Implementation and integration of NTT/INTT accelerator on RISC-V for CRYSTALS-Kyber

Implementation and integration of NTT/INTT accelerator on RISC-V for CRYSTALS-Kyber

Alessandra Dolmeta
alessandra.dolmeta@polito.it
Politecnico di Torino
Torino, Italy

Emanuele Valpreda
emanuele.valpreda@polit.it
Politecnico di Torino
Torino, Italy

Maurizio Martina
Guido Masera
name.surname@polito.it
Politecnico di Torino
Torino, Italy

ABSTRACT

This paper presents a comprehensive study on the implementation of a memory-mapped accelerator designed for Number Theoretic Transform (NTT) and Inverse Number Theoretic Transform (INTT) operations within the context of the post-quantum cryptographic algorithm CRYSTALS-Kyber. The primary focus lies in the performance evaluation of the algorithm, with a particular emphasis on minimizing the overhead associated with transferring data between the core and the implemented IP. The analysis includes a deep dive into the intricacies of data transfer, leveraging Direct Memory Access (DMA) to efficiently reduce overhead. The evaluation results show that our implementation can achieve up to 15.7× and 19.6× improvement in cycle count for NTT and INTT respectively, compared to the base software implementation. To this end, we also demonstrate the efficacy of the proposed memory-mapped accelerator in enhancing the overall performance of CRYSTALS-Kyber, thereby contributing to the advancement of secure cryptographic systems in the post-quantum era.

CCS CONCEPTS

• **Hardware**; • **Security and privacy** → **Symmetric cryptography**;

KEYWORDS

Do, Not, Us, This, Code, Put, the, Correct, Terms, for, Your, Paper

ACM Reference Format:
Alessandra Dolmeta, Emanuele Valpreda, Maurizio Martina, and Guido Masera. 2024. Implementation and integration of NTT/INTT accelerator on RISC-V for CRYSTALS-Kyber. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation email (Conference acronym 'XX).* ACM, New York, NY, USA, 3 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

In the realm of post-quantum cryptographic environments, *pure* hardware implementations present formidable performance solutions but are accompanied by drawbacks such as reduced flexibility, portability challenges, and an escalating area overhead. Conversely, a *pure* software approach emerges as a more accessible option in terms of development and maintenance, offering heightened flexibility and portability albeit at the expense of lower performance. In this post-quantum context, the pursuit of hybrid solutions becomes crucial, leveraging the distinct advantages of both hardware and software and forming the foundation for robust co-design techniques that address the unique demands of cryptographic systems in the post-quantum era. In this work, we present the architecture of an accelerator designed for NTT and INTT arithmetic operations, integrated into RISC-V. The accelerator is connected to the microcontroller thanks to a hardware/software interface and tested with CRYSTALS-Kyber, one of the first standardized PQC algorithms by NIST [10]. The implementation of this work is publicly available on GitHub. [1].

2 PRELIMINARIES

Lattice-based cryptography is a promising class of algorithms for post-quantum cryptography. A widely recognized lattice problem employed in the construction of cryptographic primitives is the Learning with Error (LWE) problem, along with its algebraic variations, the Ring-Learning with Error (RLWE) and the Module-Learning with Error (MLWE) problems. Numerous studies and analyses in the literature consistently emphasize the critical role played by hash functions and polynomial multiplication. First, LWE problems require sampling of random polynomials. On that account, most lattice-based PQC schemes rely on the hash functions defined in the SHA-3 standard. Whereas, polynomial arithmetic is executed within polynomial rings denoted as R_q. The conventional method of computing the product of two polynomials within this polynomial ring, employing the schoolbook multiplication approach, results in a computational complexity of $O(n^2)$. Numerous strategies exist to mitigate the complexity of polynomial multiplication. Notably, the NTT-based approach has gained favor among cryptographers, as it diminishes the complexity from $O(n^2)$ to $O(nlog(n))$. Two distinct approaches for computing the NTT are the Cooley-Tukey (CT) [2] and the Gentleman-Sande (GS) [4] algorithms. The two butterfly units are reported in Figure 1. The fundamental components of both algorithms are the so-called butterfly operations, involving straightforward arithmetic in Z_q.

