D4.2

Heterogeneous Acceleration for Efficient Processing

Funded by
the European Union

Revision v1.0

Work package	WP4
Task	T4.1, T4.2
Dissemination level	PU – Public, fully open. e.g., website
Deliverable type	R – Document, report (excluding periodic and final reports)
Due date	30-9-2024
Submission date	30-8-2024
Deliverable lead	ETHZ
Version	v0.1
Authors	Marco Bertuletti (ETHZ)
Contributors	Yichao Zhang (ETHZ), Alessandro Vanelli-Coralli (ETHZ), Viktor Razilov (TUD), Markus Ulbricht (IHP), Michael Roitzsch (BI)
Reviewers	Michael Roitzsch (BI)

Abstract

As processing requirements of 6G base-stations skyrocket, the HW acceleration capabilities must keep up the pace. Specialization plays a key role, leading to heterogeneous architectures, but programmability must be in focus, to ensure fast time-to-market and reusability over an evolving telecommunications standard. In this panorama RISC-V plays a key role. Nevertheless, fast and energy efficient HW is also required to be trustworthy, to meet demands of Governments and Consumers. This deliverable addresses the problem, proposing 6G heterogeneous digital accelerators for a trustworthy execution platform.

Keywords

Heterogeneous Acceleration, RISC-V, ManyCore, Vector Processor, FEC, MAC scheduling.

Document Revision History

Version	Date	Description of change	Contributor(s)
V0.1	04-09-2024	Initial version of deliverable	Marco Bertuletti (ETHZ)
V0.2	06-18-2024	Accept contributions from partners	Marco Bertuletti (ETHZ) Yichao Zhang (ETHZ), Alessandro Vanelli-Coralli (ETHZ), Viktor Razilov (TUD), Markus Ulbricht (IHP), Michael Roitzsch (BI)
V0.3	07-12-2024	Apply suggestions from reviewers	Marco Bertuletti (ETHZ) Michael Roitzsch (BI)

Contributing Partners

Abbreviation	Company name
BI	BARKHAUSEN INSTITUT
EAB	ERICSSON
CYB	CYBERUS TECHNOLOGY
EUR	EURECOM
TUD	TECHNISCHE UNIVERSITAET DRESDEN
WINGS	WINGS ICT SOLUTIONS
ETHZ	EIDGENOESSISCHE TECHNISCHE HOCHSCHULE ZUERICH
IHP	IHP MICROELECTRONICS
NNF	NOKIA NETWORKS FRANCE
IIIV	NNF/IIIV LABS
KAL	KALRAY

Disclaimer

The information, documentation and figures available in this deliverable are provided by the COREnext project's consortium under EC grant agreement **101092598** and do not necessarily reflect the views of the European Commission. The European Commission is not liable for any use that may be made of the information contained herein.

Copyright Notice

Executive Summary

D4.2 presents the development of the acceleration components described in D4.1 and their integration in a heterogeneous computing platform for efficient processing. First, the architecture of the heterogeneous platform is described. Its modular structure is flexible enough to accommodate all the accelerating components developed. The rationale for adopting RISC-V as the foundational ISA for implementing the acceleration capabilities is described. The overall workload addressed is detailed and the most computing-intensive blocks requiring hardware acceleration are highlighted. Second, the acceleration components are described. We present the progress achieved on a programmable Many-Core RISC-V accelerator, a programmable Vector processing accelerator, a FEC accelerator, and a MAC Scheduling accelerator. We evaluate the results, benchmarking the key performance indicators of the application. COREnext marks significant advancements in the acceleration of 5G base-band processing.

Table of Contents

List of Figures

List of Tables

Acronyms and Definitions

WP	Work Package
KPI	Key Performance Indicators
5G	Fifth Generation
6G	Sixth Generation
RAN	Radio Access Network
O-RAN	Open Radio Access Network Alliance
DPU	Data Processing Unit
TCU	Trusted Communication Unit
TEE	Trusted Execution Environment
RISC-V	Reduced Instruction Set Computer V
ISA	Instruction Set Architecture
NR	New Radio
SDR	Software Defined Radio
PHY	Physical Layer
UE	User Equipment
RU	Remote Unit
PUSCH	Physical Uplink Shared Channel
MAC	Medium Access Control
CP	Cyclic Prefix
FFT	Fast Fourier Transform
OFDM	Orthogonal Frequency Division Multiplexing
MIMO	Multiple Input Multiple Output
MMSE	Minimum Mean Squared Error
FEC	Forward Error Correction
DU	Distributed Unit
RRC	Radio Resource Control
PDCP	Packet Data Convergence Protocol
RLC	Radio Link Control
CRC	Cyclic Redundancy Check
QoS	Quality of Service
DRL	Deep Reinforcement Learning
RRM	Radio Resource Management
TCDM	Tightly Coupled Data Memory
DMA	Direct Memory Access

TTI	Transition Time Interval
LTE	Long Term Evolution
SIMD	Single-Instruction Multiple-Data
MPSoC	multi-processors system-on-chip
LDPC	Low Density Parity Check
LLR	Log-Likelyhood Ratio
AI	Artificial Intelligence
OFDMA	Orthogonal Frequency-Division Multiple Access
NOMA	Non-Orthogonal Multiple Access
CNN	Convolution Neural Network
DNN	Deep Neural Network

1 Introduction

5G enabled augmented reality, the Internet of things, the Internet of Vehicles, device-to-device communications, remote healthcare, machine-to-machine communications, and drones. 6G will bring new opportunities, including telepresence, autonomous driving vehicles, indoor and outdoor wireless positioning and sensing. Nevertheless, this makes the network of interconnected wireless devices bigger. It comes at the cost of more processing for base stations and more risks of attacks from malign actors hijacking the network. COREnext addresses both problems, proposing an energy-efficient and trustworthy acceleration.

