Defying the Memory Bottleneck in Hardware Accelerated Collision Detection

Andreas Raabe Frank Zavelberg

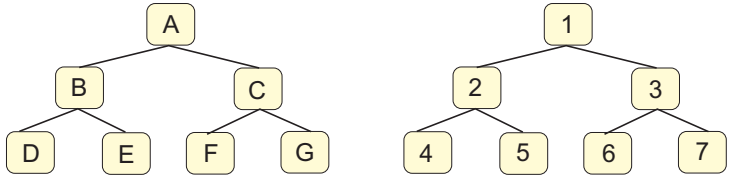
Presentation:
Andreas H. Niers

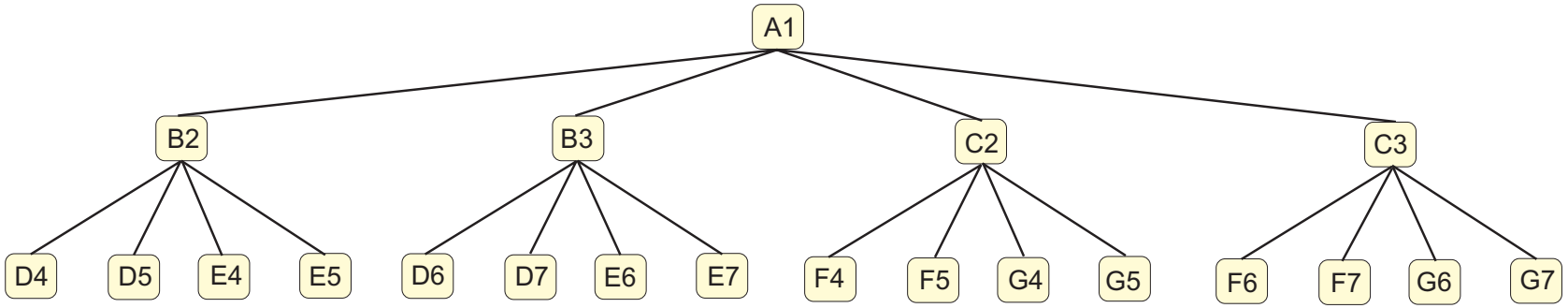
University of Bonn
Technical Computer Science
Römerstr. 164, 53117 Bonn, Germany
{raabe,zavelber,niers}@cs.uni-bonn.de

06.02.2008

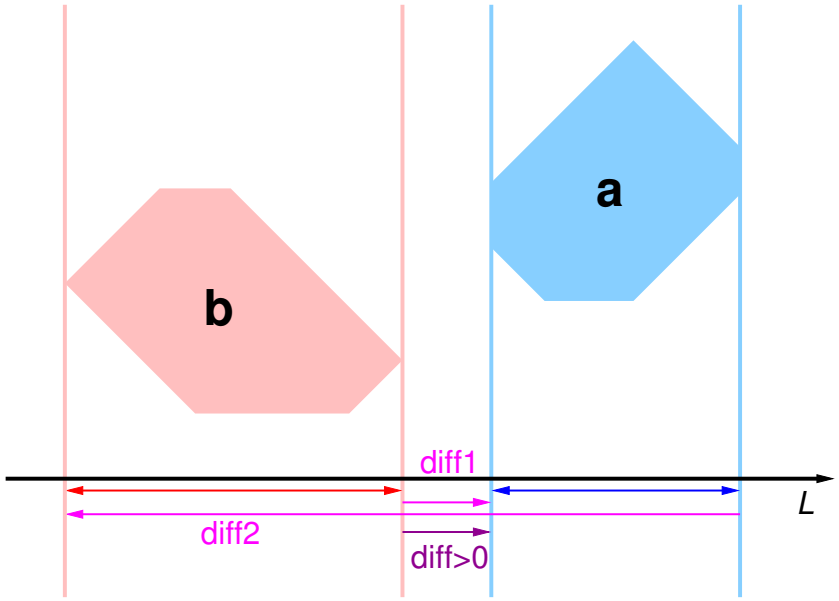
Hierarchical Collision Detection

- broad phase
  - 24-DOPs as BVs
- narrow phase
  - triangles as primitives





Separating Axis Test (SAT)

