ThoughtWorks®

PERC: Posit Enhanced Rocket Chip

CARRV at ISCA 2020

22 May 2020

Arunkumar M. V., Sai Ganesh Bhairathi, Harshal G. Hayatnagarkar
Engineering for Research, ThoughtWorks Technologies India

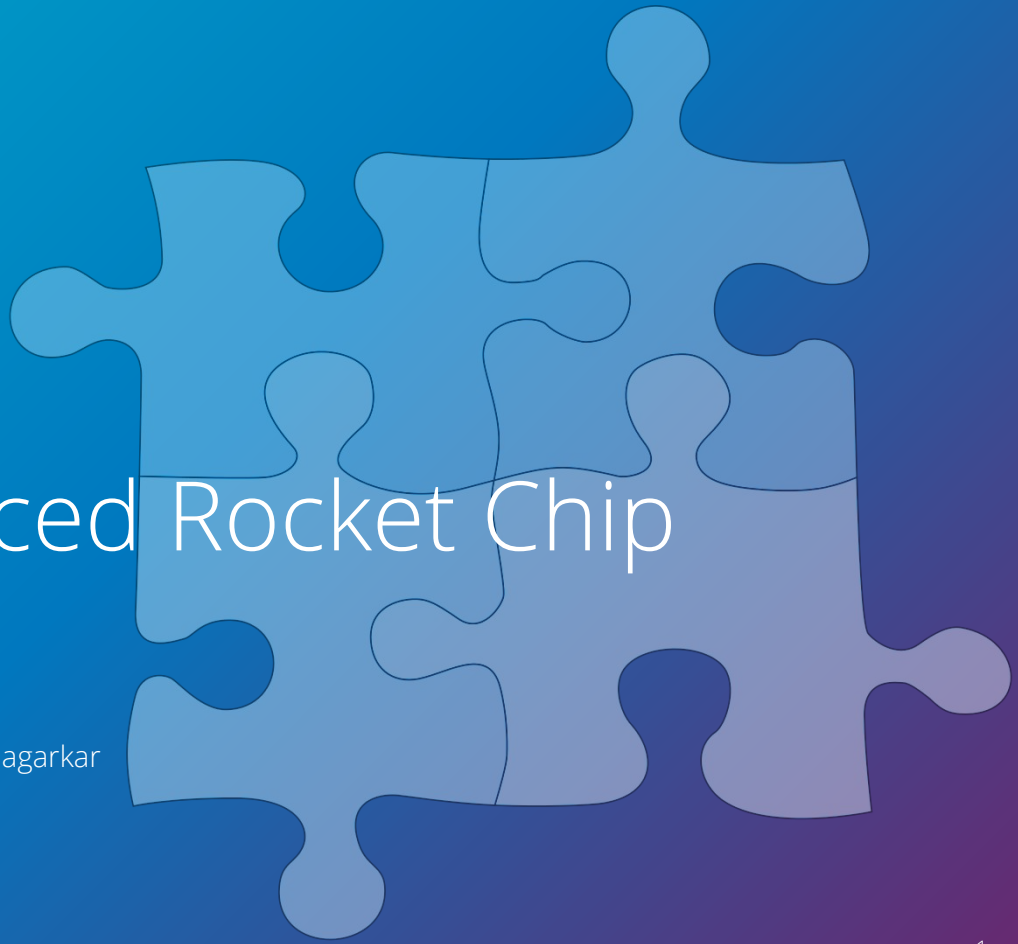


Image Credits: https://commons.wikimedia.org/wiki/File:Puzzle-4.svg

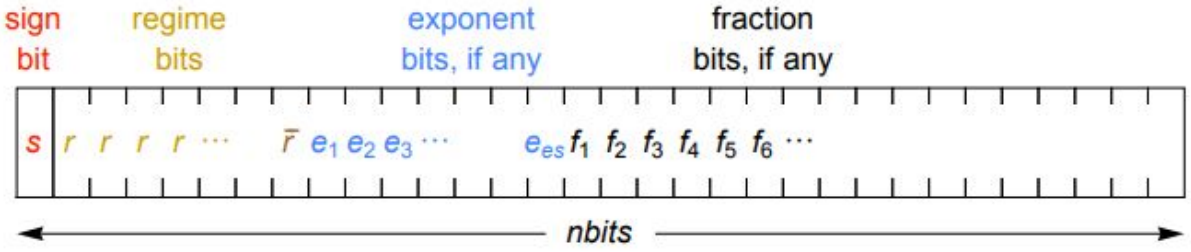
Motivation

A floating point architecture

geared towards lossless number crunching

for scientific computing

The Posit number format



- Introduces an additional field called regime.
- Run length encoded regime to scale the exponent.
- Only 2 exceptional representations, one for zero and another for ±infinity.

Why Posits?

- Crisp definition of NaR saves precious bits
- Symmetric and verifiable design
- Customizable precision and dynamic range
- Hardware-friendly number system

Why did we choose RISC V and Rocket Chip?

- For being an open ISA with support for modular extensions and custom opcode spaces
- Accompanied by a mature ecosystem of hardware IPs, software toolchains, and operating systems

- A collection of parameterized libraries of SoC components which can be “glued together” during circuit generation.
- Ability to add custom accelerators/coprocessors using the RoCC interface.

