Roadmap 2022 Milestone

A new baseline for immersive graphics

Tobias Hector, AMD
Software Architect
Roadmap 2022 Milestone

Vulkan Roadmap aims to reduce fragmentation
For mid- to high-end smartphone, tablet, laptop, console, and desktop devices

Aimed at immersive graphics experiences:
Gaming, interactive media, visualization, content authoring, etc.

Roadmap 2022 Milestone is the first step
Key Features:
- Descriptor Indexing
- Consistent subgroup support
- Multi-process scheduling
- Scalar Block Layout

Many other features raising the bar!

The Roadmap 2022 Milestone sets a new baseline for immersive graphics
Descriptor Indexing

Without Descriptor Indexing
Low max descriptors per set

Statically uniform indexing only - cannot generate dynamic indices in shader

Descriptors must be uploaded to sets before being recorded

New descriptors need new descriptor set allocations

Descriptors can require per-draw set allocations and significant host management
Descriptor Indexing enables simpler resource management both in the API and from shaders.

With Descriptor Indexing:
- High max descriptors per set - at least 500K for most resources
- Dynamic and non uniform resource indexing enables device driven rendering
- Stream descriptors in and out of the same sets while the set is in use
- No need to create new sets - allocate a handful of large sets at creation time
Subgroups

Hardware Runs Threads in Parallel
Groups of threads (subgroups) execute locally to a processing unit

Subgroup operations expose ability to coordinate these threads and communicate between them

Roadmap 2022 Baseline
Subgroups must be at least 4 invocations
Operations must be supported in fragment and compute shaders
Most operations must be available

Subgroup operations enable efficient local communication between threads
Required Subgroup Operations

**Basic**
Size query, Invocation election

**Vote**
Any, All
Reduce divergence

**Arithmetic**
Whole Subgroup
Results and Scans

**Ballot**
Finer control over divergence

**Shuffle**
Read from any other invocation

**Shuffle Relative**
Read from adjacent invocations

**Quad**
Matches derivatives in fragment shaders
Global Priority

Multiple Processes Accessing the GPU

Vulkan 1.0 included a basic priority mechanism that enabled setting per-queue priorities within a process.

Some use cases (notably composition) need to be able to identify elevated privilege across processes, to ensure screen updates hit your eye on time.

This is particularly important for XR use cases, where latency tolerance is incredibly tight.

VK_KHR_global_queue_priority provides the ability for an application to request different priorities for different queue families.

Submissions on high priority queues will be preferentially scheduled when ready.

Global queue priorities provide cross-process prioritisation.
Scalar Block Layout

Simpler Host-Device Variable Access
Scalar block layout allows alignment of variables to the size of their scalar elements for buffer memory

So a vector of four floats can be aligned to 4-byte boundaries (the size of a floating point value)

This allows the layout of data in host-side structures to map 1:1 in the majority of cases without needing to add host-side padding

Scalar block layout data layout capabilities to parity with host programming languages
Other Required Features

- 8k images
- Fragment shader stores and atomics
- $Y'CBR$ image sampling
- Anisotropic filtering
- Precise occlusion queries
- Inf/NaN preservation for 32-/16-bit float
- Depth clamp
- Depth bias clamp
- Sample rate shading
- Mirror once then clamp UV wrapping
- Per-attachment blending
- Cubemap arrays
- 32-bit draw indexing
- Draw instance offset
Raising the Baseline for Immersive Graphics

Enhanced functionality for immersive graphics
Roadmap 2022 sets a new baseline for applications targeting an immersive graphics experience

See the Vulkan 1.3 specification for the full list of additional features, limits, and extensions

Explicitly targets mid- and high-end devices across smartphone, tablet, laptop, console, and desktop

Enables newer features to be required, improving the developer experience, allowing final products to be the best they can be