Unified Cryptography Standards on RISC-V

Author:
Daniel Sanz Sobrino

October 11, 2024

Supervisors:
Sujoy Sinha Roy - IAIK, TU Graz
Aikata Aikata - IAIK, TU Graz
Andres Marín López - UPM

Introduction and Context

- **General Objective:** Integration of cryptographic modules in RISC-V architecture.

Introduction and Context

- **General Objective:** Integration of cryptographic modules in RISC-V architecture.

1. Privacy Preserving Computation through Hybrid Homomorphic Encryption.

Introduction and Context

- **General Objective:** Integration of cryptographic modules in RISC-V architecture.

1. Privacy Preserving Computation through Hybrid Homomorphic Encryption.
2. Post-quantum Security via Kyber and Dilithium algorithms.

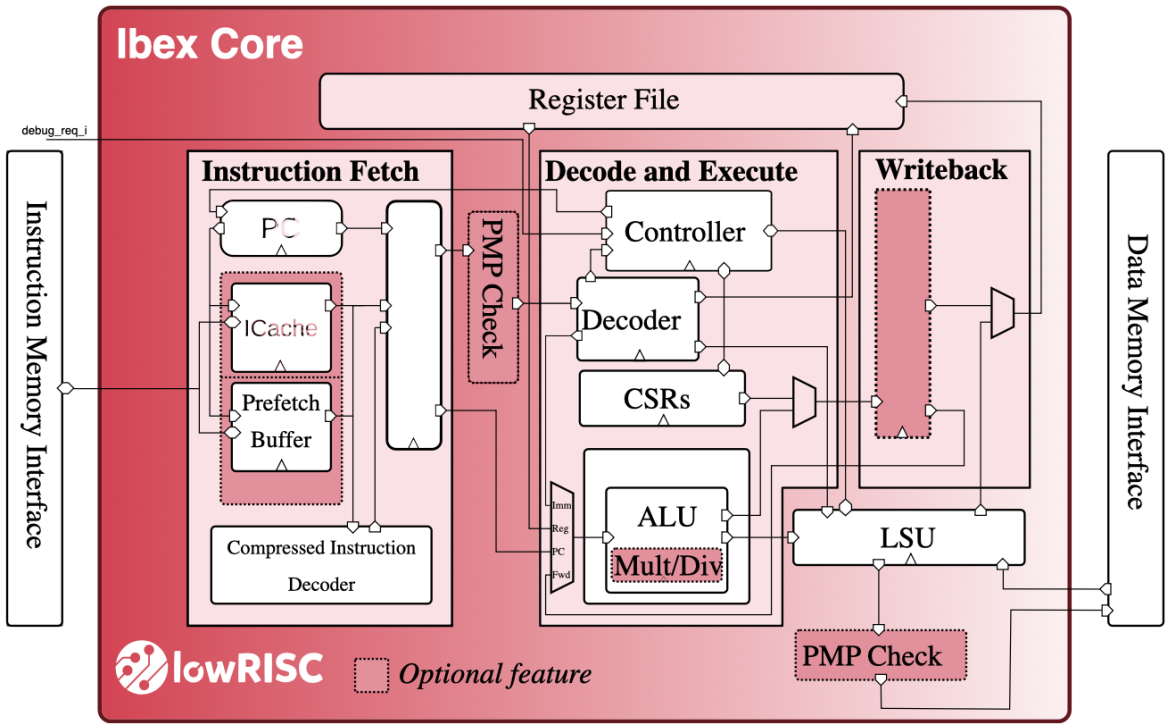
Introduction and Context

- **General Objective:** Integration of cryptographic modules in RISC-V architecture.

1. Privacy Preserving Computation through Hybrid Homomorphic Encryption.
2. Post-quantum Security via Kyber and Dilithium algorithms.
3. Evaluate the frecuency, area, power and performance of integrated modules.

