Soft Core Processor Microarchitecture

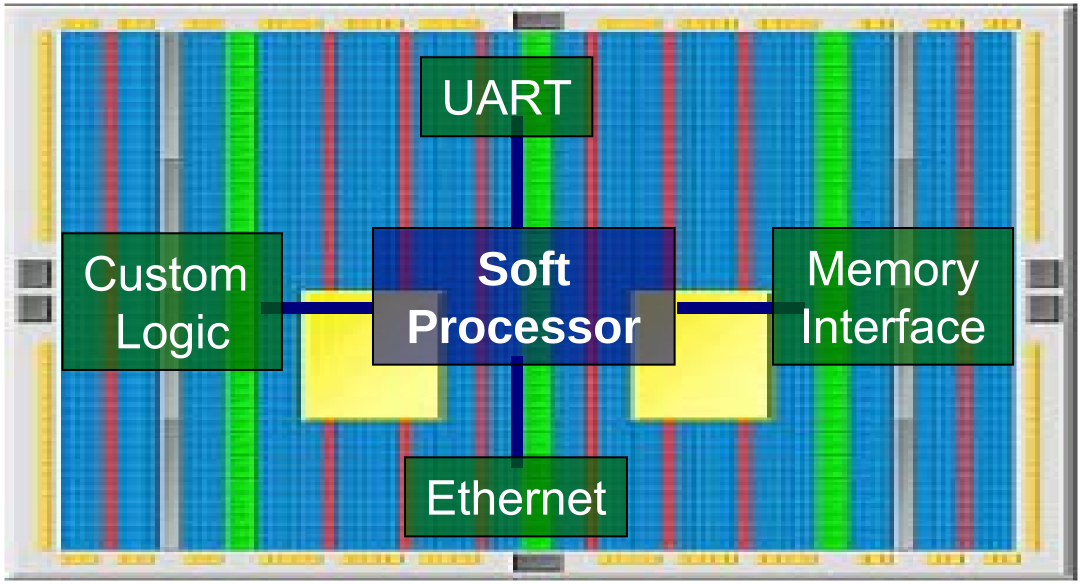
Dr. Shubhajit Roy Chowdhury,

Centre for VLSI and Embedded Systems Technology,

IIIT Hyderabad

Processors and FPGA Systems

- Processors lie at the “heart” of FPGA systems



- Performs coordination and even computation
  - Better processors => less hardware to design

Motivating Application-Specific Customizations of Soft Processors

1. FPGA Configurability
   - Can consider unlimited processor variants

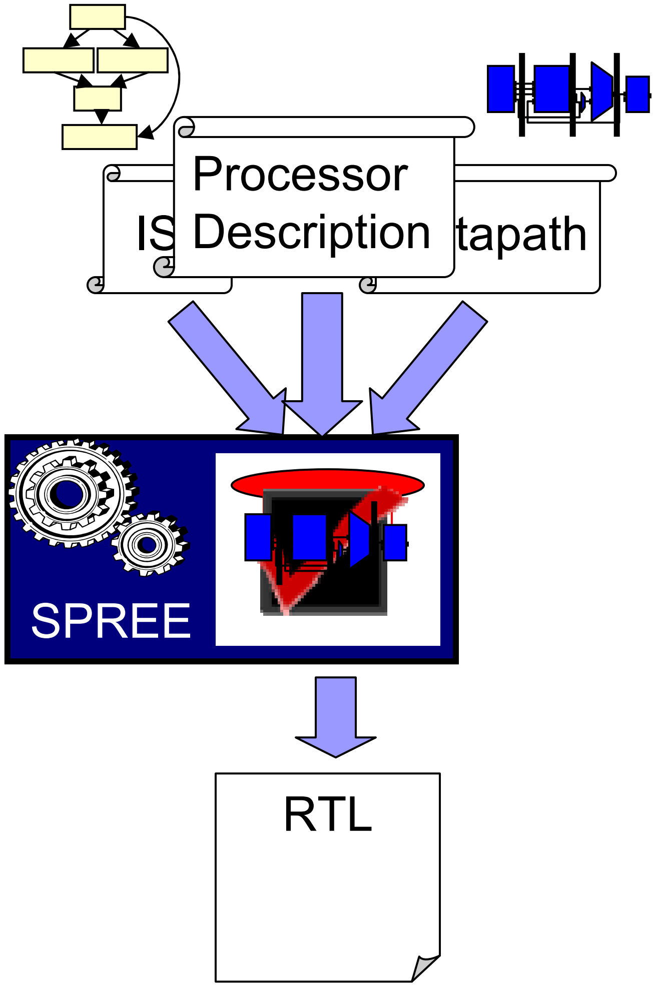
2. A soft processor might be used to run either:
   a) A single application
   b) A single class of applications
   c) Many applications, but can be reconfigured

