International Journal of Engineering Applied Science and Management
ISSN (Online): 2582-6948
Vol. 7 Issue 6, June 2026

Area-Efficient Karatsuba Accelerator Using AEKA FPGA for Ring-Binary-LWE-Based Lightweight PQC

[1]Sravani Yerukala, [2]Mahaboob Basha S, [3] Chakrapani T, [4]Ahmed Basha S, [5]Suvarna K

[1,2,3,4,5]Department of Electronics and Communication Engineering, St. Johns College of Engineering and Technology, Yemmiganur, Kurnool, Andhra Pradesh, 518360, India.

Corresponding author: [4]ahmedbasha.syed@gmail.com

ABSTRACT

The Verilog-based FPGA accelerator AEKA, which implements an area-efficient Karatsuba polynomial multiplier for Ring-Binary-LWE post-quantum cryptography (PQC), is shown in this project. By reducing the amount of multiplications needed for polynomial operations, the design minimizes the use of DSP slices, LUTs, and registers on FPGA platforms by utilizing the divide-and-conquer Karatsuba algorithm. AEKA achieves high throughput and low area-delay product by combining time-multiplexing and pipelining, making lightweight PQC possible on resource-constrained devices like embedded systems and the Internet of Things. Xilinx Vivado is used to fully implement and verify the accelerator at the circuit level. When compared to traditional polynomial multipliers, simulation and synthesis results show notable gains in resource consumption and operational efficiency. For FPGA-base post-quantum cryptography, AEKA offers a workable, fast, and small solution that permits safe, lightweight cryptographic operations without sacrificing area or performance limitations.

Keywords: FPGA-based post-quantum cryptography, DSP slice, LUT, AEKA, Xilinx Vivado, Karatsuba polynomial multiplier, Ring-Binary-LWE post-quantum cryptography

1. INTRODUCTION:

Classical cryptography systems, especially those relying on integer factorization and discrete logarithm issues, are seriously threatened by the quick development of quantum computing. The foundation of contemporary secure communication, algorithms like RSA and ECC, are susceptible to quantum assaults. Consequently, post-quantum cryptography (PQC) has become an important field of study aimed at creating cryptographic methods resistant to quantum attackers. Lattice-based cryptography, particularly Ring Learning With Errors (Ring-LWE) and its binary counterpart (Ring-Binary- LWE), has been popular among PQC candidates because of its robust security underpinnings and computational effectiveness. In lattice-based cryptography methods, polynomial multiplication is a basic operation that has a big impact on system performance.

However, when implemented in hardware, standard multiplication approaches frequently need considerable computational resources, increasing area, latency, and power consumption. In contexts with limited resources, such embedded systems and IoT devices, where effective hardware utilization is crucial, this problem becomes more pressing. Therefore, a crucial first step in making PQC feasible is to optimize polynomial multiplication. In order to overcome these difficulties, this research presents AEKA, an FPGA-based accelerator for effective polynomial multiplication in Ring-Binary-LWE systems that was created using Verilog. The Karatsuba algorithm, a divide-and-conquer strategy that uses less multiplication operations than conventional techniques, is used in the suggested architecture. The design efficiently lowers the use of FPGA resources including DSP slices, lookup tables (LUTs), and registers by reducing computational complexity, improving space efficiency. Additionally, AEKA uses sophisticated architectural strategies like pipelining and time-multiplexing to strike a balance between hardware use and performance. Xilinx Vivado is used to implement and validate the design, which shows reduced area-delay product and increased throughput when compared to other systems. All things considered, AEKA offers a scalable and effective hardware platform for post-quantum cryptographic operations, making it appropriate for secure communication systems of the future.

2. PROJECT OBJECTIVE :

Designing and implementing AEKA, an area-efficient FPGA-based accelerator for polynomial multiplication in Ring-Binary-LWE post-quantum cryptography systems, is the main goal of this project. The main goal is to use the Karatsuba method to minimize computing complexity and maximize the use of hardware resources like registers, lookup tables (LUTs), and DSP slices. The suggested design seeks to establish an effective balance between performance and area by reducing the amount of multiplication operations, making it appropriate for real-time cryptographic applications. Improving system performance via architectural strategies like pipelining and time-multiplexing is another important goal. These techniques are used to increase throughput while preserving low latency and a lower area-delay product. In order to ensure dependable operation, scalability, and appropriateness for deployment in resource-constrained areas like IoT and embedded systems, the project also attempts to evaluate the design using FPGA tools like Xilinx Vivado.

