AMD ROCm™ Release Notes v3.10

Revision	**1130**
Issue Date:	**November 2020**

Specification Agreement

This Specification Agreement (this "Agreement") is a legal agreement between Advanced Micro Devices, Inc. ("AMD") and "You" as the recipient of the attached AMD Specification (the "Specification"). If you are accessing the Specification as part of your performance of work for another party, you acknowledge that you have authority to bind such party to the terms and conditions of this Agreement. If you accessed the Specification by any means or otherwise use or provide Feedback (defined below) on the Specification, You agree to the terms and conditions set forth in this Agreement. If You do not agree to the terms and conditions set forth in this Agreement, you are not licensed to use the Specification; do not use, access or provide Feedback about the Specification. In consideration of Your use or access of the Specification (in whole or in part), the receipt and sufficiency of which are acknowledged, You agree as follows:

1. You may review the Specification only (a) as a reference to assist You in planning and designing Your product, service or technology ("Product") to interface with an AMD product in compliance with the requirements as set forth in the Specification and (b) to provide Feedback about the information disclosed in the Specification to AMD.

2. Except as expressly set forth in Paragraph 1, all rights in and to the Specification are retained by AMD. This Agreement does not give You any rights under any AMD patents, copyrights, trademarks or other intellectual property rights. You may not (i) duplicate any part of the Specification; (ii) remove this Agreement or any notices from the Specification, or (iii) give any part of the Specification, or assign or otherwise provide Your rights under this Agreement, to anyone else.

3. The Specification may contain preliminary information, errors, or inaccuracies, or may not include certain necessary information. Additionally, AMD reserves the right to discontinue or make changes to the Specification and its products at any time without notice. The Specification is provided entirely "AS IS." AMD MAKES NO WARRANTY OF ANY KIND AND DISCLAIMS ALL EXPRESS, IMPLIED AND STATUTORY WARRANTIES, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, TITLE OR THOSE WARRANTIES ARISING AS A COURSE OF DEALING OR CUSTOM OF TRADE. AMD SHALL NOT BE LIABLE FOR DIRECT, INDIRECT, CONSEQUENTIAL, SPECIAL, INCIDENTAL, PUNITIVE OR EXEMPLARY DAMAGES OF ANY KIND (INCLUDING LOSS OF BUSINESS, LOSS OF INFORMATION OR DATA, LOST PROFITS, LOSS OF CAPITAL, LOSS OF GOODWILL) REGARDLESS OF THE FORM OF ACTION WHETHER IN CONTRACT, TORT (INCLUDING NEGLIGENCE) AND STRICT PRODUCT LIABILITY OR OTHERWISE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

4. Furthermore, AMD's products are not designed, intended, authorized or warranted for use as components in systems intended for surgical implant into the body, or in other applications intended to support or sustain life, or in any other application in which the failure of AMD's product could create a situation where personal injury, death, or severe property or environmental damage may occur.

5. You have no obligation to give AMD any suggestions, comments or feedback ("Feedback") relating to the Specification. However, any Feedback You voluntarily provide may be used by AMD without restriction, fee or obligation of confidentiality. Accordingly, if You do give AMD Feedback on any version of the Specification, You agree AMD may freely use, reproduce, license, distribute, and otherwise commercialize Your Feedback in any product, as well as has the right to sublicense third parties to do the same. Further, You will not give AMD any Feedback that You may have reason to believe is (i) subject to any patent, copyright or other intellectual property

claim or right of any third party; or (ii) subject to license terms which seek to require any product or intellectual property incorporating or derived from Feedback or any Product or other AMD intellectual property to be licensed to or otherwise provided to any third party.

6. You shall adhere to all applicable U.S. import/export laws and regulations, as well as the import/export control laws and regulations of other countries as applicable. You further agree to not export, re-export, or transfer, directly or indirectly, any product, technical data, software or source code received from AMD under this license, or the direct product of such technical data or software to any country for which the United States or any other applicable government requires an export license or other governmental approval without first obtaining such licenses or approvals; or in violation of any applicable laws or regulations of the United States or the country where the technical data or software was obtained. You acknowledge the technical data and software received will not, in the absence of authorization from U.S. or local law and regulations as applicable, be used by or exported, re-exported or transferred to: (i) any sanctioned or embargoed country, or to nationals or residents of such countries; (ii) any restricted end-user as identified on any applicable government end-user list; or (iii) any party where the end-use involves nuclear, chemical/biological weapons, rocket systems, or unmanned air vehicles. For the most current Country Group listings, or for additional information about the EAR or Your obligations under those regulations, please refer to the U.S. Bureau of Industry and Security's website at *http://www.bis.doc.gov/*.