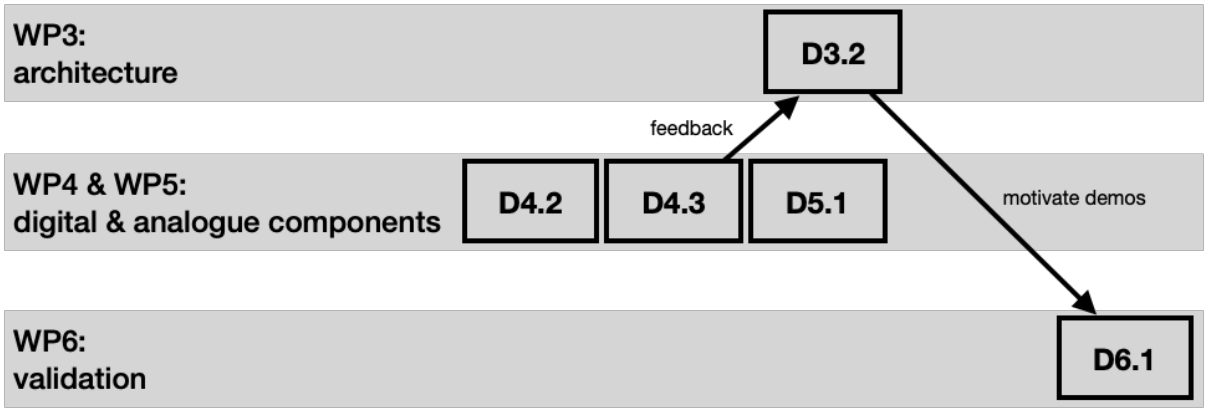

Figure 1: Overview of deliverables connected to D4.2 and flow between them.

D4.2 follows D4.1, where the concept of a trustworthy heterogeneous acceleration platform was outlined. The trustworthiness aspects are treated in D4.3, while D4.2 focuses on the progress of WP4 in developing the required acceleration capabilities. This deliverable provides results based on quantitative Key Performance Indicators (KPIs), giving feedback to WP3 to further elaborate the network architecture structure in D3.2. The quantitative evaluation of D4.2 also provides support for further tests and benchmarking in WP6, where an experimental setup to evaluate the project output will be developed

D4.2 is structured in two main parts. Section 2 describes the connection between accelerators in the proposed heterogeneous architecture and describes the addressed workloads, focusing on the most compute-intensive tasks. Section 3 presents the development advancements for the acceleration components first presented in D4.1. A summary concludes the document, reporting on the overall computing capabilities achieved.

2 COREnext Baseband Processing Platform

The COREnext project is delivering hardware building blocks towards an architecture for 6G terminal devices, 6G base stations, and edge cloud nodes. The overall COREnext architecture is described in deliverable D3.2: Integration of trustworthy disaggregated computing architecture. Besides providing the functionality and computational capacity to support upcoming 6G use cases (see D2.1: Use cases and requirements), COREnext specifically builds upon two non-functional pillars: efficiency and trustworthiness.

In this deliverable, we focus on the **acceleration hardware** necessary to deliver the operational efficiency of future 6G infrastructure. Trustworthiness aspects are addressed in the sibling deliverable D4.3: Trustworthy computation and orchestration. This section first describes how acceleration fits into the overall COREnext architecture (Subsection 2.1). We then zoom into the 6G signal processing chain and discuss which processing functions our acceleration addresses (Subsection 2.2).

2.1 Heterogeneous Processing Architecture

Energy efficiency is needed to meet sustainability goals, especially for hardware like 6G terminals and base stations, where we can expect large-scale deployment. When the same processing can be performed with less invested energy, this will scale across the entire deployed fleet of such devices. Acceleration in processing is an important tool as purpose-built accelerators can typically fulfil the same task with less energy compared to a general-purpose core.

What is important is to pick the right processing steps to accelerate to amortize development and manufacturing resources against the expected energy savings. We discuss this aspect in subsection 2.2. Equally important is the integration of acceleration elements and general-purpose processors. The latter are still needed for overall platform orchestration, while the former will contribute the energy efficiency for specific tasks. The interaction between both parts has a large impact on overall efficiency: Data movement has a major impact on both energy usage and latency of operations. A tighter integration allows more fine-grained acceleration as data must be shipped over a shorter physical distance or in smaller volume. However, more specialized accelerators tend to be dedicated devices separate from processors. While data must be shipped to them, their improved acceleration performance can result in greater overall efficiency. In COREnext, we develop and evaluate both core-integrated and dedicated accelerators for specific parts of the 6G processing chain.

2.1.1 Trustworthy Orchestration

The interaction between accelerators and general-purpose processors also raises issues of trustworthiness. As a 6G base station will process data streams on behalf of many different client devices, these data streams must be separated from each other. This need for strong isolation becomes even more apparent as third-party applications become integrated into the Radio Access Network (RAN) with initiatives like O-RAN. There, code from third-party developers will run within

the RAN to allocate and configure data streams from terminal devices while interacting with external infrastructure in first- or third-party clouds.

Given that compute and data from multiple tenants will be hosted on COREnext hardware, it is imperative that **strong isolation is embedded deeply into the platform** to prevent cyberattacks or privacy leaks. At the same time, this isolation must not introduce prohibitive overhead. We discuss our solutions in more detail in deliverable D4.3. Here, we give just a brief overview as a context for the following discussions on individual accelerator components.

Our main mechanism for isolation is control of communication paths by a trusted orchestrator. This principle is exemplified by the M³ platform (Figure 2), which is an architecture for systems-on-chip, where each compute resource like a general-purpose core or an accelerator is placed in its own isolated tile. Any communication between tiles is passed through a special security component, the Trusted Communication Unit (TCU), which enforces a communication policy. These policies are programmed into the TCU by the M³ kernel, which abstracts communication rights with a capability system that higher-level resource managers can use.

