Title: ROCm

URL: https://en.wikipedia.org/wiki/ROCm

PageID: 57013254

Categories: Category:AMD software, Category:Application programming interfaces, Category:Concurrent computing, Category:GPGPU, Category:GPGPU libraries, Category:Graphics cards, Category:Graphics hardware, Category:Heterogeneous computing, Category:Machine learning, Category:Parallel computing, Category:Supercomputers

---

ROCm [3] is an Advanced Micro Devices (AMD) software stack for graphics processing unit (GPU) programming. ROCm spans several domains, including general-purpose computing on graphics processing units (GPGPU), high performance computing (HPC), and heterogeneous computing . It offers several programming models: HIP (GPU-kernel-based programming), OpenMP (directive-based programming), and OpenCL .

ROCm is free, libre and open-source software (except the GPU firmware blobs [4]), and it is distributed under various licenses. ROCm initially stood for Radeon Open Compute platfor m ; however, due to Open Compute being a registered trademark, ROCm is no longer an acronym — it is simply AMD's open-source stack designed for GPU compute.

Background

The first GPGPU software stack from ATI /AMD was Close to Metal , which became Stream .

ROCm was launched around 2016 [5] with the Boltzmann Initiative . [6] ROCm stack builds upon previous AMD GPU stacks; some tools trace back to GPUOpen and others to the Heterogeneous System Architecture (HSA).

Heterogeneous System Architecture Intermediate Language

HSAIL [7] was aimed at producing a middle-level, hardware-agnostic intermediate representation that could be JIT-compiled to the eventual hardware (GPU, FPGA...) using the appropriate finalizer. This approach was dropped for ROCm: now it builds only GPU code, using LLVM , and its AMDGPU backend that was upstreamed, [8] although there is still research on such enhanced modularity with LLVM MLIR. [9]

Programming abilities

ROCm as a stack ranges from the kernel driver to the end-user applications.

AMD has introductory videos about AMD GCN hardware, [10] and ROCm programming [11] via its learning portal. [12]

One of the best technical introductions about the stack and ROCm/HIP programming, remains, to date, to be found on Reddit. [13]

Hardware support

ROCm is primarily targeted at discrete professional GPUs, [14] but consumer GPUs and APUs of the same architecture as a supported professional GPU are known to work with ROCm. For example, all professional GPUs of the RDNA 2 architecture are officially supported by ROCm 5.x; users report that Consumer RDNA2 units such as the Radeon 6800M APU and the Radeon 6700XT GPU also work. [15]

Professional-grade GPUs

Consumer-grade GPUs

Software ecosystem

Learning resources

AMD ROCm product manager Terry Deem gave a tour of the stack. [22]

Third-party integration

The main consumers of the stack are machine learning and high-performance computing/GPGPU applications.

Machine learning

Various deep learning frameworks have a ROCm backend: [23]

PyTorch

TensorFlow

ONNX

MXNet

CuPy [24]

MIOpen

Caffe

Iree (which uses LLVM Multi-Level Intermediate Representation (MLIR))

llama.cpp

Supercomputing

ROCm is gaining significant traction in the top 500 . [25] ROCm is used with the Exascale supercomputers El Capitan [26] [27] and Frontier .

Some related software is to be found at AMD Infinity hub .

Other acceleration & graphics interoperation

As of version 3.0, Blender can now use HIP compute kernels for its renderer cycles. [28]

Other Languages

Julia has the AMDGPU.jl package, [29] which integrates with LLVM and selects components of the ROCm stack. Instead of compiling code through HIP, AMDGPU.jl uses Julia's compiler to generate LLVM IR directly, which is later consumed by LLVM to generate native device code. AMDGPU.jl uses ROCr's HSA implementation to upload native code onto the device and execute it, similar to how HIP loads its own generated device code.

AMDGPU.jl also supports integration with ROCm's rocBLAS (for BLAS), rocRAND (for random number generation), and rocFFT (for FFTs). Future integration with rocALUTION, rocSOLVER, MIOpen, and certain other ROCm libraries is planned.