7. The Software and related documentation are "commercial items", as that term is defined at 48 C.F.R. §2.101, consisting of "commercial computer software" and "commercial computer software documentation", as such terms are used in 48 C.F.R. §12.212 and 48 C.F.R. §227.7202, respectively. Consistent with 48 C.F.R. §12.212 or 48 C.F.R. §227.7202-1 through 227.7202-4, as applicable, the commercial computer software and commercial computer software documentation are being licensed to U.S. Government end users (a) only as commercial items and (b) with only those rights as are granted to all other end users pursuant to the terms and conditions set forth in this Agreement. Unpublished rights are reserved under the copyright laws of the United States.

8. This Agreement is governed by the laws of the State of California without regard to its choice of law principles. Any dispute involving it must be brought in a court having jurisdiction of such dispute in Santa Clara County, California, and You waive any defenses and rights allowing the dispute to be litigated elsewhere. If any part of this agreement is unenforceable, it will be considered modified to the extent necessary to make it enforceable, and the remainder shall continue in effect. The failure of AMD to enforce any rights granted hereunder or to take action against You in the event of any breach hereunder shall not be deemed a waiver by AMD as to subsequent enforcement of rights or subsequent actions in the event of future breaches. This Agreement is the entire agreement between You and AMD concerning the Specification; it may be changed only by a written document signed by both You and an authorized representative of AMD.

DISCLAIMER

The information contained herein is for informational purposes only, and is subject to change without notice. In addition, any stated support is planned and is also subject to change. While every precaution has been taken in the preparation of this document, it may contain technical inaccuracies, omissions and typographical errors, and AMD is under no obligation to update or otherwise correct this information. Advanced Micro Devices, Inc. makes no representations or warranties with respect to the accuracy or completeness of the contents of this document, and assumes no liability of any kind, including the implied warranties of noninfringement, merchantability or fitness for particular purposes, with respect to the operation or use of AMD hardware, software or other products described herein. No license, including implied or arising by estoppel, to any intellectual property rights is granted by this document. Terms and limitations applicable to the purchase or use of AMD's products are as set forth in a signed agreement between the parties or in AMD's Standard Terms and Conditions of Sale.

Table of Contents

SUPPORTED OPERATING SYSTEMS

This document describes the features, fixed issues, and information about downloading and installing the AMD ROCm™ software.

It also covers fixed defects and known issues in this release.

SUPPORTED OPERATING SYSTEMS

List of Supported Operating Systems

The AMD ROCm platform is designed to support the following operating systems:

- Ubuntu 20.04.1 (5.4 and 5.6-oem) and 18.04.5 (Kernel 5.4)
- CentOS 7.8 & RHEL 7.8 (Kernel 3.10.0-1127) (Using devtoolset-7 runtime support)
- CentOS 8.2 & RHEL 8.2 (Kernel 4.18.0) (devtoolset is not required)
- SLES 15 SP2

FRESH INSTALLATION OF AMD ROCM V3.10 RECOMMENDED

A fresh and clean installation of AMD ROCm v3.10 is recommended. An upgrade from previous releases to AMD ROCm v3.10 is not supported. For more information, refer to the *AMD ROCm Installation Guide.*

Note: AMD ROCm release v3.3 or prior releases are not fully compatible with AMD ROCm v3.5 and higher versions. You must perform a fresh ROCm installation if you want to upgrade from AMD ROCm v3.3 or older to 3.5 or higher versions and vice-versa.

Note: *render group* is required only for Ubuntu v20.04. For all other ROCm supported operating systems, continue to use *video group*.

