AMD
The future is fusion

ATI Radeon HD5000 Series : In Inside View

Unleashing the Power of Parallel Compute!

Mark Fowler, Fellow AMD

June 2010

Agenda

AMD
The future is fusion

- Radeon 5xxx Product Family Highlights
- Radeon 5870 vs. 4870
- Radeon 5870 Top-Level
- Radeon 5870 Shader Core
- References / Links / Screenshots
- Questions ?

ATI Radeon™ HD 5000 Series of GPU (Evergreen)

	5870 (Cypress)	5770 (Juniper)	5670 (Redwood)	5470 (Cedar)
Process/Transistors	40nm/2.15B	40nm/1.04B	40nm/627M	40nm/292M
Stream Processors	1600	800	400	80
Peak SP Flop Rate	2.72 Teraflops	1.36 Teraflops	620 GFLOPS	120 GFLOPS
Peak DP Flop Rate	544 GFLOPS			
Texel Rate	68 Gtex/sec	34 Gtex/sec	15.5 Gtex/sec	6 Gtex/sec
Pixel Rate	27.2 Gpix/sec	13.6 Gpix/sec	6.2 Gpix/sec	3 Gpix/sec
Max Resolution	6x2560x1600	6x2560x1600	6x2560x1600	4x2560x1600
Memory Type	GDDR5 4.8Gbps	GDDR5 4.8Gbps	GDDR5 4.0Gbps	GDDR5 3.2Gbps
Max Bandwidth	153 GB/s	77 GB/s	64 GB/s	26 GB/s
Max/Idle Board Power	188W/27 W	108W/18 W	61W/14 W	13-15W

