# CIRCE
# CROSS Integrated RISC-V Cryptographic Extension

Alessandra Dolmeta, Valeria Piscopo, Maurizio Martina, Guido Masera
*Department of Electronics and Telecommunications, Politecnico di Torino, Torino, Italy

***Abstract*—Post-Quantum Cryptography (PQC) is moving from algorithm selection to deployment, where performance, energy, and software portability are key constraints, especially on embedded and IoT-class processors. Many PQC schemes stress general-purpose cores with irregular control flow, large arithmetic workloads, and heavy memory traffic. Instruction-set extensions (ISE) and tightly integrated accelerators offer a practical middle ground: they speed up dominant kernels while preserving programmability and avoiding the rigidity of fully fixed-function hardware. In this context, we target post-quantum digital signatures, which remain under active evaluation, including NIST's 2023 call for additional schemes. We focus on CROSS, a code-based signature built from zero-knowledge proofs and the Restricted Syndrome Decoding Problem, and present CIRCE: a RISC-V–integrated extension connected through the Core-V eXtension Interface (CV-X-IF). CIRCE supports both R-SDP and R-SDP(G), runs across all official parameter sets without hardware retuning, and achieves an average 2× speed-up on a Zynq UltraScale+ FPGA with an ultra-compact footprint (down to 800 LUTs / 100 FFs).**

***Index Terms*—RISC-V, PQC, CROSS, CV-X-IF, FPGA.**

## I. Introduction

As Post-Quantum Cryptography (PQC) transitions from algorithm selection to real-world deployment, efficient execution on embedded and IoT-class processors has become a critical challenge. Many PQC schemes impose substantial computational and memory demands that strain general-purpose cores, motivating the use of instruction-set extensions (ISE) and tightly integrated accelerators as a balanced alternative between pure software and rigid fixed-function hardware. While lattice-based constructions dominate current standardization outcomes (ML-KEM, ML-DSA), digital signatures remain under active evaluation. In 2023, NIST launched an additional call explicitly encouraging alternatives to structured lattices. Among the second-round candidates, CROSS (Codes and Restricted Objects Signature Scheme) [1] has emerged. It is a code-based signature scheme derived from Zero-Knowledge (ZK) proof paradigm [2] and the Restricted Syndrome Decoding Problem (R-SDP), offered in two variants: the original R-SDP formulation and the generator-based R-SDP(G), which reduces signature size through structured restrictions. Following NIST requirements, CROSS defines parameter sets targeting security categories 1, 3, and 5, corresponding to AES-128, AES-192, and AES-256 security levels. For each category, three optimization corners are proposed: Fast *(f)*, Balanced *(b)*, and Small *(s)*.

This work was funded by project SERICS (PE00000014) under the MUR National Recovery and Resilience Plan funded by the European Union - NextGenerationEU.

CROSS consists of three core procedures: KeyGen, which derives compact public/secret keys; Sign, which generates a signature via zero-knowledge proofs of restricted syndrome decoding; and Verify, which validates the proof using the public key. To quickly pinpoint the performance-critical routines, we first profiled the reference implementation using the Synopsys ASIP-Designer RISC-V core simulator [3]. This early-stage analysis identified two classes of kernels which dominate the workload: (i) hashing and pseudo-randomness generation based on SHAKE, hence on the underlying KECCAK permutation [4], and (ii) repeated finite-field operations used in syndrome computations and restricted-vector processing ( modular arithmetic over $\mathcal{R}$). Based on these findings, the complete CROSS codebase was then ported and evaluated on a real 32-bit RISC-V SoC, where cycle measurements were refined using hardware performance counters on a 4-stage pipeline.

**Contributions**. In this work, we present CIRCE (CROSS Integrated RISC-V Cryptographic Extension), a tightly coupled accelerator integrated into a RISC-V processor via the Core-V eXtension Interface (CV-X-IF). CIRCE supports both R-SDP and R-SDP(G) without hardware modifications, targeting all official parameter sets while focusing on minimal area overhead to deliver consistent performance gains.

