Integrating Nanophotonics in GPU Microarchitecture

Nilanjan Goswami[1] Zhongqi Li[2] Ajit Verma[3] Ramkumar Shankar[4] Tao Li[5]

{nil[1], zhongqi[2], aejeet[3]}@ufl.edu, [5]taoli@ece.ufl.edu [4]ramkumar@qualcomm.com

University of Florida, Gainesville, USA

Qualcomm Inc., San Diego, USA

ABSTRACT

As high-performance computing device, the GPU has exposed bandwidth and latency bottlenecks in on-chip interconnect and off-chip memory access. To eliminate such bottlenecks, we employ silicon nanophotonics and 3D stacking technologies in GPU microarchitecture. This provides higher communication bandwidth and lower latency signaling mechanisms at reduced power. Furthermore, to insulate the performance of the GPU compute cores from the interconnect bottlenecks we propose a novel interconnect aware thread scheduling scheme to alleviate the traffic congestion. We evaluate a 3D stacked GPU with 2048 SIMD cores having photonic interconnect. The photonic multiple-write-single-read crossbar network with 32B channel bandwidth on average, achieves 96% power reduction. We anticipate that for emerging workloads and microarchitectures the implications of the proposed ideas are far reaching in terms of power and performance.

Categories and Subject Descriptors

C.1.2 [**Computer Systems Organization**]: Multiprocessors – Interconnection Architectures

General Terms

Design, performance.

1. INTRODUCTION

Recent advancements in CMOS process based silicon nanophotonics have substantially mitigated the power, latency and bandwidth problem. 3D stacking technology provides low latency and high bandwidth cross-layer communication in a compact form. With significant bandwidth demand in GPU, 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 GPU shader cores and memory interface in 3D-stacked multi-layer chip seems to be an attractive alternative. In this paper, we explore a novel 3D-stacked GPU microarchitecture based on silicon nanophotonics technology. The GPU chip has a shader core layer, a cache layer, and a built-in optically connected on-chip network layer. The optical network layer possesses dense wavelength division multiplexed (DWDM) high-speed interconnect for core to memory communication. In addition, on-chip memory controllers communicate with the off-chip memory using DWDM links. Furthermore, instruction scheduling (here issue of memory requests) has direct impact on on-chip interconnect traffic. Hence, we propose a novel instruction scheduling policy that helps in alleviating interconnect traffic congestion.

PACT'12, September 19–23, 2012, Minneapolis, Minnesota, USA.
ACM 978-1-4503-1182-3/12/09.

2. MOTIVATION

Emerging GPGPU and graphics workloads are expected to have more computation load for shader cores and will exert more traffic to on-chip interconnect and memory controller. Figure 1 shows a Hotspot 5.0 based temperature profile (Matrix Transpose benchmark) of a GPU with 16 shader cores and 6 memory controllers connected using 2D-mesh interconnect. It asserts that on-chip network and memory controller components dissipate maximum power (in turn heat). Moreover, according to ITRS projection 80% of chip power will be consumed by interconnect. We propose novel power-efficient photonic GPUs that leverage optical medium for on/off-chip data transfer.

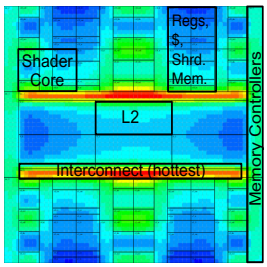

Figure 1: GPU chip temperature profile.

Our investigation of interconnect load for various GPGPU workloads using GPGPU-Sim simulator [1] reveals bursty traffic behavior. This suggests that the GPU shader cores are executing several memory instructions at a stretch for large number of threads. For example, shader core (S_i) cache miss generates interconnect traffic when it executes a thread-batch (Nvidia: *warp* and AMD: *wavefront*) (W_k) and encounters a memory instruction (I_p). If memory access pattern is random, then serving a cache miss will not result in a cache hit for the rest of the threads in the warp. Moreover, in case of memory miss, a new warp is scheduled to execute on the SIMD pipeline due to SIMD lock-step execution policy. For a large number of threads, it is expected that the next warp will also execute instruction I_p. Due to random memory access pattern, repeated execution of memory instruction warps will create on-chip network congestion. To avoid such scenarios, the warps should be scheduled in interleaved fashion, so that an instruction causing on-chip network congestion is avoided when non-memory instructions are available in the thread scheduler. We have explored a hardware solution to the shader core thread scheduler.

3. NANOPHOTONIC GPU DESIGN

3.1 Shader Core Microarchitecture

Silicon nanophotonics technology enabled GPU uses multiple DWDM based waveguides to optically connect multiple shader cores and off-chip memory arrays. To meet area constraints and improve heat dissipation capabilities, we separate shader core layer and L2 cache/memory interface layer and use face-to-face

