ARM architecture optimizations for line-rate PQC communications

1st A. Cano Aguilera
Network SW and Systems Research
NVIDIA Corporation
2066730 Yokneam Illit, Israel
acanoaguiler@nvidia.com

2th I. Tafur Monroy
Department of Electrical Engineering
Technical University of Eindhoven
5612 AP Eindhoven, The Netherlands
i.tafur.monroy@tue.nl

3th J. J. Vegas Olmos
Network SW and Systems Research
NVIDIA Corporation
2066730 Yokneam Illit, Israel
juanj@nvidia.com

4th José L. Imaña
Arquitectura de Computadores y Automática
Universidad Complutense de Madrid
28040 Madrid, Spain
jluimana@ucm.es

***Abstract*—This paper provides an introduction to the topic of ARM architecture optimization for line-rate post-quantum cryptographic (PQC) operations. In particular, we explore ARMv8 architectures and how to leverage hash functions. As quantum computing threatens traditional public-key infrastructure (PKI), the need for efficient quantum-resistant algorithms grows. The NIST PQC standardization process has chosen (until now) ML-DSA (Crystals-Dilithium) with extendable output functions (XOFs) from the SHA3 standard, specifically SHAKE128 and SHAKE256. Many of these standards have already been included into retail systems, while silicon fabs are providing dedicated PQC accelerators for low-speed systems. This paper and its presentation broadens the scope of XOFs in the Dilithium framework by incorporating alternatives like concatenated fixed variable length hashes such as SHA256, SHA512, ASCON and AES-CTR. Our current investigations lead to substantial performance enhancements when ARMv8 acceleration is applied using single instruction - multiple data (SIMD) instructions via the NEON framework. In particular, we will discuss improvements in the KeyGeneration, Signature, and Verification steps across different security parameterizations of ML-DSA in comparison with the reference code of the standard.**

***Index Terms*—post-quantum cryptography, acceleration, data centers.**

I. Introduction

The advent of quantum computers with substantial processing capabilities poses a serious threat to our current public key infrastructure (PKI). This is primarily due to their ability to execute Shor's algorithm, which jeopardizes the security of widely used cryptographic schemes based on the discrete logarithm problem—such as Diffie-Hellman key exchange, ECDH [1], and the elliptic curve digital signature algorithm, ECDSA [2]—as well as those relying on integer factorization, such as RSA.

The work was supported in part by QUARC with grant number 101073355, SMARTY with grant number 101140087 and CLEVER project with grant agreement 101097560 and in part by grant PID2021-123041OB-I00 funded by MCIN/AEI/10.13039/501100011033 and by "ERDF A way of making Europe".

Recognizing the possibility that such threats may materialize sooner than expected, the research community has proactively sought to mitigate these risks. The National Institute of Standards and Technology (NIST) initiated a competition [3] to identify cryptographic algorithms resilient to quantum attacks. In 2022, NIST announced the final candidates for standardization, and in 2023, it selected the winners. ML-KEM [4] (formerly CRYSTALS-Kyber) was chosen as the standard key encapsulation mechanism (KEM), while three digital signature algorithms (DSAs) were selected: ML-DSA [5] (CRYSTALS-Dilithium), SLH-DSA [6] (SPHINCS+), and Falcon [7]. Despite relying on the well-established hardness of the hash-collision problem, SLH-DSA has demonstrated subpar performance on various platforms. Consequently, ML-KEM and Dilithium have emerged as the leading candidates for post-quantum secure communications, with Falcon serving niche use cases requiring rapid verification [8].

In parallel, NIST launched the Lightweight Cryptography (LWC) competition [9] to standardize authenticated encryption and hashing algorithms suitable for constrained devices. In early 2023, Ascon [10] was named the winner. While both competitions address key cryptographic challenges, their intersection—developing efficient, quantum-resistant solutions for resource-limited environments—remains insufficiently explored.

Hardware acceleration plays a vital role in enhancing algorithm performance. However, designing custom hardware such as field-programmable gate arrays (FPGAs) can be costly and labor-intensive. Moreover, latency introduced by external communication between FPGAs and general-purpose processors can offset performance benefits due to memory access delays. A more practical alternative involves leveraging instruction set extensions (ISEs) in general-purpose processors. These extensions enable on-chip acceleration of computationally intensive tasks, reduce data transfer overhead between the CPU and peripherals, and eliminate the need for dedicated communication controllers—all of which are

particularly valuable for constrained platforms.

