

# Performance/Power Trade-Offs of Bitline Isolation

Se-Hyun Yang and Babak Falsafi

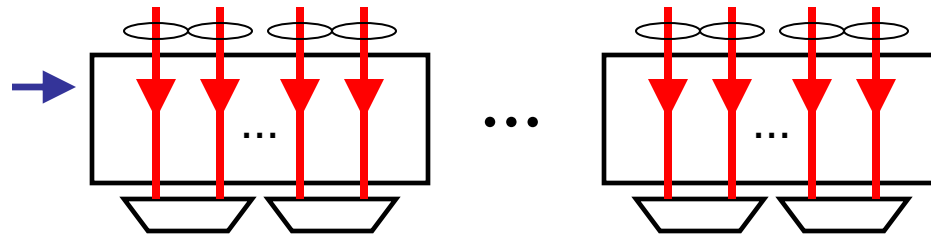
Computer Architecture Lab at Carnegie Mellon  
Electrical and Computer Engineering  
Carnegie Mellon University

# High Bitline Discharge in Caches

---

Deep sub $\mu$  high-performance caches

- Use subarrays
- Precharge entire caches statically
- No precharging delay exposed



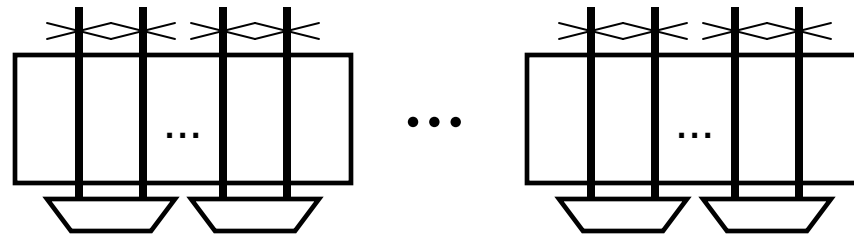
Large discharge from subarrays

# Bitline Isolation

---

Stop discharge by cutting off  $V_{dd}$ -bitline path

- A.k.a. leakage biased bitlines
- Turn off precharge devices

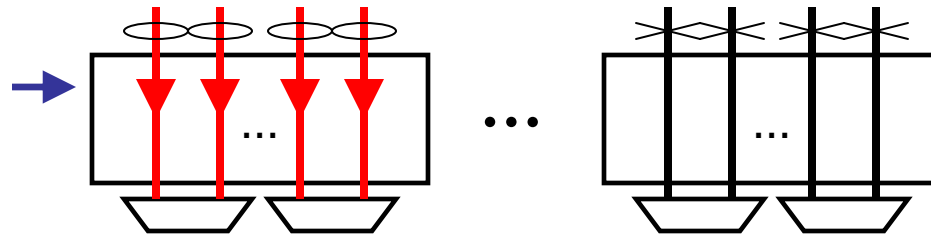


Need selective mechanisms to control

# Per-access Precharging Control

Ideally, best for energy saving

- All bitlines isolated initially
- Precharge only accessed subarrays
- On-demand wakeup using partial decoding



Can be done for free? Energy cost, Timeliness

# Contributions

---

## Bitline Isolation

- Energy: Large cost before, not in the future  
Per-access control viable in the future
- Performance: On-demand wakeup is late  
Early precharging is required
- Ideal early precharging Vs. Resizable caches  
Large opportunity (74%) for per-access control

