Exploring the Tradeoffs of Configurability and Heterogeneity in Multicore Embedded Systems

Tosiron Adegbija and Ann Gordon-Ross[+]

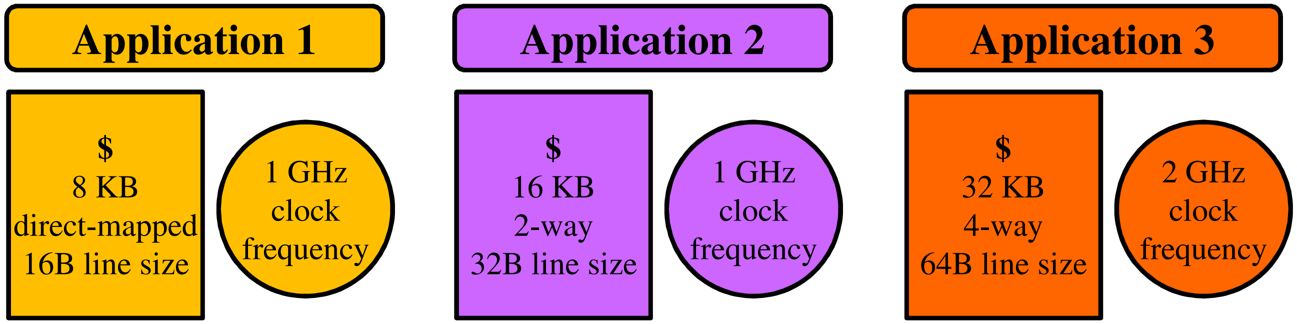
Department of Electrical and Computer Engineering
University of Florida, Gainesville, Florida, USA

[+] Also Affiliated with NSF Center for High-Performance Reconfigurable Computing

This work was supported by National Science Foundation (NSF) grant CNS-0953447

Introduction and Motivation

- Ubiquitous embedded systems have diverse design challenges
  - **Design goals**: cost, energy consumption, time-to-market, performance, etc.
  - **Design constraints**: energy, area, real time, cost, etc.
  - **Tunable parameters**: cache configuration, voltage, frequency, etc.
  - Varying per-application **parameter value requirements**
  - Specialize **configuration** to varying application characteristics (e.g., cache miss rates, instruction per cycle, etc.)
- Multicore architectures increasingly common in embedded systems
  - Alternatives to single-core architectures for achieving design goals
  - Significantly complicates design challenges



Application 1

$ 8 KB direct-mapped 16B line size

1 GHz clock frequency

Application 2

$ 16 KB 2-way 32B line size

1 GHz clock frequency

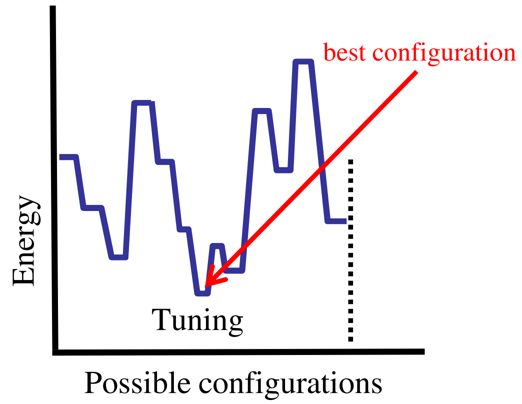
Application 3

$ 32 KB 4-way 64B line size

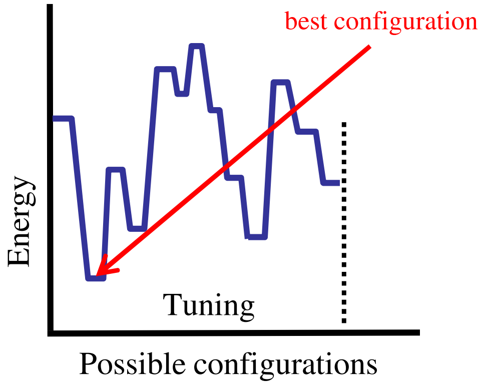
2 GHz clock frequency

Configuration Specialization

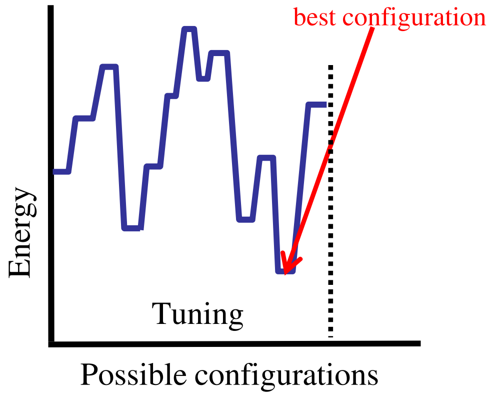
- Specialize system configuration to specific application requirements
  - Specialize for **optimization goals**: lowest energy, best performance, energy delay product (EDP), etc.
  - E.g., cache tuning saves up to 60% of energy on average
    - Balasubramonian’00, Zhang’03
- **Tuning** determines the best configuration for each executing application
  - Best/optimal configuration with respect to optimization goals
  - Tuning evaluates potential configurations to determine best configuration



Application 1



Application 2

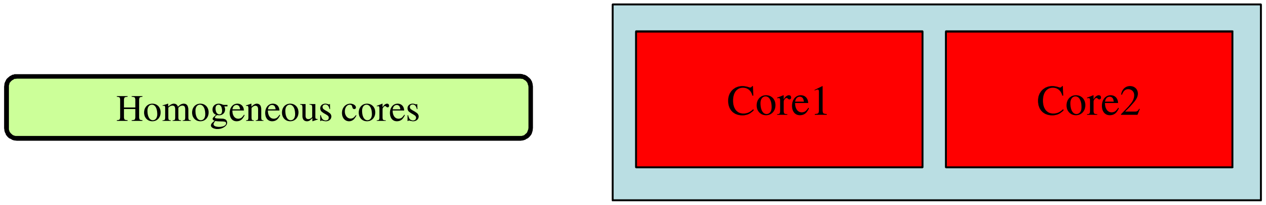


Application 3

Configurations must be specialized each application.

