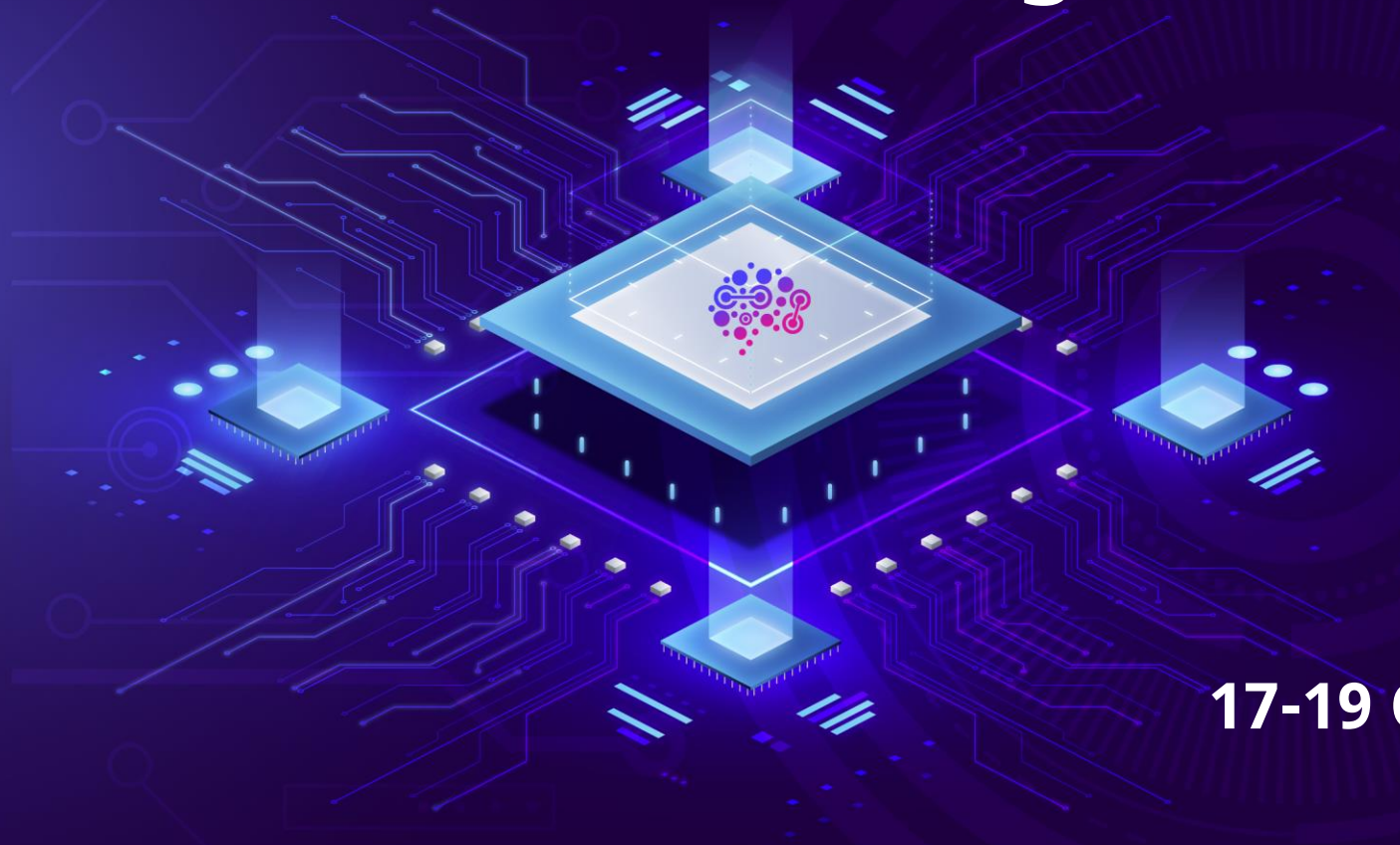
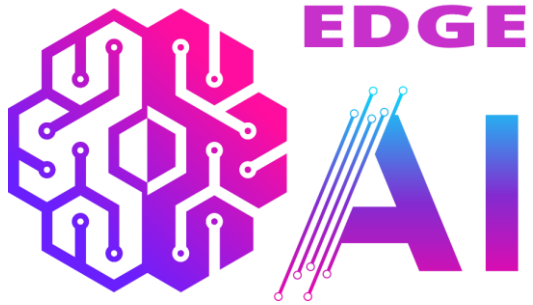
European Conference on EDGE AI Technologies and Applications - EEAI



Advancing emerging edge AI technologies and driving next generation intelligent applications.

