

Foreword

The International Workshop for OpenCL (IWOCL, which is pronounced “eye-wok-ul”) was conceived in a meeting between Simon McIntosh-Smith and Ben Bergen at the Los Alamos National Laboratory on May 8th 2012. McIntosh-Smith and Bergen lamented that there were no organized workshops or meetings for the rapidly growing OpenCL community. After testing this idea with colleagues over the next few months, they decided to create the kind of OpenCL conference they wanted to go to themselves, and thus IWOCL was born.

Based on the same model as the OpenMP conference IWOMP, the first IWOCL was organized in just two months and held over May 13-14, 2013 at Georgia Tech in Atlanta, with Prof. David Bader as the local organizer. Despite the short timescale, over 40 OpenCL users from industry and academia attended, with delegates from the US, Europe and Asia. Encouraged by this success, and with longer to organize the next conference, IWOCL 2014 was held from May 11-12 at the @Bristol science museum in Bristol, UK with Simon McIntosh-Smith as the local organizer. IWOCL 2014 saw over 110 delegates from 14 different countries, with 30 different companies and 18 different academic institutions represented. IWOCL 2014 saw the addition of a series of tutorials which included how to profile and tune OpenCL performance using Intel’s vtune profiler, one of the first public tutorials for using OpenCL with Altera FPGAs, how to use ArrayFire’s OpenCL math libraries, and one of the first public tutorials for optimizing OpenCL on ARM’s Mali GPUs. IWOCL 2015 saw the conference move to Stanford University in California, USA and the addition of an advanced OpenCL training day. Around 100 delegates from 14 countries, 7 academic institutions and 30 different companies listened to a diverse collection of full papers, technical presentations, workshops and posters. 2015 saw 27 submissions from which 19 were accepted.

All paper submissions went through peer review by a dedicated program committee of 26 reviewers, without whom IWOCL could not happen: the chairs express their sincere thanks to the program committee for all their help in pulling the IWOCL conferences.

Like the 2013-14 proceedings, the IWOCL 2015 proceedings are appearing in the ACM International Conference Proceeding Series. Topics covered include optimized math libraries, achieving performance portability, case studies of porting commercial codes to OpenCL, experiences in porting OpenCL codes to FPGAs, energy efficiency comparisons between embedded CPUs and GPUs, and tools to aid in the debugging and testing of OpenCL codes.

At the time of writing IWOCL 2016 has already been organized for April 19-21 TU Wien in Vienna, Austria. We hope that our colleagues in the diverse OpenCL community continue to find the papers, technical talks, tutorials and posters at IWOCL interesting and useful.

Simon McIntosh-Smith

IWOCL co-chair

University of Bristol, UK

Ben Bergen

IWOCL co-chair

Los Alamos National Laboratory, USA

IWOCL 2015 – Order of Paper for the Proceedings

Paper ID No.	Title of Accepted Paper
	Front Matter Copyright Page Organization Pages
1	IWOCL 2015 Opening and Closing Address
2	A Look at the OpenCL 2.0 Execution Model (technical presentation: extended abstract, slides)
3	Achieving Performance with OpenCL 2.0 on Intel Processor Graphics (technical presentation: extended abstract, slides)
4	Asynchronous OpenCL/MPI numerical simulations of conservation laws (technical presentation: extended abstract, slides)
5	Exploring the Features of OpenCL 2.0 (technical presentation: extended abstract, slides)
6	Mapping C++ AMP to OpenCL / HAS (technical presentation: extended abstract, slides)
7	Performance Optimization for a SHA-1 Cryptographic Workload Expressed in OpenCL for FPGA Execution (technical presentation: extended abstract, slides)
8	The Great Beyond: Higher Productivity, Parallel Processors and the Extraordinary Search for a Theory of Expression (technical presentation: extended abstract, slides)
9	Update on the SYCL for OpenCL Open Standard to Enable C++ Meta Programming on Top of OpenCL (technical presentation: extended abstract, slides)
10	CHO: Towards a Benchmark Suite for OpenCL FPGA Accelerators (full paper)
11	Kernel Composition in SYCL (full paper)
12	Oclgrind: An Extensible OpenCL Device Simulator (full paper)
13	GPU Compute on Snapdragon (invited talk)
14	Heterogeneous Computing: the rise of open programming frameworks (invited talk)
15	Leveraging OpenCL to Create Differentiation (invited talk)
16	OpenCL.Next (panel)
17	A Compute Model for Augmented Reality with Integrated-GPU Acceleration (poster)
18	Accelerating SGEMM with Subgroups (poster)
19	High Dynamic Range Imaging by Heterogeneous Computing in Mobile Devices (poster)
20	Nano Simbox: An OpenCL-accelerated Framework for Interactive Molecular Dynamics (poster)
21	OpenCL Accelerated Deep Learning for Visual Understanding (poster)
22	A Framework for Visualization of OpenCL Applications Execution (tutorial: extended abstract, slides)
23	Developing Optimized Libraries for Scalable OpenCL Acceleration on FPGAs (tutorial: slides)
24	Khronos SYCL for OpenCL (tutorial: extended abstract, slides)
25	Propel With OpenCL: A Deep Dive Workshop to Create, Debug, Analyze and Optimize OpenCL Applications using Intel Tools (tutorial: extended abstract, slides)

