# Renode Co-Simulation for Caravel / MPW Shuttle

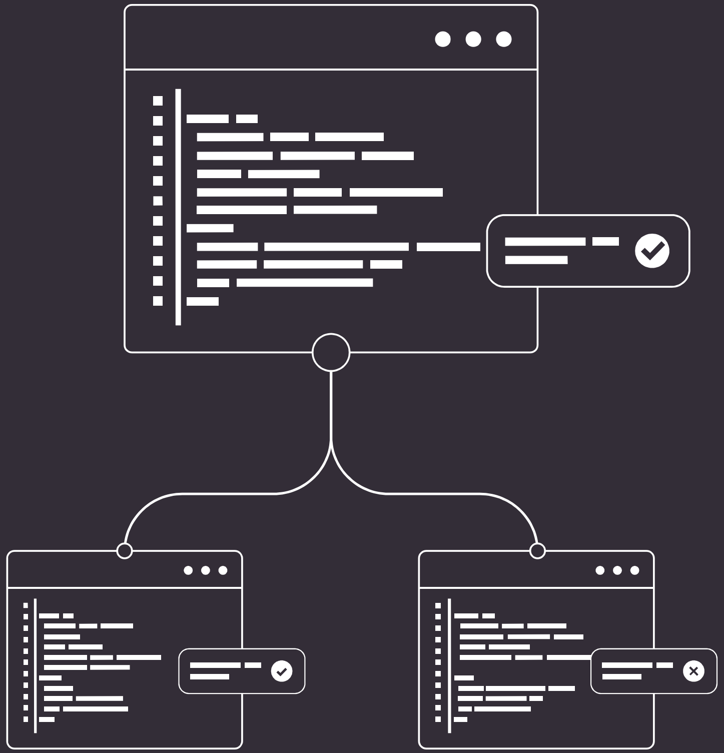
**CHIPS ALLIANCE FALL TECHNOLOGY UPDATE**
**Sunnyvale, CA, 2022-12-15**

Peter Gielda, pgielda@antmicro.com

antmicro

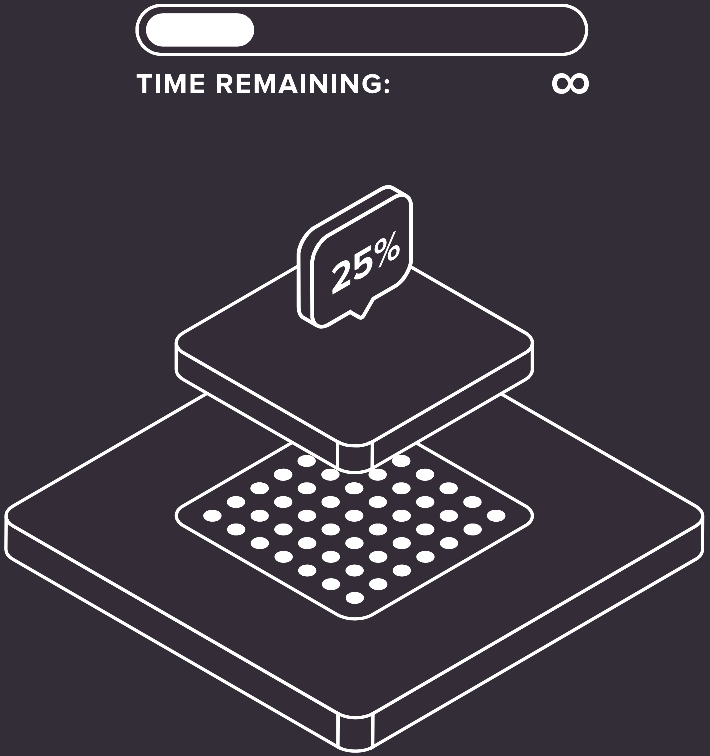
# SOFTWARE MOVES FAST

- Modern software development is based on extremely rapid iteration, using CI and failing fast
- Building a feedback loop between idea and implementation, and assessing product-market fit is relatively simple
- You can build an end-to-end “MVP” and add complexity interactively, constantly making sure things are tested