17-19 October 2023, Athens, Greece

European Conference on EDGE AI Technologies and Applications - EEAI



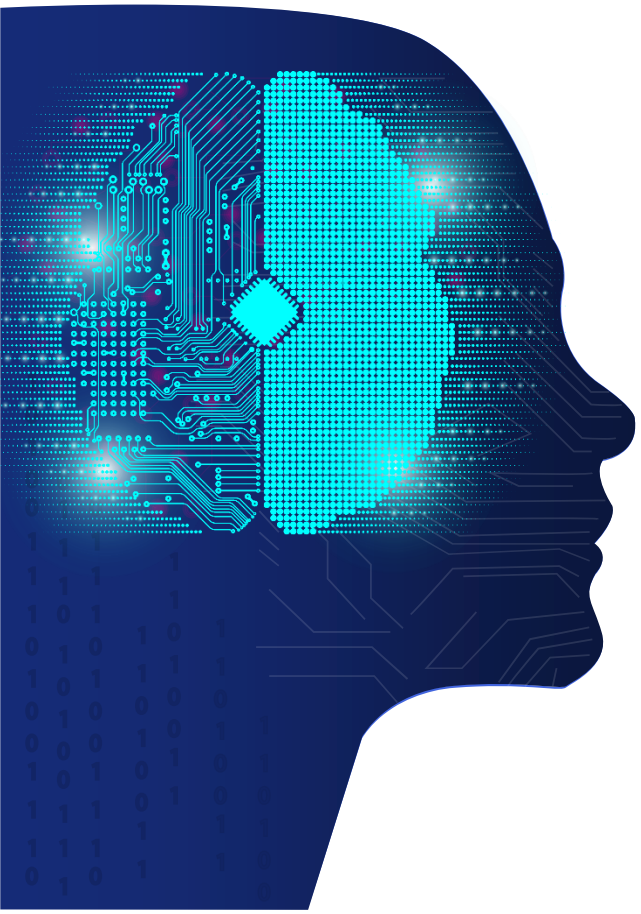
How to Accelerate Your Application on RISC-V Using a Coprocessor Interface

Dr. Christian Herber, NXP, Germany

17-19 October 2023 Athens, Greece

Copyright © 2023

Presentation Outline



- Introduction
- State-of-the-art
- Innovation Goal
- Methodology
- Design and Implementation
- Results
- Future Work
- Conclusion

