Co-Simulation and Scrum

Martin Schoeberl

Technical University of Denmark
Embedded Systems Engineering

October 21, 2025

Outline

- Agile is about test-driven development
- Co-Simulation
- More on Scrum
- Your project status
- Shorter lecture today (need to catch a flight)
- Nice lab exercise on co-simulation

Co-Simulation

- Write a *golden* model for your hardware
- Run the hardware and the model in parallel
- Compare the output
- Input is sometimes random, or constraint random
- Who is writing the reference model?
- Same error in model and implementation?

Reference Model

- A Java or Scala program (for Chisel)
- A C/C++ model
- Matlab/Simulink
- Is it cycle accurate?
- How to compare when not cycle accurate

Game of Live Example

- Conway's Game of Life
- Any live cell with two or three live neighbors survives.
- Any dead cell with three live neighbors becomes a live cell.
- All other live cells die in the next generation. Similarly, all other dead cells stay dead.

Game of Live Co-Simulation

- An exercise in the Java Introduction to programming
- The problem is highly parallel
- I will show you a Chisel (and Java) implementation
- FPGA version is extremely fast compared to the Java implementation
- It contains co-simulation
- https://github.com/schoeberl/game-of-live

Example: Processor Design

- Write an ISA simulation
- Who did Computer Architecture?
- You wrote a RISC-V ISA simulation
- You might do a RISC-V in January
- There, you should do the co-simulation

Cycle Accurate Example

- Lipsi processor
- Tiny processor core
- Showing it as Chisel motivation
- Has a cycle-accurate SW model in Scala
- Let us explore it now

Non-Cycle Accurate Example

- RISC-V core Wildcat
- Shall be a teaching reference
- Several implementations
  - ISA simulation (in Scala)
  - Pipelines: 3, 4, and 5 stages
  - Single cycle (= ISA simulation in Chisel)
  - Multi-cycle planned
- What to compare?
- How/when to compare?

Non-Cycle Accurate Example

- What to compare:
  - All data goes at some point through the register file
  - Just compare register file content
  - Not so much state (compared to memory content)
- When to compare:
  - One could track updates each clock cycle and advance only on a change
  - When to stop? Timeout if one of the two *hangs*
  - Compare register file content at the end
  - We had this in the Computer Architecture lab
- Explore it now

When is it Enough?

- How much do you need to test?
- How confident are you?
- We cannot cover all input possibilities
- Except in trivial cases
  - Parameters might help
  - Cover all cases for a 4-bit ALU
  - Assume the design also works for 32 bits
- Any other option?

Assertions in Chisel

- An Assertion statement states assumptions about a program
- You have seen `assert` in Scala in the first lab
- Can also be used in Chisel
- Assertion is checked during simulation time
- Syntax:

```
io.sum := io.a + io.b
assert(io.sum === io.a + io.b, "error
    message")
```

`require` vs `assert`

- Chisel `assert` checks value in simulation
  - Emits non-synthesizable Verilog
  - Using a Chisel expression that results in a `Bool`
  - Can also have a failure message
- Scala `assert` checks value at circuit construction
- Using a Scala expression that results in a `Boolean`
- Better use Scala's `require`
- `require` is used for input sanity checking

```
abstract class Fifo[T <: Data](gen: T, val
    depth: Int) extends Module {
  val io = IO(new FifoIO(gen))

  require(depth > 0, "Number of buffer
      elements needs to be larger than 0")
}
```

Formal Verification

- Simulation explores (only) some test cases
- Formal verification explores *all* possible behaviors
- Sounds a bit too good to be true, right?
- We will look into this next week

More on Agile Development

- Scrum is the base methodology for agile development
- Intended for small groups of developers
- Provides tools (lists)
- Defines roles
- Set of meetings
- Can also be adapted for single-person teams

Lists I

- Product backlog
  - Work to be done as a priority ordered list
  - Defines your product
  - Start with enough items to get the first and maybe the second sprint started
  - Will change as you develop (the agile part)
  - Priorities will change
  - Source for sprint
- TODO or Sprint backlog
  - Created at the start of the sprint from the Backlog
  - The work items to be done during the sprint
  - Team self-organized by pulling work from the list

Lists II

- In progress
  - Items moved from TODO into this list
  - Includes who is working on it
- Review
  - Optional, if needed
- Done
  - All work done moves there
  - The work done during a sprint, not the entire project

Scrum Board

- Visualize the Lists
- Fill at the sprint start
- Can have user stories in the Y axis
- Can be physical with Postits
- Or simple a Google doc
- Online tool, e.g., miro
- Reviewed at the sprint end
- Provides transparancy
  - Everyone sees what is going on
  - No hidden agenda

Scrum Roles

- Product Owner
  - Knows the domain
  - Makes decisions (e.g., product vision)
  - Accountable for value
  - Not a boss
- Scrum Master
  - Not a manager, a servant leader
  - Guide the team, the meetings
  - Asks questions
    - What can we change in our work?
    - What are our b igest obstacles?
- Developers
  - Do the work self organized
  - Move topics from TODO to In Progress
  - And then to Done

Daily Scrum Meeting

- Heartbeat of Scrum
- Each day, at the same time
- Maximum 15'
- Standup meeting
- Team and Scrum master meet
- Ask the following questions:[1]
  - What did you do yesterday to help the team finish the Sprint?
  - What will you do today to help the team finish the Sprint?
  - Is there any obstacle blocking you or the team from achieving the Sprint Goal?

[1] Sutherland, Jeff; Sutherland, J.J.. Scrum: The Art of Doing Twice the Work in Half the Time (p. 237). (Function). Kindle Edition.

Other Scrum Meetings

- Sprint planning
  - Decide on the goal of the sprint
  - Move tasks from backlog to TODO
- Sprint review
  - Demo what has been done—something that can be used
  - Feedback at the end of the sprint
  - Ideal with your customer
- Sprint retrospective
  - Reflect on the review
  - Find ways to improve the process
- Backlog refinement
  - Change the project/priorities
  - Can happen as you feel the need
  - Cleanup backlog
  - Shift priorities

What Do you Do?

- Having meetings?
- Using Scrum?
- Using a Scrum board?
- Or a TODO list?
- Tell me

Summary

- Co-simulation as an advanced version of testing
- Some more talking about Scrum
- Today's lab brought to you by Javad
- Now it is your turn to present