System-level process variability compensation on memory organizations. On the scalability of multi-mode memories

Concepción Sanz Pineda

(Universidad Complutense de Madrid)

Outline

- Motivation
- Multimode memories
- Methodology
- Scalability
- Results

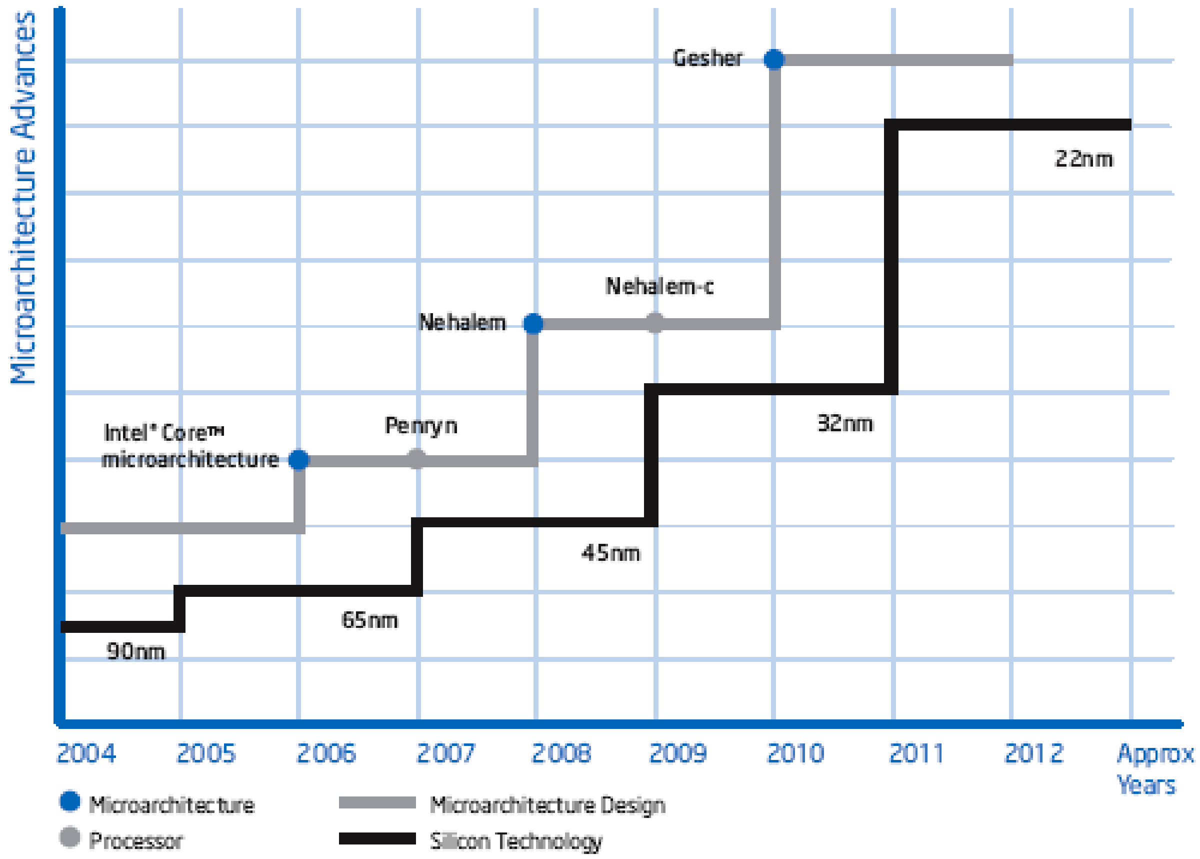
Outline

- **Motivation**
- Multimode memories
- Methodology
- Scalability
- Results

Memories: main victims of variability

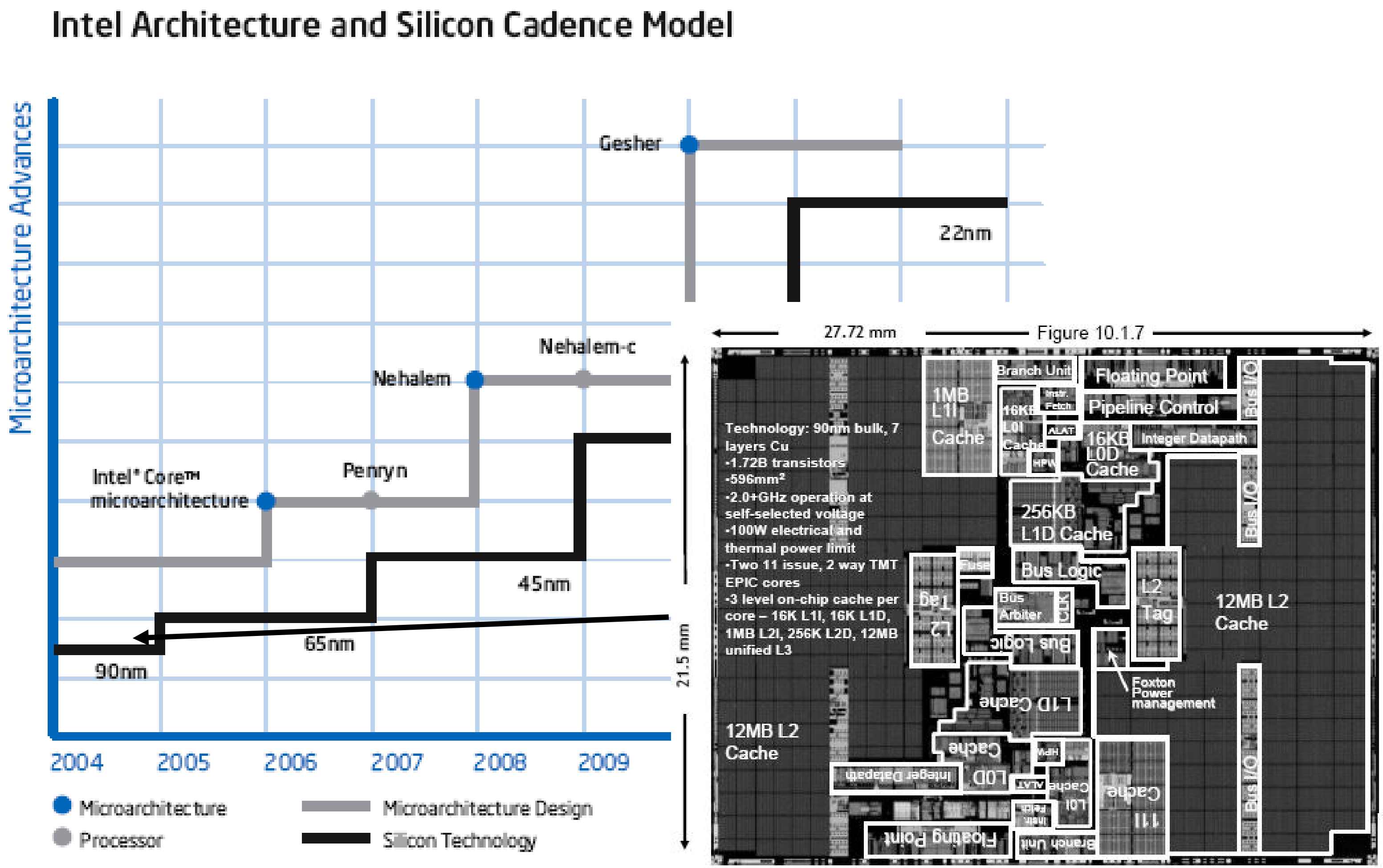
Larger and more dense on-chip memories

Intel Architecture and Silicon Cadence Model



Memories: main victims of variability

Larger and more dense on-chip memories

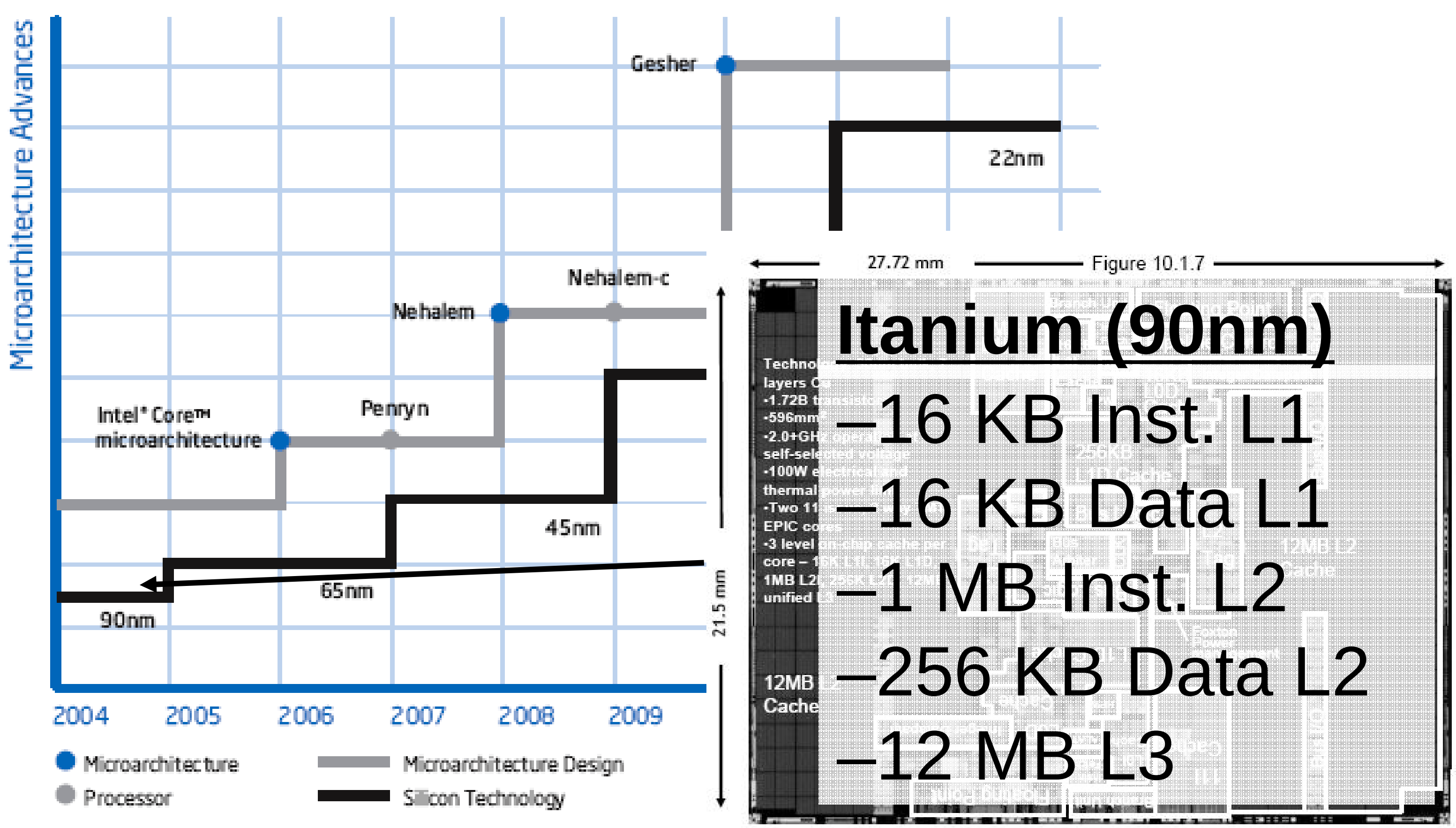


Technology: 90nm bulk, 7 layers Cu
•1.72B transistors
•596mm²
•2.0+GHz operation at self-selected voltage
•100W electrical and thermal power limit
•Two 11 issue, 2 way TMT EPIC cores
•3 level on-chip cache per core – 16K L1I, 16K L1D, 1MB L2I, 256K L2D, 12MB unified L3

Memories: main victims of variability

Larger and more dense on-chip memories

Intel Architecture and Silicon Cadence Model



Itanium (90nm)

