ANL/ALCF-17/4

Evaluation of CHO Benchmarks on the Arria 10 FPGA using Intel FPGA SDK for OpenCL

Argonne Leadership Computing Facility

About Argonne National Laboratory

Argonne is a U.S. Department of Energy laboratory managed by UChicago Argonne, LLC under contract DE-AC02-06CH11357. The Laboratory's main facility is outside Chicago, at 9700 South Cass Avenue, Argonne, Illinois 60439. For information about Argonne and its pioneering science and technology programs, see www.anl.gov.

DOCUMENT AVAILABILITY

Online Access: U.S. Department of Energy (DOE) reports produced after 1991 and a growing number of pre-1991 documents are available free via DOE's SciTech Connect (http://www.osti.gov/scitech/)

Reports not in digital format may be purchased by the public from the National Technical Information Service (NTIS):

U.S. Department of Commerce
National Technical Information Service
5301 Shawnee Rd
Alexandria, VA 22312
www.ntis.gov
Phone: (800) 553-NTIS (6847) or (703) 605-6000
Fax: (703) 605-6900
Email: **orders@ntis.gov**

Reports not in digital format are available to DOE and DOE contractors from the Office of Scientific and Technical Information (OSTI):

U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831-0062
www.osti.gov
Phone: (865) 576-8401
Fax: (865) 576-5728
Email: **reports@osti.gov**

ANL/ALCF-17/4

Evaluation of CHO Benchmarks on the Arria 10 FPGA using Intel FPGA SDK for OpenCL

prepared by
Zheming Jin, Kazutomo Yoshii, Hal Finkel, Franck Cappello

Argonne Leadership Computing Facility, Argonne National Laboratory

May 23, 2017

Evaluation of CHO Benchmarks on the Arria 10 FPGA using Intel FPGA SDK for OpenCL

Introduction

The OpenCL standard is an open programming model for accelerating algorithms on heterogeneous computing system. OpenCL extends the C-based programming language for developing portable codes on different platforms such as CPU, Graphics processing units (GPUs), Digital Signal Processors (DSPs) and Field Programmable Gate Arrays (FPGAs). The Intel FPGA SDK for OpenCL is a suite of tools that allows developers to abstract away the complex FPGA-based development flow for a high-level software development flow. Users can focus on the design of hardware-accelerated kernel functions in OpenCL and then direct the tools to generate the low-level FPGA implementations. The approach makes the FPGA-based development more accessible to software users as the needs for hybrid computing using CPUs and FPGAs are increasing. It can also significantly reduce the hardware development time as users can evaluate different ideas with high-level language without deep FPGA domain knowledge.

Benchmarking of OpenCL-based framework is an effective way for analyzing the performance of system by studying the execution of the benchmark applications. CHO is a suite of benchmark applications that provides support for OpenCL [1]. The authors presented CHO as an OpenCL port of the CHStone benchmark. Using Altera OpenCL (AOCL) compiler to synthesize the benchmark applications, they listed the resource usage and performance of each kernel that can be successfully synthesized by the compiler.

In this report, we evaluate the resource usage and performance of the CHO benchmark applications using the Intel FPGA SDK for OpenCL and Nallatech 385A FPGA board that features an Arria 10 FPGA device. The focus of the report is to have a better understanding of the resource usage and performance of the kernel implementations using Arria-10 FPGA devices compared to Stratix-5 FPGA devices. In addition, we also gain knowledge about the limitations of the current compiler when it fails to synthesize a benchmark application.

Overview of the two FPGA devices

FPGA offers a wide variety of configurable memories, high-speed I/Os, logic blocks and routing. StratixV and Arria10 series of Intel FPGAs are two products for high-performance applications. Table 1 compares the technology and resource counts of the StratixV A7 FPGA device [2] on the P385-A7 FPGA accelerator card with those of the Arria10 GX1150 FPGA device [3] on the Nallatech 385A card.

The Arria10 device features 20-nm SoC process technology and operates at 0.95 V core voltage while the StratixV device uses 28-nm technology and operates at 0.9 V. Based on the results in Table 1, the Arria10 device has approximately 1.8X more Adaptive Logic Module (ALM) resources than the StratixV device. While the fabric

clocks are the same and M20K resources are close for both FPGAs, Arria 10 has almost 6X more DSPs.

