Accelerating RISC-V Innovation with open MPACT Tools from Google

Tor Jeremiassen
Julian Mullings-Black
Yenkai Wang
Google Research

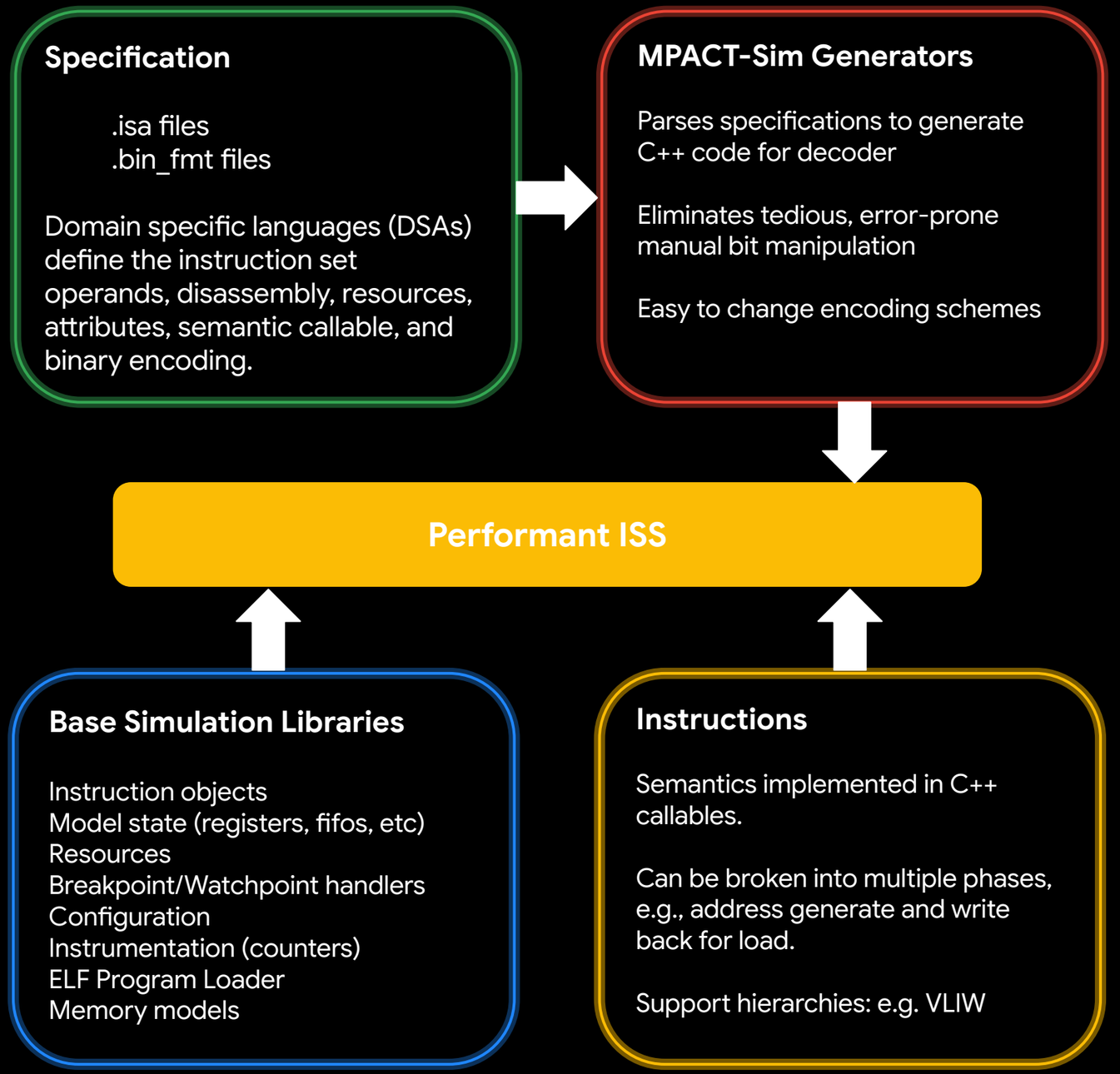
01: MPACT-Sim SDK
02: MPACT-RiscV Architecture Simulator
03: CoralNPU Simulator
04: MPACT-DV Co-Simulation

MPACT-Sim

Instruction Set Simulator SDK

Make it easy to write performant instruction set simulators quickly for a wide range of ISAs

Must be easy to modify and extend.



