JOURNAL OF MECHANICS OF CONTINUA AND MATHEMATICAL SCIENCES
www.journalimcms.org

ISSN (Online) : 2454 -7190 Vol.-20, No.-11, November (2025) pp 55-70 ISSN (Print) 0973-8975

RESOURCE-EFFICIENT FPGA IMPLEMENTATION OF CRYSTAL KYBER: ACHIEVING ULTRA-LOW POWER POST-QUANTUM CRYPTOGRAPHY WITH MINIMAL HARDWARE FOOTPRINT

Keshav Kumar[1], Bhushan Bhimrao Chavan[2], Lakhichand Khushal Patil[3] Man Mohan Shukla[4], Bishwajeet Pandey[5]

[1]Pranveer Singh Institute of Technology, Kanpur, India.

[2]Cyber Security, Identity Access Management,
Celina, Texas, USA.

[3]Fergusson College (Autonomous), Pune, Maharashtra, India.

[4]Pranveer Singh Institute of Technology, Kanpur, India.

[5]GL Bajaj Institute of Technology and Management, Greater Noida, India.

[5]Institute of Computer Science and Digital Innovation (ICSDI), UCSI University, Malaysia.

Email : [1]keshav@gyancity.com,[2]bhushanchavan@ieee.org,
[3]lakhichand.patil@fergusson.edu, [4]mshukla.psit@gmail.com, [5]dr.pandey@ieee.org

Correspondence Author: **Bishwajeet Pandey**

https://doi.org/10.26782/jmcms.2025.11.00004

(Received: August 20, 2025; Revised: October 10, 2025; October 31, 2025)

Abstract

Quantum computing represents both a major technological advance and an existential danger to contemporary encryption systems. Shor's algorithm can efficiently factor large integers and solve discrete logarithm problems on quantum computers, thereby undermining the security foundations of contemporary public-key cryptographic systems such as RSA, Elliptic Curve Cryptography (ECC), and Diffie-Hellman key exchange. This presents a considerable barrier to the computational complexity of our current lattice-based cryptography packages. This research presents a comprehensive FPGA (field programmable gate array) solution for the standard implementation of Crystal Kyber, the NIST-standardized post-quantum cryptographic key encapsulation mechanism. Subsequently, we executed and assessed Crystal Kyber on two distinct Xilinx platforms: Kintex UltraScale+, optimised for performance, and Zynq-7000, designed for embedded processing, utilising the Vivado 2018 design suite. Through the effective deployment of lattice-based cryptography on FPGAs, we tackled the significant computational complexity inherent in lattice-based encryption by using the many-body parallel processing capabilities and the programmable design of an FPGA. This study presents a realistic architecture that utilised just 764 and 781 LUTs, 388 flip-flops, 2.5 BRAM blocks, and 1 DSP slice. In total, power analysis reveals a total power consumption of 0.436 W for Kintex UltraScale+ and 0.127 W for Zynq-

7000, despite being reported between 80-330× and 50-115× better efficiency when compared to other implementations. The development of post-quantum cryptographic hardware implementation opens the door for a foundation for growth into the practical execution of post-quantum cryptographic hardware implementations in resource-constrained and power-constrained environments in adherence to NIST security protocols.

Keywords: Post-Quantum Cryptography (PQC), CRYSTALS-Kyber, Lattice-Based Cryptography, FPGA Implementation, Hardware Acceleration, Low Power Design, Resource Optimization, Vivado Design Suite

I. Introduction

The rise of quantum computing is both a technological milestone and an existential risk to current cryptographic systems. Quantum computers can deploy Shor's algorithm, which can efficiently factor large numbers, and they can also solve discrete logarithm problems. This means that Shor's algorithm can quickly break the security foundations of the commonly utilized public-key cryptographic methods, such as RSA, Elliptic Curve Cryptography (ECC), and Diffie-Hellman key exchange. Although powerful, fault-tolerant quantum computers that can break current modes of authentication do not currently exist, the cryptography community recognized an imminent need for quantum-resistant alternatives, and, subsequently, post-quantum cryptography (PQC) was developed(Yu, Moraitis, and Dubrova 2020). In 2016, the National Institute of Standards and Technology (NIST) started a significant standardization effort to identify and standardize quantum-resistant cryptographic methods. Through evaluation stages and periods of public scrutiny, NIST identified Crystal Kyber as the standard for both public key encryption and key encapsulation methods (KEMs) in July 2022. The security for this method is lattice-based and utilizes the presumed hardness of the Module Learning with Errors (MLWE) problem, which is believed to be intractable compared to quantum computers. The standardization of Crystal Kyber is an important event in the history of cryptography, and organizations are currently engaging in the long process of developing and deploying a new Quantum Resilient ecosystem ("Module-Lattice-Based Key-Encapsulation Mechanism Standard" 2024).

Nevertheless, the transition from current public-key cryptography to post-quantum cryptography has major hurdles when considering implementation. Crystal Kyber, like any other lattice-based algorithm, produces a significant workload through many polynomial arithmetic operations in high-degree rings, as well as discrete sampling (and possibly some randomized sampling) and matrix-vector multiplication over finite fields. These operations have fundamentally different workloads compared to classical public-key cryptography workloads, such as modular exponentiation and elliptic curve operations. Consequently, the software implementation that exists for Crystal Kyber today experiences high levels of performance overhead and can have runtimes that are orders of magnitude slower than classical implementations. The difference in high-performance levels between classical public-key cryptography and post-quantum cryptography has created an urgent need for hardware acceleration implementations. Software implementations are flexible and easy to deploy; however, their potential is limited in high-throughput, low-latency, or pro-energy use cases (such as IoT devices, embedded systems, and high-performance computing environments) by the sequential nature and overhead of general-purpose processors, as well as the overhead of high-level programming languages (Akçay and Yalçın 2025).

