DICE: Compressing DRAM Caches for Bandwidth and Capacity

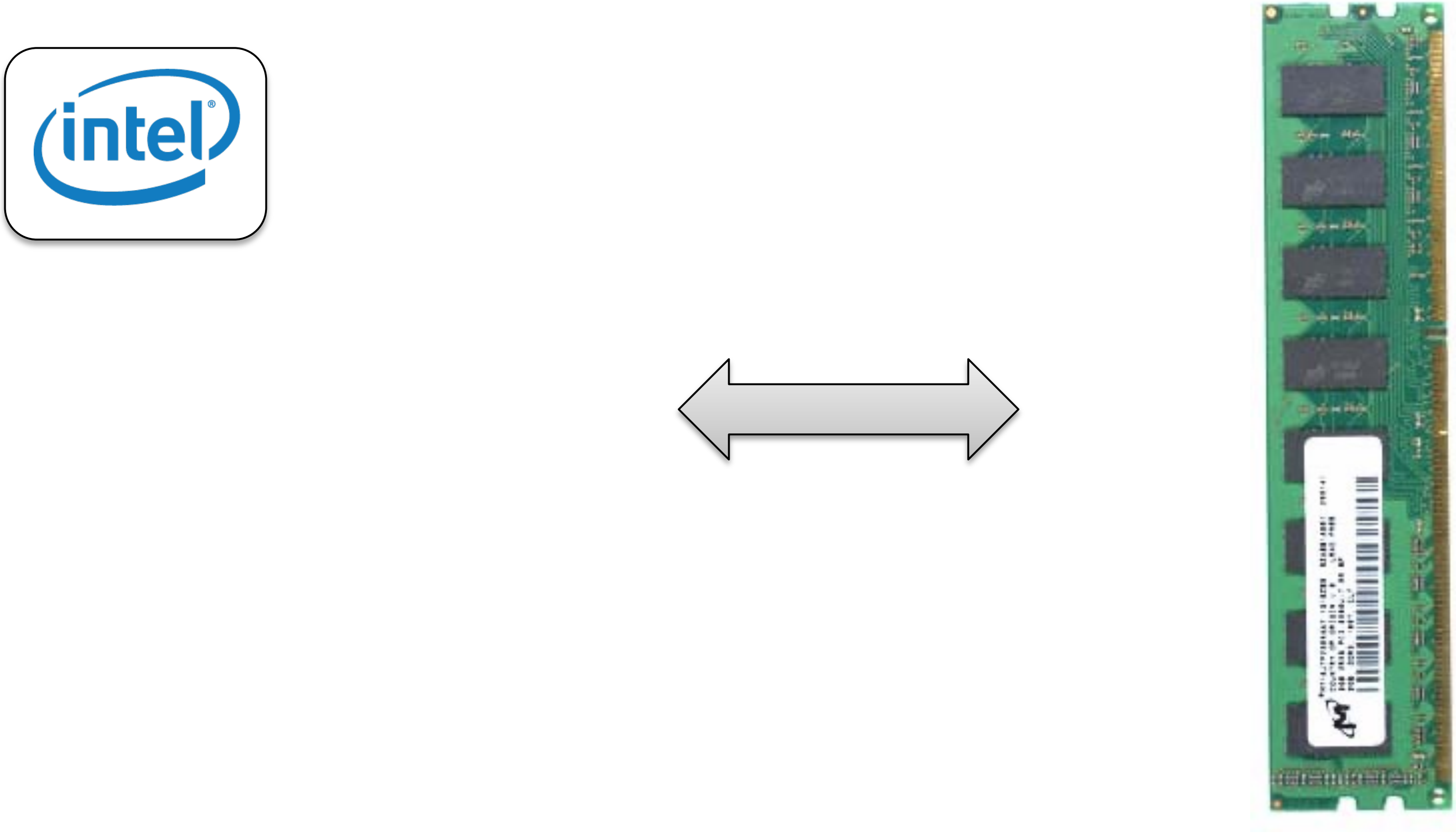
Vinson Young

Prashant Nair

Moinuddin Qureshi

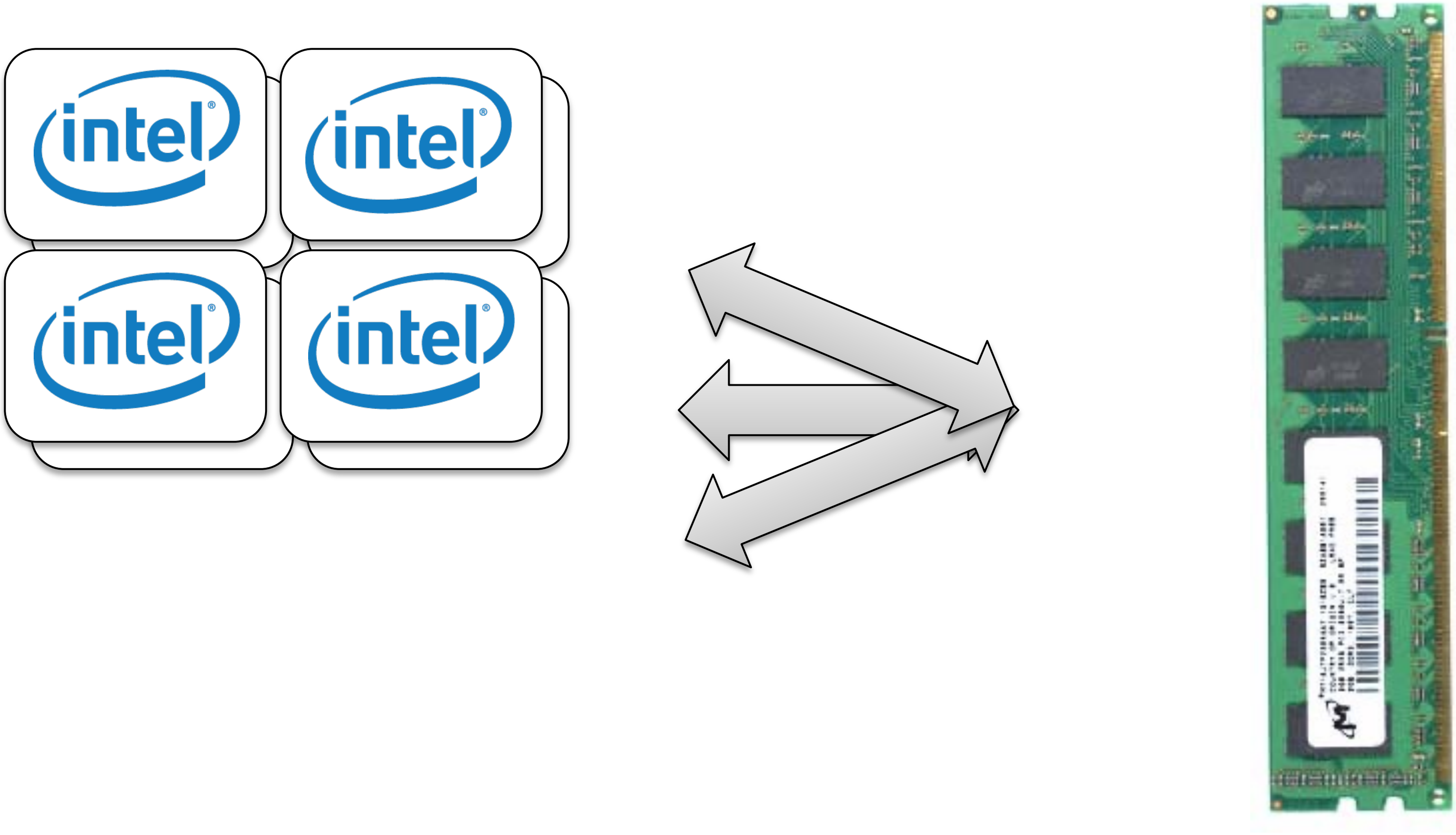
Georgia Institute of Technology

MOORE’S LAW HITS BANDWIDTH WALL



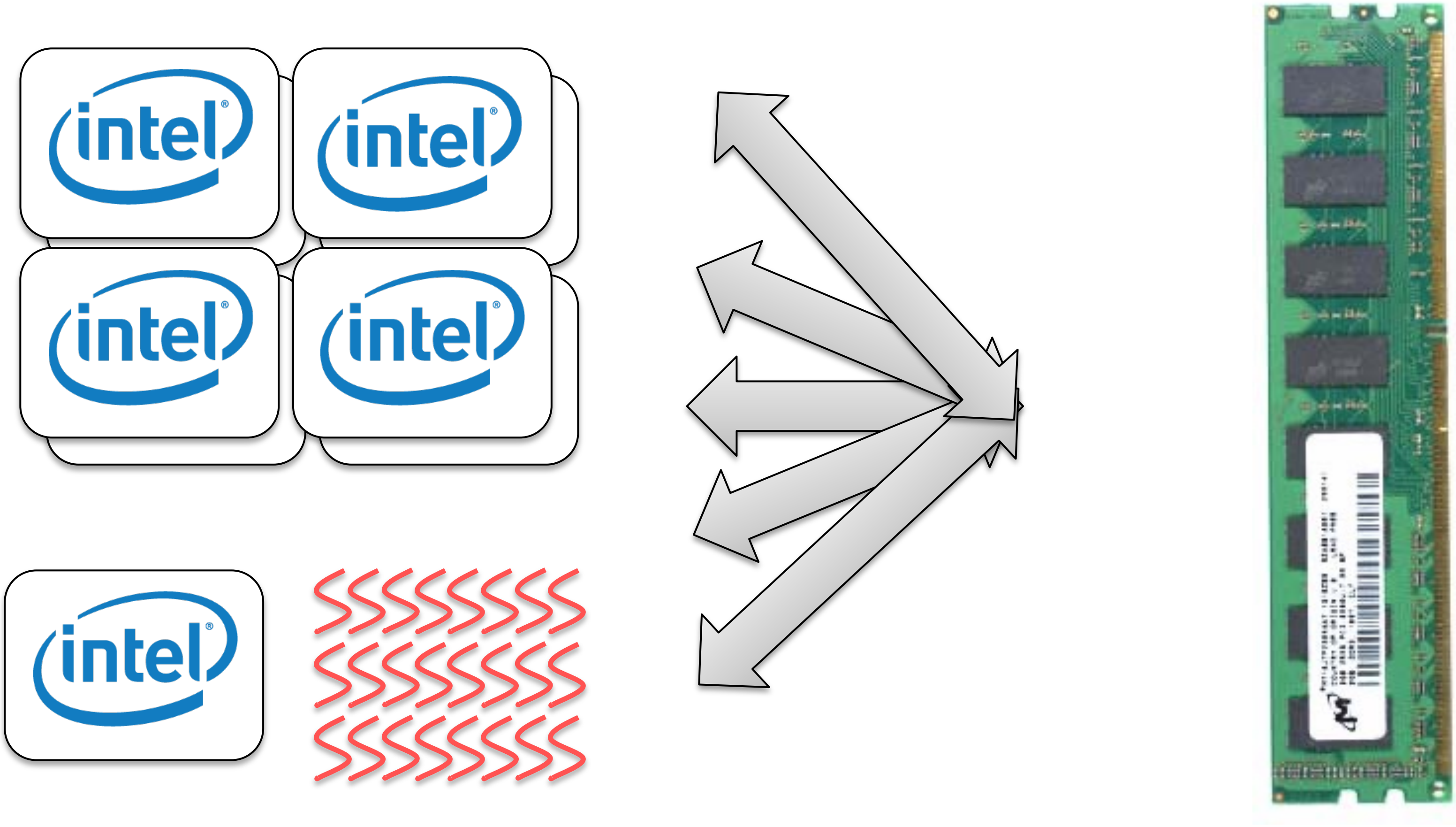
Moore’s scaling encounters Bandwidth Wall

MOORE'S LAW HITS BANDWIDTH WALL



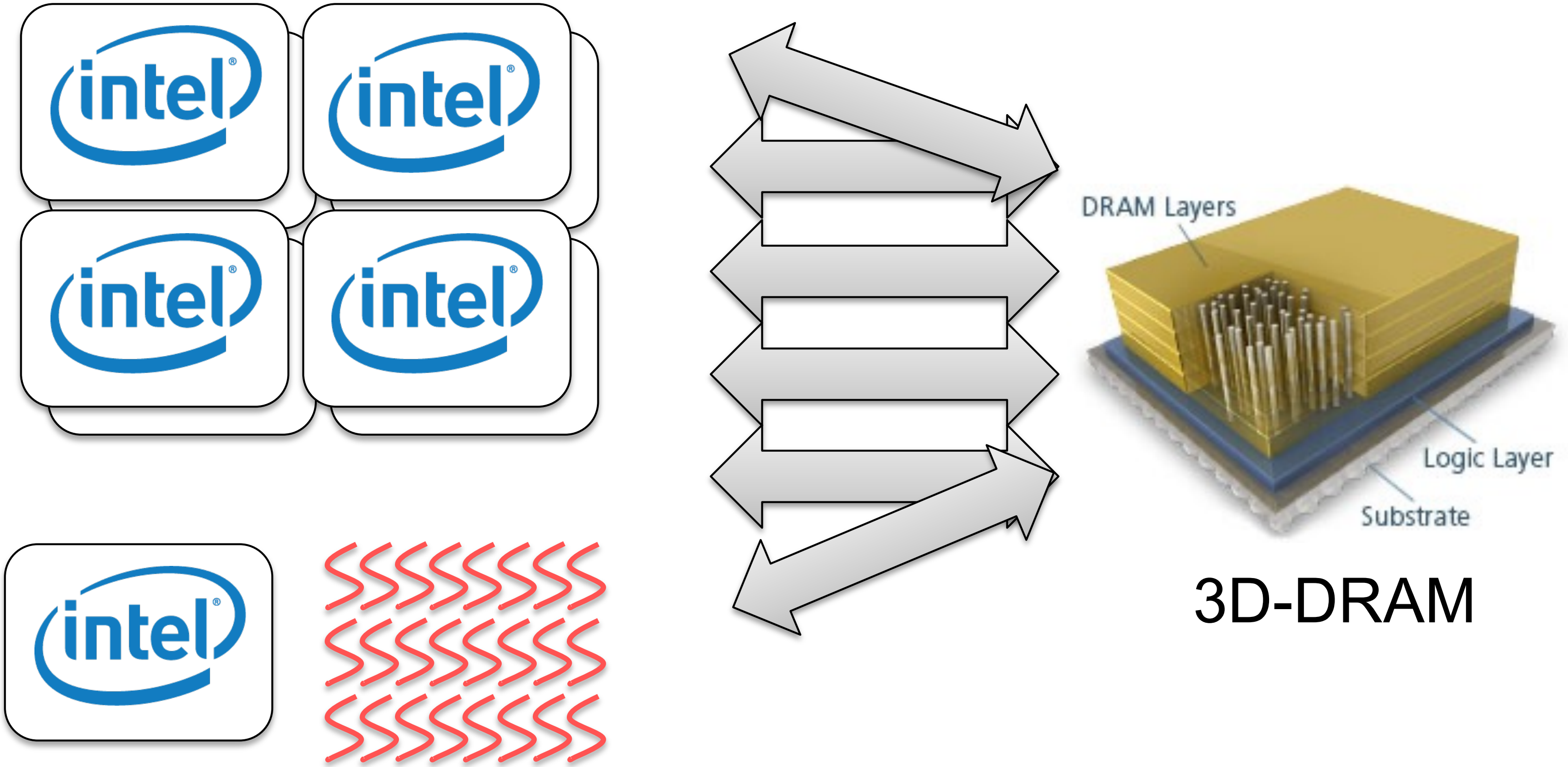
Moore's scaling encounters Bandwidth Wall

MOORE’S LAW HITS BANDWIDTH WALL



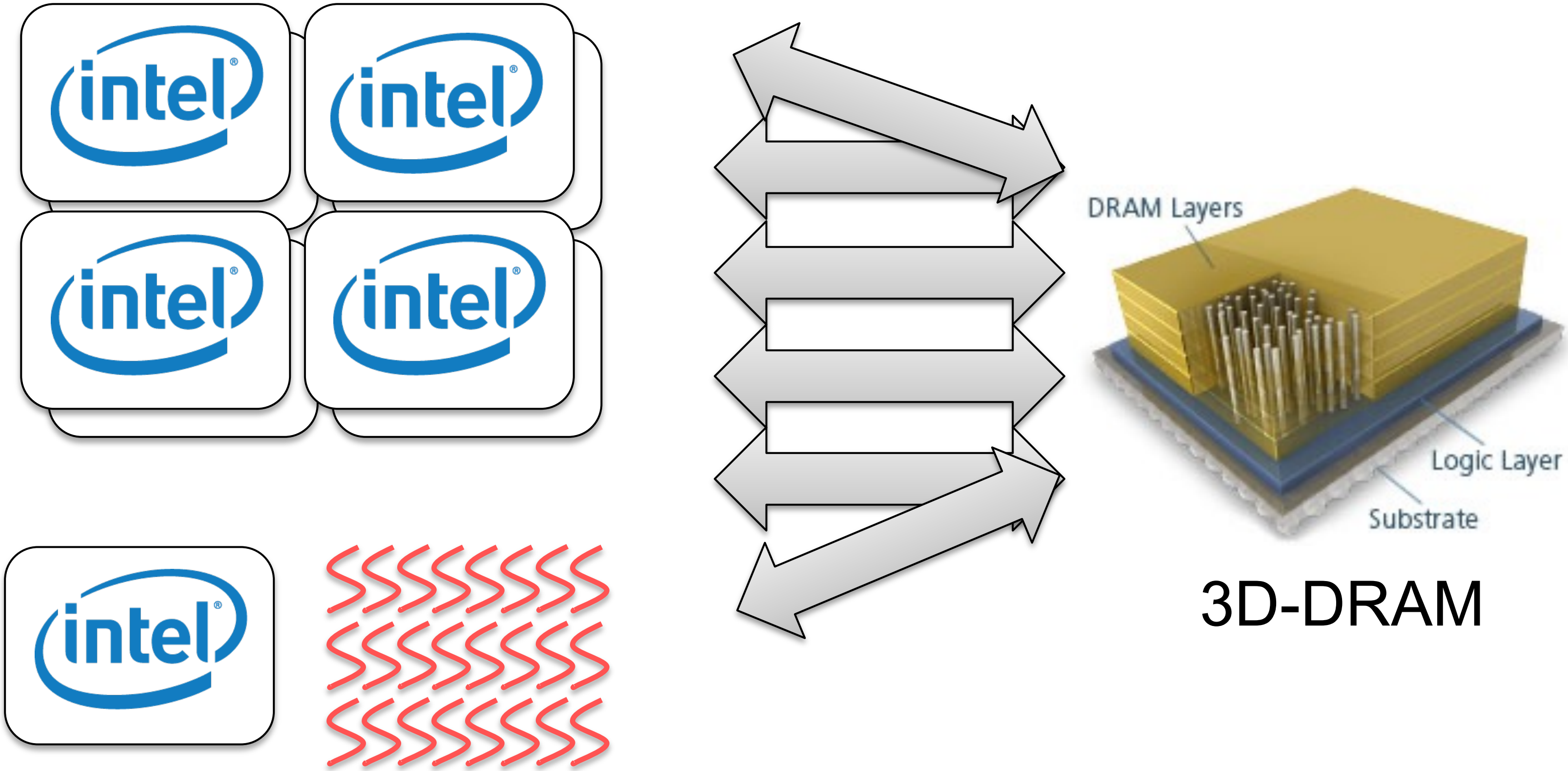
Moore’s scaling encounters Bandwidth Wall

3D-DRAM MITIGATES BANDWIDTH WALL



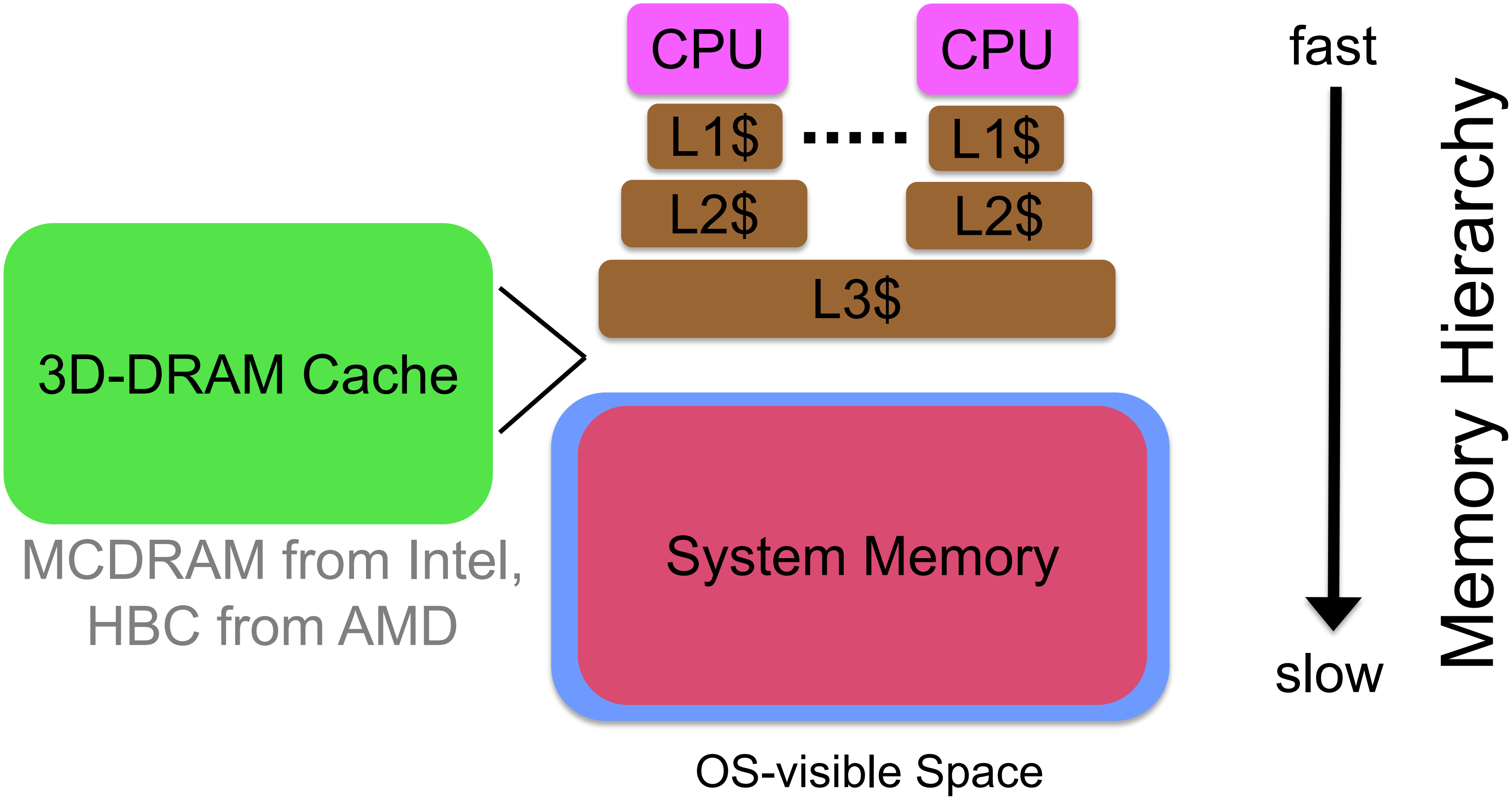
Hybrid Memory Cube (HMC) from Micron,
High Bandwidth Memory (HBM) from Samsung

3D-DRAM MITIGATES BANDWIDTH WALL



3D-DRAM improves bandwidth, but does not have capacity to replace conventional DIMM memory

3D-DRAM AS A CACHE (3D-DRAM CACHE)

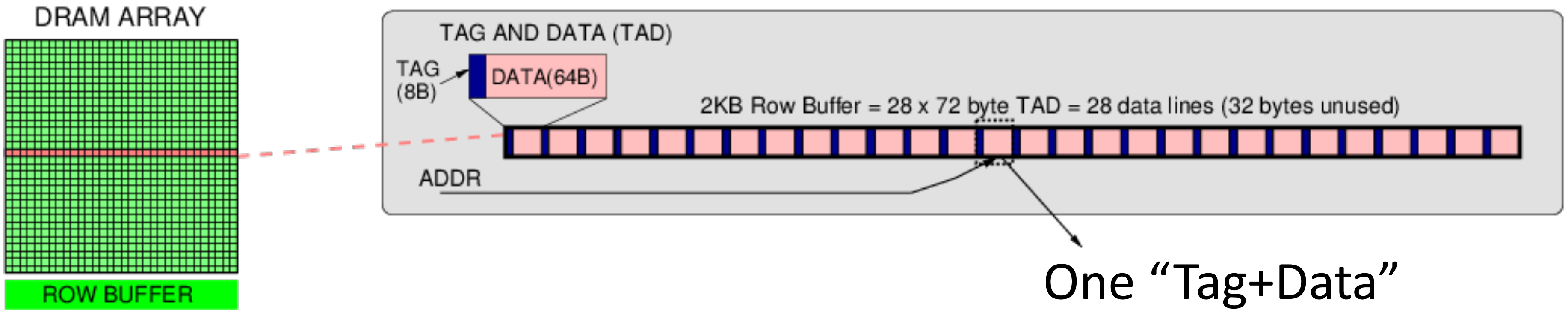


MCDRAM from Intel,
HBC from AMD

Architecting 3D-DRAM as a cache can improve memory bandwidth (and avoid OS/software change)

PRACTICAL 3D-DRAM CACHE: ALLOY CACHE

Tags “part-of-line” ➔ Alloy **Tag+Data** ➔ Avoid **Tag Serialization**

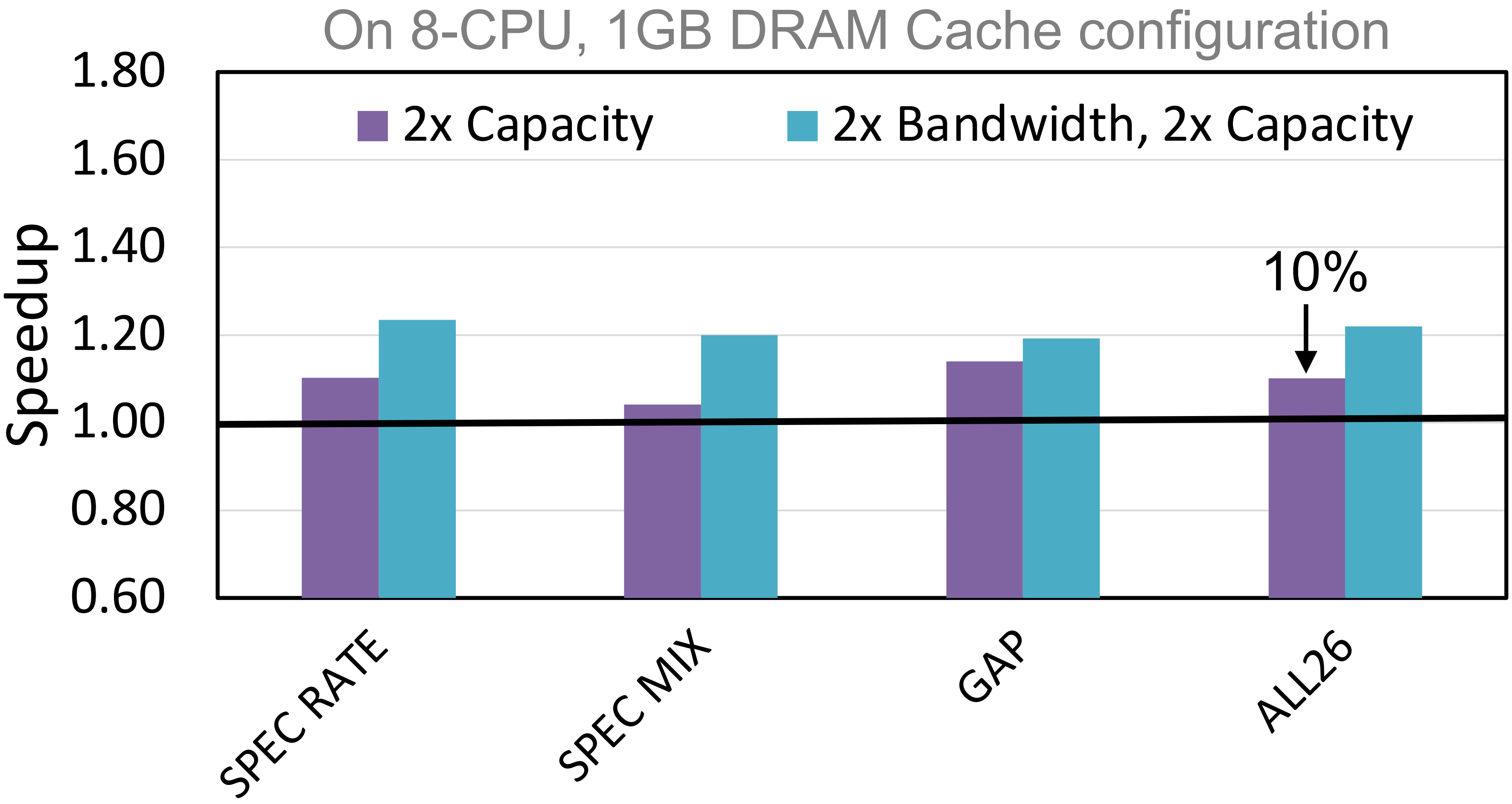

DRAM ARRAY
ROW BUFFER
TAG AND DATA (TAD)
TAG
(8B)
DATA(64B)
2KB Row Buffer = 28 x 72 byte TAD = 28 data lines (32 bytes unused)
ADDR

One “Tag+Data”

Similar to DRAM Cache in KNL: Direct-mapped, Tags in ECC

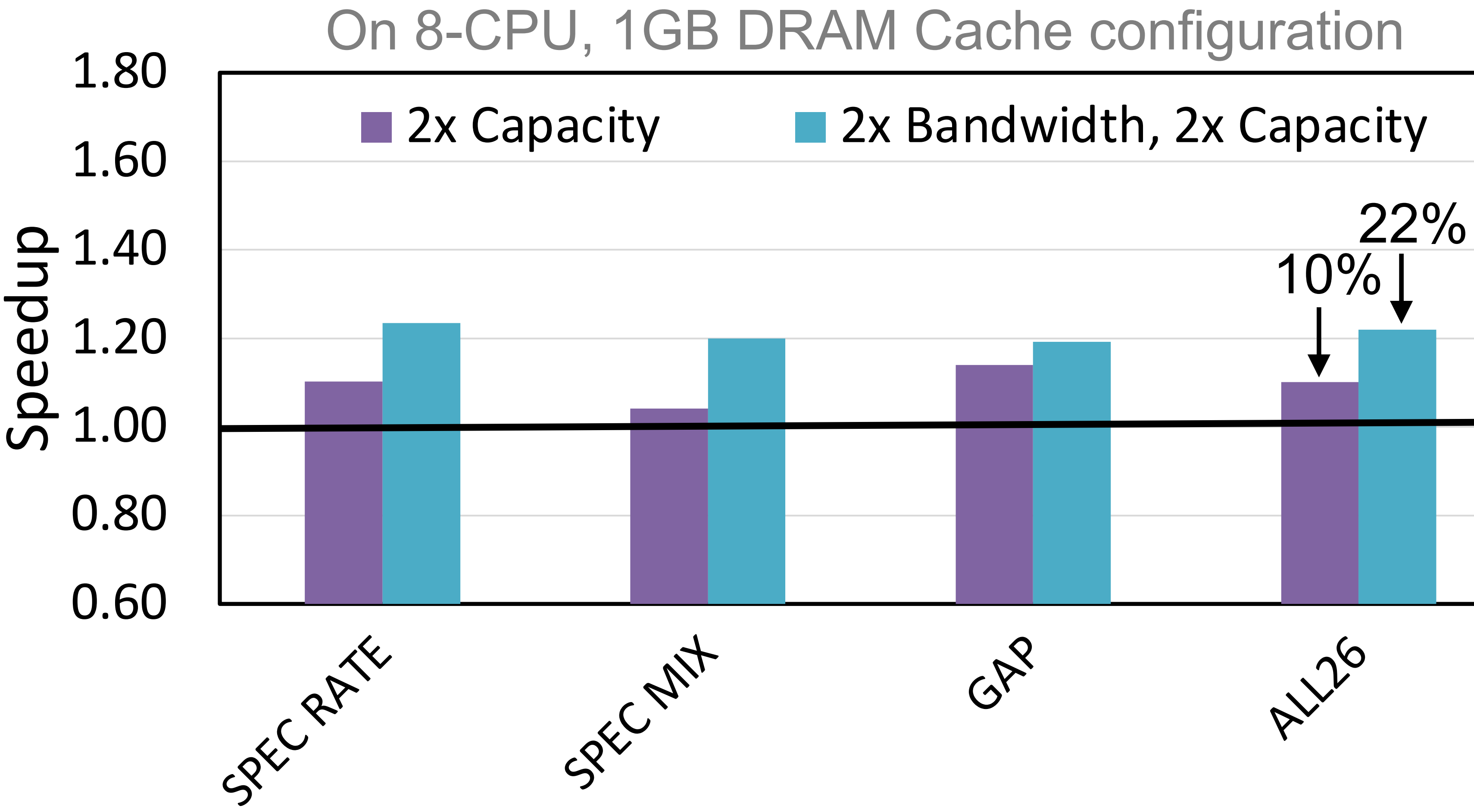
Practical DRAM cache: low latency and bandwidth-efficient

3D-DRAM CACHE BANDWIDTH IS IMPORTANT



2x-capacity cache improves performance by 10%.
And, additional 2x bandwidth increases speedup to 22%.
Improving both bandwidth and capacity is valuable.

