

# The Silicon Commons Build Together, Build Well & Build Securely

Dr. Gavin Ferris, CEO @ lowRISC CIC



# Open ISA to Open Source Silicon



**winbond**



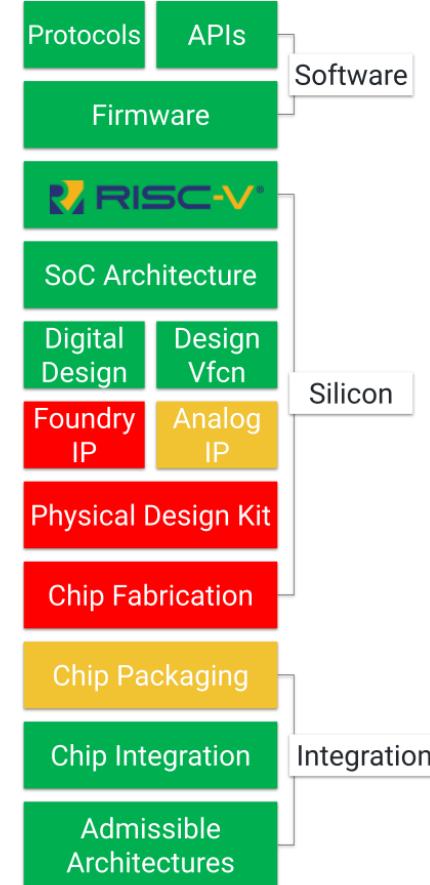
**nuvoton**

**ETH zürich**

Other Silicon



opentitan



Proprietary

Vendor

Open

# Commercial-Grade Open Source Silicon Is Here

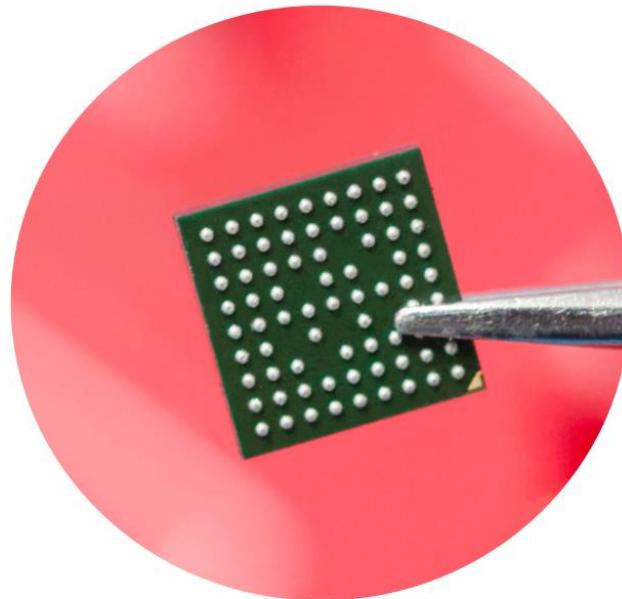
*"Nuvoton Technology Corporation [...] announced today that **Google's ChromeOS plans to use the first commercial chip built on the OpenTitan open source secure silicon design as an evolution of its security chip for Chromebooks.**"*

Nuvoton, May 2024

**nuvoton**

 opentitan

<https://www.nuvoton.com/news/news/all/TSNuvotonNews-000514>



***"Hardware security is something we don't compromise on. We are excited to partner with the dream team of Nuvoton, a valued, historic, strategic partner, and lowRISC, a leader in secure silicon, to maintain this high bar of quality."***

Prajakta Gudadhe  
Sr Director, ChromeOS Platform Engineering

 chromebook

# The Silicon Commons® Approach



Code review and approval process



Open development with clear IP provenance



Governance structure



Continuous Integration testing



Training for contributors



Accessible verification collateral



Extensive documentation



Permissive licensing to encourage re-use

# World's Most Active Open Silicon Project

RTL • design verification collateral • documentation • low-level firmware • tests

25,000+  
total commits  
(Ibex + OpenTitan)

