

The Problem

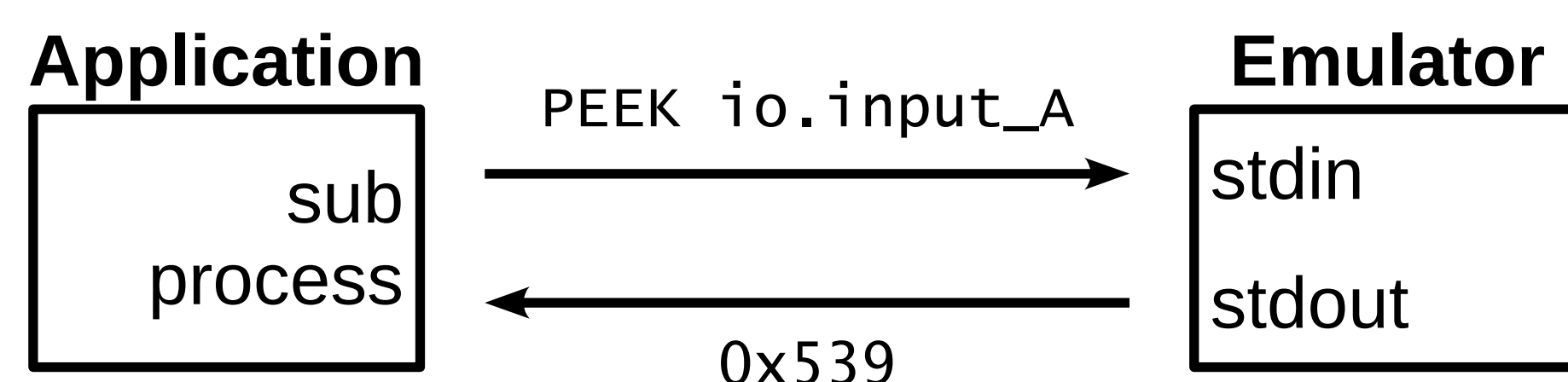
- Designing hardware is hard, and debugging is even harder, especially with only RTL
- More functionality layered on top of an emulator can be beneficial
 - ... but not everyone wants to program at the efficiency level (C++) used by the emulator
 - ... and even the C++ emulator structure / interface isn't set in stone
 - ... and new emulation technologies are being developed (DREAMER, FPGAs) with even more different structures
- which means software depending on emulation becomes **brittle and inflexible**

A Solution

- The root of the problem: **coupling** from upper-level code directly to lower-level technology (the emulation platform)
- So **decouple** through a **standard interface**!

The Chisel Debug API

- Communicate with an emulator process using stdin/stdout
 - Console universally supported, intrinsically cross-platform and cross-language, require minimal (usually no) dependencies
- Line-based, human-readable text protocol
 - Intuitive, easy to understand and use
- Access circuit elements by Chisel name
 - Simple yet robust
- Values transmitted in hexadecimal
 - No issues with data width / endianness



The Protocol

API Information

Provides details about the emulator platform

- `get_emulator_host`: returns emulator name
- `get_api_features*`: returns a list of supported features

Basic Functionality

Allows interaction with the emulated circuit (combinational logic treated as always running, overwritten outputs clobbered)

- `clock [cycles]`: step circuit
- `reset [cycles]`: step circuit in reset
- `wire_peek [name]`: read wire
- `mem_peek [name] [idx]`: read memory
- `wire_poke [name] [val]`: write wire
- `mem_poke [name] [idx] [val]`: write memory

Circuit Information

Gives basic information about the circuit

- `list_wires`: returns all accessible wires
- `list_mems`: returns all accessible memories
- `wire_width [name]`: return wire bitlength
- `mem_width [name]`: return memory bitlength
- `mem_depth [name]`: return memory size

