

TRISTAN

## RISC-V, Custom Instructions, Interfaces

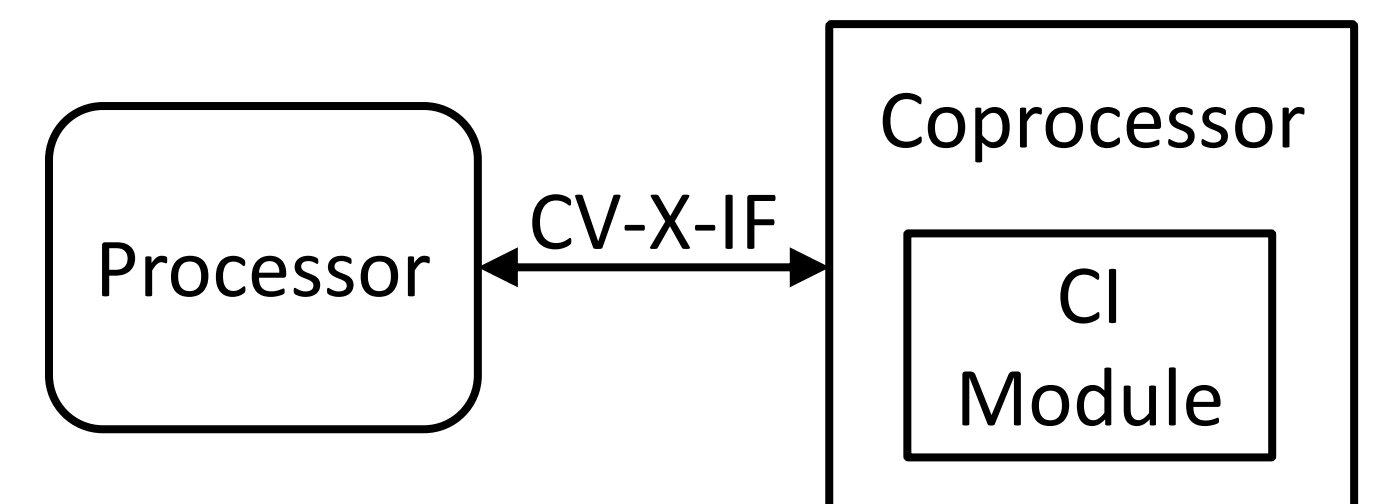
Application-specific **Custom Instructions (CIs)** facilitate efficient RISC-V-based systems while preserving the versatility that standard instructions offer. The integration of CIs in RISC-V processors is a manual, time-consuming process.

In this work, we present a way to **semi-automate the CI development process**. The main feature of our approach is a **configurable wrapper** for CI modules which is provided by the **generation flow** in RTL representation.

Our proposed flow is based on an **interface-based integration** approach. External CI modules are integrated with the processor via an CI interface. This approach allows to use and reuse the generated wrappers with multiple different processors, without changing the processor's internals.

The interface implemented in our wrapper is the **Core-V eXtension interface (CV-X-IF)** from the OpenHW Group. The recently ratified specification is still in active development, and we envision upcoming support from multiple processors and coprocessors.

One goal of utilizing CV-X-IF is to preserve the ability to immediately react to an offloaded instruction. CV-X-IF and our wrapper allow the processor to collect the **result** of an instruction in the **same cycle** in which it was issued.



