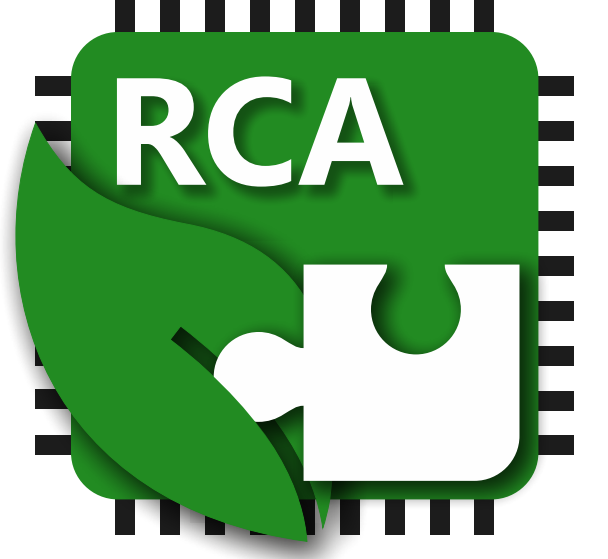


# Runtime Reconfiguration of Decoders in Minimal-area RISC-V Cores

Lukas Glantschnig, Tobias Scheipel

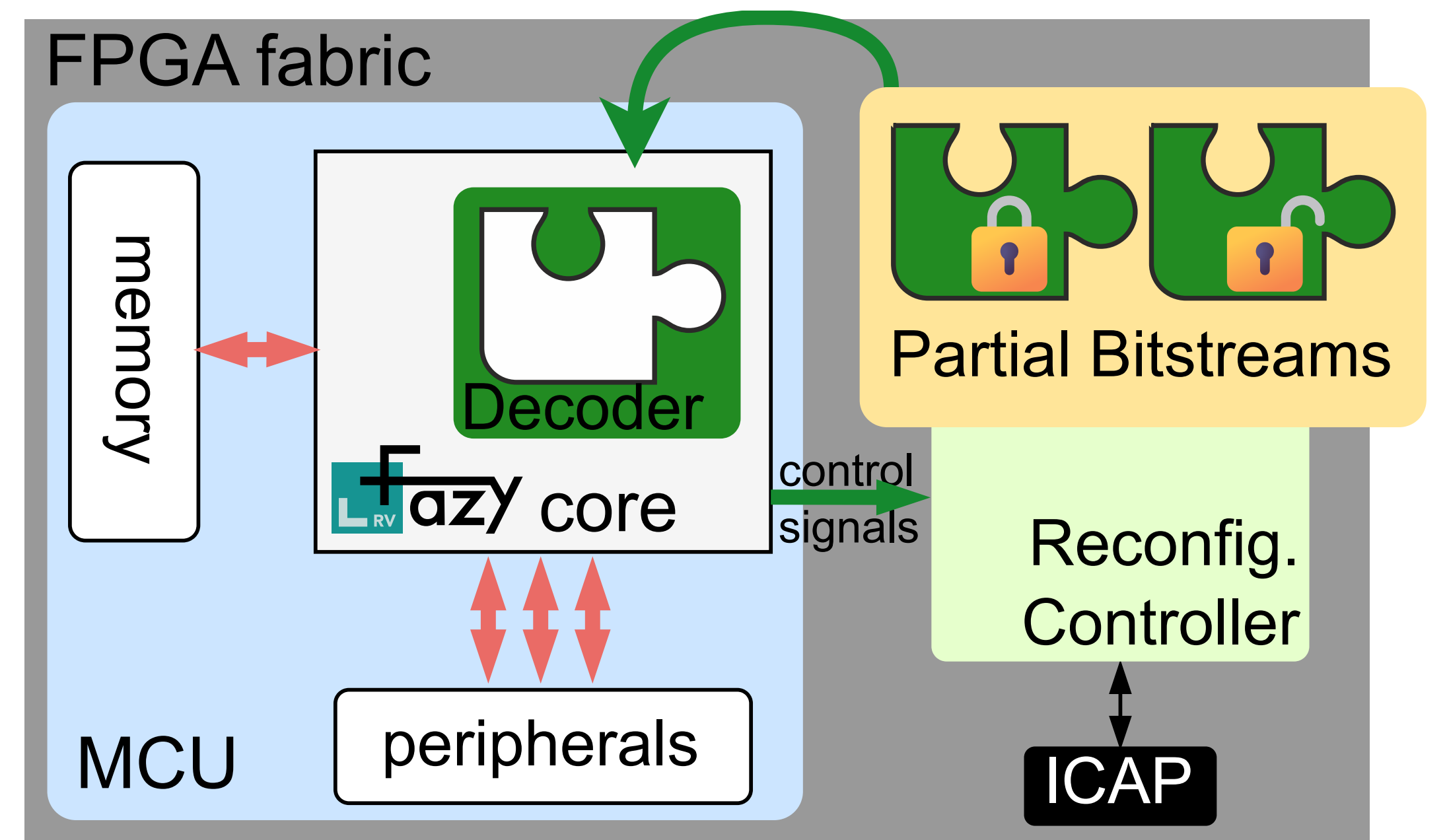
Reconfigurable Computer Architectures, Embedded Architectures & Systems Group, Institute of Technical Informatics, Graz University of Technology  
lukas.glantschnig@student.tugraz.at, tobias.scheipel@tugraz.at



