and moved to the 3D stacked memory in the same way as the conventional CPU-GPU system, and the SMs on the logic die can easily access them through vault controllers. With this design, the traditional off-chip data communication transmission between GPU and memory is reduced to on-chip communication between SMs and memory layers.

Table 1
ORION3.0 configuration.

Technology	45nm
Flit size	32 Bytes
Crossbar type	Tristate
Buffer type	SRAM
Wire layer	Intermediate
Link spacing	Single

Table 2
Area estimation/mm^2.

Architecture	# of SMs	NoC topology	NoC area	Total area
1	6	Crossbar	101.37	337.01
2	6	Concentrated DMesh	6.68	242.31
3	8	Concentrated DMesh	6.76	308.98

The area of the logic die is the major constraint of the GPU-based processing in memory architecture. According to the report from Micron [9], the first generation of HMC has a logic die size of 27mm × 27mm. Considering that technology has advanced, we conservatively assume that the area budget is 20mm × 20mm which includes SMs, NoC, vault controllers and all the interface logics. In this paper, we use 40nm technology for both the logic die and DRAMs dies. Details of the hardware configuration can be found in Table 3. Based on the area estimation from GPUWattch [18], the entire chip area is 699.64mm^2 and a single SM's area is 44.03mm^2. However, the real chip area of GTX480 is 529mm^2 [19]. Therefore, we scale the area estimation from

Table 3
Simulation parameters.

Architecture	GTX Fermi, 15 SMs and 32 PEs per SM
L1 cache/SM	16KB, 32 sets 4-way, 128B cache line
L2 cache	6 partitions, 768KB, 8-way, 128B cache line
Warp scheduler	Loose Round Robin (LRR)
DRAM	924MHz, FR-FCFS scheduler

GPUWattch according to the real chip size, and assume a single SM occupies 33.29mm^2. We also assume that a baseline configuration of 6 SMs that will not exceed 50% of the die area budget.

3.2. Design space exploration

Recent studies have shown that for memory intensive workloads, their performance will level off when the number of SMs reach a certain value, typically smaller than the maximal number of available SMs of a GPU [20]. For example, the performance achieved by running a workload on 8 SMs may be close to running on 15 SMs. This observation forms a strong basis of our NDP architecture, as the limited area budget does not permit a full-scale GPU on the logic layer. A slim version on the logic layer may be sufficiently good for memory intensive workloads. However, the NoC architecture interconnecting all components on the logic layer presents tradeoffs between area, performance and energy efficiency. The fastest NoC such as a fully connected crossbar is beneficial to achieving high performance, but will occupy significant amount of die area, restricting the number of SMs that can be integrated in. A less area-hungry NoC will increase the memory access latency, which we will show can be quite significant, but will make room for more SMs overcoming the performance loss. In this section, we propose the following three GPU-based NDP architectures with different NoCs between SMs, I/Os and vault controllers.

Architecture 1 (6 SMs and fully connected crossbar NoC - “ndp-xb”): In this architecture, all the 6 SMs are connected to 4 I/Os and 16 vault controllers through the default fully connected crossbar NoC. Intuitively, this architecture with most expensive fully connected crossbar NoC should have the best performance, while taking most area and consumes most power.

Architecture 2 (6 SMs and Concentrated Diagonally-Linked Mesh NoC - “ndp-cdmesh6”): In this architecture, we optimize NoCs and consider a concentrated diagonally-linked mesh, inspired by the diagonally-linked mesh only (DMesh) NoC in prior work [21]. DMesh simply adds diagonal links among all nodes that are two-hops away, one in x- and one in y-dimension. We use DMesh in a concentrated network where 1 or 2 SMs, 1 I/O and 4 vault controllers are concentrated into one node. Compared to the crossbar NoC, the Concentrated DMesh has better area and power efficiency. Compared to the conventional mesh, it has less latency and better performance by reducing the hop counts. As depicted in Fig. 4, all concentrated node are linked horizontally, vertically and diagonally. Therefore, each SM can locally access to at least four vaults within the concentrated node, and access to other vaults with one more hop, which will not incur too much performance decreases. In the experimental section, we will show that the Concentrated DMesh has much less area and power consumption than the crossbar. We assume the 2 extra SMs are placed diagonally in two concentrated nodes, leading to an asymmetric Concentrated DMesh NoC.

Architecture 3 (8 SMs and Concentrated Diagonally-Linked Mesh NoC - “ndp-cdmesh8”): Since Concentrated DMesh occupies less area than the fully connected crossbar, we add 2 more SMs to make use of the free chip estate. We use ORION3.0 [22] for their area estimations, which is shown in Table 2, and the configuration of ORION3.0 is shown in Table 1. Due to lack of publicly available literature about the area of vault controllers on logic die, we conservatively assume the vault controller has same area and power as the GDDR5 memory controller. From the Table 2, area difference between crossbar and Concentrated DMesh is 94.69mm^2, with which 2 more SMs can be added to increase the processing capability. The area of all the SMs, NoCs and vault controllers is 308.98mm^2, which is only 78.2% of the total area budget. Finally, the Table 2 shows the area comparison of all the three architectures.

