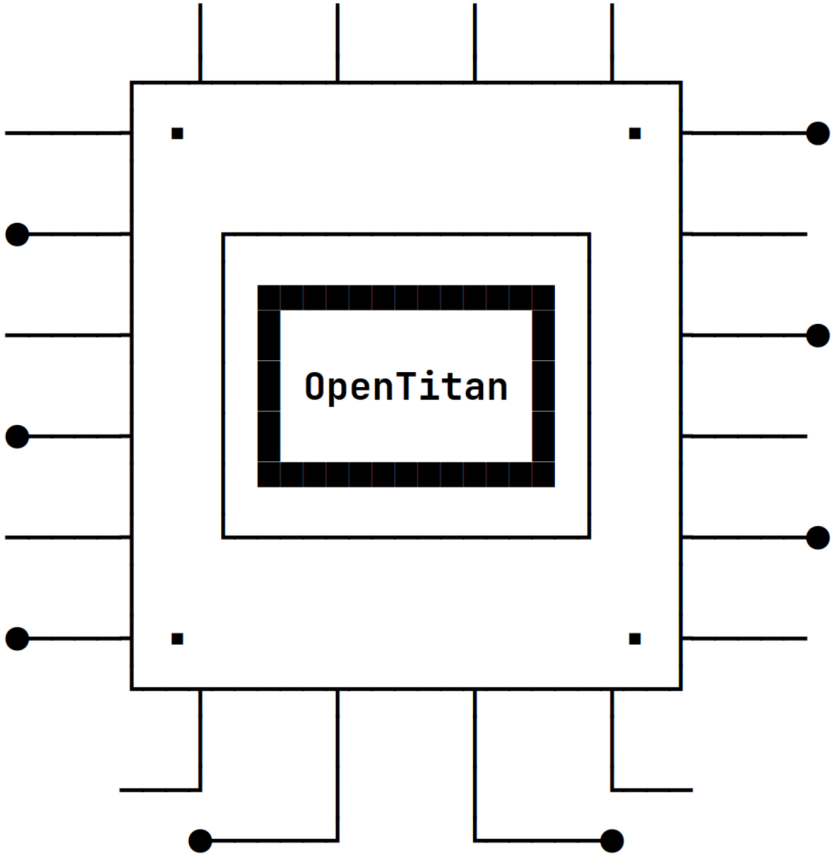
Assembling Lattices: Secure and Efficient PQC in OpenTitan®

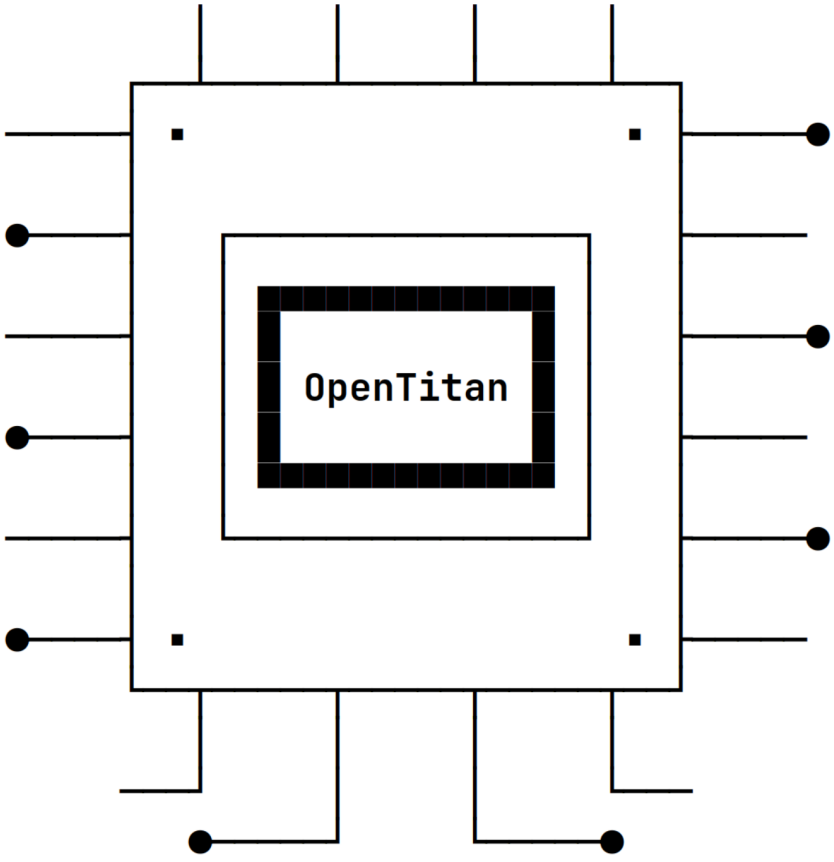
Andrea Caforio Pascal Etterli
Hakim Filali Pascal Nasahl
Pirmin Vogel

Swiss Crypto Day 2026
Zurich, 4 September

lowRISC

opentitan





OpenTitan is the world's most active and widely adopted open-source silicon project.

500K lines of SystemVerilog, 300K lines of C and Rust, 300 contributors.

Root-of-trust / attestation solution, i.e., it is a chip for cryptographic computations.

Mass-produced and integrated in a variety of commercial devices (e.g., Google Chromebooks).

In the process of being made post-quantum-ready with a dedicated accelerator for Lattice-based cryptography.