3. PROBLEM STATEMENT:

Traditional cryptographic techniques are now insecure due to the growing threat of quantum computing, necessitating the development of safe post-quantum replacements. When implemented using traditional techniques, lattice-based cryptographic algorithms like Ring-Binary-LWE require intense polynomial multiplication operations, which are costly and resource-consuming. Particularly on FPGA platforms, this leads to excessive hardware complexity, higher power consumption, and decreased efficiency. Furthermore, current polynomial multipliers frequently fall short of offering the best possible trade-off between power consumption, speed, and area. These constraints impede the practical application of post-quantum cryptography in resource-constrained devices like embedded systems and Internet of Things nodes. An optimized hardware accelerator that lowers computational overhead while preserving high speed and effective resource use is therefore desperately needed.

4. PROJECT SCOPE :

The design, implementation, and assessment of an FPGA accelerator based on Verilog for effective polynomial multiplication in post-quantum cryptography systems are all included in the project's scope. In order to increase performance metrics, it focuses on combining the Karatsuba algorithm with hardware optimization strategies including pipelining and time-multiplexing. FPGA development tools are used for the implementation, and the outcomes are examined in terms of throughput, latency, and area utilization. Additionally, the project involves using simulation and synthesis to validate the suggested design, guaranteeing its accuracy and effectiveness under various operating circumstances. The solution is intended for embedded and lightweight applications, which makes it appropriate for edge computing systems and Internet of Things devices.

In order to further improve system performance, future versions of this work might integrate with full PQC frameworks and investigate alternative effective multiplication algorithms.

5. ALGORITHM:

The Karatsuba divide-and-conquer method is used by the AEKA accelerator to multiply polynomials. The procedure iteratively divides polynomials into smaller parts, multiplies the sub-polynomials, and then combines the results using addition and subtraction operations rather than doing full polynomial multiplication. This greatly saves FPGA resources by reducing the number of multiplications from $n2n2$ to around $nlog⁡23nlog2\ 3$. To optimize throughput while limiting area and resource utilization, further optimization techniques are used, such as time-multiplexing, pipeline phases, and modular addition/subtraction.

6. EXISTING SYSTEM:

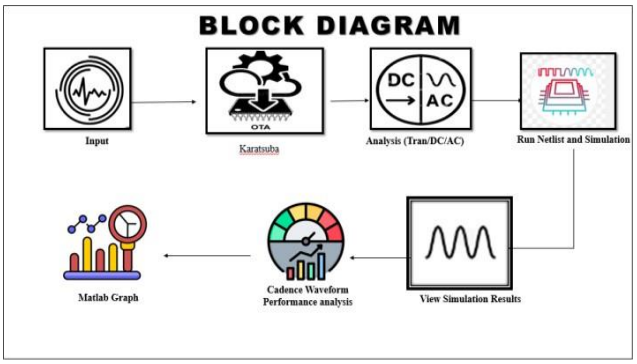
Conventional methods like schoolbook multiplication and simple Number Theoretic Transform (NTT)-based techniques are frequently employed in current implementations of polynomial multiplication for lattice-based cryptography systems. Although these methods are simple and quick to use, they frequently call for a lot of arithmetic operations, particularly adds and multiplications. They become ineffective for the huge polynomial sizes frequently employed in post-quantum cryptography as a result of the increasing computational complexity. Furthermore, these conventional techniques' hardware implementations on FPGA systems use a lot of resources, such as registers, lookup tables (LUTs), and DSP slices. Higher area utilization and latency are often the result of these architectures' absence of effective optimization strategies like resource sharing, pipelining, and time-multiplexing. As a result, these systems are not ideal for resource-constrained settings like embedded systems and the Internet of Things, where hardware efficiency and performance are crucial.

6.1. Disadvantages:

- A high computational complexity as a result of numerous multiplications
- Overuse of FPGA resources (registers, LUTs, and DSPs)
- Decreased processing speed and increased delay
- Unsuitable for devices with limited resources
- Limited optimization for power efficiency and area-delay

7. PROPOSED SYSTEM:

The suggested system presents AEKA, an area-efficient FPGA-based accelerator for polynomial multiplication in Ring-Binary-LWE cryptographic systems that was created using Verilog. It makes use of the Karatsuba algorithm, which uses a divide-and-conquer tactic to drastically cut down on the number of multiplication operations when compared to traditional techniques. Lower computing overhead and increased hardware efficiency are the immediate results of this reduction. To further improve performance, the suggested architecture incorporates cutting-edge architectural strategies including pipelining and time-multiplexing. While time-multiplexing allows for effective hardware resource reuse, pipelining boosts throughput by enabling simultaneous processing of processes. Xilinx Vivado is used to implement and validate the design, which shows enhanced area-delay product, decreased resource consumption, and increased operating efficiency, making it ideal for embedded and lightweight cryptography applications.