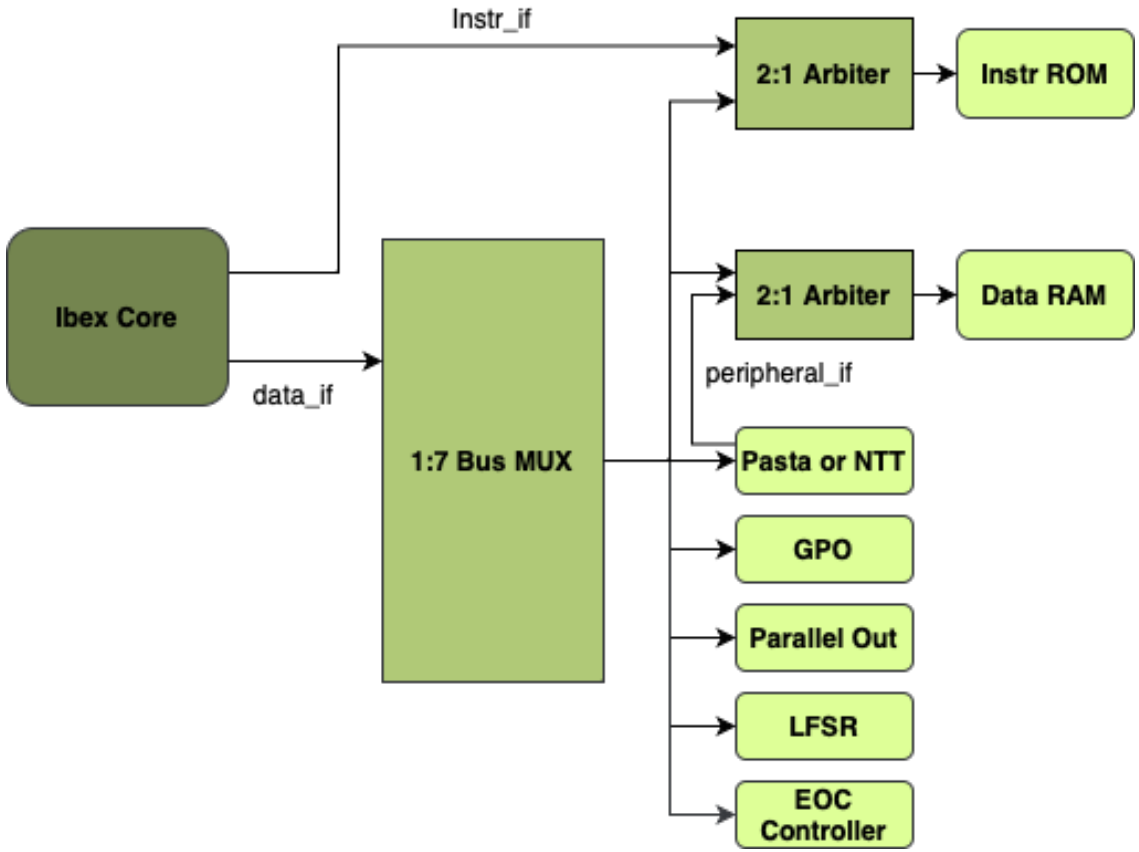
RISC-V Architecture

- An open-source Instruction Set Architecture designed for flexibility and customization.
- Low Power Consumption.
- Support and compatibility with a wide range of applications.



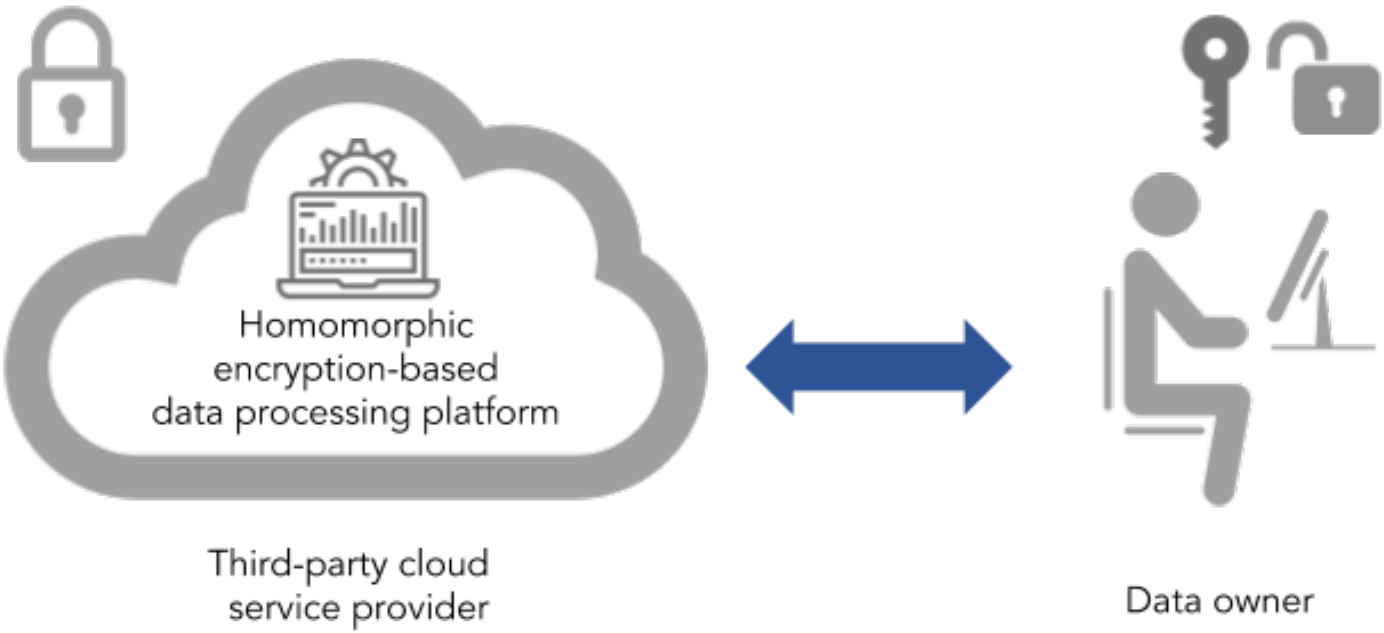
Architecture

- Ibex Core as the central processing unit, connected to RAM, ROM, and functional peripherals.
- Master-slave 32 bit bus configuration.
- Loosely Coupled Design.



