Exploring Silicon Nanophotonics in Throughput Architecture

Nilanjan Goswami, Zhongqi Li, Ramkumar Shankar, and Tao Li
University of Florida

Editors' notes:
Massively parallel emerging GPU architectures have high throughput demands. This paper explores how silicon nanophotonics and 3-D stacking technologies can meet performance and power dissipation goals for these architectures.

—Sudeep Pasricha, Colorado State University and Yi Xu, Macau University of Science and Technology

■ **Today's state-of-the-art GPUs** have several homogeneous in-order cores. The core count is doubling every 18 months or so. The in-order cores are connected to several on-chip memory controllers using on-chip interconnect [1]. The cores receive data from off-chip memory via the memory controllers. Traditionally, these cores run multiple instances of the same thread concurrently. However, GPUs can also concurrently compute multiple instances of different threads with larger problem sizes. Larger computation load requires large amount of off-chip memory accesses at a significantly higher rate. Unlike multicore CPUs, simultaneously executing threads in throughput architectures generate relatively more memory requests and create interconnect traffic hotspots. In addition, throughput architectures have comparatively more active cores than multicore CPUs, which complicates the throughput interconnect architecture optimization for bandwidth and latency improvement. Moreover, as the number of cores and problem size increase, data transfer in the on-chip interconnect will consume more power and dissipate more heat. This will result in temperature hotspots in throughput interconnect and affect the reliability of the chip.

Recent advancements in CMOS-process-based silicon nanophotonics have substantially mitigated the power, latency, and bandwidth problem [2]–[5]. Three-dimensional stacking technology [6], [7] provides low-latency and high-bandwidth cross-layer communication in a compact form. With significant bandwidth demand in throughput architecture, it is anticipated that power consumption will reach a point at which electrical interconnect and memory subsystem design will become infeasible. On the contrary, optically connected shader cores and memory interface in 3-D stacked multilayer chip seem to be an attractive alternative [2], [8]. In this paper, we propose a 3-D stacked throughput architecture based on silicon nanophotonics technology. The chip has a shader core layer, a cache layer, and a built-in optically connected on-chip network layer. The optical network layer enables dense wavelength division multiplexing (DWDM) high-speed interconnect for core-memory communication.

This paper makes the following contributions.

- We propose 3-D stacked GPU design based on 16-nm CMOS process. It connects 2048 in-order

Digital Object Identifier 10.1109/MDAT.2014.2348312
Date of publication: 15 August 2014; date of current version: 07 October 2014.

Copublished by the IEEE CEDA, IEEE CASS, IEEE SSCS, and TTTC

cores (64 shader cores) and 16 memory controllers through an all-optical DWDM-based crossbar interconnect.
- We also explore the power consumption implication of several low-power photonic crossbar designs in throughput architectures.

Why nanophotonics in throughput architecture?

Workloads in throughput architectures generate myriad memory requests due to data parallel execution paradigm of thousands of threads. Moreover, off-chip memory accesses often cannot be avoided due to limited on-chip caches and scratchpad memory size limitation. Since off-chip DRAM access speed is not improving at a faster rate over the generations, it is imperative to design low-latency on-chip interconnect to reduce overall off-chip access overhead. Such interconnect design will complement the architectural optimizations of memory accesses such as interthread memory access coalescing, etc., to reduce overall off-chip latency and bandwidth. Unlike throughput processors, traditional multicore architectures execute fewer threads simultaneously and interthread access coalescing is limited. In addition, active core count in throughput architecture is much more than multicore architectures. Hence, bandwidth requirement is drastically different for the two architecture genres. Instead of improving individual thread performance as in CPUs, throughput architecture mostly relies on overall throughput of the thousands of simultaneously executing threads. To meet such throughput demand of existing and emerging throughput workloads, interconnect bandwidth and latency improvement is imperative. Moreover, prospective throughput architectures will include intershader cache coherence. With numerous active cores and thousands of concurrent threads, coherence traffic will easily surpass the existing throughput interconnect bandwidth and latency specifications; eventually, it will supersede traditional multicore interconnect traffic demand. Since bandwidth-latency scalability is limited, power and heating problems will restrict electrical large interconnect design for deep submicrometer technology nodes. ITRS predicts silicon nanophotonics as one of the promising future on-chip communication media. Based on bandwidth, latency, and traffic demand in throughput architectures, silicon nanophotonics becomes more apt as a network-on-chip design choice compared to multicore systems.

Nanophotonic throughput architecture

Figures 1 and 2 illustrate the simplified functional unit layout of our proposed silicon-nanophotonics-technology-enabled 3-D throughput architecture, which uses multiple DWDM-based waveguides to optically connect multiple shader cores and off-chip memory arrays. To meet growing computation demand and off-chip memory access load, overall chip area and heat dissipation capabilities become major design issue for throughput architectures; multilayered 3-D throughput architecture enables better area utilization, cooling capabilities, and integration of disparate technology enabled layers. We separate shader core layer and L2 cache/memory interface layer and use face-to-face bonding to connect the two layers; face-to-face assembly enables decoupling the number of through silicon vias (TSVs) from the total number of interconnections between the layers. We incorporate a separate silicon nanophotonic optical layer that transfers the optical signals to enable shader core and memory controller communication. TSVs are used to connect the L2 cache layer and the optical layer. TSVs also provide clocking, power, and ground signals. The optical layer includes off-chip laser source, coupler, resonators, and optical interface to the off-chip memory. We use silicon waveguides to communicate between on-chip memory controllers and off-chip memory arrays with optimized DRAM chip organization.

