

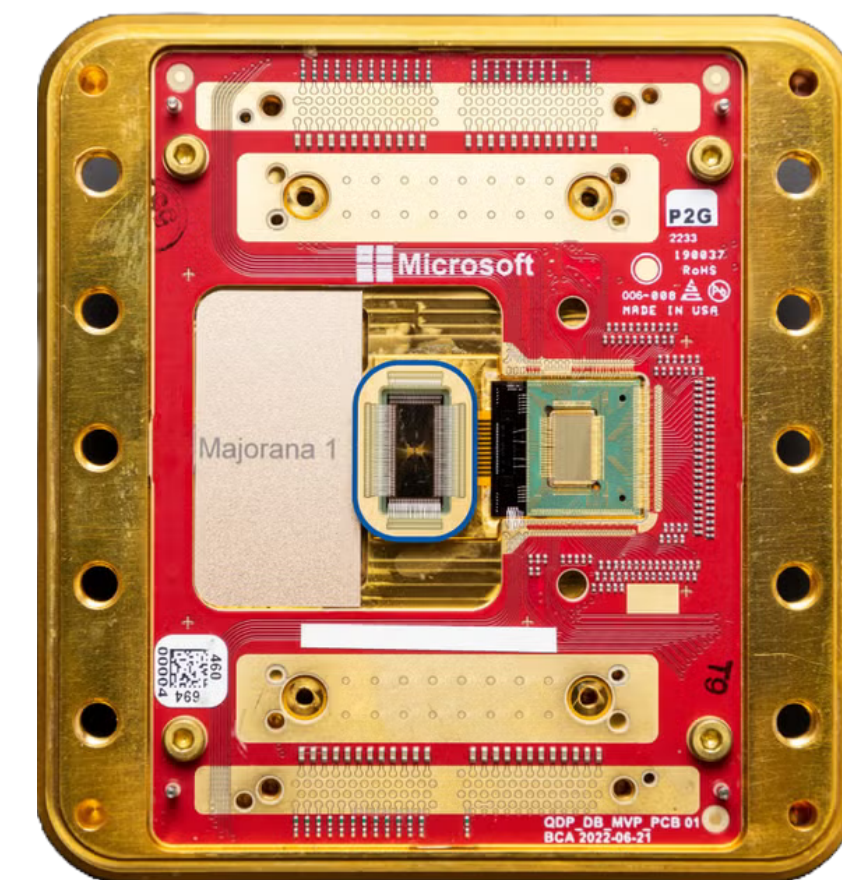
ParallelNTT: Maximizing Performance of Forward and Inverse NTT on FPGA for ML-DSA and ML-KEM

Bardia Taghavi, Reza Azarderakhsh & Mehran Mozaffari Kermani

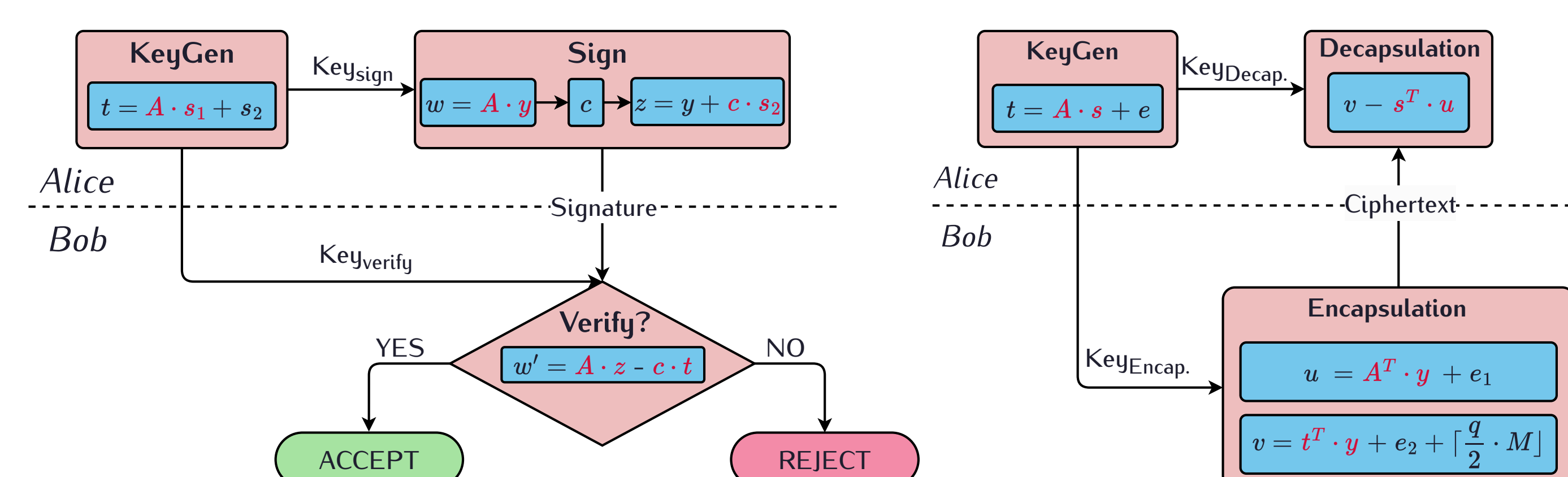
Florida Atlantic University & University of South Florida

