SYCL State of the Union
IWOCL'24

CC-BY
April 2024

SYCL so far

C++11	C++14		C++17	C++20	C++23

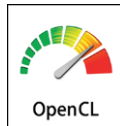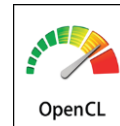|	SYCL 1.2 C++11 Single source programming		SYCL 1.2.1 C++11 Single source programming	SYCL 2020 C++17 Single source programming Lots of features for HPC Many backend options	SYCL Next ...

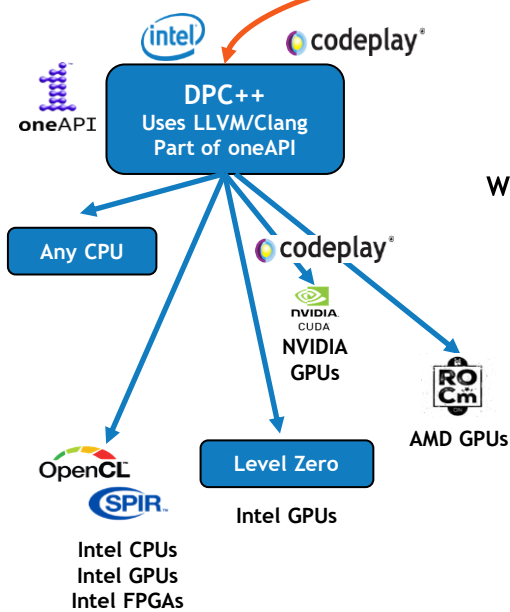| OpenCL 1.2 OpenCL C Kernel Language	OpenCL 2.1 SPIR-V in Core		OpenCL 2.2	OpenCL 3.0	
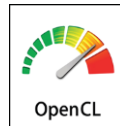
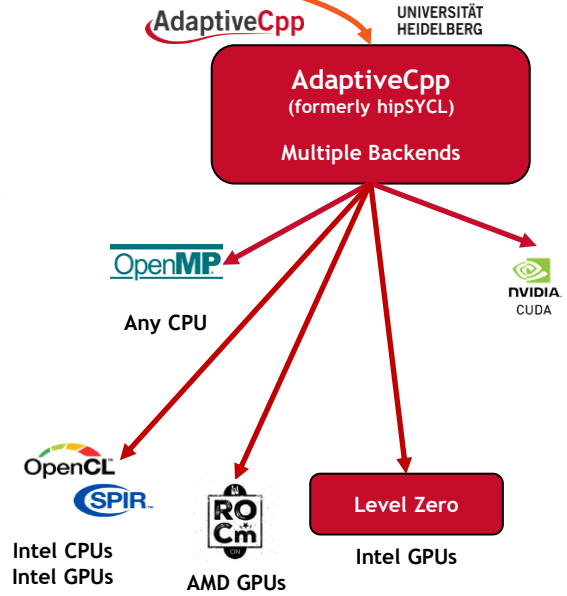| 2011	2015	2017	2020	202X

SYCL Implementations in Development

SYCL, OpenCL and SPIR-V, as open industry standards, enable flexible integration and deployment of multiple acceleration technologies

SYCL Source Code

SYCL enables Khronos to influence ISO C++ to (eventually) support heterogeneous compute

ISO

oneAPI

intel codeplay

DPC++
Uses LLVM/Clang
Part of oneAPI

Any CPU

codeplay

NVIDIA CUDA

NVIDIA GPUs

ROCm

Level Zero

AMD GPUs

OpenCL

SPIR

Intel GPUs

Intel CPUs
Intel GPUs
Intel FPGAs

WIP to upstream SYCL support in LLVM

AdaptiveCpp   UNIVERSITÄT HEIDELBERG

AdaptiveCpp
(formerly hipSYCL)

Multiple Backends

OpenMP

NVIDIA CUDA

Any CPU

OpenCL

SPIR

ROCm

Level Zero

Intel CPUs
Intel GPUs

AMD GPUs

Intel GPUs

Logos are the property of their respective owners

Intel® oneAPI DPC++/C++ Compiler *Conformant with SYCL 2020 Specification*

Unified Shared Memory, Parallel Reductions, Work Group Algorithms, Class Template Argument Deductions, Simplification of Accessors, Expanded Interoperability, and more

