Bifrost - The GPU architecture for next five billion

ARM

Hessed Choi
Senior FAE / ARM

ARM Tech Forum
June 28th, 2016

Vulkan

What is Vulkan?

- A 3D graphics API for the next twenty years
  - Logical successor to OpenGL and OpenGL ES
  - Modern, efficient design
  - Open, industry-controlled standard
- Here, now
  - Released in February, with unprecedented support
  - Available today for desktop Windows and Linux
  - Officially supported in Android™ 'N'
  - Shipping today in Samsung Galaxy S7
- Engaged, active developer community

Why ARM loves Vulkan

- A great fit for mobile graphics architectures!
  - No wasted effort trying to look like a desktop GPU
  - Designed to enable mobile-specific optimizations
- Radical commitment to efficiency
  - CPU load is greatly reduced, even on a single core
- Makes your multi-core CPU more useful!
  - Driver work can be distributed across many threads
  - This helps performance *and* power
- Makes your multi-core GPU more useful too
  - Easier for applications to keep a powerful GPU busy

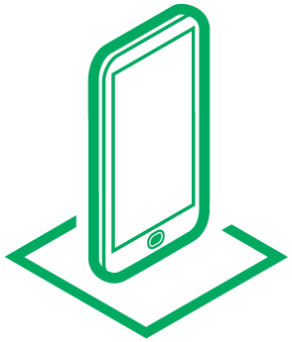
Bifrost

Bifrost: The new GPU architecture

The increasing pixel impact of modern mobile gaming continues to drive innovation

2010: Utgard

2013: Midgard



2016: Bifrost

ARM Mali processor generations

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

UTGARD

Mali-200 GPU

Mali-300 GPU

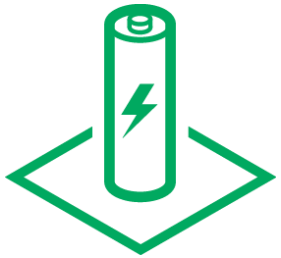
Mali-400 GPU

Mali-450 GPU

Mali-470 GPU

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

Mali-G71 efficiency drives performance



20%
Higher energy efficiency*

32
Shader cores

40%
Better performance density*

20%
Bandwidth Improvement*

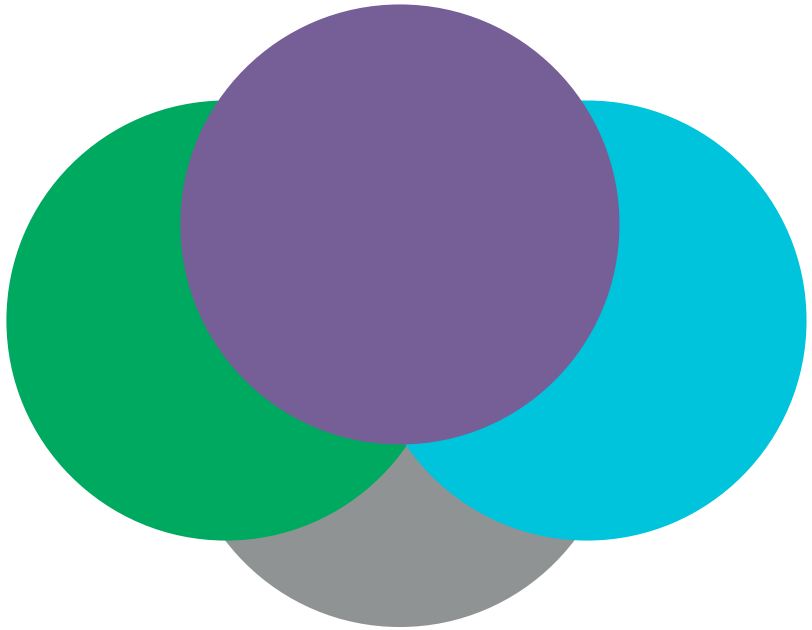
Optimized for next generation, advanced, real-world content

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

ARM

Bifrost features

- A more efficient architecture:
  - More performance overall, per mm^2 and per line of *real world* shader code
- Major shader core redesign
  - New scalar, clause-based ISA
  - New quad-based arithmetic units
  - New core fabric
- New geometry data flow
  - Reduces memory bandwidth and footprint



