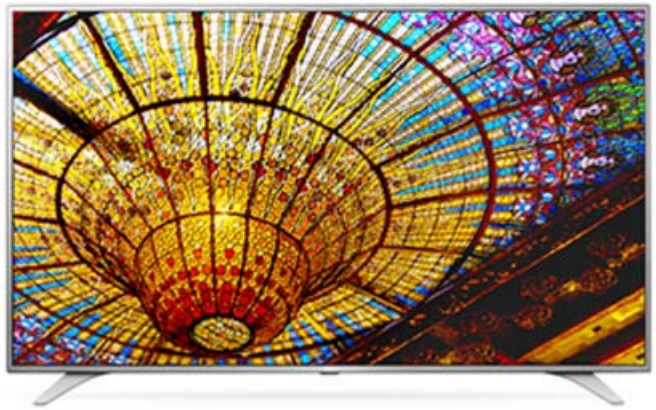
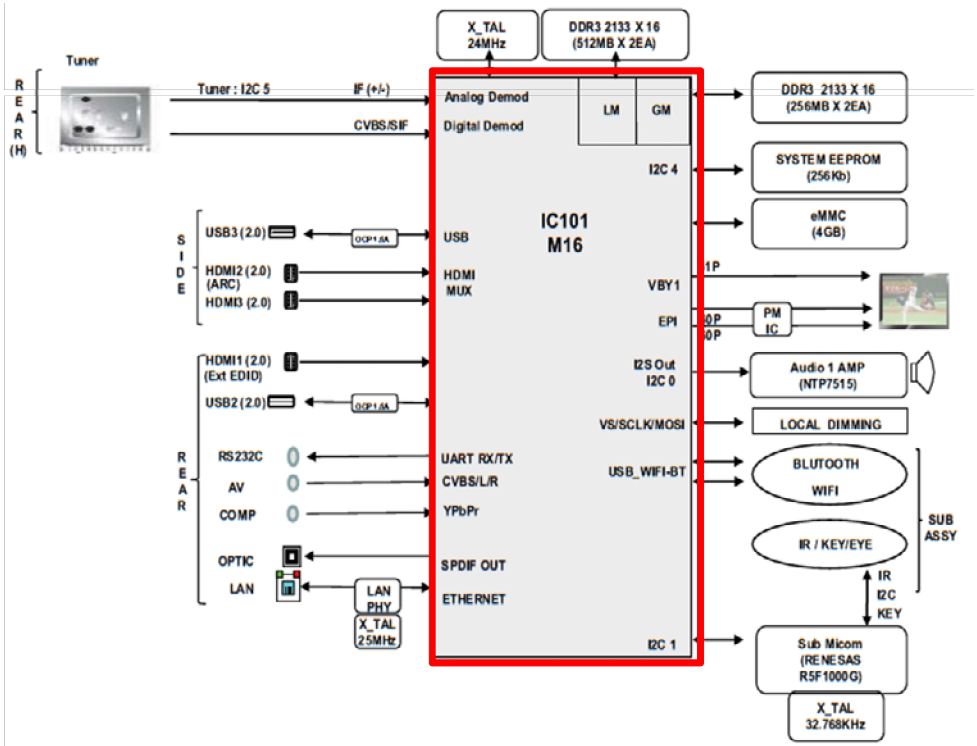
PUBLIC
EXHIBIT 21

U.S. Patent No. 8,760,454

LG / MediaTek Products

2. A unified shader comprising:	The LG 49UH6500 television and X Power LS755 phone (collectively, the "LG Products") include a unified shader.  *See* http://www.lg.com/us/support-product/lg-49UH6500.  *See* http://www.lg.com/us/cell-phones/lg-LS755-x-power-boost-mobile. The LG Products include one of the following System-on-Chips (SoCs): M16 and MediaTek MT6755M.

"2. A unified shader comprising:"



See LG LED TV Service Manual, Chassis: UA63J, Model: 43UH6500, p.28, *available at* https://lg.encompass.com/shop/model_research_docs/?file=/ZEN/sm/43UH6500UB.pdf.[1/]

Technical Specifications

Carrier	Boost Mobile®
Display	5.3" (1280 x 720) HD TFT Display
Battery	4,100 mAh non-removable
Platform	Android 6.0.1 Marshmallow
Processor	MediaTek 1.8 GHz Octa-Core MT6755M

[1/] The LG 49UH6500 television and the LG 43UH6500 television are part of the LG UH6500 Series televisions. *See* http://www.lg.com/us/support/products/documents/UH6500_Series_Spec_Sheet_Updated_10112016.pdf.

See http://www.lg.com/us/cell-phones/lg-LS755-x-power-boost-mobile.

The SoCs include one of the following ARM Mali graphics processing units (the "Mali GPUs"): T760 MP2 and T860 MP2.

		M16
Smart Function	CPU	CA53 x4 1.1GHz / 1MB
	GPU	Mali T760 MP2 (650MHz)
	OSD	Separated 2K@60p
	HEVC	4K @60,10bit
	DDR	DDR3-2133/ DDR4-2400
	Audio DSP	HiFi3 Dual @370MHz

See LG LED TV Service Manual, Chassis: UA63J, Model: 43UH6500, p.123, *available at* https://lg.encompass.com/shop/model_research_docs/?file=/ZEN/sm/43UH6500UB.pdf.

MediaTek MT6755 Helio P10 Specs

Release	Q4 2015
Process	28nm
Apps CPU	8x Cortex-A53, up to 2.0GHz
GPU	ARM Mali-T860 MP2 at 700 MHz

See http://cnoemphone.com/blog/mediatek-mt6755-helio-p10-specs-benchmark-and-smartphone-list.

The Mali GPUs share substantially similar structure, function, and operation.

High performance

With stunning graphics capabilities, ARM Mali High Performance GPUs combine GPU Compute functionality with micro-architecture enhancements and system-wide, bandwidth-saving technology to bring energy efficiency to advanced mobile and consumer devices. GPU Compute solutions enable each task to be executed on the most suitable processor within the system. The resultant efficiencies guarantee superior graphics performance and extended battery life.

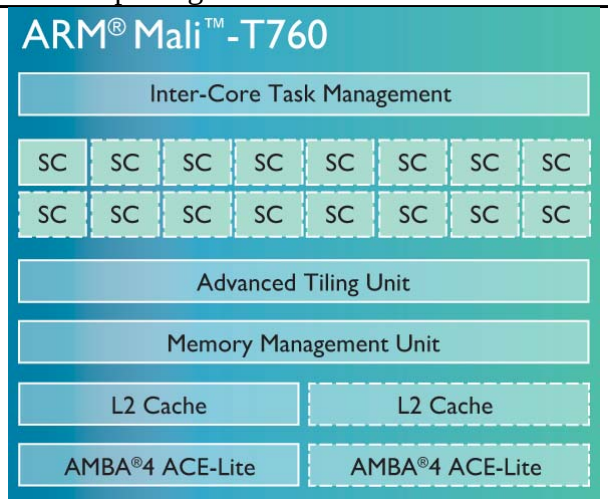
High performance GPUs:

- Mali-G71
- Mali-T860 & T880
- Mali-T760
- Mali-T628
- Mali-T624

See http://www.arm.com/products/graphics-and-multimedia/mali-gpu.

For example, the Mali GPUs include multiple shader cores (SCs).

"2. A unified shader comprising:"



See http://www.arm.com/products/multimedia/mali-gpu/high-performance/mali-t860-t880.php

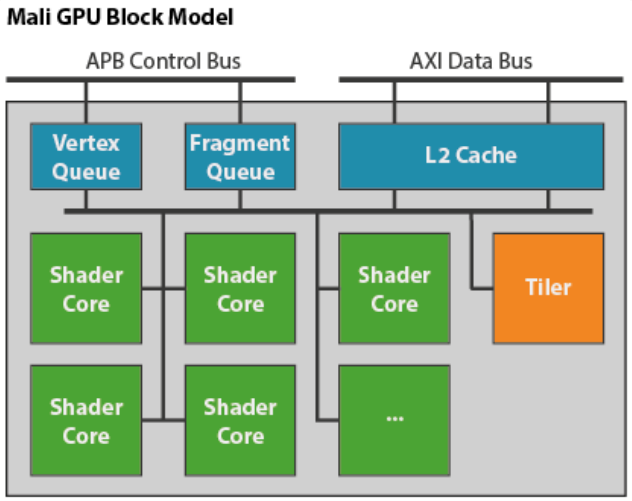
The Mali GPUs implement "a unified shader core architecture."

GPU Architecture

The "Midgard" family of Mali GPUs (the Mali-T600 and Mali-T700 series) use a unified shader core architecture, meaning that only a single type of shader core exists in the design. This single core can execute all types of programmable shader code, including vertex shaders, fragment shaders, and compute kernels.

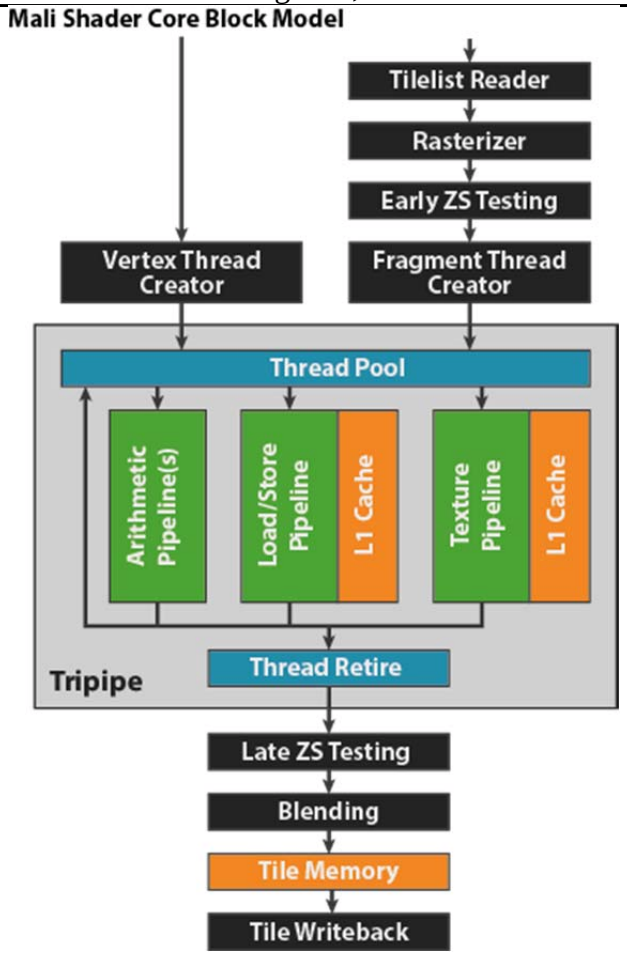
See https://community.arm.com/groups/arm-mali-graphics/blog/2014/03/12/the-mali-gpu-an-abstract-machine-part-3--the-shader-core.

a general purpose register block for maintaining data;	The LG Products include a general purpose register block for maintaining data. For example, "[e]very thread has its own registers…stack pointer and private stack[.]" Furthermore, "[s]hared read only registers are used for kernel arguments[.]" CL Execution model on Mali-T600 (2) ▪ Each work-item runs as one of the threads within a core ▪ Every Mali-T600 thread has its own independent program counter ▪ …which supports divergent threads from the same kernel ▪ caused by conditional execution, variable length loops etc. ▪ Some other GPGPU's use "WARP" architectures ▪ These share a common program counter with a group of work-items ▪ This can be highly scalable… but can be slow handling divergent threads ▪ T600 effectively has a Warp size of 1 ▪ Up to 256 threads per core ▪ Every thread has its own registers ▪ Every thread has its own stack pointer and private stack ▪ Shared read-only registers are used for kernel arguments 20 ARM Additionally, the Mali GPUs include the Mali GPU includes a Vertex Queue, Fragment Queue, Thread Pool, Load/Store Pipe, Caches, and registers.

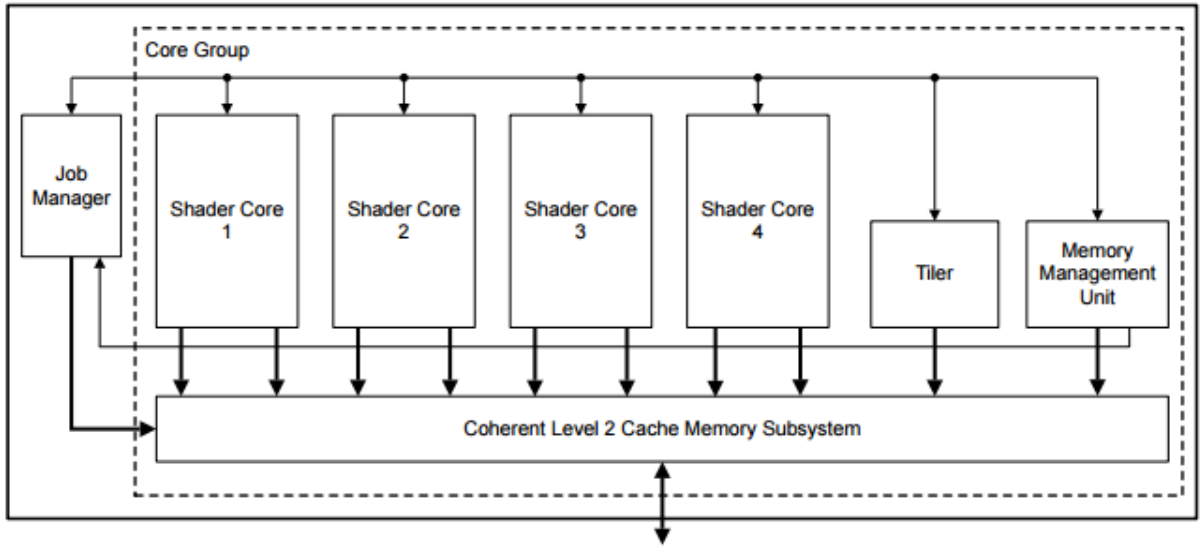


See The Mali GPU: An Abstract Machine, Part 3 - The Midgard Shader Core, https://community.arm.com/groups/arm-mali-graphics/blog/2014/03/12/the-mali-gpu-an-abstract-machine-part-3--the-shader-core.

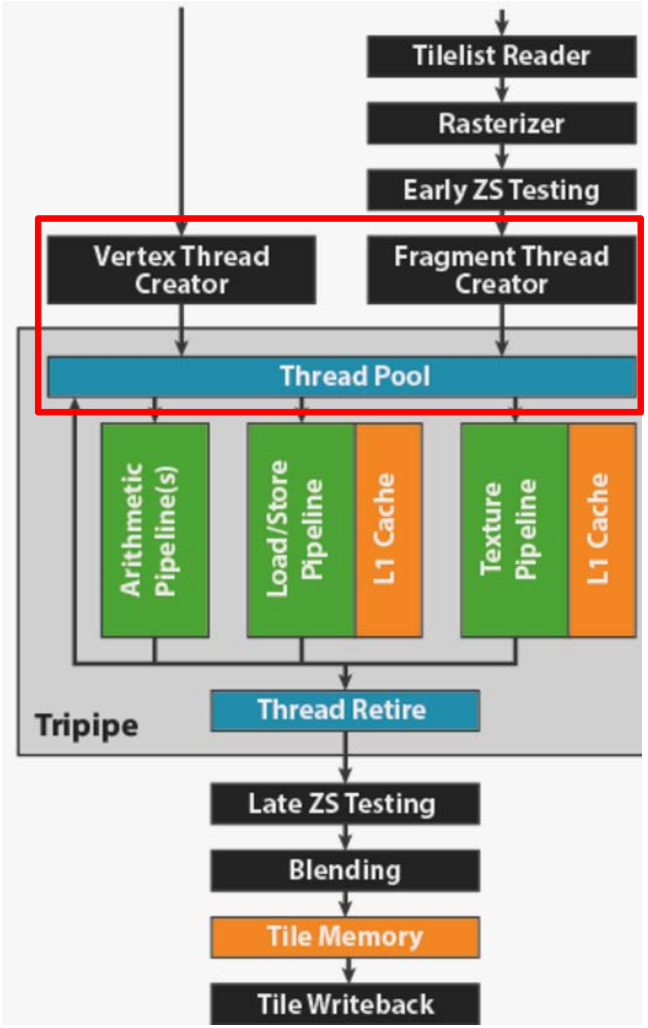
"a general purpose register block for maintaining data;"



See The Mali GPU: An Abstract Machine, Part 3 - The Midgard Shader Core, https://community.arm.com/groups/arm-mali-graphics/blog/2014/03/12/the-mali-gpu-an-abstract-machine-part-3--the-shader-core.

