ZHOU, QI

+44 07710 799432 ⋄ +86 185 5254 1785

zhouqi008012@gmail.com ⋄ Q.Zhou-24@sms.ed.ac.uk

EDUCATION

University of Edinburgh, Computer Science (BSc Hons) 2019 - 2023
Average grade 74 (A3) (equivalent GPA 4.0)

WORK EXPERIENCE

Codeplay June 2022 - Sept 2022
`https://www.codeplay.com/`

- Worked on `ComputeArota` (`CA`), a toolkit implementing heterogeneous cross-platform computing.
- Added `RV32` and `Zfh` extension (half float) support to the simulator in `CA`, such that `RV32` or `Float16` instructions can be generated by `CA` and simulated using `SPIKE`.
- Learnt about `Sollya`'s `fpminimax` function that computes a polynomial approximation for floating point operations, then investigated some test failures caused by floating point precision issues.
- Implemented a faster and more intuitive replacement of `SPIKE` for `CA` using `RISCV-64 QEMU` with a Linux operating system, where the host (client) communicates with simulator (server) using sockets.

Teaching assistant Sept 2022 - May 2023
For these courses I achieved high grade, therefore being qualified for teaching support roles:

- **Computer Architecture and Design** (INFR10076)
  - Demonstrator in workshops to help students with the 5-stage pipeline `RV32IM` core coursework:
    1. Get familiar with Xilinx PYNQ FPGA board through toy examples.
    2. Implement ALU and register file according to `RV32IM` specification.
    3. Integrate speculative result forwarding into the pipeline.
    4. Analysis critical path and optimize accordingly, for example, execute fast `int32` multiplication by utilizing FPGA's builtin `int16` multiplier in parallel.
- **Compiling Techniques** (INFR10065)
  - The coursework is a compiler from `Python` to `RISC-V` assembly using a "MLIR-lite" framework: `xDSL` `https://xdsl.dev/`.
  - Contributed in coursework template design and bug fixes.
  - Answer questions for students during workshops.

Asteria Sept 2020 - Feb 2021
`https://www.asteria-space.com/`

- Came up with possible solution for attaching a camera on a single-board computer which will be launched into space.

PROJECTS

SIMD Support for LLVM MLIR Presburger library - In progress
`llvm/llvm-project/blob/main/mlir/lib/Analysis/Presburger/Simplex.cpp`

- This library performs overflow-checked multiplication and addition on small matrix.
- Compute `23` bits or `52` bits integer using FPU could be fast, because:
  - Mantissa part of single and double precision floating point number is exactly `23` and `52` bits,
  - Exploits fused-multiply-add,

    - Sufficient for small data, not likely to trigger `SIGFPE` and redirect to slow `BigInt` algorithms,
    - Floating point overflow and imprecision checking is convenient.
- Discovered bugs in `llvm-mca` when it analysis `FMA` throughput on `zen3`.
- Analyzed performance characteristics using `google/benchmark` and `perf`.

Ukulele Tuner

- A coursework where a group of 9 work together on an assistive robotics project.
- Best Commercialization Award!
- Fully working handheld device, consists of a 3.5 inch touch screen, a microphone, a stepping motor and a raspberry pi, then connected to each other with a custom PCB and assembled in a 3D printed chassis.
- The process involves converting a string pluck recording into its corresponding vibrational frequency using Fast Fourier Transform. The resulting frequency is then compared to the expected frequency and motor rotates according to a lookup table.

Lorenz Attractor

`https://github.com/AOIDUO/LorenzAttractor/`

- A animated figure of Lorenz attractor written in Haskell.
- Demo: `https://homepages.inf.ed.ac.uk/wadler/fp-competition-2019/#(16)`.

Turing Machine Emulator

`https://github.com/AOIDUO/RegisterMachineEmulator/`

- A turing machine emulator written in `Python`.
- Partially supports marco.
- Utilized parser provided by `xDSL` to recognize instruction source code in BNF style.

SKILLS

- Practiced with `Java`, `Agda`, `Haskell`, `Python`, `Shell`, `Verilog` and `C++`,
- Also experienced in `GNU/Linux`, `git`, `GDB`, CI, many other toolchains and software testing toolkits,
- Capable of collaborative work on big repository,
- Learnt *Programming Language Foundations in Agda*, familiar with formal verification and type theory.