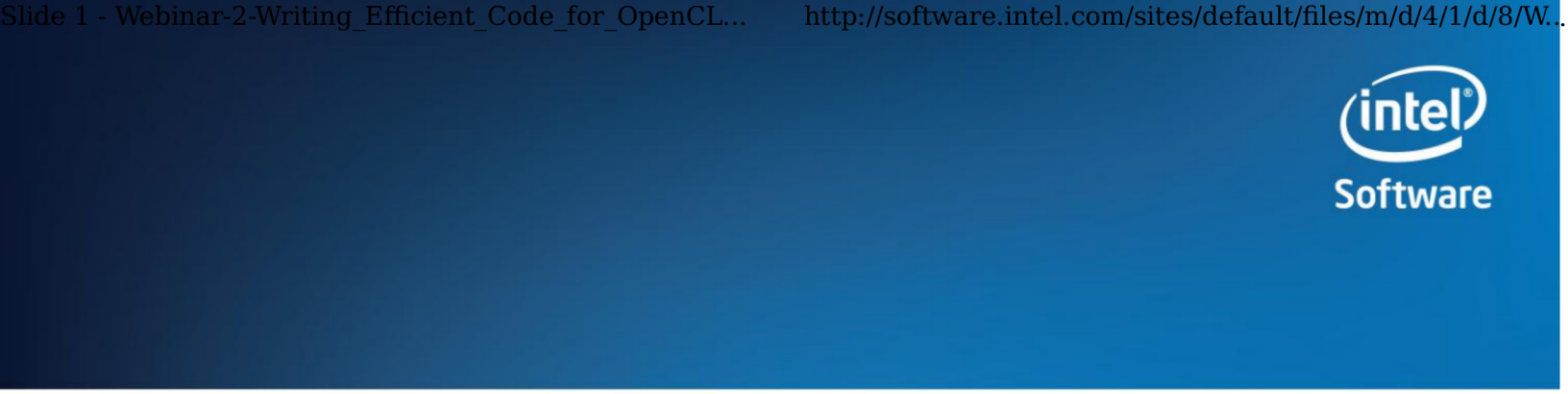


Writing Efficient Code for OpenCL* Applications on 3rd Generation Intel® Core™ Processors

Webinar #2 in the Three-part OpenCL* Webinar Series

July 18, 2012

All Webinars in the Series Available Online

- Welcome to "Writing Efficient Code for OpenCL* Applications" the 2nd in a 3-part OpenCL* webinar series presented by Intel.

- Register now for the last webinar in the series:
  - July 11 - Getting Started with Intel® SDK for OpenCL* Applications
  - July 18 - Writing Efficient Code for OpenCL* Applications
  - July 25 - Creating and Optimizing OpenCL* Applications

- Watch previous webinar:
  - Getting Started with Intel® SDK for OpenCL* Applications

To Register for Webinars in the Series
or See Videos of the Webinars Go to …
Intel.com/software/vcsource
Then click on Events in the left hand navigation

Optimization Notice

About the Presenters

Brijender Bharti

Engineering Manager,
Client Performance team,
Intel Corporation

Nao Yoshikawa

Director,
OEM Marketing & Services (Americas)
ArcSoft Inc.

Intel® SDK for OpenCL* Applications 2012

Intel Software

SDK Package

Development Environment

- Supports Both CPU and Intel HD Graphics

Development Tools

- Offline Compiler
- CPU Kernel Debugger

Documentation and Samples

- Getting Started guide
- Optimization Guide

Drivers and Support

Runtime/Drivers

- Intel® HD Graphics Driver with support for both CPU and Graphics
- Intel® SDK for OpenCL* – CPU only runtime package

Online Resources

- Code Samples
- Support Forum
- Tech Articles, and more.

Optimization Tools

- Intel® VTune Amplifier XE
- Intel® Graphics Performance Analyzers (Intel® GPA)

