

# Vulkanised!

Vulkan 1.3 Overview

Ralph Potter, Samsung



**KRONOS**  
GROUP

WEBINARS  
& MEETUPS

**Vulkan**<sup>®</sup>

# Vulkan 1.3

**Supportable on OpenGL ES 3.1-class hardware**

Same hardware baseline as Vulkan 1.0

**Integrates significant requested functionality proven as extensions**

23 extensions promoted to the Vulkan 1.3 Core API

Mandated support for new functionality ensures availability on all Vulkan 1.3 implementations

**Hardware vs. Software (mostly)**

API usability and infrastructure changes in Vulkan 1.3 Core API

(programming model changes, API interfaces, compiler etc.)

New hardware functionality in extensions and Roadmap

# Vulkan 1.3

VK\_EXT\_4444\_formats

VK\_EXT\_extended\_dynamic\_state

VK\_EXT\_extended\_dynamic\_state2

VK\_EXT\_image\_robustness

VK\_EXT\_inline\_uniform\_block

VK\_EXT\_pipeline\_creation\_cache\_control

VK\_EXT\_pipeline\_creation\_feedback

VK\_EXT\_private\_data

VK\_EXT\_shader\_demote\_to\_helper\_invocation

VK\_EXT\_subgroup\_size\_control

VK\_EXT\_texel\_buffer\_alignment

VK\_EXT\_texture\_compression\_astc\_hdr

VK\_EXT\_tooling\_info

VK\_EXT\_ycbcr\_2plane\_444\_formats

VK\_KHR\_copy\_commands2

VK\_KHR\_dynamic\_rendering

VK\_KHR\_format\_feature\_flags2

VK\_KHR\_maintenance4

VK\_KHR\_shader\_integer\_dot\_product

VK\_KHR\_shader\_non\_semantic\_info

VK\_KHR\_shader\_terminate\_invocation

VK\_KHR\_synchronization2

VK\_KHR\_zero\_initialize\_workgroup\_memory

# Dynamic Rendering

- Render passes have been a recurring area of developer feedback
- API improvements for single pass render passes
- Removes coupling to `VkRenderPass` and `VkFramebuffer` objects
- Single pass performance is equivalent to the existing API

```
VKAPI_ATTR void VKAPI_CALL vkCmdBeginRenderingKHR(  
    VkCommandBuffer          commandBuffer,  
    const VkRenderingInfoKHR* pRenderingInfo);  
  
typedef struct VkRenderingInfoKHR {  
    VkStructureType          sType;  
    const void*              pNext;  
    VkRenderingFlagsKHR      flags;  
    VkRect2D                 renderArea;  
    uint32_t                 layerCount;  
    uint32_t                 viewMask;  
    uint32_t                 colorAttachmentCount;  
    const VkRenderingAttachmentInfoKHR* pColorAttachments;  
    const VkRenderingAttachmentInfoKHR* pDepthAttachment;  
    const VkRenderingAttachmentInfoKHR* pStencilAttachment;  
} VkRenderingInfoKHR;
```

Streamlining Render Passes Blog: <https://www.khronos.org/blog/streamlining-render-passes>

Design Document: [https://github.com/KhronosGroup/Vulkan-Docs/blob/main/proposals/VK\\_KHR\\_dynamic\\_rendering.asciidoc](https://github.com/KhronosGroup/Vulkan-Docs/blob/main/proposals/VK_KHR_dynamic_rendering.asciidoc)

# Pipeline Creation

- **VK\_EXT\_pipeline\_creation\_cache\_control**
  - Gives developers more control over how and when pipelines are compiled
- **VK\_EXT\_pipeline\_creation\_feedback**
  - Provides information about the compiled pipelines, making it easier to profile and debug pipeline issues

## Extension Specifications:

[https://www.khronos.org/registry/vulkan/specs/1.3-extensions/man/html/VK\\_EXT\\_pipeline\\_creation\\_cache\\_control.html](https://www.khronos.org/registry/vulkan/specs/1.3-extensions/man/html/VK_EXT_pipeline_creation_cache_control.html)

[https://www.khronos.org/registry/vulkan/specs/1.3-extensions/man/html/VK\\_EXT\\_pipeline\\_creation\\_feedback.html](https://www.khronos.org/registry/vulkan/specs/1.3-extensions/man/html/VK_EXT_pipeline_creation_feedback.html)

# Extended Dynamic State

- Static state changes require VkPipeline objects
- Promoted VK\_EXT\_extended\_dynamic\_state and VK\_EXT\_extended\_dynamic\_state2
- Making state dynamic reduces the number of VkPipeline objects
- Dynamic state for patch control points and logic op are not promoted in Vulkan 1.3

