FLOATING-POINT ARITHMETIC IN AMD PROCESSORS

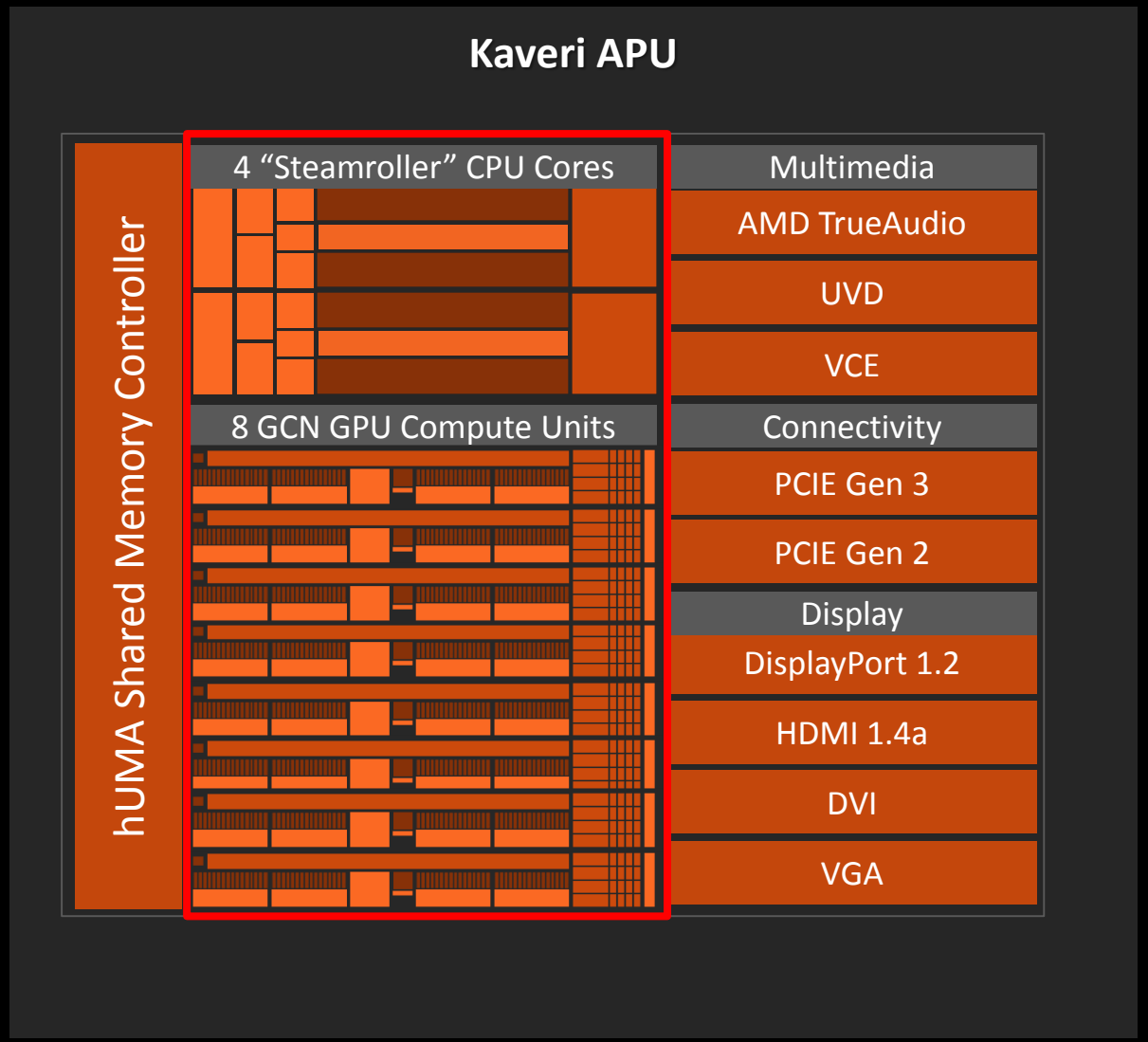
MICHAEL SCHULTE
AMD RESEARCH

JUNE 2015

AGENDA

- The Kaveri Accelerated Processing Unit (APU)
  - The Graphics Core Next Architecture and its Floating-Point Arithmetic
  - The Steamroller x86 CPU and its Floating-Point Arithmetic
- Summary and Future Directions

