CMPE 214
GPU Architecture & Programming

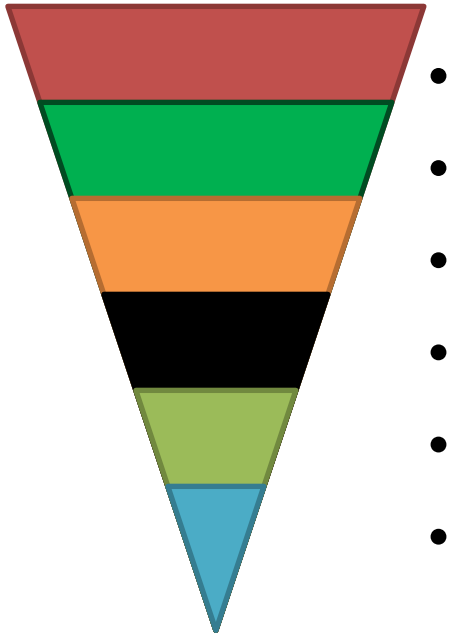
Lecture 1.
GPU Architecture Overview (4)

Haonan Wang

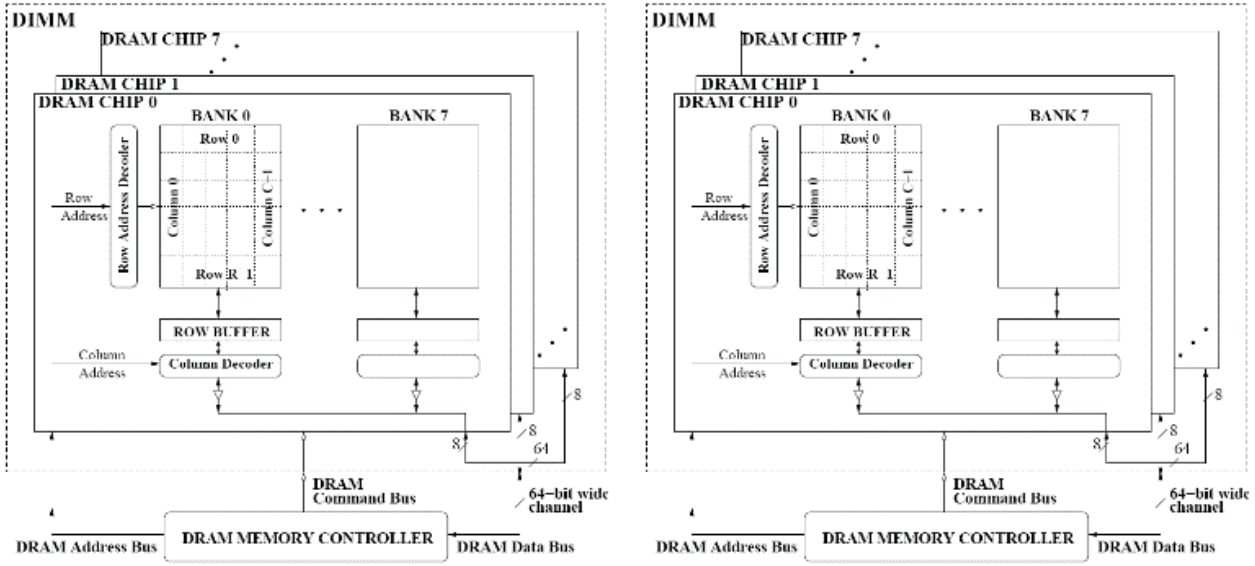
SAN JOSÉ STATE
UNIVERSITY

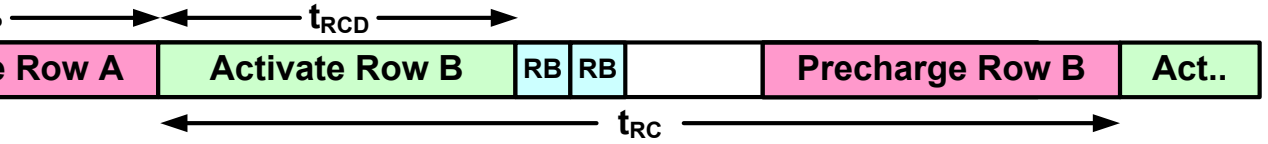
GPU Microarchitecture: Memory Partition

DRAM Organization:

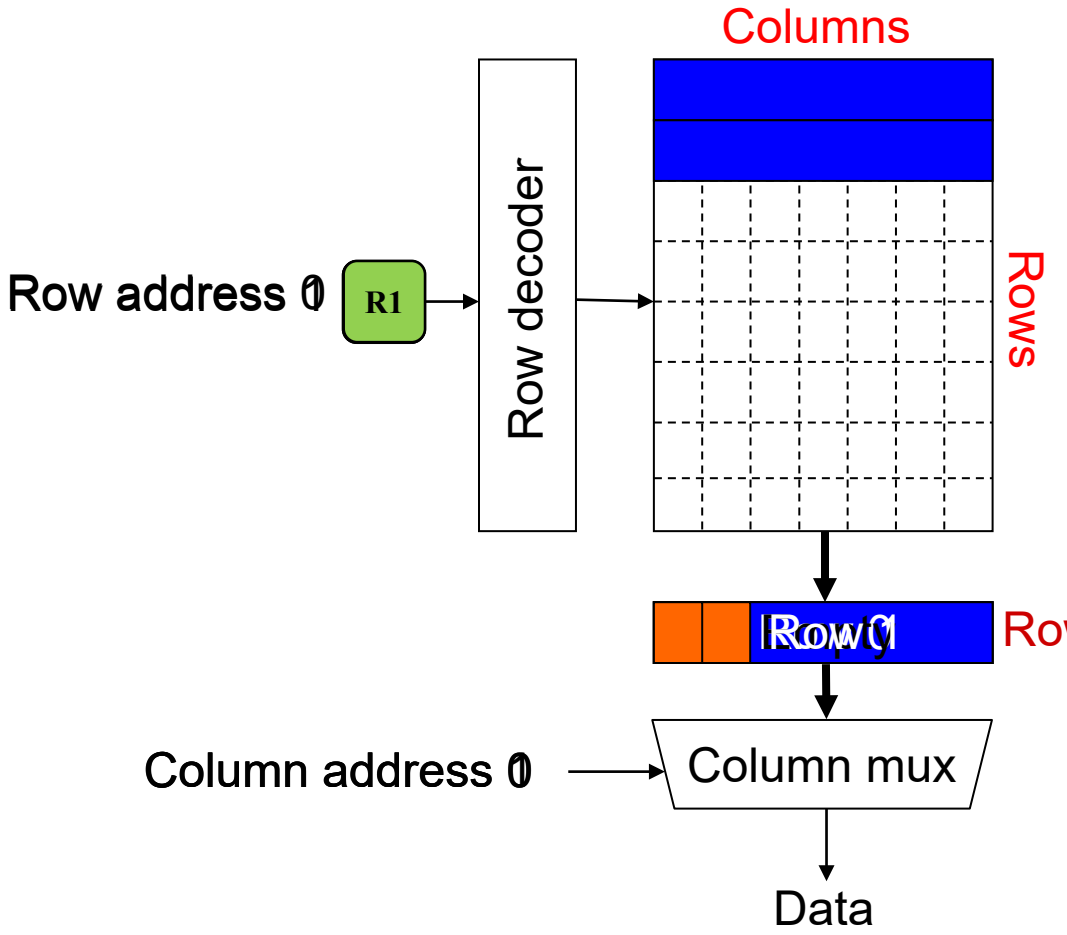


- Channel
- DIMM
- Rank
- Chip
- Bank
- Row/Column





Row Operations & Row Buffer Locality



Access Address:	Row Operation:
RBL=2 (Row 0, Column 0)	Activation
RBL=2 (Row 0, Column 1)	No operation
RBL=1 (Row 1, Column 0)	Restore, Precharge, Activation

