A 28 nm 75.6 KOPS 13 nJ Computing-in-Memory Pipeline Number Theoretic Transform Accelerator for PQC

Jialiang Zhu, Yiyang Yuan, Long Nie, Weiye Tang, Ming Li, Hao Wu, Xiaojin Zhao, *Senior Member, IEEE*, Guozhong Xing, *Member, IEEE*, and Feng Zhang, *Senior Member, IEEE*

***Abstract*—Lattice-based cryptography (LBC) exploits the learning with errors (LWE) problem and is the main algorithm standardized for Post-Quantum Cryptography (PQC). Number theoretic transforms (NTT) account for most of the latency and energy in the computation of the LWE problem. This brief presents a Compute-in-Memory (CIM) configurable-pipeline NTT accelerator for PQC. The accelerator incorporates a bidirectional pipeline array to minimize data latency, CIM processing elements to reduce memory access, and a parallel PQC circuit for LBC protocol deployment. A 28 nm chip of the accelerator consumes only 13 nJ per 256-point NTT, while achieving a throughput of 75.6 KOPS that achieves a remarkable reduction of up to 78% in clock cycles and a 45% reduction in energy consumption than state-of-the-art designs.**

***Index Terms*—Post-quantum cryptography (PQC), number theoretic transform (NTT), Compute-in-Memory (CIM), pipelined design.**

I. Introduction

POST-QUANTUM cryptography (PQC) [1] plays a crucial role in ensuring the security of information in the Internet, Internet of Things (IoT), and electronic devices in the era of quantum computing. Lattice-based cryptography (LBC) serves as the primary algorithmic framework in the National Institute of Standards and Technology (NIST) PQC standards [2]. LBC have come to light because of the hardness of inherent learning with error (LWE) problems [3], [4]. With Ring LWE problem, in which polynomial multiplication is the most time-consuming routine. For example, the polynomial multiplications occupy 78% of computation time in Kyber [5]. Polynomial multiplication is usually performed using the number theoretic transform (NTT) [6]. However, the efficient implementation of the NTT, a fundamental operation in PQC, poses a significant challenge in terms of achieving a satisfactory trade-off between computational speed and energy consumption. This limitation severely impedes the practicality and widespread adoption of cryptographic techniques on resource-constrained IoT devices.

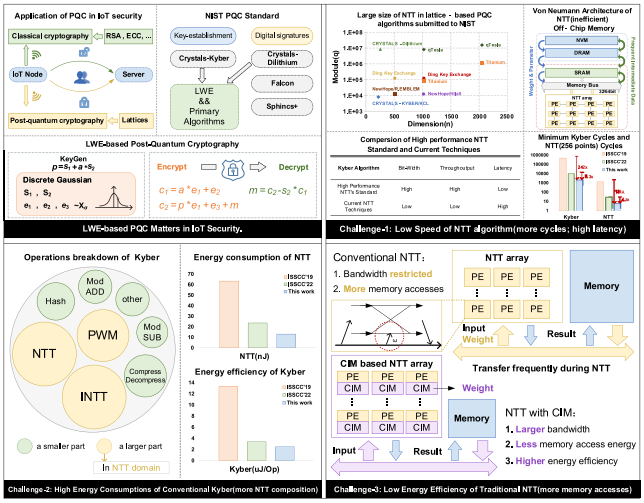


Fig. 1. Lattice-based PQC challenges.

The primary challenges associated with Lattice-based PQC is presented in Fig. 1. The utilization of high bit-width operations in the NTT leads to increased latency when implemented on low-bit-width von Neumann computing architectures using split iterative techniques. Additionally, the high-density computing nature of NTT results in considerable energy consumption, while frequent storage access further limits overall energy efficiency. Reference [7] proposes a hardware acceleration architecture tailored for polynomial operations in Kyber, while [8] employs a compact NTT structure to reduce hardware area. In [9], a pipelined NTT architecture is proposed to optimize computation latency. Meanwhile, [10] introduces a look-up-table-based modular multiplication method for NTT, aiming to reduce hardware resource consumption.

