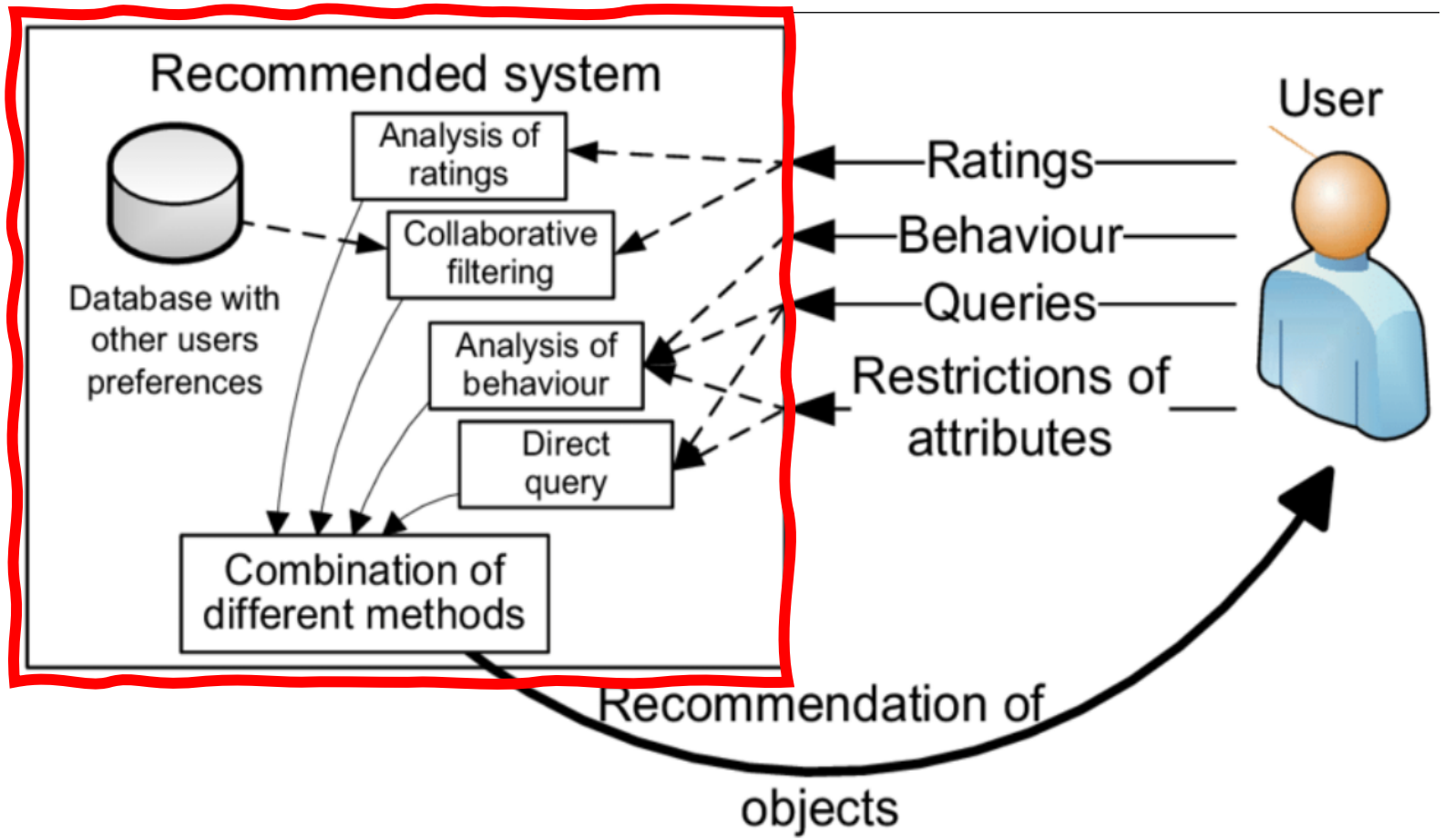
PIM-SUM: PopCount-based Fast and Reliable In-memory Summation Scheme

Fan Li, Ruizhi Zhu, Huize Li, Di Wu, Xin Xin

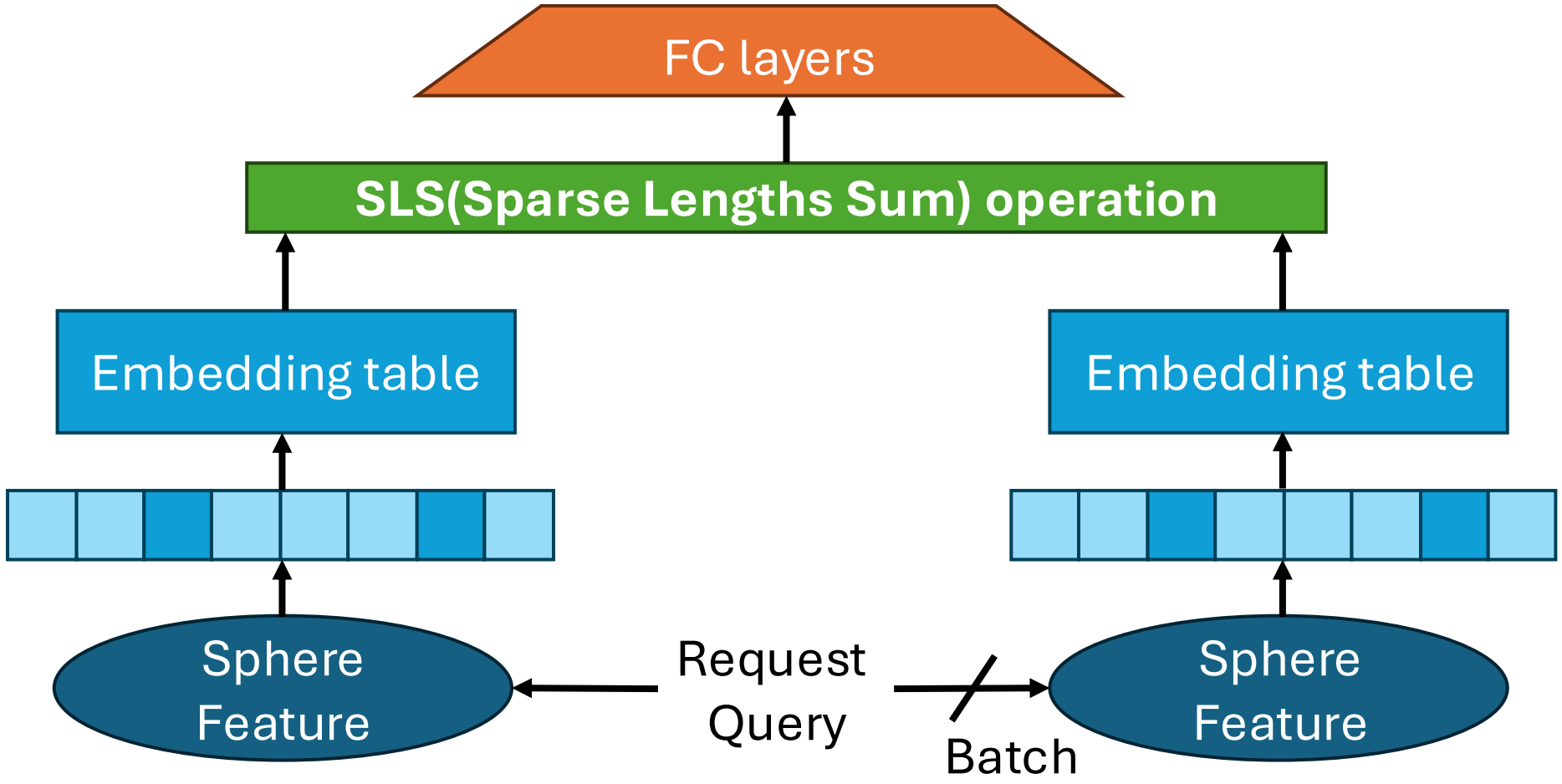
College of Engineering and Computer Science