The latest generation of Nvidia Kepler [1] GK110 series GPUs are manufactured using 28-nm process technology and require ~550-mm^2 silicon die area for fabricating 7.1 billion transistors to produce 15 shader cores [9]. Our 3-D stacked GPU microarchitecture design is based upon 16-nm technology [2]. We expect to have ~8.0 billion transistors on silicon die area of ~400 mm^2 in the shader core layer of nanophotonic 3-D stacked throughput architecture [10]. Figure 1 (top) shows the shader core layout, including an instruction cache, a thread scheduling unit, a thread dispatch unit, 32 stream processing cores (SC), 48 KB of shared memory, 32K registers, three types (texture, constant, and global memory segment) of L1 data caches, 16 load/store units (LD/ST), four special functional units (SFU), multiple geometry units, multiple texture units, and L1–L2 cache interface. Our shader core layer die

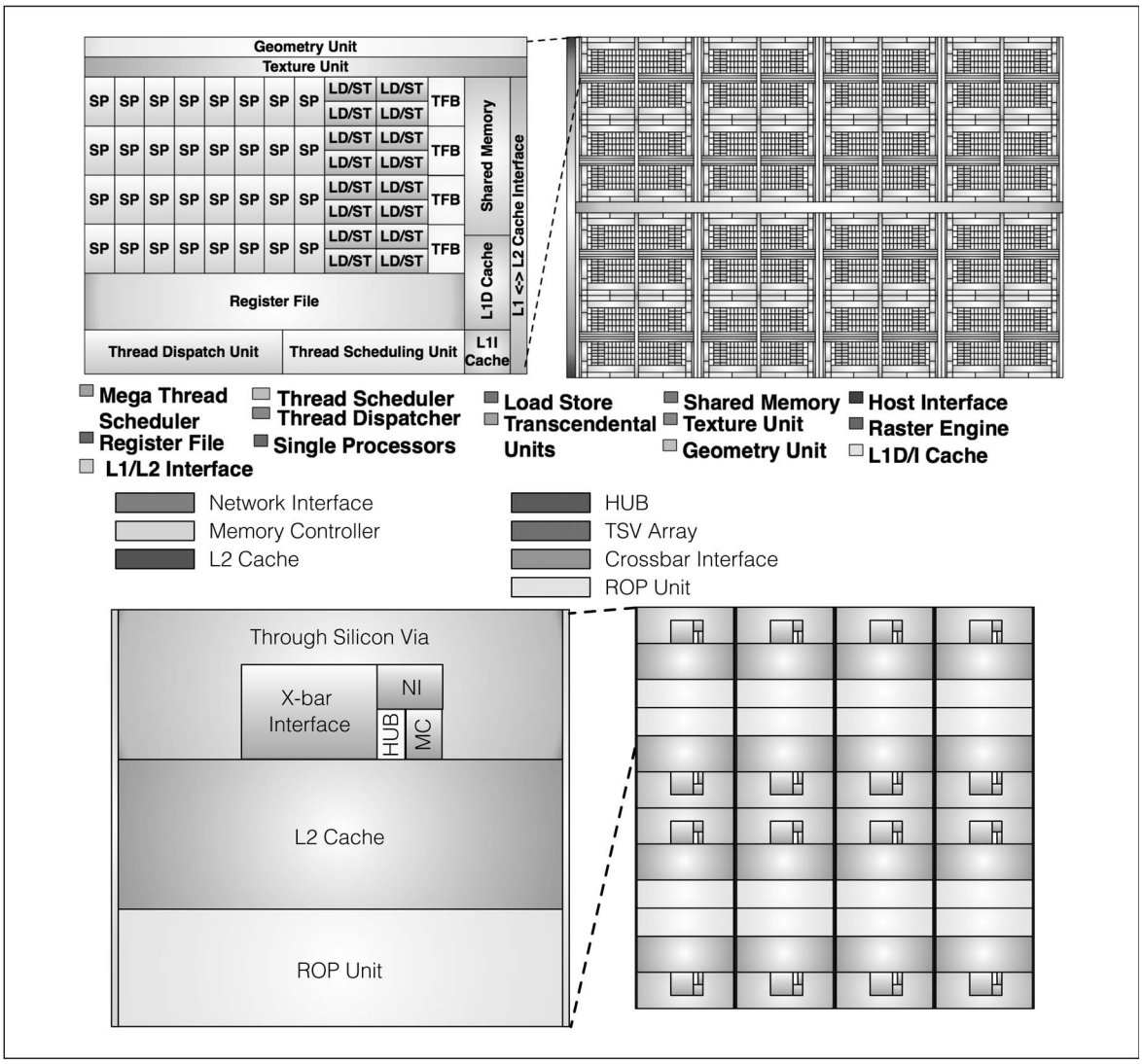

Figure 1. (Top) Shader core layer layout, which includes 64 shader cores (SC). Each SC has 32 stream processing cores (SP), 16 load/store units (LD/ST), four transcendental functional blocks (TFB), register file, thread scheduler/dispatcher, and L1 caches/shared memory. (Bottom) Cache/memory controller layer layout, which includes memory controller, TSV array, directory, HUB, network interface, and crossbar interface (area of four shader cores).

includes 64 shader cores consisting of 2048 shader pipelines along with 16 texture processing functional blocks. In our design, the thread scheduler is located at the center of the core layer layout and the host interface is placed at one side for better interfacing. Graphics-specific rasterization operations are performed by the rasterization engine, which is shared by eight shader cores. There are a total of eight raster engines available.

