

Mosaic: Exploiting the Spatial Locality of Process Variation to Reduce Refresh Energy in On-Chip eDRAM Modules

Aditya Agrawal, Amin Ansari and Josep Torrellas

<http://iacoma.cs.uiuc.edu>



- eDRAM
- Periodic Refresh Requirement
- Refresh Reduction Techniques

MOTIVATION



eDRAM

- A 1T1C dynamic memory technology.
- The bit is stored as charge on the capacitor.
- Area and leakage energy savings.
- Increasing adoption in commercial processors: IBM POWER 7, POWER 8, Intel Haswell.
- Constraint: The charge on the capacitor has to be refreshed periodically.



Periodic Refresh Requirement

- Blocks normal accesses.
- Has temperature dependence (2x every 10 °C increase).
- Susceptible to device variations.
- Refresh rate in DRAM ~ once in 64 msec (at 85 °C).
- Refresh rate in eDRAM ~ once in 100 µsec (at 95 °C).
- Impacts energy and performance.



Refresh Reduction Techniques

- Access Patterns to Memory
 - Smart Refresh (MICRO 2007): DRAM
 - Refrint (HPCA 2013): eDRAM
- Variation in Retention Times
 - RAPID (HPCA 2006): DRAM
 - Hi-ECC (ISCA 2010): eDRAM
 - RAIDR (ISCA 2012): DRAM
 - Mosaic (HPCA 2014): eDRAM



Contribution

- Expose the **on chip spatial locality** in retention times.
 - A mathematical model accessible to architects.
- Exploit the spatial locality for **refresh reduction**.
 - A hardware only solution.
 - Low area overhead (2%).
 - Significant refresh reduction (20x).



- eDRAM Cell Retention Time
- Retention Time Distribution
- Bulk Distribution, Tail Distribution
- Main Idea

BACKGROUND

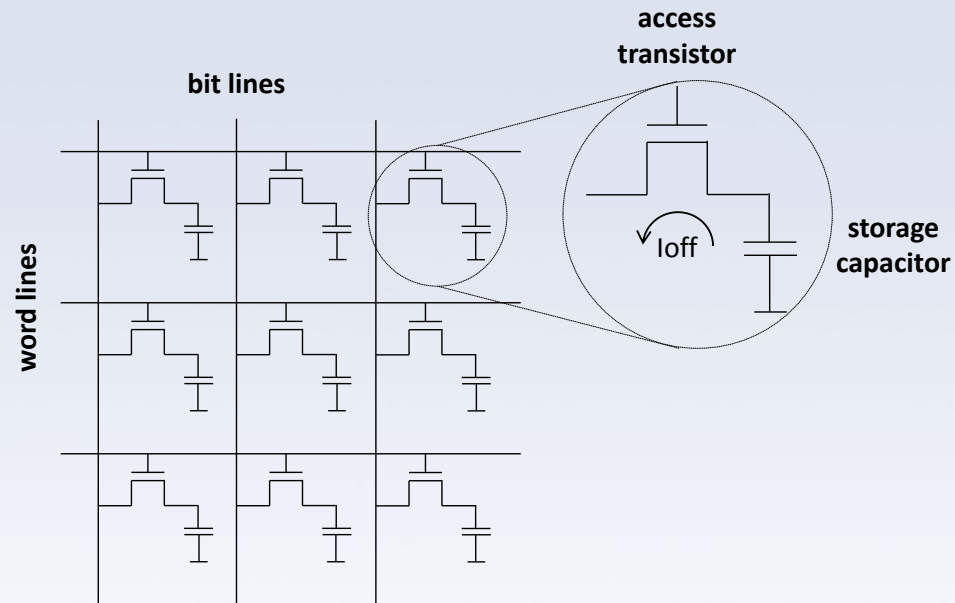


eDRAM Cell Retention Time

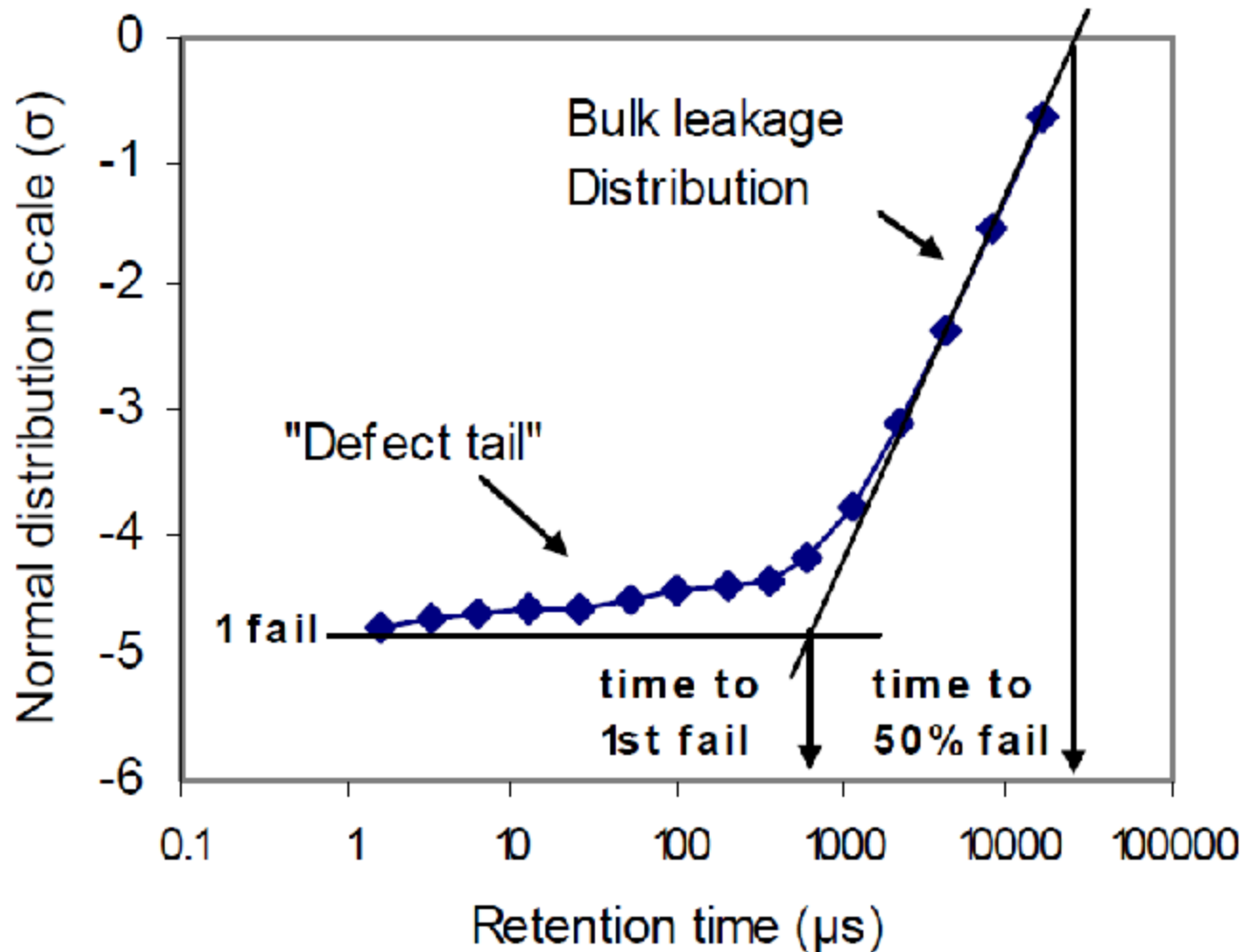
$$T_{\text{ret}} = A * 10^{(V_t * B)} \text{ sec}$$

