National Technical University of Athens
School of Electrical and Computer Engineering
Computing Systems Laboratory

A Configurable TLB Hierarchy for the RISC-V Architecture

Nikolaos Charalampos Papadopoulos, Vasileios Karakostas,
Konstantinos Nikas, Nectarios Koziris, Dionisios N. Pnevmatikatos

ncpapad@cslab.ece.ntua.gr

FPL 2020

Motivation

Configurable high-performance soft-processors are getting more attractive

- FPGA fabrics get cheaper and larger
- Expanding FPGA applications for soft processors

RISC-V and Rocket Chip Generator

- Extensible & Configurable + custom accelerators
- Tailored design to the needs of the application

Outline

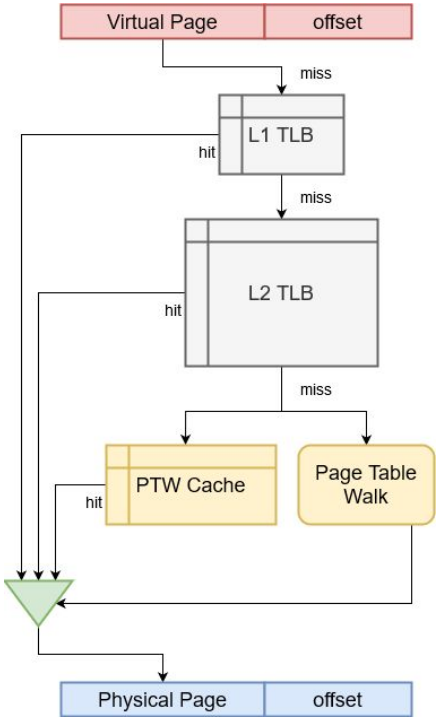
- Background
- Configurable TLB Hierarchy features
- Methodology
- Performance and Resource Results
- Related & Future work
- Conclusions

Rocket Chip Generator

- SoC Generator that produces Synthesizable RTL
  - Written in Chisel
  - Rocket core or BOOM (Berkeley Out-of-Order Machine)
  - Parameterized Tiles, Caches, Accelerators, etc.
- Library of processor parts and utilities
  - Branch predictors
  - Replacement policies
  - ...and many more

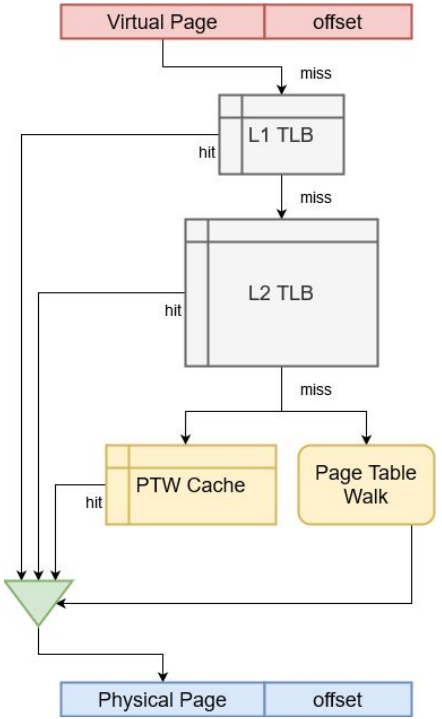
Existing MMU in Rocket Chip Generator

- Fully-associative L1 TLB
  - Separate Data/Instr L1 TLB
  - Vector of Registers
  - Fast & small (32-128 entries)
- Direct-mapped L2 TLB
  - SyncReadMem
  - Slower but larger (128-1024 entr.)
- Fully-associative PTW Cache
  - Vector of Registers
  - Keeps non-leaf nodes



Configurable TLB hierarchy in Rocket

- Kept the same overall structure
  - Lookups, refill, replacement policies, flushing
- Added about 70 LoC for the L1 TLB
- 50 LoC for the L2 TLB
- Implementation in two different editions of the RCG
  - April 2018 version
    - Supports Xilinx ZCU102
  - January 2020 version