The Impact of MPACT-Sim

Optimized for Rapid Iteration and Early Verification

Developer Velocity

Automates the tedious creation of instruction decoders. Engineers can focus on instruction semantics and iterate rapidly on ISA design changes.

HW/SW Co-Design

Enables critical pre-Silicon software development. Facilitates rapid prototyping of specialized instructions and new architectures long before hardware is finalized.

Architectural Flexibility

Supports a wide range of architectures from single issue to VLIWs, CPUs to accelerators.

Easy to customize instruction issue and to support multi-core simulation.

Ecosystem Ready

Automatically generates core pieces of assemblers and disassemblers from encoding and isa specifications. Defines a debug interface for CLI debuggers, integrated support for GDB Server, and provides integration with Renode for full-system simulations.

MPACT-RiscV

A highly configurable plug-and-play RISC-V simulator engine built upon the foundational MPACT-Sim framework.

Designed for rapid architecture exploration and pre-Si software development.

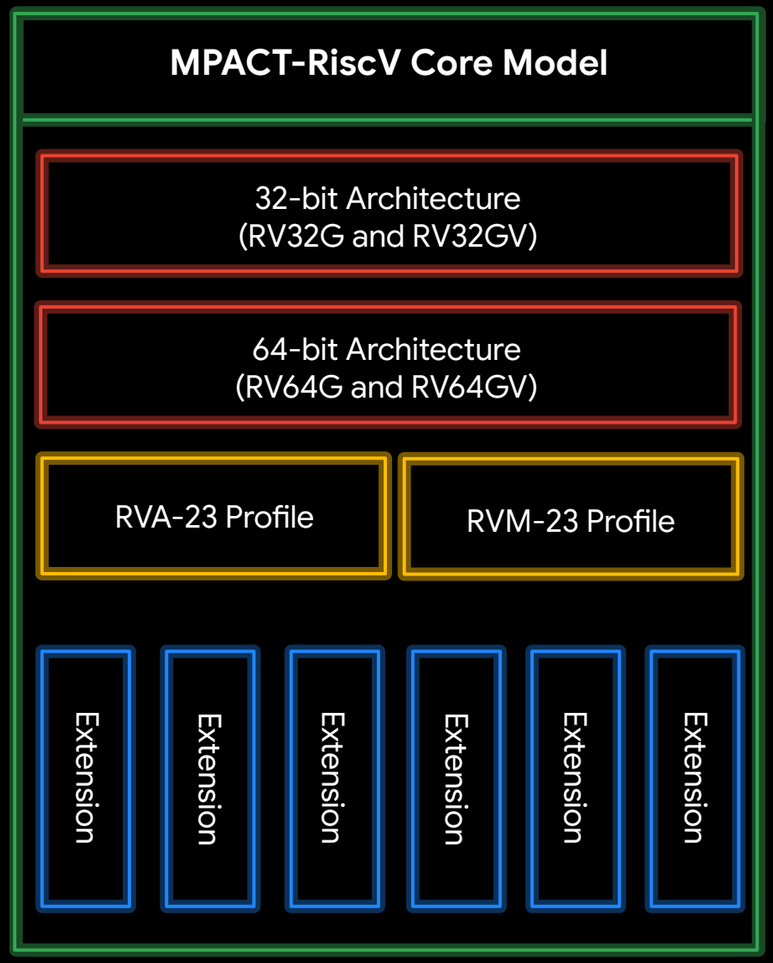
Supports both 32-bit and 64-bit with and without vector extension.

Supports RVA-23 and RVM-23 profiles.

Very easy to choose ISA extensions, or add new ones.

Multi-core support.

Open sourced and available.



Risc-V SW Dev Support

Interactive Native Debug CLI

Breakpoints and watchpoints.
Read/write registers and memory.
Step/run.
Disassembly.
Multi-core support.