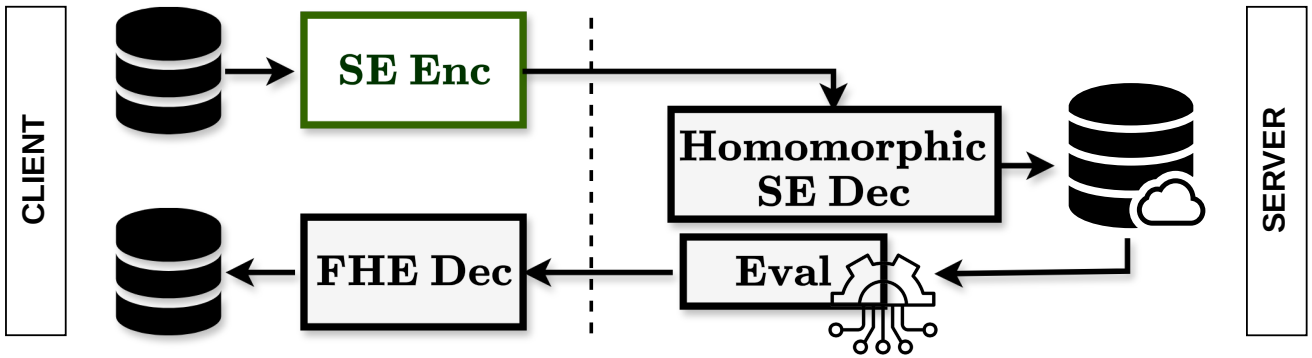
Fully Homomorphic Encryption (FHE)

- Enhances privacy by keeping data encrypted during processing.
- Requires considerable computational resources on the user side.



Hybrid Homomorphic Encryption (HHE)

- Combines symmetric encryption for data transfer with FHE for secure computation.
- Encryption requires fewer computational resources on the user side.
- Communication cost is significantly lower compared to FHE.



Methodology

1. Hardware module testing in Vivado to verify functionality.

Methodology

1. Hardware module testing in Vivado to verify functionality.
2. Software-hardware functionality test through C++/Python.

Methodology

1. Hardware module testing in Vivado to verify functionality.
2. Software-hardware functionality test through C++/Python.
3. IBEX Architecture Integration using SystemVerilog.

Methodology

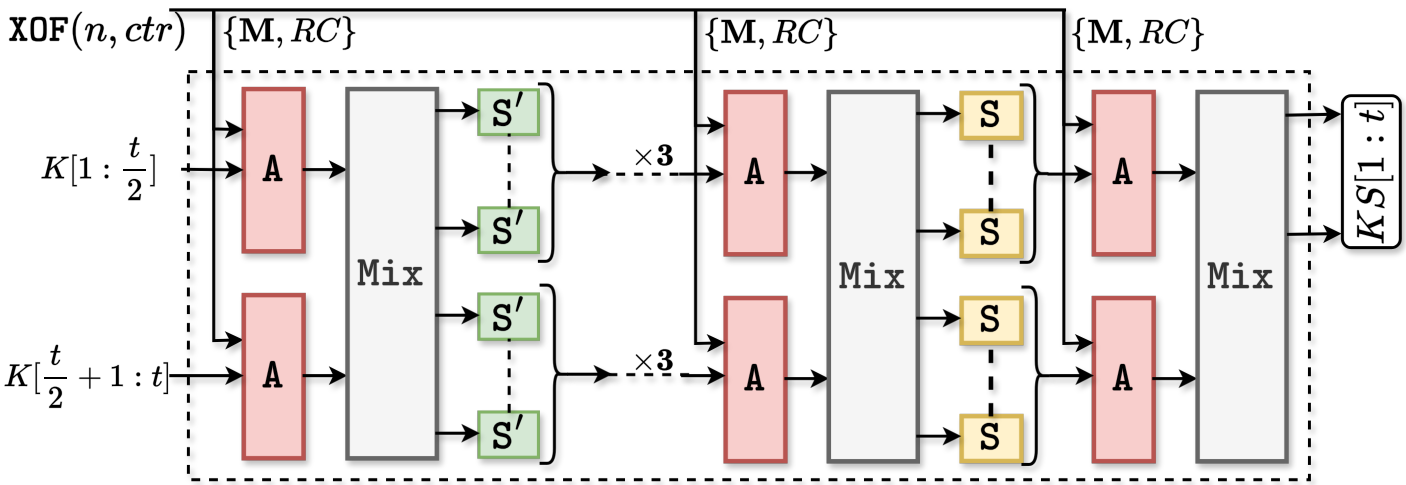
1. Hardware module testing in Vivado to verify functionality.
2. Software-hardware functionality test through C++/Python.
3. IBEX Architecture Integration using SystemVerilog.
4. Simulation using a C file that sends and receives data within the memory address range of the peripheral.

