

Radeon™ GPU Analyzer 2.4.2 – Release Notes

Highlights

- Added support for gfx1031 (RX 6700 XT) in Vulkan, DX12, DXR, DX11 and OpenGL modes.
- DXR:
 - o Improved error reporting to help diagnose State Object creation failures.
 - o You can now extract pipeline binaries even in cases where the driver generated multiple raytracing pipelines.
- Improved static analysis engine's stability.
- The following RDNA and RDNA2 instructions are now supported by the static analysis engine:
 - o V_DOT2C_F32_F16
 - o V_DOT4C_I32_I8
 - o V_DOT2_F32_F16
 - o V_DOT2_I32_I16
 - o V_DOT2_U32_U16
 - o V_DOT4_I32_I8
 - o V_DOT4_U32_U8
 - o V_DOT8_I32_I4
 - o V_DOT8_U32_U4
 - o V_FMAC_LEGACY_F32
 - o V_FMA_LEGACY_F32
- Vulkan offline mode: added support for new SPIR-V extensions and capabilities. See the accompanying Release Notes document for the complete list of supported extensions and capabilities.
- Notable bug fixes:
 - o DXR: Fixes to the command line tool's DXR help manual.
 - o DXR: Auto-generated file name extension is now aligned with DX12 mode.
 - o Fixed the compute capability for certain APU products (gfx909 -> gfx90C).

Known Issues

DXR Mode

- Disassembly for Wave32 shaders may show references of shared VGPRs.
- In scenarios where multiple raytracing pipelines are created from the same State Object, some binaries may not be of valid ELF format. A fix will be provided in an upcoming Adrenalin software release (users do not need to update the tool to consume the fix).

Vulkan Live Driver Mode

- Source to disassembly correlation is not supported by AMD's shader compiler at the moment and is therefore not supported in the UI.
- Keyboard navigation is not supported from some views in the GUI application.
- The RGA layer is a beta feature. It fails to extract the shaders and pipeline state from certain Vulkan apps.
- Notifications about the fact that modified SPIR-V binary does not match the disassembly will not appear for loaded projects (in case that you changed the SPIR-V code, did not build and re-loaded the project).

ROCM OpenCL Mode

- RDNA targets are not supported as targets for ROCM-CL modes.
- The Lightning Compiler does not support disassembling of binaries for Graphics IP v7 targets.
- OpenCL C++ kernels are not yet supported by the Lightning Compiler.

- No support for live register analysis and control-flow graph generation in this mode.
- Cycle estimate for certain VALU instructions appears as “Varies” instead of 4.

OpenGL Mode

Resource usage statistics for OpenGL mode only displays usage of SGPRs and VGPRs.

DirectX 12 Mode

Live register analysis & CFG generation require using the --isa option to generate ISA disassembly.

Vulkan Offline Modes (vk-offline, vk-spv-offline, vk-spv-txt-offline)

SPIR-V support limitations:

- The following capabilities, which are defined in the SPIR-V spec, are currently not supported:

- CapabilityMatrix
- CapabilityShader
- CapabilityGeometry
- CapabilityTessellation
- CapabilityFloat16
- CapabilityFloat64
- CapabilityInt64
- CapabilityInt64Atomics
- CapabilityGroups
- CapabilityAtomicStorage
- CapabilityInt16
- CapabilityTessellationPointSize
- CapabilityGeometryPointSize
- CapabilityImageGatherExtended
- CapabilityStorageImageMultisample
- CapabilityUniformBufferArrayDynamicIndexing
- CapabilitySampledImageArrayDynamicIndexing
- CapabilityStorageBufferArrayDynamicIndexing
- CapabilityStorageImageArrayDynamicIndexing
- CapabilityClipDistance
- CapabilityCullDistance
- CapabilityImageCubeArray
- CapabilitySampleRateShading
- CapabilityImageRect
- CapabilitySampledRect
- CapabilityInt8
- CapabilityInputAttachment
- CapabilitySparseResidency
- CapabilityMinLod
- CapabilitySampled1D
- CapabilityImage1D
- CapabilitySampledCubeArray
- CapabilitySampledBuffer
- CapabilityImageBuffer
- CapabilityImageMSArray

