Charge-Aware DRAM Refresh Reduction With Value Transformation

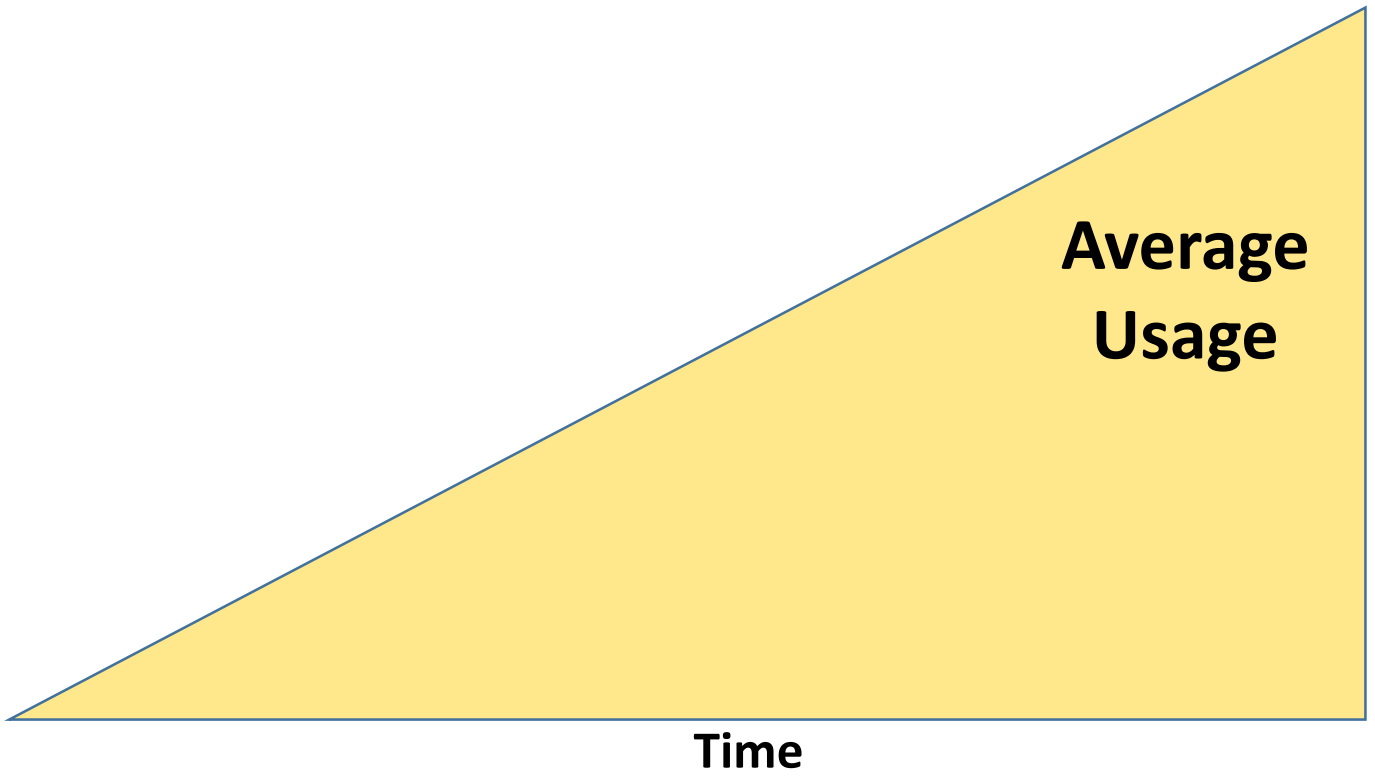
Seikwon Kim, Wonsang Kwak, Changdae Kim,

Daehyeon Baek, and Jaehyuk Huh

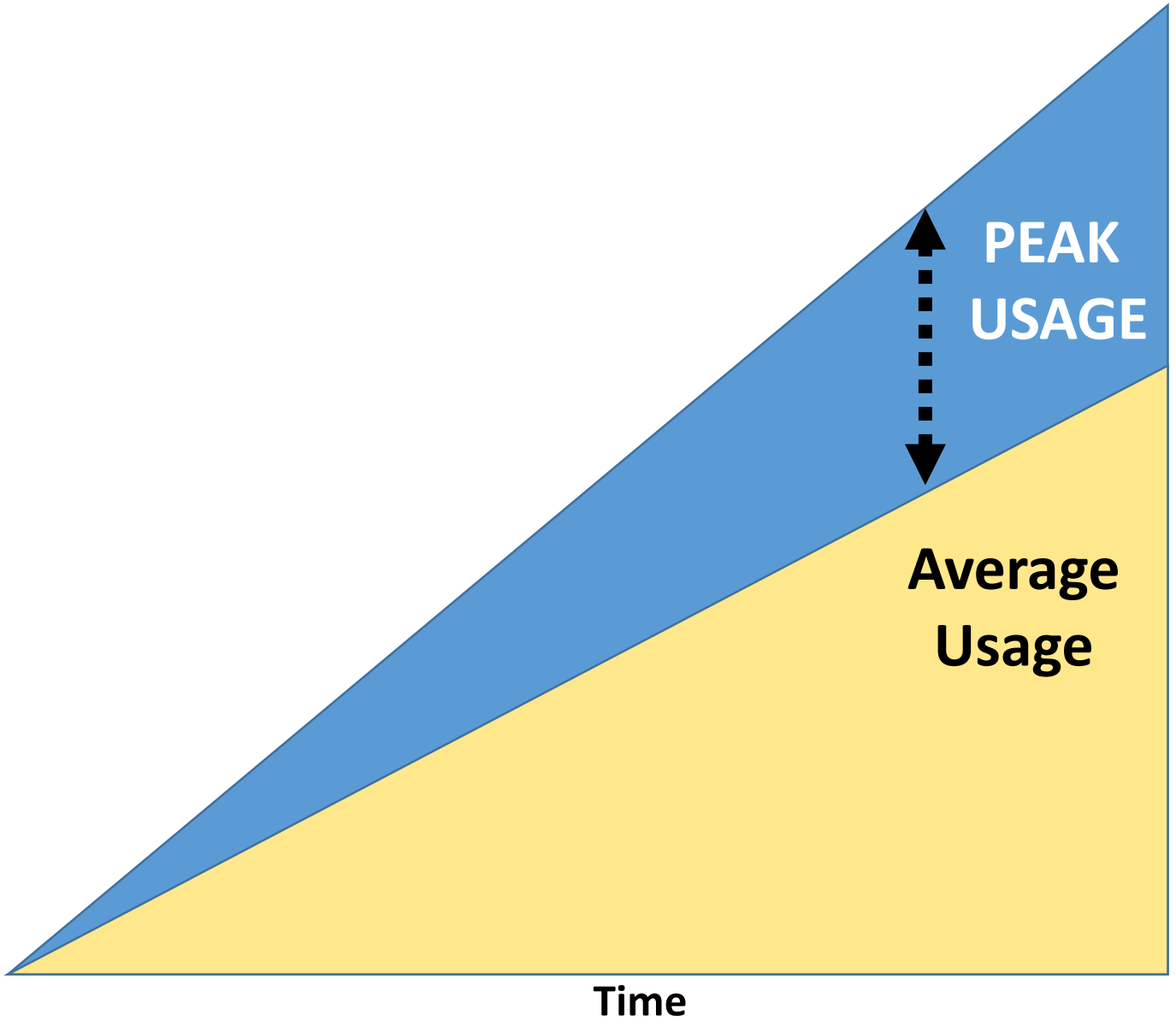
Feb. 26. 2020

KAIST SAMSUNG ETRI

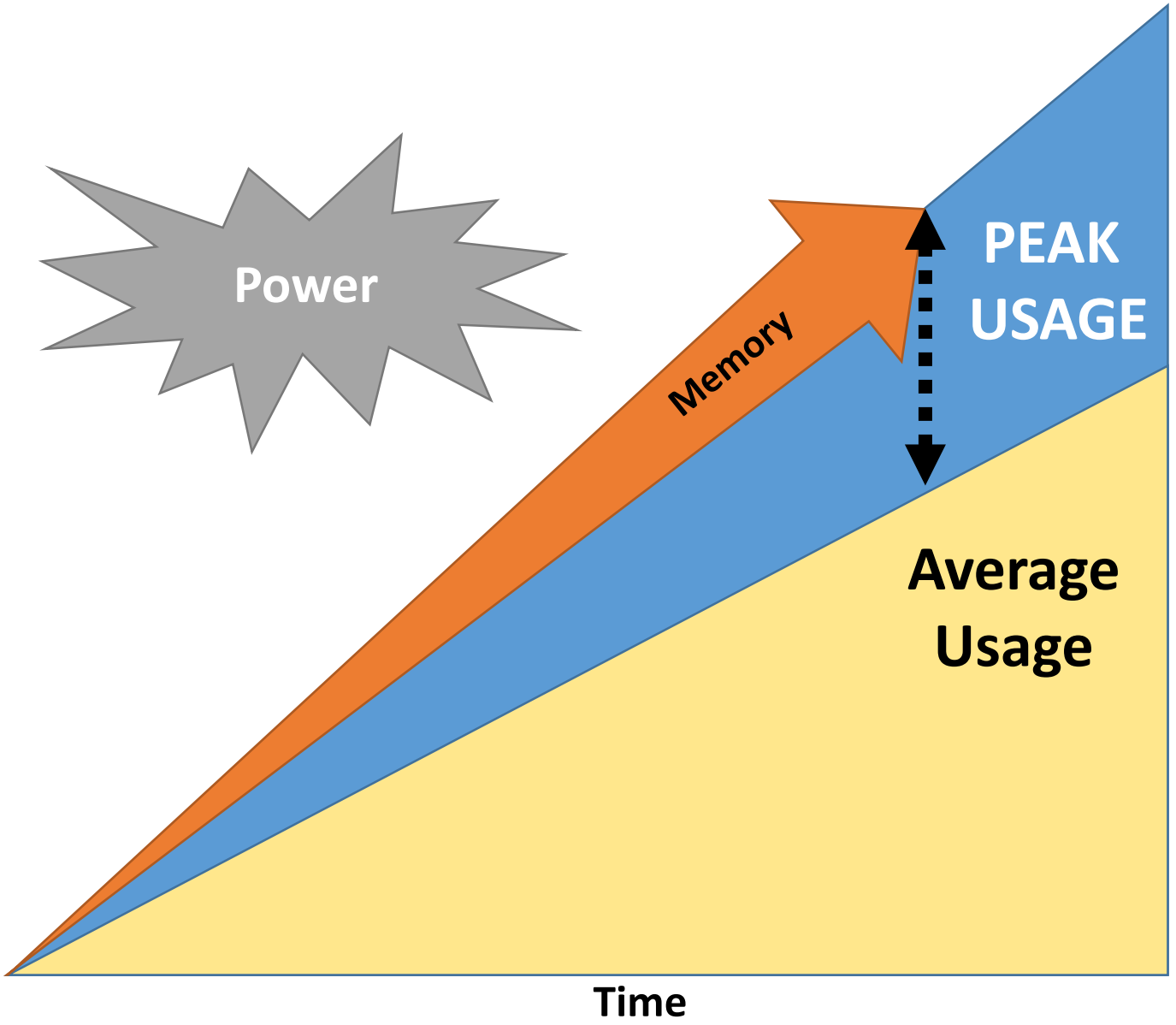
Memory Overprovisioning



Memory Overprovisioning

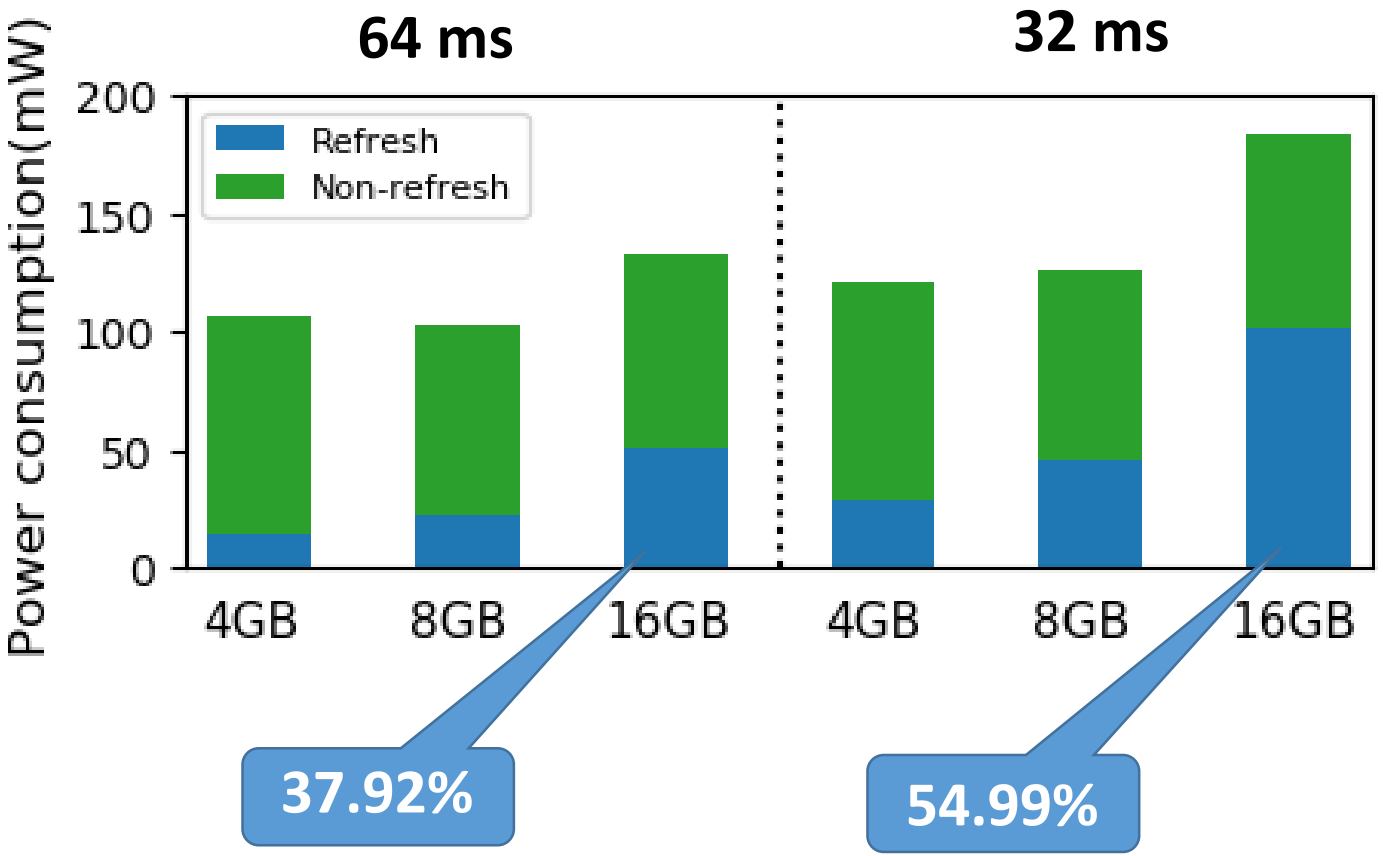


Memory Overprovisioning



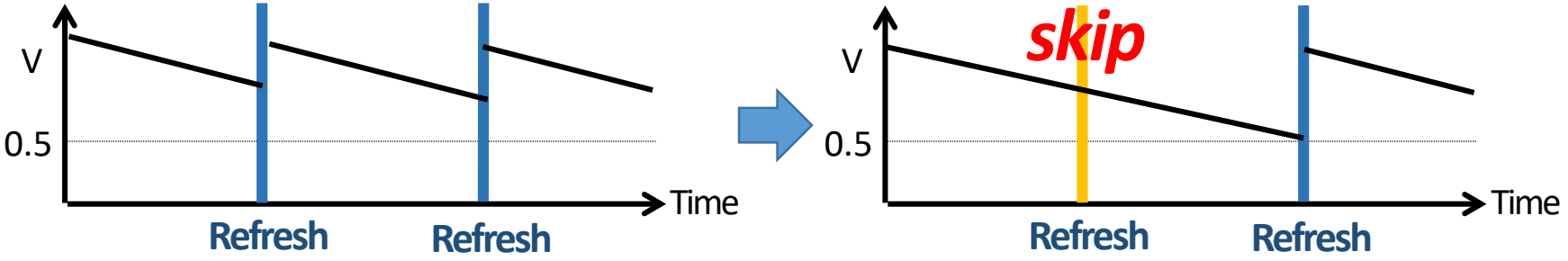
Memory Power Consumption

- Refresh power consumption
  - 32 RET w/ 16GB: 54.99% of total power



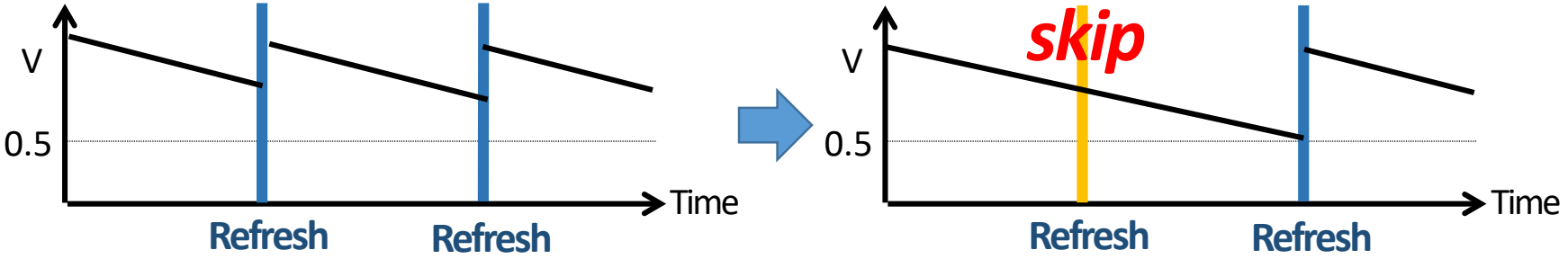
Prior Work

- Retention Time Aware: on long retention time cells

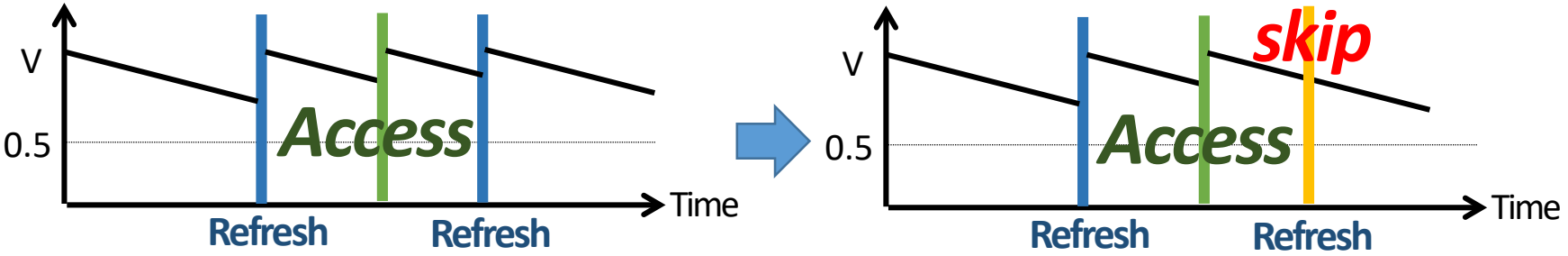


Prior Work

- Retention Time Aware: on long retention time cells

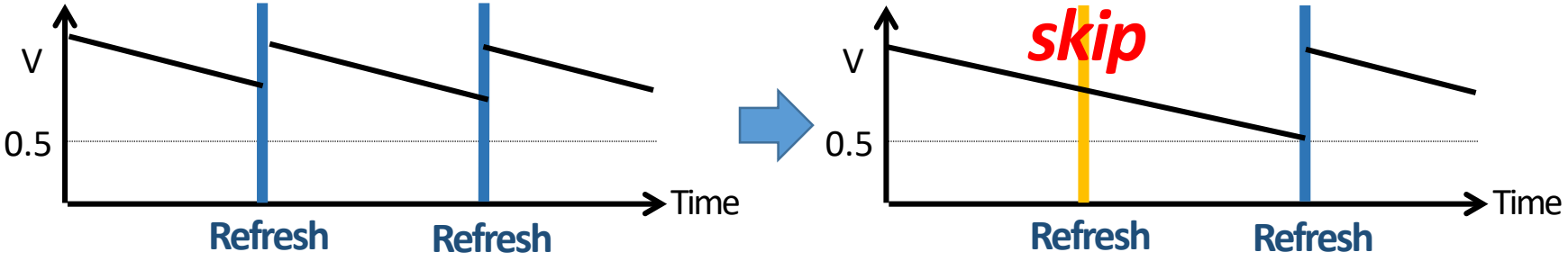


- Access Aware: on recently accessed cells

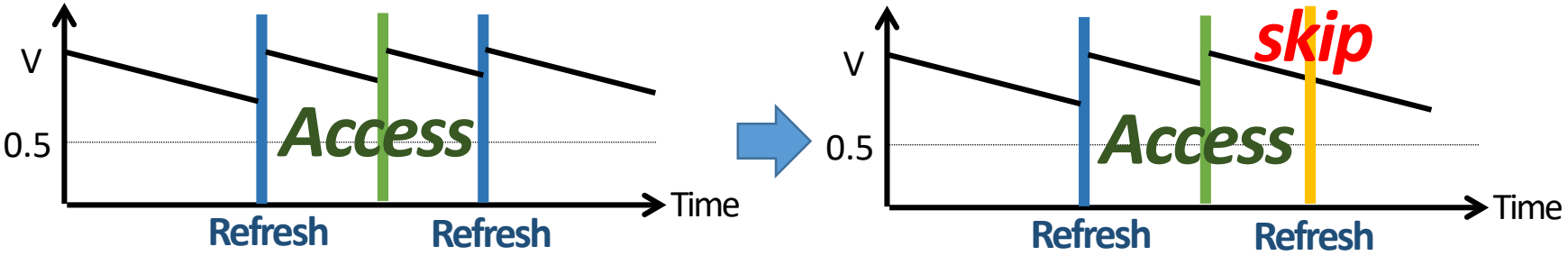


Prior Work

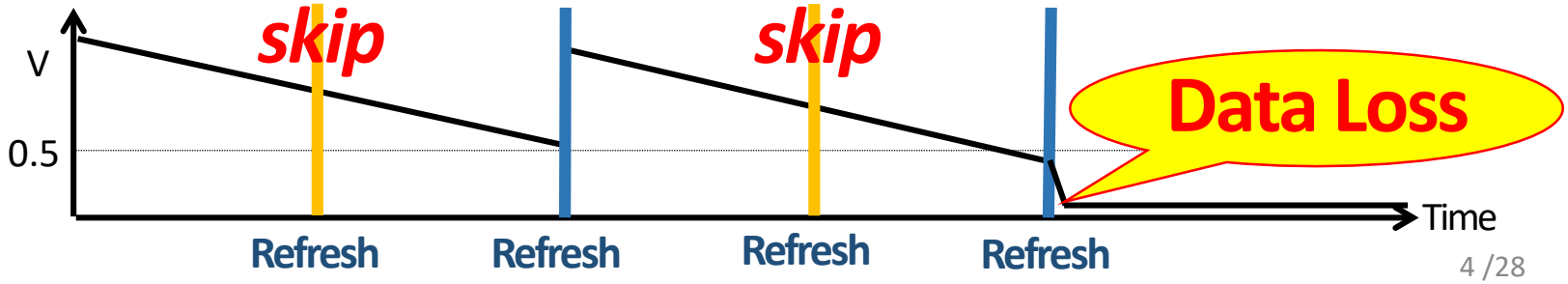
- Retention Time Aware: on long retention time cells



