



# Future Directions in Parallel Computing

---

Alexandre David

1.2.05

adavid@cs.aau.dk



# Attached Processors

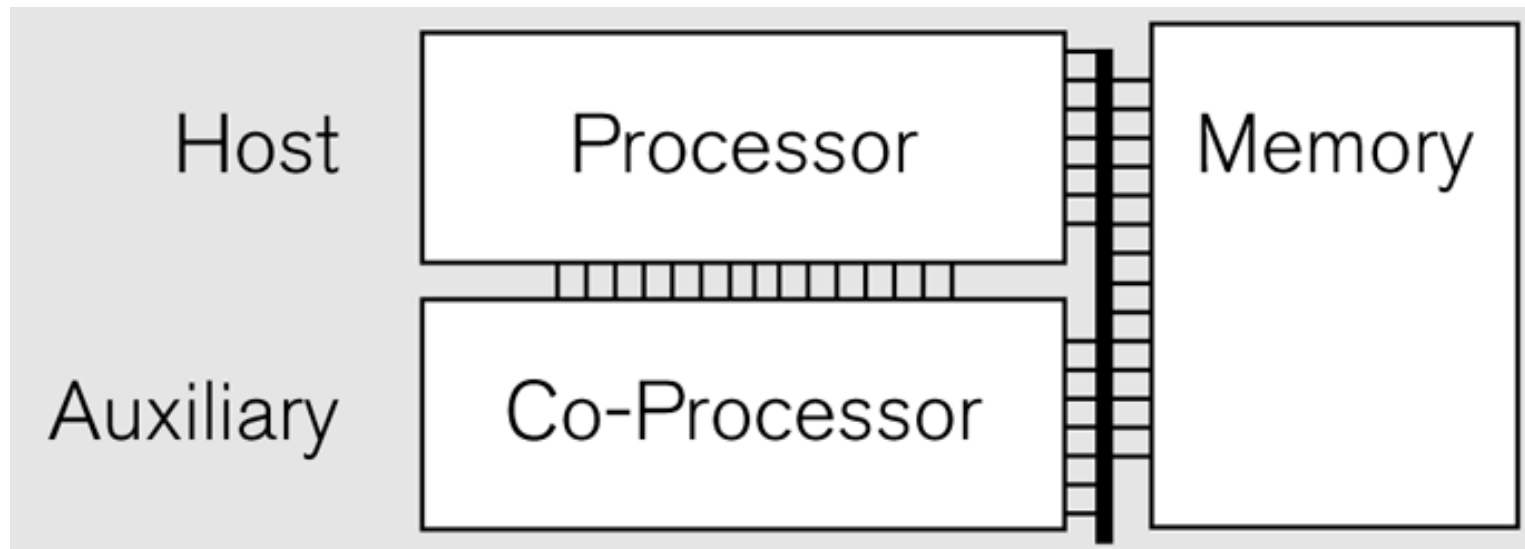
---

- Specialized processors can be more powerful & space efficient than general purpose CPUs.
  - Off-load the work to attached co-processors.  
Old days: co-processor for 80286.
  - Asymmetric.
  - Specialized high parallel algorithms run on the co-processor.
  - FPGA – GPU – Cell architecture.