–16 KB Inst. L1

–16 KB Data L1

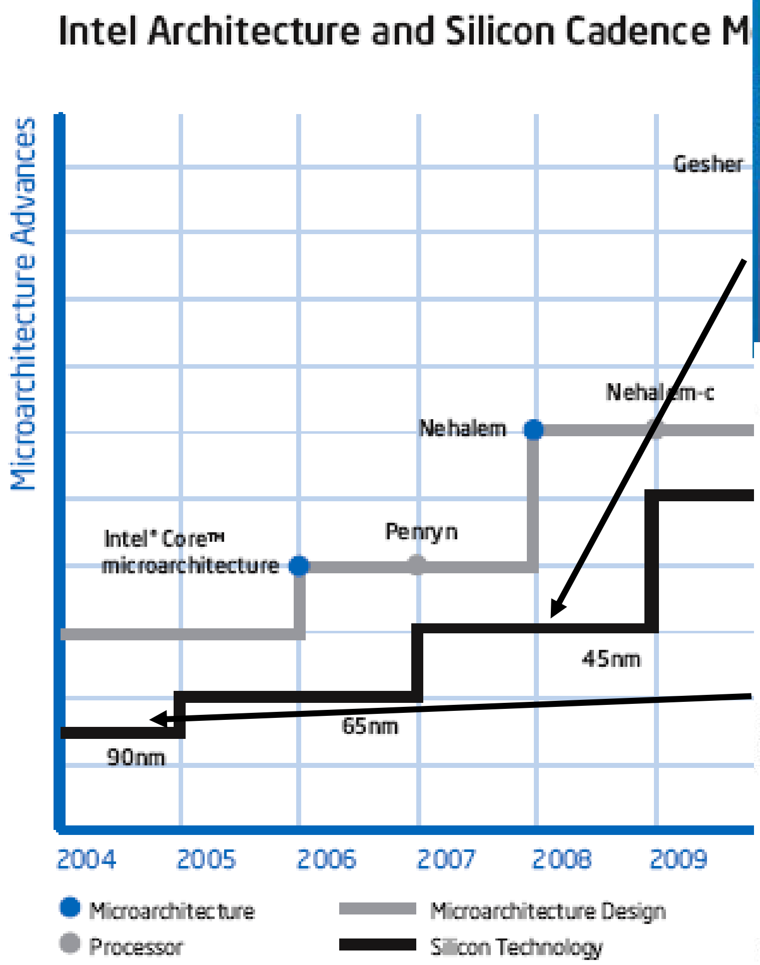
–1 MB Inst. L2

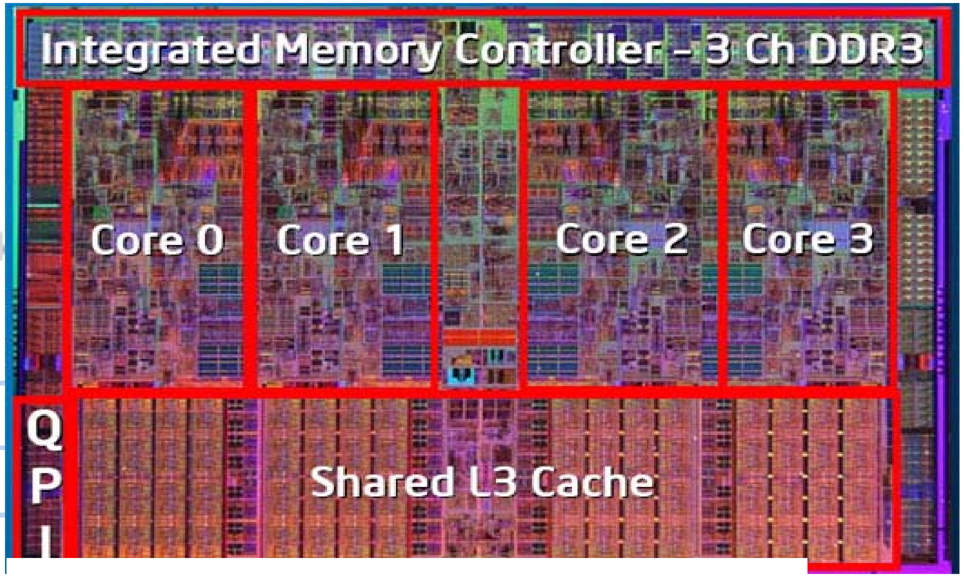
–256 KB Data L2

–12 MB L3

Memories: main victims of variability

Larger and more dense on-chip memories



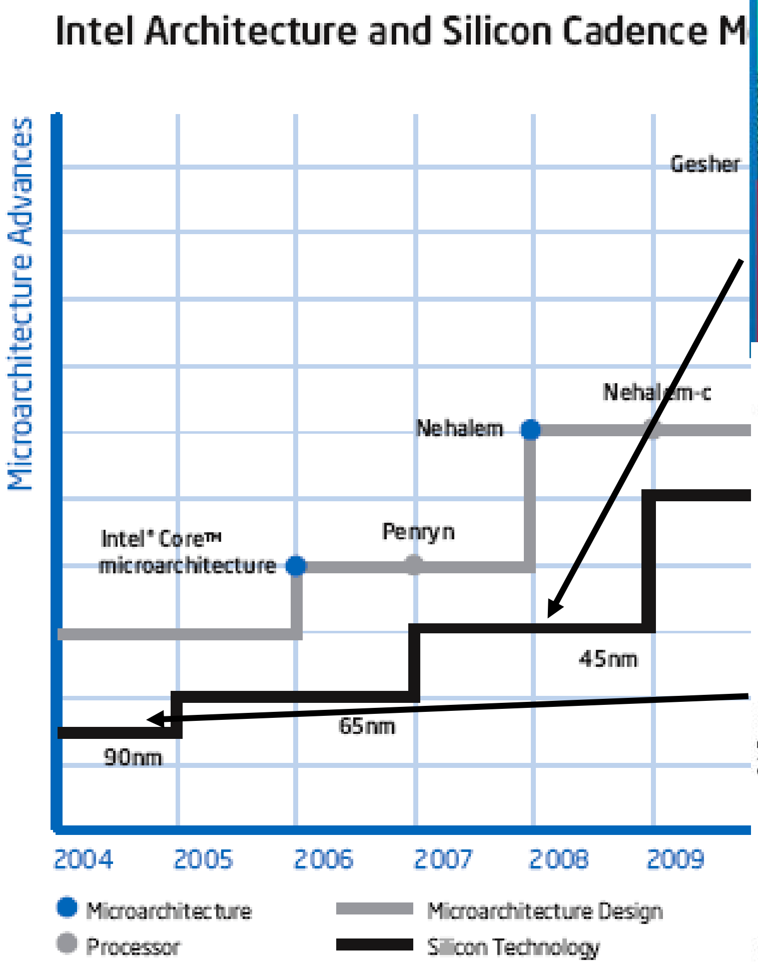


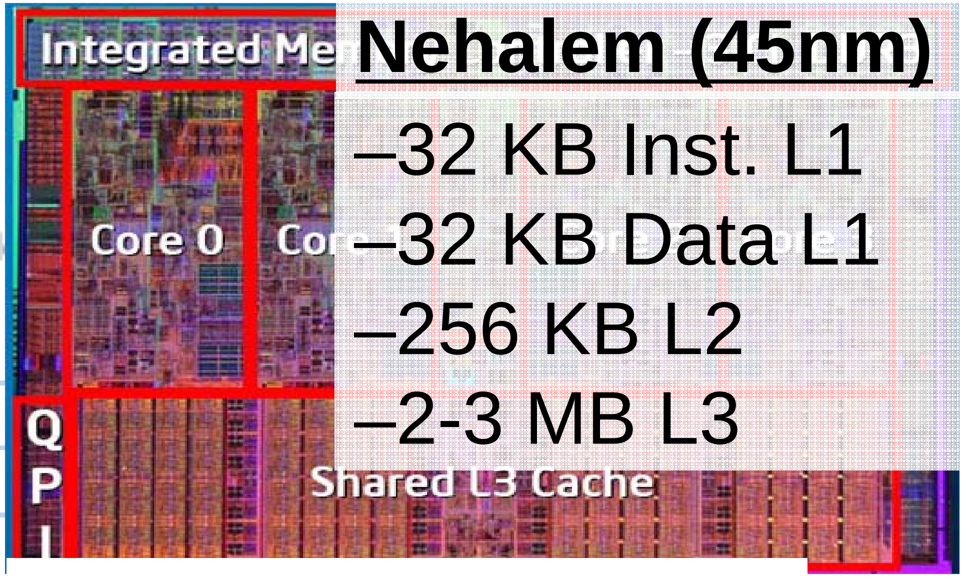
Itanium (90nm)

- –16 KB Inst. L1
- –16 KB Data L1
- –1 MB Inst. L2
- –256 KB Data L2
- –12 MB L3

Memories: main victims of variability

Larger and more dense on-chip memories





Nehalem (45nm)

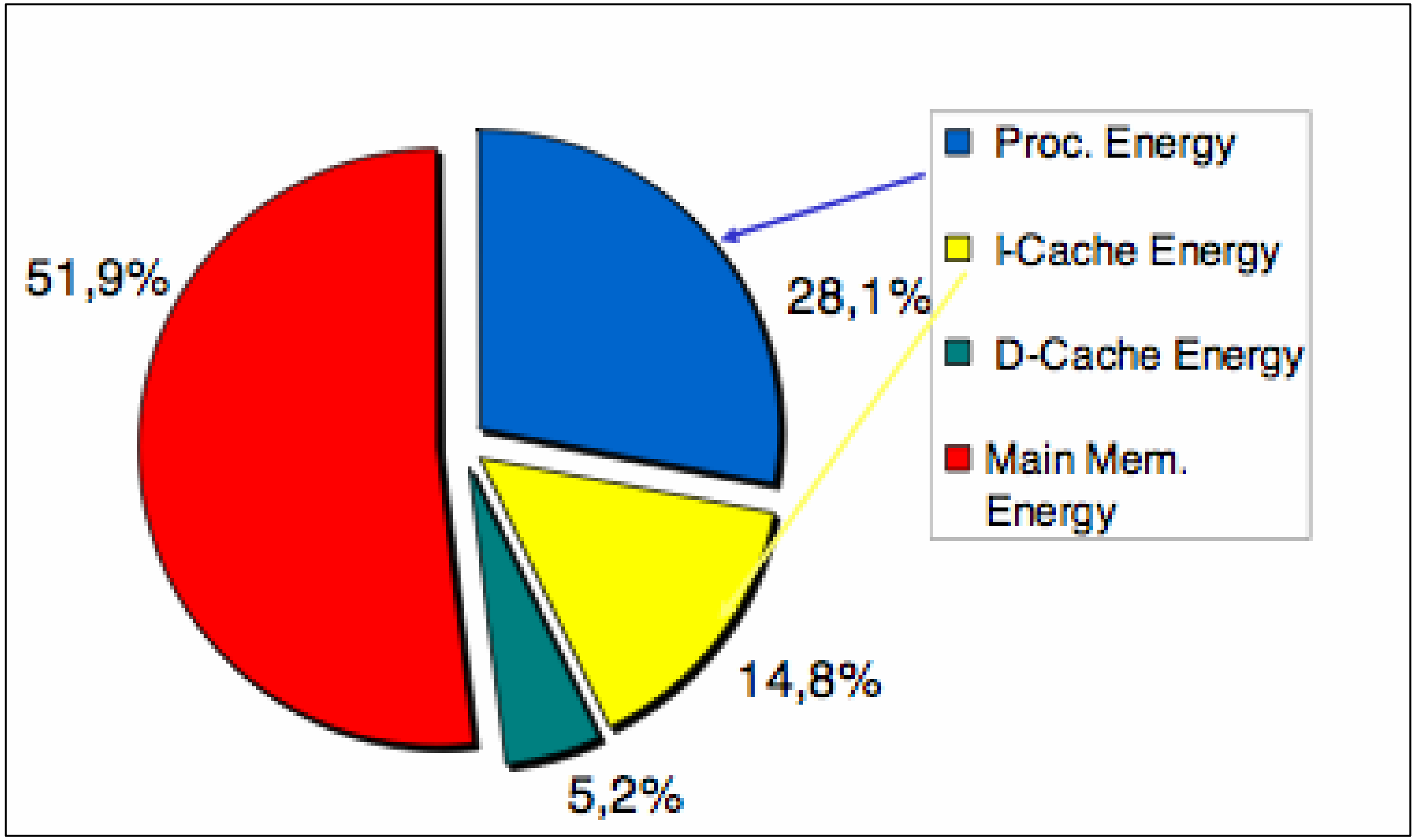
- –32 KB Inst. L1
- –32 KB Data L1
- –256 KB L2
- –2-3 MB L3

