ERUCA

Efficient DRAM **R**esource **U**tilization and Resource **C**onflict **A**voidance for Memory System Parallelism

Sangkug Lym, Heonjae Ha, Yongkee Kwon, Chun-kai Chang,
Jungrae Kim, Mattan Erez

TEXAS
The University of Texas at Austin

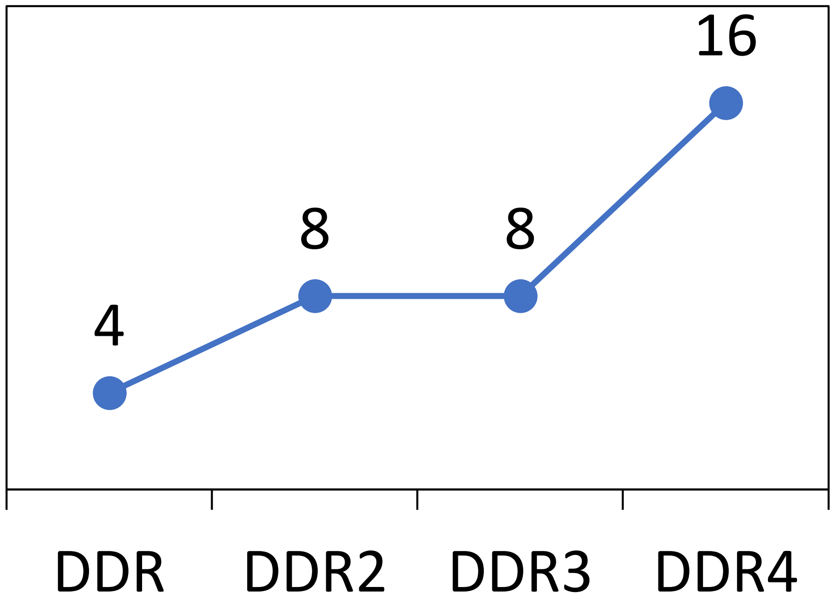
Stanford University

Microsoft

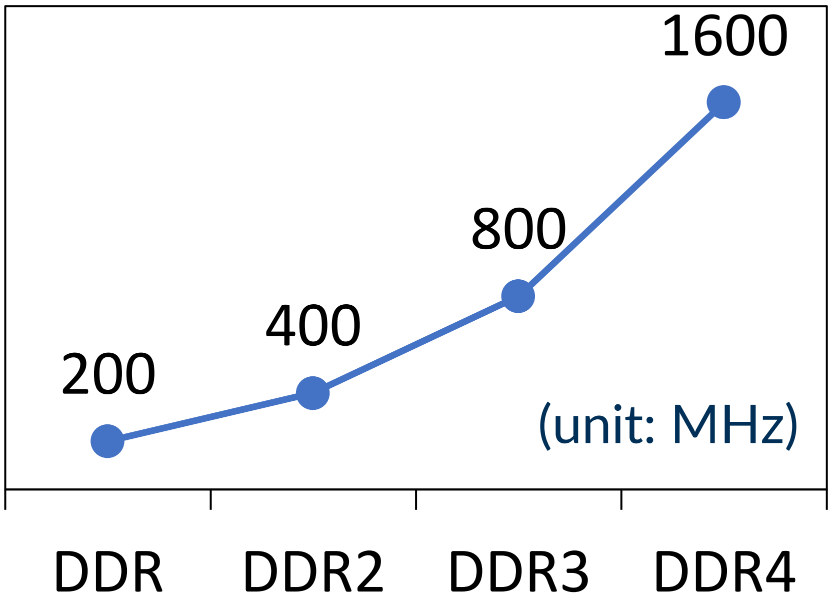
Trends in DRAM Generations

Memory system performance has improved by

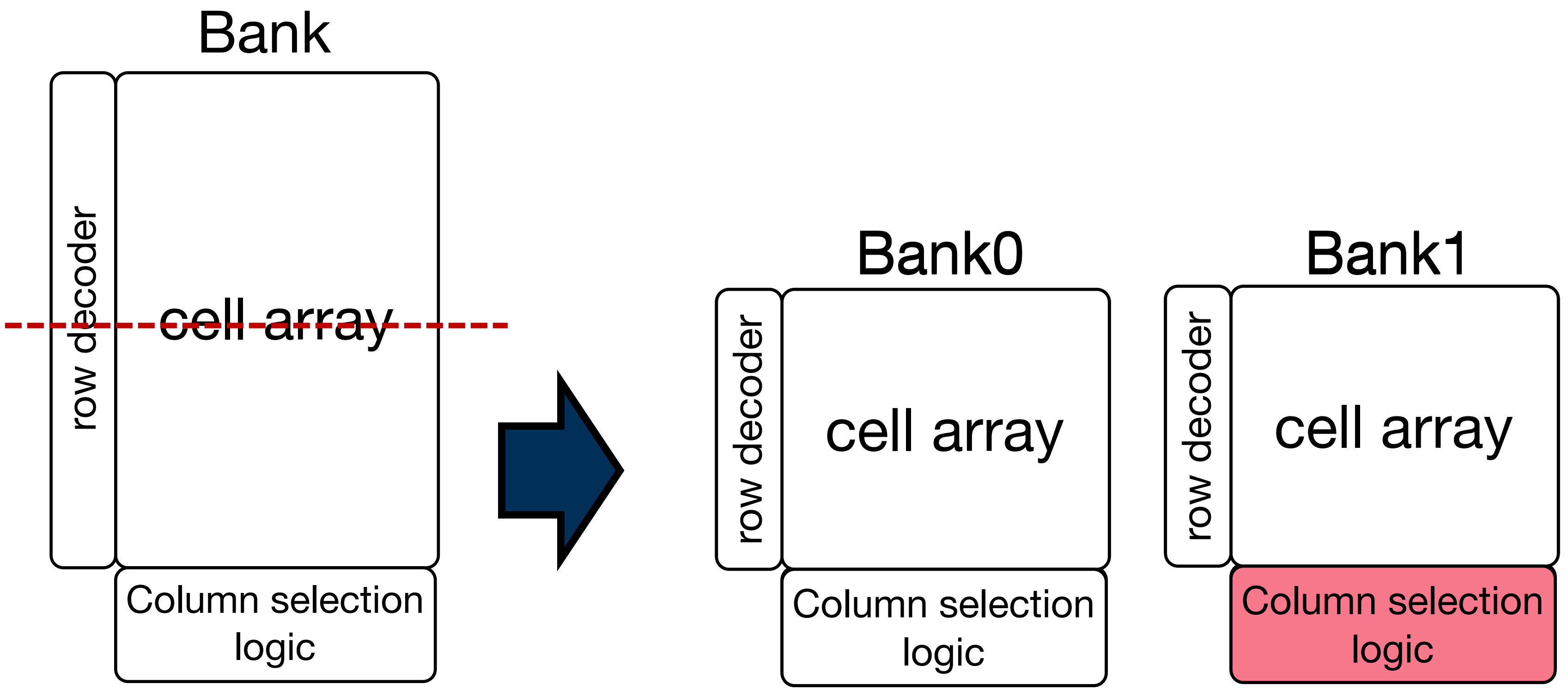
Increasing # banks



Scaling data bandwidth

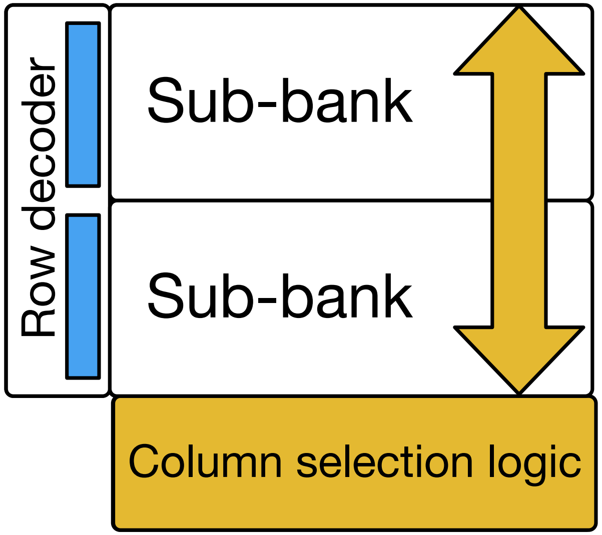


DRAM Bank Parallelism

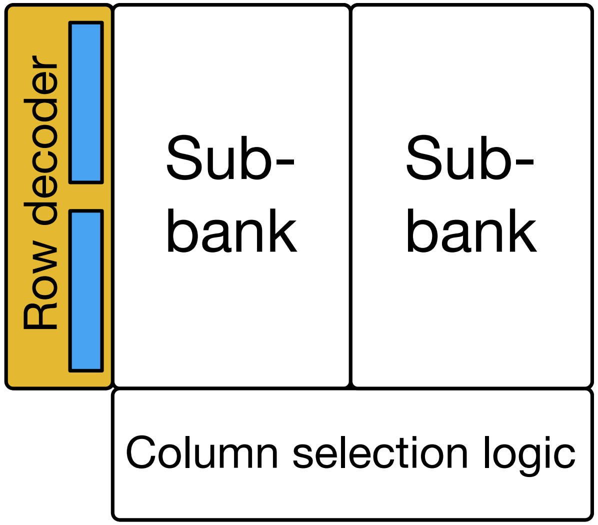


$$\frac{Bits}{DRAM\ chip\ size\ (= Cell\ area + \boldsymbol{peripheral\ logic})}$$

Sub-Banking



Bitline direction sub-banking



Wordline direction sub-banking

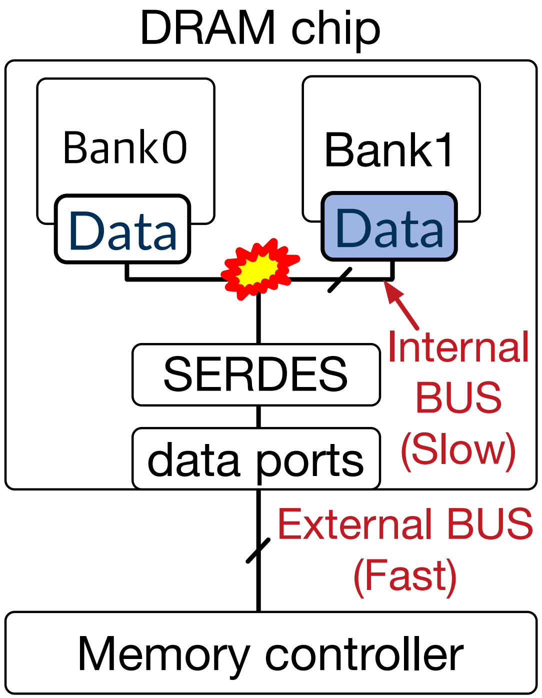
🟦 A set of row address latches

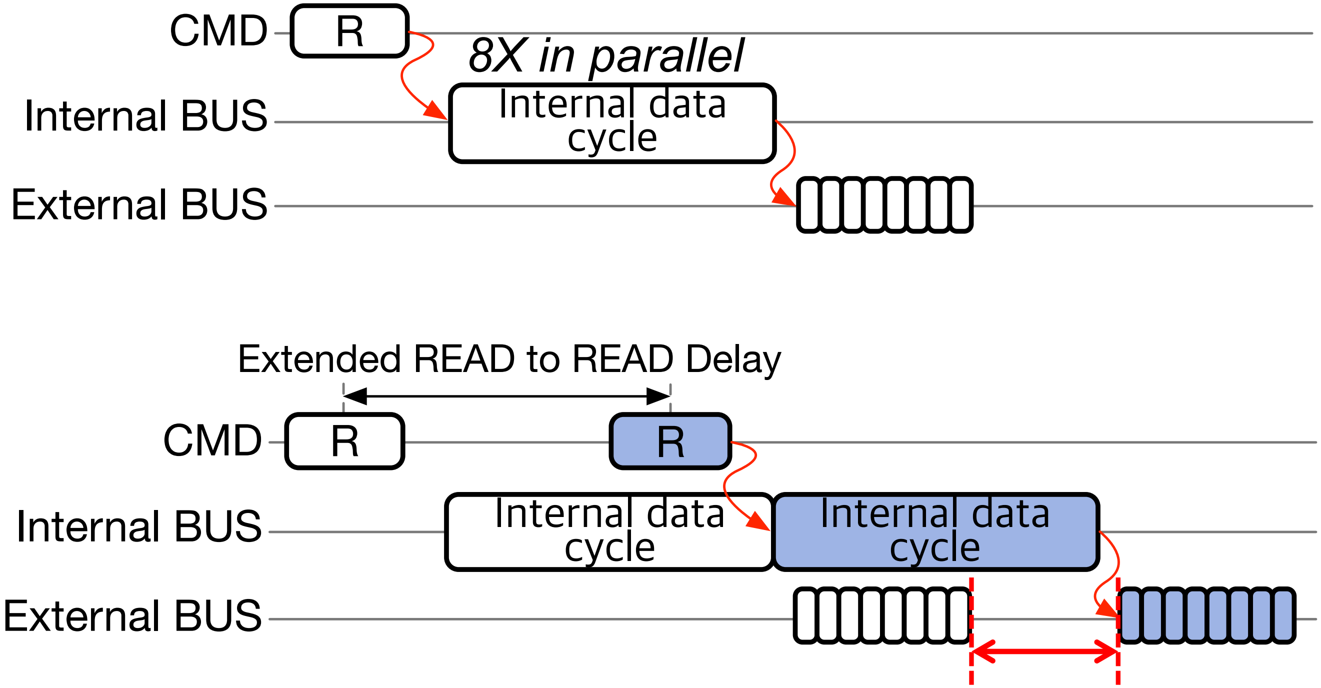
🟨 Shared between sub-banks

Sub-banking improves effective bank parallelism

Shared resource conflicts restrict memory parallelism

Channel Bandwidth Scaling

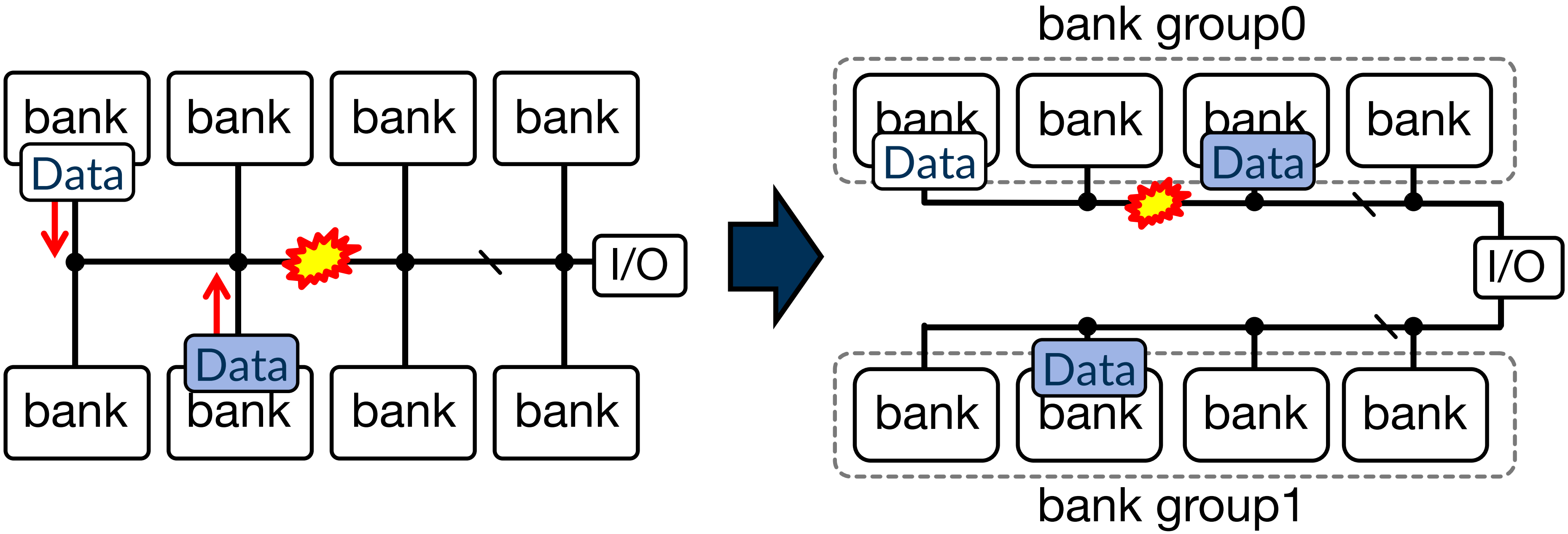




Wasted bandwidth

Conflicts on the shared data bus may delay some memory accesses

Bank Grouping



The shared in-DRAM data bus restricts memory level parallelism

DRAM is Cost Sensitive

SALP (Sub-array level parallelism) *[Kim et al. ISCA'12]*

- Area overhead: **3.03%**(MASA4), **4.75%**(MASA8)