1.5x

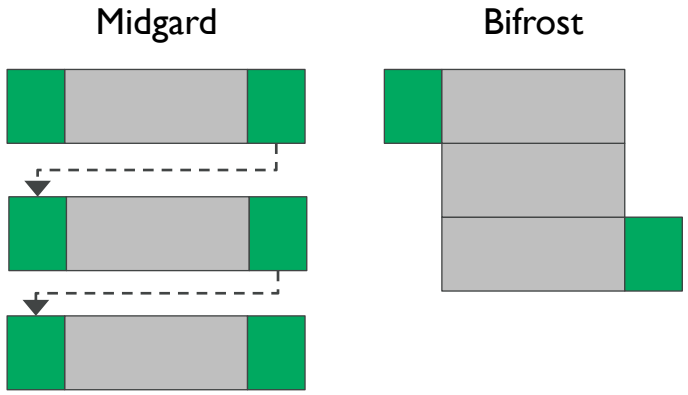
Performance improvement

Architectural innovations

Bifrost architectural innovations

Energy efficiency

- Claused shaders
- Index Driven Vertex Shading
- Wire light pipelines



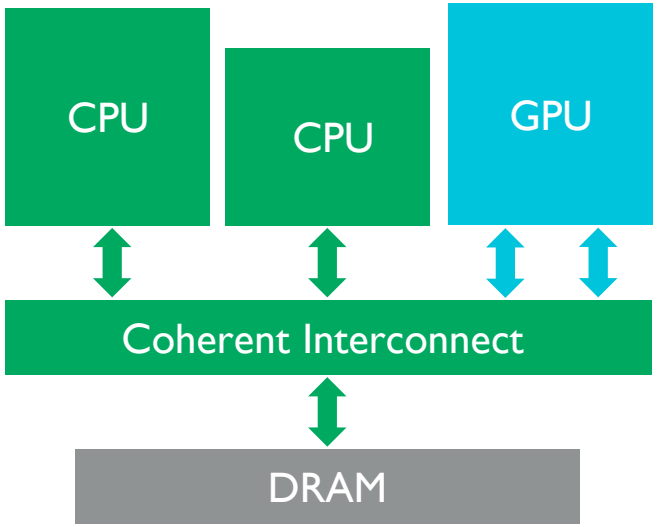
Developer friendly

- Designed for Vulkan and VR/AR

Vulkan™

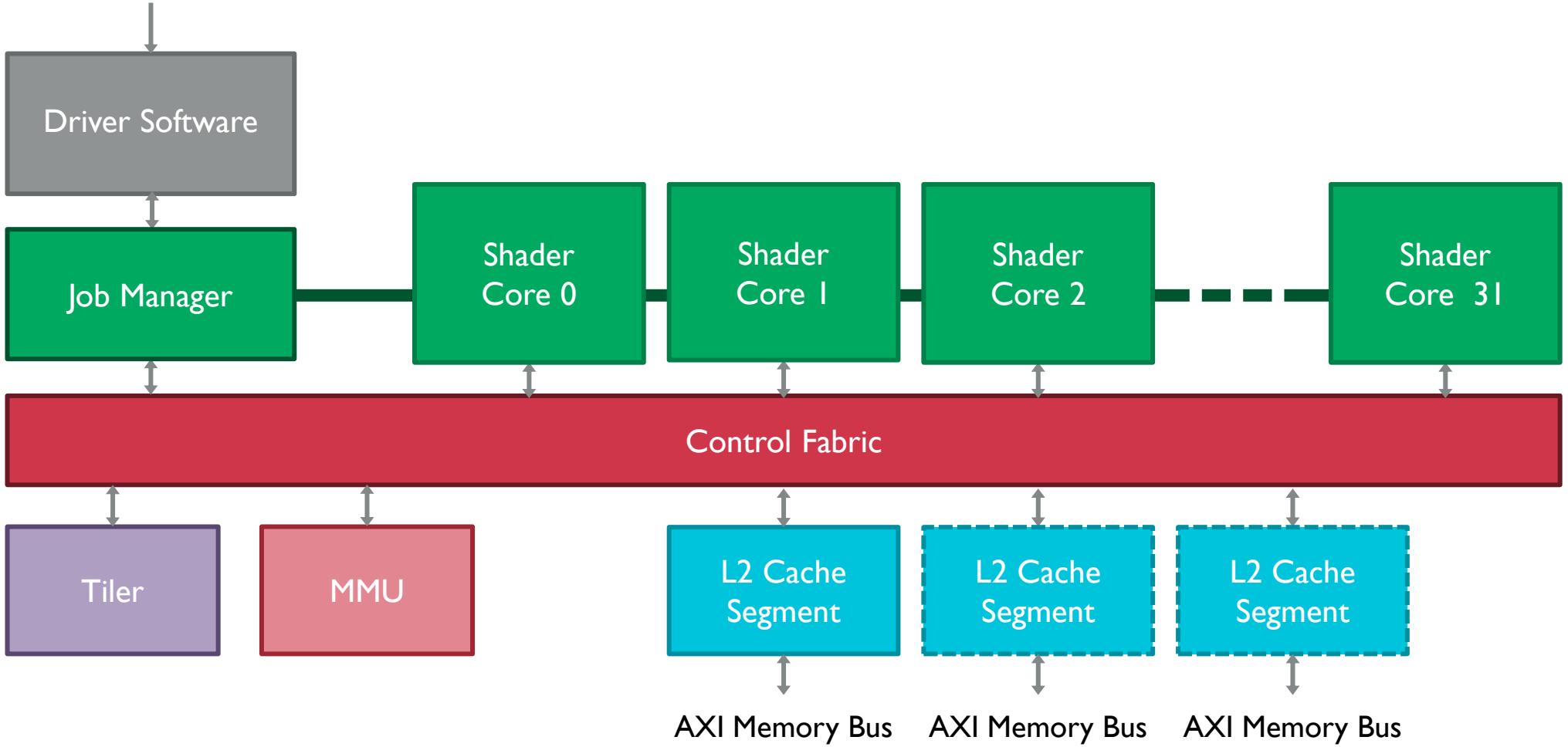
Heterogeneous computing

- Full system coherency

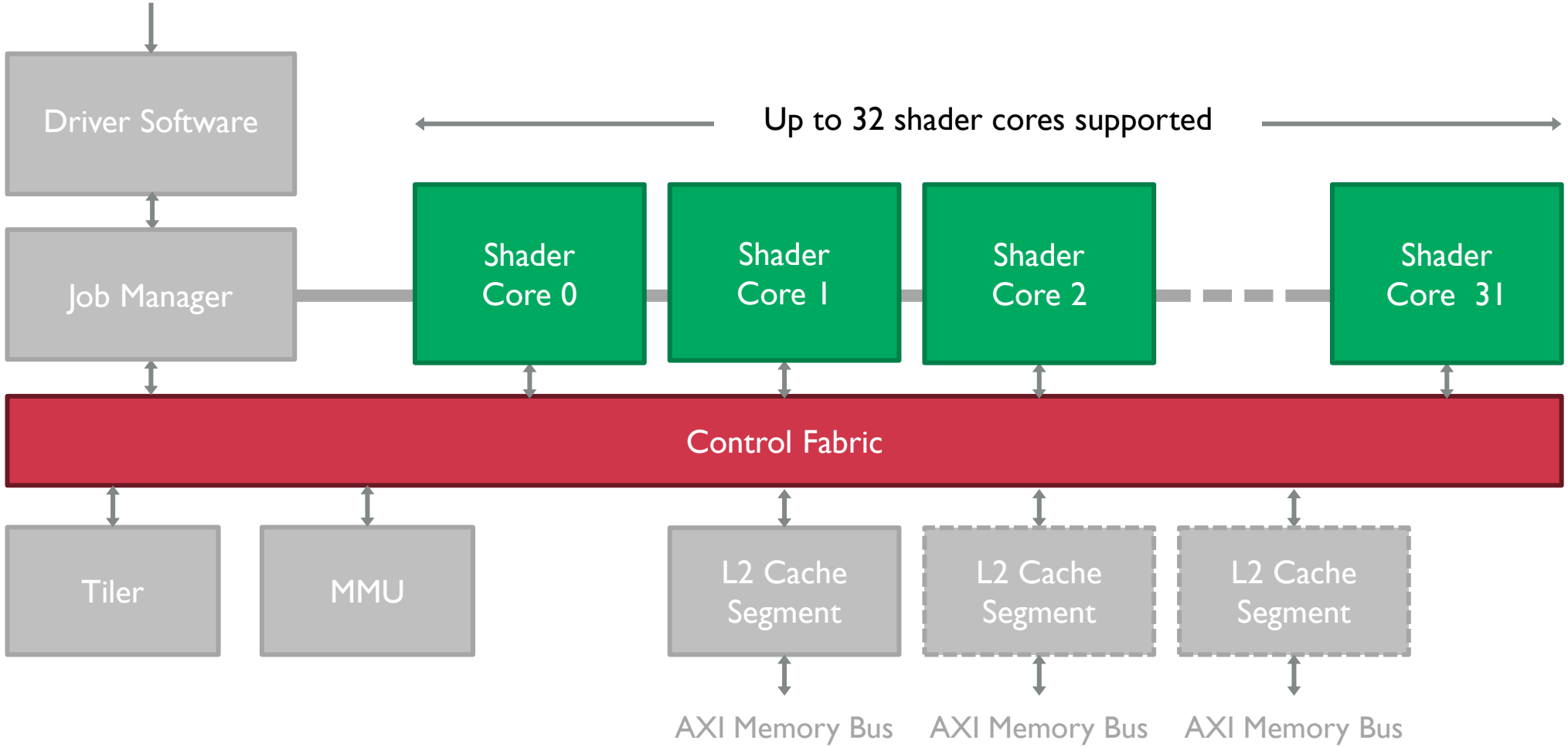


ARM

Bifrost GPU design

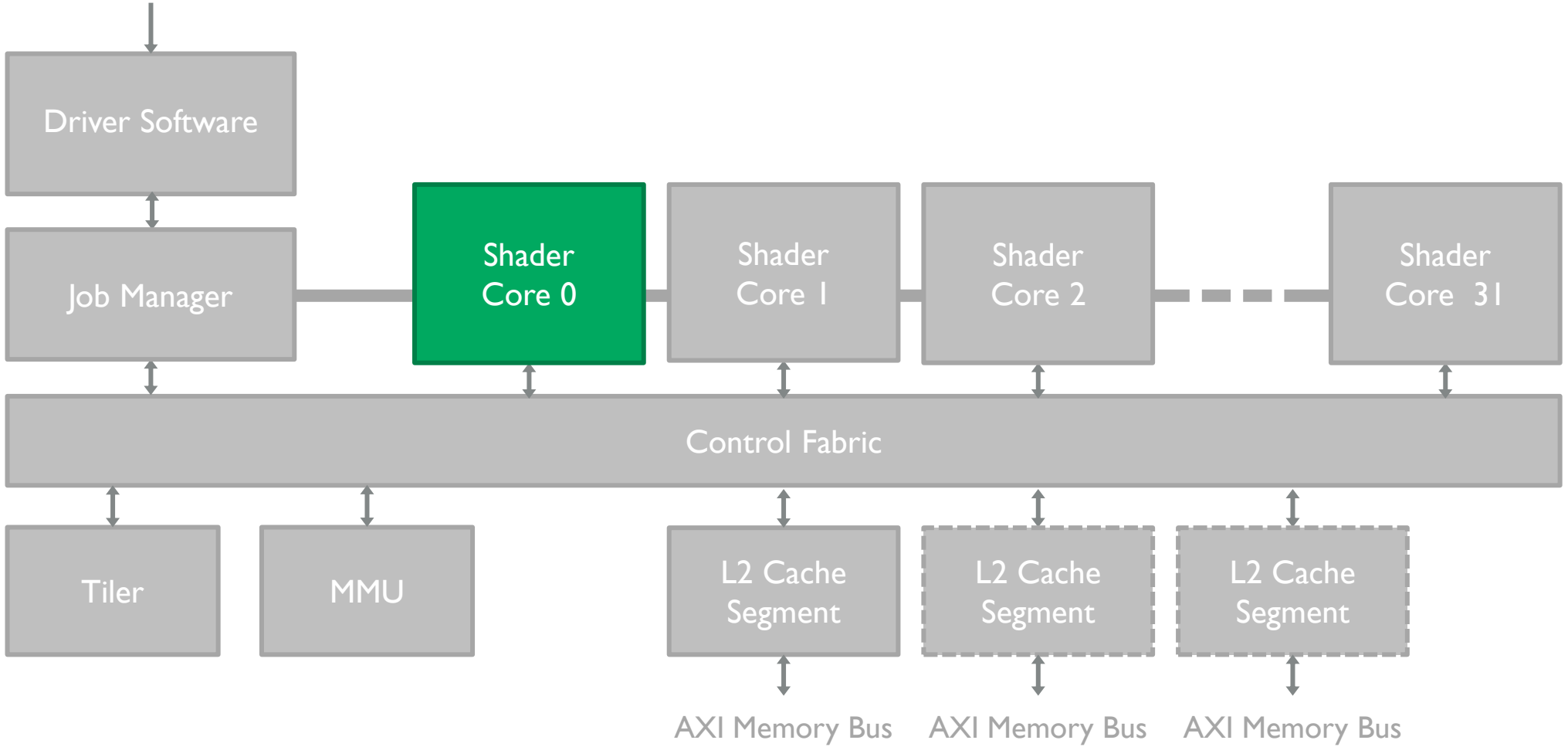


