The Bifrost GPU architecture and the ARM Mali-G71 GPU

ARM

Jem Davies
ARM Fellow and VP of Technology

Hot Chips 28
Aug 2016

Introduction to ARM Soft IP

- ARM licenses Soft IP cores (amongst other things) to our Silicon Partners
- They then make chips and sell to OEMs, who sell consumer devices
  - "ARM doesn't make chips"…
- We provide all the RTL, integration testbenches, memories lists, reference floorplans, example synthesis scripts, sometimes models, sometimes FPGA images, sometimes with implementation advice, always with memory system requirements/recommendations
- Consequently silicon area, power, frequencies, performance, benchmark scores can therefore vary quite a bit in real silicon…

ARM

ARM Mali: The world's #1 shipping GPU

140
Total licenses

65
Total licensees

27
new Mali licenses in FY15

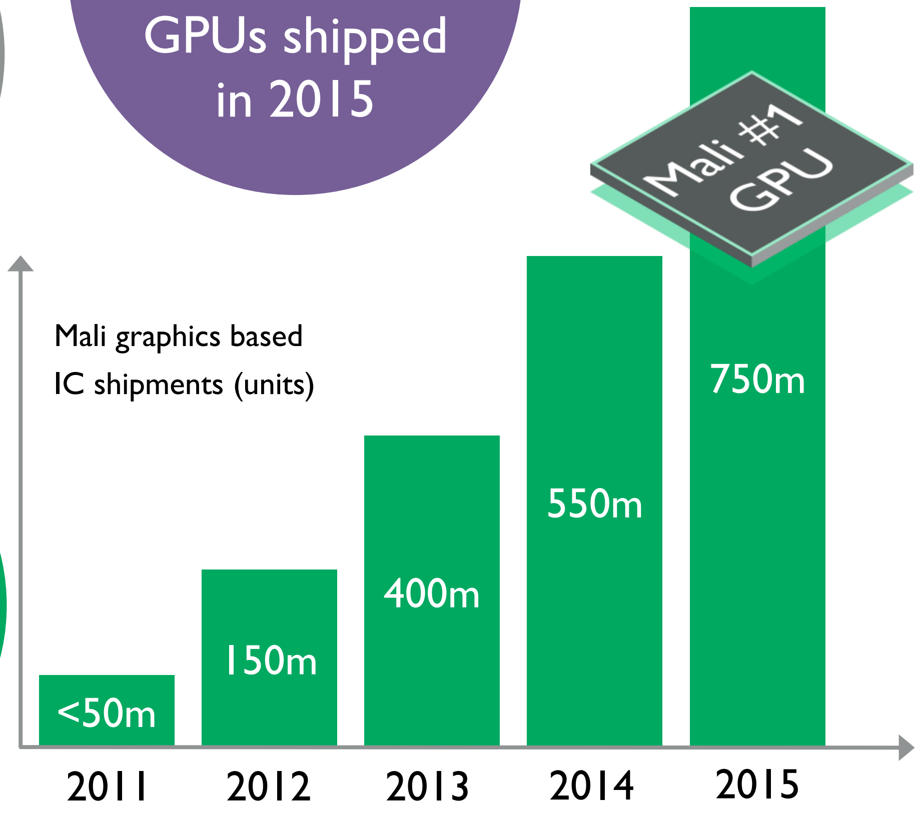
750M
Mali-based GPUs shipped in 2015

Mali is in:

~75%
of DTVs…

~50%
of tablets…

~40%
of smartphones



ARM

ARM Mali graphics processor generations

ARMMALI
Visual Technology

BIFROST

Mali-G71 GPU → …

Unified shader cores, scalar ISA, clause execution, full coherency, Vulkan, OpenCL

MIDGARD

Mali-T600 GPU series → Mali-T700 GPU series → Mali-T800 GPU series

Unified shader cores, SIMD ISA, OpenGL ES 3.x, OpenCL, Vulkan

Presented at HotChips 2015

UTGARD

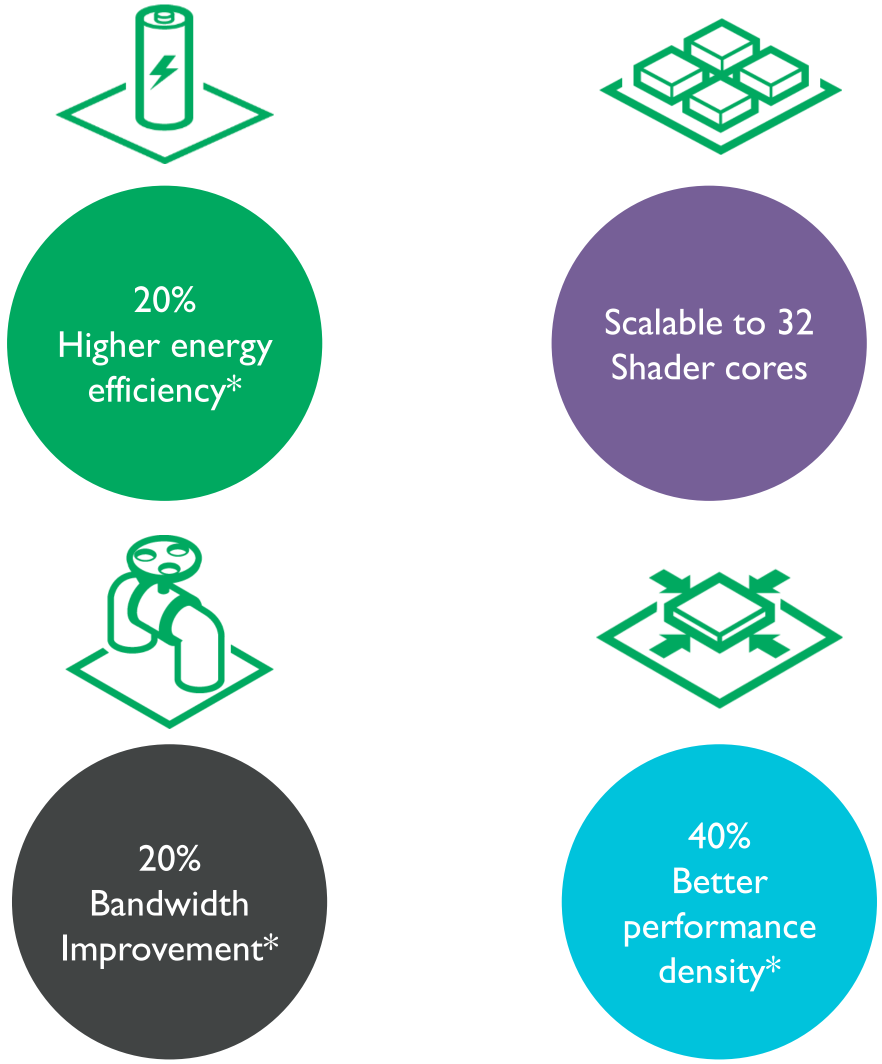
Mali-200 GPU → Mali-300 GPU → Mali-400 GPU → Mali-450 GPU → Mali-470 GPU

Separate shader cores, SIMD ISA, OpenGL ES 2.x

ARM

Bifrost features

- Leverages Mali's scalable architecture
  - Scalable to 32 shader cores
- Major shader core redesign
  - New scalar, clause-based ISA
  - New quad-based arithmetic units
- New geometry data flow
  - Reduces memory bandwidth and footprint
- Support for fine grain buffer sharing with the CPU

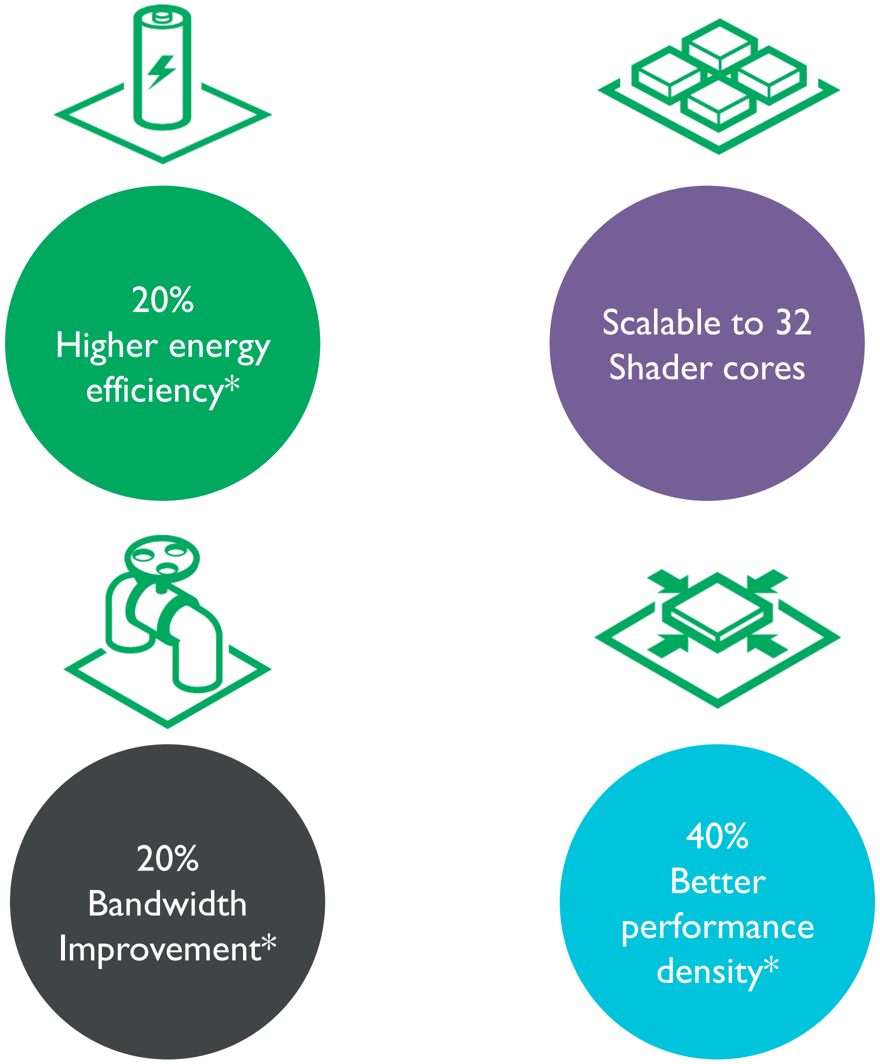


*Compared to Mali-T880 on same process node under the same conditions.

