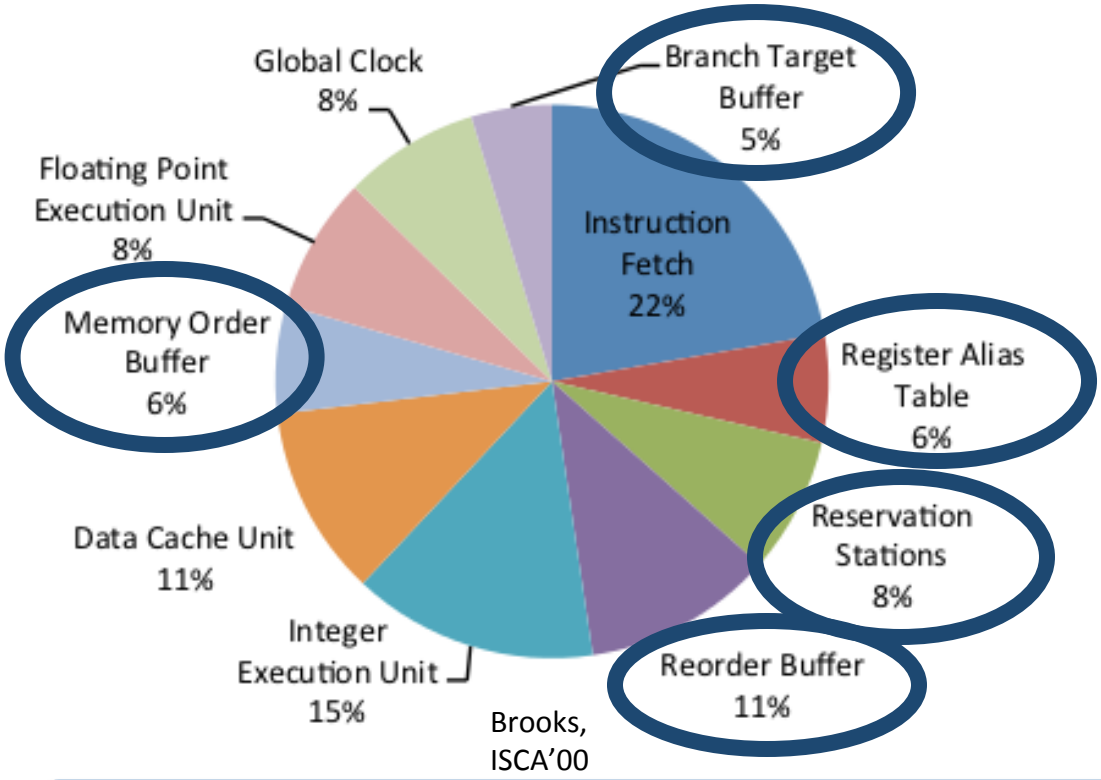
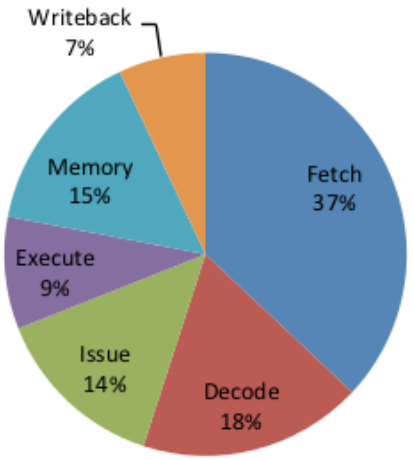
Core Energy Comparison

Out-of-Order



Brooks, ISCA'00

In-Order



Dally, IEEE Computer'08

Do we always need the extra hardware?

MorphCore: An Energy-Efficient Architecture for High-Performance ILP and High-Throughput TLP

Khubaib[*]

M. Aater Suleman[*+]

Milad Hashemi[*]

Chris Wilkerson[‡]

Yale N. Patt[*]

[*] HPS Research Group
The University of Texas at Austin

[+] Calxeda Inc.

[‡] Intel Labs

Courtesy: Prof. Yale Patt, UT Austin

The Need for an Adaptive Core

- Sometimes a single thread with high ILP
  - Need a heavy-weight out-of-order core
  - Provides high performance by exploiting ILP
- Sometimes many threads
  - Out-of-order is unnecessary
  - Need a power-efficient core
  - Provides high performance by exploiting thread-level parallelism
- We need an adaptive core that can do both
  - Exploits instruction-level parallelism when needed
  - Exploits thread-level parallelism when needed

Problem

Current core architectures do not adapt

Large cores limit performance when TLP is high

Small cores limit performance when TLP is low

Outline

- Problem Statement
- Previous Work
  - Asymmetric chip multiprocessors
  - Reconfigurable core architectures
- MorphCore
- Evaluation

Asymmetric Chip Multiprocessors

- One or few large out-of-order cores with many small in-order cores
  [Morad+ CAL'06, Suleman+ TR'07, Hill+ Computer'07, Suleman+ ASPLOS'09]
  - Limited flexibility
    - Fixed number of large and small cores
  - Migration overhead
    - Migrate the thread state/data to large core

