(847)-345-2180
jimppalomo@gmail.com

Jim Palomo

Computer Engineering Student

linkedin.com/in/jim-palomo
jimpalomo.github.io

Education

University of Illinois at Chicago (UIC) **Expected Graduation: May 2022**
Bachelor's of Science in Computer Engineering Chicago, IL

- Cumulative GPA: 3.85/4.00
- Focus: Computer Systems, Data Science & Engineering, Digital Systems & VLSI Design

Experience

Teaching Assistant [TA] **Aug — Dec 2021**
University of Illinois at Chicago Chicago, IL

- TA for ECE 366 — Computer Organization, topics on MIPS, Single/Multi Cycle CPU, Memory, Cache, and CPU Pipelining
- Prepared Office Hours to guide students on homework and projects
- Graded 9 homework assignments and gave feedback to 50+ students
- Aided in the development of 4 term projects: MIPS Assembly, Python Assembly Simulator, 8-Bit CPU, Cache Simulator
- Administered the resources section by organizing notes on software (Python, MIPS Assembly, MARS) and hardware (CircuitVerse) tools

Computer Architecture Intern **Jun — Aug 2019**
University of Illinois at Chicago Chicago, IL

- Communicated with an engineering professor and a Ph.D. student on project deadlines
- Developed on an open-source simulation platform for computer system architecture called gem5
- Established connections between different CPU chip-sets such as ARM & x86 with memory controllers, caches, and interconnects

Projects

Fault, Controllability, and PRPG Simulation — Python, CSV **Oct — Dec 2021**

- Implemented Normal, Fault, Controllability, Monte-Carlo, Pseudo-Random Pattern Generation (PRPG) simulation on various circuit test benches
- Created a text-based UI to navigate through 9 program functions including 14 circuit bench files
- Enabled program to scale and to be used for academic research by obtainable .csv file
- Generated data through 42,560 fault simulations on 1064 faults using PRPG to detect faults and to find the maximum controllability values based on the circuit level

Cache Simulator — Python **Nov 2020**

- Led a team to simulate four different types of cache configurations: Simple (1 block, block size 64 B), Direct Map (4 sets, block size 16 B), Fully Associative (4-way, block size 8 B), and Set Associative (2-way 4-set, block size 8 B)
- The simulator in Python takes in hexadecimal machine code inputs from MIPS and determines the program mode according to the provided user input
- Programmed the simulator to display detailed step-by-step cache information: memory breakdown, LRU (least recently used) specifics, and hit or miss results

DIVVY Data Hashing — C++, CSV, Valgrind, GNU Make **Apr 2020**

- Developed an application that hashes station and trip data from the DIVVY bike-sharing company
- Created a hashmap with separate hash functions for over 1500 trips and 580 bike IDs
- Added multiple commands: search by station id, abbreviation, trip id, bike id, nearby stations, and similar trips

Skills

Languages Python, C, C++, MIPS Assembly, ARM Assembly, SystemVerilog
Software Tools Linux, Git, SSH, Catch, ModelSim, MARS, Quartus, Cadence, Wireshark

Awards & Honors

Dean's List: Fall 2020 — Present