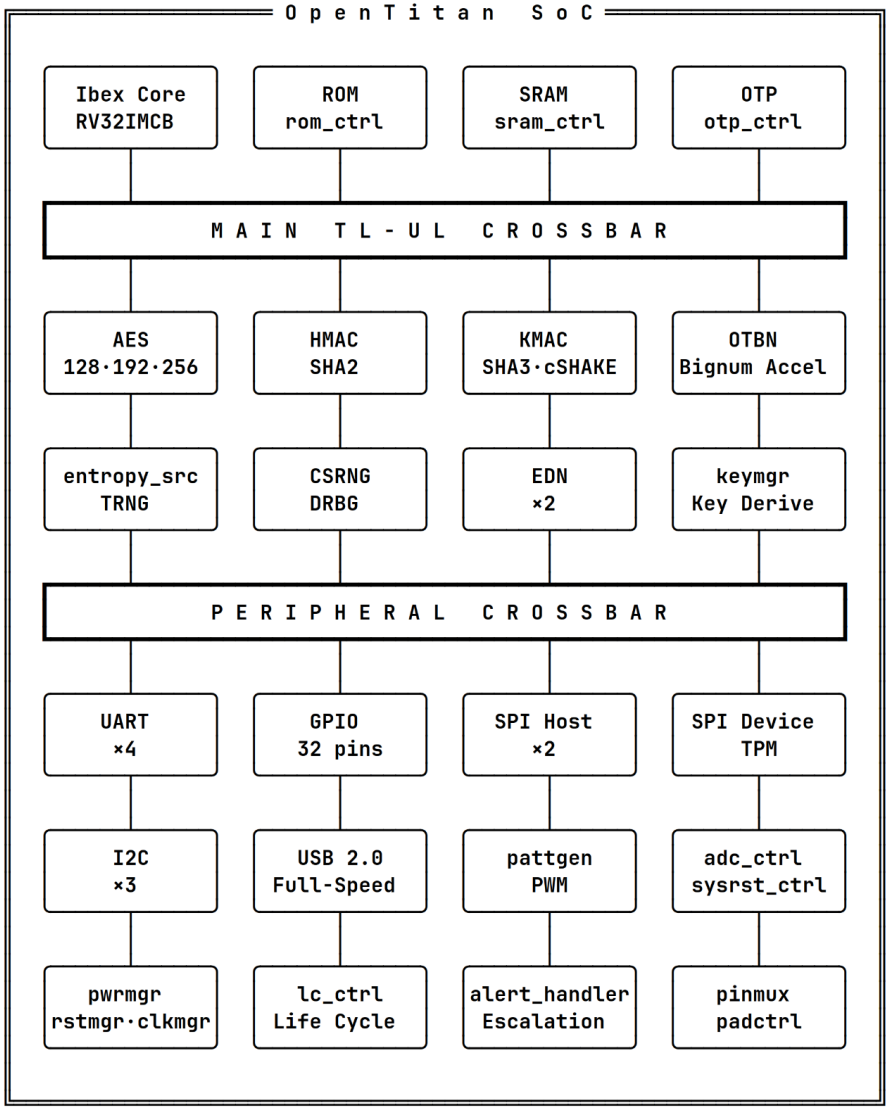
OpenTitan is the world's most active and widely adopted open-source silicon project.

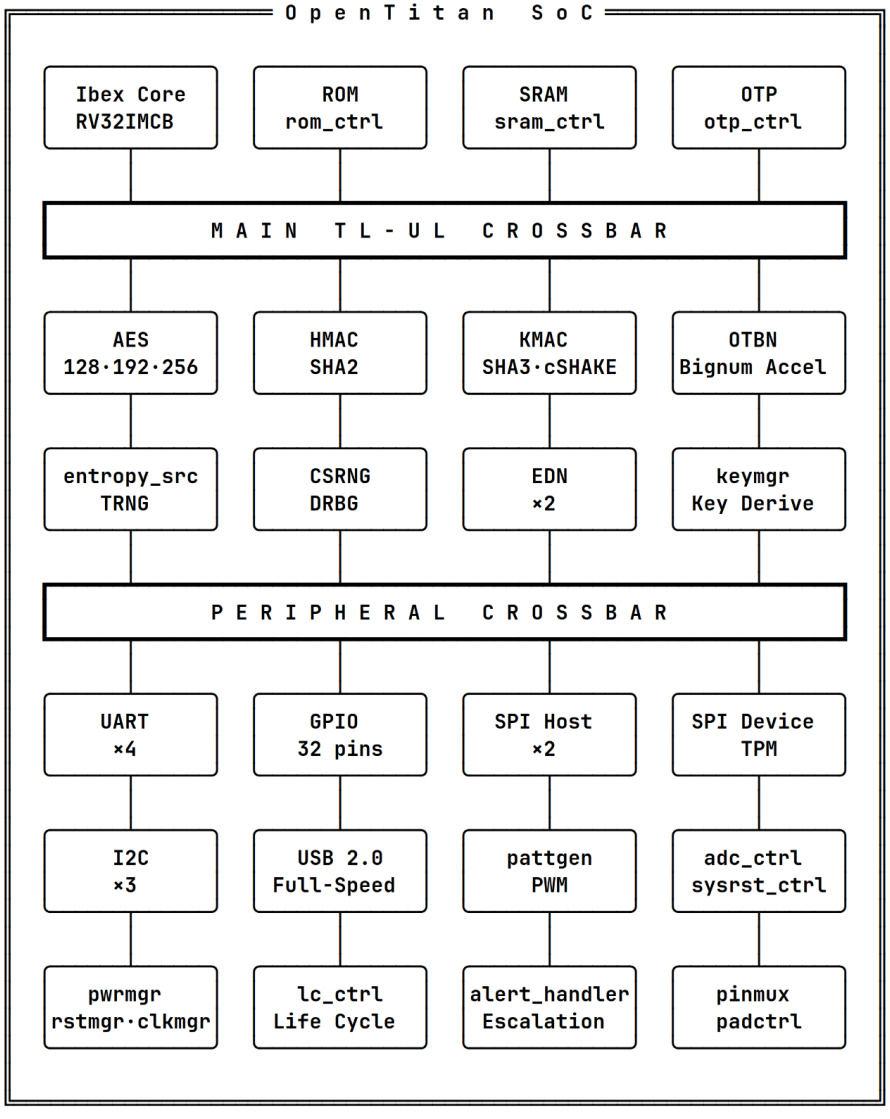
500K lines of SystemVerilog, 300K lines of C and Rust, 300 contributors.

