

An Architecture Framework for an Adaptive Extensible Processor

Noori, Hamid

Graduate school of Information Science and Electrical Engineering, Kyushu University

<https://hdl.handle.net/2324/9164>

出版情報 : SLRC プレゼンテーション, 2007-07-25. 九州大学システムLSI研究センター
バージョン :
権利関係 :





An Architecture Framework for an Adaptive Extensible Processor

Hamid Noori

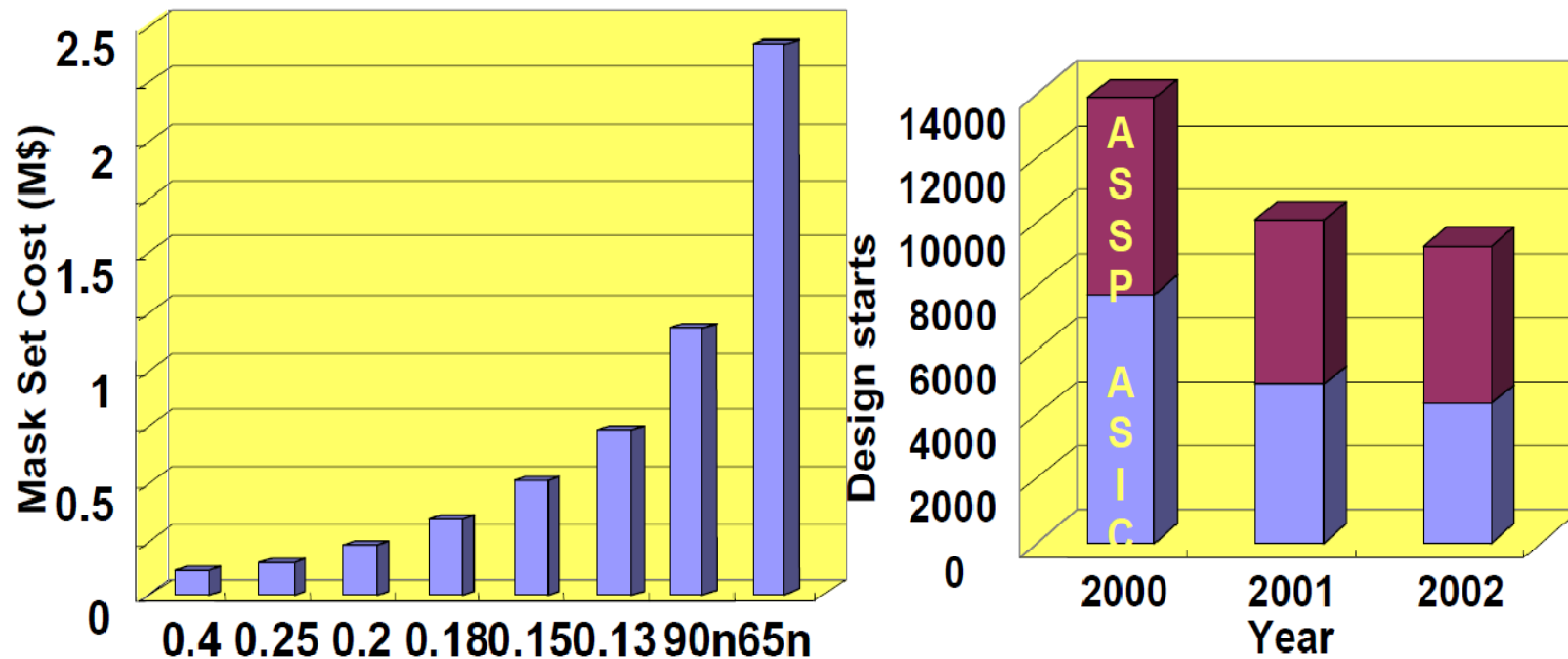
Graduate School of Information Science and Electrical
Engineering, Department of Informatics
Kyushu University

[Outline]

- Motivations
- Goal
- An Adaptive Extensible Processor
 - Extensible Processors
 - General Overview of Proposed Architecture
 - Generating Custom Instructions
 - Proposed Reconfigurable Functional Unit
- Evaluation Results

