Flash-Cosmos

In-Flash Bulk Bitwise Operations Using Inherent Computation Capability of NAND Flash Memory

Jisung Park, Roknoddin Azizi, Geraldo F. Oliveira, Mohammad Sadrosadati,
Rakesh Nadig, David Novo, Juan Gómez Luna, Myungsuk Kim, and Onur Mutlu

MICRO 2022

Executive Summary

- Background: Bulk bitwise operations are widely used in many important data-intensive applications, e.g., databases, graph processing, cryptography etc.
- Problem:
  - Performance and energy efficiency of bulk bitwise operations are bottlenecked by
    1) data movement between storage and the compute unit in traditional systems and in-storage processing (ISP)
    2) data sensing (serial reading of operands) in prior in-flash processing (IFP) techniques
  - Prior IFP techniques provide low reliability during computation
- Goal: Improve performance, energy efficiency and reliability of bulk bitwise operations in in-flash processing
- Key Ideas: Flash-Cosmos (Flash-Computation with One-Shot Multi-Operand Sensing) is an in-flash processing technique that is based on two key ideas:
  - Multi-Wordline Sensing (MWS): Enables multi-operand bulk bitwise operations with a single sensing (read) operation
  - Enhanced SLC-mode Programming (ESP): Increases the voltage margin between the erased and programmed states to provide higher reliability during in-flash computation
- Key Results: Flash-Cosmos is evaluated using 160 real 3D NAND flash chips and with a state-of-the-art SSD simulator on three real-world workloads
  - Flash-Cosmos improves the performance and energy efficiency by 3.5x and 3.3x over state-of-the-art IFP technique while providing high reliability during computation

Talk Outline

- Problem, Goals & Key Idea

- Background

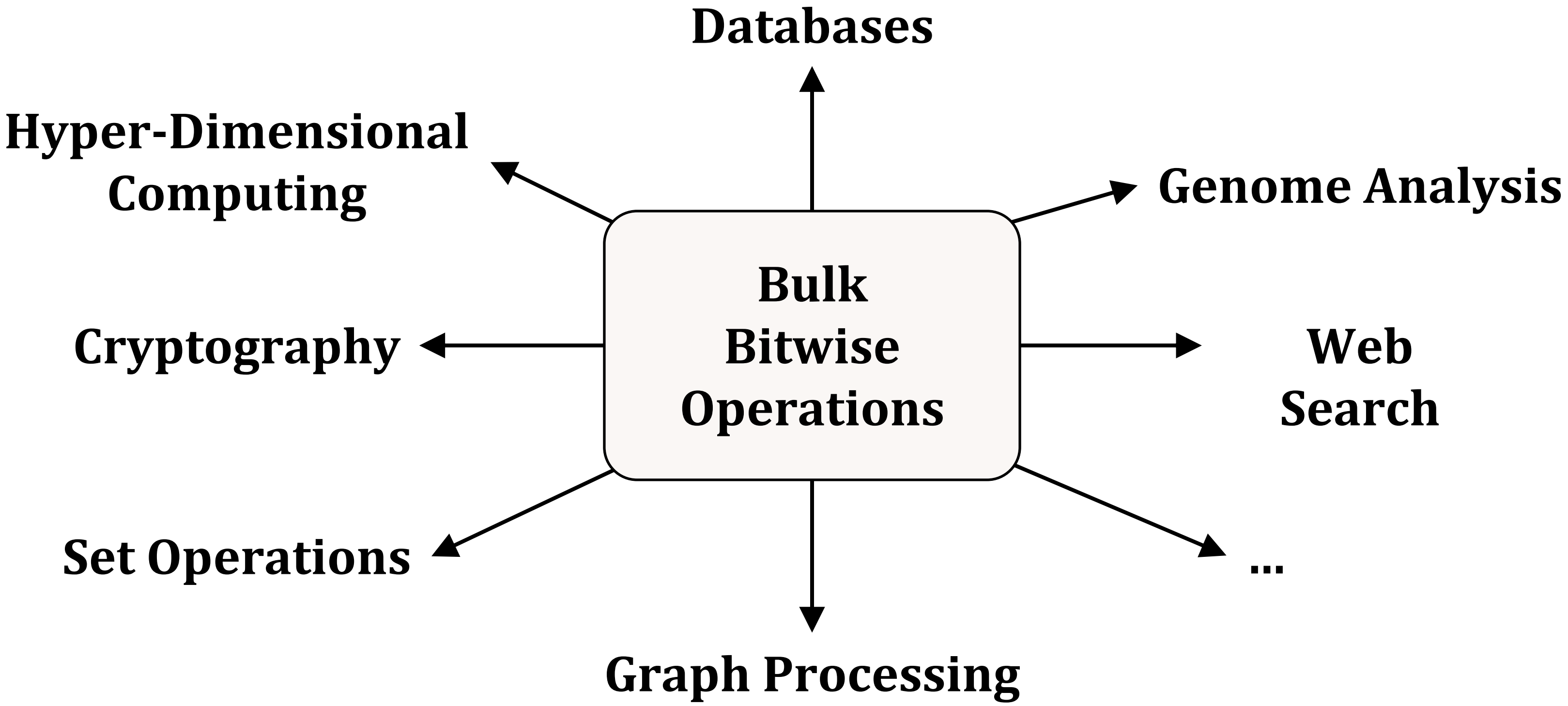
- Flash-Cosmos: Computation with One-Shot Multi-Operand Sensing

- Evaluation

- Summary

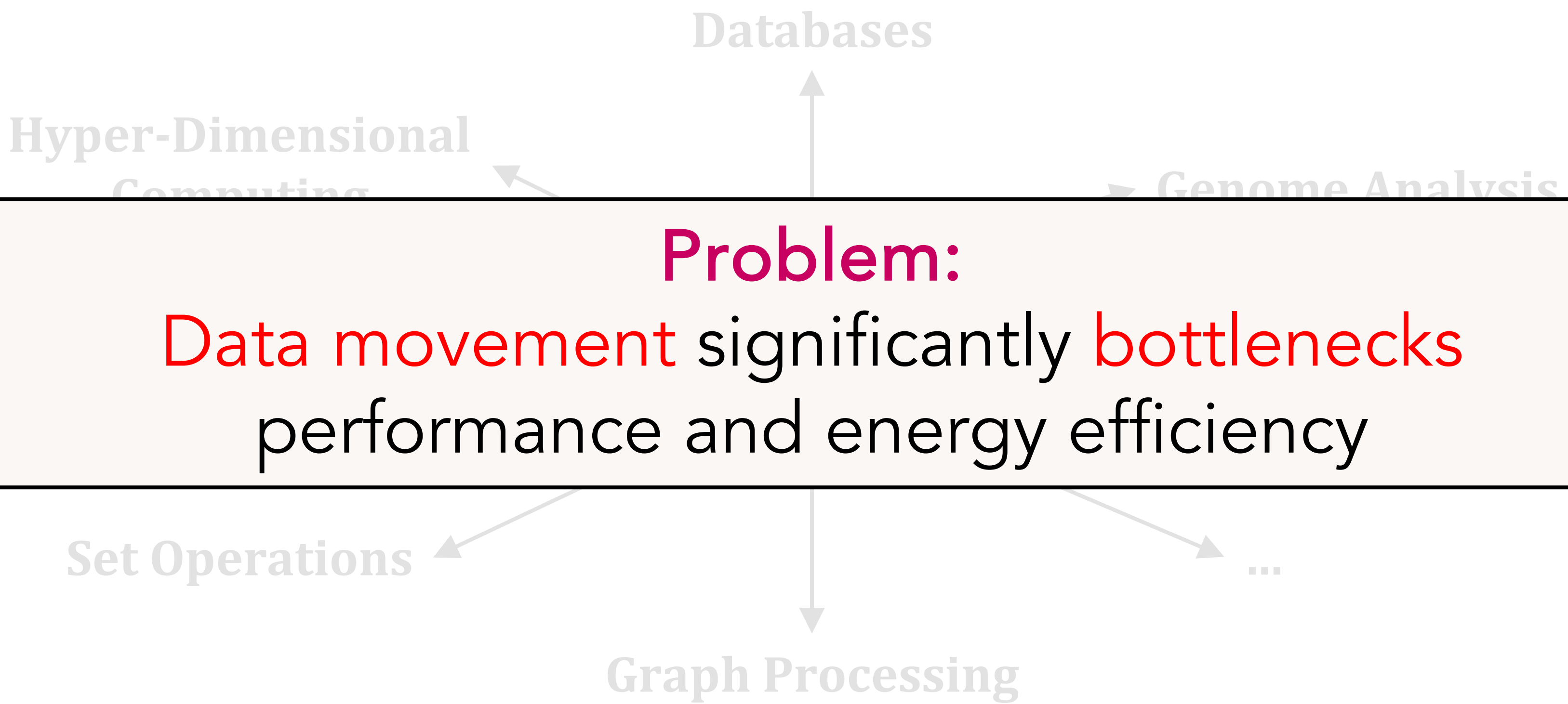
Bulk Bitwise Operations

- Widely-used computation primitive in data-intensive applications



Bulk Bitwise Operations

- Widely-used computation primitive in data-intensive applications



Databases

Hyper-Dimensional Computing

Genome Analysis

Set Operations

...

Graph Processing

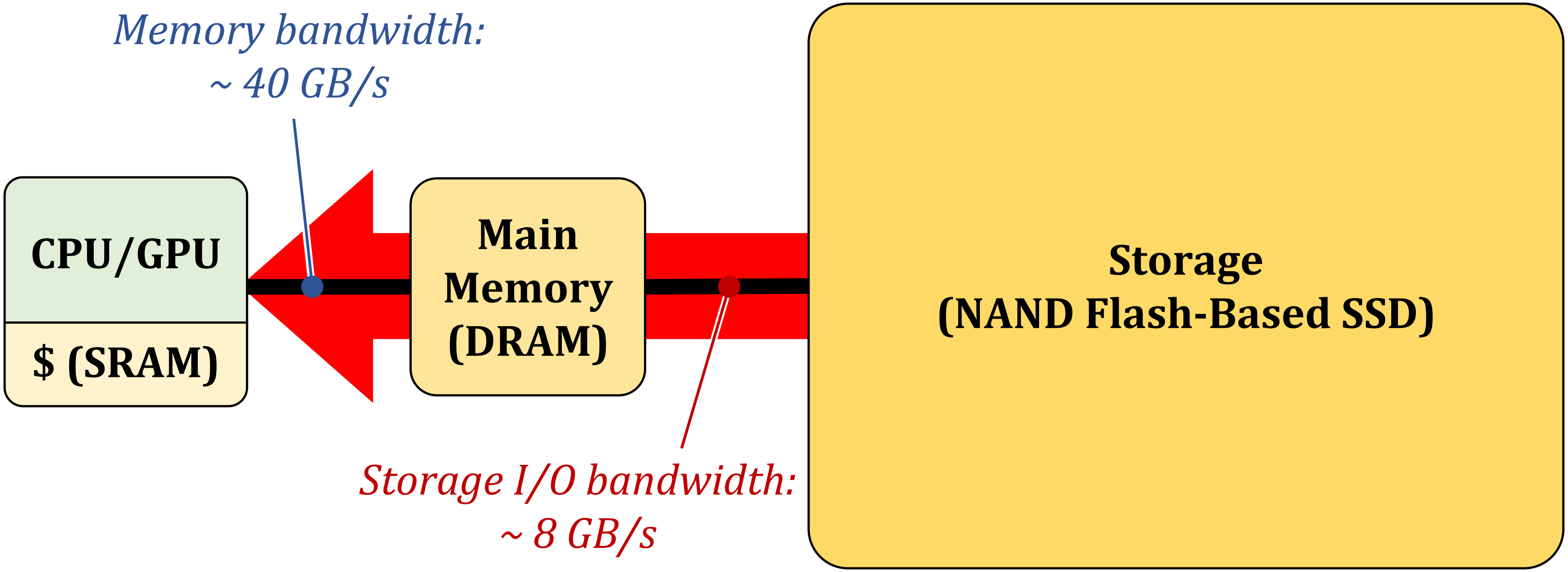
Problem:

Data movement significantly bottlenecks performance and energy efficiency

SAFARI

Data-Movement Bottleneck

- Conventional systems: Outside-storage processing (OSP) that must move the entire data to CPUs/GPUs through the memory hierarchy

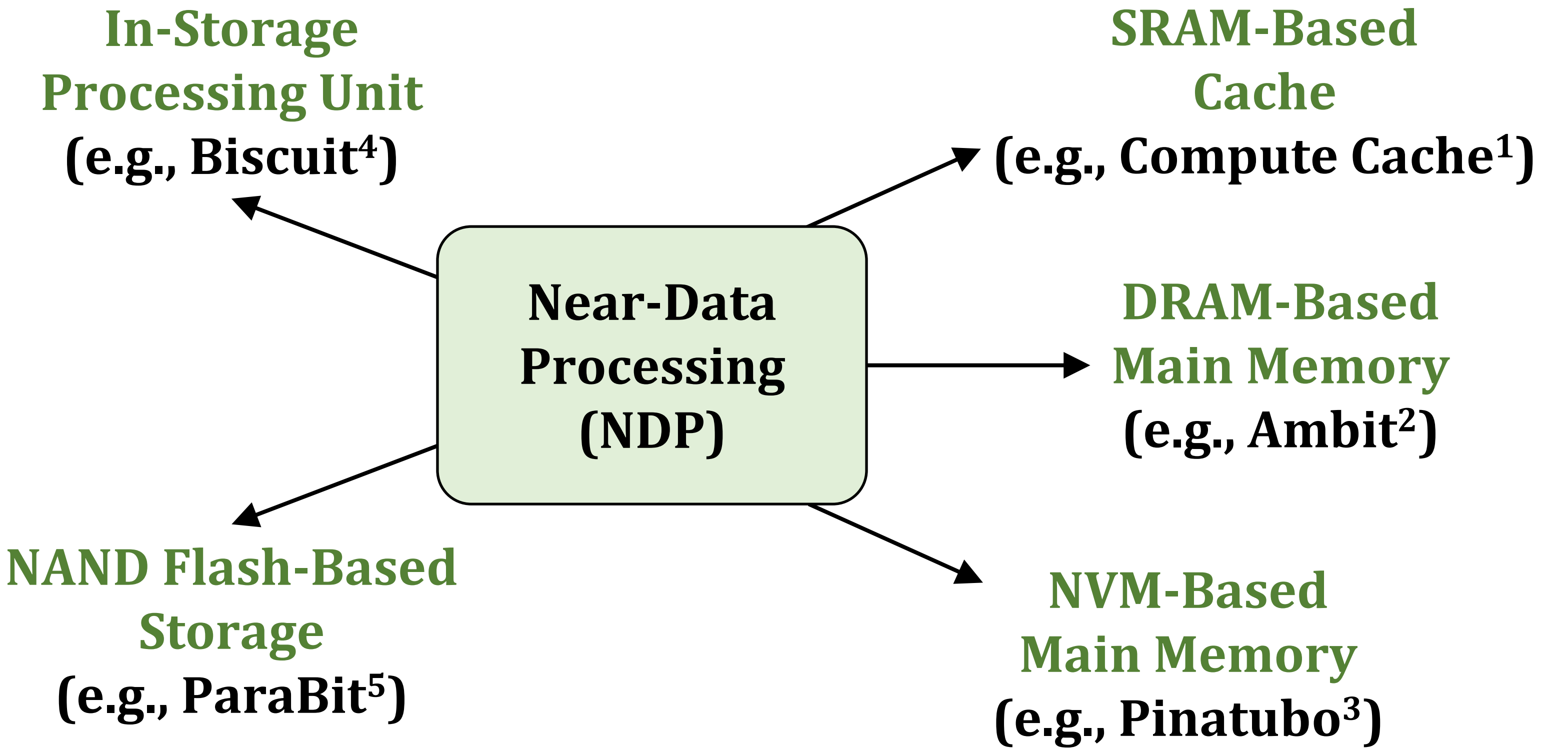


Memory bandwidth: ~ 40 GB/s
CPU/GPU
$ (SRAM)
Main Memory (DRAM)
Storage (NAND Flash-Based SSD)
Storage I/O bandwidth: ~ 8 GB/s

External I/O bandwidth of storage systems is the main bottleneck in conventional systems (OSP)

Near-Data Processing for Bitwise Operations

- Can effectively mitigate data movement by performing simple bitwise operations where the data resides



Near-Data Processing (NDP)

In-Storage Processing Unit (e.g., Biscuit[4])

SRAM-Based Cache (e.g., Compute Cache[1])

DRAM-Based Main Memory (e.g., Ambit[2])

NAND Flash-Based Storage (e.g., ParaBit[5])

NVM-Based Main Memory (e.g., Pinatubo[3])

[1] Aga+, "Compute Caches," HPCA, 2017

[2] Seshadri+, "Ambit: In-Memory Accelerator for Bulk Bitwise Operations Using Commodity DRAM Technology," MICRO, 2017

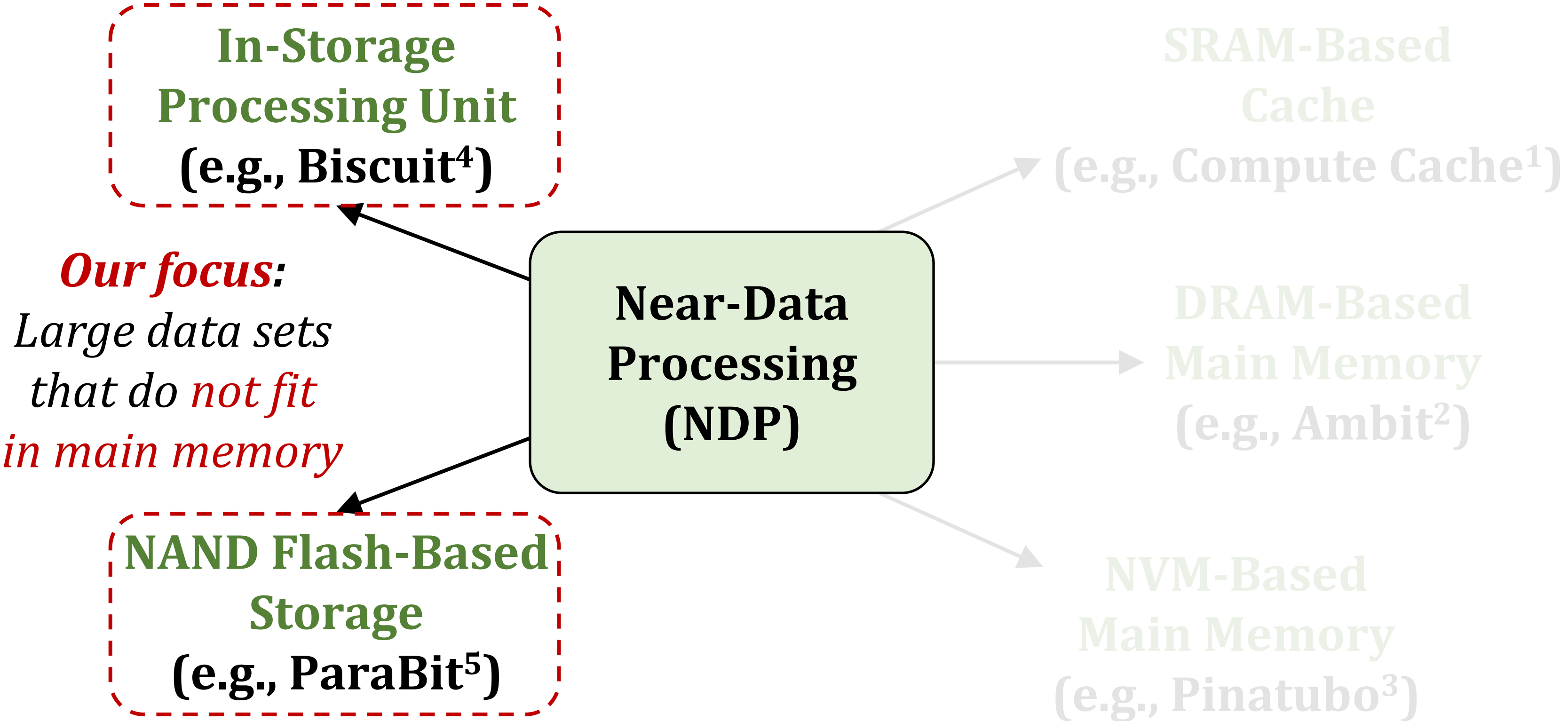
[3] Li+, "Pinatubo: A Processing-in-Memory Architecture for Bulk Bitwise Operations in Emerging Non-Volatile Memories," DAC, 2016

[4] Gu+, "Biscuit: A Framework for Near-Data Processing of Big Data Workloads," ISCA, 2016

[5] Gao+, "ParaBit: Processing Parallel Bitwise Operations in NAND Flash Memory Based SSDs," MICRO, 2021

Near-Data Processing for Bitwise Operations

- Can effectively mitigate data movement by performing simple bitwise operations where the data resides



In-Storage Processing Unit (e.g., Biscuit[4])

Our focus: *Large data sets that do not fit in main memory*

NAND Flash-Based Storage (e.g., ParaBit[5])

Near-Data Processing (NDP)

SRAM-Based Cache (e.g., Compute Cache[1])

DRAM-Based Main Memory (e.g., Ambit[2])

NVM-Based Main Memory (e.g., Pinatubo[3])

[1] Aga+, "Compute Caches," HPCA, 2017

[2] Seshadri+, "Ambit: In-Memory Accelerator for Bulk Bitwise Operations Using Commodity DRAM Technology," MICRO, 2017