THE KAVERI ACCELERATED PROCESSING UNIT (APU)



EXCELLENT COMPUTE PERFORMANCE

- 4 "Steamroller" CPU cores
- 8 Graphics Core Next (GCN) GPU compute units
- Heterogeneous System Architecture (HSA) enabled

ENHANCED MULTIMEDIA EXPERIENCES

- Video acceleration
- AMD TrueAudio
- 4 display heads

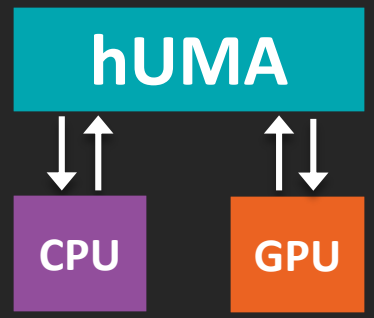
HIGH PERFORMANCE CONNECTIVITY

- 128 bits DDR3 up to 2133 MHz
- PCI-Express Gen3 x16 for discrete graphics upgrade
- PCI-Express for direct attach NVM SSD

THE KAVERI APU AND HSA

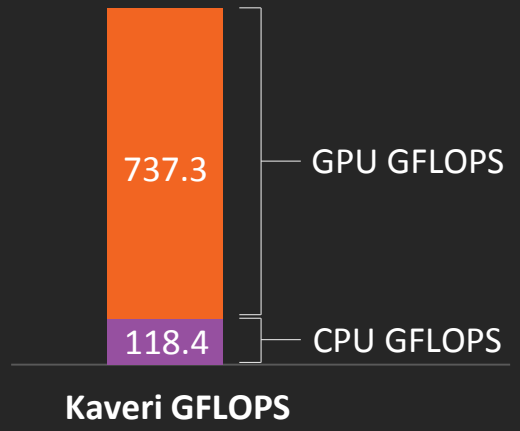
AMD

EQUAL ACCESS TO ENTIRE MEMORY



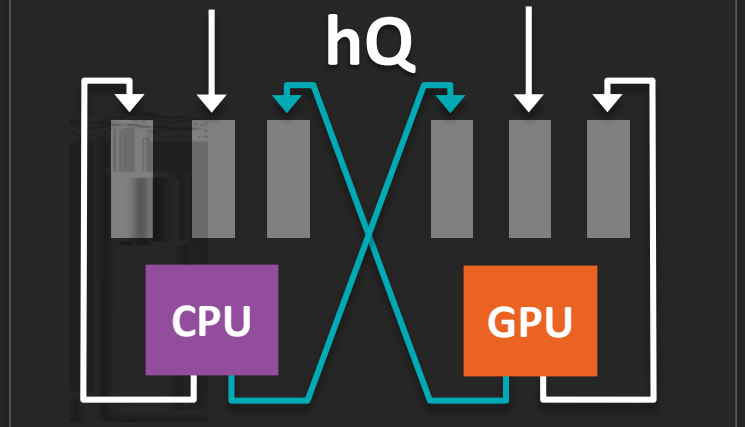
- GPU and CPU have uniform visibility into entire memory system

