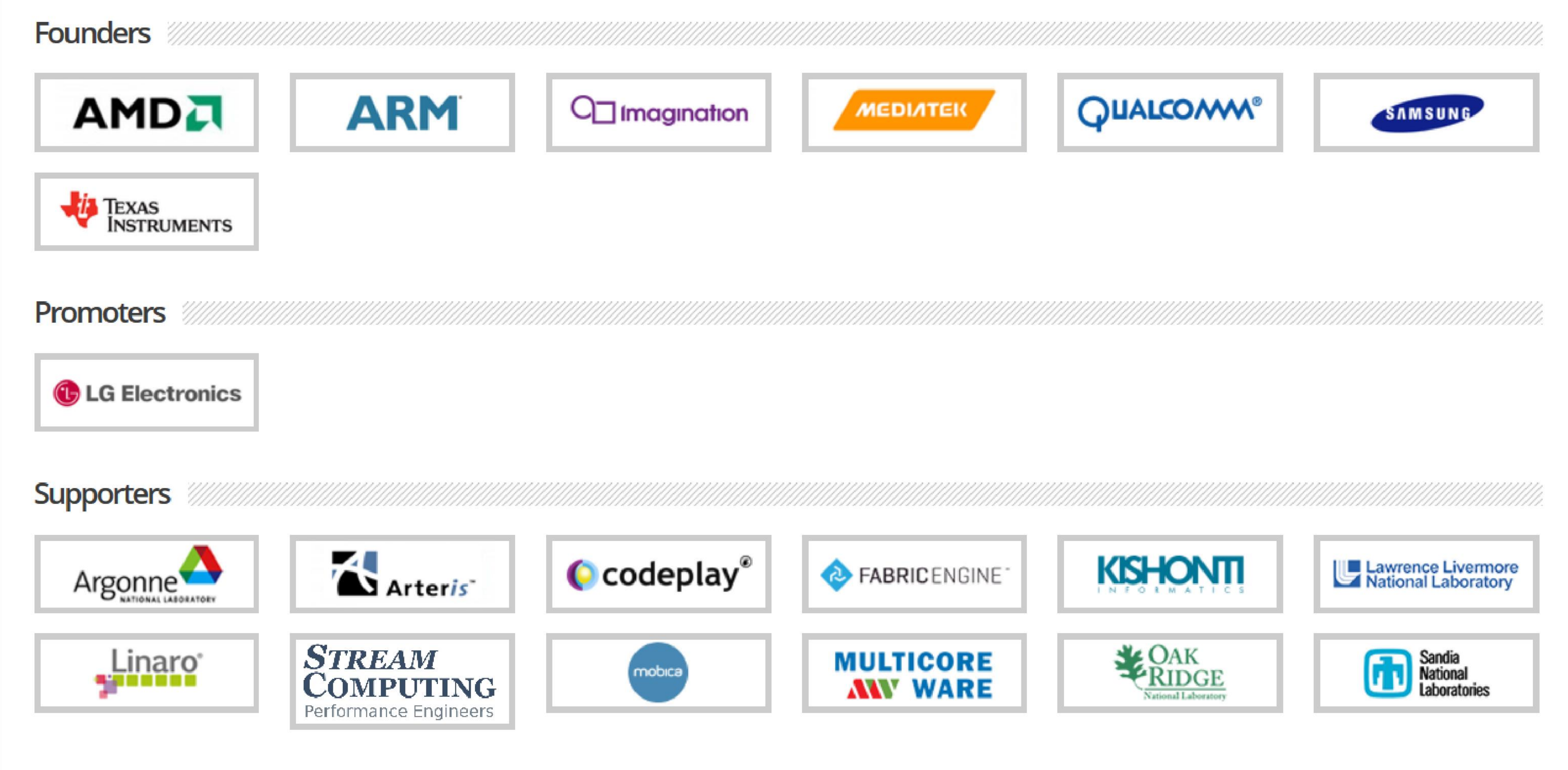
HSA

BENJAMIN COQUELLE
MAY 2015

HAS MEMBERS



Founders

- AMD
- ARM
- Imagination
- MEDIATEK
- QUALCOMM®
- SAMSUNG
- Texas Instruments

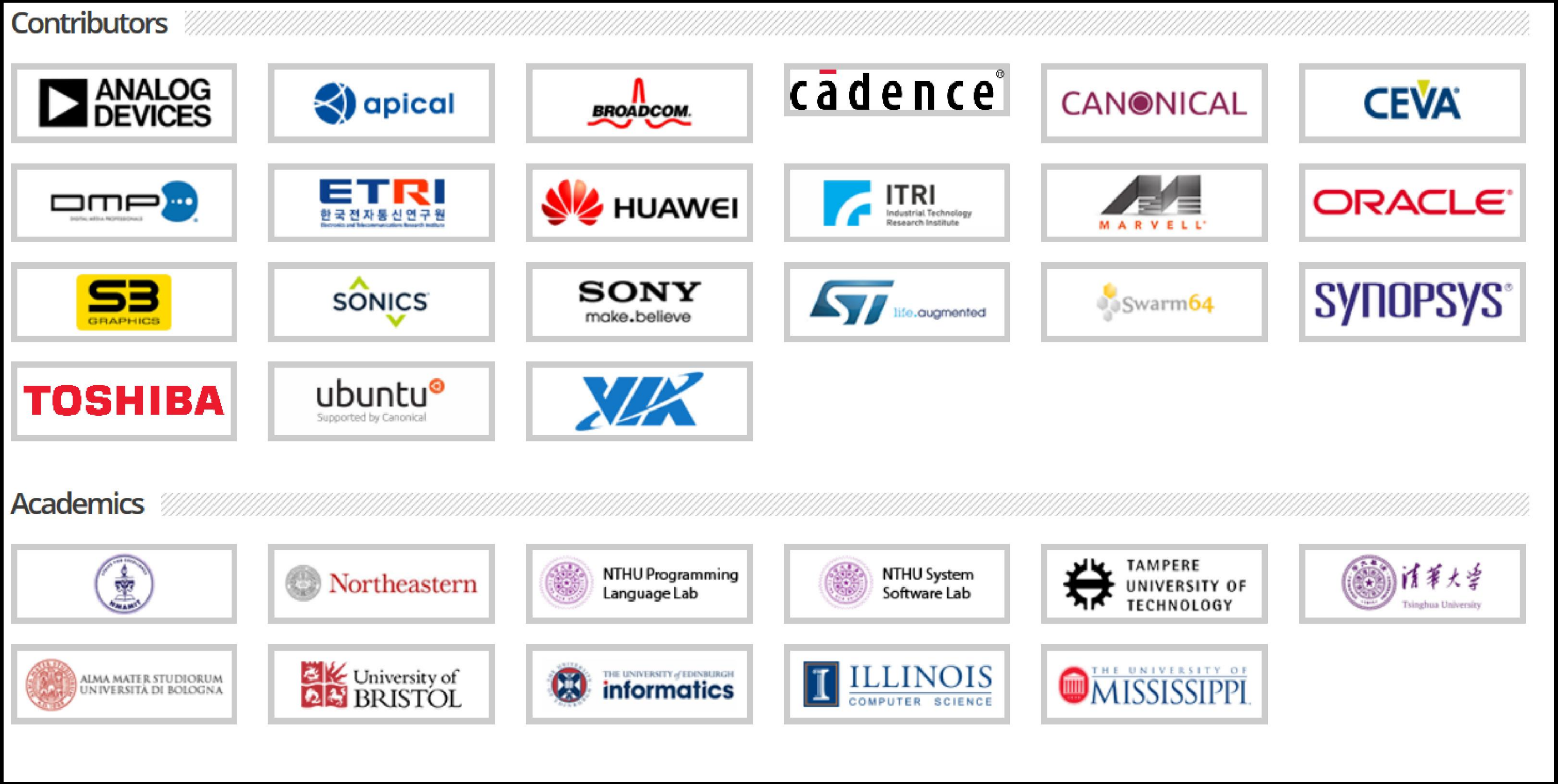
Promoters

- LG Electronics

Supporters

- Argonne National Laboratory
- Arteris
- codeplay®
- FABRICENGINE
- KISHONTI INFORMATICS
- Lawrence Livermore National Laboratory
- Linaro
- Stream Computing Performance Engineers
- mobica
- MULTICORE WARE
- Oak Ridge National Laboratory
- Sandia National Laboratories

