# Adaptive Mode Control: A Static-Power-Efficient Cache Design

Huiyang Zhou, Mark Toburen, Eric Rotenberg, Tom Conte

Center for Embedded Systems Research (CESR)
Department of Electrical & Computer Engineering
North Carolina State University
www.tinker.ncsu.edu

# Technology Trends

- Trends
  - Lower threshold voltages in deep sub-micron technologies
    - ◆ increases leakage current (sub-threshold current)
    - ◆ increases *static power dissipation*
  - Large fraction of die area occupied by on-chip caches
    - 60% of StrongARM die area is cache

# Circuit Support

- Circuit-level solution
  - SRAM cells with low-leakage operating modes
  - Insert transistors between $V_{dd}$ and Gnd rails
  - Isolating cells from power rails puts them in **sleep mode** (reduces leakage current)

# Architectural Support

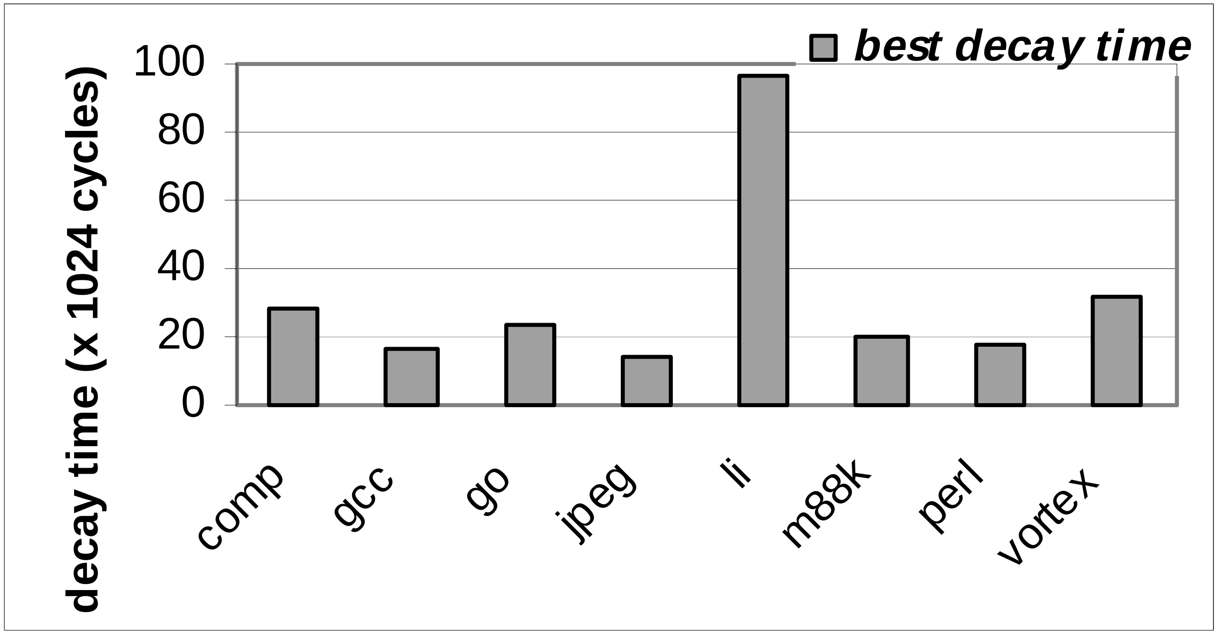
- Circuit-level technique must be controlled at the architecture level
  - Data stored in sleeping cell is unreliable or lost
  - Maximize number of sleep-mode lines while preserving performance
    - Caches tradeoff efficiency for robustness
    - Deactivate (put into sleep mode) unused cache lines

# Methods for Dynamically Deactivating Cache Lines

- Related Work
  - DRI Cache [S-H Yang, et. al.]
    - Deactivate large groups of cache lines
    - Miss rate periodically compared to statically preset *miss bound*
  - Cache Line Decay [S. Kaxiras, et. al.]
    - Deactivate individual lines after a preset *decay time*
- Per-application profiling required to determine best miss bound and decay time

# The Need for Adaptivity

- Per-benchmark cache line decay times, tuned to reduce performance by no more than 4%




- Adaptive extensions to cache line decay
  - Exploiting generational behavior [S. Kaxiras, et. al.]
  - Adaptive mode control (our approach)

# Adaptive Mode Control (AMC)

- Key idea
  - Tags are always kept active
  - Know what miss rate *could* be with all cache lines active
  - Actual miss rate can be made to precisely track hypothetical miss rate