APU GFLOPS



- Programs have access to all of Kaveri APU compute power

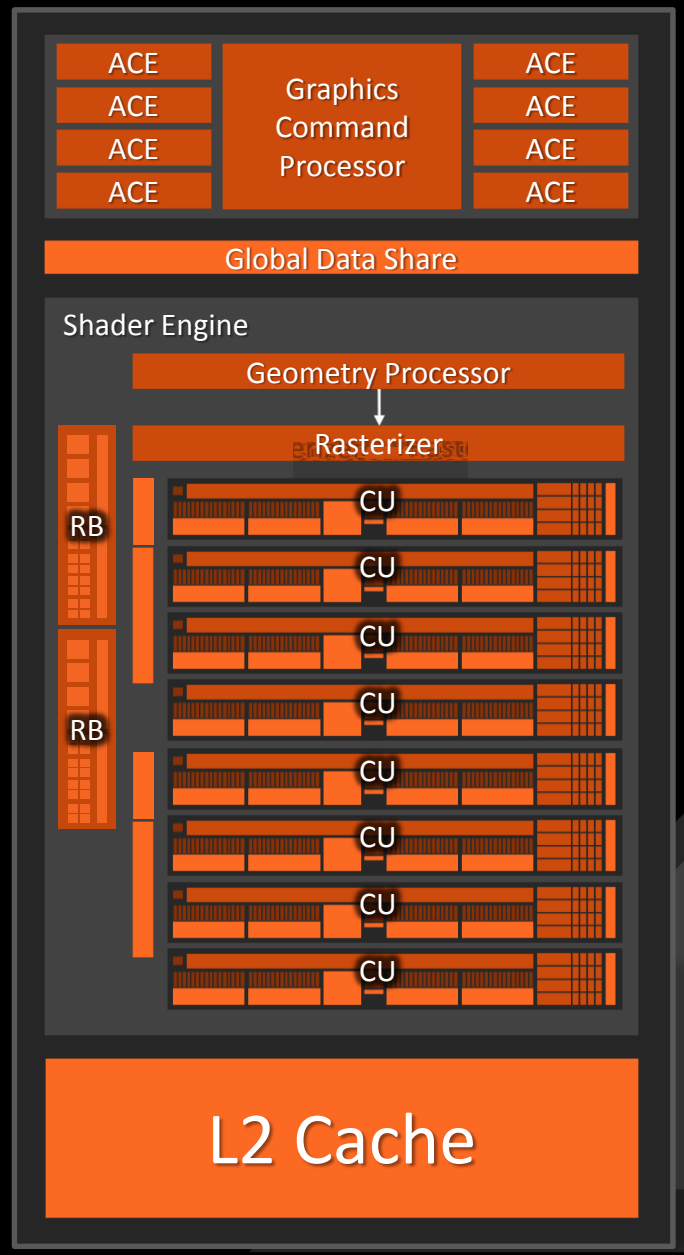
ALL-PROCESSORS-EQUAL



- GPU and CPU have equal flexibility to be used to create and dispatch work items