Software distribution

Official

Installation instructions are provided for Linux and Windows in the official AMD ROCm documentation . ROCm software is currently spread across several public GitHub repositories. Within the main public meta-repository , there is an XML manifest for each official release: using git-repo , a version control tool built on top of Git , is the recommended way to synchronize with the stack locally. [30]

AMD starts distributing containerized applications for ROCm, notably scientific research applications gathered under AMD Infinity Hub . [31]

AMD distributes itself packages tailored to various Linux distributions.

Third-party

There is a growing third-party ecosystem packaging ROCm .

Linux distributions are officially packaging (natively) ROCm, with various degrees of advancement: Arch Linux , [32] Gentoo, [33] Debian , Fedora , [34] GNU Guix , and NixOS .

There are Spack packages. [35]

Components

There is one kernel-space component, ROCk, and the rest - there is roughly a hundred components in the stack - is made of user-space modules.

The unofficial typographic policy is to use: uppercase ROC lowercase following for low-level libraries, i.e. ROCt, and the contrary for user-facing libraries, i.e. rocBLAS. [36]

AMD is active developing with the LLVM community, but upstreaming is not instantaneous, and as of January 2022, is still lagging. [37] AMD still officially packages various LLVM forks [38] [39] [9] for parts that are not yet upstreamed – compiler optimizations destined to remain proprietary, debug support, OpenMP offloading, etc.

Low-level

ROCk – Kernel driver

ROCm – Device libraries

Support libraries implemented as LLVM bitcode. These provide various utilities and functions for math operations, atomics, queries for launch parameters, on-device kernel launch, etc.

ROCt – Thunk

The thunk is responsible for all the thinking and queuing that goes into the stack.

ROCr – Runtime

The ROC runtime is a set of APIs/libraries that allows the launch of compute kernels by host applications. It is AMD's implementation of the HSA runtime API. [40] It is different from the ROC Common Language Runtime.

ROCm – CompilerSupport

ROCm code object manager is in charge of interacting with LLVM intermediate representation .

Mid-level

ROCclr Common Language Runtime

The common language runtime is an indirection layer adapting calls to ROCr on Linux and PAL on windows.

It used to be able to route between different compilers, like the HSAIL-compiler. It is now being absorbed by the upper indirection layers (HIP and OpenCL).

OpenCL

ROCm ships its installable client driver (ICD) loader and an OpenCL [41] implementation bundled together .

As of January 2022, ROCm 4.5.2 ships OpenCL 2.2, and is lagging behind competition. [42]

HIP – Heterogeneous Interface for Portability

The AMD implementation for its GPUs is called HIPAMD . There is also a CPU implementation mostly for demonstration purposes.

HIPCC

HIP builds a `HIPCC` compiler that either wraps Clang and compiles with LLVM open AMDGPU backend, or redirects to the NVIDIA compiler . [43]

HIPIFY

HIPIFY is a source-to-source compiling tool. It translates CUDA to HIP and reverse, either using a Clang-based tool, or a sed-like Perl script.

GPUFORT

Like HIPIFY, GPUFORT is a tool compiling source code into other third-generation-language sources, allowing users to migrate from CUDA Fortran to HIP Fortran. It is also in the repertoire of research projects, even more so. [44]

High-level

ROCm high-level libraries are usually consumed directly by application software, such as machine learning frameworks. Most of the following libraries are in the General Matrix Multiply (GEMM) category, which GPU architecture excels at.

The majority of these user-facing libraries comes in dual-form: hip for the indirection layer that can route to Nvidia hardware, and roc for the AMD implementation. [45]

rocBLAS / hipBLAS

rocBLAS and hipBLAS are central in high-level libraries, it is the AMD implementation for Basic Linear Algebra Subprograms .

It uses the library Tensile privately.

rocSOLVER / hipSOLVER

This pair of libraries constitutes the LAPACK implementation for ROCm and is strongly coupled to rocBLAS.

Utilities

ROCm developer tools : Debug, tracer, profiler, System Management Interface, Validation suite, Cluster management.

GPUOpen tools : GPU analyzer, memory visualizer...

External tools: radeontop (TUI overview)

Comparison with competitors

ROCm competes with other GPU computing stacks: Nvidia CUDA and Intel OneAPI .