Itanium (90nm)

- –16 KB Inst. L1
- –16 KB Data L1
- –1 MB Inst. L2
- –256 KB Data L2
- –12 MB L3

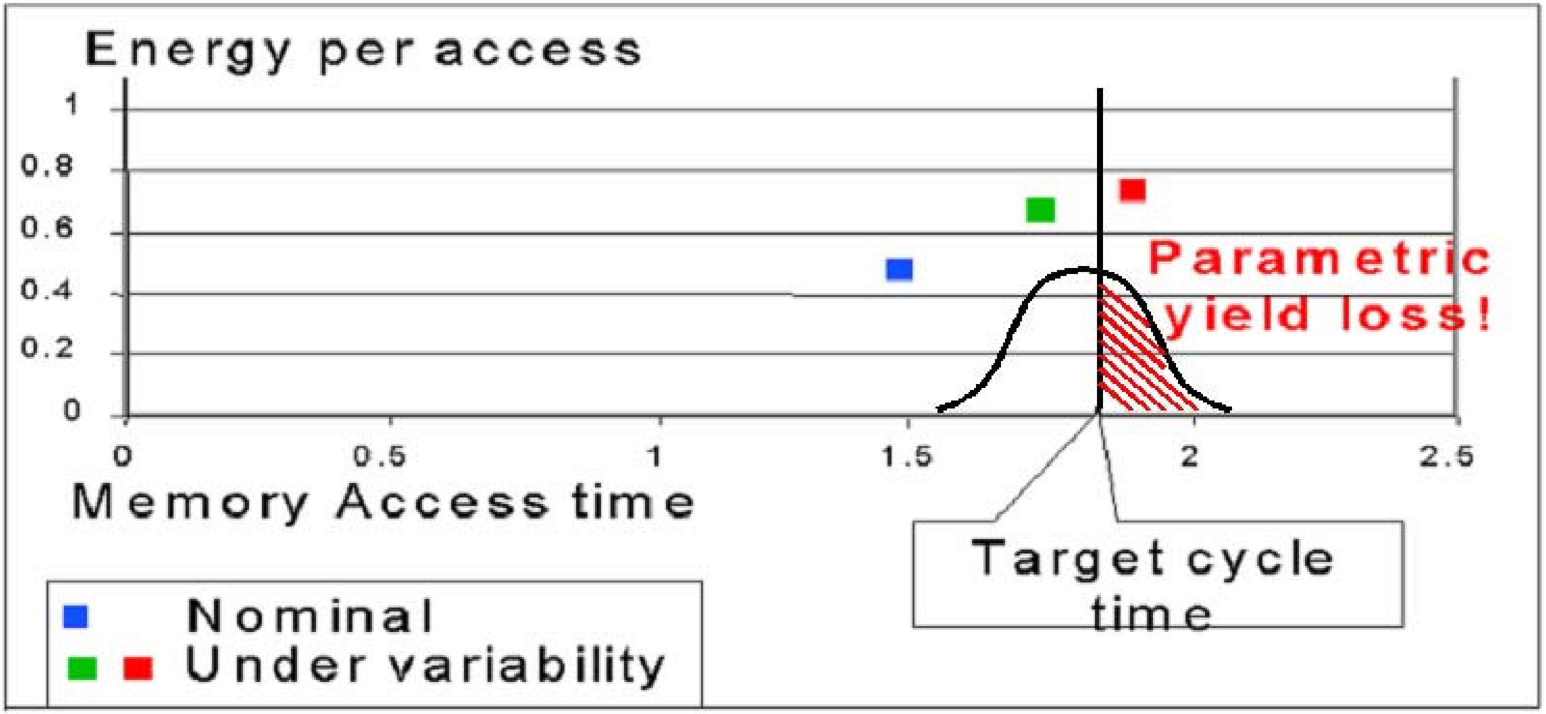
Memories: main victims of variability

Memories consume a large portion of the energy budget



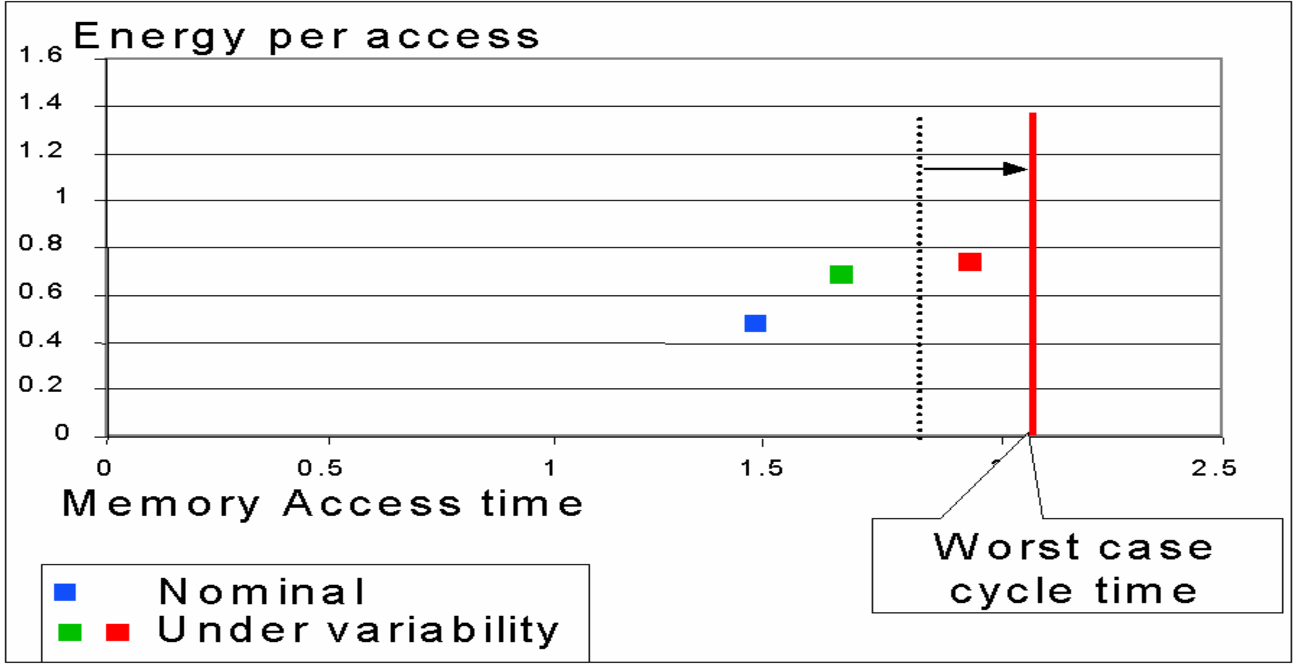
Uncertainty at platform level

- Uncertainty generated by process variation
- Platform is no longer static
- Energy/delay values are larger than expected at design time → Inoperative memories



State-of-art

- **Work done to tackle with uncertainty at system level, mostly focused on processor**
  - □ Worst case design techniques
    - Lost performance
    - Energy overhead



State-of-art

- **Work done to tackle with uncertainty at system level, mostly focused on processor**
  - Worst case design techniques
    - Lost performance
    - Energy overhead
  - Circuit techniques: DVS, body bias
  - Micro-architecture techniques: Razor

Memories require specific techniques to deal with variation

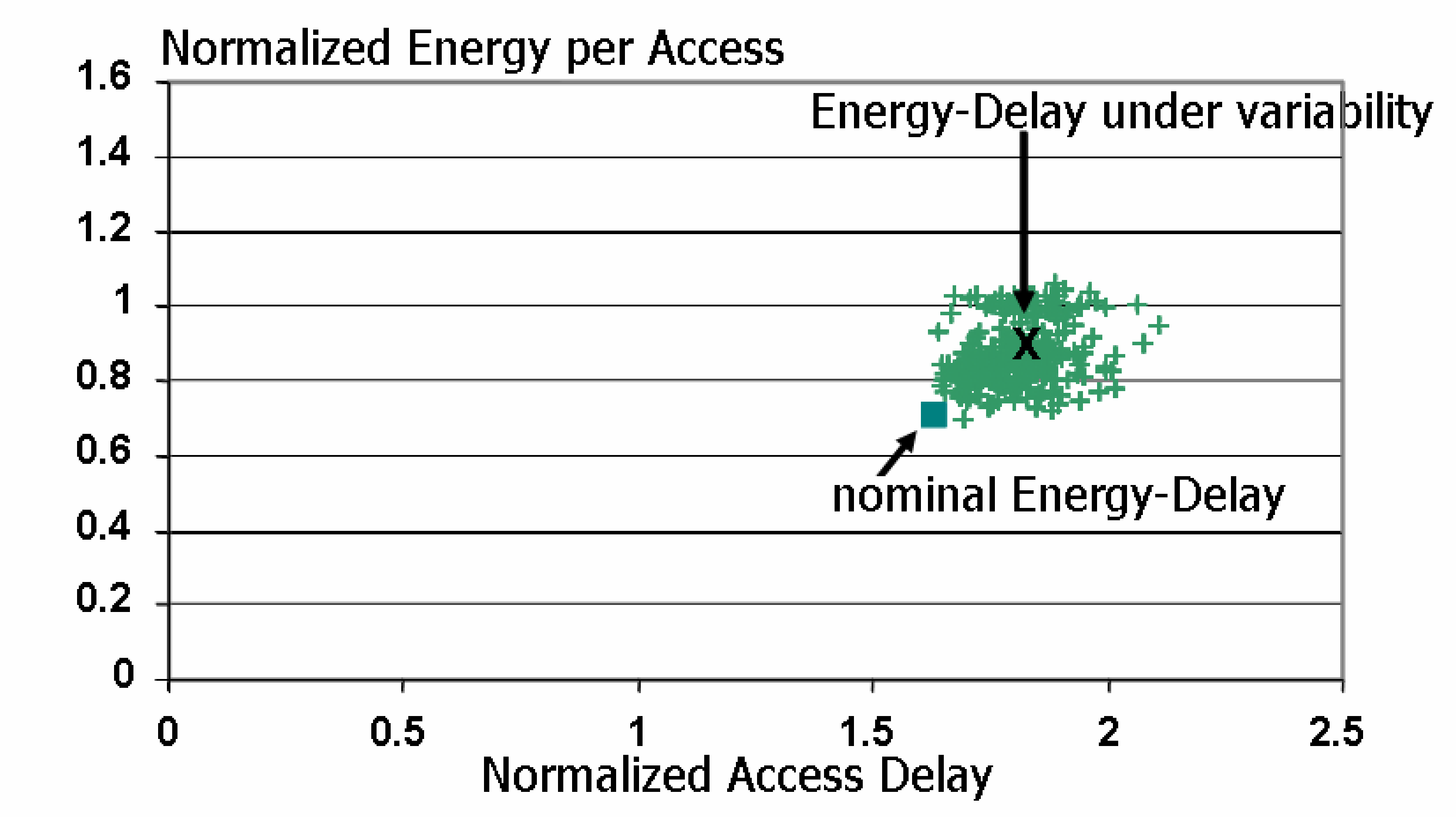
- **Meeting performance constraints while energy consumption is kept low**