To circumvent additional bit-reversal steps, it is a common practice to employ CT butterflies for the transformation into the NTT domain and GS butterflies for the reverse transformation. The illustration in Figure 1 delineates these butterfly operations. ω factor represents a root-of-unity for the polynomial and coefficient ring,

Conference acronym 'XX, June 03–05, 2018, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

[1] REPO

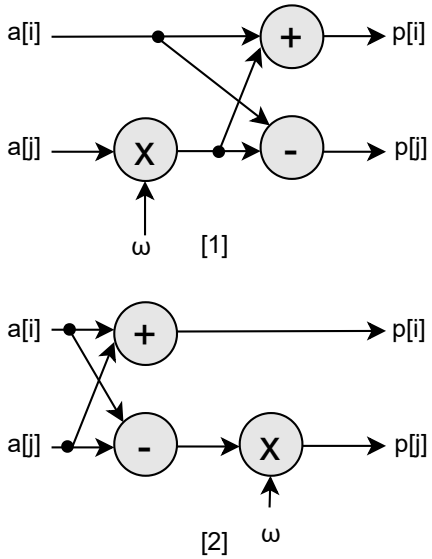

Figure 1: CT and GS Butterflies Configuration

often referred to as the twiddle factor. CRYSTALS-Kyber is a Key Encapsulation Mechanism lattice-based cryptosystem, which is constructed on the hardness of solving the Module Learning-with-Error (Module-LWE) problem. It will be exploited as tester for our accelerator.

3 DESIGN

3.1 Implementation

The hardware implementation distinguishes itself through a shared mathematical foundation, utilizing a unified butterfly unit (BU) designed to accommodate both NTT and INTT. The core disparity lies in data storage and retrieval mechanisms, as well as the execution nuances of the butterfly operation. This operation involves multiplication by a specific constant, known as the Twiddle Factor (ω in Figure 1), followed by addition and subtraction within the polynomial ring $Z_q[X]$. Twiddle Factors represent powers of $\omega_n \in Z_q[X]$, serving as the n-th root of unity, and are precomputed and stored in a BROM memory.

The architecture includes two dual-port Block RAMs (BRAMs), to store input and output polynomials. Input multiplexers facilitate the storage of input polynomials in BRAMs before any operation. Subsequently, the control unit commences operations based on start signals, and the resultant polynomial is retrieved using output multiplexers post-operation. In the case of CRYSTALS-Kyberm, both NTT and INTT undergo a segmentation into seven stages, each comprising 128 butterfly operations. Their computational structures diverge, with NTT employing Cooley-Tukey (CT) butterflies and INTT utilizing Gentleman-Sande (GS) structures. The former follows a decimation-in-time approach ($a + b \cdot \omega_n (\mathrm{mod}; q)$ and $a - b \cdot \omega_n (\mathrm{mod}; q)$), while the latter adheres to a decimation-in-frequency approach ($a + b (\mathrm{mod}; q)$ and $(a - b) \cdot \omega_n (\mathrm{mod}; q)$).

3.2 Integration

The primary component of our system is the Processing System, executing the software application and connected to other system elements. **X-HEEP**[8] (eXtendable Heterogeneous Energy-Efficient Platform) has been used, which is a RISC-V microcontroller described in SystemVerilog that can be configured to target tiny platforms as well as extended to support accelerators. X-HEEP provides a simple customized MCU that can be easily extended with your accelerator without modifying the MCU, but just instantiating it in your design. In Figure 2, a simplified version of XHEEP is reported, showing the element of interest.