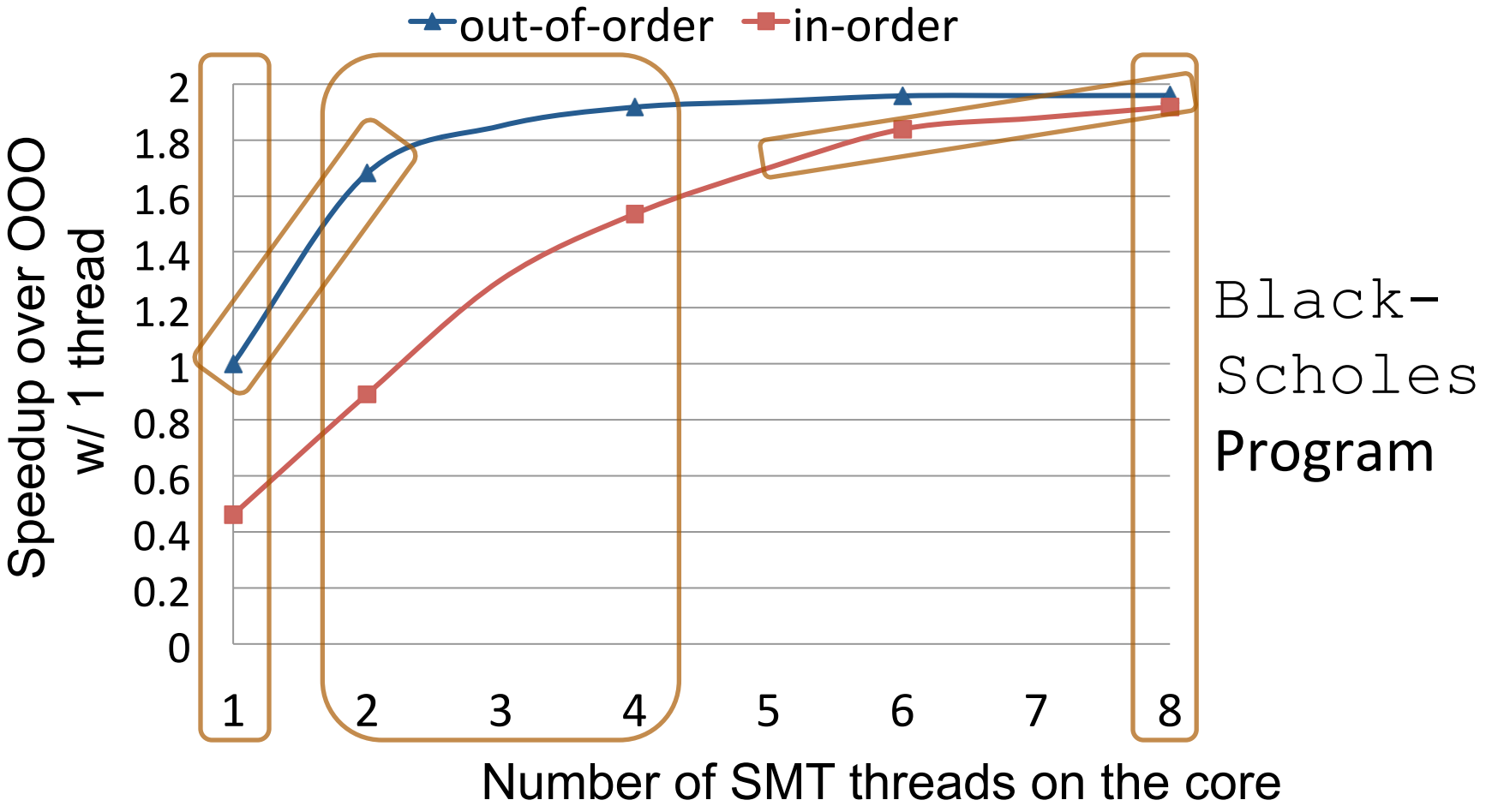
Reconfigurable Core Architectures

- Fundamental Idea
  - Build a chip with "simpler cores" and "combine" them at runtime using additional logic to form a high-performance out-of-order core
  - Core Fusion - Ipek+ ISCA'07, TFlex - Kim+ MICRO'07, Federation Cores - Tarjan+ DAC'08, and many others
- Fused core has low performance and low energy-efficiency
  - Increased latencies among its pipeline stages
- Significant mode switching overhead

Outline

- Problem Statement
- Previous Work
- MorphCore
  - Key Insights and Basic Idea
  - Design and Operation
- Evaluation

Key Insight 1: The Potential of In-Order SMT



- With 8 threads, the in-order core's performance almost matches the out-of-order core's

Key Insight 2

Minimal changes to a traditional OOO core
can transform it into a
highly-threaded in-order SMT core

Existing structures in an OOO core
can be re-used to support
highly-threaded in-order SMT execution

MorphCore: Basic Idea

The opposite of previous proposals:

A) The base design: OOO core
B) Then we add in-order SMT

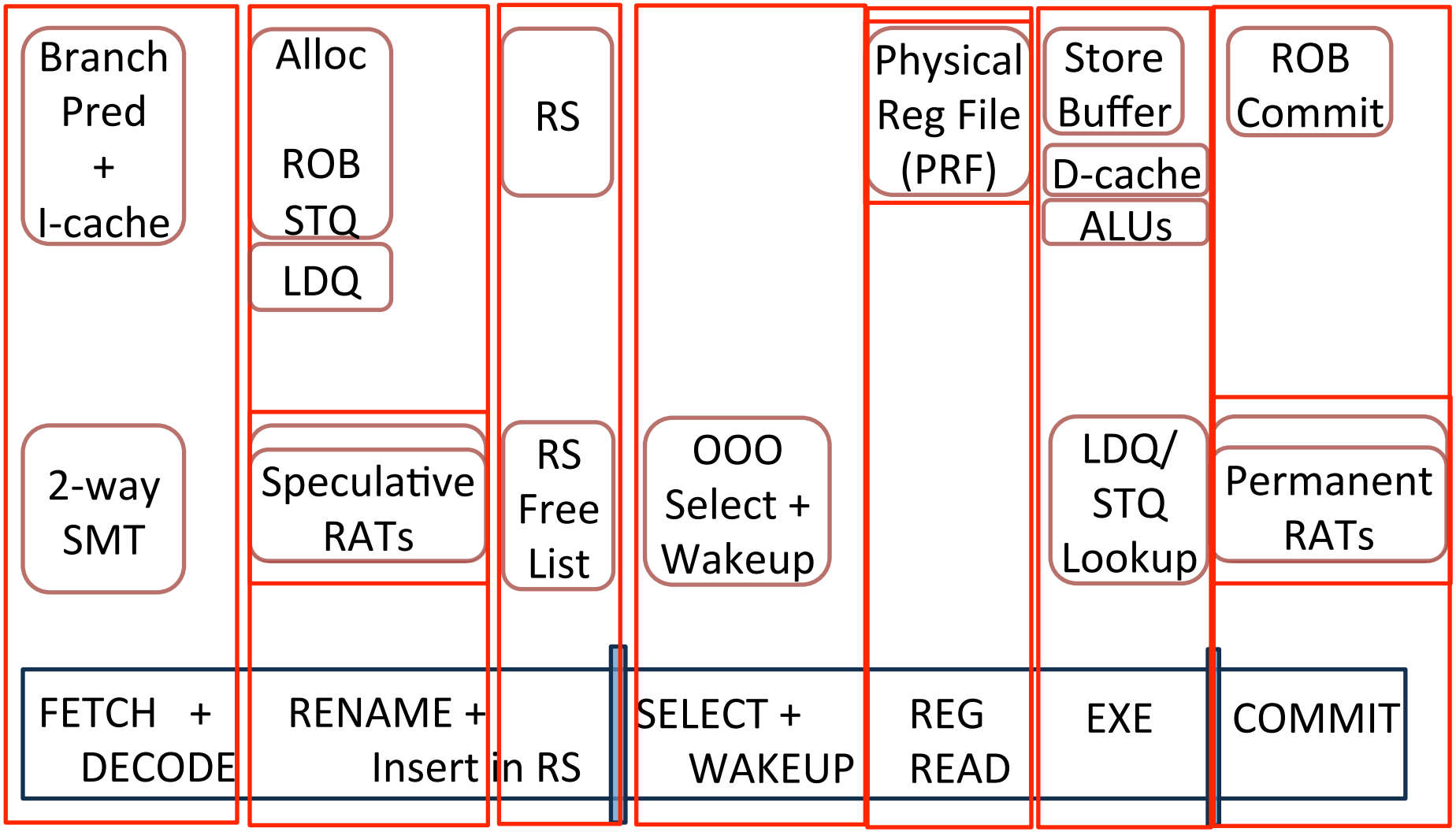
Two modes:

OutOfOrder	out-of-order core Exploits ILP High single-thread performance
InOrder	highly-threaded in-order SMT core Exploits TLP High multi-thread performance No OOO execution → Energy savings

