SYCLARA: An Open Hardware–Software Platform for Evaluating SYCL Applications on RISC-V Vector Accelerators

Mojtaba Rostami Bilandi
EURECOM
Biot, France
mojtaba.rostamibilandi@eurecom.fr

Ivan Donchev Kabadzhov
EURECOM
Biot, France
ivan.Kabadzhov@eurecom.fr

Raja Appuswamy
EURECOM
Biot, France
raja.appuswamy@eurecom.fr

Abstract

Over the past few years, RISC-V has emerged as an open and extensible architecture with flexible vector extensions (RVV) for scaling computationally-intensive workloads. On the software side, SYCL has emerged as a standardized, cross-architecture programming model that can provide performance portability across several accelerators. In this work, we describe our efforts to bring these two standards together by extending the integration of CVA6 RISC-V core with the ARA2 RVV implementation to enable SYCL kernel offload via the oneAPI Construction Kit.

CCS Concepts

• **Computer systems organization → Reduced instruction set computing**; **Reconfigurable computing**; • **Computing methodologies → Parallel programming languages**.

Keywords

RISCV, RVV, FPGA, SYCL, OCK

ACM Reference Format:
Mojtaba Rostami Bilandi, Ivan Donchev Kabadzhov, and Raja Appuswamy. 2025. SYCLARA: An Open Hardware–Software Platform for Evaluating SYCL Applications on RISC-V Vector Accelerators. In *International Workshop on OpenCL and SYCL (IWOCL '25), April 07–11, 2025, Heidelberg, Germany.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3731125.3731136

1 Introduction

Modern analytics and AI workloads are incredibly diverse and consist of a range of scalar, vector, and matrix computations. The traditional workhorse of the computing industry, the general-purpose CPU, cannot be optimized to meet such diverse requirements. Thus, there has been a growing interest in developing open, standards-based, cross-vendor approach to hardware acceleration. This interest has predominantly been driven by the rise of two key standards. On the hardware side, the open nature of RISC-V instruction set architecture (ISA) [6] has spurred the development of customized RISC-V accelerators for various domains based on advanced architectural features such as the RISC-V Vector Extension (RVV). On the software side, SYCL has emerged as a leading open standard for heterogeneous programming. Combining SYCL with RISC-V vector accelerators presents an exciting opportunity for developing fully-open, standards-based, hardware–software stack for accelerating AI and analytics. However, prior work has predominantly focused on using simulation or emulation-based infrastructure for evaluation of SYCL applications due to a lack of availability of performant, standards-compliant RVV implementations [5].

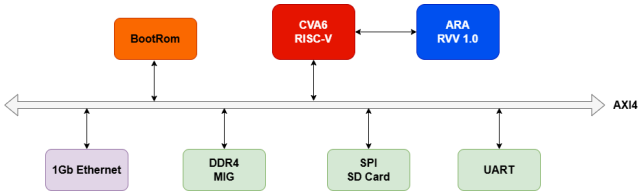

Figure 1: Linux-Capable SoC with RVV Support

In this work, we bridge this gap by developing SYCLARA[1]–an end-to-end, hardware–software platform that can be used as a testbed to evaluate SYCL and RISC-V implementations together. In particular, we build upon the recent integration of the CVA6 RISC-V core [7] with ARA2 [1] and extend it by adding Ethernet capability and enabling Linux boot from an SD card to bring up an RVV accelerator on the VCU118 platform. Using the oneAPI Construction Kit (OCK) with a customized, remote Hardware Abstraction Layer (HAL), we enable offloading of SYCL kernels to the accelerator. Using our end-to-end platform, we execute several SYCL applications and demonstrate that RVV can significantly improve execution times for computational tasks, with varying vectorization factors yielding optimal performance for different applications.

