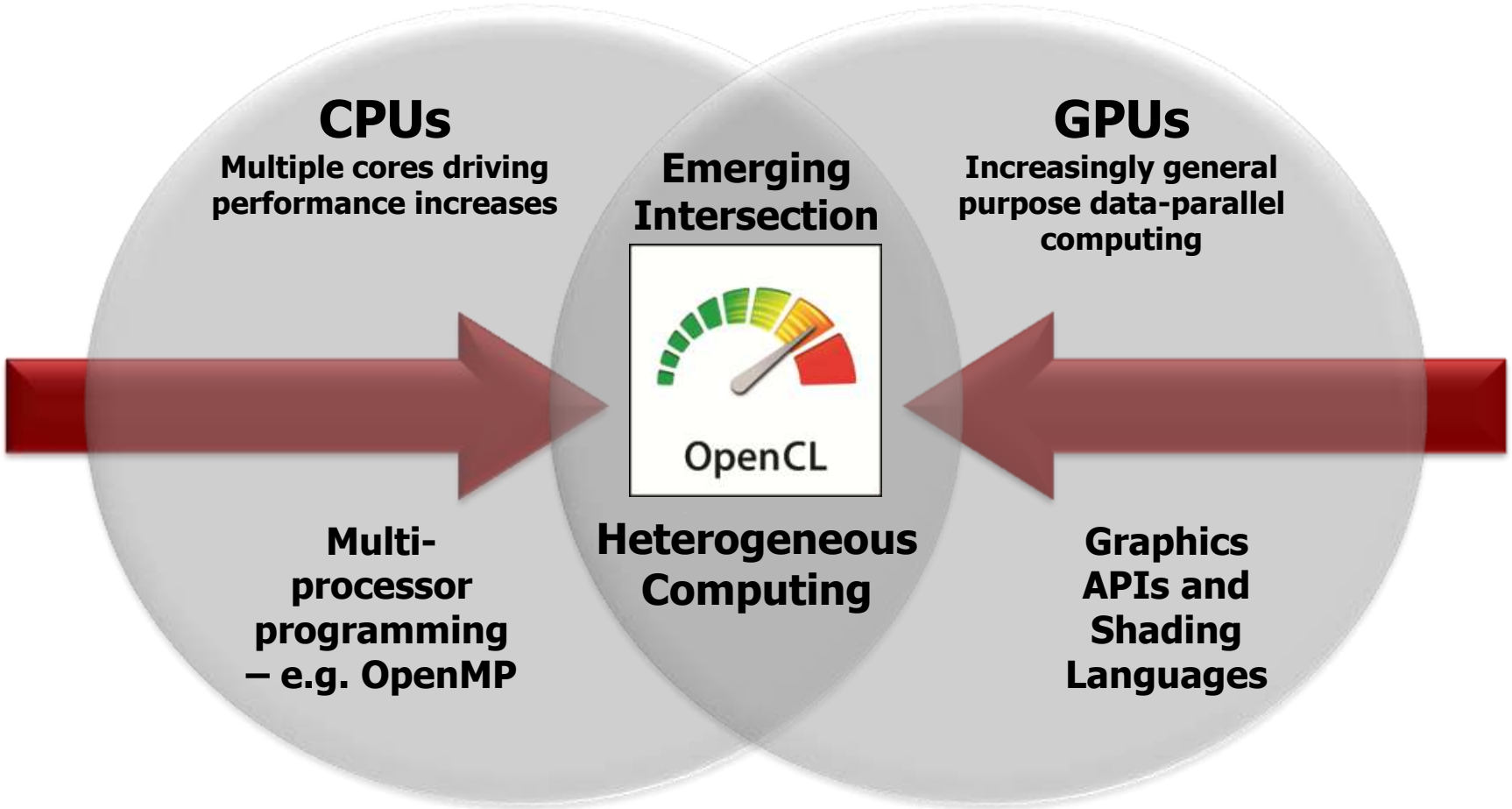
Introduction and Overview

June 2010

© Copyright Khronos Group, 2010 - Page 1

© Copyright Khronos Group, 2010 - Page 2

Processor Parallelism



OpenCL is a programming framework for heterogeneous compute resources

KHRONOS GROUP

OpenCL Working Group

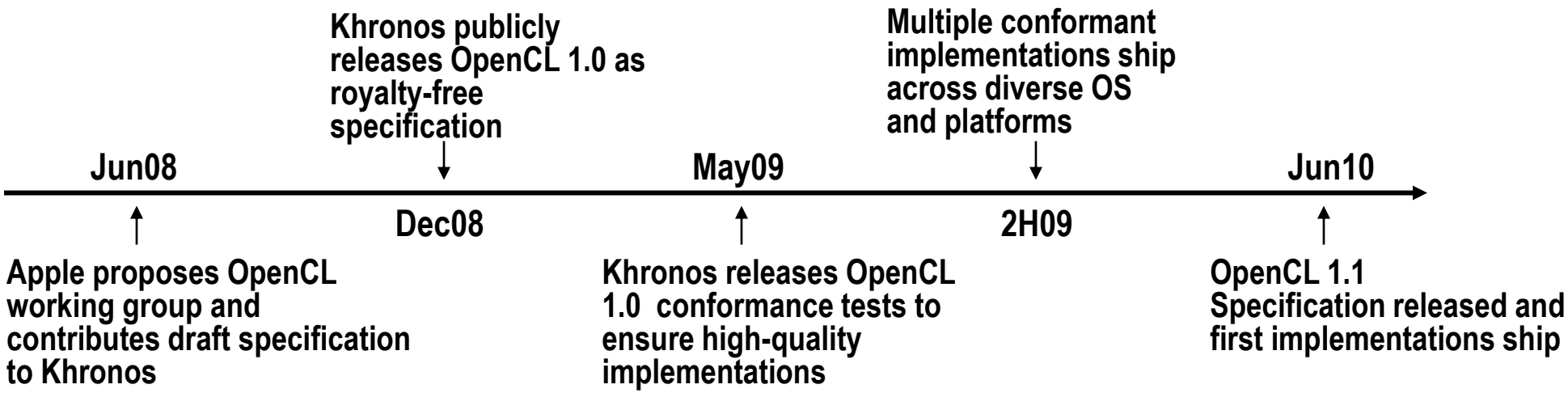
- **Diverse industry participation**
  - Processor vendors, system OEMs, middleware vendors, application developers
- **Many industry-leading experts involved in OpenCL's design**
  - A healthy diversity of industry perspectives
- **Apple made initial proposal and is very active in the working group**
  - Serving as specification editor

KHRONOS GROUP

© Copyright Khronos Group, 2010 - Page 4

OpenCL Timeline

- **Six months from proposal to released OpenCL 1.0 specification**
  - Due to a strong initial proposal and a shared commercial incentive
- **Multiple conformant implementations shipping**
  - Apple's Mac OS X Snow Leopard now ships with OpenCL
- **18 month cadence between OpenCL 1.0 and OpenCL 1.1**
  - Backwards compatibility protect software investment



© Copyright Khronos Group, 2010 - Page 5

OpenCL 1.1

- **General uplift in functionality for enhanced performance/programmability**
  - Including feedback from developer community
- **New functionality - fully backwards compatible with OpenCL 1.0**
  - New data types
    - Including 3-component vectors and additional image formats
  - Handling commands from multiple hosts
    - Processing buffers across multiple devices
  - Operations on regions of a buffer
    - Including read, write and copy of 1D, 2D or 3D rectangular regions
  - Enhanced use of events
    - To drive and control command execution
  - Additional OpenCL C built-in functions
    - Such as integer clamp, shuffle and asynchronous strided copies
  - Improved OpenGL interoperability
    - Efficient sharing of images and buffers by linking OpenCL and OpenGL events