3. Applications differ in architectural requirements
   - Can specialize architecture for each application

Research Goals

- To investigate
  1. The potential for “**Application-tuning**”
     - Tune processor microarchitecture to *favour* an application
     - Preserve general purpose functionality
  2. “**Instruction-set Subsetting**”
     - Sacrifice general purpose functionality
     - Eliminate hardware not required by application
  3. Combination of both methods

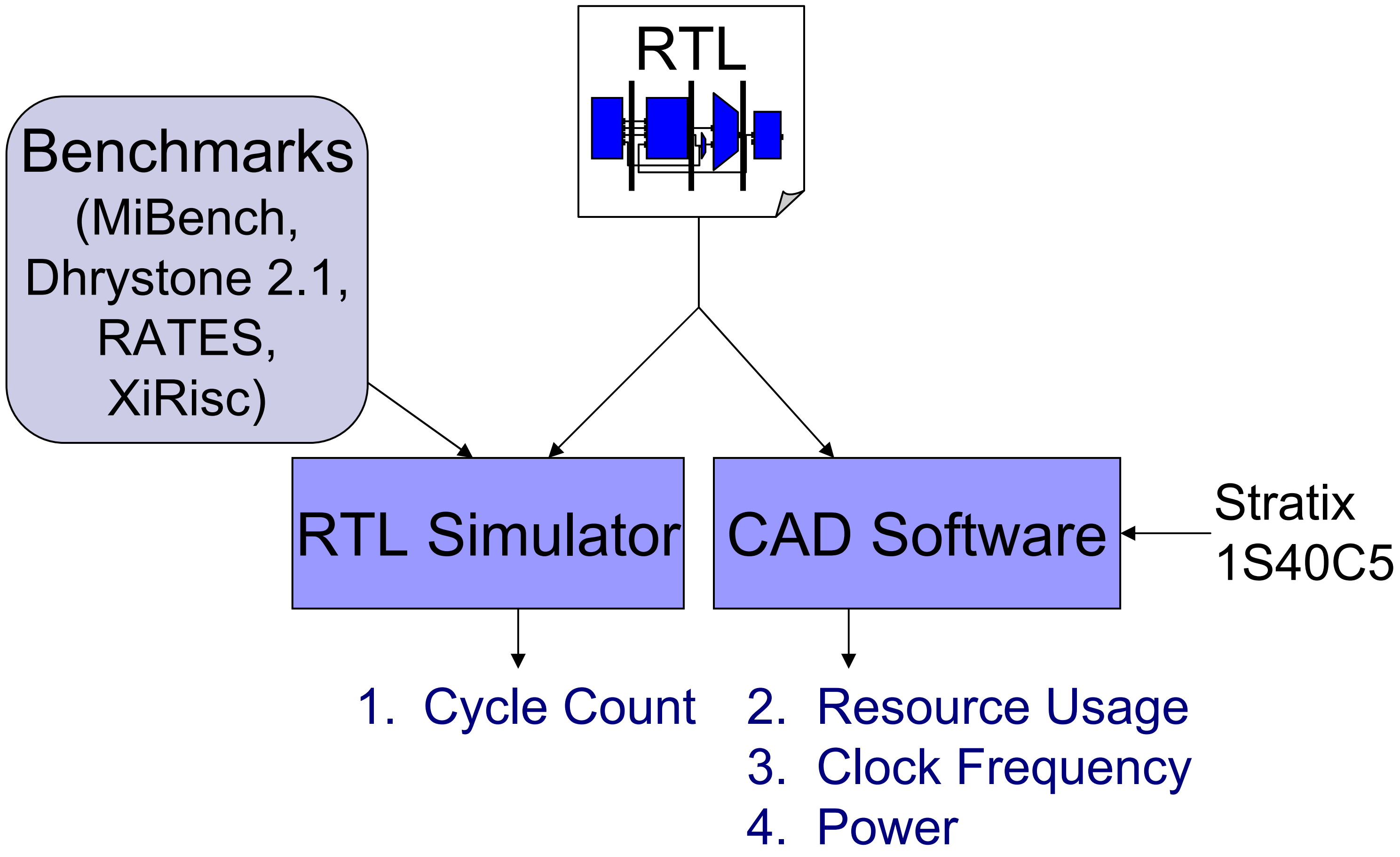
SPREE System
(Soft Processor Rapid Exploration Environment)