- For ROCm v3.5 and releases thereafter, the *clinfo* path is changed to */opt/rocm/opencl/bin/clinfo*.
- For ROCm v3.3 and older releases, the *clinfo* path remains */opt/rocm/opencl/bin/x86_64/clinfo*.

Note: After an operating system upgrade, AMD ROCm may upgrade automatically and result in an error. This is because AMD ROCm does not support upgrades currently. You must uninstall and reinstall AMD ROCm after an operating system upgrade.

ROCM MULTI-VERSION INSTALLATION UPDATE

With the AMD ROCm v3.10 release, the following ROCm multi-version installation changes apply:

The meta packages *rocm-dkms<version>* are now deprecated for multi-version ROCm installs. For example, *rocm-dkms3.7.0, rocm-dkms3.8.0*.

- Multi-version installation of ROCm should be performed by installing *rocm-dev<version>* using each of the desired ROCm versions. For example, *rocm-dev3.7.0, rocm-dev3.8.0, rocm-dev3.9.0.*
- Version files must be created for each multi-version rocm <= 3.10.0
    - command: echo <version> | sudo tee /opt/rocm-<version>/.info/version
    - example: echo 3.10.0 | sudo tee /opt/rocm-3.10.0/.info/version
- The rock-dkms loadable kernel modules should be installed using a single *rock-dkms* package.
- ROCm v3.9 and above will not set any *ldconfig* entries for ROCm libraries for multi-version installation. Users must set *LD_LIBRARY_PATH* to load the ROCm library version of choice.

NOTE: The single version installation of the ROCm stack remains the same. The *rocm-dkms* package can be used for single version installs and is not deprecated at this time.

AMD ROCm V3.10 DOCUMENTATION UPDATES

AMD ROCM INSTALLATION GUIDE

The AMD ROCm Installation Guide in this release includes the following updates:

- *Supported Environments*
- *Installation Instructions*
- *HIP Installation Instructions*

HIP DOCUMENTATION UPDATES

- HIP FAQ

For more information, see

https://rocmdocs.amd.com/en/latest/Programming_Guides/HIP-FAQ.html#hip-faq

ROCM-SMI API UPDATES

- System DMA (SDMA) Utilization API
- ROCm-SMI Command Line Interface
- Enhanced ROCm SMI Library for Events

For more information, refer to the ROCm SMI API Guide at,

https://github.com/RadeonOpenCompute/ROCm/blob/master/ROCm_SMI_API_Guide_v3.10.pdf

ROCM DATA CENTER TOOL USER GUIDE

- ROCm Data Center Tool Python Binding
- Prometheus plugin integration

For more information, refer to the ROCm Data Center Tool User Guide at:

https://github.com/RadeonOpenCompute/ROCm/blob/master/AMD_ROCm_DataCenter_Tool_User_Guide.pdf

AMD ROCM GENERAL DOCUMENTATION LINKS

- For AMD ROCm documentation, see
  https://rocmdocs.amd.com/en/latest/

- For installation instructions on supped platforms, see
  https://rocmdocs.amd.com/en/latest/Installation_Guide/Installation-Guide.html

- For AMD ROCm binary structure, see
  https://rocmdocs.amd.com/en/latest/Installation_Guide/Installation-Guide.html#build-amd-rocm

- For AMD ROCm Release History, see
  https://rocmdocs.amd.com/en/latest/Installation_Guide/Installation-Guide.html#amd-rocm-version-history

WHAT'S NEW IN THIS RELEASE

ROCM DATA CENTER TOOL

The following enhancements are made to the ROCm Data Center Tool.

Prometheus Plugin for ROCm Data Center Tool

The ROCm Data Center (RDC) Tool now provides the Prometheus plugin, a Python client to collect the telemetry data of the GPU.

The RDC uses Python binding for Prometheus and the collected plugin. The Python binding maps the RDC C APIs to Python using *ctypes*. The functions supported by C APIs can also be used in the Python binding.