[Motivations (1/2)]

■ Exploding NRE Costs



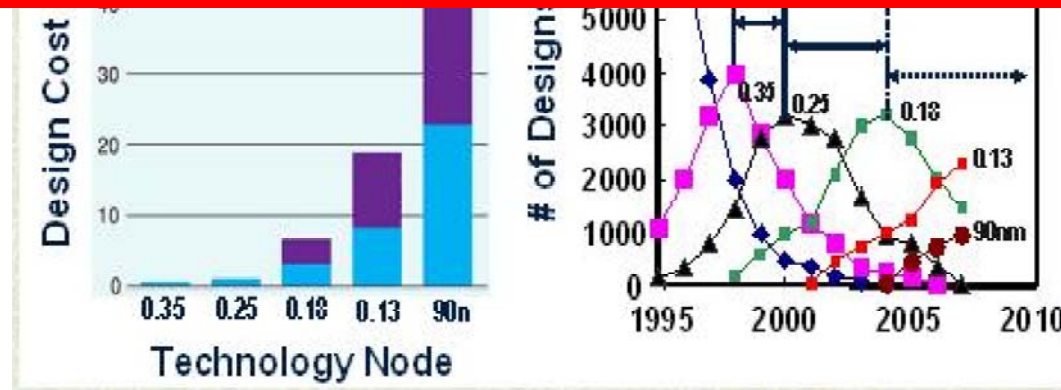
Keynote @ ASP-DAC 2007



Motivations (2/2)

- Higher design costs due to more complex SoCs
- More complex applications → more computation and power
- SoCs

This has led to the quest for a flexible and reusable embedded processor that still must achieve the required performance and energy efficiency levels.



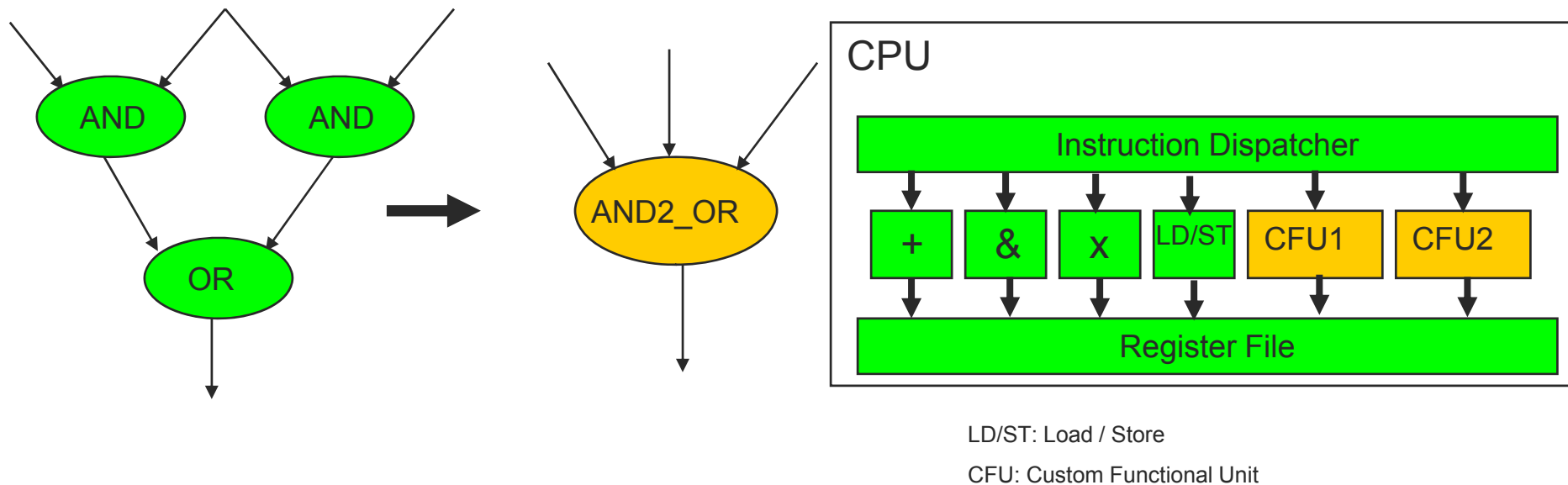
[Goal]

- Improving the performance and energy efficiency of embedded processors, while maintaining binary compatibility, and flexibility of embedded processors.