Using published data from IBM at 65 nm, $T_{\text{ret}} \sim 25 \text{ msec}$.

However, in practice eDRAMs are refreshed at $\sim 50\text{-}100 \text{ usec}$.

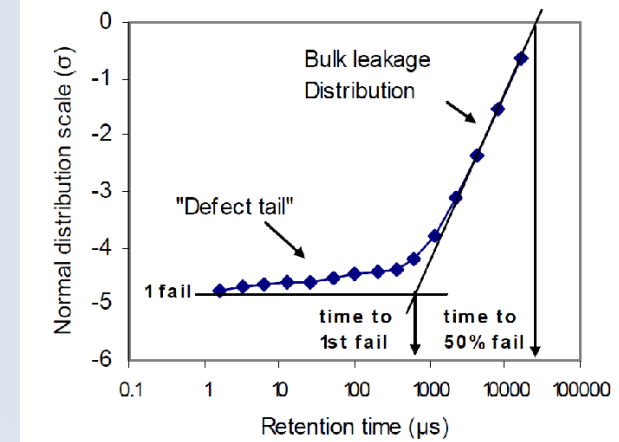


Retention Time Distribution



Bulk Distribution

- Area under the curve from $(-4 \sigma, \infty)$.
 - 99.9968% of the cells.
- Follows a log-normal distribution.
- Caused by process variation in V_t of the access transistor.
 - Includes systematic and random components.



We also know,

- V_t variation has a normal distribution.
- $\log_{10}(T_{ret}) = V_t/B + \log_{10}(A)$

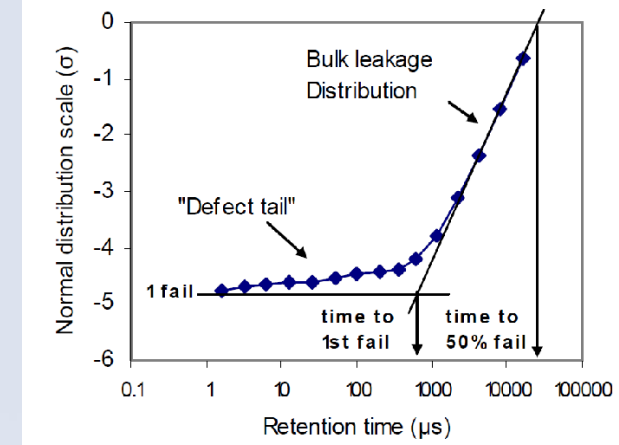
Therefore,

- Normal distribution in $V_t \rightarrow$ Log normal distribution in T_{ret} .



Tail Distribution

- Area under the curve from $(-\infty, -4 \sigma)$.
 - 0.0031% of the cells (31 ppm).
- Follows a log normal distribution.
- Caused by random manufacturing defects.
- Only a small fraction (3 ppm) is considered defective.



Main Idea

- T_{ret} is a function of V_t .
- V_t variation has spatial locality (systematic component).

Therefore,

- T_{ret} will have spatial locality.
- Exploiting this spatial locality can reduce refresh energy at low area and energy overheads.



