OpenCL Platform Model

- **One Host + one or more Compute Devices**
  - Each Compute Device is composed of one or more Compute Units
  - Each Compute Unit is further divided into one or more Processing Elements
OpenCL Memory Model

- **Private Memory**
  - Per work-item
- **Local Memory**
  - Shared within a workgroup
- **Global/Constant Memory**
  - Visible to all workgroups
- **Host Memory**
  - On the CPU

Memory management is Explicit
You must move data from host -> global -> local ... and back
OpenCL Execution Model

- **OpenCL application runs on a host which submits work to the compute devices**
  - **Context**: The environment within which work-items executes ... includes devices and their memories and command queues
  - **Program**: Collection of kernels and other functions (Analogous to a dynamic library)
  - **Kernel**: the code for a work item. Basically a C function
  - **Work item**: the basic unit of work on an OpenCL device

- **Applications queue kernel execution**
  - Executed in-order or out-of-order
An N-dimension domain of work-items

- Kernels executed across a global domain of **work-items**
- Work-items grouped into local **workgroups**
- Define the “best” N-dimensioned index space for your algorithm
  - Global Dimensions: 1024 x 1024 (whole problem space)
  - Local Dimensions: 128 x 128 (work group ... executes together)

Synchronization between work-items possible only within workgroups: barriers and memory fences

Cannot synchronize outside of a workgroup
Programming Kernels: OpenCL C

• Derived from ISO C99
  - But without some C99 features such as standard C99 headers, function pointers, recursion, variable length arrays, and bit fields

• Language Features Added
  - Work-items and workgroups
  - Vector types
  - Synchronization
  - Address space qualifiers

• Also includes a large set of built-in functions
  - Image manipulation
  - Work-item manipulation,
  - Math functions, etc.
Creating an OpenCL Program

- CPU
- GPU
- DSP

Context

- Programs
- Kernels
- Memory Objects
- Command Queue

Programs
- Kernel0
- Kernel1
- Kernel2

Memory Objects
- Images
- Buffers

Command Queue
- In order & out of order

Compile
Create data and arguments
Send for execution

© Copyright Khronos Group, 2012 - Page 6
Synchronization: Queues & Events

- Events can be used to synchronize kernel executions between queues
- Example: 2 queues with 2 devices
OpenCL 1.2 Announced in December

• Significant updates - Khronos being responsive to developer requests
  - Updated OpenCL 1.2 conformance tests available
  - Multiple implementations underway

• Backward compatible upgrade to OpenCL 1.1
  - OpenCL 1.2 will run any OpenCL 1.0 and OpenCL 1.1 programs
  - OpenCL 1.2 platform can contain 1.0, 1.1 and 1.2 devices
  - Maintains embedded profile for mobile and embedded devices
Partitioning Devices

- Devices can be partitioned into sub-devices
  - More control over how computation is assigned to compute units

- Sub-devices may be used just like a normal device
  - Create contexts, building programs, further partitioning and creating command-queues

- Three ways to partition a device
  - Split into equal-size groups
  - Provide list of group sizes
  - Group devices sharing a part of a cache hierarchy
Custom Devices and Built-in Kernels

- **Embedded platforms often contain specialized hardware and firmware**
  - That cannot support OpenCL C

- **Built-in kernels can represent these hardware and firmware capabilities**
  - Such as video encode/decode

- **Hardware can be integrated and controlled from the OpenCL framework**
  - Can enqueue built-in kernels to custom devices alongside OpenCL kernels

- **FPGAs are one example of device that can expose built-in kernels**
  - Latest FPGAs can support full OpenCL C as well

- **OpenCL becomes a powerful coordinating framework for diverse resources**
  - Programmable and non-programmable devices controlled by one run-time

Built-in kernels enable control of specialized processors and hardware from OpenCL run-time
Installable Client Driver

• Analogous to OpenGL ICDs in use for many years
  - Used to handle multiple OpenGL implementations installed on a system

• Optional extension
  - Platform vendor will choose whether to use ICD mechanisms

• Khronos OpenCL installable client driver loader
  - Exposes multiple separate vendor installable client drivers (Vendor ICDs)

• Application can access all vendor implementations
  - The ICD Loader acts as a de-multiplexor

ICD Loader ensures multiple implementations are installed cleanly

ICD Loader enables application to use any of the installed implementations
Other Major New Features in OpenCL 1.2

- **Separate compilation and linking of objects**
  - Provides the capabilities and flexibility of traditional compilers
  - Create a library of OpenCL programs that other programs can link to

- **Enhanced Image Support**
  - Added support for 1D images, 1D & 2D image arrays
  - OpenGL sharing extension now enables an OpenCL image to be created from an OpenGL 1D texture, 1D and 2D texture arrays

- **DX9 Media Surface Sharing**
  - Efficient sharing between OpenCL and DirectX 9 or DXVA media surfaces

- **DX11 surface sharing**
  - Efficient sharing between OpenCL and DirectX 11 surfaces

- And many other updates and additions..
OpenCL Desktop Implementations

OpenCL Books – Available Now!

- OpenCL Programming Guide - The “Red Book” of OpenCL

- OpenCL in Action

- Heterogeneous Computing with OpenCL

- The OpenCL Programming Book
Spec Translations

• Japanese OpenCL 1.1 spec translation available today
  - Valued partnership between Khronos and CUTT in Japan

• Working on OpenCL 1.2 specification translations
  - Japanese, Korean and Chinese
Khronos OpenCL Resources

- **OpenCL is 100% free for developers**
  - Download drivers from your silicon vendor

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

- **OpenCL 1.2 Reference Card**
  - PDF version

- **Online Man pages**
  - [http://www.khronos.org/registry/cl/sdk/1.2/docs/man/xhtml/](http://www.khronos.org/registry/cl/sdk/1.2/docs/man/xhtml/)

- **OpenCL Developer Forums**
  - Give us your feedback!
  - [www.khronos.org/message_boards/](http://www.khronos.org/message_boards/)