Joint Architecture and Circuit Techniques to Address Process and Voltage Variability

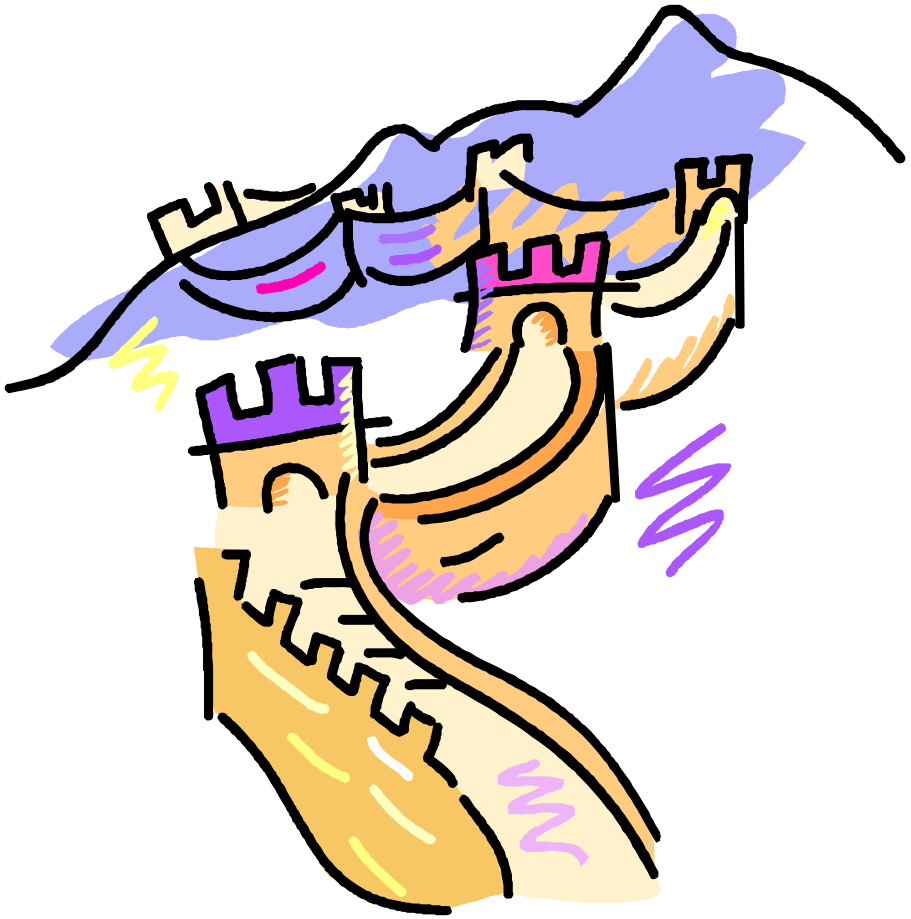
Gu-Yeon Wei & David Brooks

School of Engineering and Applied Sciences

Harvard University

The Great Wall of Collaboration

Architect



Circuit Designer

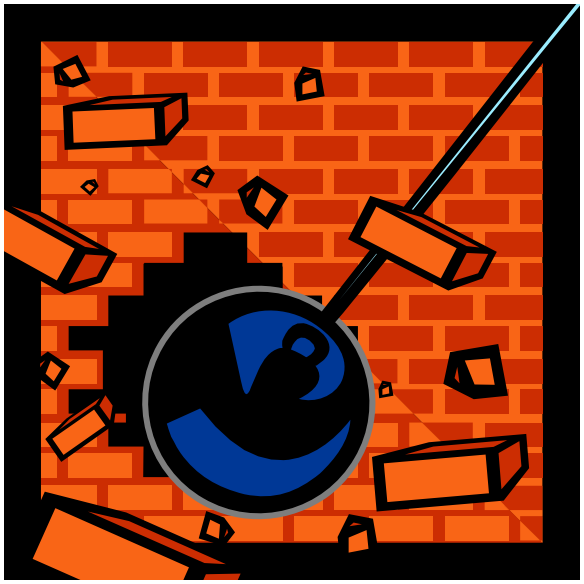
The Great Wall of Collaboration

Architect

Circuit Designer

The Great Wall of Collaboration

Architect



Circuit Designer

Architecture & Circuits Groups

Not shown:
- **Andrew**
- **Krishna**
- **Ruwan**
- **Ben**

Collaborative Projects

- SW+Arch+HW for efficient power delivery
  - Understanding Voltage Variations in CMPs Using a Distributed Power Delivery Network (DATE ’07)
  - Toward a SW Approach to Mitigate Voltage Emergencies (ISLPED’07)
  - DeCoR: A Delayed-Commit and Rollback Mechanism for Handling Inductive Noise in Processors (HPCA’08)
  - System-Level Analysis of Fast, Per-Core DVFS using On-Chip Switching Regulators (ASGI’07, HPCA’08)

- SW+Arch+HW to combat process variations
  - Mitigating the Impact of Process Variation on CPU RF and Execution Units (MICRO’06)
  - Process Variation Tolerant 3T1D-Based Cache Architectures (ASGI’07, MICRO’07)
  - A Process Variation Tolerant FPU with Voltage Interpolation and Variable Latency (ISSCC’08)

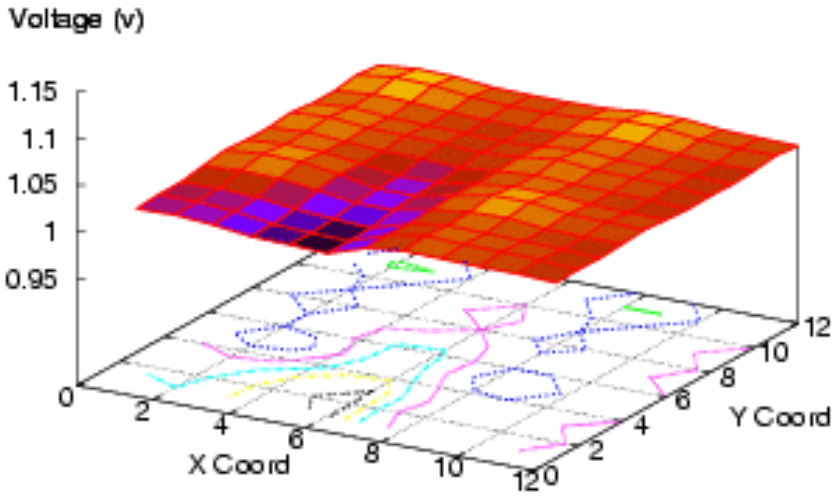
Today's Topics

- System-Level Analysis of Fast, Per-Core DVFS using On-Chip Switching Regulators
  - Wonyoung Kim, Meeta Gupta, Wei and Brooks
  - To be presented at HPCA in Feb. 2008
- Process Variation Tolerant 3T1D-Based Cache Architectures
  - Xiaoyao (Alex) Liang, Ramon Canal (UPC Barcelona), Gu-Yeon Wei and David Brooks
  - To be presented at MICRO in Dec. 2007

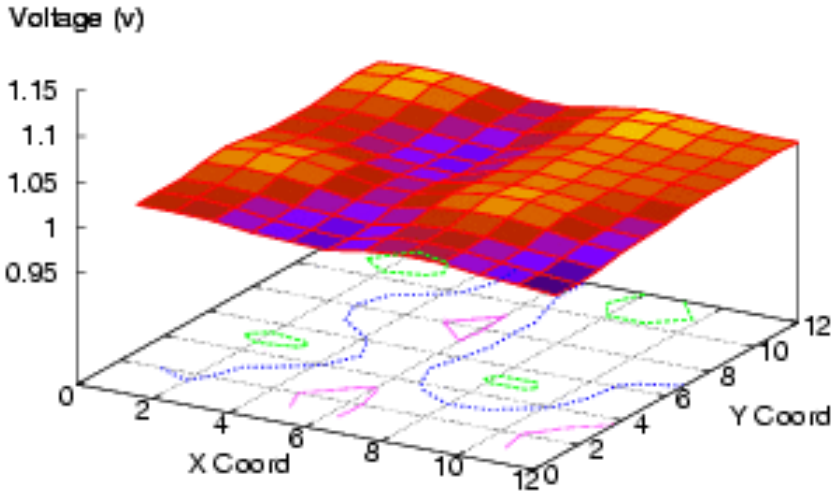
Seminar Part 1

SYSTEM-LEVEL ANALYSIS OF FAST, PER-CORE DVFS USING ON-CHIP SWITCHING REGULATORS

Voltage Variability Movie

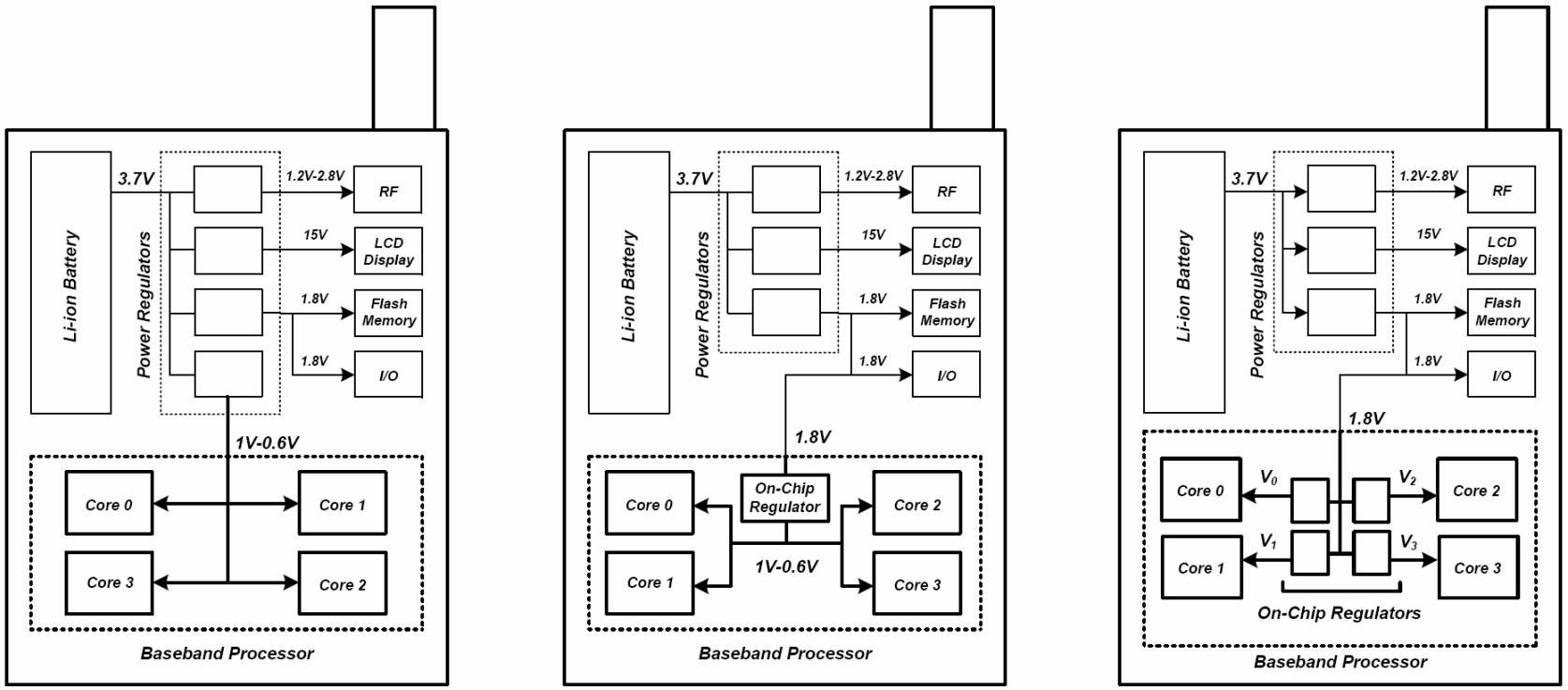


3 cores running *bzip*, 1 core idle



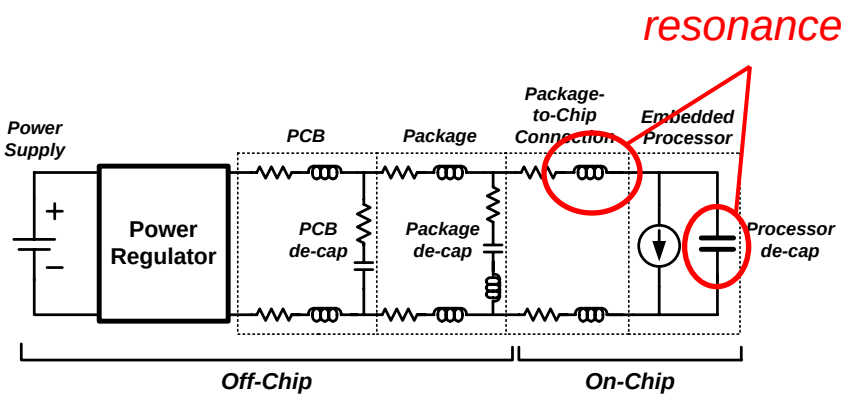
1 core running *bzip*, 3 cores idle

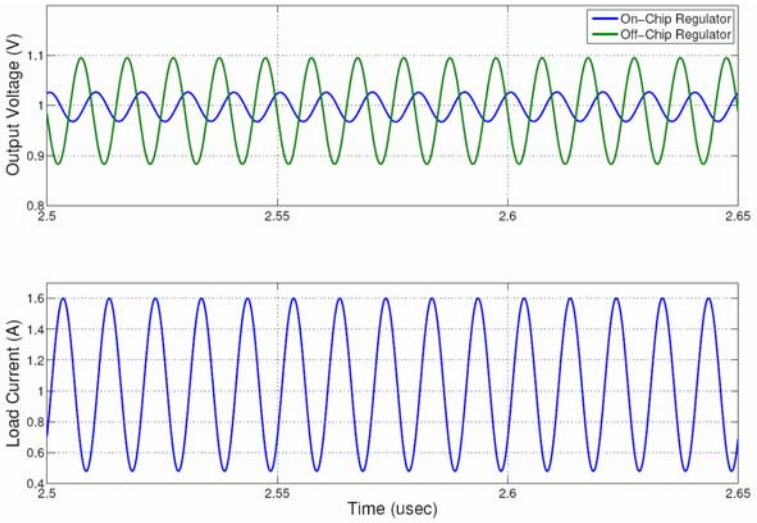
Motivating Example

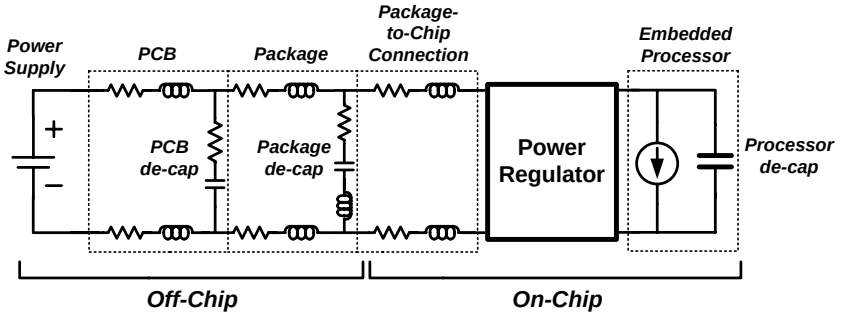


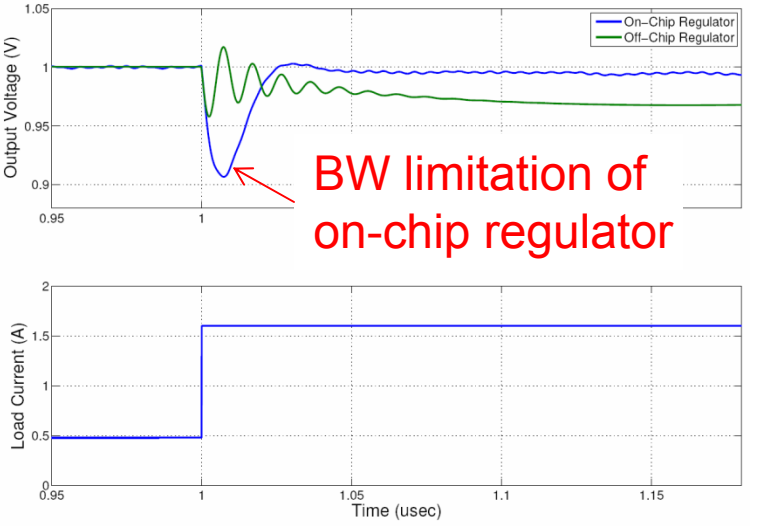
- Can we move the off-chip regulator onto the processors?
- If yes, WHY?