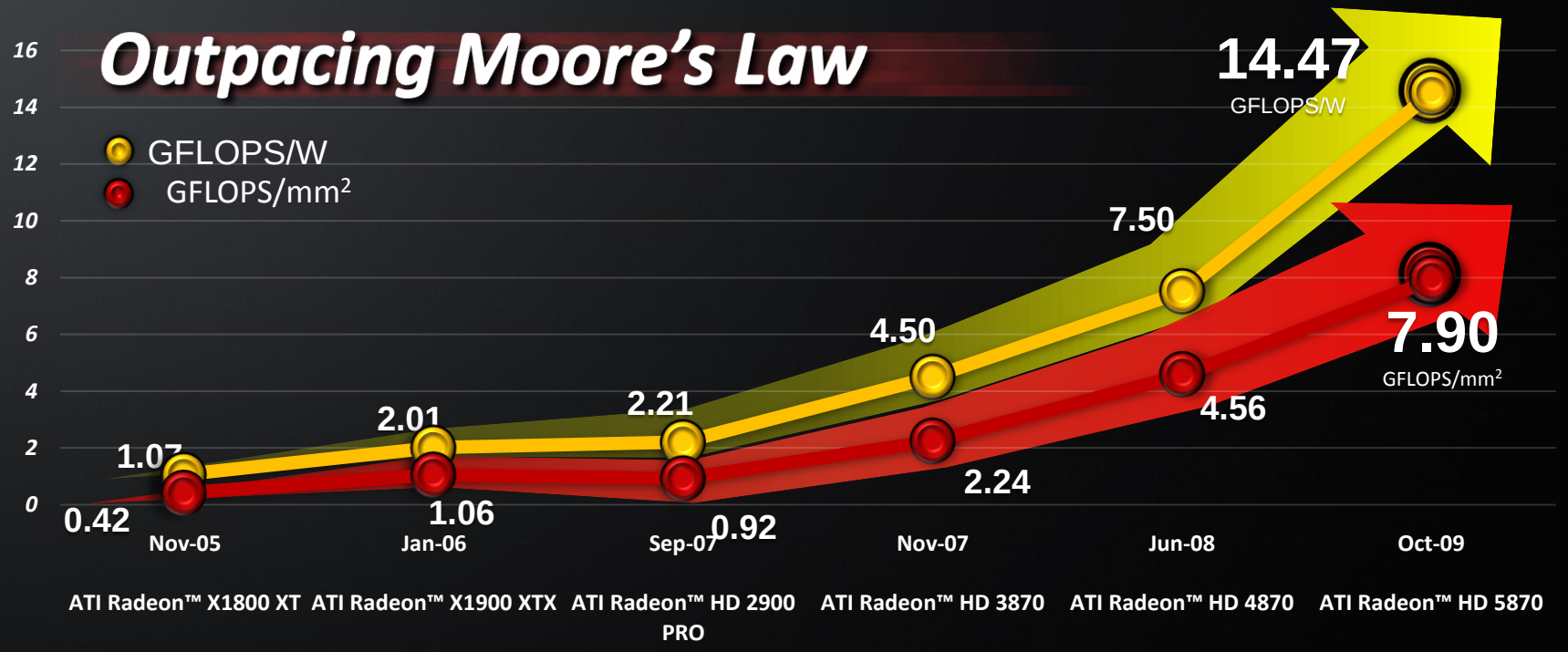
ATI Radeon™ HD 5870 GPU vs. 4870

	ATI Radeon™ HD 4870	ATI Radeon™ HD 5870	Difference
Area	263 mm^2	334 mm^2	1.27x
Transistors	956 million	2.15 billion	2.25x
Memory Bandwidth	115 GB/sec	153 GB/sec	1.33x
L2-L1 Rd Bandwidth	512 bytes/clk	512 bytes/clk	1x
L1 Bandwidth	640 bytes/clk	1280 bytes/clk	2x
Vector GPR	2.62 Mbytes	5.24 MByte	2x
LDS Memory	160 kb	640kb	4x
LDS Bandwidth	640 byte/clk	2560 bytes/clk	4x
Concurrent Threads	15872	31744	2x
Shader (ALU units)	800	1600	2x
Board Power*			
Idle	90 W	27 W	0.3x
Max	160 W	188 W	1.17x