[An Adaptive Extensible Processor]

■ Introduction to Extensible Processors



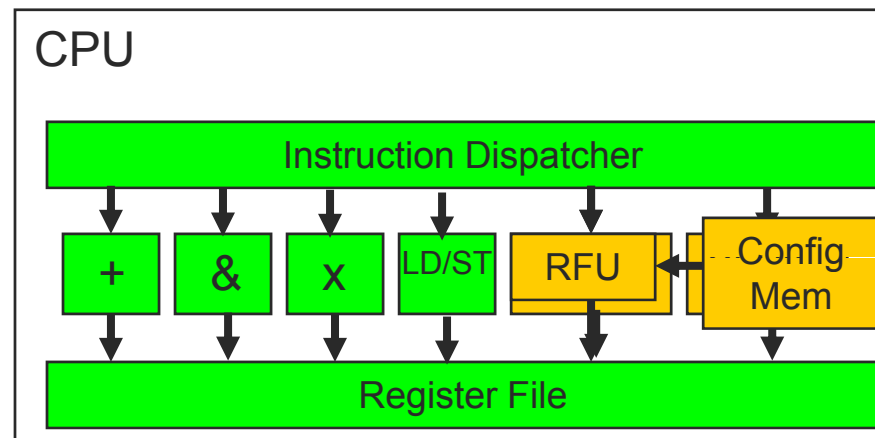
[Custom Instructions]

- Improve performance efficiency
 - Increasing parallelism
 - Reducing the latency of critical path
 - Reducing number of intermediate results written to the register file
 - Reducing cache misses
- Improve energy efficiency
 - Reducing accesses to
 - Instruction cache
 - Register file
 - Decoder
 - ALU
 - Cache misses
 - Reducing execution time (clock energy)



[Proposed Approach]

- An Adaptive Extensible Processor (ADEXOR)
 - Adding and generating custom instructions after fabrication
 - Using a reconfigurable functional unit (RFU) instead of custom functional unit