Abstract

- ✧ **The Threat:** Quantum computers are coming, and they will break today's cryptography (like RSA & ECC).
- ✧ **The Defense:** We need Post-Quantum Cryptography (PQC). Algorithms like ML-KEM and ML-DSA are the new standards.
- ✧ **The Challenge:** These new algorithms are slow because of a bottleneck called the Number Theoretic Transform (NTT).
- ✧ **Our Breakthrough:** We created ParallelNTT, a new hardware architecture that smashes this bottleneck.
- ✧ **The Result:** Our design achieves a theoretical minimum latency of $\mathcal{O}(\log_2 n)$ and is up to 24x faster than SOTA.



The Problem: NTT is the Bottleneck



- ✧ **High Computational Cost:** In PQC schemes like ML-KEM & ML-DSA, polynomial multiplication is the most expensive operation.
- ✧ **The Bottleneck:** The Number Theoretic Transform (NTT) and its inverse (iNTT) are used to speed this up, but they still consume over 40% of the power and time.
- ✧ **Need for Speed:** For fast applications (web servers, secure IoT), this bottleneck is unacceptable, since speed is everything in these applications.

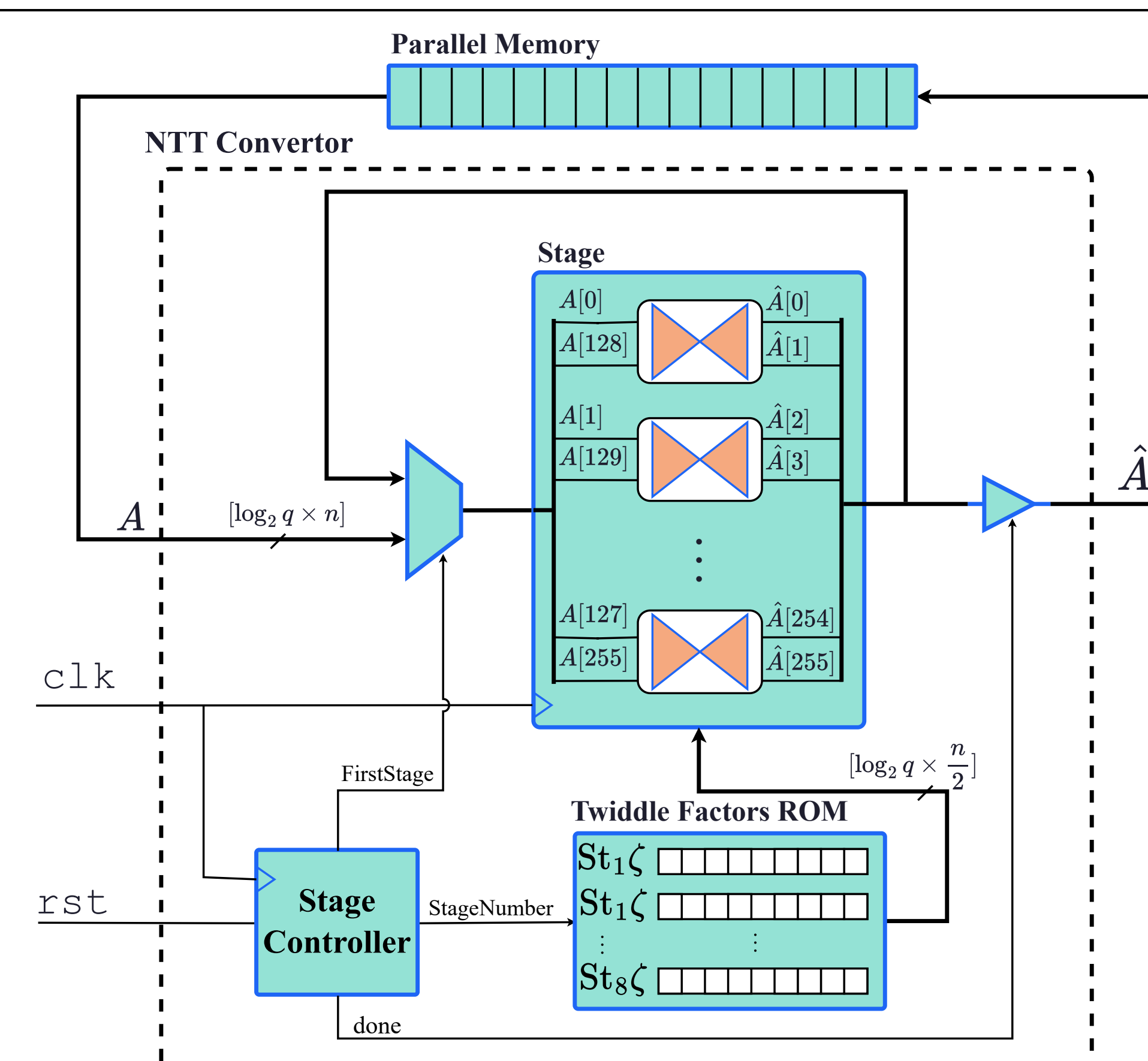
Our Solution

We designed a novel, fully parallel architecture that computes an entire NTT stage in a single clock cycle.

Key Innovations:

- ✧ **Fully Parallel Pipeline:** We eliminate sequential RAM access by feeding all 256 polynomial coefficients to the processing stage simultaneously, achieving true $\mathcal{O}(\log_2 n)$ latency.
- ✧ **Constant-Geometry Structure:** Simplifies routing and placement on the FPGA by ensuring the interconnections between Butterfly Units are identical for every stage.
- ✧ **Shared Twiddle Factor ROM:** A single ROM stores the precomputed constants (twiddle factors) for both forward (NTT) and inverse (iNTT) transforms, reducing area and increasing speed.