Scalable system design



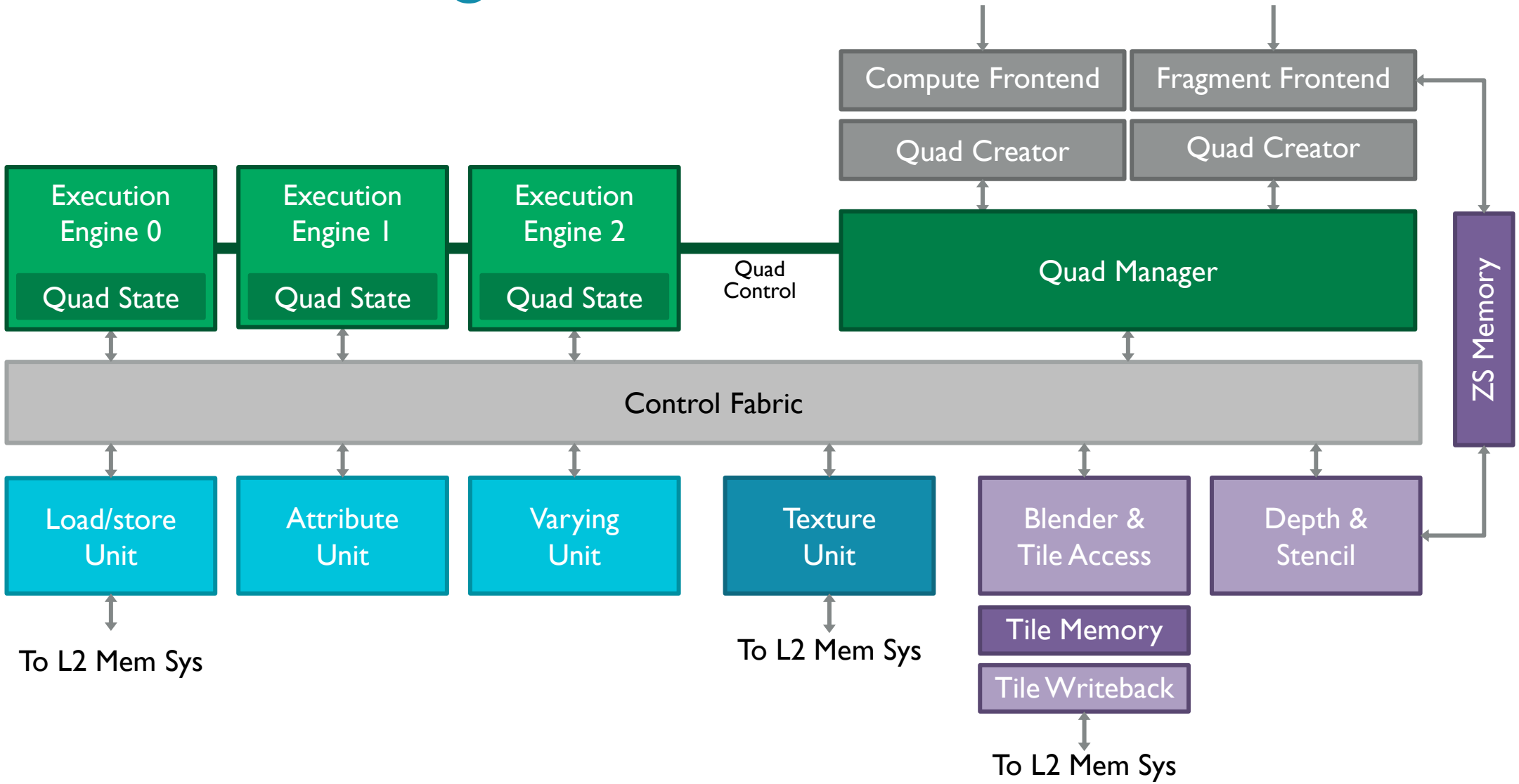
ARM

Execution core improvements



Bifrost core design

Bifrost core design

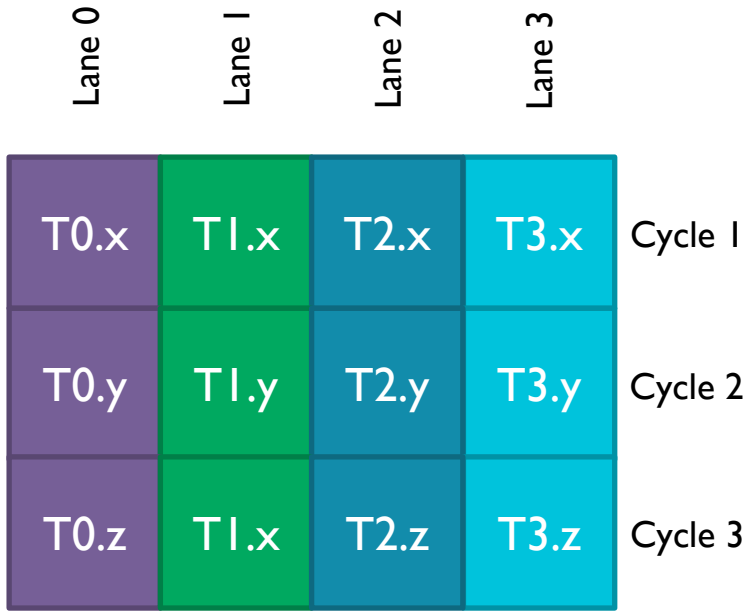


ARM

Quad vectorization

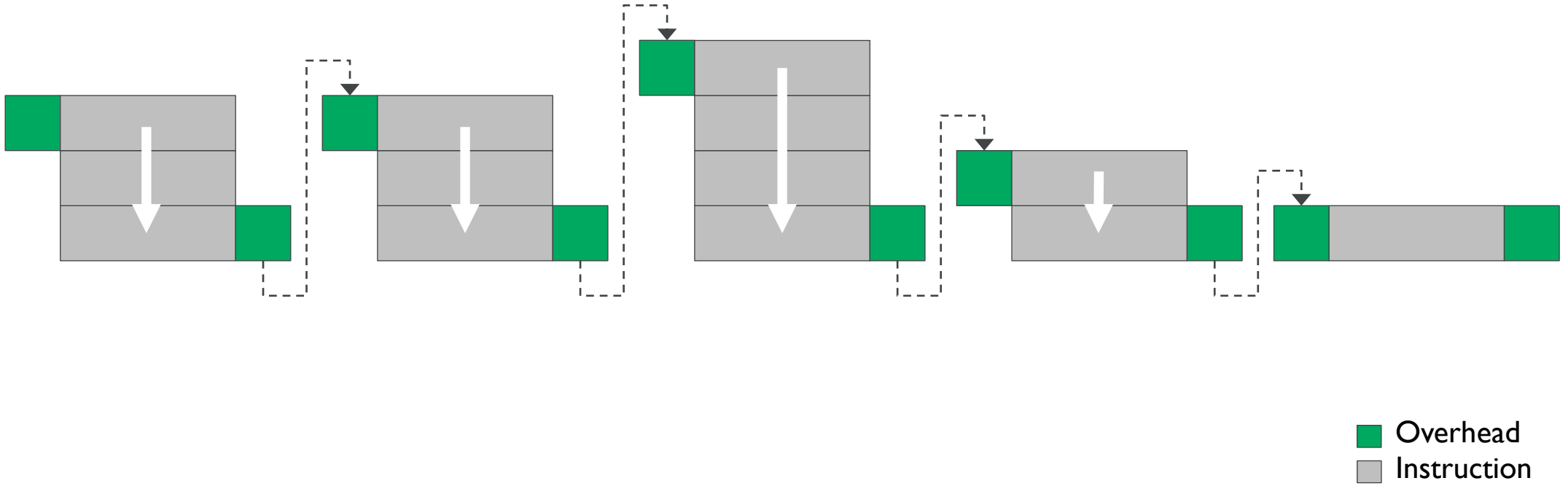
- Bifrost uses quad-parallel execution
  - Four scalar threads executed in lockstep in a "quad"
  - One quad at a time executes in each pipeline stage
  - Each thread fills one 32-bit lane of the hardware
  - 4 threads doing a vec3 FP32 add takes 3 cycles
  - Improves utilization