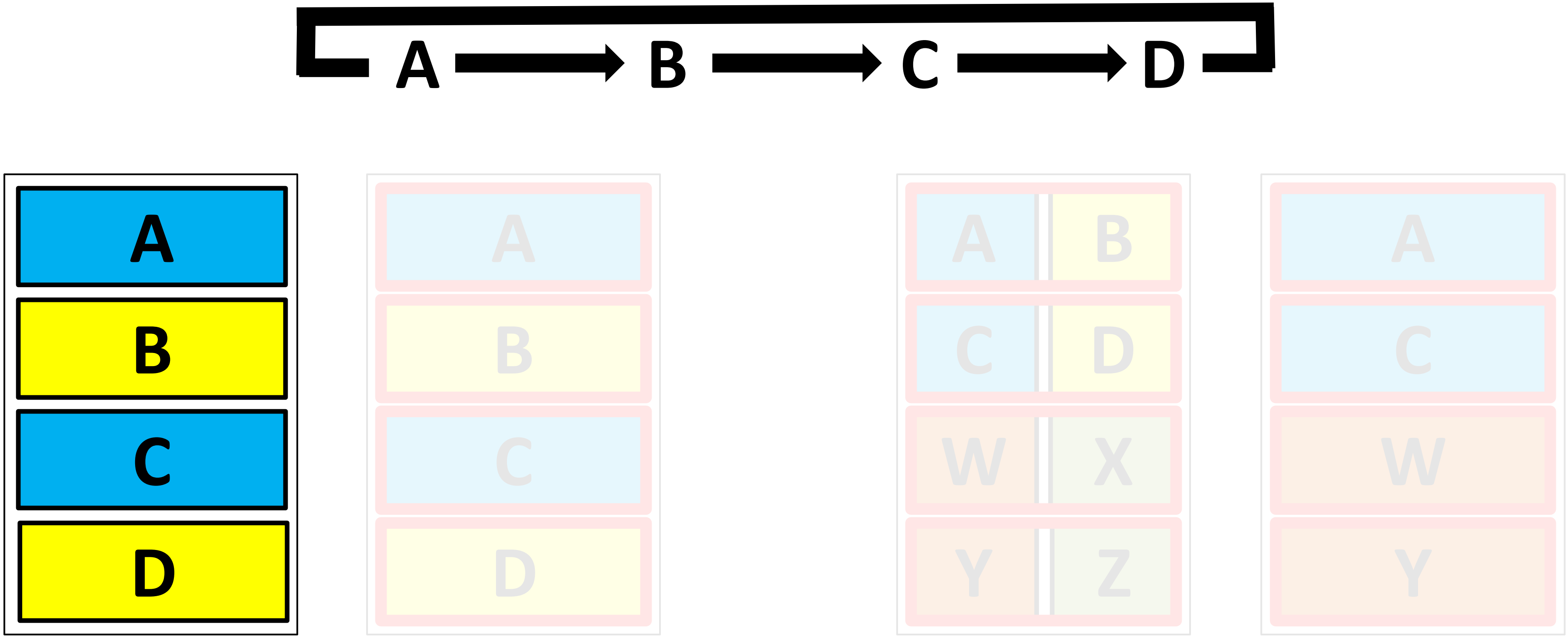
3D-DRAM CACHE BANDWIDTH IS IMPORTANT



2x-capacity cache improves performance by 10%.
And, additional 2x bandwidth increases speedup to 22%.
Improving both bandwidth and capacity is valuable.

INTRODUCTION: DRAM CACHE

Baseline: Direct-Mapped, One Data Block in an access



Baseline

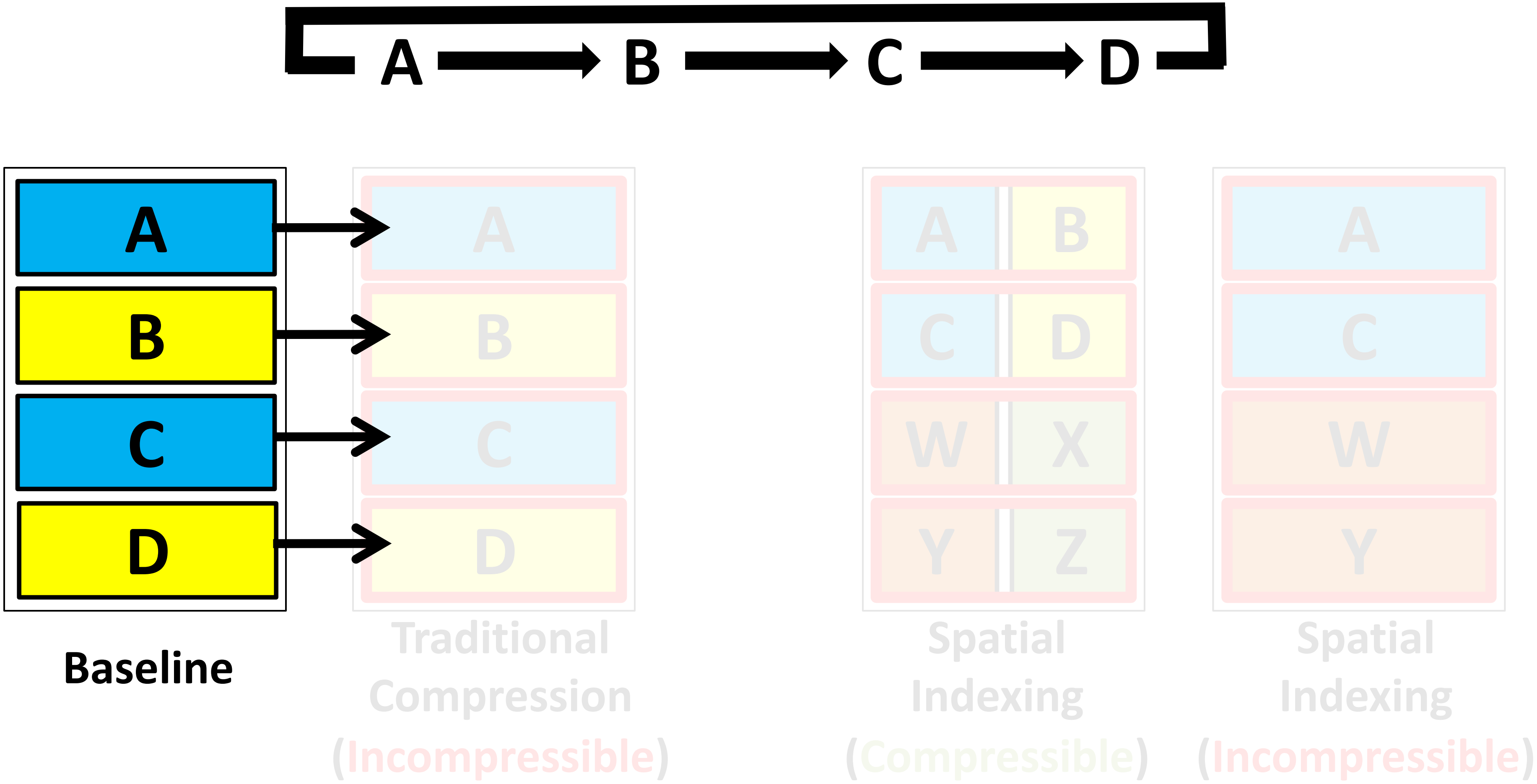
Traditional Compression (Incompressible)

Spatial Indexing (Compressible)

Spatial Indexing (Incompressible)

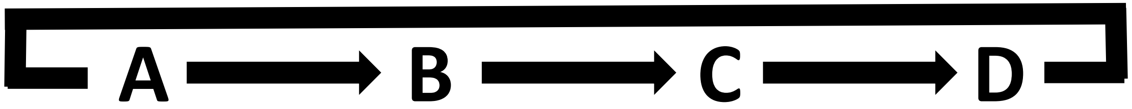
INTRODUCTION: DRAM CACHE

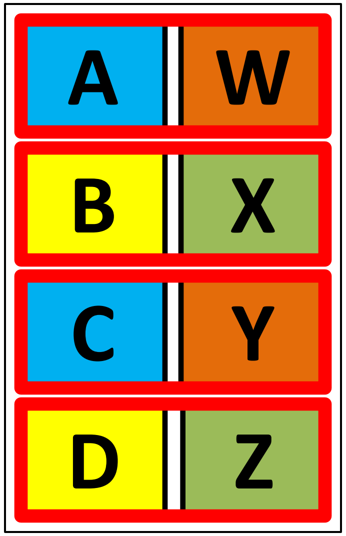
Baseline: Direct-Mapped, One Data Block in an access


A
B
C
D
A
B
C
D
A
B
C
D
A
B
C
D
W
X
Y
Z
A
C
W
Y
Baseline
Traditional Compression (Incompressible)
Spatial Indexing (Compressible)
Spatial Indexing (Incompressible)

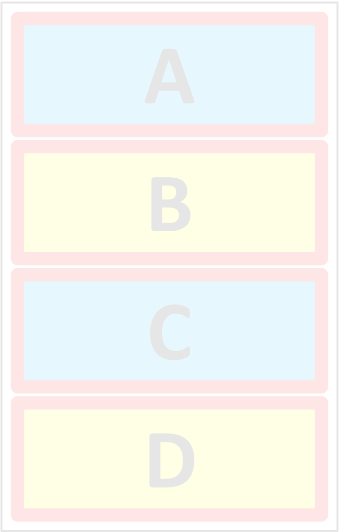
INTRODUCTION: COMPRESSED DRAM CACHE

Compression: Adds capacity

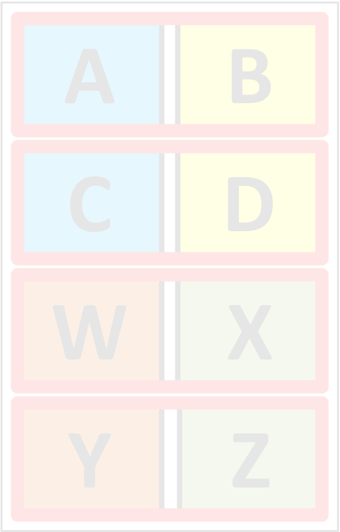




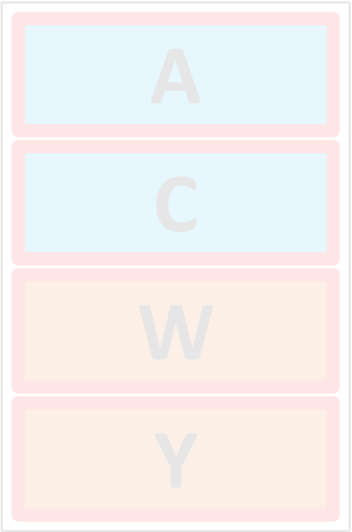
Traditional Compression (Compressible)



Traditional Compression (Incompressible)



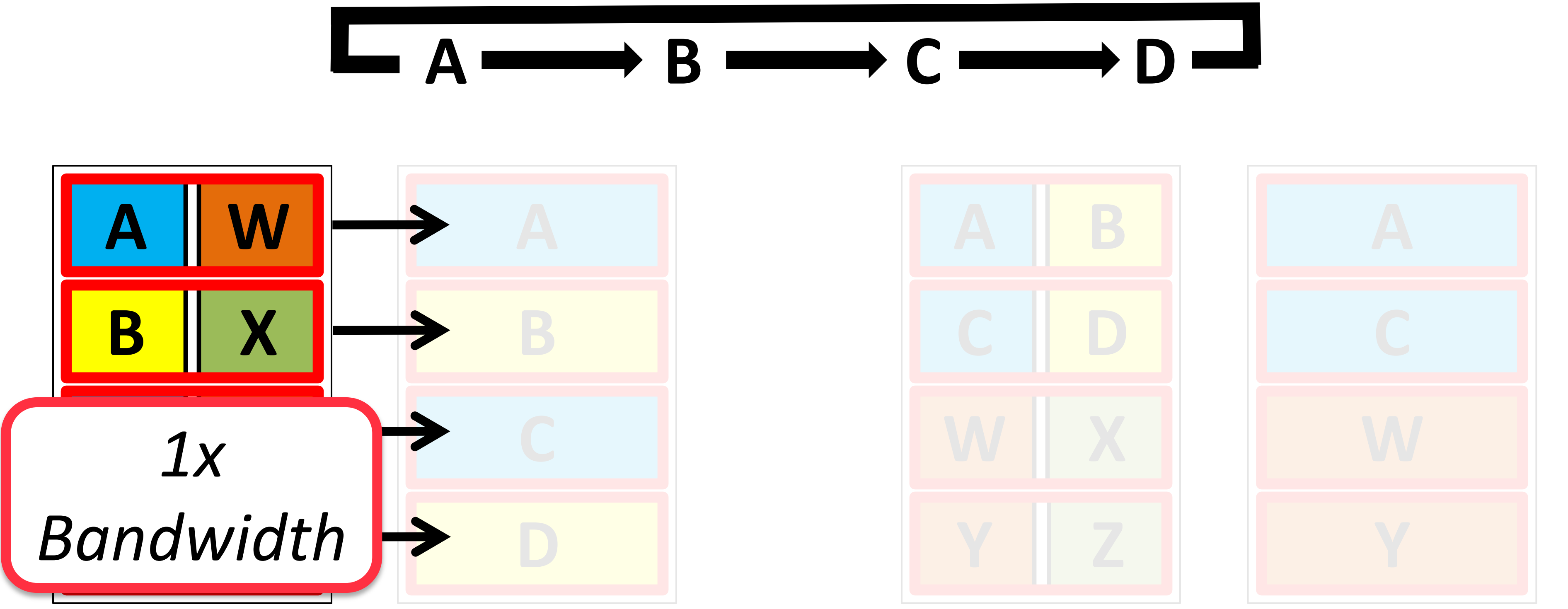
Spatial Indexing (Compressible)



Spatial Indexing (Incompressible)

INTRODUCTION: COMPRESSED DRAM CACHE

Compression: Adds capacity



1x Bandwidth

Traditional Compression (Compressible)

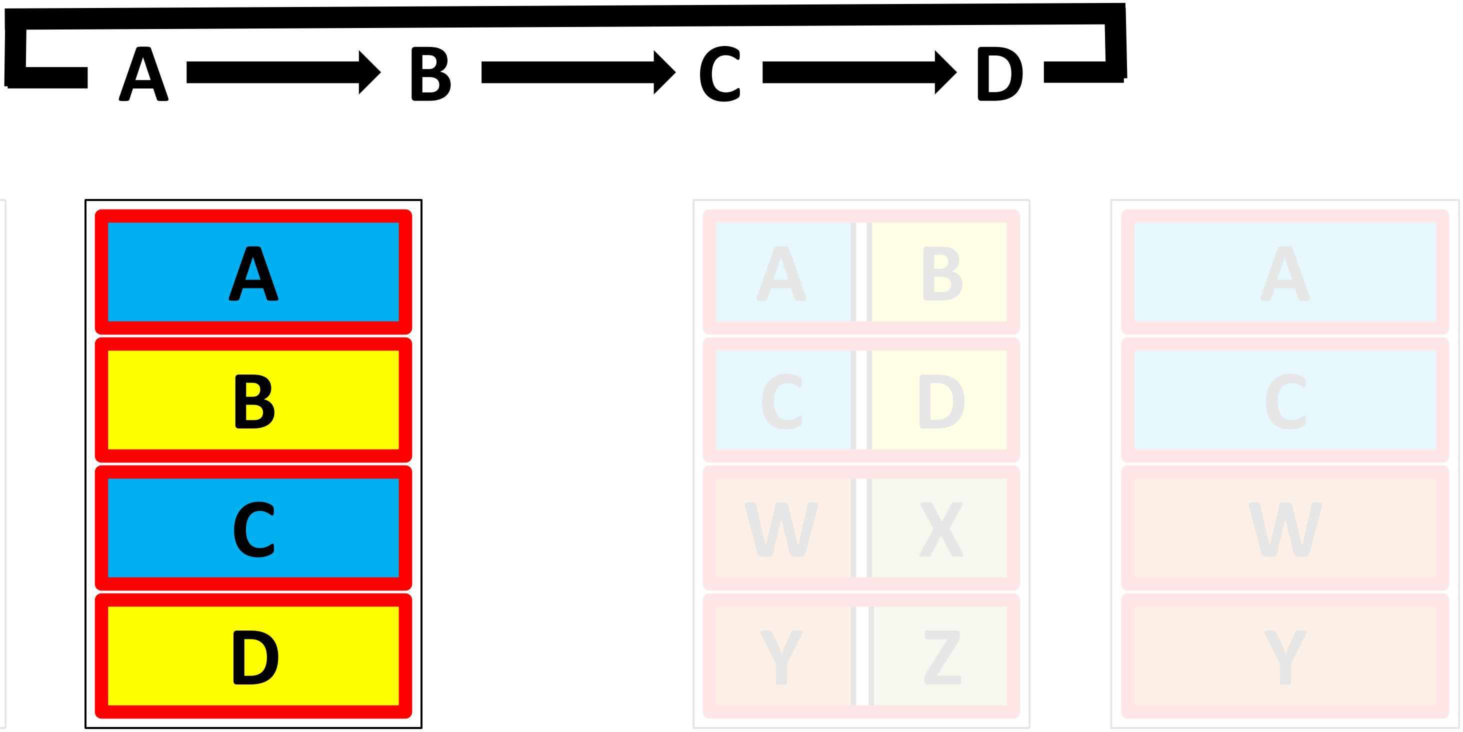
Traditional Compression (Incompressible)

Spatial Indexing (Compressible)

Spatial Indexing (Incompressible)

INTRODUCTION: COMPRESSED DRAM CACHE

Compression: Adds capacity



Traditional Compression (Compressible)

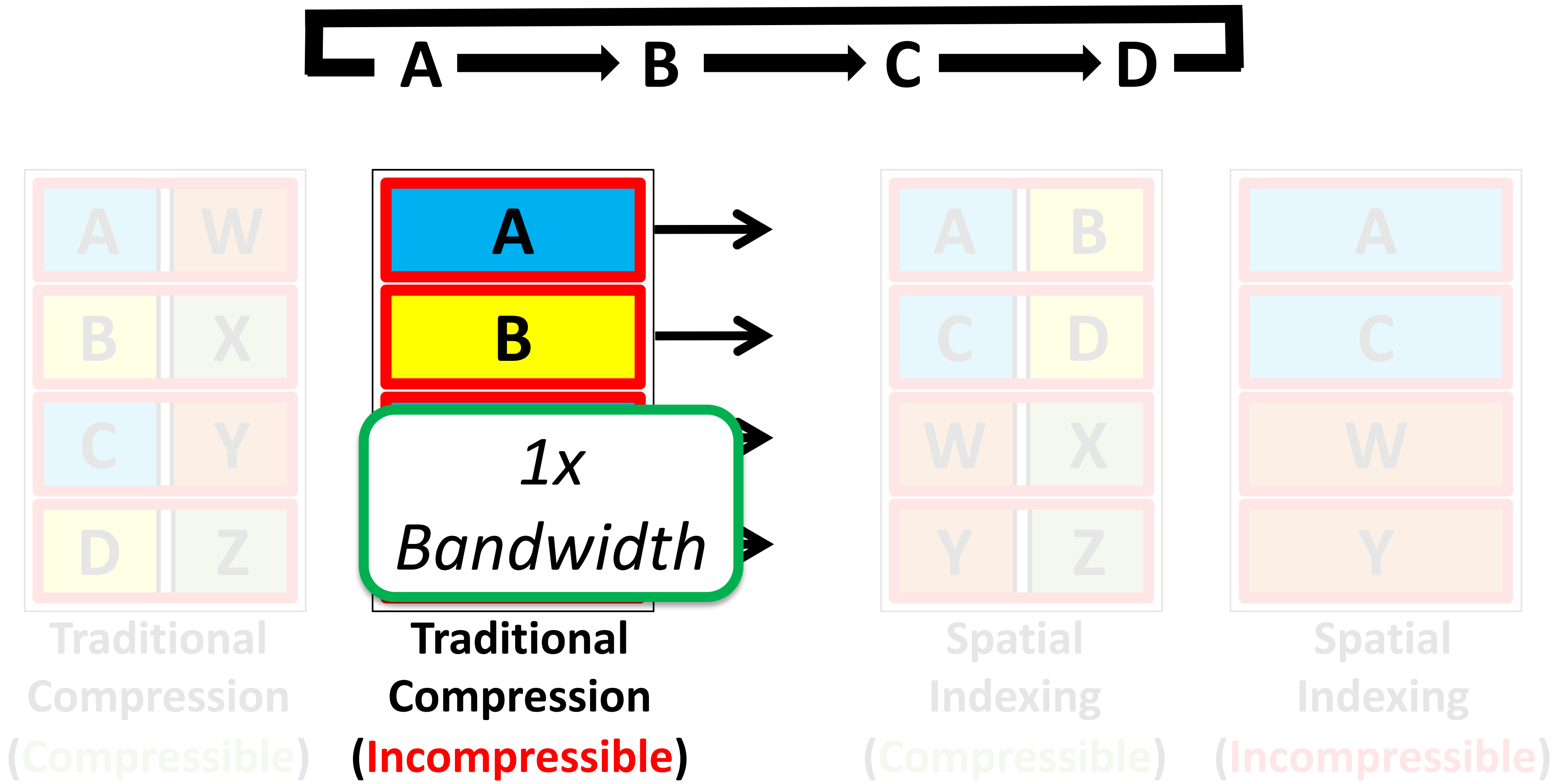
Traditional Compression (Incompressible)

Spatial Indexing (Compressible)

Spatial Indexing (Incompressible)

INTRODUCTION: COMPRESSED DRAM CACHE

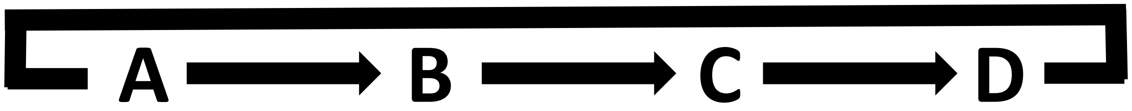
Compression: Adds capacity

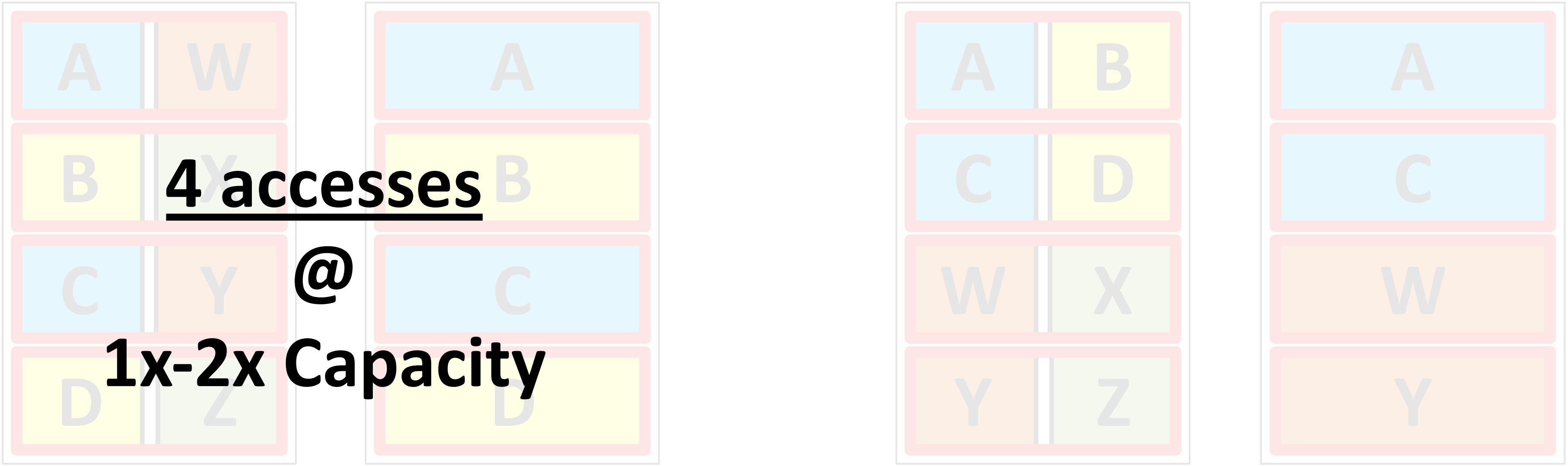

A
B
C
D
A
W
B
X
C
Y
D
Z
A
B
A
B
C
D
W
X
Y
Z
A
C
W
Y
Traditional Compression (Compressible)
Traditional Compression (Incompressible)
Spatial Indexing (Compressible)
Spatial Indexing (Incompressible)

1x Bandwidth

INTRODUCTION: COMPRESSED DRAM CACHE

Compression: Adds capacity





<u>4 accesses</u>

@

1x-2x Capacity

Traditional Compression (Compressible)

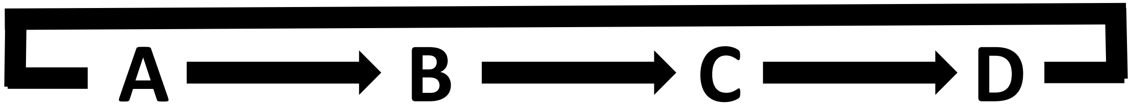
Traditional Compression (Incompressible)

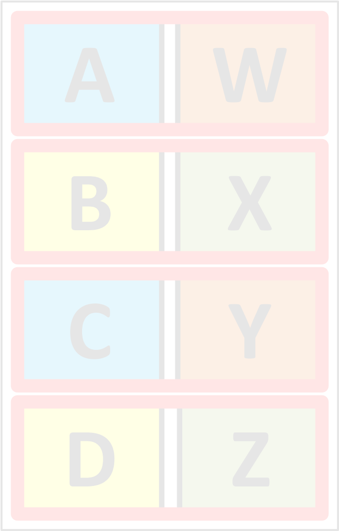
Spatial Indexing (Compressible)

Spatial Indexing (Incompressible)

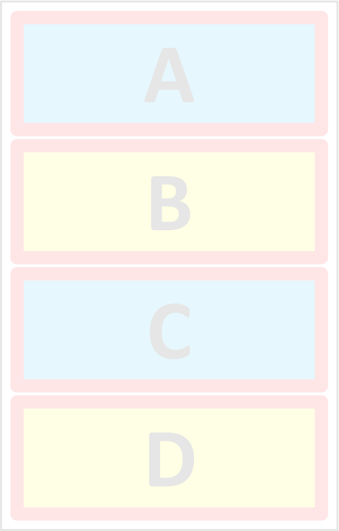
INTRODUCTION: COMPRESSED DRAM CACHE

Compression: Adds capacity, improve bandwidth?