For installation instructions, refer to the ROCm Data Center User Guide at

https://github.com/RadeonOpenCompute/ROCm/blob/master/AMD_ROCm_DataCenter_Tool_User_Guide.pdf

Python Binding

The ROCm Data Center (RDC) Tool now uses PyThon Binding for Prometheus and collectd plugins. PyThon binding maps the RDC C APIs to PyThon using *ctypes*. All the functions supported by C APIs can also be used in PyThon binding. A generic PyThon class RdcReader is created to simplify the usage of the RDC:

- Users can only specify the fields they want to monitor. RdcReader creates groups and fieldgroups, watches the fields, and fetches the fields.
- The RdcReader can support both the Embedded and Standalone mode. Standalone mode can be used with and without authentication.
- In the Standalone mode, the RdcReader can automatically reconnect to *rdcd* when connection is lost.When *rdcd* is restarted, the previously created group and fieldgroup may lose. The RdcReader can re-create them and watch the fields after a reconnect.
- If the client is restarted, RdcReader can detect the groups and fieldgroups created previously, and, therefore, can avoid recreating them.
- Users can pass the unit converter if they do not want to use the RDC default unit.

See the following sample program to monitor the power and GPU utilization using the RdcReader:

```
from RdcReader import RdcReader
from RdcUtil import RdcUtil
from rdc_bootstrap import *

default_field_ids = [
        rdc_field_t.RDC_FI_POWER_USAGE,
        rdc_field_t.RDC_FI_GPU_UTIL
]

class SimpleRdcReader(RdcReader):
    def __init__(self):
        RdcReader.__init__(self,ip_port=None, field_ids = default_field_ids,
update_freq=1000000)
    def handle_field(self, gpu_index, value):
        field_name = self.rdc_util.field_id_string(value.field_id).lower()
        print("%d %d:%s %d" % (value.ts, gpu_index, field_name,
value.value.l_int))

if __name__ == '__main__':
    reader = SimpleRdcReader()
    while True:
        time.sleep(1)
        reader.process()
```

For more information about RDC Python binding and the Prometheus plugin integration, refer to the ROCm Data Center Tool User Guide at

https://github.com/RadeonOpenCompute/ROCm/blob/master/AMD_ROCm_DataCenter_Tool_User_Guide.pdf

ROCM SYSTEM MANAGEMENT INFORMATION

System DMA (SDMA) Utilization

ROCM-SMI LIBRARY

Per-process, the SDMA usage is exposed via the ROCm SMI library. The structure *rsmi_process_info_t* is extended to include *sdma_usage*. *sdma_usage* is a 64-bit value that counts the duration (in microseconds) for which the SDMA engine was active during that process's lifetime.

For example, see the *rsmi_compute_process_info_by_pid_get()* API below.

```
/**
* @brief This structure contains information specific to a process.
*/
  typedef struct {
      - - -,
      uint64_t sdma_usage; // SDMA usage in microseconds
  } rsmi_process_info_t;
  rsmi_status_t
      rsmi_compute_process_info_by_pid_get(uint32_t pid,
          rsmi_process_info_t *proc);
```

ROCm-SMI Command Line Interface

The SDMA usage per-process is available using the following command,

```
$ rocm-smi –showpids
```

For more information, see the ROCm SMI API guide at,

https://github.com/RadeonOpenCompute/ROCm/blob/master/ROCm_SMI_API_Guide_v3.10.pdf

Enhanced ROCm SMI Library for Events

ROCm-SMI library clients can now register to receive the following events:

- GPU PRE RESET: This reset event is sent to the client just before a GPU is going to be RESET.
- GPU POST RESET: This reset event is sent to the client after a successful GPU RESET.
- GPU THERMAL THROTTLE: This Thermal throttling event is sent if GPU clocks are throttled.

For more information, refer to the ROCm SMI API Guide at:

https://github.com/RadeonOpenCompute/ROCm/blob/master/ROCm_SMI_API_Guide_v3.10.pdf

ROCm SMI – Command Line Interface Hardware Topology

This feature provides a matrix representation of the GPUs present in a system by providing information of the manner in which the nodes are connected. This is represented in terms of weights, hops, and link types between two given GPUs. It also provides the numa node and the CPU affinity associated with every GPU.