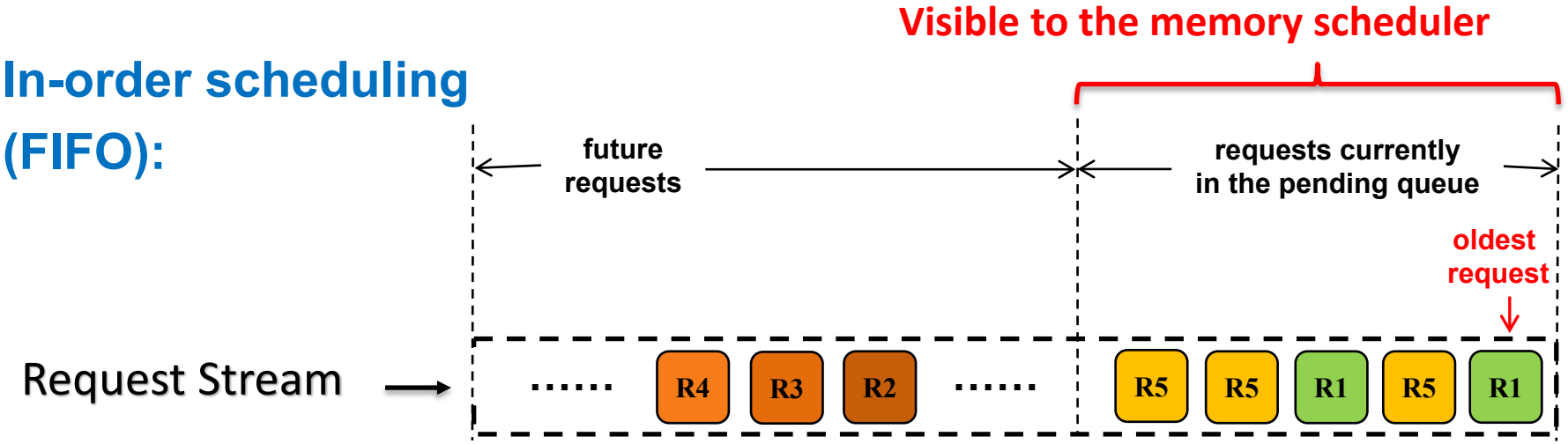
CONFLICT !

Improving Row Buffer Locality (RBL) is the key to improve DRAM efficiency

Credit: Onur Mutlu.

RBL & Memory Scheduling Schemes

In-order scheduling (FIFO):



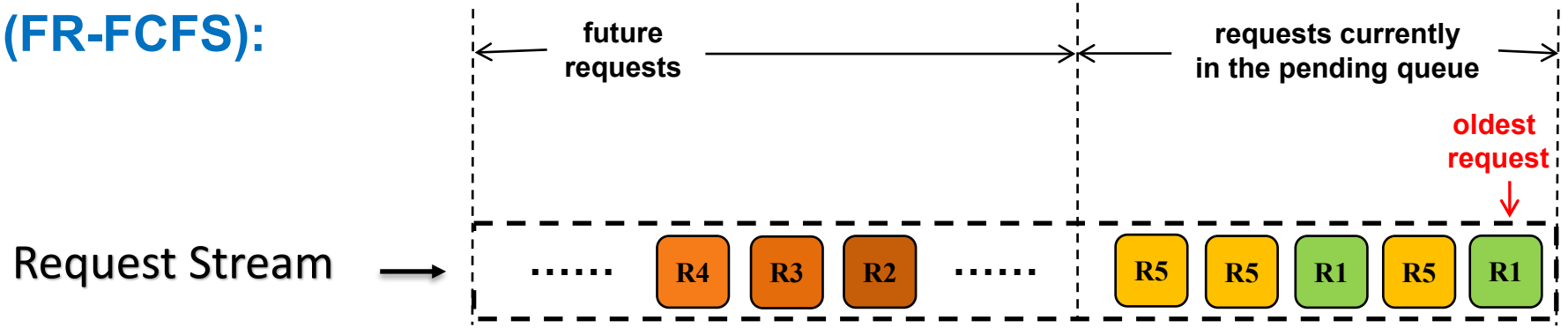
Activation Counter:

R1: Activation = 1
R5: Activation = 2
R1: Activation = 3
R5: Activation = 4
R5: Activation = 4

Same activation

Avg RBL = 5 / 4 = 1.25

Out-of-order scheduling (FR-FCFS):



Activation Counter:

R1: Activation = 1
R1: Activation = 1

Same activation

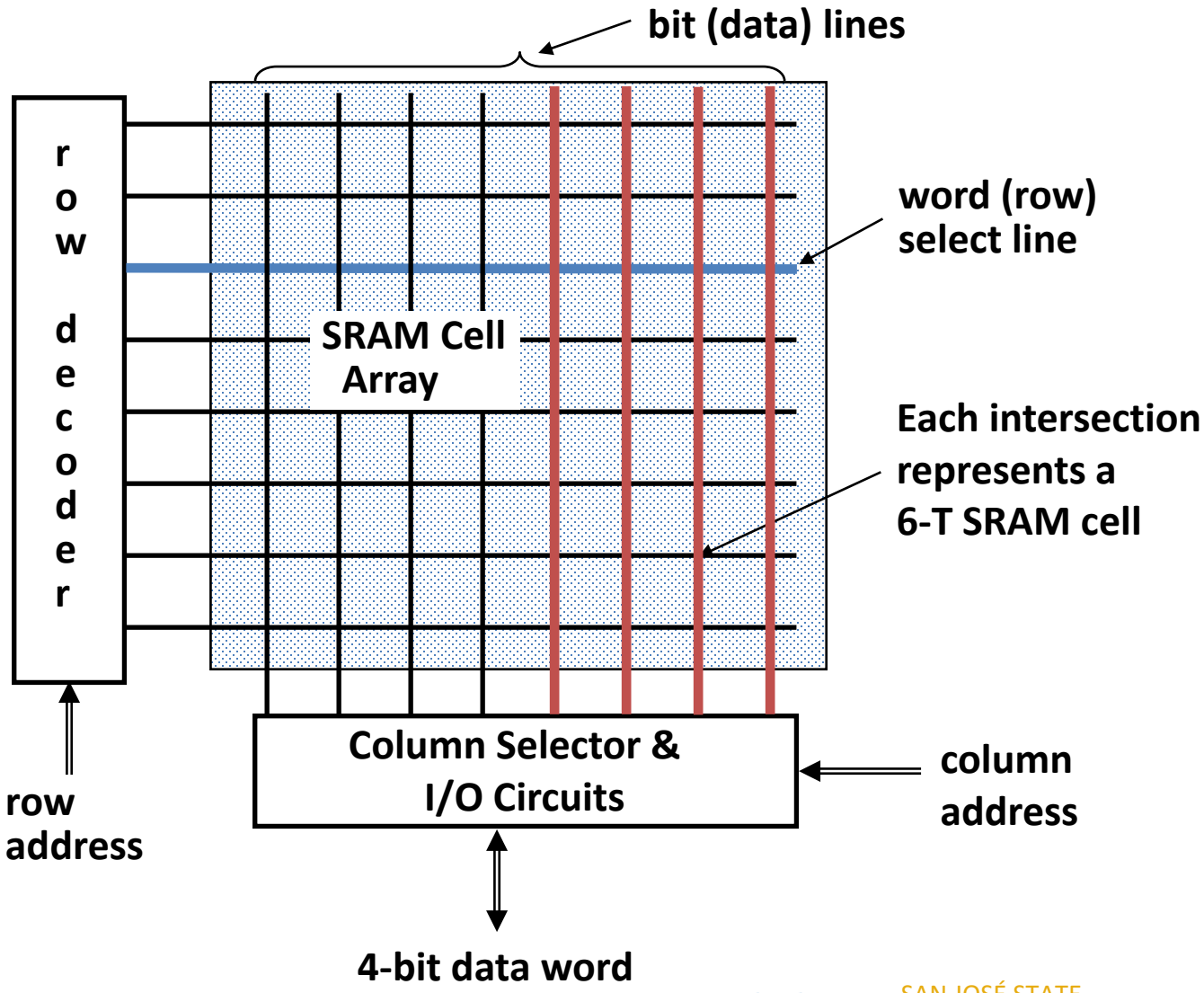
R5: Activation = 2
R5: Activation = 2
R5: Activation = 2