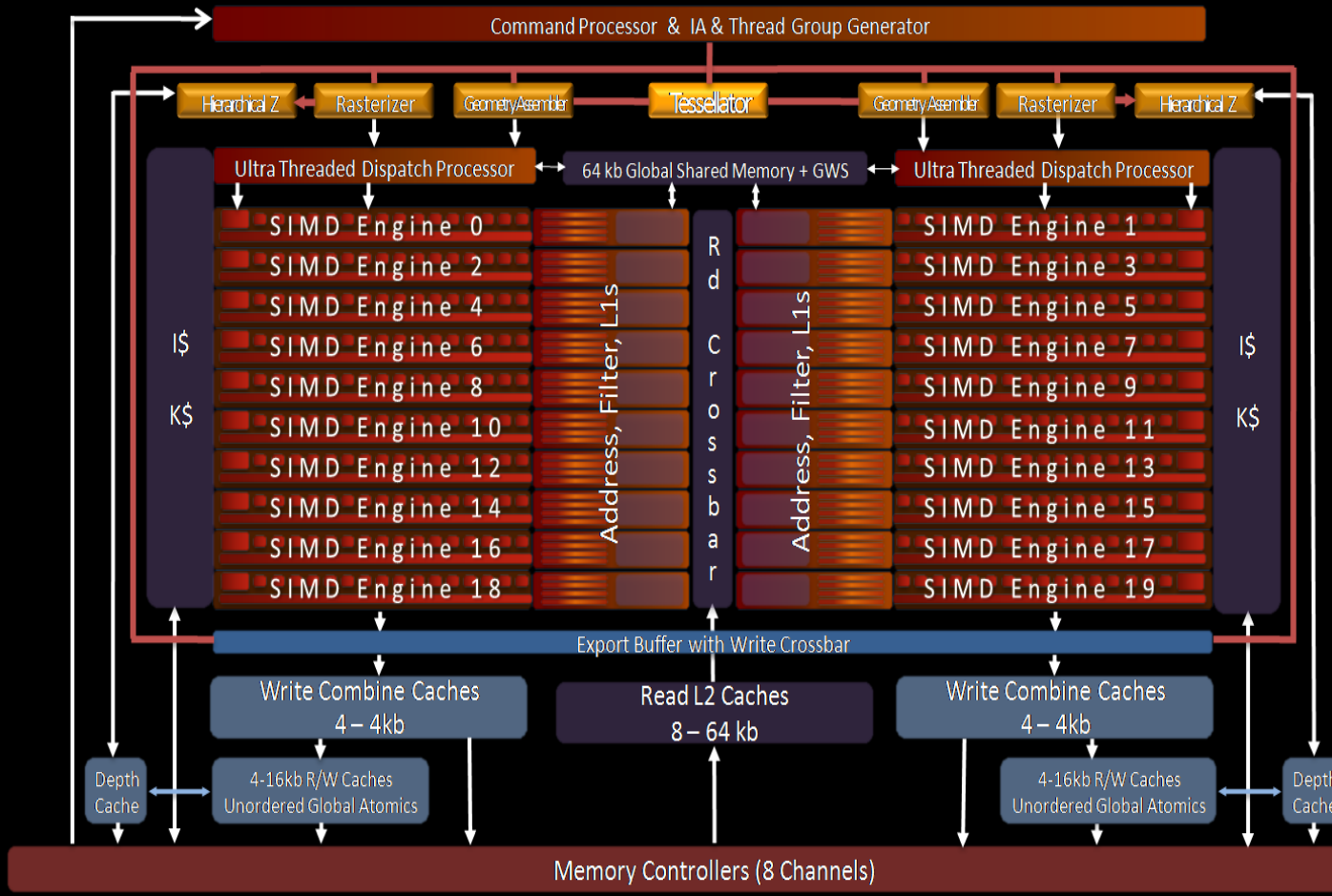
Updated APIs

- dx11 w/ CS 5.0 – Shader Model 5.0
- OpenCL 1.1
- OpenGL 4.0

AMD's GPU Efficiency Trends



Tera Scale 2 Architecture Radeon™ HD 5870



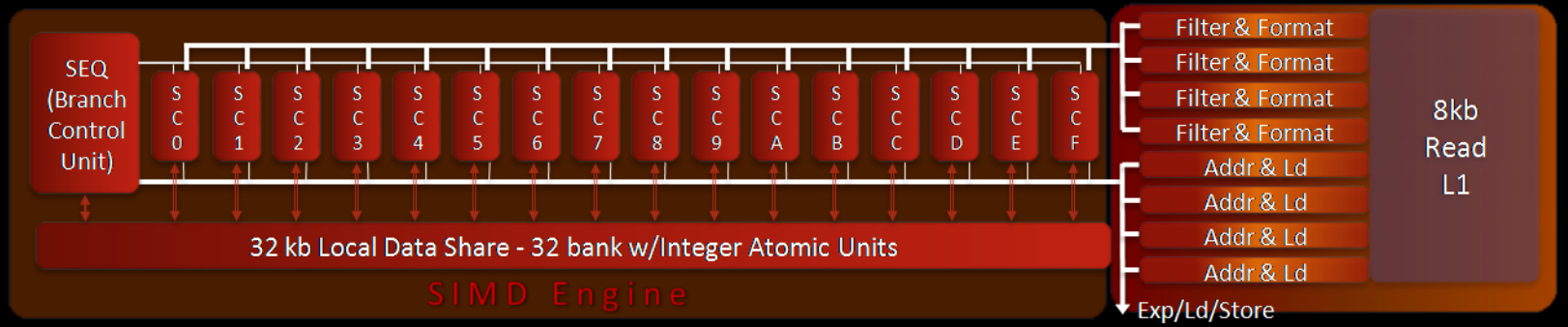
- Double the processing power of previous generation
  - 2.72 Teraflops
  - 27.2 Giga Pixels/sec
- "Dual" Unified Shader Engine
  - I$/K$ for each SE
  - Rasterizer per SE
  - 248 concurrent wave-fronts per SE
  - 16 pixel ROP units per SE
- Non-PS round-robin SE
- PS, Screen-sub-divide
- Dispatch Controller Prog
  - can group PS waves
  - cap a shader stage's inflight wave-fronts

