Dongyang Wu

wudongya@usc.edu | www.linkedin.com/in/william-wu-dongyang/

EDUCATION

- **University of Southern California (USC)** — Los Angeles, CA
  Master of Electrical Engineering; GPA: 4.0/4.0 — 08/2022-05/2024
- **The Chinese University of Hong Kong, Shenzhen (CUHKSZ)** — Shenzhen, China
  Bachelor of Computer Science and Engineering; GPA: 3.4/4.0 — 09/2018-06/2022

SKILLS

Programming Languages: Python, C/C++, Verilog, VHDL, SQL, MIPS
EDA Tools: Virtuoso, QuestaSim, Xilinx Vivado
Protocols: TCP/IP, USB, SPI, AXI, PCIe, MOESI
Tools: UNIX, Linux, Git, Makefile, CUDA

PROJECTS

- **Tomasulo Out-of-Order CPU** — 06/2023-08/2023
  - Designed CPU with Out-of-Order Execution and In-order Commitment.
  - Implemented Branch Prediction Buffer(BPB) and Return Stack Address(RAS) for speculative execution beyond branches.
  - Implemented BRAM-based Copy Free Check-pointing(CFC) with RRAT for recovery from path misprediction.
  - Implemented Store Buffer(SB), Store Address Buffer(SAB), Reorder Buffer(ROB), 2-stage Dispatch Unit, Free Register List(FRL) and Issue Unit(IU).
- **PCIe Physical Layer Design** — 06/2023-08/2023
  - Designed physical layer components for PCIe 2-lane system.
  - Implemented Elastic Buffer with Primed Method to achieve Clock Domain Crossing by adjusting Skip Ordered Set.
  - Implemented Deskew Buffer to eliminate skew between lanes.
- **512-bit 6T SRAM Array Design** — 01/2023-05/2023
  - Designed and drew layout of 1-bit SRAM cell, row/column decoder, sense amplifier, write driver, precharge circuit, latch and flip-flop with Cadence Virtuoso and GPDK 45nm.
  - Achieved the Read SNM of 210 mV and Write SNM of 395 mV by proper sizing with VDD=1V while minimizing the size of 1-bit SRAM cell.
  - Integrated components into 4 8x16-bit SRAM banks to construct a 512-bit SRAM Array with the area of 2208 nm^2 in 2.6 Ghz (cycle time=0.4 ns).
  - Measured the power consumption with Spectre, in which the average consumption for reading is 21.2 fJ, the average consumption for writing is 342 fJ and leakage is around 20 fJ.
  - Validated the correctness of all aforementioned components with vector file in Spectre and cleared DRC and LVS errors.
- **Extensible Asynchronous SNN Accerlator** — 01/2023-05/2023
  - Designed extensible asynchronous SNN accelerator in SystemVerilog with SystemVerilogCSP library on a 5x5 filter and 25x25 ifmap with stride=1.
  - Implemented fork-join computation module that can calculate part of the overall computation with pre-configured parameter for extensibility.
  - Integrated computation modules with two memory modules, which contain filter map and ifmap respectively, on a mesh network.
  - Verified correctness of computation module and the accelerator separately with timestep=2 in QuestaSim.