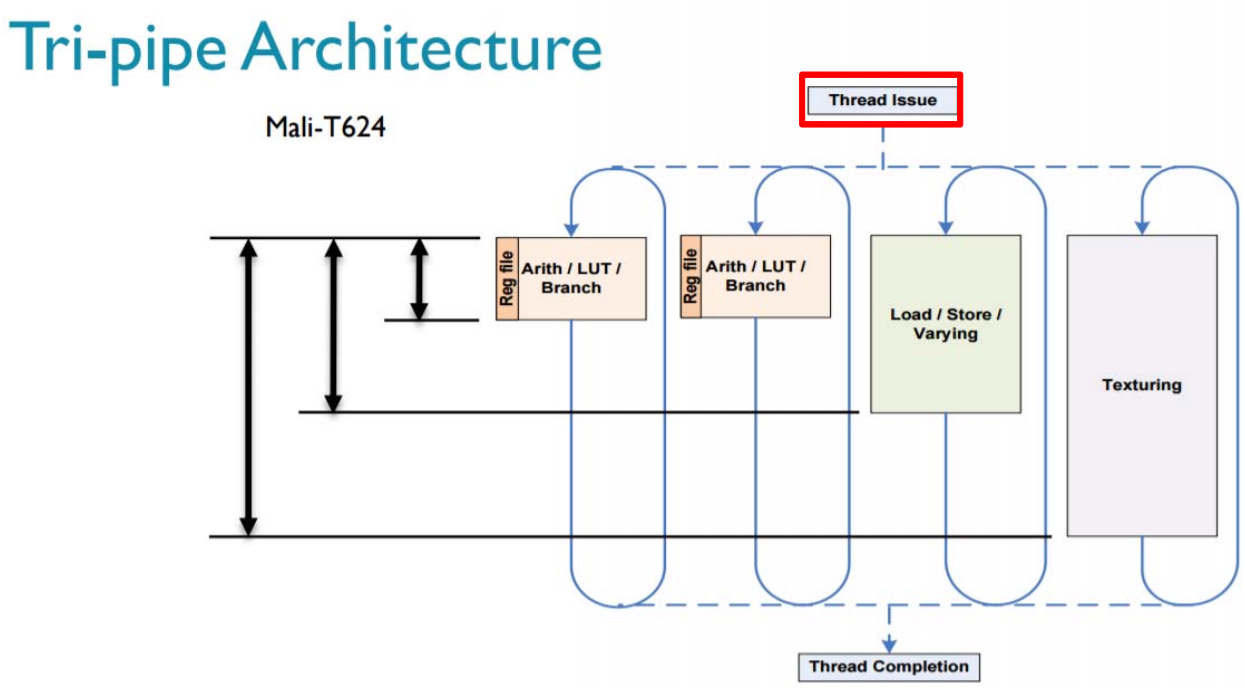
a processor unit;	The LG Products include a processor unit. For example, the Mali GPU includes multiple shader cores.  **Figure 1-1 Midgard architecture Mali GPU** **Shader cores** The shader cores handle the vertex and fragment processing stages of the graphics pipeline. The shader cores generate lists of primitives and accelerate the building of data structures, such as polygon lists and packed vertex data, for fragment processing. The shader cores also handle the rasterization and fragment processing stages of the graphics pipeline. They use the data structures and lists of primitives that are generated during vertex processing to produce the framebuffer result that appears on the screen. *See* http://malideveloper.arm.com/downloads/OpenGLES3.x/arm_mali_gpu_opengl_es_3-x_developer_guide_en.pdf.

"a sequencer, coupled to the general purpose register block and the processor unit, the sequencer maintaining instructions operative to cause the processor unit to execute vertex calculation and pixel calculation operations on selected data maintained in the general purpose register block; and"

a sequencer, coupled to the general purpose register block and the processor unit, the sequencer maintaining instructions operative to cause the processor unit to execute vertex calculation and pixel calculation operations on selected data maintained in the general purpose register block; and	The LG Products include a sequencer, coupled to the general purpose register block and the processor unit, the sequencer maintaining instructions operative to cause the processor unit to execute vertex calculation and pixel calculation operations on selected data maintained in the general purpose register block. For example, the Mali GPUs include the Vertex Thread Creator, Fragment Thread Creator, Compute Thread Creator, the Thread Pool (also known as the Thread Issue), which feed the processor unit with instructions for the processor to execute vertex and pixel operations in the general purpose register block. 

"a sequencer, coupled to the general purpose register block and the processor unit, the sequencer maintaining instructions operative to cause the processor unit to execute vertex calculation and pixel calculation operations on selected data maintained in the general purpose register block; and"

See https://community.arm.com/groups/arm-mali-graphics/blog/2014/03/12/the-mali-gpu-an-abstract-machine-part-3--the-shader-core.

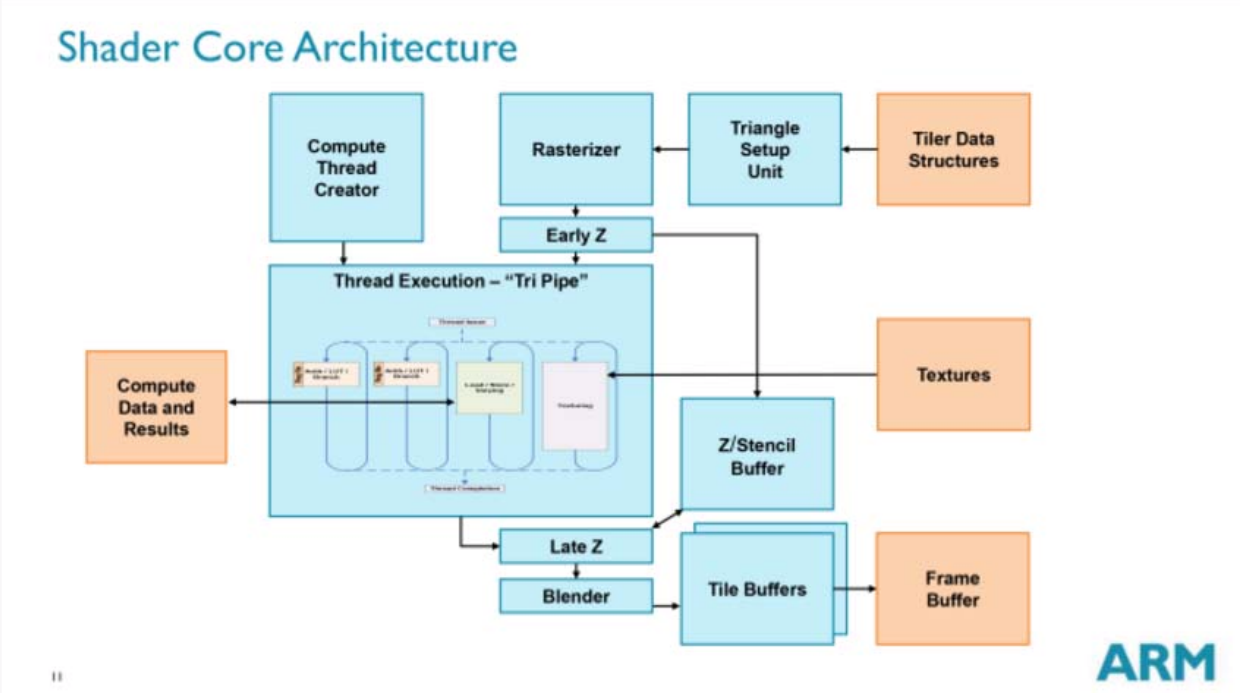


See ARM Midgard Architecture, p.11, *available at* http://malideveloper.arm.com/downloads/ARM_Game_Developer_Days/PDFs/2-Mali-GPU-architecture-overview-and-tile-local-storage.pdf.

"a sequencer, coupled to the general purpose register block and the processor unit, the sequencer maintaining instructions operative to cause the processor unit to execute vertex calculation and pixel calculation operations on selected data maintained in the general purpose register block; and"

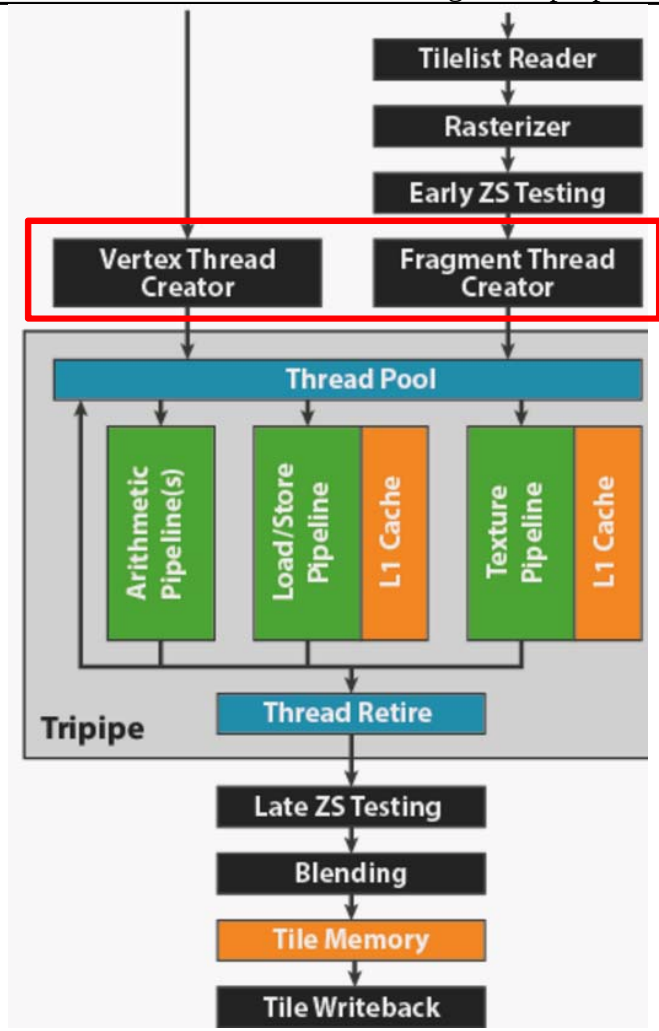
The Midgard Architecture

Diving in to the Mali architecture, we'll start with a high level overview of the architecture. What we're looking at here is a single Midgard shader core, which despite the "shader" name actually contains a whole lot more. A shader core in this context contains the actual shader core within one of Midgard's "tri pipe" shader blocks, but also contains a triangle setup unit, rasterizer, Z & stencil hardware, a ROP/blender, tiling hardware, and **a compute thread creator specifically for feeding a tri pipe with compute workloads.**



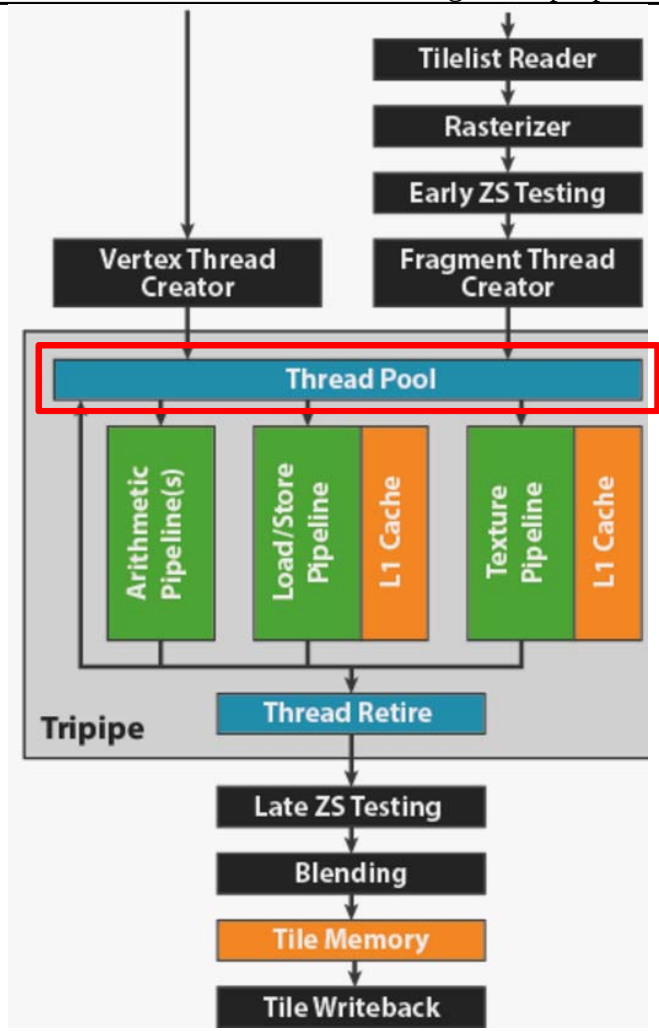
See Ryan Smith, ARM's Mali Midgard Architecture Explored, http://www.anandtech.com/show/8234/arms-mali-midgard-architecture-explored/4.

"a sequencer, coupled to the general purpose register block and the processor unit, the sequencer maintaining instructions operative to cause the processor unit to execute vertex calculation and pixel calculation operations on selected data maintained in the general purpose register block; and"



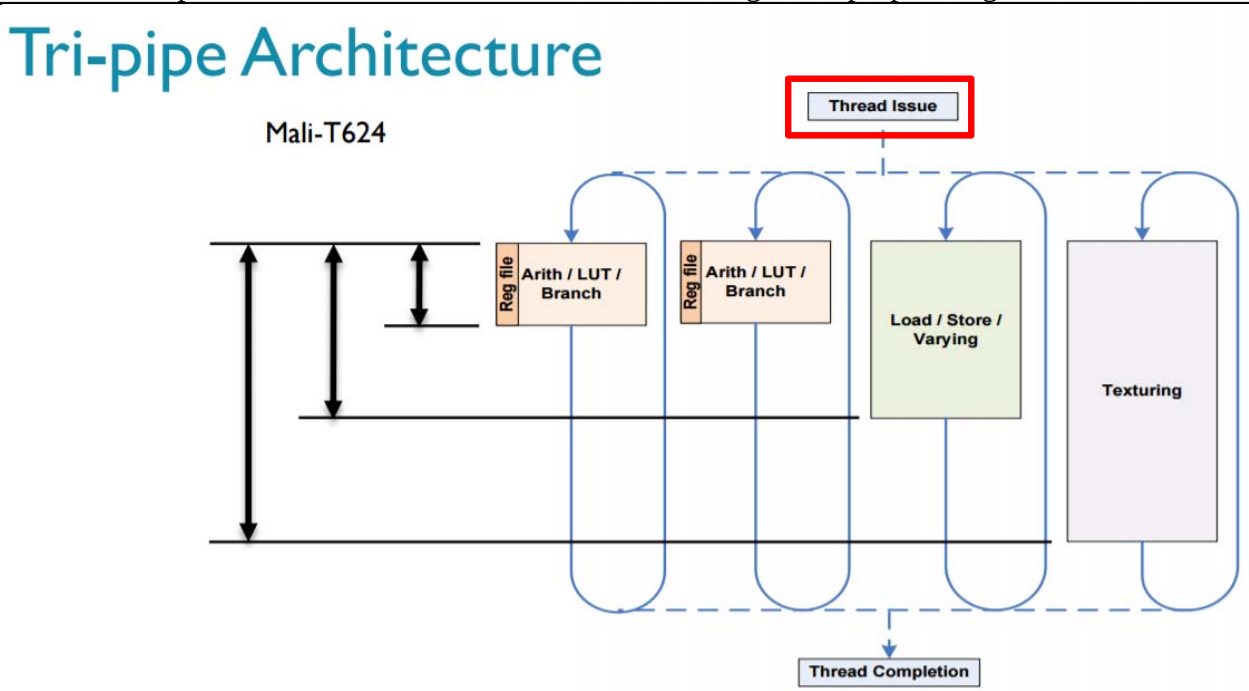
See https://community.arm.com/groups/arm-mali-graphics/blog/2014/03/12/the-mali-gpu-an-abstract-machine-part-3--the-shader-core.

"a sequencer, coupled to the general purpose register block and the processor unit, the sequencer maintaining instructions operative to cause the processor unit to execute vertex calculation and pixel calculation operations on selected data maintained in the general purpose register block; and"



See https://community.arm.com/groups/arm-mali-graphics/blog/2014/03/12/the-mali-gpu-an-abstract-machine-part-3--the-shader-core.

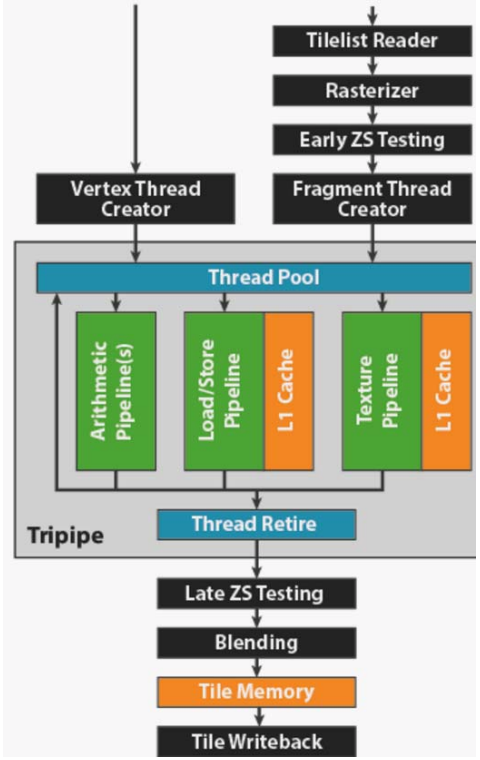
"a sequencer, coupled to the general purpose register block and the processor unit, the sequencer maintaining instructions operative to cause the processor unit to execute vertex calculation and pixel calculation operations on selected data maintained in the general purpose register block; and"



See ARM Midgard Architecture, p.11, *available at* http://malideveloper.arm.com/downloads/ARM_Game_Developer_Days/PDFs/2-Mali-GPU-architecture-overview-and-tile-local-storage.pdf.

U.S. Patent No. 8,760,454: Claim 2

" wherein the processor unit executes instructions that generate a pixel color in response to selected data from the general purpose register block and generates vertex position and appearance data in response to selected data from the general purpose register block."

wherein the processor unit executes instructions that generate a pixel color in response to selected data from the general purpose register block and generates vertex position and appearance data in response to selected data from the general purpose register block.	The LG Products include the processor unit that executes instructions that generate a pixel color in response to selected data from the general purpose register block and generates vertex position and appearance data in response to selected data from the general purpose register block. For example, the Mali GPUs include the Tripipe block that further includes the Arithmetic Pipeline(s), the Load/Store Pipeline, and the Texture Pipeline. The Arithmetic Pipeline "is a SIMD vector processing engine"; the Texture Pipeline "is responsible for all memory access to do with textures"; and the Load/Store Pipeline "is responsible for all memory access which are not related to texturing," such as reading attributes, writing varyings, and reading varyings.  *See* https://community.arm.com/groups/arm-mali-graphics/blog/2014/03/12/the-mali-gpu-an-abstract-machine-part-3--the-shader-core.

U.S. Patent No. 8,760,454: Claim 2

" wherein the processor unit executes instructions that generate a pixel color in response to selected data from the general purpose register block and generates vertex position and appearance data in response to selected data from the general purpose register block."

The Tripipe

There are three classes of execution pipeline in the tripipe design: one handling arithmetic operations, one handling memory load/store and varying access, and one handling texture access. There is one load/store and one texture pipe per shader core, but the number of arithmetic pipelines can vary depending on which GPU you are using; most silicon shipping today will have two arithmetic pipelines, but GPU variants with up to four pipelines are also available.

