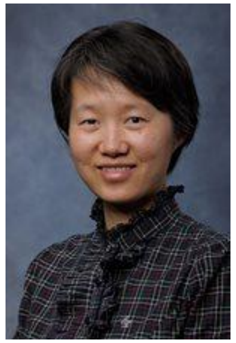
PhD Thesis Defense

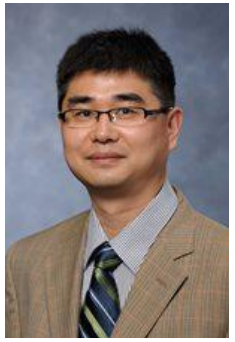
Jul 14, 2017 (Friday)

Addressing Prolonged Restore Challenges in Further Scaling DRAMs

Xianwei Zhang

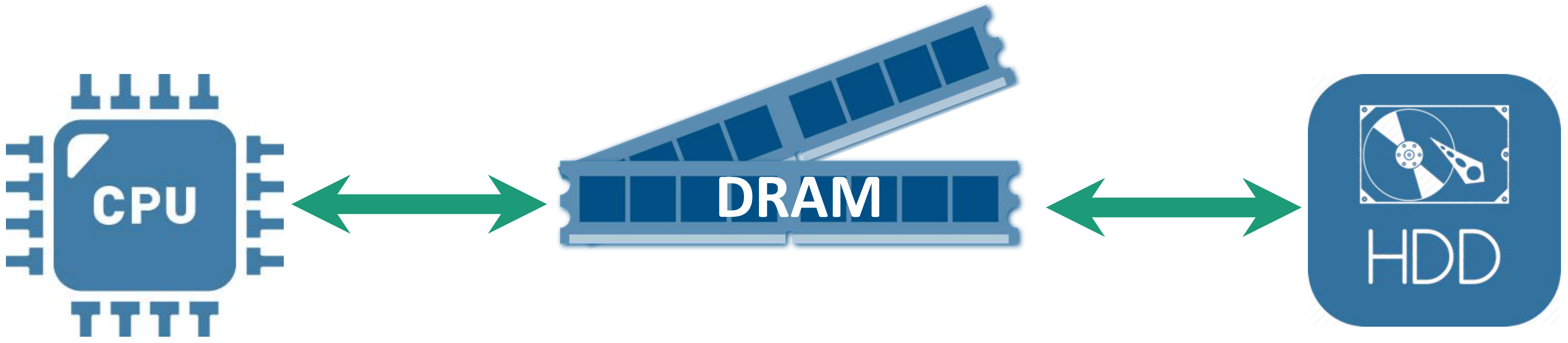
Committees:





Youtao Zhang (advisor)	Bruce R. Childers	Jun Yang	Wonsun Ahn	Guangyong Li
CS, Pitt	CS, Pitt	ECE, Pitt	CS, Pitt	ECE, Pitt

MAIN MEMORY

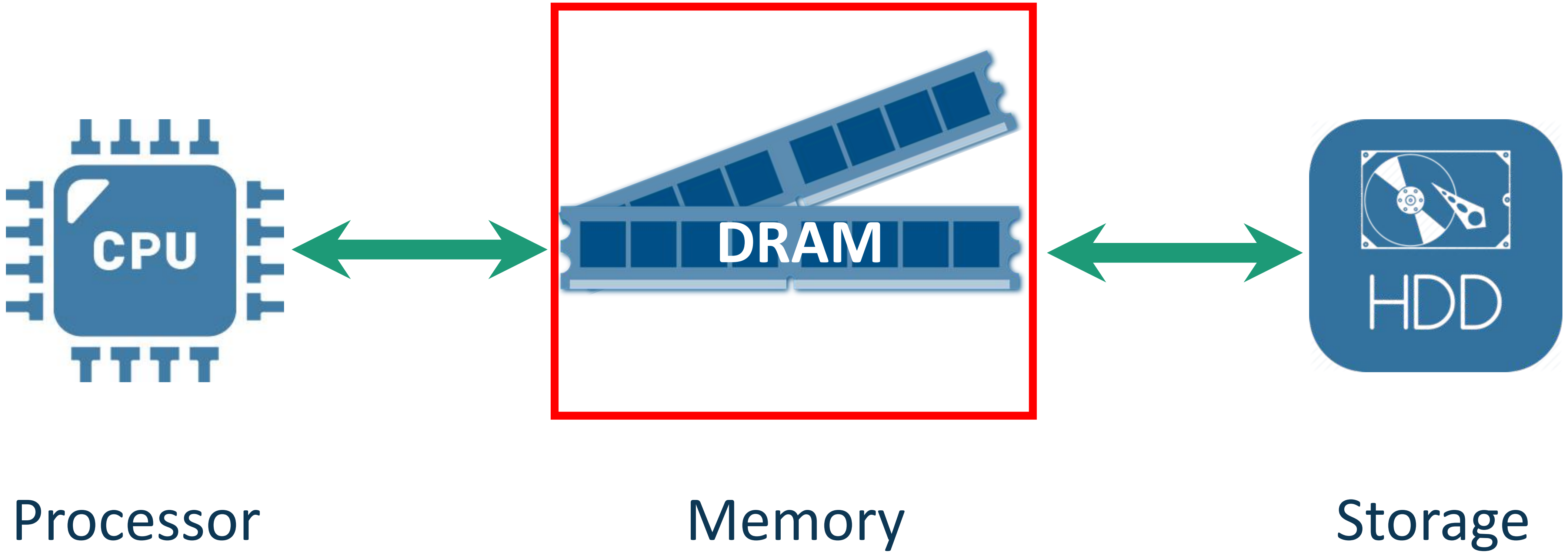


Processor

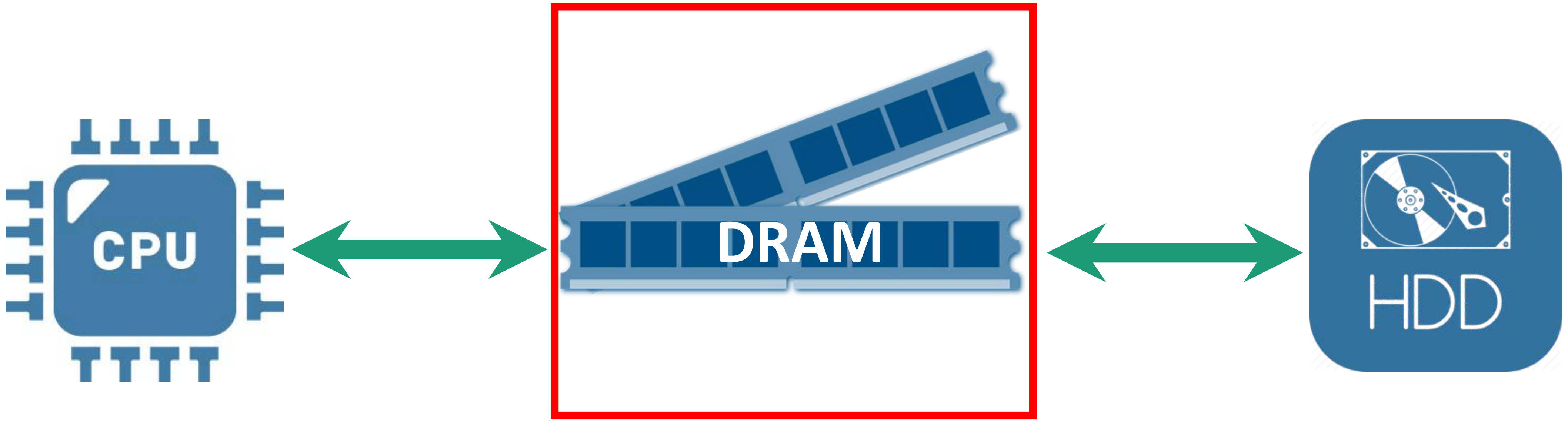
Memory

Storage

MAIN MEMORY



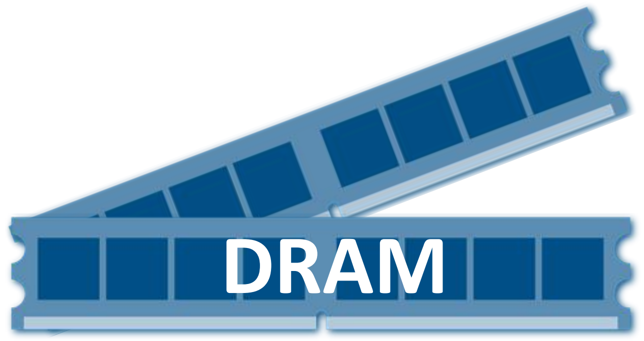
MAIN MEMORY



Processor Memory Storage

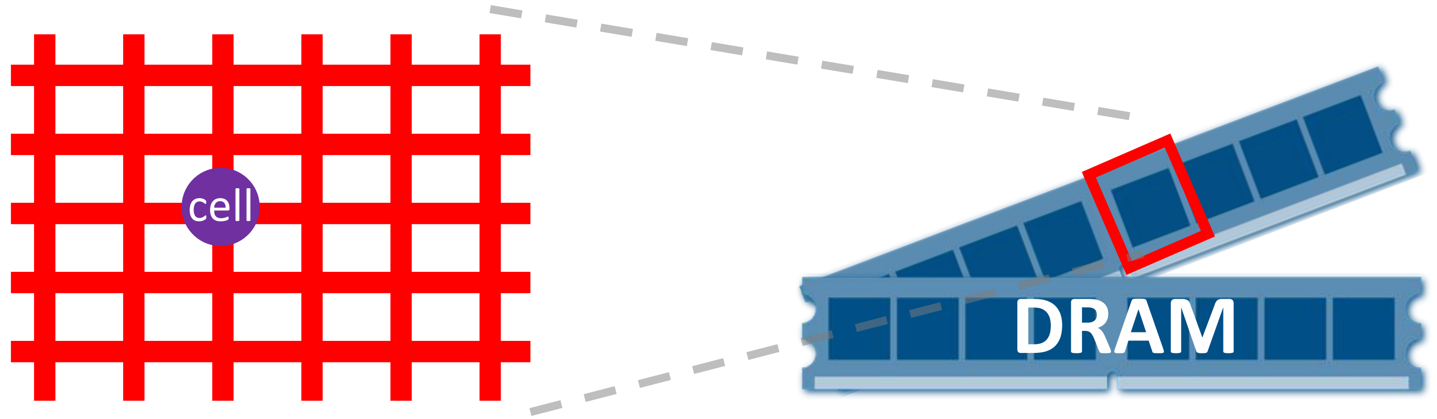
Main memory is critical for system performance

DRAM



DIMM/Chip

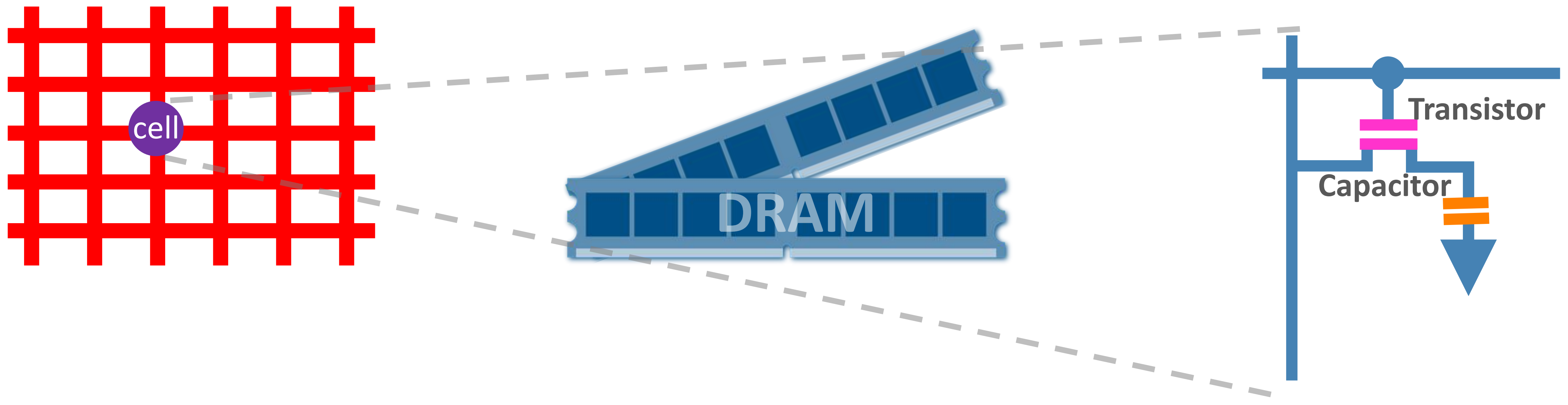
DRAM



2D Array

DIMM/Chip

DRAM

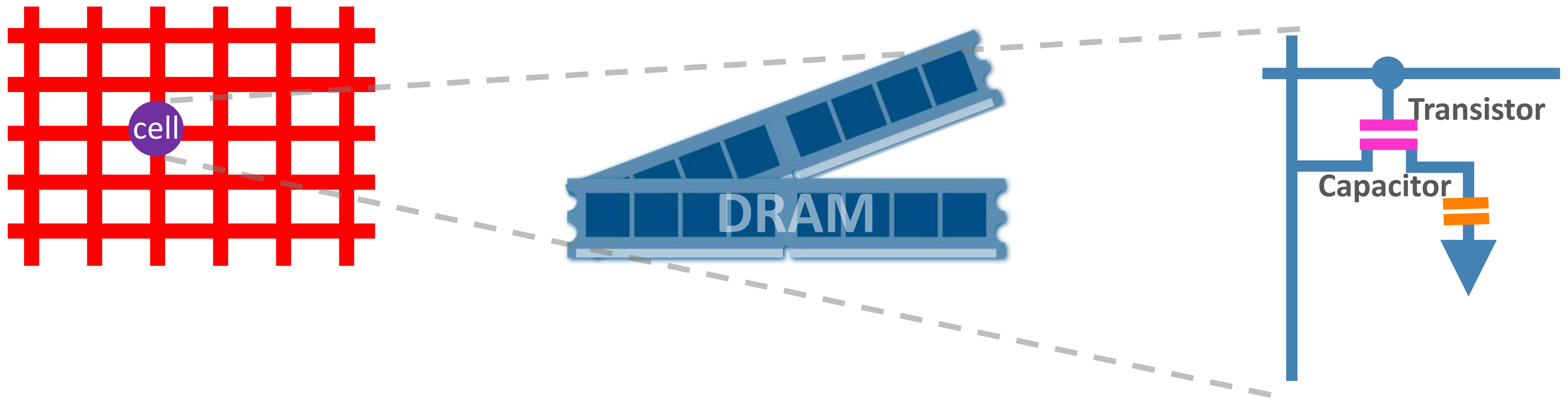


2D Array

DIMM/Chip

DRAM Cell

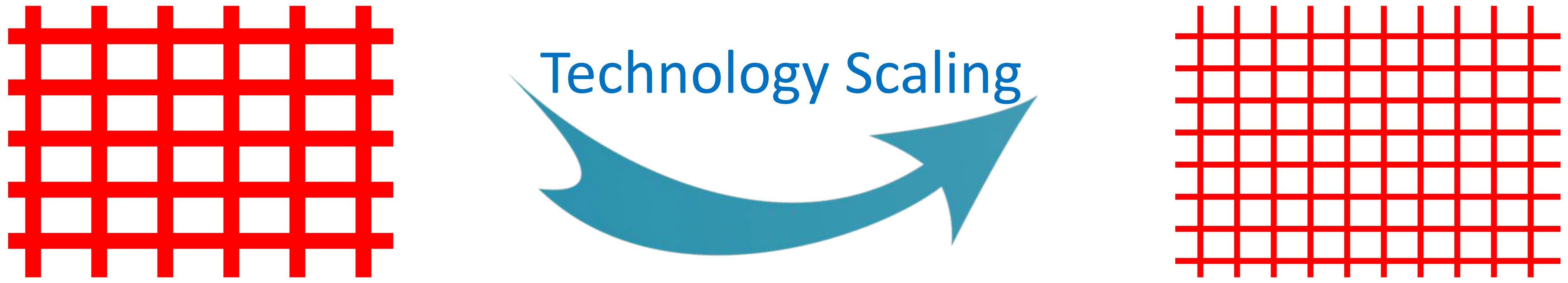
DRAM



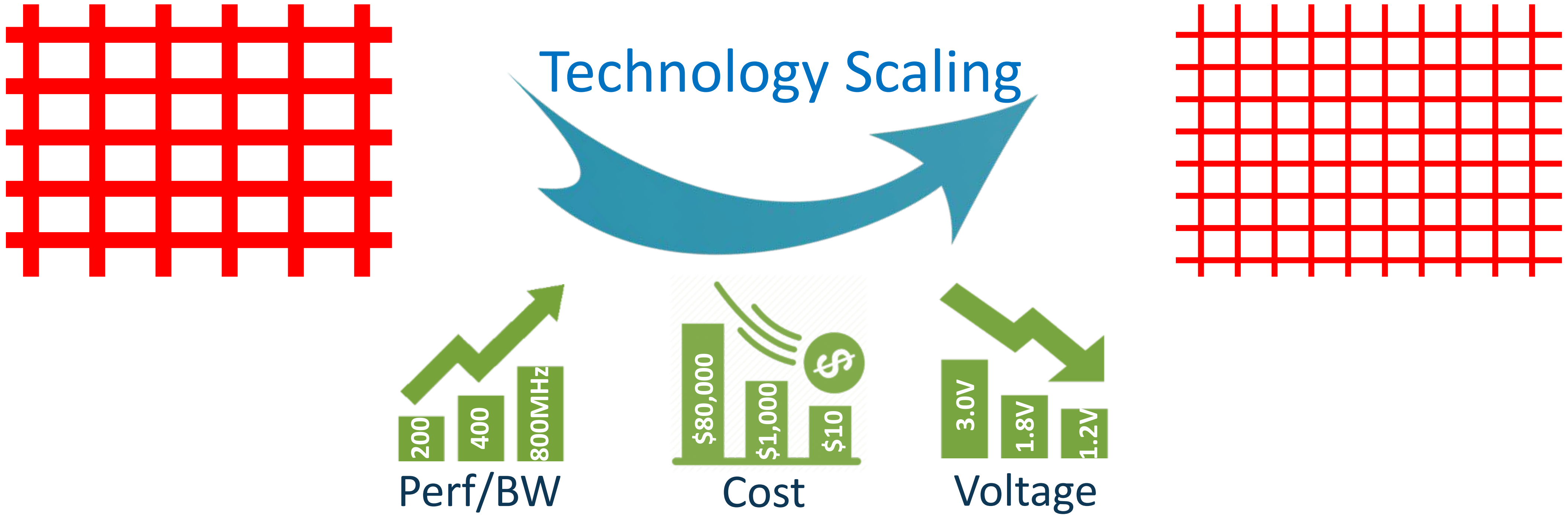
2D Array DIMM/Chip DRAM Cell

The simplicity enabled DRAM to continuously scale

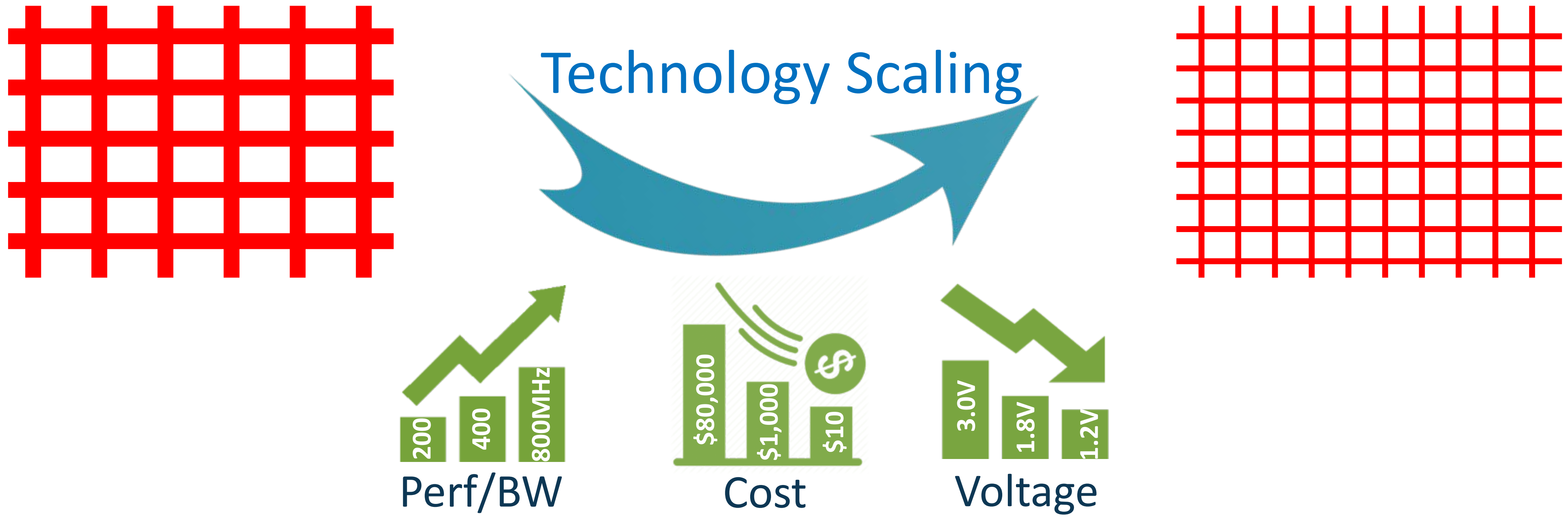
SCALING



SCALING


Technology Scaling
200
400
800MHz
Perf/BW
$80,000
$1,000
$10
Cost
3.0V
1.8V
1.2V
Voltage

SCALING



Do we still need DRAM to continue scale?

DEMANDS

Increasing Computation

DEMANDS

Increasing Computation

Data Intensive Apps

DEMANDS

amazon DynamoDB

redis

MEMCACHED

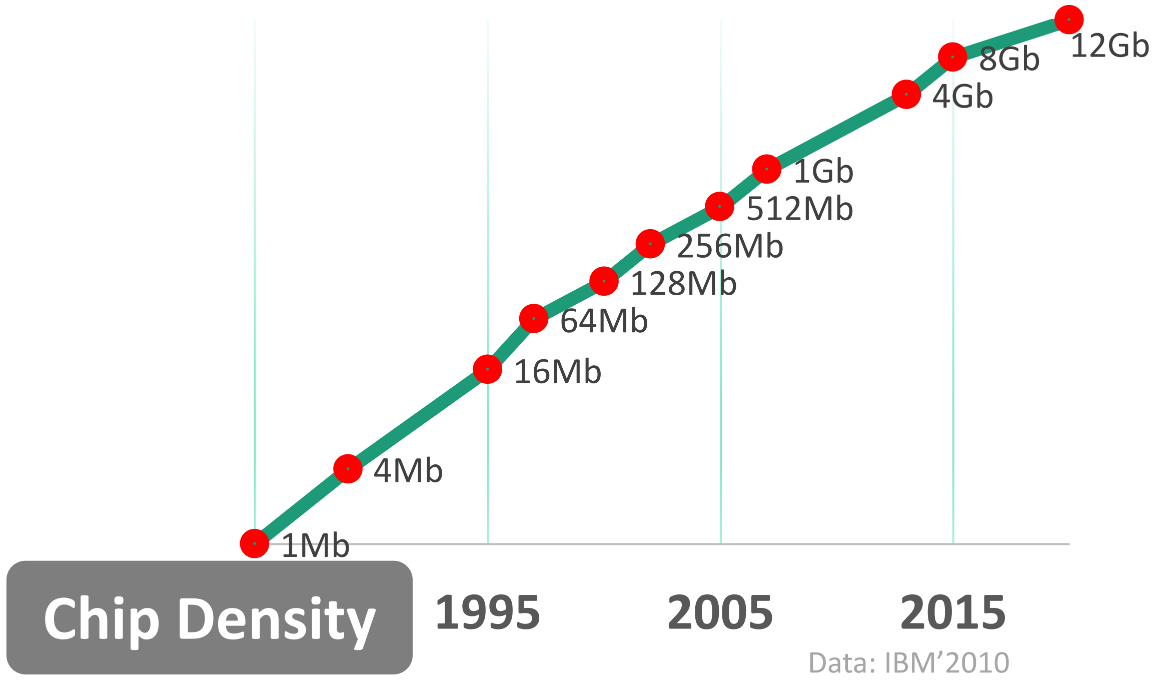
Increasing Computation   Data Intensive Apps   Tight Power Budgets

DEMANDS

Increasing Computation   Data Intensive Apps   Tight Power Budgets

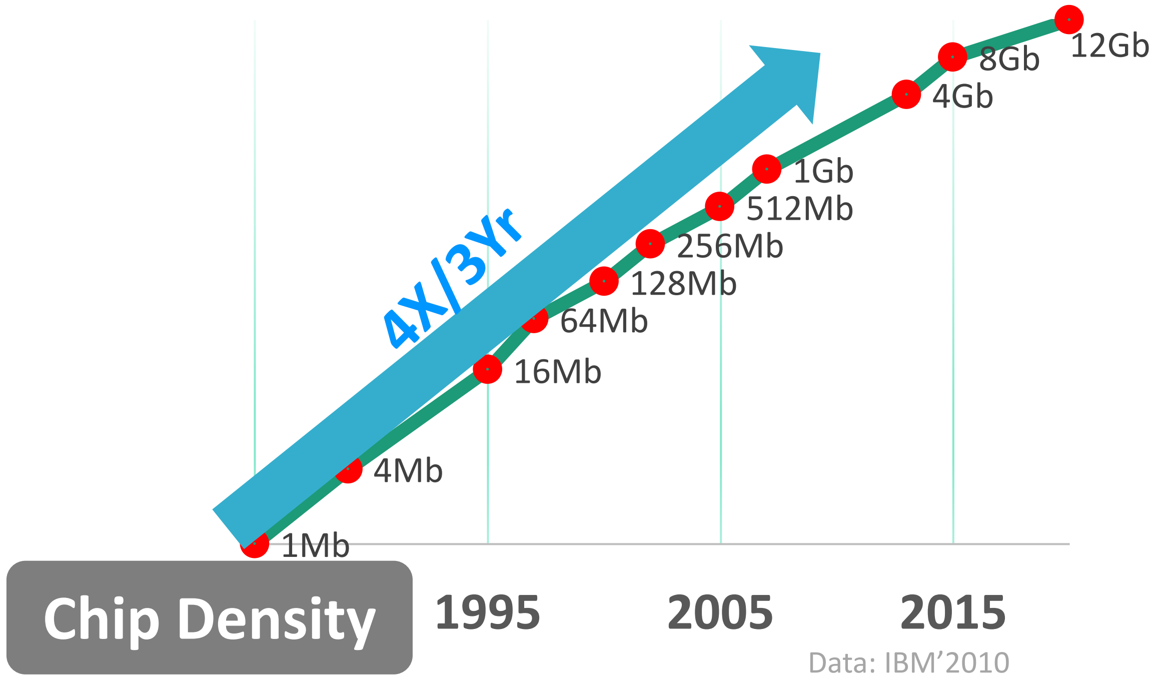
DRAM must keep scaling to meet demands

SCALING TREND



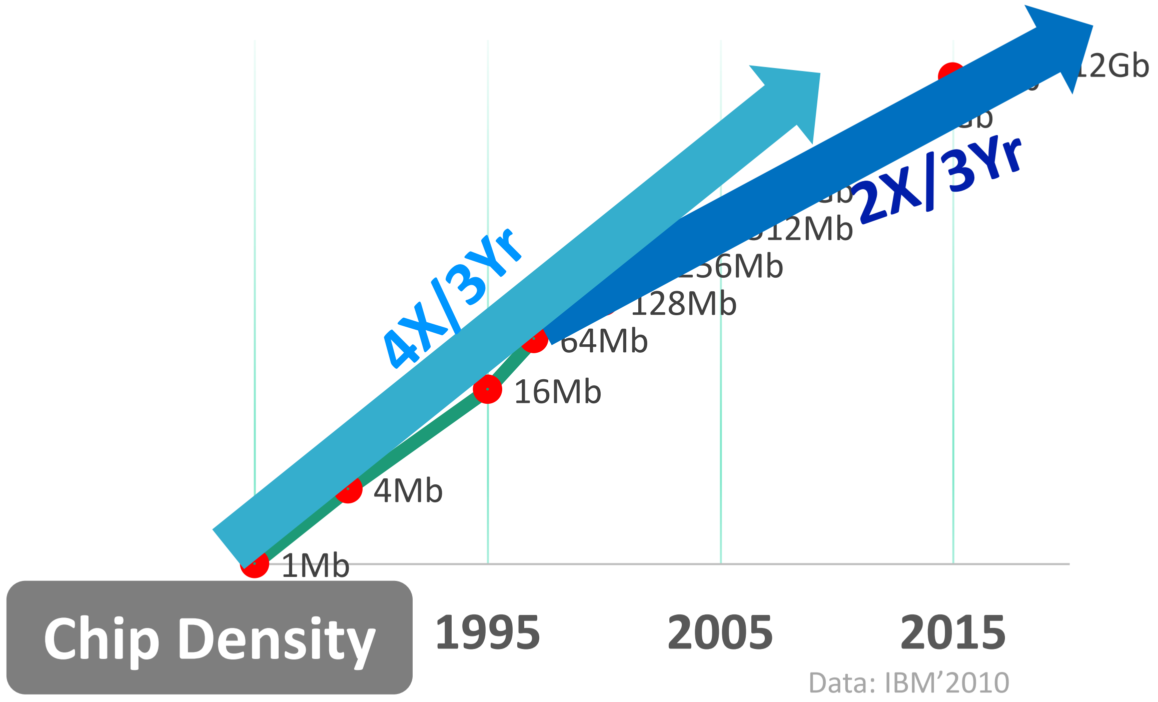
DRAM scaling is getting more difficult

SCALING TREND



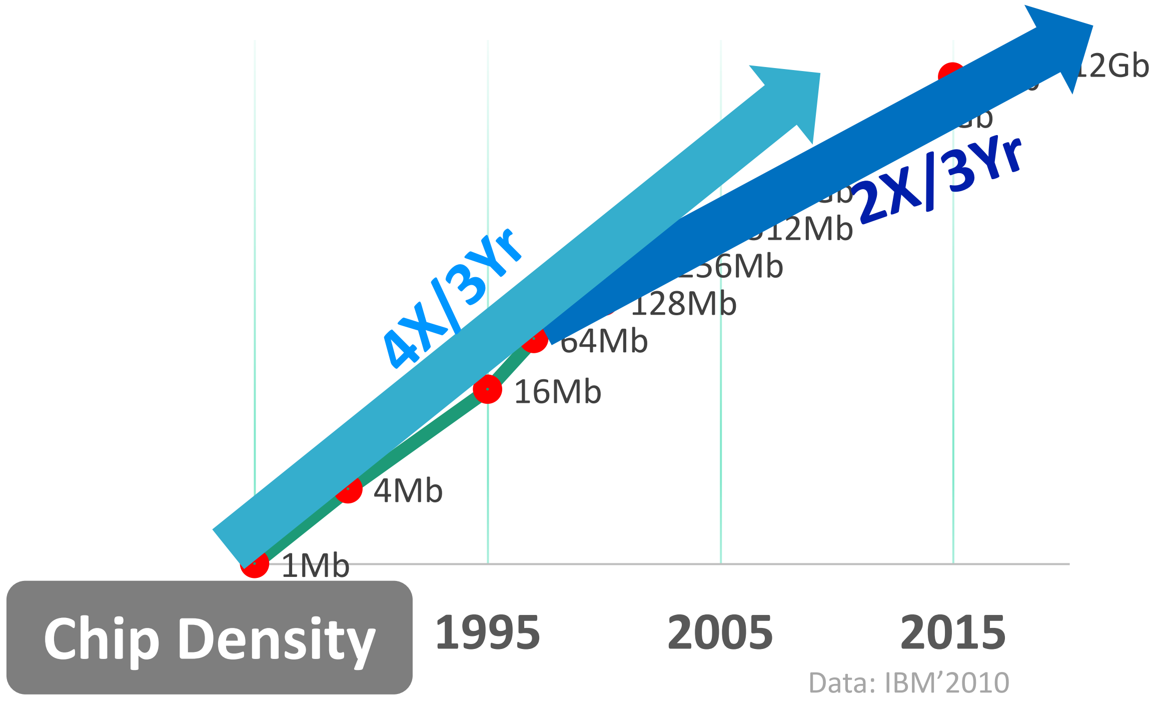
DRAM scaling is getting more difficult

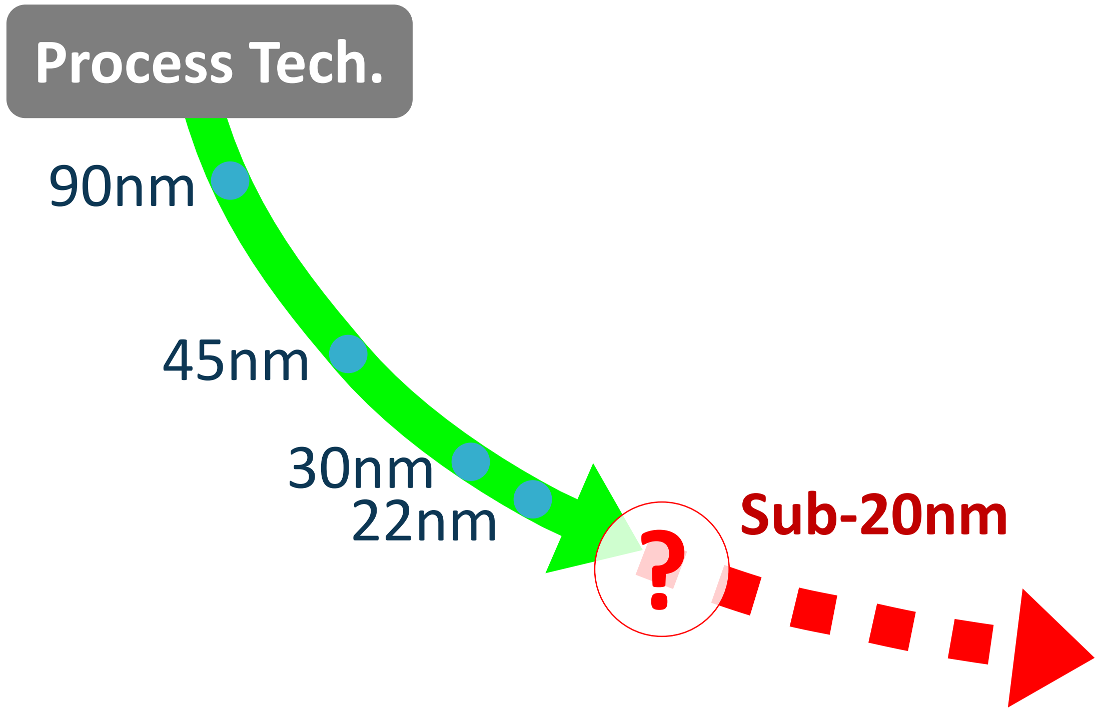
SCALING TREND



DRAM scaling is getting more difficult

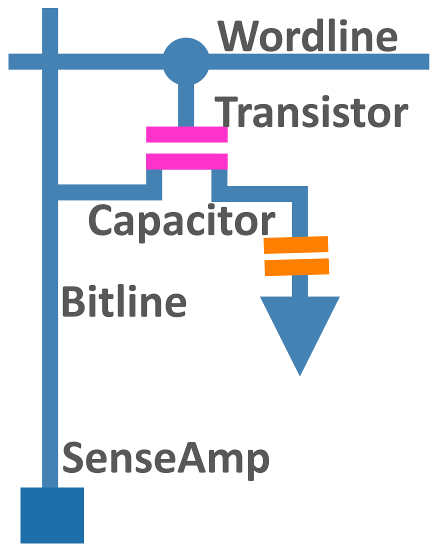
SCALING TREND



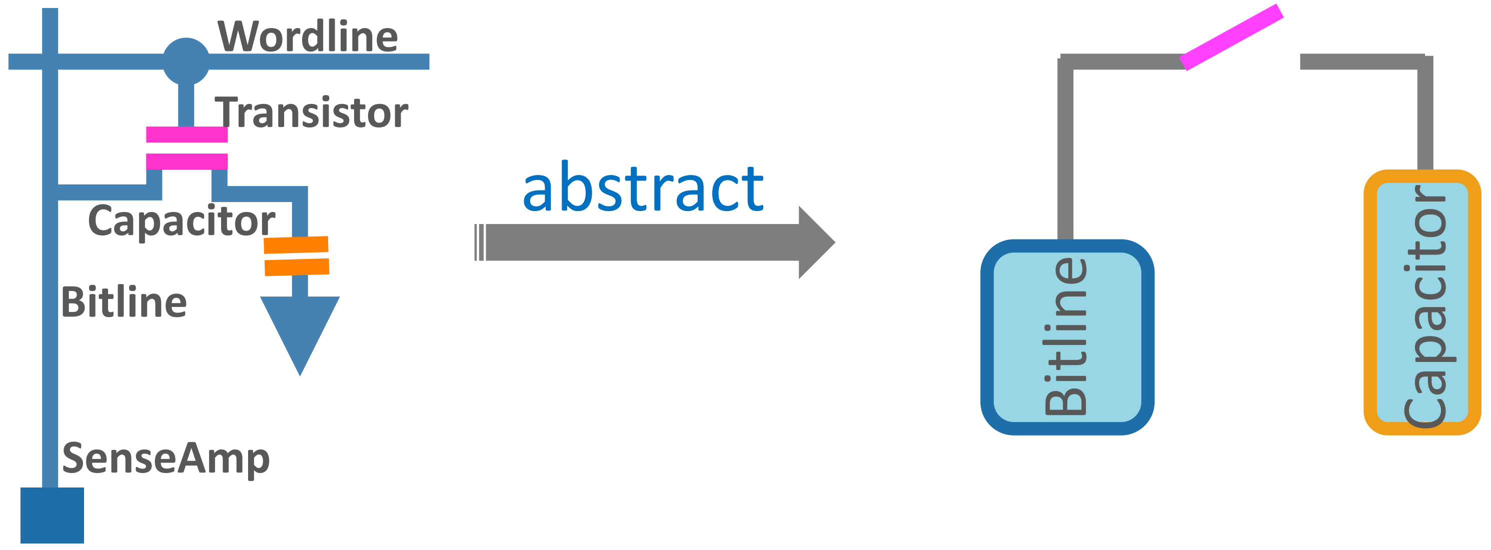


DRAM scaling is getting more difficult

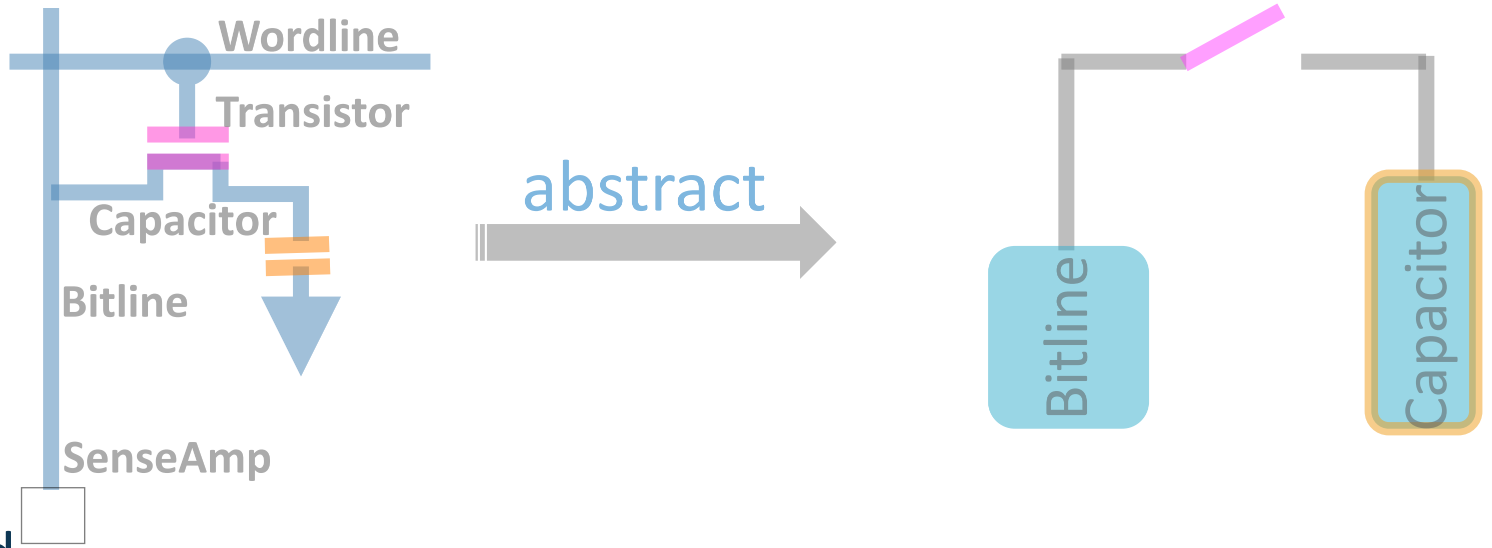
DRAM OPERATIONS



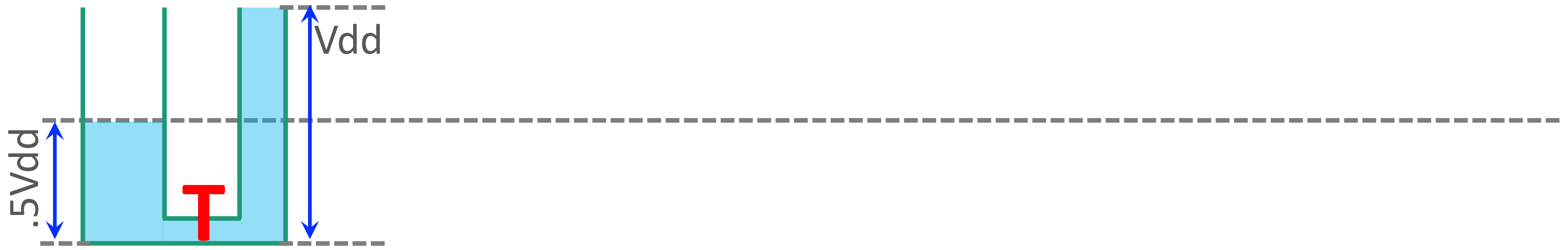
DRAM OPERATIONS



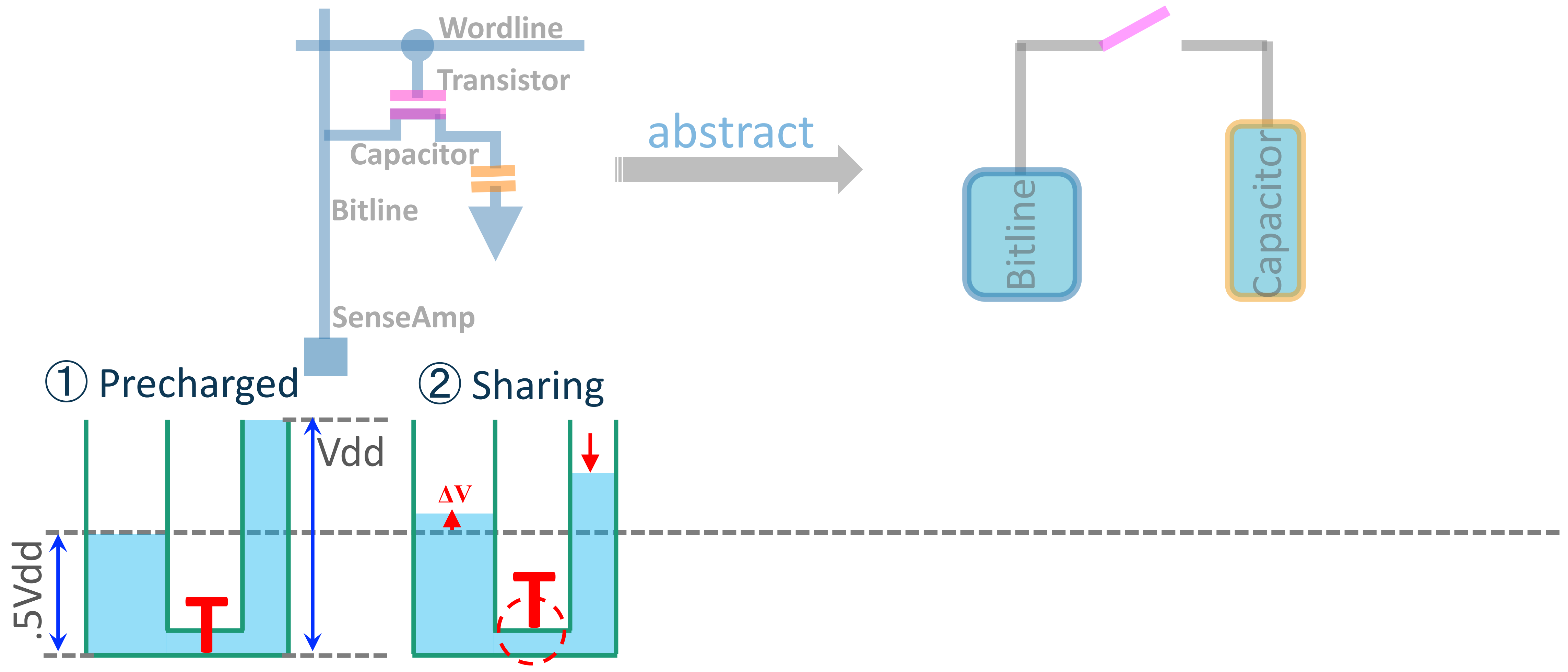
DRAM OPERATIONS



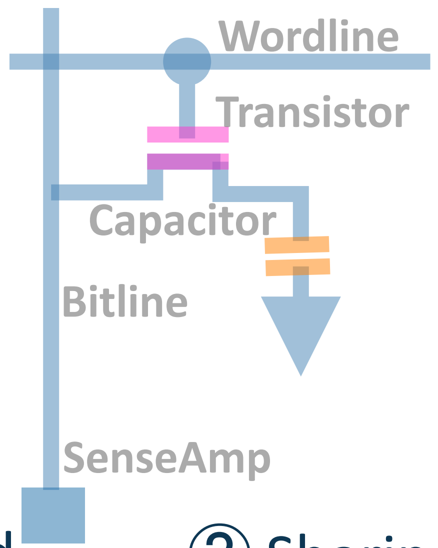
① Precharged

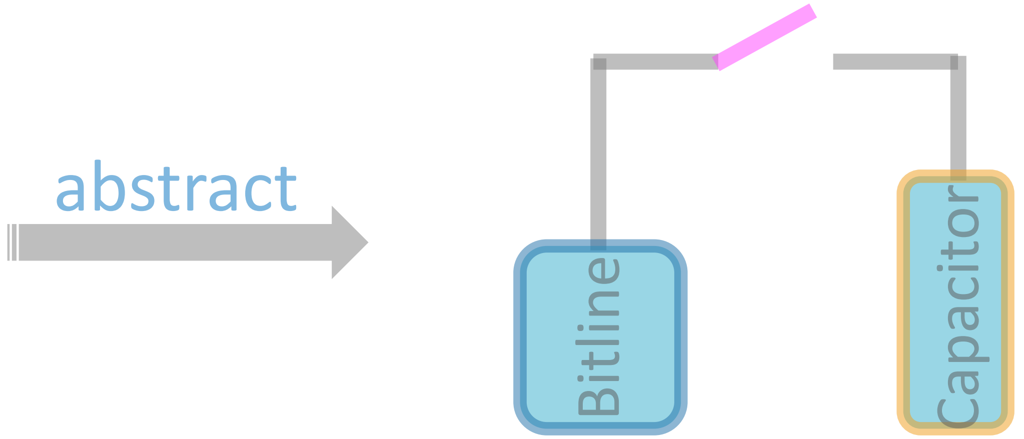


DRAM OPERATIONS

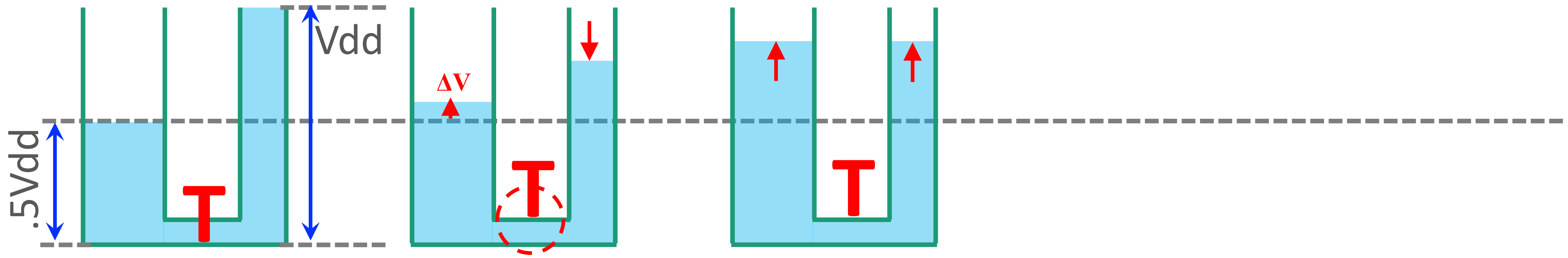


DRAM OPERATIONS

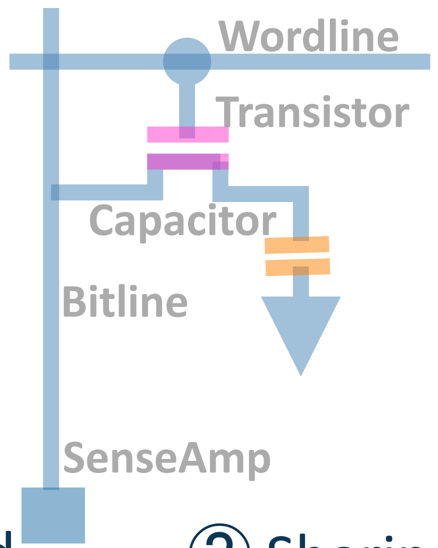


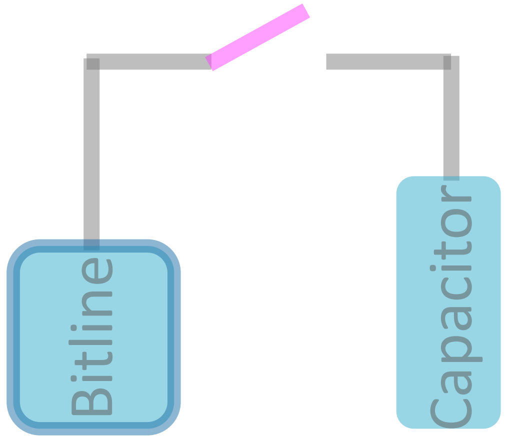


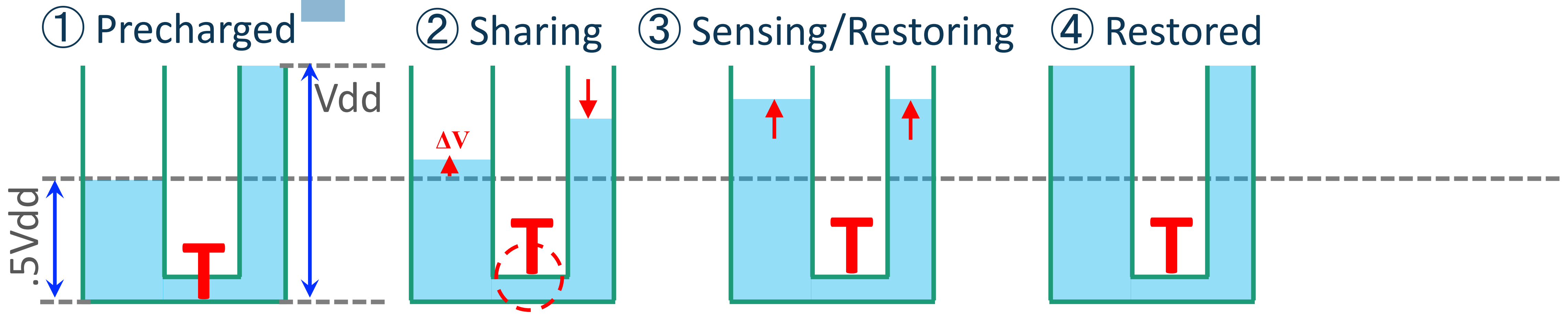
① Precharged ② Sharing ③ Sensing/Restoring



