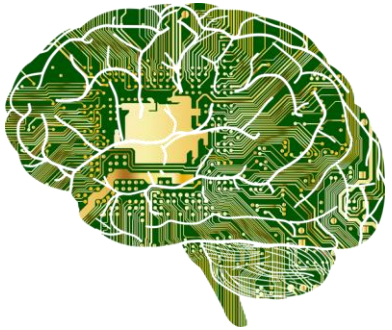
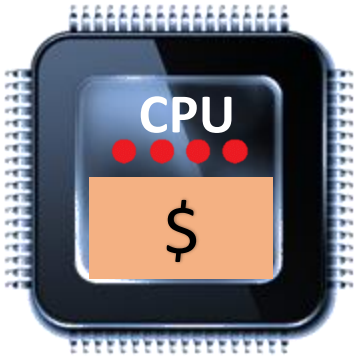
Neural Cache: Bit-Serial In-Cache Acceleration of Deep Neural Networks

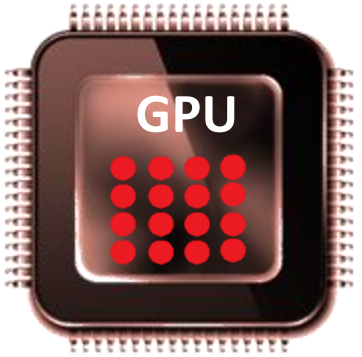
Charles Eckert Xiaowei Wang Jingcheng Wang Arun Subramaniyan
Dennis Sylvester David Blaauw Reetuparna Das

M-Bits Research Group
UNIVERSITY OF MICHIGAN

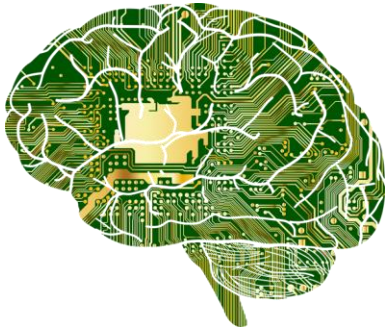
Can we transform CPU into a neural accelerator?

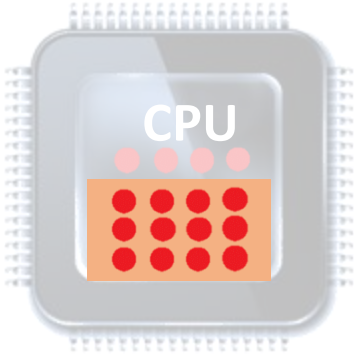




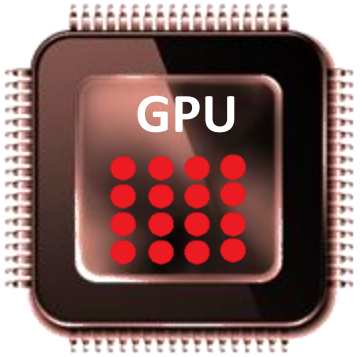


Can we transform CPU into a neural accelerator?





Neural Cache

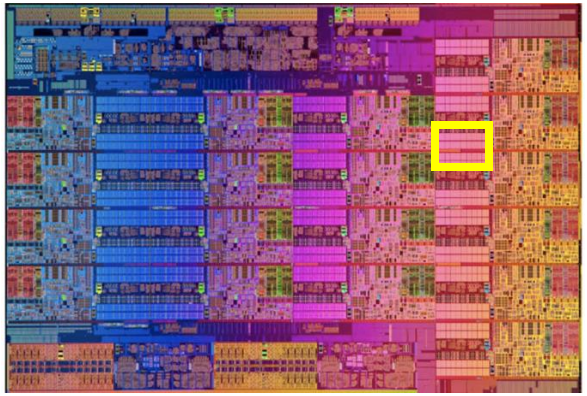


++ Parallelism

-- Data Movement

Transforming caches into massively parallel vector ALUs

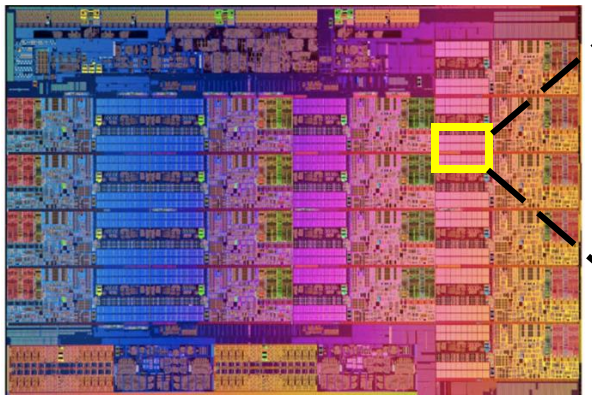
18-core Xeon processor
45 MB LLC



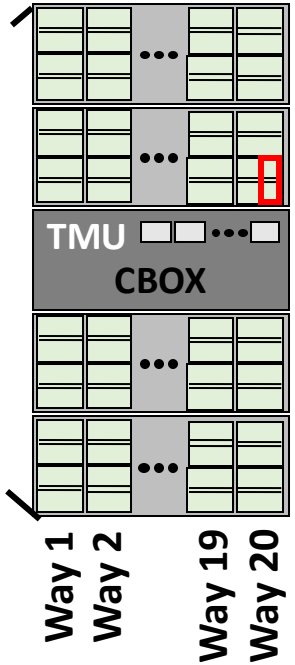
18 LLC slices

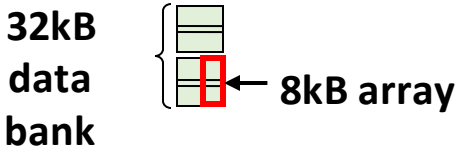
Transforming caches into massively parallel vector ALUs

18-core Xeon processor
45 MB LLC



2.5MB LLC slice



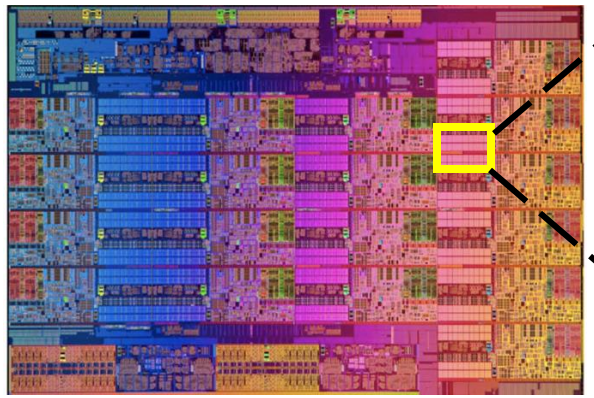


18 LLC slices

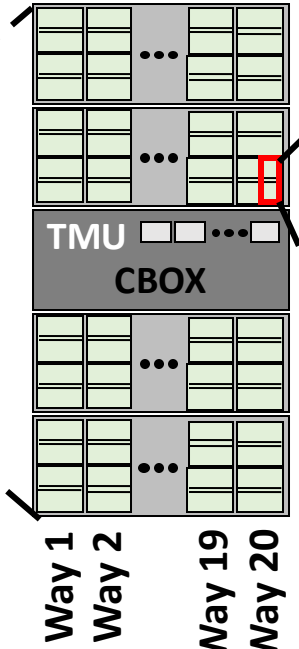
360 ways

Transforming caches into massively parallel vector ALUs

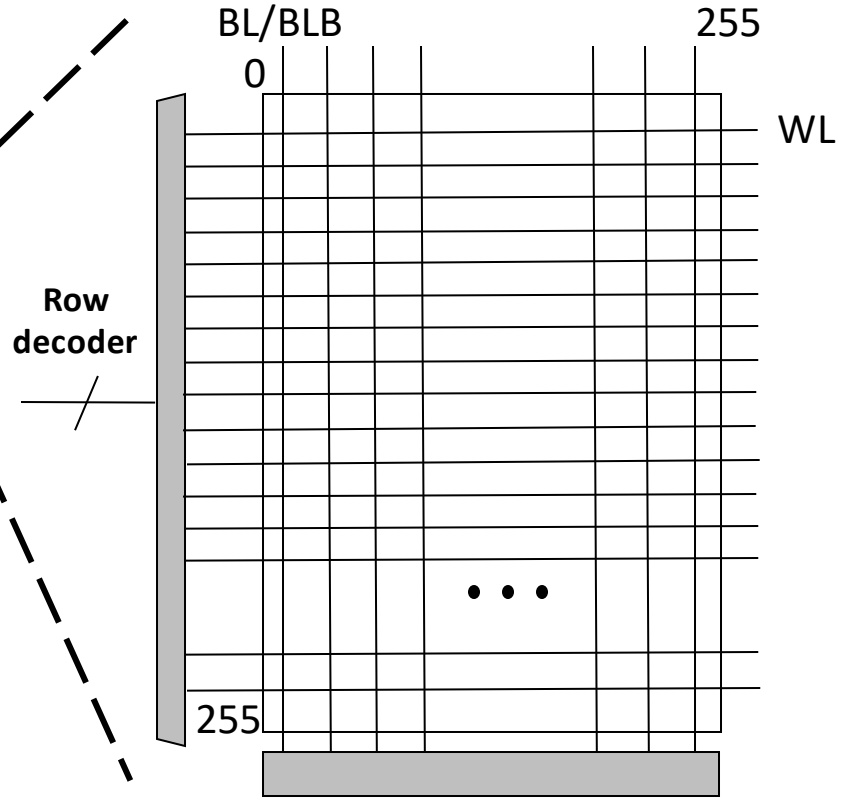
18-core Xeon processor
45 MB LLC

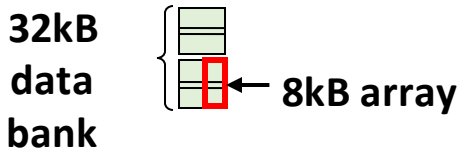


2.5MB LLC slice



8kB SRAM array



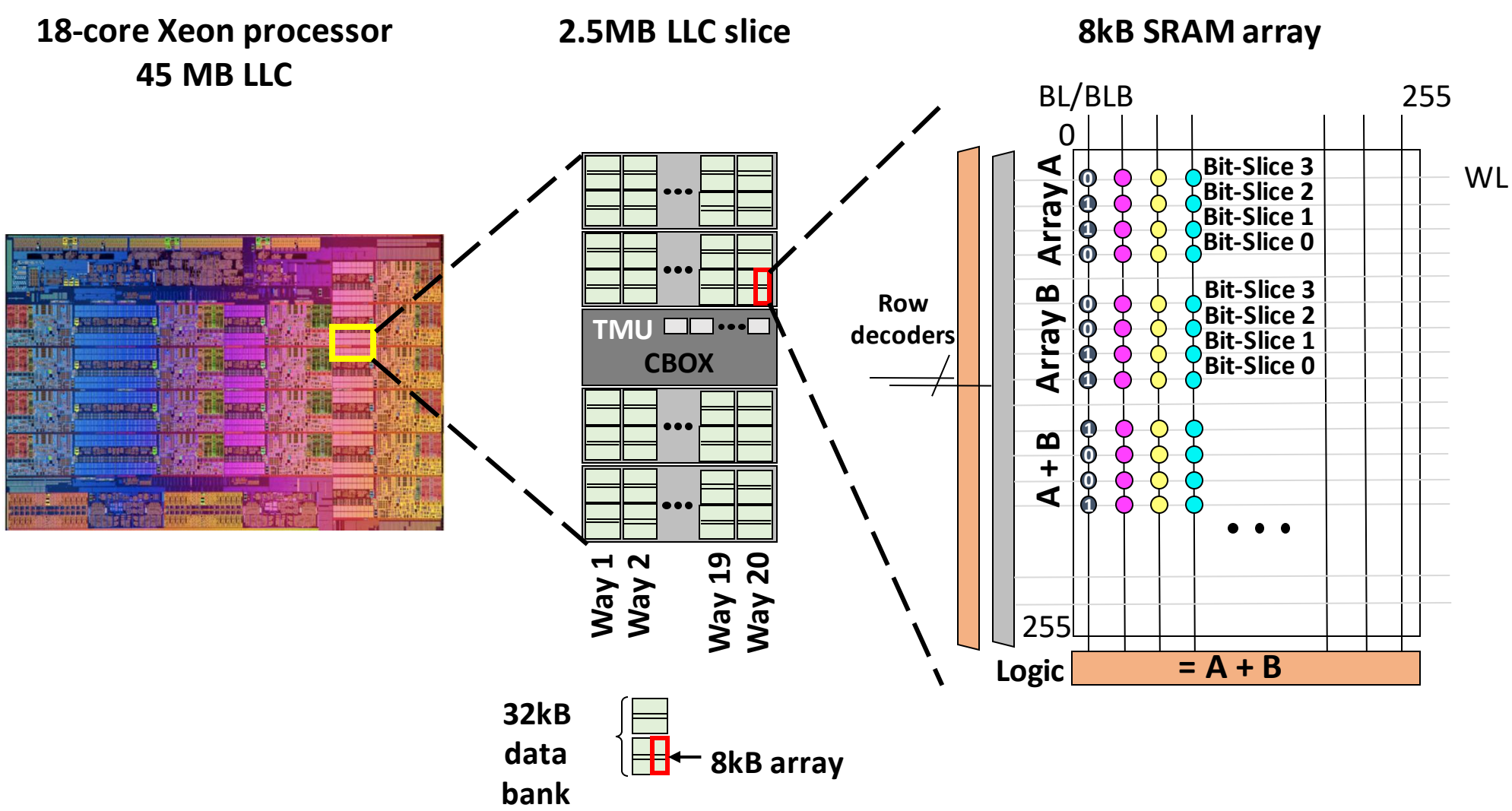


18 LLC slices

360 ways

5760 arrays

Transforming caches into massively parallel vector ALUs



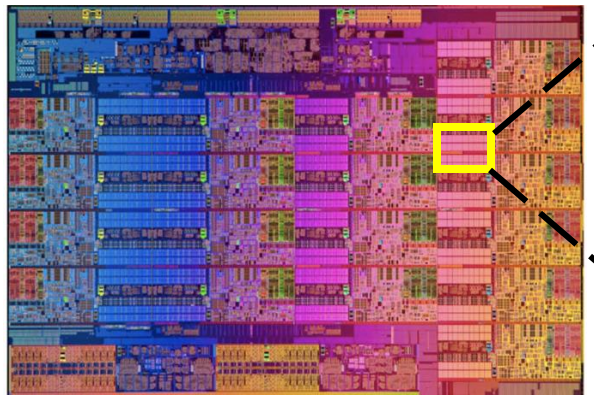
18 LLC slices

360 ways

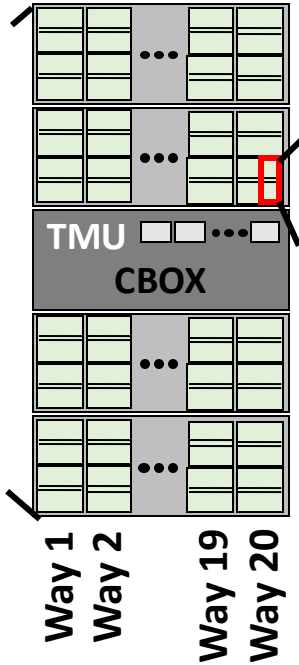
5760 arrays

Transforming caches into massively parallel vector ALUs

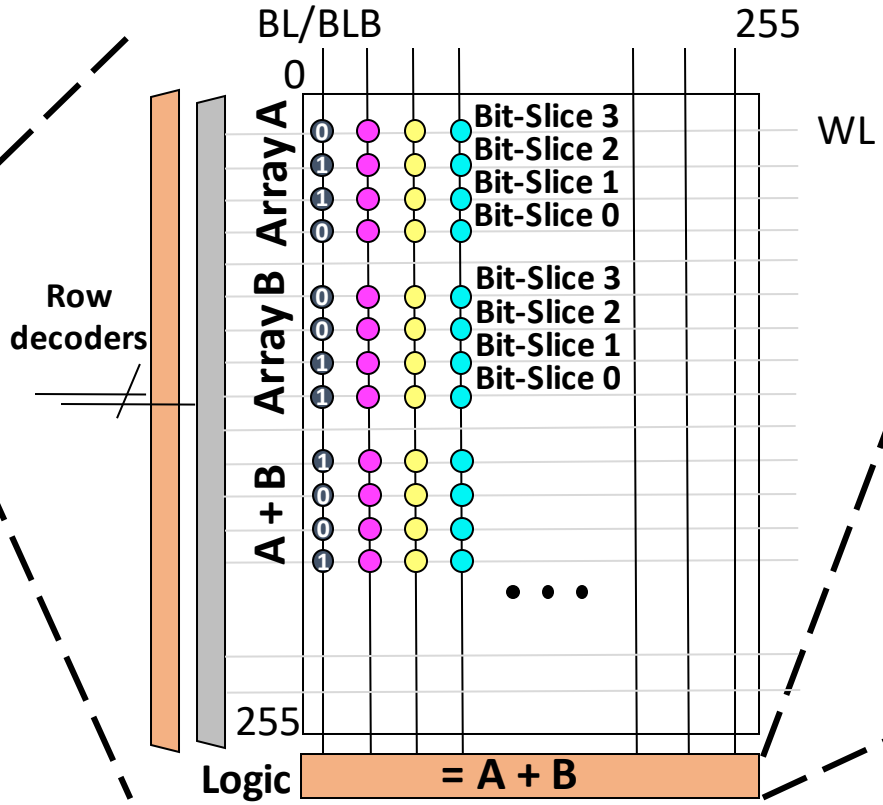
18-core Xeon processor
45 MB LLC



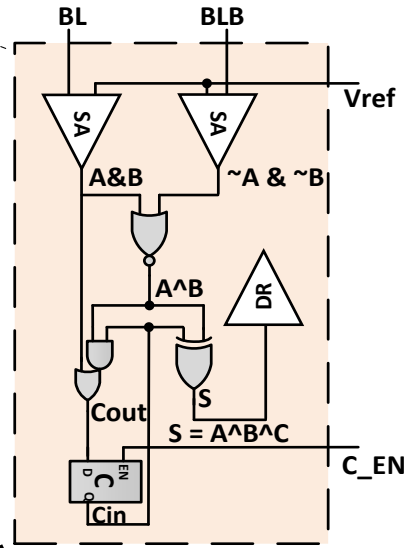
2.5MB LLC slice



8kB SRAM array



Bitline ALU



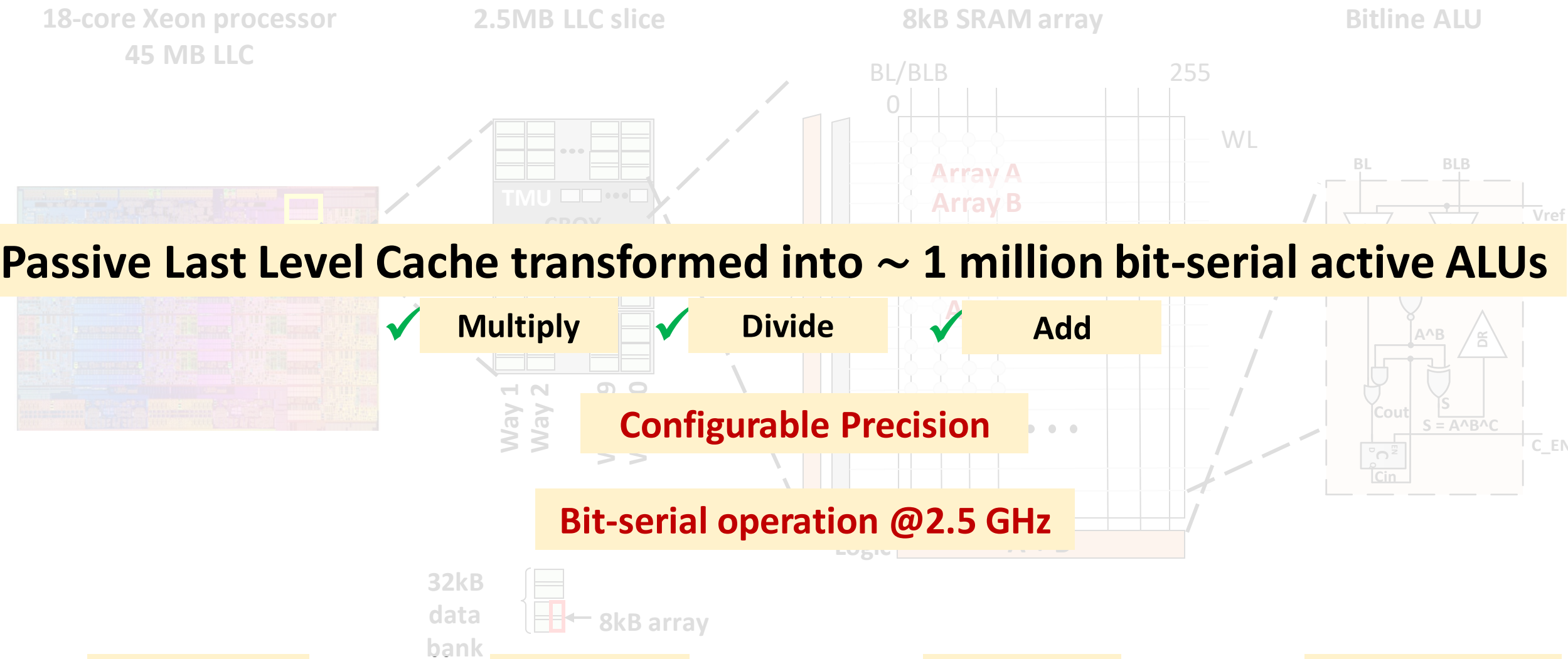
18 LLC slices

360 ways

5760 arrays

1,474,560 ALUs

Transforming caches into massively parallel vector ALUs



Passive Last Level Cache transformed into ~ 1 million bit-serial active ALUs

✓ **Multiply** ✓ **Divide** ✓ **Add**

Configurable Precision

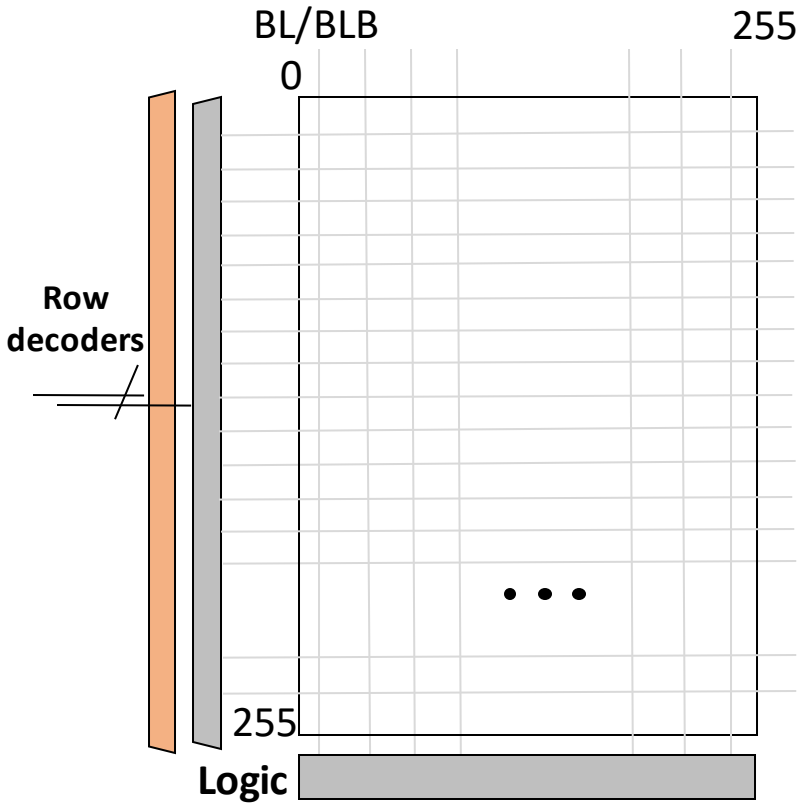
Bit-serial operation @2.5 GHz

18 LLC slices	**360 ways**	**5760 arrays**	**1,474,560 ALUs**

Why bit-serial?