Supply Noise Comparison

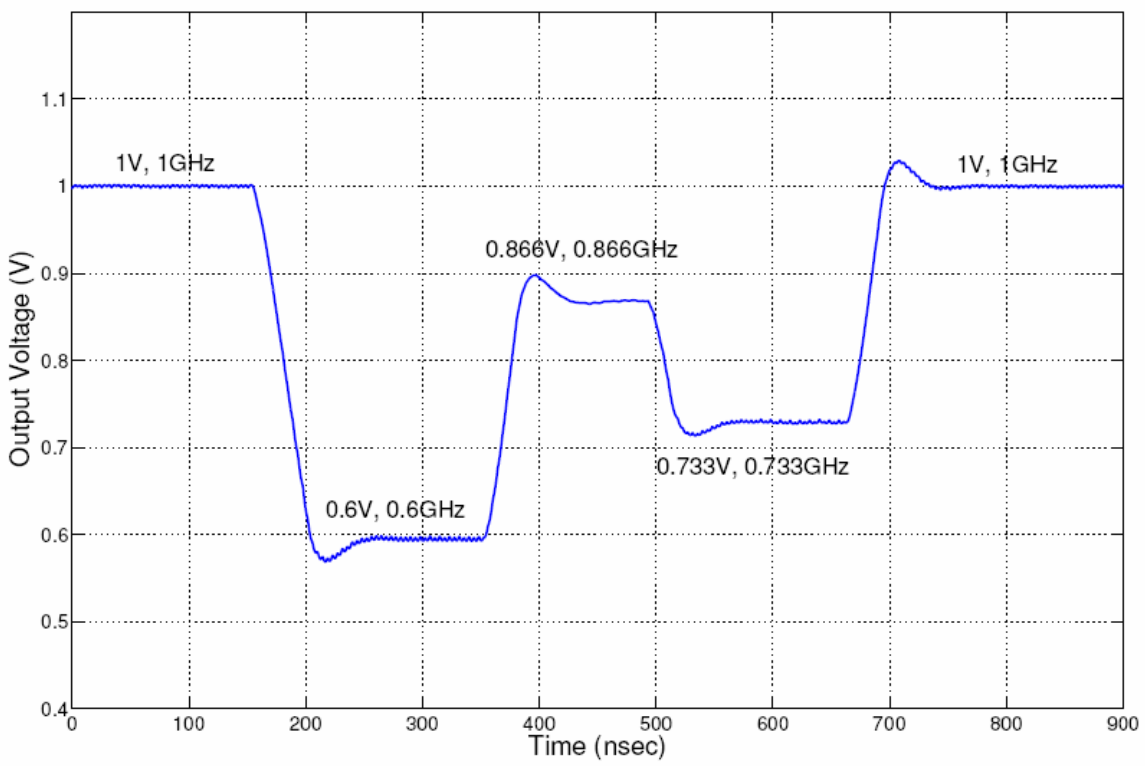








Fast DVFS



- Off-chip regulators limited to microsecond-scale transitions
- On-chip regulators enable nanosecond-scale voltage transitions
  - Can we leverage this fast switching?

Outline

- Motivation
- Offline DVFS
- On-chip regulator design
- Simulation analysis
- Summary & future work

Fast DVFS w/ On-Chip Regulators

Questions to answer:

1. Does fast DVFS offer power savings?
2. For CMPs, do we want one global supply or per-core voltage control?
3. What does an on-chip regulator cost us?
4. How can architecture help regulator design?
5. How does this all add up?

DVFS Overview

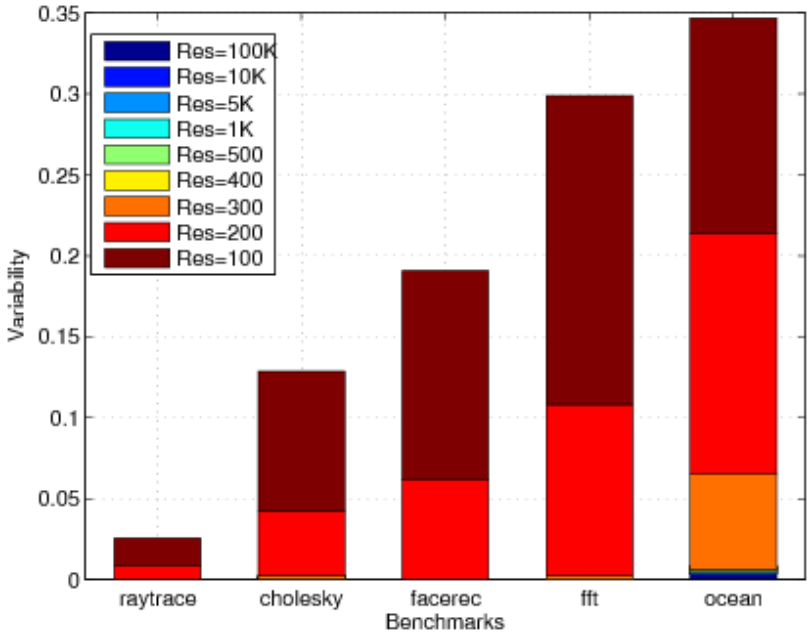
- Minimize energy consumption w/ bounded performance loss
  - Exploit CPU slack from asynchronous memory events (i.e., L2 miss) to reduce frequency (F) and voltage (V)
- Offline DVFS control
  - Formulate as integer linear programming (ILP) optimization problem
  - Oracle uses memory vs. CPU boundedness to set V/F across different windowed intervals
    - 4 V/F settings assumed
  - Compare different intervals (100ns to 100μs)

DVFS Architecture Study

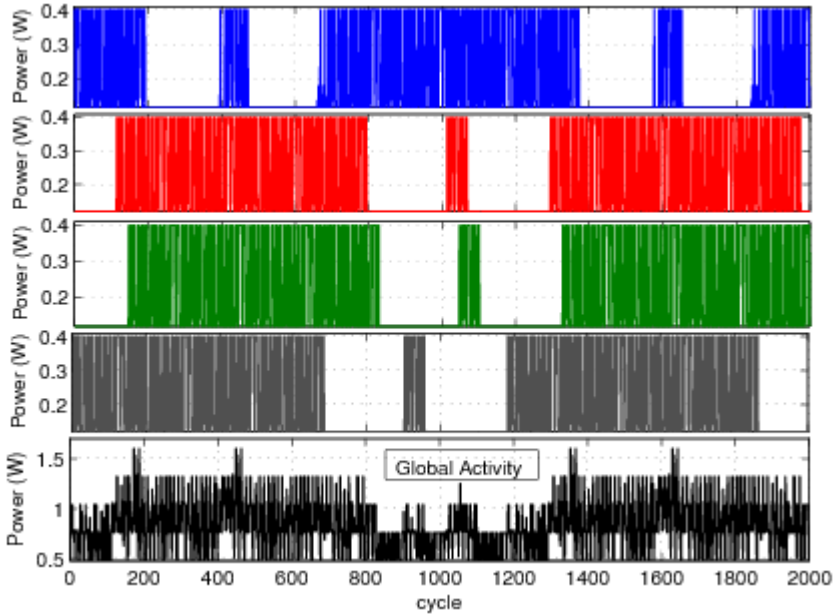
Frequency	1GHz @ 65nm	Vdd	1 V
Core area	$16mm^2$	Fetch/Issue/Retire	2/2/2
Branch Penalty	7 cycles	Branch Predictor	BTB (1K entries)
	Hybrid Branch Predictor		RAS (32 entries)
Int registers	32	FP registers	32
IL1	32KB, 32-way, 32B block Hit/Miss latency 2/1 cycles	DL1	32KB, 32-way, 32B block Hit/miss latency: 2/1 cycles MESI-protocol
ITLB entries	64	DTLB entries	128
MSHR size	8	Write Buffer size	16
L2 size	512 KB	L2 miss penalty	200 cycles

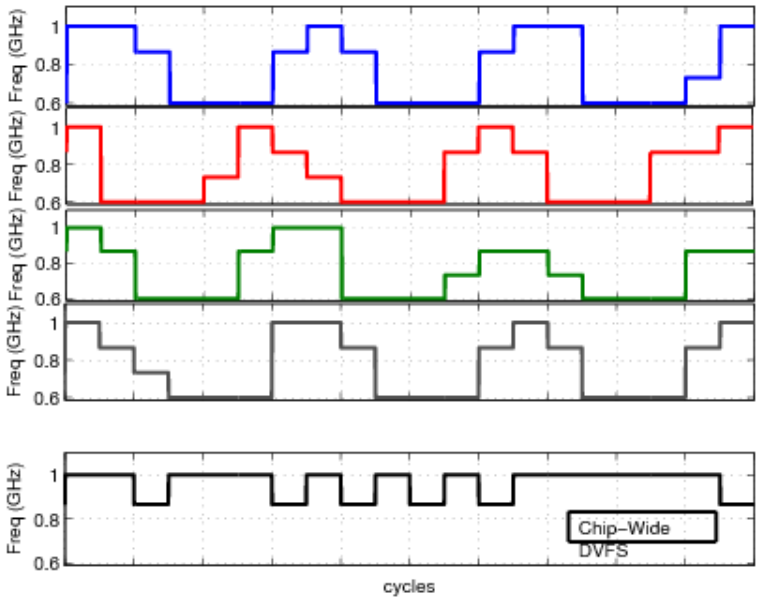
Benchmark	Description	(Memory Cycles/Total Runtime)
Ocean-con	Large Scale ocean simulation	0.47
fft	Fast Fourier Transform	0.40
facerec	CSU Face Recognizer	0.22
cholesky	Cholesky factorization	0.197
raytrace	Tachyon Ray Tracer	0.058
mcf-mcf-mcf-mcf	4-high memory-bound	0.697
mcf-mcf-mcf-applu	3-high memory-bound(*mcf*) and 1-high cpu-bound (*applu*)	0.697(mcf) and 0.051(applu)
mcf-mcf-applu-applu	2-high memory-bound(*mcf*) and 2-high cpu-bound (*applu*)	0.697(mcf), 0.051(applu)
mcf-applu-applu-applu	1-high memory-bound(*mcf*) and 3-high cpu-bound (*applu*)	0.697(mcf), 0.051(applu)
applu-applu-applu-applu	4-high cpu-bound(*applu*)	0.051

- Processor model
  - 4 simple Xscale-like in-order cores
  - Private L1, shared L2
- Simulation framework
  - SESC multi-core simulator
  - Wattch power modeling
  - Cacti cache simulator
  - Orion
  - MESI-based cache coherence
  - Multithreaded and multi-programming benchmarks



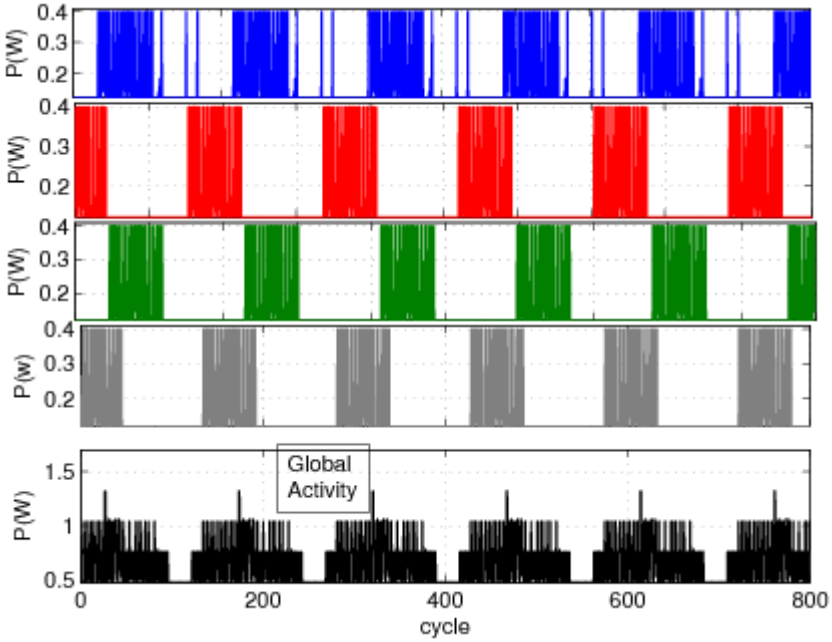
Ocean's DVFS Opportunities

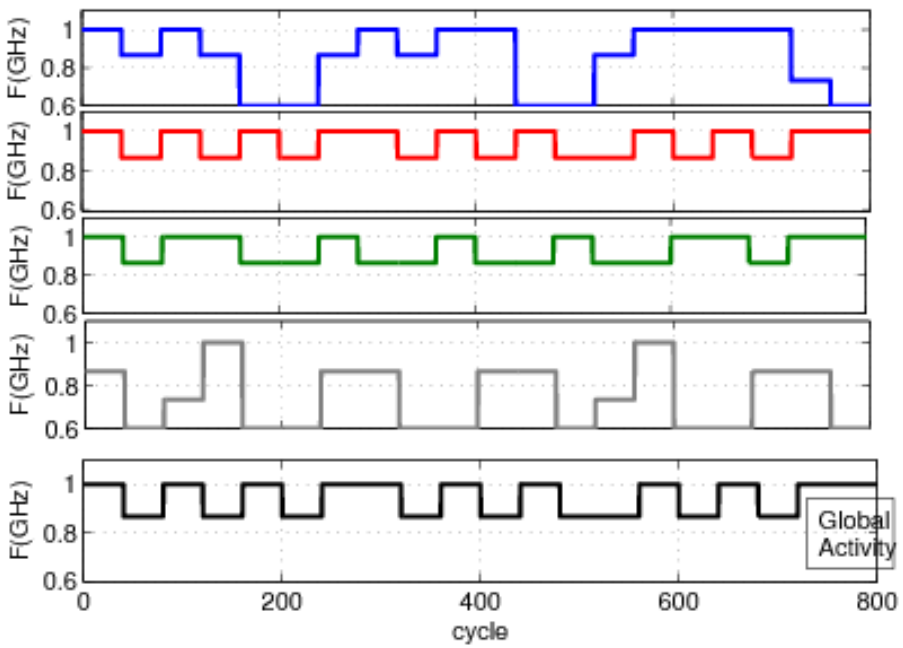




- Multithreaded *ocean* running on all 4 cores exhibits variable activity between cores
- Per-core voltage again offers more DVFS opportunities

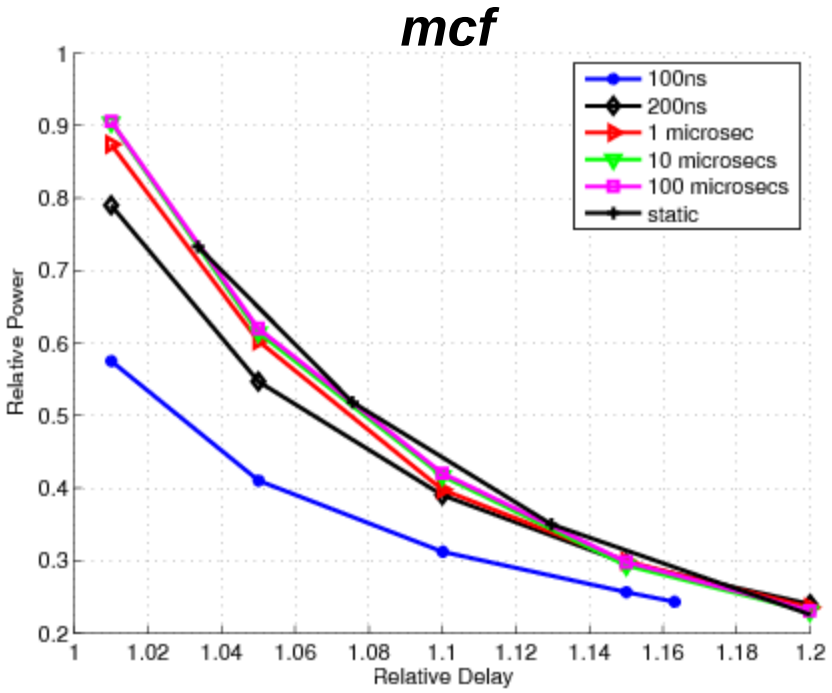
fft's DVFS Opportunities

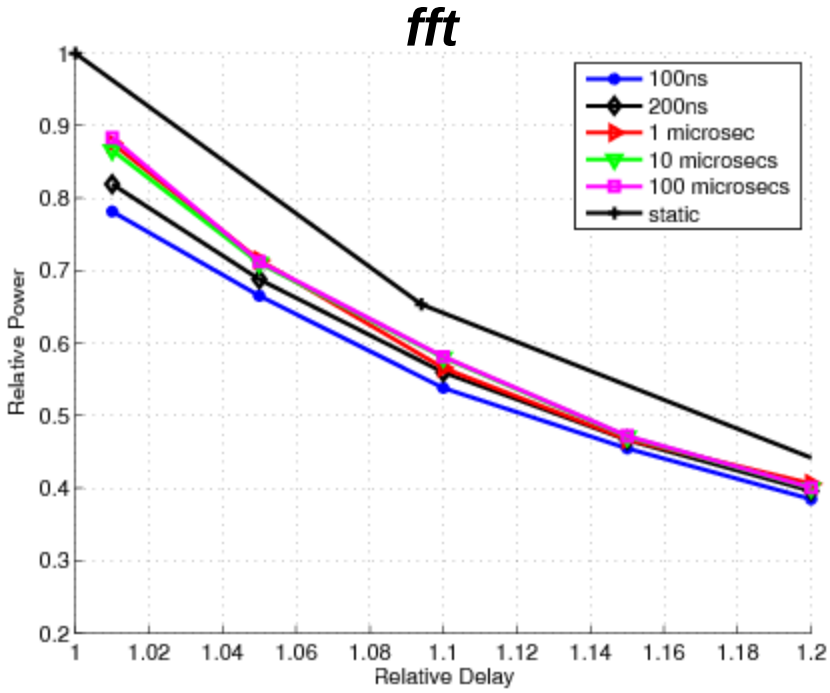




- Multithreaded *fft* running on all 4 cores exhibits variable activity between cores
- Per-core voltage offers more DVFS opportunities

