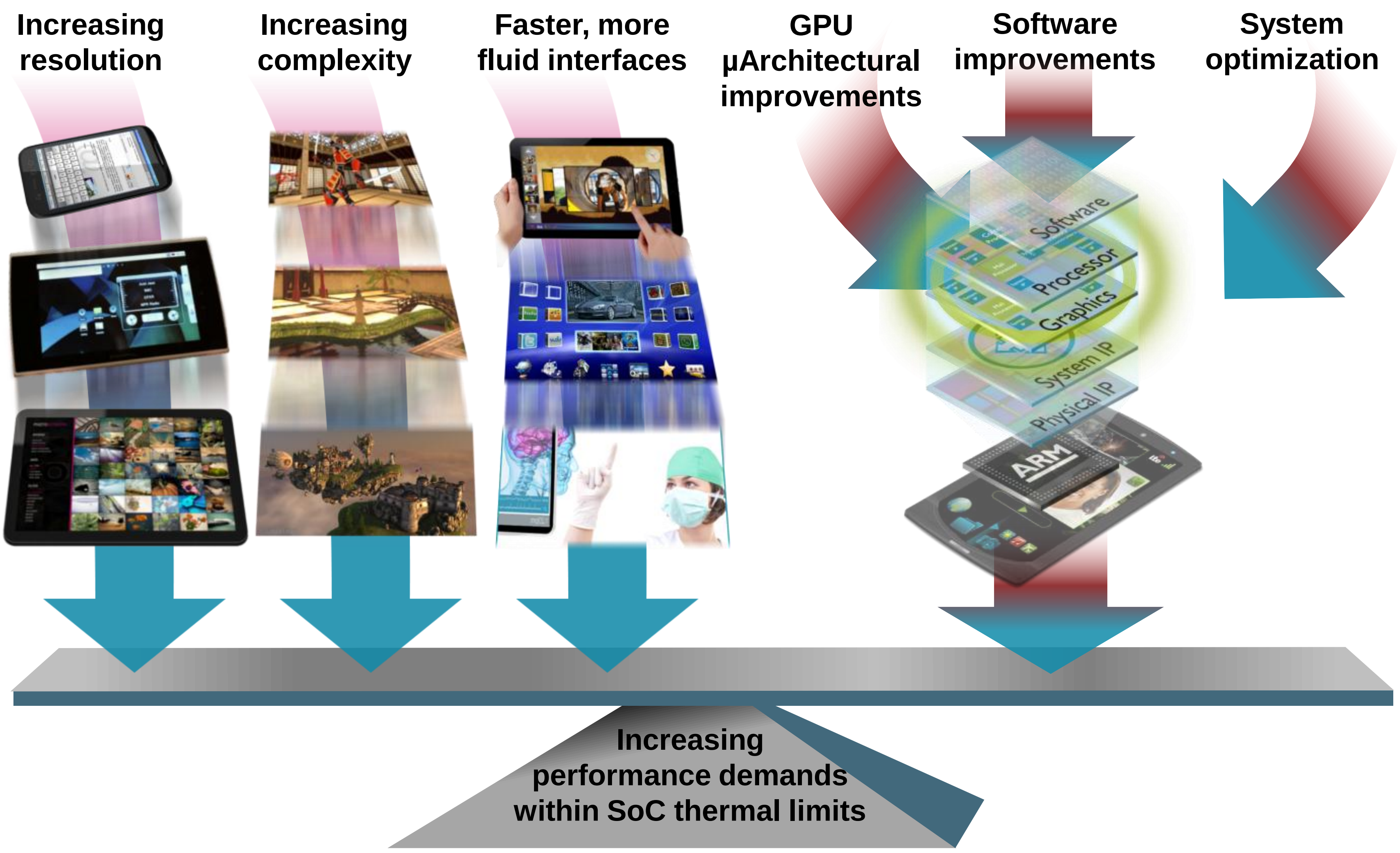
The Future of Mobile Computing

Frode Heggelund, ARM

Emerging Markets

- ARM® Mali™ GPUs are the most widely licensed graphics processor
- Mali deployed in over 230 OEM products
  - Phones, tablets, GPS, DTV, set-top boxes, DVD/Blu-ray players, printers, memory sticks, watches, cloud devices, laptops and more
- Mali GPU gaining market share
  - #1 in graphics enabled DTVs (>70%)
  - >20% Android™ smartphones
  - #1 in Android tablets (>50%)


