MANCHESTER 1824
The University of Manchester

CHO: Towards a Benchmark Suite for OpenCL FPGA Accelerators

3rd Inter. Workshop on OpenCL 2015

Geoffrey Ndu, Javier Navaridas & Mikel Lujan

MANCHESTER 1824
The University of Manchester

Talk Outline

1. Overview of FPGAs
2. Compiling OpenCL to FPGAs
3. CHO Benchmark Suite
4. Case Study: Compiling unmodified code to FPGA

MANCHESTER 1824
The University of Manchester

Motivation and Objectives

CHO is a benchmark suite for OpenCL FPGA
`http://it302.github.io/cho`

- Developed in AXLE Project
  - Advanced Analytics for Extremely Large European Databases
  - We accelerate database algorithms on FPGA
- There is no open OpenCL FPGA benchmark
  - Software benchmarks are too large and complex
- Benchmarking is important
  - Allows compiler writers to evaluate ideas qualitatively
  - Enables users benchmark diverse frameworks

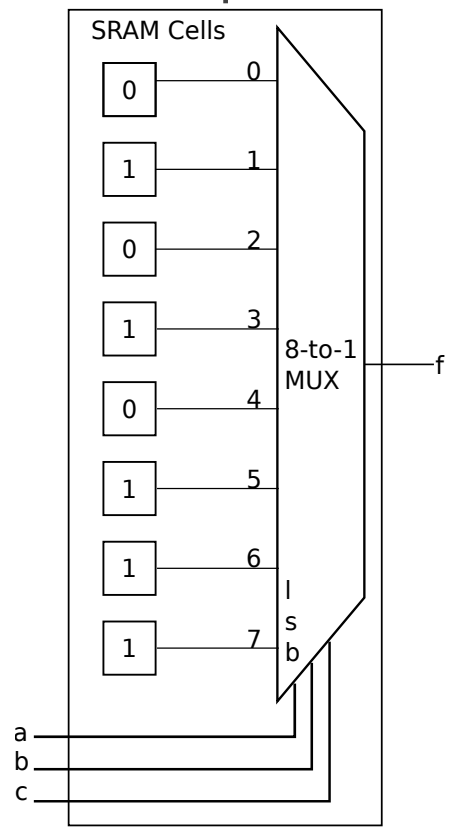
MANCHESTER 1824
The University of Manchester

What are FPGAs

Truth table

a	b	c	ab + c
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

LookUp Table



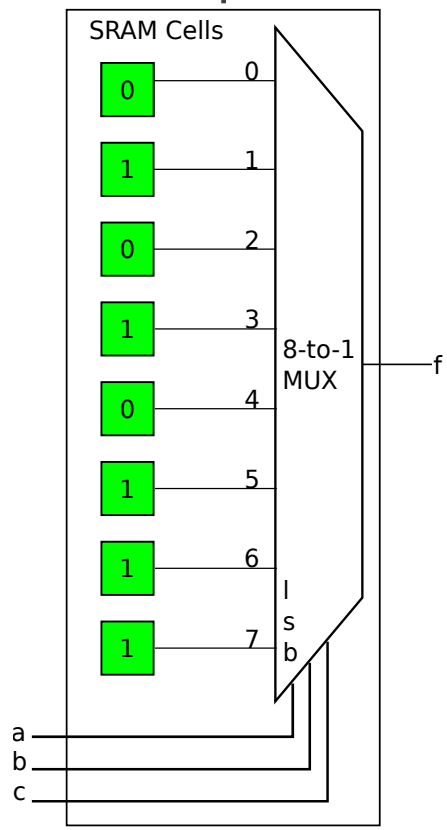
MANCHESTER 1824
The University of Manchester