Methodology

1. Hardware module testing in Vivado to verify functionality.
2. Software-hardware functionality test through C++/Python.
3. IBEX Architecture Integration using SystemVerilog.
4. Simulation using a C file that sends and receives data within the memory address range of the peripheral.
5. HDL and synthesis compilation and simulation via a Makefile using iaikflow and TUG cluster.

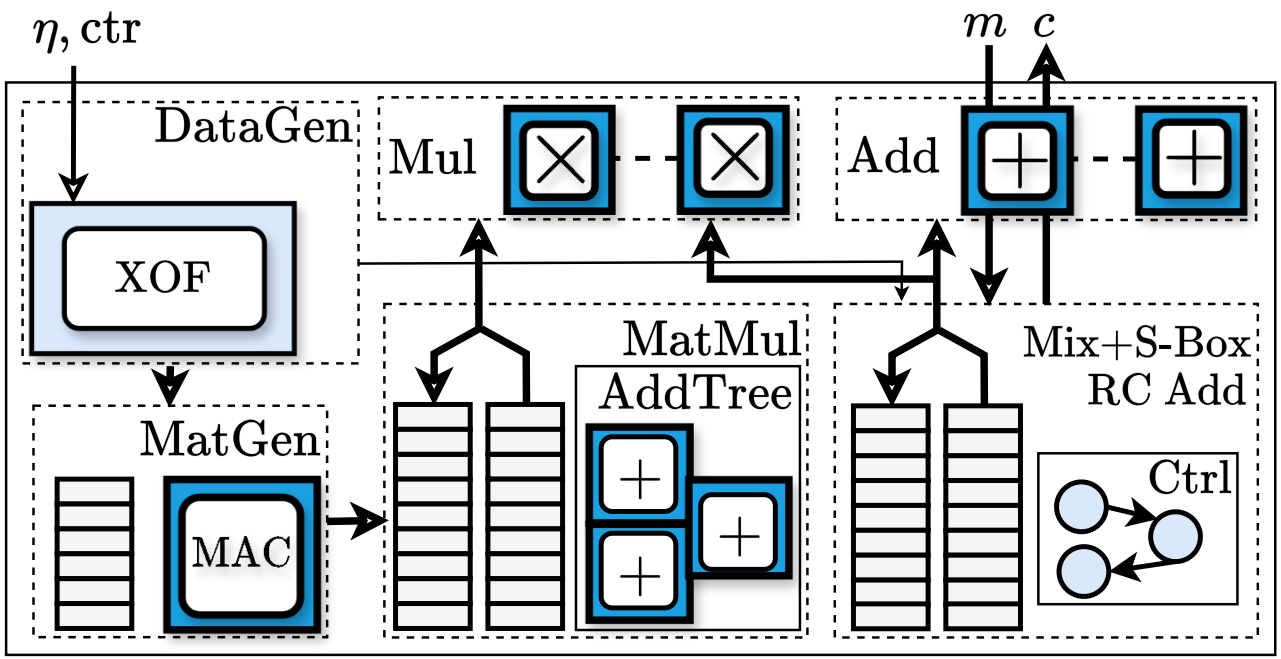
Pasta HHE Algorithm

- The algorithm processes plaintext blocks - and performs key permutations followed by plaintext addition.
- PASTA-3 or PASTA-4 comprise 3 or 4 permutation rounds.



