

# Composite-ISA Cores: Enabling Multi-ISA Heterogeneity Using a Single ISA

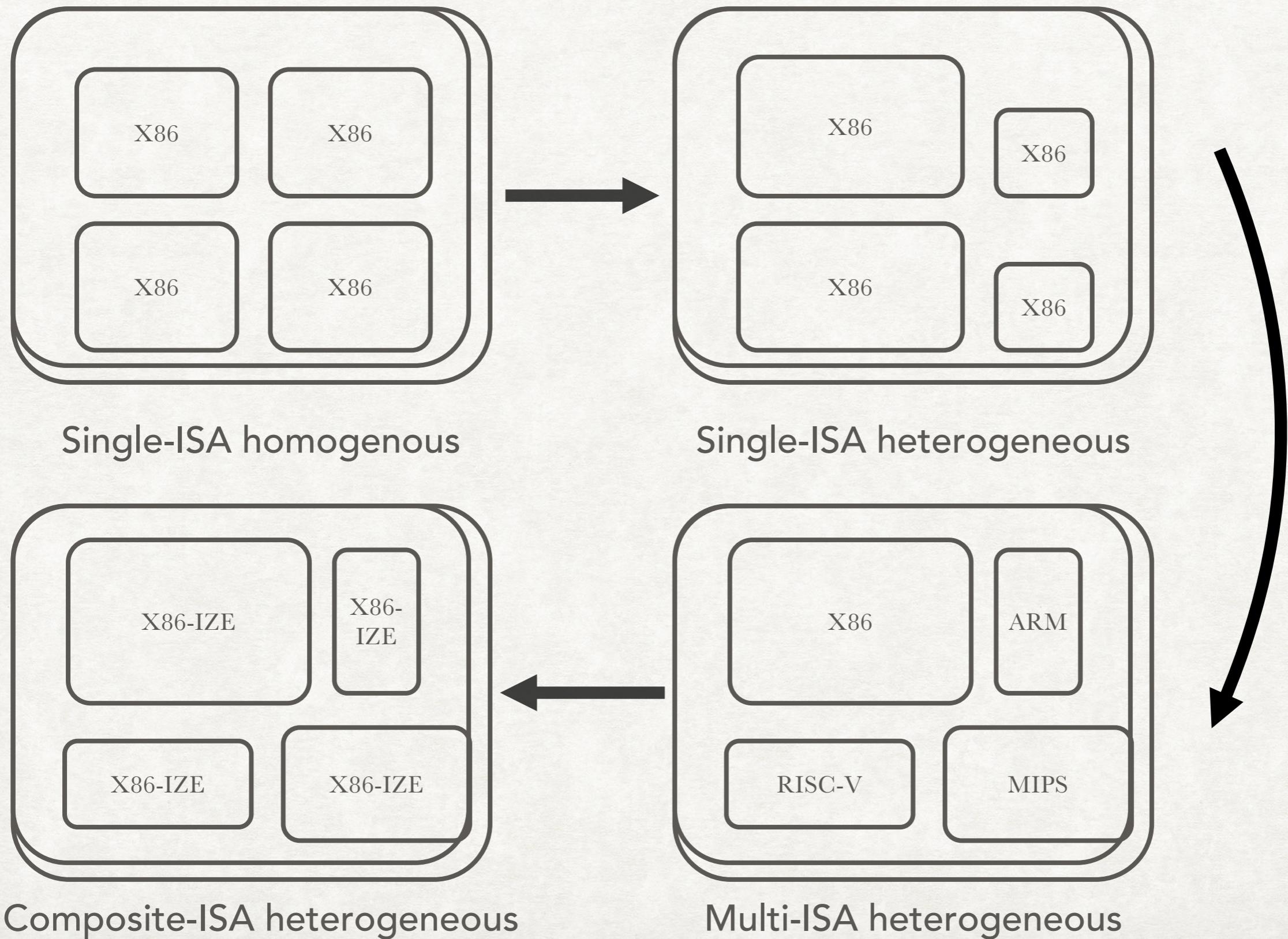
复合指令集内核：使用单一指令集实现多指令集异构

选自 2019 HPCA

# CONTENTS

- Introduction & Related Work
- ISA Feature Set Derivation
- Compiler and Runtime Strategy
- Decoder Design & Methodology
- Results & Conclusion

# INTRODUCTION & RELATED WORK

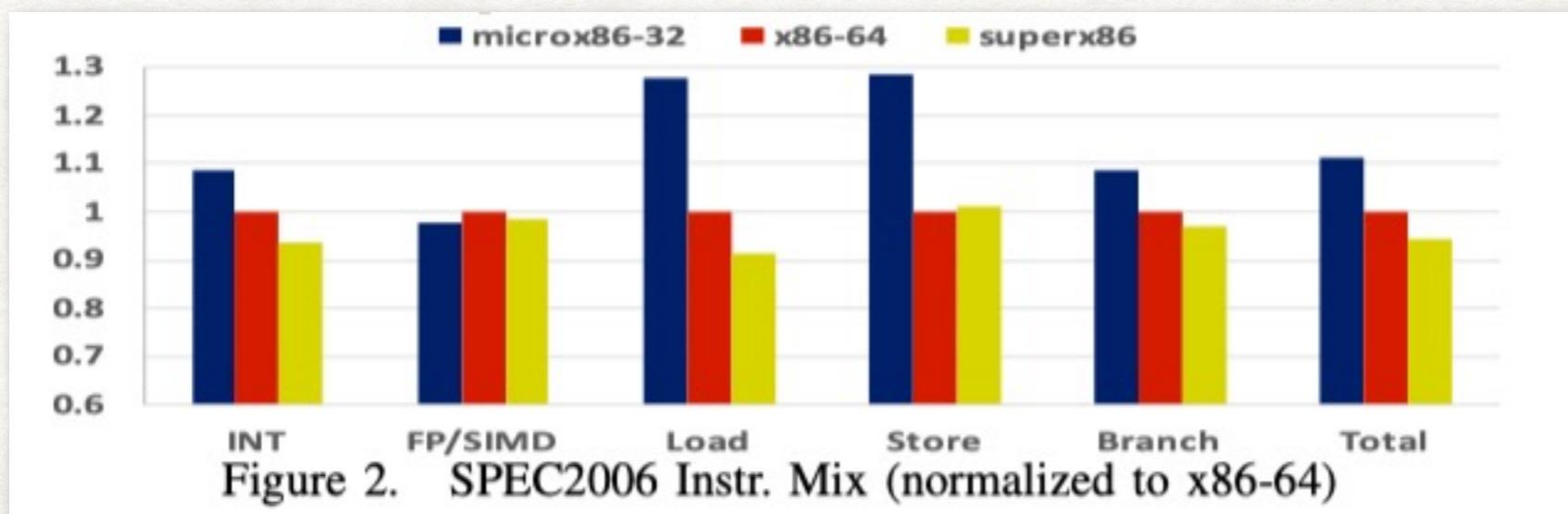


# ISA FEATURE SET DERIVATION

- superset ISA
  - resembles x86
  - add some customized features
    - Register Depth -> code optimizations & power saving
    - Register Width -> performance & efficiency
    - Instruction Complexity -> CISC & RISC
    - Predication -> Branch Prediction
    - Data-Parallel Execution -> SIMD