# Attached Processors

---



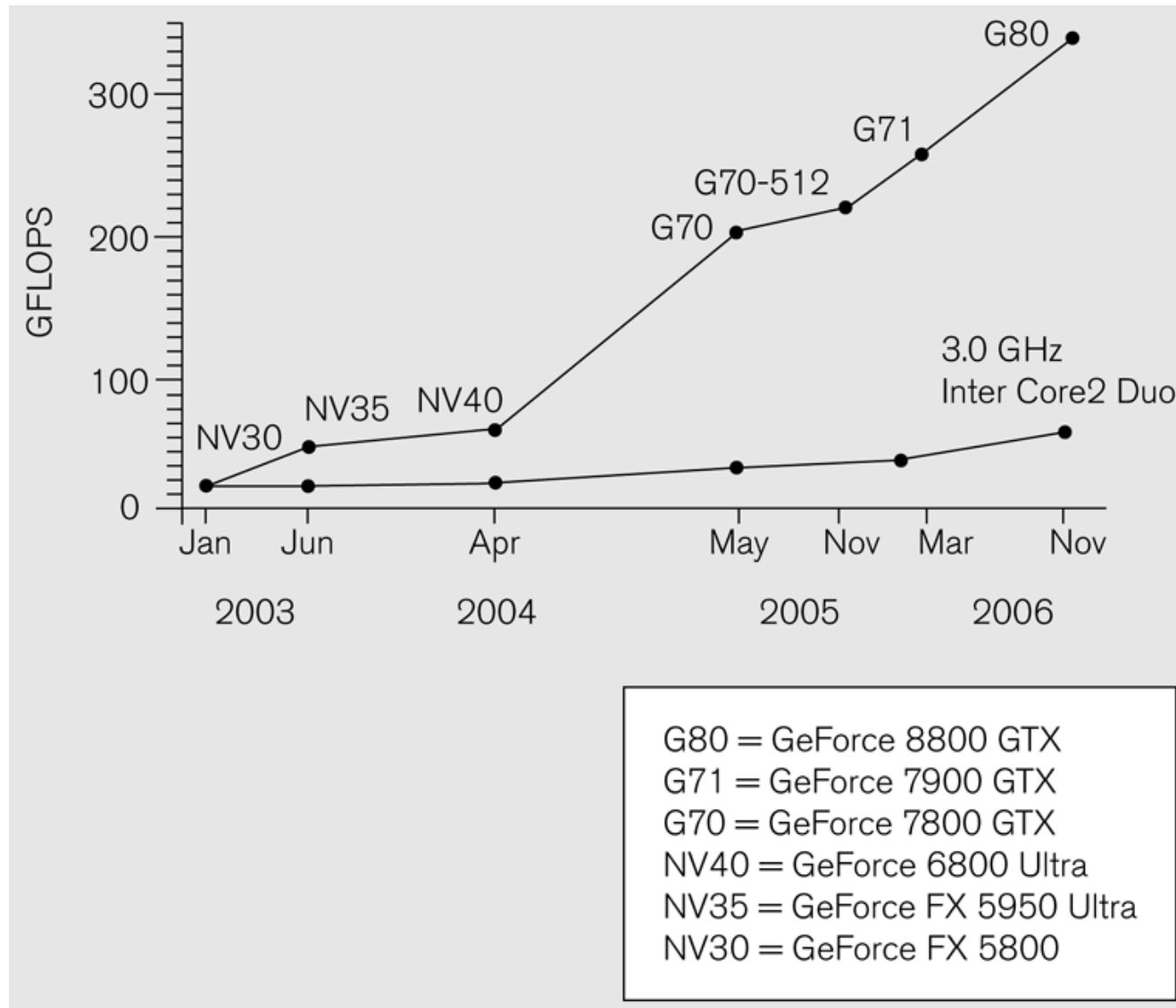


# GPU

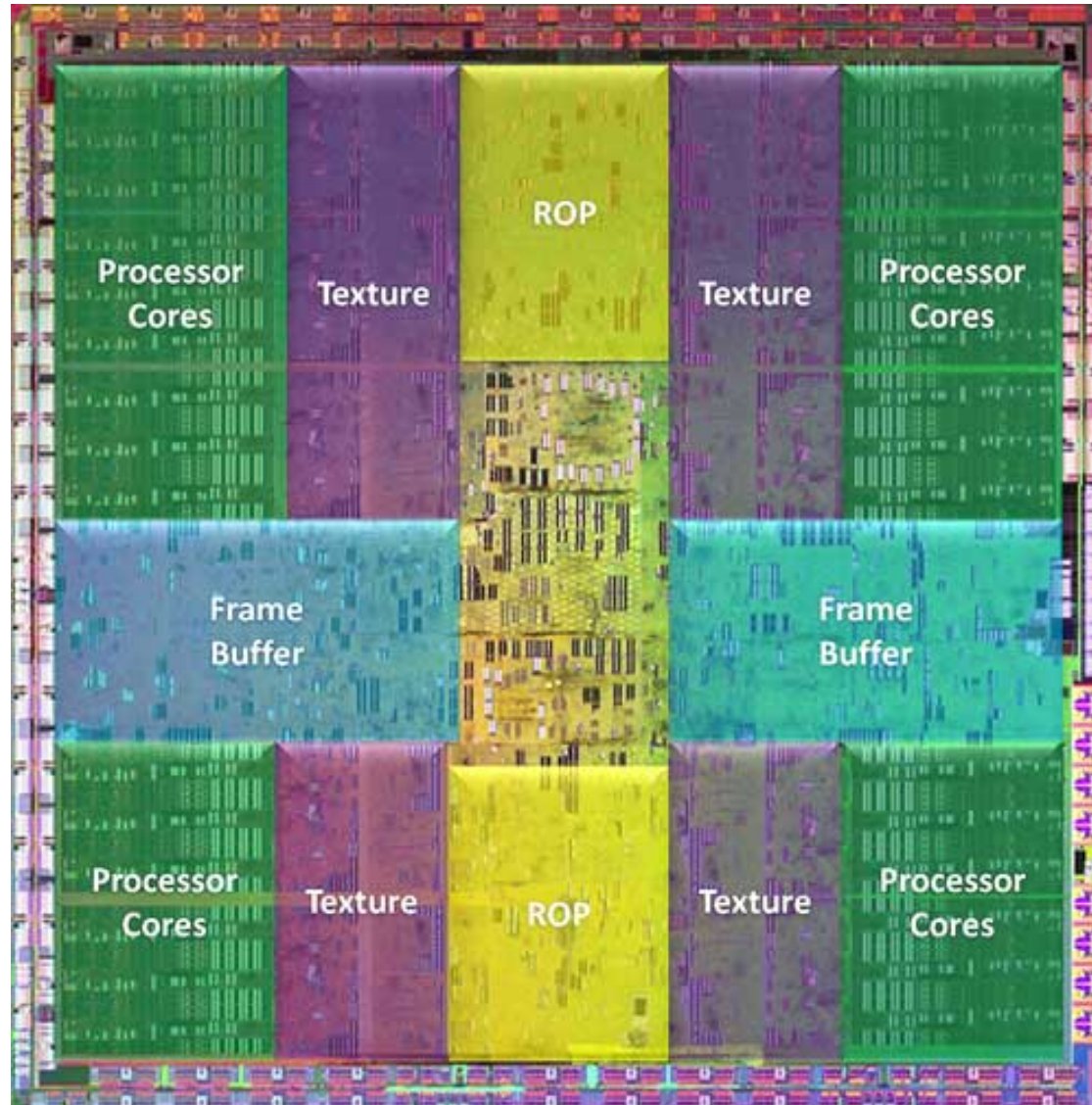
---

- GPGPU – recent effort.
  - Tremendous performance, price/performance, and quick new generations.
  - 10x flops, 10x bandwidth than CPU typical.
  - GPU is specialized for graphics rendering
    - compute intensive
    - highly parallel task
    - many processing cores
    - highly data-parallel architecture, simplified logic, no issue such as reordering instructions, branch prediction, cache etc...