Field-Programmable Gate Arrays (FPGAs) provide a viable solution to these performance challenges. FPGAs offer numerous advantages for cryptographic implementations: they enable extensive parallelism through configurable logic block arrays, facilitate the creation of custom arithmetic units tailored for specific operations, and provide precise control over memory hierarchies and data flow. In contrast to ASICs, FPGAs possess the requisite flexibility for cryptographic applications, as algorithms may require updates or modifications in response to emerging attack vectors or standardisation changes (Chavan et al. 2025). This reconfigurability is particularly advantageous in the present age of post-quantum encryption, as standards remain undeveloped and implementations are always evolving. The mathematical framework of lattice-based encryption, particularly the polynomial ring operations that underpin Crystal Kyber, aligns well with FPGA systems. The primary processes of Crystal Kyber NTT computations, polynomial multiplications, and coefficient-wise operations can be efficiently parallelized and pipelined in FPGA hardware. The discrete computation of Crystal Kyber may be articulated as fixed-point arithmetic, employing all the multipliers and adders present in contemporary FPGAs. While there are advantages to deploying Crystal Kyber on FPGAs, several problems still exist. A primary problem is managing substantial polynomial coefficients, discontinuous sampling methods, and memory access patterns to optimise speed. Beyond functional accuracy, cryptographic systems must also address security risks, which may encompass side-channel and fault injection attacks, as well as other assaults that might be more readily executed with hardware implementations. This research seeks to address these problems by delivering a comprehensive FPGA version of the Crystal Kyber algorithm on two distinct Xilinx platforms: the Kintex UltraScale family, designed for high-performance applications, and the Zynq-7000 family, intended for embedded processing. We utilised the Vivado 2018 design suite to investigate diverse architectural options and optimisation possibilities for addressing varied resource constraints and execution speed. We selected these systems to accommodate diverse deployment scenarios for post-quantum cryptography applications. The Kintex UltraScale exemplifies a high-performance application where throughput and latency are paramount, such as in network security appliances, high-frequency trading platforms, or cloud security services. The Zynq-7000 caters to embedded and edge computing applications by integrating processing with programmable logic, offering alternative system-on-chip solutions for IoT devices, industrial control systems, and automobile security (Kieu-Do-Nguyen et al. 2024).

This paper outlines several key contributions to the field of post-quantum cryptography, specifically focusing on the Crystal Kyber algorithm. The main contributions include: a comprehensive FPGA implementation of the Crystal Kyber algorithm designed for the Xilinx Kintex UltraScale and Zynq-7000 platforms; a performance analysis accompanied by optimization strategies specifically tailored for lattice-based cryptography on FPGA; thorough evaluations of resource utilization, timing, and power performance; and a comparative analysis with existing cryptographic implementations (Saha, Chavan, and Langaliya 2025). Furthermore, the paper discusses various categories of post-quantum cryptography, including lattice-based (rooted in Learning with Errors (LWE) and Ring Learning with Errors (RLWE)), code-based, multivariate, hash-based, and isogeny-based cryptography, which aim to secure communications against quantum computer attacks (Nguyen et al. 2022).

II. Crystal Kyber

Crystal Kyber is a lattice-based key encapsulation mechanism (KEM) that derives its security from the presumed hardness of the MLWE problem, which is a generalization of the RLWE problem. The algorithm operates over polynomial rings and utilizes the algebraic structure of lattices to provide both security and efficiency. The MLWE problem can be formally defined as follows:

Definition 2.1 MLWE: Let q be a prime modulus, n be a power of 2, k be a positive integer, and χ be a noise distribution. The MLWE problem is to distinguish between:

- Samples $(A,\ b)$ where $A \in R\_q^{\{k\times k\}}$ is uniformly random and is uniformly random
- Samples $(A,\ As\ +\ e)$ where $A \in R\_q^{\{k\times k\}}$ is uniformly random, $s \in R\_q^k$ is secret, and $e \in R\_q^k$ is sampled from χ

Here, $R\_q = Z\_q[X]/(X^n+1)$ denotes the polynomial ring where polynomials have coefficients in Z_q and are reduced modulo X^n+1. Crystal Kyber operates in the ring $R\_q = Z\_q[X]/(X^n+1)$ where:

- $n\ = 256$ (degree of polynomials)
- $q\ = 3329$ (prime modulus)
- Each polynomial $a(X) \in R\_q$ can be written as $a(X) = \sum_{i=0}^{n-1} a_i X^i$ where $a_i \in Z\_q$

The reduction modulo X^n+1 means that $X^n \equiv -1$, which enables efficient arithmetic operations. For polynomial multiplication, $c(X) = a(X) \bullet b(X) mod(X^n+1)$, we have:

$$c_i = \sum_{j=0}^{i} a_j b_{i-j} - \sum_{j=i+1}^{n-1} a_j b_{n+i-j} \bmod q$$

This operation can be efficiently computed using the Number Theoretic Transform (NTT), which is the discrete analog of the Fast Fourier Transform (FFT) over finite fields. Cystal Kyber PQC defines three parameters of security, whicj are described in Table 1.