HSA MEMBERS



Contributors

- ANALOG DEVICES
- apical
- BROADCOM.
- cadence
- CANONICAL
- CEVA
- DMP
- ETRI 한국전자통신연구원
- HUAWEI
- ITRI Industrial Technology Research Institute
- MARVELL
- ORACLE
- S3 GRAPHICS
- SONICS
- SONY make.believe
- ST life.augmented
- Swarm64
- SYNOPSYS
- TOSHIBA
- ubuntu Supported by Canonical
- VIA

Academics

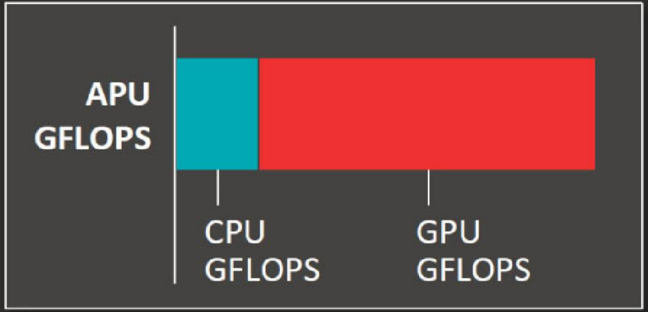
- Northeastern
- NTHU Programming Language Lab
- NTHU System Software Lab
- TAMPERE UNIVERSITY OF TECHNOLOGY
- 清華大學 Tsinghua University
- ALMA MATER STUDIORUM UNIVERSITÀ DI BOLOGNA
- University of BRISTOL
- THE UNIVERSITY of EDINBURGH informatics
- ILLINOIS COMPUTER SCIENCE
- THE UNIVERSITY OF MISSISSIPPI

HSA GOALS

- HSA means Heterogeneous System Architecture
- Defines a set of low level software components (HSAIL, memory management…)
- Manages work queues to the HSA devices (fast enqueue and signaling)
- Gives simple access to CPU and GPU in the APU
  - Single address space for CPU and GPU
- High level languages and compiler target HSA runtime and hardware to expose all the features to all the developers
  - OpenCL 2.0, C++, Fortran, OpenMP 4,….
- http://www.hsafoundation.com/
- https://github.com/hsafoundation
  - OpenMP and C/C++ : https://github.com/HSAFoundation/HSA-OpenMP-GCC-AMD
  - HSA runtime : https://github.com/HSAFoundation/HLC-HSAIL-Development-LLVM
  - HSAIL-LLVM : https://github.com/HSAFoundation/HSA-Runtime-Reference-Source
  - ….

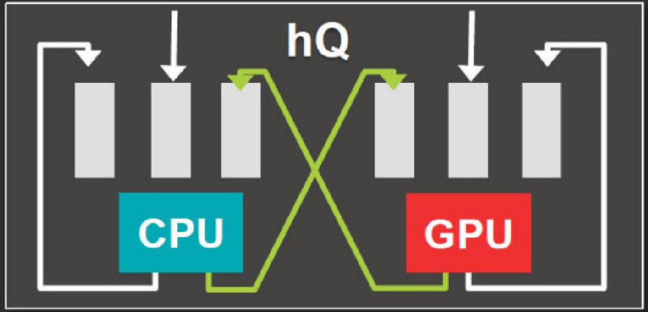
HSA FEATURES ON AMD APUS

UNLOCKING All APU GFLOPS



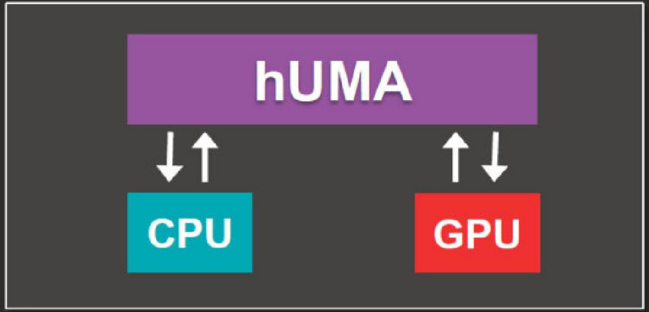
- Access to full potential of APU compute power

ALL CORES EQUAL



- Heterogeneous queuing (hQ) defines how processors interact equally
- GPU and CPU have equal flexibility to create/dispatch work