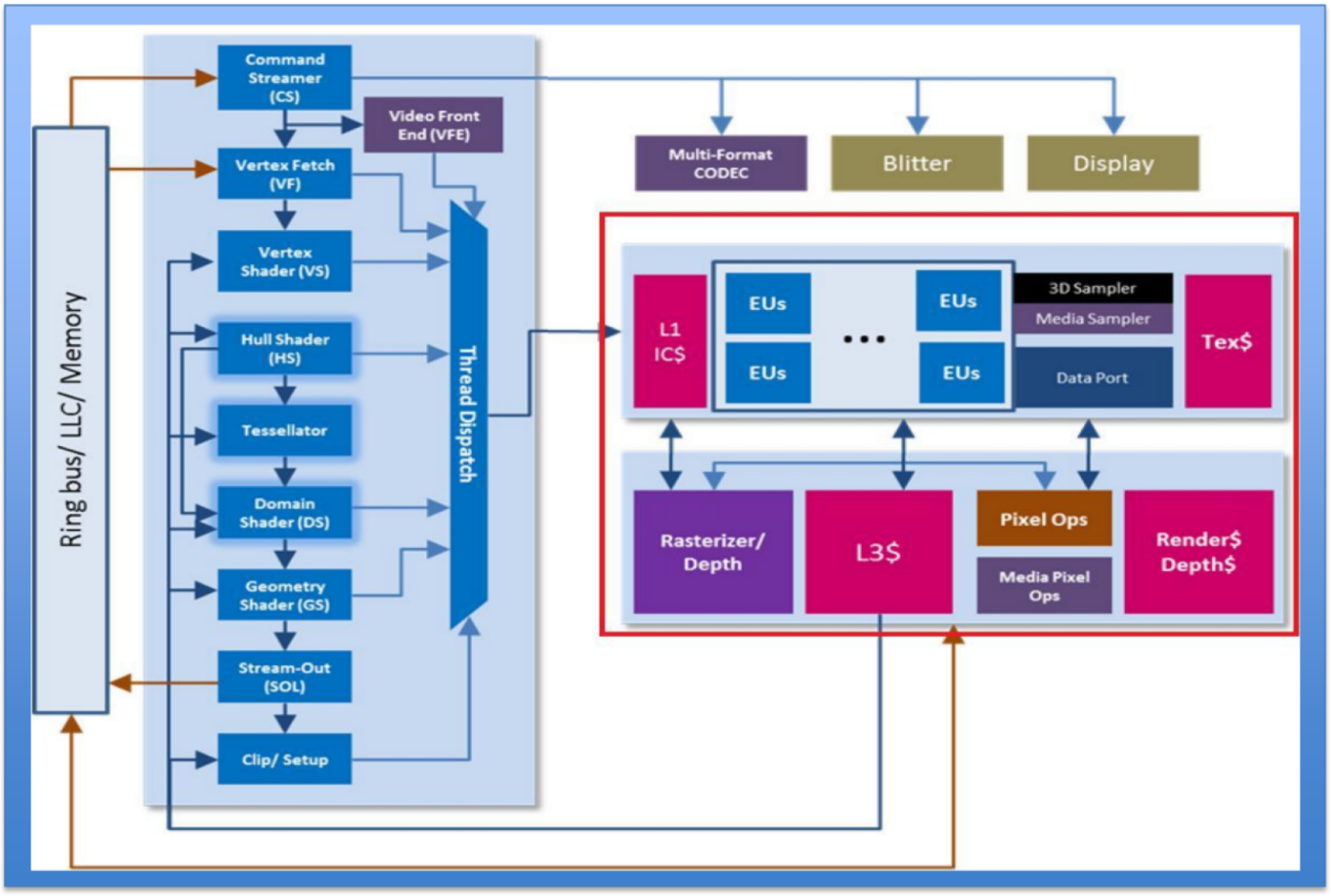
Intel SDK for OpenCL Applications Makes Development Faster and Easier

Optimization Notice

Contents

- OpenCL* on 3rd Generation Intel® Core™ Processors
- OpenCL Kernel Optimization
- Optimizing for Memory
- Intel® Quick Sync Video and OpenCL* Interoperability
- ArcSoft* Inc. Presentation
  - Utilizing OpenCL for Multimedia Applications on 3rd Generation Intel Core Processors
- What's next

Intel® HD Processor Graphics



OPENCL* KERNEL OPTIMIZATION

Avoid Invariant Operations in Kernel

Software

```
__kernel void test1(__global int* data,
int2 size, int base )
{
  int offset = size.y * base + size.x;
  float offsetF = (float)offset;
  …
}
```

```
__kernel void test1(__global int* data, int
offset, float offsetF )
{
    …
```

Move Invariant Code to Host

Optimization Notice

Avoid Initialization in Kernel


```
__kernel void something(__global int* data)
{
  size_t tid = get_global_id(0);
  if (0 == tid)
  {
    //Do initializing
   }
   barrier(CLK_GLOBAL_MEM_FENCE);
   //Regular kernel code
}
```

Move One Time Initialization to Host Code, or in a Different Kernel

Use Built-in Functions

```
dp = V0.x * V1.x + V0.y * V1.y
   + V0.z * V1.z + V0.w + V1.w;
```

→

```
dp = dot(V0, V1);
```

```
C = sqrt(A * A + B * B);
```

→

```
C = hypot(A, B);
```

```
cl =
fmin(fmax(X,minVal),maxVal);
```

→

```
cl = clamp(X,
minVal, maxVal);
```

Allow the Compiler to Optimize Better

Trade Accuracy vs. Speed

```
c[id] += a[id]*b[id];
```
→
```
c[id] = mad(a[id],b[id],c[id]);
```

```
a[id] = sin(a[id]);
```
→
```
a[id] = native_sin(a[id]);
```

Use Native_* Versions of Trigonometry Functions or Compile with –cl-fast-relaxed-math Option

Edge Conditions

```
__kernel void myKernel(__global int* data, int maxR, int maxC)
{
  int row = get_global_id(0);
  int col = get_global_id(1);
  if (row > maxRow)
  {
    …
  }
  else if (col > maxCol)
  {
    …
  }
  else
  {
    …
  }
}
```

Possible alternatives:

- Limit the NDRange
- Use padded buffers
- Set edge values using Min/Max built-ins
- Use images for Processor Graphics

Avoid Checking Edge-Conditions

Avoid Byte/Short Load and Stores

```
__kernel void k(global uchar* a, global uchar* b)
{
  int gid = get_global_id(0);
  a[gid] *= b[gid];
  . . .
}
```

```
__kernel void k(global uint4* a, global uint4* b)
{
  int gid = get_global_id(0);
  a[gid] *= b[gid];
  . . .
}
```