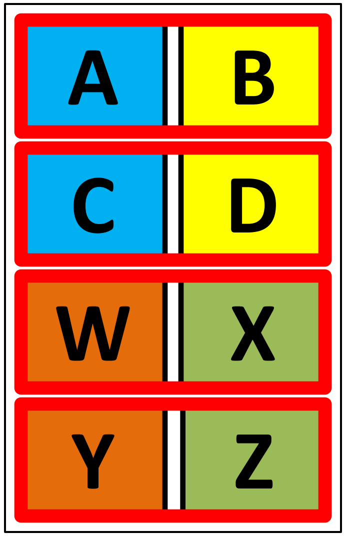




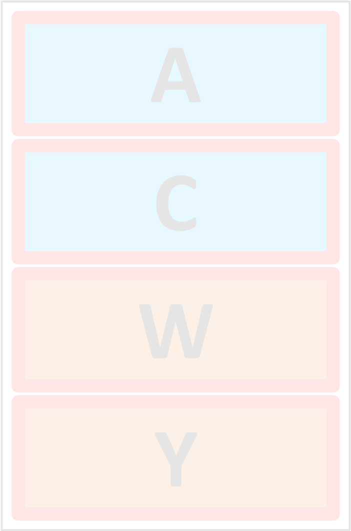
Traditional Compression (Compressible)



Traditional Compression (Incompressible)



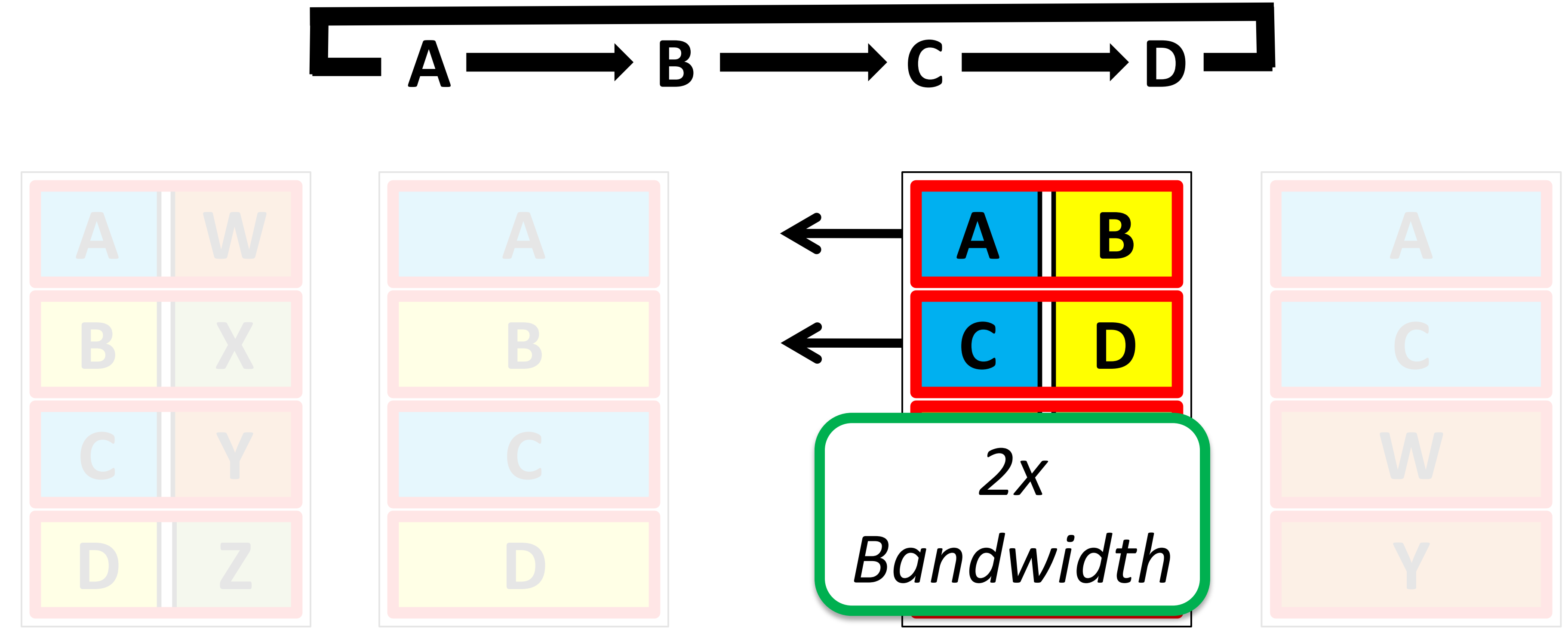
Spatial Indexing (Compressible)



Spatial Indexing (Incompressible)

INTRODUCTION: COMPRESSED DRAM CACHE

Compression: Adds capacity, improve bandwidth?



2x Bandwidth

Traditional Compression (Compressible)

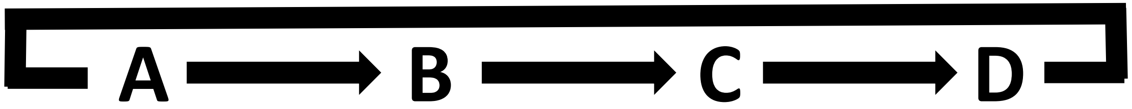
Traditional Compression (Incompressible)

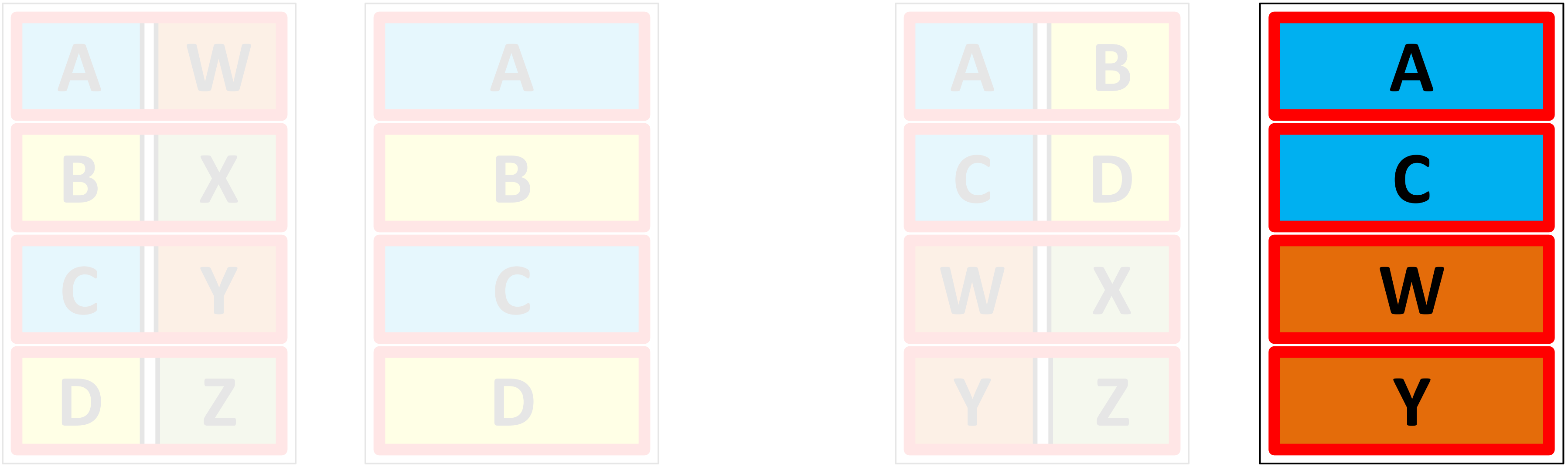
Spatial Indexing (Compressible)

Spatial Indexing (Incompressible)

INTRODUCTION: COMPRESSED DRAM CACHE

Compression: Adds capacity, improve bandwidth?

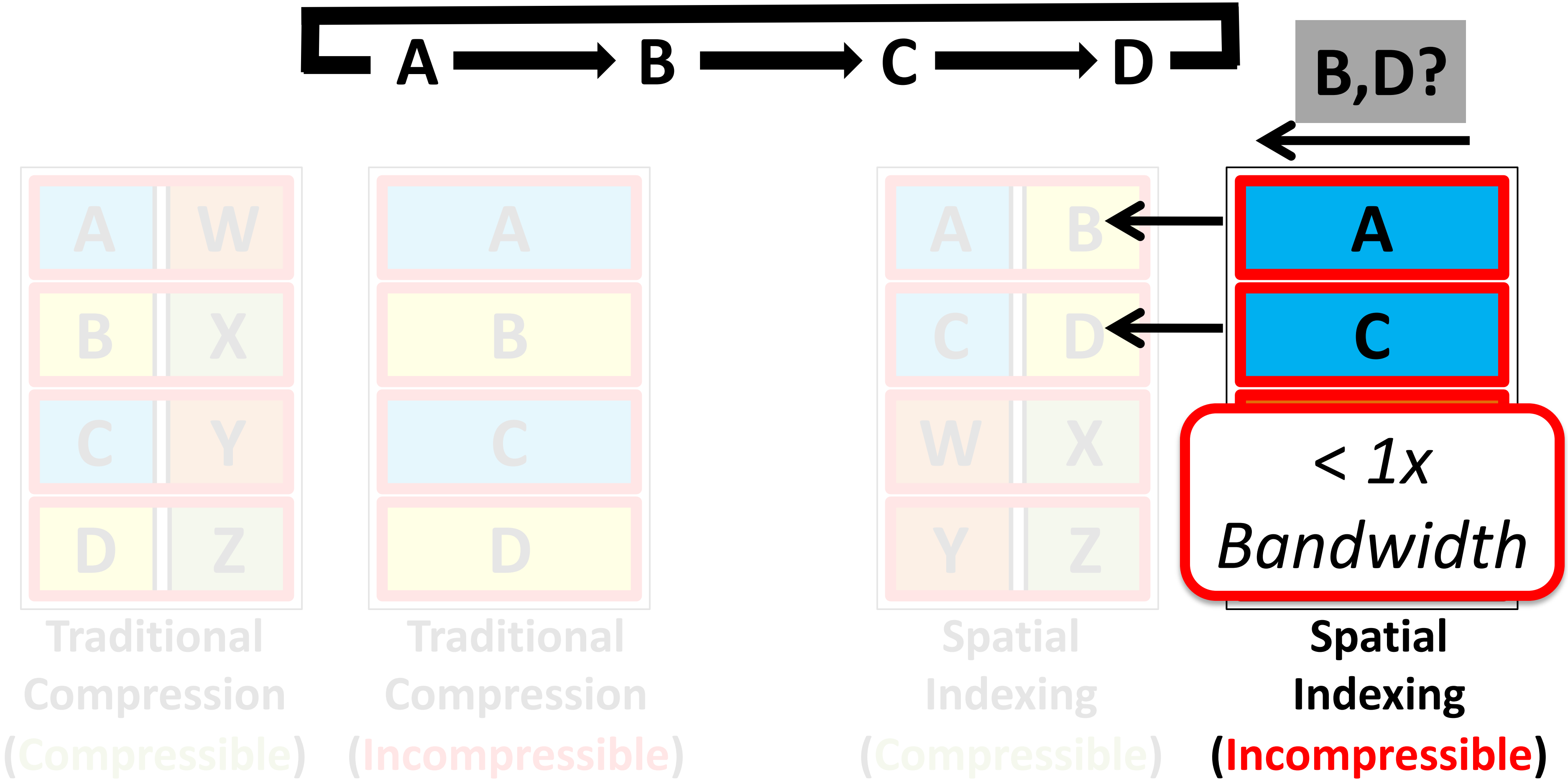




Traditional Compression (Compressible) | Traditional Compression (Incompressible) | Spatial Indexing (Compressible) | Spatial Indexing (Incompressible)

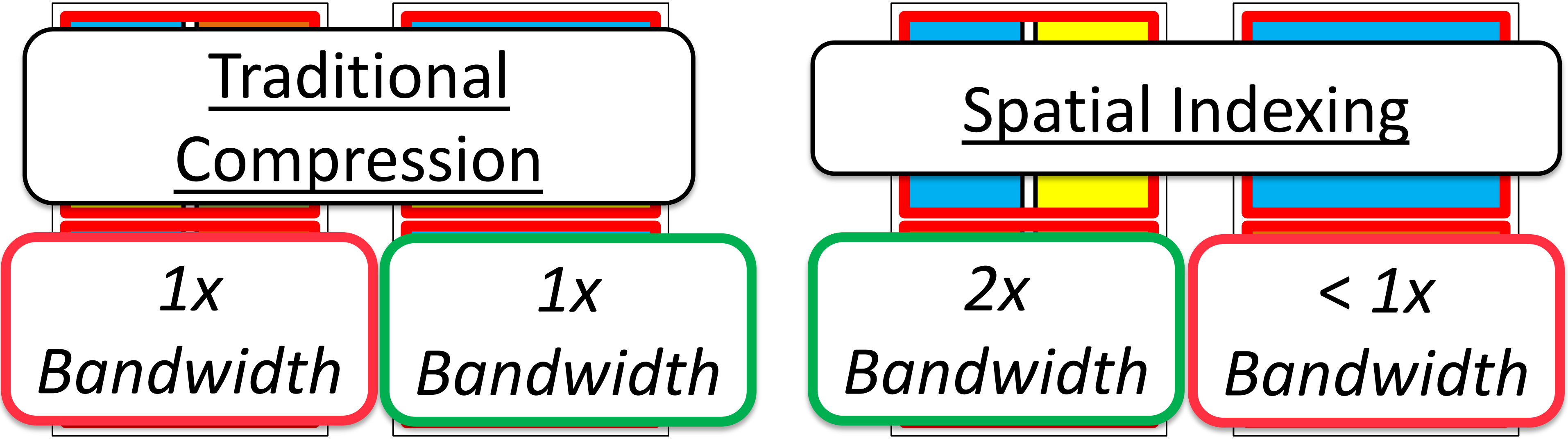
INTRODUCTION: COMPRESSED DRAM CACHE

Compression: Adds capacity, improve bandwidth?


A → B → C → D
B,D?
A
W
B
X
C
Y
D
Z
A
B
C
D
A
B
C
D
W
X
Y
Z
A
C
< 1x Bandwidth
Traditional Compression (Compressible)
Traditional Compression (Incompressible)
Spatial Indexing (Compressible)
Spatial Indexing (Incompressible)

INTRODUCTION: COMPRESSED DRAM CACHE

Compression: Adds capacity, improve bandwidth?



Traditional Compression

Spatial Indexing

1x Bandwidth

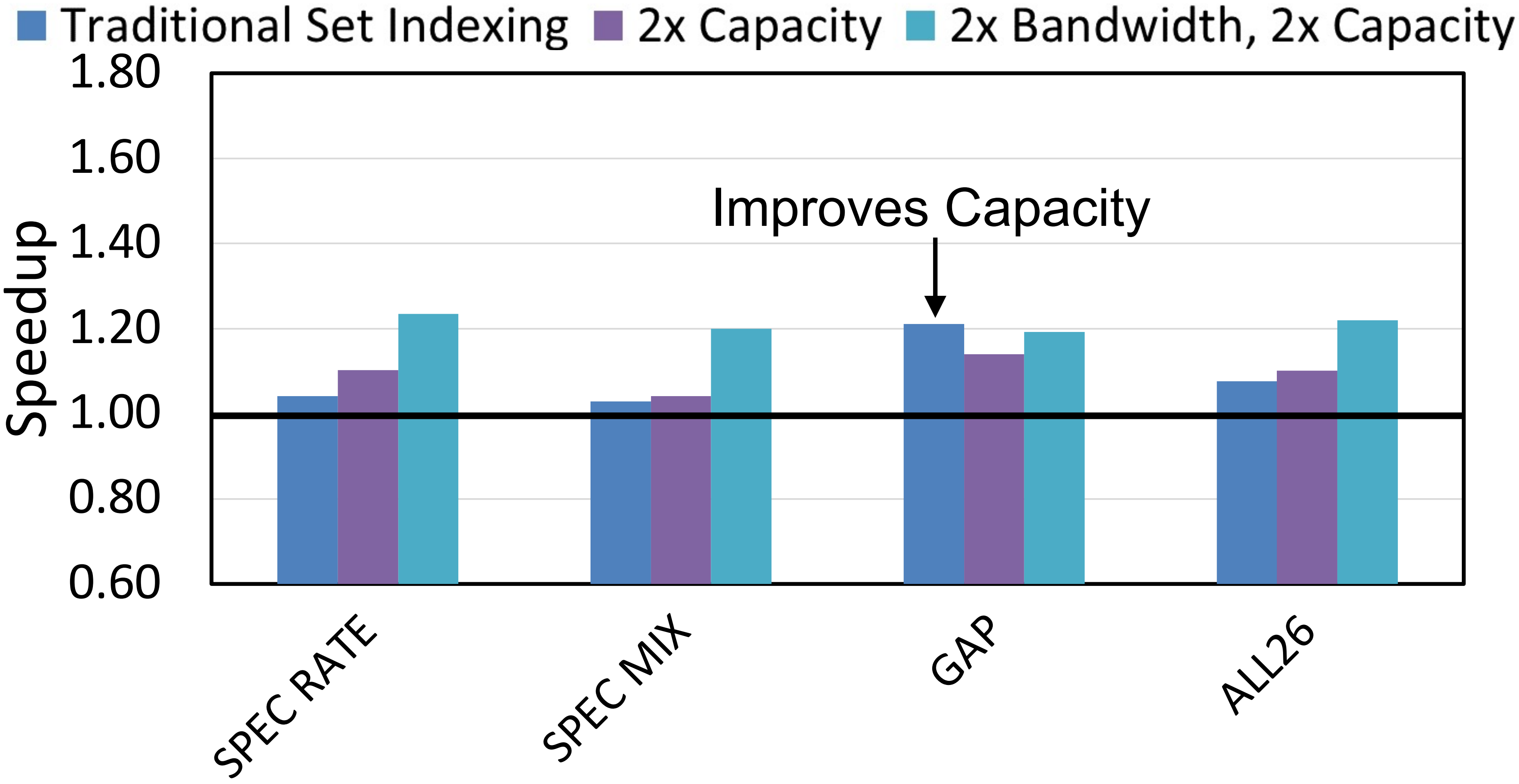
1x Bandwidth

2x Bandwidth

< 1x Bandwidth

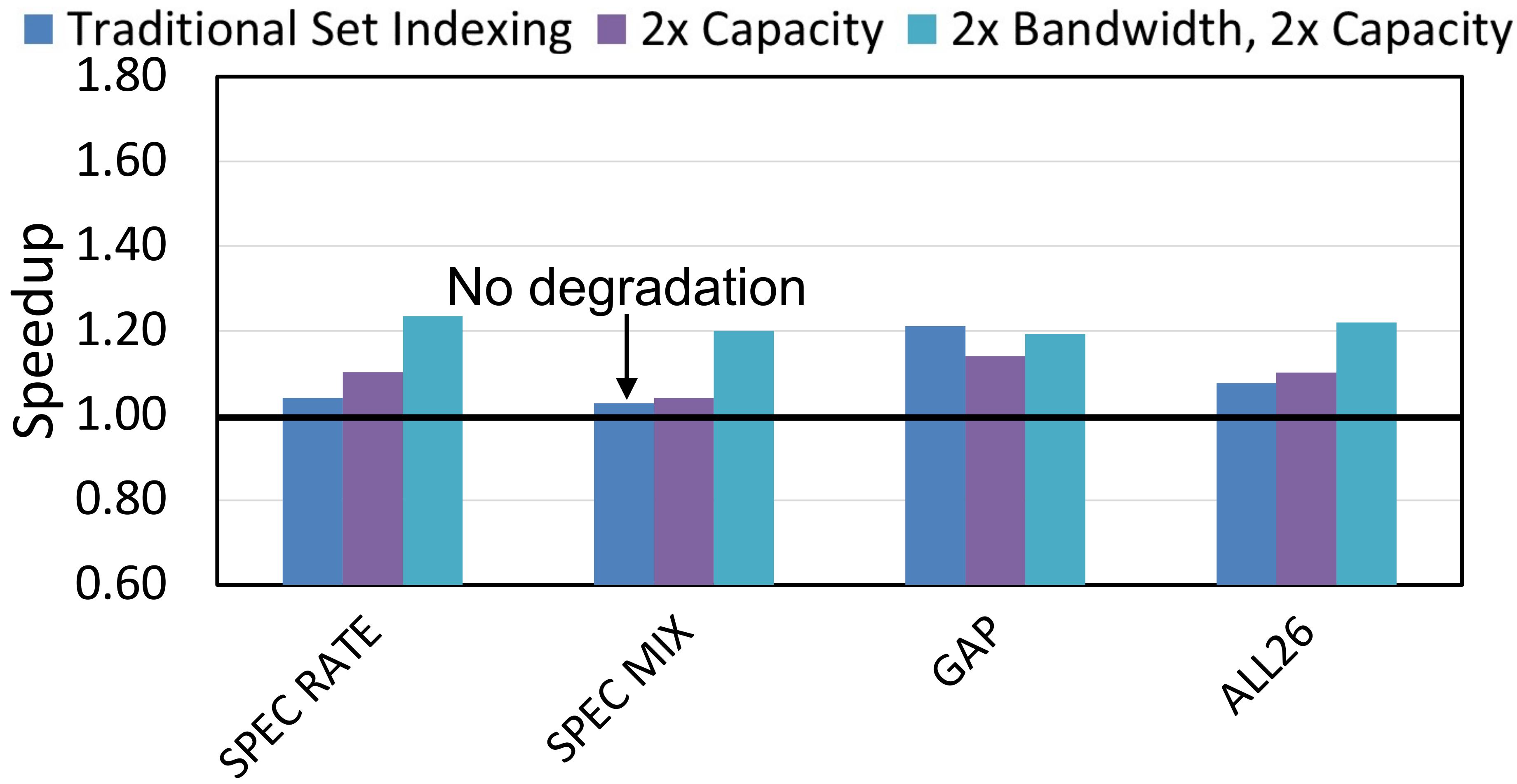
Compressible Incompressible Compressible Incompressible

INTRODUCTION: TRADITIONAL COMPRESSION



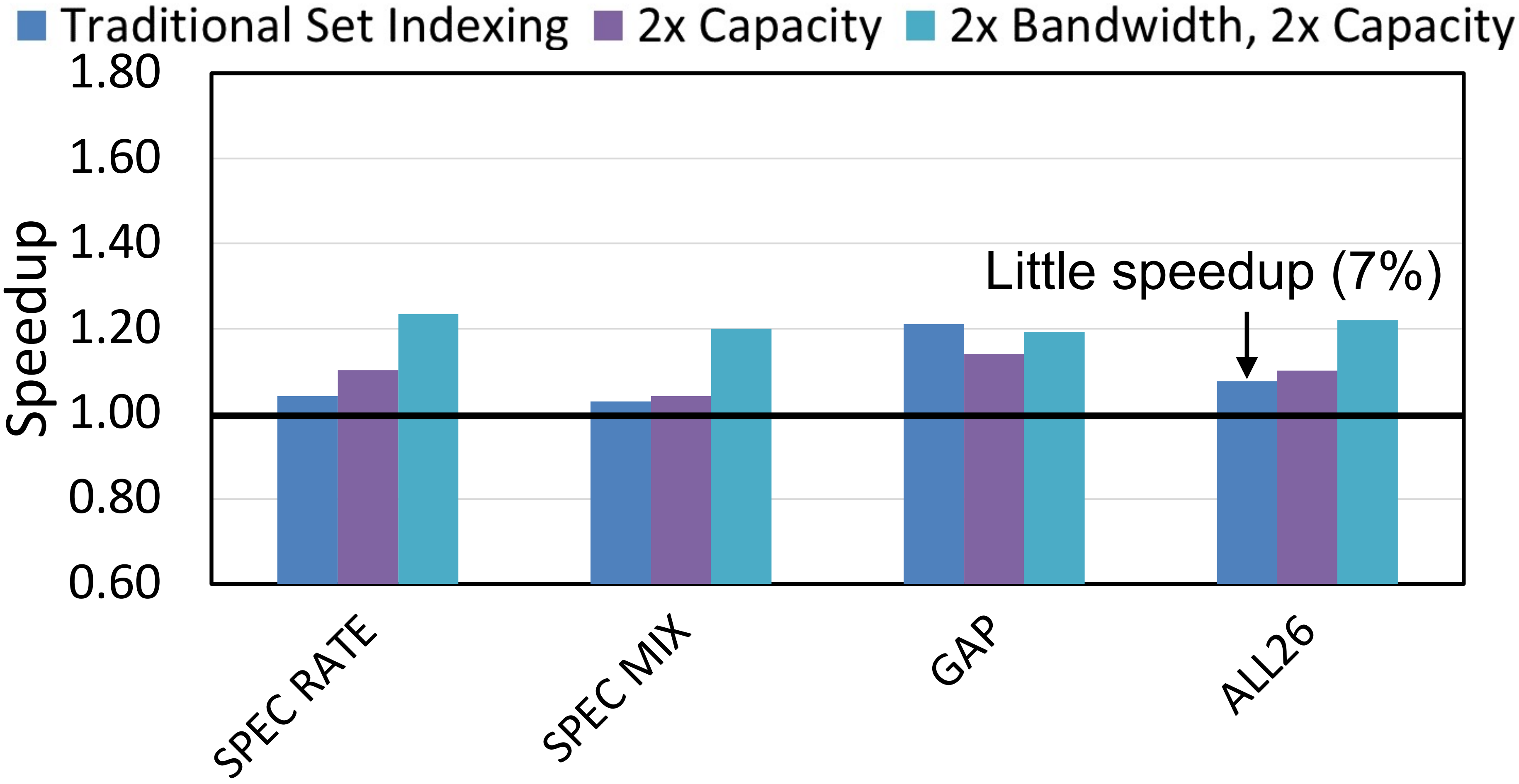
Compression for capacity (TSI) sees little speedup (7%) due to diminishing returns on giga-scale caches

INTRODUCTION: TRADITIONAL COMPRESSION



Compression for capacity (TSI) sees little speedup (7%) due to diminishing returns on giga-scale caches

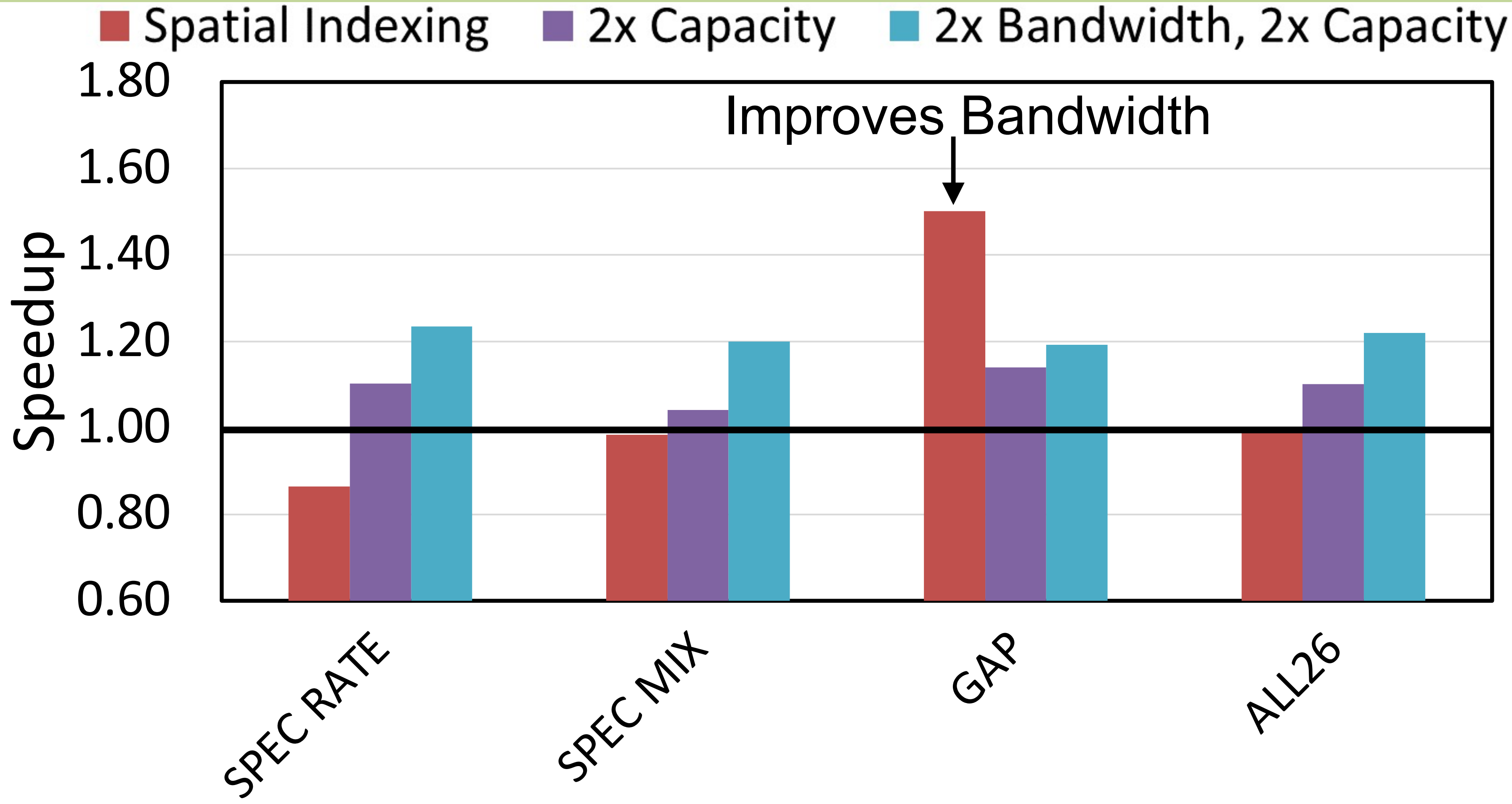
INTRODUCTION: TRADITIONAL COMPRESSION



Little speedup (7%)

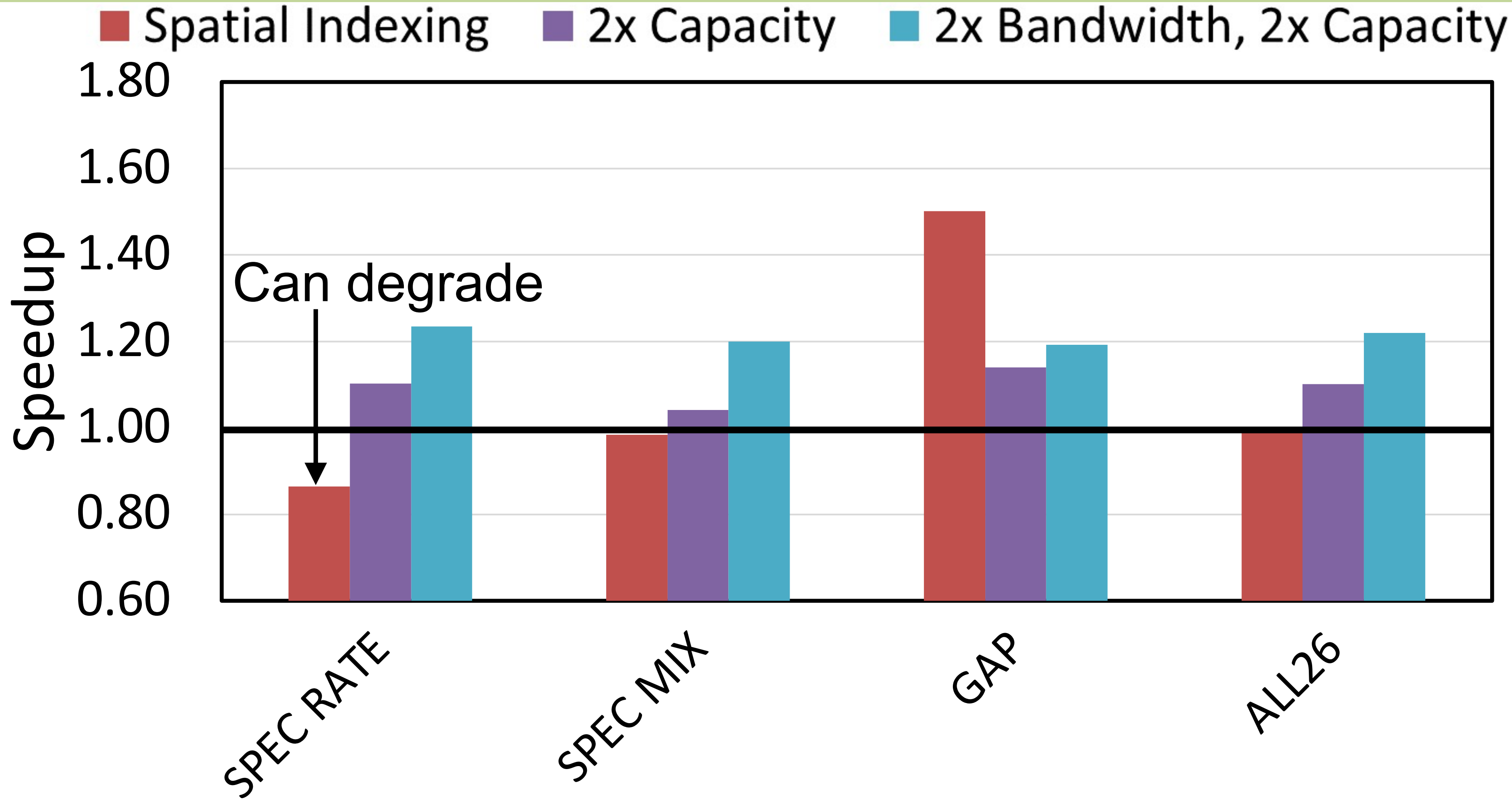
Compression for capacity (TSI) sees little speedup (7%) due to diminishing returns on giga-scale caches