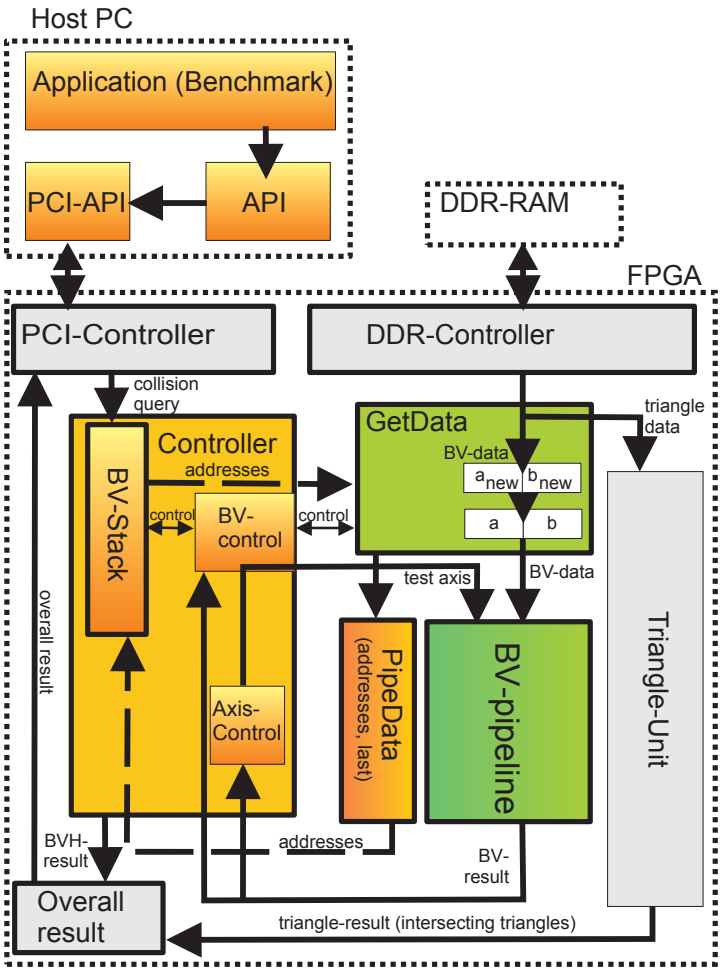


- SAT for BVs and primitives
- projection onto axis
- if $diff > 0$ axis is a separating axis $\rightarrow$ separation
- if $diff <= 0$ axis is no separating axis $\rightarrow$ possible collision
- > 1000 axes need to be tested for 24-DOPs
- only a subset is tested for k-DOPs
  $\rightarrow$ false reported collision for k-DOPs allowed (false positive)