Weight Between Two GPUs			
	GPU0	**GPU1**	**GPU2**
GPU0	0	72	72
GPU1	72	0	40
GPU2	72	40	0

Hops Between Two GPUs			
	GPU0	**GPU1**	**GPU2**
GPU0	0	3	3
GPU1	3	0	2
GPU2	3	2	0

Link Type Between Two GPUs			
	GPU0	**GPU1**	**GPU2**
GPU0	0	PCIE	PCIE
GPU1	PCIE	0	PCIE
GPU2	PCIE	PCIE	0

Numa Nodes	
GPU[0]	: (Topology) Numa Node: 1
GPU[0]	: (Topology) Numa Affinity: 1
GPU[1]	: (Topology) Numa Node: 0
GPU[1]	: (Topology) Numa Affinity: 0
GPU[2]	: (Topology) Numa Node: 0
GPU[2]	: (Topology) Numa Affinity: 0

ROCM MATH AND COMMUNICATION LIBRARIES

New rocSOLVER APIs

The following new rocSOLVER APIs are added in this release:

API	Description
GEQL2	Computes a QL factorization of a general m-by-n matrix A.
GEQLF	GEQLF_STRIDED_BATCHED computes the QL factorization of a batch of general m-by-n matrices.
ORG2L	Generates a m-by-n Matrix Q with orthonormal columns.
ORGQL	Generates a m-by-n Matrix Q with orthonormal columns.
UNG2L	Generates a m-by-n complex Matrix Q with orthonormal columns.
UNGQL	Generates a m-by-n complex Matrix Q with orthonormal columns.

API	Description
ORGTR	Generates a n-by-n orthogonal Matrix Q.
UNGTR	Generates a n-by-n unitary Matrix Q.
ORM2L	Applies a matrix Q with orthonormal columns to a general m-by-n matrix C.
ORMQL	Applies a matrix Q with orthonormal columns to a general m-by-n matrix C.
UNM2L	Applies a complex matrix Q with orthonormal columns to a general m-by-n matrix C.
UNMQL	Applies a complex matrix Q with orthonormal columns to a general m-by-n matrix C.
ORMTR	Applies an orthogonal matrix Q to a general m-by-n matrix C.
UNMTR	Applies a unitary matrix Q to a general m-by-n matrix C.

For more information, refer to

https://rocsolver.readthedocs.io/en/latest/userguide_api.html

RCCL Alltoallv Support in PyTorch

The AMD ROCm v3.10 release includes a new API for ROCm Communication Collectives Library (RCCL). This API sends data from all to all ranks and each rank provides arrays of input/output data counts and offsets.

For details about the functions and parameters, see

https://rccl.readthedocs.io/en/master/allapi.html

AOMP ENHANCEMENTS

AOMP Release 11.11-0

The source code base for this release is the upstream LLVM 11 monorepo release/11.x sources with the hash value

176249bd6732a8044d457092ed932768724a6f06

This release includes fixes to the internal Clang math headers:

- This set of changes applies to clang internal headers to support OpenMP C, C++, and FORTRAN and for HIP C. This establishes consistency between NVPTX and AMDGCN offloading and between OpenMP, HIP, and CUDA. OpenMP uses function variants and header overlays to define device versions of functions. This causes clang LLVM IR codegen to mangled names of variants in both the definition and callsites of functions defined in the internal clang headers. These changes apply to headers found in the installation subdirectory lib/clang/11.0.0/include.
- These changes temporarily eliminate the use of the libm bitcode libraries for C and C++. Although math functions are now defined with internal clang headers, a bitcode library of the C functions defined in the headers is still built for FORTRAN toolchain linking because FORTRAN cannot use c math headers. This bitcode library is installed in lib/libdevice/libm-.bc. The source build of this bitcode library is done with the aomp-extras repository and the component built script build_extras.sh. In the future, we will introduce across the board changes to eliminate massive header files for math libraries and replace them with linking to bitcode libraries.
- Added support for *-gpubnames* in Flang Driver
- Added an example category for Kokkos. The Kokkos example makefile detects if Kokkos is installed and, if not, it builds Kokkos from the Web. Refer to the script kokkos_build.sh in the bin directory on how to build Kokkos. Kokkos now builds cleanly with the OpenMP backend for simple test cases.
- Fixed hostrpc cmake race condition in the build of openmp
- Add a fatal error if missing -Xopenmp-target or -march options when -fopenmp-targets is specified. However, we do forgive this requirement for offloading to host when there is only a single target and that target is the host.
- Fix a bug in InstructionSimplify pass where a comparison of two constants of different sizes found in the optimization pass. This fixes issue #182 which was causing kokkos build failure.
- Fix openmp error message output for no_rocm_device_lib, was asserting.
- Changed linkage on constant per-kernel symbols from external to weaklinkageonly to prevent duplicate symbols when building kokkos.