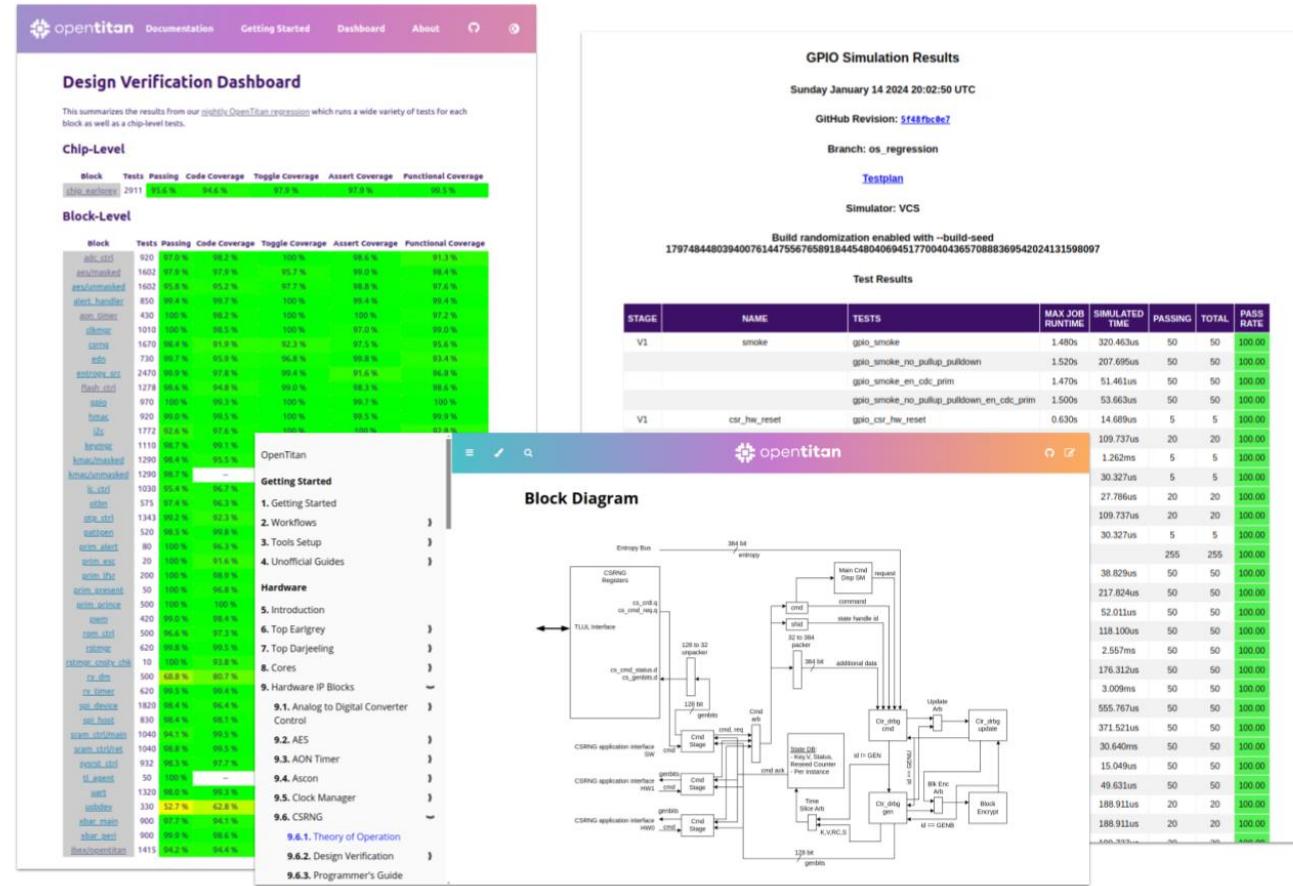
7,200+  
GitHub issues  
(Ibex + OpenTitan)

440,000+  
lines of SystemVerilog  
(Digital Design and Verification for  
Ibex + OpenTitan)

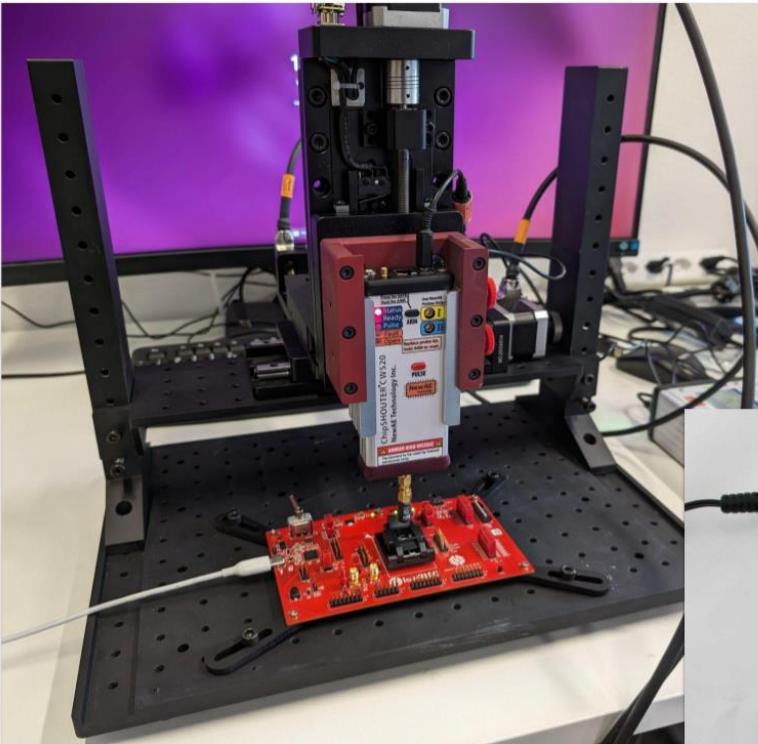
250+  
contributors  
(Ibex + OpenTitan)

3,700+  
GitHub stars  
(Ibex + OpenTitan)

40,000+  
test runs in nightly  
regressions  
(run multiple times per  
week)



# Build Securely



ot-sca - Side-Channel Analysis & Fault Injection  
Setup for OpenTitan

## About the repository

This repository contains infrastructure code useful for performing side-channel analysis (SCA) and fault injection (FI) attacks for [OpenTitan](#).