## II. Design

RISC-V has emerged as a common foundation for PQC research, with hundreds of works exploring efficient implementations of PQC algorithms through dedicated RISC-V extensions and accelerators [8]. A key enabler for this ecosystem is the availability of integration mechanisms that reduces the engineering effort of attaching custom hardware to a processor pipeline while maintaining software portability. To this end, the **Core-V eXtension Interface** (CV-X-IF) has been introduced by the OpenHW Group [9].

CV-X-IF provides a standardized path to offload custom instructions to external units with low-latency access to the core register file, avoiding invasive RTL changes in the CPU and minimizing toolchain fragmentation. Recent works highlight the effectiveness of this approach in different application domains, including cryptography [10]–[15].

Within this framework, CIRCE is designed as a tightly coupled RISC-V extension using CV-X-IF, enabling cryptographic operations to be executed through custom instructions while preserving full toolchain compatibility. The accelerator is integrated into the open-source X-HEEP SoC [16], a configurable RISC-V microcontroller platform well suited for embedded experimentation. Custom instructions are invoked

TABLE I: Comparison with state-of-the-art FPGA implementations. Runtimes are in milliseconds; **AT** is the area–time product computed as *KeSlice*× kCC (kilo-Clock Cycles). **KeSlice** for Xilinx 7-Series, † for the UltraScale series.

| Ref. | Design | Resources | | Freq. | KeyGen | | Sign | | Verify | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | LUT/FF/BRAM/DSP | KeSlice | [MHz] | [kCC] | AT | [kCC] | AT | [kCC] | AT |
| [5] | CROSS R-SDP-1-b | 25393/9076/57.0/0 | 13.6 | 108 | 4.212 | 57 | 116.64 | 1,586 | 85.86 | 1,168 |
| | CROSS R-SDP(G)-1-b | 27235/10348/37.0/27 | 11.5 | 119 | 11.071 | 127 | 90.916 | 1,046 | 62.713 | 721 |
| [6] | LESS-L1-b | 54800/39900/59.5/0 | 21.3 | 200 | 29.06 | 619 | 77.54 | 1,652 | 174.5 | 3,717 |
| [7] | Raccoon-128 | 13957/11284/30.5/4 | 7.4 | 222 | 104.642 | 774 | 278.898 | 2,064 | 101.642 | 752 |
| **OURS** | CROSS R-SDP-1-b | 765/106/0/1 | 0.095† | 30 | 468 | 44 | 38,499 | 3,657 | 20,347 | 1,933 |
| | CROSS R-SDP(G)-1-b | 1065/109/0/38 | 0.133† | | 234 | 29 | 26,692 | 3,337 | 17,439 | 2,180 |

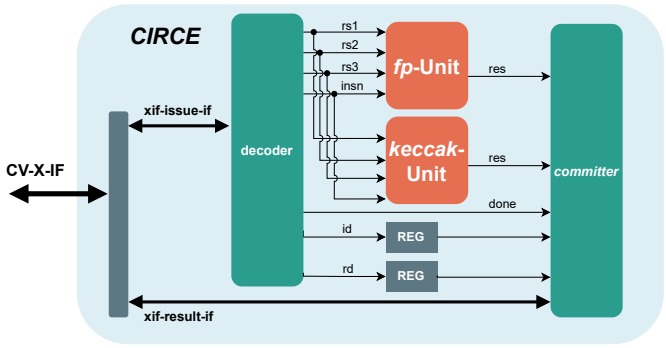


Fig. 1: CIRCE schematic integrated via CV-X-IF.

from C via inline assembly, allowing the CROSS kernels to be accelerated transparently within the original software structure. Figure 1 illustrates the high-level organization of CIRCE. The architecture follows a modular structure comprising an instruction decoder, a set of computation units, and a commit stage interfacing with the core. From a software perspective, accelerated operations behave as regular RISC-V instructions.

Two computation blocks are exposed.