Half-DRAM (Split a DRAM tile) *[Zhang et al. ISCA'14]*

- Area overhead: **1.46%**

Bank grouping

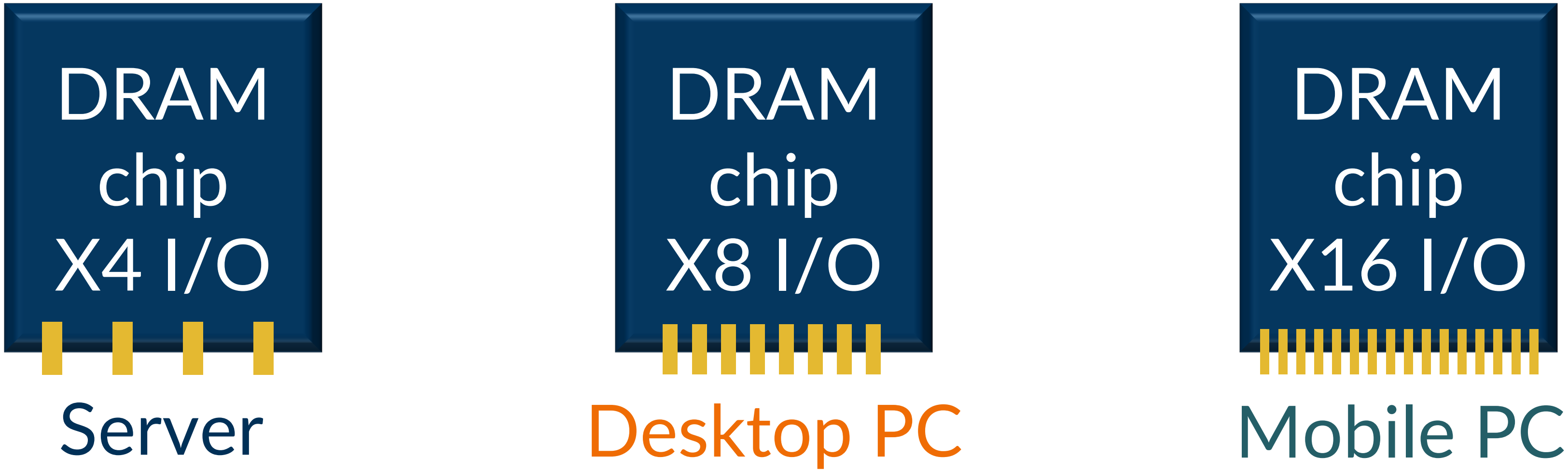
- **Double the chip-wise global data path width**

Need more memory parallelism

but at near-zero cost

Underutilized Resources?

DRAM Chips by I/O Width



DRAM chip X4 I/O

Server

DRAM chip X8 I/O

Desktop PC

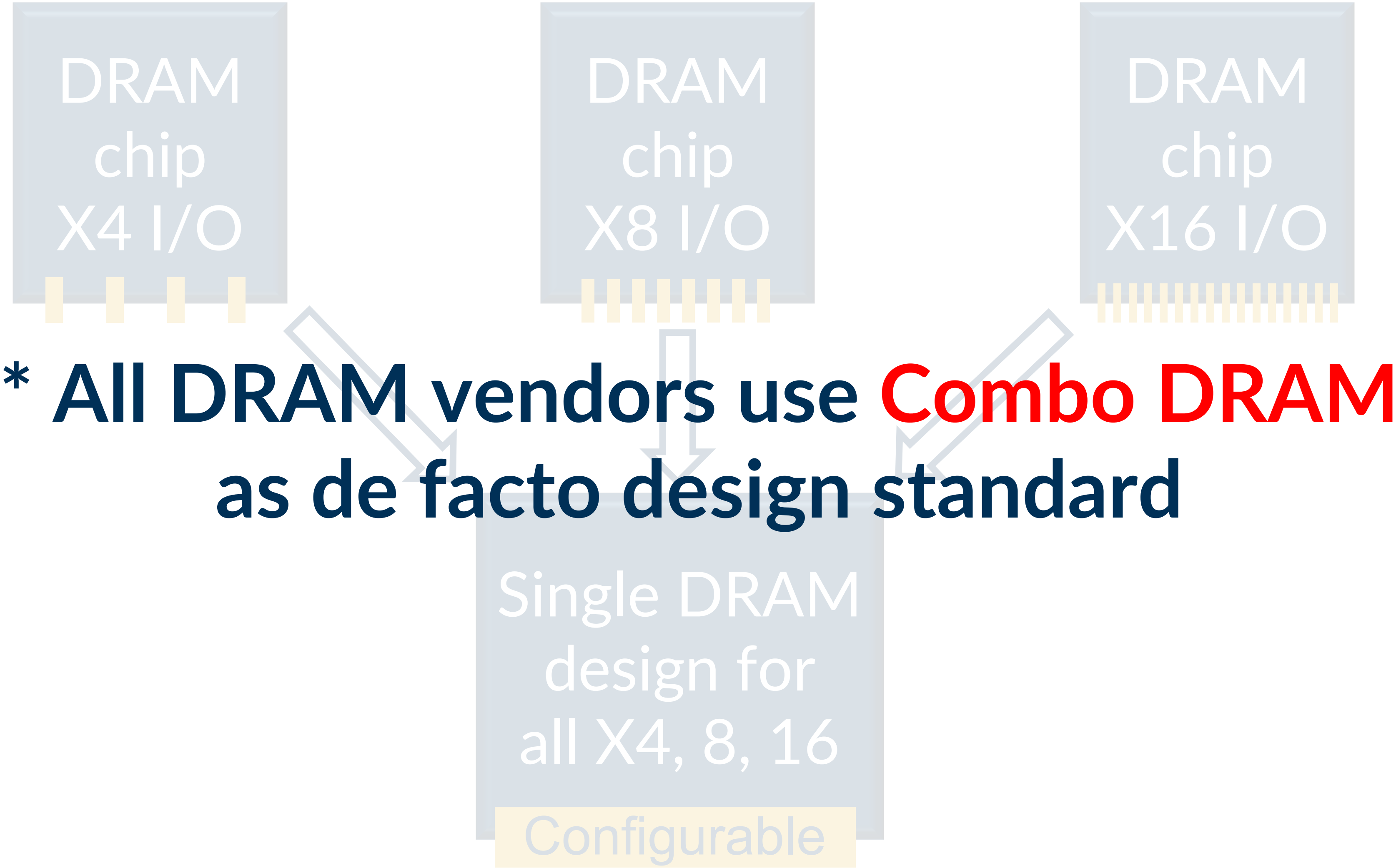
DRAM chip X16 I/O

Mobile PC

Individual DRAM chip design is expensive because

1. Development cost
2. Testing cost
3. Market uncertainty

Combo DRAM Architecture



DRAM chip X4 I/O

DRAM chip X8 I/O

DRAM chip X16 I/O

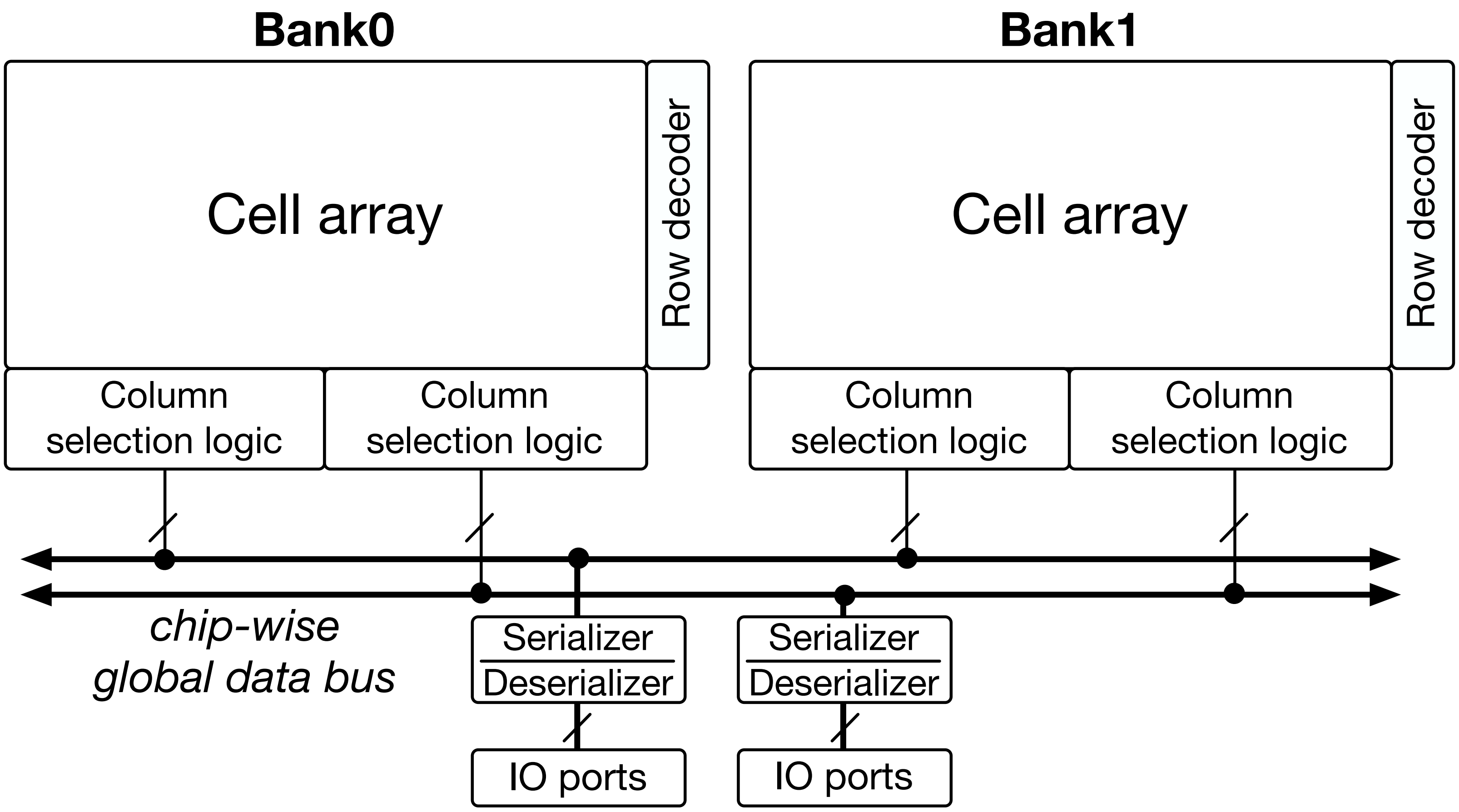
Single DRAM design for all X4, 8, 16

Configurable

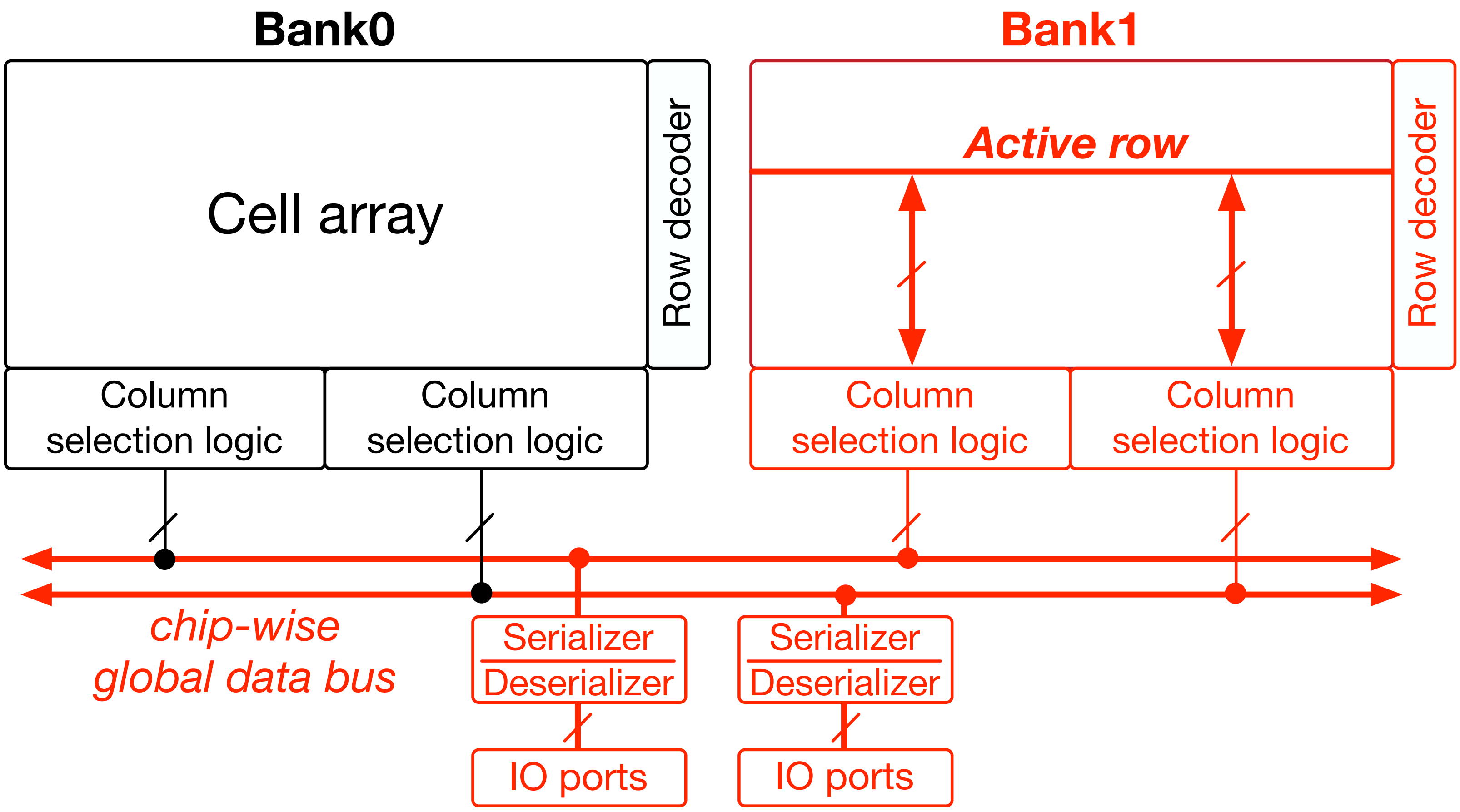
*** All DRAM vendors use Combo DRAM as de facto design standard**

** DDR3, DDR4 datasheets SK hynix, Samsung, Micron*

Combo DRAM (X8 chip)