Homogenous Cores

- Traditional homogeneous cores
  - Identical configurations
  - Severely inhibits specialization

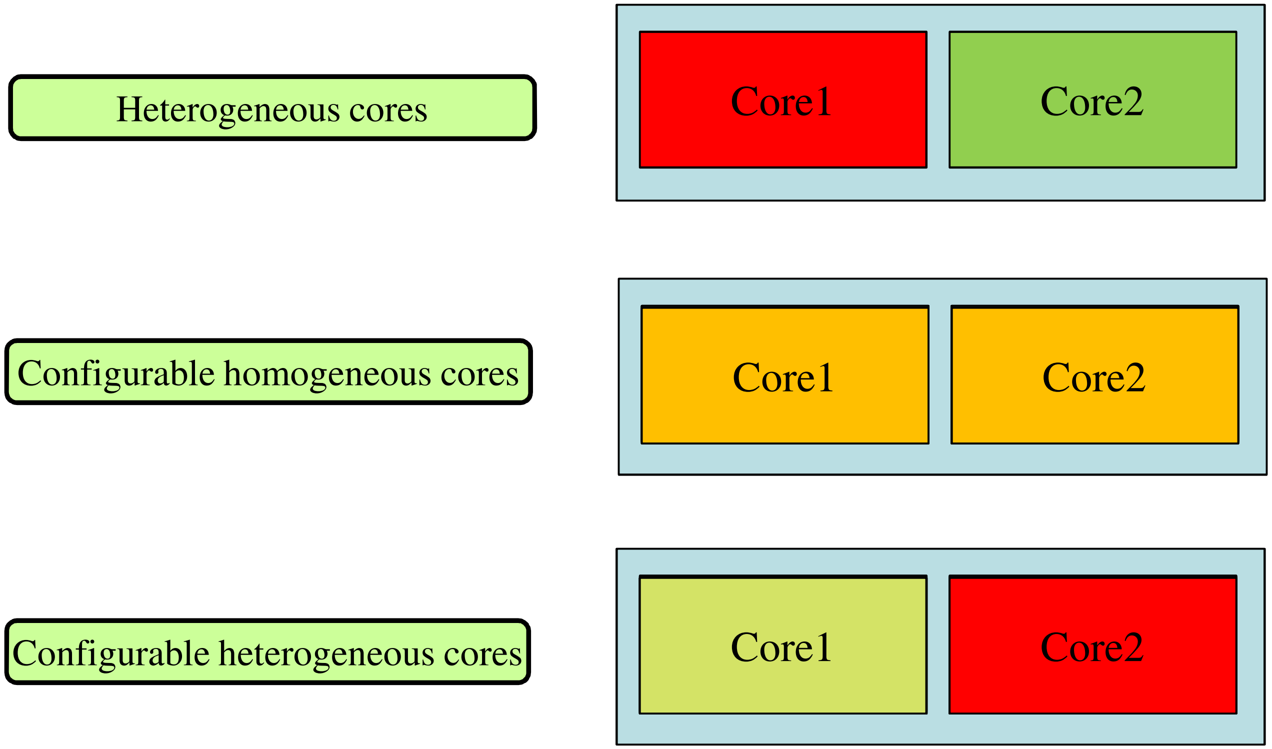


Different cores with identical configurations
Remains the same throughout system lifetime

- Previous work showed that specialization has significant impact on energy consumption
  - Limiting energy consumption is critical in embedded system
  - Cache and core frequency are key energy components
    - Our work focuses on cache and core frequency specialization

What are the methods for achieving specialization?

Specialization Methods



Heterogeneous cores

Core1

Core2

Different cores with different configurations
Remains same throughout system lifetime

Configurable homogeneous cores

Core1

Core2

Different cores with same configurations
Cores are tuned simultaneously
Configurations change dynamically