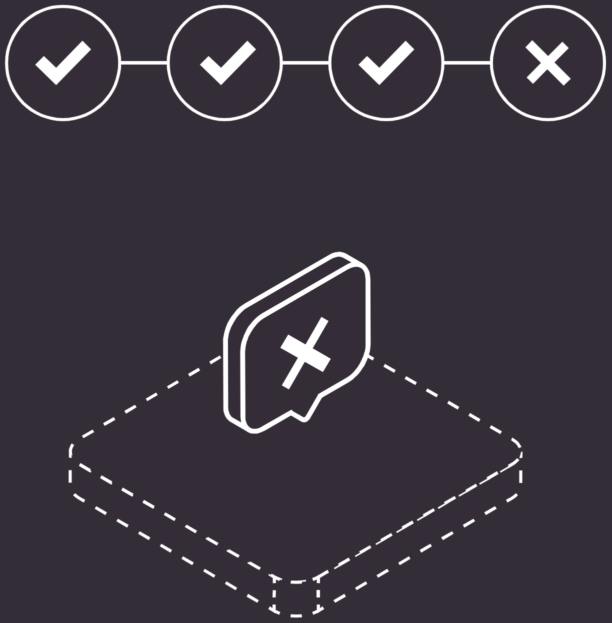
# CHIP-MAKING IS NOT AS LUCKY

- The ASIC design process does not enjoy many of those privileges
- Much more “waterfall” approach
- End-to-end system testing often needs to wait until late into the project
- System software blocked on hardware bringup, in turn blocked on manufacturing
- By the time your chip is back from the fab and your software team builds out the complete system, maybe the market has already shifted