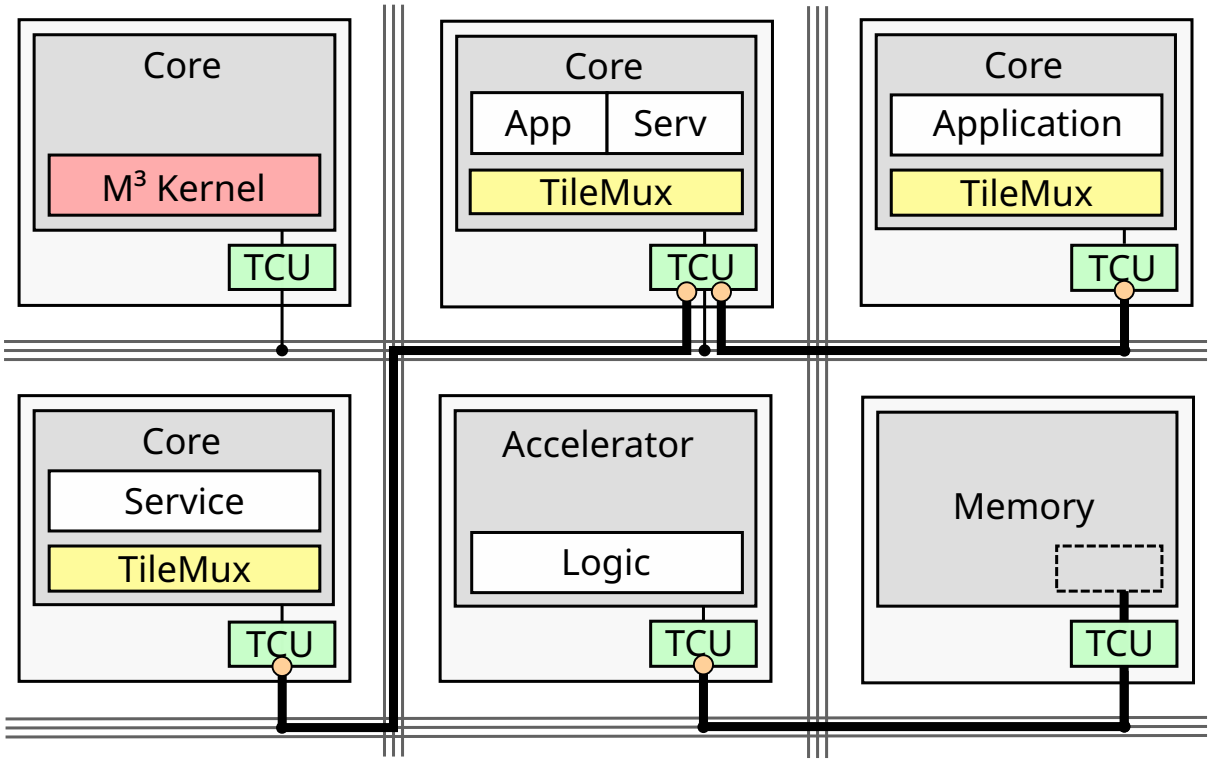

Figure 2: M³ architecture for trustworthy integration of accelerators

M³ is a research architecture, where we believe many of the accelerators discussed in this deliverable could be integrated to form an overall 6G signal processing platform. When leaving the system-on-chip level, other solutions like FPGA-based hardware with isolation for multiple tenants, network-level isolation by way of Data Processing Units (DPUs), or Trusted Execution Environments (TEEs) for distributed services become relevant. We refer to D4.3 for a more in-depth discussion.

2.1.2 RISC-V-Based Acceleration

The RISC-V Instruction Set Architecture (ISA) is an open-source and extensible instruction set, positioned to become an important cornerstone for processing platforms developed and built in Europe. There are two main reasons for the COREnext project to adopt RISC-V: RISC-V ISA is **extensible**, allowing architecture specialization and enabling hardware-software co-design, RISC-V ISA is **free and open-source**, therefore it increases trustworthiness.
The RISC-V ISA leaves a section of the encoding space free, encouraging the development of custom extensions. Extensions to the ISA can be used for architecture specialization to the targeted workload, New Radio (NR) processing in the case of COREnext. The free RISC-V ISA also pushes forward hardware-software co-design: the bottlenecks of software processing can drive the development of new instructions. In the framework of the COREnext project, this has relevance to the Software Defined Radio (SDR) approach to network function acceleration. Network functions can indeed be mapped to software for programmable or semi-programmable hardware components to speed up time to market, and increase flexibility in diverse deployment scenarios
The open-source RISC-V ISA pushes towards the development of open-source hardware and software. Open-source hardware fosters trustworthiness because it is fully transparent. It therefore establishes a perfect symmetry of knowledge across designers, and potential attackers, reducing the risk of hidden back-doors being exploited.

2.2 6G Signal Processing

The physical (PHY) layer converts digital bits to outgoing radio waves in the downlink direction and vice versa in the uplink direction. The low-PHY high-PHY nomenclature, dividing the PHY channel into two different sections, refers to the O-RAN 7.2 functional split. This option moves the User Equipment (UE) related functions of the PHY layer to network Remote Units (RUs), with the benefit of relaxing fronthaul network bitrate and delay requirements [Lar19]. By accelerating the functions of both low-PHY and high-PHY, **COREnext addresses the even more aggressive splits 7.3 and 6.** As a use-case for the acceleration of lower PHY, COREnext focuses on the Physical Uplink Shared Channel (PUSCH), one of the most latency-critical 5G channels, directly affecting the UE experience. Specifications for 5G PUSCH require a processing latency of less than 1ms.
Following PHY in the uplink direction, The Medium Access Control (MAC) layer is responsible for controlling access to the shared radio resources, and scheduling transmissions within the 5G network.

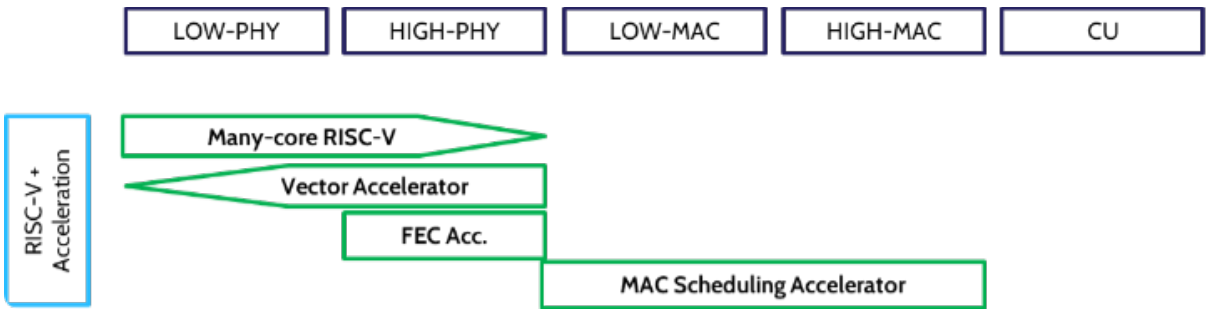

Figure 3: Layers of 5G processing and corresponding accelerators from D4.2

The next three sections detail the most compute-intensive functions of low-PHY, high-PHY, and MAC layers addressed by COREnext heterogeneous processing platform.

2.2.1 Low-PHY Processing

In the 7.2 O-RAN functional spit 5G low-PHY is executed in the RU [Lar19] and includes:

- RF conversion, Cyclic-Prefix (CP) removal, Fast Fourier Transform (FFT), and digital beamforming in the uplink direction.
- iFFT, digital beamforming, CP addition, and RF conversion in the downlink direction.

The most compute-intensive functions of the low-PHY are FFT/iFFT and beamforming. FFT/iFFT is used for demodulation/demodulation of the received/sent signal, in the framework of the Orthogonal Frequency Division Multiplexing (OFDM) algorithm. The incoming transmission bitstream is distributed across the bandwidth on multiple orthogonal subcarriers. OFDM is executed on separate data streams for each transmission antenna. Digital beamforming consists of the spatial modulation of the transmitted signal. It is a technique enabled by massive Multiple-Input Multiple-Output (MIMO) communication and involves the combination of streams from different antennas and the multiplication by beamforming coefficients computed in the upper MAC layer.