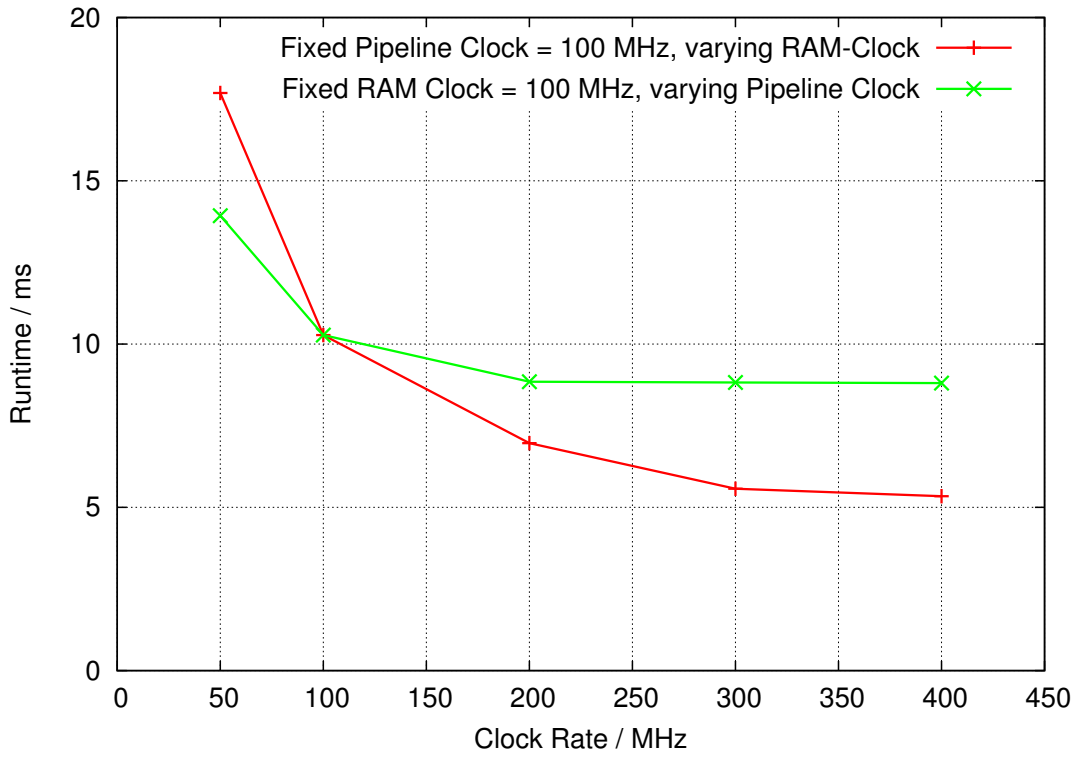
The CollisionChip-Design

Design Overview

- send tests via API
- communication over PCI
- controller schedules the tests
- GetData routes data to according test pipeline
- PipeData contains bookkeeping information
- overall result combines results
- 4 times faster than software implementation



Identifying the Bottleneck

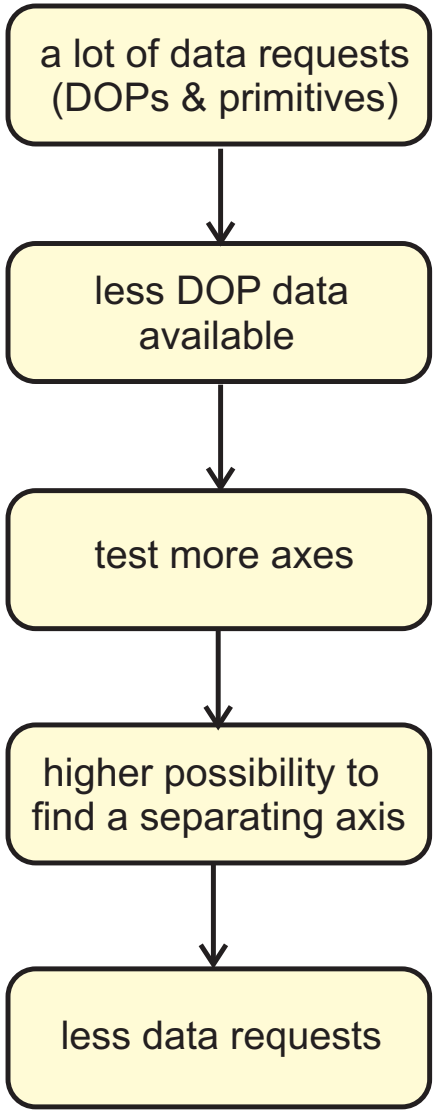


- speeding up the pipeline or using a second pipeline does not speed up overall calculation
- increasing memory bandwidth has much larger effect
- $\longrightarrow$ caching

