Project Proposal
Hardware/software test framework for garbage collected BRAM

Martha Barker (mmb2295)

February 22, 2019

1 Introduction

In this project, I will implement a test framework for a garbage collected memory I have designed as part of work done by Martha Kim's and Stephen Edwards' groups at Columbia University on a Haskell to Hardware Compiler [1]. The compiler simplifies custom accelerator design by allowing designers to write functions Haskell and compile them into SystemVerilog. Like the original Haskell applications, the generated circuits have an immutable memory model. While this enables a lot of parallelism, it also means the circuits run through memory very quickly. Automatic garbage collection recycles memory locations that can no longer be accessed to better utilize memory resources.

2 Design

I will use the FPGA on the SoC to implement compiled circuits with garbage collected memory and the ARM processor monitor and alter state.

2.1 Hardware

I have simulated the memory design using Verilator, but not yet synthesized it or ran it on an FPGA. The hardware portion of this project will be synthesizing the memory and running it on an FPGA along with several compiled applications and the logic that connects the two.

2.2 Software

I will implement the test framework using the ARM processor. The software will be able to set configuration variables for the memory design including its size and the garbage collection threshold. It will then be able to stop the application to check actual state against an expected state. It will be able to read in the

contents of BRAM as well as buffers in the application circuit and modify their contents before restarting the application circuit.

3 Evaluation

I will use this framework to first test correctness of the memory system in isolation and with real application circuits. I will then evaluate performance metrics such as percentage of cycles spent doing garbage collection for different configurations. Finally, I will identify bottlenecks to be improved in future versions.

4 Milestones

I will have three intermediate milestones before the project presentation.

4.1 Milestone 1

1. Synthesize memory system design using Altera Quartus
2. Design interface between hardware and software sections of framework
3. Run a simple example on the FPGA

4.2 Milestone 2

1. Design a comprehensive test plan for at least 2-3 interesting example circuits
2. Test correctness of memory system with application circuits

4.3 Milestone 3

1. Design performance evaluation plan for at least 2-3 interesting example circuits
2. Run performance tests with application circuits and evaluate results

5 References

[1] Richard Townsend and Martha A. Kim and Stephen A. Edwards. From Functional Programs to Pipelined Dataflow Circuits. In Proceedings of Compiler Construction (CC), pages 76-86, Austin, Texas, February 2017.