Comet: a RISC-V Core Synthesized from C++ Specifications
Simon Rokicki, Joseph Paturel, Olivier Sentieys
Univ Rennes, Inria, IRISA - Rennes, France
{simon.rokicki, joseph.paturel, olivier.sentieys}@irisa.fr

Introduction

Designing the hardware of a processor core as well as its verification flow from a single high-level specification would provide great advantages in terms of productivity and maintainability. In this work, we highlight the gain of starting from a unique high-level synthesis and simulation C++ model to design a processor core implementing the RISC-V ISA.

Development flow

What you simulate is what you synthesize

HW Design & Verification
C++ Model
SW Validation
HLS
RTL Simulation
RTL Synthesis
Physical Design

Pipelined architecture and simulator

while true do
  ftodc_tmp = fetch();
  dctoex_tmp = decode(ftodc);
  extomem_tmp = execute(dctoex);
  memtowb_tmp = memory(extomem);
  forward = forwardLogic();
  stall = stallLogic();
  if (!stall) then
    ftodc = ftodc_tmp;
    dctoex = dctoex_tmp;
    extomem = extomem_tmp;
    memtowb = memtowb_tmp;
  end
  if (forward) then
    dctoex.value1 = extomem.result;
  end
end

Development Roadmap

Current Project Status

Current project features:
- Specification of 5-stage pipelined RISC-V core for rv32i or rv32im
- Configurable cache architecture
- Support for Zephyr OS
- Simple SoC for Artyx-7 FPGAs
- Simple to use and to modify
- Already used in several research projects

Conclusion

The Comet project highlights the benefits of using HLS to develop CPU cores since it significantly reduces development and debugging time. Experimental studies show that the generated core has an area comparable with cores developed using standard development flows.

References


Get started at: https://gitlab.inria.fr/srokicki/Comet