A + B

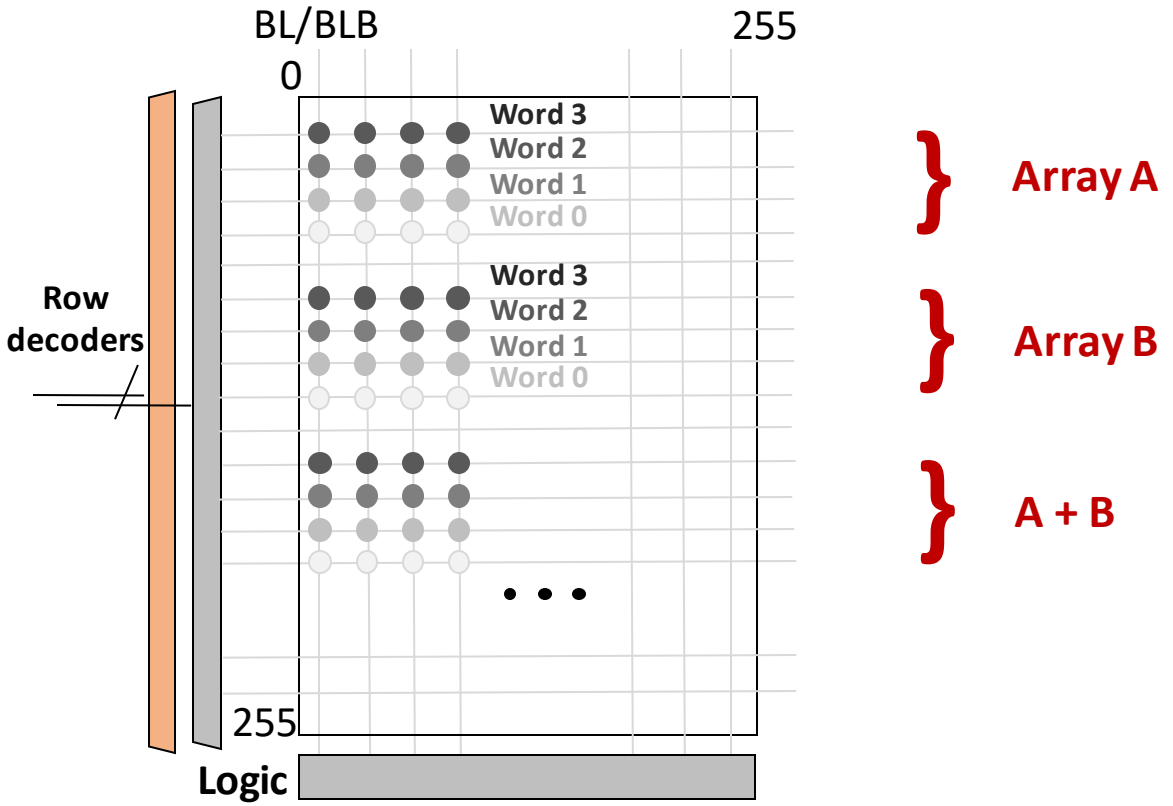
Bit-parallel arithmetic



Why bit-serial?

A + B

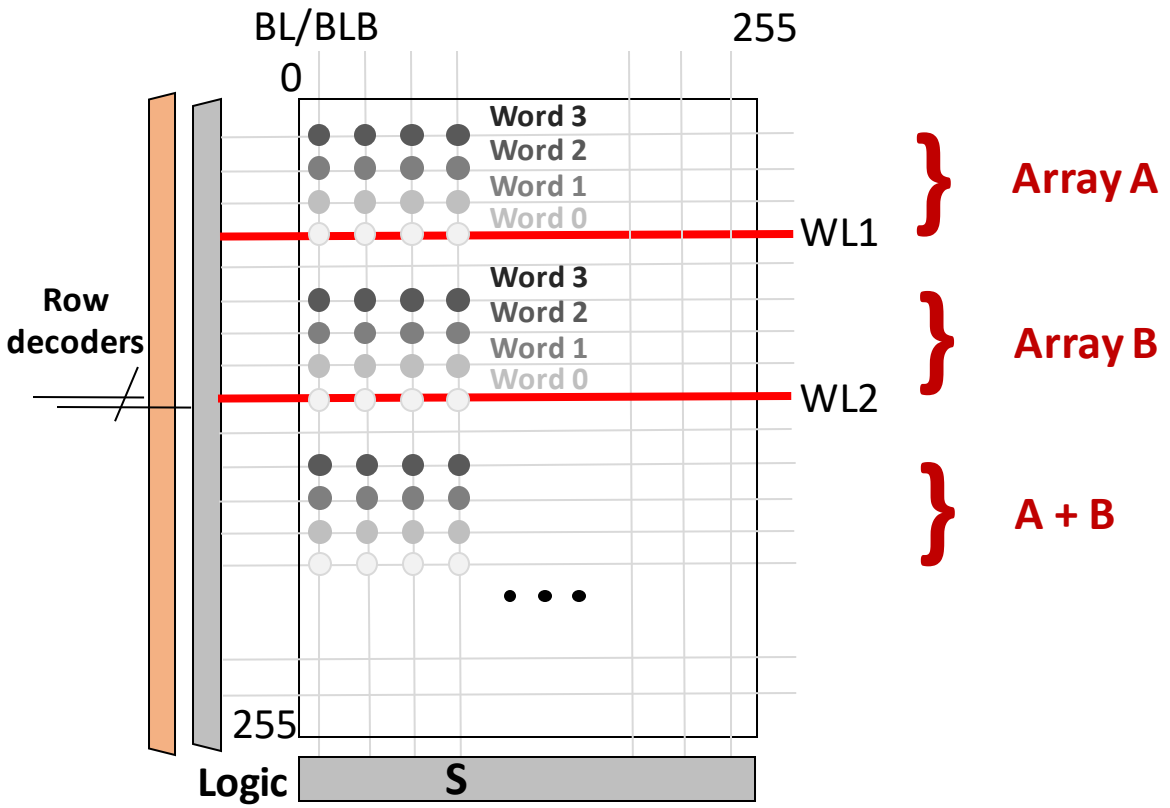
Bit-parallel arithmetic



Why bit-serial?

A + B

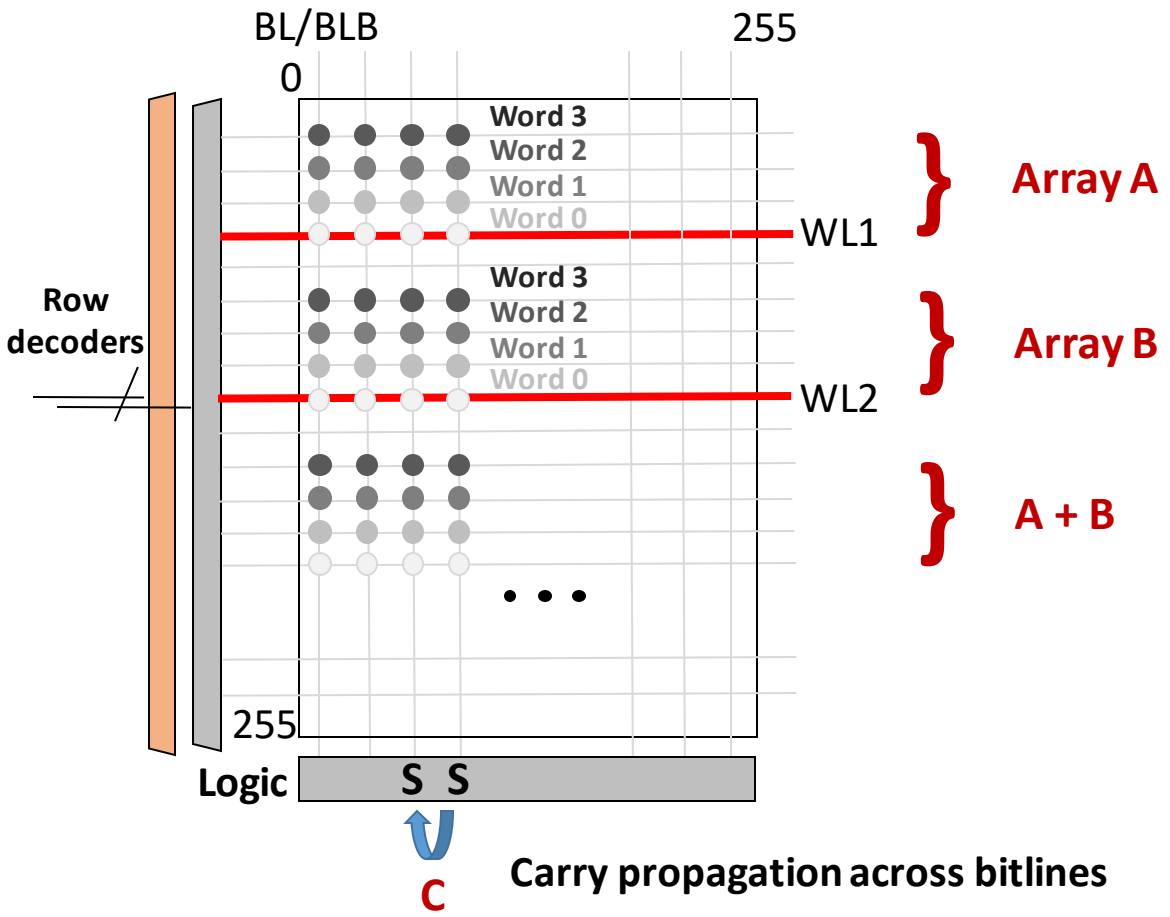
Bit-parallel arithmetic



Why bit-serial?

A + B

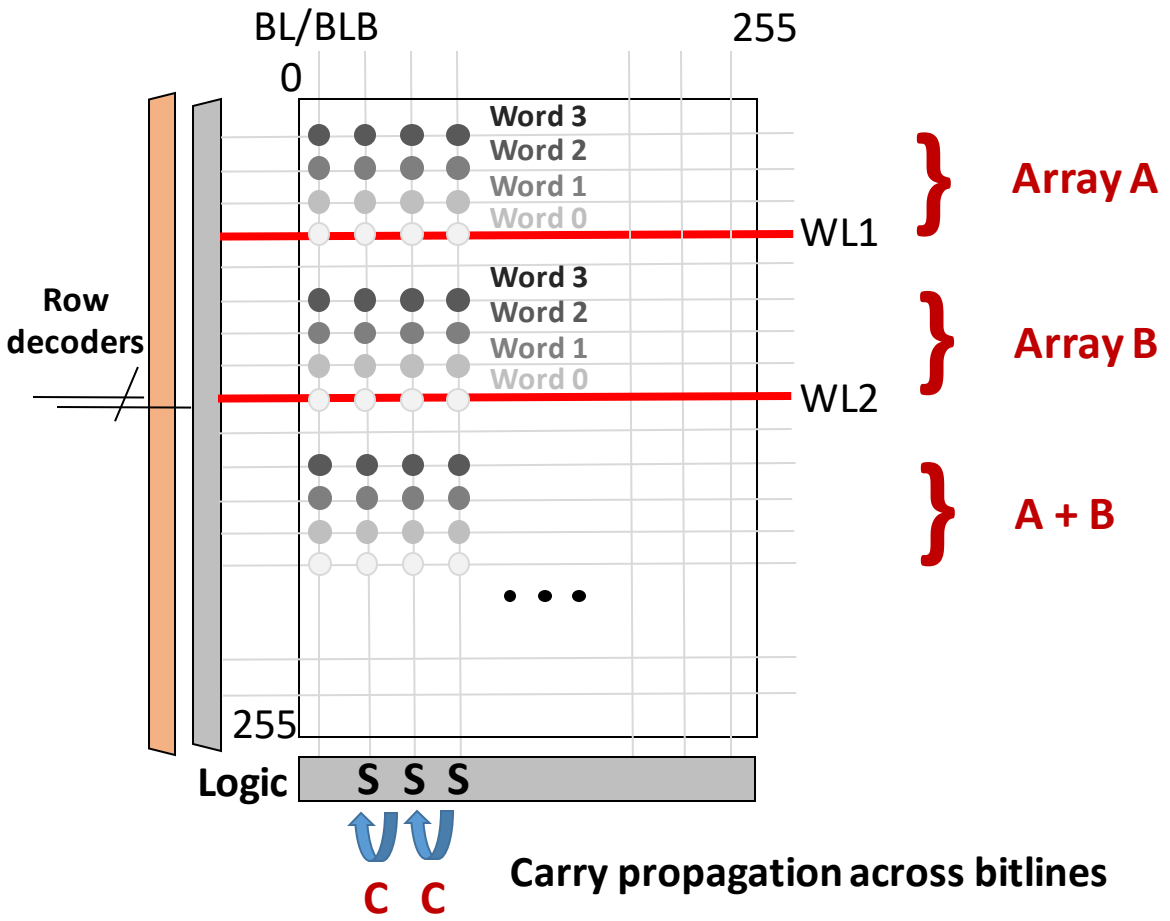
Bit-parallel arithmetic



Why bit-serial?

A + B

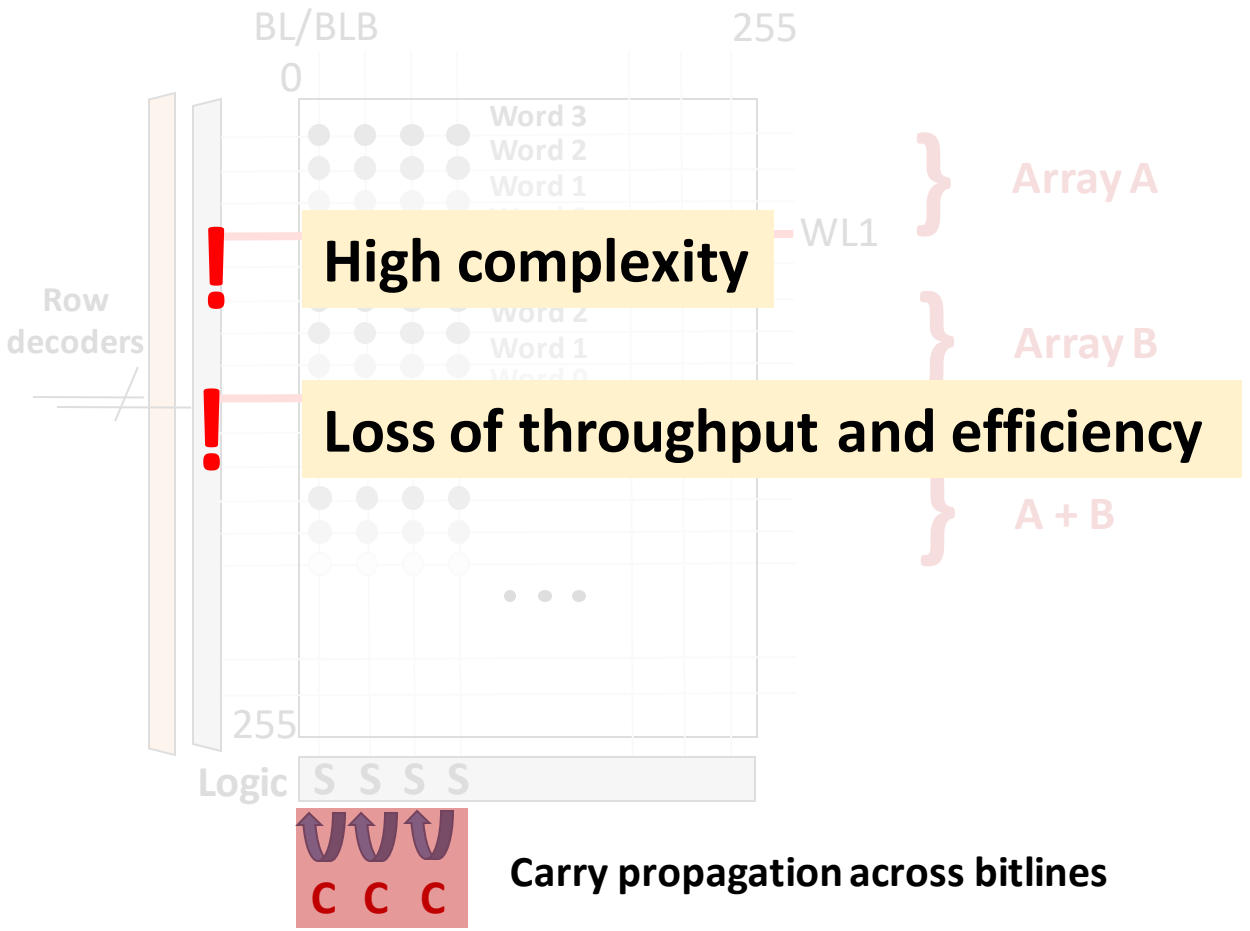
Bit-parallel arithmetic



Why bit-serial?

A + B

Bit-parallel arithmetic

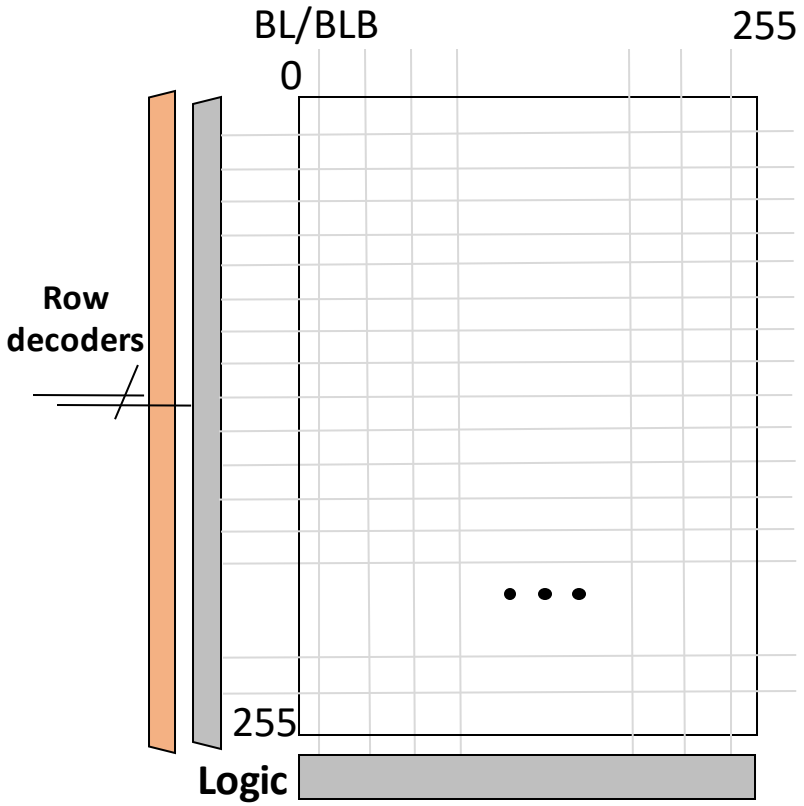


Carry propagation across bitlines

Why bit-serial?