# GPU vs. CPU Over Time



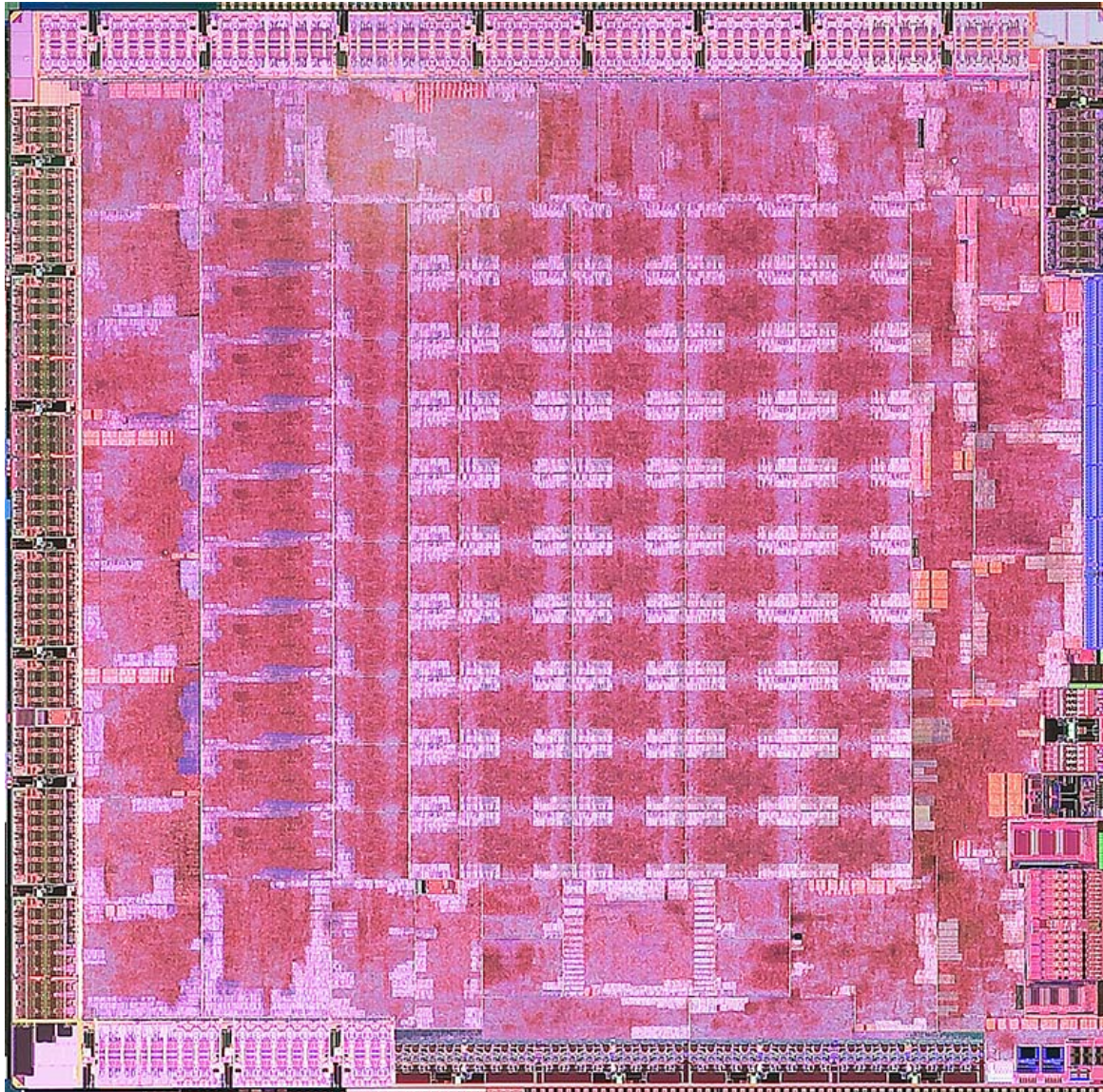
# GT200







# RV770



# RV770 (Caution: AMD's Marketing)

## Introducing the TeraScale graphics engine



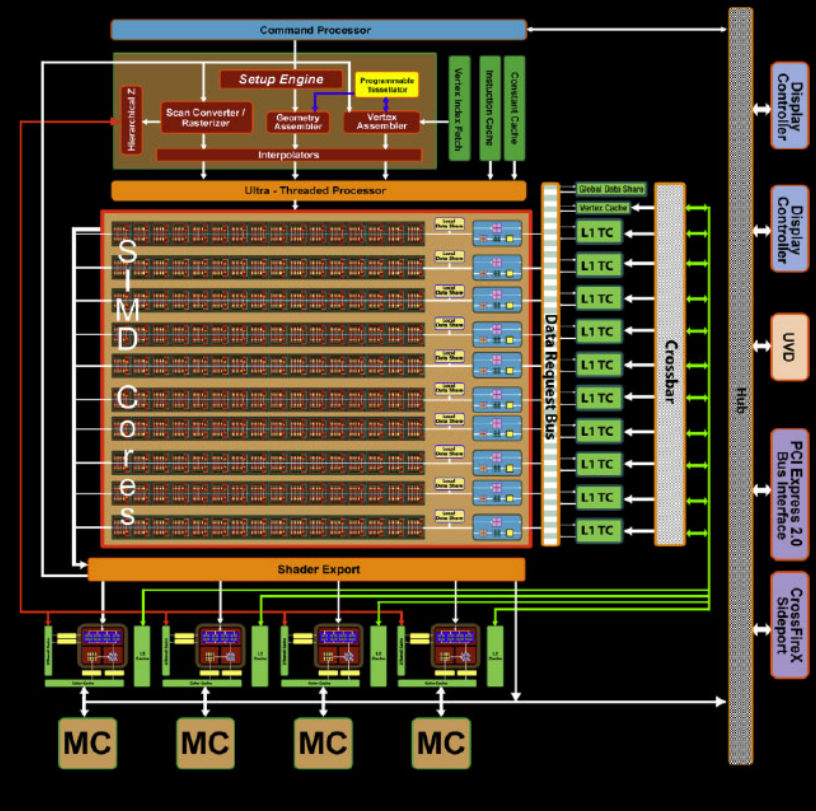
	ATI Radeon™ HD 3800	ATI Radeon™ HD 4800	Difference
<b>Process</b>	55nm	55nm	None
<b>Die Size</b>	190mm <sup>2</sup>	260mm <sup>2</sup>	1.4x
<b>Transistors</b>	666M	956M	1.4x
<b>4x AA</b>	8	16	2x
<b>Z/Stencil</b>	32	64	2x
<b>Texture</b>	16	40	2.5x
<b>Shader</b>	320	800	2.5x
<b>Bandwidth</b>	72 GB/sec	115.2 GB/sec	1.6x



