上海科技大学
ShanghaiTech University

Teaching Computer Architecture and AI Accelerator Design through the RISC-V Ecosystem

Speaker: Siting Liu
School of Information Science and Technology (SIST)
ShanghaiTech University

立志成才报国裕民

Outline

- Introduction
  - Background & challenges
- Motivation
  - Why RISC-V for teaching?
- Course material & the use of RISC-V
  - Computer architecture
  - AI accelerator design
- Takeaways

Background

上海科技大学
ShanghaiTech University

- Overview: IC education at ShanghaiTech

Devices	Circuits	Systems
Physics of Semiconductor; Intro. to Nanoelectronics; Optoelectronic Devices; Micro/Nano Processing Technology; Microelectronic Devices; etc.	Digital/Analog/RF Integrated Circuits; Digital VLSI Design Flow; Optoelectronic Devices; Micro/Nano Processing Technology; etc.	VLSI Design Automation; FPGA-based Hardware System Design; Chip Testing: Fundamentals and Applications; Computer Aided Verification; Computer Architecture; AI Computing Systems;

Challenges

- Traditional courses rely on proprietary ISAs (e.g., x86/ARM), which fundamentally restricts architectural exploration and hands-on design experiences.
- It limits the chance that students learn from practice.
- **RISC-V is open, simple yet elegant.**

Teaching Process & Course Content

- Knowledge transfer (lecture) → Learning from practise (labs & projects)
  - Computer architecture I and its project
    - Basic understanding of how a computer works;
    - Memory hierarchy & memory management;
    - Optimizations through parallelism;
  - AI accelerator design with RISC-V extension (backward design)
    - ISA extension and hardware implementation;
    - Advanced hardware/software technologies for optimization;
    - Memory hierarchy considerations;
    - Basic understanding on domain specific architecture (DSA);

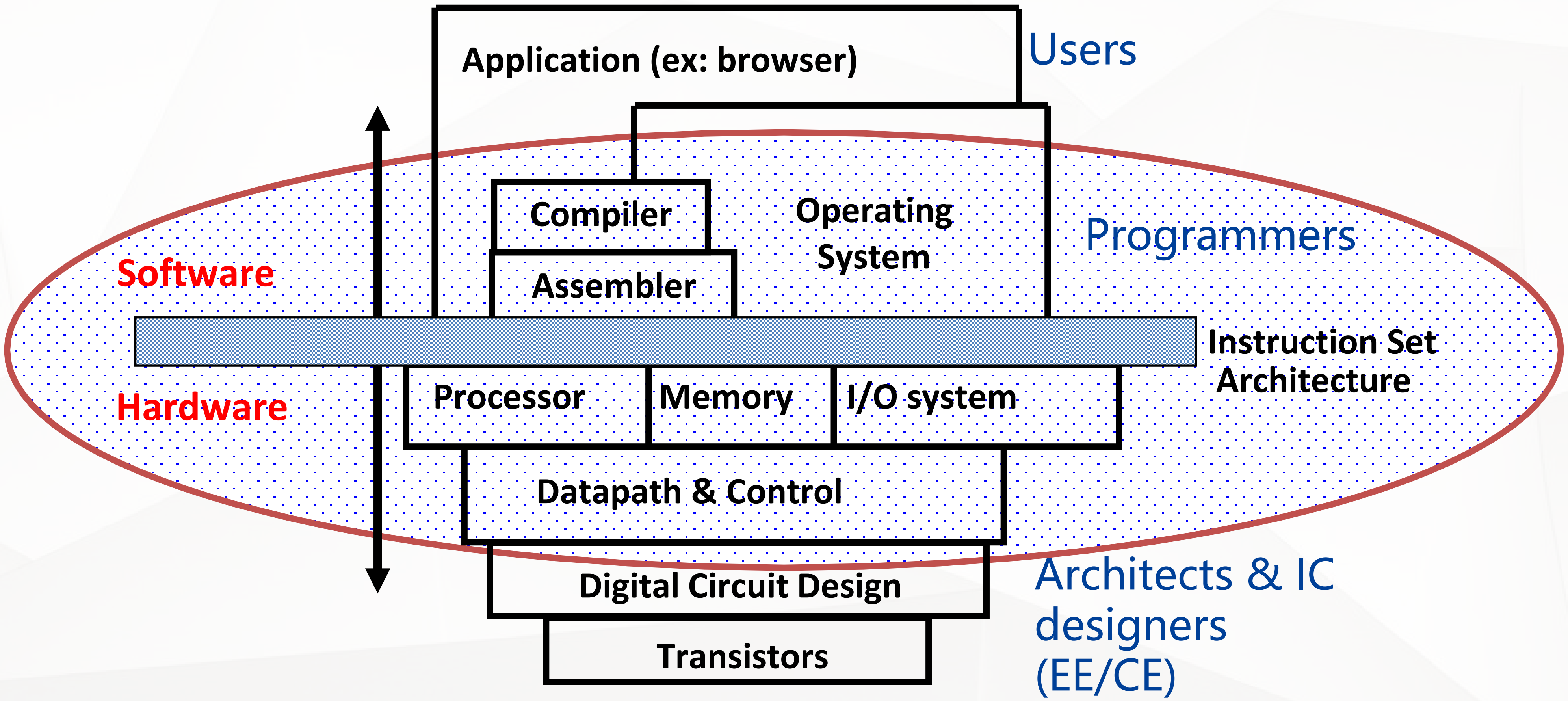
Basic Course Info.

上海科技大学 ShanghaiTech University

- **Computer architecture I and its project**
  - One of the most important required course of an undergraduate SISTors;
  - Consists of the theoretical part (4 credits) and the hands-on part (2 credits);
  - Involves 200~ students each year;
  - Developed based on UC Berkeley’s CS61c;
- **AI computing systems**
  - Optional specialized course for both graduate and undergraduate SISTors;
  - Consists of theoretical part (3 credits) and hands-on part (1 credit);
  - Involves 40~ students each year.