An Industry First:
SYCL 2020 Conformance on CPU and GPU

Open industry initiative driving a vendor-neutral software ecosystem for multiarchitecture accelerated computing. **Governed by the Linux Foundation.**

oneAPI > UXL Unified Acceleration Foundation

Intel is proud to contribute to a revolution anchored in SYCL:

An open ecosystem of
- software developers
- hardware vendors
- **compilers and development tools**
- APIs and specifications

Intel® oneAPI DPC++/C++ Compiler:
with SYCL towards Open Multiarchitecture Computing

Middleware and Frameworks

PyTorch TensorFlow scikit-learn OpenVINO NumPy XGBoost

oneAPI Industry Specification

Direct Programming   API-Based Programming

SYCL

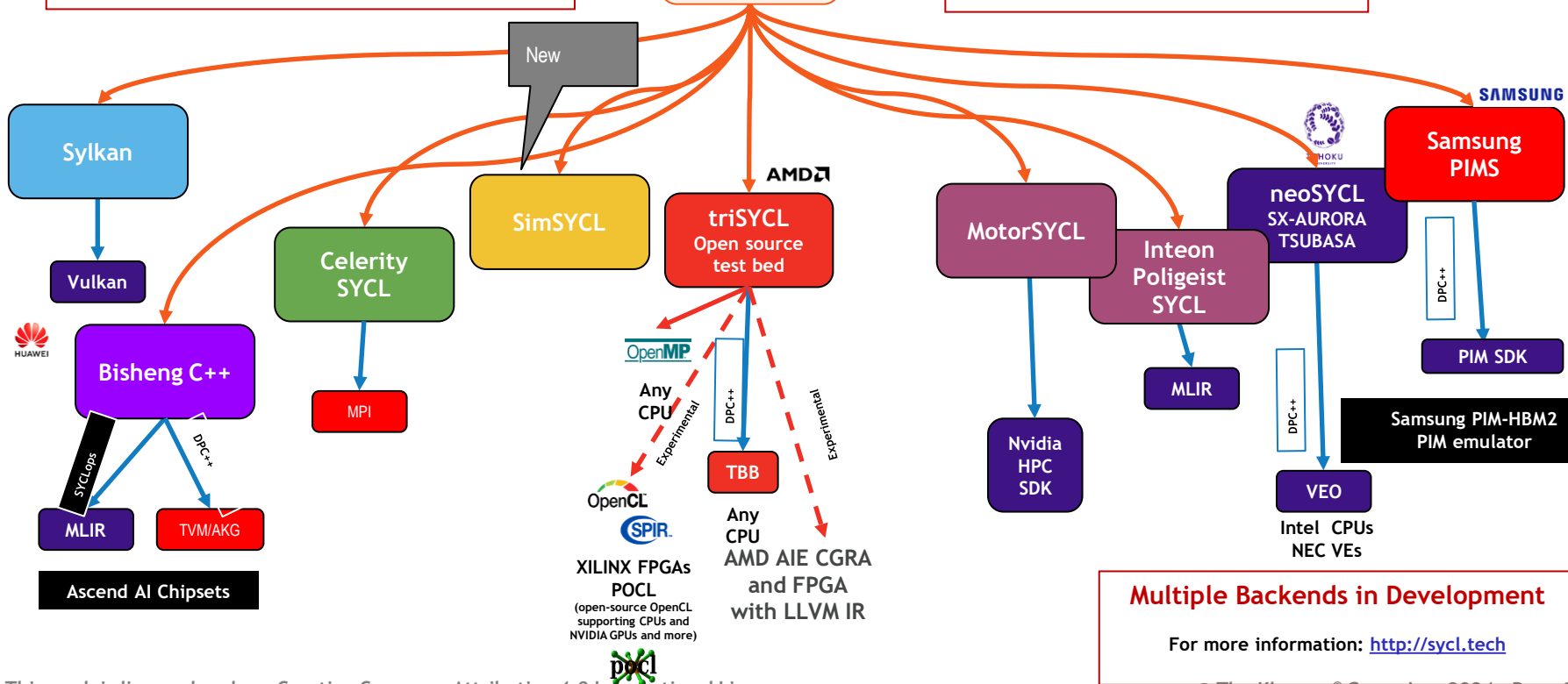| Math oneMKL | Threading oneTBB | Parallel STL oneDPL |
| Analytics/ ML oneDAL | DNN oneDNN | ML Comm oneCCL |

