Muntjac – Open Source Multicore RV64 Linux-capable SoC

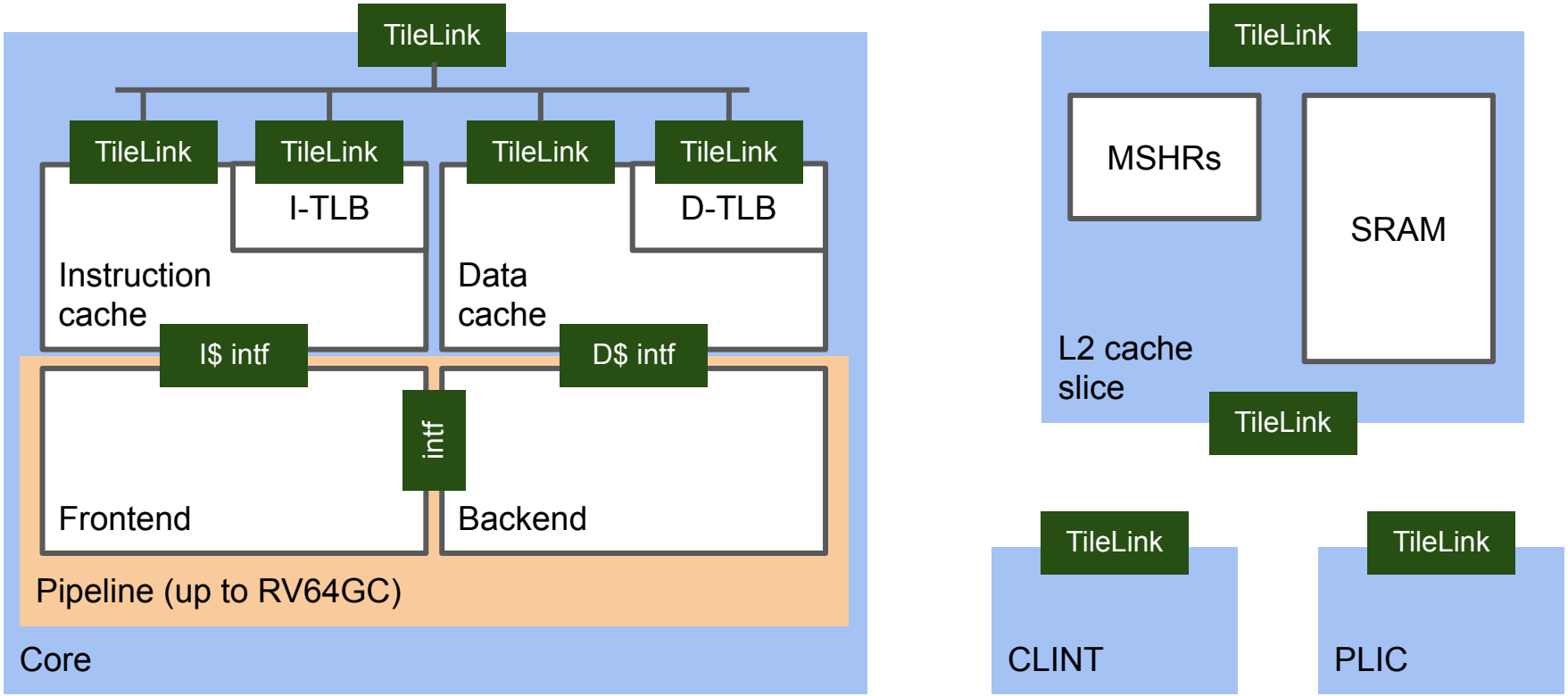
Xuan (Gary) Guo, **Daniel Bates**,
Robert Mullins, Alex Bradbury

Motivation


EXTRACTOR
How to get the most from your Integrated Extractor
APP 2314 (Mocca)
APP 2315 (Mocca)
APP 2305 (White)
APP 2310 (Silver Grey)


