Chipyard: A RISC-V Development Framework

ECE Day 2021

Alexander Lukens Karl Hallsby
Faculty Advisor: Dr. Jia Wang

Illinois Institute of Technology

April 9, 2021

About Us

Alexander Lukens

B.S. Electrical Engineering
M.S. Computer Engineering

Karl Hallsby

B.S. Computer Engineering
M.S. Computer Engineering

What is RISC-V?

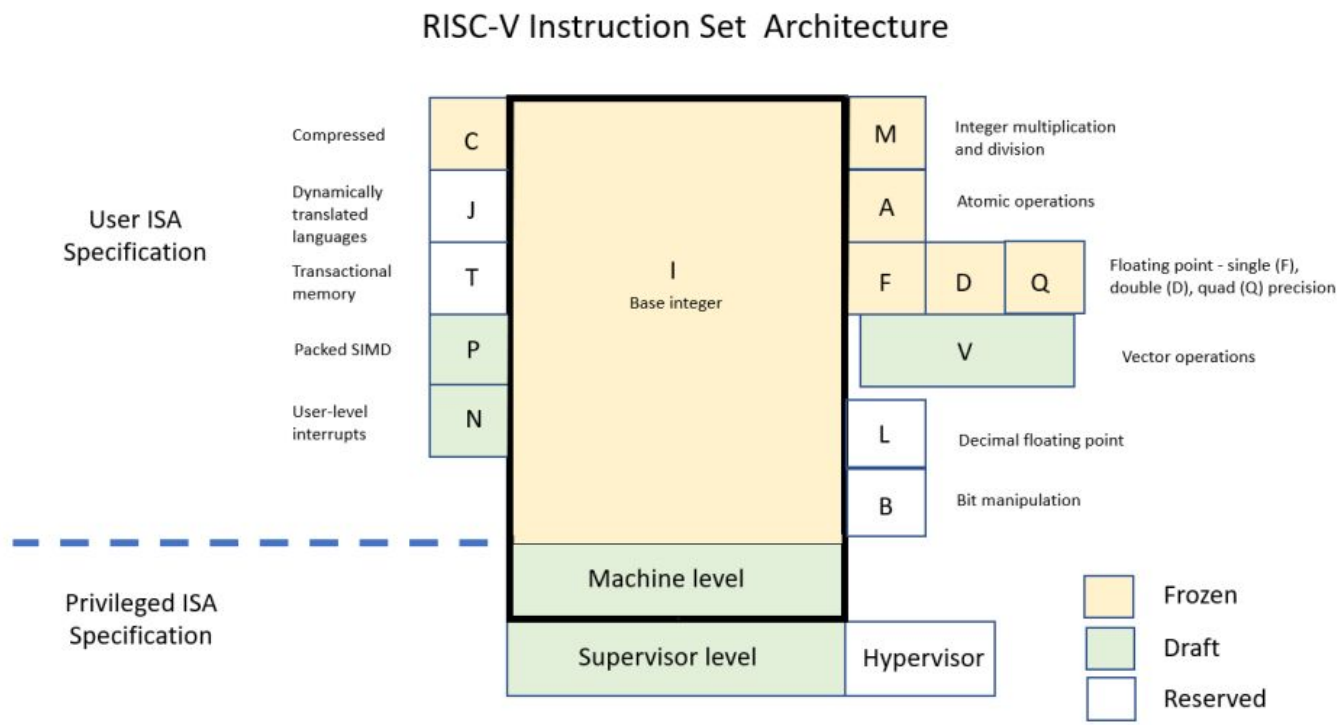

Figure: RISC-V ISA and Standard Extensions (Quinnell 2019)

- Suitable for use in applications ranging from microcontrollers to supercomputers
- Goal $\rightarrow$ become the industry standard ISA (UC-Berkeley 2016).

What is Chipyard?