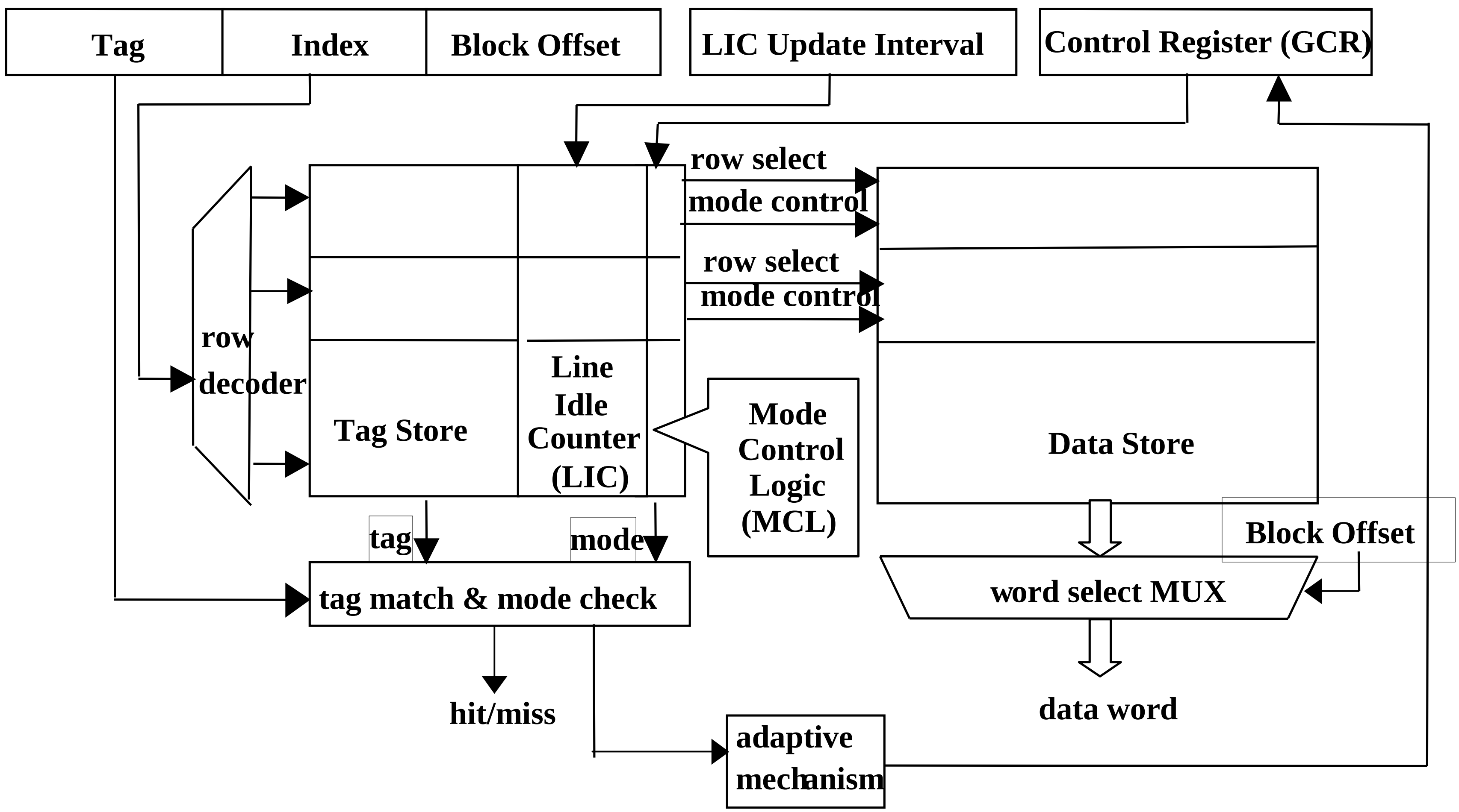
# Adaptive Mode Control (AMC)

- Can distinguish between two types of misses
  - Ideal miss
    - Tag miss
    - Would have occurred in conventional cache
  - Sleep miss
    - Tag hit, cache line in sleep mode
    - Extra miss introduced by sleep mode
- Control turn-off interval based on ratio of sleep misses to ideal misses
  - Increase turn-off interval if ratio too high
  - Decrease turn-off interval if ratio too low
  - Keep turn-off interval the same if ratio reasonable

