Trends in Heterogeneous Systems Architectures (and how they'll affect parallel programming models)

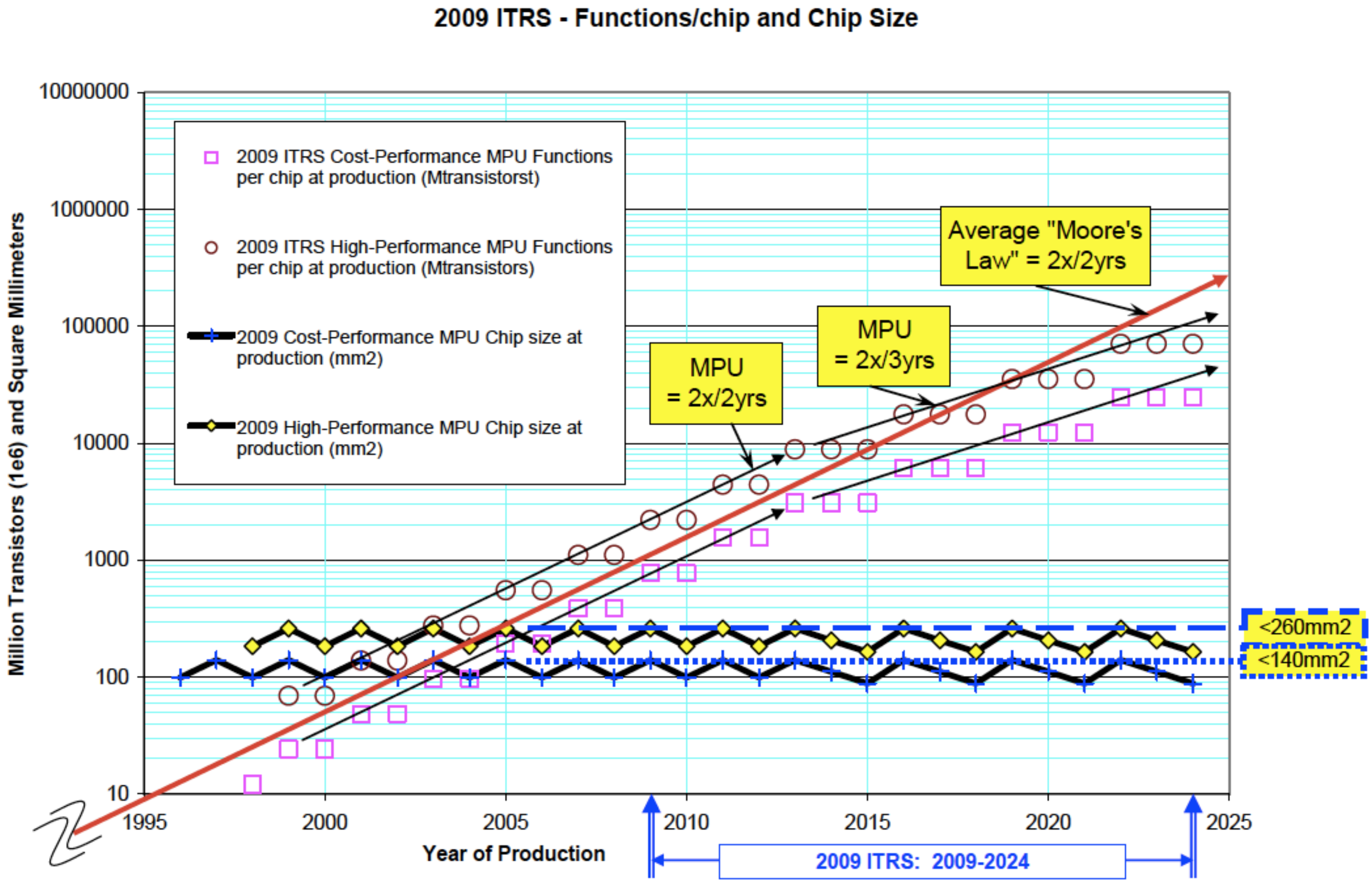
Simon McIntosh-Smith simonm@cs.bris.ac.uk
Head of Microelectronics Research
University of Bristol, UK