The ManyCore accelerators and the vector accelerator, described in sections 3.1, 3.2 and 3.3 of D4.2 are responsible for the acceleration of the most computing-intensive functions of lower PHY.

2.2.2 High-PHY Processing

The High PHY functions include:

- Encoding for forward error correction (FEC), symbol mapping, layer mapping, and precoding in the downlink.
- Channel estimation, antenna combining, channel equalization, layer demapping, symbol demapping, and decoding in the uplink.

The 7.2 O-RAN functional split proposes to process precoding on the RU, while the rest of the functions are implemented in the Distributed Unit (DU) [Lar19]. COREnext moves to split 7.3 and 6, where all the functions are moved to the RU to reduce user latency. The project provides efficient processing platforms for these tasks, focusing on computationally demanding uplink processing. As execution platforms for most of the tasks we target and optimize the ManyCore accelerators and the vector accelerator (section 3.2, 3.3).

The FEC, however requires a higher compute performance than vector processors can provide, and its execution pattern does not map well on vector or other general-purpose processors. Hence, we built an application-specific FEC accelerator in section 3.3.

2.2.3 MAC Layer

In the 7.2 O-RAN functional split, the MAC layer resides in the DU along with Radio Resource Control (RRC), Packet Data Convergence Protocol (PDCP), Radio Link Control (RLC), and higher PHY functions.

The MAC layer is responsible for managing access to the shared radio channel between multiple User Equipment (UEs) devices connected to the base station. Its key tasks include:

- Scheduling data transmissions: The MAC layer determines which UE can transmit data at a given time and for how long.
- Contention resolution: In situations where multiple UEs want to transmit simultaneously, the MAC layer implements mechanisms to resolve contention and avoid collisions.
- Flow control: The MAC layer ensures that the data rate on the radio channel doesn't exceed its capacity, preventing congestion and data loss.
- Error detection: The MAC layer adds Cyclic Redundancy Check (CRC) codes to transmitted data packets for error detection at the receiver.

Out of all these tasks, the focus in COREnext lies on the acceleration of the resource scheduling, which is critical for the MAC layer. To address the demands of high throughput and Quality of Service (QoS), an intelligent and high-performance scheduler is necessary. The scheduler tackles a complex optimization problem that considers various factors, including throughput, latency, overall network utilization, and channel conditions. Artificial Intelligence (AI), particularly deep Reinforcement Learning (RL), has shown significant potential for solving Radio Resource Management (RRM) challenges in wireless communication systems.

3 Digital Components

3.1 Programmable Many-Core RISC-V Accelerator for PHY Processing: TeraPool

To address the low latency and high-throughput of 5G NR PHY-layer, we exploit the large-scale parallelism of the massive TeraPool-SDR cluster, featuring 1024 processing cores, each supporting the RISCV32IMA ISA and application-specific extensions, and 4MiB of fully shared multi-banked L1 scratchpad.

3.1.1 Prototype Status

The TeraPool-SDR Snitch [Zar20] cores are lightweight single-stage in-order cores. They have independent instruction streams, and they can issue outstanding transactions to the shared scratchpad, hiding the latency of the cores to memory interconnect. We added to Snitch domain specific extensions. The extensions are implemented in pipelined functional units. Offloads of instructions to the functional units are marked in Snitch scoreboard. ISA-extensions including post-increment load and stores, fused multiply-add and SIMD operations with different integer precision (16b, 8b) were grouped in the Xpulpimg custom subset. Standard floating point zfinx and zhinx RISCV extensions were also added to provide floating point support without the overhead of a second register file. With the same rationale, smallfloat SIMD extensions were also integrated. Finally, we introduced complex f16 dot-product instructions, to complete a complex multiply-add in a single cycle.

The TeraPool design is physically feasible thanks to its hierarchical interconnects between cores and memory [Yic24]. There are three levels: the *Tile* contains eight cores and a fully connected crossbar to 32 KiB of SRAM Tightly Coupled Data Memory (TCDM). Eight Tiles are packed in a *SubGroup* and can send each other load/store requests via an 8x8 fully connected crossbar. In a *Group* the Tiles of four SubGroups communicate the same way via a 32x32 interconnect. Other three 32x32 interconnects are instantiated in the Group to address each Tile in the other three Groups of the cluster. The design was placed and routed in GlobalFoundries 12nm LPPLUS FinFET technology, using Synopsys' Fusion Compiler 2022.03. The placed and routed cluster occupies an area of 82mm².

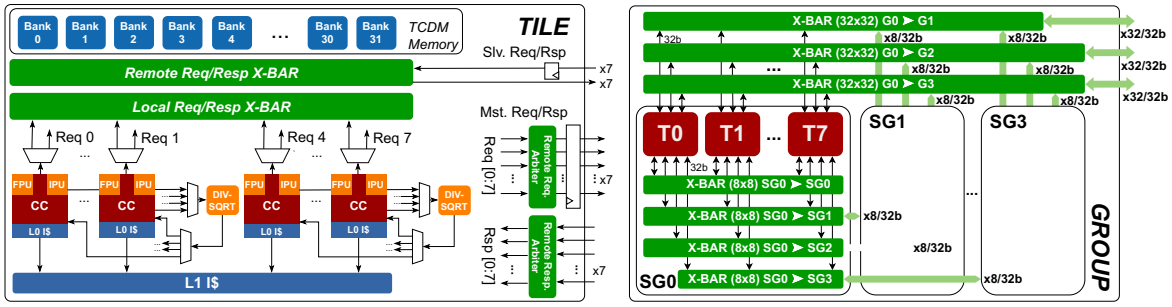

Figure 4: On the left, the TeraPool Tile, with 8 Snitch core-complexes and 2 shared division and square-root floating point units. The local Req/Resp crossbar gives 1 cycle access to the shared memory, 7 remote Req/Resp master, and slave ports connect to other hierarchical levels. On the right, the connection between the crossbars of a TeraPool Group.

At system level, a modular Direct Memory Access (DMA) split in three hierarchies was designed to move data from L2 to the large shared L1. The three levels of the DMA engine are frontend (configuration), midend (transfer split), and backend (data mover). One Tile-shared master AXI port per SubGroup supports L1 instruction cache refill or DMA-controlled data transfers. The cluster AXI ports address L2, DMA frontend, and peripherals, but they could also be used to plug TeraPool to an external host in a heterogeneous system.