See [getting started](#) for instructions.

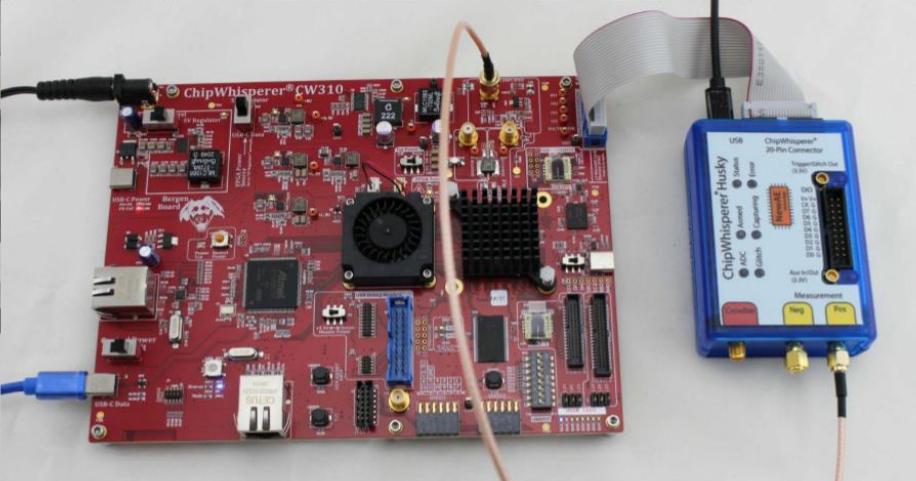
## How to contribute

Have a look at [CONTRIBUTING](#) for guidelines on how to contribute code to this repository.

## Licensing

Unless otherwise noted, everything in this repository is covered by the Apache License, Version 2.0. See [LICENSE](#) for full text.