RUBICON
Jeep

Overview: components

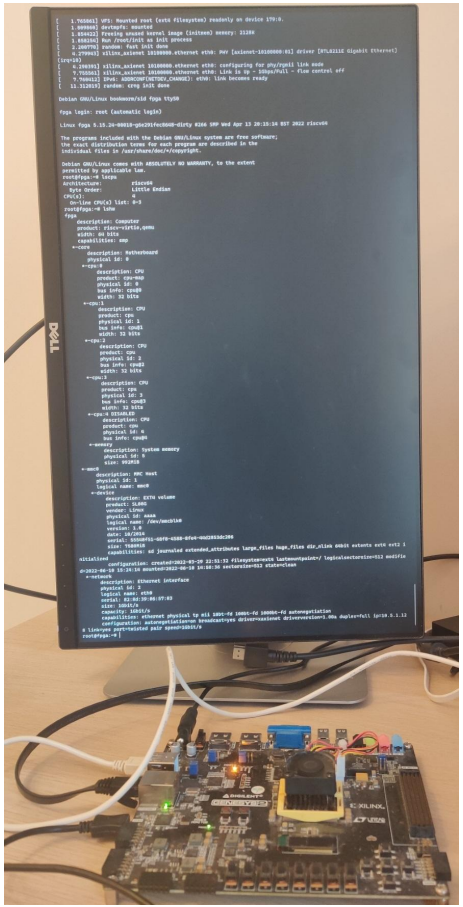


Overview: standards

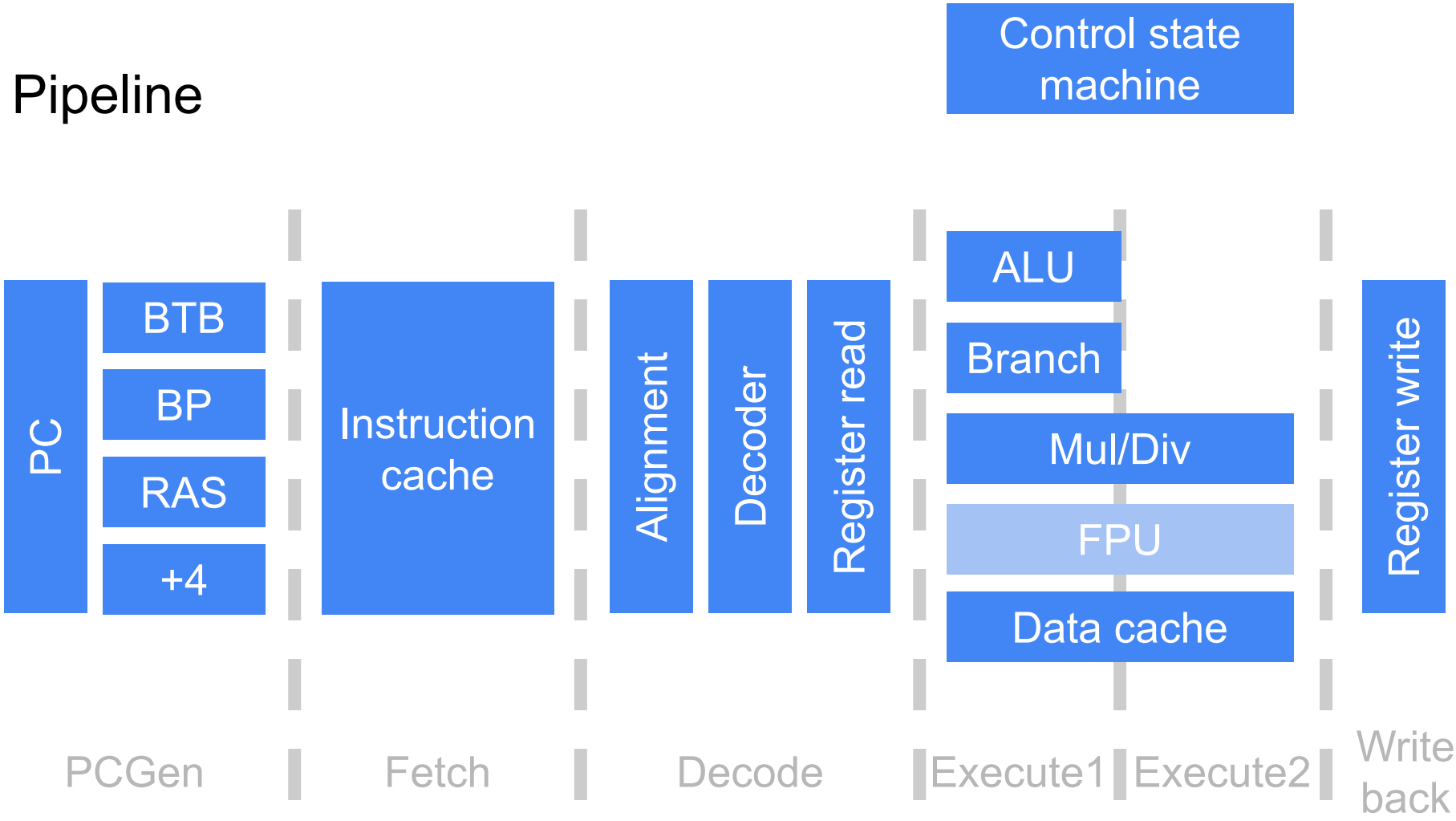
Standard	Description	Version
RV64I	Base integer instruction set	2.1
RV64M	Multiplication and division	2.0
RV64A	Atomic memory operations	2.1
RV64F RV64D	Floating point instructions	2.2 (optional)
RV64C	Compressed instructions	2.0
Zicsr	Control and status registers (CSRs)	2.0
Zifencei	Instruction fetch fence	2.0
Machine ISA Supervisor ISA	M/S/U privilege levels, Sv39 virtual addressing	1.11
TileLink	On-chip interconnect protocol	1.8.1