- First, a lightweight *keccak*-unit targets the SHAKE/SHA-3 core permutation, i.e., KECCAK-$f$[1600] [4], which updates a $5 \times 5$ state of 64-bit lanes over 24 rounds; in software, its cost is dominated by 64-bit rotations and simple non-linear Boolean patterns. Since the target platform is 32-bit, CIRCE accelerates these hotspots through fine-grained custom instructions, including two 32-bit rotation primitives operating on the high/low halves of a 64-bit word (with the offset provided in a third register) and a fused ternary operation of the form $R = A \oplus (\neg B \wedge C)$, reducing both instruction count and latency.
- Second, a compact arithmetic unit accelerates the modular operations over $\mathcal{R}$ used in syndrome decoding and restricted-vector processing, by providing custom instructions for the dominant multiply–accumulate and reduction kernels. The same instruction set and integration logic are reused for both R-SDP and R-SDP(G), ensuring architectural stability across variants.

## III. Results

CIRCE was implemented on a Zynq UltraScale+ ZCU104 FPGA (XCZU7EV) and integrated into X-HEEP, operating at 30 MHz. This frequency matches the SoC operating constraints during software/hardware bring-up; higher targets were not retained due to instability observed during JTAG-based debugging. The accelerator requires no BRAM resources, and its footprint remains extremely small: 765/106/0/1 (LUT/FF/BRAM/DSP) for the R-SDP variant and 1065/109/0/38 for R-SDP(G), corresponding to 0.095 and 0.133 KeSlice (UltraScale), respectively [17].

Across the full set of official CROSS parameter sets, CIRCE provides consistent speed-ups close to 2× for KeyGen, Sign, and Verify, without any hardware retuning between security categories. These gains are conservative, since memory transfers and non-kernel software overheads (e.g., data movement across function boundaries) remain in software.

Table I compares our design against representative FPGA implementations from the literature. While monolithic accelerators can achieve lower cycle counts by hardwiring large portions of the protocol, they typically do so at a much higher area cost. To capture this trade-off, we report the area–time product (AT) in KeSlice×kCC. Under this metric, CIRCE achieves the best AT for KeyGen, reflecting the impact of accelerating the hashing-dominated workload with minimal hardware. For Sign and Verify, our absolute cycle counts are higher than those of large accelerators, yet the very low KeSlice keeps the AT competitive (e.g., within a small-factor range of [5] and [6]), while being orders of magnitude smaller in area. Additional signature on-ramp accelerators (e.g., SDitH, MEDS, and Raccoon) follow a similar trend: they reduce latency through larger dedicated datapaths, but their significantly higher area typically leads to worse or comparable AT values for constrained deployments ( [18], [19]). Overall, the results confirm that a tightly coupled ISE can deliver meaningful end-to-end acceleration for PQC signatures while remaining feasible for IoT-class devices.

## IV. Conclusions

This work demonstrates that tightly coupled ISE are an effective and scalable approach for accelerating post-quantum digital signatures on constrained RISC-V platforms. By targeting dominant computational kernels while preserving software flexibility, CIRCE achieves a favorable trade-off between performance, area, and crypto agility.

Beyond being the first hardware implementation of CROSS, CIRCE shows that even complex zero-knowledge–based code-based signatures can be supported with negligible hardware cost. The use of CV-X-IF enables portability across cores and resilience to algorithm evolution, making the proposed approach well suited for emerging PQC standards and long-term deployment in embedded systems.

## REFERENCES


[1] M. Baldi, S. Bitzer, A. Pavoni, P. Santini, A. Wachter-Zeh, and V. Weger, "Zero knowledge protocols and signatures from the restricted syndrome decoding problem." Cryptology ePrint Archive, Paper 2023/385, 2023.

[2] S. Underwood, "The power and potential of zero-knowledge proofs," *Commun. ACM*, vol. 68, p. 11–13, July 2025.

[3] Synopsys, "ASIP Designer." https://www.synopsys.com/dw/ipdir.php?ds=asip-designer. Accessed: 2025-09-12.

[4] National Institute of Standards and Technology (U.S.), "SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions," NIST FIPS 202, National Institute of Standards and Technology, 2015.

