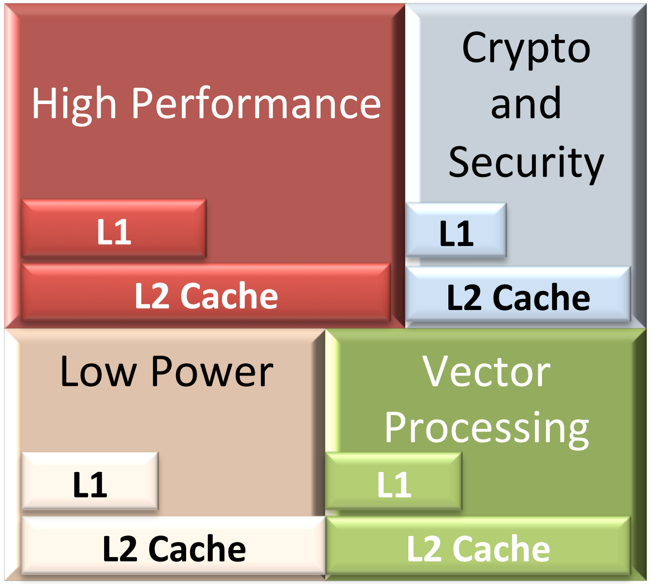
Harnessing ISA Diversity: Design of a Heterogeneous-ISA Chip Multiprocessor

Ashish Venkat Dean M. Tullsen

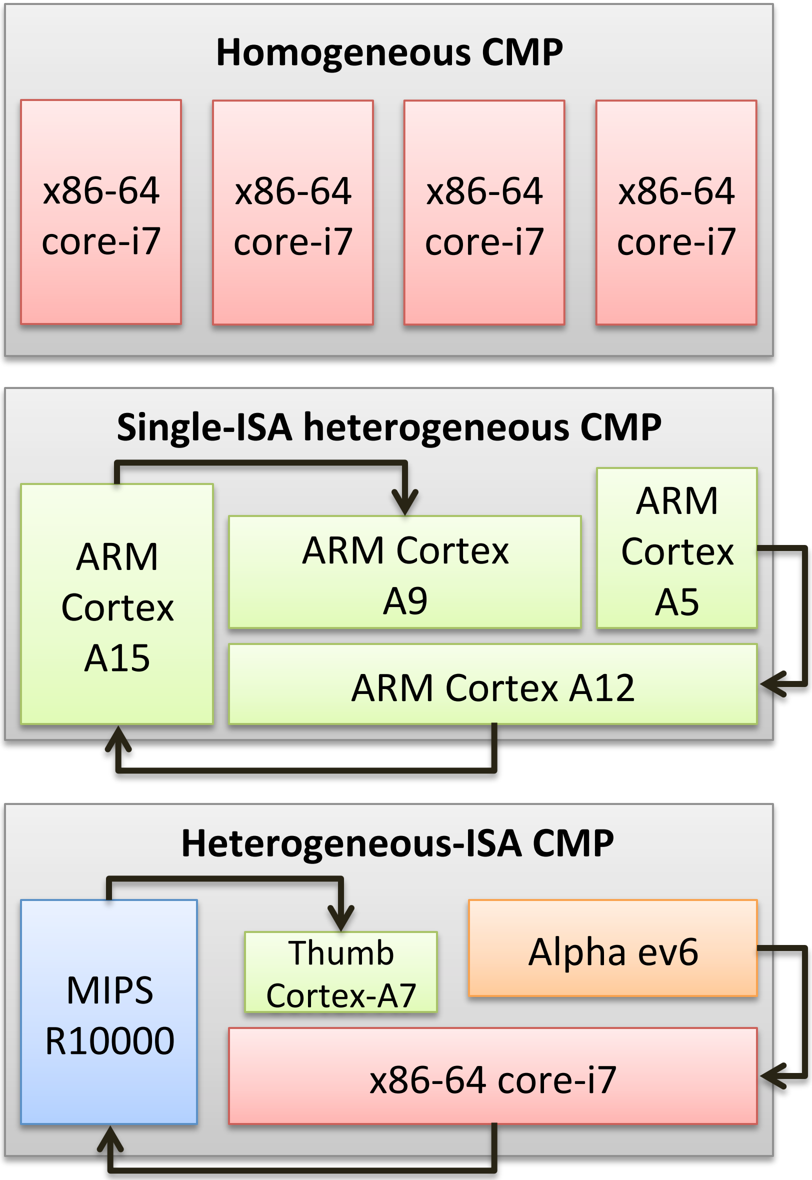
University of California, San Diego

Heterogeneous Chip Multiprocessors

- Commonplace in both general-purpose and embedded worlds.
- Heterogeneity is often exploited in two fundamental dimensions:-
  - Core Specialization: accelerate the performance of certain workloads
  - Micro-architectural Heterogeneity: use small power-efficient and large high performance cores



Heterogeneous Chip Multiprocessors



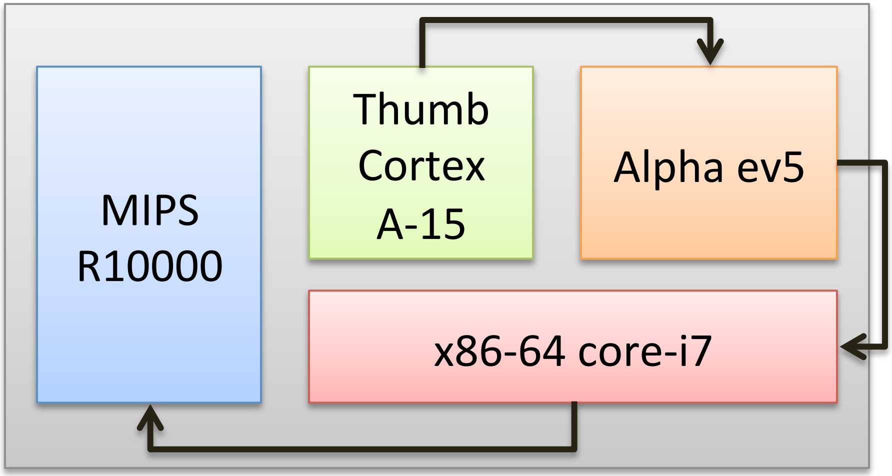
63% speedup
OR
69% energy savings with
3% performance loss*

Restricting cores to a single ISA eliminates an important dimension of heterogeneity

*Rakesh Kumar, Keith Farkas, Norm P. Jouppi, Partha Ranganathan, Dean M. Tullsen, MICRO'03

Why is ISA-heterogeneity advantageous?

- Enables ISA-microarchitecture co-design
  - There is significant synergy in combining heterogeneous-ISAs with heterogeneous hardware
- Exploits ISA-affinity
  - Applications have a natural ISA preference



Do existing ISAs provide sufficient heterogeneity?

Harnessing ISA diversity

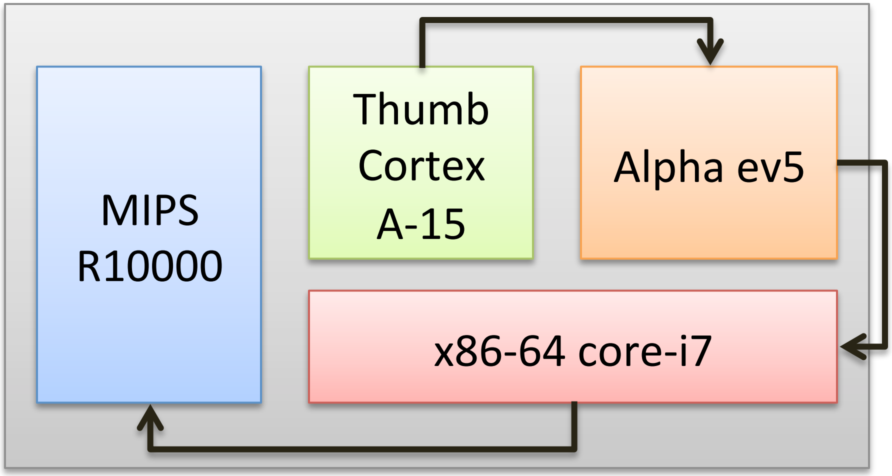
- In our design space exploration, we employ three modern ISAs:
  - ARM's energy-efficient Thumb ISA
  - The high performance x86-64 ISA
  - The simple and traditionally-RISC Alpha ISA
- They encompass several axes of ISA diversity
  - Code density, instruction complexity, register pressure, predication support, floating-point arithmetic vs emulation and SIMD processing

However . . .

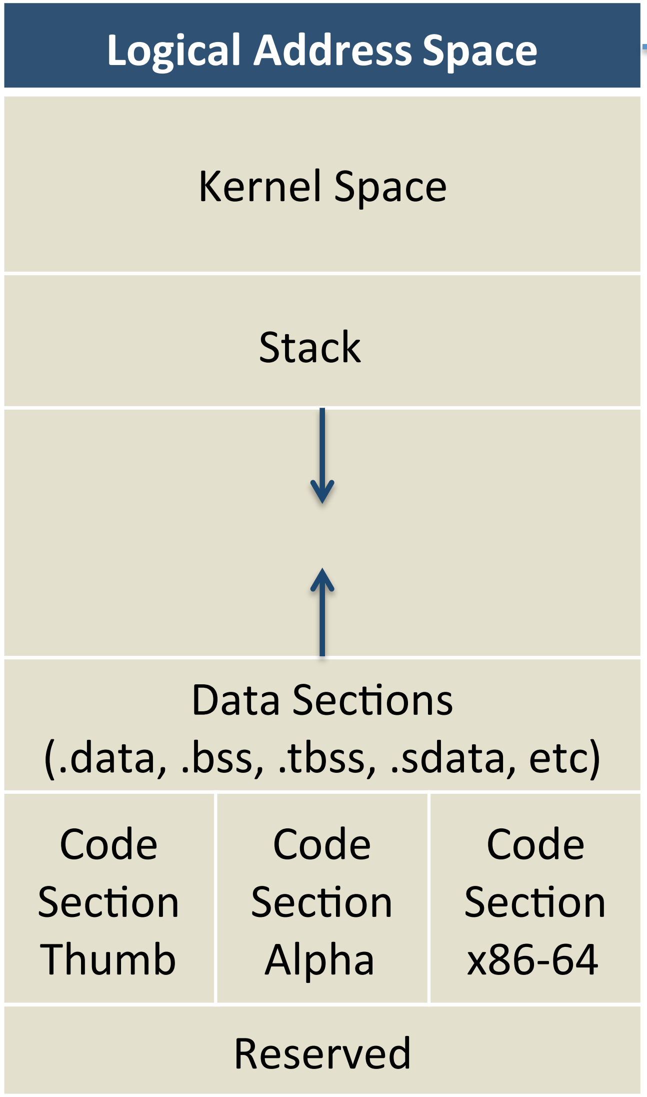
To fully harness ISA diversity, execution migration is critical

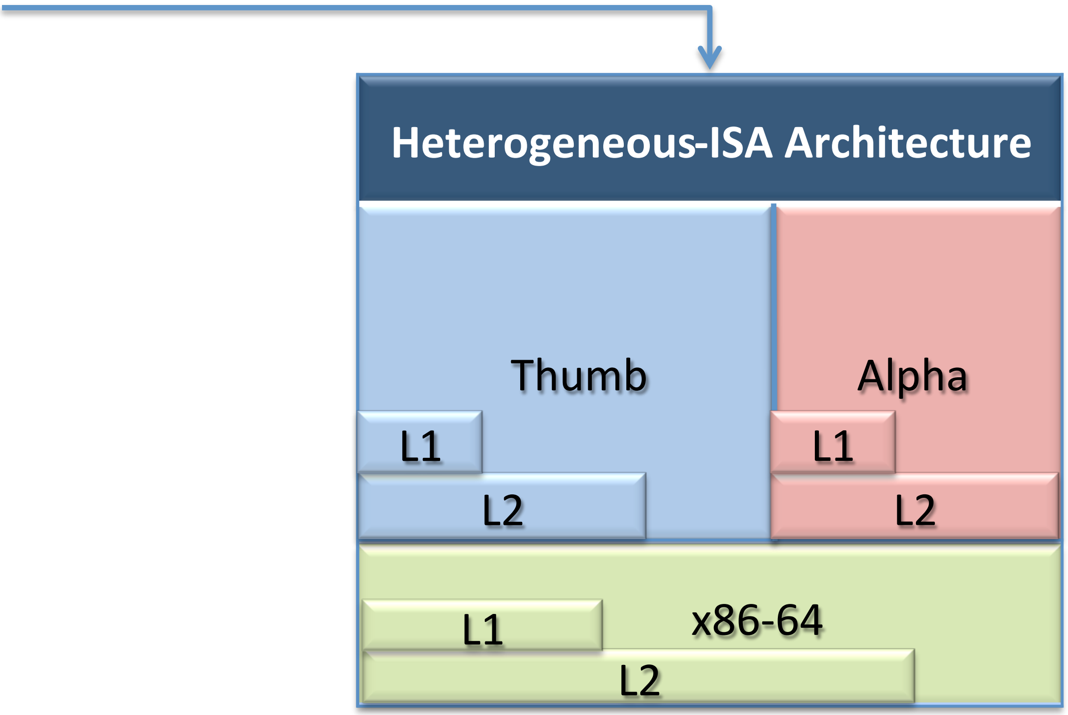
Execution Migration

- Allows an application to execute on the ISA of its preference, during different phases of execution
- Allows switching execution to a low power core when the power cord is plugged out
- Enables load balancing



