RISC-V EUROPE EXTENDED ABSTRACT

Enabling High Performance RISC-V Software for AI in the Real World

Alastair Murray, Architect, Codeplay Software

Jeremy Bennett, CEO, Embecosm

Abstract

The first part of this talk presents SYCL from the Khronos Group, an open, cross-platform abstraction layer parallel to C++ which enables heterogeneous computing. Alongside this we present oneAPI, a set of heterogeneous libraries for performance across a variety of AI workloads, and recently contributed to the Unified Acceleration(UXL) Foundation. The UXL Foundation is a cross-industry effort led by Arm, Broadcom, Fujitsu, GE Healthcare, Google Cloud, Imagination, Intel, Samsung and Qualcomm.

The second part of this talk presents a real-world example of bringing up an AI system (PyTorch) on a RISC-V based accelerator. The example is entirely open source and uses a widely available FPGA board on which to run both the host and the RISC-V based accelerator. It is available as an application note to help others achieve high performance with RISC-V based AI systems. We present data to show the performance of the system.

Introduction to SYCL, oneAPI and the UXL Foundation

Innovation with AI and HPC software demands increasing power and efficiency. To meet these demands, new and innovative hardware is being created - from CPUs to GPUs and new specialized AI accelerators, including with RISC-V. This diverse hardware is being deployed in a single system with work being split between GPUs and specialized AI processors, for example. We call this heterogeneous computing - and while it has enormous potential for performance, it comes with its own challenges that must be overcome.

RISC-V accelerators for AI and HPC are being developed across the world, with adoption of the open ISA being driven by this demand for new, innovative hardware. For continued success, it's important that any RISC-V hardware can be implemented alongside hardware from other vendors, and run software heterogeneously.

The foremost challenge is writing software that works across all available hardware in a heterogeneous system - with much hardware having vendor-specific code and optimizations that will not work on unsupported hardware. This means developers face undertaking significant development efforts to maintain separate code stacks, or become vendor-locked to specific hardware that may not best suit their use-case in future. Developers instead need a single way to write software for these processors, regardless of vendor, while also achieving the level of performance they require. In this joint presentation, we will outline the solution through open source, vendor-neutral and standards-based code.

SYCL is an open, cross-platform abstraction layer parallel to C++ which enables heterogeneous computing. It's designed to be as close to modern ISO C++ as possible for a minimal learning curve, and is maintained by the Khronos Group. Then, oneAPI provides a set of heterogeneous libraries for performance

across a variety of AI workloads, such as oneMath and oneDNN.

The oneAPI Construction Kit is a framework for enabling SYCL on your hardware. This allows developers to write their software application once and deploy it across new hardware such as RISC-V.

The oneAPI Construction Kit [1] has recently been contributed to the Unified Acceleration (UXL) Foundation. The UXL Foundation is a cross-industry effort led by Arm, Broadcom, Fujitsu, GE Healthcare, Google Cloud, Imagination, Intel, Samsung and Qualcomm - with many more companies and organizations having also joined. The Foundation's aim is to build an open, standards-based and vendor-neutral software ecosystem for HPC & AI and empower developers with single-source code for all available hardware. By bringing SYCL and oneAPI to new hardware, the oneAPI Construction Kit is crucial to this vision and we'll explore the role of the UXL Foundation in more detail during this presentation.

The Case Study

The case study was originally developed as a group student project for the final year of a masters course in electronic engineering and computer science. It has since been extended as a full "HOWTO" application note.

The target platform is a Xilinx Zynq-7010 FPGA [2]. This contains a dual-core Arm-A9 processor subsystem (PS) on which we host PyTorch [3], and a general programmable logic (PL) block on which we run a RISC-V softcore as accelerator.

We take the user through the initial architectural exploration, where a Linux PC was used as host, and an interposer was written to intercept key calls within the oneAPI run-time system. This allowed individual PyTorch operations to be handed off to the FPGA, with the Arm processor susbsystem initially acting purely as a TCP/IP server to pass actions and data to the FPGA RISC-V softcore. A simple Python emulation of the FPGA hardware was used to verify the interposer and the approach used.

We then look at how this was expanded using the oneAPI *Hardware Abstraction Layer* (HAL) to generate code for the target RISC-V softcore. At this point the entire AI system could be moved from the Linux PC to the Arm processor subsystem, with offload of key operations via the HAL to the RISC-V softcore.

Finally we show how we can now use this system to evaluate different software and hardware strategies, using Resnet-18, an 18 layer residual neural network [4] image classifier. We use selected RISC-V ISA extensions to provide optimized implementations of key PyTorch operations and measure the benefit that these extensions provide.

References

1. oneAPI Construction Kit GitHub, UXL Foundation. https://github.com/uxlfoundation/oneapi-construction-kit
2. AMD Zynq™ 7000 SoCs, www.amd.com/en/products/adaptive-socs-and-fpgas/soc/zynq-7000.html. Retrieved 7 Feb. 2025.
3. PyTorch. pytorch.org. Retrieved 7 Feb 2025.
4. He, Kaiming; Zhang, Xiangyu; Ren, Shaoqing; Sun, Jian (2016). Deep Residual Learning for Image Recognition (PDF). Conference on Computer Vision and Pattern Recognition. arXiv:1512.03385. doi:10.1109/CVPR.2016.90.