Same activation

Avg RBL = 5 / 2 = 2.5

SRAM Cache Design

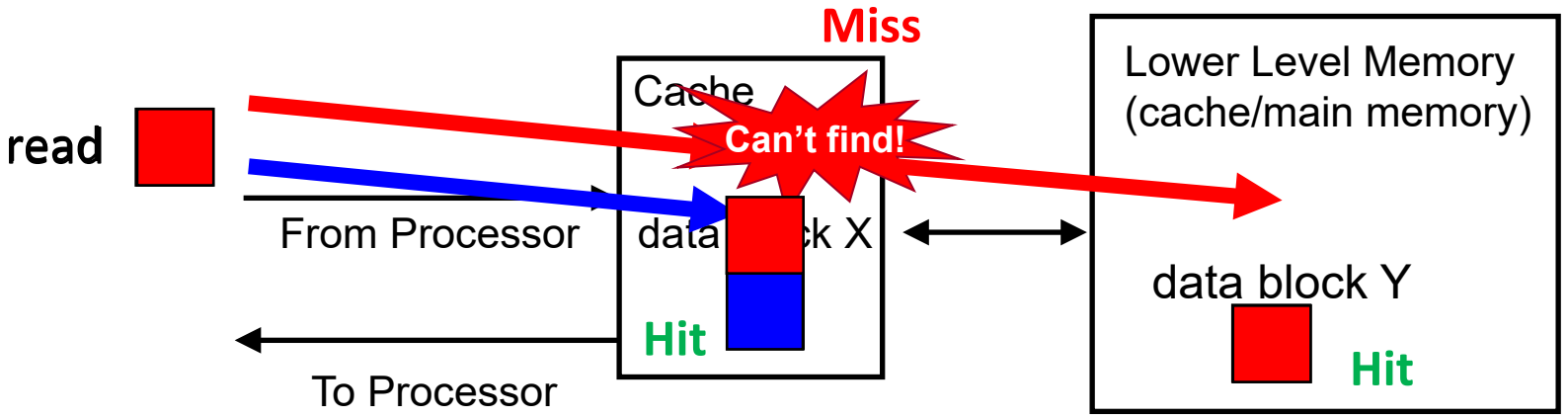
- Each row holds a data block
- Column address selects the requested word from block



Caches

- **Hit:** Data appears in some block of the cache
  - **Hit Rate:** # hits / total accesses on the cache
  - **Hit Time:** Time to access the cache

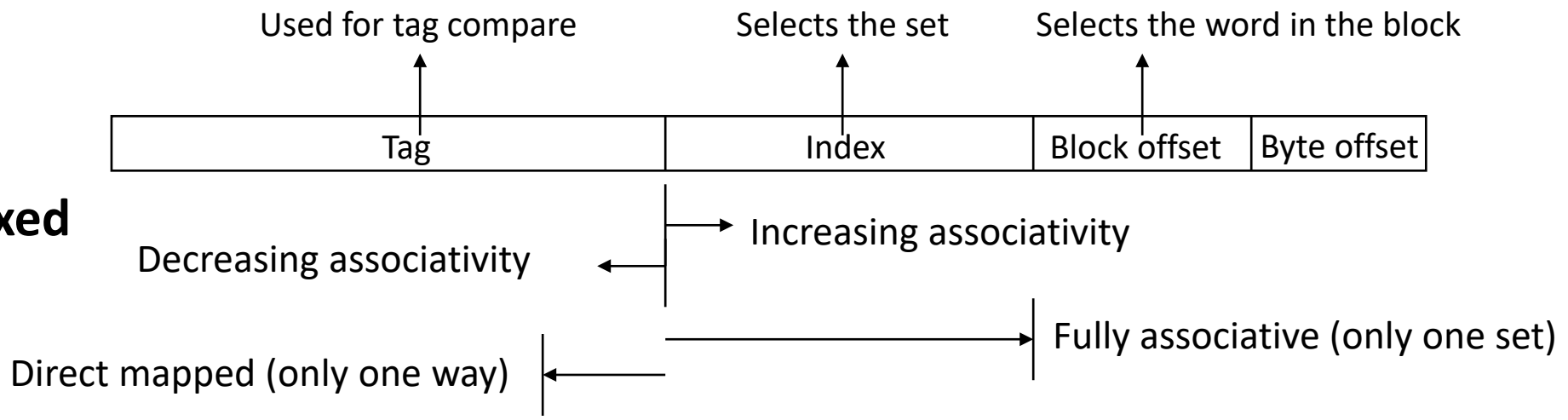
- **Miss:** Data needs to be retrieved from the lower level (and stored in cache)
  - **Miss Rate:** 1 - (Hit Rate)
  - **Miss Penalty:** Average delay in the processor caused by each miss



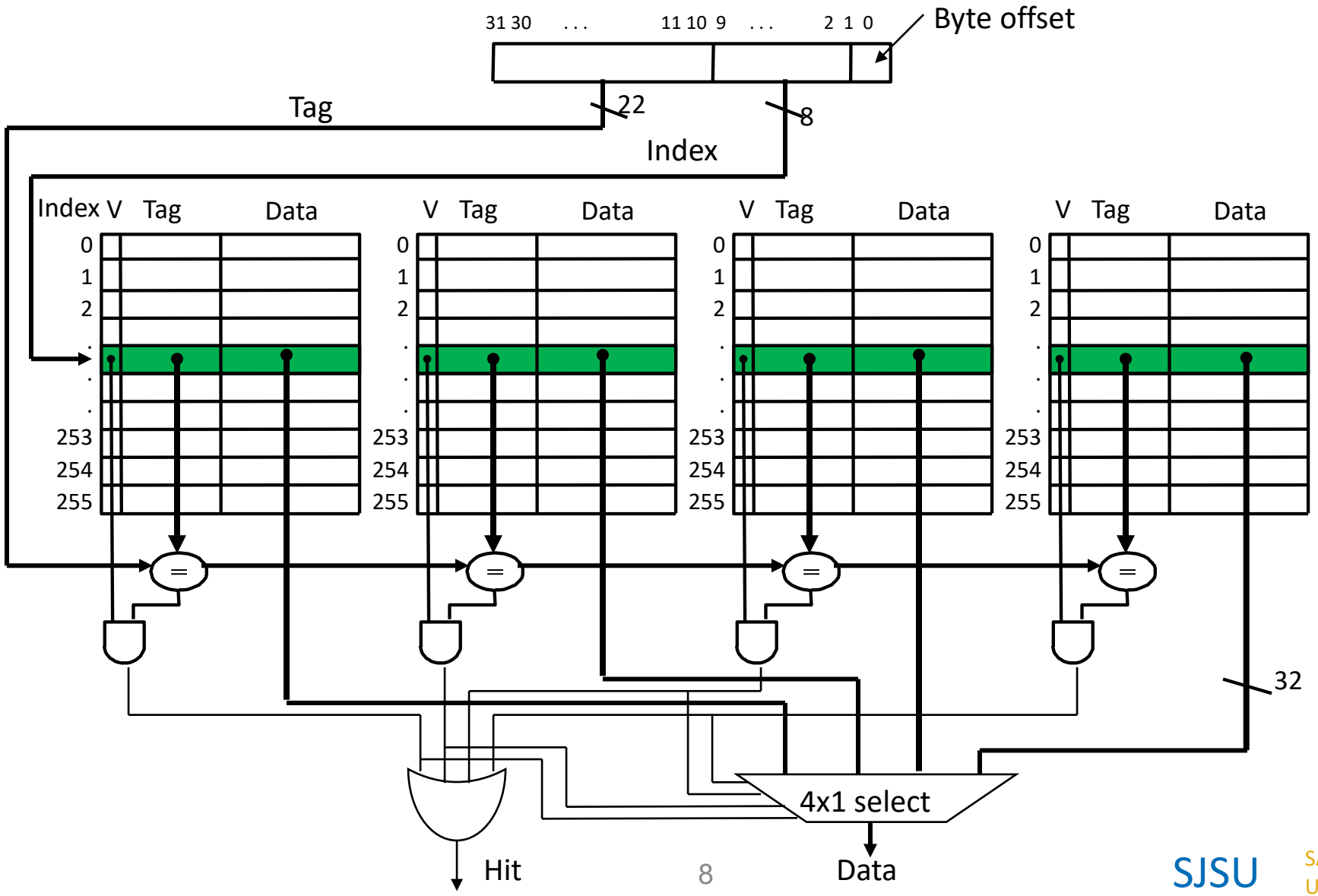
Cache Types

- **N-way Set-Associative:** Number of ways > 1 & Number of sets > 1
  - Slightly complex searching mechanism
- **Direct Mapped:** Number of ways = 1
  - Fast indexing mechanism
- **Fully-Associative:** Number of sets = 1
  - Extensive hardware resources required to search

	Way 0	Way 1	…
Set 0	block 0	block 2	
Set 1	block 1	block 3	
⋮			



Four-Way Set Associative Cache



Costs of Set Associative Caches

- **Must have hardware for replacement policy**
  - E.g., to keep track of when each way's block was used