Execution Migration in a Heterogeneous-ISA CMP





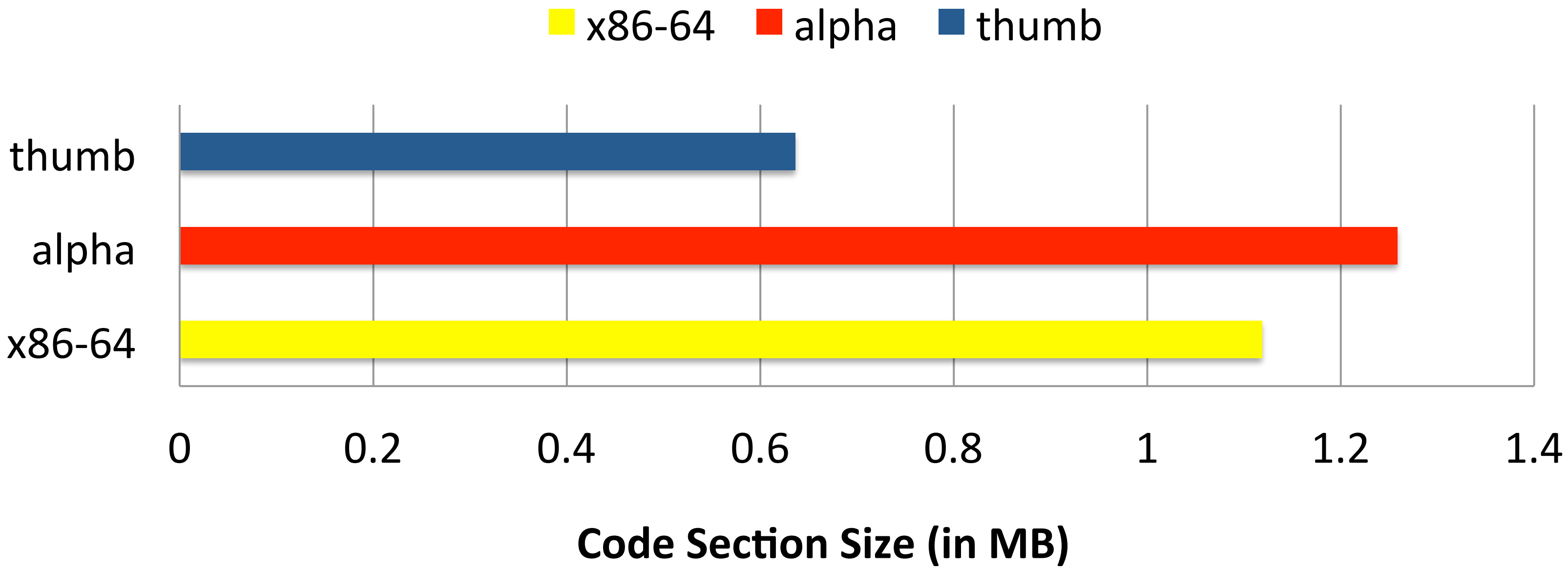
Symmetrical Fat Binary

All data objects are consistently referenced by the same address in all ISAs

Outline

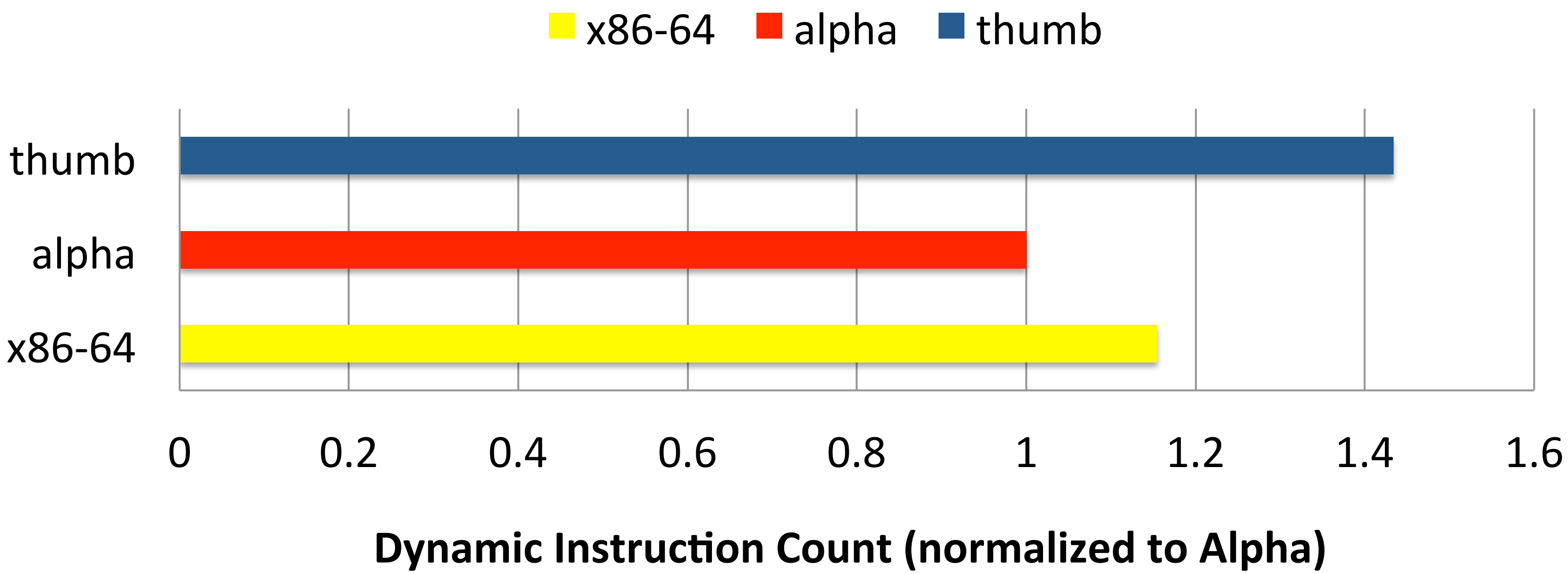
- Motivation
- ISA diversity
- Design Space Exploration
  - Navigation and Optimization
  - Inference: ISA-microarchitecture co-design
  - Inference: ISA-affinity
- Compilation and Runtime Strategy
- Key Points

ISA diversity: Code Density



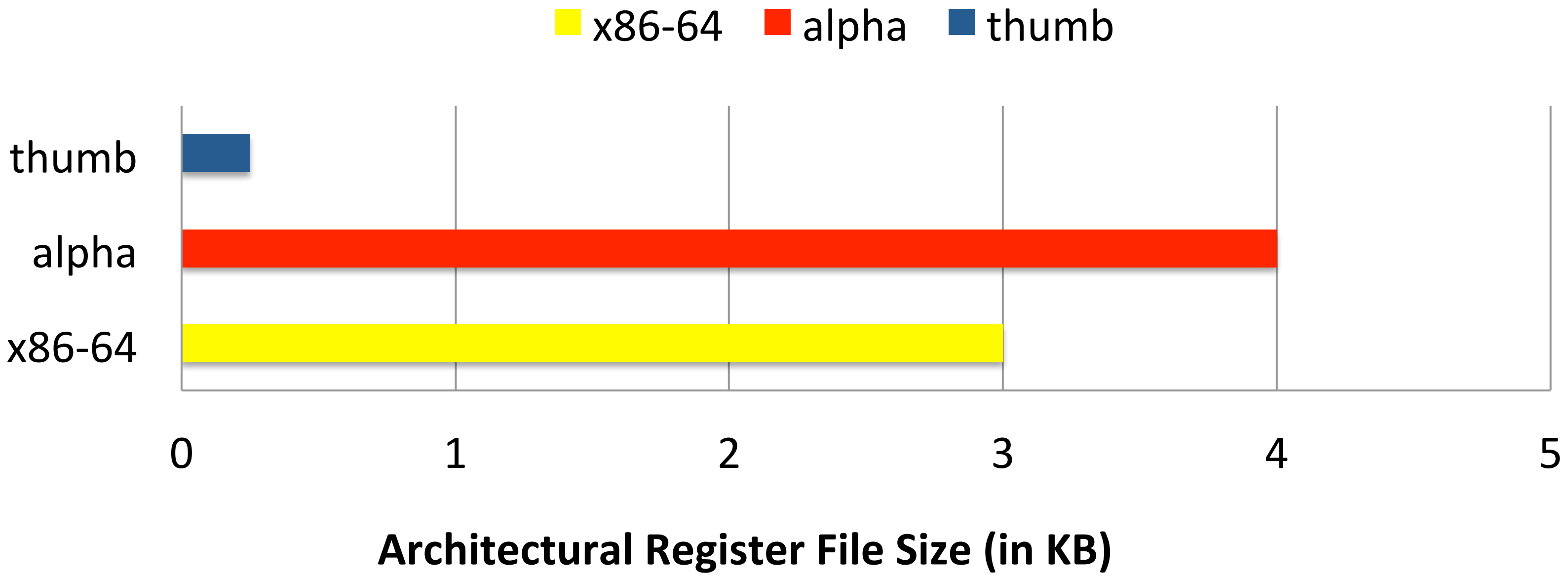
- Alpha: fixed-length encoding
- x86-64: variable-length encoding
- Thumb: code compression

ISA diversity: Instruction Complexity



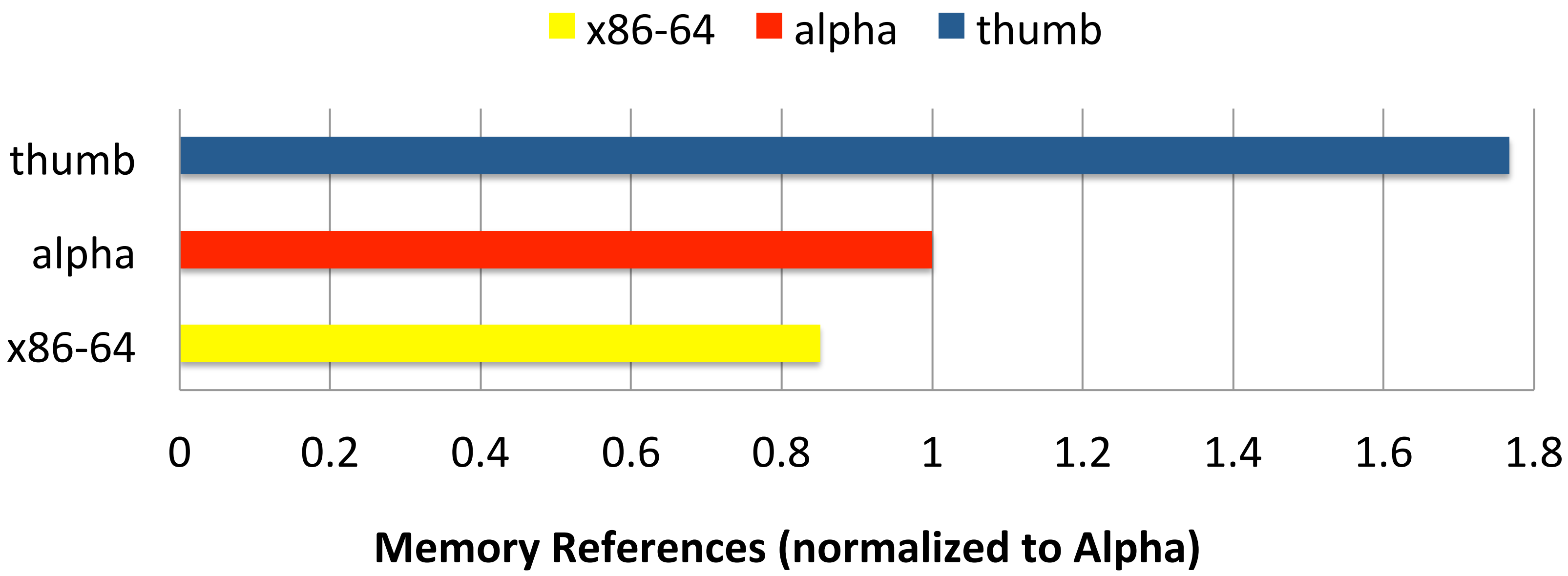
- Thumb: reduced encoding space (2-operand instructions)
- Alpha: load-store ISA (3-operand instructions)
- x86-64: 2-operand instructions + complex addressing modes

ISA diversity: Register Pressure



- Thumb: Eight 32-bit INT registers
- Alpha: Two banks of thirty-two 64-bit INT and FP registers
- x86-64: Sixteen 64-bit INT and Sixteen 128-bit SSE registers

ISA diversity: Register Pressure



Register File Tradeoffs:

- Size and Power Dissipation: thumb < x86-64 < alpha
- Register Pressure: x86-64 < alpha < thumb

