AMD ROCm™ Release Notes v3.7

Revision	**0820**
Issue Date:	**August 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. 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. AMD, the AMD Arrow logo, Radeon, Ryzen, Epyc, and combinations thereof are trademarks of Advanced Micro Devices, Inc.
Google® is a registered trademark of Google LLC.
PCIe® is a registered trademark of PCI-SIG Corporation.
Linux is the registered trademark of Linus Torvalds in the U.S. and other countries.
Ubuntu and the Ubuntu logo are registered trademarks of Canonical Ltd.

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 known issues in the AMD ROCm v3.7 release.

SUPPORTED OPERATING SYSTEMS

Support for Ubuntu 20.04

In this release, AMD ROCm extends support to Ubuntu 20.04, including dual kernel.

List of Supported Operating Systems

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

- Ubuntu 20.04 and 18.04.4 (Kernel 5.3)
- 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 SP1

FRESH INSTALLATION OF AMD ROCM V3.7 RECOMMENDED

A fresh and clean installation of AMD ROCm v3.7 is recommended. An upgrade from previous releases to AMD ROCm v3.7 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.

AMD ROCm V3.7 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*

AMD ROCM – HIP DOCUMENTATION UPDATES

Texture and Surface Functions

The documentation for Texture and Surface functions is updated and available at:

https://rocmdocs.amd.com/en/latest/Programming_Guides/Kernel_language.html

Warp Shuffle Functions

The documentation for Warp Shuffle functions is updated and available at:

https://rocmdocs.amd.com/en/latest/Programming_Guides/Kernel_language.html

Compiler Defines and Environment Variables

The documentation for the updated HIP Porting Guide is available at:

https://rocmdocs.amd.com/en/latest/Programming_Guides/HIP-porting-guide.html#hip-porting-guide

AMD ROCM DEBUG AGENT

- *ROCm Debug Agent Library*

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

AOMP ENHANCEMENTS

AOMP is a scripted build of LLVM. It supports OpenMP target offload on AMD GPUs. Since AOMP is a Clang/LLVM compiler, it also supports GPU offloading with HIP, CUDA, and OpenCL.

The following enhancements are made for AOMP in this release:

- OpenMP 5.0 is enabled by default. You can use *-fopenmp-version=45* for OpenMP 4.5 compliance
- Restructured to include the ROCm compiler
- B=Bitcode search path using hip policy HIP_DEVICE_LIB_PATH and hip-devic-lib command line option to enable *global_free for kmpc_impl_free*

Restructured *hostrpc*, including:

- Replaced *hostcall* register functions with *handlePayload*(service, payload). Note, *handlPayload* has a simple switch to call the correct service handler function.
- Removed the *WITH_HSA* macro
- Moved the *hostrpc* stubs and host fallback functions into a single library and the *include* file. This enables the stubs openmp cpp source instead of hip and reorganizes the directory *openmp/libomptarget/hostrpc*.
- Moved *hostrpc_invoke.cl* to *DeviceRTLs/amdgcn*.
- Generalized the *vargs* processing in *printf* to work for any *vargs* function to execute on the host, including a *vargs* function that uses a function pointer.
- Reorganized files, added *global_allocate* and *global_free*.
- Fixed llvm TypeID enum to match the current upstream llvm TypeID.
- Moved *strlen_max* function inside the declare target #ifdef _DEVICE_GPU in *hostrpc.cpp* to resolve linker failure seen in *pfspecifier_str* smoke test.
- Fixed *AOMP_GIT_CHECK_BRANCH* in aomp_common_vars to not block builds in Red Hat if the repository is on a specific commit hash.
- Simplified and reduced the size of *openmp* host runtime
- Switched to default OpenMP 5.0

For more information, see *https://github.com/ROCm-Developer-Tools/aomp*

ROCM COMMUNICATIONS COLLECTIVE LIBRARY

Compatibility with NVIDIA Communications Collective Library v2.7 API

ROCm Communications Collective Library (RCCL) is now compatible with the NVIDIA Communications Collective Library (NCCL) v2.7 API.