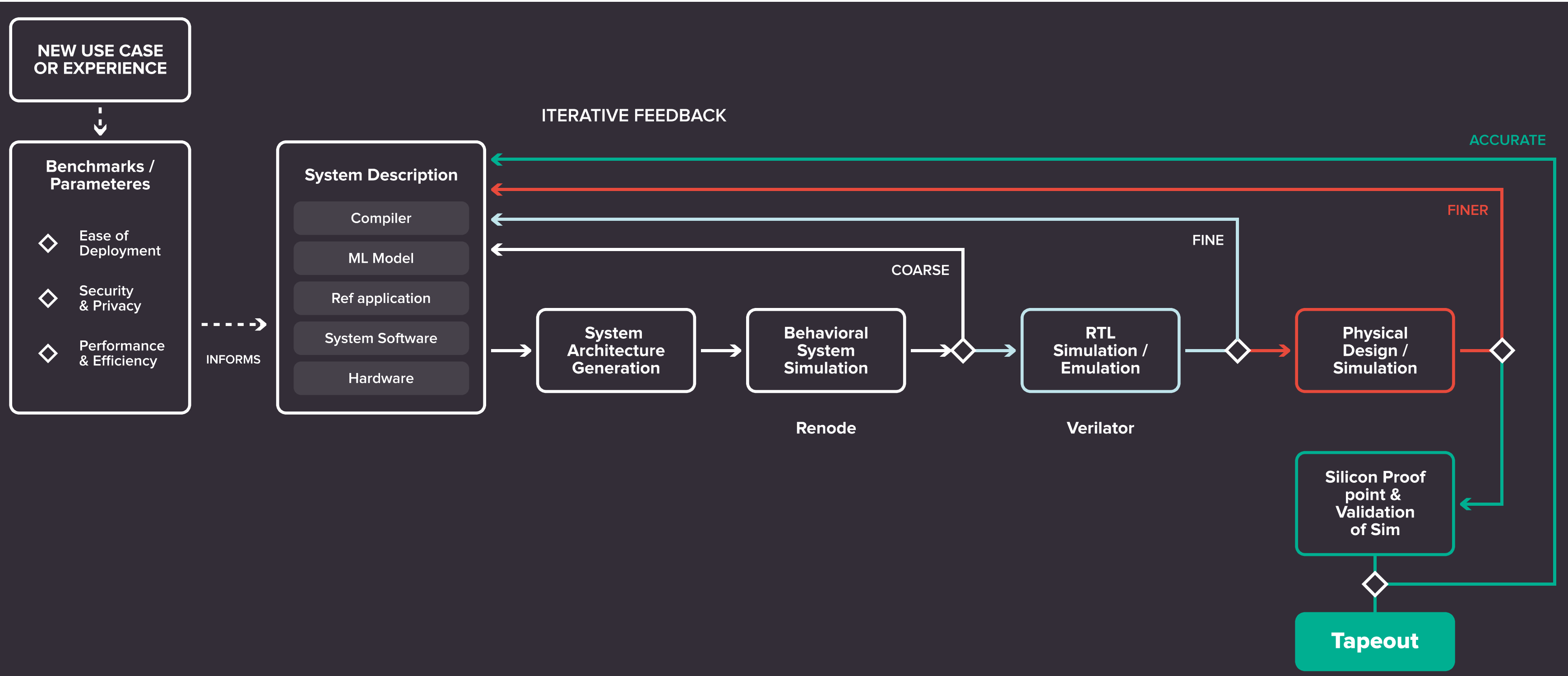


# PRE-SILICON DEVELOPMENT - TRADEOFFS

- FPGA emulation - limited and costly copies of hardware, fragile setups, also takes time
- HDL simulation testing - slow, translates into very long iterations which software people find hard to work with
- Behavioral simulation - fast, but normally leaves out various important aspects of your system

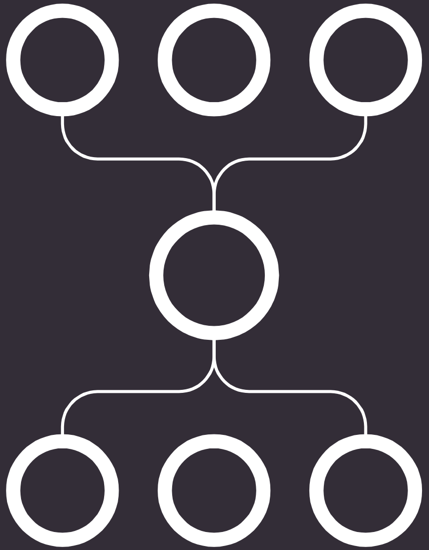


# SPARROW PROJECT EXAMPLE:
# GOOGLE RESEARCH'S SECURE ML CHIP



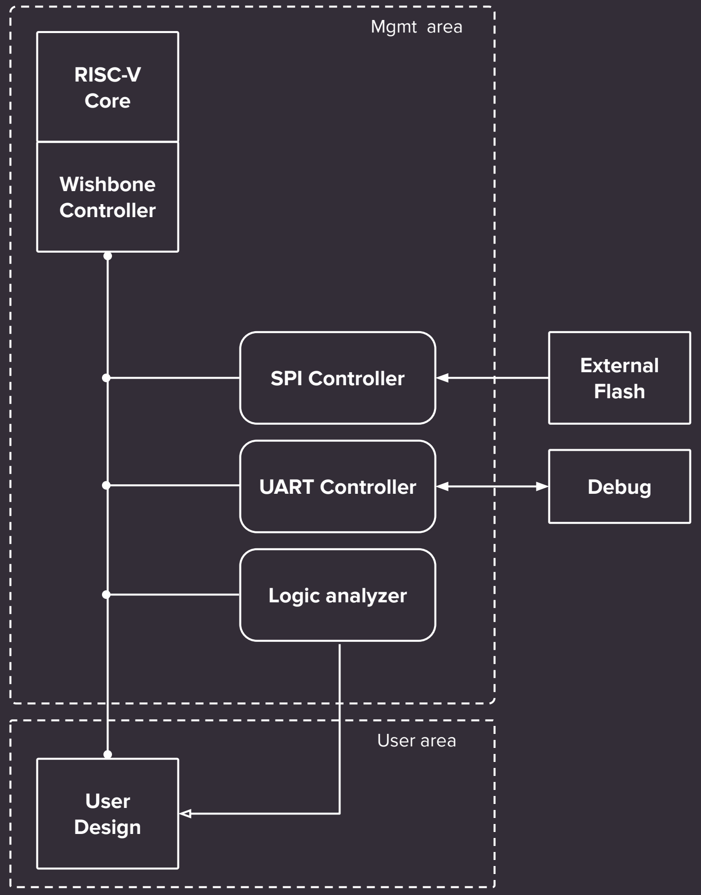
# PARTITIONING FOR CO-SIMULATION - BEST OF BOTH WORLDS?