Configurable heterogeneous cores

Core1

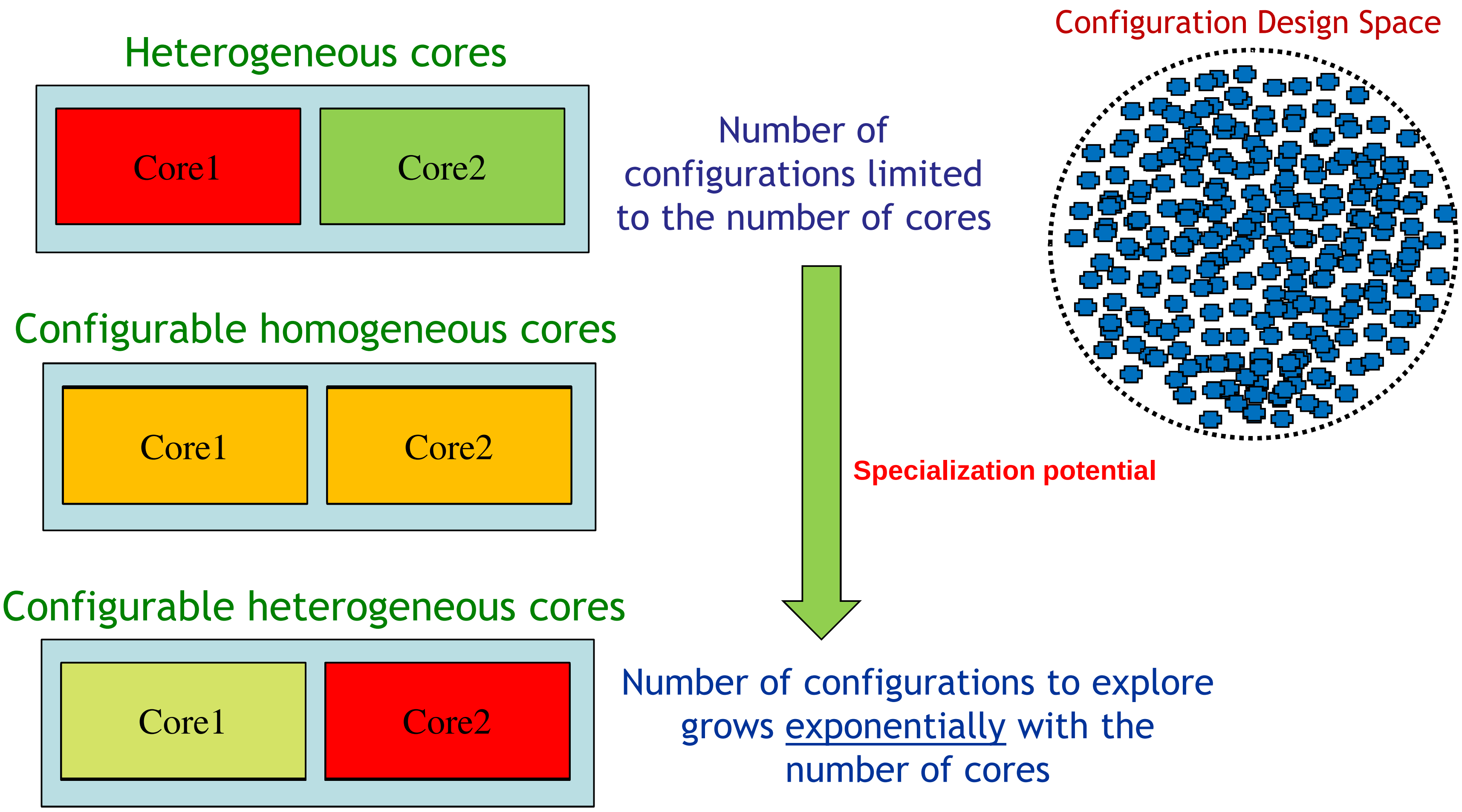
Core2

Different cores with different configurations,
Cores are tuned independently
Configurations change dynamically

Different methods have different design challenges and architecture options

Which specialization methods should designers use?

Design Challenges – Large Design Space



Heterogeneous cores

Core1 Core2

Number of configurations limited to the number of cores

Configuration Design Space

Configurable homogeneous cores

Core1 Core2

Specialization potential

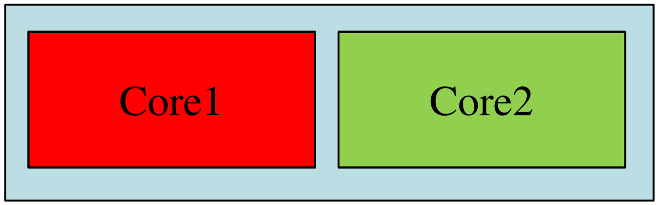
Configurable heterogeneous cores

Core1 Core2

Number of configurations to explore grows exponentially with the number of cores

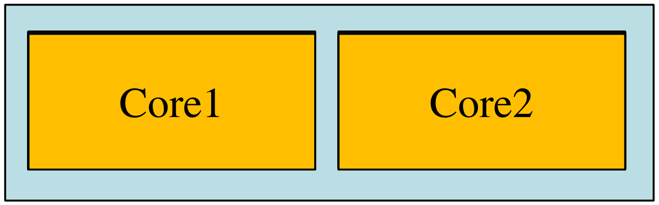
Design Challenges – Large Design Space

Heterogeneous cores



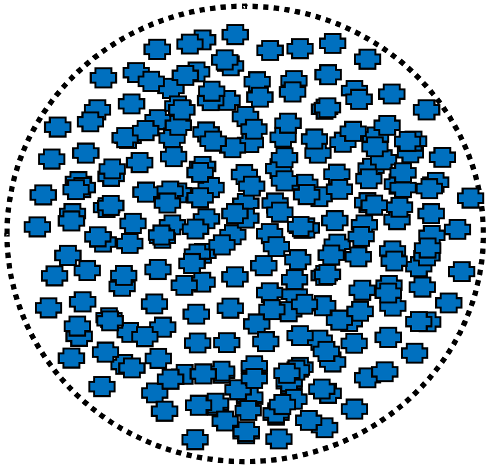
Scheduling applications to the best core