What are FPGAs

Truth table

a	b	c	ab + c
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

LookUp Table



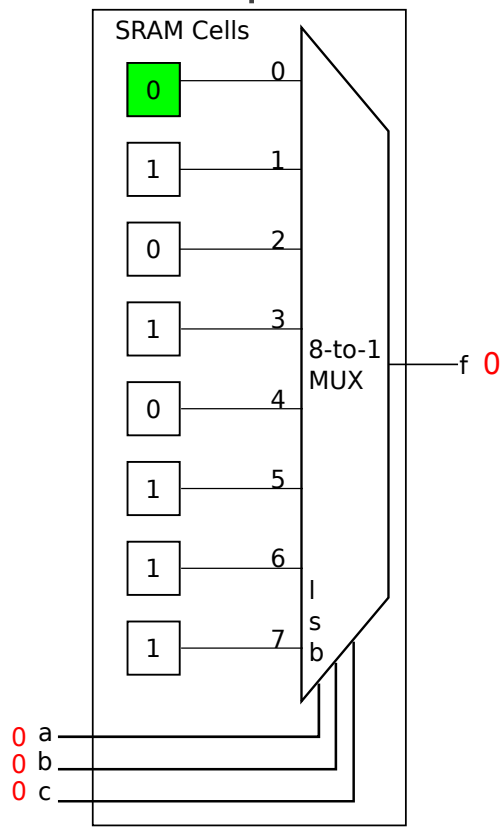
MANCHESTER 1824
The University of Manchester

What are FPGAs

Truth table

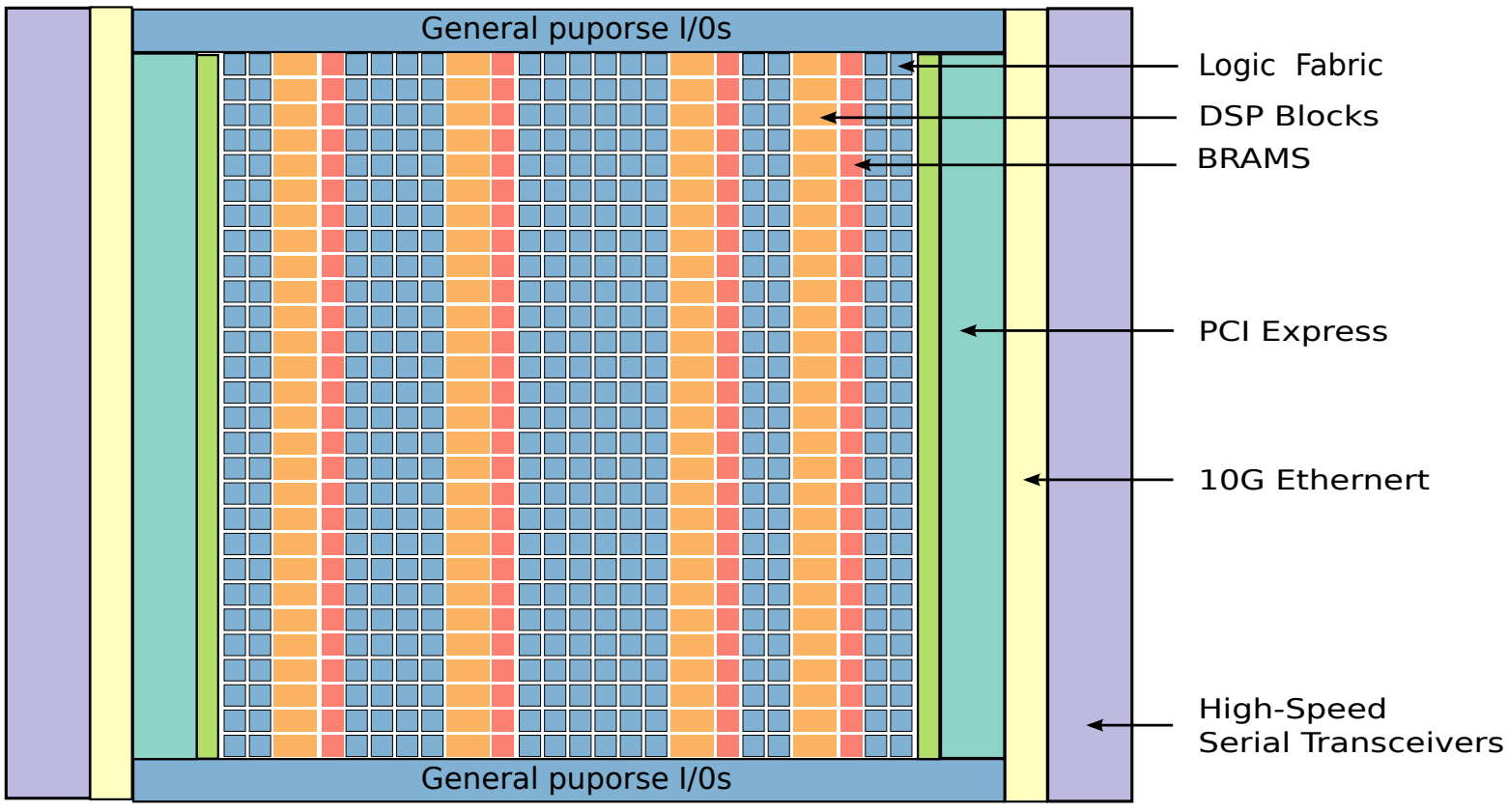
a	b	c	ab + c
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