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



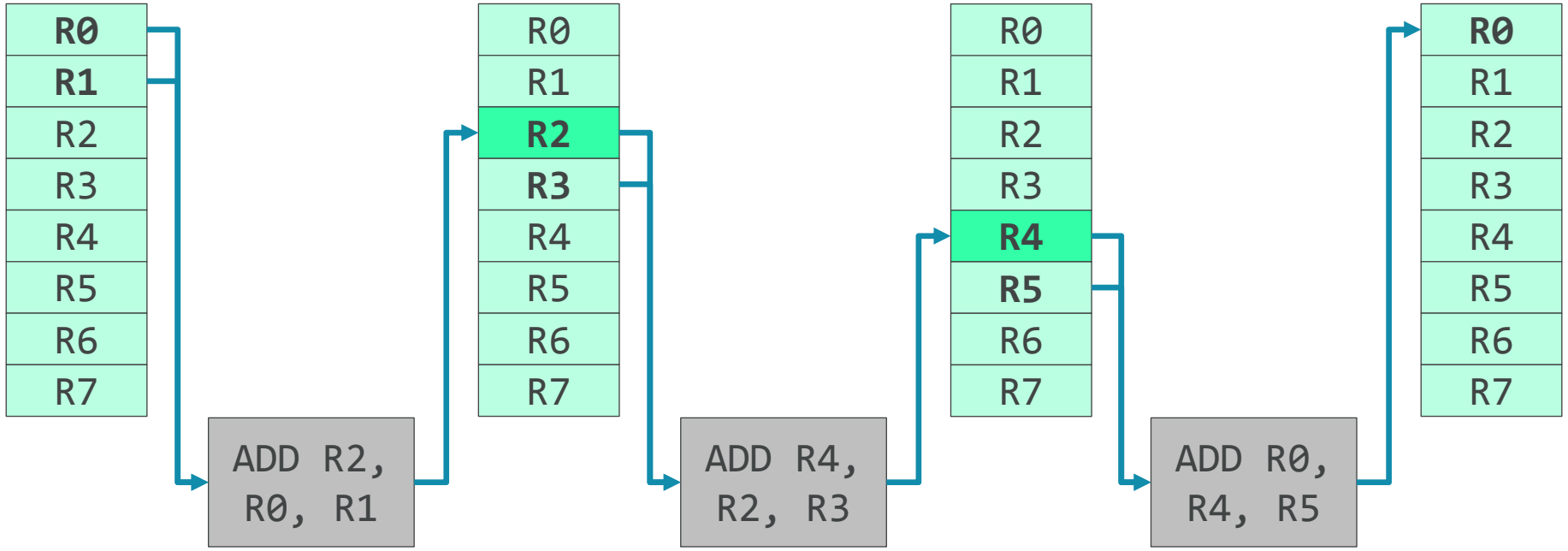
ARM

Clause execution



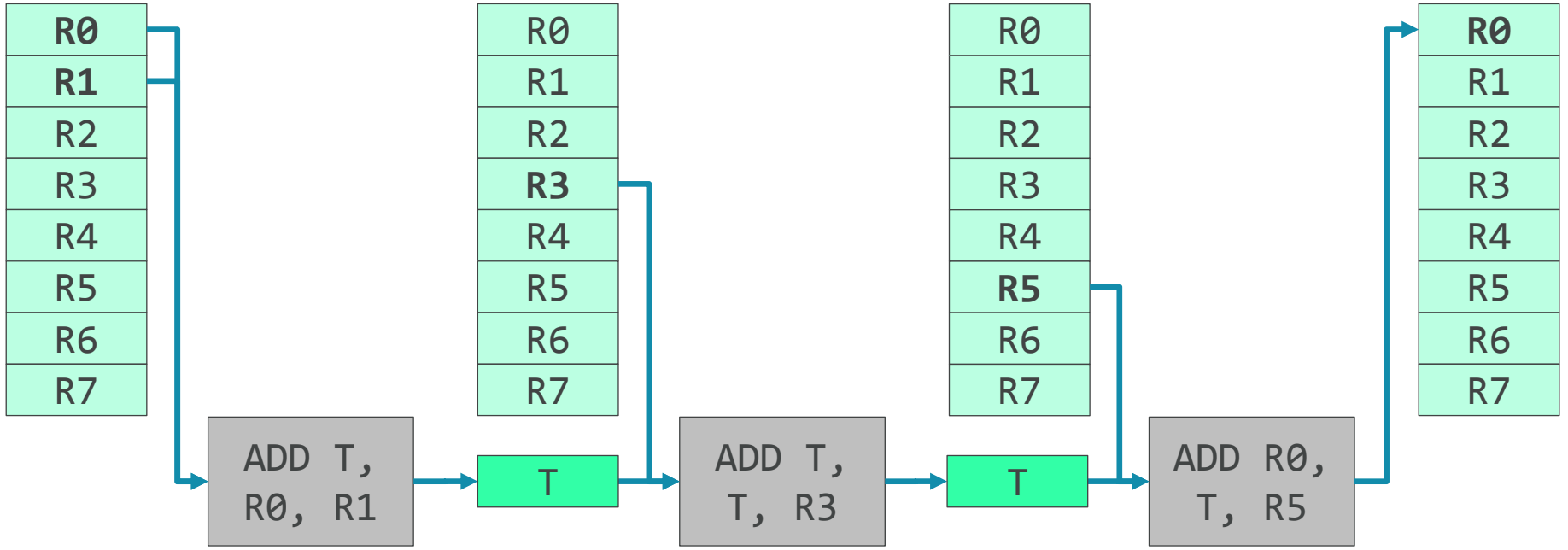
- Back-to-back execution guaranteed within a clause
- Allows aggressive optimisation

Clause execution



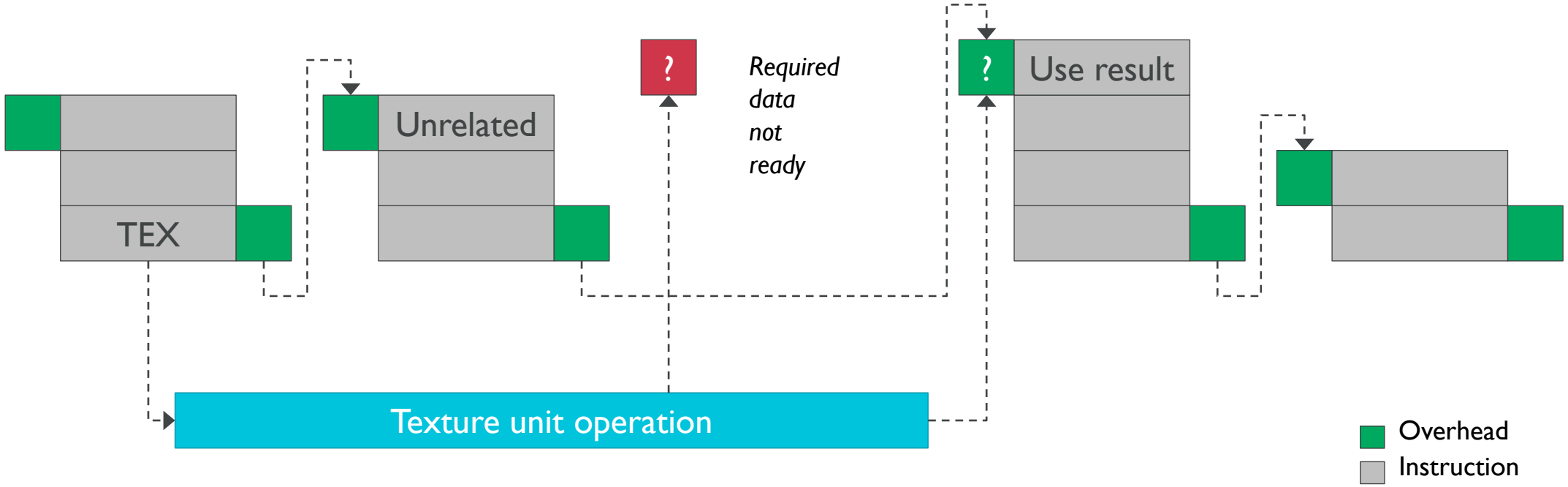
- Back-to-back register access is common
  - The result from one instruction is often only used as input to the next

