A Hardware Evaluation Study of NIST Post-Quantum Cryptographic Signature schemes

Deepraj Soni[1], Kanad Basu[1], Mohammed Nabeel[2], and Ramesh Karri[1]
[1]New York University, New York, NY, USA
[2]New York University - Abu Dhabi, Abu Dhabi, UAE

***Abstract*—As we are moving into the quantum era, classical cryptography is under risk, since quantum computers can break these complex cryptographic algorithms [1]. Researchers are developing the post-quantum cryptographic (PQC) algorithms to secure the system against quantum computers. The National Institute of Standards and Technology (NIST) started a public evaluation process to standardize quantum-resistant public key algorithms. The objective of this study to provide hardware-based comparison of the NIST Round-2 PQC signature schemes. For this, we use a High-Level Synthesis (HLS)-based hardware design methodology to map high-level C specifications of signature-based PQC algorithms into FPGA implementations.**

I. Introduction

The fundamental security protocol for any form of electronic communication is public key cryptography. The two main cryptographic functions are: (a) Public Key Encryption and (b) Digital Signature [2]. Classical public key algorithms depend upon a non-polynomial time difficult problem (i.e., integer factorisation or discrete log problem). Peter Shor from Bell Labs showed that quantum computers can solve these problems in polynomial time. Hence, quantum computers can decrypt the communication secured by classical public key cryptography algorithms [1]. For secure digital communications in the future, scientists are developing secure alternatives known as Post-Quantum Cryptography (PQC) algorithms.

NIST PQC standardization process is consolidating the candidates that are inscribed by the cryptographic community and organisations. Each candidate in this ongoing NIST process implements one of two functions: digital signature or key encapsulation mechanism (KEM). The first round of this process had 82 submissions. 17 KEM and 9 Signature schemes are chosen for round-2 based on the security strength and performance. In the ongoing round-2, the algorithms are tested and benchmarked for different attacks, software performance, and hardware performance. Security guarantees, performance on software/hardware, power utilization, and area overhead are the yardsticks for algorithm selection.

Hardware realizations of PQC algorithms can be performed by either manual RTL or HLS-based implementation. Recent research has shown that HLS-based approaches can compete with the RTL-based implementations [3]. HLS has been used in evaluating classical cryptographic algorithms in CAESAR competition [4]. For AES cryptographic core, implemented in HLS, latency and area overhead are 20% more and 3% less compared to RTL-based implementation. Till date, there is no standardization procedure regarding the hardware implementation of Round-2 PQC algorithms. This is the first paper that implements and explores the design-space of multiple signature schemes in hardware using HLS and compares the results.

We report a hardware-implementation comparison of qTESLA and CRYSTALS-Dilithium NIST round-2 PQC signature schemes. The contributions of this study are:

1) Developed systematic FPGA design flows for PQC evaluation, starting from a C specification.
2) Implemented HLS-based hardware design for 2 PQC signature schemes.
3) Improved the latency of PQC implementations using optimizations such as loop unrolling and loop pipelining.
4) Design-space exploration of key generation, signature generation and signature verification for PQC signature algorithms.
5) Performed a detailed study of two signature algorithms to explore area vs performance vs security trade-offs.

II. Post-Quantum Cryptography

Scientists are developing cryptographic algorithms that are robust against quantum computers. Based on the underlying mathematical problem, the PQC algorithms are classified in: Lattice-based Cryptography, Code-based Cryptography, Multivariate polynomial cryptography, Hash-based digital signatures, isogeny-based and other methods. This paper focuses on lattice-based signature schemes across different security levels.

A. Lattice-based cryptography

These algorithms can challenge the best known alternatives [5] Lattice-based cryptography builds on the hardness of the shortest vector problem (SVP), Closest Vector Problem (CVP) or Shortest Independent Vectors Problem (SIVP). SVP tries to find minimal possible Euclidian length of an n-dimensional lattice vector. Therefore, breaking a lattice-based cryptography algorithm is the equivalent of solving SVP. Even with a quantum computer, SVP is shown to be polynomial in n [2]. Other lattice cryptography algorithms are based on the Short Integer Solutions (SIS). If the SVP is hard in the worst-case, SIS is secure in the average case [6].