Compute Aspects of ATI Radeon™ HD 5870

AMD
The future is fusion

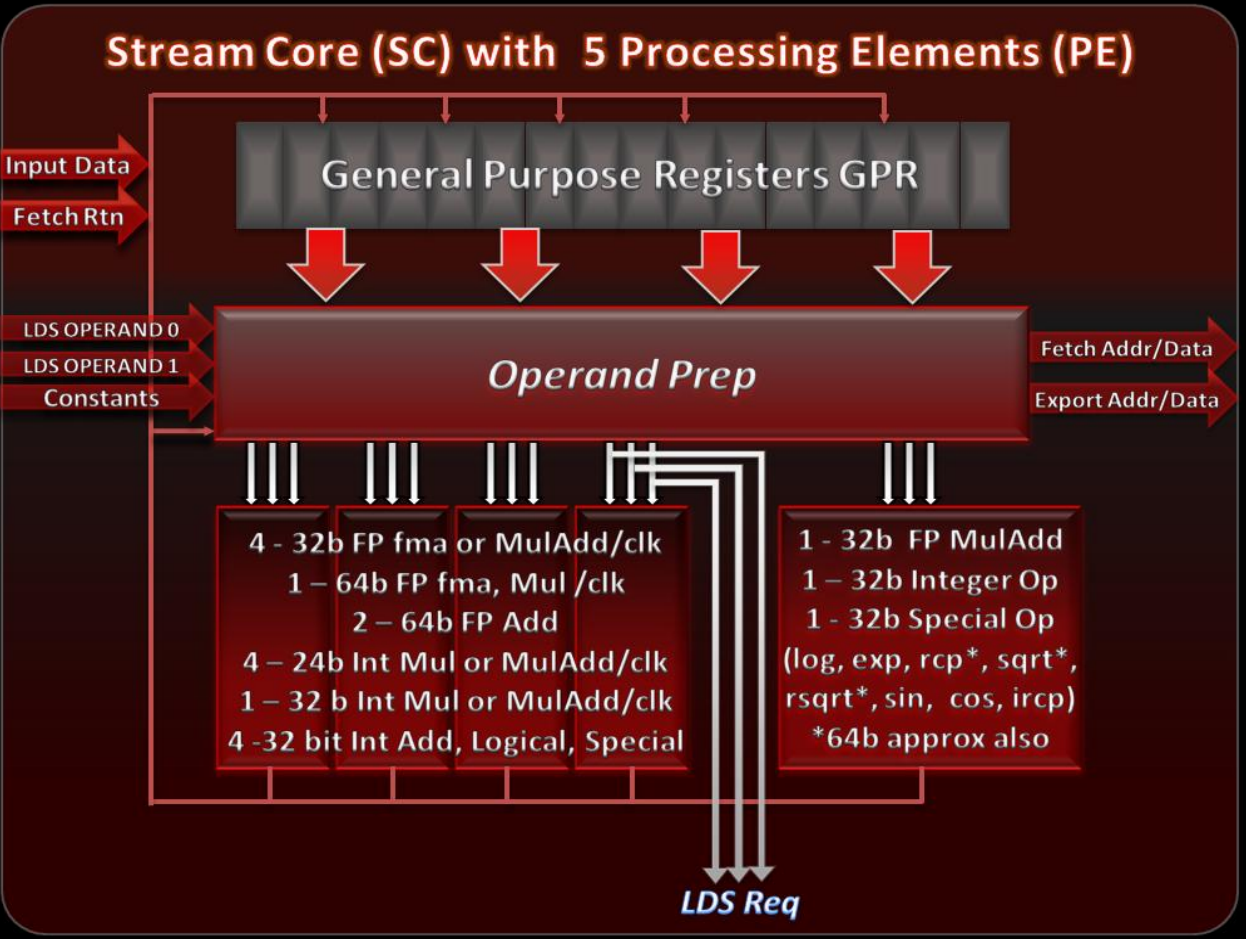
- SIMD Engine
  - Stream Cores
  - Local Data Share (LDS)
- Load / Store / Atomic Data Access
- Dispatch / Indirect Dispatch
- Global Data Share (GDS)