Root-of-trust / attestation solution, i.e., it is a chip for cryptographic computations.

Mass-produced and integrated in a variety of commercial devices (e.g., Google Chromebooks).

In the process of being made post-quantum-ready with a dedicated accelerator for Lattice-based cryptography.





The OpenTitan SoC consists of a medley of hardware IPs.

Ibex: a 32-bit RISC-V CPU.

Hardware implementations of AES (GCM and multiple modes of operation), SHA2 (HMAC), SHA3 (KMAC).

Entropy source for true random number generation and a cryptographically secure PRNG.

OpenTitan Big Number accelerator (OTBN), a separate RISC-V-like CPU for the acceleration of asymmetric cryptography.

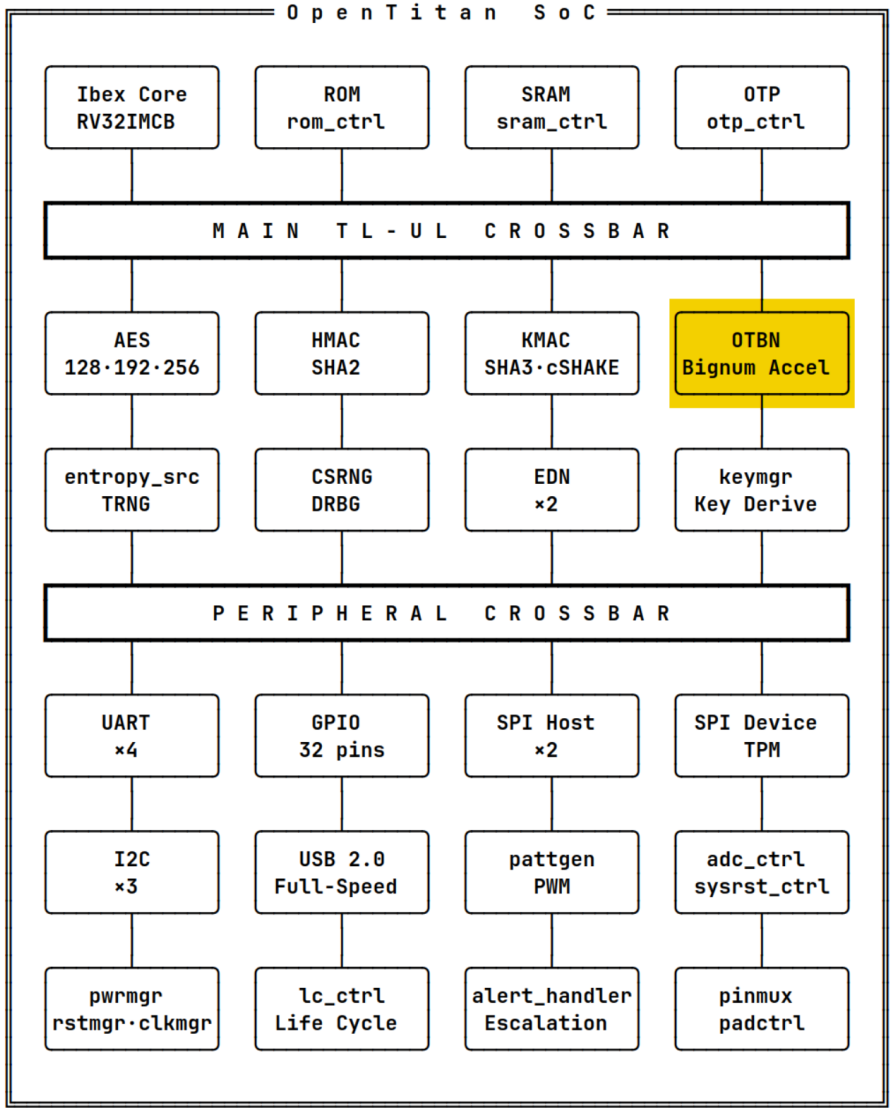
The OpenTitan SoC consists of a medley of hardware IPs.

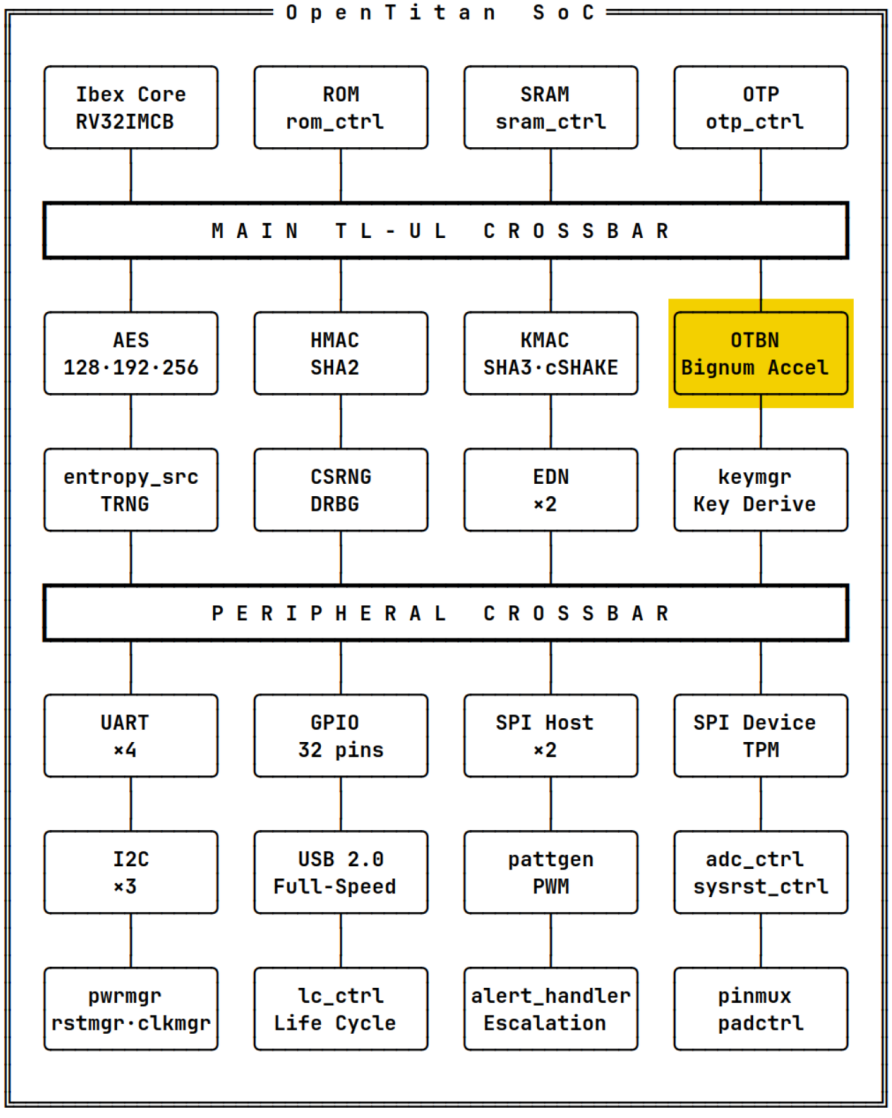
Ibex: a 32-bit RISC-V CPU.