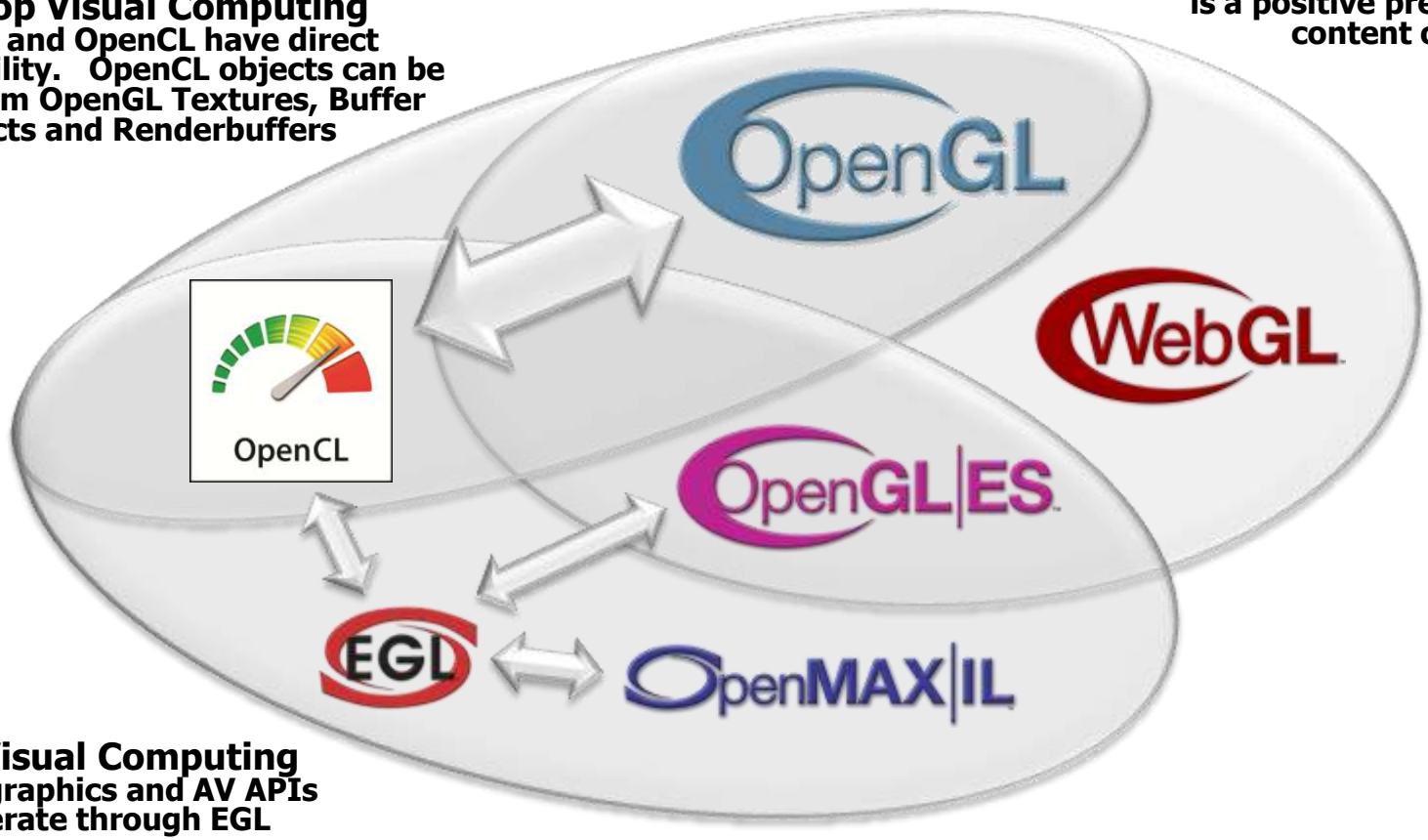
KHRONOS GROUP

© Copyright Khronos Group, 2010 - Page 6

OpenGL-based Ecosystem

Roadmap Convergence
OpenGL 4.0 and OpenGL ES 2.0 are both streamlined, programmable pipelines. GL and ES working groups are working on convergence. WebGL is a positive pressure for portable 3D content on all platforms

Desktop Visual Computing
OpenGL and OpenCL have direct interoperability. OpenCL objects can be created from OpenGL Textures, Buffer Objects and Renderbuffers



Mobile Visual Computing
Compute, graphics and AV APIs interoperate through EGL

KHRONOS GROUP

© Copyright Khronos Group, 2010 - Page 7

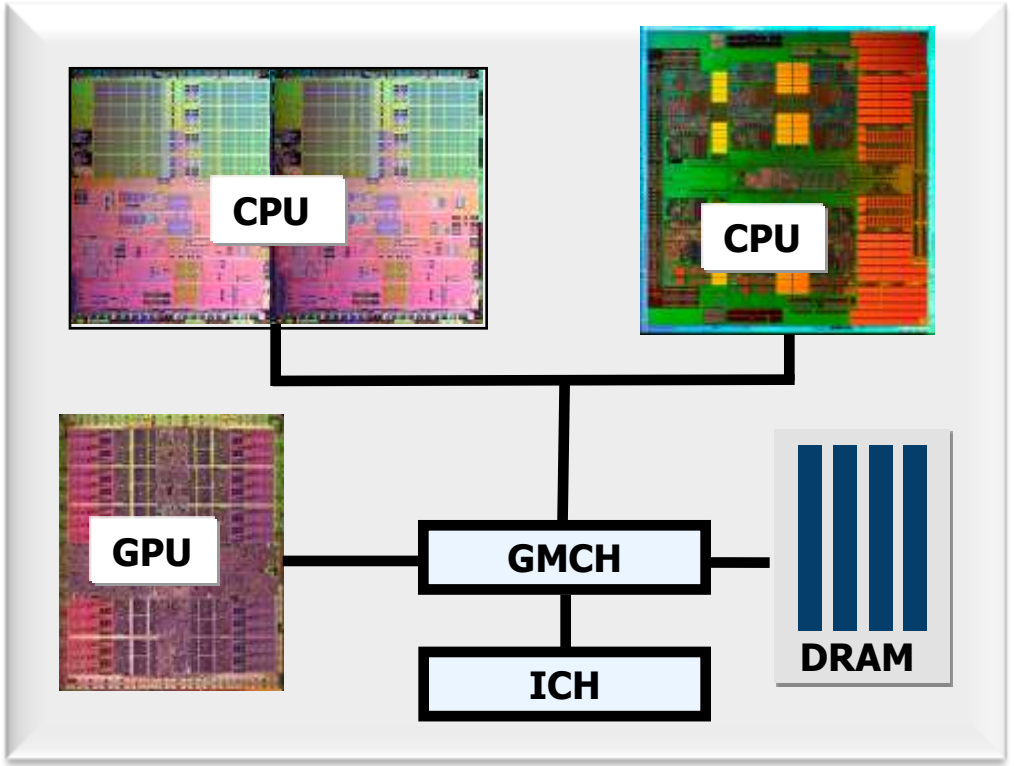
OpenCL Overview

© Copyright Khronos Group, 2010

It's a Heterogeneous World

- **A modern platform Includes:**
  - One or more CPUs
  - One or more GPUs
  - DSP processors
  - … other?