SIMD Engine



- SIMD Engine can process Wavefronts from multiple kernels concurrently
- Thread divergence within a Wavefront is enabled with Lane Masking and Branching
  - Enabling each Thread in a Wavefront to traverse a unique program execution path
- Full hardware barrier support for up to 8 Work Groups per SIMD Engine (for thread data sharing)
- Each Stream Core receives up to the following per VLIW instruction issue
  - 5 unique ALU Ops - or - 4 unique ALU Ops with a LDS Op (Up to 3 operands per thread)
- LDS and Global Memory access for byte, ubyte, short, ushort reads/writes supported at 32bit dword rates
- Private Loads and read only texture reads via Read Cache
- Unordered shared consistent loads/stores/atomics via R/W Cache
- Wavefront length of 64 threads where each thread executes a 5 way VLIW Instruction each issue
  - ¼ Wavelength (16 threads) on each clock of 4 clocks (T0-15, T16-31, T32-47, T48-T63)

Stream Core with Processing Elements (PE)



Each Stream Core Unit includes:

- 4 PE
  - 4 Independent SP or Integer Ops
  - 2 DP add or dependant SP pairs
  - 1 DP fma or mult or SP dp4
- 1 Special Function PE
  - 1 SP or Integer Operation
  - SP or DP Transcendental Ops
- Operand Prep logic
- General Purpose Registers
- Data forwarding and predication logic

Processing Element (PE) Precision Improvements

AMD
The future is fusion

- **FMA** (Fused Multiply Add), IEEE 754-2008 precise with all round modes, proper handling of Nan/Inf/Zero and full de-normal support in hardware for SP and DP
- **MULADD** instruction without truncation, enabling a MUL_{ieee} followed ADD_{ieee} to be combined with round and normalization after both multiplication and subsequent addition.
- **IEEE Rounding Modes** (Round to nearest even, Round toward +Infinity, Round toward –Infinity, Round toward zero) supported under program control anywhere in the shader. Double and single precision modes are controlled separately. Applies to all slots in a VLIW.
- **De-normal Programmable Mode** control for SP and DP independently. Separate control for input flush to zero and underflow flush to zero.
- **FP Conversion Ops** between 16 bit, 32 bit, and 64 bit floats with full IEEE 754 precision.
- **Exceptions Detection** in hardware for floating point numbers with software recording and reporting mechanism. Inexact, Underflow, Overflow, division by zero, de-normal, invalid operation

Processing Element (PE) Improved IPC

AMD
The future is fusion

- Co-issue of dependant Ops in “ONE VLIW” instruction
  - full IEEE intermediate rounding & normalization
  - Dot4 (A=A*B + C*D + E*F + G*H),
  - Dual Dot2 (A= A*B + C*D; F = G*h + I*J)
  - Dual Dependant Multiplies (A = A * B * C ; F = G * H * I;)
  - Dual Dependant Adds (A = B + C + D; E = F + G + H;)
  - Dependant Muladd (A= A*B+C + D*E; F = G*H + I + J*K)

- 24 bit integer
  - MUL, MULADD (4 – co-issue)
  - Heavy use for Integer thread group address calculation

