

ERIC KRAUSE

8815 SW 36th Ave • Portland OR, 97219 • (541) 337-5788
eric@sauerkrause.org • www.sauerkrause.org

TECHNICAL SKILLS

Proficient Languages: Verilog, C, Assembly (ARM, z80, MIPS, PicoBlaze)

Familiar Languages: C++, SystemVerilog, Python, Bash Script, L^AT_EX, AHK, Java (for Android Development)

Hardware: RTL design and debug, digital design and SoC/embedded system design and debug with FPGAs. Experience using test equipment in a laboratory setting to verify and debug digital designs.

Software: Relatively OS agnostic; equal experience with Mac/Windows/Linux. Professional experience with Xilinx Toolchain (ISE, Lab Tools, EDK, SDK). Limited experience with GNU tools (Make, GCC) and VCS (git).

RELEVANT COURSEWORK

- Microprocessor Design
- Superscalar Processor System Architecture
- Parallel Computing Architecture
- SoC Design with FPGAs
- Embedded Systems with FPGAs
- Embedded Software Programming

EDUCATION

M.S., Computer Engineering **GPA: 3.86/4.00**
Portland State University, Portland Oregon *2011-2013*

Post-Bac, Electrical/Computer Engineering **GPA: 3.88/4.00**
Portland State University, Portland Oregon *2010-2012*

B.A., Environmental Studies **GPA: 3.80/4.00**
University of Oregon, Eugene Oregon *2005-2009*

WORK HISTORY

Intel Corporation Hillsboro, Oregon

Data Center Group Intern – 2011-2013

- RTL Development of an FPGA-based DDR3 Memory Error Injector (MEI) interposer.
- Python utility development for automating testing of server platform error injection and verification using MEI.
- Optimization of FPGA RTL designs to improve timing and enhance features as per customer needs.
- Testing of runtime memory error injection and detection on Intel server platforms.

Portland State University Portland, Oregon

IEEE Tutor (Computer Engineering) – 2011-2013

- Programming/Algorithms (Verilog, C, C++ Python)
- Digital Circuits (Logic Circuits, Boolean Algebra, Programmable Logic Devices, Simulation)
- Digital Systems (Synchronous Design, Timing Analysis, State Machines, FPGA Synthesis, Microprocessors)

Free Geek Portland, Oregon

Platform Deployment Tech (Volunteer) – 2013-Current

- Computer disassembly/assembly
- Peripherals test and debug
- OS/Software installation

HONORS & AWARDS

- **Intel Professional Recognition** <http://db.tt/0nqPo3Yf>
 - Awarded for "excellent creative and technical abilities", after identifying, debugging and repairing a critical functionality in key product in 2013.
- **Ford Family Foundation Scholarships (Undergraduate and Graduate)** <http://www.tfff.org/>
 - One of fewer than 100 Scholars inducted into the prestigious Ford Family Foundation Undergraduate Scholarship program in 2005.
 - Awarded second, additional Graduate Scholarship following academic successes at University of Oregon and Portland State University.
- **Etta Kappa Nu (HKN)** <http://web.cecs.pdx.edu/~eta/>
 - IEEE Honors Society. Limited to top 25% of Department.
- **Golden Key International Honour Society** <https://www.goldenkey.org/>
 - Member since 2008

REFERENCES

Available upon request.

ACADEMIC PROJECTS

- **FPGA-Based Color-Tracking Robot (Verilog, PicoBlaze)** <http://tinyurl.com/color-fpga-bot>
 - 1st place in class competition for best term project (SoC design with FPGAs), awarded by course professors.
 - Designed and built autonomous color-seeking robot controlled by SoC in FPGA.
 - Controlled via custom Verilog code and Picoblaze soft processor in FPGA of Spartan-6E development board
 - Computer vision via CMUCam4 camera and environmental awareness via proximity/light sensors.
- **Branch Predictor/BTB simulation (C)** https://github.com/rattboi/flanders_ece486
 - Accuracy of solution ranked within top 3 best designs in Computer Architecture course when tested against traces taken from unknown processor.
 - Development of branch predictor/branch target buffer simulation in C.
 - Design of N-way associative cache, Return Address Stack, and Fully-Associative cache simulation blocks in C++
- **Microprocessor Cache Simulation (Verilog)** <https://github.com/ekrause/0xBEEFA55>
 - Source code from this project has been incorporated into Microprocessor Design course materials by professor
 - Designed and coded split-level L1 data/instruction cache simulation in Verilog
 - Project read in trace data from text file, performed cache simulation, and displayed hit/miss statistics.