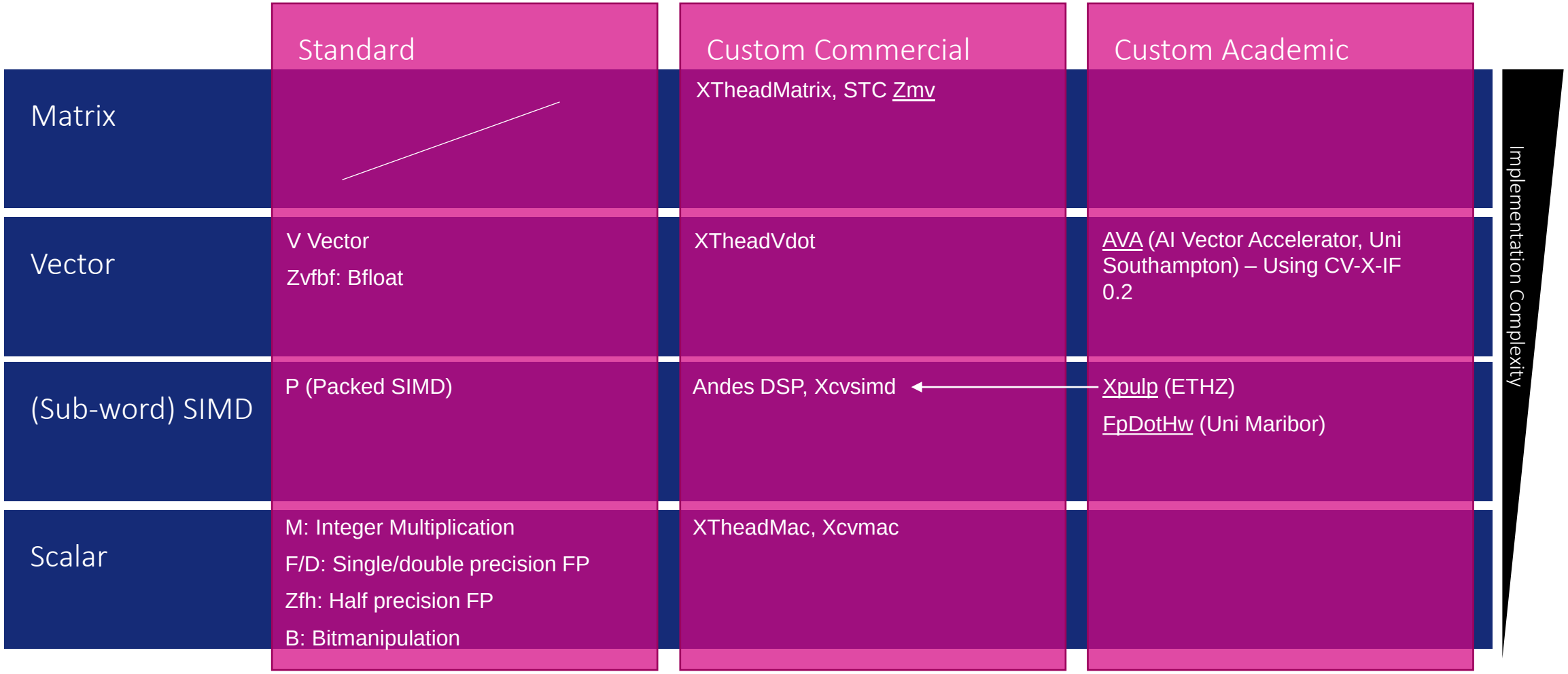
Copyright © 2023

Introduction

- RISC-V is a modular and extensible instruction set architecture
  - Pick from standard extensions with great ecosystem support
  - Add your own workload specific custom accelerator
- Coprocessor Interface as a key initiative in TRISTAN and ISOLDE projects
  - Extension of cores without requiring expertise on specific core
  - Efficiency improvement over standalone accelerators by reusing core infrastructure
  - Reuse of accelerators with multiple cores
- Core-V Extension Interface (CV-X-IF) is a coprocessor interface favored by many TRISTAN and ISOLDE members

Copyright © 2023

State-of-the-art: AI/ML Acceleration on RISC-V



Naming: Z… → extension related to the following letter. X… → custom extension, followed by vendor and then a designator

Copyright © 2023

State-of-the-art: Coprocessor Interfaces

	Fixed encoding space	Control flow	Instruction Format	Pair support	Speculative execution	Background instructions	LSU interface	Data interface
Arm® CDE	Yes	No	op dr imm op dr sr1 imm op dr sr1 sr2 imm	Yes	?	No	No	GPR, FPR
CV-IF-X	No, 16&32-bit instrs	No	1/2/3 register source operands	Yes	Yes	Yes	Yes	GPR
PicoRV PCPI	No	No	1/2 register source operands	No	No	No	No	GPR
RoCC		No	R-type (2 src, 1 dst)	No	No	Yes	Yes	GPR, FPR
NICE	Custom-0/1/2/3	No	R-type (2 src, 1 dst)	No	No	No	Yes	GPR
CXU-LI	Custom-0/1/2	No	R-type, I-type, flex	No	No	Yes	No	GPR

* Only subset of interfaces. No public information on proprietary interfaces like Andes ACE

Copyright © 2023

Innovation Goal

- **Converge industry on CV-X-IF as the go-to coprocessor interface in RISC-V**
- Advance specification to version 1.0
- Improve applicability to various pipeline depths and microarchitectures
- Superscalar/multi-threading support
- Multiple co-processors per hart
- Support for different instructions lengths (16/32/48/…)