7.1. Architecture Diagram

Fig.1. Architecture diagram of proposed system

7.2. Working process:

For Ring-Binary-LWE cryptographic systems, the AEKA system architecture is built as a modular FPGA-based accelerator that effectively completes polynomial multiplication. The input interface, control unit, Karatsuba computing core, intermediate storage registers, and output module are the main functional elements that make up the architecture. In order to govern operation flow, the control unit processes polynomial coefficients that are received through the input interface.

Polynomial Representation (Ring-Binary-LWE)

Let two polynomials be:

$$A(x) = \sum_{i=0}^{n-1} a_i x^i, \quad B(x) = \sum_{i=0}^{n-1} b_i x^i$$

Polynomial multiplication in the ring:

$$C(x) = A(x) \cdot B(x) \mod (x^n + 1)$$

By recursively dividing input polynomials into smaller sections, the Karatsuba core reduces the number of multiplications needed to conduct divide-and-conquer multiplication.

Karatsuba Decomposition

Split polynomials into two halves:

$$A(x) = A_0(x) + x^m A_1(x), \quad B(x) = B_0(x) + x^m B_1(x)$$

Then multiplication becomes:

$$C(x) = A_0B_0 + x^m[(A_0 + A_1)(B_0 + B_1) - A_0B_0 - A_1B_1] + x^{2m}A_1B_1$$

The final polynomial output is created by combining intermediate results that are kept in registers. The architecture incorporates time-multiplexing and pipelining techniques to improve performance.

$$C(x) = (A_0B_0) + x^m[(A_0 + A_1)(B_0 + B_1) - A_0B_0 - A_1B_1] + x^{2m}A_1B_1 \mod (x^n + 1)$$

By dividing the computation into several stages, pipelining increases throughput by enabling the simultaneous processing of various data sources. By sharing processing units across several tasks, time-multiplexing reduces area consumption and allows for the reuse of hardware resources. The system is appropriate for high-speed and resource-constrained applications since the overall architecture guarantees effective use of FPGA resources including DSP slices, LUTs, and flip-flops.

FLOWCHART

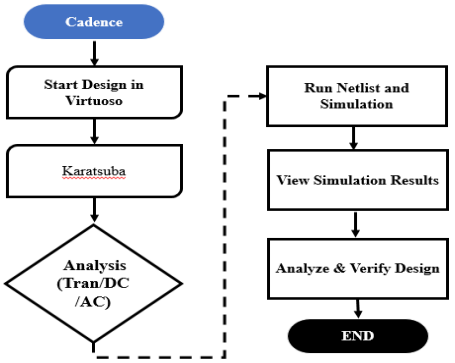


Fig.2. Flow chart diagram

The AEKA system's flow diagram starts with input initialization, which loads polynomial coefficients into the system. The input polynomials are then divided into smaller pieces by the control unit, which initiates the Karatsuba multiplication process. Intermediate outcomes are created by recursively multiplying and adding these segments. The process proceeds through several phases of computing, where the final polynomial product is produced by combining the results using addition and subtraction operations. The output module receives the computed result and stores it or transmits it for additional cryptographic processing. Additionally, pipeline phases are included in the flow diagram to guarantee uninterrupted data processing. Data flow between stages is managed by conditional control logic, which guarantees synchronization and effective execution. High-speed operation is made possible by this structured flow, which also reduces device complexity.

7.3. Advantages :

- The Karatsuba algorithm was used to reduce computational complexity.
- Effective use of FPGA resources (reduced DSP, LUT, and register consumption)
- Enhanced throughput as a result of pipelining
- Time-multiplexing reduces hardware footprint
- Fit for embedded systems and the Internet of Things with limited power and space requirements

8. LITERATURE SURVEY:

1. AxRLWE, a multilevel approximation Ring-LWE co-processor created especially for lightweight IoT applications, was proposed by A. Khalid et al. (2021) in [17]. In order to achieve acceptable cryptographic correctness while lowering hardware complexity, power consumption, and memory utilization, the architecture incorporates approximation computing techniques. The architecture strikes a compromise between resource efficiency and performance by utilizing multilevel approximation in polynomial arithmetic operations. When compared to traditional precise implementations, the co-processor shows notable gains in throughput and energy efficiency, which makes it ideal for embedded systems with limited resources. The promise of approximation computing to enable scalable and effective post-quantum cryptography solutions for Internet of Things contexts is

highlighted in this article.

2. Because Ring-Learning With Errors (Ring-LWE) systems are resistant to quantum assaults, recent developments in post-quantum cryptography (PQC) have concentrated on effective hardware implementations of these techniques. For binary Ring-LWE cryptosystems, J. Xie et al. (2022) suggested an effective hardware implementation of finite field arithmetic of the type ab + c in [23]. In order to drastically lower computing complexity and hardware resource consumption, their study focuses on efficient arithmetic operations in binary fields. Real-time cryptography applications can benefit from the design's increased throughput and low latency.