# REGISTER DEPTH

- providing variable number of programmable register
- more registers means better code optimization.
  - reg depth 32->16 leads to increase of stores, loads, integer instructions and branch instructions.
  - most compiler intermediate representations allow for a large number of virtual registers.



# REGISTER WIDTH

- providing variable size of programmable register
- Wider Reg Advantages:
  - accessing to large virtual memory, 32 to 64.
  - can be addressed as individual sub-registers
- Wider Reg Disadvantages:
  - Worse Performance
  - Decrease register depth

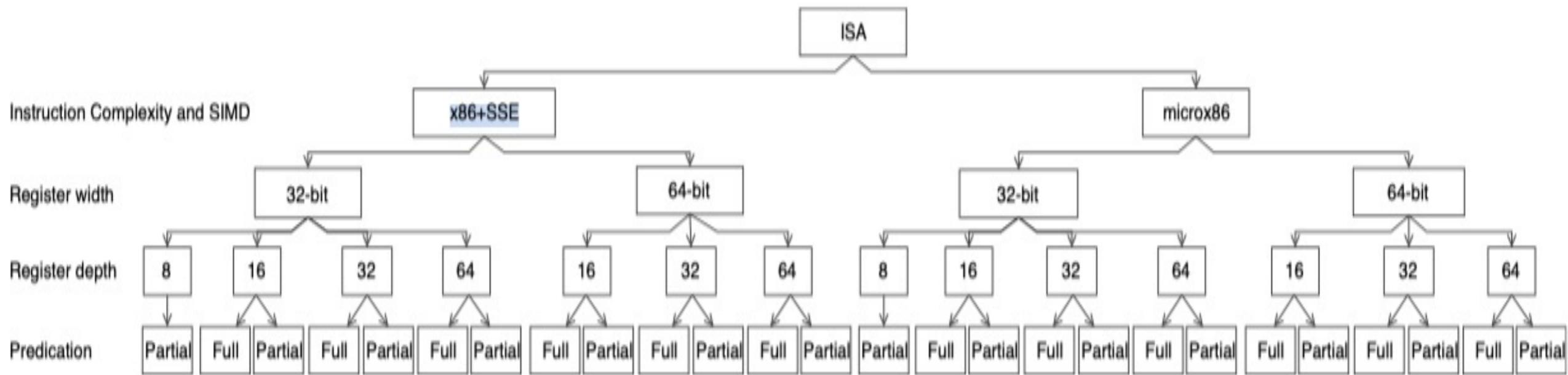


Figure 1. Derivation of Composite Feature Sets from a Superset ISA.

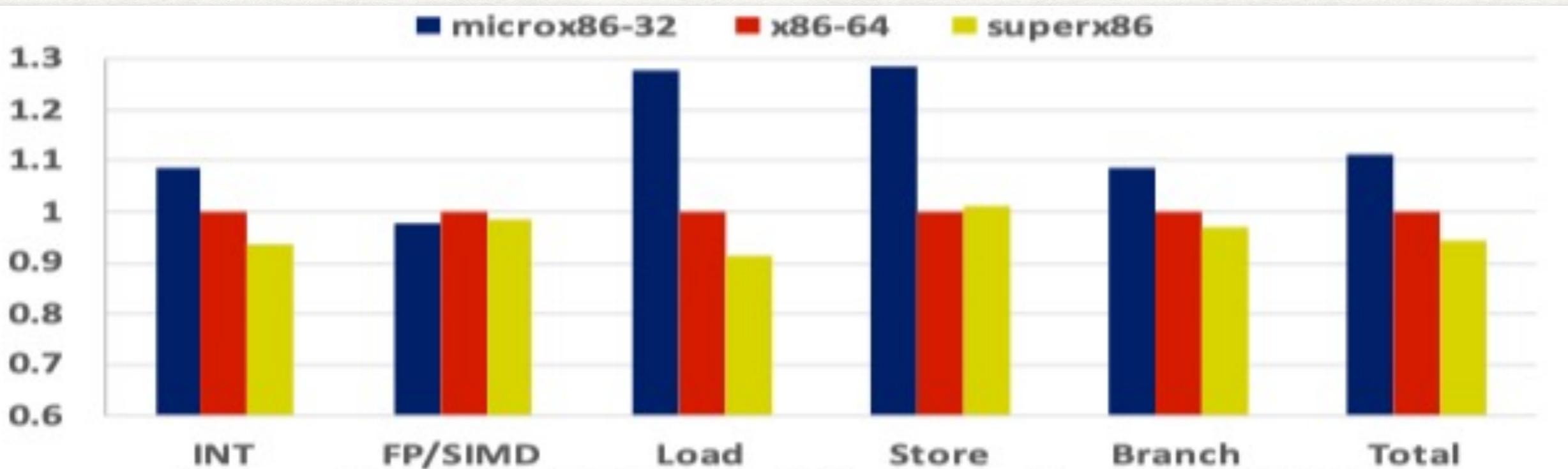


Figure 2. SPEC2006 Instr. Mix (normalized to x86-64)

# COMPILER AND RUNTIME STRATEGY

- Compiler Toolchain Development
  - generates code to efficiently take advantage of the underlying custom feature sets
  - use LLVM MC
  - allow different code regions to use different hardware.
- Migration Strategy
  - allow code regions to seamlessly migrate back and forth between different custom feature sets
  - upgrade -> no expense
  - downgrade -> extremely rare and often inexpensive