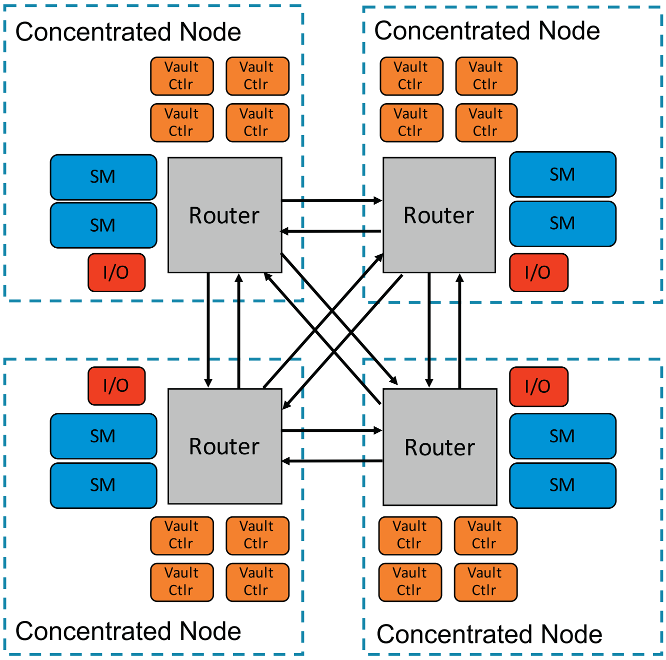

Fig. 4. Overview of the concentrated DMesh.

Summary: Table 2 compares of all three NoC architectures and their total areas. The first architecture uses crossbar to improve performance. It has the lowest NoC and memory access latencies, but more energy and area overhead. However, due to its superior performance, this architecture is quite energy efficient, as our later experiments will show. The second architecture uses less expensive NoC, the Concentrated DMesh, to save both area and power. Its performance is not as good as the crossbar, but the gap is reduced via having each router access mostly local vaults. The third architecture uses more SMs while keeping the DMesh. The hope is to gain performance through more compute power, leveraging performance advantages brought by high TLP. Contrasting the three would show how to use the area of the logic die in a most beneficial way, which will be demonstrated in our experiments.

4. Experimental evaluation

4.1. Simulation methodology and benchmarks

We use and extend the GPGPU-sim 3.2.2 [23] for running all simulations. In this paper, all the configurations are based on NVIDIA GTX480 Fermi architecture, and the key parameters are summarized in Table 3. The GPGPU-sim simulator is heavily revised to model our proposed NDP architecture, including NoC, vault controllers and 3D stacked DRAM. The DRAM timing is obtained from CACTI-3DD [24], where TSV latency overhead is included in every memory access. The resulted DRAM parameters are listed in Table 4. In the performed experiments, we use legacy CUDA benchmarks for all evaluations, as summarized in Table 5. All are from the Rodinia [25] and Parboil [26] benchmark suites. The third column presents the numbers of memory accesses per 1000 instructions, which indicates the memory intensity of the benchmarks. A high value is considered memory intensive, which we mark as “M”, and other benchmarks are considered as compute intensive and marked “C”. Note that the I/Os on logical layer are reserved for data transferring between the host processor and the NDP architecture, and they are not involved in computation. Therefore, I/O intensive applications are not specified and evaluated.

Table 4
Comparison of memory system parameters.

	GDDR5 DRAM	3D stacked memory
Technology Node	40nm	40nm
Bandwidth	177.4GB/s	320GB/s
Frequency	924MHz	1066MHz
Number of Banks	96	256

Timing Constraints (ns)	tCCD = 2.16,tRRD = 6.48
	tRCD = 13.75,tRAS = 35.0
	tRP = 12.96,tRC = 40.32
	tCL = 12.96,tWL = 4.32
	tCDLR = 5.4
		tCCD = 2.16,tRRD = 2.03
		tRCD = 7.67,tRAS = 12.40
		tRP = 6.11,tRC = 17.78
		tCL = 10.77,tWL = 4.32
		tCDLR = 5.4

Table 5
Summary of benchmarks.

Applications	MI	Type	Applications	MI	Type
sgemm	4.3	M	bfs_k1	0.3	C
sad_k1	0.2	C	lbm_k1	9.4	M
sad_k2	34.9	M	leuko_k1	0.004	C
sad_k3	179.4	M	particle_k1	0.7	C
stencil_k1	2.0	C	particle_k2	499.1	M
stencil_k2	4.0	C	particle_k3	113.1	M
bp_k1	0.7	C	spmv_k1	2.8	C
bp_k2	3.0	C	srad_k2	6.8	M
mrig_k1	27.4	M	hw_k1	3.2	C
mrig_k3	508.7	M	cfd_k1	12.5	M
histo_k1	0.9	C	cfd_k2	25.0	M
histo_k5	107.3	M			

Table 6
Summary of evaluated architectures.

Architecture	Type	NoC topology	# of SMs	Bandwidth
baseline1	Baseline	Crossbar	15	177.4GB/s
baseline2	Baseline	Crossbar	6	177.4GB/s
ndp-xb	NDP	Crossbar	6	320.0GB/s
ndp-cdmesh6	NDP	Concentrated DMesh	6	320.0GB/s
ndp-cdmesh8	NDP	Concentrated DMesh	8	320.0GB/s

4.2. Evaluated architectures

In order to show the power efficiency gained from exploring the NDP architectures and optimizing the NoC topology, we compare several design options, summarized in Table 6. The "baseline1" architecture is the GPU GTX480 configured with default parameters, including 15 SMs and GDDR5 main memory. The second architecture "baseline2" is the default GPU GTX480 with only 6 SMs, and the other parameters remain same as first configuration. The third architecture "ndp-xb" is the NDP design with 6 SMs integrated in the logic layer. The NoC architecture is a fully connected crossbar interconnecting 6 SMs, 16 vault controllers and 4 I/Os. The fourth architecture "ndp-cdmesh6" is the NDP architecture similar to "ndp-xb" except that the NoC is a Concentrated DMesh. In the fifth architecture "ndp-cdmesh8", we exploited the area saved from "ndp-xb" to "ndp-cdmesh6" with two more SMs to the fourth architecture.