EQUAL ACCESS TO ENTIRE MEMORY

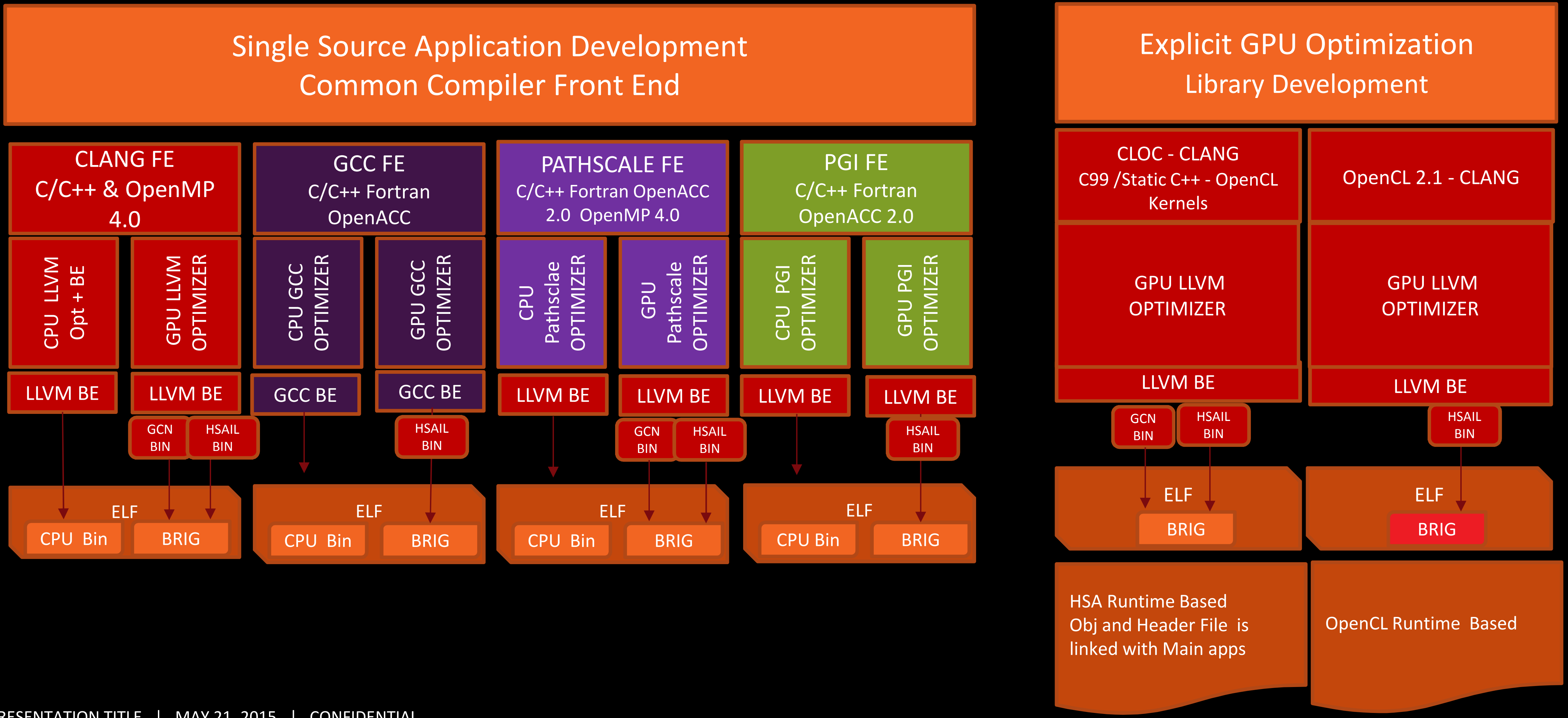


- First time ever: GPU and CPU have uniform visibility into entire memory space

UNLOCKS THE UNMATCHED COMPUTE POTENTIAL AND EFFICIENCY OF APUs

SIMPLIFIED DEVELOPMENT - GIVING YOU CONTROL WHEN YOU NEED IT

PARTNERS WILL LEVERAGE OUR CODE GENERATORS



Single Source Application Development
Common Compiler Front End

CLANG FE
C/C++ & OpenMP 4.0

CPU LLVM Opt + BE

GPU LLVM OPTIMIZER

LLVM BE

LLVM BE

GCN BIN

HSAIL BIN

ELF

CPU Bin

BRIG

GCC FE
C/C++ Fortran
OpenACC

CPU GCC OPTIMIZER

GPU GCC OPTIMIZER

GCC BE

GCC BE

HSAIL BIN

ELF

CPU Bin

BRIG

PATHSCALE FE
C/C++ Fortran OpenACC
2.0 OpenMP 4.0

CPU Pathsclae OPTIMIZER

GPU Pathscale OPTIMIZER

LLVM BE

LLVM BE

GCN BIN

HSAIL BIN

ELF

CPU Bin

BRIG

PGI FE
C/C++ Fortran
OpenACC 2.0

CPU PGI OPTIMIZER

GPU PGI OPTIMIZER

LLVM BE

LLVM BE

HSAIL BIN

ELF

CPU Bin

BRIG

Explicit GPU Optimization
Library Development

CLOC - CLANG
C99 /Static C++ - OpenCL Kernels

GPU LLVM OPTIMIZER

LLVM BE

GCN BIN

HSAIL BIN

ELF

BRIG

HSA Runtime Based Obj and Header File is linked with Main apps

OpenCL 2.1 - CLANG

GPU LLVM OPTIMIZER

LLVM BE

HSAIL BIN

ELF

BRIG

OpenCL Runtime Based

KALMAR-C++

- An Open Source implementation contributed by MulticoreWare
- Based on Clang & LLVM 3.3 and 3.5