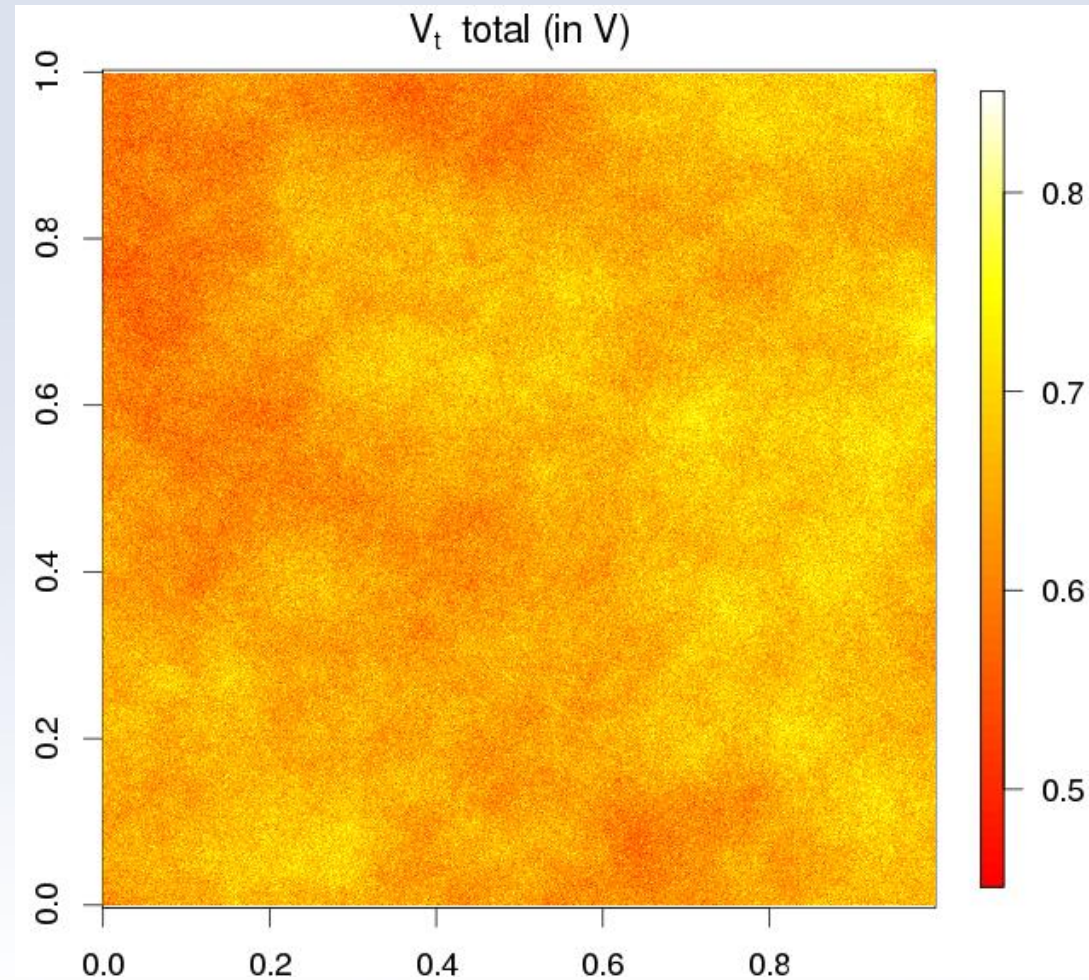
- Spatial Map of Retention Times
- Opportunity & Tradeoffs

EXPLOITING SPATIAL LOCALITY



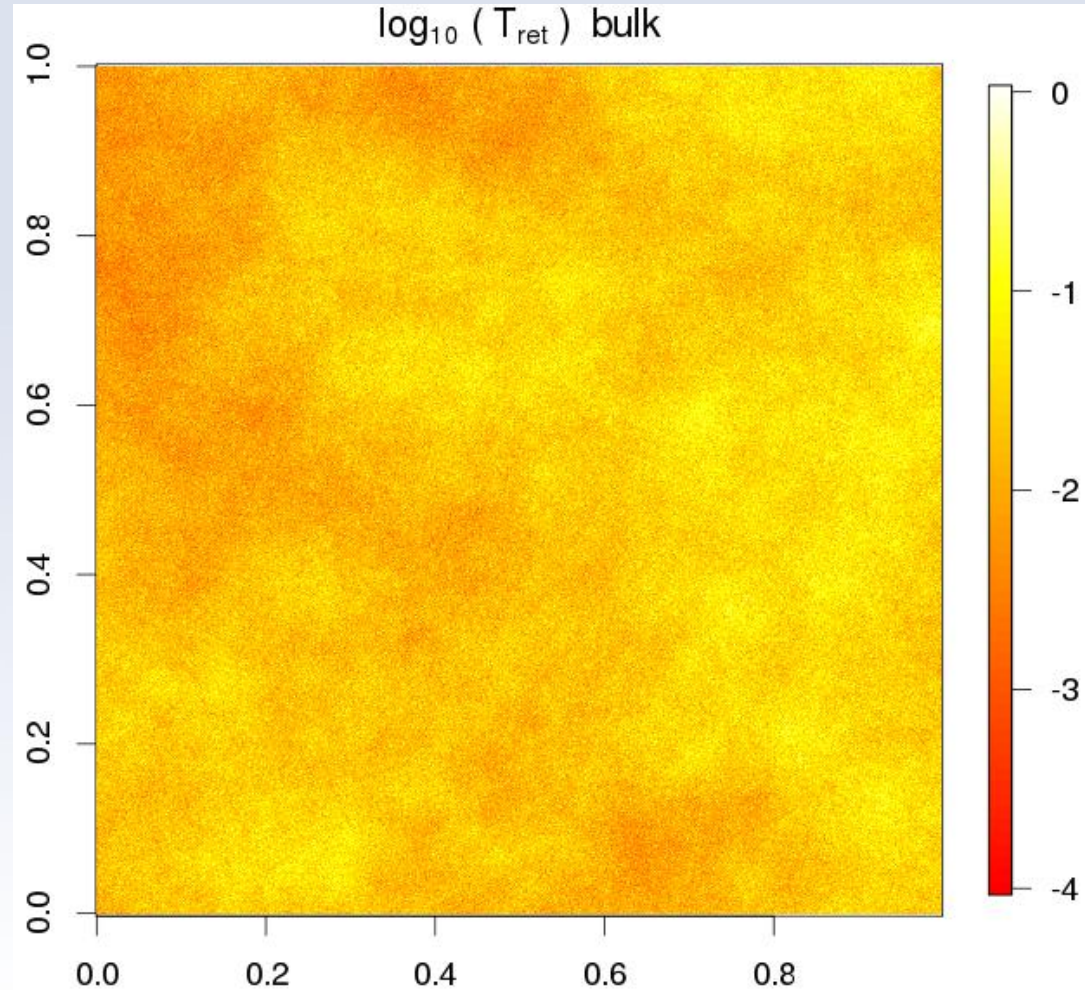
Step 1

- Obtain a spatial map of V_t using VARIUS.
- Includes the systematic and random components of V_t variation.