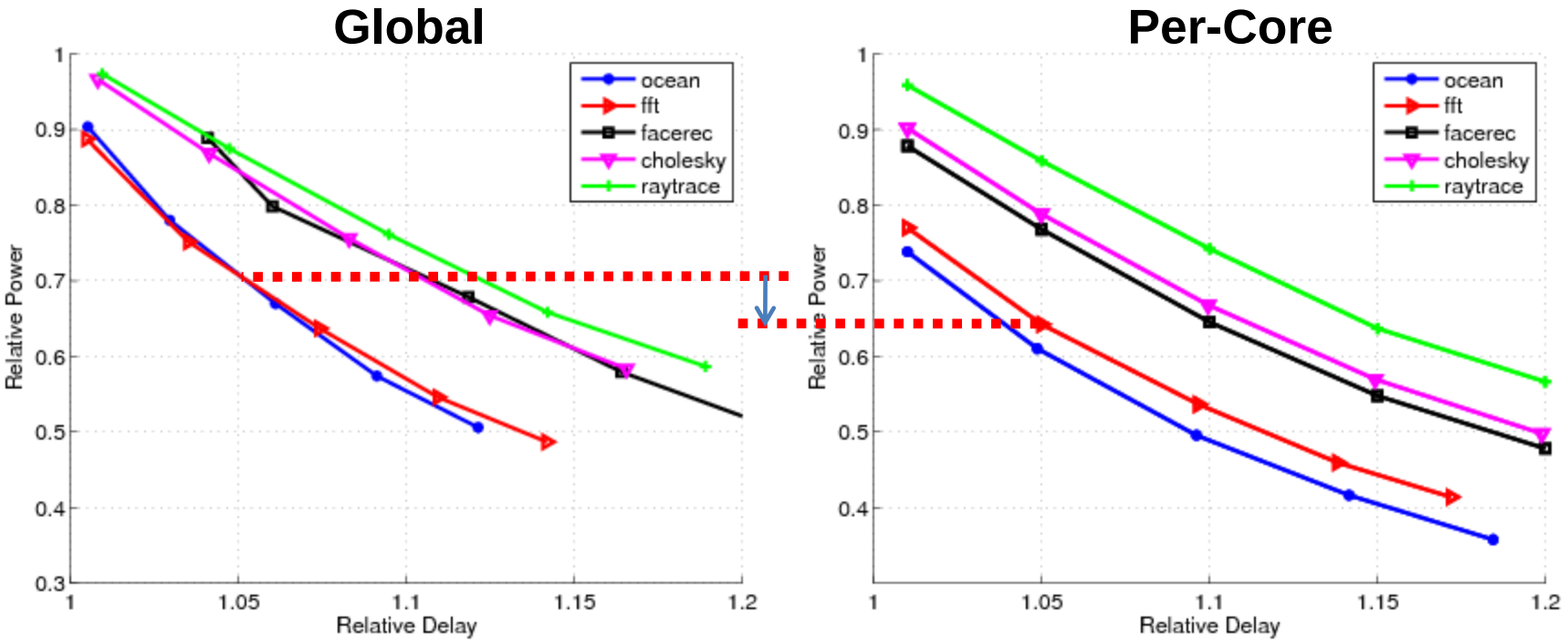
Benefits of Fine-Grained DVFS





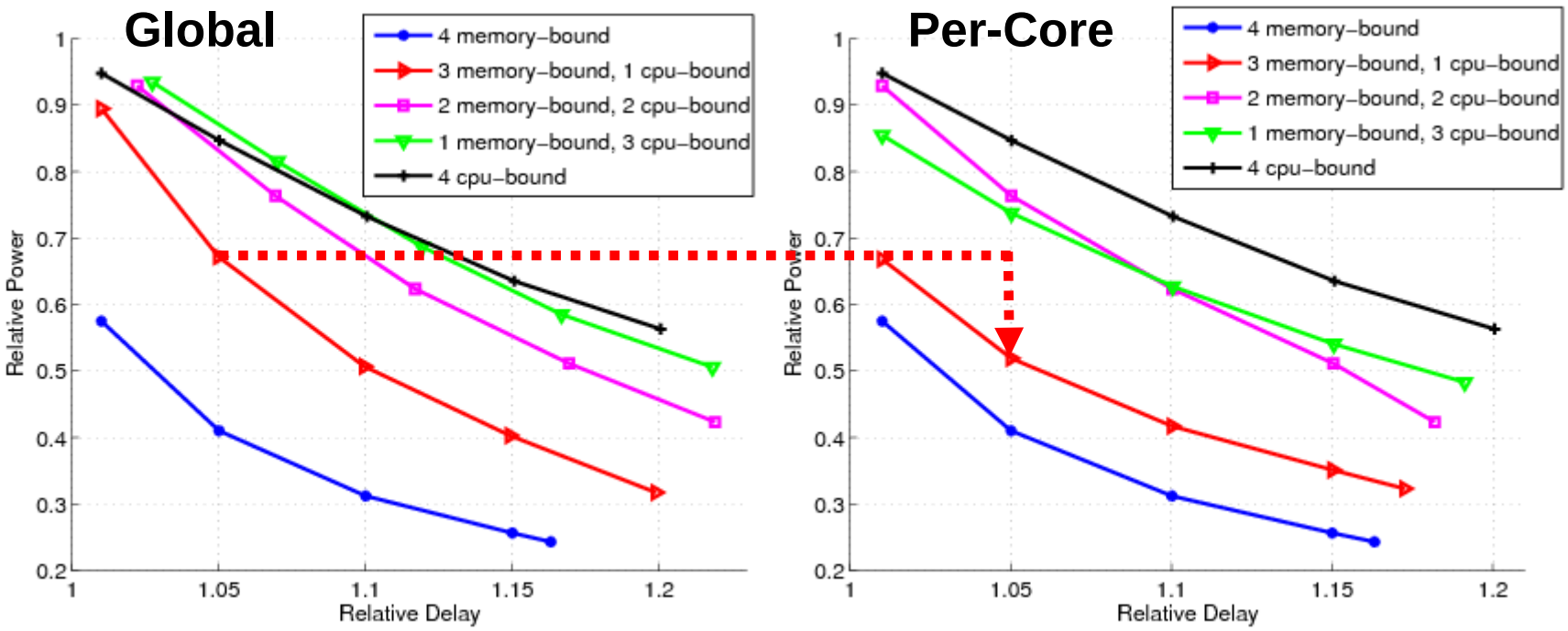
- Off-chip regulator → 100μs – static (app-level) intervals
  - OS-level DVFS control
- On-chip regulator → 100ns – 1μs intervals
  - Needs online DVFS control

Global vs. Per-Core DVFS
(multithreaded applications)



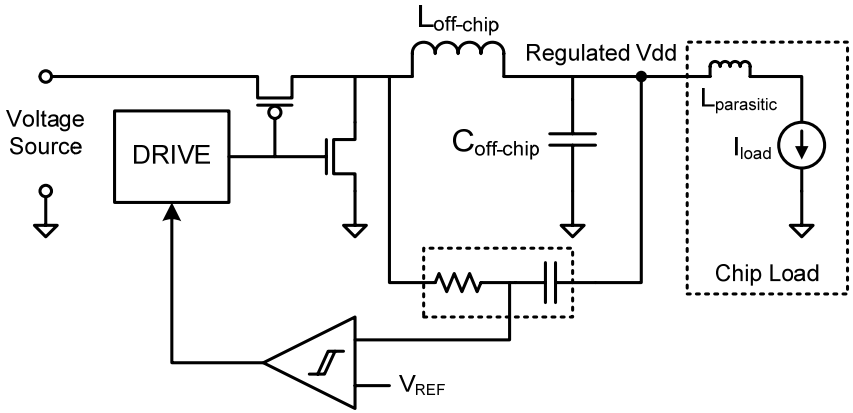
- DVFS interval = 100ns
- Per-core DVFS offers more savings
- Savings vs. benchmark trend tracks “variability”

Global vs. Per-Core DVFS
(multi-programming applications)



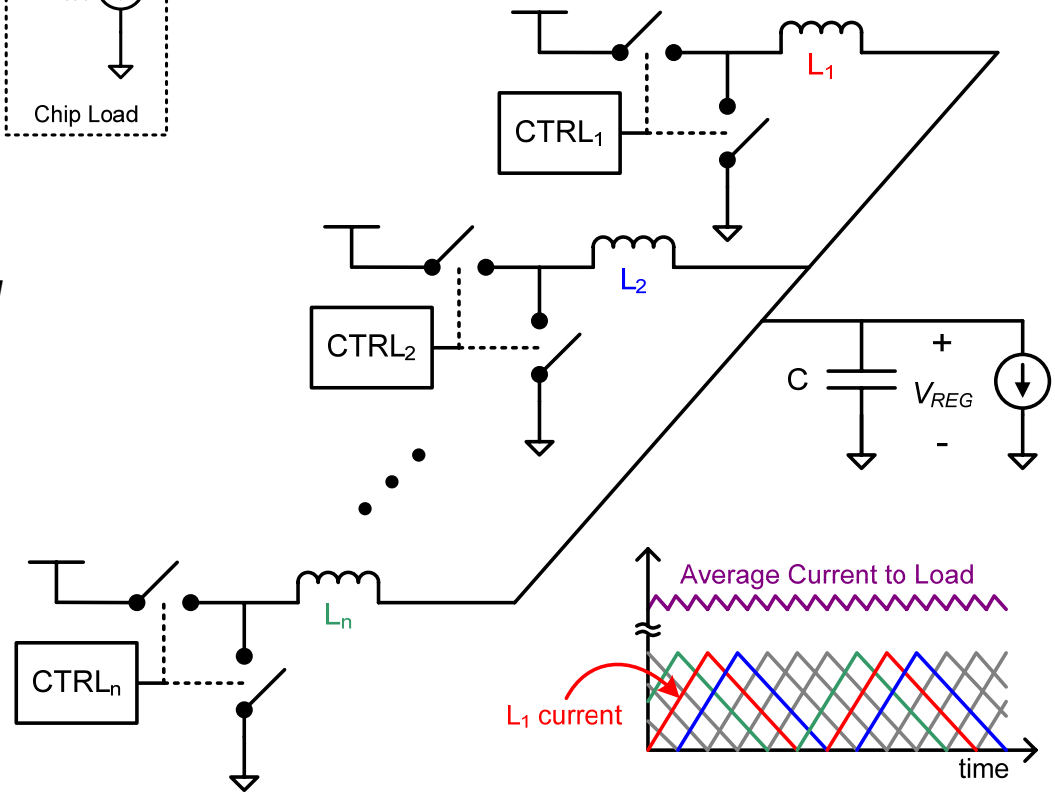
- DVFS interval = 100ns
- *mcf* = memory-bound app; *applu* = CPU-bound app
- Power savings for mix of memory- and CPU-bound apps

Regulator Design



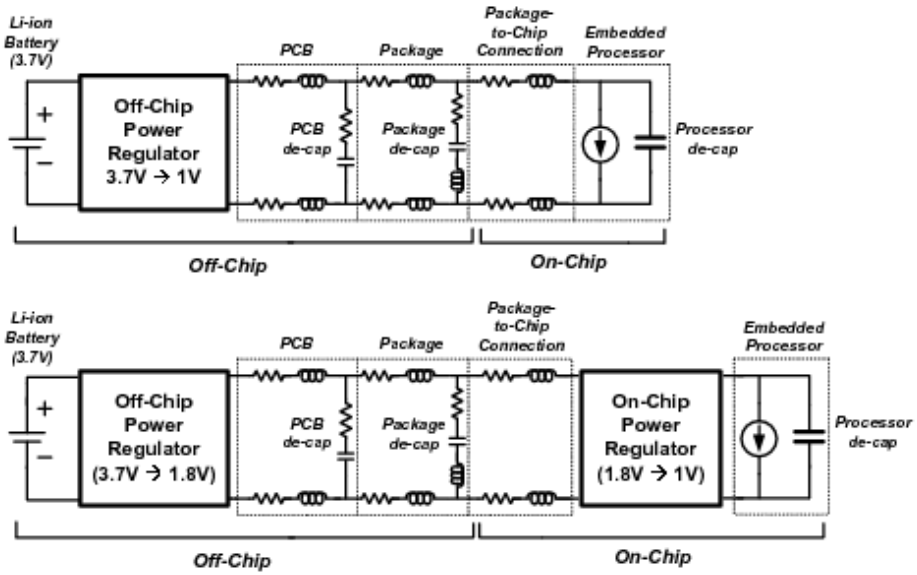
Conventional buck converter w/ hysteretic control

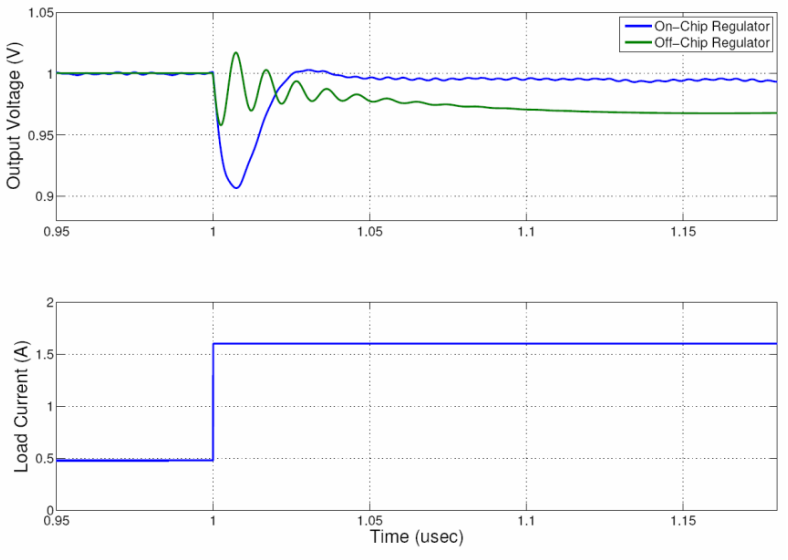
- $P_{delivered} = ½ LI^2F_{switching}$
- On-chip multiphase buck converter
  - Higher $F_{switching}$
  - Smaller L & C
  - Lower V_{ripple} and/or smaller filter C



Multi-phase buck converter

Power Delivery Options

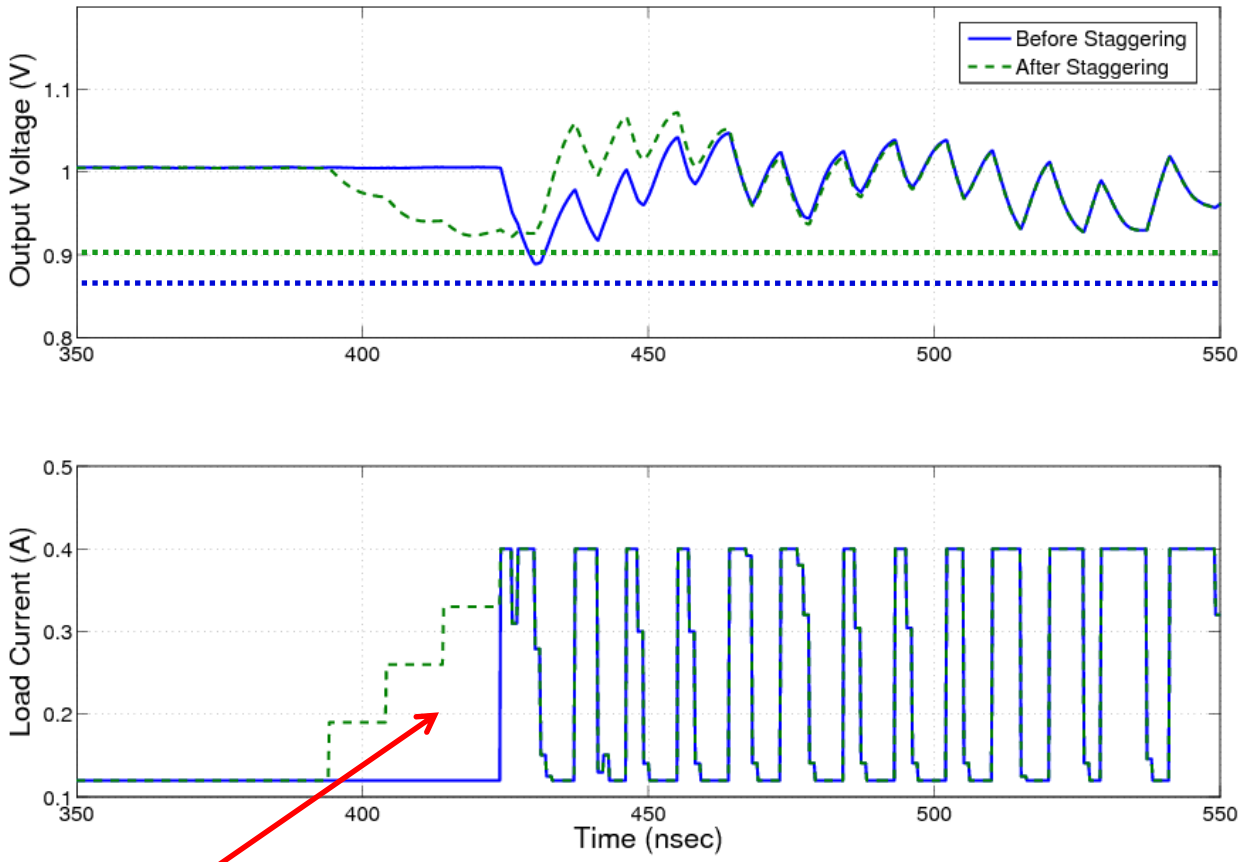




(x4 for per-core DVFS)

- Can we leverage architecture to reduce the droop?