Hardware implementations of AES (GCM and multiple modes of operation), SHA2 (HMAC), SHA3 (KMAC).

Entropy source for true random number generation and a cryptographically secure PRNG.

OpenTitan Big Number accelerator (OTBN), a separate RISC-V-like CPU for the acceleration of asymmetric cryptography.





The OTBN is a Harvard architecture (separate instruction and data memory) RISC-V-like CPU.

Ibex (main CPU) writes input data to the OTBN DMEM, loads an assembly program into the IMEM, triggers its execution then reads out results from the DMEM.

32-bit general-purpose instructions and 256-bit arithmetic operations (logic, addition, multiplication).

Designed for the acceleration of large integer arithmetic found in elliptic curves and RSA.

Not very useful for Lattice arithmetic (large polynomials of small integers).

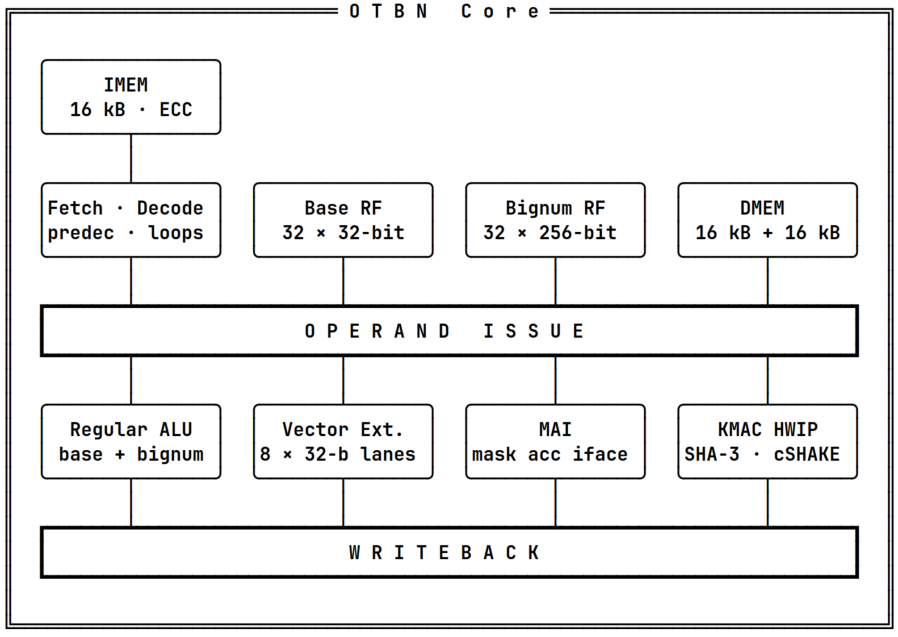
The OTBN is a Harvard architecture (separate instruction and data memory) RISC-V-like CPU.

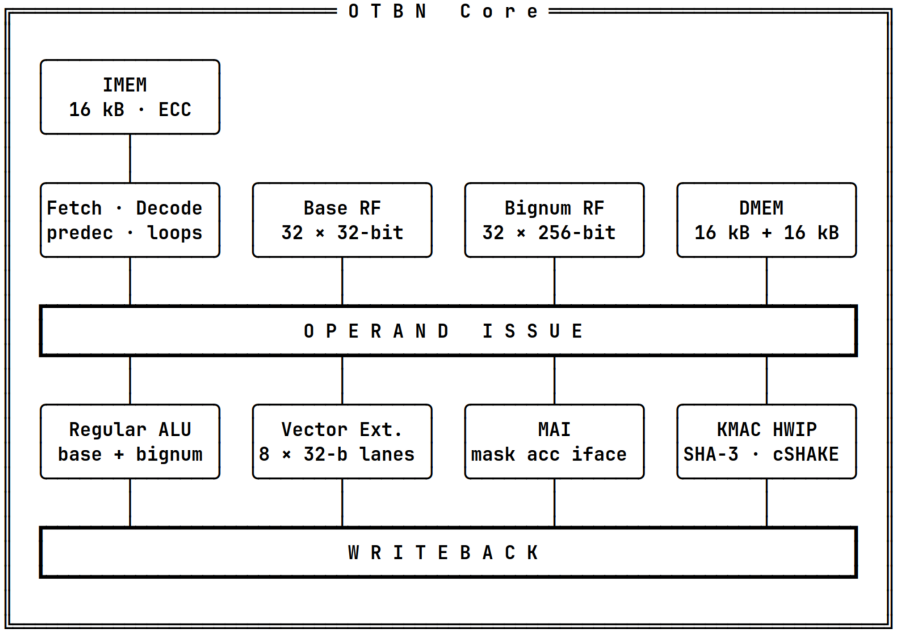
Ibex (main CPU) writes input data to the OTBN DMEM, loads an assembly program into the IMEM, triggers its execution then reads out results from the DMEM.