Our conservative estimate of shader core pipeline clock frequency is 2.0 GHz. To minimize thermal impact, the shader core layer, which consists of 2048 SIMD pipelines, is placed at the top of the 3-D stack (i.e., beneath the heat sink). The L2 data cache of size 8192 KB is placed underneath the shader core layer. Emerging throughput workloads would require more on-chip storage to reduce off-chip traffic and to enhance overall workload throughput. In our design, four shader cores are clustered together [texture processor clusters (TPCs)] to share a memory controller, network interface, and crossbar interface. This layer also incorporates render output units

(ROP) that perform pixel blending, anti-aliasing, and atomic memory operations, which are specific to the graphics applications. Figure 1 (bottom) depicts the simplified layout of the L2 cache/memory controller layer and a zoom-in view of individual L2 subsets. The L2 layer communicates with the shader core layer through interconnection network; hence, L1–L2 interface is implemented using an array of vertical TSVs connecting to the network layer. Each shader core cluster has its own subset of unified L2 cache of size 512 KB. The local L2 subset for any shader cluster implements direct shader core to local L2 interfacing (face-to-face bonding) and avoids NoC traversal. In GPGPU, often intershader communication is limited and if emerging workloads are optimized for accessing local L2 subset during higher level cache misses, interconnect traffic will only occur during nonlocal L2 subset accesses. However, it will not reduce intershader coherence traffic if present. Therefore, a separate layer of optical interconnection network implements a crossbar interface to connect nonlocal L2 subsets across different clusters. A memory controller is associated with the L2 cache subset of each cluster. The memory controllers are connected to the off-chip memory modules using optical interconnection.

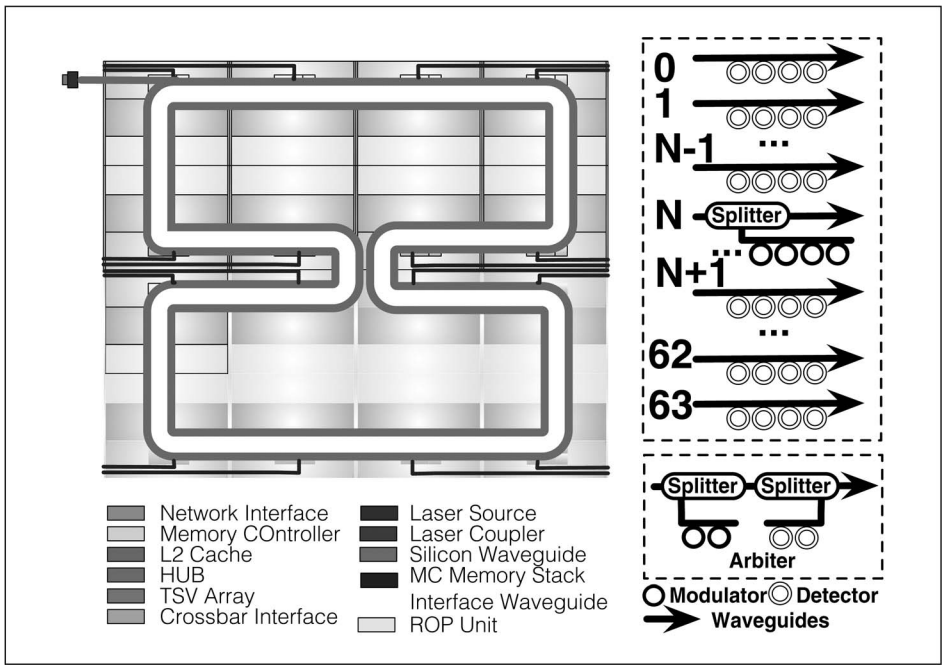

Figure 2. (Left) Optical layer: 64 waveguides, optical crossbar (right), off-chip laser source/coupler, DIMM. (Right) Crossbar module structure.