Compute-in-Memory (CIM) [11] is an emerging memory constrained computing technology that combines the functionality of logical operations based on a storage column

Received 30 July 2024; revised 11 September 2024; accepted 14 October 2024. Date of publication 16 October 2024; date of current version 27 December 2024. This work was supported in part by the National Key Research and Development Program of China under Grant 2023YFB4402400; in part by the Strategic Priority Research Program of the Chinese Academy of Sciences China under Grant XDB44000000 and Grant XDA330100; and in part by the National Natural Science Foundation of China under Grant U2341218. This brief was recommended by Associate Editor S.-B. Ko. *(Corresponding author: Feng Zhang.)*

Jialiang Zhu, Yiyang Yuan, Long Nie, Weiye Tang, Ming Li, Hao Wu, Guozhong Xing, and Feng Zhang are with the Key Laboratory of Fabrication Technologies for Integrated Circuits, Institute of Microelectronics, Chinese Academy of Sciences, Beijing 100029, China, and also with the School of Integrated Circuits, University of Chinese Academy of Sciences, Beijing 100049, China (e-mail: zhangfeng_ime@ime.ac.cn).

Xiaojin Zhao is with Shenzhen University, Shenzhen 518060, China.

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCSII.2024.3481996.

Digital Object Identifier 10.1109/TCSII.2024.3481996

architecture, with highly parallel computing capabilities and extremely low computational power consumption overhead. Reference [12] proposes a high-linearity CIM multiplier. Processing In-Memory (PIM) avoids frequent data transmission between computing units when performing locally intensive computing tasks. Previous work has shown that PIM is highly energy-efficient and fast for data-intensive tasks [13], [14], [15]. These additional features make it a promising technology for accelerating NTT [16].

This brief introduces a novel approach, the CIM Pipeline NTT accelerator, which addresses these challenges through the utilization of: (1) a configurable pipeline array architecture to optimize data path latency; (2) CIM processing elements to reduce power consumption during polynomial operations; and (3) a parallel PQC circuit for energy-efficient implementation of the Kyber algorithm. Comparing to a state-of-the-art NTT implementation [17] our NTT pipeline execution achieves a remarkable reduction of up to 78% in clock cycles and a 45% reduction in energy consumption.

II. CIM-NTT FOR PQC

NTT is an effective method to accelerate polynomial multiplication and reduce its time complexity to *O(nlog(n))*. NTT can be regarded as a generalized form of Discrete Fourier Transform (DFT) on finite fields, and its operation process is similar to that of Fast Fourier Transform (FFT). It mainly converts the coefficient representation of the polynomial into a point value representation after evaluating the polynomial at a special evaluation point. By selecting the evaluation point at the primitive n-th root-of-unity on Z_q, NTT can be completed by iterative operation. The primitive n-th root-of-unity ω on Z_q satisfies $\omega_n^n \equiv 1 \ mod \ q, \omega_n^{\frac{n}{2}} \equiv -1 \ mod \ q$ and when $i \neq n, \omega_n^i \neq 1 \ mod \ q$.

For a polynomial $a(x) \in R_q$, its coefficients are $(a_0, a_1, \ldots, a_{n-1})$, and $a(x)$ is evaluated at ω_n^n to obtain a point value representation $((\omega_n^0, \widehat{a}_0), (\omega_n^1, \widehat{a}_1), \ldots, (\omega_n^{n-1}, \widehat{a}_{n-1}))$, where $\widehat{a}_i$ satisfies:

$$\widehat{a}_i = \sum_{j=0}^{n-1} a_j \omega_n^{ij} \bmod q \ \forall \ i \in [0, n-1] \tag{1}$$