## References



```
C verify.c X
opentitan > sw > device > silicon_creator > lib > sigverify > sphincsplus > C verify.c > spx_verify(const uint32_t *sig, const uint8_t *msg, size_t msg_len, const uint32_t *pk, uint32_t *root)
1 // Copyright lowRISC contributors (OpenTitan project).
2 // Licensed under the Apache License, Version 2.0, see LICENSE for details.
3 // SPDX-License-Identifier: Apache-2.0
4 //
5 // Derived from code in the SPHINCS+ reference implementation (CC0 license):
6 // https://github.com/sphincs/sphincsplus/blob/ed15dd78650f63288c7492c00260d86154b84637/ref/sign.c
7
8 > #include "sw/device/silicon_creator/lib/sigverify/sphincsplus/verify.h"
22
23 static_assert(kSpxVerifySigWords * sizeof(uint32_t) == kSpxVerifySigBytes,
24               "kSpxVerifySigWords and kSpxVerifySigBytes do not match.");
25 static_assert(kSpxVerifyPkWords * sizeof(uint32_t) == kSpxVerifyPkBytes,
26               "kSpxVerifyPkWords and kSpxVerifyPkBytes do not match.");
27 static_assert(kSpxD <= UDNTX_MAX, "kSpxD must fit into a uint8_t.");
28 rom_error_t spx_verify(const uint32_t *sig, const uint8_t *msg_prefix_1,
29                       size_t msg_prefix_1_len, const uint8_t *msg_prefix_2,
30                       size_t msg_prefix_2_len, const uint8_t *msg,
31                       size_t msg_len, const uint32_t *pk, uint32_t *root) {
32     spx_ctx_t ctx;
33     memcpy(ctx.pub_seed, pk, kSpxN);
34
35     // This hook allows the hash function instantiation to do whatever
36     // it needs, based on the public seed.
37     ix_hash_initialize(&ctx);

    ldr = {0};
    ldr = {0};
    .addr = {0};
    lr, kSpxAddrTypeWots;
    lr, kSpxAddrTypeHashTree;
    .addr, kSpxAddrTypeWotsPk;

    if (idx_leaf_index < R || PK || M) {
        // a result of the hash domain separator.
        goto done;
    }

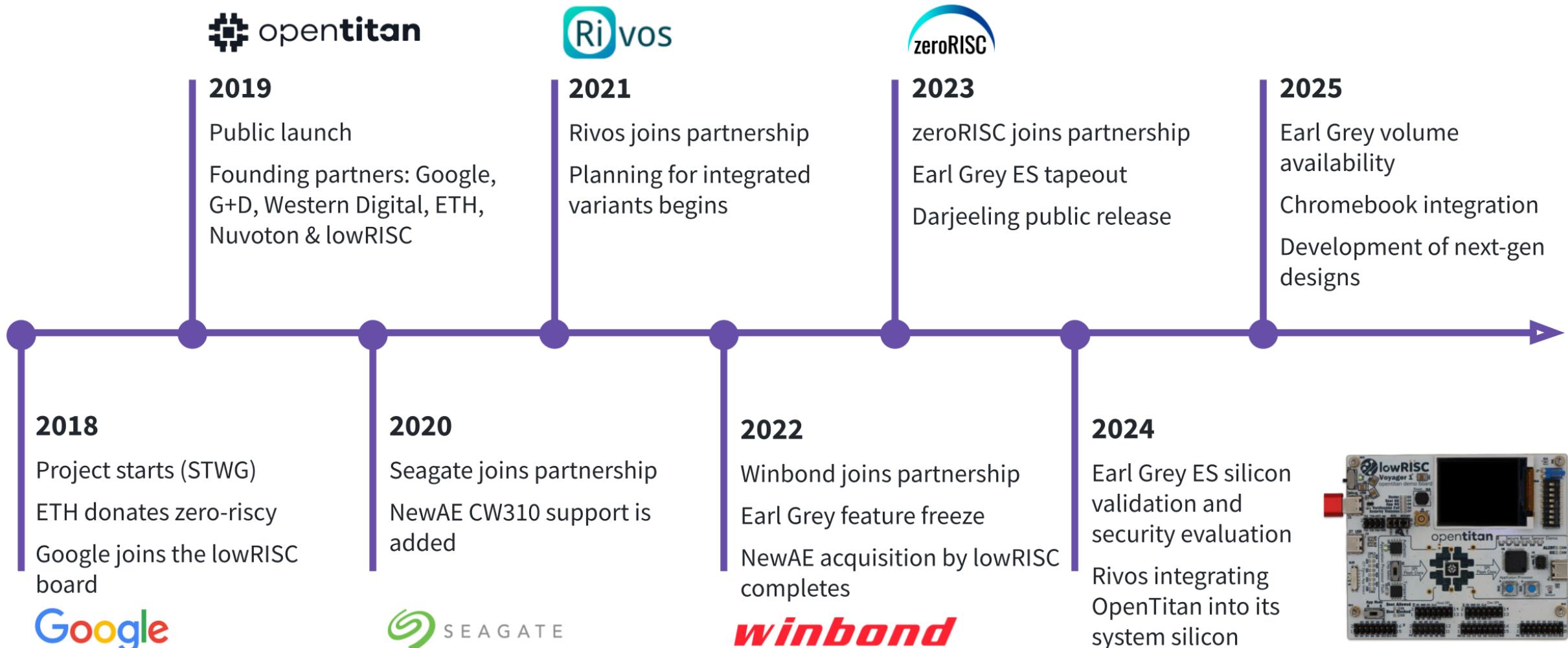
    ix_hash_message(
        msg_prefix_1_len, msg_prefix_2, msg_prefix_2_len,
        tree, &idx_leaf);

    if (idx_leaf == 0, so no need to set_layer_addr.
    lr, tree;
    .addr, idx_leaf);

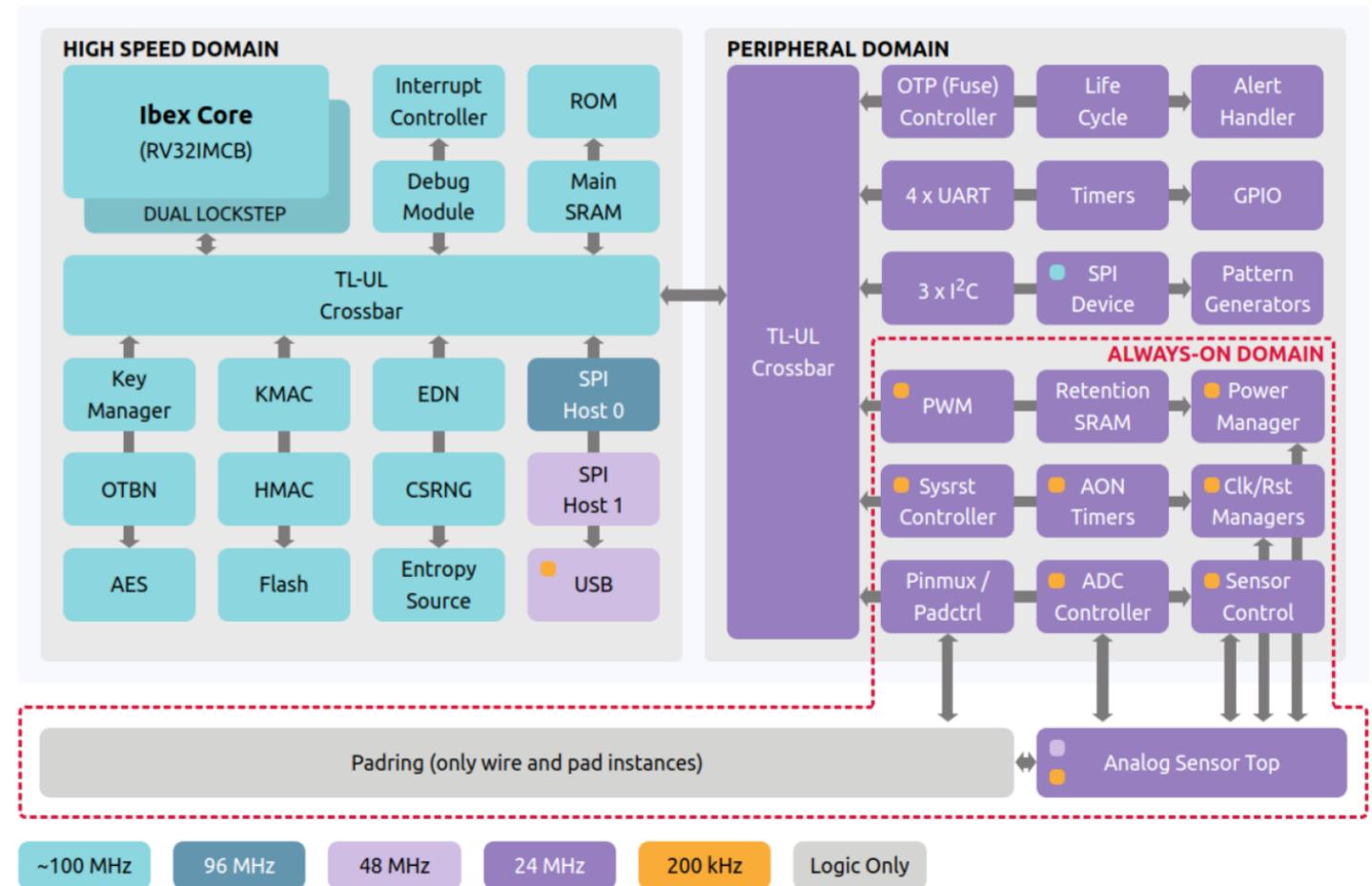
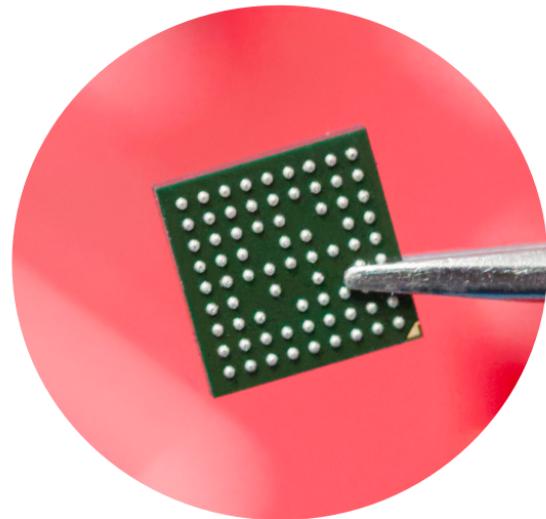
    , &ctx, &wots_addr, root);
```