3. B. J. Lucas et al. (2022) presented a lightweight hardware accelerator for binary Ring-LWE-based PQC in [24]. The suggested architecture focuses on reducing silicon space and power consumption, two factors that are crucial for embedded and Internet of Things devices. The approach guarantees great performance while upholding strict security criteria by optimizing polynomial multiplication and modular reduction techniques. This work shows that PQC algorithms can be implemented in contexts with limited resources.

4. Additionally, J. L. Imana et al. (2022) reported an effective hardware arithmetic design for inverted binary Ring-LWE cryptography in [25]. By reorganizing the inversion procedure in binary polynomial rings, their method improves arithmetic operations and lowers hardware complexity. The suggested design is appropriate for next-generation secure communication systems because it provides improved scalability and facilitates fast cryptographic operations.

9. IMPLEMENATION PROCESS:

9.1. TOP-LEVEL AEKA MODULE:

The FPGA-based accelerator's central integration unit is the Top-Level AEKA module. It ensures smooth communication and coordination amongst all submodules, including the control unit, adder/subtractor, polynomial splitter, and Karatsuba multiplier. This module receives polynomial coefficients and outputs the final multiplication results, serving as the interface between external inputs and internal processing units. Coordinating data flow between submodules like the control unit, polynomial splitter, adder/subtractor, and Karatsuba multiplier, the Top-Level AEKA module serves as the FPGA-based accelerator's primary integration and control unit. In order to produce the final result, it takes input polynomials A(x) and B(x), breaks them down into sub-polynomials, and coordinates their processing. The overall process can be stated as follows:

$$C(x) = A(x) \cdot B(x) \mod (x^n + 1) = A_0 B_0 + x^m$$

where the Top-Level module manages splitting $A(x)$
Karatsuba core, and recombining intermediate outpu
ensuring correct sequencing and pipeline alignment
pipelining, with latency defined as $\text{Latency} = N_{sta}$
steady-state operation. Additionally, time-multiplexin
$\frac{\text{Active Time}}{\text{Total Time}}$. By integrating all components into a unif
scalability, efficient FPGA resource usage, and seamle
systems.

It is in charge of controlling the flow of data between various processing stages. After receiving input polynomials and sending them to the splitter for segmentation, the module gathers the Karatsuba multiplier's processed outputs. By communicating with the control unit, which sets time and execution order, it also guarantees correct synchronization across modules. Furthermore, the Top-Level module is essential to the integration and scalability of the system. It makes it simple to integrate the design into entire cryptographic systems or expand it for higher-order polynomial computations. It guarantees effective resource use and optimized operation by arranging all parts into a single structure.

9.2. POLYNOMIAL SPLITTER MODULE:

The input polynomials must be divided into smaller pieces in order for the Karatsuba algorithm to work. This is done via the Polynomial Splitter module. This module divides the input polynomials into high and low portions so that recursive computing is possible since Karatsuba multiplication employs a divide-and-conquer technique. This module divides polynomial coefficients into two or more pieces according to the polynomial's degree. By dividing a big problem into smaller subproblems, the splitting process lowers the difficulty of multiplication.

Polynomial Representation

Let two input polynomials be:

$$A(x) = A_H(x) \cdot x^m + A_L(x)$$

$$B(x) = B_H(x) \cdot x^m + B_L(x)$$

- $A_H(x), B_H(x)$: Higher-degree parts
- $A_L(x), B_L(x)$: Lower-degree parts
- m: Splitting point (usually $m = \lceil n/2 \rceil$, where n is polynomial degree)

Splitting Operation

If a polynomial is:

$$A(x) = \sum_{i=0}^{n} a_i x^i$$

Then it is split as:

$$A_L(x) = \sum_{i=0}^{m-1} a_i x^i$$

$$A_H(x) = \sum_{i=m}^{n} a_i x^{i-m}$$

Additionally, it formats the data for pipelined or parallel processing. Additionally, by allowing the simultaneous processing of smaller polynomial segments, the Polynomial Splitter improves computational efficiency. It guarantees that data is correctly aligned and kept for use in subsequent processes.

Karatsuba Intermediate Products

After splitting:

$$P_0 = A_L(x) \cdot B_L(x)$$

$$P_1 = A_H(x) \cdot B_H(x)$$

$$P_2 = (A_L(x) + A_H(x)) \cdot (B_L(x) + B_H(x))$$

Final Combination Formula

$$A(x) \cdot B(x) = P_1 \cdot x^{2m} + (P_2 - P_1 - P_0) \cdot x^m + P_0$$

This module is crucial for lowering computational overhead and facilitating the effective hardware implementation of the Karatsuba algorithm.

