

RISC-V ARCHITECTURE TRAINING

Schedule

Jim Wang (<http://phdbreak99.github.io>)

Dec. 2019

Schedule

2-day x 8-hour

Step-by-step

Lecture + demo + lab

Schedule / Day 1 morning

- **Schedule and self-introduction**
- **[Lecture] Introduction of RISC-V open ISA**
 - History and current status of ecosystem
- **[Lecture] RISC-V ISA: unprivileged spec**
 - Basic RISC-V ISA: key concepts
- **[Lab] Setup lab env**
 - Quick introduction of lab env used in this course
- **==== *Lunch break* ====**

Schedule / Day 1 afternnnon

- **[Demo] Compile assembly code and simulate on SPIKE**
 - GNU toolchain, assembly code example, compile & link, SPIKE simulator
- **[Lab] compile assembly code and simulate on SPIKE**
 - Code your own assembly code, compile & link, run simulation on SPIKE
- **==== Tea break ====**
- **[Lecture] RISC-V ISA: privileged spec**
 - Privilege modes, exception & interrupt, physical memory & virtual memory
- **[Demo] C code with proxy kernel and front-end server**
 - RISC-V sim env components, system call workflow, use PK to run "hello world" in SPIKE
- **[Lab] create new system call (optional)**
 - Change FESVR and PK to add a new system call, write C code to call this system call

Schedule / Day 2 morning

- **[Lecture] CPU architecture with RISC-V examples**
 - Basic concepts of CPU architecture, and some real RISC-V CPU examples
- **[Demo] verification suite**
 - ISA tests and built-in-self-check, Torture: random code generator
- **[Demo] RocketChip generator**
 - CPU architecture exploration with RocketChip generator
- **[Lab] generate new CPU with RocketChip**
 - Configure RocketChip in Chisel to generator new CPU, and run benchmark
- **==== *Lunch break* ====**

Schedule / Day 2 Afternoon

- **[Demo] Freedom IDE & HiFive1 board**
 - Quick example of SiFive's Freedom IDE, and debugging HiFive dev board
- **[Lecture] Uncore components**
 - TileLink, PLIC, CLINT, CLIC, debug, trace
- **[Demo] create custom instruction**
 - Select instruction encoding, change GCC to support new instruction
- **[Lab] create custom instruction (optional)**
 - Follow the example to create custom instruction for Sigmoid
- **==== Tea break ====**
- **[Demo] QEMU full system emulator**
 - Boot linux in QEMU
 - Linux binary vs. newlib binary
- **[Lab] compile C code on QEMU (optional)**
 - Boot linux and compile C code

Self-Introduction

王 君 (Jim Wang)

<http://phdbreak99.github.io>

本科：清华大学电子工程系

博士：中科院计算所龙芯实验室

工作：

- Samsung: senior design engineer
- Marvell: design manager
- Startup: founding engineer & chief architect
- Some 404 company: SoC lead & CPU IP DRI

方向：

- CPU & SoC architecture
- Software hardware co-design