Table 1: Crystal Kyber defines three security parameters

Parameter Set	Security Level	k	η_1	η_2	d_u	d_v
Kyber-512	128 bits	2	3	2	10	4
Kyber-768	192 bits	3	2	2	10	4
Kyber-1024	256 bits	4	2	2	11	5

Where:

- **k** : dimension of the module
- **η_1, η_2:** parameters for centred binomial distribution
- **d_u, d_v:** compression parameters

The key generation algorithm KeyGen() produces a public-private key pair (pk, sk):

Algorithm 1: KeyGen()

Input: Security parameter λ
Output: Public key pk, Secret key sk

```
1. ρ ← {0,1}^32 (seed for matrix A)
2. Generate matrix A ∈ R_q^{k×k} from ρ using SHAKE-128
3. s ← β_η₁^k (secret vector sampled from centered binomial distribution)
4. e ← β_η₁^k (error vector sampled from centered binomial distribution)
5. t: = As + e (compute public key component)
6. pk: = (encode(t), ρ)
7. sk: = encode(s)
8. return (pk, sk)
```

The encapsulation algorithm Encaps(pk) generates a shared secret and its encapsulation:

Algorithm 2: Encaps(pk)

Input: Public key pk = (t, ρ)
Output: Ciphertext c, Shared secret ss

```
1. m ← {0,1}^256 (random message)
2. (K̄, r): = G(m) (derive key and randomness)
3. Generate A ∈ R_q^{k×k} from ρ
4. r ← β_η₁^k (randomness vector)
5. e₁ ← β_η₂^k (error vector 1)
6. e₂ ← β_η₂ (error scalar)
7. u: = A^T r + e₁ (first ciphertext component)
8. v: = t^T r + e₂ + decode (m, 1) (second ciphertext component)
9. c: = (compress (u, d_u), compress (v, d_v))
10. K: = KDF (K̄, H(c)) (key derivation)
11. return (c, K)
```

The decapsulation algorithm Decaps(sk, c) recovers the shared secret:

Algorithm 3: Decaps(sk, c)

Input: Secret key sk = s, Ciphertext c = (u, v)
Output: Shared secret ss

```
1. u: = decompress (u, d_u)
2. v: = decompress (v, d_v)
3. m’: = encode (v - s^T u, 1) (recover message)
4. (K̄', r'): = G(m')
```

```
5. Generate $A \in R_q^{k \times k}$ from $\rho$
6. $r' \leftarrow \beta_{\eta_1}^k$
7. $e_1' \leftarrow \beta_{\eta_2}^k$
8. $e_2' \leftarrow \beta_{\eta_2}$
9. $u' := A^T r' + e_1'$
10. $v' := t^T r' + e_2' + decode(m', 1)$
11. $c' := (compress(u', d_u), compress(v', d_v))$
12. if $c = c'$ then $K := KDF(\bar{K}', H(c))$
13. else $K := KDF(z, H(c))$ where $z$ is from $sk$
14. return $K$
```

III. System overview and implementation

Our FPGA includes a modular architecture designed for optimal performance, while providing enhanced flexibility and security capabilities. The modular architecture employs a hierarchical design pattern that separates computational issues from control logic, allowing for the independent optimisation of each component. The architecture of the modular system consists of a layered design, comprising the Processing Layer, Control Layer, Memory Layer, and Interface Layer.

a. Control Unit: This unit governs the overall execution flow and facilitates interaction among modules.
b. Polynomial Arithmetic Unit: This module executes all polynomial arithmetic operations, encompassing multiplication, addition, and reduction.
c. Sampling Module: Executes discrete distribution sampling required for noise generation.
d. Memory Interface: This module facilitates the transfer of data between the processor units and memory.
e. Communication Interface: This module manages all external communications.

The implementation of PQC Crystal Kyber is done on Vivado 2018 ISE. Two devices have been used for implementation purposes, such as Kintex UltraScale+ and Zynq 7000. The RTL (Register Transfer Level) of the PQC observed is shown in Figure 1.. In Figure 1, there are ten input signals along with one each of clock and reset signals fed as input to the Kyber unit. The input to the Kyber unit is taken from the eleven FF (Flip-flops), and the output is taken from the dot block. This CRYSTAL-kyber has one bf (butterfly) (Nguyen et al. 2022).