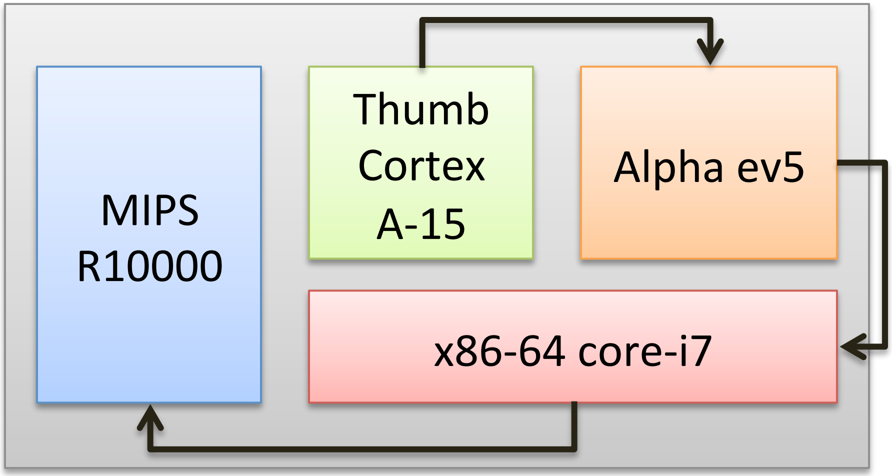
ISA diversity: Feature Sets

- Floating-point operations in Thumb
  - Emulated in software or execution is migrated to a different core
  - Thumb cores don't include FP instruction windows, register files, and functional units – 30% savings in area and 20% reduction in TDP
- SIMD operations in Alpha
  - Primitive: allows pack, unpack, max and min
  - We forgo SIMD units in Alpha to save area and power

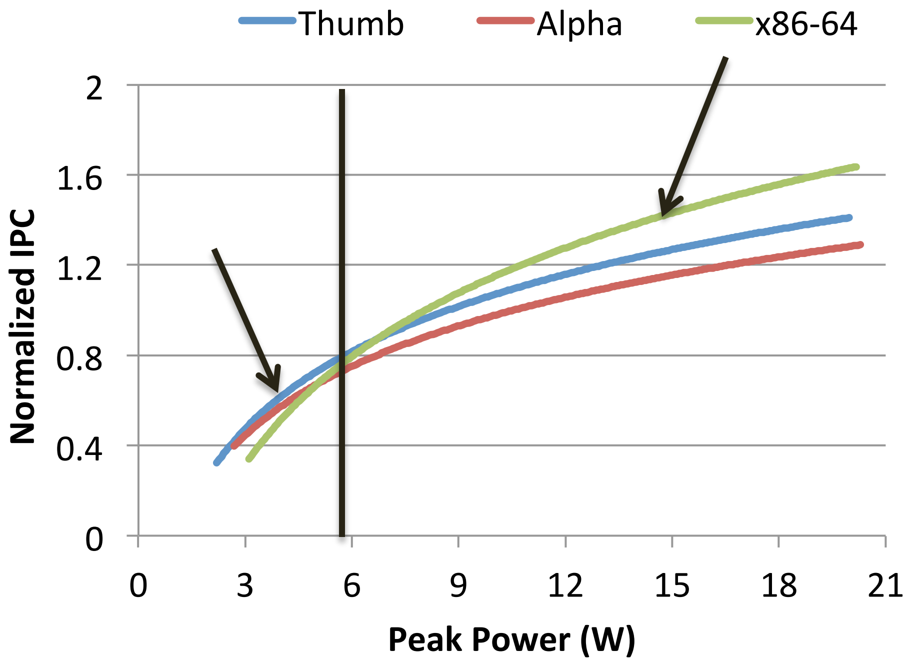
Why is ISA-heterogeneity advantageous?

- Enables ISA-microarchitecture co-design
  - Does ISA diversity complement micro-architectural heterogeneity?

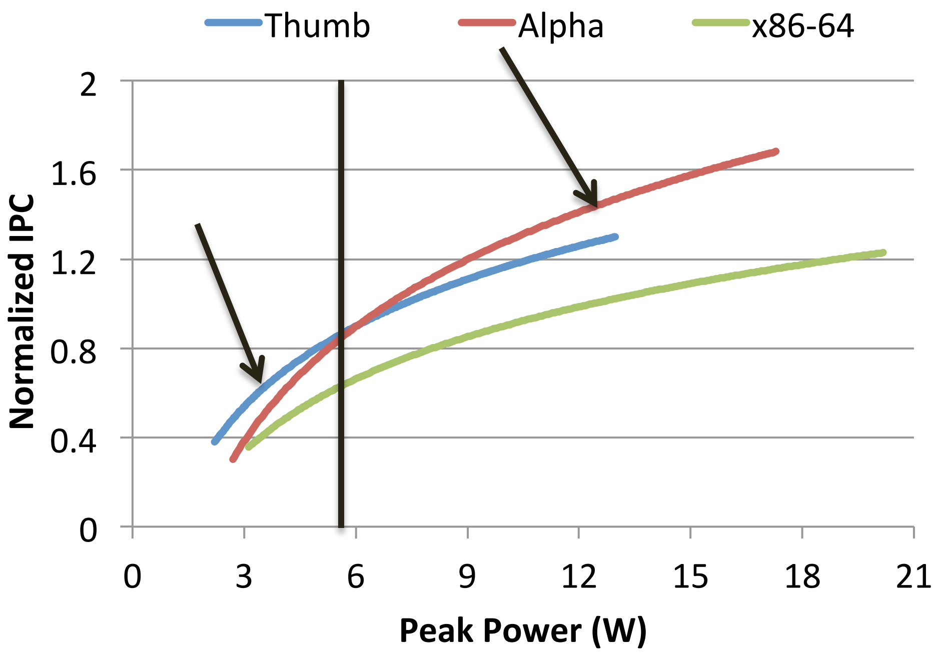
- Exploits ISA-affinity
  - Does ISA diversity enable ISA affinity?



Does ISA diversity enable ISA affinity?



bzip2 – execution phase 1



bzip2 – execution phase 2

- Phase 1 prefers x86-64
- Phase 2 prefers Alpha
- We always prefer Thumb at low power budgets

Outline

- Motivation
- ISA diversity
- Design Space Exploration
  - – Navigation and Optimization
  - – Inference: ISA-microarchitecture co-design
  - – Inference: ISA-affinity
- Compilation and Runtime Strategy
- Key Points

Design Space Exploration

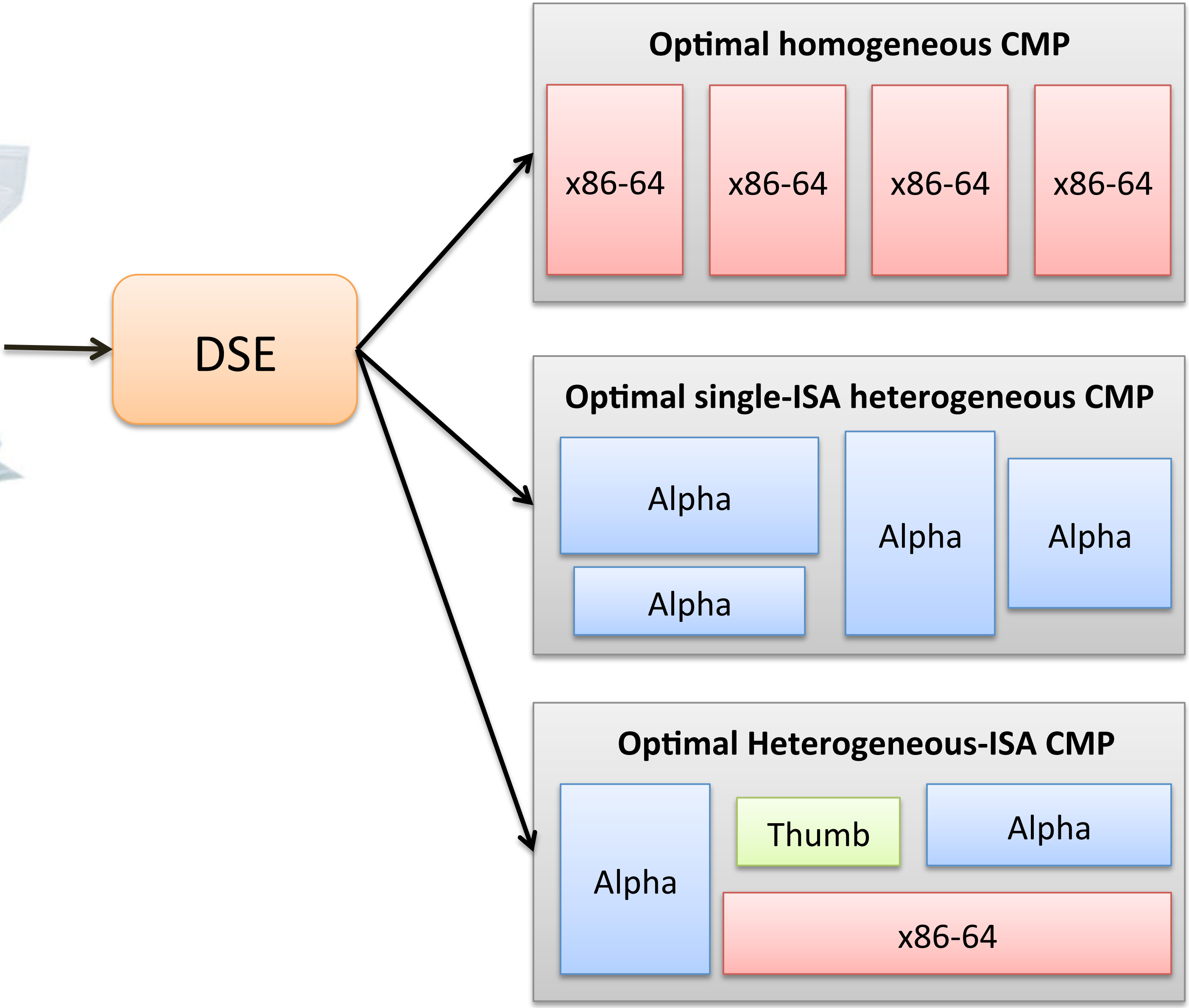
Micro-architectural parameters

+

ISA parameters

+

Budget constraints



DSE

Optimal homogeneous CMP

Optimal single-ISA heterogeneous CMP

Optimal Heterogeneous-ISA CMP

Design Space Exploration
Choice of ISAs

- To keep the design space exploration tractable, we select our target ISAs a priori
- Our target ISAs: Thumb, Alpha and x86-64
- Full ISA customization only increases the potential performance and energy gains

Design Space Exploration

Choice of micro-architectural parameters

Design Parameter	Design Choice
Execution Semantics	In-order, Out-of-order
Issue Width	1, 2, 4
Branch Predictor	Local, Tournament
Reorder Buffer Size	64, 128 entries
Physical Register File (integer)	96, 160
Physical Register File (FP/SIMD)	64, 96
Integer ALUs	1, 3, 6
Integer Multiply/Divide Units	1, 2
Floating-point ALUs	1, 2, 4
FP Multiply/Divide Units	1, 2
SIMD Units	1, 2, 4
Load/Store Queue	16,32 entries
Instruction Cache	32KB 4-way, 64KB 4-way
Private Data Cache	32KB 4-way, 64KB 8-way
Shared Last Level (L2) cache	4-banked 4MB 4-way, 4-banked 8MB 8-way

750 thousand single core and a septillion (10^{24}) 4-core configurations

Design Space Exploration

Choice of micro-architectural parameters

Design Parameter	Design Choice
Execution Semantics	In-order, Out-of-order
Issue Width	1, 2, 4
Branch Predictor	Local, Tournament
Reorder Buffer Size	64, 128 entries
Physical Register File (integer)	96, 160
Physical Register File (FP/SIMD)	64, 96
Integer ALUs	1, 3, 6
Integer Multiply/Divide Units	1, 2
Floating-point ALUs	1, 2, 4
FP Multiply/Divide Units	1, 2
SIMD Units	1, 2, 4
Load/Store Queue	16,32 entries
Instruction Cache	32KB 4-way, 64KB 4-way
Private Data Cache	32KB 4-way, 64KB 8-way
Shared Last Level (L2) cache	4-banked 4MB 4-way, 4-banked 8MB 8-way

750 thousand single core and a septillion (10^{24}) 4-core configurations

Design Space Exploration

Pruned design space

Design Parameter	Design Choice
ISAs	Thumb, Alpha, x86-64
Execution Semantics	In-order, Out-of-order
Issue Width-Function Units	1-low, 1-med, 2-med, 4-med, 4-high
Branch Predictor	Local, Tournament
ROB-IntReg-FPReg	64-96-64, 128-160-96
Load/Store Queue	16,32 entries
Cache Hierarchy	32K/4-4M/4, 32K/4-8M/8, 64K/4-4M/4, 64K/4-8M/8

