

# RISC-V Configurability in Compliance Test Framework

Milan Skála

Radek Hájek



# Agenda

- Achievements and activities of Codasip in Compliance Task Group
- What is Codasip Compliance Framework
- Framework scheme
- Automatic plugin generation
- Custom model integration process
- Framework reporting
- Automatic test selection

# Achievements and Activities of Cudasip in CTG

- Initialized RISC-V compliance test process definition
- Implementation of RV32I tests
- Proposal of test format specification
- Compliance framework implementation

# Compliance Test Format

- Unified directory structure
- File naming convention
- Test structure
  - Unified user defined macros for test sections
  - Common header files
- Signature format specification
- Glossary definition



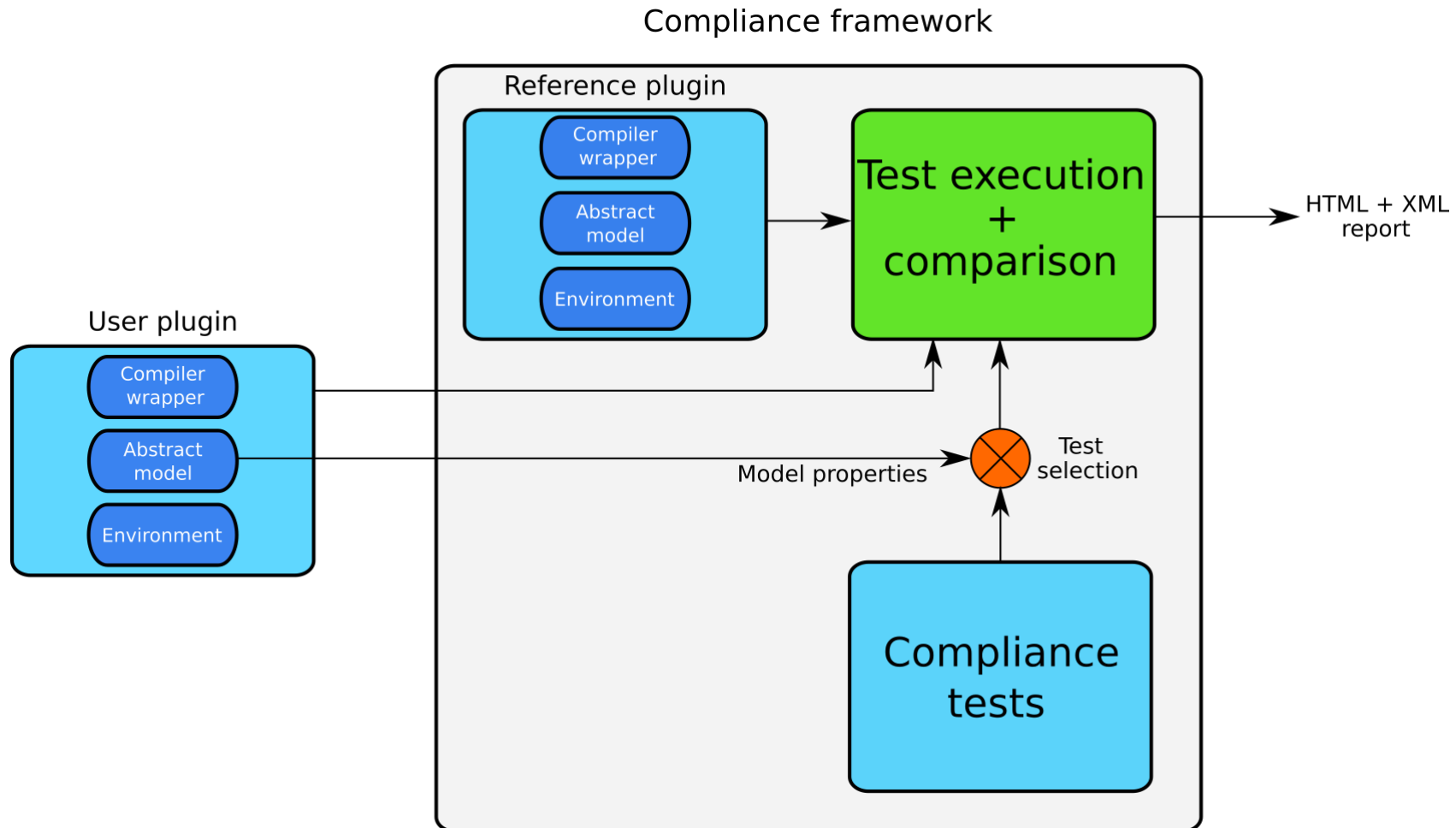
# Codasip Compliance Framework

- High-level framework for compliance testing of RISC-V models
- Highlighted features:
  - ✓ Supports any implementation of RISC-V processor (ISA, RTL, ...)
  - ✓ Easily extensible with new tests and models
  - ✓ Automatic tests selection based on tested model configuration
  - ✓ Support of custom toolchain for tests compilation
  - ✓ Simple and intuitive user interface
  - ✓ User-friendly reporting system

# Selected Technology

- Python 3
  - High-level OOP description
  - Fast code implementation
  - Easily readable
- Pytest
  - Package for automated testing
  - Supports full customization of the testing process
  - Allows implementing custom plugins and their easy integration
- xdist plugin
  - Parallel test execution

# Compliance Framework Scheme



# Plugin Structure

- Abstract model
  - Implements functional behavior of the tested model
  - Responsible for signature extraction
- Compiler wrapper
  - Needed only when using custom toolchain (not RISC-V GCC)
  - Responsible for correct assembly compilation
- Environment
  - Contains files for correct compilation process (header files, linker scripts, etc.)



# Automatic User Plugin Generation

The framework implements a plugin generator for easier model integration

The screenshot shows a window titled "RISC-V Compliance Testsuite plugin generator". The window has a dark blue background with a white sidebar on the left and a main content area on the right. The sidebar contains a list of menu items: "Introduction", "ISA, Extensions and Modes" (highlighted in orange), "Memory and interrupts", "Interrupt vectors", "Control and State Registers", "Target", and "Finish". The main content area is titled "ISA, Extensions and Modes" and contains the following text: "Choose