For $\widehat{a}(x) = (\widehat{a}_0, \widehat{a}_1, \ldots, \widehat{a}_{n-1})$, it is called the NTT positive transformation of polynomial $a(x)$ to obtain $\widehat{a}(x)$, as in

$$\widehat{a}(x) = NTT(a(x)) \tag{2}$$

Similarly, the polynomial point value representation can be transformed into its coefficient representation, specifically converting $\widehat{a}(x)$ back to $a(x)$. The detailed operational steps for this transformation are outlined as follows:

$$a_i = n^{-1} \sum_{j=0}^{n-1} \widehat{a}_j \omega_n^{-ij} \ mod \ q \ \forall \ i \in [0, n-1] \tag{3}$$

It is said that $a(x)$ is obtained by inverse NTT transformation $\widehat{a}(x)$, as in

$$a(x) = INTT(\widehat{a}(x)) \tag{4}$$

The inverse NTT and NTT operations have the same form, the main difference is that ω is used instead of ω^{-1}, and the final result is multiplied by n^{-1}. ω_n^{-1} and n^{-1} are the inverse elements of ω_n and n with the modulus q respectively. ω_n and ω_n^{-1} are rotation factors.

By using NTT and INTT, polynomials $a(x), b(x) \in R_q$ and the result $c(x) = a(x) \cdot b(x) \in R_q$of their multiplication can be expressed as follows

$$c(x) = INTT(NTT(a(x)) \odot NTT(b(x))) \tag{5}$$

where $\odot$ represents the point-wise multiplication (PWM) operation of the point value representations of the two polynomials.

A. Accelerator Architecture Design

Fig. 2 presents the architecture of the CIM-NTT accelerator. The accelerator employs a network topology comprising 7 unified Cooley-Tukey and Gentleman-Sande (CT+GS) butterfly levels arranged in a bidirectional pipeline array. Each level consists of 128 CIM-PEs that facilitate data operations and inter-level data caching. The array is configurable to support polynomial multiplication with variable lengths, ranging from n = 64 to 256, and a q specification of 24 bits, with a default value of n = 256.

Additionally, partial configuration of CIM-NTT accelerator allows enabling PWM. The CIM-PE array employs a specific mode configuration scheme, with butterfly units set to CT calculation mode for the NTT algorithm. In this mode, the array inputs data from S0 and outputs the result from S6. For PWM operation, the PEs are switched to PWM mode, with S3 as input and S6 as output. For the INTT algorithm, the PEs are in GS mode, with dataflow direction opposite to NTT. The interconnection between each level of the CIM-PE array enables seamless execution of high-bandwidth multi-operand (3072 bits) NTT computations without requiring external data transfer during computation. This eliminates redundant memory access, resulting in advantages such as strong independence, flexibility, high speed, and energy efficiency. Notably, for n = 256 NTT operations, the NTT accelerator requires only a single injection of the twiddle factor w, and the NTT data flow shown in Fig. 3 exhibits a latency of 7 clock cycles. Furthermore, the accelerator supports pipeline operations, enabling concurrent operation of each array stage and reducing the minimum average delay to as low as 1 cycle, thereby enhancing overall efficiency and performance.

B. CIM Processing Element

Fig. 4 depicts the design intricacies of the CIM-PE structure, which comprises the 4-bit CIM-Macro array, near-memory logic, and peripheral circuitry. The 4-bit CIM-Macro array utilizes an analog current mirror multiplier to efficiently perform 4-bit multiplications between the input and the content stored in the 6T-SRAM. The resulting outcome is then digitally converted using an analog-to-digital converter (ADC). The design of the CIM-PE structure revolves around the utilization of the Macro as its core, enabling compatibility with both CT and GS operations. The Macro serves a dual role as an arithmetic unit for logical operations and a flexible memory for

