Towards a Base-Station-on-Chip: RISC-V Hardware Acceleration for wireless communication.

Javier Acevedo[1] and Frank H. P. Fitzek[1,2] *

[1]Deutsche Telekom Chair of Communication Networks, TU Dresden
[2]Centre for Tactile Internet with Human-in-the-Loop (CeTI)

Abstract

The evolution of 5G and the emergence of 6G wireless communication systems impose higher demands for computing capabilities and lower power consumption in the front-end and processing circuitry. Furthermore, the incorporation of Artificial Intelligence (AI)/Machine Learning (ML) in the Radio Access Network (RAN) introduces heightened computational needs and stringent low-latency requirements for both training and inference. The concept of a Base Station on Chip (BSoC) addresses those demands by consolidating of the signal processing, neural network computations and network management functions into a single chip. This new computing platform relies on a sophisticated hardware/software co-design to optimize performance, power efficiency, and scalability, enabling a compact, yet adaptable and intelligent base station solution for next-generation wireless networks. This research investigates the efficient implementation of conventional Channel Estimation (CE), massive Multiple Input Multiple Output (mMIMO), and beamforming kernels on a state-of-the-art RISC-V vector Digital Signal Processors (DSP) to capitalize on Data Level Parallelism (DLP). Moreover, it explores how RISC-V Vector Extensions (RVV) combined with custom instructions can effectively address the throughput and latency demands of LOW Physical Layer (PHY) kernels.

Introduction

The advent of the Open Radio Access Network (Open RAN) has transformed profoundly wireless cellular networks by decoupling hardware and software into modular components, which are interconnected via open interfaces like evolved Common Public Radio Interface (eCPRI). This disaggregation fosters innovation by enabling the virtualization of the RAN function on Commercial-Off-The-Shelf (COTS) servers, reducing reliance on proprietary vendor equipment. Nevertheless, COTS hardware is built on fixed-length Single Instruction Multiple Data (SIMD) architectures, which cannot be adapted to fulfill the computational demands of the LOW PHY signal processing algorithms. In contrast, the open-source RISC-V Instruction Set Architecture (ISA) supports customizable vector lengths, facilitating the design of specialized vector processors tailored for wireless communication tasks such as CE, beamforming, and mMIMO. The aforementioned kernels represent the main signal processing algorithms performed at the Open Radio Radio Unit (O-RU) of a Open RAN-compliant base station. Therefore, congregating the execution of those kernels into a single chip promotes new computing platforms for small cells: the BSoC. This work aims to investigate how RISC-V can be leveraged to develop hardware

accelerators that meet the stringent throughput, latency, and power requirements of next-generation 6th Generation Cellular Networks (6G) base stations.

LOW PHY processing in base stations involves computationally intensive operations, including massive matrix multiplications, matrix inversions, and Fast Fourier Transform (FFT)/iFFT computations, critical for real-time CE and mMIMO. The computational complexity of these algorithms scales significantly with the number of antenna elements and system configuration. The RISC-V Instruction Set Architecture (ISA) flexibility allows for customized solutions to address this complexity efficiently. In this study, we utilize the state-of-the-art RISC-V-based Ara processor from the PULP group to implement hardware accelerators for LOW PHY algorithms. Our objectives are two-fold:

- Assess speedup in kernel execution throughout data parallelization.
- Develop custom hardware modules optimized for each signal processing kernel.