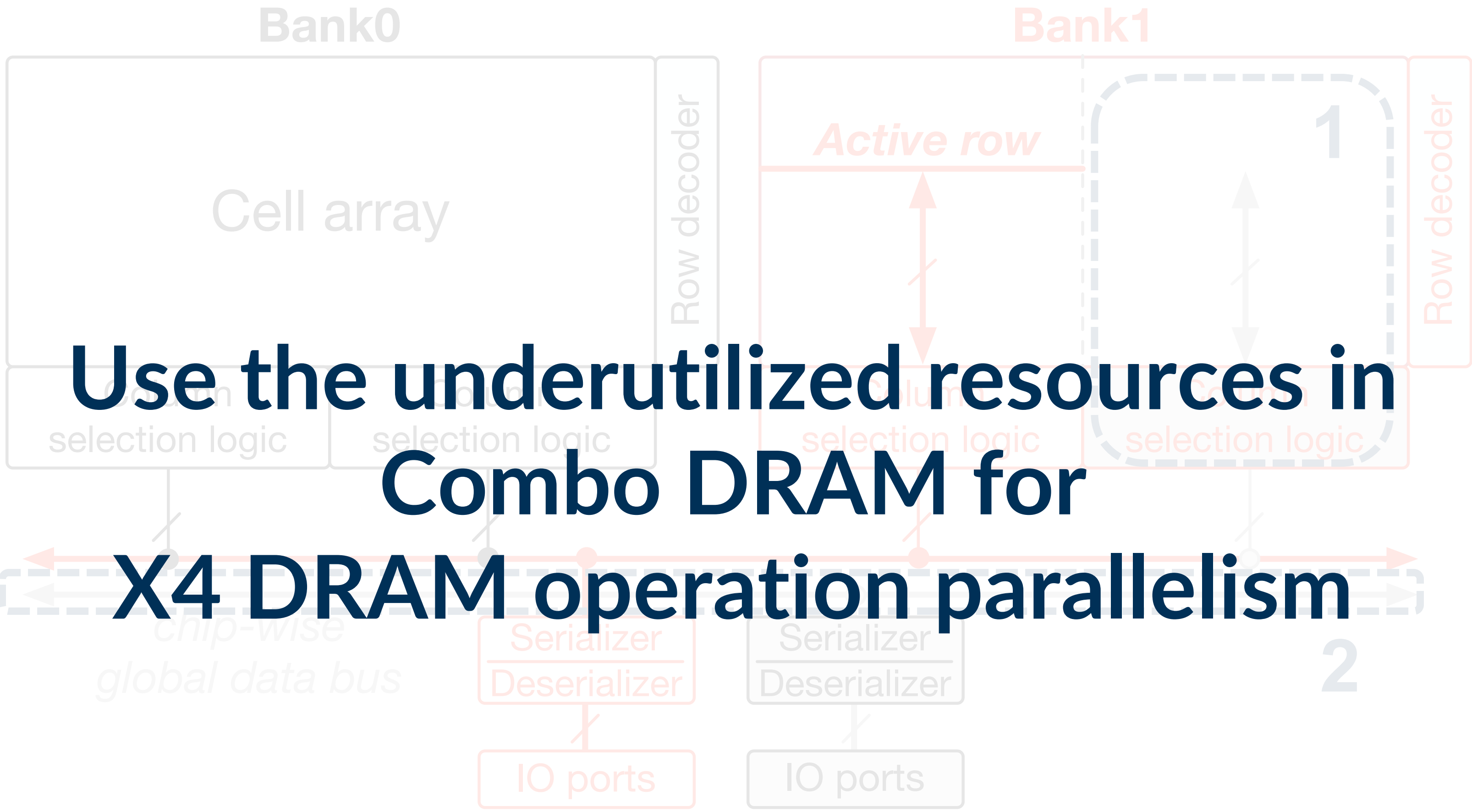


Combo DRAM (X8 chip)



All resources fully utilized!

Combo DRAM (X4 chip)



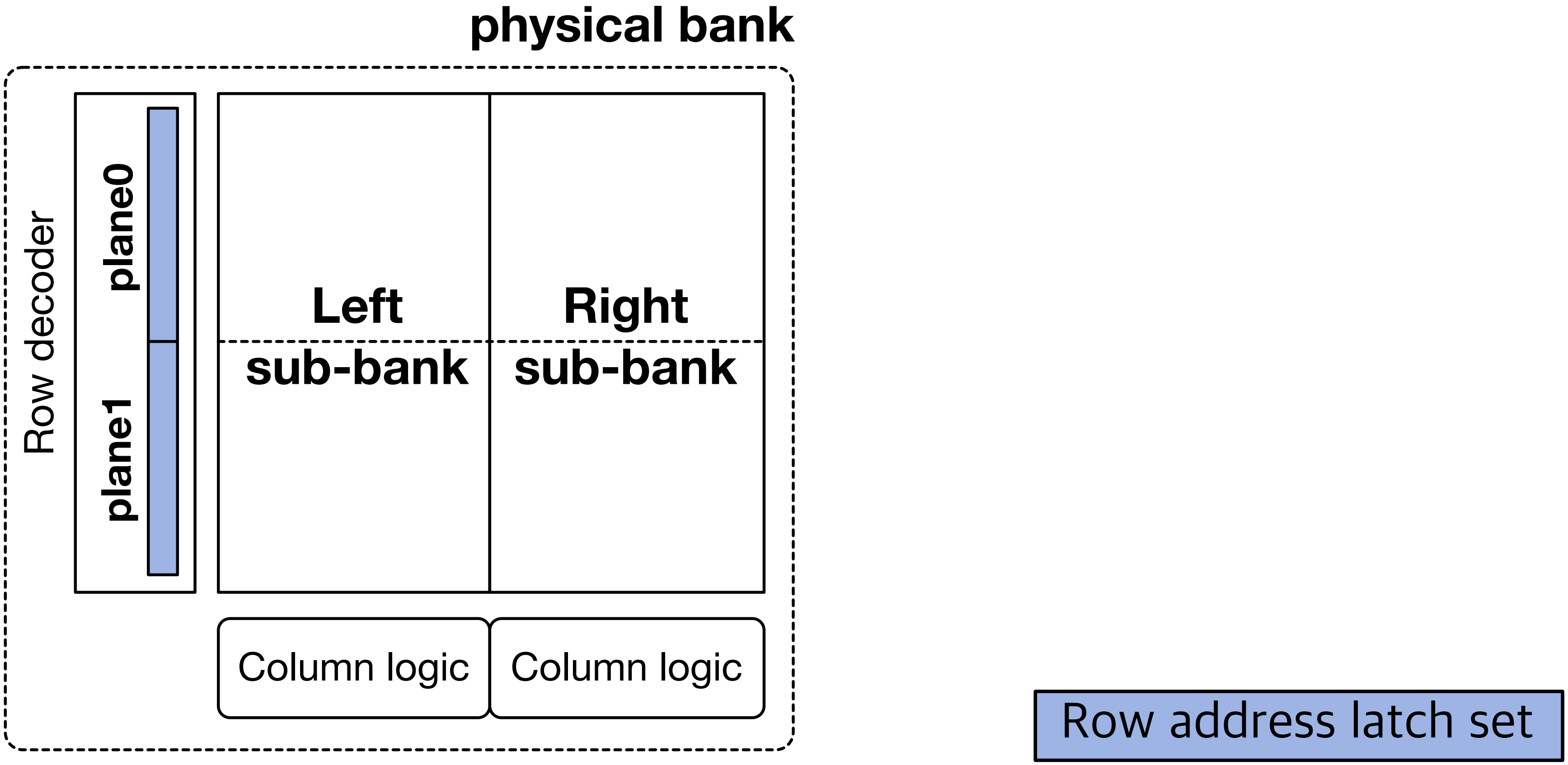
Use the underutilized resources in Combo DRAM for X4 DRAM operation parallelism

1. Bank column logic → Only half used at a time
2. Chip-wise global data bus → Half is not used

Sub-banking

(VSB, EWLR, RAP)

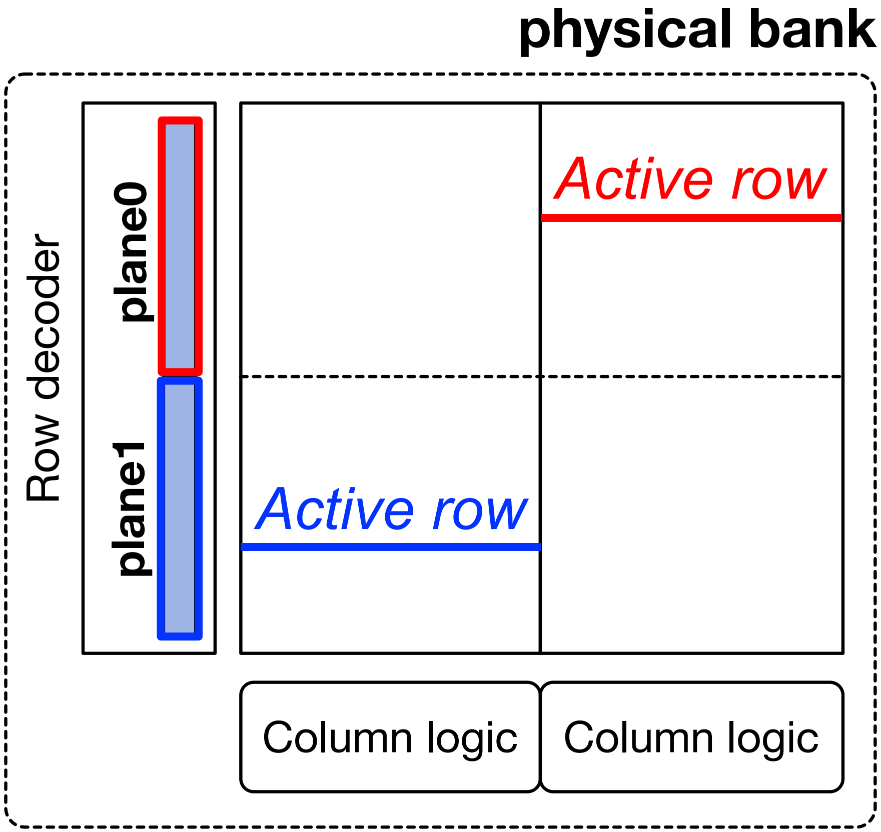
VSB: Baseline Sub-bank Design



Build a pair of sub-banks using Combo DRAM

Vertical Sub-Bank (VSB)

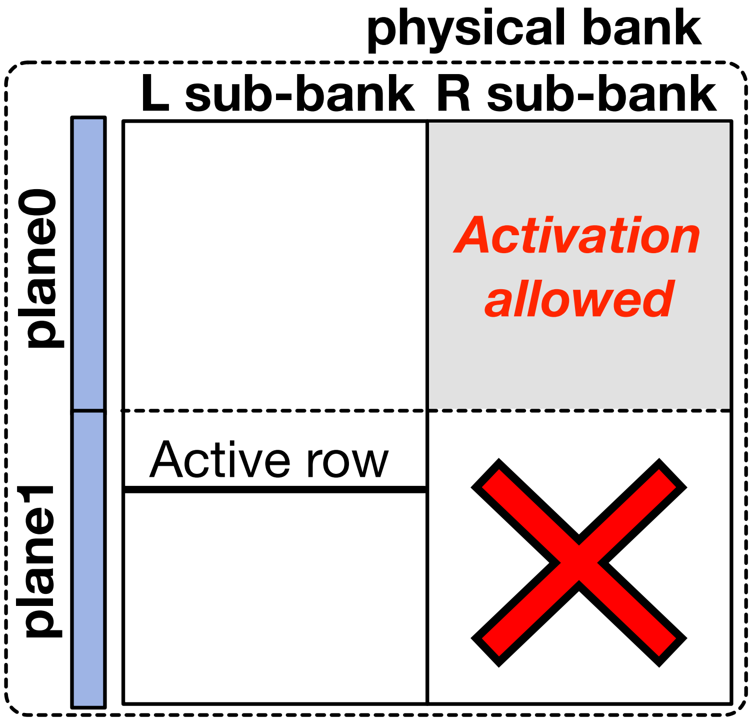
VSB: Baseline Sub-bank Design



Row address latch set

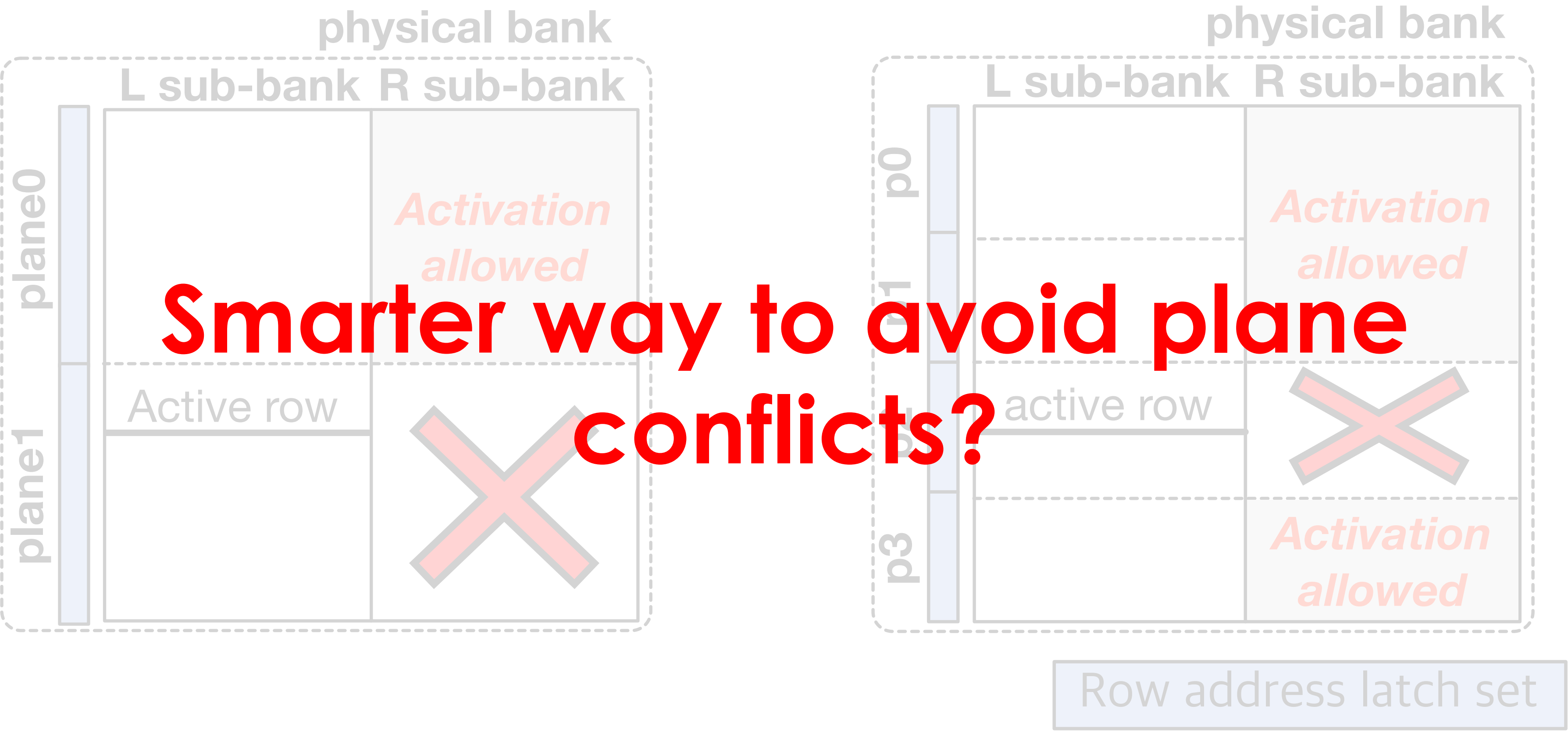
Plane: A partition of row addresses that share row address latches

Plane Conflicts



Plane conflict: Sub-bank interleaving has conflicts on the shared row address latches

Multi-Plane VSB



Smarter way to avoid plane conflicts?