University of BRISTOL

Heterogeneous Systems Architectures

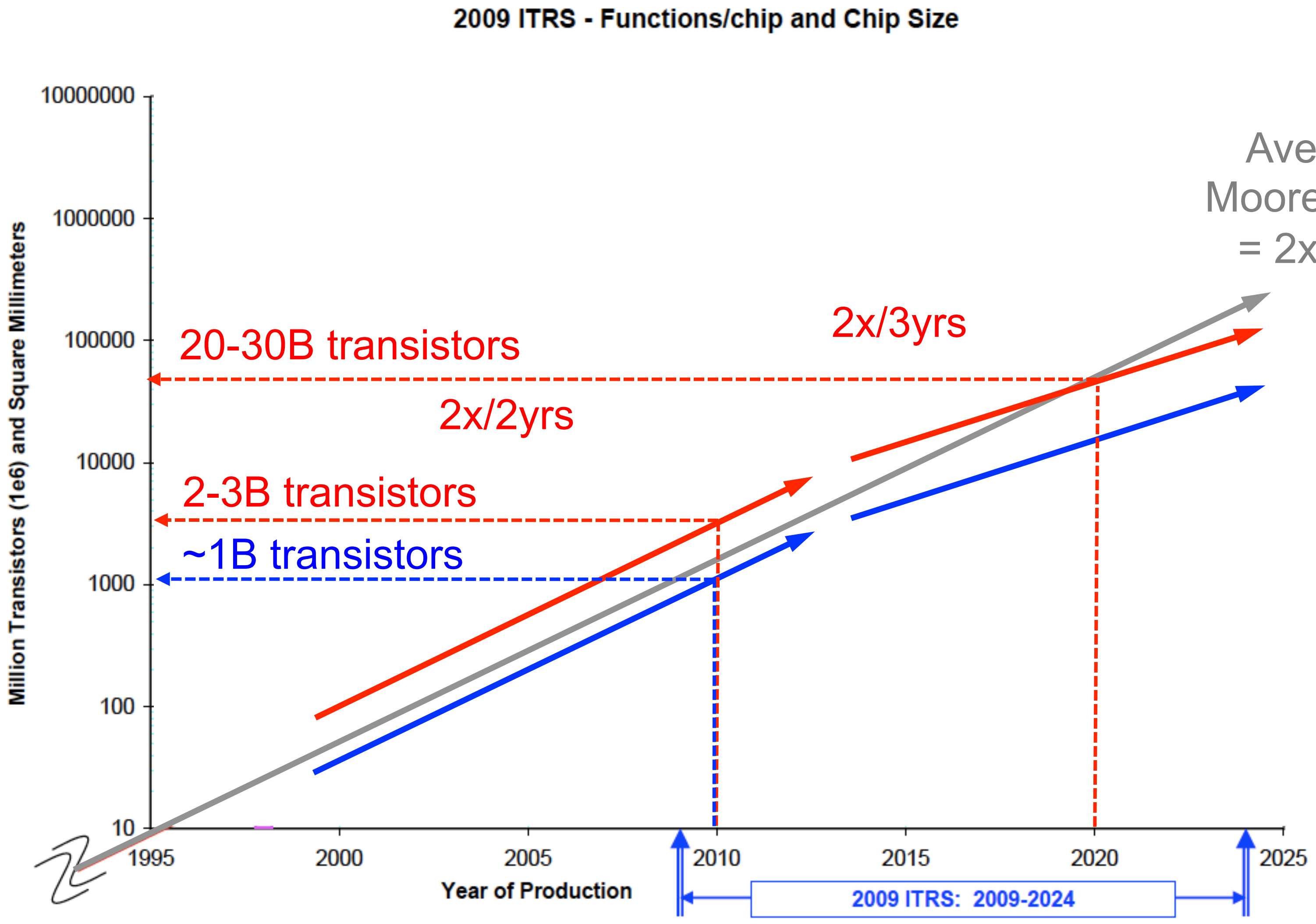
University of BRISTOL

Moore’s Law today



http://www.itrs.net/Links/2009ITRS/2009Chapters_2009Tables/2009_ExecSum.pdf

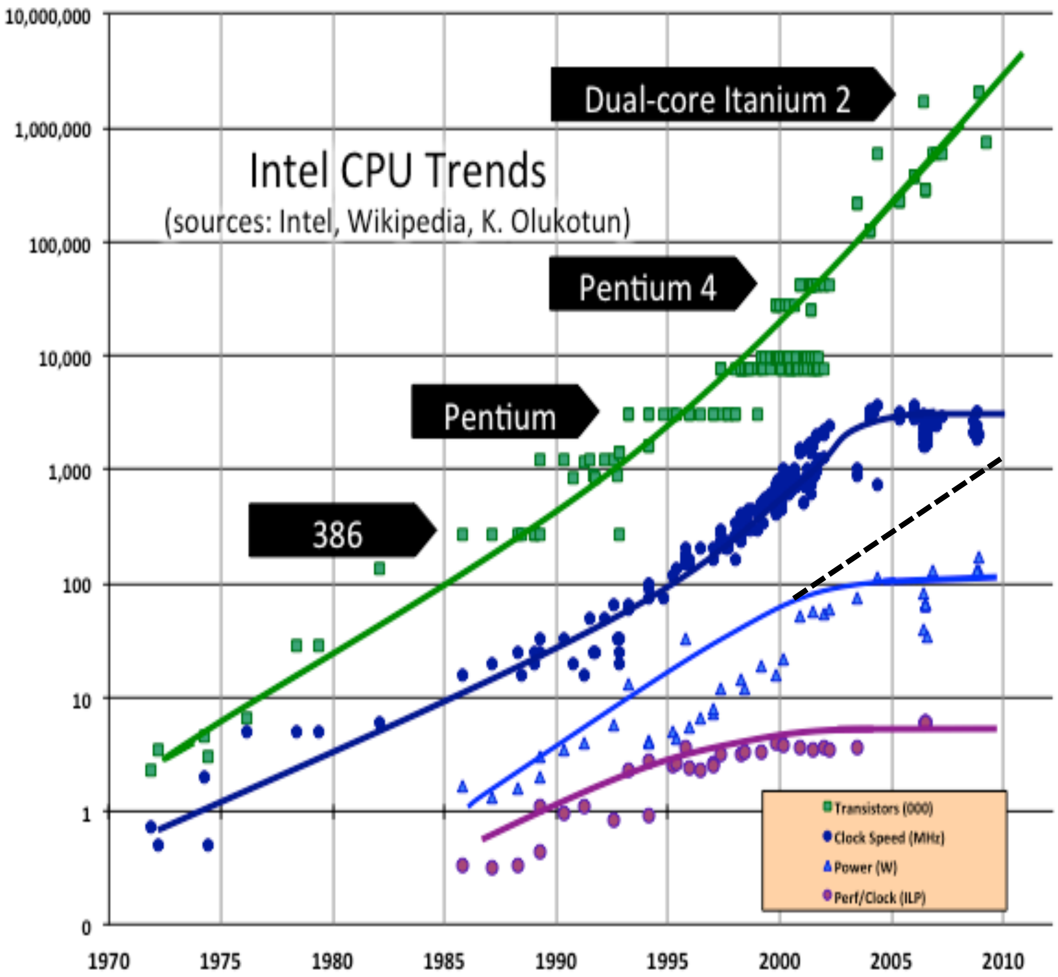
Moore’s Law today



Average Moore’s Law = 2x/2yrs

http://www.itrs.net/Links/2009ITRS/2009Chapters_2009Tables/2009_ExecSum.pdf

Important technology trends



The real Moore’s Law

The clock speed plateau

The power ceiling

Instruction level parallelism limit

Herb Sutter, “The free lunch is over”, Dr. Dobb's Journal, 30(3), March 2005. On-line version, August 2009.
http://www.gotw.ca/publications/concurrency-ddj.htm

Herb Sutter's new outlook

http://herbsutter.com/welcome-to-the-jungle/

In the twilight of Moore's Law, the transitions to multicore processors, GPU computing, and HaaS cloud computing are not separate trends, but aspects of a single trend – mainstream computers from desktops to 'smartphones' are being permanently transformed into heterogeneous supercomputer clusters. Henceforth, a single compute-intensive application will **need to harness different kinds of cores, in immense numbers**, to get its job done.

The free lunch is over. Now welcome to the hardware jungle.

Major hardware trends

The five major hardware trends that will affect Exascale software

1. Heterogeneity
2. Changes to memory hierarchies
3. The impact of fault tolerance
4. Focus on energy efficiency
5. Scale

(For a discussion of 2-5 see "Major hardware trends affecting Exascale developments and their potential impact on software", PRACE- 1IP Work Package 9 Future Technologies Workshop, Daresbury, UK, Apr 11th 2012, http://www.cs.bris.ac.uk/~simonm/publications/sms_prace_exascale_2012.pdf)

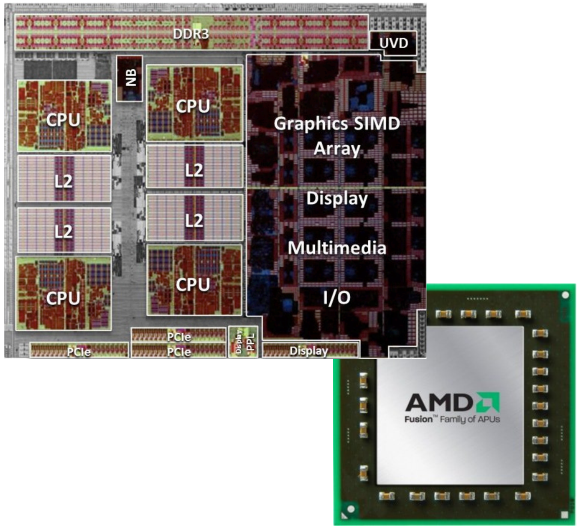
Causes of heterogeneity

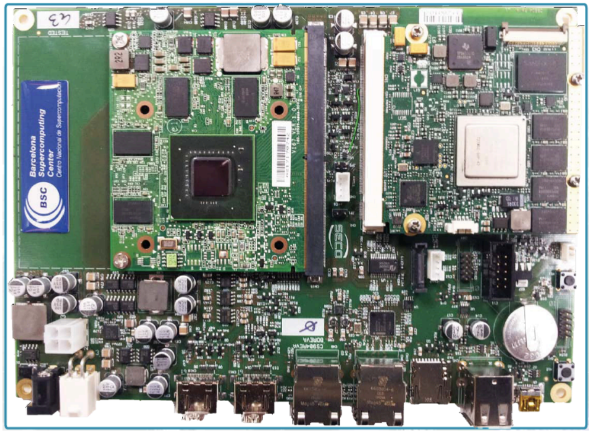
- Multiple types of core
- Interconnect
- Memory type, capacity, …
- Software (OS, middleware, tools, …)

Heterogeneous Systems

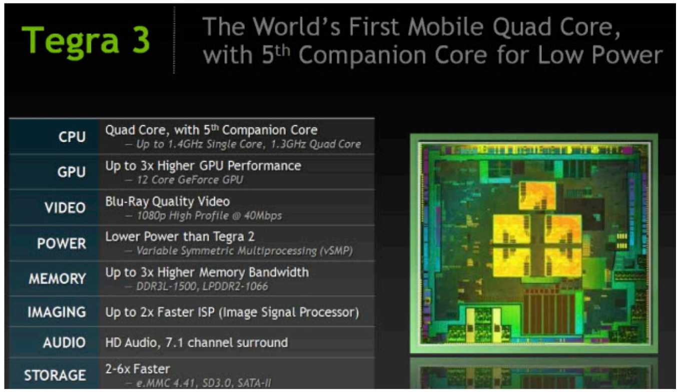
Intel MIC

AMD Llano Fusion APUs





FP7 Mont Blanc ARM + GPU



Tegra 3

The World's First Mobile Quad Core, with 5th Companion Core for Low Power

CPU	Quad Core, with 5th Companion Core – Up to 1.4GHz Single Core, 1.3GHz Quad Core
GPU	Up to 3x Higher GPU Performance – 12 Core GeForce GPU
VIDEO	Blu-Ray Quality Video – 1080p High Profile @ 40Mbps
POWER	Lower Power than Tegra 2 – Variable Symmetric Multiprocessing (vSMP)
MEMORY	Up to 3x Higher Memory Bandwidth – DDR3L-1500, LPDDR2-1066
IMAGING	Up to 2x Faster ISP (Image Signal Processor)
AUDIO	HD Audio, 7.1 channel surround
STORAGE	2-6x Faster – e.MMC 4.41, SD3.0, SATA-II

