

Coverage-Driven Verification Methodology to Verify Highly Configurable RISC-V Core



Pascal Gouédo - Dolphin Design Salaheddin Hetalani, Nicolae Tusinschi - Siemens EDA

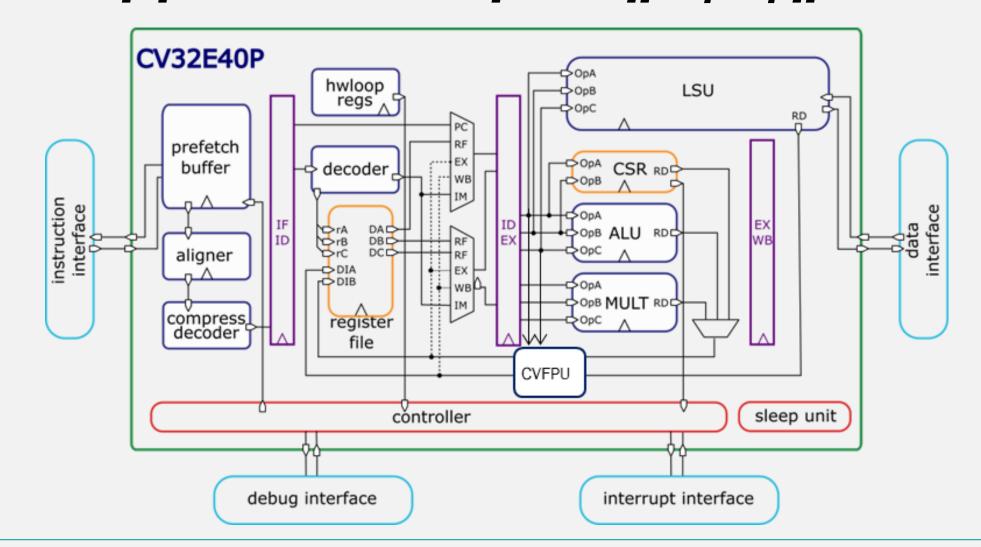
Motivation

- Ultimate freedom RISC-V offers introduces new level of processor verification challenges
- Broad range of complex architecture and μ -architecture choices available to meet PPA targets
- Simulation based verification techniques involve writing a testbench and a functional coverage model
- Debug process is slow and simulation can not hit all pipeline corner cases
- Customization introduces bugs in existing functionality and requires verification re-spins
- Questa Processor app accelerates, automates and increases quality of processor verification

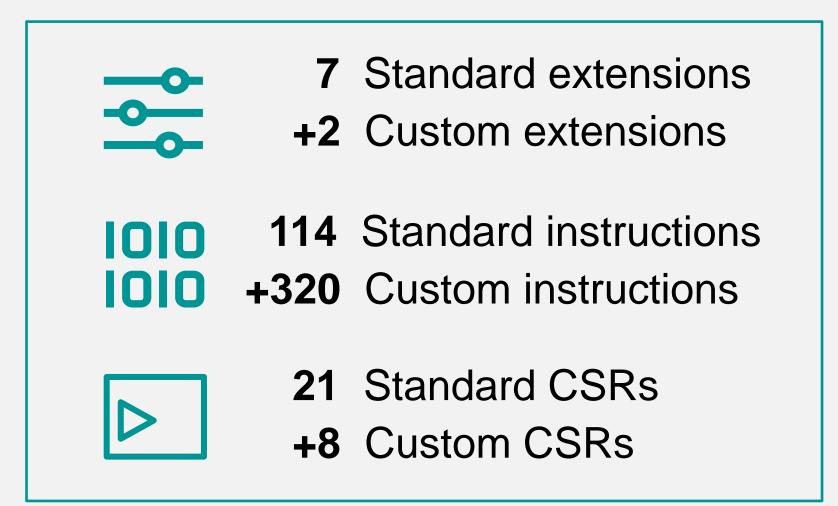
Highly customizable CV32E40Pv2 core

- 4-stage single-issue in-order pipeline
- Configurable through RTL parameters
- OBI protocol memory interfaces
- Standard external debug & interrupt support
- Floating-point extensions
- X custom instruction set extensions

RV32IM[F]C_Zicsr_Zifencei[_Zfinx][_Xpulp][_Xcluster]



Customization added to CV32E40Pv1



Verification methodology

- Design setup for the targeted configuration
- Pre-analysis configuration of X extensions
- Automated design analysis & μ -architecture extraction
- Post-analysis configuration of pipeline control signals & memory interfaces
- Automated assertion generation & memory interface constraining
- Assertion retuning to match pipeline timing behavior
- Automated assertion performance enhancement including initial value abstraction
- Assertion running and automated trace analysis of counter examples if any
- Mutation coverage analysis