B. Digital signature

The digital signatures identify the authenticity of the sender who has a secret key. Sender generates both secret key and public key. The receiver uses the public key to verify the signature signed by the sender with the secret key. An attacker cannot replicate the signed message without the secret key. The PQC digital signature algorithms consist of three functions: key-pair generation, signature generation and signature verification. According to NIST standard, the key-pair generation is realized in the function "crypto_sign_keypair" which generates the public key pk and the secret key sk. Signature generation is realized in the function "crypto_sign" which takes secret key sk, the message m plus its length $mlen$ as inputs and outputs the signature sm and signature length sm_len. Signature verification is realized in the function "crypto_sign_open" which takes public key pk, signature sm and signature length $smlen$ as inputs and outputs message m and its length $mlen$.

III. PQC Hardware Assessment

A. HLS-based Assessment Methodology

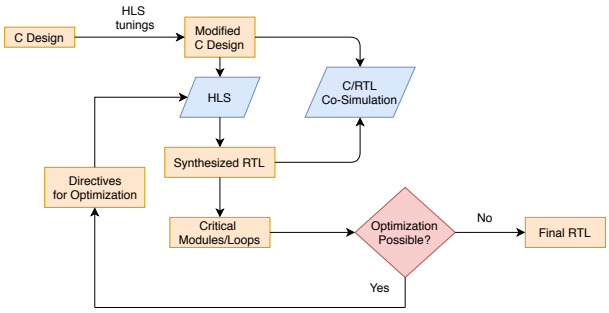


Fig. 1: HLS-based design exploration of PQC algorithms.

The HLS design exploration flow for PQC algorithms is shown in Figure 1. We modify the original NIST-submitted C code to make it HLS-suitable (e.g., replace library function, change complex hierarchy of structure, modify variable length array and pointers to a fixed dimension arrays, remove recursions, etc.). Next, we perform HLS on the synthesizable C code to generate RTL using Xilinx Vivado HLS. Vivado provides a synthesis report identifying which modules/loops in the design results in longest latency. If there are loops or functions with a latency of large number of clock cycles, we optimize them using loop unrolling and pipelining. Vivado reports the estimated clock period for synthesized designs. The tool estimates the clock period based on worst-case delay.

B. PQC Algorithms in this Study

We evaluated keypair generation, signature generation and signature verification of 2 NIST PQC algorithms–qTESLA and CRYSTALS-Dilithium. Other PQC algorithms are under the same design-flow process at different stages [7]. As a target device, we chose Xilinx Artix-7 FPGA, a standard FPGA board accepted by PQC community for hardware implementation.

C. Performance metrics

We consider latency, area and security as performance indicators. Latency is the number of clock cycles needed to produce the output from the time the input is provided. Throughput of a system is measured with initiation interval (II). II is the minimum number of clock cycles between two successive inputs. The PQC algorithms are verified with Known Answer Tests (KAT). The latency and II are the same for a single test case. Therefore, a design with lower latency indicates better throughput.

We use latency to measure the performance of the designs. We use Flip-Flops and Look-up Tables (LUTs) for resource utilisation of FPGA implementations. NIST security categories are used for security strength of the PQC algorithm.

IV. Results and Analysis

A. Baseline hardware implementations

Scheme	Sec. Level	Optimisation	FF	LUT	Clock (nsec)	Latency
qTESLA	1	Baseline	22456	108764	12.65	622681
		Unrolling	31491	125942	12.65	604123
		Pipelining	22450	108880	12.65	616527
	2	Baseline	31325	127311	14.179	18759797
		Unrolling	32707	129840	14.179	18655029
		Pipelining	173157	333958	15.1	17612174
	3	Baseline	23393	111001	12.65	3642422
		Unrolling	30994	124724	12.65	3608070
		Pipelining	23398	111122	12.65	3608510
	5	Baseline	23820	114021	12.65	32358252
		Unrolling	31752	128554	12.65	32057906
		Pipelining	23837	114152	12.65	32039615
CRYSTALS -Dilithium	1	Baseline	17783	86465	8.375	114822
		Unrolling	31401	127045	9.682	116256
		Pipelining	17310	86662	33.153	88293
	2	Baseline	17627	86458	8.375	172819
		Unrolling	31491	127224	9.682	173847
		Pipelining	17634	86656	8.375	167694
	3	Baseline	17666	86448	8.375	241102
		Unrolling	31582	127196	9.682	241106
		Pipelining	17674	86646	8.375	233420
	4	Baseline	17864	87340	8.623	316543
		Unrolling	31843	128026	9.682	315626
		Pipelining	17872	87538	8.623	305794

TABLE I: **Description:** Security versus area versus the timing of PQC Key Generation algorithms for baseline, loop unrolling and loop pipelining optimisations.