- Systems typically consist of fast and slow changing elements, “knowns” and “unknowns”
- Combine benefits of behavioral and accurate simulation - for “known” bits, simulate behaviorally; for parts undergoing heavy development, simulate from HDL
- No need to “waste” time for HDL simulation of well tested and stable parts
- Always test the whole system, do not forget the integration with other parts



# PARTITIONING: SKY130 MPW CARAVEL MANAGEMENT SOC

- Fixed LiteX/VexRiscv core-based harness
- Wishbone bus connecting to the user area
  - SPI Flash controller
  - UART controller
  - GPIO controller
  - Logic analyzer
- User designs come in hundreds of flavors
- Maps great to a co-simulation setup!



# RENODE

- Open source simulation framework
- Runs unmodified software
- Integrates with a variety of debug tools
- Supports a human-readable, modular and extendible platform description format
- Both interactive and CI usage
- Supports co-simulation with HDL implementations of peripherals via Verilator
- Can simulate whole systems, or groups of systems
- Supports LiteX (and hence, Caravel) really well!

RENODE™

github.com/renode/renode

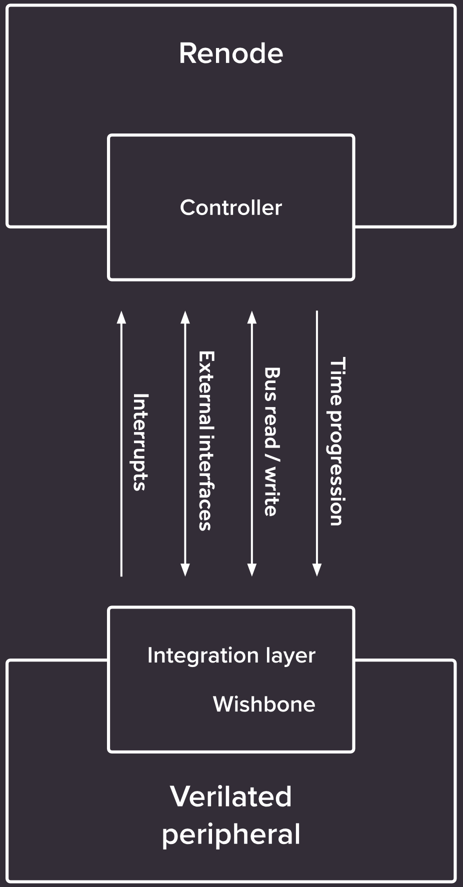
# VERILATOR

- Open source Verilog/SystemVerilog simulator
- Fast and flexible, widely recognized in the industry
- Creates multi-threaded C++ and system C modules out of the HDL definition
- Dynamic development by CHIPS Alliance members and broader community

github.com/verilator/verilator

# RENODE + VERILATOR CO-SIMULATION

- Renode provides an integration layer for Verilator
- Enables HW/SW co-simulation with verilated blocks
- No need to create Renode models of the blocks
- Allows for testing some HDL in the context of a larger system
- Already supports AXI4, AXI4Lite, Wishbone and APB3
- Allows to connect interrupts and external interfaces, like UART Rx/Tx lines
- Renode drives the clock of the connected design



