Radeon™ GPU Analyzer 2.12 – Release Notes

# Highlights

* Added support for RDNA™ 4 (gfx1201) as well as gfx1151 and gfx1152 architectures as compiler targets across all RGA's modes.
* Upgraded ISA disassembly view: you can now navigate through the disassembly more easily and identify areas with higher VGPR pressure in your shaders and kernels.
* The new disassembly view presents tooltips when hovering over opcodes in the view.
* Updated the offline OpenCL, Vulkan and OpenGL compilers.

**Notes:**

* With the upgraded ISA disassembly view clock cycle estimates are no longer presented.
* Clock cycle estimates are no longer presented by the upgraded ISA disassembly view in RGA GUI application.
* The new disassembly view presents Functional Groups, leveraging the AMD GPU machine-readable ISA documentation.
* For backward compatibility, the RGA CLI output presents Cycle estimates and Functional Unit fields similar to RGA 2.11.
* We are working on a revised performance estimate mechanism and will share more details in a future RGA release.

# 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 where the user 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.

## DirectX12 Mode

* Live register analysis & CFG generation require using the --isa option to generate ISA disassembly.
* On Windows 11, Hull shader AMDIL disassembly contains a trail of a few garbage characters.
* In offline mode, when the D3D debug layer is enabled (--debug-layer) debug output is not captured by the command line tool.

## DirectX11 Mode

* Support for gfx1100 and beyond is not yet enabled in DX11 mode (-s dx11).

## 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

CapabilityMeshShadingEXT

CapabilityFragmentShaderSampleInterlockEXT

CapabilityFragmentFullyCoveredEXT

CapabilityFragmentShaderShadingRateInterlockEXT

CapabilityInputAttachmentArrayNonUniformIndexingEXT

CapabilityAtomicStorageOps

CapabilityFragmentShaderPixelInterlockEXT

CapabilityDotProductKHR

CapabilityDotProductInputAllKHR

CapabilityDotProductInput4x8BitKHR

CapabilityDotProductInput4x8BitPackedKHR

CapabilityWorkgroupMemoryExplicitLayoutKHR

CapabilityWorkgroupMemoryExplicitLayout8BitAccessKHR

CapabilityWorkgroupMemoryExplicitLayout16BitAccessKHR

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

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

CapabilityRayTracingNV

CapabilityRayQueryKHR

CapabilityRayTracingKHR

CapabilityRayCullMaskKHR

CapabilityRayTracingProvisionalKHR

CapabilityRayQueryProvisionalKHR

CapabilityRayTraversalPrimitiveCullingProvisionalKHR

CapabilityAbortLongRaysAMD

CapabilityRayTracingPositionFetchKHR

CapabilityRayQueryPositionFetchKHR

CapabilityShaderEnqueueAMDX

CapabilityDotProductInput8x4BitPackedAMD

CapabilityCooperativeMatrixKHR

SparseCooperativeMatrixAMD

CapabilityComputeDerivativeGroupLinearNV

CapabilityComputeDerivativeGroupQuadsNV

CapabilityExpectAssumeKHR

CapabilityQuadControl

CapabilityGroupNonUniformRotateKHR

CapabilityUntypedPointersKHR

CapabilityUntypedPointersWorkgroupKHR

CapabilityUntypedPointersStorageBufferKHR

CapabilityUntypedPointersUniformKHR

CapabilityUntypedPointersPushConstantKHR

CapabilityUntypedPointersAllStorageClassesKHR

CapabilityFMAKHR

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\_KHR\_workgroup\_memory\_explicit\_layout

## SPV\_KHR\_fragment\_shader\_barycentric

## 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\_EXT\_mesh\_shader

## 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\_gpu\_shader\_half\_float\_fetch

## SPV\_AMD\_shader\_early\_and\_late\_fragment\_tests

## SPV\_ARB\_shader\_ballot

## SPV\_GOOGLE\_decorate\_string

## SPV\_GOOGLE\_hlsl\_functionality1

## SPV\_GOOGLE\_user\_type

## SPV\_AMD\_abort\_long\_rays

## SPV\_AMDX\_shader\_enqueue

## SPV\_KHR\_ray\_tracing\_position\_fetch

## SPV\_KHR\_ray\_tracing

## SPV\_KHR\_ray\_query

## SPV\_AMD\_sparse\_cooperative\_matrix

## SPV\_AMD\_int4\_dot\_product

## SPV\_NV\_shader\_atomic\_float

## SPV\_NV\_compute\_shader\_derivatives

## SPV\_KHR\_maximal\_reconvergence

## SPV\_KHR\_expect\_assume

## SPV\_KHR\_shader\_quad\_control

## SPV\_KHR\_subgroup\_rotate

## SPV\_KHR\_untyped\_pointers

## SPV\_KHR\_compute\_shader\_derivatives

## SPV\_KHR\_fma

## 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.
* 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).
* On Linux, if the OS color theme is updated while the RGA GUI application is running, the RGA GUI application will not immediately reflect these changes (the changes will be reflected the next time that you launch the application).
* Tooltips and Functional Groups are not supported for pre-RDNA targets.

## General

# 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).

# 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, OpenCL mode, OpenGL mode and DX11 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.

## 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

* It is recommended to use the latest Adrenalin drivers for the best experience in DirectX 12 and DXR modes.
* Please note that to enable the D3D12 debug layer (--debug-layer option), your Windows system must have the Graphics Tools feature installed (on Windows, go to Settings -> Apps -> Apps & features/Optional features). This is a D3D12 requirement which is unrelated to RGA.

## DirectX 11

DirectX 11 offline mode (-s dx11) is independent of the installed driver and graphics hardware and should work on any x86-based system.

## 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.

## OpenGL Mode

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