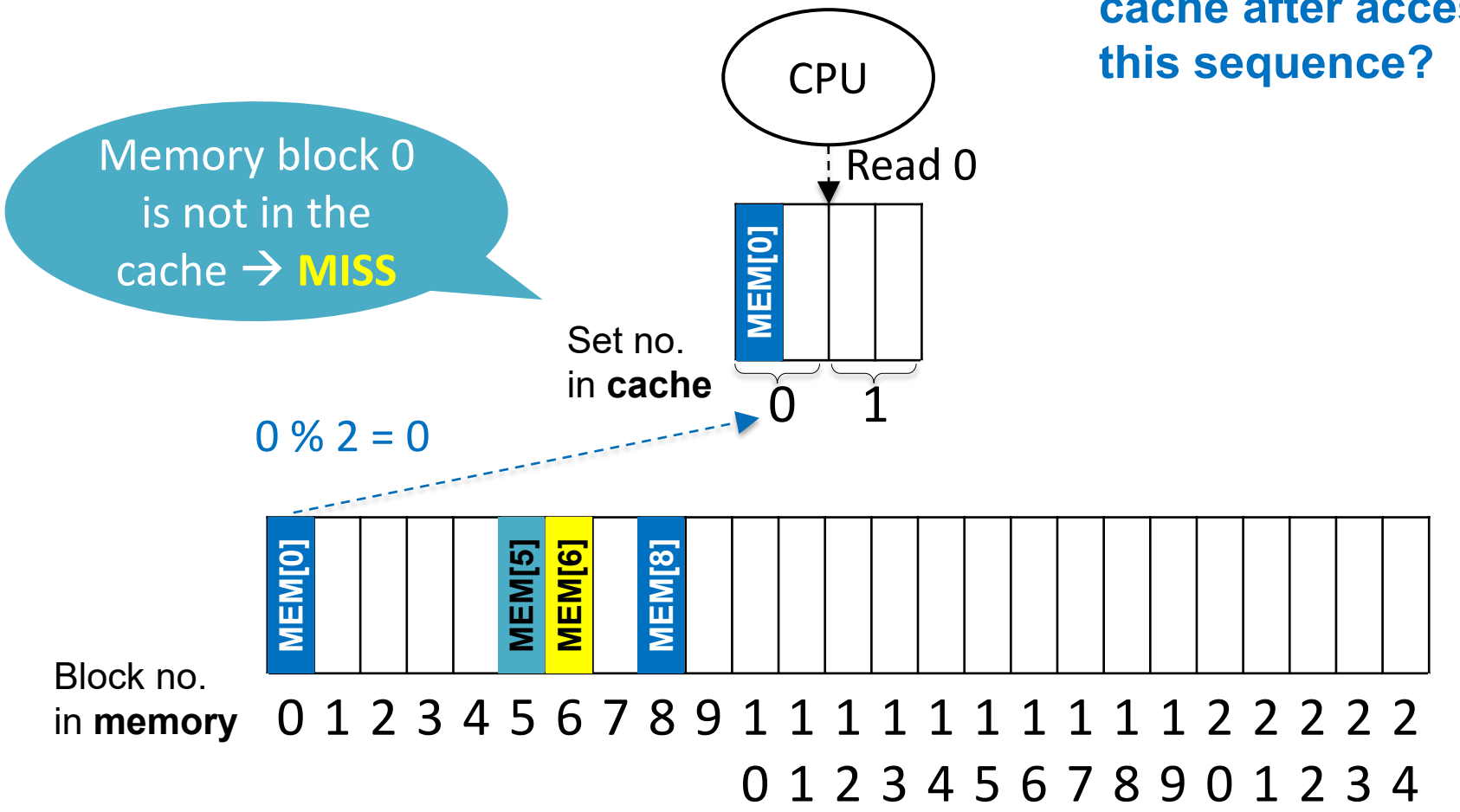
- **N-way set associative cache costs**
  - N comparators (delay and area) & MUX delay
    - Data is available after Hit/Miss decision.
    - In a direct mapped cache, the cache block is available before the Hit/Miss decision.

- **Total cache line size = valid field size + tag size + block data size + data for cache policy (e.g., time stamp, modified bit, etc.)**

Miss Example: 2-way

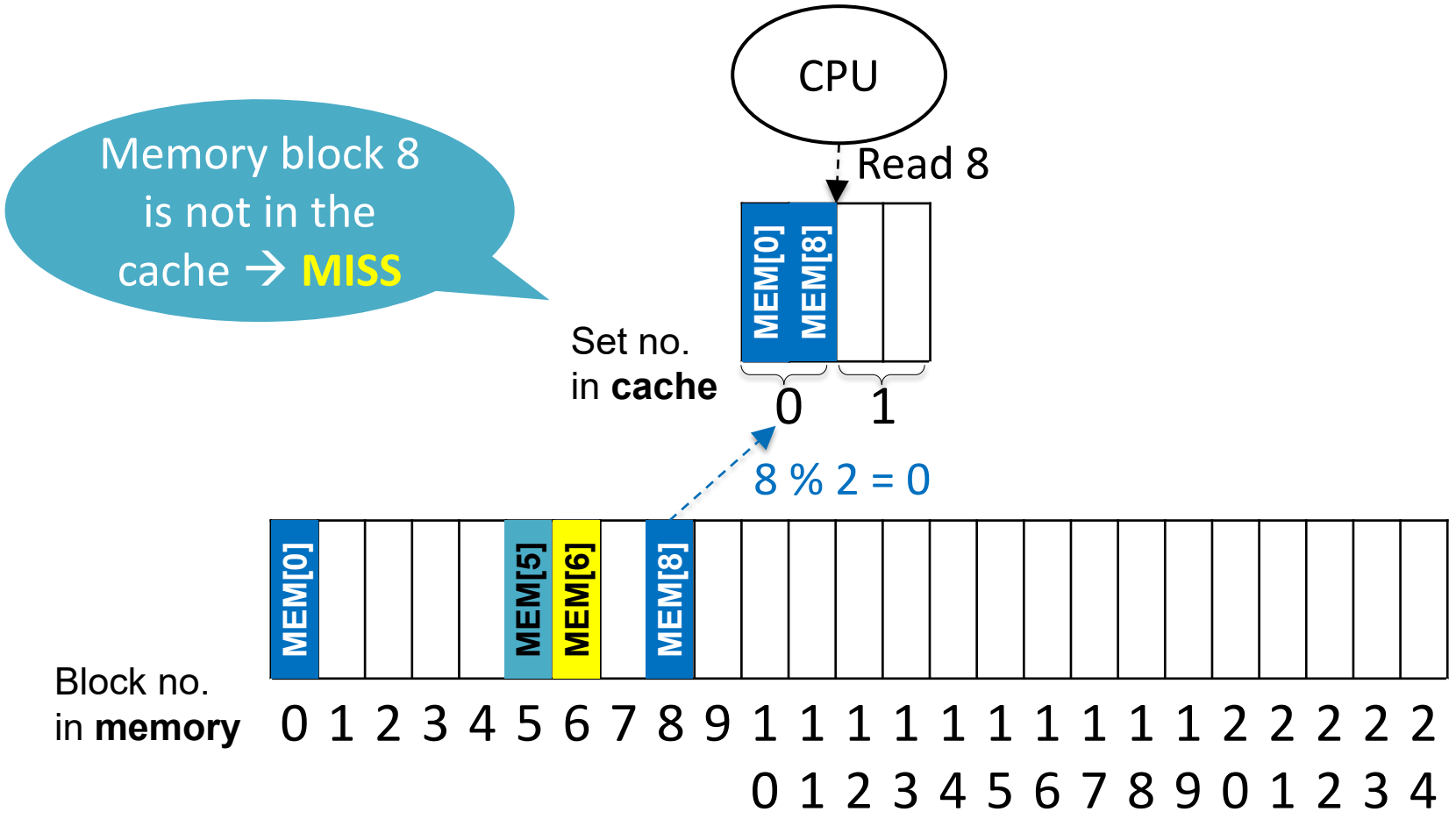
- **Memory block access sequence: 0, 8, 0, 6, 5**

Which data will be in the cache after accessing this sequence?