4.3. Power estimation and thermal analysis

The power consumption of the evaluated architectures mainly comes from three components: GPU processor, NoC and DRAMs. The GPU processor, power estimated by GPUWattch [18], includes the SMs, caches and GDDR memory controllers. The NoC power, including both of leakage and dynamic power, is evaluated in ORION3.0. For the dynamic power of 3D stacked memory, we modify the DRAM power model in GPUWattch with proper values from CACTI-3DD. We calculate the background power for DRAM devices based on the tool from Micron [27]. The background power of a 1 Gb DDR3 is 113.5mW, thus the background power for 1.5GB GPU memory is 1.362W, and 2GB 3D stacked memory is 1.816W. Moreover, for "baseline1" and "baseline2", the GPU is connected to the off-chip DRAM chips via the GDDR5 links. Prior work has shown that its power consumption is not negligible [28], and hence also modeled in our simulations. Specifically, we assume that energy per bit transfer is 15.0pJ/bit for GDDR5 parallel memory interface [28].

Similar to all 3D IC design, the GPU-based NDP architecture also has thermal constraint. SMs on the logical layer can heat the DRAMs and raise their temperature. The proper temperature for DRAM device to operate without increasing refreshing frequency is under 85 °C [29]. Eckert et al. [30] investigated the thermal feasibility of integrating logics with stacked memory, and concluded that a power density lower than $133mW/mm^2$ will not cause DRAM devices to behave improperly, with a passive heat sink. Since our modeled architecture, Fermi, is quite power hungry with its current technology, we make projection to future technology node such as 22nm, the most advanced technology that is supported in our models. We model and calculate power densities for all benchmarks, by dividing the total power by area budget. The NoC power is assumed to account for the same proportion of total power as in 40nm technology. In Fig. 5, our results show that across all benchmarks tested, which include both compute and memory intensive workloads, the power density in logic die is lower than $133mW/mm^2$. This experiment proves that the proposed NDP architectures with next generation fabrication technology are thermally feasible.

4.4. Results

Performance: Fig. 6 shows performance comparison across all evaluated architectures with instructions per cycle (IPC) normalized to "baseline2". For compute intensive benchmarks, "baseline1" achieves best performance due to its massive computing resources, but for memory intensive benchmarks, "ndp-cdmesh8" performs best and its average IPC is even higher than "baseline1". This is because for memory intensive workloads, large memory bandwidth is more important than the amount of compute resources. Hence, fewer SMs with larger bandwidth, such as "ndp-cdmesh8" is most suitable for this class of workloads. Also, overall speaking, "ndp-xb" achieves better IPC than "baseline2", indicating that with the same number of SMs, moving computing closer to memory effectively reduces memory access latency and improves performance. It is also worth nothing that, with "cdmesh" designs, memory requests generated from an SM may have different latencies depending on which vault is accessed. A local vault has shorter latency than remote vaults. Hence, a potential optimization to "cdmesh" architectures is to exploit such NUMA feature and relocate hot memory blocks to local vaults. In our experiments, the memory footprints of all test benchmarks are smaller than the capacity of 4 local vaults, and hence it is safe for us to assume all memory requests have short access latencies. Overall, the three GPU-based NDP architectures have 17.9%, 13.9% and 39.0% performance improvement over "baseline2".

Memory Access Latency: As shown in Fig. 7, we evaluate average memory access latency (i.e. the cycles for a memory fetch from SMs to vaults and data return), including the NoC latency, L2 data cache latency and DRAM vault access latency. This metric evaluates the congestion along the memory access path, so long memory access latencies would indicate that bandwidths of NoC and DRAM are bottlenecks for performances. To show how long memory access latency affects performances, we use memory intensive benchmarks, such as "cfd_k1", "cfd_k2" and " sad_k2", to illustrate

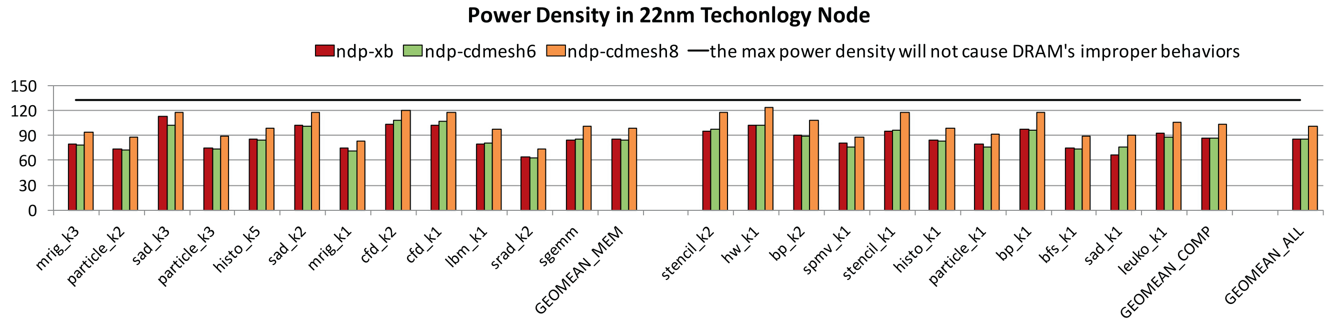

Fig. 5. Power densities (mW/mm^2) in projection to 22nm technology node.