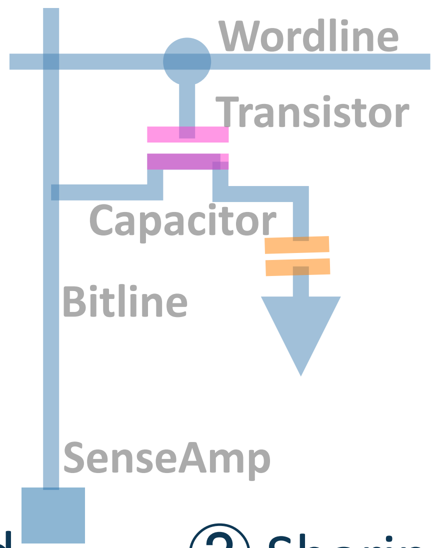
DRAM OPERATIONS

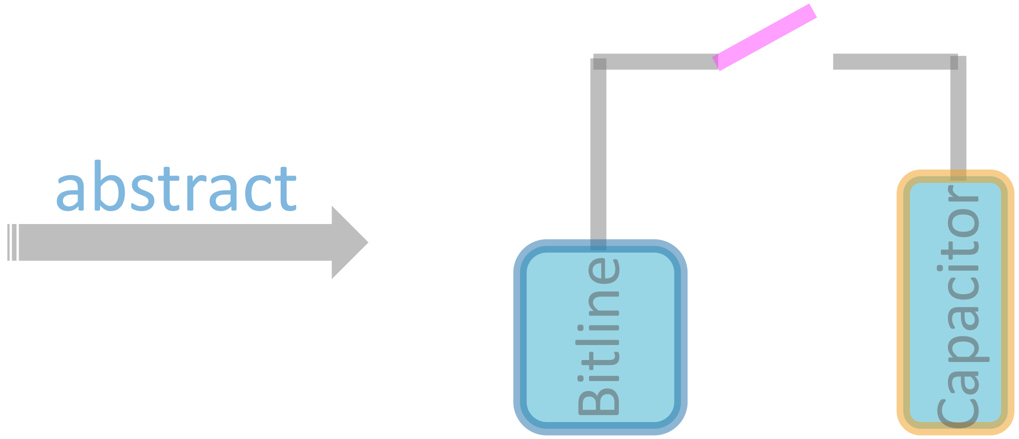


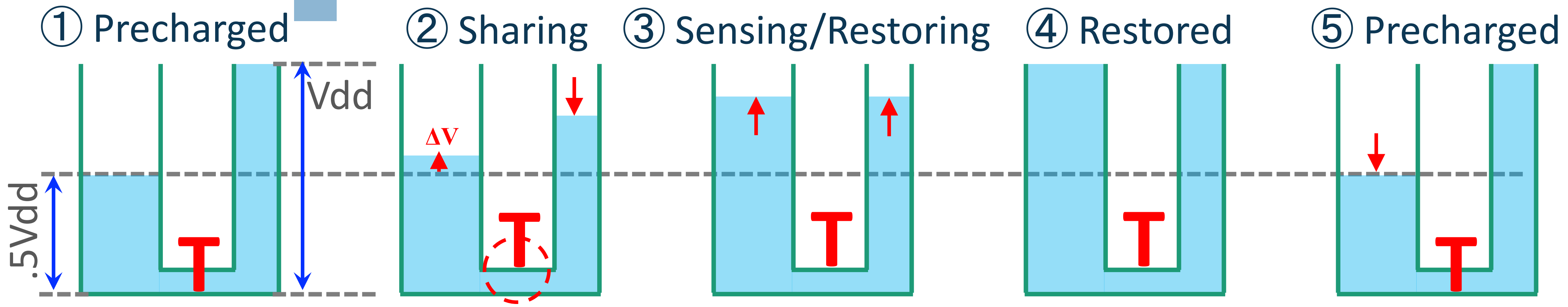




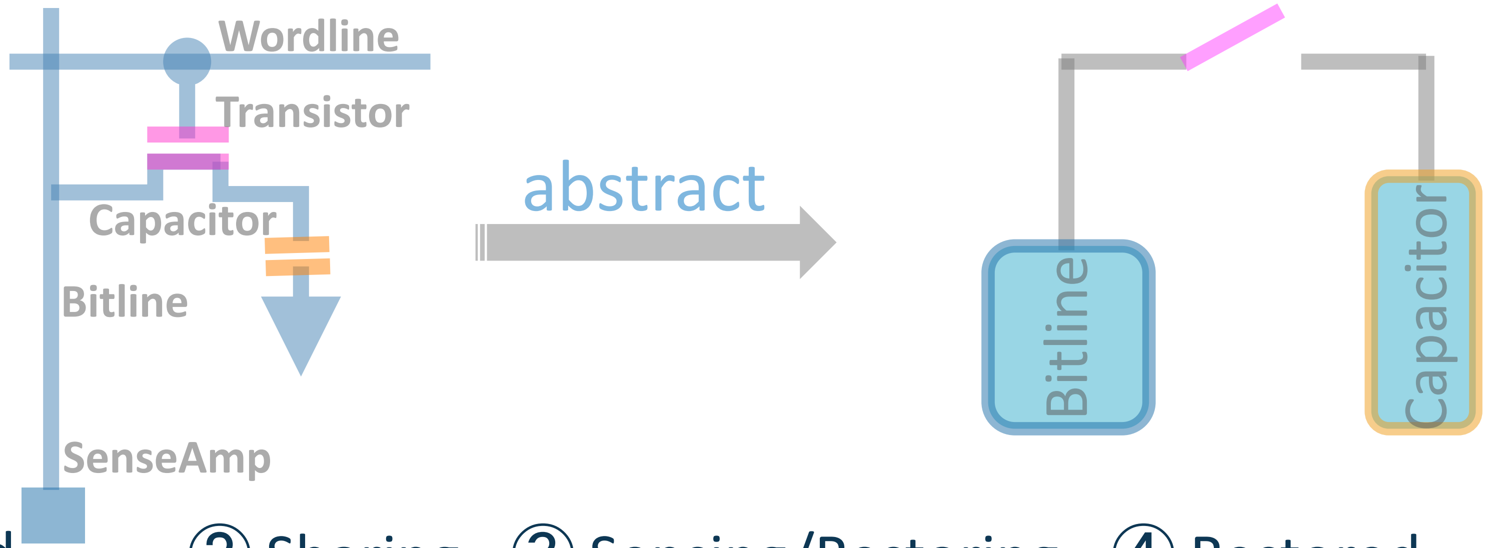
DRAM OPERATIONS

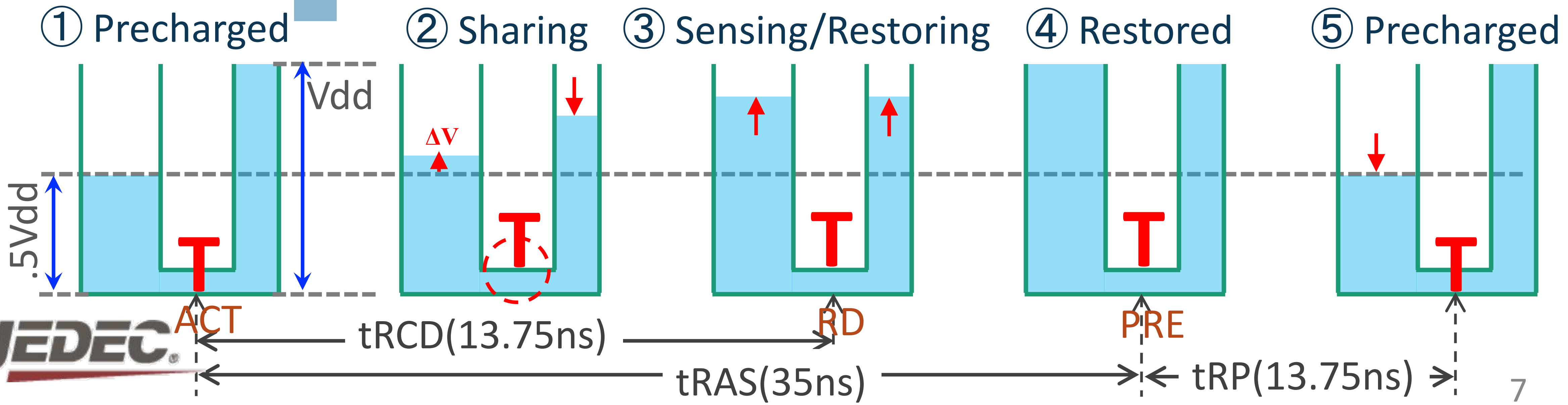




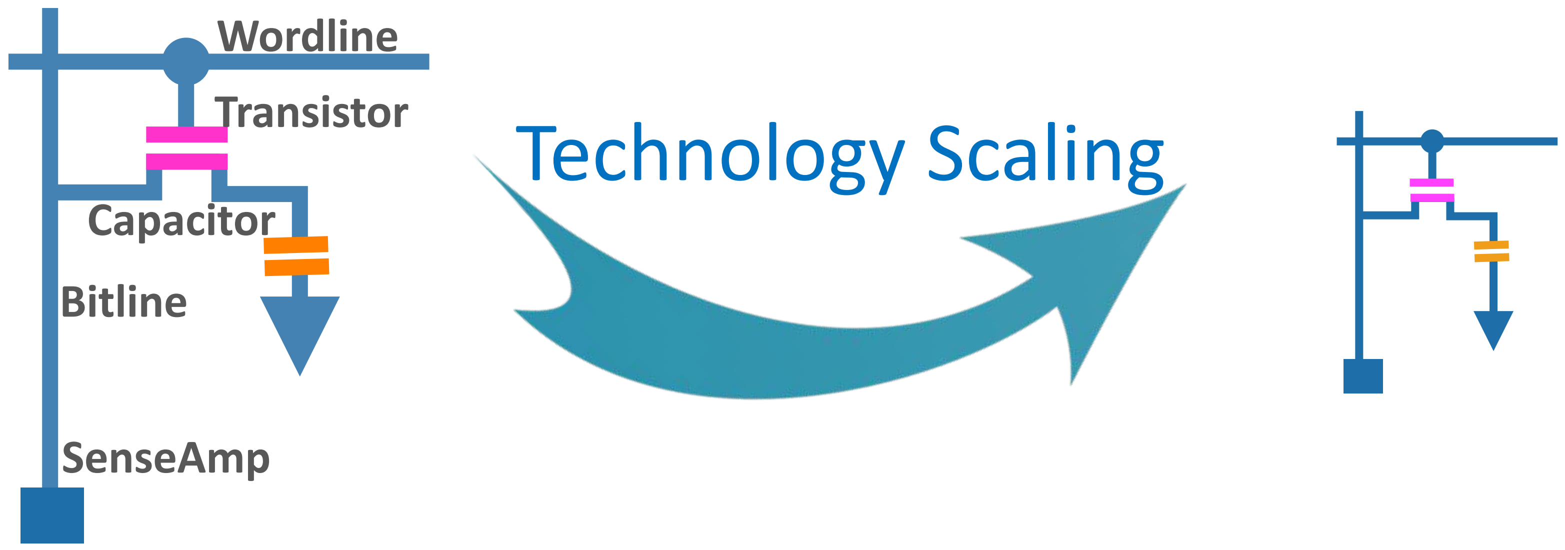


DRAM OPERATIONS

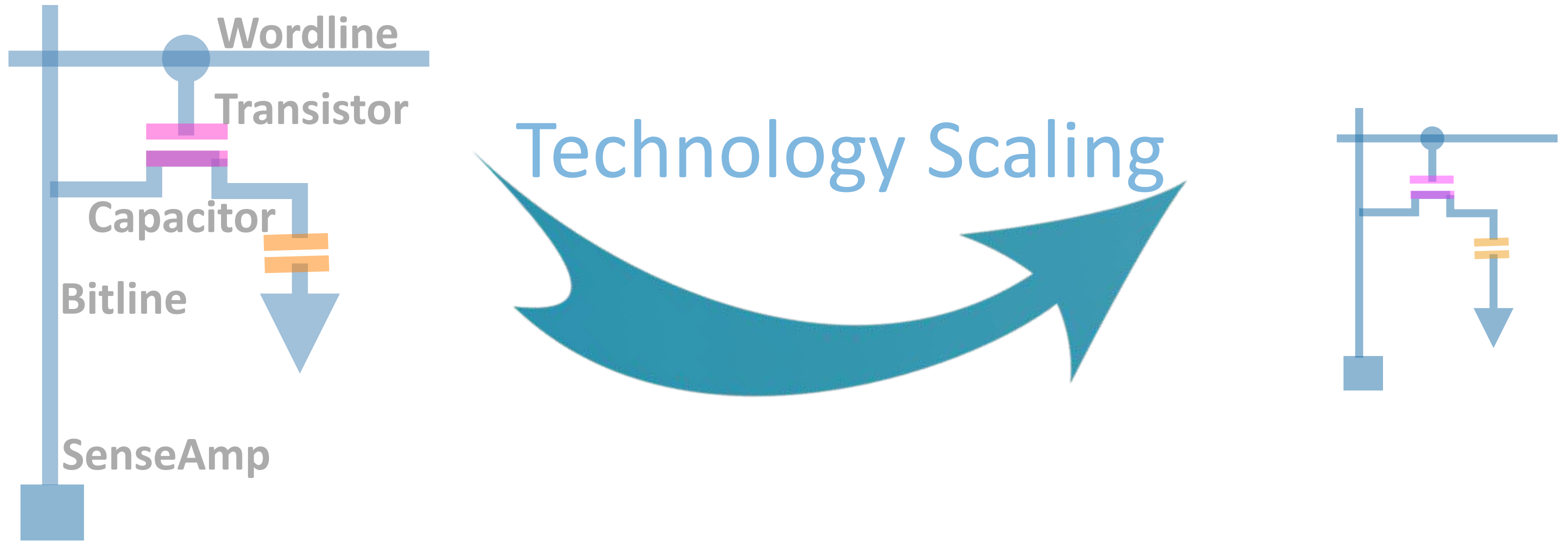




WHY DIFFICULT?



WHY DIFFICULT?

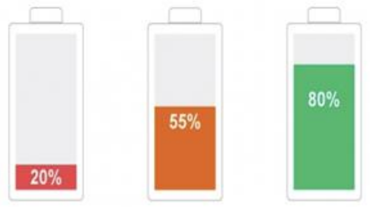


Less charge
higher leakage current

More Leaky

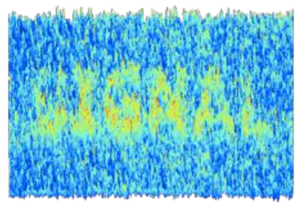
Larger resistance
Weaker signal

Longer Sensing



Larger resistance
Lower voltage

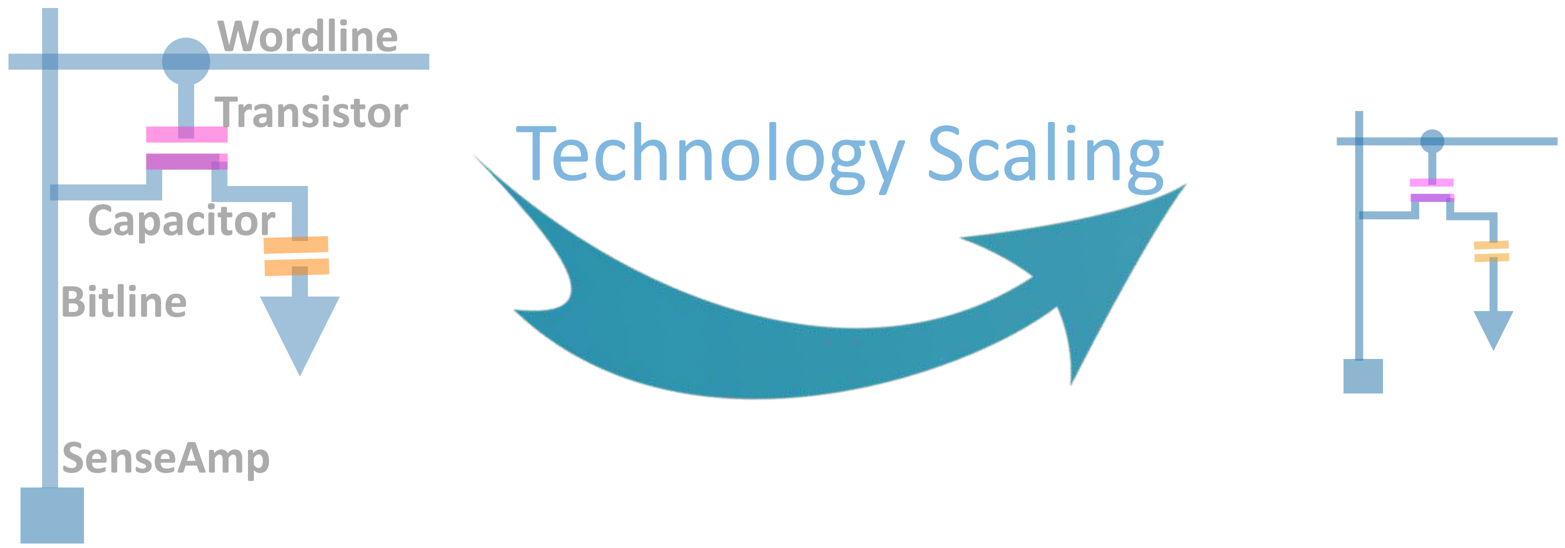
Prolonged Restore



Nearer cells
Process variations

Severer Noise

WHY DIFFICULT?

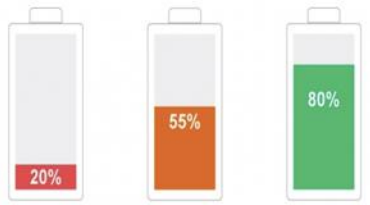


Less charge
higher leakage current

More Leaky

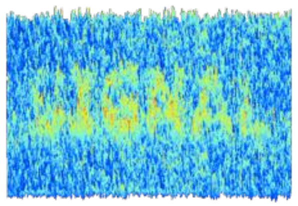
Larger resistance
Weaker signal

Longer Sensing



Larger resistance
Lower voltage

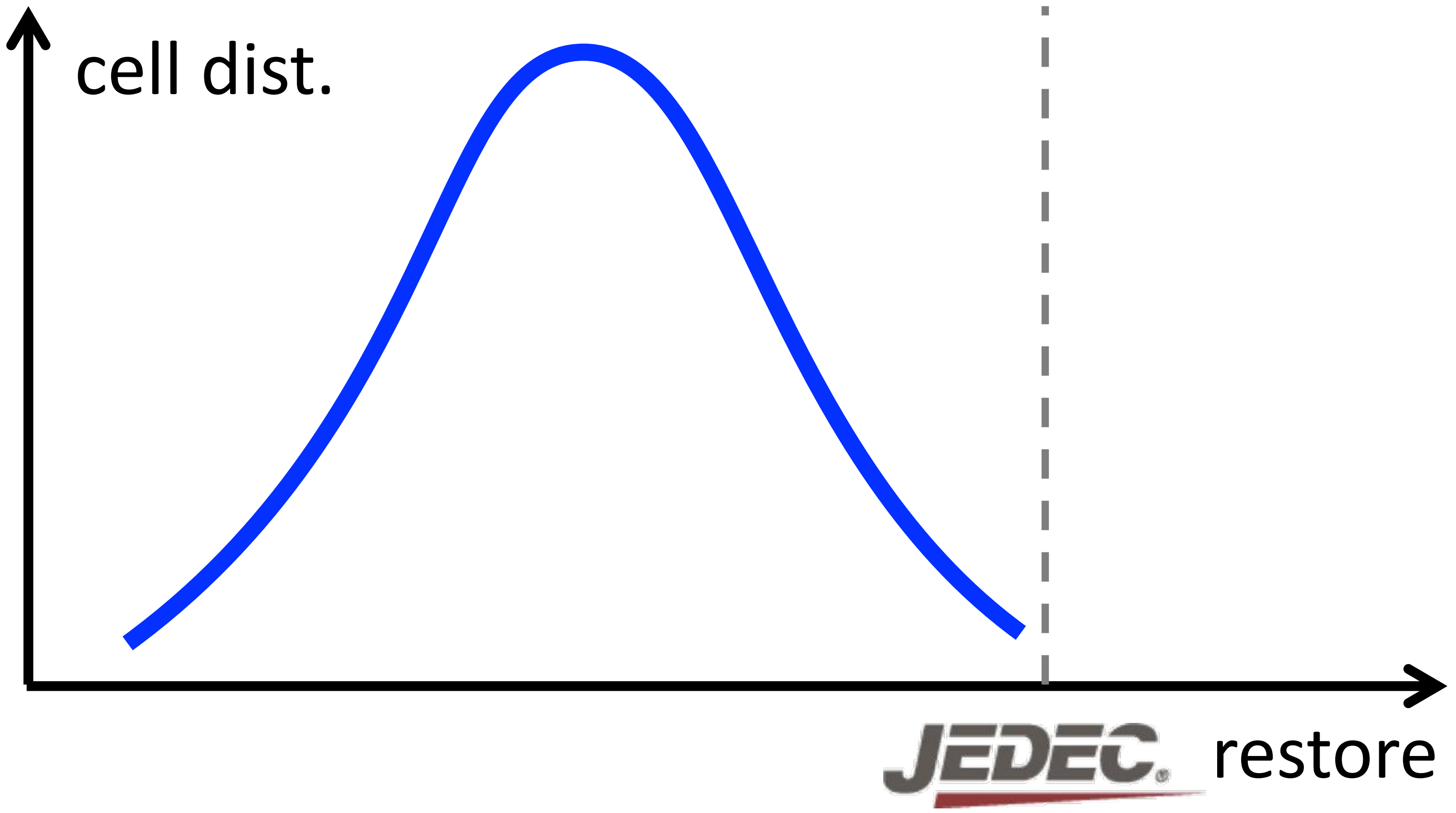
Prolonged Restore



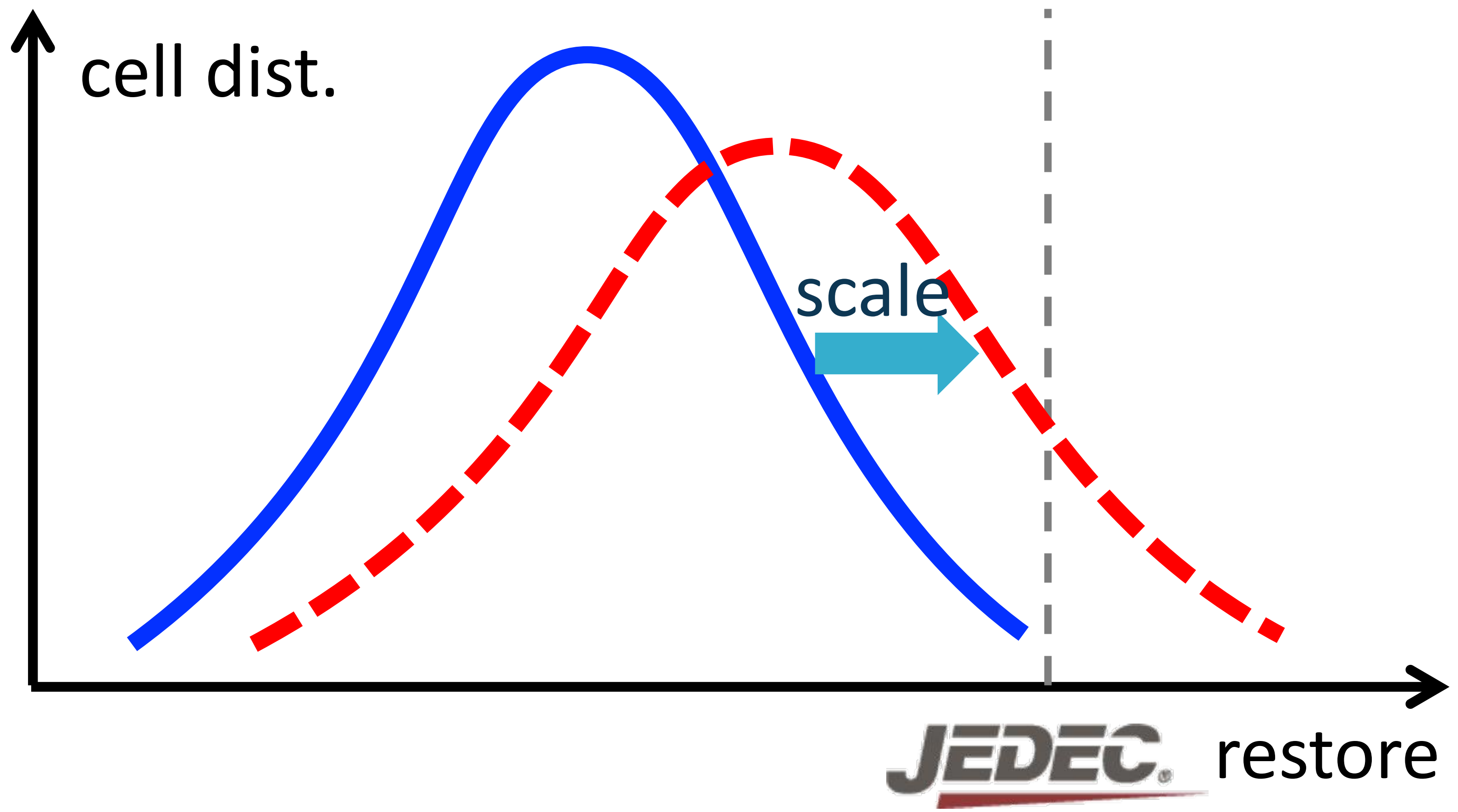
Nearer cells
Process variations

Severer Noise

RESTORE ISSUE

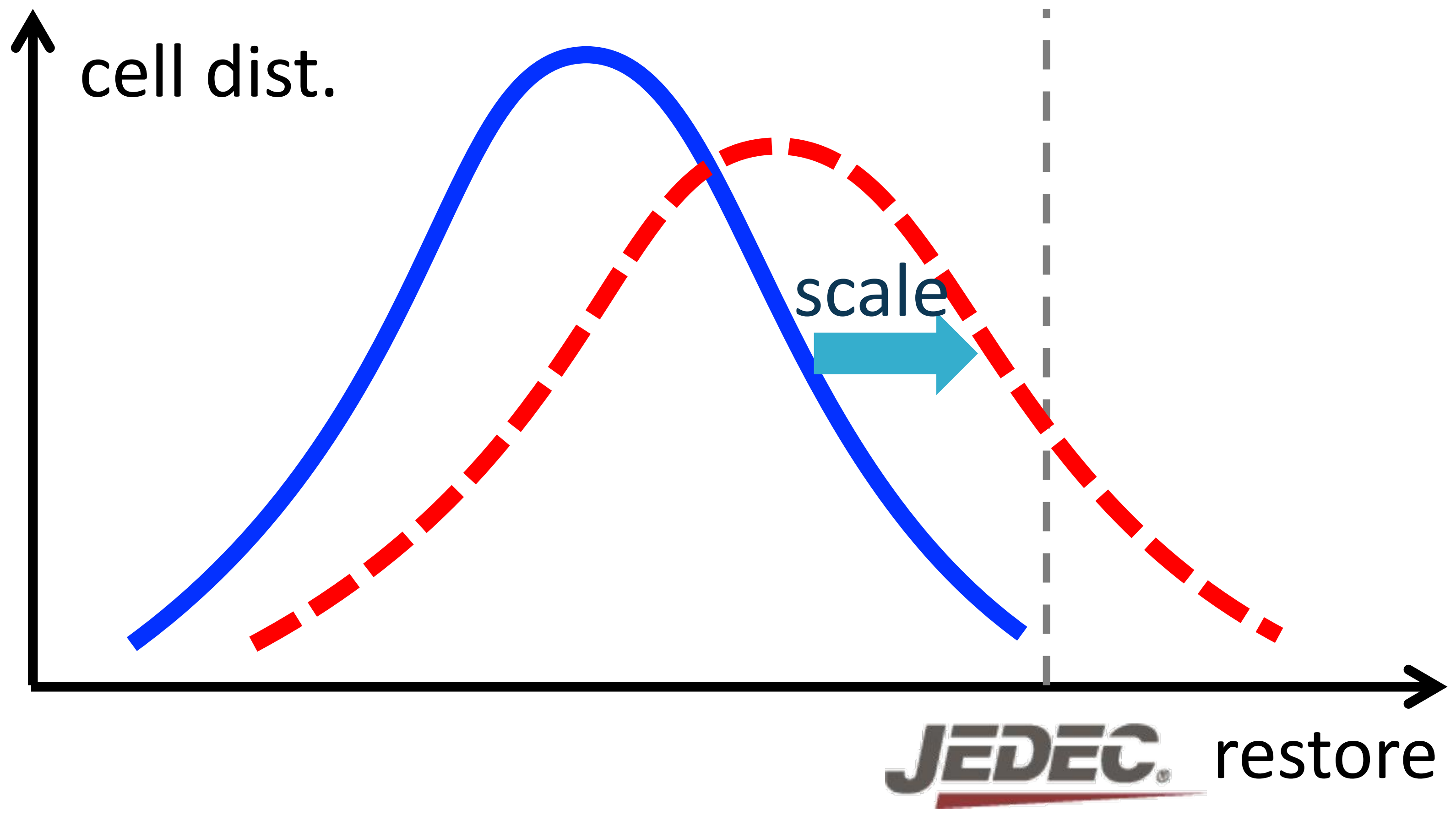


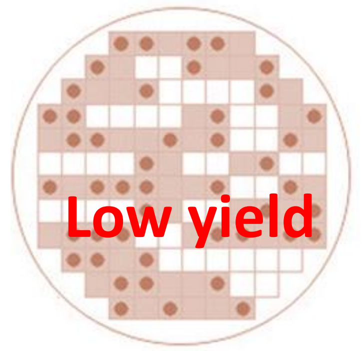
RESTORE ISSUE



More cells will be violating the JEDEC specifications

RESTORE ISSUE





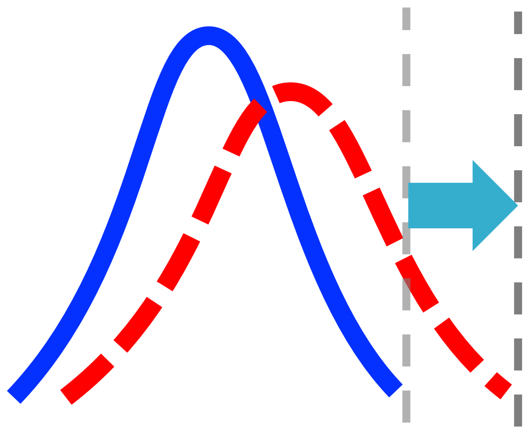
More cells will be violating the JEDEC specifications

THESIS STATEMENT

Enable DRAM further scaling without low **yield** and degraded **performance**

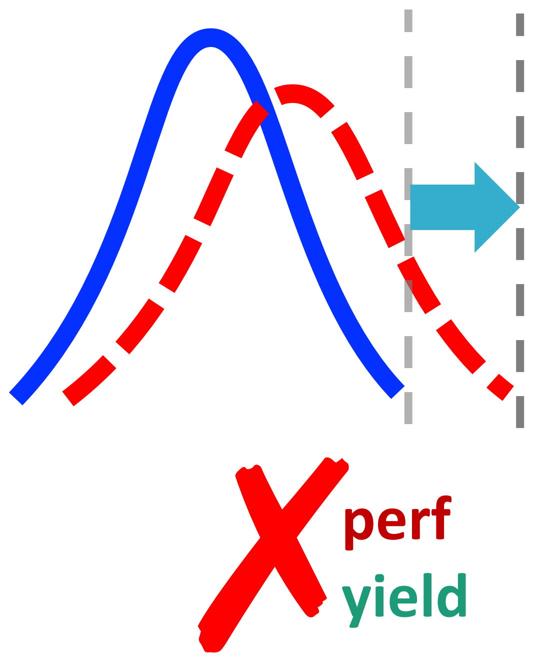
CANDIDATE SOLUTIONS

CANDIDATE SOLUTIONS



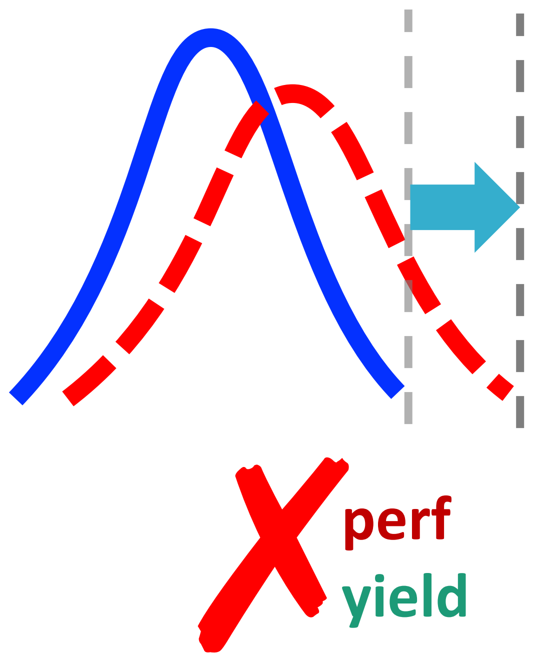
Relax standard

CANDIDATE SOLUTIONS

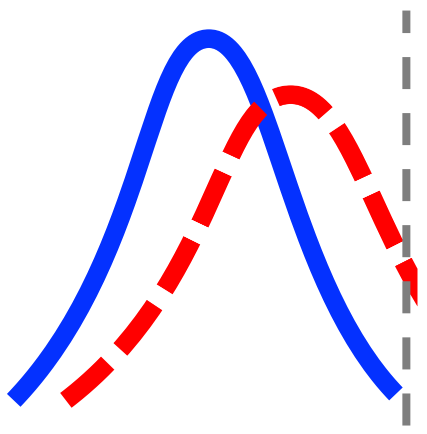


Relax standard

CANDIDATE SOLUTIONS

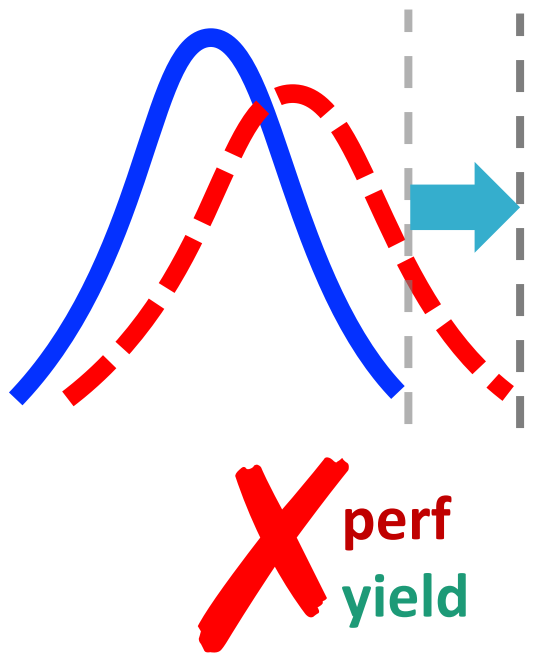


Relax standard

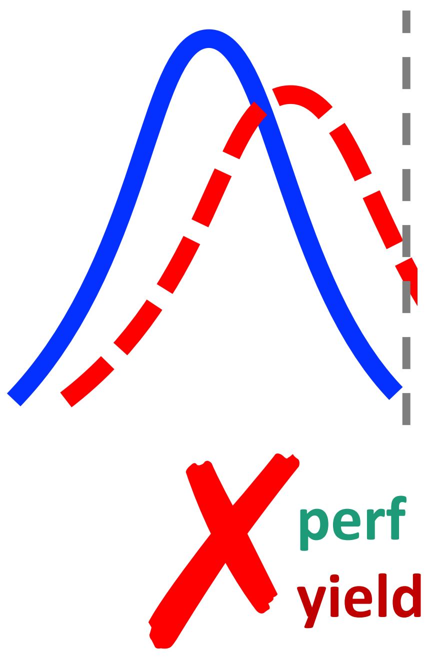


Cutoff slow ones

CANDIDATE SOLUTIONS

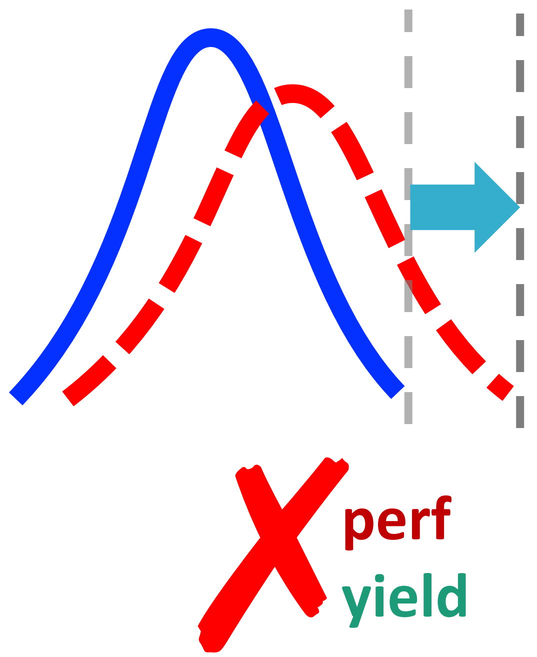


Relax standard

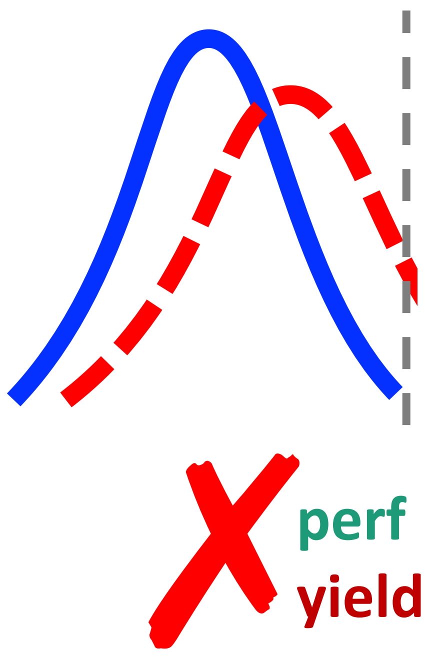


Cutoff slow ones

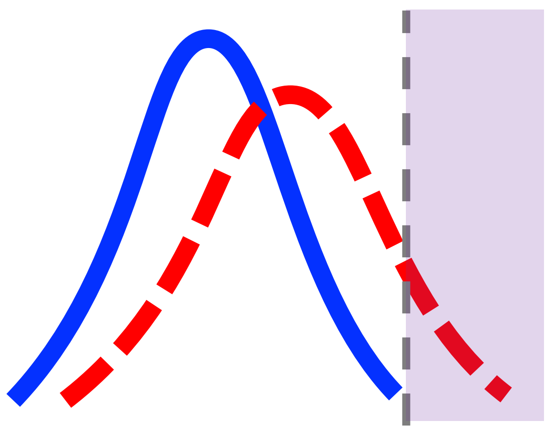
CANDIDATE SOLUTIONS



Relax standard

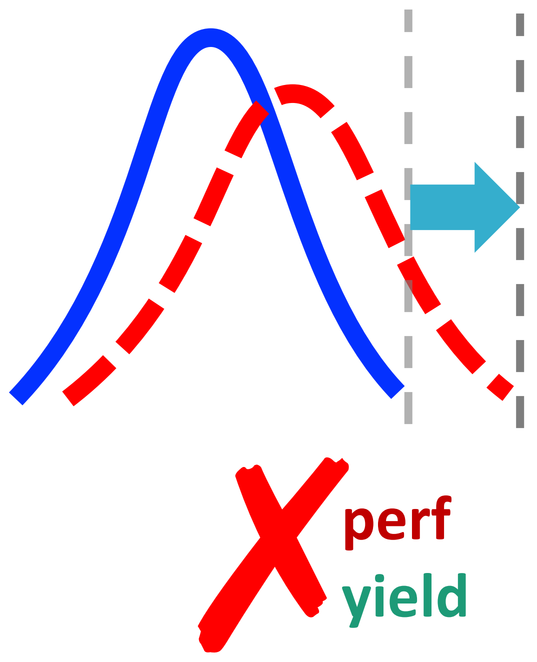


Cutoff slow ones

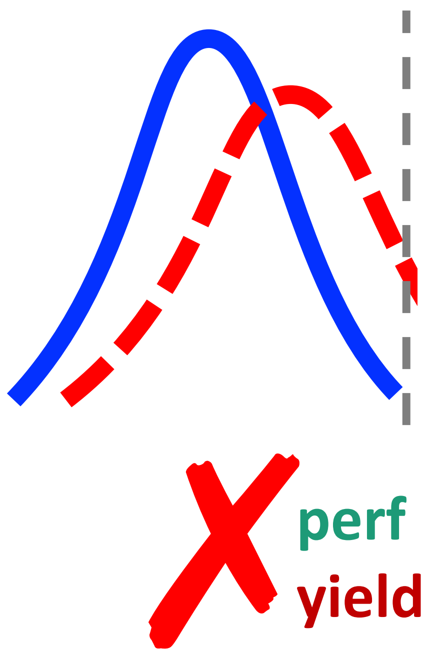


Work on slow ones

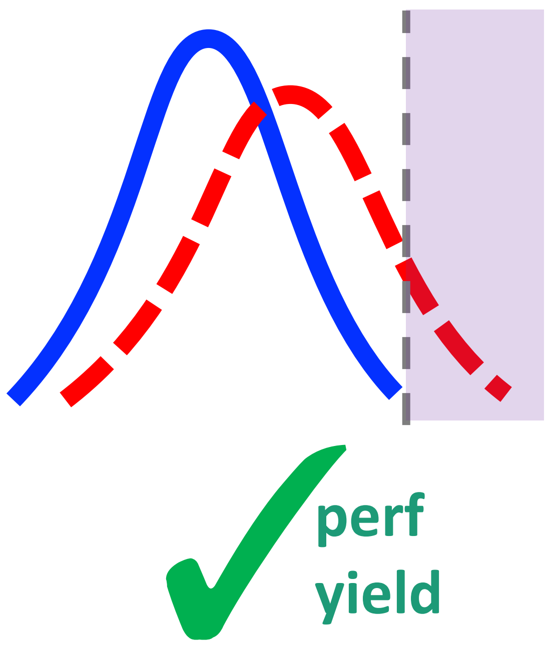
CANDIDATE SOLUTIONS



Relax standard

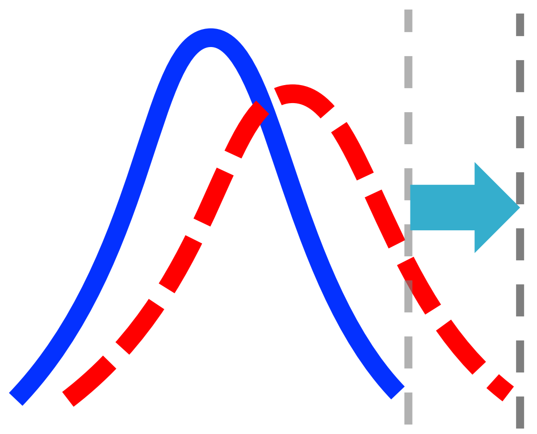


Cutoff slow ones



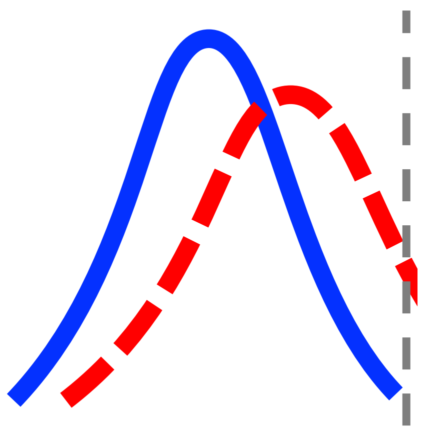
Work on slow ones

CANDIDATE SOLUTIONS



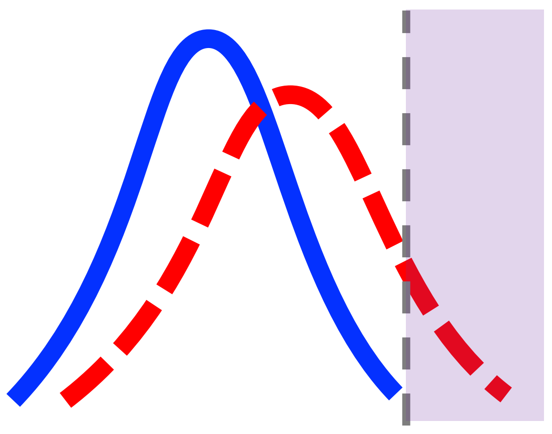
✗ perf
yield

Relax standard



✗ perf
yield

Cutoff slow ones



✓ perf
yield

Work on slow ones

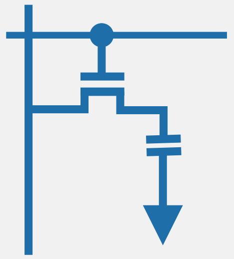
Expose slow cells to architectural levels

THESIS OVERVIEW

Address Restore Issues in Further Scaling DRAMs

THESIS OVERVIEW

Address Restore Issues in Further Scaling DRAMs

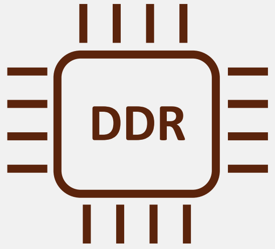


❶ **Partial restore based on refresh distance**

[RT-Next'HPCA16]

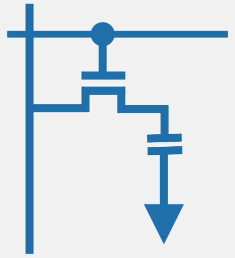
THESIS OVERVIEW

Address Restore Issues in Further Scaling DRAMs



❷ **Fast restore via reorganization and page alloc**

[CkRemap'DATE15, Alloc'TODAES17]



❶ **Partial restore based on refresh distance**

[RT-Next'HPCA16]

THESIS OVERVIEW

Address Restore Issues in Further Scaling DRAMs

❸ **Mitigate restore w/ approximate computing**
[DrMP'PACT17, Award'MemSys16]

❷ **Fast restore via reorganization and page alloc**
[CkRemap'DATE15, Alloc'TODAES17]

❶ **Partial restore based on refresh distance**
[RT-Next'HPCA16]

OUTLINE

RT-Next

Partial restore based on refresh distance

CkRemap

Fast restore via reorganization and allocation

DrMP

Mitigate restore with approximate computing

Summary and Research Directions

OUTLINE

RT-Next

Partial restore based on refresh distance

CkRemap

Fast restore via reorganization and allocation

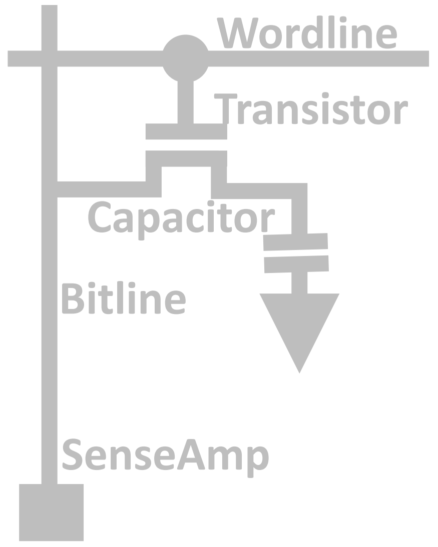
DrMP

Mitigate restore with approximate computing

Summary and Research Directions

CHARGING - RESTORE

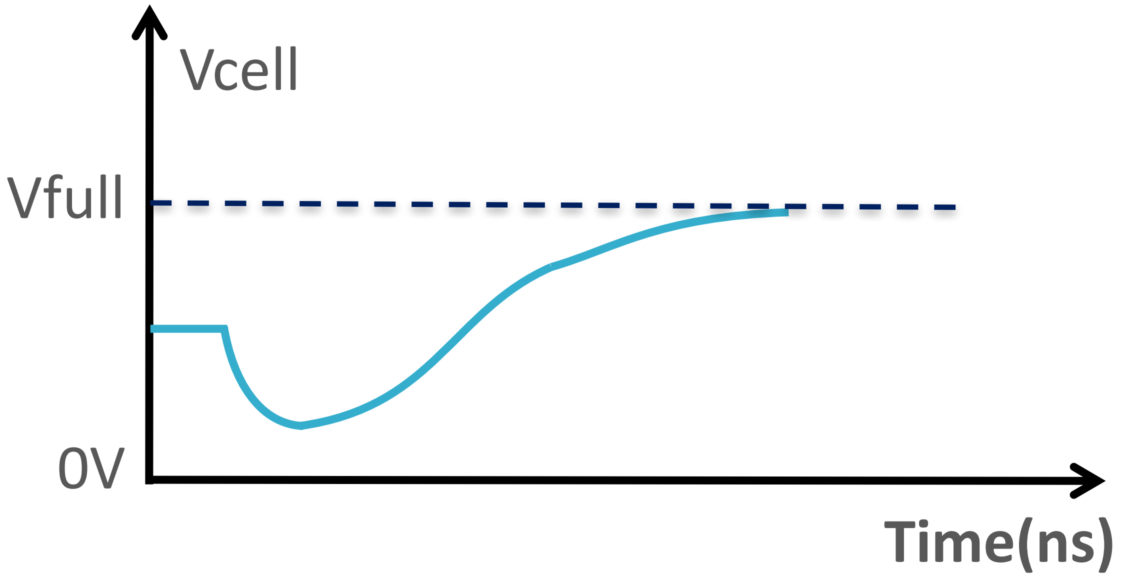
Vcell
Vfull
0V
Time(ns)