Assembler

ISA DSL files used to generate simple two pass assembler.
Supports both relocatable and executable output files.
Easy to try out new custom instructions in kernels in simulator, both standalone and integrated into existing code.

Disassembler

Easy to create standalone disassembler from instruction decoder and ELF loader.

GDB Server Integration

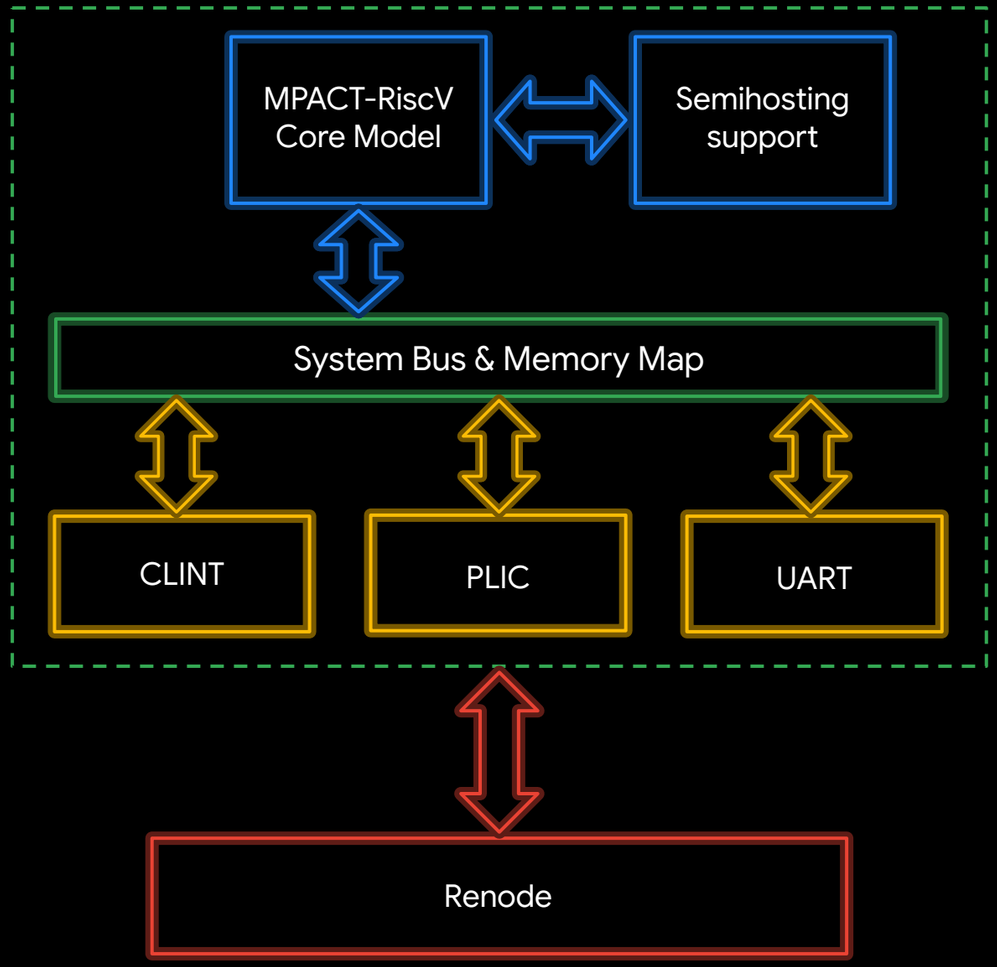
GDB Server implements gdb remote protocol
Allows for source level debugging with lldb and/or gdb

System Modeling

Out-of-the-box simulation support for essential system components allows for easy integration with system modeling.

- Semihosting support
- Core Local Interrupt Controller (CLINT)
- Platform Level Interrupt Controller (PLIC)
- Serial communication (UART)

Supports integration with Renode system modeling framework.



CoralNPU

A RISC-V compliant, full-stack platform for Edge AI

Designed by Google Research, CoralNPU provides an open-source, foundational ecosystem to run Edge AI models locally on edge devices.

Coral board™
Coral npu™

AI-First Architecture

The architecture features a scalar and vector core setup perfectly targeted to address the massive computational needs of neural networks.

