Analysis Of Benchmarks On The Enzian Research Computer

Noah Berner

Introduction

It is widely known, that the rate at which memory speed increases is smaller than the pace at which microprocessors improve. This has been true for several decades and is unlikely to change drastically in the future. Thus, memory bandwidth is expected to be an even bigger bottleneck in computation than it already is today.

A trend in recent years has been to look further than traditional multiprocessing and to explore the possibilities of stream processing on FPGAs. Intel, for example, acquired Altera, a leading FPGA manufacturer, in 2015 and NVIDIA has included Tensor Cores in their Volta architecture, which are streaming processors for matrix multiplication.

It is unlikely that FPGAs will replace traditional CPUs as their size limits their capabilities and running complete programs on FPGAs would make the architecture too complicated and thus ruining the benefits seen when running small kernels on them. They are more likely going to serve as separate accelerators which take over tasks that benefit from being offloaded to FPGAs, much like vector processors are used in today's architectures.

What is missing today however, is the possibility to compile an existing program in a high level language to a bitstream for a computer that houses an FPGA and a CPU such that the performance of the overall program is increased when comparing it to a traditional CPU-only run. The Enzian research computer [1] is the perfect platform to study the behaviour of such a hybrid architecture, as it allows for fast reconfiguration of the hardware and thus is much more versatile than other systems.

Description

This thesis will analyse the performance of the two SPEC-benchmarks lbm [2] and cactusADM [3] on an FPGA.

First, the benchmarks will be analysed by running on a traditional CPU to get a baseline and to plan the design of the FPGA layout. In a next step, using HLS, the kernels of the benchmarks will be compiled to an FPGA bitstream and analysed on an FPGA. Once this analysis is done, a further improvement would be to get the whole program to run on a CPU and FPGA hybrid architecture and analyse the performance and data exchange between the CPU and FPGA, to further improve the design. If this analysis goes well, one could consider automating the process for similar types of programs, such that the process of writing a program for a CPU and FPGA hybrid architecture is simplified.

Objectives and goals

The main goal of the thesis is to get closer to a solution for easy programming of a CPU and FPGA hybrid architecture, as this type of computer is likely to spread in the future.

The design process and analysis of running the two benchmarks on an FPGA and a CPU seperately will provide an insight on what methods to use when tackling the conversion of high level code to a bitstream for FPGAs.

Using these insights, it will likely be possible to come up with a concept of running the benchmarks on a hybrid architecture such as Enzian, which will be useful for studying the data sharing between CPU and FPGA.
Once the benchmarks are running on Enzian and using the patterns found in the data sharing, one will hopefully be able to automate the process of compiling any similar program to a bitstream for a hybrid architecture.

References

[1] Systems Group, ETH Zürich (n.d.). *Why Enzian?* Retrieved from `http://www.enzian.systems/why-enzian.html`

[2] Thomas, P. (23 January 2008). *470.lbm: SPEC CPU2006 Benchmark Description File.* Retrieved from `https://www.spec.org/cpu2006/Docs/470.lbm.html`

[3] Malcolm, T. (16 August 2011). *436.cactusADM: SPEC CPU2006 Benchmark Description.* Retrieved from `https://www.spec.org/cpu2006/Docs/436.cactusADM.html`

Timeline and To-Do List

- **February 2019:**
  Hand in proposal,
  Start analysis of benchmarks on CPU.
- **March 2019:**
  Finish analysis of benchmarks on CPU,
  Use HLS to compile kernel into bitstream for FPGA,
  Analyse performance of kernel on FPGA and tweak it.
- **April 2019:**
  Finish analysis of kernel-only performance on FPGA,
  Try running the whole benchmark on CPU and FPGA.
- **May 2019:**
  Port the running program to Enzian and figure out data-transfer between CPU and FPGA,
  Start automation process.
- **June 2019:**
  Work on automation.
- **July 2019:**
  Finish automation.
- **August 2019:**
  Finish writing thesis,
  Hand in thesis,
  Hold presentation.
- **Write thesis parallel to analysis and other work.**