9.3. KARATSUBA MULTIPLIER MODULE:

The core computational component of the AEKA system is the Karatsuba Multiplier. Compared to conventional approaches, it uses the Karatsuba algorithm to conduct polynomial multiplication, which minimizes the number of multiplications. It improves efficiency by reducing the number of multiplications for two halves of polynomials from four to three. Until a base condition is met, this module divides polynomial segments into smaller portions in a recursive manner. These smaller pieces are multiplied, and the results are combined using addition and subtraction. Performance is much enhanced and computational complexity is greatly decreased by this recursive structure. Pipelining and parallel processing techniques are used to optimize the Karatsuba Multiplier in hardware implementation.

$$A(x)B(x) = A_1B_1 \cdot x^{2m} + \left[(A_1 + A_0)(B_1 + B_0) - A_1B_1 - A_0B_0\right] \cdot x^m + A_0B_0$$

This is the exact formula used in your Karatsuba Multiplier Module, enabling:

- Reduced multiplication count
- Lower computational complexity $O(n^{\log_2 3}) \approx O(n^{1.585})$
- Efficient FPGA implementation with pipelining and parallelism

The system is appropriate for real-time cryptographic applications because of these optimizations, which guarantee fast throughput and low latency. The module contributes to the overall area-efficient architecture by making effective use of FPGA resources.

9.4. ADDER/SUBTRACTOR MODULE:

The arithmetic operations needed to aggregate the intermediate outputs produced by the Karatsuba multiplier are carried out by the Adder/Subtractor module. This module is essential to the computation process since the Karatsuba method mostly uses addition and subtraction to reconstruct the final polynomial. It carries out necessary arithmetic operations like summation and difference computation using intermediate outputs from multiplication steps. In order to combine partial results into a final polynomial output, these actions are necessary.

Final Output Reconstruction

The module also helps assemble the final polynomial:

$$P(x) = Z_2 \cdot x^{2m} + Z_1 \cdot x^m + Z_0$$

The module's architecture ensures low delay by effectively handling various processes. Additionally, the speed and resource efficiency of the Adder/Subtractor module are optimized. Combinational and sequential logic are used in its implementation to enable pipelined execution. This maintains a balance between speed and hardware utilization by preventing arithmetic operations from becoming a bottleneck in the system's overall performance.

9.5. CONTROL AND PIPELINING MODULE:

The AEKA system's overall functioning and timing are managed by the Control and Pipelining module. It produces control signals that ensure correct sequencing and synchronization of processes by coordinating the execution of several modules. This module controls execution phases and data flow, acting as the system's brain. By splitting the computation into several steps, it implements pipelining, enabling the execution of several operations at once. As a result, processing time is decreased and throughput is greatly increased. Every stage is guaranteed to function properly, free from timing problems or data conflicts, thanks to the control logic. In order to maximize resource use, this module also uses time-multiplexing techniques. It lowers the total area needed for implementation by reusing hardware components across several activities. The AEKA system is ideal for high-performance and resource-constrained applications because the Control and Pipelining module guarantees effective execution.

10. RESULTS AND DISCUSSION:

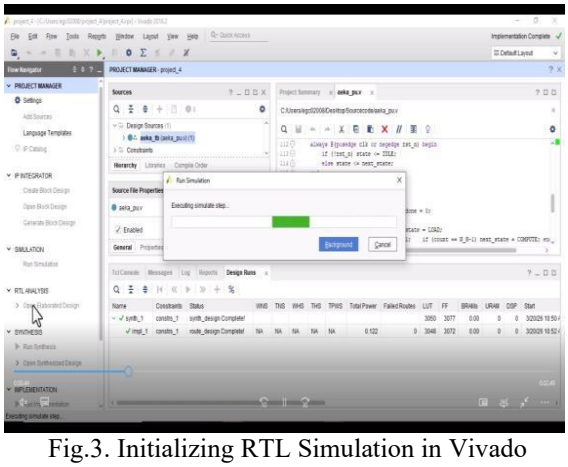


Fig.3. Initializing RTL Simulation in Vivado

This picture shows the Xilinx Vivado Design Suite's first hardware verification stage. A simulation run for the Verilog module aeka_pu has been initiated by the user.v. The tool is now carrying out the simulation stage, as indicated by the "Run Simulation" progress dialog. The source code for the project is displayed in the background, and the "Design Runs" window at the bottom shows that

the FPGA project's synthesis (synth_1) and implementation (impl_1) stages have already been successfully finished.