FIXED DEFECTS

The following defects are fixed in this release.

Defects	Resolution
hipFORT failed to be installed	Code fix
rocm-smi does not work as-is in 3.9, instead prints a reference to documentation	Code fix
--showtopo, weight and hop count shows wrong data	Code fix
Unable to install RDC on CentOS/RHEL 7.8/8.2 & SLES	Code fix
Unable to install mivisionx with error "Problem: nothing provides opencv needed"	Code fix

KNOWN ISSUES

UPGRADE TO AMD ROCM V3.10 NOT SUPPORTED

An upgrade from previous releases to AMD ROCm v3.10 is not supported. A fresh and clean installation of AMD ROCm v3.10 is recommended.

Note: After an operating system upgrade, AMD ROCm may upgrade automatically and result in an error. This is because AMD ROCm does not support upgrades currently. You must uninstall and reinstall AMD ROCm after an operating system upgrade.

DEPRECATIONS

This section describes deprecations and removals in AMD ROCm.

WARNING: COMPILER-GENERATED CODE OBJECT VERSION 2 DEPRECATION

Compiler-generated code object version 2 is no longer supported and will be removed shortly. AMD ROCm users must plan for the code object version 2 deprecation immediately.

Support for loading code object version 2 is also being deprecated with no announced removal release.

HARDWARE AND SOFTWARE SUPPORT

HARDWARE SUPPORT

ROCm is focused on using AMD GPUs to accelerate computational tasks such as machine learning, engineering workloads, and scientific computing. In order to focus our development efforts on these domains of interest, ROCm supports the following targeted set of hardware configurations.

Supported Graphics Processing Units

As the AMD ROCm platform has a focus on specific computational domains, AMD offers official support for a selection of GPUs that are designed to offer good performance and price in these domains.

NOTE: The integrated GPUs of Ryzen are not officially supported targets for ROCm.

ROCm officially supports AMD GPUs that use the following chips:

- GFX8 GPUs
  "Fiji" chips, such as on the AMD Radeon R9 Fury X and Radeon Instinct MI8
  "Polaris 10" chips, such as on the AMD Radeon RX 580 and Radeon Instinct MI6

- GFX9 GPUs
  "Vega 10" chips, such as on the AMD Radeon RX Vega 64 and Radeon Instinct MI25
  "Vega 7nm" chips, such as on the Radeon Instinct MI50, Radeon Instinct MI60 or AMD Radeon VII

ROCm is a collection of software ranging from drivers and runtimes to libraries and developer tools. Some of this software may work with more GPUs than the "officially supported" list above, though AMD does not make any official claims of support for these devices on the ROCm software platform.

The following list of GPUs is enabled in the ROCm software. However, full support is not guaranteed:

- GFX8 GPUs
  "Polaris 11" chips, such as on the AMD Radeon RX 570 and Radeon Pro WX 4100
  "Polaris 12" chips, such as on the AMD Radeon RX 550 and Radeon RX 540

- GFX7 GPUs
  "Hawaii" chips, such as the AMD Radeon R9 390X and FirePro W9100

As described in the next section, GFX8 GPUs require PCI Express 3.0 (PCIe 3.0) with support for PCIe atomics. This requires both CPU and motherboard support. GFX9 GPUs require PCIe 3.0 with support for PCIe atomics by default, but they can operate in most cases without this capability.

The integrated GPUs in AMD APUs are not officially supported targets for ROCm. As described below, "Carrizo", "Bristol Ridge", and "Raven Ridge" APUs are enabled in AMD upstream drivers and the ROCm OpenCL runtime. However, they are not enabled in the HIP runtime, and may not work due to motherboard or OEM hardware limitations. Note, they are not yet officially supported targets for ROCm.