# Methodology

---

CACTI 3.0 and SPICE simulations

- 180nm-2V, 130nm-1.7V, 100nm-1.3V, 70nm-1V

Highly modified Wattch 1.0

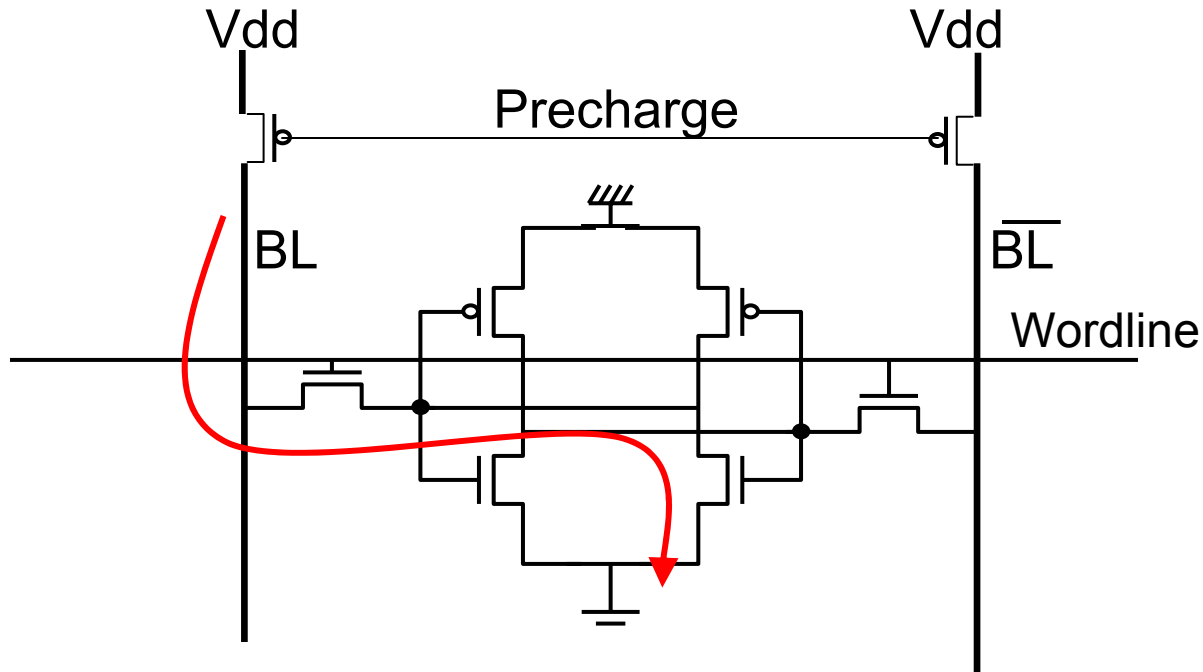
- 12 SPEC benchmarks
- 8-wide 64 Issue queue w/ 128 Active list
- 32KB 2-way set associative L1 caches

# Outline

---

- Introduction
- Methodology
- Energy Overhead
- Performance Overhead
- Per-access Vs. Resizable Caches
- Conclusions

# Bitline Leakage in SRAM cell



Leakage occurs in all subarrays

Bitline isolation: turn off bitline devices

# Sources of Energy Overhead

---

Switching precharge devices

Charging up discharged bitlines

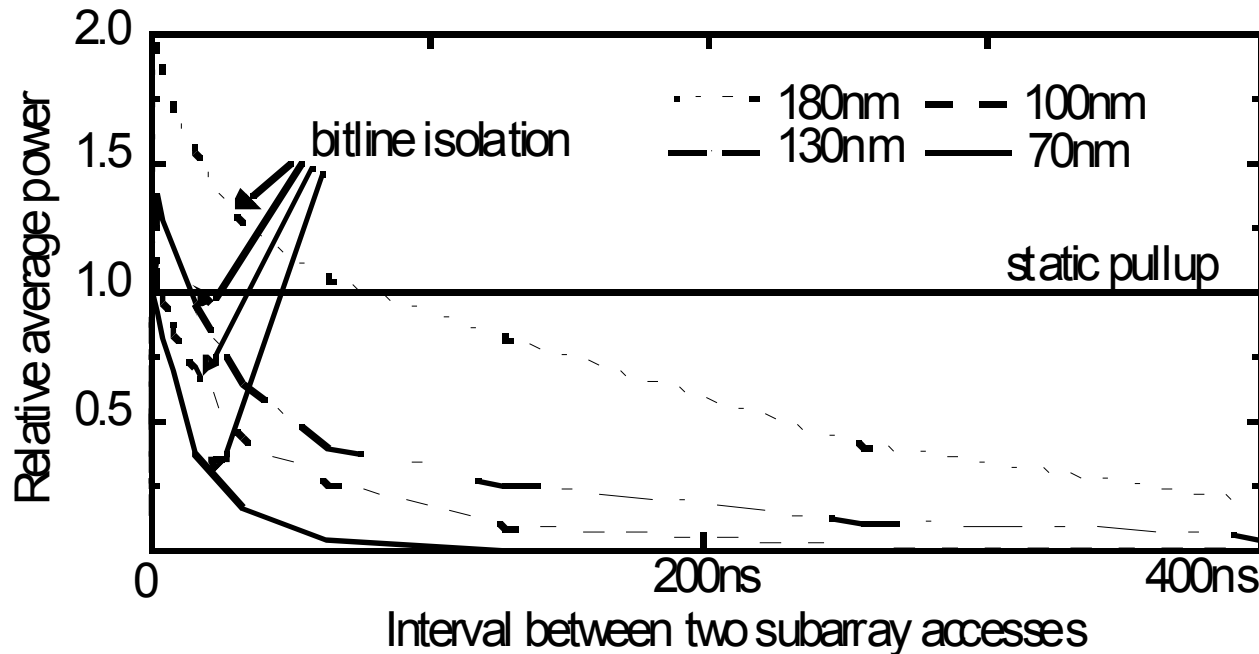
# Implications

---

What affects energy overhead?