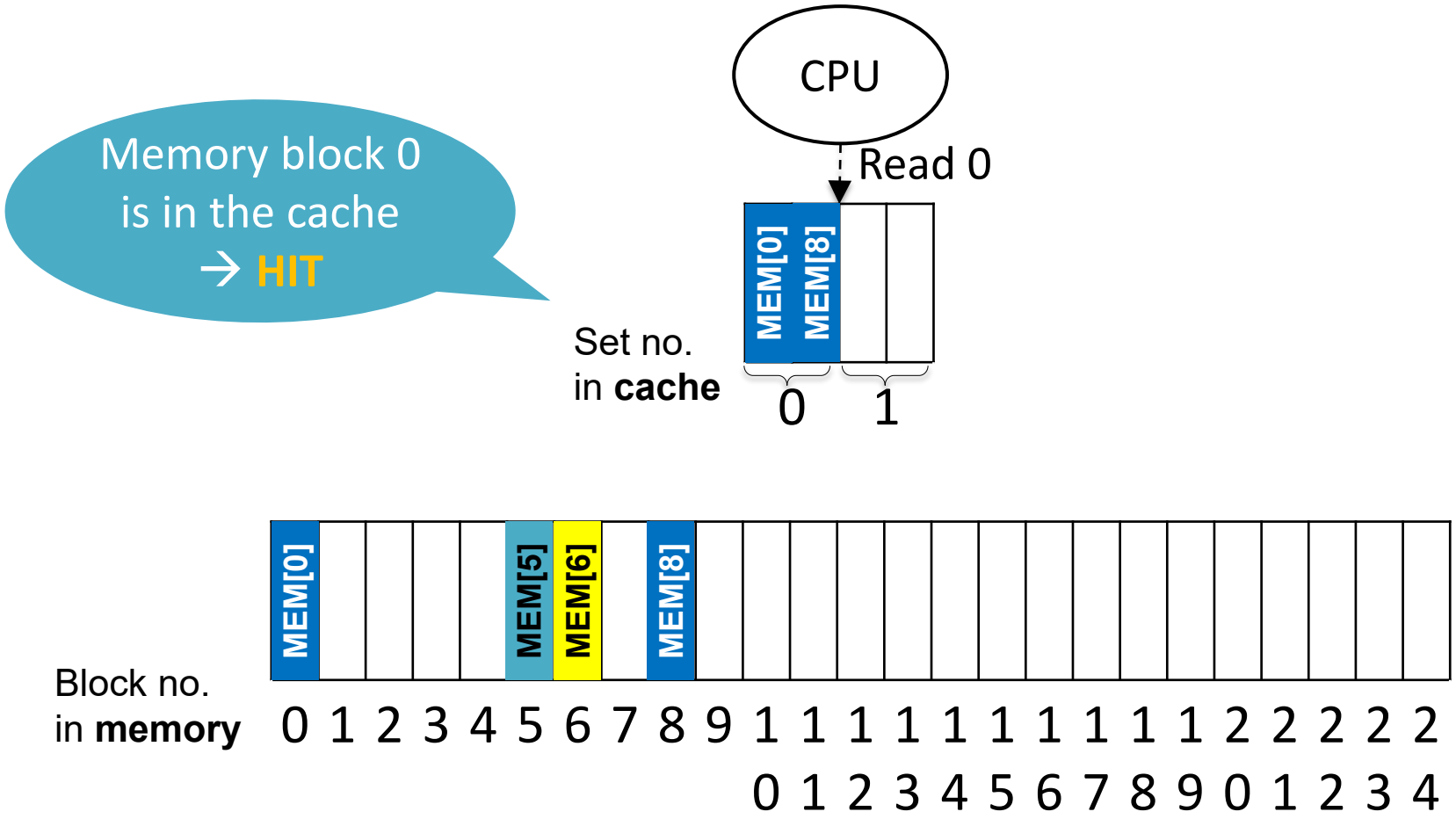
Miss Example: 2-way

- **Memory block access sequence: 0, 8, 0, 6, 5**



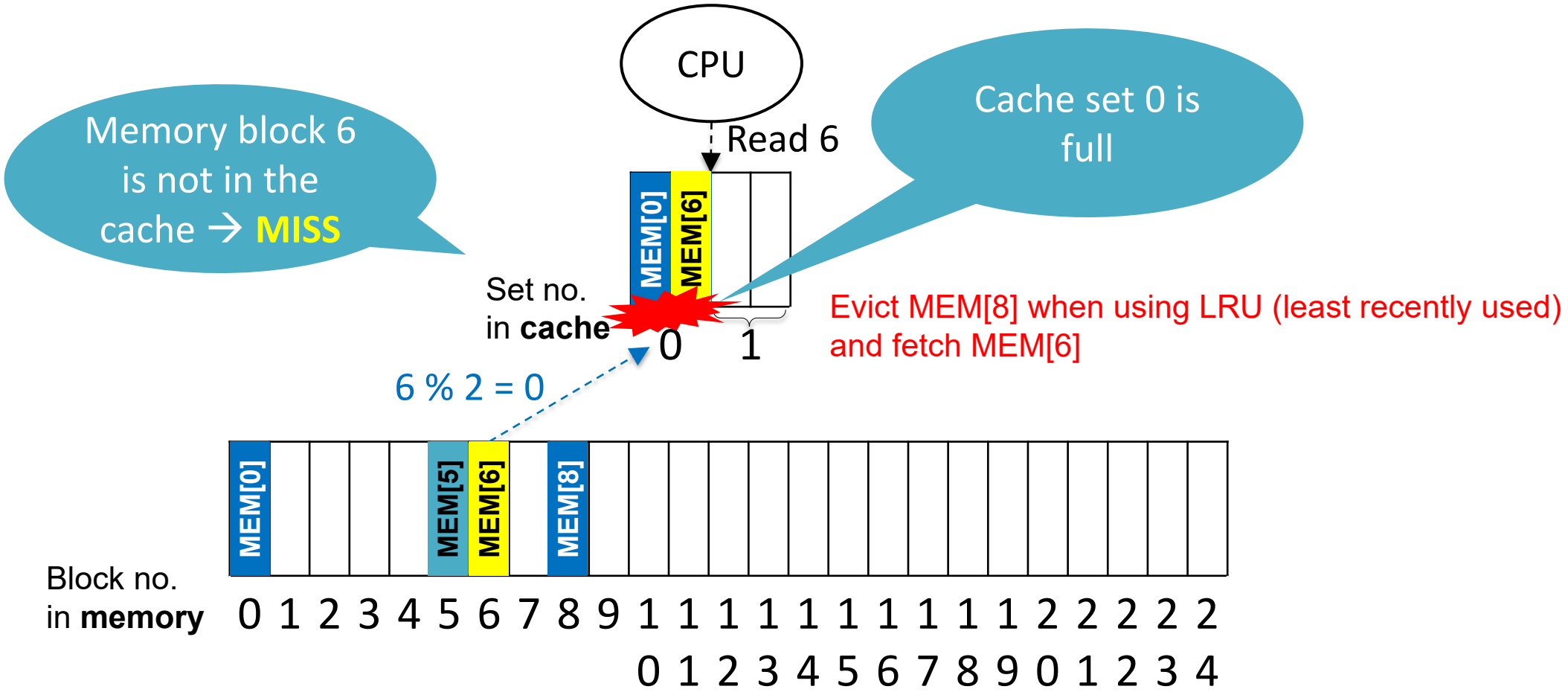
Miss Example: 2-way

- **Memory block access sequence: 0, 8, 0, 6, 5**



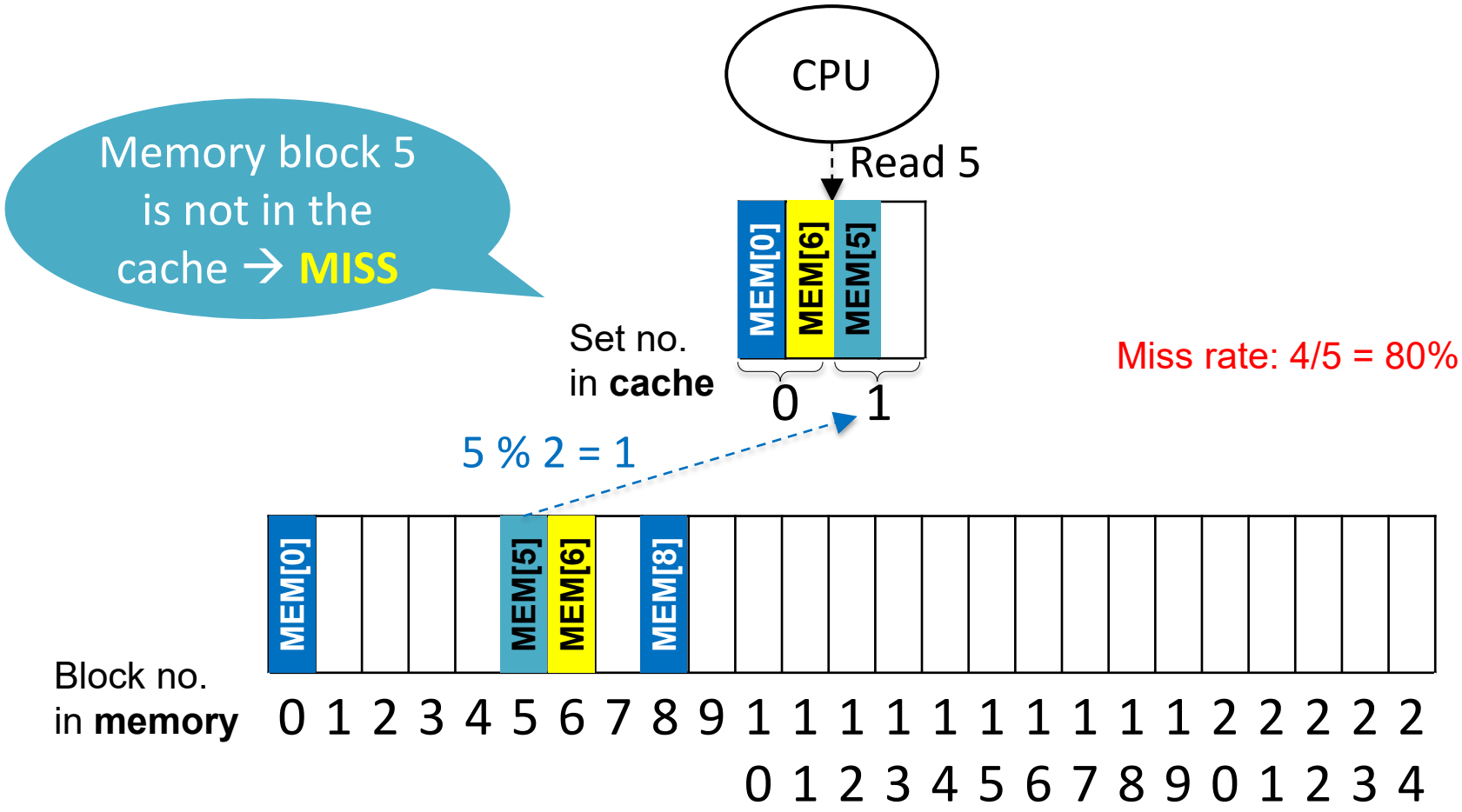
Miss Example: 2-way

- **Memory block access sequence: 0, 8, 0, 6, 5**



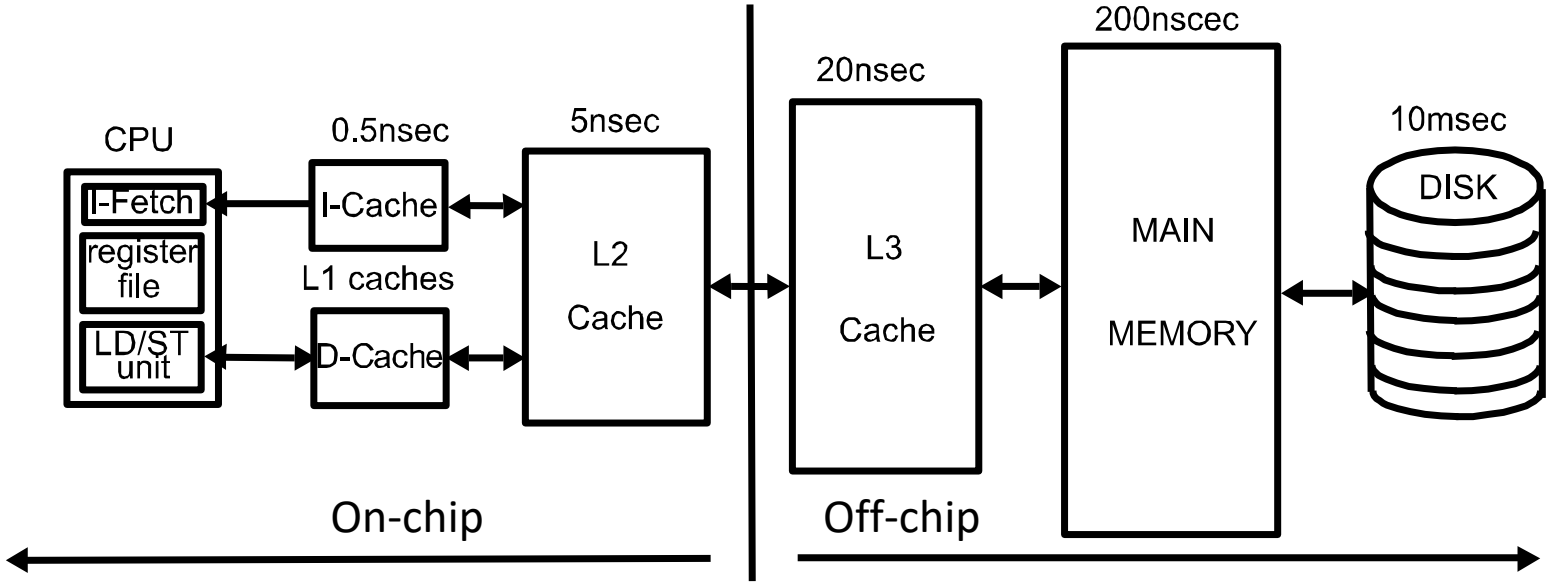
Miss Example: 2-way