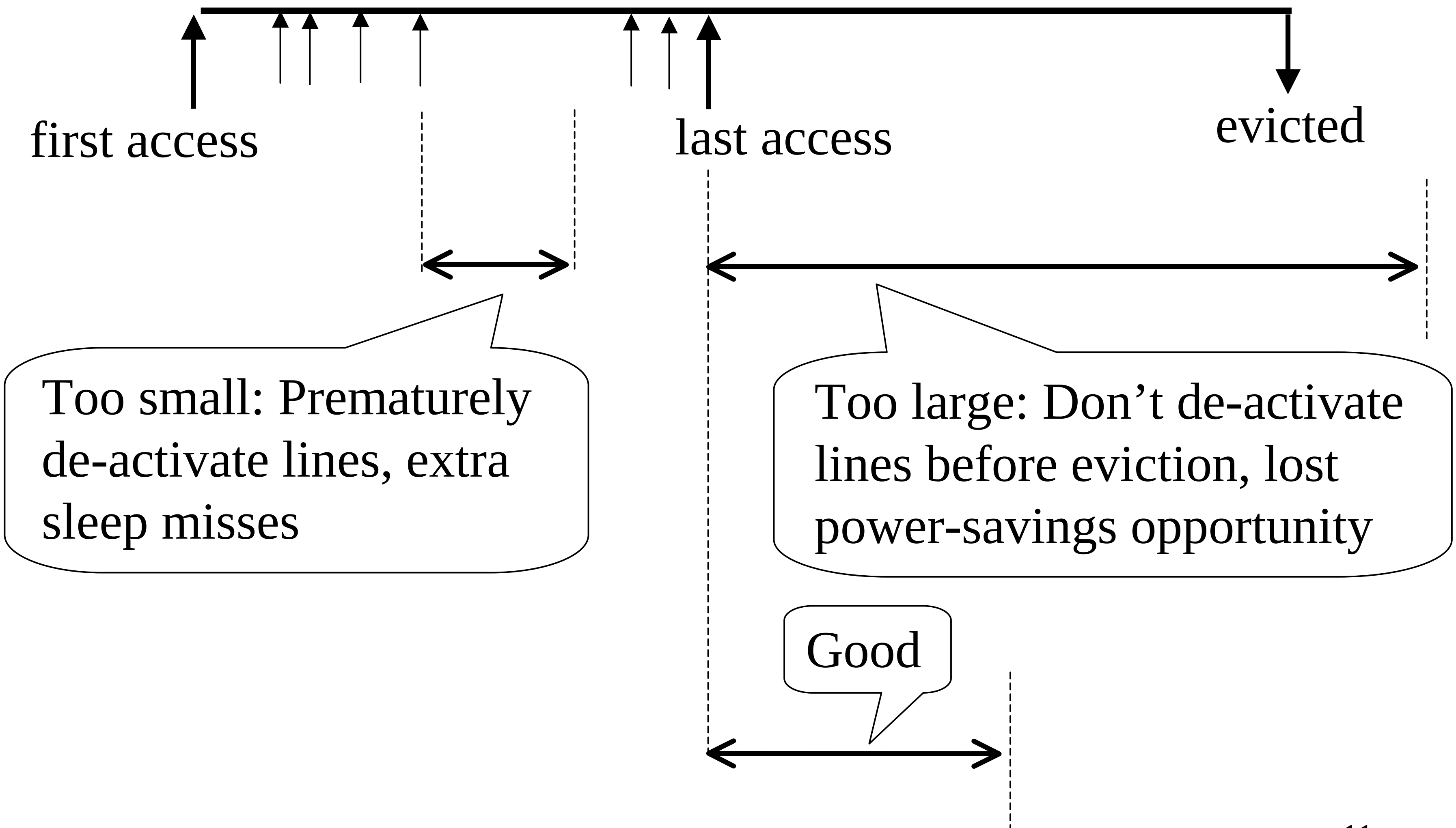
# Outline

- ✓ Introduction
- AMC cache architecture
- Adaptive mechanism (control system)
- Simulation methodology
- Results
- Conclusions