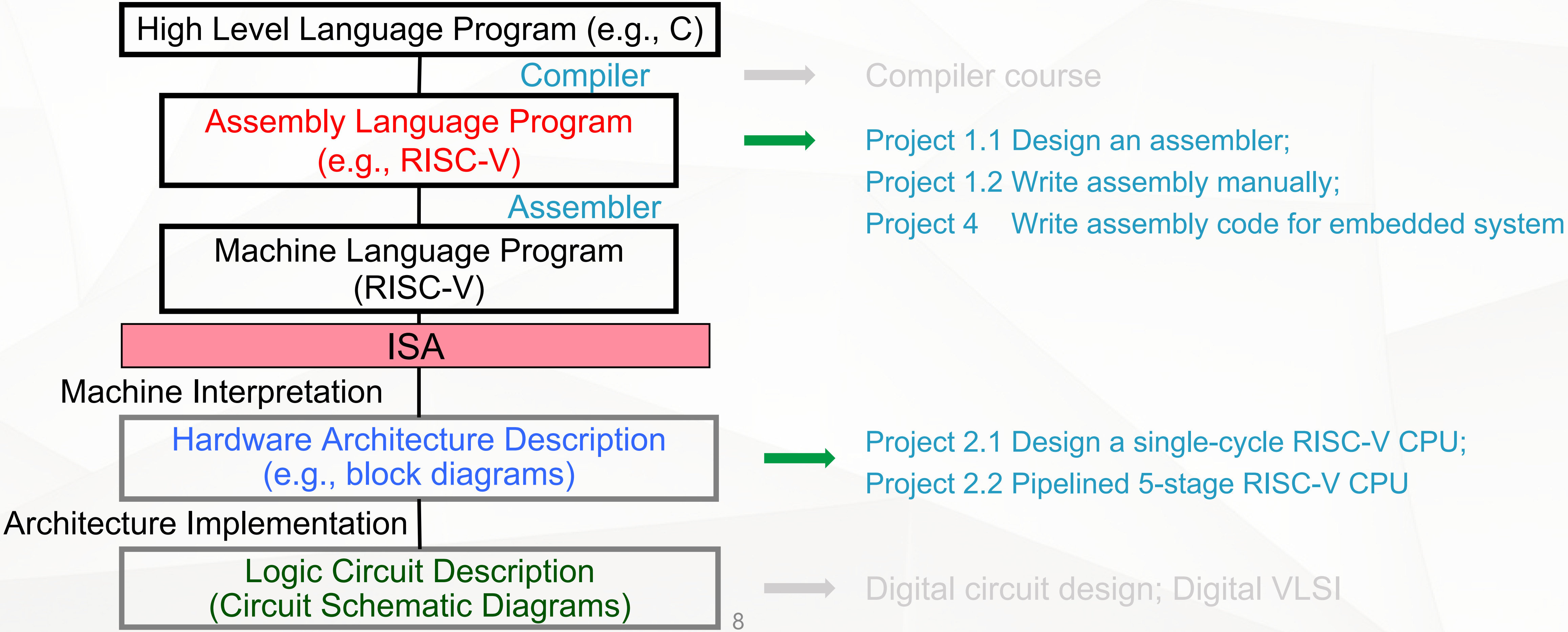
Computer architecture I

- Computer architecture I and its project



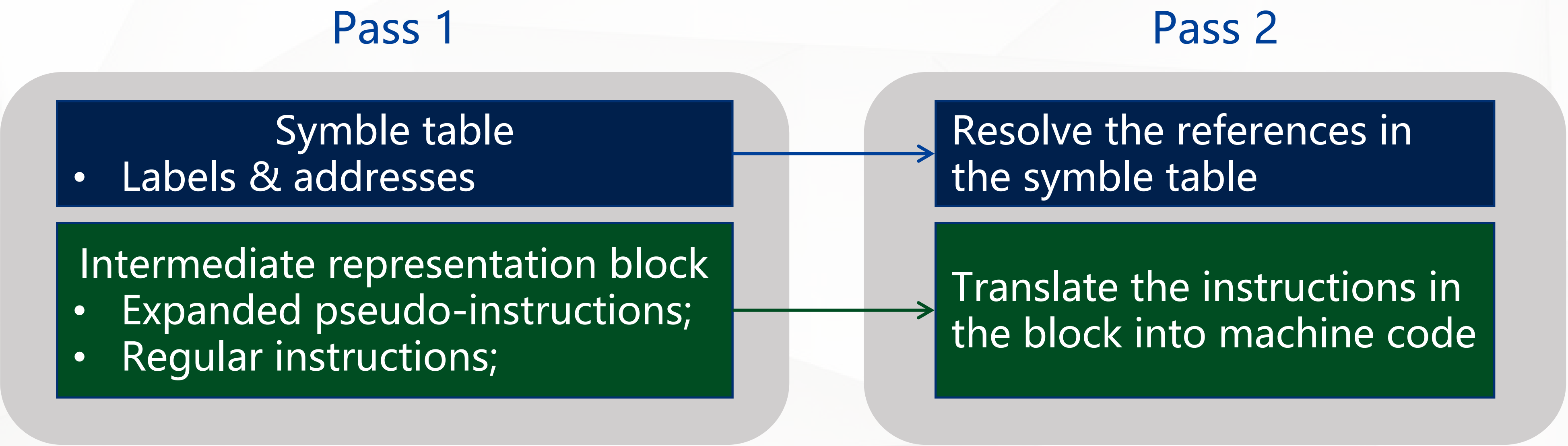
Course Materials

- Computer architecture I and its project
  - Basic understanding of how a computer works;