- **Memory block access sequence: 0, 8, 0, 6, 5**

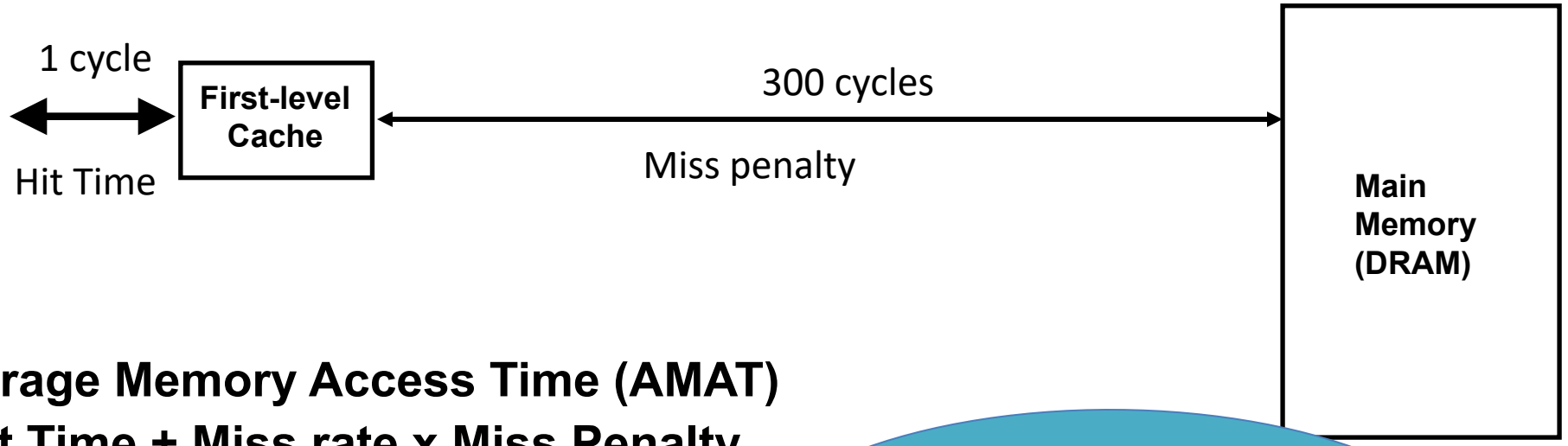


Principle of Locality

- **Temporal Locality**: If an address is referenced, it tends to be referenced again
- **Spatial Locality**: If an address is referenced, neighboring addresses tend to be referenced
- **How to create a memory system that gives the illusion of being large, cheap and fast?**
  - With hierarchy
  - With parallelism