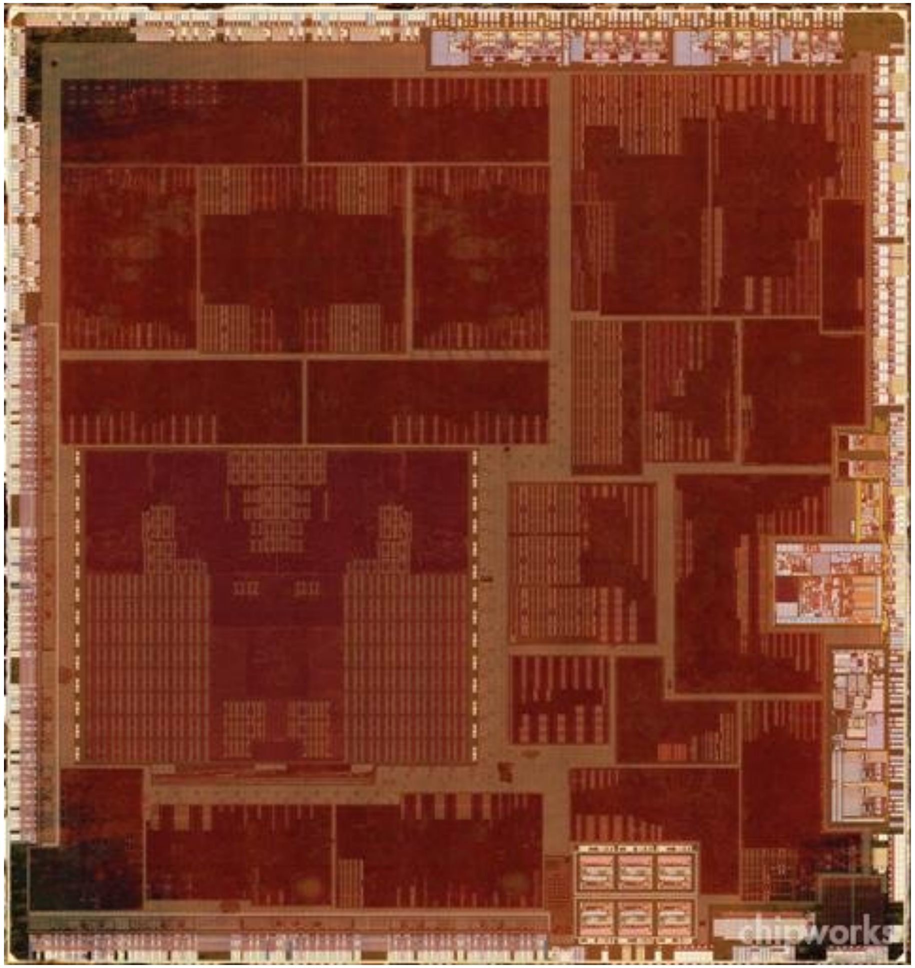
NVIDIA Tegra, Project Denver

Heterogeneity is mainstream

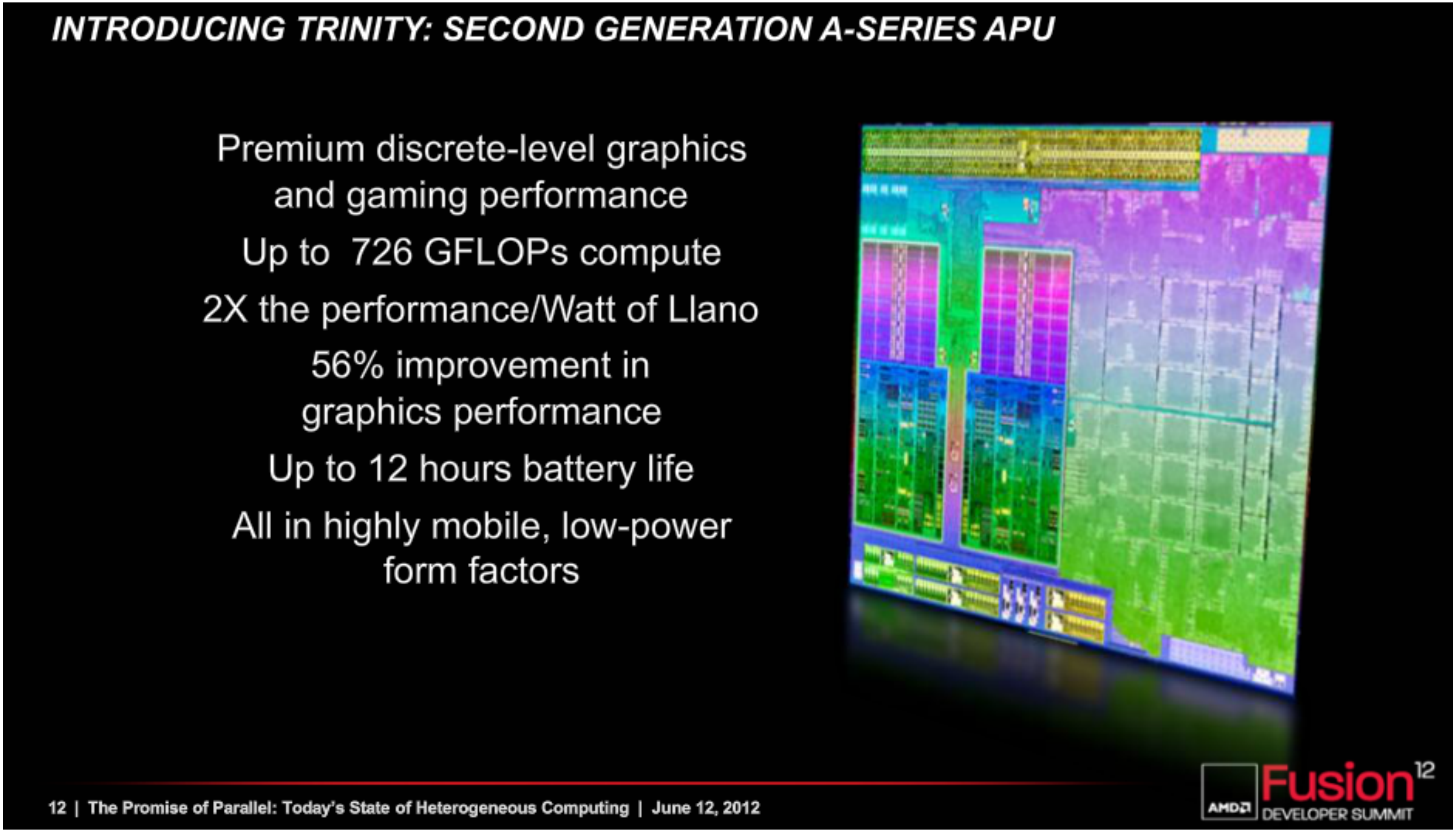
Quad-core ARM Cortex A9 CPU

Quad-core SGX543MP4+ Imagination GPU

You might have one in your pocket



Hot off the press…



INTRODUCING TRINITY: SECOND GENERATION A-SERIES APU

Premium discrete-level graphics and gaming performance

Up to 726 GFLOPs compute

2X the performance/Watt of Llano

56% improvement in graphics performance

Up to 12 hours battery life

All in highly mobile, low-power form factors

12 | The Promise of Parallel: Today's State of Heterogeneous Computing | June 12, 2012

AMD Fusion12 DEVELOPER SUMMIT

Implications

- New programming languages, models, …
- Dynamically adaptive software
  - Discover resources at run-time
- Auto-tuning
- Application frameworks and libraries

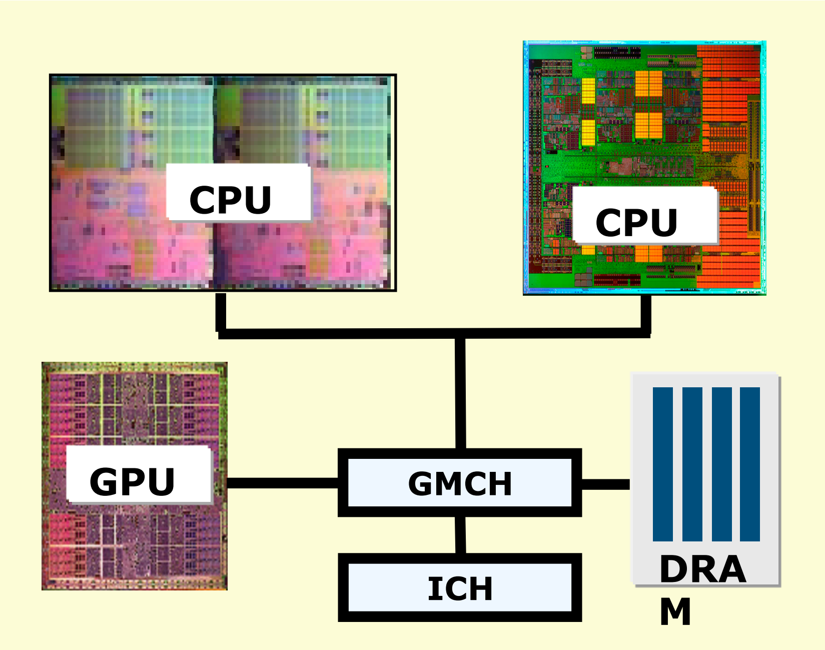
OpenCL
(Open Computing Language)