- CMOS technology: Relatively larger wire cap
- Precharge device size
  - Resistive load between  $V_{dd}$ -bitlines on cell read
  - Fast pull-up
- Subarray size
- Discharging time: average cache access interval

# Energy Overhead: Results



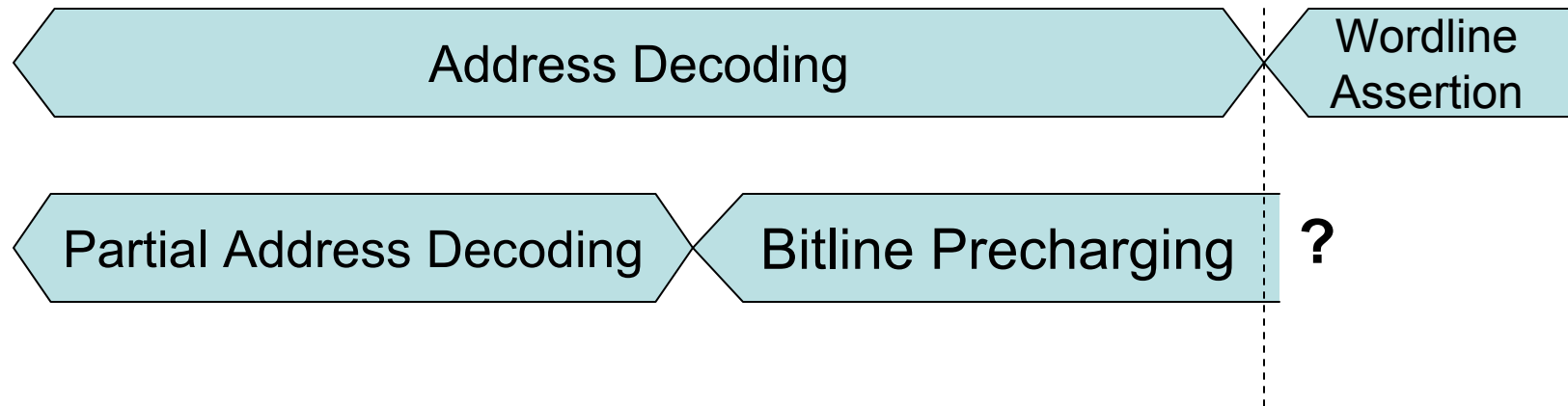
Bitline isolation energy effective in the future