# DECODE DESIGN & METHODOLOGY

Legacy Prefixes	Predicate Prefix	REXBC Prefix	REX Prefix	Opcode	ModR/M	SIB	Displacement	Immediate
Grp 1, Grp 2, Grp 3, Grp 4 (optional)	2-byte (optional)	2-byte (optional)	(optional)	1-byte or 2-byte or 3-byte	1-byte (if required)	1-byte (if required)	Address displacement of 1-, 2- or 4-bytes	Immediate data of 1-, 2- or 4-bytes or none

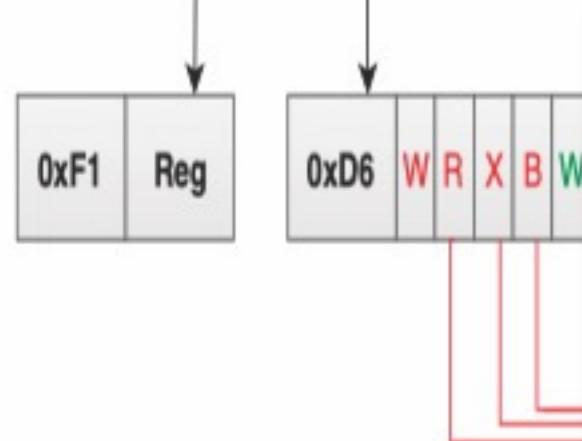


Figure 3.

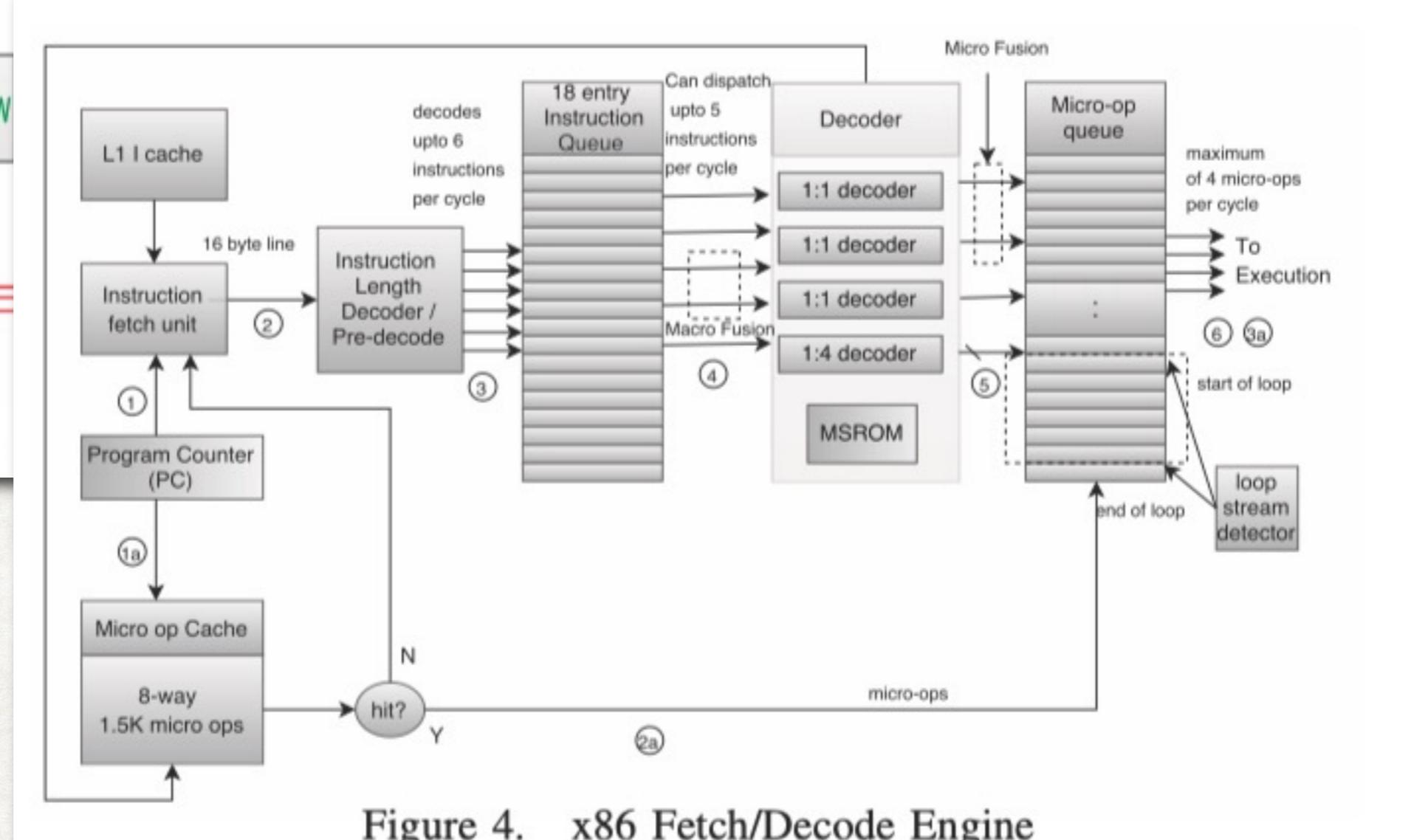


Figure 4. x86 Fetch/Decode Engine

ISA Parameter	Options
Register depth	8, 16, 32, 64 registers
Register width	32-bit, 64-bit registers
Instruction/Addressing mode complexity	1:1 macroop-microop encoding (load-store x86 micro-op ISA), 1:n macroop-microop encoding (fully CISC x86 ISA)
Predication Support	Full Predication like IA-64/Hexagon vs Partial (cmov) Predication
Data Parallelism	Scalar vs Vector (SIMD) execution
Microarchitectural Parameter	Options
Execution Semantics	Inorder vs Out-Of-Order designs
Fetch/Issue Width	1, 2, 4
Decoder Configurations	1-3 1:1 decoders, 1 1:4 decoder, MSROM
Micro-op Optimizations	Micro-op Cache, Micro-op Fusion
Instruction Queue Sizes	32, 64
Reorder Buffer Sizes	64, 128
Physical Register File Configurations	(96 INT, 64 FP/SIMD), (64 INT, 96 FP/SIMD)
Branch Predictors	2-level local, gshare, tournament
Integer ALUs	1, 3, 6
FP/SIMD ALUs	1, 2, 4
Load/Store Queue Sizes	16, 32
Instruction Cache	32KB 4-way, 64KB 4-way