Outline

- Motivation
- **Multimode memories**
- Methodology
- Scalability
- Results

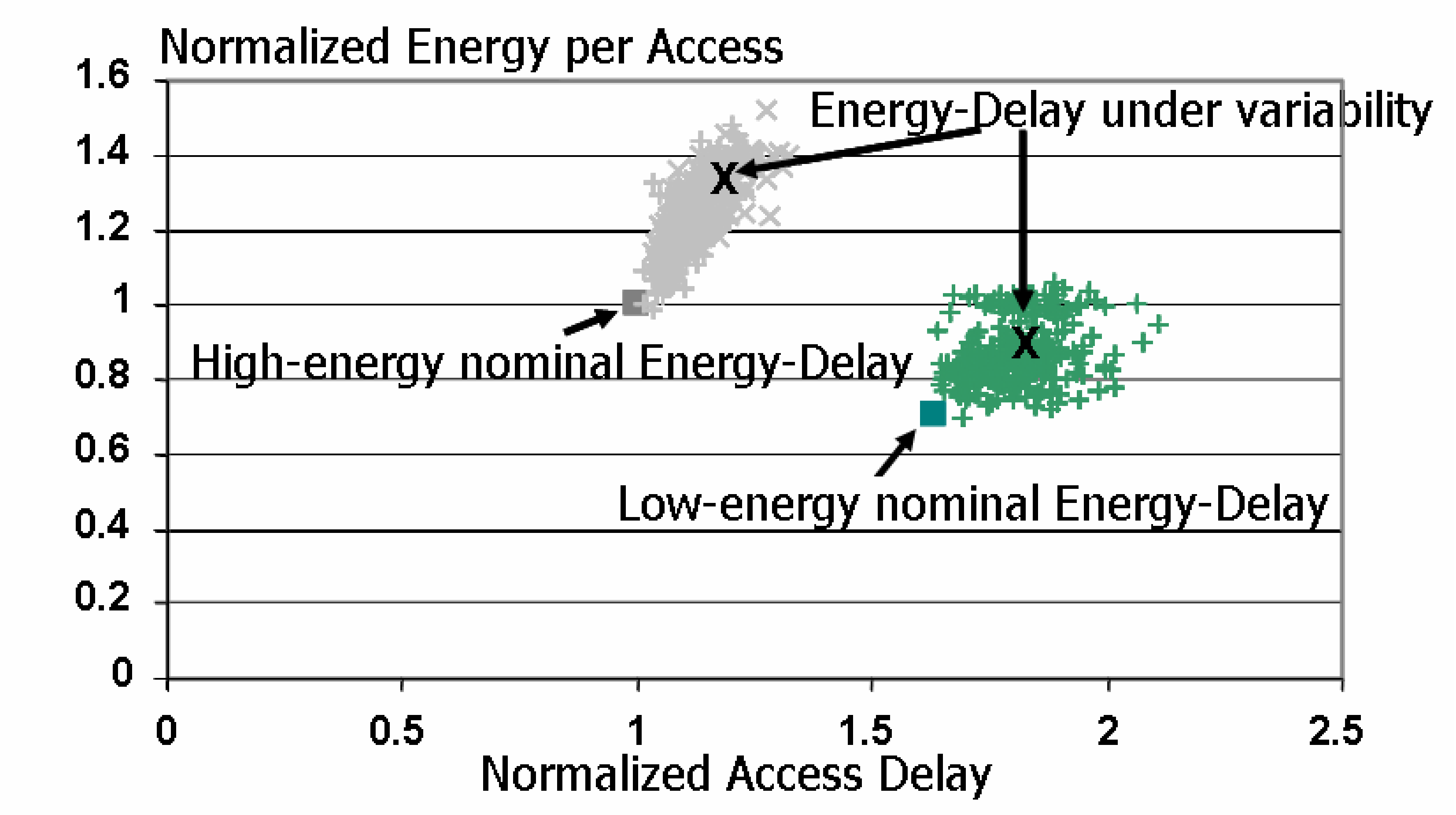
Multimode memories

Multimode memories increase system adaptability



Multimode memories

Multimode memories increase system adaptability



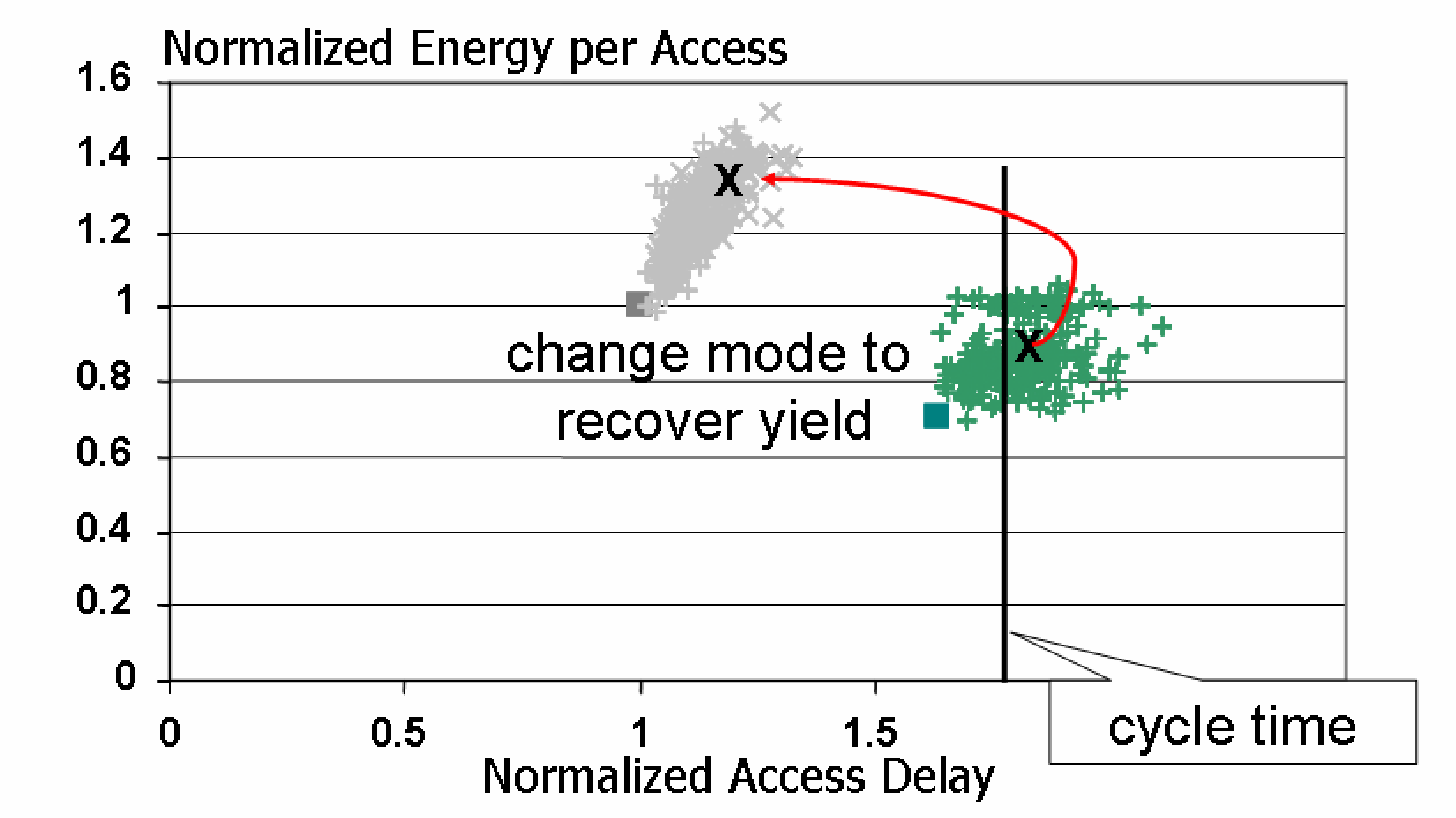
Energy-Delay under variability

High-energy nominal Energy-Delay

Low-energy nominal Energy-Delay

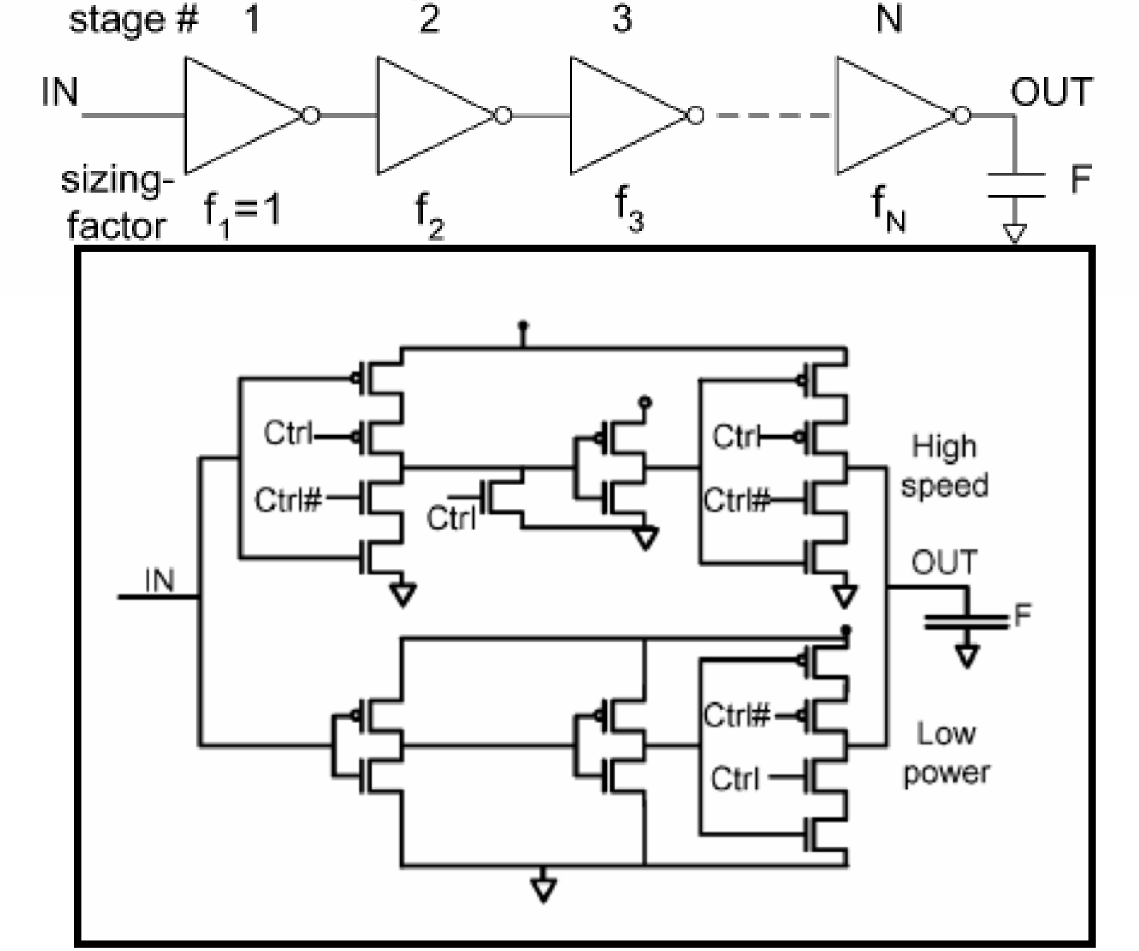
Multimode memories

Multimode memories increase system adaptability



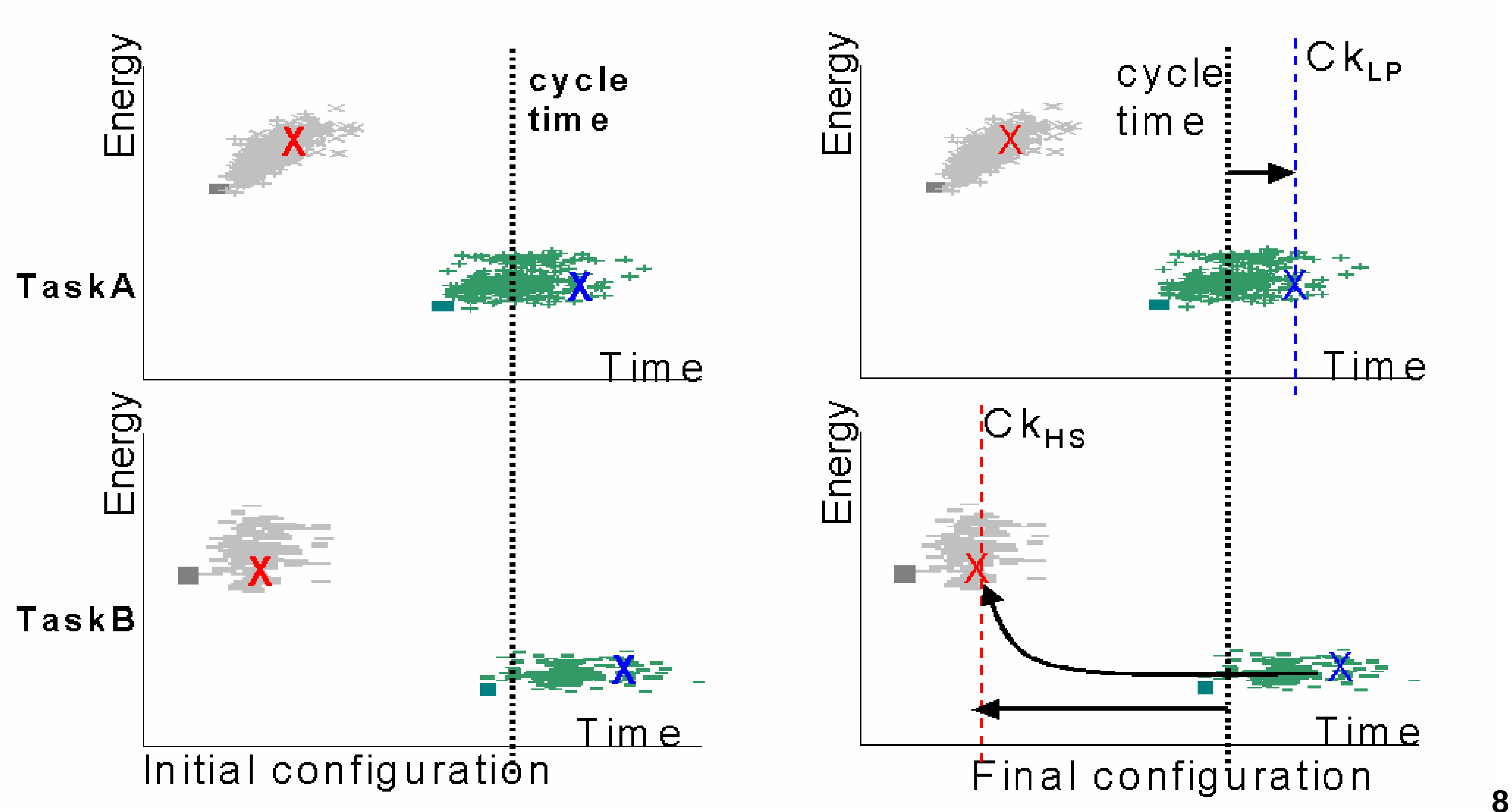
Multimode memories