Configurable homogeneous cores

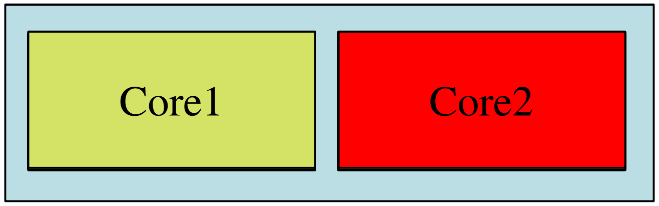


Determining the best configuration

Configuration design space



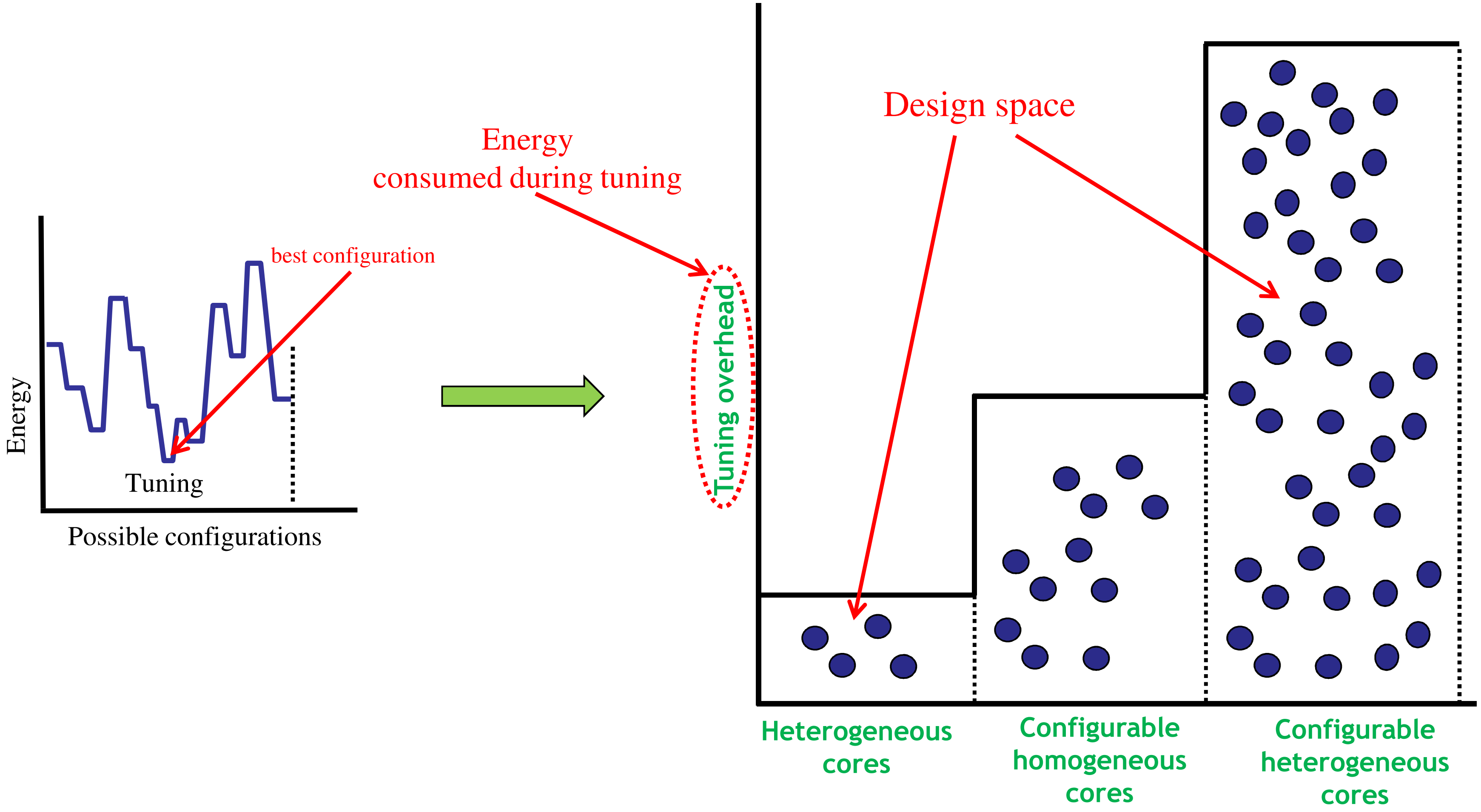
Configurable heterogeneous cores



Scheduling to the best core **AND** determining the best configuration

Using a sub-optimal schedule or configuration wastes energy!

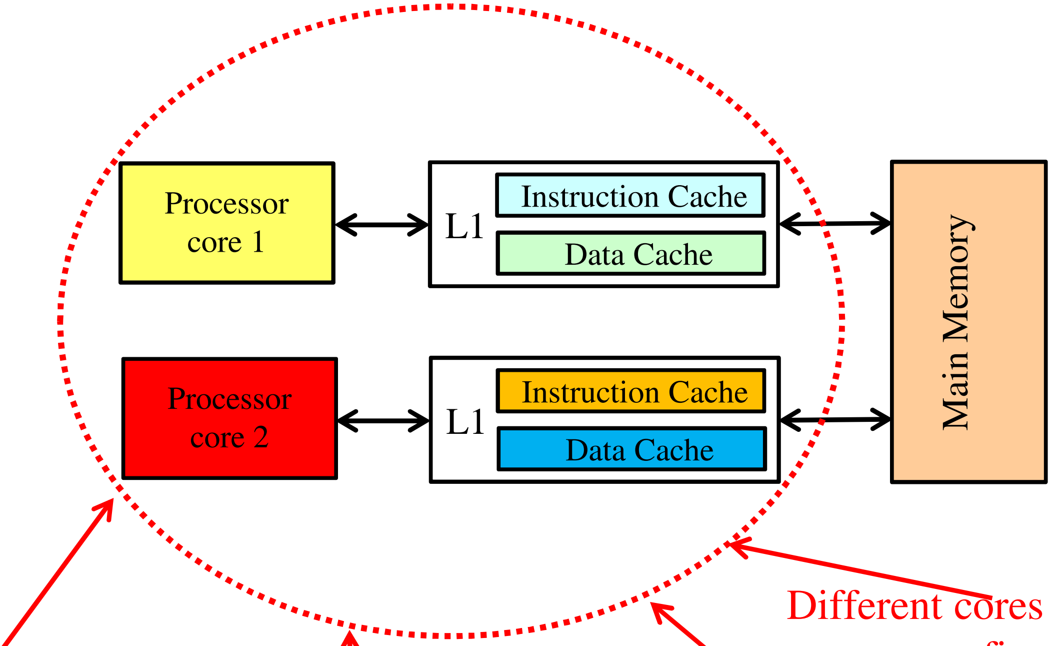
Design Challenges – Limiting Tuning Overhead



Tuning overhead typically increases with specialization options

Design Challenges

Heterogeneous Core Architectures



Different cores with different configurations

How disparate should the configurations be?

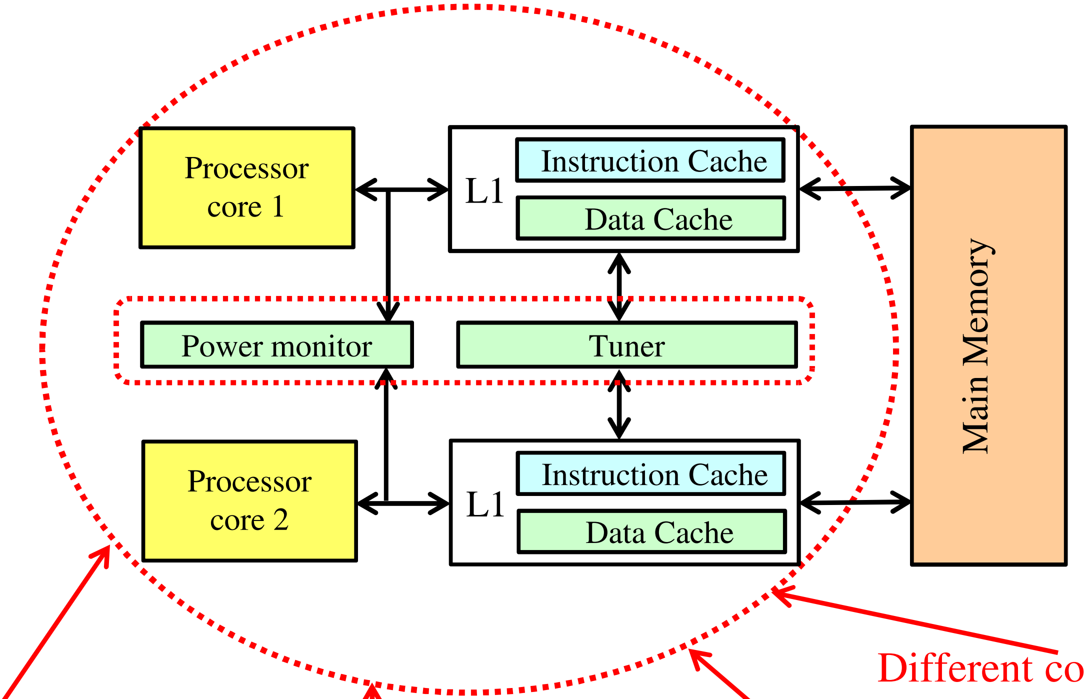
Choosing the best core configurations

E.g., core frequency, cache configurations, issue queue, reorder buffer, etc.