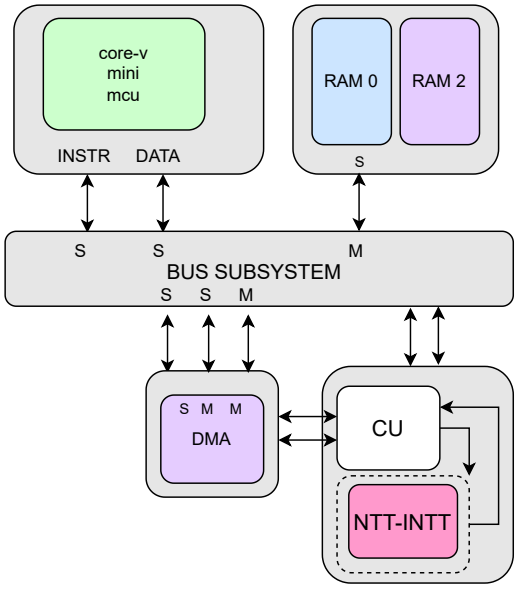

Figure 2: Simplified Architecture Schematic

The second component is the accelerator, which is specialized in polynomial multiplication. Complementing this is the Direct Memory Access (**DMA**) module, facilitating high-throughput communication between the XHEEP and the accelerator for polynomial coefficient data transfers. This is configured by XHEEP, which sets properly all the configuration registers of the DMA, and start/end address, configuring the direction, and initiating data transfers using a proper driver. This arrangement ensures efficient data transfers utilizing the 32-bit data AXI bus structure.

4 RESULTS

Our goal is to evaluate and understand the improvements in cryptographic performance achieved through the incorporation of the accelerator. In pursuit of this objective, we initially measured the clock cycles needed for individual computations of the two distinct functions. Subsequently, we subjected the entire PQC algorithm to testing. Notably, we conducted two analyses: the first without utilizing the DMA, and the second incorporating it.

As demonstrated in Table 1, the version leveraging the DMA (Direct Memory Access) exhibited notable advantages over the non-DMA version. It is worth noting that the choice of utilizing

Table 1: Cycle counts for RISC-V XHEEP platform

Function	Software	HW - w/o DMA	HW - w DMA
NTT	≈24,000	9,105	1,531
		×2.6	×**15.7**
INTT	≈30,000	9,105	1,531
		×3.3	×**19.6**

DMA is context-dependent and hinges on the volume of data to be transferred. In instances where the polynomial-size is substantial, as is the case here, leveraging the DMA proves advantageous. The DMA's efficiency in streamlining data transfers between memory and the accelerator reduces CPU involvement in data movement, minimizing latency, and contributing significantly to the observed improvement in cryptographic performance.

Table 2 shows the cycle count results for all three levels of security of CRYSTALS-Kyber on X-HEEP RISC-V platform. The three algorithms reported refer respectively to NIST-level of security 1, 3, and 5.

Table 2: Cycle counts for RISC-V X-HEEP platform

		Reference	Accelerated	Speed-up
Kyber512	KeyGen	919,896	xx	xx
	Encaps	1,021,339	xx	x
	Decaps	1,219,633	xx	xx
Kyber768	KeyGen	1,498,546	xx	x
	Encaps	1,660,861	xx	x
	Decaps	1,921,454	xx	x
Kyber1024	KeyGen	2,308,802	xx	x
	Encaps	2,501,099	xx	x
	Decaps	2,837,356	xx	x

The reference code is executed on the original SoC, while the accelerated one enables NTT/INTT accelerators.

5 COMPARISON

The results presented in Table 3 showcase a comparative analysis of various methods for NTT and INTT operations on different RISC-V devices. Notably, a significant portion of the referenced works employ tightly-coupled accelerators, directly integrated into the processor architecture.

One remarkable observation is that despite being a memory-mapped accelerator, our approach exhibits competitive performance. This stands out, considering that most other implementations leverage tightly coupled architectures known for their efficiency in cryptographic computations. Our accelerator surpasses expectations by leveraging the advantages of DMA (Direct Memory Access) and utilizing external BRAM (Block Random Access Memory) for storing the polynomial coefficients. These components collectively contribute to the efficient handling of large polynomials, resulting in comparable or even faster performance compared to tightly-coupled counterparts. This highlights the efficacy of our design in achieving accelerated cryptographic operations while operating in a memory-mapped configuration.

