Justin T. Wilford

Email jwilford@usc.edu
Website jwilford.me
GitHub github.com/JTWilford

EDUCATION

Master of Science, Computer Engineering **Aug 2022**
Bachelor of Science, Computer Engineering and Computer Science **May 2021**
Viterbi School of Engineering **GPA: 3.69**
University of Southern California – Los Angeles, CA
Notable Coursework:

MOS VLSI Circuit Design	Computer Systems Organization
Intro to System on Chip	Intro to Digital Circuit Design

SKILLS

Programming:	Python, C++, C, Java, Javascript, C#, Verilog, VHDL
Software:	Xilinx Vivado, Xilinx ISE, Cadence Virtuoso, QuestaSim
Hardware:	FPGA, RTL Design, Transistor-Level Design/Layout, Embedded Systems

EXPERIENCE

Student Researcher – ISI Reconfigurable Computing Group **December 2020 – Present**
University of Southern California: Information Sciences Institute – Marina del Rey, CA
- Researching bitstream generation for Xilinx FPGAs
- Working with open-source tools such as Prjxray
- Utilizing Xilinx Vivado for reference designs

Web Developer – Viterbi Conversations in Ethics **September 2019 – Present**
USC Viterbi School of Engineering: Engineering Writing Program – Los Angeles, CA
- Designing and maintaining an online magazine (https://vce.usc.edu/)
- Student-run publication emphasizing ethics in engineering fields
- Website utilizes WordPress with custom plugins, widgets, and themes

Student Software Developer **October 2017 – January 2021**
University of Southern California: Information Technology Services – Los Angeles, CA
- Full-Stack developer for web-based applications
- Led a team of 3 to 4 student developers to implement projects using the MEAN stack with Docker Swarm and Jenkins for deployment
- Designed a robust RESTful API utilizing a microservice architecture

PROJECTS

Pong for FPGA – More details at http://jwilford.me/projects/pong-for-fpga/
- Collaborated with another student to develop a Pong clone on an FPGA
- Created a simple and modular video game engine using Verilog and Xilinx ISE, with the intent to repurpose for other games
- Utilized a state machine to handle different functions of the game engine, such as collision detection, player movement, and VGA output
- Implemented and demonstrated the game running on a Diligent Nexys 3 FPGA

10-bit Multiply-Accumulator Schematic and Layout – More details at http://jwilford.me/projects/10-bit-mac/
- Designed a 10-bit Multiply-Accumulator at the transistor level
- Utilized Cadence Virtuoso for both schematic and layout
- Incorporated transistor sizing to maintain setup and hold times, and to decrease overall delays