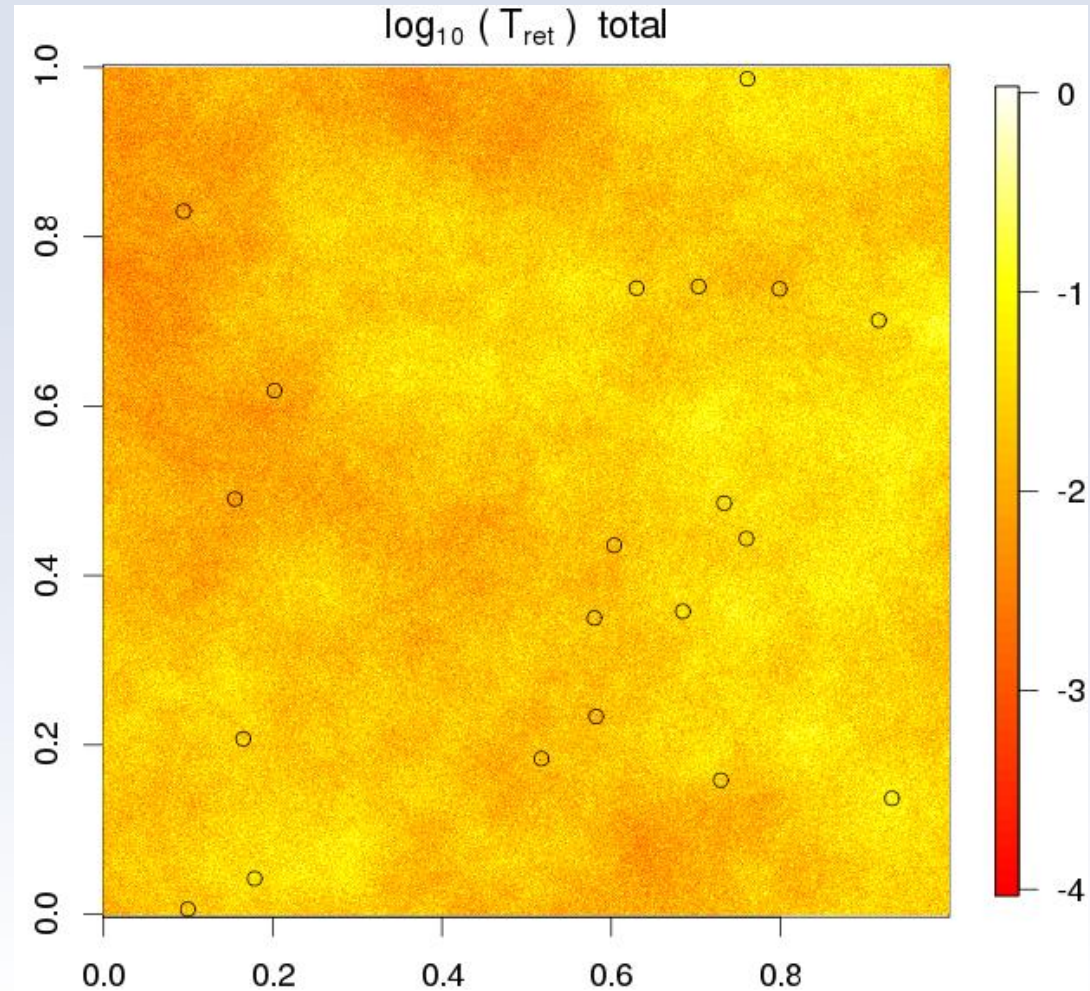
Step 2

- Cell by cell translation from V_t values to T_{ret} for the bulk distribution.
- Spatial map remains the same, the scale changes from linear to \log_{10} .