GRAPHICS CORE NEXT ARCHITECTURE

"KAVERI" GPU – GRAPHICS CORE NEXT (GCN) ARCHITECTURE

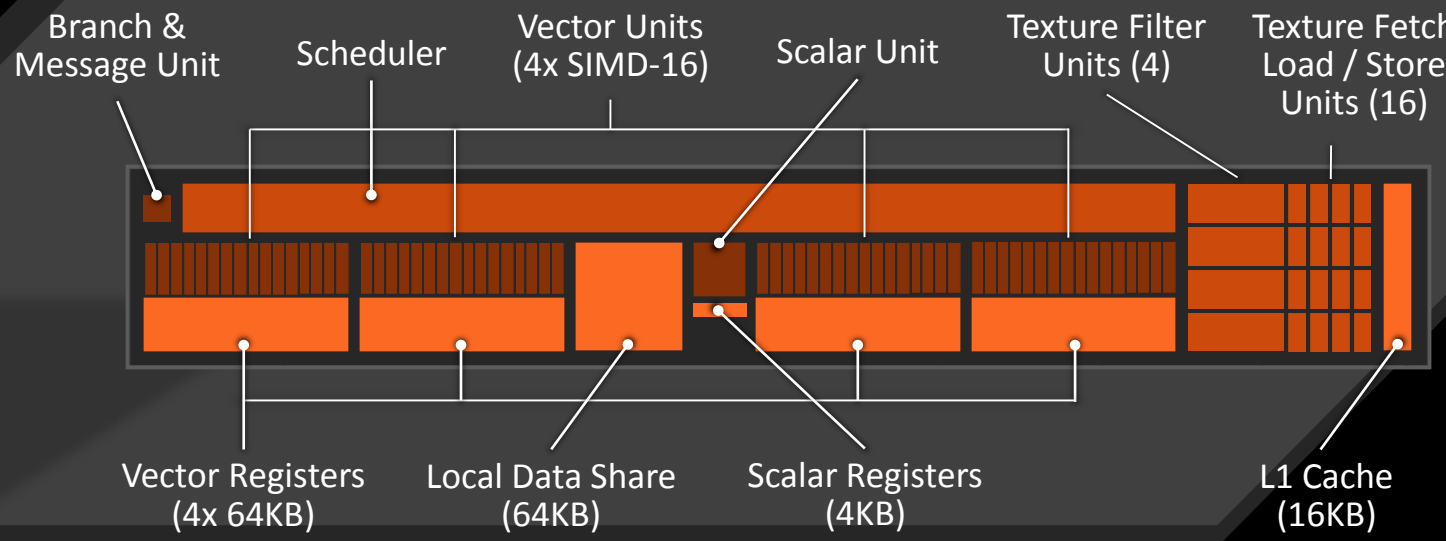


47% of "Kaveri" is dedicated for GPU

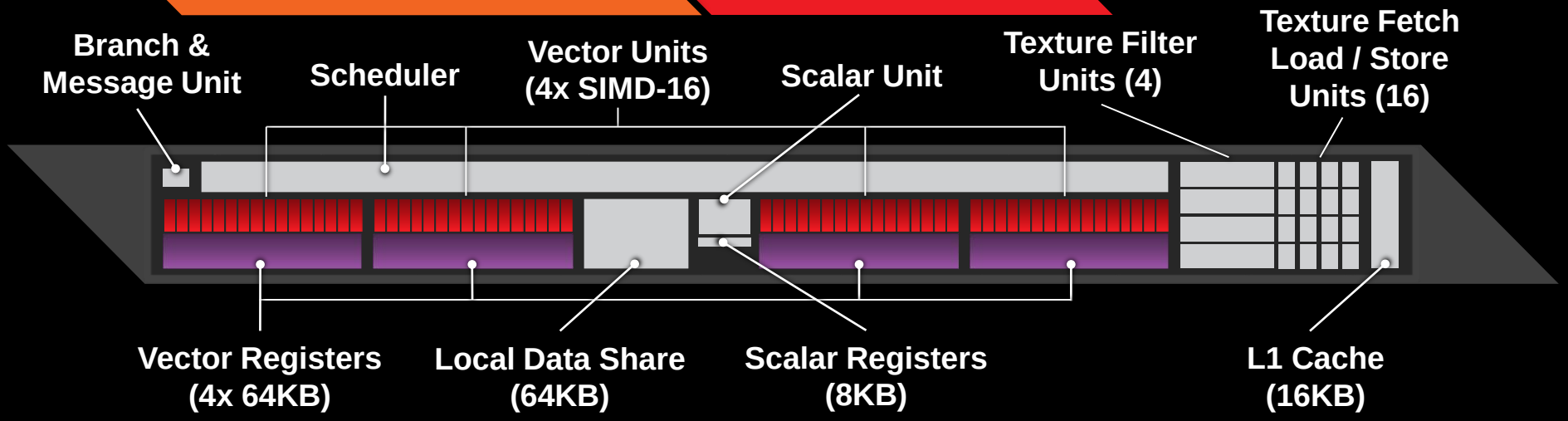
Graphics Command Processor and 8 Asynchronous Compute Engines (ACE) for Multitasking Compute