## X86-IZED VERSIONS OF THUMB , ALPHA , AND X86-64

<b>microx86-8D-32W</b>	<b>microx86-32D-64W</b>	<b>x86-16D-64W</b>
<b>Thumb-like Features</b> Load/Store Architecture Register Depth: 8 Register Width: 32 No SIMD support	<b>Alpha-like Features</b> Load/Store Architecture Register Depth: 32 Register Width: 64 No SIMD support CMOV Support	<b>x86-64-like Features</b> CISC Architecture Register Depth: 64 Register Width: 64 SIMD support CMOV Support
<b>Exclusive Features:</b> FP Support	<b>Exclusive Features:</b> None	<b>Exclusive Features:</b> None
<b>Thumb-specific Features:</b> Code Compression Fixed-length instructions (one-step decoding)	<b>Alpha-specific Features:</b> 2-address instructions Fixed-length instructions (one-step decoding) More FP Registers	<b>x86-specific Features:</b> None

# RESULTS

- Homogeneous (x86-64)
- Single-ISA Heterogeneous (x86-64 + Hardware Heterogeneity)
- Composite-ISA with Fixed Feature Sets (x86-64 + Hardware Heterogeneity + x86ized Thumb + x86ized Alpha)
- Heterogeneous-ISA (x86-64 + Alpha + Thumb + Hardware Heterogeneity)
- Composite-ISA (x86-64 + Hardware Heterogeneity + Full Feature Diversity)

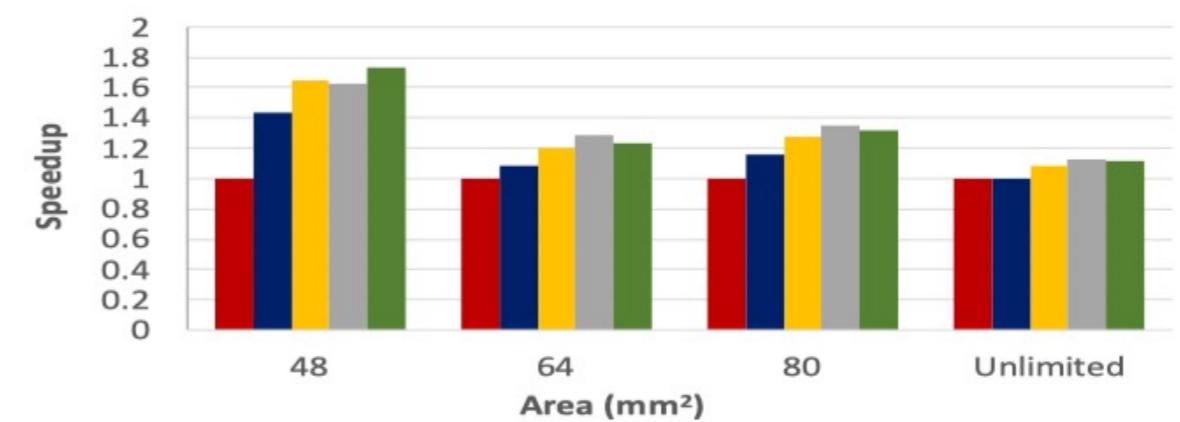
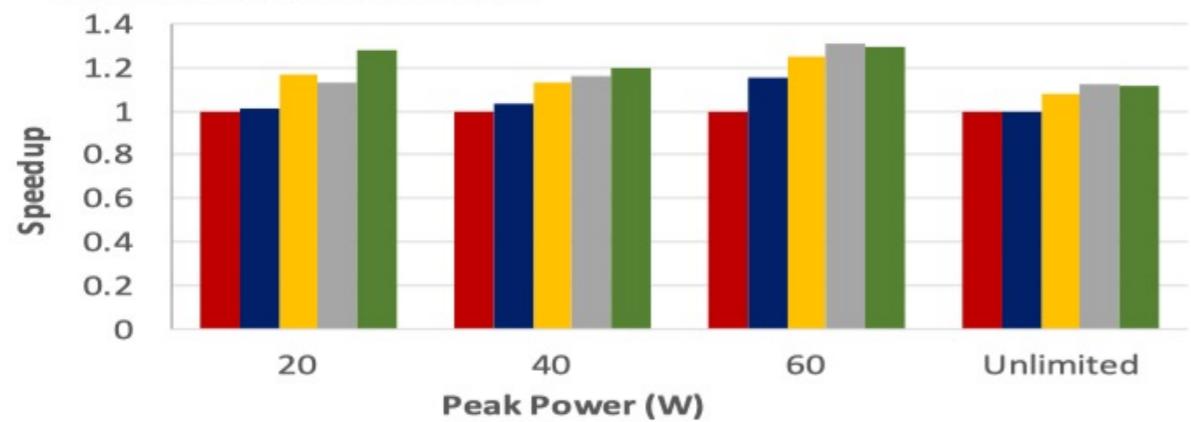


Figure 5. Multi-programmed workload throughput comparison (higher is better)