Massively Multi-threaded Machine

Unlike a traditional CPU architecture, where you will typically only have a single thread of execution at a time on a single core, the tripipe is a massively multi-threaded processing engine. There may well be hundreds of hardware threads running at the same time in the tripipe, with one thread created for each vertex or fragment which is shaded. This large number of threads exists to hide memory latency; it doesn't matter if some threads are stalled waiting for memory, as long as at least one thread is available to execute then we maintain efficient execution.

Arithmetic Pipeline: Vector Core

The arithmetic pipeline (A-pipe) is a SIMD (single instruction multiple data) vector processing engine, with arithmetic units which operate on 128-bit quad-word registers. The registers can be flexibly accessed as either 2 x FP64, 4 x FP32, 8 x FP16, 2 x int64, 4 x int32, 8 x int16, or 16 x int8. It is therefore possible for a single arithmetic vector task to operate on 8 "mediump" values in a single operation, and for OpenCL kernels operating on 8-bit luminance data to process 16 pixels per SIMD unit per clock cycle.

While I can't disclose the internal architecture of the arithmetic pipeline, our public performance data for each GPU can be used to give some idea of the number of maths units available. For example, the Mali-T760 with 16 cores is rated at 326 FP32 GFLOPS at 600MHz. This gives a total of 34 FP32 FLOPS per clock cycle for this shader core; it has two pipelines, so that's 17 FP32 FLOPS per pipeline per clock cycle. The available performance in terms of operations will increase for FP16/int16/int8 and decrease for FP64/int64 data types.

Texture Pipeline

The texture pipeline (T-pipe) is responsible for all memory access to do with textures. The texture pipeline can return one bilinear filtered texel per clock; trilinear filtering requires us to load samples from two different mipmaps in memory, so requires a second clock cycle to complete.

Load/Store Pipeline

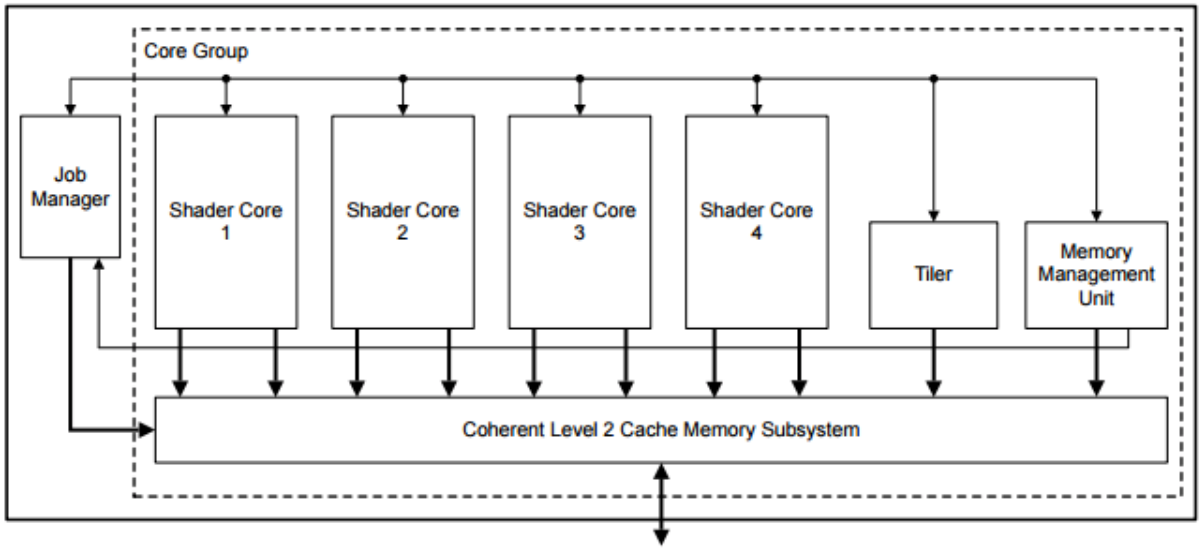
The load/store pipeline (LS-pipe) is responsible for all memory accesses which are not related to texturing. For graphics workloads this means reading attributes and writing varyings during vertex shading, and reading varyings during fragment shading. In general every instruction is a single memory access operation, although like the arithmetic pipeline they are vector operations and so could load an entire "highp" vec4 varying in a single instruction.

See https://community.arm.com/groups/arm-mali-graphics/blog/2014/03/12/the-mali-gpu-an-abstract-machine-part-3--the-shader-core.

U.S. Patent No. 8,760,454: Claim 3
"3. A unified shader comprising:"

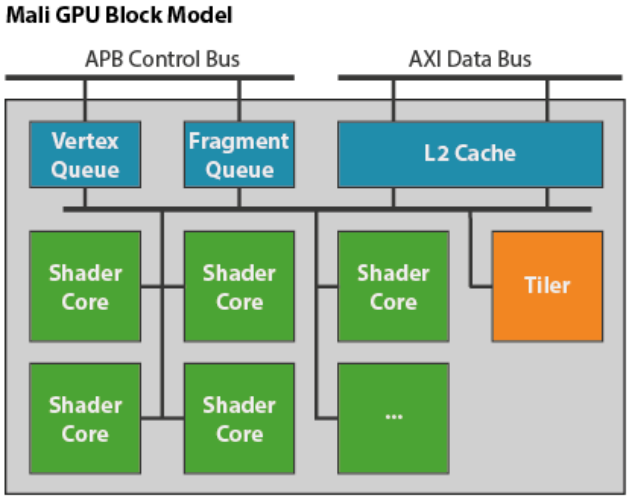
3. A unified shader comprising:	The LG Products include a unified shader. *See supr*a Claim 2.

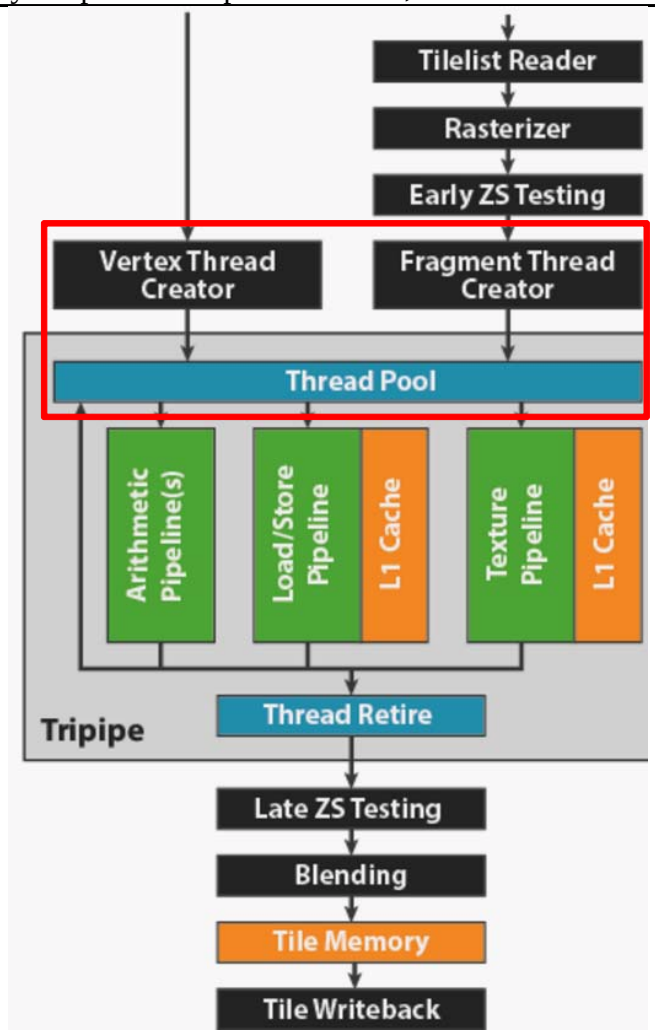
"a processor unit operative to perform vertex calculation operations and pixel calculation operations; and"

a processor unit operative to perform vertex calculation operations and pixel calculation operations; and	The LG Products include a processor unit operative to perform vertex calculation operations and pixel calculation operations. For example, the Mali GPU includes multiple shader cores, each of which handles vertex and fragment processing.  **Figure 1-1 Midgard architecture Mali GPU** **Shader cores** The shader cores handle the vertex and fragment processing stages of the graphics pipeline. The shader cores generate lists of primitives and accelerate the building of data structures, such as polygon lists and packed vertex data, for fragment processing. The shader cores also handle the rasterization and fragment processing stages of the graphics pipeline. They use the data structures and lists of primitives that are generated during vertex processing to produce the framebuffer result that appears on the screen. *See http*://malideveloper.arm.com/downloads/OpenGLES3.x/arm_mali_gpu_opengl_es_3-x_developer_guide_en.pdf.

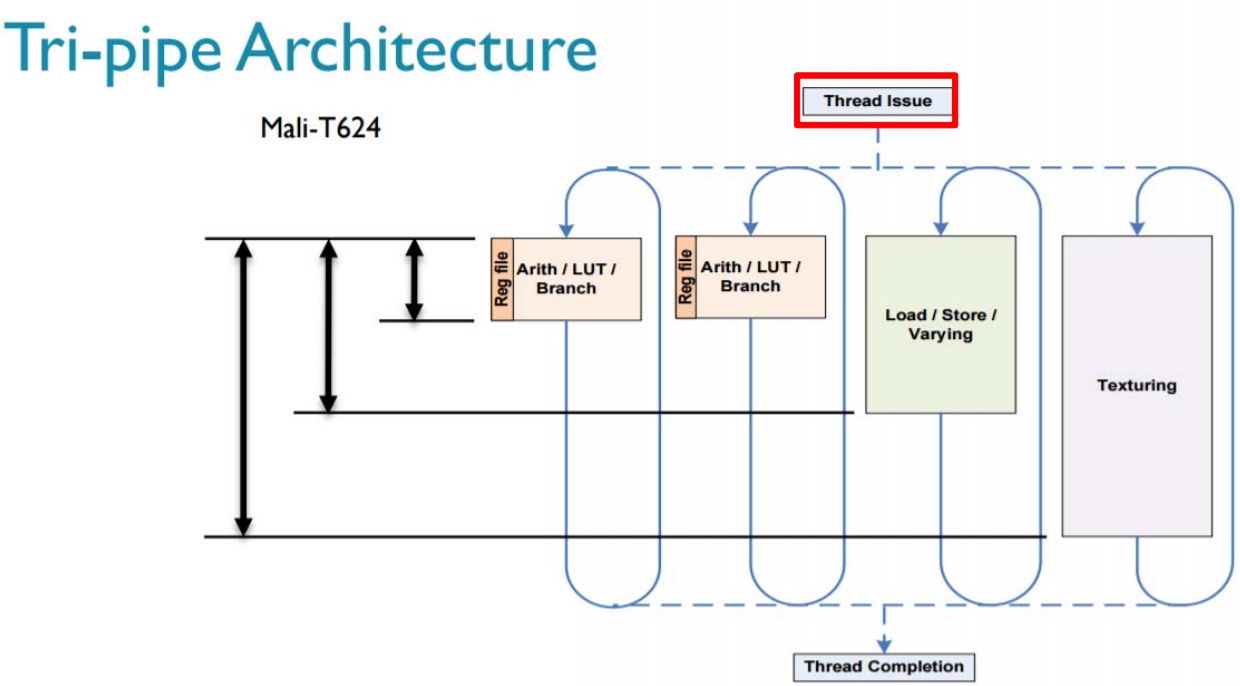
"a processor unit operative to perform vertex calculation operations and pixel calculation operations; and"

	1.5.1 About the Mali™ GPU families There are families of Mali GPUs: the Utgard architecture family, and the Midgard architecture family. **The Midgard architecture family** The Midgard architecture family of Mali GPUs have unified shader cores that perform vertex, fragment, and compute processing.The Midgard architecture Mali GPUs support OpenGL ES versions 1.1, 2.0, 3.0, 3.1, 3.2, and Vulkan. They also support compute applications with OpenCL 1.1, 1.2 and Renderscript. **The Utgard architecture family** The Utgard architecture family of Mali GPUs have a vertex processor and one or more fragment processors. They are used for graphics-only applications with OpenGL ES 1.1 and 2.0. **Note** AEP and OpenGL ES 3.0 to 3.2 do not work on Utgard GPUs. *See http*://malideveloper.arm.com/downloads/OpenGLES3.x/arm_mali_gpu_opengl_es_3-x_developer_guide_en.pdf.

shared resources, operatively coupled to the processor unit;	The LG Products include shared resources, operatively coupled to the processor unit. For example, the Mali GPUs include the Mali GPU includes a Vertex Queue and Fragment Queue coupled to the shader cores and the Thread Pool, Compute Thread Creator, Load/Store Pipe, Caches, and registers coupled to the Tri Pipe.  *See* The Mali GPU: An Abstract Machine, Part 3 - The Midgard Shader Core, https://community.arm.com/groups/arm-mali-graphics/blog/2014/03/12/the-mali-gpu-an-abstract-machine-part-3--the-shader-core.



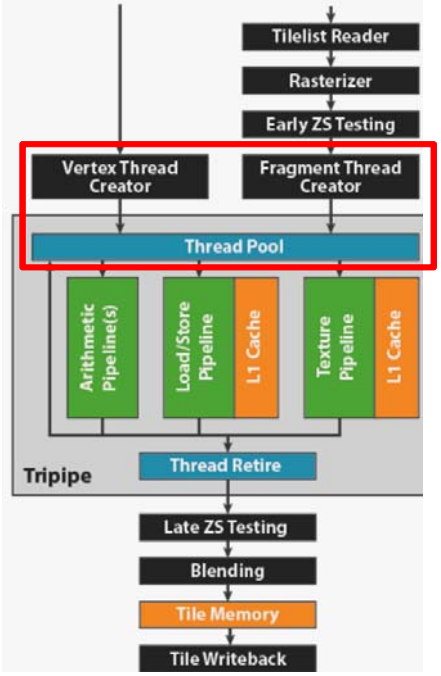
See https://community.arm.com/groups/arm-mali-graphics/blog/2014/03/12/the-mali-gpu-an-abstract-machine-part-3--the-shader-core.



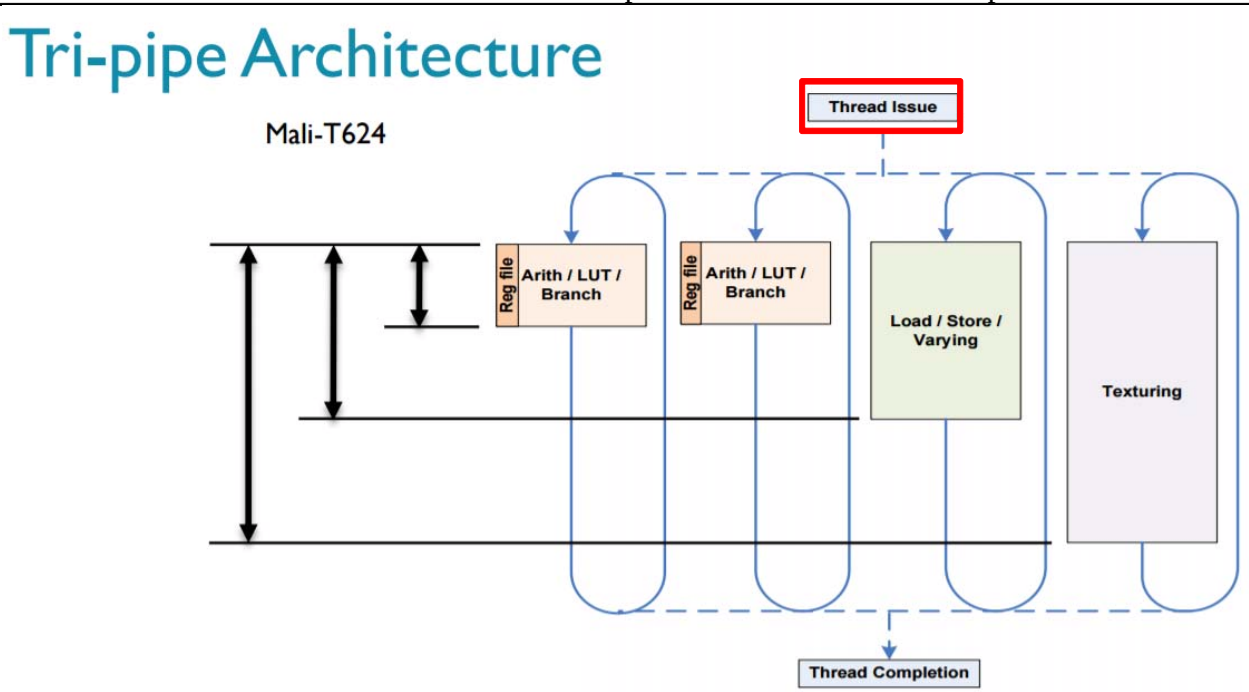
See ARM Midgard Architecture, p.11, *available at* http://malideveloper.arm.com/downloads/ARM_Game_Developer_Days/PDFs/2-Mali-GPU-architecture-overview-and-tile-local-storage.pdf.

U.S. Patent No. 8,760,454: Claim 3

"the processor unit operative to use the shared resources for either vertex data or pixel information and operative to perform pixel calculation operations until enough shared resources become available and then use the shared resources to perform vertex calculation operations."

the processor unit operative to use the shared resources for either vertex data or pixel information and operative to perform pixel calculation operations until enough shared resources become available and then use the shared resources to perform vertex calculation operations.	The LG Products include the processor unit operative to use the shared resources for either vertex data or pixel information and operative to perform pixel calculation operations until enough shared resources become available and then use the shared resources to perform vertex calculation operations. The processor unit is operative to use the shared resources for either vertex data or pixel information. For example, the Mali GPUs include the Vertex Thread Creator, Fragment Thread Creator, Compute Thread Creator, the Thread Pool (also known as the Thread Issue), which feed the processor unit with instructions for the processor to execute vertex and pixel operations.  *See* https://community.arm.com/groups/arm-mali-graphics/blog/2014/03/12/the-mali-gpu-an-abstract-machine-part-3--the-shader-core.