# RV770 (Caution: AMD's Marketing)

## Terascale Graphics Engine

- 800 highly optimized stream processing units
- New SIMD core layout
- Optimized texture units
- New texture cache design
- New memory architecture
- Optimized render back-ends for faster anti-aliasing performance
- Enhanced geometry shader & tessellator performance



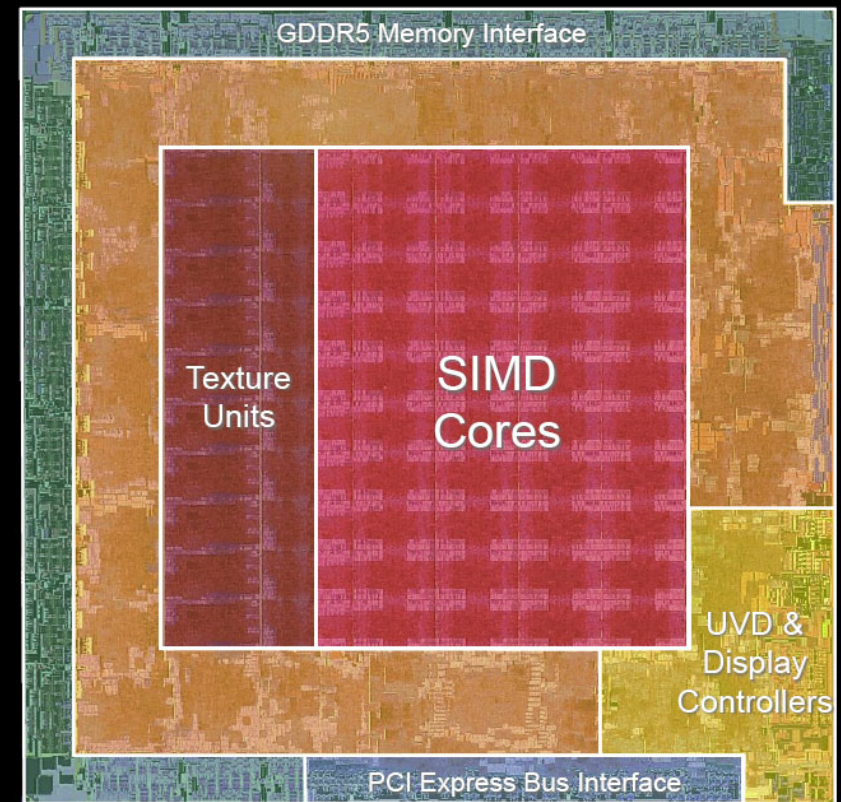
# RV770 (Caution: AMD's Marketing)