CapabilityStorageImageExtendedFormats
CapabilityImageQuery
CapabilityDerivativeControl
CapabilityInterpolationFunction
CapabilityTransformFeedback
CapabilityGeometryStreams
CapabilityStorageImageReadWithoutFormat
CapabilityStorageImageWriteWithoutFormat
CapabilityMultiViewport
CapabilitySubgroupDispatch
CapabilityNamedBarrier
CapabilityPipeStorage
CapabilityGroupNonUniform
CapabilityGroupNonUniformVote
CapabilityGroupNonUniformArithmetic
CapabilityGroupNonUniformBallot
CapabilityGroupNonUniformShuffle
CapabilityGroupNonUniformShuffleRelative
CapabilityGroupNonUniformClustered
CapabilityGroupNonUniformQuad
CapabilitySubgroupBallotKHR
CapabilityDrawParameters
CapabilitySubgroupVoteKHR
CapabilityStorageBuffer16BitAccess
CapabilityStorageUniformBufferBlock16
CapabilityStorageUniform16
CapabilityUniformAndStorageBuffer16BitAccess
CapabilityStorageInputOutput16
CapabilityDeviceGroup
CapabilityMultiView
CapabilityVariablePointersStorageBuffer
CapabilityVariablePointers
CapabilitySampleMaskPostDepthCoverage
CapabilityStorageBuffer8BitAccess
CapabilityUniformAndStorageBuffer8BitAccess
CapabilityDenormPreserve
CapabilityDenormFlushToZero
CapabilitySignedZeroInfNanPreserve
CapabilityRoundingModeRTE
CapabilityRoundingModeRTZ
CapabilityFloat16ImageAMD
CapabilityImageGatherBiasLodAMD
CapabilityFragmentMaskAMD
CapabilityStencilExportEXT
CapabilityImageReadWriteLodAMD
CapabilityInt64ImageEXT
CapabilityShaderClockKHR
CapabilityShaderViewportIndexLayerEXT
CapabilityFragmentShadingRateKHR

CapabilityFragmentDensityEXT
CapabilityShaderNonUniformEXT
CapabilityRuntimeDescriptorArrayEXT
CapabilityInputAttachmentArrayDynamicIndexingEXT
CapabilityUniformTexelBufferArrayDynamicIndexingEXT
CapabilityStorageTexelBufferArrayDynamicIndexingEXT
CapabilityUniformBufferArrayNonUniformIndexingEXT
CapabilitySampledImageArrayNonUniformIndexingEXT
CapabilityStorageBufferArrayNonUniformIndexingEXT
CapabilityStorageImageArrayNonUniformIndexingEXT
CapabilityUniformTexelBufferArrayNonUniformIndexingEXT
CapabilityStorageTexelBufferArrayNonUniformIndexingEXT
CapabilityVulkanMemoryModel
CapabilityVulkanMemoryModelKHR
CapabilityVulkanMemoryModelDeviceScope
CapabilityVulkanMemoryModelDeviceScopeKHR
CapabilityPhysicalStorageBufferAddresses
CapabilityPhysicalStorageBufferAddressesEXT
CapabilityDemoteToHelperInvocationEXT
CapabilityRayTracingProvisionalKHR
CapabilityRayQueryProvisionalKHR
CapabilityRayTraversalPrimitiveCullingProvisionalKHR
SPIR-V modes currently only support the following extensions:
SPV_KHR_shader_ballot
SPV_KHR_subgroup_vote
SPV_KHR_device_group
SPV_KHR_multiview
SPV_KHR_shader_draw_parameters
SPV_KHR_16bit_storage
SPV_KHR_storage_buffer_storage_class
SPV_KHR_8bit_storage
SPV_KHR_variable_pointers
SPV_KHR_float_controls
SPV_KHR_shader_clock
SPV_KHR_vulkan_memory_model
SPV_KHR_post_depth_coverage
SPV_KHR_non_semantic_info
SPV_KHR_physical_storage_buffer
SPV_KHR_terminate_invocation
SPV_KHR_FRAGMENT_SHADING_RATE
SPV_EXT_nonuniform_qualifier
SPV_EXT_shader_stencil_export
SPV_EXT_shader_viewport_index_layer
SPV_EXT_demote_to_helper_invocation
SPV_EXT_shader_image_atomic_int64
SPV_AMD_shader_ballot
SPV_AMD_shader_trinary_minmax
SPV_AMD_shader_explicit_vertex_parameter
SPV_AMD_gcn_shader