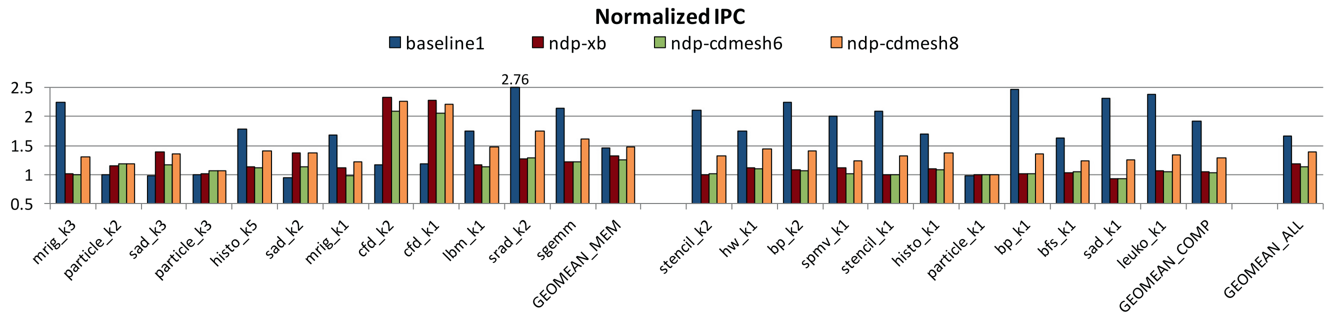

Fig. 6. Performance comparison for all architectures, the IPCs are normalized to "baseline2".

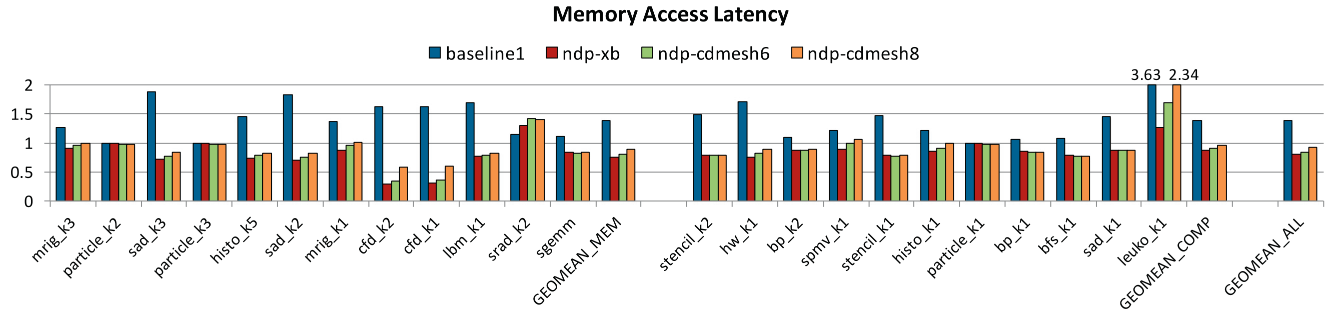

Fig. 7. Memory access latency in cycles comparison for all architectures, the cycle counts are normalized to "baseline2".

it. Taking the "cfd_k2" as an example, as shown in Figs. 6 and 7, its memory access latency in "baseline1" is as much as 102% longer than in "ndp-cdmesh8", which results in performance improvement of "ndp-cdmesh8" by 109% over "baseline1". Similarly, for those benchmarks with long memory access latency in "baseline1", the normalized IPC can be significantly improved in NDP architectures. There are two major reasons for why this happens. First, the NoC and DRAM bandwidths are bottlenecks in baseline architectures. Large number of SMs cores generate large amount of memory requests which exacerbates the bandwidth contention. Second, our proposed NDP architecture can mitigate resource contention by exploiting the significantly larger bandwidth of 3D staked memory and shorter latency with only a smaller amount of SM cores. Among three NDP architectures, the "ndp-xb" performs the best due to its high performance NoC, and the "ndp-cdmesh8" is the worst since two more SMs may put more burdens on the congestion of NoC and DRAM. Finally, for overall evaluation, all three NDP architectures have shorter memory access latencies compared to the two baselines. Specifically, the memory access latencies of "ndp-xb", "ndp-cdmesh6" and "ndp-cdmesh8" are 58.3%, 53.6% and 46.5% shorter than "baseline1", and 20.0%, 15.4% and 8.2% shorter than "baseline2" respectively.

Power consumption: The Fig. 8 shows the comparison of the leakage power among all the evaluated architectures. "baseline2", "ndp-xb" and "ndp-cdmesh6" have almost same leakage power, and other two architectures with more SMs significantly consume more leakage power. This is because the leakage power of SMs is much more dominant than of NoC and DRAMs. Fig. 9 shows the dynamic NoC power for all the evaluated architecture. The results show that our concentrated DMesh saves 68.9% dynamic power of the fully connected crossbar.