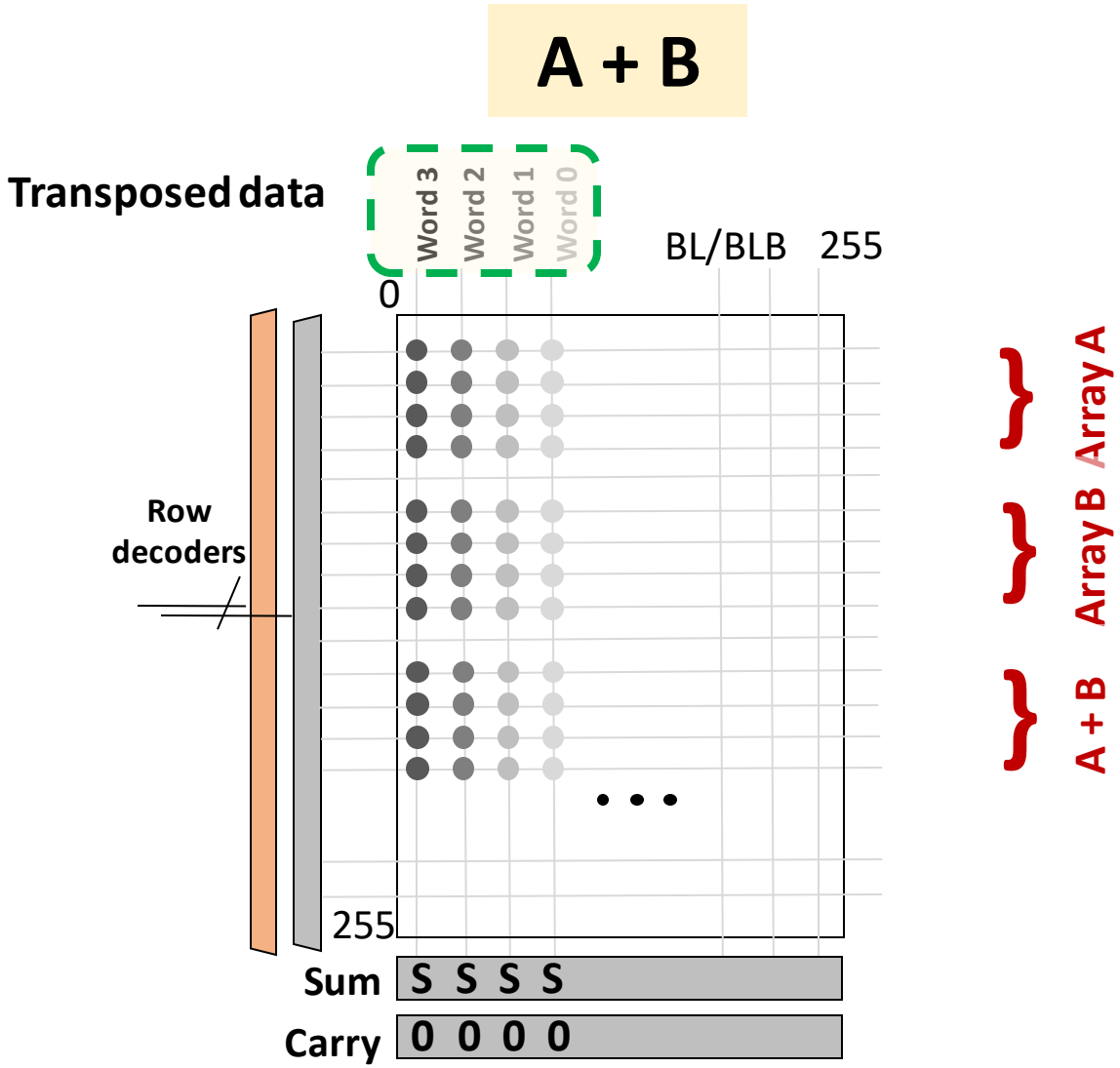
A + B

Bit-serial arithmetic



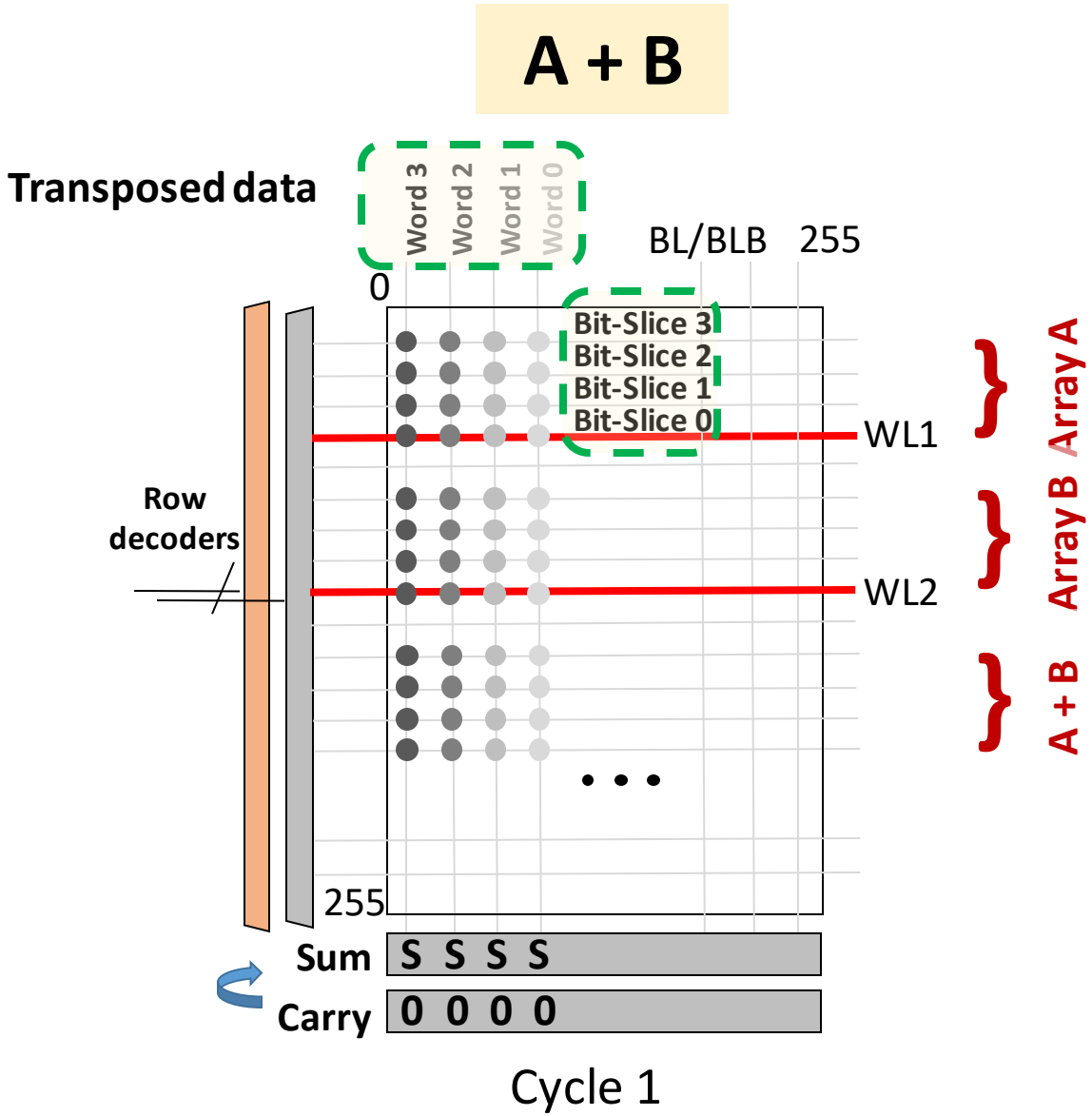
Why bit-serial?

Bit-serial arithmetic



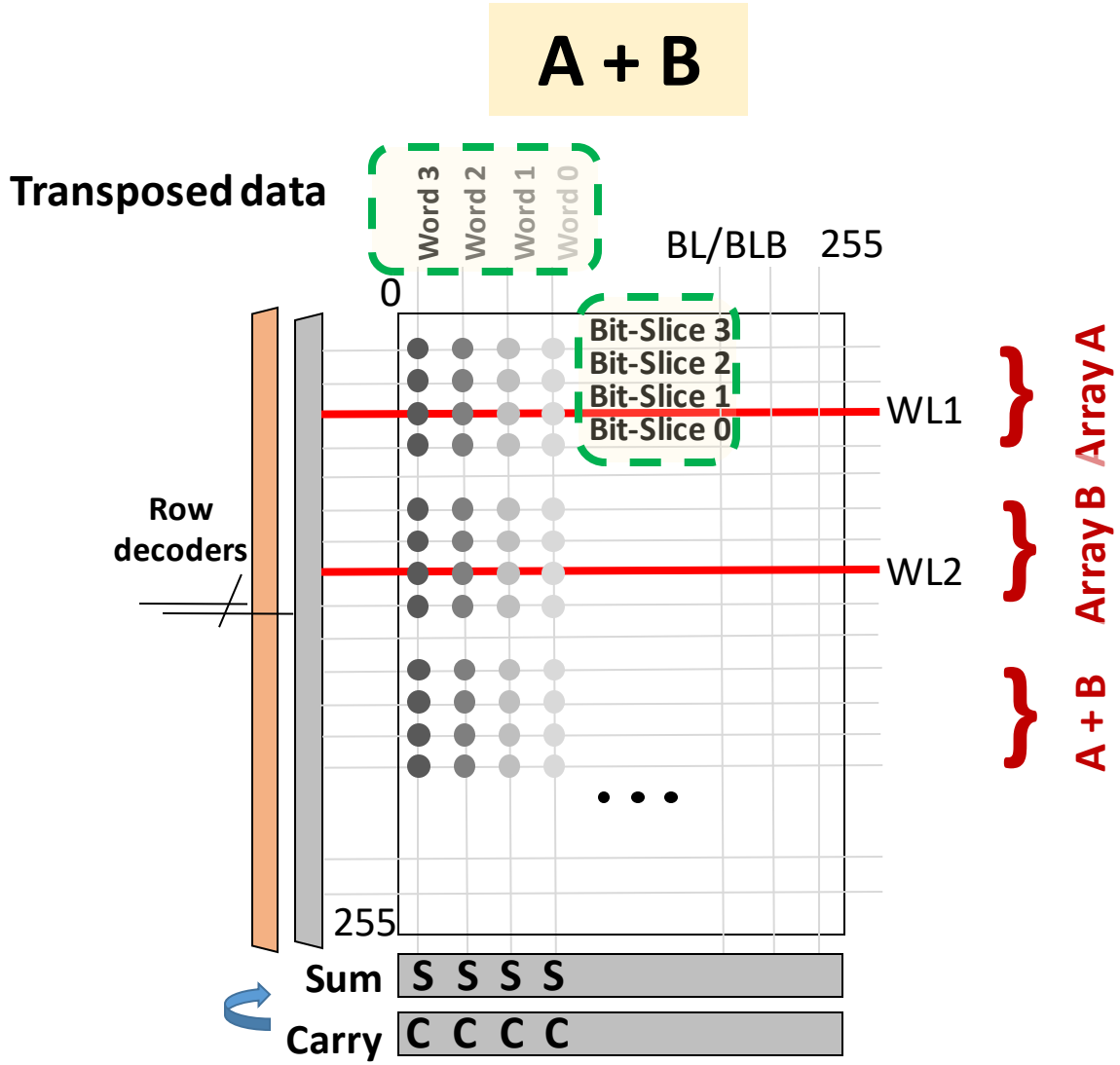
Why bit-serial?

Bit-serial arithmetic



Why bit-serial?

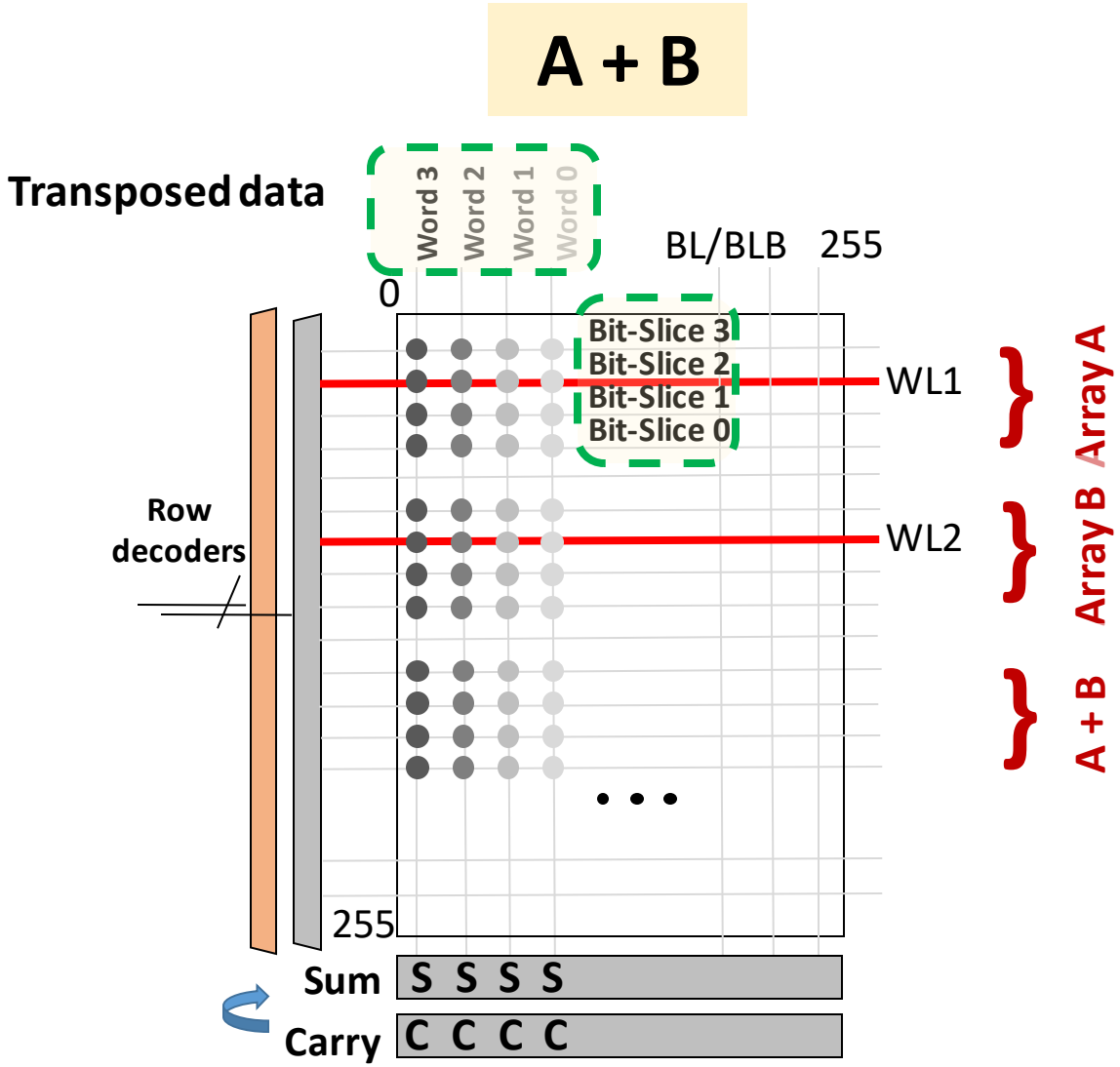
Bit-serial arithmetic



Cycle 2

Why bit-serial?

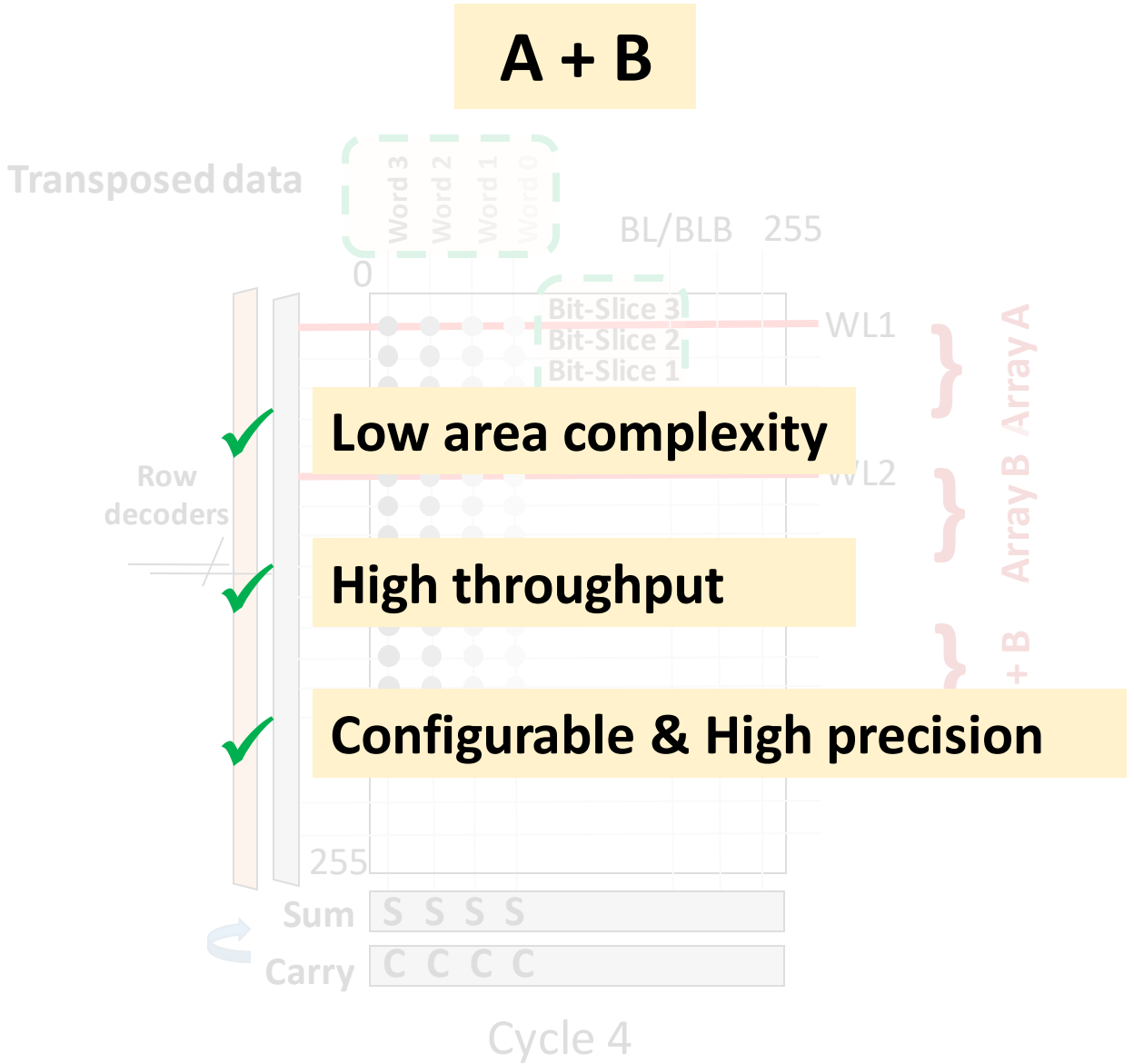
Bit-serial arithmetic



Cycle 3

Why bit-serial?

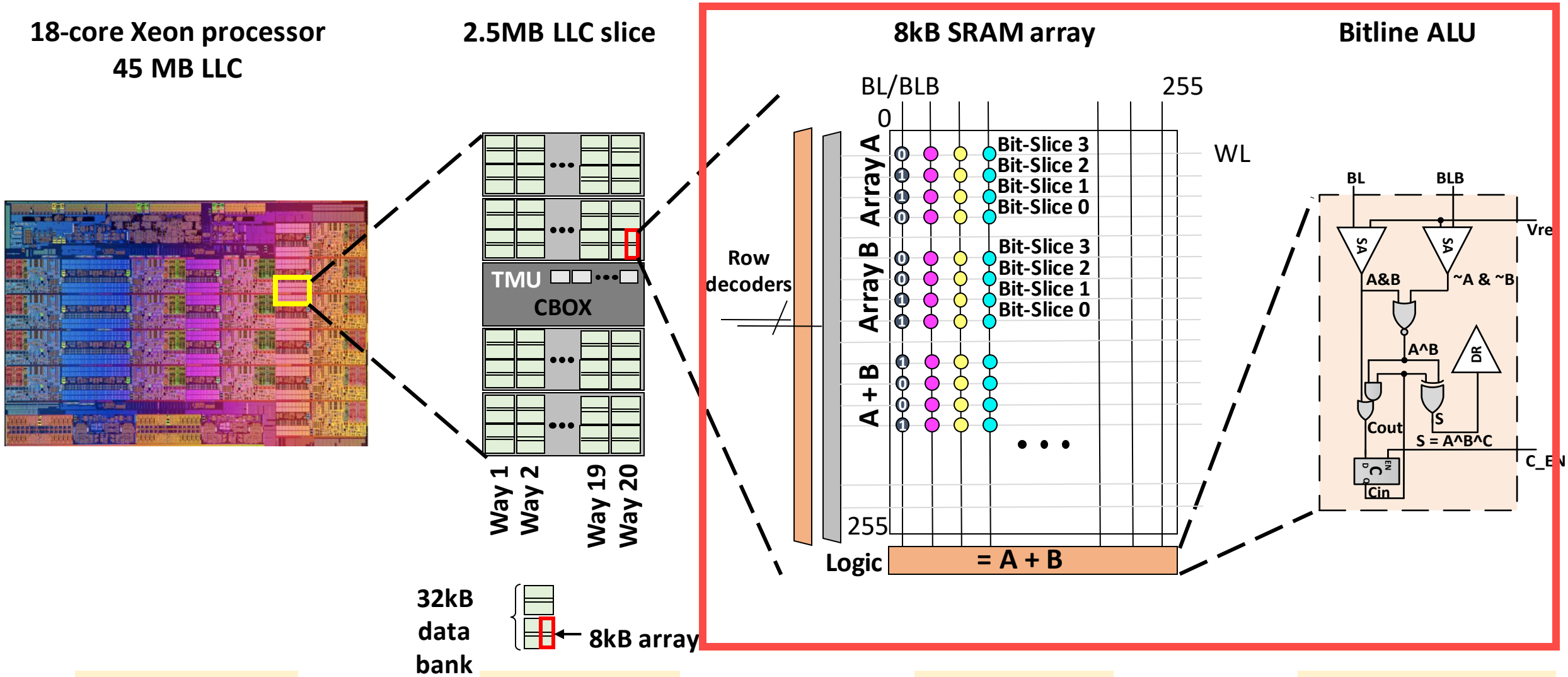
Bit-serial arithmetic



Outline

- Motivation
- **Bit-Serial Arithmetic**
- Transpose
- Mapping of Convolution to Array
- Methodology
- Results

In-SRAM Arithmetic



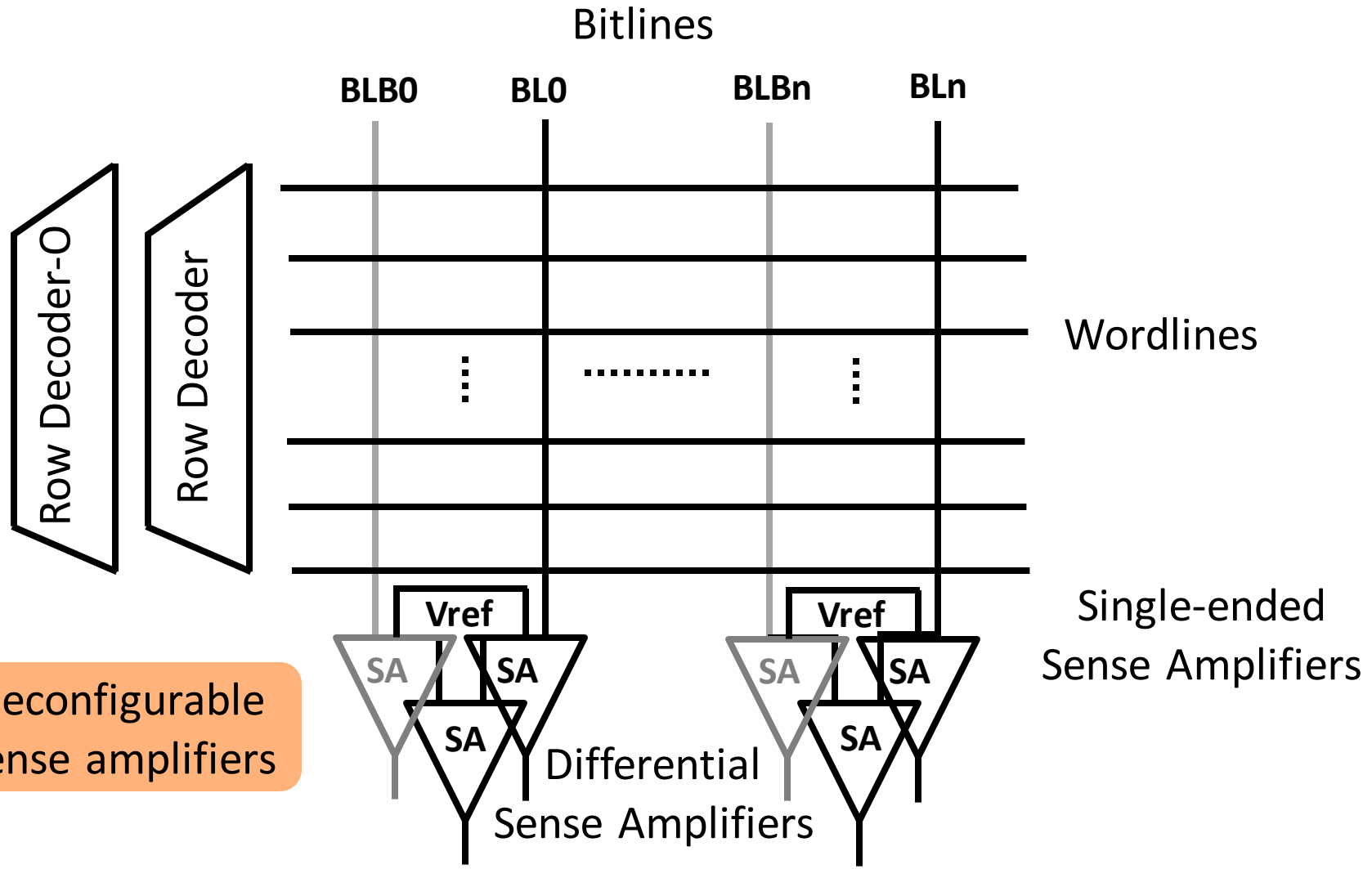
18 LLC slices | **360 ways** | **5760 arrays** | **1,474,560 ALUs**