32-bit general-purpose instructions and 256-bit arithmetic operations (logic, addition, multiplication).

Designed for the acceleration of large integer arithmetic found in elliptic curves and RSA.

Not very useful for Lattice arithmetic (large polynomials of small integers).





The challenge: making the OTBN fit for Lattice-based cryptography.

The questions:

What are the sizes of the DMEM and IMEM?

What additions (instruction set extensions) to the 256-bit ALU are necessary?

How can we efficiently sample thousands of bytes from a hash function?

What accelerators for side-channel hardening are necessary?

The challenge: making the OTBN fit for Lattice-based cryptography.

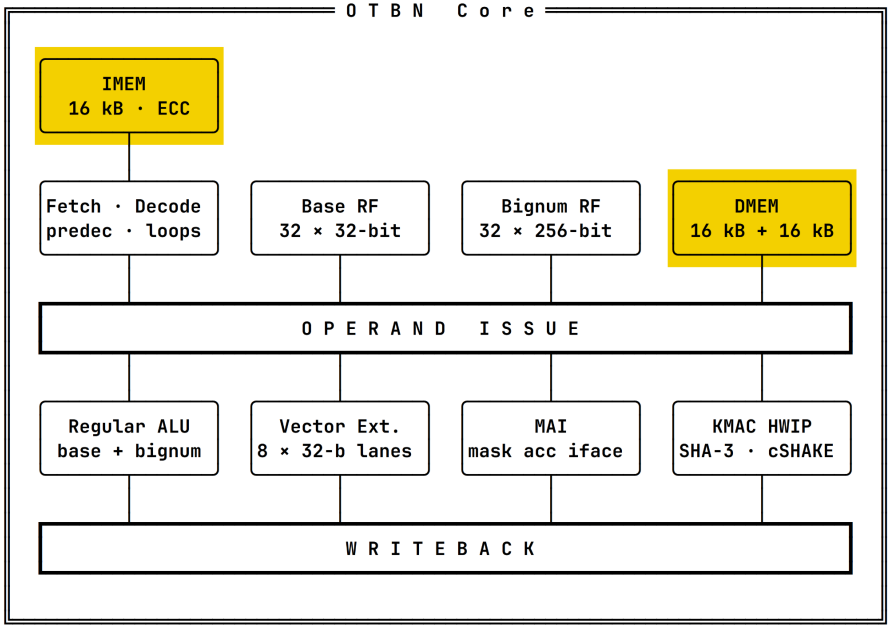
The questions:

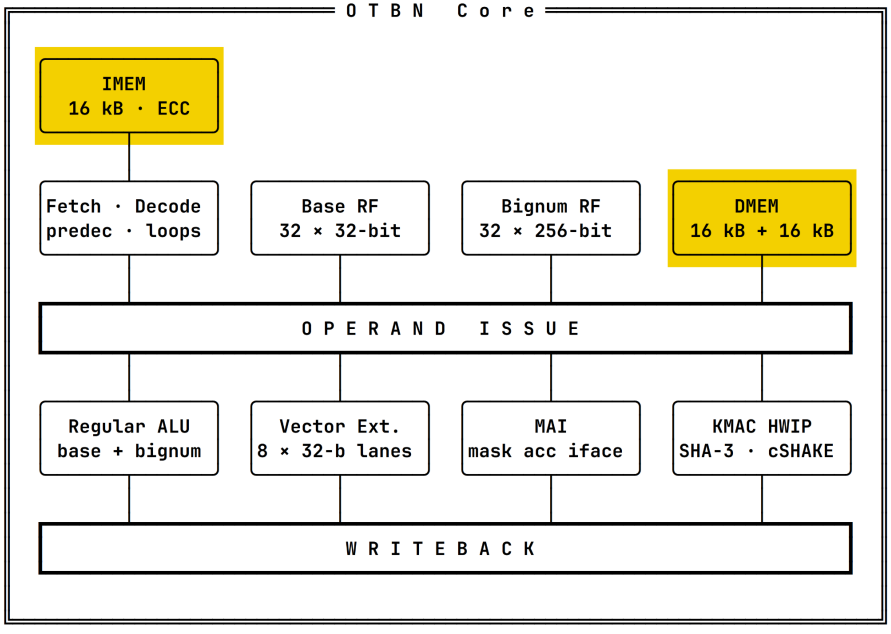
What are the sizes of the DMEM and IMEM?

What additions (instruction set extensions) to the 256-bit ALU are necessary?

How can we efficiently sample thousands of bytes from a hash function?

What accelerators for side-channel hardening are necessary?





Originally, the OTBN only had 4 KiB of DMEM and 8 KiB of IMEM (not enough for a ML-DSA secret key).

Memory requires a lot of silicon area and the area budget for OpenTitan is very small.

Optimize the ML-KEM and ML-DSA memory usage by aggressively streaming every operation, i.e., on-the-fly computation of intermediate results.