Developed at Tokyo University in 2006

- Input: Processor description
  - Made of hand-coded components
- SPREE System
  1. Verify ISA against datapath
  2. Datapath Instantiation
  3. Control Generation
     - Multi-cycle/variable-cycle FUs
     - Multiplexer select signals
     - Interlocking
     - Branch handling
- Output: Synthesizable Verilog

Back-End Infrastructure



Altera's Nios II

- Has three variations:
  - Nios II/e – unpipelined, no HW multiplier
  - Nios II/s – 5-stage, with HW multiplier
  - Nios II/f – 6-stage, dynamic branch prediction

Architectural Parameters Used in SPREE

- Multiplication Support
  - Hardware FU or software routine
- Shifter implementation
  - Flipflops, multiplier,
- Pipelining
  - Depth
    - (2-7 stages)
  - Organization
  - Forwarding

We focus on core microarchitecture

Exploration of Soft Core Processor Architectural Customizations

1. **Architectural-tuning**
2. Instruction-set subsetting
3. Combination (Arch-tuning + Subsetting)

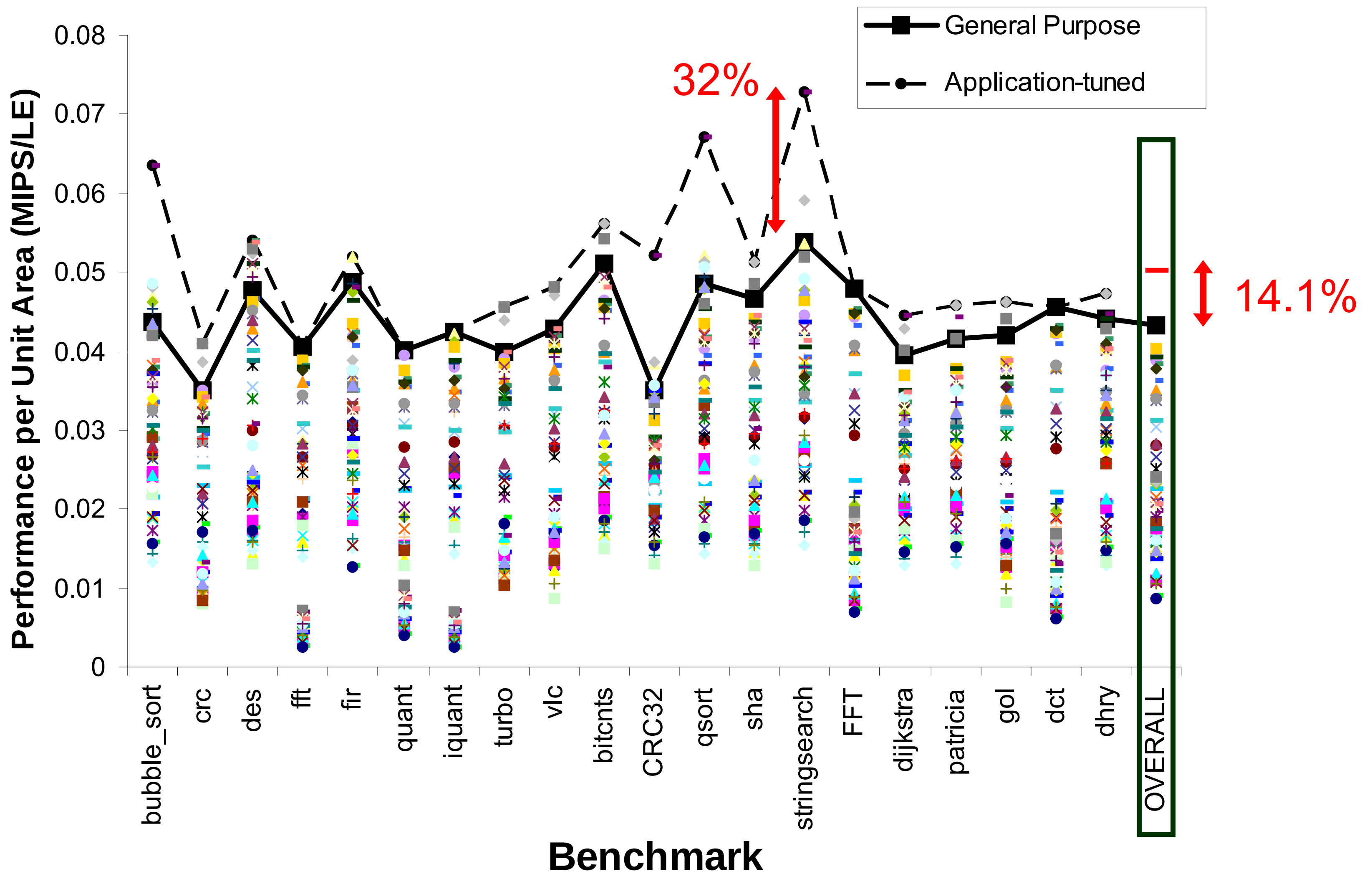
1. Architectural Tuning Experiment

- Vary the same parameters
  - Multiplication Support
  - Shifter implementation
  - Pipelining

- Determine
  1. Best overall (**general purpose**) processor
  2. Best per application (**application-tuned**)

- Metric: Performance per Area (**MIPS/LE**)
  - Basically inverse of Area-Delay product