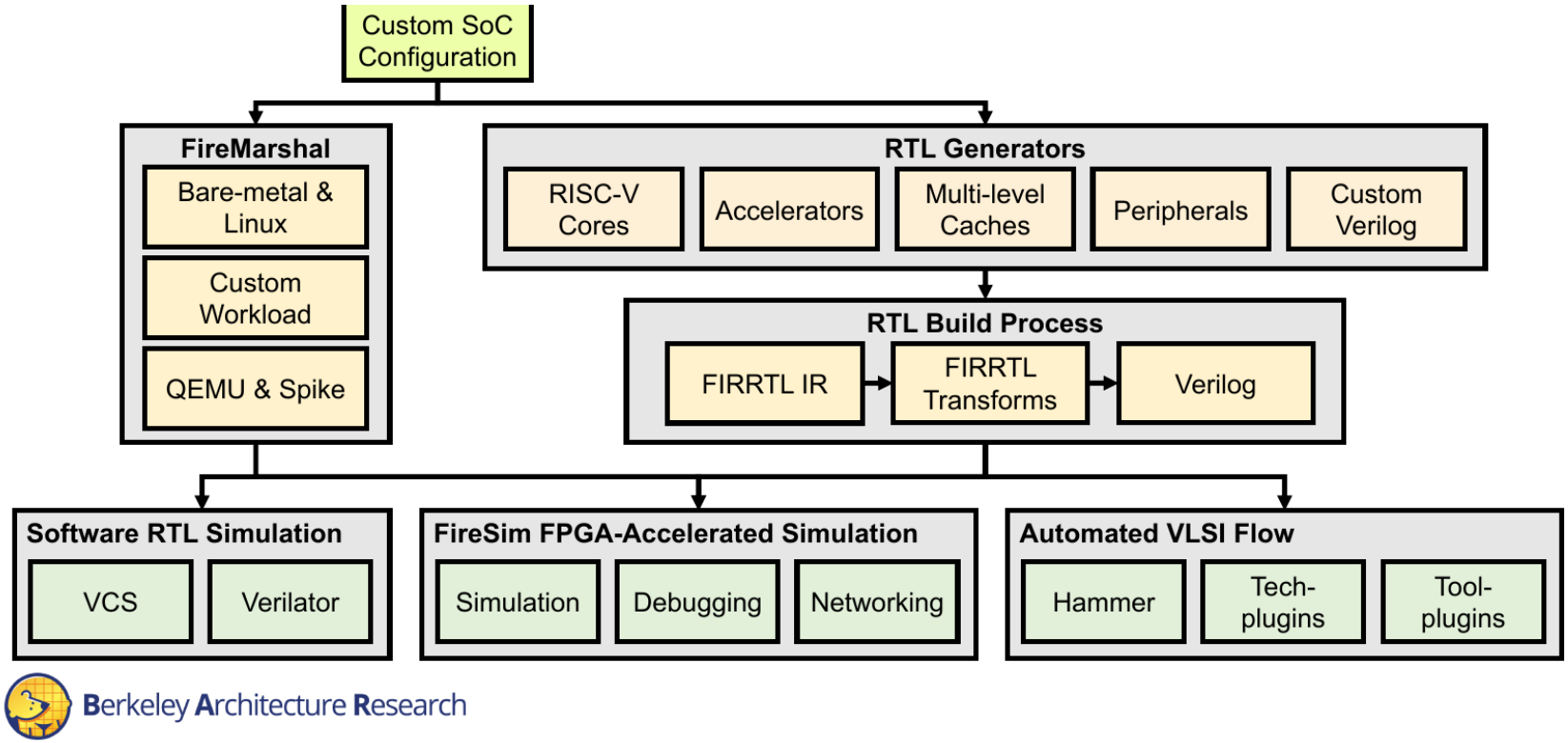

Figure: Chipyard Subcomponents (Gonzalez 2019, p. 3)

- A framework for generating and testing RISC-V CPUs
- Elaborate CPUs using declarative Scala which generates appropriate Verilog
  - Can also generate FPGA bitstreams, `mcs` files, and other data formats

What We Did

- Explore repository to more fully understand how everything works together
- Build standard CPUs, already defined in Chipyard
- Create configurations to generate custom CPUs
- Write standard CPUs out to FPGAs for testing
  - Identified issues with Chipyard's implementation of Arty FPGA support
- Upload programs to built FPGA CPU