ARM

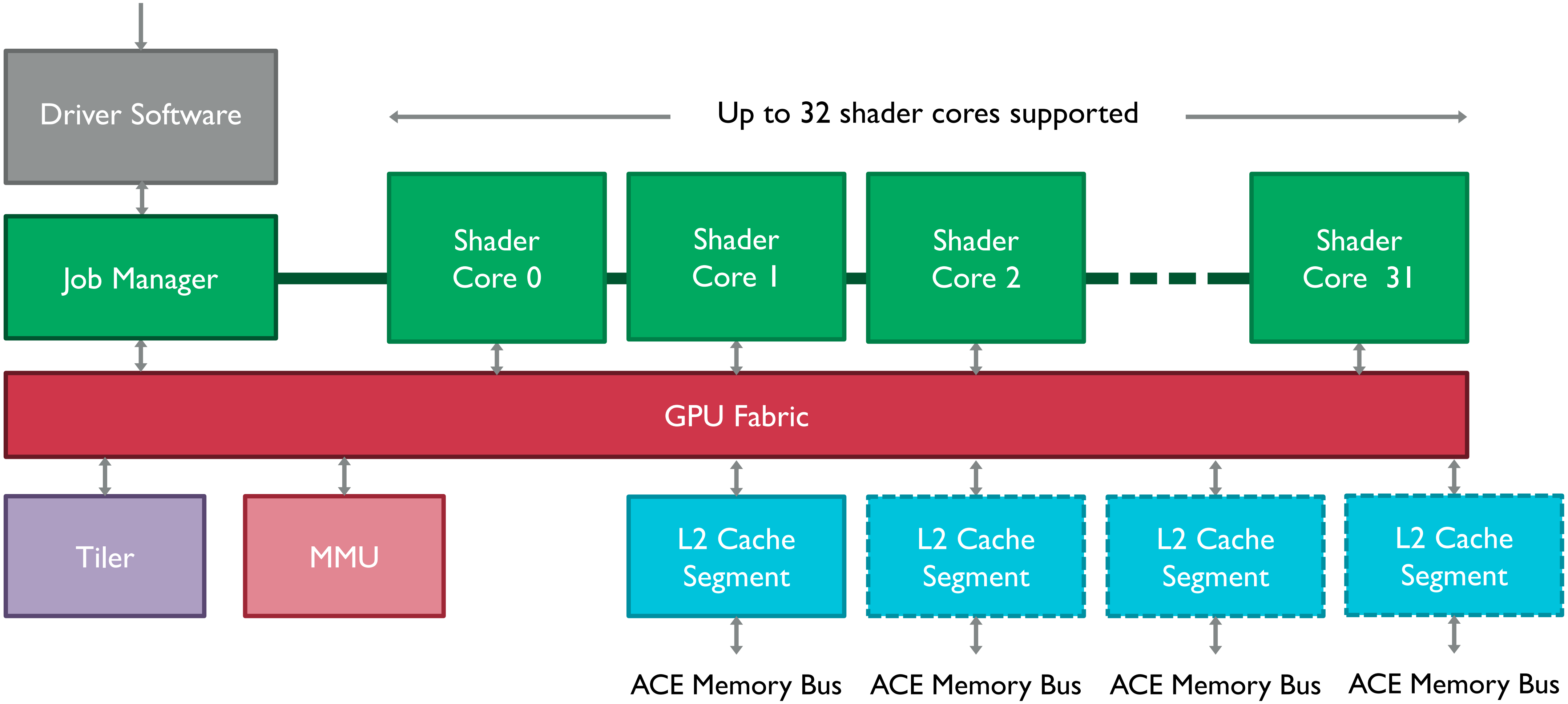
Bifrost features

- Leverages Mali's scalable architecture
  - Scalable to 32 shader cores
- Major shader core redesign
  - New scalar, clause-based ISA
  - New quad-based arithmetic units
- New geometry data flow
  - Reduces memory bandwidth and footprint
- Support for fine grain buffer sharing with the CPU



*Compared to Mali-T880, on same process node under the same conditions.

