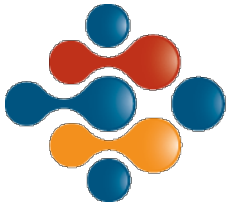


T-CREST

Best Practice for Caching of Single-Path Code

Martin Schoeberl, Bekim Cilku, Daniel Prokesch, and Peter Puschner

Technical University of Denmark

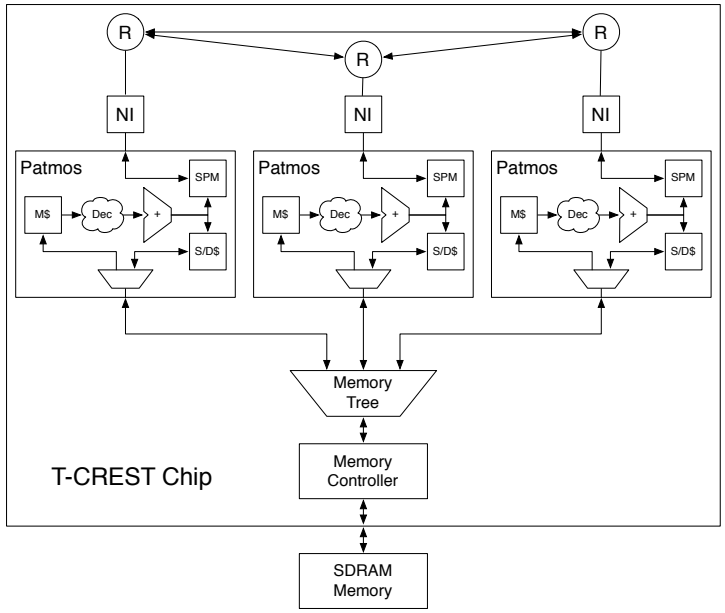
Vienna University of Technology

Context

- Real-time systems
  - Worst-case execution time (WCET) counts
- Different from average-case performance
  - Standard processors are optimized for average-case performance
- Design a processor and a compiler for real-time systems
  - The T-CREST approach

T-CREST

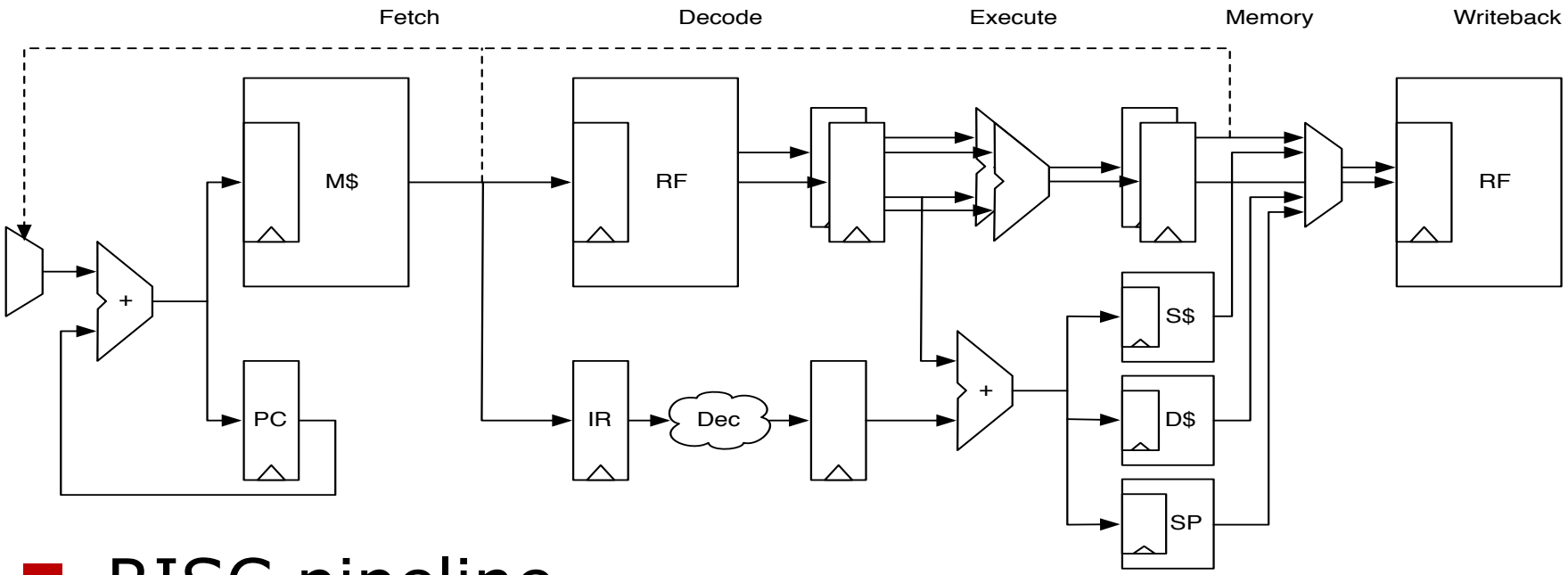
- Time-predictable multicore
  - Processor
  - Network-on-chip
  - Memory hierarchy
  - Compiler
  - WCET analysis (AbsInt aiT and platin)
- Most parts open-source
- https://github.com/t-crest