Performance per Area of All Processors



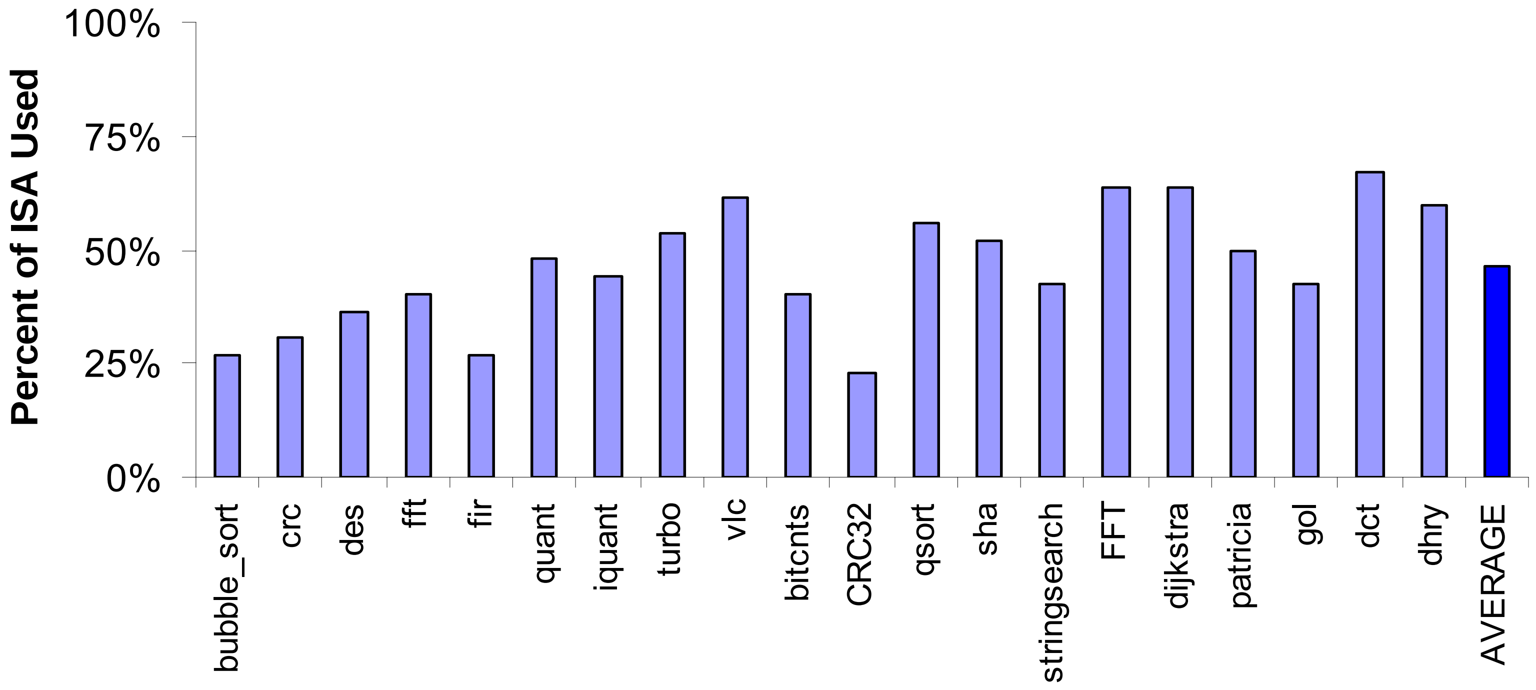
2. Instruction-set Subsetting

- SPREE automatically removes
  - Unused connections
  - Unused components

- Reduce processor by reducing the ISA
  - Can create application-specific processor
    - Eliminate unused parts of the ISA