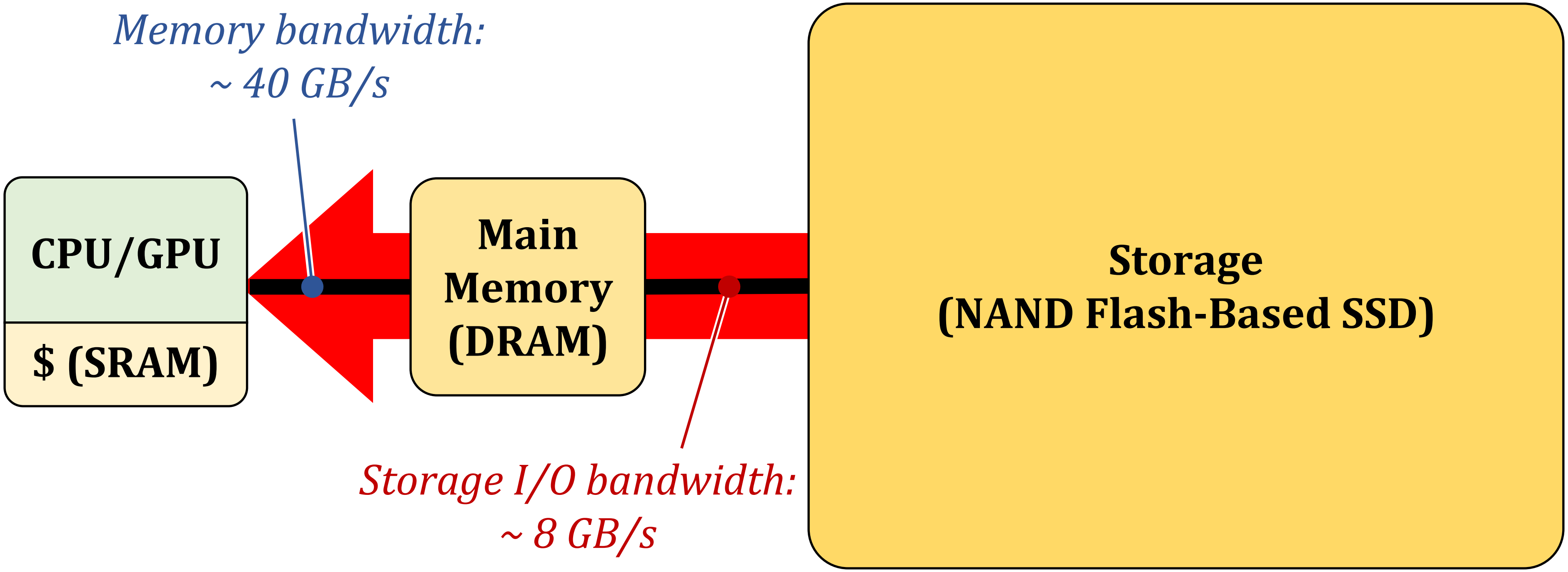
[3] Li+, "Pinatubo: A Processing-in-Memory Architecture for Bulk Bitwise Operations in Emerging Non-Volatile Memories," DAC, 2016

[4] Gu+, "Biscuit: A Framework for Near-Data Processing of Big Data Workloads," ISCA, 2016

[5] Gao+, "ParaBit: Processing Parallel Bitwise Operations in NAND Flash Memory Based SSDs," MICRO, 2021

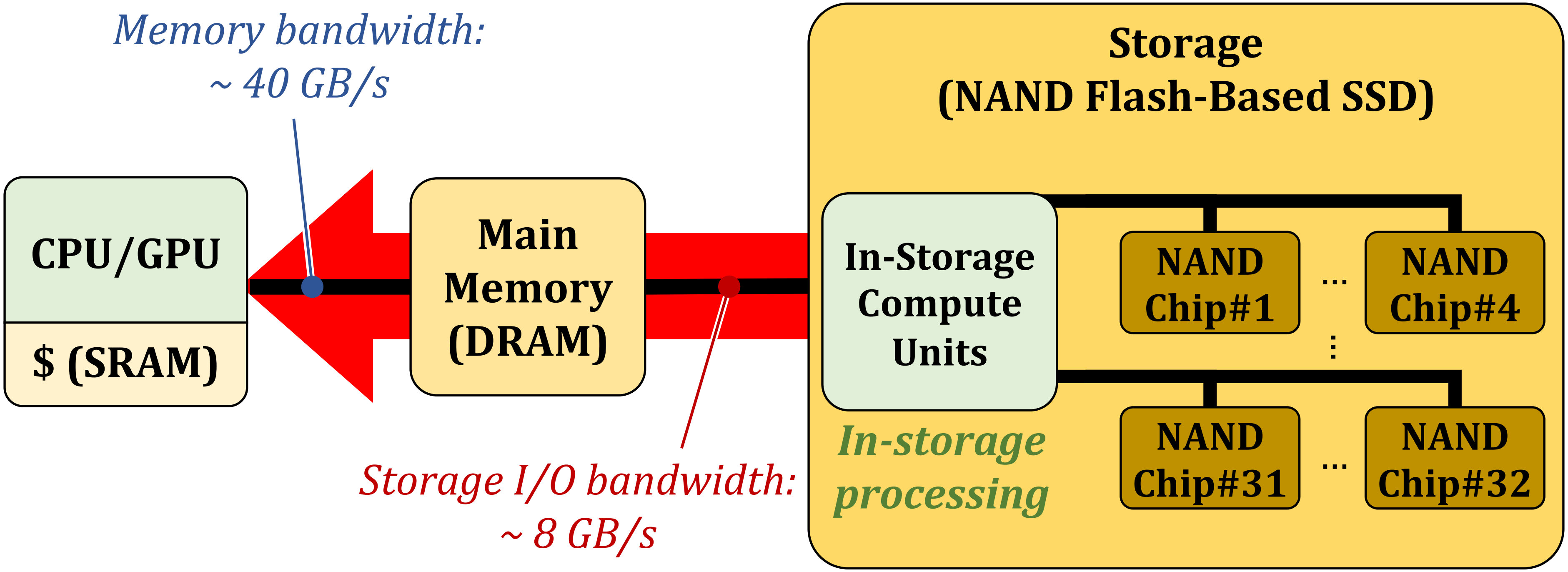
In-Storage Processing (ISP)

- Uses in-storage compute units (embedded cores or FPGA) to send only the computation results



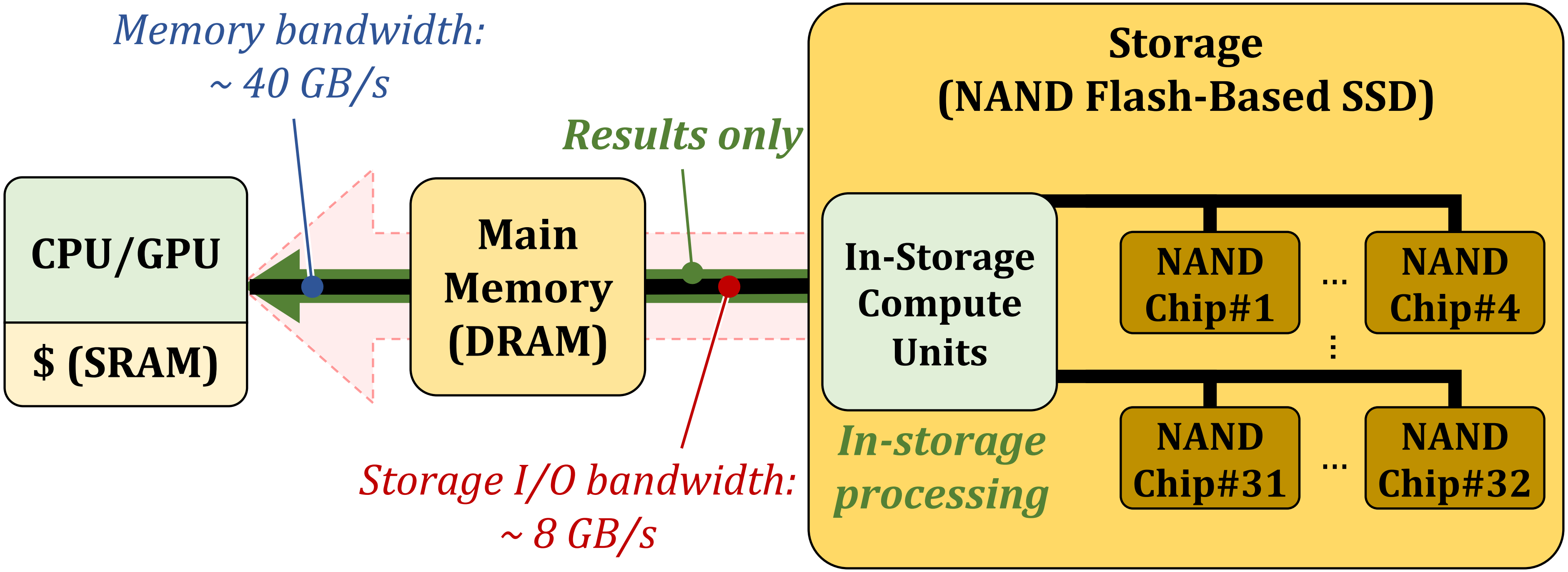
In-Storage Processing (ISP)

- Uses in-storage compute units (embedded cores or FPGA) to send only the computation results



In-Storage Processing (ISP)

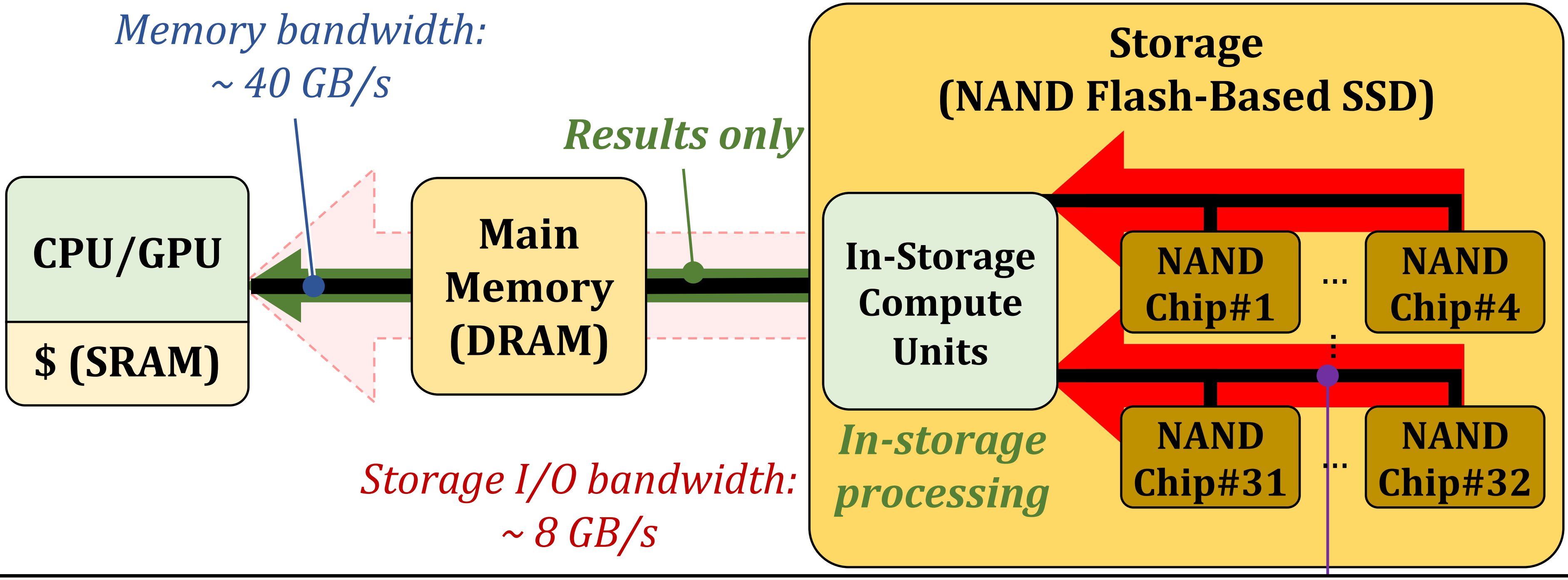
- Uses in-storage compute units (embedded cores or FPGA) to send only the computation results



ISP can mitigate data movement overhead by reducing SSD-external data movement

In-Storage Processing (ISP)

- Uses in-storage compute units (embedded cores or FPGA) to send only the computation results

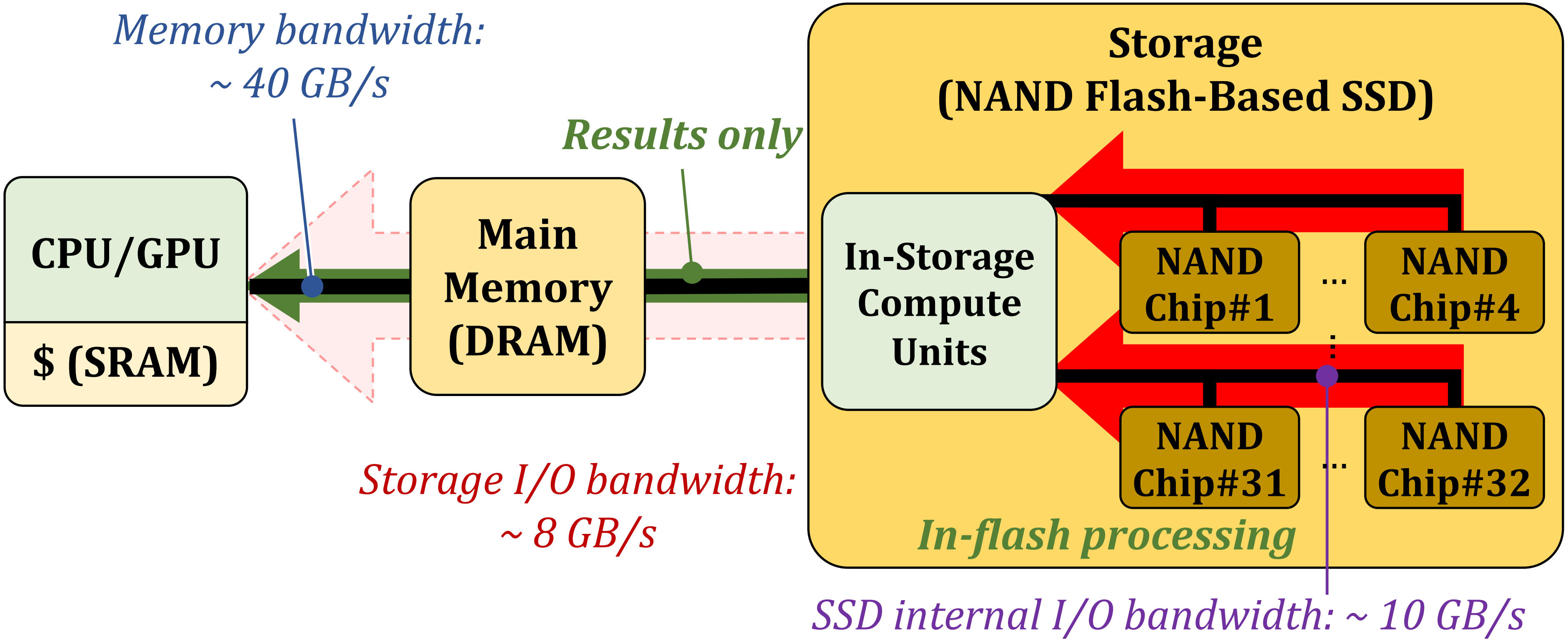


SSD-internal bandwidth
becomes the new bottleneck in ISP

SAFARI

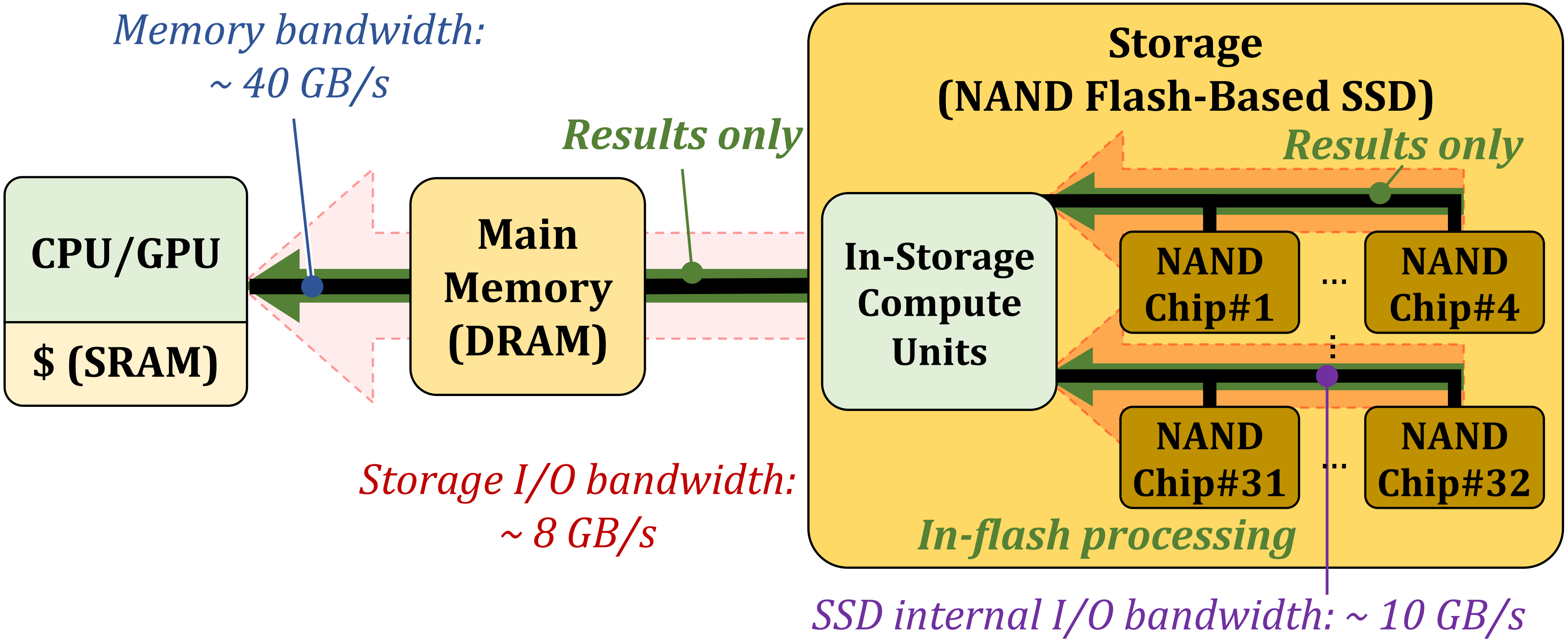
In-Flash Processing (IFP)

- Performs computation inside NAND flash chips



In-Flash Processing (IFP)

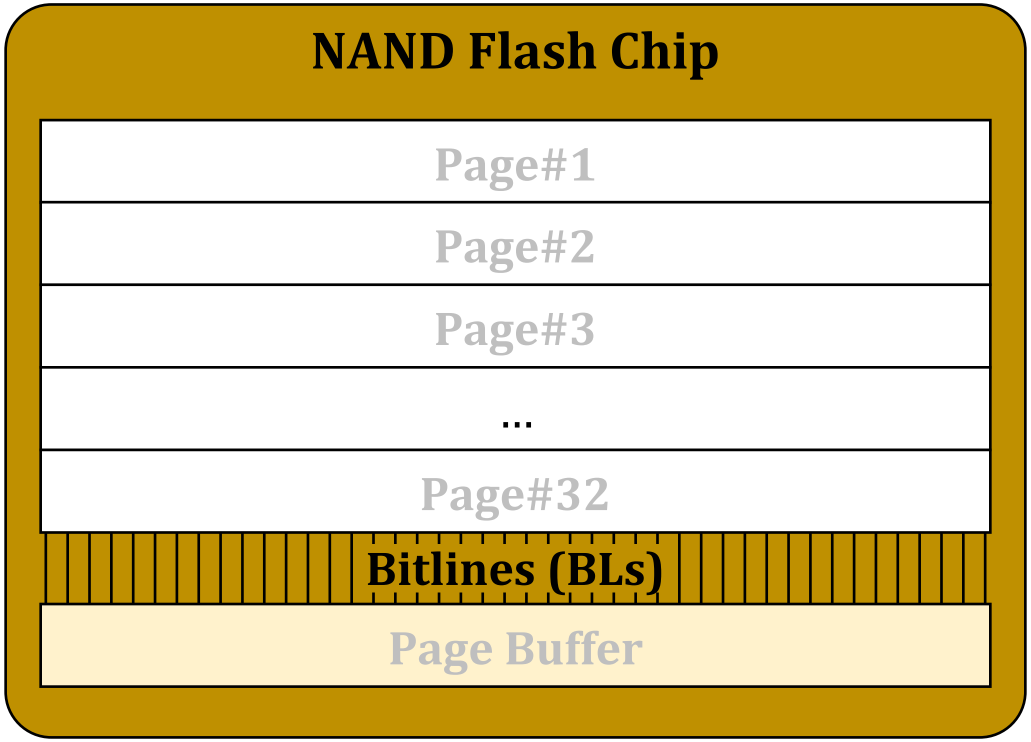
- Performs computation inside NAND flash chips