Solving the Graphics Market Challenges



Key Drivers - Features

- Higher precision
  - Transition from FP16 to FP32 with support for FP64
- Higher resolution
  - Up to 4k x 4k for Mali-400 MP
  - Up to 64k x 64k for Mali-T600 series
- The need for compute
  - But still a market for graphics-only cores
- Better defined arithmetics
  - OpenGL® ES 2.0 says little about how floating point numbers work and how much precision they should have
  - DirectX™ and OpenCL™ are very well defined

Key Drivers - System Power

- Better area density
  - Higher performance per Watt
  - Uniform power dissipation across the chip
- Minimizing bus traffic
  - Better shader performance rather than fill rate – longer shaders give better quality with less bandwidth than increased geometry
  - Transaction elimination – CRC codes eliminates bandwidth for unchanged frames
  - ASTC - Reduction of texture bandwidth while preserving quality
  - Improved caching techniques
- Scalable performance
  - Multicore

Graphics vs. Compute

- Compute typically requires higher precision
  - Full profile OpenCL is a must for the high-end
- Graphics typically require speed

- Converging the two gives interesting results
  - Access to higher-speed arithmetics with lower precision for HPC
    - OpenCL defines similar functions for full precision, at least 10-bit of accuracy and a last one with native precision (which is implementation defined). The lower precision ones can be faster
  - Access to higher precision and more functions for graphics
    - DirectX 11.1 implements lots of compute functions
  - Gives framebuffer artifacts for graphics when rendering old content with full IEEE compliance

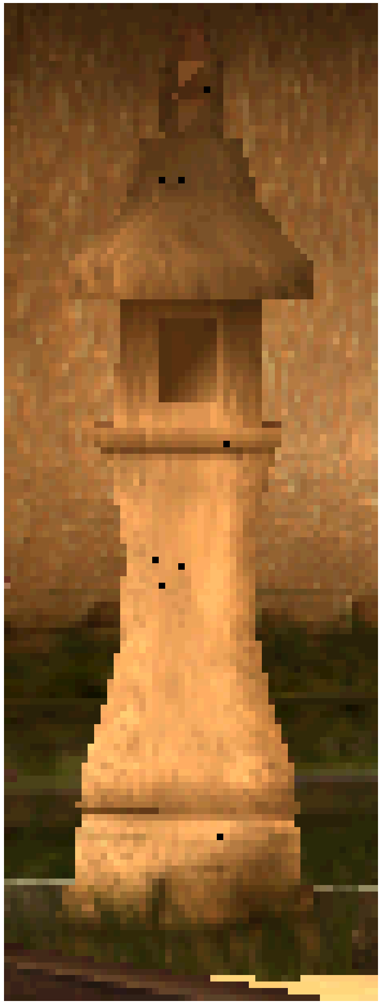
Fun Fact: Compute Introduces Artifacts

- Old mobile content was designed for OpenGL ES 2.0 HW
  - OpenGL ES 2.0 didn’t require full IEEE compliance
  - OpenCL requires full IEEE compliance
- OpenGL ES 2.0 has slightly different behaviour to IEEE compliance
  - sin(inf) = undefined for OpenGL ES 2.0
  - sin(inf) = NaN for OpenCL, so NaN on IEEE compliant cores
- Content may differ if the software driver is not smart enough
  - Example where FP16 goes to inf and give differences between OpenGL ES 2.0 and IEEE compliant hw. FP32 precision works on both cases

- IEEE Compliant, but wrong

- Not IEEE Compliant, but right

NaN is Not a Great Looking Colour



- Software drivers have to make smart decisions when to clamp values according to IEEE or OpenGL ES 2.0 transparently to the application developers