Processing Element (PE) New Integer Ops

AMD
The future is fusion

- 32b operand Count Bits Set
- 64b operand Count Bits Set
- Insert Bit field
- Extract Bit Field
- Find first Bit (high, low, signed high)
- Reverse bits
- Extended Integer Math
  - Integer Add with carry
  - Integer Subtract with borrow
- 1 bit pre-fix sum on 64b mask. (useful for compaction)
- Shader Accessible 64 bit counter
- Uniform indexing of constants

Processing Element (PE) Special Ops

AMD
The future is fusion

- Conversion Ops
  - FP32 to FP64 and FP64 to FP32 (w/IEEE conversion rules)
  - FP32 to FP16 and FP 16 to FP32 (w/IEEE conversion rules)
  - FP32 to Int/UInt and Uint/Int to FP32
- Very Fast 8 bit Sum of Absolute Differences (SAD)
  - 4x1 SAD per lane, with 4x4 8 bit SAD in one VLIW
  - Used for video encoding, computer vision
  - Will be exposed via OpenCL extension
- Video Ops
  - 8 bit packed to float and float to 8 bit packed conversion Ops
  - 4 8 bit pixel average (bilinear interpolation with programmable round)
  - Arbitrary Byte or Bit extraction from 64 bits

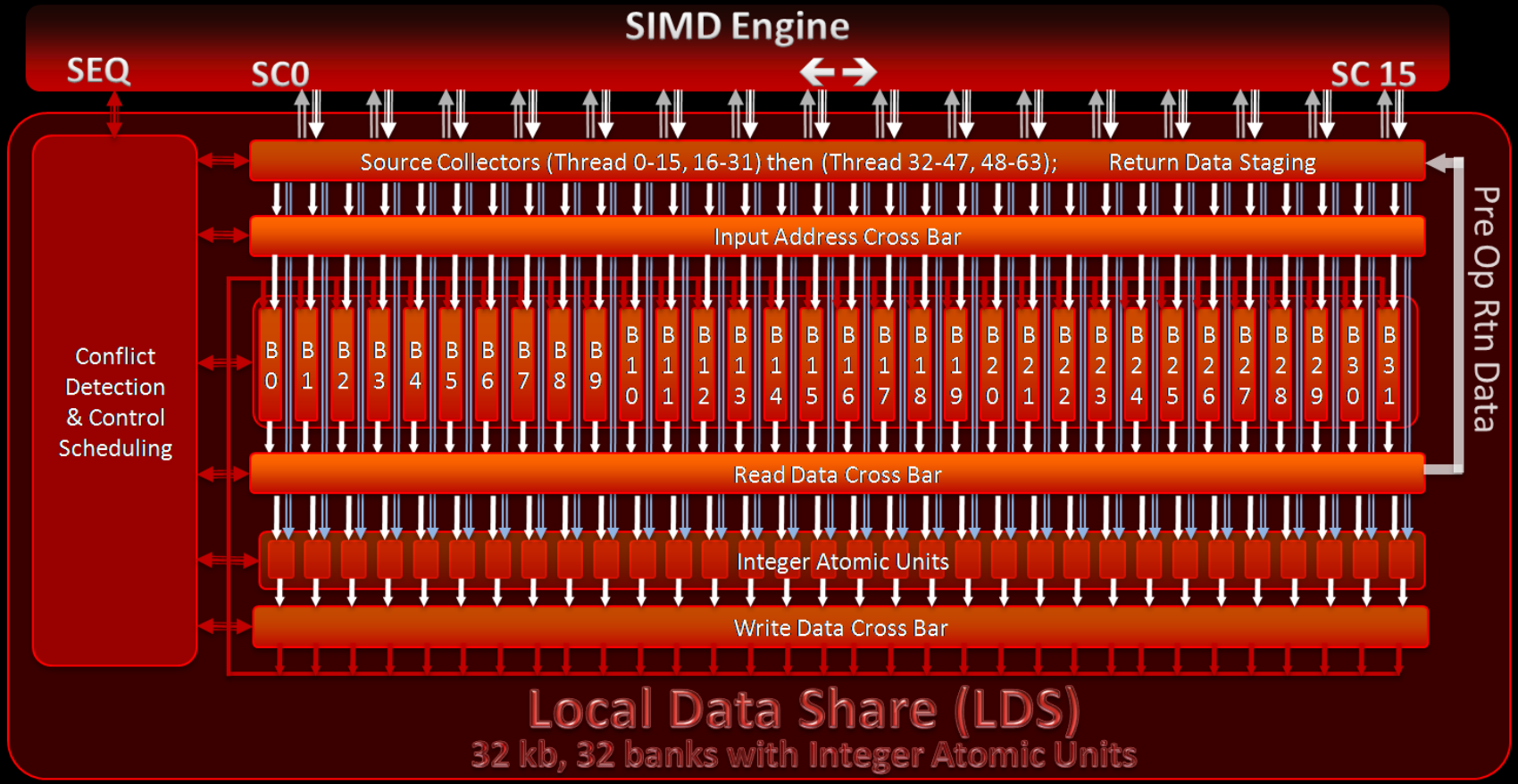
Local Data Share (LDS)