Reference	Device	NTT	INTT
[3]	RISC-V (PULPino)	1,935	1,930
[1]	RISC-V (VexRiscv)	6,868	6,867
[7]	RISC-V (Hummingbird)	4,189	3,481
[6]	CVA6 (XIF)	13,880	-
[11]	RISC-V (Ibex)	1,454	1,726
[9]	RISC-V (CVA6)	18,488	18,488
[5]	RISC-V (PicoRV32)	43,756	-
OURS	**RISC-V (XHEEP)**	**1,531**	**1,531**

Table 3: Comparison of methods on different RISC-V devices for NTT-INTT operations

6 CONCLUSIONS

Accelerators play a crucial role in fortifying post-quantum cryptography, addressing the computational challenges posed by quantum computing threats. The importance of diverse implementations becomes evident, emphasizing the need to tailor accelerators to specific algorithms and use cases. Navigating the post-quantum cryptographic landscape demands a nuanced and adaptive approach, ensuring the development of robust systems in the face of evolving threats.

REFERENCES

[1] Erdem Alkim, Hülya Evkan, Norman Lahr, Ruben Niederhagen, and Richard Petri. 2020. ISA Extensions for Finite Field Arithmetic - Accelerating Kyber and NewHope on RISC-V. Cryptology ePrint Archive, Paper 2020/049. https://eprint.iacr.org/2020/049 https://eprint.iacr.org/2020/049.

[2] James W Cooley and John W Tukey. 1965. An algorithm for the machine calculation of complex Fourier series. *Math. Comp.* 19, 90 (1965), 297–301.

[3] Tim Fritzmann, Georg Sigl, and Johanna Sepúlveda. 2020. RISQ-V: Tightly Coupled RISC-V Accelerators for Post-Quantum Cryptography. *IACR Transactions on Cryptographic Hardware and Embedded Systems* 2020, 4 (Aug. 2020), 239–280. https://doi.org/10.13154/tches.v2020.i4.239-280

[4] W Morven Gentleman and Gordon Sande. 1966. Fast Fourier transforms: for fun and profit. In *Proceedings of the November 7-10, 1966, Fall Joint Computer Conference*. 563–578.

[5] Emre Karabulut and Aydin Aysu. 2020. RANTT: A RISC-V Architecture Extension for the Number Theoretic Transform. In *2020 30th International Conference on Field-Programmable Logic and Applications (FPL)*. 26–32. https://doi.org/10.1109/FPL50879.2020.00016

[6] Jihye Lee, Whijin Kim, and Ji-Hoon Kim. 2023. A Programmable Crypto-Processor for National Institute of Standards and Technology Post-Quantum Cryptography Standardization Based on the RISC-V Architecture. *Sensors* 23, 23 (2023). https://doi.org/10.3390/s23239408

[7] Lu Li, Guofeng Qin, Yang Yu, and Weijia Wang. 2024. Compact Instruction Set Extensions for Kyber. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 43, 3 (2024), 756–760. https://doi.org/10.1109/TCAD.2023.3327104

[8] Simone Machetti, Pasquale Davide Schiavone, Thomas Christoph Müller, Miguel Peón-Quirós, and David Atienza. 2024. X-HEEP: An Open-Source, Configurable and Extendible RISC-V Microcontroller for the Exploration of Ultra-Low-Power Edge Accelerators. arXiv:2401.05548 [cs.AR]

[9] Pietro Nannipieri, Stefano Di Matteo, Luca Zulberti, Francesco Albicocchi, Sergio Saponara, and Luca Fanucci. 2021. A RISC-V Post Quantum Cryptography Instruction Set Extension for Number Theoretic Transform to Speed-Up CRYSTALS Algorithms. *IEEE Access* 9 (2021), 150798–150808. https://doi.org/10.1109/ACCESS.2021.3126208

[10] NIST 2022. . Retrieved January 25, 2024 from https://csrc.nist.gov/News/2022/pqc-candidates-to-be-standardized-and-round-4

[11] Tobias Stelzer, Felix Oberhansl, Jonas Schupp, and Patrick Karl. 2023. Enabling Lattice-Based Post-Quantum Cryptography on the OpenTitan Platform. In *Proceedings of the 2023 Workshop on Attacks and Solutions in Hardware Security* (, Copenhagen, Denmark,) *(ASHES '23)*. Association for Computing Machinery, New York, NY, USA, 51–60. https://doi.org/10.1145/3605769.3623993

Received 20 February 2024; revised 12 March 2024; accepted 5 June 2024