OpenCL Introduction

Neil Trevett
Vice President NVIDIA, President Khronos
OpenCL Working Group Chair
The Inspiration for OpenCL

CPUs
- Multiple cores driving performance increases
- Multi-processor programming - e.g. OpenMP

GPUs
- Increasingly general purpose data-parallel computing
- Graphics APIs and Shading Languages

Emerging Intersection

Heterogeneous Computing

OpenCL
The BIG Idea behind OpenCL

- OpenCL execution model ...
  - Define N-dimensional computation domain
  - Execute a kernel at each point in computation domain

Traditional Loop

```c
void vectorMult(
    const float* a,
    const float* b,
    float* c,
    const unsigned int count)
{
    for(int i=0; i<count; i++)
        c[i] = a[i] * b[i];
}
```

Data Parallel OpenCL

```c
kernel void vectorMult(
    global const float* a,
    global const float* b,
    global float* c)
{
    int id = get_global_id(0);
    c[id] = a[id] * b[id];
}
```
OpenCL - Portable Heterogeneous Computing

- Royalty-free native, cross-platform, cross-vendor standard
  - Targeting supercomputers -> embedded systems -> mobile devices

- Enables programming of diverse compute resources
  - CPU, GPU, DSP, FPGA - and hardware blocks

- One code tree can be executed on CPUs, GPUs, DSPs and hardware
  - Dynamically interrogate system load and balance across available processors

- Powerful, low-level flexibility
  - Foundational access to compute resources for higher-level engines, frameworks and languages
OpenCL Architecture

- **C Platform Layer API**
  - Query, select and initialize compute devices

- **Kernel Language Specification**
  - Subset of ISO C99 with language extensions
  - Well-defined numerical accuracy - IEEE 754 rounding with specified max error
  - Rich set of built-in functions: cross, dot, sin, cos, pow, log ...

- **C Runtime API**
  - Runtime or build-time compilation of kernels
  - Execute compute kernels across multiple devices

- **Embedded profile**
  - No need for a separate “ES” spec
  - Reduces precision requirements
OpenCL Platform Model

- A host is connected to one or more OpenCL devices
- OpenCL device is collection of one or more compute units
- A compute unit is composed of one or more processing elements
- Processing elements execute code as SIMD or SPMD
OpenCL Execution Model

- **Kernel**
  - Basic unit of executable code (~ C function)
  - Data-parallel or task-parallel

- **Program**
  - Collection of kernels and functions
    (~ dynamic library with run-time linking)

- **Command Queue**
  - Applications queue kernels & data transfers
    - Performed in-order or out-of-order

- **Work-item**
  - An execution of a kernel by a processing element (~ thread)

- **Work-group**
  - A collection of related work-items that execute on a single compute unit (~ core)

Work-group Example

# Work-items = # pixels
# Work-groups = # tiles
Work-group size = tile width * tile height
OpenCL Memory Model

• Hierarchy of memory types
  - Private memory -
    - Per work-item
  - Local memory (green)
    - Per work-group
    - Available to work-items in a given work-group
  - Global/Constant memory
    - Not synchronized
  - Host memory
    - On the CPU

• Memory management is explicit:
  - Application must move data from host → global → local and back
Executing OpenCL Programs

1. Query host for OpenCL devices
2. Create a context to associate OpenCL devices
3. Create programs for execution on one or more associated devices
4. Select kernels to execute from the programs
5. Create memory objects accessible from the host and/or the device
6. Copy memory data to the device as needed
7. Provide kernels to command queue for execution
8. Copy results from the device to the host
OpenCL Built-in Kernels

- Used to control non-OpenCL C-capable resources on an SOC - ‘Custom Devices’
  - E.g. Video encode/decode, Camera ISP ...
- Represent functions of Custom Devices as an OpenCL kernel
  - Can enqueue Built-in Kernels to Custom Devices alongside standard OpenCL kernels
- OpenCL run-time a powerful coordinating framework for ALL SOC resources
  - Programmable and custom devices controlled by one run-time
OpenCL Related Specification Roadmap

OpenCL HLM (High Level Model)
High-level programming model, unifying host and device execution environments through language syntax for increased usability and broader optimization opportunities