Copyright © 2023

Methodology

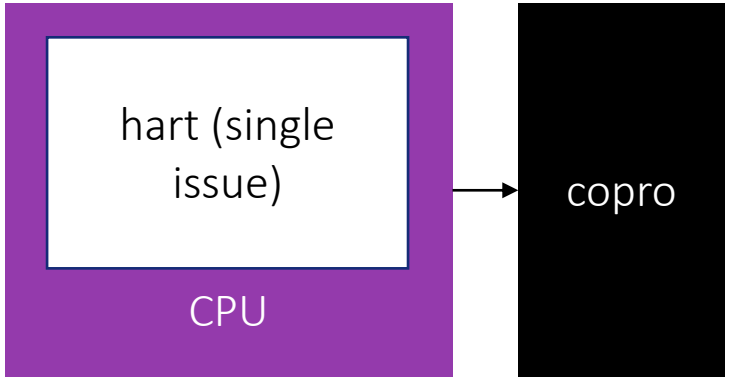
- Open-source workstream under the umbrella of the OpenHW Group
  - Repository: https://github.com/openhwgroup/core-v-xif
  - Specification: https://docs.openhwgroup.org/projects/openhw-group-core-v-xif/en/latest/
- Collaborative effort, public meetings and communication
  - Join the CV-X-IF channel on Mattermost https://mattermost.openhwgroup.org/all-users/channels/twgcorescvxif

Contributions welcome (specifications, accelerators, cores, verification etc.)