Patmos Processor

- Time-predictable processor
- Called Patmos
- Flexibility to define the instruction set
  - LLVM compiler adapted for Patmos
- Co-design for low WCET of
  - Patmos
  - Compiler
  - WCET analysis

Patmos Processor



- RISC pipeline
- Dual issue
- Special caches
- No time dependency between instructions

Hardware Description

- Chisel
  - Scala embedded Language
  - Higher level than VHDL/Verilog
- Generates two versions
  - C++ based emulator
  - Verilog based hardware description
- Cycle accurate emulation in C++ faster than VHDL/Verilog simulation
  - Based on the hardware description

Single-path Programming

- Remove input data dependent control flow decisions
  - Gives constant execution time
  - Uses (heavily) predicates
- If-conversion
  - Execute both branches
  - Use if condition for result write back
- Constant loop iterations
  - Use loop bounds
  - Exit condition for result write back

Single-path Programming

- Loops need to be bounded
  - In WCET analyzable programs anyway
- T-CREST compiler can generate single path code from C programs
  - For non-recursive programs
- Simply measure execution time

Single-Path Support in Patmos

- Constant execution time of all instructions
- Predicated instructions
  - 8 predicates
  - One is constant true
  - Write result when predicate is true
  - Otherwise do nothing (NOP instruction)
- All instructions are predicated
  - Execution time independent from predicate

Caches in Patmos

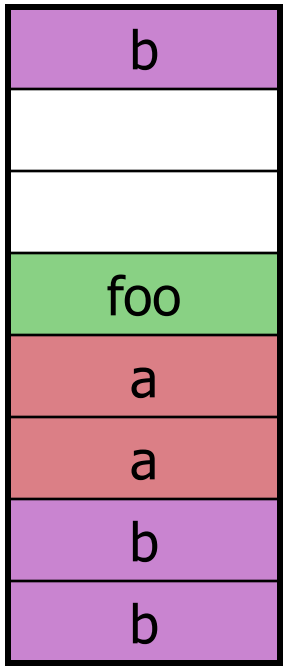
- Configurable: type and size
- For data: normal data cache, stack cache, and scratchpad memory
- For instructions:
  - Standard instruction cache
  - Prefetching instruction cache (SP)
  - Method cache
  - Scratchpad memory
    - Currently only single core (Loader issue)

Method Cache

- Originally developed for the Java processor JOP
  - Therefore called method cache
- Now also used in
  - SHAP
  - Merasa processor (CarCore)
  - Metzlaff PhD thesis
- Also in Patmos