In our design, throughput memory accesses follow these steps: When L1 cache miss occurs in the shader core, the memory request maps the address to the corresponding L2 subset. If the L1 miss address is mapped to the local L2 subset, requests do not traverse the on-chip network; instead it uses the fast face-to-face L1–L2 interface. In throughput workloads, core-to-core communication is limited and an application developer in several cases can control off-chip data that is mapped to nonlocal L2 subset. This behavior of GPGPU workloads justifies direct interfacing of the local L2 subset. In case the address is mapped to L2 subset of another cluster, the request has to traverse the optical interconnect. Since optical interconnect carries L1 cache miss traffic, in a highly data-intensive GPGPU application with arbitrary memory access pattern, the memory misses will congest the on-chip network. Moreover, interkernel data communication between the simultaneously executing threads in GPGPU applications and in future shader core coherence traffic can generate heavy L2 cache traffic. In fact, we anticipate that, with simultaneous execution of multiple kernels in emerging GPGPU workloads, the off-chip memory request traffic will surpass the L2 coherence traffic. We expect that the high-bandwidth, low-latency optical interconnect-based throughput architecture will provide an attractive solution to this problem. Figure 2 shows the simplified layout of the optical interconnect layer and crossbar interface design, respectively. Traditionally, throughput architectures require wider off-chip memory interface to meet the data request of thousands of concurrently running threads. Efficient interfacing of optically connected off-chip memory motivates the layout of the silicon waveguide. We use many-writer–single-reader (MWSR) photonic crossbar to connect the 16 TPCs using 128 waveguides (16 channels $\times$ 4 waveguides/channel $\times$ 2 directions) in both directions (shader core to memory controller and memory controller to shader core). Each waveguide is

capable of propagating 64 wavelengths (64 b per clock, i.e., 32 B per clock period in each direction) of light using DWDM. With 5-GHz clock, all 16 channels can achieve raw interconnect bandwidth of 40.96 Tb/s. The dedicated floor for optical interconnect allows us to place these waveguides in the serpentine structure (Figure 2, left), where the waveguide pitch is as low as 5.5 μm [11]. The network implements optical token-based arbitration to realize the MWSR crossbar [also the token-less single-writer–multiple-reader (SWMR) crossbar is examined]. All types of traffic (read request, write request, and read reply) enter the network through an interface buffer. In the MWSR crossbar, a reserved optical token leaves the source node and travels along with the tail of the last flit of the packet until it reaches destination. Once transfer is completed, the free token traverses the waveguide until another node grabs it. The destination nodes identify tokens. On the contrary, in SWMR, all the optical channels are writable by only one source node and there are multiple channels per destination node. The SWMR crossbar behaves like a high-throughput low-latency electrical crossbar with variable source-to-destination delay.

Our optical interconnect incorporates the MWSR-based crossbar that requires arbitration mechanism, which is used in token ring LAN system arbitration to resolve the read requests sent by multiple shader nodes. Unlike multithreaded CPU workloads, throughput workloads execute the same instruction simultaneously (SIMD lockstep execution paradigm) in different shader cores that request data from different L2 subsets. Since the shader core count is comparably more than the core count in the multi-core system, usually average token grab latency for channel reservation of a requesting shader is higher. An alternative to MWSR is the SWMR crossbar, which is relatively power hungry (more laser power is required to drive (node count −1) photodetectors as compared to one photodetector in MWSR). SWMR incorporates a separate dedicated channel for each source channel that sends packets to a destination; hence, no channel arbitration mechanism is required. The static power of the crossbar is estimated as 0.33 W [11], [12]. The dynamic energy consumption is 200 fJ/b for transmission and the dynamic power for arbitration is negligible. Figure 3 shows vertically stacked layers in the chip.

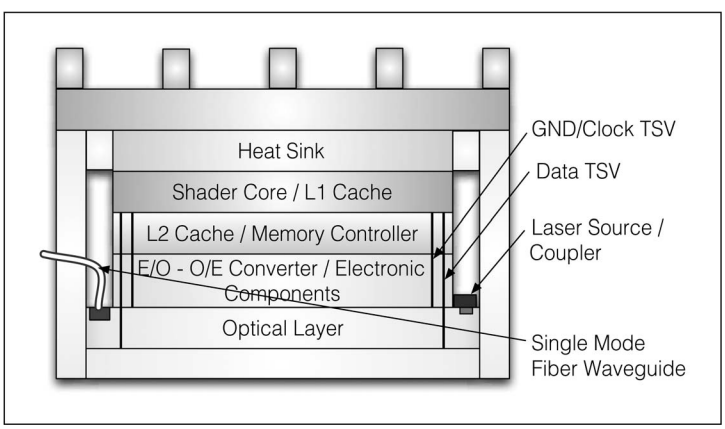

Figure 3. Cross-sectional schematic of the chip.

Evaluation

Experimental setup

In this study, we have used 17 real-world GPGPU workloads (Table 1a) from Nvidia CUDA SDK, Rodinia, and Parboil suites. Our evaluation is based on the cycle-based simulator GPGPU-Sim [13]. It has a modified version of the electrical interconnect simulator Booksim to simulate shader core and memory controller traffic. We have replaced the Booksim (Intersim in GPGPU-Sim) with the in-house optical interconnect simulator. It models the serpentine network that connects 16 TPC and 16 memory controllers using a crossbar and simulates flit-level optical traffic. We used two different system models presented in Table 1b. The baseline system has a state-of-the-art electrical network, electrically connected memory, and immediate postdenominator-based round-robin warp scheduler. Table 1c shows six optical crossbars implemented to evaluate our design.