- Access Aware: on recently accessed cells

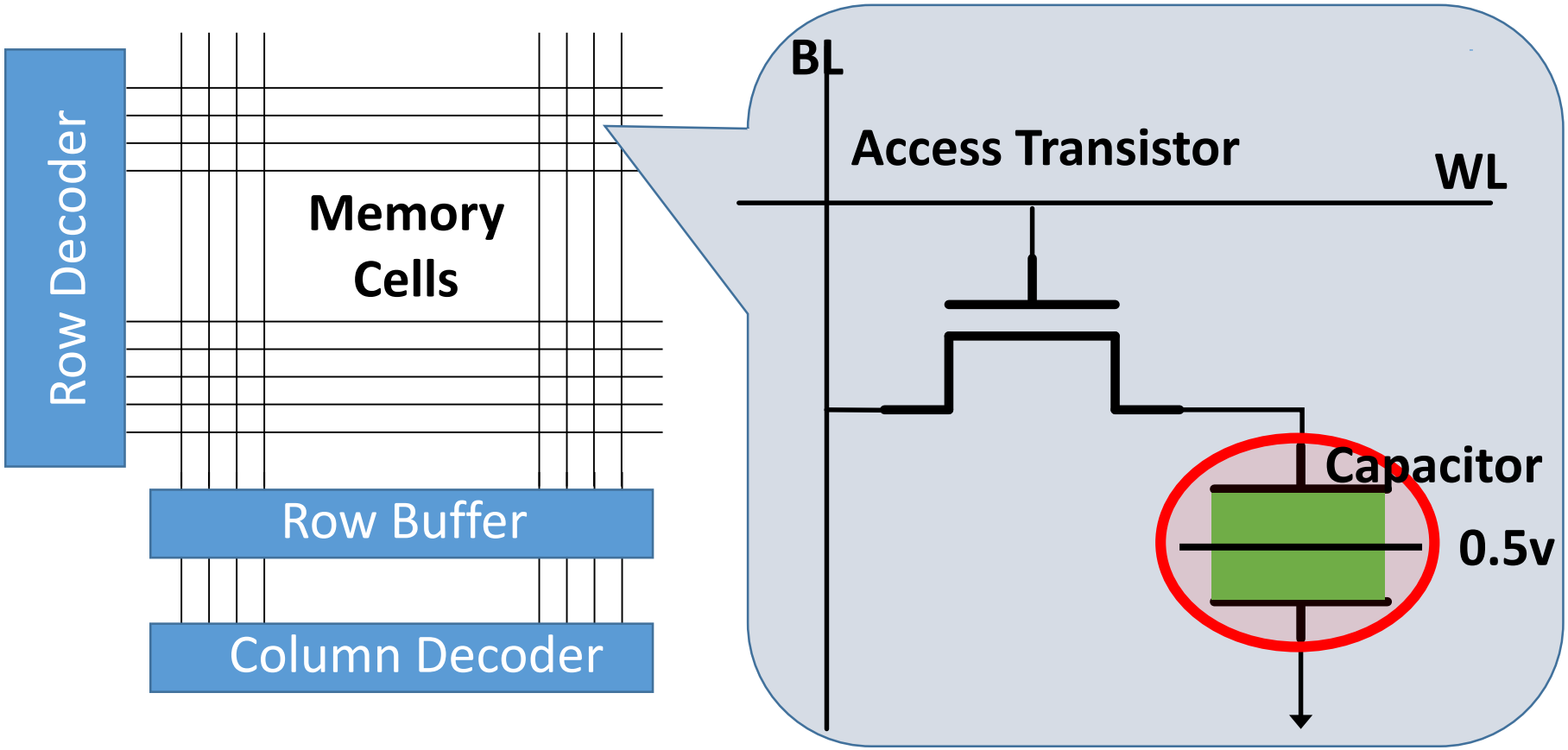


- Invalid-Data / Error-Tolerable-Data Aware



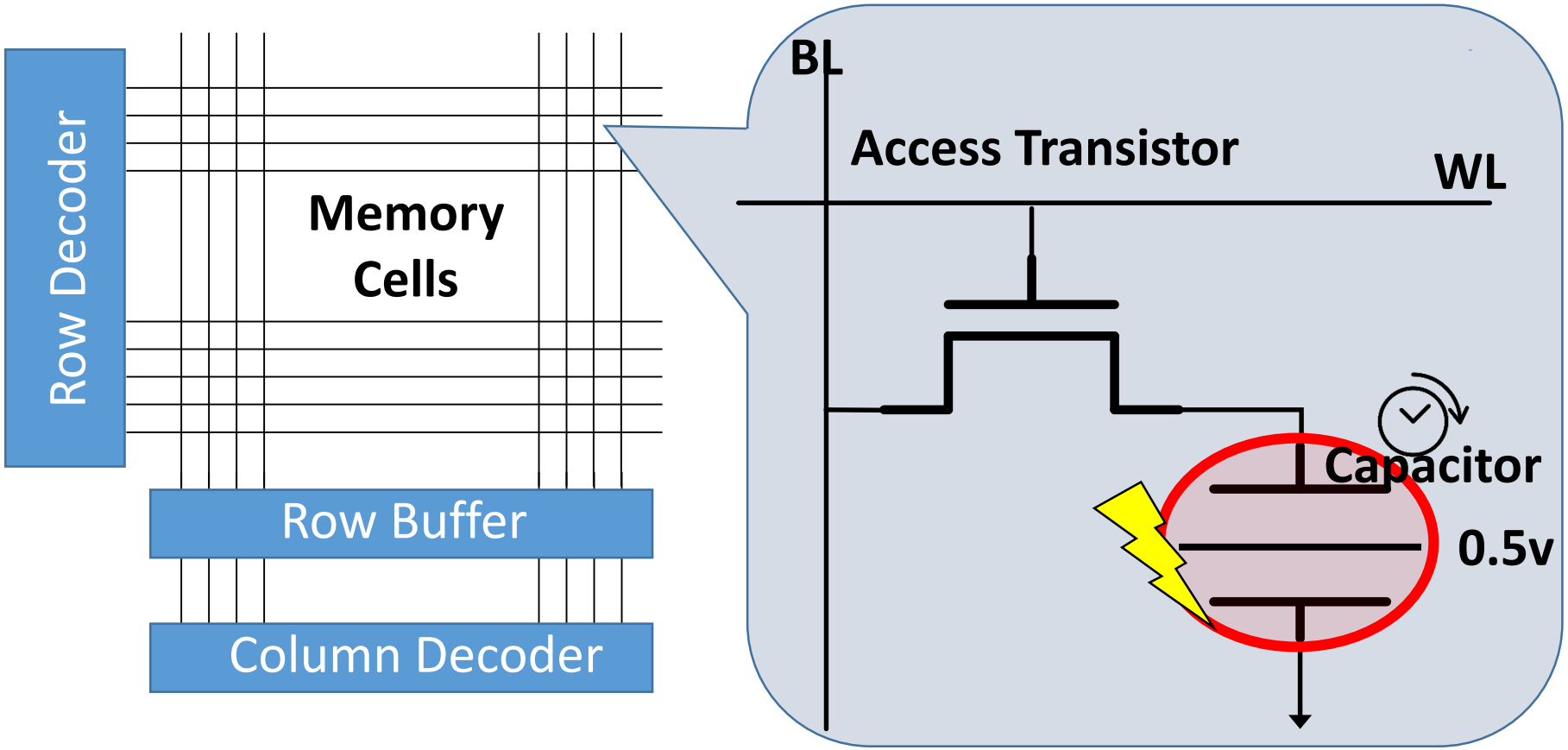
Electron Always Leaks

- Charge state if charge is higher than 0.5v
- Discharge state if charge is lower than 0.5v



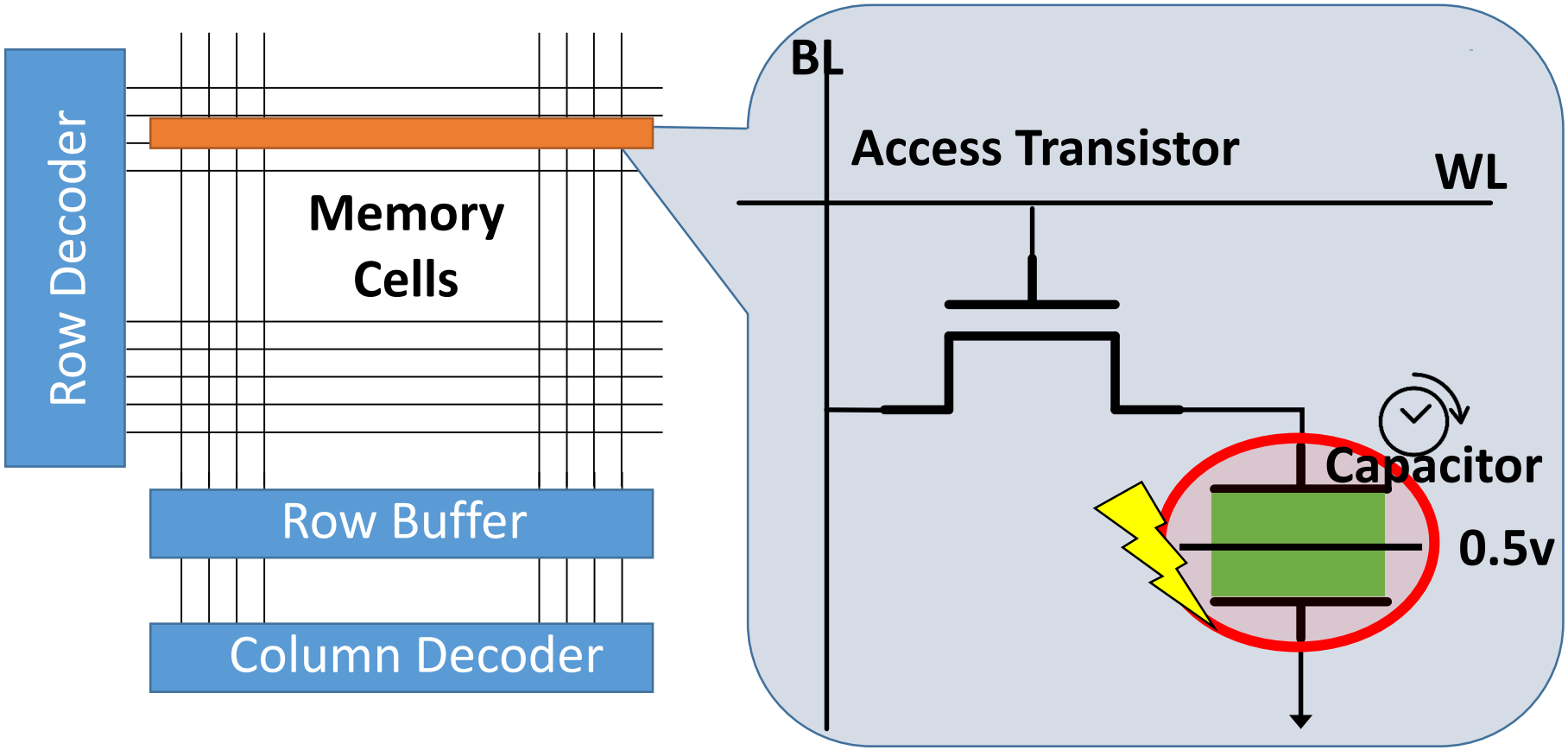
Electron Always Leaks

- Charge state if charge is higher than 0.5v
- Discharge state if charge is lower than 0.5v

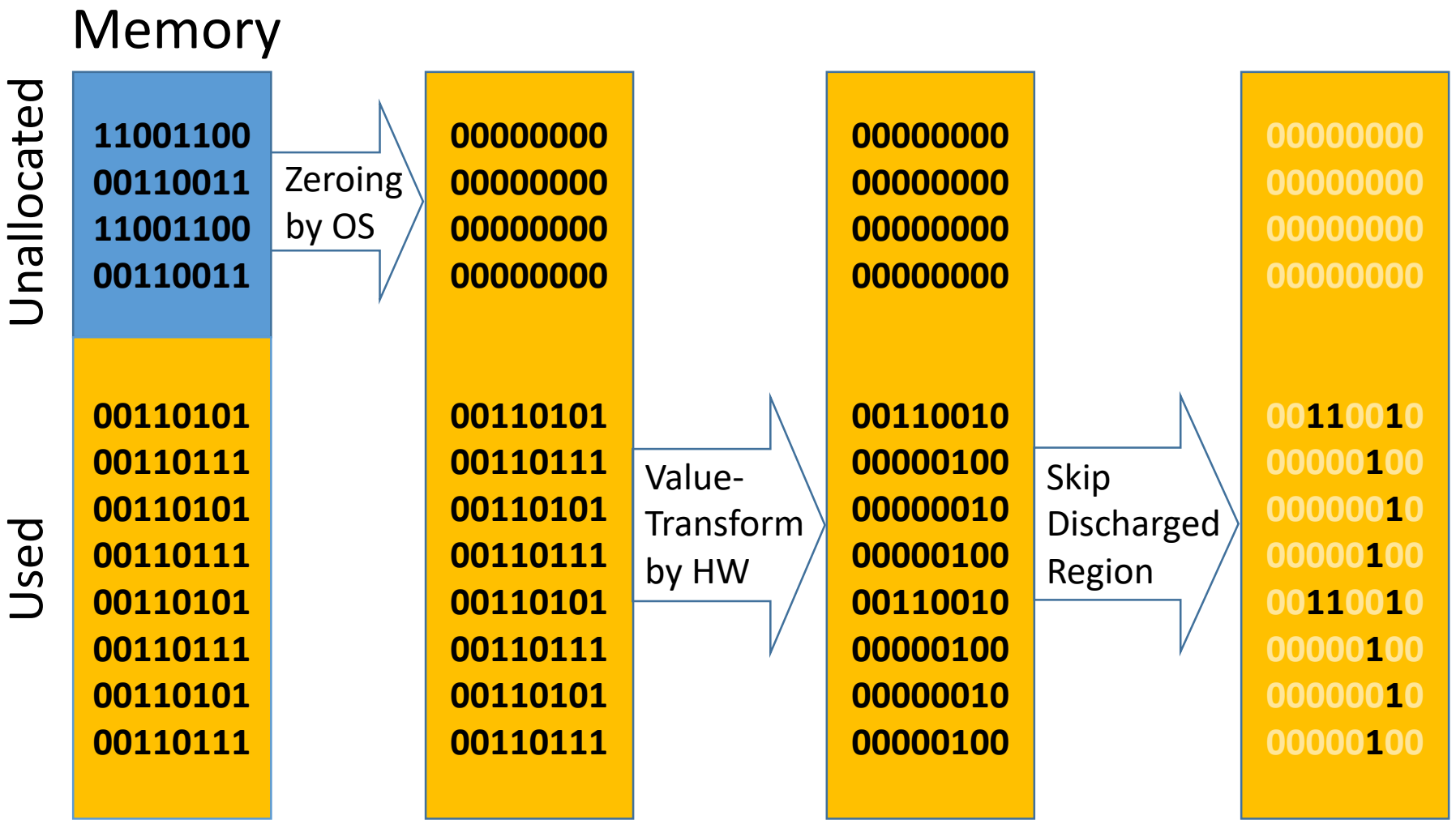


Electron Always Leaks

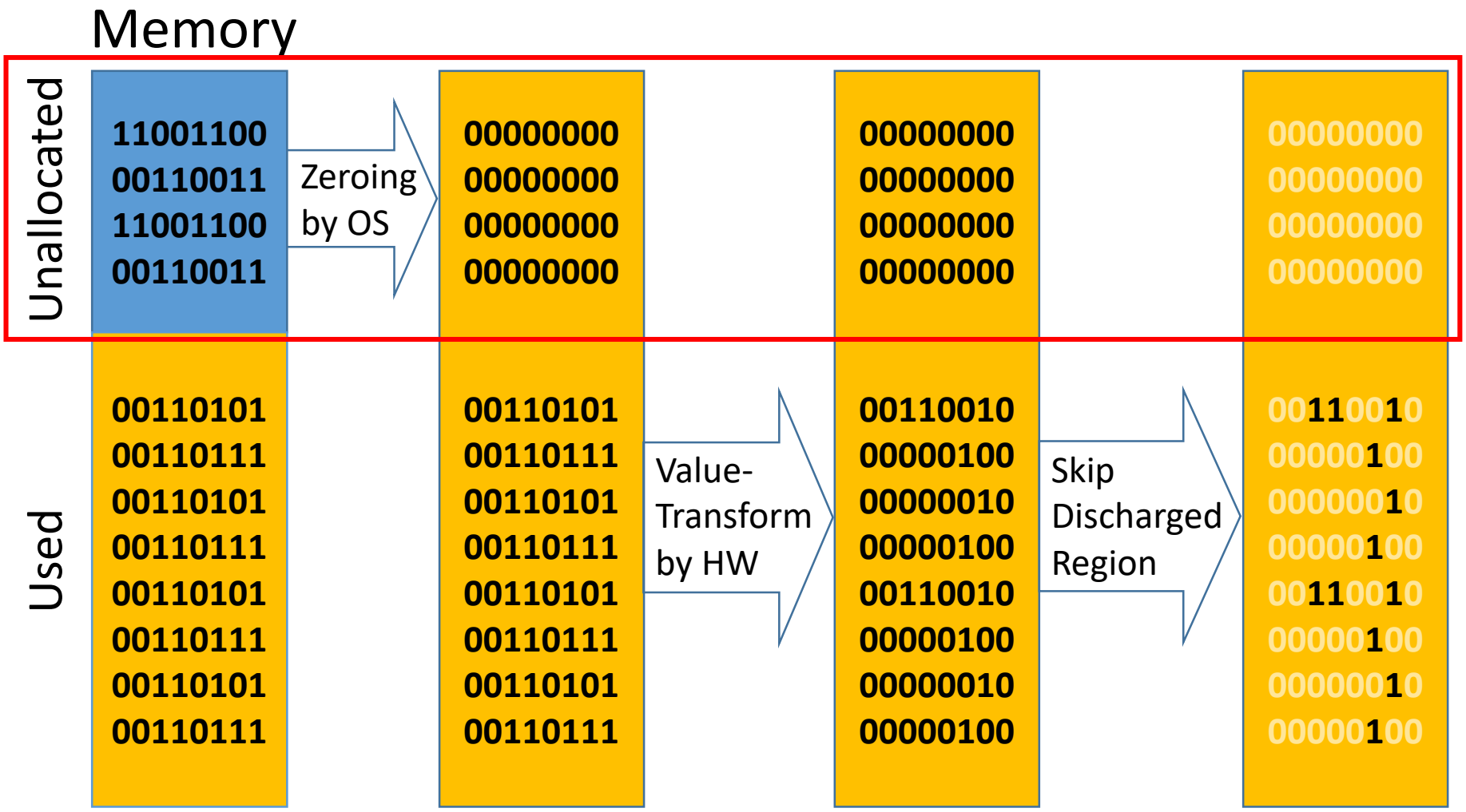
- Charge state if charge is higher than 0.5v
- Discharge state if charge is lower than 0.5v



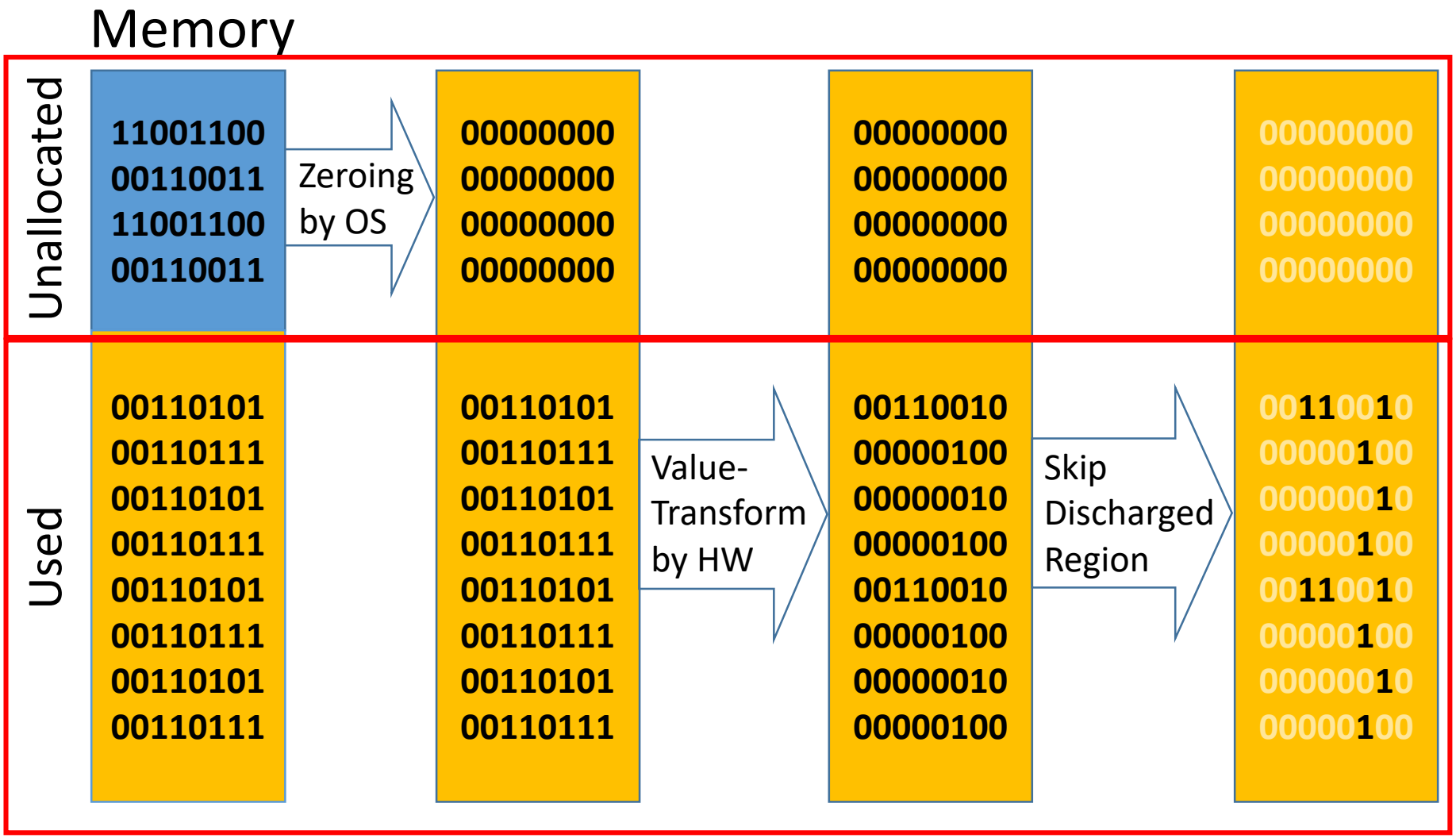
Our Idea: Charge Aware Refresh



Our Idea: Charge Aware Refresh



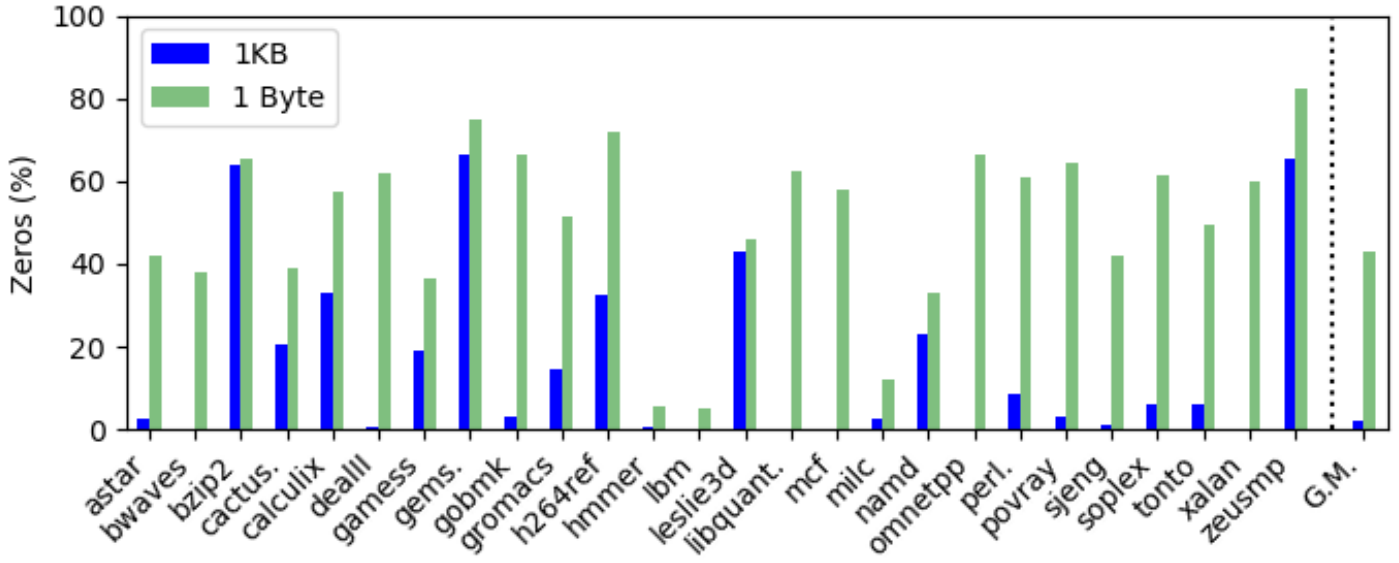
Our Idea: Charge Aware Refresh



Potential of Skipping Discharged State

- Allocated region from cloud traces: 62%
  - Alibaba trace: 88%
  - Google trace: 70%
  - Bitbrain trace: 28%

