Khronos Connects Software to Silicon

Open, royalty-free interoperability standards to harness the power of GPU, XR and multiprocessor hardware

3D graphics, augmented and virtual reality, parallel programming, inferencing and vision acceleration

Non-profit, member-driven standards organization, open to any company

Proven multi-company governance and Intellectual Property Framework

Founded in 2000

~ 200 Members | ~ 40% US, 30% Europe, 30% Asia
Khronos Compute Acceleration Standards

Higher-level Languages and APIs
Streamlined development and performance portability

Single source C++ programming with compute acceleration
Neural Network Exchange Format Trained Networks
Graph-based vision and inferencing acceleration
Third party vision, streaming and inferencing libraries

Applications, libraries, and higher-level languages and APIs can use lower-level Khronos standards to access hardware acceleration

Lower-level Languages and APIs
Explicit hardware control

Multiple programming abstractions to meet the needs of diverse software stack architectures

GPU rendering + compute acceleration
Intermediate Representation (IR) language compiler target supporting parallel execution and graphics
Heterogeneous compute acceleration

OpenCL Complements Vulkan
Not just GPU acceleration
Simpler programming model
Relatively lightweight run-time
More language flexibility, e.g., pointers
Rigorously defined numeric precision
Framework for connecting custom processors

CPU
GPU
FPGA
DSP
AI/Tensor HW
Custom Hardware

This work is licensed under a Creative Commons Attribution 4.0 International License
Apps, Libraries and Engines using OpenCL

The industry’s most pervasive, cross-vendor, open standard for low-level heterogeneous parallel programming


Desktop Creative Apps

Adobe
otley
acdsee
Modo
AUTODESK
CHAOGROUP
CyberLink
pcc
Radeon
RealFlow
LuxCoreRender
SYCL
OpenACC
PyOpenCL
OpenACC

Parallel Languages

Machine Learning Libraries and Frameworks

Intel
OPENVINO™
portDNN
Caffe
Acuity
MetaWare EV
TI DL Library (TIDL)

Molecular Modelling Libraries

CHARMM
FOLDING HOME
ForceBalance
OpenMM
GROMACS

Vision, Imaging and Video Libraries

Vision, Imaging and Video Libraries

Math and Physics Libraries

Math and Physics Libraries

Linear Algebra Libraries

OpenCL
CLBlast
CLBlast
OpenCL
OpenCL State-of-the Union

- **OpenCL 3.0 adoption is strong and growing**
  - 14 OpenCL 3.0 Adopters, second only to OpenCL 1.2 (Vulkan 1.3 has 13 Adopters)

- **Significant open-source activity**
  - Mesa Rusticl for Linux
  - clang/LLVM compilation front-ends
  - Layered implementations clspv and Ancle over Vulkan, OpenCLon12 over DX12

- **OpenCL is a popular substrate layer for higher-level models, especially SYCL**
  - The second most common offload path, after CUDA, but before SYCL, Vulkan, HIP

- **Emerging acceptance of OpenCL as compute layer over Vulkan**
  - Especially for ML, simpler programming model, more language flexibility, e.g., pointers
  - First conformant layered OpenCL 3.0 implementation

- **Regular (roughly) quarterly Releases with new unified specification format!**
  - 3.0.16 is released for IWOCL 2024 with External Memory and Semaphores finalized

- **Active extension pipeline - driven by mobile, embedded and desktop markets**
  - Recordable Command Buffers, Cooperative Matrix, Unified Shared Memory, YUV Images, Tiling Controls...
OpenCL 3.0 Adoption

Currently 14 OpenCL 3.0 Adopters, 9 already submitted conformant products - second only to OpenCL 1.2

https://www.khronos.org/conformance/adopters/conformant-products/opencl

Adopters of previous OpenCL Versions

Shipping OpenCL 3.0 Conformant Implementations
OpenCL Open-Source Project Momentum

# OpenCL-based GitHub Repos

X5 increase in the past five years

OpenCL has broken the 25K project barrier as of March 2024

This work is licensed under a Creative Commons Attribution 4.0 International License
OpenCL on GPUInfo.org

The online GPUinfo.org database is populated using the OpenCL Hardware Capability Viewer application. Available for Windows, Linux, and Android. Reads and displays OpenCL information and uploads to the database. Please download and run to help populate the database!