600 single core and 130 billion 4-core configurations

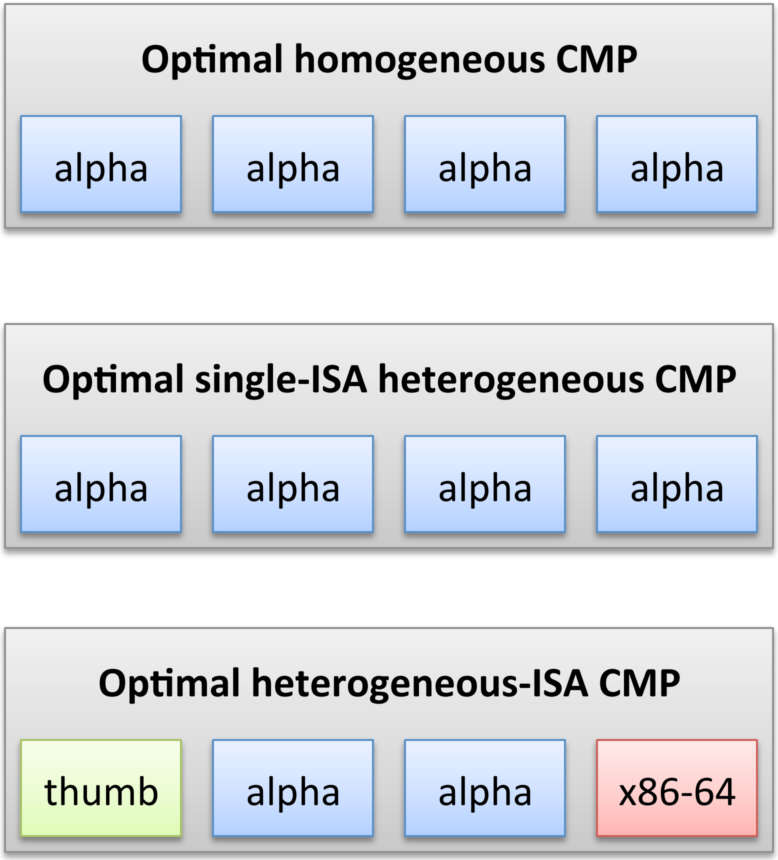
Design Space Exploration
Optimal Configurations

	Peak Power Budget (20W, 40W, 60W, unlimited)	Area Budget (48mm^2, 64mm^2, 80mm^2, unlimited)
Multi-programmed mixed workload		
Single-threaded workload		

28 optimal homogeneous, single-ISA heterogeneous and heterogeneous-ISA designs each

Design Space Exploration
Budget Constraints

Tight constraints
(all cores are small)



Optimal homogeneous CMP

alpha | alpha | alpha | alpha

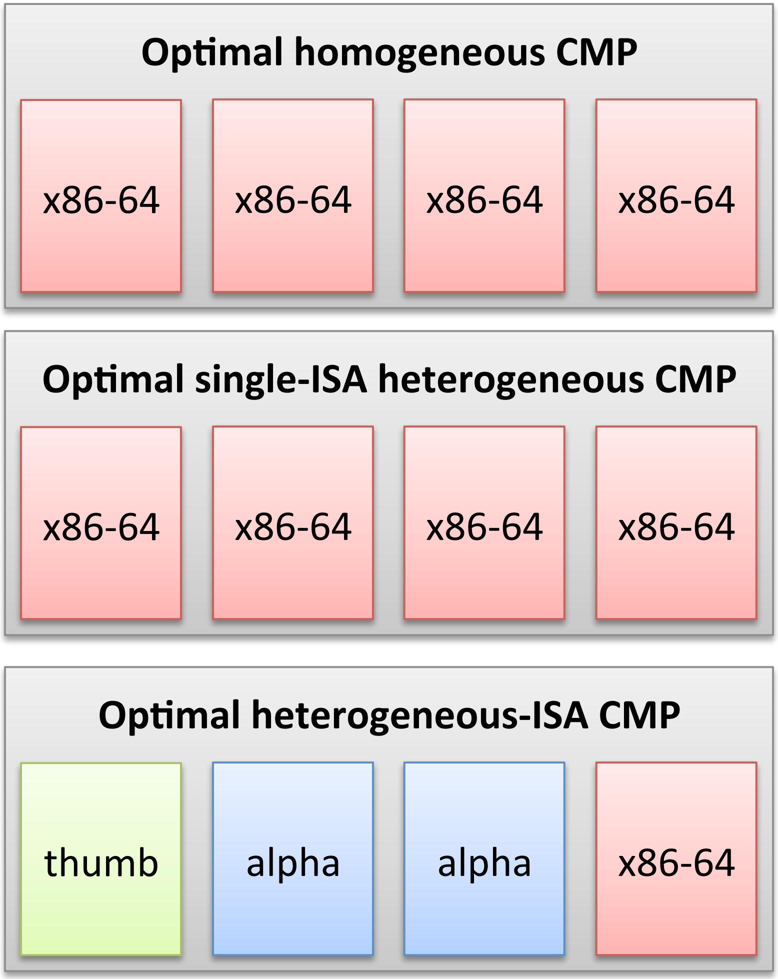
Optimal single-ISA heterogeneous CMP

alpha | alpha | alpha | alpha

Optimal heterogeneous-ISA CMP

thumb | alpha | alpha | x86-64

Liberal constraints
(all cores free to be big)



Optimal homogeneous CMP

x86-64 | x86-64 | x86-64 | x86-64

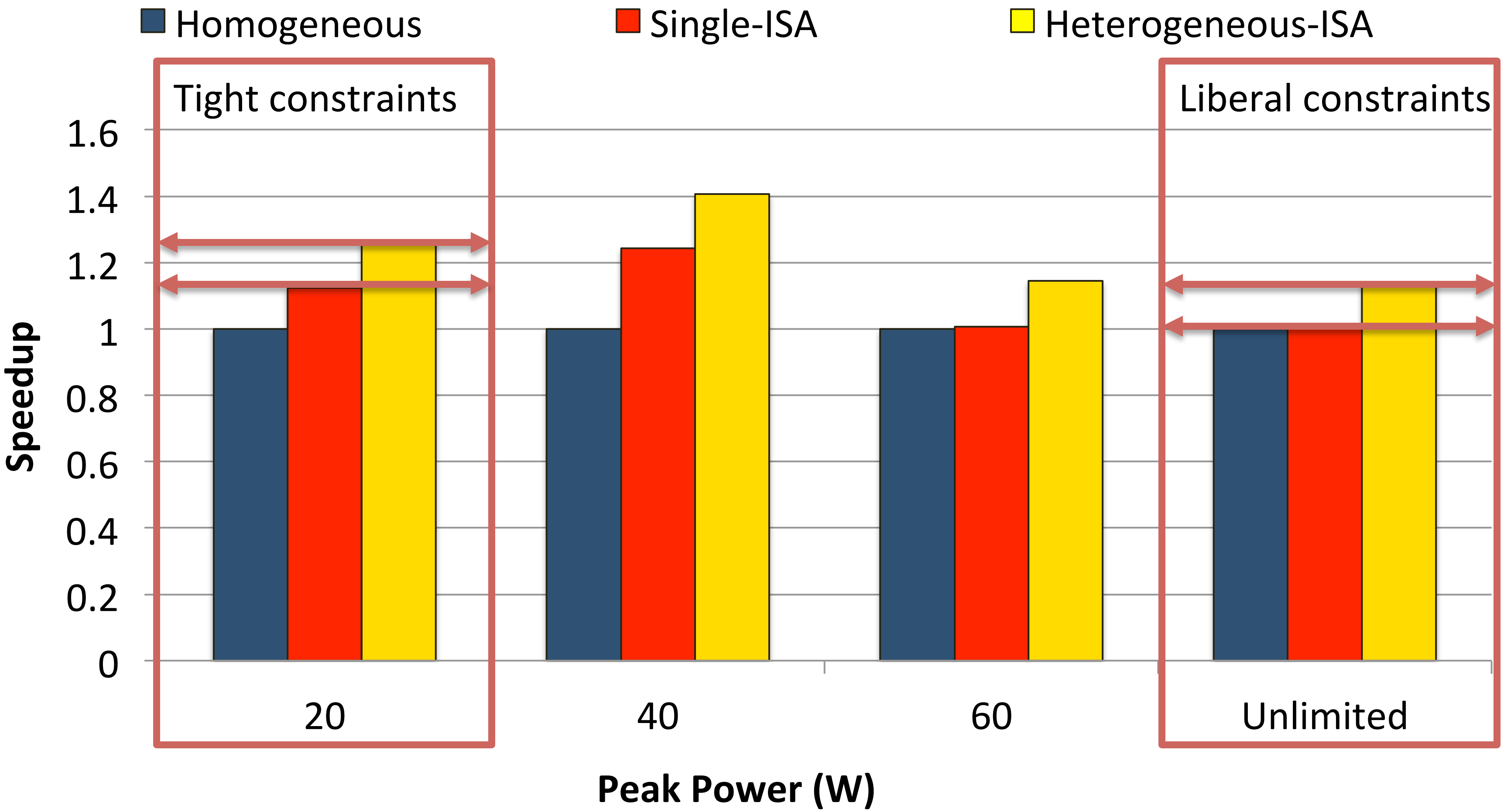
Optimal single-ISA heterogeneous CMP

x86-64 | x86-64 | x86-64 | x86-64

Optimal heterogeneous-ISA CMP

thumb | alpha | alpha | x86-64

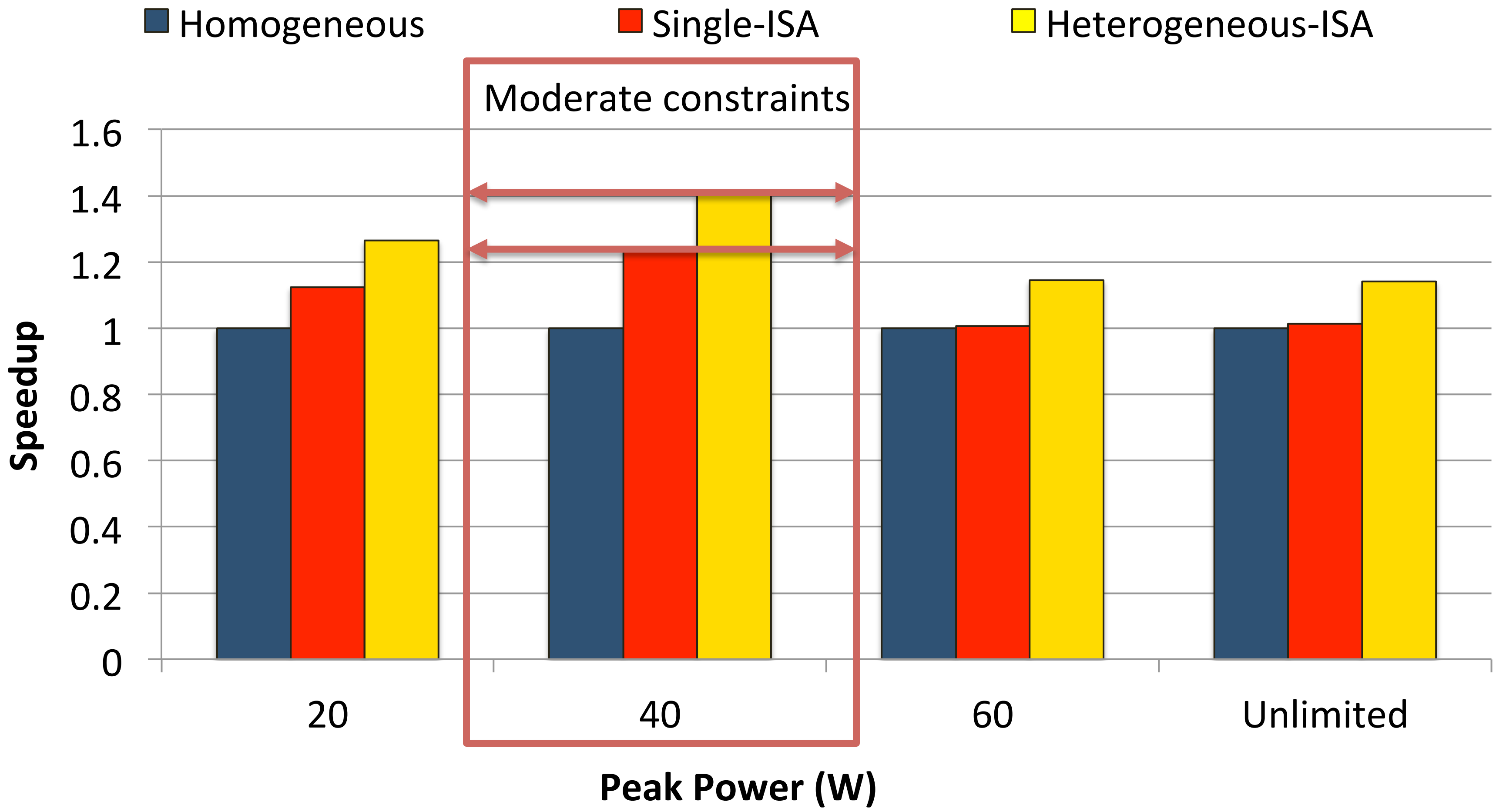
Design Space Exploration
Multi-programmed workload throughput