OpenCL

- Open standard for portable, parallel programming of heterogeneous systems
- Lets programmers write a single **portable** program that uses **all** resources in the heterogeneous platform

A modern system includes:

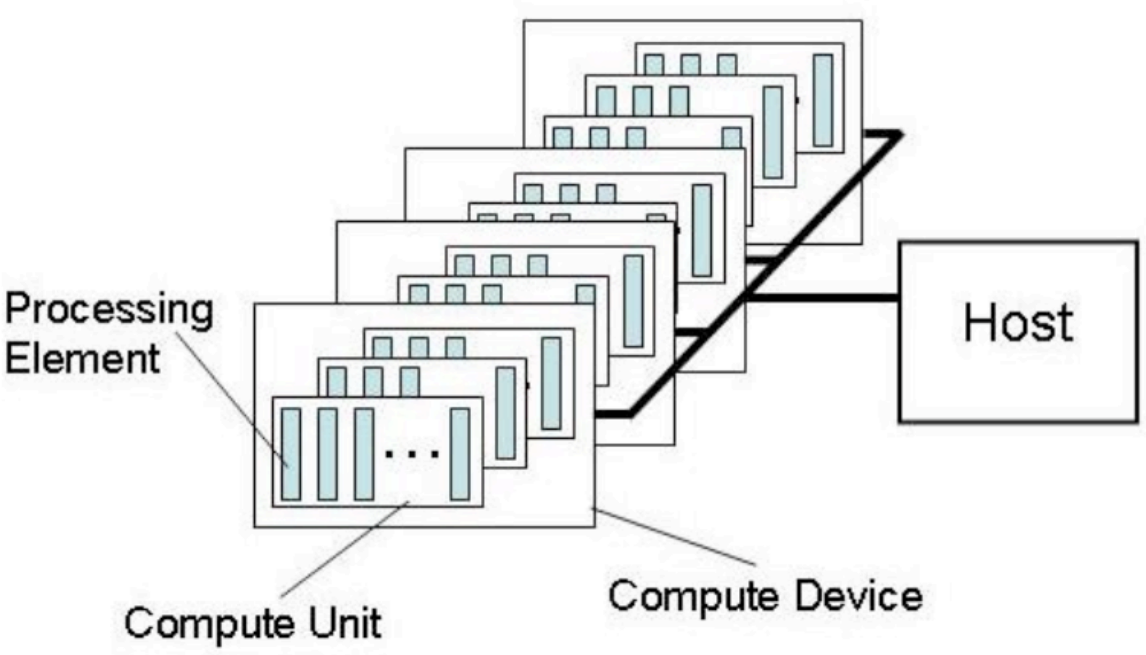
- One or more CPUs
- One or more GPUs
- DSP processors
- …other devices?



GMCH = graphics memory control hub

ICH = Input/output control hub

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

The BIG idea behind OpenCL

- Replace loops with functions (a kernel) executing at each point in a problem domain (index space).
- 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(const 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

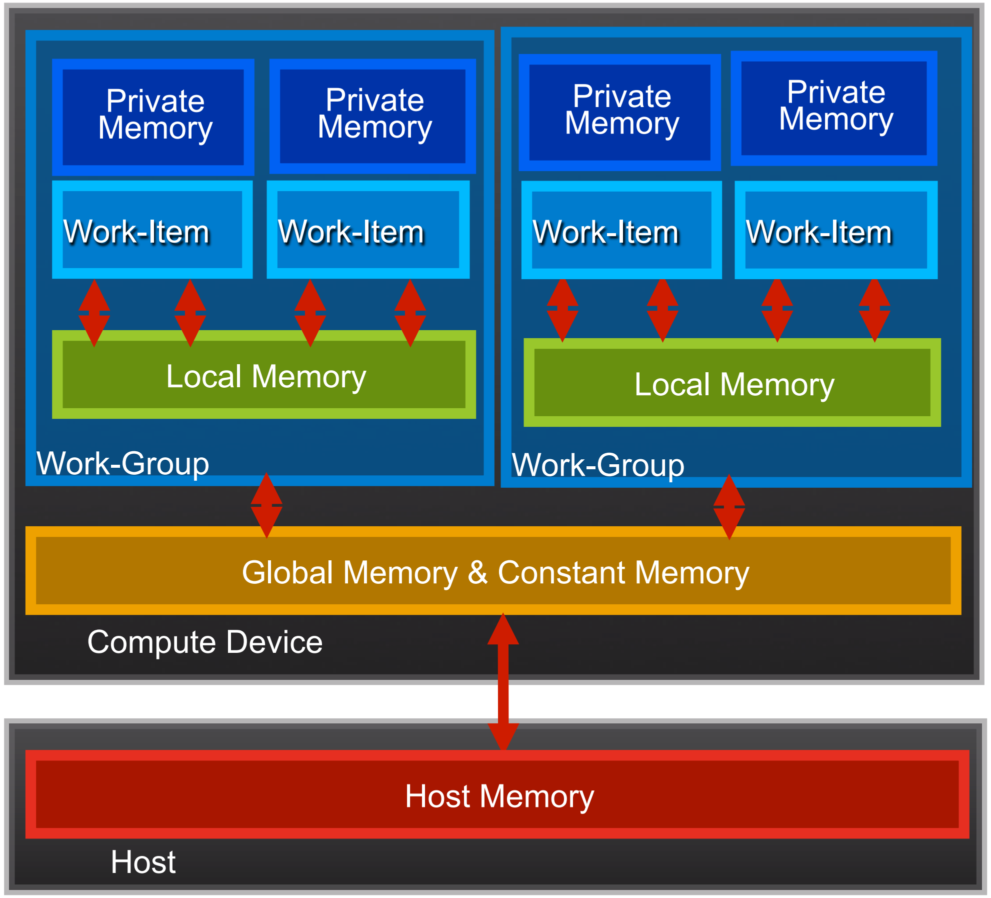
```
kernel void
dp_mul(global const float *a,
       global const float *b,
       global float *c) {
  int id = get_global_id(0);

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

} // execute over “n” work-items
```

OpenCL memory model

- Private Memory
  - Per Work-Item
- Local Memory
  - Shared within a Work-Group
- Global / Constant Memories
  - Visible to all Work-Groups
- Host Memory
  - On the CPU



Memory management is explicit
You must move data from host → global → local *and* back

Issues with OpenCL*

- It does not **compose**
  - Disjoint memory address spaces (local/global)
  - Barriers
- It provides no resource management
  - Kernels are a statically allocated resource