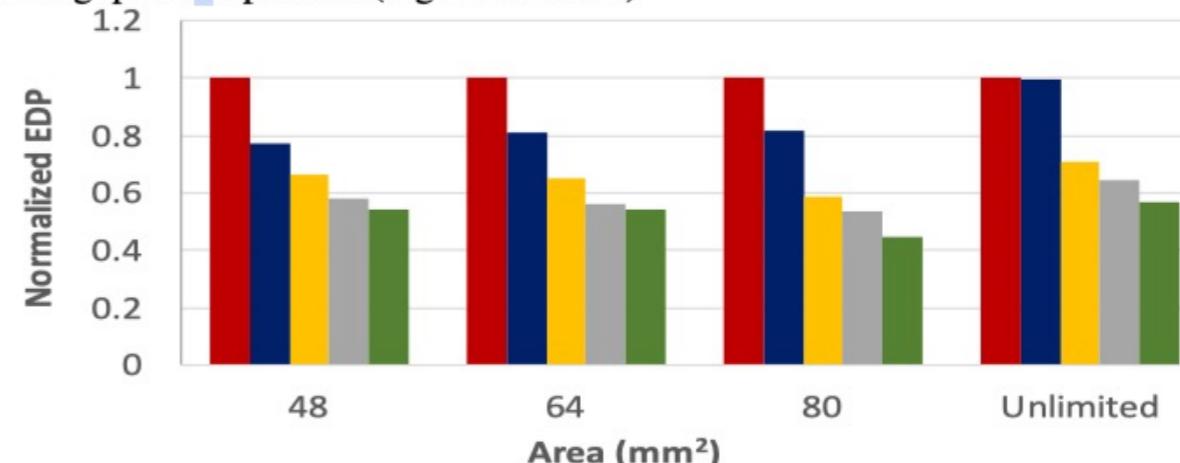
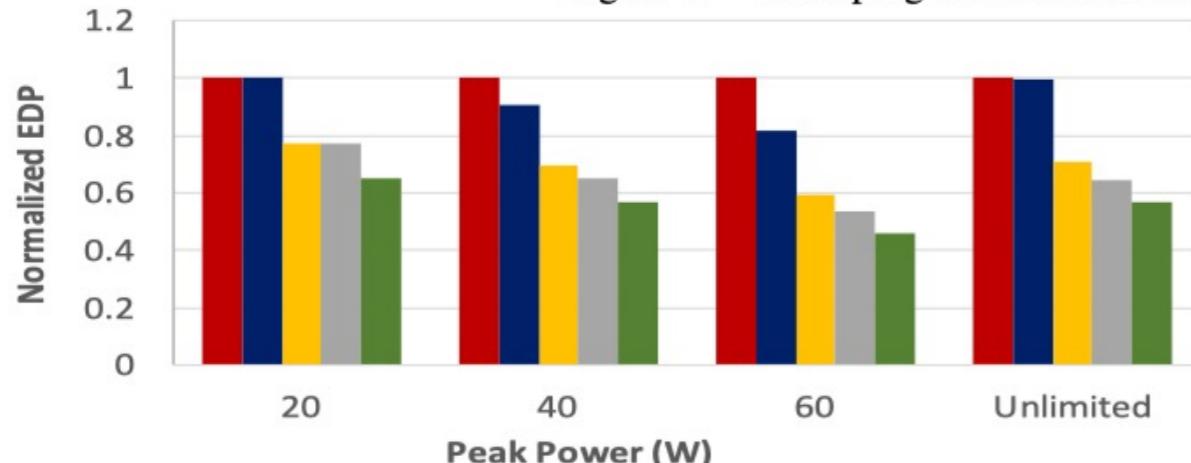


Figure 6. Multi-programmed workload EDP comparison (lower is better)

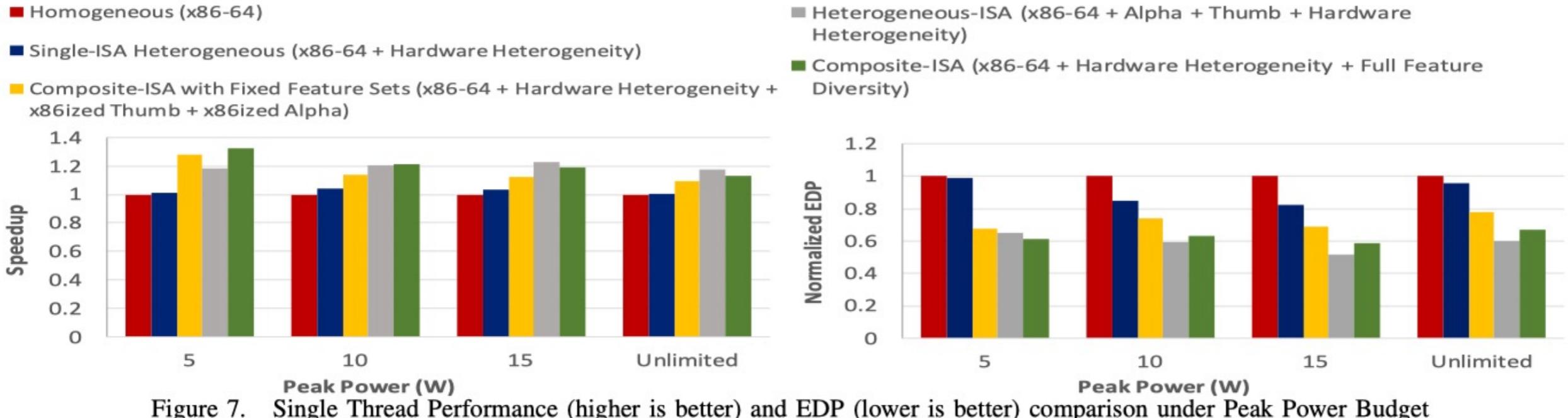


Figure 7. Single Thread Performance (higher is better) and EDP (lower is better) comparison under Peak Power Budget