We always gain more from ISA heterogeneity than hardware heterogeneity

Design Space Exploration

Multi-programmed workload throughput

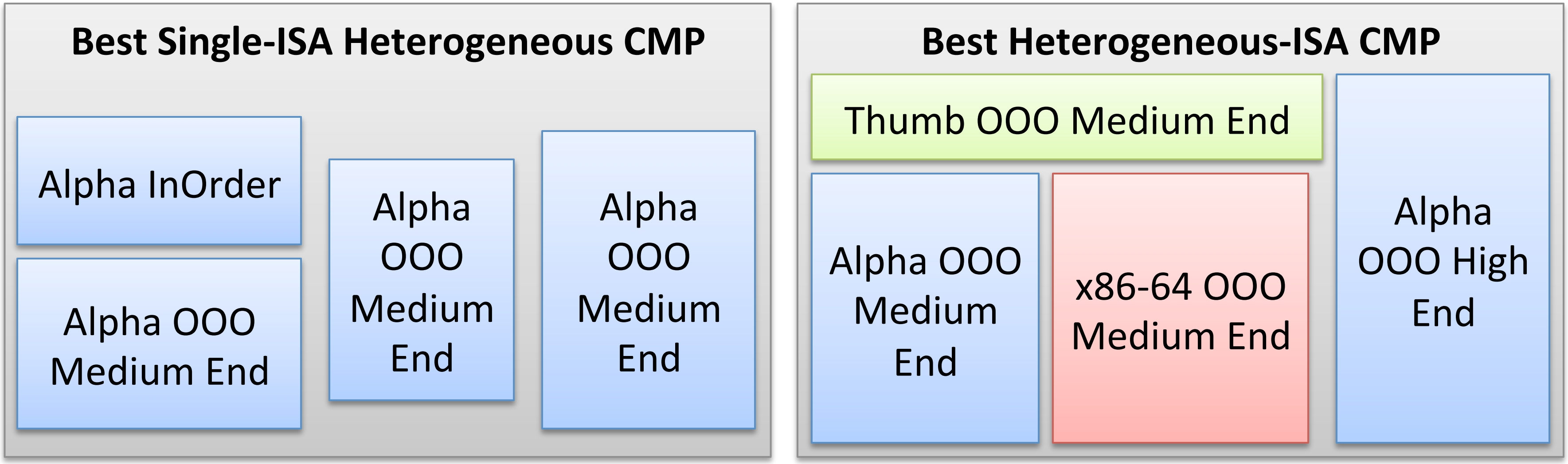


We always gain more from ISA heterogeneity than hardware heterogeneity

Design Space Exploration
Multi-programmed workload throughput

ISA-heterogeneity benefits come from:

- ISA-affinity: different code regions have a natural affinity for one ISA or another
- ISA-microarchitecture co-design: squeeze in more powerful cores into the same budget



Best Single-ISA Heterogeneous CMP

Alpha InOrder

Alpha OOO Medium End

Alpha OOO Medium End

Alpha OOO Medium End

Best Heterogeneous-ISA CMP

Thumb OOO Medium End

Alpha OOO Medium End

x86-64 OOO Medium End

Alpha OOO High End

Both designs are constrained at an area budget of 64mm^2

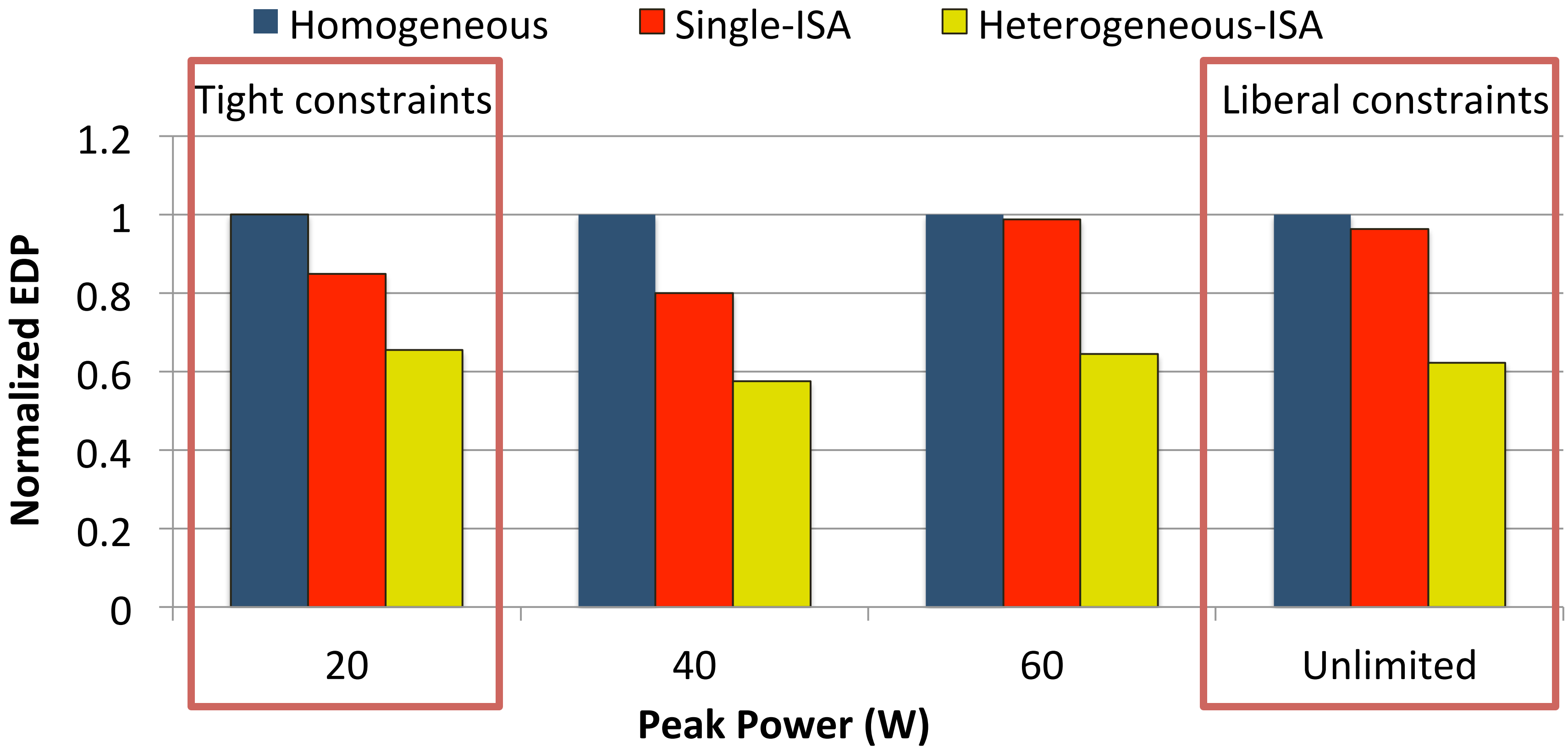
Design Space Exploration

Optimal Configurations

	Peak Power Budget (20W, 40W, 60W, unlimited)	Area Budget (48mm^2, 64mm^2, 80mm^2, unlimited)
Multi-programmed mixed workload		
Single-threaded workload		

28 optimal homogeneous, single-ISA heterogeneous and heterogeneous-ISA designs each

Design Space Exploration
Multi-programmed workload energy efficiency



- **22% energy savings and 28% reduction in EDP at ZERO performance loss**
- **We gain performance and decrease energy simultaneously**

Design Space Exploration

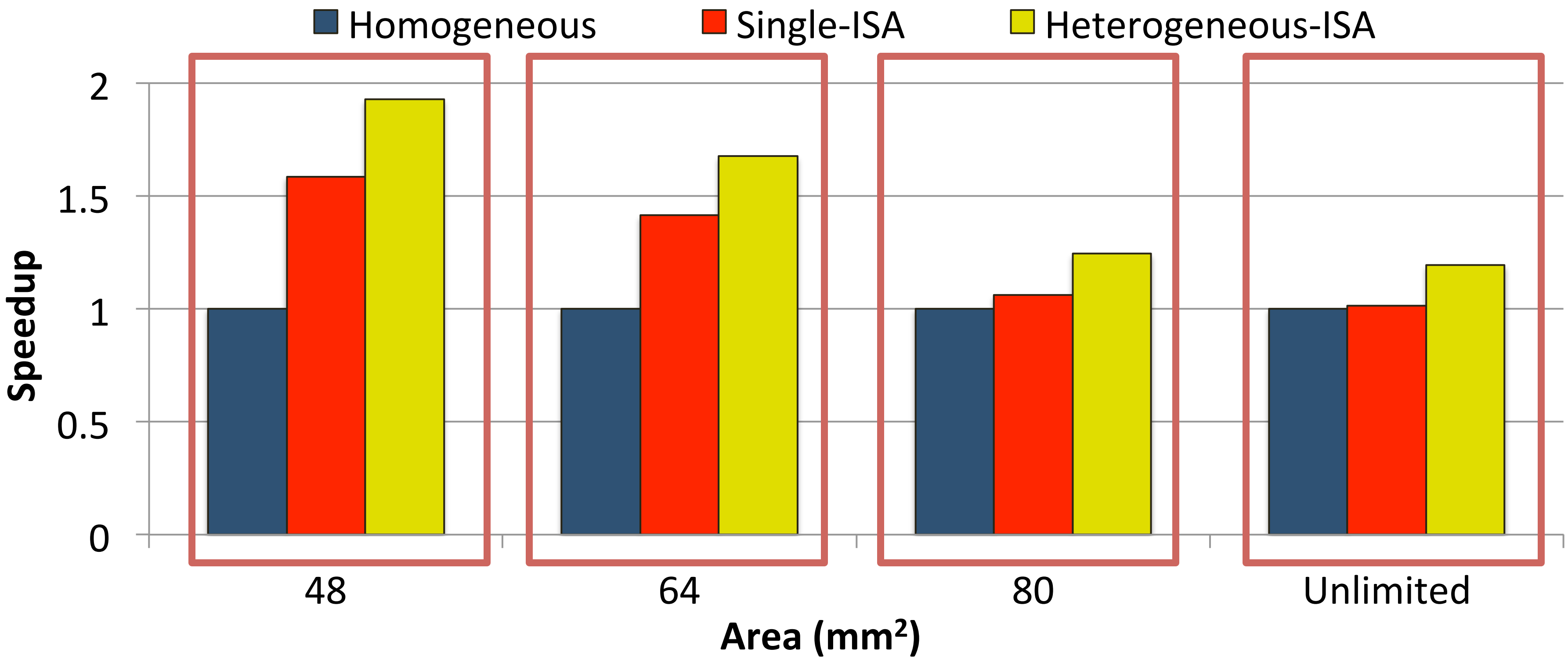
Optimal Configurations

	Peak Power Budget (20W, 40W, 60W, unlimited)	Area Budget ($48mm^2$, $64mm^2$, $80mm^2$, unlimited)
Multi-programmed mixed workload	 	
Single-threaded workload	 	

28 optimal homogeneous, single-ISA heterogeneous and heterogeneous-ISA designs each

Design Space Exploration

Single Thread Performance



- Multiple small cores and one large core optimized for high performance
- Combining the dual benefits of ISA-affinity and area efficiency of Thumb, heterogeneous-ISA CMPs provide **as much as 35% speedup, under the most tight area constraints**

Design Space Exploration

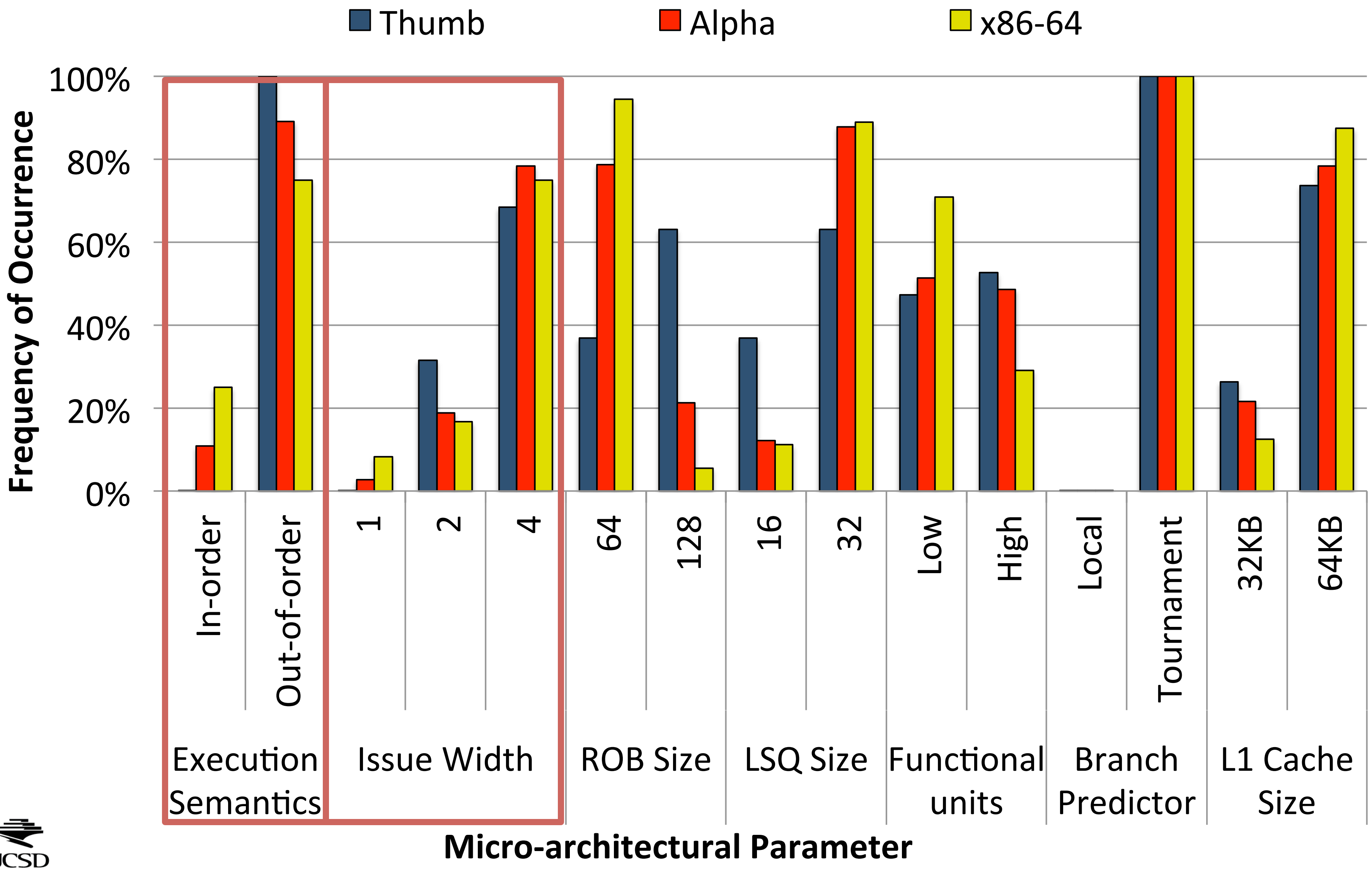
Optimal Configurations

	Peak Power Budget (20W, 40W, 60W, unlimited)	Area Budget (48mm^2, 64mm^2, 80mm^2, unlimited)
Multi-programmed mixed workload		
Single-threaded workload		