LookUp Table



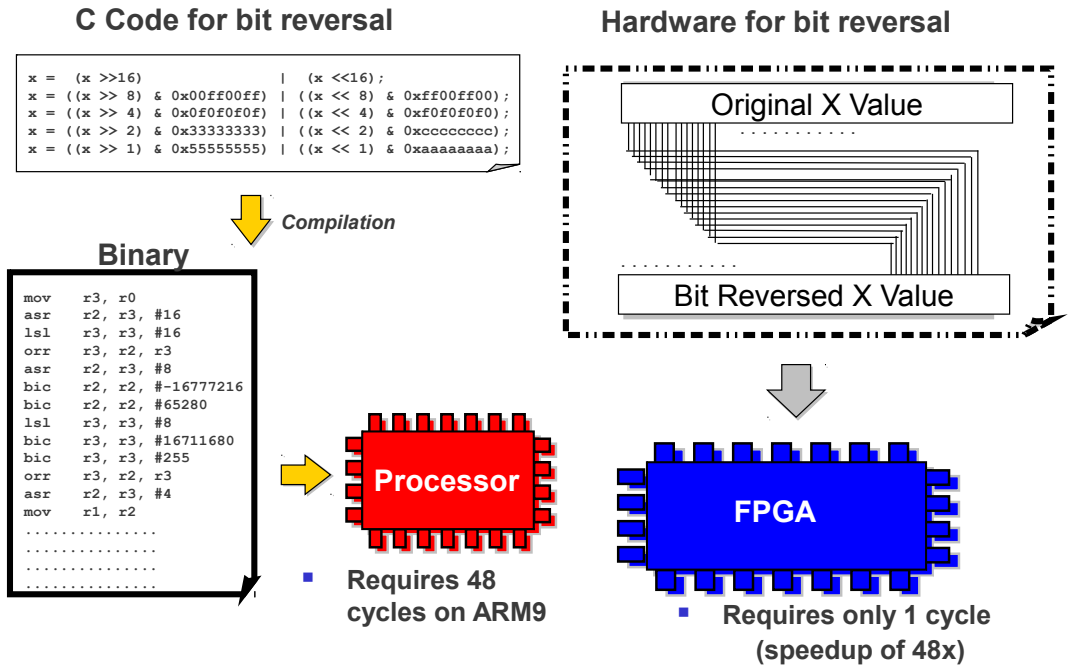
MANCHESTER 1824
The University of Manchester