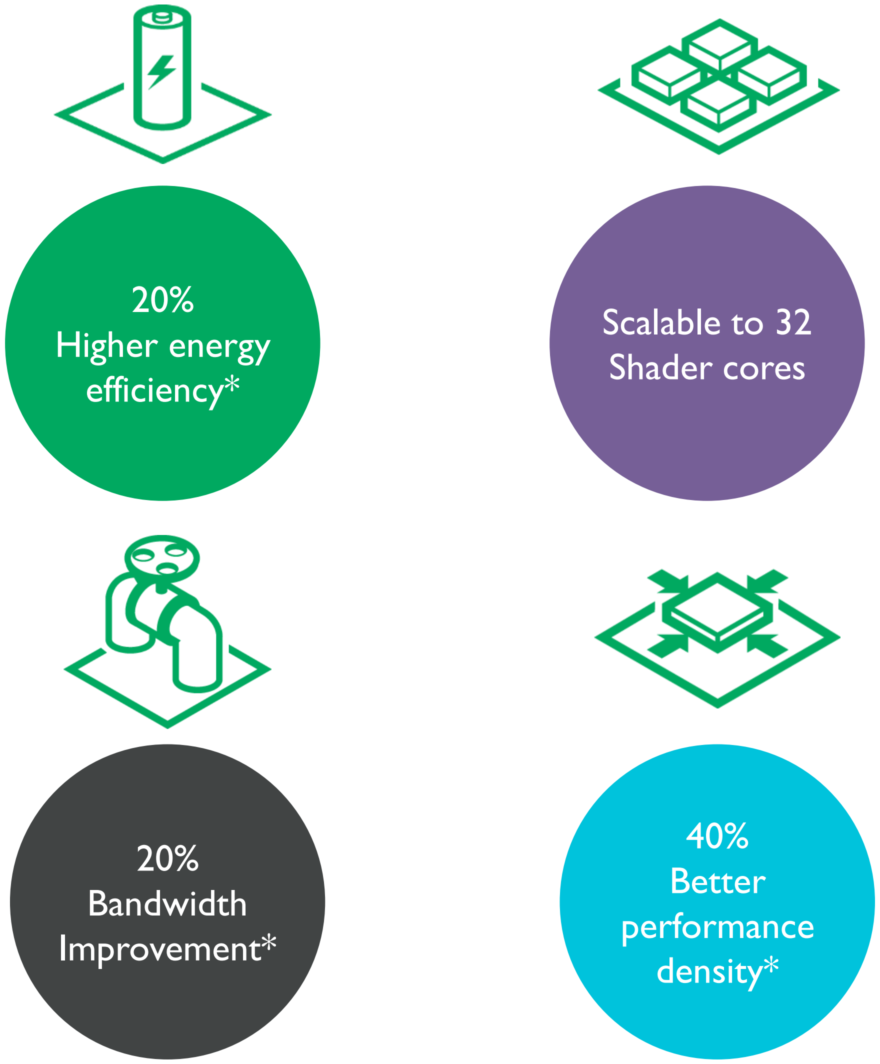
Bifrost GPU design



ARM

Bifrost features

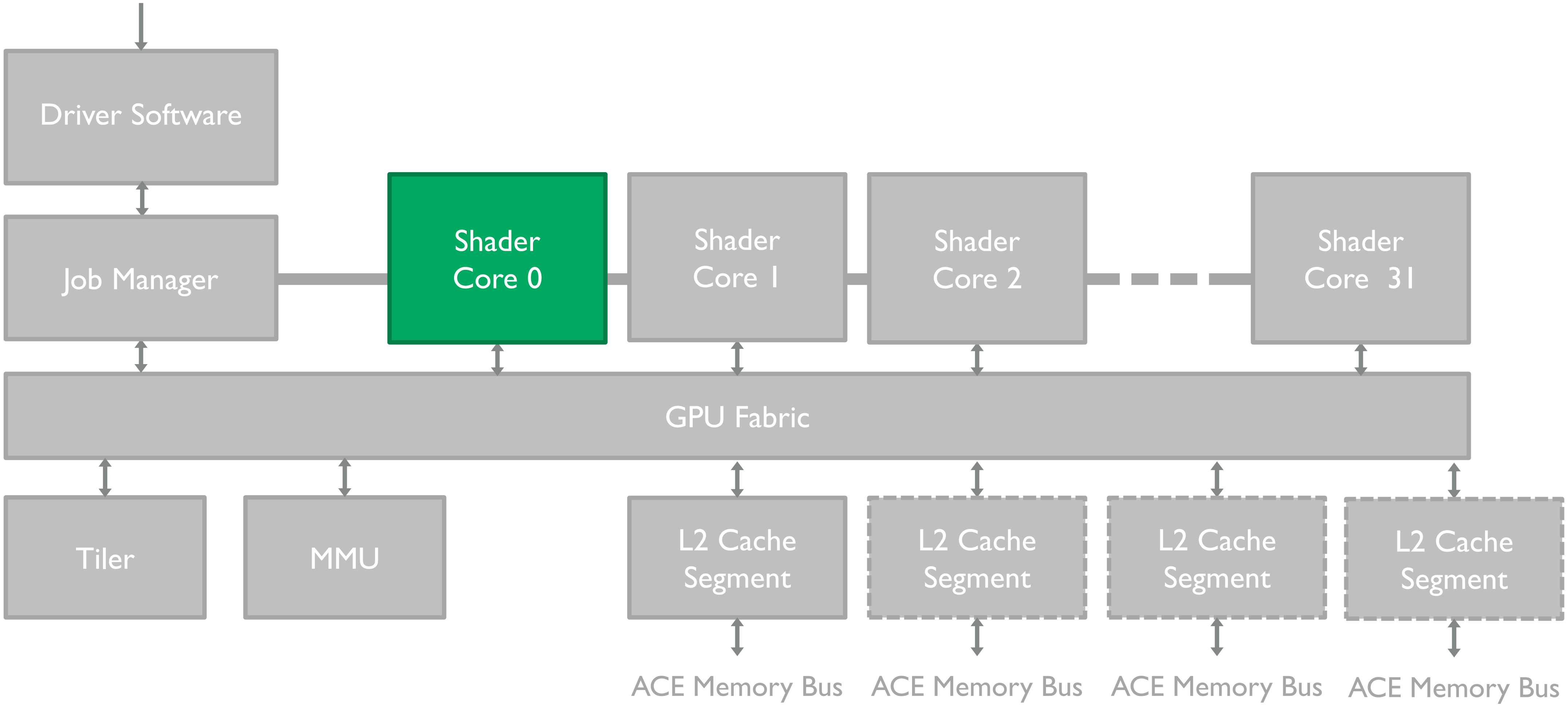
- Leverages Mali's scalable architecture
  - Scalable to 32 shader cores
- Major shader core redesign
  - New quad-based arithmetic units
  - New scalar, clause-based ISA
- New geometry data flow
  - Reduces memory bandwidth and footprint
- Support for fine grain buffer sharing with the CPU



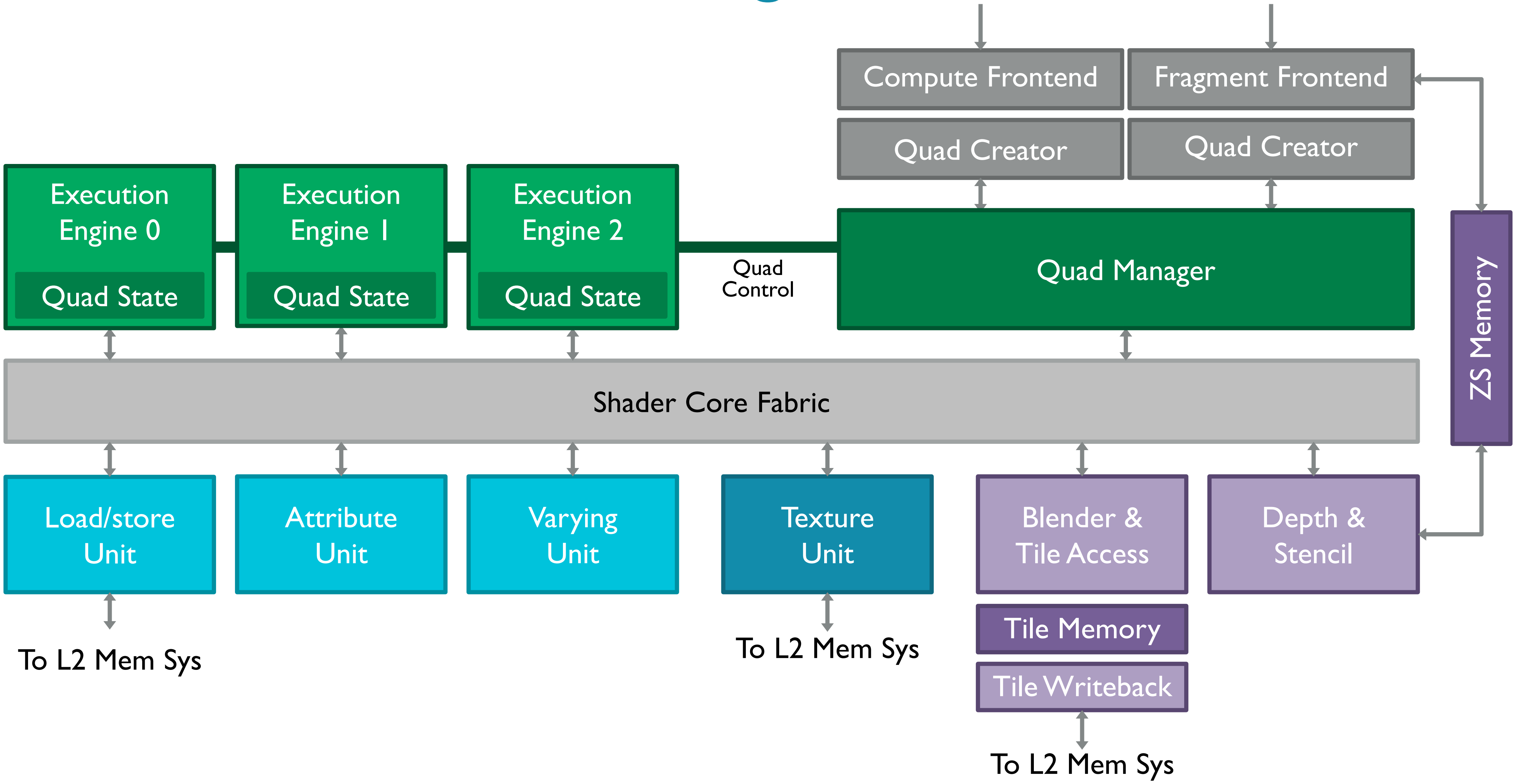
*Compared to Mali-T880, on same process node under the same conditions.