# Performance Impact

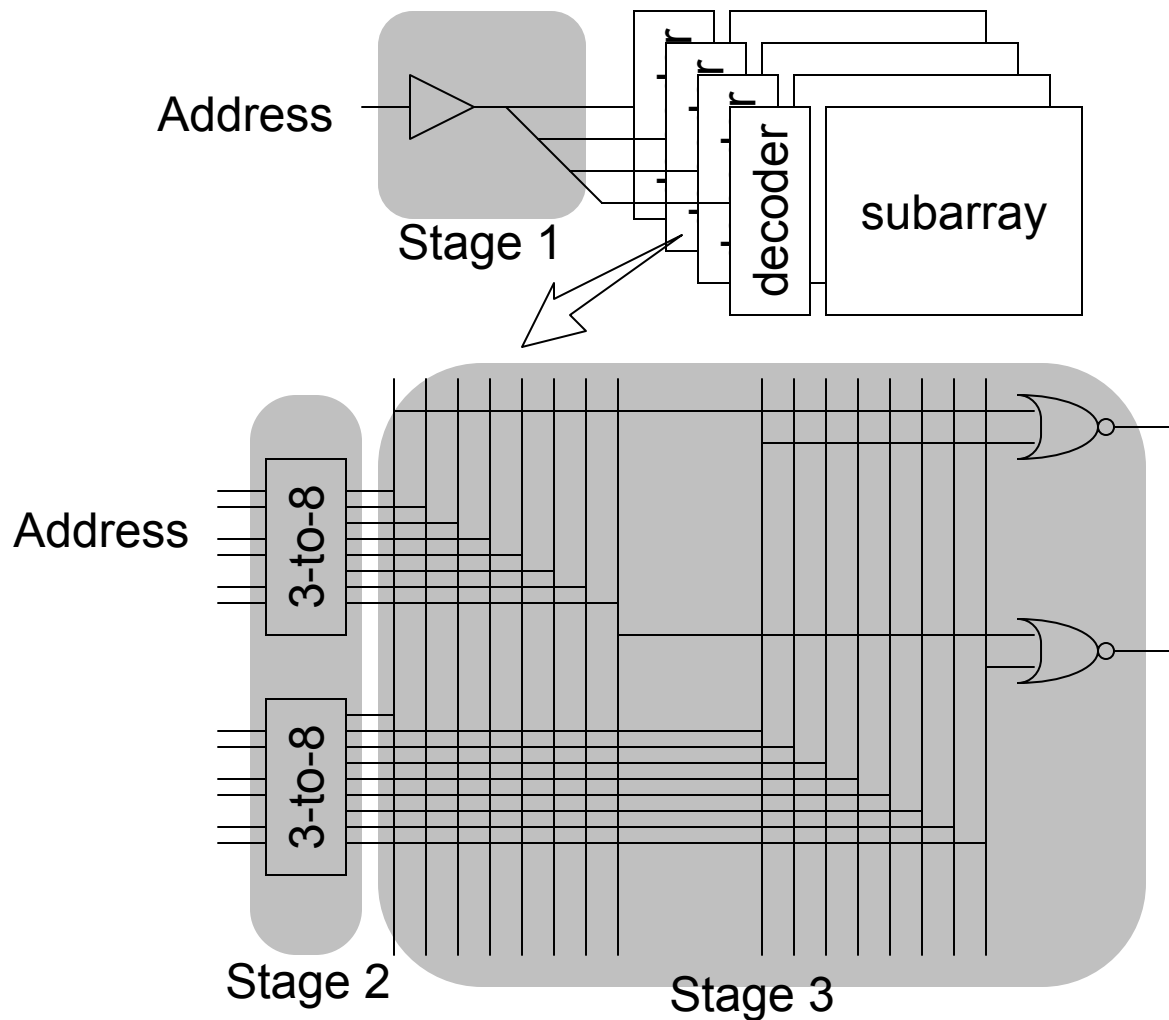
---

## On-demand precharging

- Precharge only accessed subarrays
- On-demand wakeup using partial decoding



# Cache Decoder Architecture



# Implications

---

What affects the delay?

- Precharging delay
  - CMOS technology: Longer wire delay
  - Size of subarray
- Partial address decoding
  - # of subarrays: More bits for indentifying subarray

# Performance Impact: Results

Subarray size	Feature Size (nm)	Stage 3 Delay (ns)	Bitline precharge(ns)
1KB 32-row	180	0.15	0.39
	130	0.13	0.31
	100	0.09	0.24
	70	0.06	0.16
4KB 128-row	180	0.18	0.50
	130	0.13	0.36
	100	0.10	0.28
	70	0.07	0.19