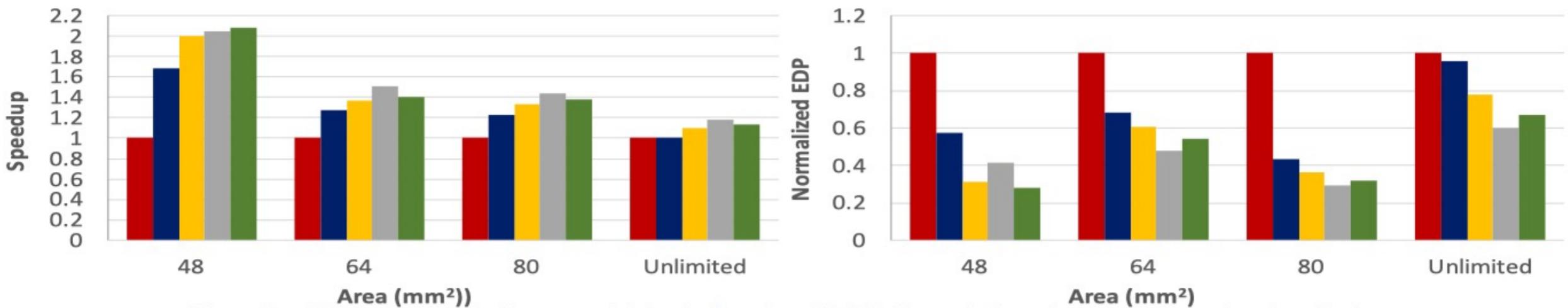
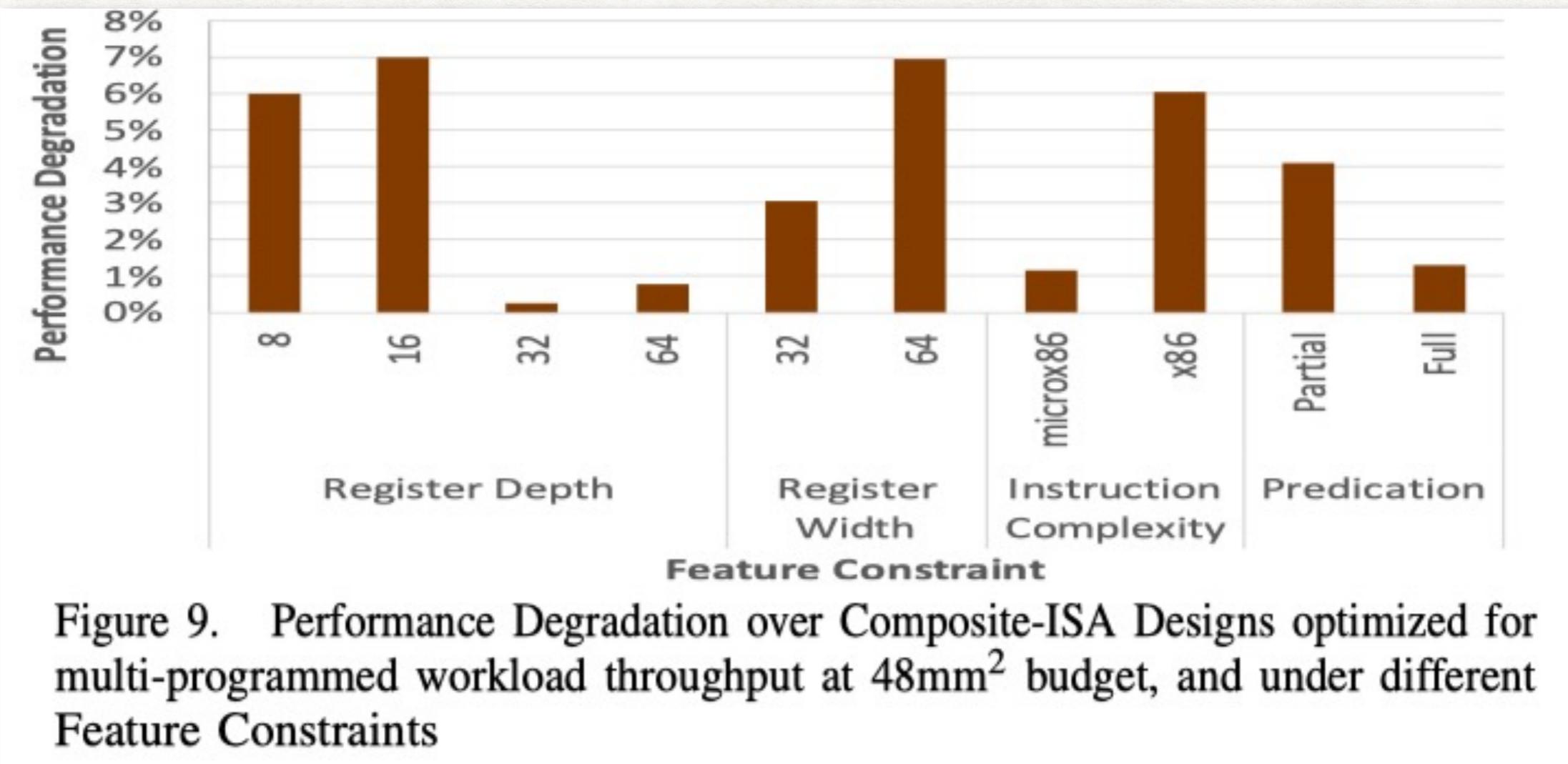


Figure 8. Single Thread Performance (higher is better) and EDP (lower is better) comparison under Area Budget