ARM

Shader core improvements

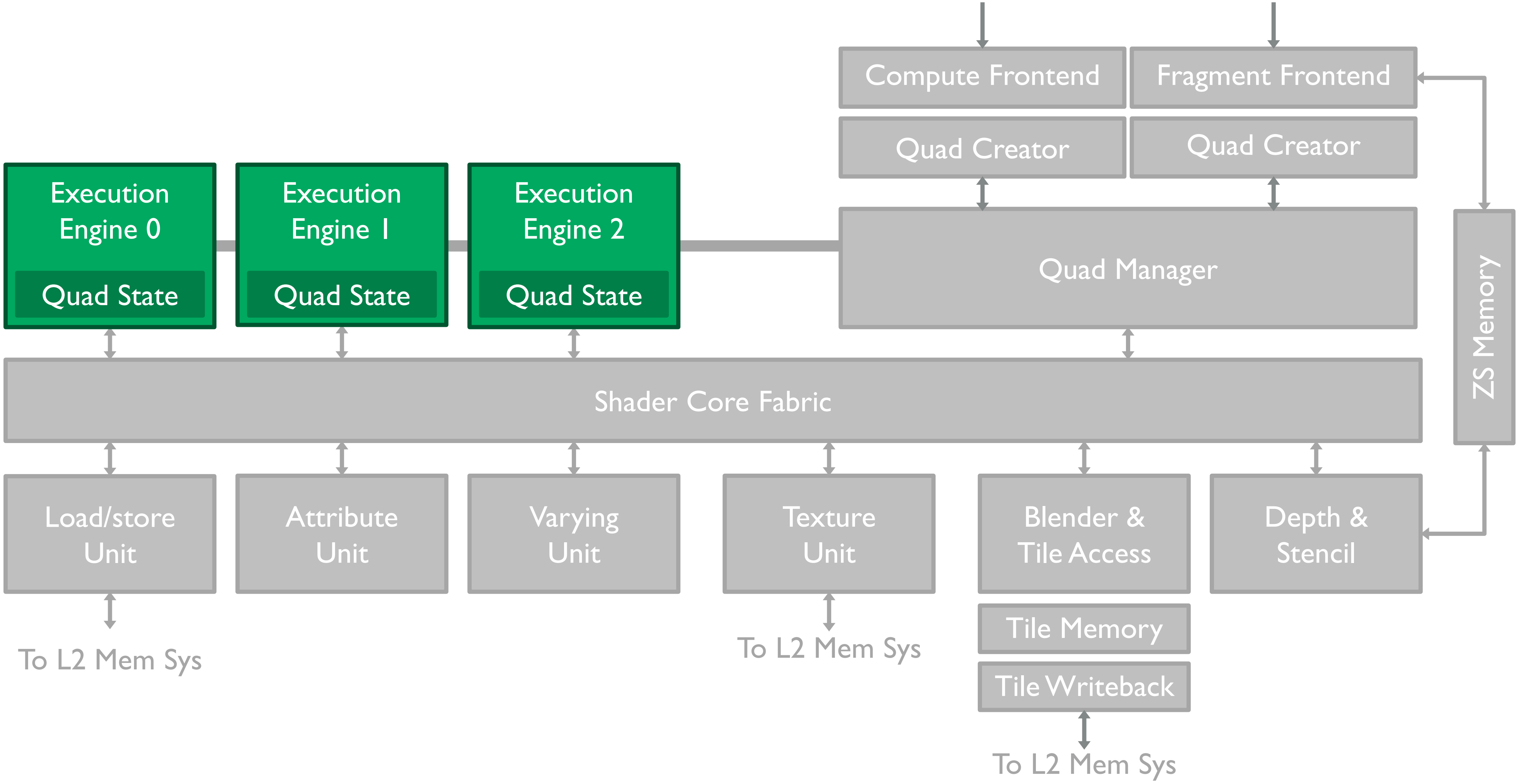


Mali-G71 shader core design



ARM

Quad execution



ARM

Recap: SIMD vectorization

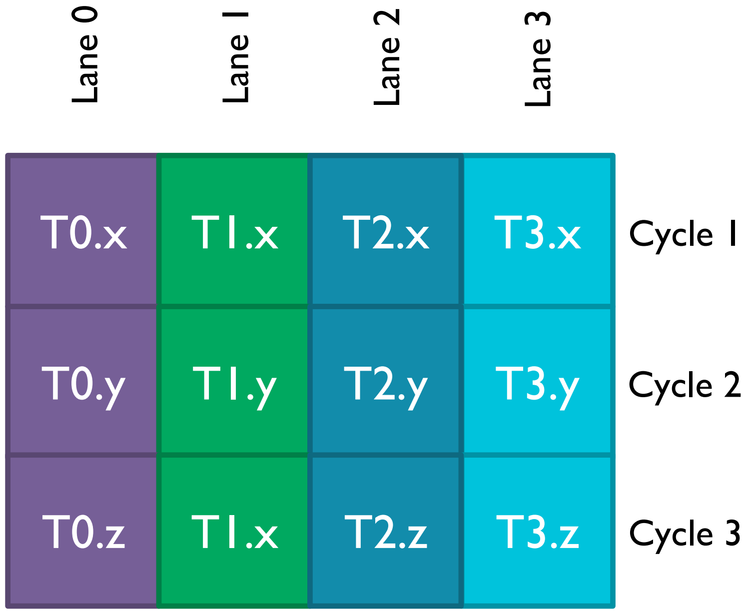
- Midgard GPUs use SIMD vectorization
  - One thread at a time executes in each pipeline stage
  - Each thread must fill the width of the hardware

	Lane 0	Lane 1	Lane 2	Lane 3
Cycle 1	T0.x	T0.y	T0.z	Idle
Cycle 2	T1.x	T1.y	T1.z	Idle
Cycle 3	T2.x	T2.y	T2.z	Idle
Cycle 4	T3.x	T3.y	T3.z	Idle

- Sensitive to shader code
  - Code always evolving
  - Compiler vectorization is not perfect
  - Have to detect combinations of operations which can be merged to fill idle lanes
  - Scalar operations can not always be merged into vectors

Quad vectorization

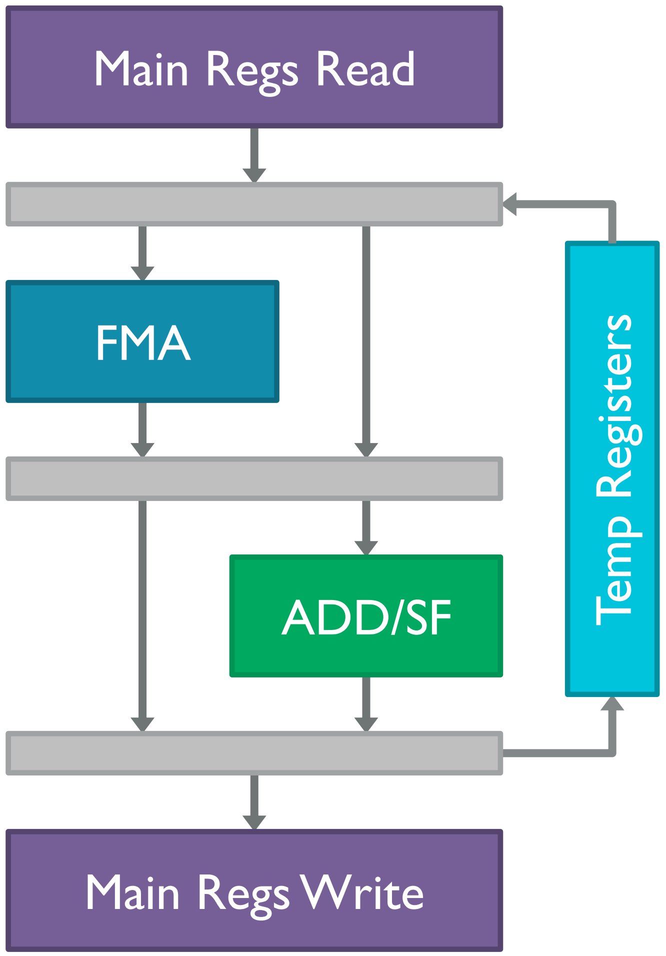
- Bifrost uses quad-parallel execution
  - Four scalar threads executed in lockstep in a "quad"
  - One quad at a time executes in each pipeline stage
  - Each thread fills one 32-bit lane of the hardware
  - 4 threads doing a vec3 FP32 add takes 3 cycles
  - Improves utilization



- Quad vectorization is compiler friendly
  - Each thread only sees a stream of scalar operations
  - Vector operations can *always* be split into scalars

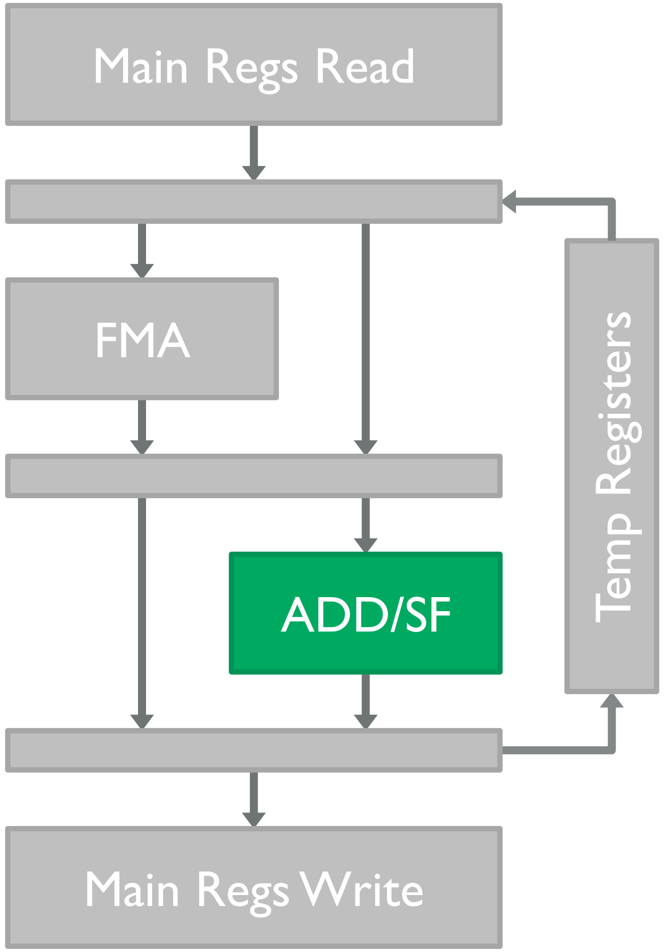
Bifrost execution engine

- Executes quad-parallel scalar operations
  - 4x32-bit multiplier FMA
  - 4x32-bit adder ADD
  - Adder includes special function unit
- Smaller and more area efficient
- Simplified layout eases compilation
  - Better scheduling in today's code
  - Better utilization
- One instruction word contains two instructions



Bifrost execution engine: Special arithmetic ops

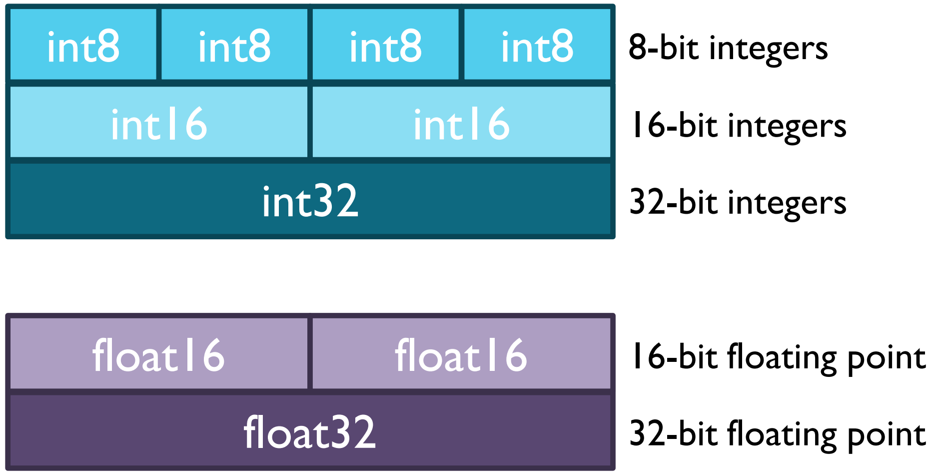
- Special function hardware is smaller than Midgard VLUT equivalent
  - Many transcendental functions supported
  - Special functions provide building blocks for compiled shader code
  - Part of the built-in function libraries