OpenCL lets Programmers write a single portable program that uses ALL resources in the heterogeneous platform



GMCH = graphics memory control hub
ICH = Input/output control hub

KHRONOS GROUP

The BIG Idea behind OpenCL

- **OpenCL execution model …**
  execute a kernel at each point in a problem domain
  - E.g., process a 1024 x 1024 image with one kernel invocation per pixel or 1024 x 1024 = 1,048,576 kernel executions

Traditional loops

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

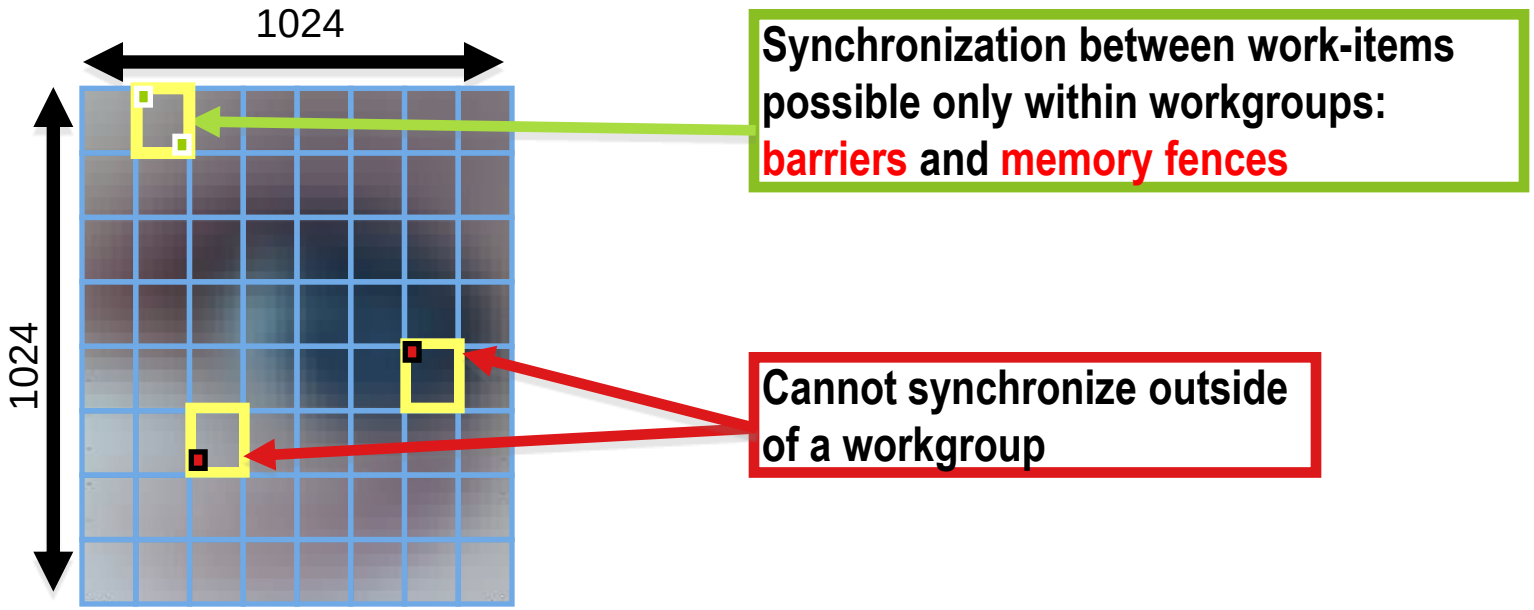
Data Parallel OpenCL

```
kernel void
dp_mul(global const float *a,
       global const float *b,
       global float *c)
{
  int id = get_global_id(0);

  c[id] = a[id] * b[id];

} // execute over "n" work-items
```

KHRONOS GROUP

An N-dimension domain of work-items

- **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)



