Zixuan Kang (Harry)

333738, Georgia Tech Station | Atlanta, GA 30332 | +1 (470)398-7810 | zkang35@gatech.edu | Chinese Citizen

Objective

Computer Engineer with strong communication skills and experience in embedded system design and in electrical circuit design. Cooperative, and adaptive to different working environments and new challenges, especially in the topic of PCB design and firmware coding. Seeking for a full-time from Summer 2022, willing to relocate.

Education

Georgia Institute of Technology | Atlanta, GA *August 2018 – Present*
Bachelor of Science in Computer Engineering, GPA 3.94 Expected Graduation, May 2022
Minor in Computer Science: System Architecture

Skills

Programming: Python, C/C++, JavaScript, VHDL, System Verilog (UVM)
Platforms: Linux (Ubuntu, Debian), Windows (WSL)
Hardware: Arduino, ARM MBED microcontroller, TI C Series Microcontroller, FPGAs, oscilloscope, logic analyzer
Software: Altera Quartus II, EAGLE, QuestaSim, Visualizer, Verdi
Communication: Design proposals, technical reports, instruction manuals, presentations (large and small audiences)
Languages: French (conversational), English (fluent), Chinese (native)

Experience

ARM Ltd. | Chandler, AZ *May 2021 – August 2021*
Verification Engineer Intern

- Learned the concept and the general structure of Universal Verification Methodology (UVM).
- Experienced the complete of designing testing sequences => writing stimulus => writing checkers => improving coverage.
- Debugged nightly regressions errors and categorize them (whether they are RTL issues or Test Bench issues). Came up with patches to fix those bugs. Helped the team to reach an important milestone (Limited Access Control, LAC).

GT Solar Racing | Atlanta, GA *August 2019 – August 2021*
Auxiliary Subteam Lead
We design and build the control system of the cars, including head lights, blinkers, taillights, and regenerative brake. We are also responsible for displaying the information about the solar card to the driver.

- Reversed engineered a dongle terminator and designed a PCB board using Tiva™ that integrates the Rudman BUS ó UART ó CAN BUS critical path instead of using the dongle terminator and an Arduino as the CAN Transceiver. Reduced the connection number on the critical path from 4 to 2 to improve the stability and save the team $200 for dongle terminator replacement.
- Designed an embedded system consists of a pair of PCB board that controls the peripheral of the solar car and wrote the firmware code for Tiva™ C Series Microcontroller used on the board, featuring an acknowledgment system that enables several GPIO board on the car at the same time to improve the modularization of different parts of the solar car.

CS 3210 Operating System Design | Atlanta, GA *January 2021 – Present*
Teaching Assistant

- Helped students with problems during classes/projects regarding topics such as multi-threading, scheduling, synchronization, communication, and access control. Projects will cover design and implementation of several operating systems components.
- For the labs of the course, we mainly modified the virtual memory system (zero-lazy allocation and copy on write forking), threading (building a one-to-one user-thread library), scheduling (A system with FIFO and RR with preemption) and the booting process of xv6(dynamically allocate space for the kernel after the bootloader instead of static space allocation).

Relevant Coursework

Advanced Computer Architecture (Graduate Level Course): This course addresses advanced issues in the design and evaluation of high-performance computer architectures. Topics include RISC design techniques, pipelining, instruction-level parallelism, pipelining, superscalar systems, high-performance memory systems, and performance evaluation.

Processor Design: An intermediate-level project-based course focuses on the principles of processor design while implementing a fully functional (but simple) pipelined processor on a real Field-Programmable Gate Array (FPGA). We will implement the processor step-by-step on a real FPGA board using Verilog. We will study the principles of hardware design and synthesis as we implement the processor. We will also examine and benchmark our design using a few applications. To be able to benchmark the design, we will learn how the hardware and software interface and interact through the Instruction Set Architecture (ISA), the machine language, and the assembler.