IFP fundamentally mitigates data movement

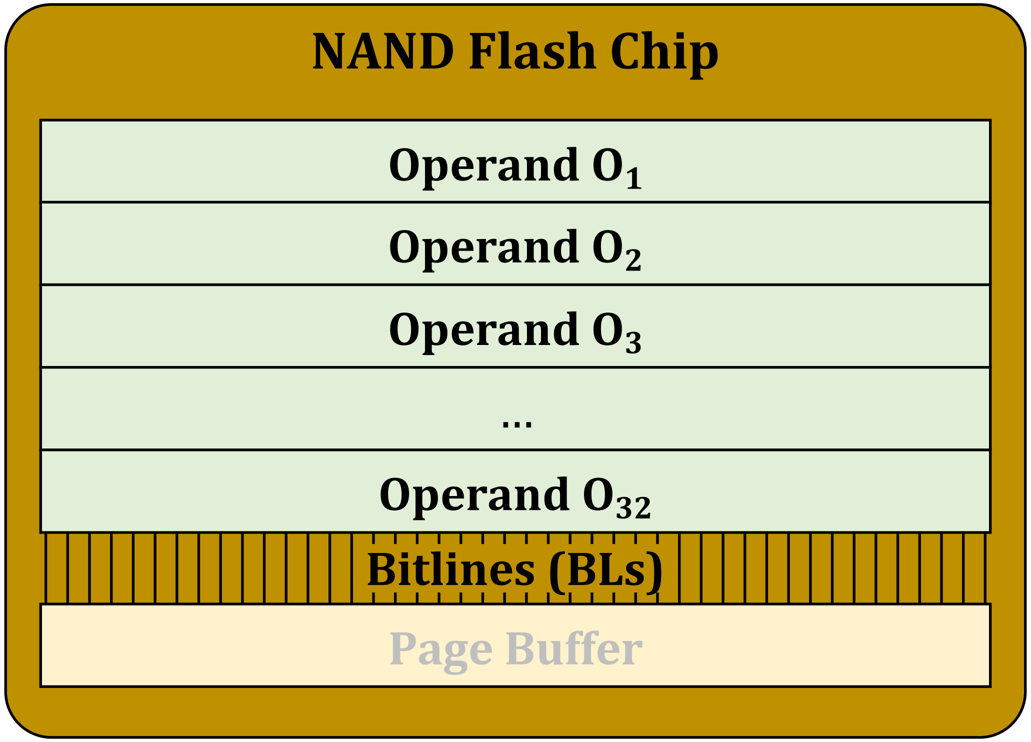
State-of-the-Art IFP Technique (1/2)

- ParaBit [Gao+, MICRO 2021]
  - Performs bulk bitwise operations inside NAND flash chips by intelligently controlling the latching circuit of the page buffer



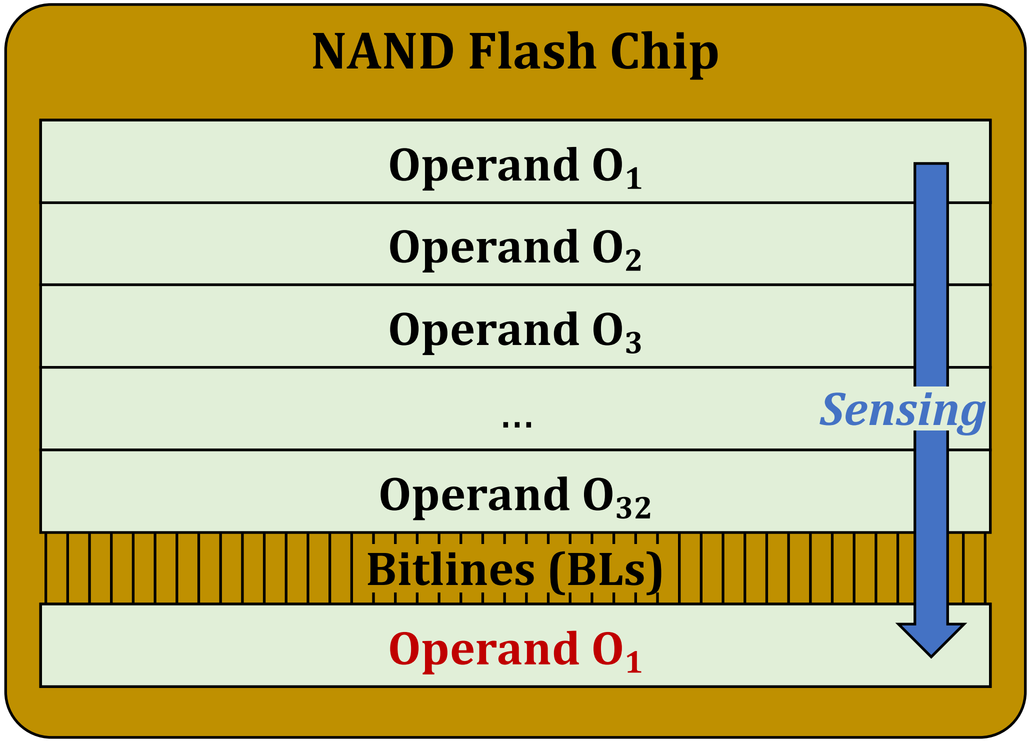
State-of-the-Art IFP Technique (1/2)

- ParaBit [Gao+, MICRO 2021]
  - Performs bulk bitwise operations inside NAND flash chips by intelligently controlling the latching circuit of the page buffer



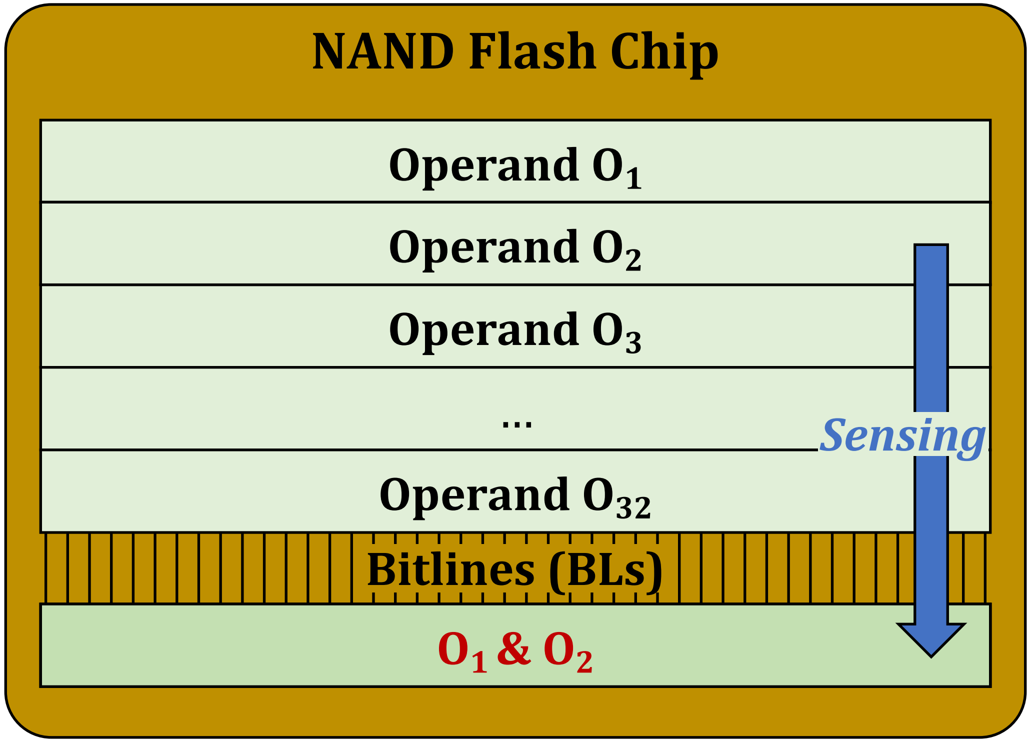
State-of-the-Art IFP Technique (1/2)

- ParaBit [Gao+, MICRO 2021]
  - Performs bulk bitwise operations inside NAND flash chips by intelligently controlling the latching circuit of the page buffer



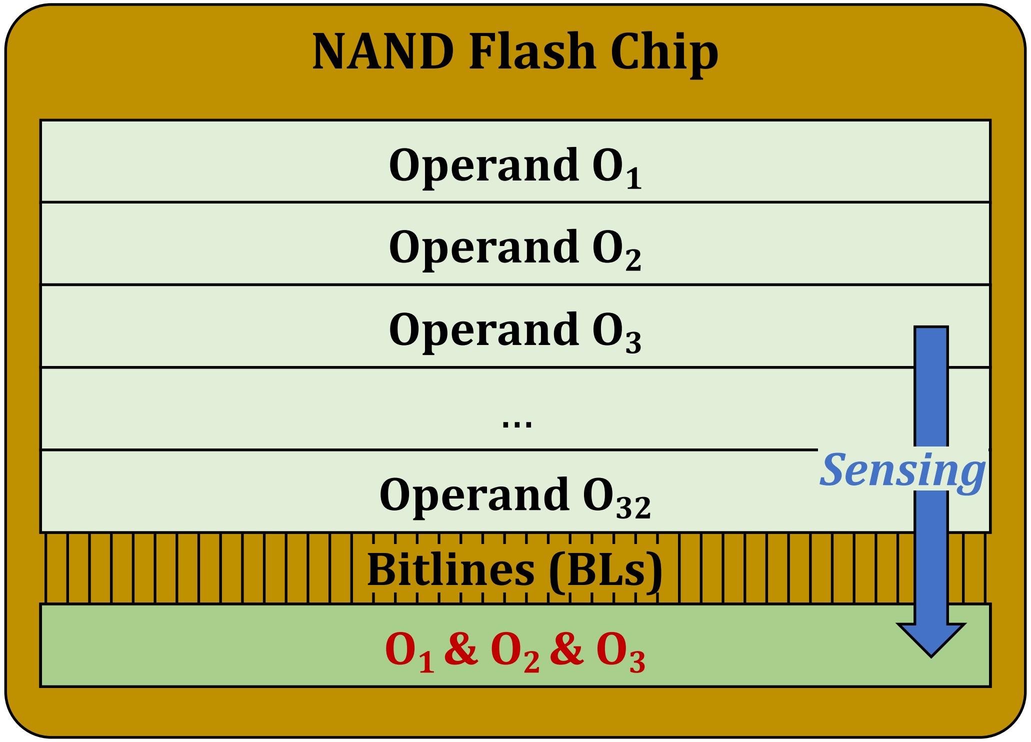
State-of-the-Art IFP Technique (1/2)

- ParaBit [Gao+, MICRO 2021]
  - Performs bulk bitwise operations inside NAND flash chips by intelligently controlling the latching circuit of the page buffer



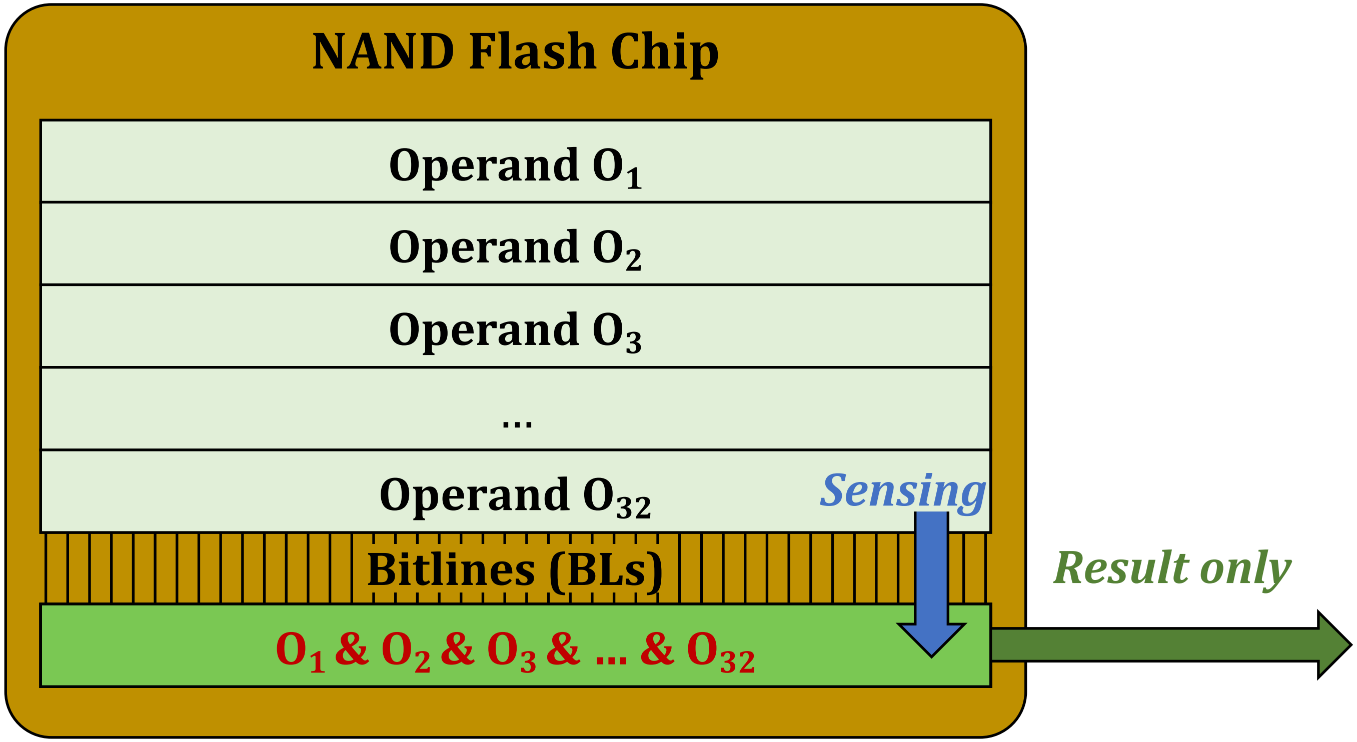
State-of-the-Art IFP Technique (1/2)

- ParaBit [Gao+, MICRO 2021]
  - Performs bulk bitwise operations inside NAND flash chips by intelligently controlling the latching circuit of the page buffer



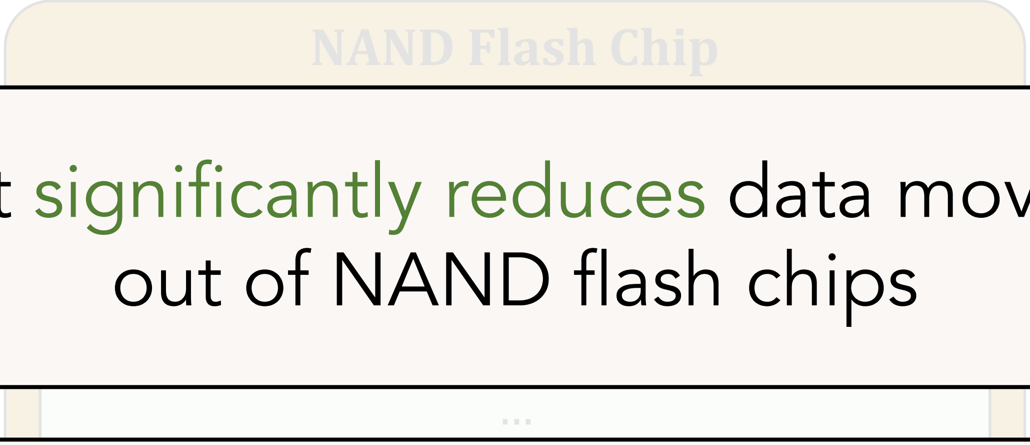
State-of-the-Art IFP Technique (1/2)

- ParaBit [Gao+, MICRO 2021]
  - Performs bulk bitwise operations inside NAND flash chips by intelligently controlling the latching circuit of the page buffer



State-of-the-Art IFP Technique (1/2)

- ParaBit [Gao+, MICRO 2021]
  - Performs bulk bitwise operations inside NAND flash chips by intelligently controlling the latching circuit of the page buffer



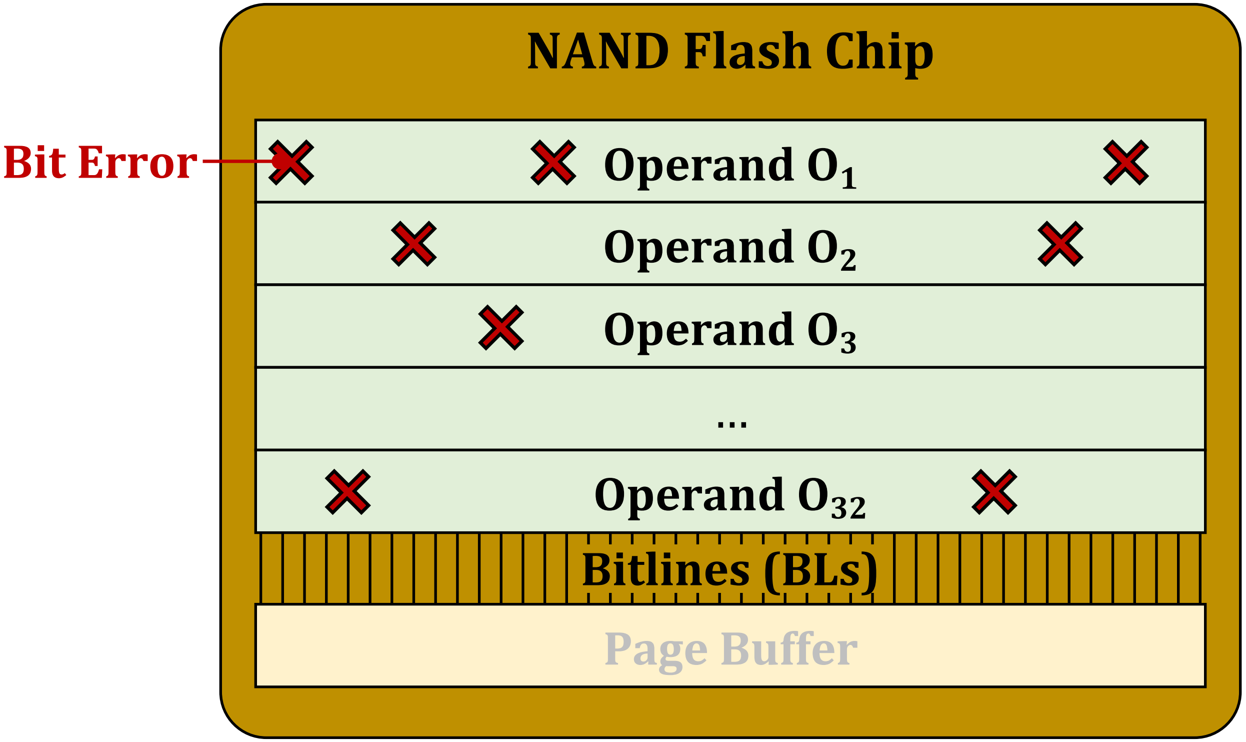
ParaBit significantly reduces data movement out of NAND flash chips

Serial sensing operations
(e.g., 32 sensing operations for 32-operand AND)
bottleneck performance and energy in ParaBit

SAFARI

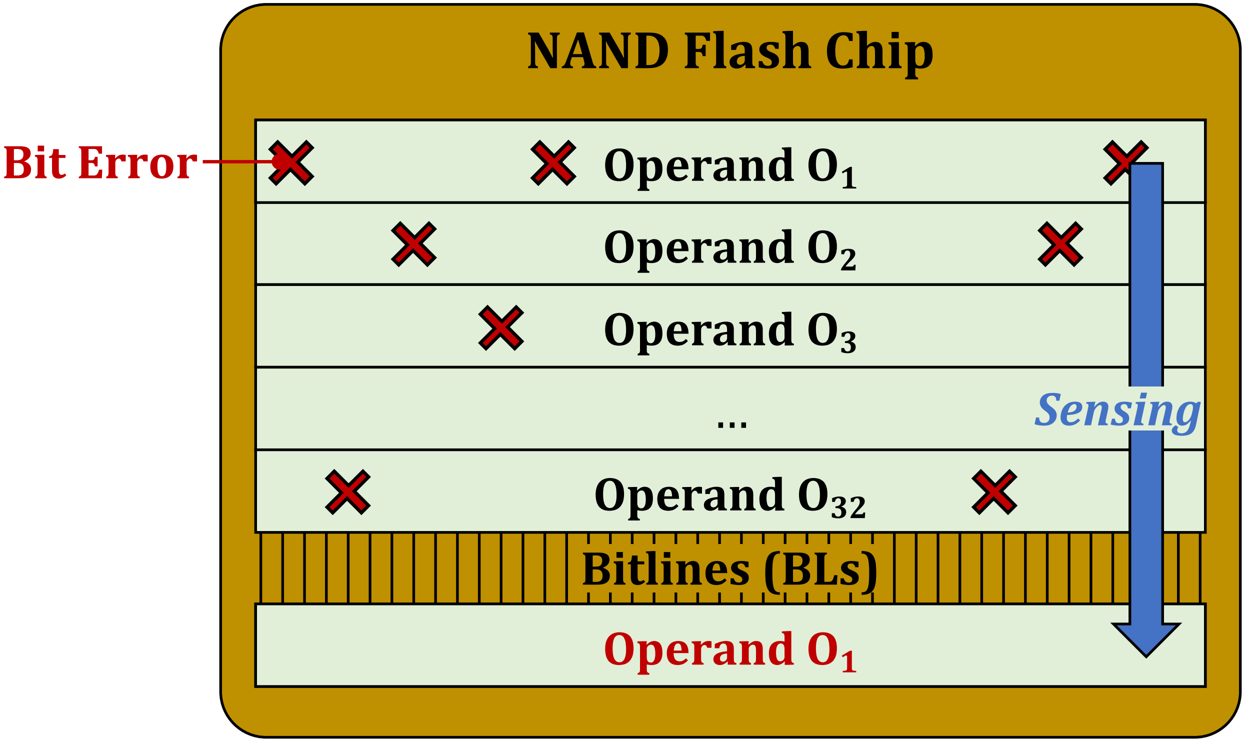
State-of-the-Art IFP Technique (2/2)