- Compensation buffer: Same voltage, different operating points **(H. Wang, M. Miranda, A. Papanikolaou, F. Catthoor "Variable tapered Pareto buffer design and implementation techniques allowing runtime configuration for low power embedded SRAMs" IEEE Trans. VLSI 2005)**



Multimode memories

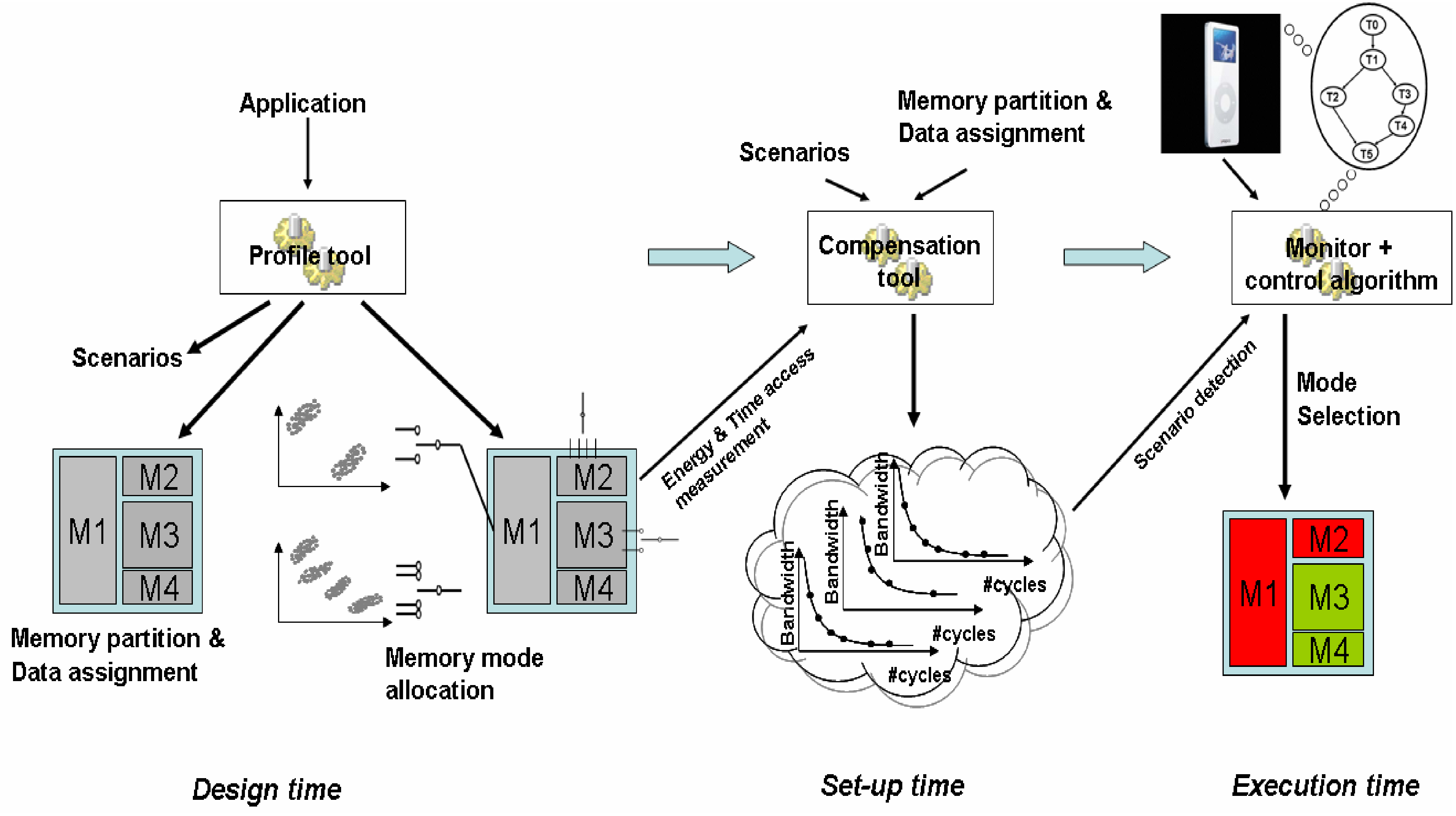
- ***Dynamic adaptation of memory mode***



Outline

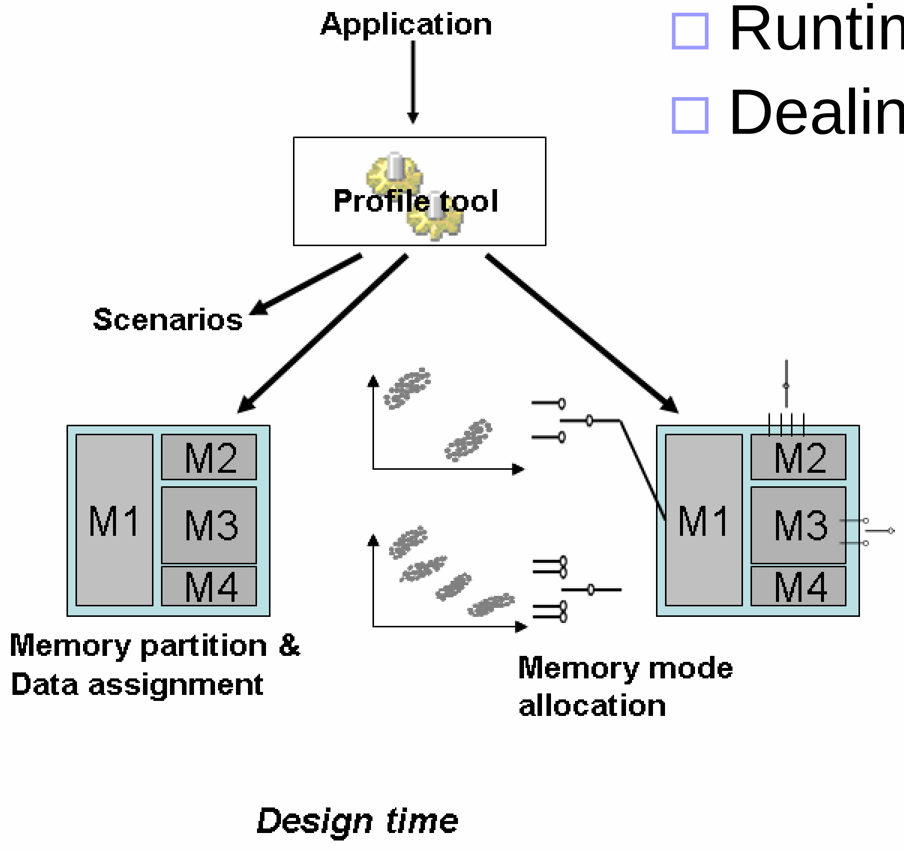
- Motivation
- Multimode memories
- **Methodology**
- Scalability
- Results

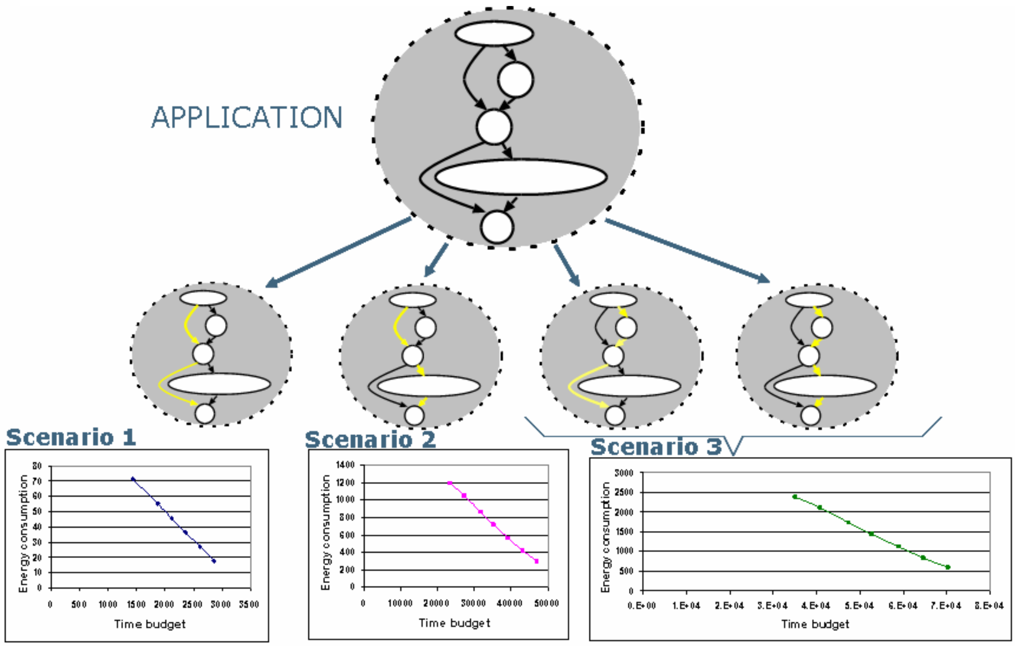
Methodology



Methodology: Design time

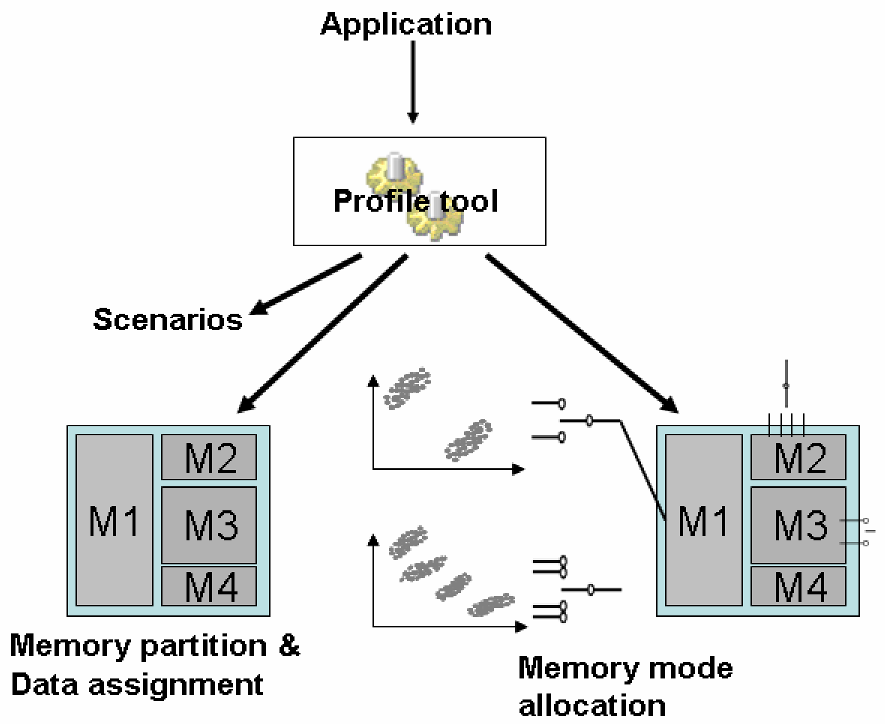
- ***Application scenario***
  - □ Design time characterization based on workload
  - □ Runtime identification
  - □ Dealing with application dynamism