- Zero values in SPECCPU2006
  - 1B-gran: 43% / 1KB-gran: 2.3%

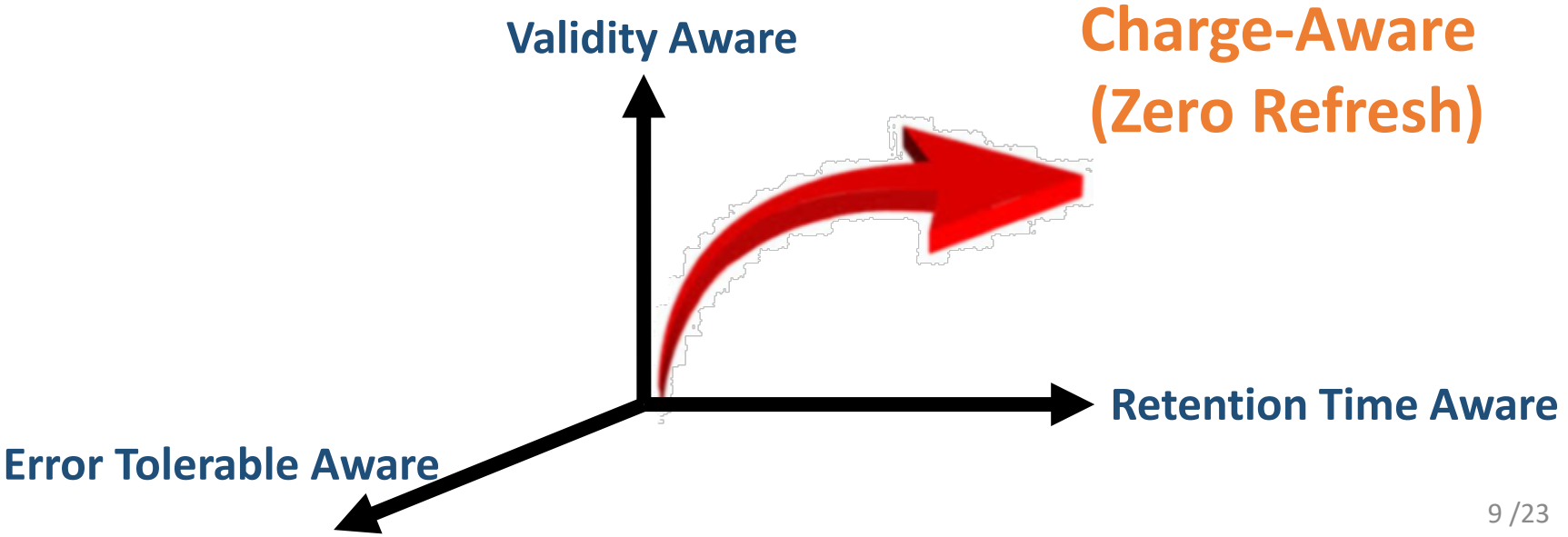


Challenges

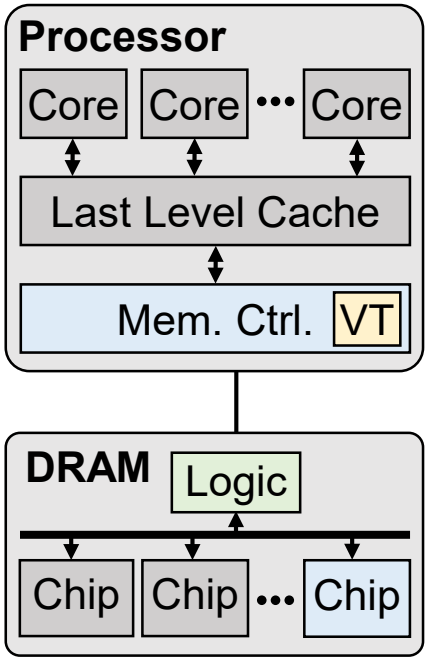
- Frequency of discharged cells
  - Generate as many zeros as possible
    → Early zeroing by OS (also for security, allocation performance)
    → Value transformation using compression technique
- Contiguity of discharged cells
  - Zero value clustering at a unit of refresh operation
    → Data rotation
    → Refresh counter modification

Contribution of Zero-Refresh

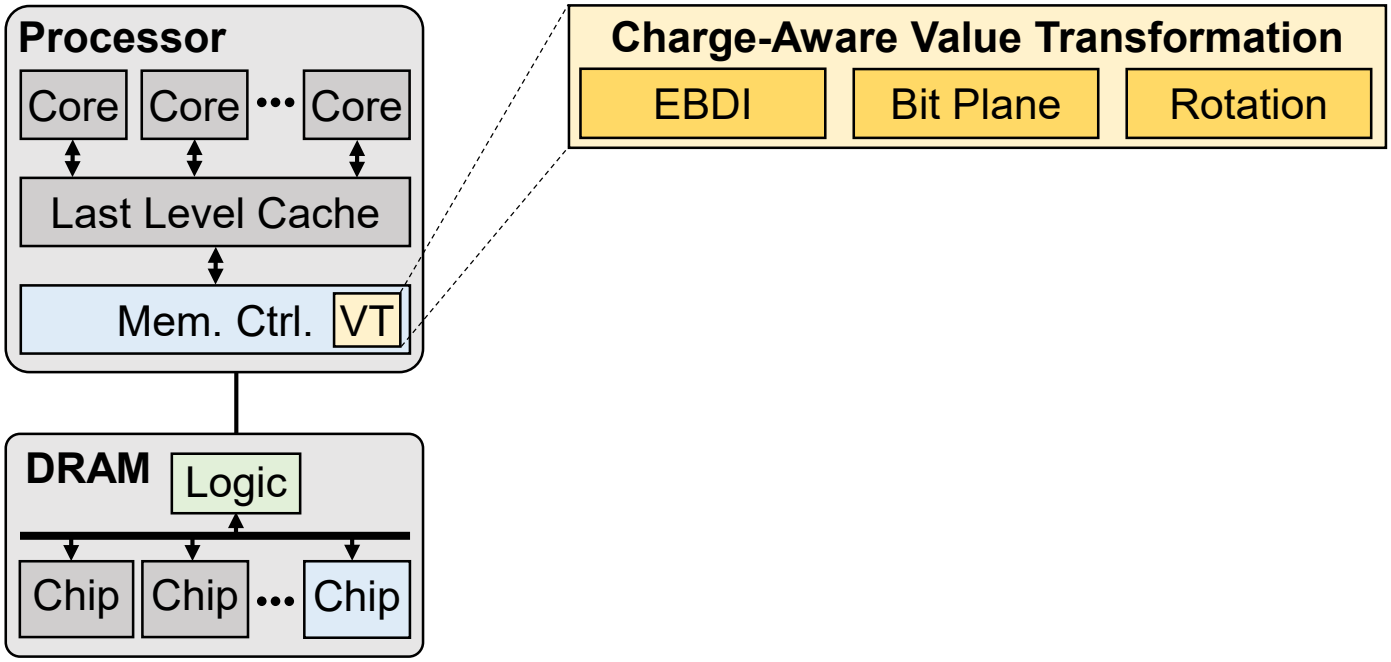
- Explore a new axis for the refresh mechanism
  - **Do not refresh discharged cells**
- Value transformation based refresh reduction
  - Generate consecutive zeros
  - Skip refresh for zero values
  - Keep data integrity



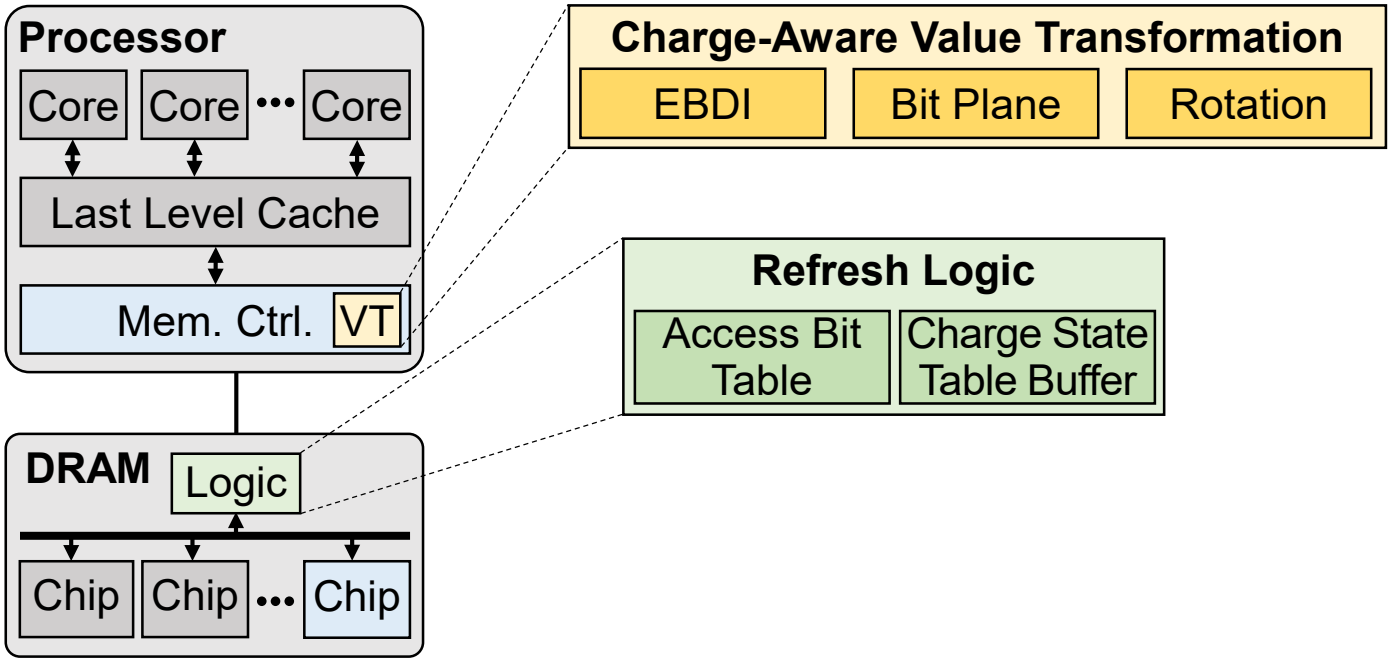
Zero-Refresh Architecture



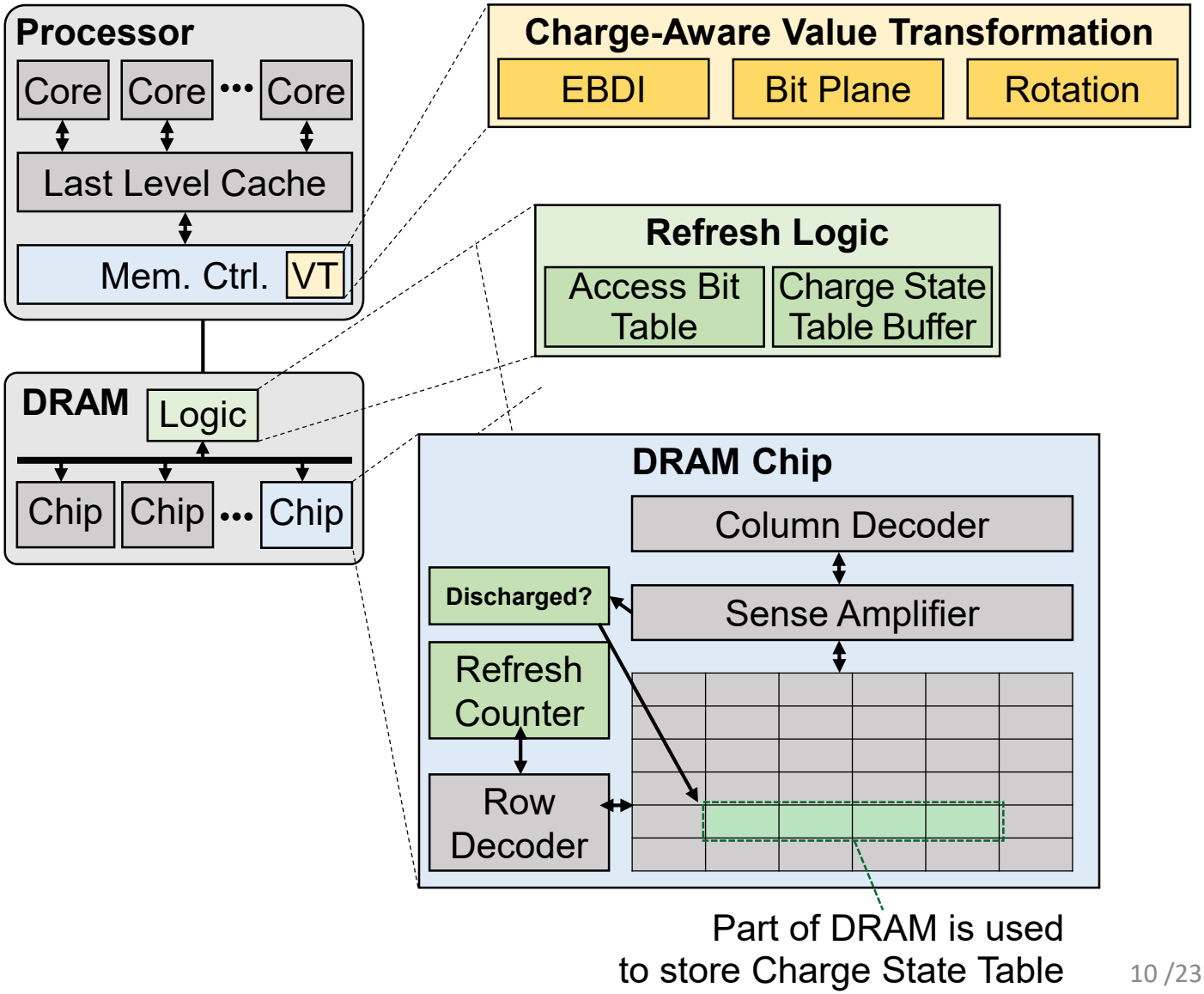
Zero-Refresh Architecture



Zero-Refresh Architecture

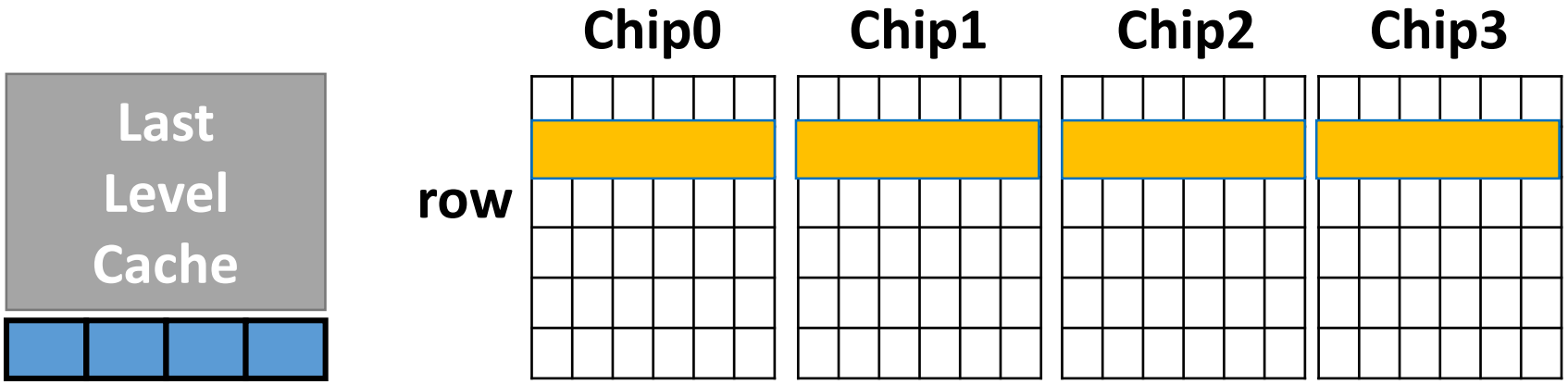


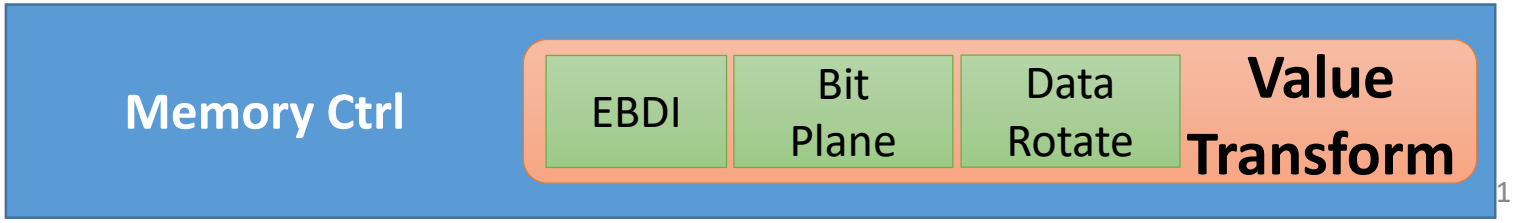
Zero-Refresh Architecture



Charge-Aware Value Transformation

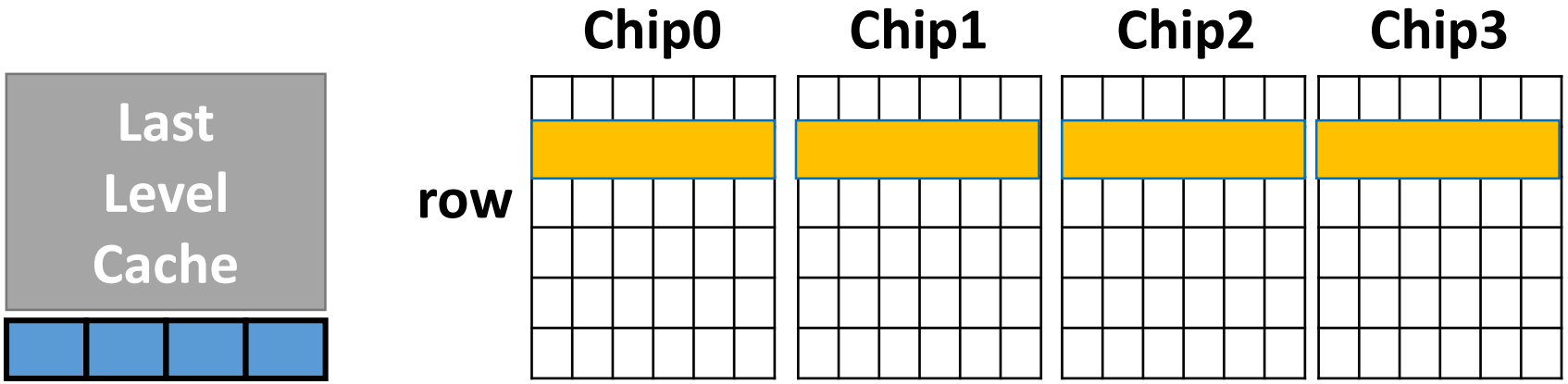
- How to make more zeros?
- How to make zeros consecutive?
- How to make fit zeros to refresh granularity?

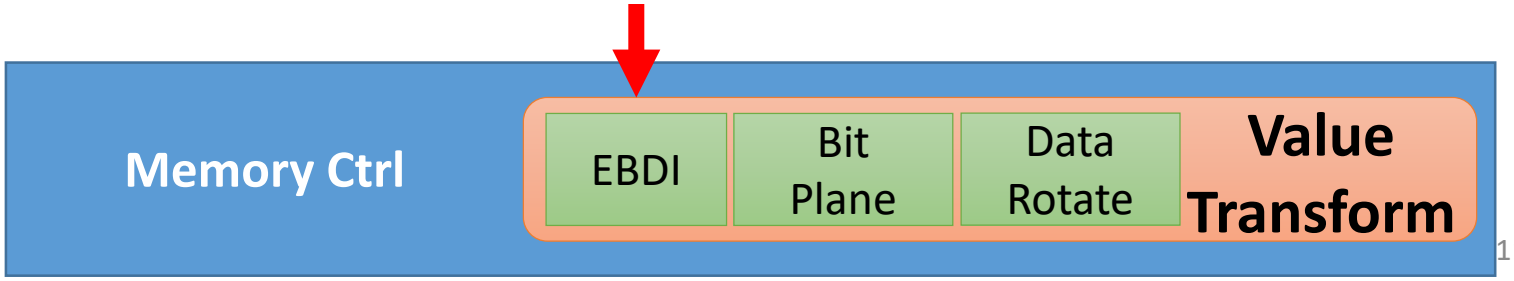




Charge-Aware Value Transformation

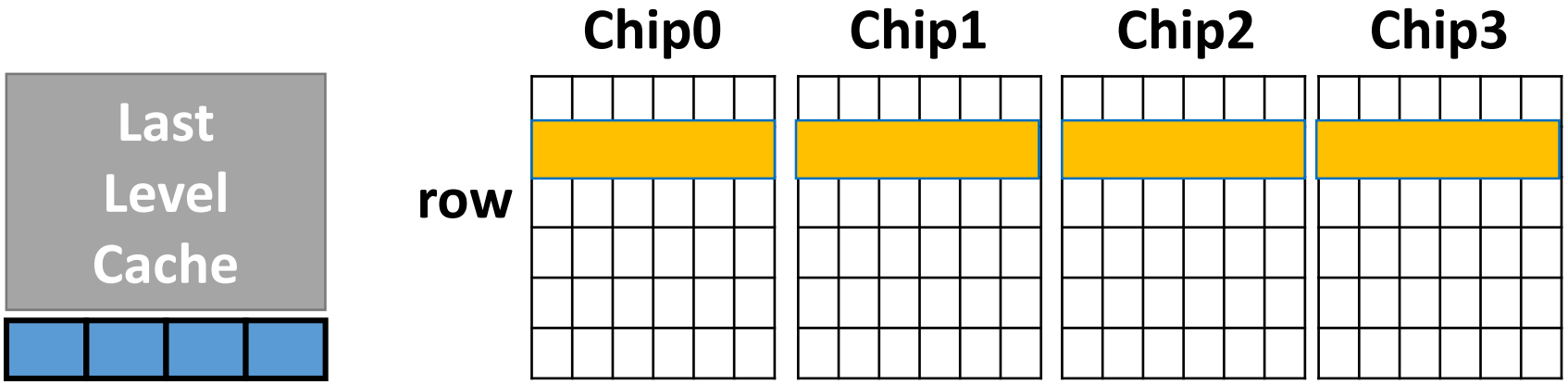
- How to make more zeros?
- How to make zeros consecutive?
- How to make fit zeros to refresh granularity?