International Workshop on OpenCL

2015 IWOCL Organization

General Chairs: Simon McIntosh-Smith (*University of Bristol, UK*)
Ben Bergen (*Los Alamos National Laboratory, USA*)

Local Arrangements Chair: 2015: Dave Glowacki (*Stanford University, USA*)
2014: Simon McIntosh-Smith (*University of Bristol, UK*)
2013: David Bader (*Georgia Tech., USA*)

Local Arrangements Committee: Tim Lewis (*Croftedge Marketing Limited, UK*)

Steering Committee Chair: Simon McIntosh-Smith (*University of Bristol, UK*)

Steering Committee: Simon McIntosh-Smith (*University of Bristol, UK*)
Ben Bergen (*Los Alamos National Laboratory, USA*)
Karl Rupp (*TU Wien, Austria*)

Proceedings Editor: Tom Deakin (*University of Bristol, UK*)

Program Committee: Francois Bodin (Irisa, France)
David Beckingsale (Warwick University, UK)
Ben Bergen (Los Alamos National Laboratory, USA)
Mike Boulton (University of Bristol, UK)
Alex Bourd (Qualcomm, USA)
Andrew Brownsword (Intel, USA)
Daniel Connors (Colorado State, USA)
Marco Cornero (ARM, UK)
Dan Curran (University of Bristol, UK)
Tom Deakin (University of Bristol, UK)
Dmitry Denisenko (Altera, USA)
Alastair Donaldson (Imperial College, UK)
Benedict Gaster (University of the West of England, UK)
Sebastian Hack (University of Saarland, Germany)
Vincent Hindriksen (Streamcomputing, The Netherlands)
David Kaeli (North Eastern University, USA)
Paul Keir (Codeplay, USA)
Anton Lokhmotov (dividiti, UK)
Pat McCormick (Los Alamos National Laboratory, USA)
Laurent Morichetti (AMD)
Mike O'Connor (University of Bristol, UK)
James Price (University of Bristol, UK)
Andrew Richards (Codeplay, UK)
Karl Rupp (Vienna University of Technology, Austria)
John Stone (UIUC, USA)
Doug Watt (Imagination Technologies, UK)

Platinum sponsors:



Gold sponsors:



Sponsors: **Silver sponsors:**



Sponsors:



Supporters:



2015: Dave Glowacki (*Stanford University, USA*)

Hosted by: 2014: Simon McIntosh-Smith (*University of Bristol, UK*)

2013: David Bader (*Georgia Tech., USA*)

Mapping C++ AMP to OpenCL / HSA

Jack Chung
MulticoreWare Inc
Taiwan

Curtis Davis
MulticoreWare Inc
USA

Jayram Ramachandran
MulticoreWare Inc
India

ABSTRACT

High-level programming languages and domain-specific languages can often benefit from the increased power efficiency of heterogeneous computing. OpenCL can serve as a compiler target for portable code generation and runtime management. By using OpenCL as the target platform, compiler writers can focus on more important, higher-level problems in language implementation. Such improved productivity can enable a proliferation of high-level programming languages for heterogeneous computing systems.