INTRODUCTION: SPATIAL INDEXING



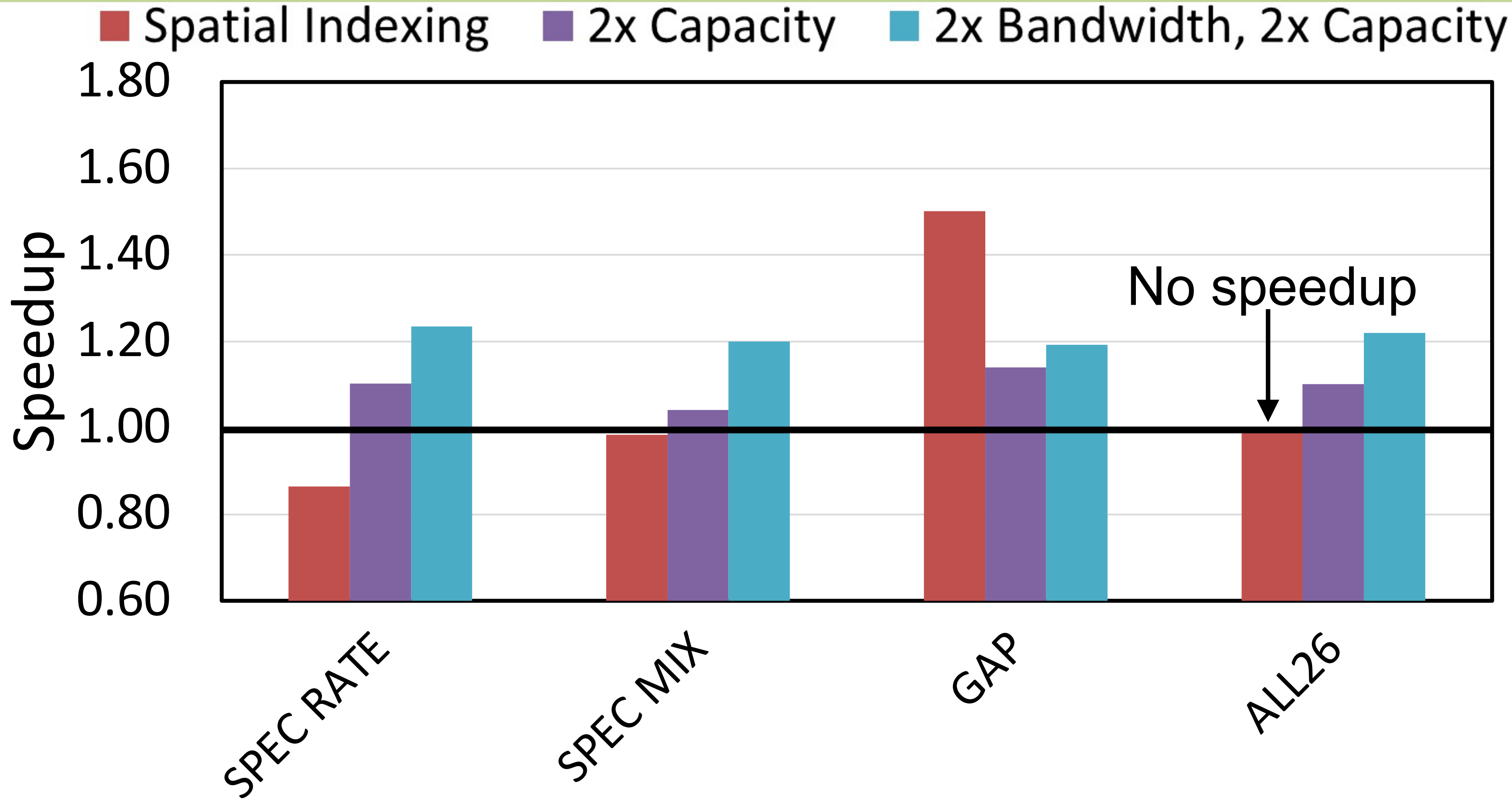
Spatial Indexing compression gets both benefits of bandwidth and capacity when lines are compressible. But, it hurts performance when lines are incompressible

INTRODUCTION: SPATIAL INDEXING



Spatial Indexing compression gets both benefits of bandwidth and capacity when lines are compressible. But, it hurts performance when lines are incompressible

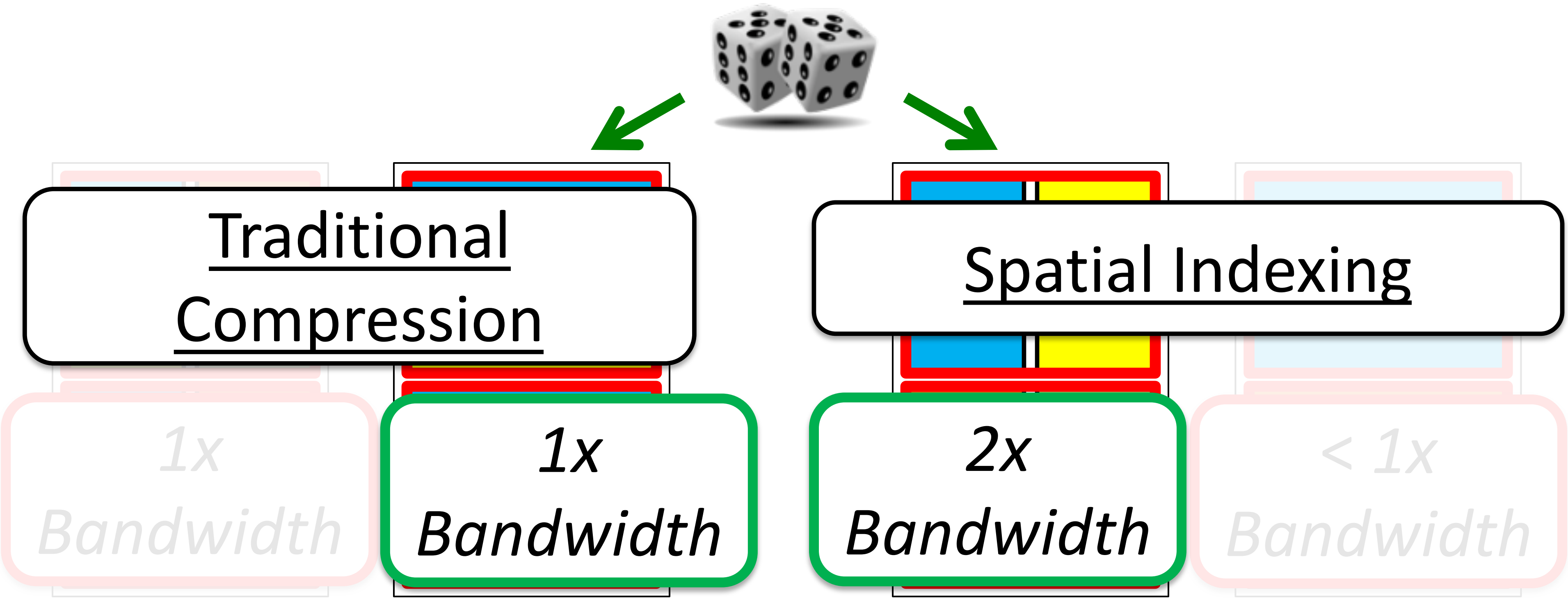
INTRODUCTION: SPATIAL INDEXING



Spatial Indexing compression gets both benefits of bandwidth and capacity when lines are compressible. But, it hurts performance when lines are incompressible

INTRODUCTION: COMPRESSED DRAM CACHE

Goal: Compression for Capacity **AND** Bandwidth



Traditional Compression

Spatial Indexing

1x Bandwidth

1x Bandwidth

2x Bandwidth

< 1x Bandwidth

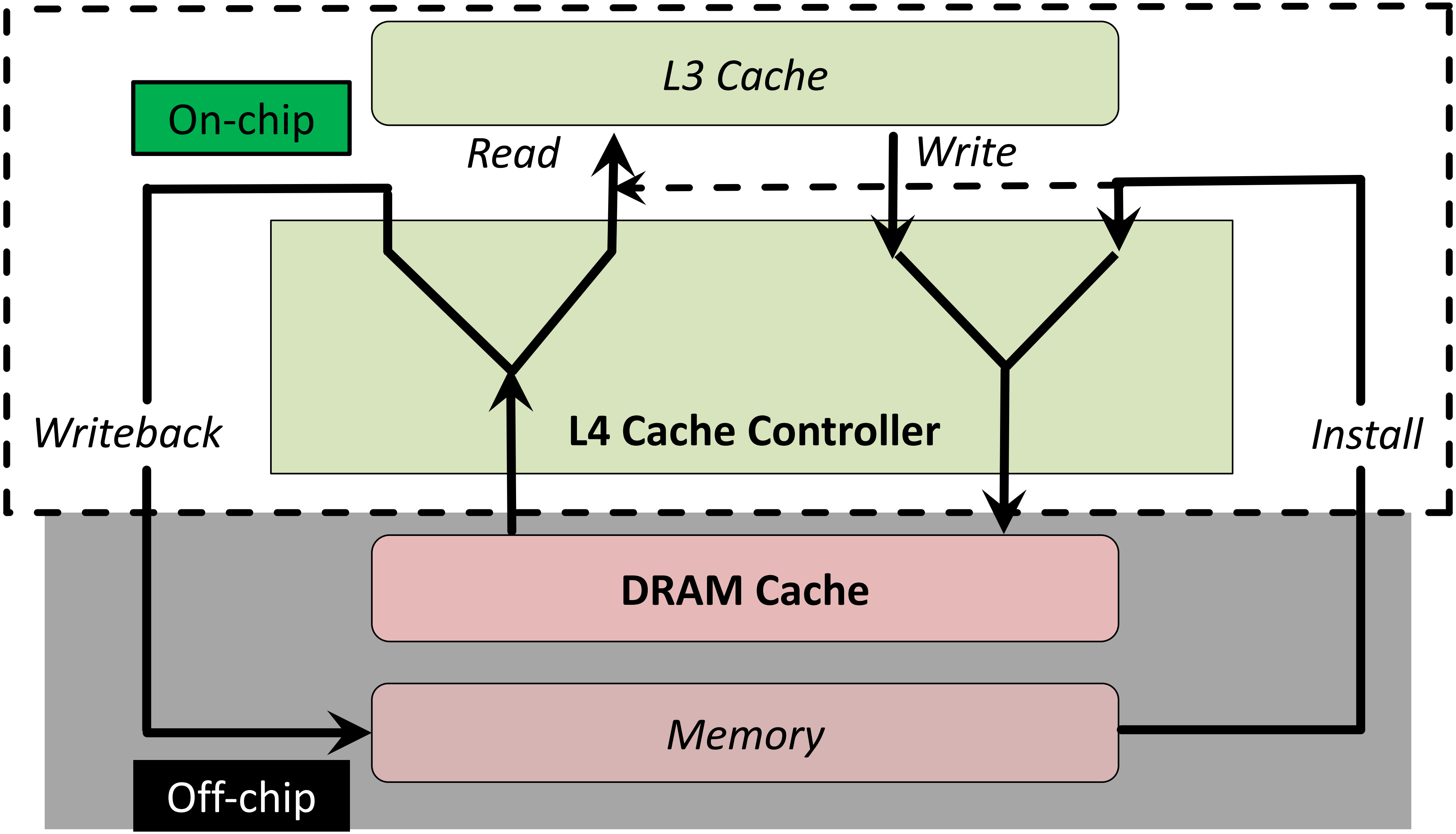
Compressible **Incompressible** **Compressible** **Incompressible**

DICE (Dynamic Index) → 19% Speedup + 36% ↓ EDP

DICE OVERVIEW

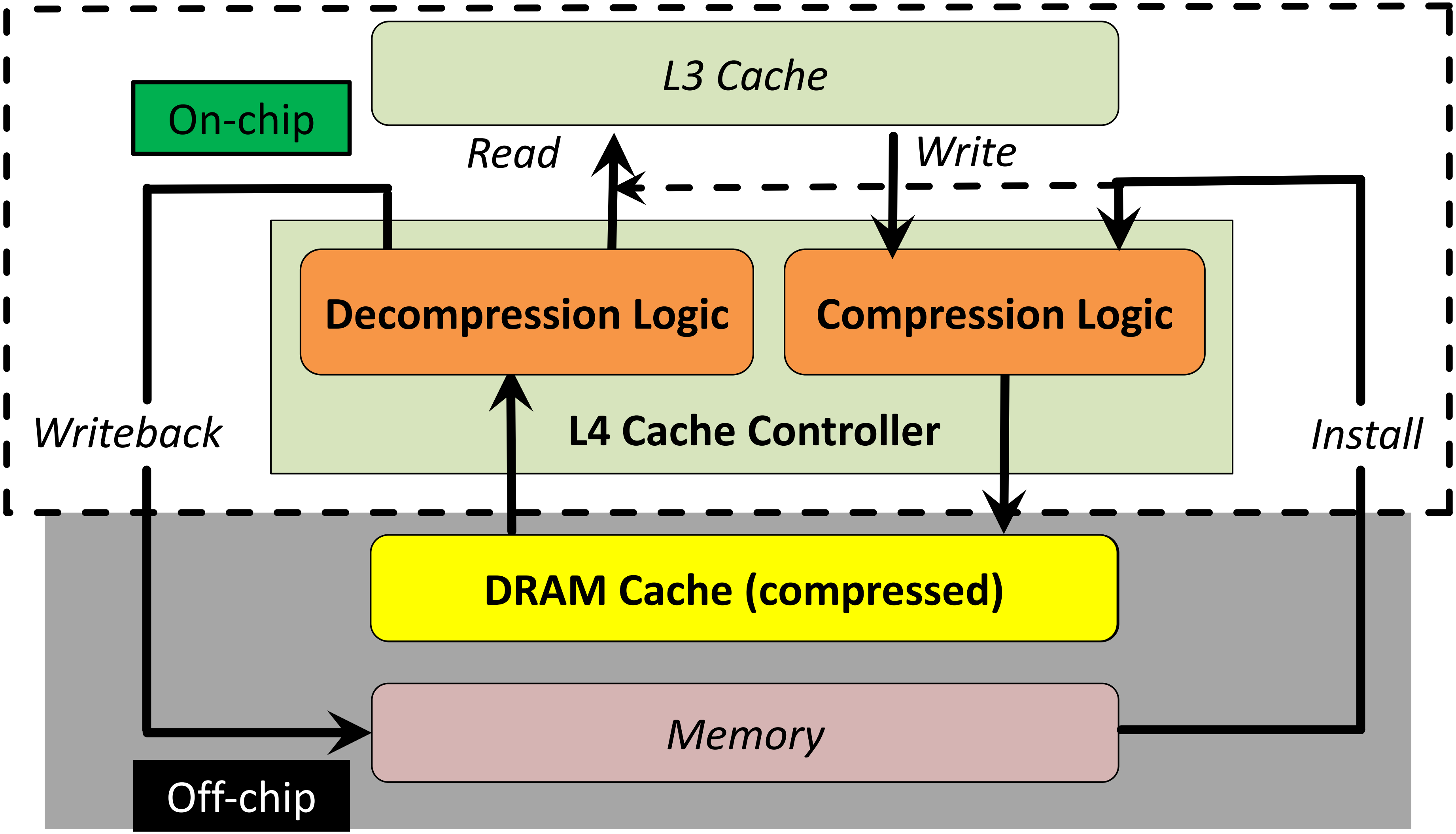
- Compressed DRAM Cache Organization ⬅
- Flexible Mapping for Quick Switching
- Dynamic Indexing ComprEssion (DICE)
  - Insertion Policy
  - Index Prediction

PRACTICAL DRAM CACHE COMPRESSION



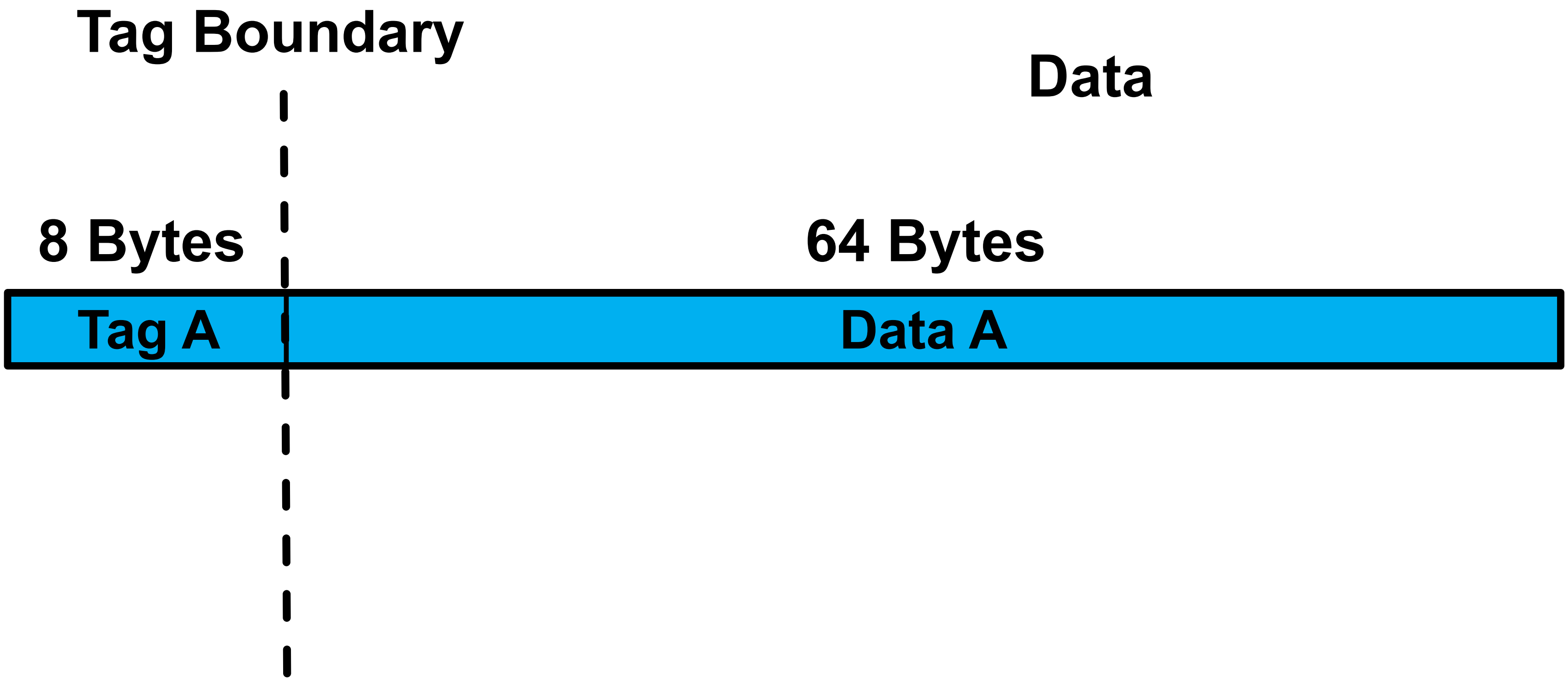
Compression: Simple changes within the controller

PRACTICAL DRAM CACHE COMPRESSION



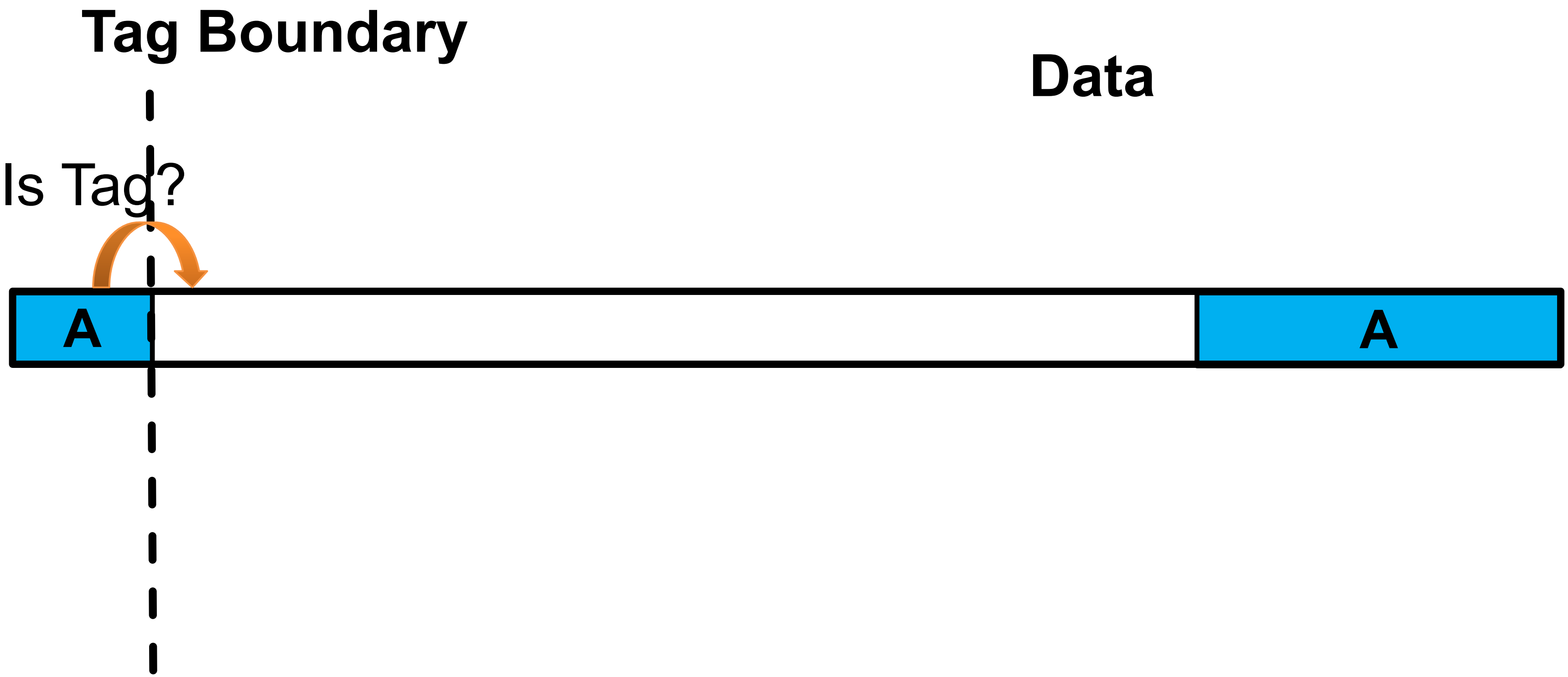
Compression: Simple changes within the controller

DRAM CACHE TAG FORMAT



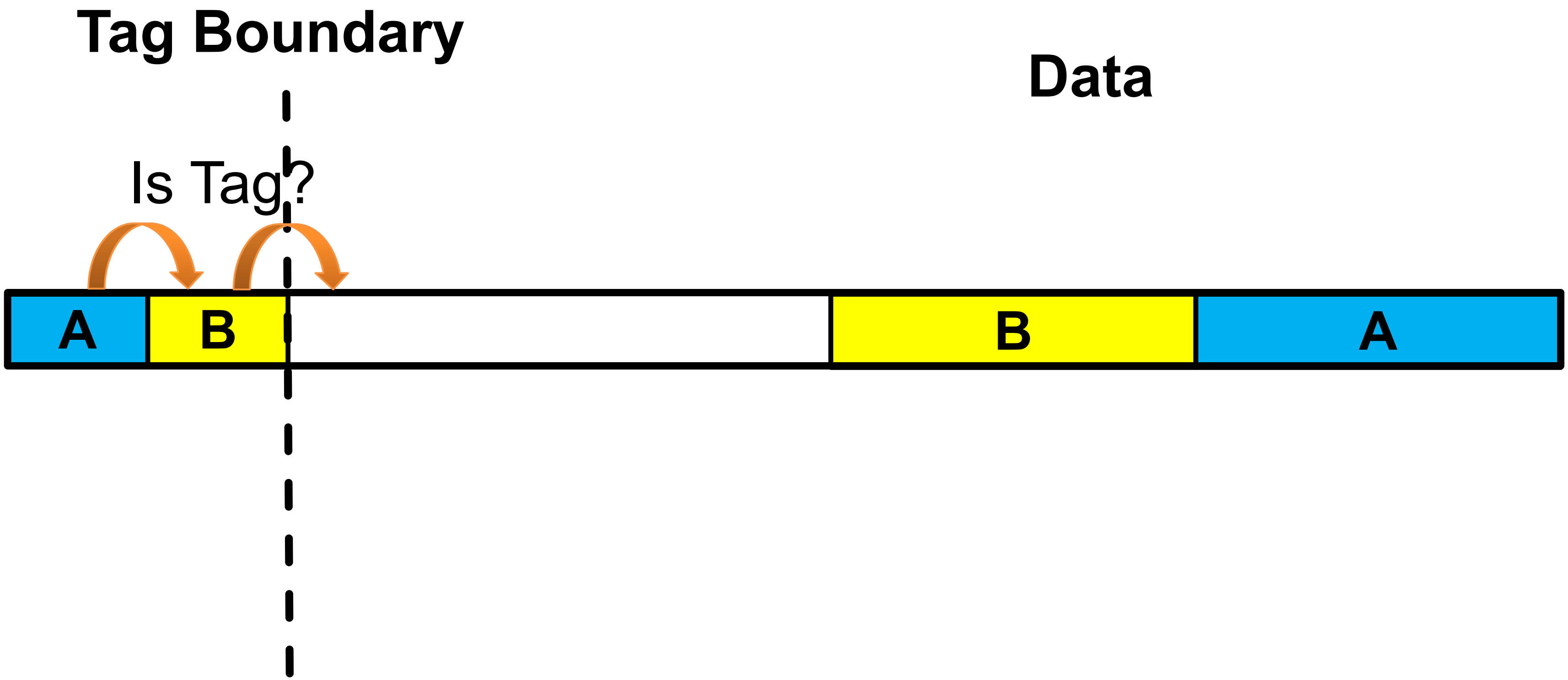
Cache controller receives 72B of tag+data. It can flexibly interpret bits as tag bits or data bits.

PROPOSED FLEXIBLE TAG FORMAT



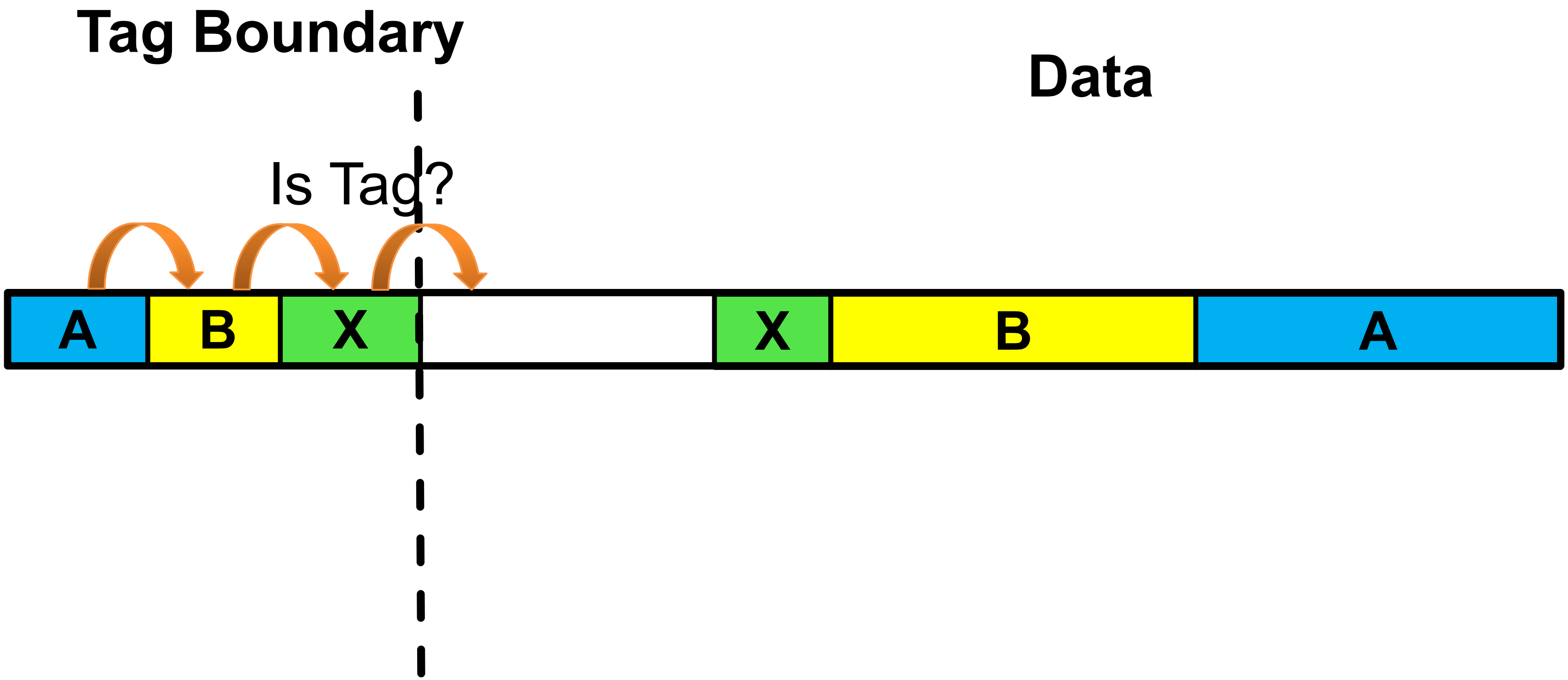
We create Tag space as needed, for up to 28 lines.
Achieves 1.6x effective capacity.