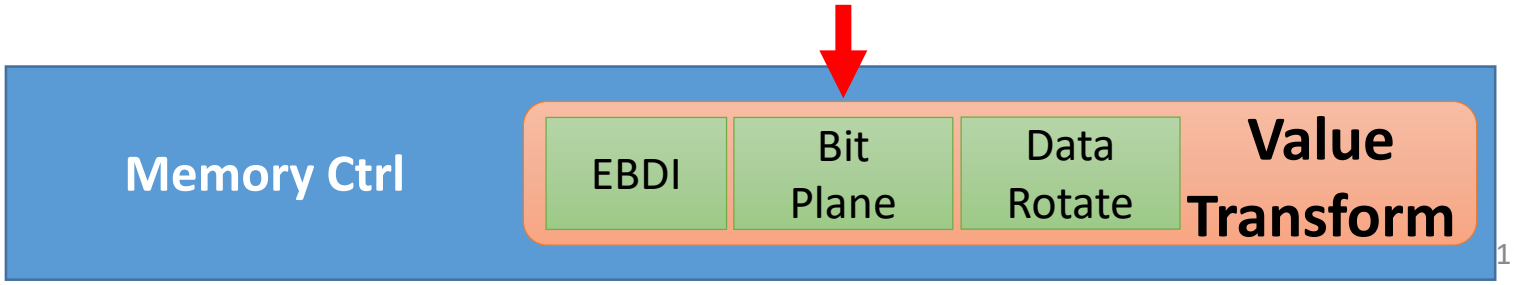




Charge-Aware Value Transformation

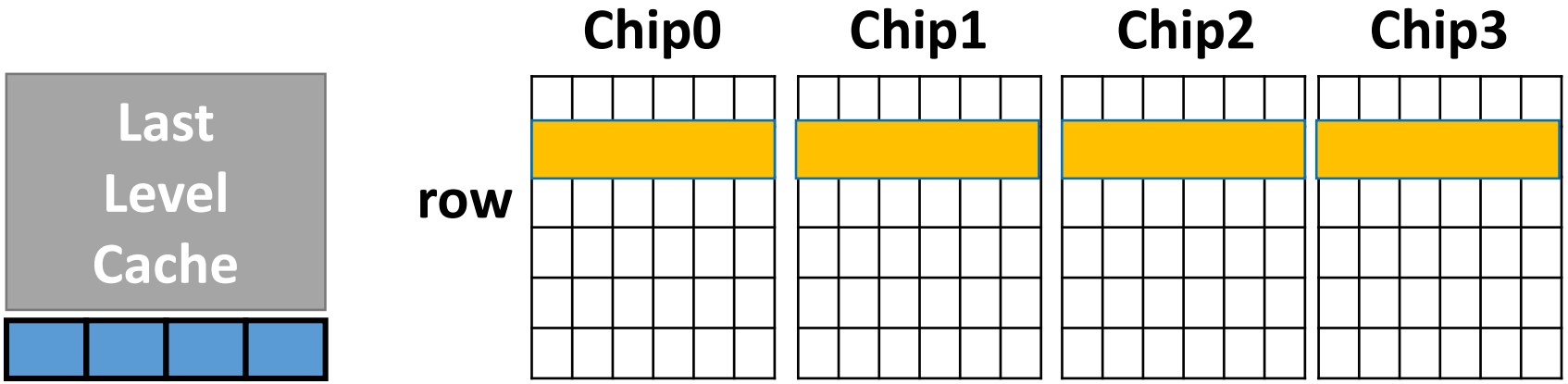
- How to make more zeros?
- How to make zeros consecutive?
- How to make fit zeros to refresh granularity?

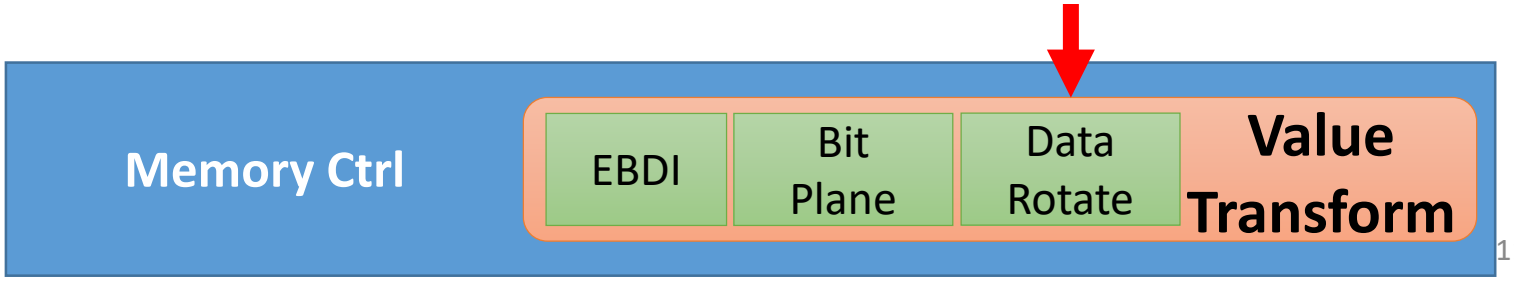




Charge-Aware Value Transformation

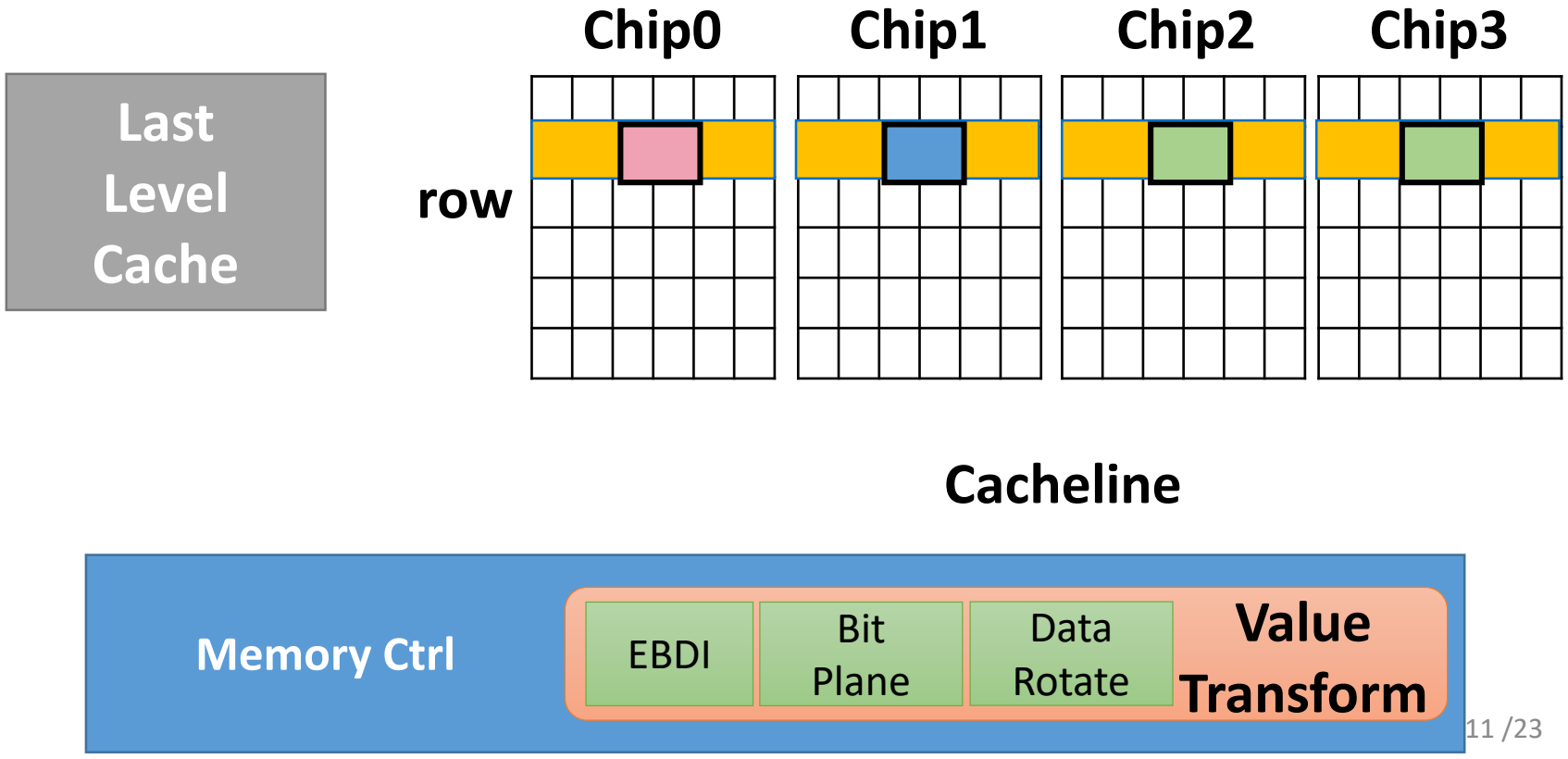
- How to make more zeros?
- How to make zeros consecutive?
- How to make fit zeros to refresh granularity?



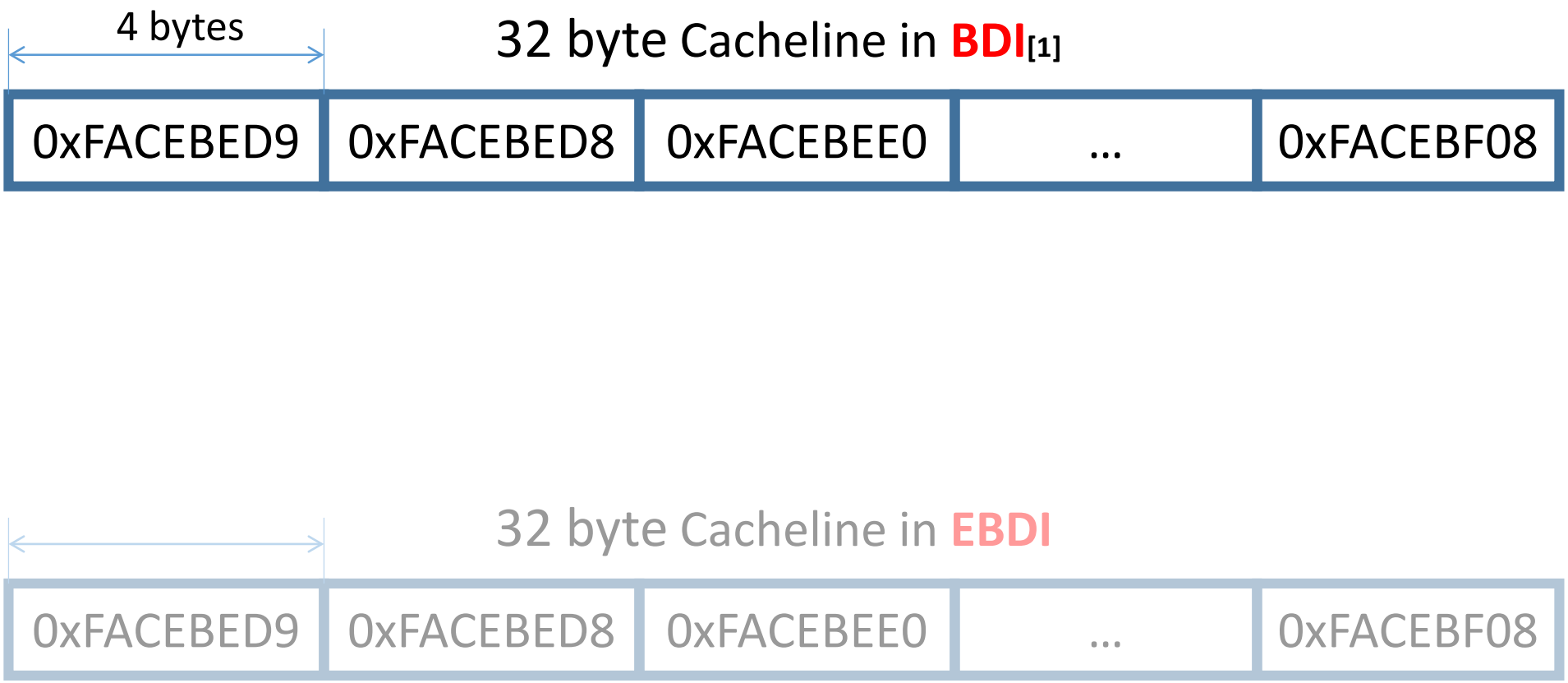


Charge-Aware Value Transformation

- How to make more zeros?
- How to make zeros consecutive?
- How to make fit zeros to refresh granularity?

