# RECAPHE: REconfigurable Polynomial Modular Computation Architectures for Unified PQC and HE Schemes

Antian Wang⋆, Kaiyuan Zhang‡, Keshab K Parhi†, Yingjie Lao‡
⋆ Department of Electrical and Computer Engineering, Purdue University Fort Wayne, Fort Wayne, IN 46805, USA, †Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN 55455, USA, ‡Department of Electrical and Computer Engineering, Tufts University, Medford, MA 02155, USA
⋆ antian.wang@pfw.edu †parhi@umn.edu ‡ {kzhang11, yingjie.lao}@tufts.edu

***Abstract*—Post-Quantum Cryptography (PQC) and Homomorphic Encryption (HE) are emerging security primitives that strengthen data protection against adversaries equipped with quantum computing capabilities. Although PQC and HE rely on similar underlying arithmetic operations, their hardware implementations are typically developed independently due to differences in key parameters such as polynomial length and modulus bit-width. This work presents a unified lattice-based polynomial modular accelerator that efficiently supports both PQC and HE primitives, bridging these two domains toward future secure computing architectures. The proposed design introduces highly reconfigurable modular computation units that enable low-overhead runtime configuration across the parameter ranges commonly used in PQC and HE schemes. This unified architecture eliminates the need for separate domain-specific accelerators by reusing shared computation structures and workload patterns across both cryptographic schemes.**

***Index Terms*—Lattice-based cryptography, Homomorphic encryption, Post quantum cryptography, FPGA, Reconfigurable computing.**

## I. Introduction

The rise of quantum computing and the increasing emphasis on data privacy have driven the development of privacy-preserving cryptographic technologies. Post-Quantum Cryptography (PQC) and Homomorphic Encryption (HE) represent two major approaches to addressing these challenges. PQC offers alternatives to traditional public-key cryptography, ensuring secure data exchange even against adversaries with quantum capabilities. Notably, CRYSTALS-Kyber [1] has been standardized as a Public-Key Encryption (PKE) and Key-Encapsulation Mechanism (KEM), while CRYSTALS-Dilithium [2] has been selected as a lattice-based digital signature scheme. In contrast, HE enables computations to be performed directly on encrypted data, supporting arithmetic operations (addition and multiplication), Boolean operations (comparison and sign), and rotation operations (automorphisms). Current HE schemes [3–5] are primarily constructed on the Ring Learning-with-Errors (R-LWE) problem.

Although PQC and HE target different applications, they share a similar arithmetic foundation, involving polynomial addition, subtraction, and multiplication over modular rings, with variations in modulus bit width and polynomial length. This computational similarity offers an opportunity to develop shared hardware platforms that support both domains, enabling fully secure end-to-end systems. *Given that the server performing HE operations should not have access to the HE decryption capability, as providing it would compromise the integrity of the computed data. In addition, configuration data should not be transmitted to the server in plaintext, as this creates the weakest point in the secure computing system. Therefore, using PQC for transmitting essential configuration data to the server is necessary.* In such systems, PQC can secure communication between clients, while HE allows encrypted data processing in domains such as cloud computing [6], healthcare [7], and large language models [8]. To meet the growing computational demands of these applications while maintaining high efficiency, a unified reconfigurable hardware architecture that supports both PQC and HE is highly desirable. Such an approach provides a practical pathway to achieve secure, resource-efficient, and flexible cryptographic processing across diverse workloads.

In this paper, we present RECAPHE, a unified hardware computation architecture designed to support both HE and PQC schemes efficiently. The proposed architecture utilizes a low-overhead, reconfigurable computing structure that dynamically adapts to the diverse parameter settings employed in PQC and HE. The main contributions of this work are summarized as follows:

- A configurable dual-scheme modular multiplier that supports widely used PQC and HE parameters with minimal resource overhead.
- A high-speed hybrid butterfly computation module capable of efficient execution across both PQC and HE schemes.

The remainder of the paper is organized as follows: Section II provides background on Lattice-Based Cryptography (LBC) and its relevance to PQC and HE. Section III details the proposed RECAPHE architecture. Section IV presents experimental results, and Section V concludes the paper.