Clause execution



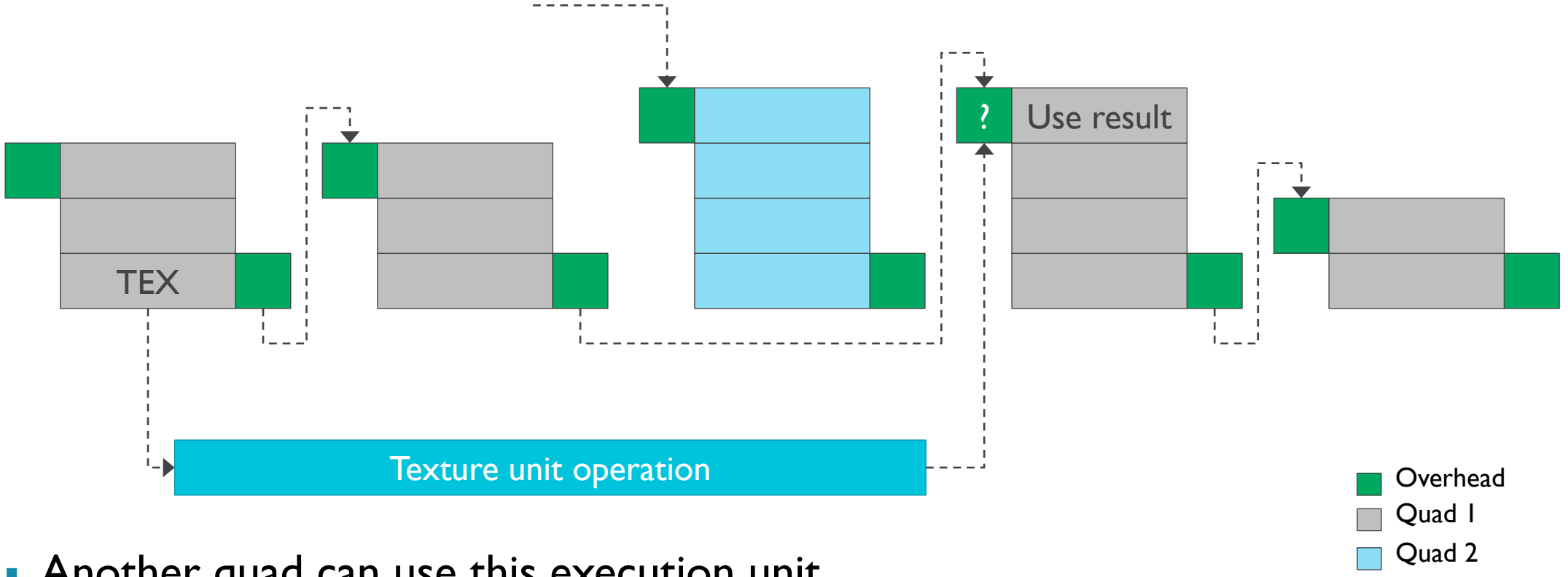
- Back-to-back register access is common
  - Register file bypass saves power.
  - Allows use of simpler, smaller register files.

Clause scheduling



- Delay next clause if asynchronous data not ready

Clause scheduling

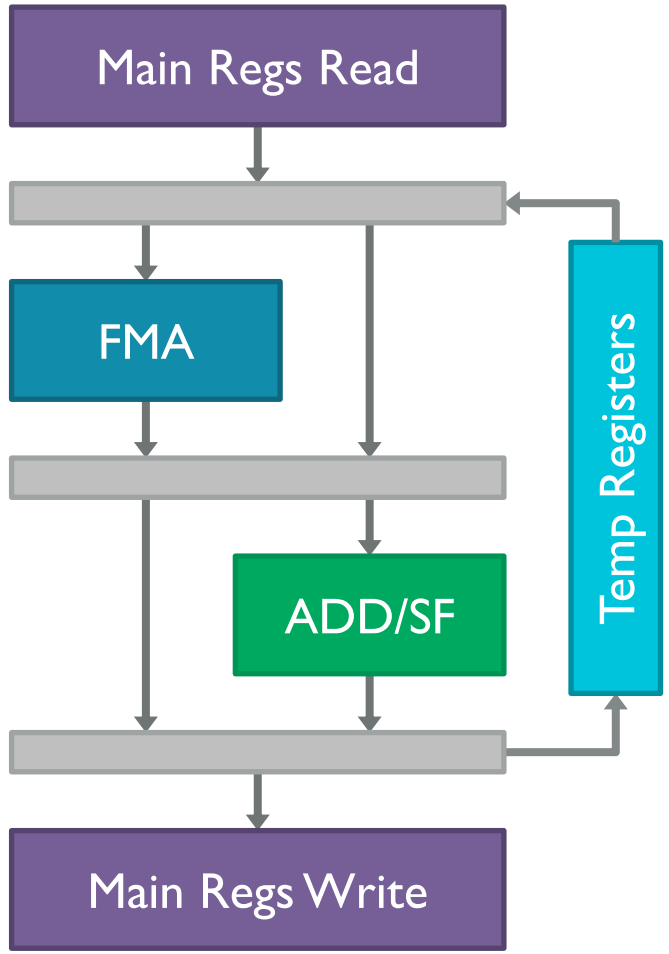


- Another quad can use this execution unit
- High utilization, high efficiency

Arithmetic functional units

Bifrost arithmetic functional units

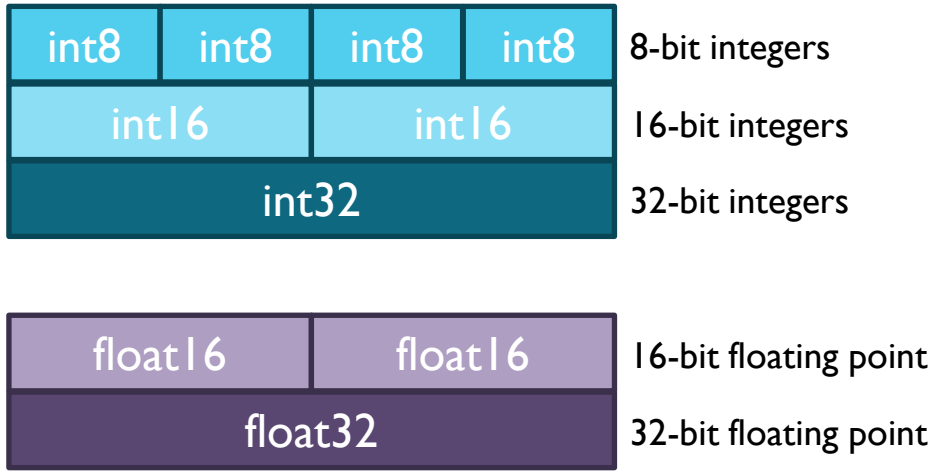
- Executes quad-parallel scalar operations
  - 4x32-bit multiplier FMA
  - 4x32-bit adder ADD
  - Adder includes special function unit
- Smaller and more area efficient
- Simplified layout eases compilation
  - Better scheduling in today's code
  - Better utilization
- One instruction word contains two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