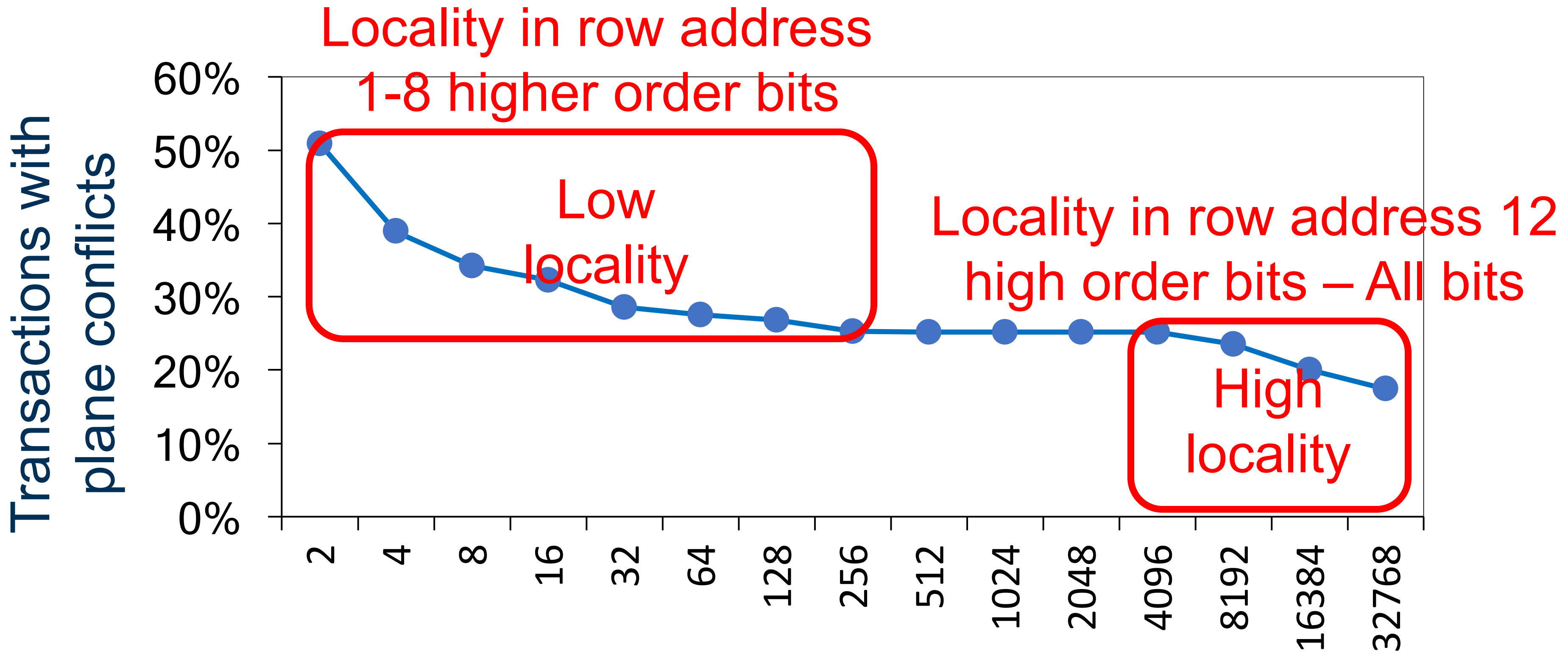
Fewer rows within a plane

→ Reduced chance for plane conflicts

But, area cost & reduced faulty row repair flexibility

Plane Conflicts by # Planes

Check how many *plane conflicts* actually occur



# planes in a bank with 6K wordlines

EWLR: Effective Wordline Range

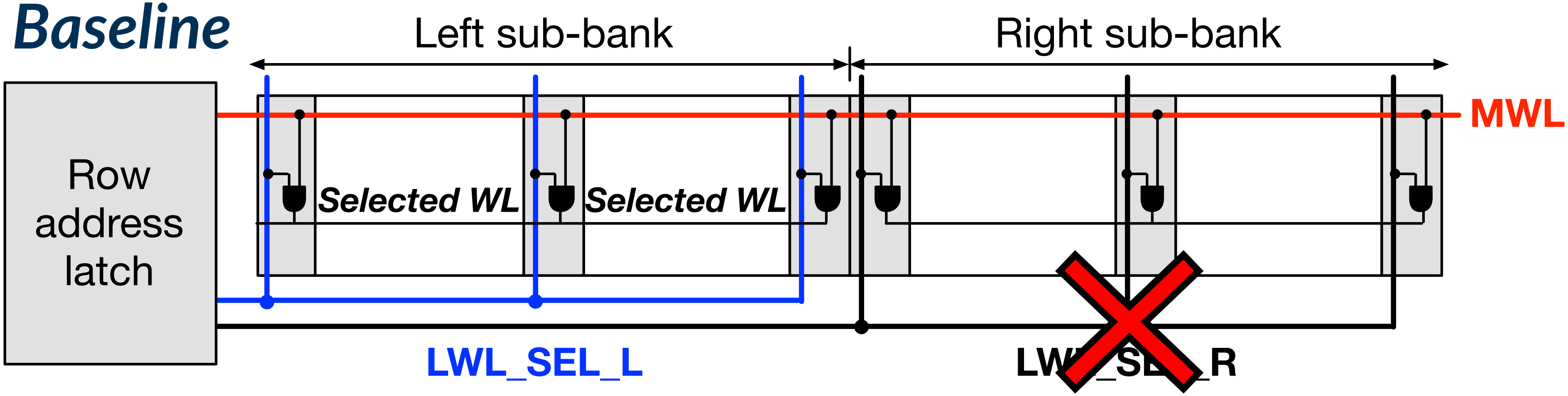
Observation: Transactions to the two sub-banks have high row address locality

EWLR *avoids* plane conflicts in the row address range with high locality

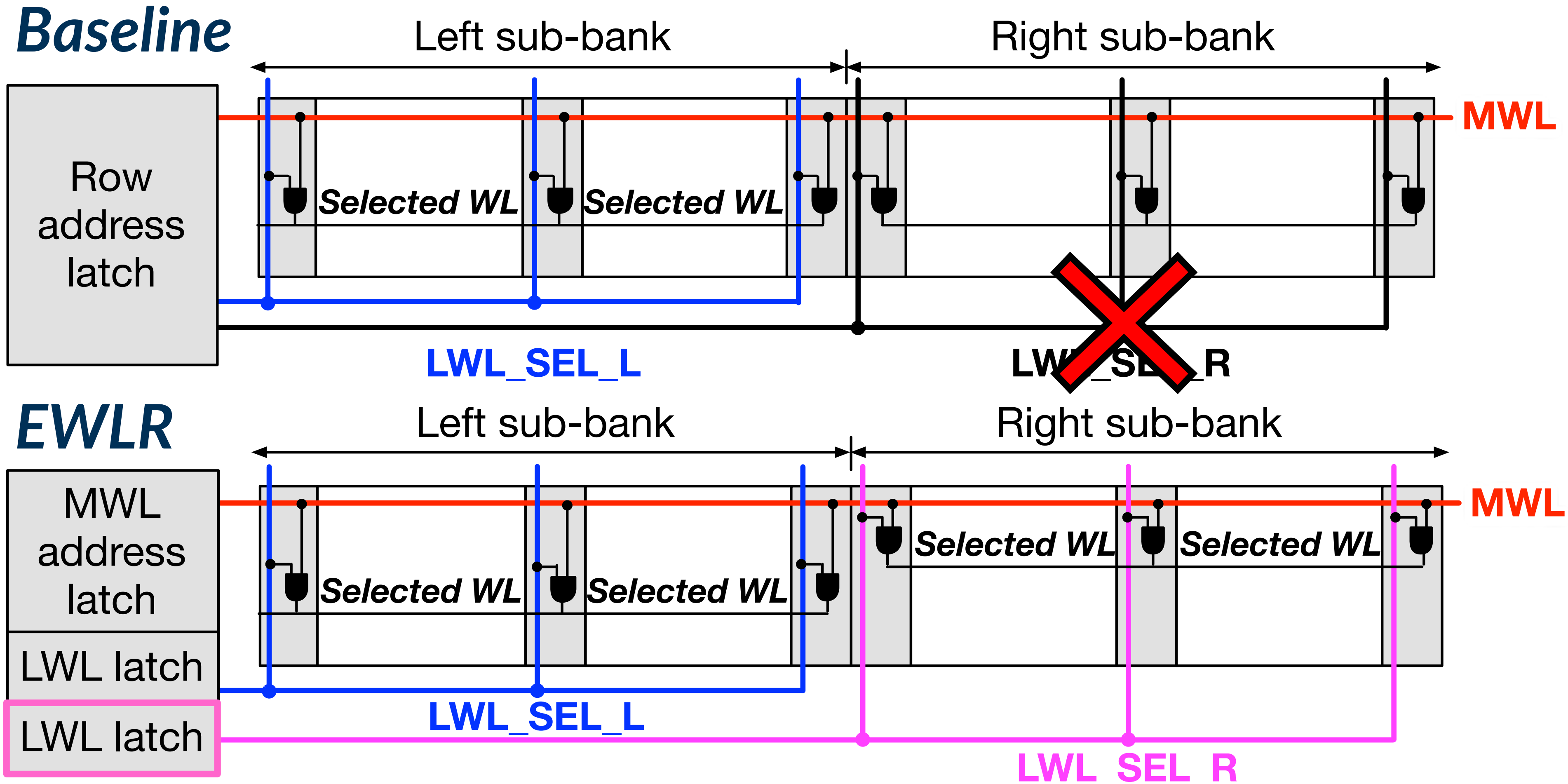
EWLR enables a subset of addresses to be activated across sub-bank pairs

EWLR: Effective Wordline Range

Baseline

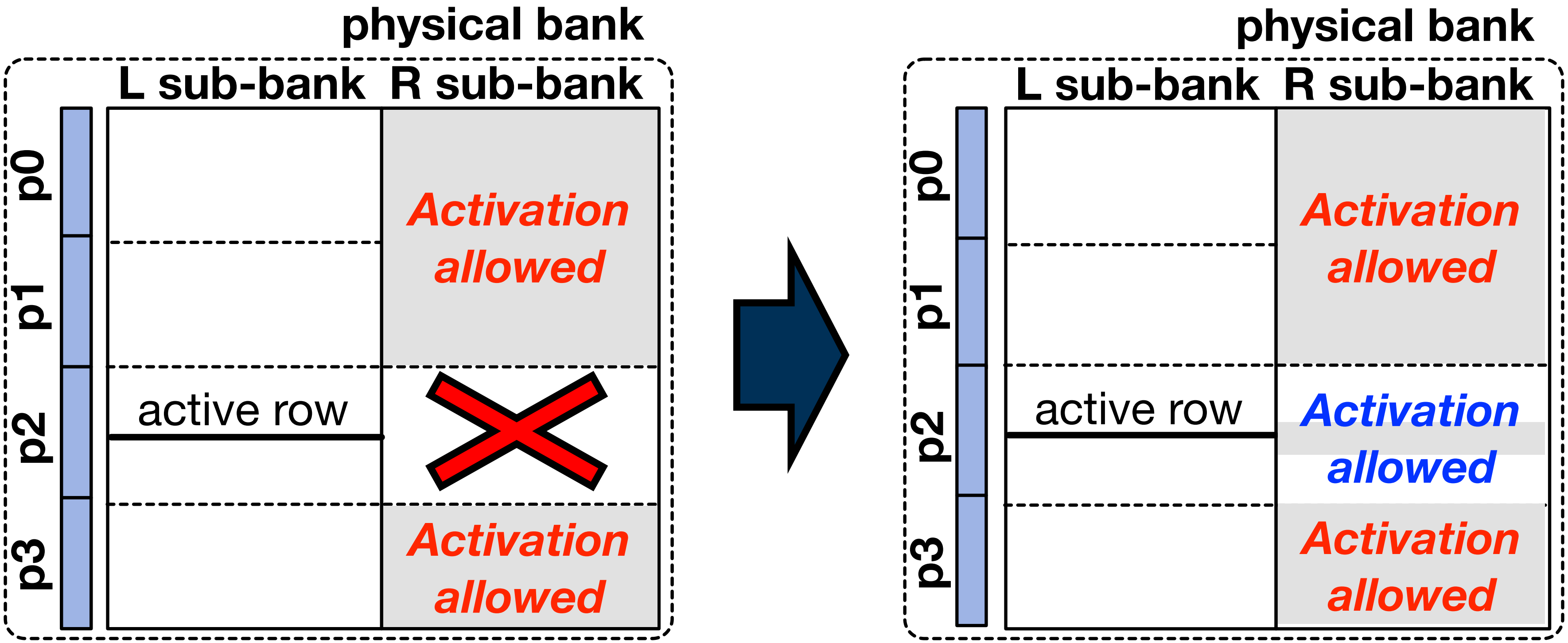


EWLR: Effective Wordline Range



Individual WL selection for each sub-bank within the shared MWL range

EWLR: Wider Sub-bank parallelism

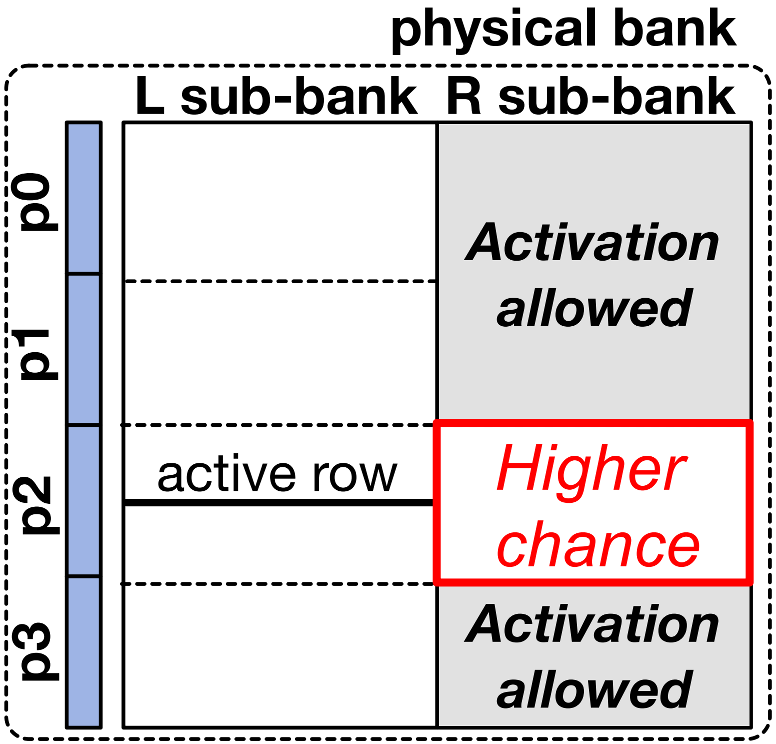


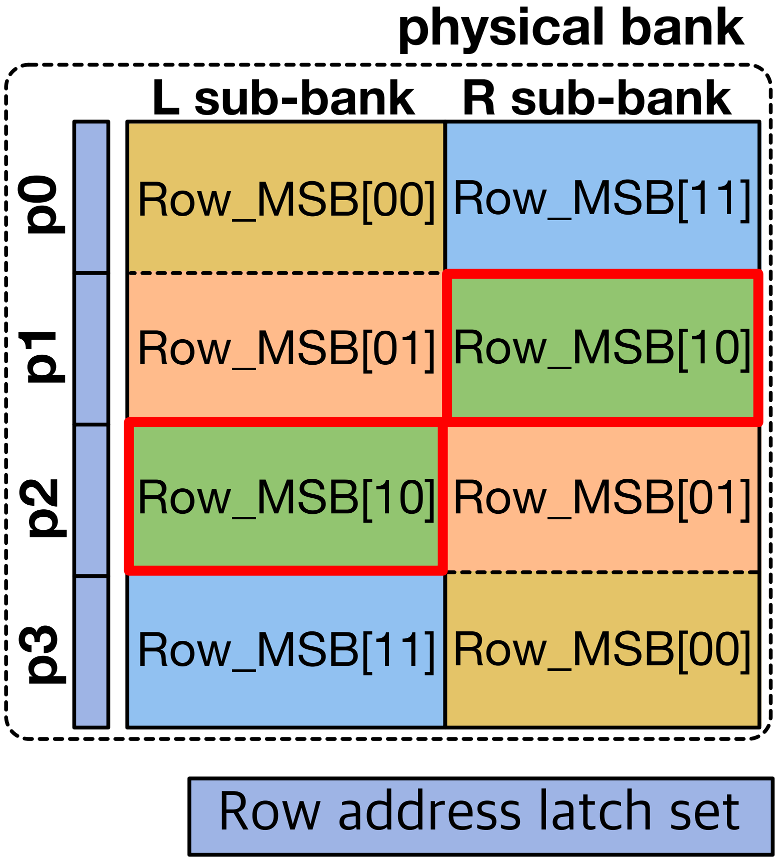
Row address latch set

RAP: Row Address Permutation

RAP removes plane conflicts for both low and high address bit locality

RAP: Row Address Permutation

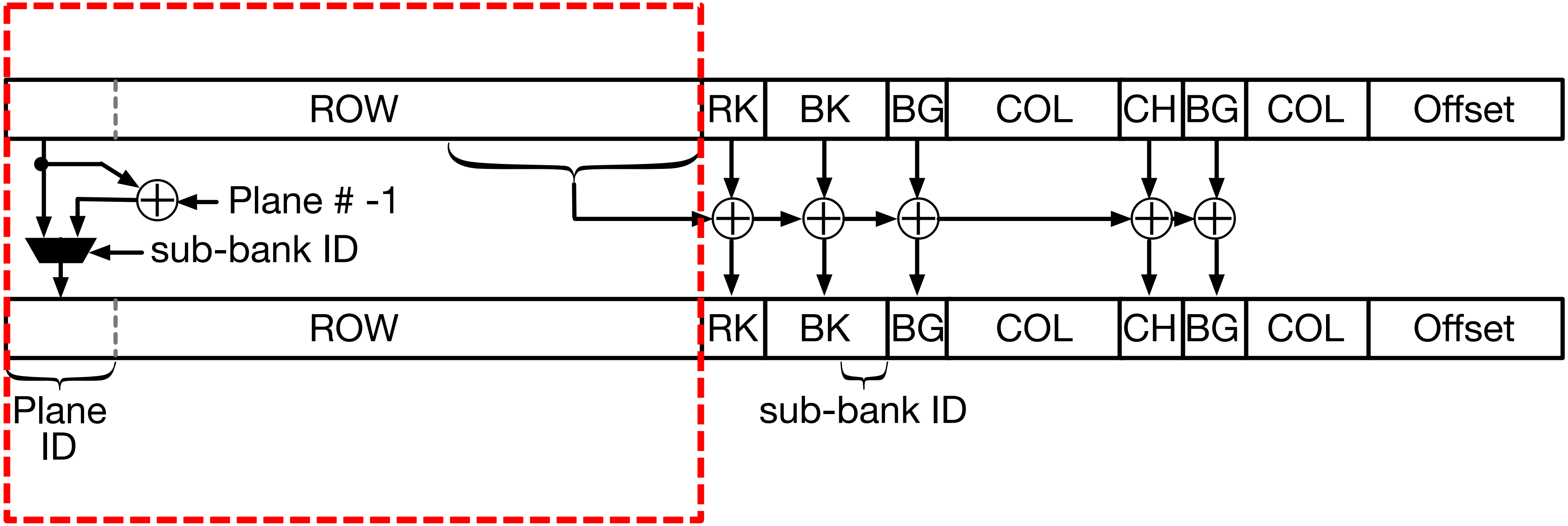




	L sub-bank	R sub-bank
p0	Row_MSB[00]	Row_MSB[11]
p1	Row_MSB[01]	Row_MSB[10]
p2	Row_MSB[10]	Row_MSB[01]
p3	Row_MSB[11]	Row_MSB[00]