We also evaluate the total dynamic power for all architectures in Fig. 10. Overall, from "baseline1" to "ndp-cdmesh6", the total dynamic power gradually reduces, and the total dynamic power of "ndp-cdmesh6" is 83.5% and 16.7% less than "baseline1" and "baseline2" respectively. "ndp-cdmesh6" consumes slightly smaller dynamic power than "ndp-xb" because the percentage of dynamic power of NoC is relative small when compared to other components such as SMs. Hence, the overall trend is that GPU-based NDP architectures consume much less power than traditional architecture such as "baseline" mainly because of the reduced number of SMs and moving actual compute closer to memory. The "ndp-cdmesh8" presents 15.4% more total power than "ndp-cdmesh6"

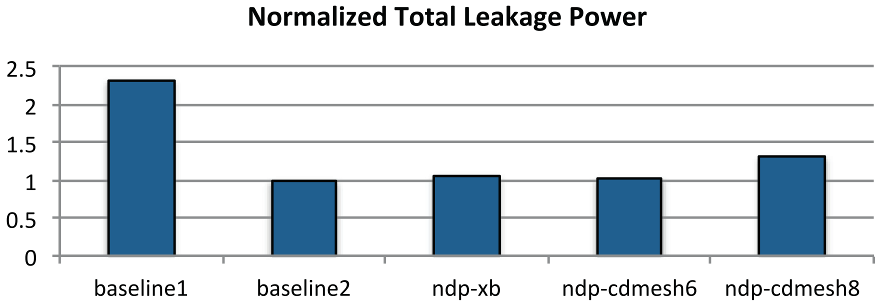

Fig. 8. Leakage power comparison for all architectures normalized to "baseline2".

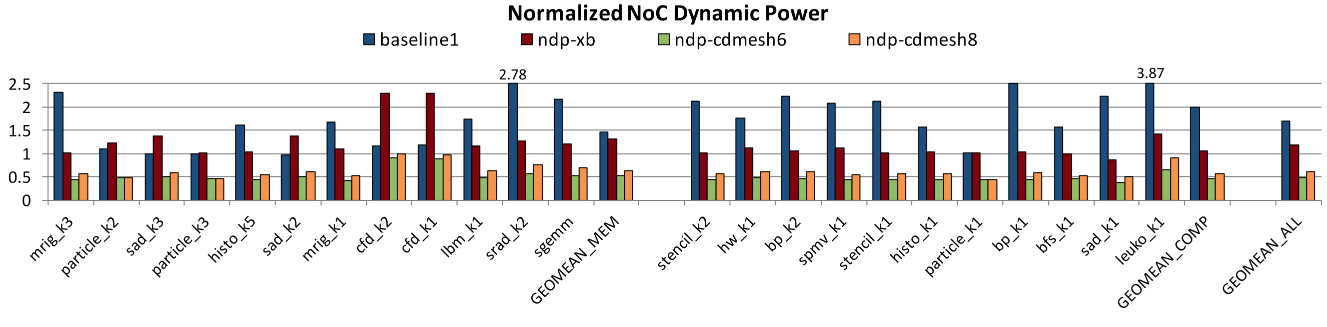

Fig. 9. NoC dynamic power comparison for all architectures normalized to "baseline2".

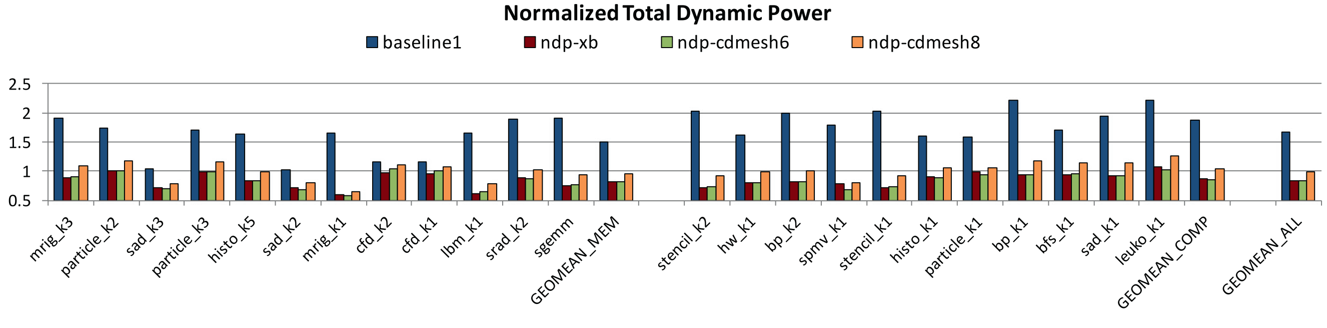

Fig. 10. Total dynamic power comparison for all architectures normalized to "baseline2".