RCCL (pronounced "Rickle") is a stand-alone library of standard collective communication routines for GPUs, implementing all-reduce, all-gather, reduce, broadcast, reduce-scatter, gather, scatter, and all-to-all. There is also initial support for direct GPU-to-GPU send and receive operations. It has been optimized to achieve high bandwidth on platforms using PCIe, xGMI as well as networking using InfiniBand Verbs or TCP/IP sockets. RCCL supports an arbitrary number of GPUs installed in a single node or multiple nodes, and can be used in either single- or multi-process (e.g., MPI) applications.

The collective operations are implemented using ring and tree algorithms and have been optimized for throughput and latency. For best performance, small operations can be either batched into larger operations or aggregated through the API.

For more information about RCCL APIs and compatibility with NCCL v2.7, see

https://rccl.readthedocs.io/en/develop/index.html

Singular Value Decomposition of Bi-diagonal Matrices

Rocsolver_bdsqr now computes the Singular Value Decomposition (SVD) of bi-diagonal matrices. It is an auxiliary function for the SVD of general matrices (function rocsolver_gesvd).

BDSQR computes the singular value decomposition (SVD) of a n-by-n bidiagonal matrix B.

The SVD of B has the following form:

```
B = Ub * S * Vb'
```

where

- S is the n-by-n diagonal matrix of singular values of B
- the columns of Ub are the left singular vectors of B
- the columns of Vb are its right singular vectors

The computation of the singular vectors is optional; this function accepts input matrices U (of size nu-by-n) and V (of size n-by-nv) that are overwritten with U*Ub and Vb’*V. If nu = 0 no left vectors are computed; if nv = 0 no right vectors are computed.

Optionally, this function can also compute Ub'*C for a given n-by-nc input matrix C.

PARAMETERS

- `[in] handle`: rocblas_handle.
- `[in] uplo`: rocblas_fill.

  Specifies whether B is upper or lower bidiagonal.

- `[in] n`: rocblas_int. n >= 0.

  The number of rows and columns of matrix B.

- `[in] nv`: rocblas_int. nv >= 0.

  The number of columns of matrix V.

- `[in] nu`: rocblas_int. nu >= 0.

  The number of rows of matrix U.

- `[in] nc`: rocblas_int. nu >= 0.

  The number of columns of matrix C.

- `[inout] D`: pointer to real type. Array on the GPU of dimension n.

  On entry, the diagonal elements of B. On exit, if info = 0, the singular values of B in decreasing order; if info > 0, the diagonal elements of a bidiagonal matrix orthogonally equivalent to B.

- `[inout] E`: pointer to real type. Array on the GPU of dimension n-1.

  On entry, the off-diagonal elements of B. On exit, if info > 0, the off-diagonal elements of a bidiagonal matrix orthogonally equivalent to B (if info = 0 this matrix converges to zero).

- `[inout] V`: pointer to type. Array on the GPU of dimension ldv*nv.

  On entry, the matrix V. On exit, it is overwritten with Vb'*V. (Not referenced if nv = 0).

- `[in] ldv`: rocblas_int. ldv >= n if nv > 0, or ldv >=1 if nv = 0.

  Specifies the leading dimension of V.

- `[inout] U`: pointer to type. Array on the GPU of dimension ldu*n.

  On entry, the matrix U. On exit, it is overwritten with U*Ub. (Not referenced if nu = 0).

- `[in] ldu`: rocblas_int. ldu >= nu.

  Specifies the leading dimension of U.

- `[inout] C`: pointer to type. Array on the GPU of dimension ldc*nc.

  On entry, the matrix C. On exit, it is overwritten with Ub'*C. (Not referenced if nc = 0).

- `[in] ldc`: rocblas_int. ldc >= n if nc > 0, or ldc >=1 if nc = 0.

  Specifies the leading dimension of C.

- `[out] info`: pointer to a rocblas_int on the GPU.

  If info = 0, successful exit. If info = i > 0, i elements of E have not converged to zero.