Logical Operations In-SRAM

Changes

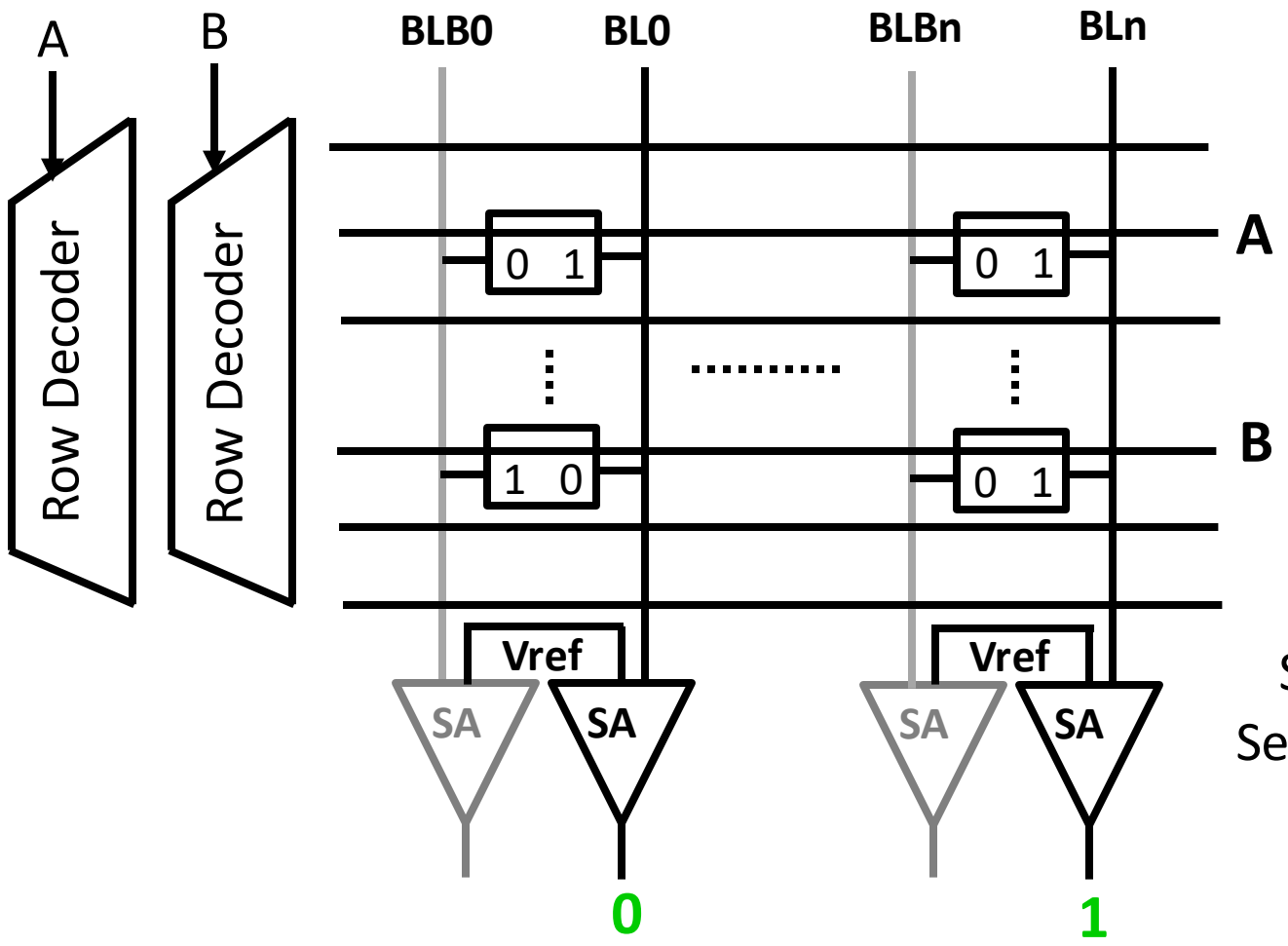
Additional row decoder

Reconfigurable sense amplifiers



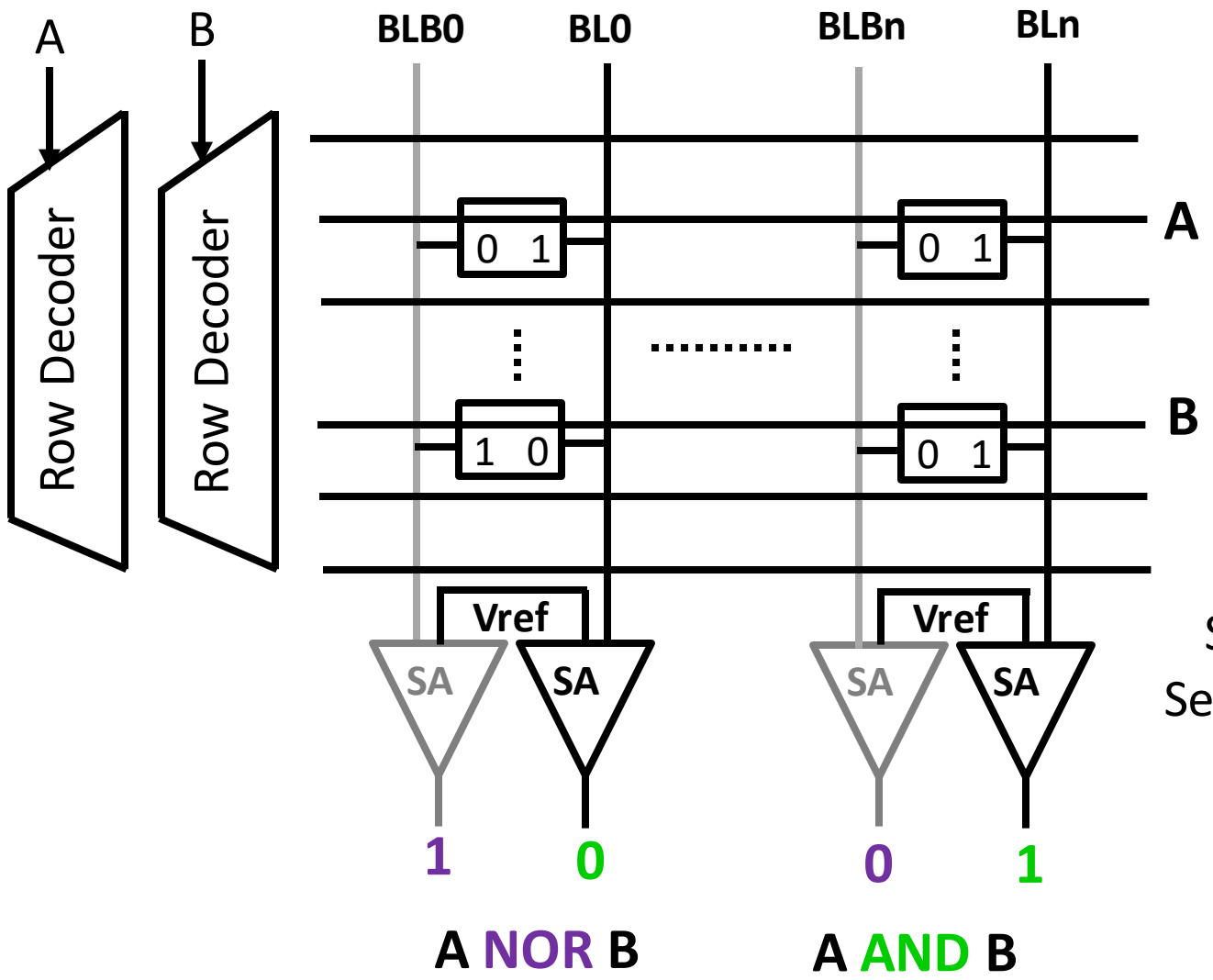
Logical Operations In-SRAM

A **AND** B



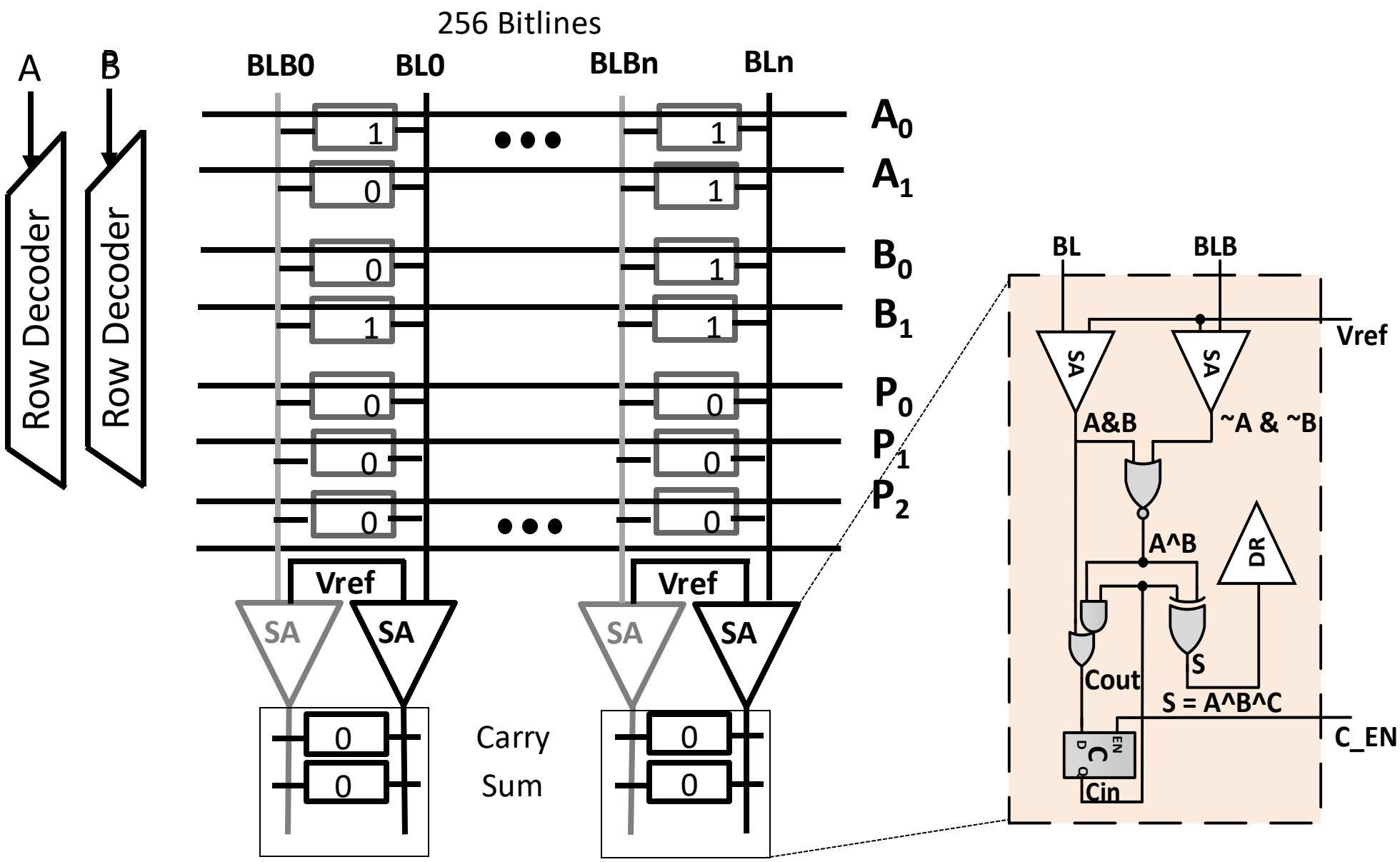
Single-ended Sense Amplifiers

Logical Operations In-SRAM

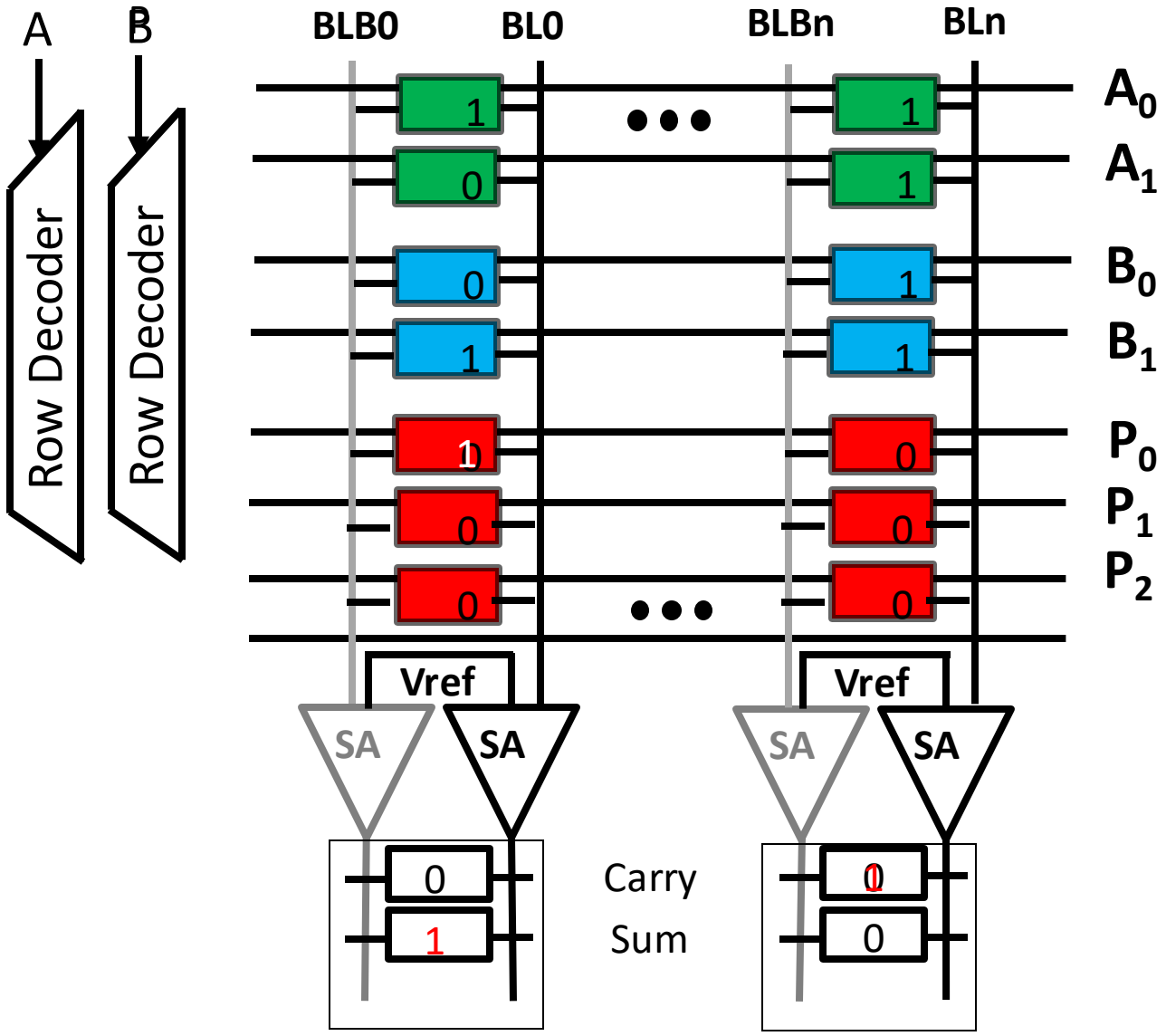


Single-ended Sense Amplifiers

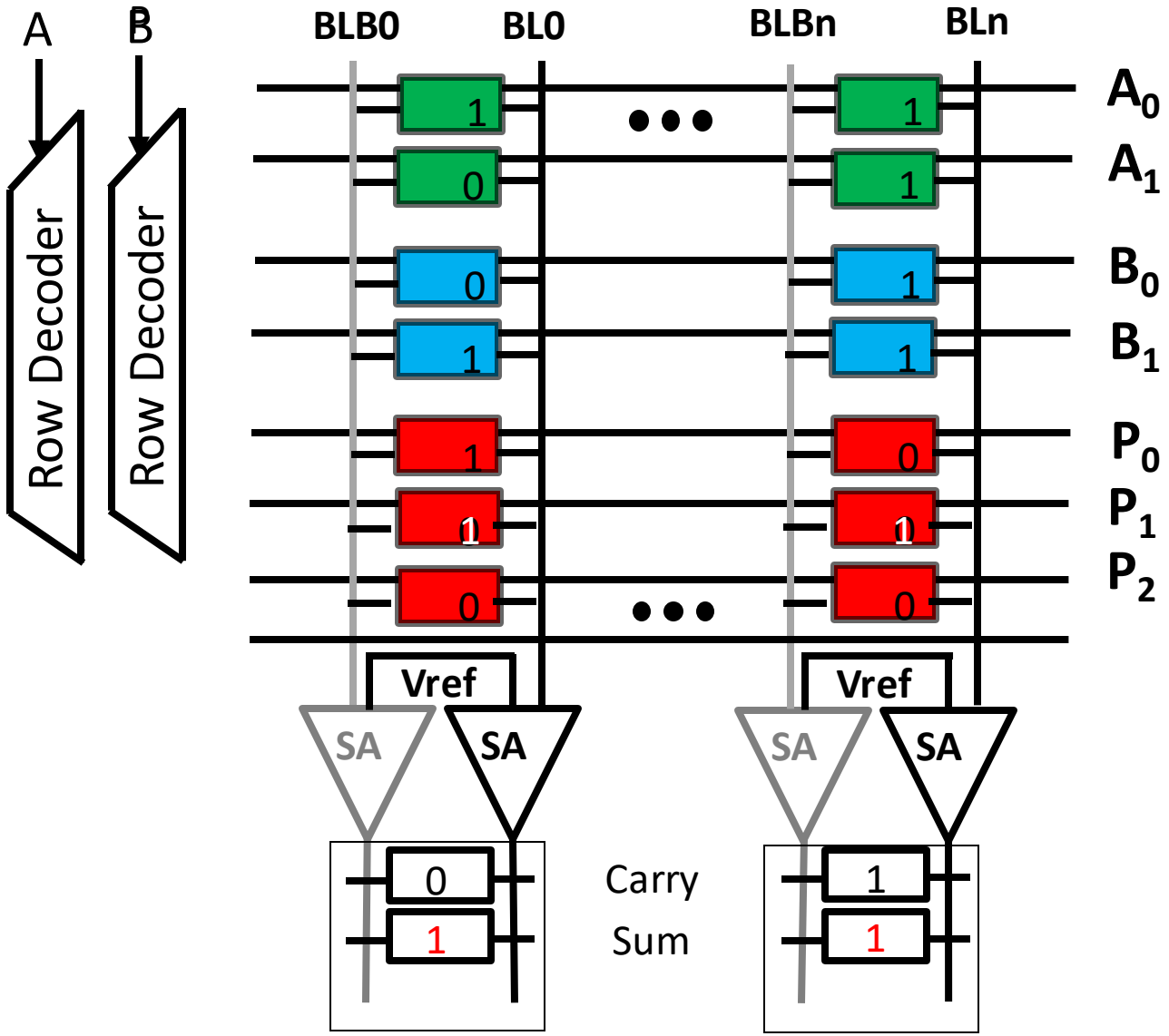
Addition In-SRAM



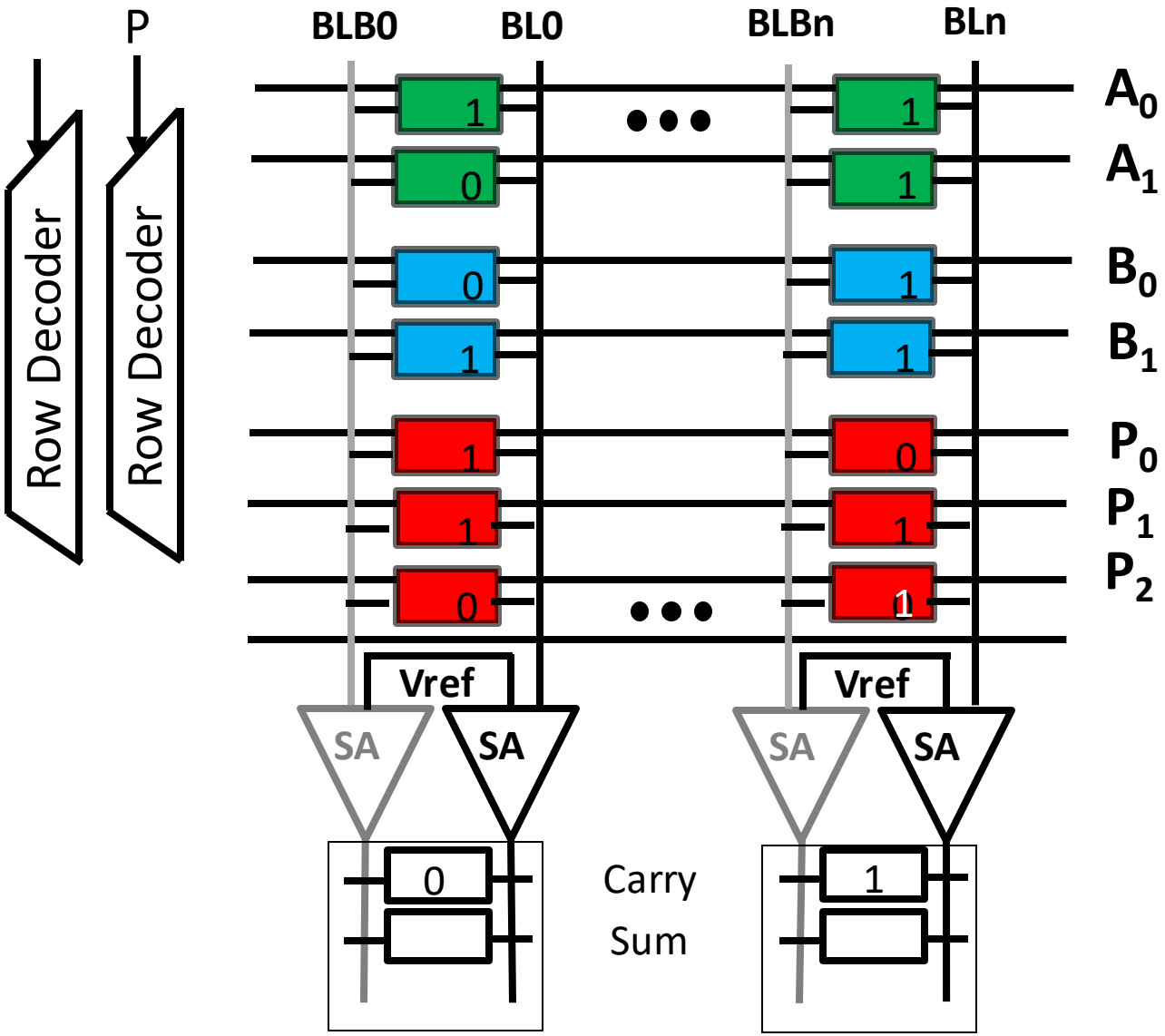
Addition [Cycle 1]