The shift from classical PKI to quantum-resistant alternatives presents both technical and strategic challenges. As outlined in NIST's guidance on transitioning to post-quantum cryptography standards [11], PKI is foundational to numerous applications—including network security protocols, secure email, digital document signing, and hybrid key exchange methods [12]. These applications span diverse hardware platforms, from high-performance servers equipped with GPUs to ultra-constrained IoT devices. Across all systems, three operations emerge as performance bottlenecks: polynomial multiplication, modular reduction, and Keccak-based hashing.

While significant progress has been made in accelerating polynomial arithmetic—for instance, through the Neon-NTT project [13], which optimized Kyber, Dilithium, and Saber on ARM Cortex-M3 and A72 processors, doubling performance in some cases—hashing remains the most computationally expensive component in lattice-based PQC. In particular, Keccak-based extendable-output functions (XOFs) like SHAKE [14] can dominate execution time, accounting for up to 85% of runtime in software implementations. Despite extensive work on optimizing Keccak on ARM platforms, achieving parity with AES and SHA-2 performance remains a challenge.

This context underscores the critical need for efficient hashing techniques in post-quantum cryptography. Although FIPS 202 [14] mandates the use of SHAKE in NIST's PQC standards, alternative bitstream generation methods have been explored. Notably, the CRYSTALS team proposed an AES-CTR-based bit random generator (BRG) during the third round of standardization [15], along with an AVX2-optimized implementation leveraging 256-bit SIMD on x86 architectures. Other investigations have examined the potential of TurboSHAKE [16] and Ascon as BRGs for Dilithium. However, no prior studies have thoroughly evaluated these alternatives in the context of Dilithium itself.

This paper will discuss the following contributions:

1) We present the first parametrization of Dilithium with a sponge construction other than SHAKE, specifically Ascon, which is well-suited for constrained devices.
2) First study of round-reduced Keccak functions (TurboSHAKE128 and TurboSHAKE256) in Dilithium.
3) The use of AES for matrix and vector sampling in Dilithium on ARMv8 architectures. Previous work focused primarily on x86, particularly AVX2-based AES instructions.
4) Novel construction for arbitrary-output BRGs based on fixed-length hash functions (e.g., SHA-256, SHA-512)

II. Experimental Results and Discussion

A. Benchmark settings

For our benchmarks, we used the ARMv8 processors from a Data Processing Unit, where all the tests were run on local mode. All implementations were compiled using aarch64-linux-gnu gcc version 11.4.0. For all tests, we used the cycle counter register *cntvct_el0* to measure the clock cycles.

DPU: NVIDIA BF2 (model MBF2H516ACEEOT) with 8 ARMv8 A72 cores dedicated to specific operations. These 8 cores can operate at a clock rate between 550 and 2750 MHz. We run a test over the three different versions of Dilithium, namely Dilithium2, Dilithium3 and Dilithium5, whose security equivalent is resumed in table I.

TABLE I
Dilithium Security Equivalent Computational Problems

Level	Security Description
II	At least as hard to break as SHA-256 (Collision search)
III	At least as hard to break as AES-192 (Collision search)
V	At least as hard to break as AES-256 (Exhaustive key search)

We analized all the constructions under analysis and instantiated them as in Table II.

B. Analysis

In lattice-based cryptographic schemes, the selection of parameters is primarily guided by considerations of computational efficiency. One of the most computationally demanding operations in these schemes is the expansion of polynomial vectors and matrices over the ring R_q, which requires multiple invocations of hash functions. In contrast, operations such as message commitment typically require only a single hash function call. This pronounced disparity in computational cost informed our decision to adopt the approach introduced by the CRYSTALS team in [15], where expansion functions are implemented using AES-CTR to improve performance, while SHAKE is preserved for the core signature generation and verification processes.

Table II shows the performance of Dilithium key functions, namely *KeyGen, Sign* and *Verify*.