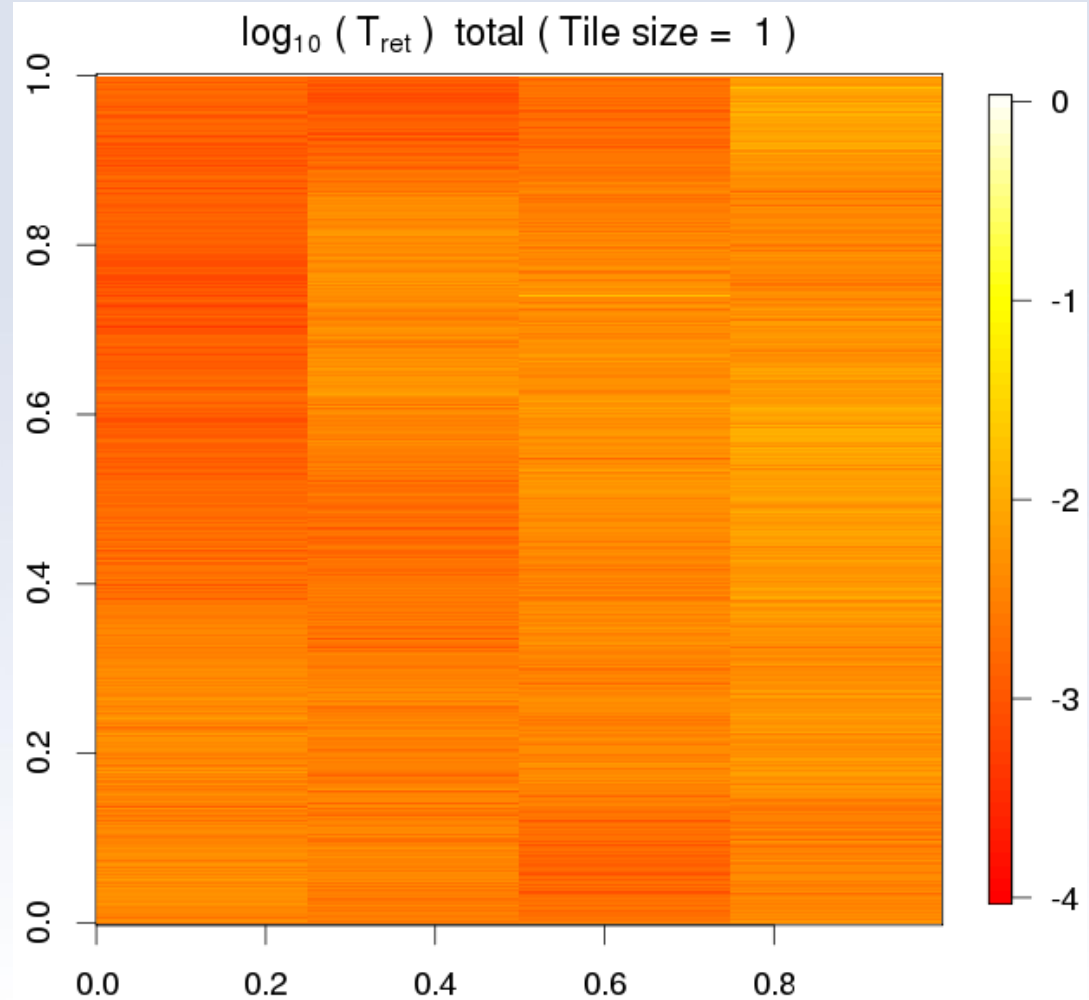
Step 3

- From IBM data: 20 ppm cells follow the tail distribution.
- Superimposing the tail distribution on the bulk distribution gives the total per-cell T_{ret} distribution.



Step 4

- Memory is accessed at a line granularity.
- We obtain a per-line T_{ret} distribution by taking the minima of the cells in the line.

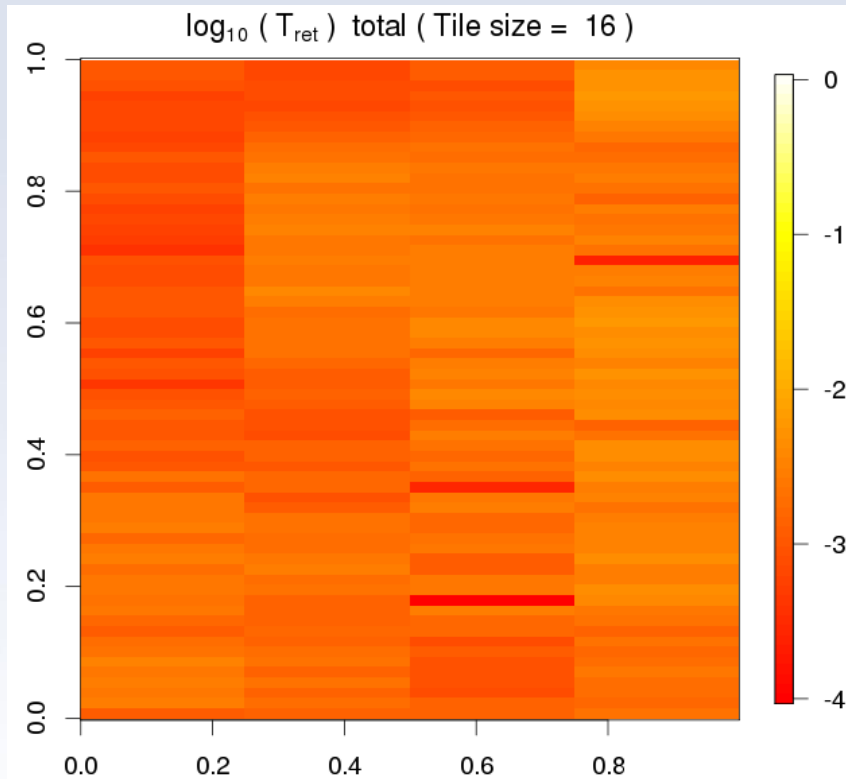


Opportunity

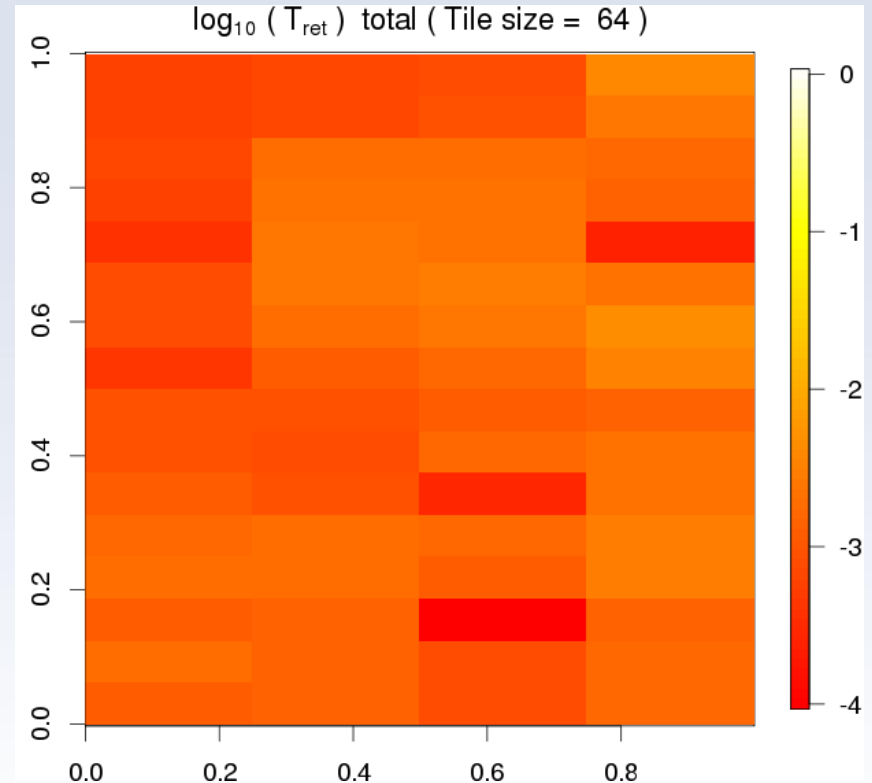
- Lower bound on the number of refreshes
 - Profile, track and refresh each line at its own rate.
 - Huge area and energy overheads.
- A better solution (**Mosaic**): Exploit spatial locality of T_{ret}
 - Logically group co-located lines into tiles.
 - Profile each tile and save the information (in a SRAM).
 - Track (using counters) and refresh each tile at its own rate.
 - Potentially with small area and energy overheads.