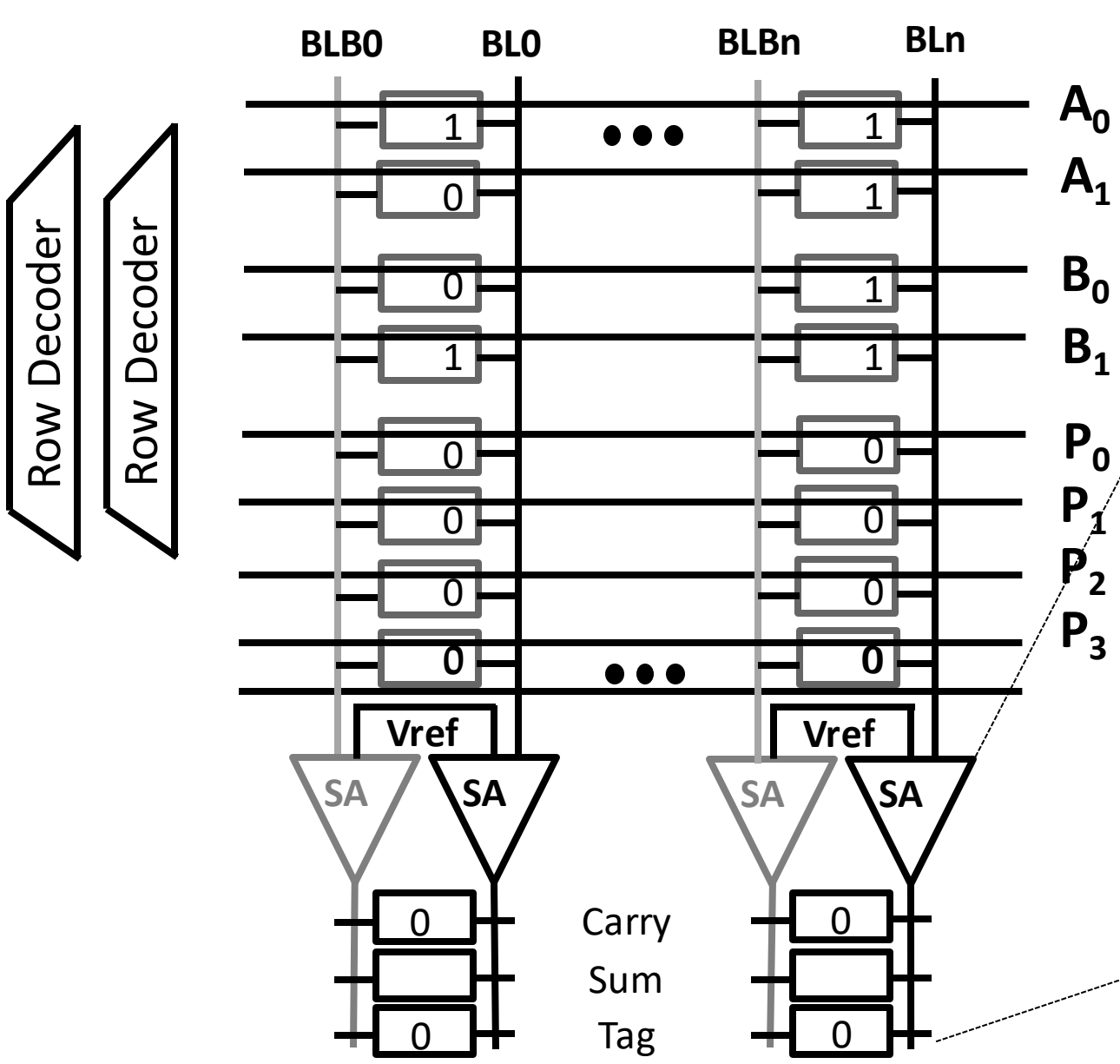
Addition [Cycle 2]

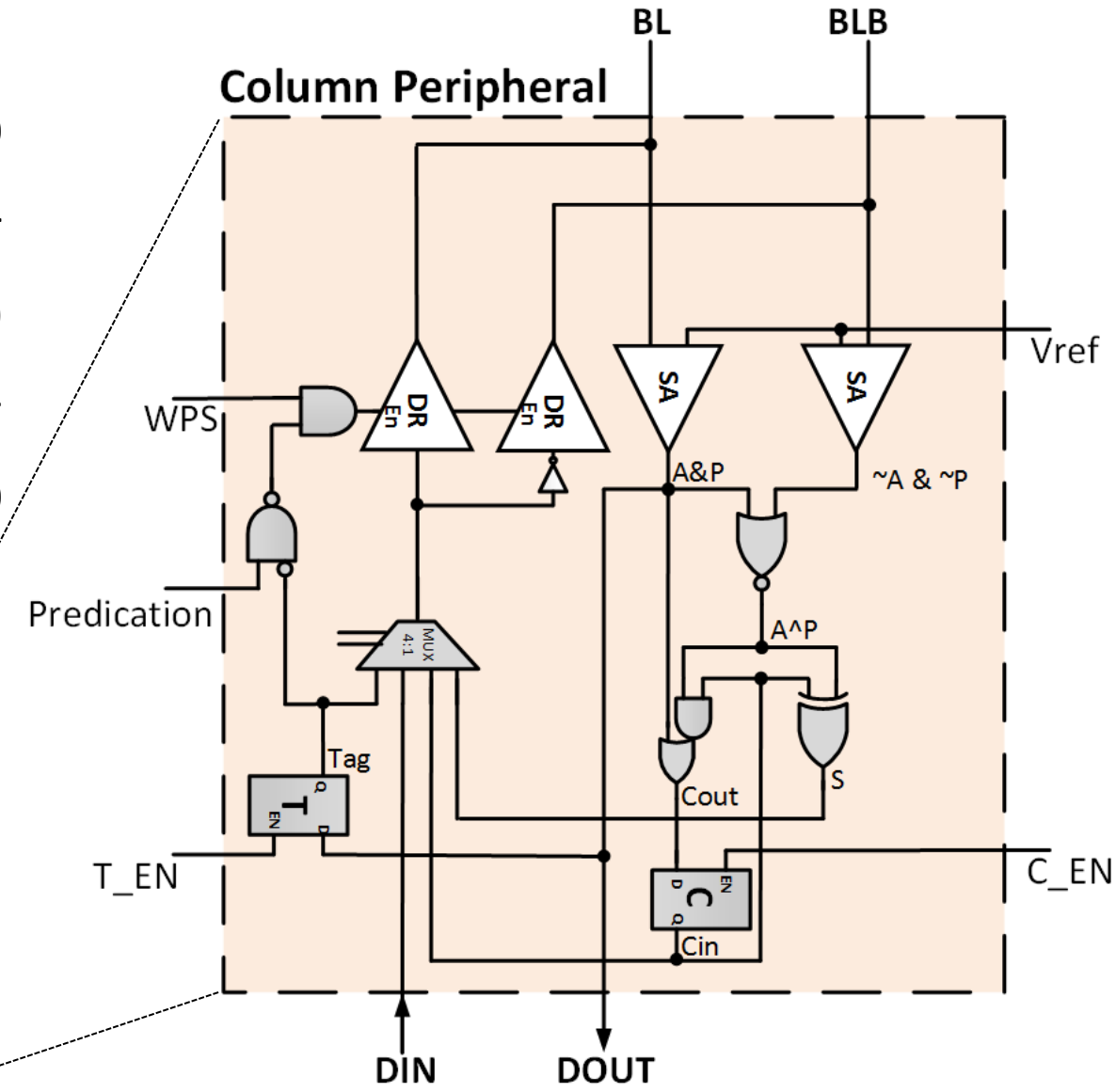


Addition [Cycle 3]

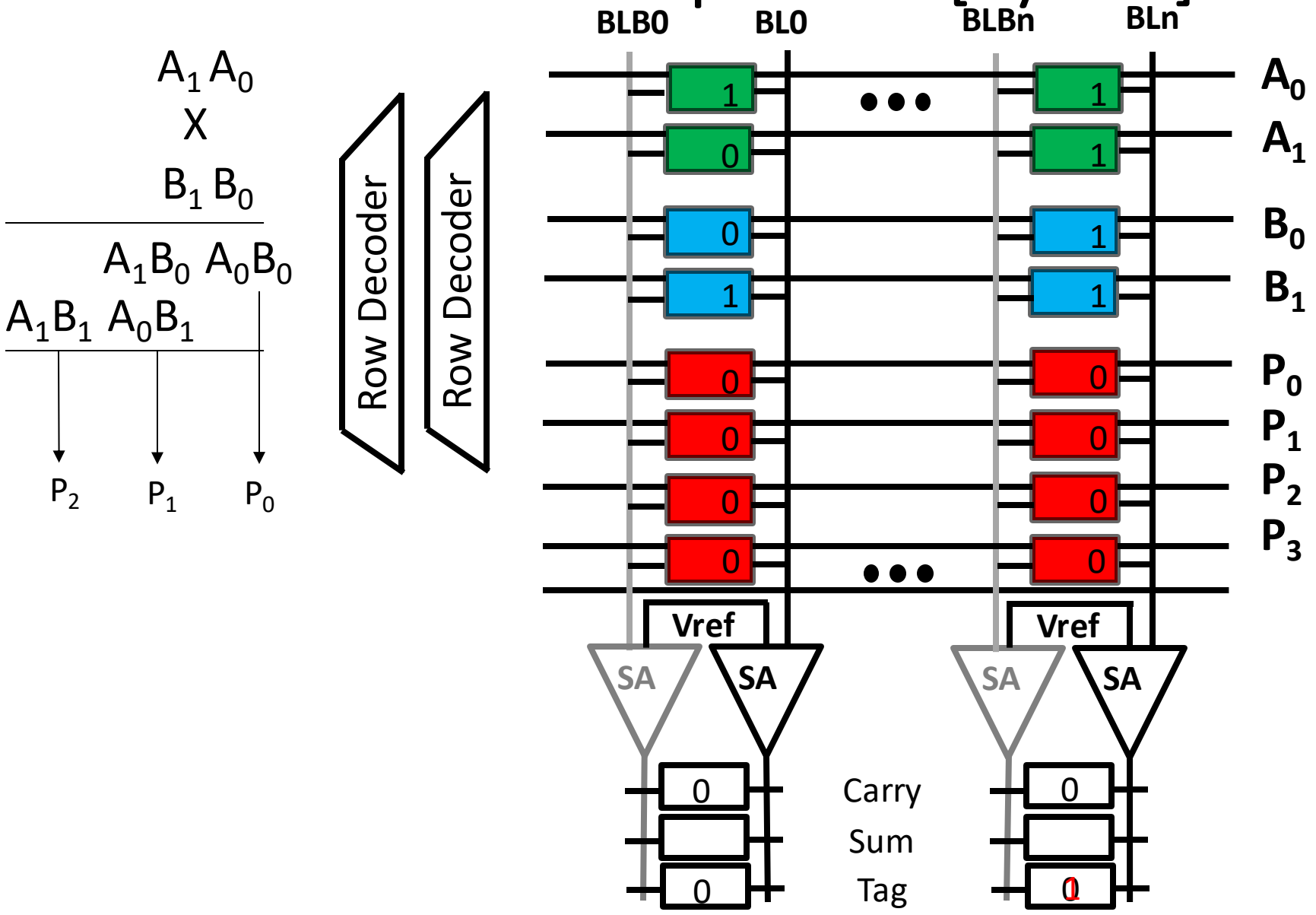


Multiplication In-SRAM

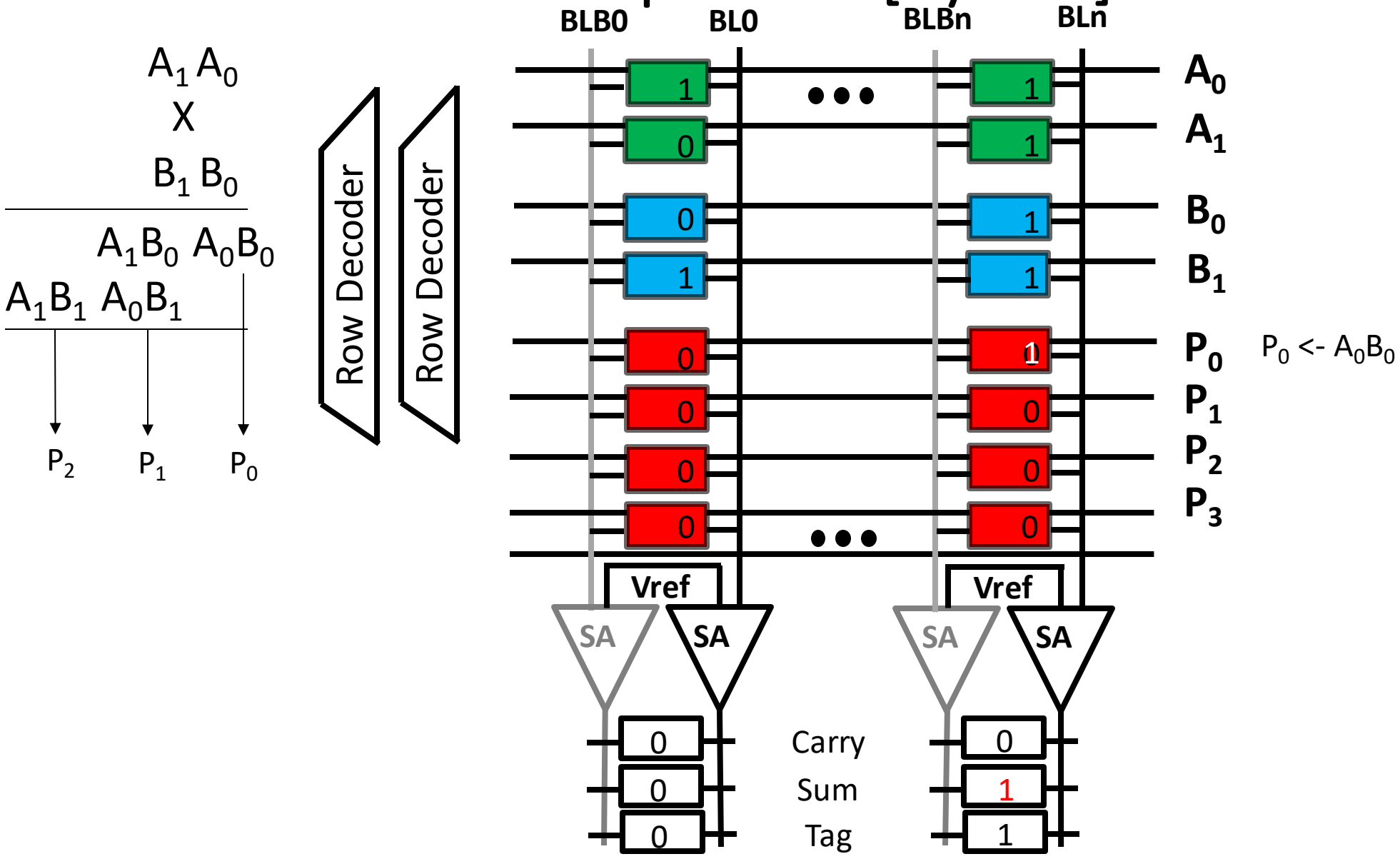




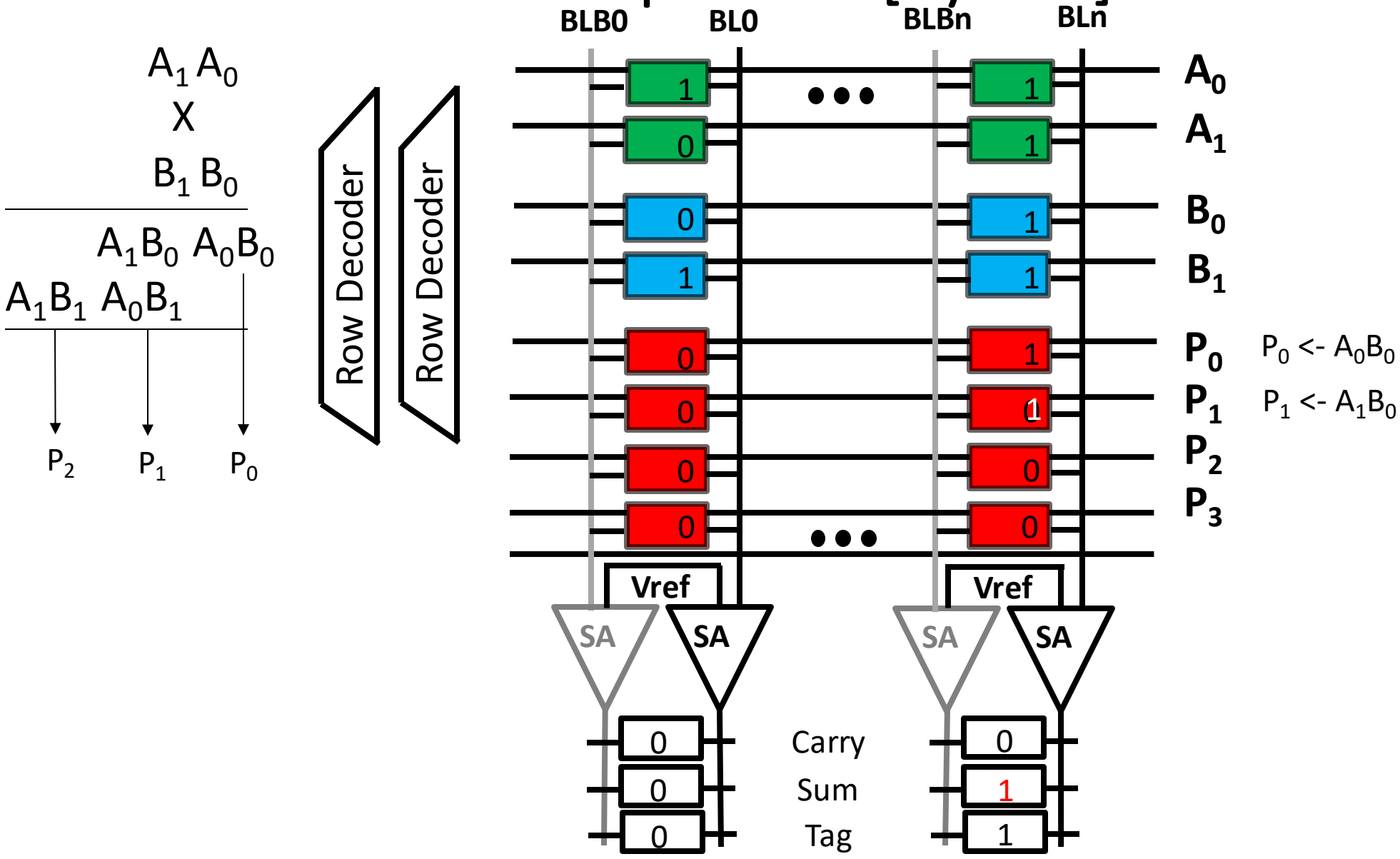
Multiplication [Cycle 1]



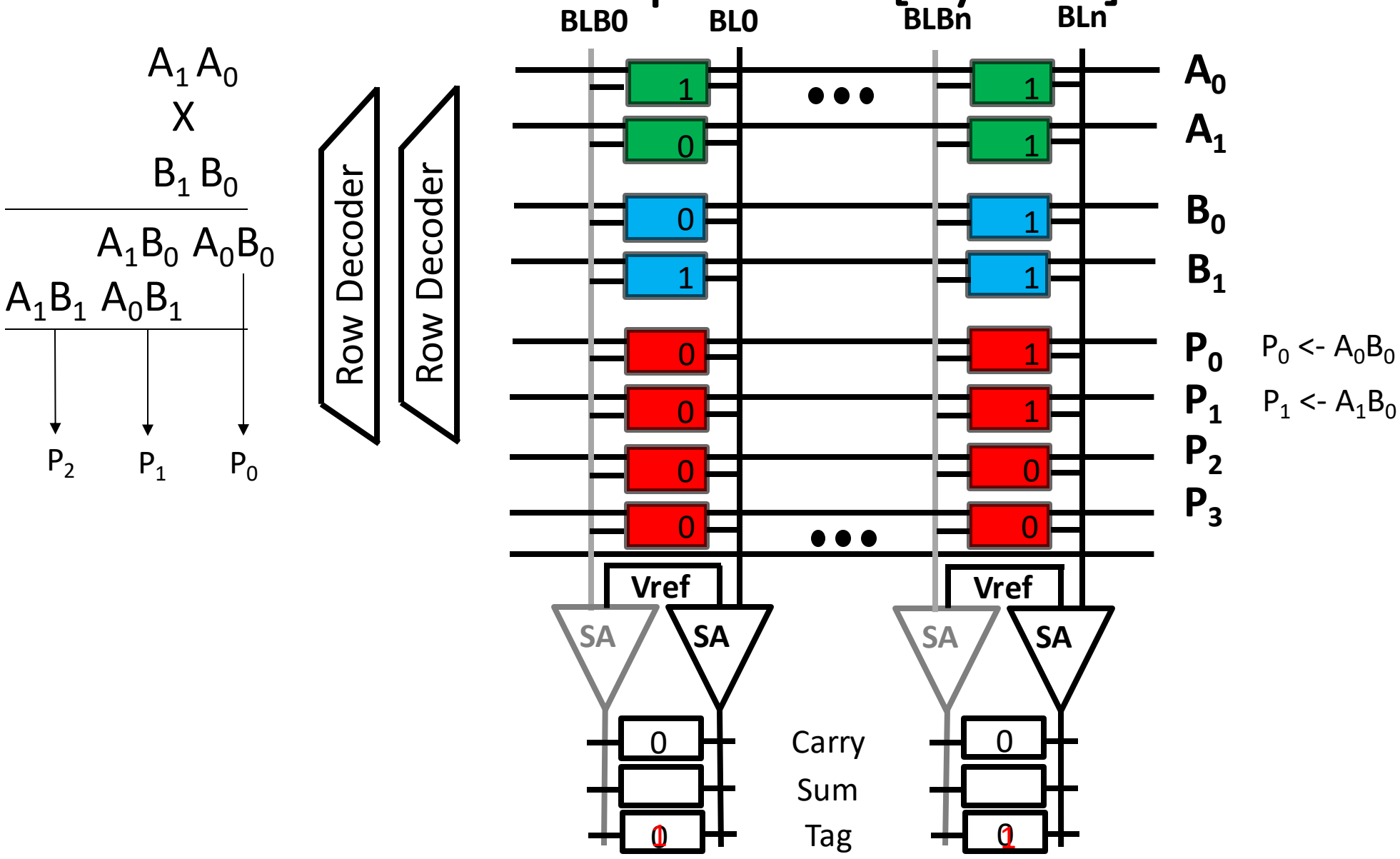
Multiplication [Cycle 2]