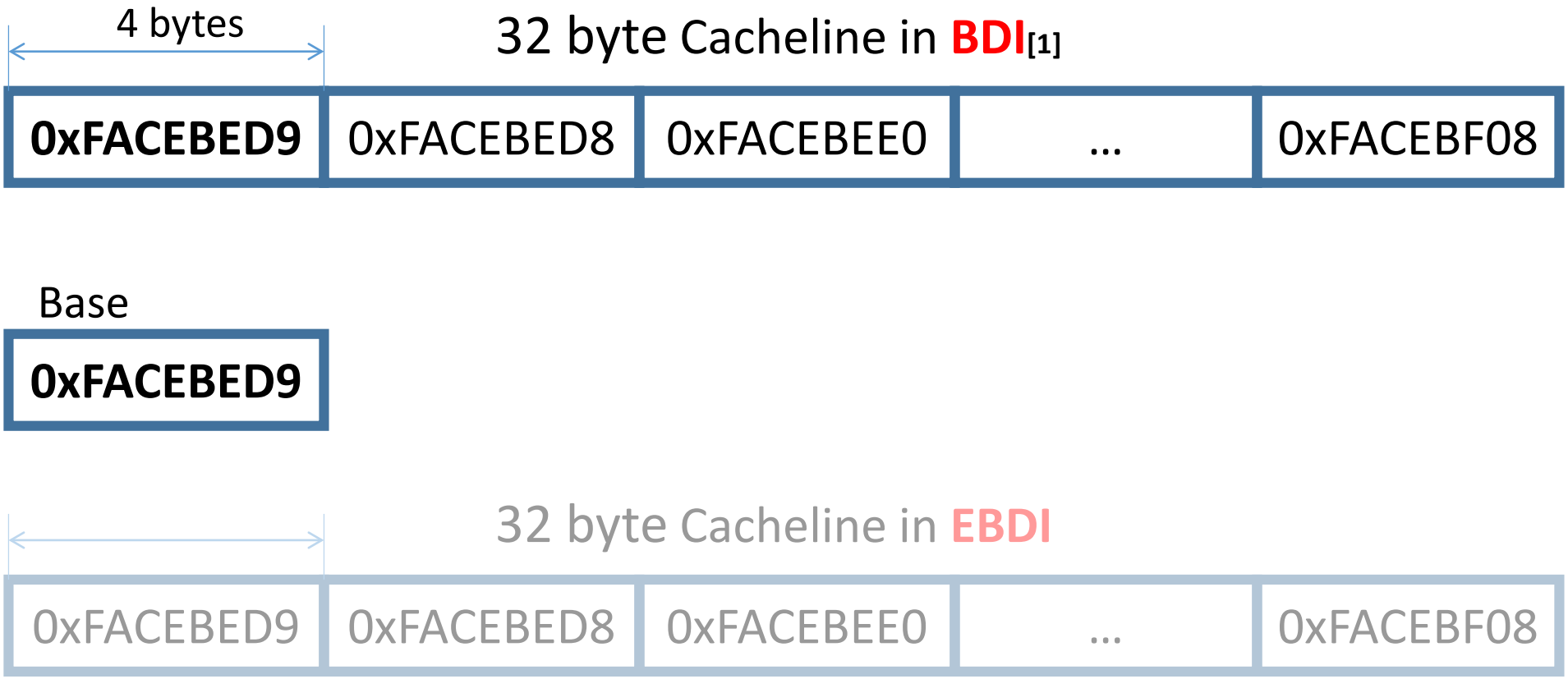


EBDI Stage: Generating Zeros



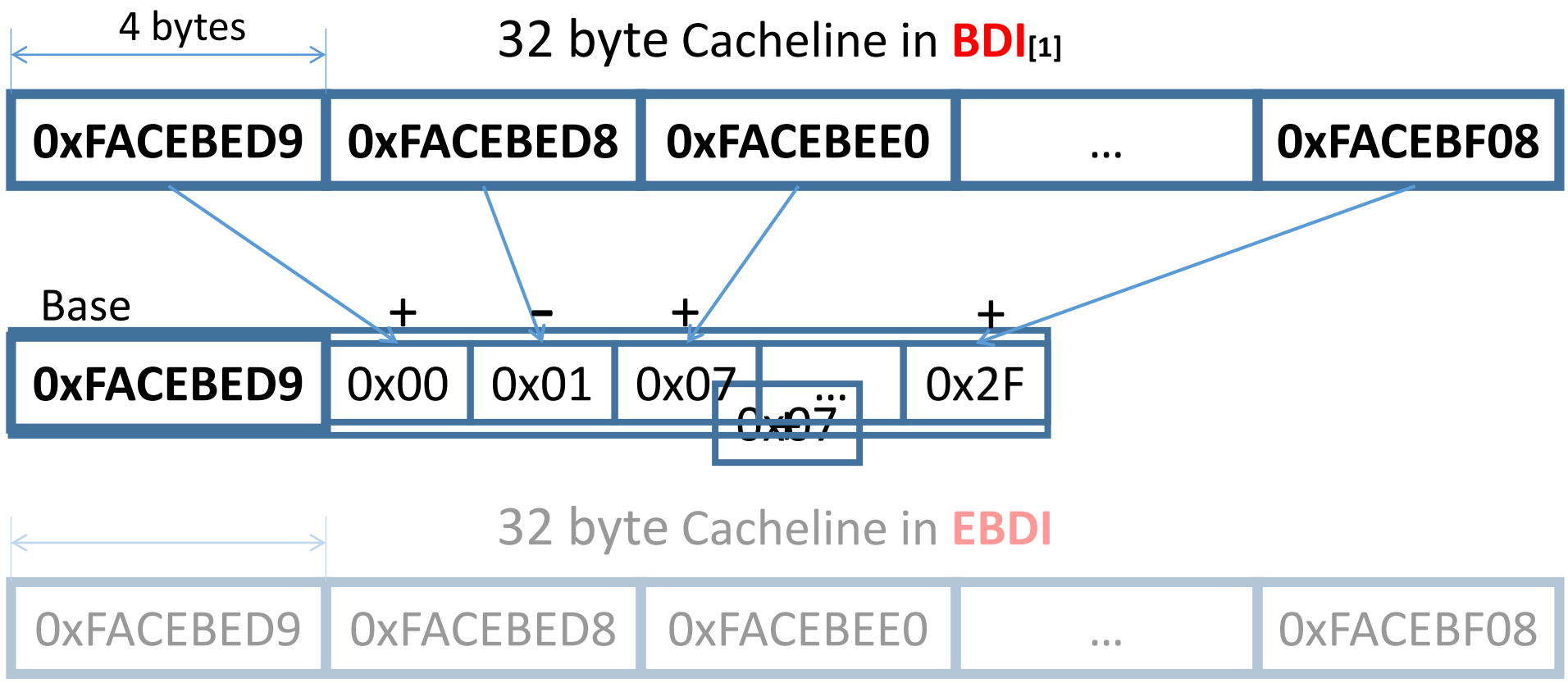
[1] Pekhimenko et al. PACT'12

EBDI Stage: Generating Zeros



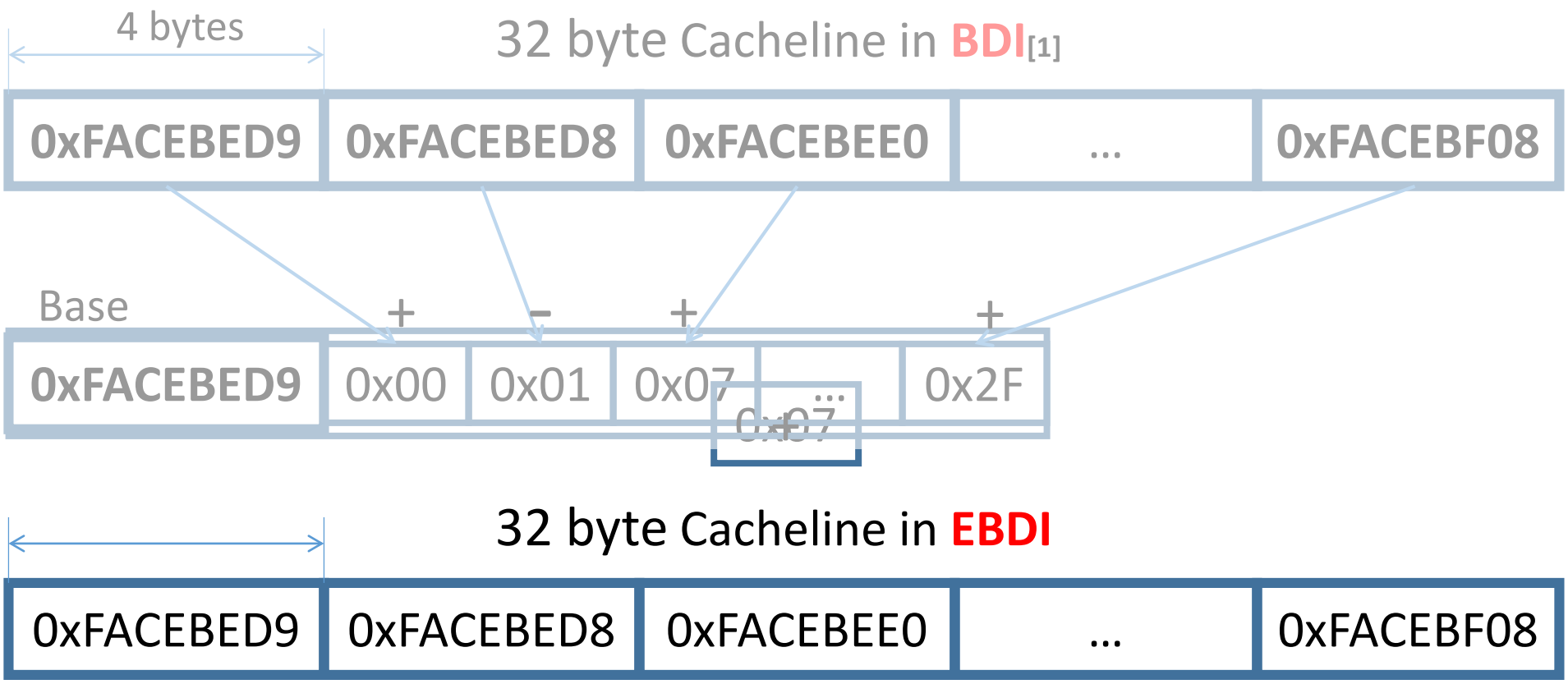
[1] Pekhimenko et al. PACT'12

EBDI Stage: Generating Zeros



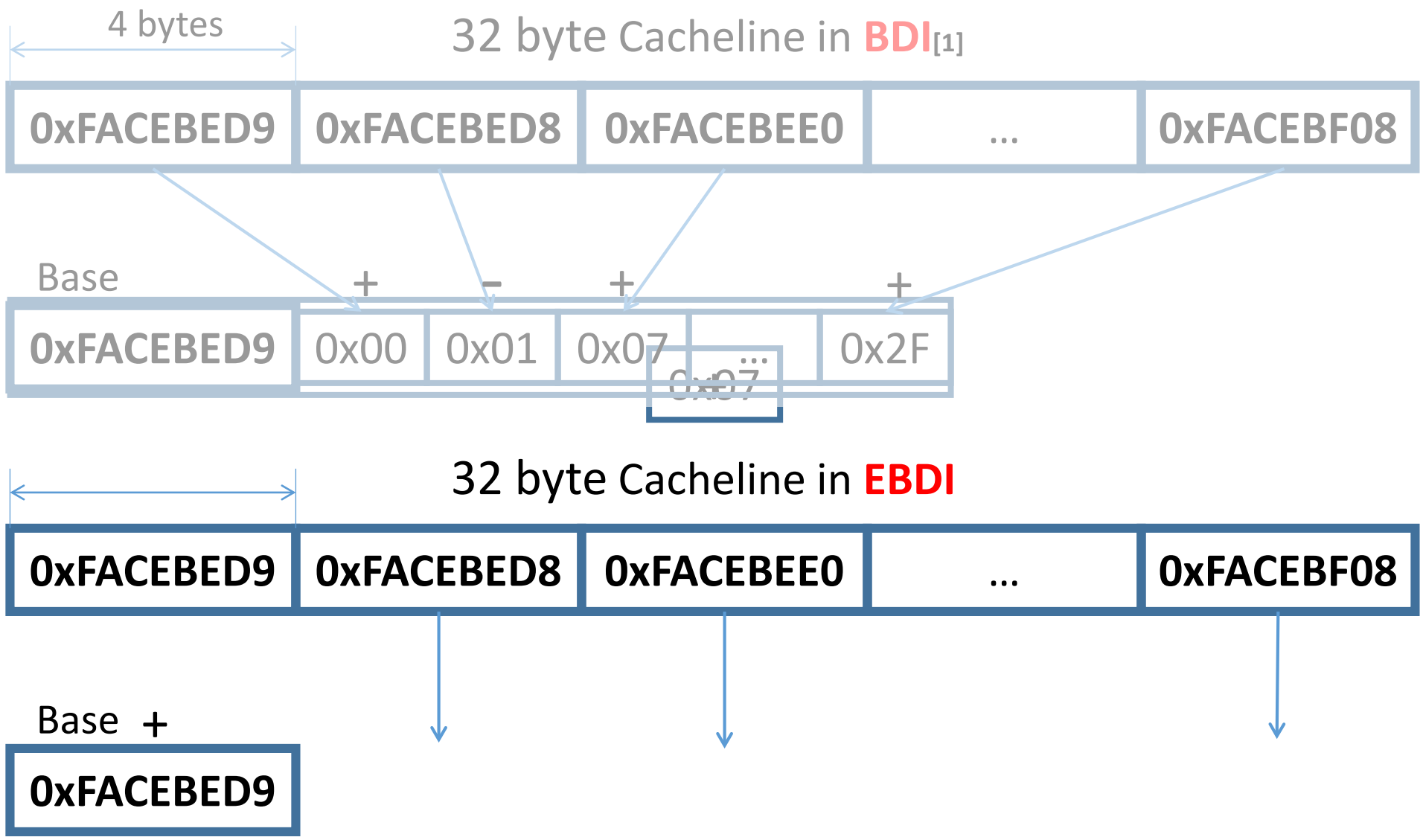
[1] Pekhimenko et al. PACT'12

EBDI Stage: Generating Zeros



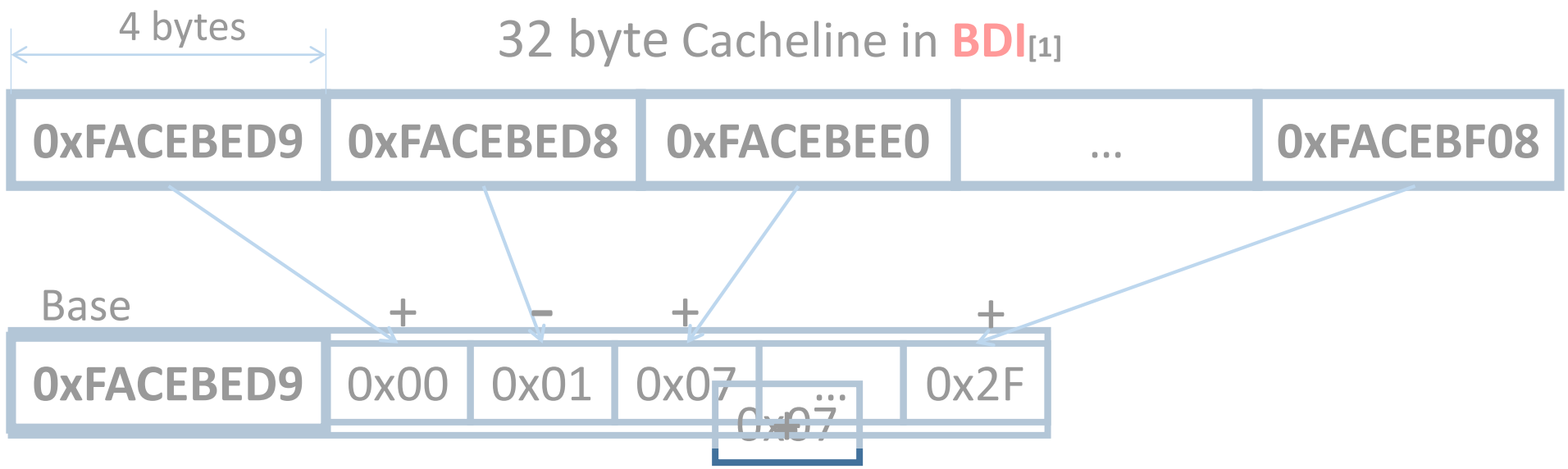
[1] Pekhimenko et al. PACT'12

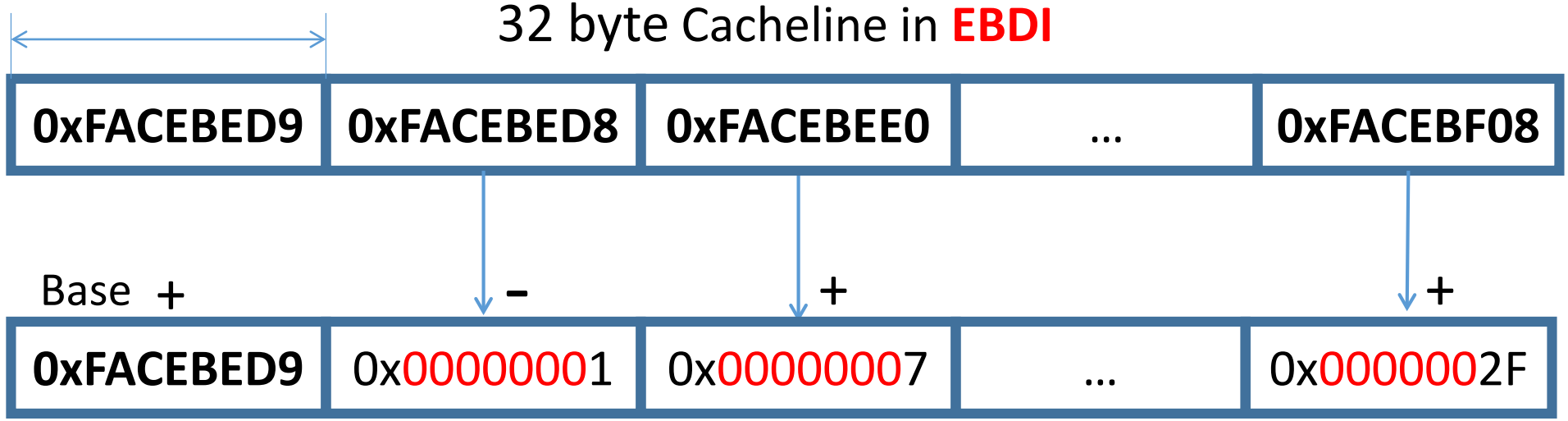
EBDI Stage: Generating Zeros



[1] Pekhimenko et al. PACT'12

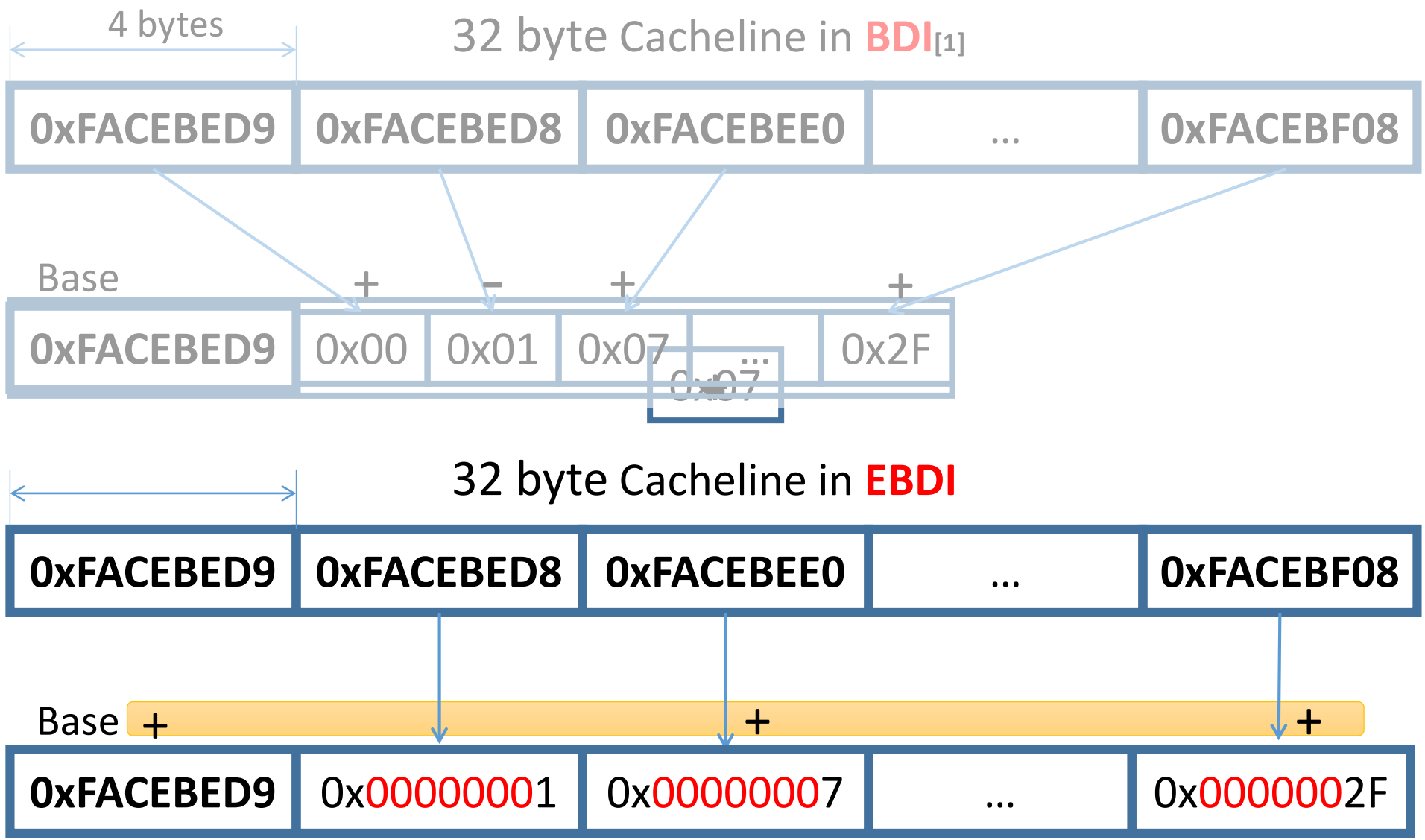
EBDI Stage: Generating Zeros





[1] Pekhimenko et al. PACT'12

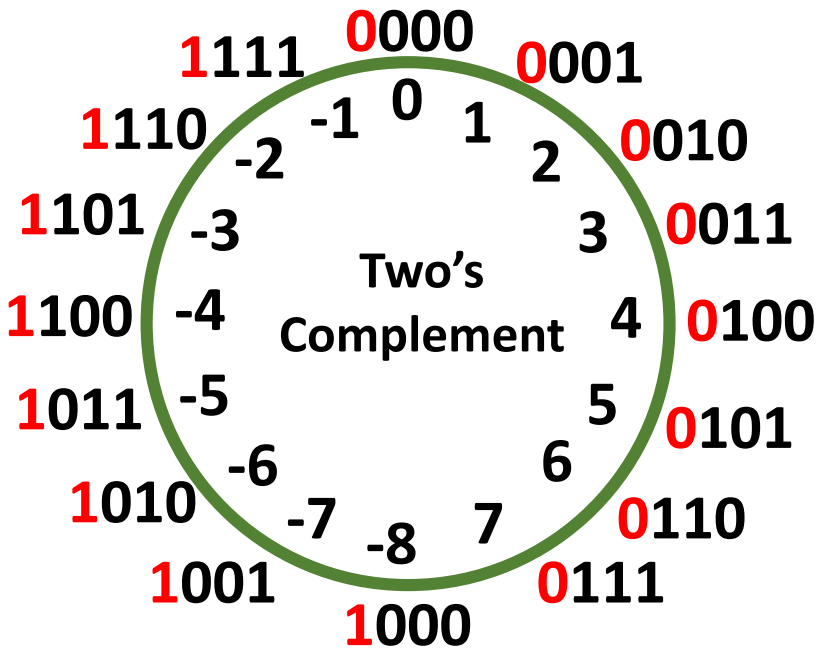
EBDI Stage: Generating Zeros



[1] Pekhimenko et al. PACT'12

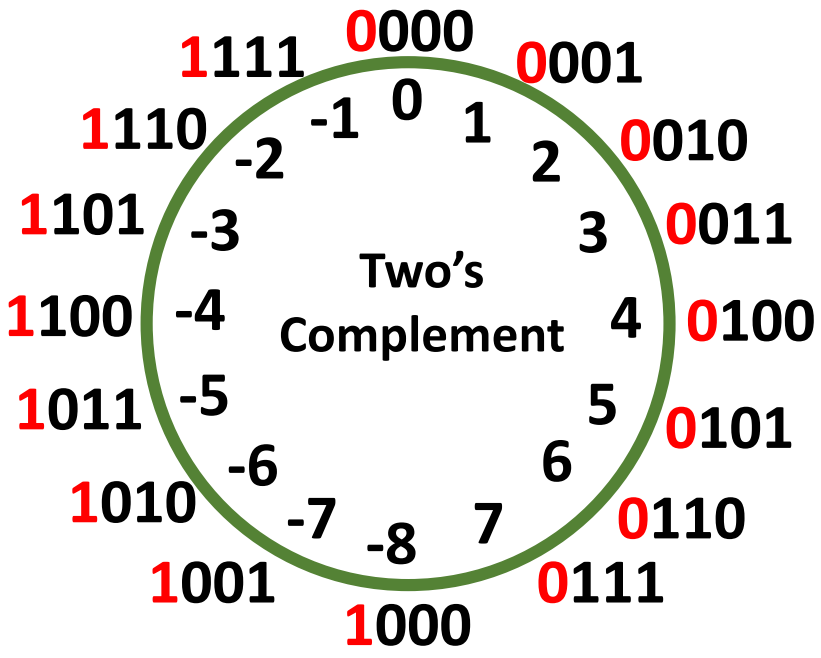
Encoded BDI

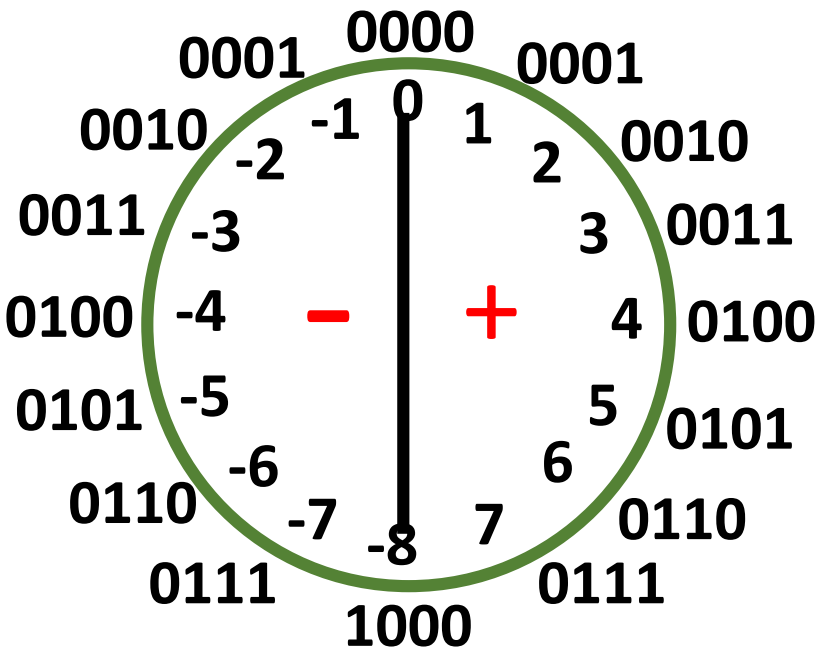
- Exploit BDI for more zeros as possible
  - Sign-bit to LSB && flip Non-sign-bits
  - Mostly zeros on higher bits



Encoded BDI

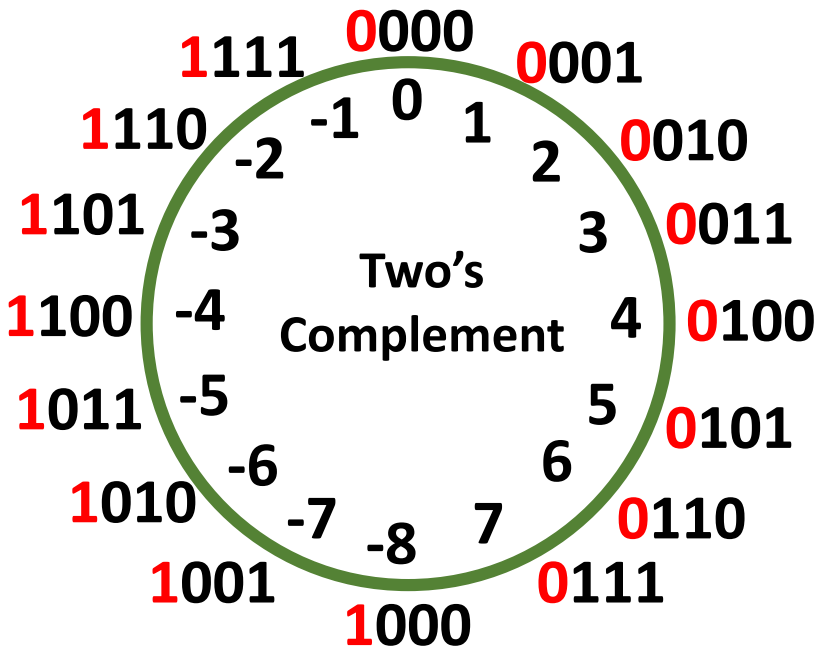
- Exploit BDI for more zeros as possible
  - Sign-bit to LSB && flip Non-sign-bits
  - Mostly zeros on higher bits

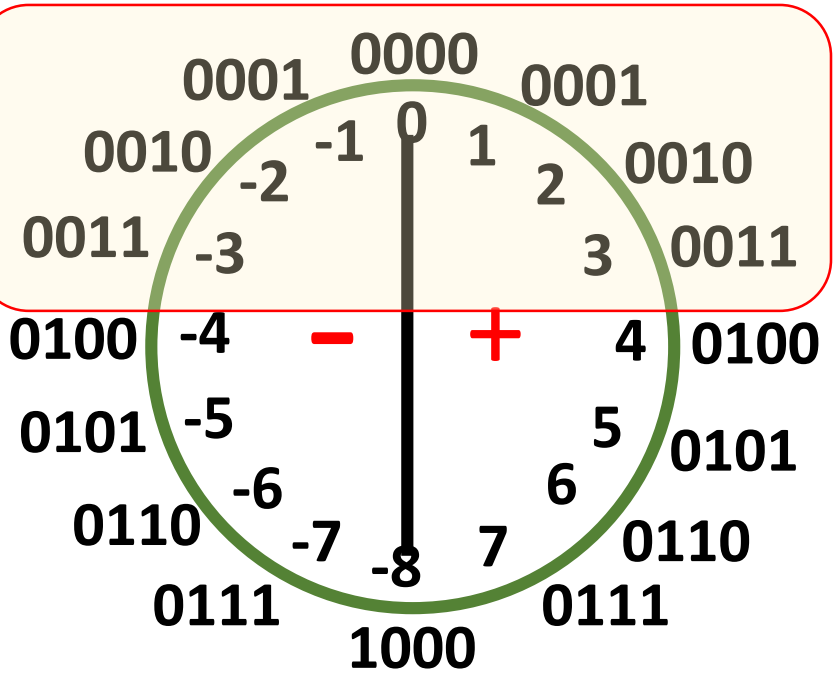




Encoded BDI

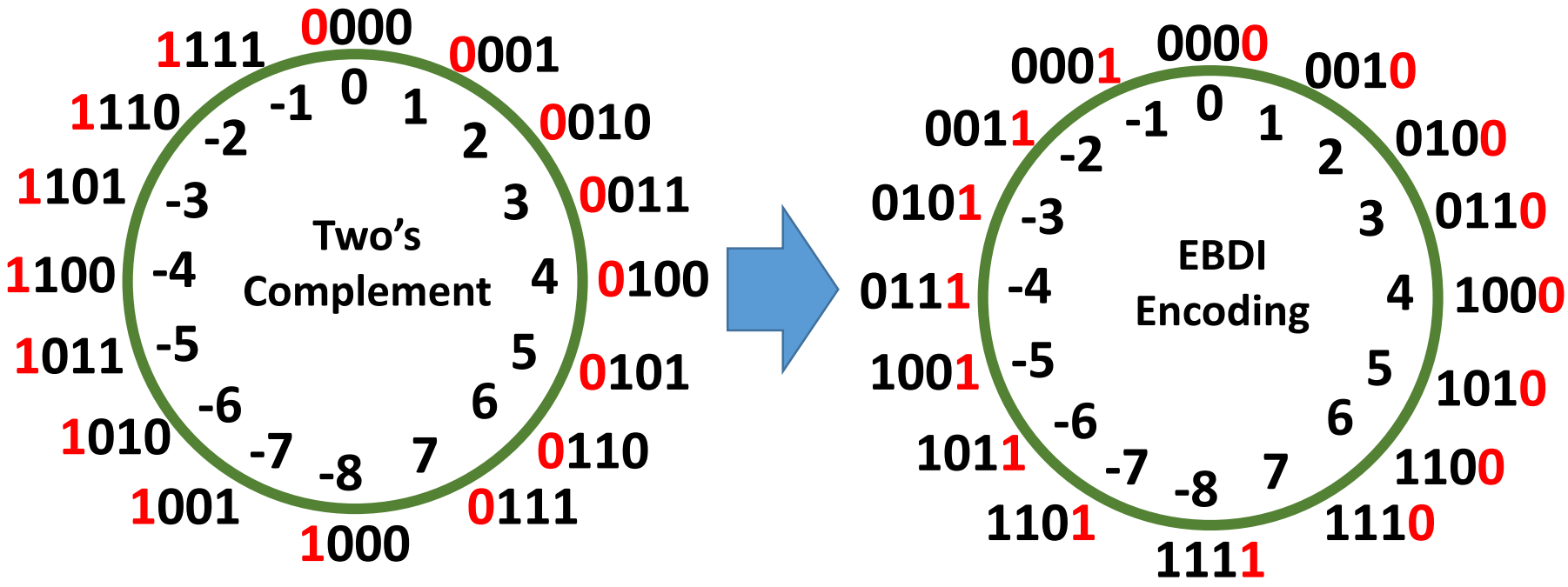
- Exploit BDI for more zeros as possible
  - Sign-bit to LSB && flip Non-sign-bits
  - Mostly zeros on higher bits



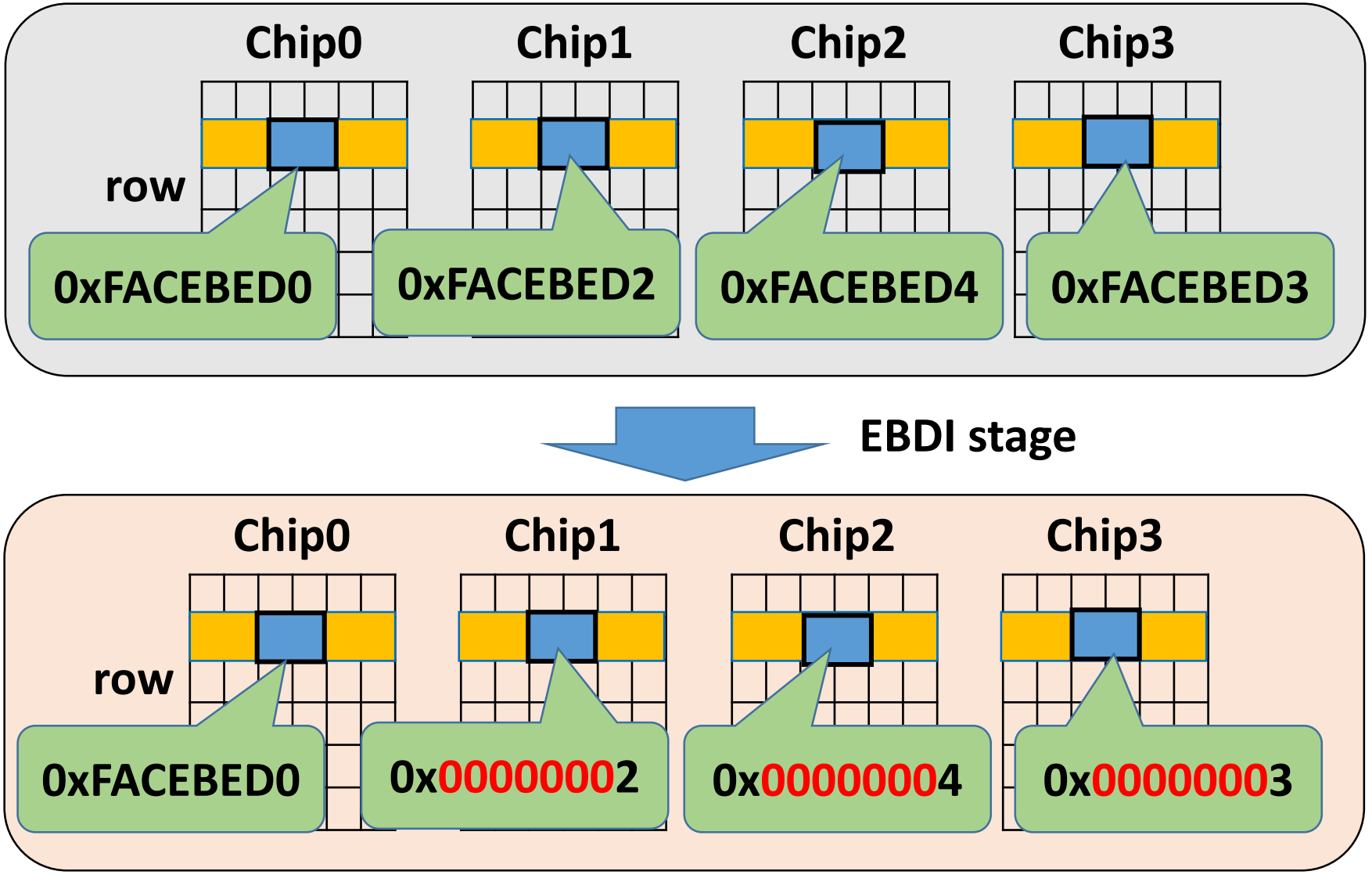


Encoded BDI

- Exploit BDI for more zeros as possible
  - Sign-bit to LSB && flip Non-sign-bits
  - Mostly zeros on higher bits