Snapshotting

Saves circuit state in emulator to be restored later, useful for back-stepping

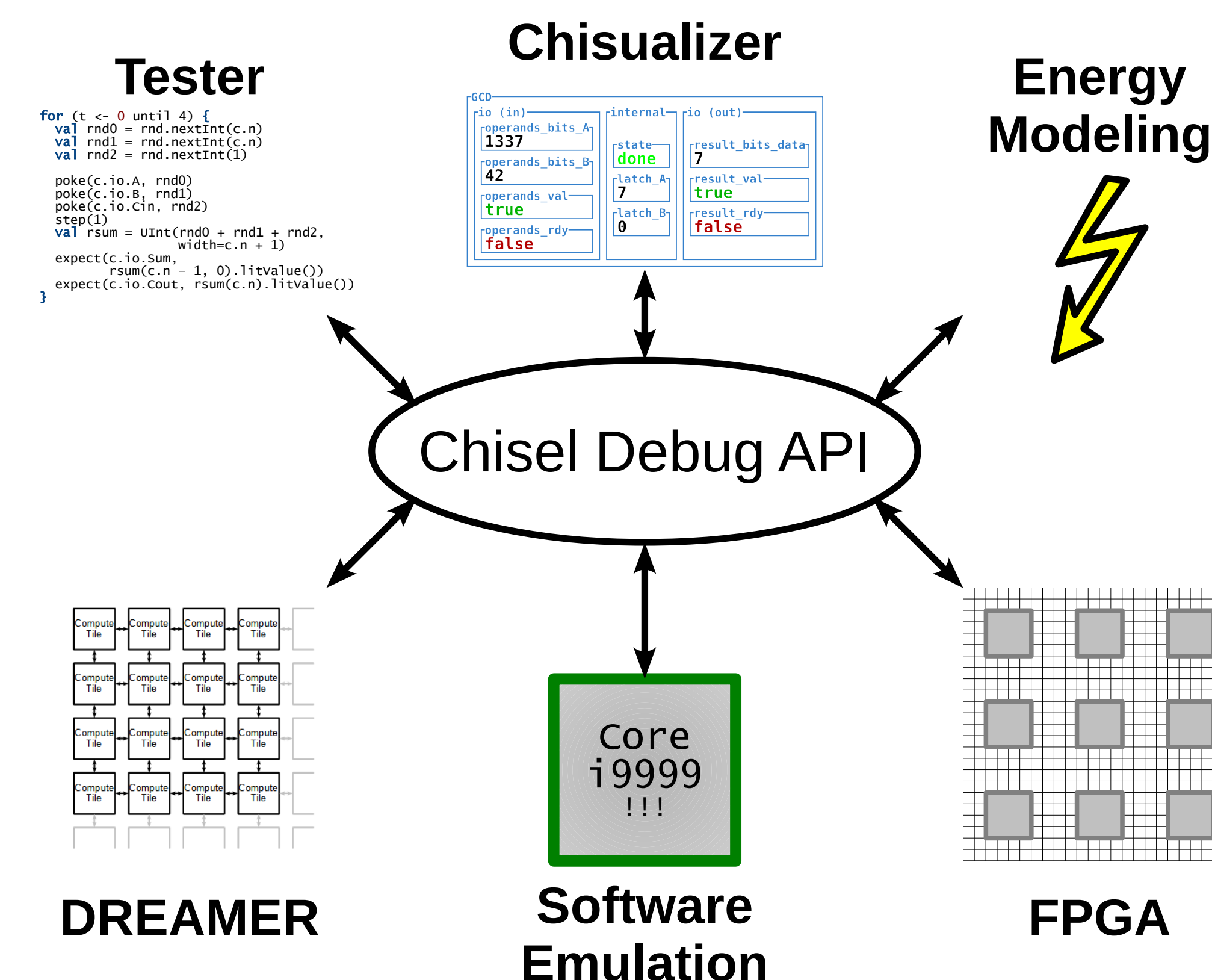
- `referenced_snapshot_save* [name]`: saves the current circuit state as 'name'
- `referenced_snapshot_restore* [name]`: restores the circuit state named 'name'
- `referenced_snapshot_dump* [name]`: dump snapshot state for persistence across emulation runs

* commands in beta, subject to change

Current Uses

Application-level Software

- **Tester**: a Scala-based Chisel RTL testbench
- **Chisualizer**: a block-diagram style visualizer
- **Energy modeling**: use emulation to generate switching activity counters



Emulation Platforms

- **Software**: Chisel compiled to optimized C++,
- **DREAMER** (in progress): a mesh-of-processors based emulation platform
- **FPGA** (upcoming): direct compilation to FPGA instrumented with circuit state access

Continuing Work

- Formal specifications and documentation
- Improving the existing specification through feedback while it's still possible
 - Hopefully no APIv6 problem
- Designing an extensions system
 - ... and adding features like logic analyzer (clock until condition), data type information, optimized / bulk access protocol, and more
- Cleaning up implementation corner cases