© Copyright Khronos Group, 2010 - Page 11

To use OpenCL, you must

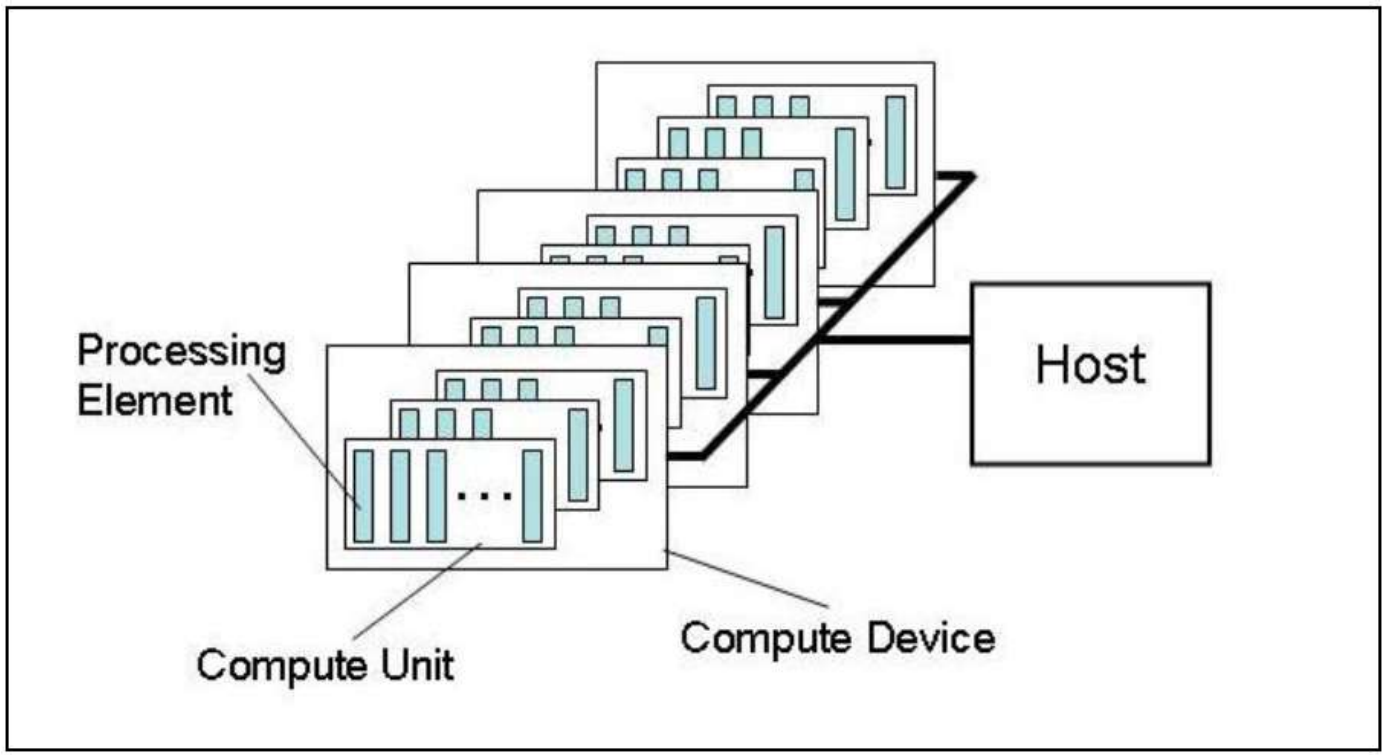
- Define the platform
- Execute code on the platform
- Move data around in memory
- Write (and build) programs

KHRONOS GROUP

© Copyright Khronos Group, 2010 - Page 12

OpenCL Platform Model

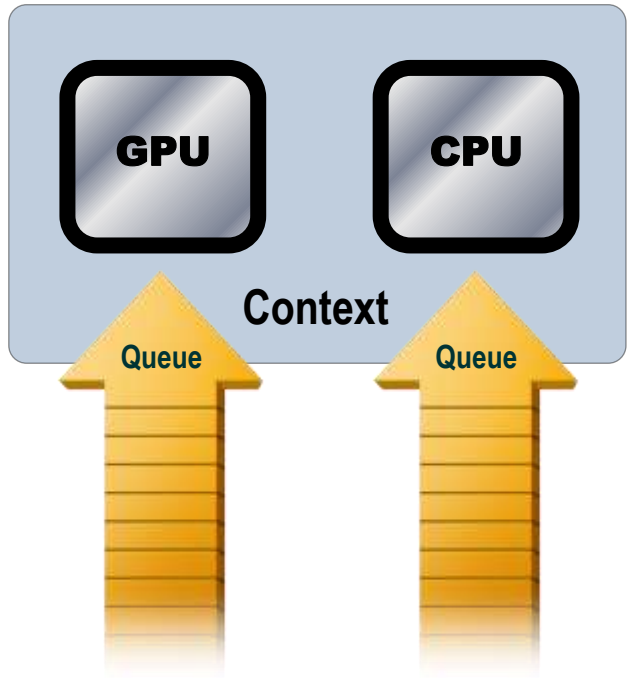
- **One Host + one or more Compute Devices**
  - Each Compute Device is composed of one or more Compute Units
    - Each Compute Unit is further divided into one or more Processing Elements