On-Device Generative AI

Hardware acceleration supports the Gemma model, enabling local conversational assistants and multimodal experiences.

Unified Developer Tools

Compile and deploy models from PyTorch, JAX, and TensorFlow through a single, unified edge toolchain.

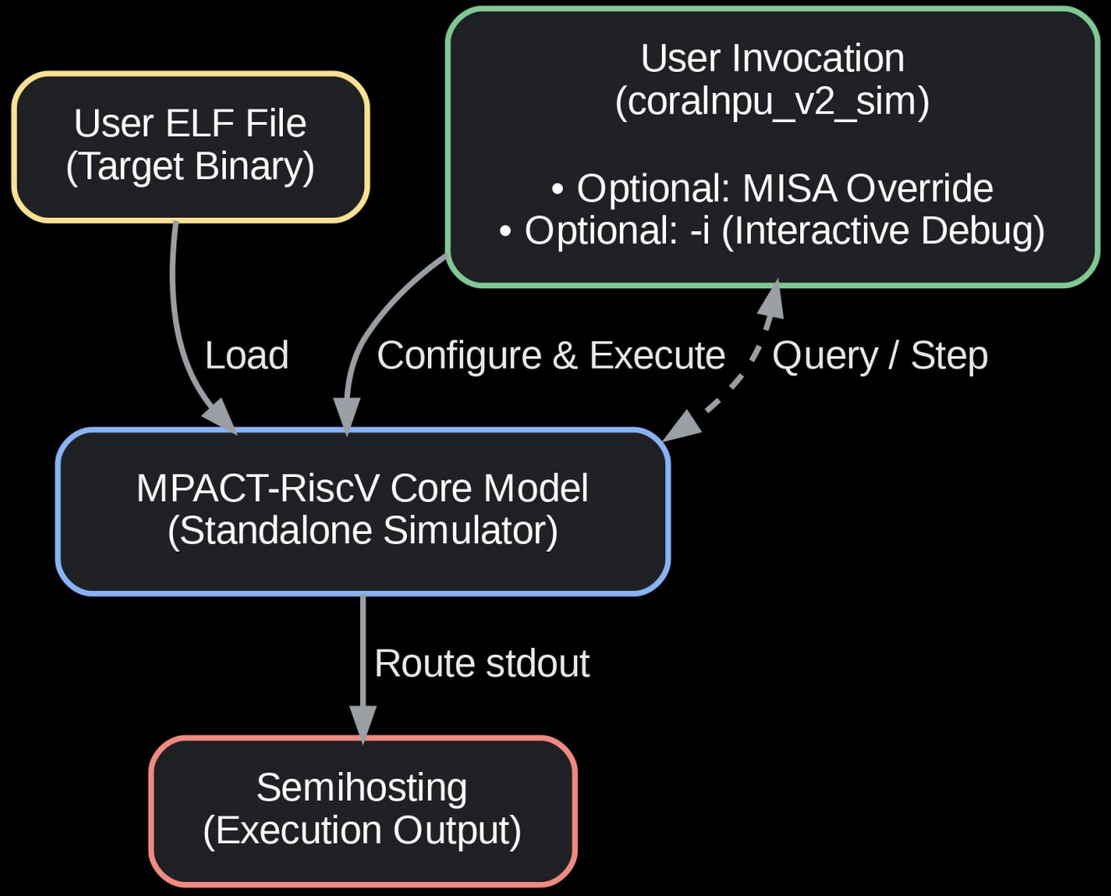
Real-World Integration

Adopted by Synaptics, the architecture powers the Astra SL2610 product line, bringing scalable edge AI experiences directly to developers worldwide.

CoralNPU-MPACT

Simulation & Debugging

- **coralnpu_v2_sim**: Parses ELF/BIN targets, maps sections to simulated ITCM, and auto-configures registers (e.g., MISA).
- **Native Debugger** (via `--interactive`):
  - Insert breakpoints and step through ML vector instructions.
  - Real-time register file and memory map queries.