- Limitations of ParaBit [Gao+, MICRO 2021]
  - Serial sensing operations become the new bottleneck
  - Erroneous computation results due to the high raw bit-error rate of NAND flash memory



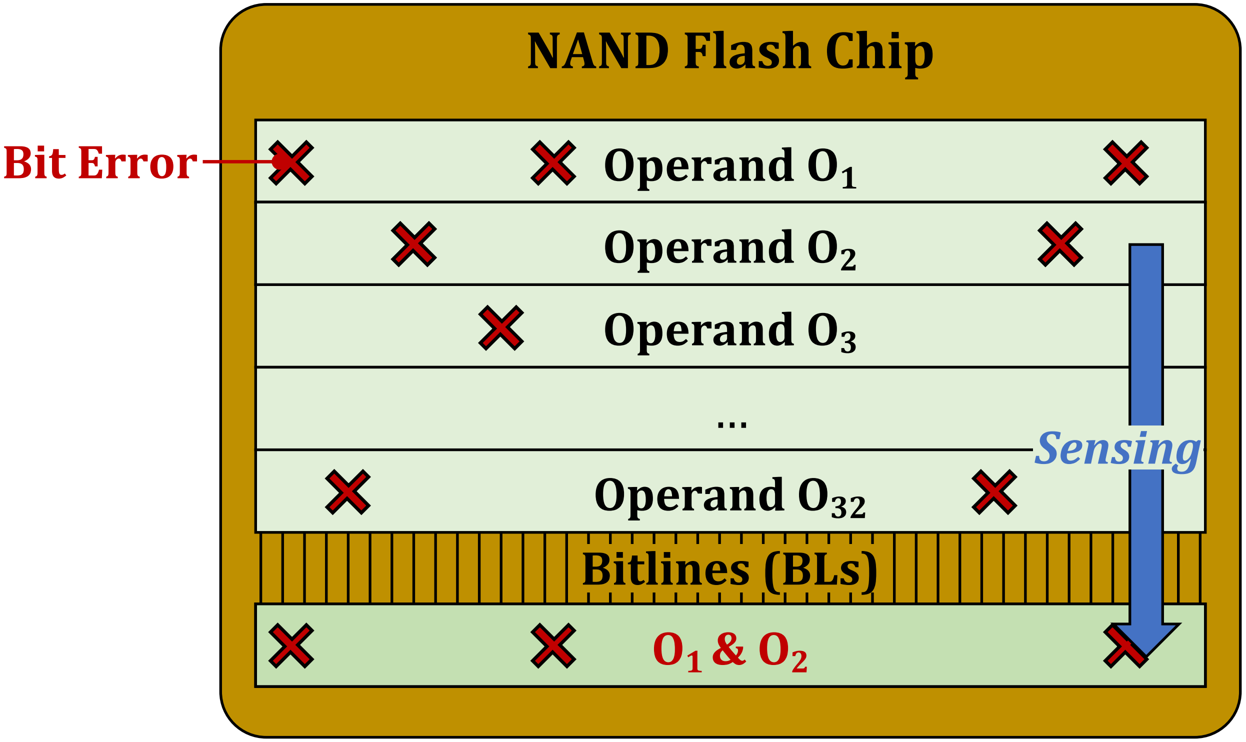
State-of-the-Art IFP Technique (2/2)

- Limitations of ParaBit [Gao+, MICRO 2021]
  - Serial sensing operations become the new bottleneck
  - Erroneous computation results due to the high raw bit-error rate of NAND flash memory



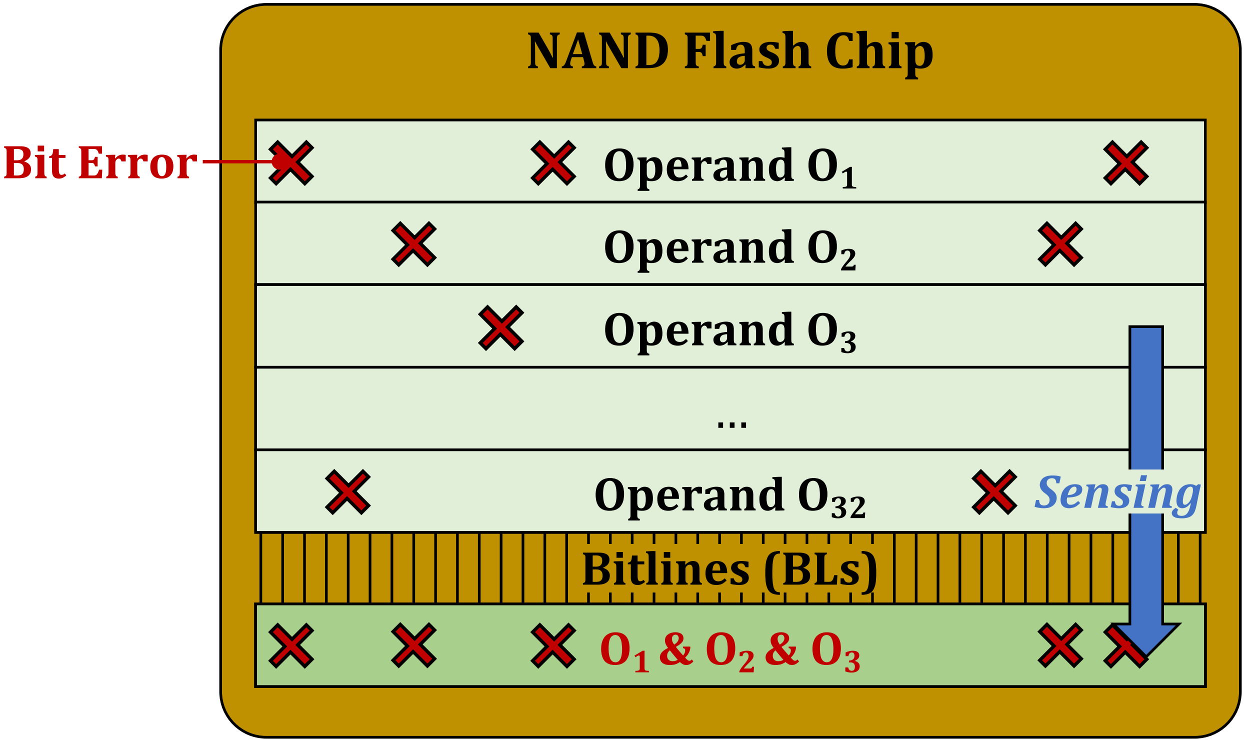
State-of-the-Art IFP Technique (2/2)

- Limitations of ParaBit [Gao+, MICRO 2021]
  - Serial sensing operations become the new bottleneck
  - Erroneous computation results due to the high raw bit-error rate of NAND flash memory



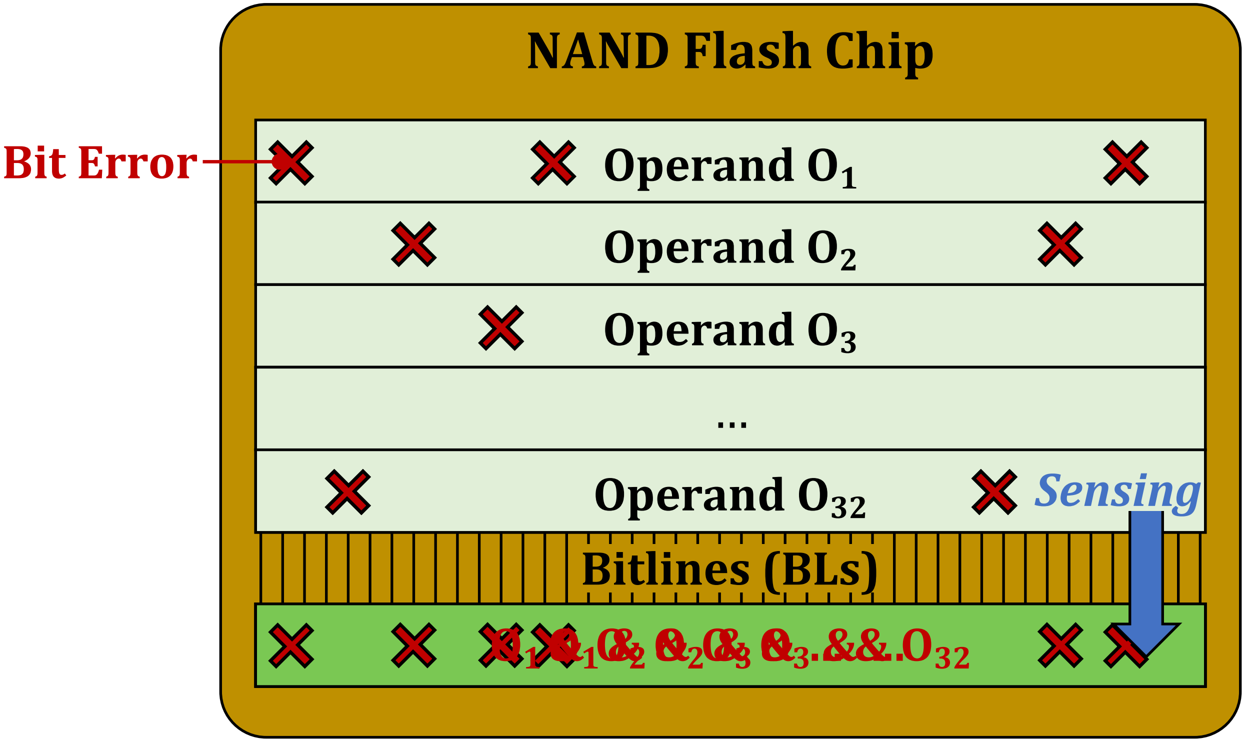
State-of-the-Art IFP Technique (2/2)

- Limitations of ParaBit [Gao+, MICRO 2021]
  - Serial sensing operations become the new bottleneck
  - Erroneous computation results due to the high raw bit-error rate of NAND flash memory



State-of-the-Art IFP Technique (2/2)

- Limitations of ParaBit [Gao+, MICRO 2021]
  - Serial sensing operations become the new bottleneck
  - Erroneous computation results due to the high raw bit-error rate of NAND flash memory



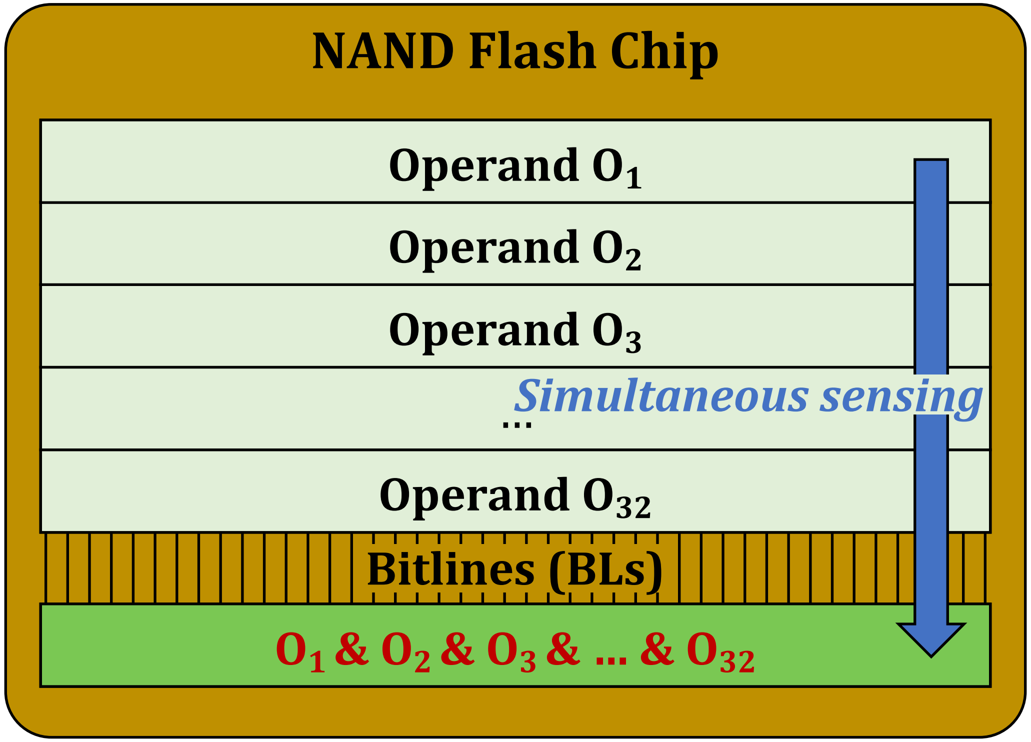
Our Goals

Address the new bottleneck of IFP
(serial sensing of operands)

Make IFP reliable
(provide accurate computation results)

Our Proposal: Flash-Cosmos

- Flash-Cosmos enables
  - Computation on multiple operands with a single sensing operation
  - Accurate computation results by eliminating raw bit errors in stored data

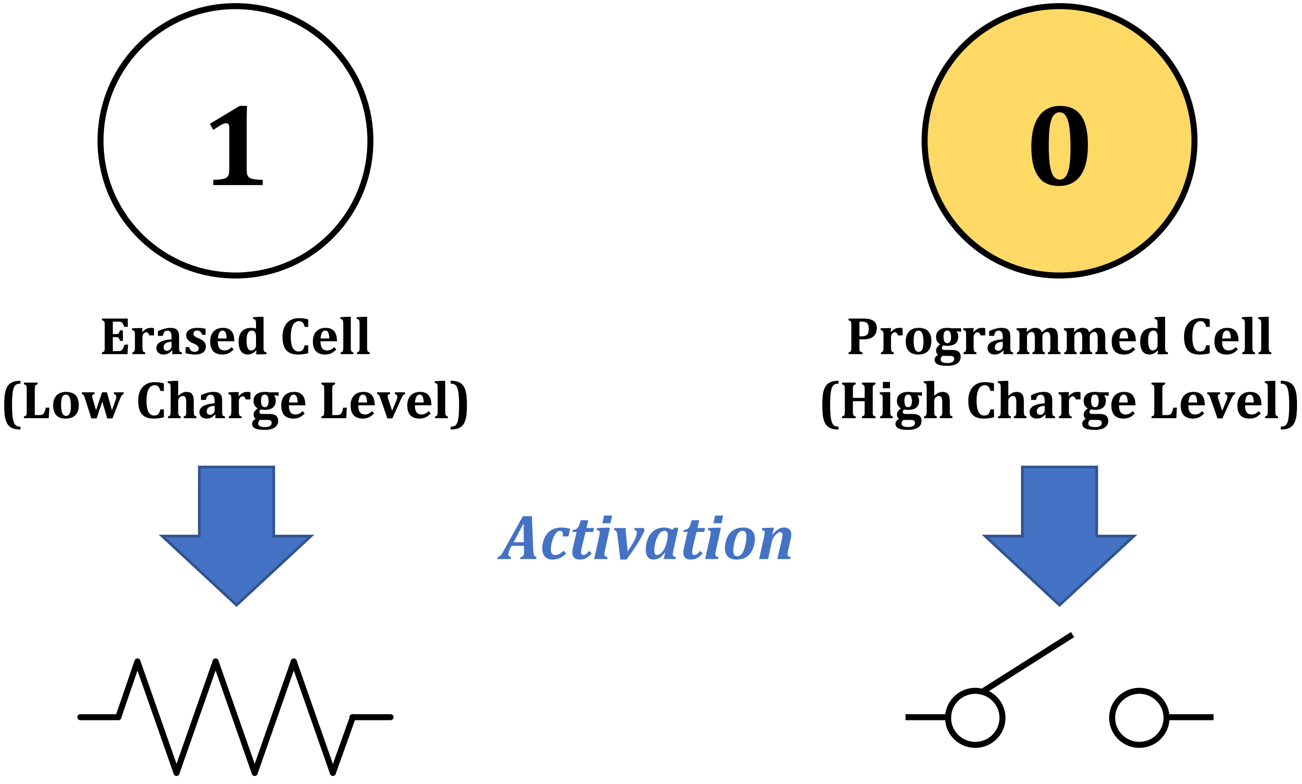


Talk Outline

- Problem, Goals & Key Idea
- Background
- Flash-Cosmos: Computation with One-Shot Multi-Operand Sensing
- Evaluation
- Summary

NAND Flash Basics: A Flash Cell

- A flash cell stores data by adjusting the amount of charge in the cell



Erased Cell (Low Charge Level)

Programmed Cell (High Charge Level)

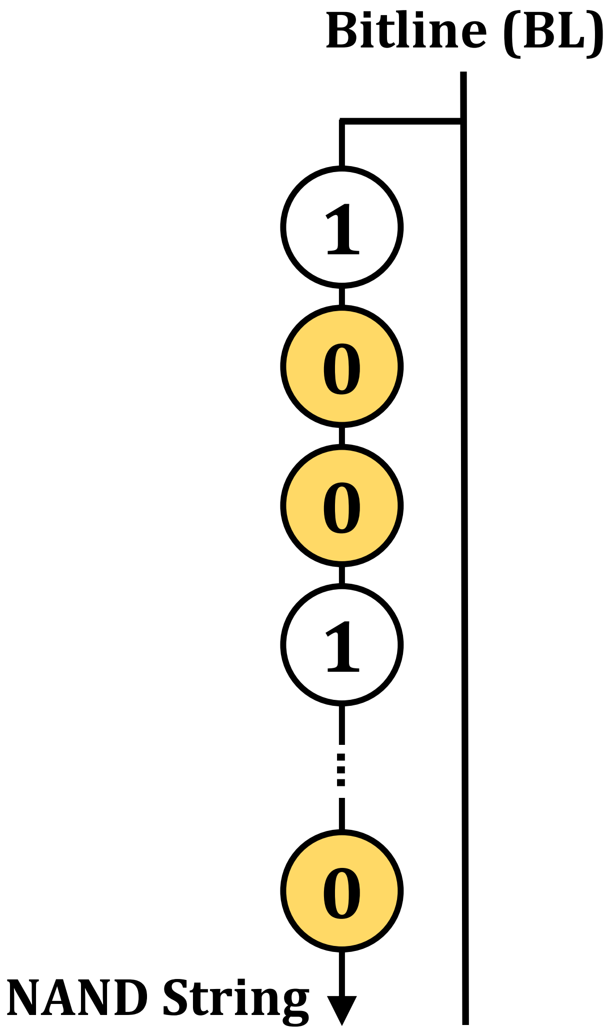
Activation

Operates as a resistor

Operates as an open switch

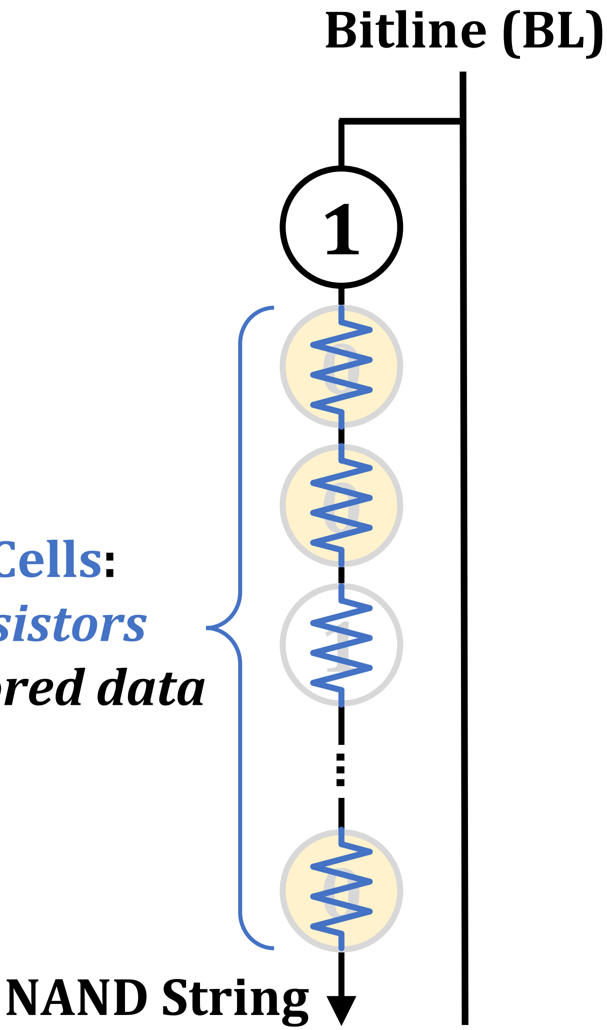
NAND Flash Basics: A NAND String

- A set of flash cells are serially connected, forming a NAND string



NAND Flash Basics: Read Mechanism (1/3)

- NAND flash memory reads data by checking the bitline current



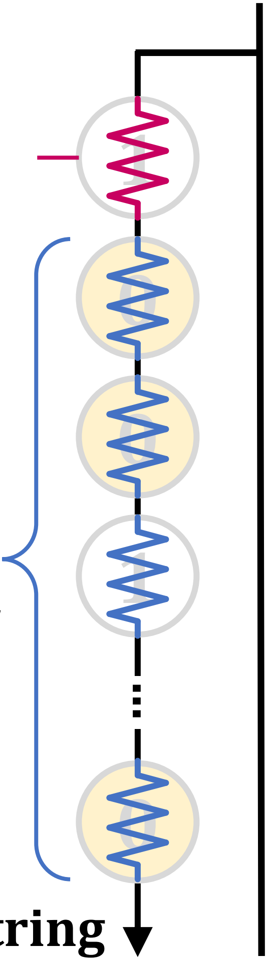
Non-Target Cells:

Operate as resistors regardless of stored data

SAFARI

NAND Flash Basics: Read Mechanism (2/3)

- NAND flash memory reads data by checking the bitline current

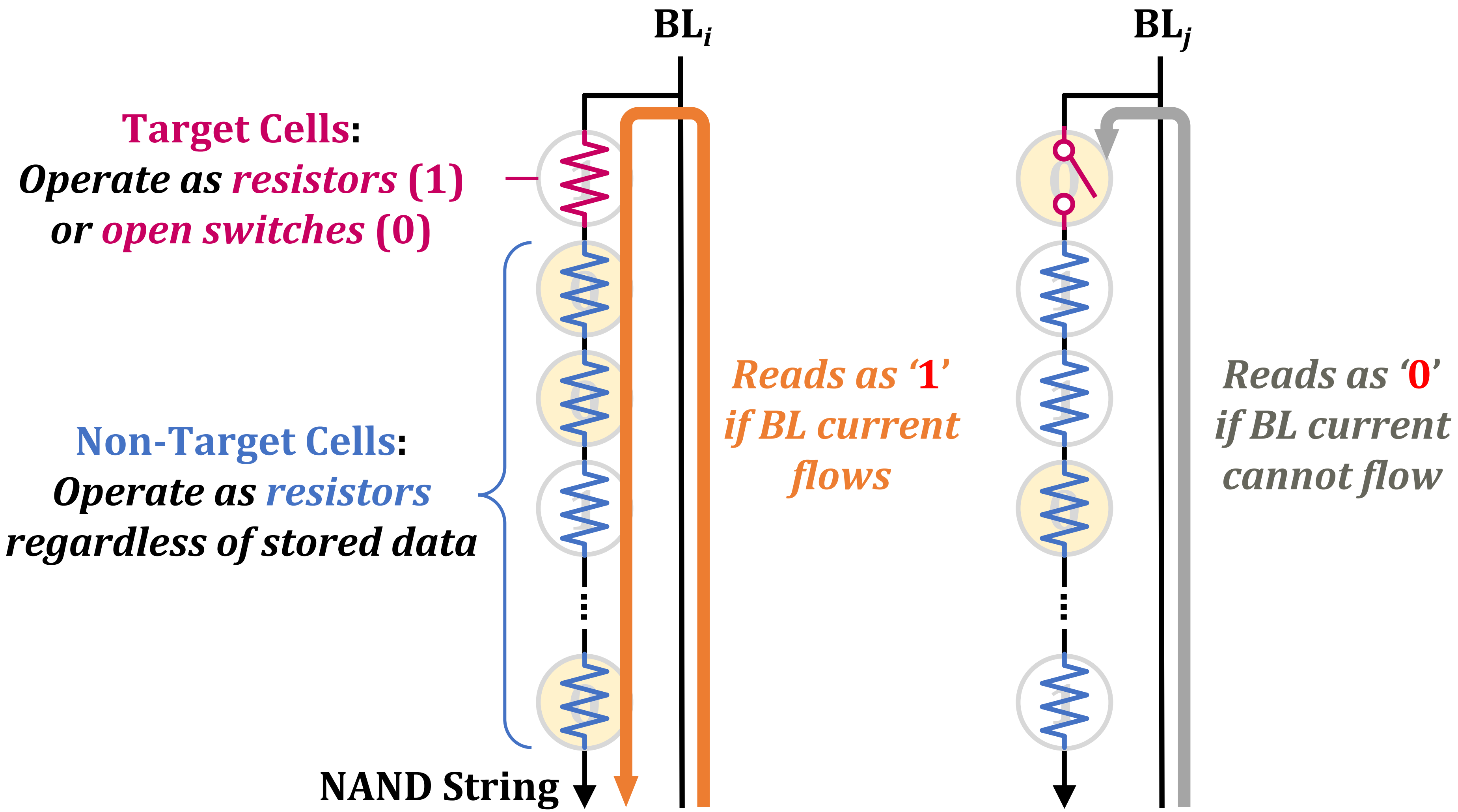


Target Cells:
Operate as resistors (1)
or open switches (0)

Non-Target Cells:
Operate as resistors
regardless of stored data

NAND Flash Basics: Read Mechanism (3/3)

- NAND flash memory reads data by checking the bitline current



Target Cells:
Operate as resistors (1)
or open switches (0)

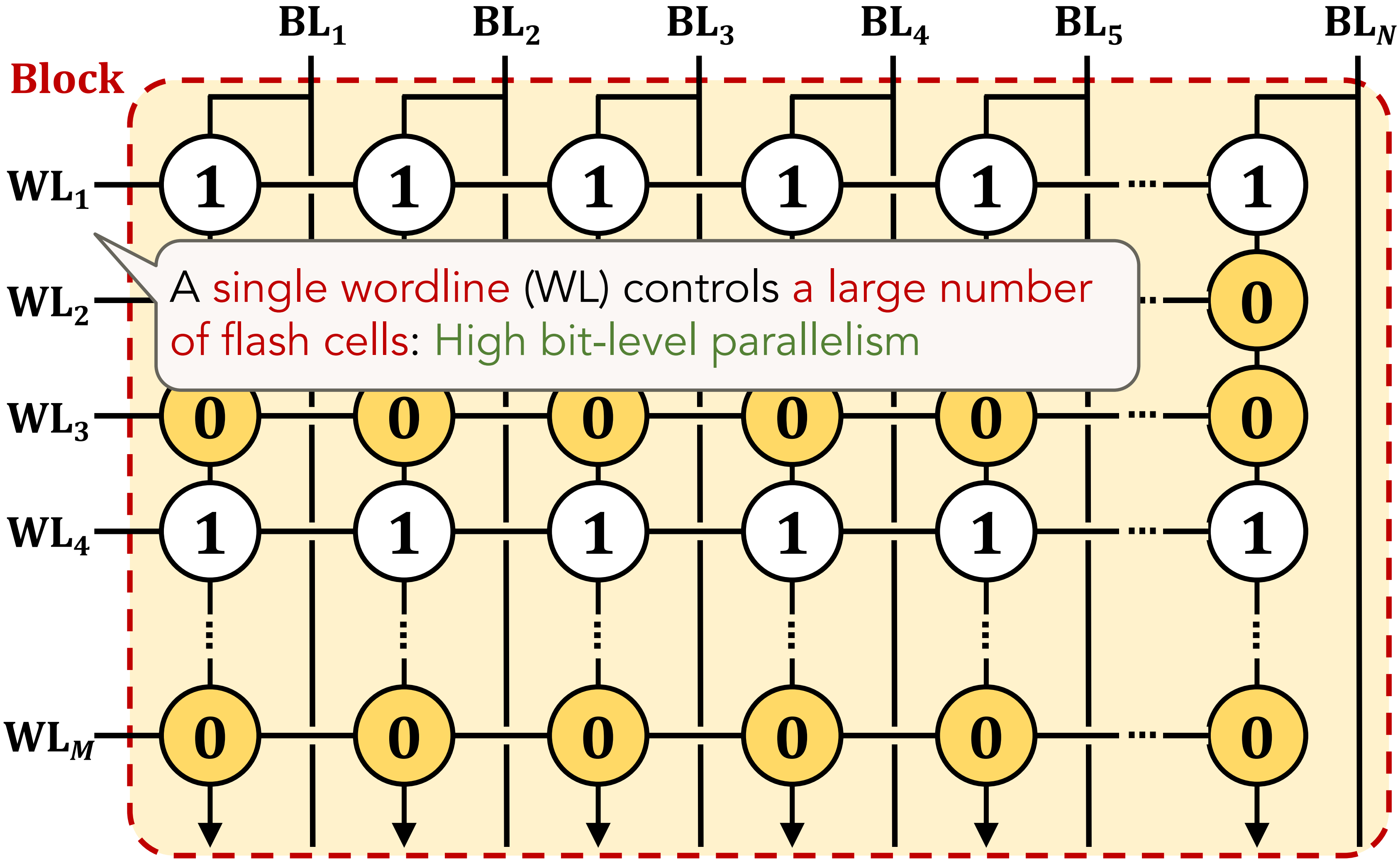
Non-Target Cells:
Operate as resistors
regardless of stored data

Reads as '1'
if BL current
flows

Reads as '0'
if BL current
cannot flow

NAND Flash Basics: A NAND Flash Block

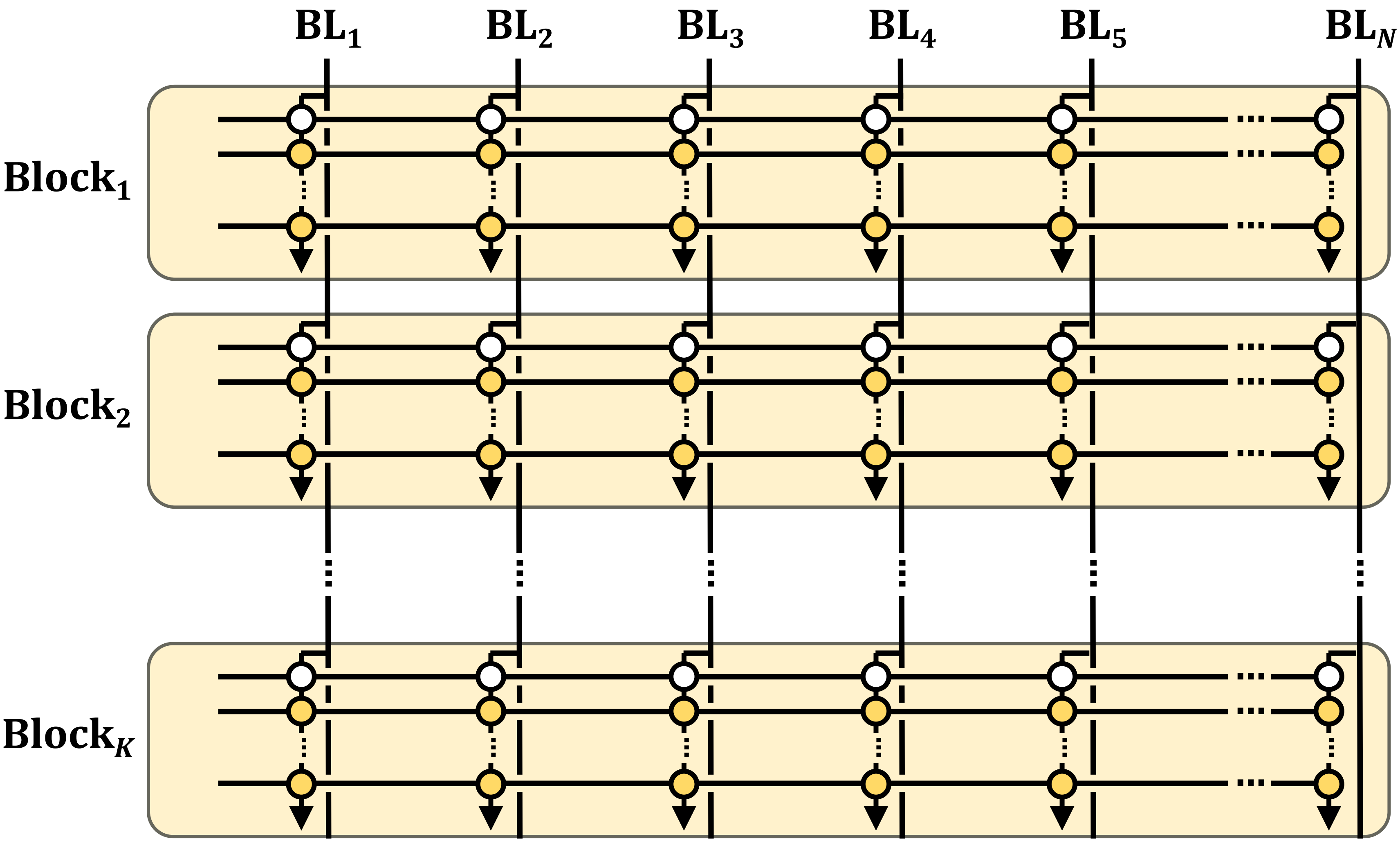
- NAND strings connected to different bitlines comprise a block



A single wordline (WL) controls a large number of flash cells: High bit-level parallelism

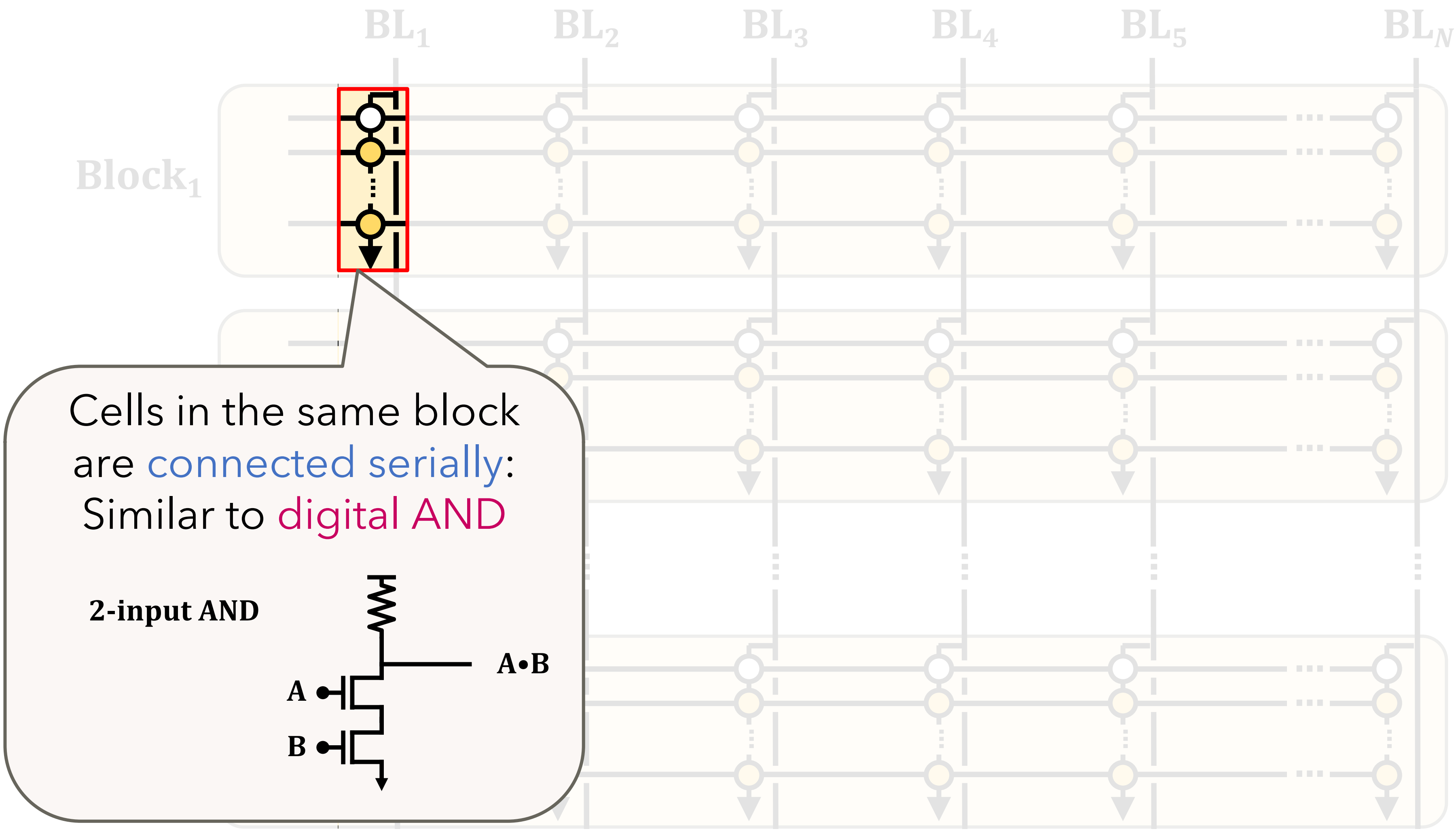
NAND Flash Basics: Block Organization

- A large number of blocks share the same bitlines



Similarity to Digital Logic Gates

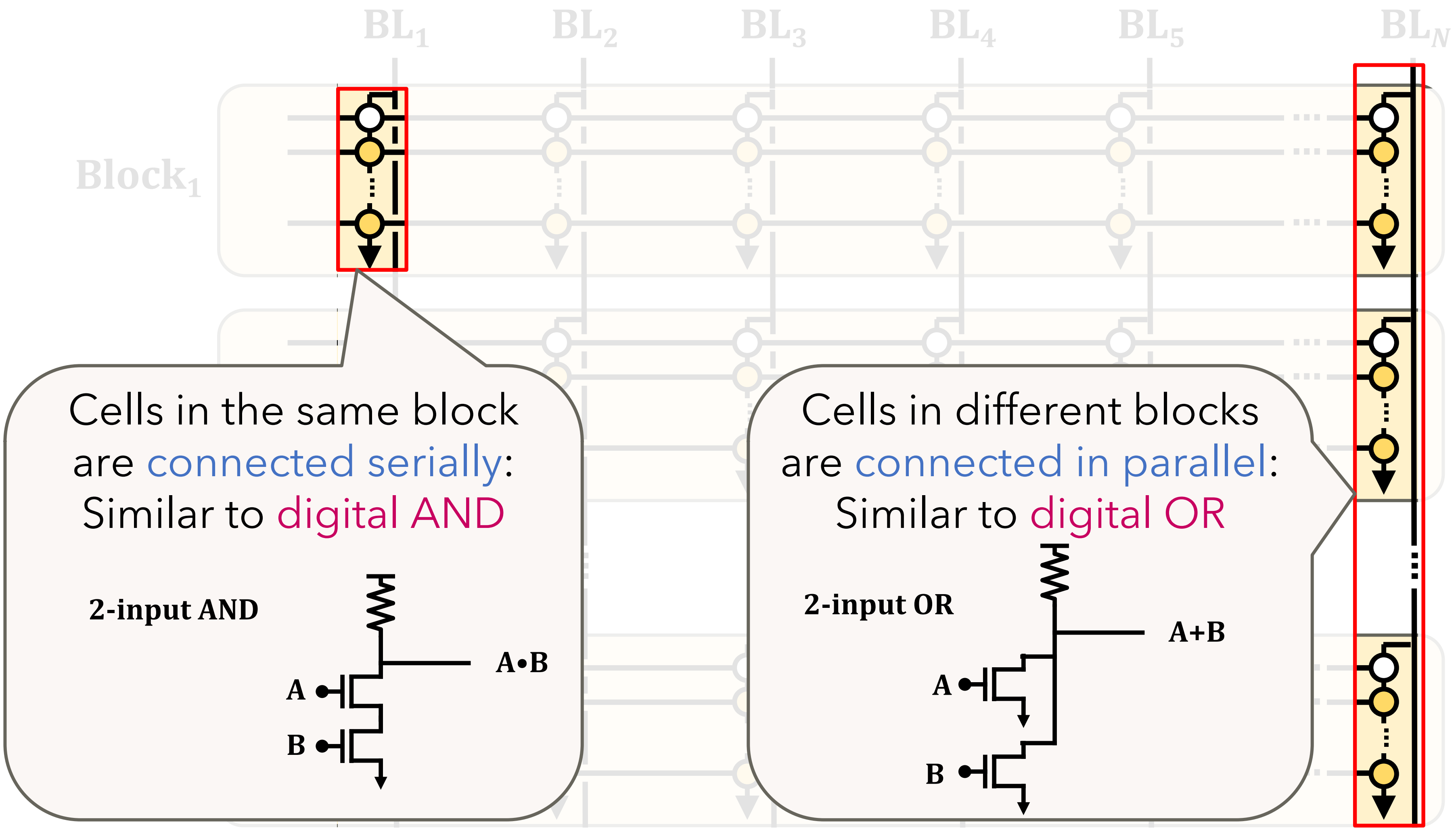
- A large number of blocks share the same bitlines



Cells in the same block are connected serially: Similar to digital AND

Similarity to Digital Logic Gates

- A large number of blocks share the same bitlines.



