Songyue (Landon) Wang

No. 19 (A), Yuquan Rd
Beijing 100049, China

+86-186-1837-6601
wangsongyue18@mails.ucas.ac.cn

EDUCATION

University of Chinese Academy of Sciences (UCAS) – Beijing, China

Bachelor of Computer Science 2018 – 2022

- **GPA**: 3.85/4.00
- **Research Interests**: Computer Architecture and Operating System
- **Main Courses**

  Data Structure, Assembly Programming, Algorithm, Artificial Intelligence, Compile Principle, Operating System, Computer Organization and Design, Advanced Computer Architecture, Computer Network, Parallel Computing.

RESEARCH EXPERIENCE

Cooperative Operation of the Fleet in the Environment of the Internet of Vehicles

Research Leader, Department of Computer Sciences, UCAS 09/2019 – 08/2020

- Constructed our experimental car robot fleet based on STM32 embedded board and Raspberry Pi.
- Realized object detection based on OpenCV and built hardware electronic control driver over STM32.
- Investigated research results of the existing papers and evaluate its effect on our fleet.

SERVE: An Agile Hardware Design and Evaluate Platform Based on CI/CD and Cloud FPGAs

Researcher, State Key Laboratory of Computer Architecture (SKLCA) 07/2020 – 09/2021

Institute of Computing Technology, Chinese Academy of Sciences (ICT, CAS)

- SERVE provided fully online flows of hardware RTL design and evaluation: stimulation, synthesis, bitstream generation and evaluation on cloud FPGAs.
- Used OpenStack and Kubernetes to build a container platform. Built cluster monitoring facilities and carried out experiments to explore the optimal number of VMs and FPGAs.
- Designed and implemented online Integrated Logic Analyzer (ILA) debugging solution for remote users. Users can login to GUI containers connected to cloud FPGAs on Kubernetes cluster.
- SERVE had provided services to more than 1000 users among universities in China, also supported hardware curriculums of UCAS, several competitions and projects of SKLCA, ICT. I also submitted a China patent.

RISC-V Async Memory Access Unit (AMU) with Message Interface Prototype System

Researcher, SKLCA, ICT, CAS 07/2021 – 06/2022

- AMU supports asynchronous memory access requests from the CPU micro-architecture for future remote memory pooling scenario, which has high bandwidth and high memory access delay. AMU also breaks the limitation of ROB and MSHR on memory access parallelism.
- In pipeline and MMU, it supports 3 instructions for req sent and get finish sequence id. In L2 cache, a scratchpad memory (SPM) is divided in the unit of cache way, which is used to store the data sent and replied by asynchronous access and req and resp queue metadata.
- I used Chisel HDL to design the hardware prototype system based on the open-source RISC-V processor NutShell (RV64IMAC, like RocketChip), designed block design and deployed it on FPGA.

ByteFPGA: FPGA Development Middle-Platform for AI Computing Acceleration

Intern Researcher, Data System Group, AI Laboratory, ByteDance Inc 11/2021 – 02/2022

- It abstracts the underlying FPGA interfaces and resources (DDR, network), and provides services based on different types of FPGA boards through a unified interface to upper applications (gzip, video encode, etc.).
- Using TCL and Bash scripts, combined with automation tools, I designed general simulation, FPGA deployment and other flows for different EDA tools (vivado, quartus, VCS, etc.).

Software Defined Rack (SDR) for Resource Disaggregation in Cloud Computing

Intern Researcher, Network Research Group, Microsoft Research Asia (MSRA) 02/2021 – 06/2022

- The expectation of SDR is to use PCIe switches to build a PCIe network, so that VMs can quickly add or delete connected resources (NVMe, GPU) and SDR should ensure the disaster recovery mechanism of the network.

- I investigated similar products (Dolphin, GIGAIO, etc.) and studied their technical papers to enhance our goals.
- Designed the framework and graph of SDR, and now I am designing the API between RackOS (SDR's management OS) and PCIe switches based on serial port protocol.