Avoid Byte or Short Loads. Load and Store in Greater Chunks

Maximizing Overall Utilization

Intel® HD Processor Graphics Device Optimization

- Optimize the number of work-groups
- Optimize the work-group size
- Use barriers in kernels wisely
- Optimize thread utilization

Amount of SLM required by kernel	Recommended work-group size	Max # of concurrently running work-groups	Comments
0	16	128	Good (Not using SLM/barriers)
4K	64	32	Good
8K	128	16	Good
16K	256	8	Good
32K	512	4	Good

Maximizing Overall Utilization

Intel® CPU Device Optimization

- Use vector data types. E.g. float8, double4, int4 etc.
- Preferred work group size for kernels is 64 or 128 work items.
  - Workgroup size multiple of 8
  - Query CL_KERNEL_PREFERRED_WORK_GROUP_SIZE_MULTIPLE parameter by calling to clGetKernelWork-groupInfo
  - Match number of work groups to logical cores.

OPTIMIZING FOR MEMORY

Buffer Objects Vs. Image Objects

- CPU Device
  - Avoid using Image Object
  - Buffer Objects are preferred
- Intel® HD Processor Graphics Device
  - **For Buffer Objects**
    - Share physical memory between CPU and Intel® HD Processor Graphics (CL_MEM_USE_HOST_PTR)
    - Preferred for Look-up tables
    - Use local memory for explicit caching of buffer values
  - **For Image Objects**
    - Sampler capabilities – clamping coordinates, interpolation
    - Packing/Unpacking of elements e.g. uchar4 to uint4
    - Prefer Image objects if access pattern is irregular (not linear)

Try Both Object Types

Shared Local Memory – Processor Graphics

Intel® HD Processor Graphics Optimization

- The size of Shared Local Memory (SLM) can be found by `clGetDeviceInfo(…CL_DEVICE_LOCAL_MEM_SIZE)`

```
__kernel void k(global int* d, global int* t,  local
int* slm)
{
  int gid = get_global_id(0);
  // populate "slm" with data from t
  barrier(CLK_LOCAL_MEM_FENCE);
  d[gid]  = slm[d[gid]];
}
```

Use SLM for Small Tables, Coefficients, Etc.

Data Copy to Shared Local Memory

(When SLM entries are not equal to number of work items)

Intel® HD Processor Graphics Optimization

```
__kernel void foo(global int* t, local int* l /*256 enteries*/)
{ //initialize SLM (performed for each work-item!)
  for( uint id = 0;  id < 256;  id ++ )
     l[id] = t[id];
  barrier(CLK_LOCAL_MEM_FENCE);
 . . .
```

```
__kernel void foo(global int* t, local int* l /*256 enteries*/)
{ //initialize SLM
int   lidx = get_local_id(0);
int   size_x = get_local_size(0);
for( uint   id = lidx; id < 256; id += size_x )
   l[id] = t[id];
barrier(CLK_LOCAL_MEM_FENCE);
  . . .
```

Divide the SLM Copy Among the Local Work Items

Data Copy to Shared Local Memory

(When SLM entries are equal to number of work items in local work group)

Intel® HD Processor Graphics Optimization

```
__kernel void foo(global int* t, l int* l)
{ //initialize SLM
int   lidx = get_local_id(0);
int   lidy = get_local_id(1);

int   id = lidx + lidy * get_local_size(0);

l[id] = t[id];

barrier(CLK_LOCAL_MEM_FENCE);
  . . .
```

Each Work Item Copies One Table Entry

Memory Access

- Intel® CPU and Intel® HD Processor Graphics share Last Level Cache (LLC)
- Intel® HD Processor Graphic specific L3 Cache
  - Cache line size is 64 bytes
- Memory can be accessed at 8-bit, 16-bit, or 32-bit quantities
- Access at least 32-bit quantities from 32-bit aligned boundaries
  - Char4
  - int

Memory Access Pattern and Size is Important for the Performance

Access Patterns to L3 Cache

Intel Software

- Consider the indexing pattern: a[gid(1)*width+gid(0)]
- Let's look at what happens with work groups processing 16 elements:
  - By row: size <16, 1>
  - By tile: size <4, 4>
  - By column: <1, 16>
- For the "row" case, 16 consecutive integers are accessed within the workgroup, so all could fit in a single cache line:

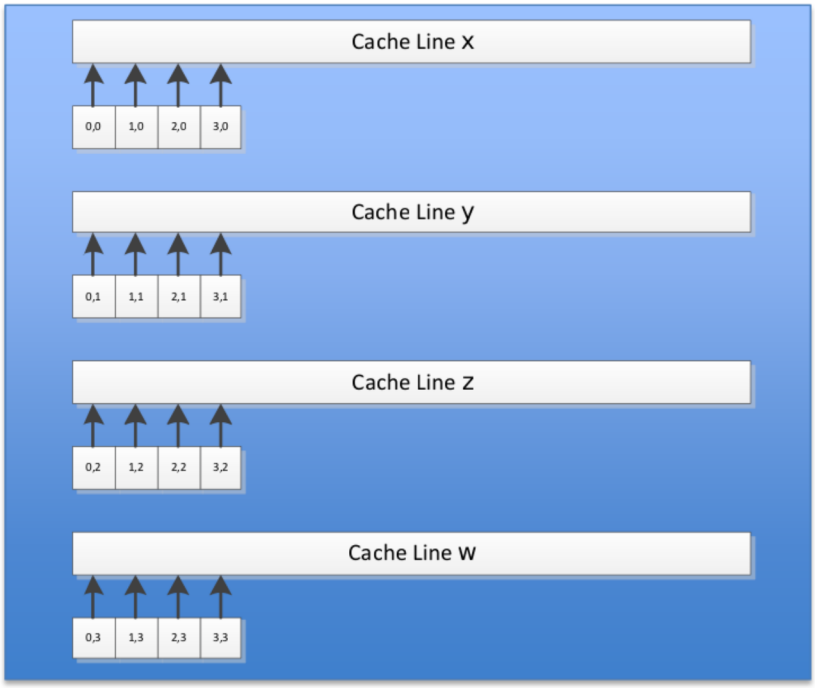
BEST Performance in This Case

Access Patterns to L3 Cache

- For the "tile" case, the global ID(1) is different for every four work items within the group, causing a stride of several cache lines:

- Only a quarter of the bandwidth is achieved!

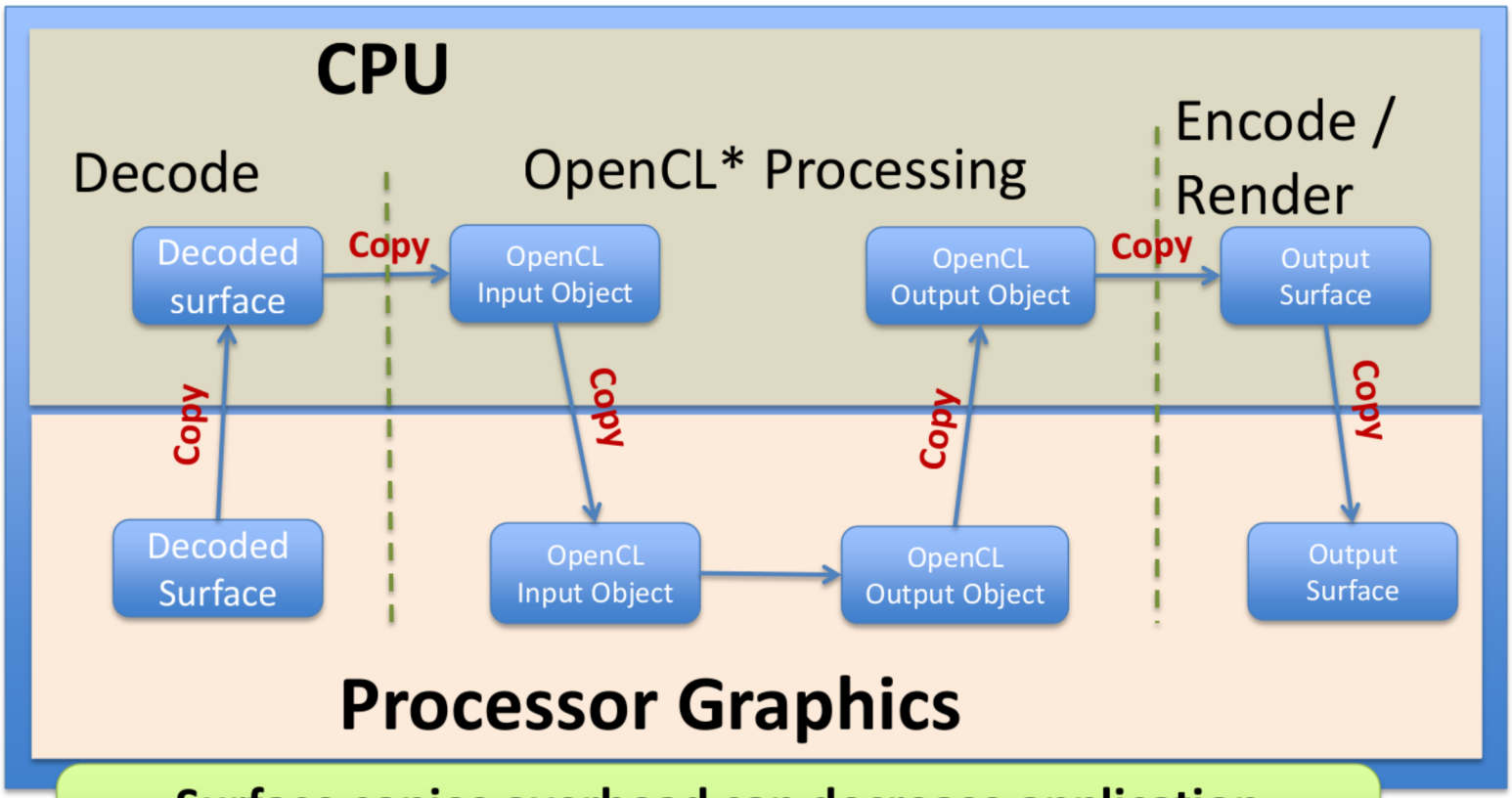
- For the "column" case, 16 cache lines are required



Performance Will be Impacted if Work Items Access Different Cache Lines in the Same Work Group