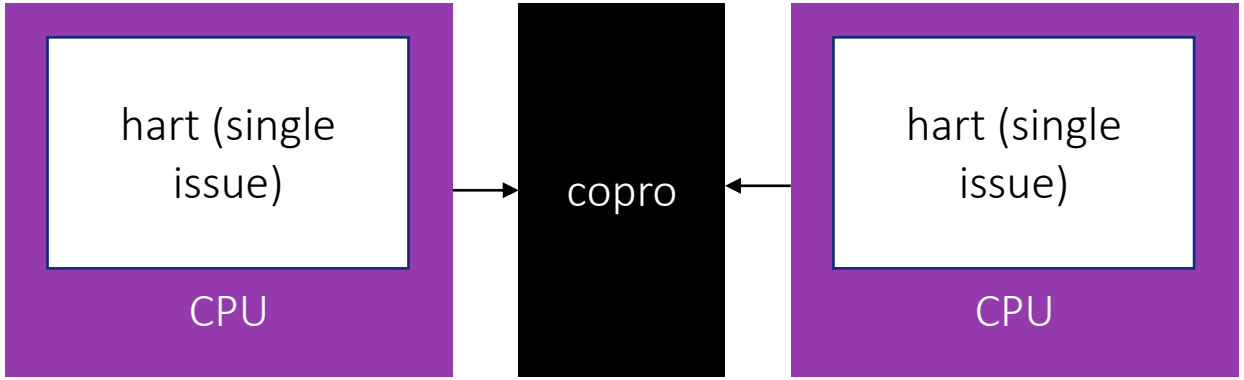
Copyright © 2023

Use-Cases: Processor – Coprocessor Configurations

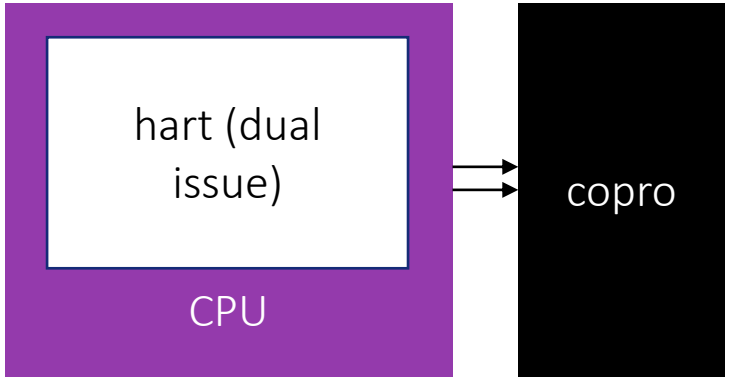
1 CPU with 1 single issue hart



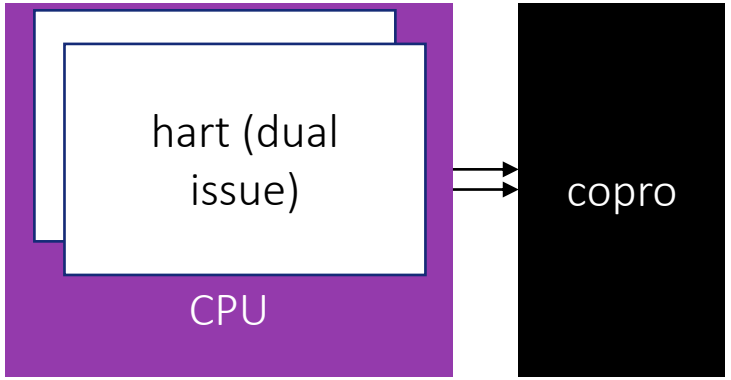
2 CPUs with 1 single issue hart each



1 CPU with 1 dual issue hart



1 CPU with 2 dual issue harts

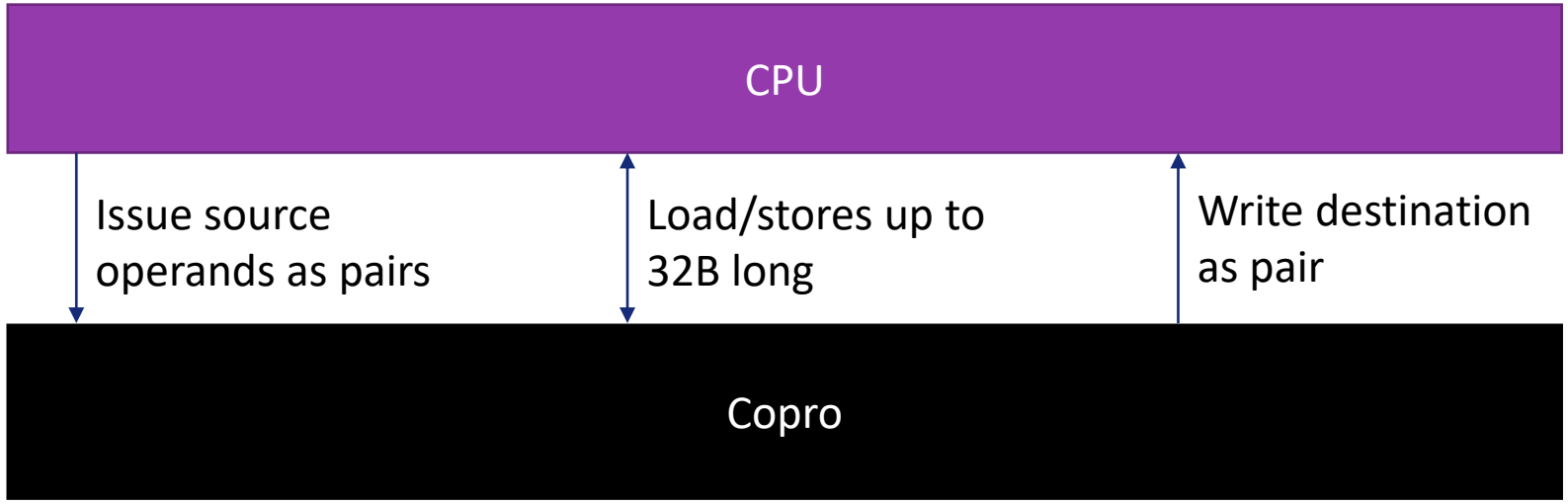


*hart = hardware thread

Copyright © 2023

Register Pair Support

- A register pair is defined as even & odd neighbours in the register file
- Two RISC-V (draft-)standard extensions using register pairs
  - Zdinx: Double precision floating point using x-registers (integer registers)
  - P (Packed SIMD): DSP operations using register pairs
- CV-X-IF is the only Copro Interface for RISC-V to support pairs
- Key feature as accelerators tend to be number crunchers that can exploit SIMD (Single Instruction Multiple Data)



Copyright © 2023

Performance Enhancing Features

- Speculation
  - CV-X-IF allows issuing instructions speculatively to the coprocessor
  - If for any reason (e.g. branch, interrupt), the instruction should not have issued, it can be killed by the coprocessor
- Out of order completion
  - Instructions can complete execution and write back out of order
    - Instructions that cannot cause exceptions can execute “in the background”
  - Instructions are committed in order
- Out of order issue
  - Currently, instructions are issued in order to the coprocessor
  - Feature under discussion for bypassing instructions in read operands stage – i.e. out of order start of execution. Stay tuned