AMD
The future is fusion

Share Data between Work Items of a Work Group designed to increase performance

- High Bandwidth access per SIMD Engine (1024b/clk) – Peak is double external R/W bandwidth (512b/clk)
- Low Latency Access per SIMD Engine
  - 0 latency direct reads (Conflict free or Broadcast)
  - 1 VLIW instruction latency for LDS indirect Op
- All bank conflicts are hardware detected and serialized as necessary with fast support for broadcast reads
- Hardware allocation of LDS space per thread group dispatch
  - Base and size stored with wavefront for private access
- 32 – byte, ubyte, short, ushort reads/writes per clk (reads are sign extended)
- 32 dwords access per clock
  - Load/Stores
  - Atomics: add, sub, inc, dec, min, max, and, or, xor, exchange, compare _swap
  - Return pre-Op value to Stream Core 4 primary Processing Elements

Local Data Share (LDS)

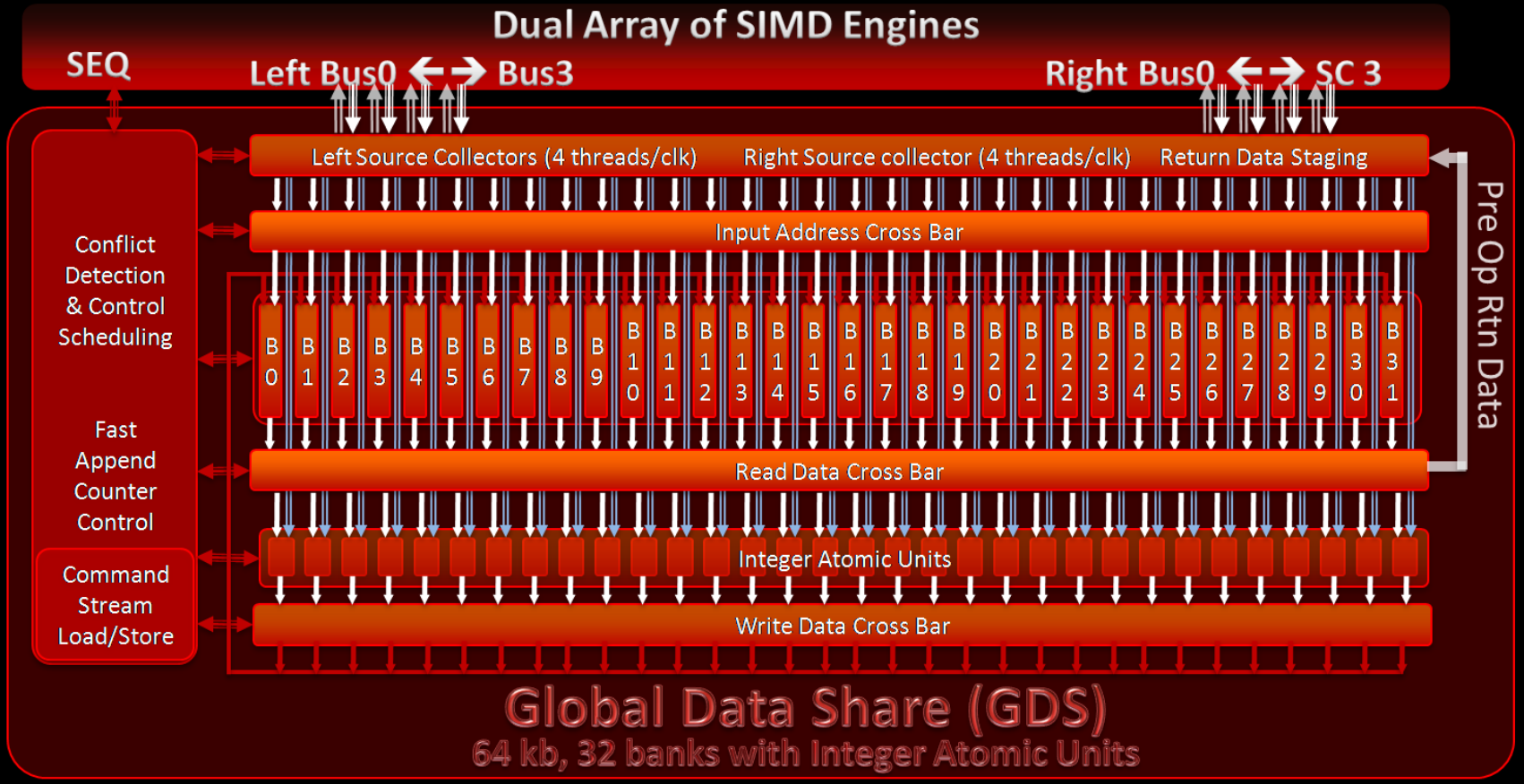


Global Data Share (GDS)

AMD
The future is fusion

- Similar to LDS except it is shared memory for entire dispatch (grid) as opposed to a work-group
- Low Latency Access to a global data shared memory between all threads in a kernel
  - 25 clks latency
  - Issued in parallel to math similar to fetch, return to GPR
- Why GDS vs. using regular global memory ?
  - Large number of threads going at small amount of memory (ex. append counters, reduction) can create choke point, and under utilization of the hardware