University of Central Florida

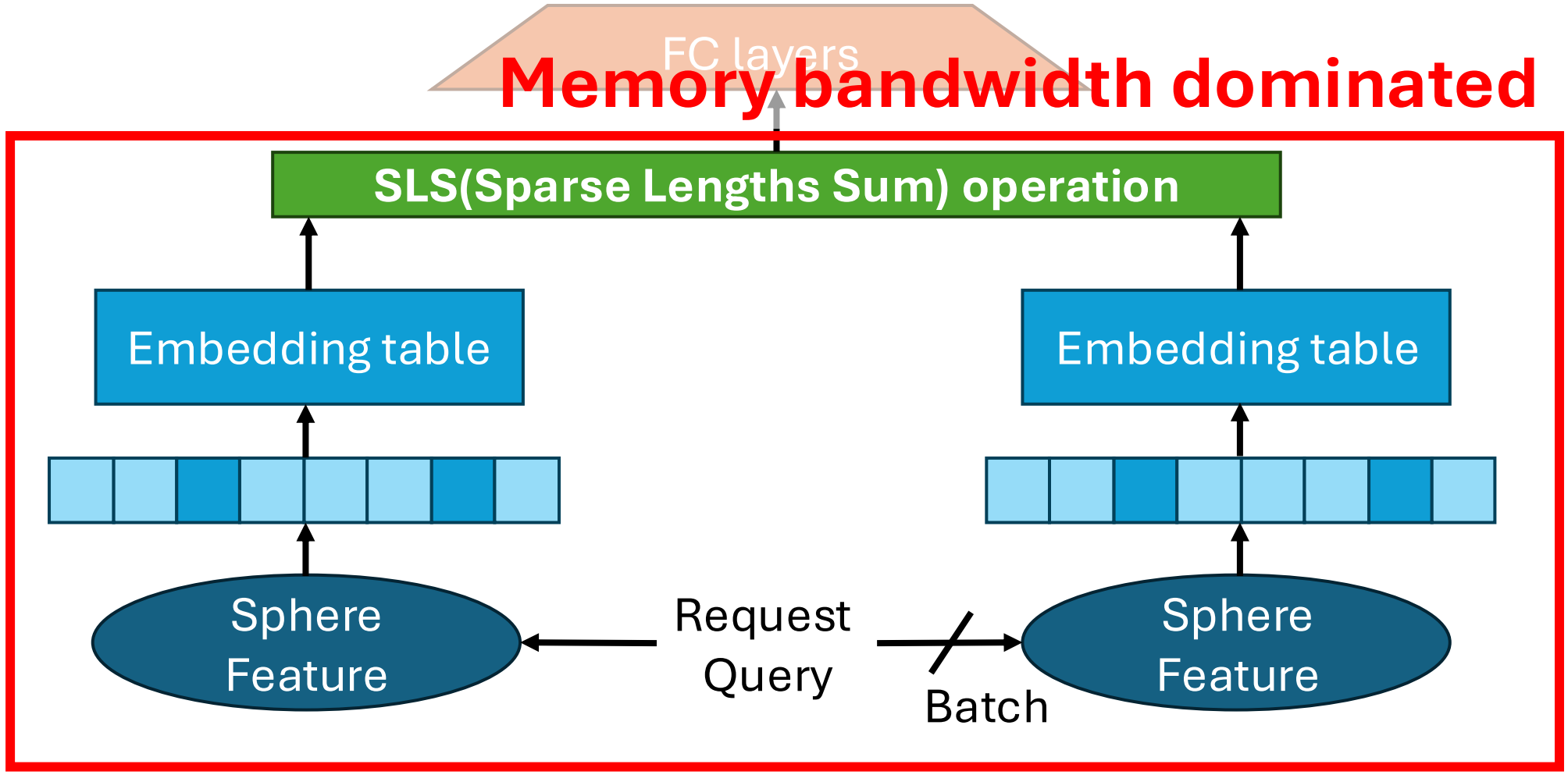
Motivation--Recommendation System



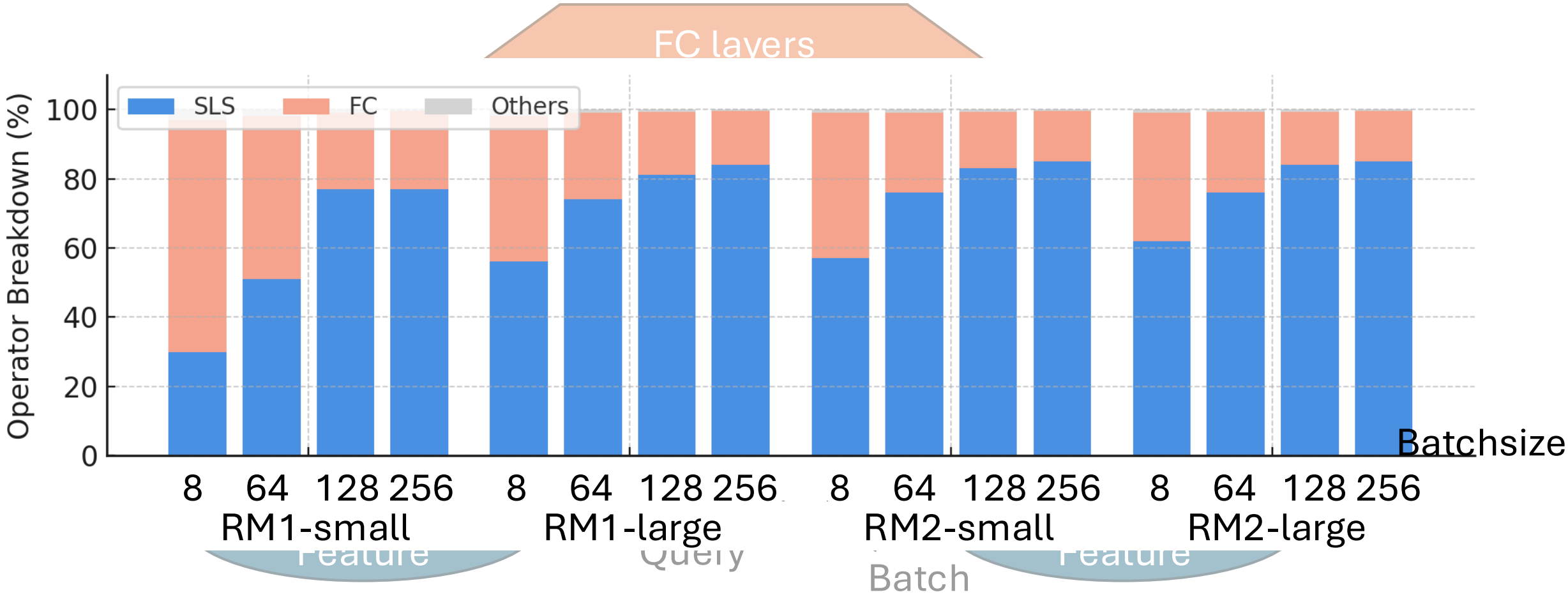
Bottleneck of Recommendation Model



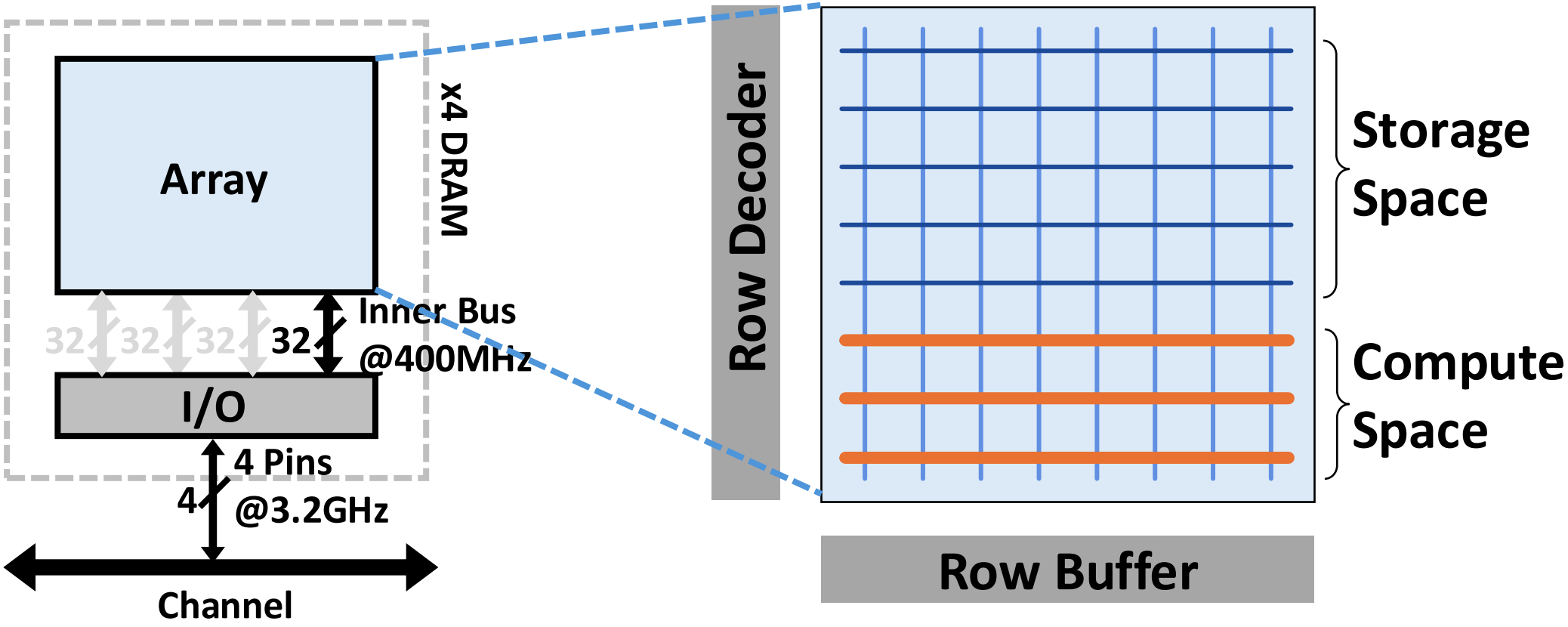
Bottleneck of Recommendation Model