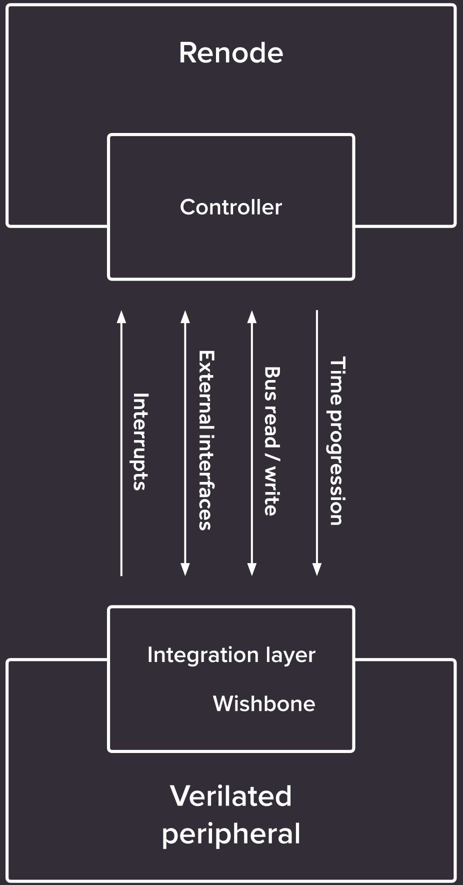
# RENODE + VERILATOR RECENT UPDATES

Recently added:

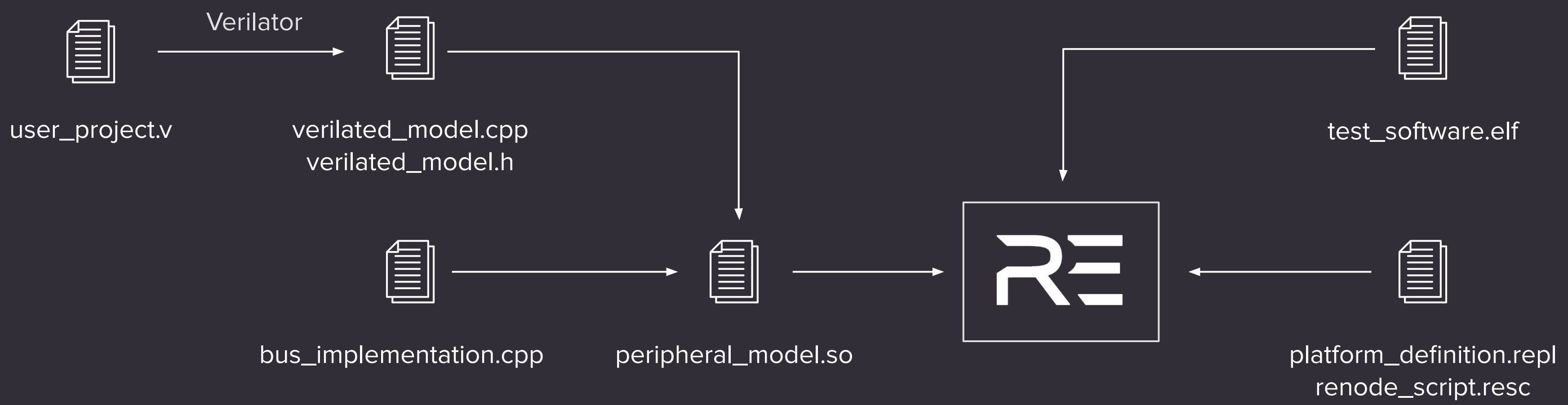
- Support for 64-bit transactions
- AXI4Lite initiator
- Reworked multi-bus support
- Improved AXI4Lite implementation
- More documentation

Planned soon:

- AXIStream
- New public CI with examples

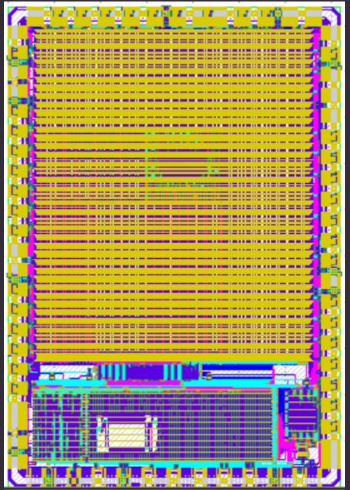


# CO-SIMULATION FLOW



# THE RENODE MPW TESTER

- Repository template with co-simulation testing automation
- Goal: allow designers to fork it and adjust to their designs
- End-to-end sample with AES Core submitted by Hanssel Morales to MPW-6 as a public project
- That project provided test software which made it easier to build an end-to-end example
- Most entries do not include any original SW!
- A better simulation platform can help encourage submissions to focus on that aspect, validating the system before the chips return from the fab



github.com/antmicro/renode-mpw-tester

github.com/Askartos/fossiAES

# CONNECTING TO RENODE

- VerilatorIntegrationLibrary - part of Renode
- Provides implementations of buses and other connection layers
- Requires user to connect signals with their design
- Take a look at github.com/antmicro/renode-verilator-integration/ for sample verilated projects and building infrastructure

```
#include <Vaes.h>
Vaes *top = new Vaes;
Wishbone* bus = new Wishbone;
//======================================
// Init bus signals
//======================================
    bus->wb_clk    = &top->clock;
    bus->wb_rst    = &top->reset;
    bus->wb_cyc    = &top->io_wbs_cyc_i;
    bus->wb_sel    = &top->io_wbs_sel_i;
    bus->wb_stb    = &top->io_wbs_stb_i;
    bus->wb_we     = &top->io_wbs_we_i;
    bus->wb_ack    = &top->io_wbs_ack_o;
    bus->wb_addr   =
      (uint64_t *)&top->io_wbs_adr_i;
    bus->wb_rd_dat =
      (uint64_t *)&top->io_wbs_dat_o;
    bus->wb_wr_dat =
      (uint64_t *)&top->io_wbs_dat_i;

//=======================================
// Init eval function
//=======================================
    bus->evaluateModel = &eval;
```

# RENODE SCRIPTING

- REPL (Renode Platform) files
  - Represents the CPU and peripherals from the Caravel mgmt SoC
  - Autogenerated from SoC description
  - Instantiates a VexRiscv CPU and peripherals
  - Allows e.g. for what-if experiments with memory sizes
- RESC (Renode Script) files
  - Sets up the simulation: platform, binaries, etc.
  - Allows the user to provide implementation of verilated blocks

