Reconfigurable Computing Lab Summer Coop Projects 2024

1. Hardware Accelerators with burst memory patterns and DMA for L2 cache FPGAs -- Dr. Lesley Shannon

Background: Soft-processors are an important part of the FPGA ecosystem. Their potential for offering high levels of configurability enables their use in systems ranging from simple micro controllers to complex many core systems. Additionally, soft-processor systems on FPGAs offer the flexibility to researchers to perform system level computer architecture. The majority of soft-processors are vendor specific or closed source, preventing their use on a wide range of FPGAs and limiting the choice available to researchers. However, Dr. Shannon's group has developed an open source RISC-V based processor called Taiga that is designed to run on both Intel and Xilinx fabrics.

Objective: This project will have a student develop multiple hardware accelerators with burst memory patterns (including an "artificial" accelerator that can be configured to generate numerous types of burst memory patterns). These accelerators will be used to extend Taiga's open source infrastructure, as well as quantify performance benefits and bottlenecks. This project provides an opportunity to learn about the low level structure of computer systems and memory system architecture with a primarily hardware focus; however, there will also be some low-level software design required.

Skills needed:

- Programming competency in either VHDL or Verilog HDL (will learn SystemVerilog during coop);
- High-level language programming skills (preferably C or C++);
- Scripting language knowledge (preferably Python and tcl)
- Competency with FPGAs and their CAD tools (either Xilinx or Altera);
- Completed ensc350 (ensc452 would be an asset)
- Designed a MicroBlaze/NIOS/ARM based SoC on an FPGA would be very helpful but is not required.
- Experience with the Linux kernel and/or device driver design would also be an asset.

Please visit my website for further information on my research interests and course offerings: http://www.ensc.sfu.ca/~lshannon/.

2. Automated Verilog Code Generation with Generative AI -- Dr. Lesley Shannon

This project provides an exciting opportunity for a summer student to engage in impactful research at the intersection of natural language processing and hardware design automation. The student will be an integral part of a collaborative team, contributing to advancements in the field and gaining valuable experience in both machine learning and hardware design.

Abstract: This project embarks on a transformative journey to revolutionize digital system design by harnessing the formidable capabilities of Generative AI, with a particular focus on Large Language Models (LLMs). The overarching objective is to propel the automation of code generation for hardware, a shift poised to significantly reduce human errors and enhance design efficiency. The project aims to explore and enhance the capabilities of Large Language Models (LLMs) in automating hardware design through the generation of high-quality Verilog code. The research is based on the fine-tuning of pre-existing LLMs on Verilog datasets compiled from GitHub and Verilog textbooks.

Project Tasks:
1. Dataset Consolidation: Gather and consolidate open-source Verilog code to create the largest training corpus for Verilog code used in training LLMs.
2.Model Fine-Tuning: Fine-tune five different pre-trained LLMs models with varying parameter counts on the created Verilog corpus. This step will result in the generation of new fine-tuned open-source models specialized for Verilog code generation.
3. Problem Set Design: Design a set of Verilog coding problems with varying difficulty levels, along with corresponding test benches to evaluate the functional correctness of the generated Verilog code.
4. Performance Evaluation: Evaluate the efficacy of the fine-tuned models against state-of-the-art general-purpose LLMs, including ChatGPT and PALM2. Perform an expanded evaluation with a new set of Verilog design challenges.
5. Training Set Augmentation: Augment the training set with Verilog textbooks and provide detailed results on the performance of models tuned on this augmented dataset.

Expected Outcomes:
- Creation of the largest training corpus for Verilog code.
- Generation of fine-tuned open-source LLM models specialized for Verilog code generation.
- Design and implementation of a set of Verilog coding problems for functional correctness evaluation.
- Comprehensive evaluation results showcasing the performance of fine-tuned models against general-purpose LLMs.

Skills Required:
- Understanding of hardware description languages (HDLs) like Verilog.
- Experience with HDL Synthesis Process and Tools
- Programming proficiency in Python.
- Familiarity with machine learning concepts, particularly with language models.

3. Updates for OpenHW CVA5 RISC-V designs for FPGAs -- Dr. Lesley Shannon

Dr. Shannon's research group created the Taiga RISC-V processor, specifically designed for FPGAs. This has been adopted as the CVA5 processor for OpenHW. We are continually updating the processor, adding functionality and working to improve its performance. The projects listed below are some examples of work we are continuing to do. The necessary skills to work on these projects include:

- Programming competency in either VHDL or Verilog HDL (will use SystemVerilog during coop);
- High-level language programming skills (preferably C or C++);
- Scripting language knowledge (preferably Python and tcl)
- Competency with FPGAs and their CAD tools (either Xilinx or Altera);
- Completed ensc350 (ensc452 would be an asset)
- Designed a MicroBlaze/NIOS/ARM based SoC on an FPGA would be very helpful but is not required.
- Experience with the Linux kernel and/or device driver design would also be an asset.

Please see a list of potential projects listed here:

1) ***Reducing the physical footprint of a RISC-V CPU by reducing the size of the register file***
Description: When software is compiled, operators are assigned registers for inputs and outputs. The compilation may have false dependencies that limit data throughput as the same register may be assigned as an operand in multiple statements while not actually having a dependency relationship. Runtime register renaming is performed by the CPU hardware at execution time to remove these false dependencies to improve execution parallelism, and throughput (i.e. instructions per cycle). While our group's RISC-V processor, Taiga (also known as CVA5 in the OpenHW Group repository), supports register renaming for its 64-entry register file, we want to investigate the impact of reducing the physical register file to 32 entries, while still supporting register renaming. Ideally, this reduction in resources will have no measurable impact on performance while reducing design complexity.

2) ***Designing a minimal footprint RISC-V processor for verification research***
Description: Designing systems that are fully verified as functional is extremely challenging due to the multitude of logical paths through processor logic. As part of our work, we are working to create a framework and abstraction for formally designed heterogeneous systems. As part of this work, we would like to perform a preliminary evaluation of our existing work on a simple processor- something much simpler than our existing Taiga RISC-V processor. As such, we are looking to have a student implement a simple fixed pipeline RISC-V processor that will be used as a preliminary test case for our formal verification work.

3) ***Designing a configurable and scalable run time performance monitoring framework***
Description: Runtime performance monitoring is an extremely active field of research. It is also a crucial aspect of efficient execution on heterogeneous systems. We are currently working to build a framework for composable heterogeneous systems, using our RISC-V Taiga core (also known as OpenHW's CVA5 core) as the basis of the system. We wish to create a runtime performance monitoring framework that can be configured to obtain appropriate analytics and eventually feed this data to the OS. The focus of this project would be on the hardware design with the development of some limited firmware (extensible to OS work if the project is successful).