Cores should be suitable for a variety of applications. Requires a priori analysis

Design Challenges

Configurable Homogenous Core Architectures



Configurability of the cores/design space

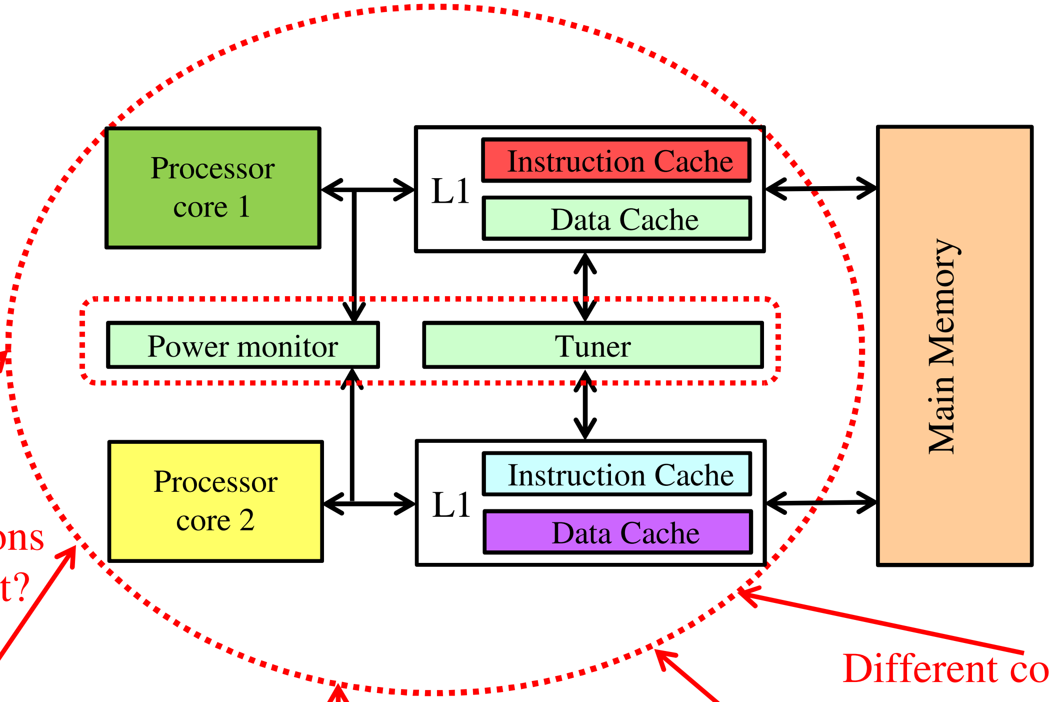
When should the configurations change during execution?

Different cores with *identical* configurations that change during execution

Requires tuning hardware (e.g., *power monitor* to measure power, and *tuner* to determine best configuration and change configurations

Design Challenges

Configurable Heterogeneous Core Architectures



Which configurations should be different?

Configurability of the cores/design space

When should the configurations change during execution?

Different cores with ***different*** configurations that change during execution

Requires tuning hardware (e.g., *power monitor* to measure power, and *tuner* to determine best configuration and change configurations

Design Challenges - Summary

- Heterogeneous cores
  - Which configurations should be different?
    - How different should the configurations be?
  - How to determine the different configurations?
    - Requires significant design time a priori analysis
- Configurable homogeneous cores
  - Imposes hardware overhead (e.g., tuner, power monitor, etc.)
  - Imposes tuning overhead
  - How often should the configuration change?
  - How configurable should the cores be?
- Configurable heterogeneous cores
  - Intersection of heterogeneous and configurable homogeneous core challenges
  - ***Significantly*** larger design space
- Our work quantifies these architectural tradeoffs and provides insight for design decisions

Experimental Setup

- Evaluated heterogeneity and configurability with respect to core frequency and cache configurations
  - Significant impact on system's overall energy
    - Nacul '04
- Energy delay product (EDP) as evaluation metric
  - EDP = $\mathit{core\_power} * \mathit{running\_time}^2$

    $= \mathit{core\_power} * (\mathit{total\_application\_cycles/system\_frequency})^2$
  - *Core_power:* cache and core's components (e.g., network interface units (NIU), peripheral component interconnect (PCI) controllers, etc.)
- McPAT calculated power consumption
- 24 multi-programmed workloads from EEMBC and Mediabench benchmark suites

Experimental Setup

- Modeled configurable/heterogeneous cores using GEM5
  - Modeled dual-core systems common in modern-day embedded systems
    - Modified GEM5 to simulate heterogeneous cores

Dual-core systems and configuration

System	Cache size	Associativity	Line size	Clock frequency
Homogeneous	32 Kbyte	4 way	64 byte	2 GHz
Configurable	16 – 32 Kbyte	1 – 4 way	16 – 64 byte	1 – 2 GHz
Heterogeneous-1	16/32 Kbyte	4 way	64 byte	1/2 GHz
Heterogeneous-2	8/16 Kbyte	4 way	64 byte	800 MHz/1 GHz
Heterogeneous-3	8/32 Kbyte	4 way	64 byte	800 MHz/2 GHz

Configuration selection options with
no extensive design time a priori analysis

Best average configuration
for all workloads after extensive
design time a priori analysis

Experimental Setup

Experimental test scenarios	
Name	**Core descriptions**
Test scenario 1	Naively-scheduled Heterogeneous-1
Test scenario 2	Optimally-scheduled Heterogeneous-1
Test scenario 3	Configurable homogeneous
Test scenario 4	Configurable heterogeneous

Highest EDP schedule (worst-case EDP)