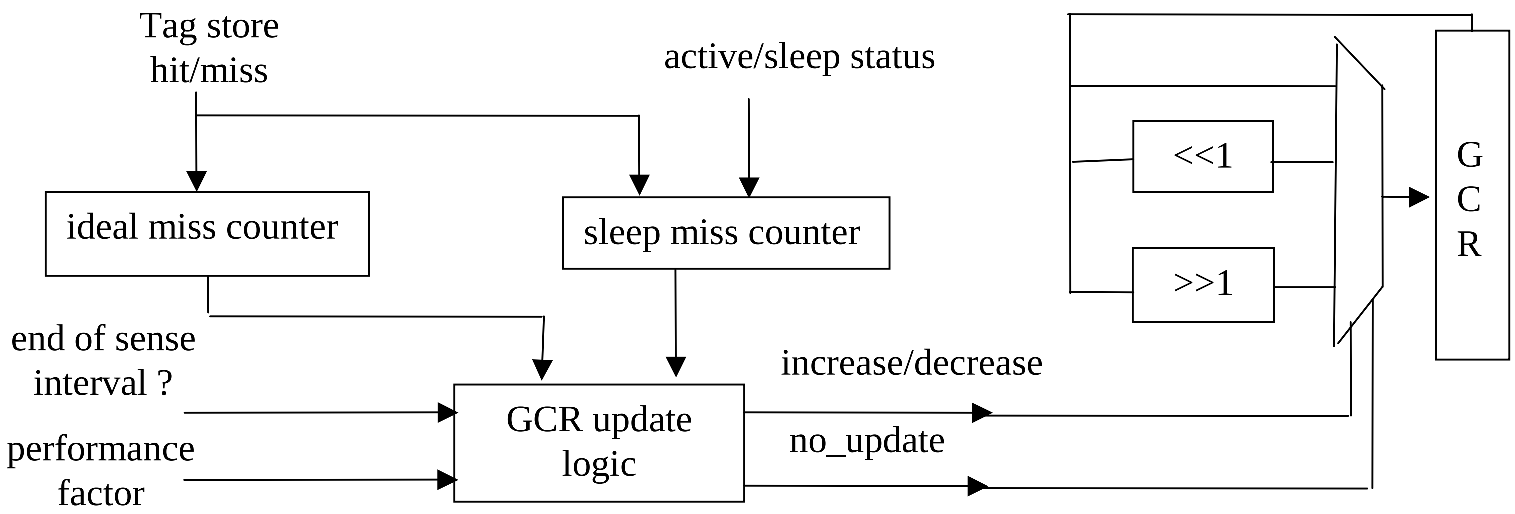
# AMC Cache Architecture



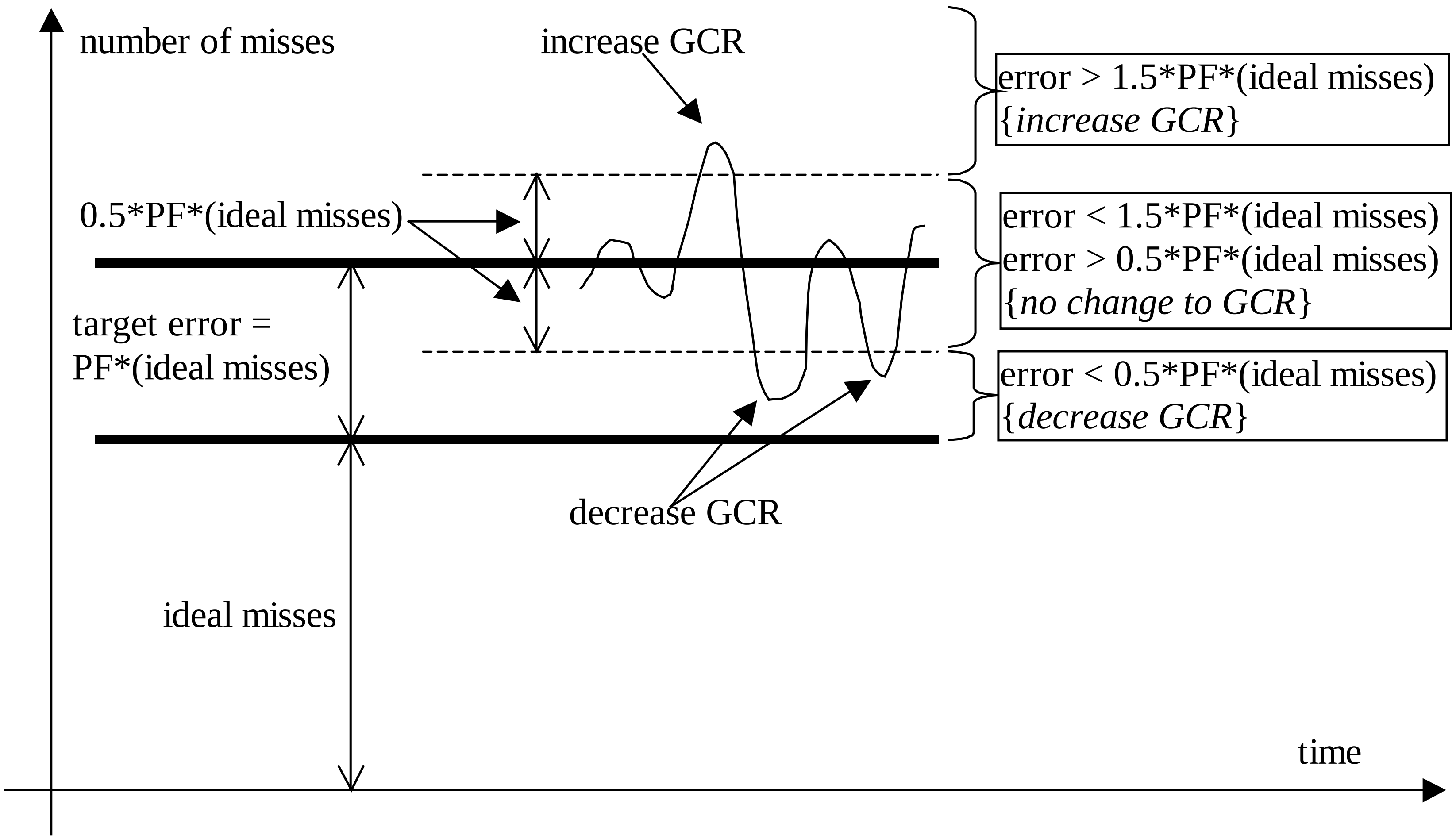
# Significance of Turn-off Interval (GCR)