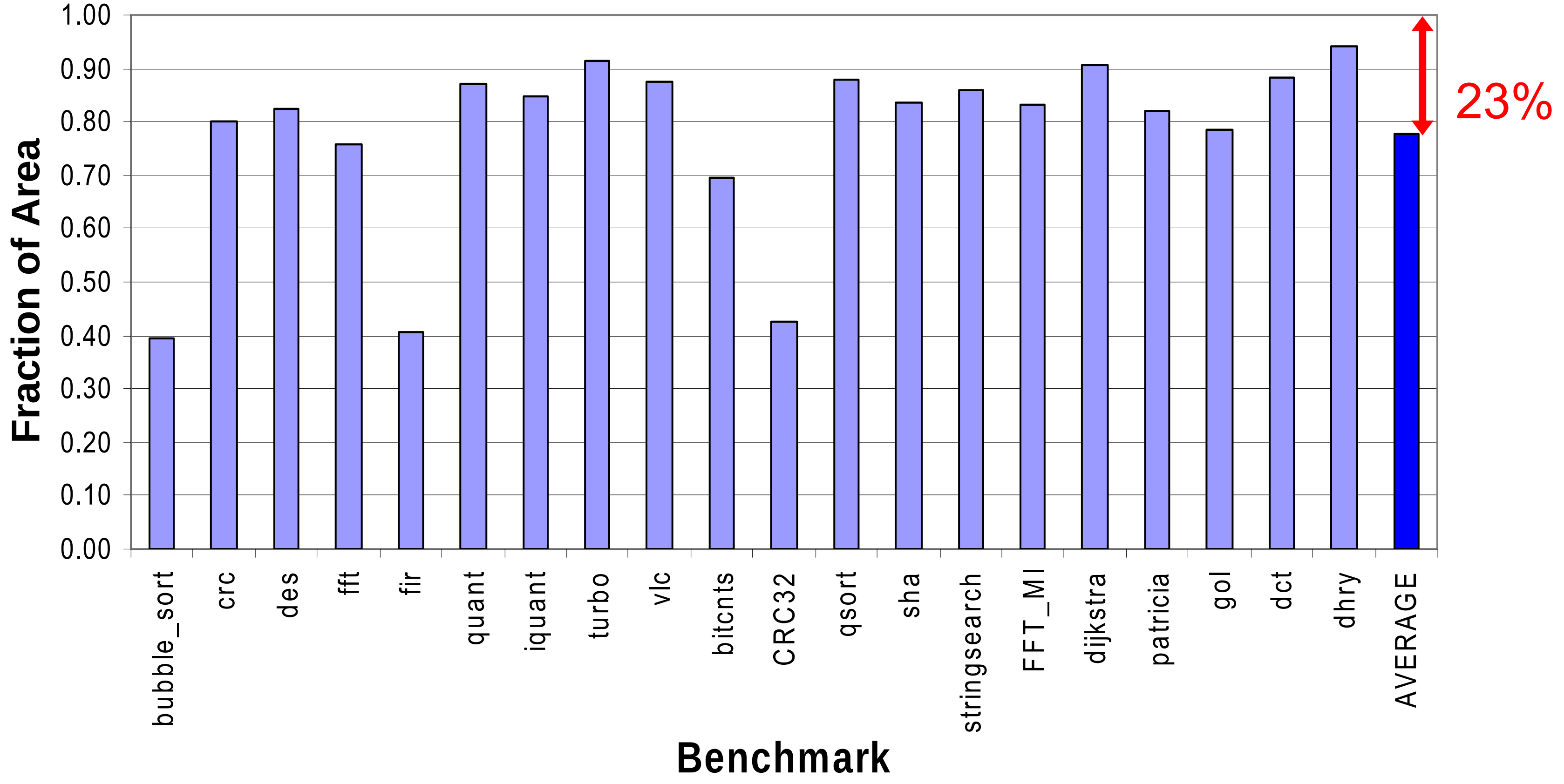
Instruction-set Usage of Benchmarks

- Applications do not use complete ISA



Strong potential for hardware reduction

Area Reduction from Subsetting



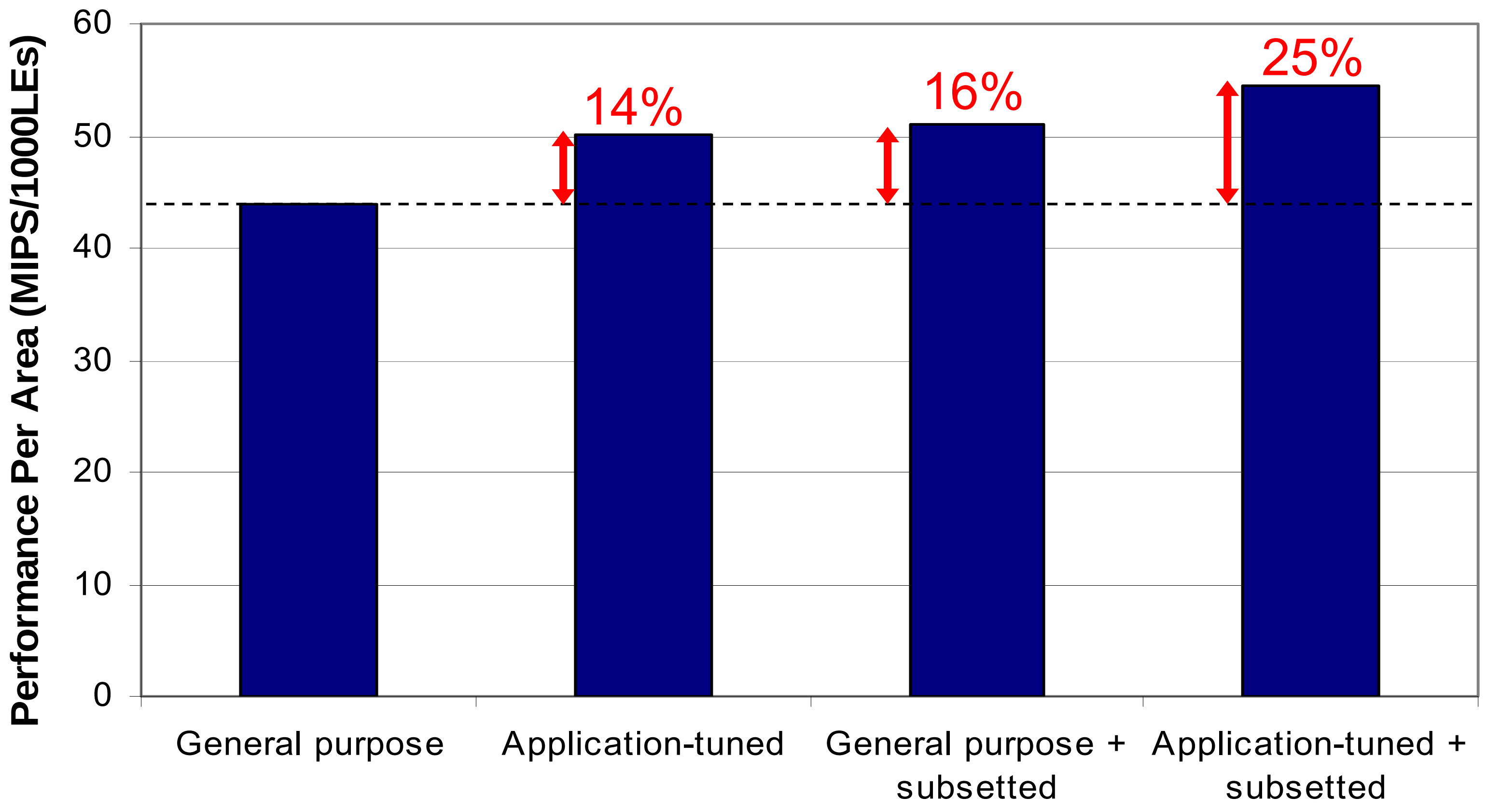
- Area reduced by 60% in some, 23% on average
- Similar reductions for energy, small impact on performance

3. Combining Application Tuning and Instruction-set Subsetting

- Subsetting is effective on its own
  - Can apply subsetting on top of tuning

- Compare different customization methods
  1. Tuning
  2. Subsetting
  3. Tuning + Subsetting

Combining Application Tuning and Instruction-set Subsetting



Tuning reduces the waste that subsetting eliminates

Summary of Presented Architectural Conclusions

- Application tuning
  - 14% average efficiency gain
  - Will increase with more architectural axes
- Instruction-set Subsetting
  - Up to 60% area & energy savings
  - 16% average efficiency gain
- Combined Tuning & Subsetting
  - 25% average efficiency gain

Future Work

- Consider other promising architectural axes
  - Branch prediction, aggressive forwarding
  - ISA changes
  - Datapaths (eg. VLIW)
  - Caches and memory hierarchy
- Compiler assistance
  - Can improve tuning & subsetting