Performance Modeling and Measurement for a Custom Memory Cube Research Platform

Yu Zou, Gongyu Wang, Herman Lam, Vinayak Deshpande, Suvrat Tedia, Alan George

NSF Center for High-Performance Reconfigurable Computing (CHREC),

Department of Electrical and Computer Engineering,

University of Florida, Gainesville, FL

{zou, wangg, hlam, deshpande, suvrat17, george}@chrec.org

# INTRODUCTION

Big data era is putting increasingly strict performance and power requirements on data-intensive workloads. The emerging 3D-stacked memory devices, such as the hybrid-memory cube (HMC) have reignited the decade-old research on near-memory processing as an attempt to address such requirements. In particular, custom memory cube (CMC) that integrates customized data-processing logic into the base logic layer of HMC has received much attention. Many architectures, algorithms, and memory-internal connection topologies [1][2][3][4] have been developed towards CMC. However, these existing works are mostly based on software-simulated 3D-stacked DRAM models. In this extended abstract, we present a methodology to model and measure performance of CMC architectures and apps, using our previous work on hardware-in-the-loop CMC research platform [5]. Details on the performance modeling and measurement are discussed in Section 2. In Section 3, we report our progress and current measured performance.

# MODELING AND MEASUREMENT

As shown in Figure 1, we identify 5 performance points (A, B, C, D, and E), within the notional CMC architecture:

* *latency A*: round-trip CMC latency + data transfer delay
* *latency B*: round-trip CMC latency
* *latency C*: CMC custom logic latency
* *latency D*: round-trip latency of HMC internal data access
* *latency E*: round-trip latency of DRAM layers
* *latency D\_E*: latency between HMC switch and vault controller (VC)

which are critical for design-space exploration (DSE) of CMC architectures and apps.

We model these performance points on our CMC research platform by mapping them to the corresponding points A’, B’, C’, D’, and E’, as shown in Figure 2. Since HMC-internal points D’ and E’ cannot be directly measured in hardware, we establish a mathematical equation to calculate *latency D’*, and two ancillary measurement points (F’ and G’) are therefore added.

Definitions of F’ and G’ are shown in below:

* *latency F’*: round-trip latency of FPGA’s access to HMC
* *latency G’*: round-trip latency of FPGA memory controller’s access to HMC

The complete mapping of measurement points is shown in below:

* *latency A = latency A’*
* *latency B = latency B’ – latency F’\_D’*
* *latency C = latency C’*
* *latency D = latency D’ = latency G’ – latency G’\_D’ = latency G’ – (latency VirtualMC + latency HMCC + latency HMC link)*
* *latency E = latency E’ = latency D’ - latency D’\_E’*

As shown in Figure 2, *latency A’* is measured in host code. *Latency B’*, *C’*, and *F’* are measured in hardware on the FPGA. *Latency G’* is measured in the performance monitor instantiated inside the low-level infrastructure. The hardware performance monitor is modified based on Micron’s original code. *Latency HMCC* and *latency HMC link* can be found respectively in HMCC specification [6] and HMC specification [7]. Long *latency VirtualMC* caused by TLB miss in address translation could be eliminated by modifying page table size to cover the whole HMC address space. *Latency D’\_E’* can be measured using current HMC simulators (e.g., HMC-Sim 2.0 [8])and be integrated into the research platform.



Figure 1. Concept diagram of notional CMC architecture



Figure 2. Concept diagram of CMC research platform

# PROGRESS AND FUTURE WORK

The progress we have made includes: (1) measurement-enabled research platform setup; (2) performance measurement for A’, B’, C’, and F’. We have conducted experiments to use the platform to monitor the performance of single-memory-operation benchmarks. Results are shown in Table 1. In the future, we will continue the work on and validation of performance measurement, and the notional CMC architecture model’s build-up.

Table 1. Single-memory-operation benchmarking results on the research platform

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Operations | A (ns) | B (cycles) | C (cycles) | F (cycles) |
| Single memory read | 1058559 | 168.4414 | 4 | 164.4414 |
| Single memory write | 1057302 | 175.6587 | 4 | 171.6587 |

# ACKNOWLEDGMENTS

This research was supported by CHREC members and by the I/UCRC Program at NSF under Grant No. IIP-1161022. We are grateful for the feedback, collaboration, and support provided by Laboratory of Physical Sciences, Micron, and Lawrence Livermore National Laboratory.

# REFERENCES

1. Gokhale, Maya, Scott Lloyd, and Chris Hajas. "Near memory data structure rearrangement." Proceedings of the 2015 International Symposium on Memory Systems. ACM, 2015.
2. Akin, Berkin, Franz Franchetti, and James C. Hoe. "Data reorganization in memory using 3d-stacked dram." ACM SIGARCH Computer Architecture News. Vol. 43. No. 3. ACM, 2015.
3. Ahn, Junwhan, et al. "A scalable processing-in-memory accelerator for parallel graph processing." ACM SIGARCH Computer Architecture News. Vol. 43. No. 3. ACM, 2015.
4. Farmahini-Farahani, Amin, et al. "Drama: An architecture for accelerated processing near memory." IEEE Computer Architecture Letters 14.1 (2015): 26-29.
5. Wang, Gongyu, et al. “A research platform for custom memory cube.” Workshop on Modeling & Simulation of Systems and Applications. University of Washington, Seattle, WA, Aug. 2016.
6. Hybrid Memory Cube Controller IP Core User Guide. https://www.altera.com/content/dam/altera-www/global/en\_US/pdfs/literature/ug/ug\_hmcc.pdf
7. HMC Specification 1.0. http://hybridmemorycube.org/files/SiteDownloads/HMC\_Specification%201\_0.pdf
8. Leidel, John D., and Yong Chen. "HMC-Sim 2.0: A Simulation Platform for Exploring Custom Memory Cube Operations." (2016).