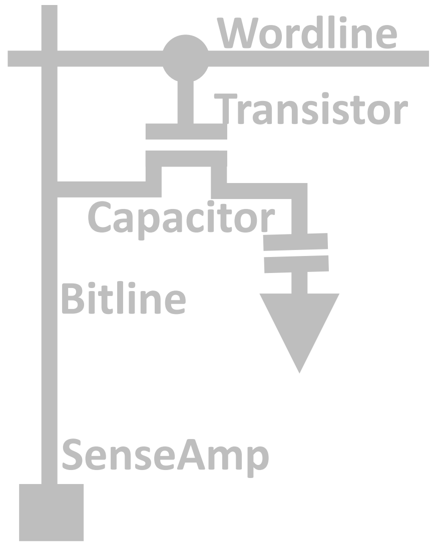


Post-access restore

- Fully charge cells
- Read (*tRAS*), Write (*tWR*)

CHARGING - RESTORE

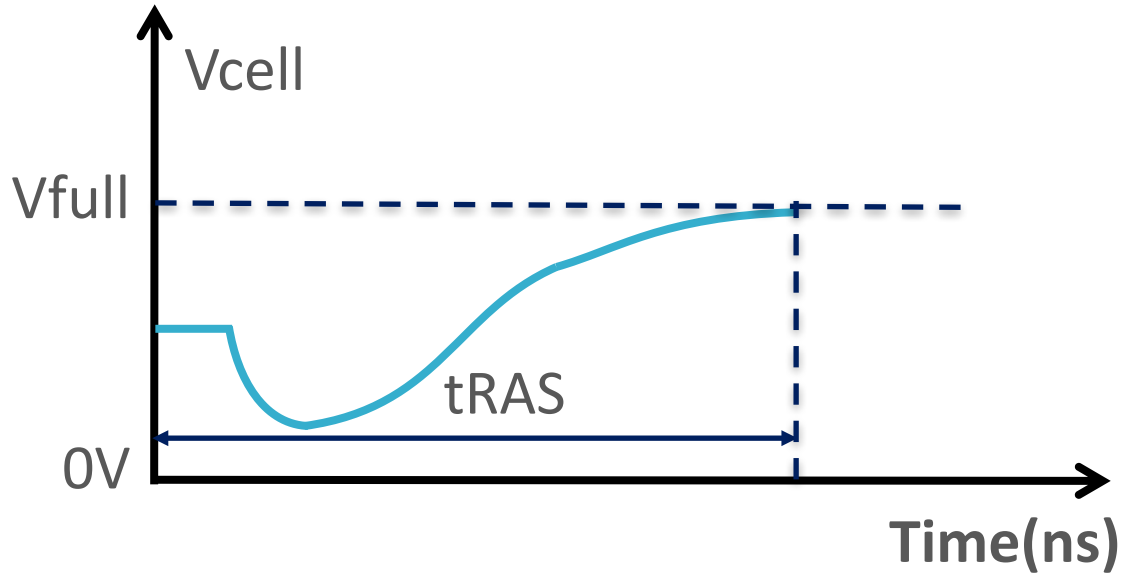


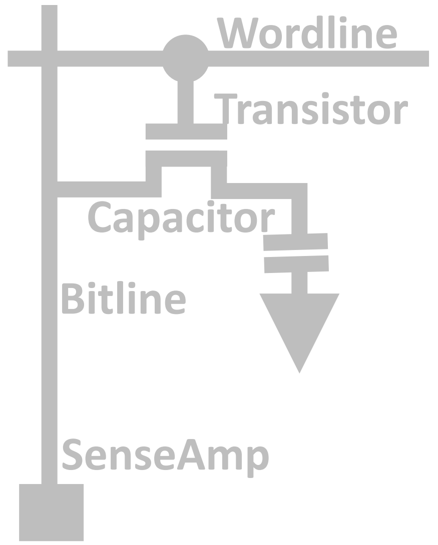


Post-access restore

- Fully charge cells
- Read (*tRAS*), Write (*tWR*)

CHARGING - RESTORE

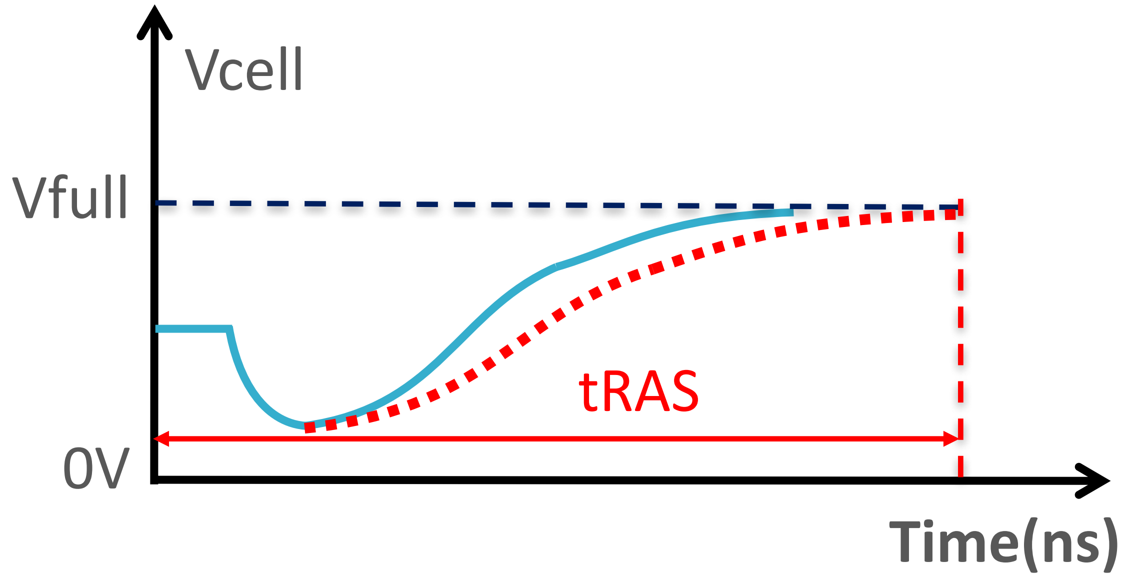


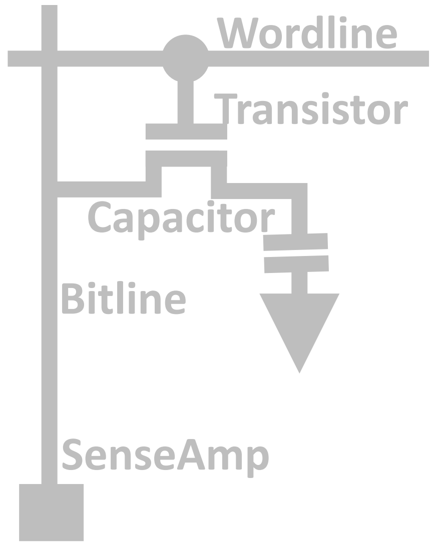


Post-access restore

- Fully charge cells
- Read (*tRAS*), Write (*tWR*)

CHARGING - RESTORE

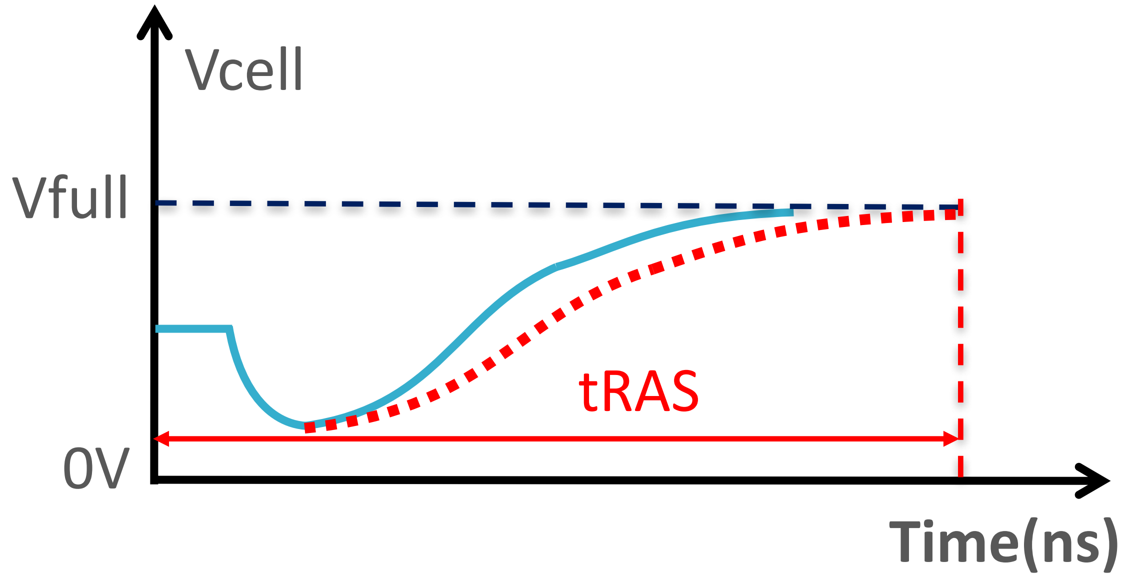


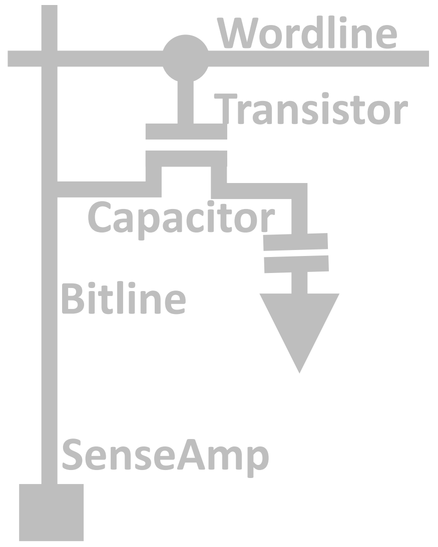


Post-access restore

- Fully charge cells
- Read (*tRAS*), Write (*tWR*)

CHARGING - RESTORE



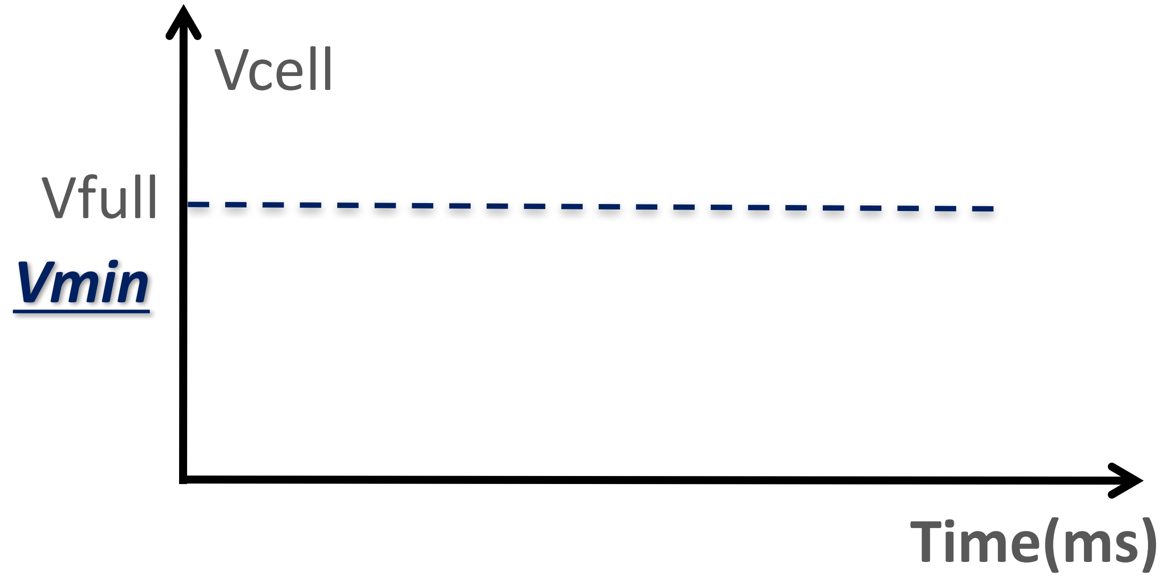


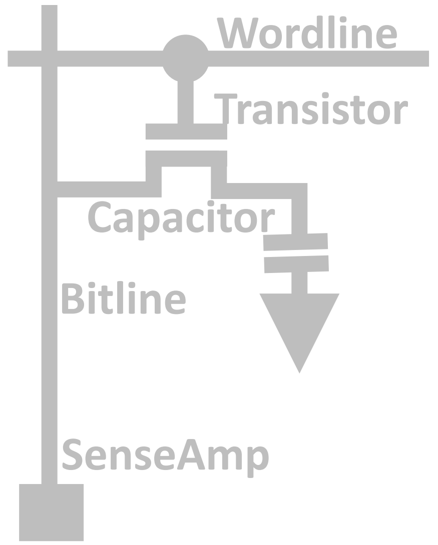
Post-access restore

- Fully charge cells
- Read (*tRAS*), Write (*tWR*)

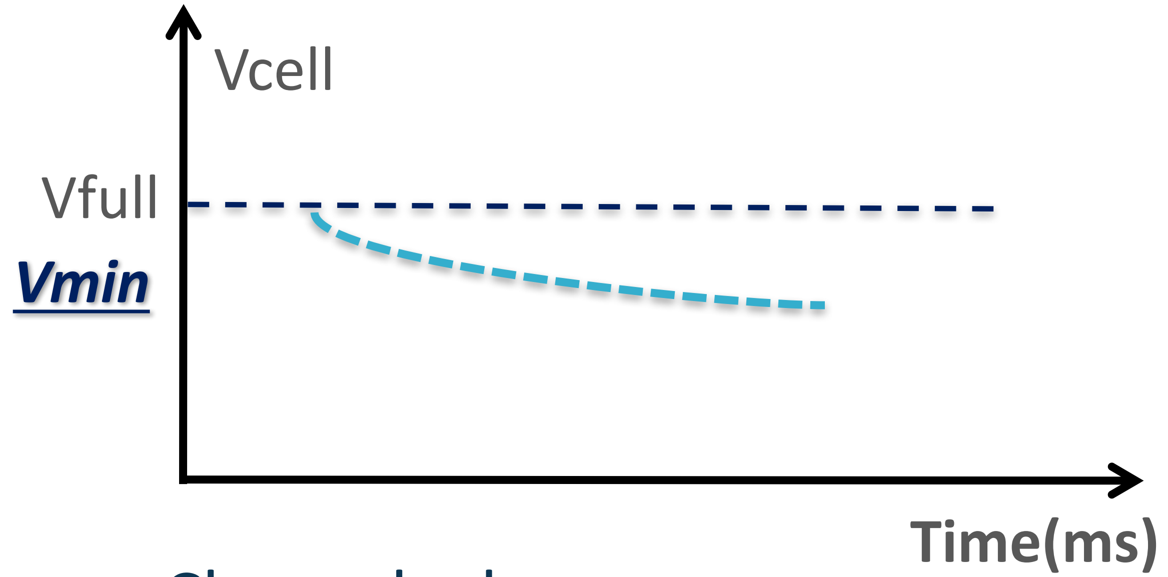
Prolonged restore leads to slow read/write

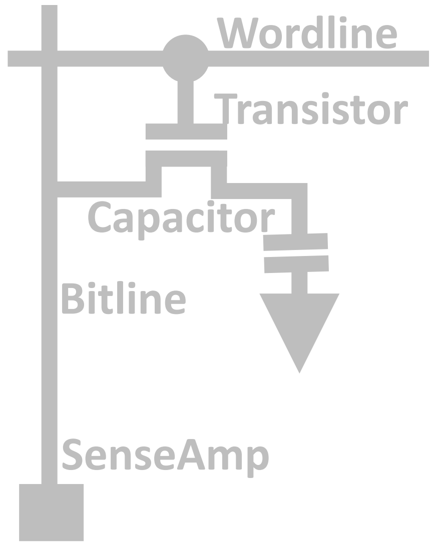
CHARGING - REFRESH





CHARGING - REFRESH

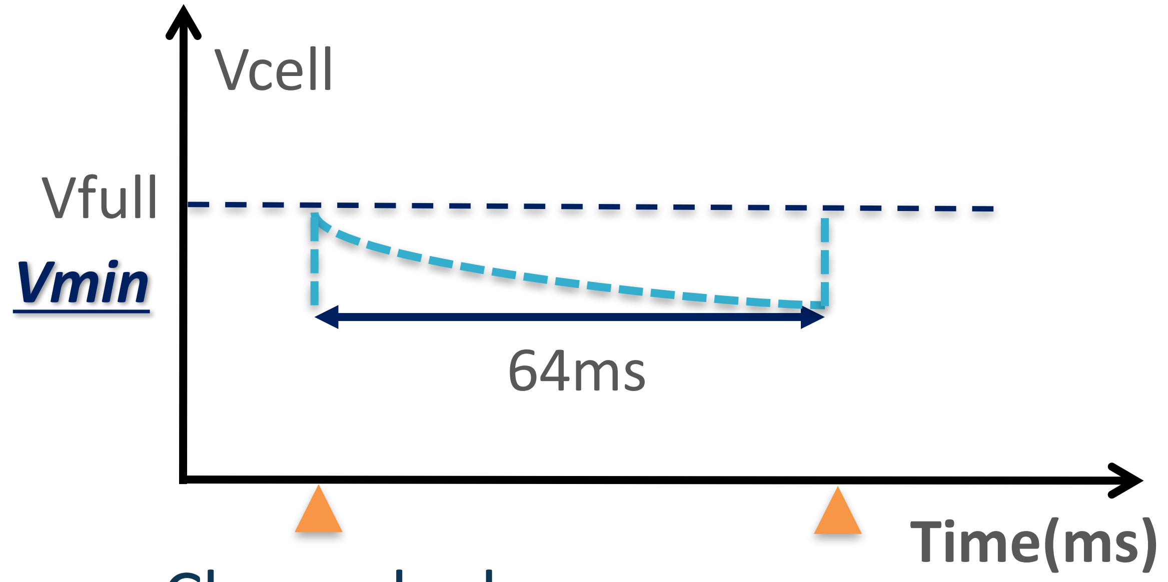


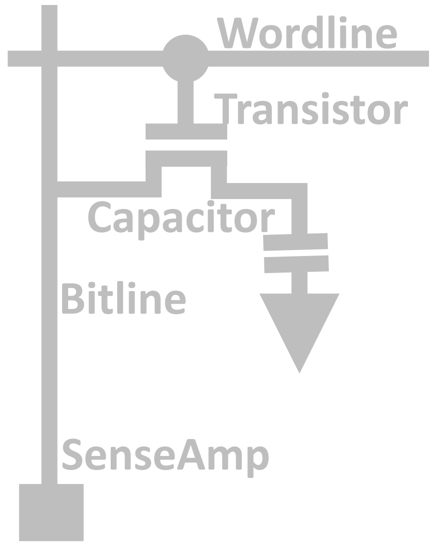


Charge leakage

- Cell charge decays over time

CHARGING - REFRESH





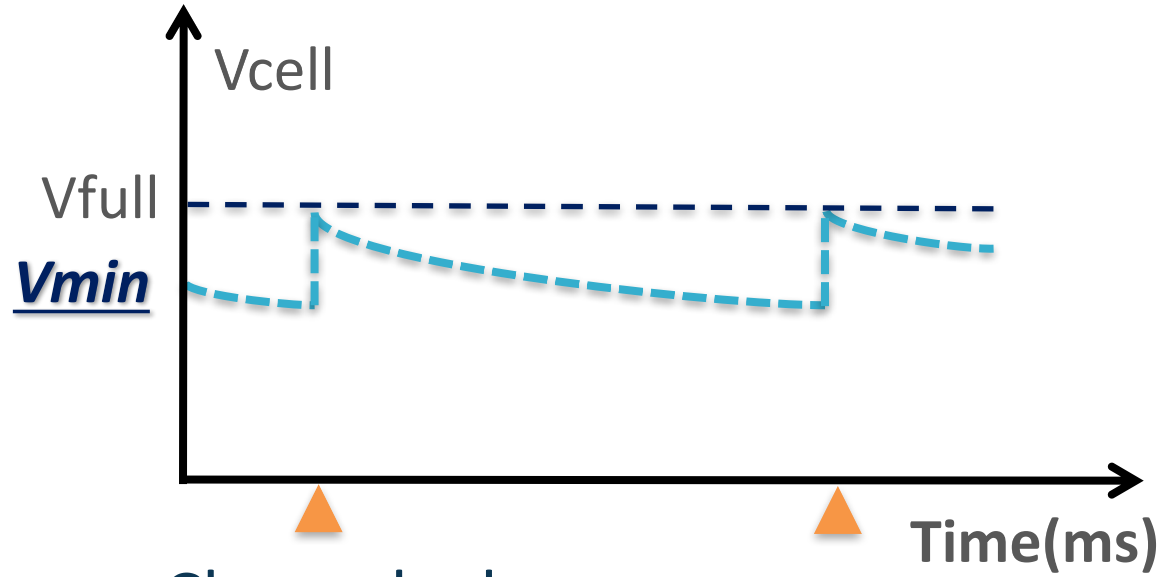
Charge leakage

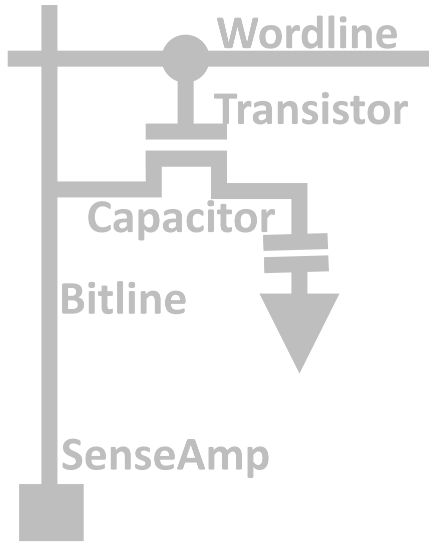
- Cell charge decays over time

Refresh operation

- Periodically fully charge cells to avoid data loss

CHARGING - REFRESH





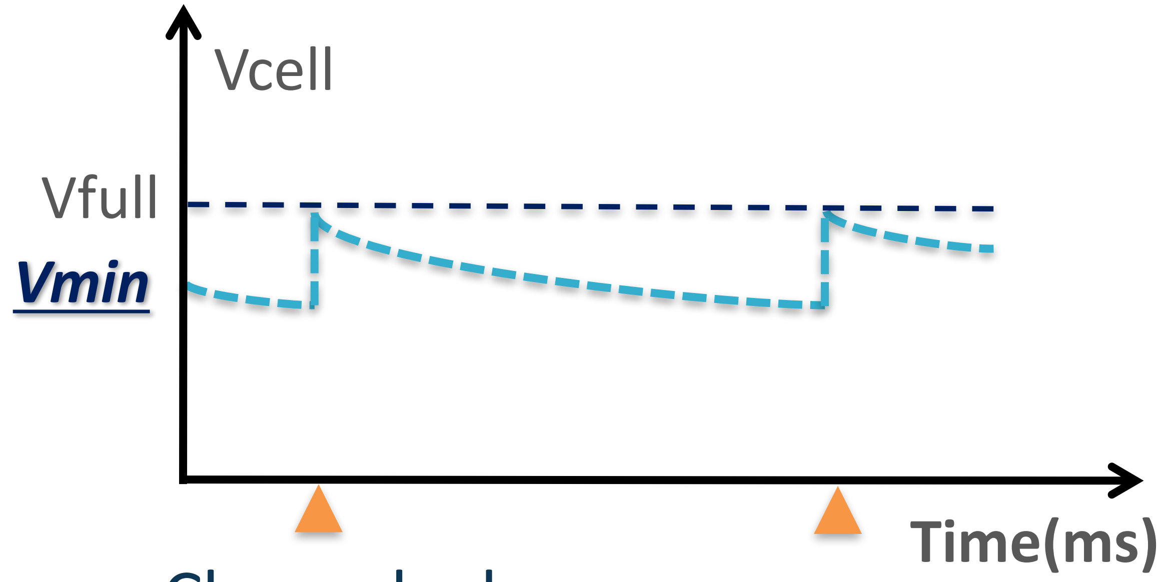
Charge leakage

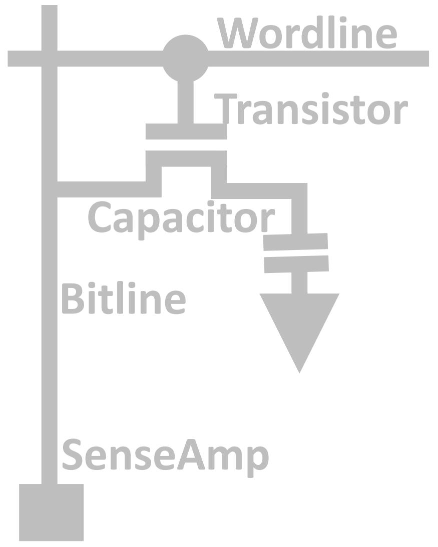
- Cell charge decays over time

Refresh operation

- Periodically fully charge cells to avoid data loss

CHARGING - REFRESH





Charge leakage

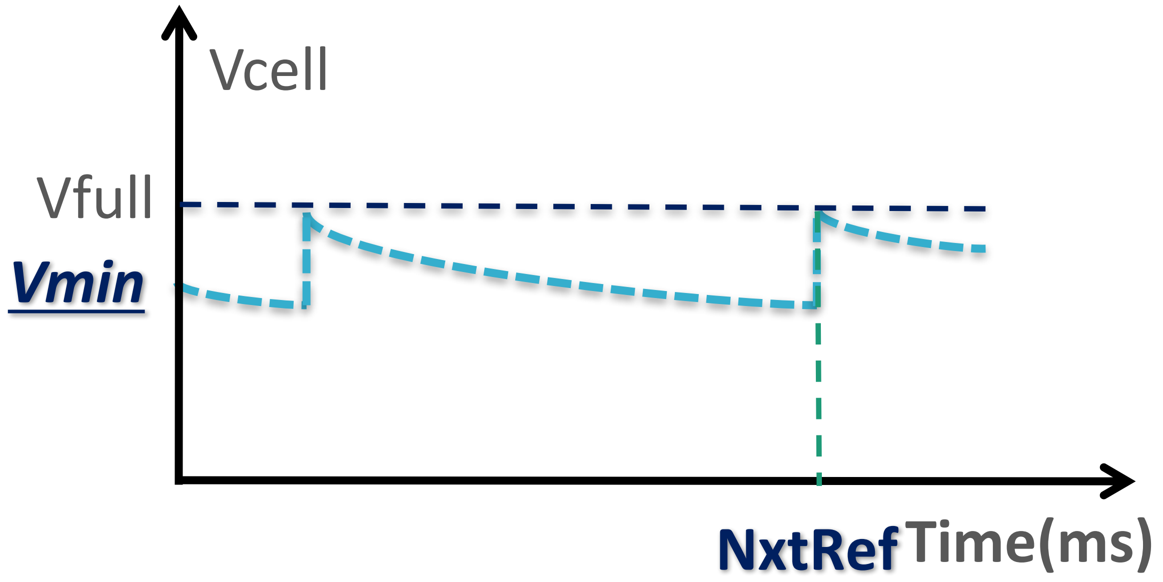
- Cell charge decays over time

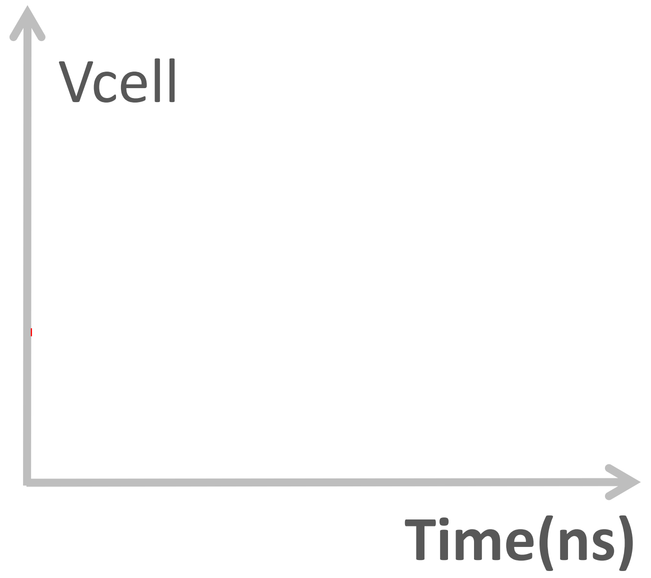
Refresh operation

- Periodically fully charge cells to avoid data loss

Do we still need to fully restore the cell after r/w?

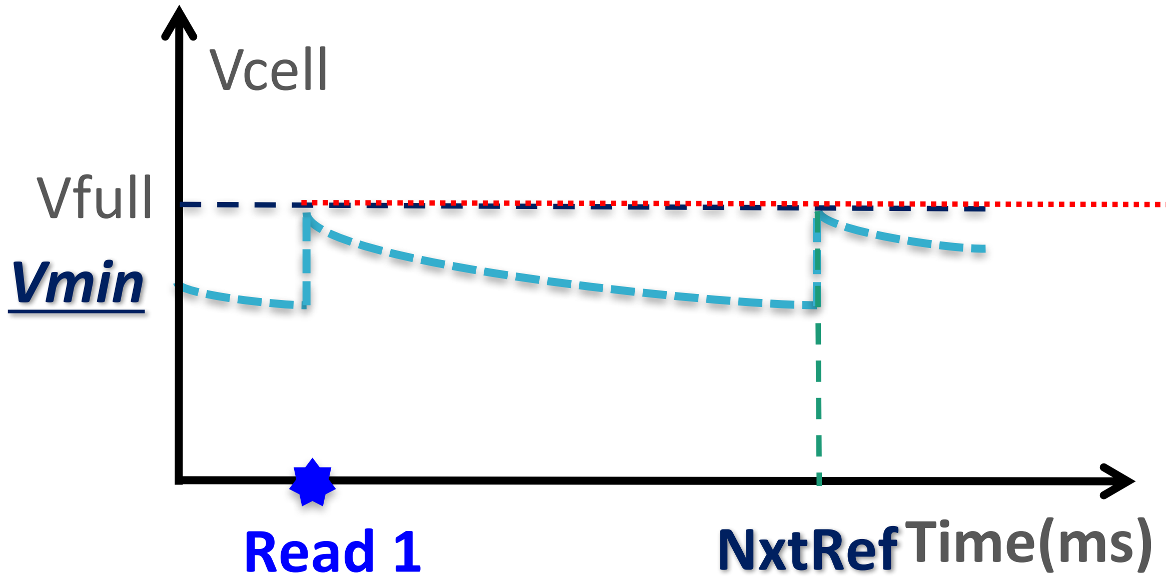
PARTIAL-RESTORE OPPORTUNITIES

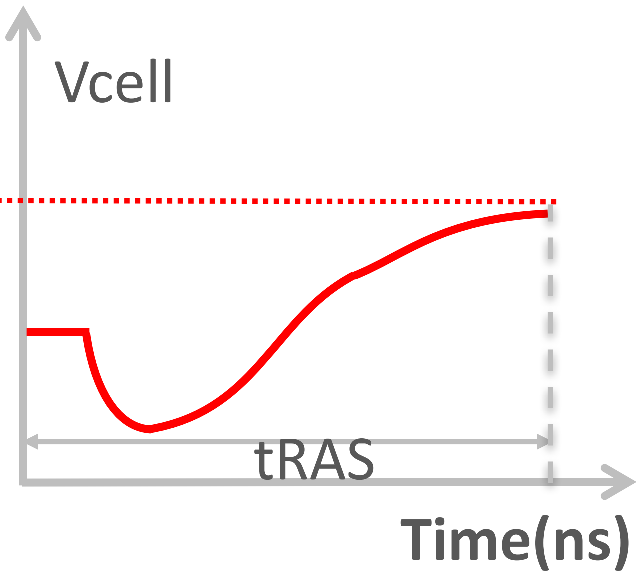




Answer: YES and NO

PARTIAL-RESTORE OPPORTUNITIES

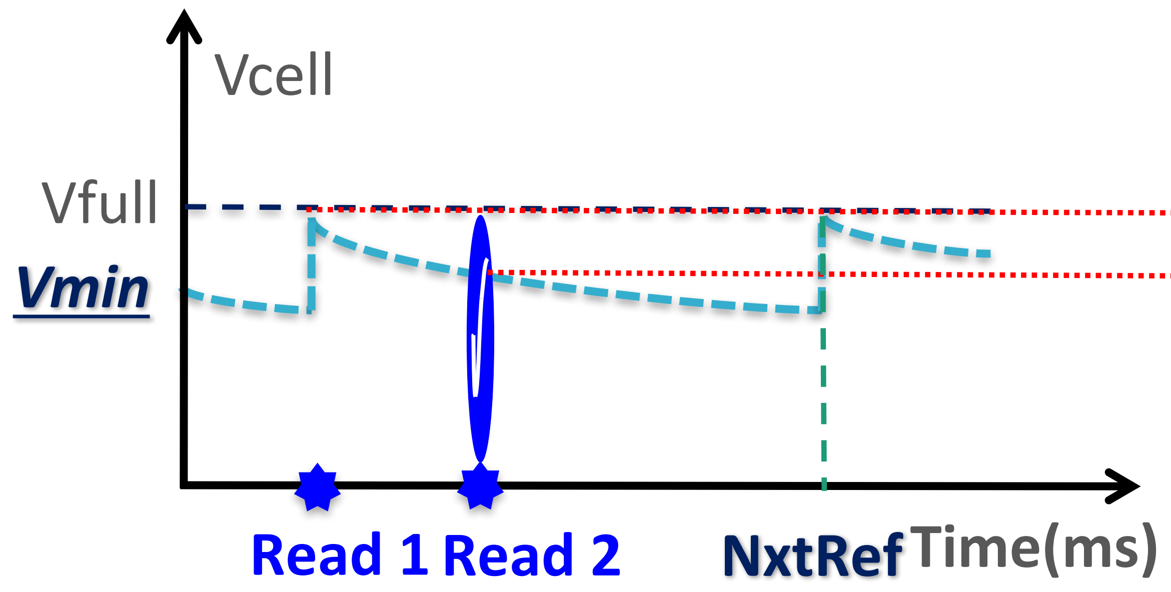


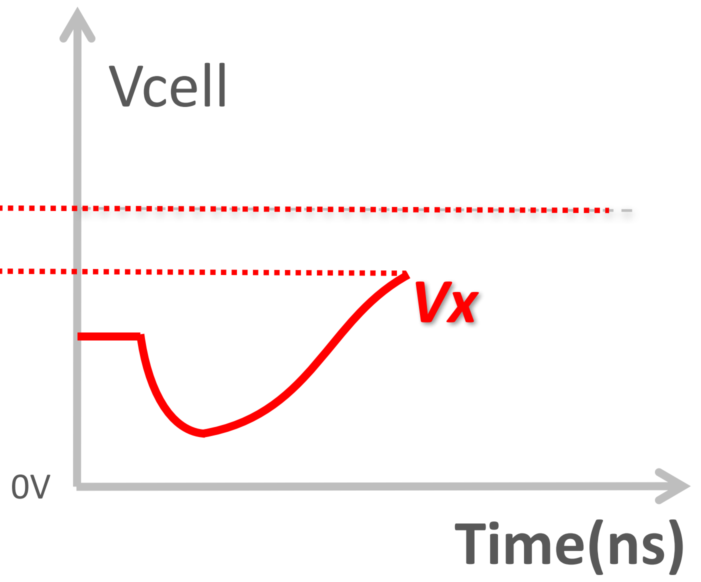


Answer: YES and NO

Read 1: Yes !

PARTIAL-RESTORE OPPORTUNITIES



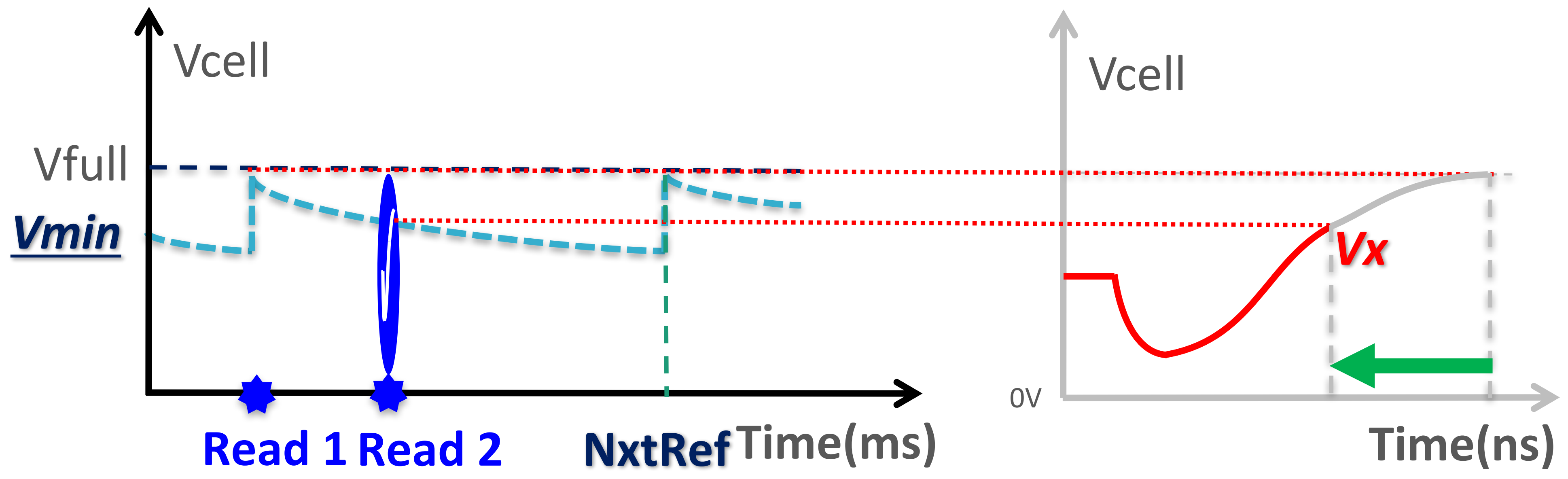


Do we always fully restore?

Read 1: Yes !

Read 2: No! It is safe to partially charge to ***Vx***

PARTIAL-RESTORE OPPORTUNITIES

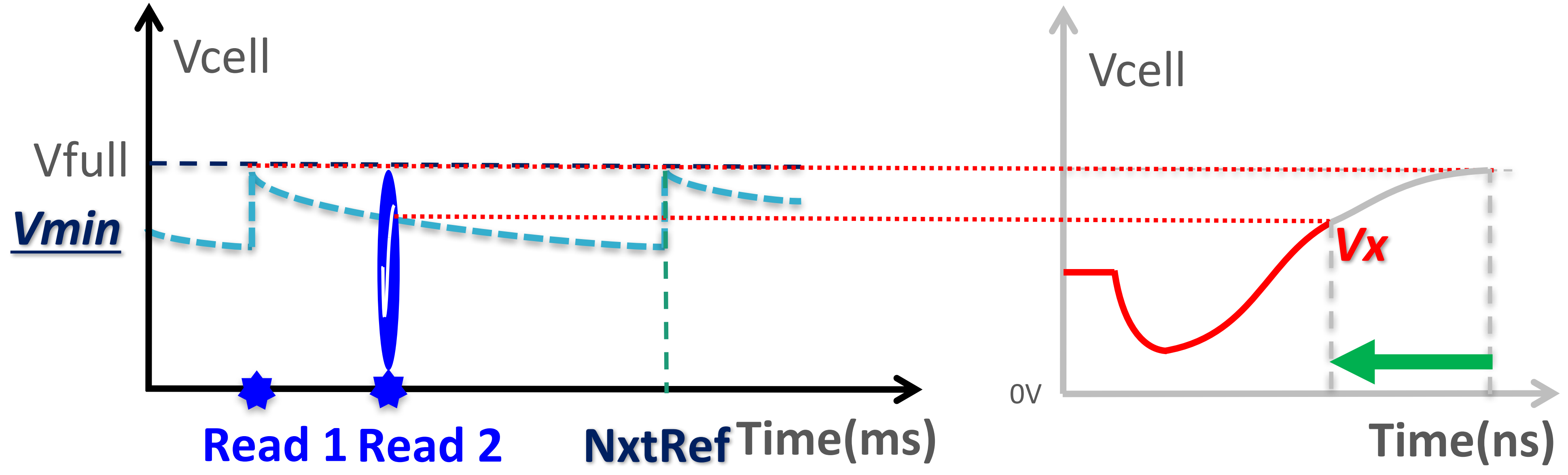


Do we always fully restore?

Read 1: Yes !

Read 2: No! It is safe to partially charge to ***Vx***

PARTIAL-RESTORE OPPORTUNITIES



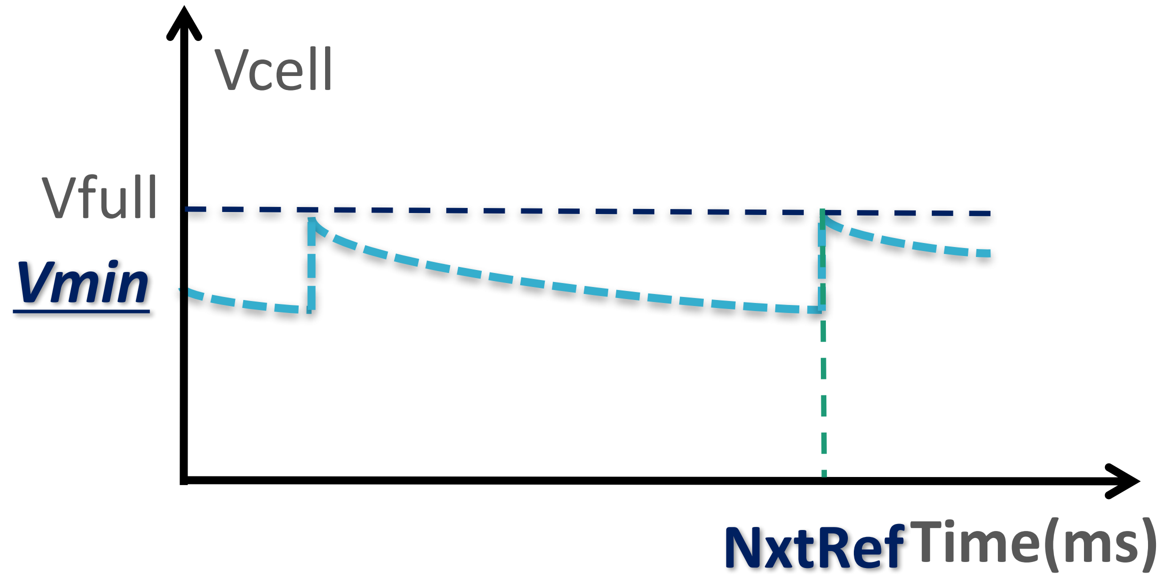
Do we always fully restore?

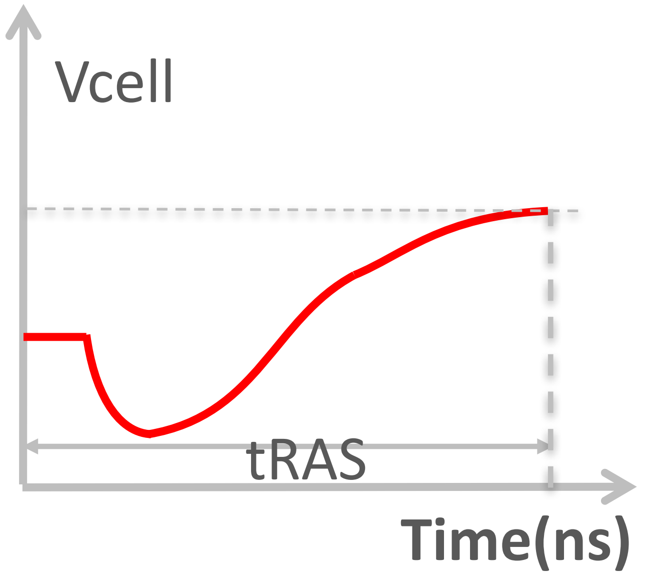
Read 1: Yes !

Read 2: No! It is safe to partially charge to ***Vx***

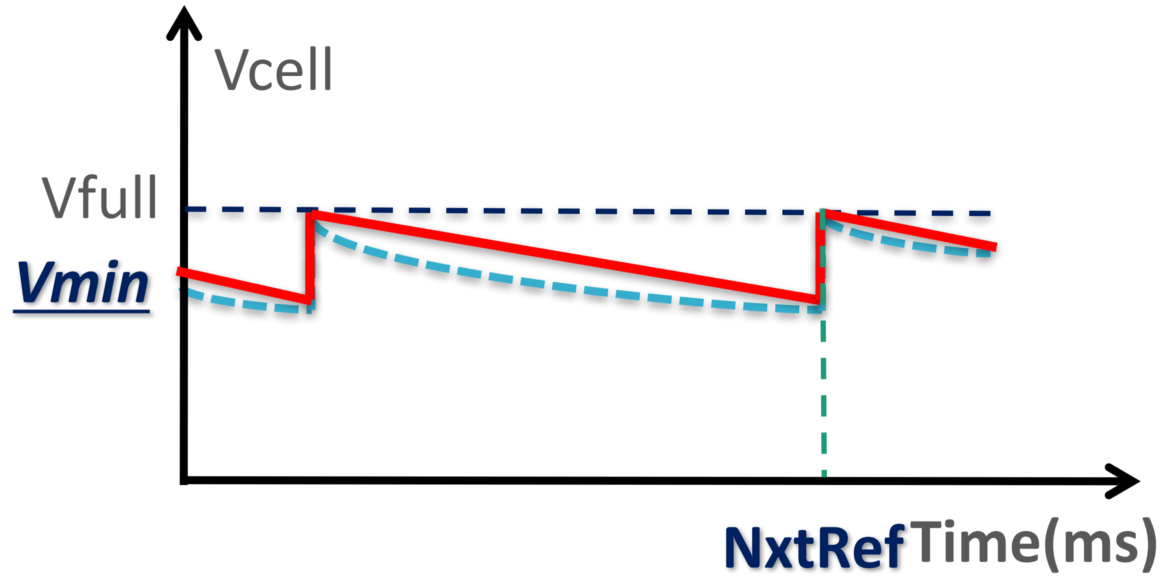
But, how should we determine Vx?