3.1.2 Interactions with Partners

The work of ETHZ for TeraPool is aligned to the effort of KAL and TUD in the acceleration of the low and high PHY of 5G-NR. Due to the highly parallel nature of the cluster, TeraPool is more suitable to handle per antenna data streams, which are typical of the OFDM and beamforming processing. However, the programmable cores of TeraPool are flexible and can also be used for part of the high-PHY processing.

Over the project duration, ETHZ had collaboration meetings to explore the techniques used to increase the memory bandwidth of the Vector accelerator developed by TUD. This was useful to assess possible integration of vector co-processors in the TeraPool architecture.

Additionally, ETHZ provided NNF and IHP the expertise regarding Cheshire Linux-capable host. The partners used this open-source platform as system host. As Cheshire could also be used as a host for TeraPool itself, the work of ETHZ is nicely aligned with that of NNF and IHP.

3.1.3 Planned Performance Measurements

Work on the TeraPool accelerator addressed two key KPIs for baseband processing solutions: latency and power consumption. The benchmarks adopted for the architecture allow the implementation of a software-defined processing chain encompassing the 7.X split of 5G, including the main operators of low and high-PHY. We implemented the kernels required to run the PUSCH of 5G in different numeric precisions.

RTL simulations analyzed the latency of the processing steps. Power simulations in the typical corner (0.8V, 25°C) were executed using Synopsys Prime-Time. We run a Transition Time Interval (TTI) of PUSCH, analyzing a high-load use case, with 4096 subcarriers, 14 symbols, 32 beams, and

4 UE transmitting on the same subcarrier. The results in terms of power consumption and processing latency for one TTI are reported in Table 1. We also report the transfer time for the computation inputs/outputs, obtained from co-simulation with a DRAMSys model of HBM2E, plugged into the system via the AXI interconnect.

The results in Table 1 highlight a negligible transfer overhead and a computation latency of the floating-point pipeline <2ms for executing PUSCH in a high load use case. Specifications require the data of a TTI to be consumed in 1ms, however, the PUSCH is in general alternated with other channels, therefore, as a rule of thumb PUSCH TTI can stretch up to 2.5ms. We conclude that TeraPool is a good candidate for the acceleration of PUSCH, and that three clusters would be necessary to guarantee the required throughput. Power consumption is on average less than 10W, comparable to industry solutions for SDR.

	Precision	Computation(ms)	Transfer(ms)	Power (W)
FFT	fixed	0.574	0.046	6.090
	floating	0.781	0.046	5.630
BF	fixed	1.390	0.006	7.220
	floating	0.388	0.006	6.280
CHE	fixed	0.024	0.008	5.400
	floating	0.023	0.008	5.480
MMSE	fixed	0.719	0.048	4.950
	floating	0.616	0.048	4.340
PUSCH symbol	fixed	DMRS: 0.305	0.108	5.84
		DATA: 2.404		
	floating	DMRS: 0.190	0.108	5.50
		DATA: 1.610		

Table 1: Computation and Transfer latency for SDR kernels implemented on TeraPool in different precisions. Power consumption of the implemented kernels in Synopsys Prime-Time simulation of the 12nm design clocked at 800MHZ.

3.2 Programmable Vector Processing Accelerator

Fixed-function accelerators excel in terms of efficiency but pose a challenge for virtualization. Therefore, we develop programmable accelerators that are placed between general-purpose processors and fixed-functions accelerators on the performance-flexibility trade-off curve.

Vector processors pose an interesting basis for such programmable accelerators and the research community's interest in them has surged. They distribute the instruction fetch and decode overhead over a vector of data items. The data items are then processed in parallel by multiple functional units, the approach used being Single-Instruction Multiple-Data (SIMD). However, in the vector programming abstraction, the length of data vectors is agnostic of the number of parallel

functional units: vectors of any size will be processed in multiple cycles. Nevertheless, vector instructions with dependencies can be executed (almost) in parallel, chaining the functional units.

Our aim to improve their ISA and microarchitecture in a way that optimizes the chaining and thus the utilization of functional units. Increased utilization leads to better performance and lower energy consumption because of reduced leakage. We investigate the problem by examining state-of-the-art open-source vector processors in the context of communications signal processing and by modelling different vector processor microarchitectures.

3.2.1 Prototype Status

Our first step was to look at state-of-the-art open-source high-performance vector processors, such as Ara [Cav22], and port high-PHY uplink kernels to the RISC-V Vector (RVV) ISA. We selected the Long Term Evolution (LTE) Uplink Receiver PHY Benchmark [Sja12], that was specifically designed for this kind of benchmarking, for its portability. We measured the runtime of the kernels on Ara. The results are in Table 2.

Our first optimization proposal is the dual vector load which loads two vectors that are operands to a follow-up binary operation in parallel to bring the execution of the latter instruction forward [Raz23]. Theoretical analysis of this ISA extension showed that it is beneficial for compute-bound and some memory-bound programs. The highest possible speedup is 33 % and we achieved a speedup of 21 % in an implementation with about 2 % area overhead [Raz23].

In our research, we identified the vector register file as a bottleneck which is even more severe when the utilization is high. To study the problem, we have built a cycle-accurate software model of a vector processor with multiple architectural options. The model allows to study the impact of bank conflicts on the runtime and the utilization of the functional units. We have developed and tried novel vector register file architectures that improve the area efficiency of a vector processor. The results have been submitted to a journal for publication and are currently under review.

Kernel Name [Sja12]	High-PHY Step (c.f. section 2.2.2)	Scalar [k cycles]	Vector [k cycles]	Speedup
Matched Filter	Channel Estimation	42	0.82	51
IFFT/FFT	Channel estimation, Channel equalization	121	N.A.	N.A.
Windowing	Channel estimation	4.7	0.84	6
Combiner Weights Calculation	Antenna Combining	12 356	N.A.	N.A.
Antenna Combining	Antenna Combining	215	2.3	93
Deinterleave	Layer demapping	1 076	214	5
Soft Symbol Demap	Symbol demapping	1 624	18	90

Table 2: Performance measurements and comparison of different High-PHY kernels [Sja12] on Ara [Cav22] using only scalar RISC-V instructions and after vectorization. The LTE parameters are 50 resource blocks, 4x4 MIMO, and 4-QAM modulation.

3.2.2 Interactions with Partners

A research collaboration is ongoing with ETHZ. ETHZ is exploring the integration of tiny vector co-processors in the ManyCore cluster described in section 3.1. TUD is providing support to extend the co-processors with dual vector-load.

TUD also collaborates with BI to integrate Ara [Cav22] into the M3 platform for trustworthy heterogenous Multi-Processors Systems-on-Chip (MPSoCs) [Asm16].