  - Separate memory (GDS) for small shared allocation, can free up global memory for use by the shader sequencer for other wave-fronts
  - GDS is parallel to global memory
- Counters for append buffers here, and supports “Ordered” append as well
  - The append buffer is built in dispatch order

Global Data Share (GDS)



References

Links to AMD's OpenCl tutorial and some samples
http://developer.amd.com/gpu/ATIStreamSDK/pages/Publications.aspx

AMD Developers Central : Samples, Tools, Downloads, White papers etc.
http://developer.amd.com/gpu/ATIStreamSDK/Pages/default.aspx

Khronos Open Cl : Specification, Introduction Slides, and Quick Reference
www.khronos.org/opencl/

Direct Compute Introduction
http://msdn.microsoft.com/en-us/directx/bb896684.aspx

http://developer.amd.com/samples/demos/pages/ATIRadeonHD5800SeriesRealTimeDemos.aspx

Order-independent transparency (OIT)

http://developer.amd.com/samples/demos/pages/ATIRadeonHD5800SeriesRealTimeDemos.aspx

DirectX® 11 Depth of Field in Action

Summary

AMD
The future is fusion

- ATI Stream™: Age of Teraflop Processing
- Leadership in performance, power and price
- Open software strategy based on standards
- Compliant DirectX® 11 Direct Compute & OpenCl 1.1
- Available Today in Stores!!

Questions!

Accelerate

CPU

GPU

Only

AMD

ATI STREAM TECHNOLOGY

OpenCL

KHRONOS GROUP

DirectCompute

Microsoft

Disclaimer and Attribution

AMD
The future is fusion

DISCLAIMER

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

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