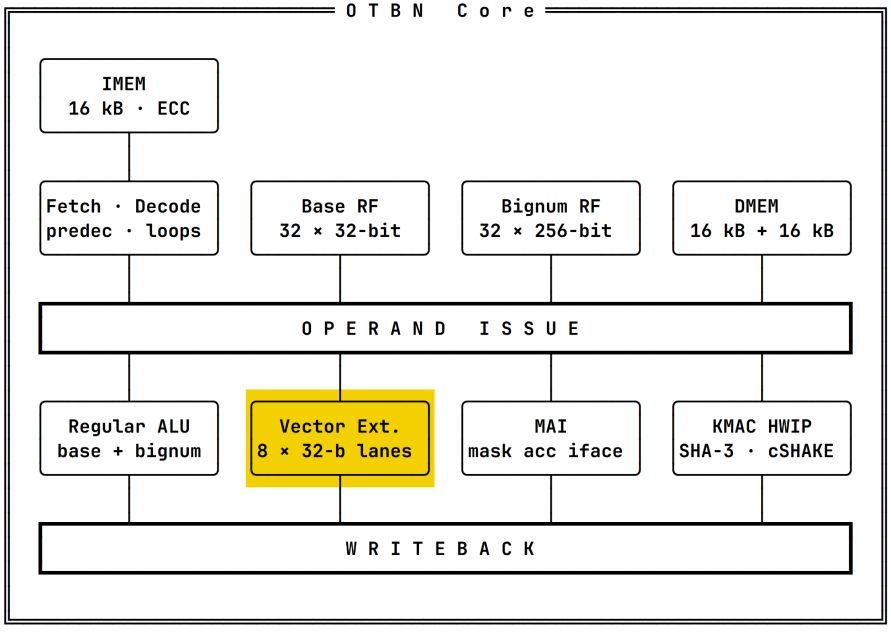
Through streaming and clever reordering of operations, we are able to implement ML-KEM (keygen, encap, decap) and ML-DSA (keygen, sign, verify) with only 32 KiB of DMEM and 16 KiB of IMEM.

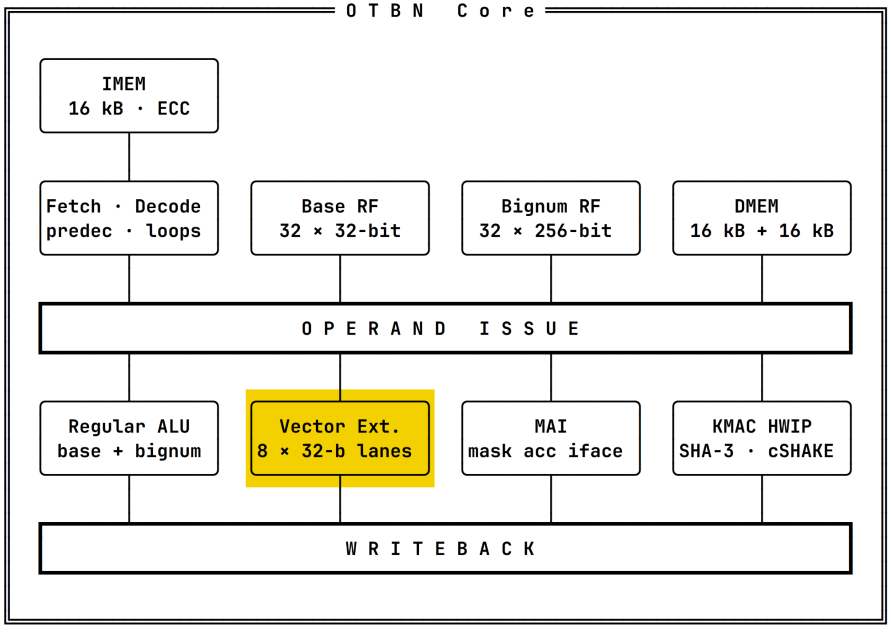
Originally, the OTBN only had 4 KiB of DMEM and 8 KiB of IMEM (not enough for a ML-DSA secret key).

Memory requires a lot of silicon area and the area budget for OpenTitan is very small.

Optimize the ML-KEM and ML-DSA memory usage by aggressively streaming every operation, i.e., on-the-fly computation of intermediate results.

Through streaming and clever reordering of operations, we are able to implement ML-KEM (keygen, encap, decap) and ML-DSA (keygen, sign, verify) with only 32 KiB of DMEM and 16 KiB of IMEM.





Our PQC implementations need to be fast enough (number of cycles) for latency-critical applications like secure boot.

Lattice arithmetic is embarrassingly parallel, therefore a perfect candidate for a SIMD-style vector extension.

We extended the OTBN ALU with a vectorized 8 x 32-bit Montgomery multiplier and modular adders to process multiple polynomial coefficients in parallel.

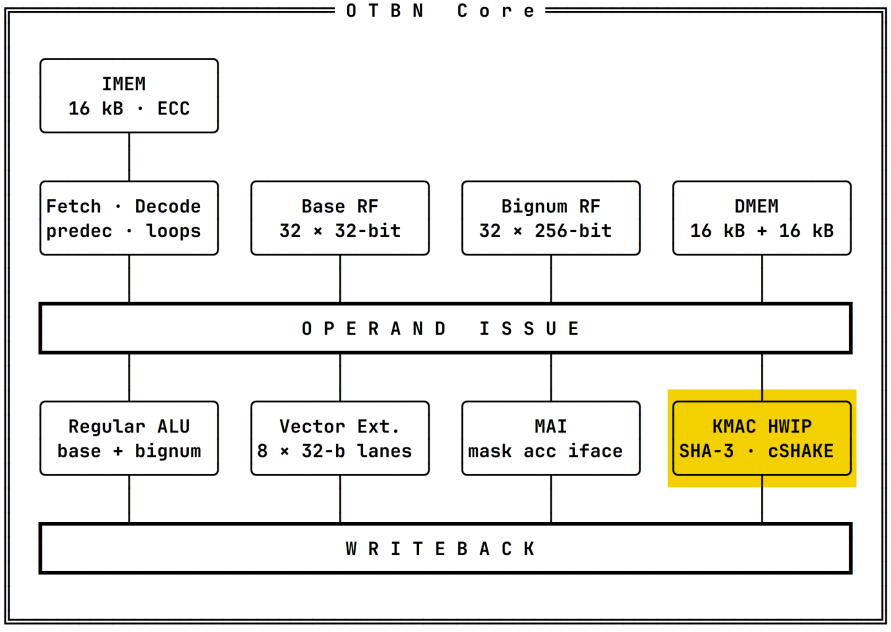
The result is extremely fast polynomial arithmetic. For instance, one NTT in 2000 clock cycles.