We define baseline implementation as the HLS-based hardware implementation of a PQC algorithm. The only optimization, if at all, performed at this stage is with the area (i.e., allocation and inline) to fit in the Artix-7 board. Tables I–Table III report the hardware and timing overhead for implementing the keypair generation, signature generation and signature verification algorithms, respectively, when synthesized with only area constraints for both PQC algorithms. Flip-flops (FFs) and Look-up Tables (LUTs) are a measure of the area overhead along with BRAM (Block RAM). For qTESLA, resource utilisation and latency increase as the security strength of algorithm increases, except security level-2. For CRYSTALS-Dilithium, the area increases slightly with higher security level. Hence, the design trade-off should be checked for latency and security level.

CRYSTALS-Dilithium and qTESLA can be compared for common security category 1–3. CRYSTALS-Dilithium provides less resource utilisation and smaller clock period compared to qTESLA for all the three components. Similarly, CRYSTALS-Dilithium provides better performance than qTESLA, except security level-1 signature verification and security level-3 signature generation. CRYSTALS-Dilithium security level-1 keypair has 10 times faster implementation than the qTESLA security level-1 keypair.

B. Optimisations

Scheme	Sec. Level	Optimisation	FF	LUT	Clock (nsec)	Latency
qTESLA	1	Baseline	23143	110287	12.667	661369
		Unrolling	27004	120471	12.667	417529
		Pipelining	23150	110370	12.667	415145
	2	Baseline	39086	137559	14.179	3696400
		Unrolling	38797	137363	14.179	3696144
		Pipelining	39086	137559	14.179	3696400
	3	Baseline	25977	125921	12.667	1030252
		Unrolling	27092	128486	12.667	615532
		Pipelining	25984	126008	12.667	587461
	5	Baseline	26324	128265	12.667	5307613
		Unrolling	26829	129520	12.667	3176029
		Pipelining	26332	128359	12.667	2870347
CRYSTALS -Dilithium	1	Baseline	20912	89709	8.738	485793
		Unrolling	41295	129989	8.738	417219
		Pipelining	20980	90266	8.738	476751
	2	Baseline	21023	89933	8.738	1259801
		Unrolling	41617	130632	8.738	1158183
		Pipelining	21094	90506	8.738	1232141
	3	Baseline	21089	89991	8.738	1659851
		Unrolling	42828	132103	8.738	1565100
		Pipelining	21160	90567	8.738	1618319
	4	Baseline	21265	91098	8.738	1133399
		Unrolling	43764	134037	8.738	1006900
		Pipelining	21322	91605	8.738	1106053

TABLE II: **Description:** Security versus area versus the timing of PQC Signature Generation algorithms for baseline, loop unrolling and loop pipelining optimisations.

In this section, we will analyze Loop unrolling and Loop pipelining optimization techniques to reduce the overall latency of the PQC algorithms. First, we find out and examine the critical functions that results in high latency of the overall PQC component. Given a specific loop scheduling and pipelining optimization, a design variant is synthesized. We try to fit the loop unrolling and pipelining optimizations in Artix-7 board (by performing additional area optimizations). These loop unrolling and pipelining optimization are incorporated in the design along with baseline optimizations. With these constraints, Table I–Table III report the design variant for loop unrolling and pipelining with minimum latency. Both optimizations improve the performance. While loop unrolling reduces the latency with an increase in area, loop pipelining reduces the latency keeping the LUTs and FFs similar as the baseline implementation.

C. Design Space Exploration

Each system has a different PQC hardware design requirement based on the usage: small IoT devices require minimum area, servers require faster performance, and sensitive communication must have highest security strength. Design-space exploration helps to find the design variant based on the requirement. A server which requires faster communication, can choose the design variant with best performance from the various alternatives presented in design space exploration.