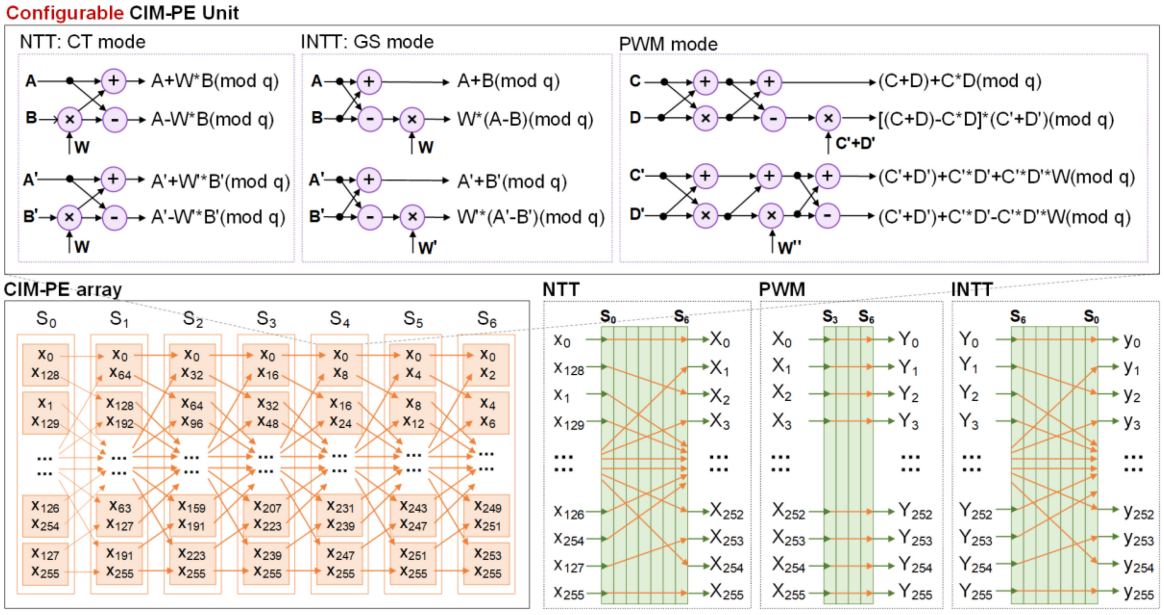


Fig. 2. Architecture of the CIM-NTT Accelerator.

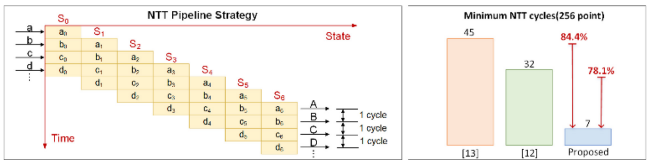


Fig. 3. Dataflow of the CIM-NTT Accelerator.

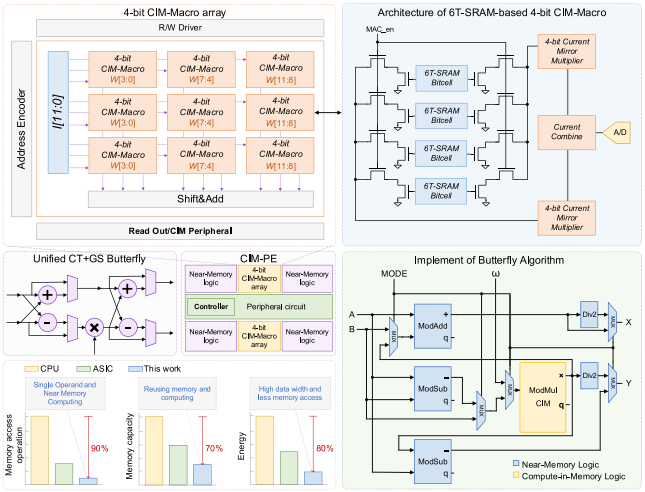


Fig. 4. System architecture of the CIM-PE.

parallelizing scalable operations. The rotation factor constants required for operations are pre-calculated and stored within the Macro, streamlining the operations to single-operand operations. The figure demonstrates how nine 4-bit Macros effectively enable a 12-bit logic operation, transforming large-scale stacking into a high-bit-width memory and operation array.