A Modern FPGA



MANCHESTER 1824
The University of Manchester

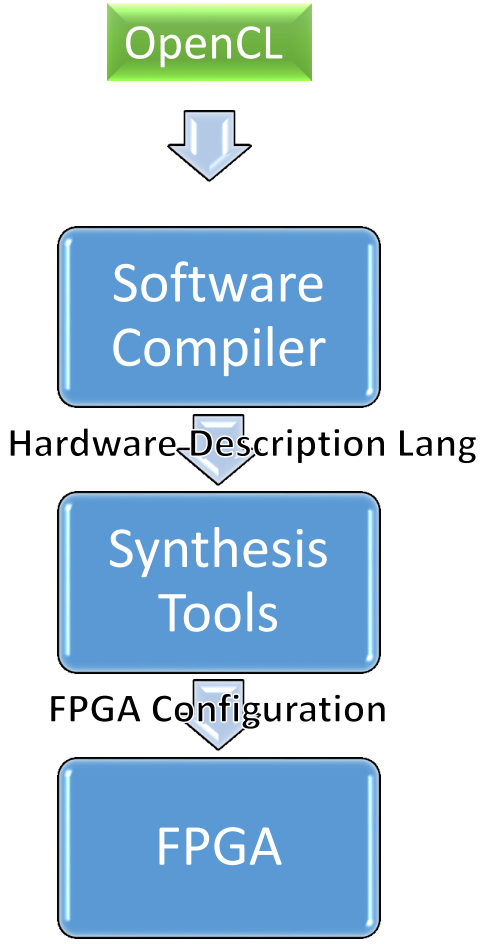
How FPGAs outperform CPUs and GPUs



(a) ARM9 bit reversal (b) FPGA bit reversal

MANCHESTER 1824
The University of Manchester

OpenCL on FPGAs



MANCHESTER 1824
The University of Manchester

CHO Benchmark Suite

- Work-in-progress
- Currently is a port of CHStone
- CHStone is the de-facto C HLS benchmark
  - HLS is hardware programming using algorithmic description
  - Easier than Verilog and VHDL type languages
  - Often trade-off usability/productivity for quality
- 12 ‘diverse’ and non-trivial applications
- Written in OpenCL task parallel model

MANCHESTER 1824
The University of Manchester

CHO's 12

Domain	Application	Description
Arithmetic	dfadd	IEC/IEEE double-precession floating-point addition
	dfdiv	IEC/IEEE double-precession floating-point division
	dfmul	IEC/IEEE double-precession floating-point multiplication
	dfsin	Double-precession floating-point sine function
Media	adpcm	Adaptive differential pulse code encoder & decoder
	gsm	GSM residual pulse excitation/long term prediction coding
	jpeg	JPEG image decoder
	motion	Motion vector decoding for MPEG-2
Cryptography	aes	Implementation of Advanced Encryption Standard
	blowfish	Blowfish Encryption Algorithm
	sha	Secure Hash Algorithm
Miscellaneous	MIPS	Simplified MIPS processors

MANCHESTER 1824
The University of Manchester

Source Level Characterization CHO

- Characterize to make sure applications non-trivial
- Built clang AST tool for analysing sources
  - Clang is front end for the C-type languages
  - Uses LLVM as back-end
- Tool walks AST and classifies nodes
- Useful for finding expensive operators

MANCHESTER 1824
The University of Manchester

Source-Level Characteristics

Kernel	Type	LoC	Functions	Variables		Statements					Operators						
				Scalar	Aggregate	*for*	*if*	*goto/break*	*switch*	*while*	Divide	Multiply	Add/Sub	Compare	Shift	Assign	Logic
adpcm	32-bit *int*	463	21	103	13	10	17	1	0	0	4	36	112	27	29	171	7
aes	32-bit *int*	667	10	42	25	16	24	31	6	0	14	95	98	41	184	389	209
blowfish	8-bit *char*	495	4	38	16	6	9	0	0	4	0	9	178	16	123	267	267
dfadd	64-bit *int*	352	81	2	18	1	43	6	0	0	0	0	29	46	21	104	34
dfdiv	64-bit *int*	280	19	96	1	1	31	0	0	2	2	4	33	45	30	100	31
dfmul	64-bit *int*	350	16	74	1	1	28	0	0	0	0	4	25	38	24	90	31
dfsin	64-bit *int*	611	31	178	1	1	74	6	0	3	2	7	54	91	45	188	65
gsm	32-bit *int*	310	12	45	7	17	16	0	0	1	0	47	168	57	23	211	11
jpeg	32-bit *int*	1061	31	204	30	34	58	11	1	11	3	52	135	92	44	342	21
mips	32-bit *int*	215	1	17	3	2	3	35	3	4	0	2	12	10	22	57	23
motion	32-bit *int*	436	13	59	15	7	21	0	0	4	2	6	40	28	16	84	7
sha	64-bit *char*	178	8	42	15	10	2	0	0	4	2	1	42	16	23	90	32