We instrumented GPGPU-Sim and the optical network simulator to extract several hardware access statistics to calculate power. We developed an architecture level GPGPU power simulator based on heavily remodeled McPAT [14] that fits into GPU pipeline. The simulator interfaces with GPGPU-Sim to calculate runtime power of major components of GPU, including an electrical NoC. The NoC model is composed of single links and a traditional four-staged router with flit buffers, arbiters, and a crossbar. The power simulator simulates the electrical NoC with 22-nm technology. We further scale the NoC parameters down to 16 nm. For the optical NoC power consumption, we used the statistics

reported in [15] and [16], as summarized in Table 1d [21]–[23]. The energy coupling loss of the laser source into the chips is reported to be ∼8 dB [20]. In order to achieve a bit error rate of 10^{-15}, each photodetector requires 5-μW power [12] to successfully receive data, and each modulator consumes 200 fJ of power to modulate one bit of data [24]. To model the trimming issue, we assume 1-μW heating power per ring per Kelvin.

Analysis

Figure 4 compares the different designs in terms of shader core to memory controller latency (flit level) and memory controller to shader core latency. Note that, in Figure 4 (top), SWMR photonic interconnect experiences relatively lower latency (81%–90% reduction) for read and write requests with respect to the MWSR-based crossbar (59%–66% reduction). The MWSR-based crossbar uses the token to implement the mutually exclusive access to the destination node. On the contrary, the SWMR-based crossbar has multiple dedicated channels to the destination nodes from a source node. Hence, in MWSR, a source node may wait even when the destination channel is idle due to unavailability of the destination token. This latency is completely hidden in the SWMR-based crossbar. However, SWMR is different from a traditional electrical crossbar because the latency in the SWMR-based optical crossbar depends upon the distance between the source and the destination node in the waveguide. In case of GPU microarchitecture, this phenomenon exhibits adverse effects when memory traffic becomes bursty. Memory controller to shader core traffic latency of the MWSR crossbar in Figure 4 (bottom) is significantly lower than (49%–95% reduction) the electrical crossbar. Comparatively, the SWMR crossbar shows lower latency (52%–94% reduction) with respect to the electrical baseline. Memory-intensive benchmarks such as BFS (90%), MM (87%), DG (77%), RAY (72%), and 64H (75%) exhibit maximum benefit from optical network. All these benchmarks have a large number of read and write memory accesses. As we increase the channel bandwidth, memory controller to shader core latency keeps on decreasing. With the increase in the flit size, there will be a smaller number of flits allocable in the network interface buffer. Hence, average wait time in buffer is decreased. Close examination of the network statistics also reveals that in MWSR the

Table 1 (a) GPGPU workloads (⋆: memory intensive). (b) GPU configuration (optical/electrical). (c) Optical crossbar configuration. (d) Optical loss in different components [12], [21]–[23].

GPGPU WORKLOADS (★: MEMORY INTENSIVE)

Workload (Abbr.)	Traffic (in 10^6 B)
Breadth First Search (BFS) ★	78.1
Fast Wash Transform (FWT)	18.0
Gaussian Elimination (GS)	0.7
Hot Spot (HS)	0.6
3D Laplace Solver (LPS) ★	23.4
Matrix Multiplication (MM) ★	120
Matrix Transpose (MT)	0.3
Path Finder (PF)	5.9
Ray Trace (RAY) ★	83.5
Speckle Reducing Anisotropic (SRAD)	0.5
Hybrid Sort (HY)	20.1
Similarity Score (SS) ★	150
Nearest Neighbor (NE)	2.0
Parallel Prefix Sum (SLA)	0.7
Adv. Encryption Std (AES)	0.9
64 Bin Histogram (64H) ★	29.4
Galerkin Solver (DG) ★	93.0

(a)

GPU CONFIGURATION (OPTICAL/ELECTRICAL)

Parameters	Optical GPU	Electrical Base
Shader core	64	64
Thread batch size	32	32
SIMD pipeline	32	32
Memory controllers	16	16
MC queue size	32	32
L1/L2 cache	32/512 KB	32/512 KB
Register file	32K	32K
Shared memory	48KB	48KB
Threads per SM	2048	2048
Interconnect topo.	Crossbar	2D Mesh
Channel bandwidth	16/32/64B	16B
Clock(SHD/Icnt/MC)	2/5/1.6GHz	2/5/1.6GHz

(b)

OPTICAL CROSSBAR CONFIGURATION

Optical Crossbar Structure	Abbr.	Channel BW
Single Write Multiple Read	OXS-16B	16 Bytes
Single Write Multiple Read	OXS-32B	32 bytes
Single Write Multiple Read	OXS-64B	64 bytes
Multiple Write Single Read	OXM-16B	16 Bytes
Multiple Write Single Read	OXM-32B	32 bytes
Multiple Write Single Read	OXM-64B	64 bytes

(c)

OPTICAL LOSS IN DIFFERENT COMPONENTS [12, 21-23]

Optical coupler	Splitter	Waveguide loss
1 dB	0.2 dB	1.3 dB/cm
Filter through	Modulator insertion loss	
1E-4 ~ 1E-2 dB	0 ~ 1 dB	

(d)

