Mobile and embedded computing on Mali™ GPUs

Anton Lokhmotov, Media Processing Division, ARM

2nd UK GPU Computing Conference, Cambridge
13 December 2010
Mali – Five Years of Business Success

Cumulative Mali GPU licensees

Industry leadership
- Most widely-licensed GPU architecture
- Performance leadership
- Energy and memory bandwidth efficiency leadership
- Software integration with ecosystem

Mali-enabled devices launched in 2010
The Perpetual Demand for Performance

Graphics processing increase of >50x

Within a device power budget average of 850mW for handheld and less than 3W for DTV

Desktop: TrueForce (complexity=10)

OpenGL ES 1.1 Samurai (complexity=1)

Increasing screen resolution >5x

WXGA

WVGA

OpenClear ES 1.1 Samurai (complexity=5)

Increasing content complexity >10x

OpenClear ES 2.0 Taiji (complexity=5)

Image credit: Rightware

Image credit: Rightware

Bringing Visual Entertainment to Life
ARM Mali-T604 architecture

- Innovative GPU architecture
  - Tri-pipe – designed for performance and flexibility (only small parts are graphics only)
  - Functional units – designed with requirements of general computing
- State-of-the-art bandwidth reduction
  - Optimized tile-based architecture
- System approach
  - CPU + GPU + Memory + Interconnect
  - Performance scalability via multicore design
ARM Mali architecture (vs. desktop)

- Barrel-threaded
- Low bandwidth
- No warp-based execution
  - Hardware: hard to build fast and efficient load/store units
  - Software: hard to understand coalescing rules
- No branch divergence either!
- Should use vectors to achieve the highest performance (or rely on auto-vectorisation)
Memory – desktop systems

- Desktop systems have non-uniform memory
  - GPU global memory is on a discrete card along with the GPU
- 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 – ARM-based systems

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

- Portable low-level interface for accelerated systems
  - Accelerated systems – fast and energy efficient hardware
  - Interface – platform, runtime, compiler, libraries, etc.
  - Low-level – close-to-the-metal, efficient
  - Portable – as in: “compliant code runs on compliant implementation”, but no guarantee of performance-portability

- New requirement for high-end GPUs and CPUs
OpenCL applications

- The scientific computing community has trail-blazed the use of GPUs for general purpose computation, showing that GPUs are:
  - Highly efficient for structured grid, dense lin. algebra, particle methods
  - Promising for unstructured grid, sparse lin. algebra, spectral methods

- Extrapolating that, the mobile and embedded computing community will find GPUs useful for:
  - Speech processing
  - Artificial intelligence
  - Entertainment
  - Image processing
OpenCL applications

- Consumer entertainment (including games)
  - Jaw-dropping graphics (e.g. using photorealistic ray tracing, or custom render pipelines)
  - Intelligent “artificial intelligence” (e.g. really smart opponents)
  - 3D spatialization of sound effects including multiplayer voice chat
  - Novel user interfaces (e.g. eye, gesture, and conversation controlled)

- Advanced image processing
  - Computational photography (e.g. digital, region based focussing)
  - Augmented reality (e.g. heads-up navigation, “live” gaming)
  - Computer vision (e.g. automotive safety applications)
  - Eye tracking
OpenCL applications

- Procedural Texture Generation
- Augmented Reality
- Computational Photography
- High Dynamic Range Imaging (HDR)
ARM OpenCL driver (v1.1, full profile)

- Plug-in architecture
  - ARMv7-A + Neon
  - Mali-T600
  - Custom Device
  - Video Decoder

- Designed for performance and portability
  - Linux, Android, Symbian and WinCE ports
ARM OpenCL compiler and library

- Compiler based on Clang/LLVM technology (+ a proprietary backend shared with OpenGL ESSL compiler)
  - Modifications to Clang to support OpenCL C are being open-sourced

- Many built-in functions are implemented directly in hardware
  - Mali-T600 balances OpenCL coverage & hardware size
  - Neon has excellent support for vector operations

- The remainder are implemented in software, by being built up from combinations of hardware instructions
  - Multiple versions implemented by template expansion in the compiler
Summary

- ARM is innovating in embedded GPUs
  - Memory bandwidth, area and power efficiency
  - True performance scalability via multicore design
  - 2x increase in performance every 12 months

- ARM is producing an extensible OpenCL platform
  - ARMv7 CPU and Mali-T600 optimised OpenCL implementations
  - Pluggable software architecture allowing support for other devices to be easily added

- ARM is open to universities and research institutes wishing to work on the opportunities provided by embedded GPU computing!