Mosaic of Tiles



Mosaic with Tile Size = 16



Mosaic with Tile Size = 64



Tradeoffs

Refresh energy savings - counter size - tile size.

- Small tiles => high refresh savings, high area overheads.
- Small counters => low refresh savings, low area overheads.

Next,

- A simple HW solution to track and refresh each tile.
- Best combination of tile size and counter size (Mosaic).
- Compare Mosaic against baseline and lower bound.



- Mosaic Hardware
- Mosaic Operation

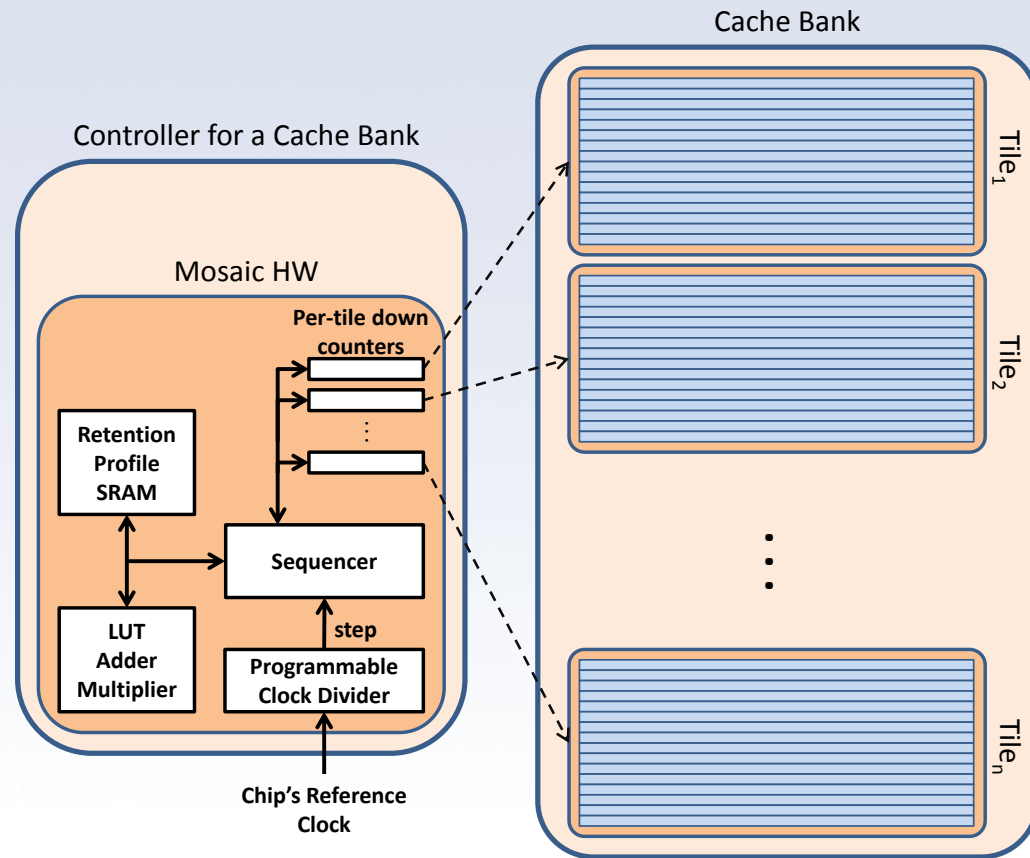
ARCHITECTURE



Mosaic Refresh Hardware

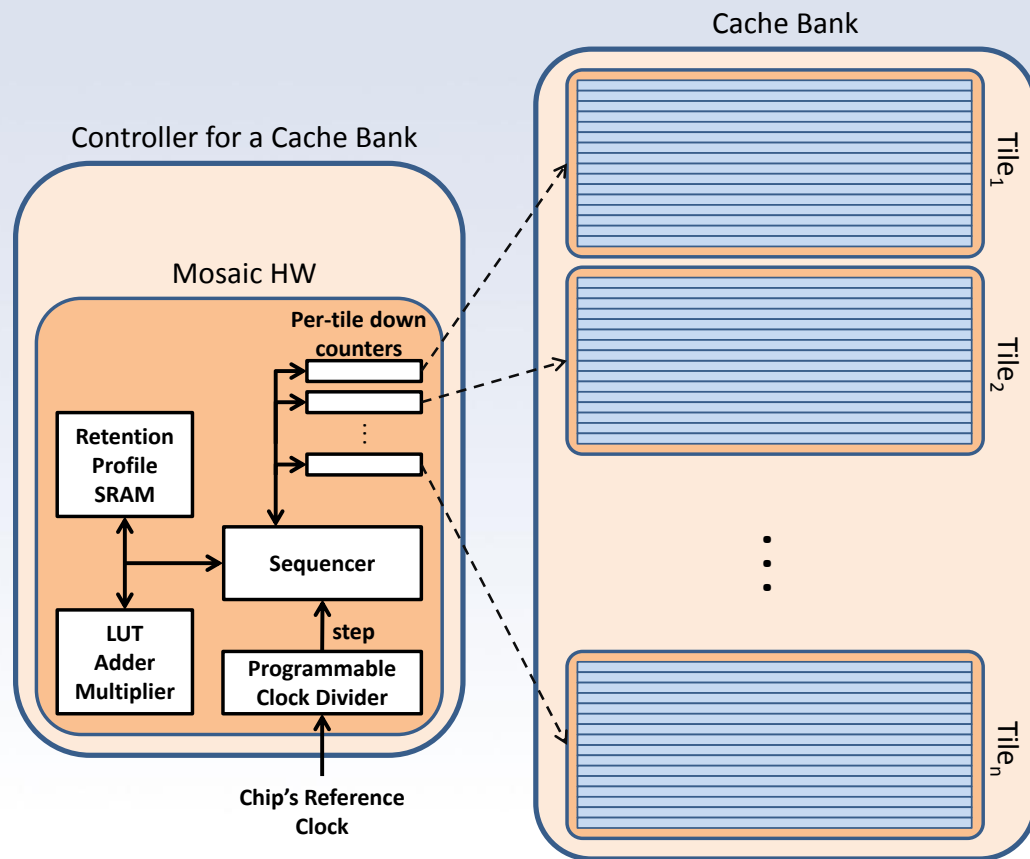
Augment the cache controller

- SRAM with a profile of tile retention times.
- Logic to track and trigger per tile refresh.