## II. Background

### A. Lattice-based Cryptography for PQC and HE

LBC is founded on the NP-hard lattice problems that remain intractable even for quantum computers [9]. Its arithmetic

simplicity and strong security guarantees make it a promising foundation for PQC and HE. The schemes supported by RECAPHE belong to a subclass of LBC: the Ring Learning with Errors (R-LWE) and Module-LWE formulations. Polynomial arithmetic in these schemes is performed over the ring $R_q = \mathbb{Z}_q[x]/(x^n + 1)$, where $x^n + 1$ is a length-$n$ irreducible polynomial and $q$ is the modulus for polynomial coefficients.

In the R-LWE construction, two polynomials are sampled to form a pair $(a(x), b(x)) \in R_q \times R_q$, where $a(x)$ is a uniformly random polynomial over $R_q$, and $b(x)$ is defined as

$$b(x) = a(x) \cdot s(x) + e(x) \in R_q, \tag{1}$$

where $s(x) \in R_q$ is the secret polynomial and $e(x) \in R_q$ represents the small error term. The Module-LWE variant extends this construction by defining $s(x)$ and $a(x)$ as $d$-dimensional vectors, i.e., $s(x)$, $a(x) \in (R_q)^d$, with each component being a polynomial in $R_q$. Readers interested in key generation, encryption, decryption procedures, and parameter settings can refer to Kyber [1], BFV [3, 4], and CKKS [5].

### B. Prior Works on Hardware Design for PQC and HE

Efficient FPGA-based architectures are essential for the deployment of PQC [10] and HE, as they enable high performance with reduced resource utilization, including Look-Up Tables (LUTs), Flip-Flops (FFs), and Digital Signal Processing (DSP) blocks. Numerous FPGA acceleration techniques have been proposed to optimize core modules in both domains.

One major design focus is modular multiplication. FPGA implementations commonly employ reduction algorithms such as Barrett reduction [11] to accelerate modular arithmetic. Designers often exploit moduli with sparse power-of-two structures [12–14] to minimize the use of multipliers. For instance, shift-and-add architectures [12, 15] and Karatsuba-based multiplication schemes [16] have been applied to PQC [17] and HE [13] implementations to reduce DSP utilization.

Polynomial multiplication over prime fields is typically accelerated using the Number Theoretic Transform (NTT), which reduces complexity from $O(n^2)$ to $O(n \log n)$. Many FPGA designs adopt unified butterfly modules [18, 19] that support both NTT and Inverse NTT (INTT) operations, and in combination with optimized twiddle-factor storage [20, 21] to minimize memory requirements. These unified arithmetic modules serve as standard building blocks for PQC and HE accelerators, enhancing hardware utilization.

Beyond arithmetic optimization, prior works have explored architectural flexibility. Some designs merge butterfly and coefficient-wise modular add/subtract operations into reconfigurable modules [22], enabling trade-offs between performance and resource efficiency. High-level folding transformations have been applied to polynomial multiplication architectures to improve computational scheduling [23, 24], and optimized memory-flow architectures [25] have enabled large-scale HE implementations on FPGAs. More recently, system-level designs have introduced custom instruction extensions for PQC [26, 27] and HE [28], improving integration within general-purpose hardware frameworks.

Despite sharing many arithmetic primitives, most existing FPGA implementations treat PQC and HE as independent domains. Few studies explicitly bridge the two. The proposed RECAPHE architecture addresses this gap by unifying key computational modules and reconfigurable design techniques, enabling efficient support for both PQC and HE within a single hardware framework.

## III. RECAPHE Architecture Design

The high-level concept of RECAPHE is to construct a configurable lattice-based cryptographic hardware architecture that efficiently supports both PQC and HE schemes by leveraging their shared computational characteristics. The reconfigurability of RECAPHE focuses on two key components: the modular multiplier and the butterfly operation module.