## Key Idea

Use Dynamic Partial Reconfiguration not only for accelerators, but to dynamically exchange the instruction decoder!

Minimal-area RISC-V processor cores rely on architectural trade-offs affecting correctness, robustness, and potentially security-relevant behavior. To mitigate these issues, this work demonstrates FPGA-based partial runtime reconfiguration for dynamically switching between compact and stricter decoder variants on demand, depending on the application requirements.



## Initial Problem & Approach

### ■ Compact Decoder Design:

- FazyRV [1] uses an Espresso-generated minimal-area decoder.
- “Don’t-care” bits reduce logic complexity and hardware area.
- Malformed instructions may therefore trigger unintended behavior.

### ■ Stricter Decoder Variant:

- Additional instruction checks reduce semantic ambiguity.
- Yields improved correctness and robustness.
- Yet, increased hardware resource consumption.

### ■ Runtime Reconfiguration:

- Decoder variants can be exchanged during runtime using Dynamic Partial Reconfiguration (DPR).
- Different code workloads can use different decoder variants.
- Robustness and area efficiency can be balanced dynamically,

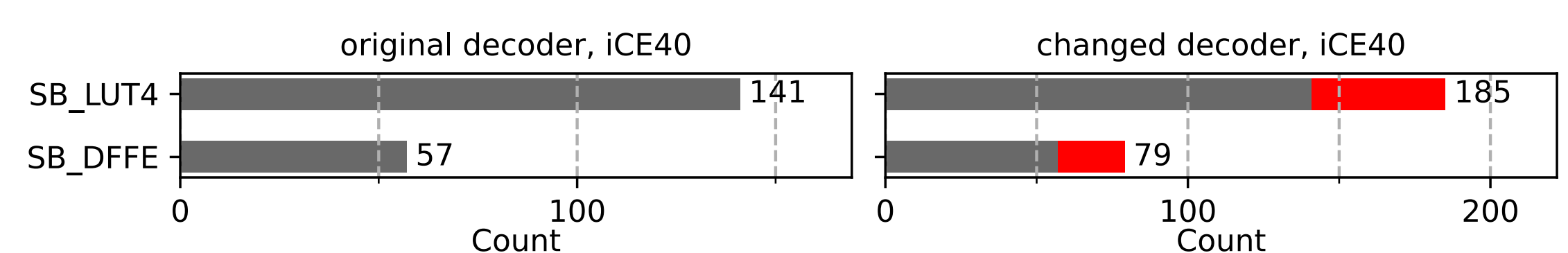
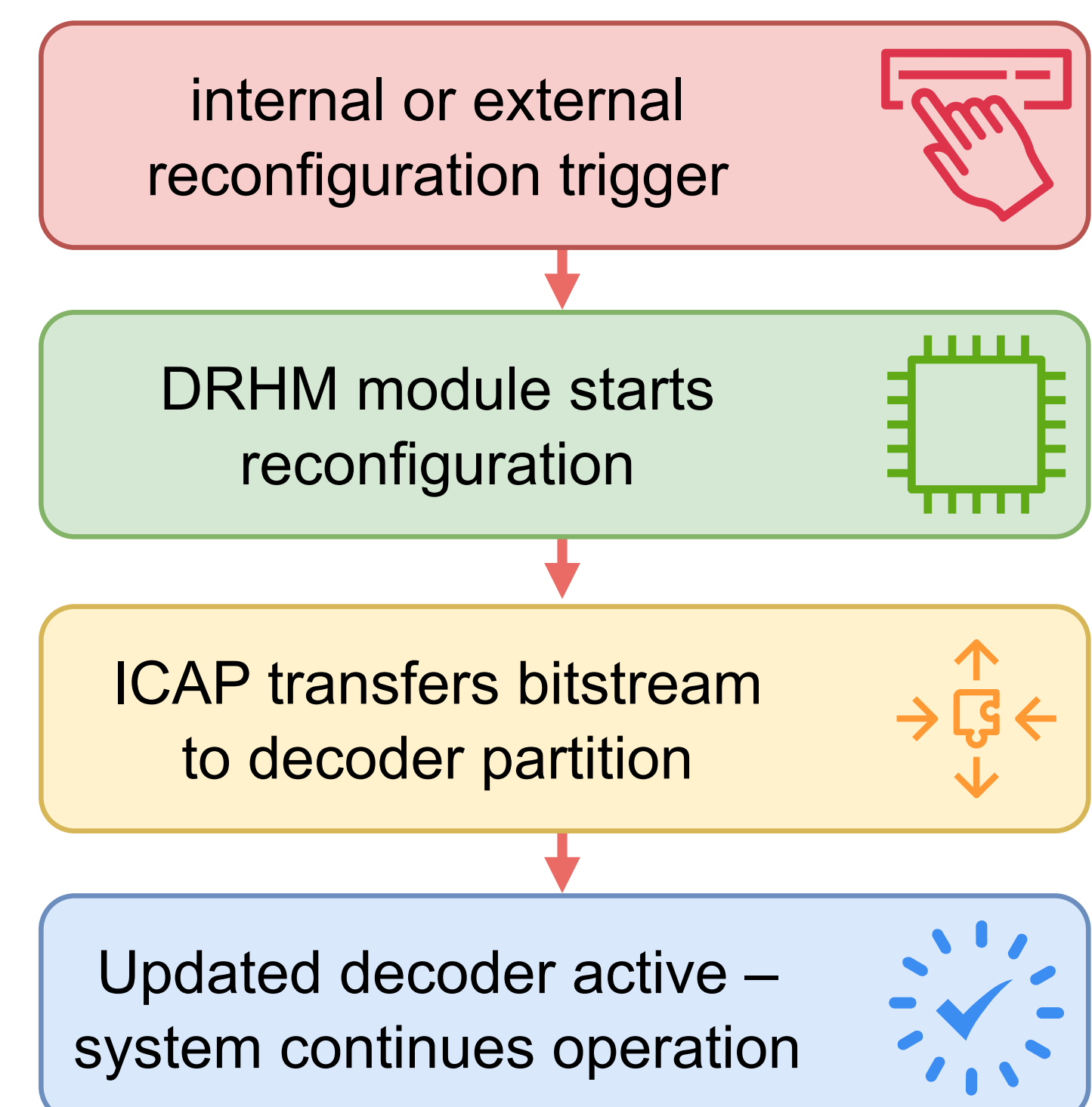
Input	Output	Instr.
trap_entry, instr, icyc2	Control Signals	
0 <...>---01101-- -	01-0-----00100001000-0000010	LUI
0 <...>---11011-- 0	011-----01100010100-00000-0	JAL
0 <...>---11011-- 1	0-11-----00100010100-0000010	JAL