Overview: out of the box

- Verilator simulators
- FPGA tutorial
  - Boot Linux on a dual core system
- Documentation
- Lots of tests + scripting

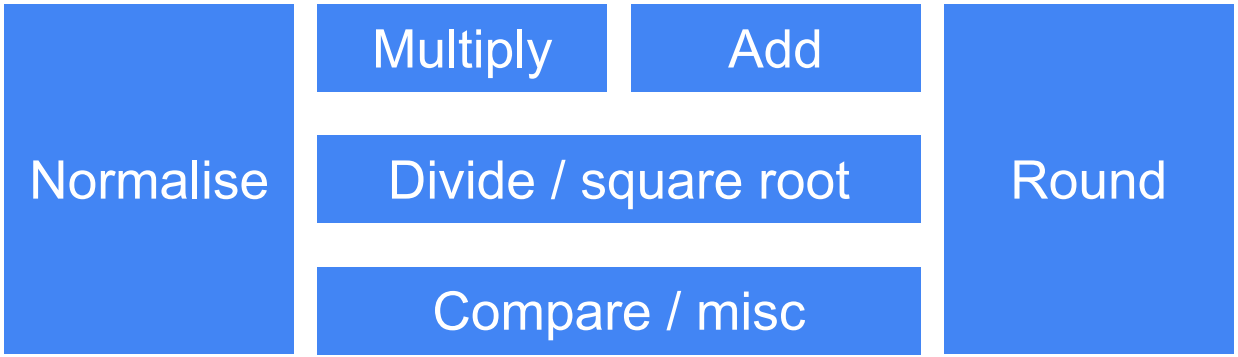


Pipeline

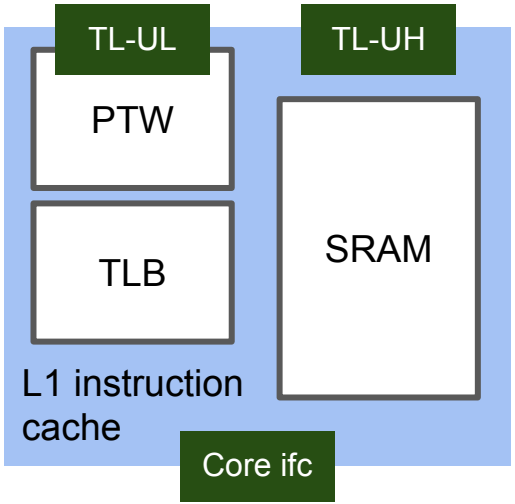

Control state machine
PC
BTB
BP
RAS
+4
Instruction cache
Alignment
Decoder
Register read
ALU
Branch
Mul/Div
FPU
Data cache
Register write
PCGen
Fetch
Decode
Execute1
Execute2
Write back

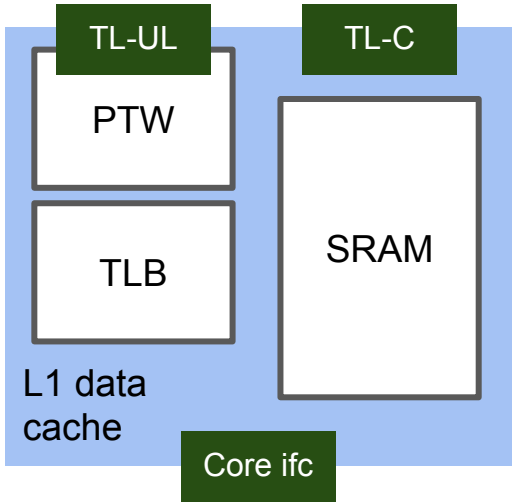
Pipeline: FPU

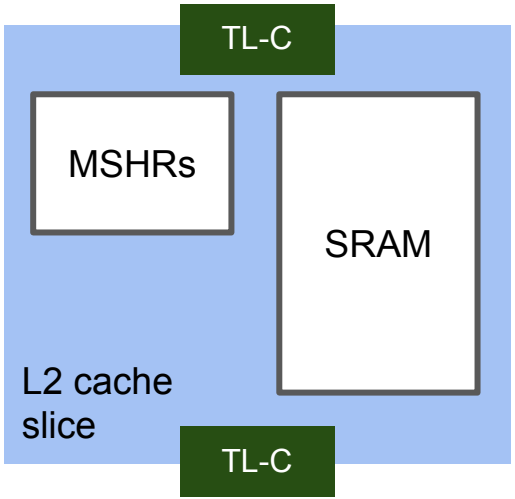
- Clean-sheet implementation
- Configurable: on, off, registers only
- No re-coding: simpler
- Test by booting Linux and running TestFloat