© Copyright Khronos Group, 2010 - Page 13

OpenCL Execution Model

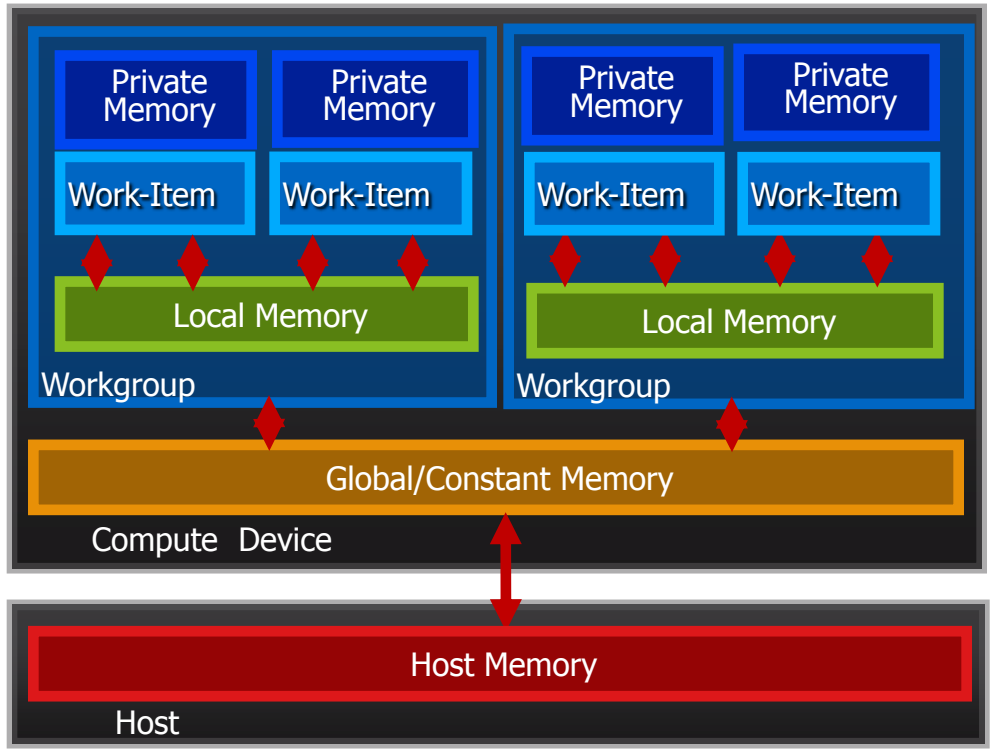
- **An OpenCL application runs on a host which submits work to the compute devices**
  - **Work item**: the basic unit of work on an OpenCL device
  - **Kernel**: the code for a work item. Basically a C function
  - **Program**: Collection of kernels and other functions (Analogous to a dynamic library)
  - **Context**: The environment within which work-items executes … includes devices and their memories and command queues
- **Applications queue kernel execution instances**
  - Queued in-order … one queue to a device
  - Executed in-order or out-of-order



© Copyright Khronos Group, 2010 - Page 14

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

© Copyright Khronos Group, 2010 - Page 15

Programming kernels: OpenCL C Language

- **A subset of ISO C99**
  - But without some C99 features such as standard C99 headers, function pointers, recursion, variable length arrays, and bit fields
- **A superset of ISO C99 with additions for:**
  - Work-items and workgroups
  - Vector types
  - Synchronization
  - Address space qualifiers
- **Also includes a large set of built-in functions**
  - Image manipulation
  - Work-item manipulation,
  - Specialized math routines, etc.

KHRONOS GROUP

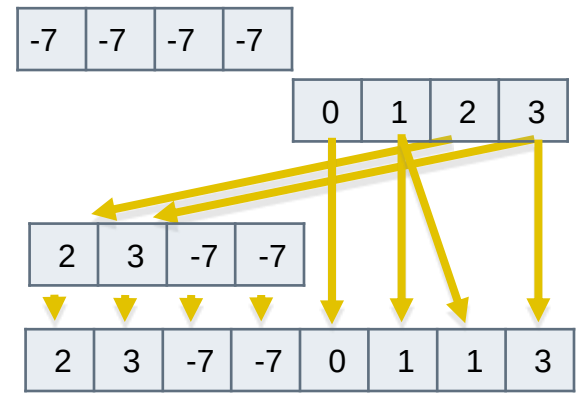
© Copyright Khronos Group, 2010 - Page 16

Programming Kernels: Data Types

- **Scalar data types**
  - char , uchar, short, ushort, int, uint, long, ulong, float
  - bool, intptr_t, ptrdiff_t, size_t, uintptr_t, void, half (storage)
- **Image types**
  - image2d_t, image3d_t, sampler_t
- **Vector data types**
  - Vector lengths 2, 4, 8, & 16 (char2, ushort4, int8, float16, double2, …)
  - Endian safe
  - Aligned at vector length
  - Vector operations
  - Built-in functions

