SPREE: Microarchitectural Exploration on FPGAs

Peter Yiannacouras, Jonathan Rose, and J. Gregory Steffan
Department of Electrical and Computer Engineering, University of Toronto
{*yiannac,jayar,steffan*}*@eecg.toronto.edu*

Field Programmable Gate Arrays (FPGAs) are now large enough to fit complete systems including processors and peripherals, hence they can be used to perform studies of processor architecture and microarchitecture. In particular, we can investigate in accurate detail the impact of microarchitectural decisions such as pipeline depth, implementation of functional units, latency of functional units, and datapath organization. However, the process of creating the register transfer level (RTL) description of a design needed by FPGA CAD tools is a very laborious one.

We have developed a system for rapidly generating an RTL description of a processor from a simple textual description. The user can select which components compose the datapath, connect the components together, and choose to generate either pipelined or unpipelined control.[1] The control for the datapath is generated automatically, so the user need not worry about the timing, hazard detection, and other bookkeeping imposed by changes to the datapath. The output of our system is a Verilog description of the entire microprocessor system which is readily synthesizable (currently using Altera's Quartus II CAD software), and can also be simulated by RTL simulators (such as Modelsim). Included in the system is a compiler infrastructure for generating the memory contents of the system from a C/C++ application. The RTL generation combined with the compiler infrastructure provides a rapid exploration environment for studying microarchitectural trade-offs.

The design and fabrication costs for a competitive ASIC design are continually increasing. Meanwhile, FPGAs are growing in size and improving in cost/performance, leading an increasing number of designs to target FPGAs for more than just prototyping. Many of these designs include microprocessors, hence it is natural to incorporate microprocessors onto the FPGA itself. Even the use of FPGAs as a simulation environment for microprocessors is indicative of an increase in migration of microprocessors onto FPGAs. FPGA makers and other 3rd party vendors now provide processor cores designed specifically for such an implementation, referred to as *soft processors*. In the context of an integrated FPGA design, a soft processor can offer many compelling advantages over a fixed-core processor. Our hypothesis is that a soft processor can be automatically customized to match the area, power, and speed requirements of a given design. As a starting point, our *Soft Processor Rapid Exploration Environment* (called SPREE) provides facilities for *real* area, speed, and power analysis of FPGA-based microprocessor architectures (see Figure 1).

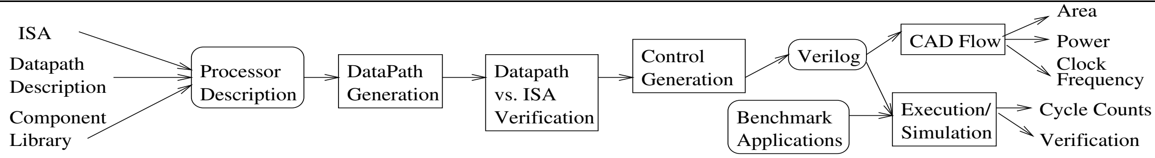

Figure 1: Block Diagram Overview of SPREE

We have performed three preliminary studies. The first is a study of the shifter implementation. We experimented with a serial shifter, a barrel shifter, and a shifter which uses the multiply unit to do the shifting. We find that since most modern FPGAs include dedicated (hard) multipliers, implementing the shifter using the multipliers is better in both speed and area. The second study focused on cycles per instruction (CPI). The generator currently emits only unpipelined control, thus we've experimented with different CPIs on unpipelined architectures and observed the effect on wall clock time and area. Finally we performed a study on ISA subsetting. SPREE supports custom subsetting of the MIPS I instruction set, thus one can customize a processor to run only a specific application and achieve up to 50% area savings and an 18% speed increase.

[1] To avoid generating custom compilers (in the short term), we require all processors use a subset of the MIPS I ISA.