Project 1.1 A RISC-V Assembler

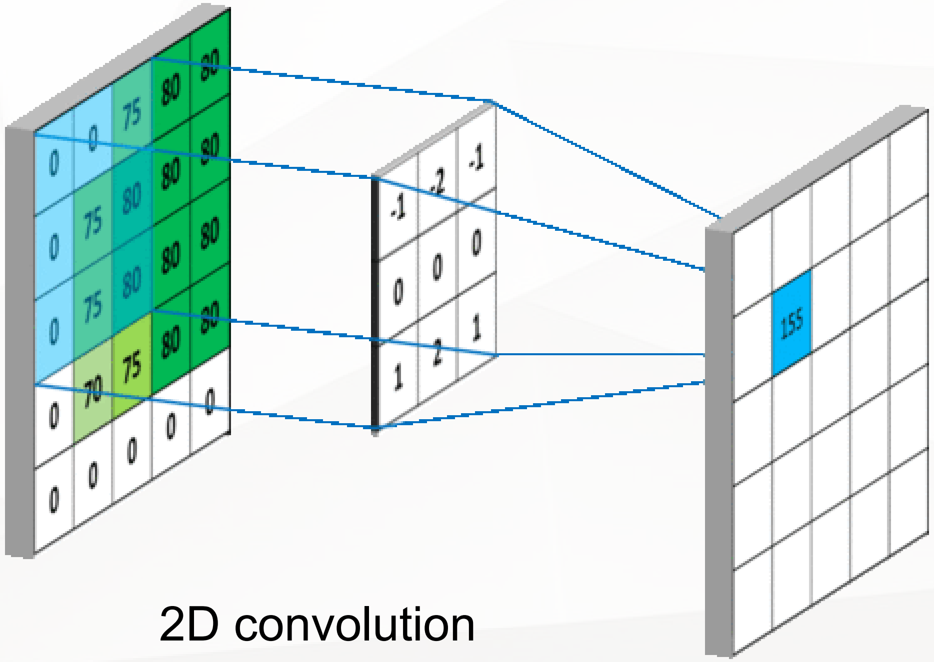
- Fortify the understanding of RV32I assembly and its encoding;
- Help students understand what an assembler does;
- Hands-on experience on building an assembler.



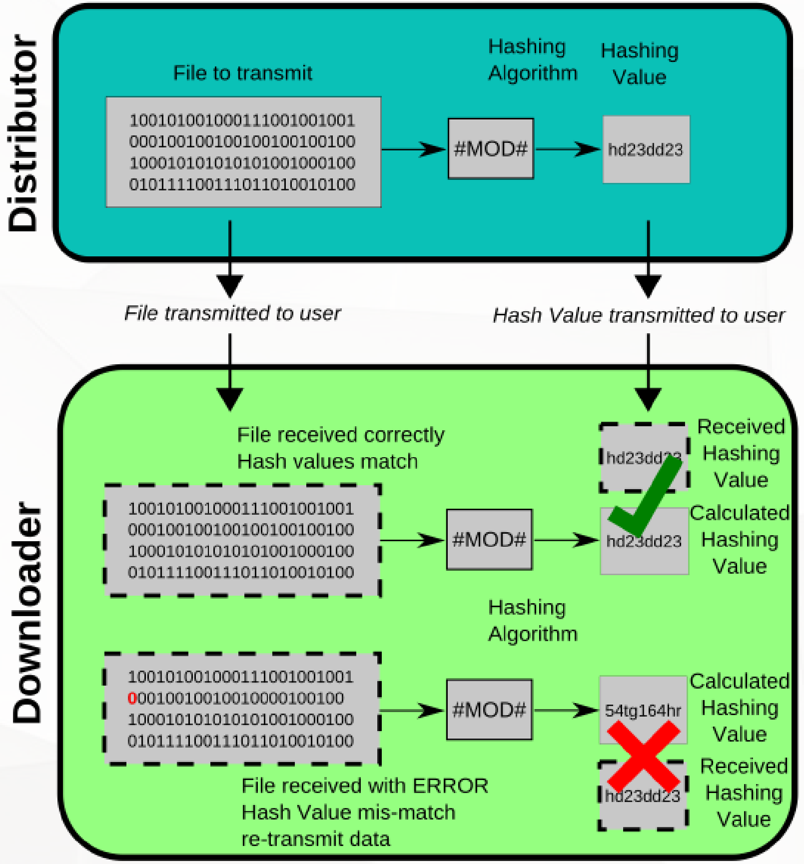
Project 1.2 RISC-V Assembly Practise

上海科技大学 ShanghaiTech University

- Use real applications to familiarize the students with
  - RISC-V assembly
  - Calling conventions