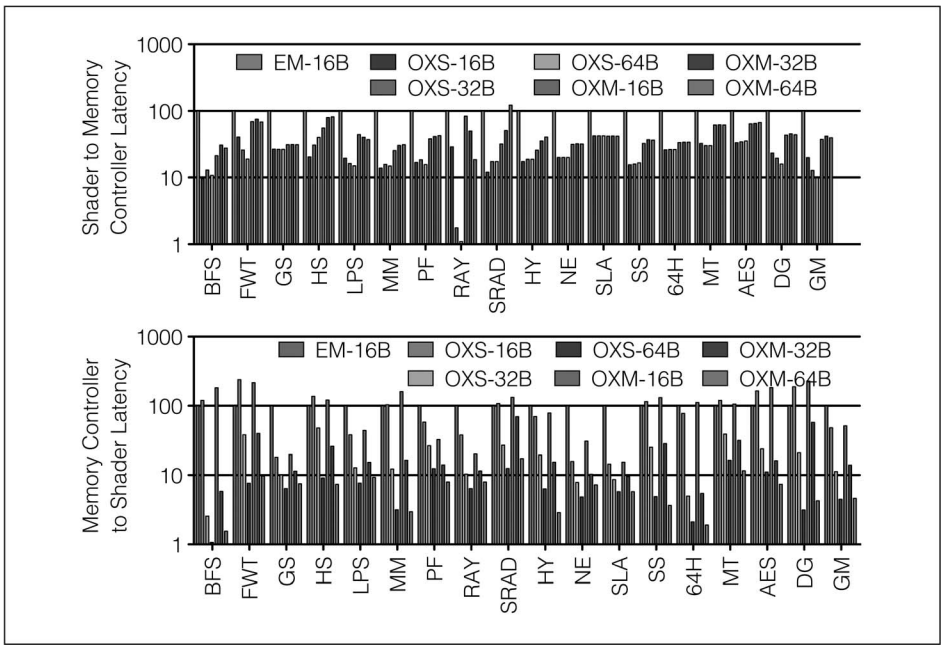

Figure 4. (Top) Shader to memory controller latency (normalized to electrical mesh 16-B, GM: geometric mean). (Bottom) Memory controller to shader latency (normalized to electrical mesh 16-B, GM: geometric mean).

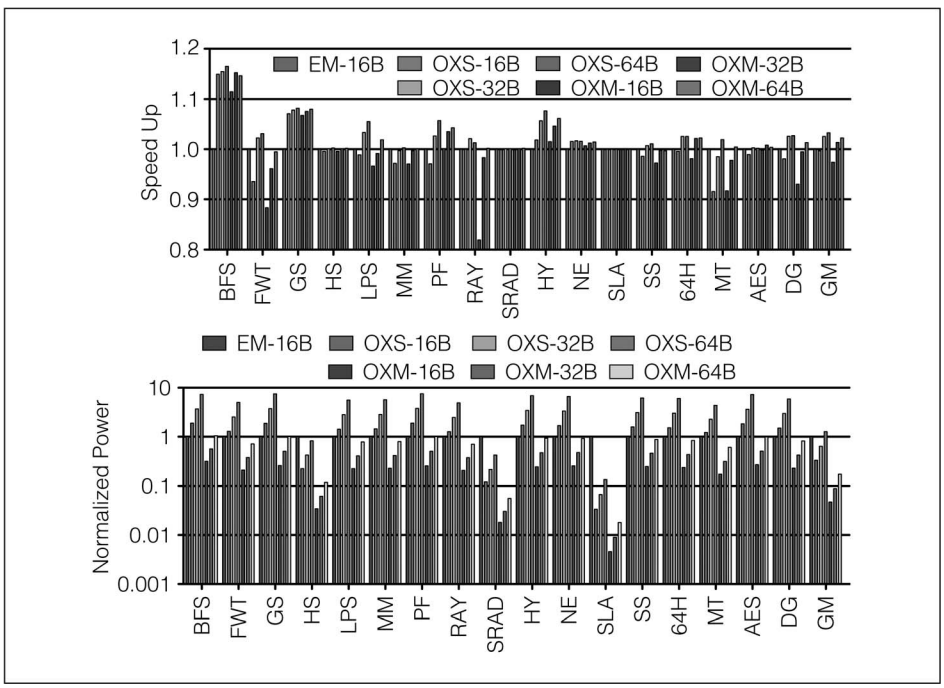

Figure 5. (Top) Speedup with respect to electrical mesh network 16-B channel bandwidth (GM: geometric mean). (Bottom) Network power consumption normalized to electrical mesh 16-B channel (GM: geometric mean).

token allocation delay largely dominates overall flit latency. Also, buffer wait time in SWMR is less than MWSR due to the lack of token allocation penalty experienced in MWSR.

Benchmarks such as BFS, SRAD, AES, PF, 64H, HY, and MM yield higher shader to memory controller latency when the channel bandwidth is increased. These benchmarks have more data read instructions (load) and less data write instructions (store). The data read request size is 8 B, which is smaller than the smallest channel bandwidth used. With the increase in channel bandwidth, more bytes are wasted and it does not reduce the overall flit count, which is the key reason behind latency improvement for larger channel bandwidth. Since memory controller to shader core traffic only has 64-B packets, it does not exhibit similar effect. With the increase in flit size, flit count reduces linearly and overall wait time in the buffer is also decreased.