Copyright © 2023

Implementations of Cores & Coprocessors

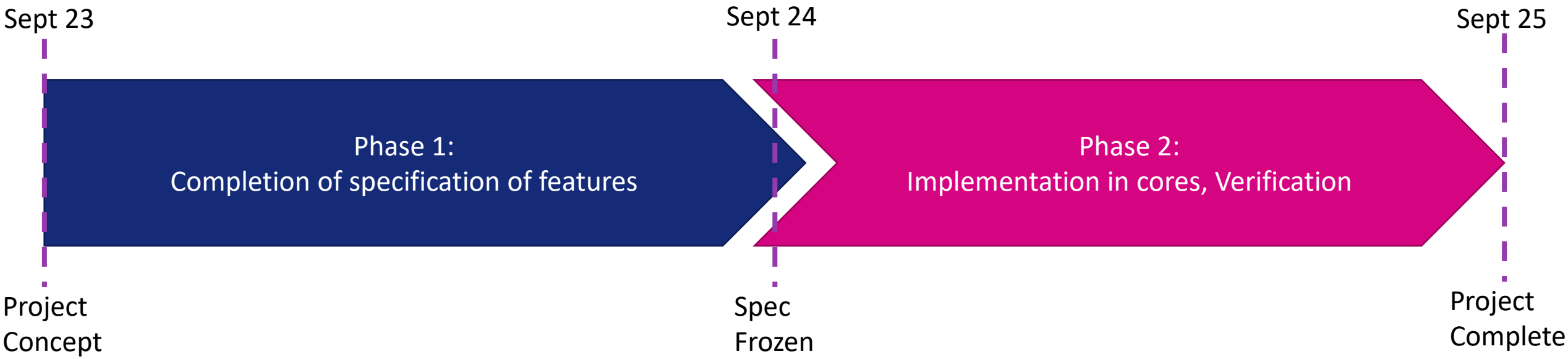
Core	Reference	Owner	Status
CV32E40X	https://github.com/openhwgroup/cv32e40x	OpenHW Group	Implementation complete
CVA6	https://github.com/openhwgroup/cva6	OpenHW Group	Implementation complete
CVE2	https://github.com/openhwgroup/cve2	OpenHW Group	Concept
NOEL-V	https://gaisler.com/noel-v	Gaisler	Planned

Extension	Reference	Owner	Public
Zve32x1p0	Vicuna	TU Graz	Yes
F/D/Q	PULP fpu_ss/CVFPU/Fpnew	OpenHW Group	Yes
Custom posit	POSIT FPU for CV32E40X	Complutense University of Madrid	Not yet
V0p8 subset	AVA (AI Vector Accelerator)	University of Southampton	Yes
Zve64d1p0	Spatz	ETH	No

Copyright © 2023

Future Work

- Features
  - Operands interface – higher performance, OoO issuing
  - Superscalar support
  - 16- and 48-bit instructions handled by the coprocessor
  - Access to floating point registers
- Implementation on CVE2 and other processors



Copyright © 2023

Conclusion

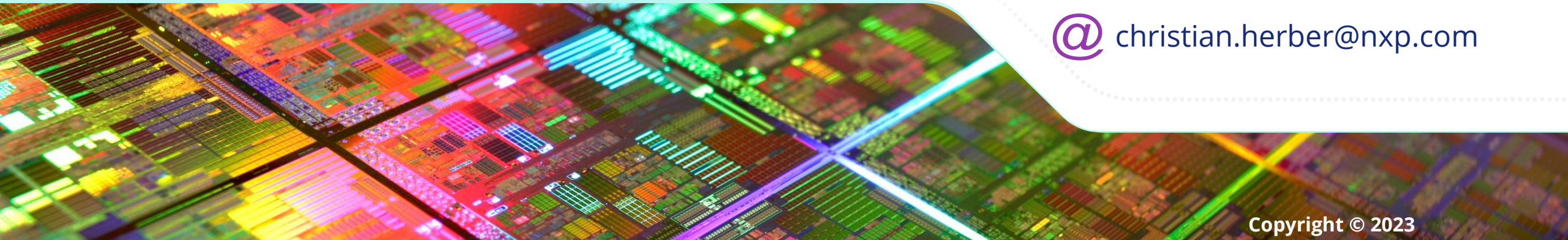
- CV-X-IF is the way to go to accelerate applications like AI/ML on a RISC-V processor
- TRISTAN has brought together several industry players committed to advance and establish CV-X-IF
- CV-X-IF's rich feature set attracts the core suppliers, which attracts the tool vendors, which sets the stage for integrators to implement their custom accelerators

Copyright © 2023

EDGE AI

Thank You

For your attention



@ christian.herber@nxp.com

Copyright © 2023