## ATI Radeon™ HD 4800 Series Architecture

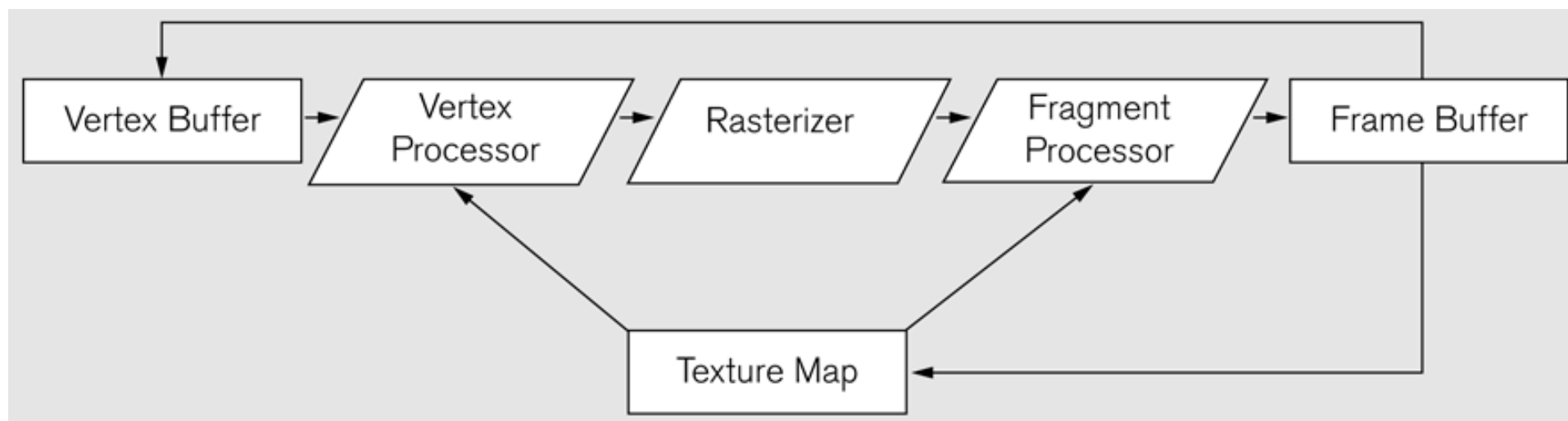


- 10 SIMD cores
  - Each with 80 32-bit Stream Processing Units (800 total)
- 40 Texture Units
- 115+ GB/sec GDDR5 memory interface

	ATI Radeon™ HD 3800	ATI Radeon™ HD 4800	Difference
<b>Die Size</b>	190 mm <sup>2</sup>	260 mm <sup>2</sup>	1.4x
<b>Memory</b>	72 GB/sec	115 GB/sec	1.6x
<b>AA Resolve</b>	32	64	2x
<b>Z/Stencil</b>	32	64	2x
<b>Texture</b>	16	40	2.5x
<b>Shader</b>	320	800	2.5x



# The Graphics Pipeline





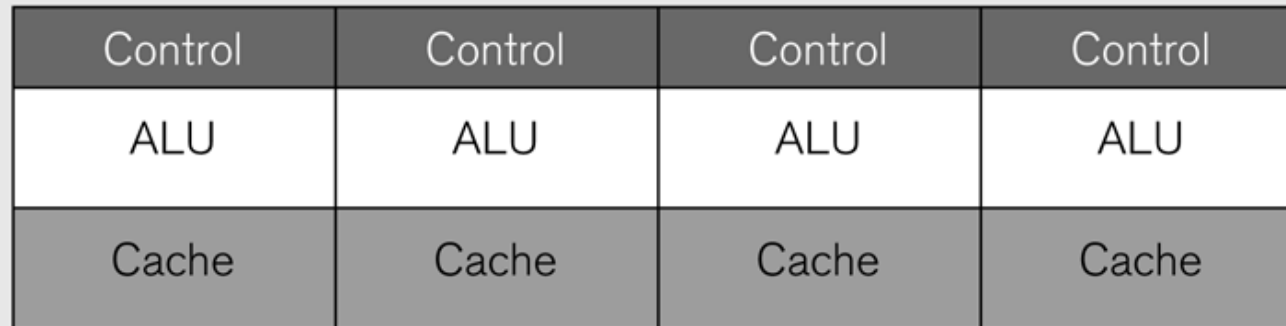
# GPU

---

- Beginning: Only for graphics.
- Over time, more programmable with more registers, more code (shaders).
- Trend: more general purpose computing.
  - Double precision FPUs, integer operations.



# CPU vs. GPU



**(a)**



**(b)**



# GPUs

---

- Computing cores: SIMD.
- Programming model (CUDA) close to C.
- Power: massively parallel floating-point computations.
  - Organize packs of threads (warp).
  - Schedule many (1000) threads at each cycle.