PQC secure boot  
(SPHINCS+)

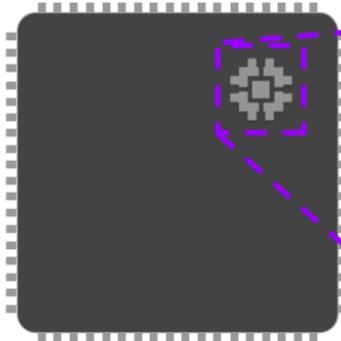
# Timeline



# The First OpenTitan® Chip Design: Earl Grey



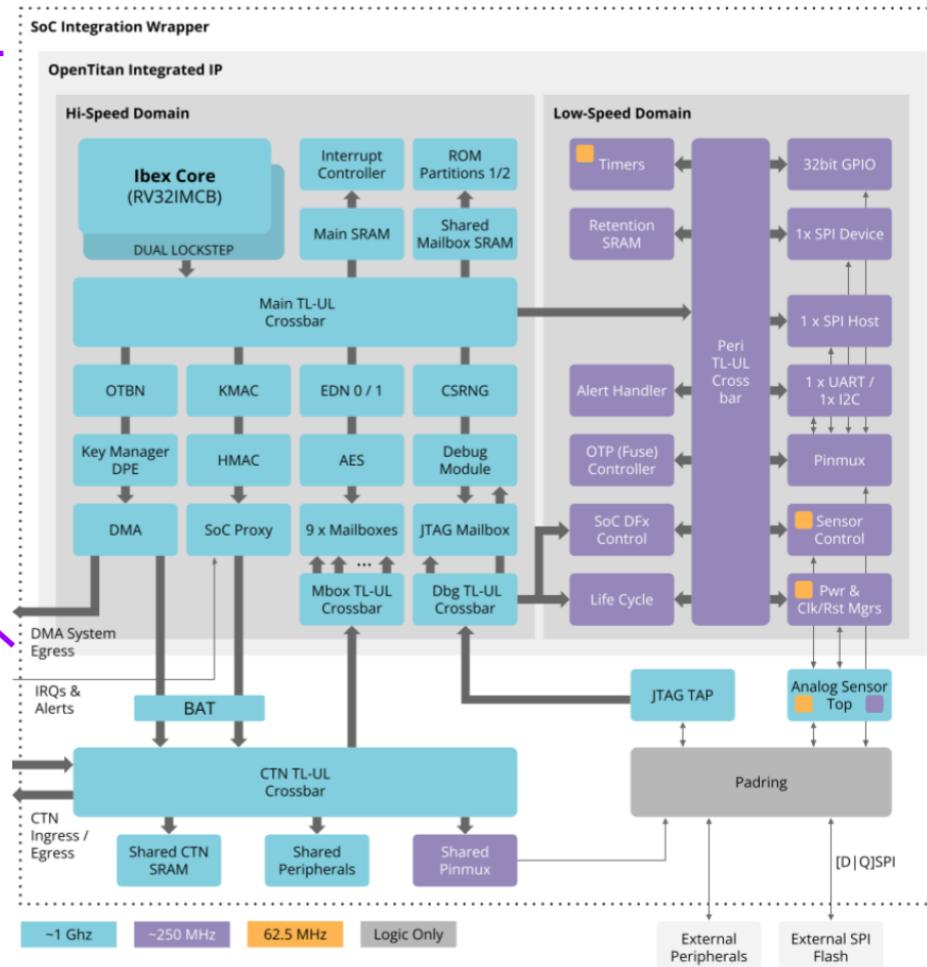
# Integrating OpenTitan®: Darjeeling and Beyond