GFX8 GPU

ROCm offers support for the following microprocessors from AMD's "gfx8" generation of GPUs.

Note: The GPUs require a host CPU and platform with PCIe 3.0 with support for PCIe atomics.

GFX8 GPUs			
Fiji (AMD)	**Polaris 10 (AMD)**	**Polaris 11 (AMD)**	**Polaris 12 (Lexa) (AMD)**
• Radeon R9 Fury • Radeon R9 Nano • Radeon R9 Fury X • Radeon Pro Duo (Fiji) • FirePro S9300 X2 • Radeon Instinct MI8	• Radeon RX 470 • Radeon RX 480 • Radeon RX 570 • Radeon RX 580 • Radeon Pro Duo (Polaris) • Radeon Pro WX 5100 • Radeon Pro WX 7100 • Radeon Instinct MI6	• Radeon RX 460 • Radeon RX 560 • Radeon Pro WX 4100	• Radeon RX 540 • Radeon RX 550 • Radeon Pro WX 2100 • Radeon Pro WX 3100

GFX9 GPU

ROCm offers support for two chips from AMD's most recent "gfx9" generation of GPUs.

GFX9 GPUs	
Vega 10 (AMD)	**Vega 7nm (AMD)**
• Radeon RX Vega 56 • Radeon RX Vega 64 • Radeon Vega Frontier Edition • Radeon Pro WX 8200 • Radeon Pro WX 9100 • Radeon Pro V340 • Radeon Pro V340 MxGPU • Radeon Instinct MI25 **Note:** ROCm does not support Radeon Pro SSG.	• Radeon VII • Radeon Instinct MI50 • Radeon Instinct MI60

SUPPORTED CPU

As described above, GFX8 GPUs require PCIe 3.0 with PCIe atomics to run ROCm. In particular, the CPU and every active PCIe point between the CPU and GPU require support for PCIe 3.0 and PCIe atomics. The CPU root must indicate PCIe AtomicOp Completion capabilities and any intermediate switch must indicate PCIe AtomicOp Routing capabilities.

The current CPUs which support PCIe Gen3 + PCIe Atomics are:

- AMD Ryzen CPUs
- CPUs in AMD Ryzen APUs
- AMD Ryzen Threadripper CPUs
- AMD EPYC CPUs
- Intel Xeon E7 v3 or newer CPUs
- Intel Xeon E5 v3 or newer CPUs
- Intel Xeon E3 v3 or newer CPUs
- Intel Core i7 v4, Core i5 v4, Core i3 v4 or newer CPUs (i.e. Haswell family or newer)
- Some Ivy Bridge-E systems

Beginning with ROCm 1.8, GFX9 GPUs (such as Vega 10) no longer require PCIe atomics. We have similarly made more options available for many PCIe lanes. GFX9 GPUs can now be run on CPUs without PCIe atomics and on older PCIe generations, such as PCIe 2.0. This is not supported on GPUs below GFX9, e.g. GFX8 cards in the Fiji and Polaris families.

If you are using any PCIe switches in your system, please note that PCIe Atomics are only supported on some switches, such as Broadcom PLX. When you install your GPUs, make sure you install them in a PCIe 3.0 x16, x8, x4, or x1 slot attached either directly to the CPU's Root I/O controller or via a PCIe switch directly attached to the CPU's Root I/O controller.

In our experience, many issues stem from trying to use consumer motherboards which provide physical x16 connectors that are electrically connected as e.g. PCIe 2.0 x4, PCIe slots connected via the Southbridge PCIe I/O controller, or PCIe slots connected through a PCIe switch that does not support PCIe atomics.

If you attempt to run ROCm on a system without proper PCIe atomic support, you may see an error in the kernel log (`dmesg`):

```
kfd: skipped device 1002:7300, PCI rejects atomics
```

Experimental support for our Hawaii (GFX7) GPUs (Radeon R9 290, R9 390, FirePro W9100, S9150, S9170) does not require or take advantage of PCIe Atomics. However, AMD recommends that you use a CPU from the list provided above for compatibility purposes.