INTEL® QUICK SYNC VIDEO AND OPENCL* INTEROPERABILITY

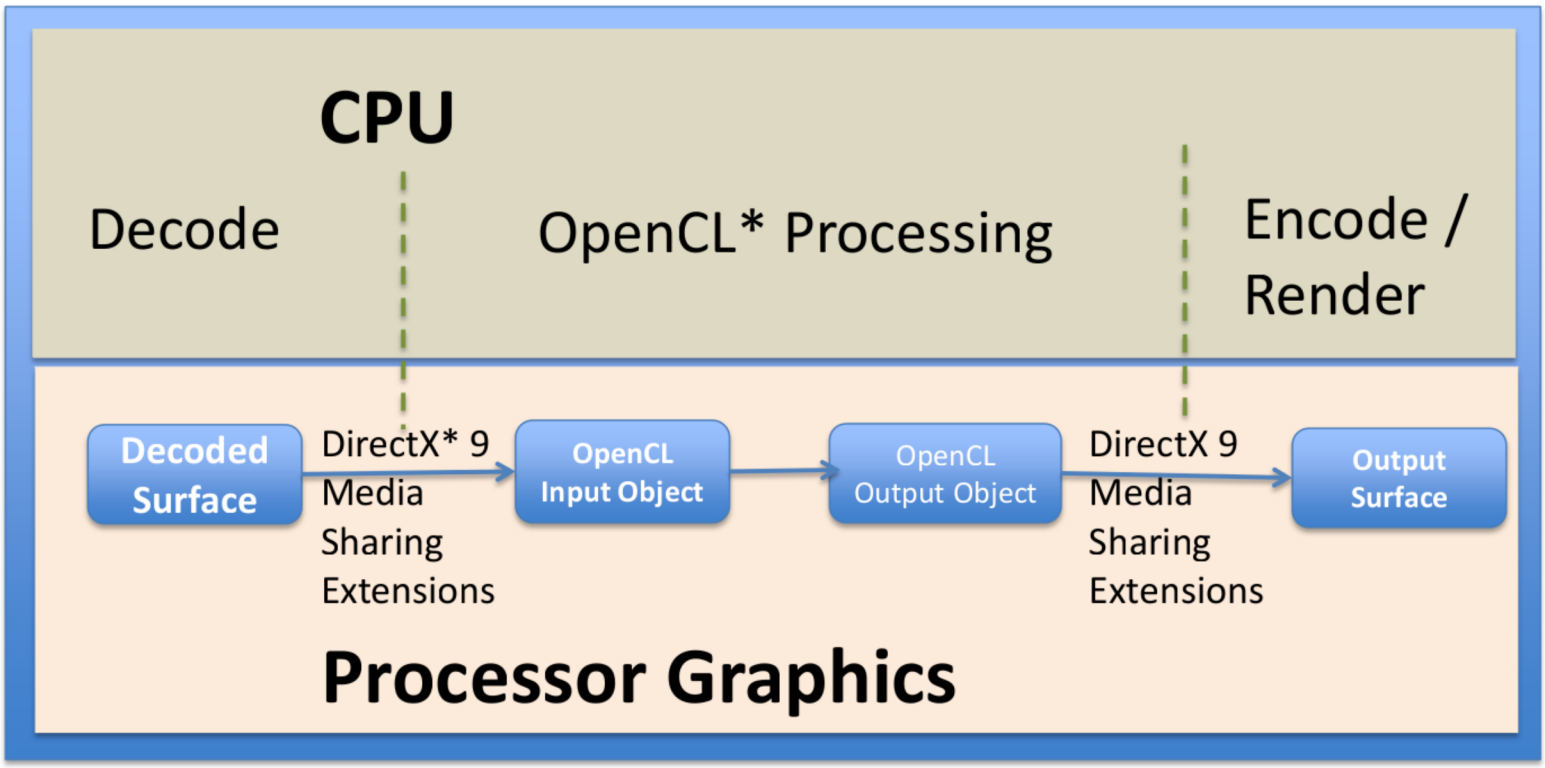
OpenCL* Kernels with Transcode Pipeline



Surface copies overhead can decrease application performance

Transcode Pipeline Using Intel® Quick Sync Video and OpenCL* Kernels

Microsoft DirectX* 9 Surface Sharing Extensions



Minimum or No surface Copies – Best Performance

Intel® Quick Synch Video USER Class Functions

Intel Software

- Intel® Quick Synch Video USER Class of Functions

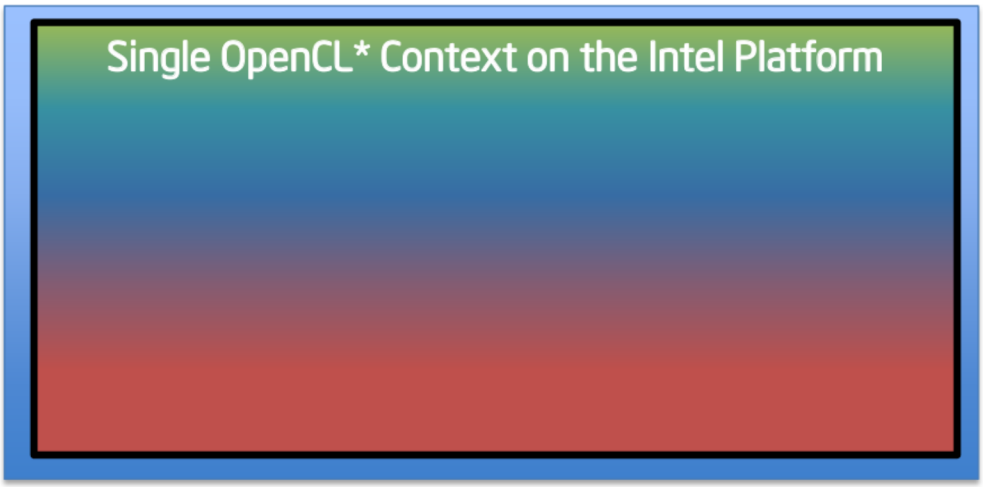
```
MFXVideoUSER_Register
MFXVideoUSER_ProcessFrameAsync
MFXVideoUSER_Unregister
```