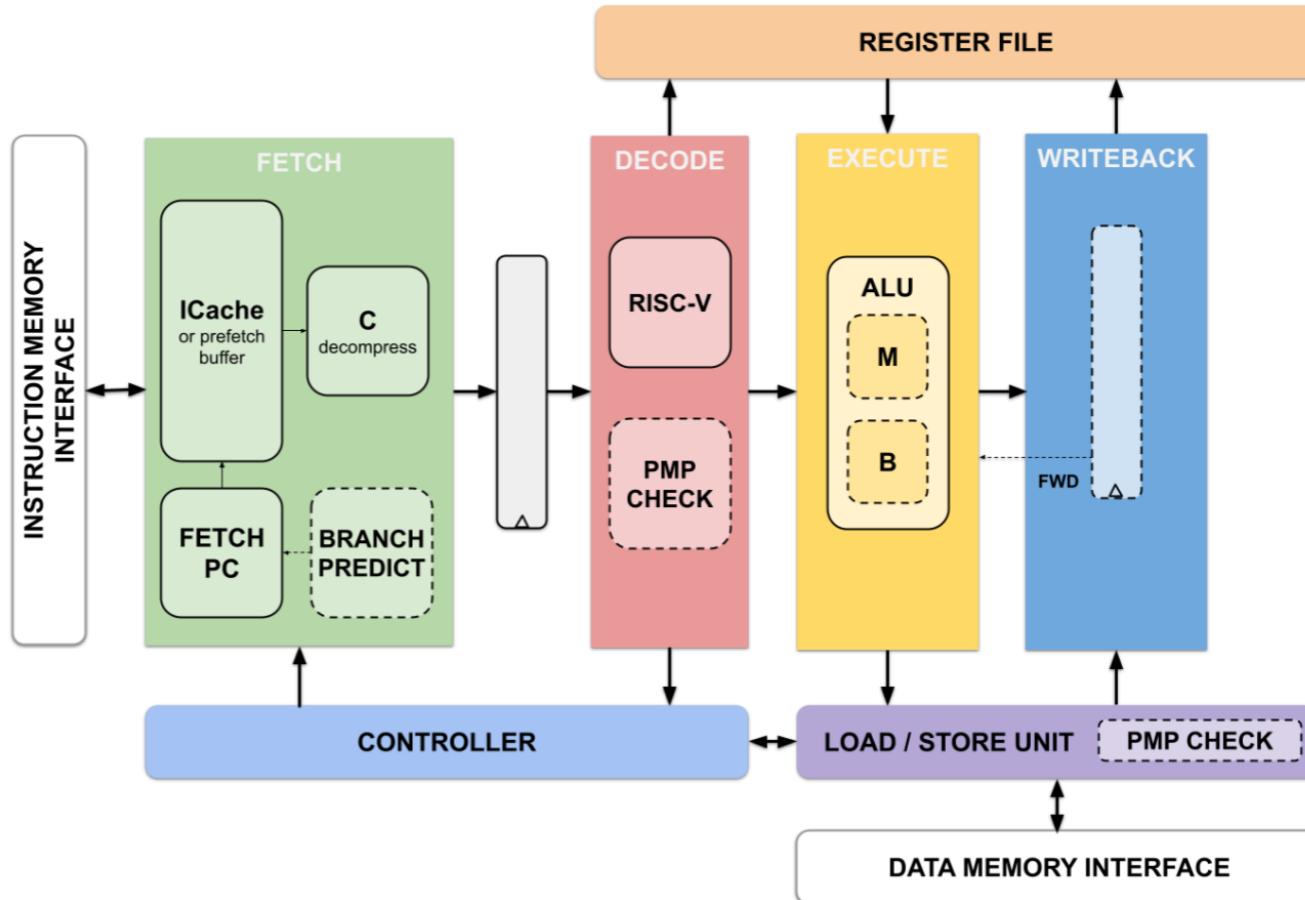
Third-party SoC  
or chiplet

*“With security as a foundational pillar, Rivos is integrating a RISC-V based root of trust solution, **OpenTitan**, directly into its system silicon.”*