Memory Hierarchy Performance

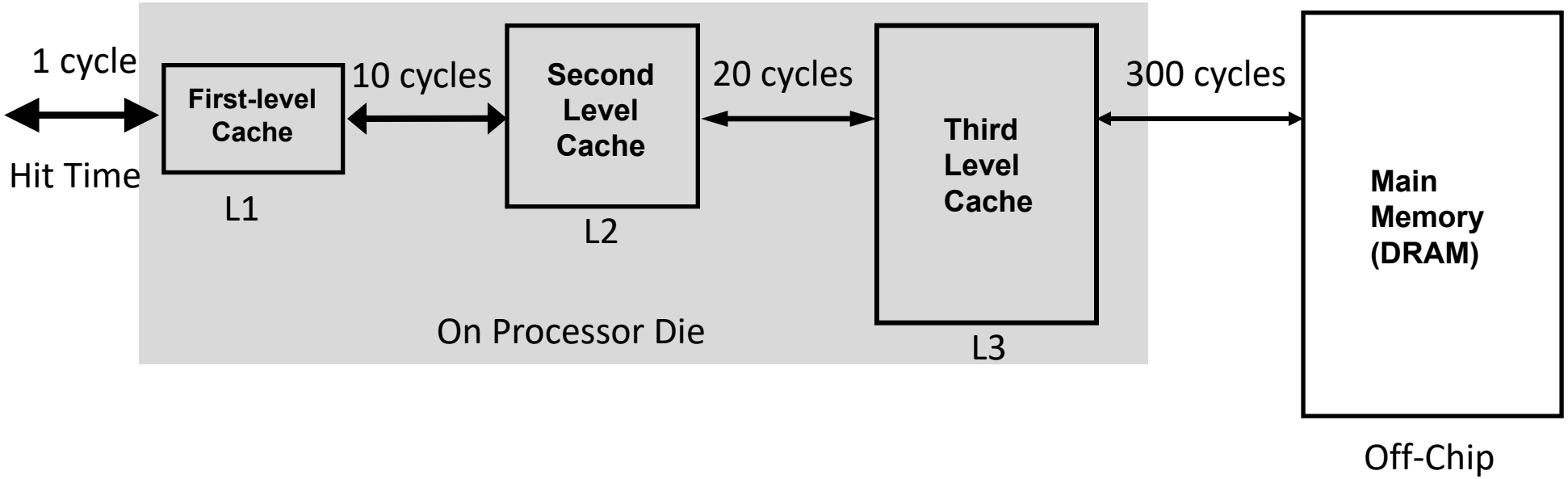


- **Average Memory Access Time (AMAT) = Hit Time + Miss rate x Miss Penalty**

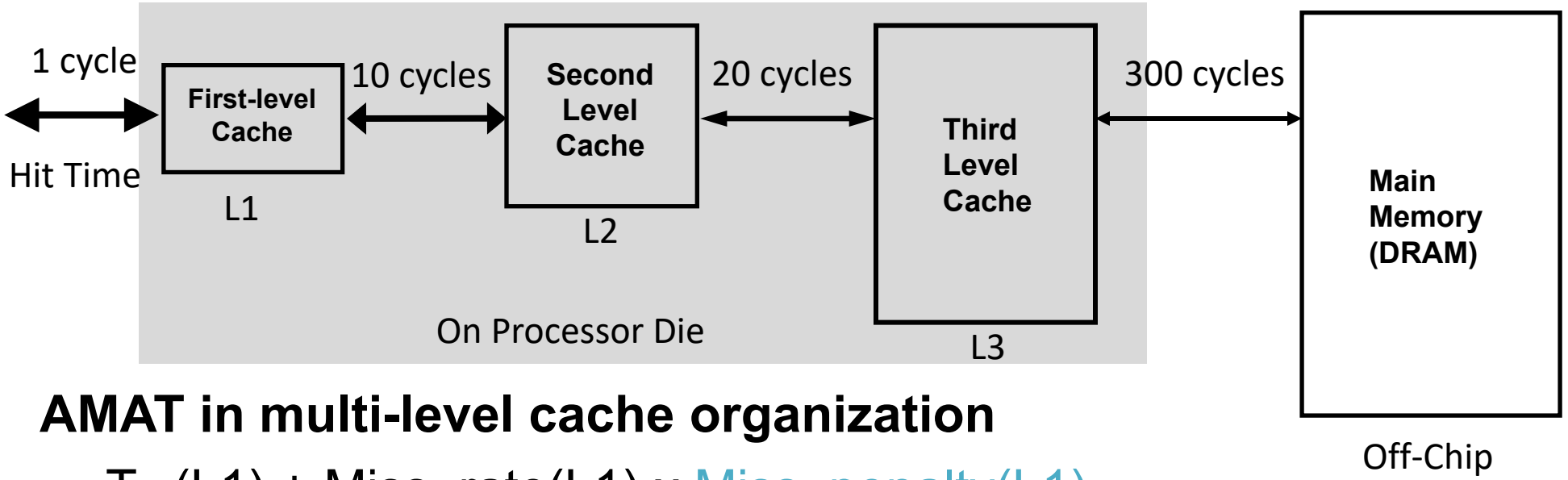
- **Example:**
  - Cache Hit = 1 cycle
  - Miss rate = 10% = 0.1
  - Miss penalty = 300 cycles
  - AMAT = T_{hit}(L1) + Miss_rate(L1) x T(Memory) = 1 + 0.1 x 300 = 31 cycles

Due to long-latency memory, AMAT is 30 cycles longer than cache latency. Can we reduce the overhead?

Reducing Penalty: Multi-Level Cache



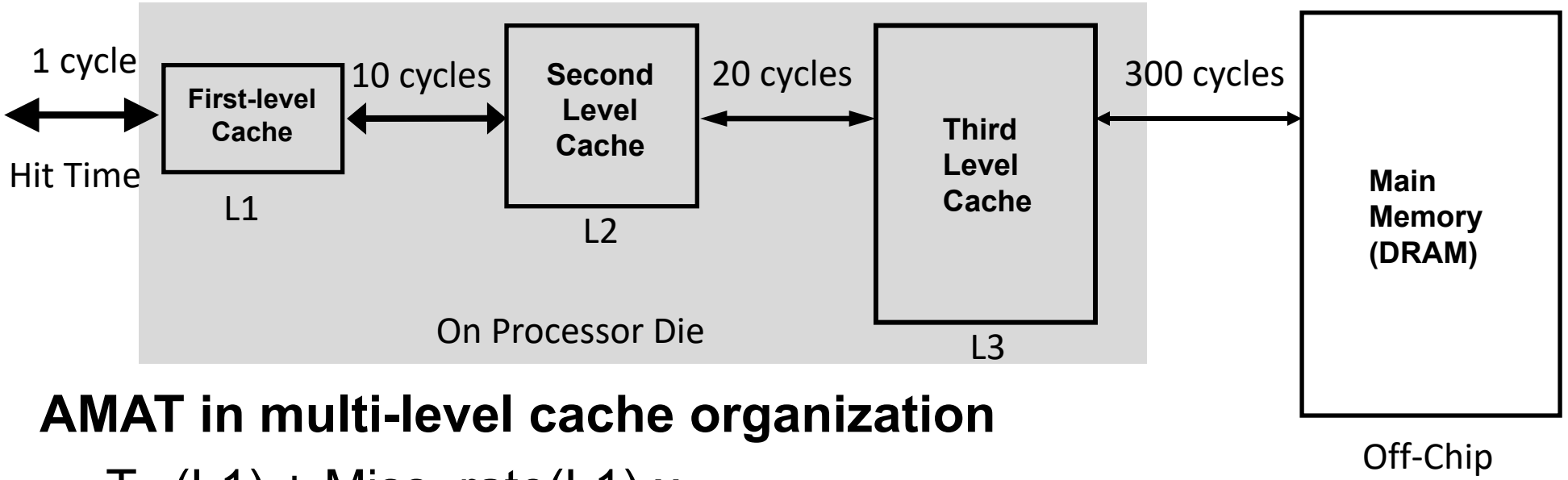
Reducing Penalty: Multi-Level Cache



- **AMAT in multi-level cache organization**

  = T_{hit}(L1) + Miss_rate(L1) x Miss_penalty(L1)