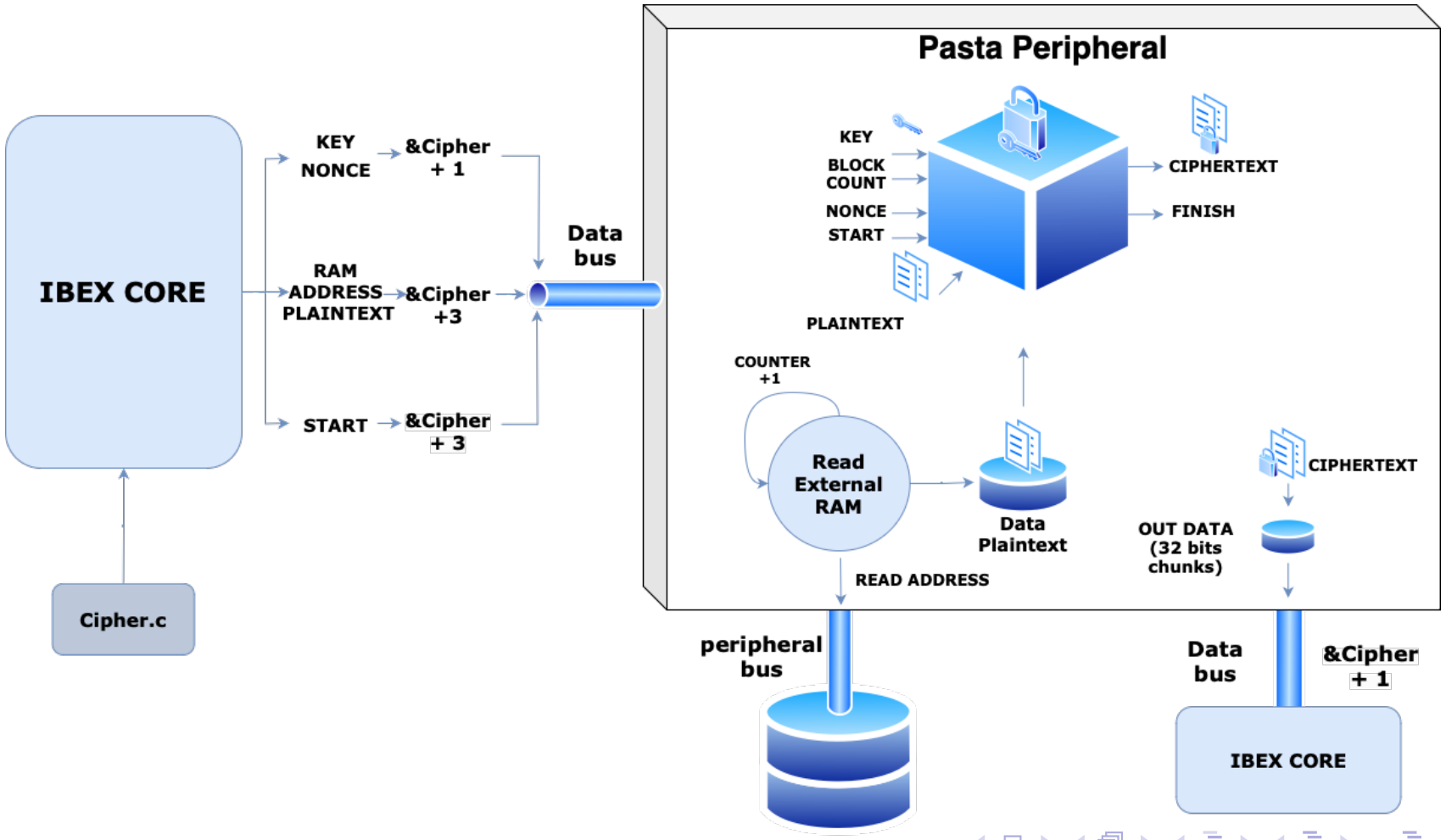
Pasta Hardware Design

- High throughput design tailored to the Rejection sampling rate.
- Uses SHAKE 128 for fast pseudo-random number generation.
- Resource sharing via common Modular multiplication/addition units.
- Parallel Matrix generation & multiplication for reduced memory footprint.



Pasta Integration

- Parallel execution of Encryption and Plaintext read for high efficiency.
- Master-Slave read/write mode for sending/receiving data to/from the peripheral.



Results for Pasta

- Results for 130 nm technology with a frequency of 10 MHz.

Metric	Max Freq (MHz)	Total Power (W)	Peripheral Area (mm^2)	Total Area (mm^2)
Pasta-4 (17-bit)	18.58	0.0094	1.7833	4.6069

Clock Cycles	Encryption Latency (μs)
1590	159

Paper Outcome

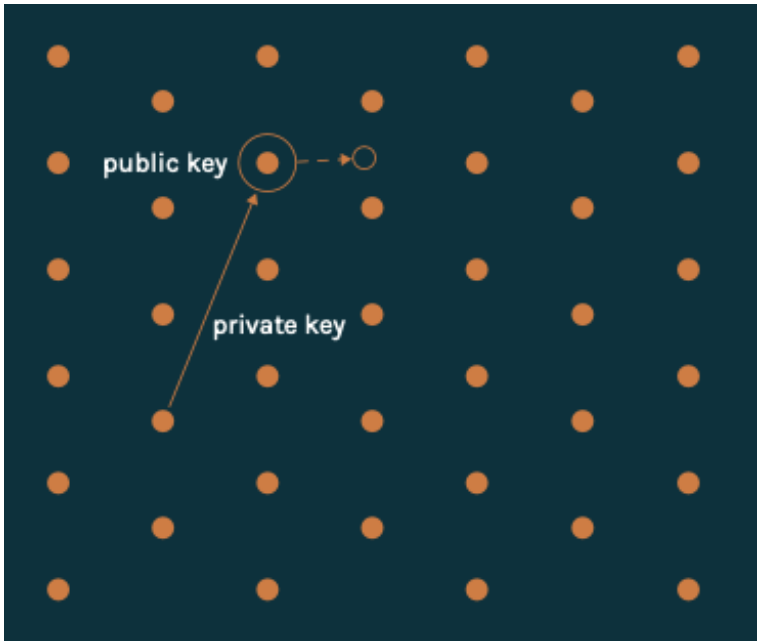
- First successful integration of Hybrid Homomorphic Encryption scheme in RISC-V.
- Paper currently in Submission.

