GPUs vs. Multicore CPUs: On a Converging Course or Fundamentally Different?

Bill Mark

Graphics Hardware 2008 Panel

June 19, 2008

Position Summary

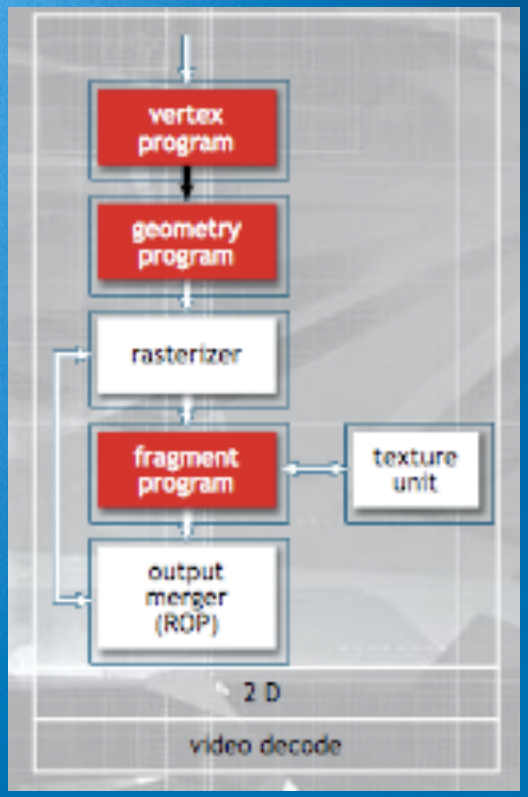
GPUs and throughput-oriented multi-core CPUs are converging

However:
 Some specialization for graphics is still important.