L1 | L2 TLB Contributions

	Vanilla L1 | L2 TLB	Configurable L1 | L2 TLB
Organization	Fully-assoc | Direct-mapped	Any associativity
Parameterization	#Entries	#Sets, #Ways (pow2)
Replacement policies	PseudoLRU/Random | No policy	Pseudo LRU/Random set-associative alternatives
Other features	Sectored L1 TLB entries	Sectored L1 TLB entries are supported too

HW & SW Development Flow

- **Hardware Flow**
  - Chisel & FIRRTL checks
  - Verilator: Cycle-accurate Simulator
  - Xilinx ZCU102 bitstream generation
- **Software flow**
  - Freedom-U-SDK
  - Minimal Buildroot distro
  - SPEC2006 benchmarks

Evaluation Metrics

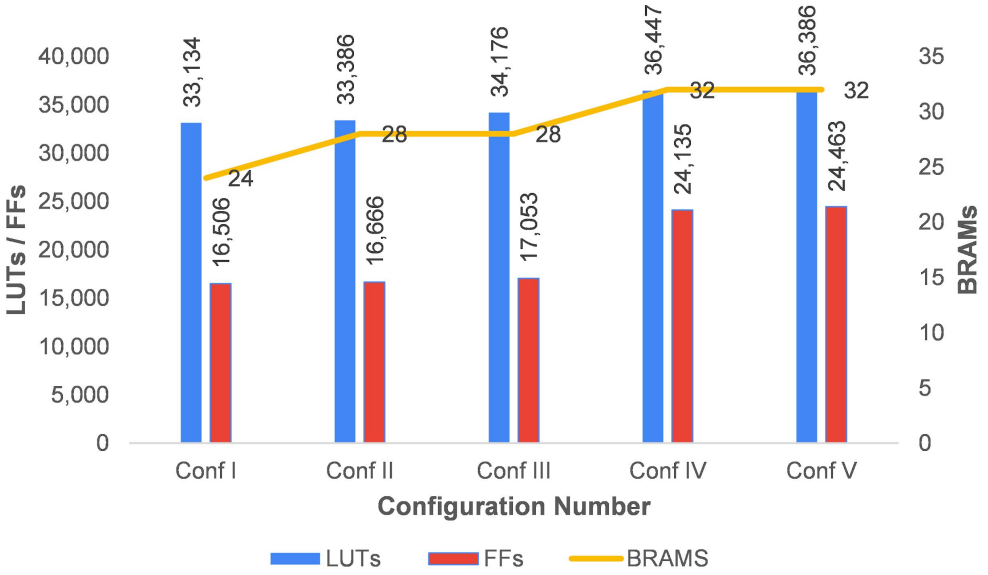
- **FPGA Resource Usage**
  - Lookup-Tables (LUTs), Flip-Flops (FFs), Block RAM (BRAMs)
- **Performance Metrics**
  - SPEC2006 benchmarks (with test input set)
    - Misses-per-kilo-Instructions (MPKI)
    - Instructions-per-cycle (IPC)

Evaluation Scenarios

Conf. No	DTLB	ITLB	L2 TLB
I	32-way, 32 entries	32-way, 32 entries	-
II	32-way, 32 entries	32-way, 32 entries	4-way, 128 entries
III	32-way, 32 entries	32-way, 32 entries	4-way, 512 entries
IV	8-way, 64 entries	8-way, 128 entries	8-way, 1024 entries
V	8-way, 128 entries	8-way, 64 entries	8-way, 1024 entries

- Configurations resembling well-known architectures
  - Conf III → ARM Cortex A57
  - Conf IV → Intel Skylake
  - Conf V → Intel Skylake (swapped I/D TLB sizes)