2 SYCLARA Platform

Figure 1 shows a high-level overview of the architecture of the SoC we have adapted and brought up. It is based on the Cheshire SoC developed by the PULP platform and integrates CVA6 [7], an open-source, high performance RISC-V processor core that implements the RV64GC ISA, with ARA [1], an open-source implementation of the RVV. The first integration of ARA with CVA6 was not capable of running Linux applications because ARA lacks an MMU. To overcome this limitation, we adopted a patched version of ARA2 and CVA6 where the MMU was shared between CVA6 and ARA. We brought up this integration on VCU118 by further adapting the DDR4 memory controller and using an SD card to hold the Linux image.

In order to offload SYCL kernels to our hardware platform, we rely on DPC++ and the recently open-sourced OCK. In particular, we used the OCK remote HAL, which provides a server and an OpenCL client that can be run on any standard Linux installation,

[1] https://gitlab.eurecom.fr/rostamib/syclara

IWOCL '25, Heidelberg, Germany

ACM ISBN 979-8-4007-1360-6/25/04
https://doi.org/10.1145/3731125.3731136

Resource usage	LUT	BRAM	DSP
SOC	318884	154	132
CVA6	47114	60	27
ARA	129512	32	102
Ethernet	1652	10	0

Table 1: SoC resource usage

	Scalar		VF=4		VF=8		VF=16	
	Conv	Matmul	Conv	Matmul	Conv	Matmul	Conv	Matmul
int32	26775	139909	19276	40008	10918	23351	6322	145803
float	28975	141210	21542	44683	11796	25508	6897	145619
double	47499	187428	22943	51325	12881	29528	7571	191197

Table 2: Scalar/vector execution time (ms) of SYCL kernels.

and use socket connections to communicate with each other. This HAL relies on the accelerator being equipped with an Ethernet interface. Thus, we integrated an Ethernet module into the SoC to enable interaction with CVA6 and ARA from a SYCL host. In a study by Chaoqun Liang et al. [2], Gigabit Ethernet was implemented and integrated into the Cheshire SoC and tested on Genesys2 and VCU118. However, they used an external Ethernet adapter with an Ethernet PHY and an RGMII interface. In contrast, we used the Xilinx 1G/2.5G BASE-X PCS/PMA or SGMII core, which converts the GMII interface to SGMII, the interface supported by the Ethernet PHY chip on the VCU118. For the MAC layer, we used an existing MAC [2]. To overcome the clock domain crossing between the AXI interconnect and the Ethernet IP, a FIFO was used to bridge the AXI data. The Ethernet driver was adapted from the lowRISC Ethernet driver. We installed the driver and its dependencies on Linux using Buildroot, incorporating patch files for the CVA6-SDK and enable SSH support.

We cross-compiled the HAL server for execution on CVA6. We compiled the HAL client to run on a local x86-64 machine. The default HAL client setup (targeting RV64GC) does not generate any vector instructions. Thus, we configured the HAL client manually by enabling the vector extension in the HAL device during compilation. Additionally, the vectorization factor can be adjusted using the environment variable `CA_RISCV_VF`. This variable acts as a multiplier for vectorization levels, allowing control over the degree of vectorization.

3 Experimental Evaluation

Having described the SYCLARA hardware–software stack, in this section, we present our experimental evaluation. The SoC has been implemented on the VCU118 evaluation board, which features a Xilinx Virtex UltraScale+ FPGA. CVA6 and ARA were clocked to run at 50 MHz. A higher clock rate was tested, but we encountered timing violations in CVA6 and ARA. The ARA configuration includes 2 lanes, and for the experiments reported here, we set the vector length (VLEN) to 2048. Table 1 presents the implementation results for the SoC, CVA6, ARA and Ethernet, showcasing resource usage such as LUTs, BRAMs, and DSPs.