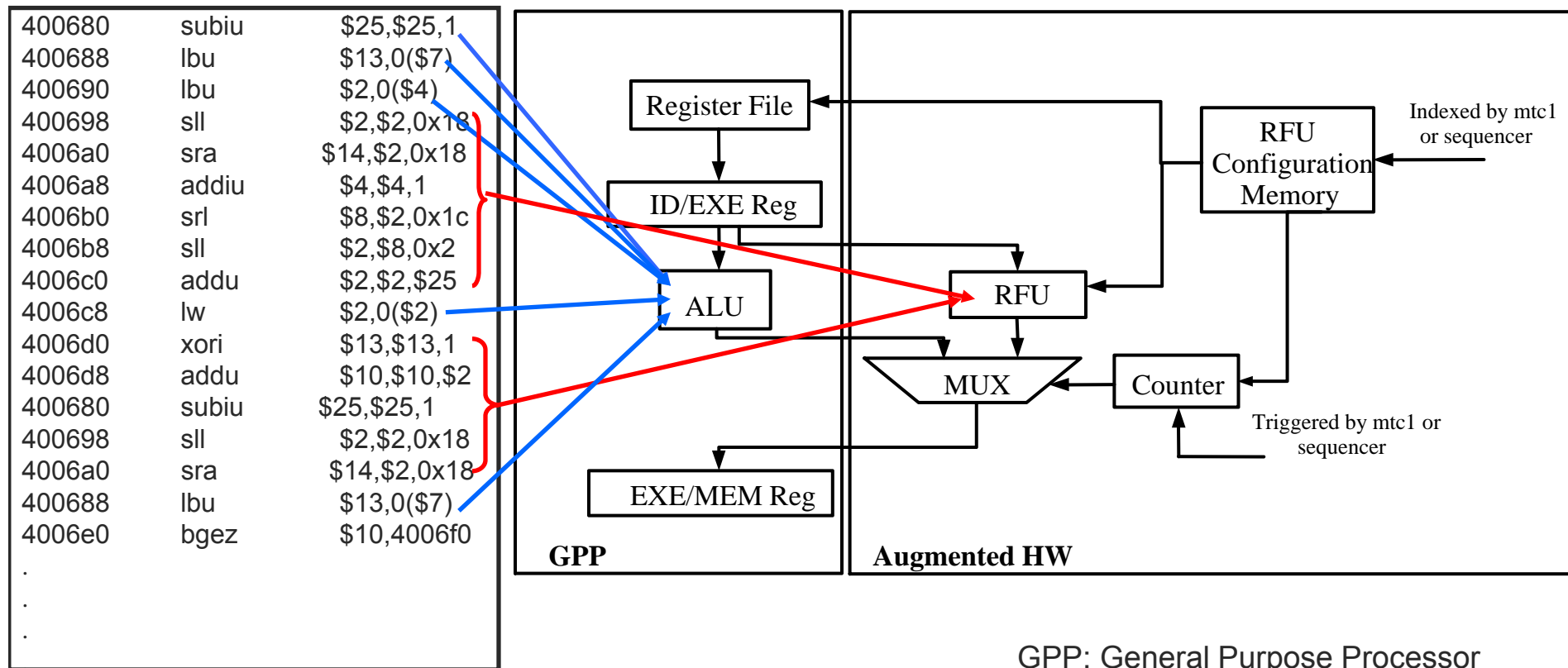


CFU: Custom Functional Unit

RFU: Reconfigurable Functional Unit



General Overview of the Proposed Architecture



Hot Basic Block

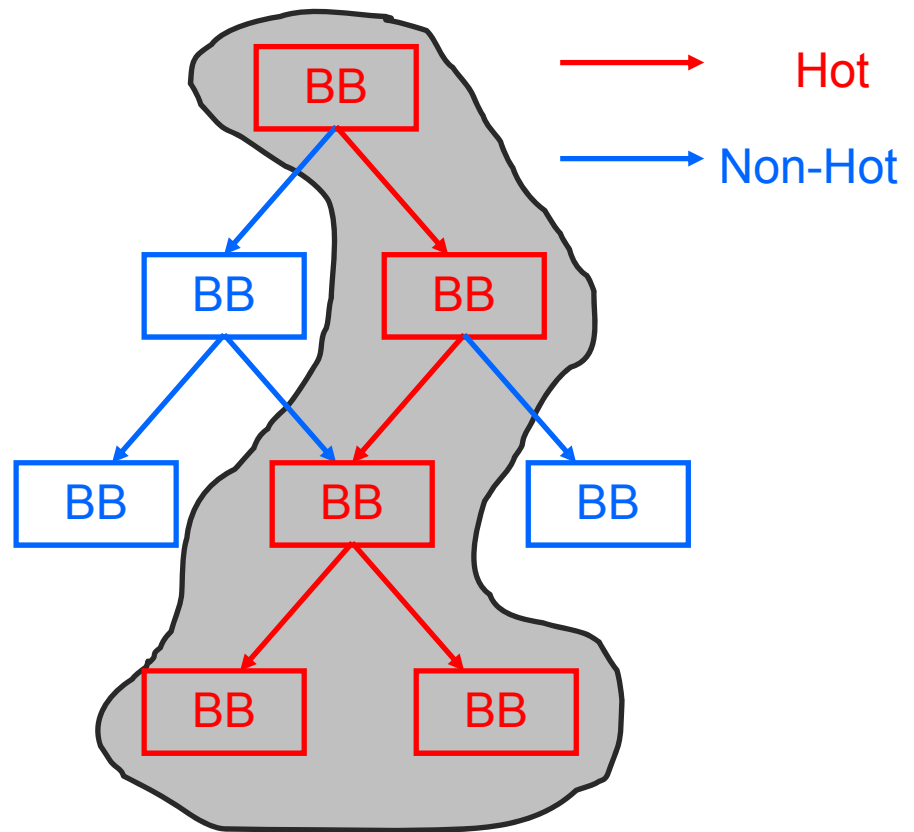
GPP: General Purpose Processor

RFU: Reconfigurable Functional Unit



[Generating Custom Instructions]