Approach

The Ara processor is a high-performance, open-source, RISC-V core designed for parallel processing. LOW PHY are characterized for having multiple arithmetic operations, which can be vectorized and hence accelerated by distributing the computation over multiple parallel lanes. For CE, we implemented LSE and MMSE kernel to determine the channel matrix H, by
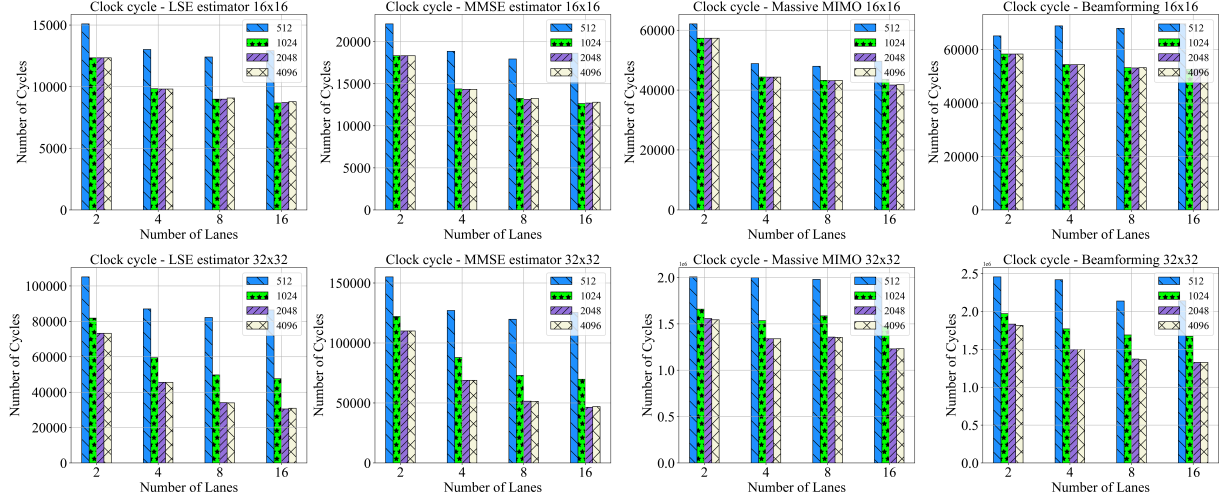
Figure 1: *Clock cycle counts for the execution of the Minimum Mean Square Error (MMSE) CE, Least-Squares (LSE) CE, mMIMO and beamforming algorithms employing different matrix sizes (16×16 and 32×32) and $VLEN$ values (512, 1024, 2048, and 4096 bit).*

solving the equation $H = YX^{-1}$ throughout the calculation of the least squares and statistics. On the other hand, we employed the Cooley-Tukey to compute the radix 4 FFT. In the case of mMIMO, we calculated the Zero-Forcing (ZF) precoder, W, by solving the equation given by $W = H^{\mathbf{H}}(HH^{\mathbf{H}})^{-1}$, where $\mathbf{H}$ represents the hermitian or conjugate transpose of the channel matrix H. Additionally, we have done an extension for the digital beamforming to construct the channel matrix including the steering vectors. In such a manner, we could represent the phase and amplitude, which are required to be applied to each antenna element to direct the beams precisely.

Preliminary results

In this work, we provide a C-based software implementation of the aforementioned wireless communication kernels. We simulated and evaluated each kernel by adjusting the number of lanes and measuring the number of clock cycles required by the Ara core to perform the computation. This approach allows us to observe how vectorization impacts the execution speedup of each kernel, consistent with the findings presented in [1, 2].

Our evaluation measures the clock cycle count of the LSE and MMSE CE, mMIMO, and beamforming. Figure 1 illustrates these results across various vector registers lengths, $VLEN$, and number of lanes. The Ara core supports 64-bit values, which dictates he number of elements processed in parallel within hardware lanes. The bigger the $VLEN$, the higher the number of elements employed for computation within a clock cycle. Matrix sizes are varied to demonstrate their effect on the number of parallel operations and,

consequently, the total clock cycles needed to complete each kernel's computation.

Conclusion and Outlook

In this study, we introduced the initial findings from the software implementation of multiple wireless communication kernels on a RISC-V vector processor. By leveraging the DLP inherent in these algorithms, we achieve a reduction in the clock cycle count as the number of parallel processing lanes increases. Future work will include a custom hardware implementation of some of those kernels and their integration into the Ara processor via AXI interfaces [3]. Hence, the development of tailored instructions to provide support to that hardware is also planned [4].

References

[1] Marco Bertuletti et al. "Efficient Parallelization of 5G-PUSCH on a Scalable RISC-V Many-Core Processor". In: *2023 Design, Automation Test in Europe Conference Exhibition (DATE)*. 2023, pp. 1–6. DOI: `10.23919/DATE56975. 2023.10137247`.

[2] Javier Acevedo, Frank H. P. Fitzek, and Patrick Seeling. "5G Channel Estimation Kernels on RISC-V Vector Digital Signal Processors". In: *2024 International Conference on Microelectronics (ICM)*. 2024, pp. 1–8. DOI: `10.1109/ICM63406.2024.10815830`.

[3] Yichao Zhang et al. *A 1024 RV-Cores Shared-L1 Cluster with High Bandwidth Memory Link for Low-Latency 6G-SDR*. 2024. arXiv: `2408.08882 [cs.DC]`. URL: `https://arxiv.org/abs/2408.08882`.

[4] Javier Acevedo et al. "Hardware Acceleration for RLNC: A Case Study Based on the Xtensa Processor with the Tensilica Instruction-Set Extension". In: *Electronics* 7.9 (2018), p. 180. DOI: `10.3390/electronics7090180`. URL: `https://doi.org/10.3390/electronics7090180`.