Análise de Viabilidade de um Acelerador PQC para Open RAN: Uma Prova de Conceito via Co-Design de Hardware/Software

Mariana C. R. Oliveira [1], Leonardo B. F. Souza [1]

[1]Escola de Tecnologia e Comunicação – Universidade Católica de Pernambuco (UNICAP) Caixa Postal 50050-900 – Recife – PE – Brasil

{mariana.00000032581@unicap.br, leonardo.souza@unicap.br}

Resumo. *A evolução para redes 5G/6G Open RAN impõe latências estritas (< 250 µs), inviabilizando a adoção da criptografia pós-quântica (ML-KEM) devido ao alto tempo de processamento do SHA-3 em software (~1375 µs). Para preencher essa lacuna, este artigo propõe uma Prova de Conceito de aceleração via co-design hardware/software. O offloading da permutação Keccak para FPGA (Cyclone 10 LP) reduziu a latência lógica para 0,013 µs. Contudo, a aferição do ciclo híbrido via interface JTAG (390 µs) evidenciou um gargalo de transporte. Este diagnóstico empírico valida a arquitetura e direciona a adoção futura do barramento SPI para atender ao padrão O-RAN.*

1. Introdução

A evolução para os padrões 5G e 6G através da arquitetura Open Radio Access Network (Open RAN) desagrega estações rádio base em unidades modulares — Rádio Unit (RU), Distributed Unit (DU) e Centralized Unit (CU) [ALBERTI et al., 2021], o que amplia a superfície de ataque ao expor interfaces anteriormente proprietárias. Paralelamente, a ameaça de computadores quânticos e a tática Harvest Now, Decrypt Later tornam obsoletos algoritmos de chave pública atuais, como Rivest-Shamir-Adleman (RSA) e Elliptic Curve Cryptography (ECC), exigindo a adoção imediata da Criptografia Pós-Quântica (PQC) [RATHI et al., 2025].

O algoritmo ML-KEM (Kyber) [NIST, 2024] utiliza funções SHA-3 cujo processamento em software excede os limites de latência do Fronthaul O-RAN (100 a 250 µs). Para preencher essa lacuna, propomos uma Prova de Conceito (PoC) via co-design. O controle lógico opera no processador (ARM Cortex-M0+), enquanto a permutação pesada (Keccak) sofre offloading para uma Field Programmable Gate Array (FPGA). A plataforma Arduino MKR Vidor 4000 valida o PQC em sistemas de borda, evidenciando que o desafio primário não é a computação bruta, mas a interconexão de dados em tempo real.

2. Fundamentação Teórica

A arquitetura Open RAN desagrega funções em RU, DU e CU [ALBERTI et al., 2021], expandindo a superfície de ataque. Embora o framework Q-RAN (Quantum-Resilient O-RAN) busque resiliência quântica [RATHI et al., 2025], integrar algoritmos PQC sob restrições de tempo real no Fronthaul O-RAN é um desafio em aberto, tornando mandatória a inserção do algoritmo ML-KEM nesses pontos críticos.

2.1. Padronização ML-KEM (Kyber)

O padrão ML-KEM baseia-se em reticulados [NIST, 2024] e sua execução é intensiva no cálculo do SHA-3/SHAKE, que utiliza a permutação Keccak. Em software, essa permutação torna-se o gargalo primário, impedindo o determinismo temporal exigido por redes de ultrabaixa latência.

2.2. Paradigma de Hardware/Software Co-Design e Offloading Seletivo

Para solucionar esse gargalo, adota-se o co-design: a CPU gerencia o controle lógico, enquanto a FPGA atua como um acelerador dedicado (IP Core — Intellectual Property Core) para o paralelismo massivo das permutações. O sucesso dessa arquitetura híbrida depende não apenas da eficiência lógica, mas estritamente da interface física de interconexão (como JTAG (Joint Test Action Group) ou SPI(Serial Peripheral Interface)), conforme ilustrado na Figura 1.

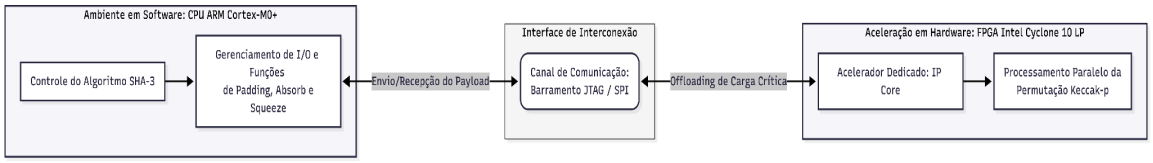

Figura 1. Arquitetura conceitual de co-design dividindo o controle lógico em software e a aceleração em hardware.

3. Metodologia

Para validar a interconexão via JTAG , utilizou-se um payload de 32 bits, embora a IP Core suporte 256 bits. Esta abordagem assume, como hipótese de trabalho, comportamento aproximadamente linear da latência de comunicação para o aumento do payload, sob a premissa de um regime dominado pela transferência de dados. Neste trabalho, o termo "testes físicos" refere-se à validação experimental executada no hardware real da placa (Arduino MKR Vidor 4000), aferindo latências via temporizadores nativos do microcontrolador, em contraposição a simulações teóricas em software.

3.1. Teste 1: Micro-benchmark do Gargalo (SHA-3 vs. Keccak)

Esta etapa foca no isolamento algorítmico, identificando onde o processamento consome mais tempo:

1. **Cenário CPU:** A função SHA-3 completa (incluindo as rotinas de padding, absorb e squeeze) é processada integralmente em software no núcleo ARM Cortex-M0+ (operando a 48 MHz) com o objetivo de estabelecer o baseline limitante de execução pura.
2. **Cenário FPGA (Validação e Calibração):** O núcleo keccak.v, responsável exclusivamente pela permutação matemática interna do SHA-3, foi sintetizado na FPGA Intel Cyclone 10 LP. Durante a execução física, a análise empírica evidenciou que a interface Virtual JTAG atua fundamentalmente como instrumento de depuração lógica. Isso permitiu atestar a integridade funcional da permutação em hardware de forma isolada, identificando a origem do gargalo de transporte.