For each of the candidate functions, we structure our evaluation across the three security levels defined by Dilithium: Dilithium2, Dilithium3, and Dilithium5. SHAKE is used as the performance baseline, given its role as the default XOF in Dilithium and its endorsement by NIST. We assess and compare the performance of SHAKE against several alternatives, including TurboSHAKE, a round-reduced variant, AES-CTR as defined in [15], as well as SHA-256, SHA-512. The comparison is based on key performance metrics such as execution time, throughput, and cycle count across the different stages of key generation, signing, and verification.

Table II presents the performance analysis of Dilithium's key functions: *KeyGen, Sign*, and *Verify*. For each function, we divide our data into three sections based on the security parameters of Dilithium, namely Dilithium2, Dilithium3, and Dilithium5. SHAKE is used as the baseline for comparison since it is the XOFs employed in Dilithium and recommended by NIST. We then compare SHAKE against TurboSHAKE, a round-reduced variant, and AES-CTR, as defined in [15]. Additionally, we evaluate SHA256 and SHA512.

The performance analysis of Dilithium2, Dilithium3, and Dilithium5 confirms the expected trade-off between security and efficiency. As we progress from Dilithium2 to Dilithium5,

TABLE II
CRYPTOGRAPHIC PRIMITIVES AND THEIR EXPANSION APPROACHES

Primitive	Approach	Expand A	Expand S	Expand Y	Rest
SHAKE	Sponge	SHAKE128	SHAKE256	SHAKE256	SHAKE256
TURBOSHAKE	Sponge	TURBOSHAKE128	TURBOSHAKE256	TURBOSHAKE256	TURBOSHAKE256
ASCON	Sponge	ASCON128	SHAKE256	SHAKE256	SHAKE256
AES	Block cipher CTR mode	AES256-CTR	AES256-CTR	AES256-CTR	SHAKE256
SHA	Hash	SHA256	SHA512	SHA512	SHAKE256

Algorithm	SHAKE	TurboSHAKE	SHA2	AES-CTR	ASCON
Keygen (Reference)					
Dilithium2	36870	29112 (**-21.04%**)	88344 (139.81%)	93135 (153.35%)	58824 (59.64%)
Dilithium3	65880	51828 (**-21.41%**)	160704 (143.83%)	170031 (158.77%)	108351 (64.75%)
Dilithium5	100827	76347 (**-24.68%**)	272271 (170.55%)	284982 (182.79%)	175170 (73.70%)
Signature (Reference)					
Dilithium2	130224	116988 (**-10.14%**)	190032 (46.00%)	209196 (60.64%)	162807 (24.99%)
Dilithium3	220683	200712 (**-9.04%**)	302856 (37.23%)	359136 (62.82%)	286593 (29.83%)
Dilithium5	275829	243558 (**-11.73%**)	466695 (69.18%)	484476 (75.65%)	366708 (32.90%)
Verification (Reference)					
Dilithium2	40920	33366 (**-18.42%**)	89004 (117.70%)	89184 (118.07%)	61350 (50.01%)
Dilithium3	65259	52293 (**-19.89%**)	155091 (137.00%)	155103 (137.00%)	103566 (58.76%)
Dilithium5	106173	82842 (**-22.62%**)	276132 (160.07%)	273747 (157.70%)	177876 (67.78%)
Keygen (Optimized)					
Dilithium2	36870	29112 (**-21.04%**)	34194 (**-7.29%**)	23196 (**-37.13%**)	-
Dilithium3	65880	51828 (**-21.41%**)	59190 (**-10.13%**)	40656 (**-38.24%**)	-
Dilithium5	100827	76347 (**-24.68%**)	87102 (**-13.14%**)	56220 (**-44.23%**)	-
Signature (Optimized)					
Dilithium2	130224	116988 (**-10.14%**)	135564 (4.07%)	107943 (**-17.12%**)	-
Dilithium3	220683	200712 (**-9.04%**)	231513 (4.91%)	183495 (**-16.85%**)	-
Dilithium5	275829	243558 (**-11.73%**)	280860 (2.06%)	218622 (**-20.77%**)	-
Verification (Optimized)					
Dilithium2	40920	33366 (**-18.42%**)	35601 (**-12.99%**)	28833 (**-29.53%**)	-
Dilithium3	65259	52293 (**-19.89%**)	53373 (**-18.23%**)	43026 (**-34.06%**)	-
Dilithium5	106173	82842 (**-22.62%**)	88083 (**-17.15%**)	64719 (**-39.04%**)	-