PROPOSED FLEXIBLE TAG FORMAT



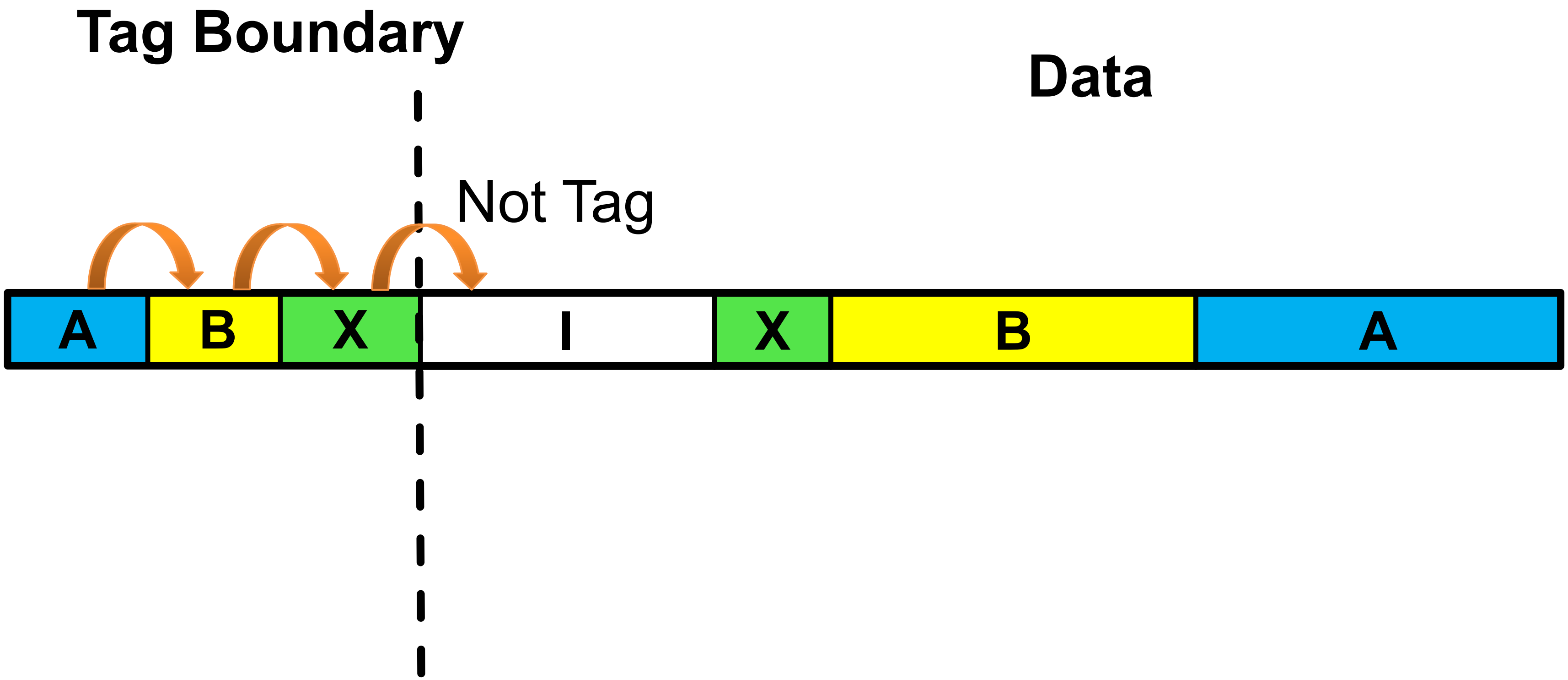
We create Tag space as needed, for up to 28 lines.
Achieves 1.6x effective capacity.

PROPOSED FLEXIBLE TAG FORMAT



We create Tag space as needed, for up to 28 lines.
Achieves 1.6x effective capacity.

PROPOSED FLEXIBLE TAG FORMAT

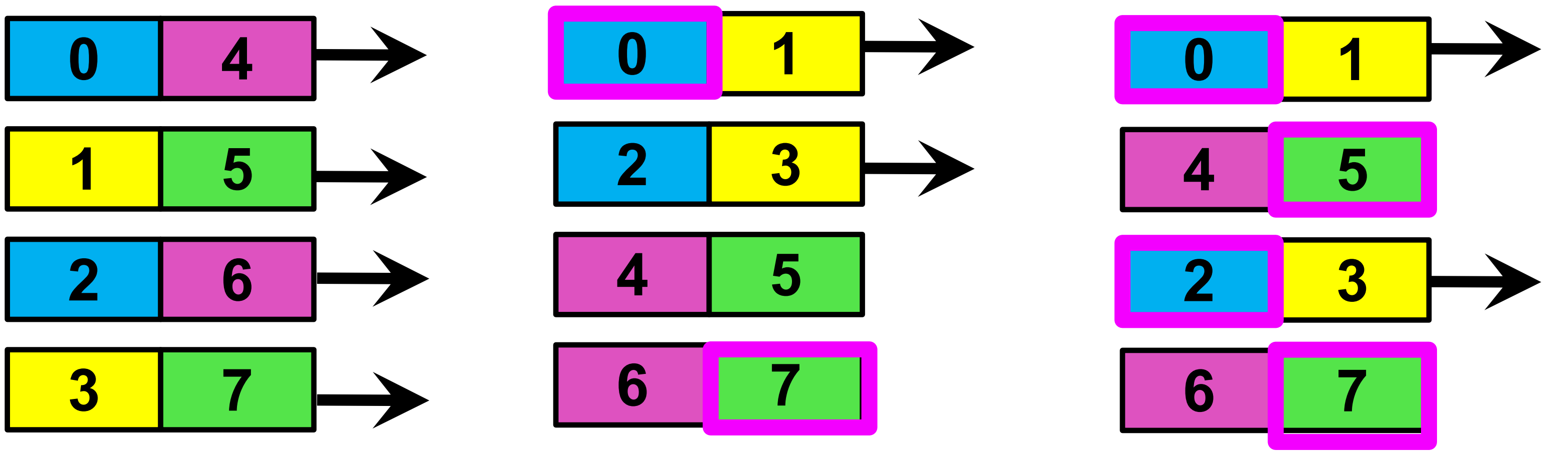


We create Tag space as needed, for up to 28 lines.
Achieves 1.6x effective capacity.

DICE OVERVIEW

- Compressed DRAM Cache Organization
- Flexible Mapping for Quick Switching ⇦
- Dynamic Indexing ComprEssion (DICE)
  - Insertion Policy
  - Index Prediction

FLEXIBLE MAPPING (*TSI* OR *BAI*)



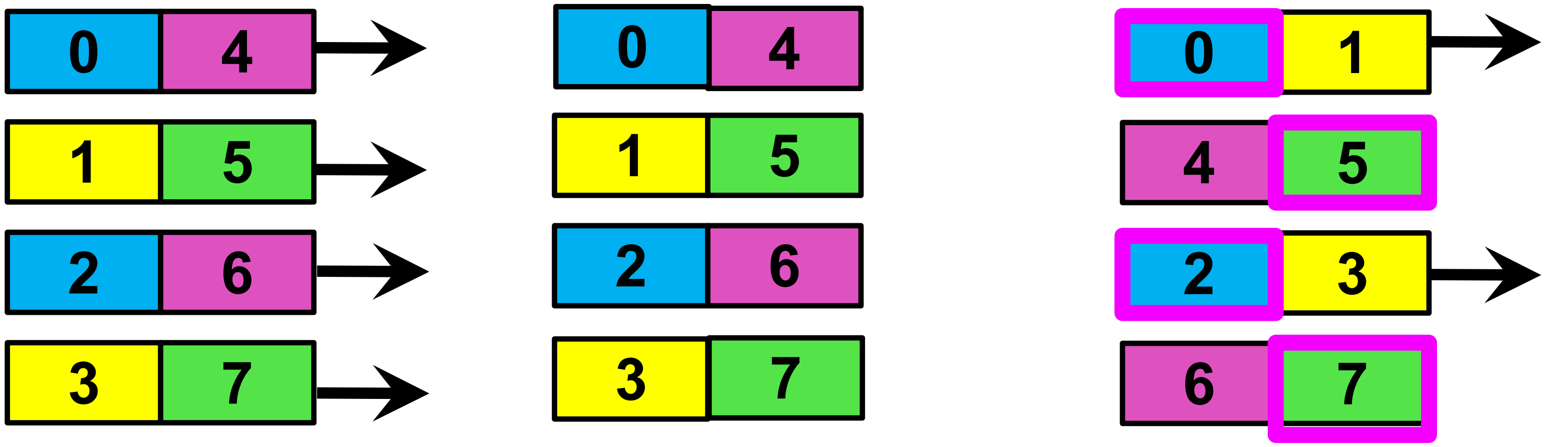
Traditional Set Indexing (TSI)

Naïve Spatial Indexing

Bandwidth-Aware Indexing (BAI)

Bandwidth-Aware Indexing (BAI) facilitates quick switching between two indices TSI and BAI.

FLEXIBLE MAPPING (*TSI* OR *BAI*)



0	4
1	5
2	6
3	7

Traditional Set Indexing (TSI)

0	4
1	5
2	6
3	7

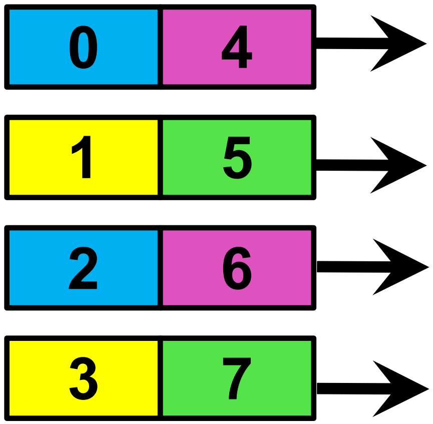
Naïve Spatial Indexing

0	1
4	5
2	3
6	7

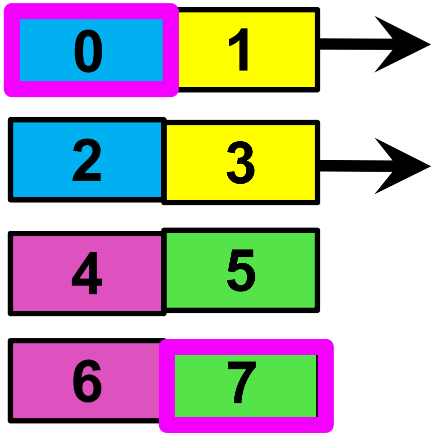
Bandwidth-Aware Indexing (BAI)

Bandwidth-Aware Indexing (BAI) facilitates quick switching between two indices TSI and BAI.

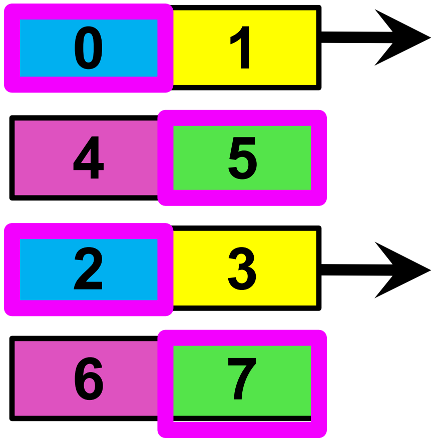
FLEXIBLE MAPPING (*TSI* OR *BAI*)



Traditional Set Indexing (TSI)



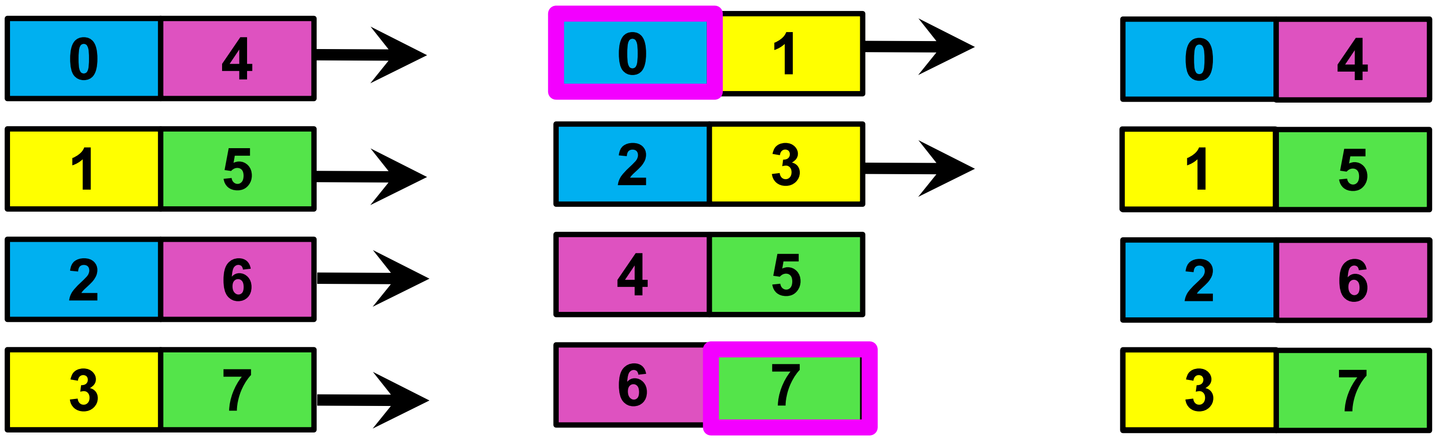
Naïve Spatial Indexing



Bandwidth-Aware Indexing (BAI)

Bandwidth-Aware Indexing (BAI) facilitates quick switching between two indices TSI and BAI.

FLEXIBLE MAPPING (*TSI* OR *BAI*)



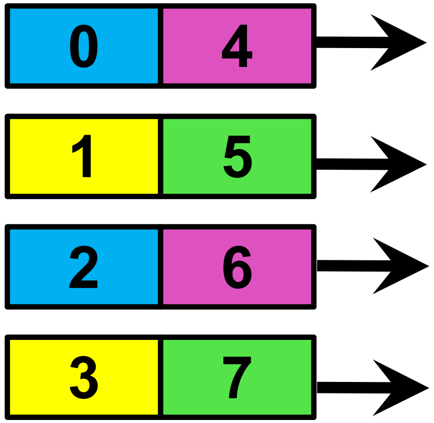
Traditional Set Indexing (TSI)

Naïve Spatial Indexing

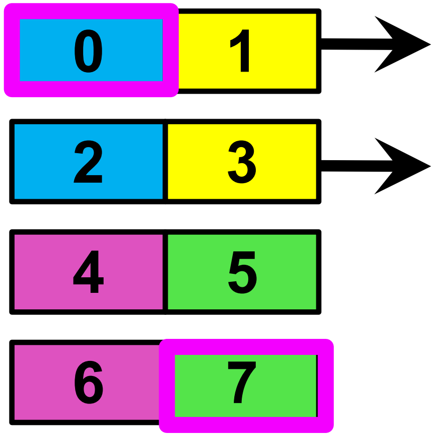
Bandwidth-Aware Indexing (BAI)

Bandwidth-Aware Indexing (BAI) facilitates quick switching between two indices TSI and BAI.

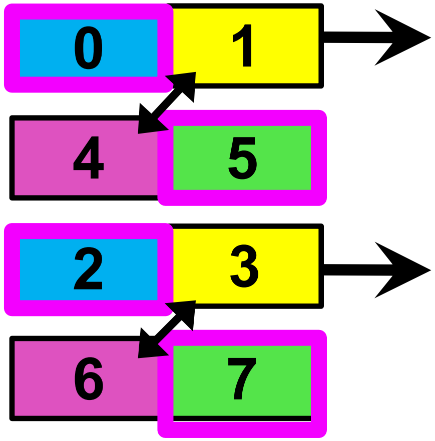
FLEXIBLE MAPPING (*TSI* OR *BAI*)



Traditional Set Indexing (TSI)



Naïve Spatial Indexing



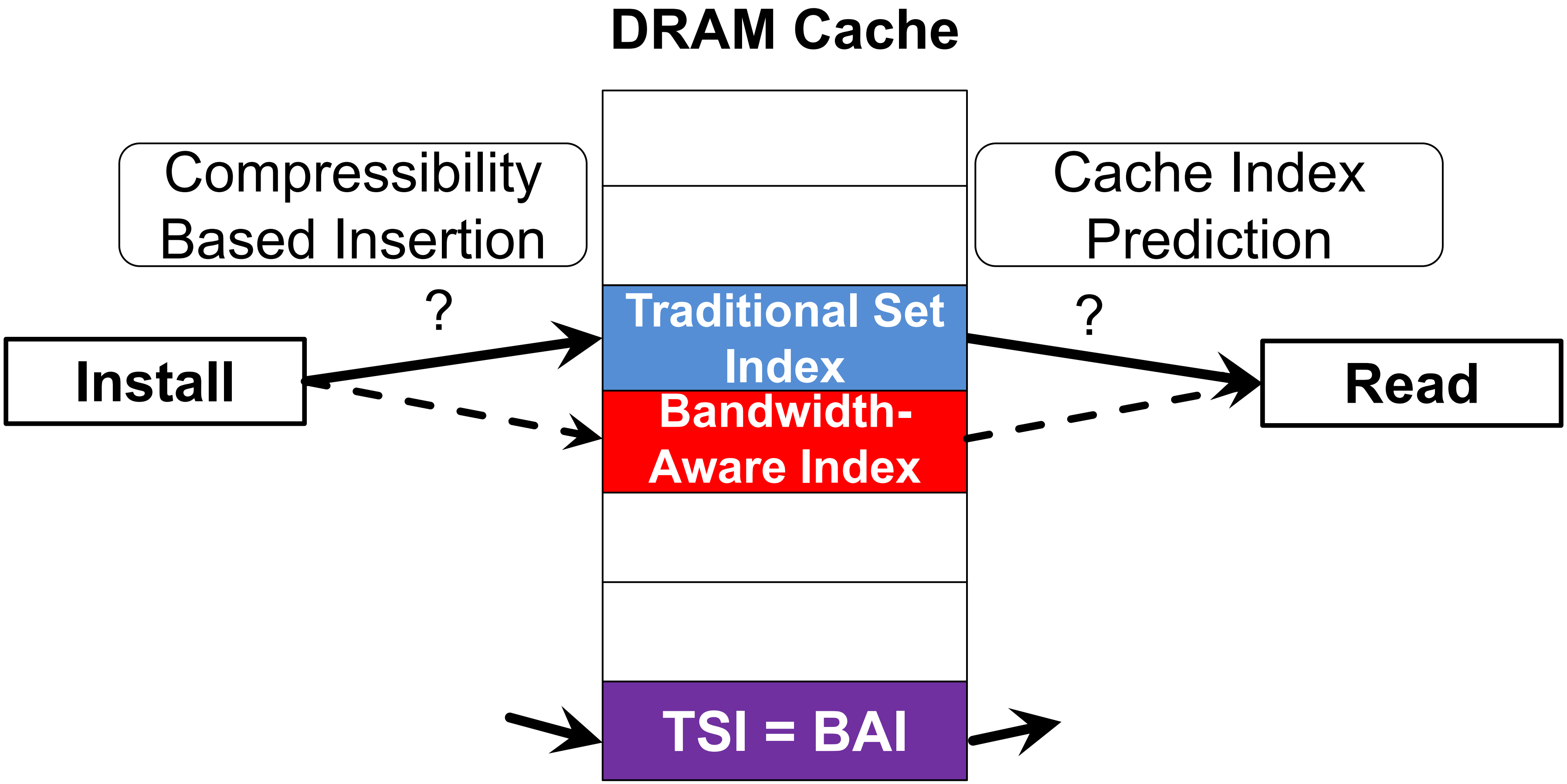
Bandwidth-Aware Indexing (BAI)

Bandwidth-Aware Indexing (BAI) facilitates quick switching between two indices TSI and BAI.

DICE OVERVIEW

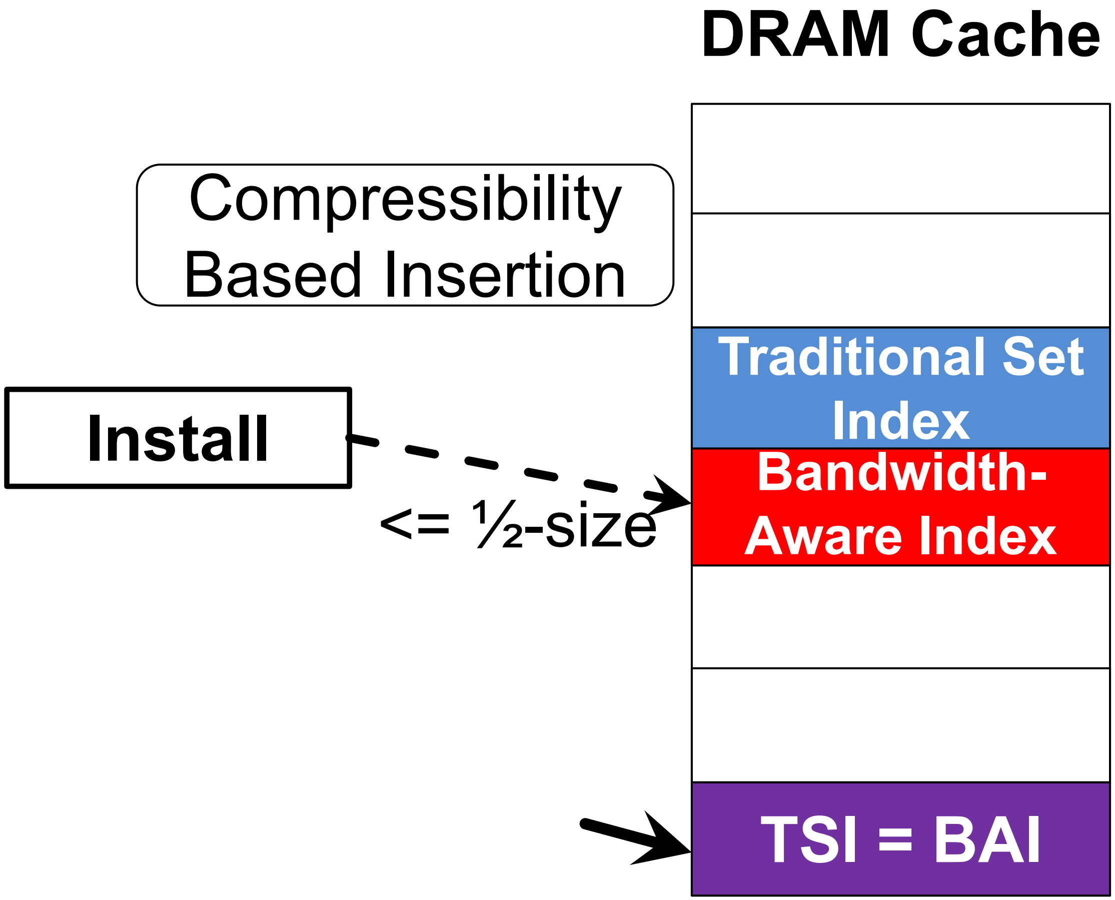
- Compressed DRAM Cache Organization
- Flexible Mapping for Quick Switching
- Dynamic Indexing ComprEssion (DICE)
  - Insertion Policy
  - Index Prediction

DICE: DYNAMIC-INDEXED COMPRESSED CACHE



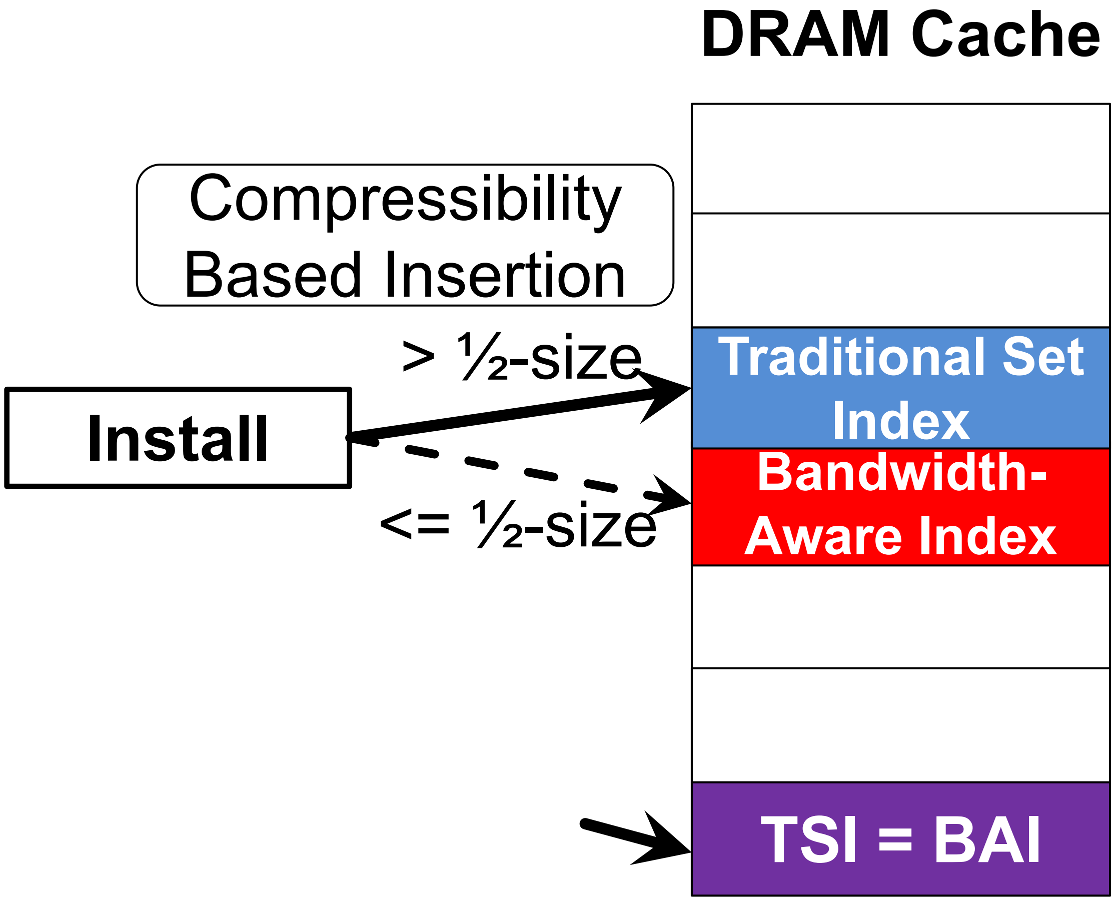
DICE: Dynamic-Indexing Cache comprEssion, decides index on install, and predicts index on read

COMPRESSIBILITY-BASED INSERTION



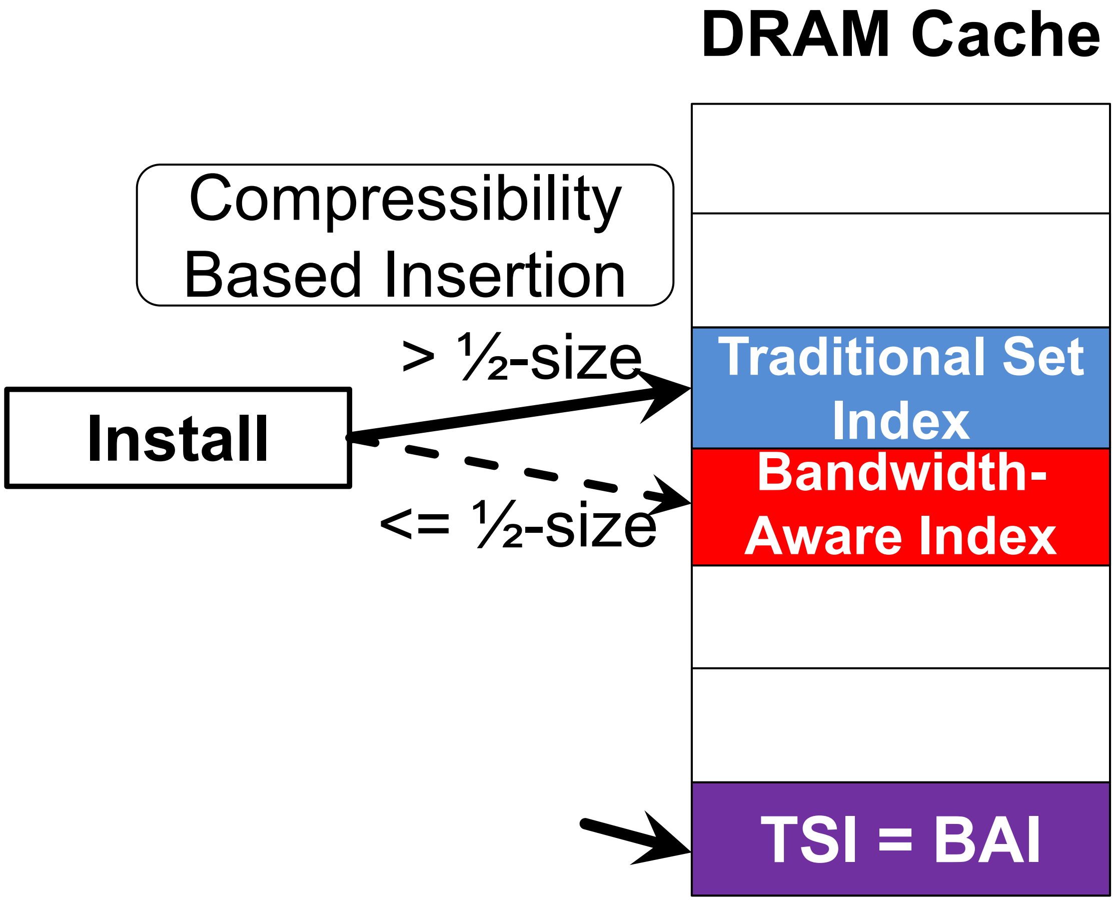
Compressibilty-based insertion uses Bandwidth-Aware Indexing when lines are compressible, and TSI otherwise

COMPRESSIBILITY-BASED INSERTION



Compressibilty-based insertion uses Bandwidth-Aware Indexing when lines are compressible, and TSI otherwise