The C++ programming language provides several high-level, developer-friendly features that are missing from OpenCL. These high-level features support software engineering practices and improve developer productivity. With the advent of OpenCL 2.0 and Heterogeneous System Architecture (HSA), more C++ language constructs can be efficiently mapped and executed on multi-core architectures. It is the compiler writer's job to translate these features into the OpenCL constructs without incurring an excessive level of overhead.

C++ AMP is a parallel programming extension to C++, and MulticoreWare have contributed to Clamp, an open source implementation. The compiler is based on Clang / LLVM, and could target multiple platforms such as OpenCL / SPIR / HSA. We present some important implementation techniques in this compiler, and we would also present how shared virtual memory, platform atomics could allow more generic C++ codes to leverage multi-core architectures.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

IWOCL '15 May 12-13, 2015, Palo Alto, CA, USA

© 2015 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-3484-6/15/05.

DOI: <http://dx.doi.org/10.1145/2791321.2791327>

The Great Beyond: Higher Productivity, Parallel Processors and the Extraordinary Search for a Theory of Expression *

Alan Ward
Texas Instruments, USA

ABSTRACT

Embedded system on a chip (SOC) vendors of today are perpetually challenged with the following goals; provide more compute capability and reduce cost and power. Unfortunately, these goals compete rather than cooperate. In order to address this, SOC designers have turned to component specialization, where multiple specialized computing engines exist on a chip, each of which can handle one or a small subset of tasks very well. Since these engines are tuned for their task, they can be smaller, thus reducing cost, and run at a lower frequency, thus reducing power, than a similarly capable generic computing engine. Of course, this diversity does drive complexity in the system with a combination of RISC/DSP/GPU/CUSTOM engines and their corresponding OS's, tool chains, communication and IPC protocols.

Concurrently, application software teams have their own perpetual challenges; lower development cost, shorten time to market and improve reliability. These goals are obviously more difficult when the software team is presented with the complexity resulting from the diverse SOC designs on which their software will run. The software task grows even more difficult with the additional requirement that it must run well over a set of SOC instantiations.

In this presentation, I will explore the use of OpenCL, beyond its more typical use cases on GPU and CPU systems, as a unifying agent between the demands on SOC software teams and the demands on SOC design teams. I will discuss OpenCL use on a multicore CPU+DSP SOC, how we have extended OpenCL to increase its acceptance and adoption within TI's software community. I will also discuss additional use cases, features and requirements of embedded software that OpenCL does not cover today, but perhaps could!

*Title Inspiration, thanks to Paul Halpern, author of "The Great Beyond: Higher Dimensions, Parallel Universes and the Extraordinary Search for a Theory of Everything"

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

IWOCL '15 May 12-13, 2015, Palo Alto, CA, USA

© 2015 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-3484-6/15/05.

DOI: <http://dx.doi.org/10.1145/2791321.2791329>

Update on the SYCL for OpenCL Open Standard to Enable C++ Meta Programming on Top of OpenCL

Andrew Richards^{*}
Codeplay, UK

ABSTRACT

SYCL is a royalty-free, open standard, higher-level C++ programming model for OpenCL. C++ developers can produce easy-to-use template libraries for OpenCL devices, as well as easily porting C++ applications to use OpenCL. By providing ease-of-use, high performance and modern C++ techniques, SYCL enables a wide range of developers to accelerate their applications and libraries.

Previously, two provisional specifications of SYCL have been released. We hope, subject to approval, to be able to present exciting news about SYCL for developers at IWOCL. This presentation will take developers through: what SYCL is, our latest news, as well as all the new possibilities SYCL enables for developers.

We are particularly keen to talk about how SYCL can work well with other C++ standards and libraries, such as C++17, to bring the high performance and widespread device support of OpenCL to a whole new community of developers.

^{*}Andrew Richards is Chair of the SYCL working group, as well as CEO of Codeplay.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

IWOCL '15 May 12-13, 2015, Palo Alto, CA, USA

© 2015 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-3484-6/15/05.

DOI: <http://dx.doi.org/10.1145/2791321.2791330>