Double is an optional type in OpenCL 1.0

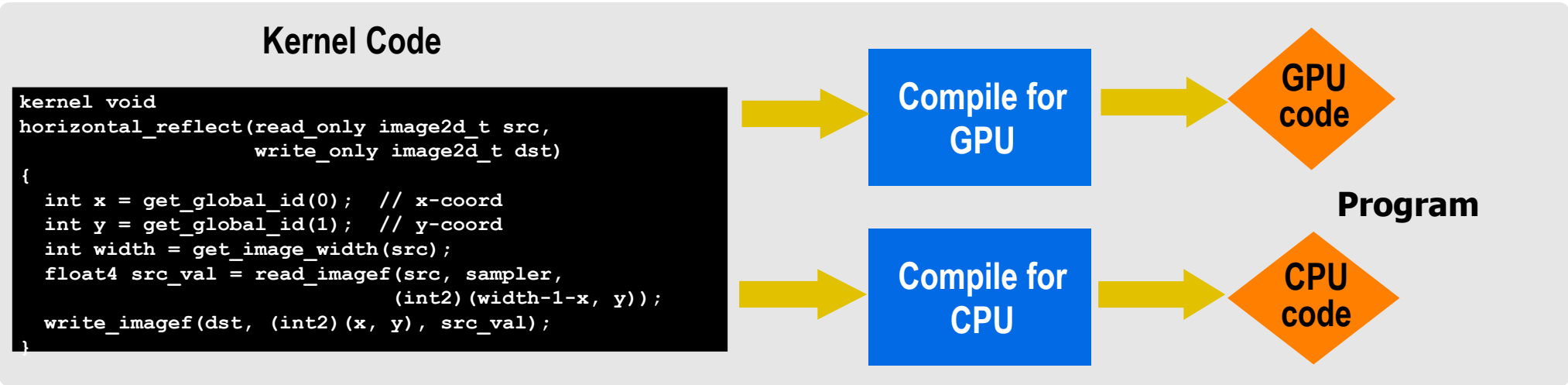
```
int4 vi0 = (int4) -7;
int4 vi1 = (int4)(0, 1, 2, 3);
vi0.lo = vi1.hi;
int8 v8 = (int8)(vi0, vi1.s01, vi1.odd);
```



© Copyright Khronos Group, 2010 - Page 17

Building Program Objects

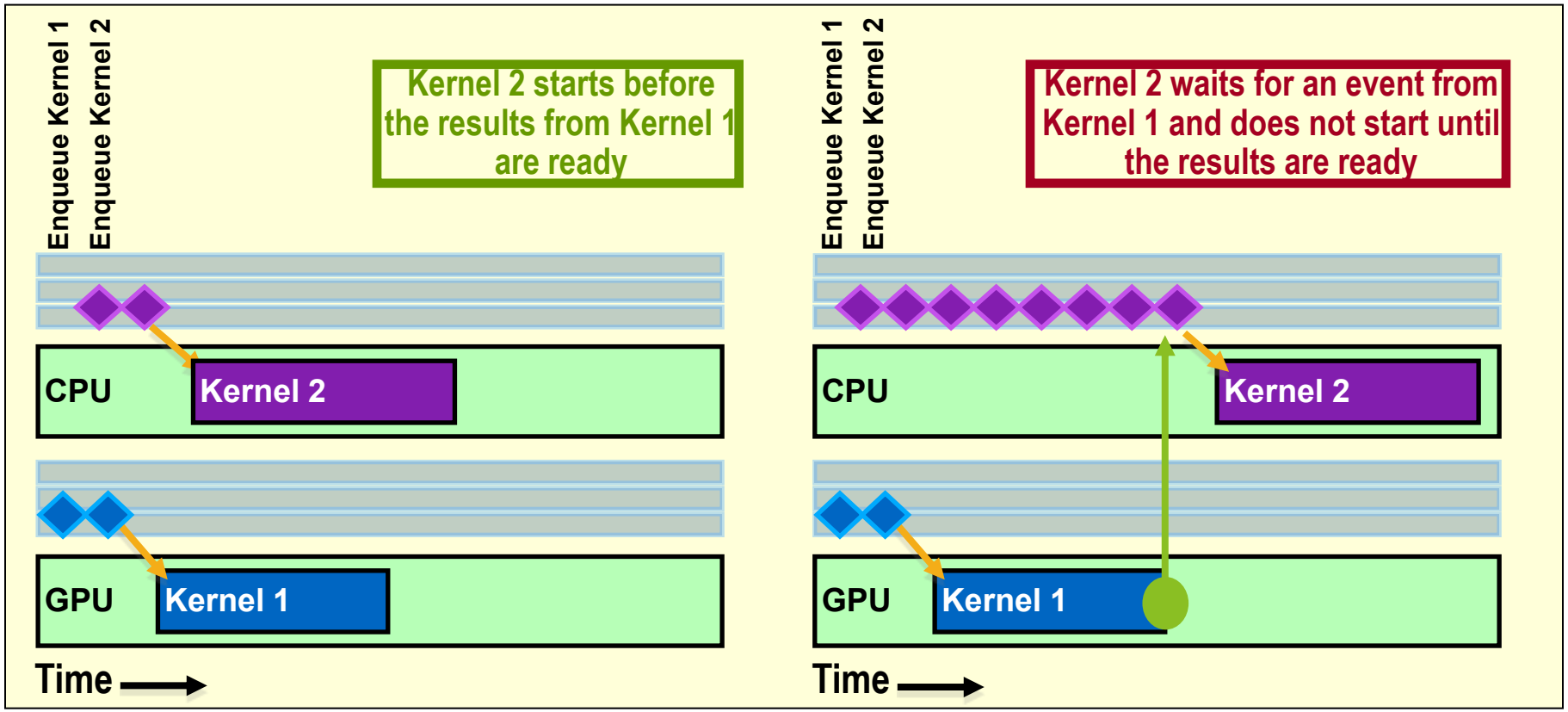
- **The program object encapsulates:**
  - A context
  - The program source/binary
  - List of target devices and build options
- **The Build process … to create a program object**
  - clCreateProgramWithSource()
  - clCreateProgramWithBinary()