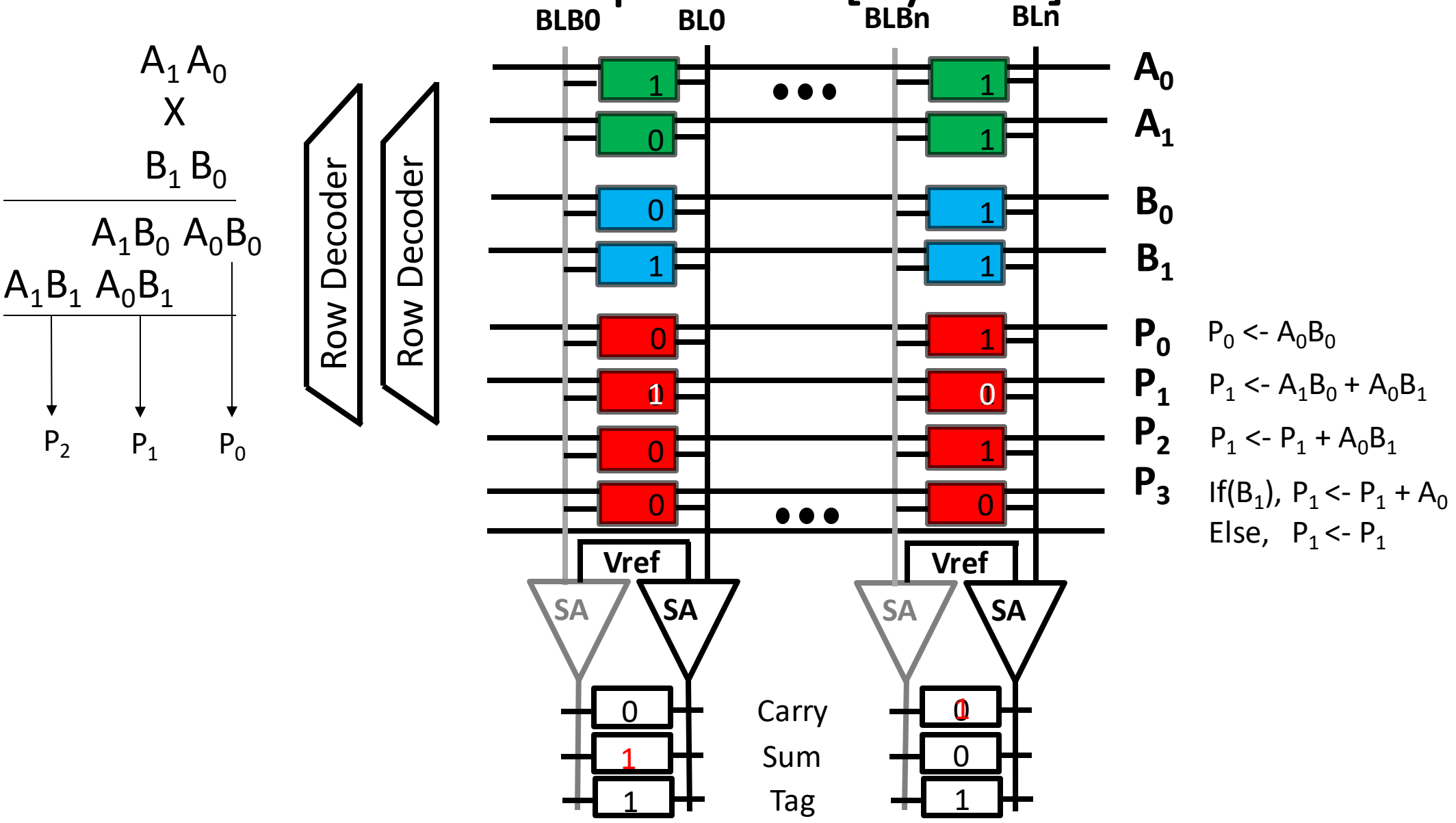
Multiplication [Cycle 3]



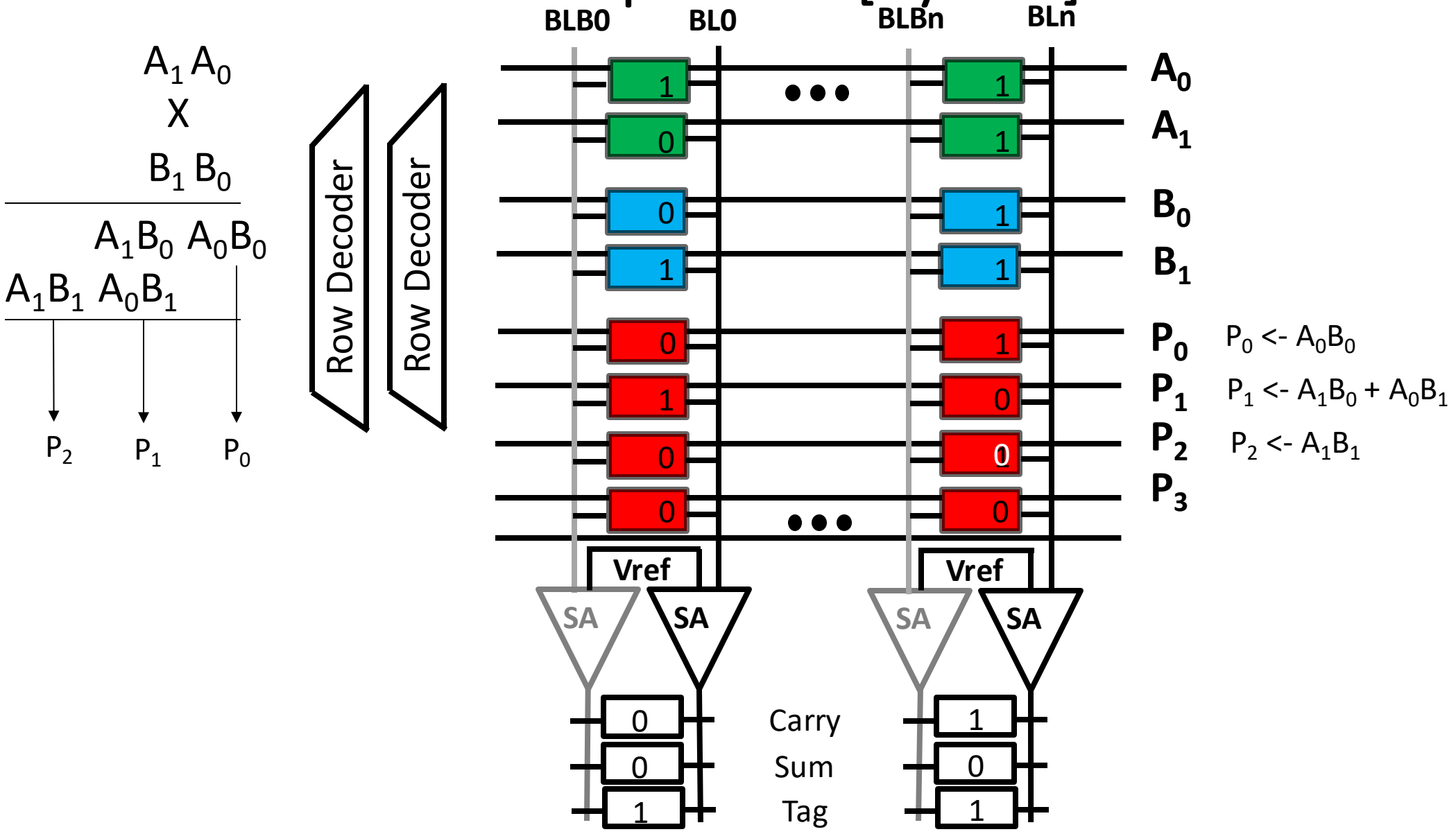
Multiplication [Cycle 4]



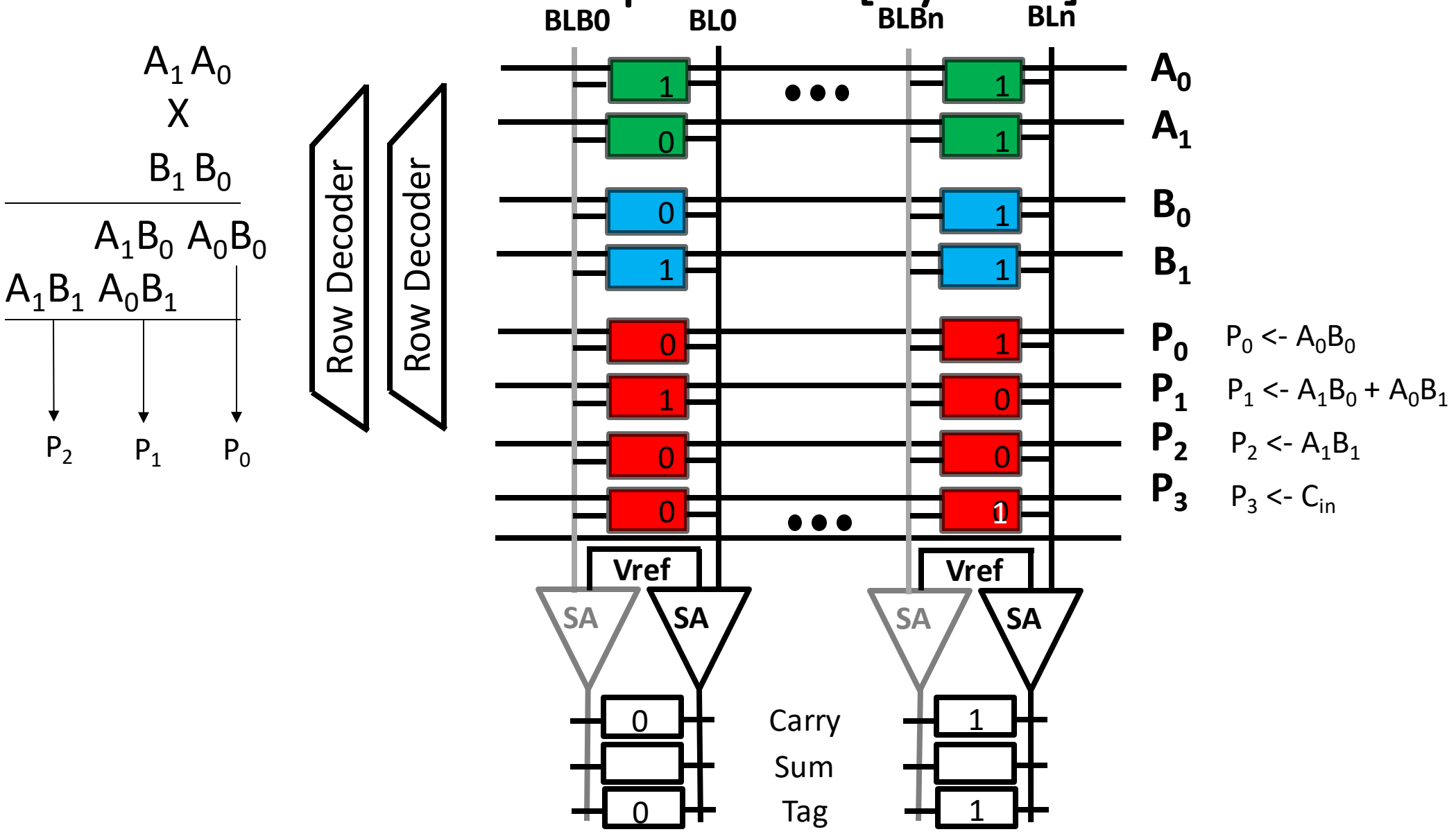
Multiplication [Cycle 5]



Multiplication [Cycle 6]

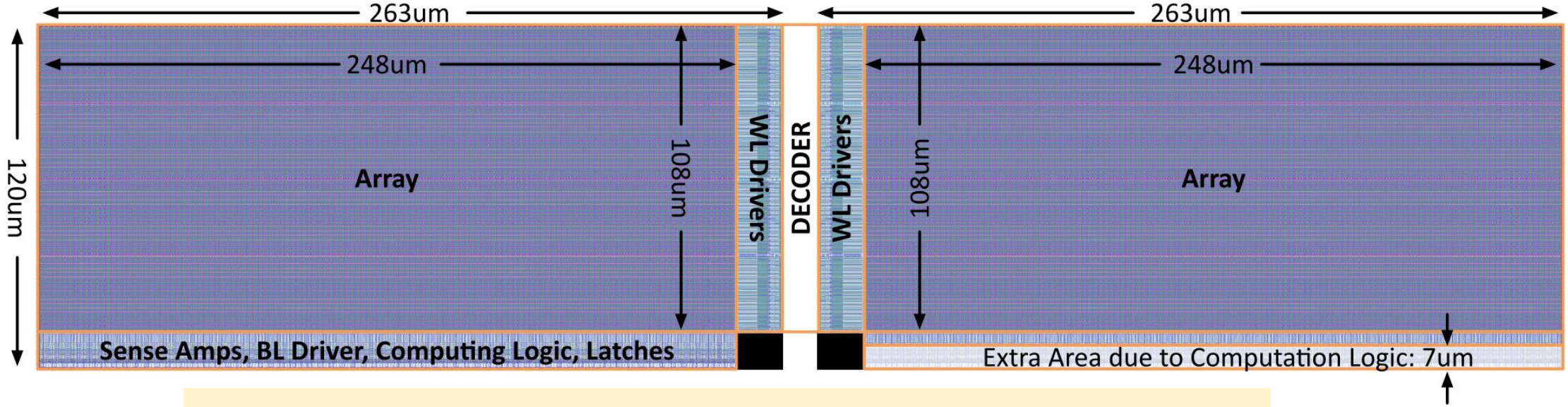


Multiplication [Cycle 7]



Supported Arithmetic

Operation	Cycles
ADD	N+1
SUB	2N+1
MUL	$N^2 + 5N - 2$
DIV	$1.5N^2 + 5.5N$
Comparison	2N+1



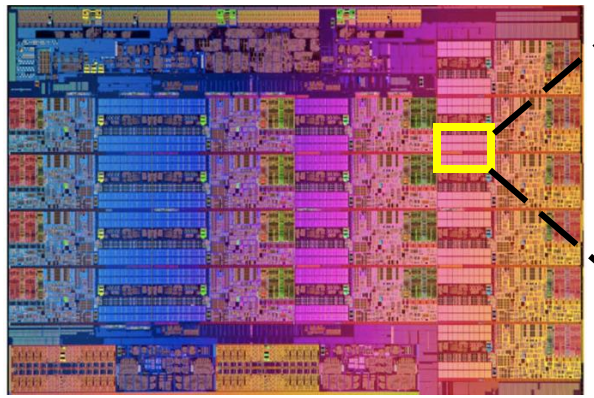
Synthesized array—7.5% area overhead

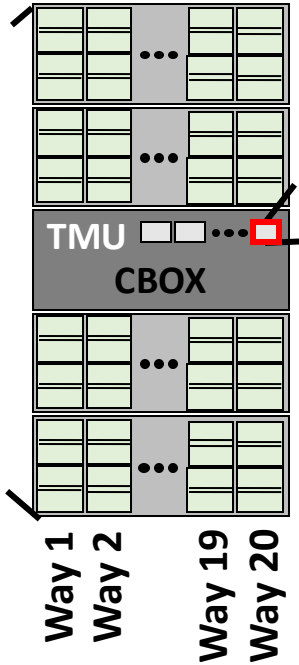
Processor Chip— 2% area overhead

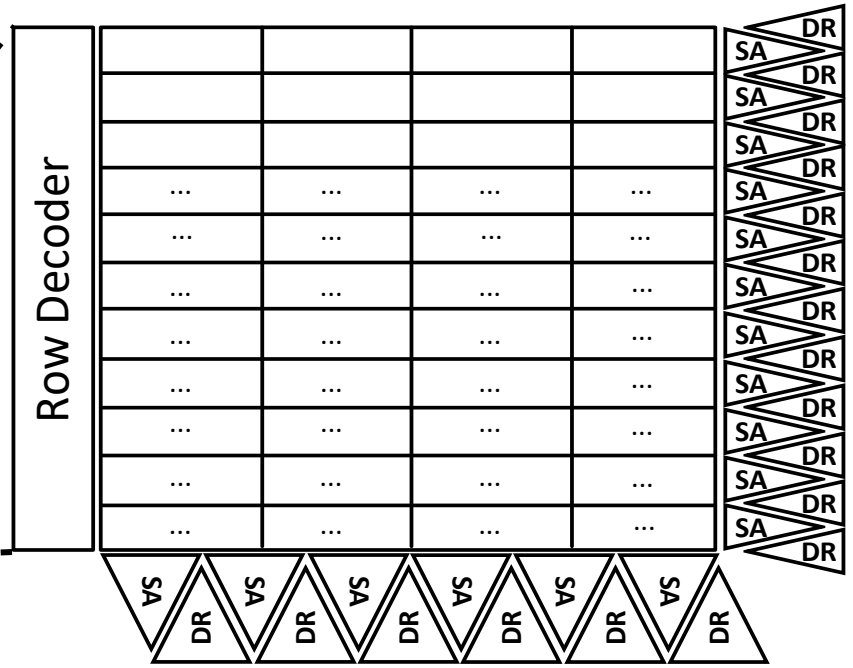
Outline

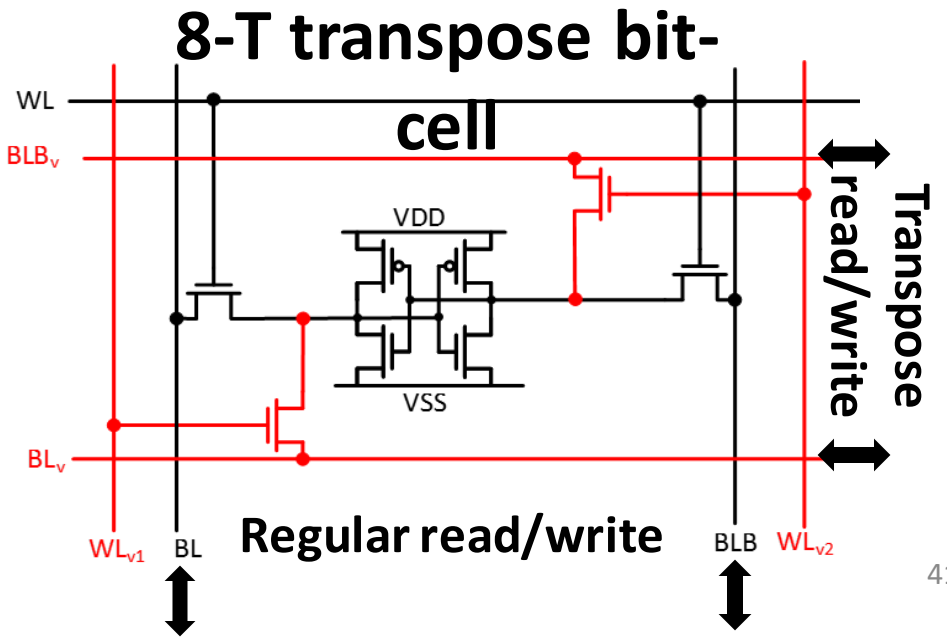
- Motivation
- Bit-Serial Arithmetic
- **Transpose**
- Mapping of Convolution to Array
- Methodology
- Results

Transpose

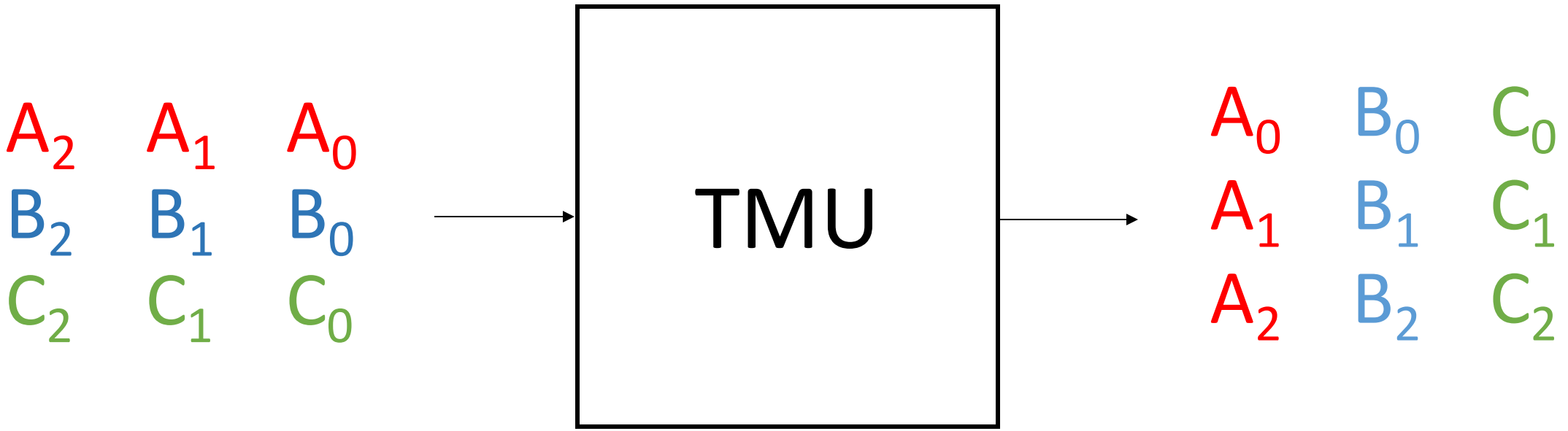



TMU
CBOX
Way 1
Way 2
Way 19
Way 20


Row Decoder
SA
DR


8-T transpose bit-cell
WL
BLB$_v$
VDD
VSS
BL$_v$
WL$_{v1}$
BL
Regular read/write
BLB
WL$_{v2}$
Transpose read/write

Transpose



Outline

- Motivation
- Transpose
- Bit-Serial Arithmetic
- **Mapping of Convolution to Array**
- Methodology
- Results