PASTA on Edge: Cryptoprocessor for Hybrid Homomorphic Encryption

Aikata Aikata, Daniel Sanz Sobrino, Sujoy Sinha Roy

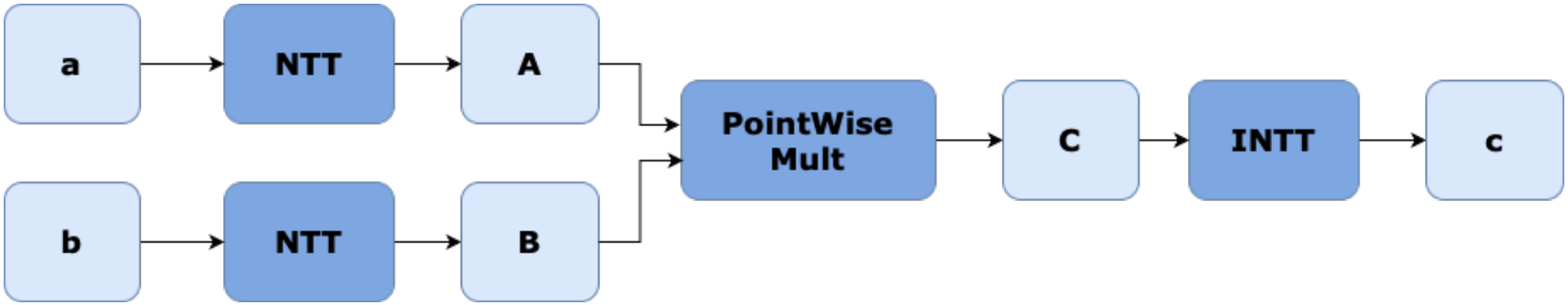
Post-Quantum Cryptography

- Public Key Infrastructure vulnerable to quantum attacks.
  - Discrete Logarithm
  - Integer Factorization
- Lattice-based Cryptography.
- New post-quantum standards August 2024 published by NIST.
  - Crystals-Kyber - FIPS 203 (KEM).
  - Crystals-Dilithium - FIPS 204 (DSA).



Number Theoretic Transform (NTT)

- Enables efficient polynomial multiplication with $O(n \log n)$ complexity.