### A. Dual-scheme Modular Multiplier

A primary distinction between PQC and HE schemes lies in the bit length of their moduli. For PQC schemes, Kyber and Dilithium employ 12-bit and 23-bit moduli, respectively, whereas HE schemes typically operate with moduli ranging from 32-bit to 60-bit. A straightforward approach would be to adapt an HE-compatible modular multiplier for PQC parameters, with the higher unused bits kept as 0; however, this results in inefficient hardware utilization, as most computational resources remain idle during smaller-bitwidth operations.

In FPGA implementations, large-bitwidth modular multiplications are realized by cascading multiple on-board DSP blocks. This cascading naturally produces partial multiplication results that can be reused for lower-bitwidth modular multiplications, such as those used in PQC schemes. For the HE configuration using a 54-bit modulus, it is possible to integrate four 12-bit modular multipliers for Kyber by adding additional configuration logic. However, such a design incurs significant overhead due to the inefficient use of the standard $27\times18$-bit DSP multipliers available on FPGAs and LUT resource consumption overhead in the overall configuration path. Unlike prior work, such as KaLi [29], which supports both PQC public-key and signature operations, our design adapts a two-parallel PQC configuration to fully utilize the 54-bit modular multiplier resources available for HE.

As illustrated in Fig. 1, the proposed 54-bit FPGA-based modular multiplier is partitioned into several 27-bit $\times$ 18-bit modular multipliers. In Algorithm 1, a baseline Barrett reduction modular multiplier is presented. The input operands $a$ and $b$ are first decomposed into 27-bit and 18-bit segments. Six partial products are then computed as $m_0 = a[26:00] \cdot b[17:00]$, $m_1 = a[26:00] \cdot b[35:18]$, $m_2 = a[26:00] \cdot b[53:36]$, $m_3 = a[53:27] \cdot b[17:00]$, $m_4 = a[53:27] \cdot b[35:18]$, $m_5 = a[53:27] \cdot b[53:36]$. Depending on the PQC or HE scheme used in the computation, these six partial results are reorganized to form the output in Step 1 of Algorithm 1. For Steps 2 through 6 of the proposed multiplier architecture, the dataflow is decomposed into two parts (upper 27-bit and lower 27-bit) when operating under the PQC scheme, or processed without decomposition under the HE scheme. For the decomposed PQC dataflow, the intermediate values $t$ and $y$ are computed separately using the PQC parameters $k$ and $q$ in Steps 2, 3, and 5 of Algorithm 1, and are then

merged into a unified 54-bit dataflow. This dataflow design maintains uniformity across the proposed modular multiplier while supporting modular multiplication for both schemes.

This configuration efficiently reuses DSP resources, achieving high utilization for both PQC and HE operations. Since the moduli used in Kyber and Dilithium are smaller than 27-bits, this approach minimizes configuration resource overhead.

**Algorithm 1** Modular multiplication with Barrett reduction.

**Input:** $a, b \in \mathbb{Z}_q$ $m = \lfloor 2^k/q \rfloor$, $k = 2\lceil \log_2 q \rceil$, Config $\in \{\text{HE}, \text{PQC}\}$
**Output:** $y = a \cdot b \bmod q$

1: $z = a \cdot b$
2: $t = (z \cdot m) >> k$
3: $y = z - (t \cdot q)$
4: **if** $y \geq q$ **then**
5: $\quad y = y - q$
6: **end if**
7: **return** $y$

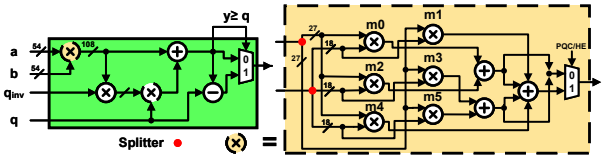


Fig. 1: Dual-scheme modular multiplier with configurable data paths for PQC and HE.

### B. High-Speed Hybrid Butterfly Operation Module

Another major difference between PQC and HE schemes is the polynomial length: PQC typically operates on length-$2^8$ polynomials, whereas HE schemes use lengths ranging from $2^{12}$ to $2^{16}$ depending on the desired security level. To support this wide parameter range, RECAPHE adopts a hybrid butterfly architecture that combines memory-based and Multi-path Delay Commutator (MDC)-based computation.