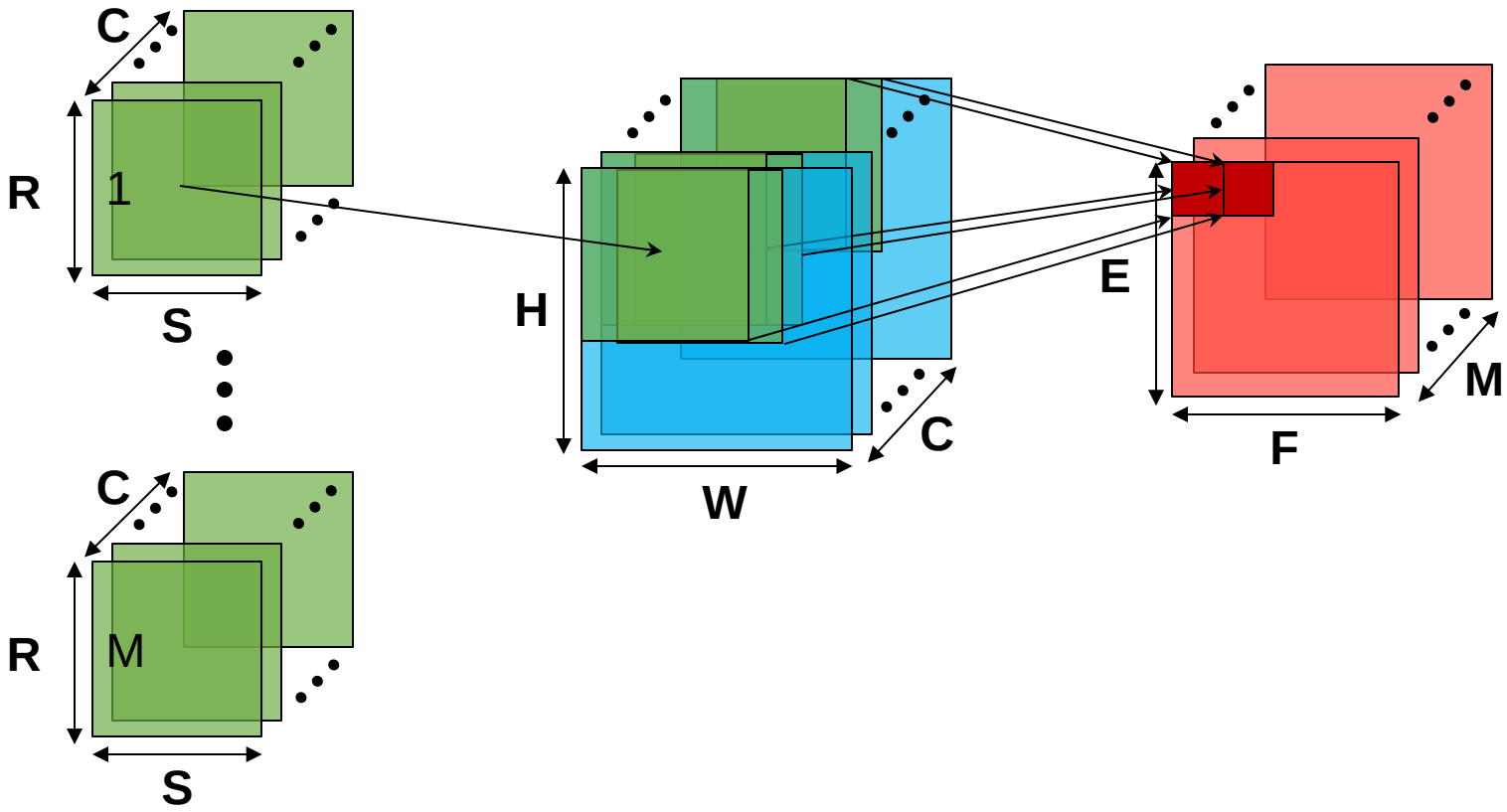
A Convolutional Layer

3D Filters (M)

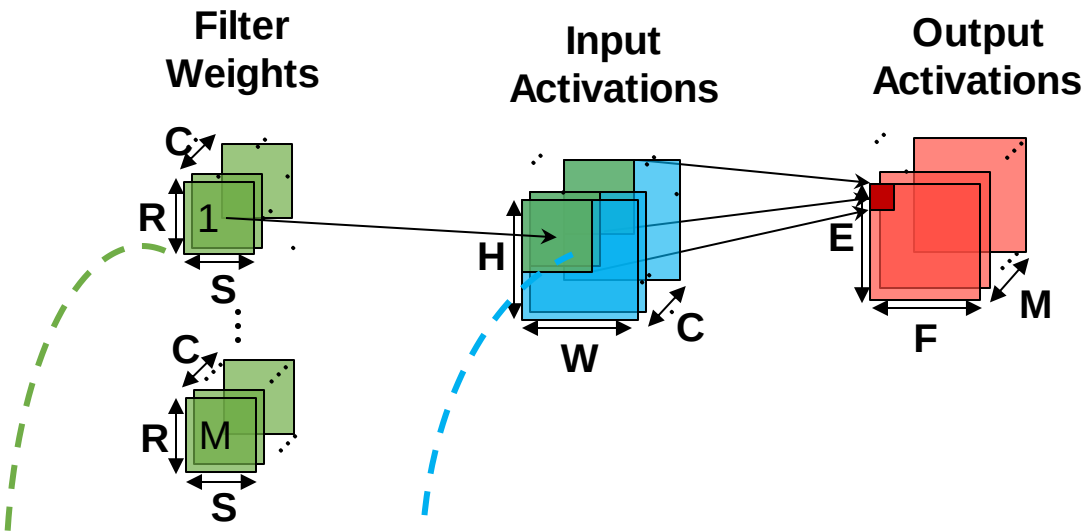
each filter: C channels
each channel: RxS weights

Input Activations
(C channels)

Output Activations
(M channels)

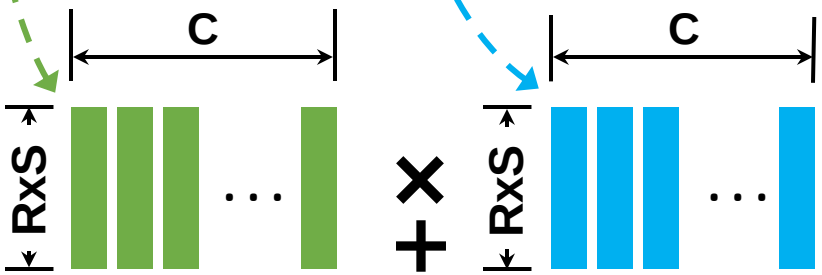


Mapping CNN to Neural Cache

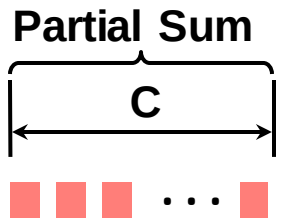


Unroll

Unroll

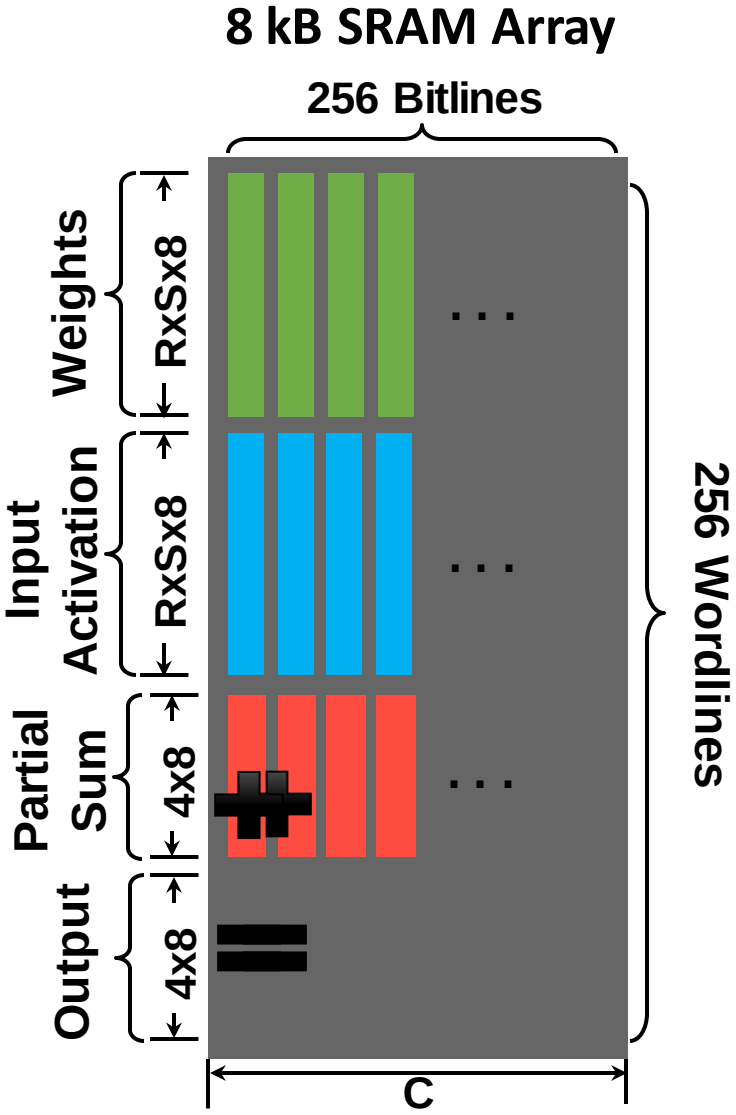


MAC

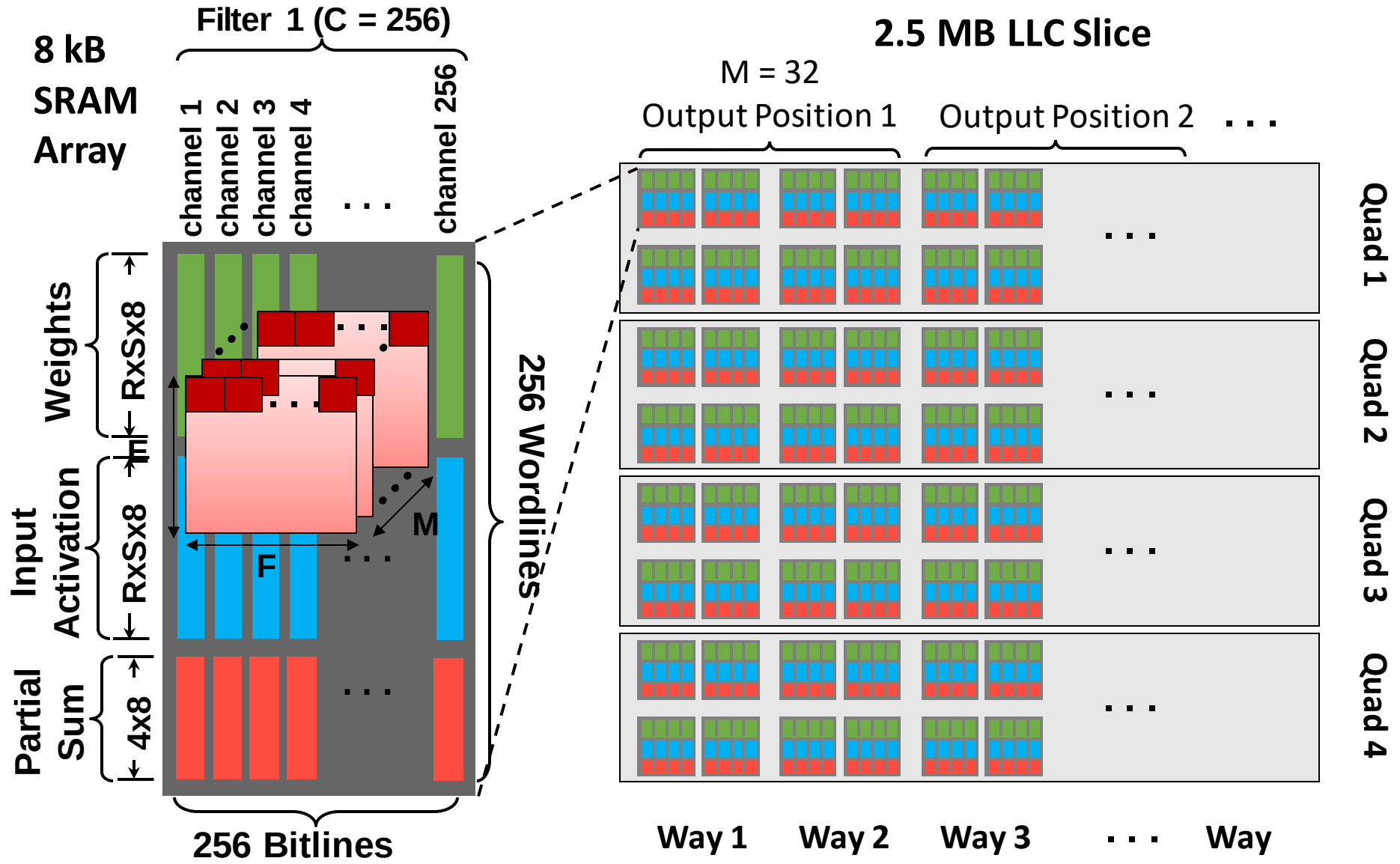


1 Output Activation

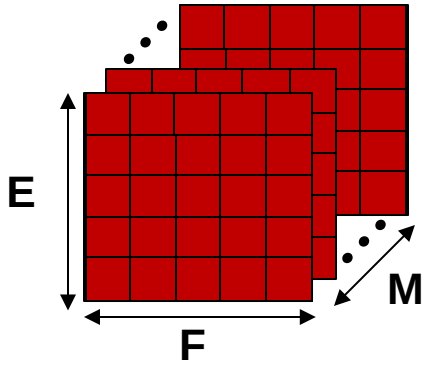
Reduction

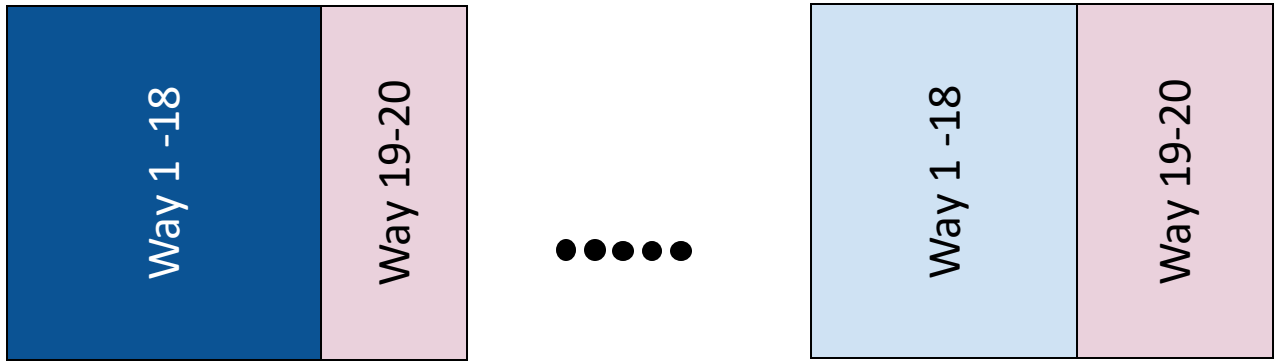


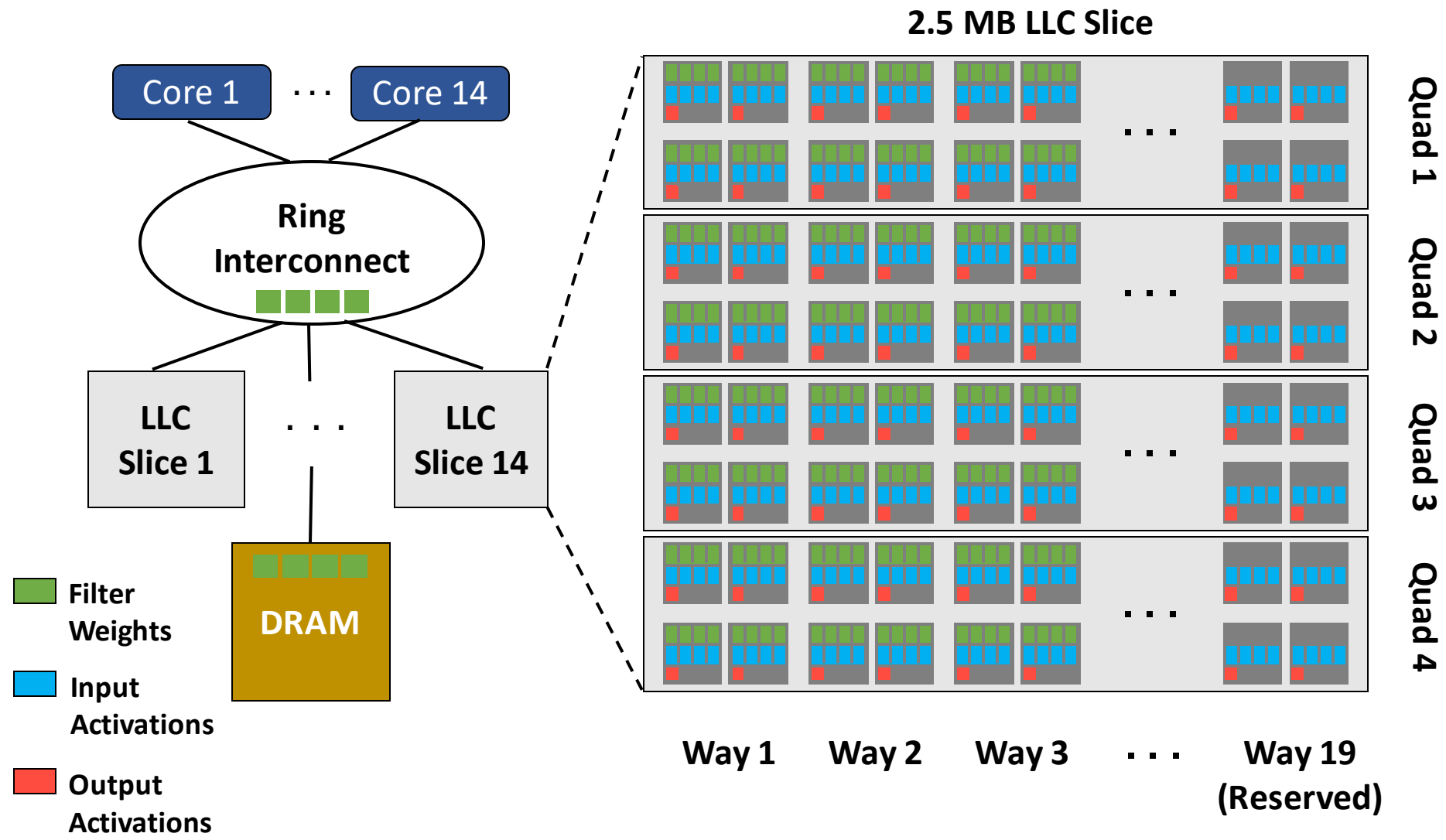
Mapping CNN to Neural Cache



Mapping of Convolution to Array






2.5 MB LLC Slice
Core 1
Core 14
Ring
Interconnect
LLC
Slice 1
LLC
Slice 14
DRAM
Filter
Weights
Input
Activations
Output
Activations
Quad 1
Quad 2
Quad 3
Quad 4
Way 1
Way 2
Way 3
Way 19
(Reserved)

Outline

- Motivation
- Transpose
- Bit-Serial Arithmetic
- Mapping of Convolution to Array
- **Methodology**
- Results

Evaluation Methodology

DNN Models

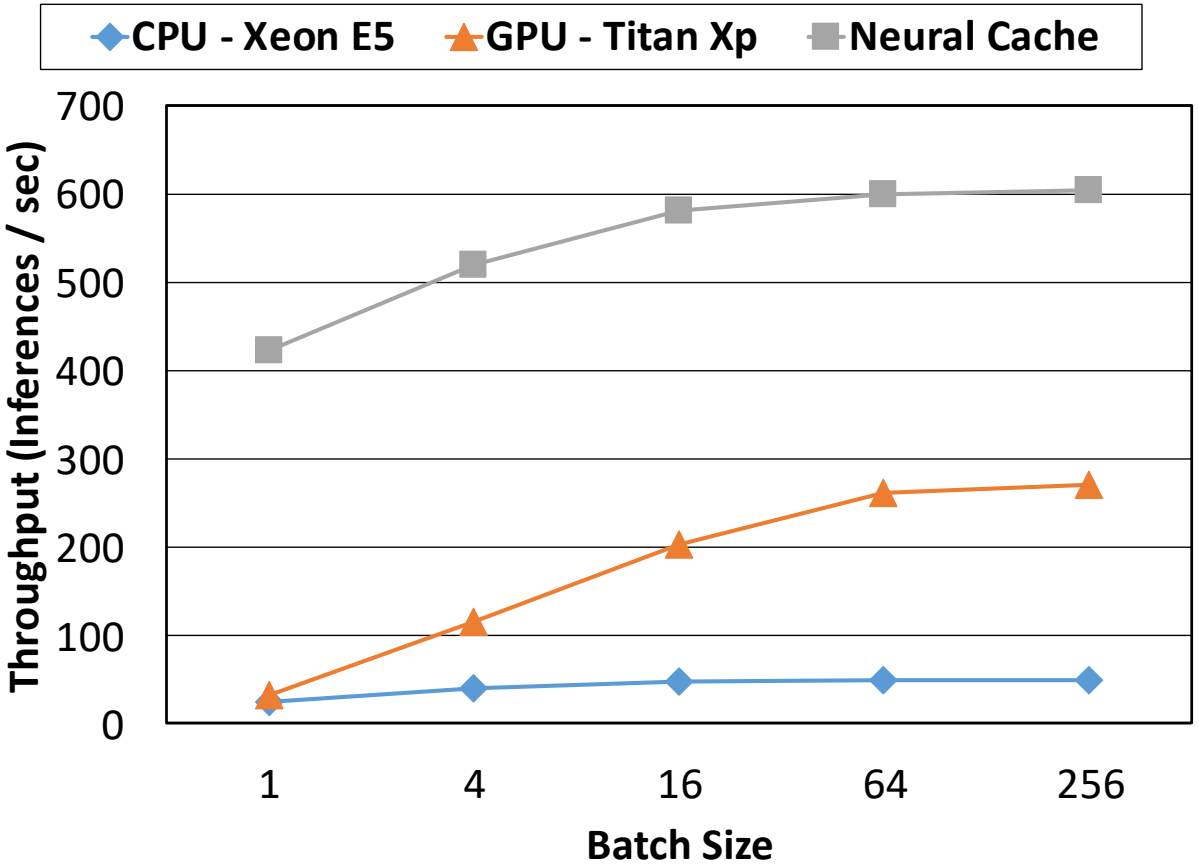
- Inception V3
- 8-bit weights and inputs