< IVA

Application results



9 Configurations



~2 hour Runtime of 70% of assertions per configuration



33 Bugs



100% Unbounded proofs



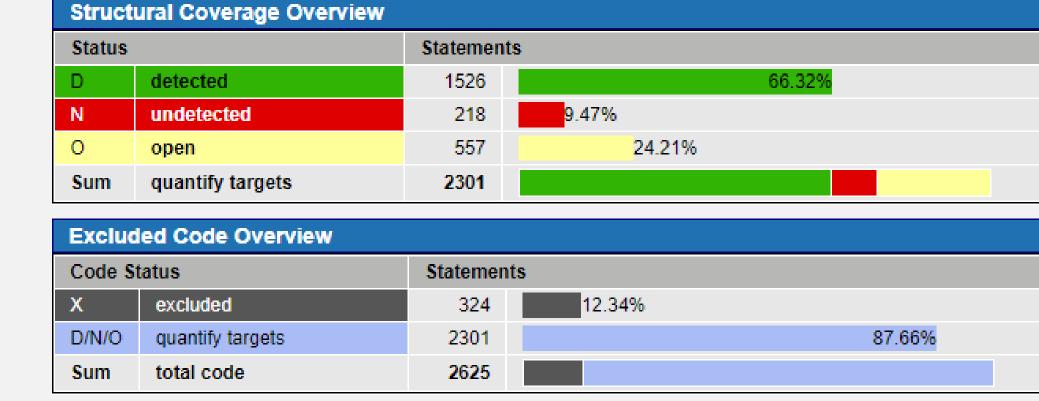
~185 Assertions per configuration

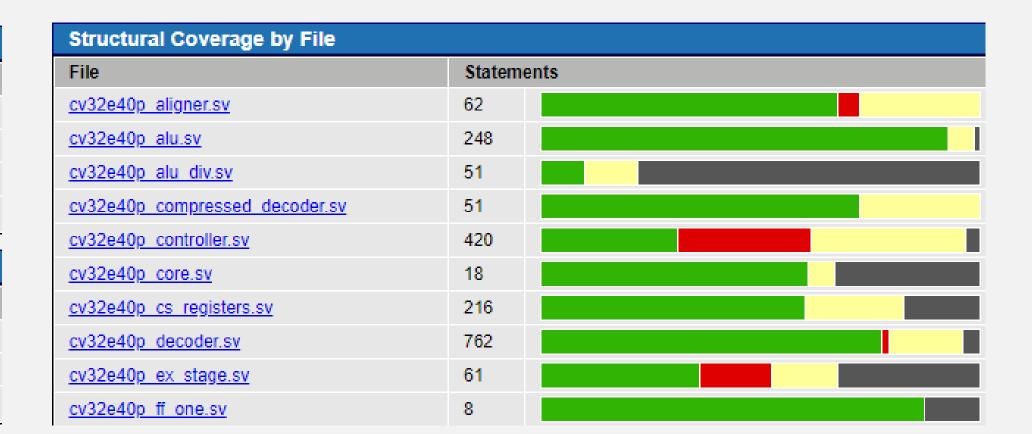
Review bug specifications here:

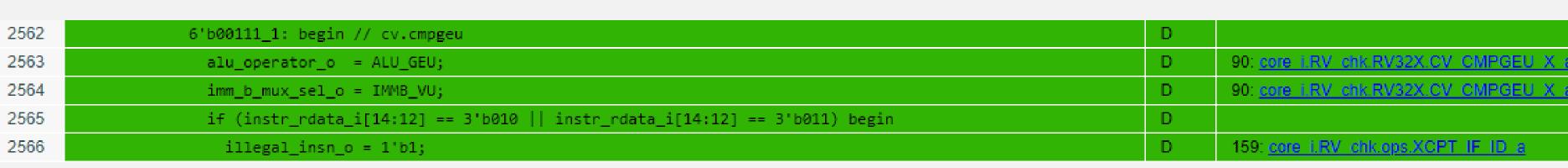
https://github.com/openhwgroup/cv32e40p/issues

Coverage results of sample configuration

- Target configuration is RV32IMC_Zicsr_Zifencei_Xpulp
- ~75% coverage achieved in less than 72 hours







Line by line source code assertion annotation

Summary

Successful coverage-driven formal verification

- Verification effort is down to adding X extension specification
- 33 bugs were identified across 9 configurations

Considerable verification speed-up

- No writing of testbench
- Optimized formal engines and assertions
- Pinpoint bugs quick fix check

High degree of automation

- No writing of functional coverage model
- μ -architecture extraction
- Assertion generation
- Trace analysis & performance enhancement