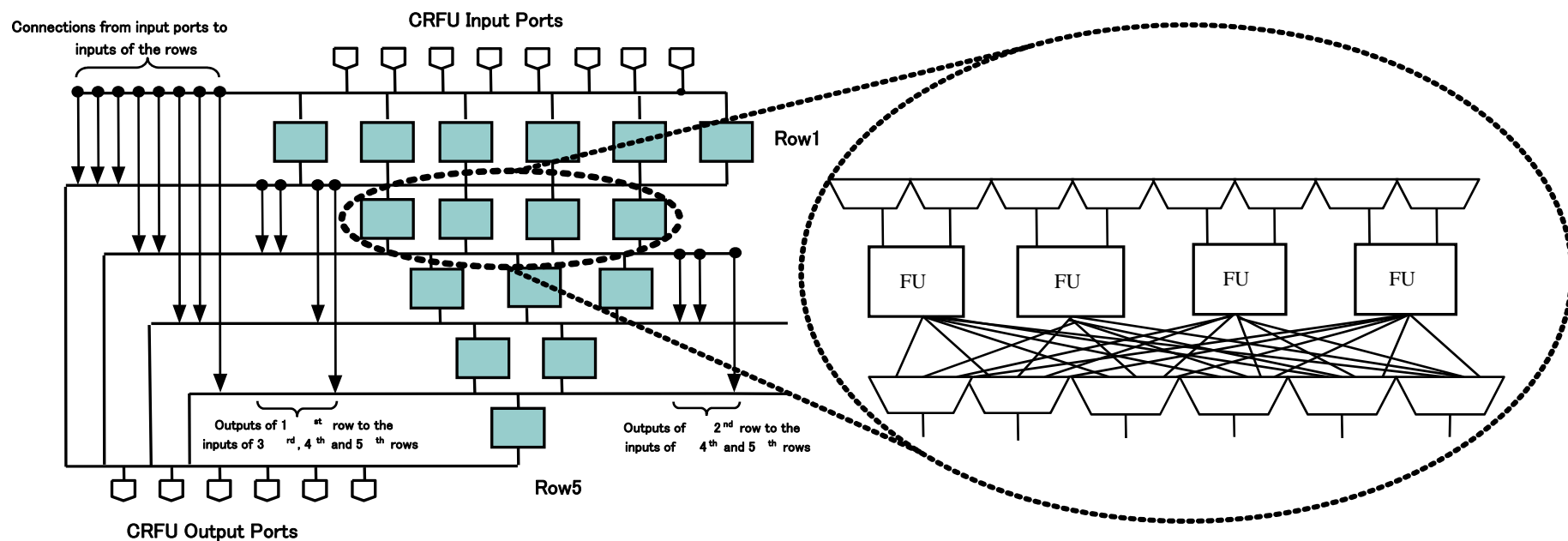
- Load, divide, multiply, floating point operations are not supported