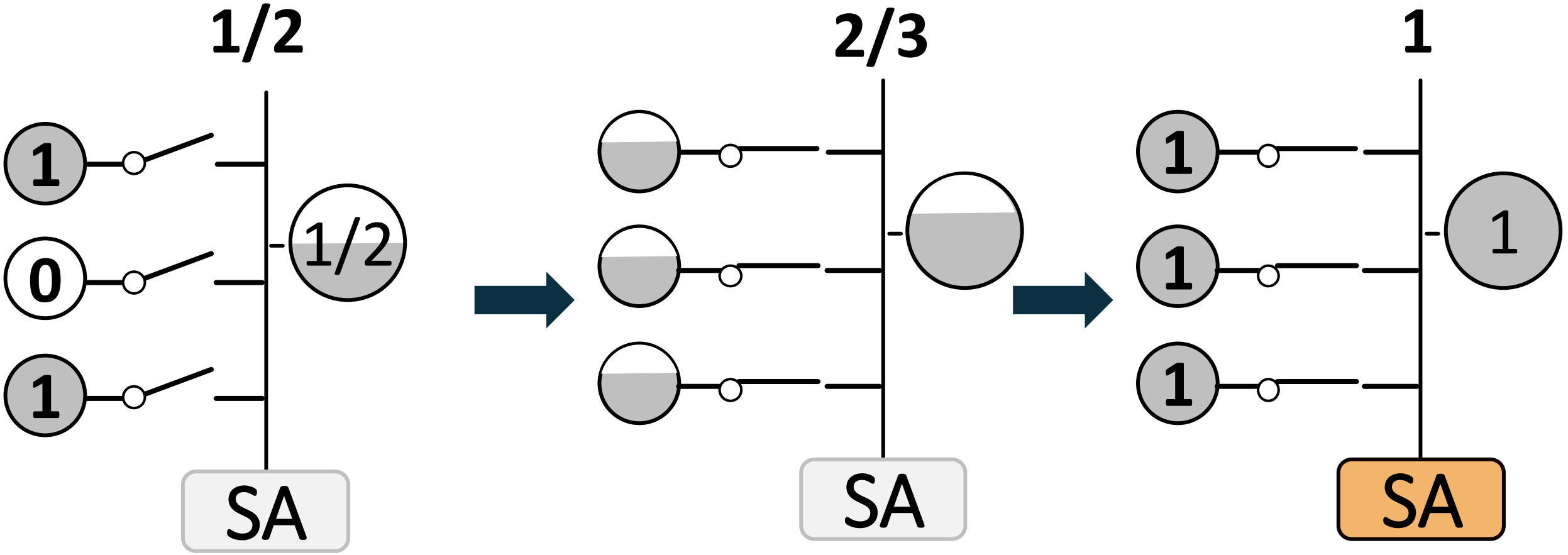
Bottleneck of Recommendation Model



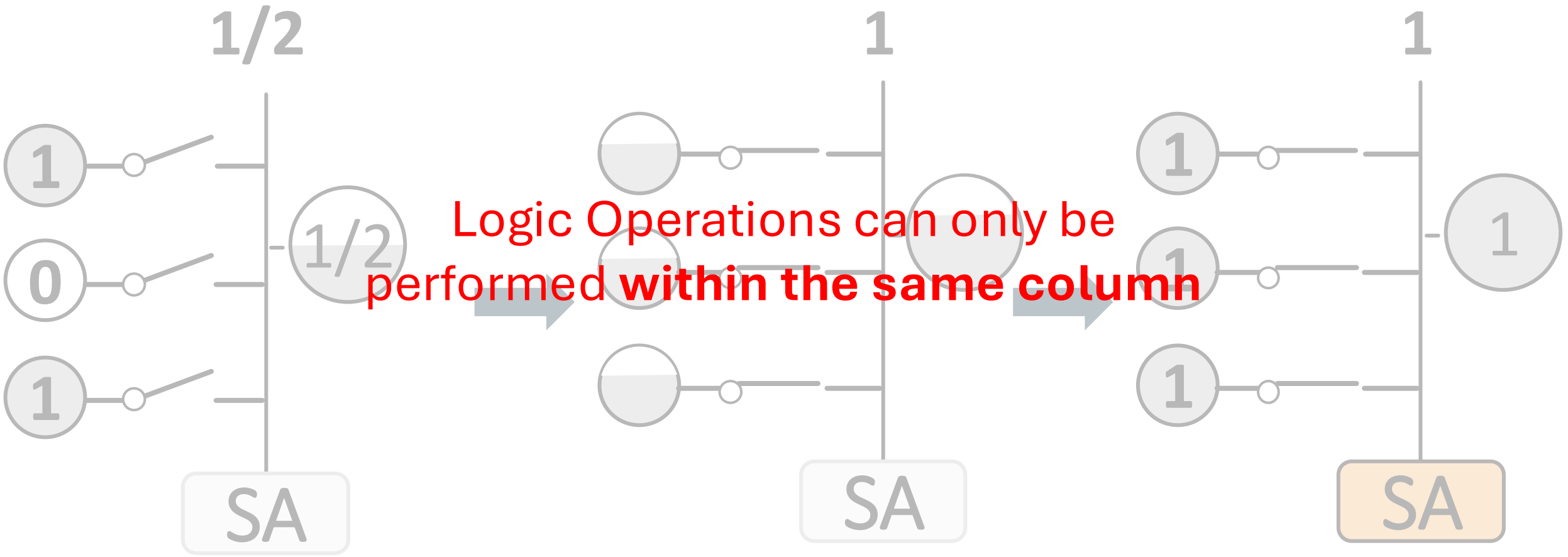
Background—Processing in Memory(PIM)



Background—Processing in Memory(PIM)

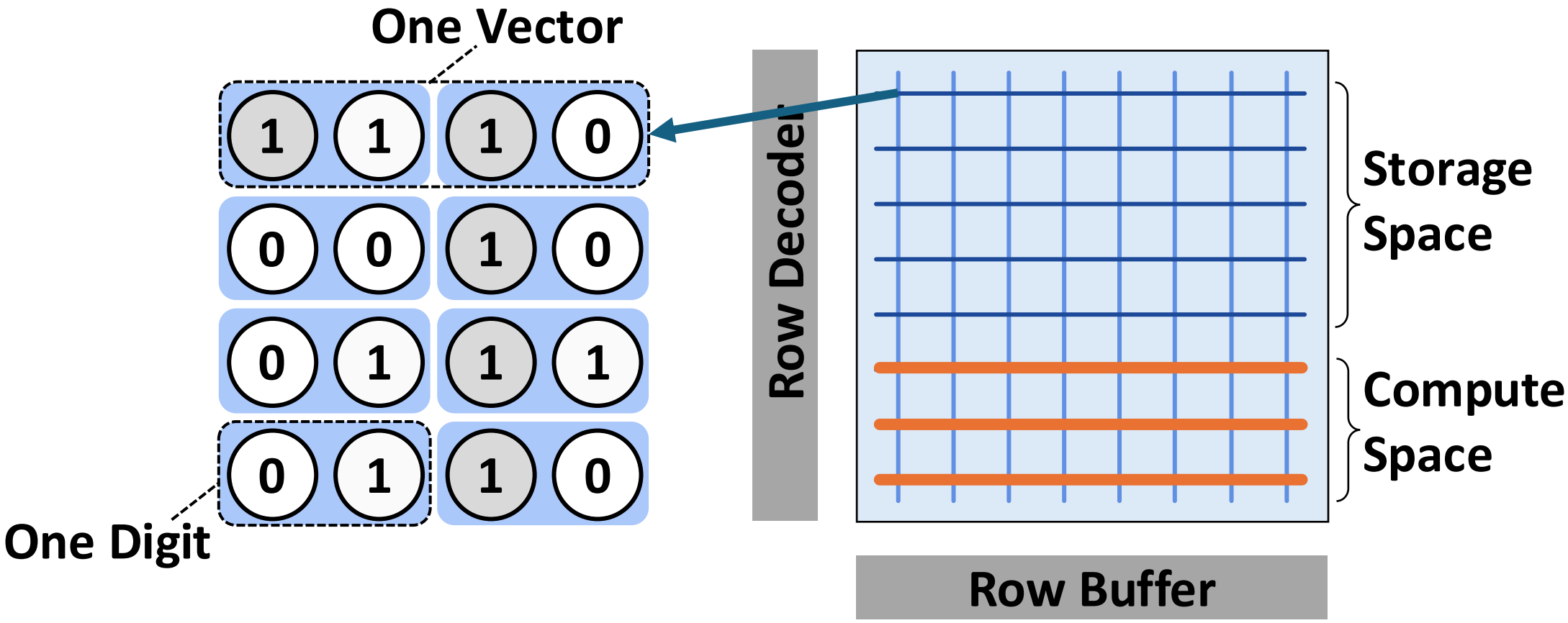


Background—Processing in Memory(PIM)