Nvidia CUDA

Nvidia's CUDA is closed-source, whereas AMD ROCm is open source. There is open-source software built on top of the closed-source CUDA, for instance RAPIDS .

CUDA is able to run on consumer GPUs, whereas ROCm support is mostly offered for professional hardware such as AMD Instinct and AMD Radeon Pro .

Nvidia provides a C/C++-centered frontend and its Parallel Thread Execution (PTX) LLVM GPU backend as the Nvidia CUDA Compiler (NVCC).

Intel OneAPI

Like ROCm, oneAPI is open source, and all the corresponding libraries are published on its GitHub Page .

Unified Acceleration Foundation (UXL)

Unified Acceleration Foundation (UXL) is a new technology consortium that are working on the continuation of the OneAPI initiative, with the goal to create a new open standard accelerator software ecosystem, related open standards and specification projects through Working Groups and Special Interest Groups (SIGs). The goal will compete with Nvidia's CUDA. The main companies behind it are Intel, Google, Arm, Qualcomm, Samsung, Imagination, and VMware. [46]

See also

AMD Software – a general overview of AMD's drivers, APIs, and development endeavors.

GPUOpen – AMD's complementary graphics stack

AMD Radeon Software – AMD's software distribution channel

References

External links

"ROCm official documentation" . AMD . February 10, 2022.

"ROCm Learning Center" . AMD . January 25, 2022.

"ROCm official documentation on the github super-project" . AMD . January 25, 2022.

"ROCm official documentation - pre 5.0" . AMD . January 19, 2022.

"GPU-Accelerated Applications with AMD Instinct Accelerators & AMD ROCm Software" (PDF) . AMD . January 25, 2022.

"AMD Infinity Hub" . AMD . January 25, 2022. — Docker containers for scientific applications.

v

t

e

x86-64

3DNow!

Geode

Duron

Sempron

Turion

Phenom

Athlon

FX

Ryzen

Opteron

Epyc

Radeon

AMD Radeon Software

AMDGPU

AMD PowerTune

CrossFire

Eyefinity

FreeSync

Mantle

AGESA

AMD Turbo Core

Cool'n'Quiet

AMD Platform Security Processor

Ryzen AI

High Bandwidth Memory (HBM)

Super Socket 7 (Super 7)

939

AM2

AM2+

AM3

AM3+

FM1

FM2

FM2+

AM1

AM4

TR4

sTRX4

AM5

Slot A

563

S1

FS1

FT1

FP2

FT3

FP3

940

F

F+

G3

G34

C32

SP3

Socket A (Socket 462)

754

List of AMD microprocessors

List of AMD graphics processing units

List of AMD accelerated processing units

List of AMD CPU microarchitectures

List of AMD chipsets

Jerry Sanders

Jerry Sanders (1969–2002)

Hector Ruiz (2002–2008)

Dirk Meyer (2008–2011)

Rory Read (2011–2014)

Lisa Su (2014–present)

ATI Technologies

SeaMicro

Xilinx

AMD–Chinese joint venture Hygon Information Technology

Hygon Information Technology

TF-AMD Tongfu Microelectronics

Tongfu Microelectronics

Intel Corp. v. Advanced Micro Devices, Inc. (2004)

Advanced Micro Devices, Inc. v. Intel Corp. (2005)

Vulkan (API)

NexGen

Spansion

AMD Live!

Performance Rating

Torrenza

GlobalFoundries

Italics indicates an unreleased product (e.g. socket)

Strikethrough indicates a product that was never released.

Mixed indicates sockets that are designed for or integrated with one or more platforms.

v

t

e

AGESA

AMDgpu

AMD Software

Vivado

Xilinx ISE

ROCm

GPUOpen

Spider

Dragon

Horus

Cool'n'Quiet

High Bandwidth Memory

PowerNow!

PowerPlay

PowerTune

Turbo Core

ASTC

AMD Wraith

Virtex

XDNA Ryzen AI

Ryzen AI

X86-64

3DNow!

