SCIENCE ▪ PASSION ▪ TECHNOLOGY

Institute of Technical Informatics ■ Inffeldgasse 16/I, 8010 Graz, Austria ■ *office@iti.tugraz.at*

FazyRV – A Scalable RISC-V Core [1, 2]

Meinhard Kissich

Institute of Technical Informatics, Graz University of Technology
meinhard.kissich@tugraz.at

Marcel Baunach

Institute of Technical Informatics, Graz University of Technology
baunach@tugraz.at

Abstract

FazyRV is an inherently scalable RISC-V RV32I core for control-oriented applications and the IoT. It aims to minimize the area demand while fulfilling performance requirements and to close the gap between prevalent 32-bit and 1-bit-serial RISC-V cores. Thus, the data path can be synthesized to a width of either 1, 2, 4, or 8 bits to process smaller "chunks" of the operands in each clock cycle. Scaling the chunk size allows a trade-off between area and performance at synthesis time. In addition, FazyRV provides manifold variants that can be combined with each data path width to select the best-fitting implementation for the target technology and system requirements.

Design Objectives

- Fulfil performance requirements with minimal area demand
- Target control-oriented applications and the IoT
- **Scalability**: Parametrizable 1, 2, 4, or 8 bit data path width
- **Abstraction**: Avoid hand optimization at the gate level
- **Manifold Variants:**
  - ↪ Feature set vs. area: "*MIN*" ⊂ "*INT*" ⊂ "*CSR*"
  - ↪ Register file: single-port BRAM, dual-port BRAM, or LUT RAM
  - ↪ ...

Related Work

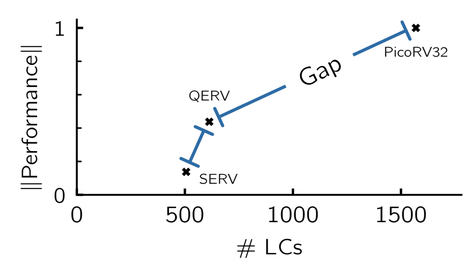
Core	Area Demand
PicoRV32	917 LUTs **(Ultra Scale)**; 761 LUTs 442 Regs **(7-Series)**
VexRiscv	504 LUTs 505 FFs **(7-Series)**; 1130 LCs **(iCE40)**
Ibex	> 2000 LUTs
Micro-riscy	1.6× smaller than Zero-riscy (aka Ibex)
SERV	125 LUTs **(7-Series)**; 198 LUTs **(iCE40)**
QERV	13% larger than SERV

- Difficult Comparison
  - ↪ Various FPGA architectures
  - ↪ Various core extensions and feature sets
  - ↪ Non-identical CAD tools and versions

build minimal **reference SoC**

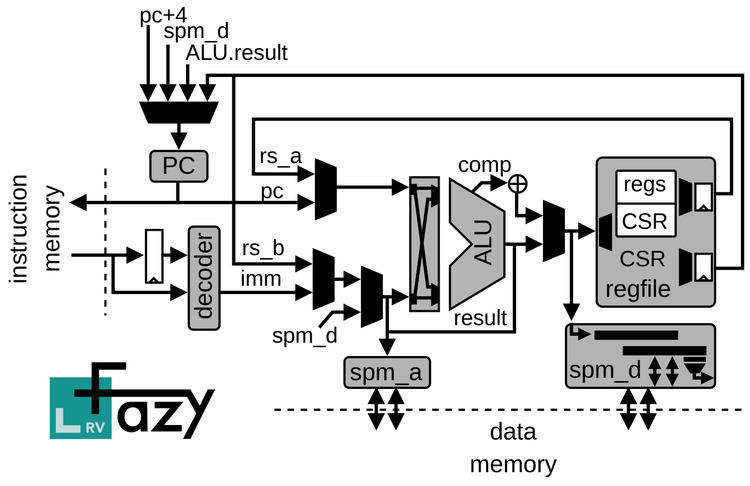
Core in SoC	CSR	# iCE40 LCs	f_{MAX} (MHz)
SERV	✗	506	109
SERV	✓	590	98
QERV	✗	613	86
QERV	✓	717	71
VexRiscv (not productive)	✗	1336	78
PicoRV32	✓	1569	71
VexRiscv (LiteX)	✓	1887	60