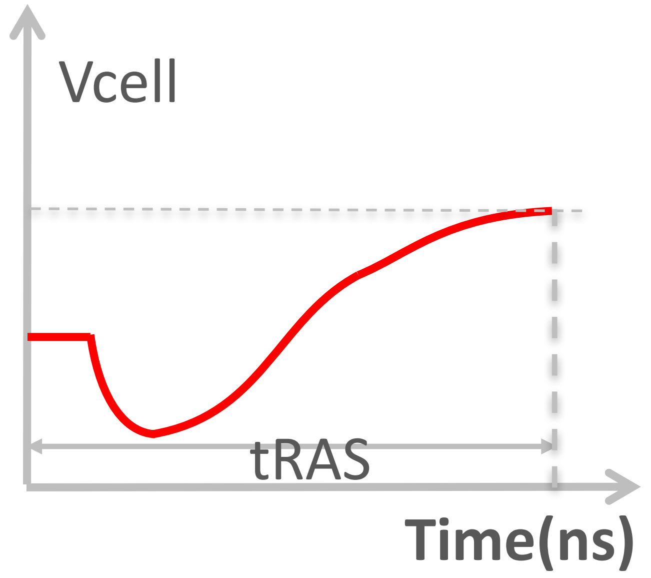
DETERMINE VX





DETERMINE VX

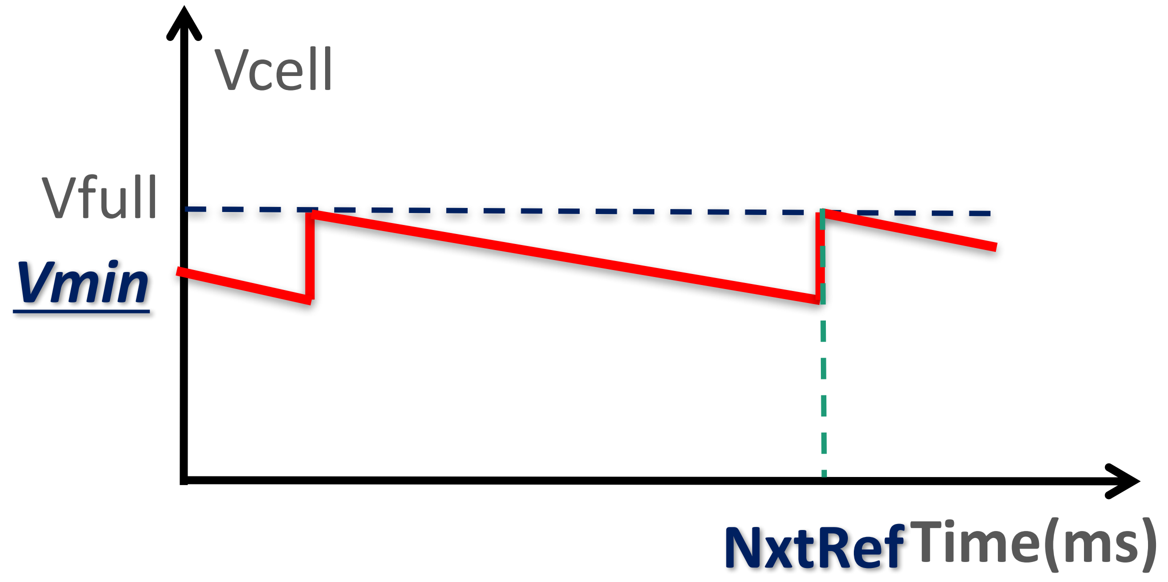


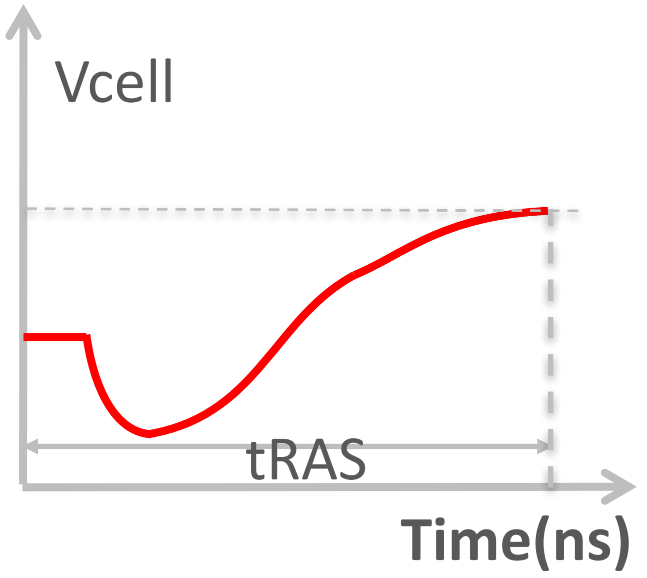


Linear restore curve

- Data is safe as long as the voltage is above decay curve

DETERMINE VX

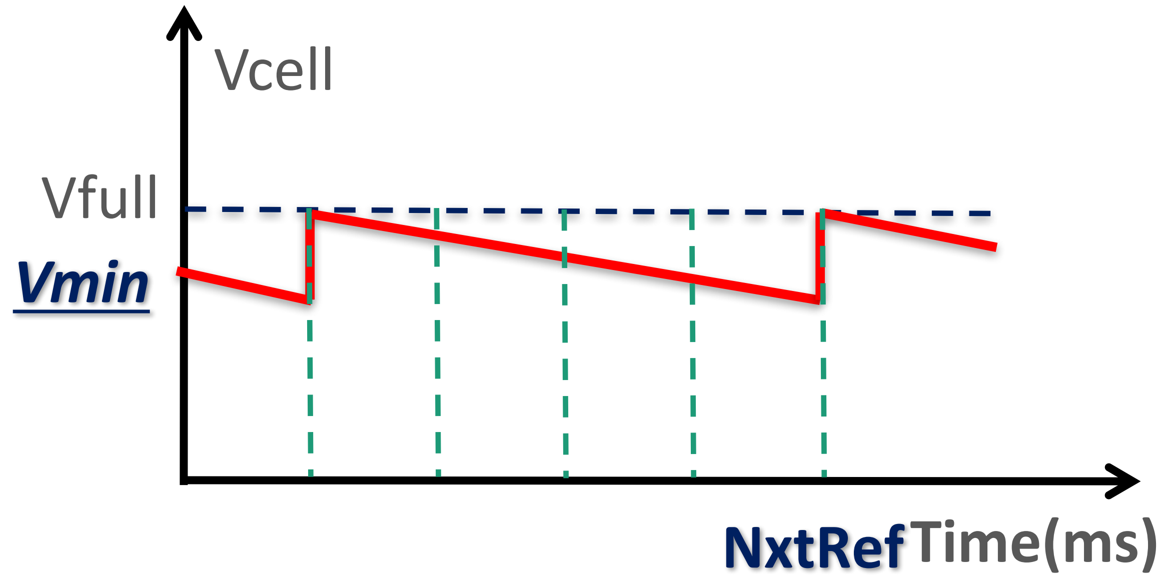


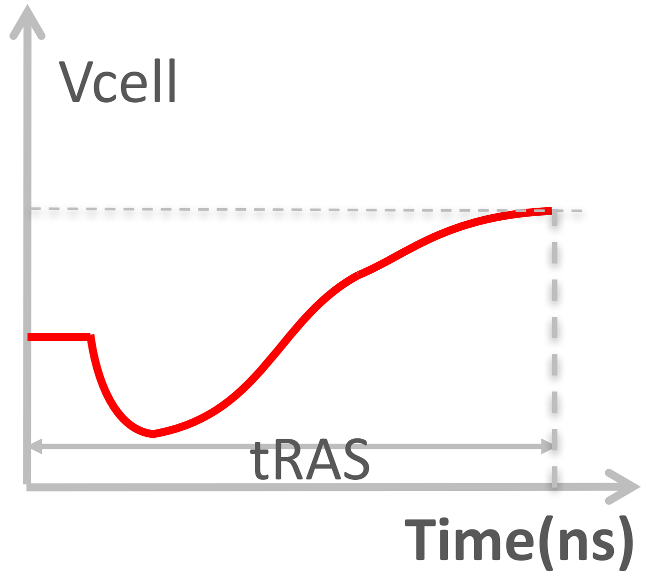


Linear restore curve

- Data is safe as long as the voltage is above decay curve

DETERMINE VX





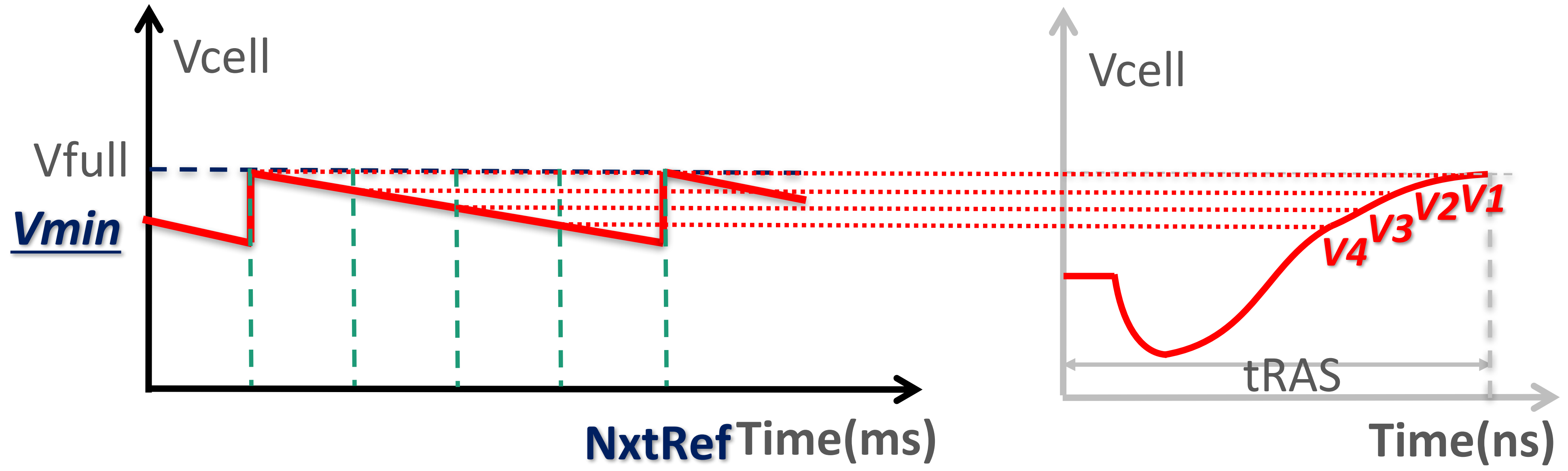
Linear restore curve

- Data is safe as long as the voltage is above decay curve

Use four sub-windows

- Save a set of timings for each

DETERMINE VX



Linear restore curve

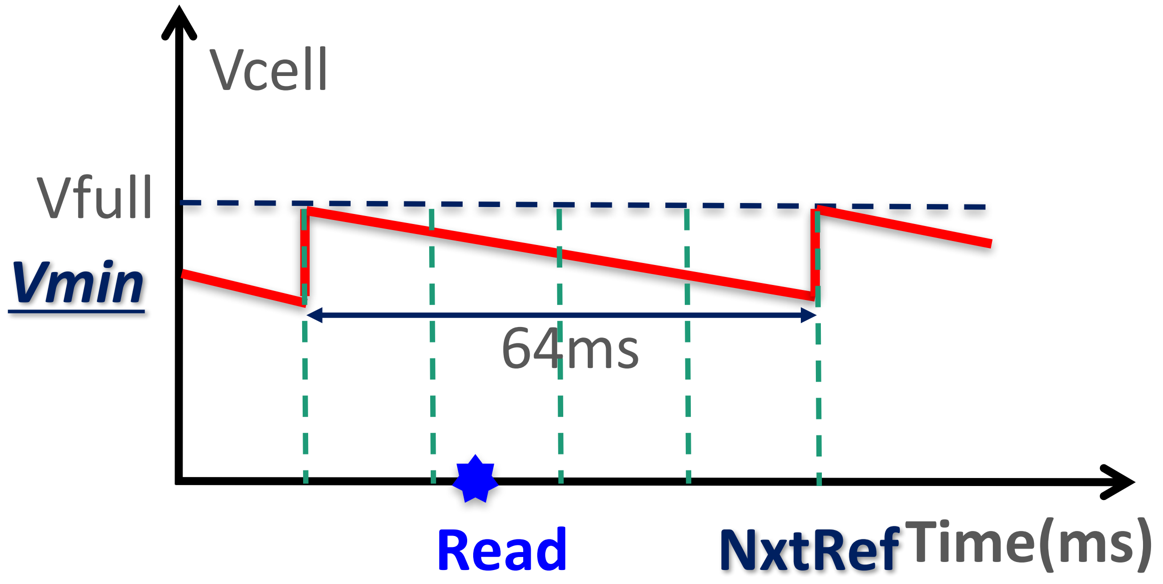
- Data is safe as long as the voltage is above decay curve

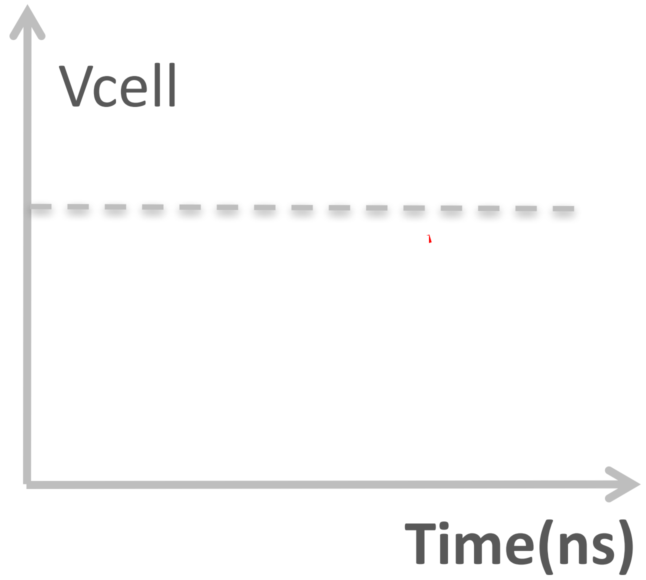
Use four sub-windows

- Save a set of timings for each

Charging goal: Vmax of each sub-window

RT-next: RESTORE W.R.T NEXT REFRESH

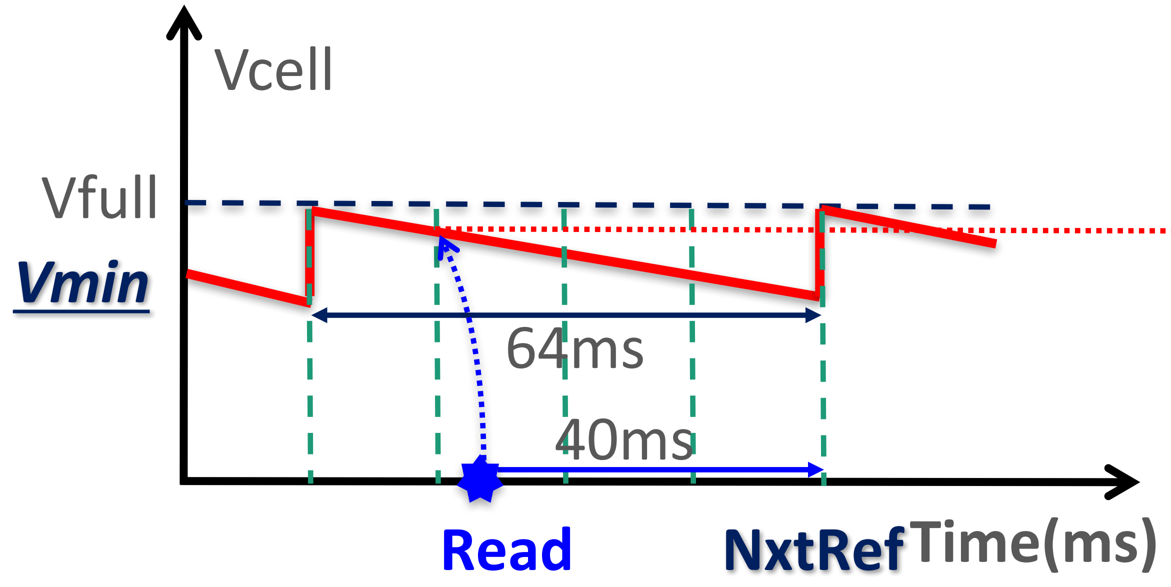


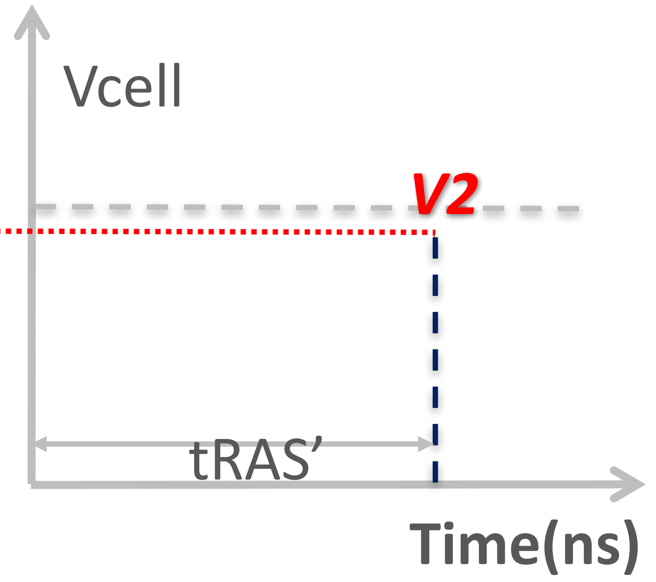


Check the sub-window read/write falls into

Apply the timings to achieve the charging goal

RT-next: RESTORE W.R.T NEXT REFRESH



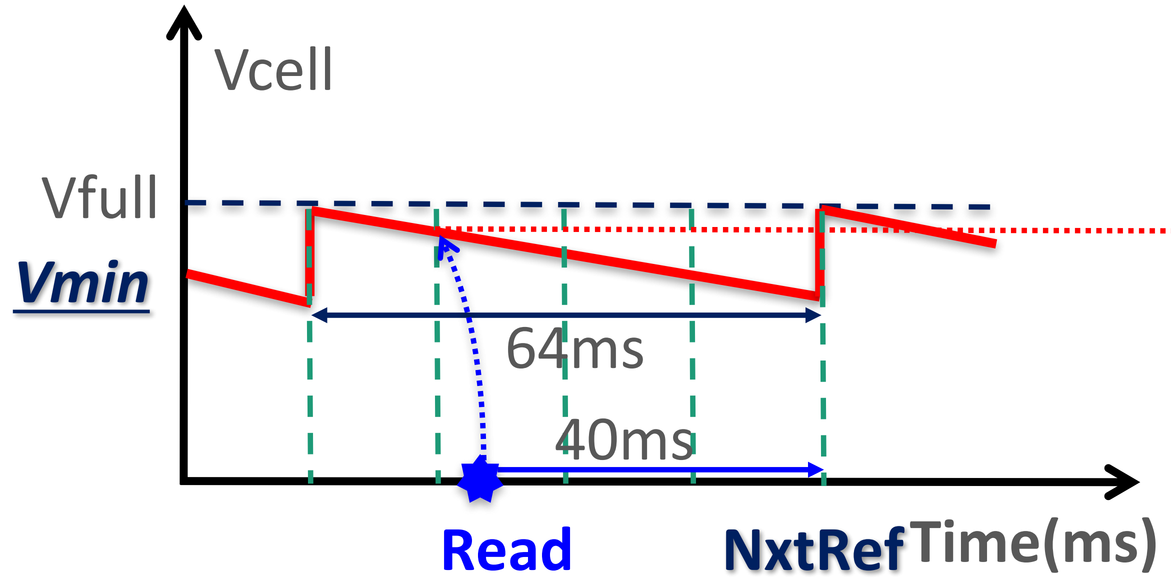


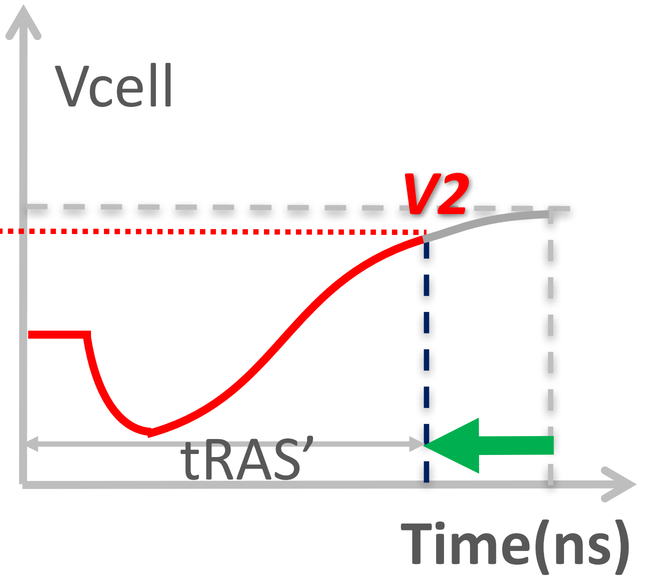
Check the sub-window read/write falls into

Apply the timings to achieve the charging goal

Example: 40ms to the next refresh, 2nd window, charge to V2

RT-next: RESTORE W.R.T NEXT REFRESH



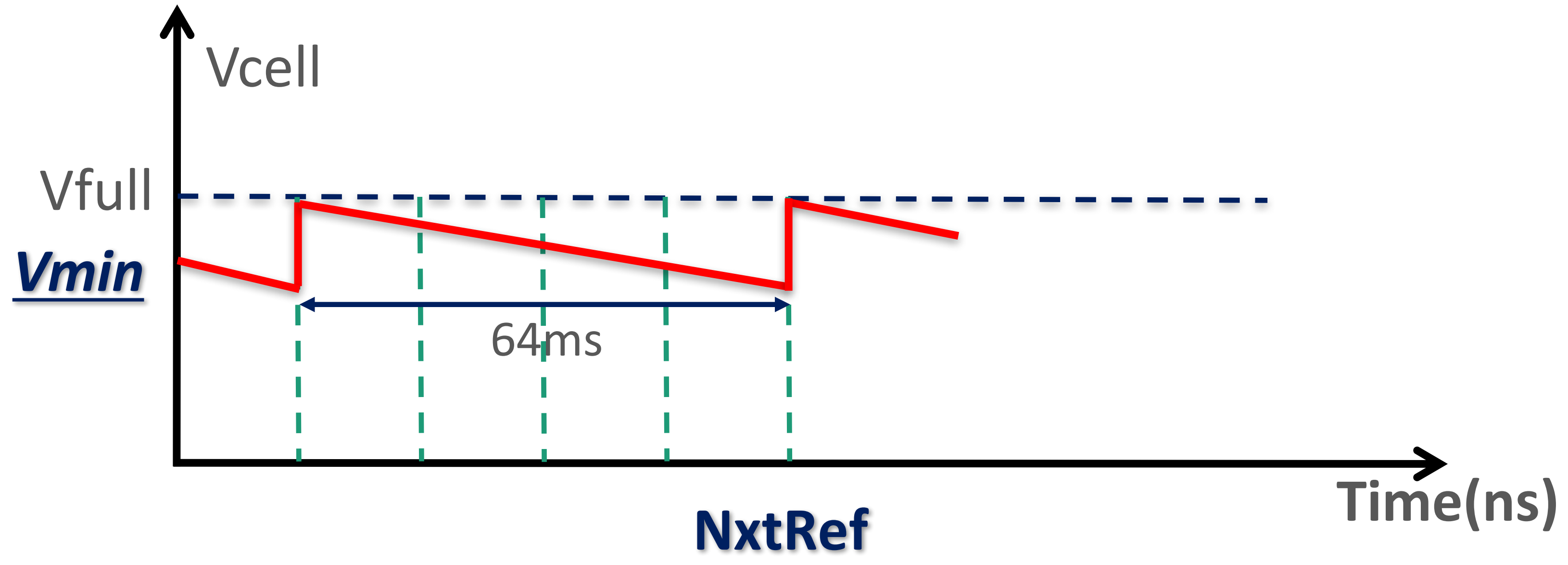


Check the sub-window read/write falls into

Apply the timings to achieve the charging goal

Example: 40ms to the next refresh, 2nd window, charge to V2

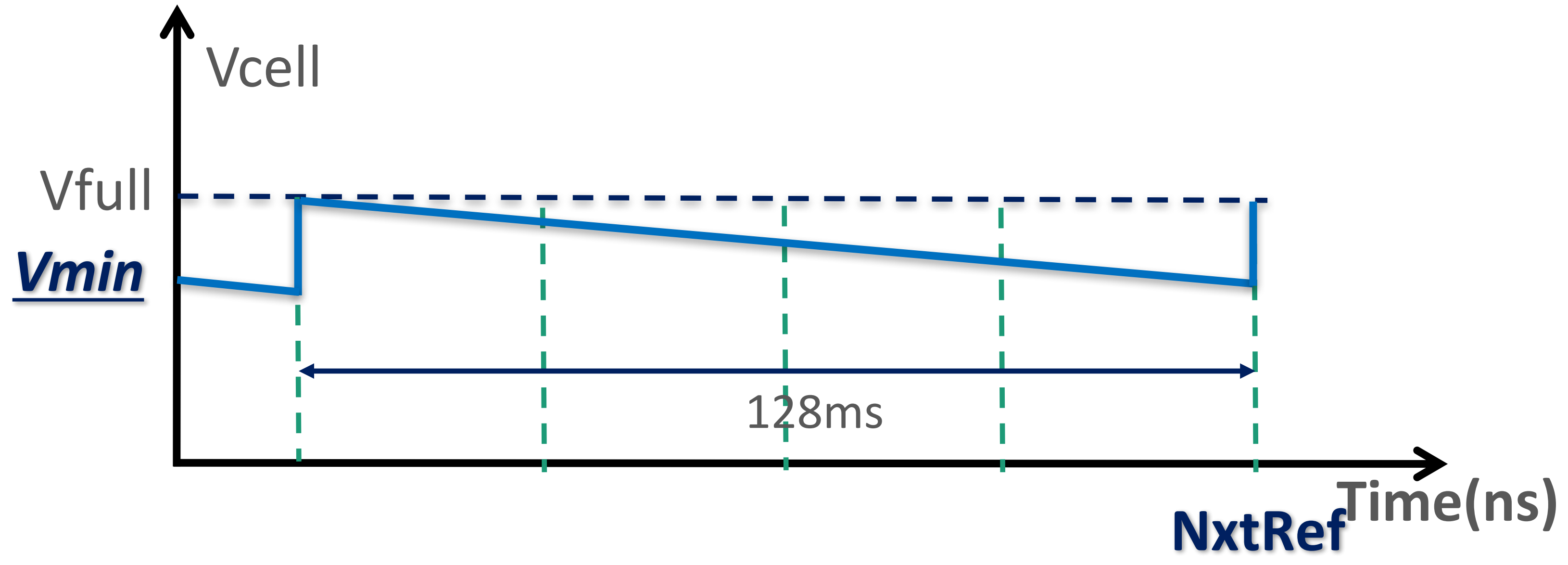
MULTI-RATE REFRESH



Multi-rate refresh

- Over 64ms row, same four-window division

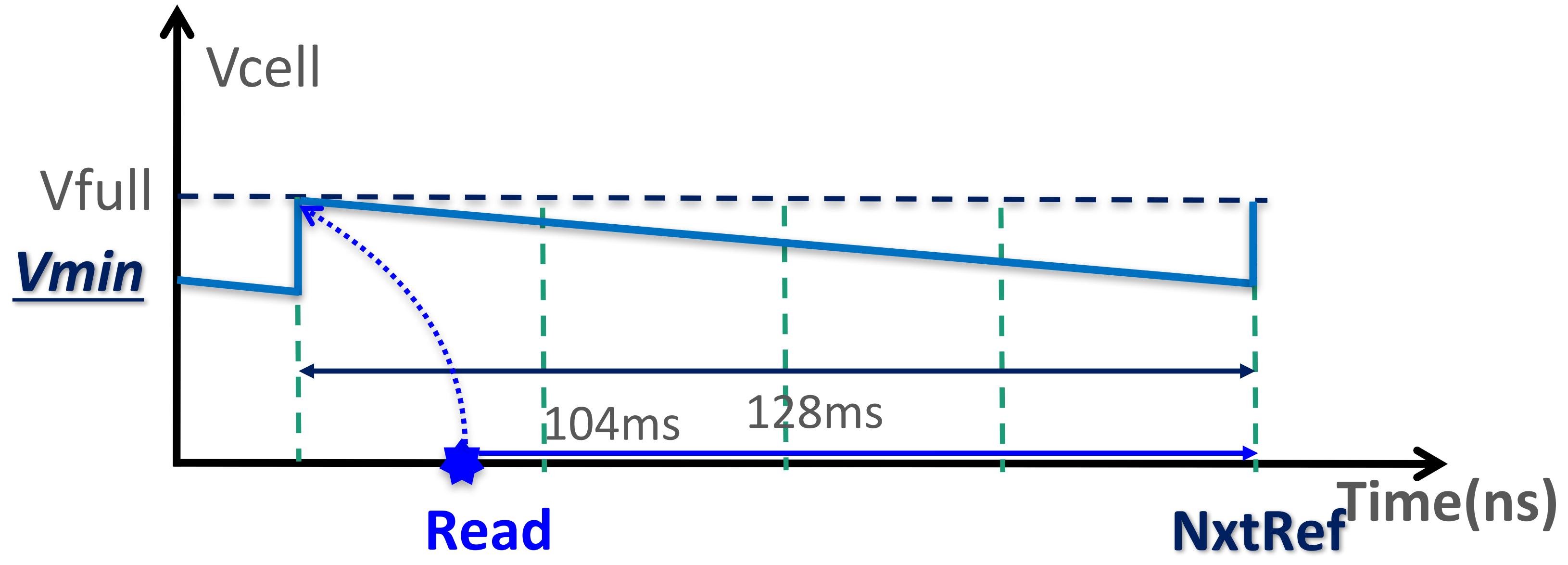
MULTI-RATE REFRESH



Multi-rate refresh

- Over 64ms row, same four-window division

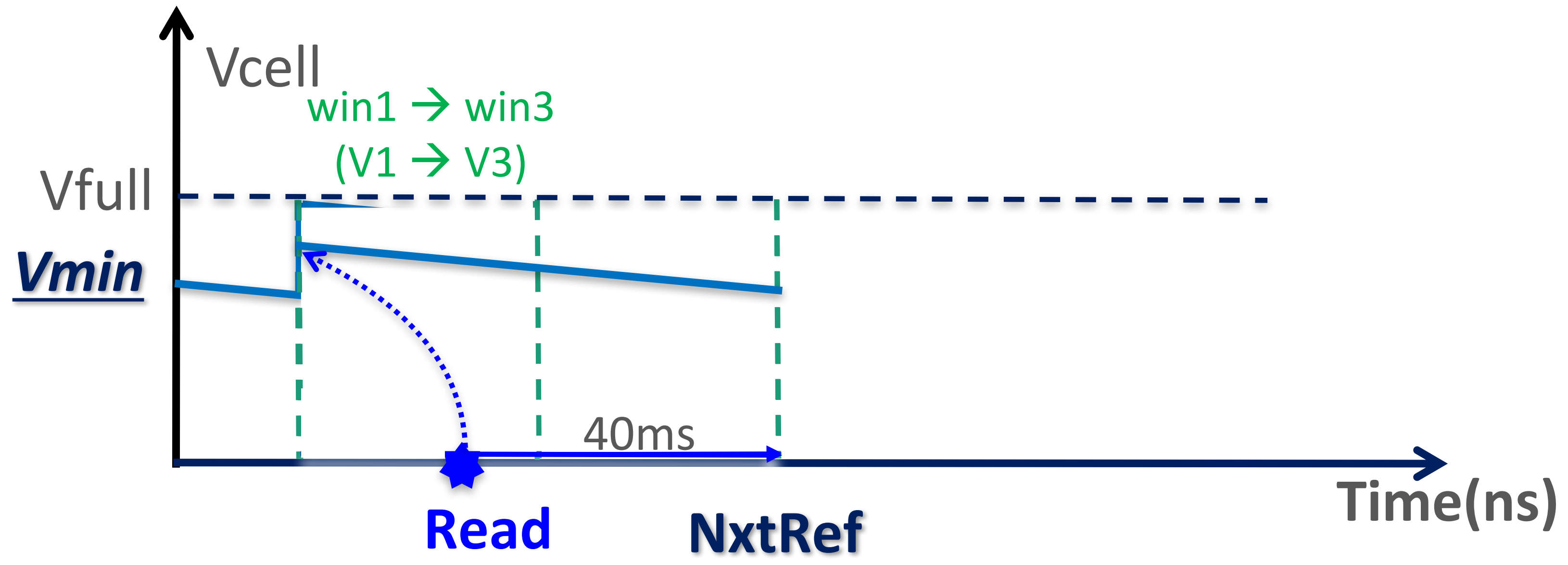
MULTI-RATE REFRESH



Multi-rate refresh

- Over 64ms row, same four-window division

REFRESH UPGRADE



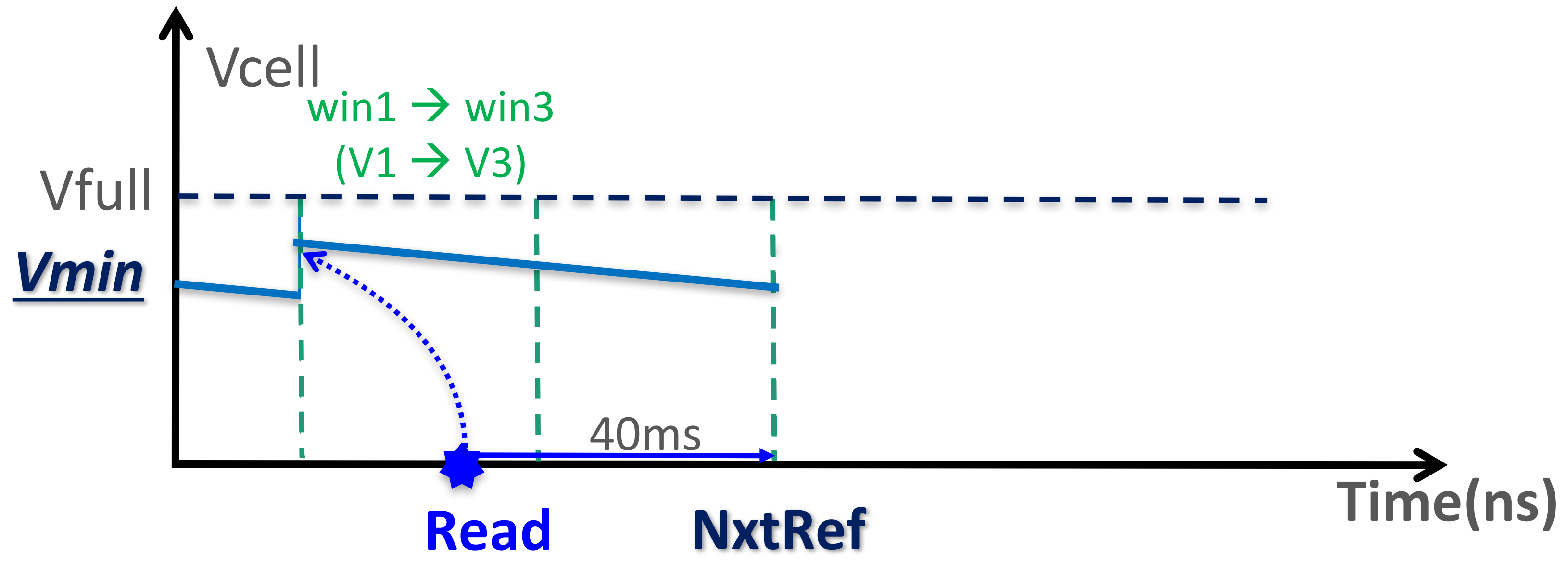
Multi-rate refresh

- Over 64ms row, same four-window division

Refresh upgrade

- More frequent refresh, the closer distance to next refresh
- Lower charging goal for restore

UPGRADE REFRESH DESIGNS



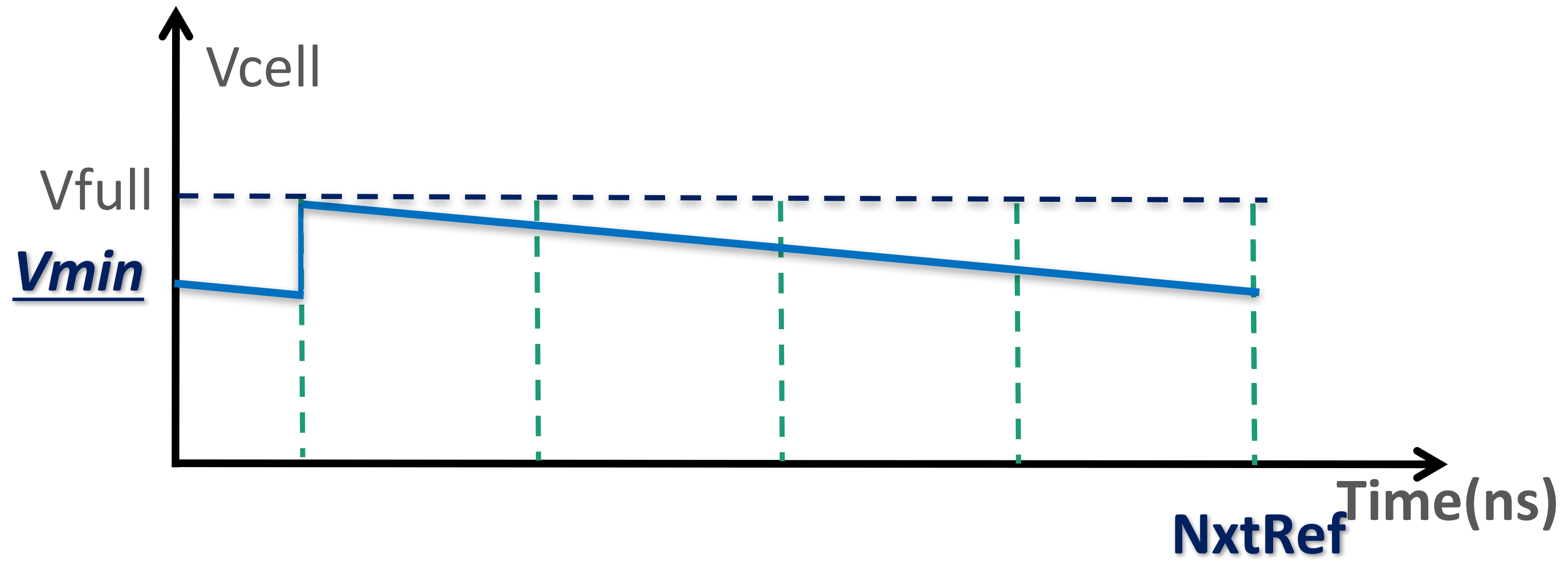
Blindly upgrade (*RT-all*)

- More refreshes, increasing overheads on performance and energy

Selectively upgrade (*RT-sel*)

- Only upgrade touched row/bin
- Back to low-rate afterwards

UPGRADE REFRESH DESIGNS



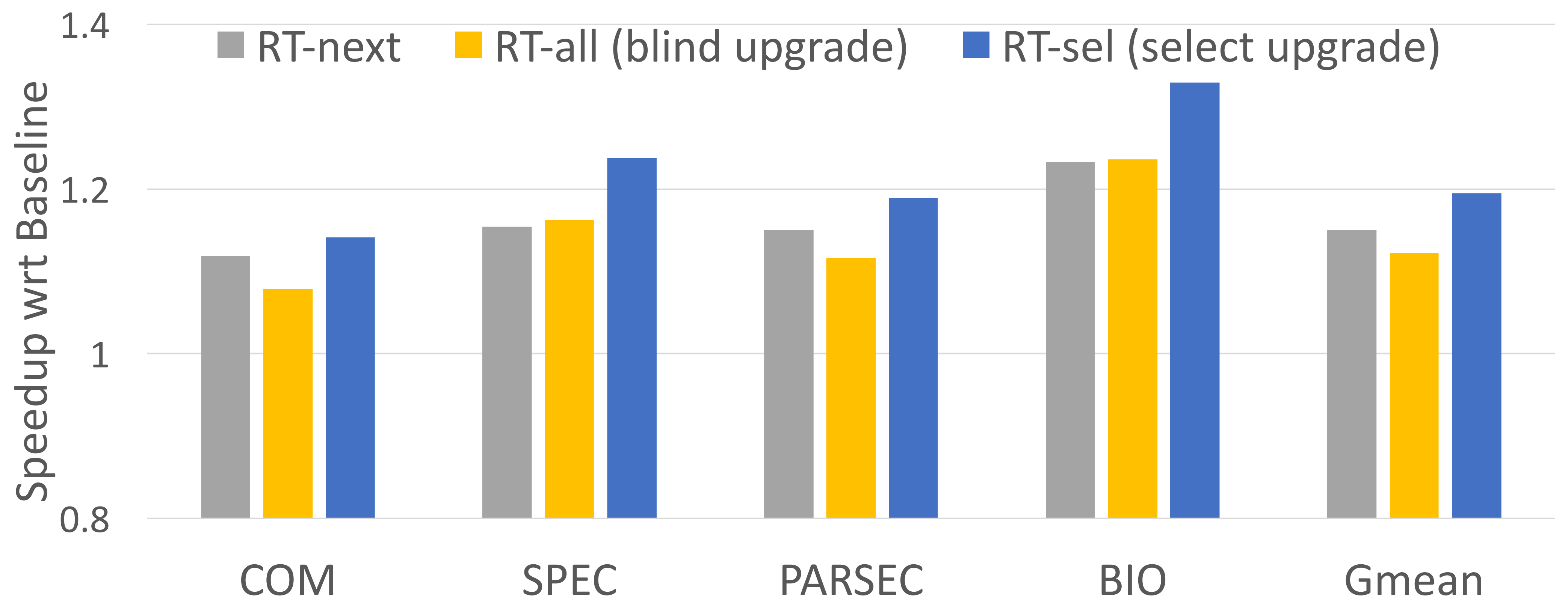
Blindly upgrade (*RT-all*)

- More refreshes, increasing overheads on performance and energy

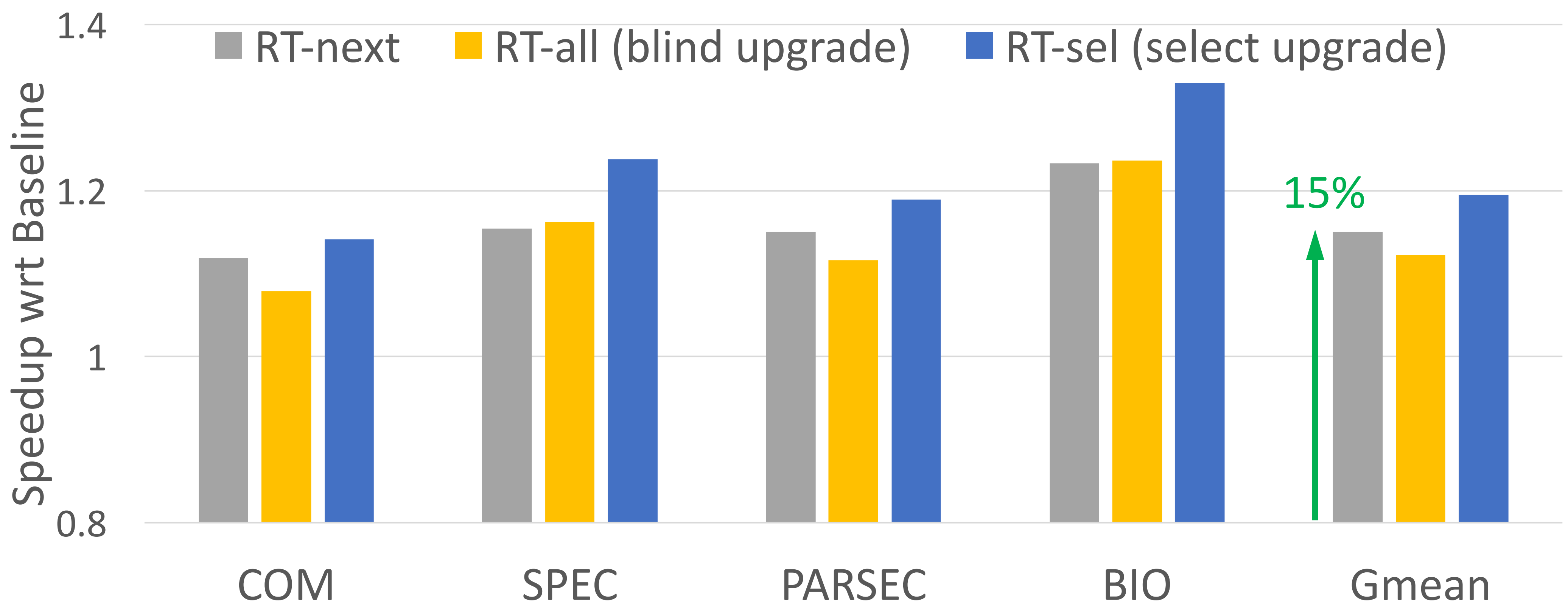
Selectively upgrade (*RT-sel*)

- Only upgrade touched row/bin
- Back to low-rate afterwards

PERFORMANCE

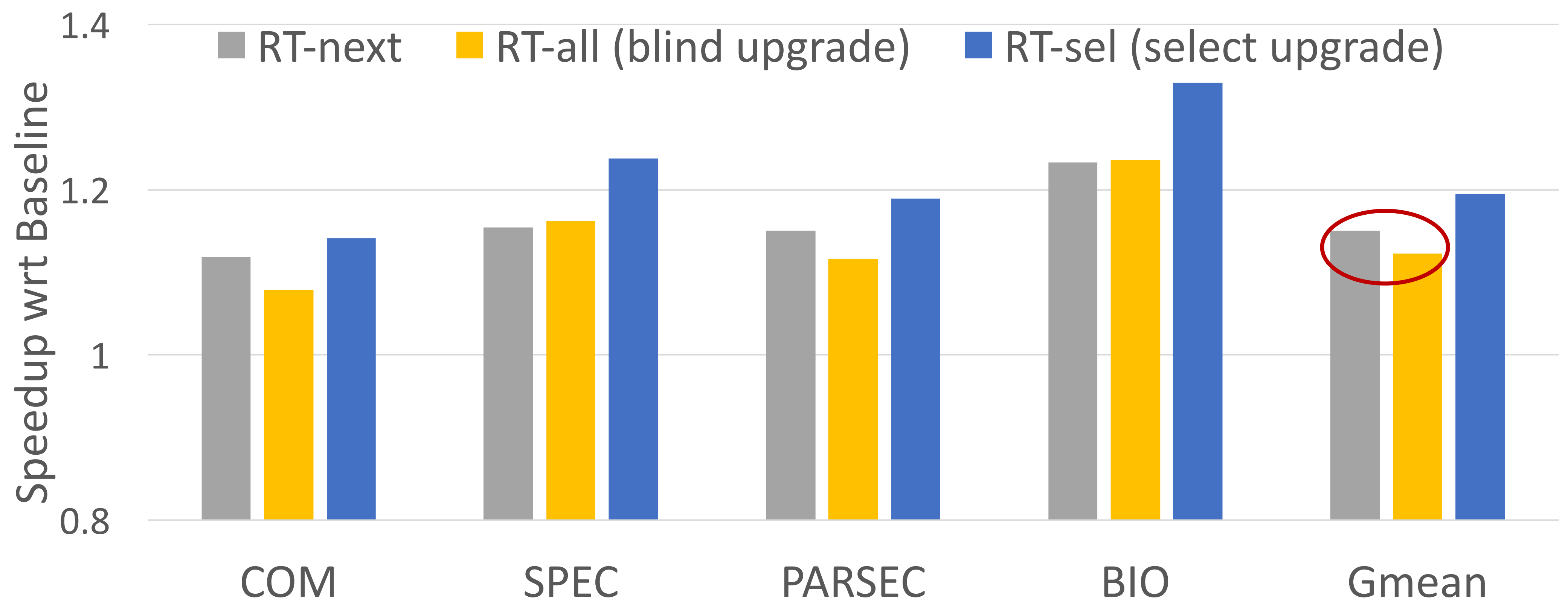


PERFORMANCE



RT-next is 15% over Baseline because of restore truncation

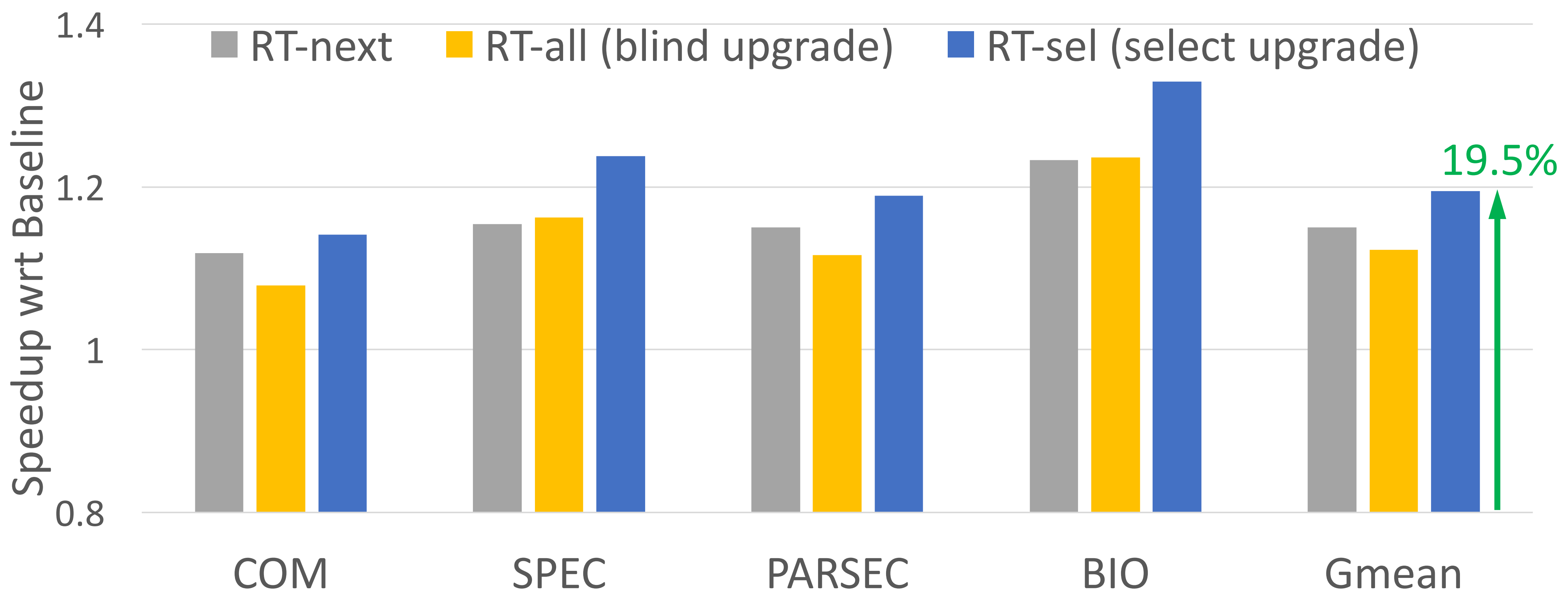
PERFORMANCE



RT-next is 15% over Baseline because of restore truncation

RT-all becomes worse because of refresh penalty

PERFORMANCE

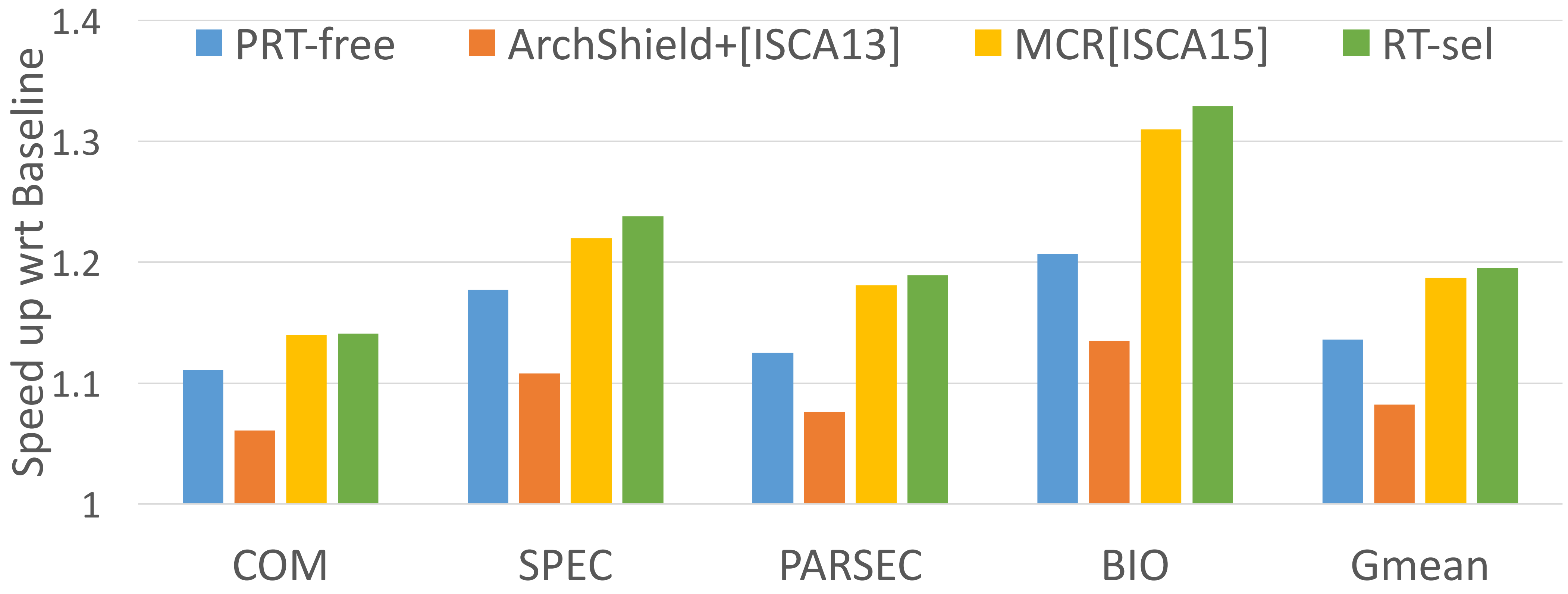


RT-next is 15% over Baseline because of restore truncation

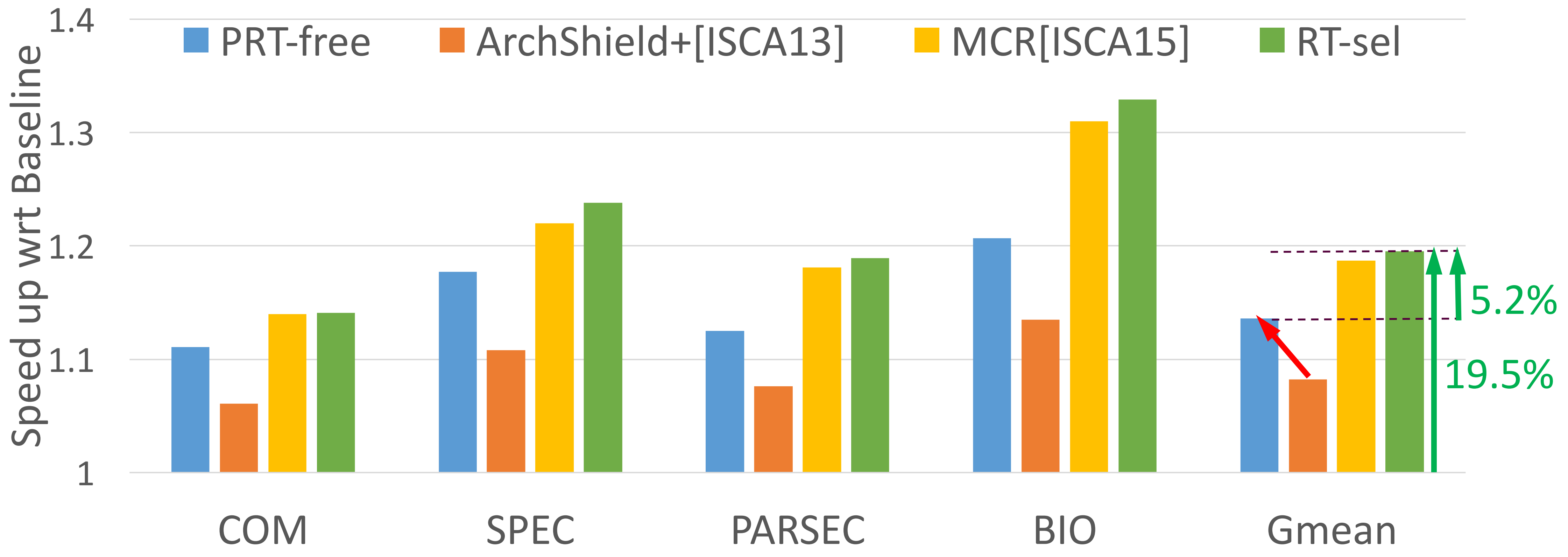
RT-all becomes worse because of refresh penalty