<https://rivos.com/technology>



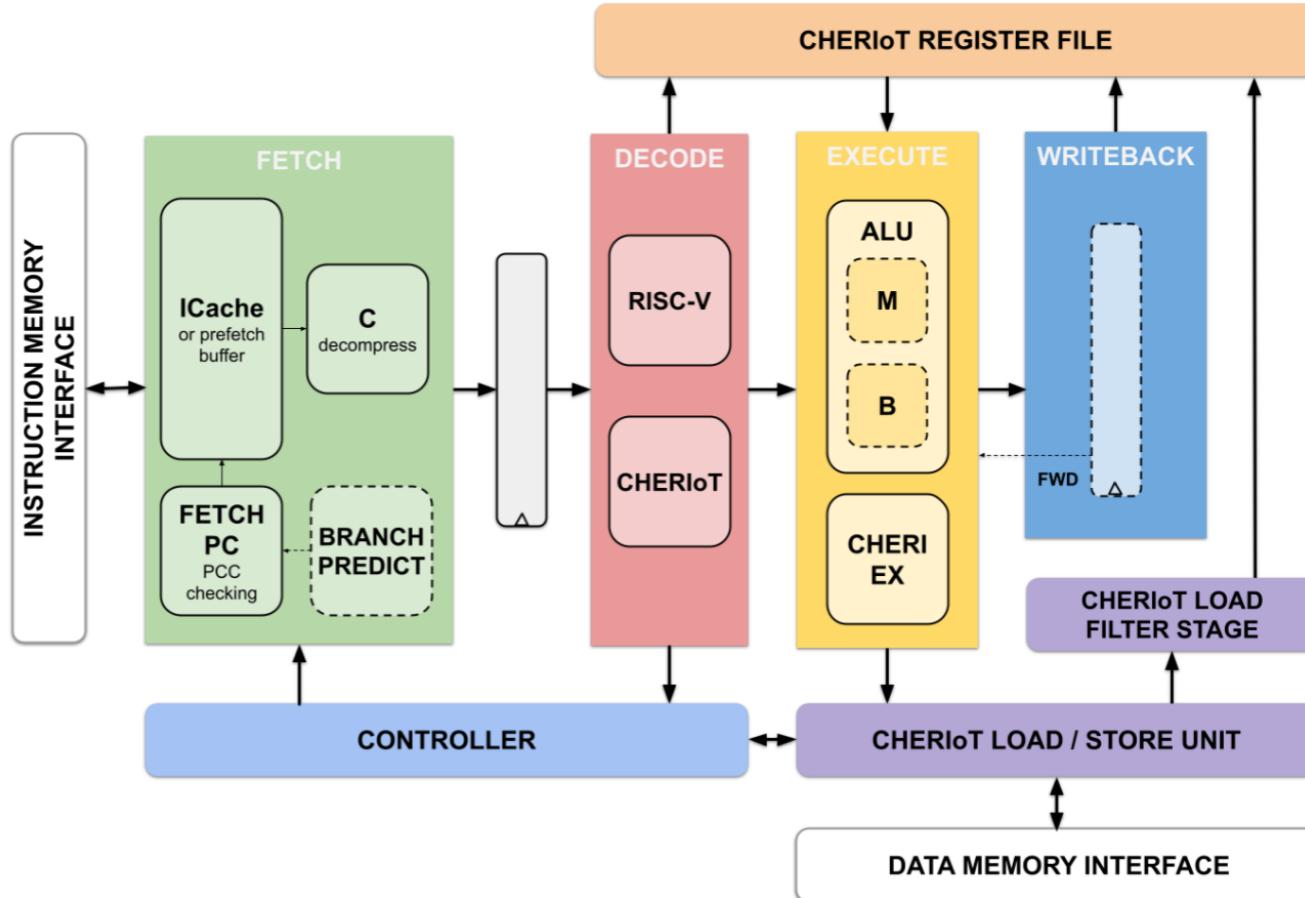
# RISC-V at the Core: Ibex®



Highly configurable **open source** RV32IMCB core with security features that include:

- Instruction cache memory scrambling
- Dual-core lockstep
- Data independent timing
- Dummy instruction insertion
- Bus and register file integrity
- Hardened PC