# Adaptive Mechanism



# How the control system works



# GCR Update Algorithm

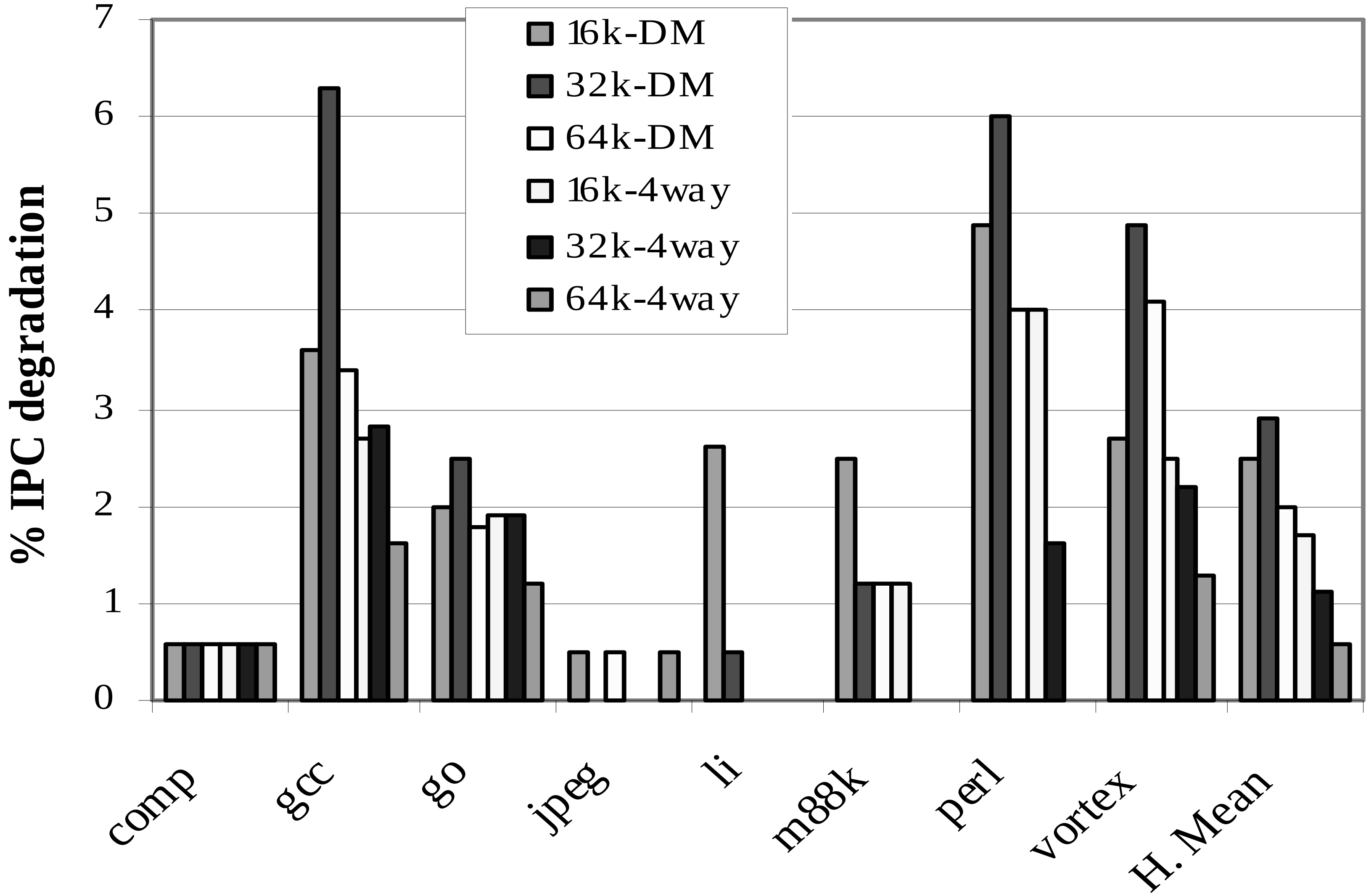
- Performance factor (PF): target ratio of sleep misses to ideal misses
  - Determines how many additional misses can be tolerated in exchange for static power savings
- Algorithm

```
if ((sleep misses) < ((ideal misses)*0.5*PF)) {
  decrease GCR: shift GCR right by one bit
}
else if ((sleep misses) > ((ideal misses)*1.5*PF)) {
  increase GCR: shift GCR left by one bit
}
else {
  do not change GCR
}
```