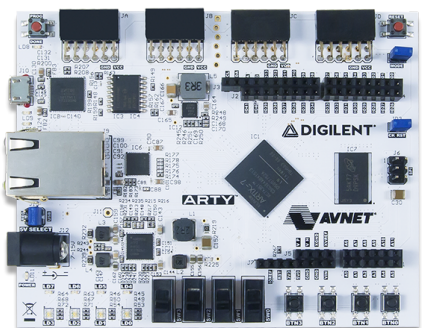

Figure: Arty FPGA

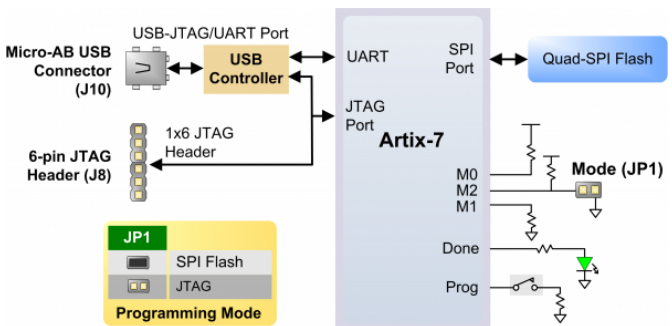

Figure: Arty UART

What We Learned

- Chipyard is under active development
- The Verilog simulation of RISC-V processors in Chipyard is very robust, allows for comprehensive testing of a design to ensure ISA compliance

```
commit e3d23f10345cfafd9a2f4f8aacfe07c2746699f9 (HEAD -> master, origin/master,
Merge: 58076cfb 7f5d8c8d
Author: Abraham Gonzalez <abe.j.gonza@gmail.com>
Date:   Mon Mar 8 15:39:27 2021 -0800

    Merge pull request #805 from ucb-bar/davidbiancolin-patch-1

    [FireChip] Update frequency selection related comments

commit 7f5d8c8dba9600a3ab47fae955802c1cf8c96290 (origin/davidbiancolin-patch-1)
Author: David Biancolin <david.biancolin@gmail.com>
Date:   Wed Feb 24 13:05:14 2021 -0800

    Update frequency selection related comments

commit 58076cfb260a3be502d6d1c25b577da39277a7fc (tag: 1.4.0)
Merge: cca3cd08 d4f8f564
Author: Jerry Zhao <jerryz123@berkeley.edu>
Date:   Wed Jan 20 15:26:47 2021 -0800

    Merge pull request #599 from ucb-bar/dev

    Chipyard 1.4.0 Release

commit d4f8f56465bd8be6e6c8dafc859322cc204d0d62
Merge: a7652cee cc580488
Author: Jerry Zhao <jerryz123@berkeley.edu>
Date:   Tue Jan 19 22:34:22 2021 -0800

    Merge pull request #767 from ucb-bar/rel-changelog

    v1.4.0 Release changelog

commit cc580488e4f70b007c99421f00802b6007e3ec14
Author: Jerry Zhao <jerryz123@berkeley.edu>
Date:   Tue Jan 19 22:24:28 2021 -0800

    Update CHANGELOG
```

Figure: Chipyard Git Log

Educational Laboratory Work

- Open nature of RISC-V allows for easy, license-free tinkering of CPU designs
- Strong documentation of RISC-V architectures due to industry use
- Open development allows for easy tracking of changes made to underlying CPU designs
- Other universities already use Chipyard and RISC-V for undergraduate computer architecture courses

Further Research Work

- Increase compatability with components on Arty FPGA board
- Design and integrate GPIO device that will assist with debugging
  - Single-step bus cycle, display current memory address, program counter, register contents, like SANPER-1
- Investigate VLSI Design flow in Chipyard