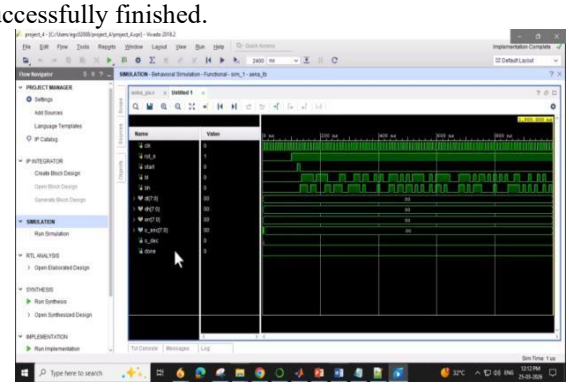


Fig.4. Behavioral Simulation Waveform Analysis

With an emphasis on the aeka_tb (testbench) top-level module, this screenshot shows the behavioral simulation results in the Vivado waveform viewer. A timing diagram displaying different digital signals, including the clock (clk), active-low reset (rst_n), and data buses like dl[7:0] and dh[7:0], is provided by the viewer. High pulses on start and toggling patterns on data lines show how the design functions over a 1000-nanosecond period. Logic levels are represented by green lines. For hardware developers to confirm that the RTL (Register Transfer Level) logic functions correctly in accordance with timing specifications, this visual representation is essential.

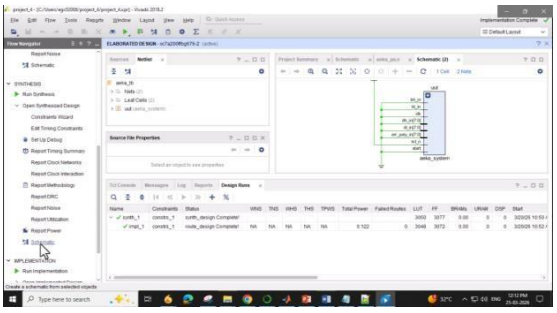


Figure 5: RTL Diagram of the Detailed Design

Vivado displays the schematic depiction of the "Elaborated Design" in this view. This step comes after the Verilog/VHDL code has been parsed by the tool, but before it has been mapped to particular FPGA gates (synthesis). A high-level block diagram of the Unit Under Test (uut), designated as aeka_system, is displayed in the center pane. The module's main input and output ports are highlighted in the schematic, along with multi-bit data buses like err_poly_in[7:0] and control signals like clk and rst_n. Designers may visually verify the structural hierarchy and connectivity of their hardware modules thanks to this abstraction.

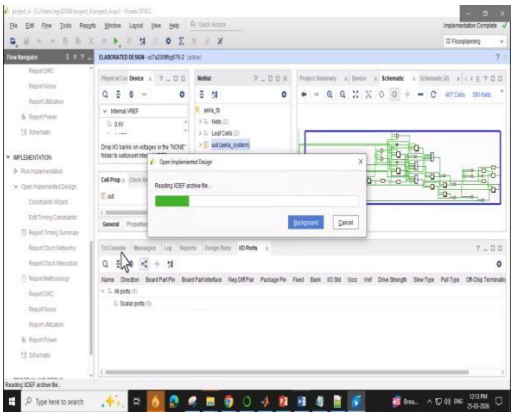


Figure 6: Gate-Level Schematic and Implemented Design Loading

The last phases of the FPGA design flow are depicted in this figure, when the user is opening the "Implemented Design." The utility is "Reading XDEF archive file," which has the finalized routing and placement data, according to a progress dialog. Compared to the earlier detailed depiction, a far more intricate gate-level schematic is visible behind the dialog. The actual hardware structure that will be written onto the device is reflected in this schematic, which shows how the design has been physically mapped onto the FPGA's internal resources, such as Look-Up Tables (LUTs) and Flip-Flops (FFs).

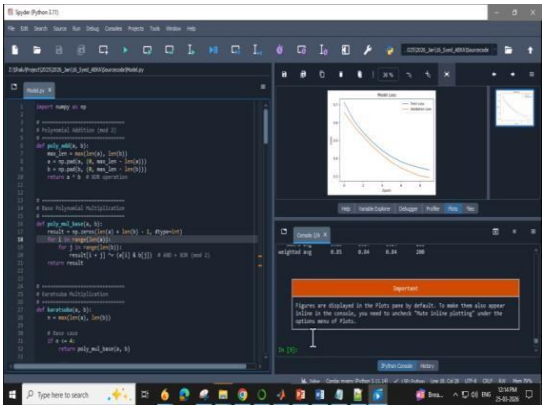


Figure 7: Modeling and Training Python Software in the Spyder IDE

This picture illustrates the transition from hardware design to software development, with Python code being written and run using the Spyder IDE. A Karatsuba multiplication method is one of the mathematical functions for polynomial arithmetic included in the script Model.py. The "Model Loss" graph in the top-right pane illustrates how the training and validation loss curves decrease over ten epochs, which is typical of the training process for machine learning models. This software element most likely acts as a data processing script or reference model to verify the functionality of the hardware modules created in the earlier Vivado stages.