In both the CT and GS models, the 2-operand modular multiplication operation $I \cdot \omega \ mod \ q$ includes the input polynomial coefficient I and a rotation factor ω. The rotation factor ω, pre-loaded into the SRAM storage unit of the CIM-PE, remains fixed throughout the computation, thereby requiring only the input I during the calculation phase. To enhance operational efficiency, a 3x3 array interconnection of 4-bit macros is employed to perform 12-bit multiplication through a process of shifting and accumulation at the output stage. The peripheral integrated configurable modular operation circuit completes the modular multiplication operation, offering adaptability to different modulus values or other parameters. This design improves speed by reducing data input overhead, allows scalability via the use of 4-bit macros, and provides flexibility through its configurable nature. These characteristics make it particularly advantageous for applications in NTT, where efficient modular multiplication is critical.

Additionally, the near-memory computing architecture of the PE optimizes performance by caching intermediate data within the Macros, eliminating the need for access to general-purpose registers and significantly reducing data access overhead. This approach reduces the number of access operations by approximately 90% compared to conventional CPUs. Furthermore, the amalgamation of storage and computation results in approximately 70% memory capacity savings through multiplexing. Finally, the architectural characteristics of high bit width and low access memory contribute to an approximate 80% reduction in power consumption.

C. Implementation of Kyber

Fig. 5 demonstrates a proposed design for a high-efficiency Kyber circuit implementation using the previously described NTT accelerator. The circuit is integrated into a System-on-Chip (SoC) architecture with a RISC-V core, functioning as a co-processor for PQC operations. The NTT accelerator executes parallel Kyber operations with parameters set to

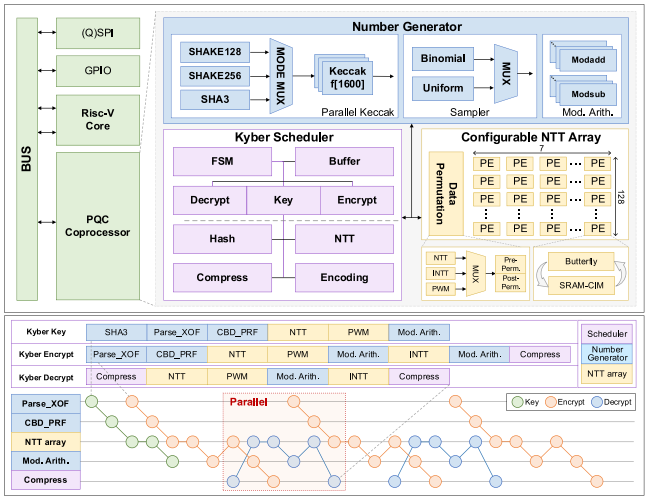


Fig. 5. Illustration of the efficient parallel implementation of Kyber algorithm and CIM-NTT array.

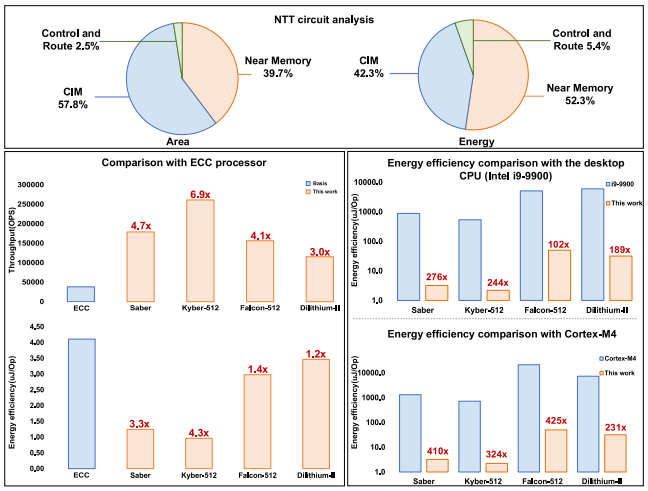
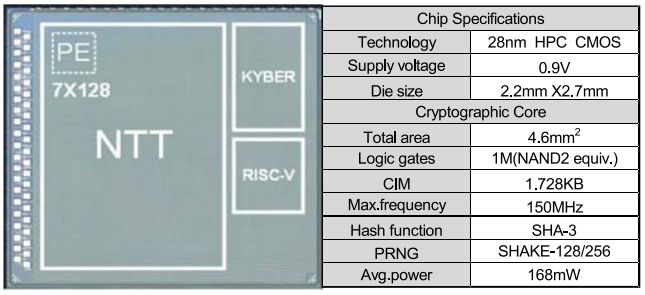


Fig. 6. PQC simulation performance comparison.