Low-Level Hardware Interface (oneAPI Level Zero)

CPU   GPU   FPGA   Other Accelerators

SYCL Experimental Development

SYCL, OpenCL and SPIR-V, as open industry standards, enable flexible integration and deployment of multiple acceleration technologies

SYCL Source Code

SYCL enables Khronos to influence ISO C++ to (eventually) support heterogeneous compute

New

Sylkan → Vulkan

Bisheng C++ (HUAWEI)
- SYCLops → MLIR
- DPC++ → TVM/AKG
- **Ascend AI Chipsets**

Celerity SYCL → MPI

SimSYCL

triSYCL — Open source test bed (AMD)
- OpenMP → Any CPU
- OpenCL / SPIR → Experimental → XILINX FPGAs POCL (open-source OpenCL supporting CPUs and NVIDIA GPUs and more) — pocl
- DPC++ → TBB → Any CPU
- Experimental → AMD AIE CGRA and FPGA with LLVM IR

MotorSYCL → Nvidia HPC SDK

Inteon Poligeist SYCL → MLIR

neoSYCL SX-AURORA TSUBASA (TOHOKU UNIVERSITY)
- DPC++ → VEO → Intel CPUs NEC VEs

Samsung PIMS (SAMSUNG)
- DPC++ → PIM SDK
- **Samsung PIM-HBM2 PIM emulator**

Multiple Backends in Development

For more information: http://sycl.tech

ISO

SYCL Next

- **Strategic path to incrementally release new features as KHR extensions**
  - Complete with <u>tests</u> and <u>implementations</u>
- **Key priorities are:**
  - syntax improvements
  - queue event performance
  - task graphs
  - compile-time properties
  - hierarchical parallelism
- **Seeking feedback on priority features from you!**

SYCL Next

Vendor extensions

KHR extensions

SYCL 2020

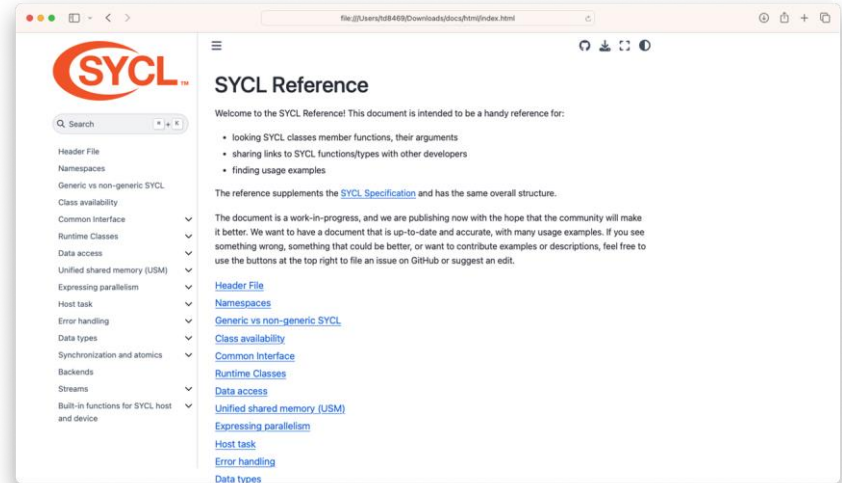Deprecated features
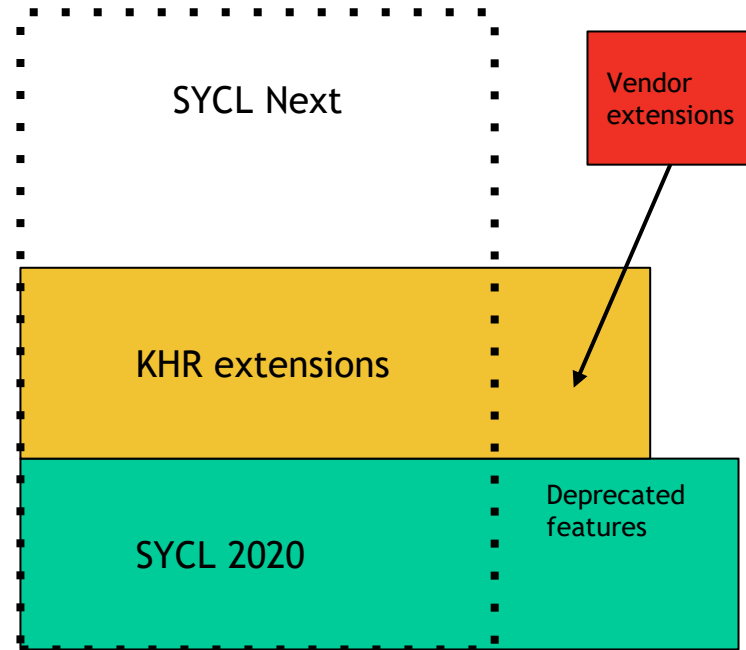
SYCL Reference