2D convolution

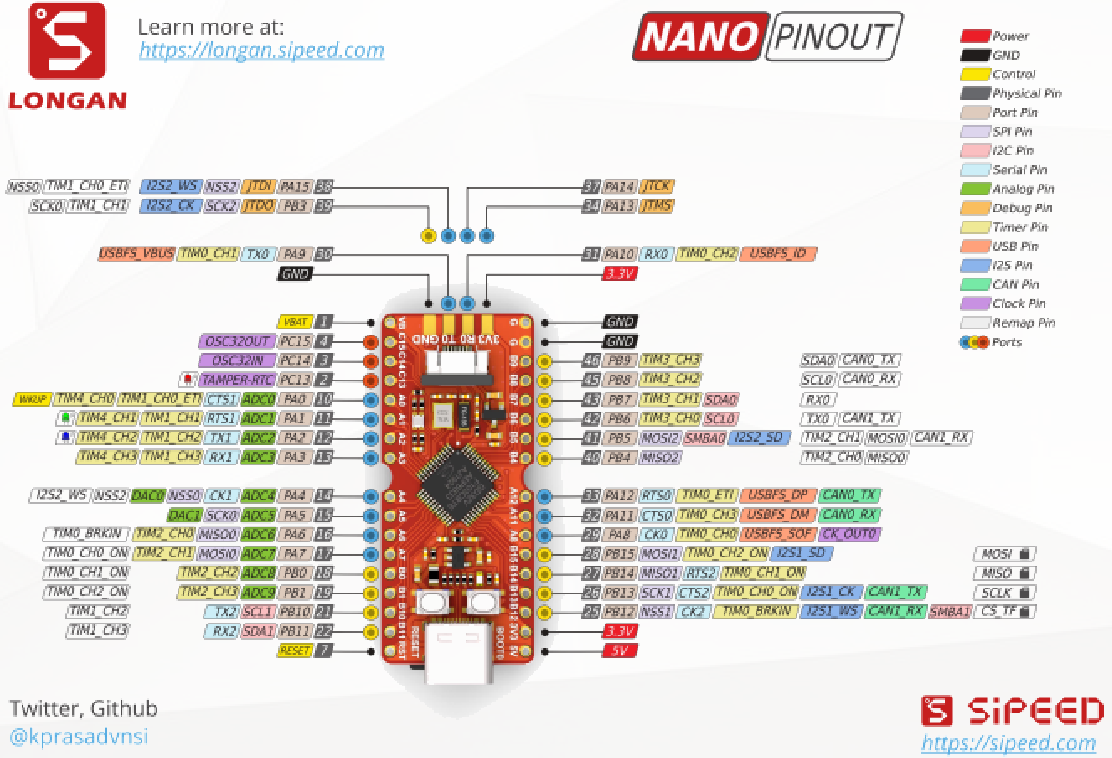


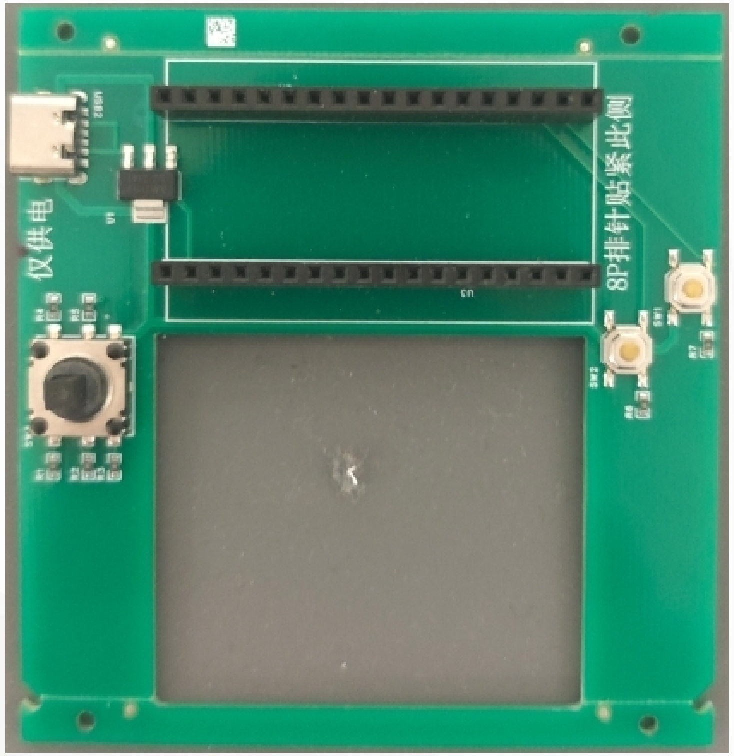
MD5 algorithm

Project 4 RISC-V Programming on Real Hardware

上海科技大学 ShanghaiTech University

- A low-cost Longan Nano development board with customized base board;
- Equipped with GigaDevice GD32VF103CBT6 MCU supporting RV32IMAC instructions;
- Carrier board designed by the teaching assistants