## CI Integration Flow

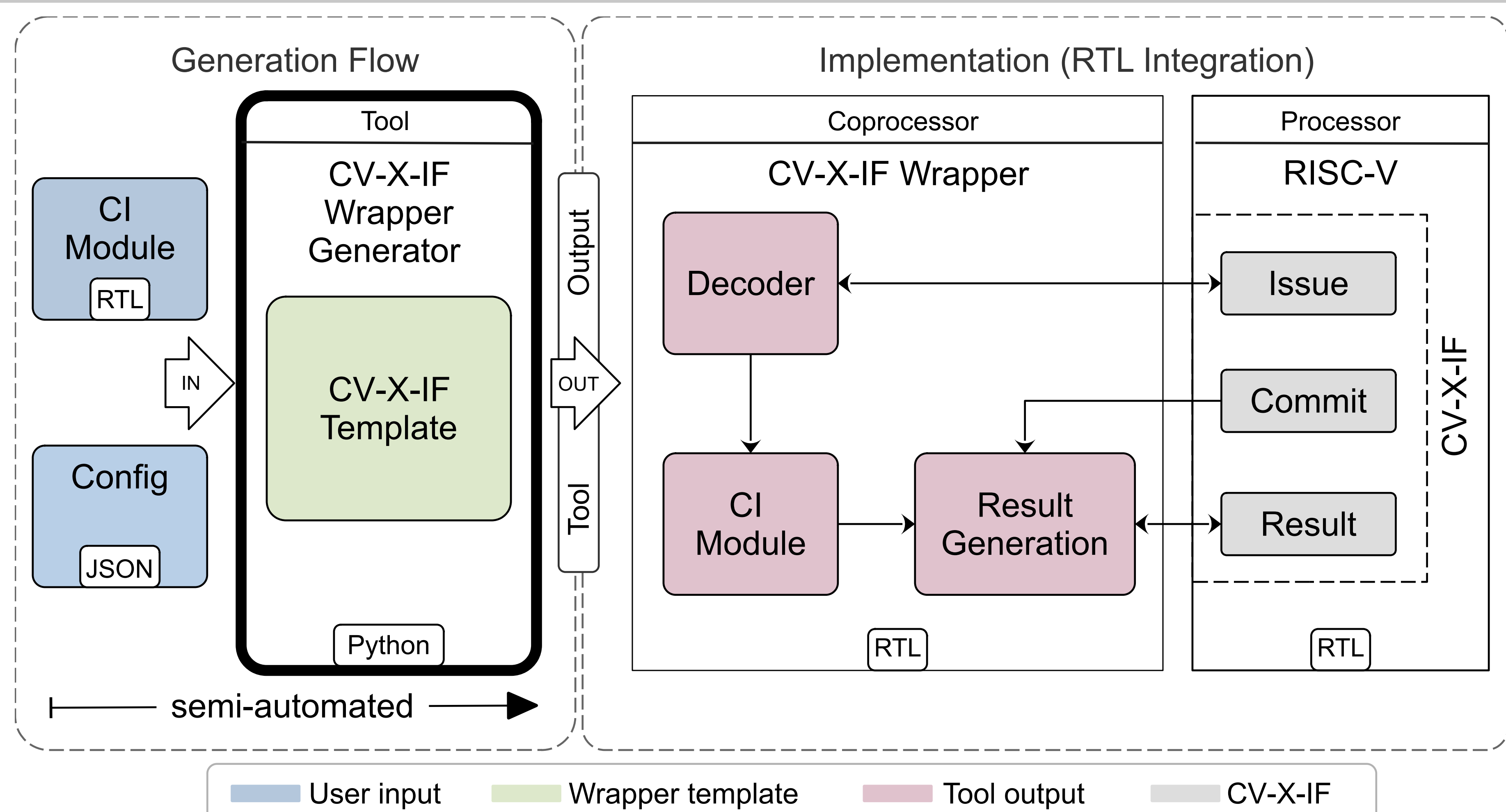
**Inputs** - what the user provides to the flow

- Custom Instructions in form of RTL modules
- Opcodes in .json format

**Output** - generated CI-specific wrapper

- Decoder: Handles instructions offloaded by the processor
- CI Module: User-provided CI execution unit
- Result Generation: Buffers results until the processor collects or dismisses them

The current wrapper version supports **R-type single-cycle CIs**. Therefore, CI modules provided by the user must consist of combinational logic. To realize a **multi-cycle CI**, the workaround is to split it into a sequence of multiple combinational CIs.



## Results - Runtime and Development Time

**Runtime improvements:**

- We validated the flow by generating a CRC and an **AES coprocessor**.
- For AES specifically, we compared the instruction and cycle count against two reference algorithms.
- The resulting reduction in runtime ranged between around **61% to 88%**.

	Algorithm	Metric	
		Instructions	Cycles
	1	88.05%	87.53%
	2	68.26%	60.52%

**Development time reduction:**

- We estimated to which degree the presented flow supports the user in reducing their working hours compared to implementing the CIs manually.
- The estimation is based on an **expert interview**.
- We created three scenarios with use cases of varying complexity.
- We also introduced weights based on the system's reusability and the user's experience level and training time.
- The estimation results yield a reduction in development time of about **64%** when using the proposed flow.

## Conclusion and Outlook

- We proposed a semi-automated **CI integration flow** that generates an RTL wrapper. The support of the Core-V eXtension interface enables rapid execution of issued CIs and compatibility with multiple RISC-V processors.
- We found **runtime improvements** for an AES use case of up to **88%**.
- We attain a **reduction in working hours** of the flow's users by around **64%**.

The results demonstrate the potential benefits of a fully automated CI development framework based on interfacing. In a follow-up action as part of TRISTAN we are currently developing a **High-Level Synthesis (HLS)** flow to add automation support for applications given in C/C++. HLS is leveraged to generate RTL code from high-level C/C++ descriptions of the CIs and the wrapper, and to further optimize for performance, power, and area.

## References

1. M. Damian et al. "SCAIE-V: an open-source SCALable interface for ISA extensions for RISC-V processors". Proc. DAC '22, pp. 169–174.
2. F. Egert. FRANCIS-V: FRAMework for iNtegrating Custom instructionS into RISC-V systems. Master's thesis. 2023.
3. CORE-V eXtension Interface. OpenHW Group, 2021.



TRISTAN



### Acknowledgements

The TRISTAN project, nr. 101095947 is supported by Chips Joint Undertaking (CHIPS-JU) and its members and including top-up funding by the Austrian Research Promotion Agency (FFG) and the program "ICT of the Future" of the Austrian Federal Ministry for Climate Action, Environment, Energy, Mobility, Innovation and Technology (BMK).