Outline

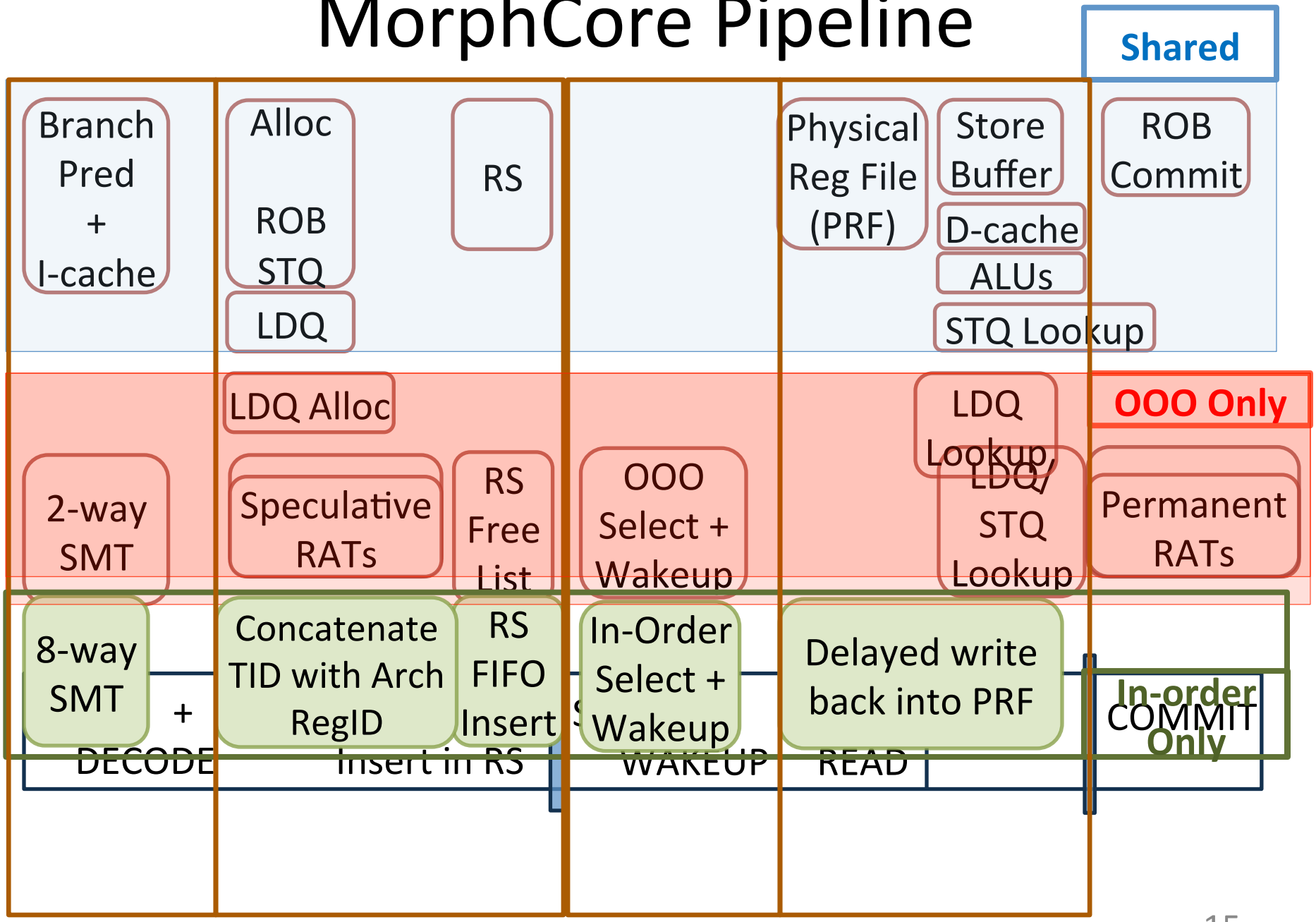
- Problem Statement
- Previous Work
- MorphCore
  - Key Insights and Basic Idea
  - Design and Operation
- Evaluation

Baseline OOO Pipeline



MorphCore Pipeline

MorphCore Pipeline



Microarchitecture Summary

- Use existing structures without modification
  - Physical Register File (PRF), Decode, Execution pipeline
- Use existing structures with minor modification
  - OOO Reservation Stations → InOrder instruction queues
  - Because of InOrder execution, delayed writeback into PRF (extra bypass)
- SMT related changes
  - Front-end (e.g. multiple PCs, branch history regs), changes in resource allocation algorithms
- In-Order instruction scheduler

Overheads

- Core area increases by 1.5%
  - Increase in SMT contexts (0.5%)
    (Note that added contexts are in-order, so no additional rename tables and physical registers)
  - InOrder Wakeup and Select Logic (0.5%)
  - Extra bypass (0.5%)
- Core frequency decreases by 2.5%
  - Add multiplexers in the critical path of 2 stages
    - Rename and Scheduling

Mode Switching Policy

- Number of active threads ≤ 2 ?

- OutofOrder when active threads ≤ 2
  - MorphCore can support up to 2 OOO threads
  - TLP is limited so execute OOO to obtain performance

- InOrder when active threads > 2
  - More than 2 threads can only run simultaneously in InOrder mode
  - TLP is high so high core throughput and energy savings can be obtained by executing threads in-order

How Mode Switching Happens?

(1) Drains the core pipeline

(2) Spills architectural registers of currently active threads to reserved ways in the private 256KB L2

(3) Turns off/on Renaming, OOO Scheduling, Load Queue

(4) Fills the architectural registers of next-active threads into PRF (update RATs when going into OutofOrder)

Currently an overhead of 300 - 450 cycles

Outline

- Problem Statement
- Previous Work
- MorphCore
- Evaluation

Methodology

- Detailed cycle-level x86 simulator
- McPAT (modified) to calculate energy/area

- Performance/energy evaluation of MorphCore vs. alternative architectures
  - Large OOO cores: optimized for single-thread
  - Medium and Small cores: optimized for multi-thread

- Workloads
  - Single-threaded (ST): 14 – SPEC 2006
  - Multi-threaded (MT): 14 – Databases, SPLASH, others

Evaluated Architectures

All comparisons on *approximately* equal area

ST : single-thread MT: multi-thread

OOO : out-of-order InO : in-order

Core	# of cores	Freq. (GHz)	Type	Issue width	SMT threads Per core	Total threads	Peak throughput ops/cycle	
							ST	MT

Evaluated Architectures

All comparisons on *approximately* equal area

ST : single-thread MT: multi-thread

OOO : out-of-order InO : in-order

Core	# of cores	Freq. (GHz)	Type	Issue width	SMT threads Per core	Total threads	Peak throughput ops/cycle	
							ST	MT
OOO-2	1	3.4	OOO	4	2	2	4	4
OOO-4	1	-5%	OOO	4	4	4	4	4

Evaluated Architectures

All comparisons on *approximately* equal area

ST : single-thread MT: multi-thread

OOO : out-of-order InO : in-order

Core	# of cores	Freq. (GHz)	Type	Issue width	SMT threads Per core	Total threads	Peak throughput ops/cycle	
							ST	MT
OOO-2	1	3.4	OOO	4	2	2	4	4
OOO-4	1	-5%	OOO	4	4	4	4	4
MED	3	3.4	OOO	2	1	3	2	6