instruction set architecture, supported extensions and modes." Below this text, there is a dropdown menu for "ISA" with "rv32i" selected. Under the "Extensions" section, there are checkboxes for M, A, F, D, Q, L, C, B, J, T, P, V, and N. Under the "Modes" section, there are checkboxes for U, S, H, and M. At the bottom right of the window, there are three buttons: "Previous", "Next", and "Cancel".

# Custom Model Integration

1. Generate a plugin using the plugin generator
  - Creates directory structure and default implementations
2. Define macros for the custom processor model
  - Program start, memory size, reset vector address, etc.
3. Implement a custom compiler wrapper if needed
  - Default implementation is auto-generated, but can be overridden
4. Implement abstract model behavior
  - Specification of arguments and execution of custom RISC-V model
5. Execute test suite

# Framework Reporting

- Framework generates reports in multiple formats:
  - XML report for machine post-processing
  - HTML report for user-friendliness
- Ability to filter tests by the result (passed, failed, skipped)

## Summary

79 tests ran in 24.64 seconds.

(Un)check the boxes to filter the results.

79 passed,  105 skipped,  0 failed,  0 errors,  0 expected failures,  0 unexpected passes

## Results

[Show all details](#) / [Hide all details](#)

▲ Result	▼ Test
Skipped ( <a href="#">hide details</a> )	compliance-tests/compliance_tests/rv64i/M/ISA/test_isa.py::test_rv64i_m_isa[REMW.S]::setup
	('compliance-tests/compliance_tests/rv64i/M/ISA/test_isa.py', 6, 'Skipped: Test requires architecture ISAS.RV64I')
Skipped ( <a href="#">hide details</a> )	compliance-tests/compliance_tests/rv64i/M/ISA/test_isa.py::test_rv64i_m_isa[REMUW.S]::setup
	('compliance-tests/compliance_tests/rv64i/M/ISA/test_isa.py', 6, 'Skipped: Test requires architecture ISAS.RV64I')
Skipped ( <a href="#">hide details</a> )	compliance-tests/compliance_tests/rv64i/M/ISA/test_isa.py::test_rv64i_m_isa[MULW.S]::setup
	('compliance-tests/compliance_tests/rv64i/M/ISA/test_isa.py', 6, 'Skipped: Test requires architecture ISAS.RV64I')
Skipped ( <a href="#">hide details</a> )	compliance-tests/compliance_tests/rv64i/M/ISA/test_isa.py::test_rv64i_m_isa[DIVW.S]::setup
	('compliance-tests/compliance_tests/rv64i/M/ISA/test_isa.py', 6, 'Skipped: Test requires architecture ISAS.RV64I')
Skipped ( <a href="#">hide details</a> )	compliance-tests/compliance_tests/rv64i/I/ISA/test_isa.py::test_rv64i_i_isa[SUBW.S]::setup
	('compliance-tests/compliance_tests/rv64i/I/ISA/test_isa.py', 21, 'Skipped: Test requires architecture ISAS.RV64I')