New resource to support SYCL developers

Inspired by cppreference.com

Short descriptions of SYCL 2020 API
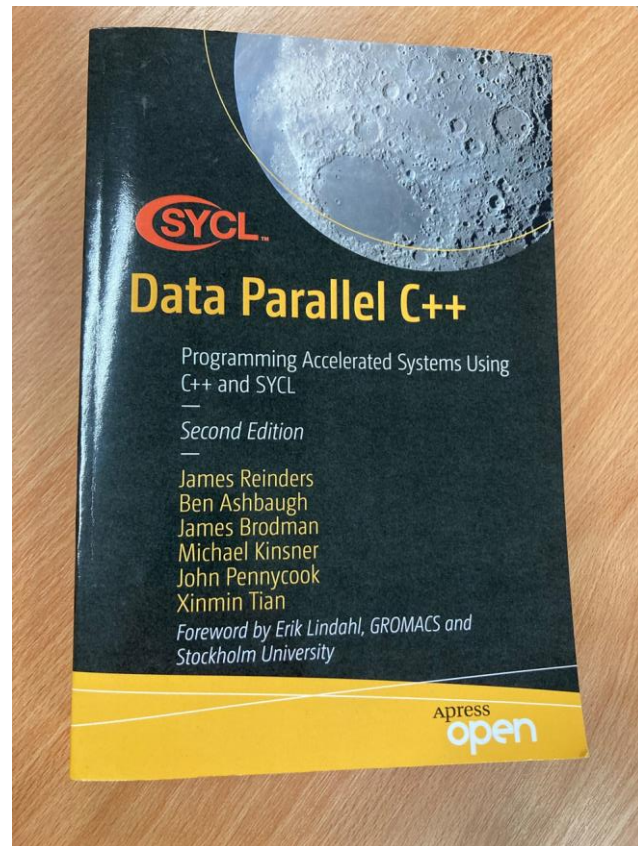
Specification remains the canonical document

https://www.khronos.org/sycl/reference

The SYCL Book (second edition)

https://link.springer.com/book/10.100
7/978-1-4842-9691-2

- New edition up to date with SYCL 2020, published Oct 4th, 2023
- Source code repository of examples
- (Free) Open Access online, or available in paperback

SYCL Developer Resources

- **I need to learn SYCL**
  - The book
  - Attend a tutorial
  - SYCL Academy: https://github.com/codeplaysoftware/syclacademy
- **I know SYCL, and need more information about an API**
  - SYCL Reference https://www.khronos.org/sycl/reference
- **I need to know the ins-and-outs of an API**
  - SYCL Spec (it's quite readable!) https://registry.khronos.org/SYCL/
- **I still need help!**
  - Forums:
    - https://community.khronos.org/c/sycl/
    - https://stackoverflow.com/questions/tagged/sycl
  - SYCL.tech: https://sycl.tech/
  - Khronos Discord: https://www.khr.io/khrdiscord
  - Ask your implementor

Get involved!

Public contributions to Specification and Conformance Tests
https://github.com/KhronosGroup/SYCL-CTS
https://github.com/KhronosGroup/SYCL-Docs

Khronos SYCL Forums, Discord/Slack Channels, Stack Overflow, and SYCL.tech

Khronos GitHub
Contribute to SYCL open source specs, CTS, tools and ecosystem

SYCL Advisory Panel

Join as an Invited Expert (no cost, sign Khronos NDA)
https://www.khronos.org/advisors/

SYCL Working Group

Join as a Khronos members
https://www.khronos.org/members/
https://www.khronos.org/registry/SYCL/