Two years since OpenCL added to the GPUInfo.org website. 1000 additional reports in the last 6 months.
OpenCL Deployment Flexibility

- **OpenCL C**
- **C++ for OpenCL**
- **SYCL**
- **DPC++**
- **AdaptiveCpp**

- **glslang**
- **DXC**
- **HIP/CUDA**
- **ChipStar**

- **MLIR**
- **IREE**

- **SPIR-V LLVM IR Translator**

- **SPIR-V Tools** (Dis)Assembler, Validator, Optimize/Remap, Fuzzer, Reducer

- **OpenCL On12 Inc. Mesa SPIR-V to DXIL**

- **Intel Level Zero**

- **OpenCL C on 3rd Party Open Source**

- **Language Definitions**

- **khronos.org**
# API Layering

<table>
<thead>
<tr>
<th>Layers Over</th>
<th>Vulkan</th>
<th>OpenGL</th>
<th>OpenCL</th>
<th>OpenGL ES</th>
<th>DX12</th>
<th>DX9-11</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vulkan</td>
<td></td>
<td></td>
<td>Zink</td>
<td>clspv + clvk Ancle RustiCL/Zink</td>
<td>GLOVE Angle</td>
<td>vkd3d-Proton vkd3d</td>
</tr>
<tr>
<td>OpenGL</td>
<td>gfx-rs Ashes</td>
<td></td>
<td></td>
<td>Angle</td>
<td></td>
<td>WineD3D</td>
</tr>
<tr>
<td>DX12</td>
<td>Dozen gfx-rs</td>
<td>Microsoft ‘GLOn12’</td>
<td>Microsoft ‘CLOn12’</td>
<td></td>
<td></td>
<td>Microsoft D3D11On12</td>
</tr>
<tr>
<td>DX9-11</td>
<td>gfx-rs Ashes</td>
<td></td>
<td></td>
<td>Angle</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Metal</td>
<td>MoltenVK gfx-rs</td>
<td></td>
<td></td>
<td>MoltenGL Angle</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**COLUMNS:** Benefit ISVs by making an API available everywhere

**ROWS:** Benefit Platforms by adding APIs

Enabled by growing robustness of open-source compiler ecosystem using SPIR-V
Layered OpenCL Implementations

**clspv + clvk**  
OpenCL over Vulkan  
Google  
- clspv open-source OpenCL kernel to Vulkan SPIR-V compiler - tracks top-of-tree LLVM and Clang - not a fork  
- clvk - prototype open-source OpenCL to Vulkan run-time API translator  
- Used by shipping apps and engines on Android e.g., Adobe Premiere Rush video editor - 200K lines of OpenCL C kernel code

**clspv + Ancle**  
OpenCL over Vulkan  
Samsung  
- Integrates clspv and OpenCL runtime into Angle code base  
- Samsung Motivation  
  “OpenCL is widely used and deployed and is making a comeback thanks to ML”  
  “OpenCL is a favored high-level (front-end) compute language! Easier to write than Vulkan”  
- Ancle makes OpenCL a first-class citizen in Android by relying on Vulkan as its Native Driver

**Rusticl over Zink**  
OpenCL over Vulkan  
Mesa  
- The Zink Gallium driver emits Vulkan API calls and now supports OpenCL Kernels

**OpenCLon12**  
OpenCL over DX12  
Microsoft  
- GPU-accelerated OpenCL on any DX12 PC and Cloud instance (x86 or Arm)
# OpenCL Acceleration in Many ML Stacks