NOT SUPPORTED OR LIMITED SUPPORT UNDER ROCM

Limited Support

- ROCm 3.10.x should support PCIe 2.0 enabled CPUs such as the AMD Opteron, Phenom, Phenom II, Athlon, Athlon X2, Athlon II and older Intel Xeon and Intel Core Architecture and Pentium CPUs. However, we have done very limited testing on these configurations, since our test farm has been catering to CPUs listed above. This is where we need community support.
  Please report these issues.
- Thunderbolt 1, 2, and 3 enabled breakout boxes should now be able to work with ROCm. Thunderbolt 1 and 2 are PCIe 2.0 based, and thus are only supported with GPUs that do not require PCIe 3.0 atomics (e.g. Vega 10). However, we have done no testing on this configuration and would need community support due to limited access to this type of equipment.

- AMD "Carrizo" and "Bristol Ridge" APUs are enabled to run OpenCL, but do not yet support HIP or our libraries built on top of these compilers and runtimes.
    - As of ROCm 2.1, "Carrizo" and "Bristol Ridge" require the use of upstream kernel drivers.
    - In addition, various "Carrizo" and "Bristol Ridge" platforms may not work due to OEM and ODM choices when it comes to key configurations parameters such as inclusion of the required CRAT tables and IOMMU configuration parameters in the system BIOS.
    - Before purchasing such a system for ROCm, please verify that the BIOS provides an option for enabling IOMMUv2 and that the system BIOS properly exposes the correct CRAT table. Inquire with your vendor about the latter.

- AMD "Raven Ridge" APUs are enabled to run OpenCL, but do not yet support HIP or our libraries built on top of these compilers and runtimes.
    - As of ROCm 2.1, "Raven Ridge" requires the use of upstream kernel drivers.
    - In addition, various "Raven Ridge" platforms may not work due to OEM and ODM choices when it comes to key configurations parameters such as inclusion of the required CRAT tables and IOMMU configuration parameters in the system BIOS.
    - Before purchasing such a system for ROCm, please verify that the BIOS provides an option for enabling IOMMUv2 and that the system BIOS properly exposes the correct CRAT table. Inquire with your vendor about the latter.

NOT SUPPORTED

- "Tonga", "Iceland", "Vega M", and "Vega 12" GPUs are not supported.
- AMD does not support GFX8-class GPUs (Fiji, Polaris, etc.) on CPUs that do not have PCIe3.0 with PCIe atomics.
    - AMD Carrizo and Kaveri APUs as hosts for such GPUs are not supported
    - Thunderbolt 1 and 2 enabled GPUs are not supported by GFX8 GPUs on ROCm. Thunderbolt 1 & 2 are based on PCIe 2.0.

In the default ROCm configuration, GFX8 and GFX9 GPUs require PCI Express 3.0 with PCIe atomics. The ROCm platform leverages these advanced capabilities to allow features such as user-level submission of work from the host to the GPU. This includes PCIe atomic Fetch and Add, Compare and Swap, Unconditional Swap, and AtomicOp Completion.

CPU Supporting PCIe 3.0 + PCIe Atomics

AMD	INTEL
Ryzen CPUs (Family 17h Model 01h-0Fh) • Ryzen 3 1300X • Ryzen 3 2300X • Ryzen 5 1600X • Ryzen 5 2600X • Ryzen 7 1800X • Ryzen 7 2700X	Intel Core i3, i5, and i7 CPUs from Haswell and beyond. This includes: • Haswell CPUs such as the Core i7 4790K • Broadwell CPUs such as the Core i7 5775C • Skylake CPUs such as the Core i7 6700K • Kaby Lake CPUs such as the Core i7 7740X • Coffee Lake CPUs such as the Core i7 8700K • Xeon CPUs from “v3” and newer • Some models of “Ivy Bridge-E” processors
Ryzen APUs (Family 17h Model 10h-1Fh – previously code-named Raven Ridge) such as: • Athlon 200GE • Ryzen 5 2400G **Note:** The integrated GPU in these devices is not guaranteed to work with ROCm.	
Ryzen Threadripper Workstation CPUs (Family 17h Model 01h-0Fh) such as: • Ryzen Threadripper 1950X • Ryzen Threadripper 2990WX	
EPYC Server CPUs (Family 17h Model 01h-0Fh) such as: • Epyc 7551P • Epyc 7601	