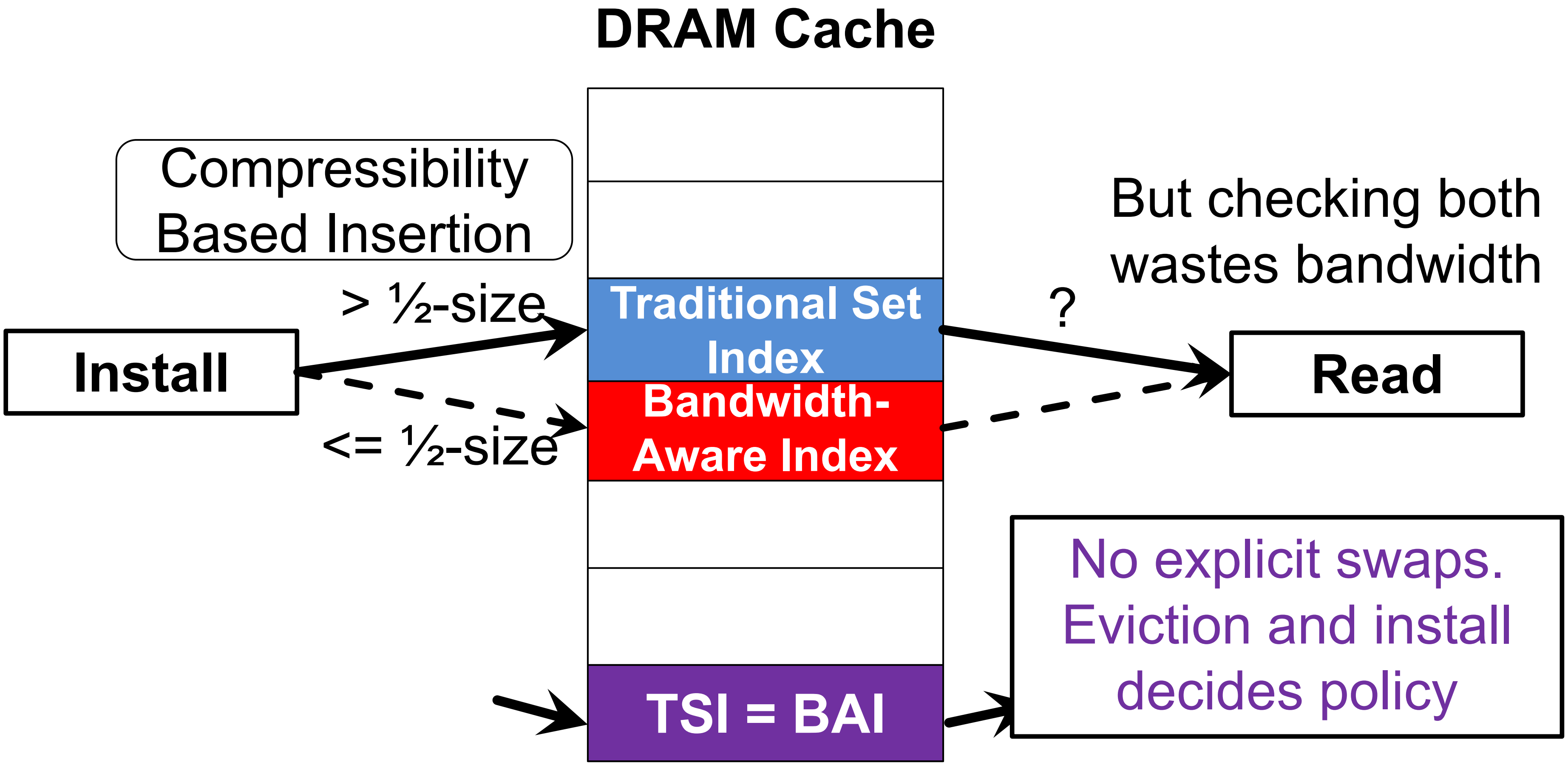
COMPRESSIBILITY-BASED INSERTION



No explicit swaps.
Eviction and install decides policy

Compressibilty-based insertion uses Bandwidth-Aware Indexing when lines are compressible, and TSI otherwise

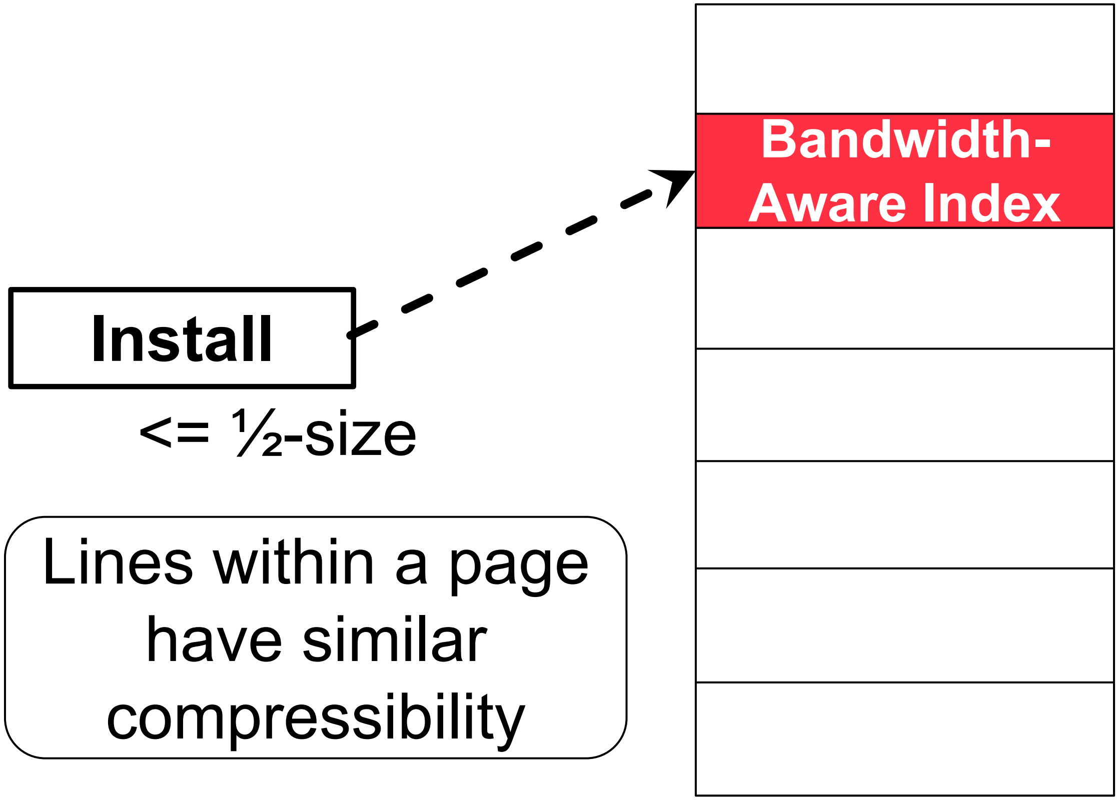
COMPRESSIBILITY-BASED INSERTION


DRAM Cache
Compressibility Based Insertion
> ½-size
<= ½-size
Install
Traditional Set Index
Bandwidth-Aware Index
?
But checking both wastes bandwidth
Read
TSI = BAI
No explicit swaps. Eviction and install decides policy

Compressibilty-based insertion uses Bandwidth-Aware Indexing when lines are compressible, and TSI otherwise

SIMILAR INTRA-PAGE COMPRESSIBILITY

Indices seen in a Compressible Page



Install

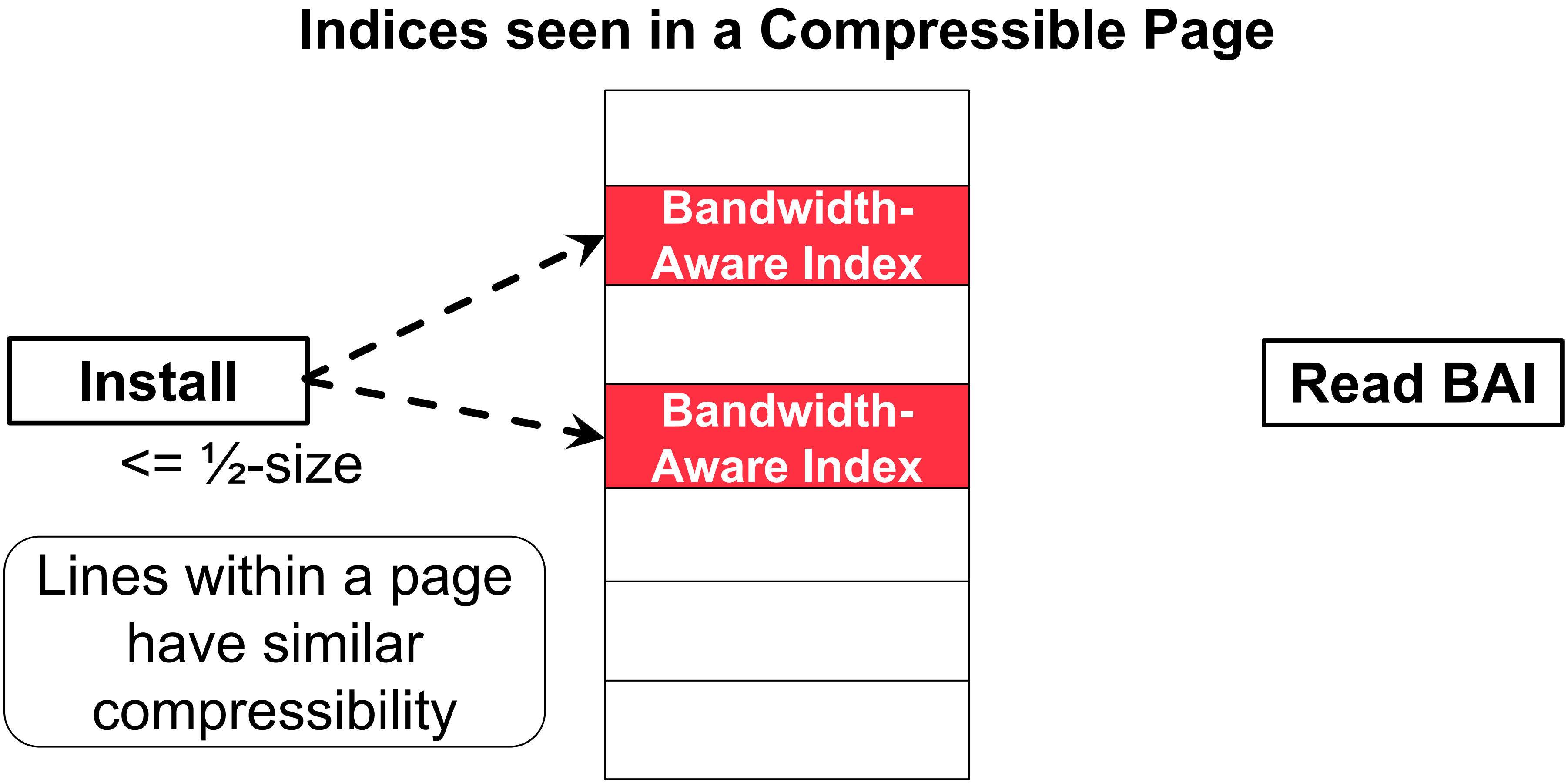
<= ½-size

Read BAI

Lines within a page have similar compressibility

DICE is likely to install lines of a page into similar index

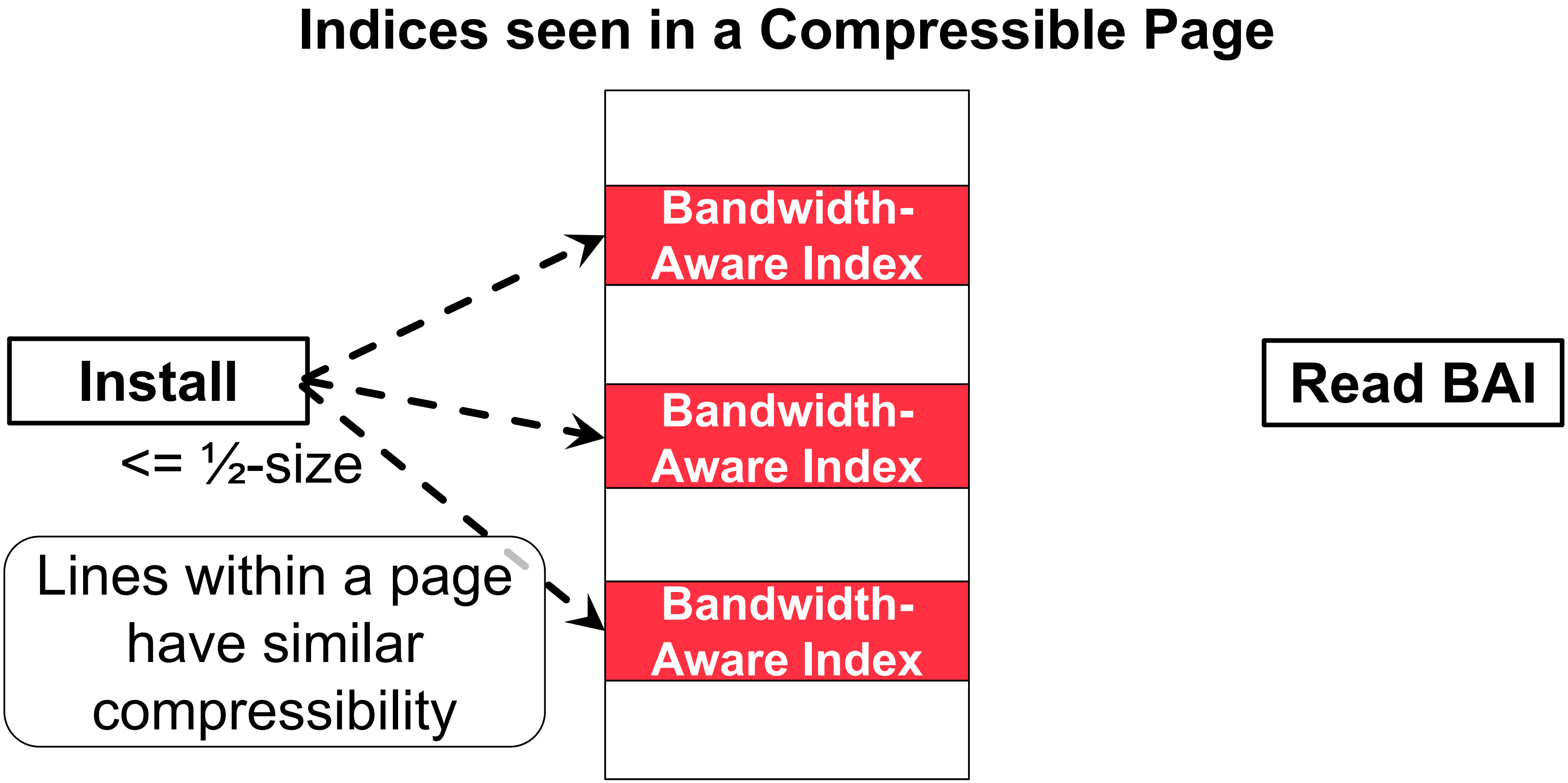
SIMILAR INTRA-PAGE COMPRESSIBILITY



Lines within a page have similar compressibility

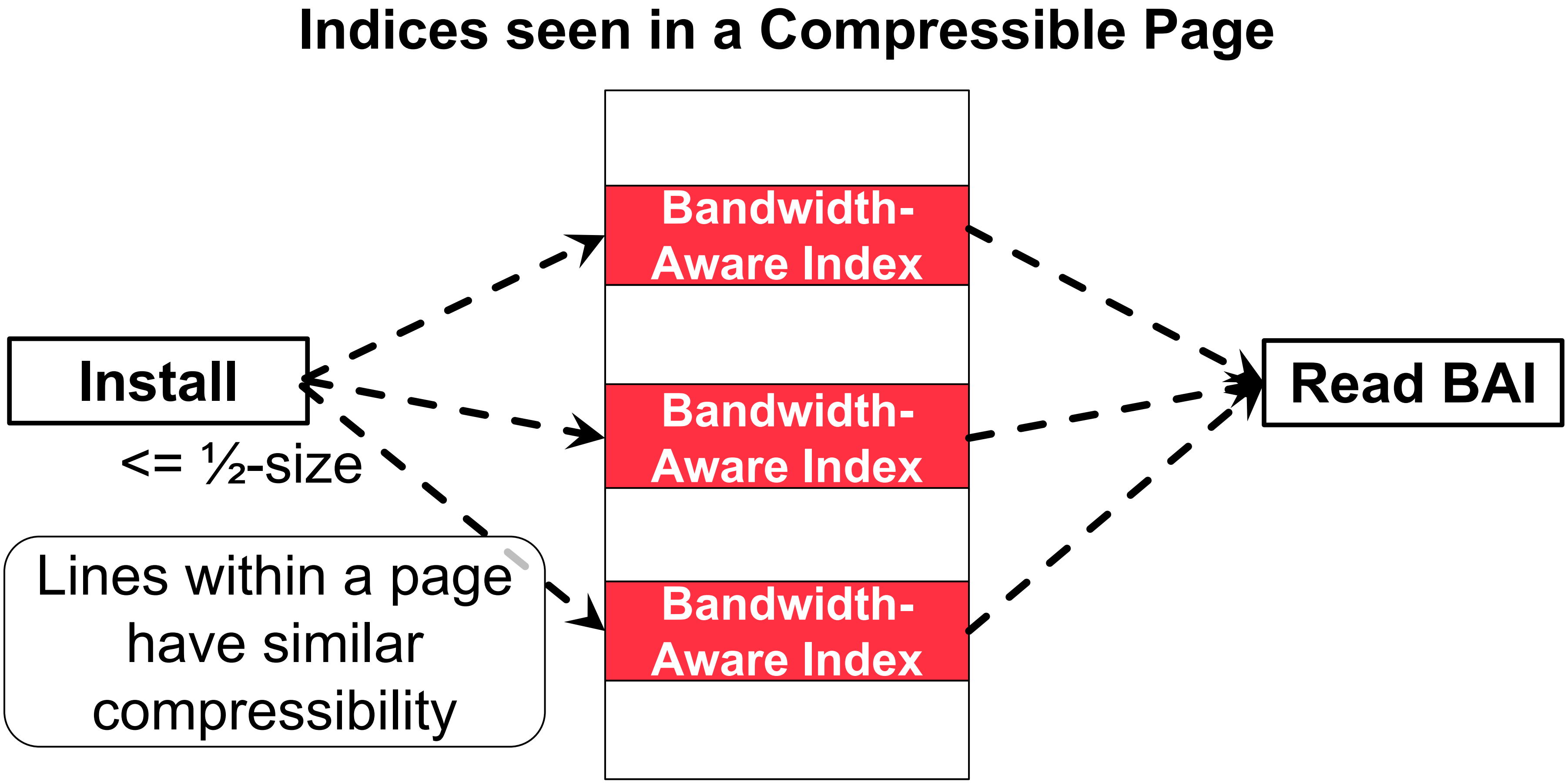
DICE is likely to install lines of a page into similar index

SIMILAR INTRA-PAGE COMPRESSIBILITY



DICE is likely to install lines of a page into similar index

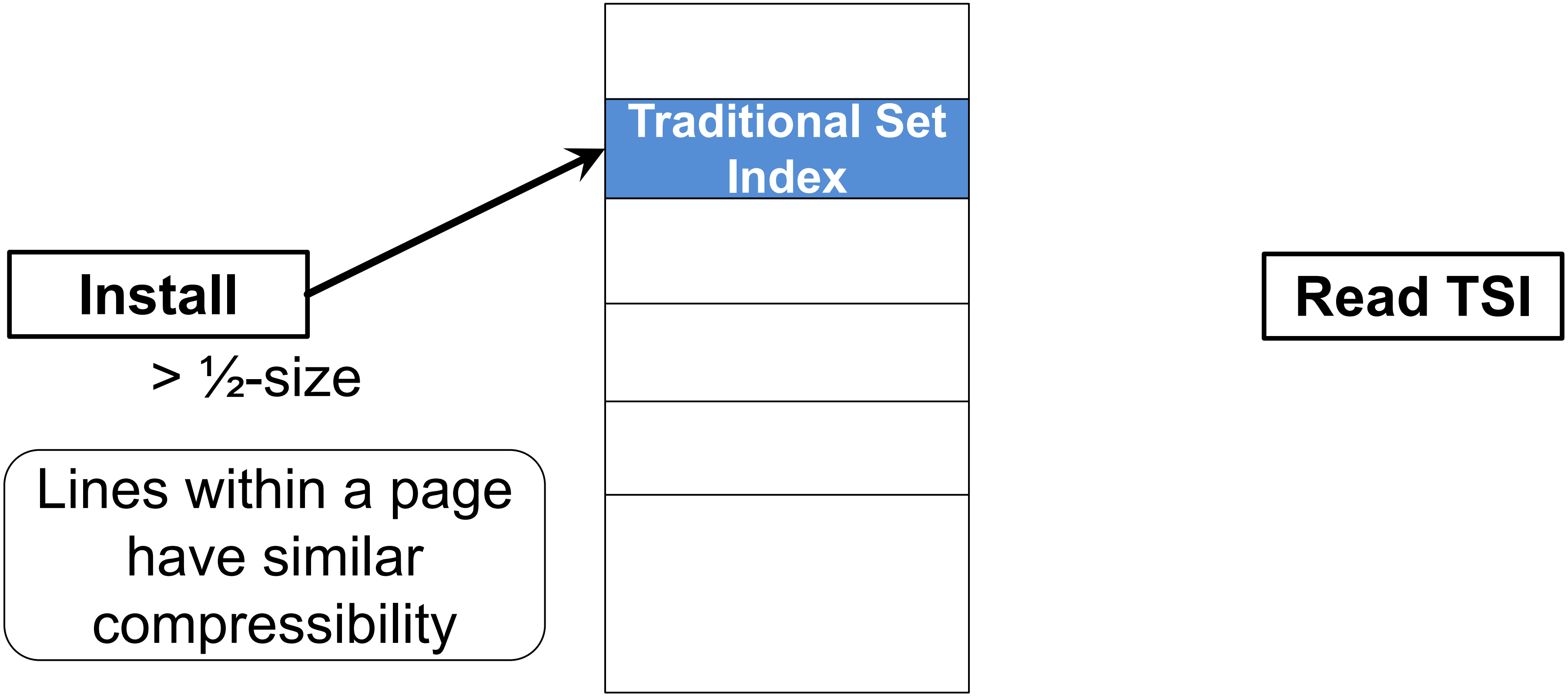
SIMILAR INTRA-PAGE COMPRESSIBILITY



Lines within a page have similar compressibility

DICE is likely to install lines of a page into similar index

SIMILAR INTRA-PAGE COMPRESSIBILITY

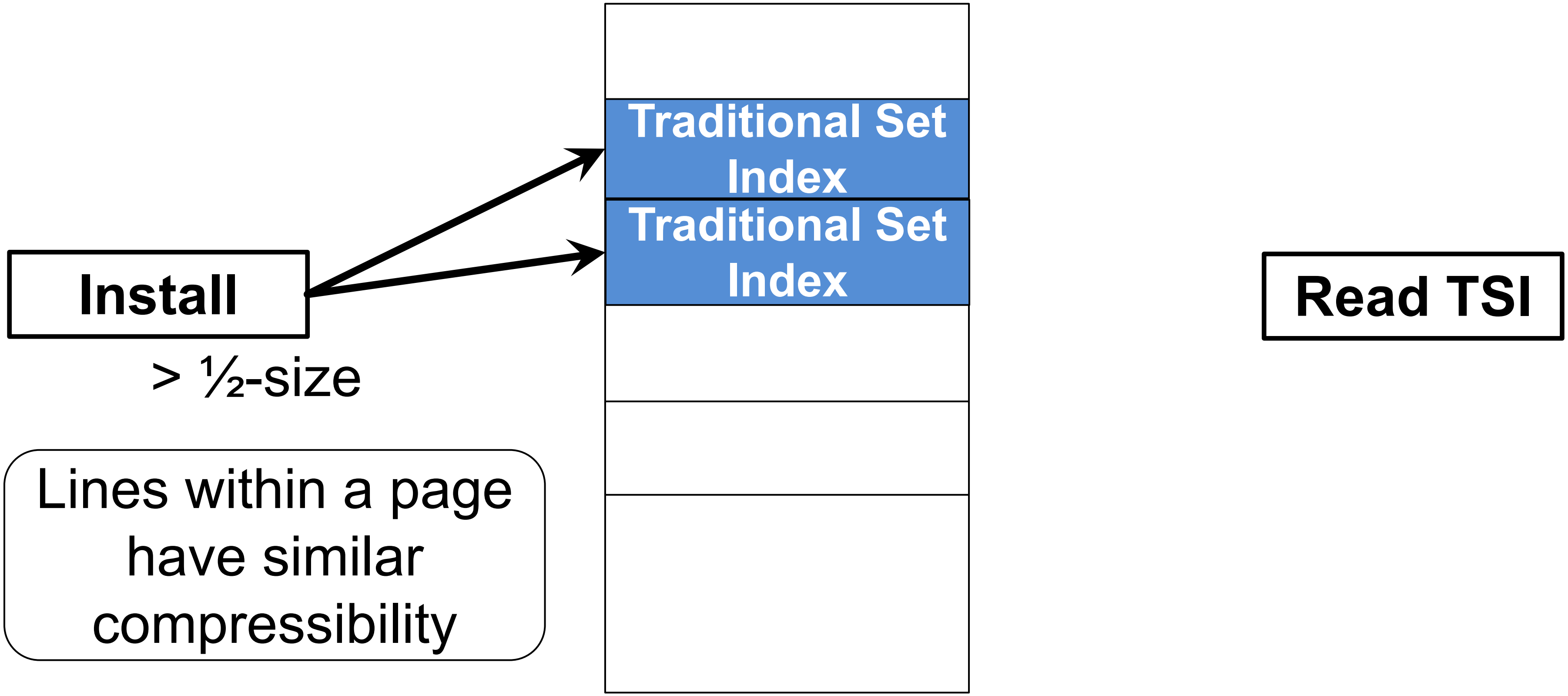


Lines within a page have similar compressibility

Thus, page-based last-time prediction of index can be accurate (94%)

SIMILAR INTRA-PAGE COMPRESSIBILITY

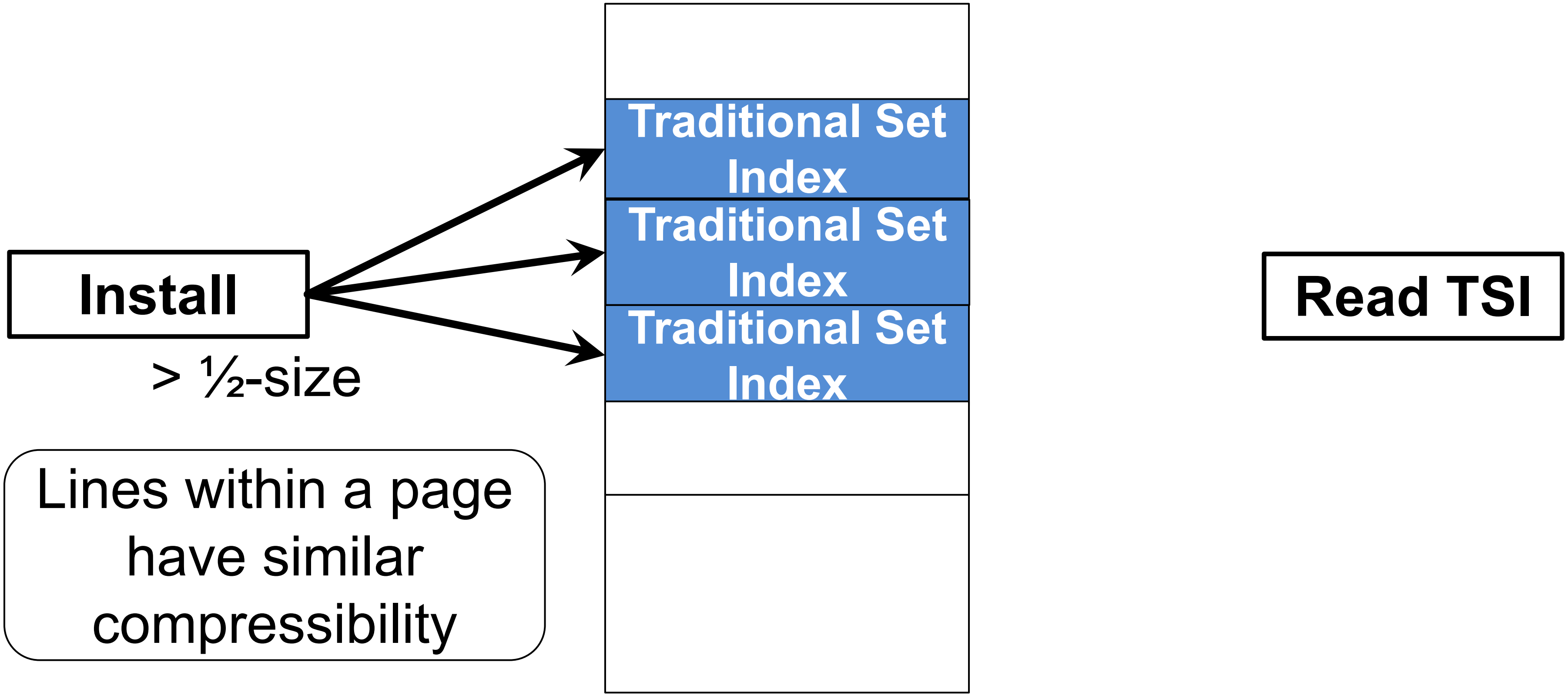
Indices seen in an Incompressible Page



Lines within a page have similar compressibility

Thus, page-based last-time prediction of index can be accurate (94%)