## Machine Learning Compilers

<table>
<thead>
<tr>
<th>Import Formats</th>
<th>Front-end / IR</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Caffe, Keras, MXNet, ONNX</td>
<td>NNVM / Relay IR</td>
<td>OpenCL, LLVM, CUDA, Metal</td>
</tr>
<tr>
<td>TensorFlow Graph, MXNet, PaddlePaddle, Keras, ONNX</td>
<td>nGraph / Stripe IR</td>
<td>OpenCL, LLVM, CUDA</td>
</tr>
<tr>
<td>PyTorch, ONNX</td>
<td>Glow Core / Glow IR</td>
<td>OpenCL, LLVM</td>
</tr>
<tr>
<td>TensorFlow Graph, PyTorch, ONNX</td>
<td>XLA HLO (MLIR)</td>
<td>OpenCL</td>
</tr>
</tbody>
</table>

## Common Steps

1. **Import Trained Network Description**
2. **Graph-level optimizations** e.g., node fusion, node lowering and memory tiling
3. **Decompose to primitive instructions and emit programs for accelerated run-times**

## Additional Machine Learning Compilers and Frameworks using OpenCL Acceleration

<table>
<thead>
<tr>
<th>Inferencing Libraries and Frameworks</th>
<th>Google TensorFlow and NNAPI</th>
<th>Embedded NN Compilers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alibaba MNN</td>
<td>portDNN</td>
<td>CEVA Deep Neural Network (CDNN)</td>
</tr>
<tr>
<td>Arm Compute Library</td>
<td>Synopsis MetaWare EV</td>
<td>Cadence Xtensa</td>
</tr>
<tr>
<td>Baidu PaddlePaddle/Paddle-Lite</td>
<td>Texas Instruments DL Library (TIDL)</td>
<td>Neural Network Compiler (XNNC)</td>
</tr>
<tr>
<td>Berkeley Caffe</td>
<td>VeriSilicon Acuity</td>
<td></td>
</tr>
<tr>
<td>Intel cuDNN and OpenVINO</td>
<td>Xiaomi Mace</td>
<td></td>
</tr>
</tbody>
</table>

This work is licensed under a Creative Commons Attribution 4.0 International License © The Khronos® Group Inc. 2024 - Page 12
OpenCL Specification Releases and Roadmap

**OpenCL 3.0.16 shipped on April 4th, 2024**
Continues the regular release cadence for new functionality and bug fixes
External memory objects and semaphores for external sharing and Interop finalized
Kernel Clock extension provisional release

**OpenCL Extension Pipeline**
Provisional, EXT and Vendor extensions - candidates for final ratification
We are listening to your input!

- Support C++ for OpenCL (EXT)
- Command Buffer Record/Replay (provisional)
- Unified Shared Memory
- Floating Point Atomics (EXT)
- Required Subgroup Size
- Generalized Image from buffer
- Image Tiling Controls
- YUV Multi-planar Images
- Cross-workgroup Barriers
- Cooperative Matrices
- Timeline Semaphores
- 32 and 64-length vectors
- Indirect Dispatch
- ML Operations
OpenCL SDK Upgrades

Open-source OpenCL SDK includes all components to develop OpenCL applications
- OpenCL Headers (include/api)
- OpenCL C++ bindings (include/cpp)
- OpenCL Utility Libraries (include/utils)
- Build system and CI

Documentation and Sample Code
- OpenCL Guide
- Code samples (samples/)
- Documentation (docs/)

Loader and Layers
- SDK and Layers Tutorial

Khronos funds SDK upgrades
Community contributions also welcome!

Spring 2022 SDK Updates
More details in the SDK Blog

- Enhanced Cmake-based build system
- Subprojects and components
- Binary releases
- Tagged SDK versions

- Enhanced SDK documentation
  In OpenCL Guide

- OpenCL 3.0 Samples
  C, C++, Python and Ruby

- Utility Libraries
  For loading kernel source and binary files

Coming Soon!