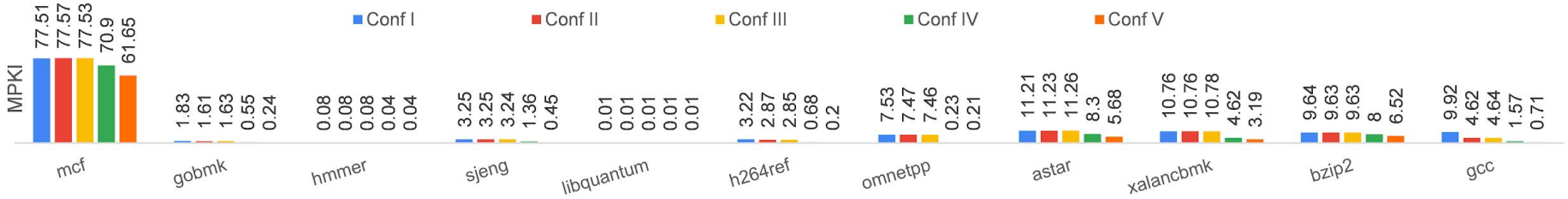
FPGA resource usage evaluation



Configuration	I	II	III	IV	V
Frequency (Mhz)	189	187	186	188	186

Conf. No	DTLB	ITLB	L2 TLB
I	32-way, 32 entries	32-way, 32 entries	-
II	32-way, 32 entries	32-way, 32 entries	4-way, 128 entries
III	32-way, 32 entries	32-way, 32 entries	4-way, 512 entries
IV	8-way, 64 entries	8-way, 128 entries	8-way, 1024 entries
V	8-way, 128 entries	8-way, 64 entries	8-way, 1024 entries

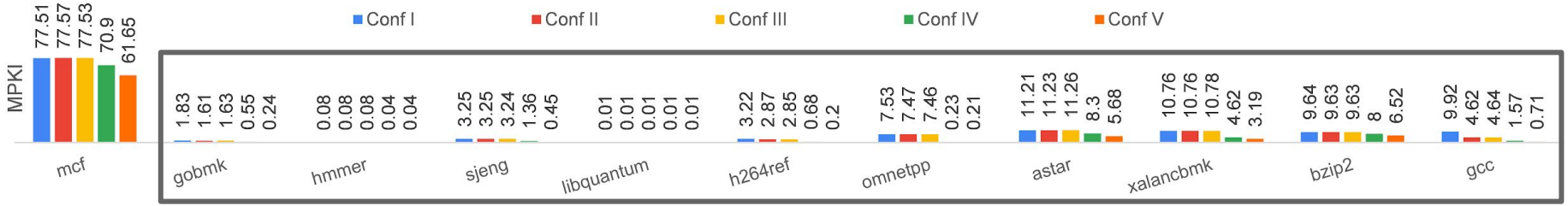
L1 TLB Performance Evaluation (MPKI)



- Most L1 TLB misses come from data accesses
- Several benchmarks show similar behavior across configurations
- But larger L1 DTLB may improve performance
- mcf stresses the TLB hierarchy the most

Conf. No	DTLB	ITLB	L2 TLB
I	32-way, 32 entries	32-way, 32 entries	-
II	32-way, 32 entries	32-way, 32 entries	4-way, 128 entries
III	32-way, 32 entries	32-way, 32 entries	4-way, 512 entries
IV	8-way, 64 entries	8-way, 128 entries	8-way, 1024 entries
V	8-way, 128 entries	8-way, 64 entries	8-way, 1024 entries

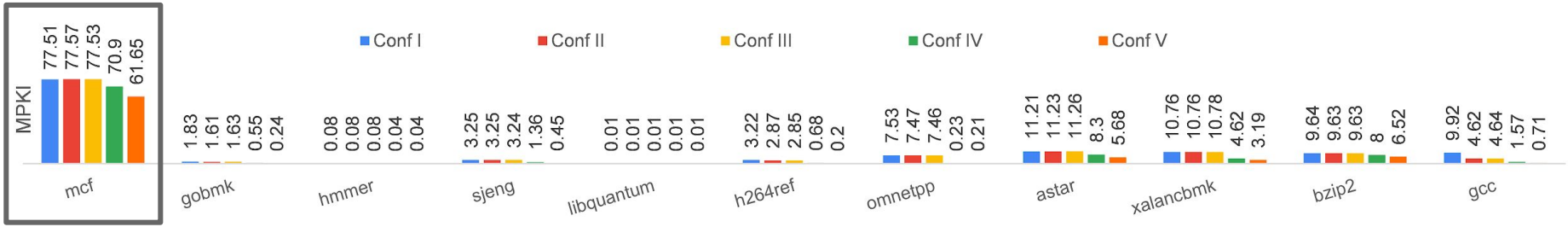
L1 TLB Performance Evaluation (MPKI)