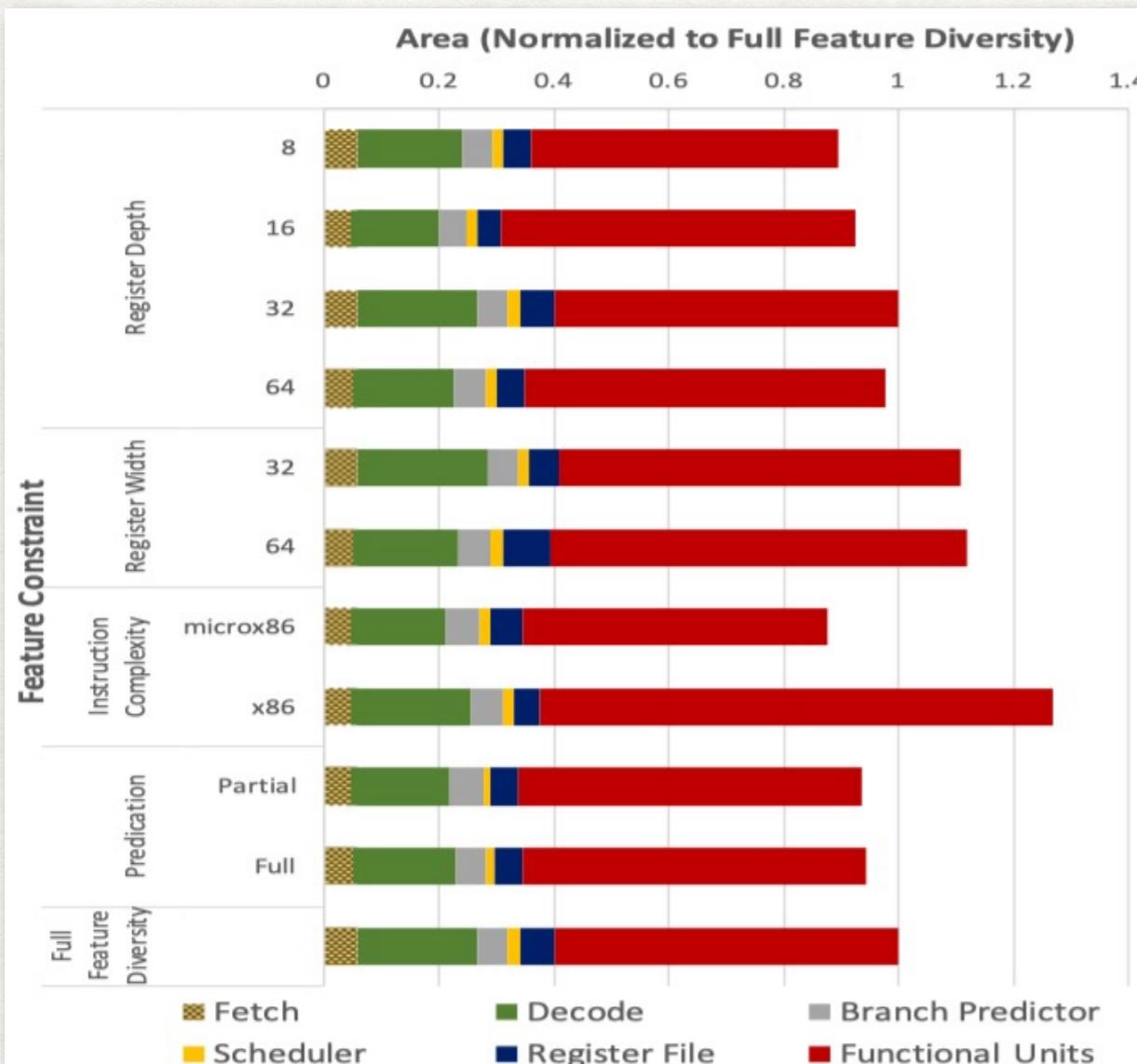


Figure 10. Transistor Investment by Processor Area normalized over that of Composite-ISA Designs optimized for multi-programmed workload throughput at  $48\text{mm}^2$  budget, and under different Feature Constraints

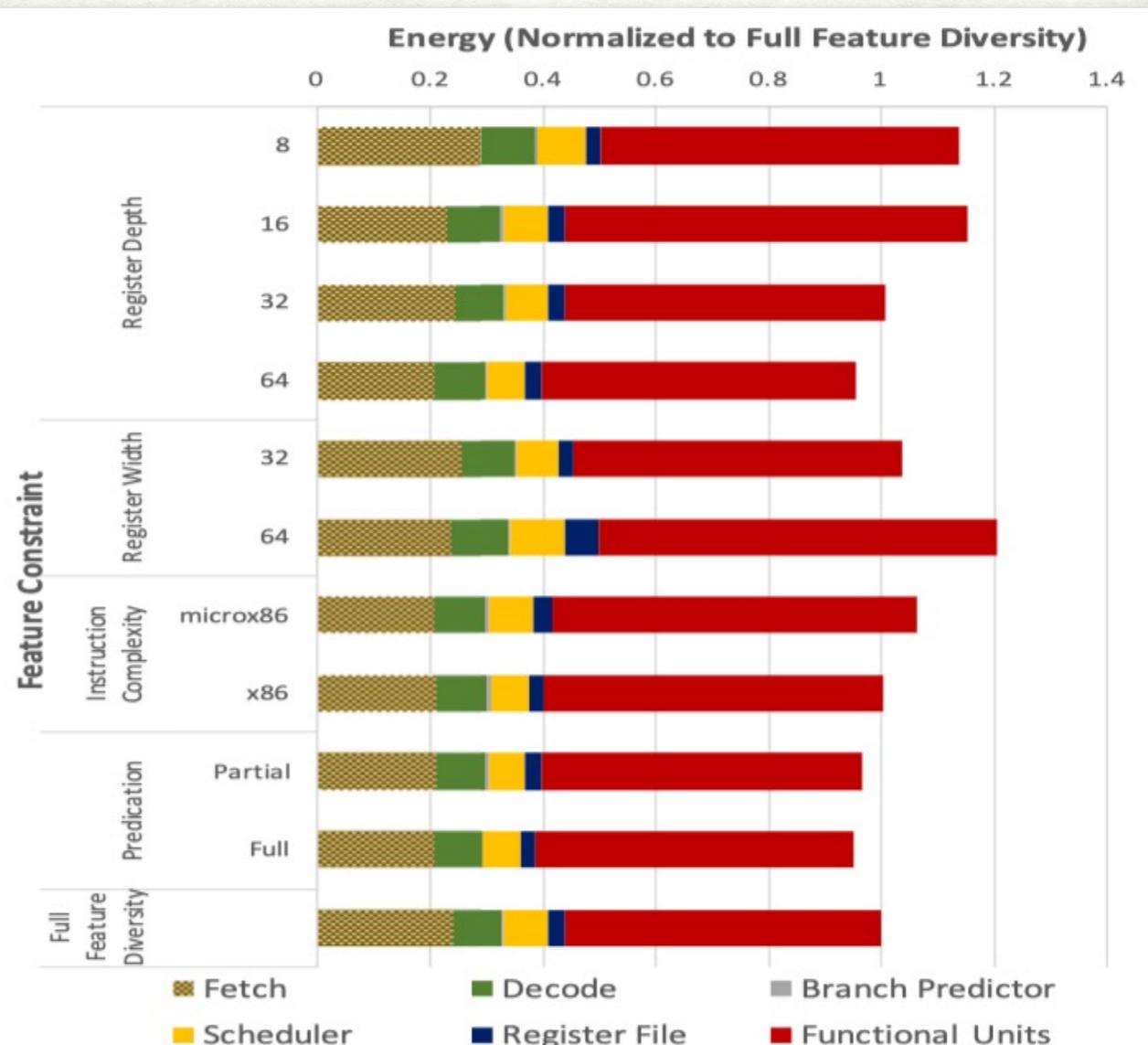


Figure 11. Processor Energy Breakdown normalized over that of Fully Custom Designs optimized for multi-programmed workload throughput at  $48\text{mm}^2$  budget, and under different Feature Constraints