Caches







Defaults:

16KB
4-way associative
MESI coherence
1* cycle

64KB
4-way associative
2 MSHRs per bank
~10-20* cycles

TileLink

We provide many verified TileLink components to allow users to build custom interconnect.

Parameter adapters

- 📈 Increase data width
- 📉 Reduce data width
- ⏫ More message IDs
- ⏬ Fewer message IDs
- ⏪ Shift ID range
- ➗ Reduce max message length

Protocol adapters

- 🔗 TL-C <-> TL-UH
- 💱 TileLink <-> AXI
- 🪨 TileLink <-> FPGA BRAM

Topology

- ▶️ Multiplex
- ◀️ Demultiplex
- 🔊 Broadcast

Buffers

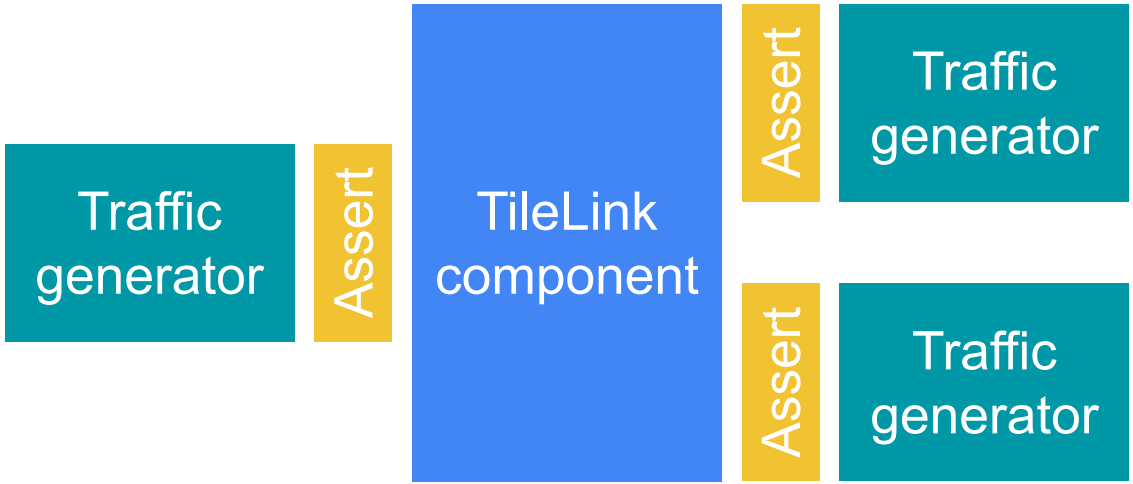
- 🕛 Clock domain crossing
- ⏳ Timing isolation
- 🔁 Ensure FIFO responses

Verification: processor

- Lots of high-quality test suites available
  - riscv-tests (ISA compliance)
  - riscv-dv (random tests - find corner cases)
- Boot Linux + run multithreaded stress tests (e.g. PARSEC)
- >90% line coverage
  - Most lines missing are unused “default” cases in switch statements
- More work planned
  - Next: functional coverage

Verification: TileLink

- No open source test suites available?
  - Make one ourselves!
- >99% line coverage, very high functional coverage



Summary

- Muntjac is a 64-bit RISC-V processor which “runs Linux well”
  - 3.0 CoreMarks/MHz, 2.2 Dhrystone MIPS/MHz
  - ~90MHz on Xilinx Kintex 7 (without FPGA optimisation)
- Easy to understand, modify, and test
- Next steps:
  - Better branch prediction
  - Dual issue
  - Higher performance memory system
  - More testing/verification
  - ASIC work

Thank you

Talk to us if Muntjac might be useful for your projects, or if we should be aware of something you're doing!

Many thanks to lowRISC for the support.

github.com/lowRISC/muntjac

Gary.Guo@cl.cam.ac.uk

Daniel.Bates@cl.cam.ac.uk

Robert.Mullins@cl.cam.ac.uk

asb@asbradbury.org