- Lower C++ AMP codes to:
  - OpenCL SPIR
  - OpenCL C
  - HSA HSAIL / BRIG
  - x86-64 (CPU fallback)
- Compatible with all major OpenCL stacks, and HSA
- https://bitbucket.org/multicoreware/cppamp-driver-ng/wiki/Home

OPENCL 2.0 MEMORY MANAGEMENT

FINE GRAIN BUFFER SVM

```
//CL_MEM_SVM_FINE_GRAIN_BUFFER means host and device can
//concurrently access the buffer, thus no more data
//transfer….

float* Buffer = (float*)clSVMAlloc(ctx, CL_MEM_READ_WRITE |
                               CL_MEM_SVM_FINE_GRAIN_BUFFER,
                               1024 * sizeof(float), 0);


//fill the buffer from host, no data transfer

for (int i=0; i<1024; i++)

         Buffer[i] = ….;


// use your SVM buffer in you OpenCL kernel on device
directly

clSetKernelArgSVMPointer(my_kernel, 0, Buffer);


clEnqueueNDRangeKernel(queue, my_kernel,…)
```

```
//create device buffer


cl_mem DeviceBuffer =  clCreateBuffer(ctx,
         CL_MEM_READ_WRITE, 1024*sizeof(float), NULL, &err
);


//create host buffer

float* hostBuffer = new float[1024];

for (int i=0; i<1024; i++)

          hostBuffer ] = ….;

//data transfer happens here

clEnqueueWriteBuffer(queue, DeviceBuffer,… , hostBuffer);



//use our device buffer on device

clSetKernelArg(my_kernel,0,sizeof(cl_mem), &DeviceBuffer );


clEnqueueNDRangeKernel(queue, my_kernel,…)
```