In HE configurations, memory-based butterfly modules are preferred since they efficiently handle large polynomial lengths using ping-pong operations between on-chip memory banks. For PQC, however, the low polynomial length results in poor memory utilization, making a fully memory-based design inefficient. PQC implementations typically favor MDC-based butterfly modules for higher throughput and lower latency.

To address these differing requirements, we propose a hybrid architecture comprising groups of eight butterfly computation units that can be dynamically configured as either memory-based or MDC-based modules. When configured in memory-based mode, the units perform standard 8-parallel ping-pong butterfly operations between two sets of Block RAMs (BRAM). In MDC-based mode, the same eight butterfly units interconnect to form a length-256 bidirectional butterfly network, supporting both NTT and INTT computations as described in [30]. The bidirectional structure eliminates the need for separate NTT and INTT hardware, improving hardware reuse. Independent twiddle-factor memories are maintained for the two configurations to eliminate the need to load twiddle factors for PQC schemes. Although the bi-directional butterfly module can perform both NTT and INTT within a unified architecture, it cannot execute them simultaneously because the NTT input port also serves as the INTT output port, and vice versa. However, consecutive NTT or INTT operations can be supported through appropriate data post-processing. This design effectively reduces resource usage.

Due to Kyber's parameter setting, where $q = 3329$ does not satisfy $q \equiv 1 \pmod{512}$, the first stage of the NTT and the last stage of the INTT differ when operating over length-256 polynomials instead of length-512. This stage, referred to as the *special stage*, requires no arithmetic computation. Omitting it would complicate reconfiguration because Kyber's NTT and INTT would then need to align with the seventh stage rather than the eighth in the pipeline, thereby increasing routing and control complexity. To preserve pipeline uniformity, this stage is implemented as a lightweight *swap stage* that exchanges input coefficients without arithmetic operations. This solution maintains consistent data flow across PQC and HE configurations while minimizing reconfiguration overhead.

## IV. Experimental Results

### A. Experimental Setting

The RECAPHE architecture is implemented in Verilog HDL and synthesized for the Xilinx U280 FPGA acceleration board, which is representative of cloud-oriented privacy-preserving computing platforms. The implementation setup follows configurations similar to those used in prior works [13, 30]. The design integrates two pairs of modular coefficient-wise computation units and butterfly operation modules, supporting polynomial lengths up to $2^{16}$ to accommodate both PQC and HE parameter ranges.

### B. Result Analysis and Discussion

*a) Module-wise resource consumption:* As summarized in Table I, the resource utilization of various modular multiplier configurations is reported in terms of LUT, FF, and DSP usage, along with the FPGA equivalent area calculated using the model from [31]. The proposed reconfigurable design achieves a 20.04% reduction in equivalent area compared to implementing separate modular multipliers for Kyber, Dilithium, and the 54-bit HE schemes with a reasonable increase in LUT and FF without introducing routing overhead. This dual-scheme design thus provides a favorable balance between resource efficiency and computational flexibility.

TABLE I: Resource utilization comparison of modular multiplier configurations

| Parameter | LUT/FF/DSP | Equiv. Area |
|---|---|---|
| Kyber | 172/38/2 | 249.25 |
| Dilithium | 273/89/5 | 579.375 |
| 54-bit HE | 1038/464/30 | 3317.5 |
| Unified | 1869/883/34 | 3977.625 |

The resource consumption for the three types of butterfly module is presented in Table II, showing that the hybrid structure saves 35.80% of the equivalent area consumption. Note that the URAM is not listed in the equivalent area computation, as it is only used for storing the intermediate polynomial and