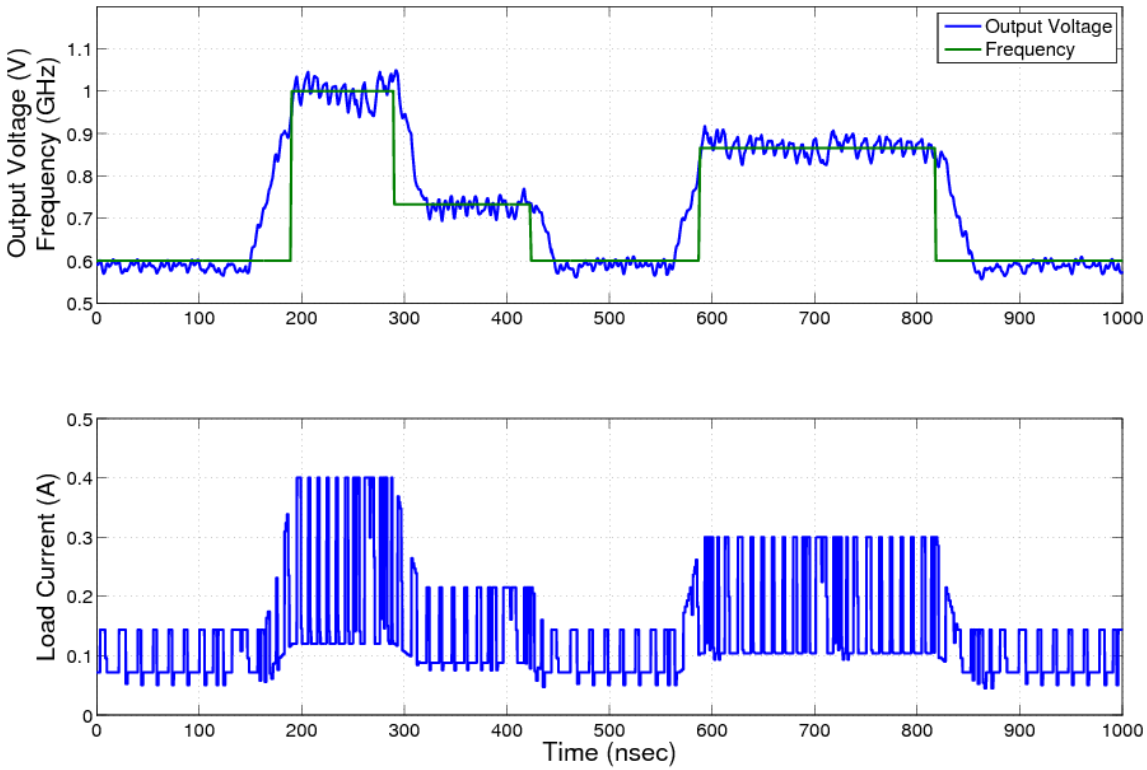
Current Staggering



Voltage margins

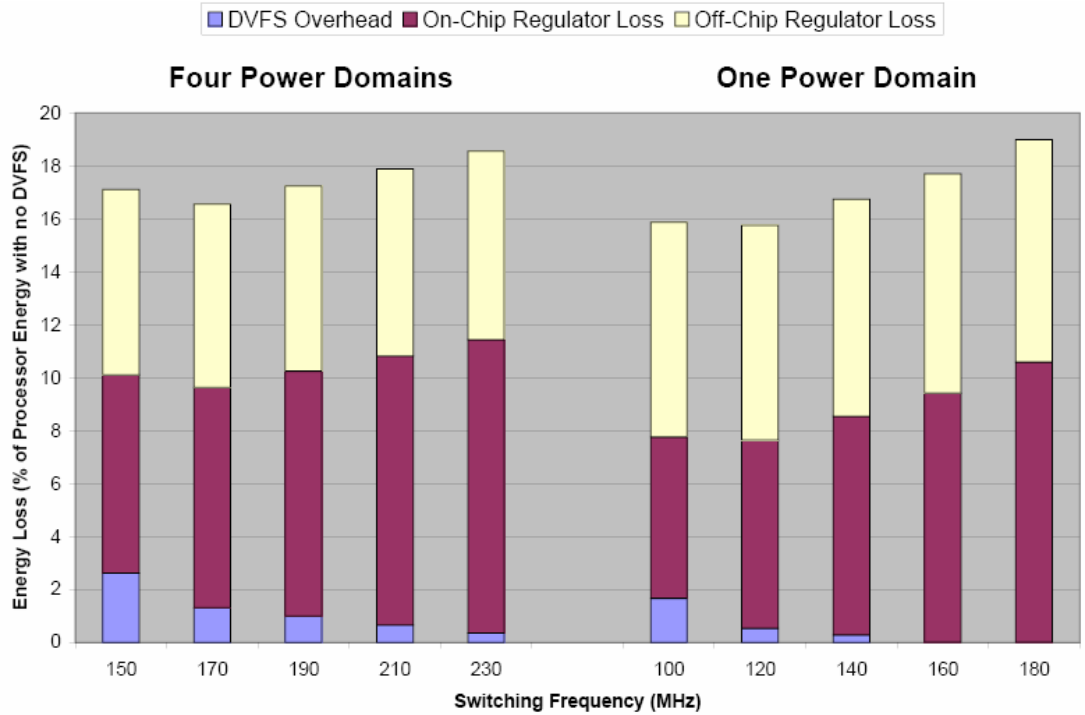
- Burn power to reduce voltage droop

Voltage Transition Overhead



- Scale up voltage before increasing frequency
- Drop frequency before decreasing voltage
- Power overhead = area between curves

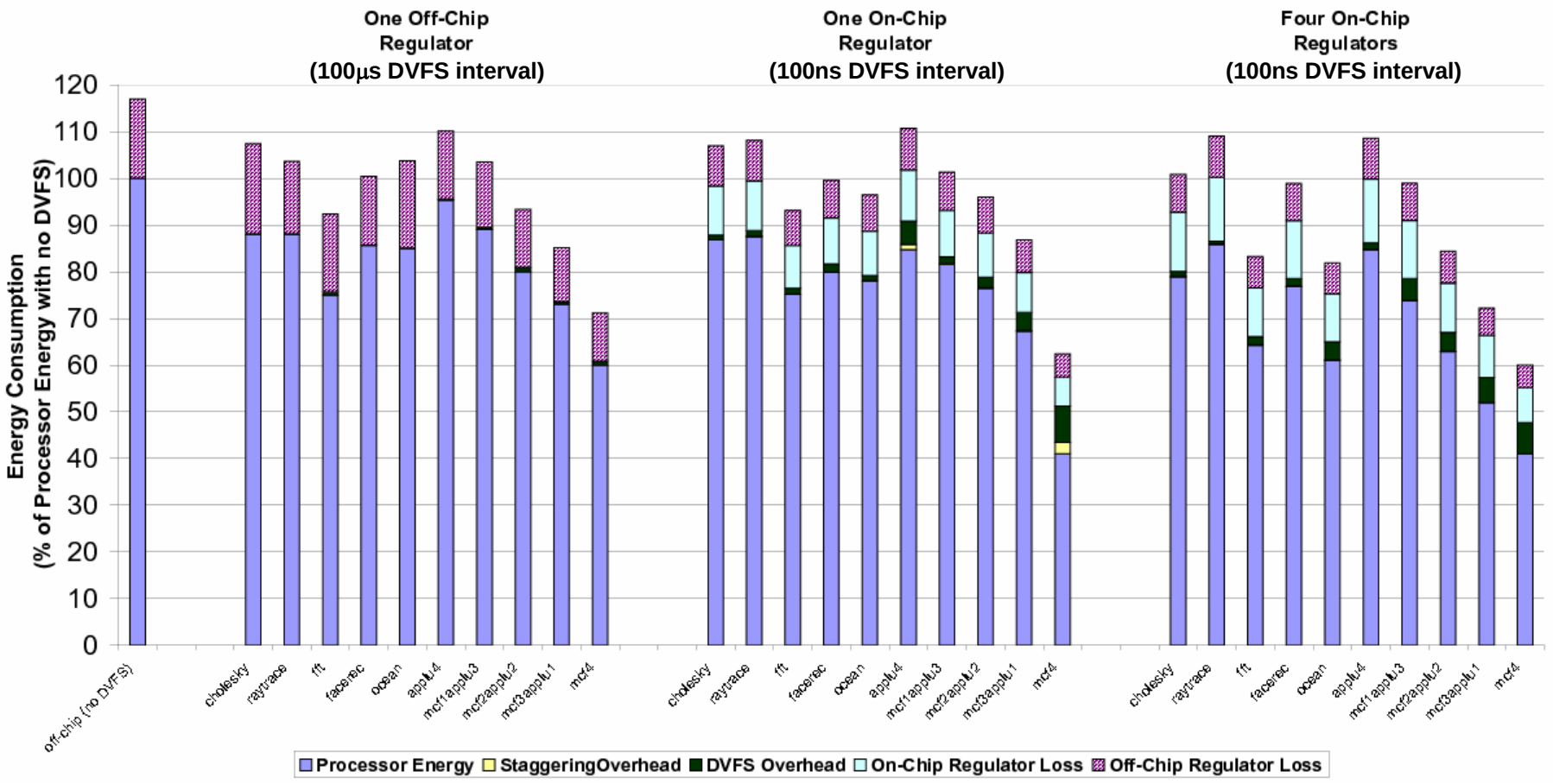
Regulator Specifications



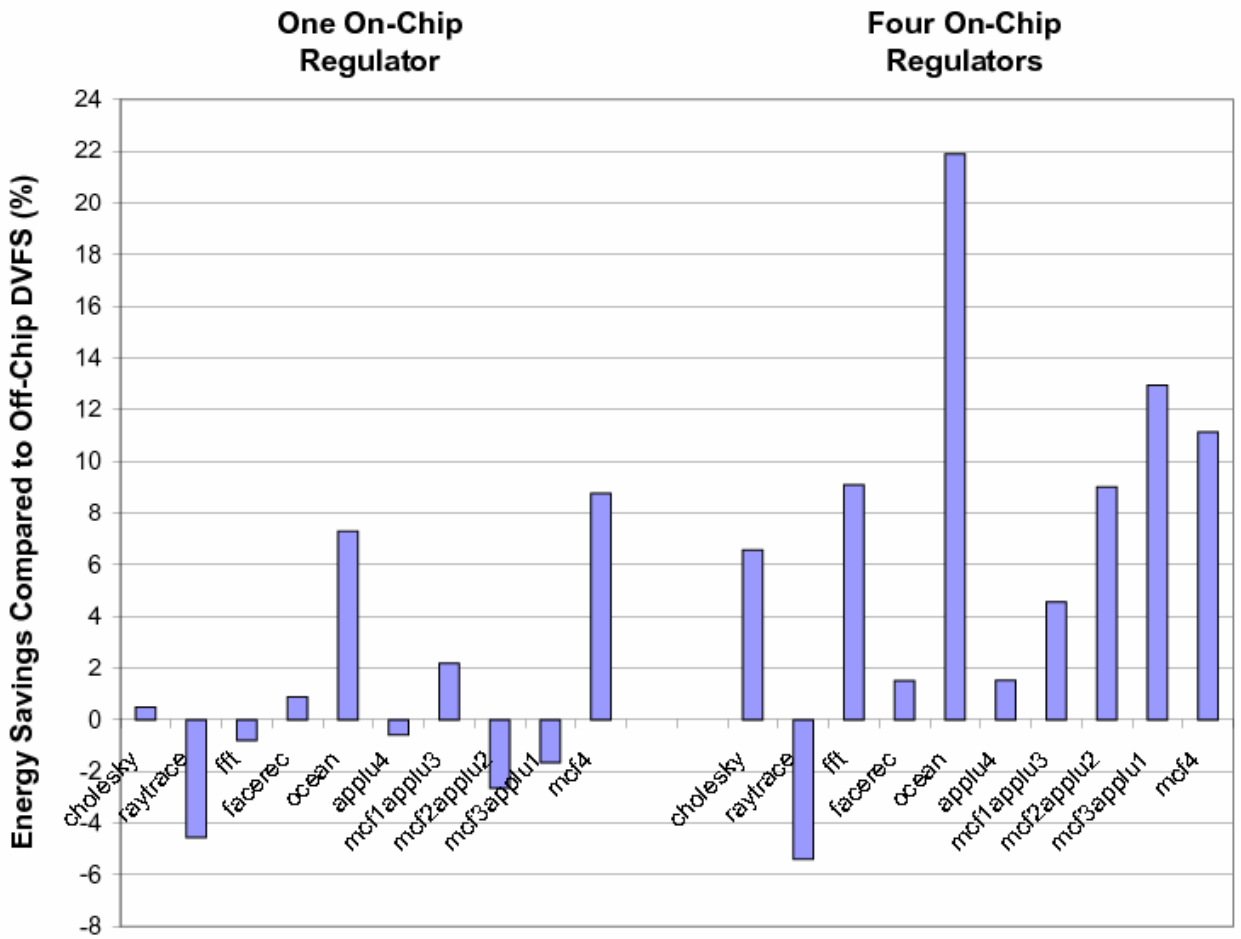
- Optimized $F_{switching}$ with respect to losses
  - Balance DVFS overhead with regulator loss

	off chip	off-chip & on-chip	off-chip & on-chip
	One Power Domain		Four Power Domains
# of phases for on-chip regulator		8	2
On-chip regulator switching frequency (MHz)		170	120
Decoupling capacitance (nF)	40	40	10 per core(total:40)
Voltage margin (%)	±10		

Energy Breakdown Comparison



Relative Energy Savings



Putting It All Together

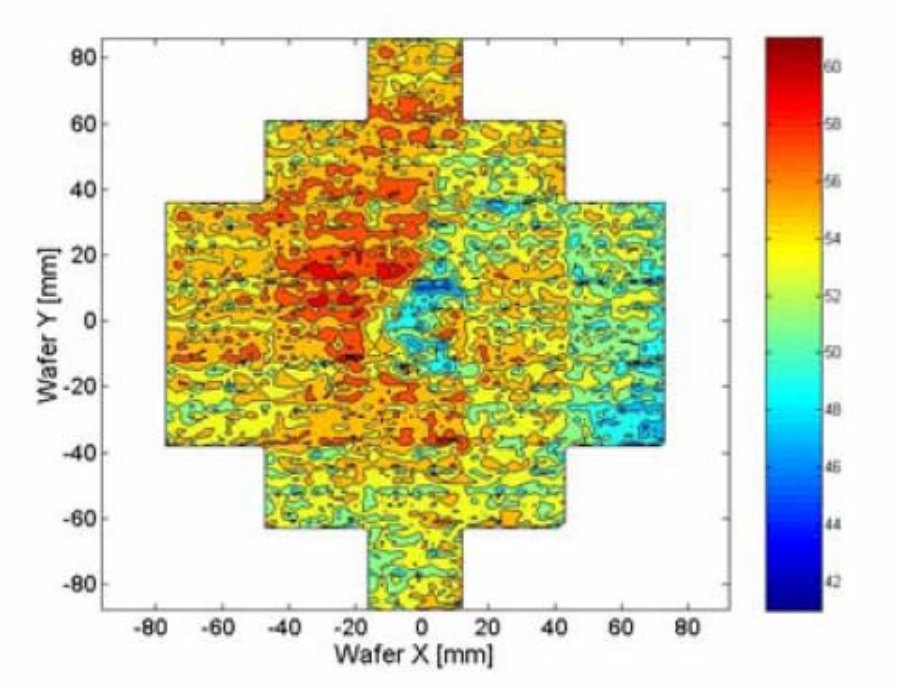
- Energy savings with fast DVFS offset by
  - On-chip regulator loss
  - Voltage transition power overhead
  - Current staggering overhead
- Per-core DVFS attractive for CMP systems
  - Must consider scalability of on-chip regulators

- Next steps:
  - Meeta is investigating fast DVFS scaling algorithms to leverage fast, fine-grained voltage switching
  - Wonyoung is designing the regulator