Cells in the same block are connected serially: Similar to digital AND

Cells in different blocks are connected in parallel: Similar to digital OR

Talk Outline

- Problem, Goals & Key Idea
- Background
- Flash-Cosmos: Computation with One-Shot Multi-Operand Sensing
- Evaluation
- Summary

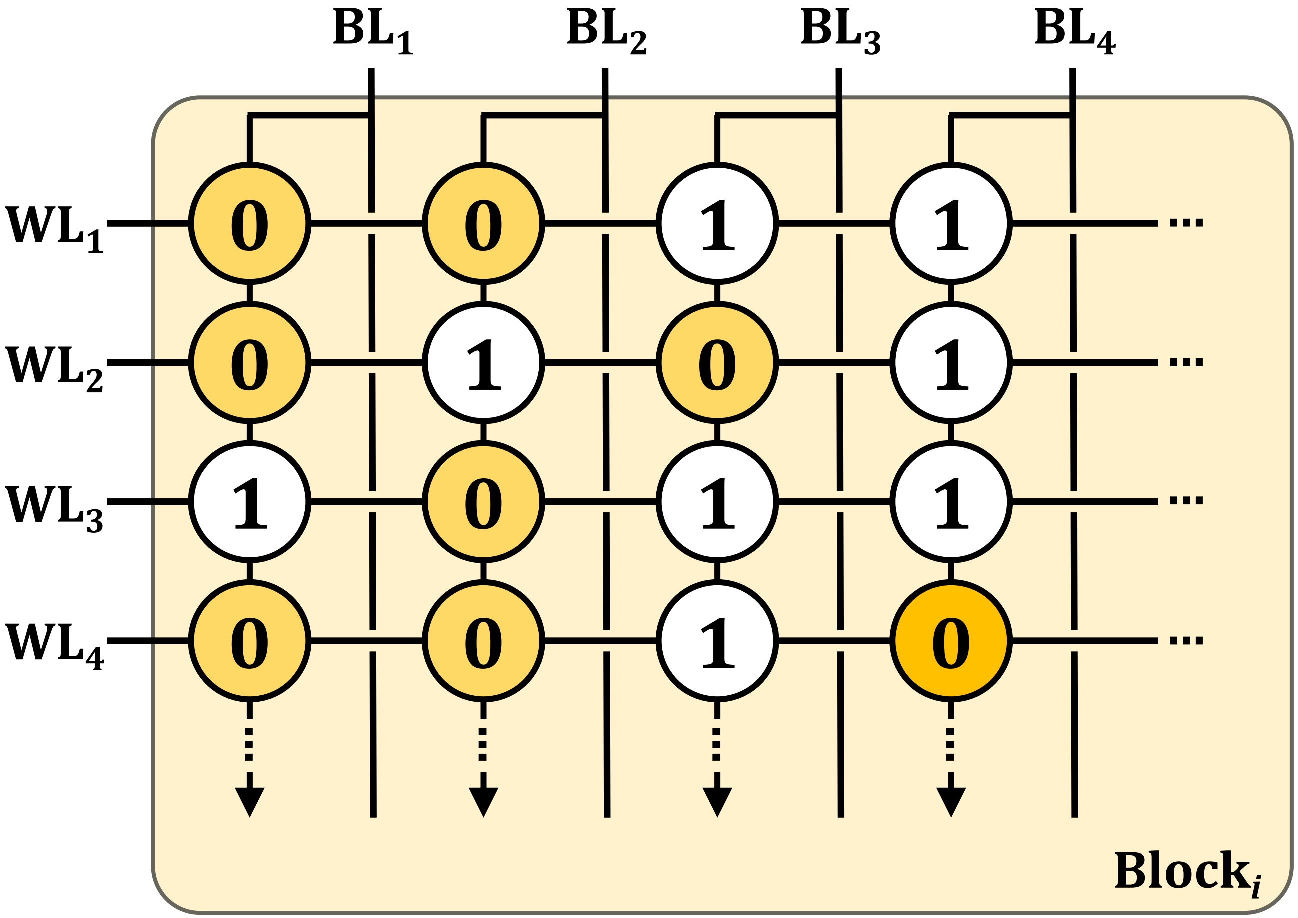
Key Ideas

Multi-Wordline Sensing (MWS)
to enable in-flash bulk bitwise operations via a single sensing operation

Enhanced SLC-Mode Programming (ESP)
to eliminate raw bit errors in stored data (and thus in computation results)

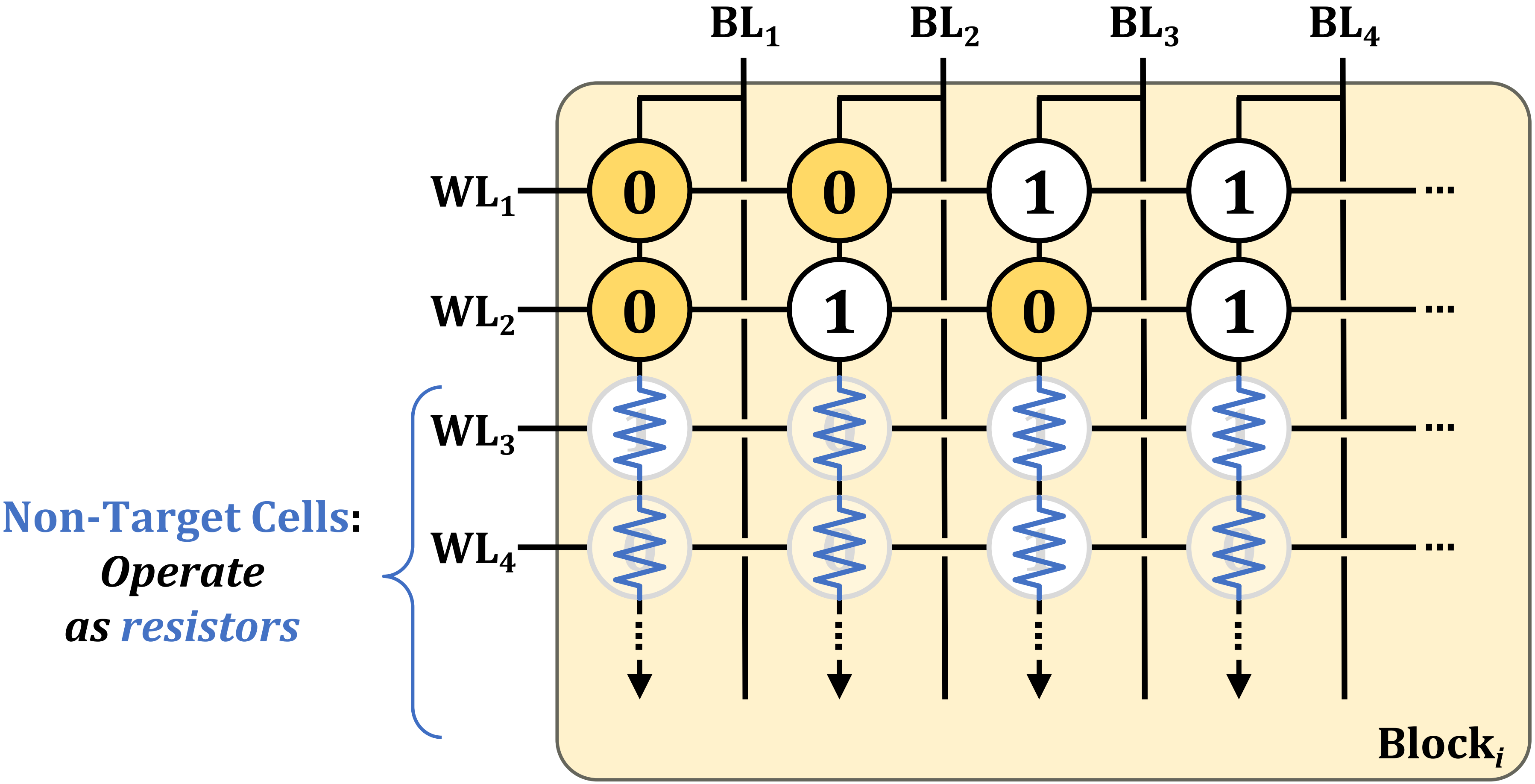
Multi-Wordline Sensing (MWS): Bitwise AND

- Intra-Block MWS:
  Simultaneously activates multiple WLs in the same block
  → Bitwise AND of the stored data in the WLs



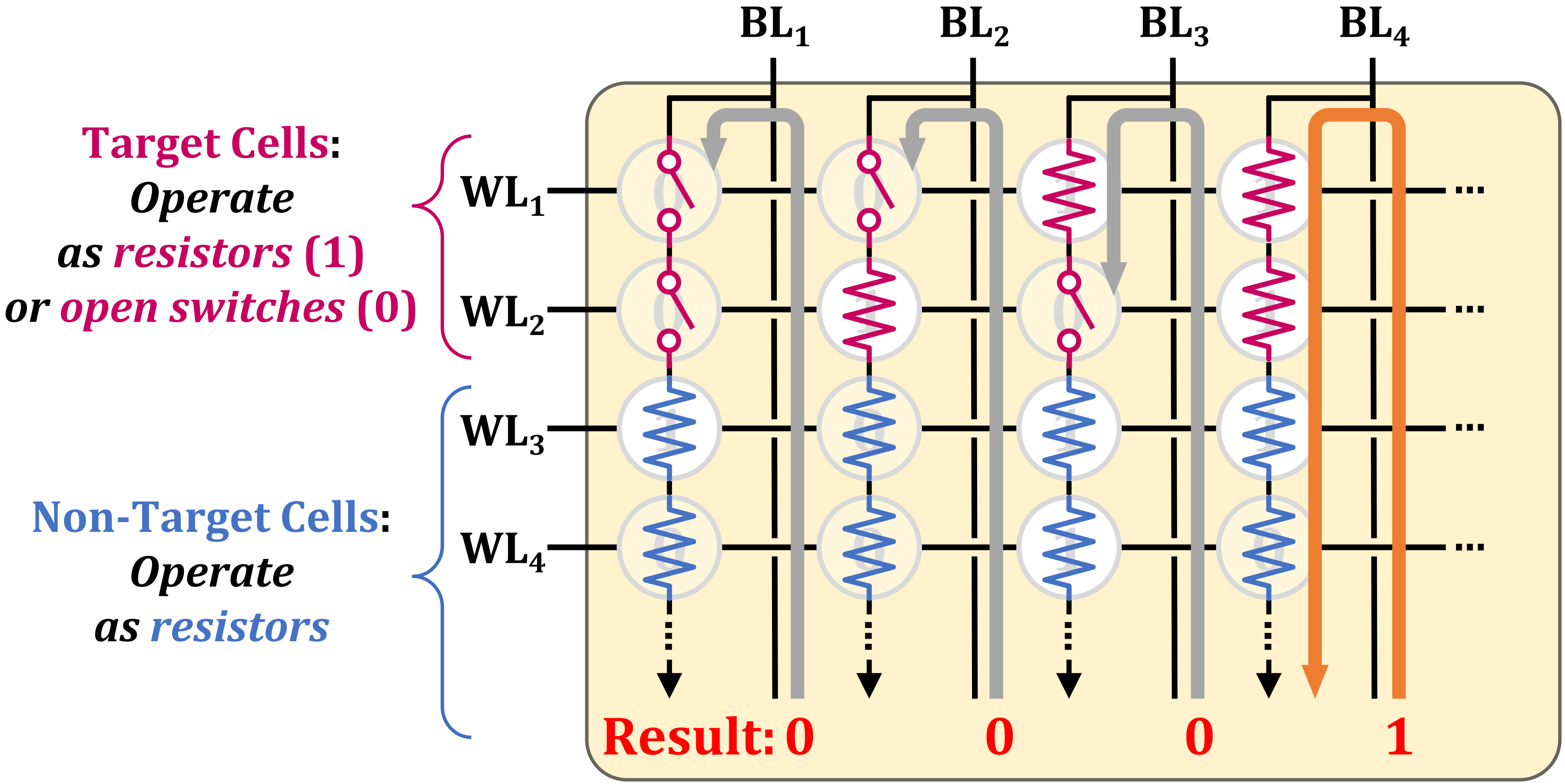
Multi-Wordline Sensing (MWS): Bitwise AND

- Intra-Block MWS:
  Simultaneously activates multiple WLs in the same block
  → Bitwise AND of the stored data in the WLs



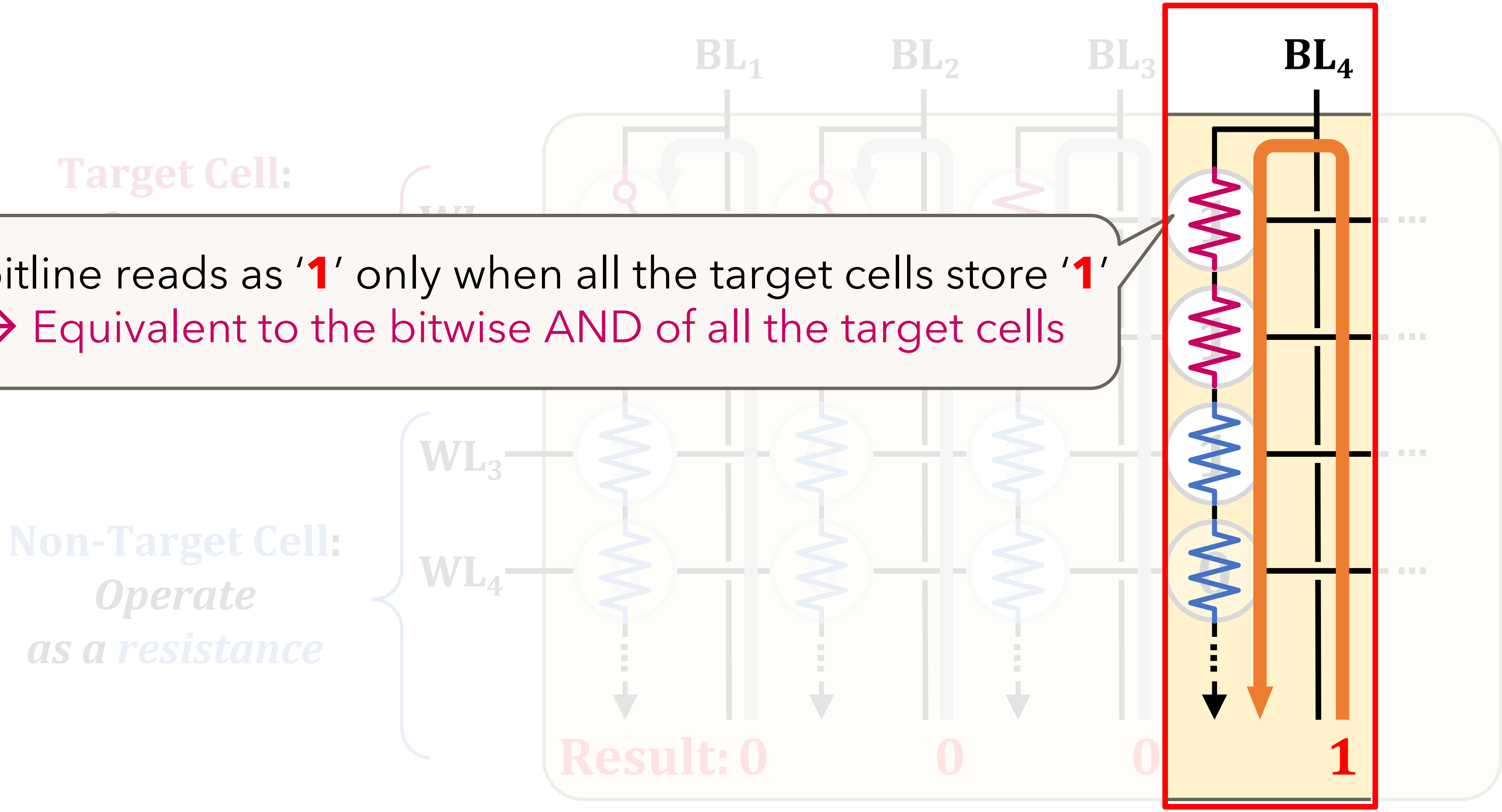
Multi-Wordline Sensing (MWS): Bitwise AND

- Intra-Block MWS:
  Simultaneously activates multiple WLs in the same block
  → Bitwise AND of the stored data in the WLs



Multi-Wordline Sensing (MWS): Bitwise AND

- Intra-Block MWS:
  Simultaneously activates multiple WLs in the same block
  → Bitwise AND of the stored data in the WLs

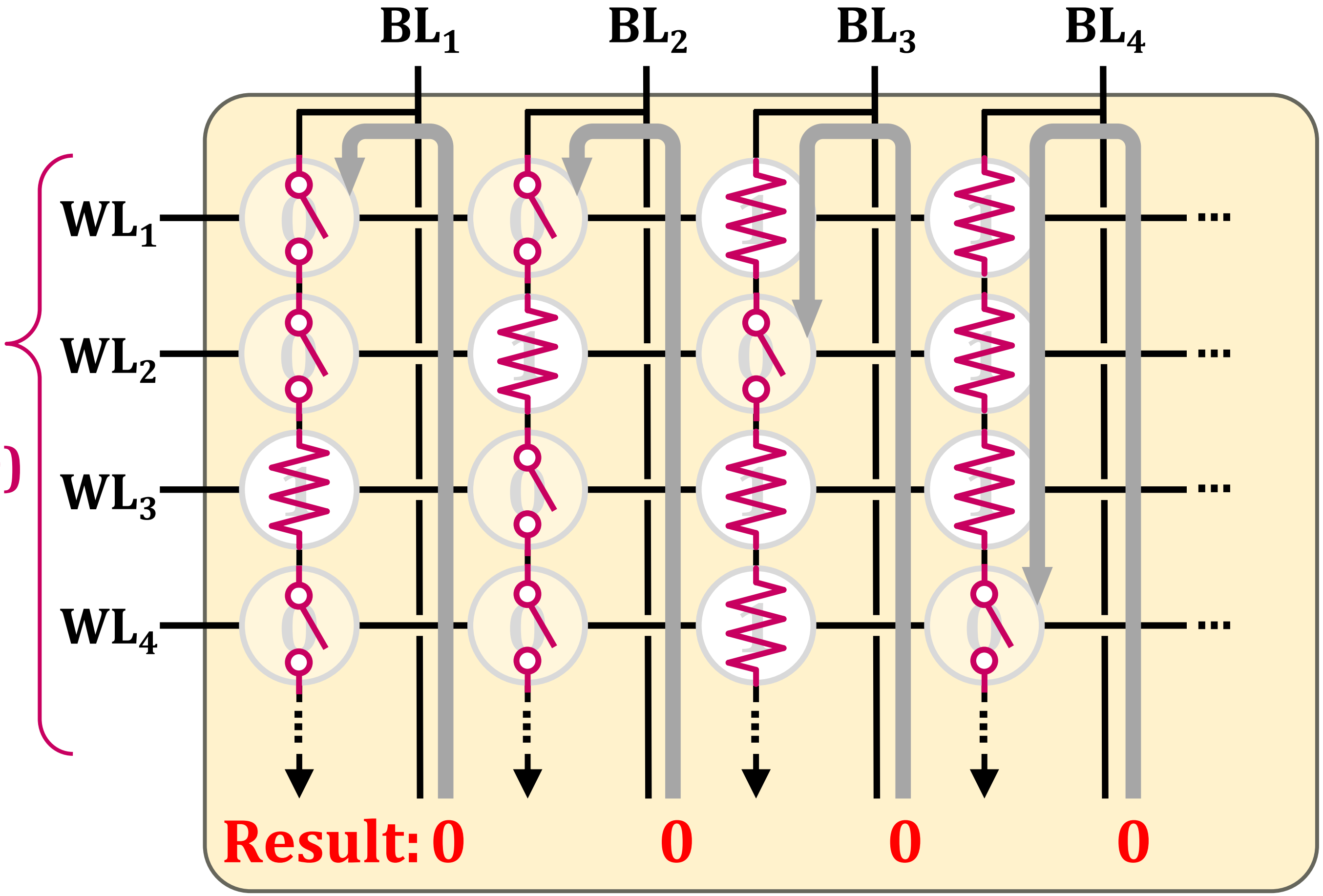


A bitline reads as '**1**' only when all the target cells store '**1**'
→ Equivalent to the bitwise AND of all the target cells

Multi-Wordline Sensing (MWS): Bitwise AND

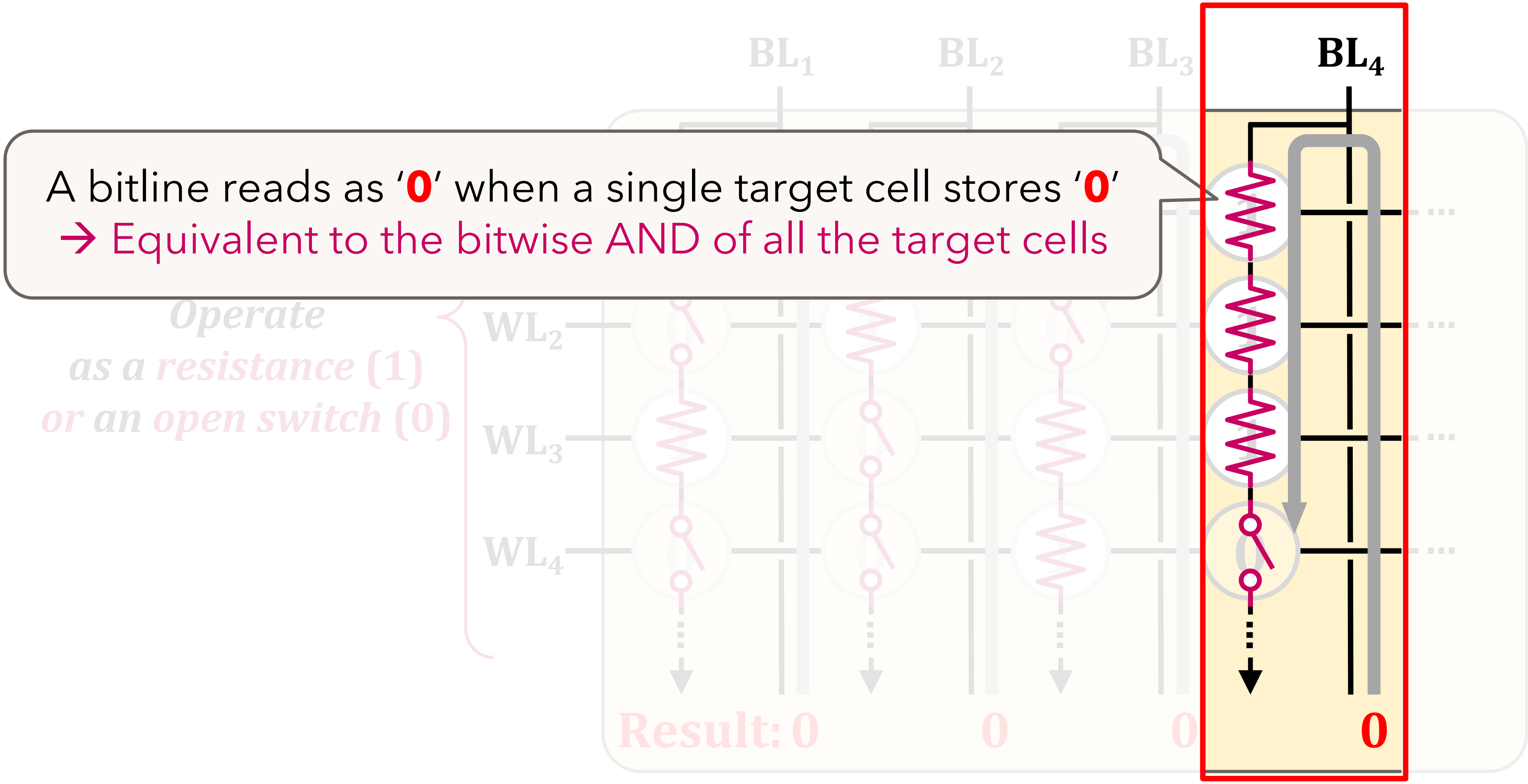
- Intra-Block MWS:
  Simultaneously activates multiple WLs in the same block
  → Bitwise AND of the stored data in the WLs