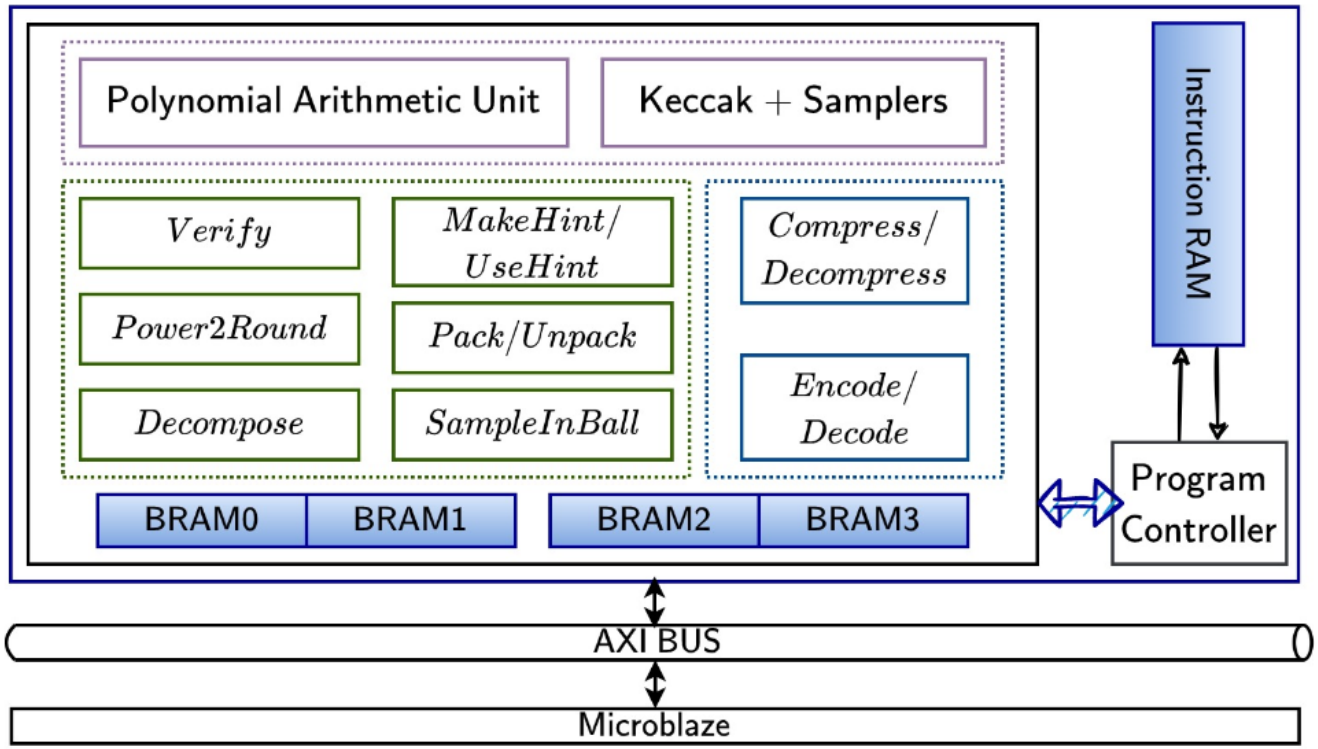


$$\boldsymbol{c} = INTT^{\psi^{-1}}(NTT^{\psi}(\boldsymbol{a}) \circ NTT^{\psi}(\boldsymbol{b}))$$

$$INTT(\begin{bmatrix}1467\\2807\\3471\\7621\end{bmatrix} \circ \begin{bmatrix}2489\\7489\\6478\\6607\end{bmatrix}) = INTT(\begin{bmatrix}2888\\6407\\2851\\2992\end{bmatrix})$$

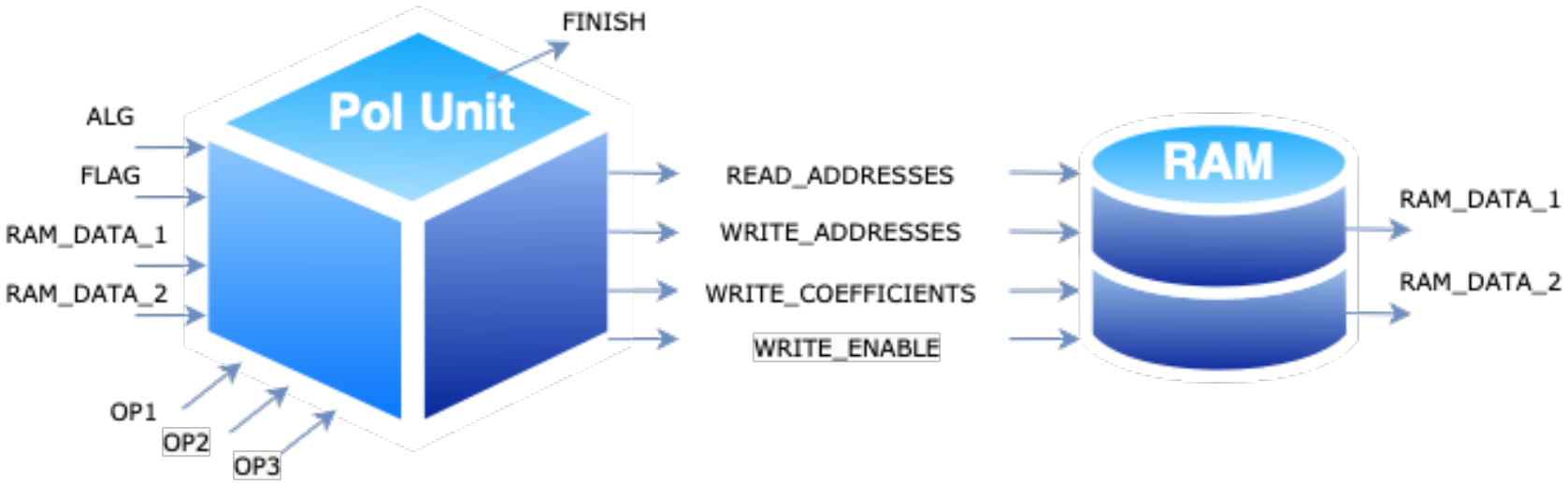
KaLi Architecture

- Unified architecture for *Crystals-Kyber* and *Crystals-Dilithium*.
- The Polynomial Arithmetic Unit, Keccak, and samplers together utilize **80%** of the available resources.
- Modulo q: 12 bits for Kyber and 23 bits for Dilithium.