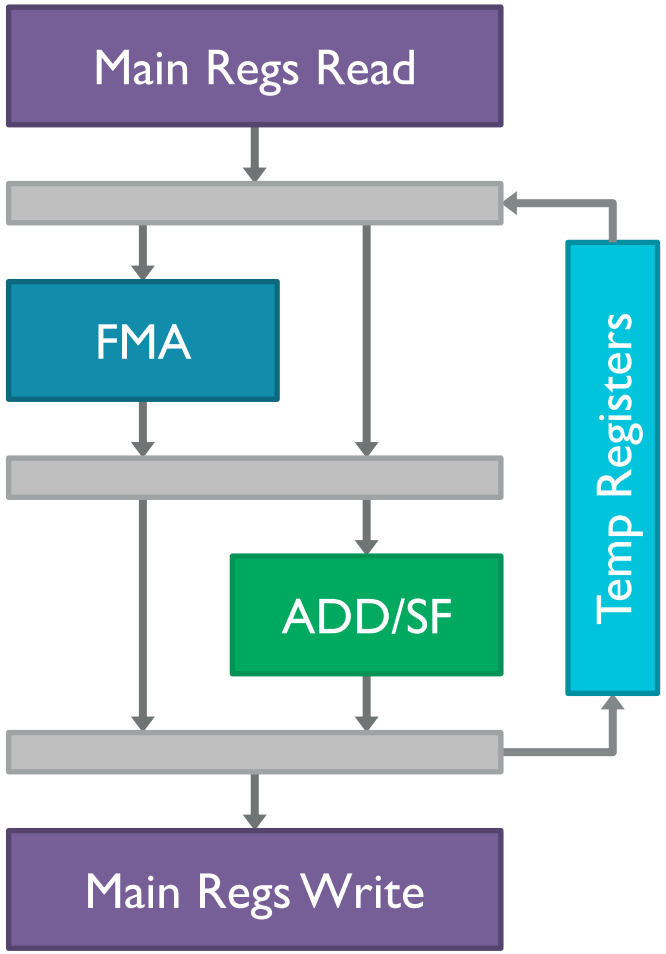


ARM

Bifrost arithmetic functional units

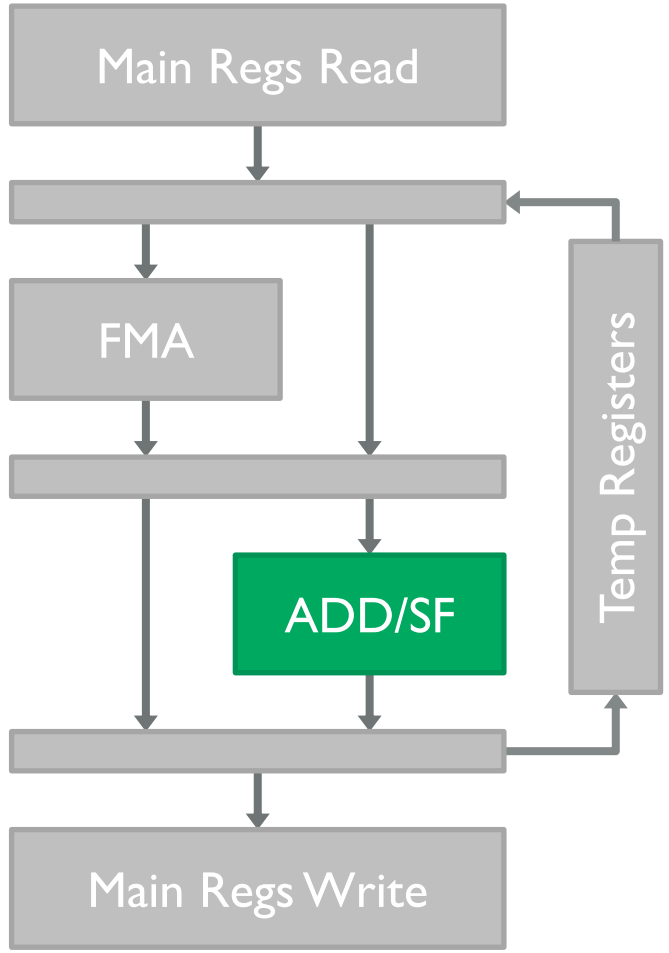
- Retains support for smaller width data types
  - Integers useful for deep learning
  - 2x performance for FP16 useful for pixel shaders