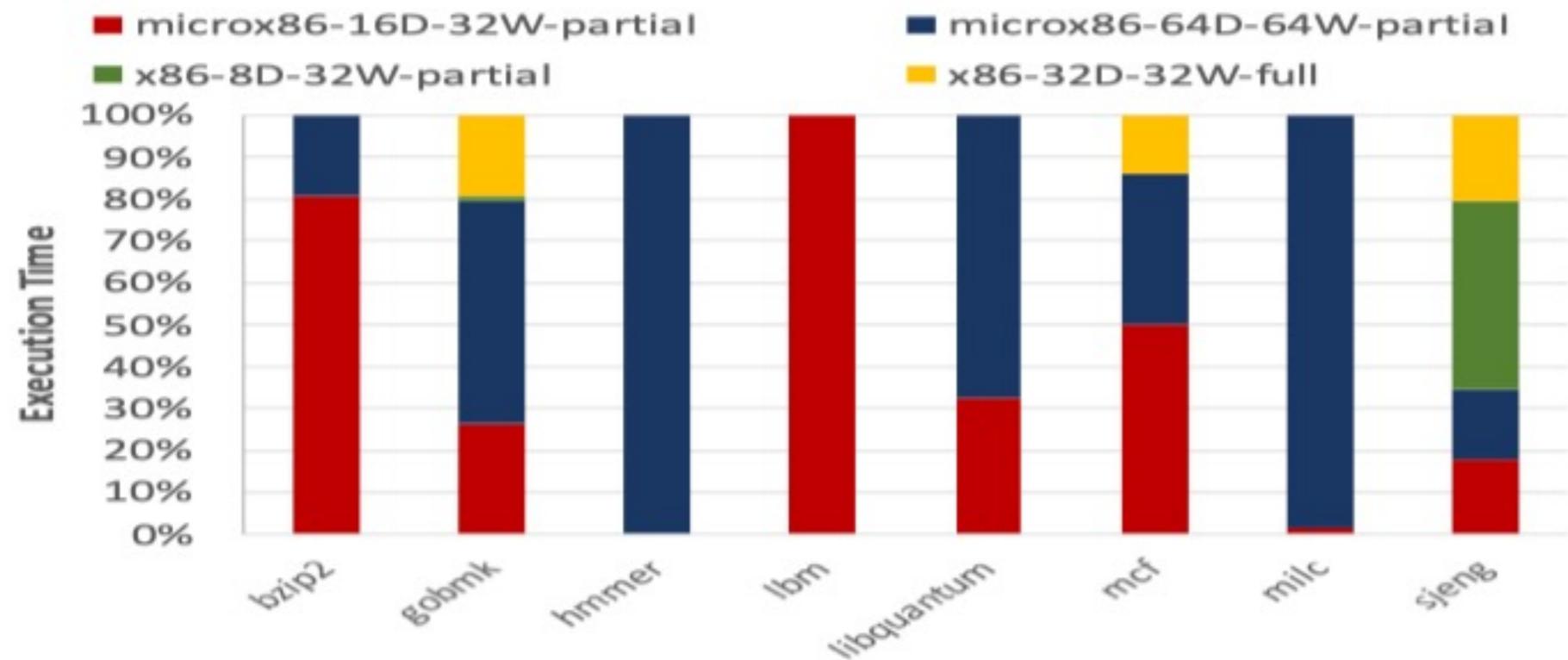


Figure 12. Execution Time Breakdown on the best composite-ISA CMP optimized for Single Thread Performance under 10W Peak Power Budget

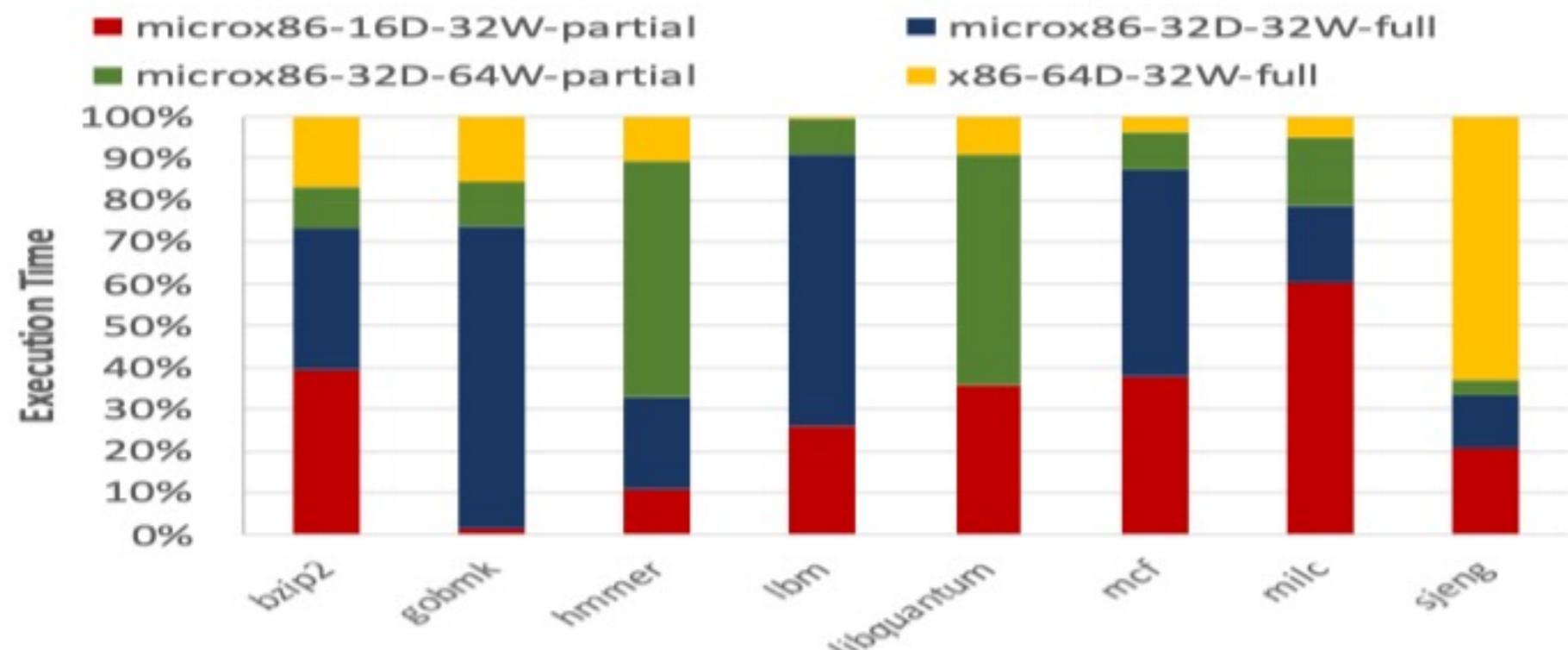


Figure 13. Execution Time Breakdown on the best composite-ISA CMP optimized for Multi-programmed Throughput under 48mm<sup>2</sup> Area Budget

# CONCLUSION

- enables the full performance and energy benefits of multi-ISA design
- eliminate the issues of multi-vendor licensing, cross-ISA binary translation, and state transformation.
- richer set of ISA design choices
- more efficiency than highly optimized but inflexible existing-ISA
- better performance and energy savings than single-ISA heterogenous designs