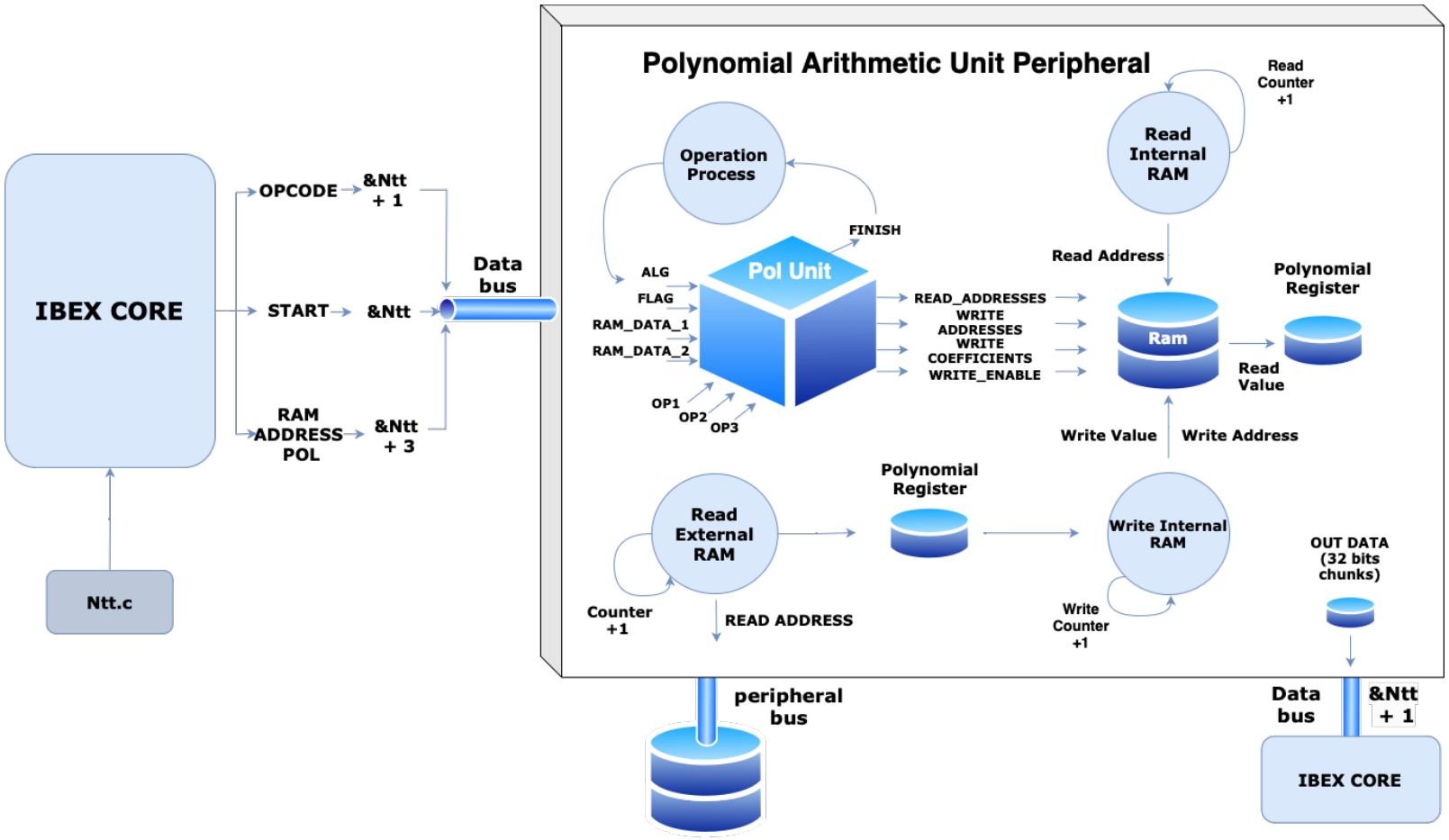
KaLi Polynomial Arithmetic Unit

- The 64-bit length of the module must be adapted to the 32-bit bus width of the RISC-V architecture.
- The hardware accelerator must support both algorithms.



Integration of the Polynomial Arithmetic Unit

- Two RAMs with 128 addresses and 64 bits per address.
- Opcodes for Kyber and Dilithium - Mult, Add, Sub, NTT, INTT, Read, and Write.



Performance Results for Polynomial Arithmetic Unit

- Results for 130 nm technology with a frequency of 10 MHz.

Module	Max Freq (MHz)	Total Power (W)	Peripheral Area (mm^2)	Total Area (mm^2)
Pol Arithmetic	18.63	0.00896	2.502 / 0.2825	6.7729

Operation	Dilithium		Kyber	
	Clock Cycles	Latency (μs)	Clock Cycles	Latency (μs)
NTT	527	52.7	239	23.9
Mult	141	14.1	154	15.4
Add	141	14.1	77	7.7
Sub	141	14.1	77	7.7
INTT	527	52.7	239	23.9

Future Directions

- Implement cryptographic modules using a tightly coupled design within RISC-V for improved efficiency.
- Integrate the complete KaLi architecture or critical modules directly into hardware for a hardware-software codesign.
- Optimize communication protocols and memory management techniques to enhance performance.

Unified Cryptography Standards on RISC-V

Author:
Daniel Sanz Sobrino

October 11, 2024

Supervisors:
Sujoy Sinha Roy - IAIK, TU Graz
Aikata Aikata - IAIK, TU Graz
Andres Marín López - UPM