	CPU (2 sockets)	GPU (1 card)	Neural Cache
Processor	Intel Xeon E5-2597 v3, 2.6GHz, 28 cores, 56 threads	Nvidia Titan Xp, 1.6GHz, 3840 cuda cores	2.5GHz Compute SRAM, 1032192 Bit-serial ALUs
On-chip memory	78.96 MB	9.14 MB	70 MB (Dual Socket)
Off-chip memory	64 GB DRAM	12 GB DRAM	64 GB DRAM
Profiler / Simulator (Performance)	TensorFlow tfprof	TensorFlow tfprof	Cycle accurate simulator + C Microbench
Profiler / Simulator (Energy)	Intel RAPL Interface	NVIDIA System Management Interface	SPICE simulation + Intel RAPL Interface

Outline

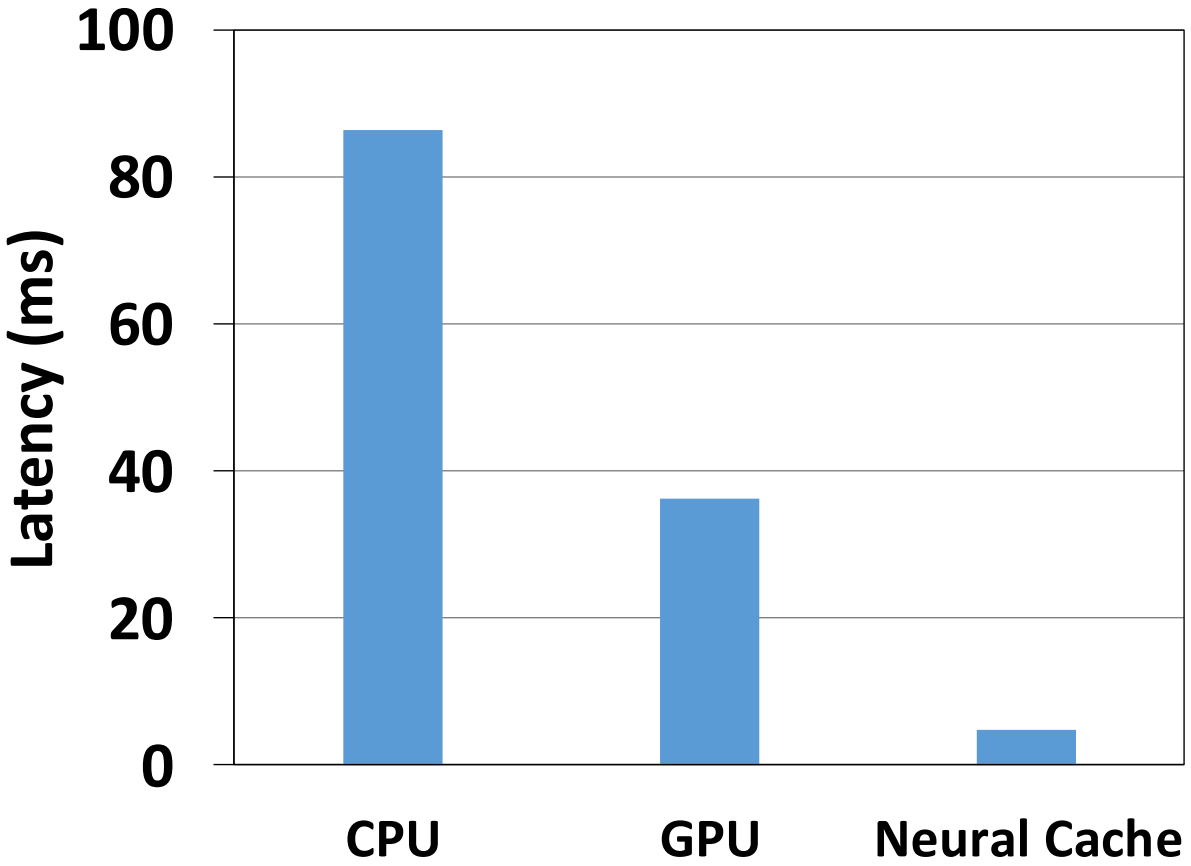
- Motivation
- Transpose
- Bit-Serial Arithmetic
- Mapping of Convolution to Array
- Methodology
- **Results**

Throughput



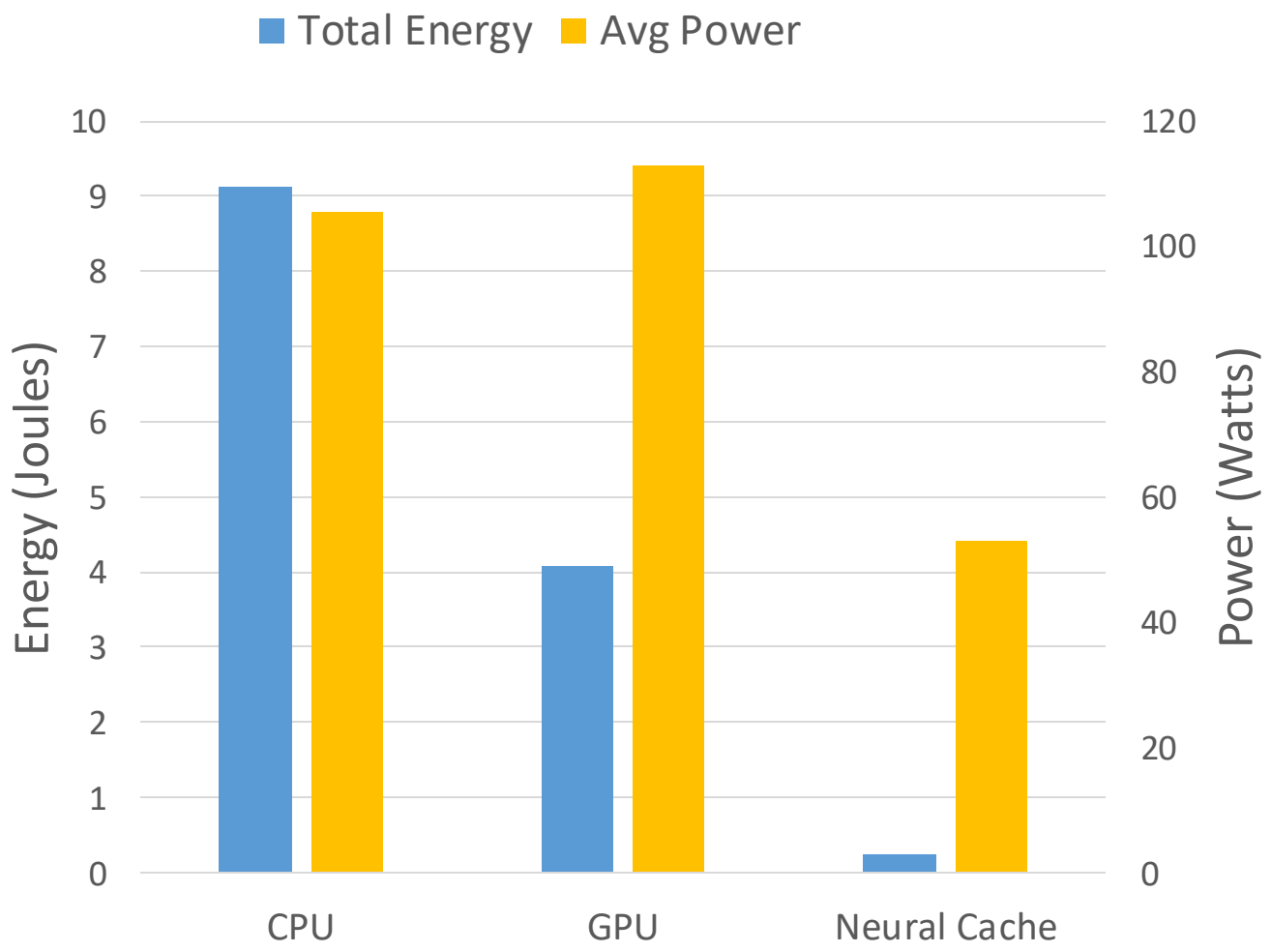
2.2x Improved throughput over GPU

Latency



7.7x Latency improvement over GPU

Power/Energy Comparison



Neural Cache Summary

Repurpose Cache to Data Parallel DNN Accelerator

Massively Parallel Bit-Serial In-SRAM Arithmetic Data Layout for CNNs

12x

20x

.. over server class CPU at **2% area overhead**

2x

16x

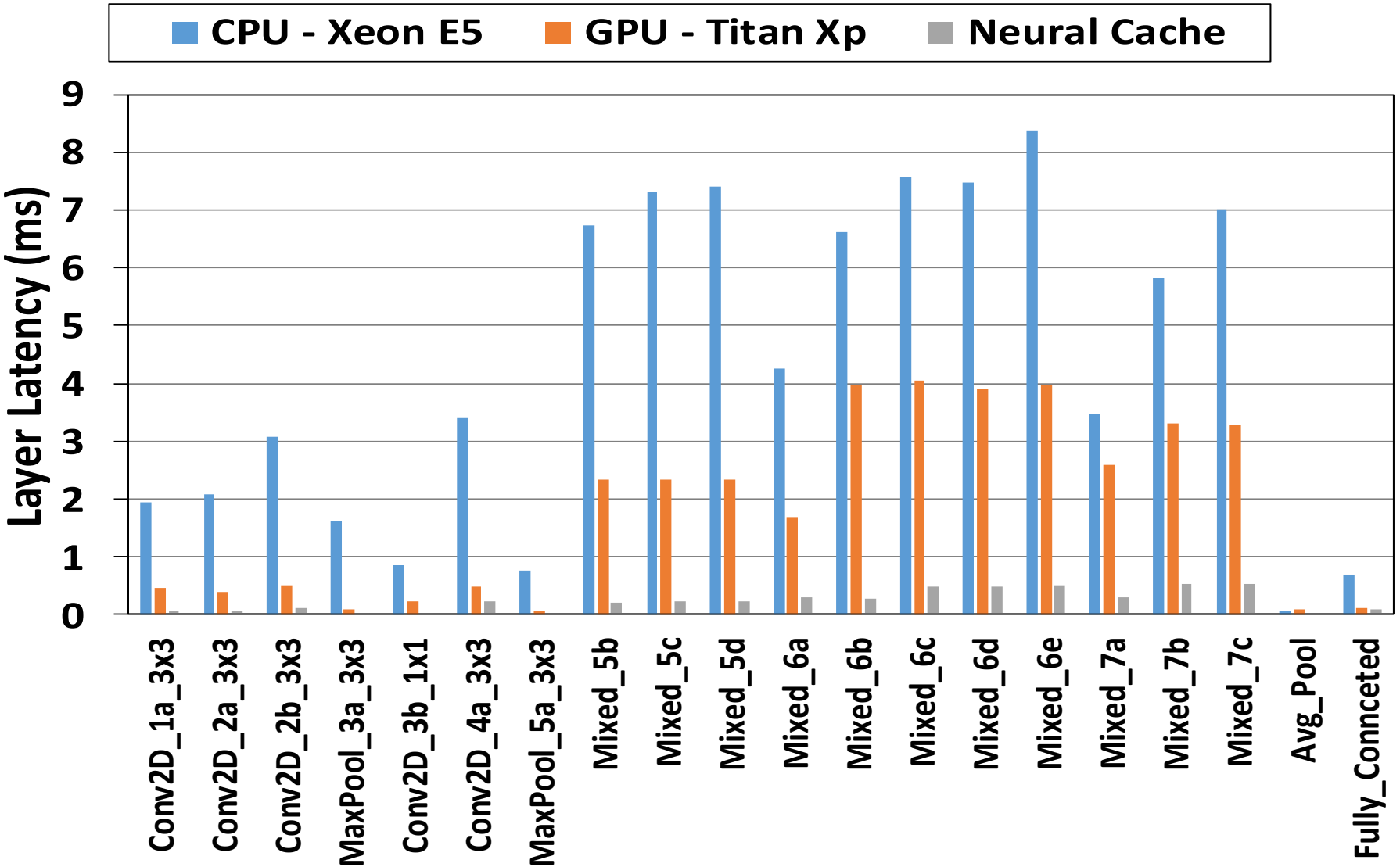
.. over server class GPU

Neural Cache: Bit-Serial In-Cache Acceleration of Deep Neural Networks

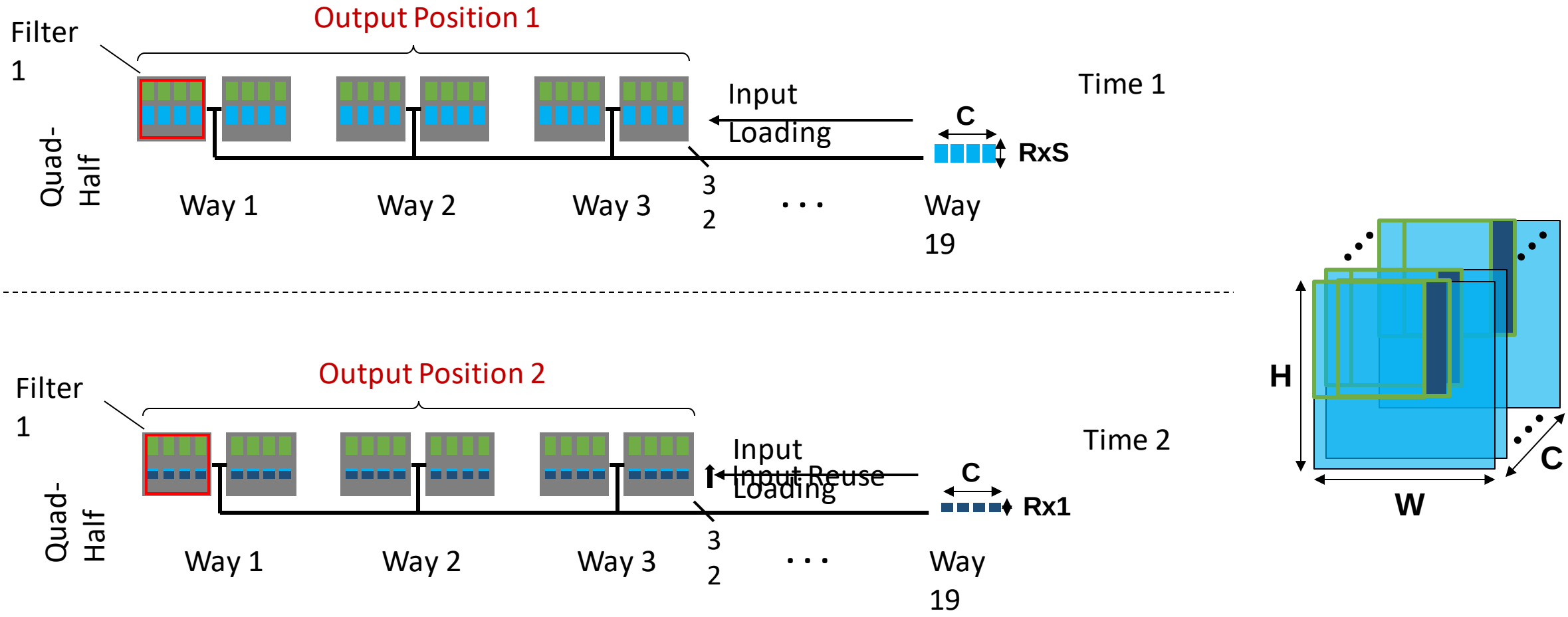
Charles Eckert Xiaowei Wang Jingcheng Wang Arun Subramaniyan
Ravi Iyer Dennis Sylvester David Blaauw Reetuparna Das

M-Bits Research Group
UNIVERSITY OF MICHIGAN

Latency Breakdown

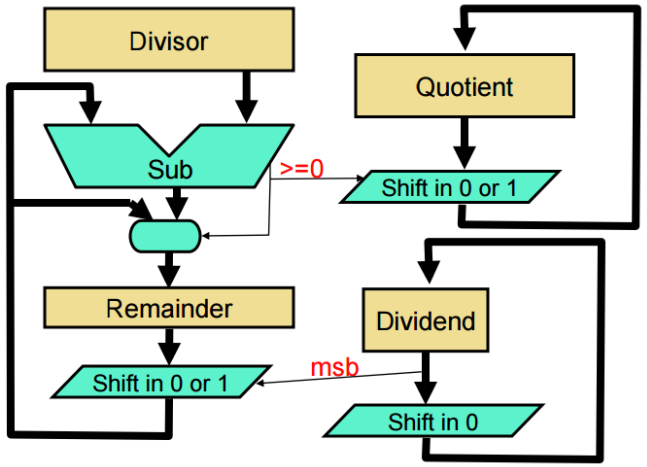


Reusing Inputs across Filter Slides

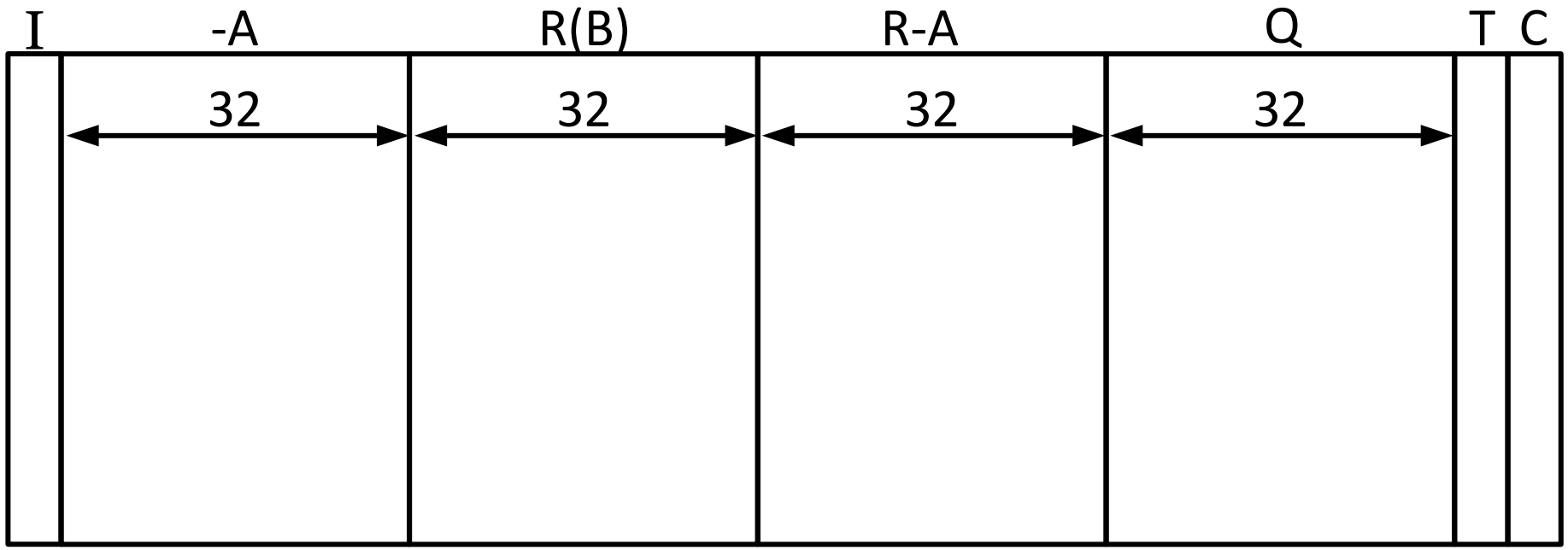


Div Algorithm 1 B/A = Q … R

1. Invert A (n cycles)
2. Initialize Q to all 0 (n cycles)



Extra Hardware Required:
Immediate Column



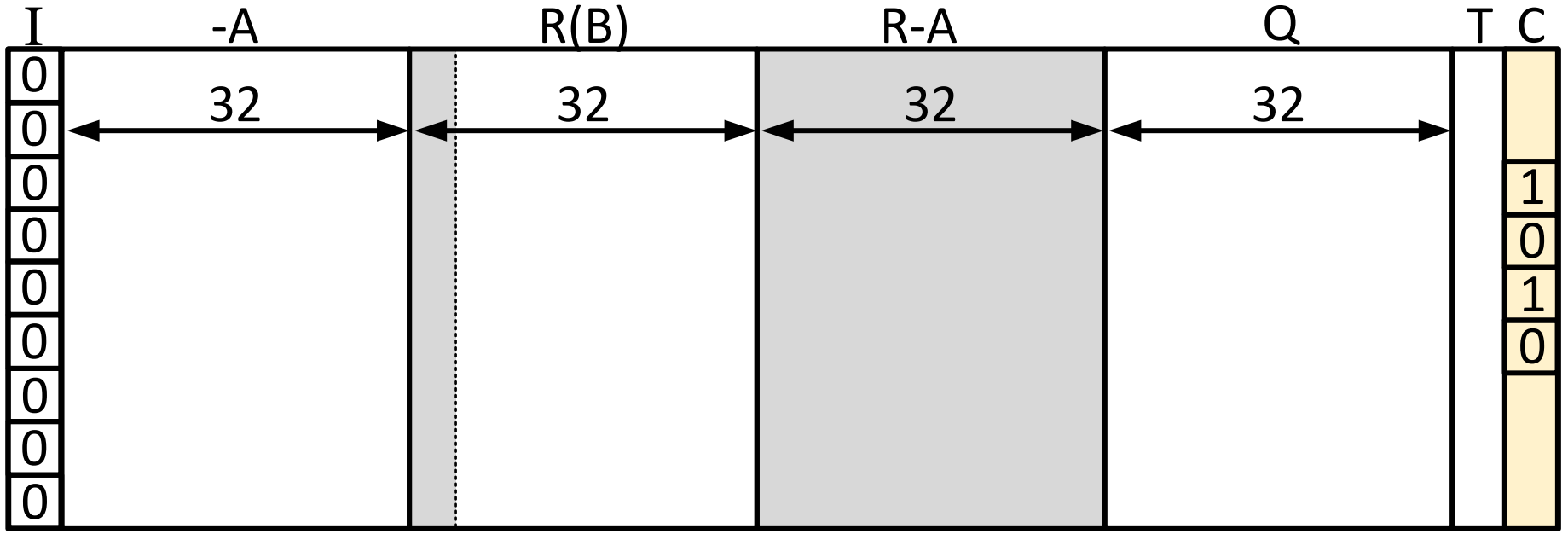
Div Algorithm 2 B/A = Q … R

for i in range (n)

3. Set C to 1 (1 cycle)
4. ADD $R_{n-1:n-1-i}$, $-A_{i:0}$, C → $R\text{-}A_{i:0}$
5. ADDI $-A_{n:i+1}$, 0, C → $R\text{-}A_{n:i+1}$ (n cycles)

   (if C_{out} is 1, R-A positive → R>=A

   if C_{out} is 0, R-A negative → R<A)



Div Algorithm 3 B/A = Q … R

6. Copy C → T (1)
7. Copy C → Q (1)
8. If (Tag==1) (R≥A), R-$A_{i:0}$ → $R_{n-1:n-1-i}$ (i cycle)

Cycles: $n+n+(1+n+1+1)*n + (1+n)*n/2 = 1.5n^2+5.5n$

1712 cycles when n=32