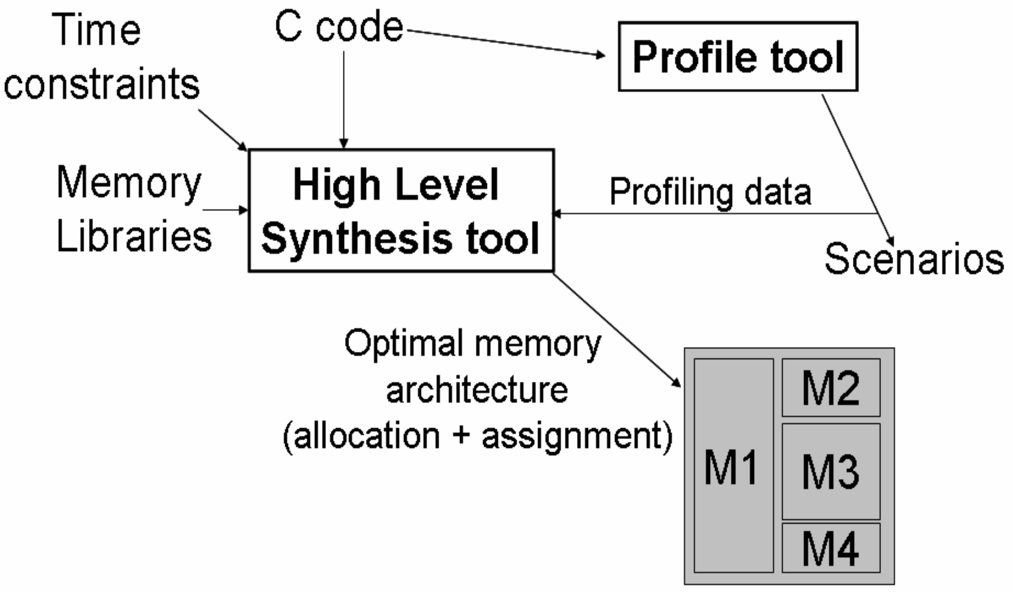


Methodology: Design time

- ***Energy-efficient Memory partition and Data assignment***
- ***Memory mode allocation***

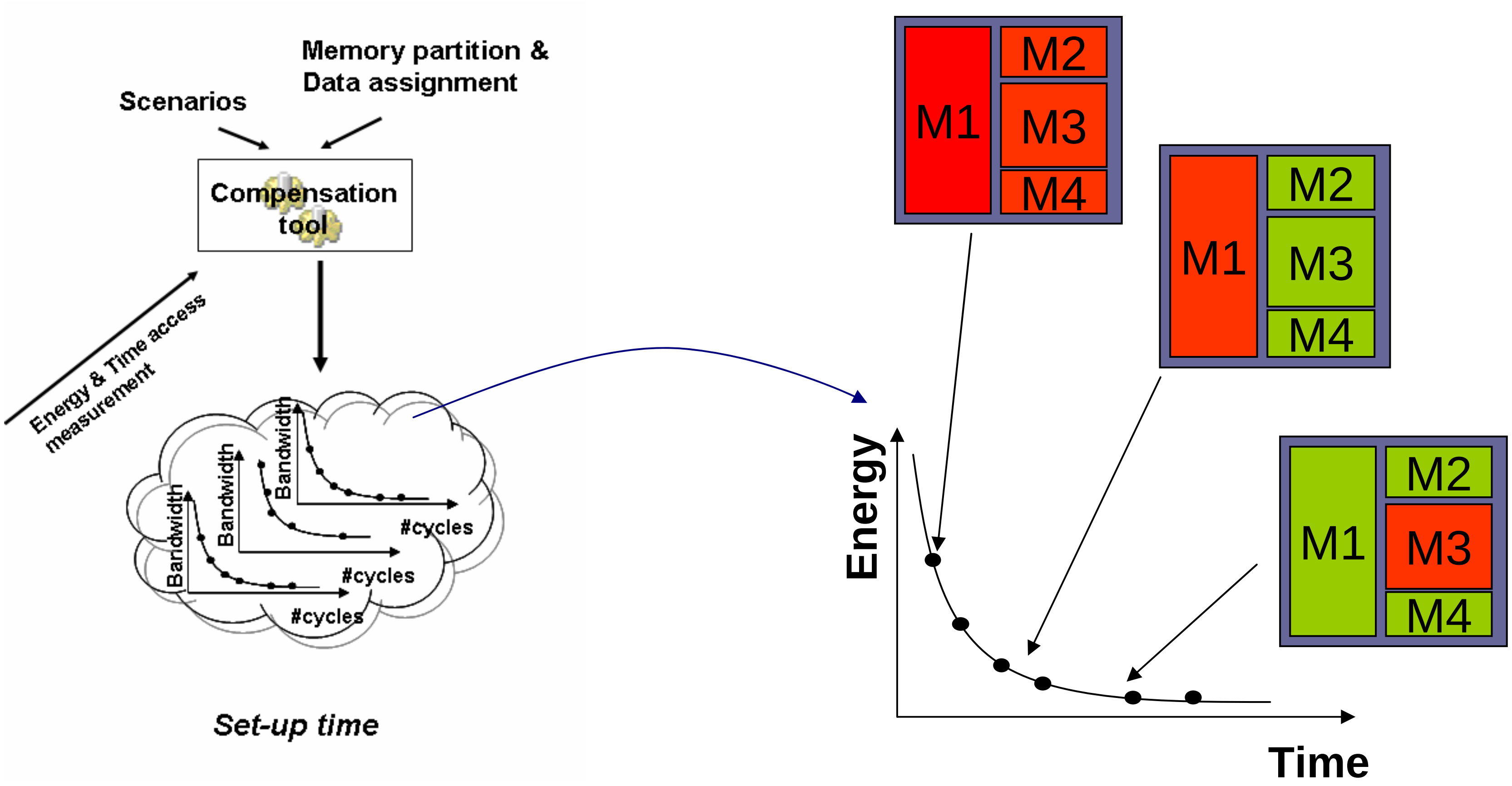


Design time

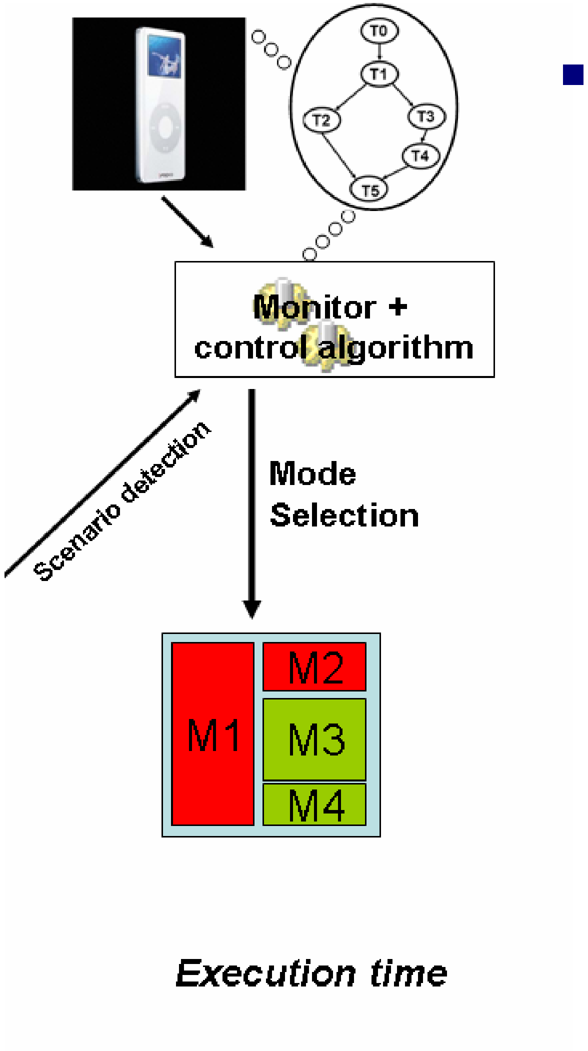


Methodology: Set-up time

- ***Pareto curve generation based on scenario***
  - □ Measurement of actual memory parameters



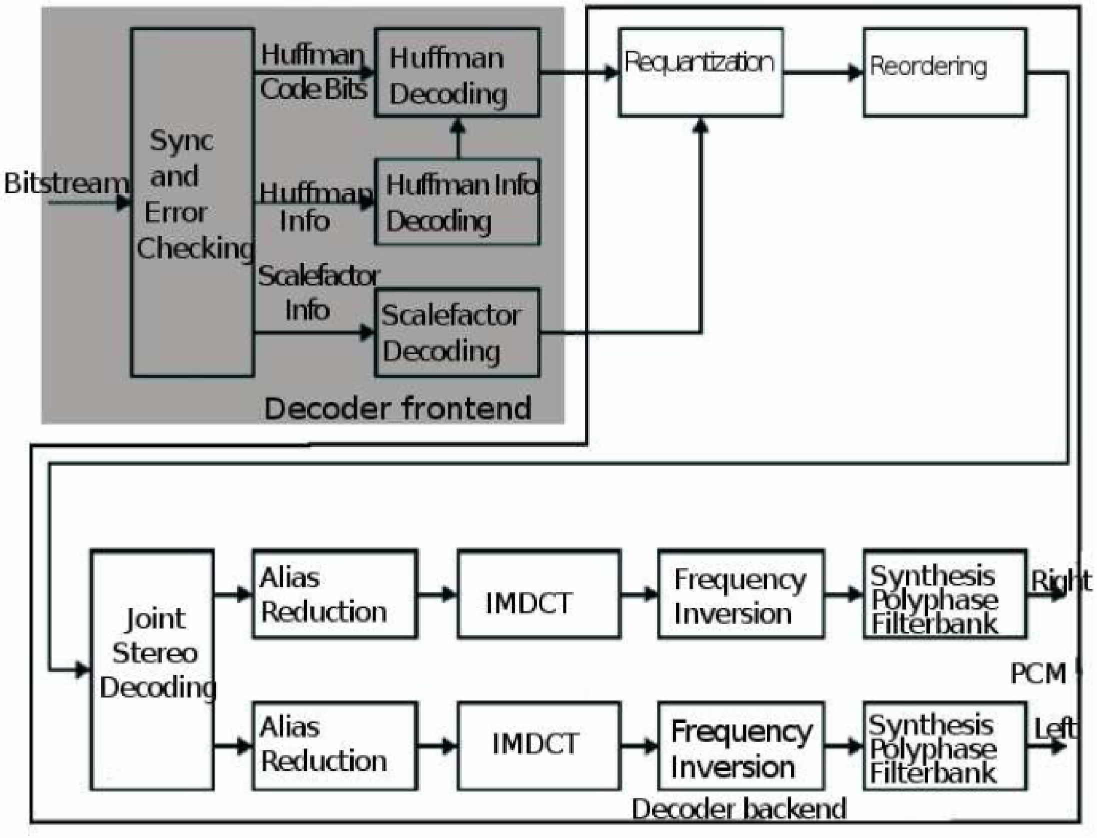
Methodology: Execution time



- ***Memory reconfiguration***
  - □ Application monitoring
    - □ Current scenario detection
  - □ Memory calibration when necessary
    - □ Switch to pre-stored memory mode

Experimental environment

- MP3 Decoder
  - □ Original code without scenarios (worst case)
  - □ Optimized code using scenarios
  - □ Multitasked implementation