- Most L1 TLB misses come from data accesses
- Several benchmarks show similar behavior across configurations
- But larger L1 DTLB may improve performance
- mcf stresses the TLB hierarchy the most

Conf. No	DTLB	ITLB	L2 TLB
I	32-way, 32 entries	32-way, 32 entries	-
II	32-way, 32 entries	32-way, 32 entries	4-way, 128 entries
III	32-way, 32 entries	32-way, 32 entries	4-way, 512 entries
IV	8-way, 64 entries	8-way, 128 entries	8-way, 1024 entries
V	8-way, 128 entries	8-way, 64 entries	8-way, 1024 entries

L1 TLB Performance Evaluation (MPKI)



- Results for L1 Data and Instruction TLBs
- Most L1 TLB misses come from data accesses
- Several benchmarks show similar behavior across configurations
- But larger L1 DTLB may improve performance
- mcf stresses the TLB hierarchy the most

Conf. No	DTLB	ITLB	L2 TLB
I	32-way, 32 entries	32-way, 32 entries	-
II	32-way, 32 entries	32-way, 32 entries	4-way, 128 entries
III	32-way, 32 entries	32-way, 32 entries	4-way, 512 entries
IV	8-way, 64 entries	8-way, 128 entries	8-way, 1024 entries
V	8-way, 128 entries	8-way, 64 entries	8-way, 1024 entries

L2 TLB Performance Evaluation (MPKI)

- L2 TLB misses are rare for most benchmarks
- Larger L2 TLB reach may reduce page walks
  - Configurations IV and V
- mcf improves significantly as L2 TLB increases

Conf. No	DTLB	ITLB	L2 TLB
I	32-way, 32 entries	32-way, 32 entries	-
II	32-way, 32 entries	32-way, 32 entries	4-way, 128 entries
III	32-way, 32 entries	32-way, 32 entries	4-way, 512 entries
IV	8-way, 64 entries	8-way, 128 entries	8-way, 1024 entries
V	8-way, 128 entries	8-way, 64 entries	8-way, 1024 entries

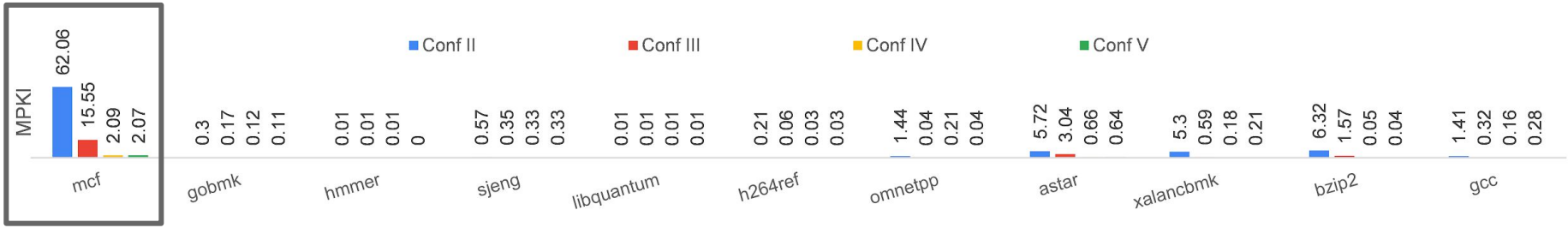
National Technical University of Athens CSLab

L2 TLB Performance Evaluation (MPKI)

- L2 TLB misses are rare for most benchmarks
- Larger L2 TLB reach may reduce page walks
  - Configurations IV and V
- mcf improves significantly as L2 TLB increases