Method Cache

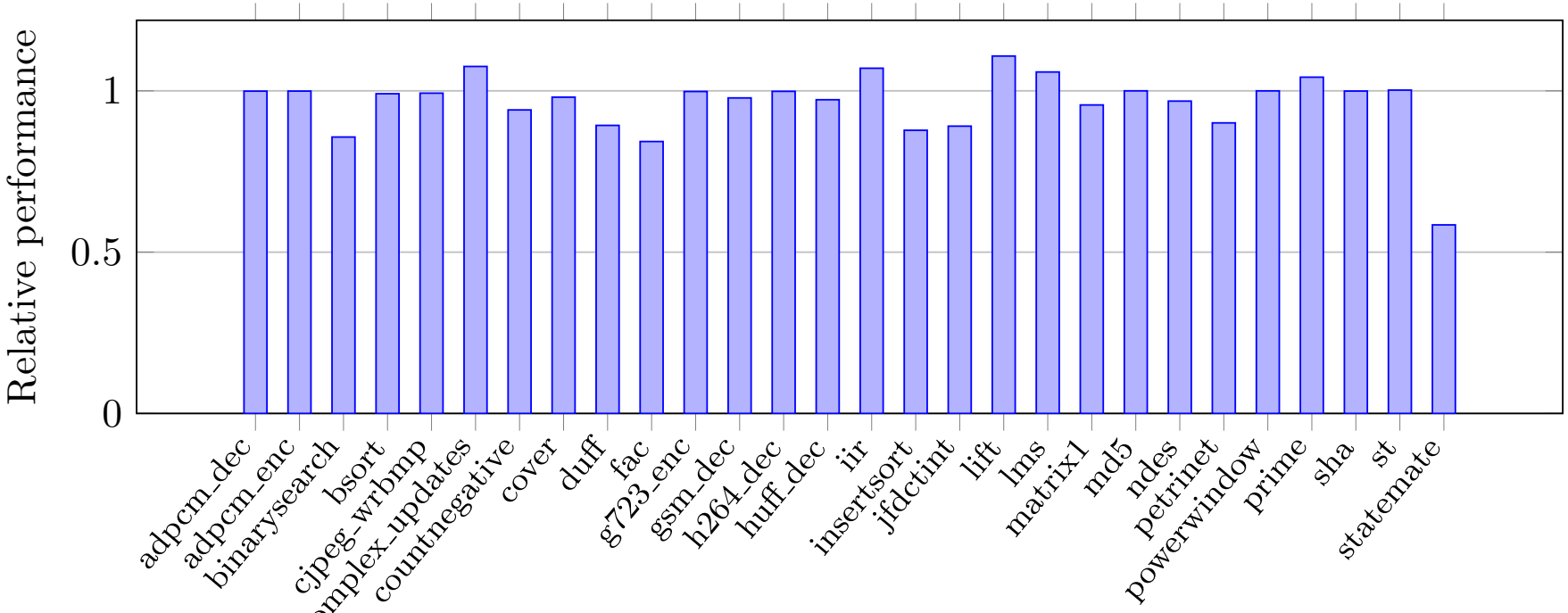
- Caches whole method/functions
  - May load unused instructions
- Misses only on call or return
  - Other instructions guaranteed hits
- Cache is divided in blocks
- Method can span several blocks
- Continuous blocks for a method
- Replacement FIFO
- Tag memory: One entry per block