AVX

XOP

CVT16/F16C

FMA FMA4 FMA3

FMA4

FMA3

BMI ABM BMI1 TBM

ABM

BMI1

TBM

SSE5

ASF

AES

v

t

e

Wonder

Mach

Rage

All-in-Wonder (before 2000)

R100

R200

R300

R400

R500

All-in-Wonder (after 1999)
HD 2000
HD 3000
HD 4000
HD 5000
HD 6000
HD 7000
HD 8000
200
300
400
500
RX Vega
600
RX 5000
RX 6000
RX 7000
RX 9000
Unified Video Decoder (UVD)
Video Coding Engine (VCE)
Video Core Next (VCN)
TrueAudio
Eyefinity
FreeSync
PowerTune
CrossFire
Hybrid Graphics
HyperMemory
HyperZ
HSA
RDNA 2 3 4
2
3
4
AMD Radeon Software HD3D
HD3D
ROCm
AMDGPU

GPUOpen TressFX

TressFX

HLSL2GLSL

AMD APP SDK

Catalyst

Close to Metal

CodeAnalyst

GPU PerfStudio

Mantle

CodeXL

Radeon Pro

Radeon Instinct

FireGL/FirePro

FireMV

FireStream

Flipper (GameCube)

Xenos (Xbox 360)

Hollywood (Wii)

Liverpool (PlayStation 4)

Durango (Xbox One)

Neo (PlayStation 4 Pro)

Scorpio (Xbox One X)

Atari VCS (2021)

PlayStation 5

Xbox Series X/S

Steam Deck

v

t

e

Intel GT Xe Arc

GT

Xe

Arc

Nvidia GeForce Quadro Tesla Tegra

GeForce

Quadro

Tesla

Tegra

AMD Radeon Radeon Pro Instinct

Radeon

Radeon Pro

Instinct

Matrox

InfiniteReality

NEC µPD7220

3dfx Voodoo

S3

Glaze3D

Apple silicon

Jingjia Micro

Tseng Labs

SiS

Adreno

Apple silicon

Mali

PowerVR

VideoCore

Vivante

Imageon

Intel 2700G

Compute kernel

Fabrication CMOS FinFET MOSFET

CMOS

FinFET

MOSFET

Graphics pipeline Geometry Vertex

Geometry

Vertex

HDR rendering

MAC

Rasterisation Shading

Shading

Ray-tracing

SIMD SIMT

SIMT

Tessellation

T&L;

Tiled rendering

Unified shader model

Blitter

Geometry processor

Input–output memory management unit

Render output unit

Shader unit

Stream processor

Tensor unit

Texture mapping unit

Video display controller

Video processing unit

DMA

Framebuffer

SGRAM GDDR GDDR2 GDDR3 GDDR4 GDDR5 GDDR6 GDDR7

GDDR

GDDR2

GDDR3

GDDR4

GDDR5

GDDR6

GDDR7

HBM HBM2 HBM2E HBM3 HBM-PIM HBM3E

HBM2

HBM2E

HBM3

HBM-PIM

HBM3E

Memory bandwidth

Memory controller

Shared graphics memory

Texture memory

VRAM

IP core

Discrete graphics Clustering Switching

Clustering

Switching

External graphics
Integrated graphics
System on a chip
Clock rate
Display resolution
Fillrate Pixel/s Texel/s
Pixel/s
Texel/s
FLOP/s
Frame rate
Performance per watt
Transistor count
2D Scrolling Sprite Tile
Scrolling
Sprite
Tile
3D GI Texture
GI
Texture
ASIC
GPGPU
Graphics library
Hardware acceleration
Image processing Compression
Compression
Parallel computing
SIMT
Vector processor
Video coding Codec
Codec
VLIW
v
t
e
Distributed computing
Parallel computing
Parallel algorithm
Massively parallel

Cloud computing
High-performance computing
Multiprocessing
Manycore processor
GPGPU
Computer network
Systolic array
Bit
Instruction
Thread
Task
Data
Memory
Loop
Pipeline
Temporal
Simultaneous (SMT)
Simultaneous and heterogenous
Speculative (SpMT)
Preemptive
Cooperative
Clustered multi-thread (CMT)
Hardware scout
PRAM model
PEM model
Analysis of parallel algorithms
Amdahl's law
Gustafson's law
Cost efficiency
Karp–Flatt metric
Slowdown
Speedup
Process
Thread
Fiber
Instruction window
Array
Multiprocessing