Selected Projects

7–Stage Sequential Single-Issued RISC-V CPU (RV64IMA)

- It can boot Linux 4.18.0 (with BusyBox) and xv6-riscv on FPGA, and score for running Coremark is 357@250MHz.
- Memory access path: 4-way L1 I-cache and D-cache (write-through), and MMU supported Sv39 standard.
- Complete CSR and CLINT design, supports handling all interrupts and exceptions within the specification.
- We also build a framework to compare the processor registers with a simulator at every cycle when simulating.
- I contributed to system booting, design of memory access unit, pipeline framework and atomic instructions.

7–Stage Sequential Dual-Issued MIPS CPU (MIPS R1) 3rd Prize of NSCSCC' 21 (mentioned below)

- Supports 82 instructions of MIPS R1, integrates I-Cache, D-Cache and TLB we designed, and it accesses memory through the AXI4 interface. Now it has entered the **tape-out** process at SMIC@110nm sponsored by OSCPU plan.
- Carried out back-end evaluation and timing optimization, it can boot an embedded OS RT-Thread on FPGA.
- A few modern processor technologies are implemented, such as branch predictor and issue/commit queues.

A Unix-like OS based on Dual Core Rocket Chip (RV64GC) Evaluation: 100/100

- OS run on a programmed FPGA (Xilinx PYNQ) with dual core Rocket Chip's bitstream, file system on SD-card, and a NIC driver was integrated in OS. Processes can send/receive TCP/IP packets through NIC on FPGA board.
- Isolation mechanism: kernel runs in S-Mode, user processes run in M-mode.
- OS can schedule processes on dual core, sync-primitives and communication also support on dual core.
- Shell support rich functions (syscalls) such as load ELF files from file system to memory and execute it.

Compiler from a C-like Language to RISC-V ASM (RV64GC) Evaluation: 98/100

- It can handle a series of data structures and grammars such as arrays, loops, recursion etc. with ANTLR tool. The process includes grammar analysis, grammar tree construction, target language generation, etc.

Teaching Experience

Experiment on Computer Organization and Design (COD) 2021 Spring

Teaching Assistant, Department of Computer Science, UCAS 03/2021 – 07/2021

- Contributed to curriculum project design, maintained and improved the FPGA cloud platform (SERVE) used in course, lectured to students and graded the students' RTL design. This class taught totaled 100+ undergraduates.

Practice on Operating System (OS) 2021 Fall

Teaching Assistant, Department of Computer Science, UCAS 09/2021 – 12/2021

- Designed course projects, organized and participated in course Q&A sessions and wrote instruction manual to help students' debugging. Class taught totaled 100+ undergraduates major in CS.

Honors & Awards

- **Outstanding Student**, UCAS (2019, 2020 and 2021)
- **2nd Academic Scholarship**, UCAS (2019, 2020 and 2021)
- **National College Student Scholarship**, UCAS (2021)
- **3rd Award** of 5th National Student Computer System Capability Challenge 2021 (NSCSCC 2021): CPU Design Track (Loongson Cup), Ministry of Education of China (Co-Organizer: Xilinx Inc and Loongson Co., Ltd)

Relevant Skills

- **Language** English (fluent, TOEFL ITP Plus: 547), Chinese Mandarin (native)
- **Software Skills** Programming: C, C++, Python, ASM (x86, RISC-V), tcl and Linux bash
  Performance evaluation tools; Machine learning (PyTorch); Git and DevOps tools.
- **Hardware Skills** RTL Design (Verilog, Chisel); FPGA (Xilinx Vivado, verilator)
- **Knowledge** Basic cloud computing technology and experience (virtualization, container, etc.)
  Familiar with CPU architecture, memory hierarchy, transmission bus, etc.
  Familiar with hardware and software co-design: OS, hardware driver, etc.