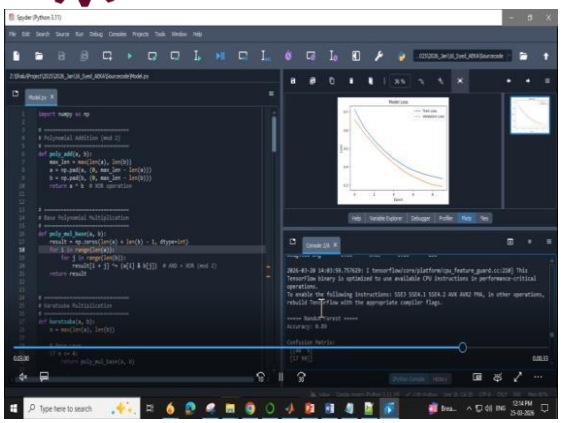


Figure 8: Evaluation of Machine Learning Models and Performance Measures

The results of the Python script's execution on the Spyder console are the main focus of this last image. The console shows the performance data for a "Random Forest" classifier following a sequence of system alerts about TensorFlow tuning. The accuracy of the model was 0.89 (89%). A confusion matrix that provides a thorough breakdown of accurate and inaccurate predictions for several classes is presented beneath the accuracy. These outcomes serve as the project's software-side validation, offering statistical proof of the underlying algorithms' effectiveness on a particular dataset

11. CONCLUSION:

An area-efficient FPGA-based accelerator for polynomial multiplication in Ring-Binary-LWE post-quantum cryptography systems is successfully demonstrated by the AEKA project. When compared to traditional multiplication methods, the design greatly lowers computing cost by utilizing the Karatsuba algorithm. By increasing throughput and reducing the use of hardware resources, the combination of pipelining and time-multiplexing significantly improves system performance. The suggested method's viability and effectiveness are confirmed by the Verilog implementation and FPGA tool validation. Overall, the system strikes a good balance between power consumption, speed, and area, which makes it ideal for applications with limited resources like embedded systems and the Internet of Things. The research emphasizes how crucial algorithmic and architectural optimization is to provide secure and lightweight cryptographic systems. For upcoming post-quantum cryptography applications, AEKA offers a scalable and dependable platform.

12. FUTURE ENCHANCEMENT:

An area-efficient FPGA-based accelerator for polynomial multiplication in Ring-Binary-LWE post-quantum cryptography systems is successfully demonstrated by the AEKA project. When compared to traditional multiplication methods, the design greatly lowers computing cost by utilizing the Karatsuba algorithm. By increasing throughput and reducing the use of hardware resources, the combination of pipelining and time-multiplexing significantly improves system performance. The suggested method's viability and effectiveness are confirmed by the Verilog implementation and FPGA tool validation. Overall, the system strikes a good balance between power consumption, speed, and area, which makes it ideal for applications with limited resources like embedded systems and the Internet of Things. The research emphasizes how crucial algorithmic and architectural optimization is to provide secure and lightweight cryptographic systems. For upcoming post-quantum cryptography applications, AEKA offers a scalable and dependable platform.

13. REFERENCES:

[1] Daniel J. Bernstein and Lange Tanja. 2017. Post-quantum cryptography. Nature 549, 7671 (2017), 188–194.

[2] Post-Quantum Cryptography. Retrieved from https://csrc.nist.gov/Projects/post-quantum cryptography

[3] PQC Standardization Process: Announcing Four Candidates to be Standardized Plus Fourth Round Candidates. Retrieved from https://csrc.nist.gov/News/2022/pq c-candidates-to-be-standardized-and-round-4

[4] National Science Foundation (NSF) 2022 Secure and Trustworthy Cyberspace Principal Investigators' Meeting (SaTC PI Meeting'22) Break Out Group Reports/Slides: Security in a Post-Quantum World. (2022), slides page 4. Retrieved from https://cps-vo.org/group/satc-pimtg22/breakouts

[5] D. Micciancio and C. Peikert. 2013. Hardness of SIS and LWE with small parameters. In Advances in Cryptology – CRYPTO 2013 (Lecture Notes in Computer Science). Springer21–39.DOI:10.1007/978-3-642-40041-4_2 DOI: 10.1007/978-3-642-40041-4_2

[6] F. Gopfert, C. Vredendaal, and T. Wunderer. 2017. A hybrid lattice basis reduction and quantum search attack on LWE. In Proceedings of the International Workshop on Post-Quantum Cryptography, Springer, Cham, 184–202.

[7] J. Buchmann, F. Gopfert, T. Guneysu, T. Oder, and T. Poppelmann. 2016. High-performance and lightweight lattice-based public-key encryption. In Proceedings of the 2nd ACM International Workshop on IoT Privacy, Trust, and Security, 1–8.