"the processor unit operative to use the shared resources for either vertex data or pixel information and operative to perform pixel calculation operations until enough shared resources become available and then use the shared resources to perform vertex calculation operations."

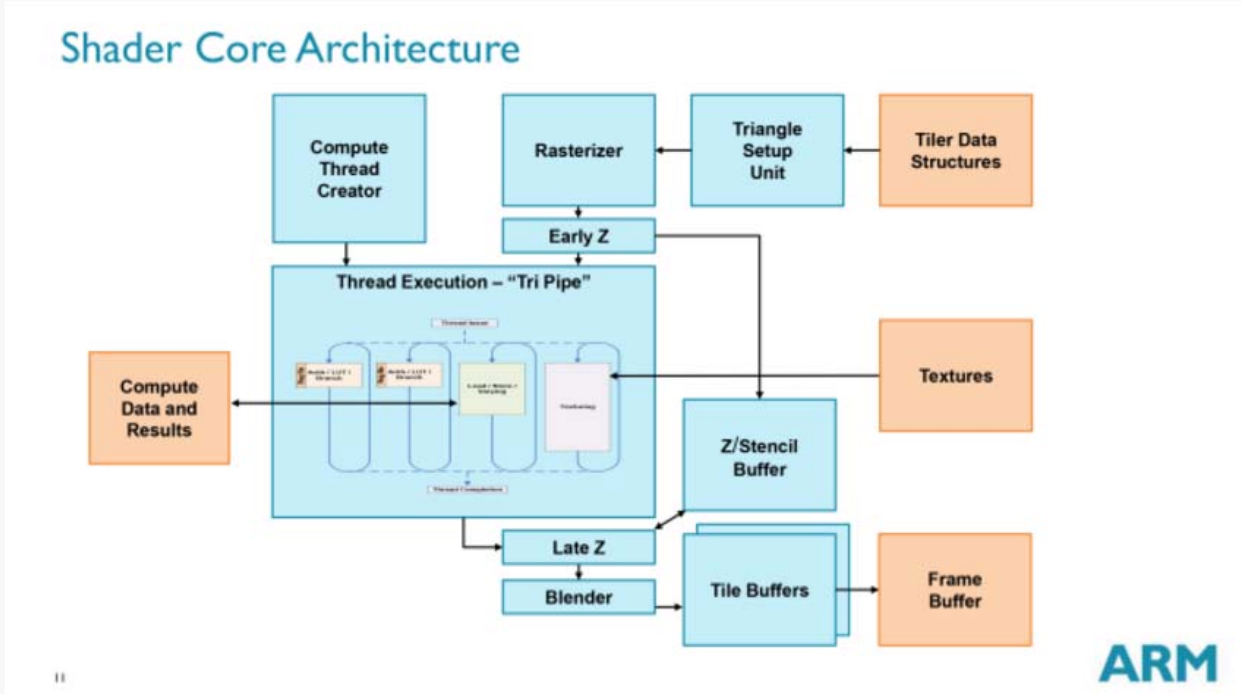


See ARM Midgard Architecture, p.11, *available at* http://malideveloper.arm.com/downloads/ARM_Game_Developer_Days/PDFs/2-Mali-GPU-architecture-overview-and-tile-local-storage.pdf.

"the processor unit operative to use the shared resources for either vertex data or pixel information and operative to perform pixel calculation operations until enough shared resources become available and then use the shared resources to perform vertex calculation operations."

The Midgard Architecture

Diving in to the Mali architecture, we'll start with a high level overview of the architecture. What we're looking at here is a single Midgard shader core, which despite the "shader" name actually contains a whole lot more. A shader core in this context contains the actual shader core within one of Midgard's "tri pipe" shader blocks, but also contains a triangle setup unit, rasterizer, Z & stencil hardware, a ROP/blender, tiling hardware, and **a compute thread creator specifically for feeding a tri pipe with compute workloads.**



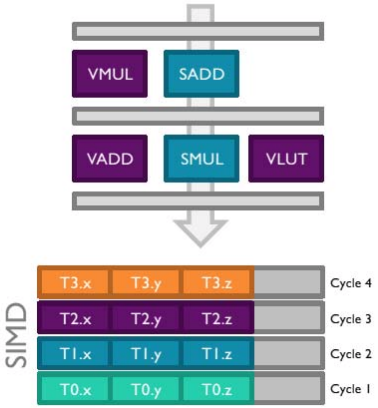
See Ryan Smith, ARM's Mali Midgard Architecture Explored, http://www.anandtech.com/show/8234/arms-mali-midgard-architecture-explored/4.

The processor unit is operative to perform pixel calculation operations until enough shared resources become available and then use the shared resources to perform vertex calculation operations. For example, the arithmetic and load/store pipelines "can progress under a pending Texture instruction."

"the processor unit operative to use the shared resources for either vertex data or pixel information and operative to perform pixel calculation operations until enough shared resources become available and then use the shared resources to perform vertex calculation operations."

Mali-T880 GPU Shader Core - Arithmetic pipeline

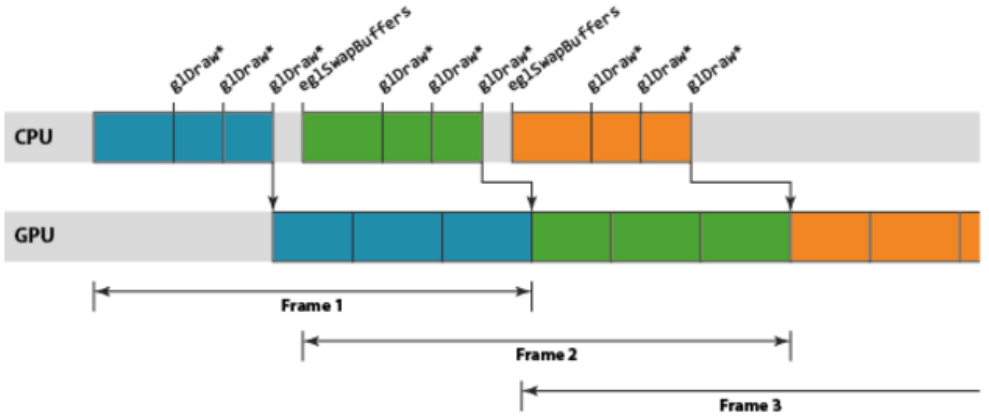
- Arithmetic ISA on Midgard is SIMD + VLIW
  - Three vector units (128-bit datapath)
  - 4-lane FP32 or 8-lane FP16 for graphics
  - 16-lane int8 for compute
  - Two scalar units (32-bit datapath)
- One thread at a time executes in each pipeline stage



- Limited amount of out-of-order parallelism
  - Arith and Load/Store can progress under a pending Texture instruction

See ARM, The ARM Mali –T880 Mobile GPU, p.19, *available at* http://www.hotchips.org/wp-content/uploads/hc_archives/hc27/HC27.25-Tuesday-Epub/HC27.25.50-GPU-Epub/HC27.25.531-Mali-T880-Bratt-ARM-2015_08_23.pdf/.

To remove this idle time we use the OpenGL ES driver to maintain the illusion of synchronous rendering behavior, while actually processing rendering and frame swaps asynchronously under the hood. By running asynchronously we can build a small backlog of work, allowing a pipeline to be created where the GPU is processing older workloads from one end of the pipeline, while the CPU is busy pushing new work into the other. The advantage of this approach is that, provided we keep the pipeline full, there is always work available to run on the GPU giving the best performance.



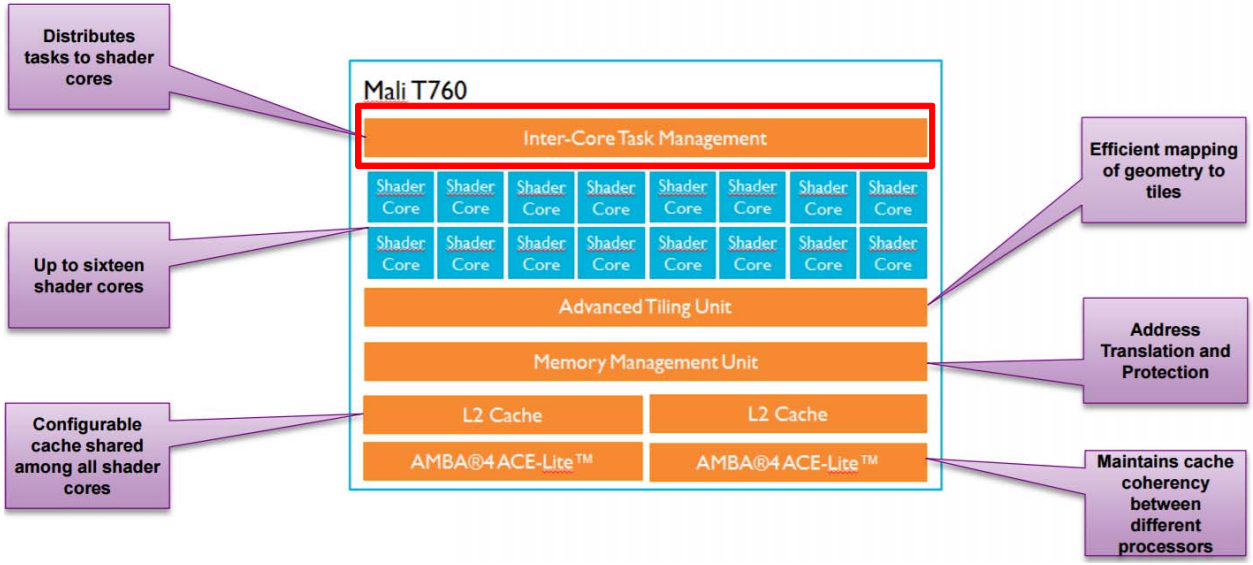
See https://community.arm.com/groups/arm-mali-graphics/blog/2014/02/03/the-mali-gpu-an-abstract-machine-part-1.

In addition, the Mali GPUs include the Inter-Core Task Management that assigns tasks to the shader cores.

> The shared hardware in Midgard is primarily concerned with managing the interaction of the shader cores, followed by providing the L2 cache and all further memory interfaces for accessing main memory and/or the CPU cache. In the case of Mali-T760 there is **1 task management unit** and memory management unit, but 2 sets of L2 cache and the AMBA interface that connects the GPU to the rest of the system.

See http://www.anandtech.com/show/8234/arms-mali-midgard-architecture-explored/4.

"the processor unit operative to use the shared resources for either vertex data or pixel information and operative to perform pixel calculation operations until enough shared resources become available and then use the shared resources to perform vertex calculation operations."

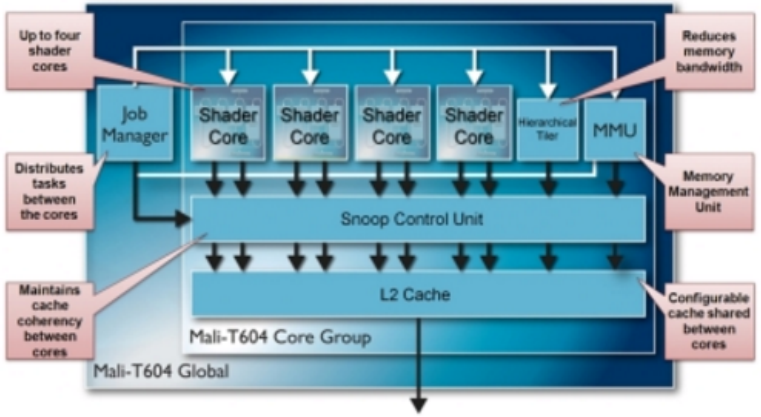


See http://malideveloper.arm.com/downloads/ARM_Game_Developer_Days/PDFs/2-Mali-GPU-architecture-overview-and-tile-local-storage.pdf.

"the processor unit operative to use the shared resources for either vertex data or pixel information and operative to perform pixel calculation operations until enough shared resources become available and then use the shared resources to perform vertex calculation operations."

Adding a bit more detail, the T604 includes four shader cores, each of which contains two arithmetic pipelines, one texturing pipeline, and one load/store unit. The four shaders share a coherent L2 cache, an MMU, a tiler, and a Job Manager. This latter block is a key component because the shaders are multithreaded. The Job Manager can dynamically move threads among the shaders.

Mali-T604 high-level architecture

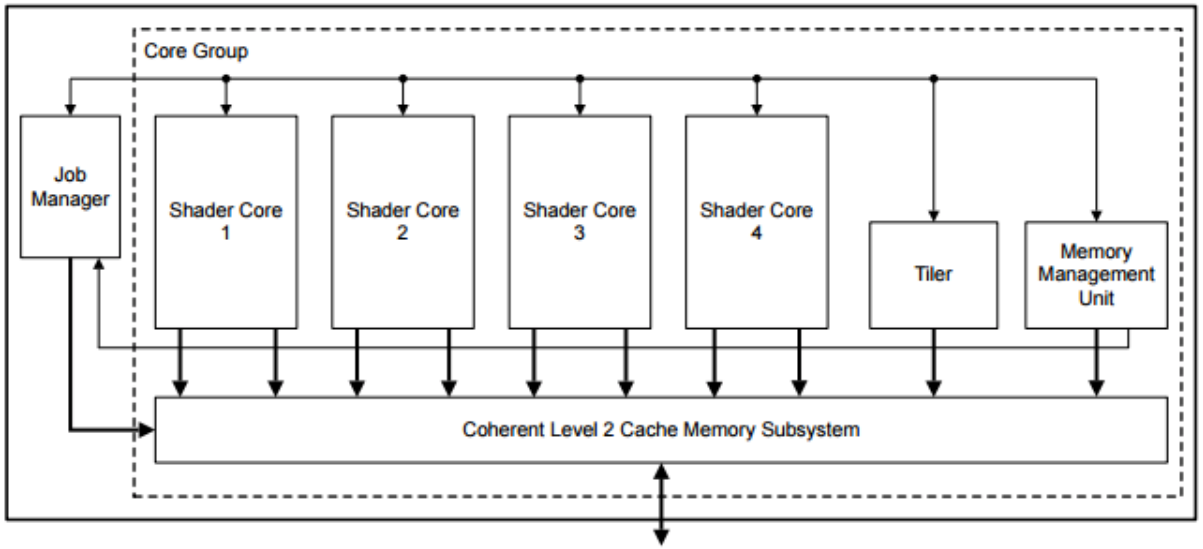


This dynamic load allocation, in turn, allows the host system to exert considerable control over the energy consumption of the core —vital to high-performance mobile use. But it also requires that the threads be light-weight. And that puts stress on the L2 cache, which must hold any state not local to the shader.

See http://www.eetimes.com/document.asp?doc_id=1278897.

4. A unified shader comprising:	The LG Products include a unified shader. *See supra* Claim 2.

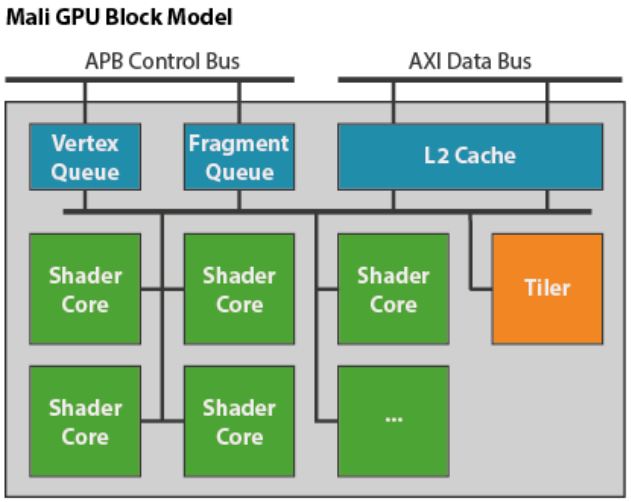
"a processor unit operative to perform vertex calculation operations and pixel calculation operations; and"

a processor unit operative to perform vertex calculation operations and pixel calculation operations; and	The LG Products include a processor unit operative to perform vertex calculation operations and pixel calculation operations. For example, the Mali GPU includes multiple shader cores, each of which handles vertex and fragment processing.  **Figure 1-1 Midgard architecture Mali GPU** **Shader cores** The shader cores handle the vertex and fragment processing stages of the graphics pipeline. The shader cores generate lists of primitives and accelerate the building of data structures, such as polygon lists and packed vertex data, for fragment processing. The shader cores also handle the rasterization and fragment processing stages of the graphics pipeline. They use the data structures and lists of primitives that are generated during vertex processing to produce the framebuffer result that appears on the screen. *See http*://malideveloper.arm.com/downloads/OpenGLES3.x/arm_mali_gpu_opengl_es_3-x_developer_guide_en.pdf.