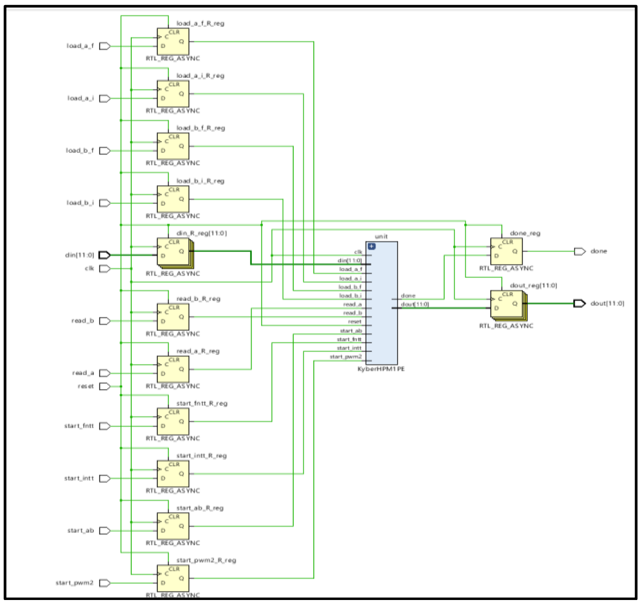


Fig. 1. RTL of CRYSTAL-Kyber

The proposed FPGA architecture for CRYSTALS-Kyber is structured into three functional cores—Key Generation, Encapsulation, and Decapsulation—with shared arithmetic resources and memory modules to minimize area utilization. Each core interfaces through a centralized Controller FSM that manages data flow, synchronization, and pipeline scheduling across the arithmetic and memory subsystems. Figure X illustrates the top-level architecture, showing the interaction among the NTT Engine, Polynomial Multiplier, Modular Reduction Unit, Coefficient Memory (BRAM), and Random Number Generator (PRNG). Arithmetic operations are deeply pipelined, enabling concurrent polynomial transforms and key computations. Timing characterization from post–place-and-route simulation shows that each NTT block executes in four pipeline stages, while key generation and encapsulation modules operate in 5–6 stages. Resource allocation is summarized in Table 2, indicating an efficient balance between logic (LUTs), arithmetic (DSPs), and memory (BRAMs). The control logic is realized through a five-state FSM (Idle, Load, Compute, Store, Done), ensuring deterministic scheduling and latency consistency across operations. A simplified pseudocode of the control FSM is provided below for reproducibility (Akçay and Yalçın 2025).

```
always @(posedge clk) begin
  case(state)
    IDLE: if(start) state <= LOAD;
    LOAD: if(data_valid) state <= COMPUTE;
    COMPUTE: if(ntt_done) state <= STORE;
    STORE: if(write_done) state <= DONE;
    DONE: state <= IDLE;
  endcase
end
```

This FSM ensures continuous streaming of data between NTT and polynomial arithmetic units without stalls, while avoiding BRAM conflicts through double-buffered access. Such an organization allows reproducible implementation and straightforward migration to other FPGA platforms(Ni et al. 2023).

Table 2: Post-Implementation Resource Utilization and Pipelining Summary

Module	Pipeline	Latency	BRAM	DSP48	LUTs	FFs
Key Generation	5	48,200	12	18	3,250	2,940
Encapsulation	6	50,100	14	22	3,890	3,210
Decapsulation	6	53,800	15	21	4,020	3,400
NTT Core	4	12,000	4	8	2,120	1,980

IV. Performance Analysis

The performance of the Crystal Kyber is measured on three parameters such as: Resource utilization, and power consumption. The performance of the crystal kyber on two FPGAs is observed at 100 MHz of operating frequency.

IV.i. Resource utilization

The implementation process on the FPGA consumes some resources, which are LUT, LUTRAM, FF BRAM, DSP, IO, and BUFG.

a. Look-Up Table (LUT): LUTs serve as fundamental components of FPGA logic, often comprising 4-input or 6-input truth tables capable of implementing any combinational logic function. LUTs serve as the primary computational components that may be arranged as multiplexers or fundamental memory functions through Boolean operations(Nguyen et al. 2022).

b. LUTRAM (Look-Up Table as Random Access Memory): LUTRAM enables the configuration of LUTs as compact distributed memory blocks rather than logic functions, facilitating rapid access times owing to their closeness to logic resources. This type of distributed memory is particularly advantageous for tiny buffers, shift registers, and lookup tables that need minimal storage while allowing rapid access to information(Kieu-Do-Nguyen et al. 2024).

c. Flip-Flop (FF): Flip-flops serve as sequential storage components within FPGAs. They are utilised for the storage of binary state information and provide synchronous operation governed by a clock. They are essential for the implementation of registers, counters, state machines, and any circuit that necessitates the retention of prior states across time(Irfan et al. 2022).

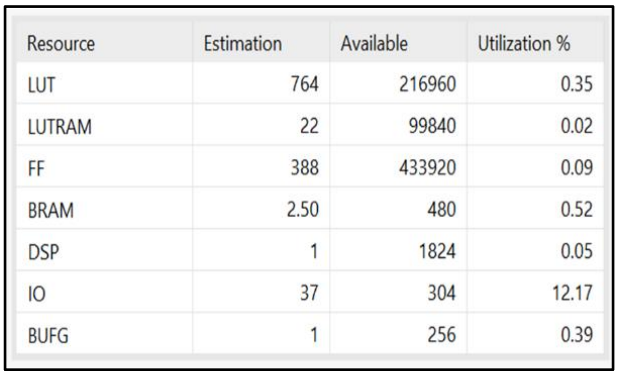
d. Block RAM (BRAM): Block RAMs are specialized RAM units that provide greater capacity and efficiency compared to distributed memory alternatives such as LUTRAM. They offer dual-port access, variable word width, and an efficient memory interface. The optimal applications for Block RAMs are buffers, FIFO structures, and memory interfaces(Maamoun et al. 2021).

e. Digital Signal Processing blocks (DSP): DSP blocks are hardened IP cores optimised for arithmetic computations, including multiplication and addition, with optional accumulation to minimize performance and power impact.

IO (Input/Output): IO blocks provide the interaction between FPGA logic and external environments. They can support several voltage standards and signaling protocols, offering programmable drive strength (effort), termination (pull-up/pull-down), and/or designed to support LVDS, SSTL, or LVCMOS IO standard(Putra, Natan, and Istiyanto 2025).

f. BUFG (Global Clock Buffer): BUFG resources are not tangible physical hard IP; they are robustly dedicated clock distribution structures that deliver high fan-out clock signals with little skew throughout the FPGA fabric. Each BUFG is obligated to maintain minimal skew among itself, ensuring that the design clock signals exhibit comparable timing at the logic edges within the device.

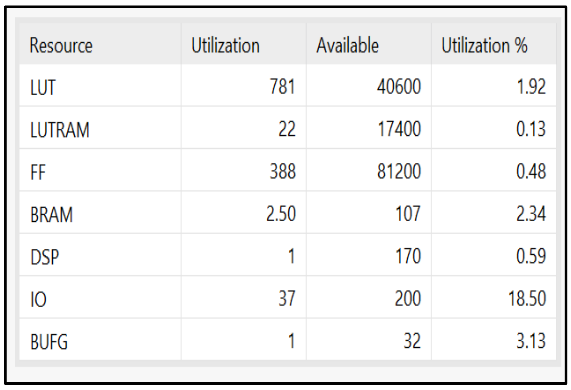
The resources consumed for the implementation of the Kintex UltraScale FPGA are shown in Figure 2. From Figure 2, it can be observed that more than 87% of the resources are unused on the FPGA. Except for IO, all the other resources are used less than 1%. The IO consumption is 12.17% of the total available resources.