The Posit Hardware Units

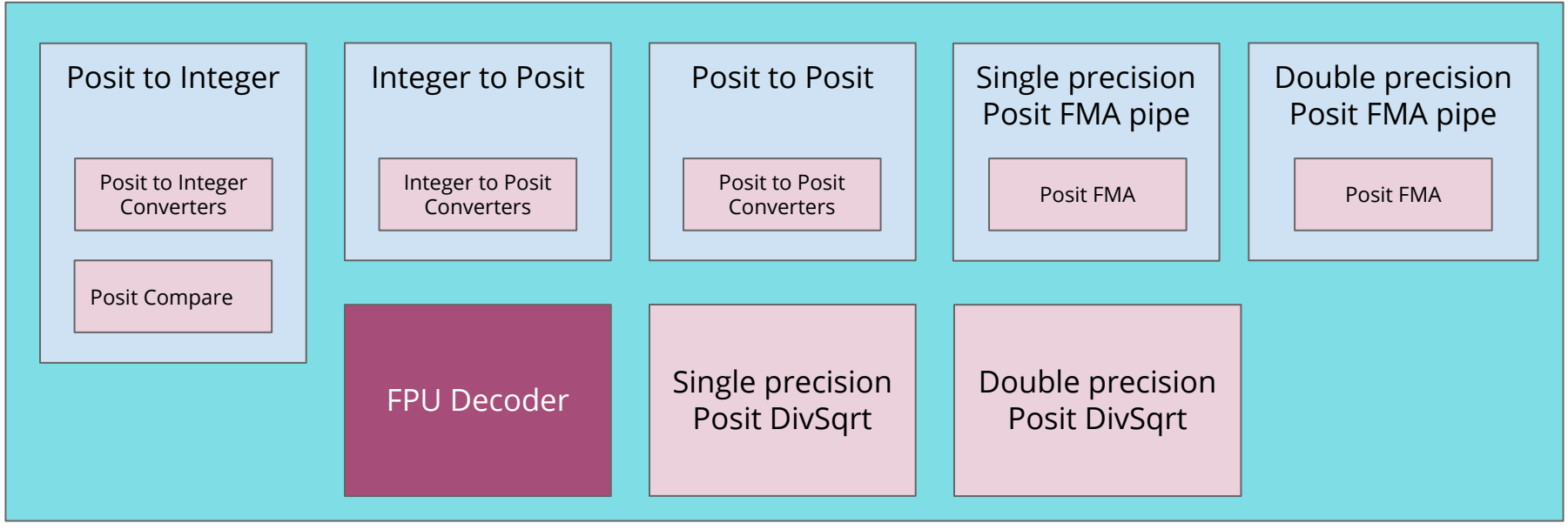
- Designed using Chisel.
- Conceived as a drop-in replacement for Berkeley Hardfloat.
- Parameterized units for different posit sizes and exponent sizes.
- Tested using Universal Numbers C++ template library enabled by Verilator.

RISC V support for Posits

- 2 possible approaches to provide posit ISA support:
  - Overload the F and D extensions.
  - Utilize the custom instruction opcode space.
- Our approach: Overload the existing ISA extensions.
  - Involves minimal variation to the existing RISC V software toolchain.
  - Evaluate an alternate FPU in the Rocket core pipeline.

The Posit Processing Unit

Combines the various posit hardware units and serves as an alternate FPU



Posit configuration of ps = 32 and es = 2 for single precision (F instr)
ps = 64 and es = 3 for double precision (D instr)

Verifying the PPU

- Modified the RISC V ISA test suite and added tests using F and D instructions for posits

- Tests were run using the Rocket chip verilator emulator on a circuit instance with the in-house FPU replaced with the PPU.

Early synthesis results

The PPU occupies ~15949 slice LUTs on a *Spartan 7* FPGA compared to ~12659 slice LUTS of Rocket Chip's optimized FPU.

Observations

- The Rocket Chip journey
- Using Chisel and the ecosystem
- Missing compiler support for posit arithmetic
- RISC V extension for posit arithmetic

Conclusion and future work

Open source posit arithmetic chisel library integrated with Rocket chip

- Drop-in replacement of Berkeley hardfloat (F and D instructions)
- Verified using Verilator and custom test-generator

Work-in-progress

- Support for quire operations for posit compliance, (not mandated for floats)
- Verify on hardware and optimize resource utilization

Future work

- Posit arithmetic RoCC accelerator
- Compiler support for posit arithmetic

Hope

Once mature,

this work will find its place in Rocket chip

Important links

- Posit Chisel Library Source code on GitHub https://github.com/thoughtworks/hardposit-chisel3
- PERC Source code on GitHub https://github.com/arunkmv/rocket-chip

- Talk on posit arithmetic by Dr. John L. Gustafson https://www.youtube.com/watch?v=aP0Y1uAA-2Y&t=4689s
- Next Generation Arithmetic: Posit Hub https://posithub.org

Thank you!

Arun
arunkumar.mv@thoughtworks.com

Sai
saiganeb@thoughtworks.com

Harshal
harshalh@thoughtworks.com

ThoughtWorks®