Radeon™ GPU Analyzer 2.6.2 – Release Notes

# Highlights

* **OpenGL mode is now an offline mode (-s opengl):**
  + Compile OpenGL shaders for any supported target, independently of the installed graphics card and driver.
  + You can now generate static hardware resource usage information support for OpenGL shaders.
* Support for legacy targets (GFX8 and older) was removed

**Note**: on Ubuntu, the minimum required OS version is Ubuntu 20.04.

# Known Issues

## Vulkan Live Driver Mode

* Source to disassembly correlation is not supported by AMD’s shader compiler 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).

## Offline OpenCL Mode

* OpenCL C++ kernels are not yet supported by the Lightning Compiler.
* Cycle estimate for certain VALU instructions appears as “Varies” instead of 4.

## DirectX12 Mode

* Live register analysis & CFG generation require using the --isa option to generate ISA disassembly.
* On Windows 11, when running the same RGA command more than once with the --il option in the command, the AMDIL disassembly for vertex shaders shows up corrupted.
* On Windows 11, Hull shader AMDIL disassembly contains a trail of a few garbage characters.

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

SPIR-V support limitations:

1. The Vulkan Offline modes currently only support the following SPIR-V capabilities:

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

CapabilityAtomicFloat32MinMaxEXT

CapabilityAtomicFloat64MinMaxEXT

1. The Vulkan Offline 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

## GUI Application

* VGPR Pressure feature: certain instructions (image\_\* in particular) may report more live registers than actually used.
* “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.
* Certain GDS instructions are being misclassified as SALU.
* Changing disassembly columns can be sluggish on certain systems in projects with multiple .cl files.
* On systems with default Unicode path for the RGA projects folder, the tool will not be able to save files. A workaround has been provided in version 2.6.1 (allowing the user to change the folder in which projects are saved).
* Cannot rename source files whose file name contains a whitespace character.

# Notes for OpenCL Mode Users

The Offline OpenCL 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 opencl –h as arguments (look for the “Alternative OpenCL Lightning 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 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 the Offline Vulkan, OpenGL and OpenCL modes, 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.

## 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.

## OpenGL Offline Mode (opengl)

The new OpenGL mode (introduced in RGA 2.6.2) is 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.

## Offline OpenCL Mode

Offline OpenCL mode (-s opencl) is independent of the installed driver and graphics hardware and should work on any x86-based system.