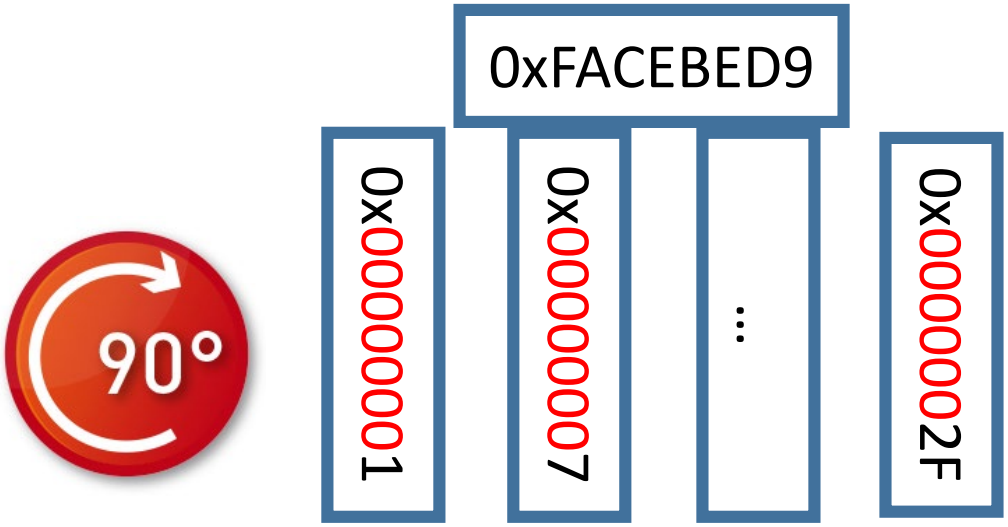
Memory State



Bit Plane Stage: Consecutive Zeros

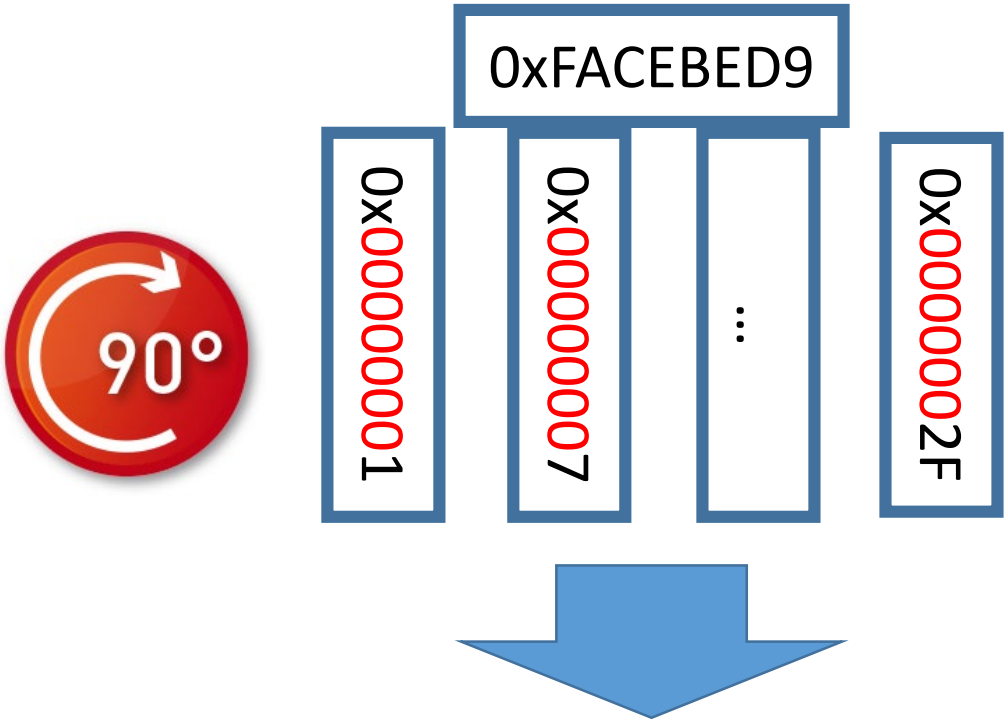
Bit Plane Stage: Consecutive Zeros

0xFACEBED9	0x00000001	0x00000007	…	0x0000002F



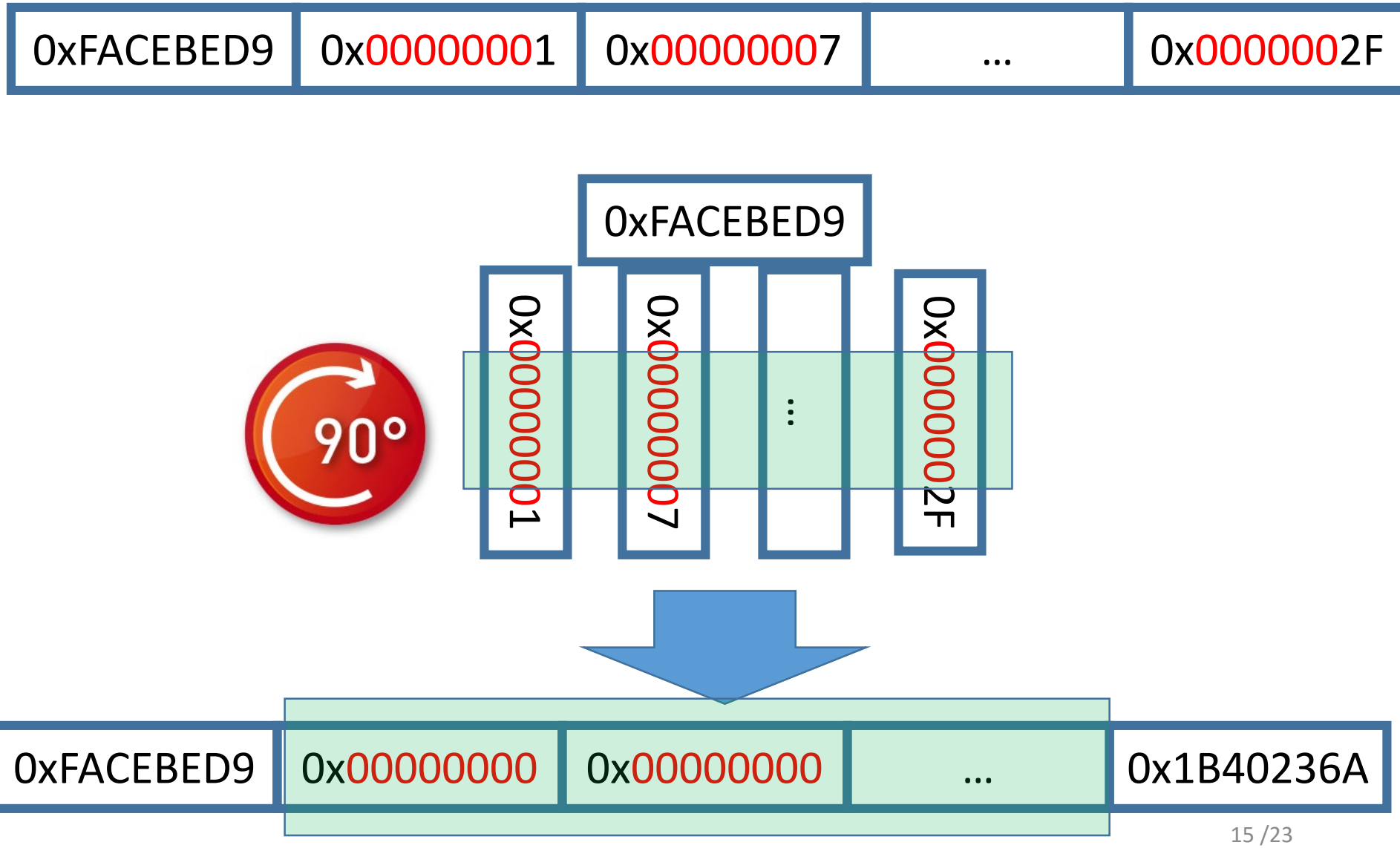
Bit Plane Stage: Consecutive Zeros

0xFACEBED9	0x00000001	0x00000007	…	0x0000002F

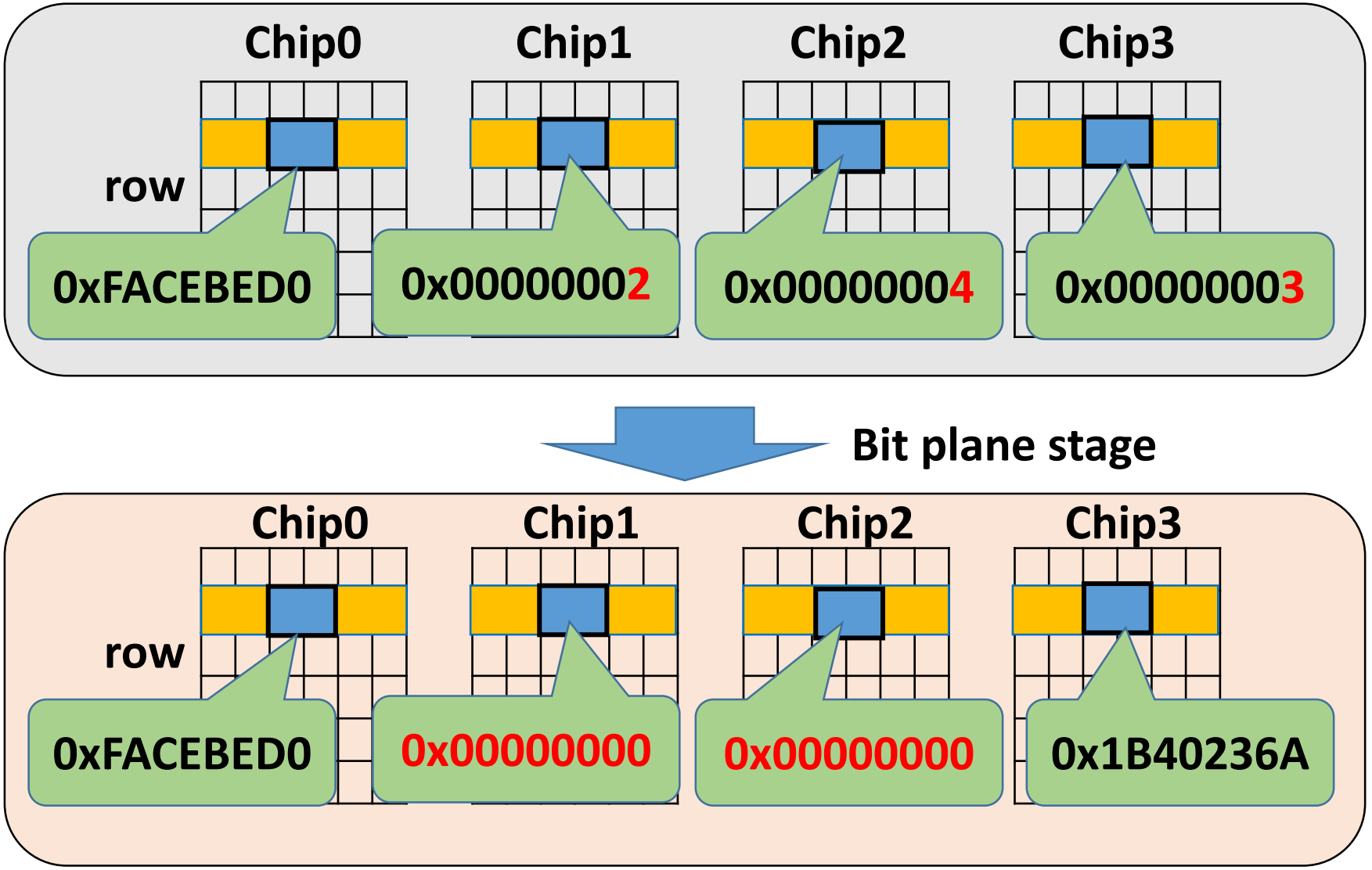


0xFACEBED9	0x00000000	0x00000000	…	0x1B40236A

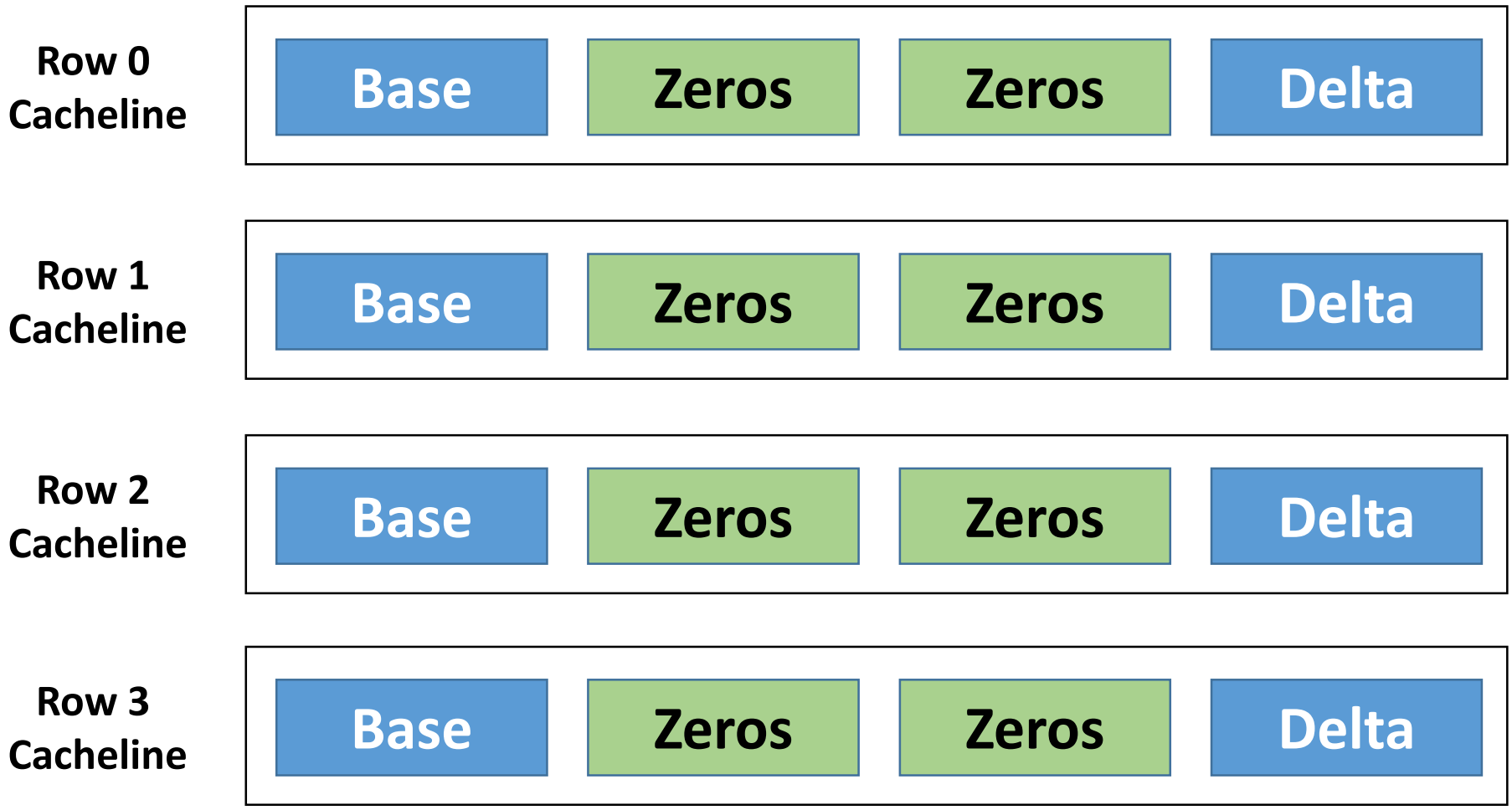
Bit Plane Stage: Consecutive Zeros



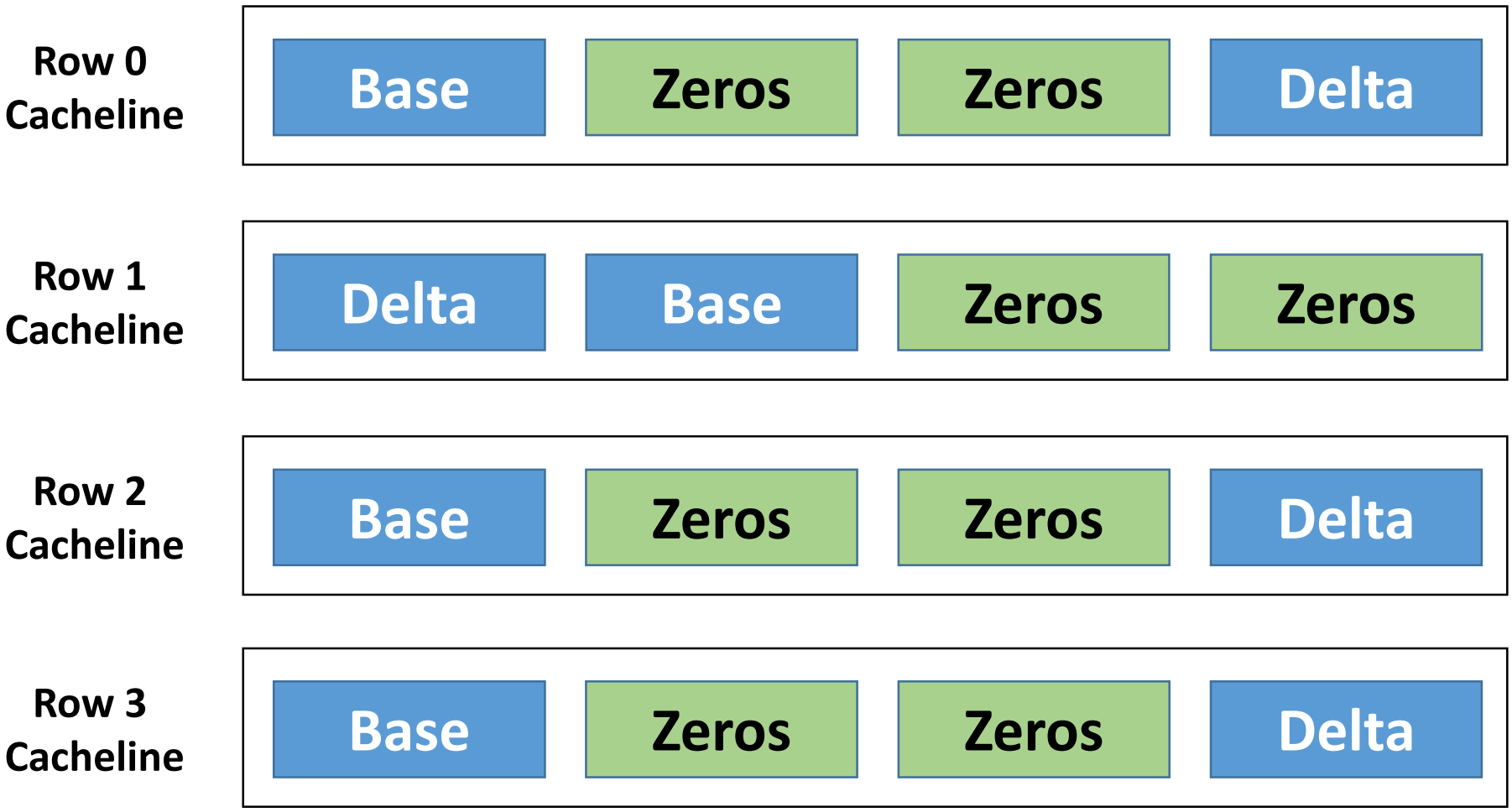
Memory State



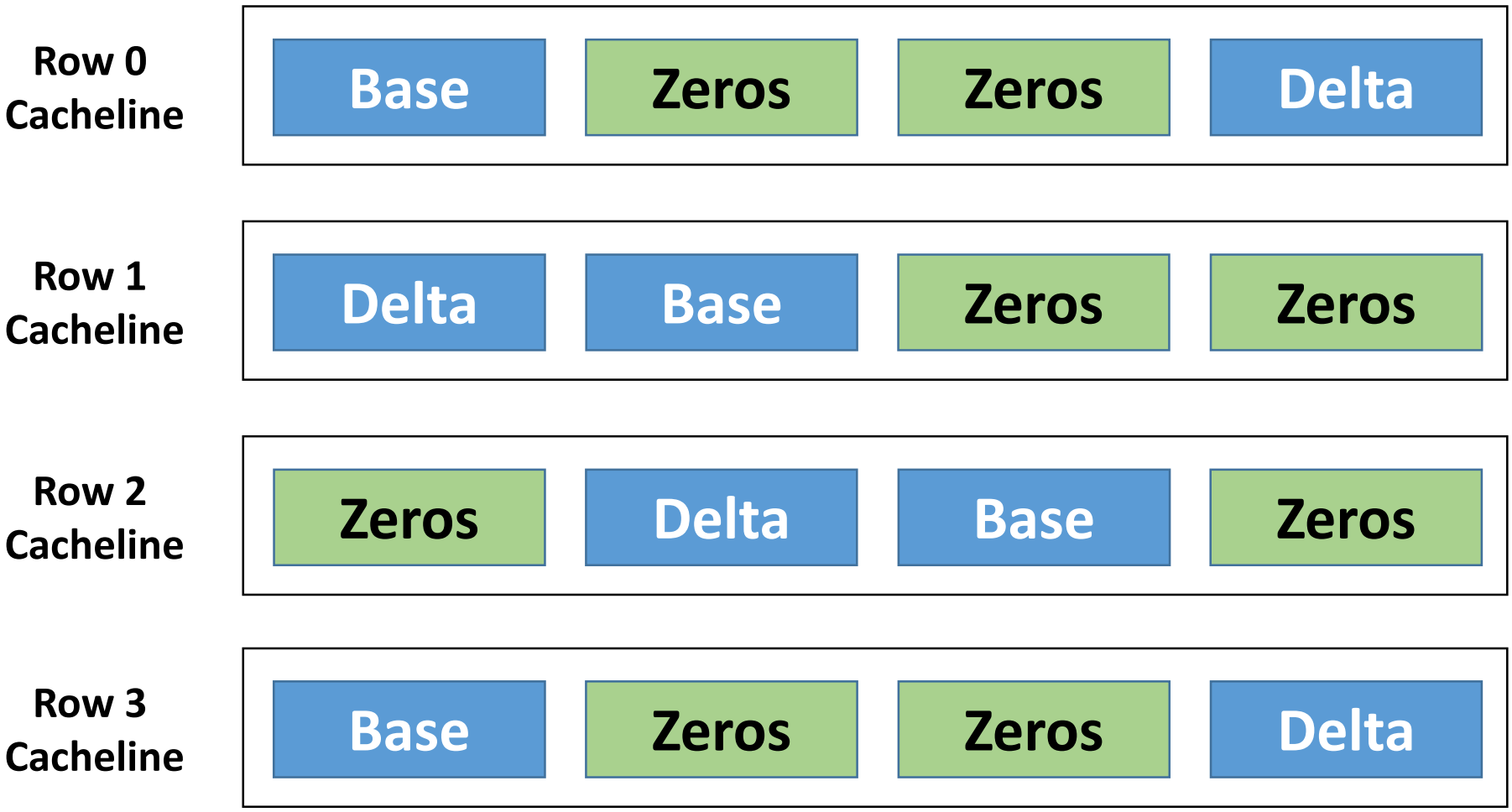