## Evaluation & Results

- Reuse of existing FazyRV test and verification infrastructure.
- Stricter decoder variant has increased hardware resource utilization:
  - +31% LUTs / +39% Flip-flops
- → Resource utilization increase depends on the selected FazyRV datapath width.
- → No measurable impact on the achievable  $f_{max}$  in this setting.

## System Architecture

- Reconfigurable decoder partition integrated into FazyRV.
- Partial bitstreams stored inside external QSPI flash memory.
- Runtime reconfiguration managed by the Dynamic Reconfiguration Hardware Manager (DRHM) [2] module.
- Decoder reconfiguration triggered by CPU or external event.
- Only the decoder partition is exchanged; execution resumes without full FPGA reconfiguration [3].



## Bibliography

- [1] Meinhard Kissich and Marcel Baunach. “FazyRV: Closing the Gap between 32-Bit and Bit-Serial RISC-V Cores with a Scalable Implementation”. In: Proceedings of the 21st ACM Int’l Conference on Computing Frontiers. May 2024, pp. 240–248.
- [2] Florian Angermair. “Dynamic Partial Reconfigurable Hardware Management Module for a RISC-V Based Microcontroller”. Master’s thesis. Graz University of Technology, Institute of Technical Informatics. Graz, Austria. Sep. 2025. Supervisor: Tobias Scheipel.
- [3] Tobias Scheipel, Florian Angermair, and Marcel Baunach. “moreMCU: A Runtime-reconfigurable RISC-V Platform for Sustainable Embedded Systems”. In: Proceedings of the 25th Euromicro Conference on Digital System Design. Maspalomas, Spain, 2022, pp. 24–31.



This poster is licensed under CC BY 4.0 International.  
<https://creativecommons.org/licenses/by/4.0/>  
 Lukas Glantschnig, Tobias Scheipel  
 TU Graz 2026  
<https://www.scheipel.com/>