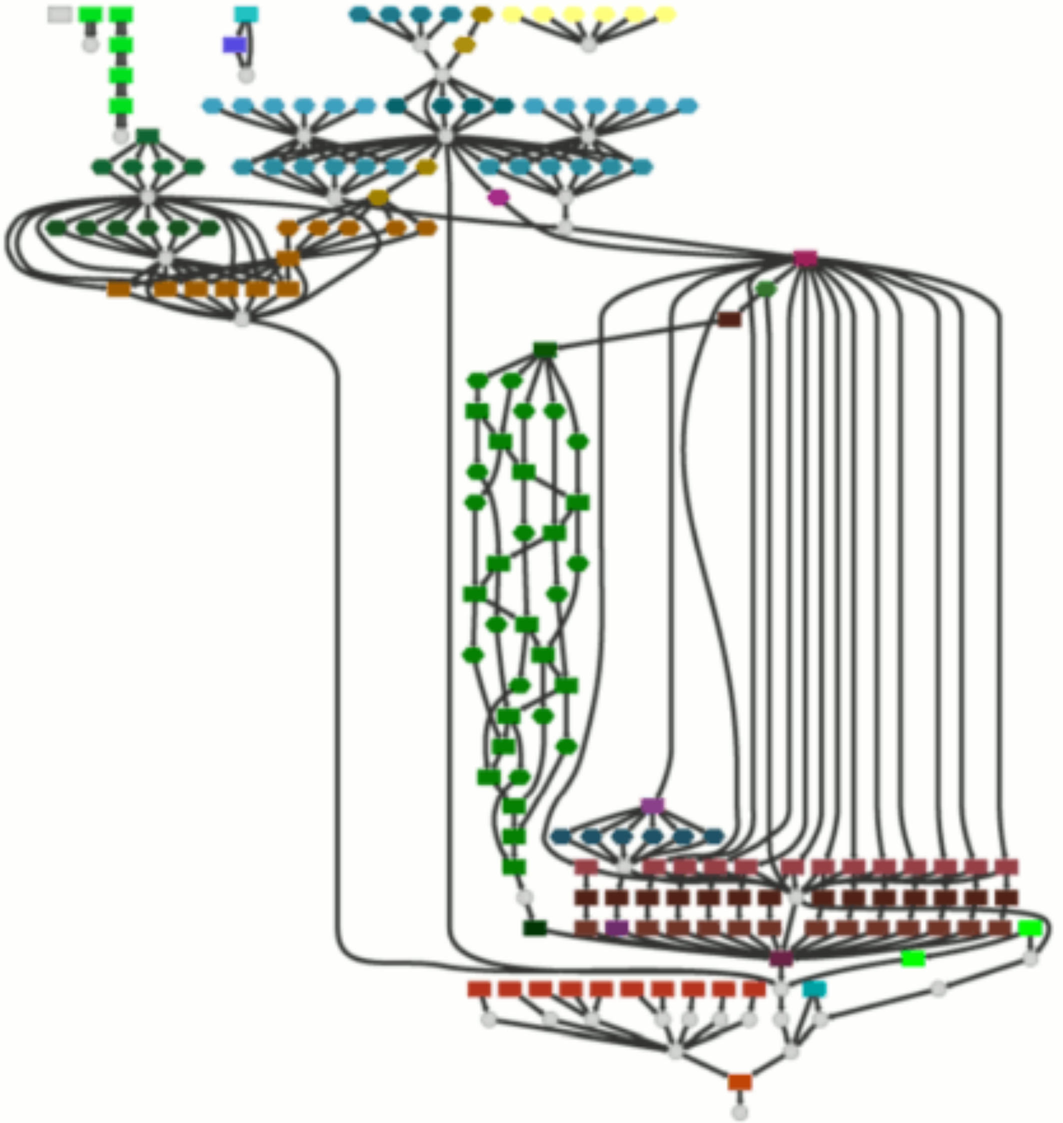
* And most other parallel programming languages

Composability

Need to support many algorithms, even within a single application

Task farms, pipelines, data parallelism, …

See similar composability challenges with OpenMP and parallel libraries, for example



University of BRISTOL

Heterogeneous System Architecture (HSA)

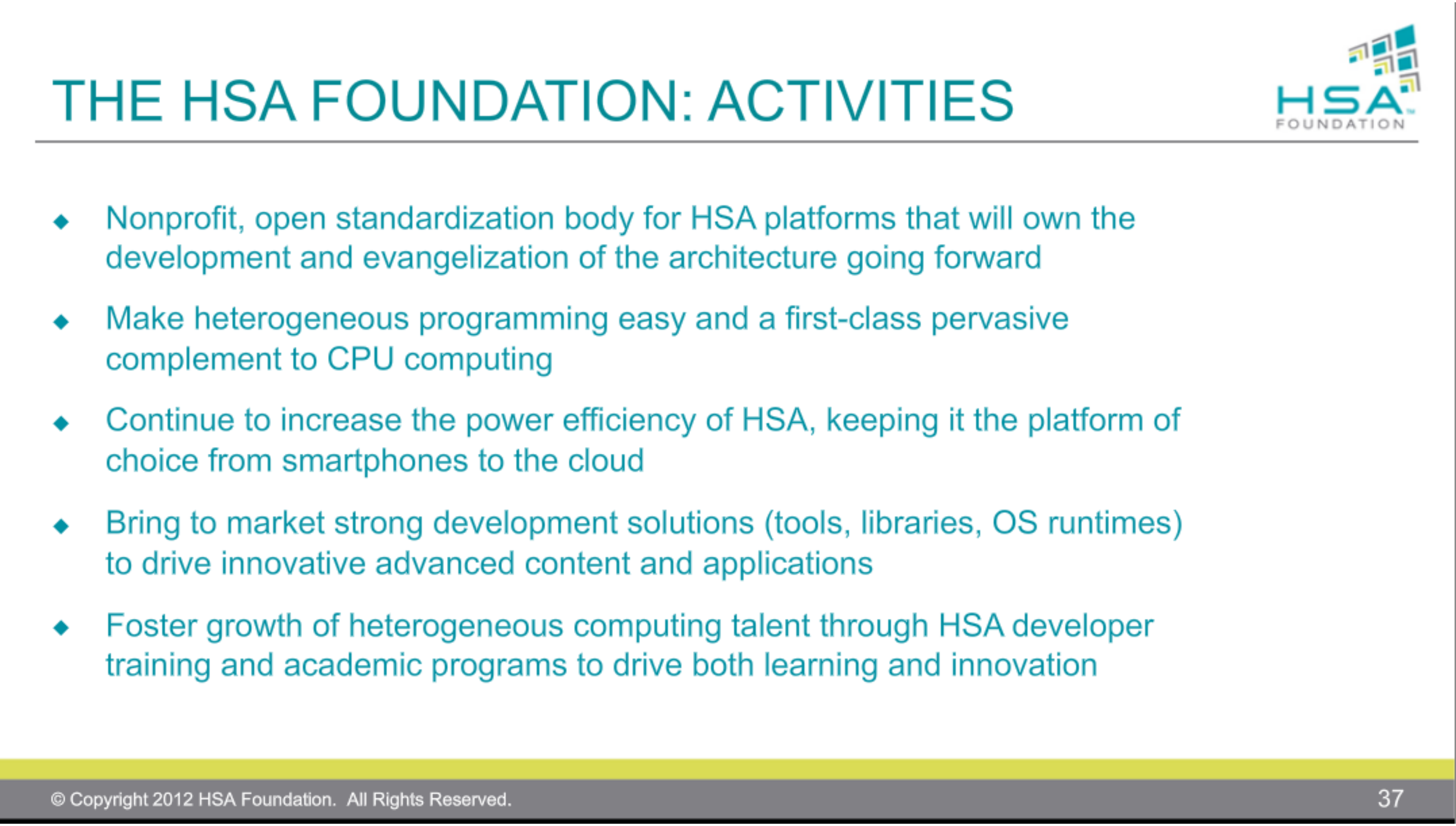
HSA overview

- The HSA Foundation launched this in June 2012 and already includes AMD, ARM, Imagination Technology and Texas Instruments

- HSA is a new, ***open*** architecture specification
  - HSAIL virtual ISA
  - HSA memory model
  - HSA dispatch

- Provides an optimised platform architecture for heterogeneous programming models such as OpenCL, C++AMP, Android’s RenderScript et al