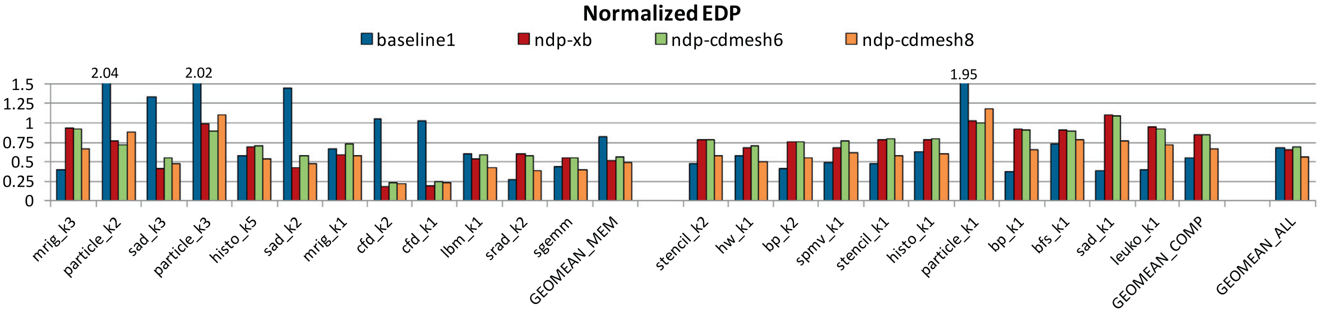

Fig. 11. Energy-delay product comparison for NDP architectures, the EDPs are normalized to "baseline2".

simply due to its larger number of SMs, but it is still 68.1% and 1.3% less than two baselines.

Power efficiency: Fig. 11 shows the comparison of the normalized energy-delay product (EDP) for three NDP architectures. For memory intensive benchmarks, all three NDP architectures can yield better results than both baselines, especially for "ndp-cdmesh8", it has 34.1% and 51.8% EDP reduction from "baseline1" and "baseline2" respectively. However, for compute intensive benchmarks, even though all three NDP architectures perform better than "baseline2", but "baseline1" is the best of all because its large number of SMs can benefit compute intensive benchmarks. Overall speaking, the "ndp-cdmesh8" is 11.6% and 43.8% better than two baselines respectively.

Fig. 12 compares the normalized energy-delay square product (ED^2) for three NDP architectures. As we can see, even though the ED^2 evaluation favors performance over energy consumptions, the three NDP architectures still defeat two baselines for memory intensive benchmarks, especially for "ndp-cdmesh8". It has 24.1%

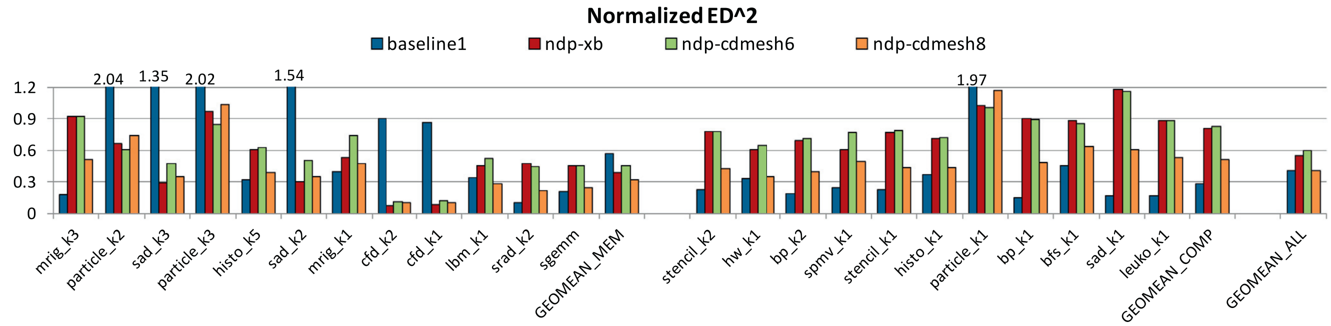

Fig. 12. ED^2 comparison for NDP architectures, the EDs^2 are normalized to “baseline2”.

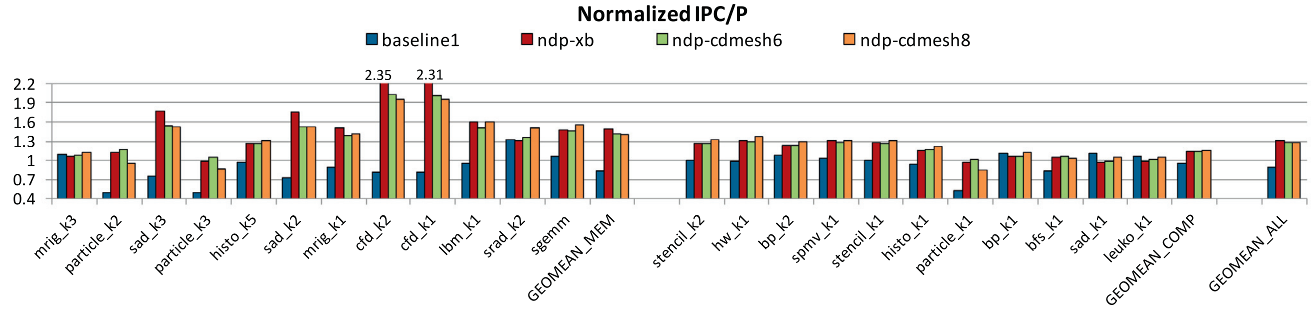

Fig. 13. Performance per Watt comparison for all architectures, the IPCs per Watt are normalized to “baseline2”.

and 67.4% ED^2 reduction from “baseline1” and “baseline2” respectively. Similar to EDP evaluations, for compute intensive benchmarks, “baseline1” is still the best of all since it has much more SMs than other architectures, and all three NDP architectures perform better than “baseline2”. It is worth noting that, for overall evaluation, the “ndp-cdmesh8” is slightly better than “baseline1” by 3% in ED^2 reduction, which means it performs comparably to “baseline1” even when weighing performance more than energy for compute intensive benchmarks.