RT-sel achieves the best result by balancing refresh and restore

COMPARE TO STATE-OF-ARTS

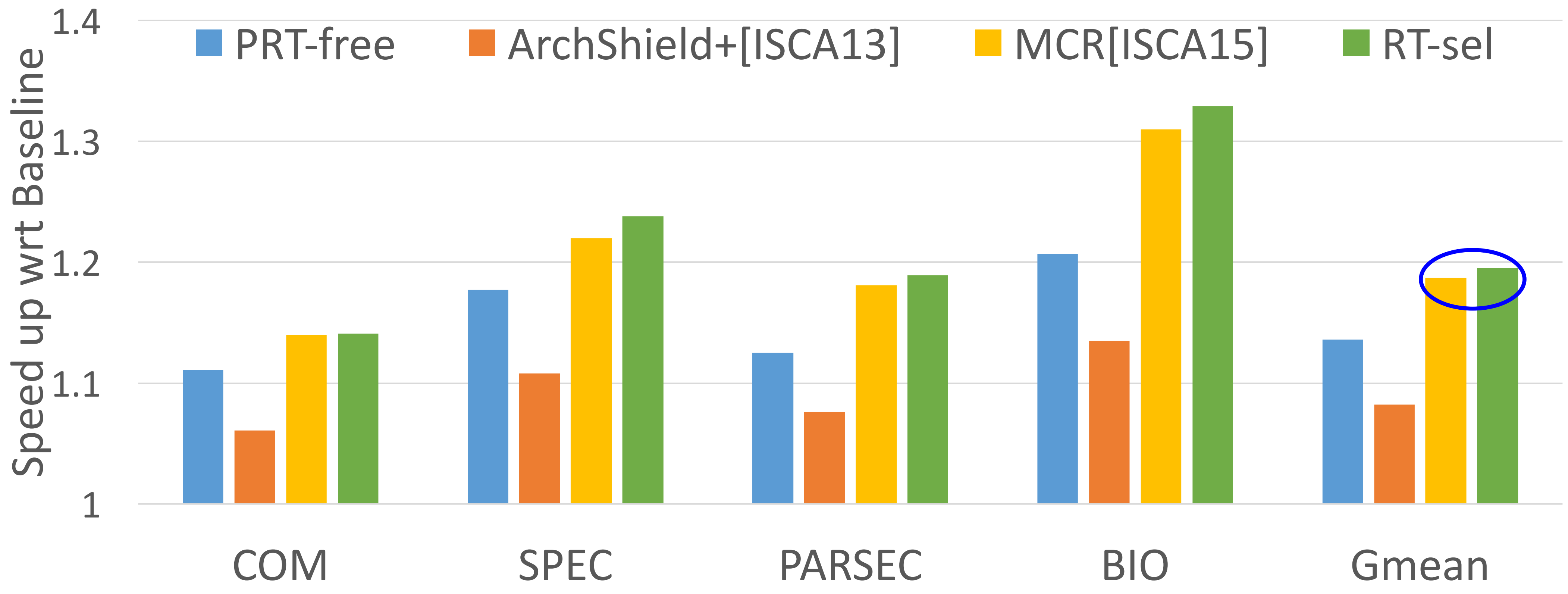


COMPARE TO STATE-OF-ARTS



While ArchShield+ is close to PRT-free, RT-sel is 5.2% better

COMPARE TO STATE-OF-ARTS



While ArchShield+ is close to PRT-free, RT-sel is 5.2% better

While losing 50% capacity, MCR is still worse

SUMMARY: RT-

Prolonged restore issue in future DRAM
Restore and refresh are strongly correlated

RT-next: truncate restore w/ refresh distance
RT-sel: expose more restore opportunities

Balances refresh and restore, beats state-of-arts
Performance: 19.5% improvement

OUTLINE

RT-Next

Partial restore based on refresh distance

CkRemap

Fast restore via reorganization and allocation

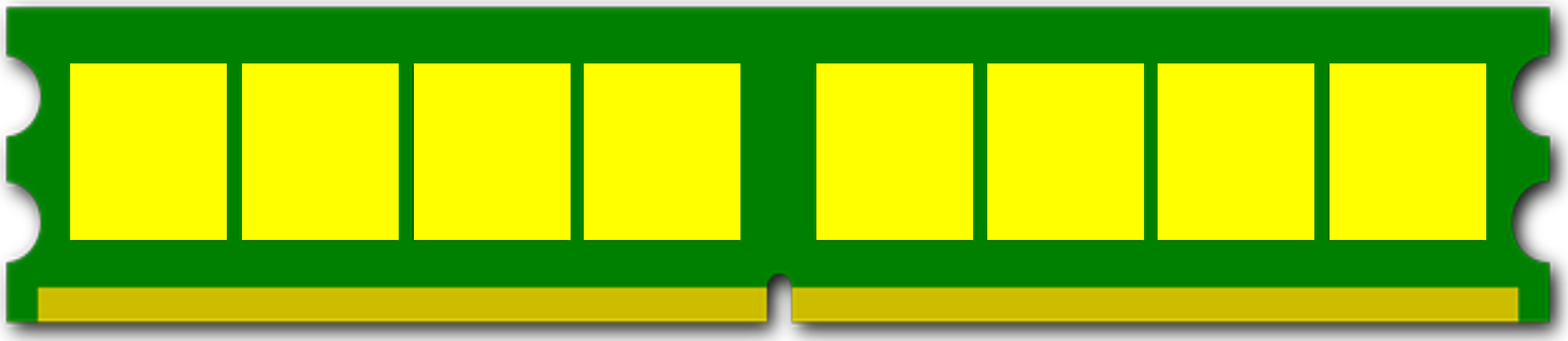
DrMP

Mitigate restore with approximate computing

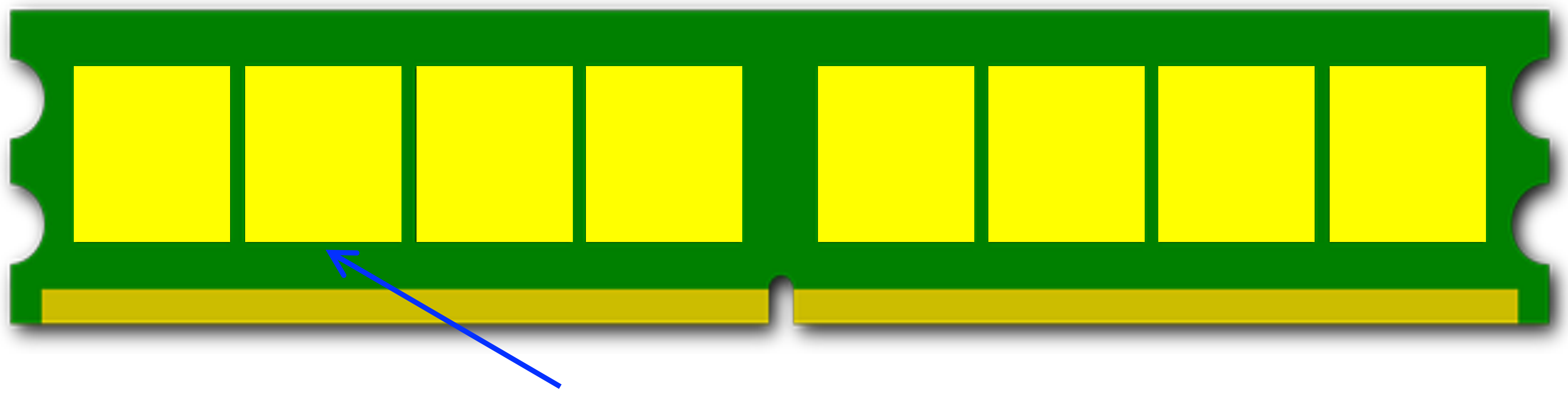
Summary and Research Directions

OUTLINE

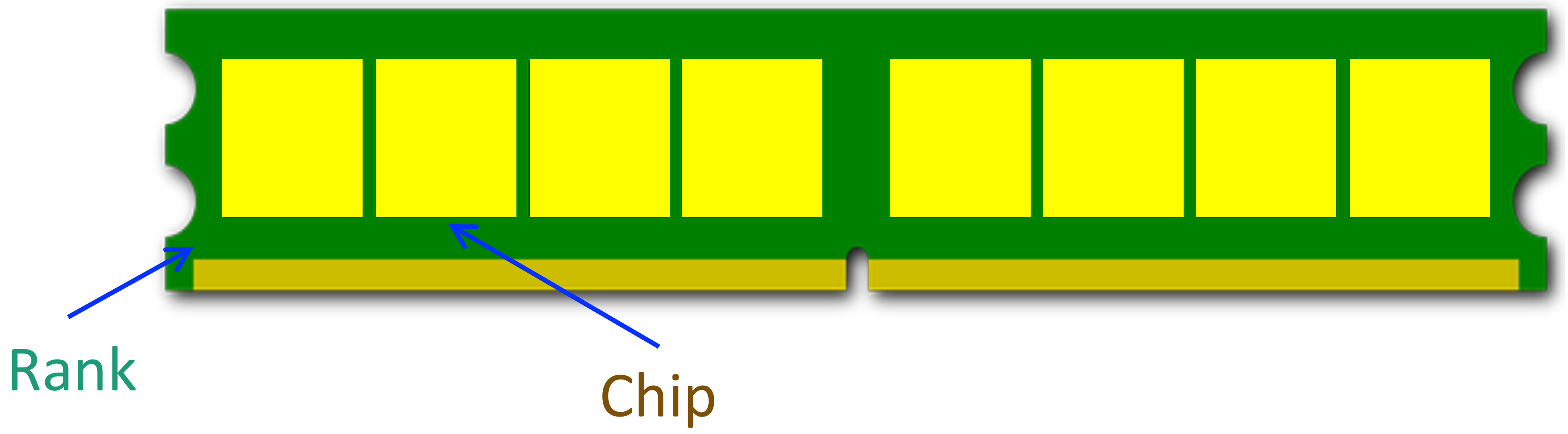
DRAM ORGANIZATION



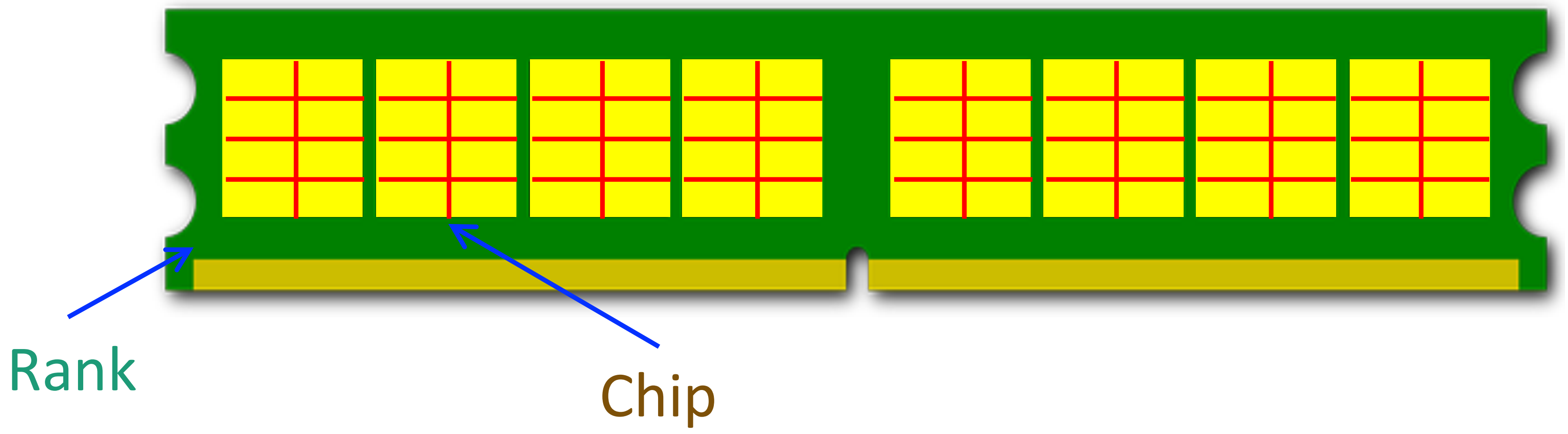
DRAM ORGANIZATION



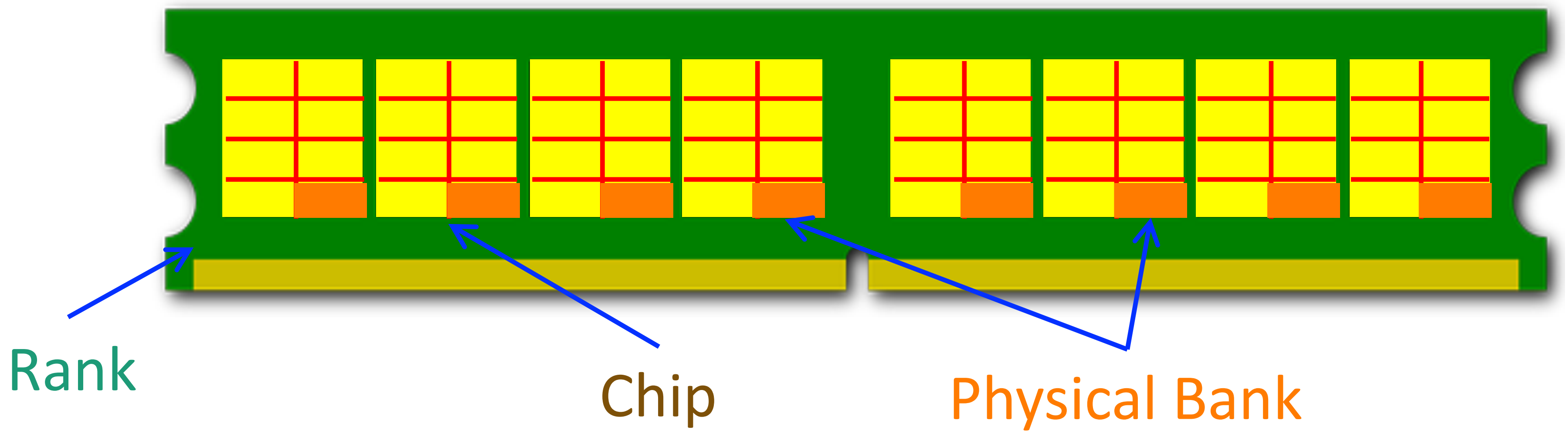
DRAM ORGANIZATION



DRAM ORGANIZATION

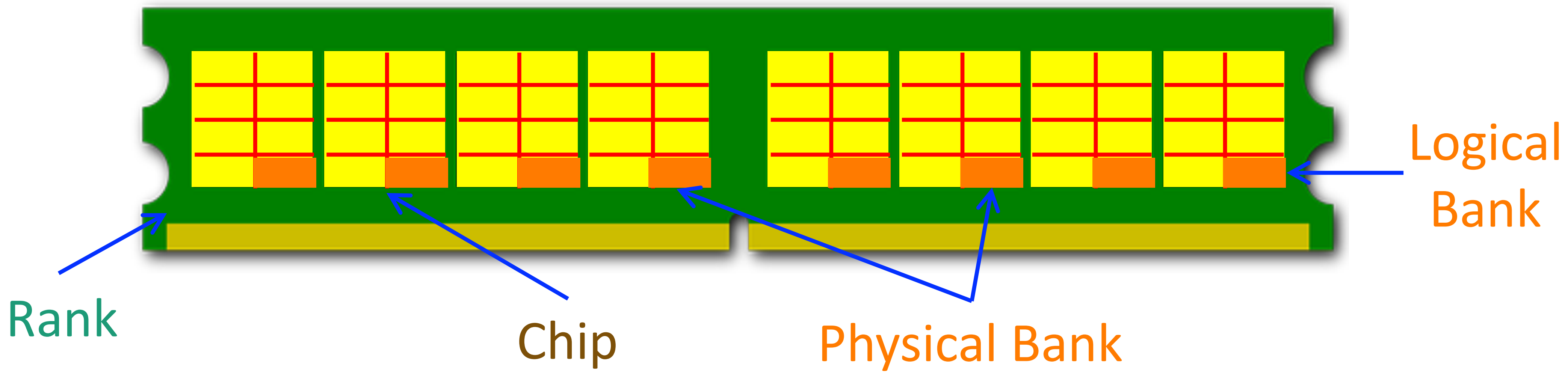


DRAM ORGANIZATION



Physical bank: chip level, a portion of memory arrays

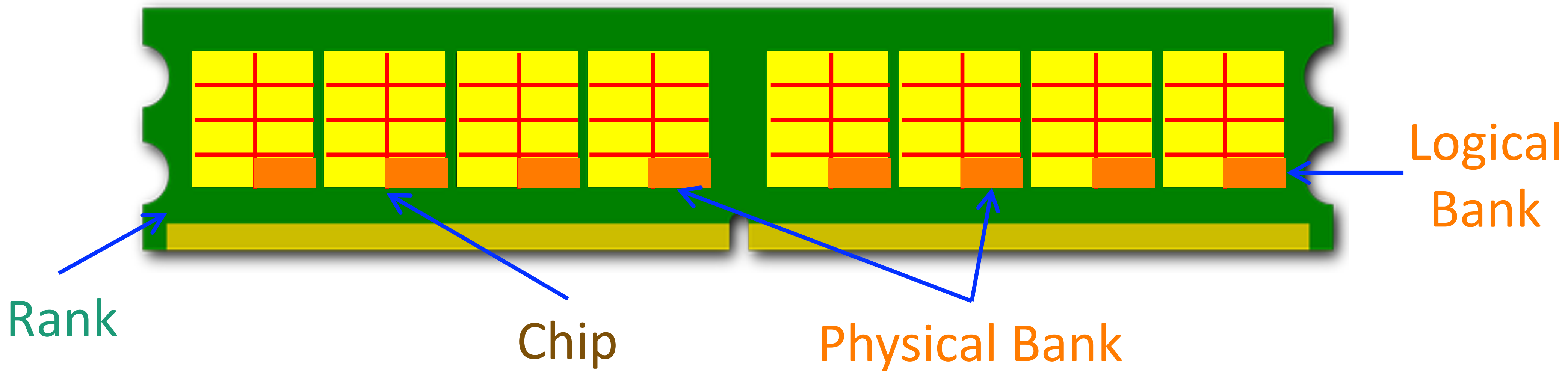
DRAM ORGANIZATION



Physical bank: chip level, a portion of memory arrays

Logical bank: rank level, one physical bank from each chip

DRAM ORGANIZATION

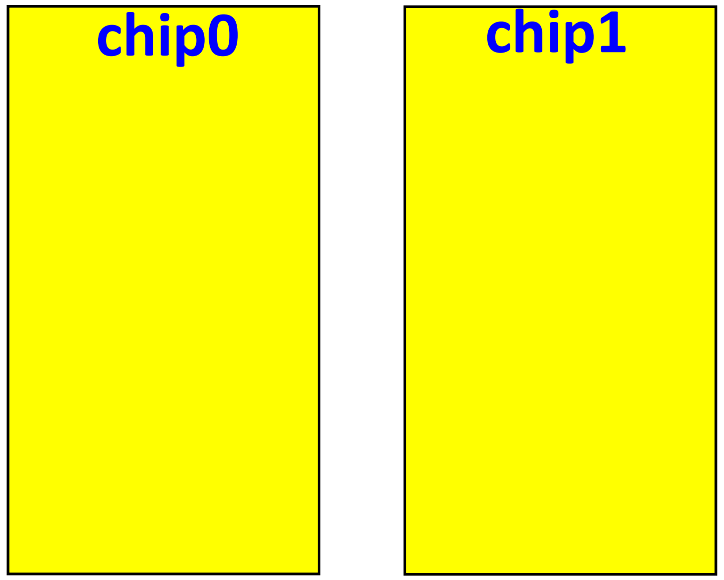


Physical bank: chip level, a portion of memory arrays

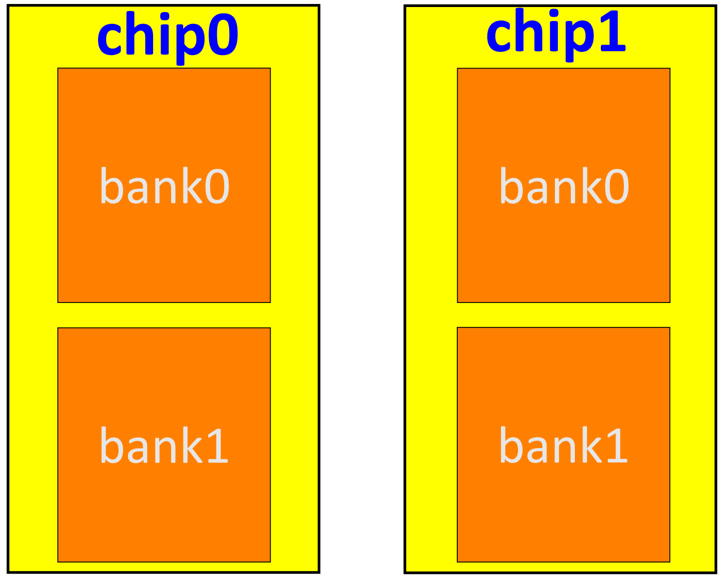
Logical bank: rank level, one physical bank from each chip

How to utilize the organization to solve restore?

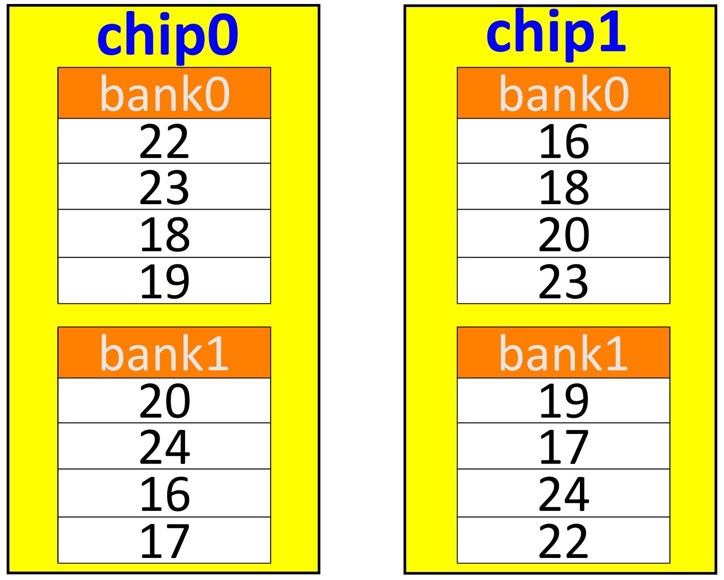
MOTIVATION



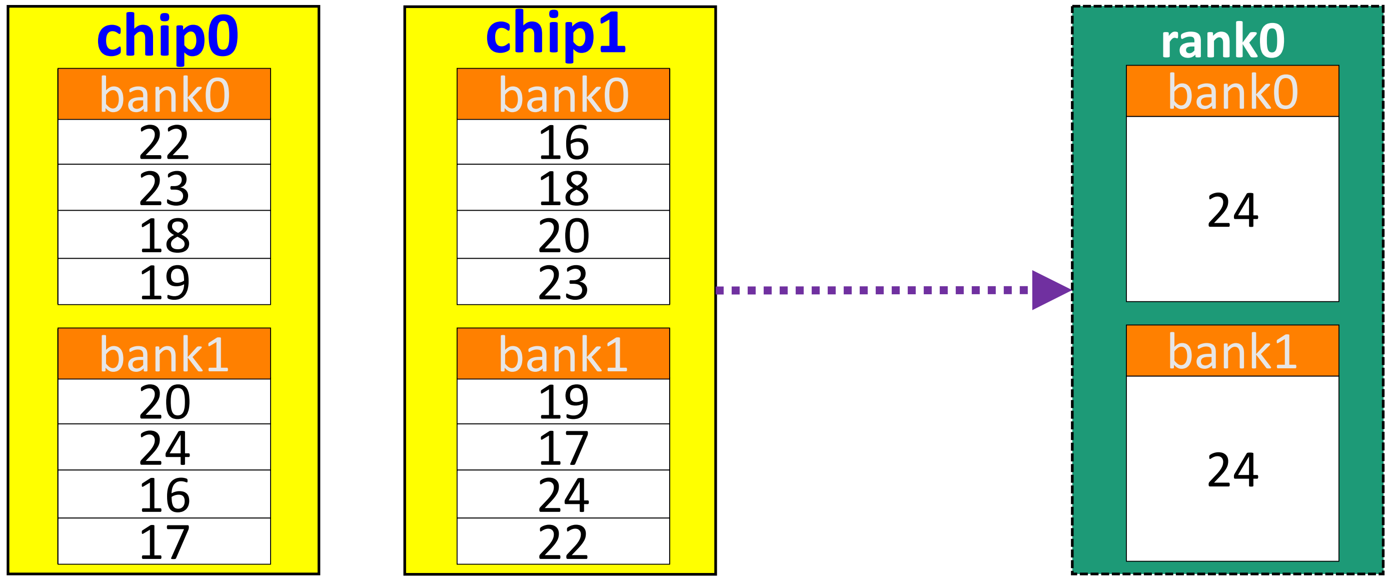
MOTIVATION



MOTIVATION

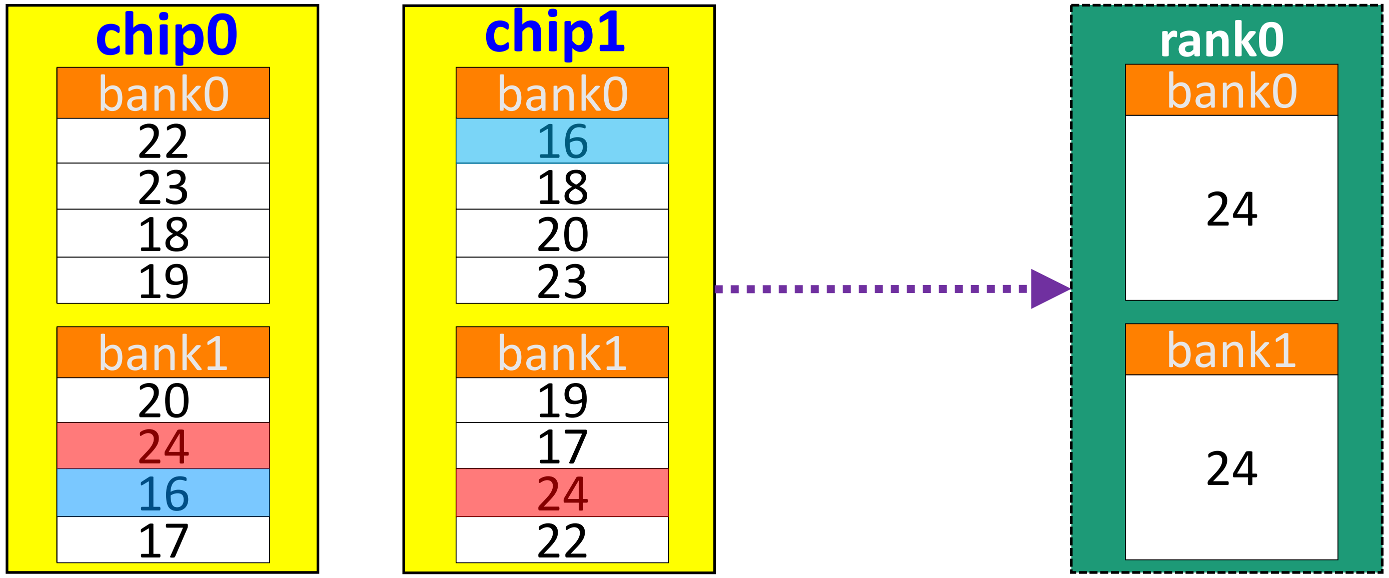

chip0
bank0
22
23
18
19
bank1
20
24
16
17
chip1
bank0
16
18
20
23
bank1
19
17
24
22

MOTIVATION



Single set of timings for the whole memory

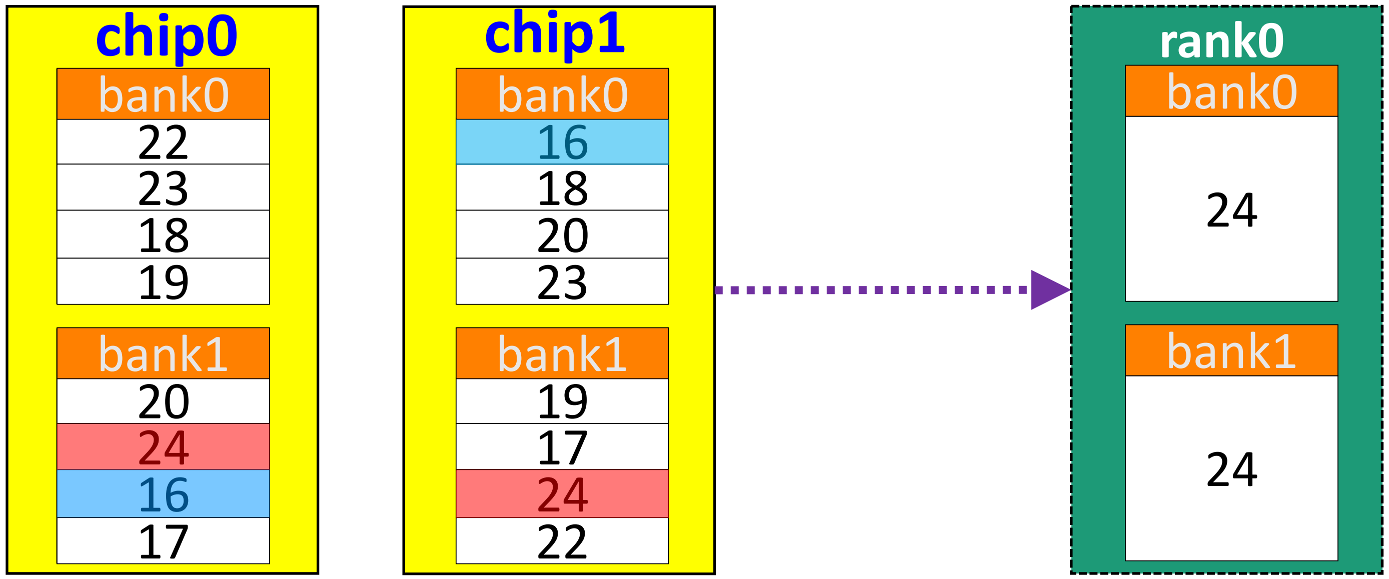
MOTIVATION



Single set of timings for the whole memory

Cells are more statistical in smaller nodes

MOTIVATION

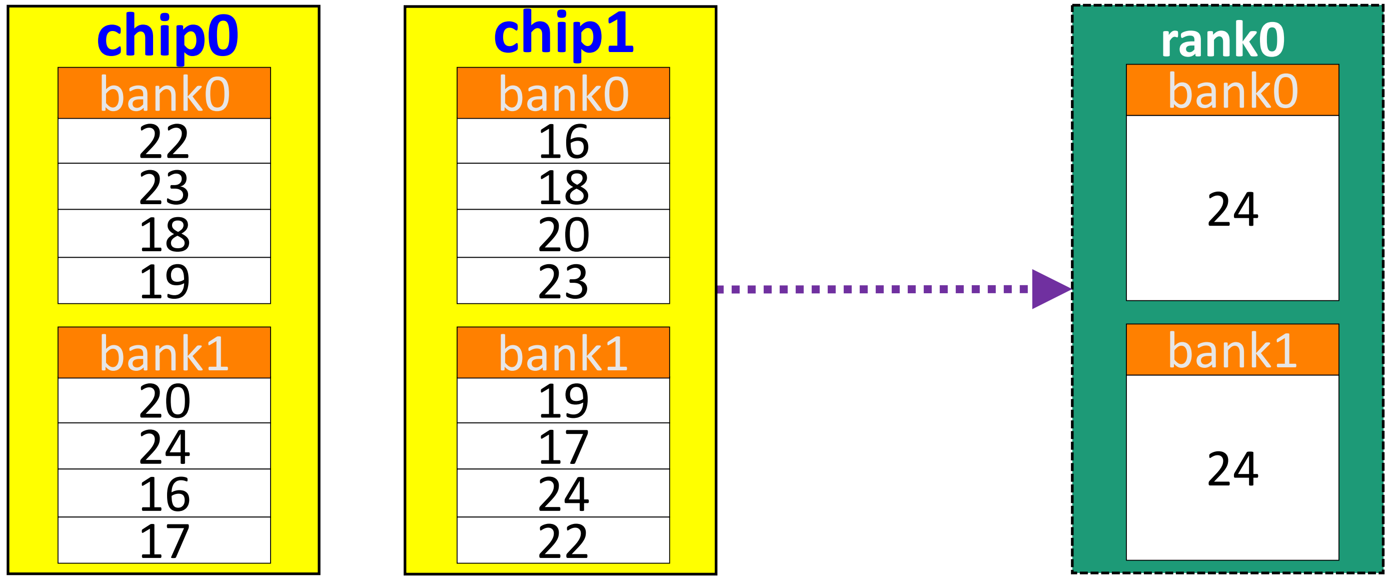


Single set of timings for the whole memory

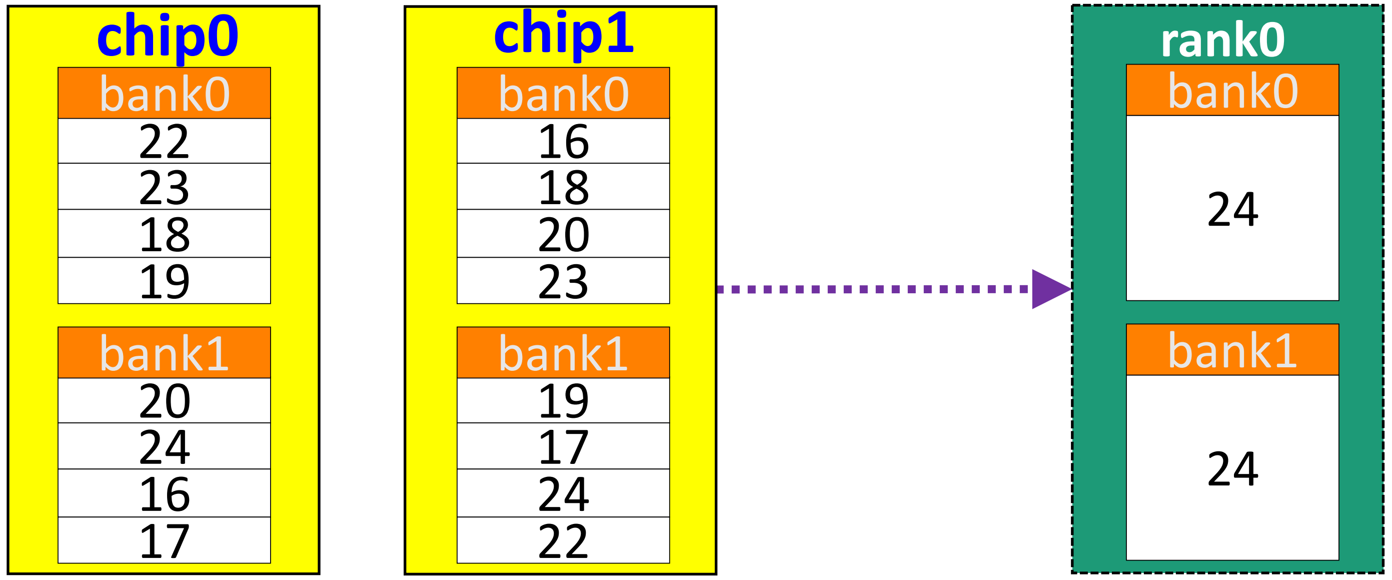
Cells are more statistical in smaller nodes

Too pessimistic to decide by the worst case

CHUNK-SPECIFIC RESTORE



CHUNK-SPECIFIC RESTORE

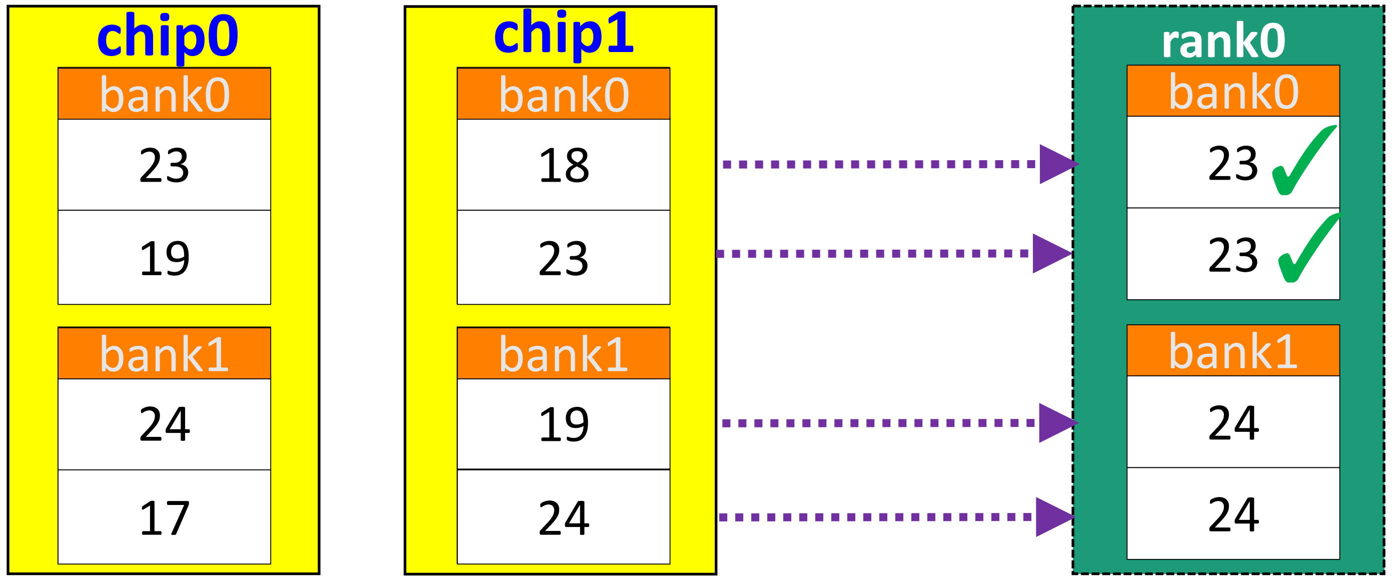


Partition each chip bank into multi chunks

Set chunk-level timings

Expose timings to memory controller (MC)

CHUNK-SPECIFIC RESTORE

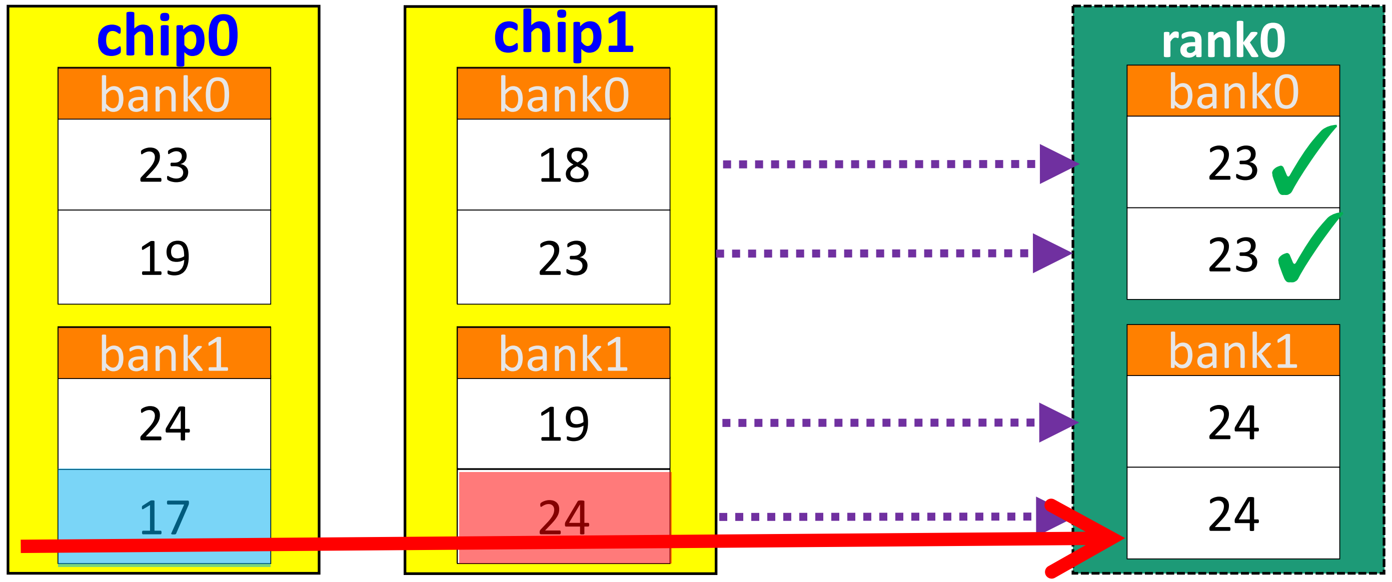


Partition each chip bank into multi chunks

Set chunk-level timings

Expose timings to memory controller (MC)

CHUNK-SPECIFIC RESTORE



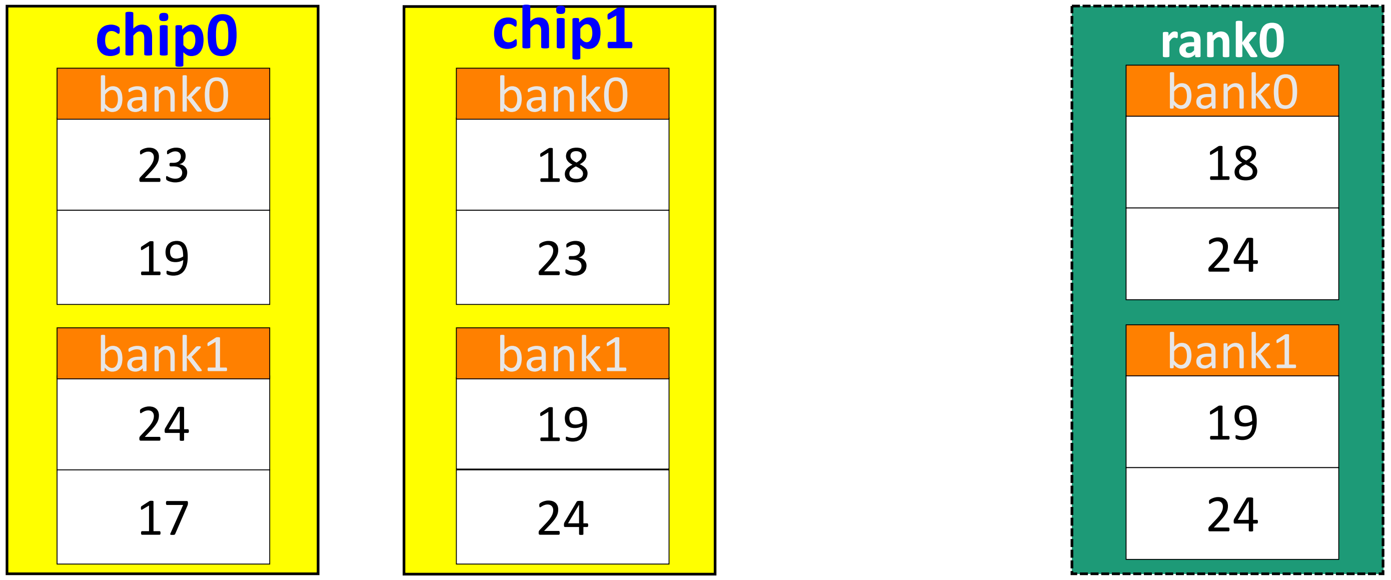
Partition each chip bank into multi chunks

Set chunk-level timings

Expose timings to memory controller (MC)