Chip Specifications	
Technology	28nm HPC CMOS
Supply voltage	0.9V
Die size	2.2mm X2.7mm
Cryptographic Core	
Total area	4.6mm^2
Logic gates	1M(NAND2 equiv.)
CIM	1.728KB
Max.frequency	150MHz
Hash function	SHA-3
PRNG	SHAKE-128/256
Avg.power	168mW

Fig. 7. Chip micrograph and performance summary.

n = 256 and q = 13. It employs an efficient Keccak1600 core which supports SHA3-256, SHA3-512, SHAKE-128 and SHAKE-256 modes to concurrently process a 1600-bit state in parallel, generating hashes, pseudo-random numbers (PRNG), and driving discrete distribution samplers simultaneously. All computing modules, including compression and encoding, utilize high-bit-width parallel computing techniques. Atomic operations are conducted using a 12-bit standard and employ 256-way parallel operations to deliver a combined computing and storage bandwidth of 3072 bits. Intermediate computational data can be cached within the storage array of the CIM-NTT architecture, thereby substantially reducing storage consumption and facilitating high-performance computing with ultra-high bandwidth.

The scheduling state machine enables a data flow resembling a pipeline during key generation, encryption, and decryption processes. The algorithm's different links, such as NTT, centered binomial distribution (CBD), and Parse, have varying operation cycles, necessitating the scheduling of these links to achieve balanced delays. In the implementation of the specific algorithm circuit, the three computational processes—key generation, encryption, and decryption—are tightly integrated within the hardware architecture. These processes are designed to be pipelined and time-division multiplexed across each operational module, ensuring efficient resource utilization and enhanced overall performance, resulting in a remarkable increase in throughput to 75.6 KOPS.

III. Evaluation

As shown in Fig. 6, a simulation-based performance comparison with ECC processors and PQC software implementations for supported algorithms. The evaluated circuit operates at a frequency of 150 MHz and a voltage of 0.9 V. It utilizes a hybrid design that combines the CIM circuit, near-memory logic, and peripheral circuits, with the CIM circuit delivering superior energy efficiency per unit area. In our study, the NTT-based implementation of PQC achieves 3.0-6.9×higher throughput compared to existing ECC processors, while also exhibiting 1.2-4.3x higher energy efficiency. Moreover, when compared to desktop CPU software implementations, energy efficiency is improved by 102-276x, and compared to embedded CPU software implementations, energy efficiency is improved by 231-425x.

IV. 28-nm CMOS Measurement

This chip integrates multiple circuits, including CIM-NTT accelerator, RISC-V processor, and Kyber algorithm hardware unit. It is fabricated using a 28 nm HPC CMOS process, has a size of 2.2mm x 2.7mm, and operates at a frequency of 150MHz. The area of the CIM-NTT introduced in this brief is 4.6mm^2. It incorporates a 1.728 KB CIM and 1 million equivalent gates. Further details regarding the chip's characteristics and a die photo are presented in Fig. 7.

Table I displays the measured results and performance comparison of previous NTT accelerators for PQC in general. Our work demonstrates the average power consumption of 168 mW and energy efficiency of 2.2 μJ/Op when executing the Kyber-512 algorithm. The NTT clock cycles is only one-tenth of that in [7]. The implementation of 256-point NTT algorithm achieves an energy consumption of only 13 nJ, 47.6% lower than [18], while achieves a remarkable reduction of up to 78% in clock cycles and a 45% reduction in energy consumption than [17]. The data throughput of NTT is 11x than [19]. When compared to [20], our work achieves significantly smaller latency cycles of 241x, respectively, and

