

The Trireme Platform: Delta Processor Demo

November 29, 2022

1 Introduction

The Delta Processor is an RV64im system based on an Out-of-Order core connected to a main memory through a cache hierarchy. It is meant to provide an example of using the Trireme Platform's Out of Order CPU core. Figure 1 shows a high-level block diagram of the Delta Processor. Parameters for the L1 and L2 caches are not exposed to the top level test bench, but can be directly edited in the **delta_processor** module located in the rtl directory. Parameters for each L1 cache are provided to the cache hierarchy as a list of concatenated 32-bit integers. The main memory connects an FPGA BRAM to the L2 cache.

2 Running a Simulation

These directions assume you have a version of Modelsim (or Questasim) correctly installed. The simulation scripts included with the Trireme Platform require that Modelsim is added to your PATH variable. By default, Modelsim installs itself into your home directory. To add it to your PATH, add something like the following to your .bashrc file:

```
# For Modelsim Altera Starter Edition installed with Quartus II 15.0
export PATH=$PATH:~/altera/15.0/modelsim_ase/bin
# For Modelsim Altera Starter Edition installed with Quartus Prime 18.1
export PATH=$PATH:~/intelFPGA_lite/18.1/modelsim_ase/bin
# For Questasim Intel FPGA Edition installed with Quartus Prime Pro 21.3
export PATH=$PATH:~/intelFPGA_pro/21.3/questa_fse/bin
```

To run the top level Delta Processor test bench, execute the run_test script from the modelsim directory. The run_test script will simulate one or more verilog test benches who's module names are provided as an argument.

```
./run_test tb_delta_processor
```

When you run the simulation script, the verilog modules are compiled and the simulator runs the specified test bench(s). Listing 4 in the Appendix shows the default output for the **tb_delta_processor** test bench.

3 Demo Programs

The modelsim/binaries directory stores binaries used in test bench simulation. One compiled example binary named gcd64_262144.vmh is included. This is a rv64i binary that executes "bare-metal" and sets the stack pointer to 256k (262144).

Simulation of the included test bench should complete in under 60 seconds. Changing the PROGRAM parameter in the tb_delta_processor.v file in the rtl directory will change which program the test bench executes. The example_programs directory includes the source C code for the gcd64_262144.vmh application, in addition to the elf file output by the RISC-V toolchain, a binary disassembly (.dump) and a copy of the .vmh file.

The program was compiled with the following compiler wrapper command. Additional programs can be compiled by downloading the Trireme Platform's software stack and providing different C code as input.

```

PROGRAM=gcd64_262144
./tireme_gcc -o applications/binaries/$PROGRAM \
  --vmh applications/binaries/$PROGRAM.vmh \
  --dump applications/binaries/$PROGRAM.dump \
  --raw-binary applications/binaries/$PROGRAM.bin \
  --ram-size 262144 --start-addr 0 --link-libgloss nosys_tireme64 \
  --stack-addr 262144 --stack-size 65536 --heap-size 65536 \
  --64-bit \
  applications/src/$PROGRAM.c

```

4 Appendix

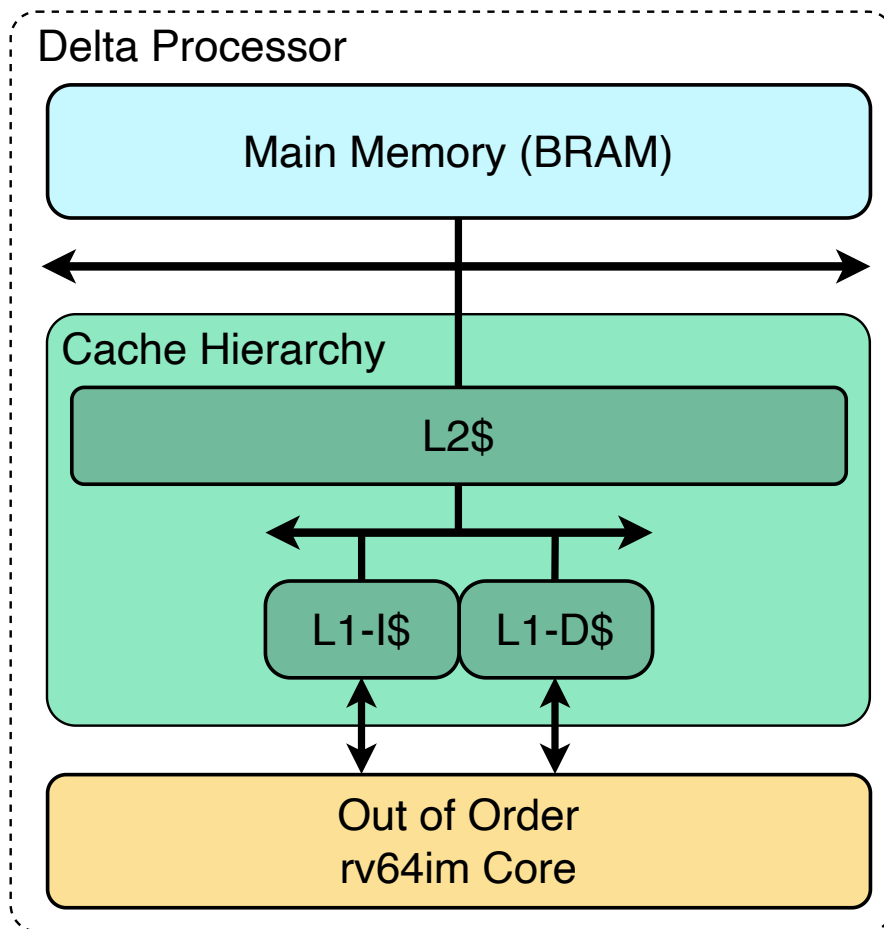


Figure 1: A high-level block diagram of the Delta Processor.

```

# *****
# * The Trireme Platform - Delta Processor Test Bench
# *****
#
# Use the PROGRAM parameter to execute different program binaries
# Loading ./binaries/gcd64_262144.vmh
#
#
# Run Time (cycles):          10476
# Dumping reg file states:
# Reg Index, Value
#      0: 0000000000000000
#      1: 00000000000000b0
#      2: 0000000000040000
#      3: 0000000000000000
#      4: 0000000000000000
#      5: 0000000000000000
#      6: 0000000000000000
#      7: 0000000000000000
#      8: 0000000000000000
#      9: 0000000000000010
#     10: 0000000000000010
#     11: 0000000000000010
#     12: 0000000000000000
#     13: 0000000000000000
#     14: 0000000000000010
#     15: 0000000000000010
#     16: 0000000000000000
#     17: 0000000000000000
#     18: 0000000000000000
#     19: 0000000000000000
#     20: 0000000000000000
#     21: 0000000000000000
#     22: 0000000000000000
#     23: 0000000000000000
#     24: 0000000000000000
#     25: 0000000000000000
#     26: 0000000000000000
#     27: 0000000000000000
#     28: 0000000000000000
#     29: 0000000000000000
#     30: 0000000000000000
#     31: 0000000000000000
#
# tb_delta_processor Test Complete!
#
#

```

Listing 4: Default output for the tb_delta_processor test bench.