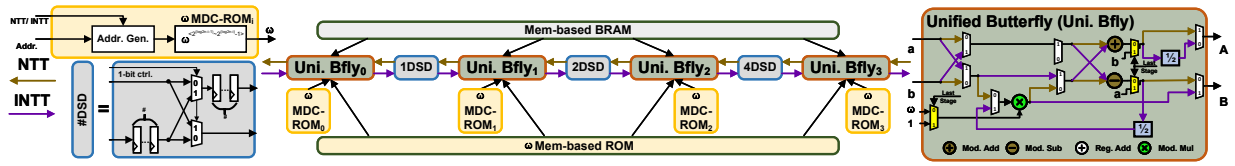


Fig. 2: Hybrid butterfly module combining memory-based and MDC-based computation for $n = 8$, modified from [30].

is not involved in the computation. Additionally, the twiddle factor for MDC-based butterfly operation is implemented in LUT/FF due to the small size of the constant values.

TABLE II: Resource utilization comparison of hybrid butterfly module configurations

| Parameter | LUT/FF/DSP | BRAM/URAM | Equiv. Area |
|---|---|---|---|
| Mem-only (8-par) | 31623/15942/272 | 5.5/4 | 38198.5 |
| MDC-only | 42447/17154/272 | 125/32 | 64956.0 |
| Hybrid (8-par) | 46734/18773/272 | 125/32 | 66230.125 |

*b) Overall resource consumption:* The evaluation results are summarized in Table III. The implementation instantiates three reconfigurable butterfly modules, two coefficient-wise modular arithmetic modules, and associated memory blocks for intermediate data transfers, similar to the design in [30]. The results demonstrate the feasibility and efficiency of a unified reconfigurable hardware architecture that concurrently supports PQC and HE workloads. Specifically, the RECAPHE can execute three length-256 PQC-parameter NTT or INTT operations in parallel within 1.15 $\mu$s, and a 54-bit HE length-$2^{16}$ NTT or INTT operation within 219.10 $\mu$s. These measurements confirm that runtime configurability introduces negligible latency overhead while maintaining high throughput for both domains. Compared to HERMES [30], the RECAPHE removes the grouping of moduli used for CKKS operations while increasing configuration logic in supporting dual-scheme computation, resulting in differences in LUT and FF resource consumption as compared with the RECAPHE(Scaled) and HERMES with an identical number of butterfly modules. However, this simplification improves configurability, reduces design complexity without degrading overall performance, and preserves the algorithmic structures of both PQC and HE schemes. Similarly, compared to [25] and [32], the reduced resource consumption and broader support for LBC schemes demonstrate RECAPHE's advantage.

## V. Conclusion

In this work, we propose RECAPHE, a unified computation architecture that supports both lattice-based PQC and HE algorithms. The architecture enables seamless seamless integration of PQC parameters into the hardware design while maintaining compatibility with HE parameters on an FPGA-based data center acceleration board. Compared to conventional HE architectures, RECAPHE achieves low resource and performance overhead with additional support of PQC parameters.

## Acknowledgment

This work is supported in part by the National Science Foundation (NSF) under grant numbers CCF-2243053, CCF-2412357, and SaTC-2426299.

## References

[1] J. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, P. Schwabe, G. Seiler, and D. Stehlé, "CRYSTALS-kyber: A CCA-secure module-lattice-based KEM," in *2018 IEEE European Symposium on Security and Privacy (EuroS&P)*, IEEE, 2018, pp. 353–367.

[2] L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, P. Schwabe, G. Seiler, and D. Stehlé, "Crystals-dilithium: A lattice-based digital signature scheme," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pp. 238–268, 2018.

[3] J. Fan and F. Vercauteren, "Somewhat practical fully homomorphic encryption.," *IACR Cryptology ePrint Archive*, vol. 2012, p. 144, 2012.

[4] S. Halevi, Y. Polyakov, and V. Shoup, "An improved rns variant of the BFV homomorphic encryption scheme," in *Cryptographers' Track at the RSA Conference*, Springer, 2019, pp. 83–105.

[5] J. H. Cheon, A. Kim, M. Kim, and Y. Song, "Homomorphic encryption for arithmetic of approximate numbers," in *International Conference on the Theory and Application of Cryptology and Information Security*, Springer, 2017, pp. 409–437.

