Bridging the Performance-Programmability Gap for FPGAs via OpenCL: A Case Study with OpenDwarfs

K. Krommydas*, A. Helal°, A. Verma°, W. Feng*

*Department of Computer Science, Virginia Tech

°Department of Electrical and Computer Engineering, Virginia Tech

VirginiaTech

SyNeRGy
http://synergy.cs.vt.edu/

OpenDwarfs
github.com/vtsynergy/OpenDwarfs

Goal: Identify Altera OpenCL optimizations to address the performance/programmability gap for FPGAs

Motivation

- FPGAs have been widely used across application domains
- Fast ***performance*** has come at the expense of ***programmability*** *(performance – programmability gap)*
- Adoption of the OpenCL programming model by FPGA vendors (Altera, Xilinx) can facilitate bridging the gap

Goal:

Identify optimization techniques, apply to the OpenDwarfs benchmark suite, and evaluate their effectiveness (performance, resource utilization)

Experimental Setup

OpenDwarfs Benchmarks Used

Dwarf	Benchmark	Input data
N-body Methods	GEM	nucleosome 80 1 0
Structured Grid	SRAD	2048x2048 FP matrix, 128 iterations
Dense Linear Algebra	LUD	2048x2048 FP matrix
Unstructured Grid	CFD	missile.domn.0.2M

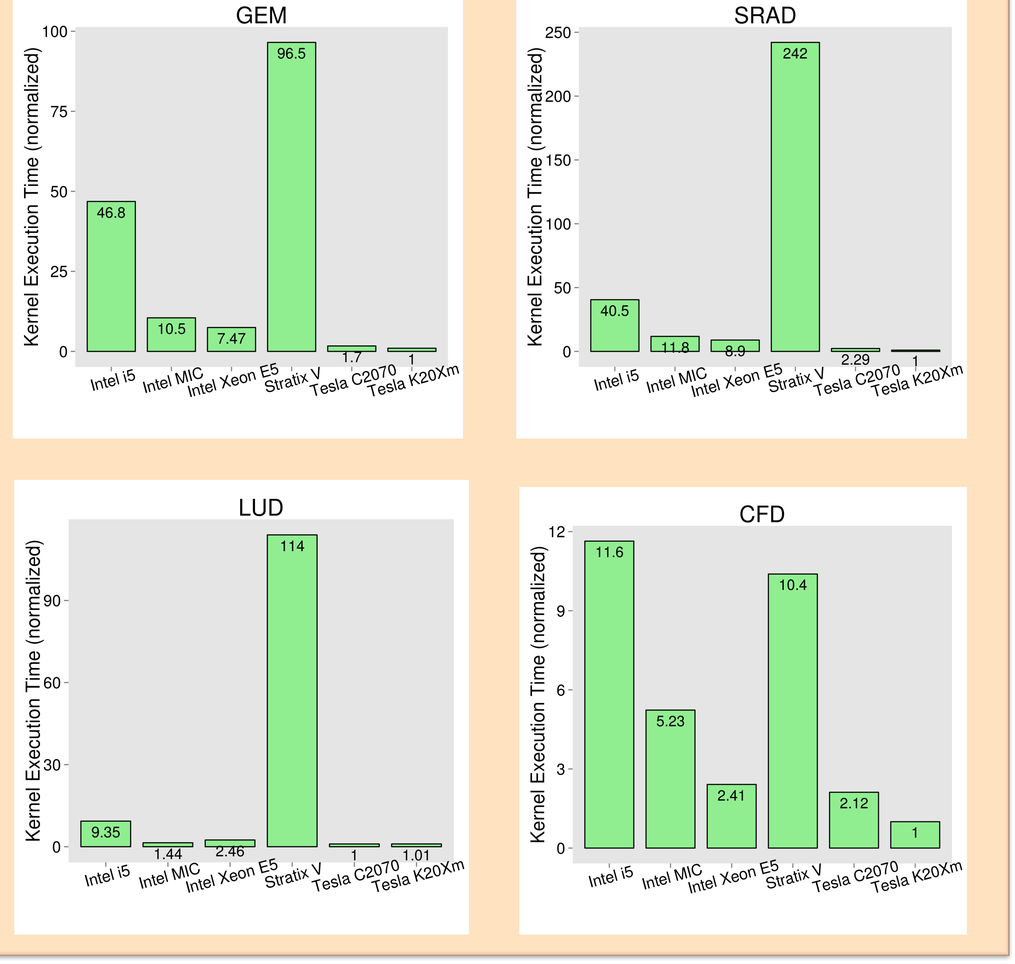
Fixed Test Architectures' Specifications