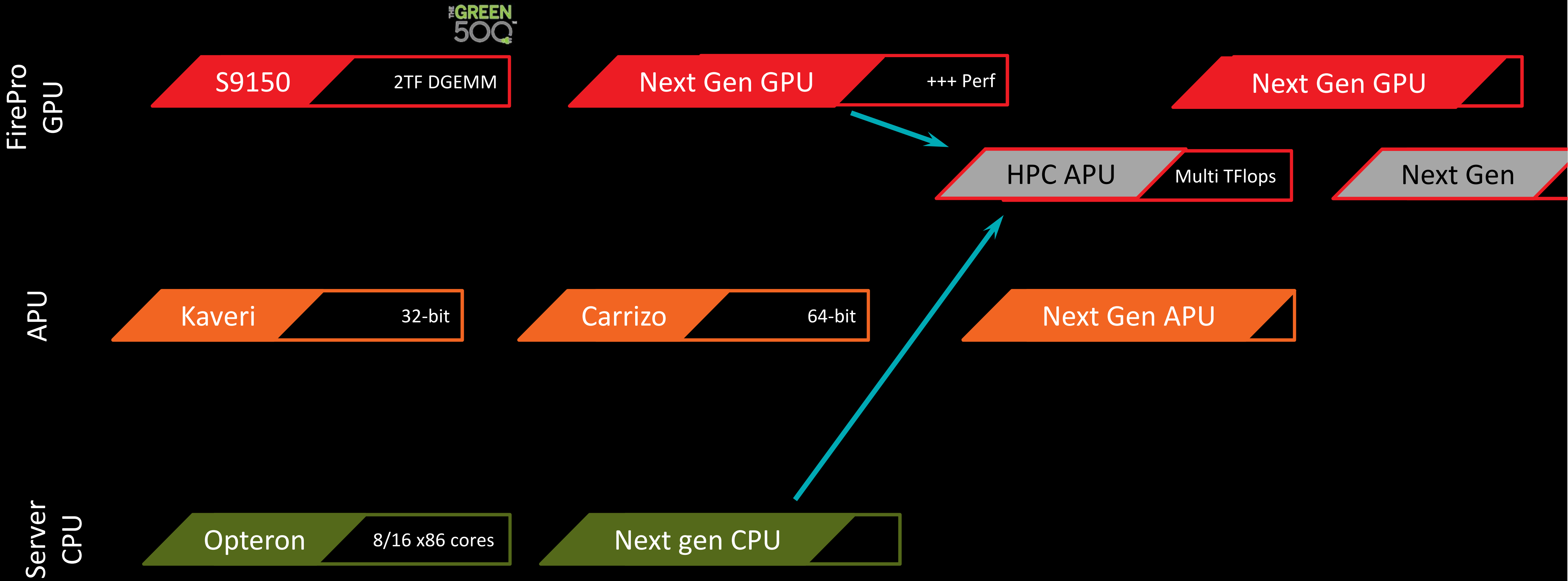
OPENCL 2.0 MEMORY MANAGEMENT

FINE GRAIN SYSTEM

```
//no more OpenCL API needed to allocate data, simply use your favorite memory allocation function : new, malloc…
float* Buffer = (float*)malloc(1024*sizeof(float))

//fill the buffer from host, no data transfer
for (int i=0; i<1024; i++)
          Buffer[i] = ….;

// use your SVM buffer in you OpenCL kernel on device directly
clSetKernelArgSVMPointer(my_kernel, 0, Buffer);

clEnqueueNDRangeKernel(queue, my_kernel,…)
```