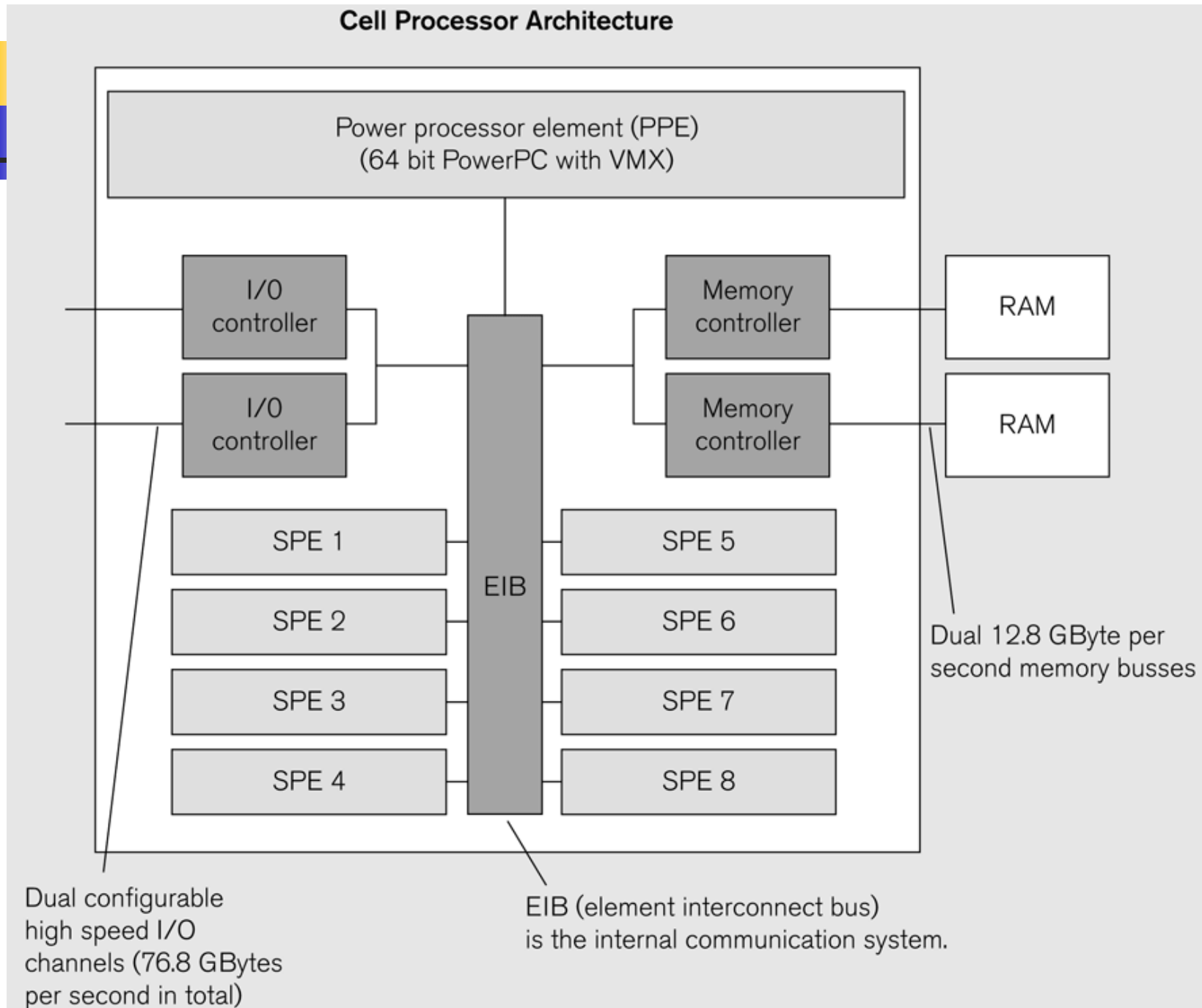


# Cell Processor

---

- Designed for video games & multimedia applications.
- Architecture:
  - 1 PPE 64 bit PowerPC, dual-threaded.
  - 8 SPE, connected via element interconnect bus (EIB). Powerful SIMD processors.
  - EIB ~ 200GB/s.
  - SPE: no cache but local small memory (256kB)  
128 bit SIMD RISC processor, vector operations.

## Cell Processor Architecture







# Grid Computing

---

- Henrik's lecture



# Transactional Memory

---

- Idea: Memory behaves like database.
  - Transactions ensure the 4 ACID properties:
    - Atomicity – all operations complete or none does.
    - Consistency – storage updated as if there is a serial ordering.
    - Isolation – effect of the transaction is equivalent to the effect of some isolated execution.
    - Durability – the changes persist.
- A transaction is *committed* or *aborted*.



# Transactions vs. Locks

---

- Deadlocks with locks. Not with transactions.
- Locks enforce sequential executions. Transactions offer more concurrency.
- Finer granularity with transactions.
- Locks do not compose well. Deadlock possible, priority inversion possible...
- Transactional memory difficult to implement. More in SPO course.