Mosaic Operation

```
At every step (50  $\mu$ sec)
for (all tiles in the cache) {
  Decrement counter
  if (count == 0) {
    Schedule tile refresh
    Read retention profile SRAM
    Reload counter
  }
}
```



- Architectural Parameters
- Tools & Applications
- Design Comparison

EVALUATION SETUP



Evaluation Setup

| Architectural parameters | |
|--------------------------|---------------------------|
| Chip | CMP with 16 2-issue cores |
| IL1/DL1 | 32 KB, private |
| L2 | 256 KB, private |
| L3 (eDRAM) | 16 MB, 16 banks, shared |
| L3 bank | 1 MB |
| Network | 4 x 4 torus |
| Coherence | MESI directory at L3 |



Evaluation Setup

| Tools & Applications | |
|-------------------------|------------------|
| Architectural Simulator | SESC |
| Timing & Power | McPAT & CACTI |
| Synthesis | Design Compiler |
| Statistics | R |
| Variation | VARIUS |
| Applications | SPLASH-2, PARSEC |



Design Comparison

- Baseline:
 - All lines refreshed at 50 μ sec.
- RAIDR:
 - Applied to eDRAMs.
 - Lines refreshed at 50, 100 or 200 μ sec.
- Mosaic:
 - Tile size of 32 lines, 6 bit counter per tile.
 - L3 area overhead of 2%.
- Ideal (lower bound):
 - Tile size of 1.

