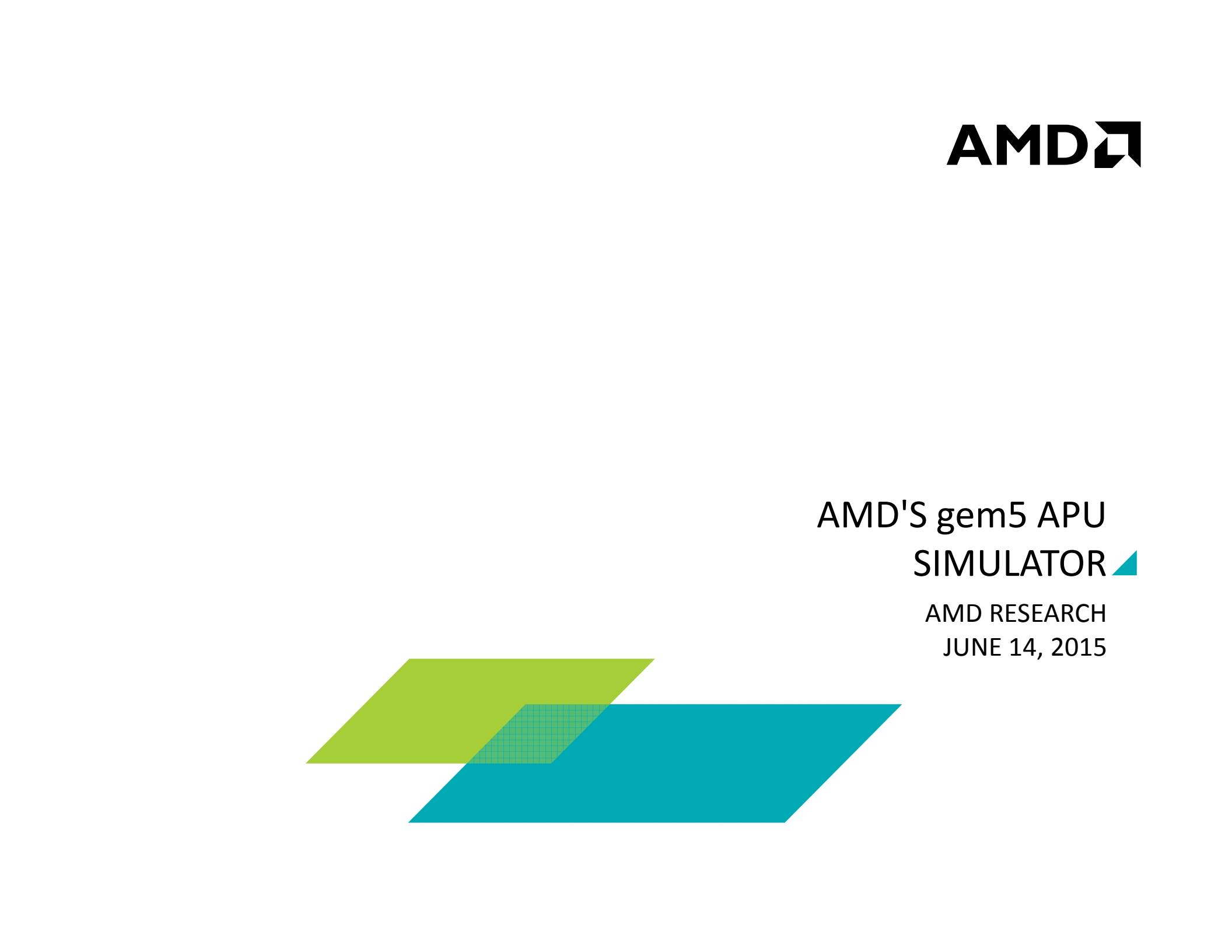

AMD
AMD'S gem5 APU
SIMULATOR
AMD RESEARCH
JUNE 14, 2015

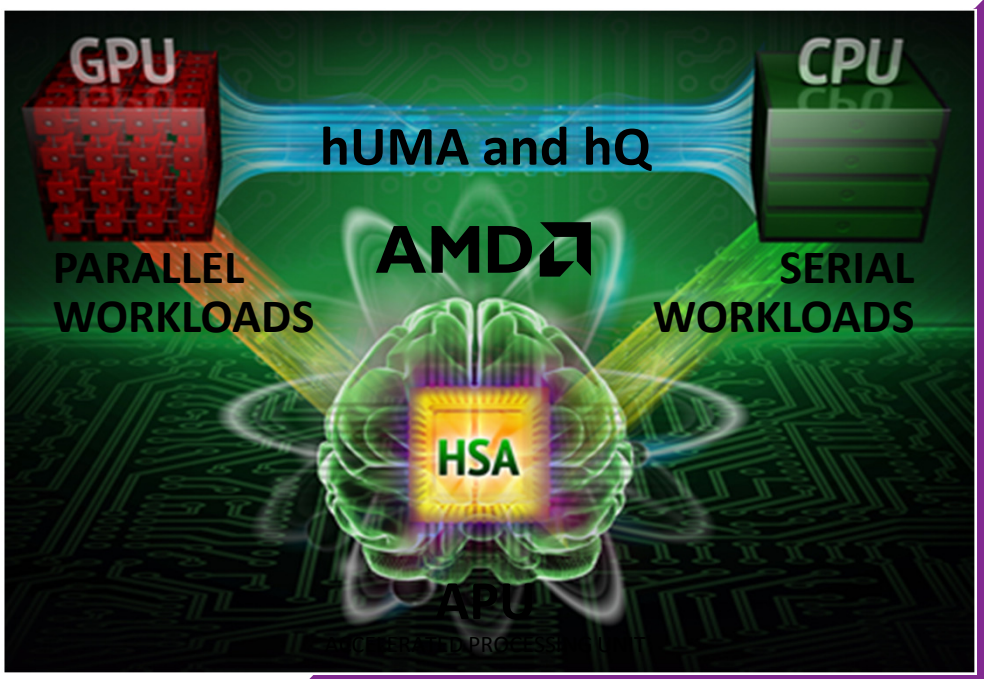
OVERVIEW

AMD

- Introducing AMD's gem5 APU Simulator
  - Extends gem5 with a GPU timing model
  - Supports Heterogeneous System Architecture in SE mode
  - Includes several Ruby-based APU memory systems
- This talk
  - Heterogeneous System Architecture
  - Execution flow
  - Release schedule
- Future plans

WHAT IS HSA?

HETEROGENEOUS SYSTEM ARCHITECTURE



Processor design that makes it easy to harness the **entire computing power of an APU** for faster and more power-efficient devices, including personal computers, tablets, smartphones, and cloud servers

HSA BUILDING BLOCKS

http://hsafoundation.com

http://github.com/HSAFoundation

HSA Hardware Building Blocks

- Shared Virtual Memory
  - Single address space
  - Coherent
  - Pageable
  - Fast access from all components
  - Can share pointers
- Architected User-Level Queues
- Signals
- Context Switching
- Platform Atomics
- Defined Memory Model

Open-Source

HSA Platform System Arch Specification

Industry standard, architected requirements for how devices share memory and communicate with each other

HSA Software Building Blocks

- HSAIL
  - Portable, parallel, compiler IR
  - Instruction definition

Open-Source

HSA Programmer's Reference Manual

- HSA Runtime
  - Create queues
  - Allocate memory
  - Device discovery

Open-Source

HSA System Runtime Specification

- Multiple high level compilers
  - CLANG/LLVM/HSAIL
  - C++, OpenMP, OpenACC, Python

Open-Source

Industry standard compiler IR and runtime to enable existing programming languages to target the GPU

APU SIMULATION SUPPORT

AMD

HSA Hardware Building Blocks

- Shared Virtual Memory
  - Single address space
  - Coherent
  - Pageable
  - Fast access from all components
  - Can share pointers
- Architected User-Level Queues
- Signals
- Context Switching
- Platform Atomics
- Defined Memory Model