Model	Intel i5-2400	Intel Xeon E5-2700	Intel MIC 7100	Tesla C2070	Tesla K20X
Type	CPU	CPU	Co-proc.	GPU	GPU
Freq. (GHz)	3.1	2.7	1.238	1.15	0.732
Cores	4	12	61	14	14
SIMD (SP)	8	8	16	32	192
GFLOPS (SP)	198.4	518.4	2415.6	1030	3950
On-Chip mem.	7.125	33.375	32.406	3.375	3.032
B/W (GB/s)	21	59.7	352	148.42	250
Process (nm)	32	22	22	40	28
TDP (W)	95	130	270	238	235

Reconfigurable (FPGA) Test Architecture Specifications

Board: BittWare S5-PCIe-HQ-D8 (Altera Stratix V FPGA)
OpenCL Version: Altera OpenCL SDK v14.0

Architecture-Agnostic Kernel Performance of OpenDwarfs across Fixed and Reconfigurable Architectures



Altera OpenCL Optimizations

Algorithmic refactoring *(Refact.)*

Compute unit replication *(CU)*

Use of *restrict* and *constant* clauses *(Restrict, Constant)*

Loop unrolling *(Unroll)*

Use of kernel vectorization *(SIMD)*

Compiler resource-driven optimizations

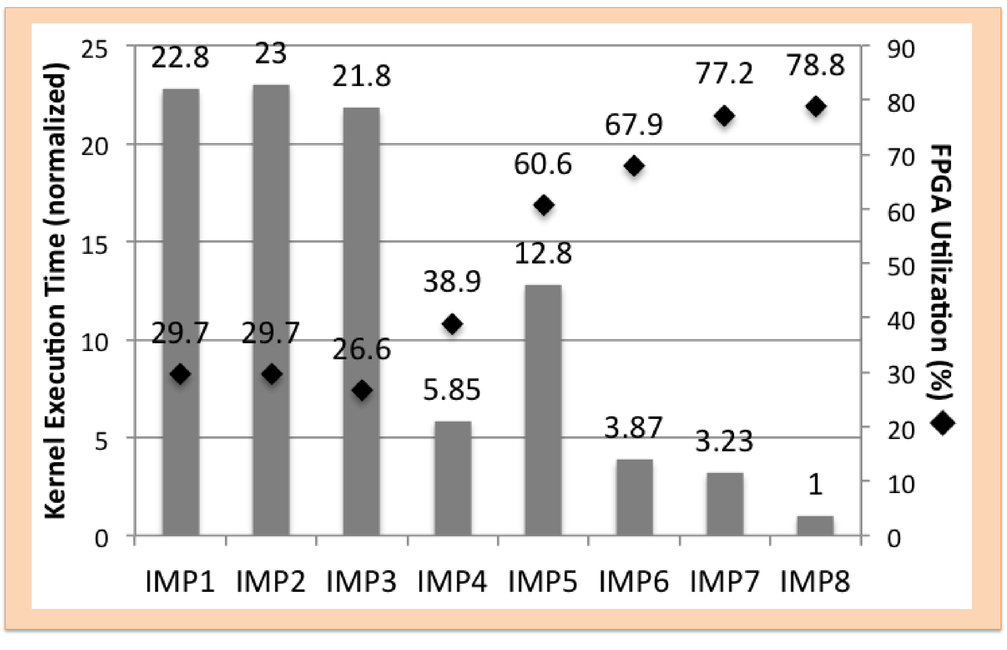
Keywords in parentheses correspond to optimizations shown in table in the Results section

FPGA Results: Electrostatic Potential Calculation – GEM: An N-body Dwarf

Implem.	Refact.	Restrict	Constant	SIMD	CU	Unroll
IMP1				1	1	1
IMP2			✓	1	1	1
IMP3	✓		✓	1	1	1
IMP4		✓	✓	1	1	1
IMP5			✓	1	1	4
IMP6			✓	8	1	1
IMP7	✓		✓	16	1	1
IMP8		✓	✓	8	1	1

SIMD: Single-Instruction Multiple Data CU: Compute Unit

GEM benchmark versions
(IMP1: Baseline implementation
IMP2-IMP8: Optimized implementations)



Kernel execution time & FPGA utilization for baseline and optimized implementations

This work was supported in part by NSF I/UCRC IIP-1266245 via the NSF Center for High-Performance Reconfigurable Computing (CHREC)