# Benchmarking under the hood of OpenCL FPGA platforms

Kazutomo Yoshii
Argonne National Laboratory

Hal Finkel
Argonne National Laboratory

Franck Cappello
Argonne National Laboratory

## 1. INTRODUCTION

The end of Moore's law creates a significant turning point for computer architecture. Today, performance is largely limited by energy, power, and cooling. Heterogeneity and radical new architecture designs are keys to achieving higher energy proportionality. In mobile computing, heterogeneity is well adopted in system-on-chip designs (e.g., to improve battery life). In high-performance computing (HPC), graphics processing units (GPUs) are now being accepted as efficient heterogeneous accelerators for certain workloads. FPGAs are also attracting considerable attention because their reconfigurability allows the hardware to be customized for different workloads in order to attain both higher performance and energy efficiency. In addition, the advent of high-level synthesis technology such as OpenCL for FPGAs, competitive floating-point capability, and CPU-FPGA hybrid designs can lower major hurdles for the FPGA adoption process in HPC. Nevertheless, the characteristics of FPGAs particularly with high-level synthesis are little studied. Since FPGAs run slower (e.g., 200 MHz) than do CPUs/GPUs, it is crucial to exploit pipeline parallelism and avoid pipeline stalls due to memory operations.

In this paper, we present a brief summary of our OpenCL microbenchmark that primarily targets the data path between off-chip memory and OpenCL system-side implementation.

## 2. BENCHMARKING UNDER THE HOOD

Figure 1 depicts an example of an OpenCL FPGA data path. The data path may involve a hard memory controller (MC) and firmware-level memory interfaces, which may coalesce load requests, combine store requests, cache contents, and/or apply other optimization mechanisms. Such designs differ from platform to platform (possibly version to version). The specifications of all components are not guaranteed to be available; when specifications are fully disclosed, one needs hardware-level knowledge and significant efforts to understand their characteristics. Additionally, how OpenCL parameters will map memory interfaces into users' codes is unclear.

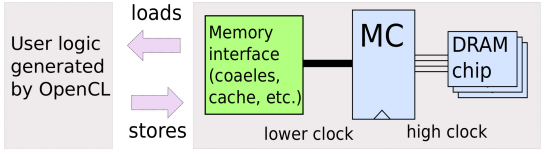


Figure 1: Example of under-the-hood data path

As of this writing, only a few benchmarks, such as Rodinia and CHO, take into account heterogeneous computing and target FPGAs; and none of them target the data path in a fine-grained manner. Therefore, we are designing our own OpenCL-based microbenchmark code, called *iabench*, with extensibility in mind. The code includes functionality to measure both performance and energy consumption of each OpenCL kernel. The predefined kernels included in *iabench* currently cover several memory access patterns, such as random access patterns and binary search patterns, combined with simple computations. Random memory access patterns are of great interest to the HPC community because emerging algorithms such as sparse matrix computation, graph algorithms, and memory-intensive Monte Carlo simulation are unfit for complicated deep memory hierarchy in CPUs/GPUs, thus potentially wasting energy.

We have tested *iabench* on reference platforms that include the Nallatech 385A Altera Arria 10-based FPGA accelerator board (Altera OpenCL 1.2) and Intel Xeon E5-2670 (Sandy Bridge) CPU (Intel OpenCL 1.2). Figure 2 is a comparison between the Nallatech FPGA board and the Intel CPU on an *iabench*'s OpenCL kernel[1] using *iabench*. The maximum memory bandwidth of the FPGA board is 34.1 GB/s while that of the Intel CPU is 54.1 GB/s. On this particular kernel, the (absolute) bandwidth of the FPGA board is slightly lower than that of the CPU. The reason is partially that the FPGA has fewer memory channels than does the CPU (two channels on the FPGA and four channels on the CPU). In terms of the percentage to the peak bandwidth, the FPGA shows better performance than does the CPU. The FPGA outperforms the CPU with regard to energy efficiency.

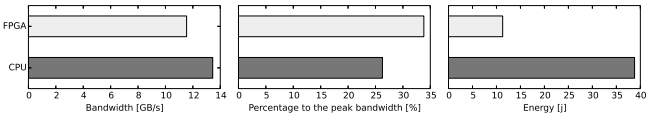


Figure 2: A comparison between FPGA and CPU

## 3. CONCLUSION

We have designed and implemented an initial version of *iabench* that targets the memory subsystem in OpenCL FPGA platforms. We expect that *iabench* can assist users in finding optimal OpenCL parameters and mapping applications into OpenCL FPGA more efficiently and can possibly identify bottlenecks in the current generation of hardware. We will continue to extend *iabench* (e.g., more kernels) and add other reference platforms (e.g., GPUs). We also plan to investigate custom data formats and explore new memory technologies such as hybrid-memory cubes.

## Acknowledgments

This material is based upon work supported by the U.S. Department of Energy Office of Science, under contract DE-AC02-06CH11357. We thank our summer intern Yingyi Luo from Northwester University who contributed to the benchmark project. We also thank Seda Ogrenci-Memik and Gokhan Memik at Northwester University for fruitful discussions.

[1]Its loop body picks a 64-byte aligned location, loads eight double-precision values, and computes four interpolations.

## Government License