Slow & fast chunks can still be combined together

FAST CHUNK W/ REMAPPING

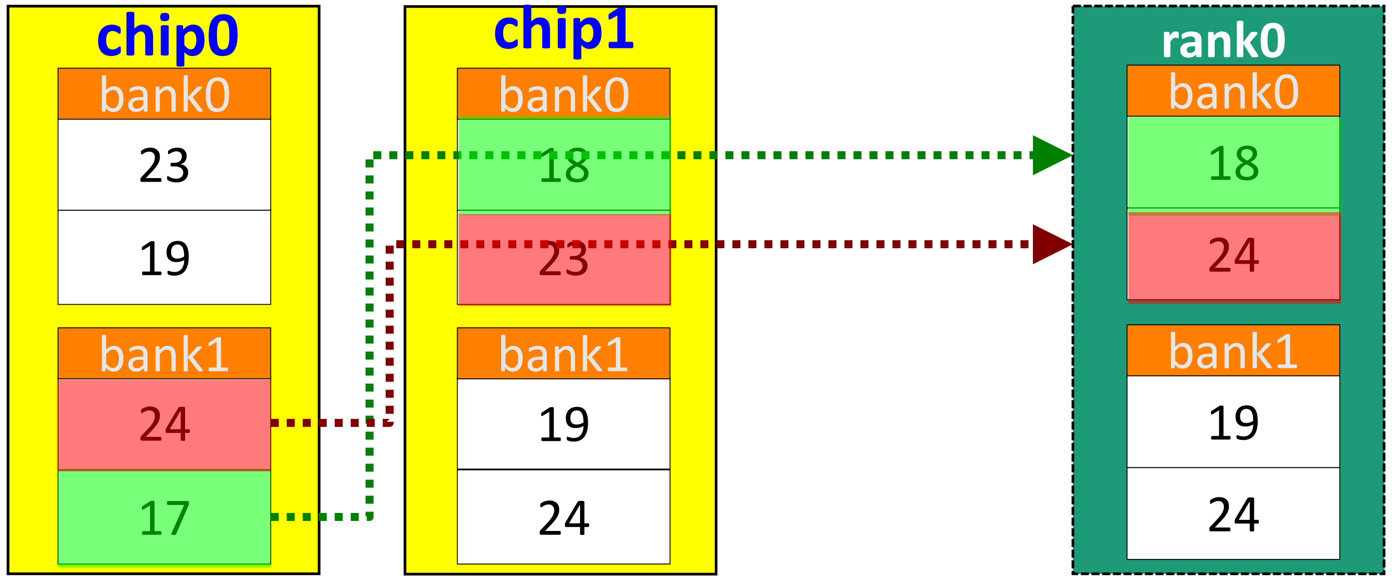


Partition bank into chunks

Detect chip-chunk timings

Remap chunks within each chip-bank

FAST CHUNK W/ REMAPPING

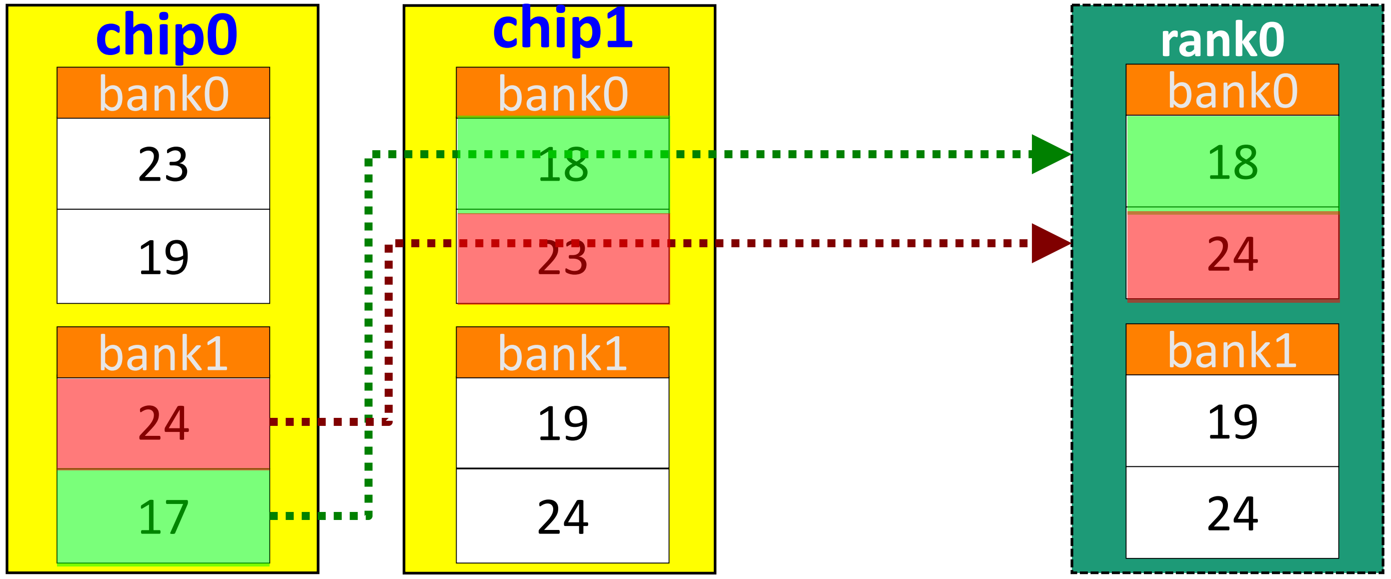


Partition bank into chunks

Detect chip-chunk timings

Remap chunks within each chip-bank

FAST CHUNK W/ REMAPPING



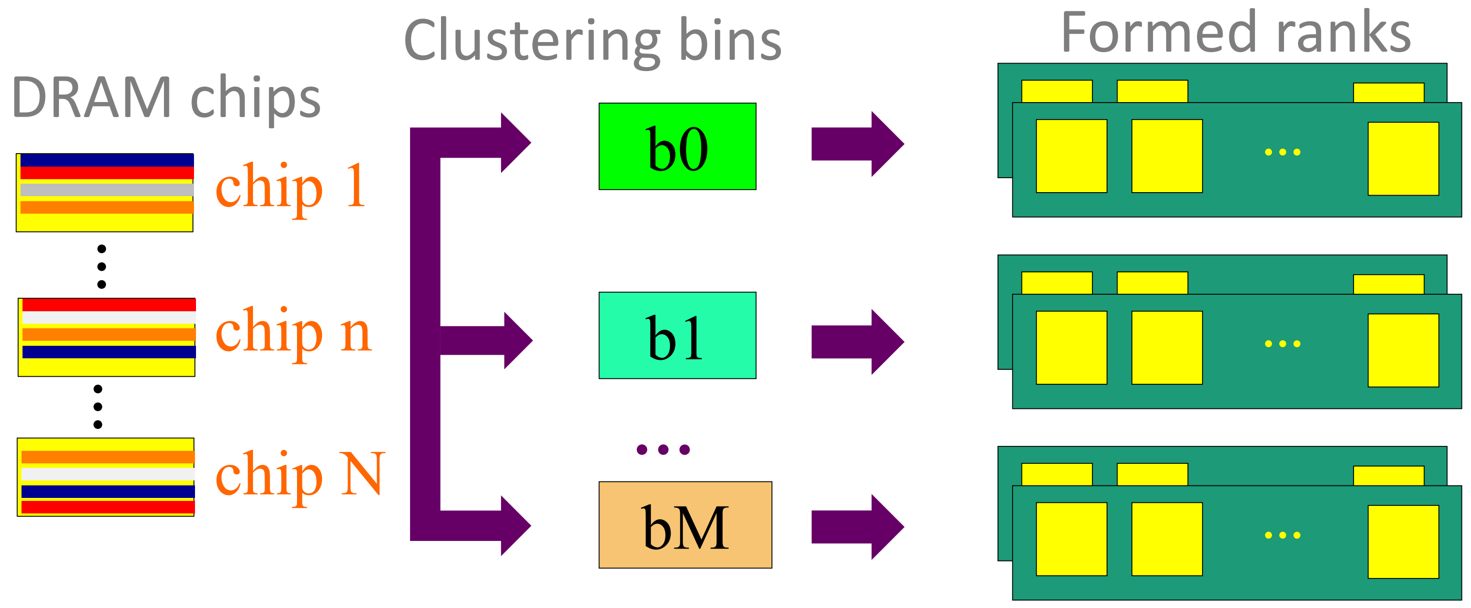
Partition bank into chunks

Detect chip-chunk timings

Remap chunks within each chip-bank

Bad chip leads to slow rank even w/ remapping

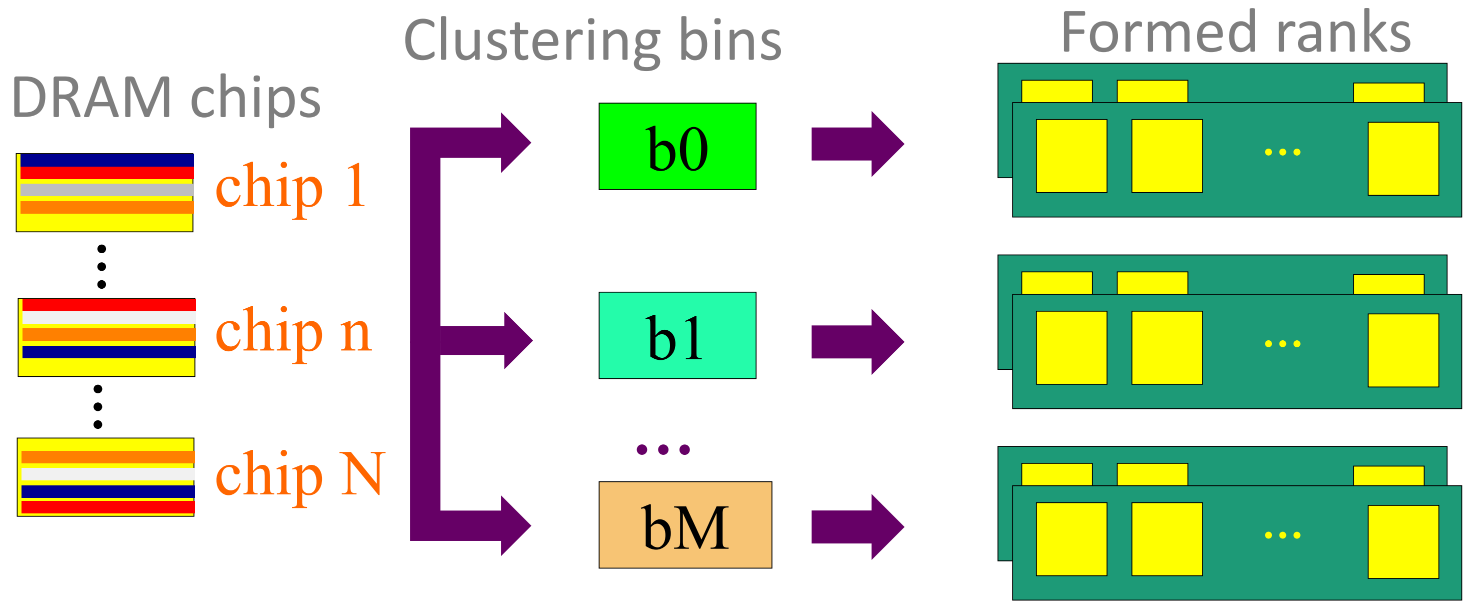
RANK CONSTRUCTION (BIN)



Cluster chips into bins using similarity

Construct ranks using chips from each bin

RANK CONSTRUCTION (BIN)

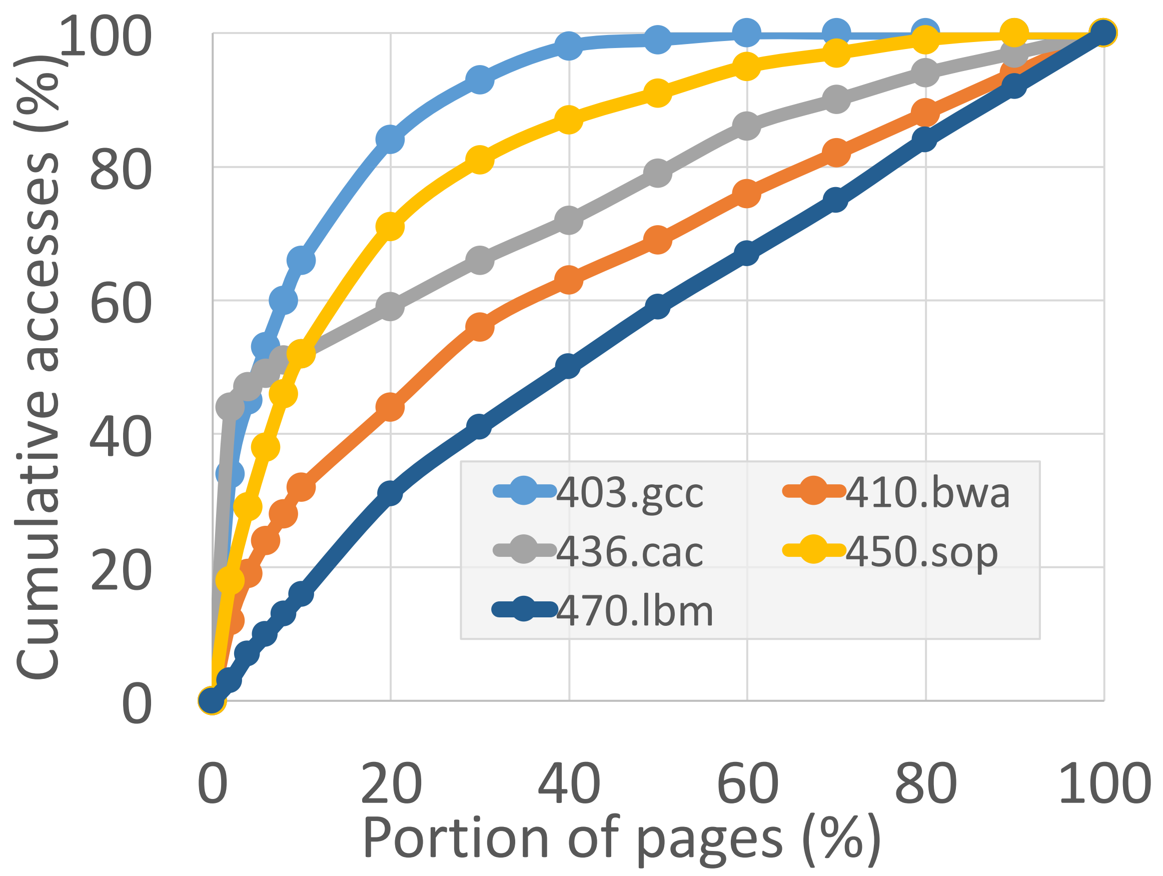


Cluster chips into bins using similarity

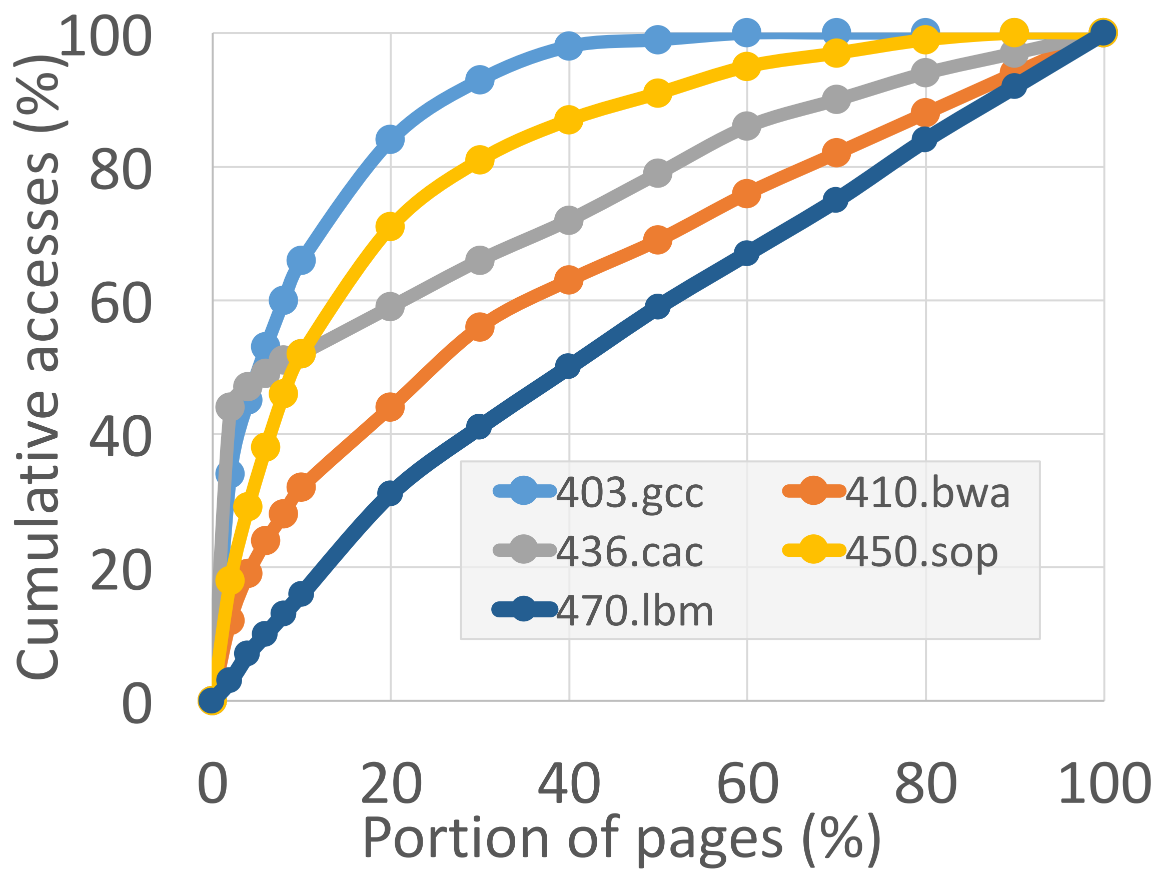
Construct ranks using chips from each bin

How to fully utilize the exposed fast regions?

RESTORE-AWARE PAGE ALLOCATION

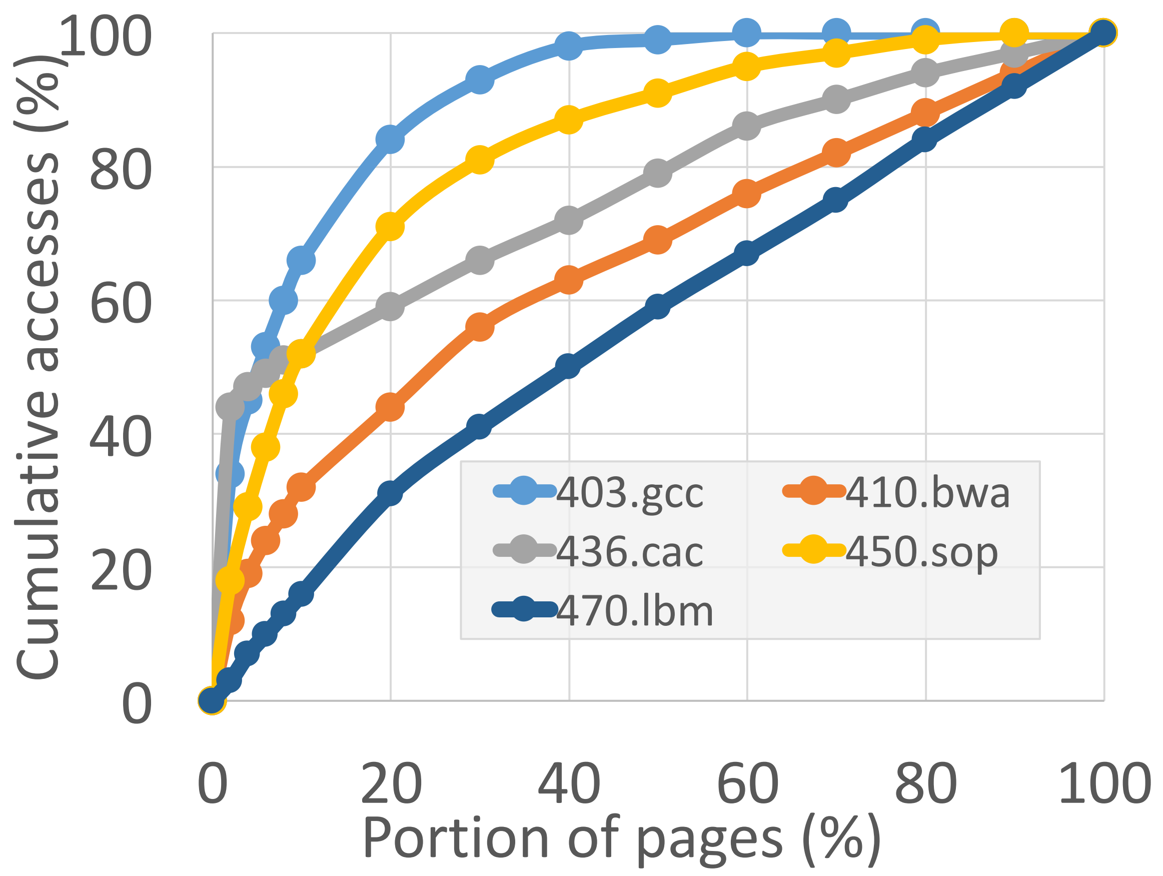


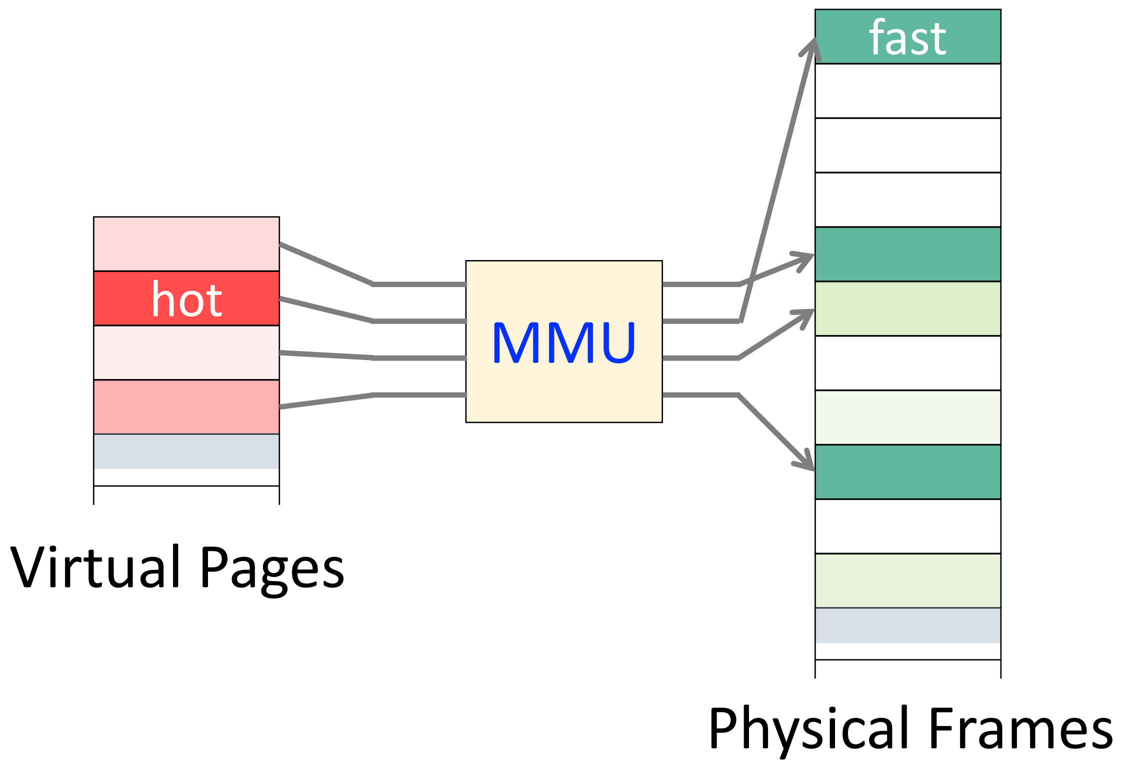
RESTORE-AWARE PAGE ALLOCATION



Accesses come from a small set of pages

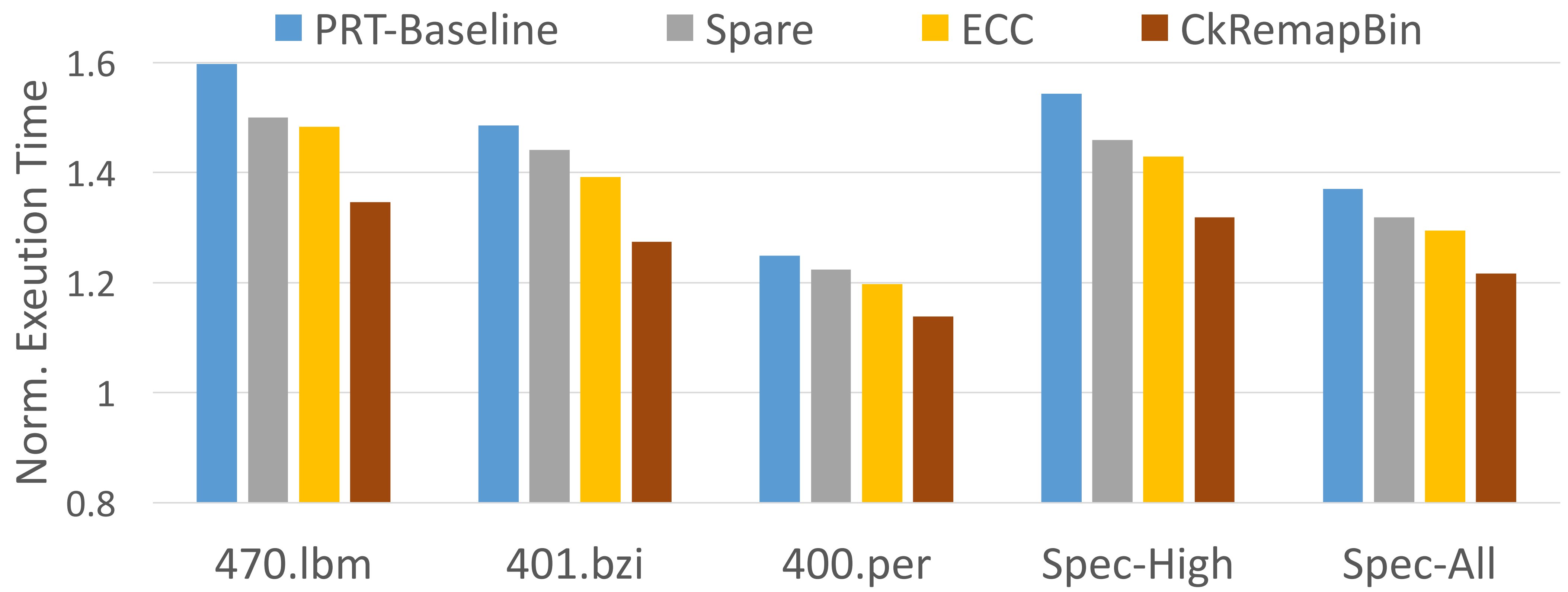
RESTORE-AWARE PAGE ALLOCATION



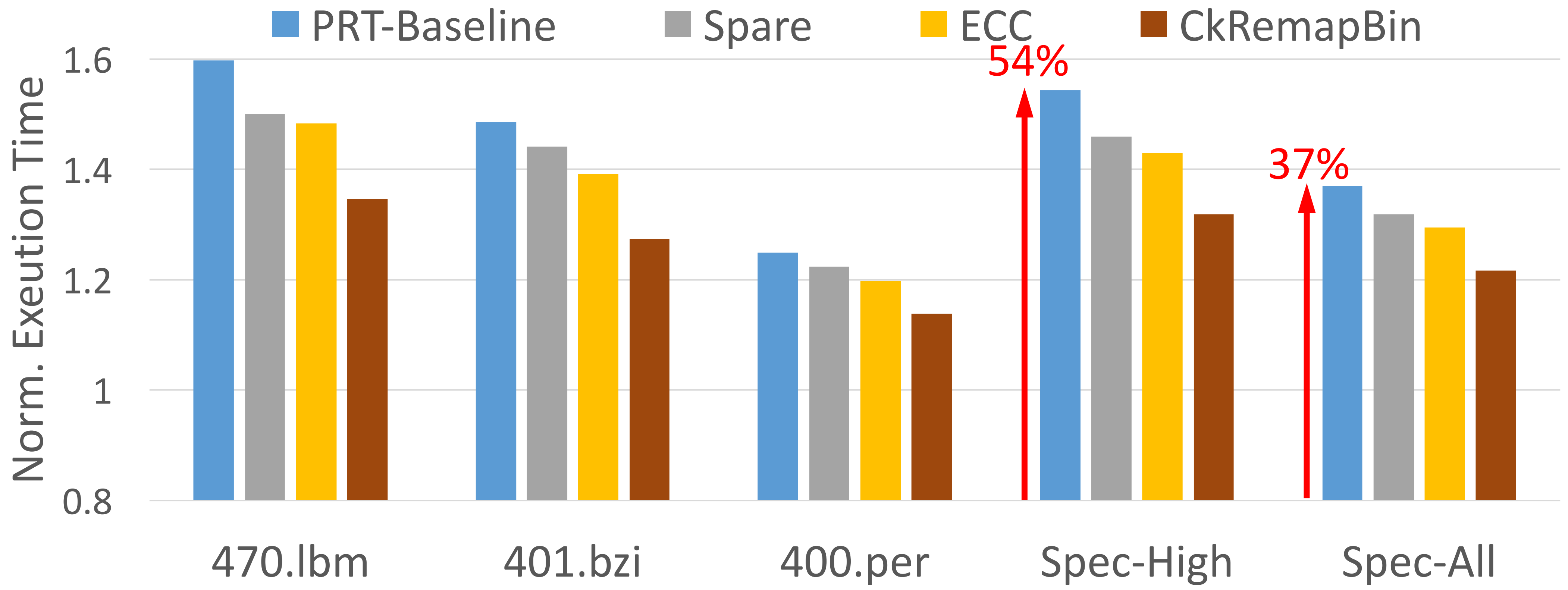


Accesses come from a small set of pages

PERFORMANCE

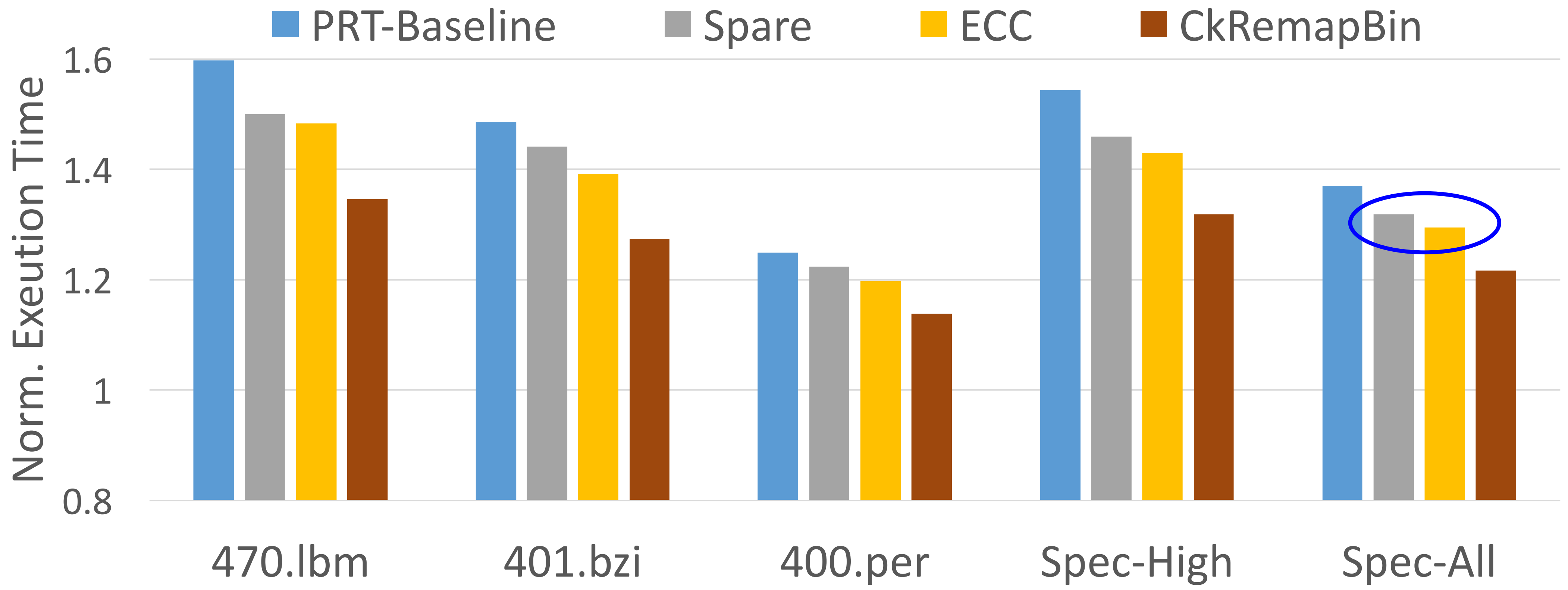


PERFORMANCE



Prolonged restore significantly hurts performance

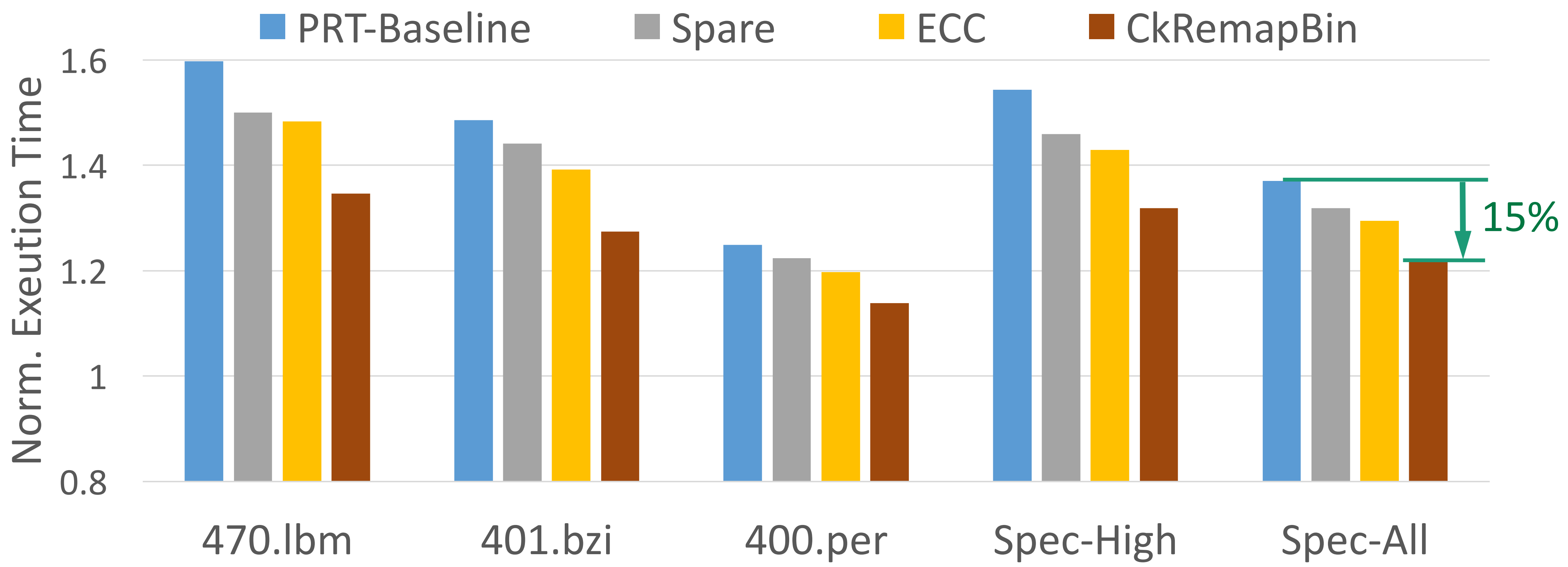
PERFORMANCE



Prolonged restore significantly hurts performance

Classical repair approaches offer limited help

PERFORMANCE

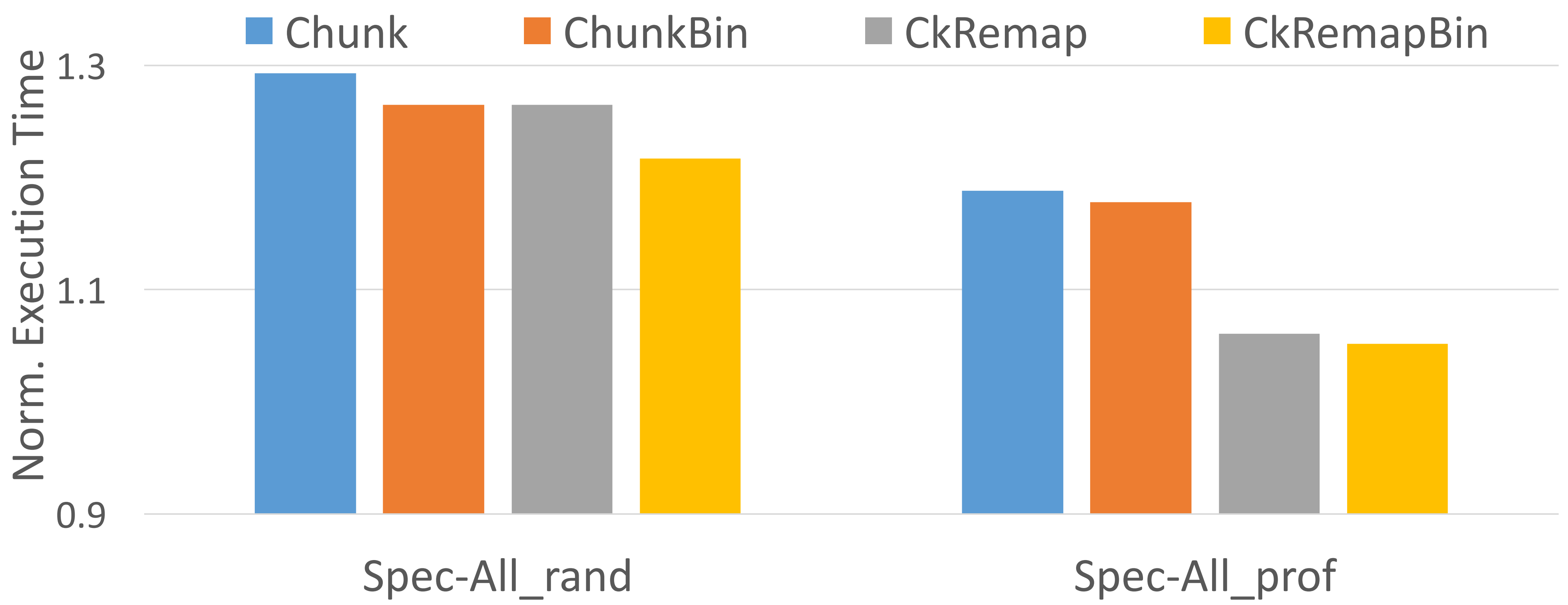


Prolonged restore significantly hurts performance

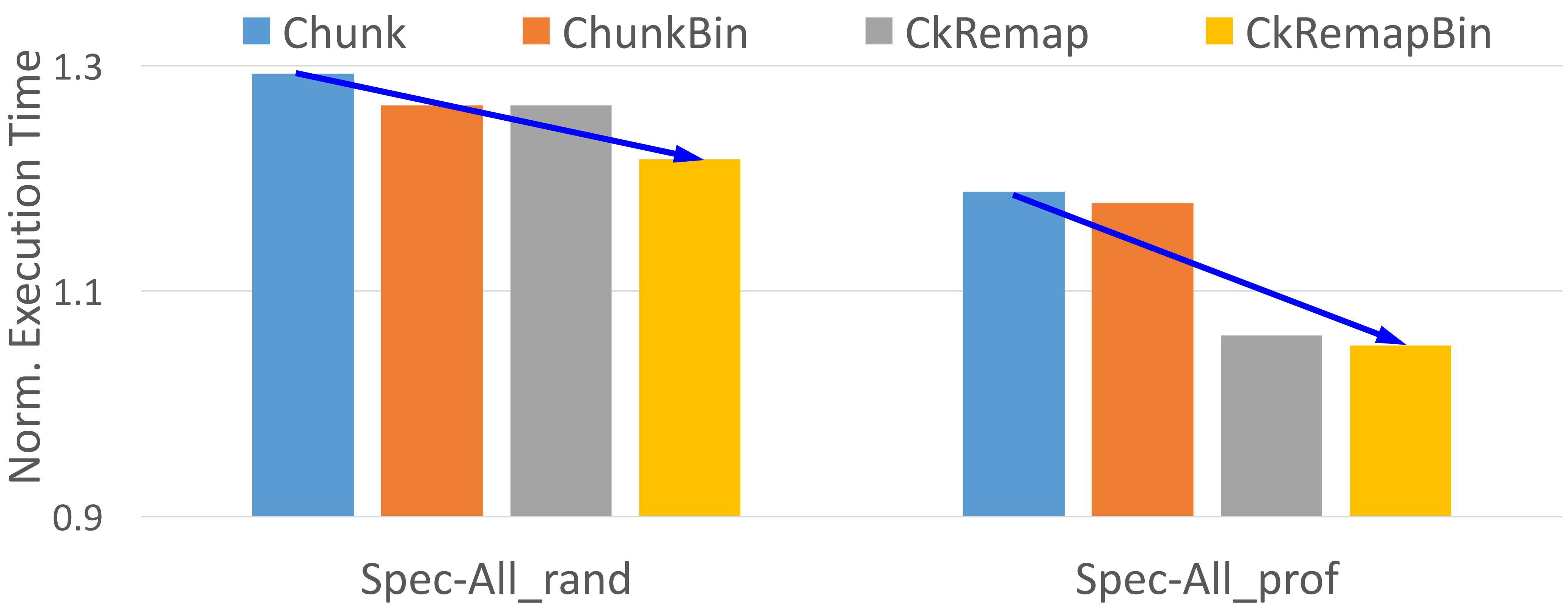
Classical repair approaches offer limited help

With chunk remap and rank construction, avg 15% shorter

PAGE ALLOCATION EFFECTS



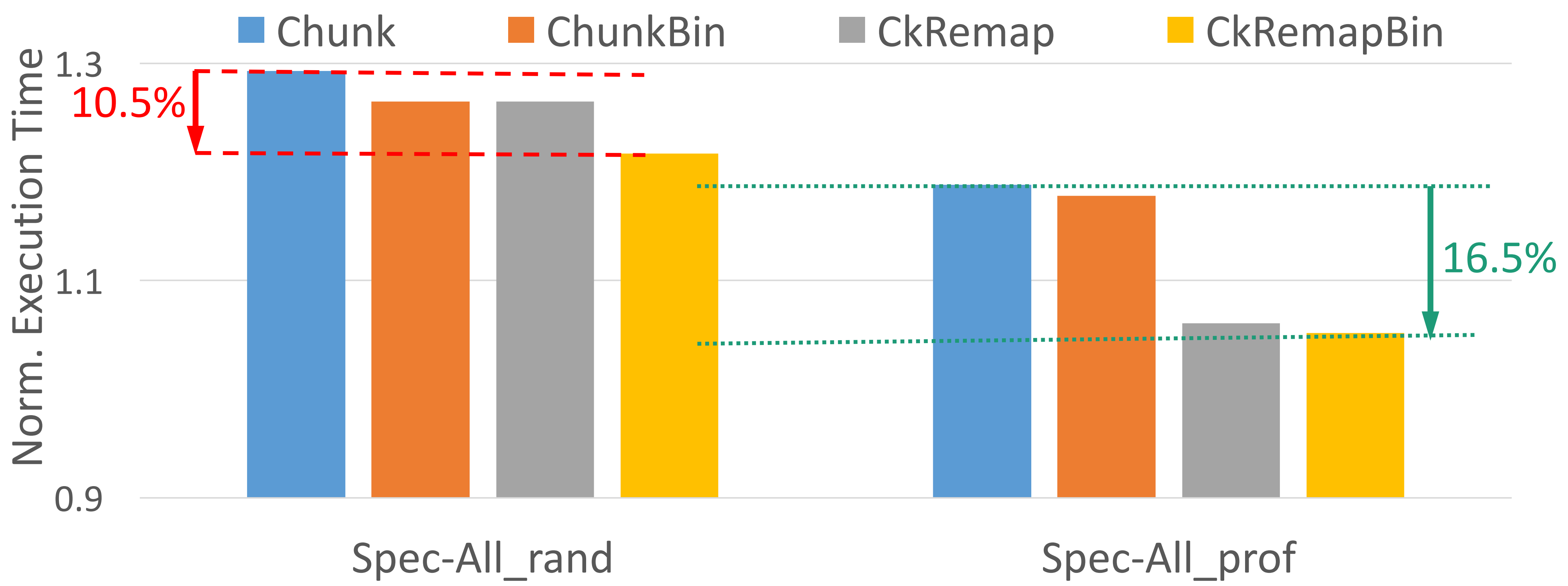
PAGE ALLOCATION EFFECTS



Chunk-remap & rank-construction expose more fast chunks

- provide more opportunities for page-allocation

PAGE ALLOCATION EFFECTS



Chunk-remap & rank-construction expose more fast chunks

- provide more opportunities for page-allocation

Restore-aware page allocation effectively reduce time

SUMMARY: CkRemap

Further scaling restore has serious PV effects
Worse-case based approaches are ineffective

CkRemap: construct fast chunks via remapping
PageAlloc: fully utilize the exposed fast regions

Performance: as high as 25% avg improvement
Page alloc: hotness-aware alloc maximize gains

OUTLINE

RT-Next

Partial restore based on refresh distance

CkRemap

Fast restore via reorganization and allocation

DrMP

Mitigate restore with approximate computing

Summary and Research Directions

OUTLINE

RT-Next

Partial restore based on refresh distance

CkRemap

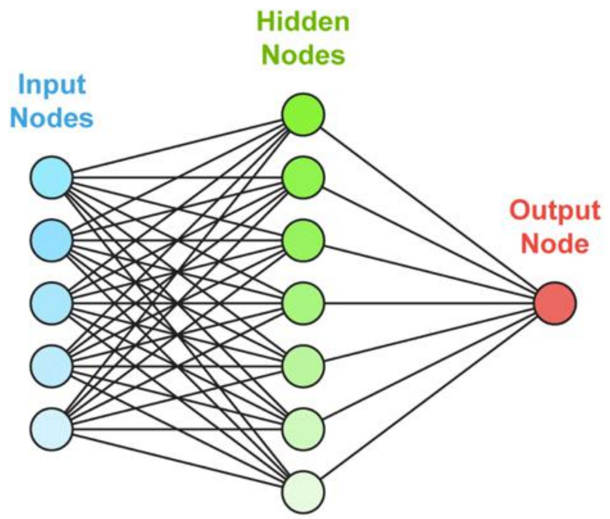
Fast restore via reorganization and allocation

DrMP

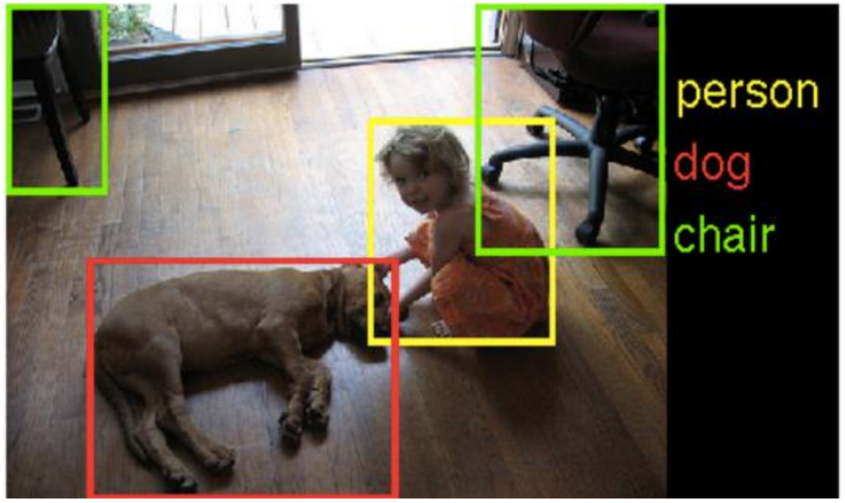
Mitigate restore with approximate computing

Summary and Research Directions

APPLICATION CHARACTERISTICS



Credit: *www-d0.fnal.gov*



Credit: *image-net.org*

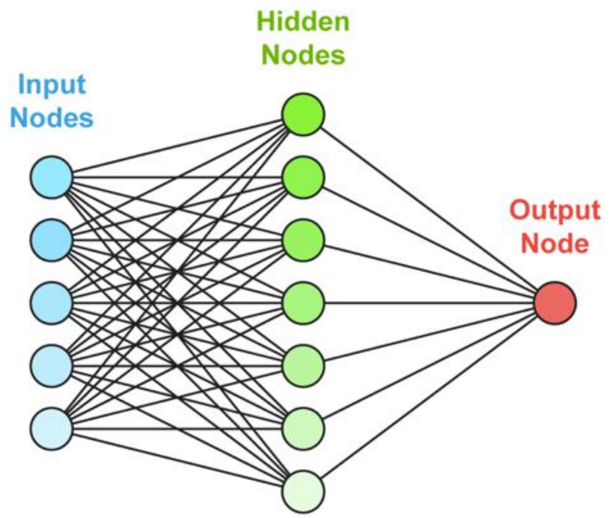
Credit: *www.itbusiness.ca/*

Machine Learning

Computer Vision

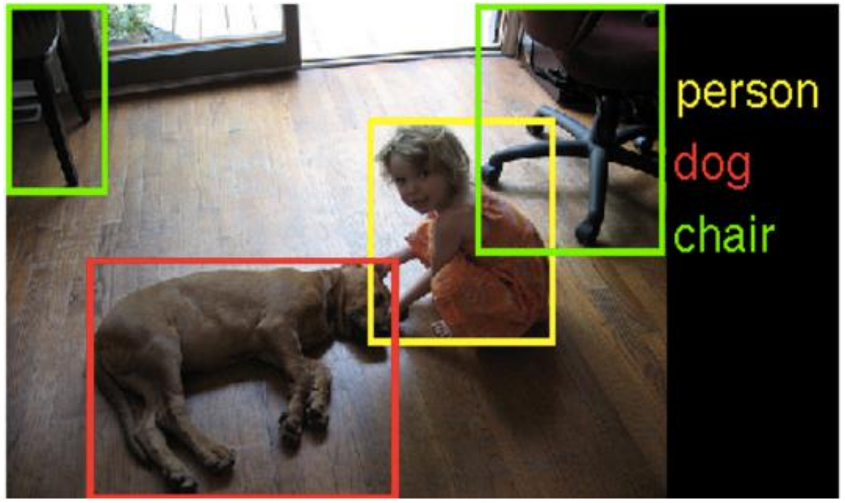
Big Data Analytics

APPLICATION CHARACTERISTICS



Credit: *www-d0.fnal.gov*

Machine Learning



Credit: *image-net.org*

Computer Vision

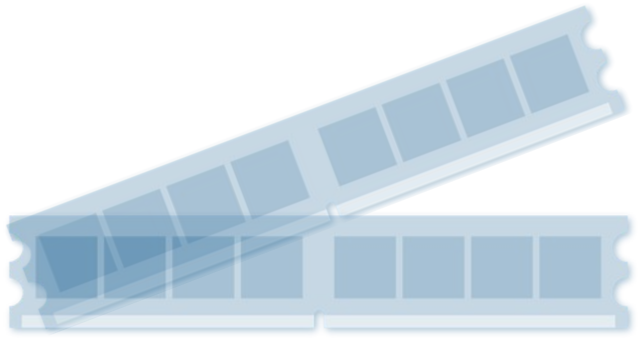
Credit: *www.itbusiness.ca/*

Big Data Analytics

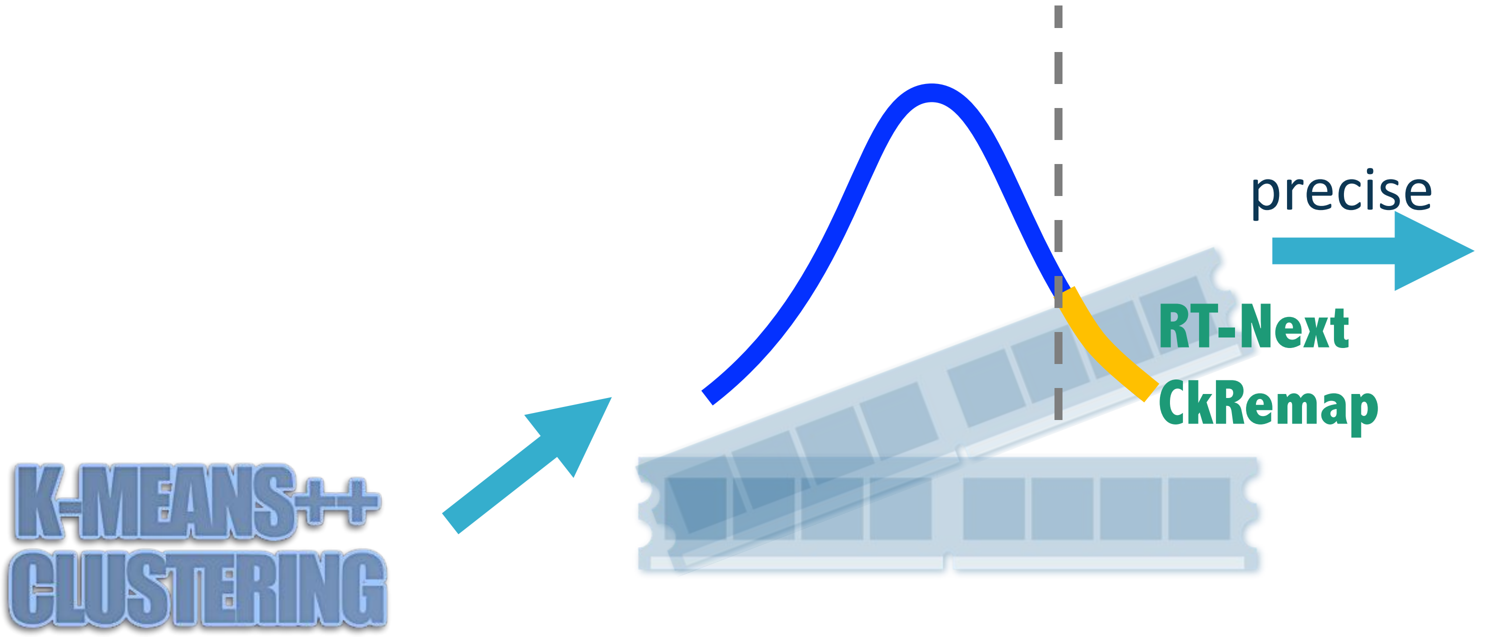
Many applications can tolerate accuracy loss