Map the rows with high locality to other planes

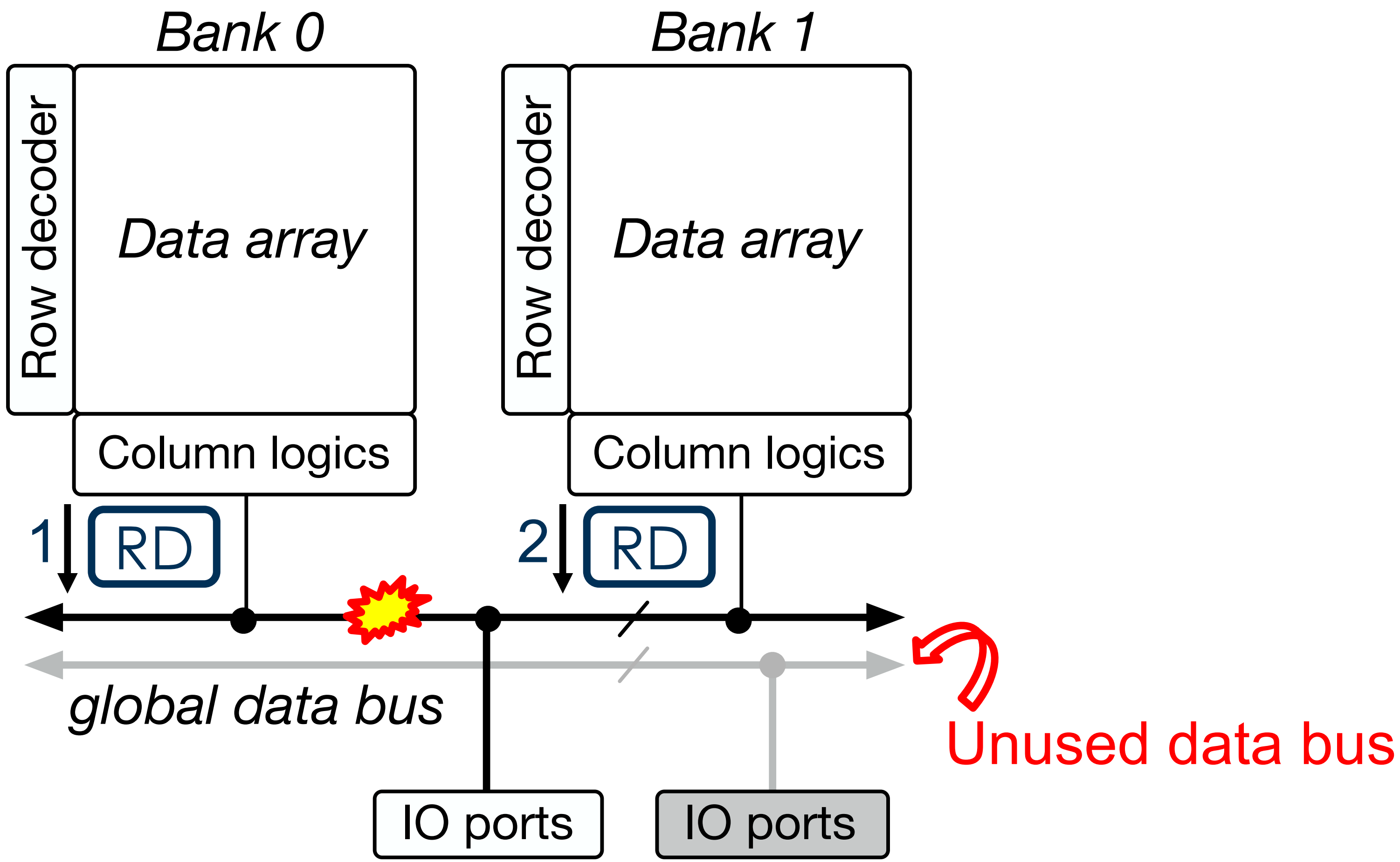
RAP: Row Address Permutation



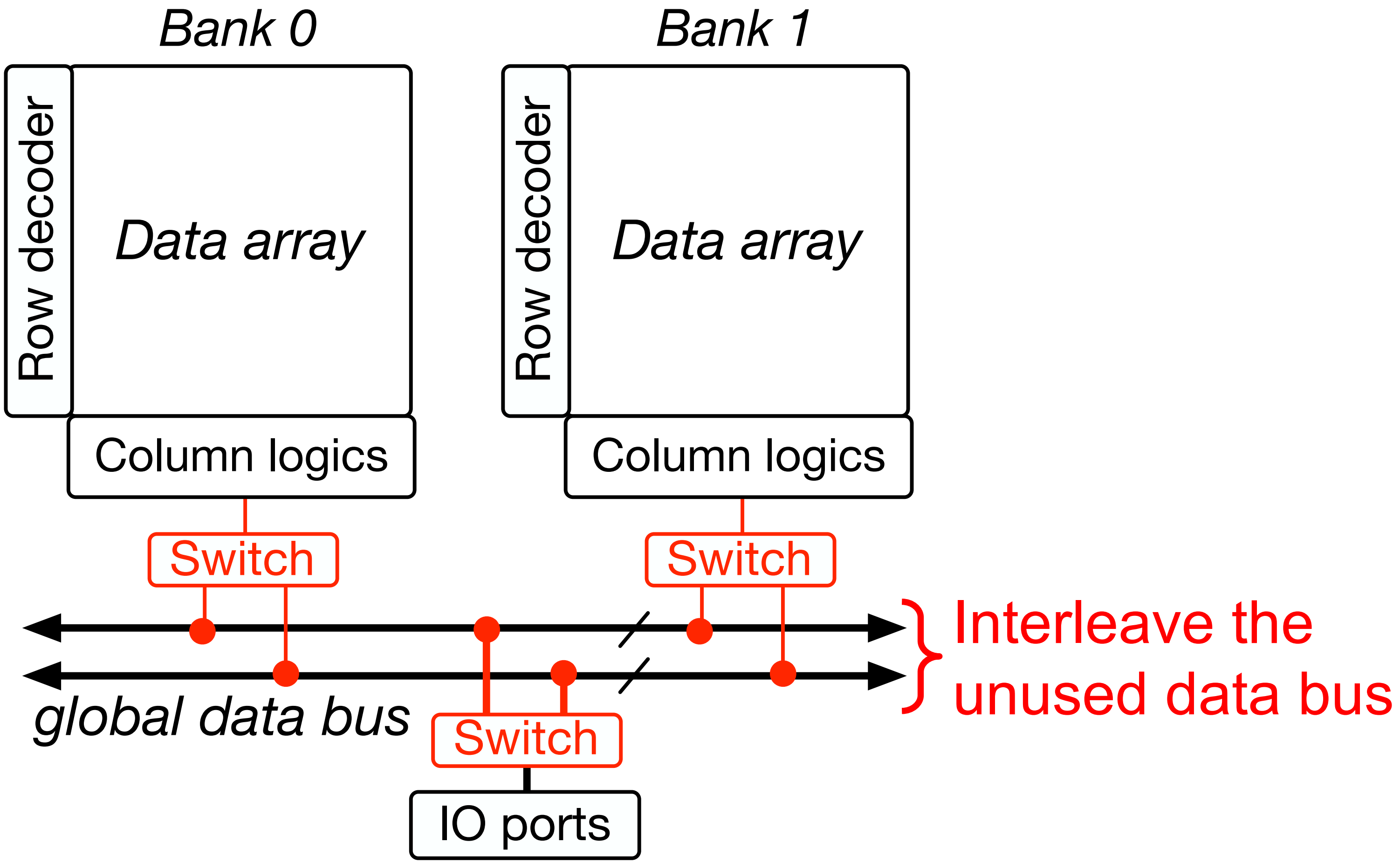
RAP does not require any DRAM changes

Dual Data Bus (DDB)

Conflicts on Global Data Bus



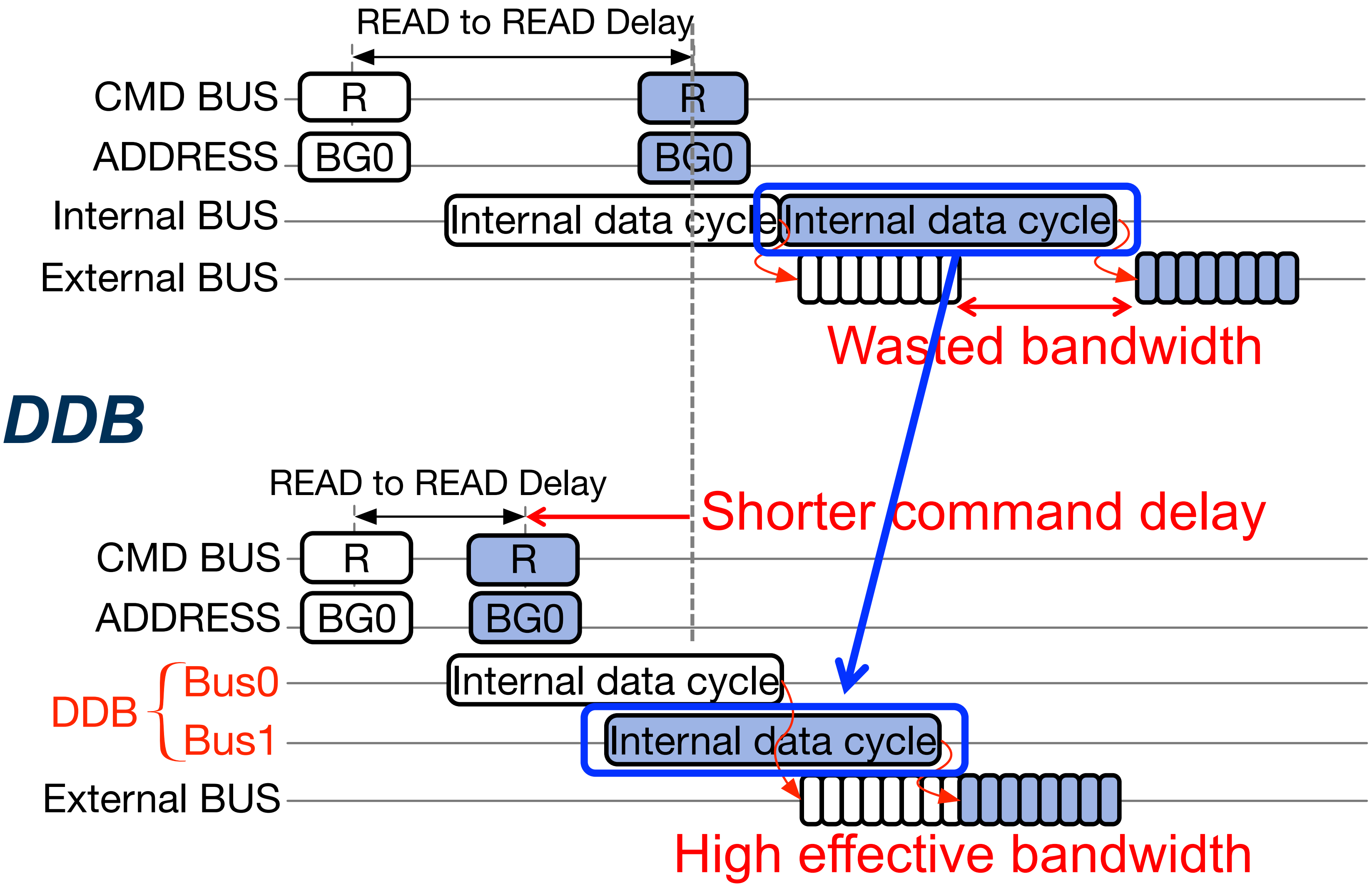
DDB (Dual Data Bus)