SIMILAR INTRA-PAGE COMPRESSIBILITY

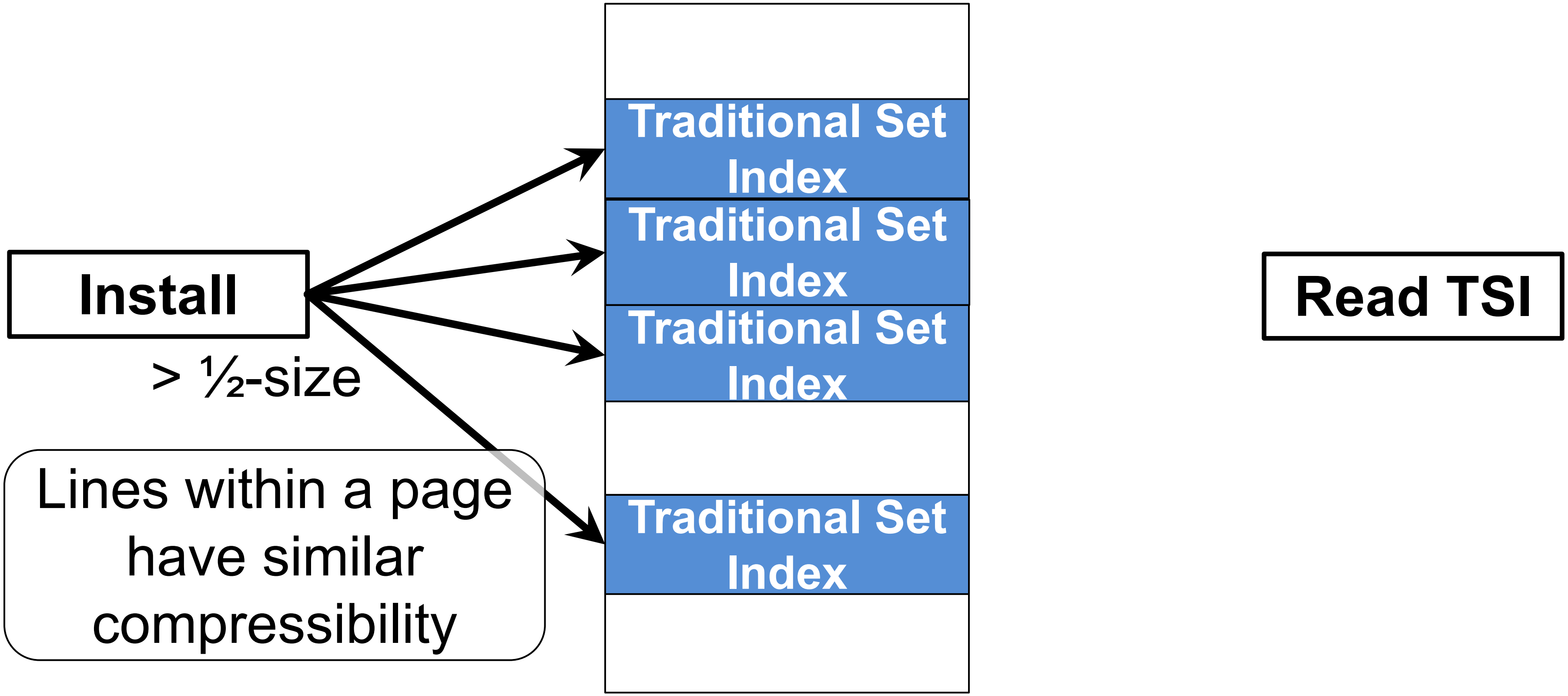


Lines within a page have similar compressibility

Thus, page-based last-time prediction of index can be accurate (94%)

SIMILAR INTRA-PAGE COMPRESSIBILITY

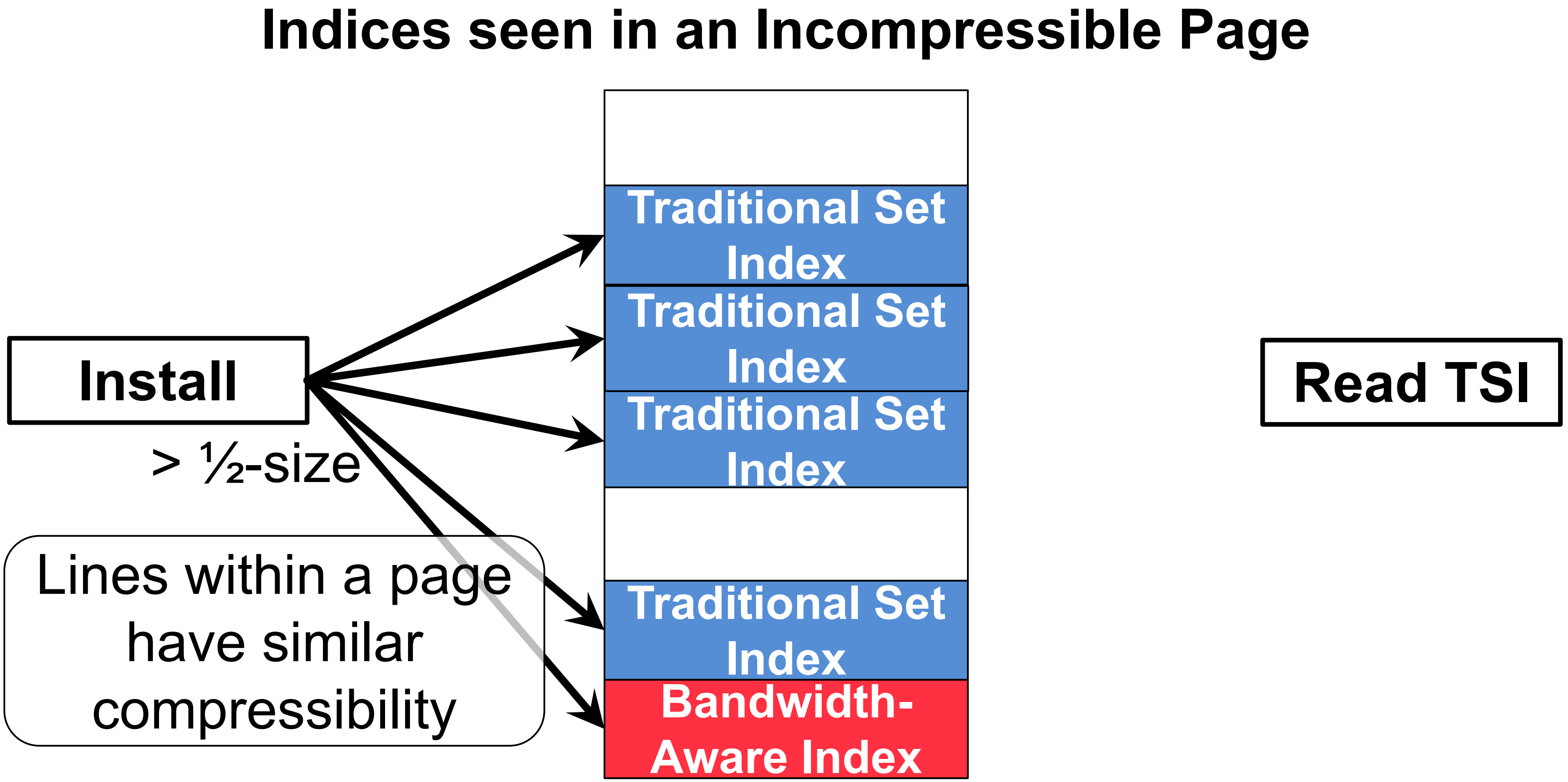
Indices seen in an Incompressible Page



Lines within a page have similar compressibility

Thus, page-based last-time prediction of index can be accurate (94%)

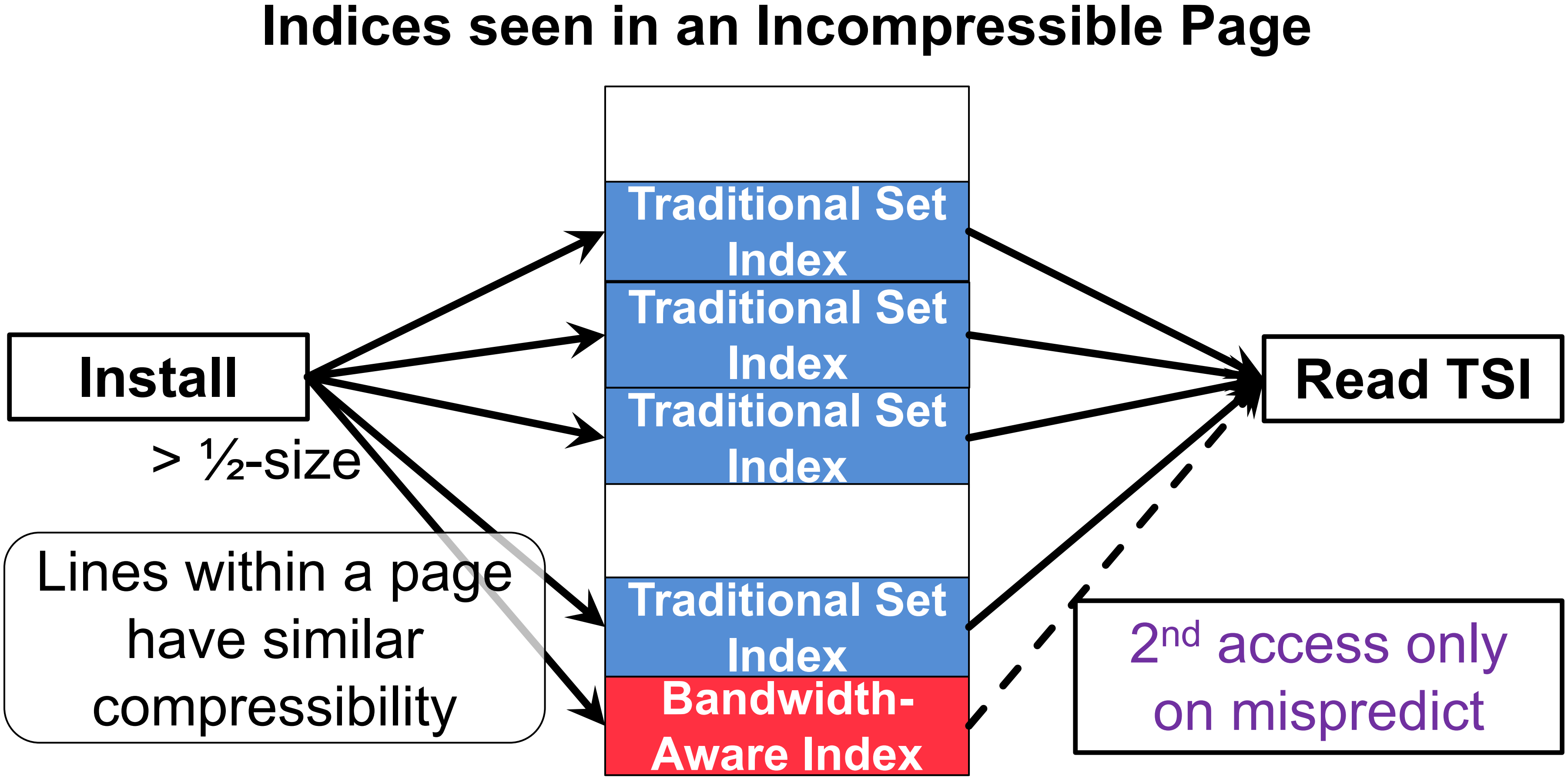
SIMILAR INTRA-PAGE COMPRESSIBILITY



Lines within a page have similar compressibility

Thus, page-based last-time prediction of index can be accurate (94%)

SIMILAR INTRA-PAGE COMPRESSIBILITY

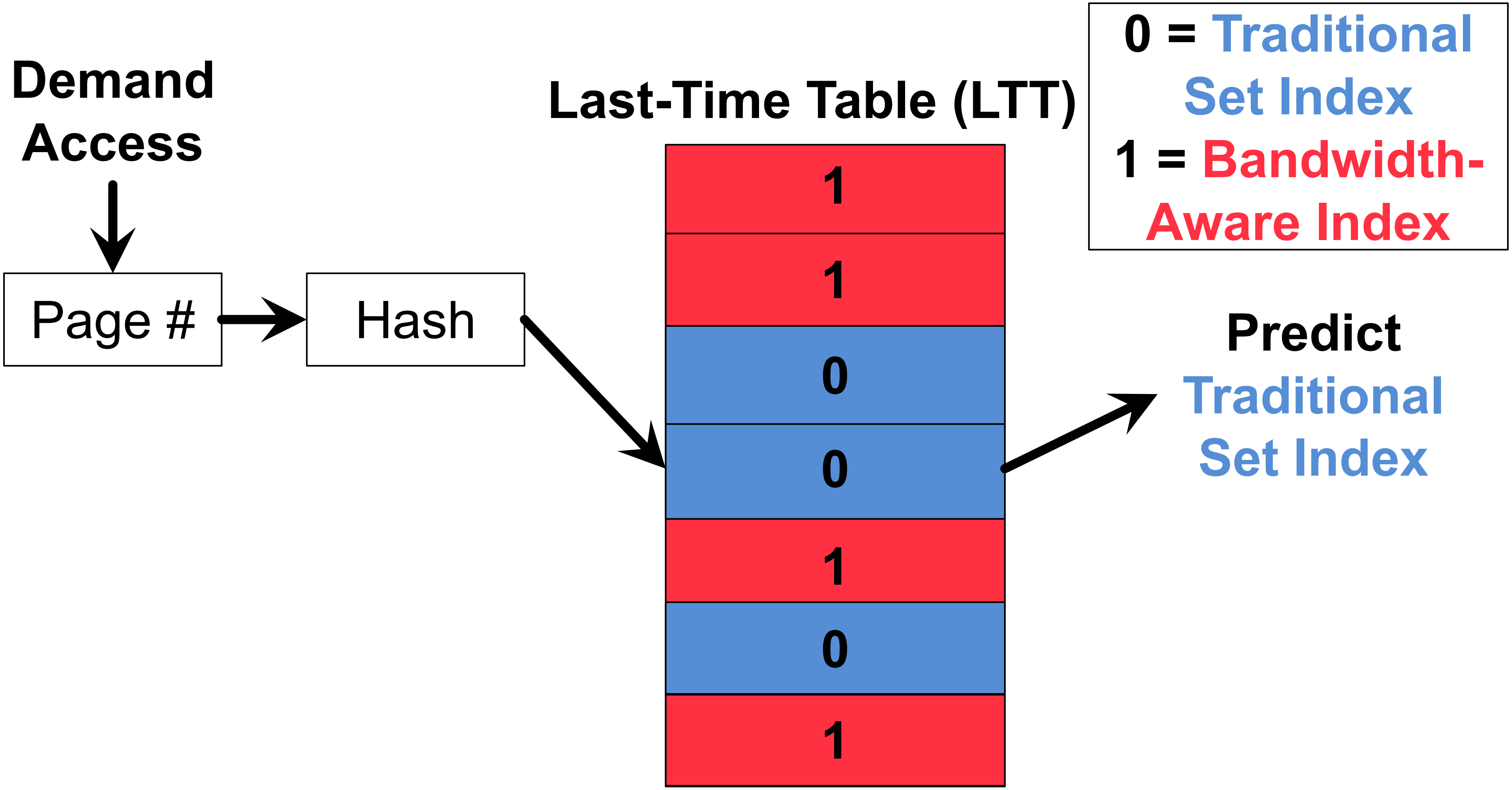


Lines within a page have similar compressibility

2nd access only on mispredict

Thus, page-based last-time prediction of index can be accurate (94%)

PAGE-BASED CACHE INDEX PREDICTOR (CIP)

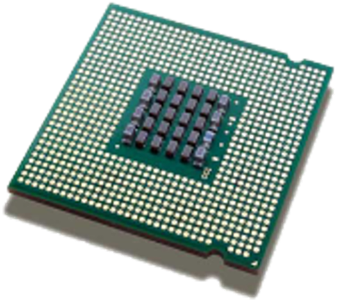


Page-based last-time prediction exploits similar intra-page compressibility, to achieve high prediction accuracy (94%)

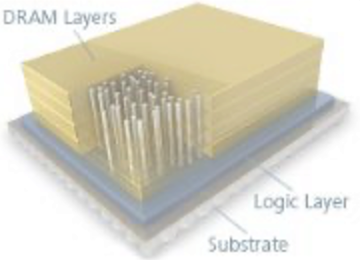
DICE OVERVIEW

- Compressed DRAM Cache Organization
- Flexible Mapping for Quick Switching
- Dynamic Indexing (DICE)
  - Insertion Policy
  - Index Prediction
- Results ⇦

METHODOLOGY (1/8TH KNIGHTS LANDING)



CPU

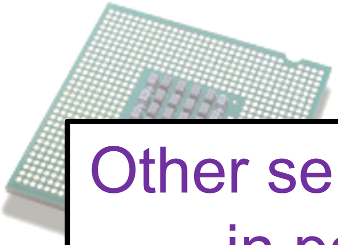


Stacked DRAM

Commodity DRAM

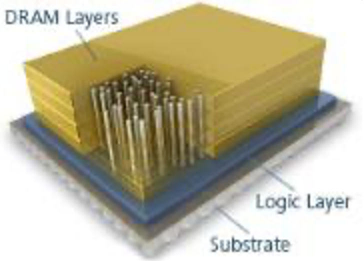
- Core Chip
  - 3.2GHz 4-wide out-of-order core
  - 8 cores, 8MB shared last-level cache
- Compression
  - FPC + BDI

METHODOLOGY (1/8TH KNIGHTS LANDING)



Other sensitivities in paper

CPU

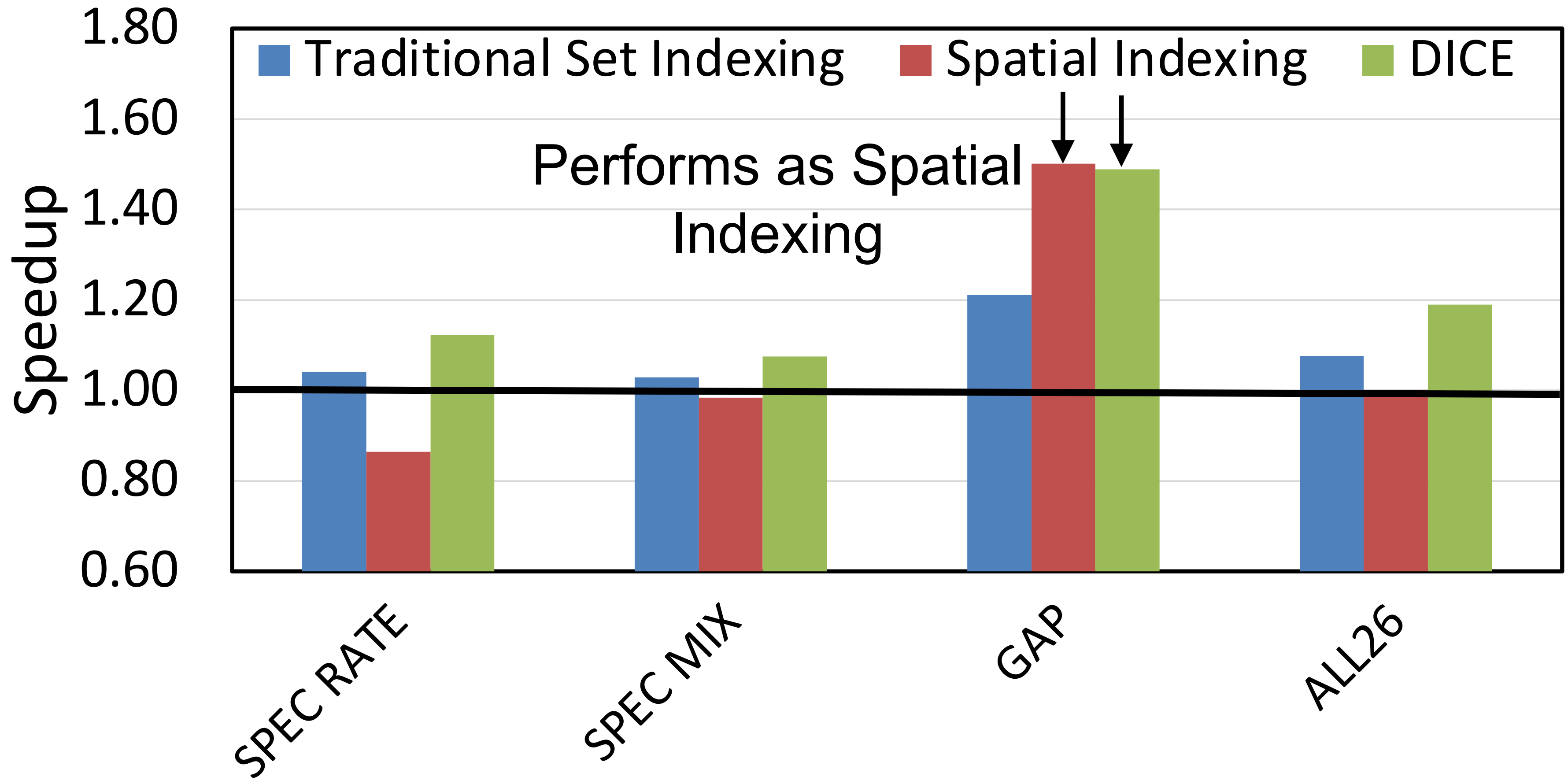


Stacked DRAM

Commodity DRAM

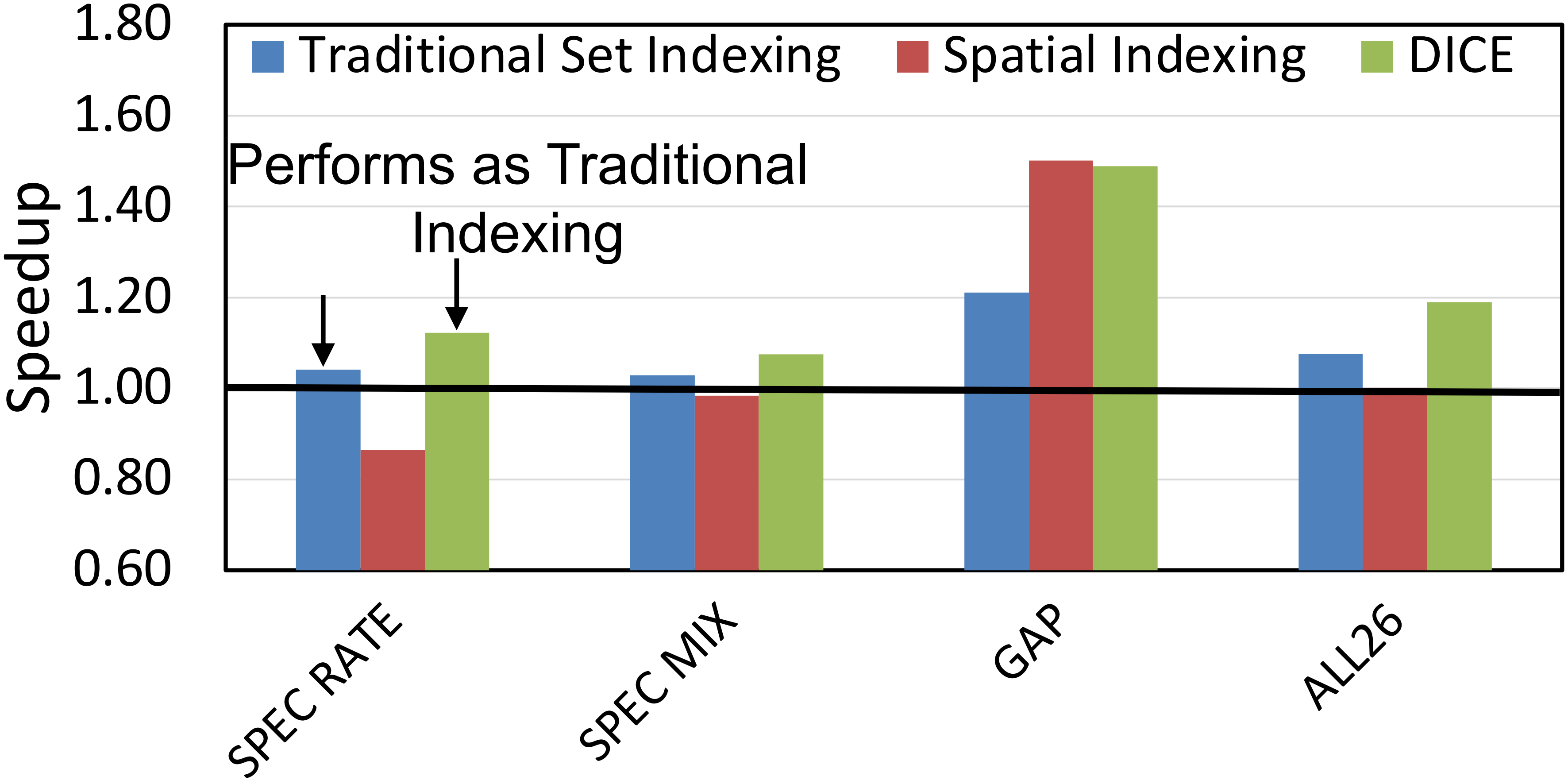
	Stacked DRAM	Commodity DRAM
Capacity	1GB	32GB
Bus	DDR1.6GHz, 128-bit	DDR1.6GHz, 64-bit
Channels	4 channels	1 channel
Bandwidth	100 GBps	12.5 GBps
Latency	35ns	35ns

DICE RESULTS



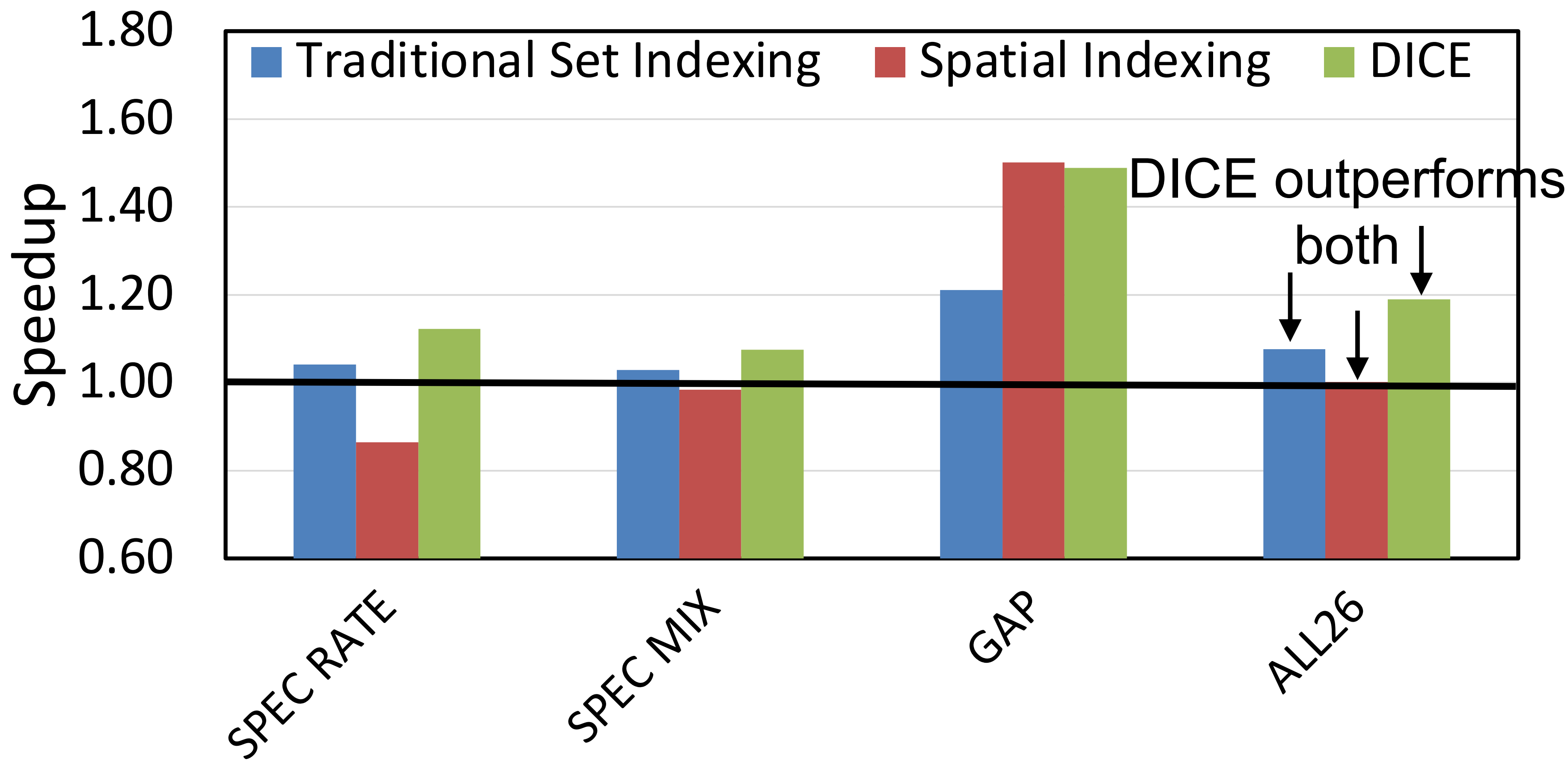
DICE improves performance over both Spatial Indexing and Traditional Indexing with fine-grain decision (19%)

DICE RESULTS



DICE improves performance over both Spatial Indexing and Traditional Indexing with fine-grain decision (19%)

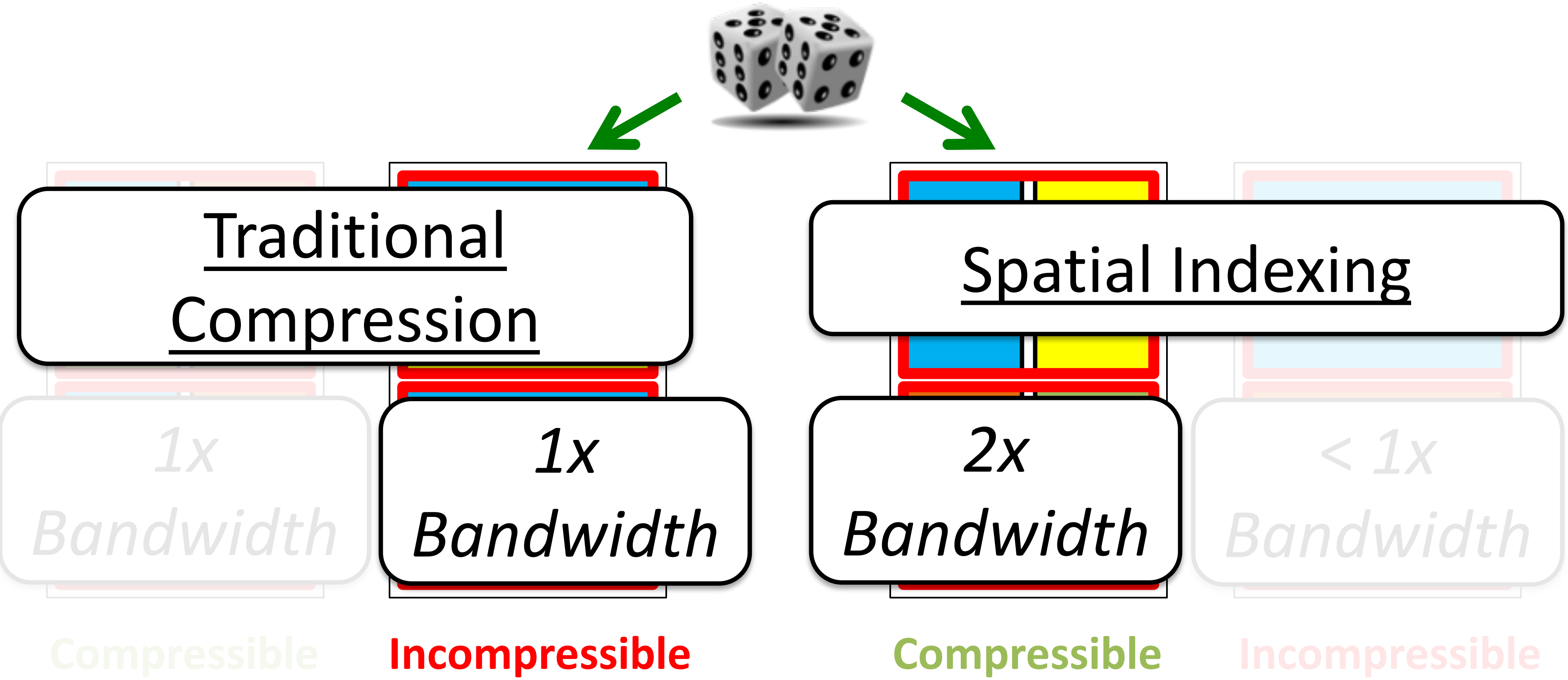
DICE RESULTS



DICE improves performance over both Spatial Indexing and Traditional Indexing with fine-grain decision (19%)