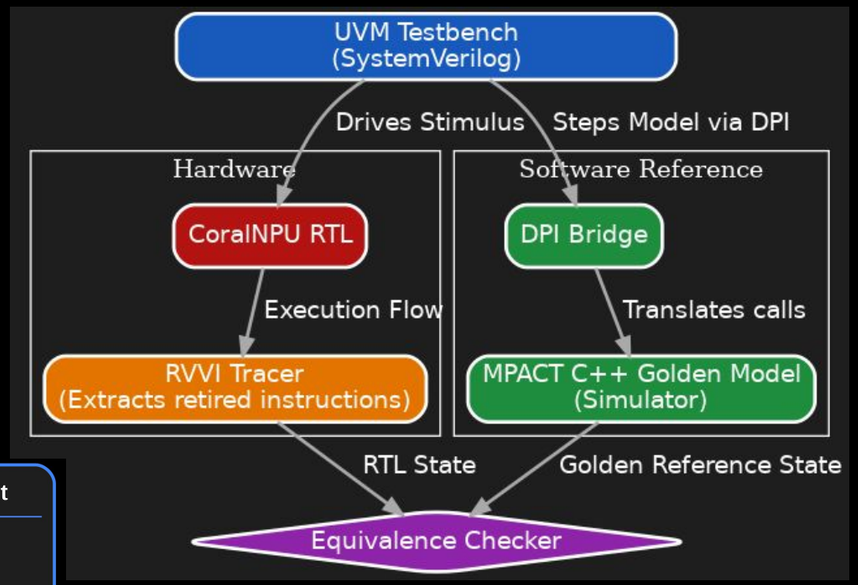


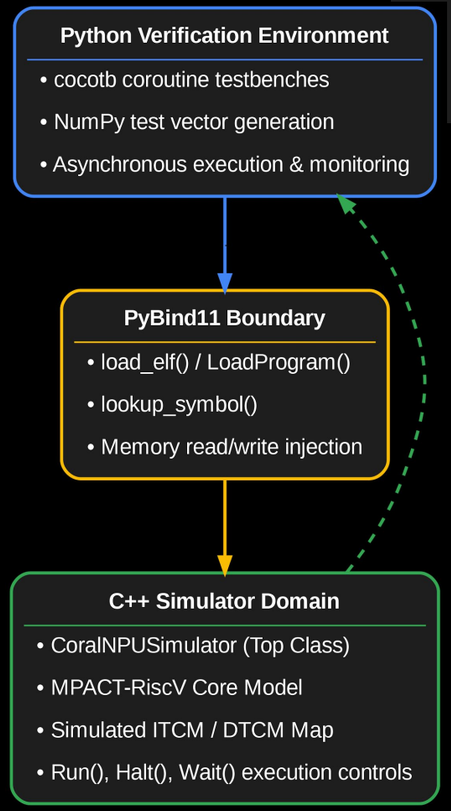
Coral board™
Coral npu™

CoralNPU-MPACT

Verification & Test Reuse

- **UVM Control:** SystemVerilog DPI bridges UVM testbenches to MPACT C++ models for lockstep RTL simulation.
- **Golden Reference Execution:** MPACT serves as the functional ground-truth.
- **State Equivalence Checking:** RVVI enables cycle-by-cycle comparison of hardware-retired instruction states against MPACT golden values.
- **Core Simulator Bridging:** PyBind11 wrappers expose CoralNPUSimulator controls (LoadProgram, Run, Halt) to Python.
- **High-Velocity Test Reuse:** Enables rapid software validation using asynchronous, cocotb-based RTL testbenches.
- **Direct Memory Injection:** Python tests utilize `lookup_symbol` to inject NumPy-generated test vectors into the NPU memory map pre-execution.