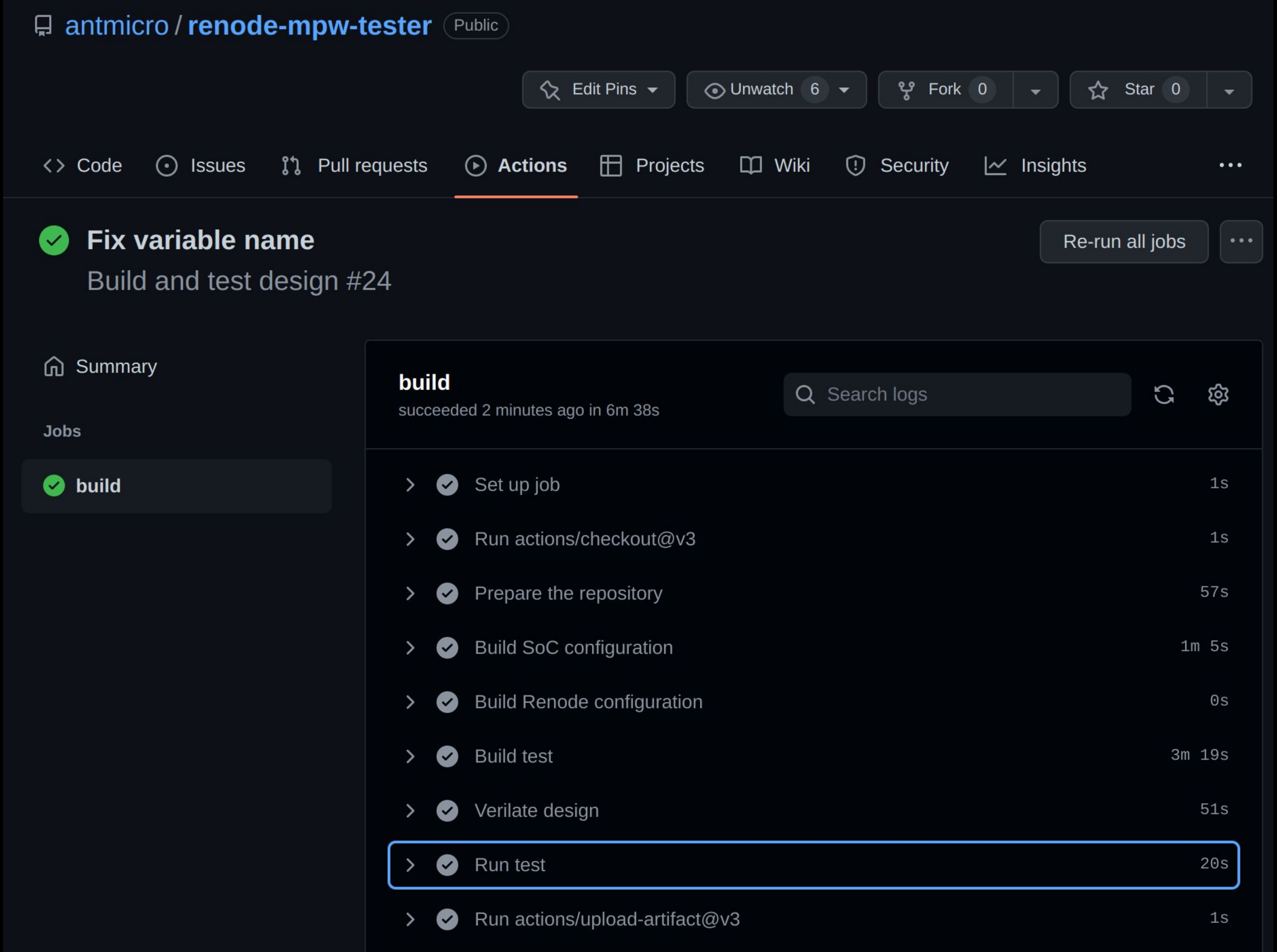
```
// REPL Renode Platform

[...]
user_space:
Verilated.BaseDoubleWordVerilatedPeripheral @
sysbus <0x30000000, +0x1000>
    frequency: 100000000
    useAbsoluteAddress: true
```

```
// RESC Renode Script

[...]
user_space SimulationFilePath @libVtop.so
```

# TESTING IN ROBOT FRAMEWORK

- Provides testing automation for Renode
- Executes Renode-specific commands to set up the simulation
- Starts the simulation and waits for certain events to collect the simulation results
- Text-file based, easy to modify, no need to recompile

```
[...]

Execute Command     include @design.resc
Create Log Tester   5000
Start Simulation
Wait For Line In Log
      “Write at ${ADDRESS}, value ${VALUE}
```



Fix variable name

Build and test design #24

build

succeeded 2 minutes ago in 6m 38s

```
Set up job                      1s
Run actions/checkout@v3         1s
Prepare the repository          57s
Build SoC configuration         1m 5s
Build Renode configuration      0s
Build test                      3m 19s
Verilate design                 51s
Run test                        20s
Run actions/upload-artifact@v3  1s
```

# RUNNING MPW TESTER LOCALLY

- Leverages a consistent layout of MPW projects
- Point to your design and test file
- We compile the design and run tests with the whole SoC

```
# download prerequisites, compilers etc
~/renode-mpw-tester# ./prepare.sh


# build the design, Renode configuration and test
~/renode-mpw-tester# ./build.sh -v aes -t aes_test -I Vaes.h -c Vaes


# run Renode
~/renode-mpw-tester# renode-run test artifacts/*robot
```

