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

<table>
<thead>
<tr>
<th>Application 1</th>
<th>Application 2</th>
<th>Application 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>$ 8 KB direct-mapped 16B line size</td>
<td>$ 16 KB 2-way 32B line size</td>
<td>$ 32 KB 4-way 64B line size</td>
</tr>
<tr>
<td>1 GHz clock frequency</td>
<td>1 GHz clock frequency</td>
<td>2 GHz clock frequency</td>
</tr>
</tbody>
</table>
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

Configurations must be specialized each application.
Homogenous Cores

- Traditional homogeneous cores
  - Identical configurations
  - Severely inhibits specialization

- 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
  - Different cores with different configurations
  - Remains same throughout system lifetime

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

- Configurable heterogeneous cores
  - 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

Configurable homogeneous cores

Configurable heterogeneous cores

Configuration Design Space

Number of configurations limited to the number of cores

Specialization potential

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

- **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
Choosing the best core configurations
Cores should be suitable for a variety of applications. Requires a priori analysis

How disparate should the configurations be?
E.g., core frequency, cache configurations, issue queue, reorder buffer, etc.
Design Challenges
Configurable Homogenous Core Architectures

Different cores with *identical* configurations that change during execution

When should the configurations change during execution?

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

Configurability of the cores/design space
Design Challenges
Configurable Heterogeneous Core Architectures

Different cores with different configurations that change during execution

When should the configurations change during execution?

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

Configurability of the cores/design space

Which configurations should be different?
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 = \( \text{core_power} \times \text{running_time}^2 \)
    - \( \text{EDP} = \text{core_power} \times (\text{total_application_cycles} / \text{system_frequency})^2 \)
  - \text{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

<table>
<thead>
<tr>
<th>System</th>
<th>Cache size</th>
<th>Associativity</th>
<th>Line size</th>
<th>Clock frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Homogeneous</td>
<td>32 Kbyte</td>
<td>4 way</td>
<td>64 byte</td>
<td>2 GHz</td>
</tr>
<tr>
<td>Configurable</td>
<td>16 – 32 Kbyte</td>
<td>1 – 4 way</td>
<td>16 – 64 byte</td>
<td>1 – 2 GHz</td>
</tr>
<tr>
<td>Heterogeneous-1</td>
<td>16/32 Kbyte</td>
<td>4 way</td>
<td>64 byte</td>
<td>1/2 GHz</td>
</tr>
<tr>
<td>Heterogeneous-2</td>
<td>8/16 Kbyte</td>
<td>4 way</td>
<td>64 byte</td>
<td>800 MHz/1 GHz</td>
</tr>
<tr>
<td>Heterogeneous-3</td>
<td>8/32 Kbyte</td>
<td>4 way</td>
<td>64 byte</td>
<td>800 MHz/2 GHz</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Test scenario 1</th>
<th>Core descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test scenario 2</td>
<td>Optimally-scheduled Heterogeneous-1</td>
</tr>
<tr>
<td>Test scenario 3</td>
<td>Configurable homogeneous</td>
</tr>
<tr>
<td>Test scenario 4</td>
<td>Configurable heterogeneous</td>
</tr>
</tbody>
</table>

Used exhaustive search to determine best configurations

Highest EDP schedule (worst-case EDP)

Lowest EDP schedule

Experimental test scenarios
Results - Homogenous Core System

<table>
<thead>
<tr>
<th>Configurable</th>
<th>16 – 32 Kbyte</th>
<th>1 – 4 way</th>
<th>16 – 64 byte</th>
<th>1 – 2 GHz</th>
</tr>
</thead>
<tbody>
<tr>
<td>Heterogeneous-1</td>
<td>16/32 Kbyte</td>
<td>4 way</td>
<td>64 byte</td>
<td>1/2 GHz</td>
</tr>
</tbody>
</table>

- Naively-scheduled Heterogeneous-1: **15%** EDP savings
- Optimally-scheduled Heterogeneous-1: **16%** EDP savings
- Configurable homogeneous: **16%** EDP savings
- Configurable heterogeneous: **29%** EDP savings

![Graph showing EDP normalized to the homogeneous core system across various test scenarios.](image-url)
Results

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

<table>
<thead>
<tr>
<th></th>
<th>Size</th>
<th>Way</th>
<th>Cache Size</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Heterogeneous-1</td>
<td>16/32 Kbyte</td>
<td>4</td>
<td>64 byte</td>
<td>1/2 GHz</td>
</tr>
<tr>
<td>Heterogeneous-2</td>
<td>8/16 Kbyte</td>
<td>4</td>
<td>64 byte</td>
<td>800 MHz/1 GHz</td>
</tr>
<tr>
<td>Heterogeneous-3</td>
<td>8/32 Kbyte</td>
<td>4</td>
<td>64 byte</td>
<td>800 MHz/2 GHz</td>
</tr>
</tbody>
</table>

Heterogeneous-1: 16% EDP savings
Heterogeneous-2: 7% EDP increase
Heterogeneous-3: 19% EDP savings
## Results – Heterogeneous Core Specialization

<table>
<thead>
<tr>
<th>Heterogeneous-1</th>
<th>16/32 Kbyte</th>
<th>4 way</th>
<th>64 byte</th>
<th>1/2 GHz</th>
</tr>
</thead>
<tbody>
<tr>
<td>Heterogeneous-2</td>
<td>8/16 Kbyte</td>
<td>4 way</td>
<td>64 byte</td>
<td>800 MHz/1 GHz</td>
</tr>
<tr>
<td>Heterogeneous-3</td>
<td>8/32 Kbyte</td>
<td>4 way</td>
<td>64 byte</td>
<td>800 MHz/2 GHz</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Configurable</th>
<th>16 – 32 Kbyte</th>
<th>1 – 4 way</th>
<th>16 – 64 byte</th>
<th>1 – 2 GHz</th>
</tr>
</thead>
<tbody>
<tr>
<td>Heterogeneous-1</td>
<td>16/32 Kbyte</td>
<td>4 way</td>
<td>64 byte</td>
<td>1/2 GHz</td>
</tr>
</tbody>
</table>

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?