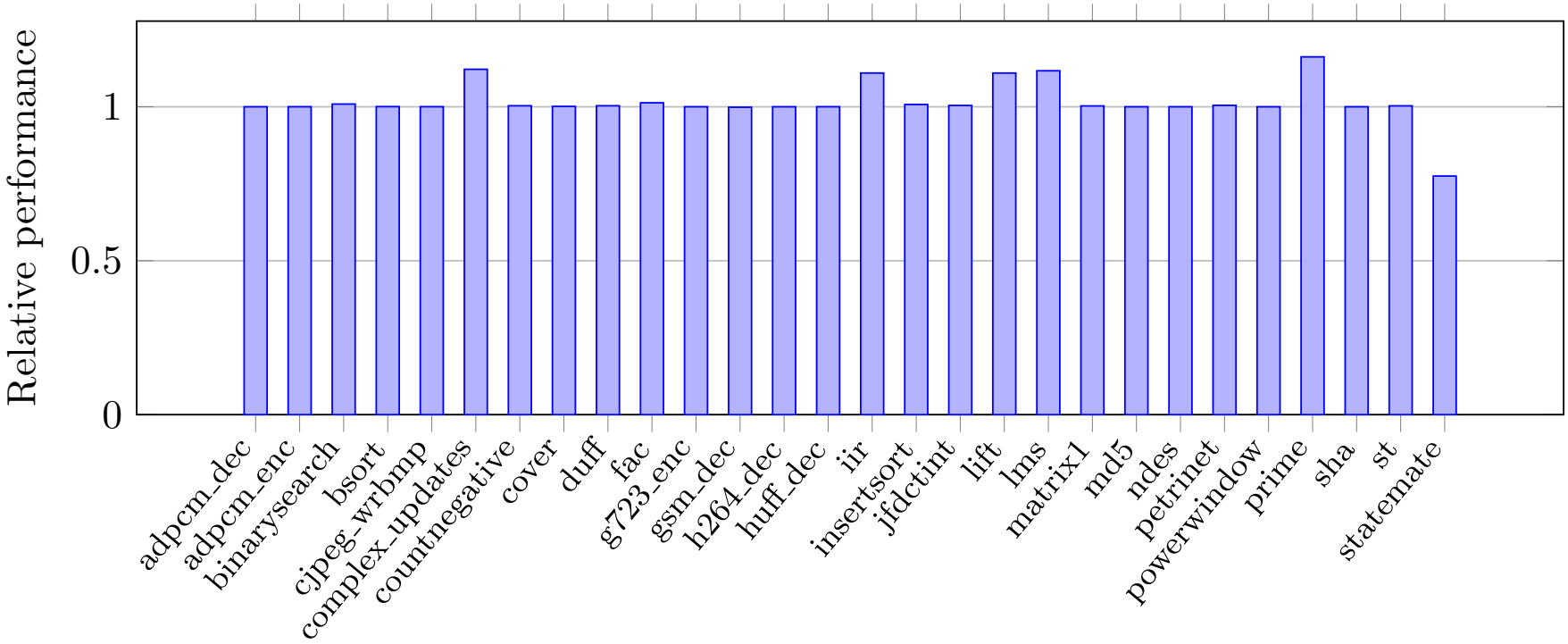
Evaluation

- TACLeBench benchmarks V 1.9
  - Self-contained benchmarks
- Patmos configured for DE2-115 FPGA board
- 8 KB instruction cache
  - 16 methods when method cache
- Cycle accurate emulator to collect the data