Announced 12/6/2012

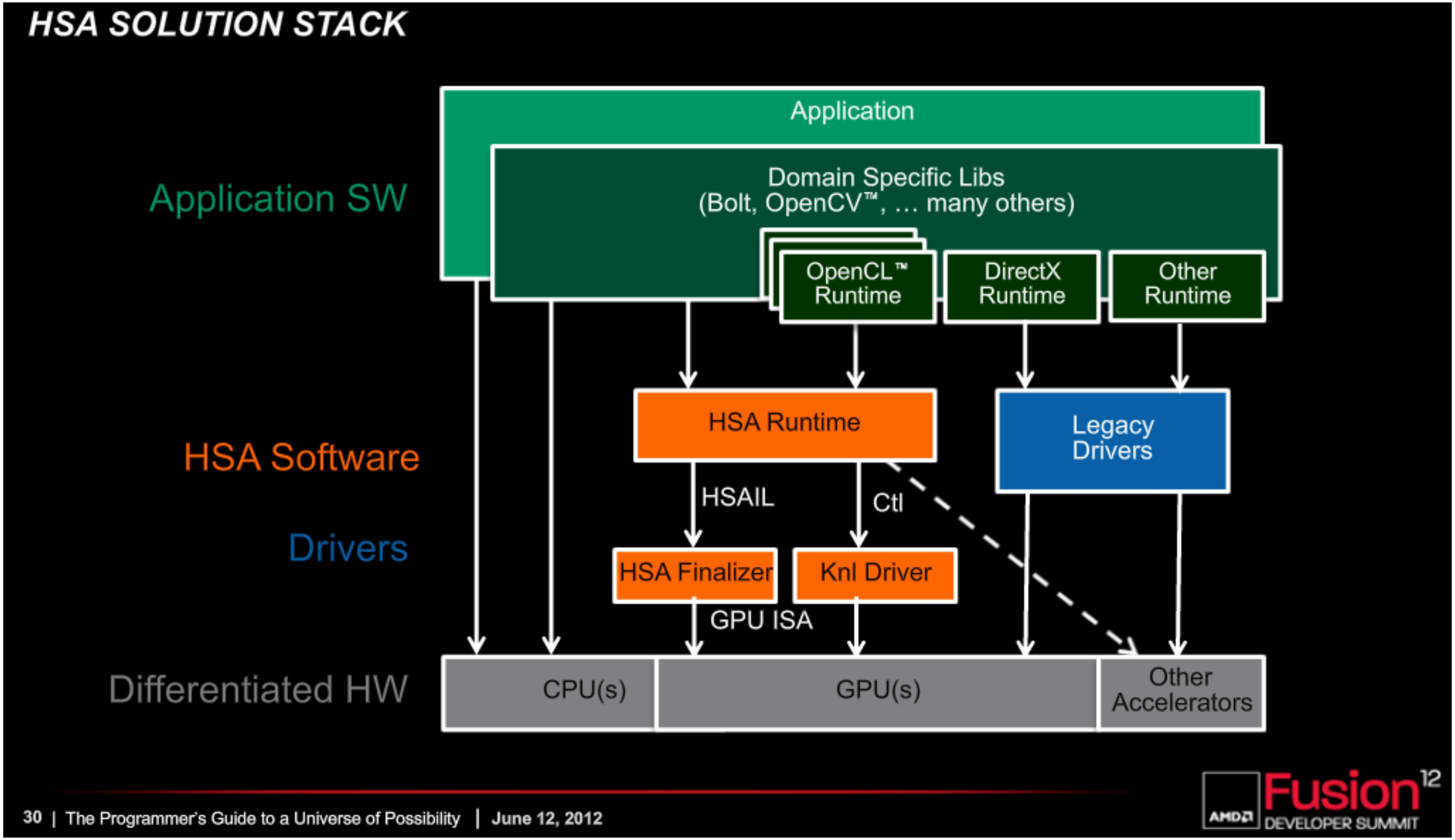


THE HSA FOUNDATION: ACTIVITIES

- Nonprofit, open standardization body for HSA platforms that will own the development and evangelization of the architecture going forward
- Make heterogeneous programming easy and a first-class pervasive complement to CPU computing
- Continue to increase the power efficiency of HSA, keeping it the platform of choice from smartphones to the cloud
- Bring to market strong development solutions (tools, libraries, OS runtimes) to drive innovative advanced content and applications
- Foster growth of heterogeneous computing talent through HSA developer training and academic programs to drive both learning and innovation

http://hsafoundation.com/

HSA overview



HSA SOLUTION STACK

Application SW

HSA Software

Drivers

Differentiated HW

Application

Domain Specific Libs
(Bolt, OpenCV™, … many others)

OpenCL™ Runtime

DirectX Runtime

Other Runtime

HSA Runtime

Legacy Drivers

HSAIL

Ctl

HSA Finalizer

Knl Driver

GPU ISA

CPU(s)

GPU(s)

Other Accelerators

30 | The Programmer's Guide to a Universe of Possibility | June 12, 2012

AMD Fusion12 DEVELOPER SUMMIT

HSA features: simplifying programming

- Much <u>finer grained integration</u> of CPU and GPU cores in silicon
- <u>Unified address space</u> for all cores
- Will support GPU context switching, preemption
- PGAS-style distributed arrays
  - Memory hierarchy abstraction to address function composition
- First class barrier objects
  - Aids composability

HSA Intermediate Layer (HSAIL)

- Virtual ISA for parallel programs
- Similar idea to LLVM's IR – goal is to be a good target for compilers
- *Finalised* to specific ISA by a JIT compiler
- Features:
  - Explicitly parallel
  - Support for exceptions, virtual functions and other high-level features
  - Syscall methods (I/O, printf etc.)
  - Debugging support

HSA memory model

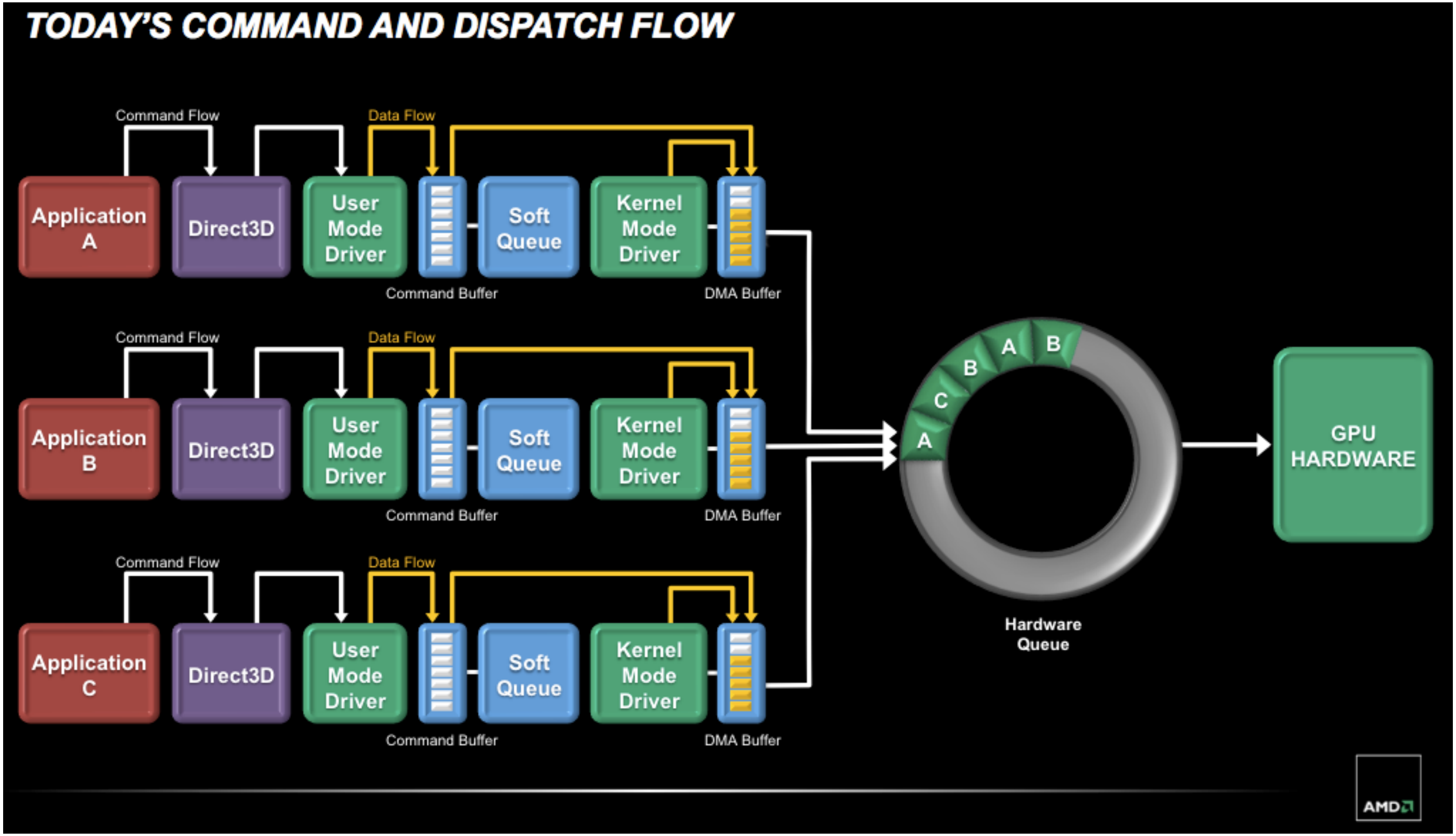
- Compatible with C++11, Java and .NET memory models
- Relaxed consistency
- Designed to support both managed language (such as Java) and unmanaged languages (such as C)
- Will make it much easier to develop 3rd party compilers to target a wide range of heterogeneous products
  - Fortran, C++, C++AMP et al