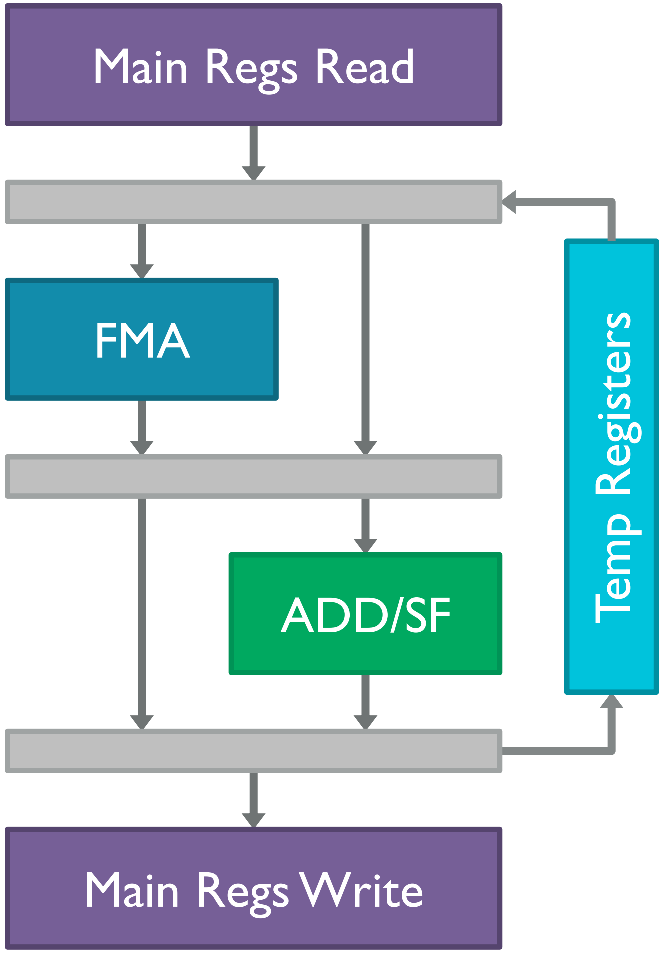


ARM

Bifrost execution engine: Functional units

- Retains support for smaller width data types
  - 2x performance for FP16 useful for pixel shaders

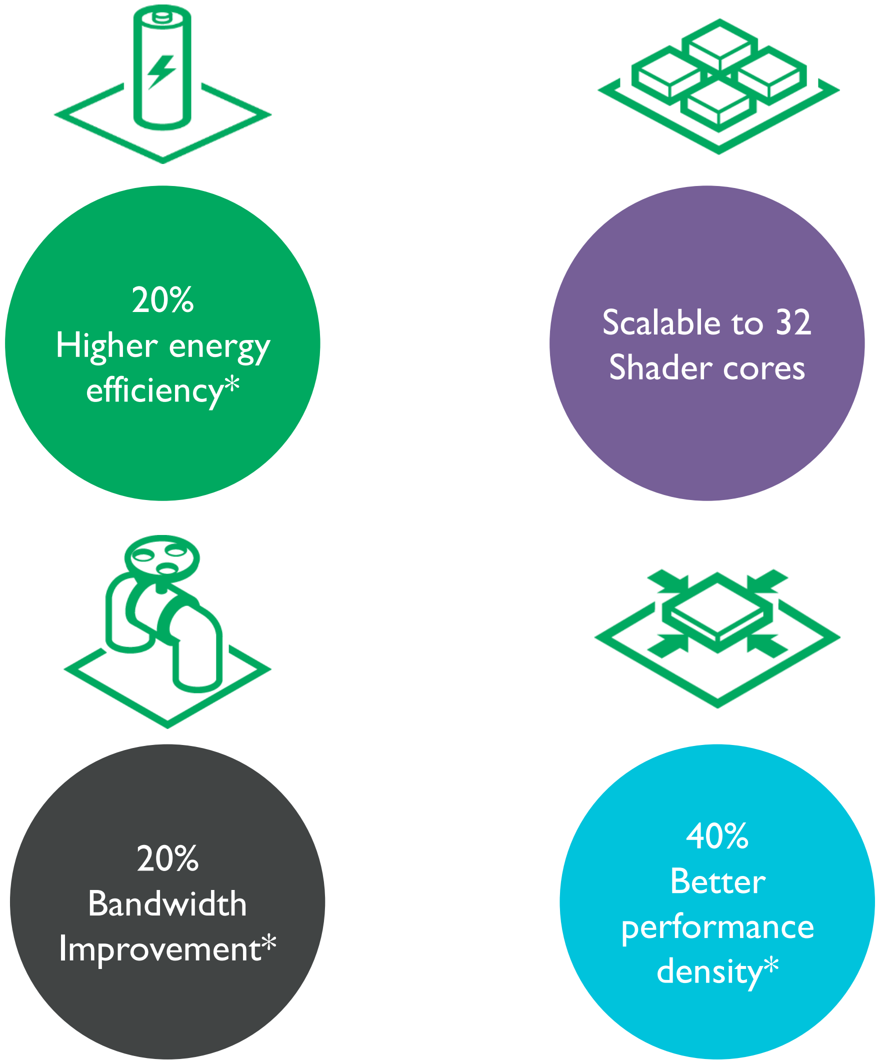




ARM

Bifrost features

- Leverages Mali's scalable architecture
  - Scalable to 32 shader cores
- Major shader core redesign
  - New quad-based arithmetic units
  - New scalar, clause-based ISA
- New geometry data flow
  - Reduces memory bandwidth and footprint
- Support for fine grain buffer sharing with the CPU



*Compared to Mali-T880, on same process node under the same conditions.

Clause execution

- A group of instructions which executes atomically
- Architectural state visible after clause completion
- Bypass path registers exposed to the compiler
- Non-deterministic instructions on clause boundaries

Bifrost shader clause example

Linear Source

```
LOAD.32  r0, [r10]
FADD.32  r1,r0,r0
FADD.32  r2,r1,r1
FADD.32  r3,r2,r2
FADD.32  r4,r3,r3
FADD.32  r3,r3,r4
FADD.32  r0,r3,r3
STORE.32 r0, [r10]
```

Bifrost shader clause example

Linear Source

```
LOAD.32   r0, [r10]
FADD.32   r1,r0,r0
FADD.32   r2,r1,r1
FADD.32   r3,r2,r2
FADD.32   r4,r3,r3
FADD.32   r3,r3,r4
FADD.32   r0,r3,r3
STORE.32 r0, [r10]
```

Load is variable length, so clause must be split from use of r0

Next clause uses r0, so must wait for load to complete

Basic Clause Compile

```
{
LOAD.32   r0, [r10]
}
wait(load) {
FADD.32   r1,r0,r0
FADD.32   r2,r1,r1
FADD.32   r3,r2,r2
FADD.32   r4,r3,r3
FADD.32   r3,r3,r4
FADD.32   r0,r3,r3
STORE.32 r0, [r10]
}
```


ARM

Bifrost shader clause example

Linear Source

```
LOAD.32   r0, [r10]
FADD.32   r1,r0,r0
FADD.32   r2,r1,r1
FADD.32   r3,r2,r2
FADD.32   r4,r3,r3
FADD.32   r3,r3,r4
FADD.32   r0,r3,r3
STORE.32 r0, [r10]
```

Basic Clause Compile

Load is variable length, so clause must be split from use of r0

```
{
LOAD.32   r0, [r10]
}
```

Next clause uses r0, so must wait for load to complete

```
wait(load) {
FADD.32   r1,r0,r0
FADD.32   r2,r1,r1
FADD.32   r3,r2,r2
FADD.32   r4,r3,r3
FADD.32   r3,r3,r4
FADD.32   r0,r3,r3
STORE.32 r0, [r10]
}
```

Optimized Clause Compile

```
{
LOAD.32   r0, [r10]
}
```

Use temporaries: only 2 register file accesses in 6 cycles

```
wait(load) {
FADD.32   t0,r0,r0
FADD.32   t1,t0,t0
FADD.32   t0,t1,t1
FADD.32   t1,t0,t0
FADD.32   t0,t1,t1
FADD.32   r0,t0,t0
}
```

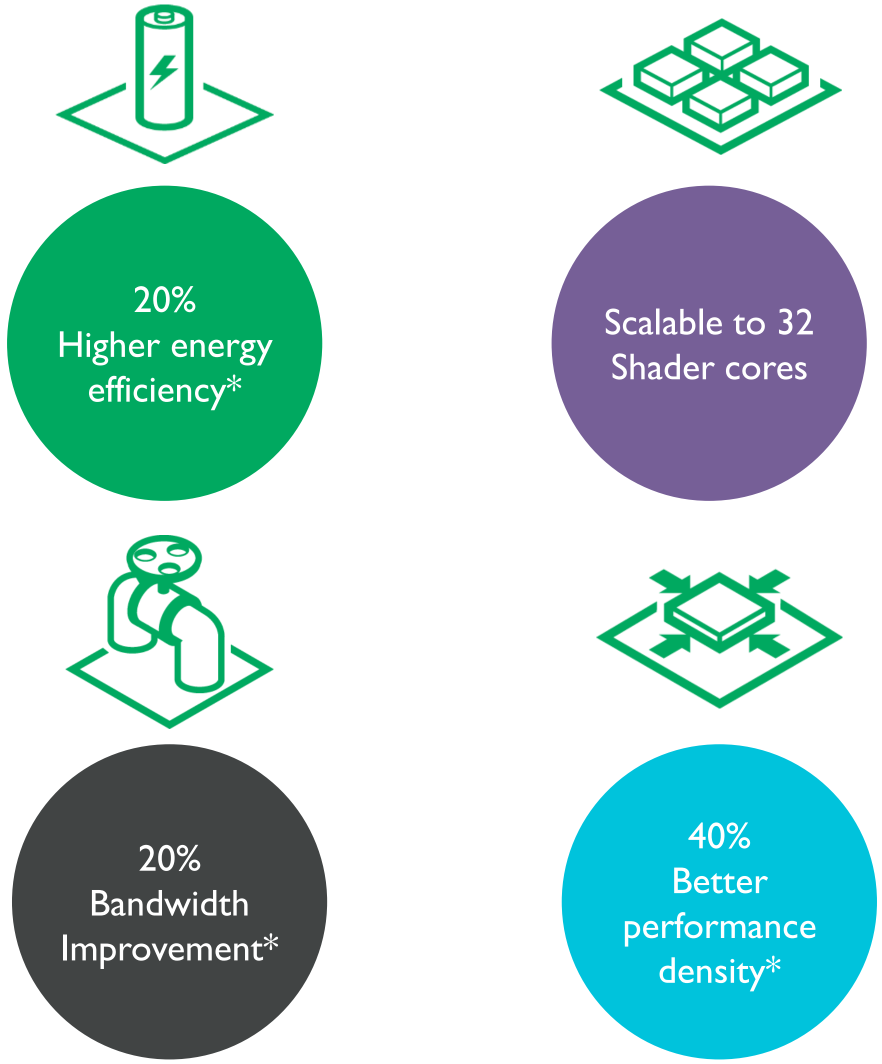
Store may also stall, so split to help scheduling

```
{
STORE.32 r0, [r10]
}
```