In addition, we also present performance per watt of all benchmarks, as shown in Fig. 13. The performance per Watt (IPC per Watt) indicates how effectively an architecture uses the power to do the actual computing. Different from the conclusion of EDP evaluation, for both of memory intensive and compute intensive benchmarks, all three NDP architectures have higher performance per watt than the two baselines. The improvement is more significant for memory intensive applications. In addition, we found that a crossbar is better suited for memory intensive workloads in terms of performance per Watt. Whereas a simpler NoC such as CDMesh is more suitable for compute intensive workloads. Overall, “baseline2” performs 11.2% better than “baseline1”, and the three GPU-based NDP architectures achieve 30.7%, 27.9% and 28.0% improvement than “baseline2”.

5. Conclusion

In this paper, we exploit power efficient designs for GPU-based NDP architecture. With cycle-accurate performance simulation, area and power estimations, experiments shows that GPU-based NDP architectures reduce total power consumption and improve the power efficiency by cutting down the number of SMs, moving actual computes closer to memory and leveraging the high bandwidth of 3D stacked memory. In addition, with further optimizing the NoCs in NDP architectures, higher power and area efficiency can be achieved. With area saved from replacing fully connected crossbar with much smaller Concentrated DMesh NoC, more SMs can be integrated to compensate the performance loss. Finally, the proposed architectures can achieve up to 43.8% reduction in EDP and 41.9% improvement in performance per Watt.

References

[1] S.W. Keckler, W.J. Dally, B. Khailany, M. Garland, D. Glasco, Gpus and the future of parallel computing, IEEE Micro 31 (5) (2011) 7–17.
[2] B. Hoefflinger, Itrs: The international technology roadmap for semiconductors, in: Chips 2020, Springer, 2012, pp. 161–174.
[3] S. Borkar, Role of interconnects in the future of computing, J. Lightwave Technol. 31 (24) (2013) 3927–3933.
[4] R. Balasubramonian, J. Chang, T. Manning, J.H. Moreno, R. Murphy, R. Nair, S. Swanson, Near-data processing: insights from a micro-46 workshop, Micro IEEE 34 (4) (2014) 36–42.
[5] D. Zhang, N. Jayasena, A. Lyashevsky, J.L. Greathouse, L. Xu, M. Ignatowski, Top-pim: throughput-oriented programmable processing in memory, in: Proceedings of the 23rd International Symposium on High-Performance Parallel and Distributed Computing, ACM, 2014, pp. 85–98.
[6] S.H. Pugsley, J. Jestes, H. Zhang, R. Balasubramonian, V. Srinivasan, A. Buyuktosunoglu, A. Davis, F. Li, Ndc: analyzing the impact of 3d-stacked memory+logic devices on mapreduce workloads, in: Performance Analysis of Systems and Software (ISPASS), 2014 IEEE International Symposium on, IEEE, 2014, pp. 190–200.
[7] A. Farmahini-Farahani, J.H. Ahn, K. Morrow, N.S. Kim, Nda: Near-dram acceleration architecture leveraging commodity dram devices and standard memory modules, in: High Performance Computer Architecture (HPCA), 2015 IEEE 21st International Symposium on, IEEE, 2015, pp. 283–295.
[8] Q. Zhu, B. Akin, H.E. Sumbul, F. Sadi, J.C. Hoe, L. Pileggi, F. Franchetti, A 3d-stacked logic-in-memory accelerator for application-specific data intensive computing, in: 3D Systems Integration Conference (3DIC), 2013 IEEE International, IEEE, 2013, pp. 1–7.
[9] J.T. Pawlowski, Hybrid memory cube (hmc), in: Hotchips, 23, 2011, pp. 1–24.
[10] T. Zhang, C. Xu, K. Chen, G. Sun, Y. Xie, 3d-swift: a high-performance 3d-stacked wide io dram, in: Proceedings of the 24th Edition of the Great Lakes Symposium on VLSI, ACM, 2014, pp. 51–56.
[11] M. Shevgoor, J.-S. Kim, N. Chatterjee, R. Balasubramonian, A. Davis, A.N. Udipi, Quantifying the relationship between the power delivery network and architectural policies in a 3d-stacked memory device, in: Proceedings of the 46th Annual IEEE/ACM International Symposium on Microarchitecture, ACM, 2013, pp. 198–209.
[12] B. Akin, J. Hoe, F. Franchetti, Hamlet: Hardware accelerated memory layout transform within 3d-stacked dram, in: High Performance Extreme Computing Conference (HPEC), 2014 IEEE, 2014, pp. 1–6, doi:10.1109/HPEC.2014.7040954.

[13] J. Ahn, S. Hong, S. Yoo, O. Mutlu, K. Choi, A scalable processing-in-memory accelerator for parallel graph processing, in: Proceedings of the 42nd Annual International Symposium on Computer Architecture, ACM, 2015a, pp. 105–117.

[14] J. Ahn, S. Yoo, O. Mutlu, K. Choi, Pim-enabled instructions: a low-overhead, locality-aware processing-in-memory architecture, in: Proceedings of the 42nd Annual International Symposium on Computer Architecture, ACM, 2015b, pp. 336–348.