MANCHESTER 1824
The University of Manchester

IR-Level Characterization

- Compiler often translate software IR to HDL (verilog)
- IR-level provide missing info at source level
  - E.g. Num of loops left after optimization
- LLVM IR seems to be popular choice nowadays
  - LLVM is an open source modular compiler kit
  - Altera SDK is LLVM-based
- Built Clang/LLVM tool for IR-Level characterization

MANCHESTER 1824
The University of Manchester

IR-Level Characteristics

Kernel	Number Basic Blocks	Number Instructions	Number Loops
adpcm	95	2240	19
aes	186	5685	23
blowfish	58	5441	10
dfadd	199	1421	1
dfdiv	115	1212	3
dfmul	90	846	1
dfsin	469	4216	2
gsm	149	1493	18
jpeg	661	10261	109
mips	46	472	3
motion	1743	15121	326
sha	79	1364	30

MANCHESTER 1824
The University of Manchester

Case Study with CHO

- How easy to compile non-trivial algorithms on FPGAs
  - Especially when I/O interfacing is involved
- Could unmodified software OpenCL run on FPGA

MANCHESTER 1824
The University of Manchester

Synthesizing CHO

- Altera OpenCL SDK 14.1
- Nallatech P385-A7 FPGA accelerator card
  - Stratix V FPGA
  - 8GB DDR3 RAM
  - 8-lane Gen 3 PCIe
  - 1/2 length and height card
- Unmodified software applications

MANCHESTER 1824
The University of Manchester

Summary of Synthesis

Kernel	Synthesizable	Notes
adpcm	✓	
aes	✓	
blowfish	✓	
dfadd	✓	
dfdiv	✓	
dfmul	✓	
dfsin	✓	
gsm	✓	
jpeg	✗	Front-end error
mips	✓	
motion	✗	Front-end error
sha	✓	

MANCHESTER 1824
The University of Manchester

Difference of Major Versions

Altera SDK 14.1

Kernel	Synthesizable	Notes
adpcm	✓	
aes	✓	
blowfish	✓	
dfadd	✓	
dfdiv	✓	
dfmul	✓	
dfsin	✓	
gsm	✓	
jpeg	✗	Front-end error
mips	✓	
motion	✗	Front-end error
sha	✓	

Altera SDK 13.1

Kernel	Synthesizable	Notes
adpcm	✗	Frontend error
aes	✗	Frontend crash
blowfish	✓	
dfadd	✓	
dfdiv	✗	Frontend crash
dfmul	✓	
dfsin	✗	Frontend crash
gsm	✓	
jpeg	✗	Frontend error
mips	✗	Backend error
motion	✗	Frontend crash
sha	✗	Backend error

