Benchmark Description

Ten real-world (System)Verilog designs with safety properties were collected from Open Source projects. Only immediate assertions and assumptions were used in the designs under test. One benchmark (`picorv32-check`) used assertions for safety properties intermixed with the actual design in the same modules. All other benchmarks used a formal test-bench with assertions and assumptions that simply instantiates a synthesizable Verilog design.

VexRiscv-regch0-{15,20,30} These three benchmarks are taken from riscv-formal [1]. They check the VexRiscv processor core for register file consistency for the instruction retired in cycle 15, 20, or 30. The only nontrivial BMC step for those benchmarks are step 17, 22, and 32 respectively, and therefore this numbers are used as bounds for these three benchmarks.

picorv32-{check,pcregs} These two benchmarks are taken from PicoRV32 [2], another RISC-V processor. The former checks the invariants encoded in safety properties in the core itself (`make check` in the PicoRV32 codebase) and the latter used the PicoRV32 RVFI port to check register file and PC consistency between instructions. These benchmarks simply get harder with increasing bounds. In the benchmark bounds of 30 and 20 cycles were used respectively.

ponylink-slaveTXlen-{sat,unsat} These benchmarks are taken from PonyLink [3], a point-to-point communication core that uses a single wire (or differential pair) in half-duplex mode. The benchmarks are part of a proof that shows that the PonyLink core will never keep transmitting indefinitely, regardless of the initial state. The first one tries to prove that the core is bound to stop transmitting after 228 cycles (which fails in is thus SAT), and the second one tries to prove that the core is bound to stop transmitting after 229 cycles (which succeeds and is thus UNSAT). The bounds for this benchmarks are 230 and 231 respectively, with only the last cycle being nontrivial.

zipcpu-{busdelay,pfcache,zipmmu} These three benchmarks are taken from ZipCPU [4]. They are various proofs for safety properties of auxiliary components in the ZipCPU system. The ZipCPU repository contains more proofs similar to them. This three have been selected because they are the longest running proofs in the repository at the time we checked. The bounds used for this benchmarks are 100, 100, and 30 respectively.

References

1. Wolf, C.: riscv-formal. `https://github.com/cliffordwolf/riscv-formal`
2. Wolf, C.: picorv32. `https://github.com/cliffordwolf/picorv32`
3. Wolf, C.: PonyLink. `https://github.com/cliffordwolf/PonyLink`
4. Gisselquist, D.: zipcpu. `https://github.com/ZipCPU/zipcpu`