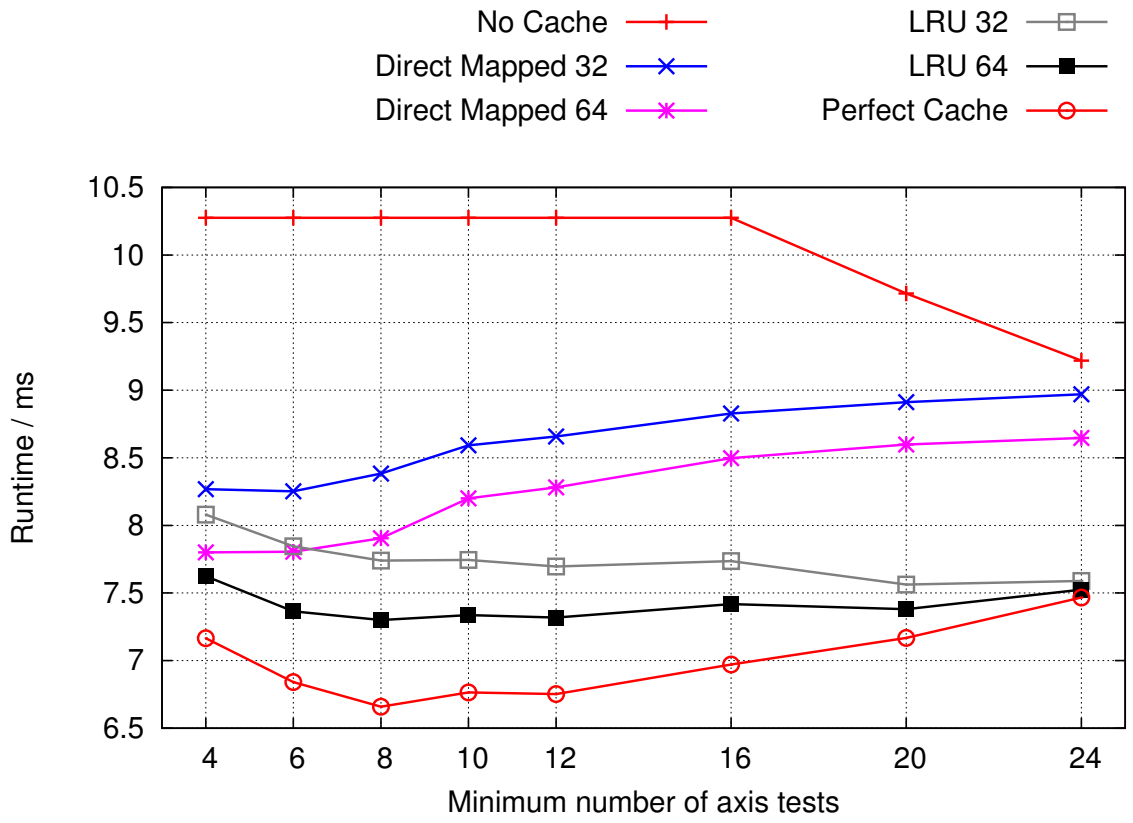
Push Architecture

push architecture

- start new calculation when new pair is ready to be tested
- minimum number of axis tests needs to be defined
  $\longrightarrow$ avoid a fast descent
- avoids memory and pipeline stalls
- feedback
  $\longrightarrow$ relaxation of the bottleneck