TABLE I
COMPARISON TO PRIOR WORKS

	CICC'18[19]	ISSCC'19[20][a]	TCAS-1'20[18][e]	DATA'2021[7]	ISSCC'22[17]	ESSCIRC'23[21]	**This work**
Technology	40nm	40nm	28nm	Artix-7	28nm	28nm	**28nm**
Frequency (MHz)	300	12-72	300	172	500	0.5-157	**150**
Area (mm^2)	2.05	0.28	-	-	3.6	1.69	**4.6**
Voltage (V)	0.9	0.68-1.1	0.9	-	0.9	0.64-1.1	**0.9**
Power (mW)	140	7~10	~30	-	39~368	12.58	**168**
Multi-mathematical problems	-	Lattice	Lattice	Lattice	Lattice, code, multivariate	Lattice	**Lattice**
Supported Lattice Crypto Primitives	Ring-LWE	Ring-LWE & Module-LWE	Ring-LWE & Module-LWE	Module-LWE	Ring-LWE & Module-LWE	FHE(BFV)	**Module-LWE**
Supported Lattice Parameters	N: 64-2048 q: 32-bit conf.	N: 64-2048 q: 24-bit conf.	N: 64-2048 q: 24-bit conf.	N: 256 q: 12-bit conf.	q: 24-bit conf.	N: 4096 q: 109-bit conf.	**N:64-2048 q: 24-bit conf.**
NIST 3-rd round standardization	-	Kyber, Dilithium	Kyber	Kyber	Kyber, Dilithium	-	**Kyber**
NTT Performance (N=256)							
Cycles	160	1288	45	69	32	12303	**1-7**
Energy (nJ)	31	63.4	24.6	-	2.5-23.6[f]	2.58	**13**
Kyber-512 scheme (keygen + encaps. + decaps.)							
Throughput (KOPS)	-	0.2[b]	2.1[c]	-	47.9	-	**75.6**
Energy eff. (uJ/OP)	-	26.6[b]	12.8[c]	-	3.4	-	**2.2**
Latency (cycles)	-	479644	-	-	10433	-	**1986**

a:some data are collected from the extended paper b:the NIST-round 1 variants c:the NIST-round 2 variants d:the data normalized to 28 nm process e:post-simulation results. Not silicon verified f: estimated from minimum power consumption and number of clock cycle.

provides 12x higher energy efficiency when running Kyber-512. The processing efficiency of NTT is 1575x over the design [21].

V. CONCLUSION

This brief presents a CIM configurable-pipeline NTT accelerator for PQC. The accelerator incorporates a bidirectional pipeline array to minimize data latency, CIM processing elements to reduce memory access, and a parallel PQC circuit for lattice-based cryptographic protocol deployment. Overall, our proposed CIM-NTT has high processing capabilities and energy efficiency and is suitable for IoT devices.

REFERENCES

[1] H. Nejatollahi, N. Dutt, S. Ray, F. Regazzoni, I. Banerjee, and R. Cammarota, "Post-quantum lattice-based cryptography implementations: A survey," *ACM Comput. Surv.*, vol. 51, no. 6, pp. 1–41, 2019.

[2] G. Alagic et al., *Status Report on the Third Round of the NIST Post-Quantum Cryptography Standardization Process*, Nat. Inst. Stand. Technol., Gaithersburg, MD, USA, 2022.

[3] O. Regev, "On lattices, learning with errors, random linear codes, and cryptography," *J. ACM*, vol. 56, no. 6, pp. 1–40, Sep. 2009.

[4] V. Lyubashevsky, C. Peikert, and O. Regev, "On ideal lattices and learning with errors over rings," in *Proc. Annu. Int. Conf. Theory Appl. Cryptogr. Techn.*, 2010, pp. 1–23.

[5] M. Bisheh-Niasar, R. Azarderakhsh, and M. Mozaffari-Kermani, "A monolithic hardware implementation of kyber: Comparing apples to apples in PQC candidates," in *Proc. Int. Conf. Cryptol. Inf. Secur. Latin America*, 2021, pp. 108–126.