Lowest EDP schedule

Used exhaustive search to determine best configurations

Results - Homogenous Core System

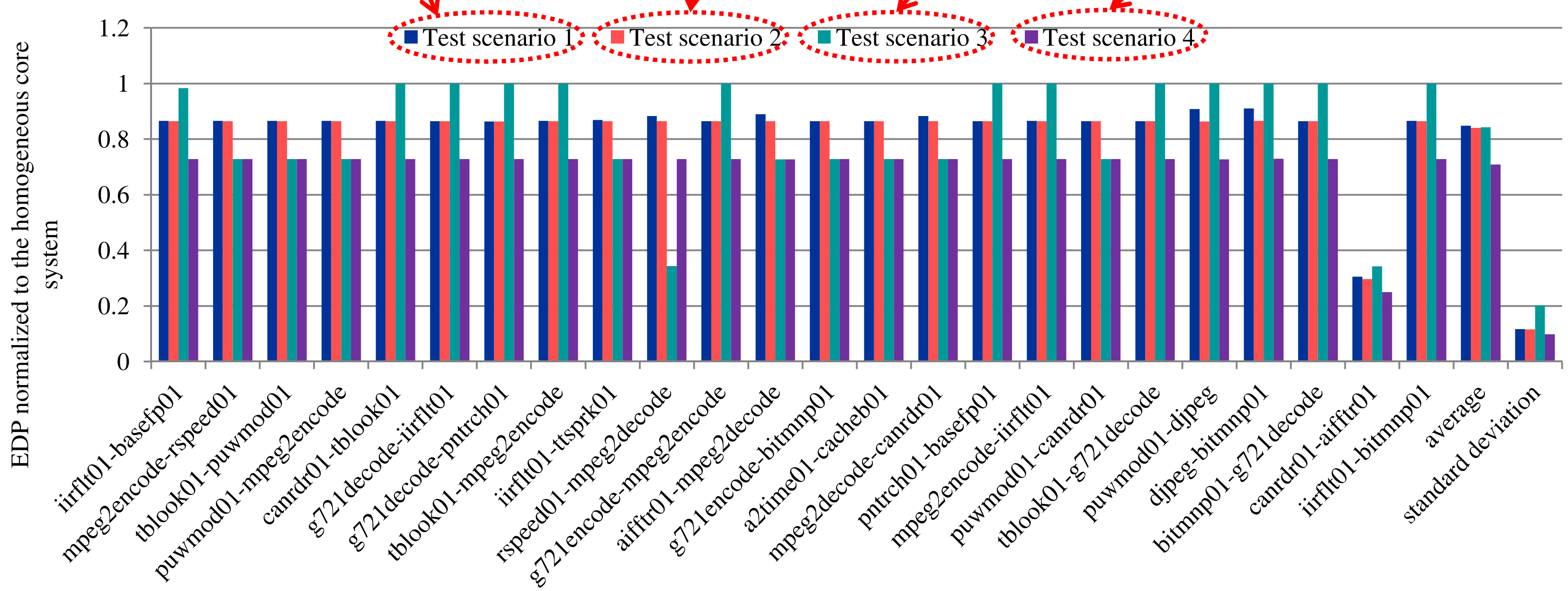
Configurable	16 – 32 Kbyte	1 – 4 way	16 – 64 byte	1 – 2 GHz
Heterogeneous-1	16/32 Kbyte	4 way	64 byte	1/2 GHz

Optimally-scheduled Heterogeneous-1: **16%** EDP savings

Configurable heterogeneous: **29%** EDP savings

Naively-scheduled Heterogeneous-1: **15%** EDP savings

Configurable homogeneous: **16%** EDP savings



Results

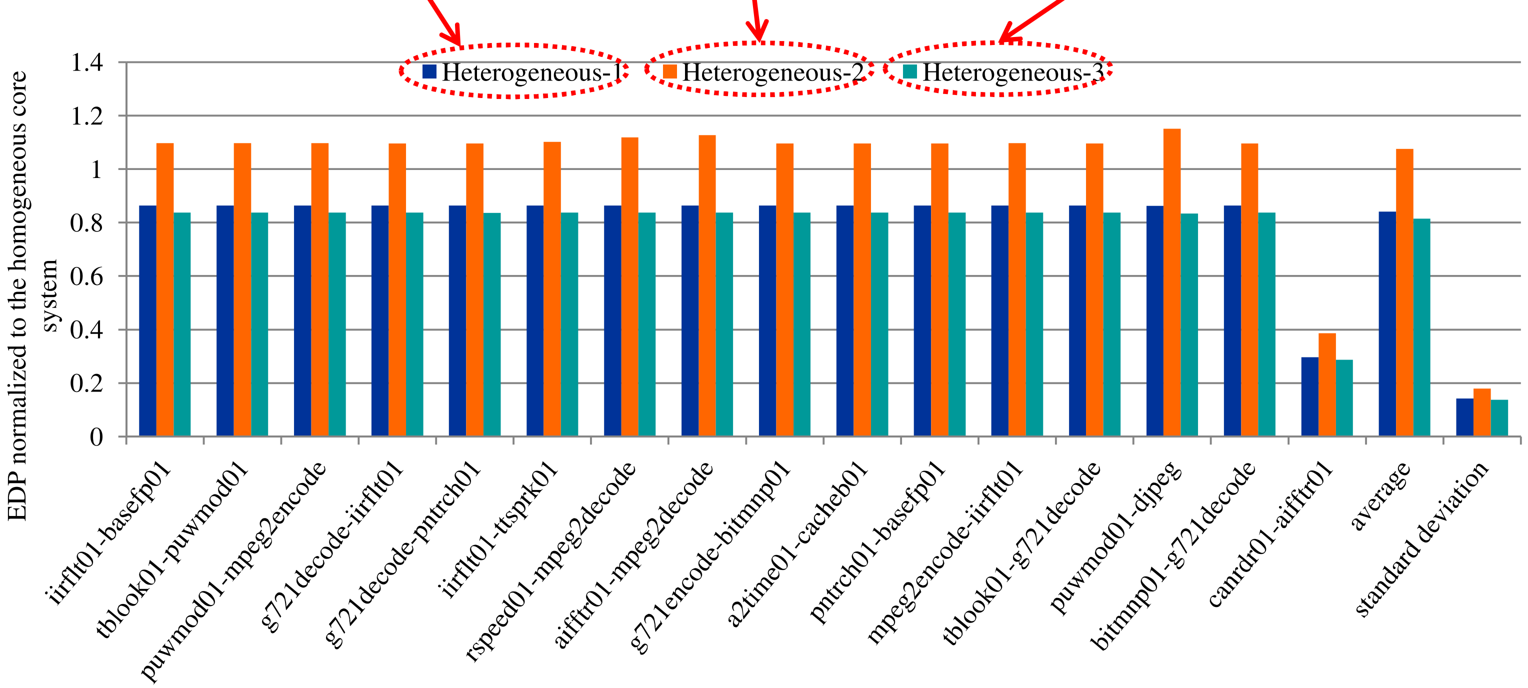
- Optimally-scheduled Heterogeneous-1, -2, and -3 compared to homogeneous core