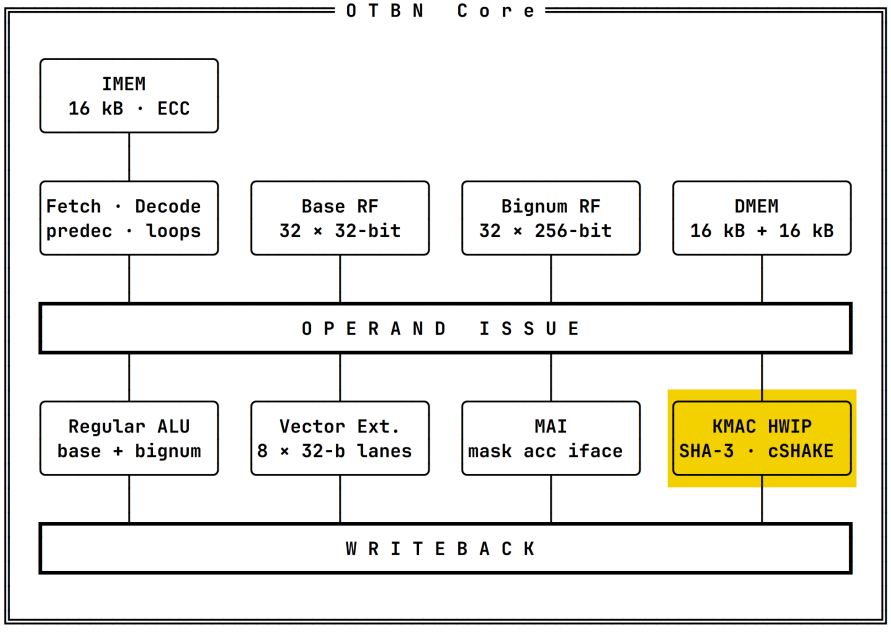
Our PQC implementations need to be fast enough (number of cycles) for latency-critical applications like secure boot.

Lattice arithmetic is embarrassingly parallel, therefore a perfect candidate for a SIMD-style vector extension.

We extended the OTBN ALU with a vectorized 8 x 32-bit Montgomery multiplier and modular adders to process multiple polynomial coefficients in parallel.

The result is extremely fast polynomial arithmetic. For instance, one NTT in 2000 clock cycles.





The bottleneck in ML-KEM and ML-DSA is not arithmetic but sampling.

Tens of thousands of bytes of data need to be sampled out of a XOF (SHAKE).

OpenTitan has a SHAKE hardware block that is accessible to the Ibex but not the OTBN.

So, we connected those two with a zero-overhead interface.

It is now possible to sample entire polynomials in a few thousand clock cycles depending on the sampling algorithm.

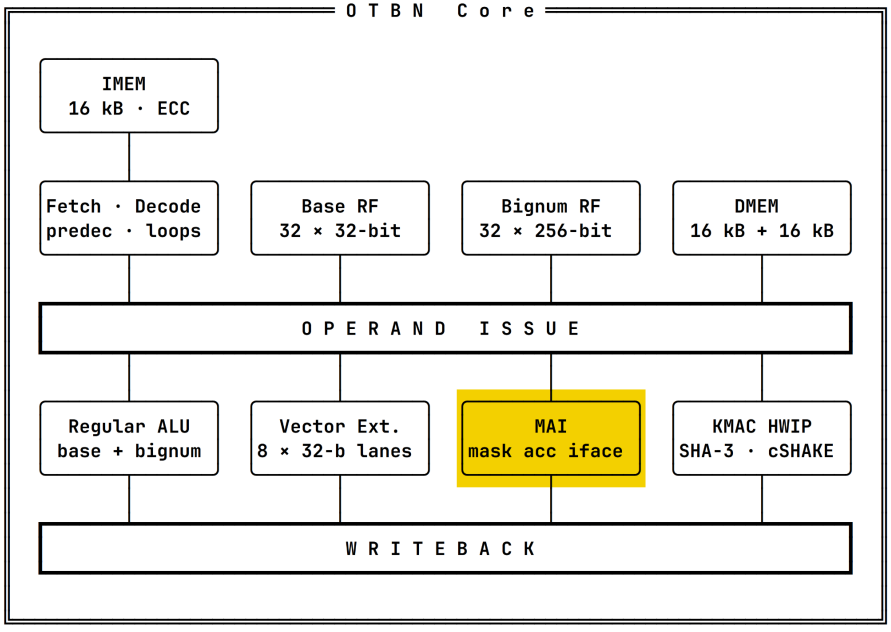
The bottleneck in ML-KEM and ML-DSA is not arithmetic but sampling.

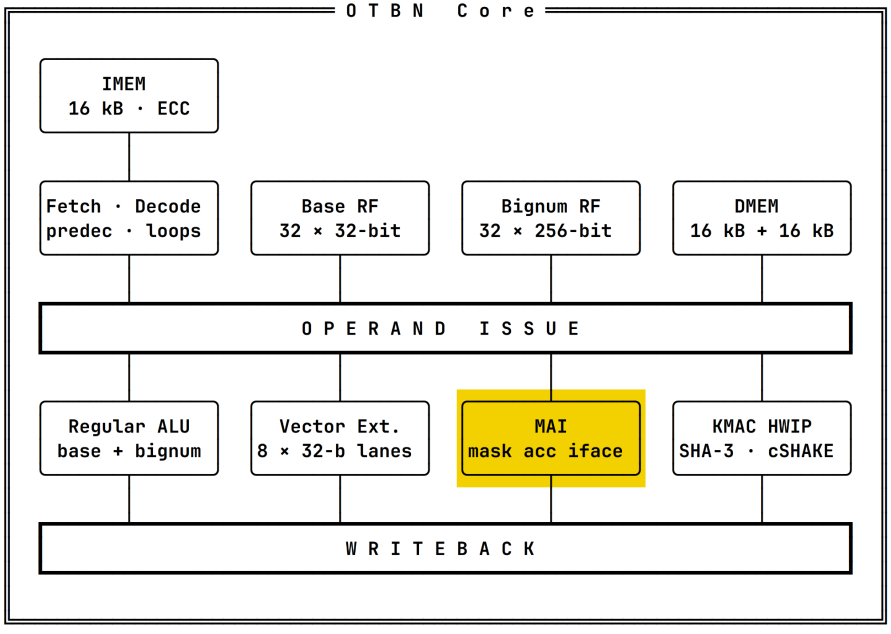
Tens of thousands of bytes of data need to be sampled out of a XOF (SHAKE).