Reducing Penalty: Multi-Level Cache

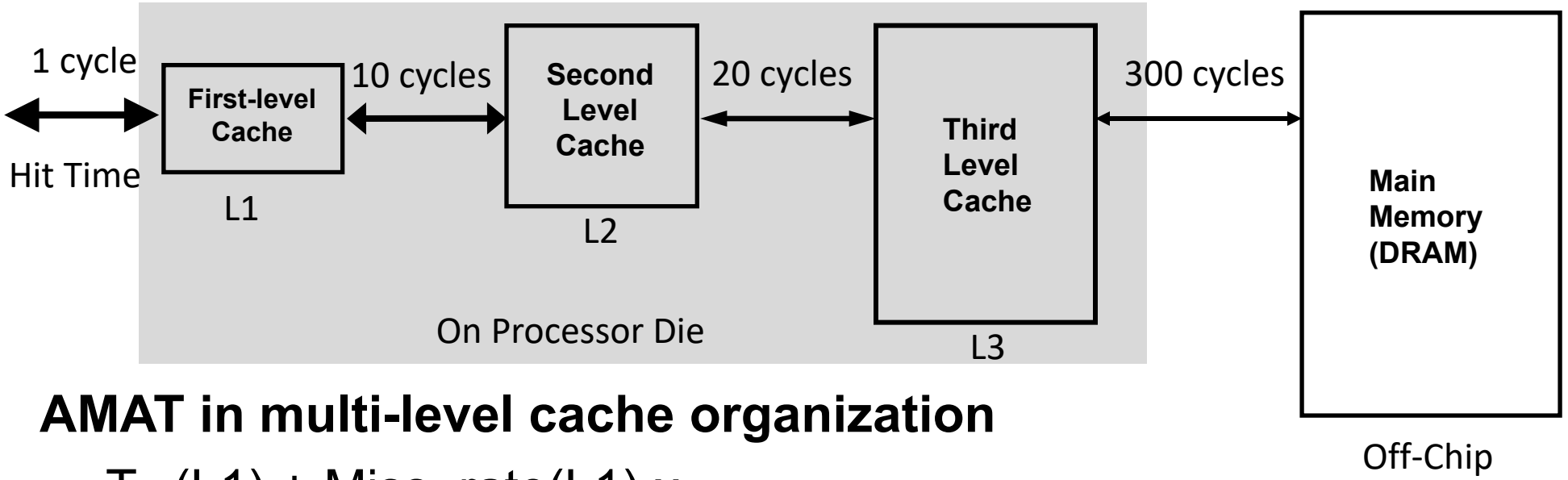


- **AMAT in multi-level cache organization**

  $= T_{hit}(L1) + Miss\_rate(L1) \times$

  $[\ T_{hit}(L2) + Miss\_rate(L2) \times Miss\_penalty(L2)\ ]$

Reducing Penalty: Multi-Level Cache



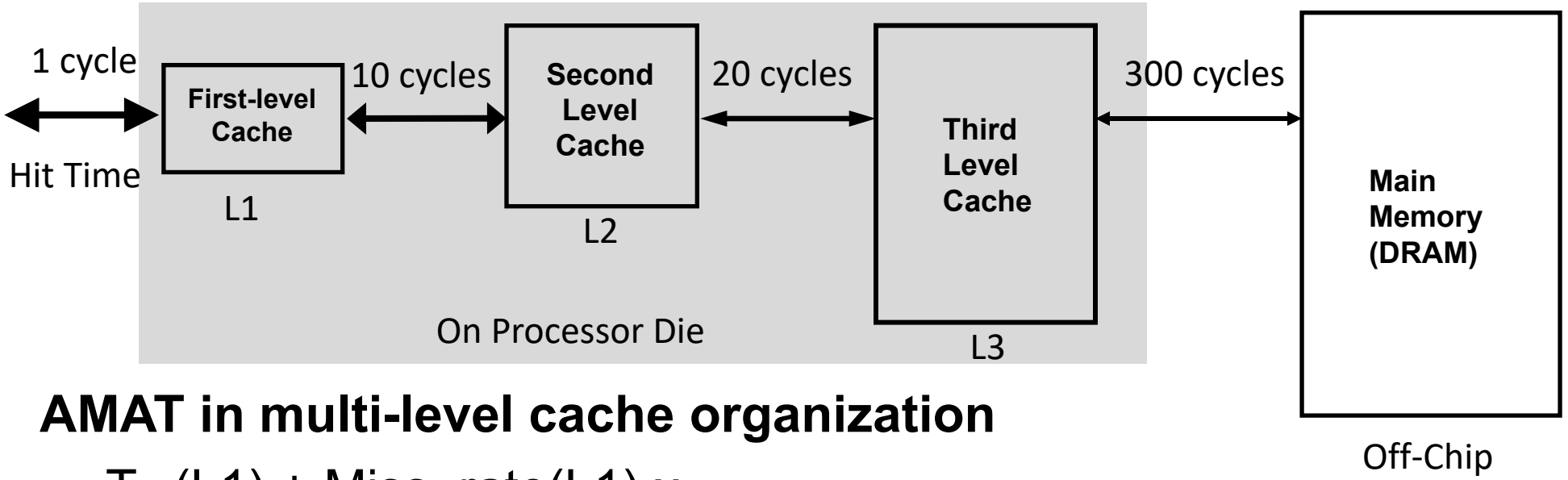
- **AMAT in multi-level cache organization**

  $= T_{hit}(L1) + Miss\_rate(L1) \times$

  $[\ T_{hit}(L2) + Miss\_rate(L2) \times$

  $\{\ T_{hit}(L3) + Miss\_rate(L3) \times Miss\_penalty(L3)\ \}\ ]$

Reducing Penalty: Multi-Level Cache



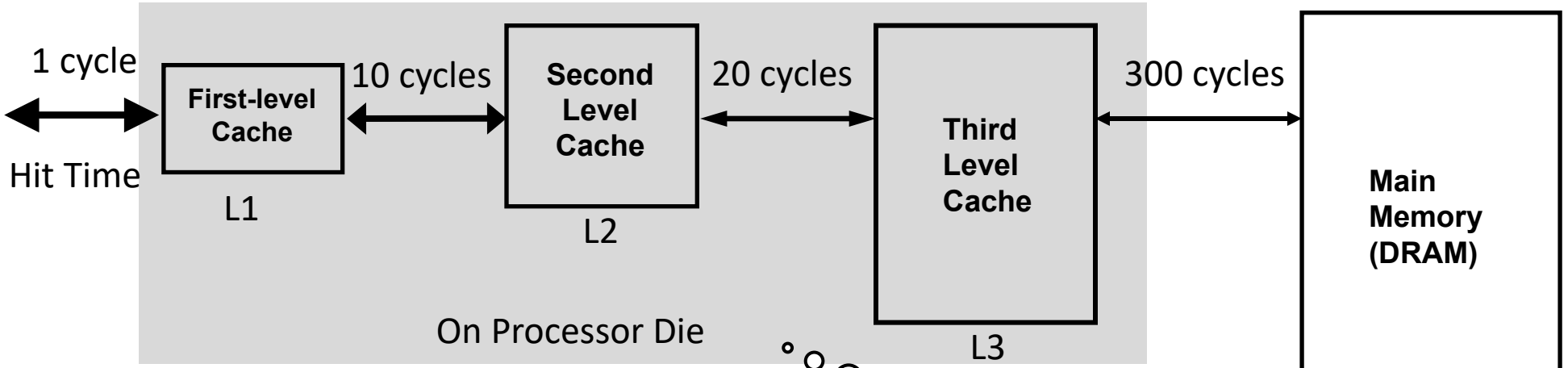
- **AMAT in multi-level cache organization**

  $= T_{hit}(L1) + Miss\_rate(L1) \times$

  $[\ T_{hit}(L2) + Miss\_rate(L2) \times$

  $\{\ T_{hit}(L3) + Miss\_rate(L3) \times T(memory)\ \}\ ]$

Reducing Penalty: Multi-Level Cache



- **AMAT in multi-level cache orc**

  = T_{hit}(L1) + Miss_rate(L1) x

  [T_{hit}(L2) + Miss_rate(L2) x

  { T_{hit}(L3) + Miss_rate(L3) x T(memory) }]

Are more levels always good?

- **Example:**
  - Miss rate of L1, L2, L3 = 10%, 5%, 1%, respectively
  - AMAT = 1 + 0.1 x [10 + 0.05 x { 20 + 0.01 x 300 }] = 2.115 cycles

Vs. 31 cycles
14.7x speedup!

Conclusion Time

What is Row Buffer Locality?

Row reuse rate

What are the row operations for the DRAM?

Activation, restore, precharge

Conclusion Time

What are some cache replacement policies?

LRU, LFU, RR, FIFO, ML based, etc.

How is the memory address divided for cache indexing?

Tag, set, word, byte

Conclusion Time

What are the two memory localities?

Temporal, spatial

What is AMAT?

Hit time + miss rate x miss penalty

SAN JOSÉ STATE UNIVERSITY *powering* SILICON VALLEY