3.2.3 Planned Performance Measurements

Using a theoretical model, we want to measure the utilization and runtime of the kernels in Table 2 and [Raz22] and compare an ideal vector processor and state-of-the-art designs with our optimizations.

3.3 FEC Accelerator

IHP, within the last reporting period, was able to characterize a floorplan of Low-Density Parity Check (LDPC) decoder compatible with 802.11n WLAN standard, which can deliver decoding throughput up to 1200 Gb/s for coded stream and 1000 Gb/s for uncoded data. The design has been verified in 28 nm CMOS technology with the maximal clock frequency up to 2000 MHz, assuming worst-case conditions (slowest process, 0.9V, 125 C). This relatively high clock frequency has been achieved due to fully unrolled, pipelined architecture. In our case, we decided to employ seven pipeline stages for every iteration, as depicted in Figure 5**Figure 5**.

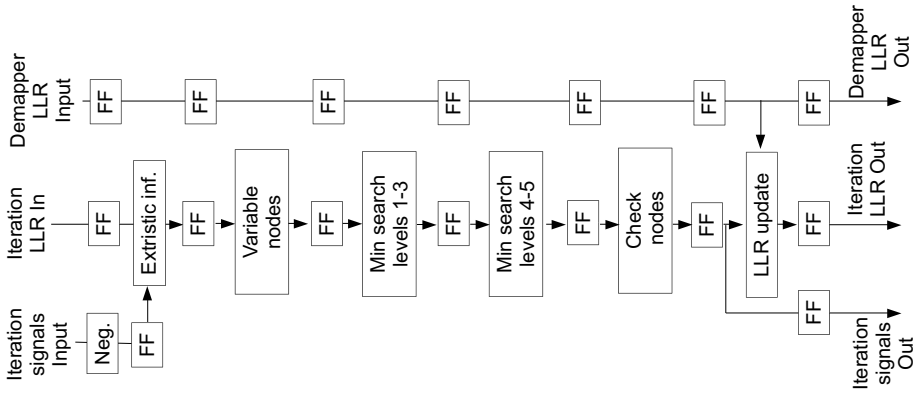

Figure 5: Pipeline of the designed LDPC decoder.

The design supports a codeword length of 648 bits with 4-bit quantization of incoming Log-Likelihood Ratios (LLR). Thus, the processor word consists of 2592 bits that are shifted through the pipeline in every clock cycle. The employed decoding algorithm is a typical min-sum with a standard IEEE 802.11n (648,540) parity matrix. Every operation of the LDPC decoding uses 1 clock cycle, except the minimum search for LLR values of each parity equation. The min-search modules use a binary tree comparator architecture divided into two clock cycles. Each comparison tree has five stages, and in this IEEE standard realization, one minimum LLR value out of 22 candidates is searched.

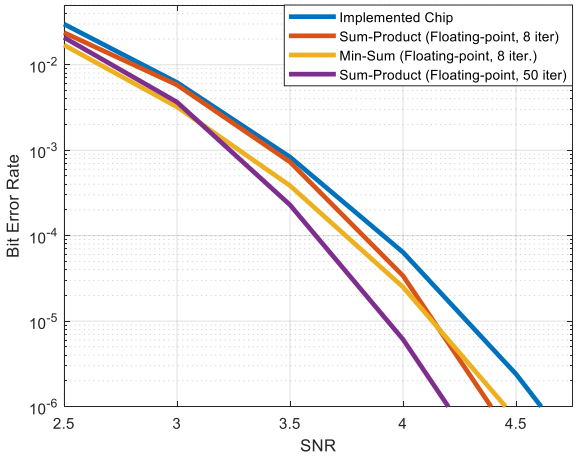

Figure 6: Bit error rate performance of the implemented chip as compared to the simulated results of floating-point min-sum and floating-point sum-product decoding algorithms.

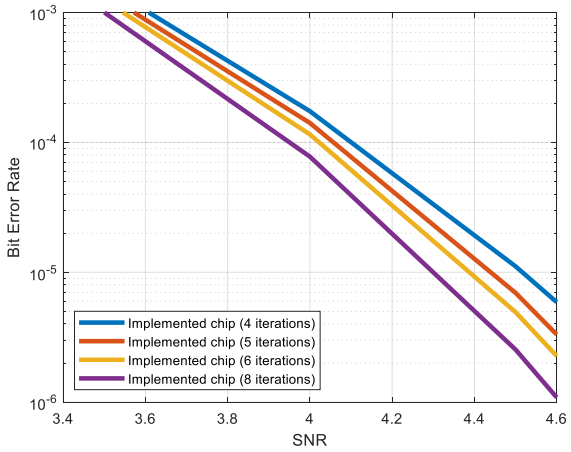

Figure 7: Bit error rate performance of the implemented chip in view of the number of performed decoding iterations.

Figure 6 compares the bit error correction performance of the implemented VHDL decoder with the floating-point simulation of min-sum and sum-product algorithms, while Figure 7 depicts the bit error rate performance as a function of the number of performed decoding iterations. Figure 8 explains the impact of the number of LLR quantization bits on the bit error rate performance.

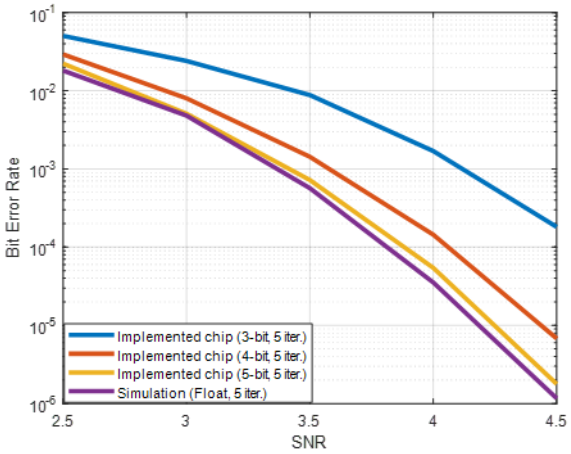

Figure 8: Bit error rate performance of the implemented chip in view of the number of LLR quantization bits.

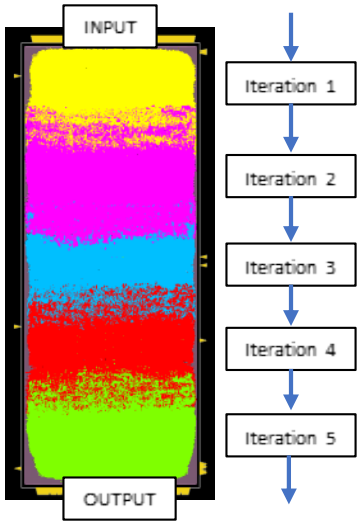

Figure 9: Chip floorplan of fully unrolled LDPC decoder with highlighted iterations. Each color in the chip floorplan indicates one iteration implemented as separate hardware block.