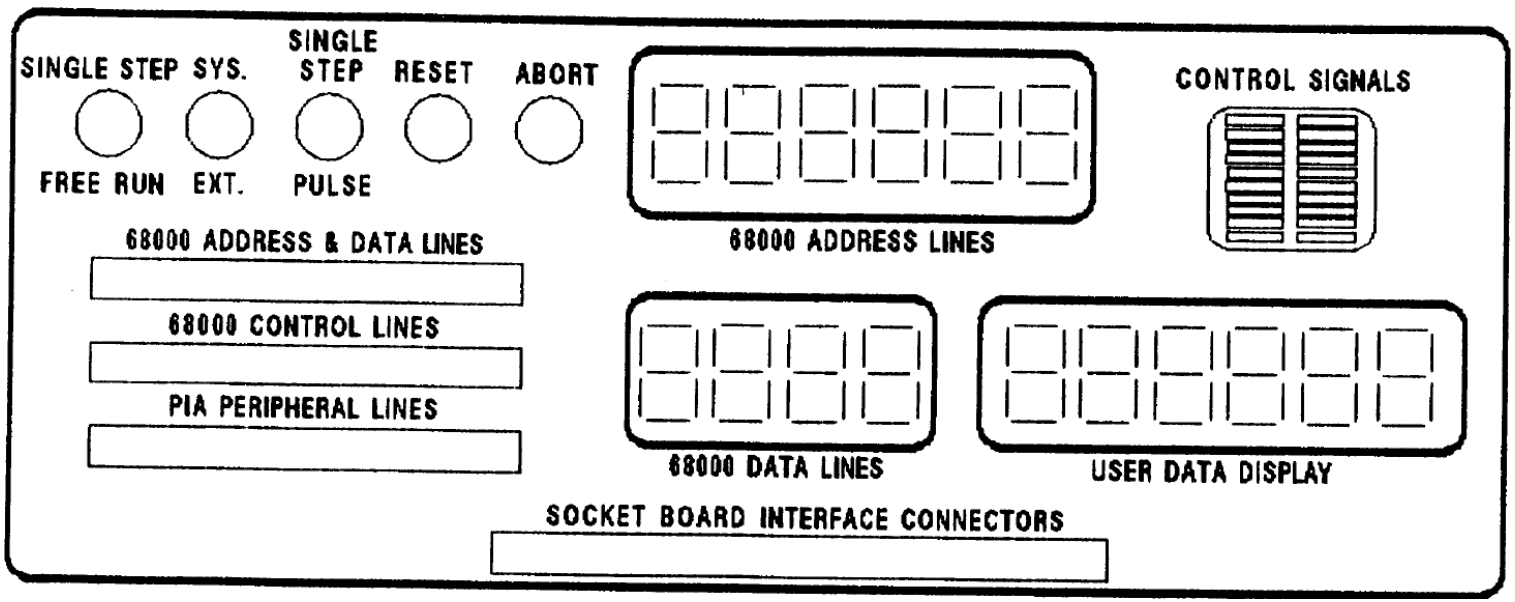

Figure: SANPER-1 Front Panel (Saniie and Perich 1990, p. 22)

Conclusion/Deliverables

- Chipyard offers many tools that greatly enhance the development of larger RISC-V CPU designs
- We are preparing a documentation manual covering everything we did. This includes:
  - Environment Setup & Configuration
  - Basic Steps for Generating a first CPU
  - A walkthrough of the Chipyard repository's structure
  - A discussion on how to write FPGA images out and use them for testing

Demonstration

References

Alon Amid et al. "Chipyard: Integrated Design, Simulation, and Implementation Framework for Custom SoCs." In: *IEEE Micro* 40.4 (2020), pp. 10–21. ISSN: 1937-4143. DOI: 10.1109/MM.2020.2996616.

Abraham Gonzalez. *Building Custom RISC-V SoCs in Chipyard*. 2019. URL: https://fires.im/micro19-slides-pdf/03_building_custom_socs.pdf.

Rich Quinnell. "Creating a Custom Processor with RISC-V." English. In: *EETimes* (Mar. 29, 2019). URL: https://www.eetimes.eu/creating-a-custom-processor-with-risc-v/ (visited on 04/01/2021).

Dr. Jafar Saniie and Stephen H. Perich. *SANPER-1 Educational Lab Unit. User's Manual and Lab Experiments*. 68000 Based Microcomputer. 1990.

UC-Berkeley. *RISC-V State of the Union*. 2016. URL: https://riscv.org/wp-