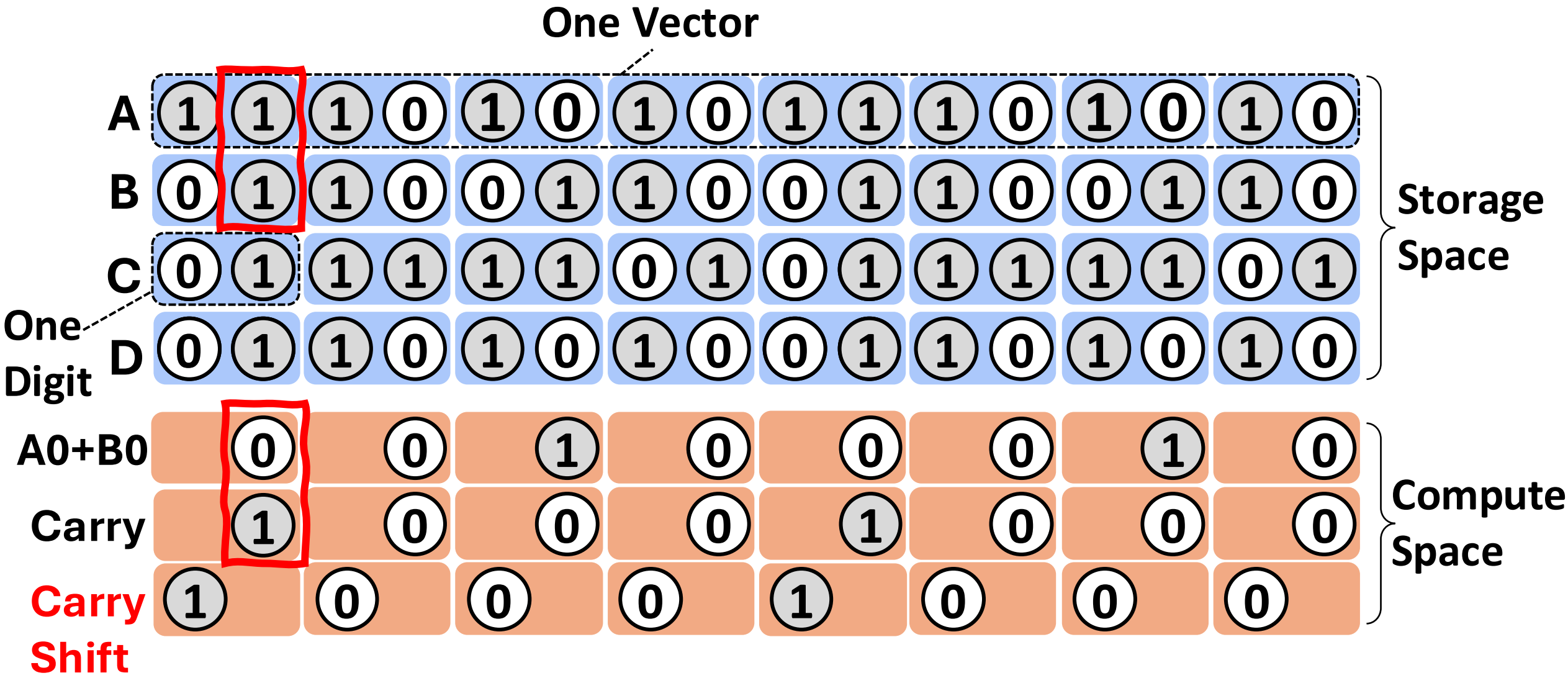


Logic Operations can only be performed **within the same column**

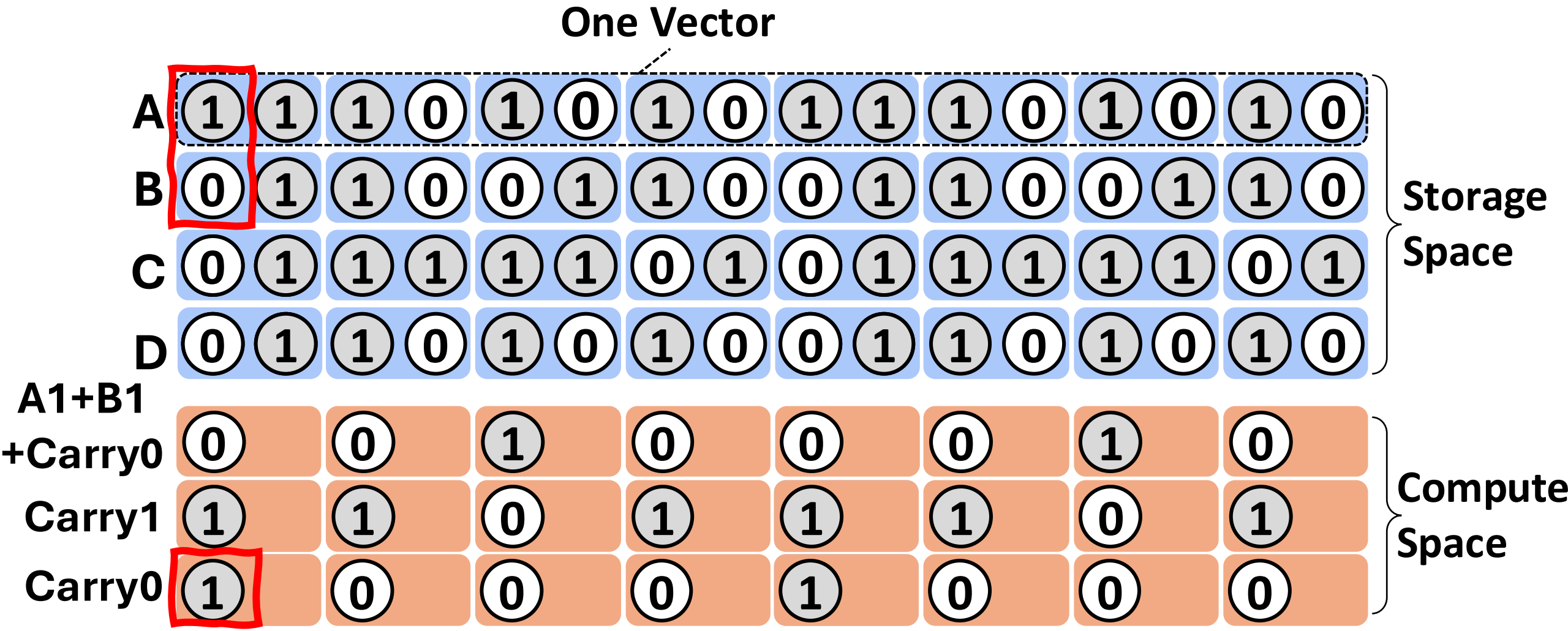
Mapping


One Vector
1 1 1 0
0 0 1 0
0 1 1 1
0 1 1 0
One Digit
Row Decoder
Storage Space
Compute Space
Row Buffer

Baseline


One Vector
A 1 1 1 0 1 0 1 0 1 1 1 0 1 0 1 0
B 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0
C 0 1 1 1 1 1 0 1 0 1 1 1 1 1 0 1
D 0 1 1 0 1 0 1 0 0 1 1 0 1 0 1 0
One Digit
Storage Space
A0+B0 0 0 1 0 0 0 1 0
Carry 1 0 0 0 1 0 0 0
Carry 1 0 0 0 1 0 0 0
Shift
Compute Space

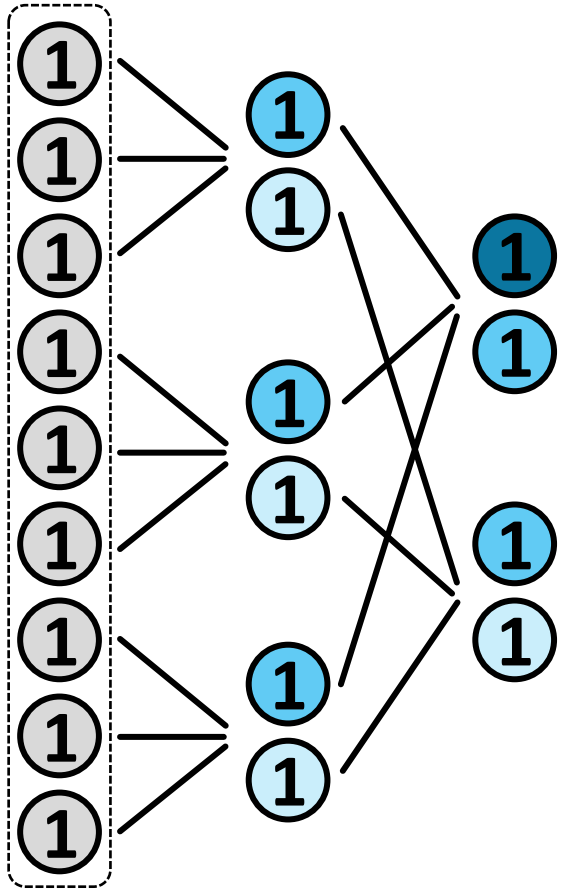
Baseline


One Vector
A 1 1 1 0 1 0 1 0 1 1 1 0 1 0 1 0
B 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0
C 0 1 1 1 1 1 0 1 0 1 1 1 1 1 0 1
D 0 1 1 0 1 0 1 0 0 1 1 0 1 0 1 0
Storage Space
A1+B1 0 0 1 0 0 0 1 0
+Carry0
Carry1 1 1 0 1 1 1 0 1
Carry0 1 0 0 0 1 0 0 0
Compute Space

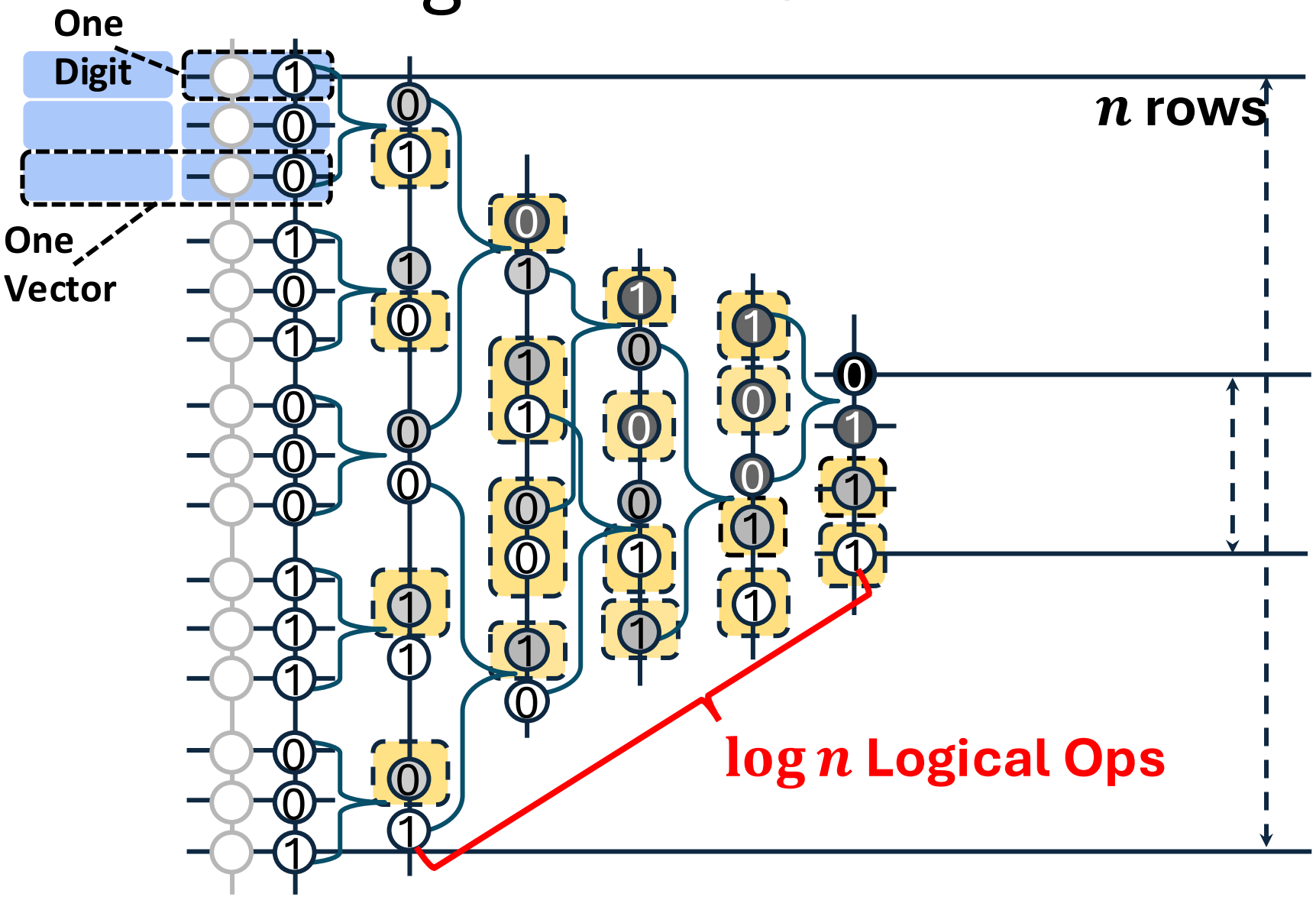
Our Design--PIMSUM

Steps:

1. Aggregate 3 rows in the same level into 2 rows on the next level.
2. Pass unaggregated rows into future process



Our Design--PIMSUM



Waiting for next stage

1 MSb

1

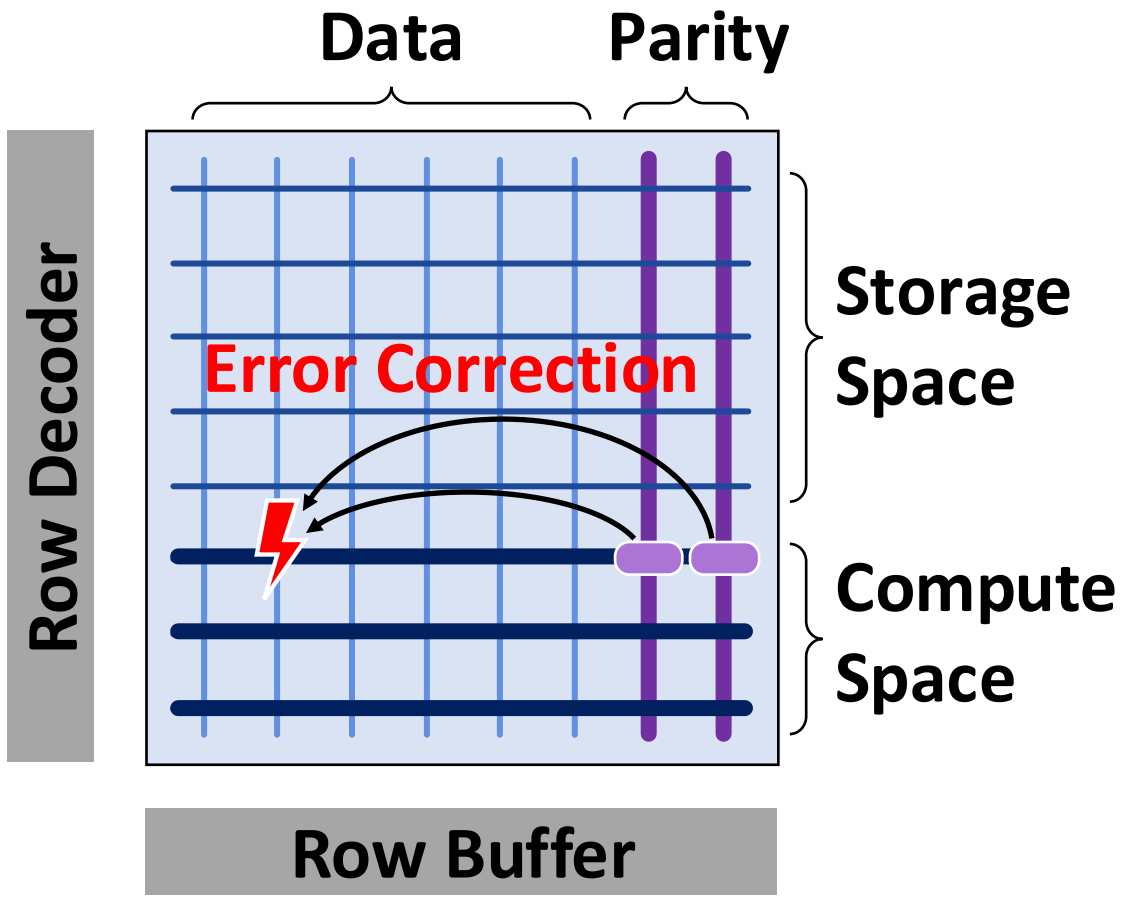
1

1 LSb

Reliability--PIMSUM

$$\begin{cases} p_{a0} = a_0 + a_1 + a_2 \\ p_{a1} = a_0 + 2a_1 + 3a_2 \end{cases}$$

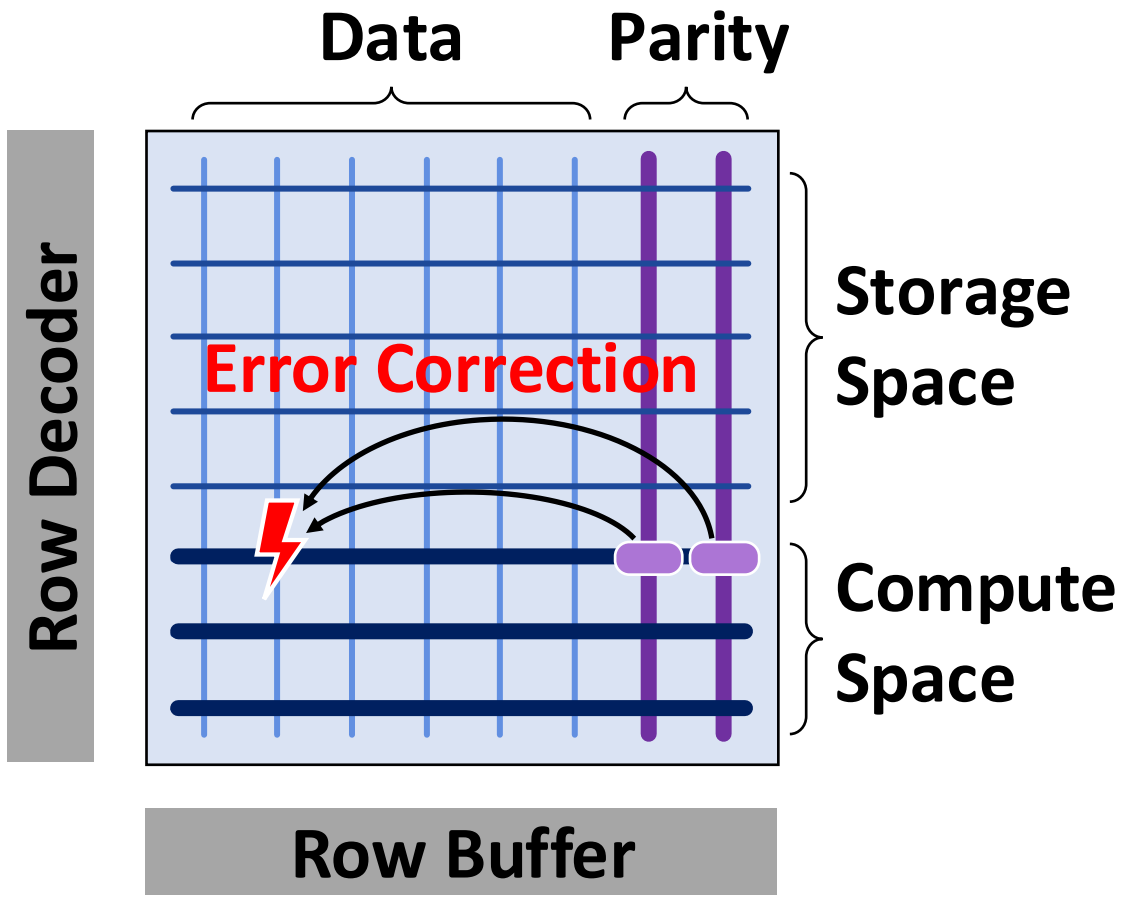
$$\begin{cases} p_{a0} = a_0 + (a_1 + e) + a_2 \\ p_{a1} = a_0 + 2(a_1 + e) + 3a_2 \end{cases}$$



Reliability--PIMSUM

$$\begin{cases} p_{a0} = a_0 + a_1 + a_2 \\ p_{a1} = a_0 + 2a_1 + 3a_2 \end{cases}$$

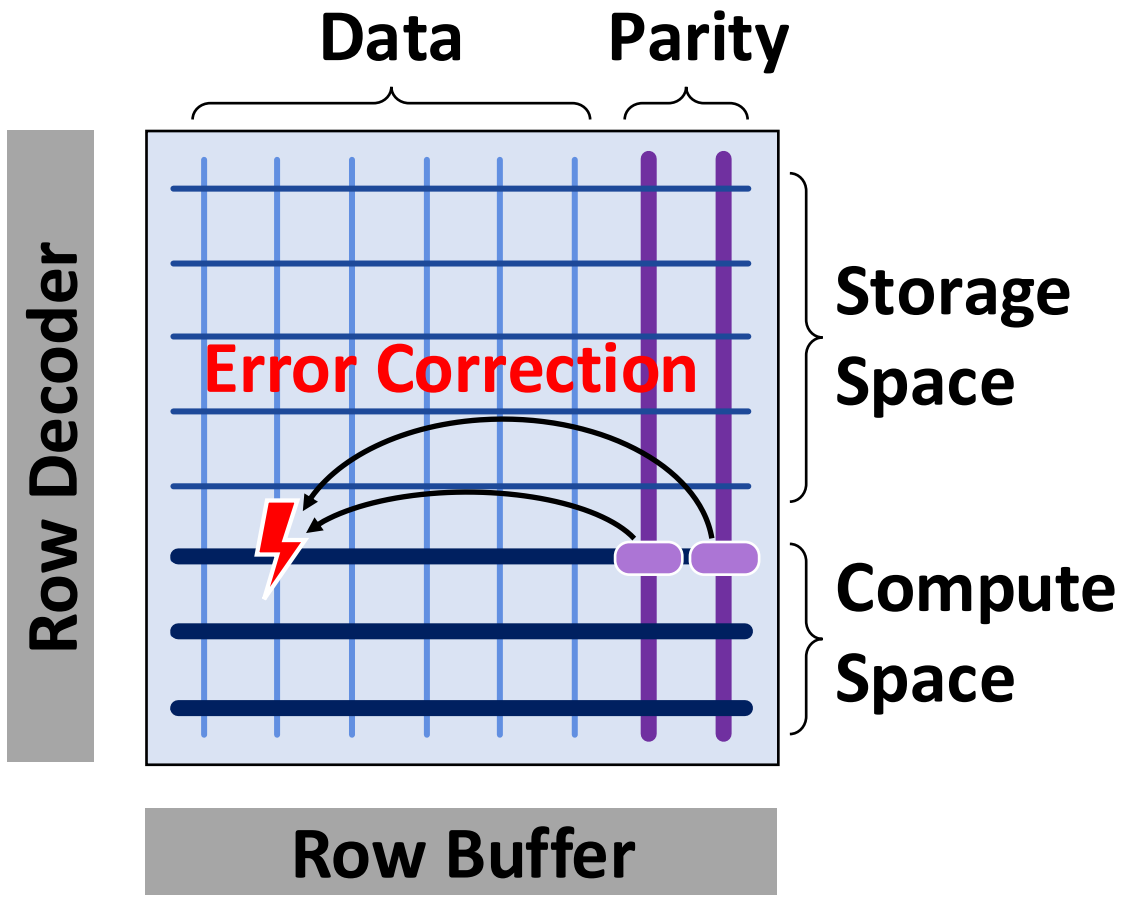
$$\begin{cases} p_{a0} = a_0 + (a_1 + e) + a_2 \\ p_{a1} = a_0 + 2a_1 + 2e + 3a_2 \end{cases}$$