Heterogeneous-1	16/32 Kbyte	4 way	64 byte	1/2 GHz
Heterogeneous-2	8/16 Kbyte	4 way	64 byte	800 MHz/1 GHz
Heterogeneous-3	8/32 Kbyte	4 way	64 byte	800 MHz/2 GHz

Heterogeneous-1: **16%** EDP savings

Heterogeneous-2: **7%** EDP *increase*

Heterogeneous-3: **19%** EDP savings



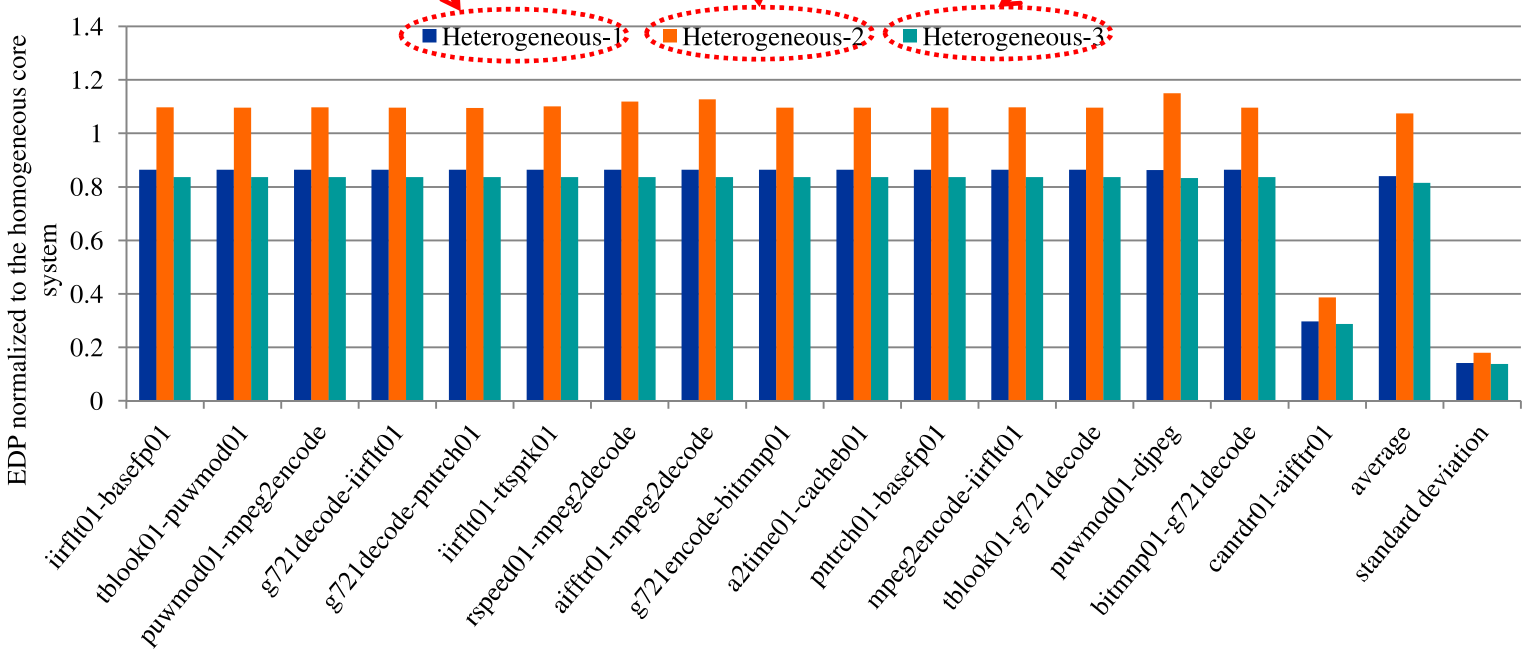
Results – Heterogeneous Core Specialization

Heterogeneous-1	16/32 Kbyte	4 way	64 byte	1/2 GHz
Heterogeneous-2	8/16 Kbyte	4 way	64 byte	800 MHz/1 GHz
Heterogeneous-3	8/32 Kbyte	4 way	64 byte	800 MHz/2 GHz

Heterogeneous-1: **16%** EDP savings

Heterogeneous-2: **7%** EDP *increase*

Heterogeneous-3: **19%** EDP savings



Increased core diversity with effective scheduling enhances benefits of heterogeneity!

