Radeon™ GPU Analyzer 2.4 – Release Notes

# Highlights

* RGA is now available as part of the **Radeon Developer Tool Suite** on [GPUOpen](https://gpuopen.com/tools/), together with RGP, RMV and RDP. For the best experience, we encourage you to use the suite to ensure compatibility between the different tools. On GitHub, we will continue the individual RGA source code and binary releases as usual.
* Added support for **gfx1030 targets (RX 5800, RX 5800 XT and RX 5900 XT)** in DirectX® 12, DirectX® 11, Vulkan®, Vulkan-Offline, OpenGL® and OpenCL™ modes.
* Added support for **gfx1011** and **gfx909** targets in DirectX 12, DirectX 11, Vulkan, Vulkan-offline and OpenGL modes.
* **DirectX 12 mode**:
  + You can now enable the D3D® 12 debug layer to troubleshoot pipeline creation failures by adding the **--debug-layer** option to your command line.
  + For RDNA™ and Vega targets, you can now generate disassembly for the entire pipeline elf binary using the **--elf-dis** option.
  + Updated the **DXC** package that is bundled with the tool.
* **Vulkan mode**:
  + The tool can now digest disassembly from AMD’s LLVM-based pipeline compiler (**LLPC**).
  + RGA will now automatically pick the amdgpu-pro driver on machines where RADV installed.
* **Vulkan-offline mode**: RGA now accepts Vulkan driver pipeline dumps as an input (**.pipe files**).
* **GUI application** now remembers the window size for subsequent runs after closing.
* **Notes**:
  + The GUI application was renamed from “RadeonGPUAnalyzerGUI” to “RadeonGPUAnalyzer”.
  + Legacy OpenCL mode (-s cl) is discontinued on Linux, with AMD’s OpenCL stack moving to ROCr.

# Notable Bug Fixes

* **DirectX 12 mode:**
  + For RDNA and Vega targets, the disassembly now reflects the correct hardware stage in cases where stages are merged.
  + Fixed issue when binary generation happened only for a single device in case of multiple target devices in the same build.
  + You can now generate pipeline binaries (-b option) without generating ISA disassembly (--isa).
  + Fixes to DirectX 12 mode command line tool help manual.
* **Vulkan mode**: fixed an issue where the tool did not work on machines that did not have AMD driver installed. With this fix, the tool is again using a build of the Vulkan driver that is packaged with the tool.
* **GUI application**:
  + Copying disassembly from the disassembly view to the clipboard now includes labels.
  + Ctrl+S in Global Setting view did not prompt the user with a confirmation dialog.
  + Fixed widget highlighting issues in Settings view.
* **Legacy OpenCL mode**: live register analysis and control flow generation are now supported in this mode.
* **OpenGL**: fixed --version option to print the system OpenGL version.
* **Linux**:you can now invoke the tool from a path that contains whitespace characters.
* **Infrastructure:** you can now use the **--cli-only** prebuild script option to generate the solution only for the command line tool.
* [[GitHub-69]](https://github.com/GPUOpen-Tools/radeon_gpu_analyzer/issues/69) Removed usage of obsolete Vulkan enumerations for integration with latest Vulkan SDK.
* [[GitHub-66]](https://github.com/GPUOpen-Tools/radeon_gpu_analyzer/issues/66) Added “X Vega M GL and AMD Radeon RX Vega M GH” product names.

# Known Issues

## Vulkan Live Driver mode (-s vulkan)

* 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 (rocm-cl 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.

## 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.
* Cycle estimate for certain VALU instructions appears as “Varies” instead of 4.

## Legacy OpenCL (cl mode)

* OpenCL mode (-s cl) does not support the following target devices: Bristol Ridge, Lexa (gfx804).
* IL disassembly extraction is not supported when the compilation goes through the Lightning Compiler (the decision whether to compile through the Lightning Compiler is being done by the OpenCL driver and is not controlled by RGA).
* Compilation is disabled in this mode for gfx900.

## OpenGL (opengl mode)

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

## DirectX12 (dx12 mode)

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

## Vulkan offline (vk-offline, vk-spv-offline, vk-spv-txt-offline modes)

SPIR-V support limitations:

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

CapabilityGroupNonUniformClustered,

CapabilityStoragePushConstant16,

CapabilityStoragePushConstant8,

CapabilityInputAttachmentArrayNonUniformIndexingEXT

CapabilityAddresses

CapabilityLinkage

CapabilityKernel

CapabilityVector16

CapabilityFloat16Buffer

CapabilityInt64Atomics

CapabilityImageBasic

CapabilityImageReadWrite

CapabilityImageMipmap

CapabilityPipes

CapabilityDeviceEnqueue

CapabilityLiteralSampler

CapabilityGenericPointer

CapabilityFragmentDensityEXT

CapabilityPhysicalStorageBufferAddressesEXT

CapabilitySampleMaskOverrideCoverageNV

CapabilityGeometryShaderPassthroughNV

CapabilityShaderViewportIndexLayerNV

CapabilityShaderViewportMaskNV

CapabilityShaderStereoViewNV

CapabilityPerViewAttributesNV

CapabilityMeshShadingNV

CapabilityImageFootprintNV

CapabilityFragmentBarycentricNV

CapabilityComputeDerivativeGroupQuadsNV

CapabilityShadingRateNV

CapabilityGroupNonUniformPartitionedNV

CapabilityRayTracingNV

CapabilityComputeDerivativeGroupLinearNV

CapabilitySubgroupShuffleINTEL

CapabilitySubgroupBufferBlockIOINTEL

CapabilitySubgroupImageBlockIOINTEL

1. SPIR-V modes currently only support the following extensions:

SPV\_KHR\_post\_depth\_coverage

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\_EXT\_nonuniform\_qualifier

SPV\_EXT\_shader\_stencil\_export

SPV\_EXT\_shader\_viewport\_index\_layer

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\_gpu\_shader\_half\_float\_fetch

SPV\_AMD\_texture\_gather\_bias\_lod

SPV\_AMD\_gpu\_shader\_int16

SPV\_AMD\_shader\_fragment\_mask

SPV\_AMD\_shader\_image\_load\_store\_lod

SPV\_KHR\_variable\_pointers

SPV\_EXT\_nonuniform\_qualifier

SPV\_GOOGLE\_decorate\_string

SPV\_GOOGLE\_hlsl\_functionality1

SPV\_KHR\_vulkan\_memory\_model

# Notes for OpenCL Mode Users

The legacy “–s cl” mode will continue to use the OpenCL compiler, which is included in the Radeon driver package, while the “-s rocm-cl” mode is independent of the driver. 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 (dx11) 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 (vulkan)

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 (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 (dx12) and DirectX 11 (dx11)

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

## Legacy OpenCL (cl mode)

On Windows, it is recommended to use the latest Adrenalin drivers for the best experience in Legacy OpenCL mode. Note that on Linux, Legacy OpenCL mode is discontinued, with AMD’s OpenCL stack moving to ROCr.

## ROCM OpenCL (rocm-cl mode)

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

## OpenGL (-s opengl)

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