Reliability--PIMSUM

$$e_{ind} = \frac{p_{a1}{}' - p_{a1}}{p_{a0}{}' - p_{a0}}$$

$$e = p_{a0}{}' - p_{a0}$$

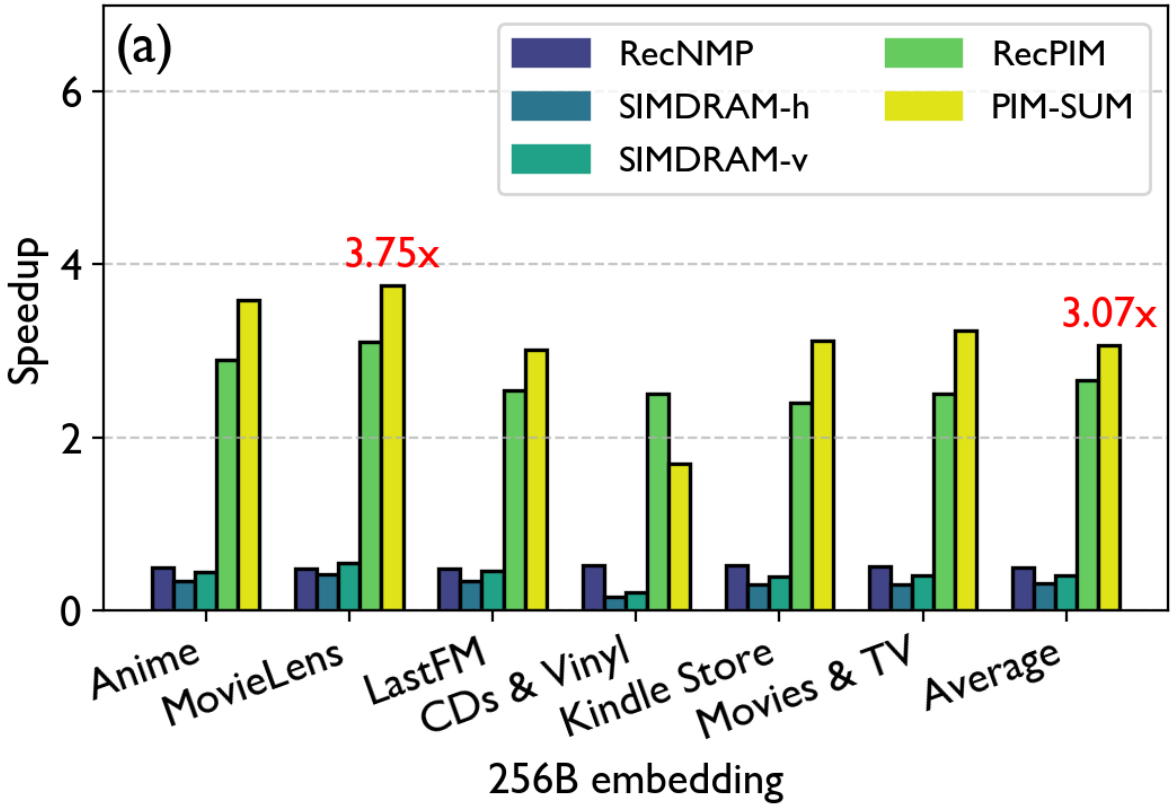


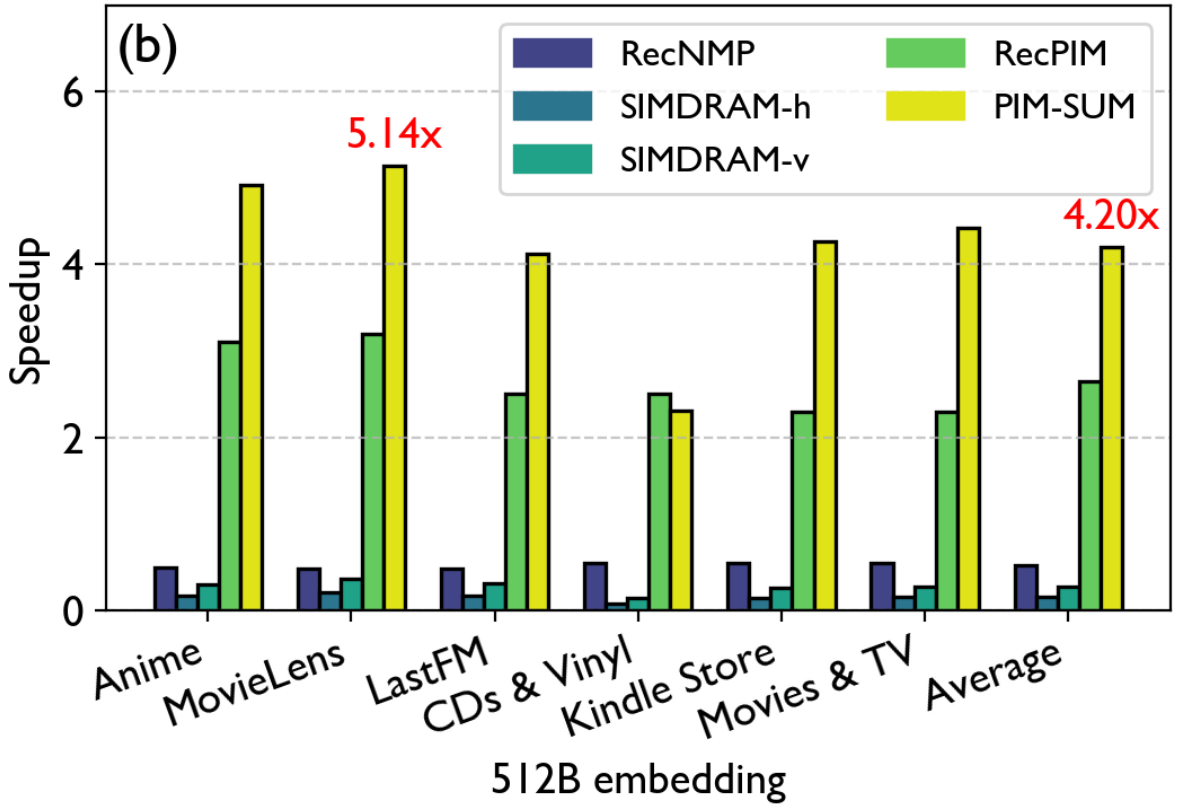
Summary--PIMSUM

1. Eliminates all logical operations that do not reduce the number of rows.
2. Performs accumulation without requiring carry shifts.
3. Guarantees computational correctness through ECC-based reliability support.

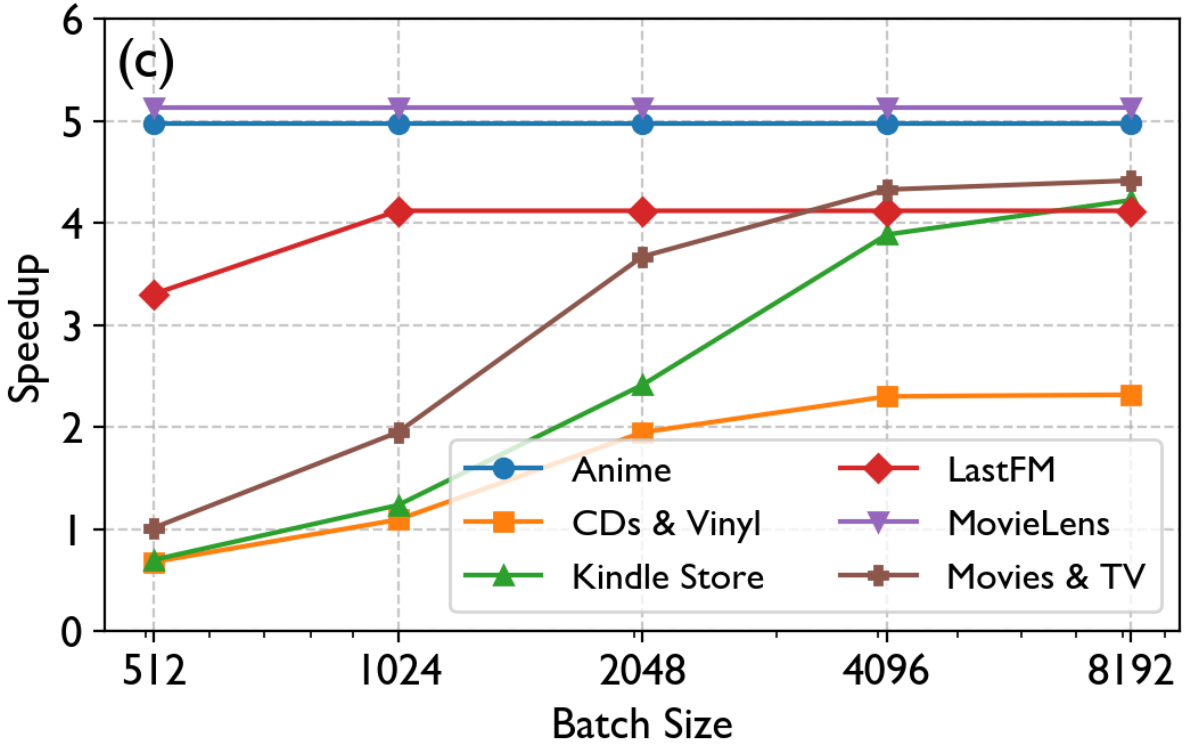
Mapping	Carry-shift free	Read efficiency
Horizontal		✓
Vertical	✓	
PIM-SUM	✓	✓