TABLE III
ARM CPU CYCLES FOR DILITHIUM2, DILITHIUM3, DILITHIUM5 WITH REFERENCE AND NEON-BASED IMPLEMENTATIONS.

cryptographic strength increases at the expense of higher computational requirements. Since Dilithium relies on SHAKE as its baseline, we use it as our reference point for comparison. The results show that SHAKE provides balanced performance across all variants, making it the preferred choice for robust and standardized implementations. However, the increasing computational cost from Dilithium2 to Dilithium5 highlights the additional overhead incurred when increasing security parameters, particularly in key generation and signature verification. TurboSHAKE, as a round-reduced version of SHAKE, achieves speed-ups across all operations. The most significant improvement is observed in key generation, where it offers a 21–25% reduction in computation time. However, key generation is the least critical function in Dilithium, as public and secret keys are typically computed only once in a standard PKI and remain valid for extended periods. For signature generation, the performance gain remains around 10% across all three security levels, while verification sees a more substantial improvement, ranging from 19% for Dilithium2 to 22% for Dilithium5. These results indicate that reducing SHAKE's rounds enables faster certificate signing and verification with minimal effort.

More noticeable performance differences emerge when comparing alternative hash functions, particularly AES-CTR and SHA256. When these are used in their reference implementations, performance degrades significantly. Key generation times for SHA2 and AES-CTR more than double across all cases, with SHA2 exhibiting the best relative performance. This behavior arises from the fact that both AES and SHA2 are heavily reliant on hardware acceleration.

Signature computation fares slightly better, with SHA-256 incurring a slowdown of 37.23–69.18% and AES-CTR ranging from 60–75% slower than SHAKE. However, when hardware acceleration is enabled, performance improves drastically for SHA256 and AES-CTR. In Figure table II we can see how SHA256 surpasses SHAKE128 in both key generation and verification, achieving a 7–13% speed-up for key generation and a 13–17.15% improvement in verification. However, signature generation remains slower than SHAKE128.

Finally, AES-CTR emerges as the fastest hashing method when optimizations are applied. For Dilithium2, performance improves by 37.13%, 17.12%, and 29.53% in key generation, signing, and verification, respectively, when ARM NEON AES instructions are used. Similar trends are observed for Dilithium3 and Dilithium5, with the most significant gains achieved in Dilithium5, where key generation improves by 37.13%, signature generation by 17.12%, and verification by 29.53%.

During the presentation, we will discuss the following conclusions:

- Hardware acceleration is crucial to improving the performance of PQC algorithms.
- Verification benefits more from optimization than signature generation.
- AES-CTR is the most efficient XOF for Dilithium when using NEON instructions.
- The use of fixed-variable output hashes can improve Dilithium's performance, as demonstrated by SHA256. Exploring even faster primitives may yield better results.

III. Conclusion

This work presents, for the first time, a comprehensive exploration of alternative hash constructions aimed at improving the performance of Dilithium (CRYSTALS-Dilithium) on ARMv8 architectures. Our findings demonstrate that substantial performance enhancements can be achieved by strategically selecting or substituting the XOFs employed in lattice-based PQC.

We show that replacing SHAKE with TurboSHAKE can yield performance improvements ranging from 9% to 25%, underscoring that reducing the number of Keccak rounds offers tangible benefits when SHAKE is required. Furthermore, our results indicate that AES-CTR emerges as the most efficient option when AES hardware acceleration is available, enabling up to 44% faster key generation and 29% faster verification compared to the SHAKE-based baseline.

Additionally, traditional hash functions such as SHA-256, despite their fixed-length output, can also provide notable gains—up to 17% improvement in verification when utilizing NEON instructions. This suggests promising avenues for further optimization and the potential adoption of more efficient quantum-resistant hash functions within Dilithium.