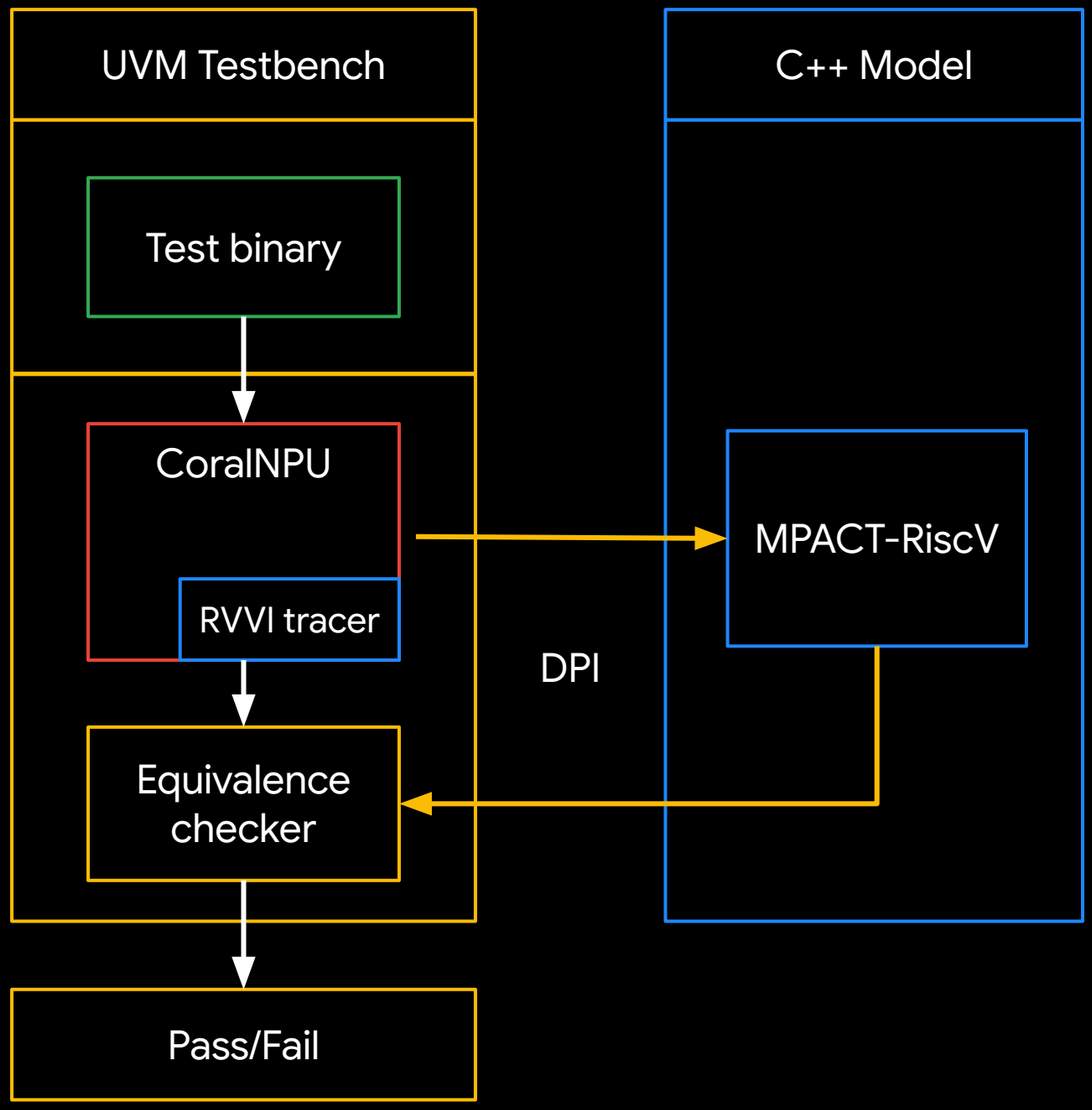


Coral

MPACT for DV

Co-Simulation for CoralNPU

- DPI bridges UVM testbench and MPACT C++ model.
- Equivalence checking uses rvviTrace on the RTL side.
- Successfully verified over 400 test binaries.
- Proves feasibility of MPACT as a golden reference.



Coral board™
Coral npu™

Conclusion

The MPACT ecosystem provides a unified, scalable path from a foundational software development kit all the way to verifiable SystemVerilog test benches.

Open sourced, in use, and available:

- MPACT-Sim: https://github.com/google/mpact-sim
  - Tutorial: https://developers.google.com/mpact-sim
- MPACT-RiscV: https://github.com/google/mpact-riscv
- CoralNPU Sim: https://github.com/google-coral/coralnpu-mpact
- CoralNPU: https://github.com/google-coral/coralnpu

Coral