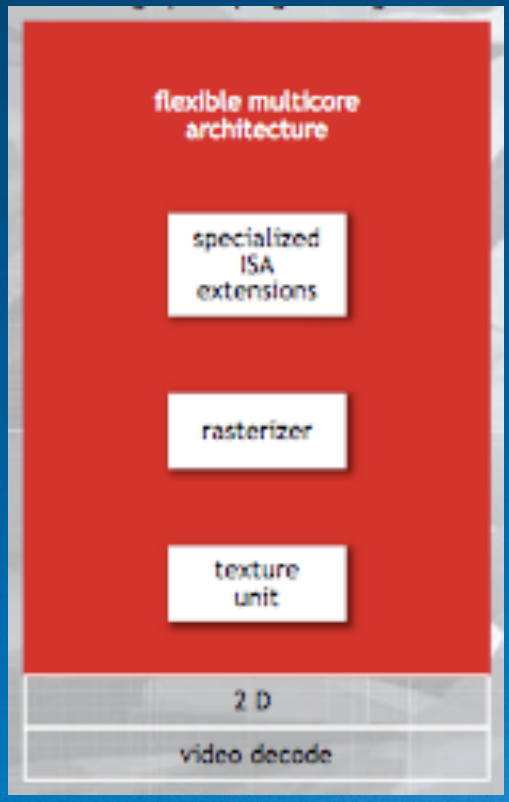
intel

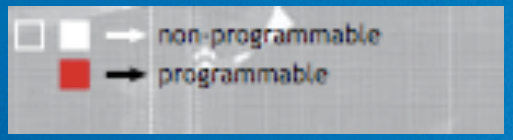
Why convergence is possible

Past: HW dictates framework

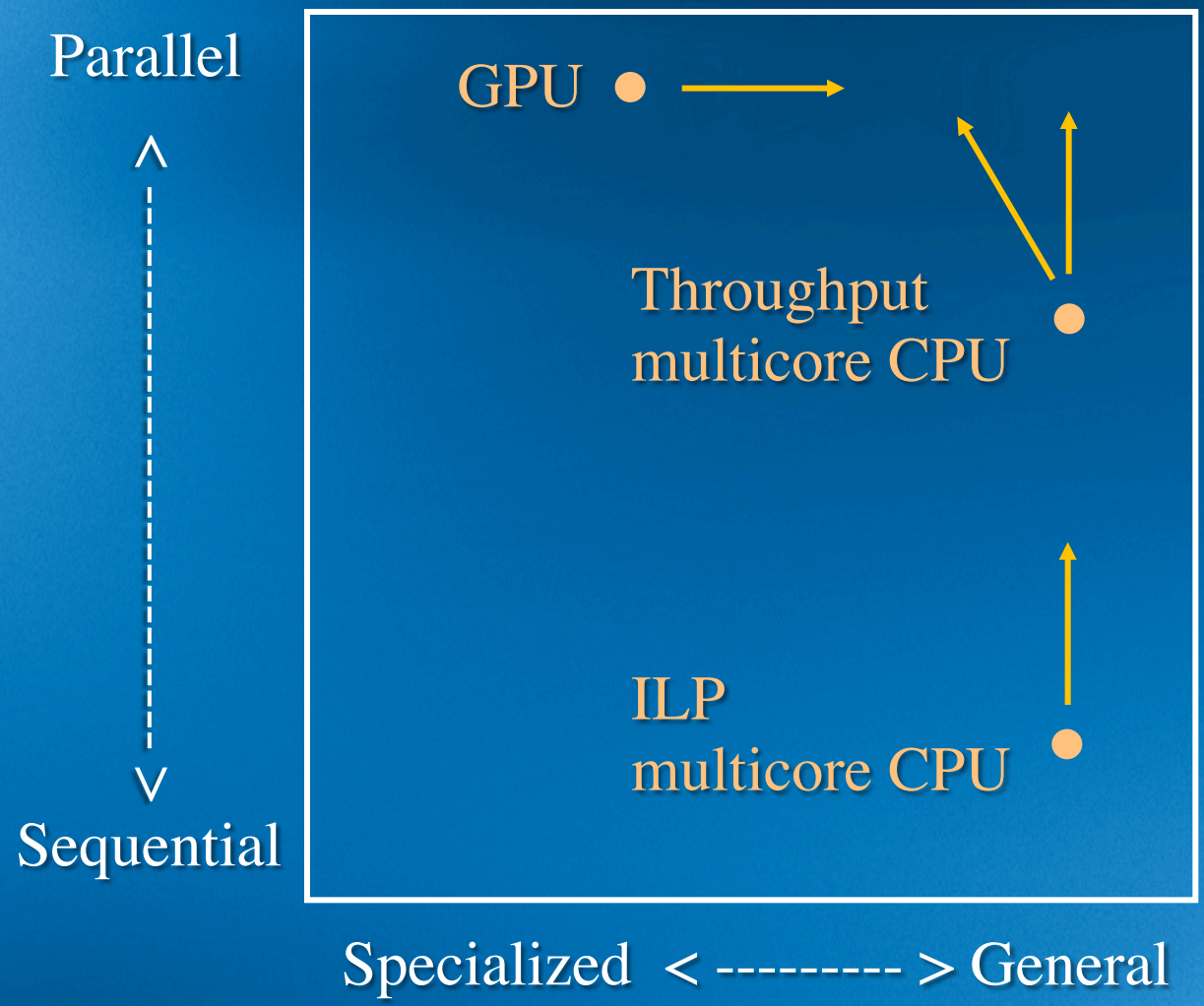


Future: SW defines framework





GPUs converging with throughput CPUs



Traditional CPU/GPU differences

	ILP CPU	Traditional GPU
# of cores	1	Many
Wide SIMD float?	No	Yes
Specialized HW units?	No	Yes
Clock rate	High	Low
DRAM bandwidth	Low	High
Cache/scratch size	Large	Small
Programming model	General purpose	Very constrained
Direct HW access	Yes	No – via driver/JIT
Generality	Any application	Just 3D rendering

intel

Throughput multi-core vs. modern GPU

	Throughput CPU	Modern GPU
# of cores	Many	Many
Wide SIMD float?	No	Yes
Specialized HW units?	No	Yes
Clock rate	Moderate	Moderate
DRAM bandwidth	Medium or High	High
Cache/scratch size	Moderate	Moderate
Programming model	General purpose	Constrained
Direct HW access	Yes	No – via driver/JIT
Generality	Any application	3D rendering + GPGPU

intel

Throughput multi-core for graphics vs. modern GPU

	Graphics Throughput CPU	Modern GPU
# of cores	Many	Many
Wide SIMD float?	Yes	Yes
Specialized HW units?	Yes	Yes
Clock rate	Moderate	Moderate
DRAM bandwidth	High	High
Cache/scratch size	Moderate	Moderate
Programming model	General purpose	Constrained
Direct HW access	If desired	No – via driver/JIT
Generality	Any application	3D rendering + GPGPU

intel

Remaining differences

	Graphics Throughput CPU	Modern GPU
# of cores	Many	Many
Wide SIMD float?	Yes	Yes
Specialized HW units?	Yes*	Yes*
Clock rate	Moderate	Moderate
DRAM bandwidth	High	High
Cache/scratch size	Moderate	Moderate
Programming model	General purpose	Constrained
Direct HW access	If desired	No – via driver/JIT
Generality	Any application	3D rendering + GPGPU

* Choice of specialized units could differ depending on various factors.

Possible differences in more detail

- Details of Z buffer algorithm:
  - How sorting, Z culling, etc. work
  - When and how DRAM is accessed
  - Exact HW/SW tradeoffs
- Flexibility of programming model
  - Task parallelism?
  - Flexibility of communication and synchronization
  - Work scheduling mechanisms
- Memory models:
  - Scratchpad vs. cache vs. coherent cache, etc.

intel

Why flexibility is useful for rendering

Standard Z buffer has trouble with "advanced" effects

- Z buffer is good for primary visibility of opaque surfaces
- Anything else has problems:
  - Shadows
  - Partial transparency
  - Motion blur; depth of field
  - Volumetric effects (smoke, fire)
  - Global illumination
  - …

“Hacks” for Z buffer are brittle

- You can hack any effect you want for a specific case
- But hacks are brittle:
  - Not robust
  - Not interoperable with each other
- This is a big problem for content creation

- Example: shadows + partial transparency

intel

Converged HW will allow more algorithmic flexibility

- Enhanced Z buffer pipelines
- REYES
- Raytracing
- Better integration of scene management with rendering
- …

intel

Conclusion

- Throughput CPU HW is converging with GPU HW
- But, some specialization for rendering is still critical
  - Intel definitely understand this
- Flexibility will benefit rendering as well as other uses

intel

END