SPV_AMD_gpu_shader_half_float
SPV_AMD_texture_gather_bias_lod
SPV_AMD_gpu_shader_int16
SPV_AMD_shader_fragment_mask
SPV_AMD_shader_image_load_store_lod
SPV_AMD_shader_texel_buffer_explicit_format
SPV_AMD_property_id_attachment
SPV_AMD_anisotropic_lod_compensation
SPV_ARB_shader_ballot
SPV_GOOGLE_decorate_string
SPV_GOOGLE_hlsl_functionality1
SPV_GOOGLE_user_type
SPV_KHR_ray_tracing
SPV_KHR_ray_query

GUI Application

- “Correlation Disabled” notification in the source code editor is not being saved for projects after they were closed.
- Certain SALU instructions are being misclassified as VALU instructions.

Notes for OpenCL Mode Users

The “-s rocm-cl” mode uses the Lightning Compiler package that ships with RGA, which is based on clang.

As of version 2.0, RGA allows developers to replace the Lightning Compiler package that ships with the product with a user-provided LLVM-based package. For more information, see the Radeon GPU Analyzer GUI app’s help manual, or run the command line tool with `-s rocm-cl -h` as arguments (look for the “Alternative ROCm OpenCL compiler” section).

Notes for DirectX 11 Mode Users

RGA’s DirectX 11 (-s dx11) mode will use the driver that is associated with the primary display adapter, by default. If your primary display adapter is not an AMD GPU, or if you would like RGA to use a driver that is associated with a different display adapter that is installed on your system, use the `--adapters` and `--set-adapter <id>` command line switches in order to instruct RGA to use the relevant driver.

System Requirements

It is generally recommended to use RGA with the latest Radeon Software version. Specifically, to target the RDNA architecture, the latest Radeon Software version is required (except for all Vulkan® modes and the rocm-cl mode, which are independent of the driver).

Vulkan Mode

To use the installed driver in Vulkan mode:

- a. Vulkan SDK 1.1.97.0 or later is required.
- b. Latest Adrenalin or amdgpu-pro driver is required.

RGA ships with a compatible Vulkan driver to support users who do not have an AMD driver installed on their system. In cases where RGA fails to detect the installed driver, it falls back to using the bundled driver. To make

sure that you are using the latest compiler and can compile for the latest Radeon targets, we strongly recommend running on a machine that has an AMD driver installed.

[Vulkan Offline Modes \(vk-offline, vk-spv-offline, vk-spv-txt-offline\)](#)

All Vulkan offline modes (vk-offline, vk-spv-offline and vk-spv-txt-offline) are independent of the installed driver and graphics hardware and should work on any x86-based system.

[DirectX 12 and DirectX 11 Modes](#)

It is recommended to use the latest Adrenalin drivers for the best experience in DirectX 12 and DirectX 11 modes.

[ROCM OpenCL Mode](#)

ROCM OpenCL mode (rocm-cl) is independent of the installed driver and graphics hardware and should work on any x86-based system.

[OpenGL Mode](#)

OpenGL mode on Linux requires the latest amdgpu-pro driver.