Shader Engine for Graphics and Compute

8 Compute Units (CUs) = 32 16-lane SIMD Units for Arithmetic Processing

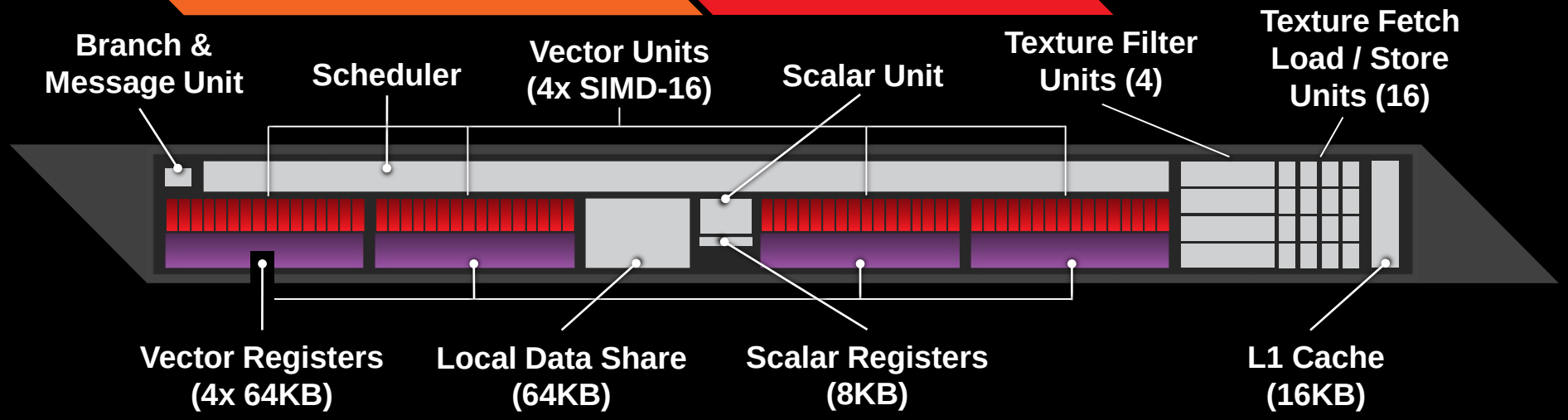


GCN Compute Unit SIMD Specifics



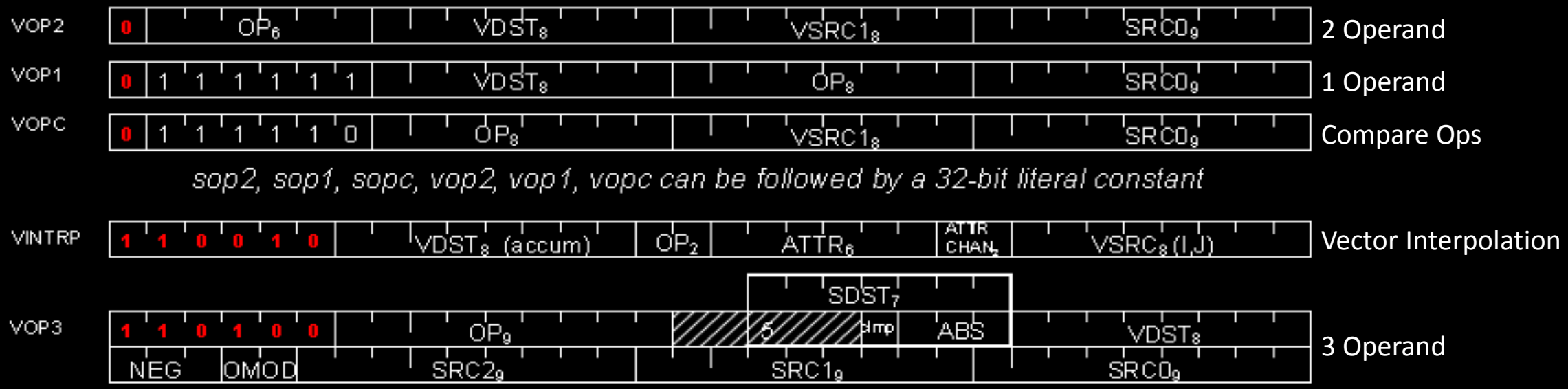
- Each Compute Unit (CU) contains 4 SIMD Vector Units.
- Each SIMD Vector Unit has:
  - A 16-lane integer and floating-point vector ALU
  - 64KB Vector General Purpose Register (VGPR) file
  - A 48-bit Program Counter
  - Instruction buffer for 10 wavefronts
    - A wavefront is a group of 64 threads: the size of one VGPR
- A 64-thread wavefront issues to a 16-lane SIMD Unit over four cycles

GCN Compute Unit — SIMD Specifics



- Each CU can have 40 wavefronts in-flight
  - Up to 2560 parallel threads
- Each 16-lane SIMD Vector Unit
  - Supports half, single, and double precision floating-point arithmetic
  - Issues one half or single precision instruction per-lane per-clock (including FMA)
  - Issues double-precision operations at a reduced rate