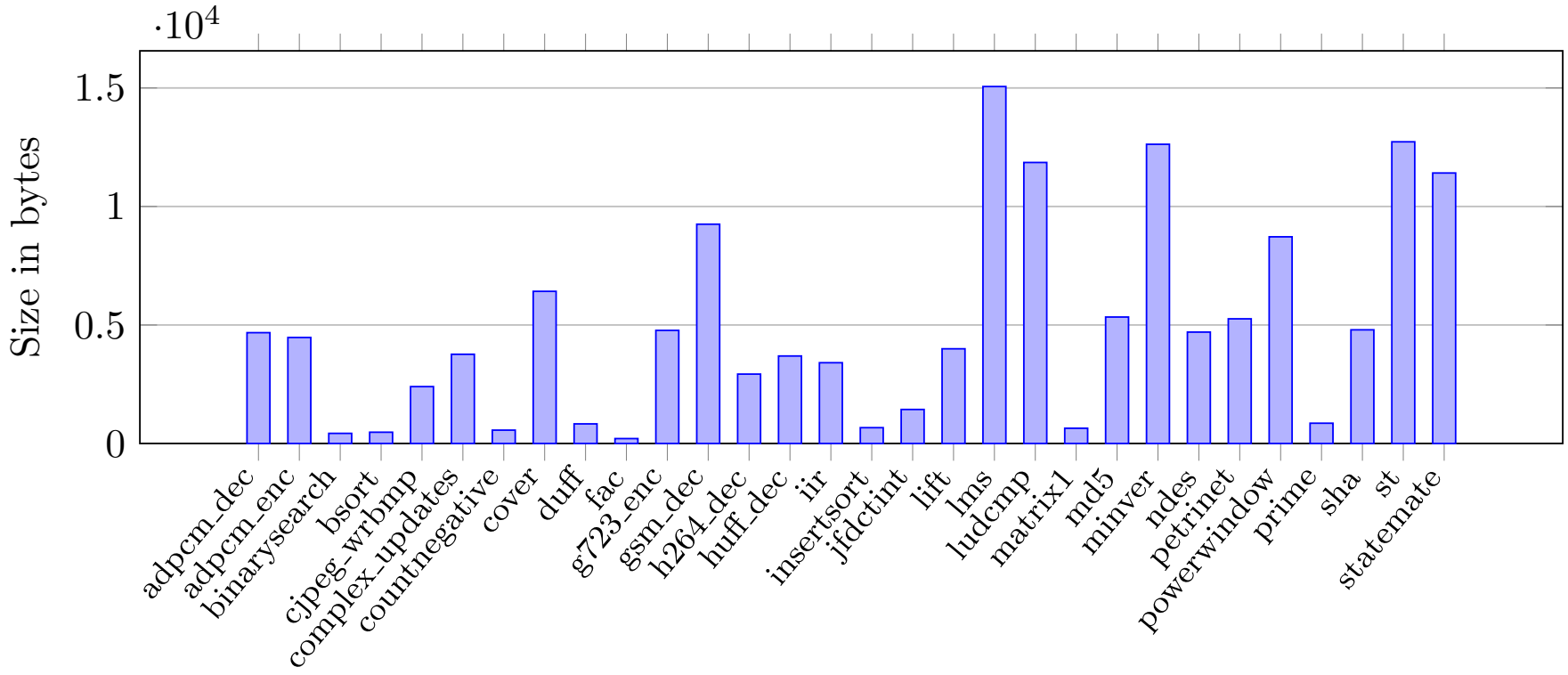
Method vs. Standard Cache



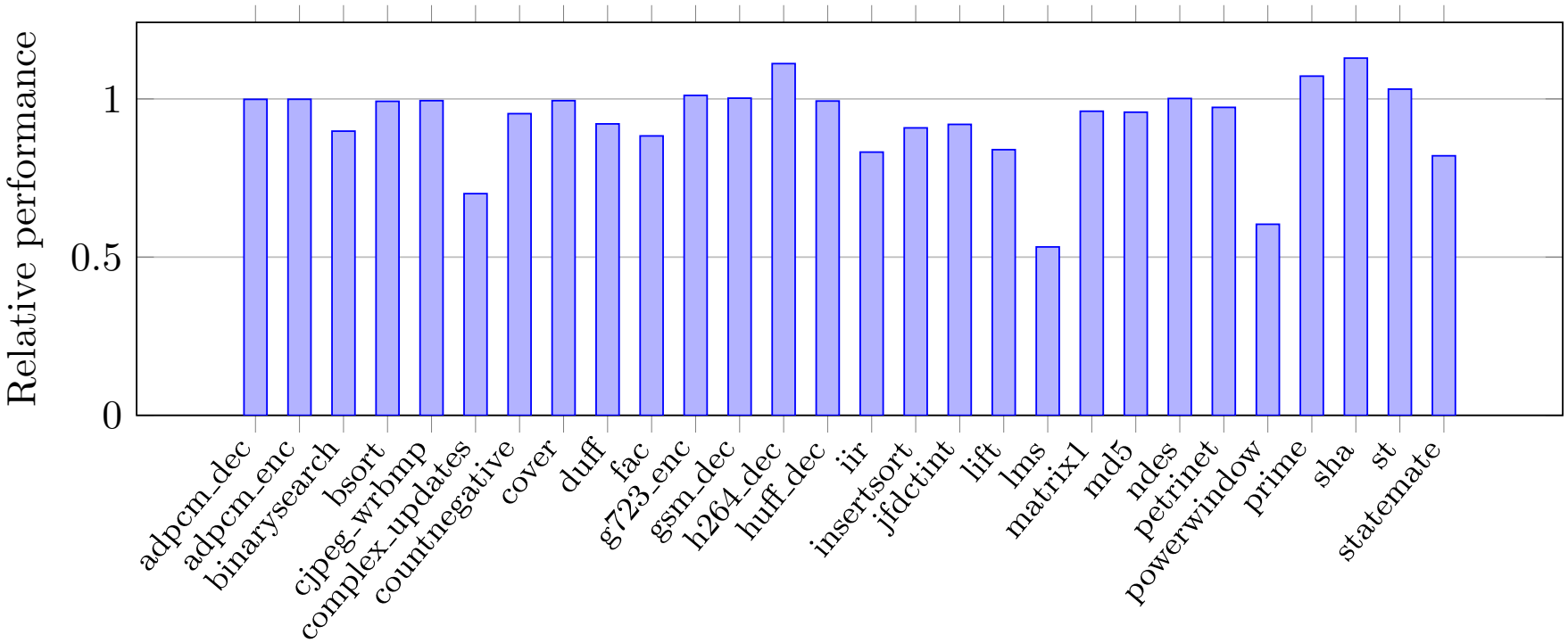
2-way vs Direct Mapped



Dynamic Benchmark Sizes



Method vs Standard Cache 2 KB



Reproducing the Results

README.md

Best Practice for Caching of Single-Path Code

This folder contains information and scripts to run the evaluation experiments for the paper "Best Practice for Caching of Single-Path Code", Martin Schoeberl, Bekim Cilku, Daniel Prokesch, and Peter Puschner, accepted for WCET 2017.

Prerequisites: T-CREST

We use the open-source platform T-CREST for our experiments. Therefore, you need all T-CREST tools installed. A brief installation instruction can be found at the Patmos repository.

We also provide a VM with Ubuntu where all needed packages are installed. However, that VM is used in teaching and does not contain the latest version of T-CREST. Therefore, you need to reinstall T-CREST there with:

```
rm -rf t-crest
mkdir ~/t-crest
cd ~/t-crest
git clone https://github.com/t-crest/patmos-misc.git misc
./misc/build.sh
```

The Benchmarks

Conclusion

- Single-path code gives constant execution time
- Compared different caching organizations
- No single winner
- In FPGA we can use application specific caching