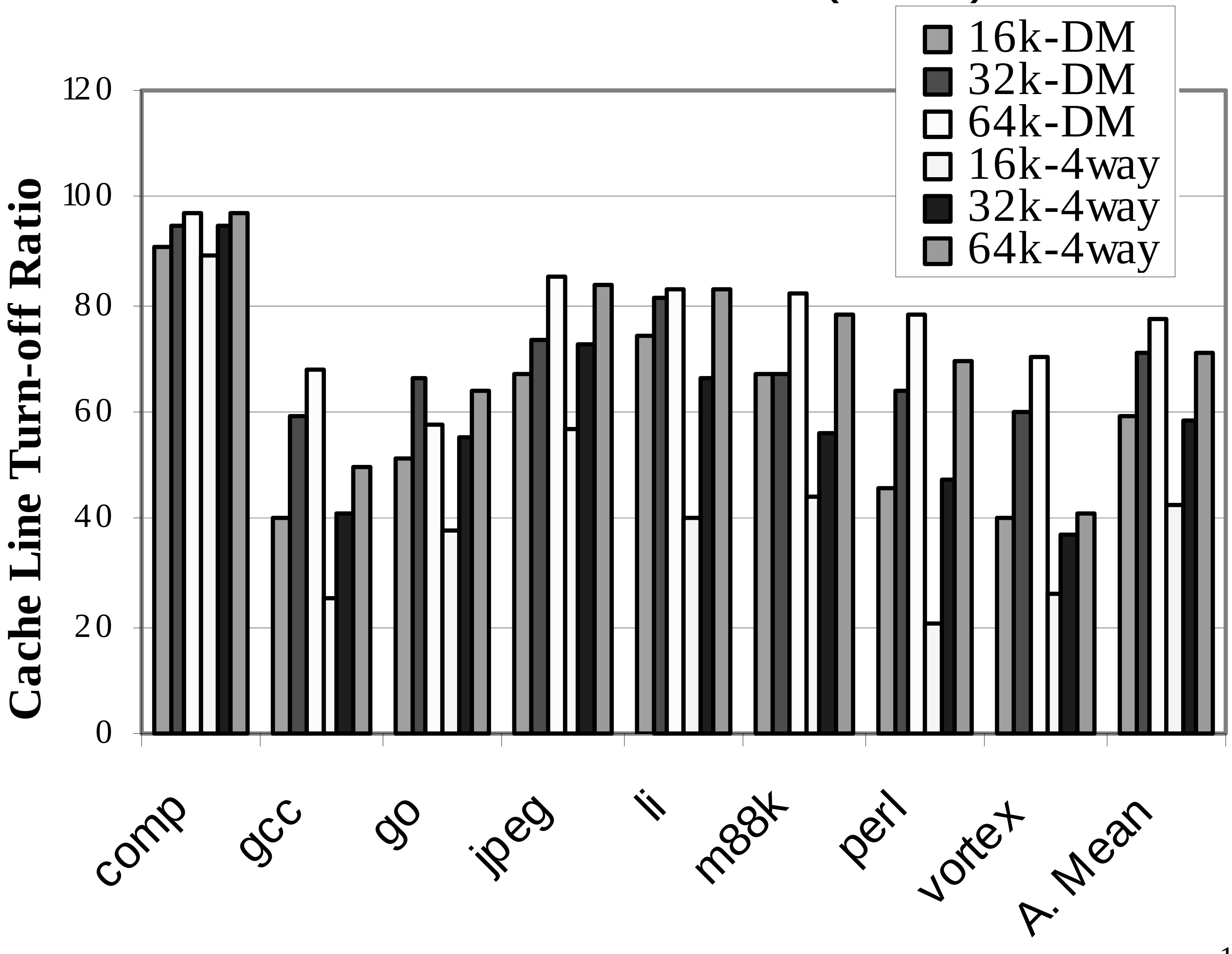
# Simulation Methodology

- A MIPS R10000-like, dynamically scheduled, 4-way issue superscalar processor
- Instruction and data caches
  - 16 KB / 32 KB / 64 KB
  - Direct-mapped and 4-way set-associative
  - 64-byte blocks
- I-cache hit time = 1 cycle; miss penalty 12 cycles
- D-cache hit time = 2 cycles; miss penalty 14 cycles
- AMC
  - PF = ½
  - Sense interval = 1 million cycles
  - LIC update interval = 2048 cycles