3.3.1 Prototype Status

Considering the performance simulations of the implemented netlist in Figure 6 to Figure 8, five unrolled iterations with four-bit LLR quantization have been selected for the final floorplan. This resulted in a chip area of ~5 mm² with the layout shown in Figure 9. Due to serious congestion problems of signal routing representing the parity matrix hardware connections, core utilization is only 46%. This means that 54% of silicon is not used for transistor implementation, only for the realization of metal connections for data routing. The LLR data encoding in the implemented chip follows two schemes. All the input and output LLR values are represented as U2-encoded integers. Such representation allows the use of all standard addition and subtraction operations available in VHDL libraries. The only exception are the min-search modules, which use sign-amplitude representation. This reduces the comparison complexity in the hardware. Thus, before and after the min-search operation is performed, the data is converted between U2 and sign-amplitude representations. A large chip area overhead and power consumption overhead are caused by

shifting the demapper LLR values through the whole processing pipeline. Those values are needed at the end of the iteration processing to add to the computed LLR correction values. Considering the whole chip architecture with five iterations, where in every iteration, seven stages are implemented, the shift register has 5×7=35 stages. Each stage consists of 648x4=2592 FFs. Thus, in total, 35x2592=90720 FFs are used only to synchronize the demapper LLR data with the decoding hardware. Such a massive amount of FFs consumes significant silicon area and dissipates considerable power. Thus, it would be beneficial to convert this FF-shift-register to an SRAM-memory-based implementation, but at the targeted clock frequency of 2000 MHz, we cannot generate an SRAM array that can work with this timing. If the clock frequency is reduced to approx. 500 MHz, with the resulting decoding throughput of 250 Gbps, then a considerable reduction in area and power is possible. Moreover, at 500 MHz, the pipeline length can be reduced from seven to four stages, and additional reductions in chip size are possible. Thus, the proposed structure can also be used to implement slower decoders with significantly reduced demands on resources.

3.3.2 Interactions with Partners

In respect to the FEC accelerator, there are no interactions planned. Lead partner IHP will of course consult with other components as well as the overall architecture to ensure alignment of development and architectural fit.

3.3.3 Planned Performance Measurements

The main targeted performance measure is the decoding throughput, where we were able to achieve up to 1200 Gb/s for coded stream and 1000 Gb/s for uncoded data. Secondary performance indicators are area and power, which can be greatly improved by using SRAMS instead of FFs, at the cost of decoding throughput.

3.4 MAC Scheduling Accelerator

Scheduling downlink and uplink data streams is a critical function of the MAC layer in 5G/6G networks. The need for ultra-high throughput and stringent QoS requirements necessitates an intelligent, high-performance scheduler capable of making decisions with low latency. The scheduler must effectively address a complex optimization problem that involves various parameters such as throughput, latency, overall network utilization, and channel conditions (e.g., SNR or SINR). Radio Resource Management (RRM) challenges were solved by deep Reinforcement Learning (RL) artificial intelligence models.

We propose a MAC architecture comprising two processing domains: one for the control plane and another for the data plane. These domains are differentiated by their processing operations; control operations are executed on CPUs, while user data is processed by a domain-specific accelerator, which is independently connected to the upper layers to enhance performance.

The control plane utilizes an AI accelerator to make radio resource allocation decisions, which are subsequently communicated to the data plane. For the AI accelerator, we are considering the NVIDIA Deep Learning Accelerator (NVDLA), currently being integrated into our RISC-V based CRISPY platform.

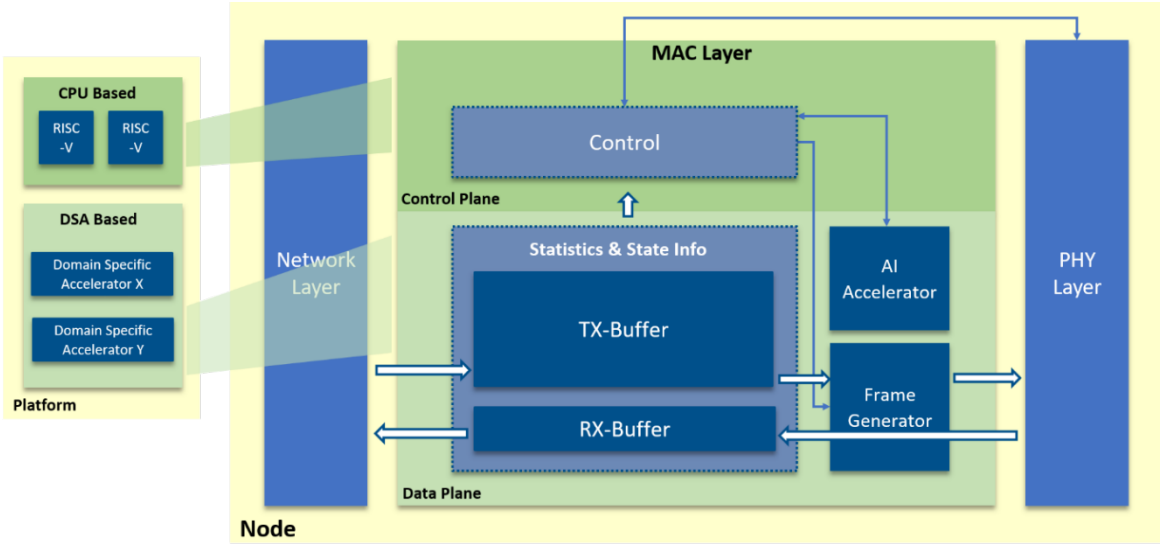

Figure 10: Block diagram of MAC with dedicated AI accelerator. This split is also suitable for FPGA development boards with CPU based processing side.

3.4.1 Prototype Status

We consider a single-cell multiuser system in which data is transmitted via MIMO technology. Additionally, we assume a single RF processing chain at the base station, which is connected to K antennas, communicating with M users. Both classical Orthogonal Frequency-Division Multiple Access (OFDMA) and the emerging Non-Orthogonal Multiple Access (NOMA) methods are investigated. For simplicity, we initially ignore the case of multiple RF chains and the associated user grouping problem, though this may be introduced at a later stage.

As part of our investigations, we plan to use NS-3 as a system-level network simulator for both the training and initial evaluation of various AI-based scheduling agents. NS-3 provides integration with OpenAI Gym, a Python toolkit for developing RL algorithms.

