GVSoC: A Highly Configurable, Fast and Accurate Full-Platform Simulator for RISC-V based IoT processors

Paper review

Vinicius Peixoto

MO801 - Tópicos em Arquitetura de Computadores

Sep 17th 2025

Agenda

1. Introduction
2. Target architecture
3. Criticism
4. My first section
5. My second section

Introduction

Architecture-level simulation

- Different levels of abstraction
- Tradeoff between accuracy vs. simulation speed
- **Functional simulators**
  - Only simulate behavior (no uarch nuances)
  - Typically very fast, often inaccurate
  - Example: Spike

Architecture-level simulation

- **Timing simulators**
  - Model the uarch of the target (cache hierarchy, pipelines, branch pred., ...)
  - Much slower than functional simulation
  - **Cycle-accurate**: accurate cycle-by-cycle simulation
    - Example: Verilator (System Verilog → C++ executable)
  - **Instruction-level**: instruction-by-instruction simulation
    - Faster than cycle-accurate
    - Example: Gem5's O3CPU model

Architecture-level simulation

- **Timing simulators**
  - **Event-driven**: events instead of cycles
    - event = change of state in the system occurring at a certain point in time
    - Schedule events in a queue based on their latency
    - Jump directly to time of occurrence of next event
    - Skipping idle cycles → consistent savings in simulation time
    - Examples: SystemC + Transaction Level Modeling (TLM), RISC-V-TLM

State of the art

- Established timing simulation solutions lack flexibility (e.g. testing SoCs)
  - Cycle-accurate sims: slow, adding peripherals is cumbersome
  - Timing sims: faster, but still difficult to extend
- **Author's proposal**: a highly flexible, event-driven simulator targeted at full system emulation

Target architecture

PULP

- **P**arallel **U**ltra-**L**ow **P**ower platform
  - Open-source heterogeneous computing platform
  - RISC-V MCU (PULP SoC) + parallel programmable accelerator (PULP CL) + peripherals
  - Separate clock domains for easier workload tuning

PULP SoC

- FC (Fabric Controller): RISC-V processor
  - Manages the peripheral subsystem
  - Offloads compute-intensive tasks to the accelerator
  - Equipped with 256KiB - 2MiB of SRAM
    - Stores the code and application data
    - Paper calls it L2 (?)

PULP SoC

- FC (Fabric Controller): RISC-V processor
  - Comprehensive set of peripherals (JTAG, SPI, I2C, I2S, GPIOs, ...)
    - I/O DMA (called uDMA) for L2 memory ↔ peripheral data transfer
  - HyperBUS DDR interface
    - 8-bit high-speed bus for memory expansion (external DRAM, flash, ...)

PULP PL

- a

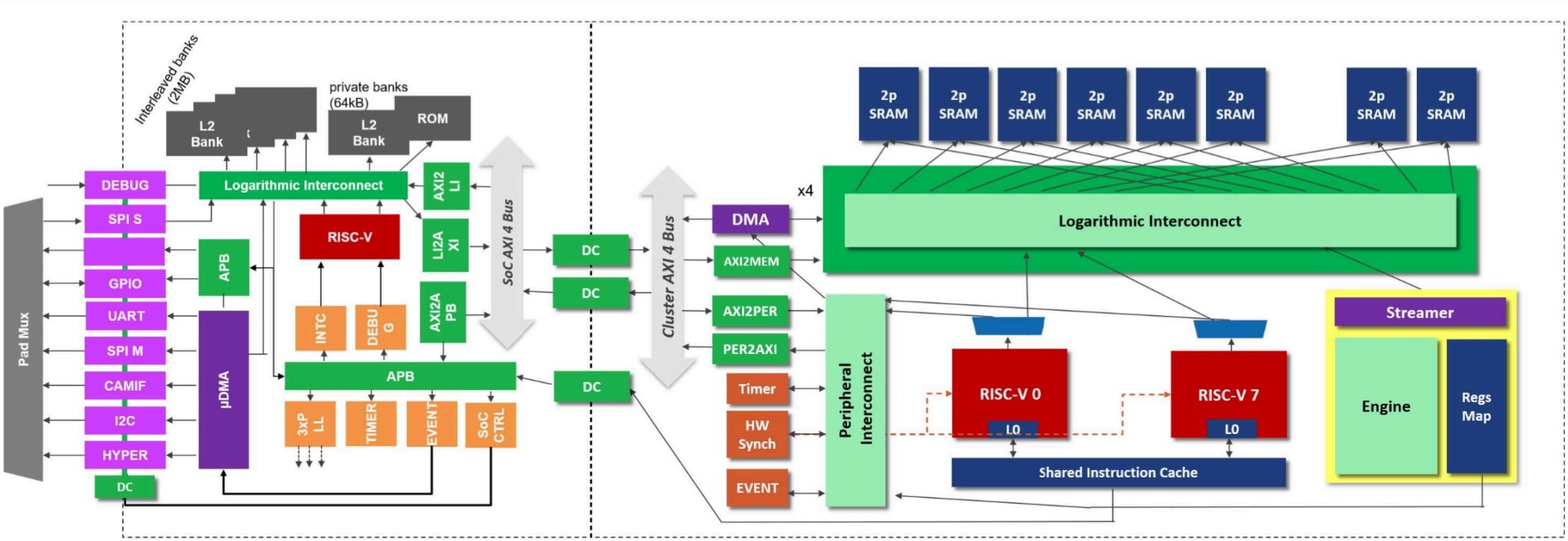

Figure 1: The PULP SoC architecture.

Criticism

- Low memory even for IoT devices, unfeasible for testing intensive workloads

My first section

For $f = \frac{dF}{dx}$ we know that

$$\int_a^b f(x)\,dx = F(b) - F(a)$$

See `https://en.wikipedia.org/wiki/Fundamental_theorem_of_calculus`

slide without a title

My second section

```
function heron(x)
    r = x
    while abs(r^2 - x) > eps()
        r = (r + x / r) / 2
    end
    return r
end

@test heron(42) ≈ sqrt(42)
```

Something very important