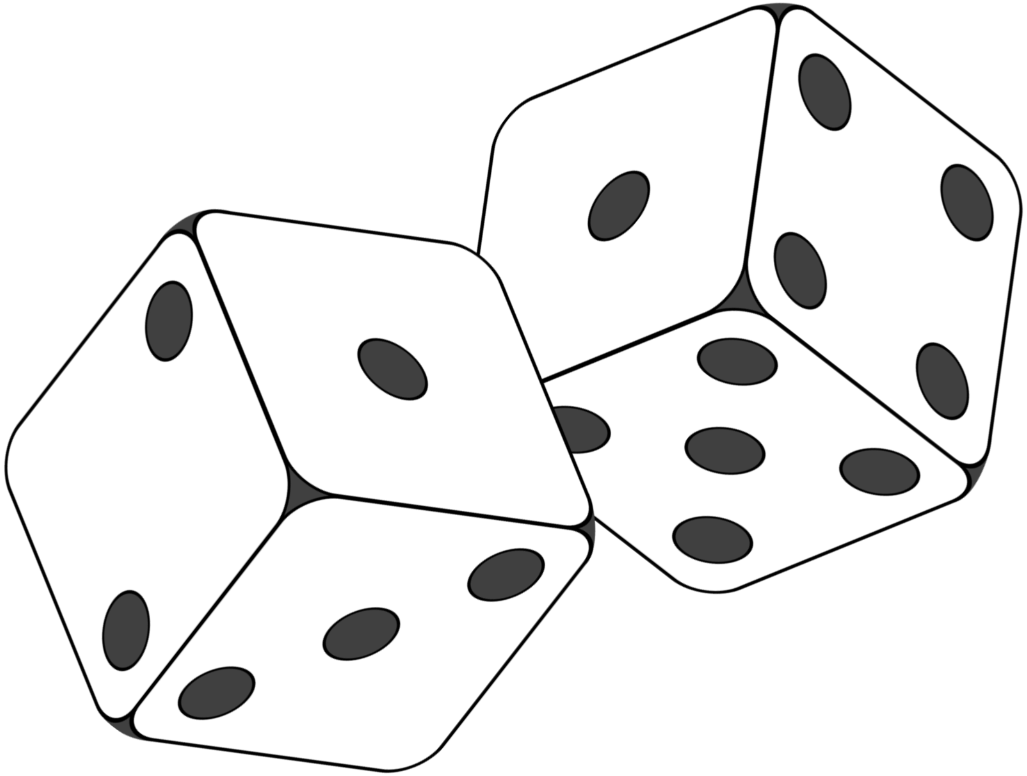
INTRODUCTION: COMPRESSED DRAM CACHE

Goal: Compression for Capacity **AND** Bandwidth



DICE (Dynamic Index) → 19% Speedup + 36% ⇩ EDP

THANK YOU



EXTRA SLIDES

- Extra Slides

DIFFERENT CACHE SENSITIVITIES

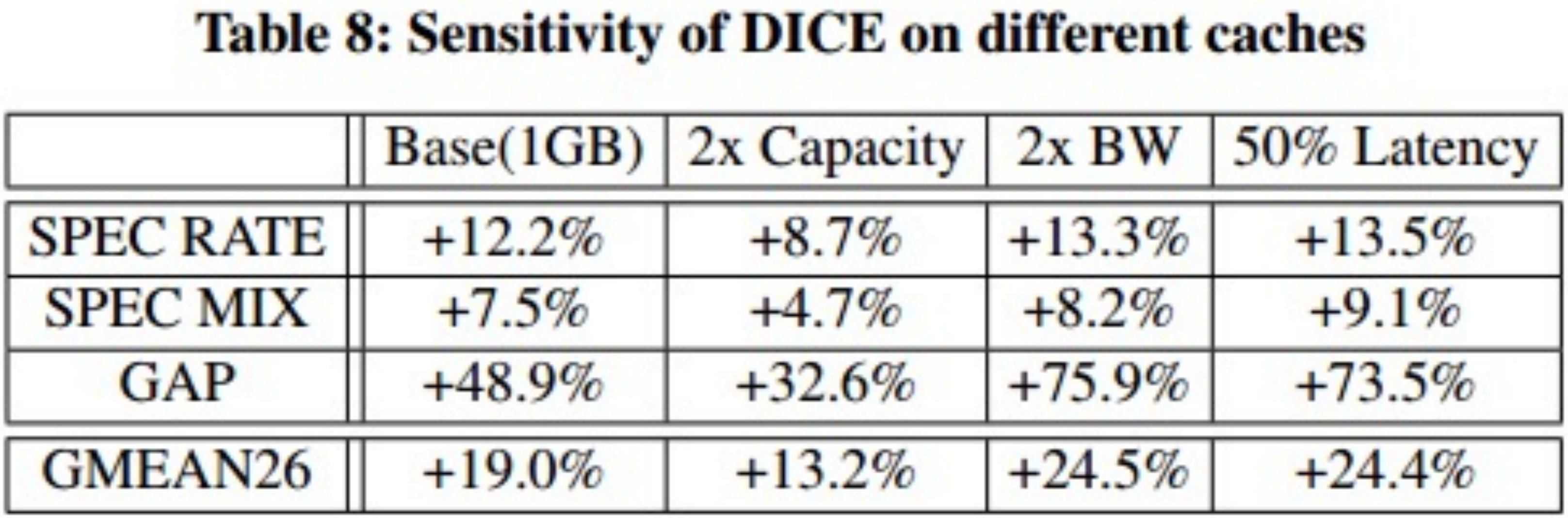


Table 8: Sensitivity of DICE on different caches

	Base(1GB)	2x Capacity	2x BW	50% Latency
SPEC RATE	+12.2%	+8.7%	+13.3%	+13.5%
SPEC MIX	+7.5%	+4.7%	+8.2%	+9.1%
GAP	+48.9%	+32.6%	+75.9%	+73.5%
GMEAN26	+19.0%	+13.2%	+24.5%	+24.4%

COMPARISON TO PREFETCH

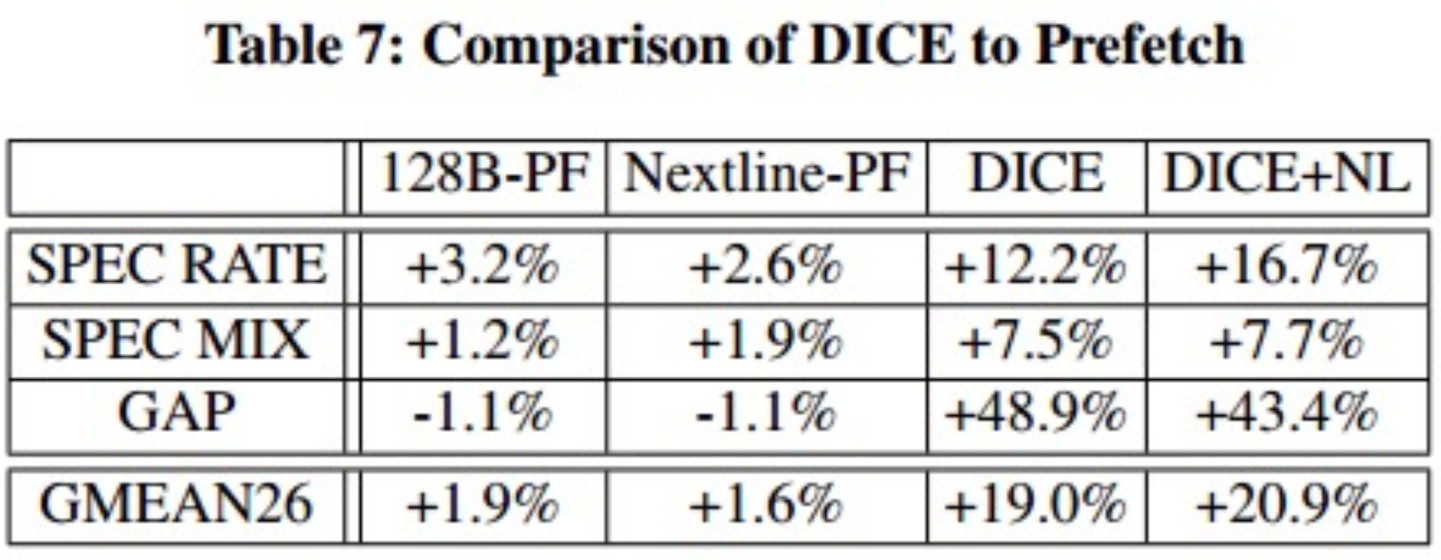


Table 7: Comparison of DICE to Prefetch

	128B-PF	Nextline-PF	DICE	DICE+NL
SPEC RATE	+3.2%	+2.6%	+12.2%	+16.7%
SPEC MIX	+1.2%	+1.9%	+7.5%	+7.7%
GAP	-1.1%	-1.1%	+48.9%	+43.4%
GMEAN26	+1.9%	+1.6%	+19.0%	+20.9%

COMPARISON TO SRAM /MEMORY COMPRESSION

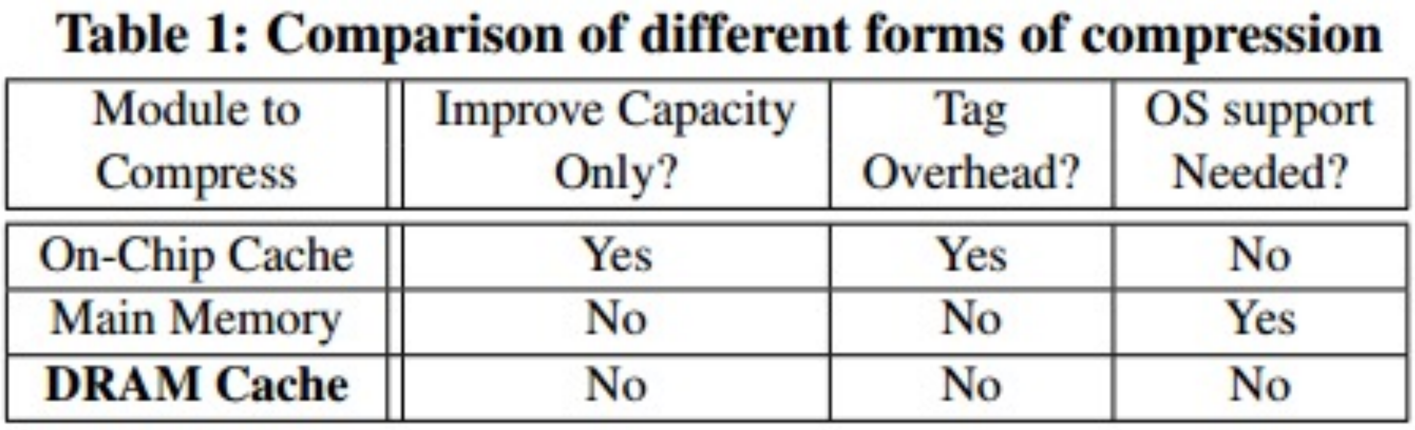
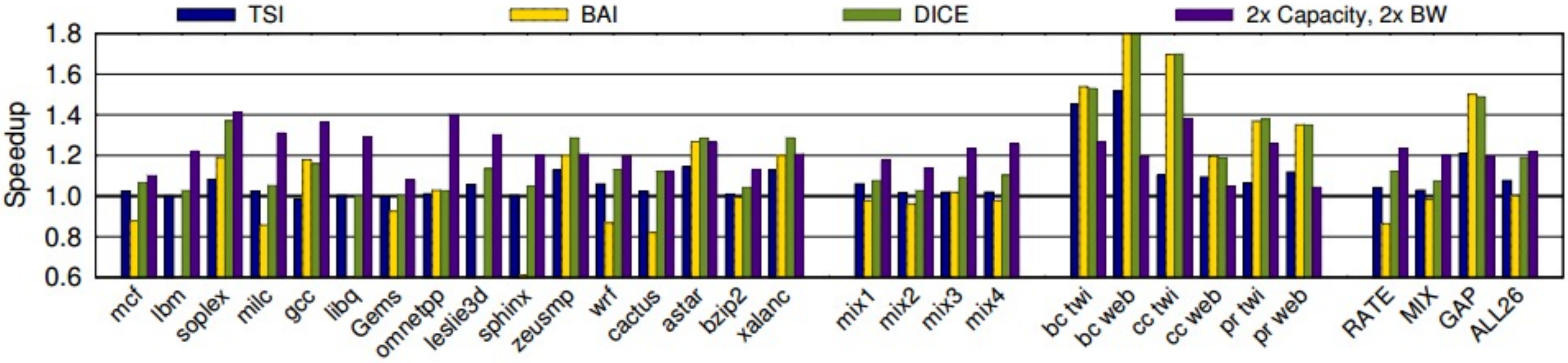


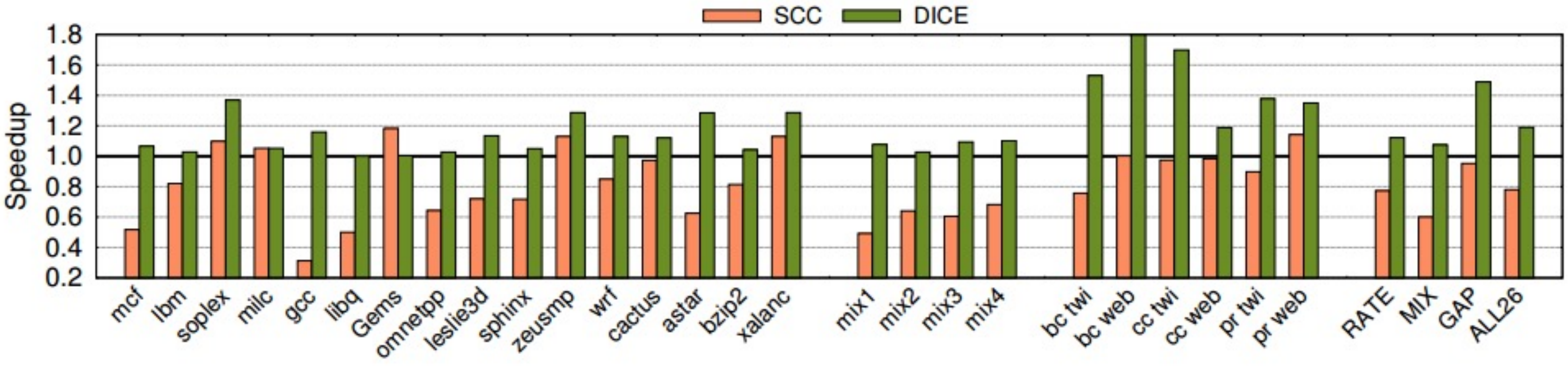
Table 1: Comparison of different forms of compression

Module to Compress	Improve Capacity Only?	Tag Overhead?	OS support Needed?
On-Chip Cache	Yes	Yes	No
Main Memory	No	No	Yes
DRAM Cache	No	No	No

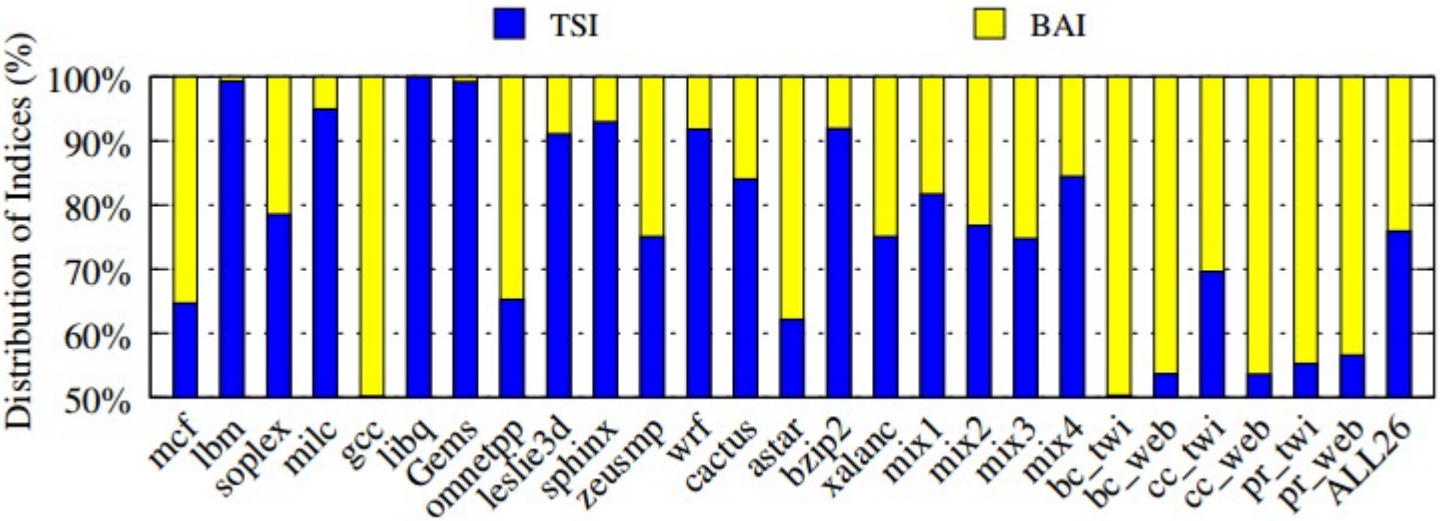
FULL RESULTS (MIXED COMPRESSIBILITY)



SRAM CACHE COMPRESSION ON DRAM CACHE



DISTRIBUTION FOR INDEX DECISION



DICE INSERTION THRESHOLD

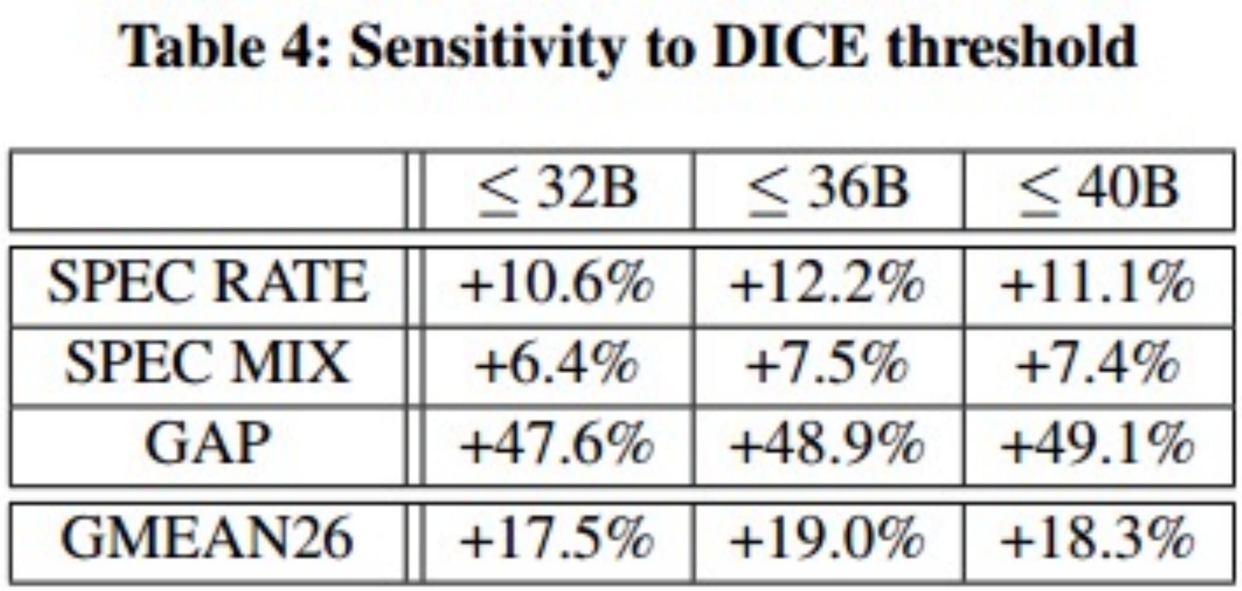


Table 4: Sensitivity to DICE threshold

	$\leq 32B$	$\leq 36B$	$\leq 40B$
SPEC RATE	+10.6%	+12.2%	+11.1%
SPEC MIX	+6.4%	+7.5%	+7.4%
GAP	+47.6%	+48.9%	+49.1%
GMEAN26	+17.5%	+19.0%	+18.3%

EFFECTIVE CAPACITY

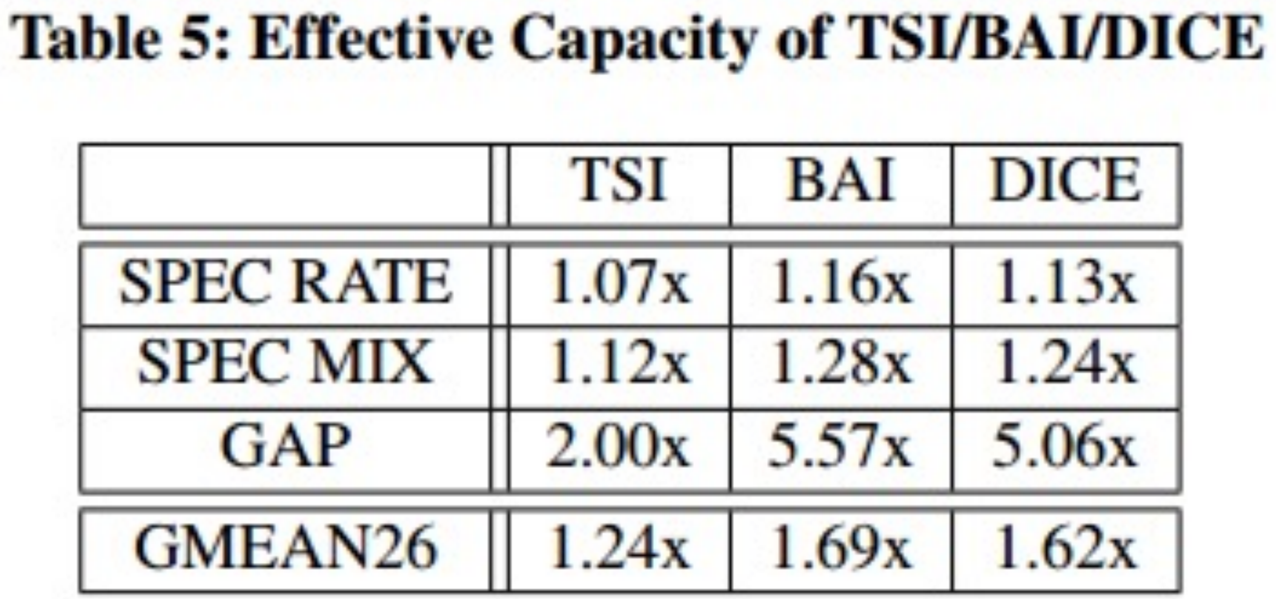


Table 5: Effective Capacity of TSI/BAI/DICE

	TSI	BAI	DICE
SPEC RATE	1.07x	1.16x	1.13x
SPEC MIX	1.12x	1.28x	1.24x
GAP	2.00x	5.57x	5.06x
GMEAN26	1.24x	1.69x	1.62x

L3 HIT RATE IMPROVEMENT

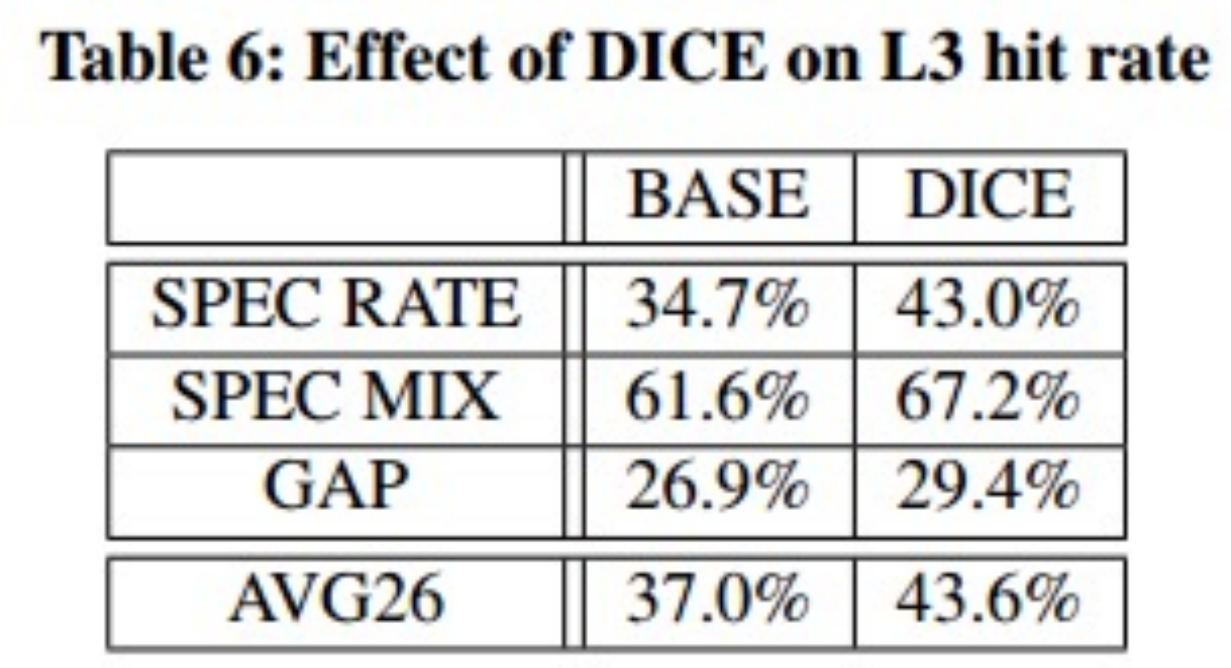
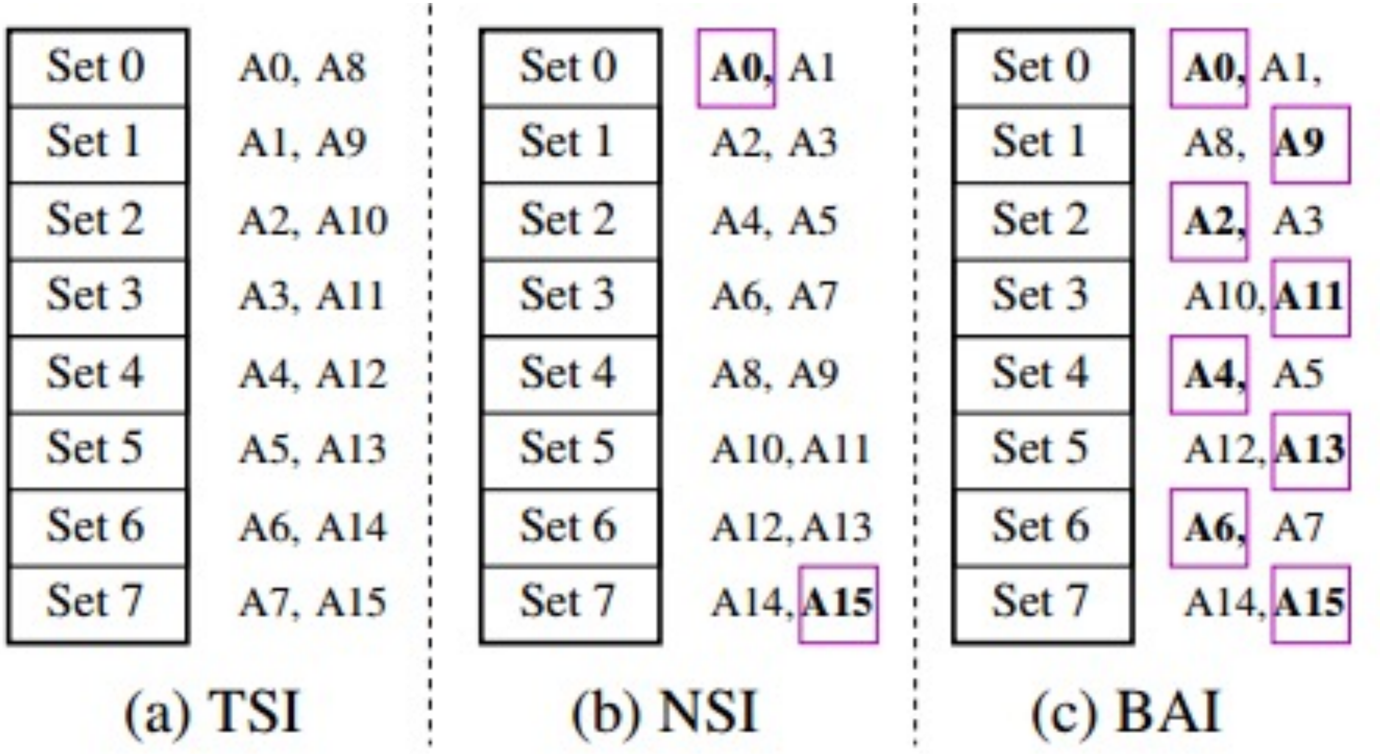


Table 6: Effect of DICE on L3 hit rate

	BASE	DICE
SPEC RATE	34.7%	43.0%
SPEC MIX	61.6%	67.2%
GAP	26.9%	29.4%
AVG26	37.0%	43.6%

LARGER TSI VS. BAI EXAMPLE



Set 0	A0, A8
Set 1	A1, A9
Set 2	A2, A10
Set 3	A3, A11
Set 4	A4, A12
Set 5	A5, A13
Set 6	A6, A14
Set 7	A7, A15

(a) TSI

Set 0	**A0,** A1
Set 1	A2, A3
Set 2	A4, A5
Set 3	A6, A7
Set 4	A8, A9
Set 5	A10, A11
Set 6	A12, A13
Set 7	A14, **A15**

(b) NSI

Set 0	**A0,** A1,
Set 1	A8, **A9**
Set 2	**A2,** A3
Set 3	A10, **A11**
Set 4	**A4,** A5
Set 5	A12, **A13**
Set 6	**A6,** A7
Set 7	A14, **A15**

(c) BAI