OpenTitan has a SHAKE hardware block that is accessible to the Ibex but not the OTBN.

So, we connected those two with a zero-overhead interface.

It is now possible to sample entire polynomials in a few thousand clock cycles depending on the sampling algorithm.





Our ML-KEM and ML-DSA implementations are slated for FIPS-certification. This means they need to be secure against first-order side-channel attacks and fault-injection attacks with one precise fault.

We designed a dedicated hardware accelerator for the most prominent masking operations in Lattice cryptography.

Vectorized mask converter (arithmetic-boolean, boolean-arithmetic), vectorized secure adder.

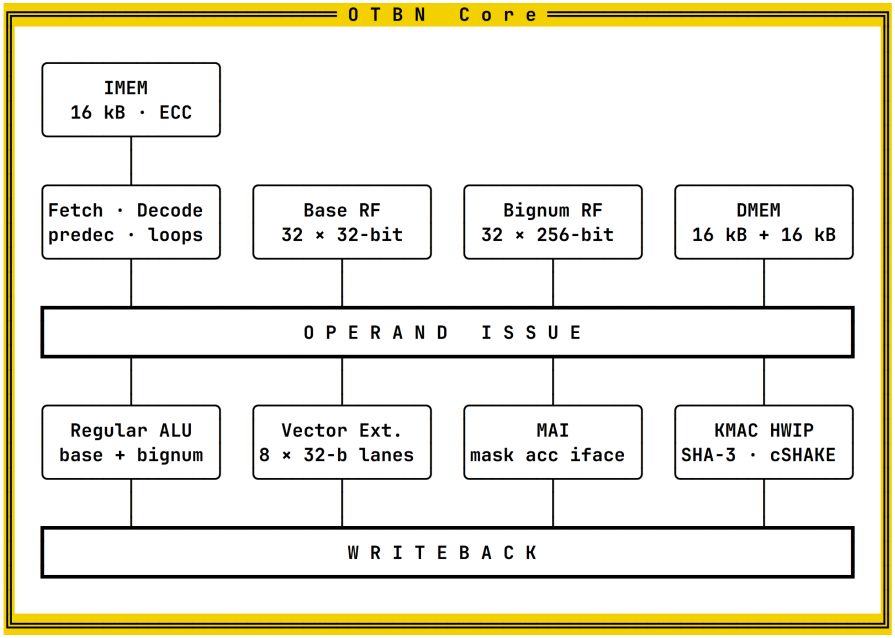
First experiments indicate complete first-order security for our masked ML-DSA implementation.

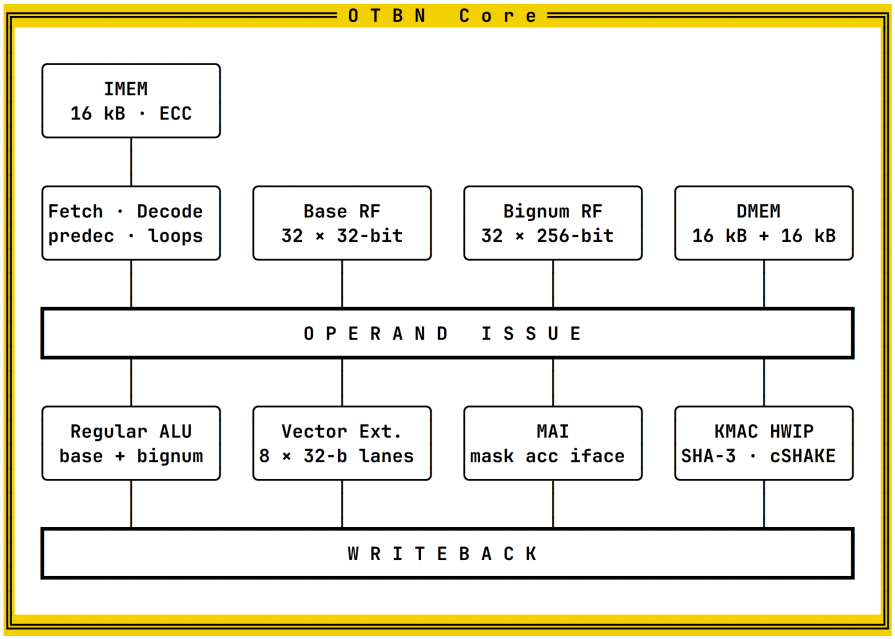
Our ML-KEM and ML-DSA implementations are slated for FIPS-certification. This means they need to be secure against first-order side-channel attacks and fault-injection attacks with one precise fault.

We designed a dedicated hardware accelerator for the most prominent masking operations in Lattice cryptography.

Vectorized mask converter (arithmetic-boolean, boolean-arithmetic), vectorized secure adder.

First experiments indicate complete first-order security for our masked ML-DSA implementation.





The OpenTitan PQC suite is one of the first open-source hardware-accelerated, masked and fault-injection-hardened implementation of ML-DSA with ML-KEM being in the works.

It can verify a ML-DSA signature in 300K clock cycles and it takes 800K cycles for one rejection loop in the signature generation function. Fast enough for latency-critical applications.

Aggressively memory-optimized.

Acceptable silicon area overhead.

The OpenTitan PQC suite is one of the first open-source hardware-accelerated, masked and fault-injection-hardened implementation of ML-DSA with ML-KEM being in the works.

It can verify a ML-DSA signature in 300K clock cycles and it takes 800K cycles for one rejection loop in the signature generation function. Fast enough for latency-critical applications.

Aggressively memory-optimized.

Acceptable silicon area overhead.

Get in touch.