bonding to connect the two layers. We incorporate a separate silicon nanophotonic optical layer that transfers the optical signals to enable shader core and memory controller communication. We use through silicon vias (TSV) to connect the L2 cache layer and optical layer. TSVs also provide clocking, power and ground signals. The optical layer includes laser source, coupler, resonators and optical interface to the off-chip memory. We use silicon nanophotonic waveguides to communicate between on-chip memory controllers and off-chip memory arrays with optimized the DRAM chip organization. Figure 2 shows the simplified layout of the GPU optical interconnection network layer design. Efficient interfacing of optically connected off-chip memory motivates layout of the silicon waveguide. We use many-writer-single-reader (MWSR) photonic crossbar to connect the clusters using 128 waveguides (16 channels × 4 waveguides/channel × 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 bits per clock, i.e. 32B per clock period in each direction) of light using DWDM. With 5 GHz clock, all the 16 channels can achieve raw interconnect bandwidth of 40.96 Tb/s. The dedicated floor for optical interconnect in 3D GPU allows us to place these waveguides in the serpentine structure, where the waveguide pitch is as low as 5.5µm. The optical interconnect incorporates MWSR based crossbar that requires arbitration mechanism, which are used in token ring LAN system arbitration to resolve the read requests sent by multiple shader nodes. An alternative to MWSR is single-writer-multiple-read (SWMR) crossbar, which is relatively power hungry. The static power of the crossbar is estimated as 0.33W. The dynamic energy consumption is estimated as 200 fJ/bit for data transmission and the dynamic power for arbitration is negligible.

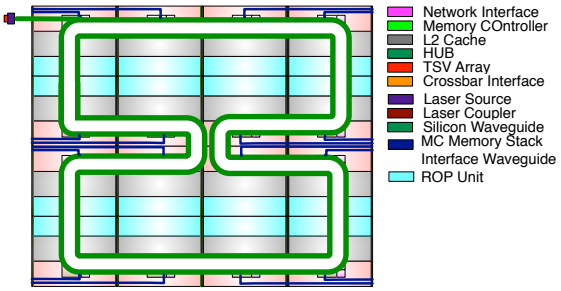

Figure 2: GPU Optical layer having 64 waveguides, optical crossbar and off-chip laser source/coupler.

3.2 Interconnect Aware Warp Scheduler

GPU runs thousands of threads (with same instruction sequence) concurrently in single instruction multiple thread (SIMT) fashion. Due to control flow divergence and memory miss divergence, threads running within a shader core progress unevenly. Hence, at any thread scheduling cycle, several ready instructions of different types (ALU/memory access) are present. The availability of different types of instructions at the scheduling cycle varies due to the instruction mix of the GPGPU program. Based on the last level cache miss information, we design a novel thread scheduler that can interleave the memory access requests in between arithmetic operations. In addition to reducing on-chip optical network congestion, the proposed scheme also hides the memory access latency of various on-chip memories whenever possible. We carryout the operation in two stages: warp classification and deterministic scheduling. When the warp with a certain PC completes the decode stage of the shader core pipeline, the instruction classification for all the other warps with the same PC is obtained. Using the information, our scheme deterministically saves the memory access latencies of several cache (off-chip memory, texture, and constant) misses by knowing the instruction classification at the scheduling stage.

4. RESULTS AND CONCLUSION

Figure 3 shows the speedup and power consumption characteristics of the different designs. On average, SWMR crossbar shows almost 87% power saving and 1% performance degradation as compared to electrical 16B mesh baseline. As expected, average MWSR power saving is comparatively higher (98%) with only 2% performance degradation. Memory intensive workloads benefits most due to the optical interconnect. BFS shows maximum power saving of 89% due to its heavy memory access in MWSR. We recommend MWSR based crossbar with 32B channel bandwidth as the best solution that comes with 96% power saving and 1% average increase in performance. On average, SWMR 16B crossbar has 0.04% lower performance than 16B electrical mesh, but as we increase the bandwidth to 32B and 64B the performance increase is 3% and 5% respectively. Maximum performance increase is observed in BFS (17%) with 64B channel bandwidth. However, MWSR based crossbar shows almost 4% improvement in performance, which is attributed to the large amount delay experienced by the packets to grab the destination token.

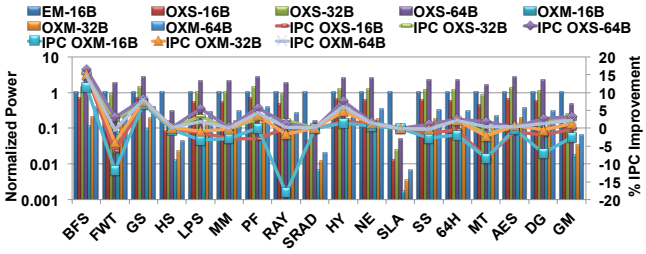

Figure 3: Power/performance impacts of GPGPU workloads.

Interconnect aware warp scheduling improves performance of BFS, HS, NE, AES, SRAD, and NE over round-robin scheduling. BFS shows maximum improvement of 9.7%, which is a direct effect of 86% reduction in memory miss stall. HS is less memory intensive workload than BFS; 95% memory miss stall reduction in HS provides 6.1% IPC improvement. Experiments reveal that overall memory latency largely depends upon interconnect; hence severe network congestion affects the overall performance.

5. ACKNOWLEDGEMENT

This work is supported in part by NSF grants 1117261, 0937869, 0916384, 0845721(CAREER), 0834288, 0811611, 0720476, and by Microsoft Research Trustworthy Computing, Safe and Scalable Multi-core Computing Awards. Authors acknowledge UF HPC center for providing computational resources.

6. REFERENCES

[1] Bakhoda, A.; Yuan, G.L.; Fung, W.W.L.; Wong, H.; Aamodt, T.M.; "Analyzing CUDA workloads using a detailed GPU simulator," *IEEE International Symposium on Performance Analysis of Systems and Software*, pp.163-174, 26-28 April 2009.