A RISC-V based accelerator for Post Quantum Cryptography

Ambily Suresh[1]*, Andrew Wilson[1], Diego Gigena-Ivanovich[2], Manuel Freiberger[1], and Willibald Krenn[1]

Abstract

Post-Quantum Cryptography (PQC) is a topic of increased interest in the past decade, both with regards to the cryptosystem definition and the hardware and software implementations to perform at optimum efficiency. We present our ongoing work on the implementation of RISC-V based accelerators for PQC algorithms, in particular the Classic McEliece Key Encapsulation Mechanism. Our system includes a PQC accelerator and an Open-HW Group CVA-6 core along with a PQC-specific instruction set. This presentation describes the architecture, performance estimates, and demonstration plans in the future.

Introduction

Modern digital communications rely on three important cryptographic functions for the secure transfer of information - Public Key Encryption (PKE), Digital Signatures, and Key Exchange. Of particular interest is PKE, which provides secure communications without prior agreement on a secret key. Current PKE algorithms rely on mathematical problems which are "assumed hard to solve", such as prime factorisation, discrete logarithms, and elliptic curve solving. Following the development of Shor's and Grover's algorithms, along with the developments in quantum computing over the years, the search computation for these classic PKE problems would receive a significant speed-up. This would make the central mathematical problems more easily solvable (less time-complex), making the existing algorithms no longer secure.

Post Quantum Cryptography (PQC) involves the development of security algorithms and cryptosystems that are secure against attacks which exploit the advantages provided to processing performed by quantum computers. PQC algorithms fall into two categories of problems: the "Learning with Errors" problem for Lattice-based algorithms and the Syndrome Decoding problem for the Code-based algorithms.

As part of various Chips Joint Undertaking (Chips-JU)-funded projects, we are working on the development of a number of open source accelerators – with a special focus on cryptography and neural networks and with the final goal of developing a safe and secure, federated learning system. Implementing quantum safe, cryptographic algorithms on an FPGA or ASIC based platform is crucial to establishing secure communications in such a system. This paper will talk about the system architecture for the RISC-V processor and accelerator along with the software (SW) development and hardware (HW) verification plans.

*Corresponding author: ambily.suresh@silicon-austria.com

Hardware Architecture

We have currently picked the Classic McEliece (CM) Key Encapsulation Method (KEM) [1], which is a code-based system and a finalist in the National Institute of Standards and Technology (NIST)'s efforts to select and standardise PQC algorithms[1]. The system has shown resilience for high-security applications such as military and space so far. We base our overall HW design on previous FPGA based implementations of the CM cryptosystem [2] and primitives, aiming to accelerate the key functions involved at multiple points in the execution of CM. This gives more flexibility in which algorithms and security levels can be implemented, and easier modification of the overall functionality of the system.

Our current implementation has selected the Number Theoretic Transform (NTT) as the key primitive to accelerate. This is a PQC operation that exploits the convolution theorem and provides a "linearithmic" ($O(n \log n)$ from polynomial $O(n^2)$) time-complexity speed-up to commonly executed polynomial multiplications. One key advantage of accelerating the NTT operation is that it is useful in other non-PQC computations such as error-correction and Homomorphic Encryption (HE), while also providing better performance for the symmetric-key portion of lattice-based communications.

A major task in our work is the integration of the accelerator with a suitable RISC-V core and extending the instruction set (ISA) for the accelerator operations. Our system incorporates multiple open source modules, the main one being the Open-HW Group's[2] CVA6 RISC-V core. The core also gives us the option to use the the Core-V-eXtensible-InterFace (CV-X-IF), to transfer any special instructions to the co-processor. It provides native support for the implementation of

[1] https://csrc.nist.gov/projects/post-quantum-cryptography/
[2] https://www.openhwgroup.org/

the AXI-4 bus[3] and has plenty of flexibility in configuration parameters, as a result of which it has received a significant amount of attention from the community. Currently the PQC implementation is used as a memory mapped accelerator via the AXI bus, as shown in Fig. 1, while optimising the SW to come up with specific instructions for a co-processor implementation via the CV-X-IF interface. One final module is a memory management unit for the larger-keys to boost efficiency and security in key storage and retrieval.