Results – Configurable Core Specialization

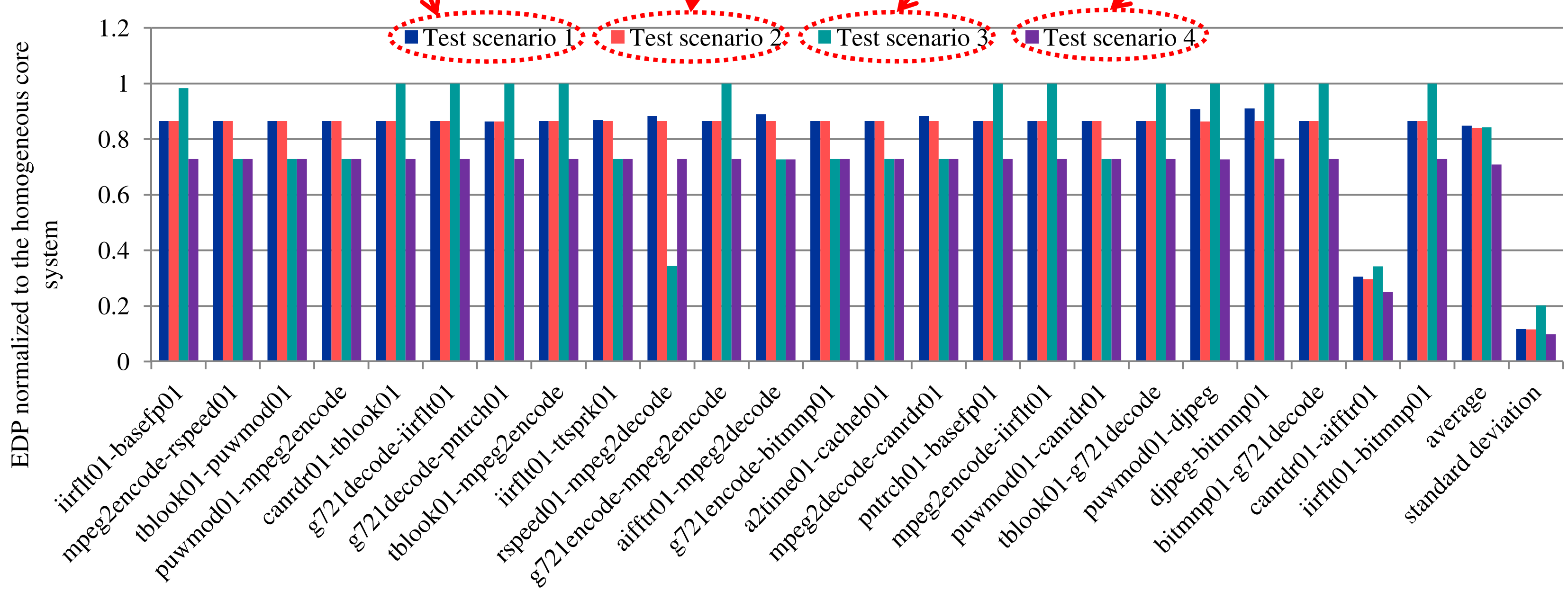
Configurable	16 – 32 Kbyte	1 – 4 way	16 – 64 byte	1 – 2 GHz
Heterogeneous-1	16/32 Kbyte	4 way	64 byte	1/2 GHz

Naively-scheduled Heterogeneous-1: **15%** EDP savings

Optimally-scheduled Heterogeneous-1: **16%** EDP savings

Configurable homogeneous: **16%** EDP savings

Configurable heterogeneous: **29%** EDP savings



Independently tuned configurable heterogeneous cores achieves maximum EDP savings!

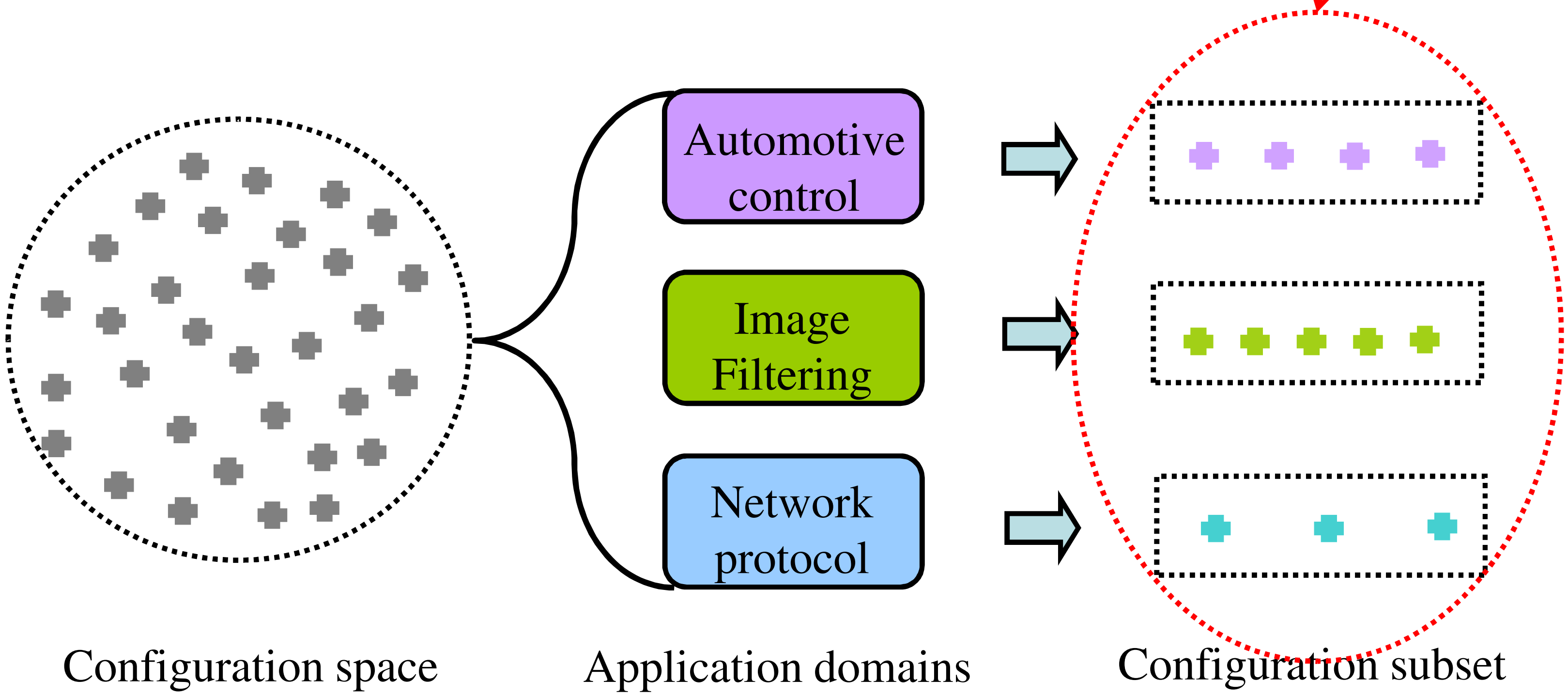
Conclusions

- Evaluated tradeoffs of heterogeneity and configurability in system specialization
  - Quantified EDP savings for heterogeneity, configurability, and configurable heterogeneity compared to homogeneous cores
  - Provided insights and guidelines for designers
    - Best EDP savings achieved with configurable heterogeneous cores
      - Configurable heterogeneous cores leverage benefits of heterogeneity and configurability
- Future work
  - Explore and evaluate the impact of reducing configurable heterogeneous cores’ design space by configuration subsetting

Future Work

- Configuration design space subsetting
  - Viana ’06

Tuning searches a significantly reduced design space!



Questions?