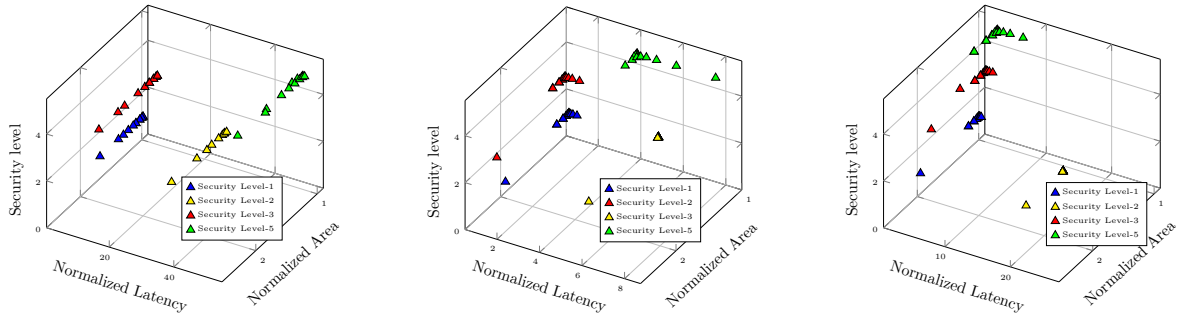
Scheme	Sec. Level	Optimisation	FF	LUT	Clock (nsec)	Latency
qTESLA	1	Baseline	17683	86095	12.667	95968
		Unrolling	25609	101108	12.667	65488
		Pipelining	17690	86160	12.667	65450
	2	Baseline	37840	145720	14.179	1921185
		Unrolling	37642	145576	14.179	1920929
		Pipelining	37840	145720	14.179	1921185
	3	Baseline	17597	84765	12.667	250425
		Unrolling	25804	100572	12.667	152889
		Pipelining	17604	84834	12.667	152027
	5	Baseline	17885	87858	12.667	728419
		Unrolling	26324	120906	12.667	356563
		Pipelining	17967	87963	12.667	353516
CRYSTALS -Dilithium	1	Baseline	15073	64930	8.738	146740
		Unrolling	43915	122802	9.83	118128
		Pipelining	15080	65147	8.738	143662
	2	Baseline	15141	65074	8.738	214832
		Unrolling	44323	123447	9.83	176480
		Pipelining	15148	65293	8.738	209707
	3	Baseline	15161	65055	8.738	292782
		Unrolling	44329	123500	9.83	242901
		Pipelining	15169	65274	8.738	285100
	4	Baseline	15179	65141	8.738	380536
		Unrolling	44474	123746	9.83	317256
		Pipelining	15187	65360	8.738	369787

TABLE III: **Description:** Security versus area versus the timing of PQC signature verification algorithms of baseline, loop unrolling and loop pipelining optimisations.

We will explore the design-space for qTESLA and CRYSTALS-Dilithium. Different loop unrolling and pipelining optimizations create different level of parallelism of the design. Enumerating all possible loop unrolling and pipelining options will generate a series of latency, area and security level vectors. Each vector is a unique point in 3-dimension design space of latency, area and security. Figure 2 shows the design space exploration points for qTESLA. For keypair generation, security level 2 and 5 have much high latency than security level 1 and 3 for all design variants. The latency saturates after initial loop unrolling optimization. For CRYSTALS-Dilithium security level-1 keypair generation, the loop unrolling optimization is reported in Table I. If the further optimization is performed, the FFs and LUTs are increased 2 times but the latency remains the same. Even if the area overhead is increased significantly, the latency does not improve. Figure 2 shows that the latency increases linearly as the security level increases except for sub-optimal security level-2. Figure 3 presents the design space exploration points for CRYSTALS-Dilithium. For all the three components, as the security level increases, the increase in latency is linear and gradual. However, the area overhead remains same for all the security level.