- Resource Demand of Minimal Reference SoC

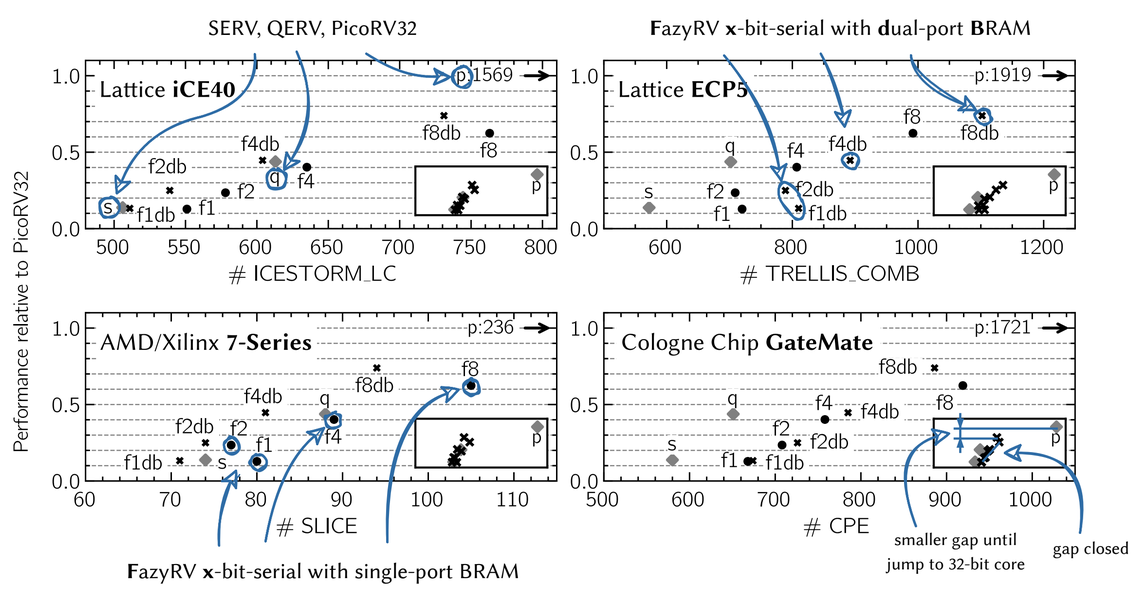


no fine-granular *area vs. performance* scaling possible

Design

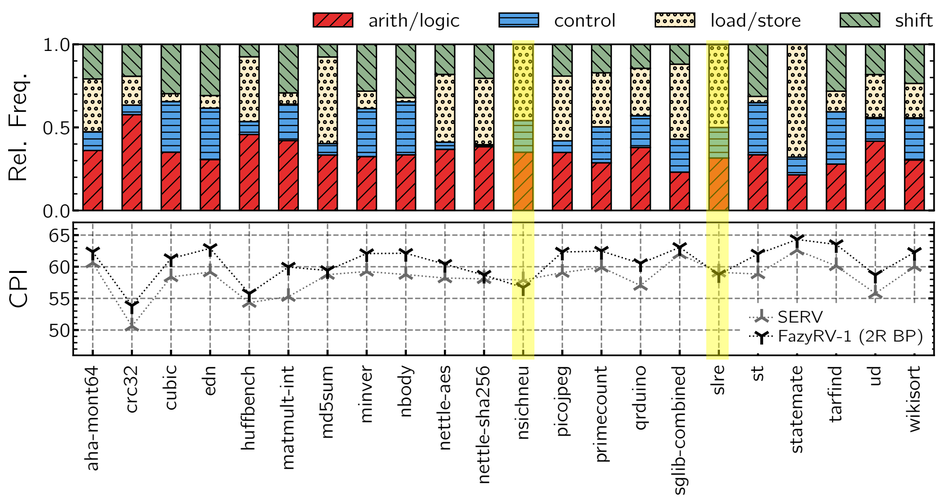


- Block Diagram
  - ↪ `spm_a`: memory address *serial* → *parallel* conversion
  - ↪ `spm_d`: data *serial* ↔ *parallel* conversion, shifting, zero/sign extending
  - ↪ `regfile`: BRAM and/or logic, some CSR must be in logic due to parallel writes

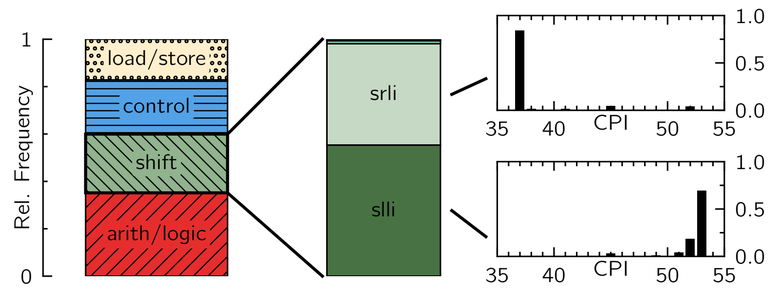


- Area vs. Performance of **Reference SoC**

Evaluation



- Embench Benchmark Results
  - ↪ SERV on average lower CPI than 1-bit-serial FazyRV (bottom plot)
  - ↪ Two exceptions: `nsichneu` and `slre`
  - ↪ Faster benchmarks: low relative frequency of *shift* instructions (upper plot)



- Optimize Shift Instructions
  - ↪ Embench suite [3]: About $^1/_4$ of instructions are shifts
  - ↪ Most shift instructions are I-type shifts (especially `srli` and `slli`)
  - ↪ Left shifts: implemented through cyclic shifts to the right to save area
  - ↪ Conclusion: invest more area to make left shifts with small immediate faster

- Further Links
  - ↪ FazyRV Repository → [1]
  - ↪ FazyRV Paper → [2]
  - ↪ FazyRV-Exotiny SoC + TT06 tapeout → [4, 5]
  - ↪ Yosys Community Spotlight → [6]

References

[1] meiniKi/FazyRV, `https://github.com/meiniKi/FazyRV`
[2] M. Kissich and M. Baunach, "FazyRV: Closing the Gap between 32-Bit and Bit-Serial RISC-V Cores with a Scalable Implementation", in Proc. of the 21st ACM Int. Conf on Computing Frontiers (CF '24), 2024.
[3] Andrew Burgess et al. 2023. Embench™: Open Benchmarks for Embedded Platforms. GitHub. `https://github.com/embench/embench-iot`
[4] meiniKi/FazyRV-ExoTiny, `https://github.com/meiniKi/FazyRV-ExoTiny`
[5] meiniKi/tt06-FazyRV-ExoTiny, `https://github.com/meiniKi/tt06-FazyRV-ExoTiny`
[6] YosysHQ Blog, `https://blog.yosyshq.com/p/community-spotlight-fazyrv`

FazyRV Core