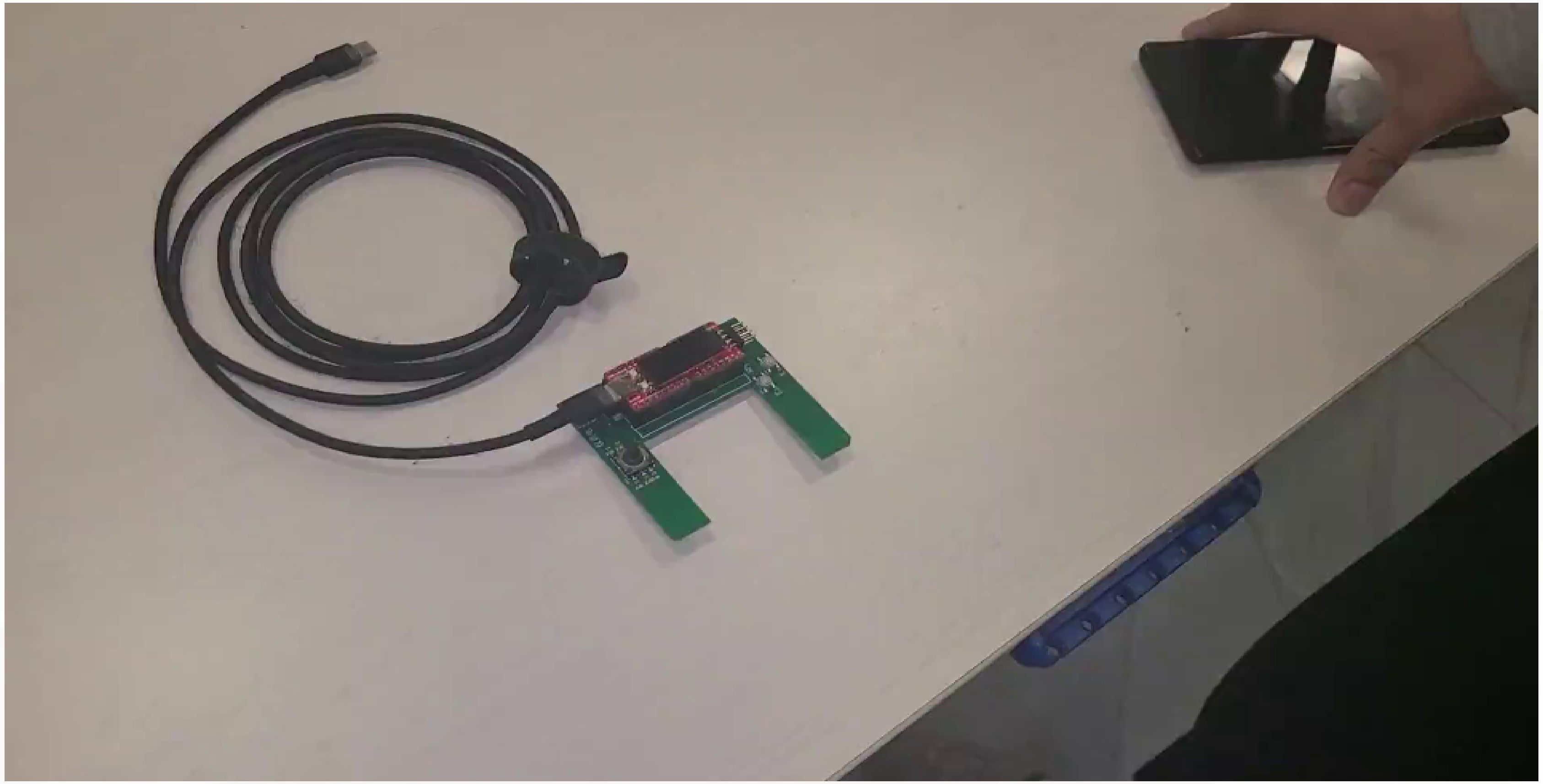




Project 4 RISC-V Programming on Real Hardware

上海科技大学 ShanghaiTech University

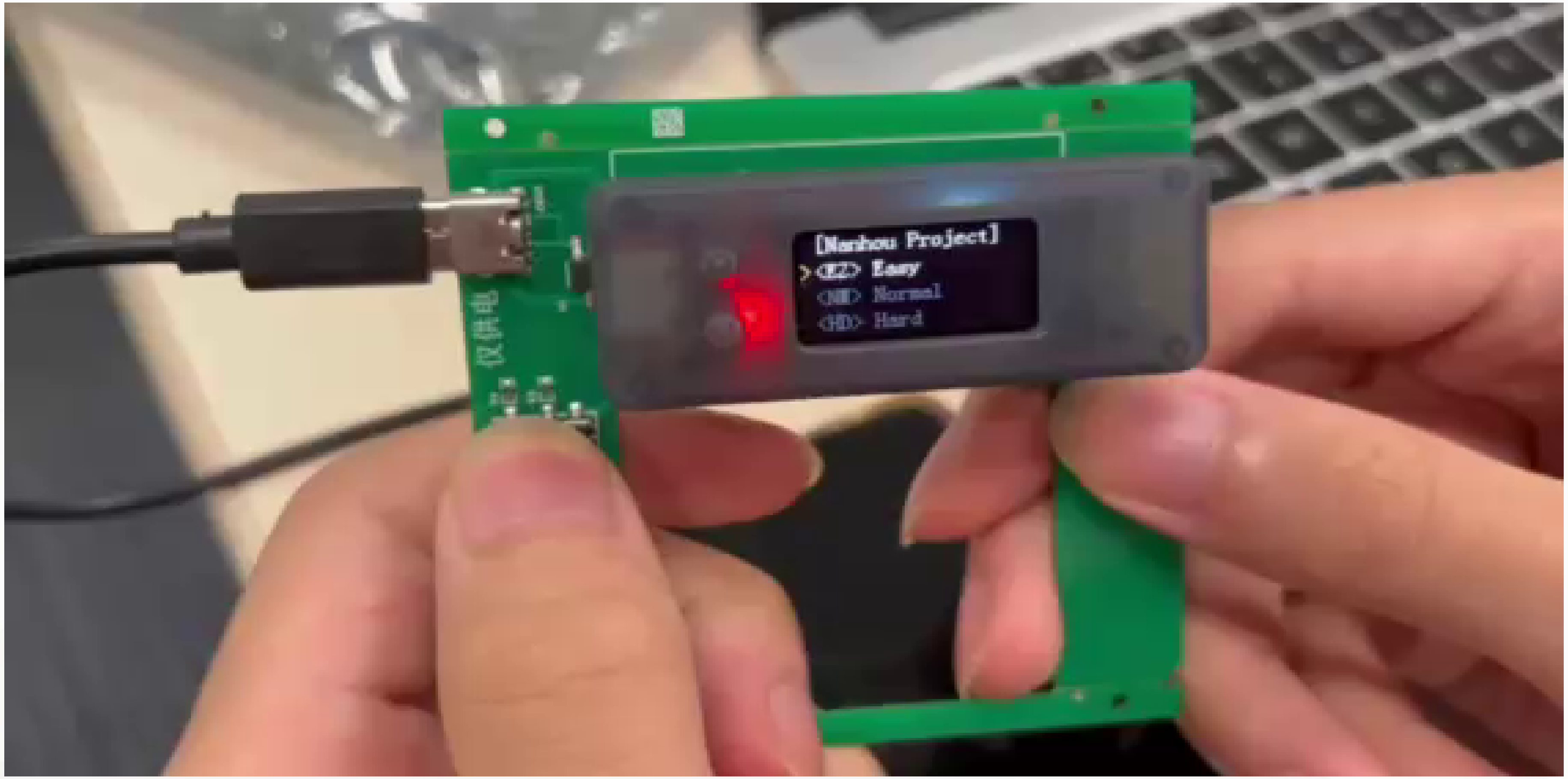
- Reference design provided by the teaching assistant



