Masaad Khan

Bay Area, California | Austin, Texas | Remote
(408) 202-8830 • mak4668@utexas.edu • github.com/mkhan825 • linkedin.com/in/masaad-khan

EDUCATION

The University of Texas at Austin **August 2019 – Present**
B.S., Electrical and Computer Engineering, 3.56/4.00 Overall GPA
Relevant Coursework: Verification of Digital Systems, Computer Architecture, Digital Logic Design, Operating Systems

WORK EXPERIENCE

Intel **May 2022 – August 2022**
Xeon SoC Pre-silicon Verification Intern (Full-time) *Remote*

- Designed a methadology and the intiial uploader scripts to help the adoption of a new internal Python database API
- Converted a Perl post-processing checker to Python and uploaded the large log files to the new database to increase performance
- Enhanced the System Verilog test bench used to verify the SoC in different configurations and automated its generation

Tenstorrent **January 2022 – May 2022**
Design Verification Intern – RISC-V CPU Team (Full-time) *Austin, TX*

- Compiled and modified Google's open-source System Verilog *riscv-dv* random instruction generator to create tests
- Constrained tests to help build testbenches for different blocks and ran this stimuli on VCS and Whisper to ensure functionality
- Migrated Tenstorrent's enhancements of the vector, floating point, load/store units - and more - to Google's head commit
- Generated diagrams in React using Python and data from SQL to visualize the RTL interfaces; streamlining testbench generation

Tesla **January 2021 – August 2021**
Silicon Development Intern – Autopilot Hardware Group (Full-time) *Palo Alto, CA*

- Aided bringup of the newly-received compute die by delivering a temperature and voltage heartbeat so the hardware team wouldn't melt the silicon
- Provided SPI, temperature, and voltage drivers for third-party IPs, and a software solution for a bug in the SPI silicon
- Implemented, integrated and rigorously tested a software architecture designed to funnel temperature and voltage data over the buggy SPI
- Synced a python script to protobuf messages received from the hardware's UART, and automated sending requests to the hardware via .JSON files
- Cleared a SW bottleneck blocking coworkers from building QSPI images by creating a Python script with linux-like commands working around the LittleFS library
- Gained experience writing firmware running ARM CMSIS RTOS wrapper for FreeRTOS, including drivers, interrupt handlers

RESEARCH

Wireless Networking and Communications Group - UT Austin **August 2021 – Present**
Undergraduate Researcher *Austin, TX*

- Generated image datasets of Airsim drone simulations in Unreal Engine using C++ and Python; made to help train a
- Coordinated with another undergraduate student to deliver object-detection using MobilenetV2 on the Jetson Nano
- Provided a networking stack to detect latency for a center-less cloud of Jetson Nanos using ZeroMQ and Google Protobuf
- Worked closely with a PhD student under researching In/Out of Distribution inputs to neural networks

PROJECTS

Cycle-level CPU Simulation **August 2021 – Decenmber 2021**

- Generated image datasets of Airsim drone simulations in Unreal Engine using C++ and Python; made to help train a CV model
- Coordinated with another undergraduate student to deliver object-detection using MobilenetV2 on the Jetson Nano
- Provided a networking stack to detect latency for a center-less cloud of Jetson Nanos using ZeroMQ and Google Protobuf
- Worked closely with a PhD student under researching In/Out of Distribution inputs to neural networks

Cycle-level CPU Simulation **August 2021 – Decenmber 2021**

- Generated image datasets of Airsim drone simulations in Unreal Engine using C++ and Python; made to help train a CV model
- Coordinated with another undergraduate student to deliver object-detection using MobilenetV2 on the Jetson Nano
- Provided a networking stack to detect latency for a center-less cloud of Jetson Nanos using ZeroMQ and Google Protobuf

- Worked closely with a PhD student under researching In/Out of Distribution inputs to neural networks

SKILLS

- **Programming Languages:** System Verilog, Verilog, C, Python, C++, Perl, ARM Thumb/Risc-V Assembly
- **Libraries:** ARM CMSIS RTOS, FreeRTOS, Protobuf, Nanopb, Threading (Python), PySerial, PyFTDI, LittleFS (Python), PyTest
- **Computer Science:** Object Oriented Programming, Data Structures, Machine Learning, Operating Systems, Agile
- **Software Applications:** VCS, UVM, Whisper, Vivado, Git, Lauterbach, Linux, MATLAB, KiCad, Autodesk Eagle, Latex