Table 1. Device overview of two FPGAs

Device	Technology	Logic elements	ALMs	Register	M20K memory bits	DSPs	Fabric clock (MHz)
Stratix 5SGXMA7	28nm	622K	234,720	939K	50Mb	256	800
Arria 10AX115	20nm	1150K	427,200	1708800	54260Kb	1518	800

Nallatech 385A

Nallatech 385A is a PCIe-based FPGA accelerator card that features an Arria 10 GX1150 FPGA device, PCIe × 8 Generation 3 host interface, and two banks of 4GB DDR3 memory. The theoretical peak floating-point performance is 1.5 TFLOPS and the theoretical peak memory bandwidth approximately 34 GB/s.

Summary of CHO Kernels

There are 12 kernels in the CHO benchmarks. “dfadd”, “dfdiv”, “dfmul” and “dfsin” are implementations of double-precision floating-point addition, division, multiplication and sine function respectively. “adpcm” is an implementation of ITU G.722 Adaptive Differential Pulse-Code Modulation algorithm used in the encoding and decoding audio signals. The “adpcm” OpenCL kernel for FPGA is not available in the Git repository. “gsm” is an implementation of Linear Predictive Coding, a method of encoding good quality speech at a low bit rate. “jpeg” is an implementation of the JPEG still picture compression standards. “motion” is an implementation of the Motion vector decoding for MPEG-2. “aes” implements the AES symmetric-key crypto-system. “blowfish” is an implementation of the encryption function of a symmetric-key block cipher. “sha” implements the Secure Hash Algorithm, a cryptographic hash function. “mips” is an implementation of a simplified MIPS processors.

It should be noted that there are two OpenCL kernels for “dfdiv”. One kernel unrolls the loop 40 times and another one does not unroll the loop. We use the unrolled version. The “dfmul” OpenCL kernel contains the #pragma unroll” directive that completely unrolls the loops.

Experimental setup

In this work, a host system is set up with a 3.4 GHz Intel Xeon E5-2687W processor. The PCI Express provides Gen3×8 connection. Linux kernel 3.10.0 is installed as the operating system. We used the Intel’s FPGA SDK for OpenCL version 16.0.2 Pro Prime (i.e. current compiler) for the experimental results. We use the same OpenCL attributes as in [1] for each kernel.

We use the command "`aoc <kernel>.cl -o <kernel>.aocx`" to generate an FPGA binary configuration file for a kernel. The optimization mode for logic synthesis is "Performance", which is the default mode set in the Nallatech board support package. For each benchmark, the kernel name "kernel.aocx" must be renamed to avoid the ambiguous instances error generated during the synthesis phase of compilation.

When comparing the kernel implementations using the previous and current compilers, only the kernels that are successfully compiled using both compilers are selected.

Experimental results

There are four kernels that fail to be compiled successfully using the current compiler. Table 2 lists the kernels and the causes of failure. The "motion" kernel fails at the code generation phase, and "jpeg" fails at the synthesis phase, and "sha" and "dfadd" fail during the routing phase.

Table 2. Causes of kernel compilation failure using the current compiler

Kernel	Cause of failure
sha	Routing congestion
dfadd	Routing congestion
jpeg	Data array has more than 2^{23} bits
motion	Compiler optimization assertion error

Table 3 lists the resource usage and Fmax of each kernel that can be successfully compiled into a binary configuration file (aocx).

Table 3. FPGA resource usage and Fmax of the implemented kernels on Arria 10

Kernel	ALUTs	Memory bits	Registers	DSP	Fmax (MHz)
aes	239910	4106114	244880	1	115
blowfish	99355	9200030	193087	0	207
dfdiv	347683	4428376	572971	396	138
dfmul	92786	3910658	148997	240	211
dfsin	158615	5753952	373481	180	116
gsm	48273	3893166	92996	143	207
mips	53980	2761730	98259	10	265

Compared to the DSP results in [1], Figure 1 shows the current compiler tries to take advantage of the rich DSPs for the "dfdiv", "dfmul" and "dfsin" kernels.