Project 4 RISC-V Programming on Real Hardware

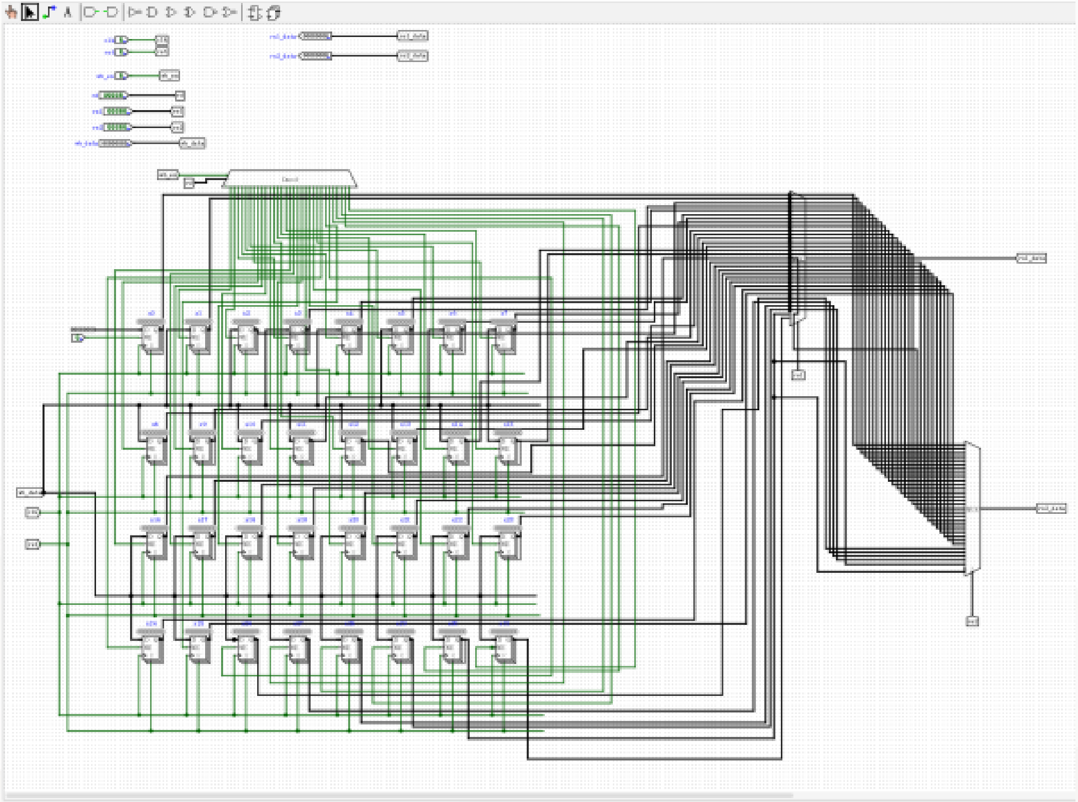
上海科技大学
ShanghaiTech University

- Students’ work: shooting game

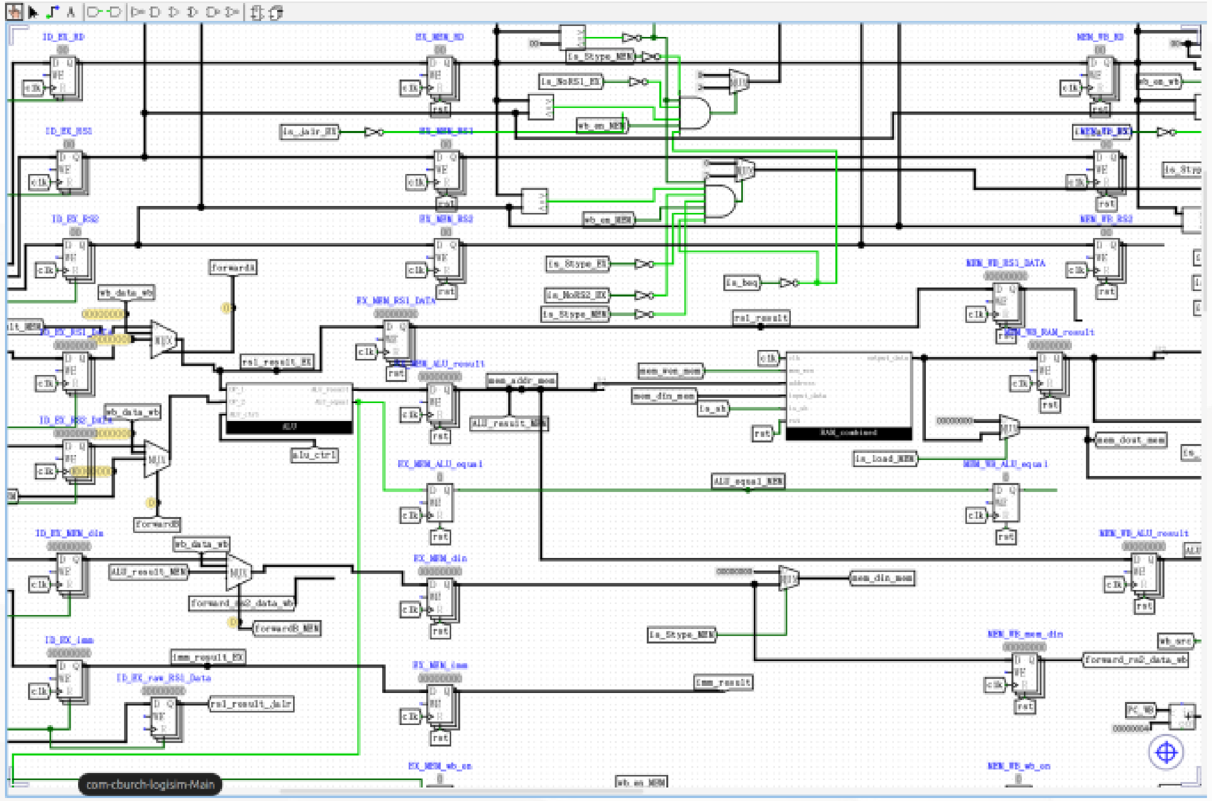


Project 2 Simple RISC-V CPU Design

- Employ the open source software Logisim to implement a CPU design;
- Pipeline the CPU to understand various hazards and how to solve them.