ARM

Bifrost features

- Leverages Mali's scalable architecture
  - Scalable to 32 shader cores
- Major shader core redesign
  - New scalar, clause-based ISA
  - New quad-based arithmetic units
- New geometry data flow
  - Reduces memory bandwidth and footprint
- Support for fine grain buffer sharing with the CPU

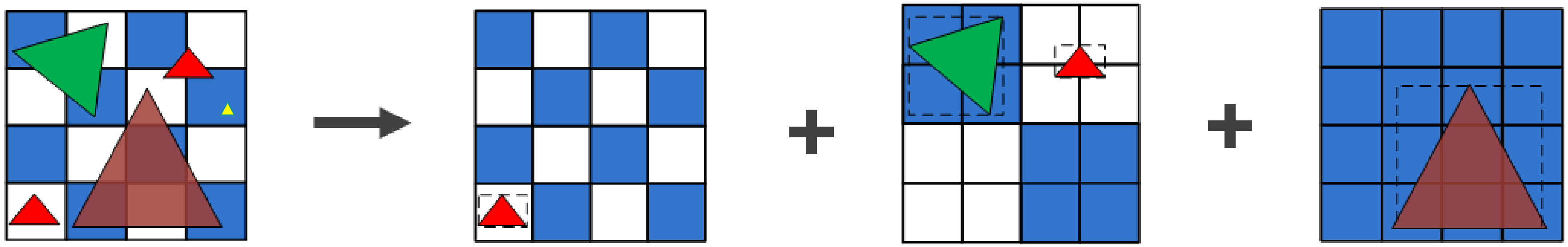


*Compared to Mali-T880, on same process node under the same conditions.

ARM

Tiler changes

- Bifrost uses the same underlying hierarchical binning design as Midgard

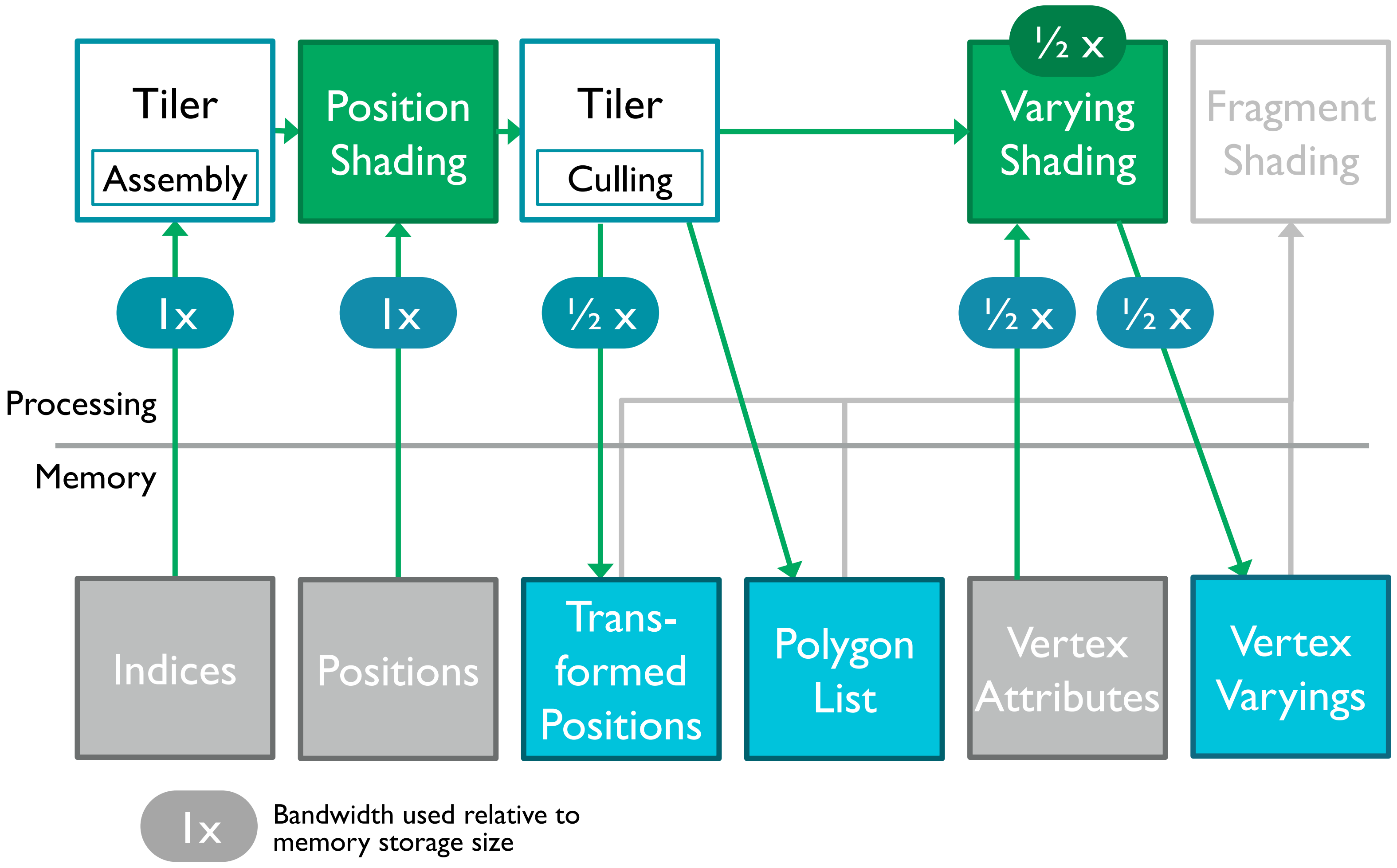


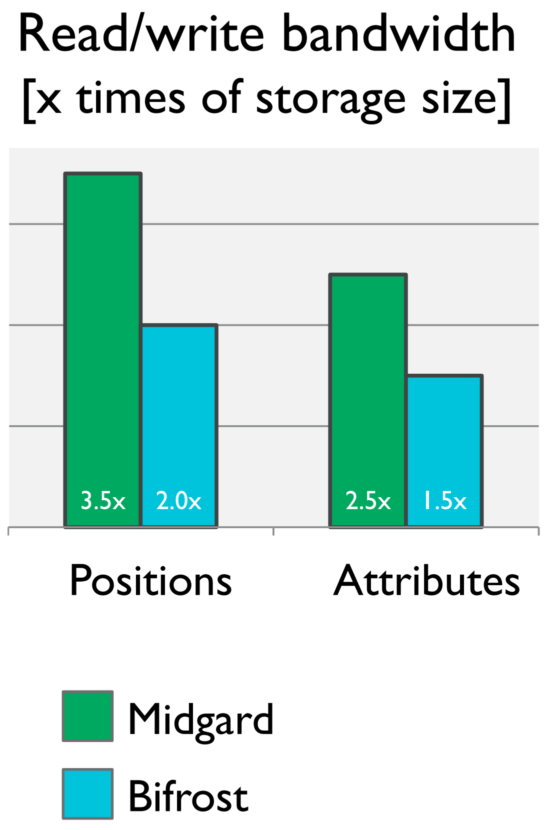
- Significantly redesigned tiler memory structures
  - Minimum buffer allocations eliminated
  - Buffer allocation granularity now finer
  - Micro-triangle elimination reduces the number of primitives stored in bin buffers for geometry-dense scenes

- Cumulative effect of all changes is up to 95% reduction in tiler memory footprint