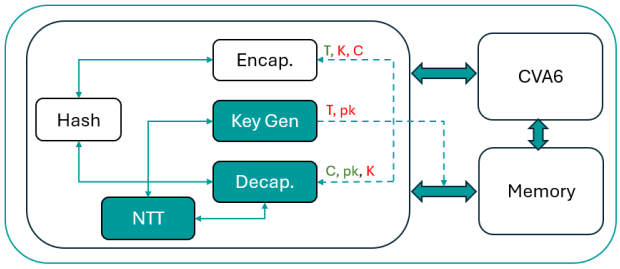

Figure 1: *Architecture of our RISC-V based Classic McEliece PQC system. The modules developed as part of this work are highlighted in dark/turquoise with other open source modules in white. The inputs and outputs from the memory to the PQC modules are marked in green and red respectively. C is the cipher texts, K is the session key, T is the public key, pk is the private key which consists of (δ, c, g, α, s) [1]*

The CM cryptosystem defines three main mathematical functions [1]:

1. KEYGEN – generates the public and private key pairs from random bits
2. ENCAP – generates a cipher text and session key from a public key and random bits
3. DECAP – outputs a session key when given a cipher text and a private key.

We reuse the basic frame works for the three main modules from the open source implementation in [2], and benchmark the performance for an AMD Virtex™ Ultrascale™ FPGA (xcvu37p). While the original implementation is parameterised to generate multiple KEMs, we currently target only the mceliece348864 parameter set [1], and the first estimates of the performance are summarised in Table. 1.

module	LUTs	latency (ms)	time × area
encap	977	0.14	0.13
decap	17109	0.16	2.74
keygen	26674	1.16	30.94

Table 1: *Performance evaluation based on mceliece348864 parameter set and VCU128 implementation. We are aiming to parameterise and optimise an NTT implementation to suit multiple KEM/parameter sets.*

[3] https://github.com/pulp-platform/axi

Firmware and Software

Our SW and firmware design includes a modular compiler that not only optimises the SW to specific architectures but also ensures compatibility across heterogeneous HW. We use the RISC-V tool-chains from the Open-HW group along with the CVA6 core to test the RISC-V core and peripherals, with on-going efforts to introduce and optimise PQC specific instructions in the ISA. Open-source projects offer a robust suite of peripherals and interface bridges designed to facilitate the development, simplifying the design and verification process. We plan to exploit the Multi-Level Intermediate Representation (MLIR) framework[4], as it provides a flexible approach to compilation with a gradual lowering of code abstraction to object code, while enabling optimisations at each stage, making it well suited for the heterogeneous landscape of RISC-V HW.

Summary

This paper gives an overview of our PQC accelerator for the CM cryptosystem. We are performing functional simulations and optimisations on the overall HW design to determine the range of parameters and coefficients to be accelerated by the NTT module. The complete system including the CVA6 processor will be implemented and tested on an AMD Virtex™ UltraScale™ VCU128 Evaluation Kit. Our next milestone would be to tape out a quantum-safe SoC to be used in test demonstrators, primarily targeted for automotive applications.

Acknowledgements

This research received funding from the Austrian Research Promotion Agency and the Austrian ministry for Climate Action, Environment, Energy, Mobility, Innovation and Technology under project FO999899263 the Chips Joint Undertaking and its members Austria, Czechia, France, Germany, Italy, Romania, Spain, Sweden, Switzerland under the ISOLDE project (no. 101112274)

References

[1] Martin R. Albrecht et al. *The Classic McEliece Public Key Cryptosystem; Cryptosystem specification, Design rationale, and NIST Round 4 Submission Overview*, Oct. 2022.

[2] Po-Jen Chen et al. Complete and improved fpga implementation of classic mceliece. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, page 71–113, Jun. 2022.

[4] https://mlir.llvm.org/