[6] X. Feng, S. Li, and S. Xu, "RLWE-oriented high-speed polynomial multiplier utilizing multi-lane stockham NTT algorithm," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 67, no. 3, pp. 556–559, Mar. 2020.

[7] F. Yaman, A. C. Mert, E. Öztürk, and E. Savaş, "A hardware accelerator for polynomial multiplication operation of CRYSTALS-KYBER PQC scheme," in *Proc. Design, Autom. Test Eur. Conf. Exhib. (DATE)*, Grenoble, France, 2021, pp. 1020–1025.

[8] Y. Xing and S. Li, "A compact hardware implementation of CCA-secure key exchange mechanism CRYSTALS-KYBER on FPGA," *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, vol. 2021, no. 2, pp. 328–356.

[9] Z. Ye, R. C. C. Cheung, and K. Huang, "PipeNTT: A pipelined number theoretic transform architecture," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 69, no. 10, pp. 4068–4072, Oct. 2022.

[10] B. Kim, J. Park, S. Moon, K. Kang, and J.-Y. Sim, "Configurable energy-efficient lattice-based post-quantum cryptography processor for IoT devices," in *Proc. IEEE 48th Eur. Solid State Circuits Conf. (ESSCIRC)*, Milan, Italy, 2022, pp. 525–528.

[11] N. Verma et al., "In-memory computing: Advances and prospects," *IEEE Solid State Circuits Mag.*, vol. 11, no. 3, pp. 43–55, Aug. 2019.

[12] Z. Tong, Y. Zhao, J. Zhang, Z. Lin, X. Lin, and X. Wu, "In-memory transposable multibit multiplication based on diagonal symmetry weight block," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 31, no. 9, pp. 1454–1458, Sep. 2023.

[13] M. Imani et al., "FloatPIM: In-memory acceleration of deep neural network training with high precision," in *Proc. ACM/IEEE 46th Annu. Int. Symp. Comput. Archit. (ISCA)*, 2019, pp. 802–815.

[14] J. Ahn et al., "A scalable processing-in-memory accelerator for parallel graph processing," *ACM SIGARCH Comput. Archit. News*, vol. 43, no. 35, pp. 105–117, 2016

[15] D. Lavenier, J.-F. Roy, and D. Furode, "DNA mapping using processor-in-memory architecture," in *Proc. IEEE Int. Conf. Bioinformat. Biomed.*, 2016, pp. 1429–1435.

[16] H. Nejatollahi, S. Gupta, M. Imani, T. S. Rosing, R. Cammarota, and N. Dutt, "Crypto PIM: In-memory acceleration for lattice-based cryptographic hardware," in *Proc. 57th ACM/IEEE Design Autom. Conf. (DAC)*, Jul. 2020, pp. 1–6.

[17] Y. Zhu et al., "A 28nm 48KOPS 3.4 μJ/Op agile crypto-processor for post-quantum cryptography on multi-mathematical problems," in *Proc. ISSCC*, 2022, pp. 514–516.

[18] G. Xin et al., "VPQC: A domain-specific vector processor for post-quantum cryptography based on RISC-V architecture," in *Proc. IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 67, no. 8, pp. 2672–2684, Aug. 2020.

[19] S. Song, W. Tang, T. Chen, and Z. Zhang, "LEIA: A 2.05mm^2 140mW lattice encryption instruction accelerator in 40nm CMOS," in *Proc. IEEE CICC*, 2018, pp. 1–4.

[20] U. Banerjee, A. Pathak, and A. P. Chandrakasan, "An energy-efficient configurable lattice cryptography processor for the quantum-secure Internet of Things," in *Proc. ISSCC*, 2019, pp. 46–48,

[21] S. Das, M. V. D. Hagen, S. Patil, C. Erbagci, B. Lucia, and K. Mai, "A 10.33 μJ/encryption homomorphic encryption engine in 28nm CMOS with 4096-degree 109-bit polynomials for resource-constrained IoT clients," in *Proc. ESSCIRC*, 2023, pp. 193–196.