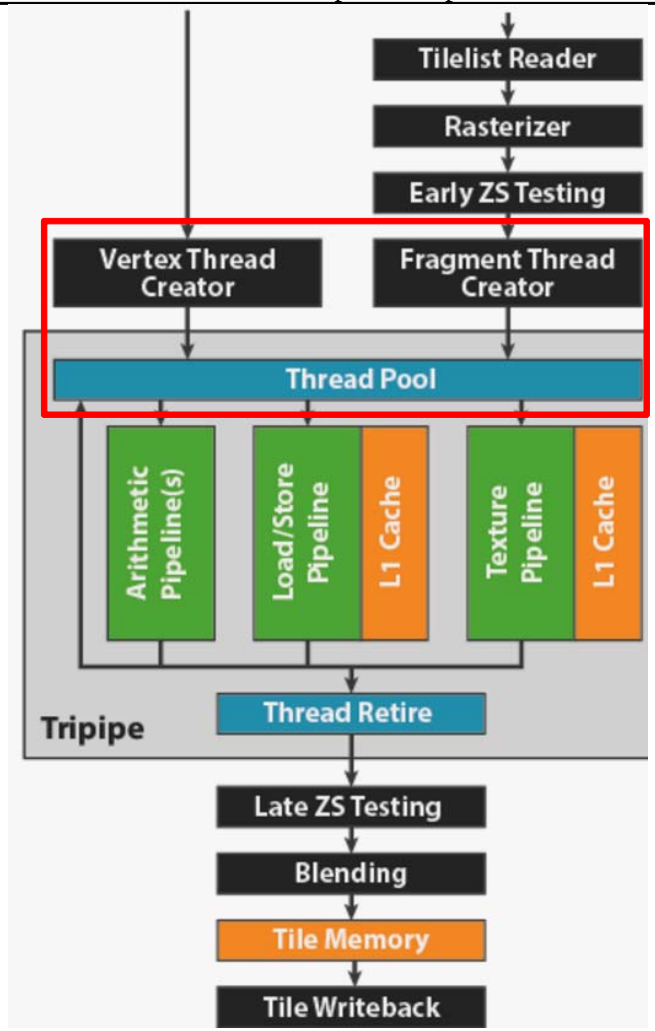
"a processor unit operative to perform vertex calculation operations and pixel calculation operations; and"

	1.5.1 About the Mali™ GPU families There are families of Mali GPUs: the Utgard architecture family, and the Midgard architecture family. **The Midgard architecture family** The Midgard architecture family of Mali GPUs have unified shader cores that perform vertex, fragment, and compute processing.The Midgard architecture Mali GPUs support OpenGL ES versions 1.1, 2.0, 3.0, 3.1, 3.2, and Vulkan. They also support compute applications with OpenCL 1.1, 1.2 and Renderscript. **The Utgard architecture family** The Utgard architecture family of Mali GPUs have a vertex processor and one or more fragment processors. They are used for graphics-only applications with OpenGL ES 1.1 and 2.0. **Note** AEP and OpenGL ES 3.0 to 3.2 do not work on Utgard GPUs. *See http*://malideveloper.arm.com/downloads/OpenGLES3.x/arm_mali_gpu_opengl_es_3-x_developer_guide_en.pdf.

"the processor unit operative to use the shared resources for either vertex data or pixel information and operative to perform vertex calculation operations until enough shared resources become available and then use the shared resources to perform pixel calculation operations."

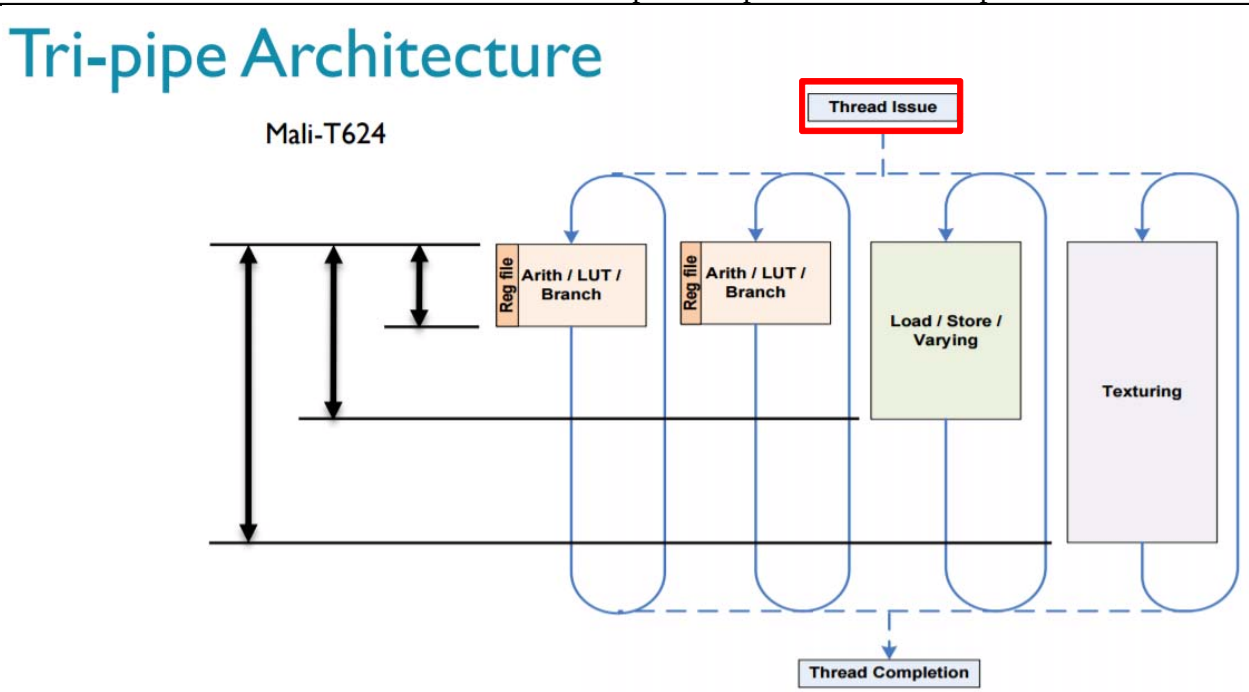
shared resources, operatively coupled to the processor unit;	The LG Products include shared resources, operatively coupled to the processor unit. For example, he Mali GPUs include the Mali GPU includes a Vertex Queue and Fragment Queue coupled to the shader cores and the Thread Pool, Compute Thread Creator, Load/Store Pipe, Caches, and registers coupled to the Tri Pipe.  *See* The Mali GPU: An Abstract Machine, Part 3 - The Midgard Shader Core, https://community.arm.com/groups/arm-mali-graphics/blog/2014/03/12/the-mali-gpu-an-abstract-machine-part-3--the-shader-core.

"the processor unit operative to use the shared resources for either vertex data or pixel information and operative to perform vertex calculation operations until enough shared resources become available and then use the shared resources to perform pixel calculation operations."



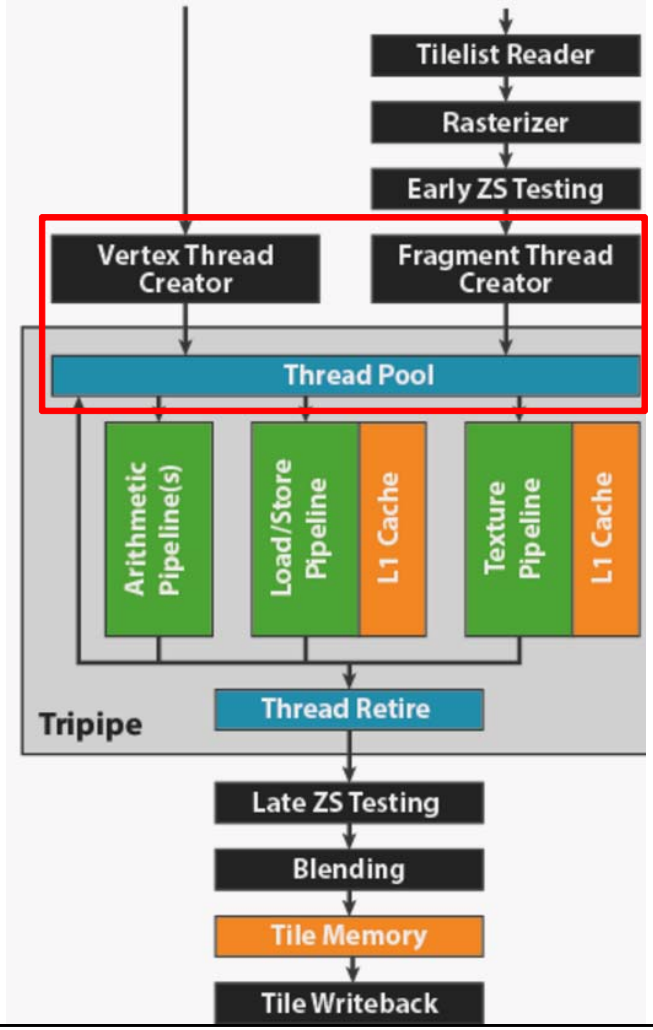
See https://community.arm.com/groups/arm-mali-graphics/blog/2014/03/12/the-mali-gpu-an-abstract-machine-part-3--the-shader-core.

"the processor unit operative to use the shared resources for either vertex data or pixel information and operative to perform vertex calculation operations until enough shared resources become available and then use the shared resources to perform pixel calculation operations."



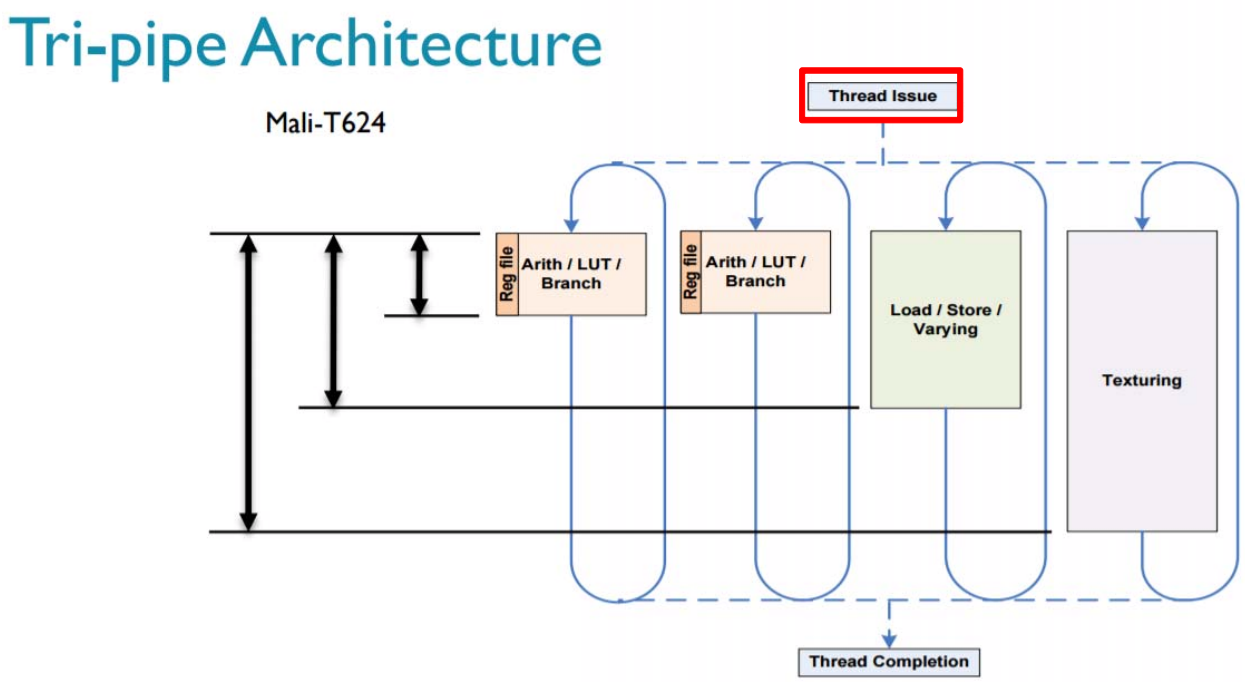
See ARM Midgard Architecture, p.11, *available at* http://malideveloper.arm.com/downloads/ARM_Game_Developer_Days/PDFs/2-Mali-GPU-architecture-overview-and-tile-local-storage.pdf.

"the processor unit operative to use the shared resources for either vertex data or pixel information and operative to perform vertex calculation operations until enough shared resources become available and then use the shared resources to perform pixel calculation operations."

the processor unit operative to use the shared resources for either vertex data or pixel information and operative to perform vertex calculation operations until enough shared resources become available and then use the shared resources to perform pixel calculation operations.	The LG Products include the processor unit operative to use the shared resources for either vertex data or pixel information and operative to perform vertex calculation operations until enough shared resources become available and then use the shared resources to perform pixel calculation operations. The processor unit is operative to use the shared resources for either vertex data or pixel information. For example, the Mali GPUs include the Vertex Thread Creator, Fragment Thread Creator, Compute Thread Creator, the Thread Pool (also known as the Thread Issue), which feed the processor unit with instructions for the processor to execute vertex and pixel operations. 

"the processor unit operative to use the shared resources for either vertex data or pixel information and operative to perform vertex calculation operations until enough shared resources become available and then use the shared resources to perform pixel calculation operations."

See https://community.arm.com/groups/arm-mali-graphics/blog/2014/03/12/the-mali-gpu-an-abstract-machine-part-3--the-shader-core.

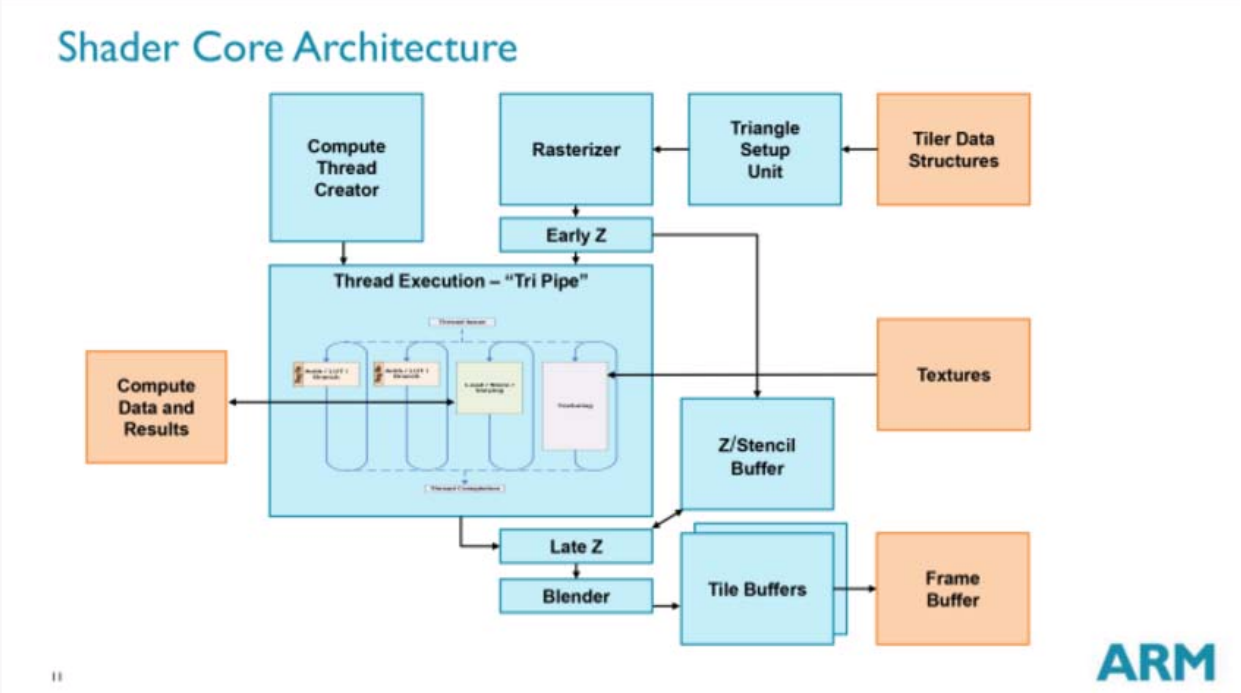


See ARM Midgard Architecture, p.11, *available at* http://malideveloper.arm.com/downloads/ARM_Game_Developer_Days/PDFs/2-Mali-GPU-architecture-overview-and-tile-local-storage.pdf.

"the processor unit operative to use the shared resources for either vertex data or pixel information and operative to perform vertex calculation operations until enough shared resources become available and then use the shared resources to perform pixel calculation operations."

The Midgard Architecture

Diving in to the Mali architecture, we'll start with a high level overview of the architecture. What we're looking at here is a single Midgard shader core, which despite the "shader" name actually contains a whole lot more. A shader core in this context contains the actual shader core within one of Midgard's "tri pipe" shader blocks, but also contains a triangle setup unit, rasterizer, Z & stencil hardware, a ROP/blender, tiling hardware, and **a compute thread creator specifically for feeding a tri pipe with compute workloads.**



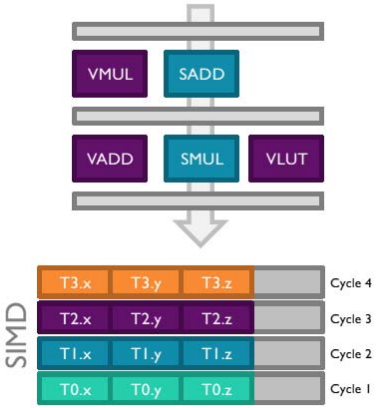
See Ryan Smith, ARM's Mali Midgard Architecture Explored, http://www.anandtech.com/show/8234/arms-mali-midgard-architecture-explored/4.

The processor unit is operative to perform pixel calculation operations until enough shared resources become available and then use the shared resources to perform pixel calculation operations. For example, the arithmetic and load/store pipelines "can progress under a pending Texture instruction."

"the processor unit operative to use the shared resources for either vertex data or pixel information and operative to perform vertex calculation operations until enough shared resources become available and then use the shared resources to perform pixel calculation operations."

Mali-T880 GPU Shader Core - Arithmetic pipeline

- Arithmetic ISA on Midgard is SIMD + VLIW
  - Three vector units (128-bit datapath)
  - 4-lane FP32 or 8-lane FP16 for graphics
  - 16-lane int8 for compute
  - Two scalar units (32-bit datapath)
- One thread at a time executes in each pipeline stage

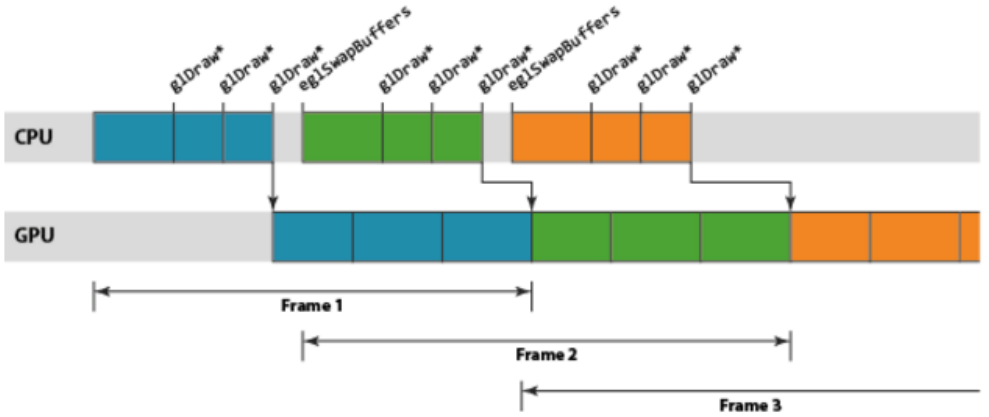


- Limited amount of out-of-order parallelism
  - Arith and Load/Store can progress under a pending Texture instruction

See ARM, The ARM Mali –T880 Mobile GPU, p.19, *available at* http://www.hotchips.org/wp-content/uploads/hc_archives/hc27/HC27.25-Tuesday-Epub/HC27.25.50-GPU-Epub/HC27.25.531-Mali-T880-Bratt-ARM-2015_08_23.pdf/.

