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:
  + Improved error reporting to help diagnose State Object creation failures.
  + 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:
  + V\_DOT2C\_F32\_F16
  + V\_DOT4C\_I32\_I8
  + V\_DOT2\_F32\_F16
  + V\_DOT2\_I32\_I16
  + V\_DOT2\_U32\_U16
  + V\_DOT4\_I32\_I8
  + V\_DOT4\_U32\_U8
  + V\_DOT8\_I32\_I4
  + V\_DOT8\_U32\_U4
  + V\_FMAC\_LEGACY\_F32
  + 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:
  + DXR: Fixes to the command line tool’s DXR help manual.
  + DXR: Auto-generated file name extension is now aligned with DX12 mode.
  + 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.

## DirectX12 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:

1. 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:

1. Vulkan SDK 1.1.97.0 or later is required.
2. 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.