- ✧ **Efficient K²-RED Reduction:** We use K²-RED for modular reduction, which is highly optimized for NTT as one of the multiplication operands is a known constant.

ParallelNTT Hardware Architecture



Evaluation & Results

We implemented ParallelNTT on both high-performance Xilinx Virtex UltraScale+ and low-cost Artix-7 FPGAs. Our results show a massive leap in performance compared to state-of-the-art implementations.

- ✧ **Blazing Fast Frequency:** Runs up to 400 MHz (Virtex UltraScale+) and 250 MHz (Artix-7).
- ✧ **Record-Low Latency:** Just 15 ns for ML-KEM and 17 ns for ML-DSA on Virtex UltraScale+.
- ✧ **Massive Speedup:** 24x faster for ML-KEM and 21x faster for ML-DSA than the best previous designs.

Design	Scheme	Latency(μ s)	LUTs	FFs	DSPs	ATP
[1]	KEM	0.4	18296	12134	64	14353
[2]	KEM	1.24	904	811	4	3137
[3]	DSA	0.39	13804	11019	0	7532
TW, VU+	KEM	0.015	19065	8682	128	696
TW, VU+	DSA	0.017	34595	18565	256	1887
TW, A7	KEM	0.024	20962	8682	128	1160
TW, A7	DSA	0.024	41681	18855	240	3111

Conclusion

ParallelNTT represents a significant advancement in the performance of NTT for MLBC schemes. By fully parallelizing the architecture and optimizing memory access, we achieve **unprecedented speed and efficiency**, making it a compelling choice for PQC applications.

References

- [1] M. Saoudi, A. Kermiche, O. H. Benhaddad, N. Guetmi, and B. Allailou, "Low latency FPGA implementation of NTT for Kyber," *Microprocessors and Microsystems*, vol. 107, p. 105059, 2024.
- [2] Y. Itabashi, R. Ueno, and N. Homma, "Efficient modular polynomial multiplier for NTT accelerator of Crystals-Kyber," in *2022 25th Euromicro Conference on Digital System Design (DSD)*, pp. 528–533, 2022.
- [3] T.-H. Nguyen, B. Kieu-Do-Nguyen, C.-K. Pham, and T.-T. Hoang, "High-speed NTT accelerator for CRYSTAL-Kyber and CRYSTAL-Dilithium," *IEEE Access*, vol. 12, pp. 34918–34930, 2024.