EE290C Homework 1

Readings

1) J. Bachrach, DAC'12
2) A. Izraelevitz, ICCAD'17
3) A. Wang, DAC'18

Chisel Simulation

The DAC 2012 paper discusses how total simulation time is a function of compilation time, the number of simulation cycles, and the amount of time per cycle. Today's version of Chisel no longer uses the C++ emulator discussed in the paper, but instead has a common testing API that can use multiple backends:

1) A FIRRTL interpreter that directly executes FIRRTL circuits
2) Verilator[1] backend that compiles FIRRTL -> Verilog, and then generates and compiles C++ executable models of the Veriliog.

Here is some real benchmarking data from testing a Chisel design[2] with the FIRRTL interpreter and Verilator backends. The design is parameterized, so smaller (low complexity) and larger (high complexity) versions of the design are tested. Different parameterizations of the design design are simulated with both backends for a different number of cycles.

Test Cycles	Complexity	Interp. Time (ms)	Verilator Time (ms)
10	1	25	1350
10	51	275	1426
10	101	305	1460
100	1	23	1309
100	51	316	1355
100	101	521	1453
1000	1	41	1363
1000	51	1460	1420
1000	101	3147	1500
10000	1	228	1670
10000	51	19510	1625
10000	101	X[3]	1782

[1] Verilator is an open-source alternative to VCS. Both work by generating and compiling C++ executable models of Verilog source code. There is also a VCS backend that you can use if you have VCS on your system.

[2] The design is an FIR filter. "Complexity" in this table is the number of taps.

1) Which backend would you use to run unit tests for a small block? Which backend would you use to run integration tests for a whole RISC-V core? Explain your answer.

2) You can compute a rough estimate of the "frequency" of a simulation by dividing the number of simulation cycles by simulation time. You shouldn't include compilation and other start-up costs in this compilation time, so don't forget to subtract that out from the total simulation time. Use the 10 cycle simulations as a rough estimate of compilation time and other startup costs. Roughly, what frequency does the interpreter and Verilator achieve for each design complexity?

3) FPGAs can be a useful way to accelerate simulation. Let's say that building and deploying a design for FPGA takes at least 15 minutes and will run at 10 MHz. Approximately how many cycles do you need to simulate for FPGA emulation to be worthwhile for this design?

Bootcamps

You will upload your completed bootcamp notebooks to bcourses. Please upload both the notebook file (`.ipynb`) and an HTML version (in the notebook, click File->Download As->HTML). Be sure to evaluate your cells!

[3] This simulation took too long, use the 1000-cycle datapoint instead.