"the processor unit operative to use the shared resources for either vertex data or pixel information and operative to perform vertex calculation operations until enough shared resources become available and then use the shared resources to perform pixel calculation operations."

To remove this idle time we use the OpenGL ES driver to maintain the illusion of synchronous rendering behavior, while actually processing rendering and frame swaps asynchronously under the hood. By running asynchronously we can build a small backlog of work, allowing a pipeline to be created where the GPU is processing older workloads from one end of the pipeline, while the CPU is busy pushing new work into the other. The advantage of this approach is that, provided we keep the pipeline full, there is always work available to run on the GPU giving the best performance.



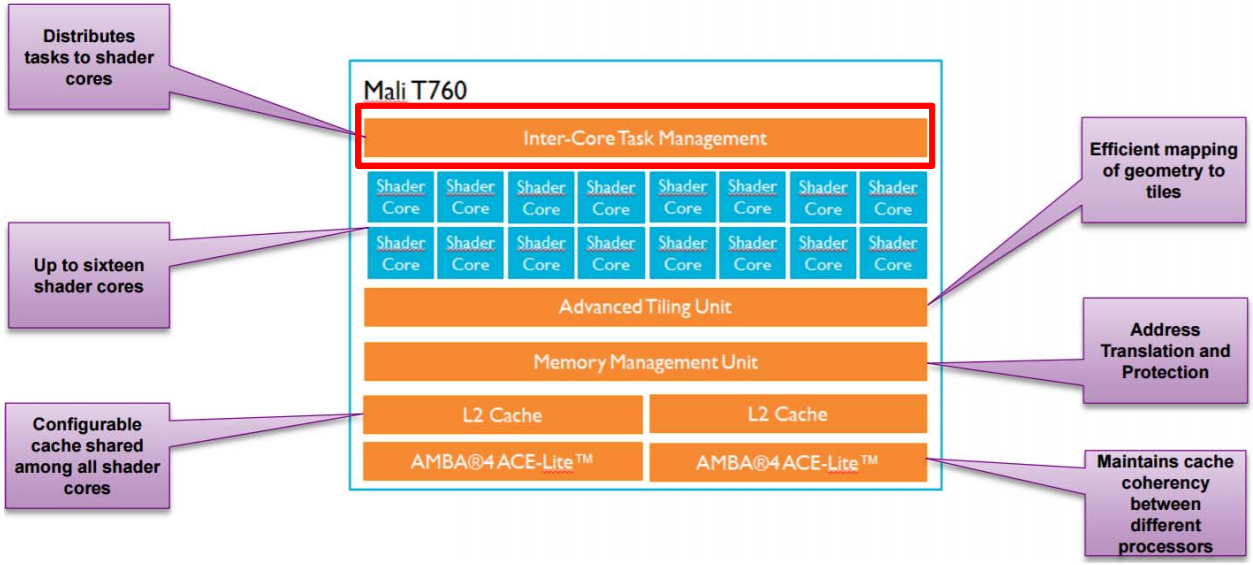
See https://community.arm.com/groups/arm-mali-graphics/blog/2014/02/03/the-mali-gpu-an-abstract-machine-part-1.

In addition, the Mali GPUs include the Inter-Core Task Management that assigns tasks to the shader cores.

> The shared hardware in Midgard is primarily concerned with managing the interaction of the shader cores, followed by providing the L2 cache and all further memory interfaces for accessing main memory and/or the CPU cache. In the case of Mali-T760 there is **1 task management unit** and memory management unit, but 2 sets of L2 cache and the AMBA interface that connects the GPU to the rest of the system.

See http://www.anandtech.com/show/8234/arms-mali-midgard-architecture-explored/4.

"the processor unit operative to use the shared resources for either vertex data or pixel information and operative to perform vertex calculation operations until enough shared resources become available and then use the shared resources to perform pixel calculation operations."

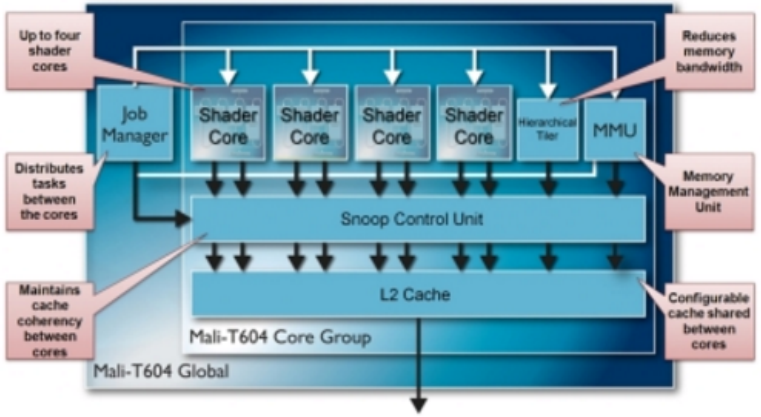


See http://malideveloper.arm.com/downloads/ARM_Game_Developer_Days/PDFs/2-Mali-GPU-architecture-overview-and-tile-local-storage.pdf.

"the processor unit operative to use the shared resources for either vertex data or pixel information and operative to perform vertex calculation operations until enough shared resources become available and then use the shared resources to perform pixel calculation operations."

Adding a bit more detail, the T604 includes four shader cores, each of which contains two arithmetic pipelines, one texturing pipeline, and one load/store unit. The four shaders share a coherent L2 cache, an MMU, a tiler, and a Job Manager. This latter block is a key component because the shaders are multithreaded. The Job Manager can dynamically move threads among the shaders.

Mali-T604 high-level architecture

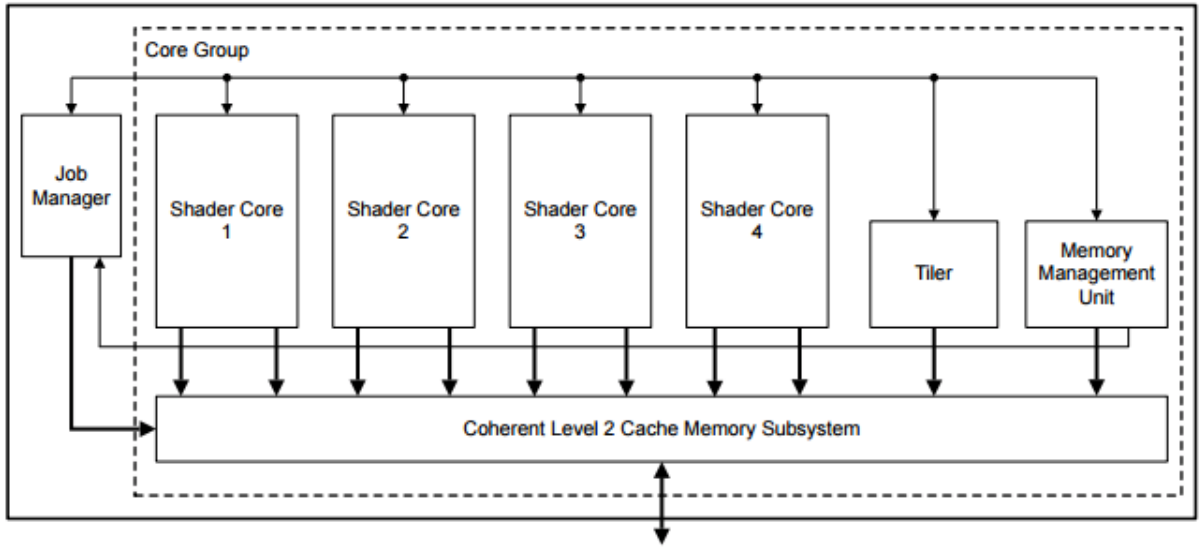


This dynamic load allocation, in turn, allows the host system to exert considerable control over the energy consumption of the core —vital to high-performance mobile use. But it also requires that the threads be light-weight. And that puts stress on the L2 cache, which must hold any state not local to the shader.

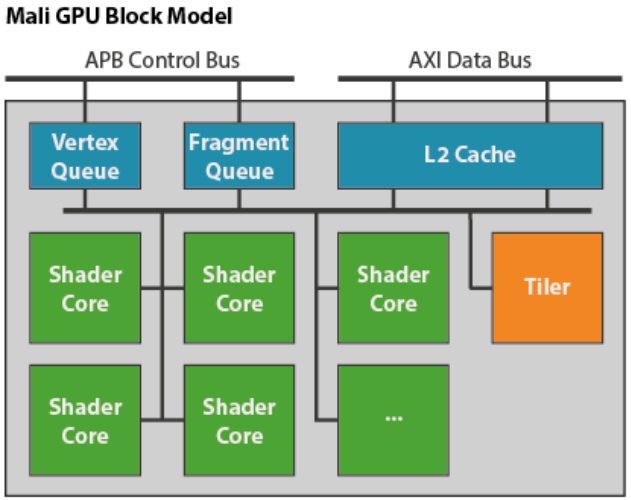
See http://www.eetimes.com/document.asp?doc_id=1278897.

U.S. Patent No. 8,760,454: Claim 5
"5. A unified shader comprising:”

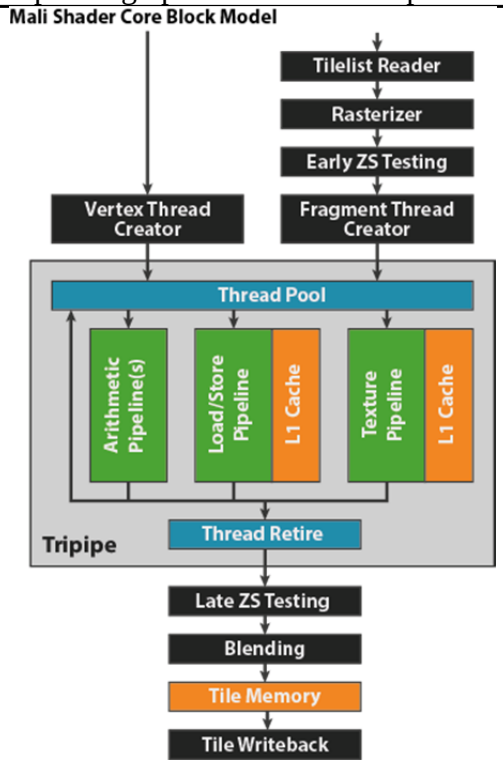
5. A unified shader comprising:	The LG Products include a unified shader. *See supra* Claim 2.

a processor unit;	The LG Products include a processor unit. For example, the LG Products include a plurality of shader cores.  **Figure 1-1 Midgard architecture Mali GPU** **Shader cores** The shader cores handle the vertex and fragment processing stages of the graphics pipeline. The shader cores generate lists of primitives and accelerate the building of data structures, such as polygon lists and packed vertex data, for fragment processing. The shader cores also handle the rasterization and fragment processing stages of the graphics pipeline. They use the data structures and lists of primitives that are generated during vertex processing to produce the framebuffer result that appears on the screen. *See* http://malideveloper.arm.com/downloads/OpenGLES3.x/arm_mali_gpu_opengl_es_3-x_developer_guide_en.pdf.

"a sequencer coupled to the processor unit, the sequencer maintaining instructions operative to cause the processor unit to execute vertex calculation and pixel calculation operations on selected data maintained in a store depending upon an amount of space available in the store."

a sequencer coupled to the processor unit, the sequencer maintaining instructions operative to cause the processor unit to execute vertex calculation and pixel calculation operations on selected data maintained in a store depending upon an amount of space available in the store.	The LG Products include a sequencer coupled to the processor unit, the sequencer maintaining instructions operative to cause the processor unit to execute vertex calculations and pixel calculation operations on selected data maintained in a store depending upon an amount of space available in the store. The LG Products include a sequencer coupled to the processor unit. For example, the Mali GPU includes a Vertex Queue and Fragment Queue shared by the shader cores and a Thread Pool, Load/Store Pipe, Caches, and registers within each shader core.  *See* The Mali GPU: An Abstract Machine, Part 3 - The Midgard Shader Core, https://community.arm.com/groups/arm-mali-graphics/blog/2014/03/12/the-mali-gpu-an-abstract-machine-part-3--the-shader-core.

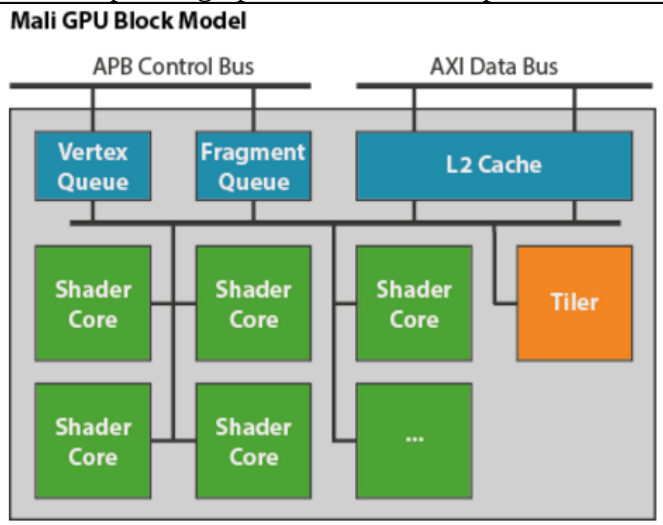
"a sequencer coupled to the processor unit, the sequencer maintaining instructions operative to cause the processor unit to execute vertex calculation and pixel calculation operations on selected data maintained in a store depending upon an amount of space available in the store."



See The Mali GPU: An Abstract Machine, Part 3 - The Midgard Shader Core, https://community.arm.com/groups/arm-mali-graphics/blog/2014/03/12/the-mali-gpu-an-abstract-machine-part-3--the-shader-core.

The sequencer maintains instructions operative to cause the processor unit to execute vertex calculations and pixel calculation operations on selected data maintained in a store depending upon an amount of space available in the store. For example, the vertex queue stores the vertex workloads while the fragment queue stores the fragment workloads.

"a sequencer coupled to the processor unit, the sequencer maintaining instructions operative to cause the processor unit to execute vertex calculation and pixel calculation operations on selected data maintained in a store depending upon an amount of space available in the store."



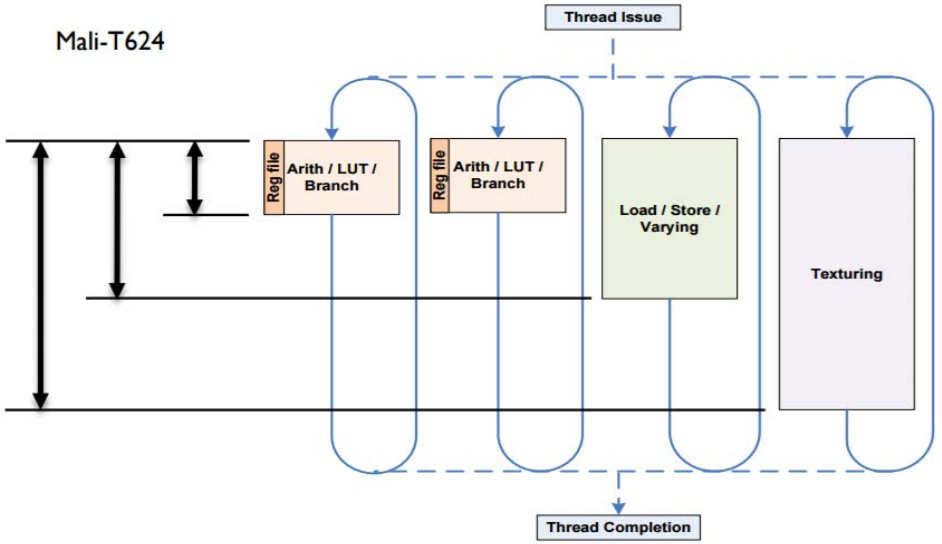
The graphics work for the GPU is queued in a pair of queues, one for vertex/tiling workloads and one for fragment workloads, with all work for one render target being submitted as a single submission into each queue. Workloads from both queues can be processed by the GPU at the same time, so vertex processing and fragment processing for different render targets can be running in parallel (see the first blog for more details on this pipelining methodology). The workload for a single render target is broken into smaller pieces and distributed across all of the shader cores in the GPU, or in the case of tiling workloads (see the second blog in this series for an overview of tiling) a fixed function tiling unit.

See The Mali GPU: An Abstract Machine, Part 3 - The Midgard Shader Core, https://community.arm.com/groups/arm-mali-graphics/blog/2014/03/12/the-mali-gpu-an-abstract-machine-part-3--the-shader-core.

The Thread Pool issues instructions to the three types of pipelines in the shader core for processing.

"a sequencer coupled to the processor unit, the sequencer maintaining instructions operative to cause the processor unit to execute vertex calculation and pixel calculation operations on selected data maintained in a store depending upon an amount of space available in the store."