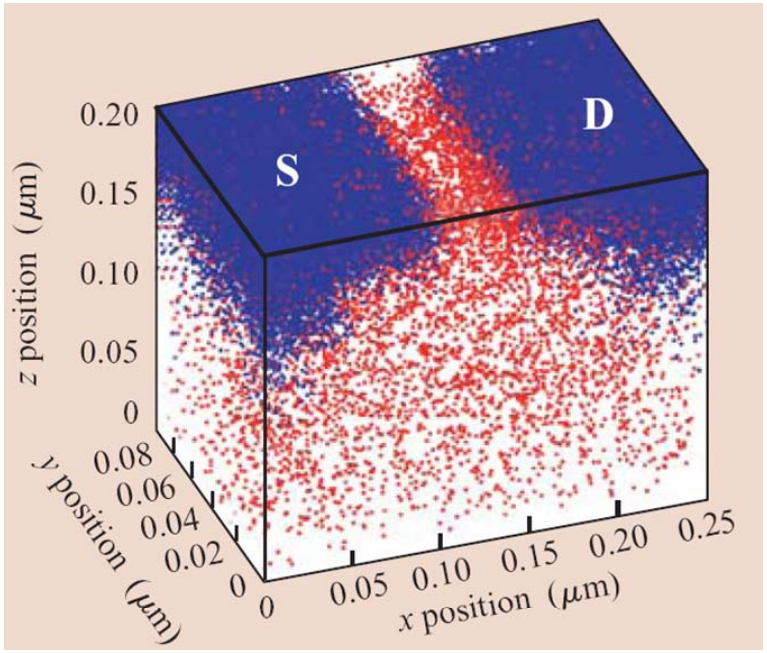
Seminar Part 2

PROCESS VARIATION TOLERANT 3T1D-BASED CACHE ARCHITECTURES

Process Variation



(Source: Friedberg, SPIE'06)

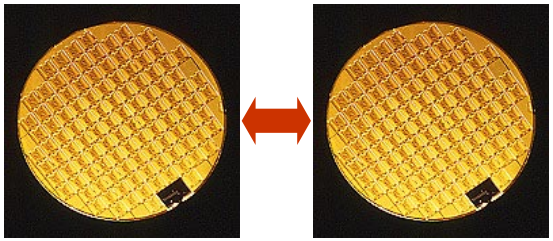


(Source: K. Bernstein,IBM J.R&D'06)

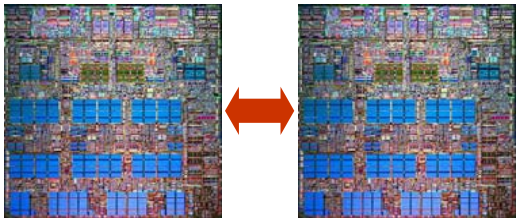
- As Moore's Law continues and on-chip dimensions get smaller, imperfections in the fabrication process affect device performance more and more…
- Past: Worried about wafer-to-wafer, chip-to-chip variations
- Now: Worry about within-die, transistor-to-transistor variations

Variability Trends

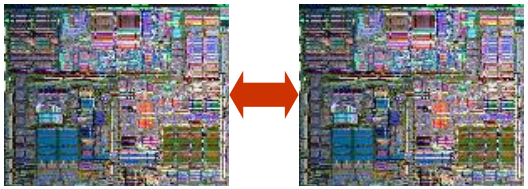
- In the past…



wafer to wafer

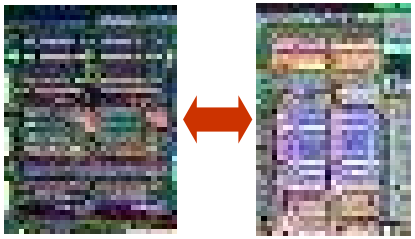


chip to chip

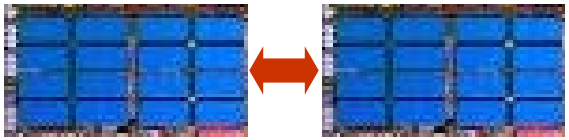


core to core

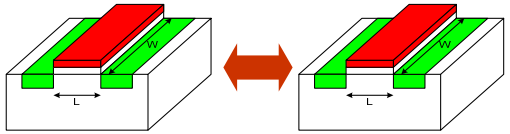
- Now…



block to block



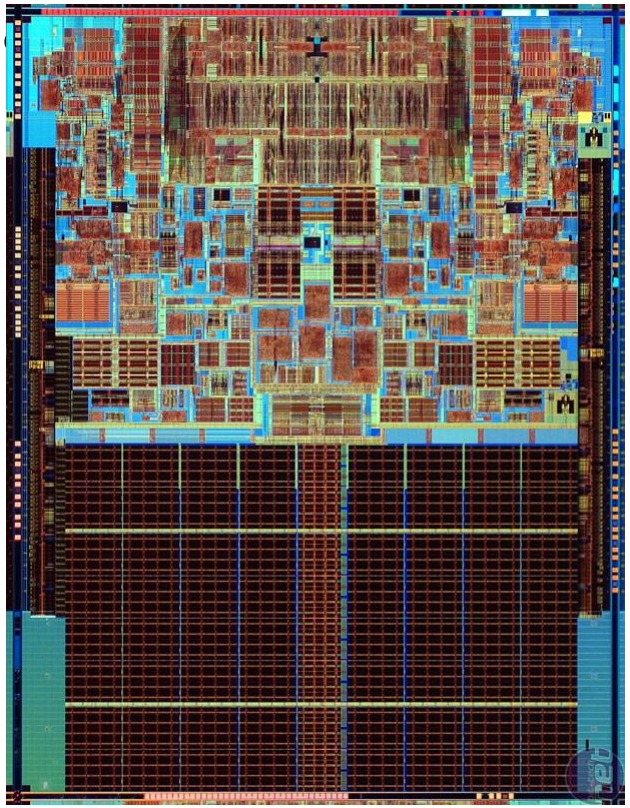
array to array



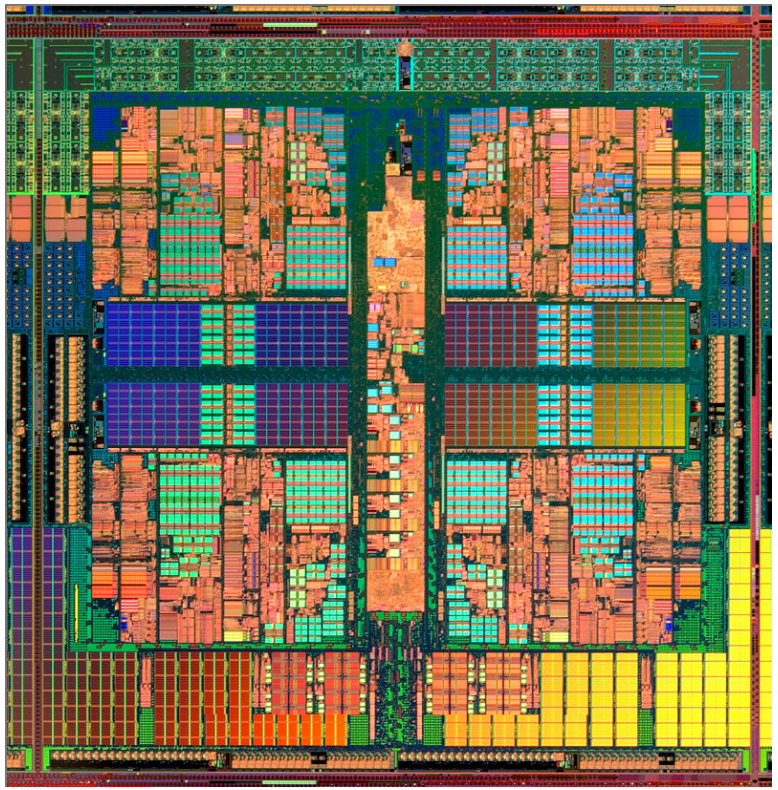
transistor to transistor

On-Chip Memory

- On-chip memory is a huge fraction of die



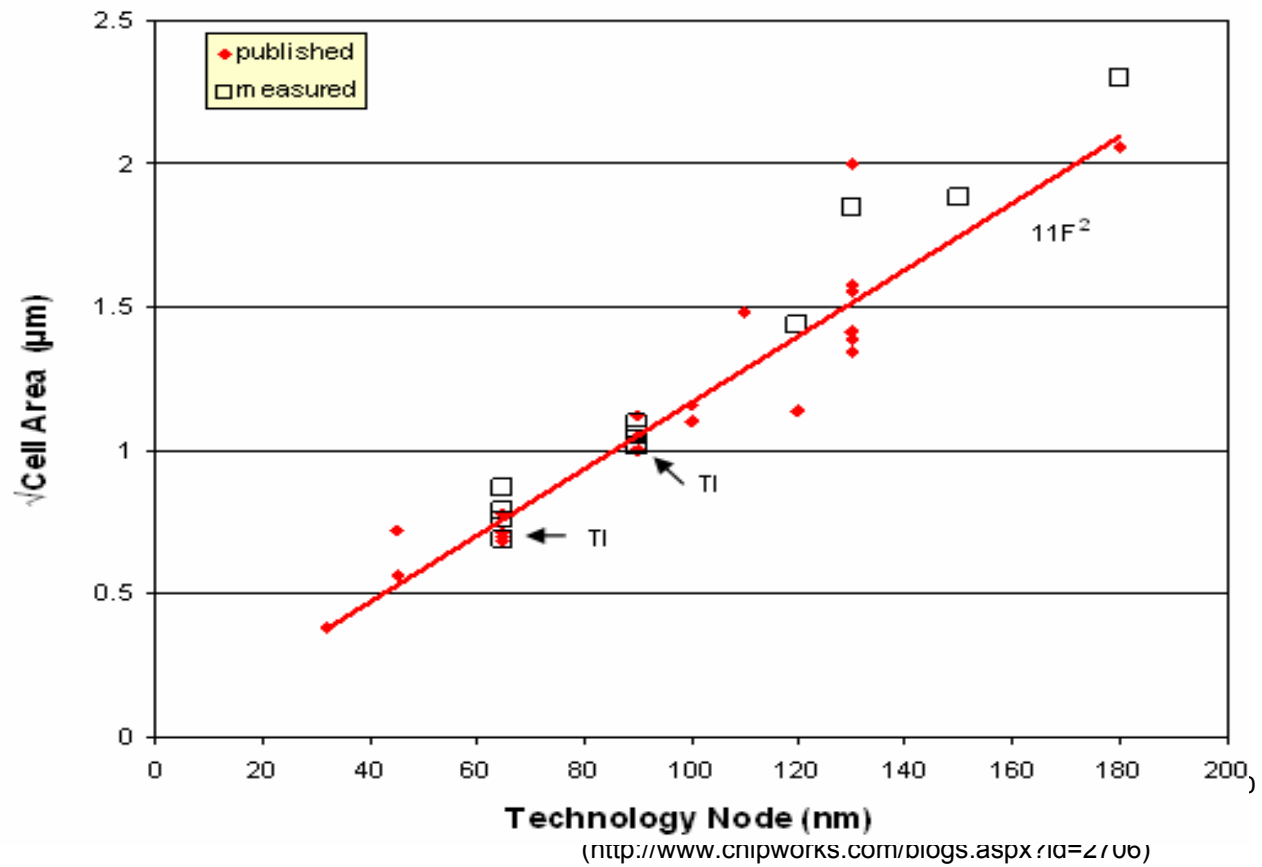
Intel Core2Duo



AMD Barcelona

SRAM scaling: A Tale of Two Conferences?

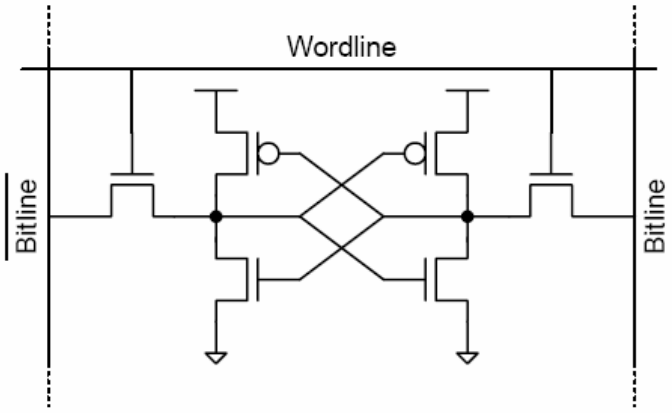
From IEDM
From ISSCC

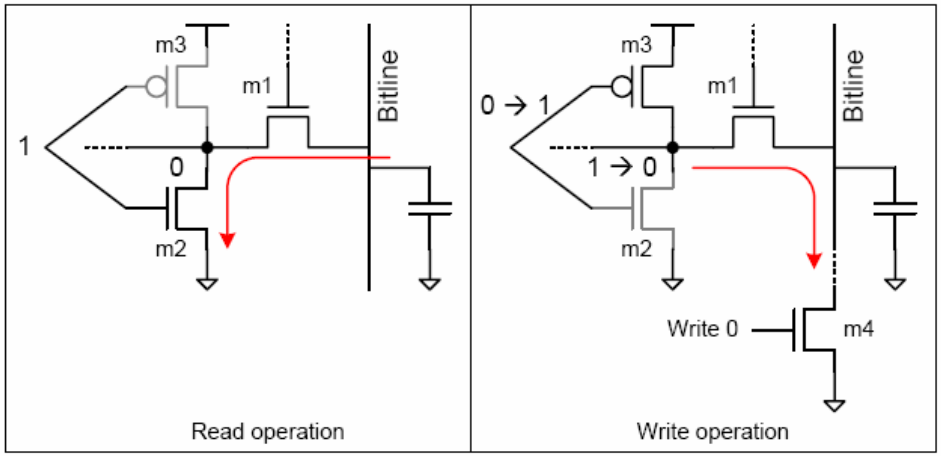


(http://www.chipworks.com/blogs.aspx?id=2706)

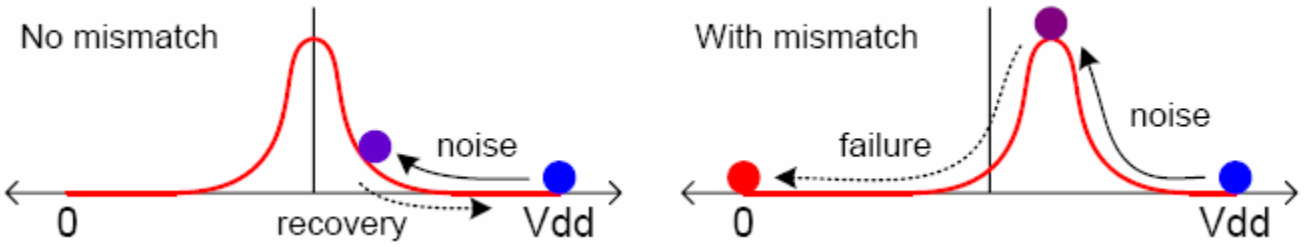
- Is SRAM scaling slowing down?
- Plots include circuit techniques to improve reliability (e.g., dual voltage, boosted WL, etc.)

