ARM Midgard Architecture

Anton Lokhmotov, ARM

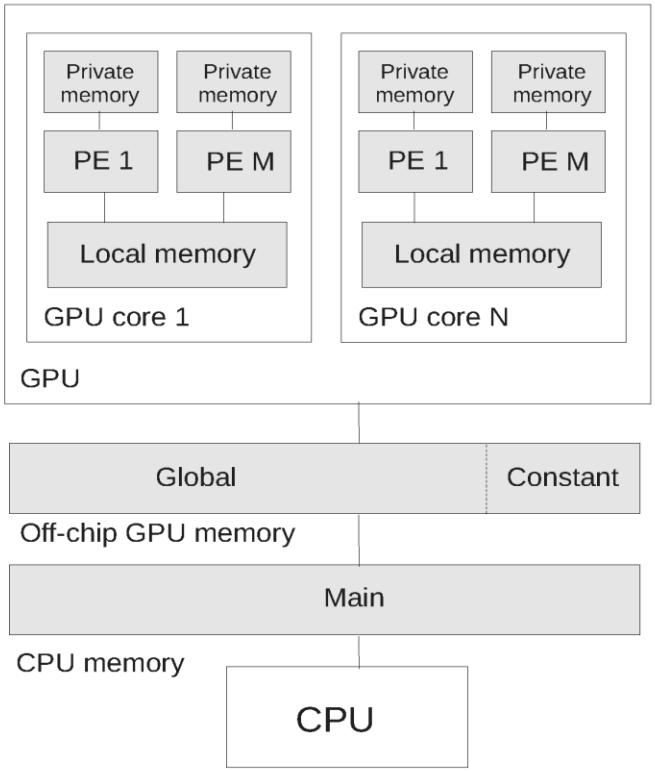
(OpenCL tutorial, HiPEAC'11)

ARM Mali (Midgard) GPU Architecture

- OpenCL v1.1 (full profile) compliant, with focus on:
  - Performance, flexibility and scalability
  - Memory bandwidth, area and energy efficiency
  - System performance (CPU + GPU + memory + interconnect)
- Barrel-threaded (like AMD/NVIDIA)
- No SIMT execution (unlike AMD/NVIDIA)
  - Hardware view: hard to build fast and efficient load/store units
  - Software view: hard to understand coalescing rules
  - No branch divergence either!
- SIMD execution (like AMD)
  - Should use vectors to achieve the highest performance (or rely on automatic vectorisation)

Memory in desktop systems

- Desktop systems have non-uniform memory
  - GPU is on a discrete card along with GPU (__global) memory
- Data must be physically copied between CPU (main) memory and GPU memory
  - Some algorithms take longer to perform the copying than to execute just on the CPU



Memory in embedded systems

- Most ARM-based systems have uniform memory
  - GPU __global memory allocated in main memory (but fully cached in the GPU's caches)
  - GPU __local memory is also allocated in main memory
- Cheap copying between CPU and GPU
  - Cache coherency operations are faster than physical copying