VECTOR INSTRUCTIONS



- 1 - 3 source operands from Vector General Purpose Register (VGPR) file
- 1 source operand can be from one of several other sources (e.g., 32-bit constant, scalar unit, local memory)
- 32-bit instruction encoding for most common vector ALU ops
- 64-bit instruction encoding enables 3 source operands and input/output modifiers
  - Can apply absolute value or negate any input operand
  - Can multiply result by 0.5, 1, 2, or 4
  - Can clamp result to the range [-1.0, +1.0]

VECTOR UNIT FLOATING-POINT ARITHMETIC DATATYPES AND MODES

AMD

- **Floating-point Arithmetic Datatypes**
  - 16-bit, 32-bit, and 64-bit floating-point numbers as defined by IEEE 754-2008
  - 32-bit constant floating-point values can be expanded to 64 bits with zero padding on LSBs
- **IEEE Rounding Modes**
  - Round to nearest even, Round toward +Infinity, Round toward –Infinity, Round toward zero
  - Provided under software control anywhere in the program
  - Single and double precision rounding modes are controlled separately
- **Sub-normal Handling Modes**
  - Separate control for input sub-normal flush to zero and underflow flush to zero
  - Provided under software control anywhere in the program
  - Separate control for single and double precision numbers
- **Exceptions Support**
  - Inexact, underflow, overflow, division by zero, sub-normal, invalid operation, and integer divide by zero operation
  - Provided in hardware with mechanisms for software recording and reporting

VECTOR UNIT FLOATING-POINT ARITHMETIC OPERATIONS

AMD

- **FMA** (Fused Multiply Add)
  - Single cycle issue instruction
  - IEEE 754-2008 precise with all round modes and proper handling of Nan/Inf/Zero
  - Full sub-normal support in hardware

- **MULADD** (Multiply Add)
  - Single cycle issue instruction
  - IEEE MUL followed by IEEE ADD with round and normalization after both multiplication and subsequent addition

- **VCMP** (Vector Compare)
  - A full set of compare operations designed to fully implement all the IEEE 754-2008 comparison predicates
  - Return 1 bit per work-item to a named scalar register pair called the Vector Condition Code (VCC) and optionally to an Execute Mask register