- OpenCL* and DirectX* 9 Media Sharing APIs

```
clGetDeviceIDsFromDX9INTEL
clCreateFromDX9MediaSurfaceINTEL
clEnqueueAcquireDX9ObjectsINTEL
clEnqueueReleaseDX9ObjectsINTEL
```

Sample Code Available with Intel® SDK for OpenCL* Applications

Best Performance Using Multiple OpenCL* Devices

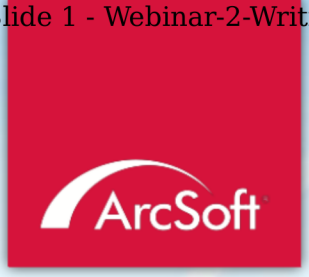


- Partitioning the work between devices
- Sharing resources efficiently
- Unified code base
- Synchronization (across command queues)

Intel® SDK for OpenCL* Applications - Optimization Guide

http://software.intel.com/sites/landingpage/opencl/optimization-guide/index.htm

Partner with Intel to Improve Application Performance on the Intel Platform



Utilizing OpenCL* for Multimedia Applications on 3rd Generation Core™ Intel Processors

July 2012
Nao Yoshikawa
Director, OEM Marketing and Services
ArcSoft. Inc.

Company Overview

ArcSoft is an industry-leading software developer of multimedia technologies and applications for PC, Mac, mobile phones, tablets, smart TVs, and cameras.

- Established in 1994
- Headquartered in Silicon Valley, California
- Over 800 employees worldwide

Great alliance with Intel

- Multi-core / multi-thread optimization on Dual / Quad core Intel CPUs
- SSE, AVX optimization
- Intel® Media SDK, Intel® Quick Sync Video support
- PAVP support for Blu-ray Disc* playback
- Working on OpenCL* optimization

ArcSoft®

Why OpenCL*?

•OpenCL* is an open standard heterogeneous parallel computing framework.

•Scalable. Balance resources between CPU and GPU
- Free CPU resource for non-graphic tasks
- Lower power consumption
- Scalable for various system configuration

•Cross-platform. Highly portable to various platforms
- PC, Mac*, Android*, etc
- OpenCL Embedded Profile for mobile and embedded devices

• Interoperable with Microsoft* DirectX* Video Acceleration (DXVA)
- Avoid extra copies of video buffer
- Easy to include in video decoding / transcoding pipeline