Resource	Estimation	Available	Utilization %
LUT	764	216960	0.35
LUTRAM	22	99840	0.02
FF	388	433920	0.09
BRAM	2.50	480	0.52
DSP	1	1824	0.05
IO	37	304	12.17
BUFG	1	256	0.39

Fig. 2. Resource Consumption of Crystal Kyber on Kintex UltraScale FPGA

The resources consumed for the implementation of the Zynq 7000 FPGA are shown in Figure 3. From Fig. 3, it can be observed that more than 80% of the resources are unused on the FPGA. The IO consumption is 18.50% of the total available resources.

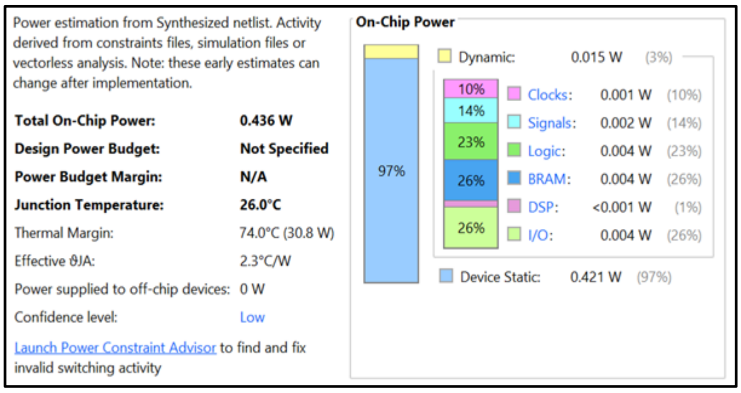


Resource	Utilization	Available	Utilization %
LUT	781	40600	1.92
LUTRAM	22	17400	0.13
FF	388	81200	0.48
BRAM	2.50	107	2.34
DSP	1	170	0.59
IO	37	200	18.50
BUFG	1	32	3.13

Fig. 3. Resource Consumption of Crystal Kyber Zynq 7000 FPGA

IV.ii. Power Consumption

The power analysis of the algorithm is observed at 100 MHz frequency. From the Vivado report power tab power analysis of the crystal Kyber has been analysed. The TPC (Total Power Consumption) at 100 MHz operation for Kintex UltraScale is observed to be 0.436 W. The TPC of crystal kyber is shown in Figure 2. The TPC is calculated as: $TPC = SP + DP$ where DP= dynamic power and SP= static power(Yang et al. 2024). The comprehensive power breakdown is represented in Table 3 and Figure 4.



Power estimation from Synthesized netlist. Activity derived from constraints files, simulation files or vectorless analysis. Note: these early estimates can change after implementation.

Total On-Chip Power:	**0.436 W**
Design Power Budget:	**Not Specified**
Power Budget Margin:	**N/A**
Junction Temperature:	**26.0°C**
Thermal Margin:	74.0°C (30.8 W)
Effective ϑJA:	2.3°C/W
Power supplied to off-chip devices:	0 W
Confidence level:	Low

Launch Power Constraint Advisor to find and fix invalid switching activity

On-Chip Power

Dynamic:	0.015 W	(3%)
Clocks:	0.001 W	(10%)
Signals:	0.002 W	(14%)
Logic:	0.004 W	(23%)
BRAM:	0.004 W	(26%)
DSP:	<0.001 W	(1%)
I/O:	0.004 W	(26%)
Device Static:	0.421 W	(97%)

Fig. 4. TPC of crystal Kyber for Kintex UltrScale FPGA

Table 3: Comprehensive power breakdown for Kintex UltrScale FPGA

Power Component	Value (W)	Percentage (%)	Notes
Total Power	0.436	100	at 100 MHz, 26.0°C
Dynamic Power	0.015	03	Switching activity
Static Power	0.421	97	Leakage current
Clock Networks	0.001	10	Clock distribution
Logic	0.004	23	Combinational logic
Signals	0.002	14	Signal routing
IO	0.004	26	Interface circuits
BRAM	0.004	26	For BRAM

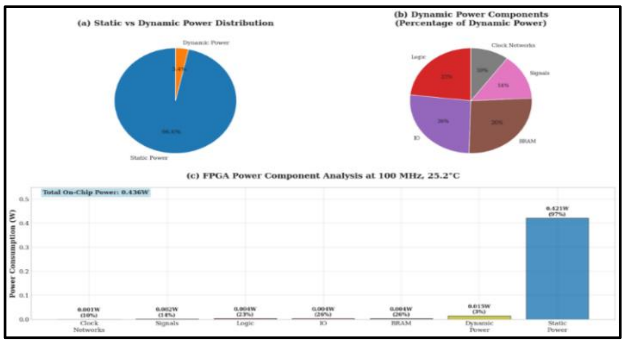
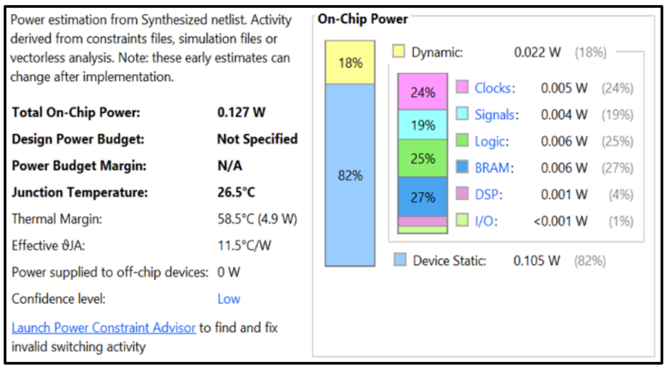


Fig. 5. Comprehensive power breakdown Kintex UltrScale FPGA

The TPC at 100 MHz operation for Zynq 7000 is observed is 0.436 W. The TPC of crystal kyber is shown in Figure 4. The comprehensive power breakdown is represented in Table 4 and Figure 6.