28 optimal homogeneous, single-ISA heterogeneous and heterogeneous-ISA designs each

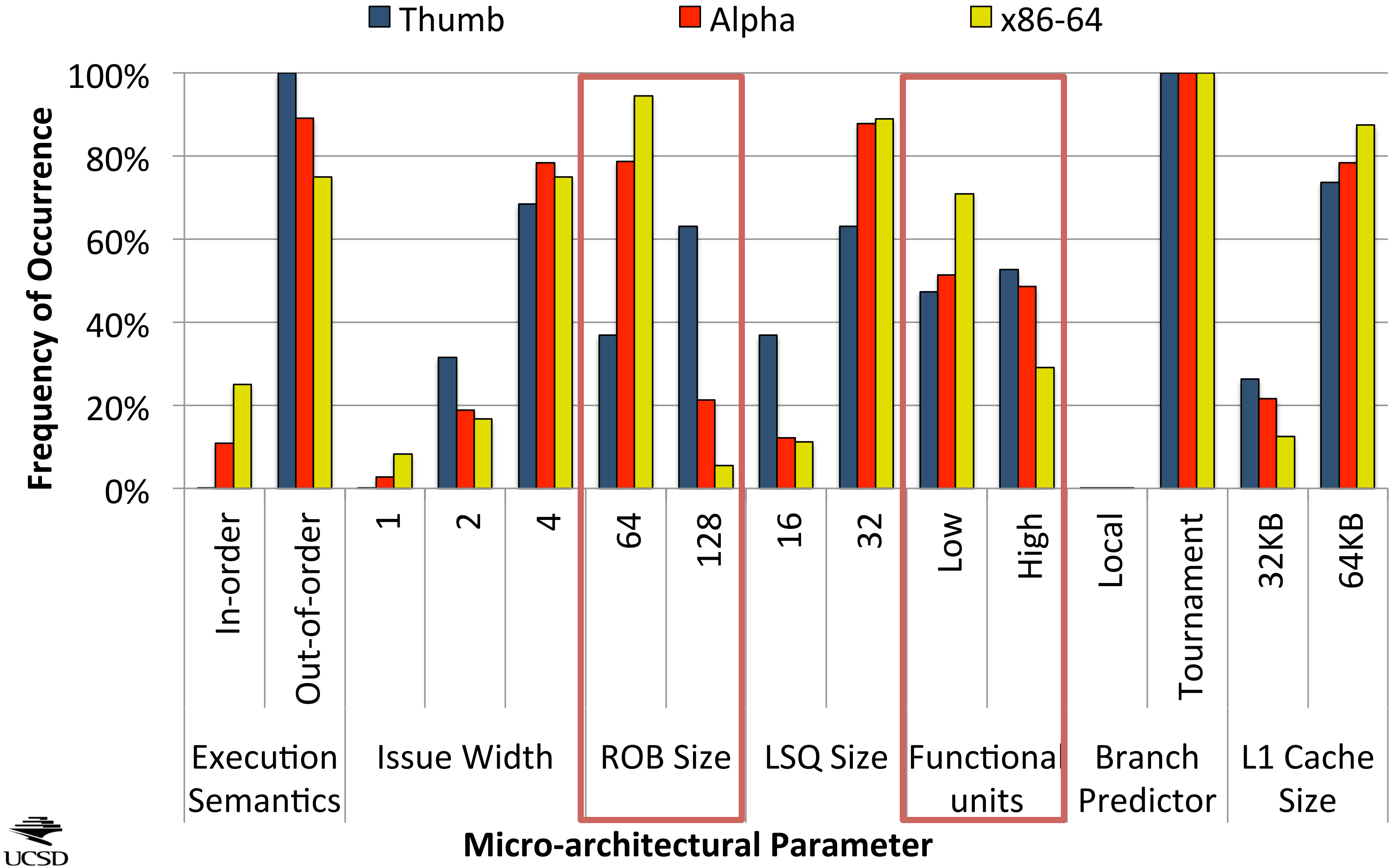
Design Space Exploration

Inferences – ISA-microarchitecture co-design



Design Space Exploration

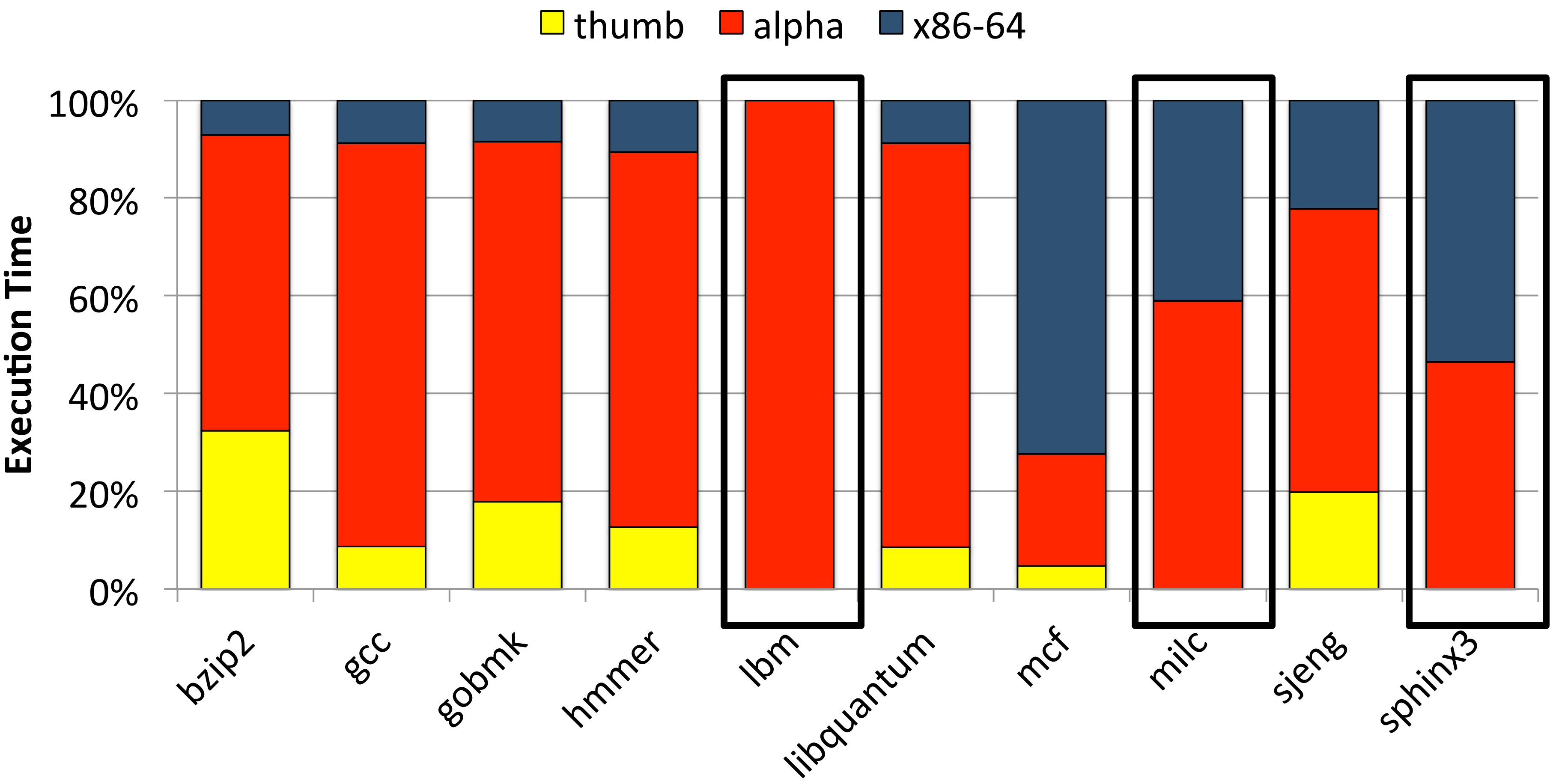
Inferences – ISA-microarchitecture co-design