- **3 Operand Selection Operations**
  - V_MIN3, V_MAX3, V_MED3

VECTOR UNIT FLOATING-POINT ARITHMETIC OPERATIONS

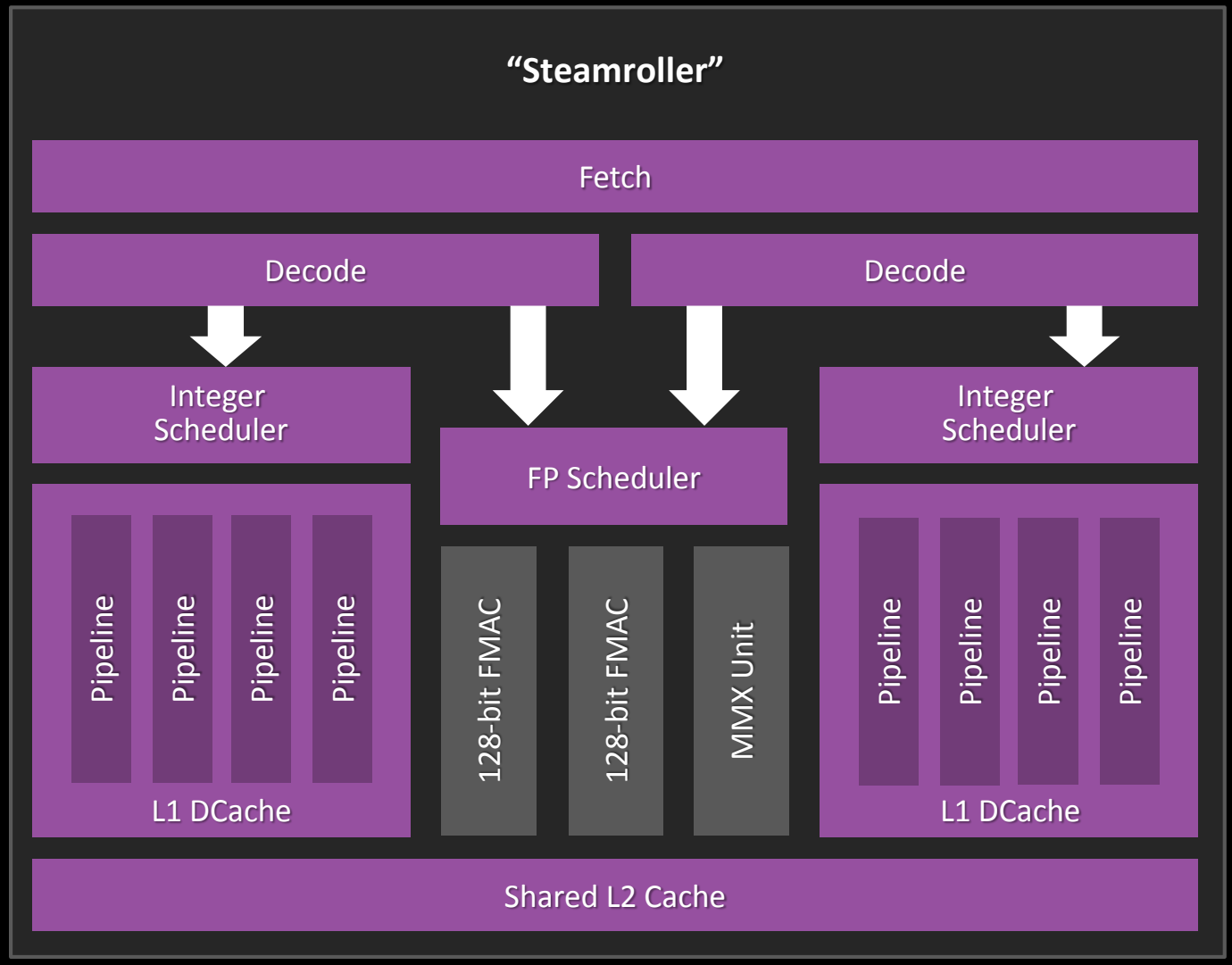
AMD

- **FP Conversion Ops**
  - Between 16-bit, 32-bit, and 64-bit floating-point values with full IEEE 754-2008 precision and rounding
- **64-bit Transcendental Approximations**
  - Hardware based double precision approximations for reciprocal, reciprocal square root and square root
- **16-bit and 32-bit Transcendental Approximations**
  - Hardware based approximations for reciprocal, reciprocal square root, square root, exponent, logarithm, sine, cosine
- **Several other FP Arithmetic Operations**
  - Basic arithmetic operations, min and max, interpolation, rounding and truncation, extract exponent or mantissa, etc.
- **24-bit Integer MUL/MULADD/LOGICAL/SPECIAL at full single-precision rates**
  - Heavily utilized for integer thread group address calculation
- **References:**
  - http://amd-dev.wpengine.netdna-cdn.com/wordpress/media/2013/07/AMD_GCN3_Instruction_Set_Architecture.pdf