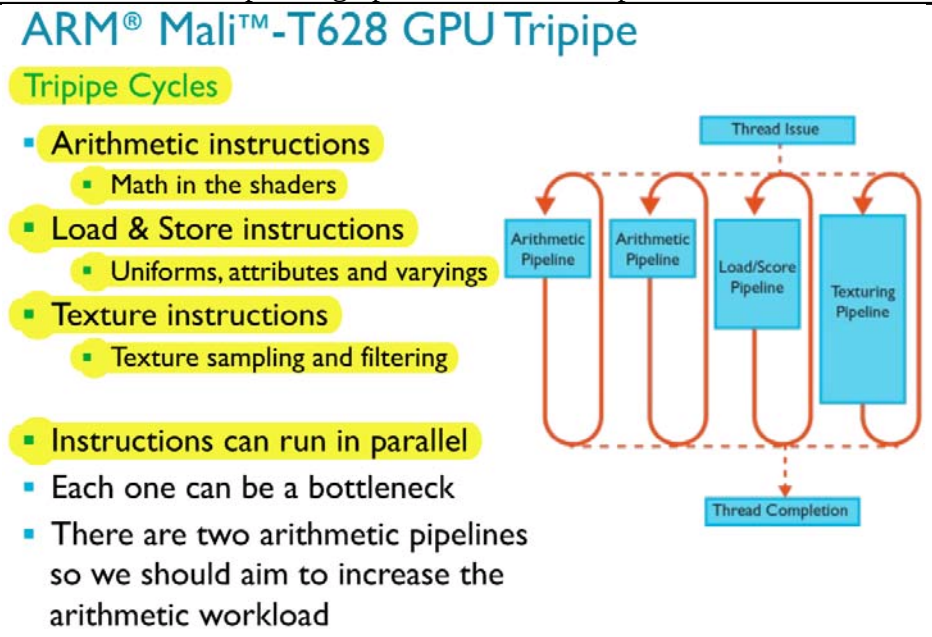
Tri-pipe Architecture



- Unified shader architecture
  - Fragment and vertex shaders
  - Geometry and compute shaders
- Very high throughput graphics
- Multiple parallel pipelines
  - Two low-latency arithmetic pipes
  - 256 simultaneous threads
  - Low-latency for computation

See http://malideveloper.arm.com/downloads/ARM_Game_Developer_Days/PDFs/2-Mali-GPU-architecture-overview-and-tile-local-storage.pdf.

"a sequencer coupled to the processor unit, the sequencer maintaining instructions operative to cause the processor unit to execute vertex calculation and pixel calculation operations on selected data maintained in a store depending upon an amount of space available in the store."



See ARM, ARM Tools Part 2, Best Optimization Practices for Mobile Platforms, p.11, *available at* http://malideveloper.arm.com/downloads/ARM_Game_Developer_Days/PDFs/6%20-%20ARM%20Tools%20Part%202-%20Best%20Optimization%20Practices%20for%20Mobile%20Platforms.pdf.

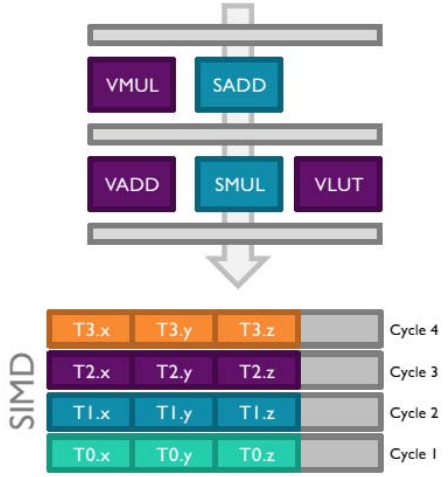
> **Massively Multi-threaded Machine**
>
> Unlike a traditional CPU architecture, where you will typically only have a single thread of execution at a time on a single core, the tripipe is a massively multi-threaded processing engine. There may well be hundreds of hardware threads running at the same time in the tripipe, with one thread created for each vertex or fragment which is shaded. This large number of threads exists to hide memory latency; it doesn't matter if some threads are stalled waiting for memory, as long as at least one thread is available to execute then we maintain efficient execution.

See https://community.arm.com/groups/arm-mali-graphics/blog/2014/03/12/the-mali-gpu-an-abstract-machine-part-3--the-shader-core.

"a sequencer coupled to the processor unit, the sequencer maintaining instructions operative to cause the processor unit to execute vertex calculation and pixel calculation operations on selected data maintained in a store depending upon an amount of space available in the store."

Mali-T880 GPU Shader Core - Arithmetic pipeline

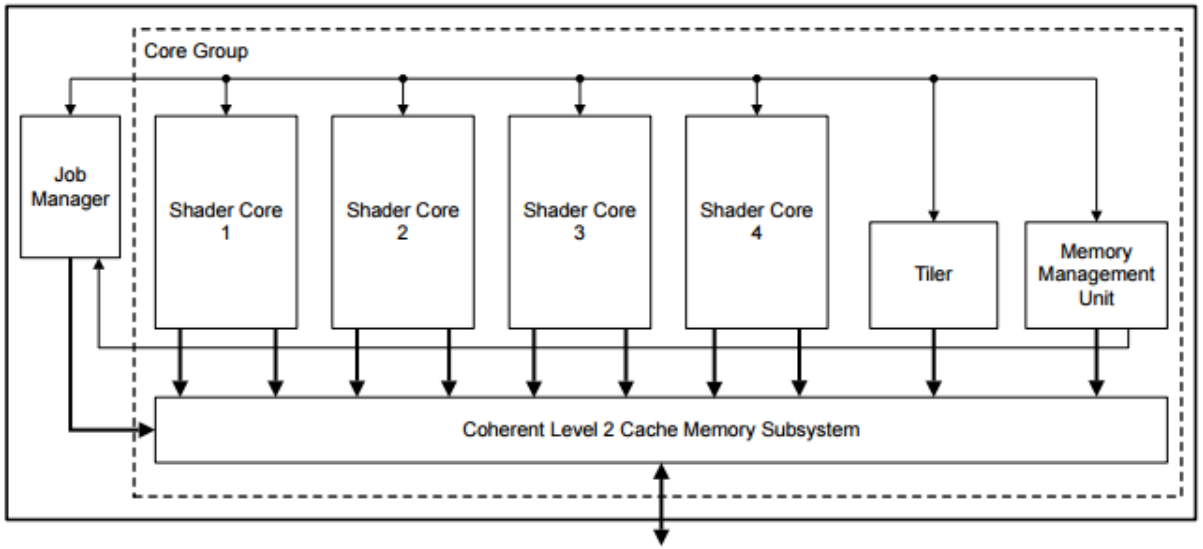
- Arithmetic ISA on Midgard is SIMD + VLIW
  - Three vector units (128-bit datapath)
  - 4-lane FP32 or 8-lane FP16 for graphics
  - 16-lane int8 for compute
  - Two scalar units (32-bit datapath)
- One thread at a time executes in each pipeline stage
- Limited amount of out-of-order parallelism
  - Arith and Load/Store can progress under a pending Texture instruction



See ARM, The ARM Mali –T880 Mobile GPU, p.19, *available at* http://www.hotchips.org/wp-content/uploads/hc_archives/hc27/HC27.25-Tuesday-Epub/HC27.25.50-GPU-Epub/HC27.25.531-Mali-T880-Bratt-ARM-2015_08_23.pdf/.

11. A unified shader comprising:	The LG Products include a unified shader. *See supra* Claim 2.

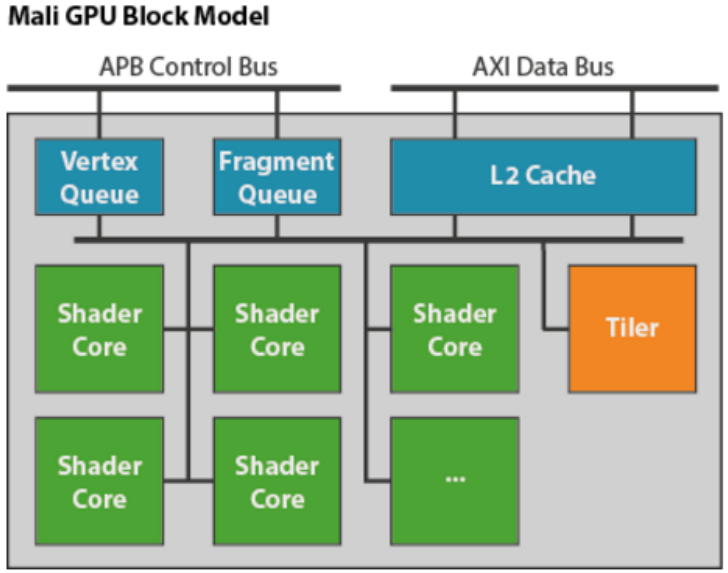
" a processor unit flexibly controlled to perform vertex manipulation operations and pixel manipulation operations based on vertex or pixel workload; and "

a processor unit flexibly controlled to perform vertex manipulation operations and pixel manipulation operations based on vertex or pixel workload; and	The LG Products include a processor unit flexibly controlled to perform vertex manipulation operations and pixel manipulation operations based on vertex or pixel workload. The LG Products include a processor unit that flexibly performs vertex manipulation operations and pixel manipulation operations.  **Figure 1-1 Midgard architecture Mali GPU** **Shader cores** The shader cores handle the vertex and fragment processing stages of the graphics pipeline. The shader cores generate lists of primitives and accelerate the building of data structures, such as polygon lists and packed vertex data, for fragment processing. The shader cores also handle the rasterization and fragment processing stages of the graphics pipeline. They use the data structures and lists of primitives that are generated during vertex processing to produce the framebuffer result that appears on the screen. *See http*://malideveloper.arm.com/downloads/OpenGLES3.x/arm_mali_gpu_opengl_es_3-x_developer_guide_en.pdf.

" a processor unit flexibly controlled to perform vertex manipulation operations and pixel manipulation operations based on vertex or pixel workload; and "

	1.5.1 About the Mali™ GPU families There are families of Mali GPUs: the Utgard architecture family, and the Midgard architecture family. **The Midgard architecture family** The Midgard architecture family of Mali GPUs have unified shader cores that perform vertex, fragment, and compute processing.The Midgard architecture Mali GPUs support OpenGL ES versions 1.1, 2.0, 3.0, 3.1, 3.2, and Vulkan. They also support compute applications with OpenCL 1.1, 1.2 and Renderscript. **The Utgard architecture family** The Utgard architecture family of Mali GPUs have a vertex processor and one or more fragment processors. They are used for graphics-only applications with OpenGL ES 1.1 and 2.0. **Note** AEP and OpenGL ES 3.0 to 3.2 do not work on Utgard GPUs. *See http*://malideveloper.arm.com/downloads/OpenGLES3.x/arm_mali_gpu_opengl_es_3-x_developer_guide_en.pdf. The processors unit performs vertex and pixel manipulation operations based on vertex or pixel workload.

" a processor unit flexibly controlled to perform vertex manipulation operations and pixel manipulation operations based on vertex or pixel workload; and "

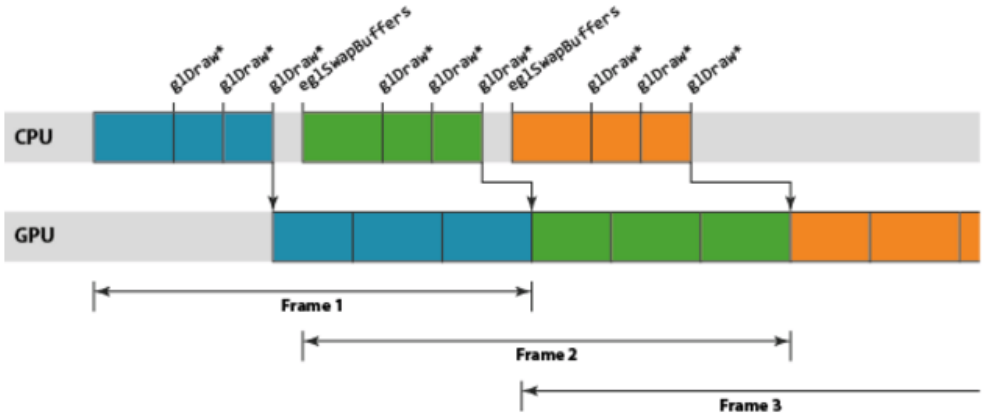


The graphics work for the GPU is queued in a pair of queues, one for vertex/tiling workloads and one for fragment workloads, with all work for one render target being submitted as a single submission into each queue. Workloads from both queues can be processed by the GPU at the same time, so vertex processing and fragment processing for different render targets can be running in parallel (see the first blog for more details on this pipelining methodology). The workload for a single render target is broken into smaller pieces and distributed across all of the shader cores in the GPU, or in the case of tiling workloads (see the second blog in this series for an overview of tiling) a fixed function tiling unit.

See https://community.arm.com/groups/arm-mali-graphics/blog/2014/03/12/the-mali-gpu-an-abstract-machine-part-3--the-shader-core.

" a processor unit flexibly controlled to perform vertex manipulation operations and pixel manipulation operations based on vertex or pixel workload; and "

To remove this idle time we use the OpenGL ES driver to maintain the illusion of synchronous rendering behavior, while actually processing rendering and frame swaps asynchronously under the hood. By running asynchronously we can build a small backlog of work, allowing a pipeline to be created where the GPU is processing older workloads from one end of the pipeline, while the CPU is busy pushing new work into the other. The advantage of this approach is that, provided we keep the pipeline full, there is always work available to run on the GPU giving the best performance.

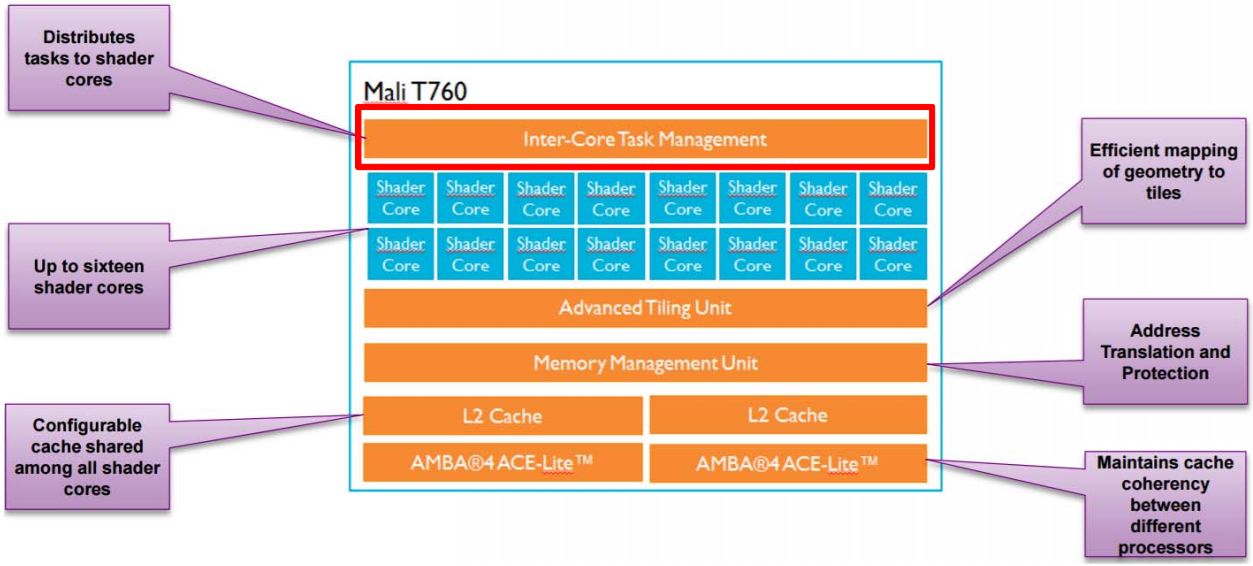


See https://community.arm.com/groups/arm-mali-graphics/blog/2014/02/03/the-mali-gpu-an-abstract-machine-part-1.

The shared hardware in Midgard is primarily concerned with managing the interaction of the shader cores, followed by providing the L2 cache and all further memory interfaces for accessing main memory and/or the CPU cache. In the case of Mali-T760 there is **1 task management unit** and memory management unit, but 2 sets of L2 cache and the AMBA interface that connects the GPU to the rest of the system.

See http://www.anandtech.com/show/8234/arms-mali-midgard-architecture-explored/4.

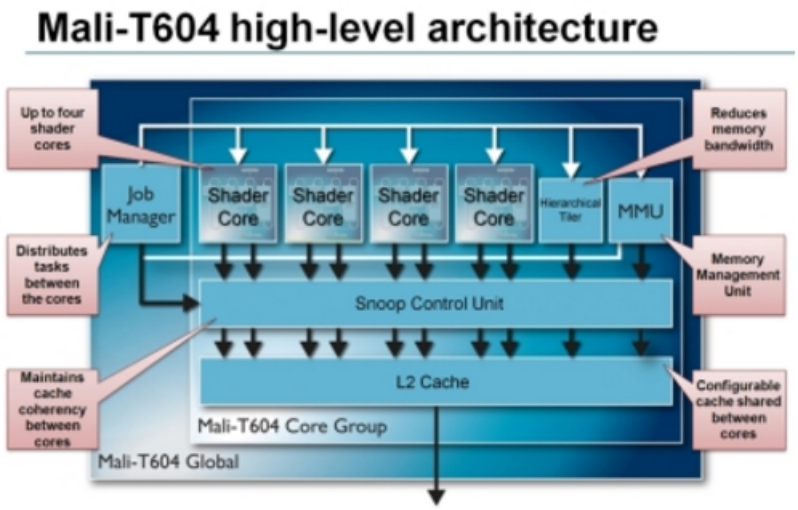
" a processor unit flexibly controlled to perform vertex manipulation operations and pixel manipulation operations based on vertex or pixel workload; and "



See http://malideveloper.arm.com/downloads/ARM_Game_Developer_Days/PDFs/2-Mali-GPU-architecture-overview-and-tile-local-storage.pdf.

" a processor unit flexibly controlled to perform vertex manipulation operations and pixel manipulation operations based on vertex or pixel workload; and "

Adding a bit more detail, the T604 includes four shader cores, each of which contains two arithmetic pipelines, one texturing pipeline, and one load/store unit. The four shaders share a coherent L2 cache, an MMU, a tiler, and a Job Manager. This latter block is a key component because the shaders are multithreaded. The Job Manager can dynamically move threads among the shaders.