A register file implementation from the students' submissions (project 2.1)



Part of the pipelined CPU implementation with pipieline registers and forwarding mechanism from the students' submissions (project 2.2)

AI Computing Systems

上海科技大学 ShanghaiTech University

Software | Hardware

Neural network (NN) models

Hardware-aware NN compression
Quantization and pruning

Special circuitry for neural network:
- Systolic array;
- Special function unit;

Computer architecture review &
parallelism in NN accelerators

Machine learning compiler

Memory hierarchy considerations

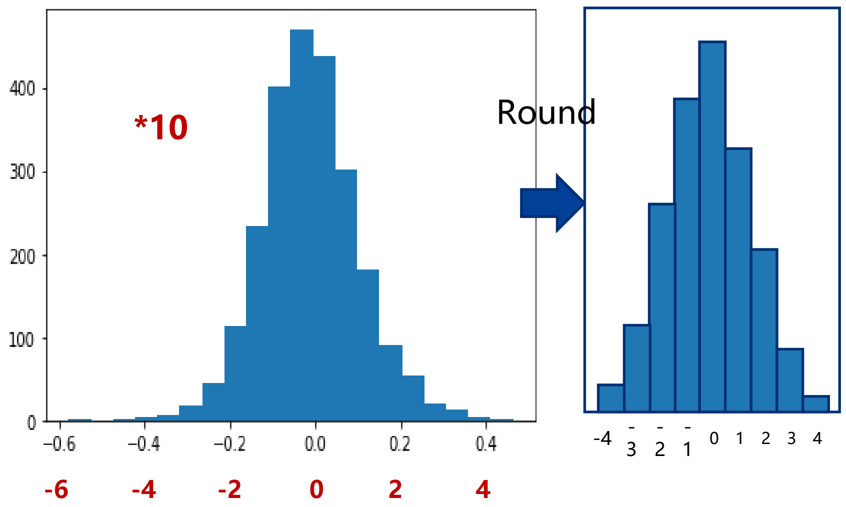
Sparsity

Advanced technologies
- Neuromorphic computing;
- Computing-in-memory; etc.