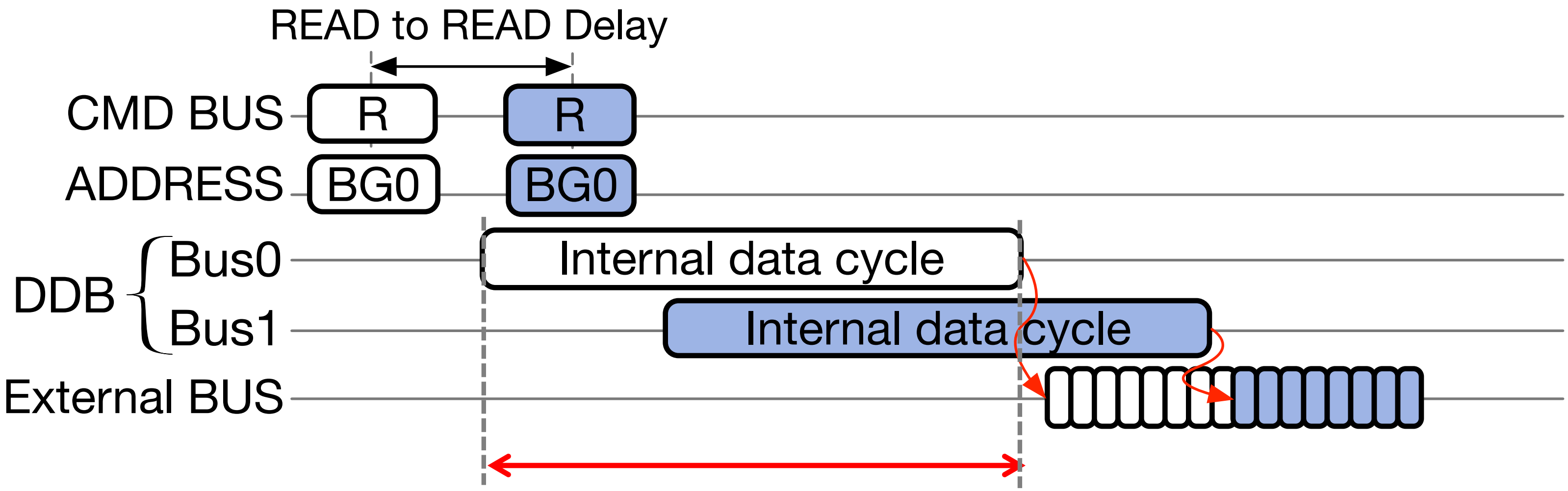
Interleave the unused data bus

Build a switchable dual data bus using the unused data bus in Combo DRAM

Timing Diagram



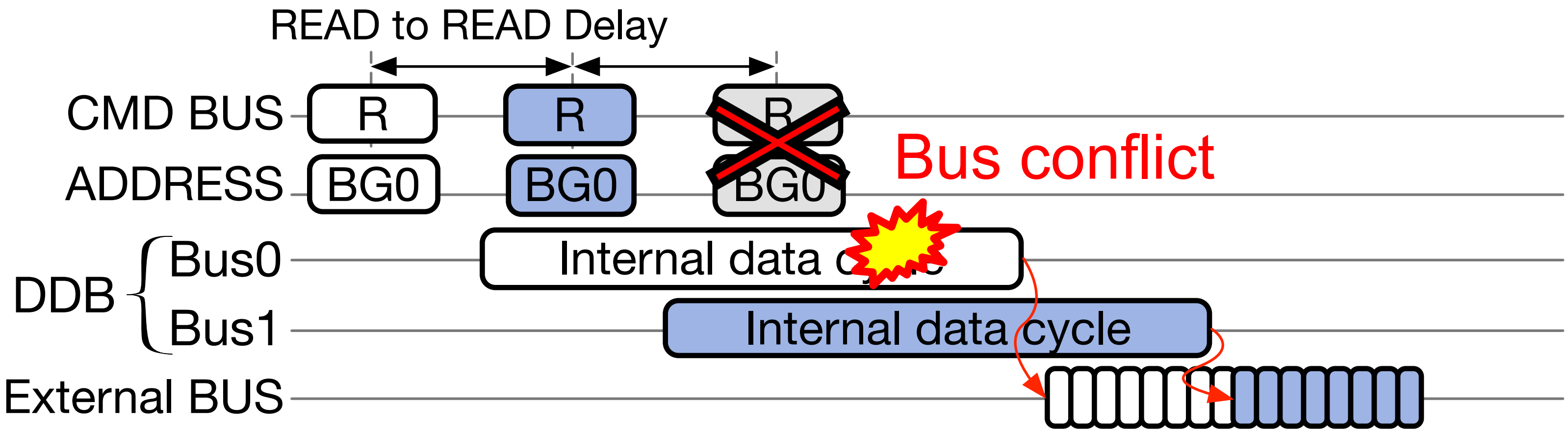
Larger Frequency Gap



Longer data cycle time

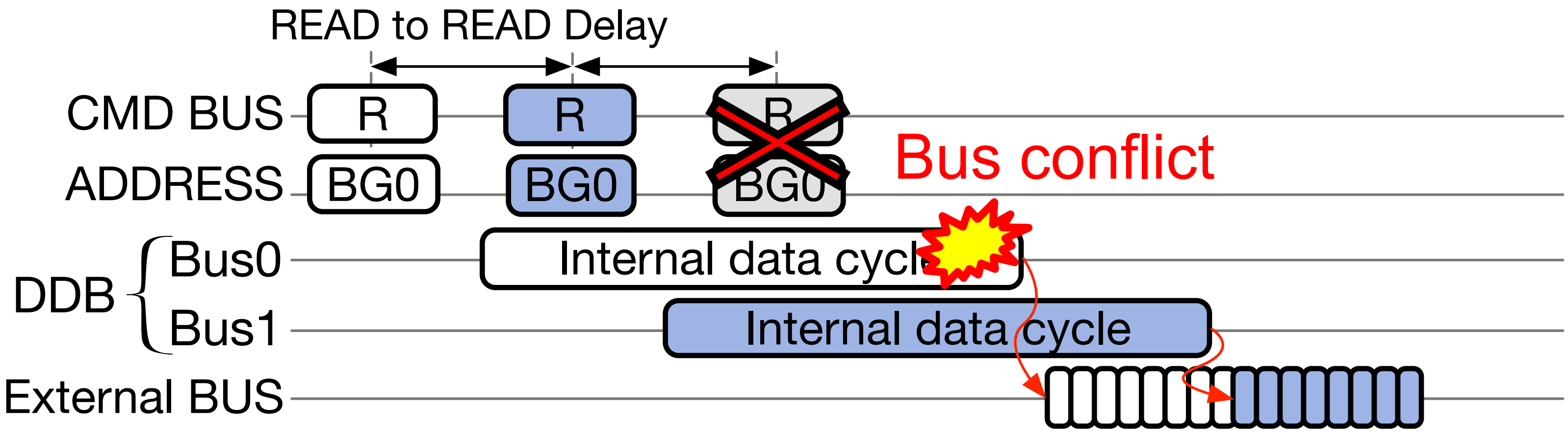
Large frequency gap between channel and DRAM core clock

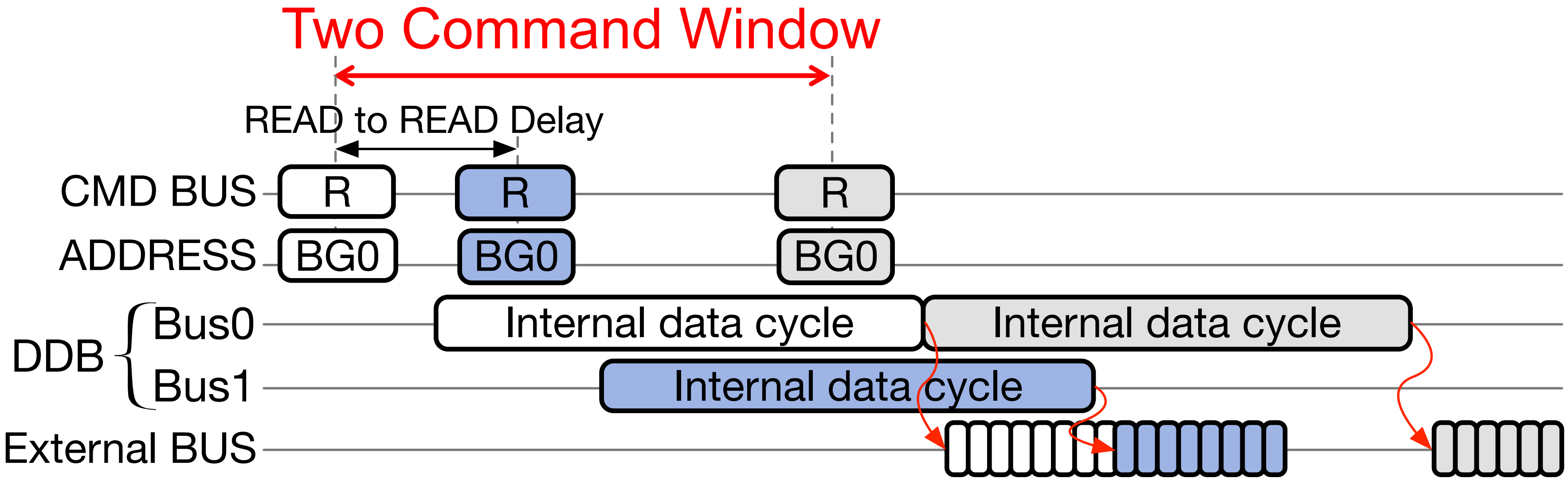
Larger Frequency Gap



Large frequency gap between channel and DRAM core clock

Command Window for Frequency Scaling





Evaluation

Evaluation Environment

DRAM system:

- DDR4 (16Banks, 4BG), 1.33GHz, 2 Channel x 1Rank
- Intel Skylake address mapping [1]

Simulators:

- CPU: SniperSim (Pin-based, OoO) [2]
- Memory system: USIMM (Trace based) [3]

Memory allocation:

- OS version: Linux 3.19.0-32-generic
- Memory fragmentation: 10%, 50% [4]

Workload:

- 8 (4 core SPEC2006 bench mixes)

[1] P. Pessl, D. Gruss, C. Maurice, and S. Mangard, "Reverse engineering intel DRAM addressing and exploitation," *CoRR*, vol. abs/1511.08756, 2015

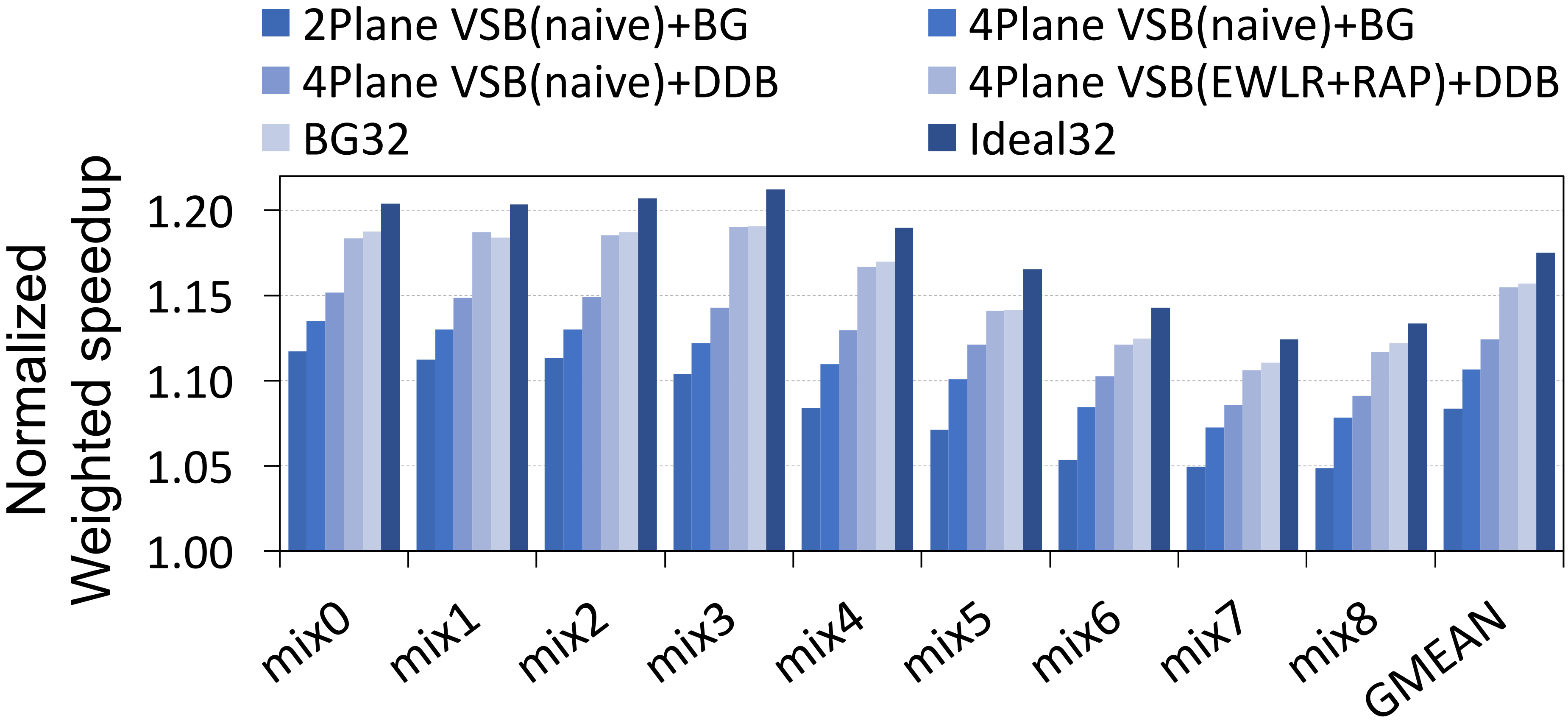
[2] M. Gorman and A. Whitcroft, "The what, the why and the where to of anti- fragmentation," in *Ottawa Linux Symposium*, vol. 1, pp. 369–384, Citeseer,2006.

[3] N. Chatterjee, R. Balasubramonian, M. Shevgoor, S. Pugsley, A. Udipi, A. Shafiee, K. Sudan, M. Awasthi, and Z. Chishti, "Usimm: the utah simulated memory module," *University of Utah, Tech. Rep*, 2012.

[4] Y. Kwon, H. Yu, S. Peter, C. J. Rossbach, and E. Witchel, "Coordinated and efficient huge page management with ingens," in *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, pp. 705–721, USENIX Association, 2016.

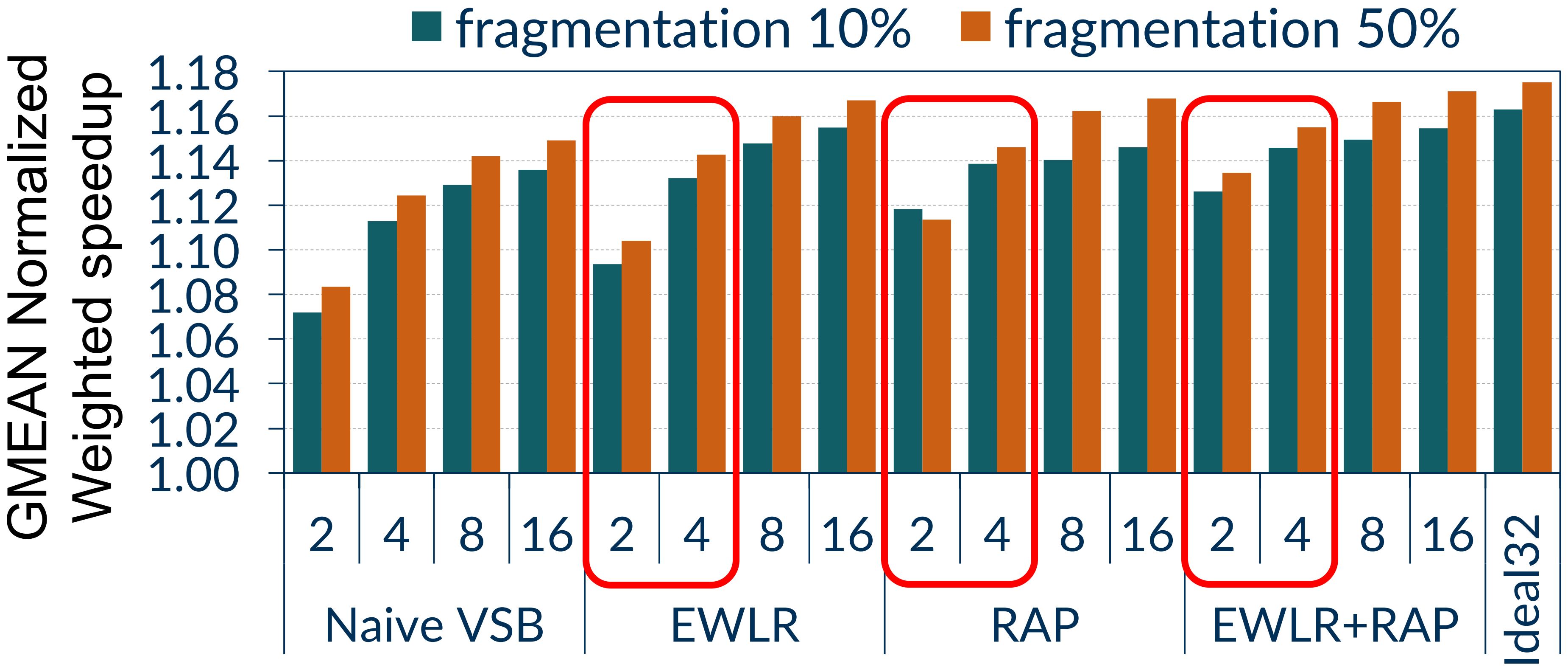
Overall Weighted Speedup

- Naive 2Plane, 4Plane VSB → 8%, 10%
- +DDB has small performance gain → 12%
- +(EWLR+RAP) → 16% (similar to BG32)