UCSD

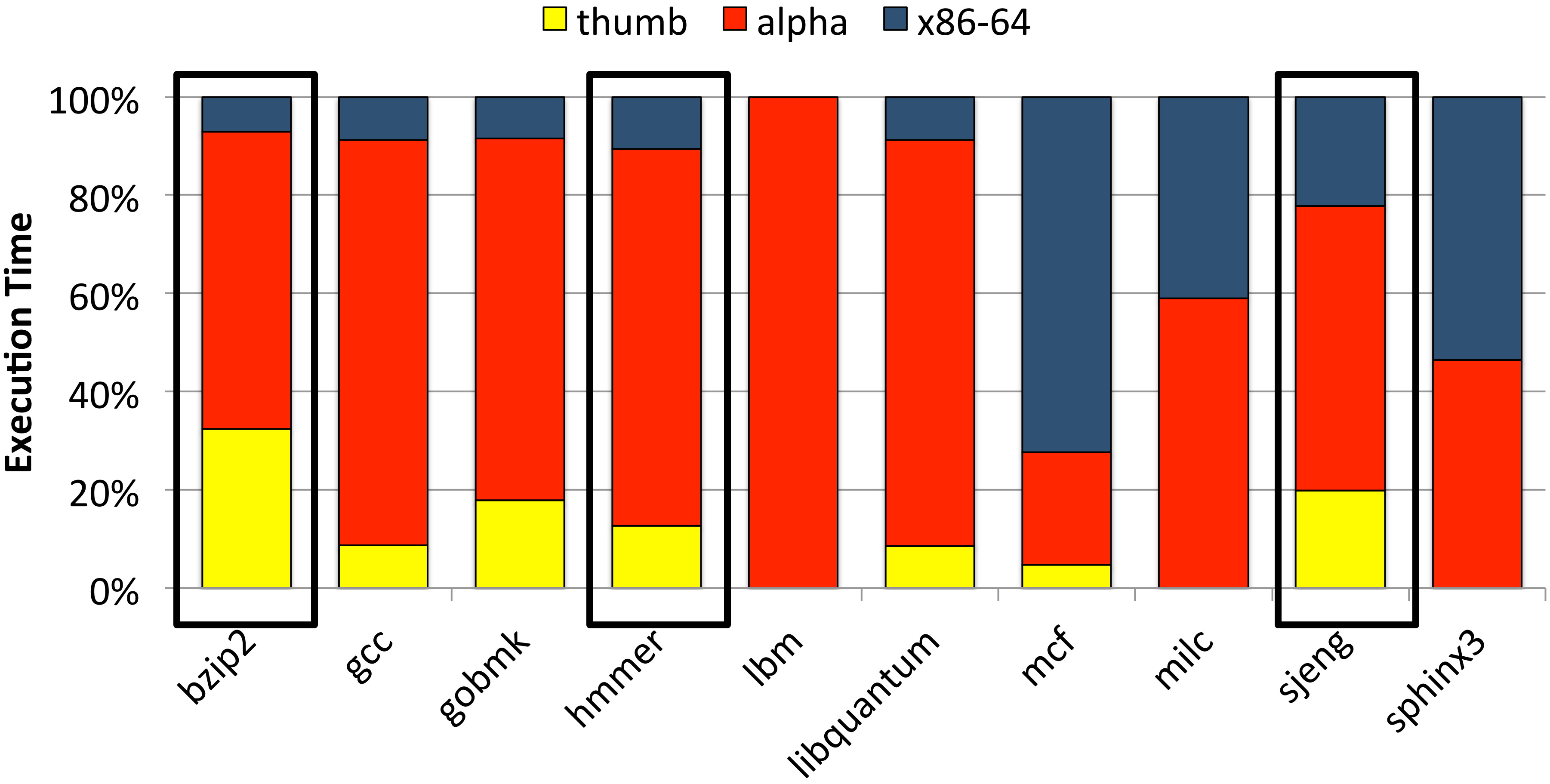
Design Space Exploration

Inferences – ISA-affinity



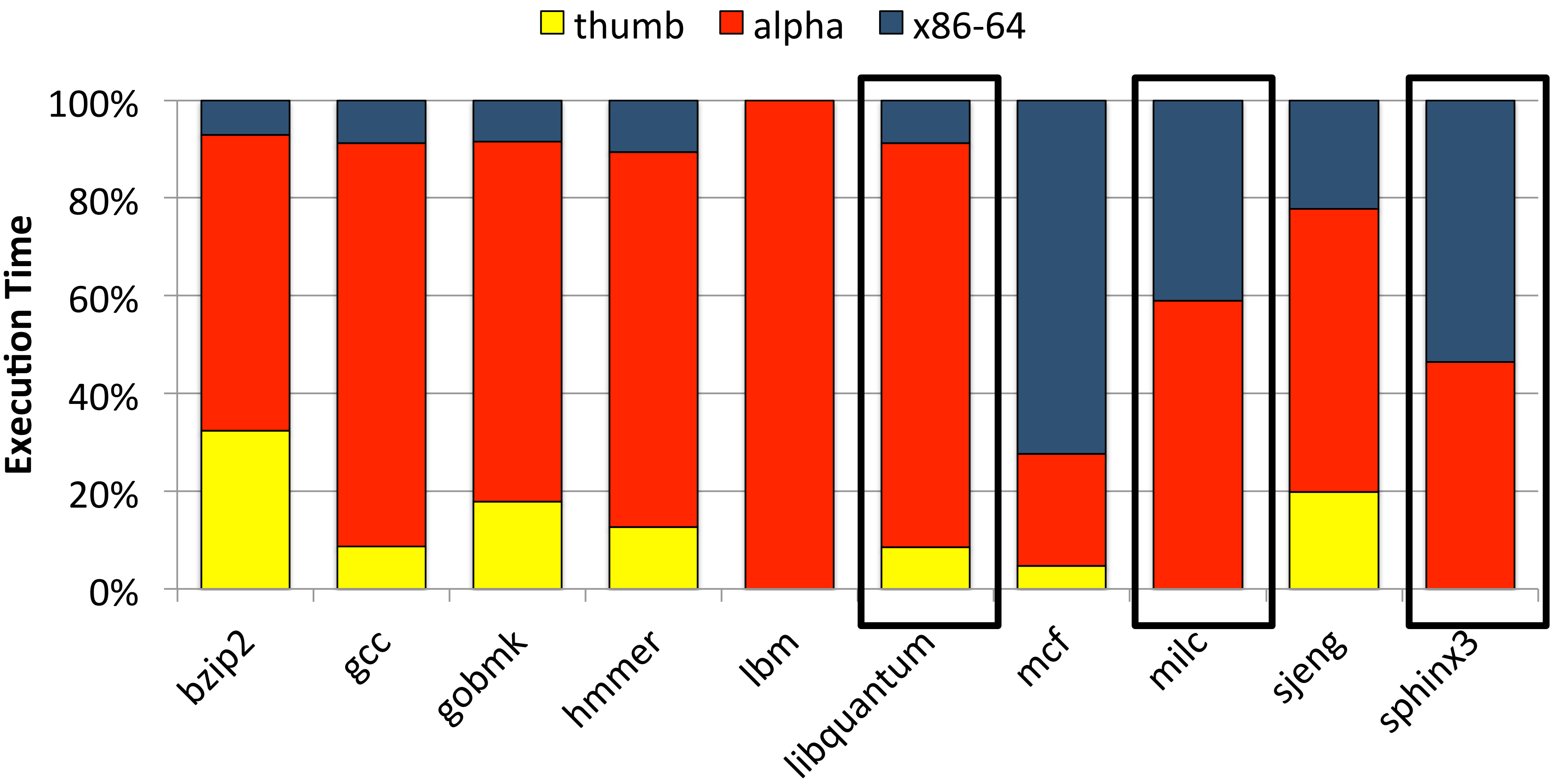
Design Space Exploration

Inferences – ISA-affinity



Design Space Exploration

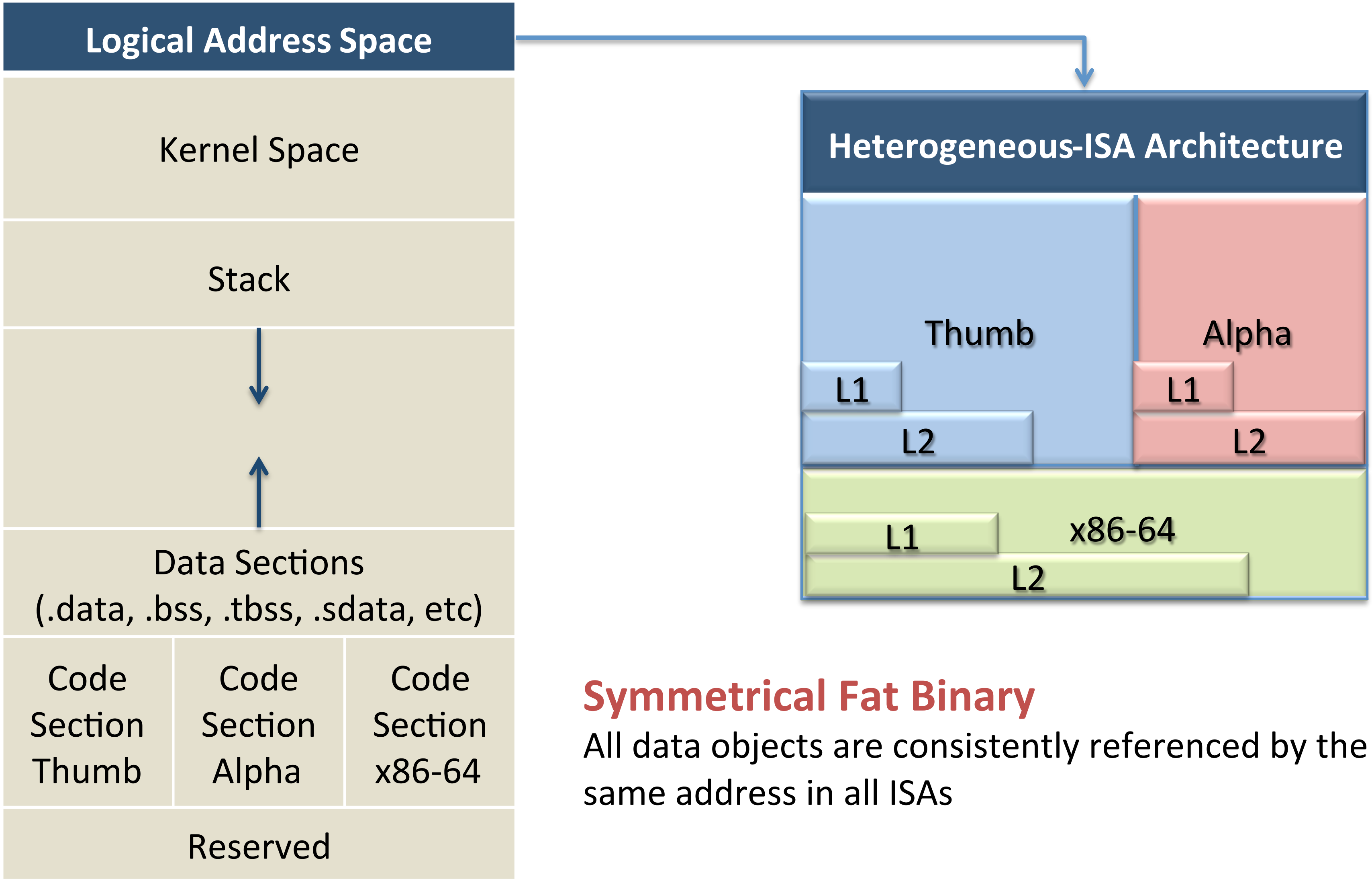
Inferences – ISA-affinity



Outline

- Motivation
- ISA diversity
- Design Space Exploration
  - Navigation and Optimization
  - Inference: ISA-microarchitecture co-design
  - Inference: ISA-affinity
- Compilation and Runtime Strategy
- Key Points

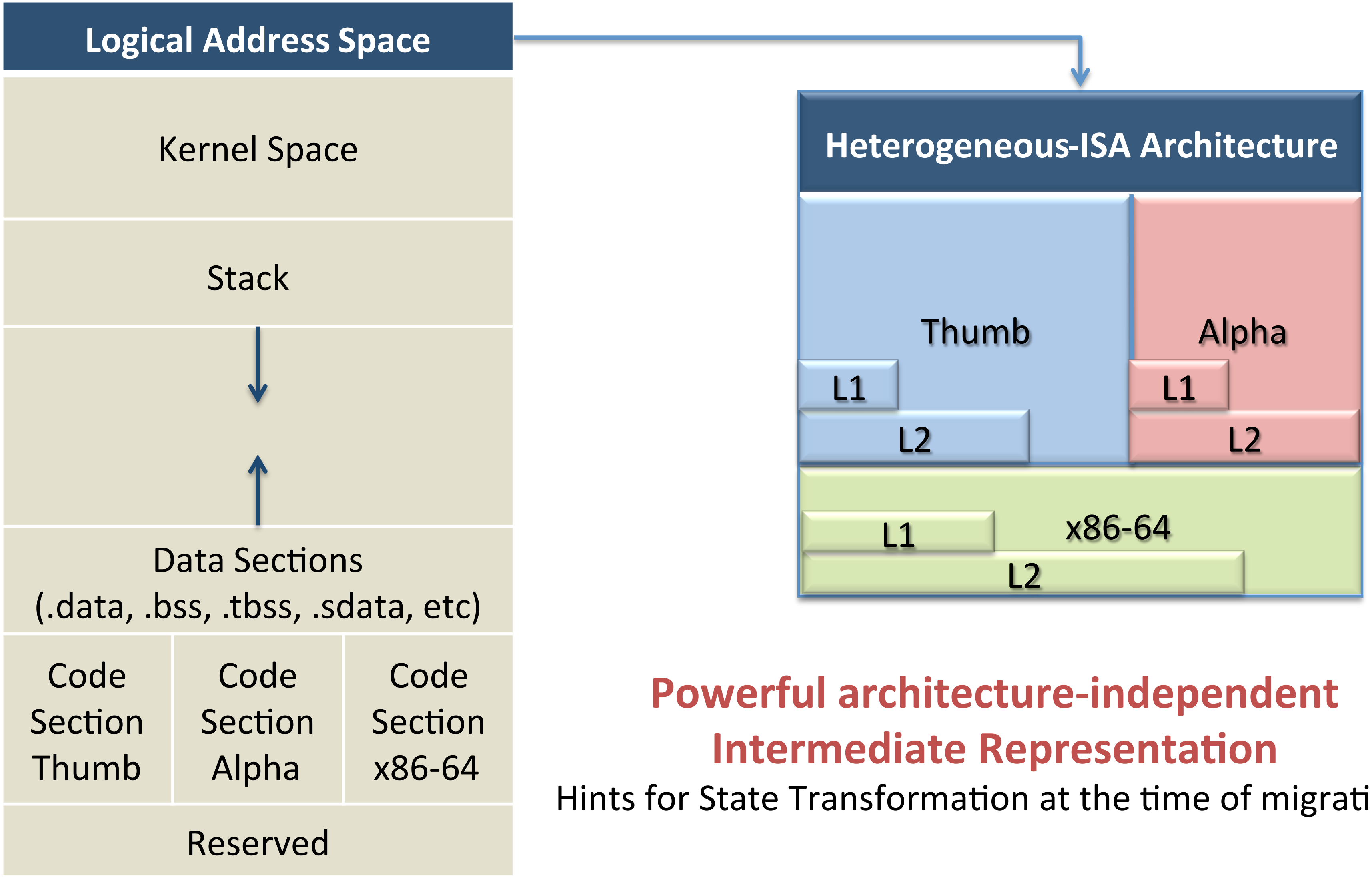
Compilation and Runtime Strategy



Symmetrical Fat Binary

All data objects are consistently referenced by the same address in all ISAs

Compilation and Runtime Strategy



Logical Address Space

Kernel Space

Stack

Data Sections (.data, .bss, .tbss, .sdata, etc)