BB: Basic Block



Architecture of the RFU

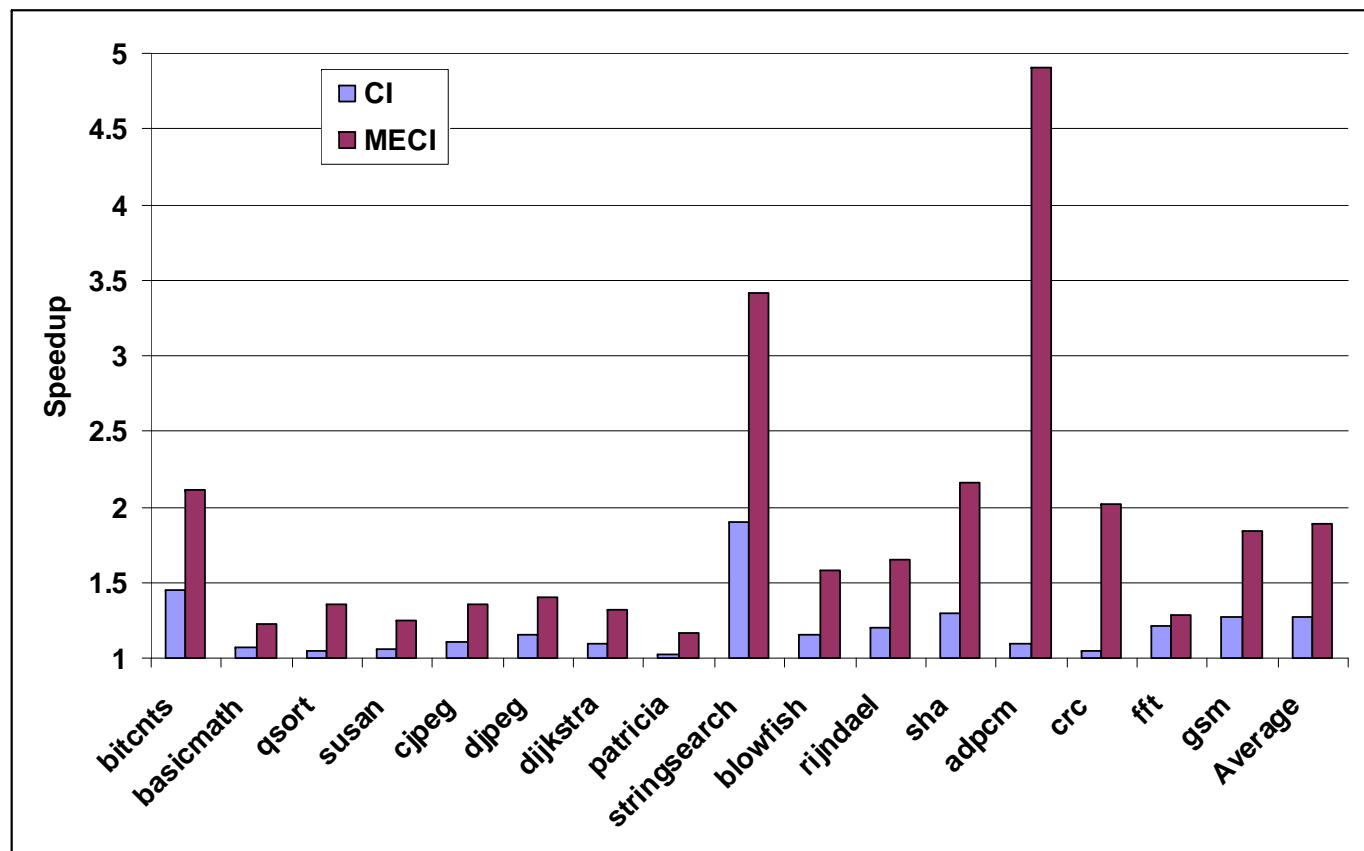


[Evaluation Results]