Data Rotation Stage



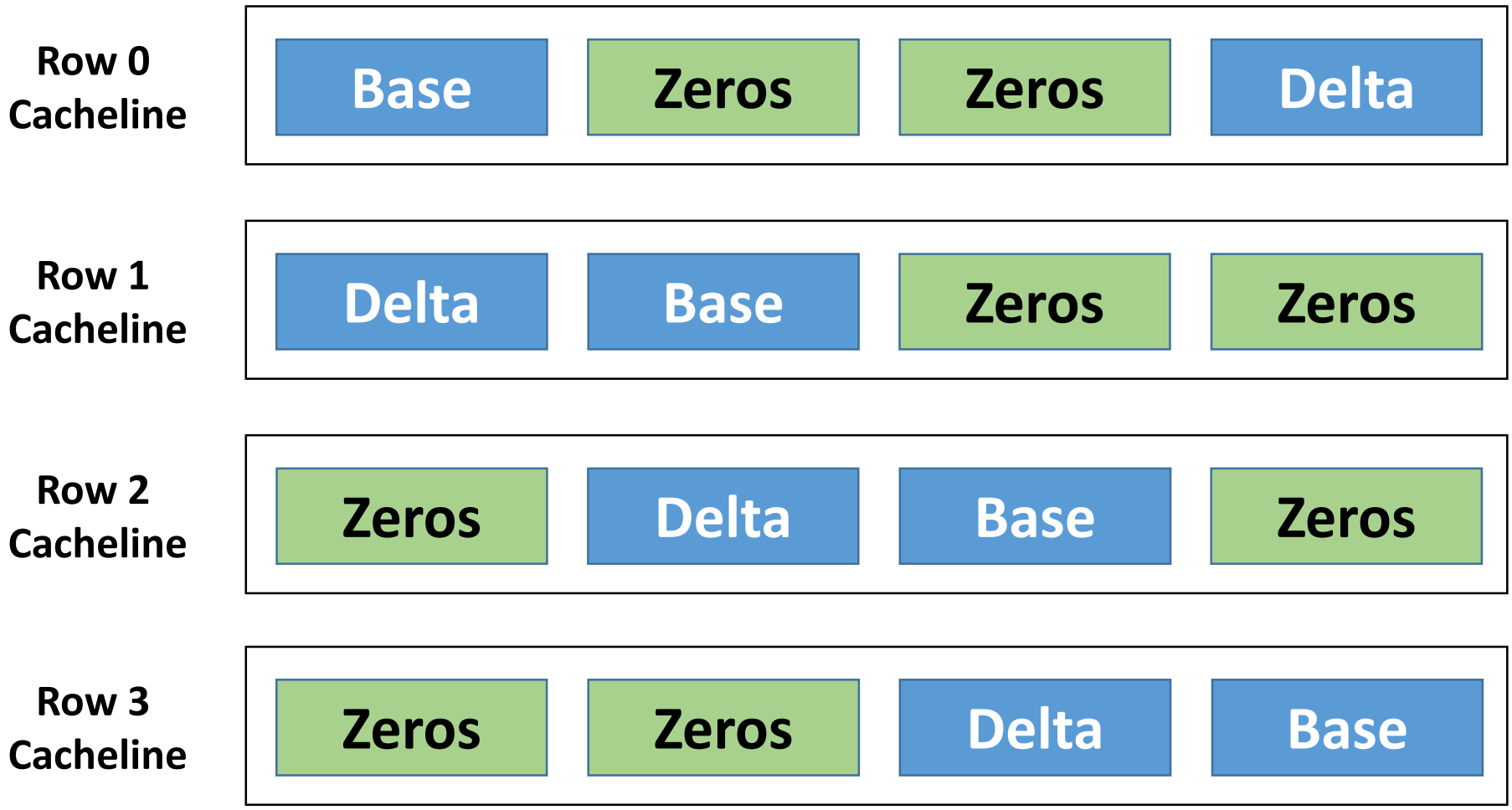
Data Rotation Stage



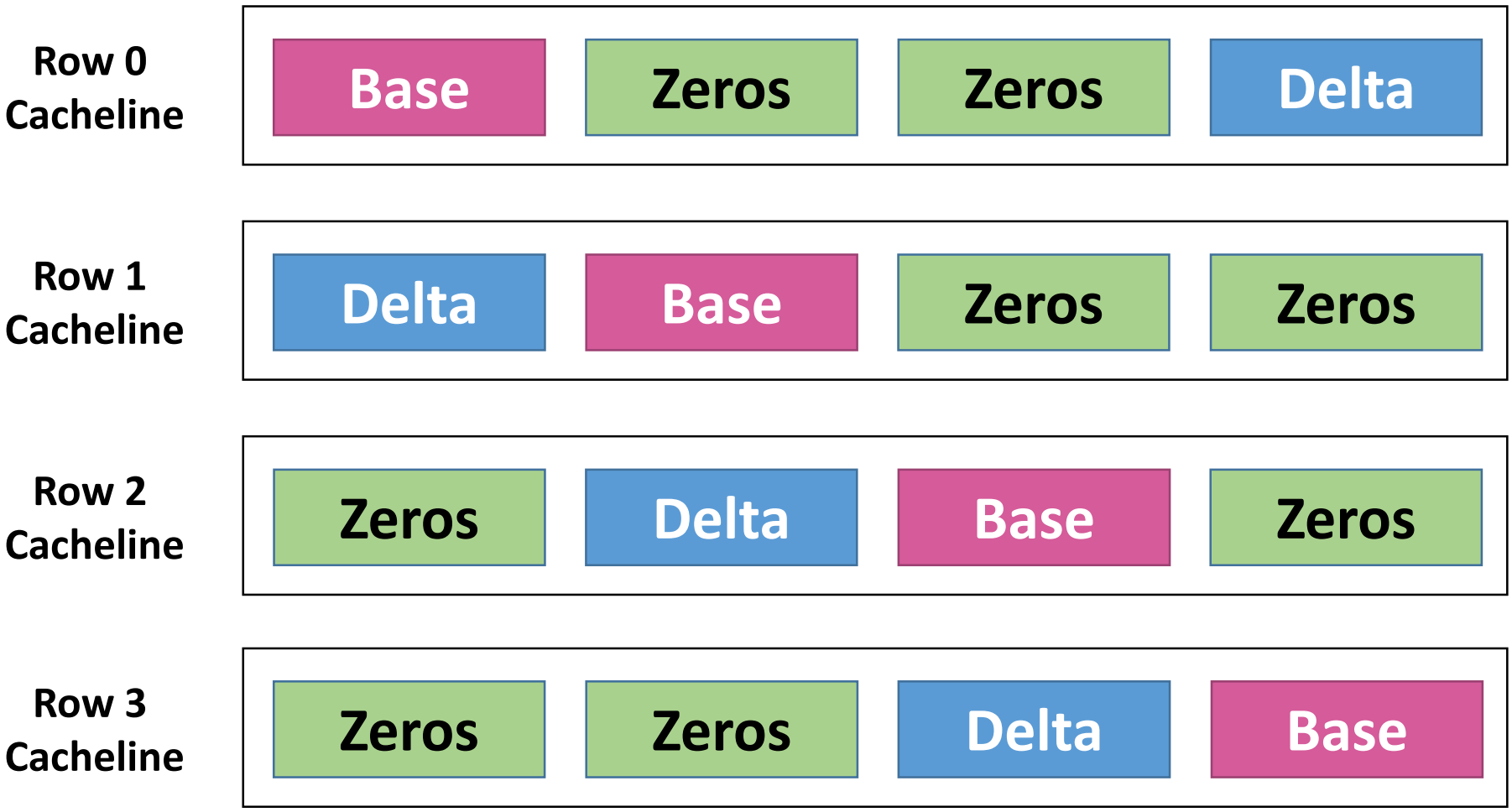
Data Rotation Stage



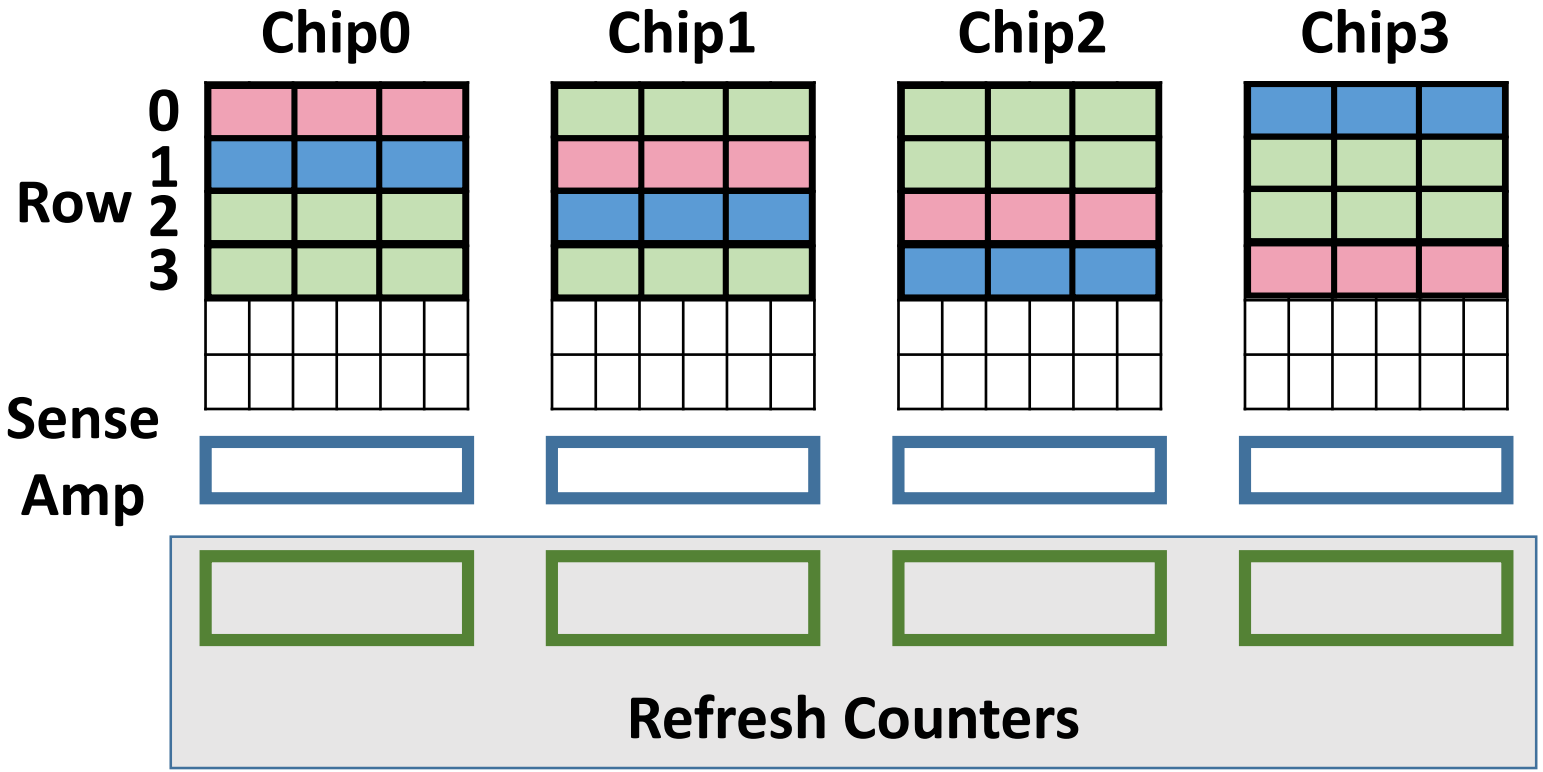
Data Rotation Stage



Data Rotation Stage

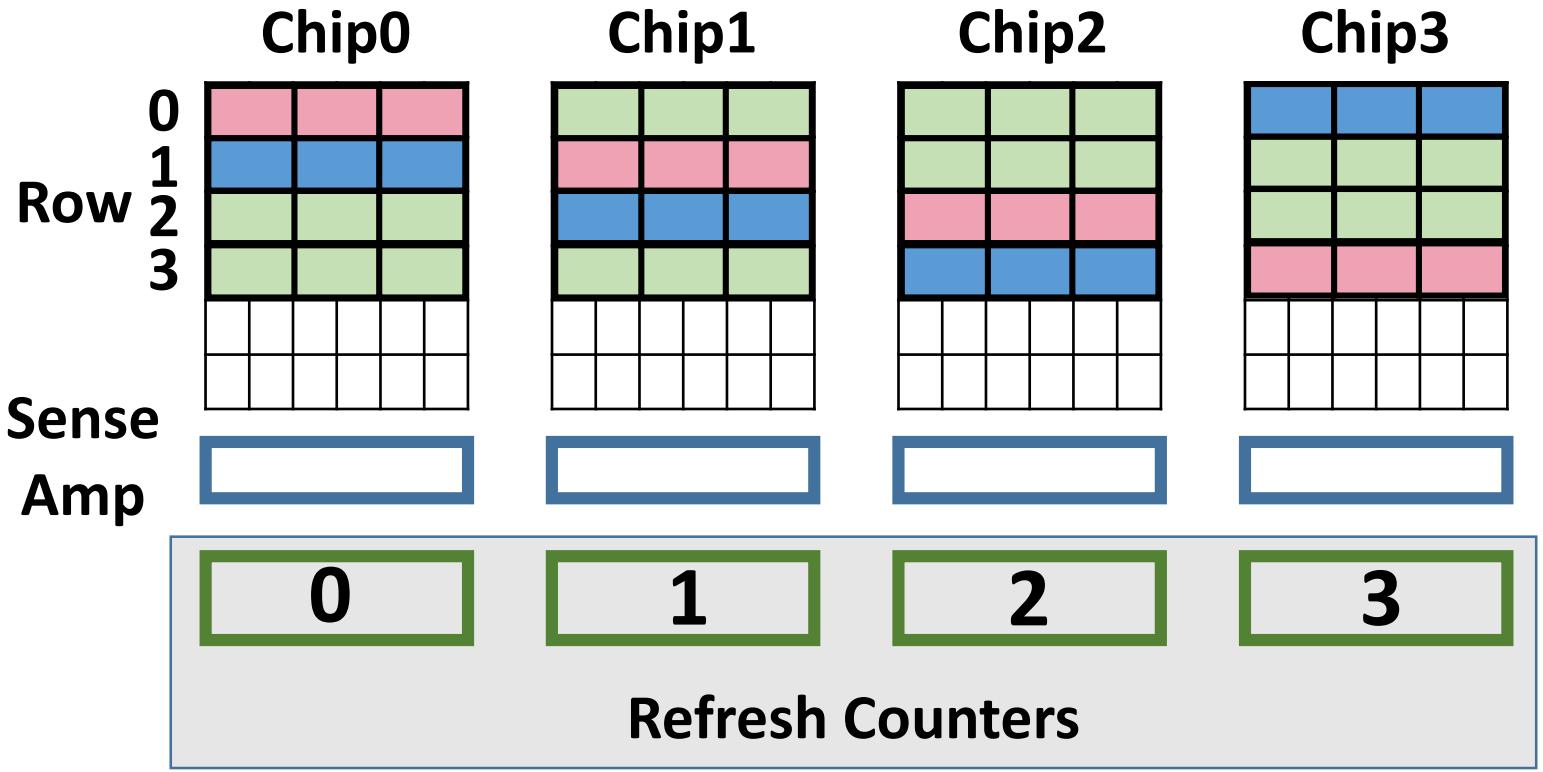


Refresh Counter Modification



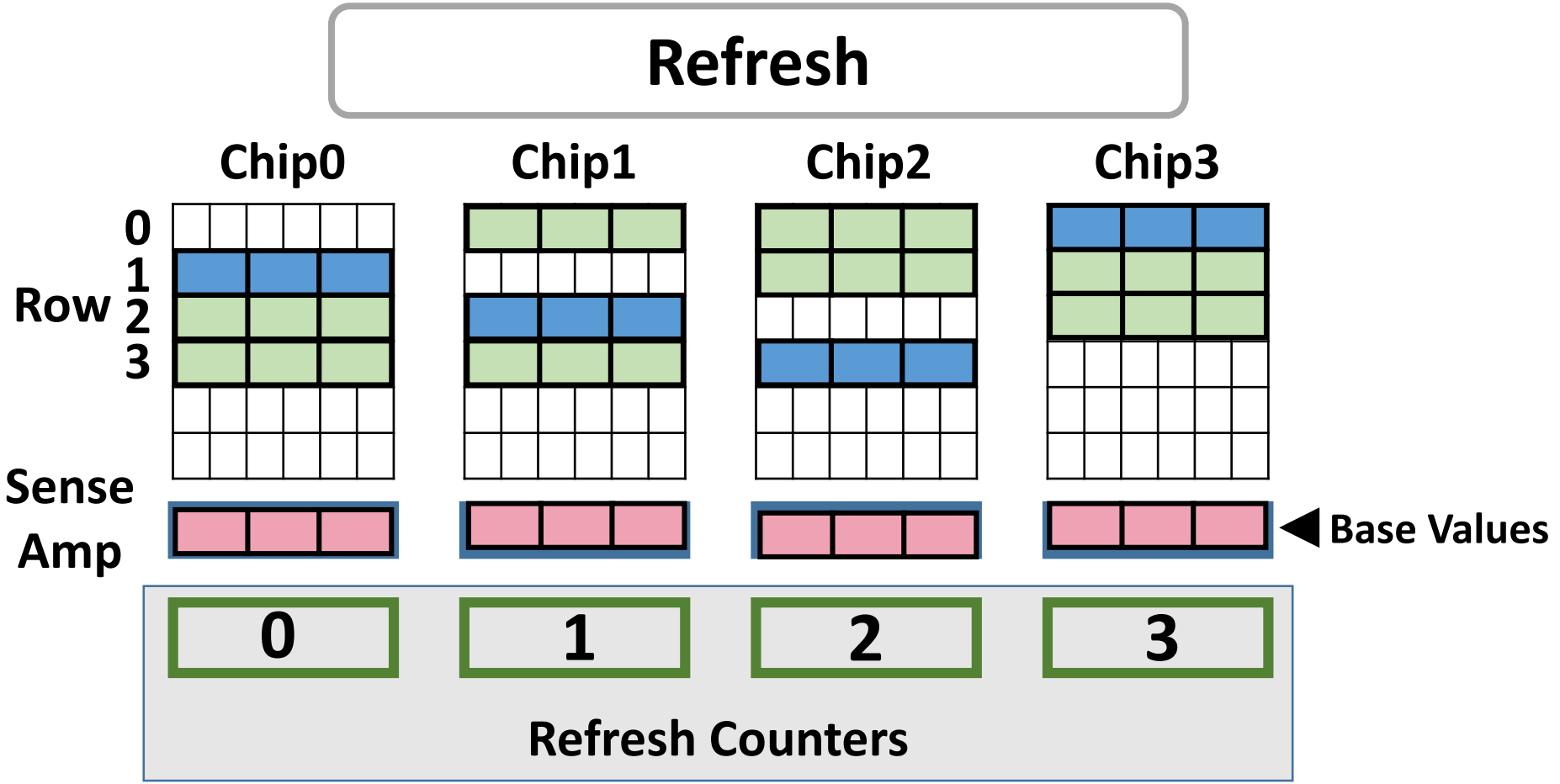
$$RefreshRow = \big((initRow + n) mod\ numChip\big) + \left(\frac{n}{numChip}\right) * numChip$$

Refresh Counter Modification



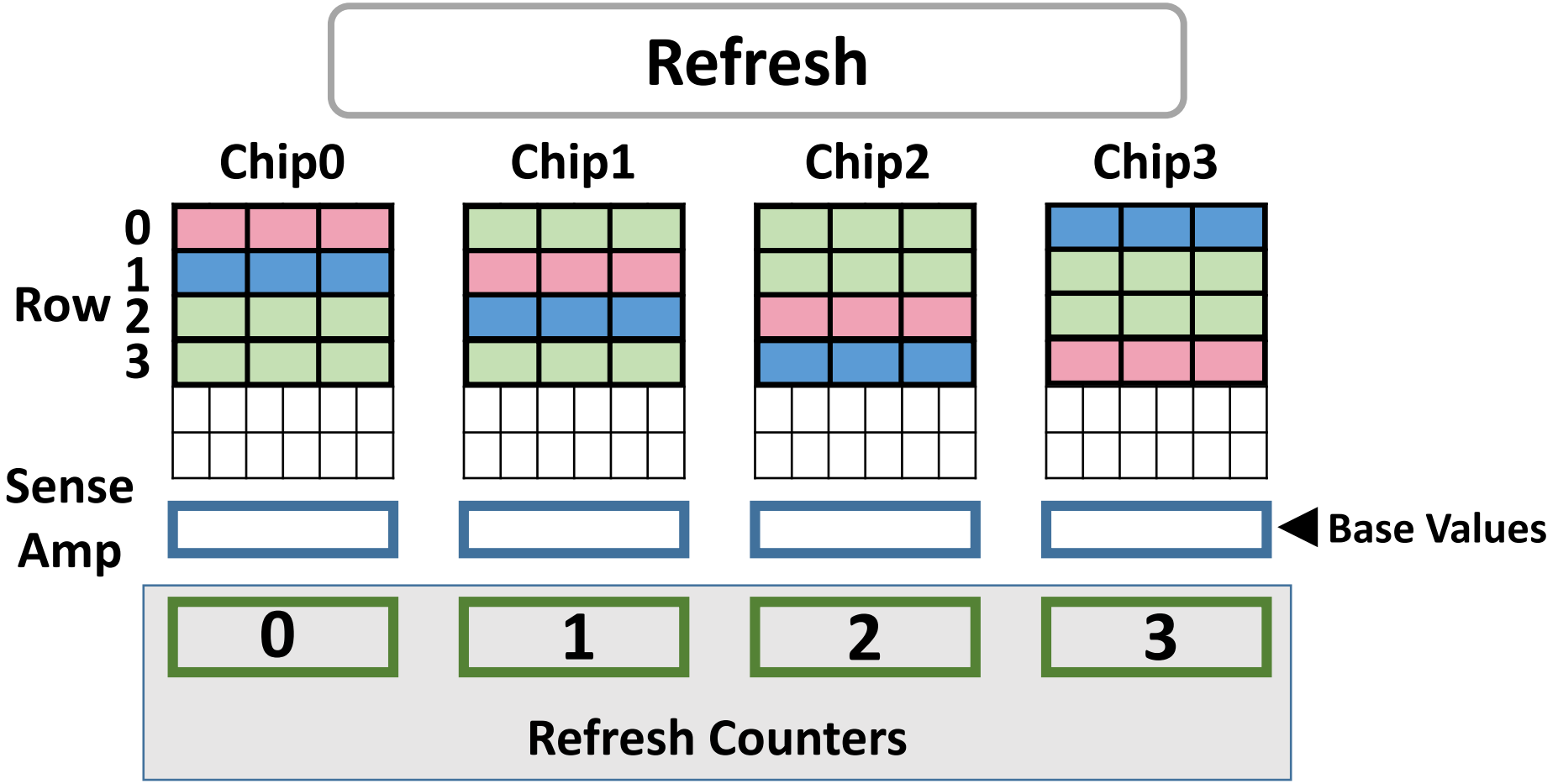
$$RefreshRow = \big((initRow + n) mod\ numChip\big) + \left(\frac{n}{numChip}\right) * numChip$$