STEAMROLLER X86 CPU

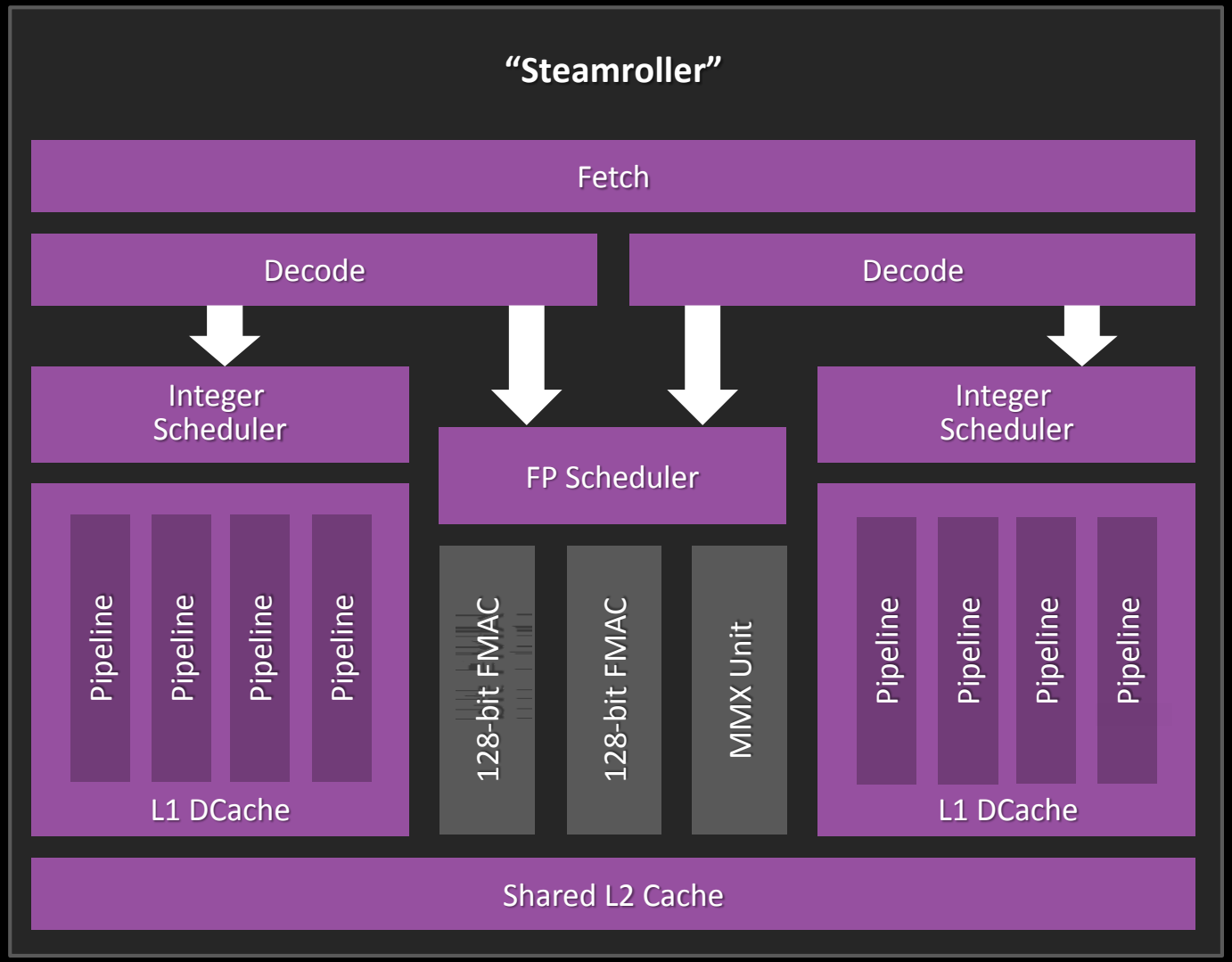
STEAMROLLER CORE MICROARCHITECTURE – SHARED FPU

- Two tightly linked cores share resources to increase efficiency
- Floating-point co-processor organization
- Dual 128-bit FMAC pipes
- MMX Unit for packed integer arithmetic
- Unified scheduler for both threads



STEAMROLLER FPU – INSTRUCTION SET ARCHITECTURE

- Supports all of the x87 FP operations
  - 80-bit floating-point numbers
  - Full support for IEEE 754-2008
- Several ISA extensions including MMX, SSE to SSE 4.2, AES, CLM, AVX, and AVX2
  - Vector operations for multimedia, security, and scientific computing
- Several integer and floating-point instructions with up to 3 source operands



SUMMARY AND FUTURE DIRECTIONS

SUMMARY AND FUTURE DIRECTIONS

AMD

- Vector integer and floating-point arithmetic is very important in CPUs and GPUs
  - GPUs are optimized for high-throughput and power efficiency
  - CPUs are optimized for low-latency and power efficiency
  - Integrating CPUs and GPUs provides efficient processing of parallel and serial code using high-level languages
- Support for IEEE 754-2008 floating-point arithmetic is essential
- Several additional operations provided for graphics, multimedia, and scientific computing
- Future Directions
  - Power-efficient floating-point arithmetic
  - Efficient support for multiple precisions
  - Efficient vector floating-point reduction and fused operations
  - New operations for emerging graphics, multimedia, and scientific computing applications

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