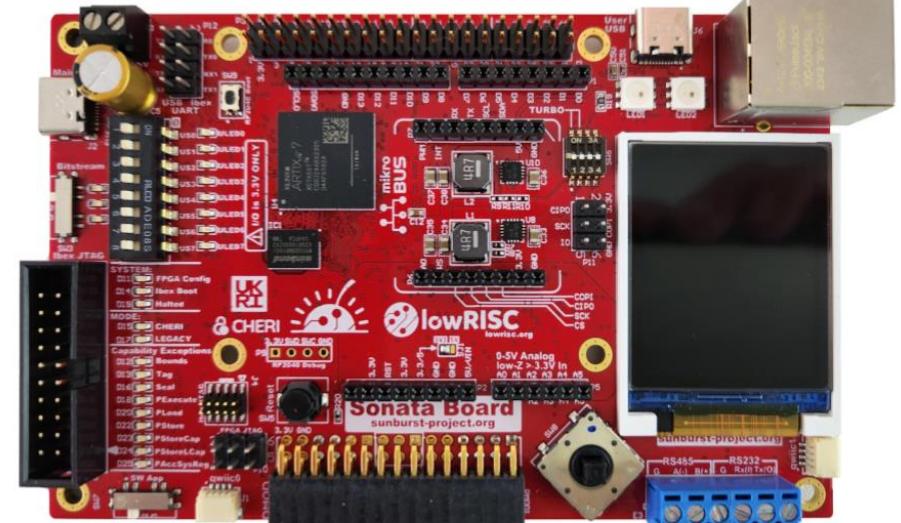
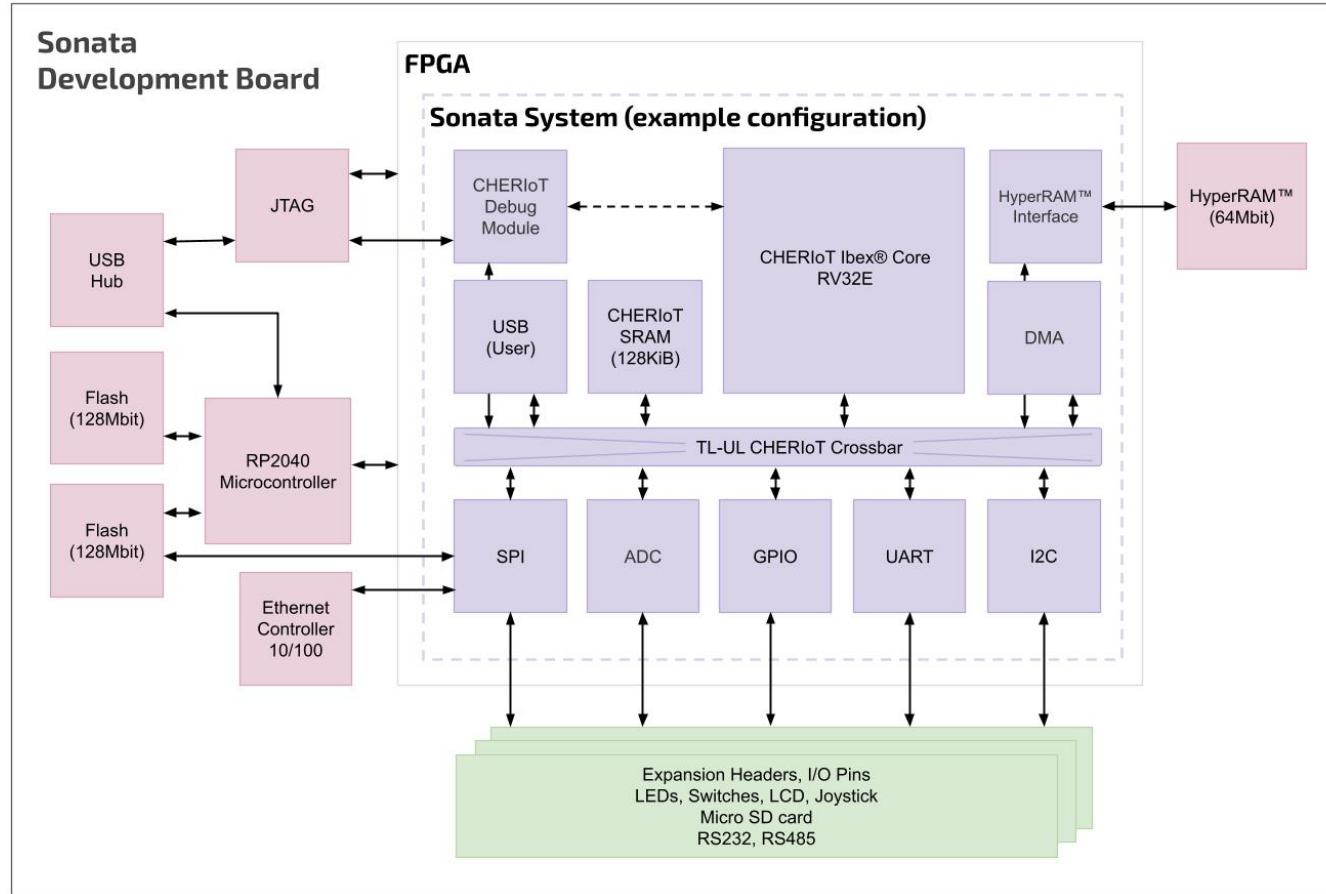
# Evolving Ibex®: Memory Safety with CHERIoT



*"This is truly important foundational work, as it will help make CHERIoT-Ibex the world's first production grade, open-source CHERI-enabled microcontroller core. We're looking forward to seeing it broadly leveraged in commercial designs, bringing much-needed hardware security – in an efficient manner – to a broad swathe of critical applications."*

Tony Chen  
Partner Security Architect, Microsoft

# Securing Operational Technology: Sonata



Delivered by  
Innovate UK,  
EPSRC and ESRC

Digital Security  
by Design

**sunburst-project.org**

DSbD/UKRI Project Grant Number: 107540

# Build Together, Build Well & Build Securely



*lowRISC's full-stack engineering team can help develop and integrate open source silicon IP in **your** products!*

*Contact us at [info@lowrisc.org](mailto:info@lowrisc.org) to find out more*