© Copyright Khronos Group, 2010 - Page 18

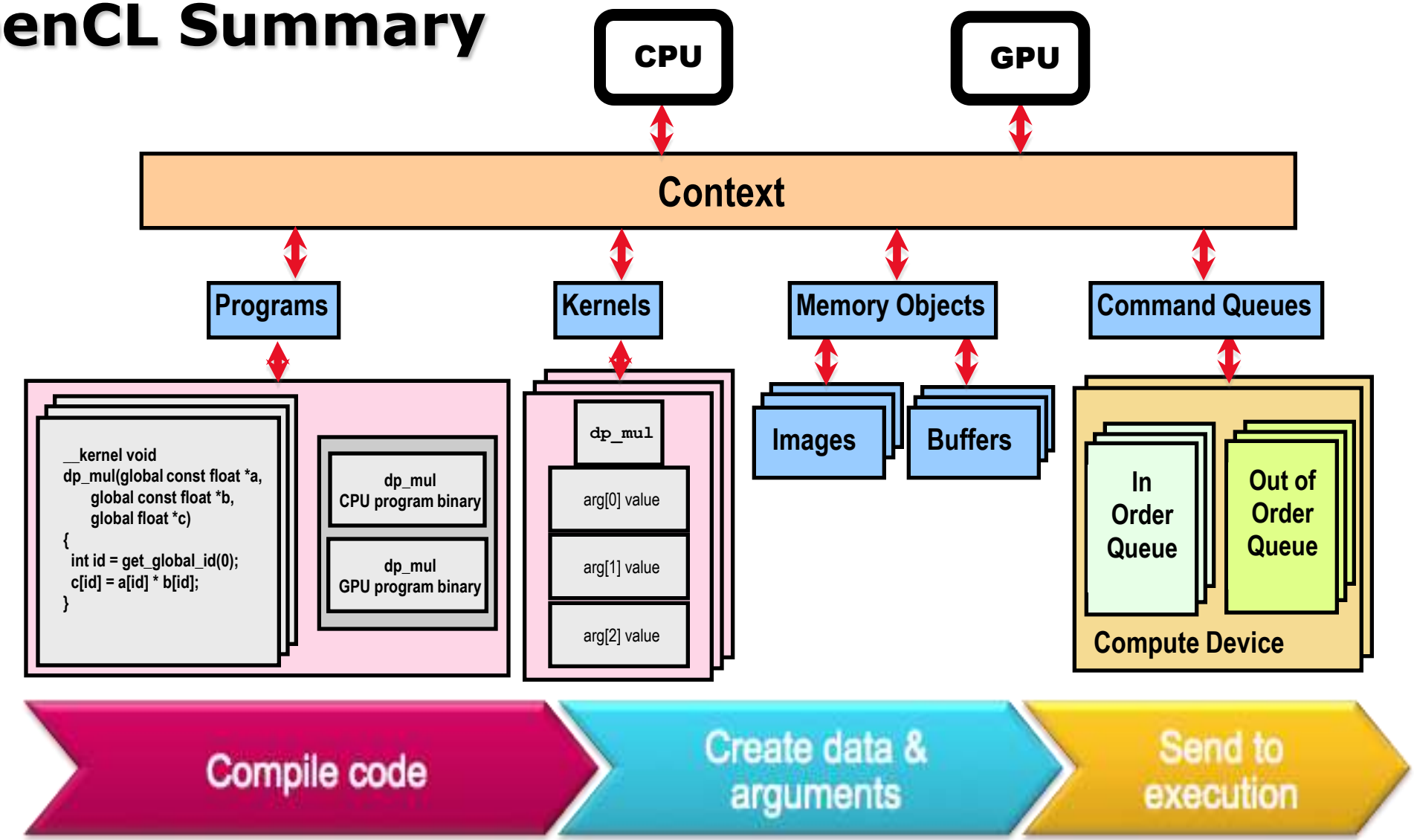
OpenCL Synch: Queues & Events

- Events can be used to synchronize kernel executions between queues
- Example: 2 queues with 2 devices



© Copyright Khronos Group, 2010 - Page 19

OpenCL Summary



© Copyright Khronos Group, 2010 - Page 20