To evaluate the performance improvement of RVV compared to non-RVV, we tested two SYCL kernels: 1D Convolution (conv), where we convolve two arrays with 4k entries, and matrix multiplication (matmul) of $(150, 300) \times (300, 600)$ matrices, as shown in Table 2. We also executed other database workloads on SYCLARA [3], but we do not report it here due to lack of space. We executed the two kernels on CVA6 both in scalar mode and vector using ARA. Clearly, these results demonstrate that SYCL compiler and runtime toolchains are capable of exploiting RISC-V vector accelerators to improve performance, as the 1D convolution execution time improves up to 6.27×, and matrix multiplcation improves up to 6×. However, increasing the vectorization factor (CA_RISCV_VF) does not have a uniform impact across all applications. For instance, conv achieves the best performance with a VF of 16, but matmul does not improve beyond VF of 8. In ongoing work, we are analyzing SYCLARA to understand the root cause of these differences.

[2]https://github.com/alexforencich/verilog-ethernet

4 Conclusion

In this work, we provided an overview of our SYCLARA platform that builds on OCK to offload SYCL computations on the ARA2 RVV accelerator integrated with the CVA6 RISC-V CPU. Using SYCLARA, we presented a preliminary evaluation of a few kernels to demonstrate that SYCL compiler toolchains can exploit RVV to improve performance on real RISC-V hardware. Given the lack of RVV implementations, we believe that SYCLARA provides a valuable framework for furthering research on open, standards-based hardware acceleration of analytics and AI.

Acknowledgments

We thank Uwe Dolinksy and Colin Davidson for their support in setting up OCK, and Matteo Perotti for his support in bringing up ARA. This work was supported by the Horizon Europe Project "SYCLOPS", funded by the European Union HE Research and Innovation programme under grant agreement No. 10109287.

References

[1] Matheus Cavalcante, Fabian Schuiki, Florian Zaruba, Michael Schaffner, and Luca Benini. 2020. Ara: A 1-GHz+ Scalable and Energy-Efficient RISC-V Vector Processor With Multiprecision Floating-Point Support in 22-nm FD-SOI. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 28, 2 (2020), 530–543. doi:10.1109/TVLSI.2019.2950087

[2] Chaoqun Liang, Alessandro Ottaviano, Thomas Benz, Mattia Sinigaglia, Luca Benini, Angelo Garofalo, and Davide Rossi. 2024. A Gigabit, DMA-enhanced Open-Source Ethernet Controller for Mixed-Criticality Systems. In *Proceedings of the 21st ACM International Conference on Computing Frontiers: Workshops and Special Sessions*. 55–58.

[3] Eugenio Marinelli and Raja Appuswamy. 2021. XJoin: Portable, parallel hash join across diverse XPU architectures with oneAPI. In *Proceedings of the 17th International Workshop on Data Management on New Hardware* (Virtual Event, China) *(DAMON '21)*. Article 11, 5 pages. doi:10.1145/3465998.3466012

[4] Matteo Perotti, Matheus Cavalcante, Renzo Andri, Lukas Cavigelli, and Luca Benini. 2024. Ara2: Exploring Single-and Multi-Core Vector Processing with an Efficient RVV 1.0 Compliant Open-Source Processor. *IEEE Trans. Comput.* (2024).

[5] Muhammad Tanvir, Kumudha Narasimhan, Mehdi Goli, Ouadie El Farouki, Svetlozar Georgiev, and Isaac Ault. 2022. Towards performance portability of AI models using SYCL-DNN. In *Proceedings of the 10th International Workshop on OpenCL* (Bristol, United Kingdom, United Kingdom) *(IWOCL '22)*. Association for Computing Machinery, New York, NY, USA, Article 23, 3 pages. doi:10.1145/3529538.3529999

[6] Andrew Waterman, Krste Asanovic, and John Hauser. 2019. The RISC-V instruction set manual, volume II: Privileged architecture. *RISC-V Foundation* (2019), 1–4.

[7] F. Zaruba and L. Benini. 2019. The Cost of Application-Class Processing: Energy and Performance Analysis of a Linux-Ready 1.7-GHz 64-Bit RISC-V Core in 22-nm FDSOI Technology. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 27, 11 (Nov 2019), 2629–2640. doi:10.1109/TVLSI.2019.2926114