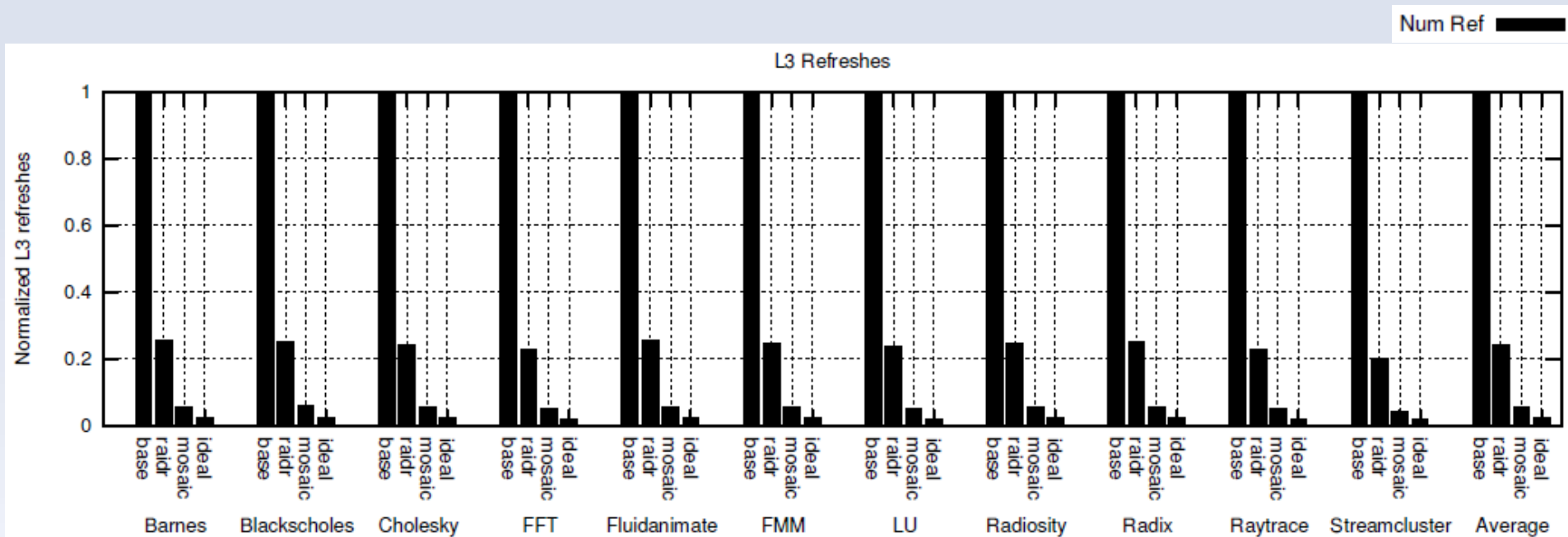


- Refresh Count
- Execution Time
- L3 Energy

EVALUATION



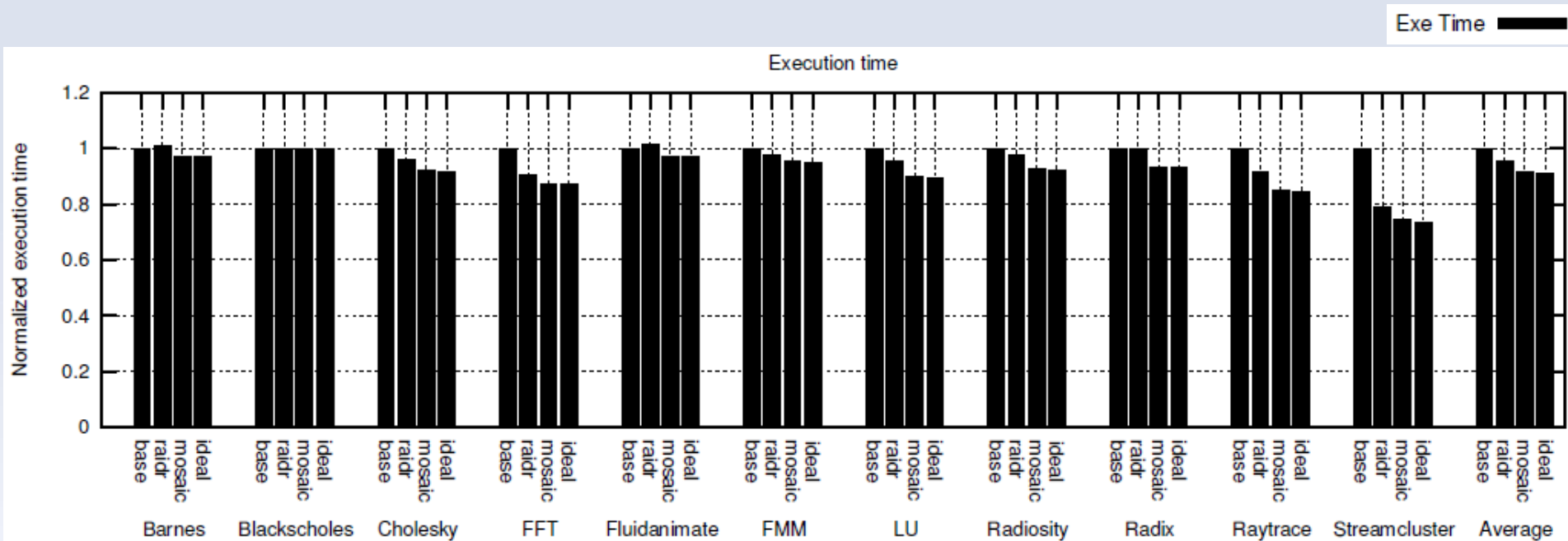
Refresh Count



- RAIDR reduces the number of L3 refreshes by 4x.
- Mosaic reduces the number of L3 refreshes by 20x.
- Mosaic is within 2.5x of the lower bound (ideal).



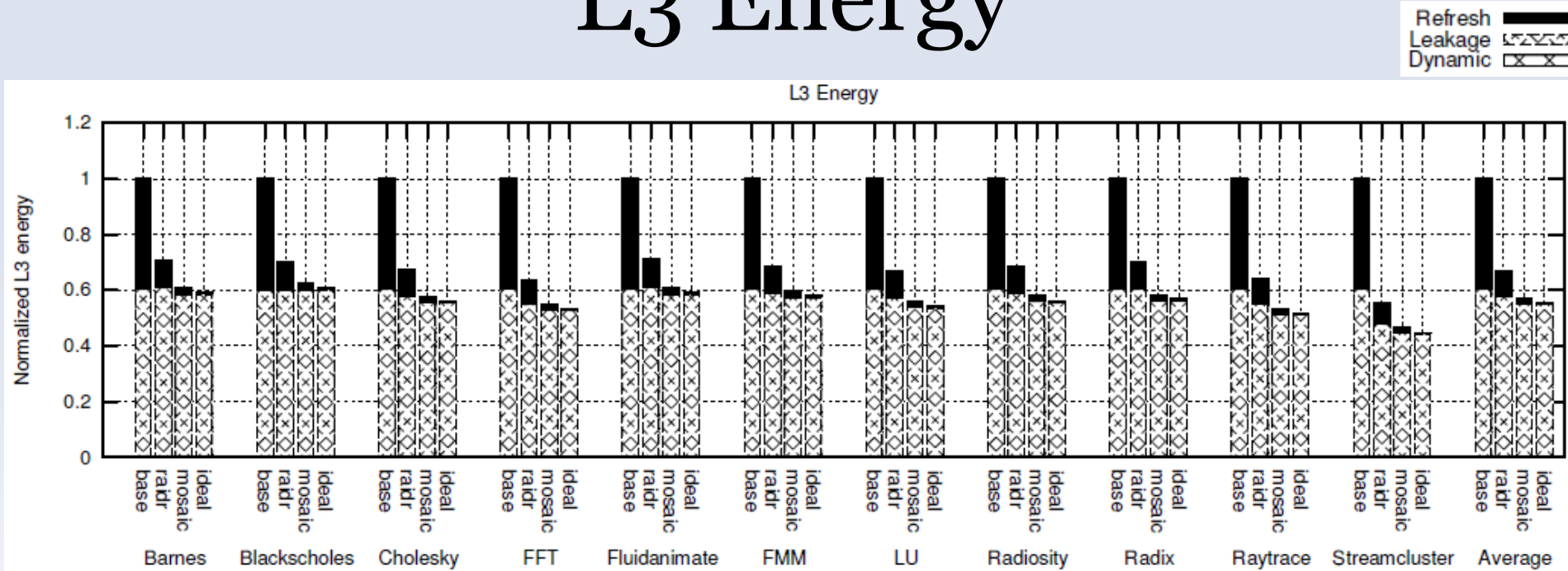
Execution Time



- Performance improves because of reduced cache blocking.
- Mosaic reduces execution time by 9%.
- Ideal reduces execution time by 10%.



L3 Energy



- L3 energy reduction comes from savings in refresh energy and leakage energy.
- Mosaic saves 43% of L3 energy.



Conclusion

- Exposed the **on chip spatial locality** of retention times.
 - A mathematical model accessible to architects.
- Exploited the spatial locality for **refresh reduction**.
 - A hardware only solution.
 - Low L3 area overhead (2%).
 - Significant refresh reduction (20x).
 - Saves 43% energy in L3.



Mosaic: Exploiting the Spatial Locality of Process Variation to Reduce Refresh Energy in On-Chip eDRAM Modules

Aditya Agrawal, Amin Ansari and Josep Torrellas

<http://iacoma.cs.uiuc.edu>