[6] F. Zhao, C. Li, and C. F. Liu, "A cloud computing security solution based on fully homomorphic encryption," in *16th ICACT*, IEEE, 2014, pp. 485–488.

[7] K. Munjal and R. Bhatia, "A systematic review of homomorphic encryption and its contributions in healthcare industry," *Complex & Intelligent Systems*, vol. 9, no. 4, pp. 3759–3786, 2023.

[8] L. W. Folkerts and N. G. Tsoutsos, "Testing robustness of homomorphically encrypted split model llms," in *2025 Design, Automation & Test in Europe Conference (DATE)*, IEEE, 2025, pp. 1–7.

[9] O. Regev, "On lattices, learning with errors, random linear codes, and cryptography," *Journal of the ACM (JACM)*, vol. 56, no. 6, pp. 1–40, 2009.

[10] J. Xie, W. Zhao, H. Lee, D. B. Roy, and X. Zhang, "Hardware circuits and systems design for post-quantum cryptography–a tutorial brief," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 71, no. 3, pp. 1670–1676, 2024.

[11] P. Barrett, "Implementing the Rivest Shamir and Adleman public key encryption algorithm on a standard digital signal processor," in *Conference on the Theory and Application of Cryptographic Techniques*, Springer, 1986, pp. 311–323.

[12] U. Banerjee, T. S. Ukyab, and A. P. Chandrakasan, "Sapphire: A configurable crypto-processor for post-

TABLE III: FPGA Performance comparison of RECAPHE with prior works

| Work | Scheme | FPGA | $(\log n, q)$ | Freq (GHz) | KLUT | KFF | DSP | BRAM(36K) | URAM |
|---|---|---|---|---|---|---|---|---|---|
| RECAPHE | CKKS | U280 | $(16, 54 \times 33), (8, 12 \text{ or } 23)$ | 300 | 392 | 330 | 1908 | 656.5 | 96 |
| RECAPHE(Scaled) | CKKS | U280 | $(16, 54 \times 33), (8, 12 \text{ or } 23)$ | 300 | 532 | 386 | 2724 | 656.5 | 96 |
| HERMES[30] | CKKS | U280 | $(12, 54 \times 33)$ | 300 | 466 | 462 | 3844 | 319.5 | 36 |
| FAB[25] | CKKS | U280 | $(16, 54 \times 23)$ | 300 | 899 | 2073 | 5120 | 1920 | 960 |
| HEAWS[32] | BFV-HPS | VirtexU+ | $(12, 30 \cdot 6)$ | 200 | 582 | 634 | 2,256 | 1,890 | 392 |


quantum lattice-based protocols," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pp. 17–61, 2019.

[13] A. Wang, W. Tan, Z. Xu, T. Wei, C. Ding, K. K. Parhi, and Y. Lao, "HEDWIG: Homomorphic encryption accelerator design using bfv-hps with high-speed fixed-point approximation," in *Proceedings of the 2025 ACM/SIGDA International Symposium on Field Programmable Gate Arrays*, 2025, pp. 184–184.

[14] W. Tan, A. Wang, Y. Lao, X. Zhang, and K. K. Parhi, "Pipelined high-throughput NTT architecture for lattice-based cryptography," in *2021 Asian Hardware Oriented Security and Trust Symposium (AsianHOST)*, IEEE, 2021, pp. 1–4.

[15] Z. Liu, H. Seo, S. S. Roy, J. Großschädl, H. Kim, and I. Verbauwhede, "Efficient Ring-LWE encryption on 8-bit AVR processors," in *International Workshop on Cryptographic Hardware and Embedded Systems*, Springer, 2015, pp. 663–682.

[16] A. Karatsuba, "Multiplication of multidigit numbers on automata," in *Soviet physics doklady*, vol. 7, 1963, pp. 595–596.

[17] W. Tan, B. M. Case, A. Wang, S. Gao, and Y. Lao, "High-speed modular multiplier for lattice-based cryptosystems," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 68, no. 8, pp. 2927–2931, 2021.