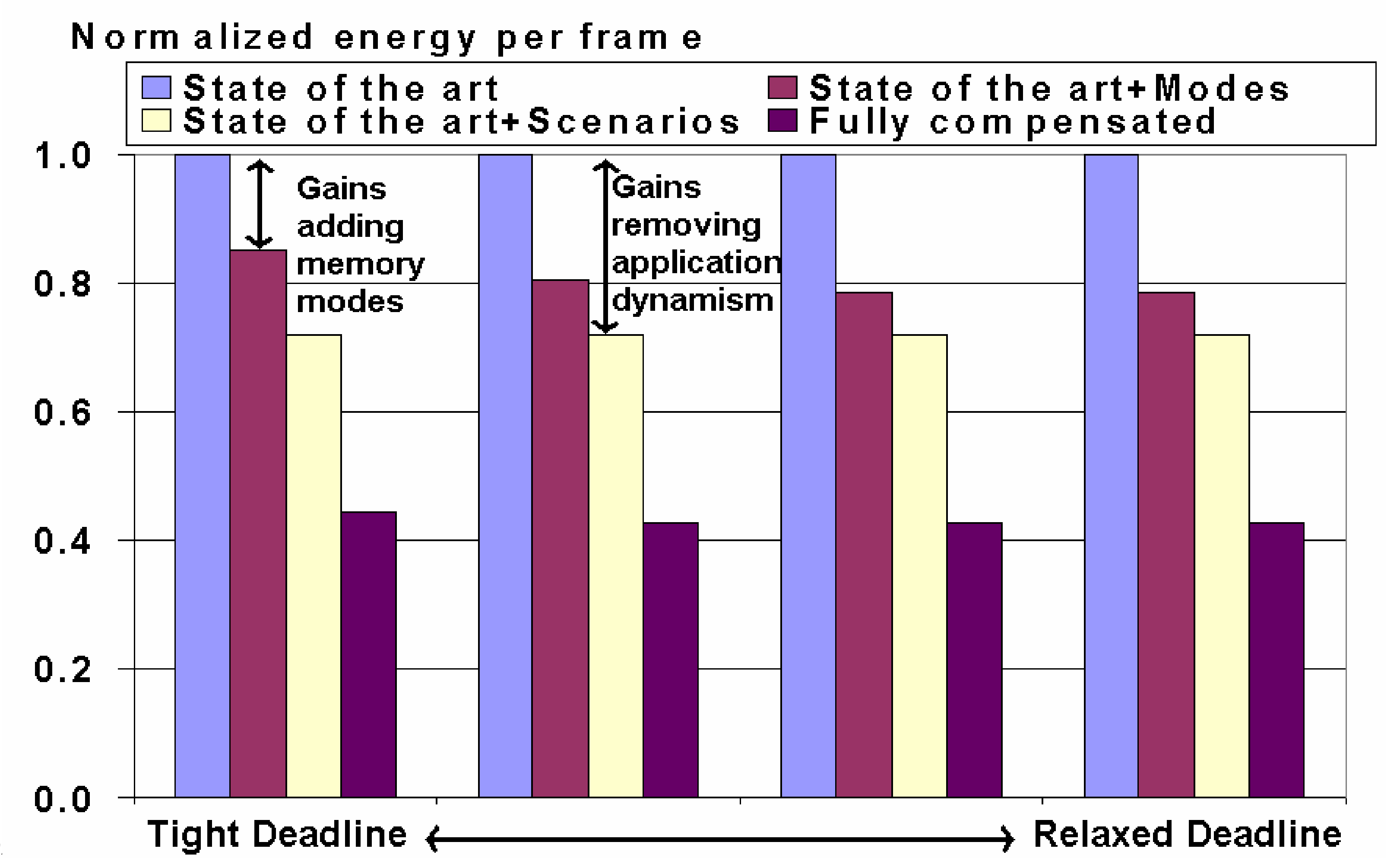
Experimental environment

- MP3 Decoder
  - □ Optimized code using scenarios
  - □ Original code without scenarios (worst case)
  - □ Multitasked implementation

- Memory architecture
  - □ Memory partition: 10 memories (4, 8,16 and 32 KB)
  - □ Energy aware data assignment
  - □ 2/4/8 modes per memory

Methodology results

Compensation methodology saves up to 60% of energy keeping constraints! (considering 2 modes per memory)

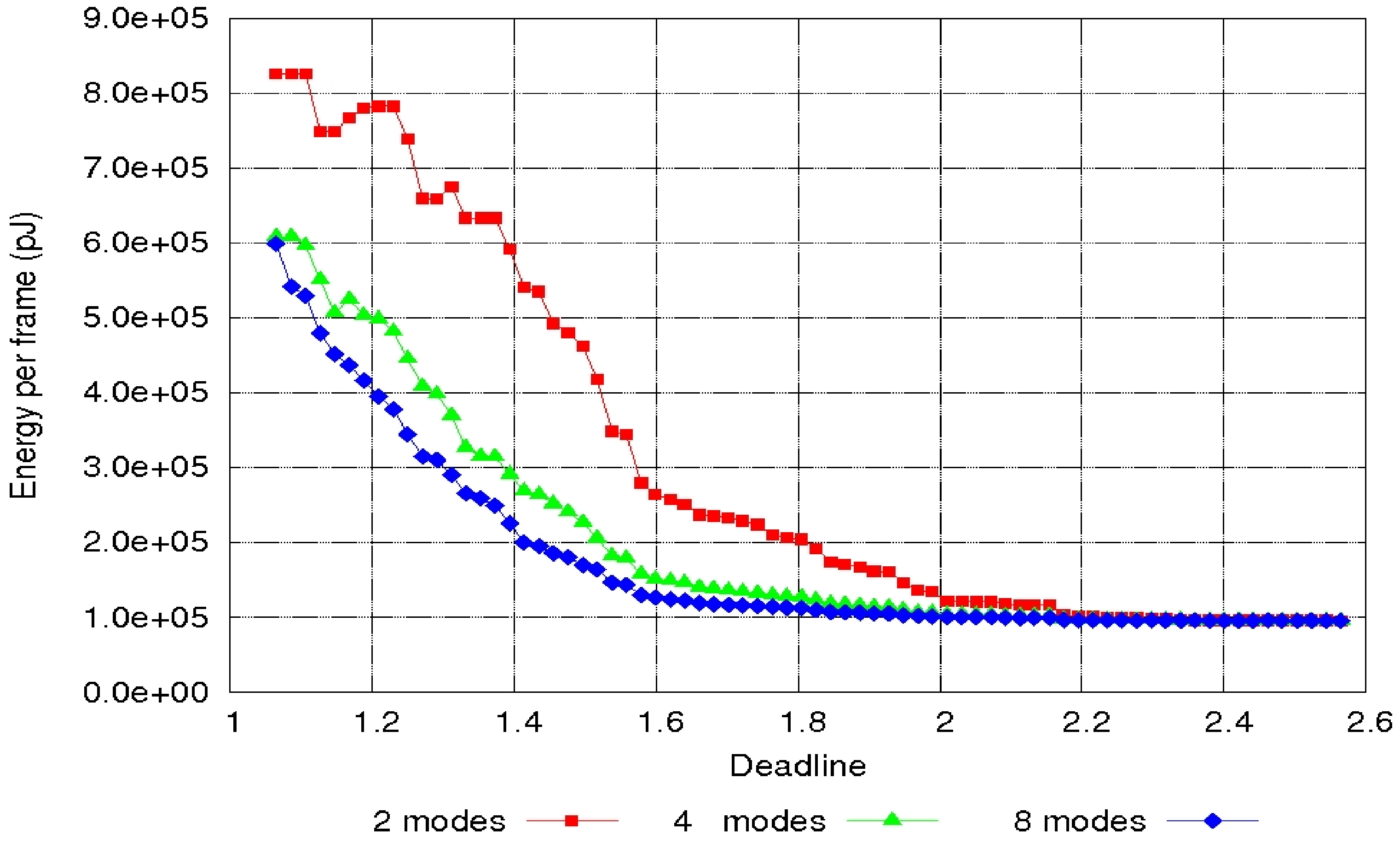


Outline

- Motivation
- Multimode memories
- Methodology
- **Scalability**
- Results

Scalability of memory modes

- Assuming all memories with same number of modes
- More modes mean more energy savings
- Energy reduction is not proportional to #modes



Scalability: area and complexity problems

- The number of modes impacts on
  - □ Set-up time: increases the execution time of control algorithms
  - □ Area

- We need to trade-off energy savings, area and algorithm complexity
  - □ Heterogeneous mode allocation

Outline

- Motivation
- Multimode memories
- Methodology
- Scalability
- **Results**

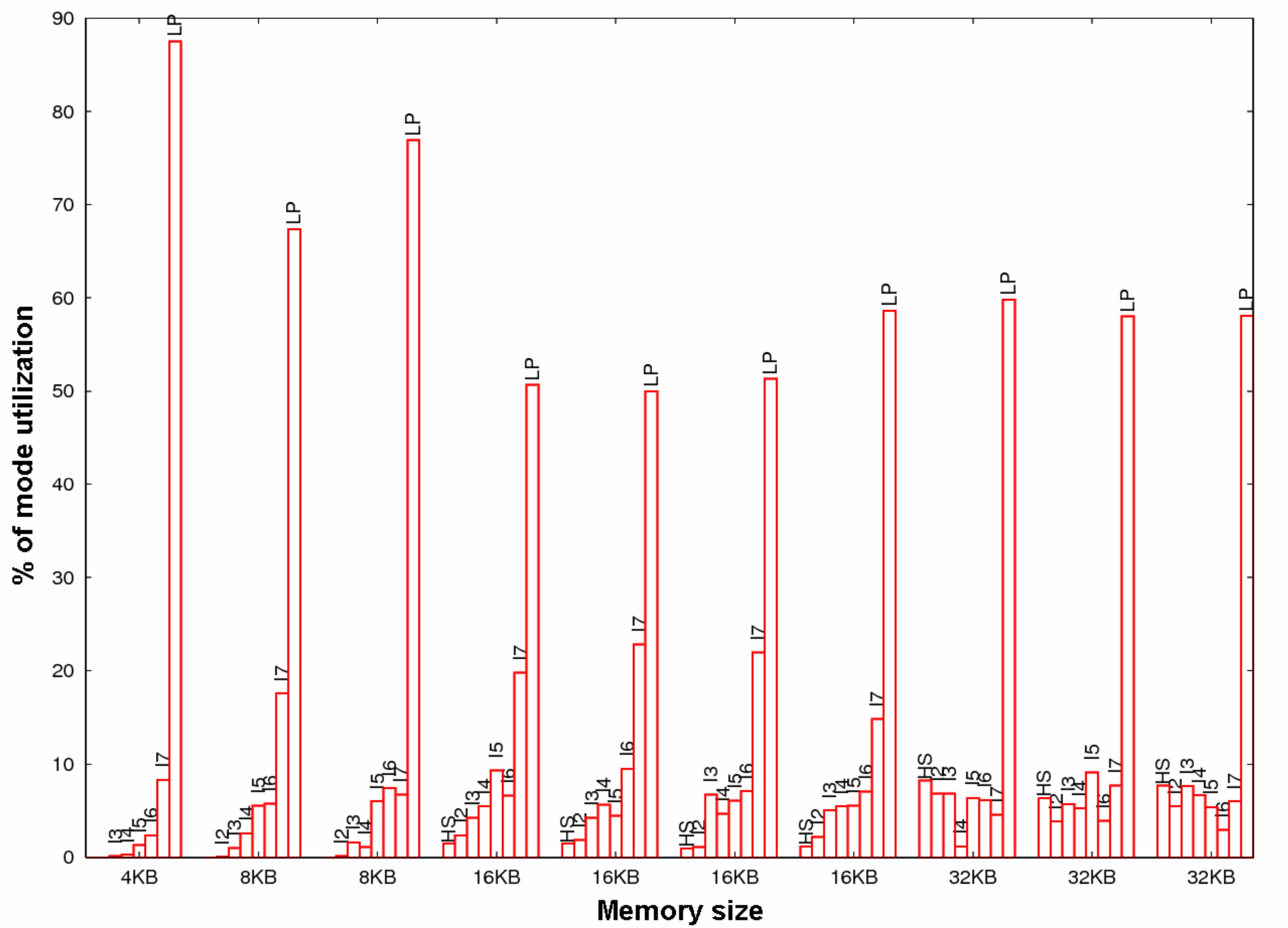
Heterogeneous mode allocation

How to choose the right distribution?

