Open Paid Project Position:

From Software Patterns to Reconfigurable Hardware Instructions

Motivation & Summary

Modern embedded processors increasingly rely on specialized hardware to improve performance and energy efficiency. However, fixed accelerators are inflexible, while generic FPGA fabrics are often too costly in area, delay, and energy for tightly coupled instruction-level use. In this student assistant position, you will help us bridge this gap: we analyze common software workloads, identify recurring RISC-V instruction patterns, and match them to possible hardware implementations. The goal is to find out which software patterns are promising candidates for runtime-reconfigurable custom instructions in future processor/eFPGA systems.

Recommended Prior Knowledge

- C programming/low-level software understanding
- Basic computer architecture, ideally RISC-V
- Interest in hardware/software co-design
- Verilog/SystemVerilog, FPGA tools, or compiler/toolchain experience

What You Will Learn

- How software workloads behave at instruction level
- How custom instructions can accelerate recurring code patterns
- How software patterns map to hardware structures such as LUT logic, adder trees, permutation networks, MAC blocks, or memory-coupled units

Position Type

- Student project assistant position
- ~10 h/week (€689.8/month), flexible arrangement
- Possible continuation as Master's thesis
- Work embedded in an ongoing research project on runtime-reconfigurable processor architectures

Student Target Groups

- Computer Science (CS)
- Information and Computer Engineering (ICE)
- Electrical Engineering (EE)

Goals & Tasks

- Prepare representative C workloads from, e.g., embedded systems software.
- Compile and analyze workloads for RISC-V and generate static or dynamic instruction traces.
- Identify recurring instruction patterns.
- Characterize candidate patterns by, e.g., operands, dependencies.
- Match software patterns to possible hardware implementations, e.g., LUT-style logic, or memory-coupled units.
- Estimate whether a pattern is worth turning into a runtime-reconfigurable hardware instruction.
- (Optionally) implement and evaluate selected candidates as small SystemVerilog prototypes.

Expected Output

SW Workloads → Instruction Patterns → HW Mapping → Candidates

The expected result is a documented pattern-to-hardware mapping: a small benchmark collection, analysis scripts, a catalogue of recurring instruction patterns, and a ranked list of candidates that appear suitable for runtime-reconfigurable hardware implementation.

Contact & Information

Ass.Prof. Dr. Tobias Scheipel (tobias.scheipel@tugraz.at)

Institute of Technical Informatics
Embedded Architectures & Systems Group
Reconfigurable Computer Architectures