MANCHESTER 1824
The University of Manchester

CPU vs FPGA

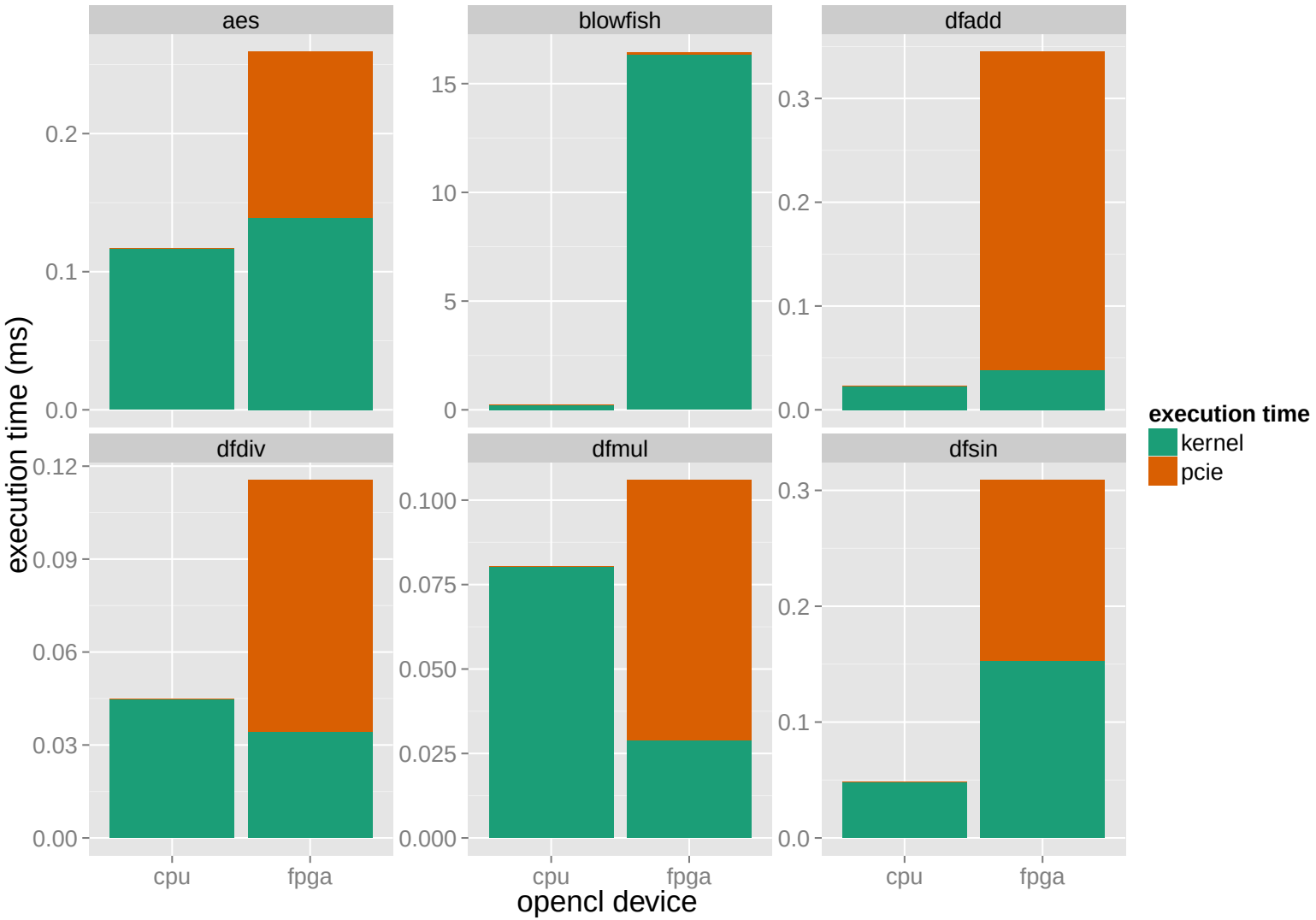
- Compare performance FPGA vs CPU
- Kernels not optimized for FPGA
- Some applications failed on the FPGA
  - 3 produced incorrect results
  - 2 never finished execution

CPU	Intel Xeon CPU E31245 @ 3.30GHz
Memory Size	16 GB DDR3
OpenCL SDK	Intel Kernel Builder for OpenCL 1.4.0.117

CPU Platform Features

MANCHESTER 1824
The University of Manchester

Performance



MANCHESTER 1824
The University of Manchester

Concluding

- Straightforward to compile software OpenCL to FPGA
- Kernels most likely need to be tuned for FPGA
- Compiler is getting better
- Working on CHO 2.0
  - Data parallel OpenCL kernels
  - More modern application e.g. database algorithms
  - Tuned for FPGAs

MANCHESTER 1824
The University of Manchester

AXLE Project

www.axleproject.eu

Thank You for Listening

The research leading to these results has received funding from the European Union's Seventh Framework Programme (FP7/2007-2013) under grant agreement no 318633. We wish to thank Altera University Program for their software donation.