Comparing Caching Techniques



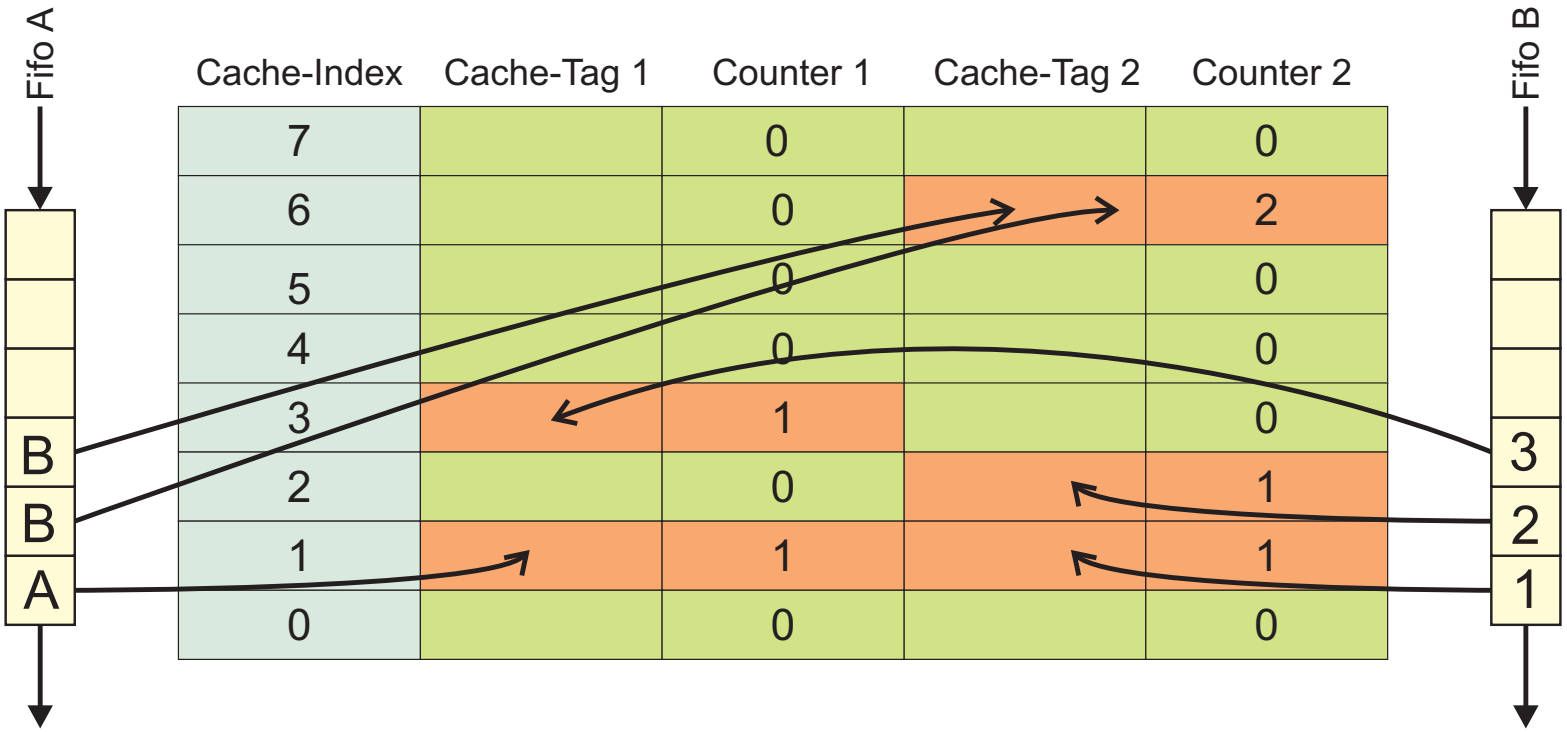
Direct Mapped Cache

- hardware efficient
- performs far below the optimum