RESTORE-BASED APPROXIMATION

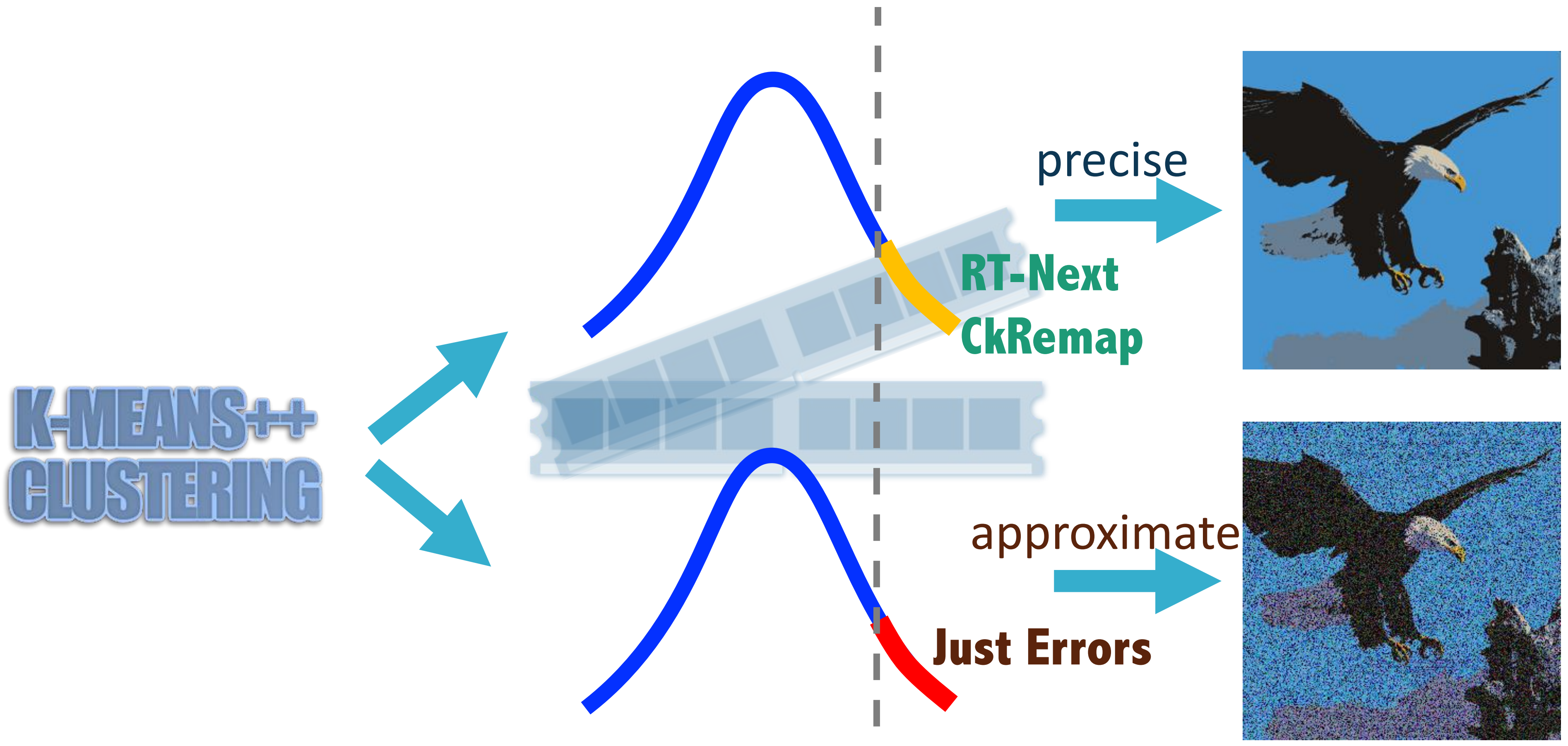
K-MEANS++
CLUSTERING



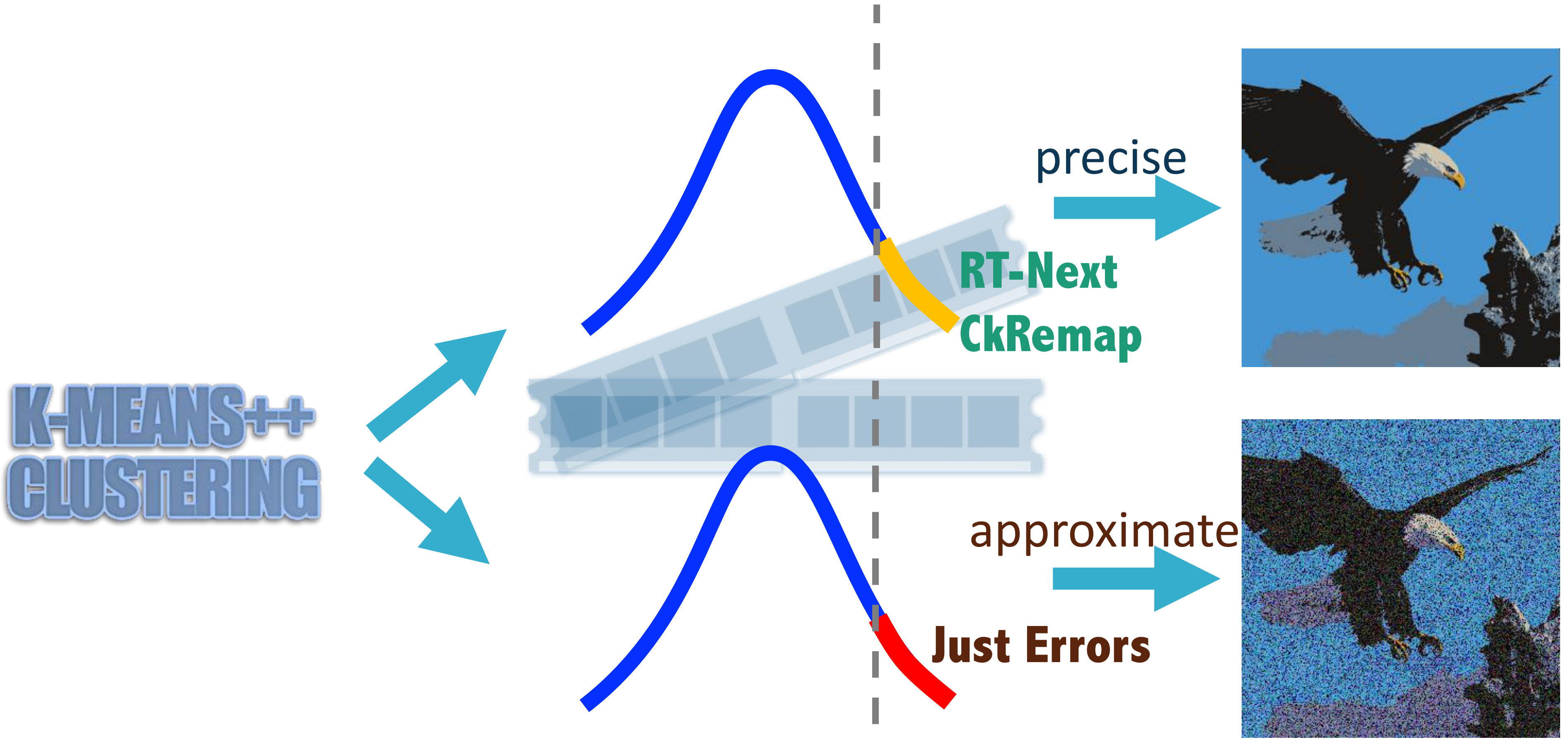
RESTORE-BASED APPROXIMATION



RESTORE-BASED APPROXIMATION

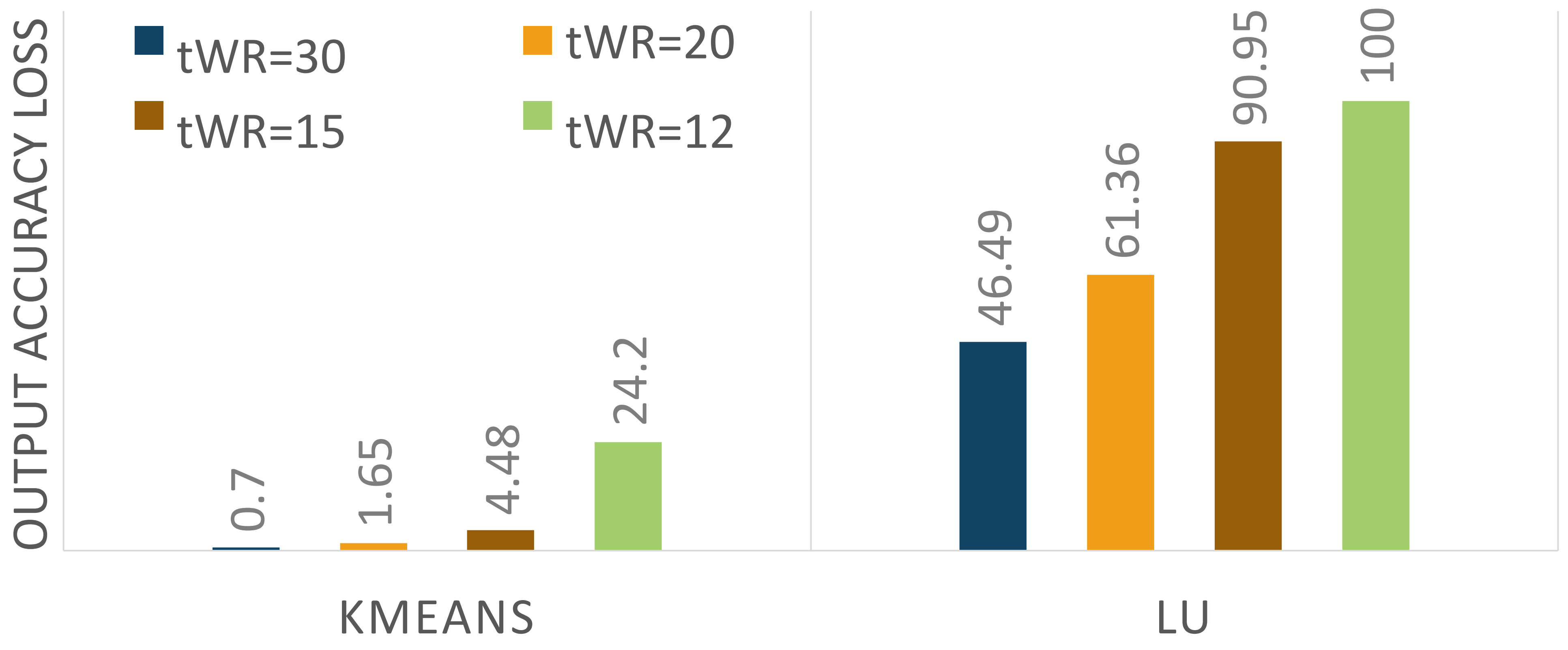


RESTORE-BASED APPROXIMATION

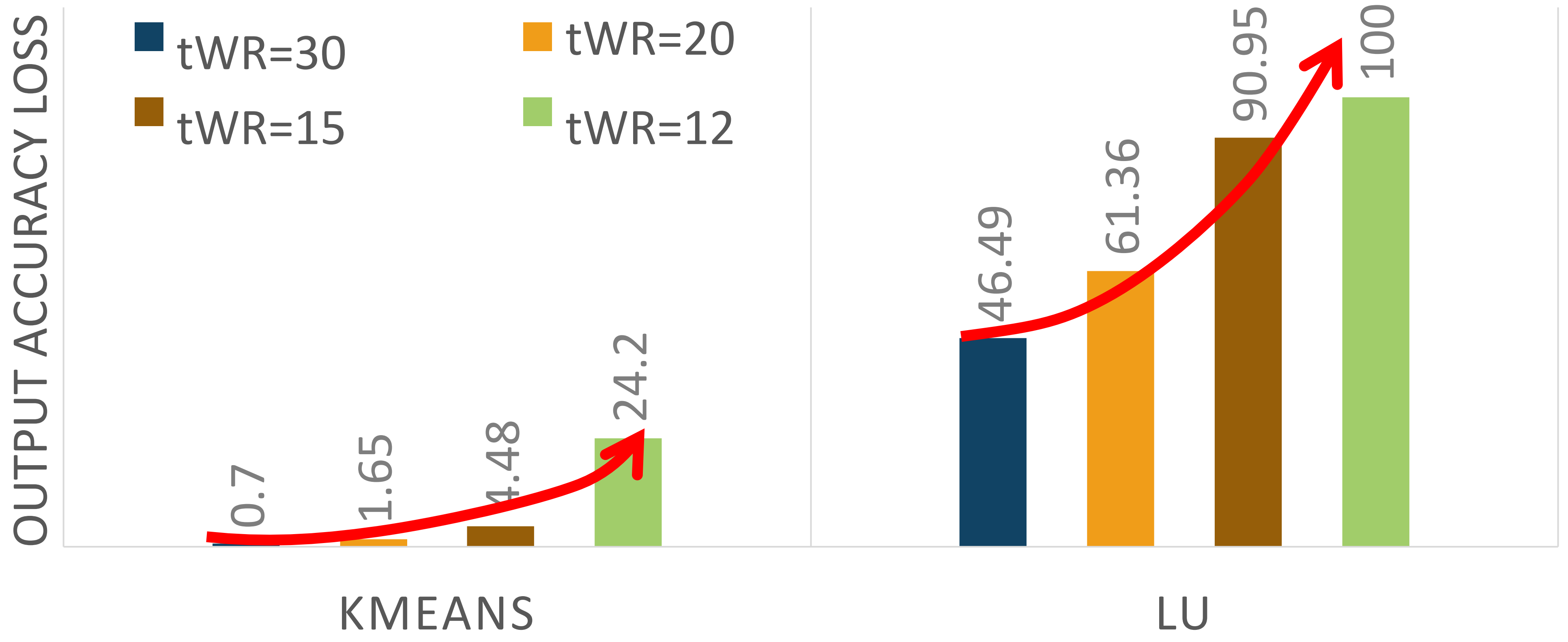


Will the final output always be acceptable?

MOTIVATION RESULTS

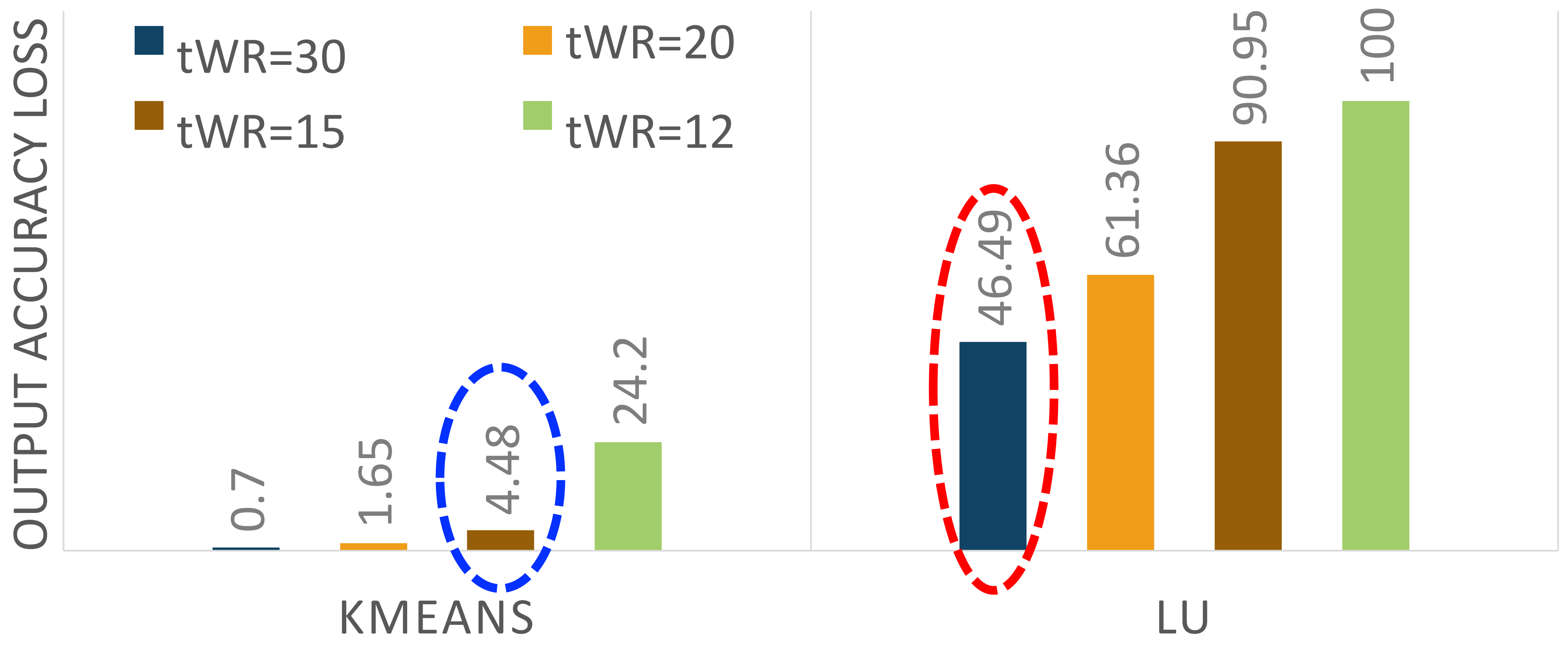


MOTIVATION RESULTS



Accuracy loss steadily enlarges along tWR decrease

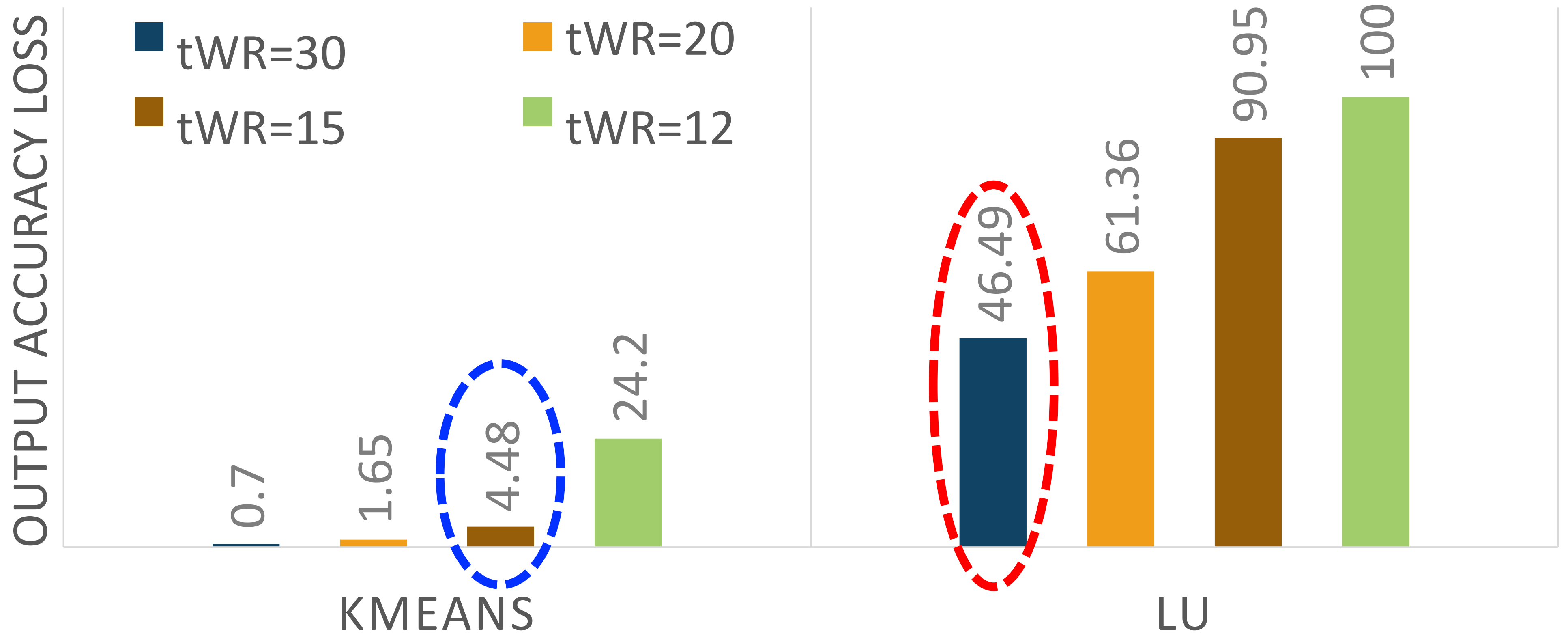
MOTIVATION RESULTS



Accuracy loss steadily enlarges along tWR decrease

Applications show vastly different behaviors

MOTIVATION RESULTS



Accuracy loss steadily enlarges along tWR decrease

Final output quality must be controlled

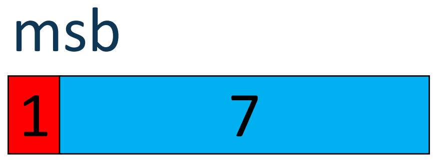
CRITICAL DATA

error-sensitive	error-resilient
pointers	pixels
jump targets	neuron weights
meta data	video frames

Critical data cannot be approximated

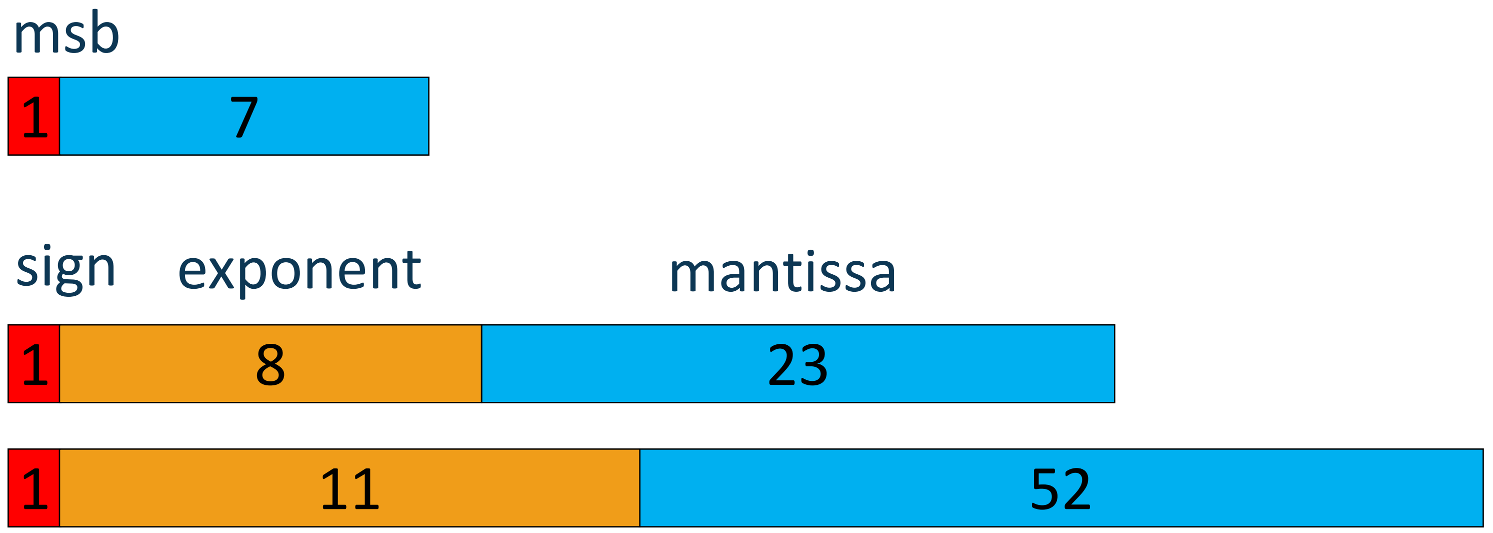
BITS ARE NOT EQUALLY IMPORTANT

BITS ARE NOT EQUALLY IMPORTANT

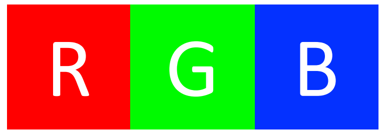


Int/byte
R
G
B

BITS ARE NOT EQUALLY IMPORTANT



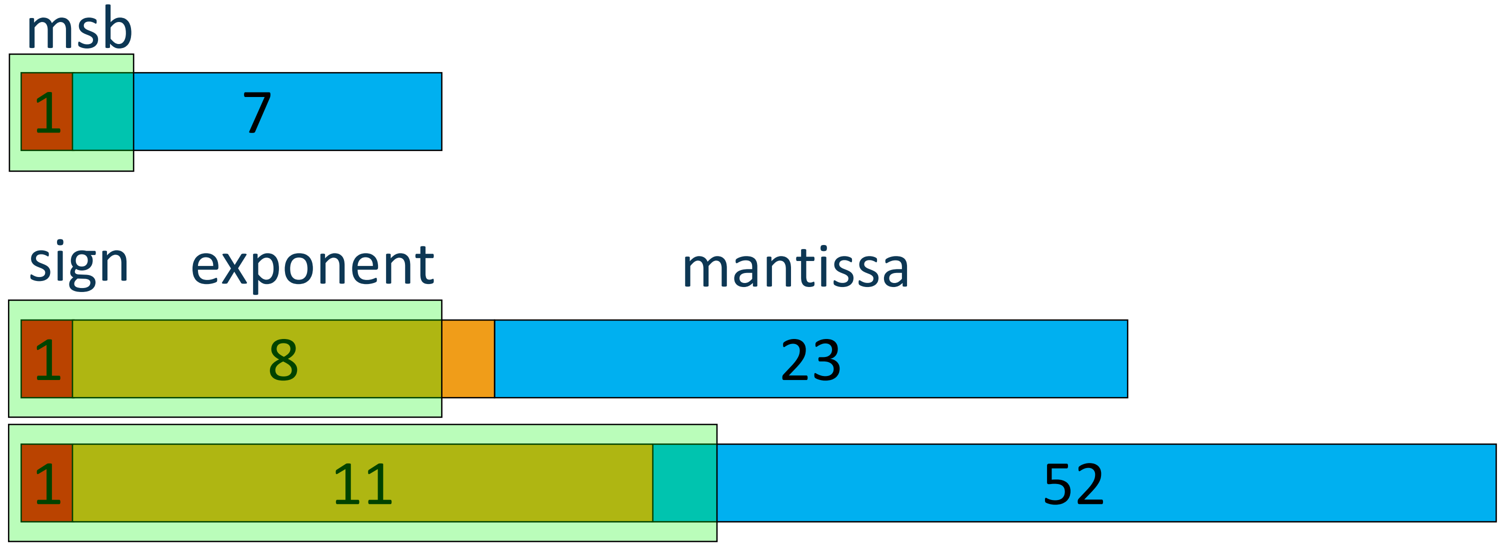
Int/byte



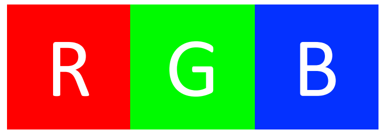
Float

Double

BITS ARE NOT EQUALLY IMPORTANT



Int/byte

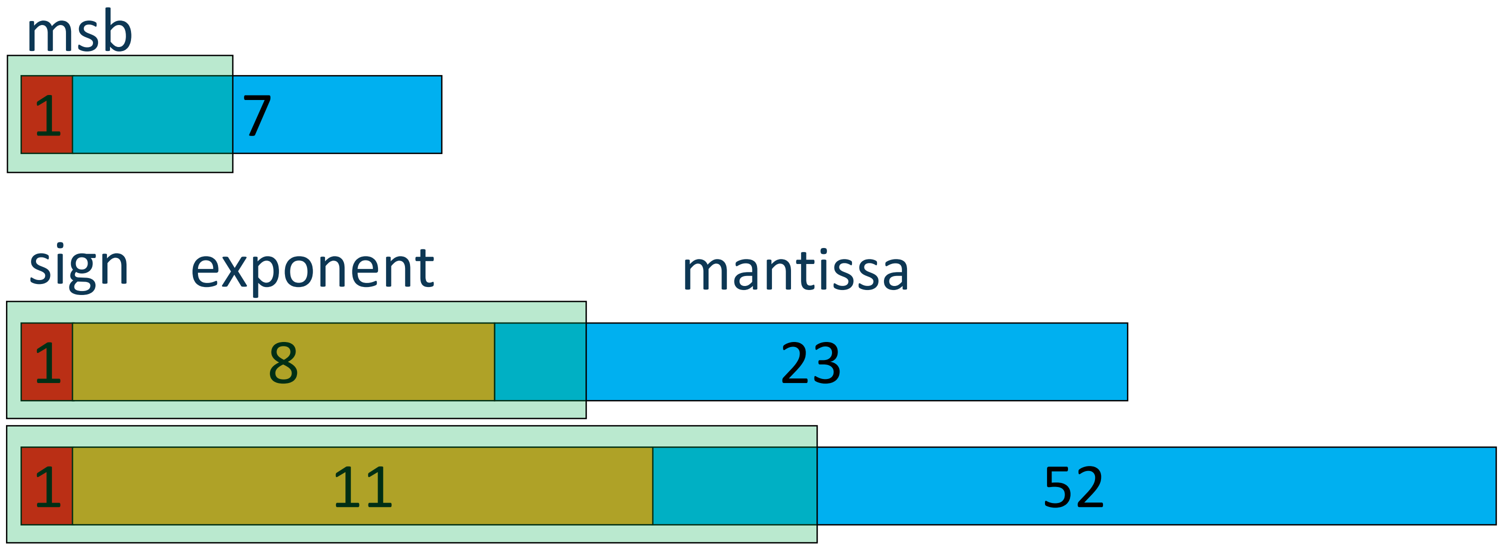


Float

Double

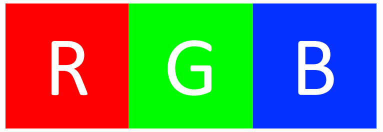
25%

BITS ARE NOT EQUALLY IMPORTANT



50%

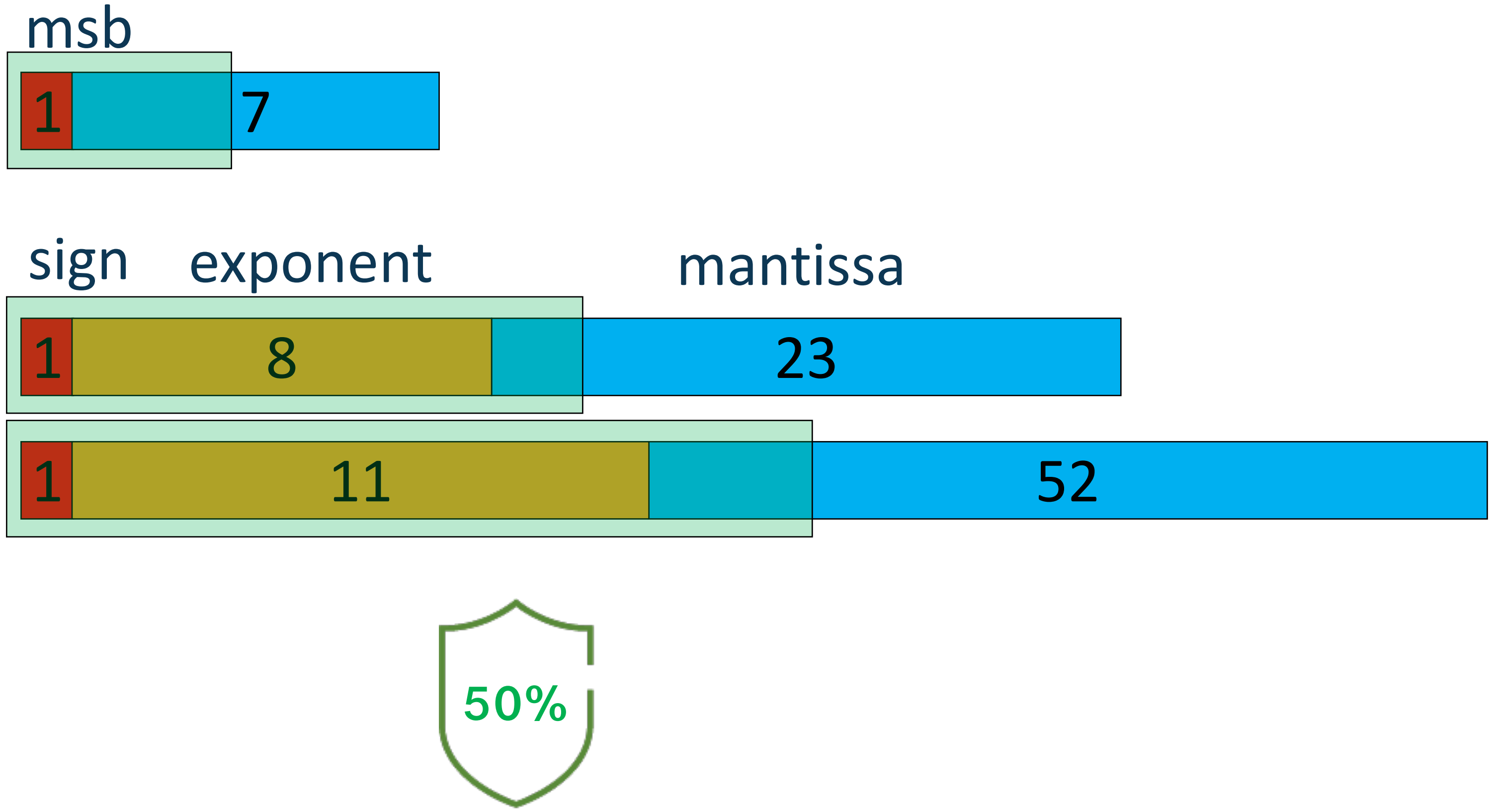
Int/byte



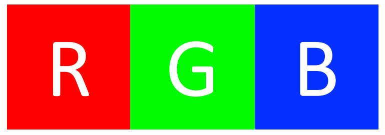
Float

Double

BITS ARE NOT EQUALLY IMPORTANT



Int/byte

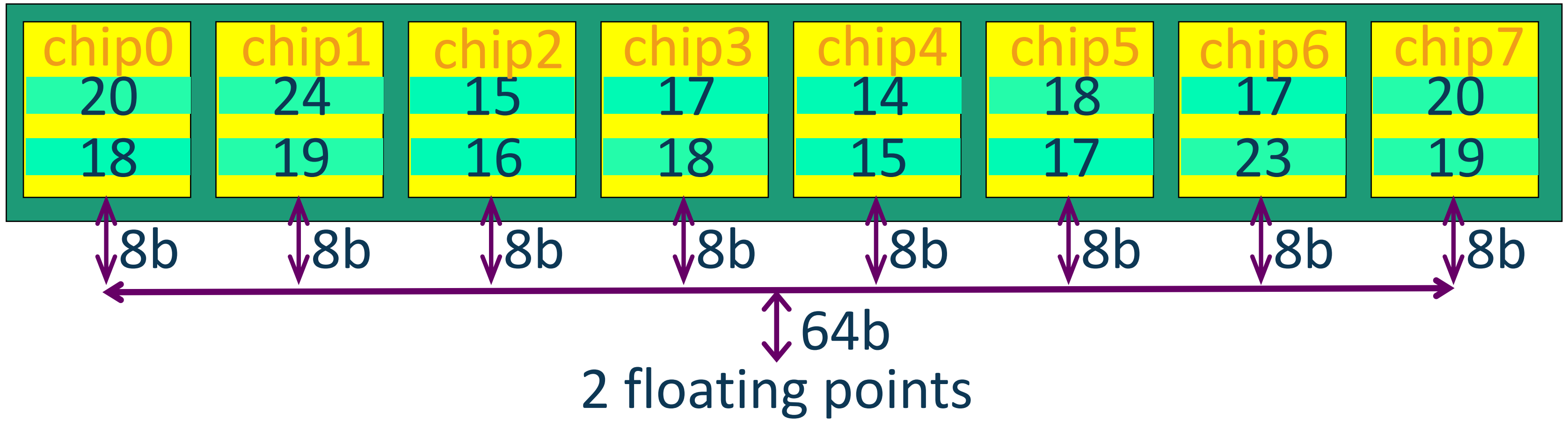


Float

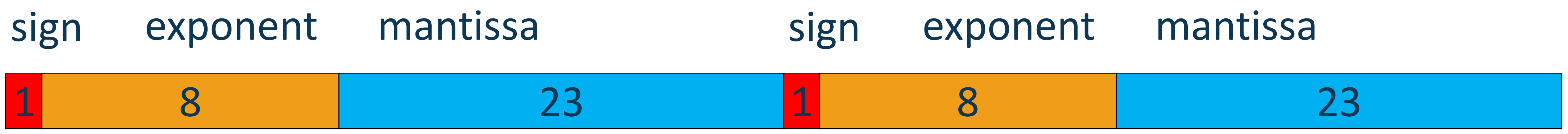
Double

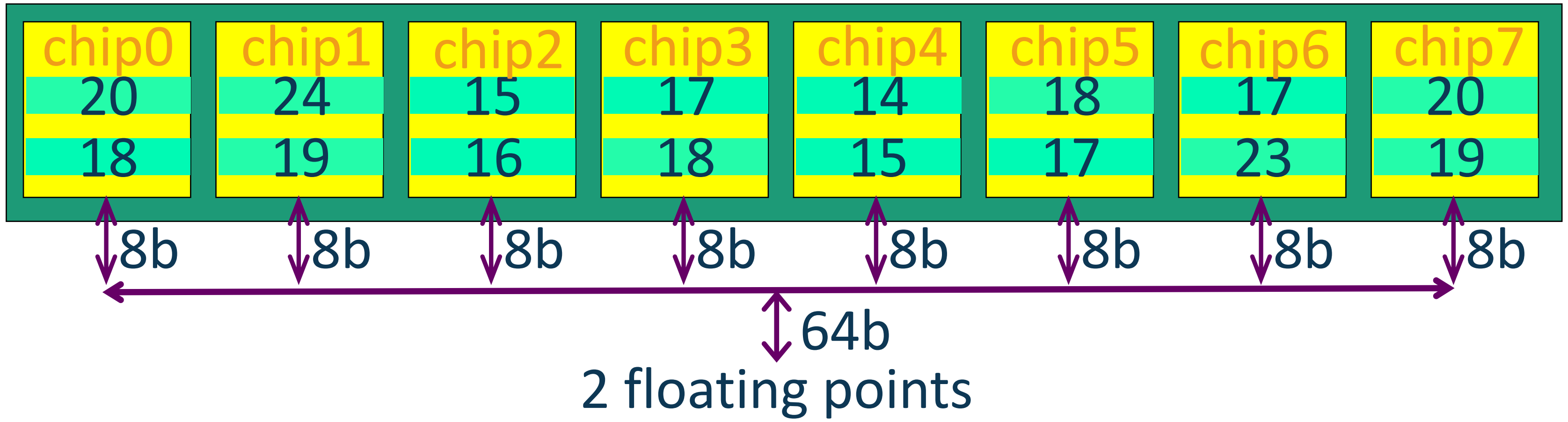
There is a tradeoff between accuracy and overhead

DrMP: APPROXIMATE DRAM ROW

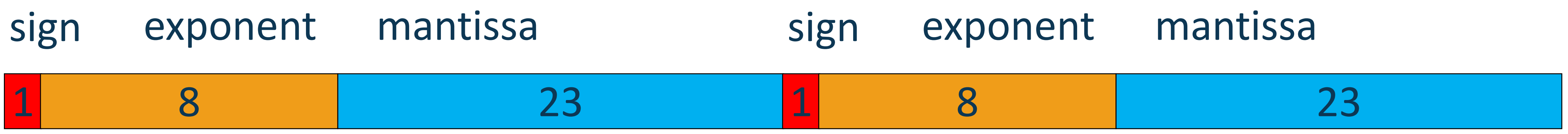


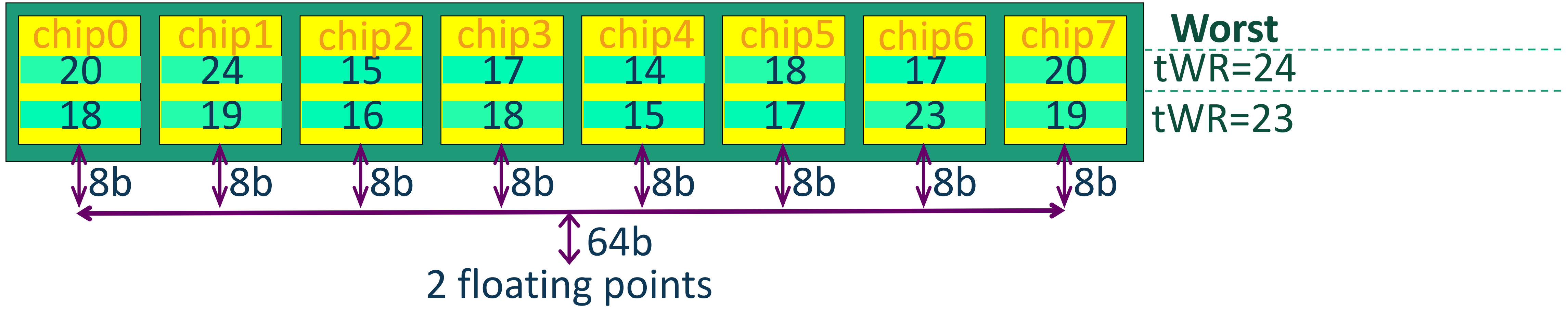
DrMP: APPROXIMATE DRAM ROW



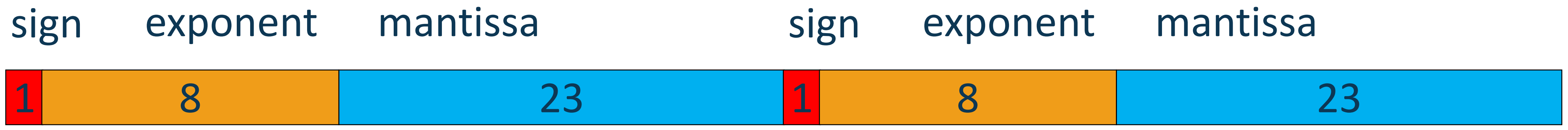


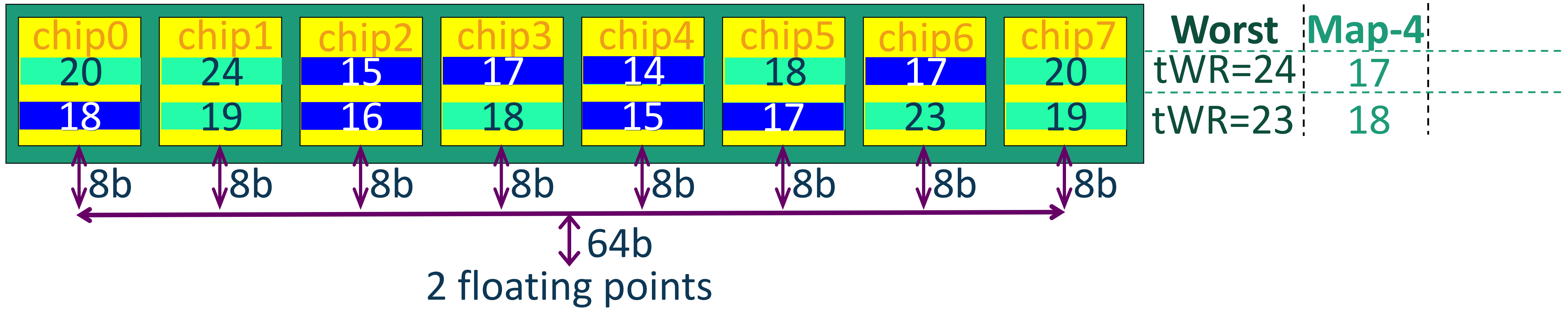
DrMP: APPROXIMATE DRAM ROW



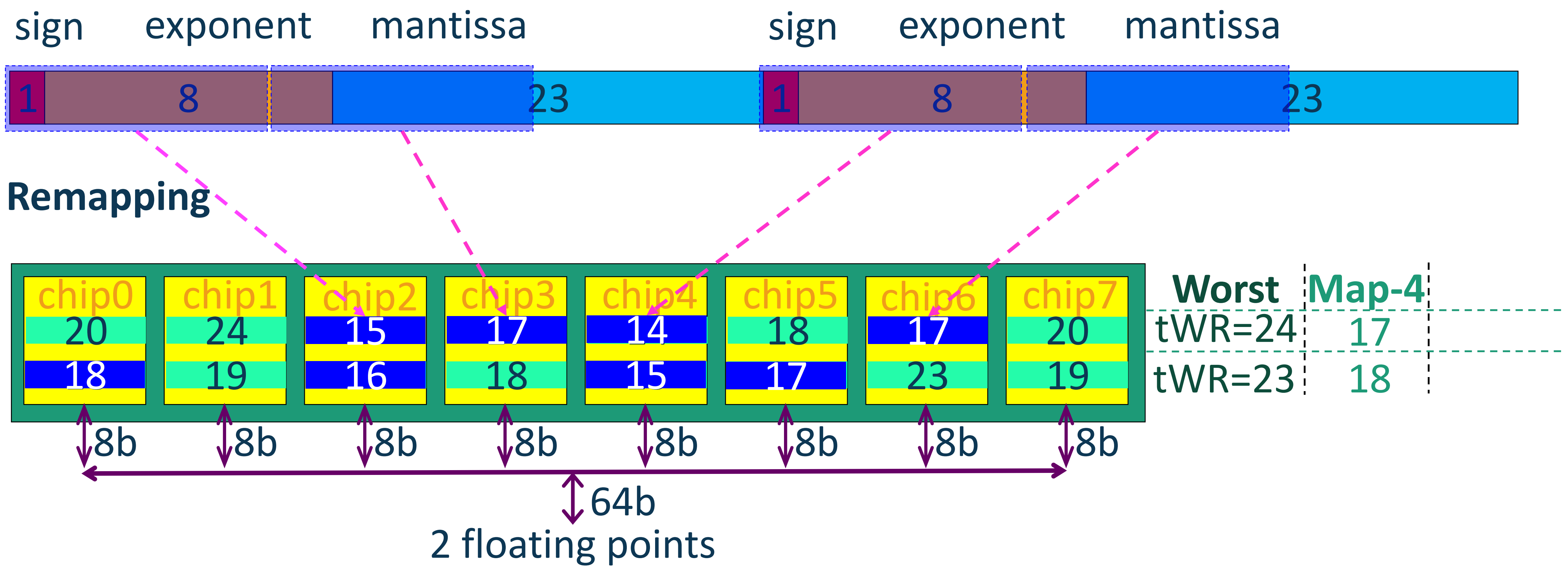


DrMP: APPROXIMATE DRAM ROW

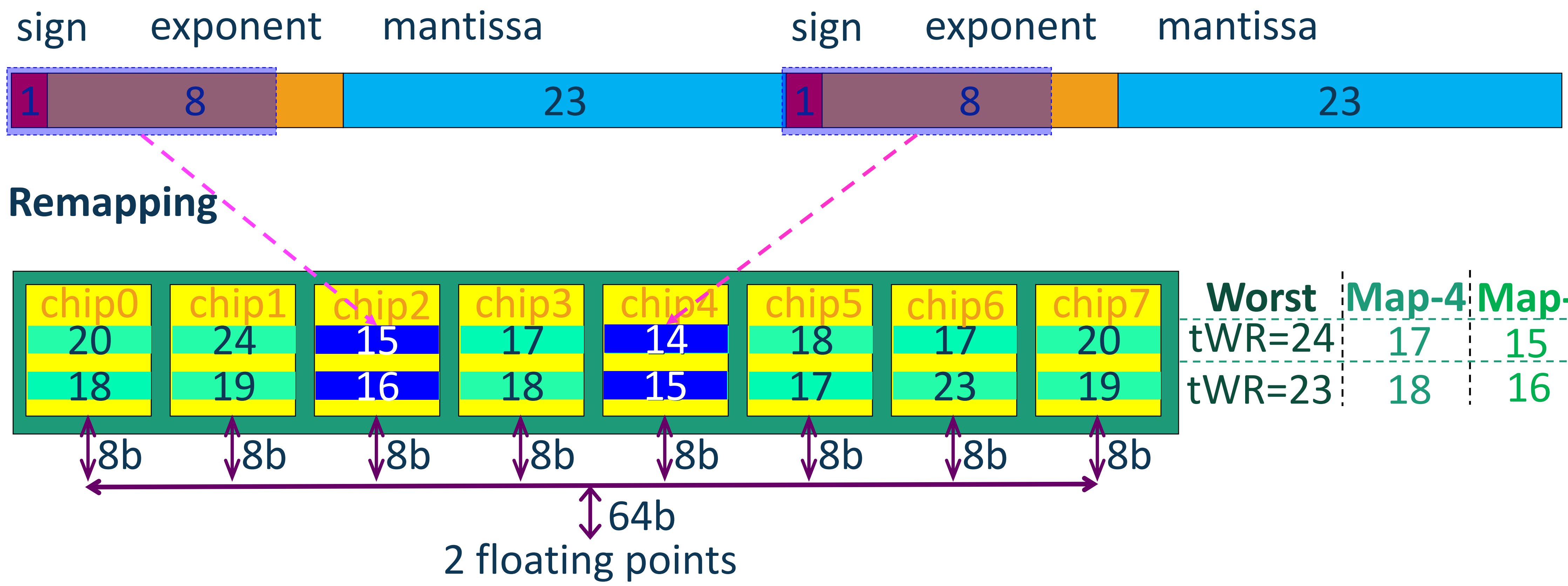




DrMP: APPROXIMATE DRAM ROW

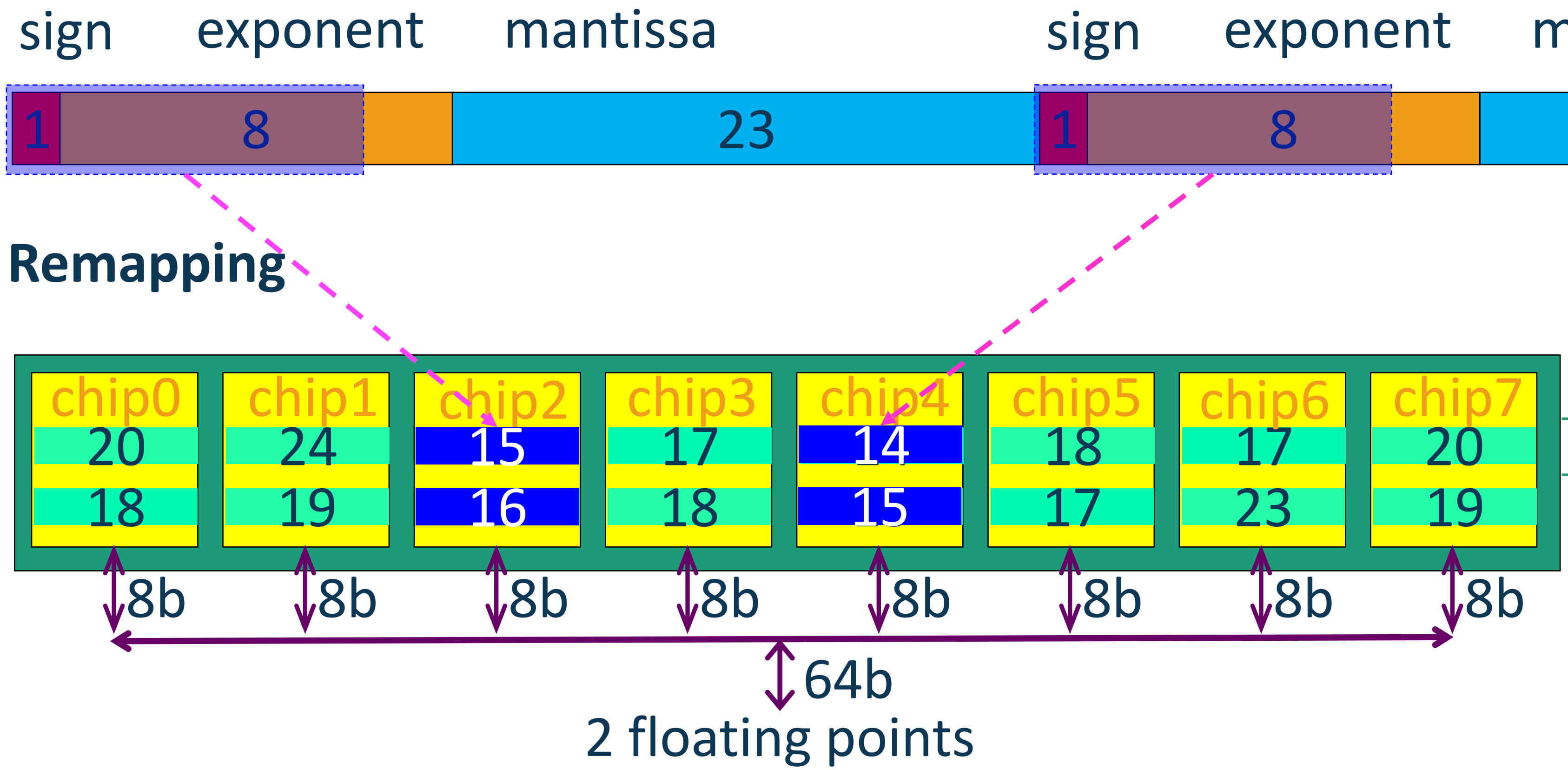


DrMP: APPROXIMATE DRAM ROW



Worst	Map-4	Map-2
tWR=24	17	15
tWR=23	18	16

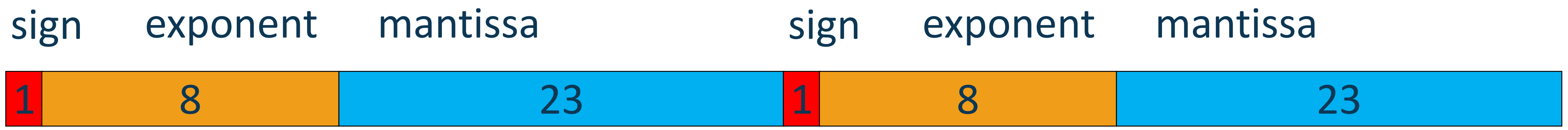
DrMP: APPROXIMATE DRAM ROW



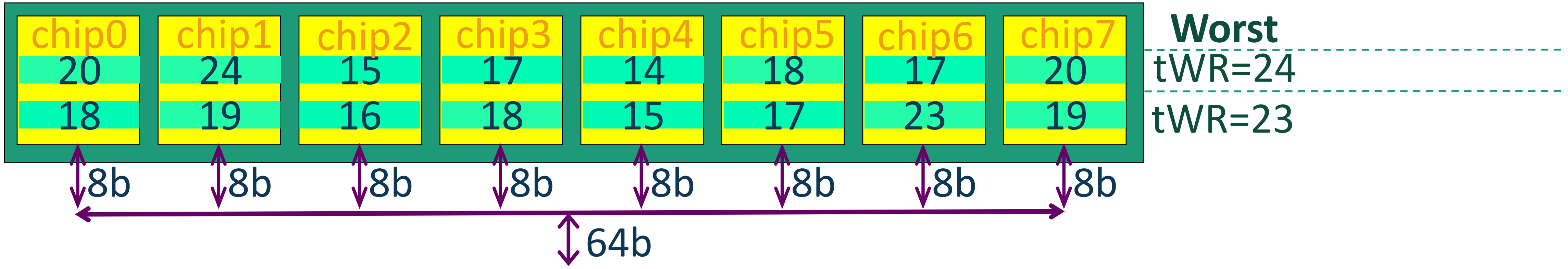
Worst	Map-4	Map-2
tWR=24	17	15
tWR=23	18	16

What if there aren't that much approx data?