Criteria to add extra modes:

- Based on size → large memories
- Based on data allocation → most accessed memories

Heterogeneous mode allocation



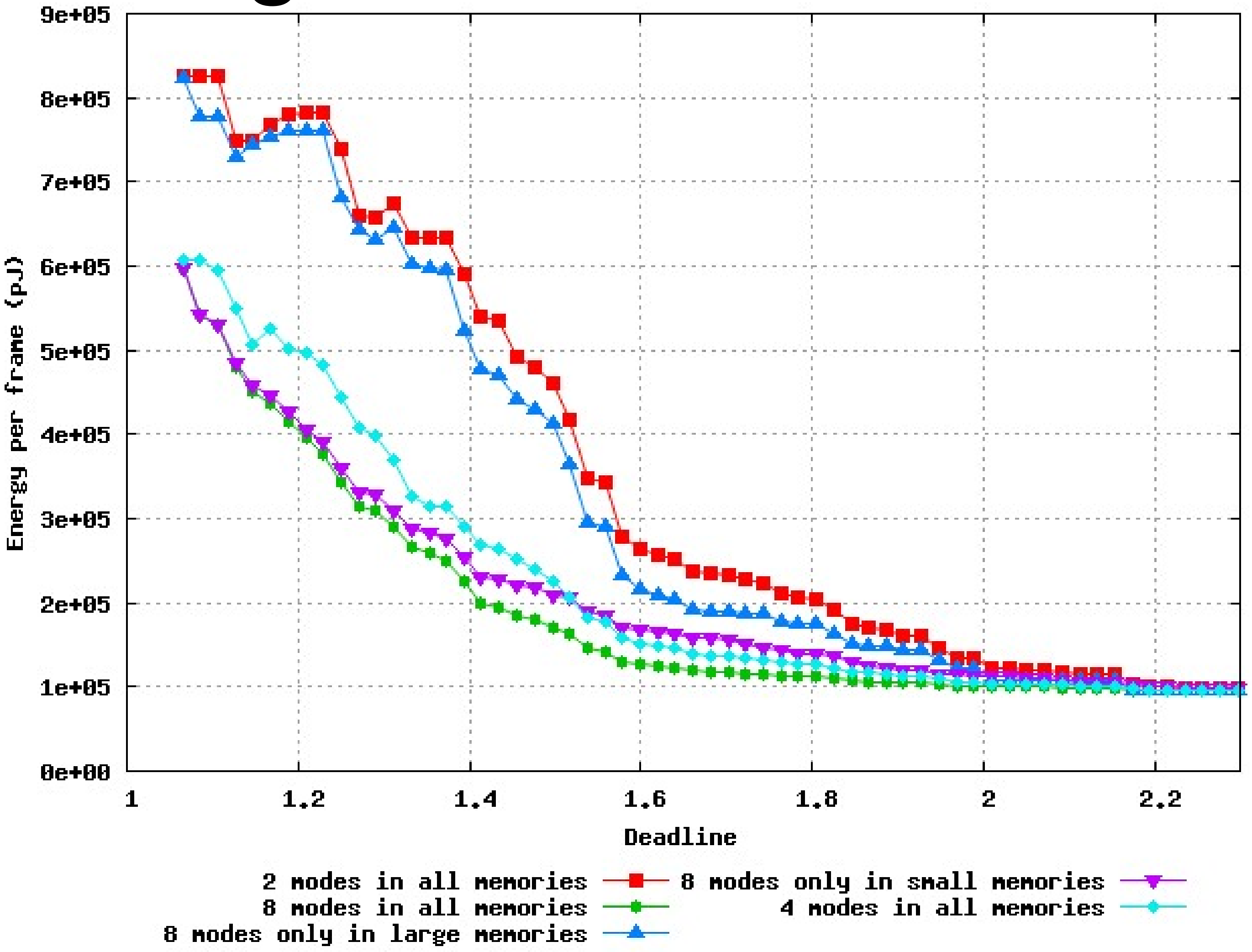
- **Criteria 1: Allocate more modes on large memories**

Heterogeneous mode allocation

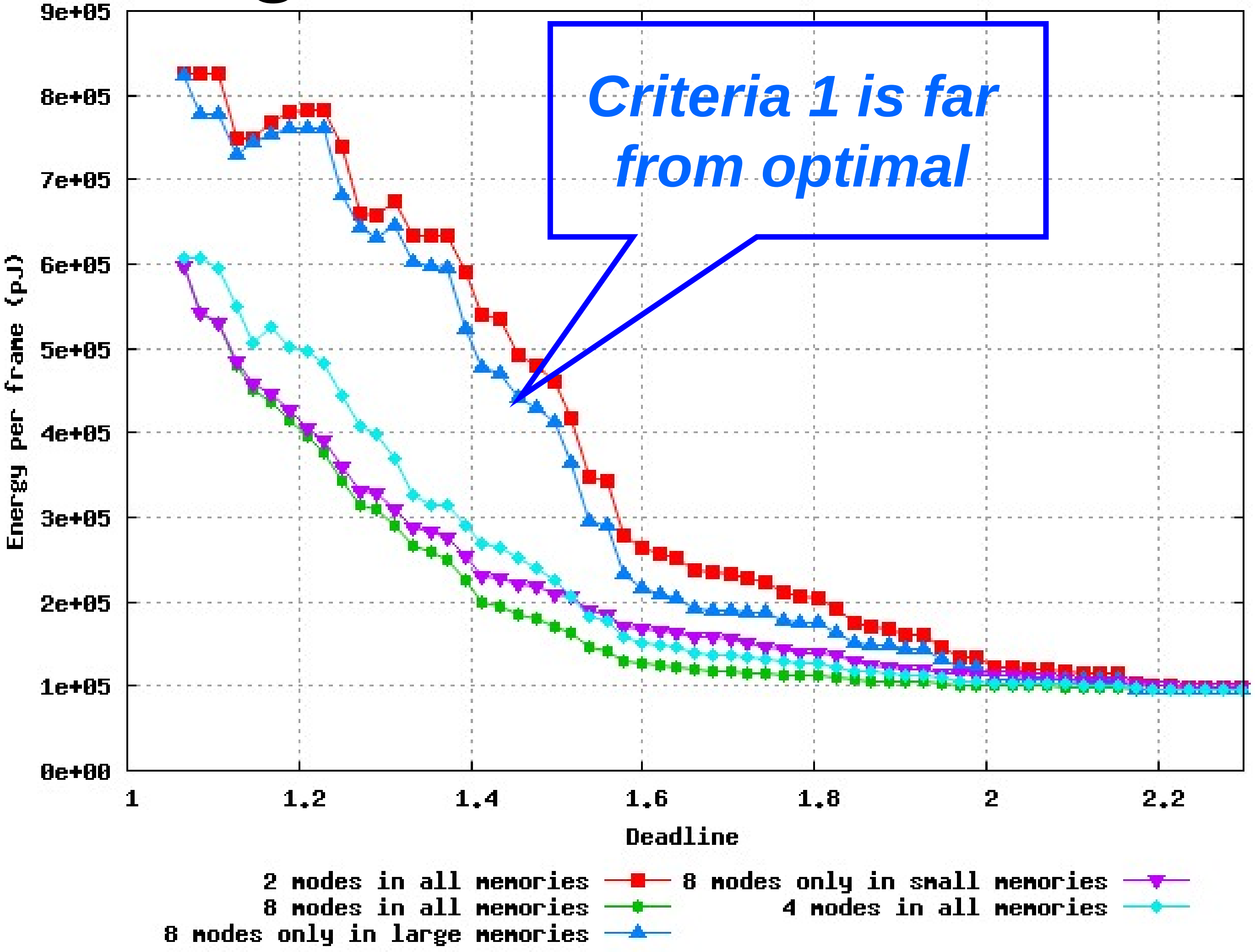
Memory size	#memories	Freq. Access
4KB	1	50.6%
8KB	2	42%
16KB	4	5.4%
32KB	3	2%

- **Criteria 2: Allocate more modes to most accessed memories**

Heterogeneous mode allocation

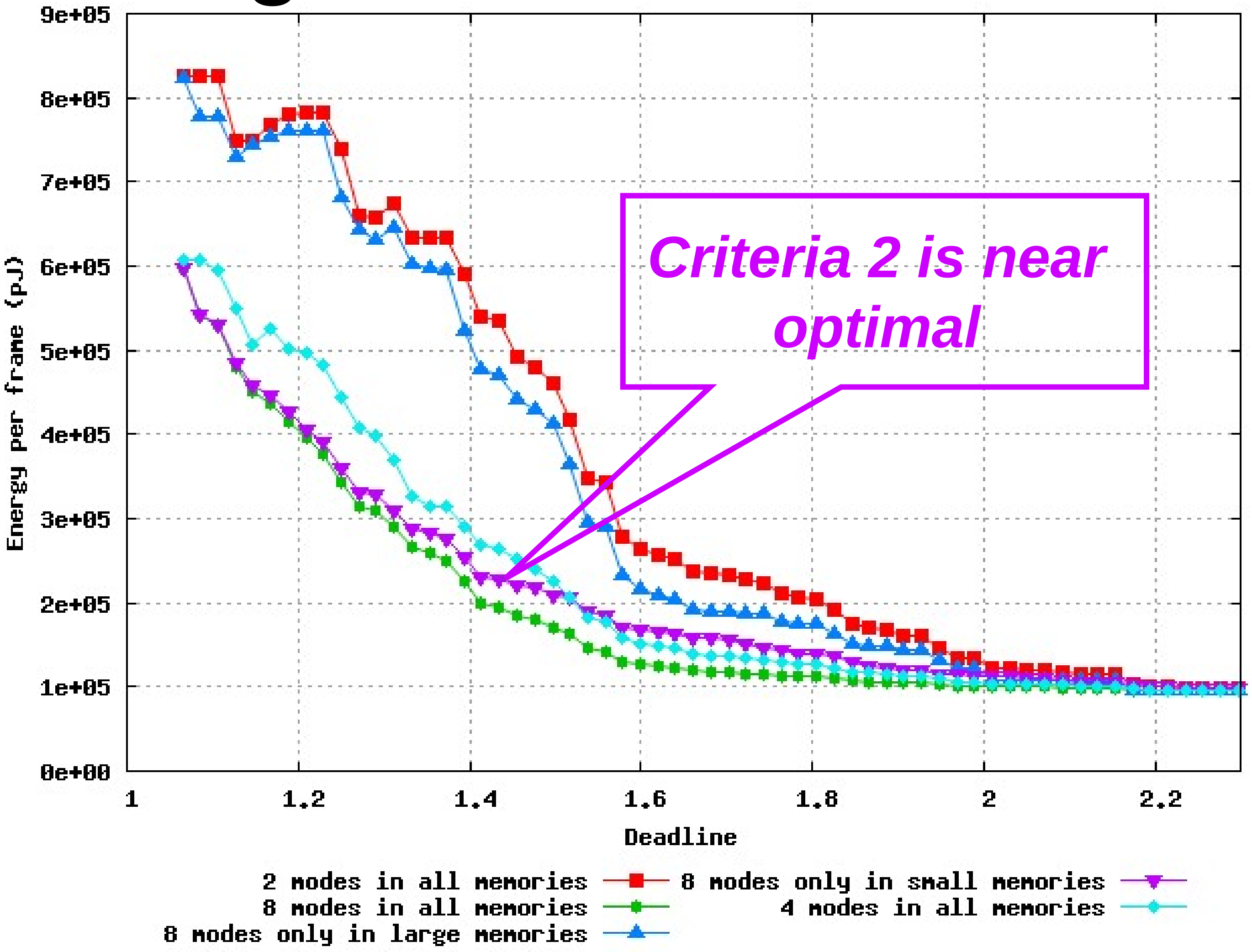


Heterogeneous mode allocation



Criteria 1 is far from optimal

Heterogeneous mode allocation



Criteria 2 is near optimal

Conclusions

❑Memory mode selection is closely related to data assignment

❑Heterogeneous mode allocation:

- Has almost no impact on memory area
- Reduces algorithm complexity

Future steps

- Could we move work from set-up time to design time?
- Tackle effects of aging and temperature

Thanks!!!

Questions?