Memory coherence

Cache coherence

Cache invalidation

Barrier

Synchronization

Application checkpointing

Stream processing

Dataflow programming

Models Implicit parallelism Explicit parallelism Concurrency

Implicit parallelism

Explicit parallelism

Concurrency

Non-blocking algorithm

Flynn's taxonomy SISD SIMD Array processing (SIMT) Pipelined processing Associative processing MISD MIMD

SISD

SIMD Array processing (SIMT) Pipelined processing Associative processing

Array processing (SIMT)

Pipelined processing

Associative processing

MISD

MIMD

Dataflow architecture

Pipelined processor

Superscalar processor

Vector processor

Multiprocessor symmetric asymmetric

symmetric

asymmetric

Memory shared distributed distributed shared UMA NUMA COMA

shared

distributed

distributed shared

UMA

NUMA

COMA

Massively parallel computer

Computer cluster Beowulf cluster

Beowulf cluster
Grid computer
Hardware acceleration
Ateji PX
Boost
Chapel
HPX
Charm++
Cilk
Coarray Fortran
CUDA
Dryad
C++ AMP
Global Arrays
GPUOpen
MPI
OpenMP
OpenCL
OpenHMPP
OpenACC
Parallel Extensions
PVM
pthreads
RaftLib
ROCm
UPC
TBB
ZPL
Automatic parallelization
Deadlock
Deterministic algorithm
Embarrassingly parallel
Parallel slowdown
Race condition
Software lockout
Scalability
Starvation
Category: Parallel computing

v

t

e

Abstract machine

Stored-program computer

Finite-state machine with datapath Hierarchical Deterministic finite automaton Queue automaton Cellular automaton Quantum cellular automaton

with datapath

Hierarchical

Deterministic finite automaton

Queue automaton

Cellular automaton

Quantum cellular automaton

Turing machine Alternating Turing machine Universal Post–Turing Quantum Nondeterministic Turing machine Probabilistic Turing machine Hypercomputation Zeno machine

Alternating Turing machine

Universal

Post–Turing

Quantum

Nondeterministic Turing machine

Probabilistic Turing machine

Hypercomputation

Zeno machine

Belt machine

Stack machine

Register machines Counter Pointer Random-access Random-access stored program

Counter

Pointer

Random-access

Random-access stored program

Microarchitecture

Von Neumann

Harvard modified

modified

Dataflow

Transport-triggered

Cellular

Endianness

Memory access NUMA HUMA Load–store Register/memory

NUMA

HUMA

Load–store

Register/memory

Cache hierarchy

Memory hierarchy Virtual memory Secondary storage

Virtual memory

Secondary storage

Heterogeneous

Fabric

Multiprocessing

Cognitive

Neuromorphic

Orthogonal instruction set

CISC

RISC

Application-specific

EDGE TRIPS

TRIPS

VLIW EPIC

EPIC

MISC

OISC

NISC

ZISC

VISC architecture

Quantum computing

Comparison Addressing modes

Addressing modes

Motorola 68000 series

VAX

PDP-11

x86

ARM

Stanford MIPS

MIPS

MIPS-X

Power POWER PowerPC Power ISA

POWER

PowerPC

Power ISA

Clipper architecture

SPARC

SuperH

DEC Alpha

ETRAX CRIS

M32R

Unicore

Itanium

OpenRISC

RISC-V

MicroBlaze

LMC

System/3x0 S/360 S/370 S/390 z/Architecture

S/360

S/370

S/390

z/Architecture

Tilera ISA

VISC architecture

Epiphany architecture

Others

Pipeline stall

Operand forwarding

Classic RISC pipeline

Data dependency

Structural

Control

False sharing

Scoreboarding

Tomasulo's algorithm Reservation station Re-order buffer

Reservation station

Re-order buffer

Register renaming

Wide-issue

Branch prediction

Memory dependence prediction

Bit Bit-serial Word

Bit-serial

Word

Instruction

Pipelining Scalar Superscalar

Scalar

Superscalar

Task Thread Process

Thread

Process

Data Vector

Vector

Memory

Distributed

Temporal

Simultaneous Hyperthreading Simultaneous and heterogenous

Hyperthreading

Simultaneous and heterogenous

Speculative

Preemptive

Cooperative

SISD

SIMD Array processing (SIMT) Pipelined processing Associative processing SWAR

Array processing (SIMT)

