# Data Flow Pipes: A SYCL Extension for Spatial Architectures

Michael Kinsner
Intel Corporation
michael.kinsner@intel.com

John Freeman
Intel Corporation
john.freeman@intel.com

***Abstract*—FIFOs are a common construct in design for spatial and data flow architectures. OpenCL 2.0 defined a "pipe" feature to expose the FIFO construct, but the design didn't meet all needs of spatial architectures. This talk describes a pipes extension to the Khronos SYCL single-source, C++-based programming framework, that exposes a pipe abstraction which closes the gaps in the OpenCL design, while also offering a more usable interface. The C++ type system is leveraged to provide static connectivity guarantees without extensive compiler implementation effort, and to provide well-defined interaction with C++ features. The described extension provides a usable interface that can also act as a substrate for additional abstractions to be built on top. This talk will motivate the utility of FIFOs/pipes in high level language FPGA design, describe the SYCL pipes extension and its mapping to SPIR-V and OpenCL, and provide examples of use in common spatial design patterns.**

***Keywords—FPGA, pipes, SYCL, data flow, spatial***

## I. Extended Abstract

FPGAs can be classified as a spatial compute architecture, in which resources on distinct regions of a device perform compute simultaneously (same clock cycle) for different components or elements of an algorithm decomposition. This is a distinct conceptual model from Instruction Set Architecture (ISA) machines, and exposes a different set of opportunities and challenges when formulating efficient algorithms.

A fundamental construct in spatial decomposition and expression of many algorithms is a lightweight communication mechanism between regions of the device, which can simultaneously carry both data and control (synchronization) information. One such construct is a First In, First Out (FIFO) primitive, which is used extensively when describing high performance designs in hardware description languages such as Verilog. FIFOs provide both semantic isolation between regions of a spatial design, such as rate decoupling and buffer capacity to create a flexible data path that is more tolerant of dynamic memory access stalls, but also a practical isolation that can reduce complexity for the Computer Aided Design (CAD) algorithms that translate developer intent into an FPGA programming bitstream. As such, FIFO constructs have significant utility when expressing spatial algorithm decompositions in high level languages, and have been exposed through the pipes feature of OpenCL 2.0 [1].

The OpenCL 2.0 pipes feature was designed to be implementable across many classes of devices, including ISA-based vector machines, and isn't the ideal expression of FIFOs for FPGAs. Features such as dynamic pipe connectivity (determined at kernel launch time) and reservations have resulted in low adoption by FPGA vendors. The SYCL programming language provides a new opportunity to define a solution that fills the gaps in the OpenCL pipe model from an FPGA perspective.

SYCL is a single source, C++-based offload accelerator programming framework from the Khronos group [2], that sits at an abstraction level above that of OpenCL. It has been designed to be implementable on top of existing OpenCL implementations, while providing a higher level and more usable interface from a developer's perspective.

This talk will motivate the utility of pipe constructs in high performance spatial designs, and will then describe an extension that adds pipes to the SYCL language. This extension was developed by Intel, but is implemented in open source and is a proof of concept that aims to become the industry standard expression of pipes within SYCL. The talk will detail: (1) the mechanisms and syntax to express inter-kernel, intra-kernel, host ↔ device, and host ↔ I/O pipe communication in SYCL, (2) the novel type-based approach that uses the C++ type system to guarantee that a toolchain can infer compile-time static pipe connectivity between kernels without compiler modification, and to guarantee well defined interactions with C++ features, (3) mapping of the extension to SPIR-V and OpenCL-based implementations, (4) the principle that other pipe abstractions, such as an instance-based interface, can be built on top of the described extension, and (5) examples of the extension used to describe common patterns in FPGA high-level design.

FIFOs are a fundamental construct in spatial and data flow architectures, and as presented in this talk, are being exposed in modern high level programming models. With the intent of becoming the industry-accepted mechanism to express data flow FIFOs in SYCL, this talk aims to describe and communicate key aspects of the proposed extension, but also to garner feedback from academia and potential users of the extension, as it evolves into a final form.

## II. References

[1] The Khronos OpenCL Working Group, "The OpenCL 2.0 Specification," 2015, https://www.khronos.org/registry/OpenCL/specs/opencl-2.0.pdf.

[2] The Khronos SYCL Working Group, "The SYCL 1.2.1 Specification," 2019, https://www.khronos.org/registry/SYCL/specs/sycl-1.2.1.pdf.

## III. AUTHOR BRIEF BIOS (MAX 300 WORDS EACH)

Michael Kinsner holds a PhD in computer engineering, and has worked at Altera and then Intel on FPGA high level design programming languages and compilers, for eight years. Michael is an Intel FPGA representative within the Khronos group, and participates in development of the OpenCL and SYCL standards.

John Freeman has worked at Altera and then Intel on FPGA synthesis and high level design compilers for ten years. He currently manages the Intel FPGA high level design compiler team.