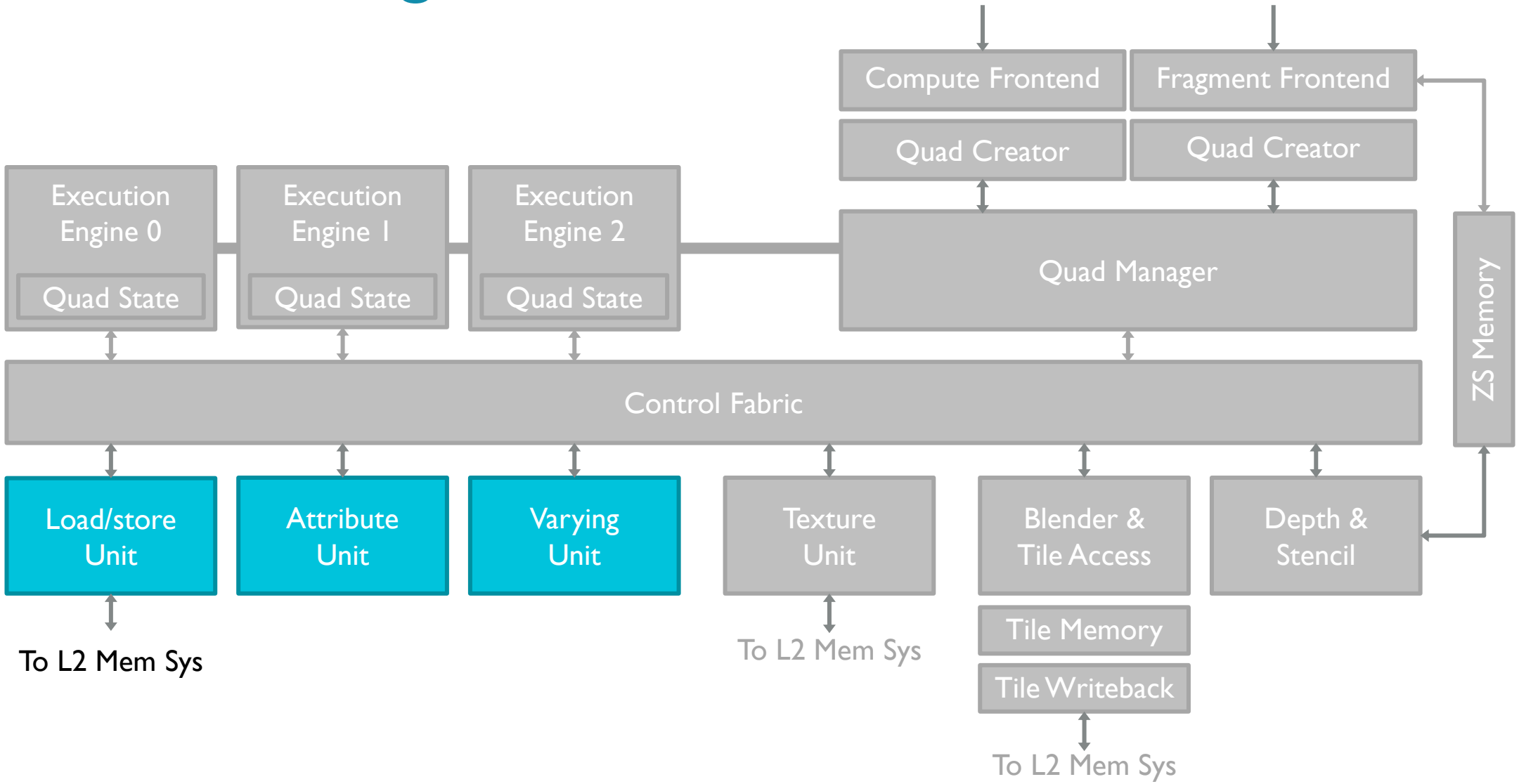
Special arithmetic operations

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



Load/store units

New core design



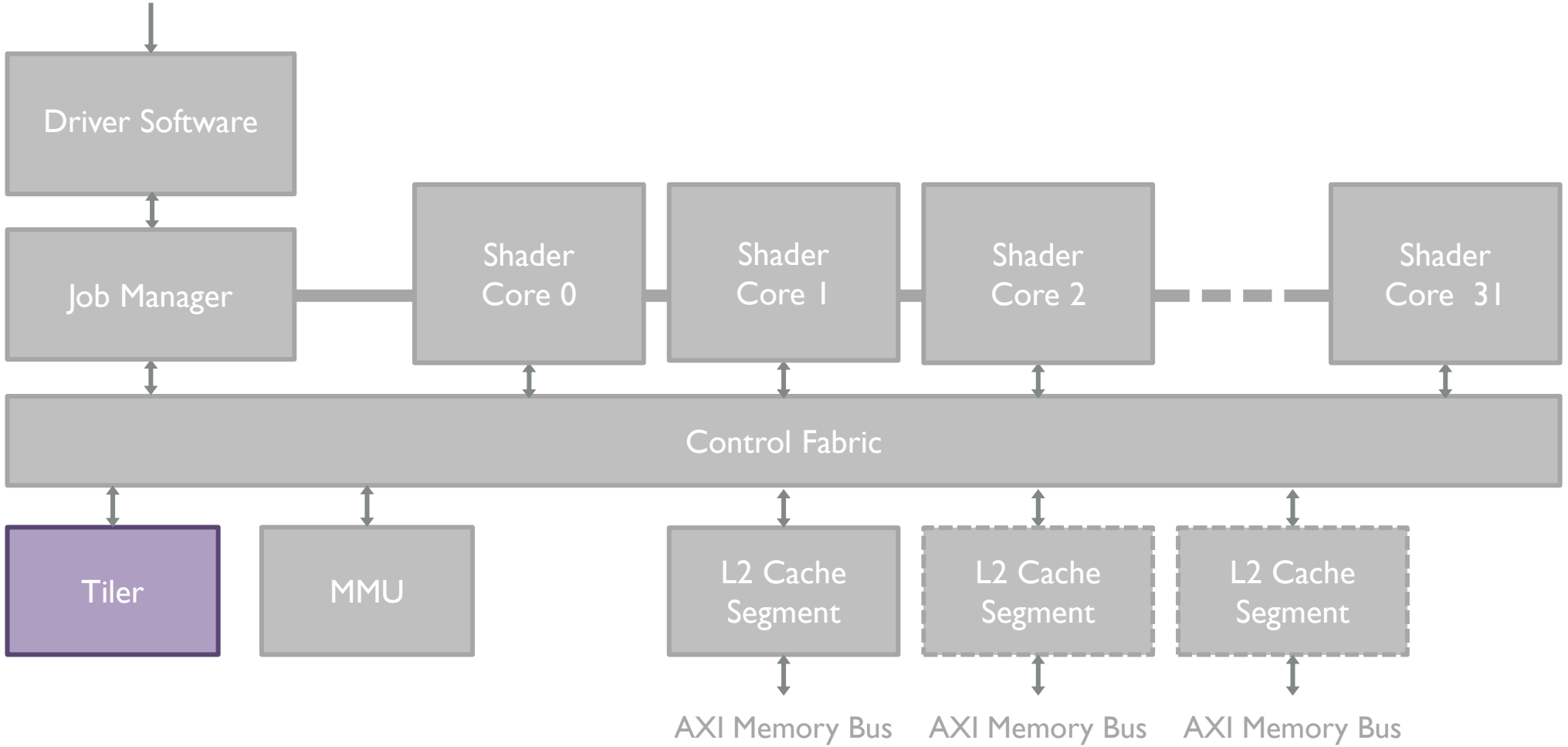
Bifrost load/store units

- Separate units, scheduled separately, for better utilization

Load/store Unit	Attribute Unit	Varying Unit
▪ Handles most general memory accesses ▪ Includes memory address translation and coherent caching	▪ Handles attribute indexing and addressing ▪ Defers to load/store for actual memory access	▪ Handles varying interpolation ▪ Lower power, but more range and precision than Midgard