Conf. No	DTLB	ITLB	L2 TLB
I	32-way, 32 entries	32-way, 32 entries	-
II	32-way, 32 entries	32-way, 32 entries	4-way, 128 entries
III	32-way, 32 entries	32-way, 32 entries	4-way, 512 entries
IV	8-way, 64 entries	8-way, 128 entries	8-way, 1024 entries
V	8-way, 128 entries	8-way, 64 entries	8-way, 1024 entries

L2 TLB Performance Evaluation (MPKI)



- L2 TLB misses are rare for most benchmarks
- Larger L2 TLB reach may reduce page walks
  - Configurations IV and V
- mcf improves significantly as L2 TLB increases

Conf. No	DTLB	ITLB	L2 TLB
I	32-way, 32 entries	32-way, 32 entries	-
II	32-way, 32 entries	32-way, 32 entries	4-way, 128 entries
III	32-way, 32 entries	32-way, 32 entries	4-way, 512 entries
IV	8-way, 64 entries	8-way, 128 entries	8-way, 1024 entries
V	8-way, 128 entries	8-way, 64 entries	8-way, 1024 entries

System Performance Evaluation (IPC)

Benchmark	I	II	III	IV	V
mcf	0.13	-	7.7 %	15.4 %	15.4 %
gobmk	0.44	-	-	2.3 %	2.3 %
hmmer	0.58	-	-	-	-
sjeng	0.55	1.8 %	1.8 %	1.8 %	3.6 %
libquantum	0.44	-	-	-	-
h264ref	0.77	1.4 %	1.4 %	2.6 %	2.6 %
omnetpp	0.35	2.9 %	5.7 %	5.7 %	5.7 %
astar	0.36	-	-	2.8 %	2.8 %
xalancbmk	0.36	2.8 %	8.3 %	8.3 %	8.3 %
bzip2	0.51	2.0 %	4.0 %	5.9 %	5.9 %
gcc	0.44	2.2 %	2.2 %	4.5 %	4.5 %

System Performance Evaluation (IPC)

Benchmark	I	II	III	IV	V
mcf	0.13	-	7.7 %	15.4 %	15.4 %
gobmk	0.44	-	-	2.3 %	2.3 %
hmmer	0.58	-	-	-	-
sjeng	0.55	1.8 %	1.8 %	1.8 %	3.6 %
libquantum	0.44	-	-	-	-
h264ref	0.77	1.4 %	1.4 %	2.6 %	2.6 %
omnetpp	0.35	2.9 %	5.7 %	5.7 %	5.7 %
astar	0.36	-	-	2.8 %	2.8 %
xalancbmk	0.36	2.8 %	8.3 %	8.3 %	8.3 %
bzip2	0.51	2.0 %	4.0 %	5.9 %	5.9 %
gcc	0.44	2.2 %	2.2 %	4.5 %	4.5 %

System Performance Evaluation (IPC)

Benchmark	I	II	III	IV	V
mcf	0.13	-	7.7 %	15.4 %	15.4 %
gobmk	0.44	-	-	2.3 %	2.3 %
hmmer	0.58	-	-	-	-
sjeng	0.55	1.8 %	1.8 %	1.8 %	3.6 %
libquantum	0.44	-	-	-	-
h264ref	0.77	1.4 %	1.4 %	2.6 %	2.6 %
omnetpp	0.35	2.9 %	5.7 %	5.7 %	5.7 %
astar	0.36	-	-	2.8 %	2.8 %
xalancbmk	0.36	2.8 %	8.3 %	8.3 %	8.3 %
bzip2	0.51	2.0 %	4.0 %	5.9 %	5.9 %
gcc	0.44	2.2 %	2.2 %	4.5 %	4.5 %

Related & Future Work

- Improving soft-processor performance
  - Prior work targets hand optimized HDL code
  - Improvements in Chisel compiler → Cheaper & better FPGA mappings

- Reduce resource usage in FPGA simulation
  - Fully-assoc. TLBs are CAMs → FPGA-hostile structure

Conclusions

- Enabled further configurability in the Rocket Chip Generator
- Our design can output any L1/L2 TLB organization/size
- Evaluated resource usage & application performance

https://github.com/ncppd/rocket-chip

Thank you!