Figure 5 show the speedup and power consumption characteristics of the different designs. On average, the SWMR 16-B crossbar has 0.04% lower performance than 16-B electrical mesh, but as we increase the bandwidth to 32 B and 64 B, the performance increase is 3% and 5%, respectively. Interestingly, benchmarks that are dominated by a large number of memory accesses demonstrate better performance increase. Maximum performance increase is observed in BFS (17%) with 64-B channel bandwidth. However, the MWSR-based crossbar shows almost 4% improvement

in performance, which is attributed to the large amount of delay experienced by the packets to grab the destination token.

The power consumption in the optical crossbar decreases drastically (Figure 5, bottom). On average, the SWMR crossbar shows almost 73% power saving and 1% performance degradation as compared to electrical 16-B mesh baseline. As expected, average MWSR power saving is comparatively higher (98%) with only 2% performance degradation. Memory-intensive workloads benefit most due to the optical interconnect. SRAD, SSLA, and HS show maximum power saving due to its heavy memory access in MWSR. We recommend the MWSR-based crossbar with 32-B channel bandwidth as the best solution that comes with 91% power saving and 1% average increase in performance. From our experimental results, the static power constitutes significant portion of the whole network power. This result is in accordance with [24].

Related work

Architectures such as Corona [2], Firefly [4], Phastlane [3], and Flexishare [5] have demonstrated 3-D stacked multicore chip with optical token-based arbitration, electrical intracluster communication with optical crossbar-based intercluster communication, switch-based photonic interconnect, electrical–optical token stream-based arbitration for channel assignment, and credit distribution in the NoC, respectively. However, investigation of the power and performance of silicon-nanophotonic-assisted GPU is still lacking. In this paper, we make the first step in exploring photonics-enabled GPU microarchitecture that integrates shader cores, caches, and off-chip optical memory interfaces in the different layers of 3-D stacked chip. We have customized [17] to design off-chip memory stacking and developed sequential and concurrent multiple memory access scheduling. Although Al Maashri et al. [18] have explored the 3-D stacked cache architecture in multilayer GPU implementation, they have not addressed the issues of NoC congestion or off-stack memory access bottleneck. Morris et al. in [19] proposed a 3-D photonic CPU interconnect that can dynamically reconfigure without system intervention and allocate channel bandwidth.

We have proposed a 3-D stacked nanophotonic throughput architecture that provides 91% average reduction in NoC dynamic power (in turn heat) with 4% average increase in performance and up to 95% average NoC latency reduction as compared to the state-of-the-art electrical on-chip interconnect-based GPU. Our experiments reveal that, in MWSR-type interconnects, the overall flit latency is largely dominated by the token ring delay. In contrast, SWMR suffers from buffer wait latency. Furthermore, with increasing off-chip memory demand, we expect that the proposed optically connected throughput architecture will attain further improvements. ■

References

[1] Nvidia Corporation, "NVIDIA's next generation CUDA compute architecture: Kepler GK110," white paper. [Online]. Available: http://www.nvidia.com/content/PDF/kepler/NVIDIA-Kepler-GK110-Architecture-Whitepaper.pdf

[2] D. Vantrease et al., "Corona: System implications of emerging nanophotonic technology," in *Proc. 35th Annu. Int. Symp. Comput. Architect.*, Washington, DC, USA, pp. 153–164, DOI: 10.1109/ISCA.2008.35.

[3] M. J. Cianchetti, J. C. Kerekes, and D. H. Albonesi, "Phastlane: A rapid transit optical routing network," in *Proc. 36th Annu. Int. Symp. Comput. Architect.*, Austin, TX, USA, 2009, pp. 441–450.

[4] Y. Pan et al., "Firefly: Illuminating future network-on-chip with nanophotonics," in *Proc. 36th Annu. Int. Symp. Comput. Architect.*, New York, NY, USA, pp. 429–440, DOI: 10.1145/1555754.1555808.

[5] Y. Pan, J. Kim, and G. Memik, "FlexiShare: Channel sharing for an energy-efficient nanophotonic crossbar," in *Proc. IEEE 16th Int. Symp. High Performance Comput. Architect.,* Jan. 2010, DOI: 10.1109/HPCA.2010.5416626.

[6] L. Eeckhout, H. Vandierendonck, and K. De Bosschere, "Designing computer architecture research workloads," *Computer*, vol. 36, no. 2, pp. 65–71, Feb. 2003.

[7] K. Hoste and L. Eeckhout, "Microarchitecture-independent workload characterization," *IEEE Micro*, vol. 27, no. 3, pp. 63–72, May/Jun. 2007.

[8] S. Beamer et al., "Re-architecting DRAM with monolithically integrated silicon photonics," in *Proc. IEEE 37th Int. Symp. Comput. Architect.*, New York, NY, USA, pp. 129–140, DOI: 10.1145/1815961.1815978.

[9] P. V. Bolotoff, "A quick analysis of the NVIDIA Fermi architecture," 2010. [Online]. Available: http://alasir.

com/articles/nvidia_fermi_architecture/gt200_gt300_architecture.shtml

[10] N. Goswami, A. Verma, and T. Li, "GPU-PowerSim: A power simulation framework for throughput processors," 2012. [Online]. Available: http://www.ideal.ece.ufl.edu/main.php?action=gpu-powersim

[11] A. Joshi et al., "Silicon-photonic clos networks for global on-chip communication," in *Proc. 3rd ACM/IEEE Int. Symp. Networks-on-Chip*, May 2009, pp. 124–133, DOI: 10.1109/NOCS.2009.5071460.