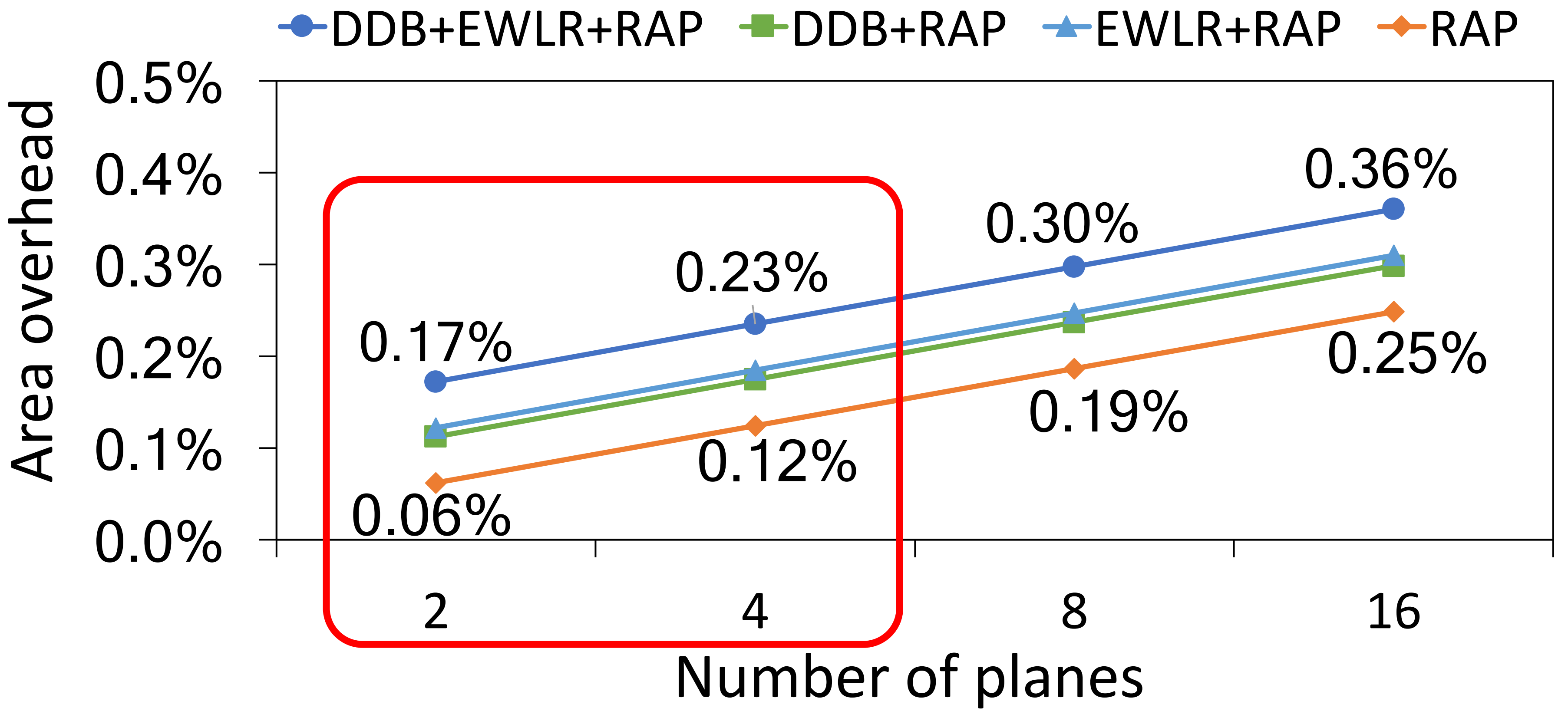
Performance Sensitivity

- Diminishing gain with increasing number of planes
- Small performance gain by EWLR+RAP
- Robust to high memory fragmentation



DRAM Area Overhead

2Plane VSB based ERUCA: 0.06 ~0.17%
4Plane VSB based ERUCA: 0.12 ~0.23%



Conclusion

Problem:

- Memory system parallelism restriction by the intra and inter bank resource conflicts

Goal:

- Avoid resource conflicts at near-zero cost

Observation:

- Combo DRAM
- Row address locality between sub-banks

Proposed schemes:

- **VSB:** Near-zero cost sub-banking scheme
- **EWLR/RAP:** Shared row address latch conflict avoidance using low address locality
- **DDB:** Low cost switchable data bus construct

ERUCA

Efficient DRAM **R**esource **U**tilization and Resource **C**onflict **A**voidance for Memory System Parallelism

Sangkug Lym, Heonjae Ha, Yongkee Kwon, Chun-kai Chang, Jungrae Kim, Mattan Erez

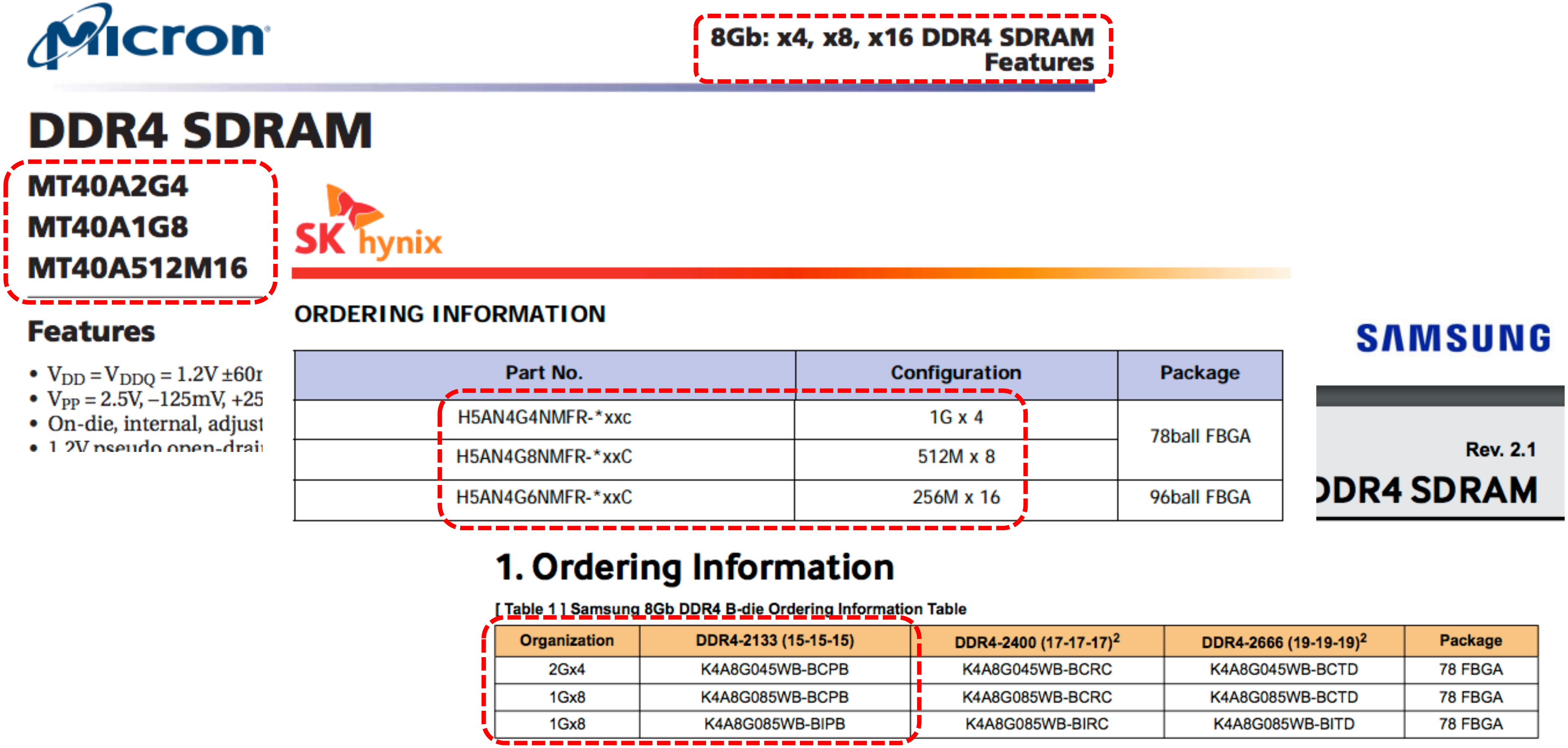
TEXAS The University of Texas at Austin

Stanford University

Microsoft

Appendix

Combo DRAM Architecture



Micron

8Gb: x4, x8, x16 DDR4 SDRAM
Features

DDR4 SDRAM

MT40A2G4
MT40A1G8
MT40A512M16

Features

- $V_{DD} = V_{DDQ} =$ 1.2V ±60r
- $V_{PP} =$ 2.5V, –125mV, +25
- On-die, internal, adjust
- 1.2V pseudo open-drai

SK hynix

ORDERING INFORMATION

Part No.	Configuration	Package
H5AN4G4NMFR-*xxc	1G x 4	78ball FBGA
H5AN4G8NMFR-*xxC	512M x 8	
H5AN4G6NMFR-*xxC	256M x 16	96ball FBGA

SAMSUNG

Rev. 2.1

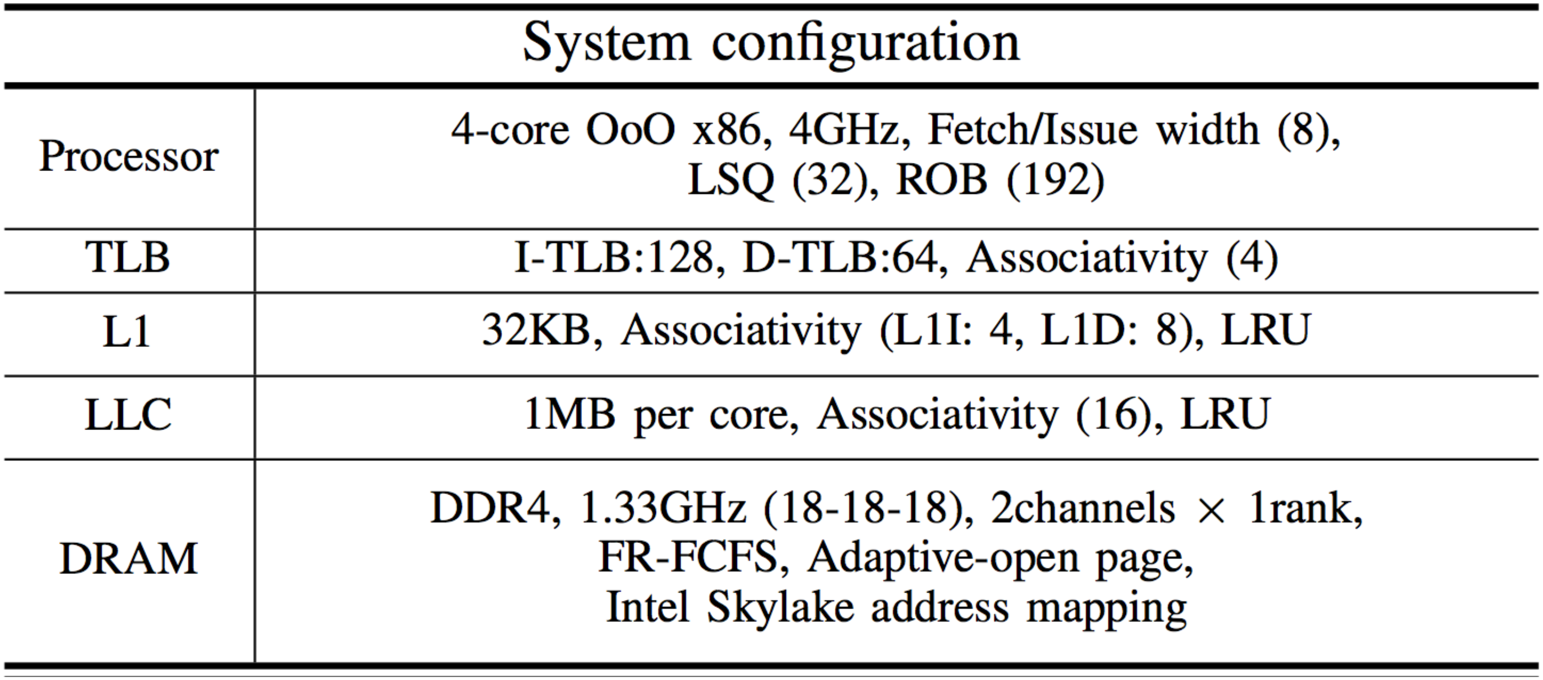
DDR4 SDRAM

1. Ordering Information

[Table 1] Samsung 8Gb DDR4 B-die Ordering Information Table

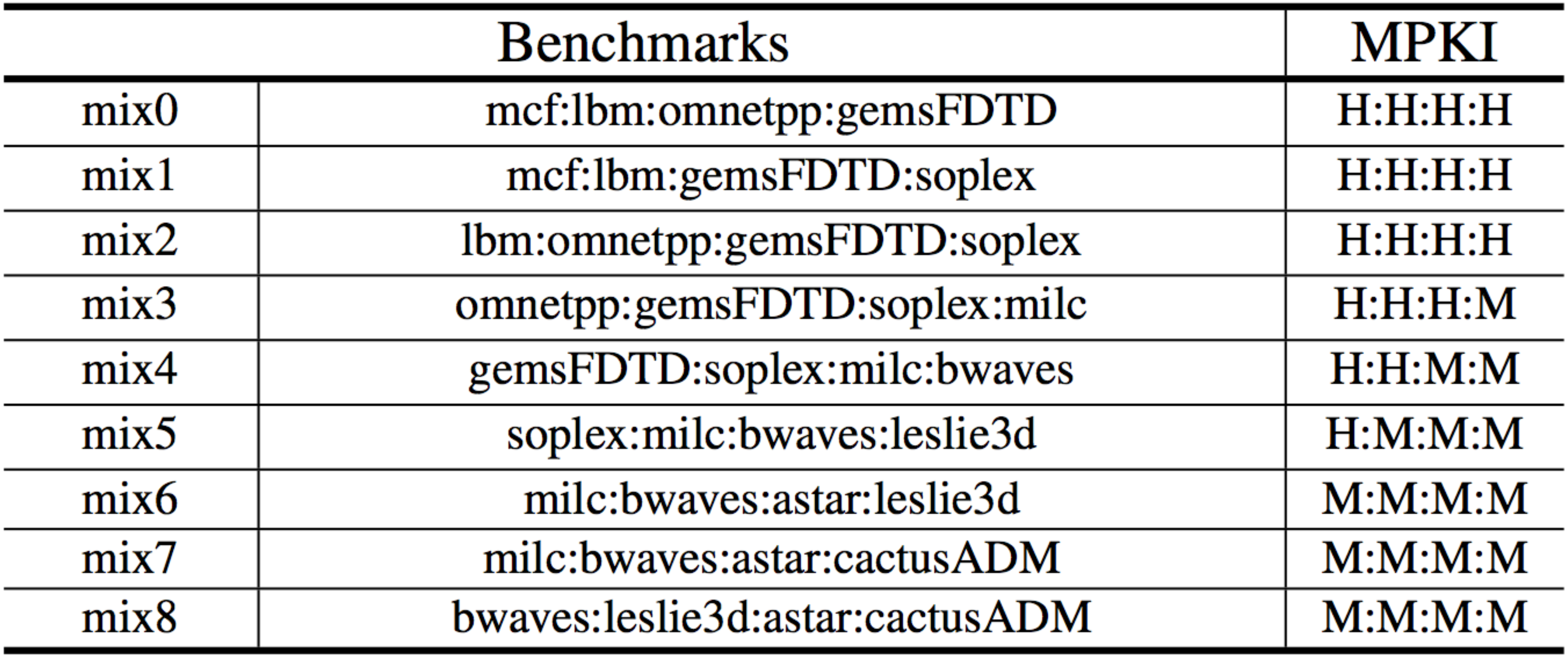
Organization	DDR4-2133 (15-15-15)	DDR4-2400 (17-17-17)^2	DDR4-2666 (19-19-19)^2	Package
2Gx4	K4A8G045WB-BCPB	K4A8G045WB-BCRC	K4A8G045WB-BCTD	78 FBGA
1Gx8	K4A8G085WB-BCPB	K4A8G085WB-BCRC	K4A8G085WB-BCTD	78 FBGA
1Gx8	K4A8G085WB-BIPB	K4A8G085WB-BIRC	K4A8G085WB-BITD	78 FBGA

Evaluation Environment



System configuration	
Processor	4-core OoO x86, 4GHz, Fetch/Issue width (8), LSQ (32), ROB (192)
TLB	I-TLB:128, D-TLB:64, Associativity (4)
L1	32KB, Associativity (L1I: 4, L1D: 8), LRU
LLC	1MB per core, Associativity (16), LRU
DRAM	DDR4, 1.33GHz (18-18-18), 2channels × 1rank, FR-FCFS, Adaptive-open page, Intel Skylake address mapping