OpenCL* in ArcSoft Products

SimHD*

SD-to-HD video upscaling

- TotalMedia* Theatre
- ShowBiz*

Sim3D*

2D-to-3D video conversion

- TotalMedia* Theatre
- ShowBiz*

OpenCL* in ArcSoft Products

H.264 encoding

H.264 video encoding

* currently using Intel® Quick Sync Video on Intel platform

- ShowBiz*
- MediaConverter*

Panorama stitching

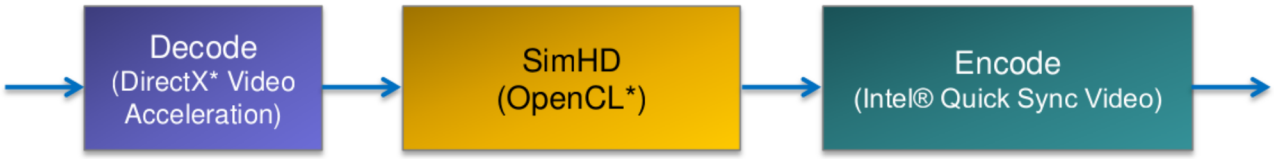
Stitch multiple pictures into one

- Panorama Maker*

ArcSoft

SimHD*

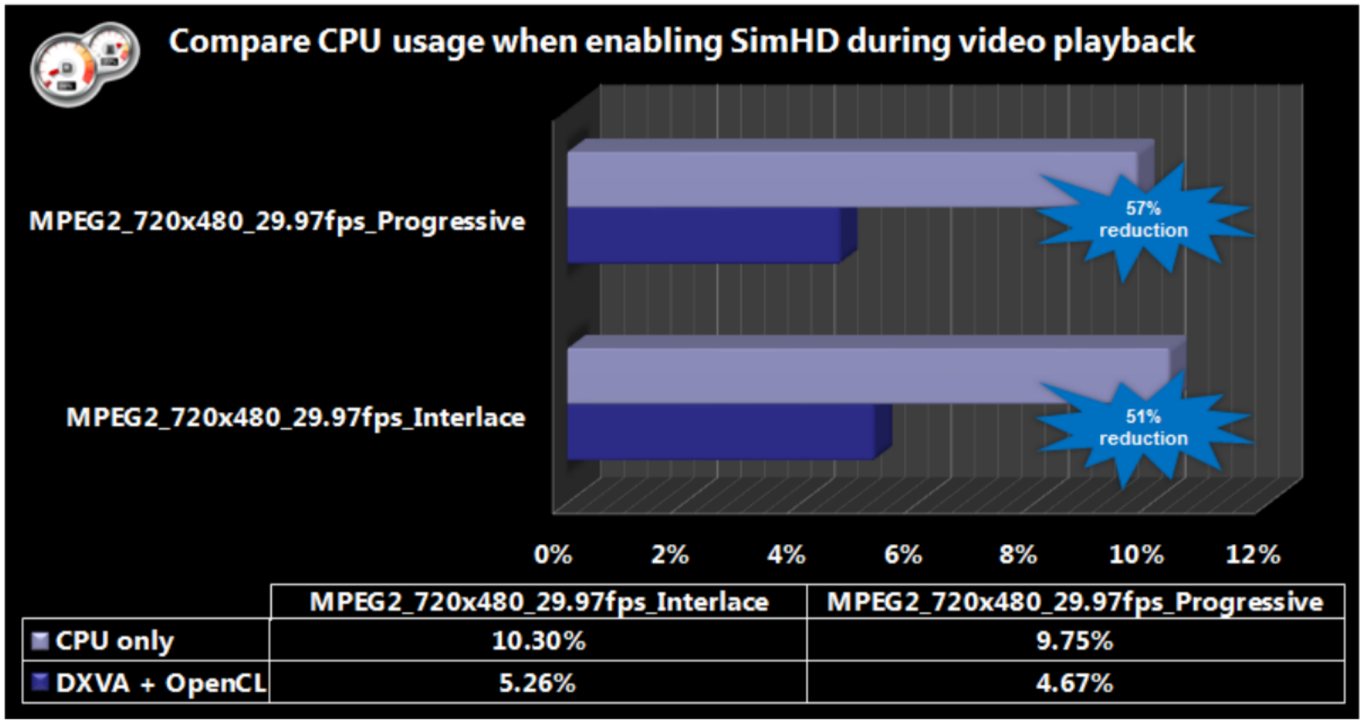
- SimHD*: SD-to-HD video upscaling engine
- Available in ArcSoft TotalMedia* Theatre (universal player) for playback
- Available in ArcSoft ShowBiz* (video editing & authoring) for encoding



ArcSoft

SimHD* Benchmark

- Test Platform
  - 3rd Generation Intel® Core™ Processor, 1.80GHz (4 core) with Intel® HD Graphics 4000
  - Intel® HD Graphics Driver Version: 8.15.10.2669



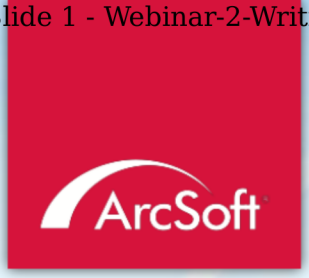
	MPEG2_720x480_29.97fps_Interlace	MPEG2_720x480_29.97fps_Progressive
CPU only	10.30%	9.75%
DXVA + OpenCL	5.26%	4.67%

Performance tests done by ArcSoft Inc. and are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary.

OpenCL* Tools, Progress and Goal

- Tools
  - Intel® SDK for OpenCL* Applications
  - DirectX* / OpenCL Interop API
  - Intel® Graphics Performance Analyzers (Intel® GPA) tuning tool
- Progress
  - Fine-tuning & optimizing OpenCL SimHD* on 3rd Generation Intel Core™ Processor
    - ✓ Optimizing playback pipeline to support DirectX Video Acceleration (DXVA) decoding and OpenCL SimHD together by using DirectX / OpenCL interop API
    - ✓ Reduce CPU usage, balance between CPU and GPU
  - Optimizing kernel to fully take advantage of Intel® HD Graphics
    - ✓ Use local share memory instead of global memory to improve data access rate
    - ✓ Use simple calculation instead of reading data from local/global memory
    - ✓ Reduce number of registers to increase the parallelism of the kernel
    - ✓ Use logical operation instead of comparison/condition operation
- Goal
  - Apply OpenCL to various ArcSoft* multimedia applications / engines