# Framework Reporting: Console

```
===== test session starts =====
platform linux -- Python 3.6.5, pytest-3.6.3, py-1.5.4, pluggy-0.6.0
ISA: ISAS.RV32I
Toolchain: /home/jenkins/git/codix_berkelium/models/codix_berkelium/work/ia/sdk/bin/
Plugin path: /home/jenkins/git/compliance-tests/plugins/codasip_all/rvtest_plugin.py
RISC-V model path: /home/jenkins/git/codix_berkelium/models/codix_berkelium/work/ia/sdk/bin/codix_berkelium-ia-isimulator
Reference environment: /home/jenkins/git/compliance-tests/golden_model/p
rootdir: /home/jenkins/git, inifile:
plugins: xdist-1.23.0, forked-0.2, html-1.19.0, metadata-1.7.0
collected 184 items

compliance_tests/rv32i/A/U/test_isa.py sssssssss
compliance_tests/rv32i/C/ISA/test_isa.py .....
compliance_tests/rv32i/C/U/test_isa.py s
compliance_tests/rv32i/D/U/test_isa.py sssssssss
compliance_tests/rv32i/F/U/test_isa.py sssssssssss
compliance_tests/rv32i/I/ISA/test_isa.py .....S
compliance_tests/rv32i/I/M/test_isa.py sssssssss
compliance_tests/rv32i/I/S/test_isa.py ssssss
compliance_tests/rv32i/I/U/test_isa.py sssssssssssssssssssssssssssssssssssssss
compliance_tests/rv32i/M/ISA/test_isa.py sssssssss
compliance_tests/rv64i/I/ISA/test_isa.py sssssssss
compliance_tests/rv64i/M/ISA/test_isa.py ssss

----- generated xml file: /home/jenkins/git/compliance-tests/rvtest_work/report/report.xml -----
----- generated html file: /home/jenkins/git/compliance-tests/rvtest_work/report/report.html -----
RISC-V model with ISA configuration: RV32IC
Base ISA: 54/54 passed (OK)
C: 25/25 passed (OK)
===== 79 passed, 105 skipped in 24.42 seconds =====
```

# Automatic Test Selection

- Framework selects and executes only tests that should pass on a specific RISC-V configuration
- Uses Pytest markers
  - Easy way to filter tests
  - Can be parametrized
  - Extensible for any RISC-V feature or configuration option

# Automatic Test Selection

```
@pytest.mark.find_files(path=os.path.join('rv32i', 'M', 'ISA'), pattern='\.S$')
@pytest.mark.architecture(isa=[ISAS.RV32I], extensions=[RiscVExtensions.M])
def test_rv32i_m_isa(file, reference, user, work_dir):

    output_ref = os.path.join(work_dir, os.path.basename(file)+'.ref.xexe')
    output_test = os.path.join(work_dir, os.path.basename(file)+'.test.xexe')

    metadata = {'isa': 'rv32im'}
    # Compile assemblies
    reference.compiler.run(file, output_ref, metadata)
    user.compiler.run(file, output_test, metadata)
    # Execute on golden model and user model
    reference = reference.model.run(output_ref, metadata)
    actual = user.model.run(output_test, metadata)
    # Compare signatures
    assert reference.signature == actual.signature, "Signature mismatch"
```

# Supported Targets

- Codasip SDK + Codasip simulator (Bk series)
- RISC-V GCC + Codasip simulator (Bk series)
- RISC-V GCC + riscvOVPsim by Imperas
- ri5cy verilator – RI5CY core + verilator
- Codasip SDK + Codasip RTL (proprietary)

# Thank you

Questions?



[skala@codasip.com](mailto:skala@codasip.com)

[www.codasip.com](http://www.codasip.com)

Codasip GmbH