LRU Cache

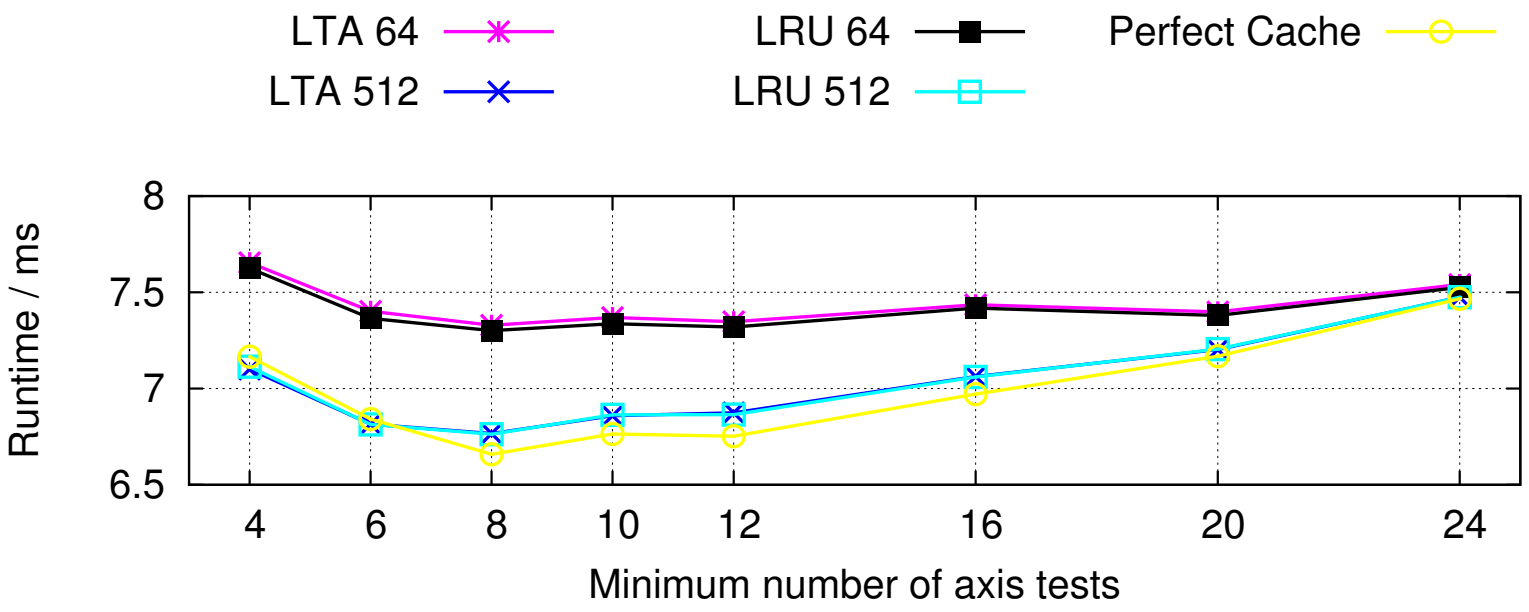
- good performance
- much hardware costs

Lockable Two-Way Set-Associative Cache (LTA)