Extension Specifications:

[https://www.khronos.org/registry/vulkan/specs/1.3-extensions/man/html/VK\\_EXT\\_extended\\_dynamic\\_state.html](https://www.khronos.org/registry/vulkan/specs/1.3-extensions/man/html/VK_EXT_extended_dynamic_state.html)

[https://www.khronos.org/registry/vulkan/specs/1.3-extensions/man/html/VK\\_EXT\\_extended\\_dynamic\\_state2.html](https://www.khronos.org/registry/vulkan/specs/1.3-extensions/man/html/VK_EXT_extended_dynamic_state2.html)

# Image Formats

- **VK\_EXT\_texture\_compression\_astc\_hdr**
  - VK\_FORMAT\_ASTC\_4x4\_SFLOAT\_BLOCK
  - ...
- **VK\_EXT\_4444\_formats**
  - VK\_FORMAT\_A4R4G4B4\_UNORM\_PACK16
  - VK\_FORMAT\_A4B4G4R4\_UNORM\_PACK16
- **VK\_EXT\_ycbcr\_2plane\_444\_formats**
  - VK\_FORMAT\_G10X6\_B10X6R10X6\_2PLANE\_444\_UNORM\_3PACK16
  - VK\_FORMAT\_G12X4\_B12X4R12X4\_2PLANE\_444\_UNORM\_3PACK16
  - VK\_FORMAT\_G16\_B16R16\_2PLANE\_444\_UNORM
  - VK\_FORMAT\_G8\_B8R8\_2PLANE\_444\_UNORM
- **The formats added by these extensions are optional in core**
- **But implementations will now recognize the enums and give valid results to queries**

# Shader Extensions

- **VK\_KHR\_shader\_integer\_dot\_product**
  - Particularly useful for ML applications, but available for general usage
- **VK\_KHR\_shader\_non\_semantic\_info**
  - Adds support for SPIR-V instructions that are ignored by the driver, but may aid tools such as debuggers
- **VK\_KHR\_shader\_terminate\_invocation**
  - Replace SPIR-V OpKill with better specified behaviour
  - This is a better match for the “discard” instruction in GLSL
- **VK\_EXT\_shader\_demote\_to\_helper\_invocation**
  - This is a better match for the “discard” instruction in HLSL



# Usability, Infrastructure and Maintenance

- **VK\_KHR\_synchronization2**
  - Usability improvements for synchronization
- **VK\_EXT\_image\_robustness**
  - Defined behaviour for out-of-bounds operations
- **VK\_EXT\_inline\_uniform\_block**
  - Store uniforms in descriptor set storage
- **VK\_EXT\_private\_data, VK\_EXT\_tooling\_info**
  - Improvements to layer and tooling support
- **VK\_EXT\_subgroup\_size\_control**
  - Greater control/introspection of subgroup behaviour
- **VK\_EXT\_texel\_buffer\_alignment**
  - Alignment queries for uniform and storage texel buffers
- **VK\_KHR\_copy\_commands2, VK\_KHR\_format\_feature\_flags2**
  - Infrastructure extensions to supporting future API extension
- **VK\_KHR\_maintenance4**
  - Bugfixes and usability improvements
- **VK\_KHR\_zero\_initialize\_workgroup\_memory**
  - Zero-initialized memory for security conscious use cases

# Other Changes

- **Support for SPIR-V version 1.6**
- **Support for the following features is now mandatory:**
  - `bufferDeviceAddress`
  - `vulkanMemoryModel`
  - `vulkanMemoryModelDeviceScope`
  - `vulkanMemoryModelAvailabilityVisibilityChains`
- **`maxInlineUniformTotalSize` limit is added to provide the total size of all inline uniform block bindings in a pipeline layout**

# More Information

Vulkan: <https://www.vulkan.org/>

- Press Release: <https://khr.io/xj>
- Final specification: <https://khr.io/vulkan13spec>
- Spec GitHub Repo: <https://khr.io/vulkan13github>
- Discord Link for community discussion: <https://khr.io/vulkan13discord>
- Vulkan 1.3 Blog post: <https://www.khronos.org/blog/vulkan-1.3-and-roadmap-2022>
- Google Android Baseline 2021 Profile blog:  
<https://www.khronos.org/blog/vulkan-1.3-and-roadmap-2022>

**Another Vulkanised Webinar coming after SDK release**  
Will cover SDK features in detail