[8] B. Sreehari, V. Sankar, R. S. Lopez, K. S. Vaishnav, and C. M. Stuart. 2023. A review on FPGA implementation of lightweight cryptography for wireless sensor network. In Proceedings of the 2023 International Conference on Power, Instrumentation, Control and Computing (PICC), IEEE, 1–6.

[9] K. Shahbazi and S.-B. Ko. 2021. Area and power efficient post-quantum cryptosystem for IoT resource-constrained devices. Microprocessors and Microsys tems 84 104280. DOI: 10.1016/j.micpro.2021.10428 0

[10] J. Xie, P. He, X. Wang, and J. L. Imaña. 2022. Efficient hardware implementation of finite field arithmetic \(AB+C\) for binary Ring-LWE based post-quantum cryptography. IEEE Transactions on Emerging Topics in Computing 10, 2 (2022), 1222–1228.

[11] P. He T. Bao J. Xie and M. Amin. 2022. FPGA implementation of compact hardware accelerators for Ring-Binary-LWE based post-quantum cryptography. ACM Transactions on Reconfigurable Technology and Systems 1–24. DOI: 10.1145/3569457

[12] J. Pollard. 1971. The fast Fourier transform in a finite

field. Mathematics of computation 25, 114 (1971), 365–374.

[13] M. Liu and P. Nguyen. 2023. Solving BDD by enumeration: An update. In Topics in Cryptology—CT-RSA 2013, E. Dawson, ed., Springer Berlin, vol. 7779, 293–309.

[14] J.-P. Deschamps J. L. Imana and G. D. Sutter. 2009. Hardware implementation of finite-field arithmetic McGraw-Hill Companies Inc.

[15] A. Karatsuba and Y. Ofman. 1963. Multiplication of multidigit numbers on automata. In Soviet Physics Doklady 7, 7 (1963), 595–596.

[16] Y. Agus, M. A. Murti, F. Kurniawan, N. D. Cahyani, and G. B. Satrya. 2020. An efficient implementation of NTRU encryption in post-quantum internet of things. In Proceedings of the 2020 27th International Conference on Telecommunications (ICT), IEEE, 1–5.

[17] A. Khalid, S. Bian, C. Wang, M. O'Neill, and W. Liu. 2021. Axrlwe: A multilevel approximate ring-lwe co-processor for lightweight IoT applications. IEEE Internet of Things Journal 9, 13(2021), 10492–10501.

[18] N. Zhang B. Yang C. Chen S. Yin S. Wei and L. Liu. 2020. Highly efficient architecture of NewHope-NIST on FPGA using low-complexity NTT/INTT. IACR Transactions on Cryptographic Hardware and Embedded Systems (2020) 49–72. DOI:10.13154/tches.v2020.i2.49-72

[19] Y. Xing and S. Li. 2021. A compact hardware implementation of CCA-secure key exchange mechanism CRYSTALS-KYBER on FPGA. IACR Transactions on Cryptographic Hardware and Embedded Systems. 328–356. DOI:10.46586/tches.v2021.i2.328-356

[20] J. Howe, C. Moore, M. O'Neill, F. Regazzoni, T. Guneysu, and K. Beeden. 2016. Lattice-based encryption over standard lattices in hardware. In Proceedings of the 53rd Annual Design Automation Conference (DAC), 162.

[21] S. Roy, F. Vercauteren, N. Mentens, D. Chen, and I. Verbauwhede. 2014. Compact ring-LWE cryptoprocessor. In Proceedings of the International Workshop on Cryptographic Hardware and Embedded Systems, 371–391.

[22] Viet B. Dang, Farnoud Farahmand, Michal Andrzejczak, Kamyar Mohajerani, Duc T. Nguyen, and K. Gaj. 2020. Implementation and benchmarking of round 2 candidates in the NIST post-quantum cryptography standardization process using hardware and software/hardware co-design approaches. Cryptology ePrint Archive: Report 2020/795 (2020). https://par.nsf.gov/biblio/10175000

[23] J. Xie, P. He, X. Wang, and J. L. Imana, "Efficient hardware imple-mentation of finite field arithmetic ab + c for binary ring-lwe based post-quantum cryptography," IEEE Transactions on Emerging Topics in Computing, vol. 10, no. 2, pp. 1222–1228, 2022.

[24] B. J. Lucas and et al., "Lightweight hardware implementation of binary ring-lwe pqc accelerator," IEEE Computer Architecture Letters, 2022.

[25] J. L. Imana, P. He, T. Bao, Y. Tu, and J. Xie, "Efficient hardware arith- ˜ metic for inverted binary ring-lwe based post-quantum cryptography," IEEE Transactions on Circuits and Systems I: Regular Papers, 2022.