2020 Architecture Research with RISC-V and Chisel

Yunsup Lee
Andrew Waterman
Jonathan Bachrach, Scott Beamer, David Patterson, Krste Asanovic

Facilitating Credible Architecture Research

- Simulators are useful for measuring cycle counts—useless for power, area, cycle time
- But microarchitects must balance all of these!
- Need an end-to-end research methodology to close the feedback loop…
- …but traditional HW design approaches with commercial ISAs are far too tedious

RISC-V ISA

31 27	26 22	21 17	16 12	11 10	9 7	6 0	
rd	rs1	rs2	funct10			opcode	R-type
rd	rs1	rs2	rs3	funct5		opcode	R4-type
rd	rs1	imm[4:0]	imm[11:5]		funct3	opcode	I-type
imm[4:0]	rs1	rs2	imm[11:5]		funct3	opcode	B-type
rd	LUI-imm[19:0]					opcode	L-type
jump offset [24:0]						opcode	J-type

inst[4:2] inst[6:5]	000	001	010	011	100	101	110	111 (> 32)
00	LOAD	LOAD-FP	MISC-MEM		OP-IMM	MISC	OP-IMM-32	
01	STORE	STORE-FP	AMO		OP	LUI	OP-32	
10	MADD	MSUB	NMSUB	NMADD	OP-FP			
11	BRANCH	JALR	J	JAL		SYSCALL		

- Straightforward 32-bit instruction encoding
- Supports compressed encodings and extended-length instructions

Why a new ISA?

- Provide a ***realistic*** but ***open*** ISA that captures important details of commercial general-purpose ISA designs and that is suitable for hardware implementation.
- Provide a ***small*** but ***complete*** base ISA that avoids "over-architecting" for a particular microarchitecture style or implementation technology, but which allows efficient implementation in any of these.

RISC-V Software Stack

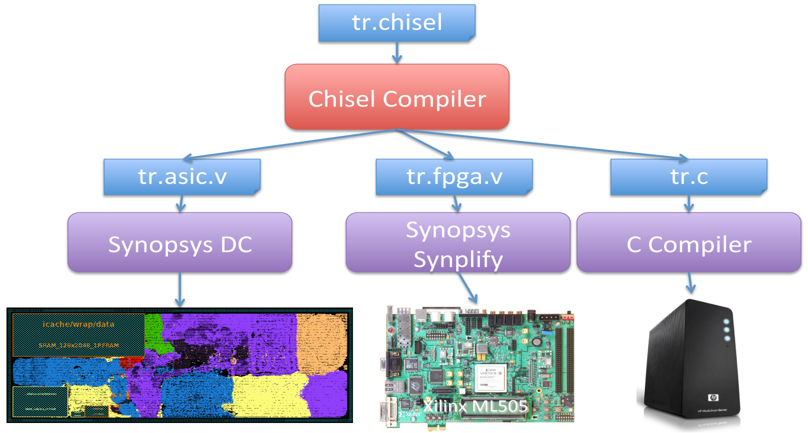
- GCC 4.4 with Newlib C library
- "Proxy Kernel" to support POSIX calls by forwarding to a Linux host machine
- Coming soon, Linux and ROS/Tessellation

Chisel

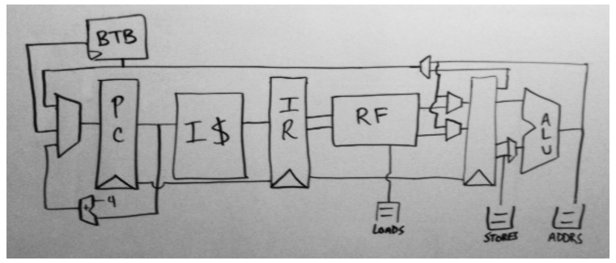
- Is a **H**ardware **C**onstruction **L**anguage
- Is ***not*** a **H**ardware **D**escription **L**anguage: no un-synthesizable constructs
- Is type-safe and more expressive than Verilog
- Is embedded in Scala

```
class Cpu extends Comp {
  override val io: ProcLines
    = new ProcLines();
  def init () = {
    val c = new CtlPath();
    val d = new DatPath();
    c.io.ctl      <> d.io.ctl;
    c.io.dat      <> d.io.dat;
    io.imem.ctl >< c.io.imem;
    io.imem.dat >< d.io.imem;
    io.dmem.ctl >< c.io.dmem;
    io.dmem.dat >< d.io.dmem;
    io.host      >< d.io.host;
  }
}
```

Chisel Backends

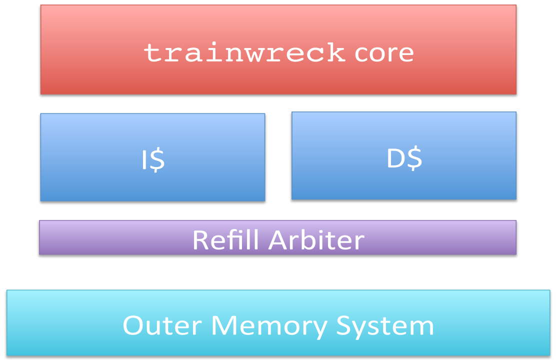


trainwreck: A RISC-V Prototype

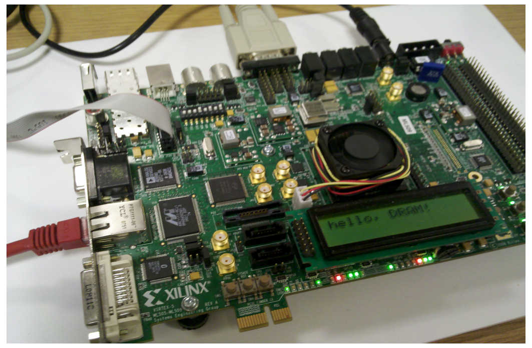


- Simple 3-stage pipeline with decoupled data memory interface
- 64-bit fixed-point datapath, double-precision FP

Target Machine



DEMO



- Hand written Trainwreck on FPGA & ASIC
- Chiseled Trainwreck on FPGA & ASIC