Electronic Design

Dr. ANDREA CAFORIO, Senior Engineer, and DR. PASCAL NASAHL, Lead, Silicon Security Team, lowRISC, https://lowrisc.org/

Building an Efficient Side-Channel-Resilient Post-Quantum Root-of-Trust Design

This article dives deep into the engineering tradeoffs required to achieve side-channel-resilient PQC implementations on modern root-of-trust devices that must meet high-assurance security certifications.

The transition to post-quantum cryptography (PQC) is becoming an increasingly practical concern rather than a purely theoretical one. If large-scale quantum computers become viable, they could break the mathematical problems that current public-key systems such as RSA and elliptic curve cryptography rely on. As a result, many of today's digital signature and key exchange mechanisms would no longer be secure.

This challenge is different from the situation with symmetric encryption like AES, where security can generally be preserved by increasing key sizes. In contrast, existing public-key schemes need to be replaced with entirely new quantum-resistant algorithms.

The need for this transition has become more concrete following the standardization of several post-quantum algorithms by the National Institute of Standards and Technology (NIST) in August 2024, along with policy initiatives in multiple countries encouraging migration to PQC within the coming decade.[1,2,3] For semiconductor providers, this means that planning for quantum-resistant cryptographic support in future systems is already necessary.

Why SCA Hardening is a Must for RoTs Implementing PQC

For root-of-trust (RoT) devices such as OpenTitan,[4] this planning must also include deliberate physical security hardening. These systems typically target high-assurance certifications like Common Criteria (CC), where the evaluation assumes an attacker with *High* attack potential.

In that context, functional correctness alone isn't sufficient. An implementation that simply produces the correct mathematical result may still be vulnerable if it leaks information through physical side channels, such as power consumption or electromagnetic emanation. The cryptographic algorithms therefore need to be implemented with explicit protections against side-channel analysis (SCA).

Without these countermeasures, an attacker may be able to recover private key material by observing power consumption or electromagnetic emissions during operations such as signature generation. This risk exists regardless of how strong or "quantum-resistant" the underlying cryptographic primitive may be. Hence, protecting the PQC implementation with countermeasures is a must.

To deter SCA attacks, implementations typically rely on masking countermeasures. In a first-order masked design, sensitive values are split into two random shares so that no single intermediate value in the data path is correlated with the underlying secret.

When implemented correctly, it prevents first-order information leakage through power or EM measurements. That is, the statistical mean of such side-channel measurements will not be correlated with the actual secrets. In addition, higher-order leakage, such as the variance, typically disappears in the measurement noise.

The Masking Penalty: Performance and Memory Bottlenecks

Masking is a robust defense for securing the execution of a cryptographic algorithm against SCA attacks. At its core,

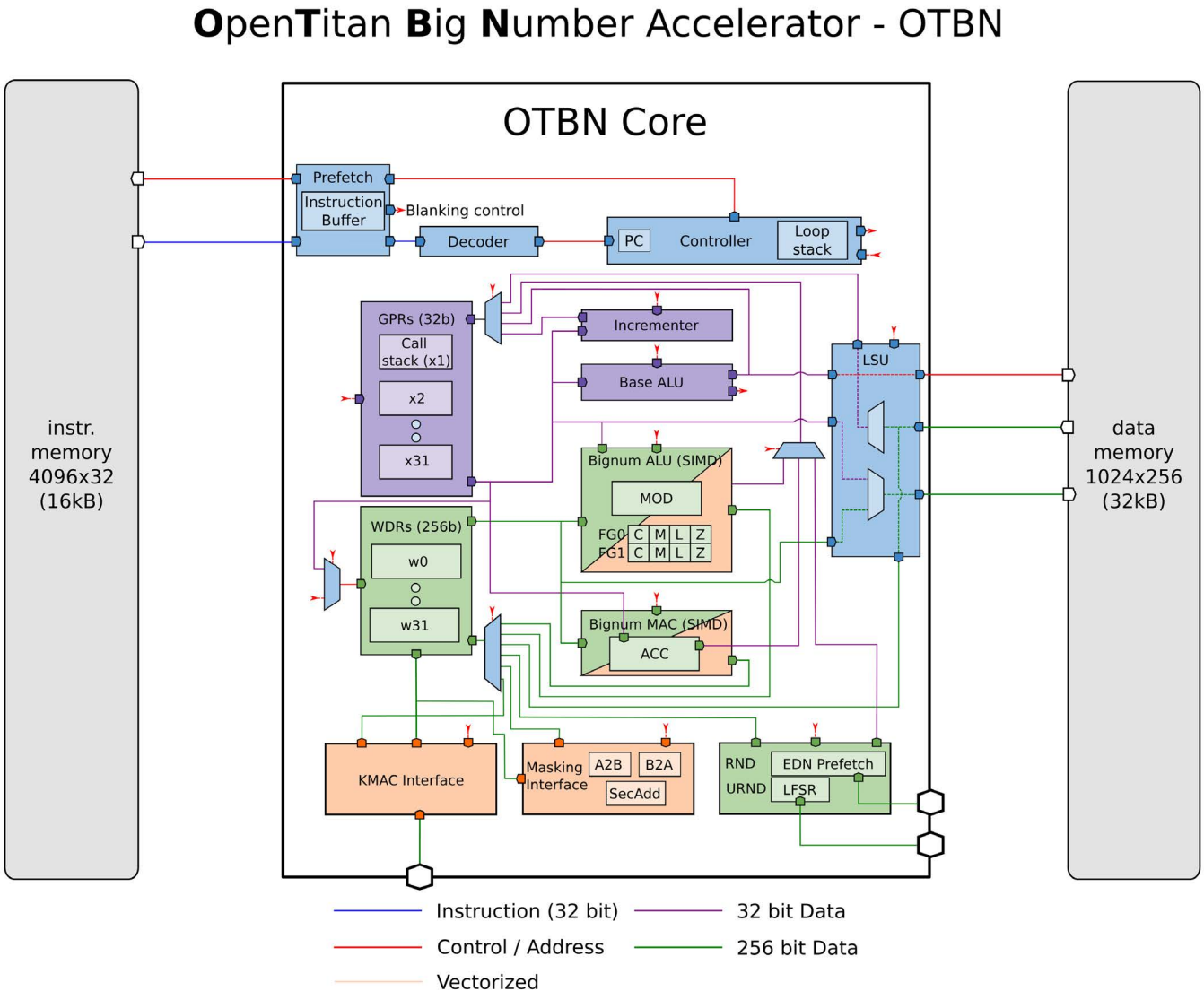
masking relies on the principle of randomly decomposed computations to remove the statistical dependency between the intermediate secret values and the device's power consumption or electromagnetic emission. The decomposed computations can then be performed in different ways:

- Hardware implementations typically perform the computations in parallel using spatially redundant circuits to enable high-speed execution.
- Software implementations typically time-multiplex the computations on the same hardware data paths to reduce the area overhead.

While hardware often provides better performance, software provides the critical flexibility needed for PQC hardening. Because PQC masking is still a highly active area of research with evolving insights, the ability to iterate and update implementations in software is a vital advantage, even if it necessitates navigating significant performance tradeoffs.

Our own investigations in collaboration with renowned researchers revealed[5] that a masked software implementation of ML-DSA-87 can be more than an order of magnitude slower than an unprotected implementation. A large portion of this overhead comes from mask domain conversions.

ML-DSA mixes primitives that are most naturally implemented with different masking schemes: Boolean (e.g., SHA3-based hash operations) and arithmetic (e.g., lat-



Schematic of the OpenTitan Big Number Accelerator. Objects colored in orange indicate the PQC accelerators, i.e., the SIMD extension as part of the bignum and MAC ALUS, the KMAC interface and the masking interface composed of the A2B/B2A converters, and a secure adder.

tice crypto). Converting between these domains are called Arithmetic to Boolean (A2B) and Boolean to Arithmetic (A2B) conversions. In practice, these conversions become a dominant performance bottleneck.

Because ML-DSA is envisioned to be used during secure boot, this performance hit is particularly critical. Every extra cycle spent on cryptographic operations directly extends the device's boot time, delaying the availability of the entire system.

Masking also increases the memory footprint. Each sensitive value must be represented as shares, which effectively multiplies the storage requirements for large vectors and polynomials. For many RoT microarchitectures, this pushes memory requirements beyond typical memory configurations.

Hardware-Software Co-Design: The OpenTitan PQC Implementation

The following paragraphs highlight the hardware/software co-design approach that the OpenTitan project has engineered to provide a fully SCA hardened ML-DSA implementation. A schematic summary of this section in the form of an architectural diagram is shown in the *figure*.

As hinted at above, securing cryptographic algorithms against side-channel attacks is often synonymous with significant performance decreases. The decomposition of sensitive variables into multiple independent shares requires that the functions operating on these variables be decomposed accordingly.

Such subfunctions are more complex than their unshared parent from which they derive and come with specific requirements on the composition of the underlying circuits in terms of gates and randomness. As a result, the decomposition of even the simplest functions such as a 2-bit AND gate can result in a circuit that's 10X to 20 X larger and requires multiple cycles to compute its output.

This overhead is further amplified if one chooses to implement these shared functions purely in software, where the penalty in terms of code size and running can be prohibitive, especially on resource-constrained devices.

To remedy the overhead of a shared/masked PQC implementation on its performance metrics, we identified the most salient functions that form the basis of a shared lattice-based cryptography and offloaded their computation to a set of dedicated accelerators with the OpenTitan Big Number Accelerator (OTBN).