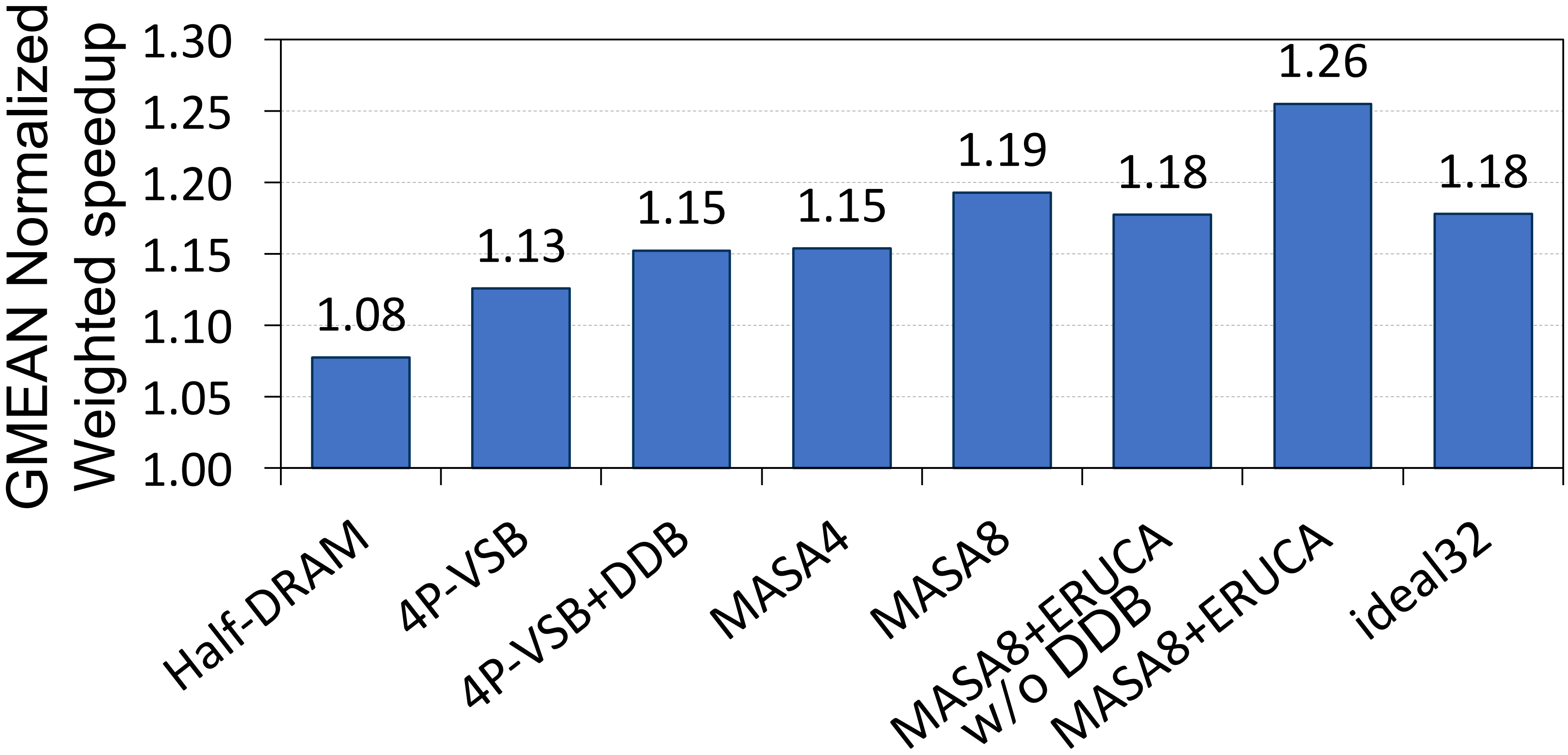
Benchmark Mixes



Benchmarks		MPKI
mix0	mcf:lbm:omnetpp:gemsFDTD	H:H:H:H
mix1	mcf:lbm:gemsFDTD:soplex	H:H:H:H
mix2	lbm:omnetpp:gemsFDTD:soplex	H:H:H:H
mix3	omnetpp:gemsFDTD:soplex:milc	H:H:H:M
mix4	gemsFDTD:soplex:milc:bwaves	H:H:M:M
mix5	soplex:milc:bwaves:leslie3d	H:M:M:M
mix6	milc:bwaves:astar:leslie3d	M:M:M:M
mix7	milc:bwaves:astar:cactusADM	M:M:M:M
mix8	bwaves:leslie3d:astar:cactusADM	M:M:M:M

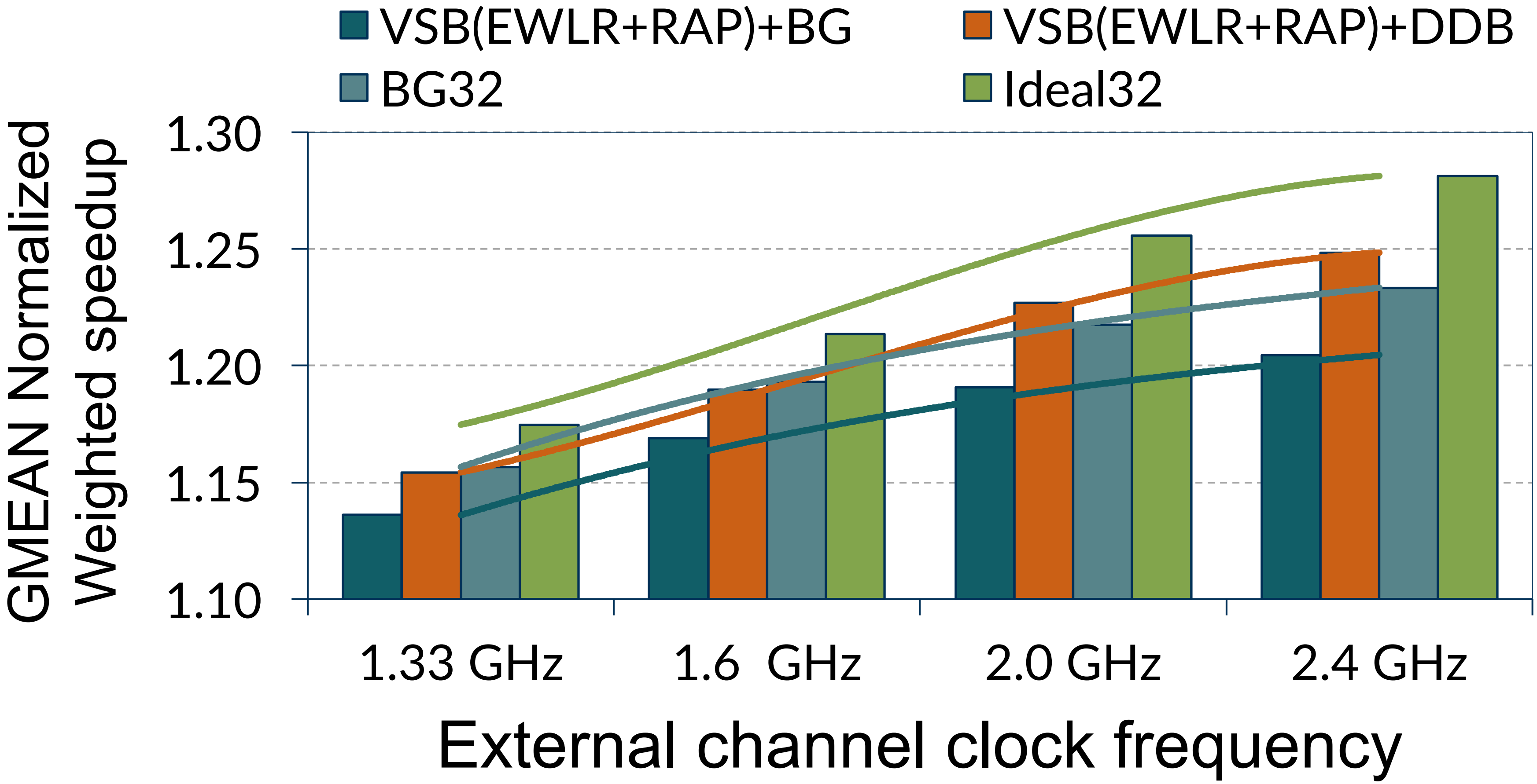
Comparison to the Prior Work

ERUCA has significant synergy with SALP
→ Resolve in-DRAM bus conflicts

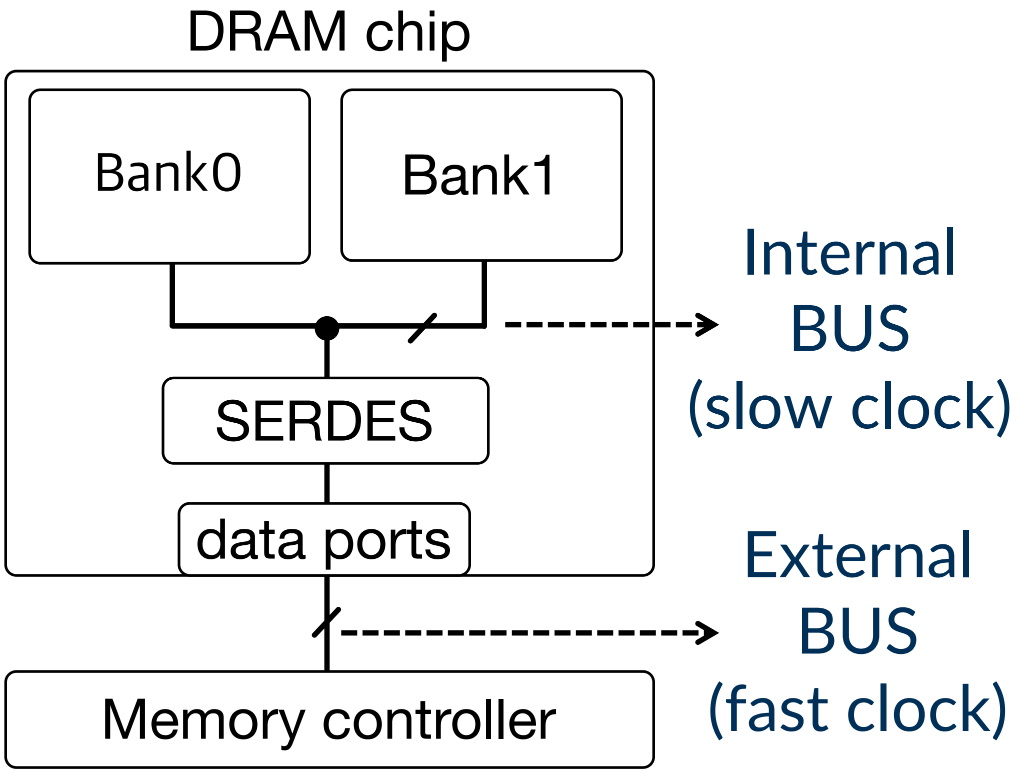


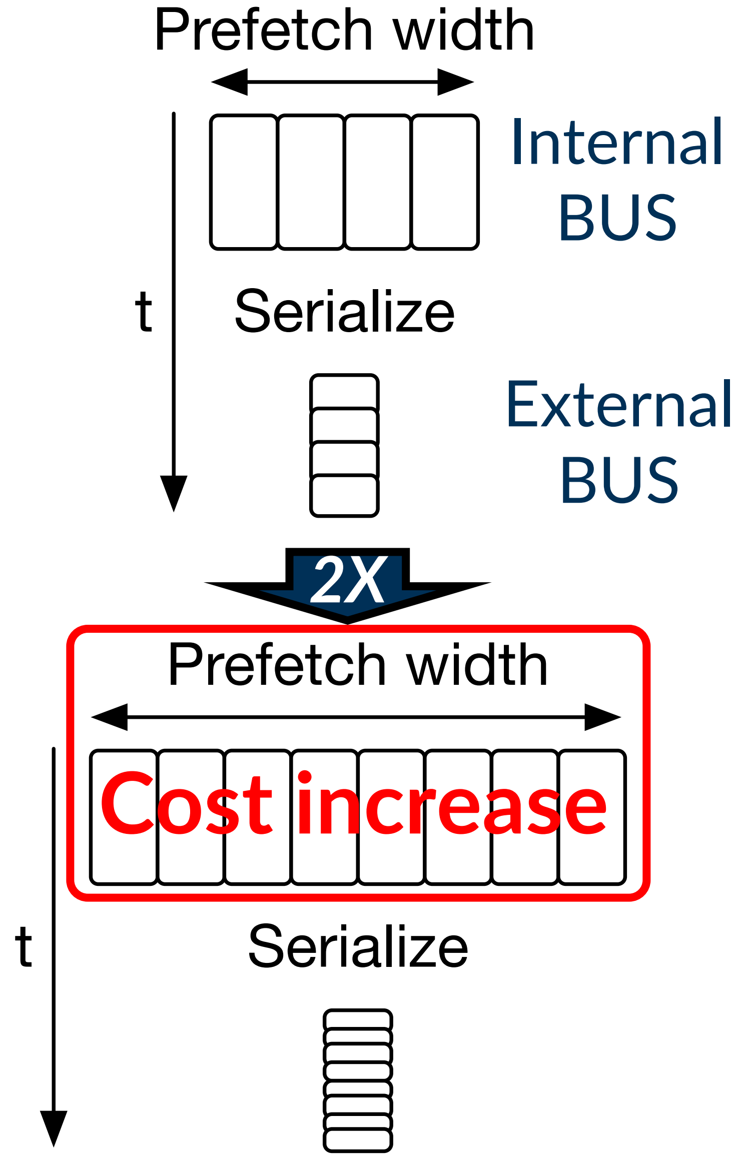
DDB: Frequency Scaling

- Bank grouping saturates by increasing bandwidth
- DDB scales with linear performance gain



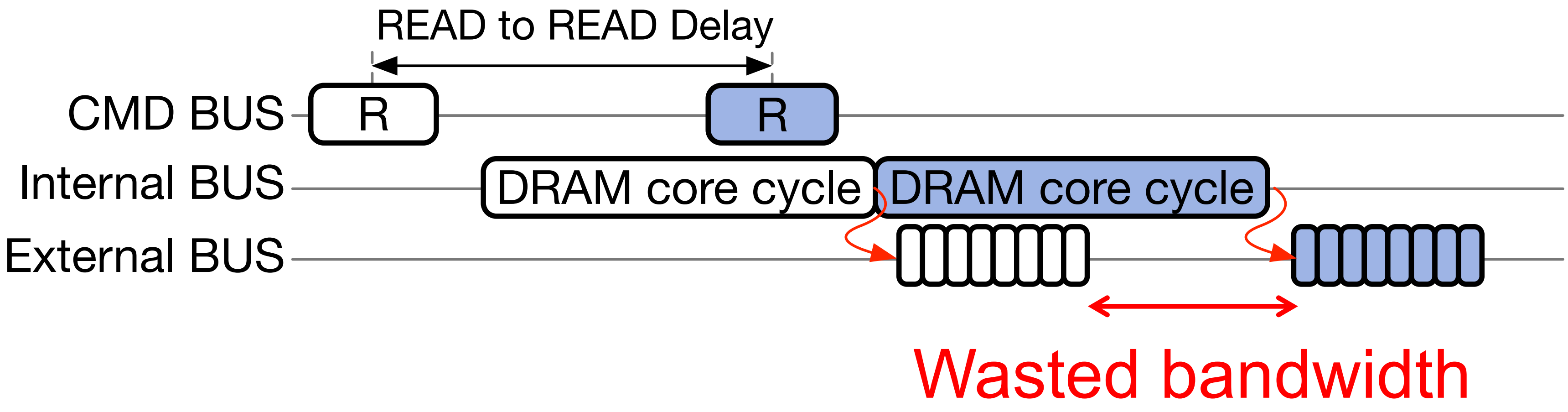
Channel Bandwidth Scaling





No prefetch width increase in DDR4

	DDR	DDR2	DDR3	DDR4
Internal clock (MHz)	200	200	200	200
External clock (MHz)	200	400	800	1600
Prefetch width	2X	4X	8X →	8X



Wasted bandwidth