Problems with 6T





- Susceptibility to process variations (PV)
  - Performance variations (Read/Write delay variations)
  - Bit flips due to voltage noise and leakage
  - Stuck at faults b/c too much mismatch



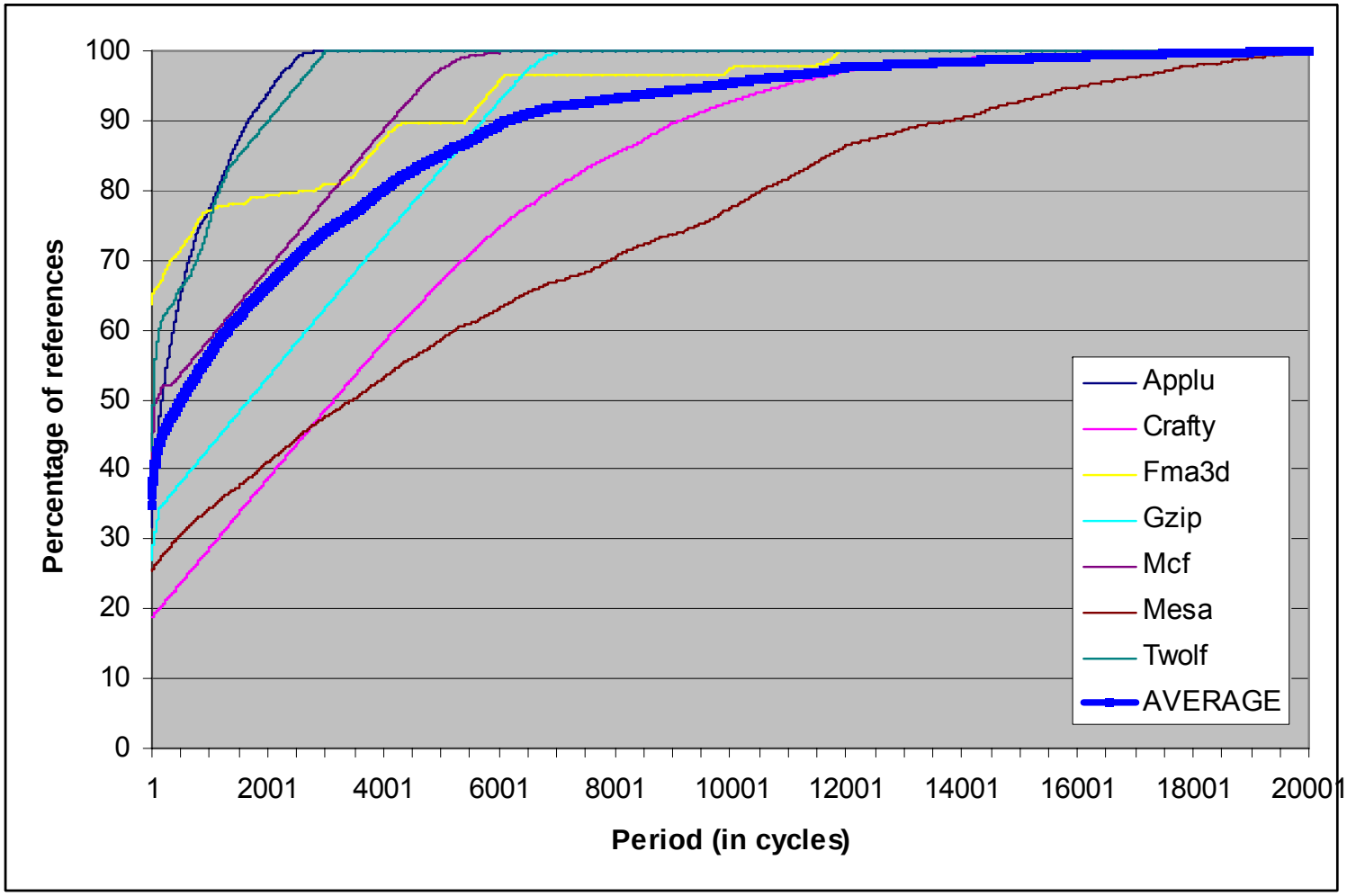
Dealing with variability in memories

- Microarchitectural techniques
  - Traditional ideas to deal with soft errors
    - Parity or ECC
    - Cache scrubbing
  - PVT-induced soft errors much more frequent than radiation-induced soft errors
    - Must understand the system-level issues
- What's the problem?
  - Fighting or feedback
    - Sensitive to mismatch

    →Boosted array or wordline voltage?
  - Bitline leakage
    - Large variations in leakage currents

    →Shorter bitlines?

Data Usage in L1

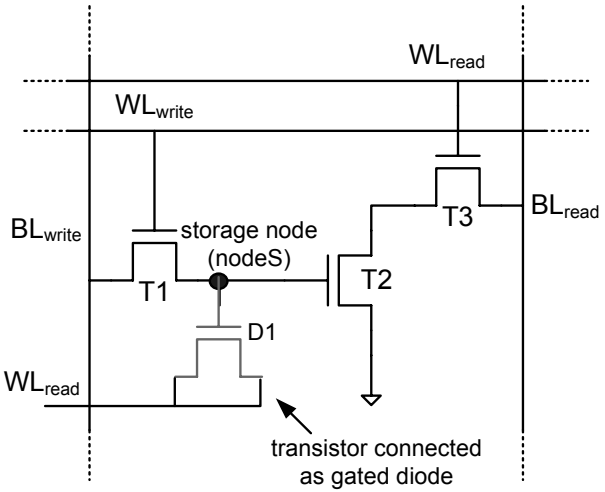


- On average, 90% of data accessed in first 6K cycles

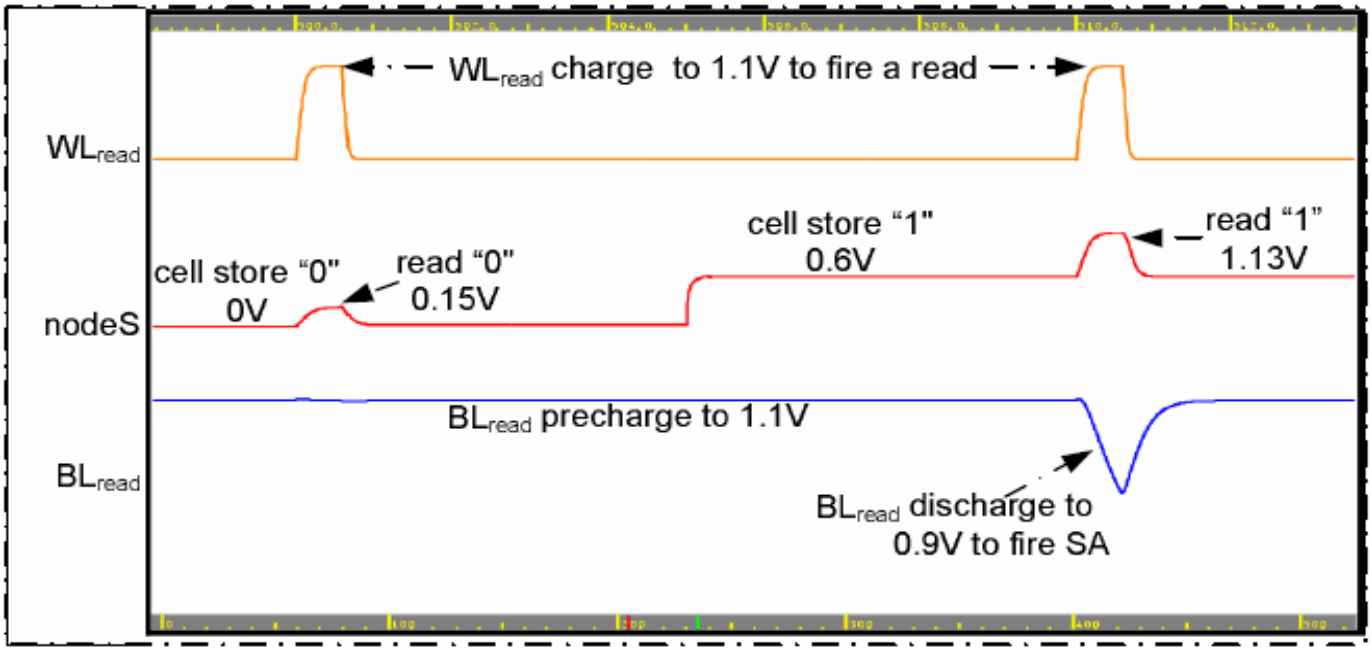
Proposed Solution

- Use 3T1D dynamic cells to replace 6T cells
  - W. K. Luk et al., "A 3-transistor dram cell with gated diode for enhanced speed and retention time," *Symp. on VLSI Circuits*, June 2006.
- Why?
  - Higher immunity to process variations
  - **Absorb delay variation into cell "retention time"**
  - No inherent fighting → no bit flips
  - Lower power (leakage and dynamic)
  - Higher density possible
- But what about refresh?
  - Use architectural insights and techniques to deal with dynamic data storage
- Where?
  - Analyzed register files (RF) and L1 data caches
  - eDRAMs being considered for L2 caches and above…

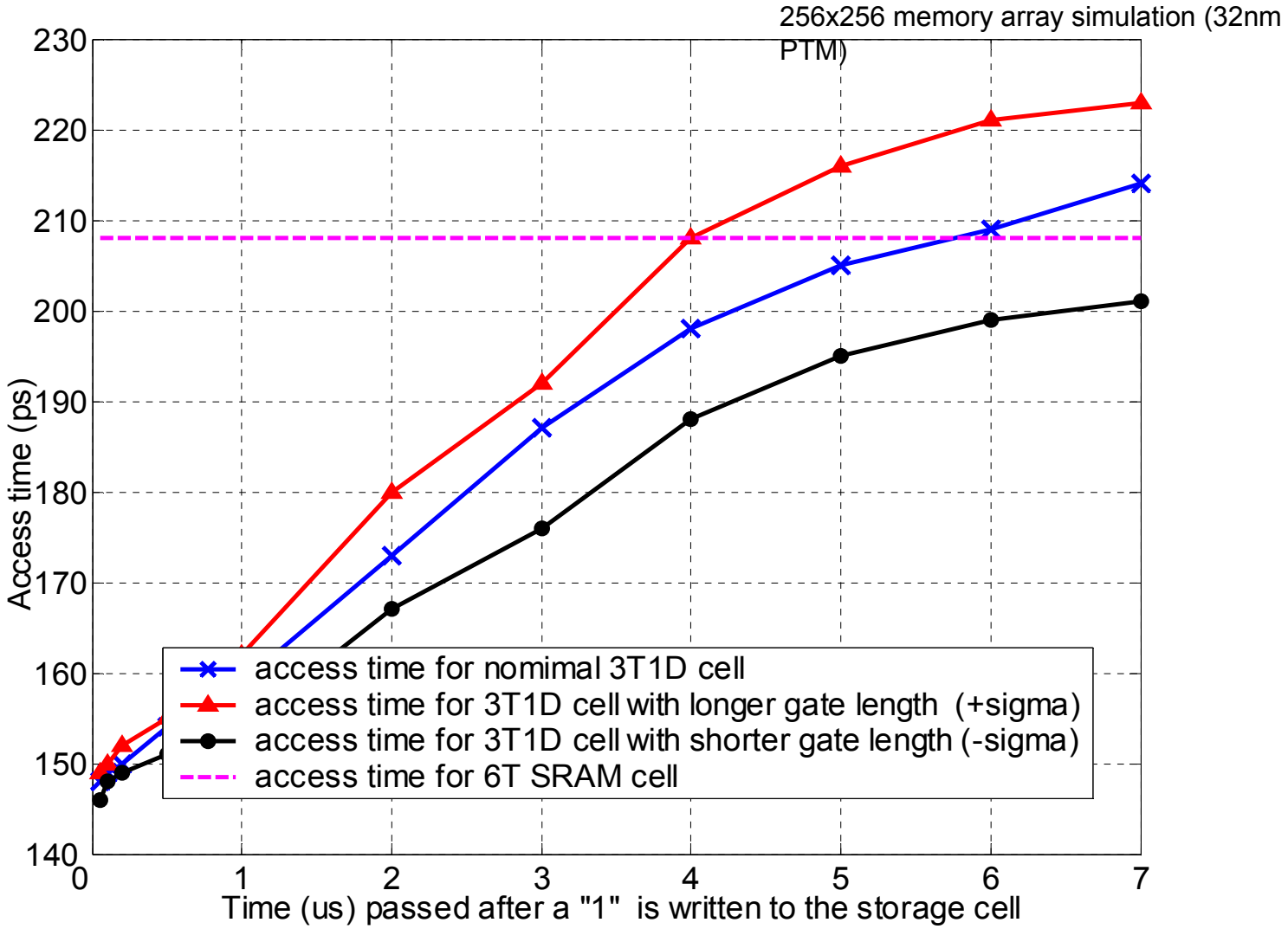
What is a 3T1D cell?



- Gated-diode selectively boosts stored data (“1”) during reads
- Non-destructive reads allows for column multiplexing



Retention Time vs. Access Time



- What retention time is “good enough”?

Simulation Setup

- Baseline: 4-wide Out-of-order machine
  - 20FO4 pipelines
  - 80-entry RF
  - 64KB, 4-way set-associative I- and D-caches
- *sim-alpha* simulator used to calculate instructions per cycle (IPC)
- 8 SPEC2000 benchmarks

Variation Model

- Monte Carlo analysis of process variation impact on memory cell delay and power
  - 32nm PTM, Vdd = 1V
  - Considered typical and extreme PV scenarios
  - Correlations based on Friedberg's chip measurements

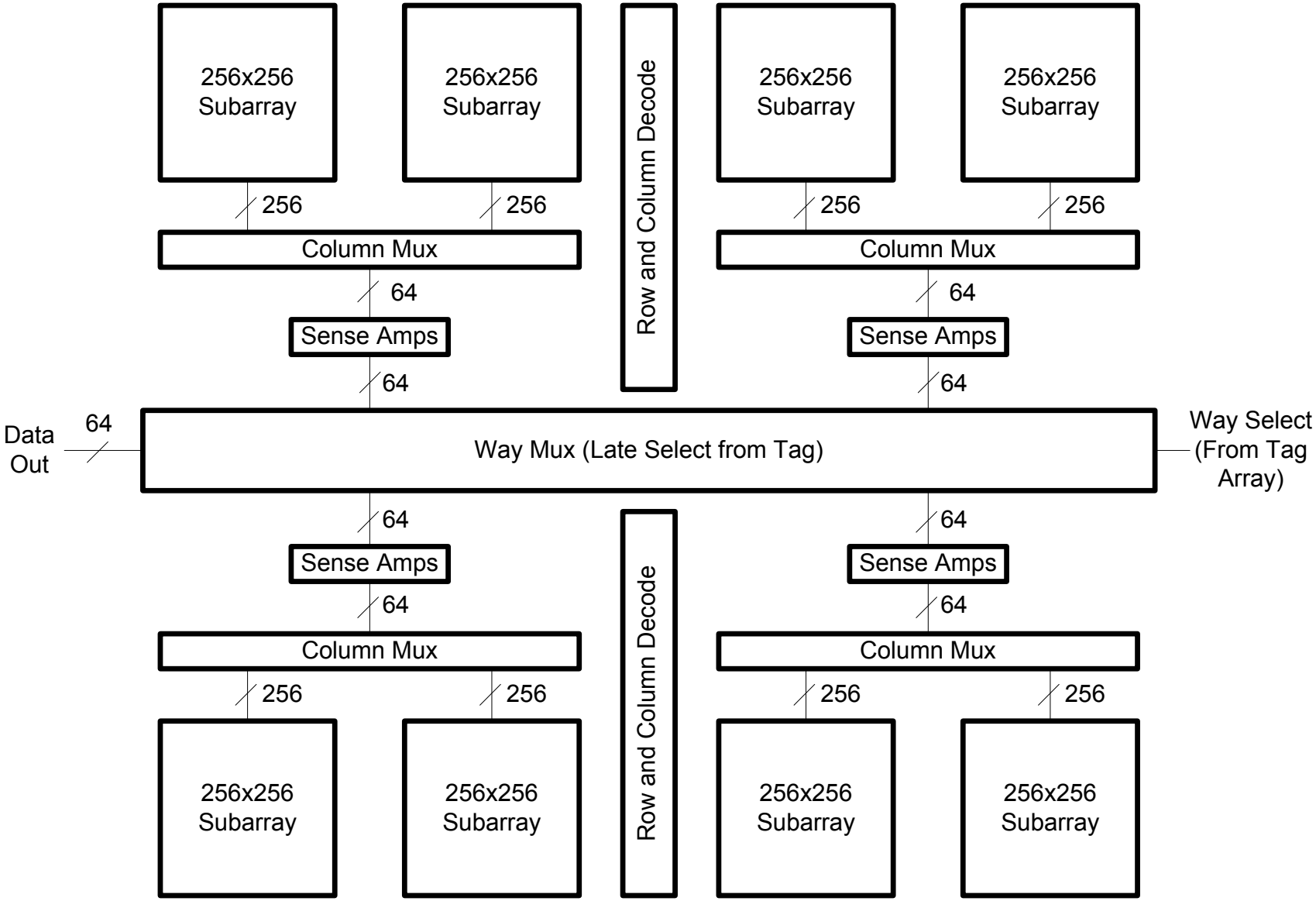
	Typical	Severe
$\sigma L/L_{nominal}$ (WID)	5%	7%
$\sigma V_{th}/V_{th}$ (WID)	10%	15%
$\sigma L/L_{nominal}$ (D2D)	5%	5%

Cache Configuration

- 64KB cache
  - 4-way Set Associative, 512b cache lines
  - 2 Read/1 Write ports
  - 8 256x256 subarrays
  - 64 Sense Amps per subarray

