# Libre-SOC: From architecture and simulation to test silicon, and beyond

## A design for a fully documented and transparent hybrid CPU-GPU-VPU core, for a family of System-on-Chip products

Cesar Strauss
cestrauss@gmail.com


Sponsored by NLnet

FOSDEM 2023

# What is Libre-SOC?

- A hybrid CPU-GPU-VPU core
- A family of System-on-chip products
- For: Routers, cellphones, laptops, HPC
- POWER instruction set
- Innovative vector extension
- Additional 3D opcodes

# Why do we need it?

- Free digital hardware design (LGPL)
- Avoid binary blobs and proprietary drivers
- Fully Documented (Open Standards, no NDA, no RE)
- Transparent and trustworthy (no vendor lock-in)

# How to pull it off?

- Grants and Donations
- Community of experts (Usenet, IRC, Research)
- Commercial partners

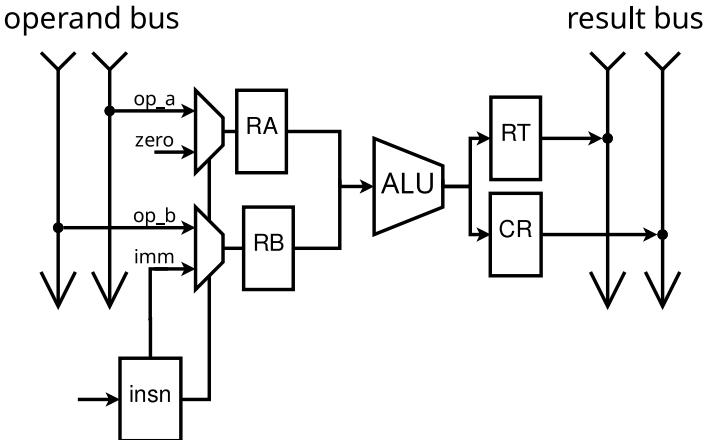
# Multi-issue, out-of-order execution Architecture

- Fetch $\rightarrow$ Decode $\rightarrow$ Issue $\rightarrow$ Execute pipeline
- Parallel decoders
- Vectorized issue
- Parallel execution units
- Scoreboard dependency tracking and hazard avoidance
- Branch prediction / $\mu$-op cache?

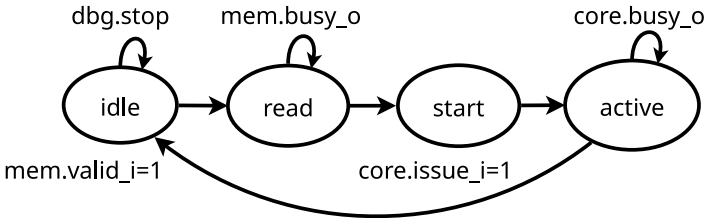
# Staged development

- From machine readable specification to simulation and implementation
- CSV tables → Auto-generated decoder
- Pseudo-code → Auto-generated simulator
- Unit tests: QEMU → ISA simulator → HDL simulator → FPGA → ASIC
- Development: Function Units → Comp. Units → FSM core → In-order core

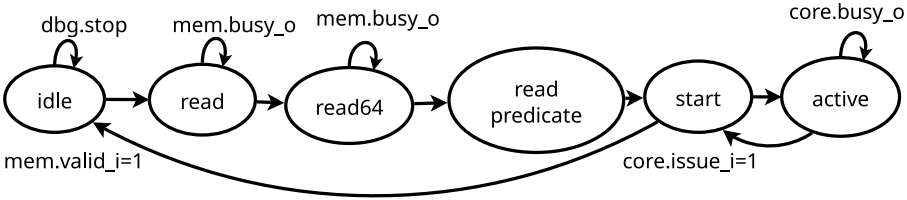
# Comp Unit Formal verification



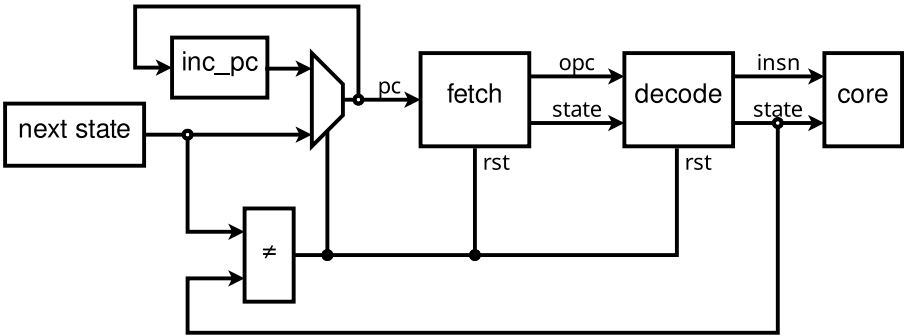
# Simple Core


dbg.stop
mem.busy_o
core.busy_o
idle
read
start
active
mem.valid_i=1
core.issue_i=1

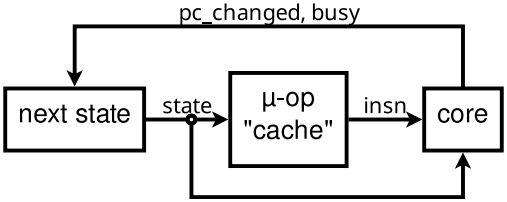
# SVP64 Core


dbg.stop
mem.busy_o
mem.busy_o
core.busy_o
idle
read
read64
read predicate
start
active
mem.valid_i=1
core.issue_i=1

# In-order Core



# In-order Core with $\mu$-op cache



# Where we are

- Reproducible developer environment (Debian chroot)
- FPGA booted
  - Bare-metal (serial console)
  - Zephyr real-time OS (networked)
  - Linux (serial console)
- Test silicon produced
- Binutils port (incl. new vector instructions)
- new instructions being submitted to OpenPOWER foundation
- Crypto and Multimedia algorithms being ported and simulated

# Where we are going

- Port and boot a Linux distro on FPGA with our Core
- Full toolchain (incl. vectorization)
- Further define the ISA extensions
- What we need
  - Developers (SW and HW), Doc. writers and Testers!

# Chip Production:


"Our mission is to deliver a high performance microprocessor family (Vantage) optimised for our Revolutionary Vector Instruction Prefix (Vector +1) for the POWER® ISA that speeds up product development, increases performance and cuts power consumption for complex computation applications."

Look for them among the audience!

# The end

# Thank you

# Questions?

- Main page: `https://libre-soc.org`
- Source: `https://git.libre-soc.org`
- Discussion: `http://lists.libre-soc.org`
- LiberaChat IRC #libre-soc
- Sponsored by NLnet
- http://nlnet.nl/PET
- https://libre-soc.org/nlnet/#faq