That set contains a shared 32-bit adder and both an A2B and B2A converter. All three circuits are vectorized and can operate multiple 32-bit words in parallel to amortize their multicycle nature. A secure shared adder is in fact the fundamental building block of the A2B and B2A converters. There are multiple well-established techniques on how to bootstrap these converters in a secure manner from a single secure adder.

This architectural choice reflects a strategic balance between performance and flexibility:

- **Hardware for the known:** We have dedicated hardware to handle mask conversion — an operation that's both computationally costly and theoretically well-understood.
- **Software for the evolving:** By keeping the high-level SCA hardening of ML-DSA in software, we retain the flexibility to adapt to new research. Since side-channel protection for lattice-based schemes is a relatively nascent field, this allows us to update our countermeasures without requiring a full silicon redesign.

The inclusion of these accelerators into the OpenTitan fold is indicative of a tradeoff. By increasing the circuit footprint by a reasonable amount (these three mask-conversion accelerators are small compared to the overall size of the OpenTitan SoC), we're able, according to preliminary measurements, to bound the performance overhead of a fully masked ML-DSA implementation to the 2X to 4X range. This makes it feasible to use the algorithm in performance-critical applications such as secure boot.

Moreover, the accelerators allow us to significantly reduce the code size of our hardened PQC implementations, which are now only insignificantly larger than their unhardened counterparts.

Vectorized Arithmetic

The hardening of sensitive functions in PQC algorithms doesn't prevent more general optimizations; they can even benefit from each other through well-engineered composition. The presence of vectorized A2B/B2A converters is extended to the vectorization of arithmetic operations such as addition, subtraction, and multiplication, whereby computation between conversions can proceed seamlessly without the need to ever rearrange data in any way.

Given that modular arithmetic is the basis of all computation in PQC schemes, having them vectorized in a SIMD fashion (provided as an OTBN instruction set extension) further softens the performance impact of the SCA countermeasures. Since the OTBN already contains a rich set of various adders, subtractors, and multipliers, their vectorization only induces a moderate circuit overhead and in turn makes it possible to save code size, i.e., memory area.

The efficiency gains obtained through the integration of both the mask-conversion accelerator and the SIMD instruction set extension as part of the OTBN are ultimately futile, though, if the ML-DSA and ML-KEM implementations have no performant way of obtaining large amounts of randomness from a hash function to feed into their sampling routines.

For example, the various sampling routines in ML-DSA account for more than half of the running time. This translates to many tens of thousands of bytes that need to be squeezed out of a hash function for the computation of a single signature.

OpenTitan already contains a hardened KMAC module that instantiates a set of SHA3-adjacent algorithms that are required in both ML-DSA and ML-KEM, which is accessible by the host CPU but doesn't interface with the OTBN. The implementation of this KMAC-OTBN interface is the last cornerstone of our OpenTitan PQC suite.

It's important to note that effectiveness of the mask-conversion accelerators for SCA hardening, SIMD arithmetic, and the KMAC interface is closely tied to the semantics of the actual standardized specification of the PQC algorithms and their interpretation. Intermediate variables can be shuffled, precomputed, or generated on-the-fly to save on data memory, which in turn can have a dramatic impact on the running time of the algorithm.

In our PQC implementation, we made a diligent effort to sensibly implement the specifications and find a middle ground that allowed us to both capitalize on the aforementioned OTBN additions while keeping the memory footprint reasonable.

In summary, the OpenTitan PQC suite introduces the following additions:

1. OTBN mask-conversion accelerators, i.e., secure adder, A2B/B2A converters.
2. OTBN SIMD instruction set extension for vectorized arithmetic.
3. KMAC-OTBN interface to efficiently generate large amounts of randomness.
4. Implementation of the standardized algorithmic specifications in a way to best leverage points 1 to 3 to minimize the data memory footprint while still meeting performance requirements.

Dr. Andrea Caforio is a Senior Engineer in the Silicon Security Team at lowRISC C.I.C., specializing in cryptographic implementations and analysis. He holds a PhD from the Ecole Polytechnic Fédérale de Lausanne (EPFL), where his research focused on cryptanalysis and optimization of cryptographic primitives.

Dr. Pascal Nasahl is the Silicon Security Team Lead at lowRISC C.I.C., specializing in fault injection and side-channel analysis and hardening. He holds a PhD from Graz University of Technology, where his research focused on fault injection countermeasures and system security. His professional background includes various hardware security roles within the semiconductor and security evaluation industries.

References

1. https://csrc.nist.gov/pubs/ir/8547/ipd
2. https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Crypto/PQC-joint-statement-2025.pdf
3. https://www.ncsc.gov.uk/news/pqc-migration-roadmap-unveiled
4. https://opentitan.org/
5. https://www.research-collection.ethz.ch/entities/publication/d573d76d-9cae-48d3-b149-5bddd86a14cf