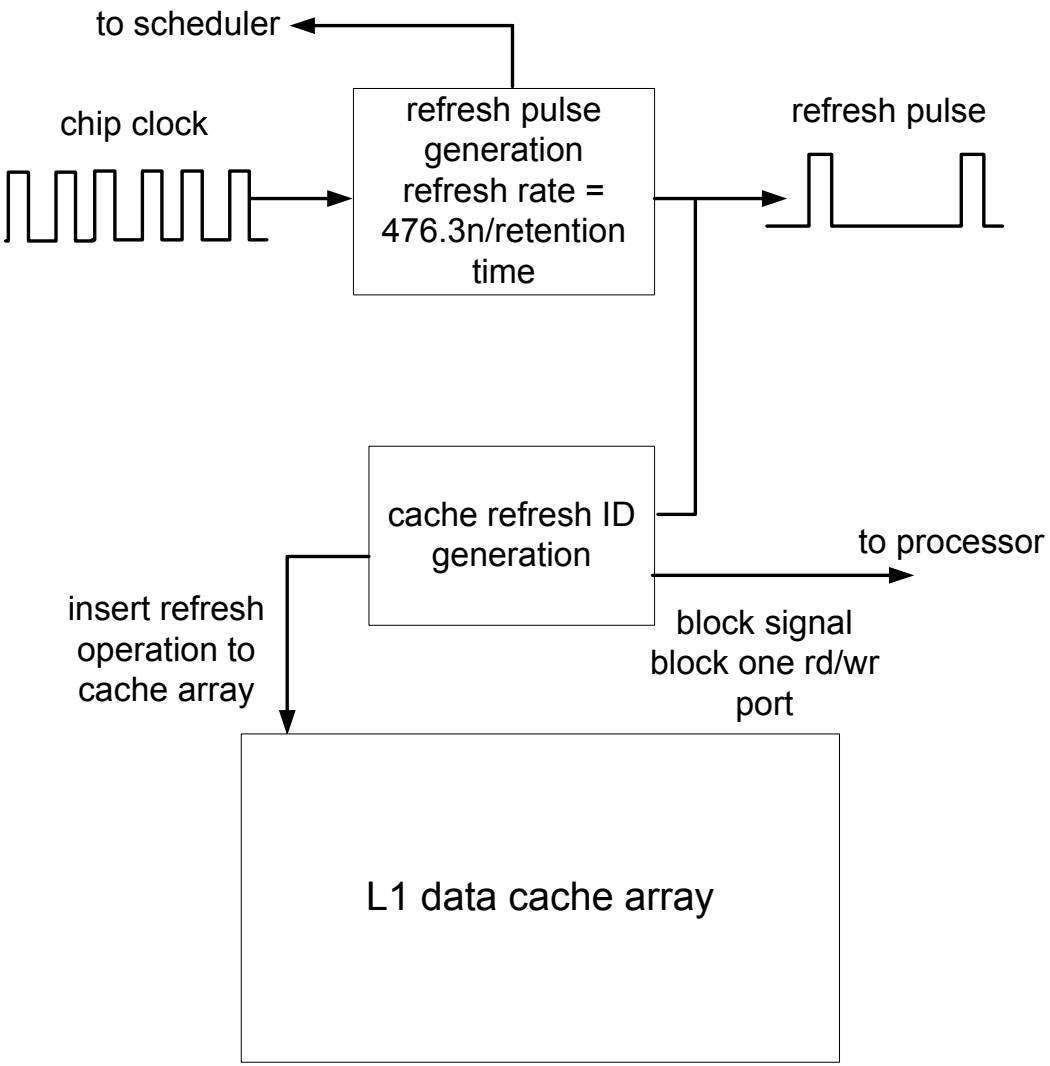
Three Stage Cache Access

Cache Data Array Floorplan

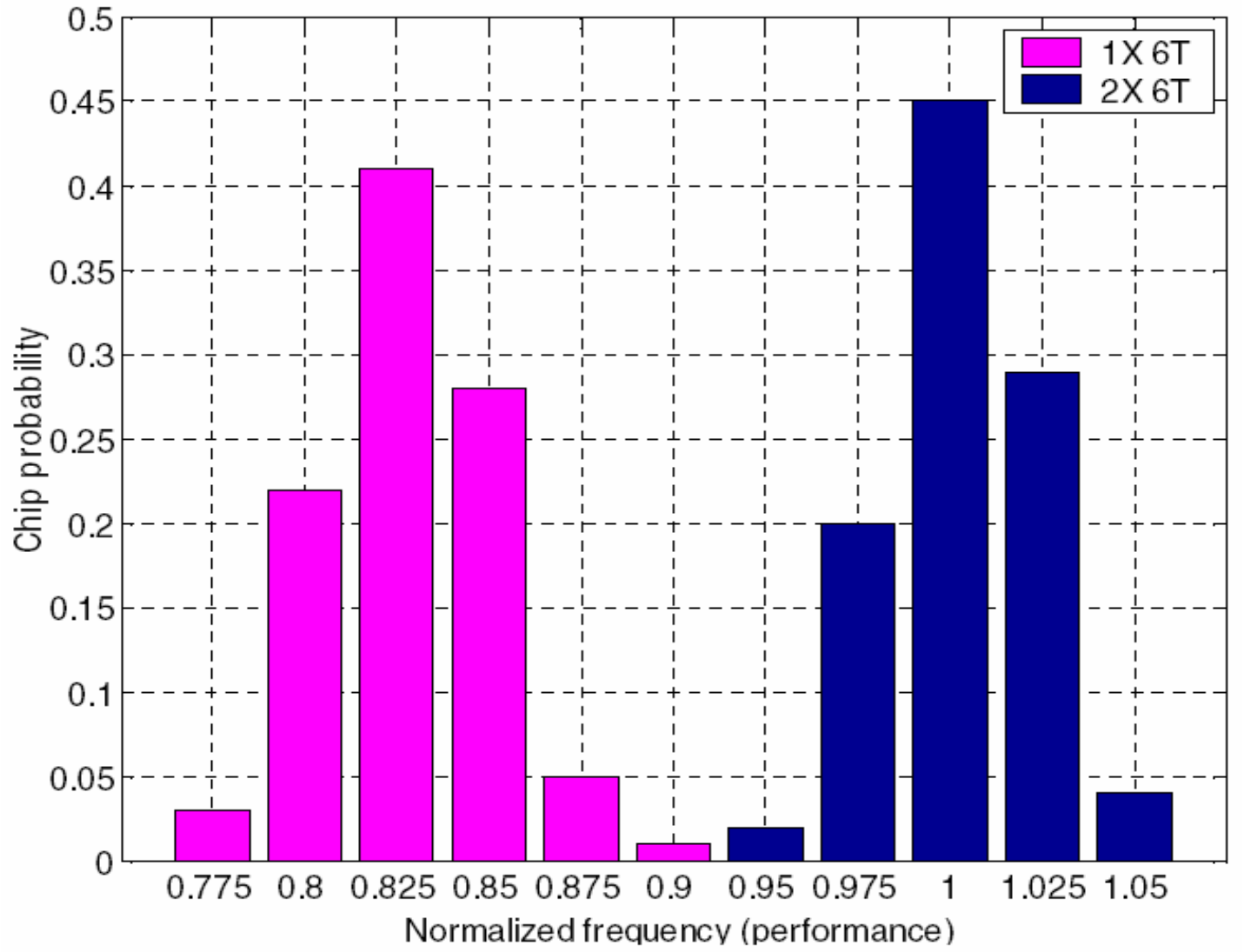


Global Refresh Scheme

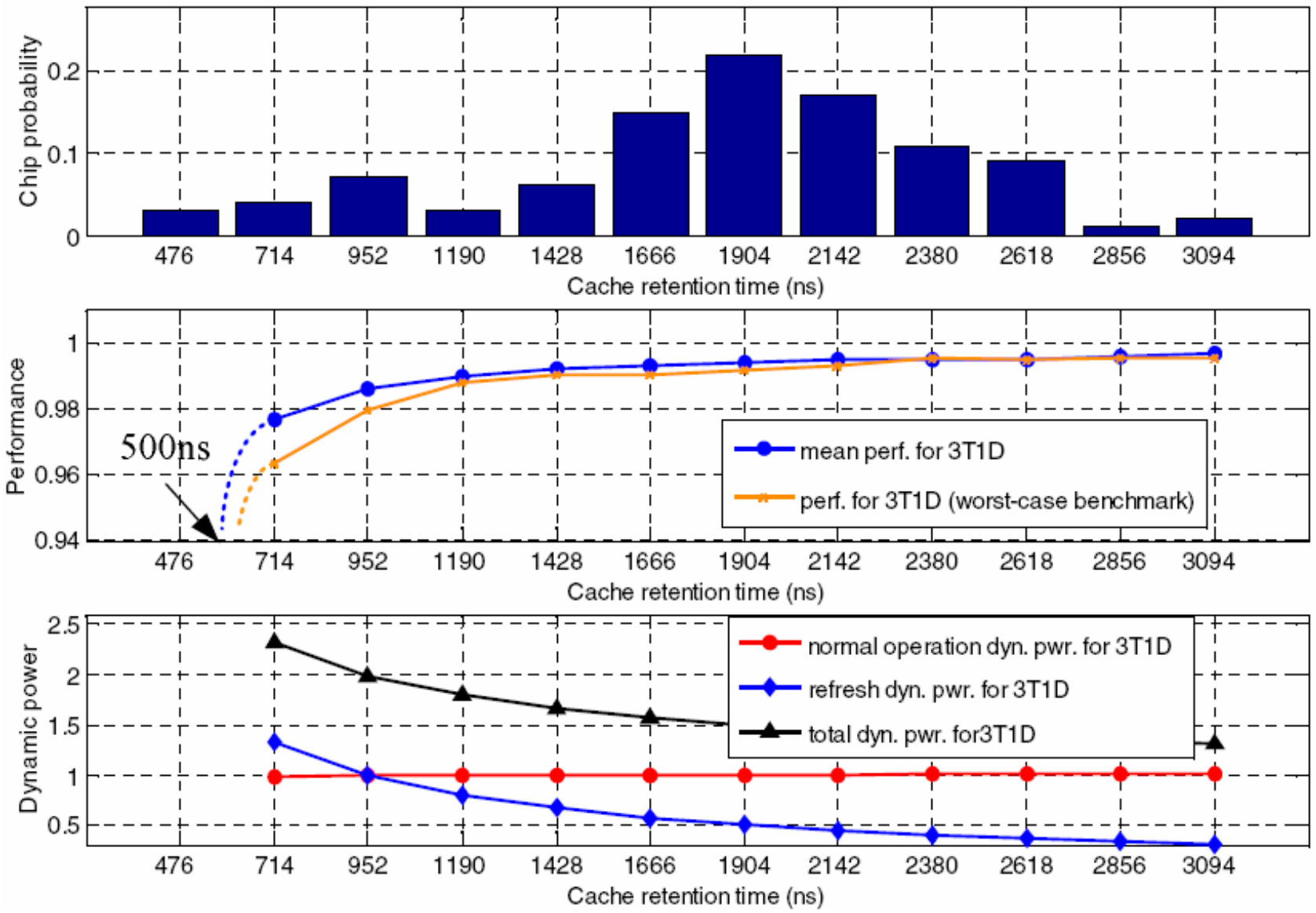
- 8 cycles to refresh one cache line (SA-limited)
- 2K cycles to refresh entire cache (476ns @ 4.3GHz)
- ~6µs retention time (no variations)
- Refresh takes 8% of cache bandwidth
- IPC hit < 1%