ARM

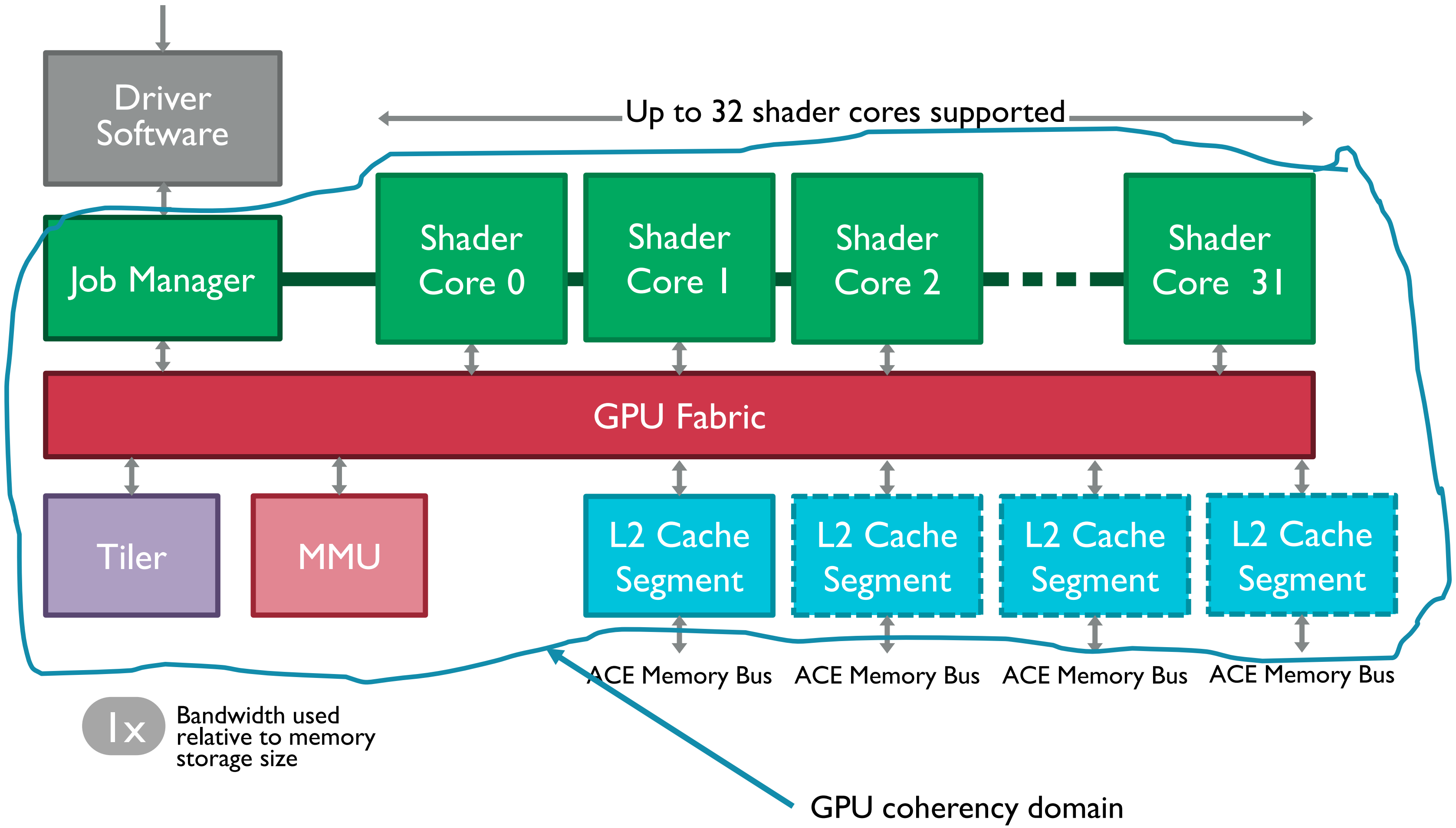
Index-driven position shading





ARM

Index-driven position shading

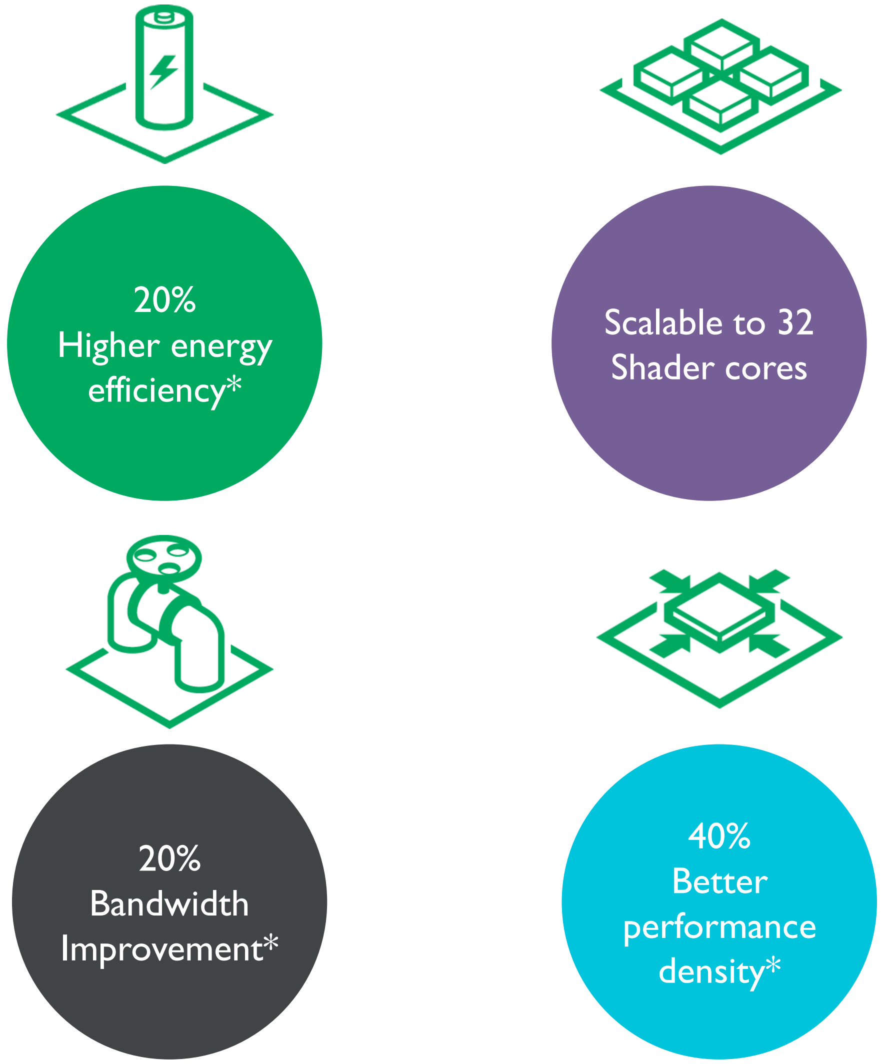


- Leverages existing coherency flows
- Multiple shader cores write transformed positions into a shared memory fifo.
- The fixed function Tiler reads the transformed positions directly via shared memory reads. No manual flushing required (fifo values are most likely resident in the L2C, but don't have to be)
- Once the tiler has read the positions, they are no longer needed and may be discarded

ARM

Bifrost features

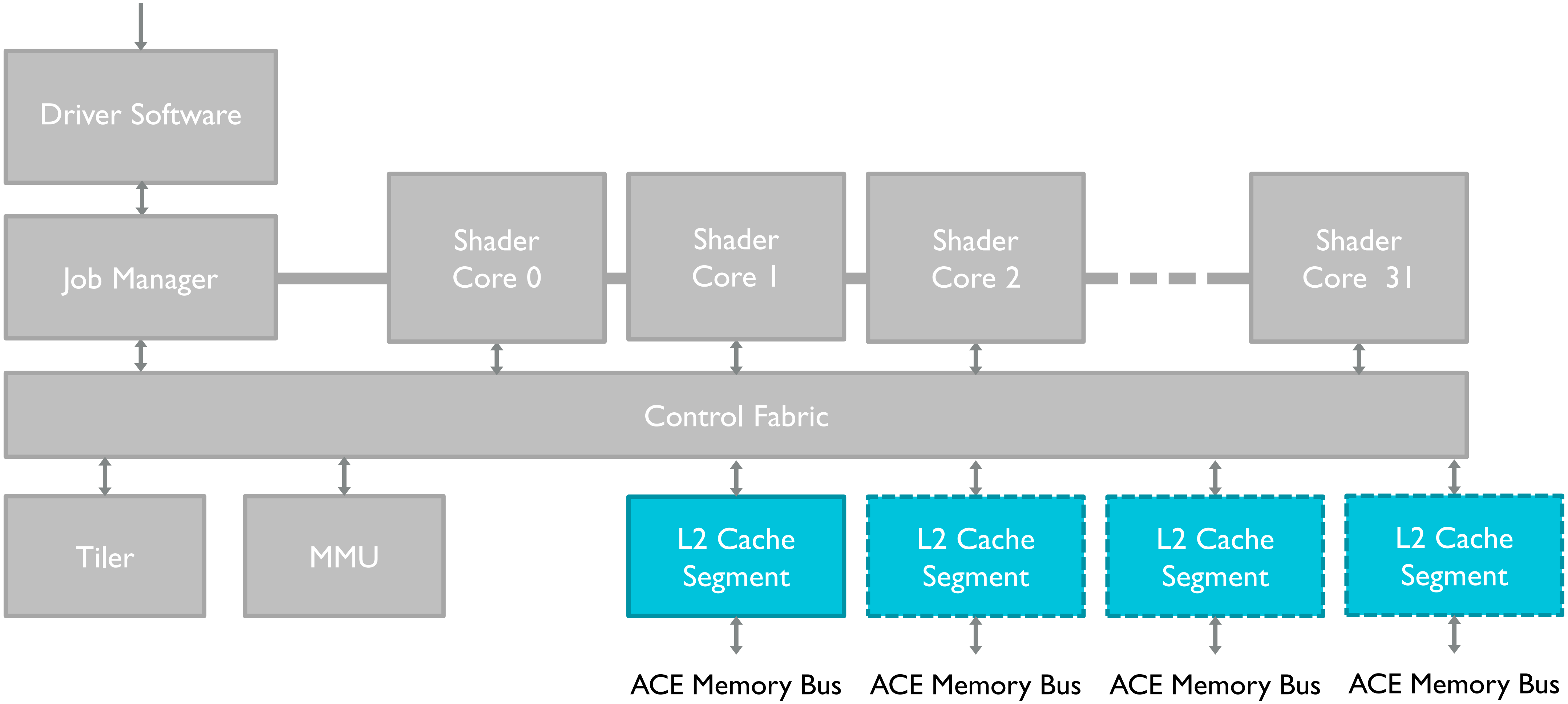
- Leverages Mali's scalable architecture
  - Scalable to 32 shader cores
- Major shader core redesign
  - New scalar, clause-based ISA
  - New quad-based arithmetic units
- New geometry data flow
  - Reduces memory bandwidth and footprint
- Support for fine grain buffer sharing with the CPU



*Compared to Mali-T880, on same process node under the same conditions.