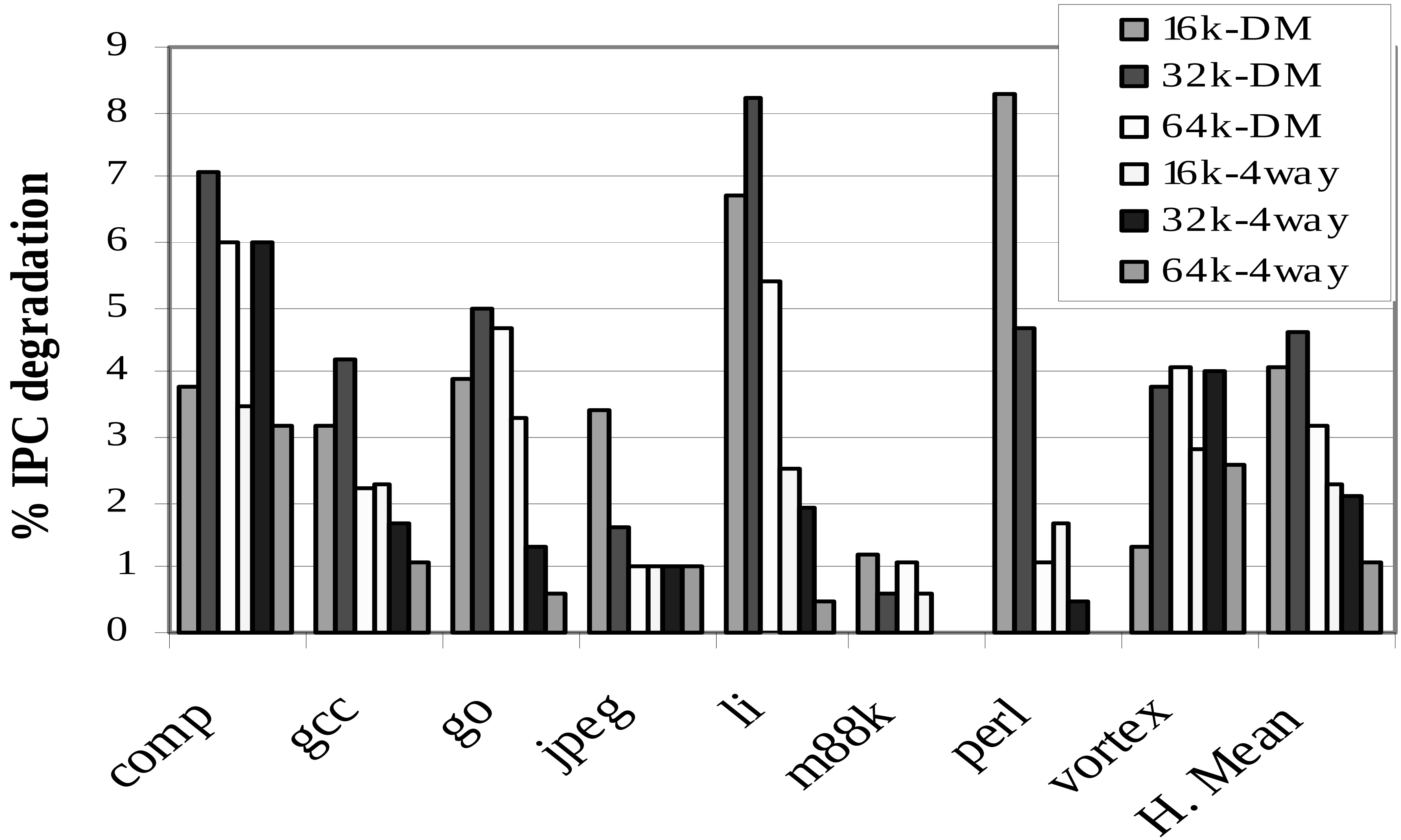
# AMC I-Cache Results



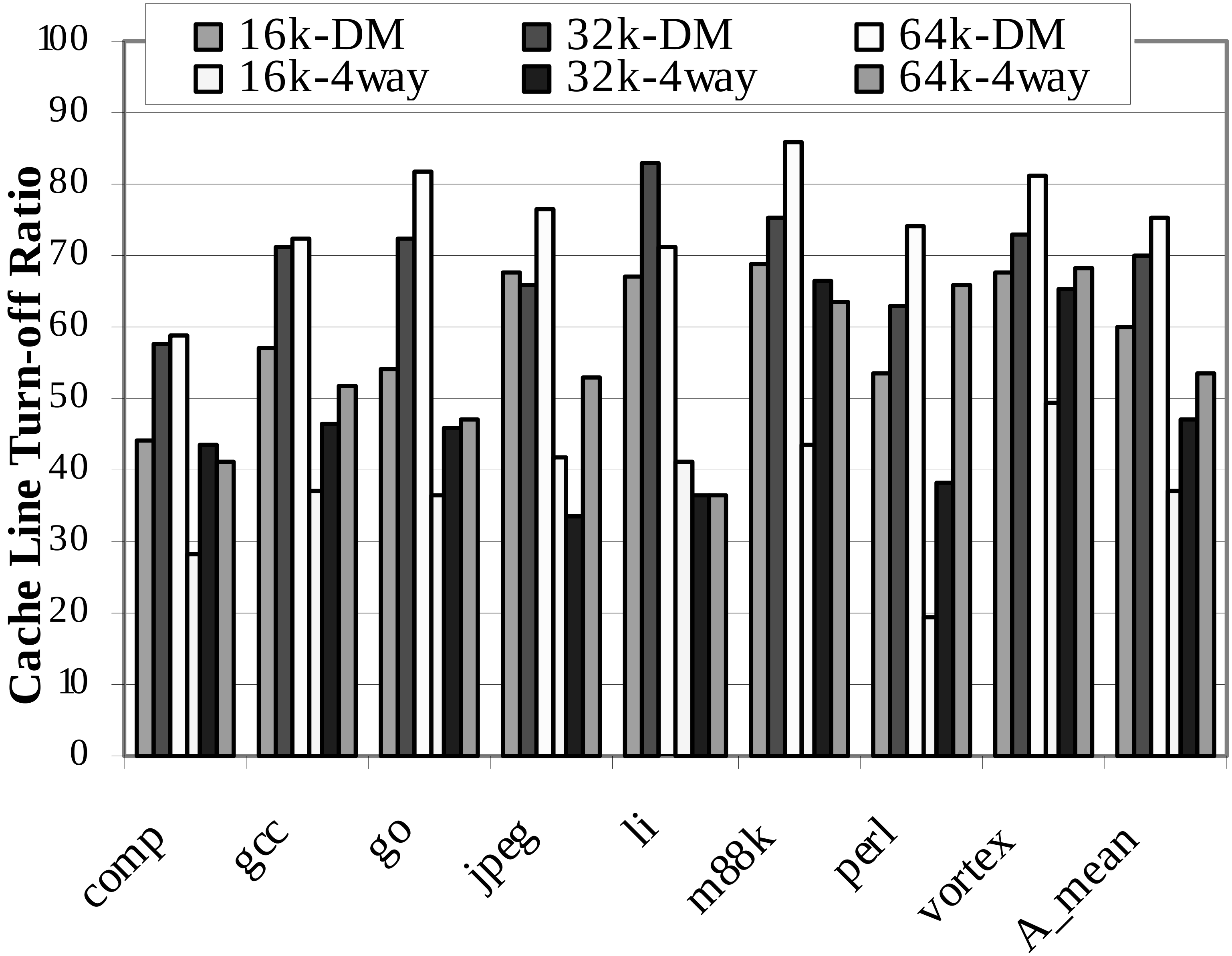
# AMC I-Cache Results (cont.)



# AMC D-Cache Results



# AMC D-Cache Results (cont.)



# AMC I-Cache and D-Cache Results

- AMC can be applied simultaneously to both the instruction cache and data cache
- 64kB 2-way I-cache and 64kB 4-way D-cache
  - Turn-off ratios of 73% and 56% for I-cache and D-cache, respectively
  - Performance degradation is only 1.8%

# Conclusion

- Key idea: The tag store is always kept active
  - Enables hypothetical performance without sleep mode to be determined and used to control real performance
  - Improvement over setting arbitrary and static performance targets
- Proposed a control system that keeps the number of sleep misses within a certain factor of ideal misses
- AMC is an effective means for improving static-power-efficiency in caches while maintaining good performance
- Uncovered interesting trends, e.g., higher associativity yields lower turn-off ratios

# Power Analysis

- See companion technical report for detailed power analysis
  - EDP and other metrics
  - Static and dynamic power analysis, including dynamic overhead for additional L2 requests and dynamic plus static overhead of LIC counters
  - We used Compaq 0.35µm 21264 I-cache technology
  - [H. Zhou, et. al., Technical Report, NCSU, Nov. 2000]

# Future Work

- Power analysis using projected technology data
- Compare with *generational cache line decay* approach
- Reducing power dissipation further
  - Keep only partial tags active
    - Increase static power savings
  - Exploit non-destructive sleep-mode circuit design
    - [K. Noii, et. al., ISPLED, 1998]
    - Eliminate dynamic power increase due to refreshing sleep-mode data from L2 cache
    - Eliminate dynamic power increase due to writing dirty data to L2 before deactivating cache line

# Contact Information


**Huiyang Zhou** hzhou@eos.ncsu.edu

**Mark Toburen** mctobure@eos.ncsu.edu

**Eric Rotenberg** ericro@eos.ncsu.edu

**Tom Conte** conte@eos.ncsu.edu

North Carolina State University

www.tinker.ncsu.edu