3.4.2 Interactions with Partners

As part of our ongoing integration of the AI accelerator into the RISC-V based CRISPY platform, we are exploring its application to Ericsson's Convolution Neural Network (CNN) based RF-fingerprinting algorithm. Our primary objective is to enhance performance, particularly with respect to latency, which is crucial for successful deployment in real-world environments. The collaboration between Ericsson and IHP is focused on providing robust hardware support for emerging Deep Neural Network (DNN) based 6G applications. Within this partnership, IHP will supply advanced simulation tools to fine-tune the DNN model to the hardware backend and optimize hardware configurability options to maximize key performance indicators. Our efforts will primarily investigate the system performance-cost trade-off, aiming for an optimized hardware-software co-design. We plan to test the optimized configuration on an FPGA prototype of the accelerator, which we hope will allow us to conduct accurate performance analyses. This work is independent of the developments that we conduct on AI supported radio resource management.

3.4.3 Planned Performance Measurements

The objective of the scheduling accelerator is to optimize throughput, latency and overall network utilization under changing channel conditions.

4 Summary

This document describes the progress of the COREnext project on building a **heterogeneous processing platform** for next generation telecommunications. The focus is on achieving a throughput compliant with 5G and beyond standards, while keeping power consumption low. The design of our heterogeneous computing architecture also targets trustworthiness. To achieve our target, we rely on the isolation capabilities of the M³ architecture. Our designs develop on the open-source RISC-V ISA, which we consider a key enabler for fully European hardware. The extensible instruction set of RISC-V is also particularly appealing for telecommunications, because it allows to implement domain-specific instructions and **fosters hardware-software codesign.**

We address the key and most computationally intensive processing steps in the lower and higher PHY: FFT/iFFT, beamforming, channel estimation, decoding for the uplink and forward error correction. To carry on these workloads, we developed a RISC-V ManyCore programmable accelerator and a RISC-V programmable vector processor, embracing the software-defined radio paradigm. For the computation intensive FEC workload we also developed a dedicated accelerator. We also target the acceleration of the MAC layer. We develop a system comprising two processing domains: one for the control plane, orchestrated by the CRISPY host, and another for the data plane, handled by a domain specific accelerator.

Acceleration Component	KPI	Benefit Measure
Programmable Many-Core RISC-V Accelerator for PHY Processing: TeraPool	Latency and Power Consumption of a 5G-PUSCH TTI	1.72ms @ 5.5W (typical corner, 0.8V. 25°C)
Programmable Vector Processing Accelerator	Speedup of LTE kernels vector implementation vs scalar	Speedup 5 (layer demapping) to 93 (antenna combining)
FEC Accelerator	Throughput	1000Gb/s @2GHz uncoded 1200Gb/s @2GHz coded
MAC Scheduling Accelerator	Throughput, Latency and Network utilization under changing channel conditions	work in progress

Table 3: Key Performance Indicators per acceleration components

The implemented systems meet the desired **latency and throughput constraints** for the 5G application under exam: the RISC-V ManyCore accelerator runs a demanding uplink application in a time compatible with the 5G-TTI lenght (1ms). Power consumption is competitive with industry solutions. Optimizations on the ISA of the Vector accelerator reduced the latency required to run RAN specific tasks. The FEC accelerator completes the set of functions needed to implement the entire PHY on the heterogeneous platform. With 5-bit LLR quantization bits it achieves almost the same BER performance of a min-sum and sum-product floating point algorithm, it delivers decoding throughput up to 1200 Gb/s for coded stream and 1000 Gb/s for uncoded data.

Experiments on the RISC-V vector processor demonstrated interesting opportunities for the vectorization of the workload, suggesting that combining vector cores with a ManyCore architecture could be material for further exploration.

5 References

[Lar19] L. M. P. Larsen, A. Checko and H. L. Christiansen, '' in IEEE Communications Surveys & Tutorials, vol. 21, no. 1, pp. 146-172, Firstquarter 2019, doi: 10.1109/COMST.2018.2868805.

[Asm16] N. Asmussen, M. Völp, B. Nöthen, H. Härtig and G. Fettweis, "M3: A Hardware/Operating-system Co-design to Tame Heterogeneous Manycores," in 21st International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS), Atlanta, GA, USA, 2016.

[Zar20] F. Zaruba, F. Schuiki, T. Hoefler and L. Benini, "Snitch: A Tiny Pseudo Dual-Issue Processor for Area and Energy Efficient Execution of Floating-Point Intensive Workloads," in *IEEE Transactions on Computers*, vol. 70, no. 11, pp. 1845-1860, 1 Nov. 2021, doi: 10.1109/TC.2020.3027900.

[Yic24] Y. Zhang, M. Bertuletti, S. Riedel, M. Cavalcante, A. Vanelli-Coralli, and L. Benini. 2024. TeraPool-SDR: An 1.89TOPS 1024 RV-Cores 4MiB Shared-L1 Cluster for Next-Generation Open-Source Software-Defined Radios. In Proceedings of the Great Lakes Symposium on VLSI 2024 (GLSVLSI '24). Association for Computing Machinery, New York, NY, USA, 86–91.

[Cav19] M. Cavalcante, F. Schuiki, F. Zaruba, M. Schaffner and L. Benini, "Ara: A 1-GHz+ Scalable and Energy-Efficient RISC-V Vector Processor With Multiprecision Floating-Point Support in 22-nm FD-SOI," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 28, no. 2, pp. 530-543, Feb. 2020, doi: 10.1109/TVLSI.2019.2950087.

[Raz22] V. Razilov, E. Matúš and G. Fettweis, "Communications Signal Processing Using RISC-V Vector Extension," 2022 International Wireless Communications and Mobile Computing (IWCMC), Dubrovnik, Croatia, 2022, pp. 690-695, doi: 10.1109/IWCMC55113.2022.9824961.

[Raz23] V. Razilov, J. Zhong, E. Matúš and G. Fettweis, "Dual Vector Load for Improved Pipelining in Vector Processors," 2023 IEEE Symposium in Low-Power and High-Speed Chips (COOL CHIPS), Tokyo, Japan, 2023, pp. 1-6, doi: 10.1109/COOLCHIPS57690.2023.10121996.

[Sja12] M. Själander, S. A. McKee, P. Brauer, D. Engdal and A. Vajda, "An LTE Uplink Receiver PHY benchmark and subframe-based power management," 2012 IEEE International Symposium on Performance Analysis of Systems & Software, New Brunswick, NJ, USA, 2012, pp. 25-34, doi: 10.1109/ISPASS.2012.6189203.