Xi-Zhu Wang | ASIC / Digital IC Design & ML Accelerator Hardware

z790305000@gmail.com • github.com/xizhuwang
Taipei/Taoyuan, Taiwan | Open to relocation

Summary

M.S. EE @ NTUST. Hands-on **RTL design** + **assertion-based verification** + **synthesis/P&R** and FPGA/SoC prototyping; research on **HBM-PIM** for **multi-user LLM inference scheduling**.

Target: **ASIC/Digital IC** / **SoC/Accelerator** roles.

NTUST

M.S., Electrical Engineering *2024 – 2026 (Exp.)*

B.S., Electrical Engineering (Double Major: Finance) *2020 – 2024*

Skills

RTL/verification: Verilog; VCS, Verdi; ModelSim; SV properties/assertions; Jasper (formal); regression scripting (Bash, Python)

PD/Signoff: Synopsys DC/ICC2; Cadence Innovus; Calibre DRC/LVS; timing/power reports; P&R flow integration

SW/FPGA: C/C++, Python, Linux, Git; Vivado, Vitis HLS/Vitis-AI (ZCU102/Pynq-Z2/MPS3); OpenROAD

AI HW: CNN accelerator; LLM inference: scheduling/quantization; bandwidth/latency modeling

Experience & Projects

NTUST **Taipei**

HBM-PIM Multi-User LLM Scheduling (ACM SAC '26) *2024 – Present*

- Paper: *Adaptive Multi-User Scheduling for Large Language Model Inference on HBM-Based Processing in-Memory Accelerators.* ACM SAC '26. doi:10.1145/3748522.3779794
- Designed adaptive scheduling policies under bandwidth/memory-pressure constraints; analyzed mixed-precision serving trade-offs (latency/energy/accuracy).
- Produced implementation-oriented evaluation artifacts (workload abstraction, model parameters, reproducible analysis scripts).

NTUST **Taipei**

Teaching Assistant – Computer Organization *Apr 2025 – Jul 2025*

- Supported labs on CPU datapath/control and RTL debugging; guided students using **ModelSim** for simulation and waveform-based verification.
- Assisted OpenROAD-based digital implementation exercises (flow setup, timing/power report interpretation); built Python/Shell autograding scripts.

Timing Correlator and 16-QAM MIMO Detector (TSMC 16nm ADFP) *Sep 2026 – Dec 2026*

- Implemented K-Best based detector in Verilog; ran synthesis and P&R in TSMC 16nm ADFP.
- Post-synthesis area: **17655.4** um^2 (Design Compiler).

RISC CPU + MNIST CNN Accelerator (TSMC 16nm ADFP) *Apr 2025 – Jun 2025*

- Integrated CNN accelerator with 32-bit RISC CPU for end-to-end MNIST inference (89% accuracy).
- Completed **post-layout** implementation in TSMC 16nm ADFP; achieved ∼100 MHz; estimated **2.46 mW** dynamic power.
- Built lightweight SV verification: signal checks + **property assertions** for key control/handshake behaviors; automated runs via scripts.
- Post-layout area: **17655.4** um^2 (Innovus).

Standard-Cell Placement Legalization (C++, Open-Source) *Jul 2024 – Dec 2024*

- Implemented legalization algorithm to minimize displacement under placement constraints; improved avg displacement by **18%**.
- Evaluation: ∼**50k cells**, runtime ∼**10 min**; engineering focus on stable data structures and constraint handling.
- Repo: github.com/xizhuwang/Standard-Cell-Placement-Legalization

NTUST **Taipei**

Research Assistant / EDA Server Admin *Jun 2024 – Present*

- Managed shared EDA infrastructure and tape-out related flows across multiple nodes; improved license/job workflow, cutting prep time by **20%**.
- Supported students/researchers on tool usage, regression stability, and reproducible build environments.

TSRI Training

- Design Compiler (Logic Synthesis, ADFP–TSMC 16nm); IC Compiler II (Cell-based P&R & Verification); Catapult HLS (AI HW Architecture); SystemVerilog; Jasper Formal Property Verification; Emerging NVM Array (Design/Analysis/Testing)

Links

GitHub: github.com/xizhuwang

SoC Course: socv-1132 (course-based; contributed integration/scripts/lab solutions)