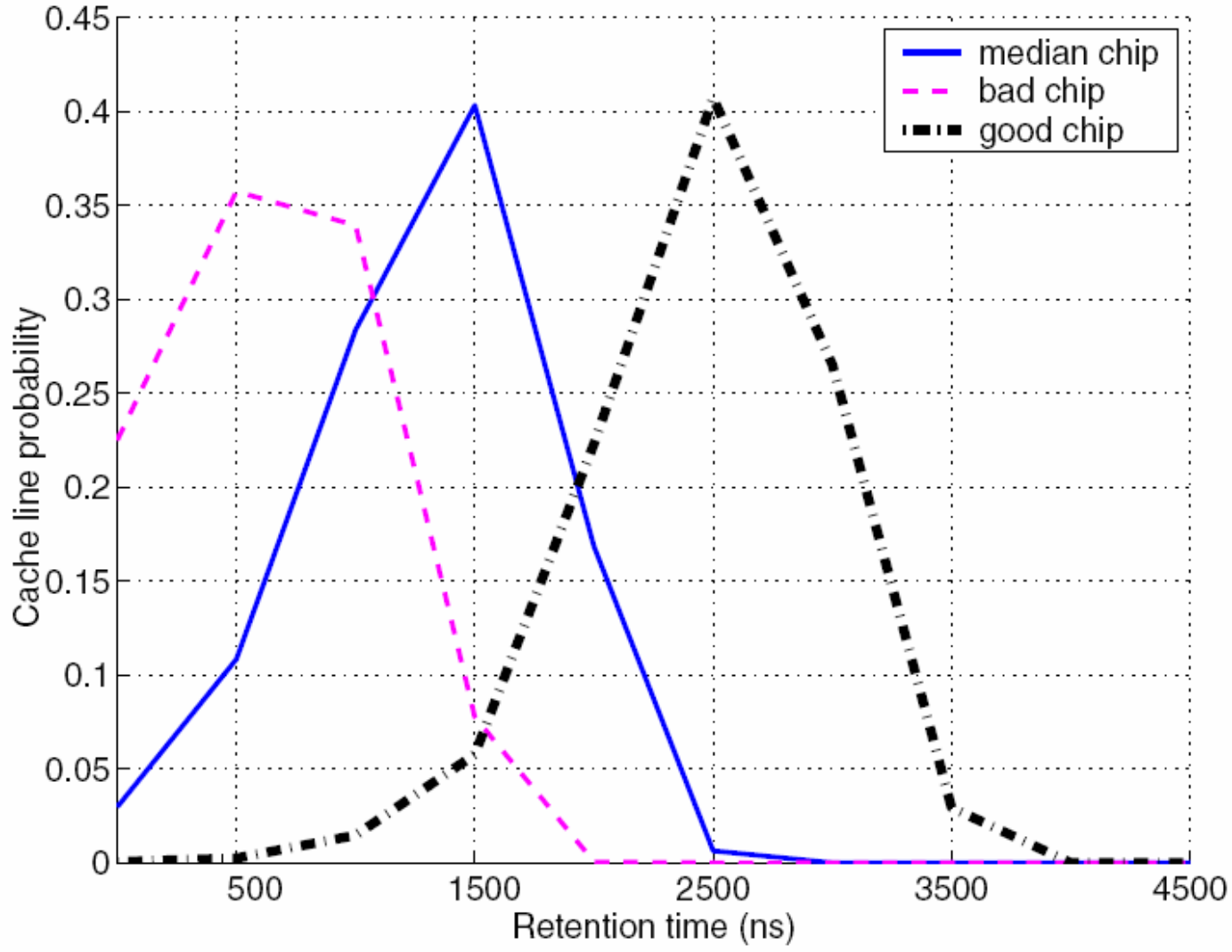
6T Performance under typical variations



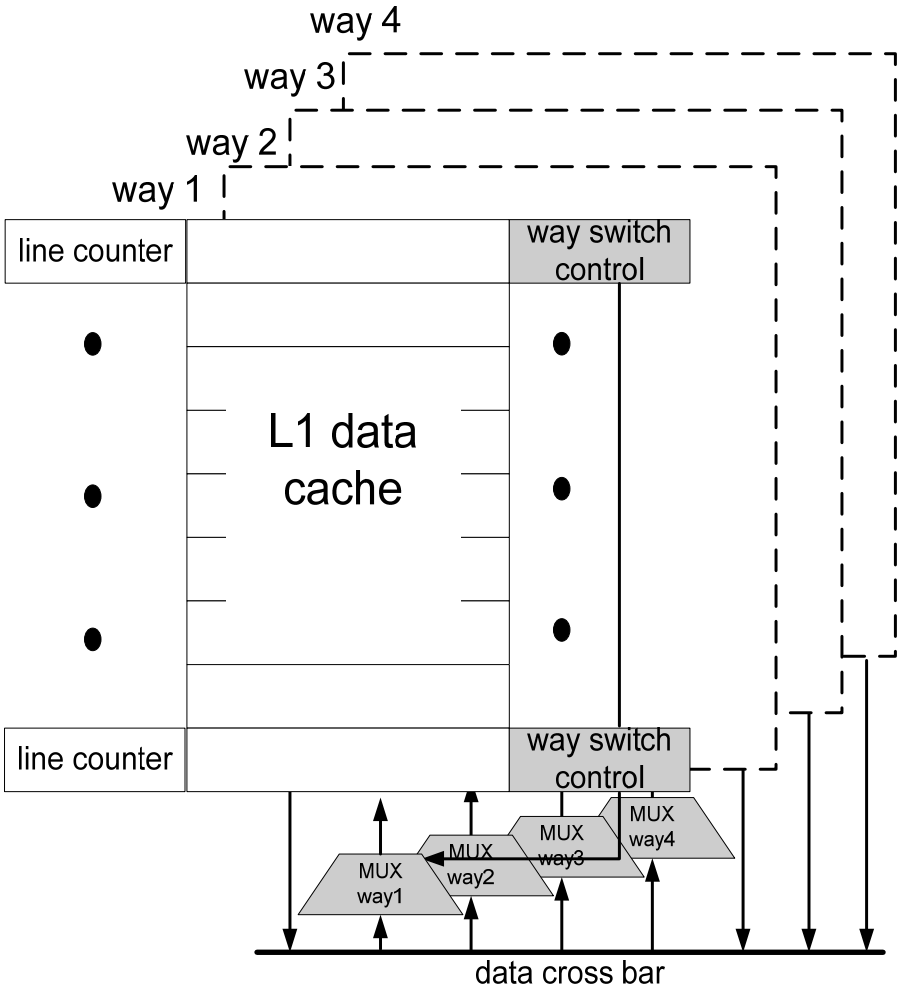
3T Performance under typical variations



Three chips under severe variations



Line-Level Schemes: Refresh Policies

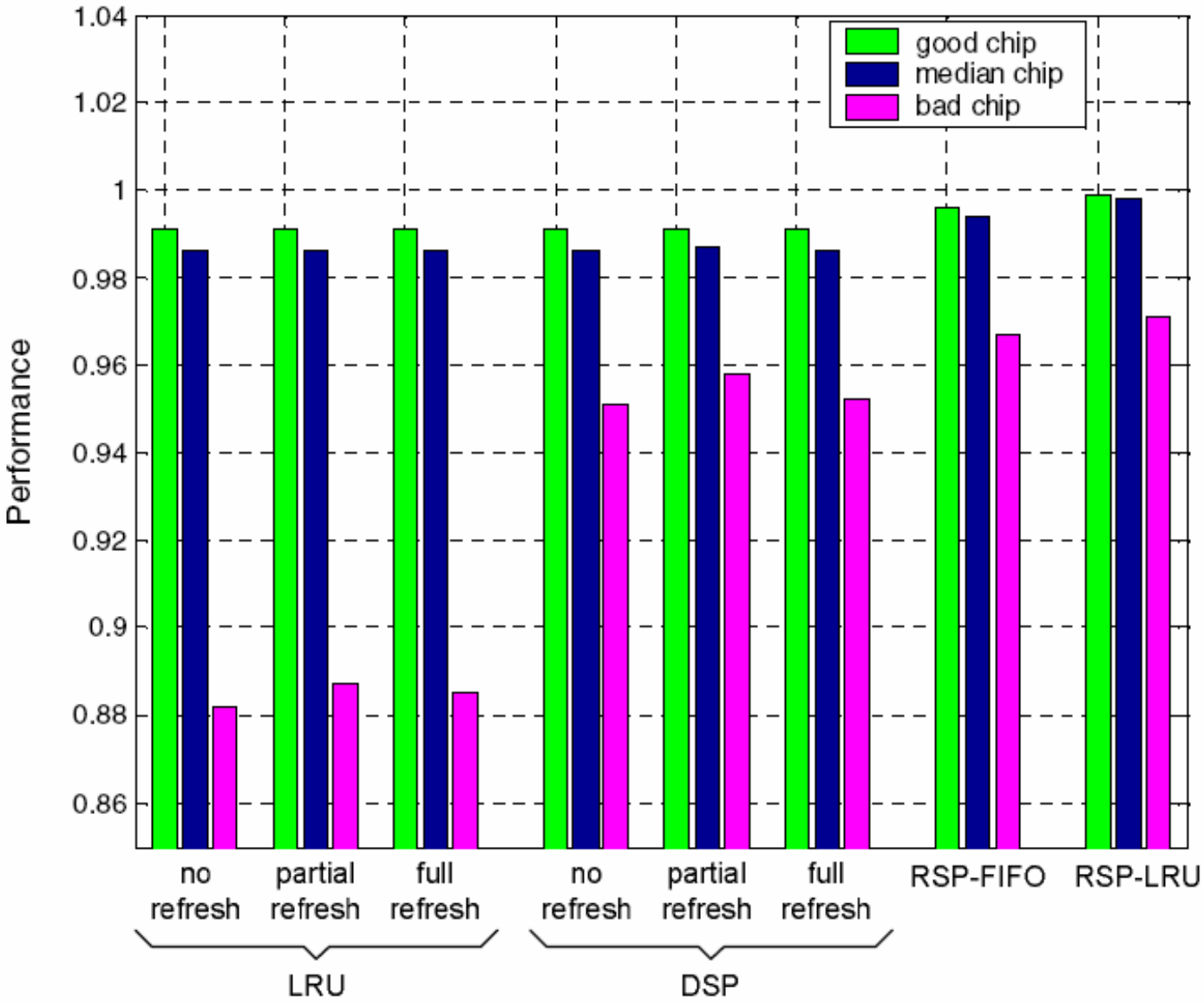


- Refresh Policies
  - Full-refresh: Per-line counter forces refresh when needed
  - No-refresh: Rely on L2 inclusion properties
  - Partial-refresh: Threshold counter chooses one of the two policies

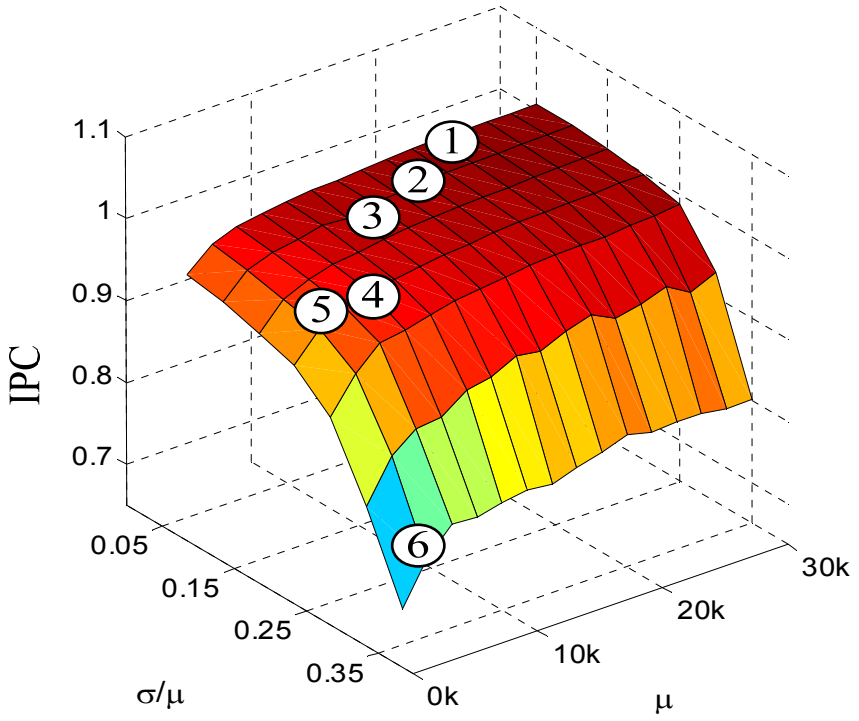
Line-Level Schemes: Replacement Policies

- Replacement Policies
  - Dead-sensitive Placement
    - Avoid using “dead” lines when performing placement
  - Retention-sensitive placement (RSP-FIFO)
    - Order lines in descending retention time
    - New lines are assigned the longest retention time line (and old ones reshuffle)
  - Retention-sensitive placement (RSP-LRU)
    - MRU block is assigned the longest retention time

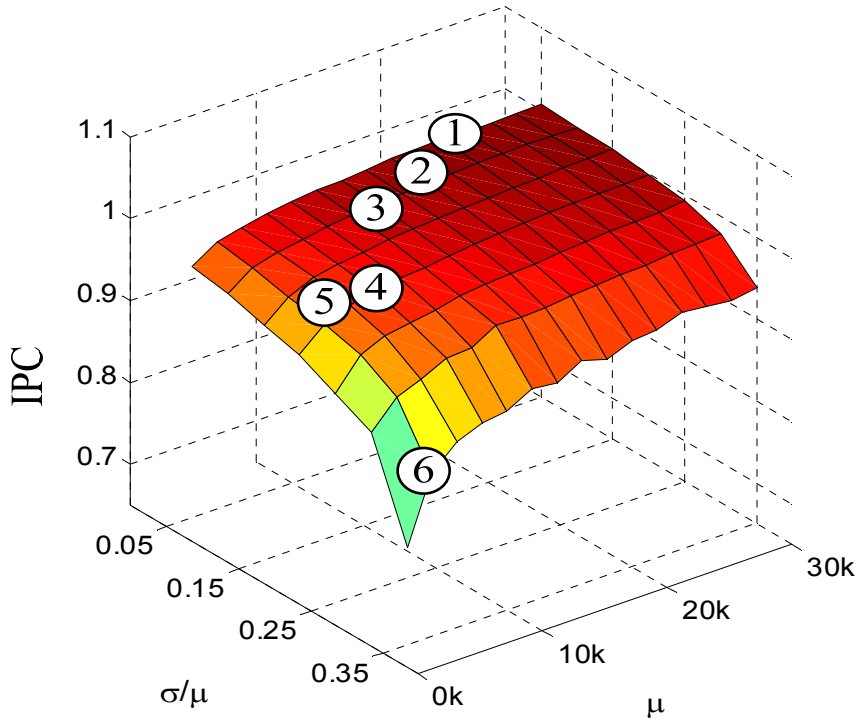
Evaluating Policies



Pushing policies to the limits



no-refresh/LRU

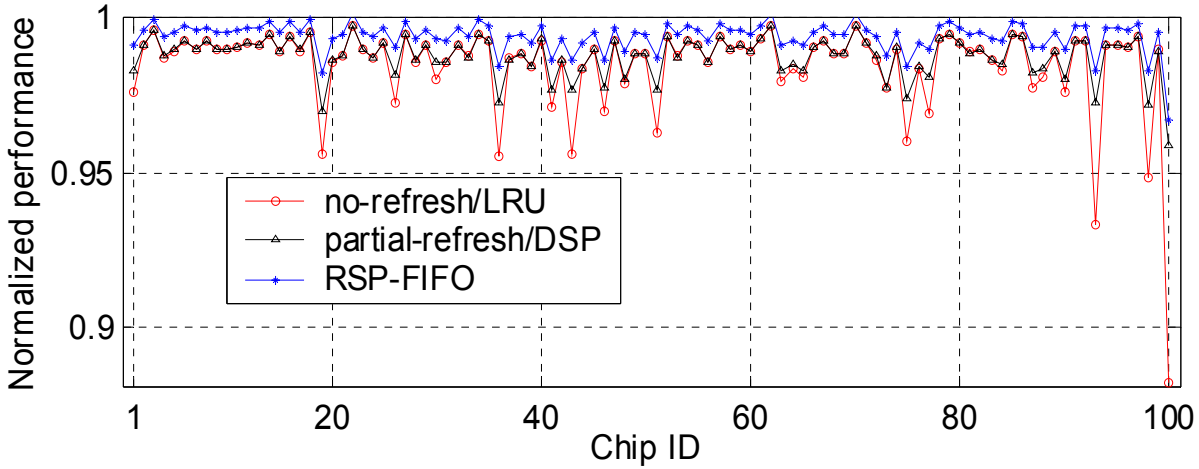


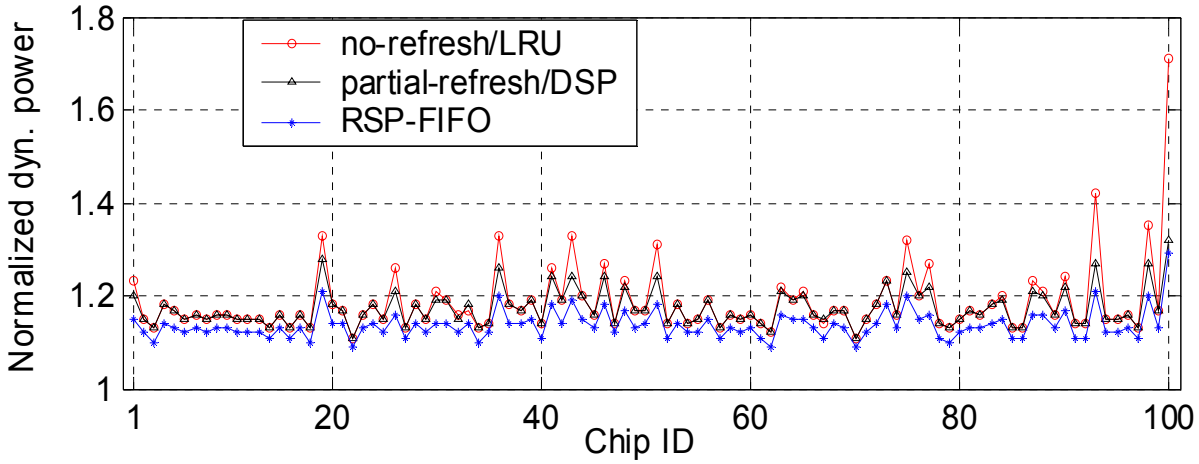
partial-refresh/DSP

1. 65nm, typical, 1.1V
2. 45nm, typical, 1.1V
3. 32nm, typical, 1.1V
4. 32nm, severe, 1.1V
5. 32nm, typical, 0.9V
6. 32nm, severe, 0.9V

Power Analysis (Dynamic)

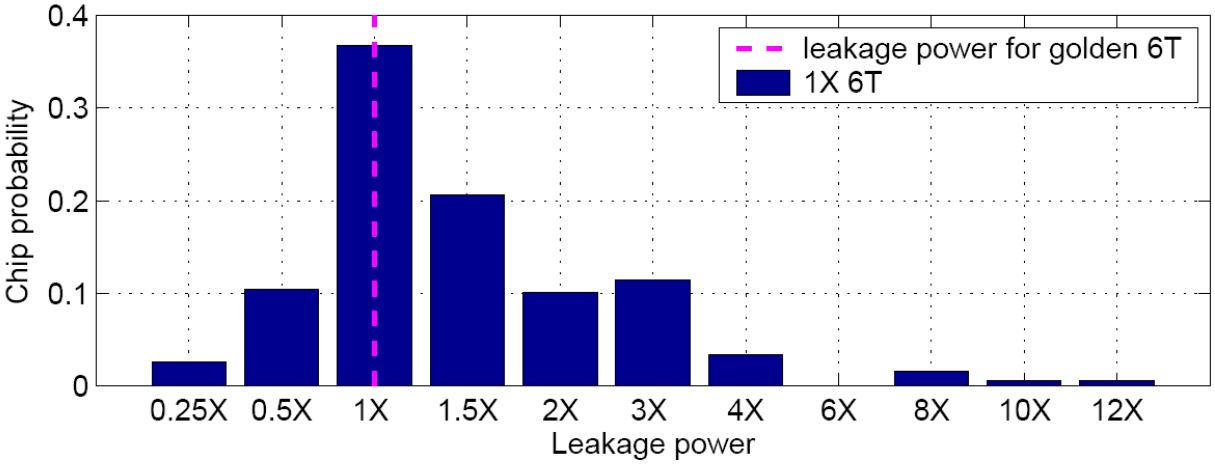
- Refresh power is small (~10% overhead for better schemes)



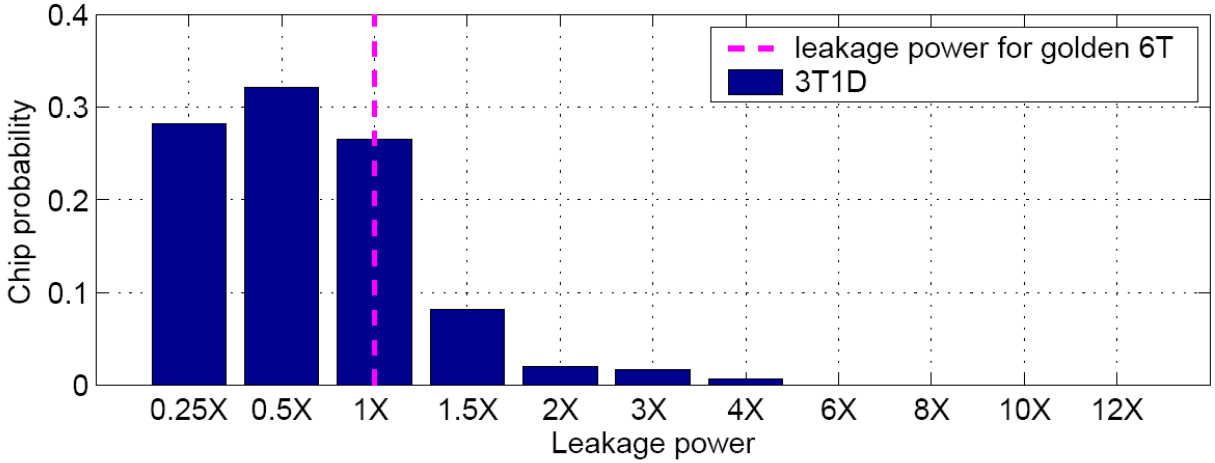


Power Analysis (Leakage)

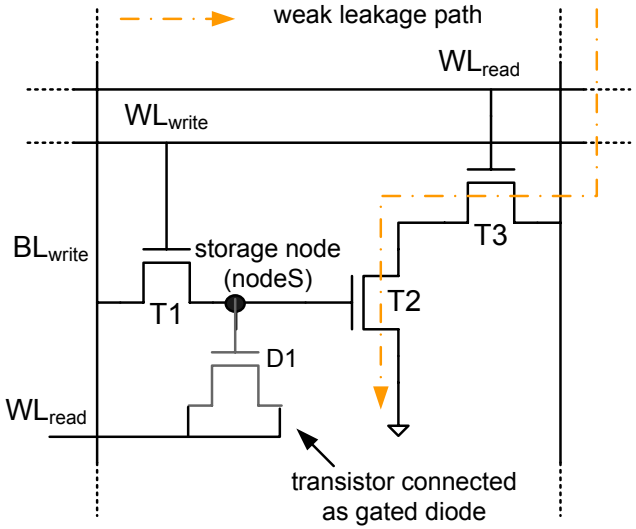
- Substantial leakage savings



a. Cache leakage power distribution for 1X 6T



b. Cache leakage power distribution for 3T1D



Reliable Memory Summary

- Transient nature of data in L1 cache allows for architecturally-simple refresh schemes for 3T1D memories
- Provides PV-tolerant on-chip memories
  - Comparable performance to “ideal” 6T
  - Lower leakage power
  - Low HW overhead
- Similar results observed for 3T1D register files and instruction caches
- Test chip planned for fab in Spring 2008