HSA Software Building Blocks

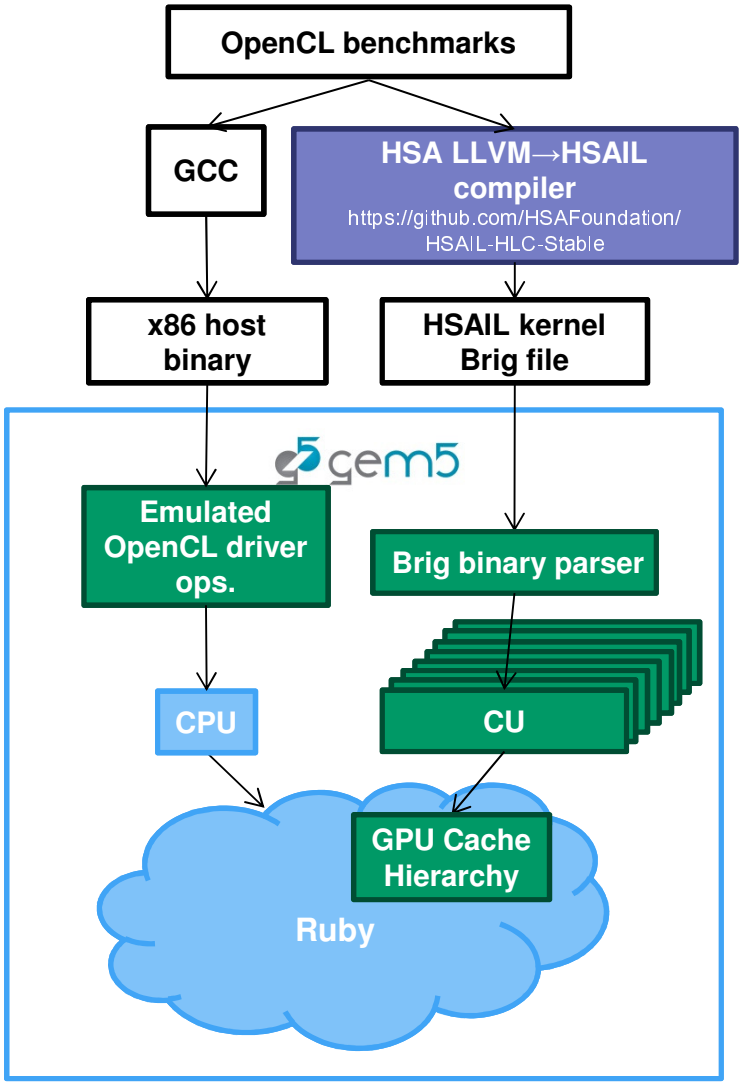
- HSAIL
  - Portable, parallel, compiler IR
  - Instruction definition
- HSA Runtime (OpenCL™ Runtime)
  - Create queues
  - Allocate memory
  - Device discovery
- Multiple high-level compilers
  - CLANG/LLVM/HSAIL
  - C++, OpenMP, OpenACC, Python

Legend
Included in first release
Work-in-progress / may be released
Longer term work

APU SIMULATION FLOW

AMD

- GPGPU flow based on **gem5**
  - **w/ AMD added components**
  - Supports HSAIL/BRIG kernels
- System emulation simulation
  - No OS or device driver
  - Execution-driven evaluation
    - GPU directly executes HSAIL
    - CPU executes x86
- Details
  - GCN GPU model*
  - Multiple GPU cache hierarchies
    - Write-through
    - Read-for-ownership



[*] AMD Graphics Core Next (GCN) Architecture, White Paper. http://www.amd.com/Documents/GCN_Architecture_whitepaper.pdf. June 2012.

RELEASE SCHEDULE

AMD

- First patches posted for review on May 11
- Posting all patches by the end of October

- We appreciate the active discussion and reviews
  - We made many, many modifications based on reviewer feedback
  - Please keep the “cost/benefit” analysis in mind when asking for changes

AMD

Questions?