Target Cell:
Operate
as a resistance (1)
or an open switch (0)



Multi-Wordline Sensing (MWS): Bitwise AND

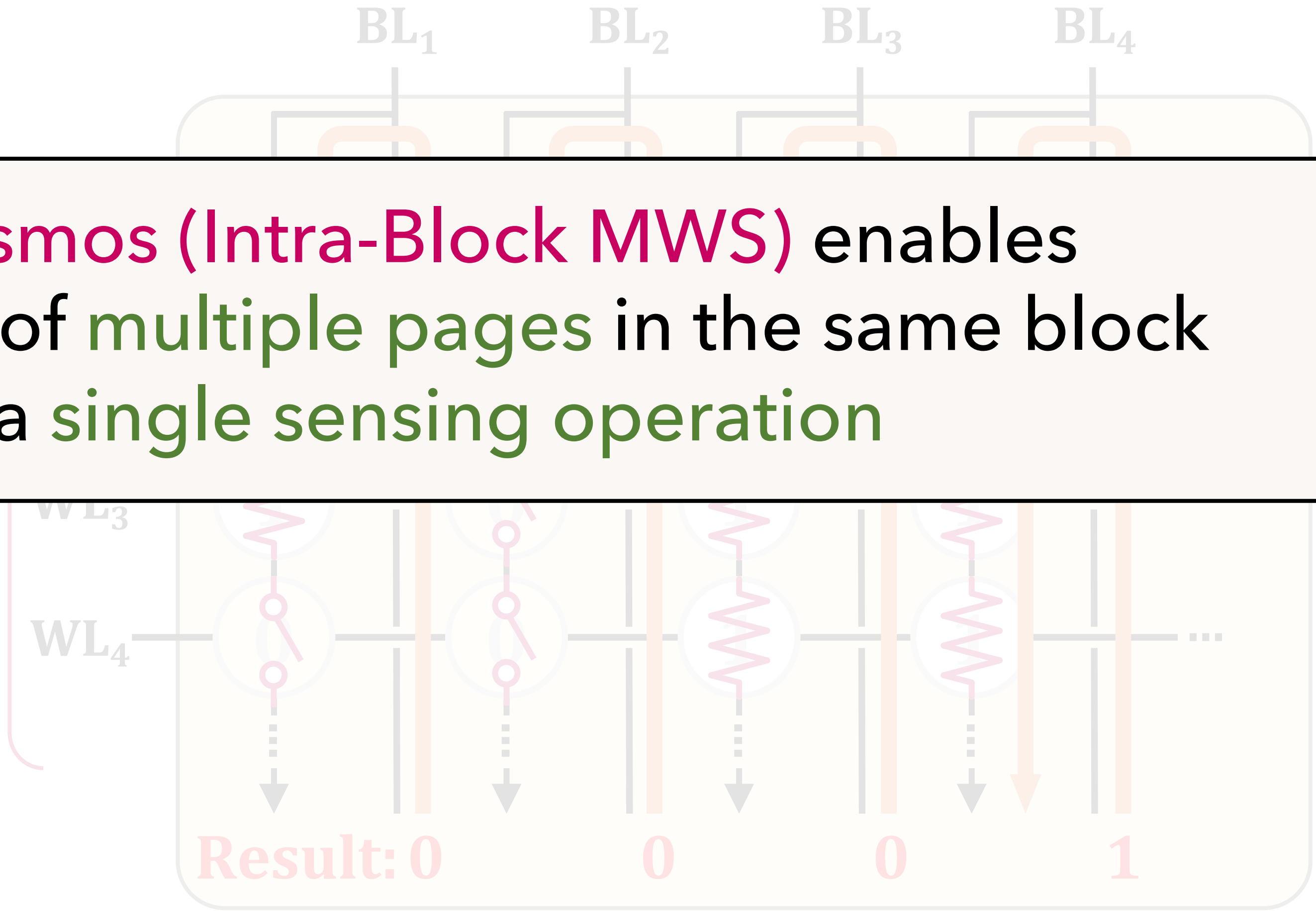
- **Intra-Block MWS**:
  Simultaneously activates multiple WLs in the same block
  → Bitwise AND of the stored data in the WLs



A bitline reads as '**0**' when a single target cell stores '**0**'
→ Equivalent to the bitwise AND of all the target cells

Multi-Wordline Sensing (MWS): Bitwise AND

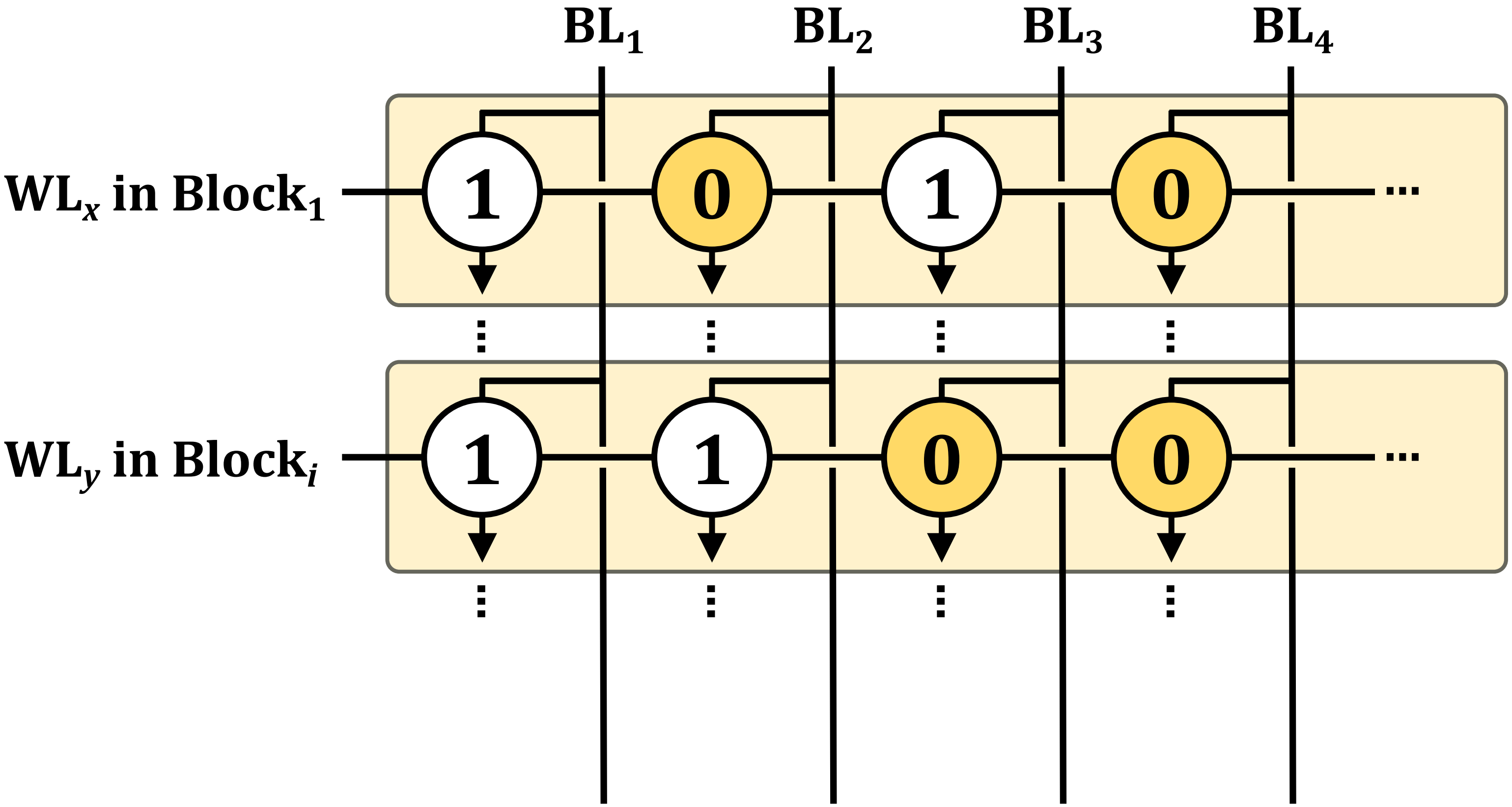
- Intra-Block MWS:
  Simultaneously activates multiple WLs in the same block
  → Bitwise AND of the stored data in the WLs



Flash-Cosmos (Intra-Block MWS) enables bitwise AND of multiple pages in the same block via a single sensing operation

Multi-Wordline Sensing (MWS): Bitwise OR

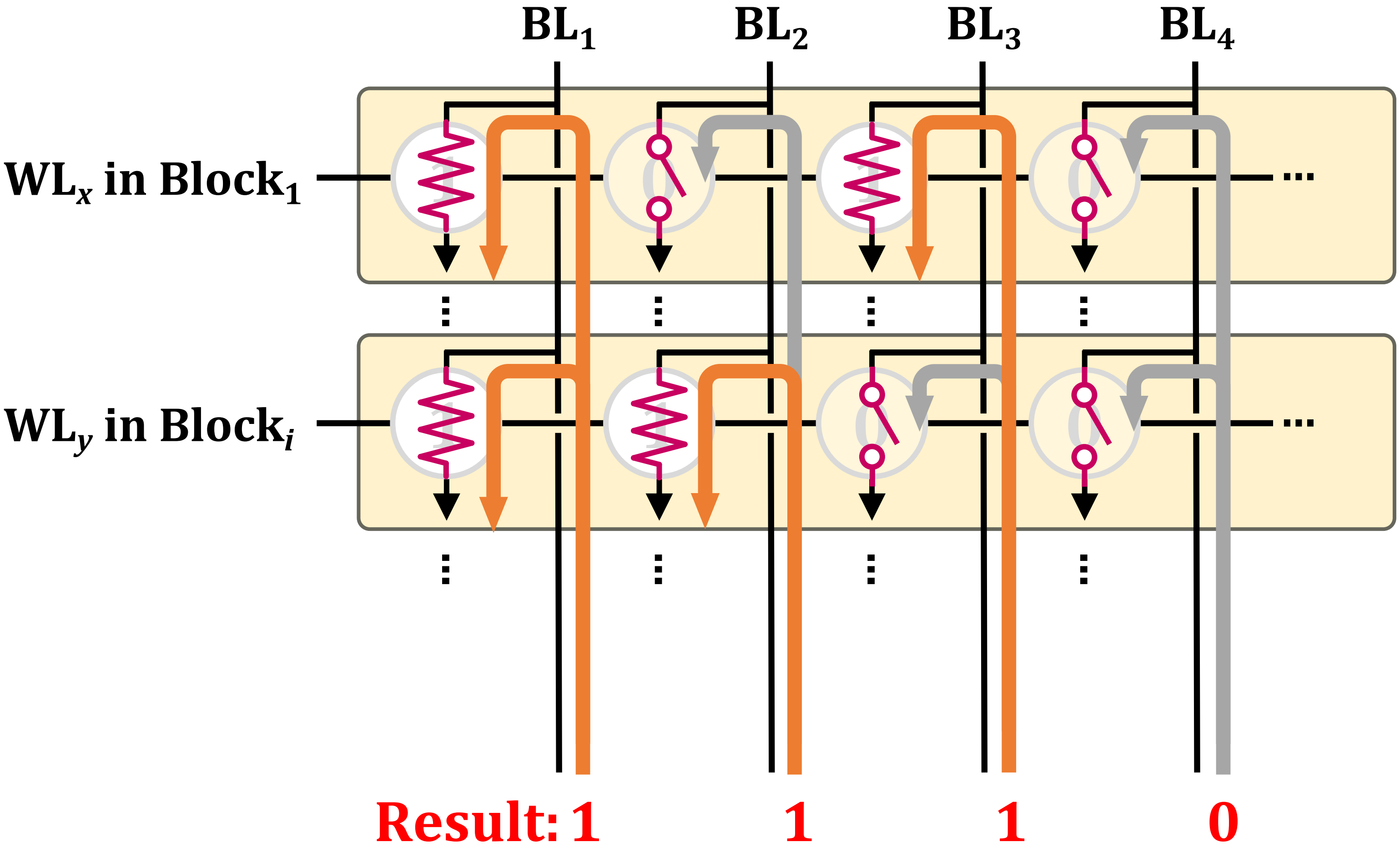
- Inter-Block MWS:
  Simultaneously activates multiple WLs in different blocks
  → Bitwise OR of the stored data in the WLs



SAFARI

Multi-Wordline Sensing (MWS): Bitwise OR

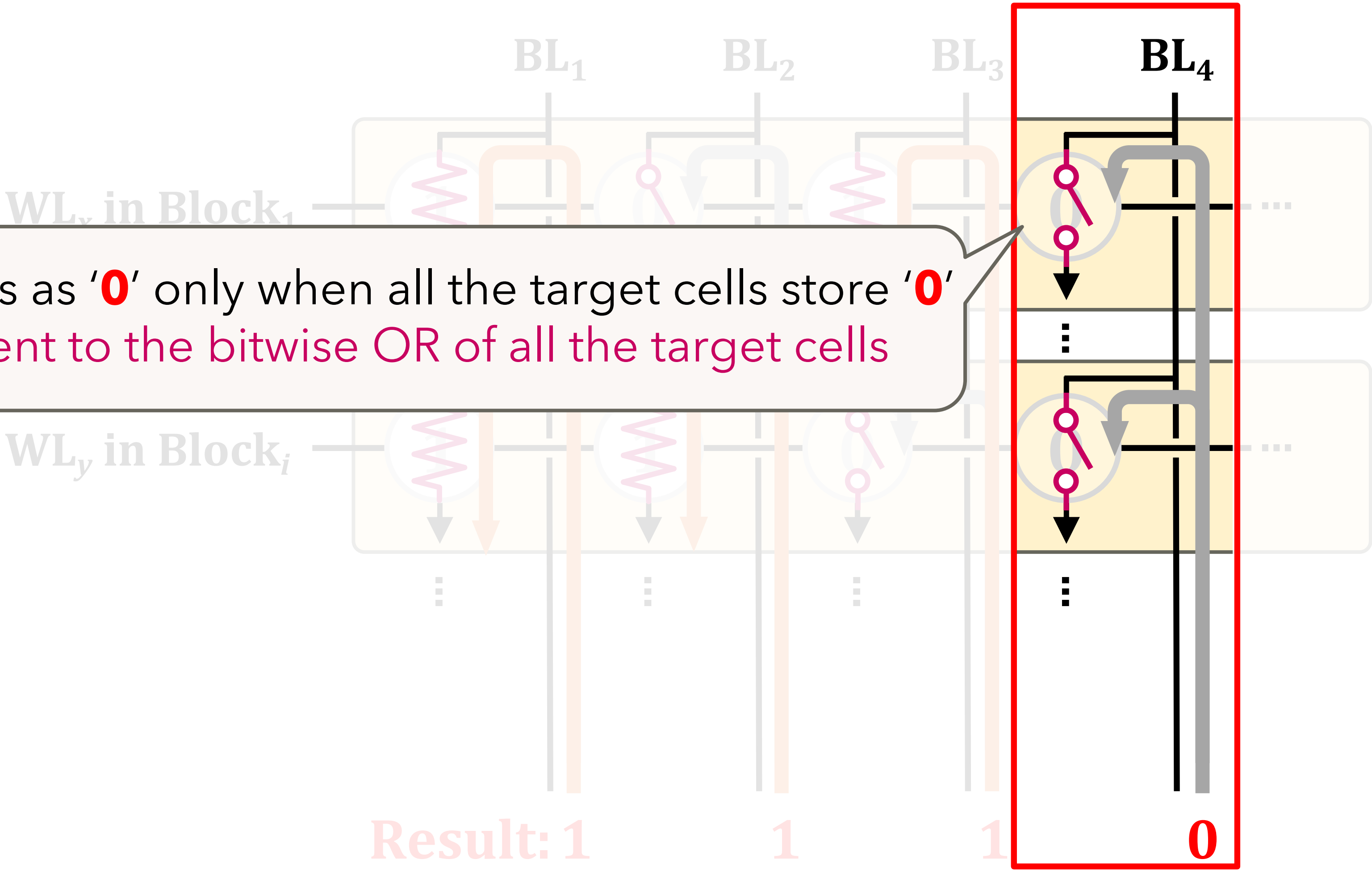
- Inter-Block MWS:
  Simultaneously activates multiple WLs in different blocks
  → Bitwise OR of the stored data in the WLs



$\mathbf{BL_1}$
$\mathbf{BL_2}$
$\mathbf{BL_3}$
$\mathbf{BL_4}$
$\mathbf{WL_x}$ in $\mathbf{Block_1}$
$\mathbf{WL_y}$ in $\mathbf{Block_i}$
Result: 1
1
1
0

Multi-Wordline Sensing (MWS): Bitwise OR

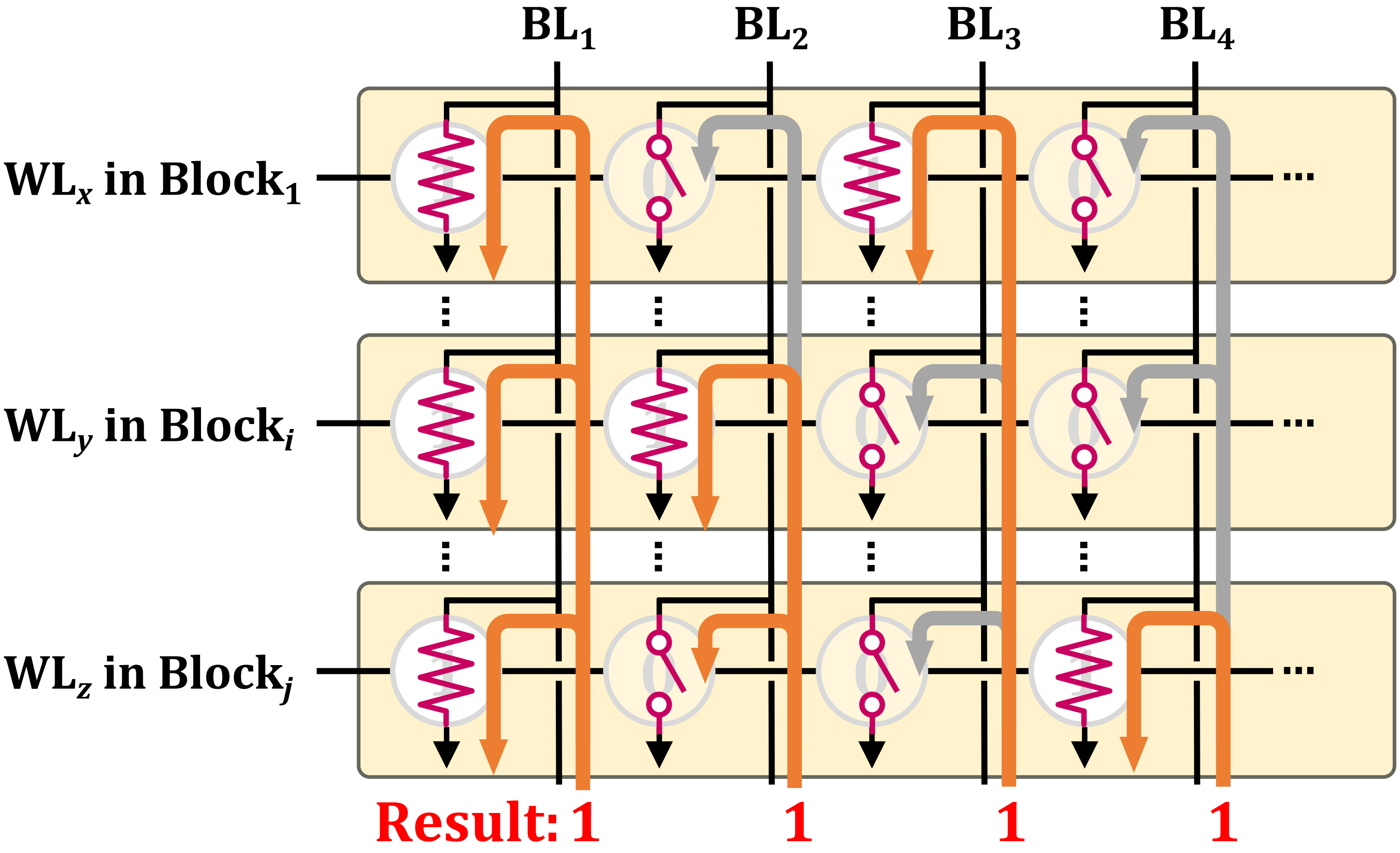
- **Inter-Block MWS**:
  Simultaneously activates multiple WLs in different blocks
  → Bitwise OR of the stored data in the WLs