Code Section Thumb	Code Section Alpha	Code Section x86-64

Reserved

Heterogeneous-ISA Architecture

Powerful architecture-independent Intermediate Representation

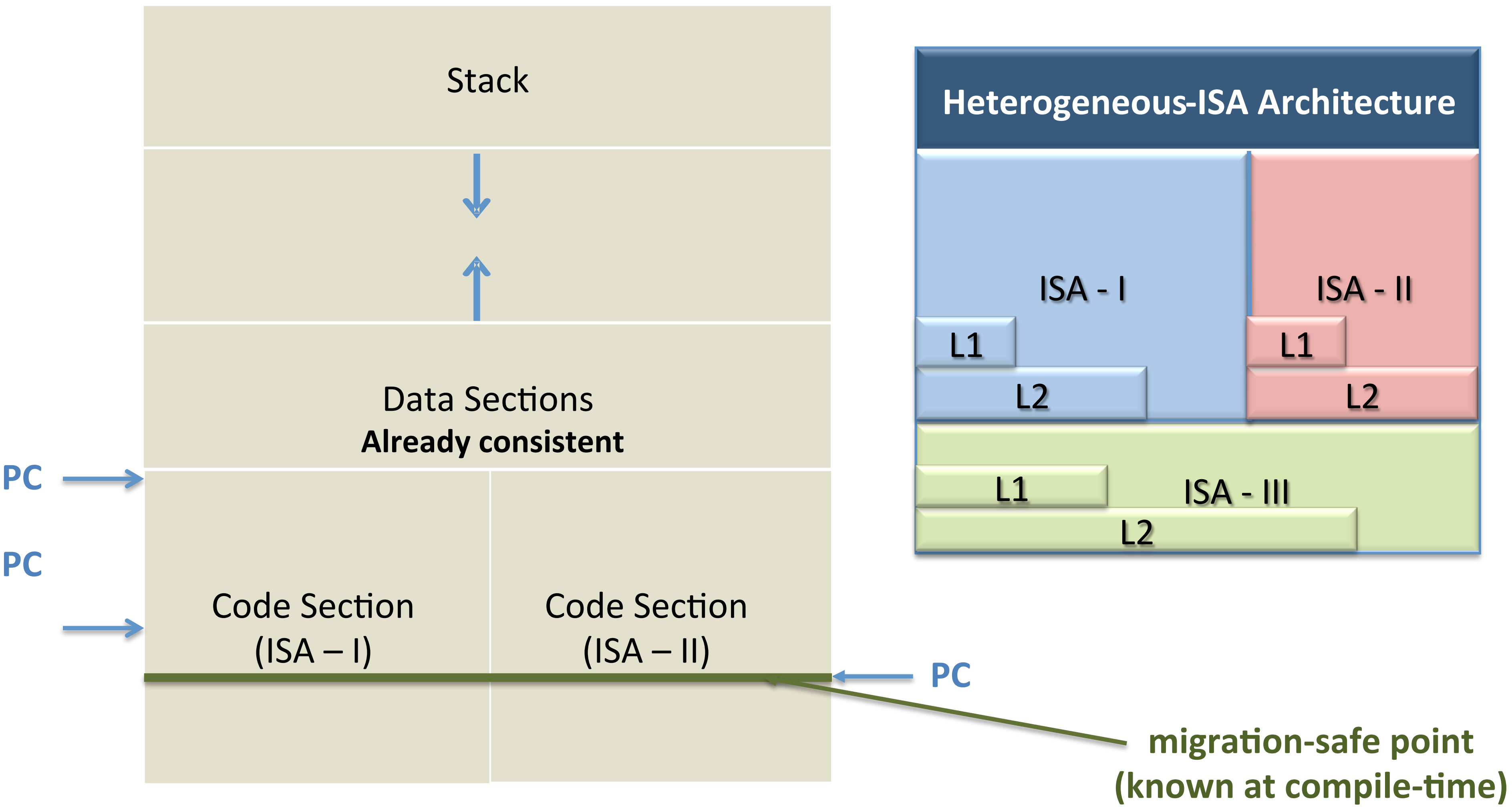
Hints for State Transformation at the time of migration

Overview of migration

Program state transformation on core -II

Migration requested !!

Switch to core - II !!

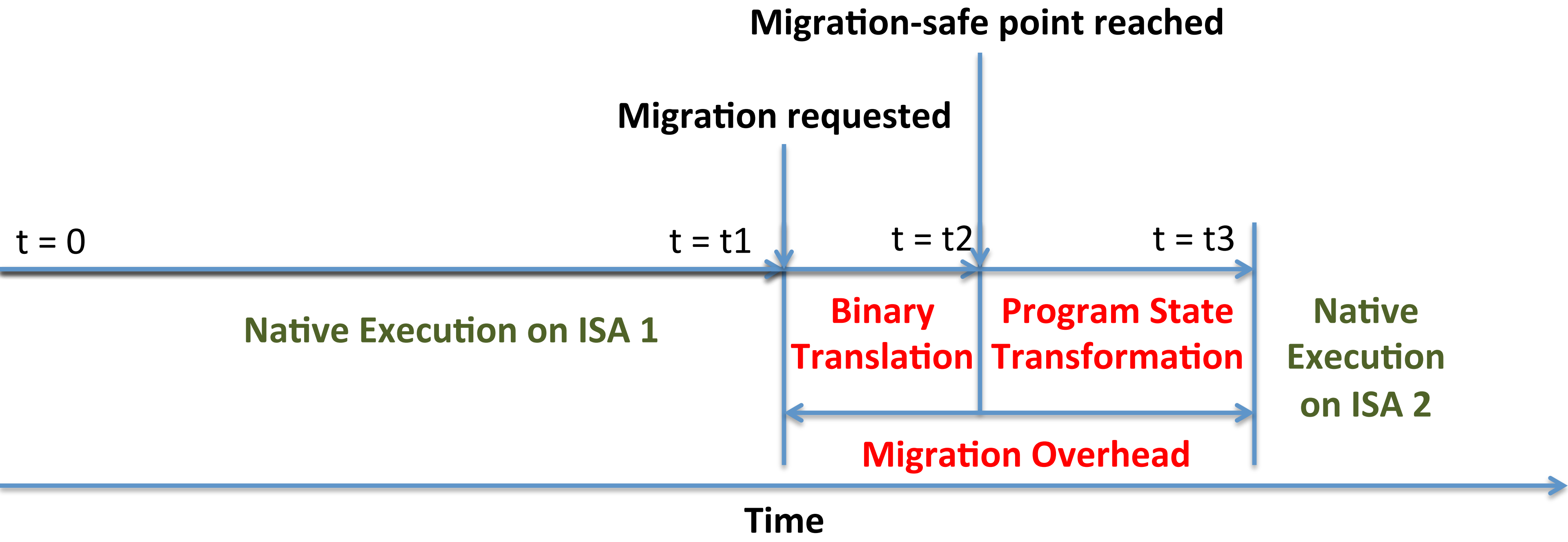


Native execution on core - I

Binary translation on core - II

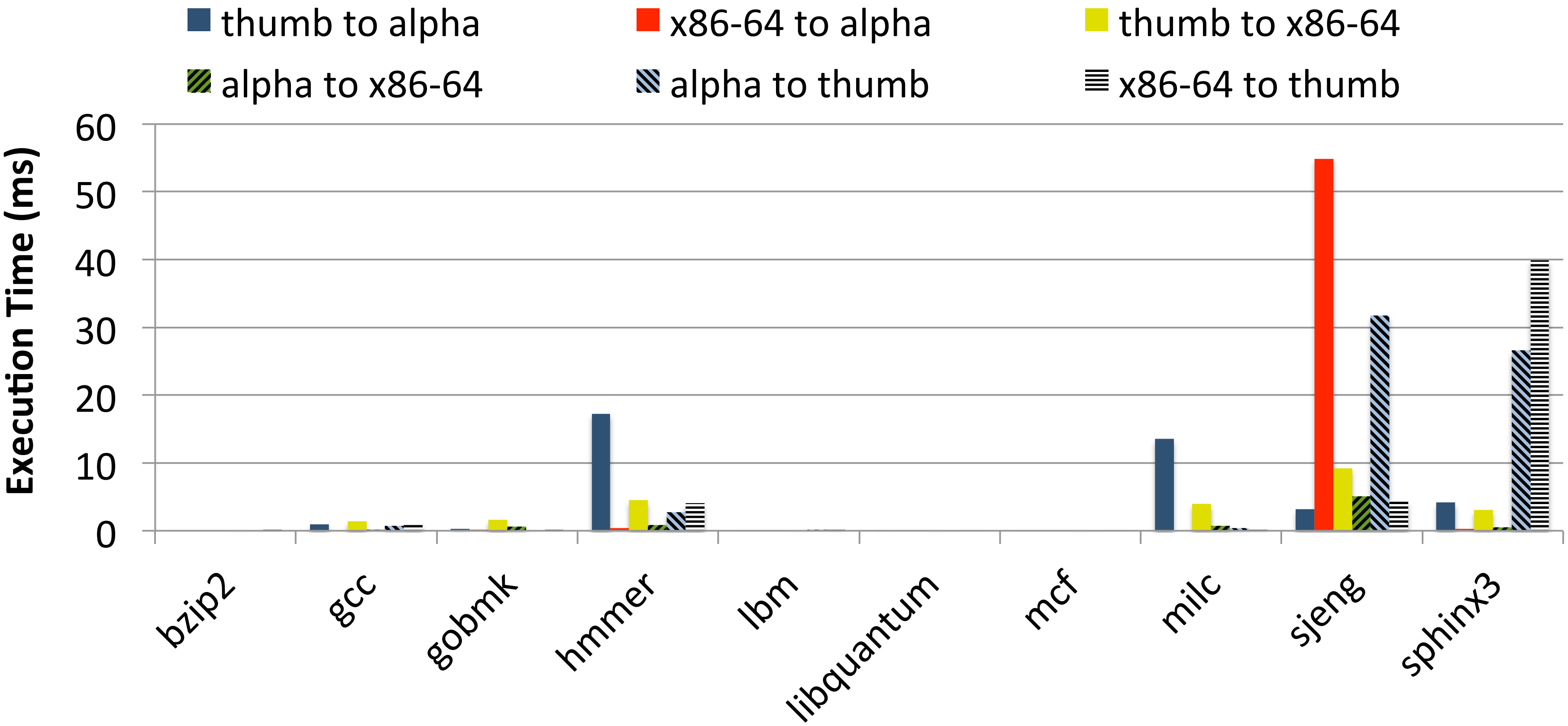
Native execution on core - II

Execution Migration Timeline


Migration-safe point reached
Migration requested
t = 0
t = t1
t = t2
t = t3
Native Execution on ISA 1
Binary Translation
Program State Transformation
Native Execution on ISA 2
Migration Overhead
Time

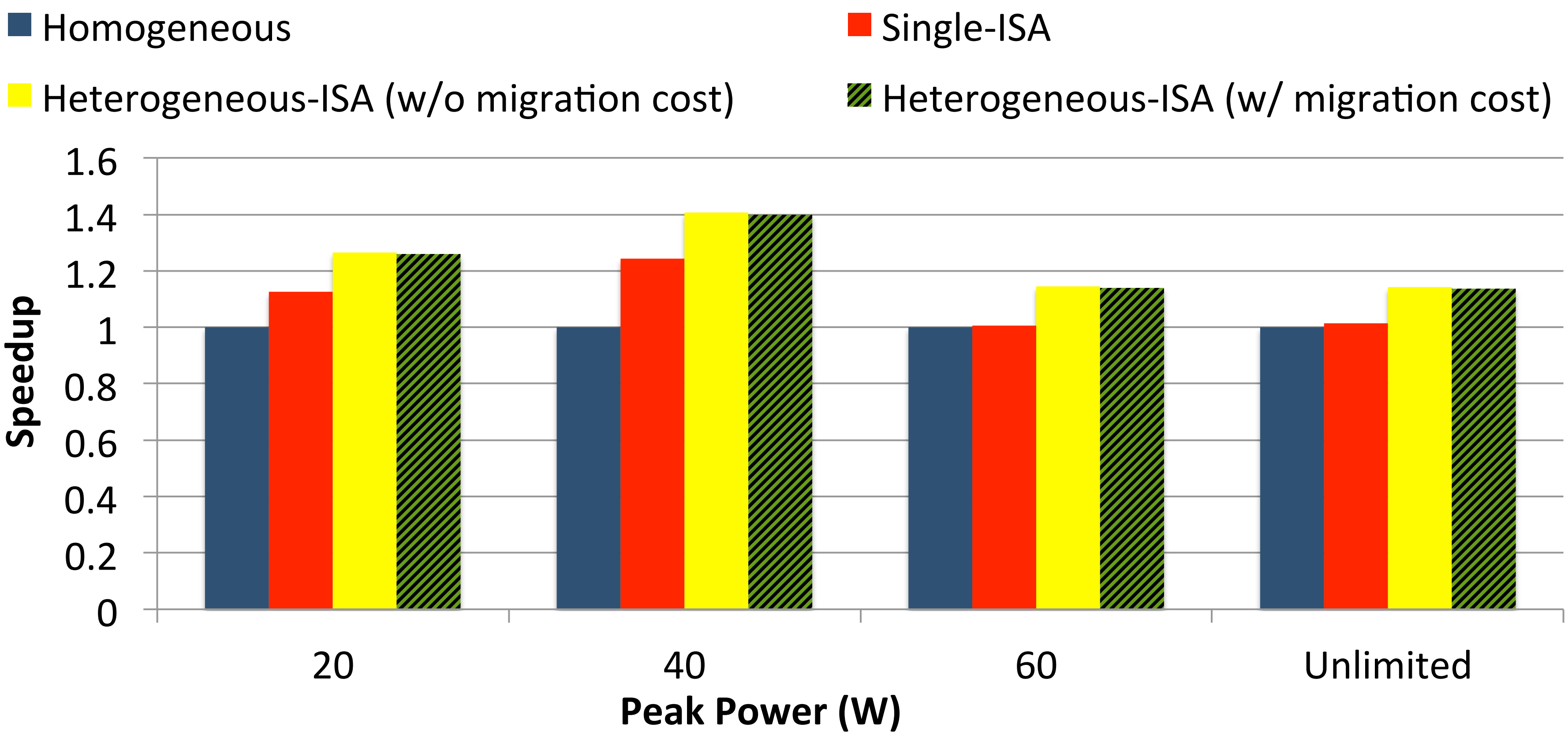
Migration Overhead = Binary Translation + Program State Transformation

Migration Cost – arbitrary migrations



- Average migration cost: 4 milliseconds
- Binary Translation time dominates the migration cost

Speedup accounting for Migration Cost



- Performance degradation: 0.4-0.7%
- Overall speedup due to migration: 11%

Key Points

- Heterogeneous-ISA CMP can outperform the best Single-ISA heterogeneous CMP by as much as 21% or provide 23% energy savings and 32% reduction in EDP.
- ISA-microarchitecture co-design is critical. There is significant synergy in combining hardware heterogeneity and ISA heterogeneity.
- Where hardware heterogeneity alone cannot provide any benefits, ISA-affinity still continues to provide both performance and energy gains.

Thank You!