ArcSoft®



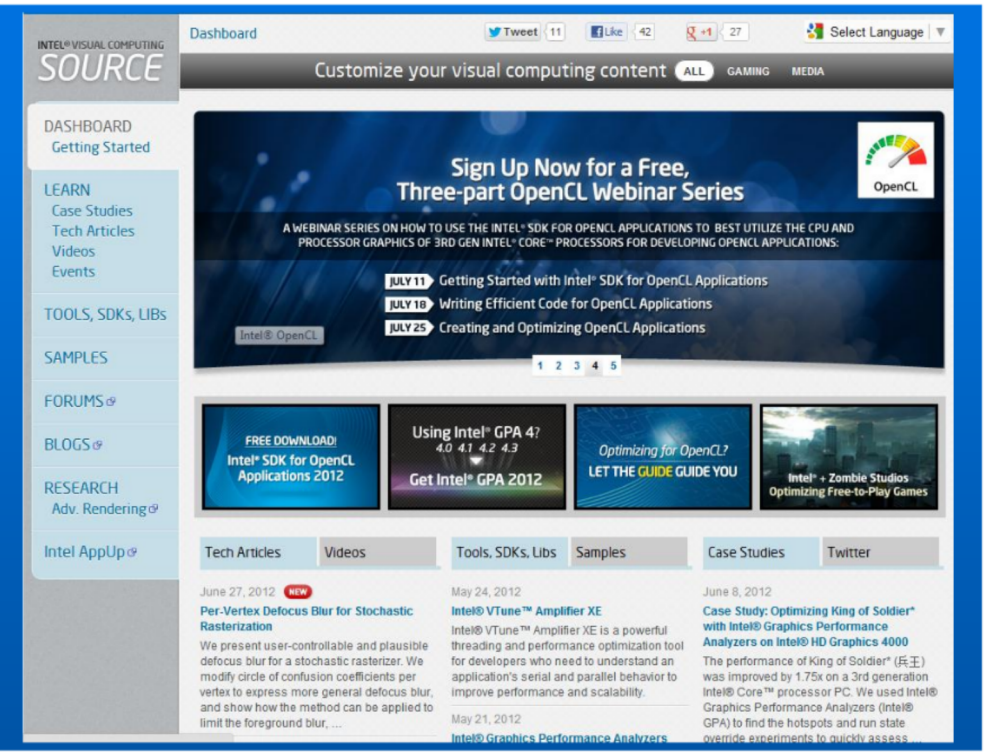
Thank you!

Q&A

The Intel® Visual Computing Source

Your Single Source for Information and Support

Software



- Free Downloads of Intel® Visual Computing Tools
- Code Samples
- Tech Articles
- Case Studies
- Forums
- Beta Programs

intel.com/software/vcsource
Intel.com/software/opencl

Programming with Intel® SDK for OpenCL* Applications Tools

Tool Name	Used for	Included with the SDK for OpenCL	Supported devices and operating systems:		
			Windows* OSs		Linux* OSs
			CPU	GPU	CPU
Offline Compiler (Standalone)	Create and Build	✓	✓	✓	✓
Offline Compiler (Microsoft Visual Studio* plug-in)	Create and Build	✓	✓	✓	
Step by Step Kernel Debugger	Debug	✓	✓		
Intel® VTune™ Amplifier XE	Tune	For purchase on intel.com	✓		✓
Intel® GPA System Analyzers	Tune	For free download on intel.com	✓	✓	
Intel® GPA Platform Analyzers with OpenCL* Traces Views	Tune	For free download on intel.com	✓		

- The Intel® VTune™ Amplifier XE 2011 is avalable for download at http://software.intel.com/en-us/articles/intel-vtune-amplifier-xe/
- Intel® Graphics Performance Analyzers 2012 (Intel® GPA) is avalable as a free download at www.intel.com/software/gpa

Want to See What Will Come Next? Register Now for Next Week's Webinar

Register now for the last webinar in the series

- July 11 - Getting Started with Intel® SDK for OpenCL* Applications
- July 18 - Writing Efficient Code for OpenCL* Applications
- **July 25 - Creating and Optimizing OpenCL* Applications**

To Register for Webinars in the Series
or See Videos of the Webinars Go to …
Intel.com/software/vcsource
Then click on Events in the left hand navigation

Meet With Us

Mark the date and come and meet us in person

- **Siggraph 2012**
  August 5-9 | Los Angeles, California | Los Angeles Convention Center
- **IDF USA**
  September 11 – 13 | San Francisco, California
- **Supercomputing 2012**
  November 10 – 16, 2012 | Salt Lake City, Utah | Salt Palace Convention Center

software.intel.com/en-us/articles/vcsource-learn-events

WHAT'S NEXT?

- Download the Intel® SDK for OpenCL* Applications 2012 at No Cost
- Download Intel® Media SDK 2012 at No Cost
- Learn about OpenCL* tools: Register to next week webinar
- Join us at upcoming events
- Follow us on Twitter: @IntelOpenCL
- Join our users forum and post your questions

Legal Disclaimer & Optimization Notice

INFORMATION IN THIS DOCUMENT IS PROVIDED "AS IS". NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. INTEL ASSUMES NO LIABILITY WHATSOEVER AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO THIS INFORMATION INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

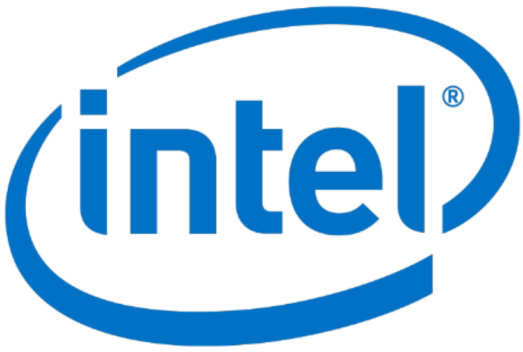
Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products.

Code snippets provided in this presentation are for illustrative purposes only. Intel disclaims any and all implied or express warranties associated with the code snippets, and any and all use of such code snippets is at the sole discretion and exclusive risk of the user.

Optimization Notice

Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice.

Notice revision #20110804


intel