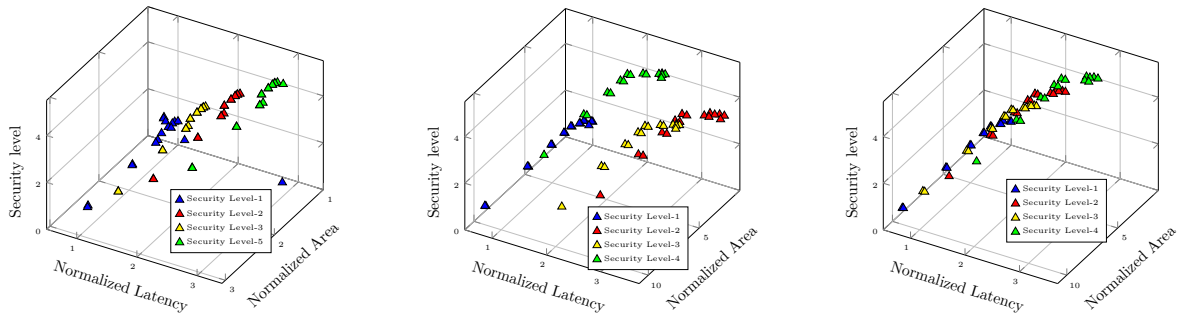
V. Key Takeaways

In this study, we have implemented 2 PQC signature schemes using a common design framework and a common target FPGA platform. In the end of this ongoing study, we expect to the design-space exploration 26 NIST PQC Compe-



(a) (b) (c)

Fig. 2: Design-space exploration of qTesla, normalised with baseline security level-1 LUT and latency for functions: (a) Key Generation, (b) Signature Generation and (c) Signature Verification.



(a) (b) (c)

Fig. 3: Design-space exploration of CRYSTALS-Dilithium, normalised with baseline security level-1 LUT and latency for functions: (a) Key Generation, (b) Signature Generation and (c) Signature Verification.

tition Round-2 algorithms. Key takeaways of this preliminary study are:

1) For qTESLA, higher the security strength, higher the latency and timing overhead except for security level-2. Section IV-A mentions that qTesla security level-2 has inefficient software implementation. Hence, the hardware implementation has high area and high latency.
2) For qTESLA, for all PQC functions, the optimization techniques follow a similar trend. Loop pipelining has better performance than both loop unrolling and baseline implementations. The area overhead for loop pipelining is less than loop unrolling, while it is almost similar to area overhead of the baseline implementation. Hence, loop pipelining is a better optimization technique for qTESLA.
3) For CRYSTALS-Dilithium, loop unrolling provides slightly better performance than loop pipelining across the security level, for all PQC functions. However, the area overhead is at least 10% higher for loop unrolling. Hence, loop pipelining is better optimization technique for both signature schemes used in this study.
4) With an increase in security strength, latency of designs increase significantly, but not the area overhead.
5) For CRYSTALS-Dilithium, the clock period increases with the optimizations. Loop unrolling changes signature verification frequency from ∼114MHz to ∼102MHz.
6) CRYSTALS-Dilithium has less area requirement compared to qTESLA for security levels 1-3, for key generation, signature and verification.

REFERENCES

[1] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM Review*, vol. 41, no. 2, pp. 303–332, 1999.

[2] L. Chen, L. Chen, S. Jordan, Y.-K. Liu, D. Moody, R. Peralta, R. Perlner, and D. Smith-Tone, *Report on post-quantum cryptography*. US Department of Commerce, National Institute of Standards and Technology, 2016.

[3] E. Homsirikamol and K. Gaj, "Can high-level synthesis compete against a hand-written code in the cryptographic domain? a case study," in *2014 International Conference on ReConFigurable Computing and FPGAs*, 2014.

[4] E. Homsirikamol and K. G. George, "Toward a new hls-based methodology for fpga benchmarking of candidates in cryptographic competitions: The caesar contest case study," in *2017 International Conference on Field Programmable Technology (ICFPT)*, pp. 120–127, IEEE, 2017.

[5] D. Micciancio, "Lattice-based cryptography," in *Encyclopedia of Cryptography and Security*, pp. 713–715, Springer, 2011.

[6] R. Cramer, L. Ducas, and B. Wesolowski, "Short stickelberger class relations and application to ideal-svp," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 324–348, Springer, 2017.

[7] K. Basu, D. Soni, M. Nabeel, and R. Karri, "Nist post-quantum cryptography- a hardware evaluation study." Cryptology ePrint Archive, Report 2019/047, 2019. https://eprint.iacr.org/2019/047.