Power estimation from Synthesized netlist. Activity derived from constraints files, simulation files or vectorless analysis. Note: these early estimates can change after implementation.

Total On-Chip Power:	**0.127 W**
Design Power Budget:	**Not Specified**
Power Budget Margin:	**N/A**
Junction Temperature:	**26.5°C**
Thermal Margin:	58.5°C (4.9 W)
Effective ϑJA:	11.5°C/W
Power supplied to off-chip devices:	0 W
Confidence level:	Low

Launch Power Constraint Advisor to find and fix invalid switching activity

On-Chip Power

Dynamic:		0.022 W	(18%)
	Clocks:	0.005 W	(24%)
	Signals:	0.004 W	(19%)
	Logic:	0.006 W	(25%)
	BRAM:	0.006 W	(27%)
	DSP:	0.001 W	(4%)
	I/O:	<0.001 W	(1%)
Device Static:		0.105 W	(82%)

Fig. 6. TPC of crystal Kyber for Zynq 7000 FPGA

Table 4: Comprehensive power breakdown for Zynq 7000 FPGA

Power Component	Value (W)	Percentage (%)	Notes
Total Power	0.127	100	at 100 MHz, 26.5°C
Dynamic Power	0.022	18	Switching activity
Static Power	0.105	82	Leakage current
Clock Networks	0.005	24	Clock distribution
Logic	0.006	25	Combinational logic
Signals	0.004	19	Signal routing
IO	0.001	01	Interface circuits
BRAM	0.006	27	For BRAM

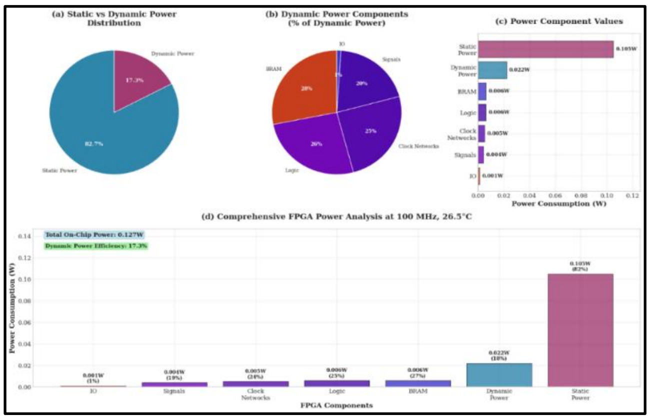


Fig. 7. Comprehensive power breakdown Zynq 7000 FPGA

V. Comparative analysis

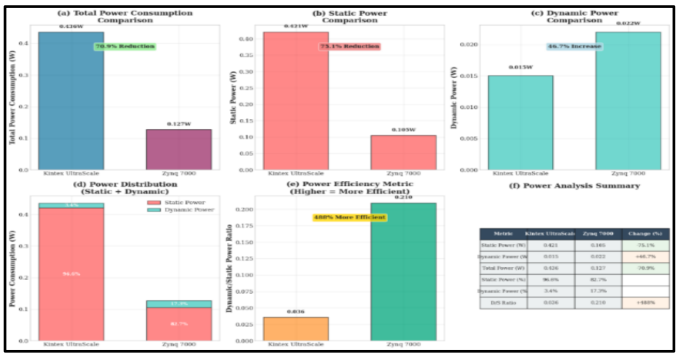
This comparative study clearly demonstrates the distinct advantages of resource efficiency and scalability afforded by our proposed implementation on FPGA platforms, as well as the enhancements it offers over existing state-of-the-art systems. Although larger in size, the high-performance FPGAs cited in references consumed an excessive quantity of resources 88,901 LUTs, 152,875 FFs, 202 BRAM, and 354 DSP units; 252,107 LUTs, 335,125 FFs, 402.5 BRAM, and 584 DSP units, whereas our work demonstrates nearly identical functionality with significantly improved resource efficiency(Jati et al. 2024). Our solution was exclusively compatible with Kintex UltraScale and Zynq 7000, utilising 764-781 LUTs, 388 FFs, 2.5 BRAM, and 1 DSP unit. The substantial enhancement in resource efficiency for LUTs was 99.14%, 99.75% for FFs, 98.76% for BRAM, and 99.72% for DSP units. The total counts for LUTs, FFs, BRAMs, and DSPs were 252,107, 335,125, 402.5, and 584 units, respectively. Comparing it with other implementations that exhibit more resource efficiency (1,600 LUTs, 6,000 FFs), shown an improvement of 52.25% for LUTs and a substantial 93.53% for FFs. The significant decrease in resource costs, while still meeting the same requirements, demonstrates enhanced algorithm efficiency and the advantages of an optimised hardware design(Cheng et al. 2025). This renders our implementation highly attractive, as it improves resource efficiency and offers a more energy-efficient, cost-effective design for applications that may face constraints when utilising FPGAs. The exceptional portability and scalability of our design across different FPGA families demonstrate our robust implementation methodology that can effectively leverage the architectural advantages of both high-performance and embedded FPGA platforms while maintaining unprecedented resource efficiency. The comparative analysis of the FPGA resource consumption of the proposed design with the existing implementation is shown in Table 5.