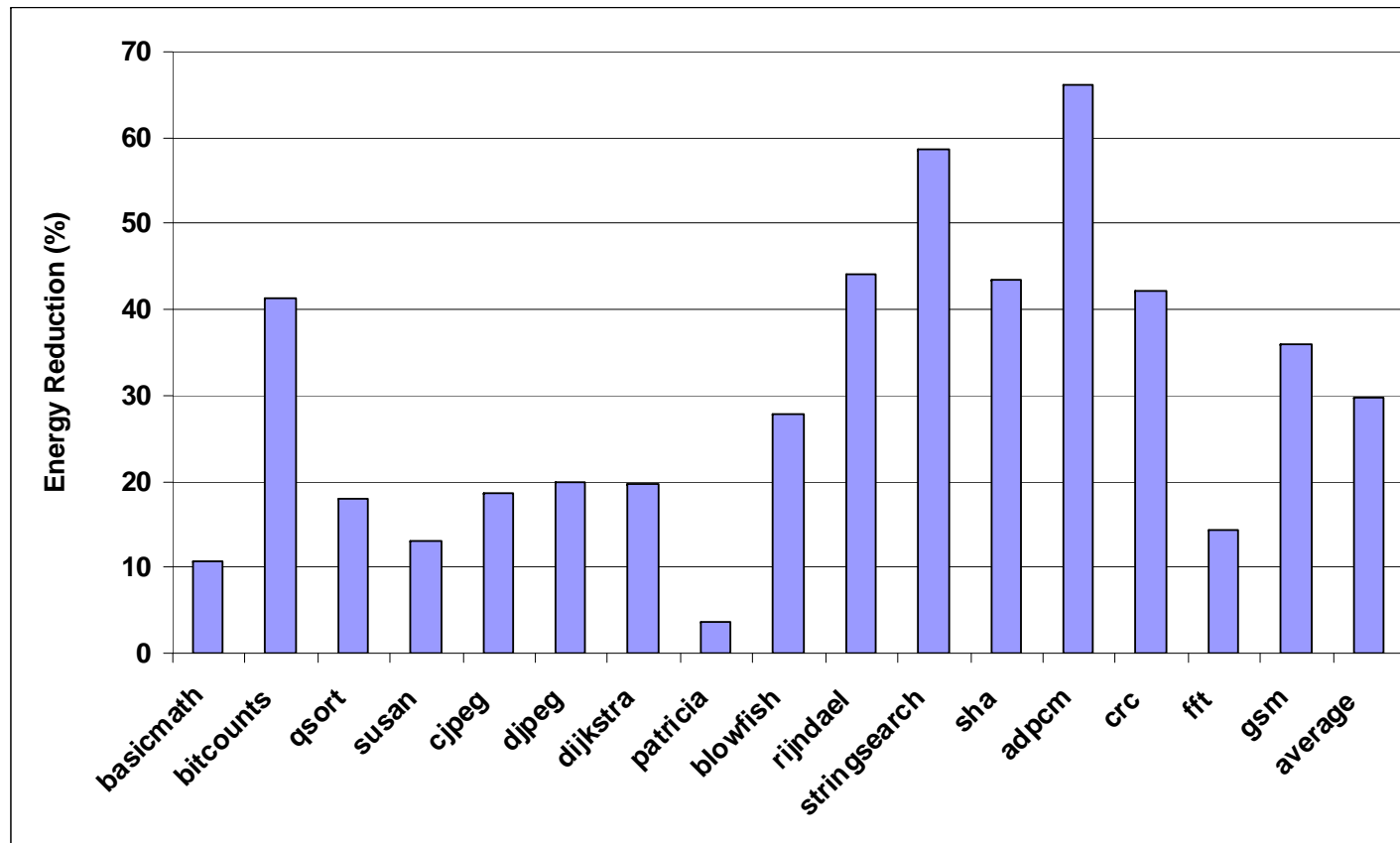
Issue	1-way
L1- I cache	16K, 2 way, 1 cycle latency for hit, 20 cycles for miss
L1- D cache	16K, 4 way, 1 cycle latency for hit, 20 cycles for miss
Execution units	1 integer unit, 1 floating point unit , 1 divider (8 cycles), 1 multiplier (5 cycles)
Branch predictor	bimodal
Branch prediction table size	256
Extra branch misprediction latency	3
Clock frequency	135 MHz



Performance Evaluation



[Energy Evaluation]



[Area Overhead]

- Base Processor 0.18 μ m (4.5mm²) + Cache (3.2mm²) = 7.7mm²
- Area Overhead: 37%

Conclusions

- An architecture framework for an adaptive extensible processor
- Generating and adding custom instructions after chip-fabrication
- No new compiler, no source code modification and recompiling
- Evaluation
 - Speedup: max. 4.9 and average 1.9
 - Energy saving: max. 67% and 30% in average
 - Area overhead: 36%





Thank you for your attention