For more information, see

https://rocsolver.readthedocs.io/en/latest/userguide_api.html#rocsolver-type-bdsqr

rocSPARSE_gemmi() Operations for Sparse Matrices

This enhancement provides a dense matrix sparse matrix multiplication using the CSR storage format.

`rocsparse_gemmi` multiplies the scalar αα with a dense $m\times k$m×k matrix AA and the sparse $k\times n$k×n matrix BB defined in the CSR storage format, and adds the result to the dense $m\times n$m×n matrix CC that is multiplied by the scalar ββ, such that

C:=α⋅op(A)⋅op(B)+β⋅CC:=α⋅op(A)⋅op(B)+β⋅C

with

op(A)= A,AT,AH,if trans_A == rocsparse_operation_noneif trans_A == rocsparse_operation_transposeif trans_A == rocsparse_operation_conjugate_transposeop(A)={A,if trans_A == rocsparse_operation_noneAT,if trans_A == rocsparse_operation_transposeAH,if trans_A == rocsparse_operation_conjugate_transpose and

op(B)= /(()//B,BT,BH,if trans_B == rocsparse_operation_noneif trans_B == rocsparse_operation_transposeif trans_B == rocsparse_operation_conjugate_transposeop(B)={B,if trans_B == rocsparse_operation_noneBT,if trans_B == rocsparse_operation_transposeBH,if trans_B == rocsparse_operation_conjugate_transpose

Note: This function is non-blocking and executed asynchronously with the host. It may return before the actual computation has finished.

For more information and examples, see

https://rocsparse.readthedocs.io/en/master/usermanual.html#rocsparse-gemmi

KNOWN ISSUES

The following are the known issues in this release.

(AOMP) 'Undefined Hidden Symbol' Linker Error Causes Compilation Failure in HIP

The HIP example device_lib fails to compile due to unreferenced symbols with Link Time Optimization resulting in 'undefined hidden symbol' errors.

This issue is under investigation and there is no known workaround at this time.

MIGraphX Fails for fp16 Datatype

The MIGraphX functionality does not work for the fp16 datatype.

The following workaround is recommended:

- Use the AMD ROCm v3.3 of MIGraphX
  Or
- Build MIGraphX v3.7 from the source using AMD ROCm v3.3

Missing Google Test Installation May Cause RCCL Unit Test Compilation Failure

Users of the RCCL *install.sh* script may encounter an RCCL unit test compilation error. It is recommended to use CMAKE directly instead of *install.sh* to compile RCCL. Ensure Google Test 1.10+ is available in the CMAKE search path.

As a workaround, use the latest RCCL from the GitHub development branch at:

https://github.com/ROCmSoftwarePlatform/rccl/pull/237

Issue with Peer-to-Peer Transfers

Using peer-to-peer (P2P) transfers on systems without the hardware P2P assistance may produce incorrect results.

Ensure the hardware supports peer-to-peer transfers and enable the peer-to-peer setting in the hardware to resolve this issue.

Partial Loss of Tracing Events for Large Applications

An internal tracing buffer allocation issue can cause a partial loss of some tracing events for large applications.

As a workaround, rebuild the roctracer/rocprofiler libraries from the GitHub 'roc-3.7' branch at:

- *https://github.com/ROCm-Developer-Tools/rocprofiler*
- *https://github.com/ROCm-Developer-Tools/roctracer*

GPU Kernel C++ Names Not Demangled

GPU kernel C++ names in the profiling traces and stats produced by *'—hsa-trace'* option are not demangled.

As a workaround, users may choose to demangle the GPU kernel C++ names as required.

'rocprof' option '--parallel-kernels' Not Supported in This Release

'rocprof' option '--parallel-kernels' is available in the options list, however, it is not fully validated and supported in this release.

Random Soft Hang Observed When Running ResNet-Based Models

A random soft hang is observed when running ResNet-based models for a loop run of more than 25 to 30 hours. The issue is observed on both PyTorch and TensorFlow frameworks.

You can terminate the unresponsive process to temporarily resolve the issue.

There is no known workaround at this time.

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.

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 GPUS

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 GPUS

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 CPUS

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 3.7.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.

Current CPUs which support 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	