A bitline reads as '**0**' only when all the target cells store '**0**'
→ Equivalent to the bitwise OR of all the target cells

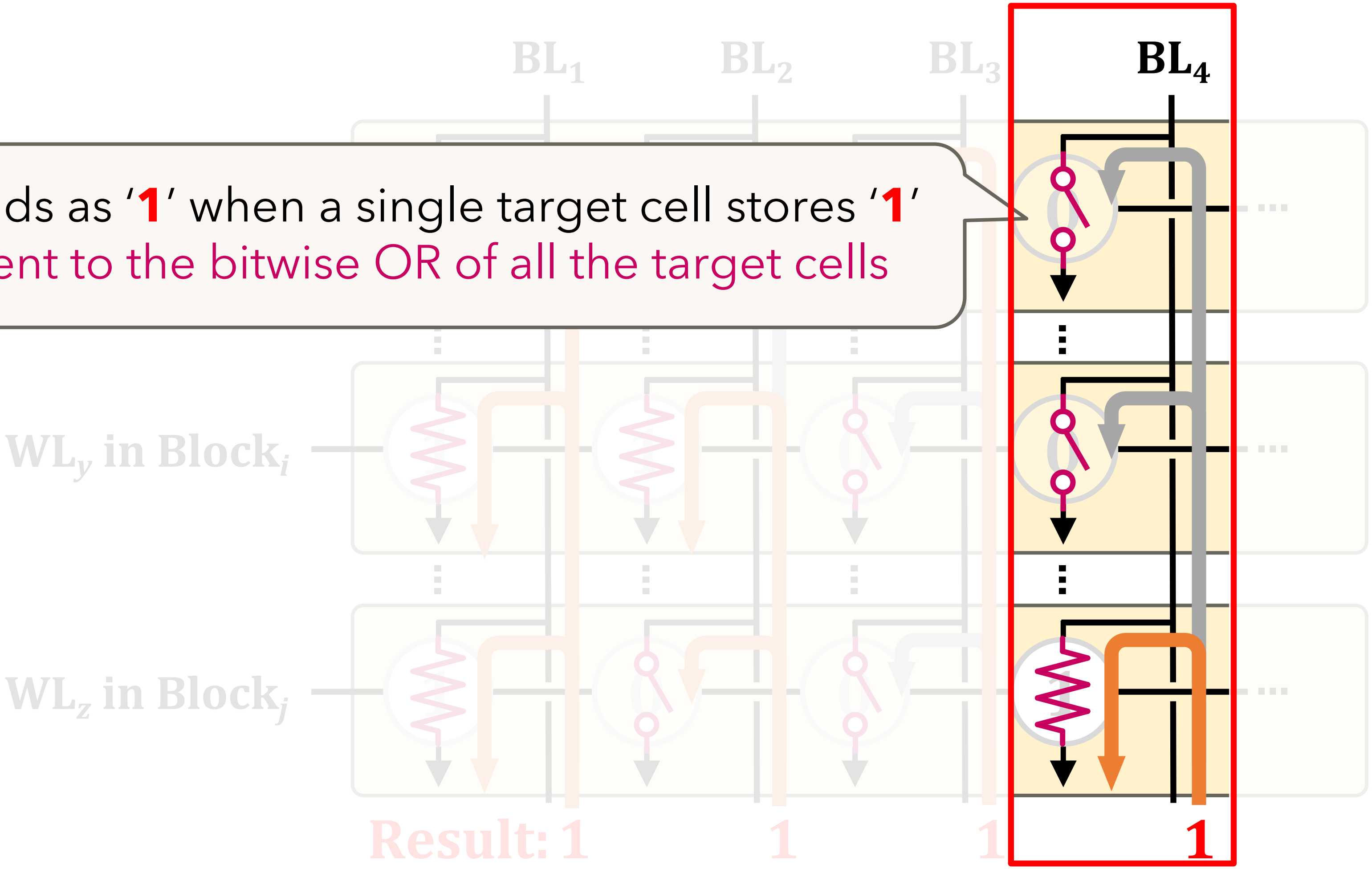
Multi-Wordline Sensing (MWS): Bitwise OR

- Inter-Block MWS:
  Simultaneously activates multiple WLs in different blocks
  → Bitwise OR of the stored data in the WLs



Multi-Wordline Sensing (MWS): Bitwise OR

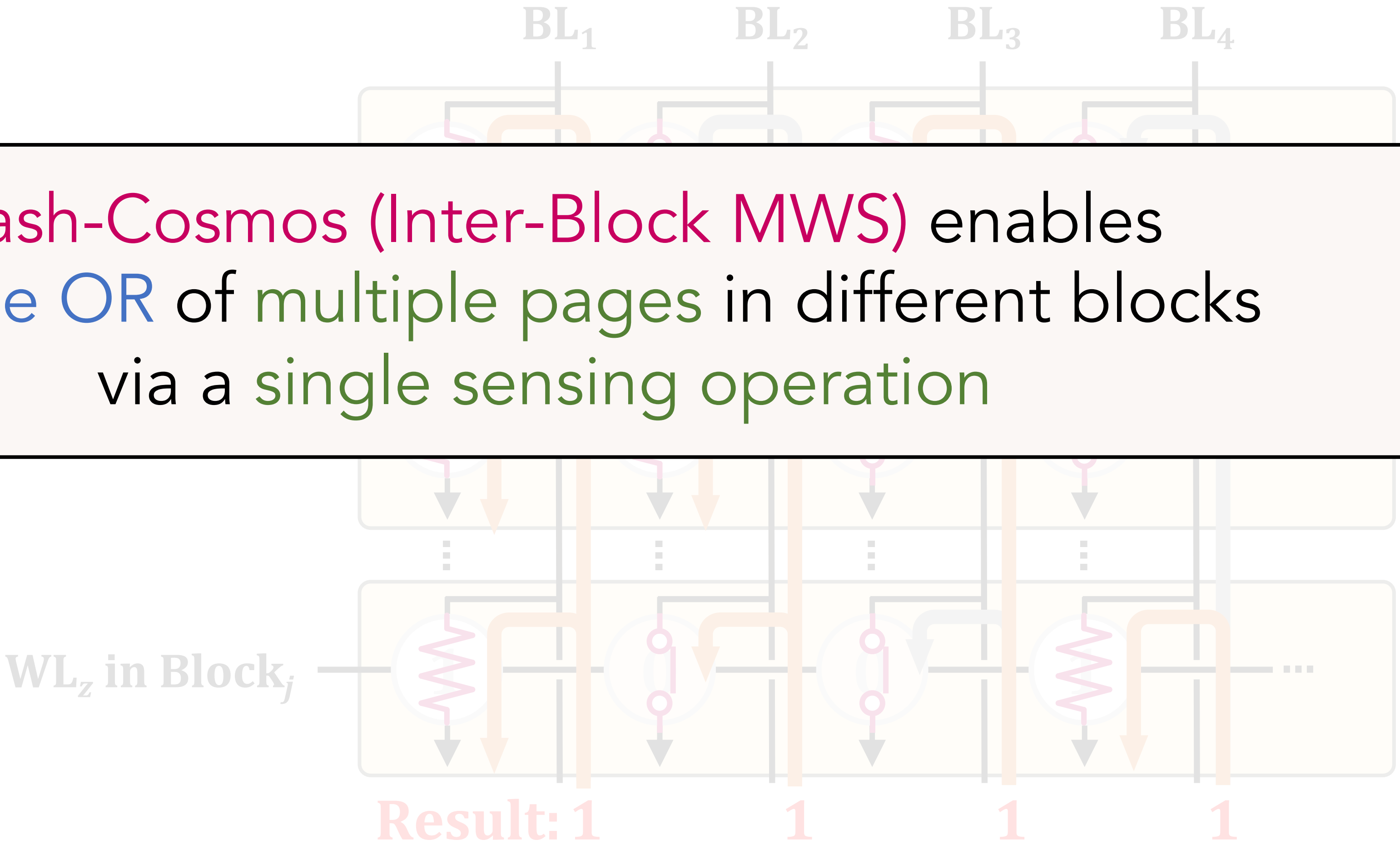
- Inter-Block MWS:
  Simultaneously activates multiple WLs in different blocks
  → Bitwise OR of the stored data in the WLs



A bitline reads as '1' when a single target cell stores '1'
→ Equivalent to the bitwise OR of all the target cells

Multi-Wordline Sensing (MWS): Bitwise OR

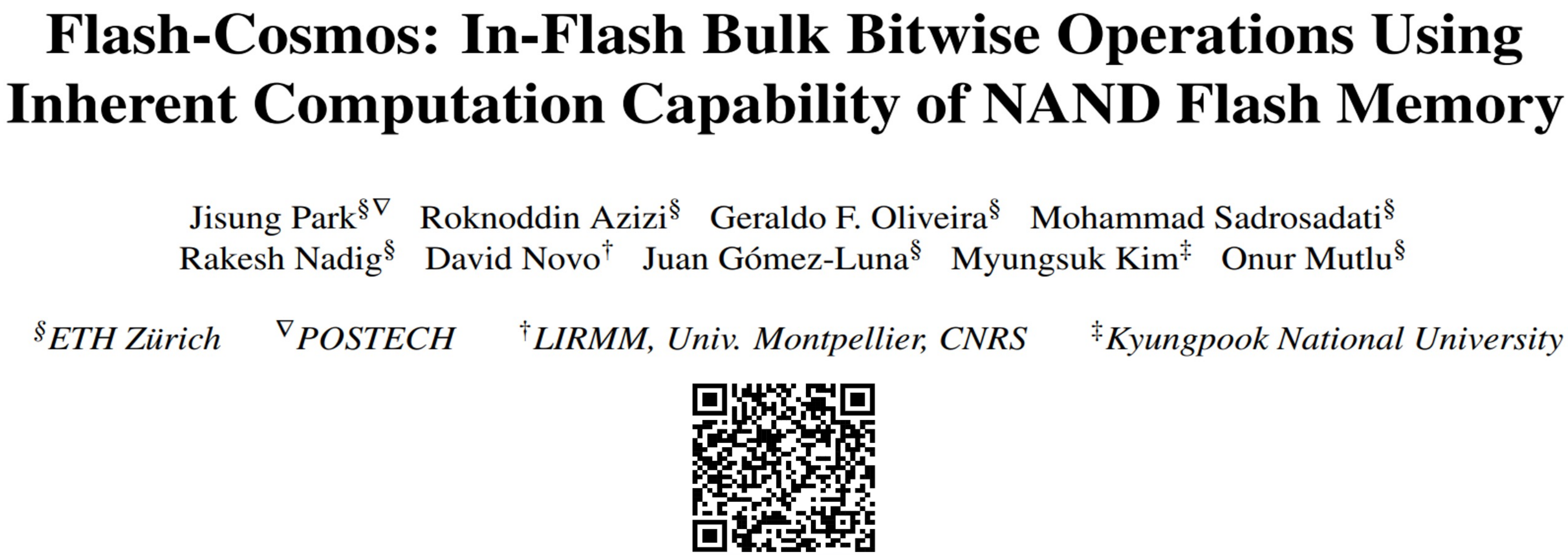
- Inter-Block MWS:
  Simultaneously activates multiple WLs in different blocks
  → Bitwise OR of the stored data in the WLs



Flash-Cosmos (Inter-Block MWS) enables bitwise OR of multiple pages in different blocks via a single sensing operation

Other Types of Bitwise Operations

Flash-Cosmos also enables
other types of bitwise operations
(NOT/NAND/NOR/XOR/XNOR)
leveraging existing features of NAND flash memory



Flash-Cosmos: In-Flash Bulk Bitwise Operations Using Inherent Computation Capability of NAND Flash Memory

Jisung Park[§∇] Roknoddin Azizi[§] Geraldo F. Oliveira[§] Mohammad Sadrosadati[§]
Rakesh Nadig[§] David Novo[†] Juan Gómez-Luna[§] Myungsuk Kim[‡] Onur Mutlu[§]

[§]*ETH Zürich* [∇]*POSTECH* [†]*LIRMM, Univ. Montpellier, CNRS* [‡]*Kyungpook National University*

https://arxiv.org/abs/2209.05566.pdf

Key Ideas

Multi-Wordline Sensing (MWS)
to enable in-flash bulk bitwise operations
via a single sensing operation

Enhanced SLC-Mode Programming (ESP)
to eliminate raw bit errors in stored data
(and thus in computation results)

Enhanced SLC-Mode Programming (ESP)

- Goal: eliminate raw bit errors in stored data (and computation results)

- Key ideas
  - Programs only a single bit per cell (SLC-mode programming)
    - Trades storage density for reliable computation
  - Performs more precise programming of the cells
    - Trades programming latency for reliable computation

Maximizes the reliability margin between the different states of flash cells

Enhanced SLC-Mode Programming (ESP)

- To eliminate raw bit errors in stored data (and computation results)

Flash-Cosmos (ESP) enables
reliable in-flash computation
by trading storage density & programming latency

Storage & latency overheads affect
only data used in in-flash computation

Talk Outline

- Problem, Goals & Key Idea
- Background
- Flash-Cosmos: Computation with One-Shot Multi-Operand Sensing
- Evaluation
- Summary

Evaluation Methodology

- Real-device characterization
  - To validate the feasibility and reliability of Flash-Cosmos
  - Using 160 48-WL-layer 3D Triple-Level Cell NAND flash chips
    - 3,686,400 tested wordlines
  - Under worst-case operating conditions
    - Under a 1-year retention time at 10K P/E cycles
    - Worst-case data patterns

- System-level evaluation
  - Using the state-of-the-art SSD simulator (MQSim [Tavakkol+, FAST'18])
  - Three real-world applications
    - Bitmap Indices (BMI): Bitwise AND of up to ~1,000 operands
    - Image Segmentation (IMS): Bitwise AND of 3 operands
    - K-clique Star Listing (KCS): Bitwise OR of up to 32 operands
  - Baselines
    - Outside-Storage Processing (OSP): A multi-core CPU (Intel i7-11700K)
    - In-Storage Processing (ISP): An in-storage hardware accelerator
    - ParaBit [Gao+, MICRO'21]: State-of-the-art in-flash processing mechanism

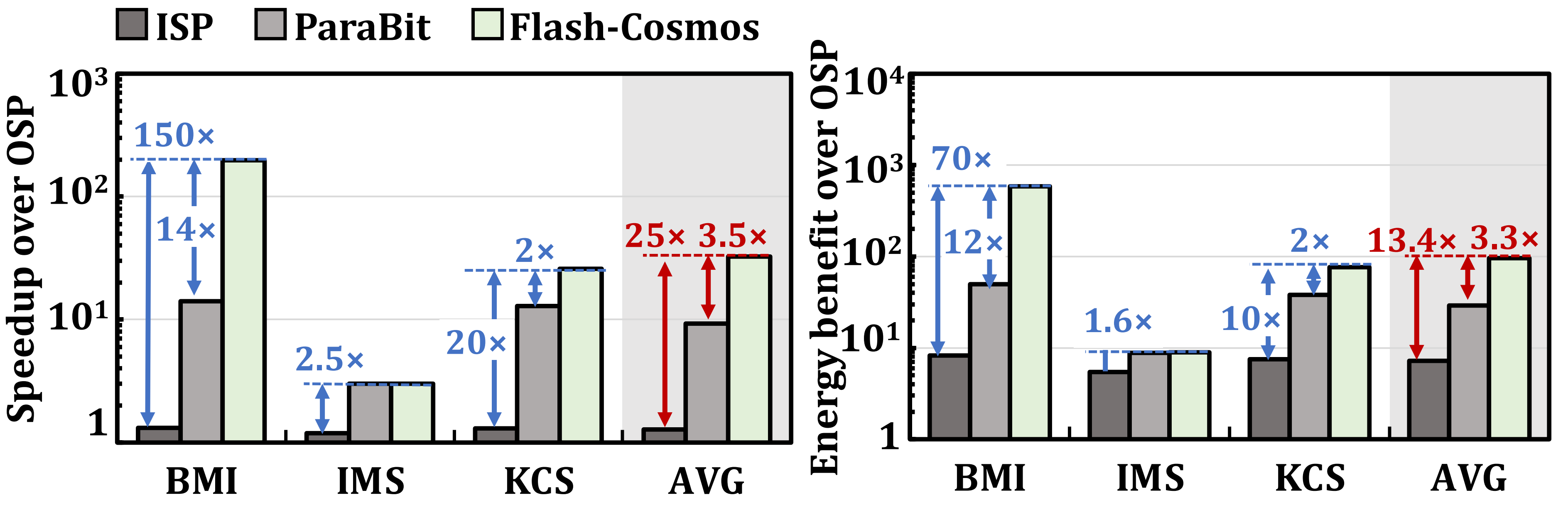
Results: Real-Device Characterization

Both intra- and inter-block MWS operations require no changes to the cell array of commodity NAND flash chips

Both MWS operations can activate multiple WLs (intra: up to 48, inter: up to 4) at the same time with small increase in sensing latency (< 10%)

ESP significantly improves the reliability of computation results (no observed bit error in the tested flash cells)

Results: Performance & Energy



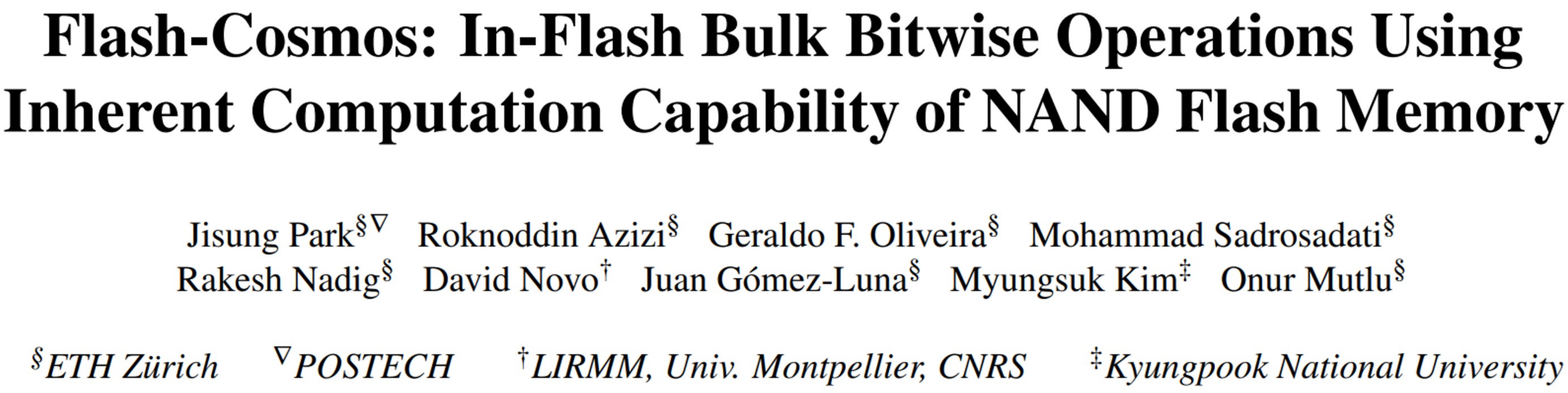
Flash-Cosmos provides significant performance & energy benefits over all the baselines

The larger the number of operands, the higher the performance & energy benefits

More in the Paper

- Support for other types of bitwise operations
  - NOT/NAND/NOR/XOR/XNOR
- More detailed real-device characterization results
- Optimizations to improve bitwise operation performance
- Flash-Cosmos command interface
- System support
- Overhead analysis

More in the Paper



Flash-Cosmos: In-Flash Bulk Bitwise Operations Using Inherent Computation Capability of NAND Flash Memory

Jisung Park[§∇] Roknoddin Azizi[§] Geraldo F. Oliveira[§] Mohammad Sadrosadati[§]
Rakesh Nadig[§] David Novo[†] Juan Gómez-Luna[§] Myungsuk Kim[‡] Onur Mutlu[§]

[§]*ETH Zürich* [∇]*POSTECH* [†]*LIRMM, Univ. Montpellier, CNRS* [‡]*Kyungpook National University*

https://arxiv.org/abs/2209.05566.pdf

Summary: Flash-Cosmos

The first work that enables in-flash multi-operand bulk bitwise operations with a single sensing operation and high reliability

Improves performance by 32x/25x/3.5x over OSP/ISP/ParaBit

Improves energy efficiency by 95x/13.4x/3.3x over OSP/ISP/ParaBit

Low-cost & requires no changes to flash cell arrays

Flash-Cosmos

In-Flash Bulk Bitwise Operations Using Inherent Computation Capability of NAND Flash Memory

Jisung Park, Roknoddin Azizi, Geraldo F. Oliveira, Mohammad Sadrosadati,
Rakesh Nadig, David Novo, Juan Gómez Luna, Myungsuk Kim, and Onur Mutlu

MICRO 2022