- Upstream to Kitware’s FindOpenCL.cmake
  Enhances OpenCL:: namespace

- Packaging and Distribution Support
  Build packages from the SDK
  Package newer versions of OpenCL
  Ease cross-platform installation, including PPAs

- Enhanced SDK Validation Layers
  Object lifetime, Input parameters, SPIR-V
Discussion Topics

- How can we reduce desktop fragmentation
  - Need of universal SPIR-V ingest
  - Promote the idea of SPIR-V ingestion front-end to LLVM?
  - Leverage Microsoft’s SPIR-V in LLVM?
  - Layered implementations may help?

- Provide more support and encouragement for layered OpenCL implementations?
  - Clspv/Ancle, Microsoft OpenCLon12, Rusticl/Zink
  - Does Rusticl over Zink on MoltenVK work for OpenCL on Apple?
  - OpenCL on Pi - maybe through Rusticl over Zink/Vulkan?

- How encourage Tensorflow and PyTorch direct support for OpenCL (not just TensorFlow Lite)
  - Increased investment in TVM as an open source path to other stacks?
  - Strengthen operations for ML: coop matrix, Subgroup requirements for wavefront/warp size, Built-in Kernels?

- How increase effectiveness as target layer e.g., for SYCL and OpenMP
  - Approach OpenMP for backend cooperation once we have SPIR-V backend in LLVM?

- Market demand for OpenCL Safety Critical Profile?
  - OpenCL IS already being deployed in SC markets
  - Backend for SYCL SC?
OpenCL Advisory Panel

Working Group makes decisions on standards evolution

**Working Group**
- Khronos Members
  - Any company can join.
  - Membership Fee.
  - Covered by NDA and IP Framework

**Shared Email list and Repository**
- Hosted by Khronos.
  - Under Khronos NDA
- Working groups can share draft specifications and accept detailed design contributions as Panel Members are covered by IP Framework

**Advisory Panel**
- Panel Members
  - Invited industry experts.
  - $0 Cost.
  - Covered by NDA and IP Framework

Regular meetings to give feedback on roadmap and draft specifications
Please reach out to opencl-chair@lists.khronos.org if you wish to apply
Developers - Please Give Us Feedback!

- Give us your feedback on the OpenCL spec GitHub
  - What could be added to the OpenCL ecosystem to make you more productive?
  - What API and Language features do you most need?
  - [https://github.com/KhronosGroup/OpenCL-Docs](https://github.com/KhronosGroup/OpenCL-Docs)

- Please download and run the GPUinfo OpenCL Hardware Capability Viewer
  - [https://opencl.gpuinfo.org/download.php](https://opencl.gpuinfo.org/download.php)

- Consider applying to join the OpenCL Advisory Panel!
  - Email [opencl-chair@lists.khronos.org](mailto:opencl-chair@lists.khronos.org)
OpenCL Resources

- OpenCL Home Page
  - https://www.khronos.org/opencl/
- OpenCL Registry for OpenCL core and extension specifications
  - https://www.khronos.org/registry/OpenCL/
- C++ for OpenCL Documentation
  - https://github.com/KhronosGroup/Khronosdotorg/blob/master/api/opencl/assets/CXX_for_OpenCL.pdf
- OpenCL SDK
  - https://github.com/KhronosGroup/OpenCL-SDK
- OpenCL Guide
  - https://github.com/KhronosGroup/OpenCL-Guide
- OpenCL Specification Source
  - https://github.com/KhronosGroup/OpenCL-Docs
- OpenCL Conformant Products
  - https://www.khronos.org/conformance/adopters/conformant-products/opencl
- GPUinfo.org Hardware Database
  - https://www.gpuinfo.org/
- Layered OpenCL implementations - clspv/clvk and OpenCLon12
  - https://github.com/google/clspv
  - https://github.com/kpet/clvk
  - https://github.com/microsoft/OpenCLon12