Evaluated Architectures

All comparisons on *approximately* equal area

ST : single-thread MT: multi-thread

OOO : out-of-order InO : in-order

Core	# of cores	Freq. (GHz)	Type	Issue width	SMT threads Per core	Total threads	Peak throughput ops/cycle	
							ST	MT
OOO-2	1	3.4	OOO	4	2	2	4	4
OOO-4	1	-5%	OOO	4	4	4	4	4
MED	3	3.4	OOO	2	1	3	2	6
SMALL	3	3.4	InO	2	2	6	2	6

Evaluated Architectures

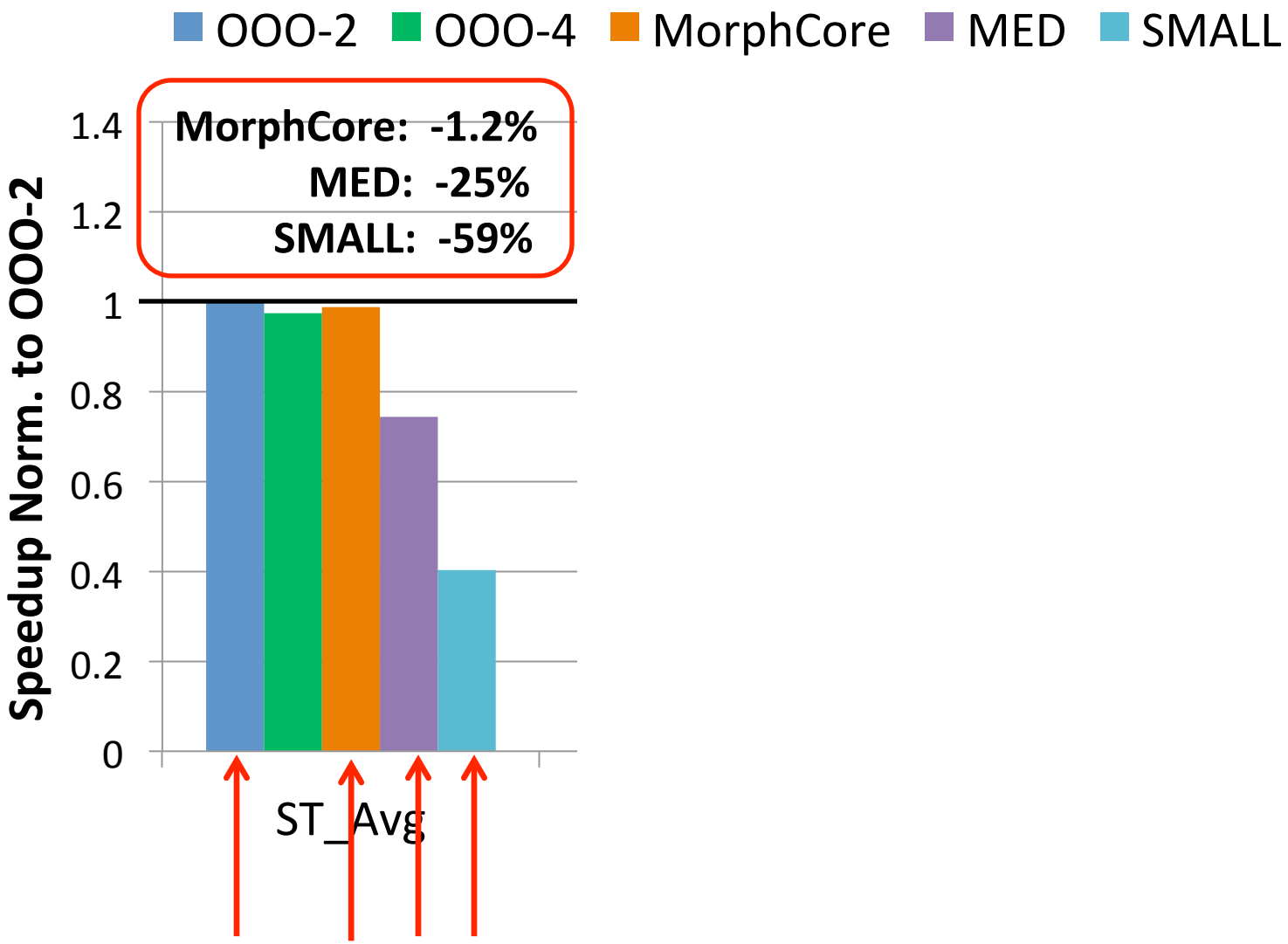
All comparisons on *approximately* equal area