[12] A. Melloni, M. Martinelli, G. Cusmai, and R. Siano, "Experimental evaluation of ring resonator filters impact on the bit error rate in non return to zero transmission systems," *Opt. Commun.*, vol. 234, pp. 211–216, 2004.

[13] A. Bakhoda, G. L. Yuan, W. W. L. Fung, H. Wong, and T. M. Aamodt, "Analyzing CUDA workloads using a detailed GPU simulator," in *Proc. Int. Symp. Performance Anal. Syst. Softw.*, 2009, DOI: 10.1109/ISPASS.2009.4919648.

[14] J. A. Stratton et al., "Parboil: A revised benchmark suite for scientific and commercial throughput computing," in *Cntr. Reliable High-Performance Comput.*, 2012.

[15] NVIDIA Developer Zone, *N-Queens Solver*. [Online]. Available: http://forums.nvidia.com/index.php?showtopic=76893

[16] S. Pai, M. J. Thazhuthaveetil, and R. Govindarajan, "Improving GPGPU concurrency with elastic kernels," in *Proc. 18th Int. Conf. Architect. Support Programm. Lang. Oper. Syst.*, Houston, TX, USA, 2013, pp. 407–418.

[17] NVIDIA CUDA Zone, "CUDA code samples," 2013. [Online]. Available: https://developer.nvidia.com/gpu-computing-sdk

[18] A. Al Maashri, G. Sun, X. Dong, V. Narayanan, and Y. Xie, "3D GPU architecture using cache stacking: Performance, cost, power and thermal analysis," in *Proc. IEEE Int. Conf. Comput. Design*, Lake Tahoe, CA, USA, 2009, pp. 254–259.

[19] R. Morris, A. K. Kodi, and A. Louri, "Dynamic reconfiguration of 3D photonic networks-on-chip for maximizing performance and improving fault tolerance," in *Proc. 45th Annu. IEEE/ACM Int. Symp. Microarchitect.*, 2012, pp. 282–293.

[20] H. Takesue, N. Matsuda, E. Kuramochi, W. J. Munro, and M. Notomi, "An on-chip coupled resonator optical waveguide single-photon buffer," *Nature Commun.,* vol. 4, Oct. 2013, article 2725, DOI: 10.1038/ncomms3725.

[21] V. R. Almeida, C. A. Barrios, R. R. Panepucci, and M. Lipson, "All-optical control of light on a silicon chip," *Nature*, vol. 431, pp. 1081–1084, 2004.

[22] Q. Xu, S. Manipatruni, B. Schmidt, J. Shakya, and M. Lipson, "12.5 Gbit/s carrier-injection-based silicon micro-ring silicon modulators," *Opt. Exp.*, vol. 15, pp. 430–436, 2007.

[23] Z. Li, R. Zhou, and T. Li, "Exploring high-performance and energy proportional interface for phase change memory systems," in *Proc. IEEE 19th Int. Symp. High Performance Comput. Architect.*, 2013, pp. 210–221.

[24] G. T. Reed, G. Mashanovich, F. Y. Gardes, and D. J. Thomson, "Silicon optical modulators," *Nature Photon.*, vol. 4, no. 8, pp. 518–526, 2010.

Nilanjan Goswami is an Architecture and Modeling Engineer at a leading product development company. His research interests include emerging-technology-based throughput processor design, power-performance co-optimization of throughput core architecture, interconnect, and renewable-energy-based throughput architectures. Goswami has a PhD in electrical and computer engineering from the University of Florida, Gainesville, FL, USA. He is a member of the IEEE.

Zhongqi Li is currently with Qualcomm Inc., San Diego, CA, USA, where he works as Adreno GPU Performance Architect for the Snapdragon mobile processor. Prior to that, he worked in Marvell Semiconductor as a Processor Performance Engineer (intern) for Marvell's next-generation ARM processor. His current research interests include CPU/GPU architecture, network-on-chip, and multicore processor system. Li has a BS and an MS from the University of Electronic Science and Technology of China, Chengdu, China (2006 and 2009, respectively) and a PhD from the University of Florida, Gainesville, FL, USA (2012).

Ramkumar Shankar is a GPU Architecture Engineer at Qualcomm Technologies, Inc., San Diego, CA, USA, where he works on the next-generation GPUs for mobile computing. His research focuses on GPGPU architecture optimization and workload characterization. Shankar has an MS in electrical and computer engineering from the University of Florida, Gainesville, FL, USA.

Tao Li is an Associate Professor in the Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL, USA. His research interests include computer architecture, microprocessor/memory/storage system design, virtualization technologies, energy-efficient/sustainable/dependable data center, cloud/big data computing platforms, the impacts of emerging technologies/applications on computing, and evaluation of computer systems. Li has a PhD in computer engineering from the University of Texas at Austin, Austin, TX, USA.

■ Direct questions and comments about this article to Nilanjan Goswami, ECE Department, University of Florida, Gainesville, FL 32611 USA; nil@ufl.edu.