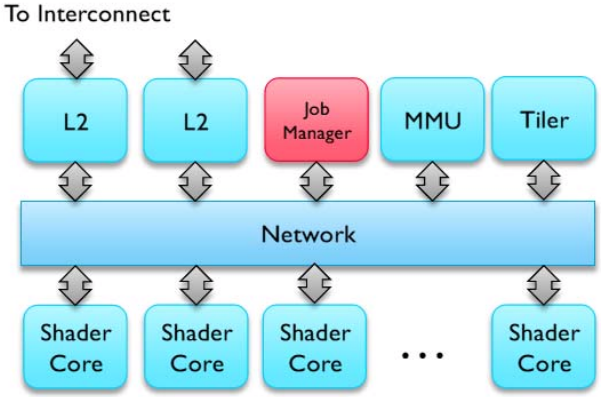
This dynamic load allocation, in turn, allows the host system to exert considerable control over the energy consumption of the core —vital to high-performance mobile use. But it also requires that the threads be light-weight. And that puts stress on the L2 cache, which must hold any state not local to the shader.

See http://www.eetimes.com/document.asp?doc_id=1278897.

" a processor unit flexibly controlled to perform vertex manipulation operations and pixel manipulation operations based on vertex or pixel workload; and "

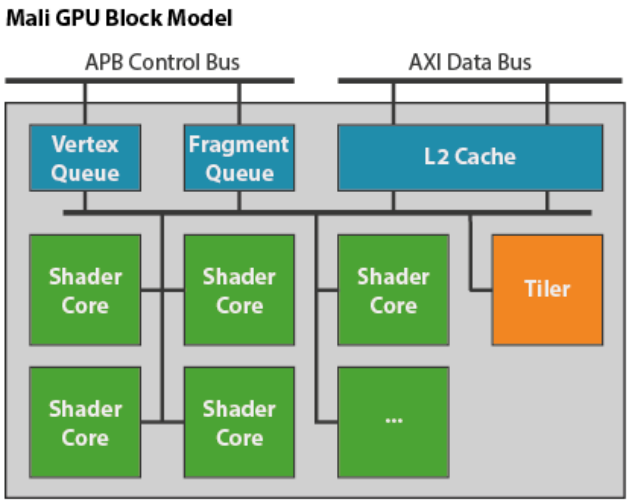
Mali-T880 GPU Block Diagram - Job Manager

- Fixed function block responsible for interfacing with the driver
  - Reads job descriptors from memory
  - Tracks inter-job dependencies
  - Distributes jobs across shader cores
  - Splits jobs into per-core tasks

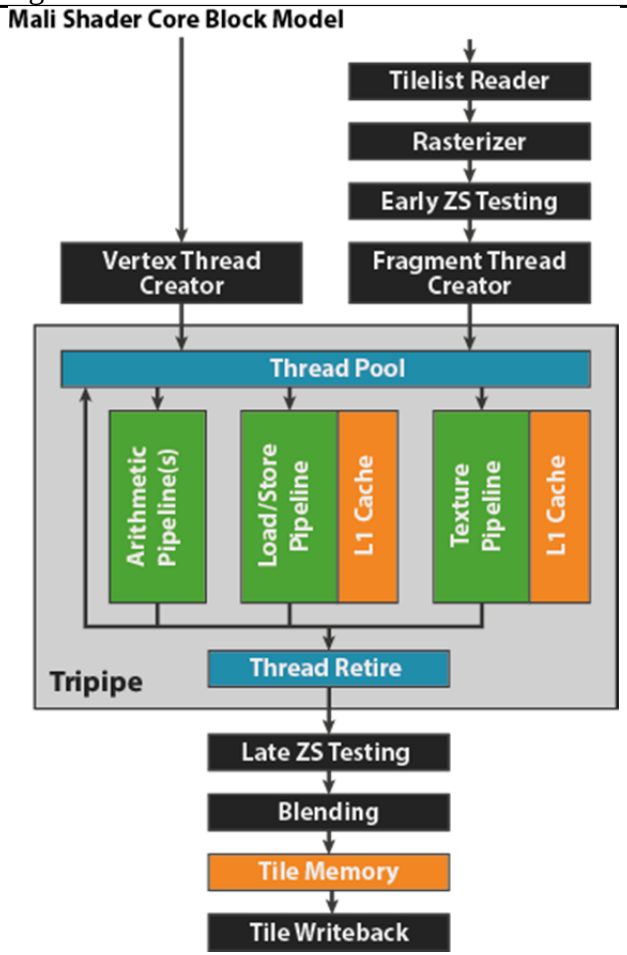


See ARM, The ARM Mali –T880 Mobile GPU, p.9, *available at* http://www.hotchips.org/wp-content/uploads/hc_archives/hc27/HC27.25-Tuesday-Epub/HC27.25.50-GPU-Epub/HC27.25.531-Mali-T880-Bratt-ARM-2015_08_23.pdf/.

"an instruction store and wherein the processor unit of the unified shader performs the vertex manipulation operations and pixel manipulation operations at various degrees of completion based on switching between instructions in the instruction store."

an instruction store and wherein the processor unit of the unified shader performs the vertex manipulation operations and pixel manipulation operations at various degrees of completion based on switching between instructions in the instruction store.	The LG Products include an instruction store and wherein the processor unit of the unified shader performs the vertex manipulation operations and pixel manipulation operations at various degrees of completion based on switching between instructions in the instruction store. The Mali GPUs include an instruction store. For example, the Mali GPU includes a Vertex Queue, Fragment Queue, Thread Pool, Load/Store Pipe, Caches, and registers.  *See* The Mali GPU: An Abstract Machine, Part 3 - The Midgard Shader Core, https://community.arm.com/groups/arm-mali-graphics/blog/2014/03/12/the-mali-gpu-an-abstract-machine-part-3--the-shader-core.

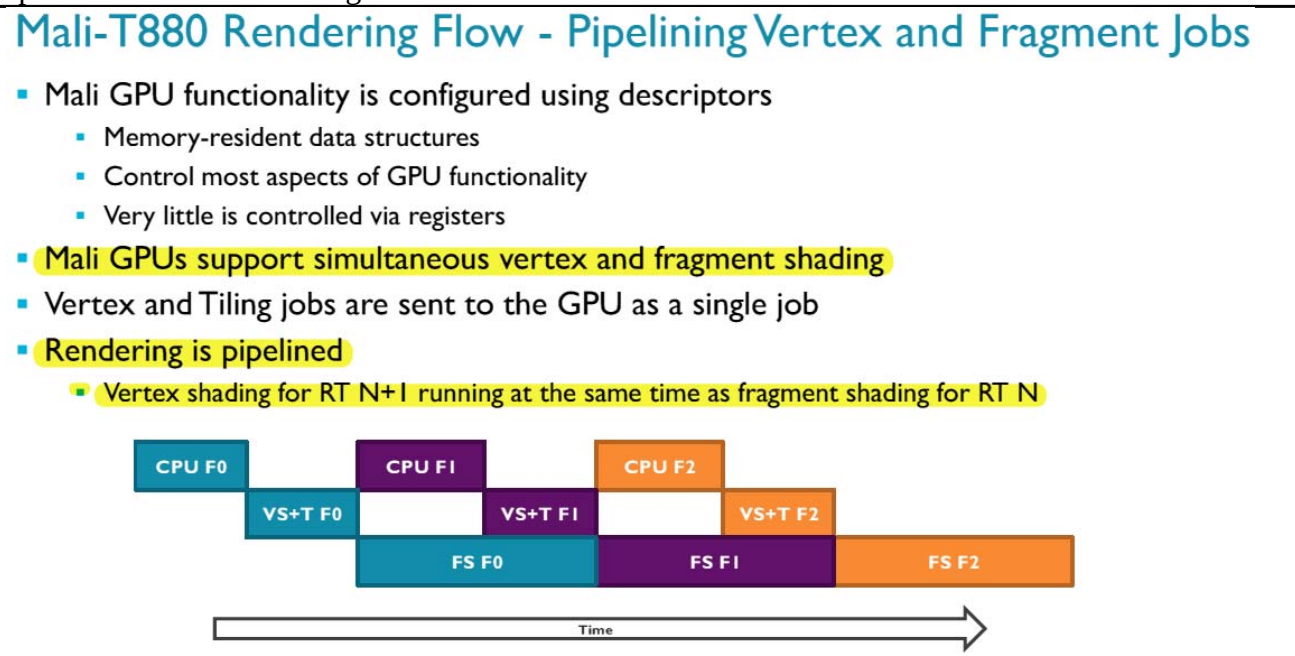
"an instruction store and wherein the processor unit of the unified shader performs the vertex manipulation operations and pixel manipulation operations at various degrees of completion based on switching between instructions in the instruction store."



See The Mali GPU: An Abstract Machine, Part 3 - The Midgard Shader Core, https://community.arm.com/groups/arm-mali-graphics/blog/2014/03/12/the-mali-gpu-an-abstract-machine-part-3--the-shader-core.

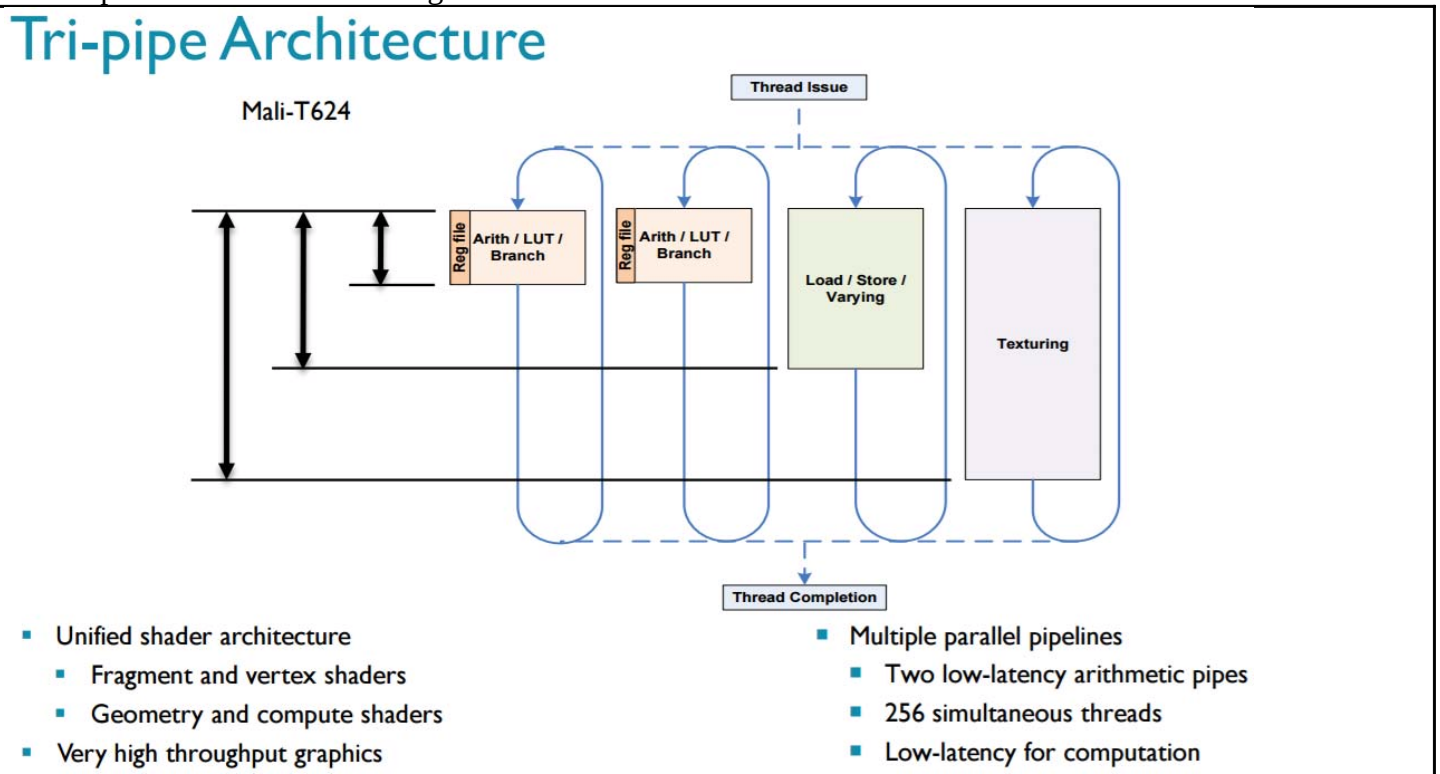
The processor unit of the unified shader performs vertex and pixel manipulation operations at various degrees of completion based on switching between instructions in the instruction store. For example, “Mali GPUs support simultaneous vertex and fragment shading[.]” Moreover “[v]ertex shading for RT N+1 run[s] at the same time as fragment shading for RT N[.]”

"an instruction store and wherein the processor unit of the unified shader performs the vertex manipulation operations and pixel manipulation operations at various degrees of completion based on switching between instructions in the instruction store."

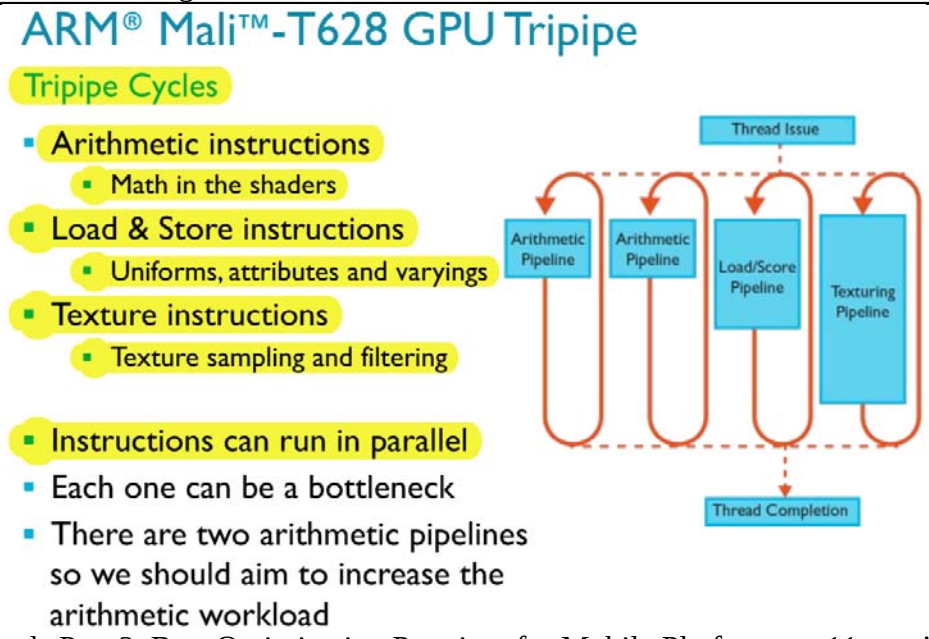


See ARM, The ARM Mali –T880 Mobile GPU, p.9, *available at* http://www.hotchips.org/wp-content/uploads/hc_archives/hc27/HC27.25-Tuesday-Epub/HC27.25.50-GPU-Epub/HC27.25.531-Mali-T880-Bratt-ARM-2015_08_23.pdf/.

"an instruction store and wherein the processor unit of the unified shader performs the vertex manipulation operations and pixel manipulation operations at various degrees of completion based on switching between instructions in the instruction store."

	 See http://malideveloper.arm.com/downloads/ARM_Game_Developer_Days/PDFs/2-Mali-GPU-architecture-overview-and-tile-local-storage.pdf.

"an instruction store and wherein the processor unit of the unified shader performs the vertex manipulation operations and pixel manipulation operations at various degrees of completion based on switching between instructions in the instruction store."



See ARM, ARM Tools Part 2, Best Optimization Practices for Mobile Platforms, p.11, *available at* http://malideveloper.arm.com/downloads/ARM_Game_Developer_Days/PDFs/6%20-%20ARM%20Tools%20Part%202-%20Best%20Optimization%20Practices%20for%20Mobile%20Platforms.pdf.

Massively Multi-threaded Machine

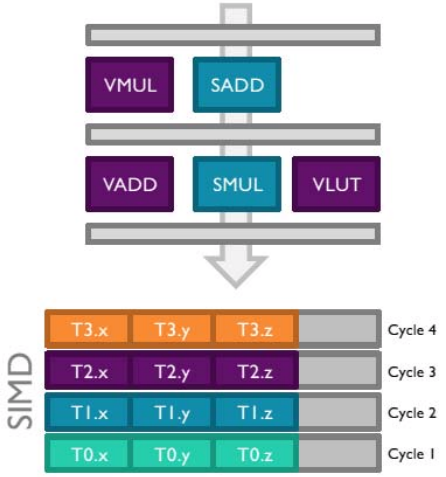
Unlike a traditional CPU architecture, where you will typically only have a single thread of execution at a time on a single core, the tripipe is a massively multi-threaded processing engine. There may well be hundreds of hardware threads running at the same time in the tripipe, with one thread created for each vertex or fragment which is shaded. This large number of threads exists to hide memory latency; it doesn't matter if some threads are stalled waiting for memory, as long as at least one thread is available to execute then we maintain efficient execution.

See https://community.arm.com/groups/arm-mali-graphics/blog/2014/03/12/the-mali-gpu-an-abstract-machine-part-3--the-shader-core.

"an instruction store and wherein the processor unit of the unified shader performs the vertex manipulation operations and pixel manipulation operations at various degrees of completion based on switching between instructions in the instruction store."

Mali-T880 GPU Shader Core - Arithmetic pipeline

- Arithmetic ISA on Midgard is SIMD + VLIW
  - Three vector units (128-bit datapath)
  - 4-lane FP32 or 8-lane FP16 for graphics
  - 16-lane int8 for compute
  - Two scalar units (32-bit datapath)
- One thread at a time executes in each pipeline stage
- Limited amount of out-of-order parallelism
  - Arith and Load/Store can progress under a pending Texture instruction



See ARM, The ARM Mali –T880 Mobile GPU, p.19, *available at* http://www.hotchips.org/wp-content/uploads/hc_archives/hc27/HC27.25-Tuesday-Epub/HC27.25.50-GPU-Epub/HC27.25.531-Mali-T880-Bratt-ARM-2015_08_23.pdf/.