Refresh Counter Modification



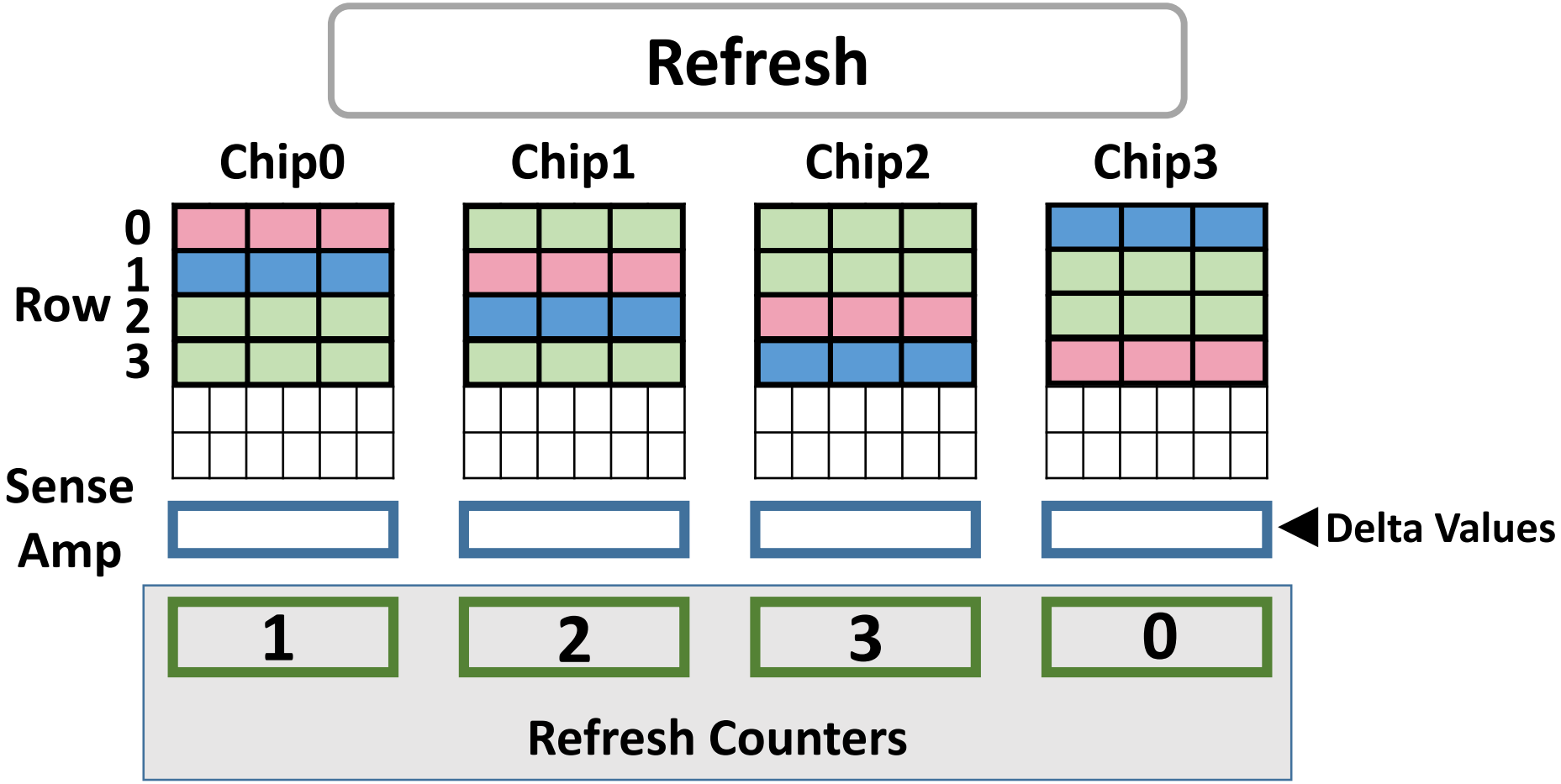
$$RefreshRow = \big((initRow + n) mod\ numChip\big) + \left(\frac{n}{numChip}\right) * numChip$$

Refresh Counter Modification



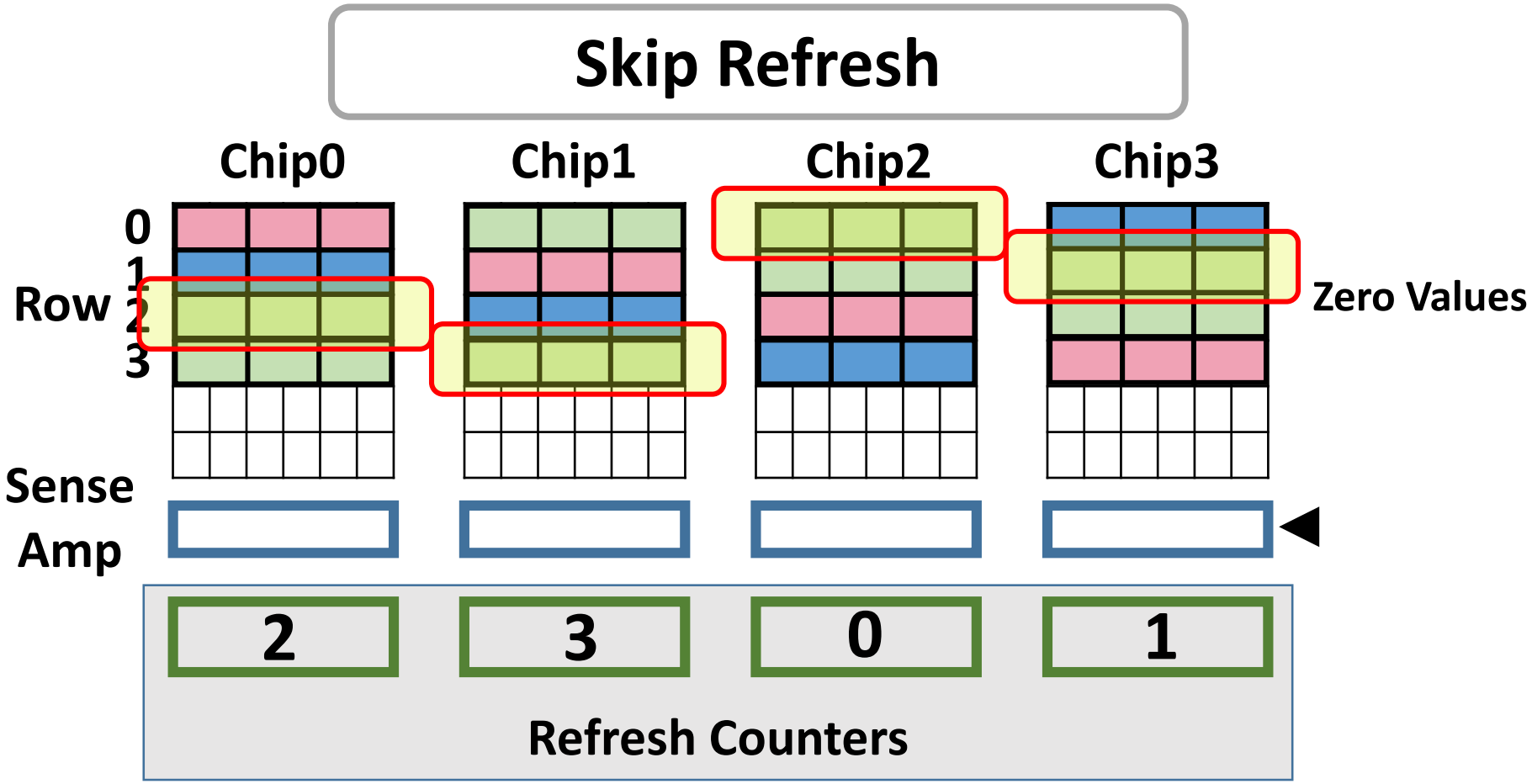
$$RefreshRow = \big((initRow + n) mod\ numChip\big) + \left(\frac{n}{numChip}\right) * numChip$$

Refresh Counter Modification



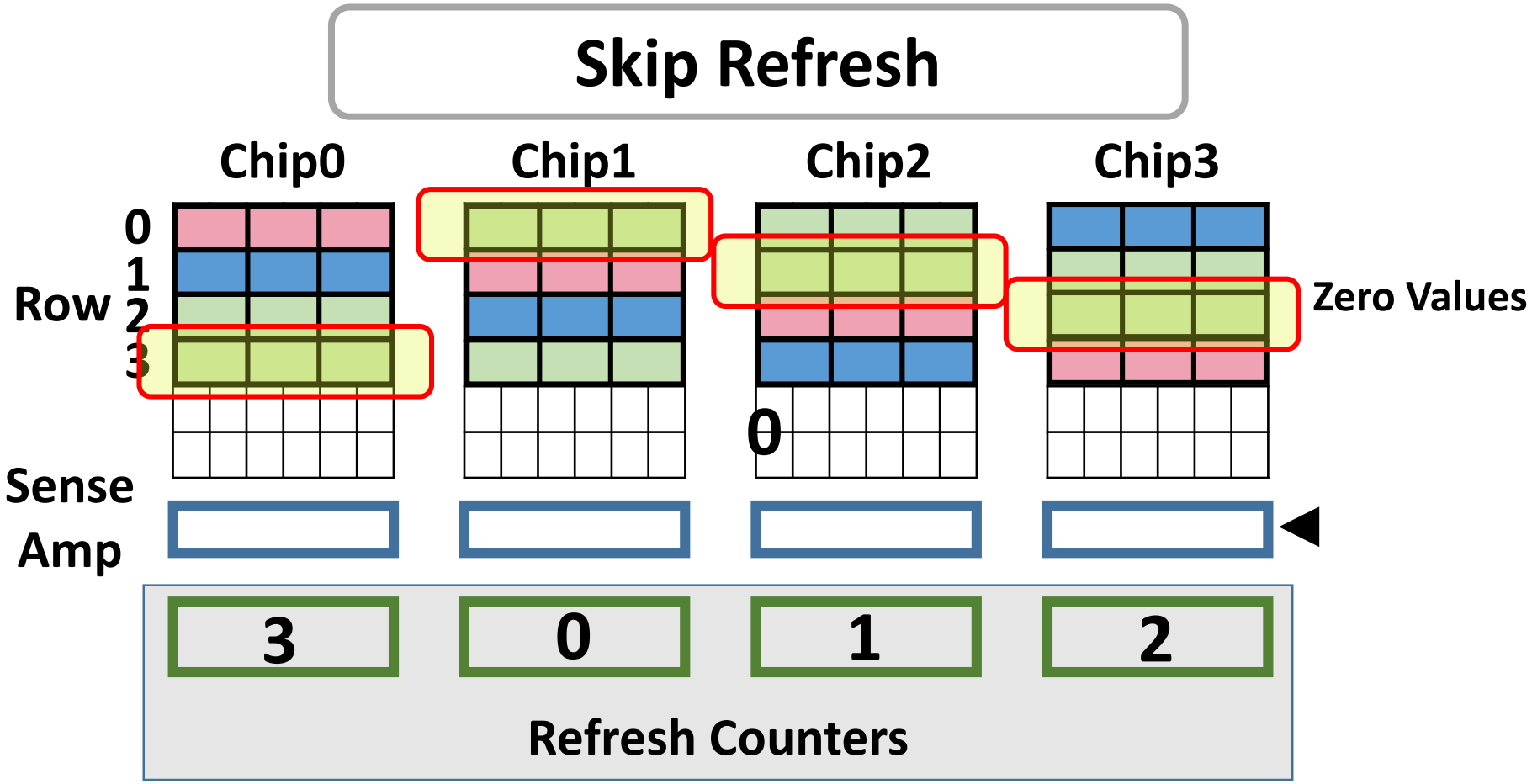
$$RefreshRow = \left((initRow + n) mod\ numChip\right) + \left(\frac{n}{numChip}\right) * numChip$$

Refresh Counter Modification



$$RefreshRow = \big((initRow + n) mod\ numChip\big) + \left(\frac{n}{numChip}\right) * numChip$$

Refresh Counter Modification



$$RefreshRow = \big((initRow + n) mod\ numChip\big) + \left(\frac{n}{numChip}\right) * numChip$$

Overhead of Zero-Refresh Module

- Performance Overhead: 1~2 cycle
  - Encoded BDI : 1~2 cycle
  - Bit plane stage: negligible (wire connection)
  - Data rotation stage: negligible (wire connection)

- Energy Overhead: 15pJ
  - EBDI: 15 pJ per operation

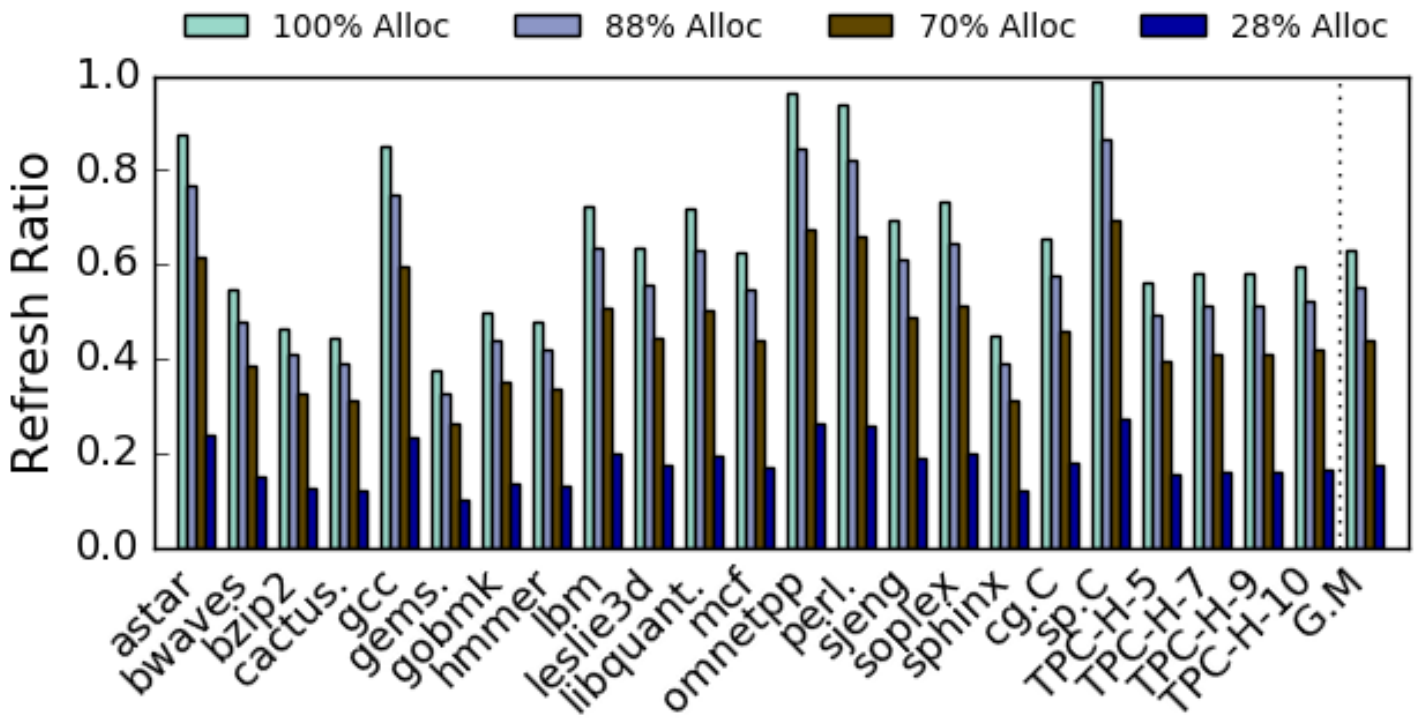
Simulation Configuration

- McSimA+ for core architecture
- GEMS for cache coherence
- DRAMSIM2 for memory

CPU Processor	OoO x86 ISA, 4GHz, 4 cores
CPU L1 Cache	32KB I cache, 32KB D cache 64B cacheline, 8 ways
CPU L2 Cache	2MB, 64B cacheline, 32 ways
Memory Configuration	32GB capacity, 8 chips, 8 banks, 4KB row buffer

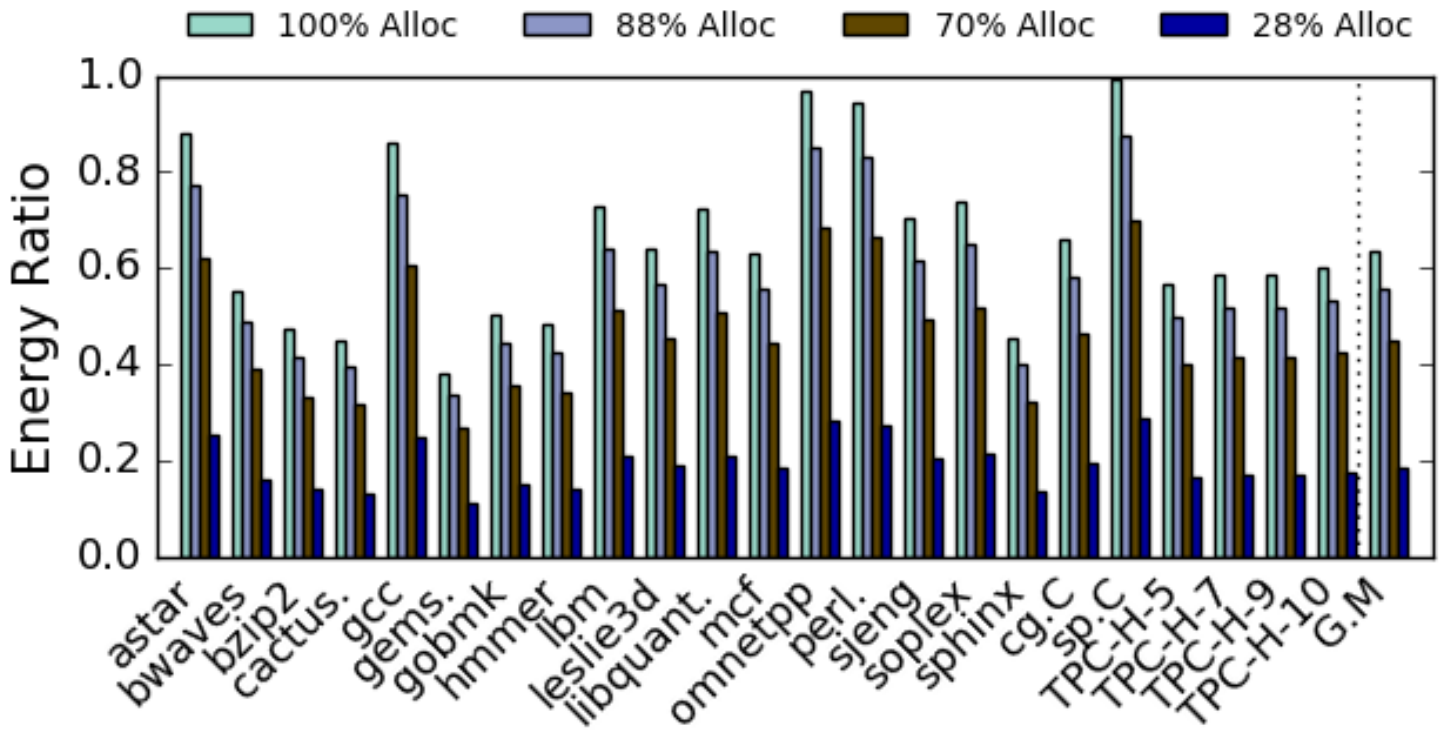
Refresh Reduction

- Zero-Refresh reduces 37.2% of refreshes
  - Alibaba: 44.7%
  - Google: 55.9%
  - Bitbrain: 83.0%



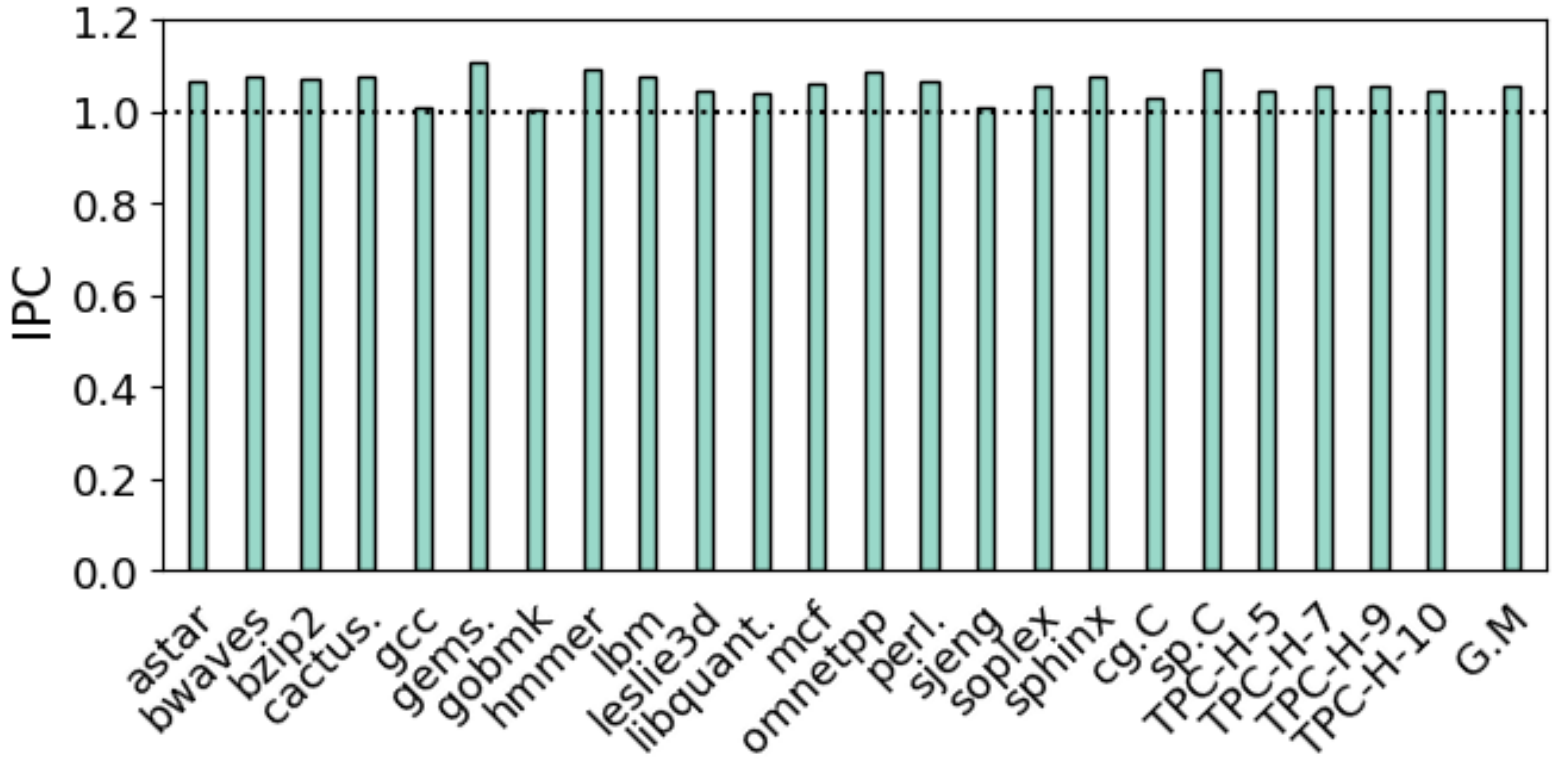
Energy Saving

- Reduces 37.13% of energy consumption
  - Zero-Refresh module: 15pJ per access
  - Leakage power: 2.71mW → Please refer to paper



Performance

- Zero-Refresh improves 4.4% of IPC



Conclusion

- ✓ We present a charge-aware refresh reduction
  - ✓ OS transparent: early-zeroing recommended
  - ✓ HW based value transformation

- ✓ More on paper!
  - ✓ Zero-Refresh Optimizations
  - ✓ Nature of DRAM
    - ✓ True-cell and anti-cell
    - ✓ DRAM Burst Mode
  - ✓ Evaluation and analysis
    - ✓ Row buffer size sensitivity
    - ✓ Compare to access aware refresh