HSA dispatch

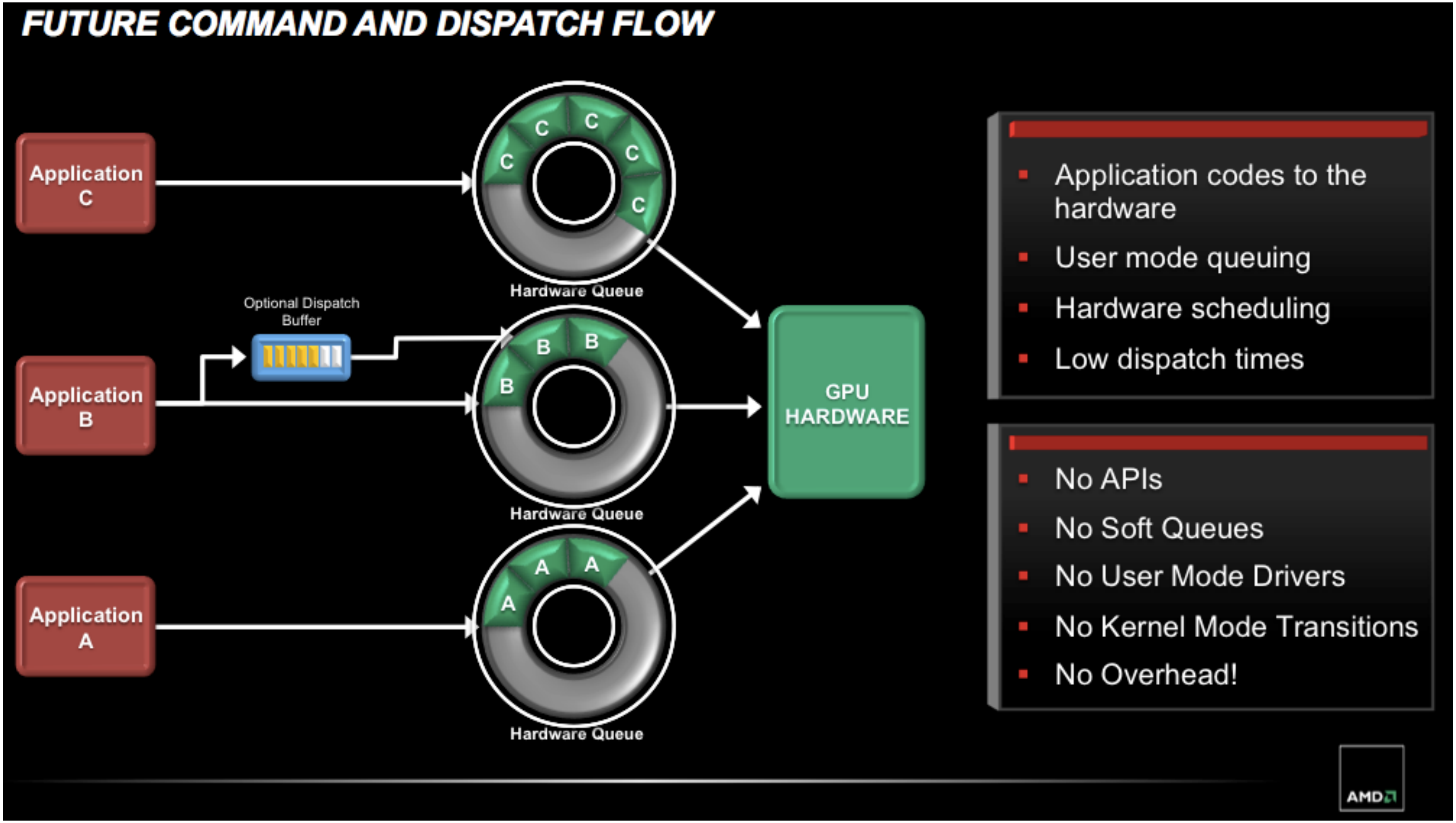
- HSA designed to enable heterogeneous task queuing
  - A work queue per core (CPUs, GPUs et al)
  - Distribution of work into queues
  - Load balancing by work stealing

- Any core can schedule work for any other, including itself
- Significant reduction in overhead of scheduling work for a core

Today's command and dispatch flow



How does HSA improve this?

