# CPU Project Handover

Your Name

2025-08-25

# Overview

- 32-bit RISC-V CPU core
- Quick orientation for new maintainers
- Demonstrate simulation flow & architecture

# Agenda

1. Context
2. Repository Tour
3. Vivado Simulation
4. Controller & Datapath
5. Results & Handover Tips
6. Recording Workflow
7. Wrap-Up

# Context

- Goal: simple RV32I CPU, single-cycle control
- Key components: controller FSM + datapath blocks
- Toolchain: Icarus, GTKWave, optional Vivado XSIM

# Repository Tour (VS Code)

- `README.md` : build & test commands
- `src/` : RTL modules (`leaf`, `glue`, `top`)
- `tb/` : testbenches (`<name>_tb.v`)
- `docs/` : module docs & schematics (`images/`)
- `Makefile` : automation for lint, build, run, schem

# Vivado Simulation – Top CPU

- Compile: `make vivado comp=cpu`
- Run **xsim** and open waveforms
- Show:
  - Program counter
  - Register file writes
  - Memory interface signals
- Test program: ADD, SUB, LOAD, STORE, BRANCH

# Controller & Datapath

- **Controller FSM**
  - Decodes instruction, asserts control signals
  - States: fetch $\rightarrow$ decode $\rightarrow$ execute $\rightarrow$ memory $\rightarrow$ writeback
- **Datapath**
  - ALU, register file, memory, muxes
  - Control signals drive multiplexers & next PC
- Use schematic: `images/cpu.svg`

# Results & Handover Tips

- Run all tests: `make` (Icarus) or `make vivado-all`
- Generate schematics: `make schem` or `make schem-top comp=cpu`
- Known limits: single-cycle, no hazard handling
- Note TODOs & next steps in `docs/handover.md`

# Recording Workflow

1. **Prepare**
   - Pre-run tests, arrange waveforms
   - Open slides & code in advance
2. **Record**
   - Use OBS or similar
   - Start with slides $\rightarrow$ VS Code $\rightarrow$ Vivado
3. **Narrate**
   - Explain commands, waveforms, schematic
4. **Keep** under 10 minutes

# Wrap-Up

- Reiterate project goals & status
- Point to repo and docs for follow-up
- Thank the viewer