FPGA Implementation of a RISC-V processor

Problem Statement

Michael Mortensen

25/09/2017

General overview

As the use of FPGAs increases, the importance of optimized processors for FPGAs will increase. This thesis will attempt to produce a soft RISC-V processor optimized for speed and area for the Altera DE2-115 architecture. Working with a softcore CPU is very different from working with a hardcore CPU. A softcore CPU is implemented entirely on the logic primitives of an FGPA board whereas a hardcore processor is implemented on dedicated hardware. This makes a softcore processor more flexible and easy to customize and thus optimize for the specific situation in which it will be used. A hardcore CPU on the other hand can typically achieve much faster processing speeds since they are not limited by fabric speed.

The RISC-V (Reduced Instruction Set Computing - "five") ISA, as the name implies, follows the RISC principles. The general idea is to have a small set of simple and general instructions, rather than a large set of complex and specialized instructions. The term "reduced" describes the fact that the amount of work any single instruction accomplishes is reduced compared to the amount of work done by a complex instruction of a typical CISC CPU.

RISC-V was developed in the Computer Science Division of the EECS Department at the University of California, Berkeley. It was originally created to support research and education but is now becoming an industry standard instruction set and is therefor becoming increasingly popular in education as well as industry. This is why the RISC-V ISA will be the focal point for this thesis.

Other well known RISC ISAs include: MIPS, ARM, ARC, AVR, and SPARC.

Specifications and limitations

A working 5-stage pipelined 64-bit version of the RISC-V ISA will be designed that will support memory-reference instructions, arithmetic-logical instructions, and conditional branch instructions. The CPU will contain hardware for hazard detection to support a no-operation instruction (nop) and forwarding hardware to operations in the execution stage for resolving data hazards.

Floating-point capabilities and exception handling is however not going to be implemented due to the limited scope of the project.

As the project evolves, added elements to the CPU will be analyzed for their contribution to the overall performance and area. The methodology used to measure performance will be investigated and suitable benchmarks will provide basis for performance comparisons while software tools for the Altera DE2-115 FPGA board will provide information related to resource use.