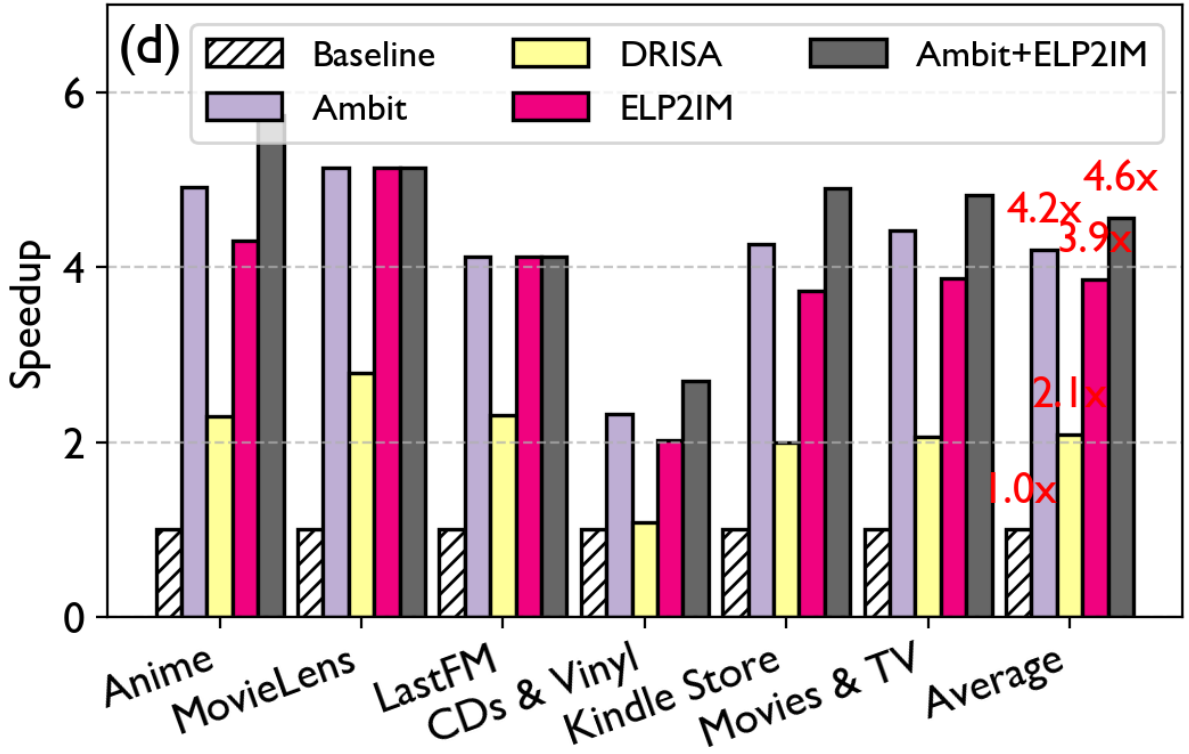
Performance--PIMSUM



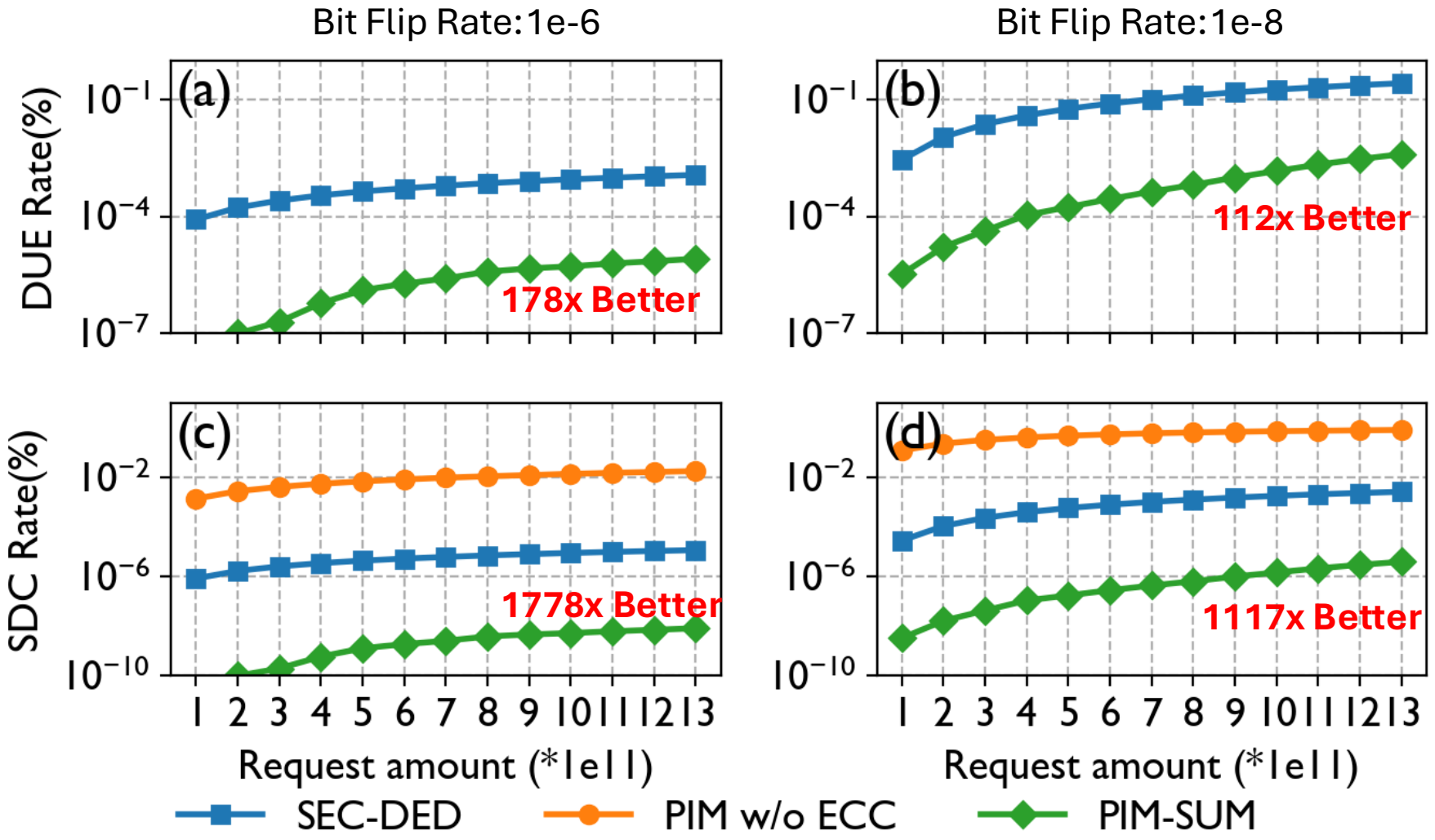


Performance--PIMSUM





Reliability--PIMSUM

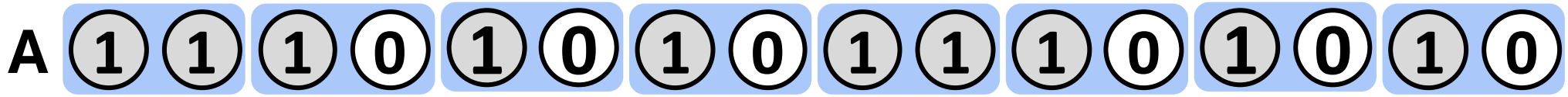


Thank you for listening
Q&A

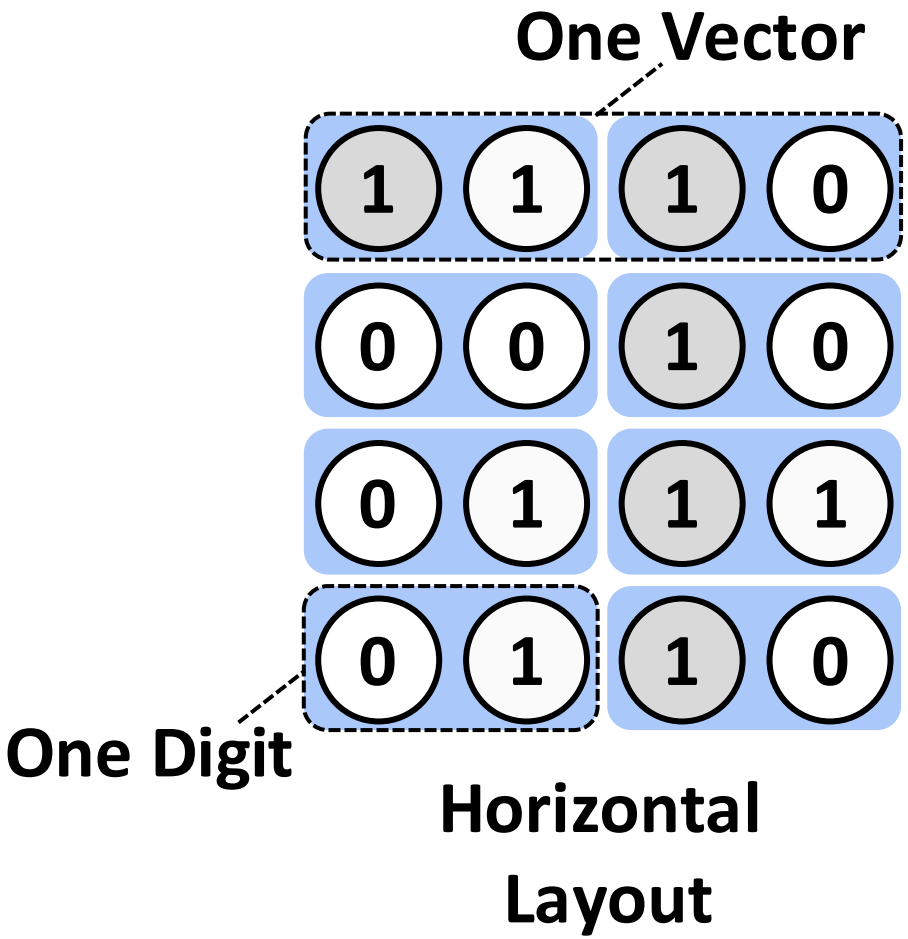
Summary--Horizontal Layout

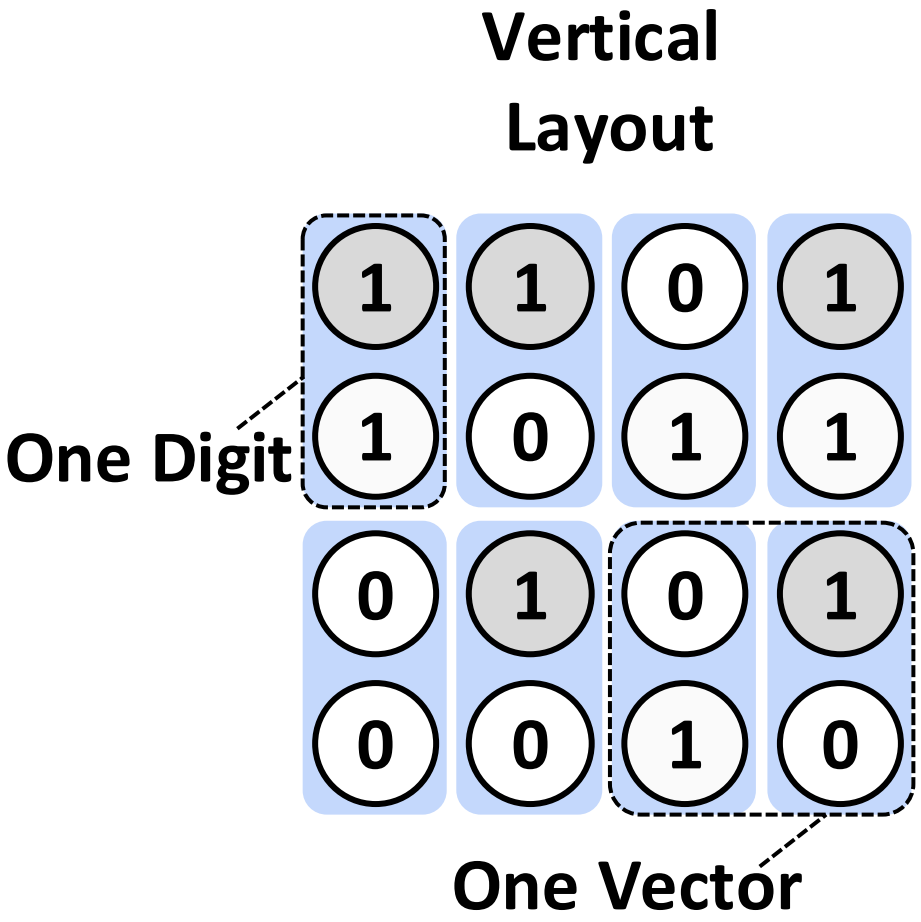
- Digit width: **w(=2)**
- Embedding vector element count: **n(=8)**
- Aggregation scale: **k(=4)**

- Cost: **w**(carry shift) * **k** (row summation) + **1** (mem_read)

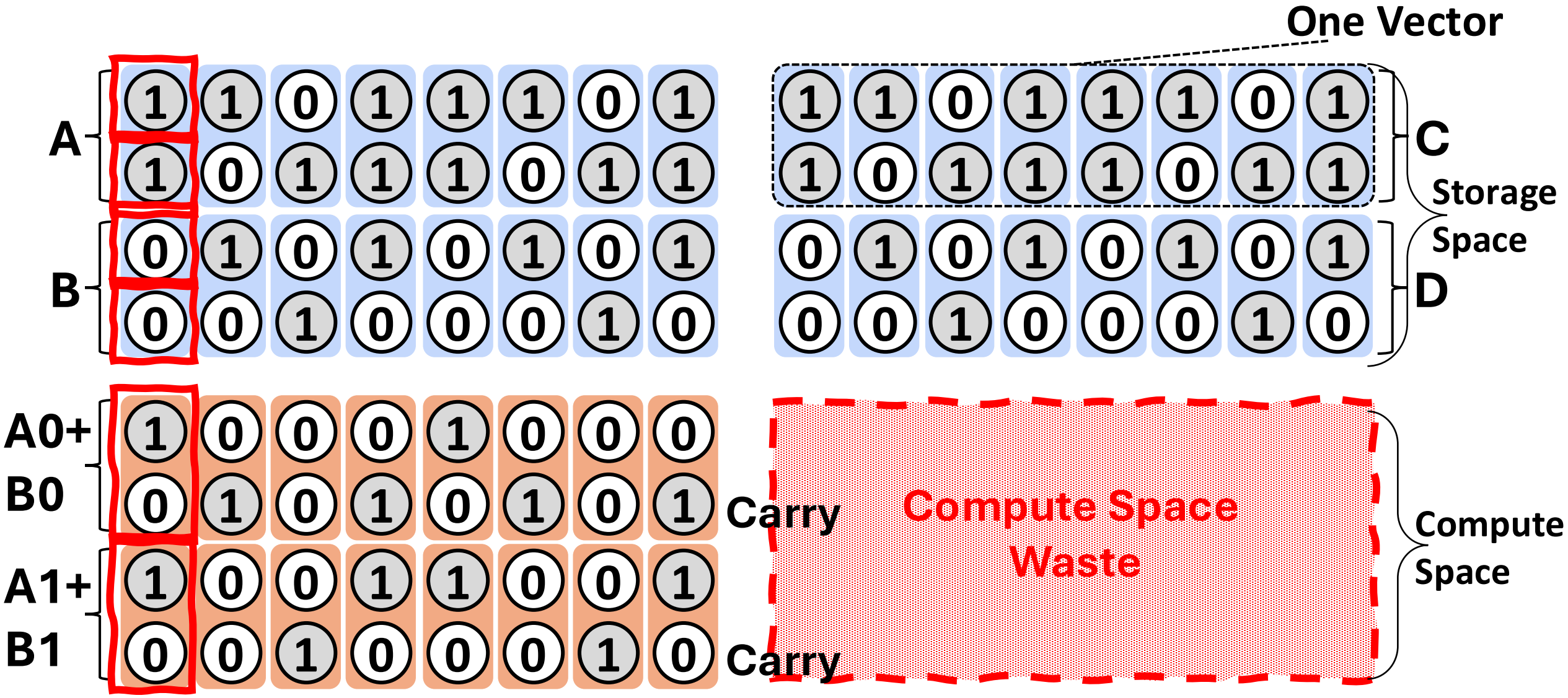


Mapping


One Vector
1 1 1 0
0 0 1 0
0 1 1 1
0 1 1 0
One Digit
Horizontal
Layout


Vertical
Layout
1 1 0 1
1 0 1 1
0 1 0 1
0 0 1 0
One Digit
One Vector

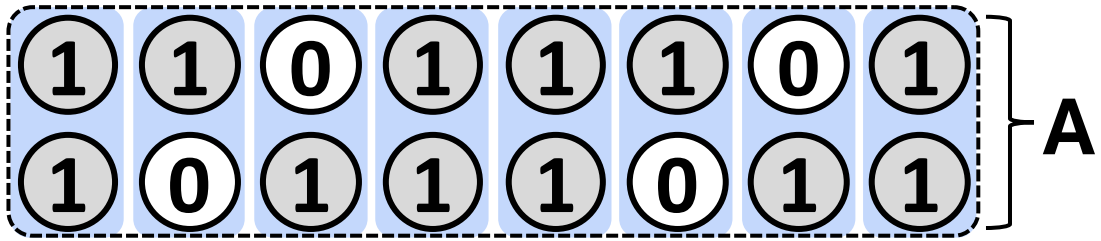
Intuitive Approach--Vertical Layout



Summary--Vertical Layout

- Digit width: **w(=2)**
- Embedding vector element count: **n(=8)**
- Aggregation scale: **k(=4)**

- Cost: **k**(rows) * **n**(row serial) + **w** (mem_read)



Evaluation--Methodology

Processor	32 cores, 4.0GHz, non-inclusive L1: 16KB, 64B cacheline, 8-way asso. L2: 256KB, 64B cacheline, 8-way asso. LLC: 1MB/core, 64B cacheline, 16-way asso.
Memory Controller	Page management: open-page Scheduler: FR-FCFS Data buffer latency: 4
DRAM Memory	DDR5-3200, 4GB, ×4 I/O width 2 channels, 2 ranks, 8 banks-groups, 4 banks/bank-group 8Kb local row buffer CL-nRCD-nRP: 24-24-24 nCCDS-nCCDL-nCCDLWR: 8-8-32
PIM Boards	Ambit, DRISA, ELP2M, Ambit + ELP2M
Dataset Workloads	**Anime**, **MovieLens**, **LastFM**, **Amazon**: CDs&Vinyl, Kindle Store, TV&movie