[15] F. Sadi, B. Akin, D.T. Popovici, J.C. Hoe, L. Pileggi, F. Franchetti, Algorithm/hardware co-optimized sar image reconstruction with 3d-stacked logic in memory, 2014,

[16] High bandwidth memory (hbm) dram (jesd235), 2013, (https://www.jedec.org/standards-documents/docs/jesd235). [Online; accessed 01-July-2015].

[17] Hybrid Memory Cube Specification 2.0, 2014, (http://www.hybridmemorycube.org/files/SiteDownloads/HMC-30G-VSR_HMCC_Specification_Rev2.0_Public.pdf). [Online; accessed 01-Feb-2015].

[18] J. Leng, T. Hetherington, A. ElTantawy, S. Gilani, N.S. Kim, T.M. Aamodt, V.J. Reddi, Gpuwattch: enabling energy optimizations in gpgpus, ACM SIGARCH Comput. Arch. News 41 (3) (2013) 487–498.

[19] NVIDIA GeForce GTX 480 GF100 Has Landed, 2010, (http://techgage.com/article/nvidia_geforce_gtx_480_-_gf100_has_landed). [Online; accessed 01-July-2015].

[20] J.T. Adriaens, K. Compton, N.S. Kim, M.J. Schulte, The case for gpgpu spatial multitasking, in: High Performance Computer Architecture (HPCA), 2012 IEEE 18th International Symposium on, IEEE, 2012, pp. 1–12.

[21] P. Nair, C.-C. Chou, M.K. Qureshi, A case for refresh pausing in dram memory systems, in: High Performance Computer Architecture (HPCA2013), 2013 IEEE 19th International Symposium on, IEEE, 2013, pp. 627–638.

[22] A. Kahng, B. Lin, S. Nath, Orion3.0: a comprehensive noc router estimation tool, Embed. Syst. Lett. IEEE PP (99) (2015), doi:10.1109/LES.2015.2402197. 1–1

[23] A. Bakhoda, G.L. Yuan, W.W. Fung, H. Wong, T.M. Aamodt, Analyzing cuda workloads using a detailed gpu simulator, in: Performance Analysis of Systems and Software, 2009. ISPASS 2009. IEEE International Symposium on, IEEE, 2009, pp. 163–174.

[24] K. Chen, S. Li, N. Muralimanohar, J.H. Ahn, J.B. Brockman, N.P. Jouppi, Cacti-3dd: Architecture-level modeling for 3d die-stacked dram main memory, in: Proceedings of the Conference on Design, Automation and Test in Europe, EDA Consortium, 2012, pp. 33–38.

[25] S. Che, M. Boyer, J. Meng, D. Tarjan, J.W. Sheaffer, S.-H. Lee, K. Skadron, Rodinia: A benchmark suite for heterogeneous computing, in: Workload Characterization, 2009. IISWC 2009. IEEE International Symposium on, IEEE, 2009, pp. 44–54.

[26] J.A. Stratton, C. Rodrigues, I.-J. Sung, N. Obeid, L.-W. Chang, N. Anssari, G.D. Liu, W.-M. Hwu, Parboil: a revised benchmark suite for scientific and commercial throughput computing, Center for Reliable and High-Performance Computing (2012).

[27] Calculating Memory System Power for DDR3 Introduction, 2007, (http://www.micron.com). [Online; accessed 01-May-2015].

[28] H. Wang, C.-J. Park, G.-s. Byun, J.H. Ahn, N.S. Kim, Alloy: Parallel-serial memory channel architecture for single-chip heterogeneous processor systems, in: High Performance Computer Architecture (HPCA), 2015 IEEE 21st International Symposium on, IEEE, 2015, pp. 296–308.

[29] Ddr3 sdram standard (jesd79-3f), 2012, (https://www.jedec.org/standards-documents/docs/jesd-79-3d). [Online; accessed 01-July-2015].

[30] Y. Eckert, N. Jayasena, G.H. Loh, Thermal feasibility of die-stacked processing in memory, WoNDP: 2nd Workshop on Near-Data Processing, 2014.

Wen Wen received the B.S. and M.S. degrees in Electronic Engineering from Southeast University in 2011 and 2014 respectively. He is now pursuing for his PhD degree in Electrical and Computer Engineering from University of Pittsburgh. His research mainly focuses on computer architecture.

Jun Yang received the BS degree in computer science from Nanjing University, China, in 1995, the PhD degree in computer science from the University of Arizona in 2002. She is an associate professor in the electrical and computer engineering department, University Pittsburgh. She is the recipient of US NSF Career Award in 2008. She has best paper awards from ICCD 2007 and ISLPED 2013. Her research interests include low power, and temperature-aware microarchitecture designs, emerging non-volatile memory technologies, 3D microarchitecture and networks-on-chip.

Youtao Zhang received the PhD degree in computer science from the University of Arizona in 2002. He is an associate professor in Computer Science Department, University Pittsburgh. His research interests are in the areas of computer architecture, program analysis and optimization. He is the recipient of US NSF Career Award in 2005, the distinguished paper award of the IEEE/ACM International Conference on Software Engineering (ICSE) conference in 2003, the most original paper award of the International Conference on Parallel Processing (ICPP) conference in 2003. He is a member of the ACM and the IEEE.