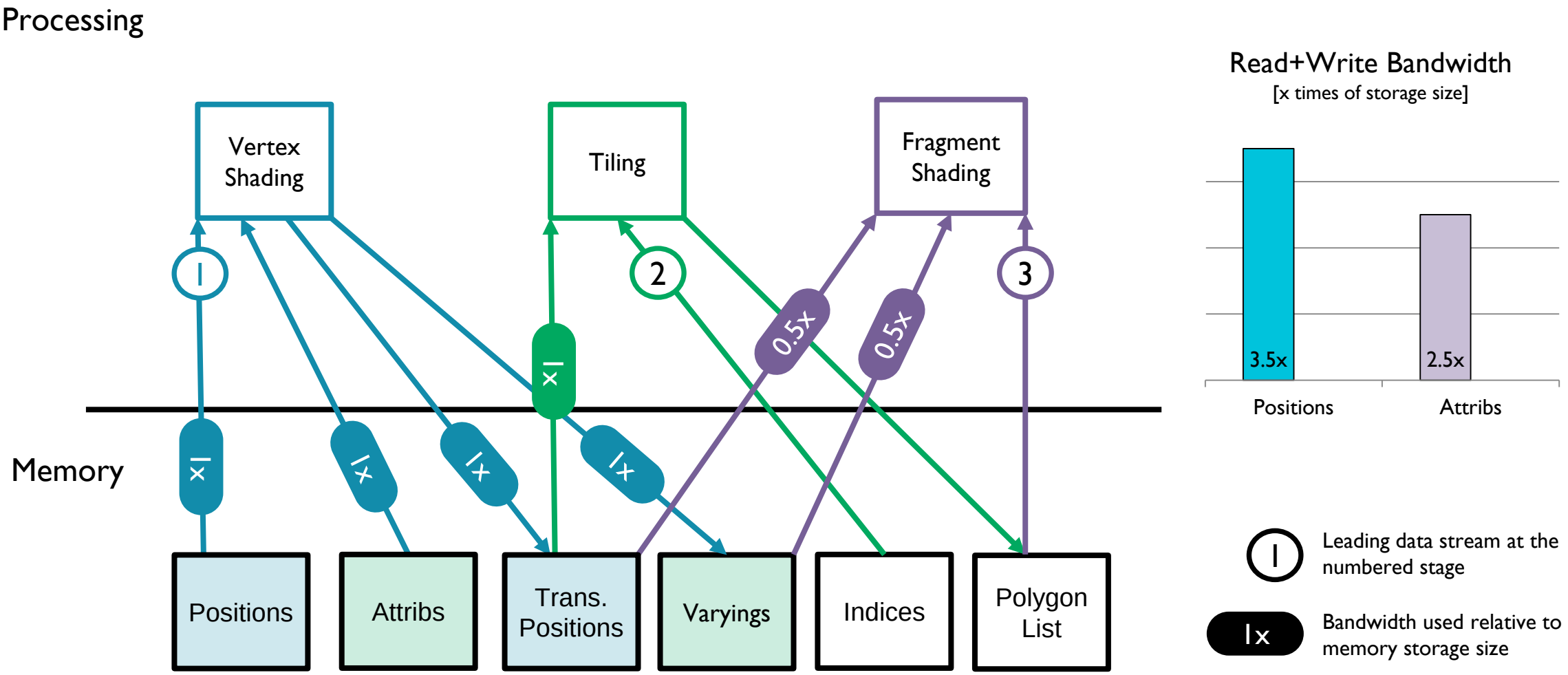
Tiler

Geometry flow improvement



ARM

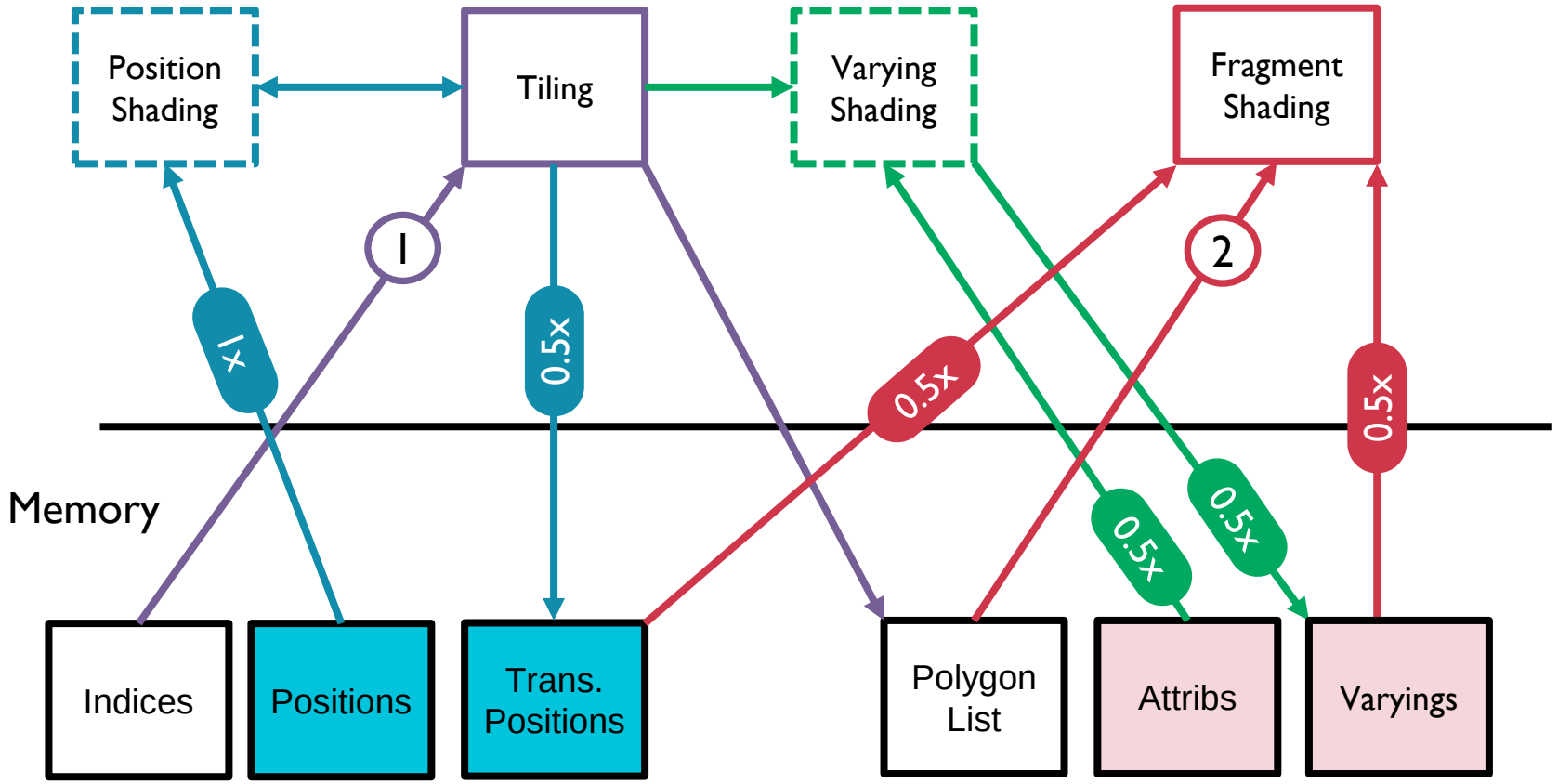
Geometry flow: Midgard



ARM

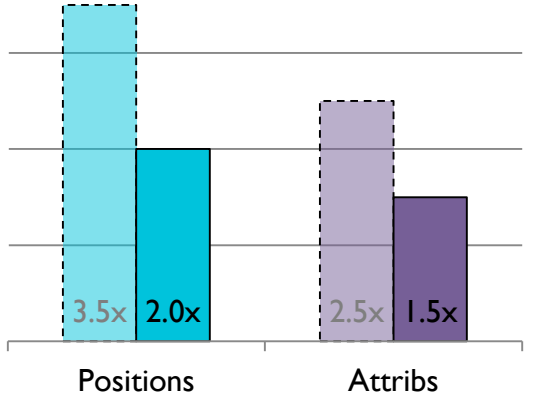
Geometry flow: Bifrost - index-driven vertex shading

Processing



Read+Write Bandwidth

[x times of storage size]



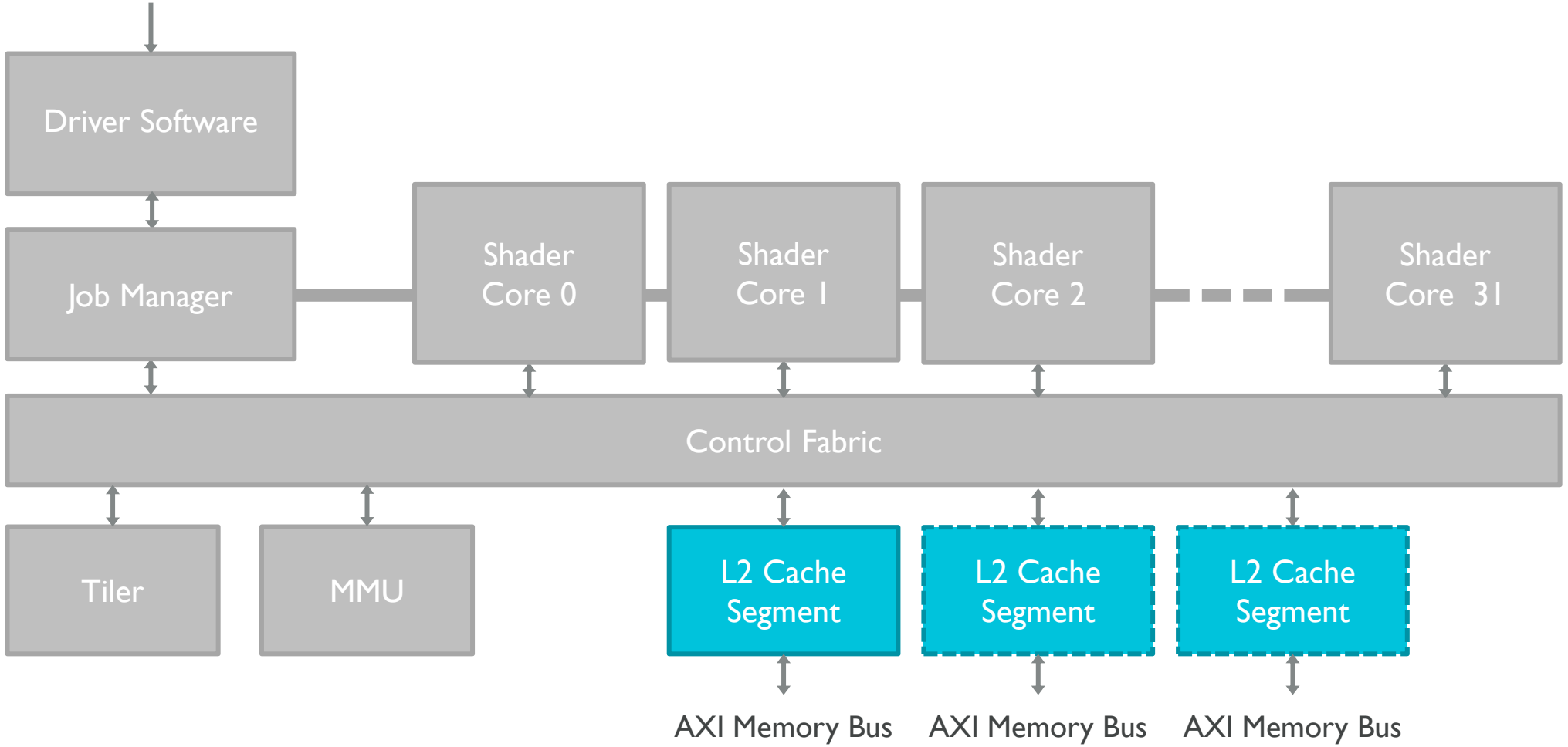
1 Leading data stream at the numbered stage

1x Bandwidth used relative to memory storage size

ARM

Memory system

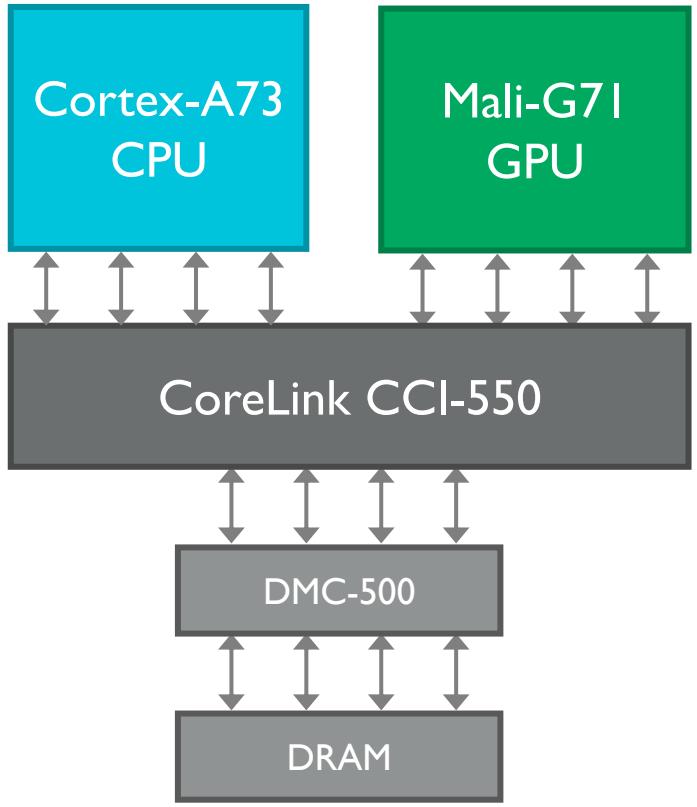
Memory system



Full coherency using 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 AXI which improves LPDDR4 performance



ARM