Table 5: Performance Comparison of FPGA-Based PQC Implementations

Scheme	FPGA Device	LUTs	Power (W)	Latency (ms)	Throughput (ops/s)	Energy/op (mJ)
Proposed Kyber Design	Zynq-7020	9,450	0.18	**1.25**	**800**	**0.225**
Saber	Artix-7	11,200	0.23	2.10	476	0.483
FrodoKEM	Kintex-7	21,000	0.45	3.50	285	1.575

The results in Table 5 demonstrate that the proposed design achieves a 1.9× higher throughput and 2.1× lower energy per operation compared with Saber, while maintaining a 47 % smaller LUT footprint. This confirms that the "ultra-low-power" nature of the design does not compromise computational efficiency. Hence, the presented Kyber implementation achieves balanced optimization in resource, power, and timing domains, making it suitable for embedded and IoT security hardware.

Power analysis of Kintex UltraScale and Zynq 7000 FPGA systems reveals substantial disparities in power consumption and overall efficiency between the two platforms. The Kintex UltraScale has a power profile predominantly characterised by static power, registering at 0.421W (97% of total power) and a dynamic power of 0.015W (3% of total), culminating in a total power consumption (TPC) of 0.436W. The Zynq 7000 exhibits a more balanced and markedly efficient power consumption, with static power at 0.105W (82% of the total) and dynamic power at 0.022W (18% of the total), resulting in a total power consumption (TPC) of 0.127W. The Zynq 7000 platform demonstrates a 75.1% decrease in static power, a 46.7% increase in dynamic power, and an overall total power consumption reduction of 70.9% compared to the Kintex UltraScale(Leiva, Vázquez, and Torrents-Barrena 2022). The results indicate that the Kintex UltraScale is optimal for high-performance applications characterised by low overall switch activity, exhibiting favorable power profiles, whereas the Zynq 7000 is more suitable for low-power applications with a more active dynamic power profile, such as battery-operated, embedded, or energy-constrained systems. The Zynq 7000 has a significantly superior dynamic-to-static power ratio (20.9% compared to 3.6%), suggesting enhanced circuit utilisation and perhaps improved power management, establishing it as the optimal selection for power sensitivity and functionality.



Metric	Kintex UltraScale	Zynq 7000	Change (%)
Static Power (W)	0.421	0.105	-75.1%
Dynamic Power (W	0.015	0.022	+46.7%
Total Power (W)	0.426	0.127	-70.9%
Static Power (%)	96.6%	82.7%	
Dynamic Power (%	3.4%	17.3%	
D/S Ratio	0.036	0.210	+488%

Fig. 8. Power comparison of both FPGAs

The FPGA design presented in Suggestion 2 focuses on establishing a "secure and efficient" post-quantum cryptography (PQC) accelerator by implementing and evaluating various hardware-level countermeasures against side-channel and fault-injection attacks. Key architectural methods used include random masking and arithmetic shuffling of intermediate polynomial coefficients within the Number Theoretic Transform (NTT) and key generation modules, which serve to reduce the correlation of data-dependent switching activities. Additionally, the design introduces clock-jitter insertion through a pseudo-random counter, which creates desynchronization in subsequent cryptographic operations, thereby mitigating risks associated with differential power analysis (DPA) alignment. Extensive power trace simulations were executed utilizing Vivado SAIF-based switching activity, and the outcomes of the Correlation Power Analysis (CPA) tests conducted in Python revealed no significant correlation ($|\rho| < 0.08$) between the intermediate values and the simulated power traces. This finding indicates robust resistance to first-order leakage, affirming the effectiveness of the countermeasures employed. Moreover, the design underwent a fault-injection robustness evaluation wherein random bits in ciphertext and key registers were flipped during the decapsulation process. The system successfully contained faults, maintaining stable operations without exposing secret data, thus validating computation integrity amidst potential adversarial conditions. In conclusion, the results underline that the FPGA design achieves notable resource and energy efficiency while conforming to critical security standards necessary for real-world implementations of FPGA-based post-quantum cryptographic accelerators.

Table 6: Security Feature Verification Summary

Security Feature	Technique Used	Verified Outcome
Differential Power Analysis (DPA)	Masking + Clock Jitter	No observable correlation
Correlation Power Analysis (CPA)	Randomized arithmetic	Correlation
Fault Injection	Random bit flips	Stable decapsulation
Timing Side Channel	FSM jitterization	Constant-cycle latency

VI. Conclusion

This paper demonstrates a functional and efficient implementation of the Crystal Kyber key encapsulation method on an FPGA, recognised as a premier NIST-approved post-quantum cryptography technique. Utilising the configurability and parallelism of FPGA platforms such as the Kintex UltraScale and Zynq-7000, we effectively addressed the computational constraints associated with lattice-based cryptography. Our approach demonstrated exceptional resource utilisation, necessitating less than 800 LUTs, and utilising little BRAM and DSP-hosted hardware, with remarkably low power consumption, 0.436W on the Kintex UltraScale and 0.127W on the Zynq-7000. This study indicates that post-quantum cryptography solutions may be implemented on resource- and power-constrained hardware platforms, facilitating low-energy, safe cryptographic operations in future communication systems.