[18] Y. Geng, X. Hu, M. Li, and Z. Wang, "Rethinking parallel memory access pattern in number theoretic transform design," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 70, no. 5, pp. 1689–1693, 2023.

[19] N. Zhang, B. Yang, C. Chen, S. Yin, S. Wei, and L. Liu, "Highly efficient architecture of NewHope-NIST on FPGA using low-complexity NTT/INTT," *IACR TCHES*, vol. 2020, no. 2, pp. 49–72, 2020.

[20] M. Bisheh-Niasar, R. Azarderakhsh, and M. Mozaffari-Kermani, "High-speed ntt-based polynomial multiplication accelerator for post-quantum cryptography," in *2021 IEEE 28th Symposium on Computer Arithmetic (ARITH)*, IEEE, 2021, pp. 94–101.

[21] X. Hu, J. Tian, M. Li, and Z. Wang, "AC-PM: An area-efficient and configurable polynomial multiplier for lattice based cryptography," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 70, no. 2, pp. 719–732, 2022.

[22] Y. Su, B.-L. Yang, C. Yang, and S.-Y. Zhao, "ReMCA: A reconfigurable multi-core architecture for full RNS variant of BFV homomorphic evaluation," *IEEE TCAS I*, vol. 69, no. 7, pp. 2857–2870, 2022.

[23] W. Tan, S.-W. Chiu, A. Wang, Y. Lao, and K. K. Parhi, "PaReNTT: Low-latency parallel residue number system and NTT-based long polynomial modular multiplication for homomorphic encryption," *IEEE Transactions on Information Forensics & Security*, vol. 19, pp. 1646–1659, 2024.

[24] S.-W. Chiu and K. K. Parhi, "Architectures for serial and parallel pipelined NTT-based polynomial modular multiplication," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2025.

[25] R. Agrawal, L. de Castro, G. Yang, C. Juvekar, R. Yazicigil, A. Chandrakasan, V. Vaikuntanathan, and A. Joshi, "FAB: An fpga-based accelerator for bootstrappable fully homomorphic encryption," in *2023 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*, IEEE, IEEE, 2023, pp. 882–895.

[26] Y. Cui, J. Chen, Z. Ni, Z. Zhang, C. Wang, and W. Liu, "Instruction-based high-performance hardware controller of CRYSTALS-kyber with balanced resource utilization," *IEEE Transactions on Circuits and Systems I: Regular Papers*, 2025.

[27] Z. Ni, A. Khalid, W. Liu, and M. O'Neill, "A highly hardware efficient ml-kem accelerator with optimised architectural layers," *ACM Transactions on Embedded Computing Systems*, vol. 24, no. 2, pp. 1–24, 2025.

[28] Y. Yang, R. Kannan, and V. K. Prasanna, "OLA: An FPGA-based overlay accelerator for privacy preserving machine learning with homomorphic encryption," in *Proceedings of the 2025 ACM/SIGDA International Symposium on Field Programmable Gate Arrays*, 2025, pp. 127–138.

[29] A. Aikata, A. C. Mert, M. Imran, S. Pagliarini, and S. S. Roy, "KaLi: A crystal for post-quantum security using kyber and dilithium," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 70, no. 2, pp. 747–758, 2022.

[30] A. Wang, K. Zhang, K. K. Parhi, and Y. Lao, "HERMES: Homomorphic encryption over residual number system for multi-level evaluations," in *IEEE International Conference on Computer-Aided Design*, IEEE, 2024.

[31] W. Liu, S. Fan, A. Khalid, C. Rafferty, and M. O'Neill, "Optimized schoolbook polynomial multiplication for compact lattice-based cryptography on fpga," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 27, no. 10, pp. 2459–2463, 2019.

[32] F. Turan, S. S. Roy, and I. Verbauwhede, "HEAWS: An accelerator for homomorphic encryption on the amazon AWS FPGA," *IEEE TC*, vol. 69, no. 8, pp. 1185–1196, 2020.