ST : single-thread MT: multi-thread

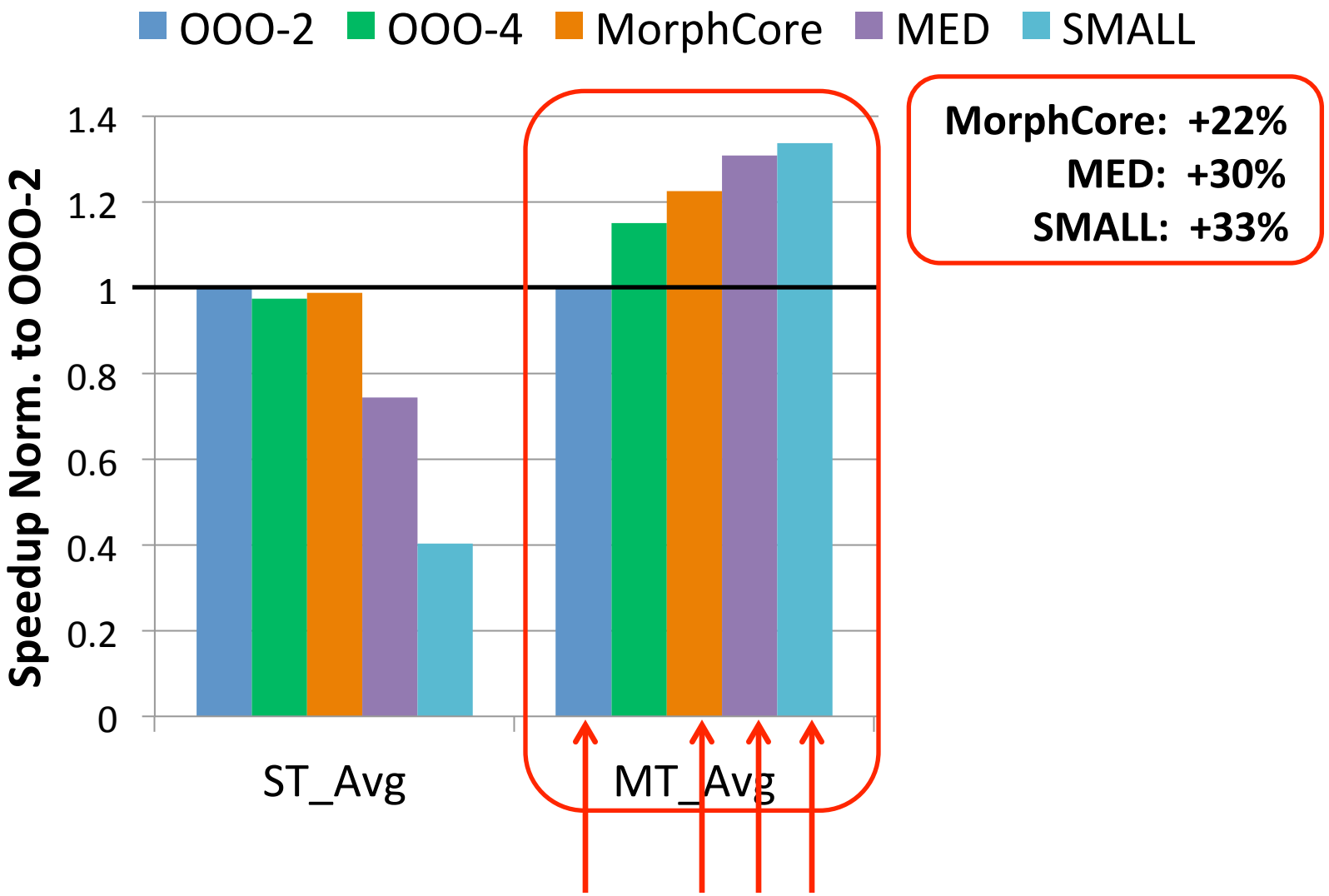
OOO : out-of-order InO : in-order

Core	# of cores	Freq. (GHz)	Type	Issue width	SMT threads Per core	Total threads	Peak throughput (ops/cycle)	
							ST	MT
OOO-2	1	3.4	OOO	4	2	2	4	4
OOO-4	1	-5%	OOO	4	4	4	4	4
MED	3	3.4	OOO	2	1	3	2	6
SMALL	3	3.4	InO	2	2	6	2	6
MorphCore	1	-2.5%	OOO/ InO	4	2 OOO/ 8 InO	2 OOO/ 8 InO	4	4