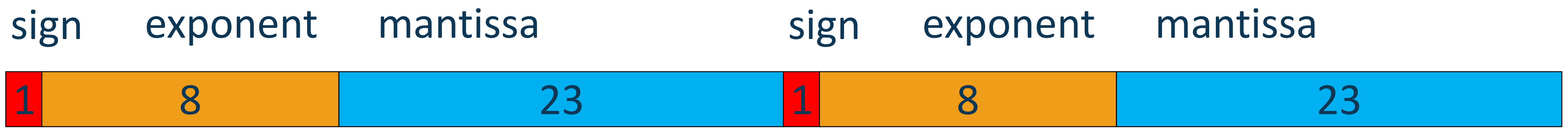
DrMP’: PRECISE + APPROX



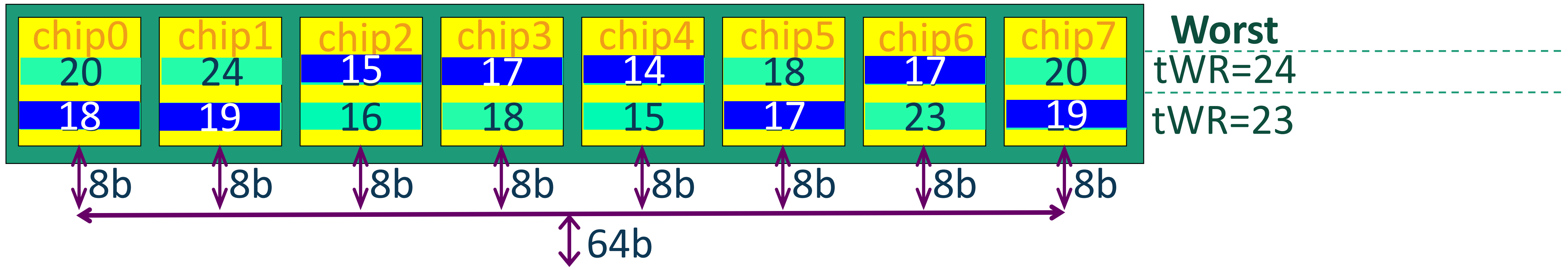
Precise + Approx



DrMP’: PRECISE + APPROX



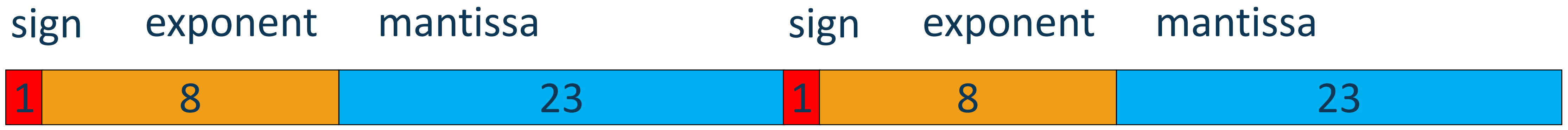
Precise + Approx



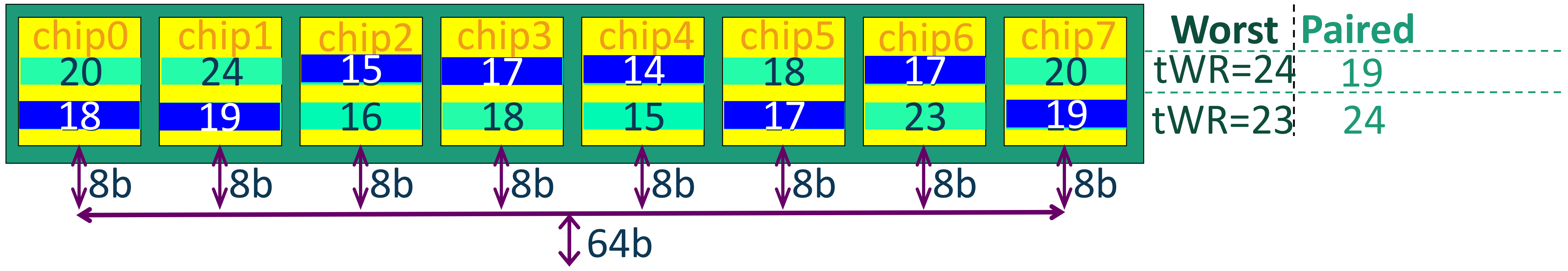
Pair two rows to re-combine chip segments

- Choose smaller one from each location to form a fast one (Precise)

DrMP': PRECISE + APPROX



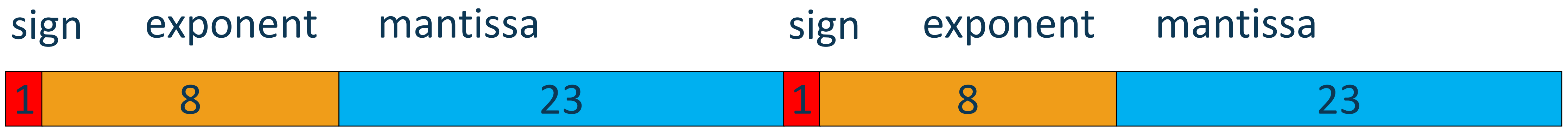
Precise + Approx



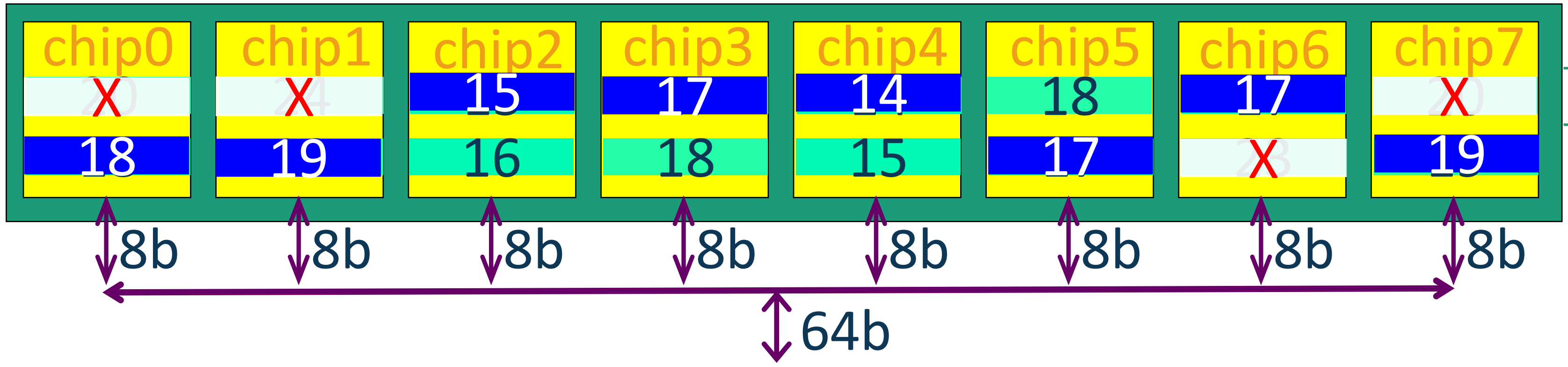
Pair two rows to re-combine chip segments

- Choose smaller one from each location to form a fast one (Precise)

DrMP': PRECISE + APPROX



Precise + Approx



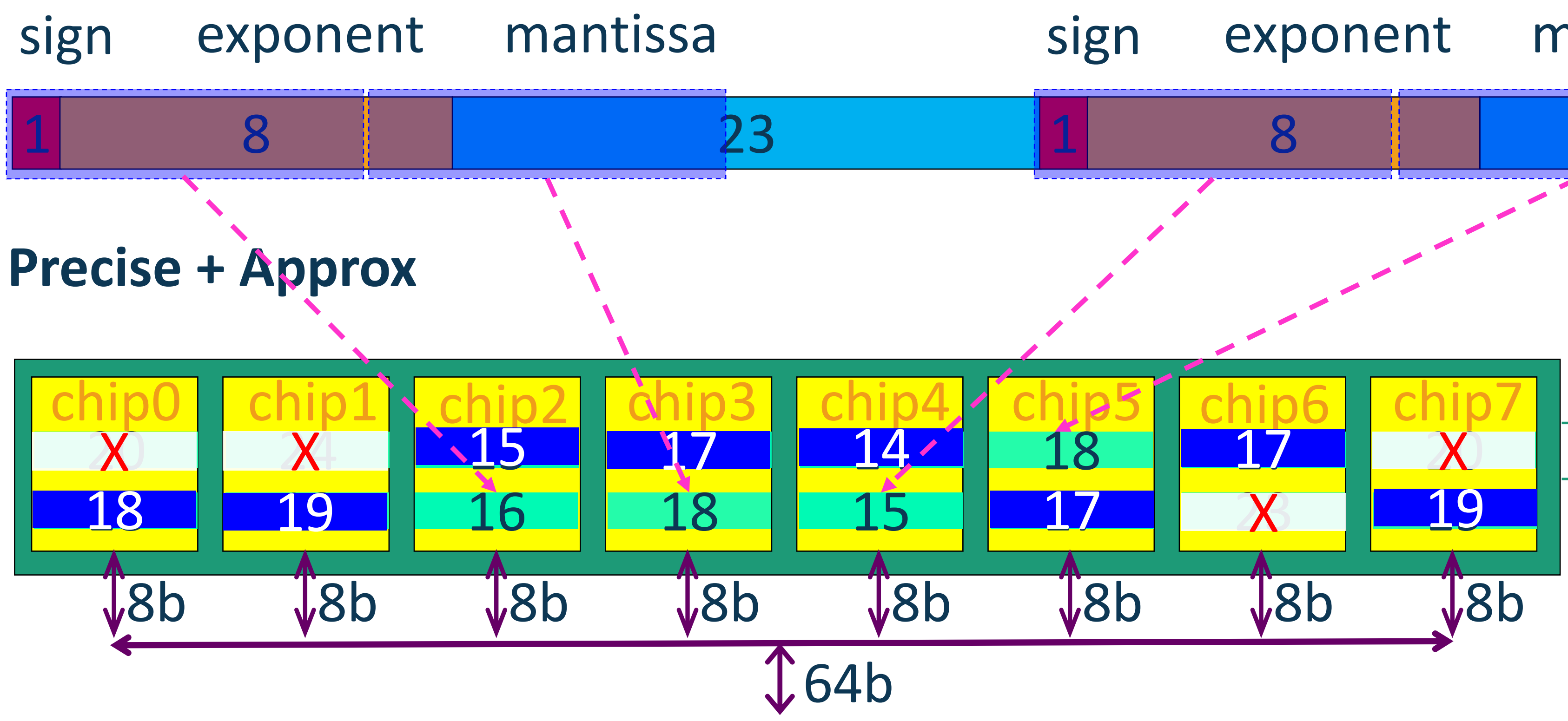
Worst	Paired	Approx
tWR=24	19	19
tWR=23	24	18

Pair two rows to re-combine chip segments

- Choose smaller one from each location to form a fast one (Precise)

Guarantee partial precise for the other slow row

DrMP': PRECISE + APPROX



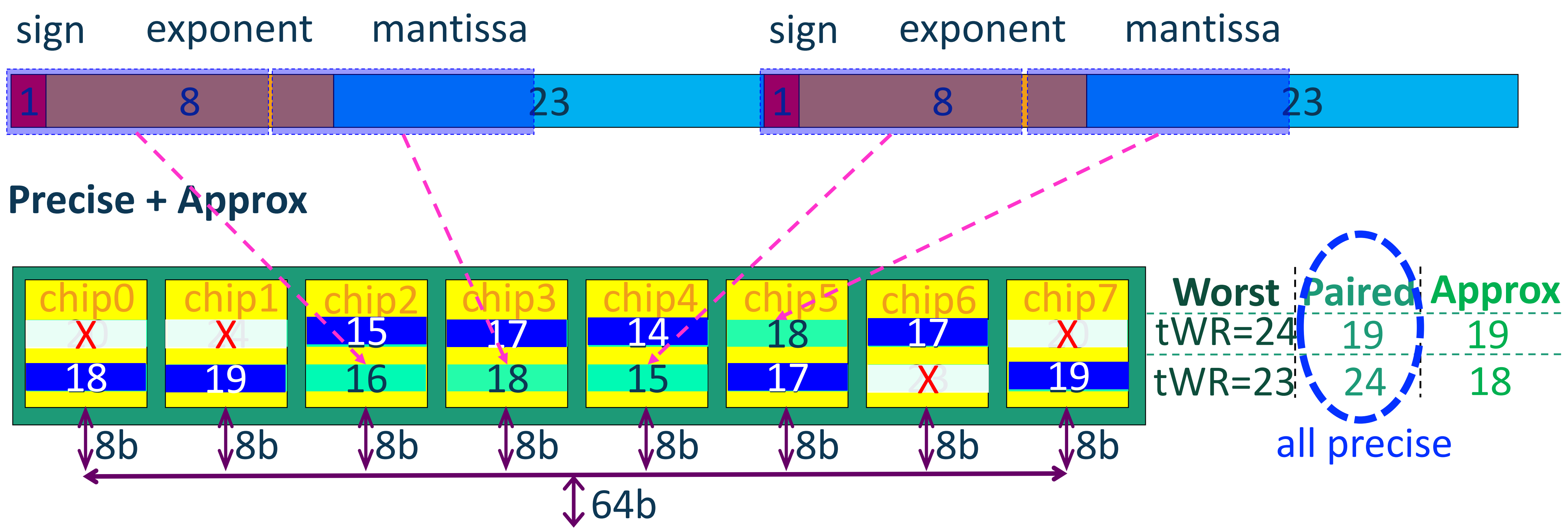
Worst	Paired	Approx
tWR=24	19	19
tWR=23	24	18

Pair two rows to re-combine chip segments

- Choose smaller one from each location to form a fast one (Precise)

Guarantee partial precise for the other slow row

DrMP': PRECISE + APPROX

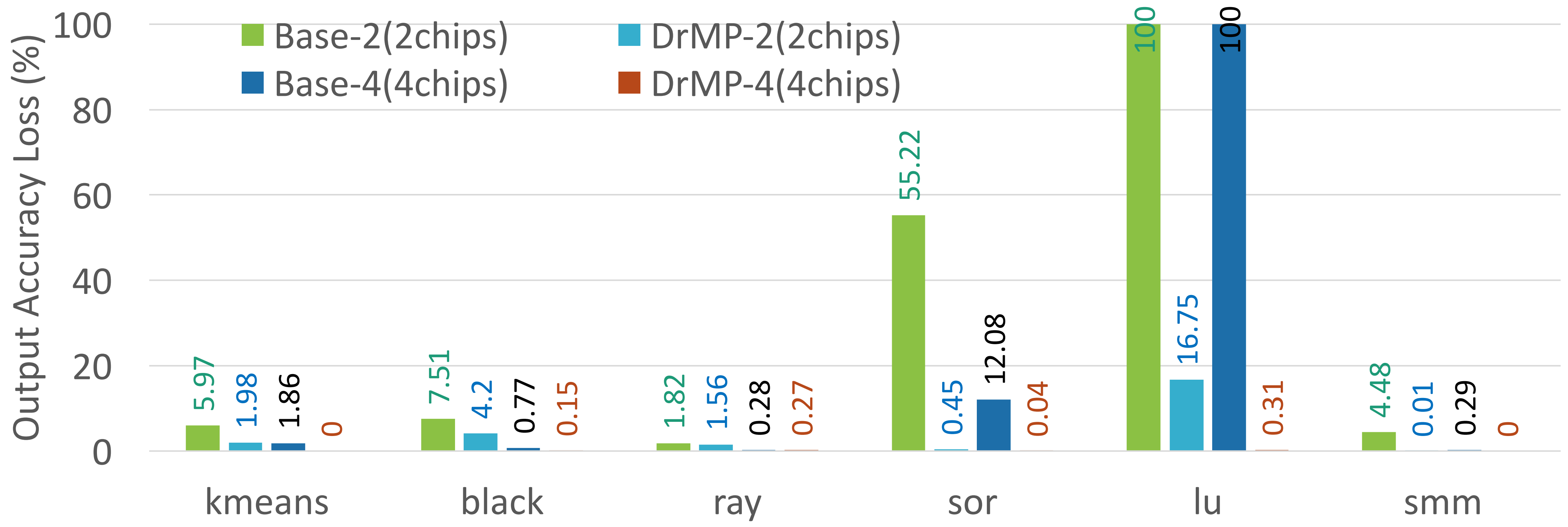


Pair two rows to re-combine chip segments

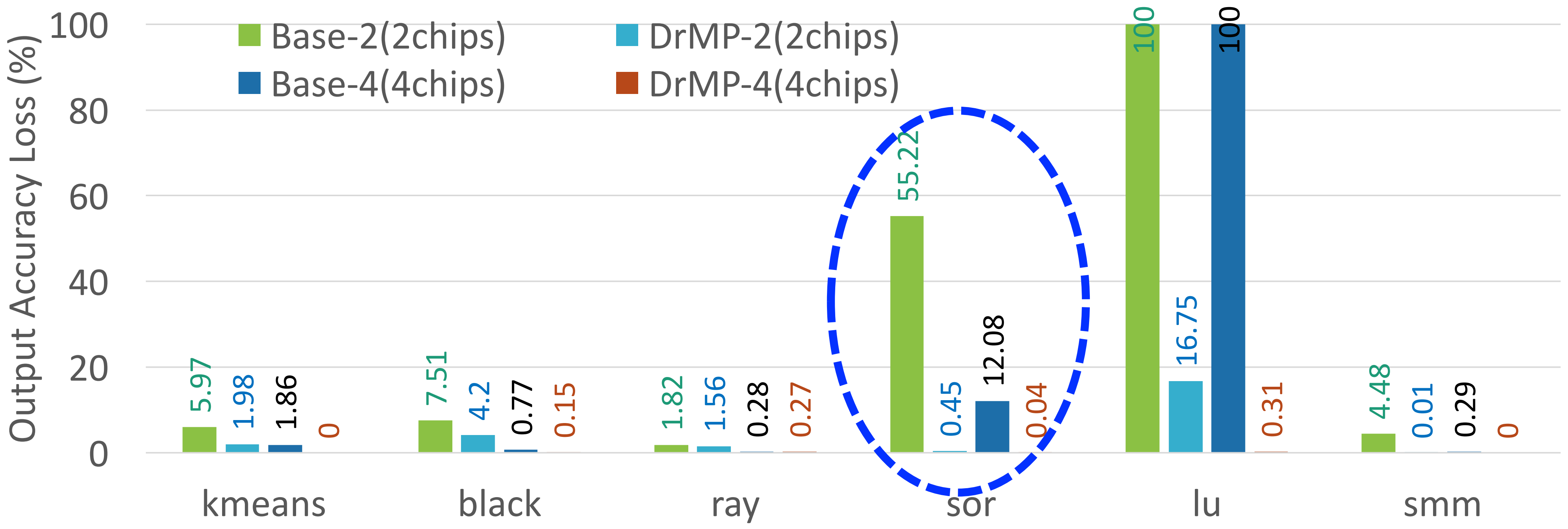
- Choose smaller one from each location to form a fast one (Precise)

Guarantee partial precise for the other slow row

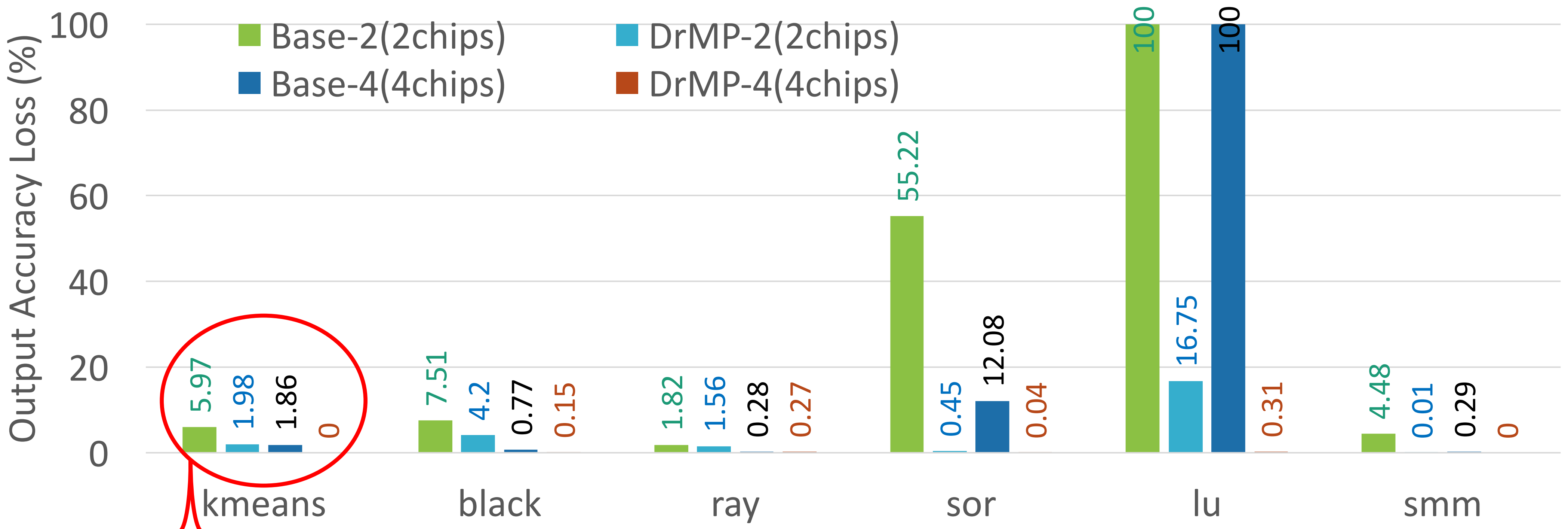
OUTPUT QUALITY

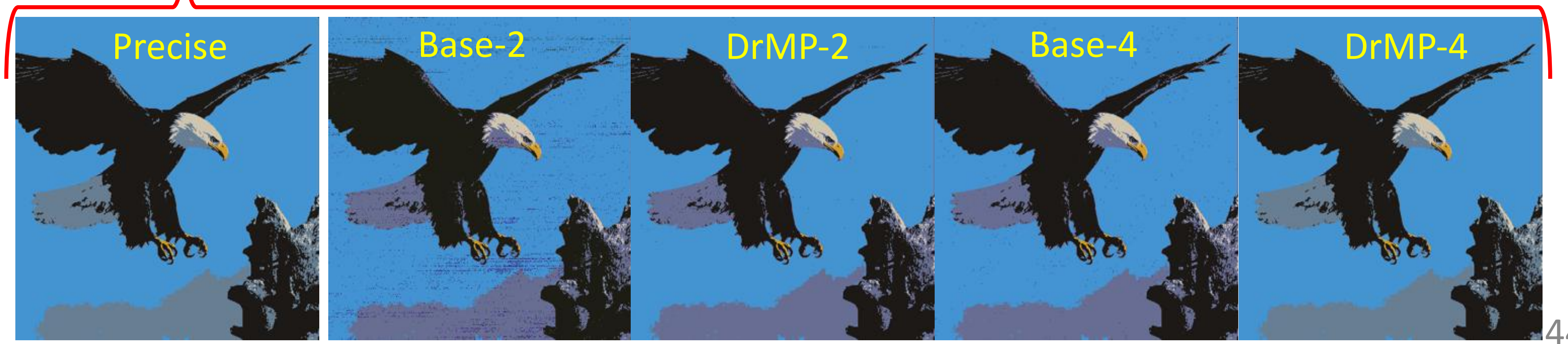


OUTPUT QUALITY

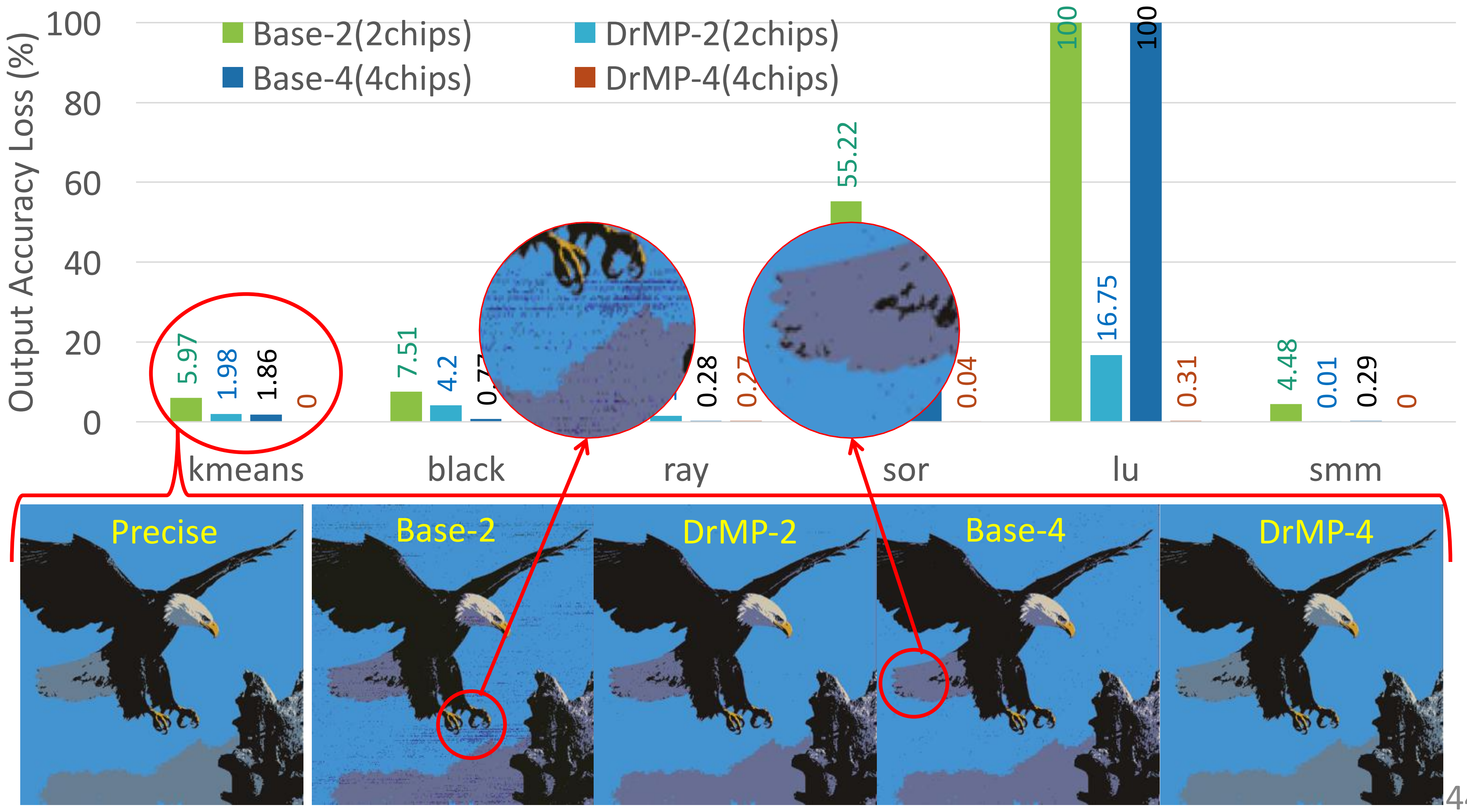


OUTPUT QUALITY

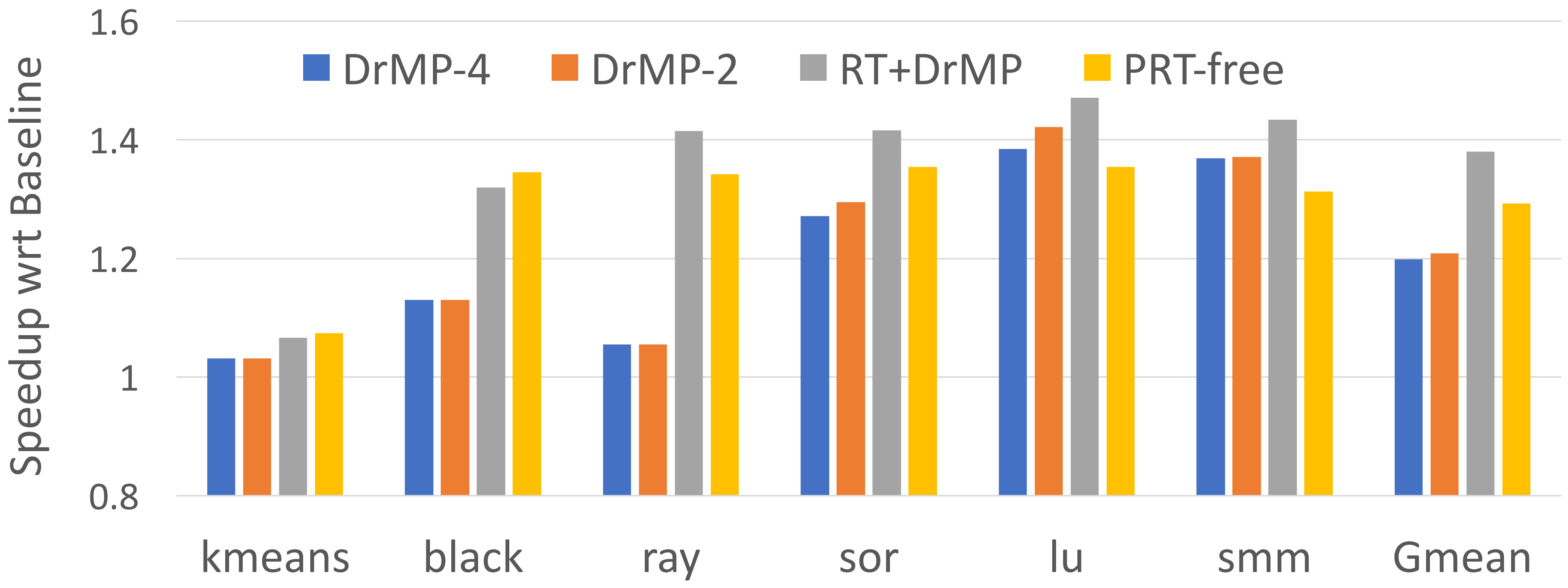




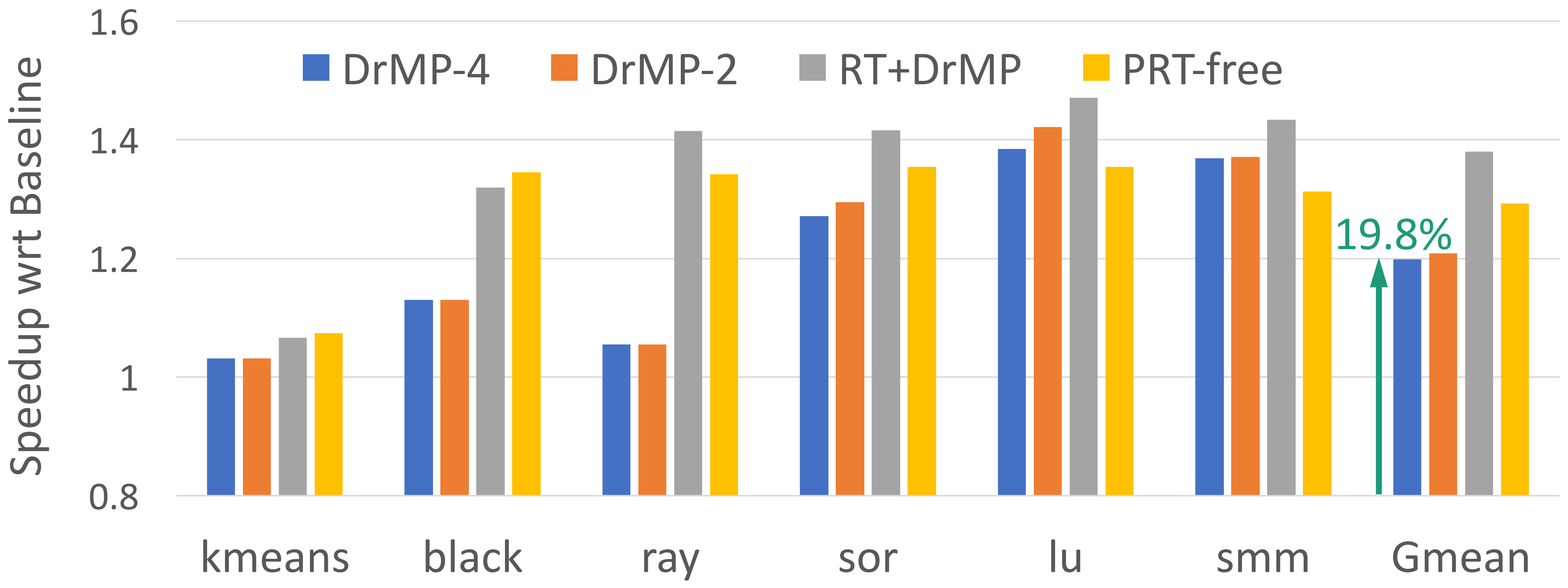
OUTPUT QUALITY



PERFORMANCE

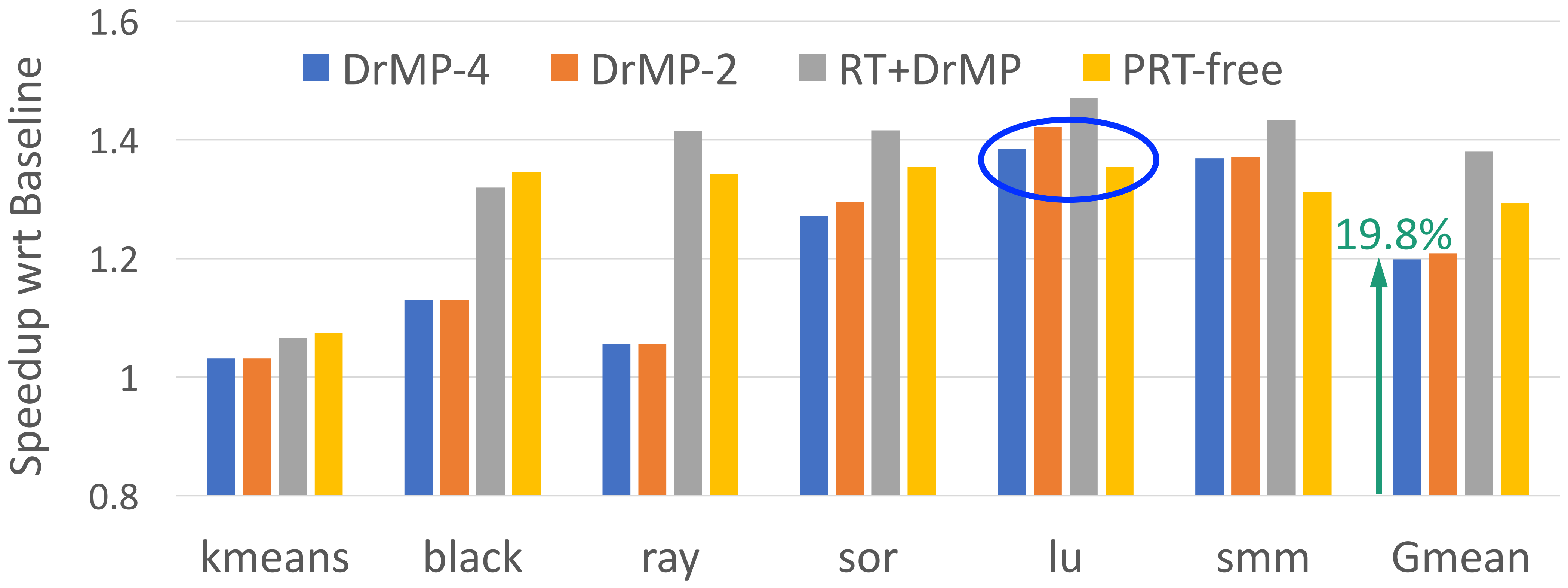


PERFORMANCE



DrMP achieves 19.8% performance improvement

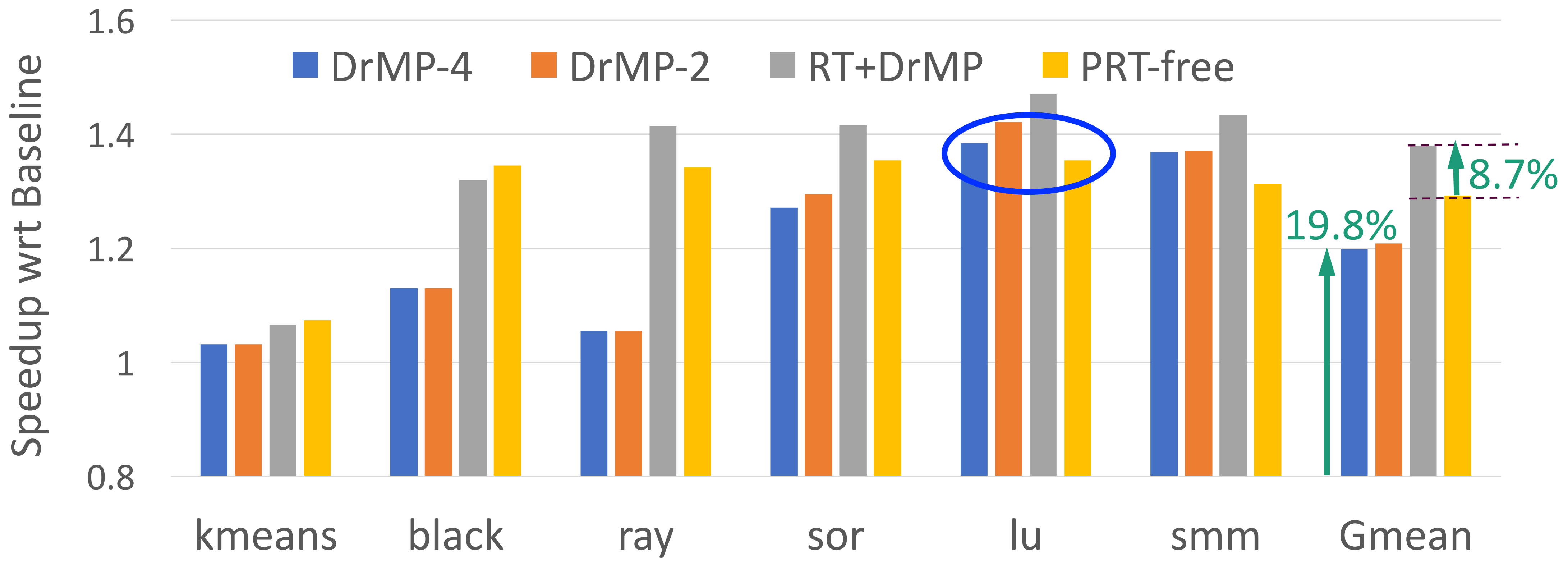
PERFORMANCE



DrMP achieves 19.8% performance improvement

- For apps with dominant approx data accesses, DrMP outperforms PRT-free

PERFORMANCE



DrMP achieves 19.8% performance improvement

- For apps with dominant approx data accesses, DrMP outperforms PRT-free

Orthogonal to RT

- RT+DrMP is 8.7% better than PRT-free

SUMMARY: DrMP

Many applications can tolerate output quality loss
Restore can be used for approximate computing

DrMP: balance restore reductions and accuracy
DrMP’: support both approximate and precise

Output quality: no more than 1% accuracy loss
Performance: 19.8% improvement

OUTLINE

RT-Next

Partial restore based on refresh distance

CkRemap

Fast restore via reorganization and allocation

DrMP

Mitigate restore with approximate computing

Summary and Research Directions

OUTLINE

RT-Next

Partial restore based on refresh distance

CkRemap

Fast restore via reorganization and allocation

DrMP

Mitigate restore with approximate computing

Summary and Research Directions

SUMMARY

DRAM must keep scaling to meet increasing demands
Prolonged restore time has become a major hurdle

RT-next: truncate restore using the time distance to next refresh
CkRemap: construct fast access regions using DRAM organization
DrMP: mitigate restore while guarantee acceptable output loss

Performed pioneering studies on restore via modeling & simu
Developed comprehensive schemes to mitigate restore issue

Supported under NSF grants: CCF-1422331, CNS-1012070, CCF-1535755 and CCF-1617071

COMPARISON TO PRIOR ARTS

Sharing/Sensing timing reduction

- Optimize DRAM internal structures [CHARM'ISCA13, TL-DRAM'HPCA13, etc]
- Utilize existing timing margins [NUAT'HPCA14, AL-DRAM'HPCA15, etc]

DRAM restore studies

- Identify the restore scaling issue [Co-arch'MEM14, tWR'Patent15, etc]
- Reduce restore timings [AL-DRAM'HPCA15, MCR'ISCA15, etc]

Memory-based approximate computing

- Optimize storage density and lifetime [PCM/SSD'MICRO13, PCM'ASPLOS16, etc]
- Skip DRAM refresh [Flikker'ASPLOS11, Alloc'CASES15, etc]

COMPARISON TO PRIOR ARTS

sense

Sharing/Sensing timing reduction

- Optimize DRAM internal structures [CHARM'ISCA13, TL-DRAM'HPCA13, etc]
- Utilize existing timing margins [NUAT'HPCA14, AL-DRAM'HPCA15, etc]

We are working at orthogonal restore issue in future DRAMs

restore

DRAM restore studies

- Identify the restore scaling issue [Co-arch'MEM14, tWR'Patent15, etc]
- Reduce restore timings [AL-DRAM'HPCA15, MCR'ISCA15, etc]

approx

Memory-based approximate computing

- Optimize storage density and lifetime [PCM/SSD'MICRO13, PCM'ASPLOS16, etc]
- Skip DRAM refresh [Flikker'ASPLOS11, Alloc'CASES15, etc]

COMPARISON TO PRIOR ARTS

sense

Sharing/Sensing timing reduction

- Optimize DRAM internal structures [CHARM'ISCA13, TL-DRAM'HPCA13, etc]
- Utilize existing timing margins [NUAT'HPCA14, AL-DRAM'HPCA15, etc]

We are working at orthogonal restore issue in future DRAMs

restore

DRAM restore studies

- Identify the restore scaling issue [Co-arch'MEM14, tWR'Patent15, etc]
- Reduce restore timings [AL-DRAM'HPCA15, MCR'ISCA15, etc]

We are working at future DRAMs with more effective solutions

approx

Memory-based approximate computing

- Optimize storage density and lifetime [PCM/SSD'MICRO13, PCM'ASPLOS16, etc]
- Skip DRAM refresh [Flikker'ASPLOS11, Alloc'CASES15, etc]

COMPARISON TO PRIOR ARTS

sense

Sharing/Sensing timing reduction

- Optimize DRAM internal structures [CHARM’ISCA13, TL-DRAM’HPCA13, etc]
- Utilize existing timing margins [NUAT’HPCA14, AL-DRAM’HPCA15, etc]

We are working at orthogonal restore issue in future DRAMs

restore

DRAM restore studies

- Identify the restore scaling issue [Co-arch’MEM14, tWR’Patent15, etc]
- Reduce restore timings [AL-DRAM’HPCA15, MCR’ISCA15, etc]

We are working at future DRAMs with more effective solutions

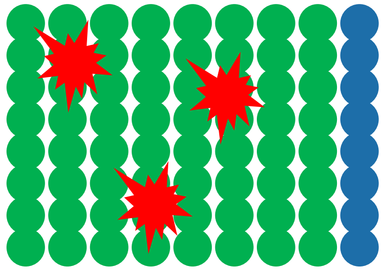
approx

Memory-based approximate computing

- Optimize storage density and lifetime [PCM/SSD’MICRO13, PCM’ASPLOS16, etc]
- Skip DRAM refresh [Flikker’ASPLOS11, Alloc’CASES15, etc]

We are the first work on restore-based approximation

FUTURE RESEARCH DIRECTIONS

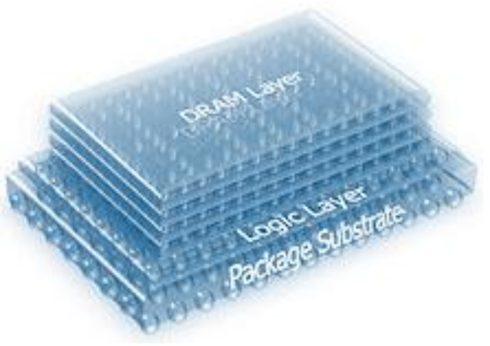


Solve restore from **reliability** perspective

- Treat Slow restore cells as faulty ones
- Design stronger error correction codes

Study **security** issues of restore variation

- Restore variation info is DRAM's fingerprint
- Solve both info leakage and slow restore



Explore restore in 3D **stacked** DRAM

- Stacking has thermal management issue
- Reduce restore with temperature-aware solutions

PUBLICATIONS

Xianwei Zhang, Youtao Zhang, Bruce Childers and Jun Yang
[HPCA’2016] Restore Truncation for Performance Improvement in Future DRAM Systems

Xianwei Zhang, Youtao Zhang, Bruce Childers and Jun Yang
[TODAES’2017] On the Restore Time Variations of Future DRAM Memory
[DATE’2015] Exploiting DRAM Restore Time Variations in Deep Sub-micron Scaling

Xianwei Zhang, Youtao Zhang, Bruce Childers and Jun Yang
[PACT’2017] DrMP: Mixed Precision-aware DRAM for High Performance Approximate and Precise Computing
[MemSys’2016] AWARD: Approximation-aWAre Restore in Further Scaling DRAM

Xianwei Zhang, Lei Zhao, Youtao Zhang and Jun Yang
[ICCD’2015] Exploit Common Source-Line to Construct Energy Efficient Domain Wall Memory based Caches
Xianwei Zhang, Youtao Zhang and Jun Yang
[ICCD’2015] DLB: Dynamic Lane Borrowing for Improving Bandwidth and Performance in Hybrid Memory Cube
[ICCD’2015] TriState-SET: Proactive SET for Improved Performance in MLC Phase Change Memories
Xianwei Zhang, Lei Jiang, Youtao Zhang, Chuanjun Zhang and Jun Yang
[ISLPED’2013] WoM-SET: Lowering Write Power of Proactive-SET based PCM Write Strategy Using WoM Code

ACKNOWLEDGEMENTS

Profs. *Youtao Zhang, Bruce Childers* and *Jun Yang*
- great guidance, and all resources

Profs. *Wonsun Ahn* and *Guangyong Li*
- valuable inputs into research studies

UPitt and NSF
- financial supports (TA/Fellowship and Research grants)

All members in the lab
- insightful discussions

Friends and colleagues
- help both in and outside researches

Family
- endless support and always understand

PhD Thesis Defense

Jul 14, 2017 (Friday)

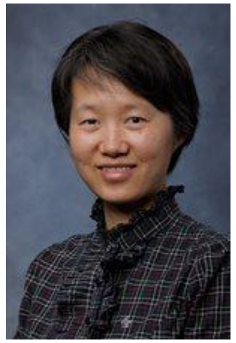
Addressing Prolonged Restore Challenges in Further Scaling DRAMs

Xianwei Zhang

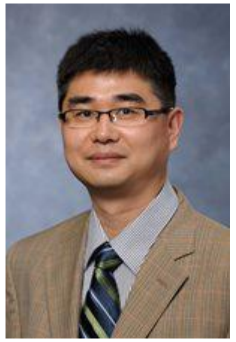
Committees:

Youtao Zhang (advisor)
CS, Pitt

Bruce R. Childers
CS, Pitt


Jun Yang
ECE, Pitt

Wonsun Ahn
CS, Pitt


Guangyong Li
ECE, Pitt