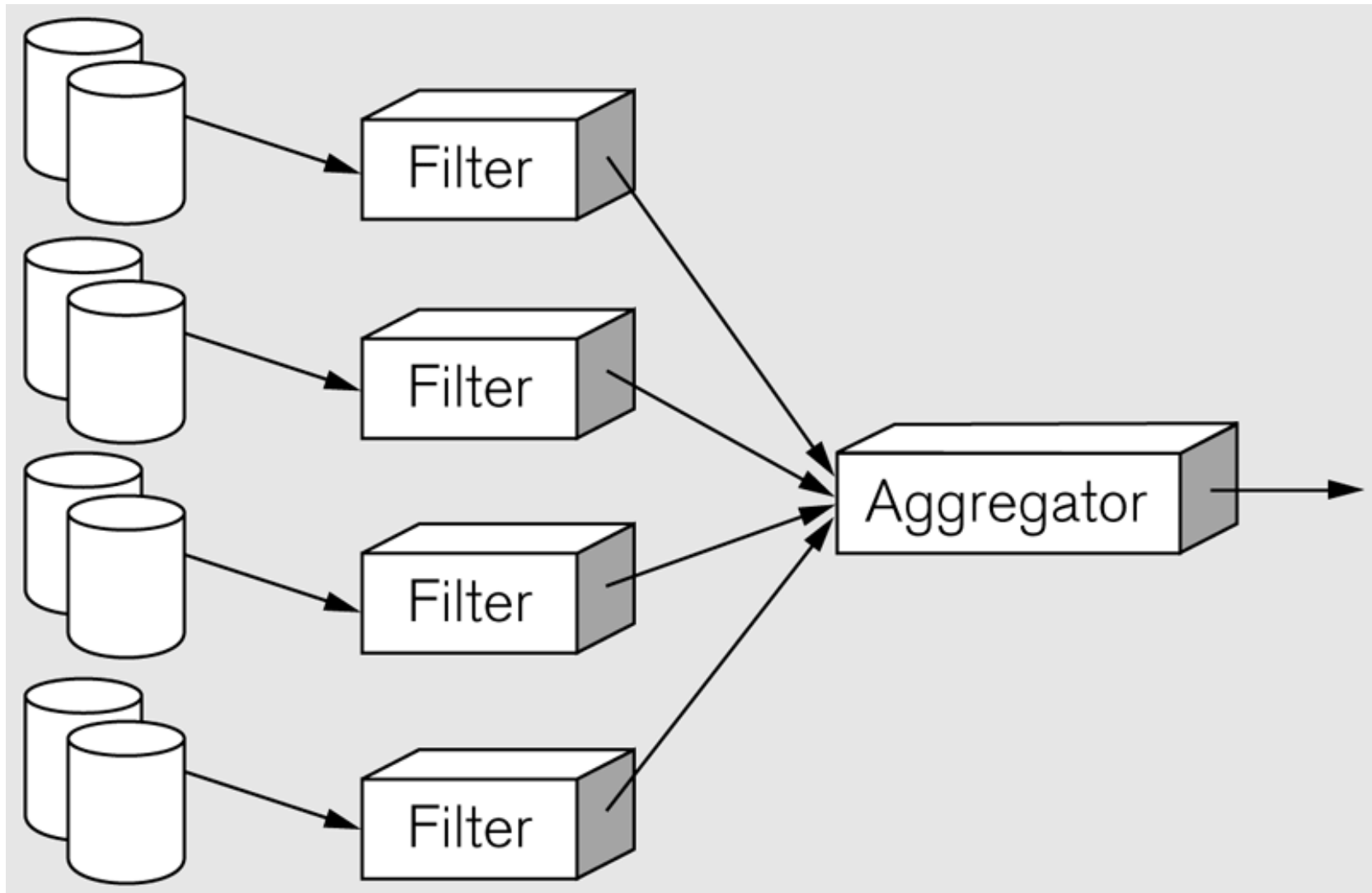
# MapReduce

---

- Tool for searching huge data archives using a plug-in style framework.
  - Used by Google to compute PageRank.
- Scale critical (very big).
  - Tree with leaves = disks.
  - Computations produce tables of values.
  - Data streamed & filtered through a map function (output = key,value pairs).
  - The pairs are streamed to an aggregator – reduce function.



# MapReduce





# Emerging Languages

---

- Chapel – Cray
- Fortress – Sun Microsystems
- X10 – IBM