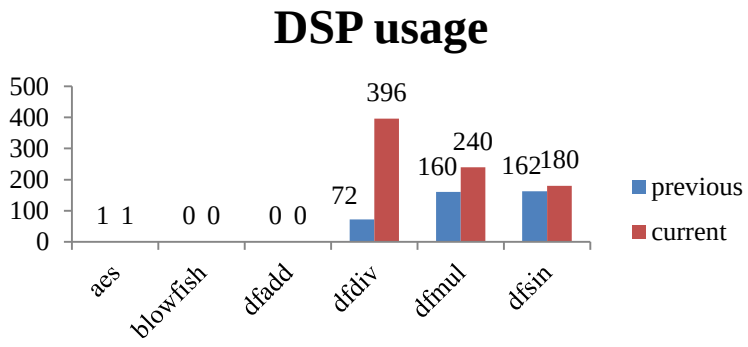

Figure 1 DSP usage of each kernel compiled with the previous and current

Figure 2 Registers usage of each kernel compiled with the previous and current compilers

Figure 3 Memory bits usage of each kernel compiled with the previous and current compilers

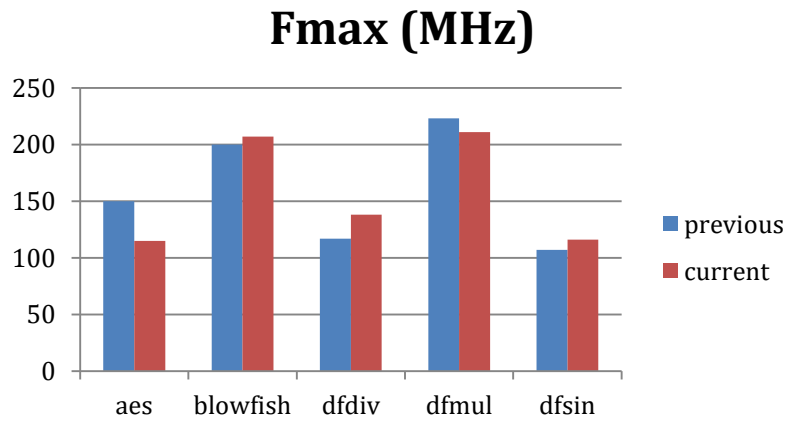

Figure 4 Fmax of each kernel compiled with the previous and current compilers

As shown in Figure 2, the new compiler decreases the registers usage for “aes” and “blowfish” by 17% and 28% respectively, but “dfdiv” requires 2X more registers.

For the memory bits usage, Figure 3 shows the new compiler significantly reduces the memory bits for “aes” and “dfsin”. However, the memory bits increases slightly for the other three kernels.

As shown in Figure 4, the “Performance” optimization mode does not generate each implementation with a higher Fmax (maximum frequency) for each kernel. Overall there is no significant Fmax improvement using the new compiler for Arria 10 FPGAs.

Finally, we run each kernel on the FPGA card and all kernels produce the correct results. Though the authors give a bar graph for the execution time of each kernel, the exact time is not displayed in the graph. Therefore, we list the kernel execution time in Table 4. For each kernel, we use the average execution time.

Table 4. Execution time of the implemented kernels on the Nallatech 385A card.

Kernel	Time (ms)
aes	0.09
blowfish	3.973
dfdiv	0.08
dfmul	0.013
dfsin	0.135
gsm	0.024
mips	0.33

Conclusion

The report compares the FPGA resource usage and performance of the kernels using the previous and current AOCL compilers. Compared to the previous compiler that fails to generate correct results for each kernel, the current compiler can generate a correct FPGA implementation for each kernel though “sha” and “dfadd” fail at the routing phase. The “dfdiv” kernel consumes significantly more resources using the current compiler. It is not clear if authors may use different OpenCL attributes for their synthesis result. Though the optimization mode for logic synthesis is “Performance”, there is no significant Fmax improvement using Arria 10 FPGA versus Stratix V FPGA. Since the execution time of almost all the kernels is less than 1 ms, the benchmark applications may be extended to support a longer run for each kernel.

Acknowledgments

This material is based upon work supported by the U.S. Department of Energy Office of Science, under contract DEAC02-06CH11357.

Reference

[1] Geoffrey Ndu, Javier Navaridas, Mikel Lujan , “CHO: Towards a Benchmark Suite for OpenCL FPGA Accelerators” In 3rd IWOCL International Workshop on OpenCL ; 11 May 2015-13 May 2015; Palo Alto, California, USA. 2015.
[2] Stratix V Device Overview
[3] Arria 10 Device Overview

Argonne Leadership Computing Facility
Argonne National Laboratory
9700 South Cass Avenue, Bldg. 240
Argonne, IL 60439

www.anl.gov