OpenCL 2.0
Significant enhancements to memory and execution models to expose emerging hardware capabilities and provide increased flexibility, functionality and performance to developers

OpenCL SPIR 1.2 Provisional released at SIGGRAPH 2013
SPIR (Standard Parallel Intermediate Representation)
LLVM-based, low-level Intermediate Representation for IP Protection and as target back-end for alternative high-level languages

OpenCL 2.0 Finalized here at SIGGRAPH Asia 2013!
OpenCL Milestones

• 24 month cadence for major OpenCL 2.0 update
  - Slightly longer than 18 month cadence between versions of OpenCL 1.X

• Significant feedback from the developer community on Provisional Specification
  - Many suggestions were incorporated into the final 2.0 specification
  - Other feedback will be considered for future specification versions

- OpenCL 1.0 released. Conformance tests released Dec08
- OpenCL 1.1 Specification and conformance tests released Jun10
- OpenCL 1.2 Specification and conformance tests released Nov11
- OpenCL 2.0 Provisional Specification released for public review Jul13
- OpenCL 2.0 Specification finalized and conformance tests released Nov13

© Copyright Khronos Group 2013 - Page 12
Broad OpenCL Implementer Adoption

- Multiple conformant implementations shipping on desktop and mobile
  - For CPUs and GPUs on multiple OS
- Android ICD extension released in latest extension specification
  - OpenCL implementations can be discovered and loaded as a shared object
- Multiple implementations shipping in Android NDK
  - ARM, Imagination, Vivante, Qualcomm, Samsung ...
OpenCL as Parallel Compute Foundation

- 100+ tool chains and languages leveraging OpenCL
  - Heterogeneous solutions emerging for the most popular programming languages

OpenCL provides vendor optimized, cross-platform, cross-vendor access to heterogeneous compute resources
Widespread Developers Leveraging OpenCL

- Broad uptake of OpenCL in commercial applications
  - For desktop and increasingly mobile apps
- “OpenCL” on Sourceforge, Github, Google Code, BitBucket finds over 2,000 projects
  - x264
  - Handbrake
  - FFMPEG
  - JPEG
  - VLC
  - OpenCV
  - GIMP
  - ImageMagick
  - IrfanView
  - Hadoop, Memcched
  - Aparapi - A parallel API (for Java)
  - Bolt - a Unified Heterogeneous Library
  - Sumatra - next generation of compute enabled Java
  - WinZip
  - Crypto++
  - Bullet physics library
  - Etc. Etc.
OpenCL Academic Traction

- OpenCL at over 100 Universities Worldwide
  - Teaching multi-faceted programming courses
    - Research with top-tier Universities globally
- Complete University Kits available
  - Presentation w/instructor & speaker notes
  - Example code, & sample application
- Growing textbook ecosystem
  - US, Japan, Europe, China and India
- Number of papers referencing OpenCL on Google Scholar is growing rapidly
  - Over 2000 papers in 2012
- Commercial OpenCL training courses
  - [http://www.accelereyes.com/services/training](http://www.accelereyes.com/services/training)
Major Benchmarks Leveraging OpenCL

- PCMark 8 uses OpenCL
  - Video Chat and Video Group Chat
  - Batch Video Edit

- BasemarkCL, CompuBench use OpenCL as leading indicators of platform performance

- Reviewed performance benchmarks use heterogeneous computing via OpenCL
  - AnandTech, Tom’s Hardware Guide

- End-user benchmarks transitioning to use heterogeneous computing
  - E.g. Ludashi (China) is using OpenCL
Give us YOUR Feedback!

- Full OpenCL 2.0 Documentation available
  - Final Specification
  - Header files
  - Reference Card
  - Online Reference pages

- OpenCL Registry contains all specifications
  - [www.khronos.org/registry/cl/](http://www.khronos.org/registry/cl/)

- Open Resources Area
  - Community submitted resources
  - [http://www.khronos.org/opencl/resources](http://www.khronos.org/opencl/resources)

- Public Forum and Bugzilla is open for comments
  - All feedback welcome!
OpenCL Presentations in This Session

- OpenCL 2.0 Overview
  - Allen Hux, Intel

- Accelerated Science - use of OpenCL in Land Down Under
  - Tomasz Bednarz, CSIRO
  - Sydney Khronos Chapter Leader