Early precharging is desirable

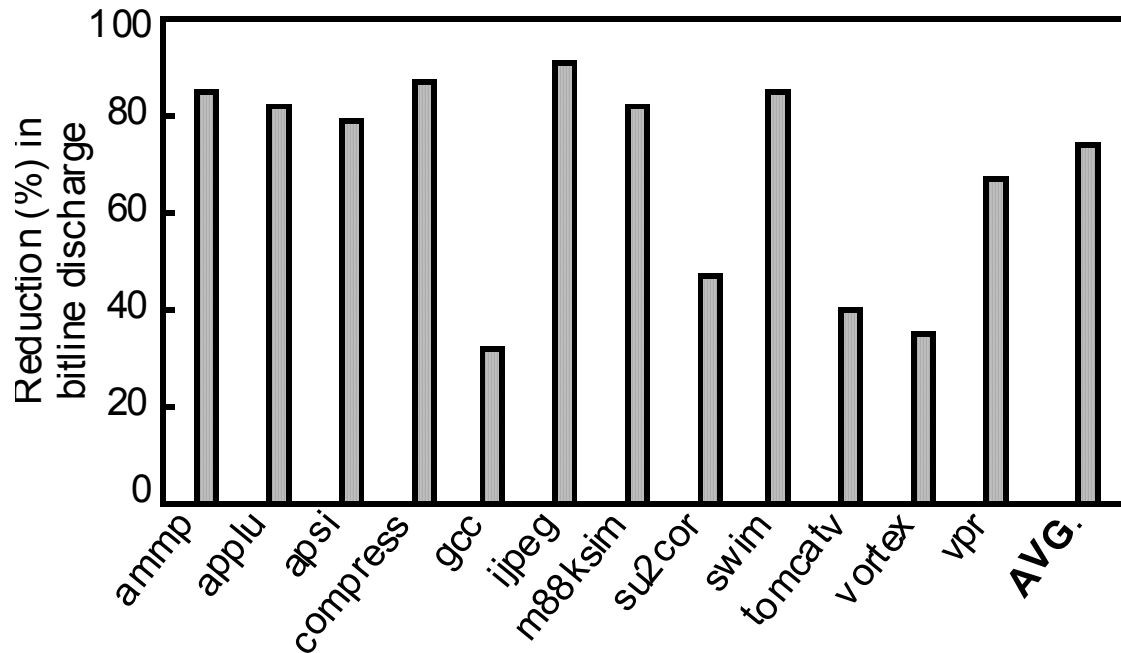
# Per-Access Vs. Resizable Caches

---

Resizable caches [Albonesi][Yang et. al]

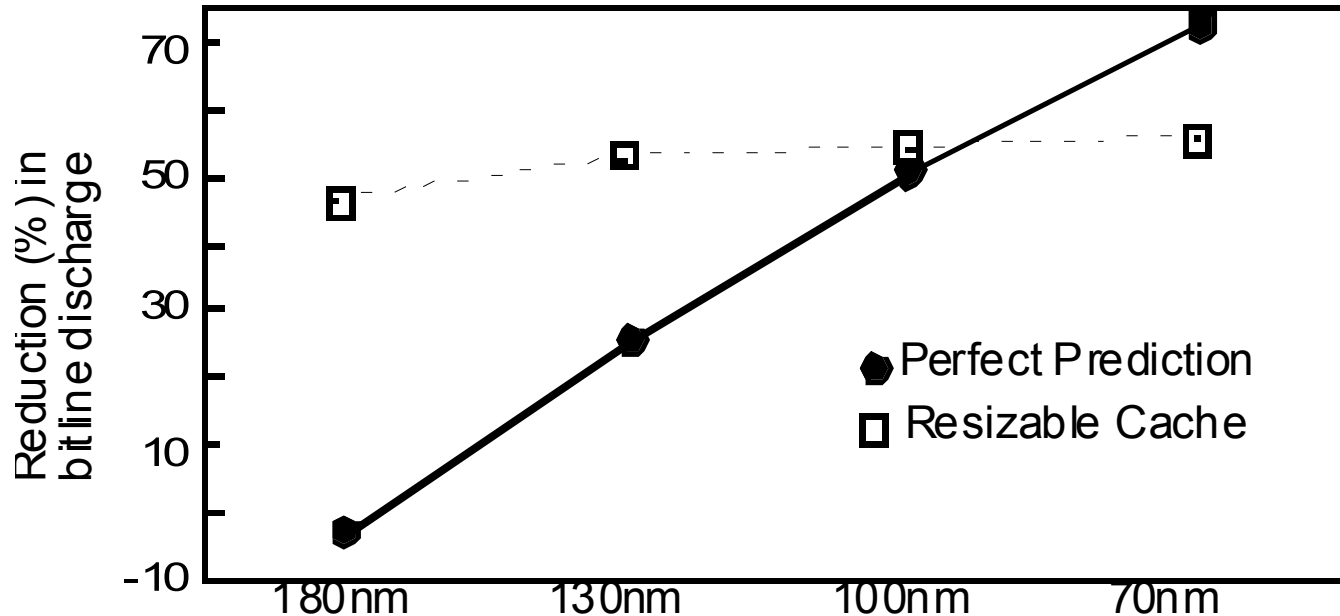
- Monitor/Adapt infrequently
- Energy/time overhead amortized in large interval
  - Important in the past, not in the future
- Possibly suboptimal control
  - Coarse-grain
  - Less sensitive

# Opportunity



- 74% opportunity for instruction caches
- 70nm technology

# Comparison: Resizable Caches



Resizable caches: consistent over technologies  
 Per-access control: capturing opportunity

# Conclusions

---

- Smaller energy overhead in the future  
Per-access fine control viable in the future
- On-demand wakeup is late  
Early precharging to avoid performance hit
- 74% opportunity for per-access control for 70nm  
Significantly less opportunity for the past  
Resizable caches good for the all generations

# For more information

---



PowerTap Project

<http://www.ece.cmu.edu/~powertap>

Computer Architecture Lab

Carnegie Mellon University