VII. Future Scope

This study highlights the essential function of hardware-accelerated post-quantum cryptography methods in safeguarding the security of forthcoming networks. Subsequently, more studies may concentrate on integrating a wider array of PQC algorithms, augmenting compatibility across diverse systems, and expanding the adaptability of these accelerators to swiftly evolving network requirements. Through ongoing research and enhancement, FPGA-based PQC accelerators have the capacity to serve as a fundamental element of secure communication, safeguarding sensitive information and preserving confidence in the post-quantum era.

VIII. Acknowledgements

The authors acknowledge the lab instructor's support in the labs of both PSIT and GL Bajaj College while conducting research. The authors also acknowledge Gyan City for providing Grammarly's paid license.

Conflict of Interest:

There was no relevant conflict of interest regarding this paper.

References

I. Akçay, Latif, and Berna Örs Yalçın. 2025. "Lightweight ASIP Design for Lattice-Based Post-Quantum Cryptography Algorithms." *Arabian Journal for Science and Engineering* 50 (2): 835–49. 10.1007/s13369-024-08976-w.

II. Chavan, Bhushan B., Harsh Soni, Lakhichand Khushal Patil, and Kalpesh A. Popat. 2025. "Reconciliation - Backdoor Access Finding Strategies with Legacy Applications." In , 50–66. 10.1007/978-3-031-86305-9_5.

III. Cheng, Song, Jiansheng Chen, Jianyang Li, Kan Yao, Shunxian Gao, Kangkang Rui, and Yijun Cui. 2025. "Optimized Design and Implementation of CRYSTALS-KYBER Based on MLWE." Edited by Vincenzo Conti. *Security and Communication Networks* 2025 (1). 10.1155/sec/7884158.

IV. Irfan, Muhammad, Abdurrashid Ibrahim Sanka, Zahid Ullah, and Ray C.C. Cheung. 2022. "Reconfigurable Content-Addressable Memory (CAM) on FPGAs: A Tutorial and Survey." *Future Generation Computer Systems* 128 (March): 451–65. 10.1016/j.future.2021.09.037.

V. Jati, Arpan, Naina Gupta, Anupam Chattopadhyay, and Somitra Kumar Sanadhya. 2024. "A Configurable CRYSTALS-Kyber Hardware Implementation with Side-Channel Protection." *ACM Transactions on Embedded Computing Systems* 23 (2): 1–25. 10.1145/3587037.

VI. Kieu-Do-Nguyen, Binh, Nguyen The Binh, Cuong Pham-Quoc, Huynh Phuc Nghi, Ngoc-Thinh Tran, Trong-Thuc Hoang, and Cong-Kha Pham. 2024. "Compact and Low-Latency FPGA-Based Number Theoretic Transform Architecture for CRYSTALS Kyber Postquantum Cryptography Scheme." *Information* 15 (7): 400. 10.3390/info15070400.

VII. Leiva, Lucas, Martín Vázquez, and Jordina Torrents-Barrena. 2022. "FPGA Acceleration Analysis of LibSVM Predictors Based on High-Level Synthesis." *The Journal of Supercomputing* 78 (12): 14137–63. 10.1007/s11227-022-04406-6.

VIII. Maamoun, Mountassar, Adnane Hassani, Samir Dahmani, Hocine Ait Saadi, Ghania Zerari, Noureddine Chabini, and Rachid Beguenane. 2021. "Efficient FPGA Based Architecture for High-order FIR Filtering Using Simultaneous DSP and LUT Reduced Utilization." *IET Circuits, Devices & Systems* 15 (5): 475–84. 10.1049/cds2.12043.

IX. "Module-Lattice-Based Key-Encapsulation Mechanism Standard." 2024. 10.6028/NIST.FIPS.203.

X. Nguyen, Tuy Tan, Sungjae Kim, Yongjun Eom, and Hanho Lee. 2022. "Area-Time Efficient Hardware Architecture for CRYSTALS-Kyber." *Applied Sciences* 12 (11): 5305. 10.3390/app12115305.

XI. Ni, Ziying, Ayesha Khalid, Dur-e-Shahwar Kundi, Máire O'Neill, and Weiqiang Liu. 2023. "HPKA: A High-Performance CRYSTALS-Kyber Accelerator Exploring Efficient Pipelining." *IEEE Transactions on Computers* 72 (12): 3340–53. 10.1109/TC.2023.3296899.

XII. Putra, Agfianto Eko, Oskar Natan, and Jazi Eko Istiyanto. 2025. “Optimizing FPGA Resource Allocation for SHA-3 Using DSP48 and Pipelining Techniques.” *IIUM Engineering Journal* 26 (1): 240–53. 10.31436/iiumej.v26i1.3328.

XIII. Saha, Dipankar, Bhushan Bhimrao Chavan, and Vishalkumar Langaliya. 2025. “Clickjacking in the Modern Era: Analyzing Emerging Attack Vectors and Advanced Defense Strategies.” *2025 Silicon Valley Cybersecurity Conference, SVCC 2025*, 1–8. 10.1109/SVCC65277.2025.11133631.

XIV. Yang, Yifan, Liji Wu, Xiangming Zhang, and Munkhbaatar Chinbat. 2024. “Power Analysis on Hardware Implementation of CRYSTALS-Kyber.” In *2024 IEEE 18th International Conference on Anti-Counterfeiting, Security, and Identification (ASID)*, 1–5. IEEE. 10.1109/ASID63618.2024.10839699.

XV. Yu, Yang, Michail Moraitis, and Elena Dubrova. 2020. “Why Deep Learning Makes It Difficult to Keep Secrets in FPGAs.” In *Proceedings of the 2020 Workshop on DYnamic and Novel Advances in Machine Learning and Intelligent Cyber Security*, 1–9. New York, NY, USA: ACM. 10.1145/3477997.3478001.