# LIVE DEMO

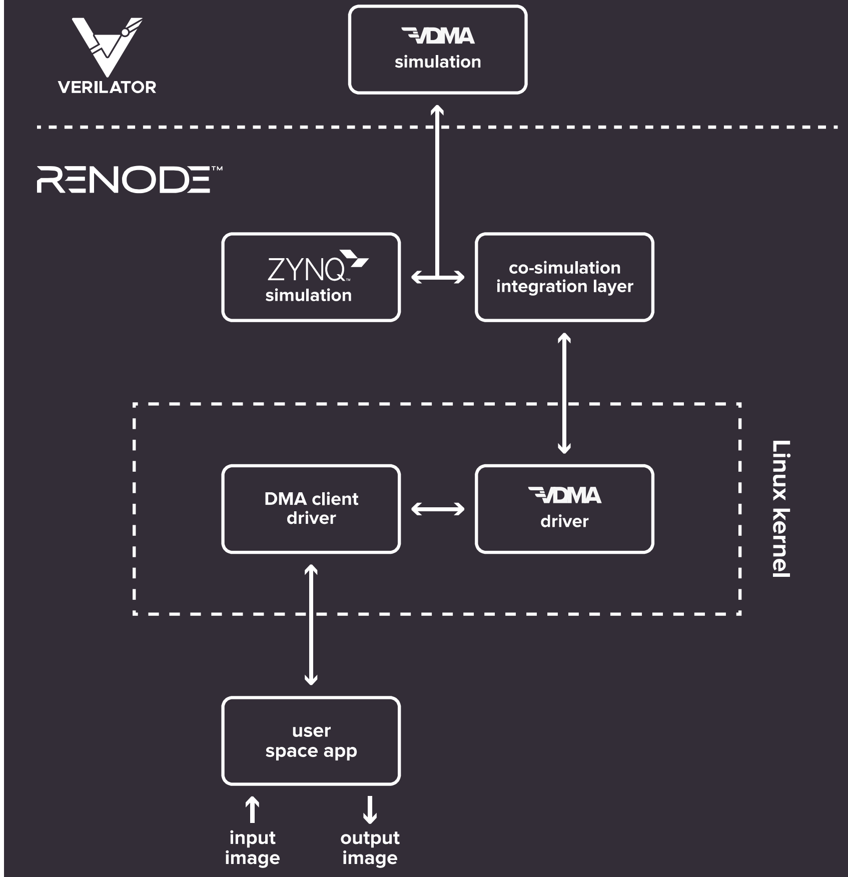
You can try Renode MPW Tester from your browser, using Google Colab!

# ANOTHER USE CASE: MIX OF PROPRIETARY AND OPEN SOURCE IP

- End-to-end RTL simulation is not even always possible - you need to have RTL!
  - ▫ You also need to be able to share it with software teams
- Software teams like speed anyway - just use co-simulation

# MORE ADVANCED USE CASE: FPGA SOCS

- The “known” / “unknown” partitioning is also found in FPGA SoC setups such as Xilinx Zynq or Microchip PolarFire SoC
- Co-simulation lets you test RTL against simulated full SoC system running e.g. Linux
- Example: Antmicro’s Chisel-based FastVDMA testing on Zynq, where the Verilated DMA is tested using Linux drivers and user space
- Good showcase of how this works in practice for a real-world, more advanced application
- Read more in our blog note, and see the test



# SUMMARY

- For the MPW shuttle submissions, we want to encourage a broader systems approach, testing concepts including software
- More collaboration between teams and projects using a unified approach
- The co-simulation flow scales to other scenarios as well, and real-world advanced chip design
- Looking forward to hearing about your use cases!

github.com/antmicro/renode-mpw-tester

antmicro

# THANK YOU
# FOR YOUR ATTENTION!