ARM

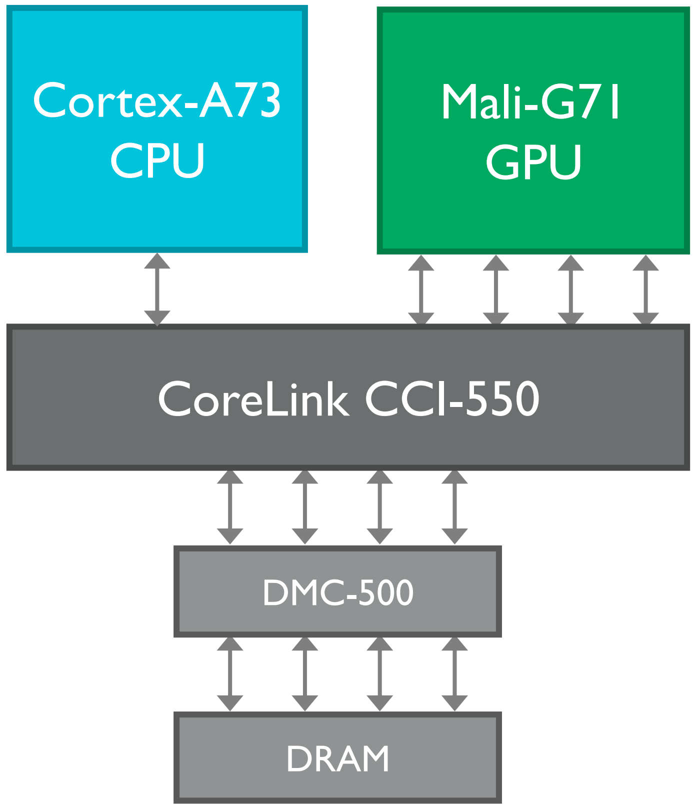
Memory system



Full coherency using AMBA ACE protocol

Memory system

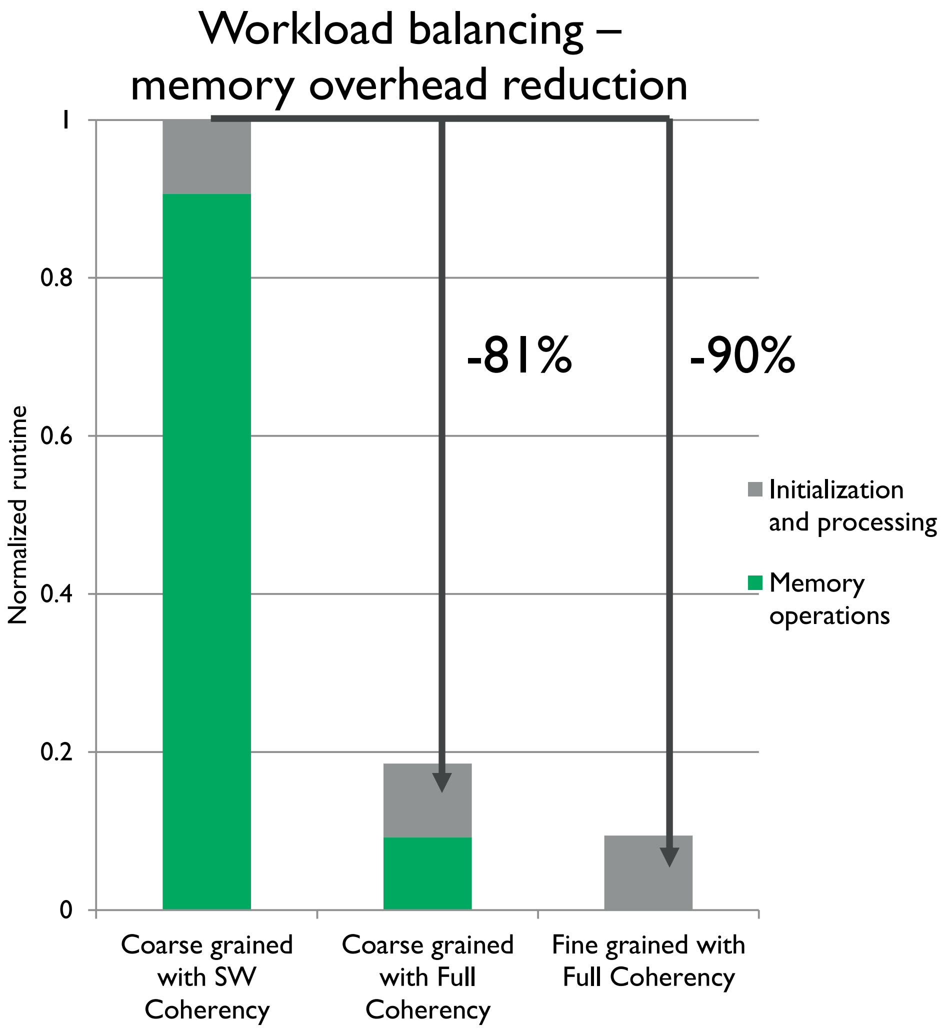
- Full system coherency support
  - Supports tightly coupled CPU+GPU use cases
- L2 cache improvements
  - Single logical L2 cache makes software easier
  - Fewer partial lines written to memory which improves LPDDR4 performance
- Supports TrustZone



ARM

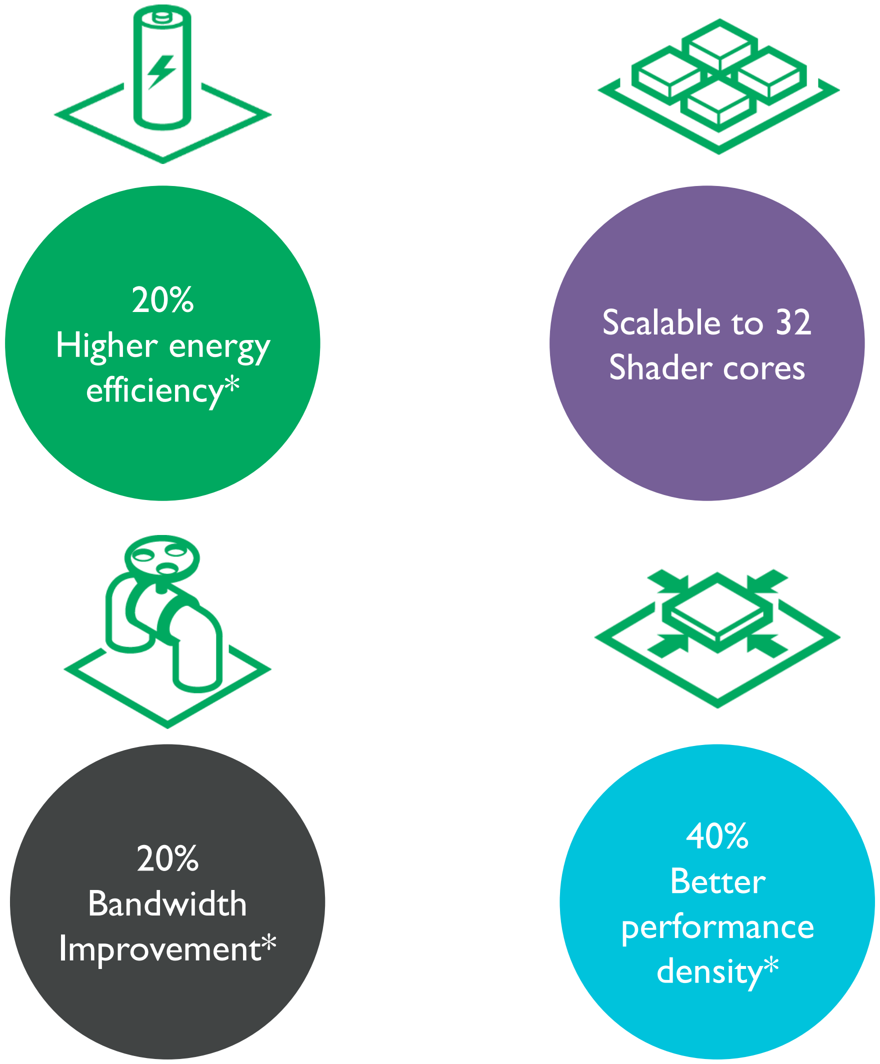
Next-generation heterogeneous computing

- OpenCL 2.0 Introduces Shared Virtual Memory
- Mali-G71 goes one step further with fine grained buffers
- Significantly eases development and enables truly heterogeneous use case



Summary

- Leverages Mali's scalable architecture
  - Scalable to 32 shader cores
- Major shader core redesign
  - New scalar, clause-based ISA
  - New quad-based arithmetic units
- New geometry data flow
  - Reduces memory bandwidth and footprint
- Support for fine grain buffer sharing with the CPU



*Compared to Mali-T880, on same process node under the same conditions.

ARM

Thank you!

ARM