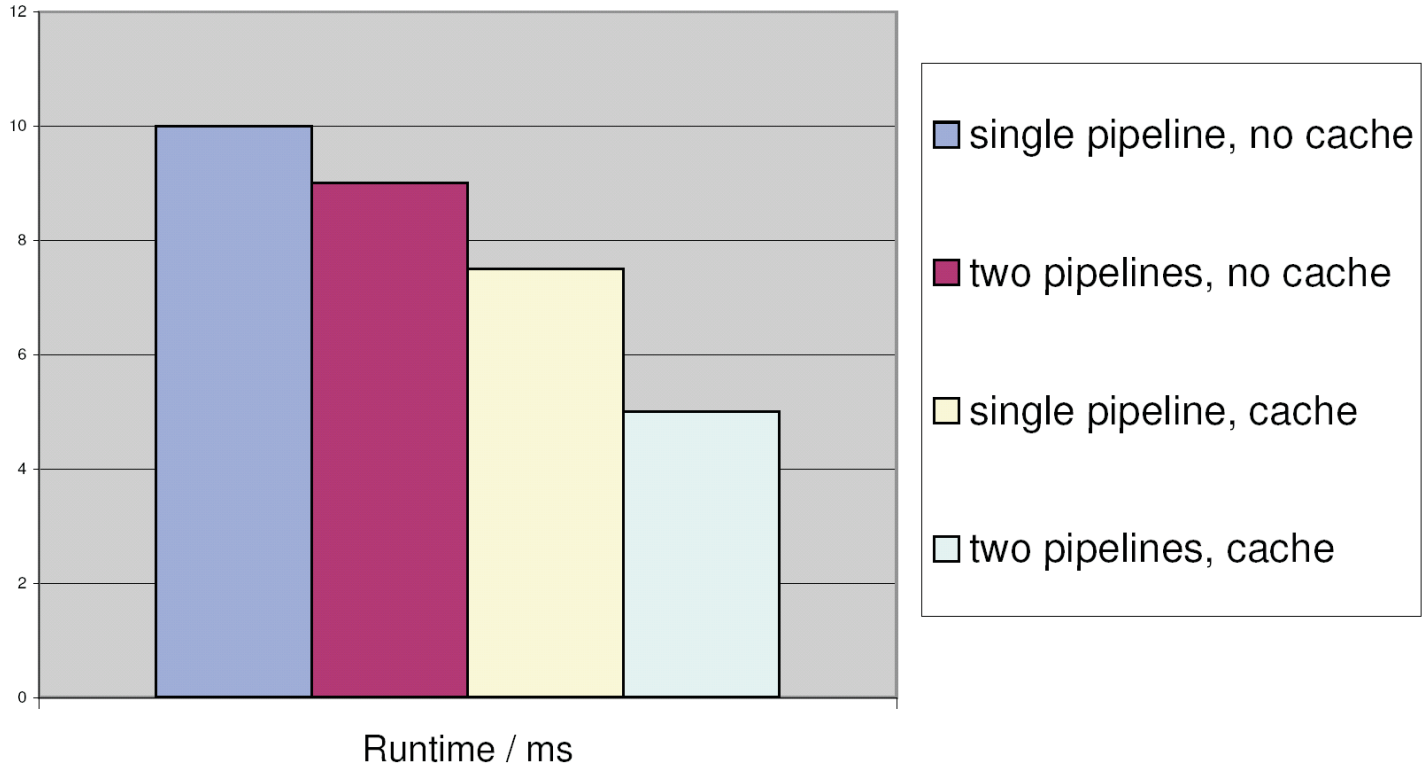
- novel LTA cache developed
- addresses of test pairs are stored in FIFOs
- 2-way set associativity avoids replacement
- Entry Locking
- stall memory if locked entry needs to be replaced (0.0008 %)

LTA Cache Performance



- LRU 512 and LTA 512 perform close to the optimum
- LRU is prohibitively expensive
- LTA 512 has very low hardware consumption
  - on Xilinx Virtex II FPGA:
  - 7 % of the slices
  - 5 % of slice flip flops
  - 22 % of the Block RAMs

CollisionChip Performance With Caching



- speedup: 35 % using a single pipeline
- using a cache enabels parallel implementation of several pipelines
- with cache and second pipeline:
  $\longrightarrow$ 10 times faster than software implementation

Conclusion

conclusion

- novel approach for hardware-accelerated high-speed collison detection
- push architecture
- novel LTA cache
  - performes close to the theoretical optimum
  - low hardware consumption
  - implemented and tested on Xilinx Virtex II FPGA
- using a cache enables using multiple pipelines
- speedup of 10 could be achieved

Conclusion

conclusion

- novel approach for hardware-accelerated high-speed collison detection
- push architecture
- novel LTA cache
  - performes close to the theoretical optimum
  - low hardware consumption
  - implemented and tested on Xilinx Virtex II FPGA
- using a cache enables using multiple pipelines
- speedup of 10 could be achieved

Thank you for your attention!