Pipelined processing

Associative processing

SWAR

MISD

MIMD SPMD

SPMD

Transistor count

Instructions per cycle (IPC) Cycles per instruction (CPI)

Cycles per instruction (CPI)

Instructions per second (IPS)

Floating-point operations per second (FLOPS)

Transactions per second (TPS)

Synaptic updates per second (SUPS)

Performance per watt (PPW)
Cache performance metrics
Computer performance by orders of magnitude
Central processing unit (CPU)
Graphics processing unit (GPU) GPGPU
GPGPU
Vector
Barrel
Stream
Tile processor
Coprocessor
PAL
ASIC
FPGA
FPOA
CPLD
Multi-chip module (MCM)
System in a package (SiP)
Package on a package (PoP)
Embedded system
Microprocessor
Microcontroller
Mobile
Ultra-low-voltage
ASIP
Soft microprocessor
System on a chip (SoC)
Multiprocessor (MPSoC)
Cypress PSoC
Network on a chip (NoC)
Coprocessor
AI accelerator
Graphics processing unit (GPU)
Image processor
Vision processing unit (VPU)
Physics processing unit (PPU)
Digital signal processor (DSP)
Tensor Processing Unit (TPU)

Secure cryptoprocessor

Network processor

Baseband processor

1-bit

4-bit

8-bit

12-bit

15-bit

16-bit

24-bit

32-bit

48-bit

64-bit

128-bit

256-bit

512-bit

bit slicing

others variable

variable

Single-core

Multi-core

Manycore

Heterogeneous architecture

Core

Cache CPU cache Scratchpad memory Data cache Instruction cache replacement policies coherence

CPU cache

Scratchpad memory

Data cache

Instruction cache

replacement policies

coherence

Bus

Clock rate

Clock signal

FIFO

Arithmetic logic unit (ALU)

Address generation unit (AGU)

Floating-point unit (FPU)

Memory management unit (MMU) Load–store unit Translation lookaside buffer (TLB)

Load–store unit

Translation lookaside buffer (TLB)

Branch predictor

Branch target predictor

Integrated memory controller (IMC) Memory management unit

Memory management unit

Instruction decoder

Combinational

Sequential

Glue

Logic gate Quantum Array

Quantum

Array

Processor register

Status register

Stack register

Register file

Memory buffer

Memory address register

Program counter

Hardwired control unit

Instruction unit

Data buffer

Write buffer

Microcode ROM

Counter

Multiplexer

Demultiplexer

Adder

Multiplier CPU

CPU

Binary decoder Address decoder Sum-addressed decoder

Address decoder

Sum-addressed decoder

Barrel shifter

Integrated circuit 3D Mixed-signal Power management

3D

Mixed-signal

Power management

Boolean

Digital

Analog

Quantum

Switch

PMU

APM

ACPI

Dynamic frequency scaling

Dynamic voltage scaling

Clock gating

Performance per watt (PPW)

History of general-purpose CPUs

Microprocessor chronology

Processor design

Digital electronics

Hardware security module

Semiconductor device fabrication

Tick–tock model

Pin grid array

Chip carrier

v

t

e

Floating point

Numerical stability

System of linear equations

Matrix decompositions

Matrix multiplication (algorithms)

Matrix splitting

Sparse problems

CPU cache

TLB

Cache-oblivious algorithm

SIMD

Multiprocessing

ATLAS

MATLAB

Basic Linear Algebra Subprograms (BLAS)

LAPACK

Specialized libraries

General purpose software

Computer programming

Free and open-source software

Linux

Engineering

Electronics

Technology

Mathematics