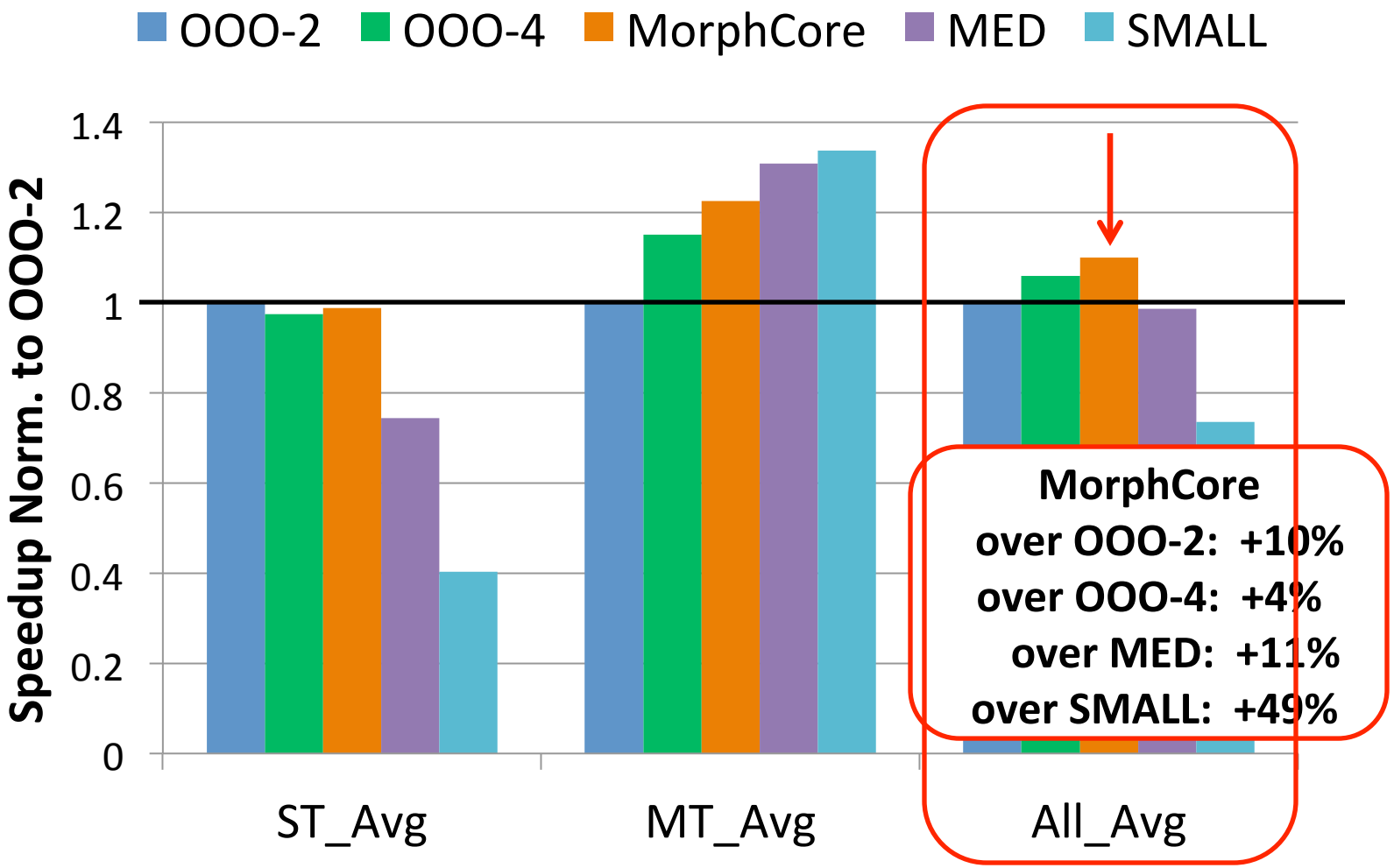
Performance: Single-thread



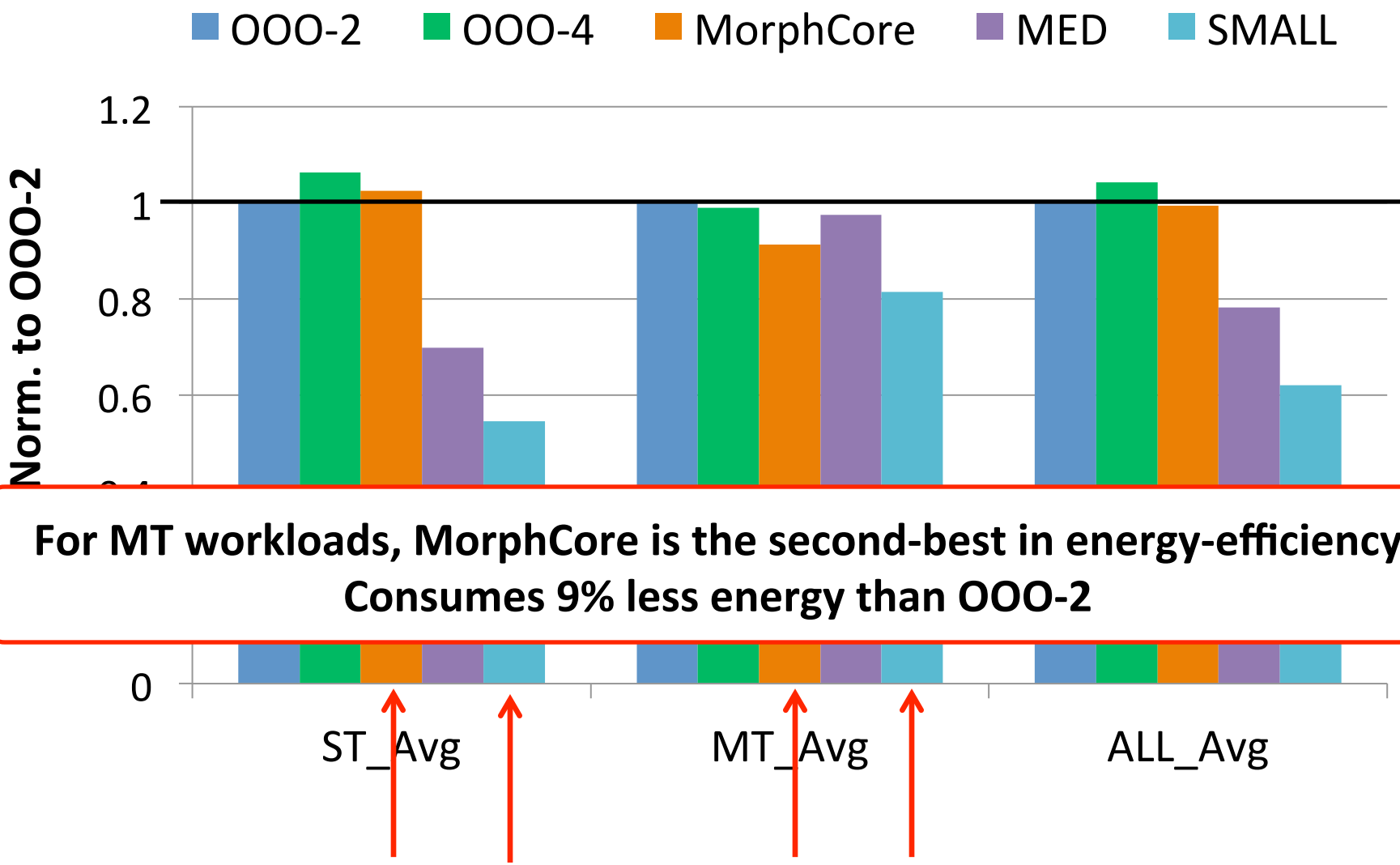
Performance: Multi-thread



Performance: Both ST and MT

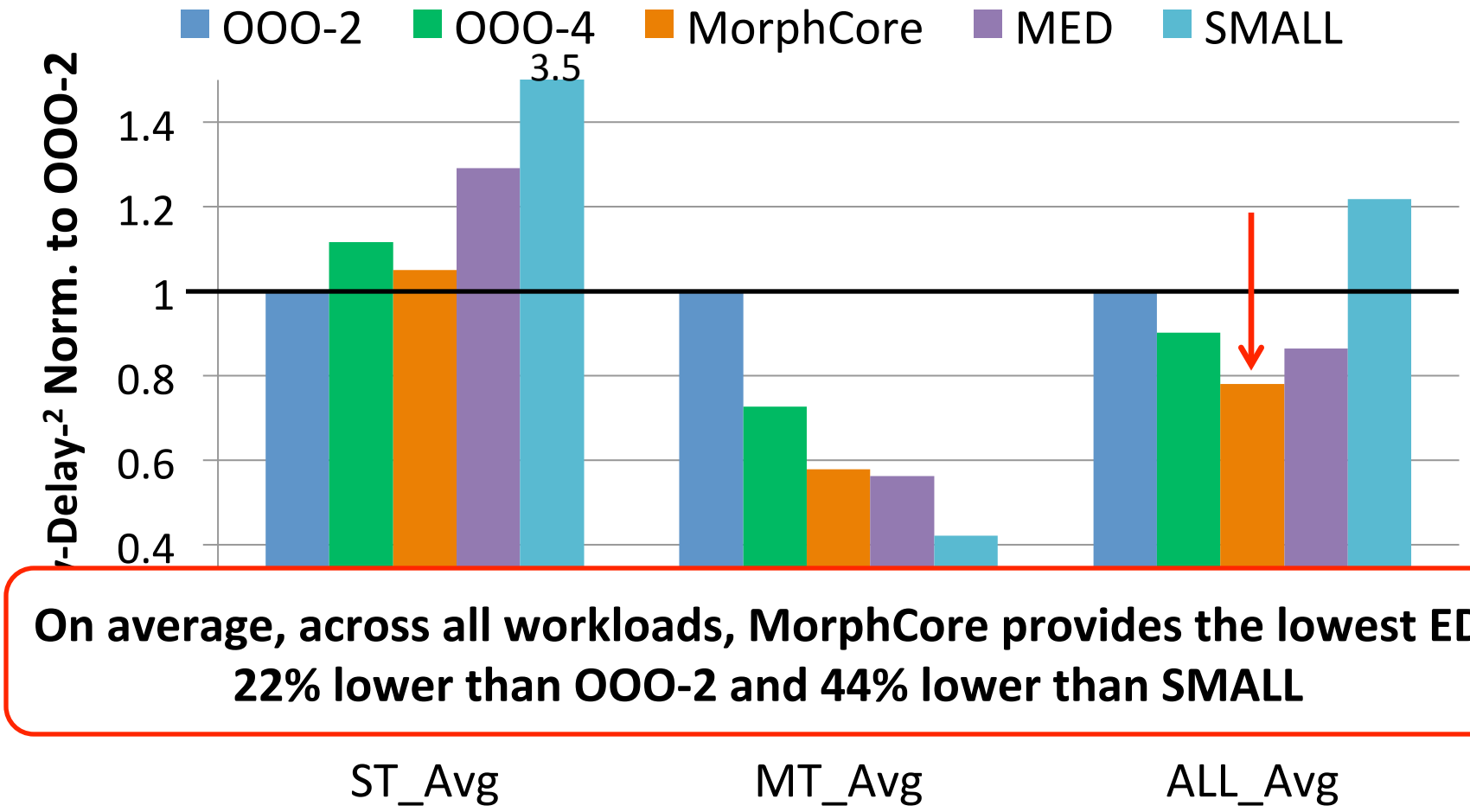


Energy



For MT workloads, MorphCore is the second-best in energy-efficiency
Consumes 9% less energy than OOO-2

Energy-delay-squared (ED^2)



On average, across all workloads, MorphCore provides the lowest ED^2
22% lower than OOO-2 and 44% lower than SMALL

Summary

- MorphCore adapts well to both single-thread and multi-thread workloads
- Requires minimal changes to a traditional OOO core
- Operates in two modes:
  - OOO core when TLP is low
  - Highly-threaded in-order SMT core when TLP is high
- Significantly outperforms other alternative architectures