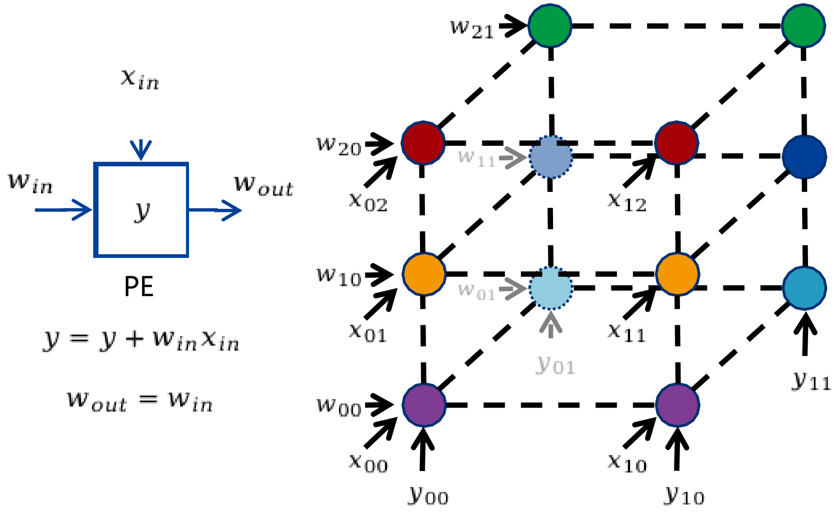
Lab Course Materials

上海科技大学 ShanghaiTech University

Lab 1 Quantize a neural network



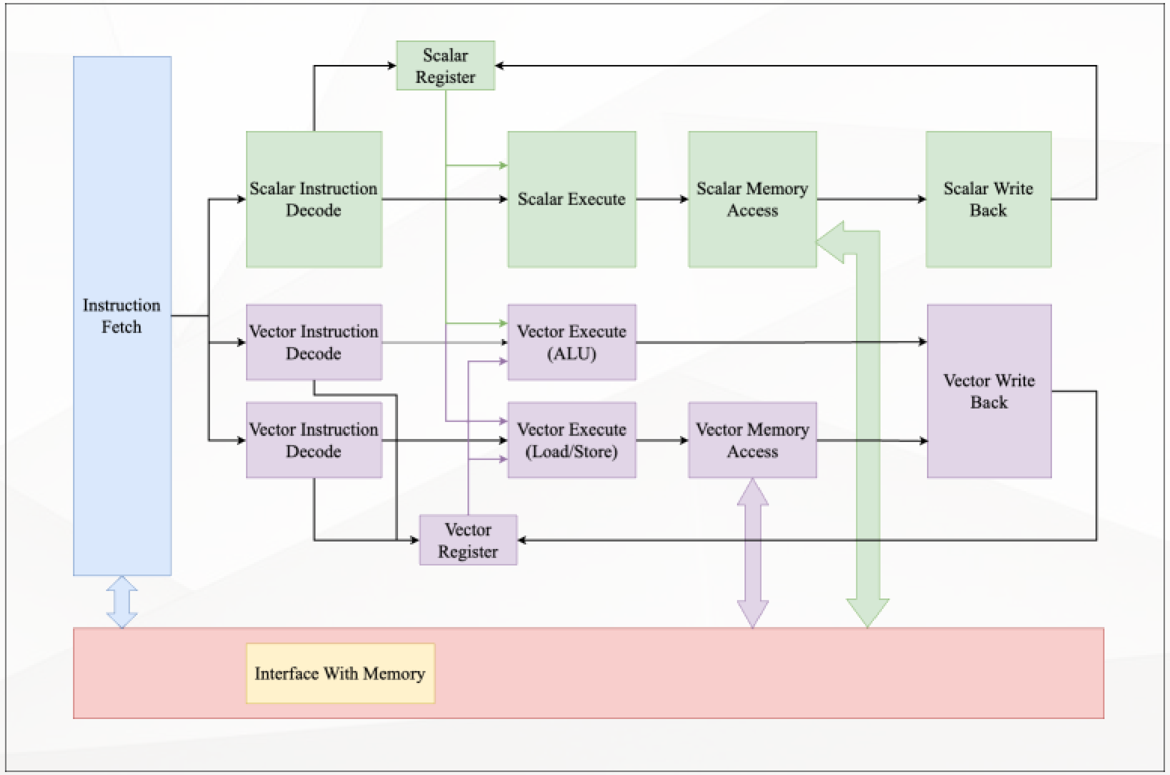
Lab 2 Hardware acceleration

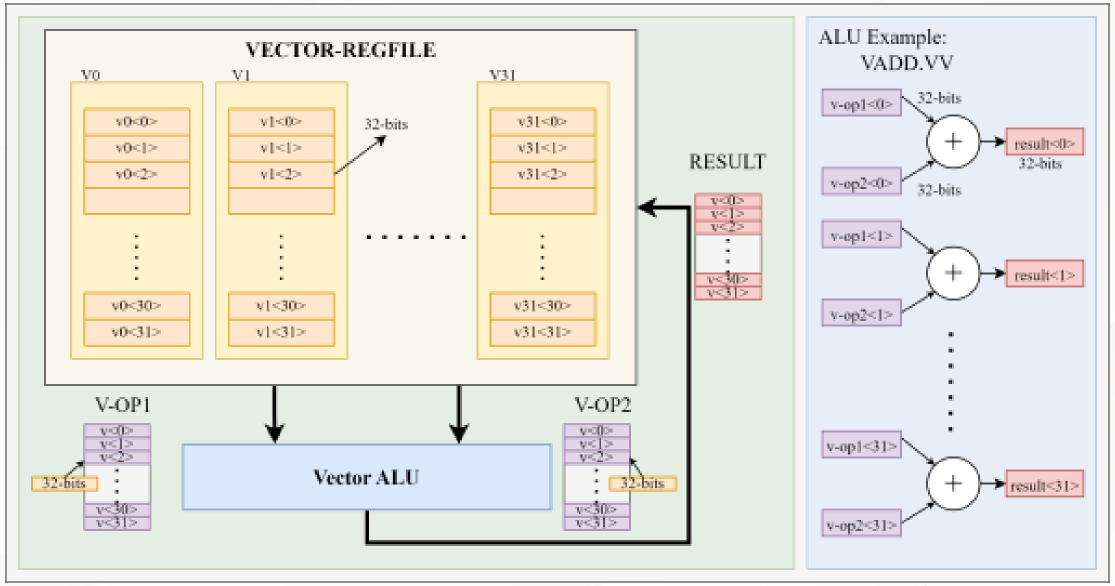


Lab 3: Vector Extension

上海科技大学 ShanghaiTech University

- Toy RISC-V processor with multi-issue, vector extension or custom multiply-accumulate (MAC) instruction;
- Learn the concept of vector processor through the implementation of vector extension;





Detailed datapath consisting of the vector register file and ALU

Custom Instructions for Neural Networks

上海科技大学 ShanghaiTech University

- **Project: design a neural network accelerator**
- Customize RISC-V instructions for neural network computations;
- Improve and evaluate the performance of a neural network accelerator.

Inst	Format	Implementation
mac	mac rd, rs1, rs2	x[rd] += x[rs1] * x[rs2]

shift	shift rd, rs1, rs2	for i != VLMAX -1:
		    x[rd][i] = x[rs1][i+1]
		for i = VLMAX:
		    x[rd][i] = x[rs2][0]

max	max rd, rs1, rs2	if x[rs1][i]>x[rs2][i]:
		    x[rd][i] = x[rs1][i]
		else:
		    x[rd][i] = x[rs2][i]

Rshift	Rshift rd, rs1, rs2	x[rd] = torch.round(x[rs2])

Some of the custom instructions from the students' project reports

```
Max-Run-Cycles = 500000000
Initialing RAM ...
Using simulated 64MB RAM
The image is /home/ubuntu/projects/project/projects/project/sw/build/cal-riscv64-mycpu.bin
Initial RAM done !!!
Initialing Data ...
The image is /home/ubuntu/projects/project/projects/project/data/bin/data.bin
Load Data done !!!
The program is running now......
-----------------------------------------------------------
Forward pass complete.
29 -4 17 -6 -2 -16 3 -38 18 -6

HALT-0
- /home/ubuntu/projects/project/projects/project/hw/vsrc/RV64I/rvcpu.v:501: Verilog $finish

-----------------------------------------------------------
The program finished after 528048 cycles.
Save the data into file /home/ubuntu/projects/project/projects/project/data/bin/save.bin
Done
```

Simulation results show that the neural network model is computed in 528,048 cycles, **541x** faster than the baseline scalar implementation (285,585,074 cycles)

Takeaways

- RISC-V is simple, open and modular
  - It enables better engagement of the students in the courses that have been impossible for years;
  - The students gain not only insterest but also hands-on experience on building computer hardware and lower-level software;

Thanks for your attention

Q & A