AMD HPC Roadmap Trends

AMD



FirePro GPU

S9150 2TF DGEMM

Next Gen GPU +++ Perf

Next Gen GPU

HPC APU Multi TFlops

Next Gen

APU

Kaveri 32-bit

Carrizo 64-bit

Next Gen APU

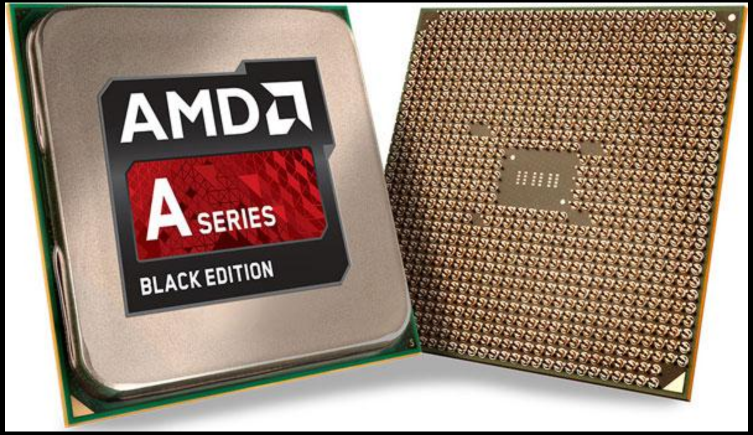
Server CPU

Opteron 8/16 x86 cores

Next gen CPU

AMD HSA HARDWARE

- **Kaveri APU** – 2014

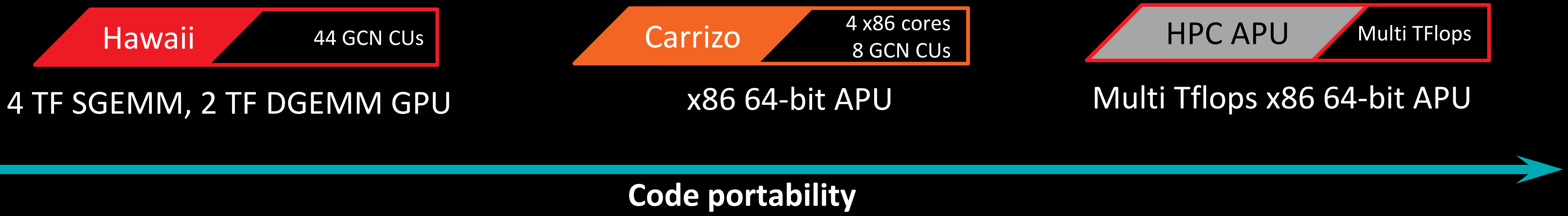


- **Carrizo APU** – 2015
  - http://www.amd.com/en-us/press-releases/Pages/amd-discloses-architecture-2015feb23.aspx
- **“HPC APU”** – beyond 2016
  - Multi-Teraflops APU for servers and workstation

Develop your code now for tomorrow's platforms

AMD

- Now — Best performance with FirePro GPU
- Summer 2015 — AMD Carrizo APU x86 64-bit laptop for code testing



OpenCL HSA FOUNDATION OpenMP

Introducing AMD Compute Toolkit (ACT)

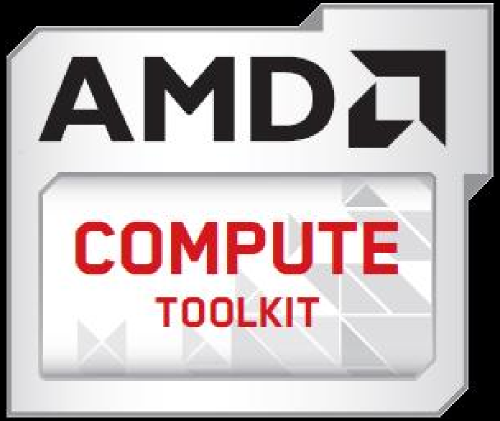
Performance, Power, Productivity

Coming soon

- Comprehensive developer solution providing ultimate performance and power efficiency while enhancing productivity

- ACT includes
  - AMD Compute SDK supporting the latest OpenCL APIs from Khronos
  - AMD CodeXL - a tool suite with leading-edge performance and power profilers, code analyzers and debuggers
  - AMD's accelerated compute libraries
  - Documentation and training to get you up to speed fast
  - Developer support forums to get your questions answered quickly

- Find everything you need in one place with ACT



Delivering full potential from the heterogeneous resources on your platform
http://developer.amd.com/

AMD CodeXL 1.7

- CodeXL helps SW developers get the best performance on AMD platforms
- Debug, Profile and Analyze applications
  - On local and remote hosts
- Power Profiling
- AMD CPUs, GPUs and APUs
- Multiple platforms and Operating Systems
  - Standalone application for Windows® and Linux®
  - Integrated into Microsoft® Visual Studio®
- Free to download and use

Debug

Profile

Build & Analyze

http://developer.amd.com/tools-and-sdks/opencl-zone/codexl/

DISCLAIMER & ATTRIBUTION

The information presented in this document is for informational purposes only and may contain technical inaccuracies, omissions and typographical errors.

The information contained herein is subject to change and may be rendered inaccurate for many reasons, including but not limited to product and roadmap changes, component and motherboard version changes, new model and/or product releases, product differences between differing manufacturers, software changes, BIOS flashes, firmware upgrades, or the like. AMD assumes no obligation to update or otherwise correct or revise this information. However, AMD reserves the right to revise this information and to make changes from time to time to the content hereof without obligation of AMD to notify any person of such revisions or changes.

AMD MAKES NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE CONTENTS HEREOF AND ASSUMES NO RESPONSIBILITY FOR ANY INACCURACIES, ERRORS OR OMISSIONS THAT MAY APPEAR IN THIS INFORMATION.

AMD SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE. IN NO EVENT WILL AMD BE LIABLE TO ANY PERSON FOR ANY DIRECT, INDIRECT, SPECIAL OR OTHER CONSEQUENTIAL DAMAGES ARISING FROM THE USE OF ANY INFORMATION CONTAINED HEREIN, EVEN IF AMD IS EXPRESSLY ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

ATTRIBUTION