3.2. Teste 2: Validação de Desempenho da IP Core

Para atestar a eficácia da PoC, o desempenho foi segmentado para separar a eficiência lógica do overhead de interconexão, considerando o modelo: *T_total = T_setup + T_transfer + T_sync + T_processing*.

1. **Limite Teórico Inferior (Latência Computacional):** A arquitetura do núcleo funcional keccak.v foi estruturada em Verilog com o auxílio do assistente de IA Gemini para prototipagem rápida. A validade técnica deste artefato específico foi confirmada por meio de simulação RTL e timing analysis, ferramentas utilizadas para isolar e estabelecer a latência estrita da lógica digital antes da integração.
2. **Latência Transporte (Sistêmica):**A validação física monitorou o canal de comunicação CPU-FPGA para aferir o tempo de resposta total, buscando isolar a métrica que reflete majoritariamente as componentes *T_setup, T_transfer* e *T_sync*.

3.3. Teste 3: Co-Design de Hardware e Software (Sistema Híbrido)

O estágio final consolida a arquitetura focada no gargalo do ML-KEM, integrando o acelerador físico enquanto a lógica de controle opera em *software*.

1. **Fluxo de Execução:** O ARM Cortex-M0+ gerencia a estrutura do SHA-3 e o offloading da permutação para a FPGA. A lógica de controle em C foi estruturada com suporte de IA (Gemini) e teve sua integridade validada pelos autores via testes físicos.
2. **Análise de Viabilidade Arquitetural:** O ciclo híbrido completo foi avaliado para extrair a métrica de speedup em relação à execução pura em software. Essa etapa analítica visa validar a eficácia do co-design e diagnosticar empiricamente a penalidade imposta pela interface de interconexão (JTAG), estabelecendo a fundamentação técnica para a transição futura rumo à interface SPI nativa.

4. Resultados e Discussão

A Tabela 1 apresenta a análise comparativa entre o *baseline* em software e a arquitetura de *offloading* proposta, sumarizando os ganhos de latência e o *speedup* obtidos em cada cenário. Os valores de SPI na Tabela 1 consideram apenas a latência de transferência, sendo a latência total discutida na Seção 5. A comparação entre cenários distingue latência de transferência e latência sistêmica total, conforme modelo apresentado.

Tabela 1. Comparação de latência entre CPU e FPGA.

Nível de Teste	Plataforma	Latência da Lógica (*μs*)	Latência de Transporte (I/O - *μs*)	Speedup	Requisito O-RAN
SHA-3 (Baseline)	Software (ARM Cortex-M0+)	1375	0	N/A	Não atende

IP Core (Limite Teórico)	**Hardware(Simulação RTL)**	**0,013**	**0**	**N/A**	**Atende (Computacional)**
SHA3 (Híbrido)	CPU + FPGA + JTAG	0,013	~390	3,52x	Gargalo I/O
SHA3 (Híbrido)	**CPU + FPGA + SPI**	**0,013**	**~64**	**21,4x**	**Estimado: Atende**

4.1. Análise do Desempenho e Latência

A avaliação estabeleceu um baseline de 1375 µs em software, contrastando com o limite teórico inferior de 0,013 µs (13,17 ns) alcançado pela lógica paralela na FPGA. A síntese do acelerador exigiu 7.210 Elementos Lógicos (47% da capacidade da Cyclone 10 LP) e 6.729 registradores, operando com a Fmax de 75,90 MHz (vazão de 19,4 Gbps) e dissipando 77,50 mW (4,37 mW dinâmicos). Conforme destacado em diretrizes recentes da ENISA e na literatura de transição PQC para 5G, o offloading para hardware dedicado é essencial para não comprometer a energia e a vazão da rede. Embora a Fmax obtida seja inferior a implementações puras de alto rendimento, ela reflete o compromisso arquitetural ideal: atende às restrições severas de área de sistemas embarcados de borda e, simultaneamente, entrega um expressivo speedup lógico. Contudo, a validação física via JTAG registrou um ciclo de ~390 µs. Isso comprova que a eficiência intrínseca do acelerador é mascarada pelo gargalo de transporte (I/O). Esse diagnóstico valida a arquitetura de co-design ao isolar as latências e fundamenta a adoção de um barramento de alto desempenho.

5. Conclusão e Trabalhos Futuros

O co-design provou ser promissor para adequar o ML-KEM às restrições da Open RAN. Os testes confirmam que o gargalo sistêmico é a interconexão (JTAG: 390 µs), não a computação do Keccak em hardware (apenas 0,013 µs). Mapeado esse overhead, os trabalhos futuros focarão na migração para um barramento SPI (estimado em ~64 µs de transferência) e na avaliação de payloads maiores para validar a linearidade do canal de comunicação, visando o pleno cumprimento do rigoroso limite de 250 µs do 5G.

6. Referências

ALBERTI, A. M. et al. (2021) "OpenRAN: A Conexão do Futuro", Inatel.

RATHI, V. et al. (2025) "Q-RAN: Quantum-Resilient O-RAN Architecture", coRAN Labs.

NIST (2024) "FIPS 203: Module-Lattice-Based Key-Encapsulation Mechanism Standard", National Institute of Standards and Technology.

ENISA (2021) "Post-Quantum Cryptography", European Union Agency for Cybersecurity.