Finally, although sponge-based constructions naturally integrate with Dilithium, our evaluation reveals that ASCON—the only sponge-based function assessed apart from Keccak—did not outperform SHAKE. Nonetheless, given that ASCON targets constrained environments, it may offer improved performance on 32-bit ARM architectures or other resource-limited platforms in terms of computational efficiency and code size. During the presentation, we will provide a comprehensive overview of PQC technologies, current state-of-the-art on the acceleration techniques to reduce the complexity of said systems, and a deeper overview of our research activities in this field.

References

[1] Elaine Barker et al. Recommendation for pair-wise key-establishment schemes using discrete logarithm cryptography, 2018-04-16 2018.

[2] National Institute of Standards and Technology. Digital signature standard (dss). FIPS Publication 186, May 1994.

[3] National Institute of Standards and Technology (NIST). Announcing approval of three federal information processing standards (fips) for post-quantum cryptography. https://csrc.nist.gov/news/2024/postquantum-cryptography-fips-approved, August 13 2024. Accessed: 2024-10-23.

[4] National Institute of Standards and Technology. Module-lattice-based key encapsulation mechanism standard. Federal Information Processing Standards Publication NIST FIPS 203 ipd, Department of Commerce, Washington, D.C., 2023.

[5] National Institute of Standards and Technology. Module-lattice-based digital signature standard. Federal Information Processing Standards Publication NIST FIPS 204 ipd, Department of Commerce, Washington, D.C., 2023.

[6] National Institute of Standards and Technology. Stateless hash-based digital signature standard. Federal Information Processing Standards Publication NIST FIPS 205, Department of Commerce, Washington, D.C., 2024.

[7] Pierre-Alain Fouque, Jeffrey Hoffstein, Paul Kirchner, Vadim Lyubashevsky, Thomas Pornin, Thomas Prest, Thomas Ricosset, Gregor Seiler, William Whyte, and Zhenfei Zhang. Falcon: Fast-fourier lattice-based compact signatures over ntru. Technical report, 2020.

[8] Abraham Cano Aguilera, Carlos Rubio Garcia, Daniel Lawo, José Luis Imaña, Idelfonso Tafur Monroy, and Juan José Vegas Olmos. In-line rate encrypted links using pre-shared post-quantum keys and dpus. *Scientific Reports*, 14(1), September 2024.

[9] National Institute of Standards and Technology. Lightweight cryptography, n.d. Accessed: 2023-10-30.

[10] Meltem Sönmez Turan, Kerry A. McKay, Donghoon Chang, Jinkeon Kang, and John Kelsey. Ascon-based lightweight cryptography standards for constrained devices. NIST Special Publication (SP) NIST SP 800-232 ipd, National Institute of Standards and Technology, Gaithersburg, MD, 2024.

[11] National Institute of Standards and Technology. Interagency report 8547: [insert title here]. NIST Interagency/Internal Report (NISTIR) 8547, U.S. Department of Commerce, 2024. Accessed: [Insert Date Here].

[12] A. Cano Aguilera, R. Abu Bakar, F. Alhamed, C. Rubio Garcia, J.L. Imaña, I. Tafur Monroy, F. Cugini, and J.J Vegas Olmos. First line-rate end-to-end post-quantum encrypted optical fiber link using data processing units (dpus). In *Optical Fiber Communication Conference (OFC) 2024*, United States, March 2024. Optica Publishing Group. 2024 Optical Fiber Communications Conference and Exhibition (OFC) ; Conference date: 24-03-2024 Through 28-03-2024.

[13] Hanno Becker, Vincent Hwang, Matthias J. Kannwischer, Bo-Yin Yang, and Shang-Yi Yang. Neon ntt: Faster dilithium, kyber, and saber on cortex-a72 and apple m1. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2022(1):221–244, Nov. 2021.

[14] Morris J. Dworkin. Sha-3 standard: Permutation-based hash and extendable-output functions:, 2015-07-01 04:07:00 2015.

[15] Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, Peter Schwabe, Gregor Seiler, and Damien Stehlé. CRYSTALS-Dilithium – Algorithm Specifications and Supporting Documentation (Version 3.1), feb 2021. Specification document (update from February 2021).

[16] Guido Bertoni, Joan Daemen, Seth Hoffert, Michaël Peeters, Gilles Van Assche, Ronny Van Keer, and Benoît Viguier. TurboSHAKE. Cryptology ePrint Archive, Paper 2023/342, 2023.