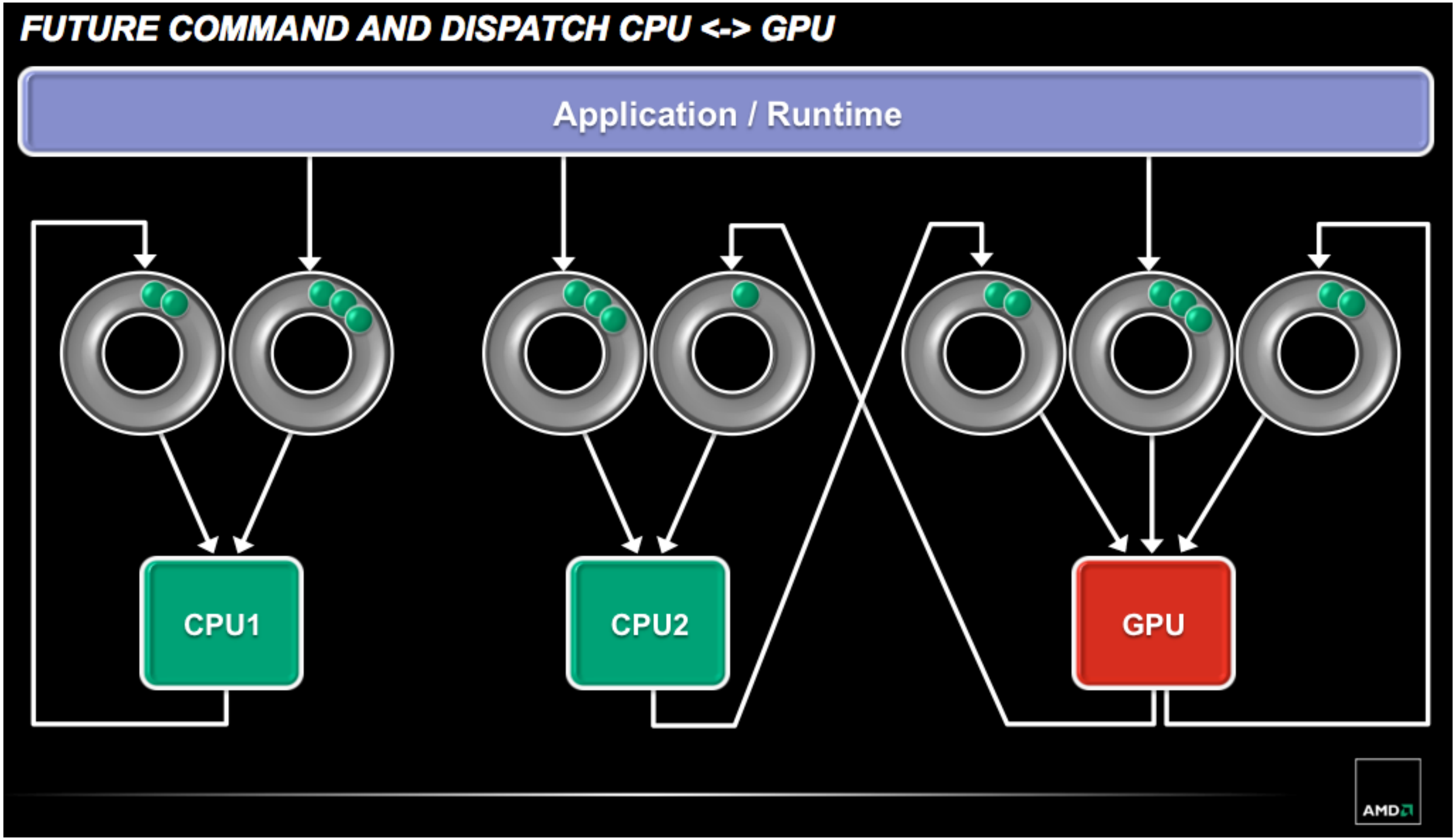


FUTURE COMMAND AND DISPATCH FLOW

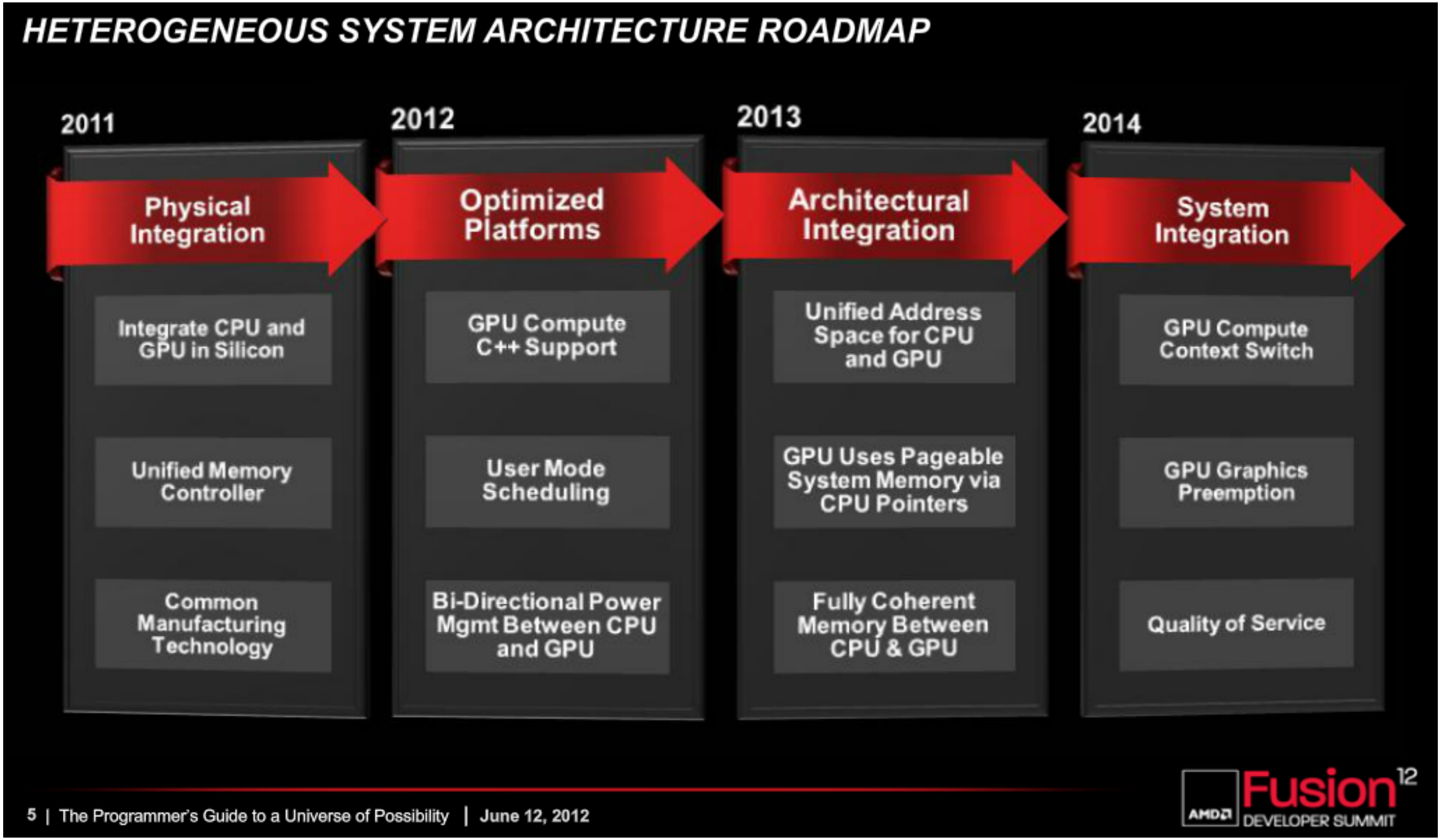
- Application codes to the hardware
- User mode queuing
- Hardware scheduling
- Low dispatch times

- No APIs
- No Soft Queues
- No User Mode Drivers
- No Kernel Mode Transitions
- No Overhead!

Heterogeneous interoperability


FUTURE COMMAND AND DISPATCH CPU <-> GPU

HSA roadmap (from AMD)



HETEROGENEOUS SYSTEM ARCHITECTURE ROADMAP

2011

Physical Integration

- Integrate CPU and GPU in Silicon
- Unified Memory Controller
- Common Manufacturing Technology

2012

Optimized Platforms

- GPU Compute C++ Support
- User Mode Scheduling
- Bi-Directional Power Mgmt Between CPU and GPU

2013

Architectural Integration

- Unified Address Space for CPU and GPU
- GPU Uses Pageable System Memory via CPU Pointers
- Fully Coherent Memory Between CPU & GPU

2014

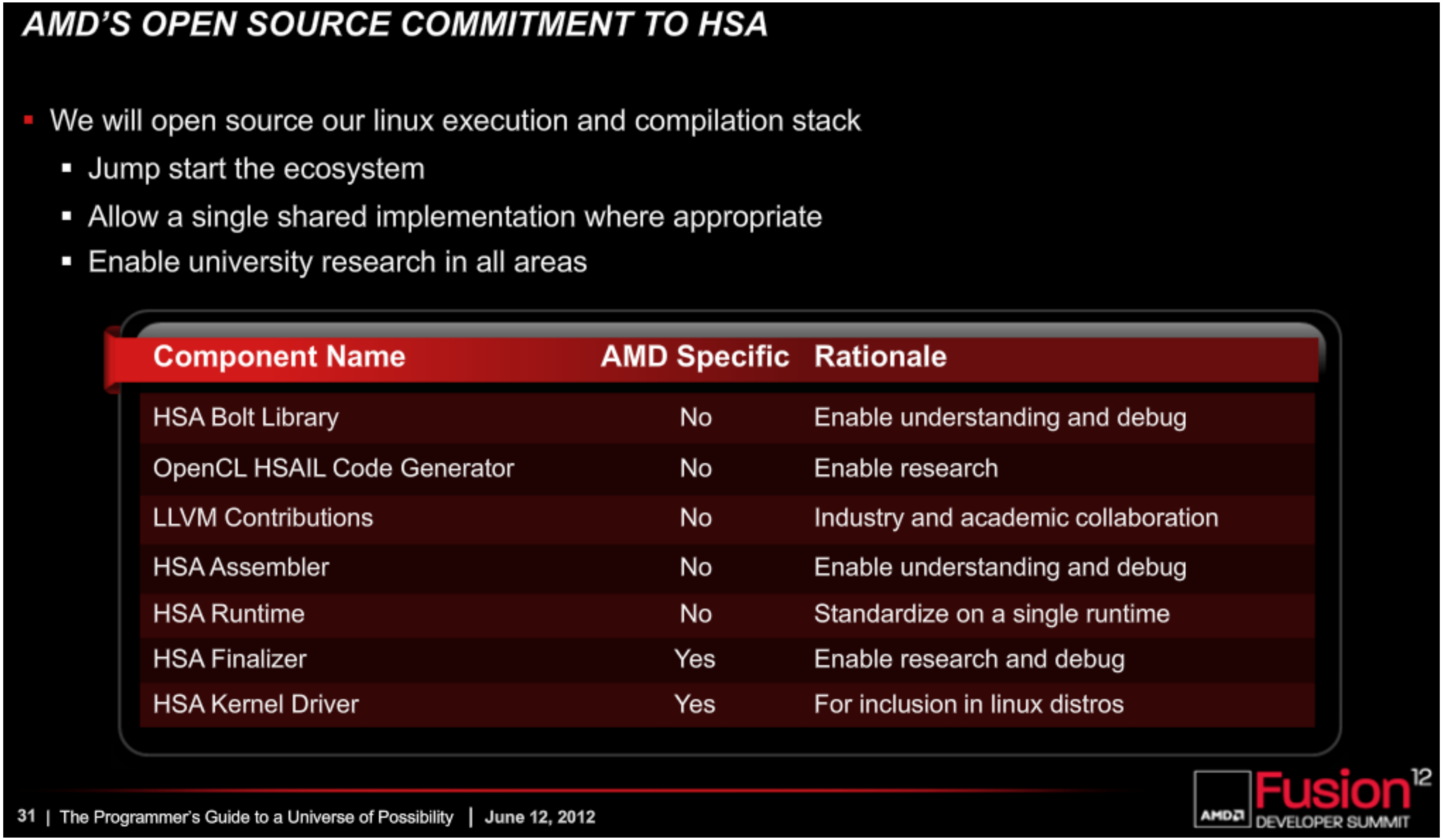
System Integration

- GPU Compute Context Switch
- GPU Graphics Preemption
- Quality of Service

5 | The Programmer's Guide to a Universe of Possibility | June 12, 2012

AMD Fusion12 DEVELOPER SUMMIT

HSA tools released as open source



AMD'S OPEN SOURCE COMMITMENT TO HSA

- We will open source our linux execution and compilation stack
  - Jump start the ecosystem
  - Allow a single shared implementation where appropriate
  - Enable university research in all areas

Component Name	AMD Specific	Rationale
HSA Bolt Library	No	Enable understanding and debug
OpenCL HSAIL Code Generator	No	Enable research
LLVM Contributions	No	Industry and academic collaboration
HSA Assembler	No	Enable understanding and debug
HSA Runtime	No	Standardize on a single runtime
HSA Finalizer	Yes	Enable research and debug
HSA Kernel Driver	Yes	For inclusion in linux distros

31 | The Programmer's Guide to a Universe of Possibility | June 12, 2012

Conclusions

- Heterogeneity is an increasingly important trend
- The market is finally starting to create and adopt the necessary open standards
  - Proprietary models likely to start declining now
  - Don’t get locked into any one vendor!
- Parallel programming models are likely to (re)proliferate
- Exciting times ahead!