[5] P. Karl, F. Antognazza, A. Barenghi, G. Pelosi, and G. Sigl, "High-Performance FPGA Accelerator for the Post-quantum Signature Scheme CROSS." Cryptology ePrint Archive, Paper 2025/1161, 2025.

[6] L. Beckwith, R. Wallace, K. Mohajerani, and K. Gaj, "A high-performance hardware implementation of the less digital signature scheme," in *Post-Quantum Cryptography* (T. Johansson and D. Smith-Tone, eds.), (Cham), pp. 57–90, Springer Nature Switzerland, 2023.

[7] Z. Ni, A. Khalid, Z. Zhang, Y. Cui, W. Liu, and M. O'Neill, "Hraccoon: A high-performance configurable sca resilient raccoon hardware accelerator," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, vol. 2025, p. 413–436, Jun. 2025.

[8] D.-T. Dam *et al.*, "A Survey of Post-Quantum Cryptography: Start of a New Race," *Cryptography*, vol. 7, no. 3, p. 40, 2023. Includes discussion of RISC-V based PQC accelerators.

[9] OpenHW Group, "Core-V eXtension Interface (CV-X-IF) Documentation." https://docs.openhwgroup.org/projects/openhw-group-core-v-xif/en/latest/intro.html, 2025. Accessed: 2025-08-25.

[10] C.-A. Popovici, A. Stan, N.-A. Botezatu, and V.-I. Manta, "RiscADA: RISC-V Extension for Optimized Control of External D/A and A/D Converters," *Electronics*, vol. 14, no. 15, 2025.

[11] A. Dolmeta, M. Martina, and G. Masera, "Exploring the New CV-X-IF Interface to Customize RISC-V Instruction Sets: A Case of Study in Cryptography," in *Applications in Electronics Pervading Industry, Environment and Society* (M. Ruo Roch, F. Bellotti, R. Berta, M. Martina, and P. Motto Ros, eds.), (Cham), pp. 39–47, Springer Nature Switzerland, 2025.

[12] A. Dolmeta, S. Di Matteo, E. Valea, M. Carmona, A. Loiseau, M. Martina, and G. Masera, "TYRCA: A RISC-V Tightly-Coupled Accelerator for Code-Based Cryptography," in *2025 Design, Automation & Test in Europe Conference (DATE)*, pp. 1–7, 2025.

[13] K. Hepola, T. R. Arachchige, J. Multanen, and P. Jääskeläinen, "Fully Automatic Compiler Retargeting and CV-X-IF Hardware Interface Generation for RISC-V Custom Instructions," in *2024 IEEE Nordic Circuits and Systems Conference (NorCAS)*, pp. 1–7, 2024.

[14] L. Waucquez and A. Rodriguez, "EROS: Extensible Reliable Offloading Solution," in *Proceedings of the 22nd ACM International Conference on Computing Frontiers: Workshops and Special Sessions*, CF '25 Companion, (New York, NY, USA), p. 82–85, Association for Computing Machinery, 2025.

[15] J. Lee, W. Kim, and J.-H. Kim, "A Programmable Crypto-Processor for National Institute of Standards and Technology Post-Quantum Cryptography Standardization Based on the RISC-V Architecture," *Sensors*, vol. 23, no. 23, 2023.

[16] S. Machetti, P. D. Schiavone, T. C. Müller, M. Peón-Quirós, and D. Atienza, "X-HEEP: An Open-Source, Configurable and Extendible RISC-V Microcontroller for the Exploration of Ultra-Low-Power Edge Accelerators," 2024.

[17] AMD Xilinx, *UltraScale Architecture Configurable Logic Block*. AMD, v1.9 ed., Dec. 2022. https://docs.xilinx.com/r/en-US/ug574-ultrascale-clb.

[18] S. Deshpande, J. Howe, J. Szefer, and D. Yue, "SDitH in hardware." Cryptology ePrint Archive, Paper 2024/069, 2024.

[19] S. Deshpande, Y. Lee, M. Nawan, K. Nawaz, R. Niederhagen, Y. Paek, and J. Szefer, "Unified MEDS accelerator." Cryptology ePrint Archive, Paper 2025/796, 2025.