DISCLAIMER & ATTRIBUTION

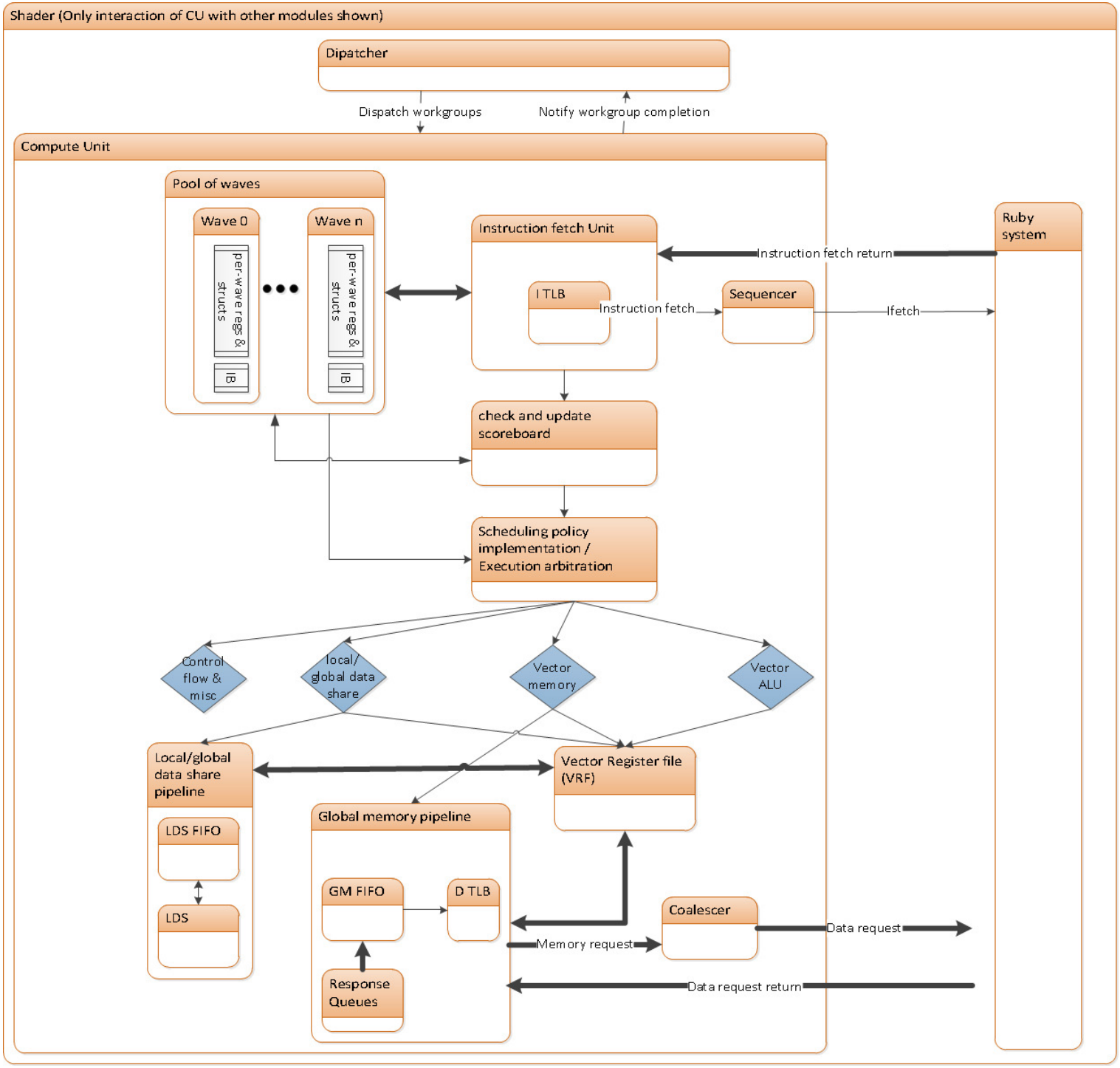
The information presented in this document is for informational purposes only and may contain technical inaccuracies, omissions and typographical errors.

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


Shader (Only interaction of CU with other modules shown)
Dipatcher
Dispatch workgroups
Notify workgroup completion
Compute Unit
Pool of waves
Wave 0
Wave n
per-wave regs & structs
IB
Instruction fetch Unit
I TLB
Instruction fetch
Sequencer
Ifetch
Instruction fetch return
Ruby system
check and update scoreboard
Scheduling policy implementation / Execution arbitration
Control flow & misc
local/ global data share
Vector memory
Vector ALU
Local/global data share pipeline
LDS FIFO
LDS
Vector Register file (VRF)
Global memory pipeline
GM FIFO
D TLB
Response Queues
Coalescer
Memory request
Data request
Data request return