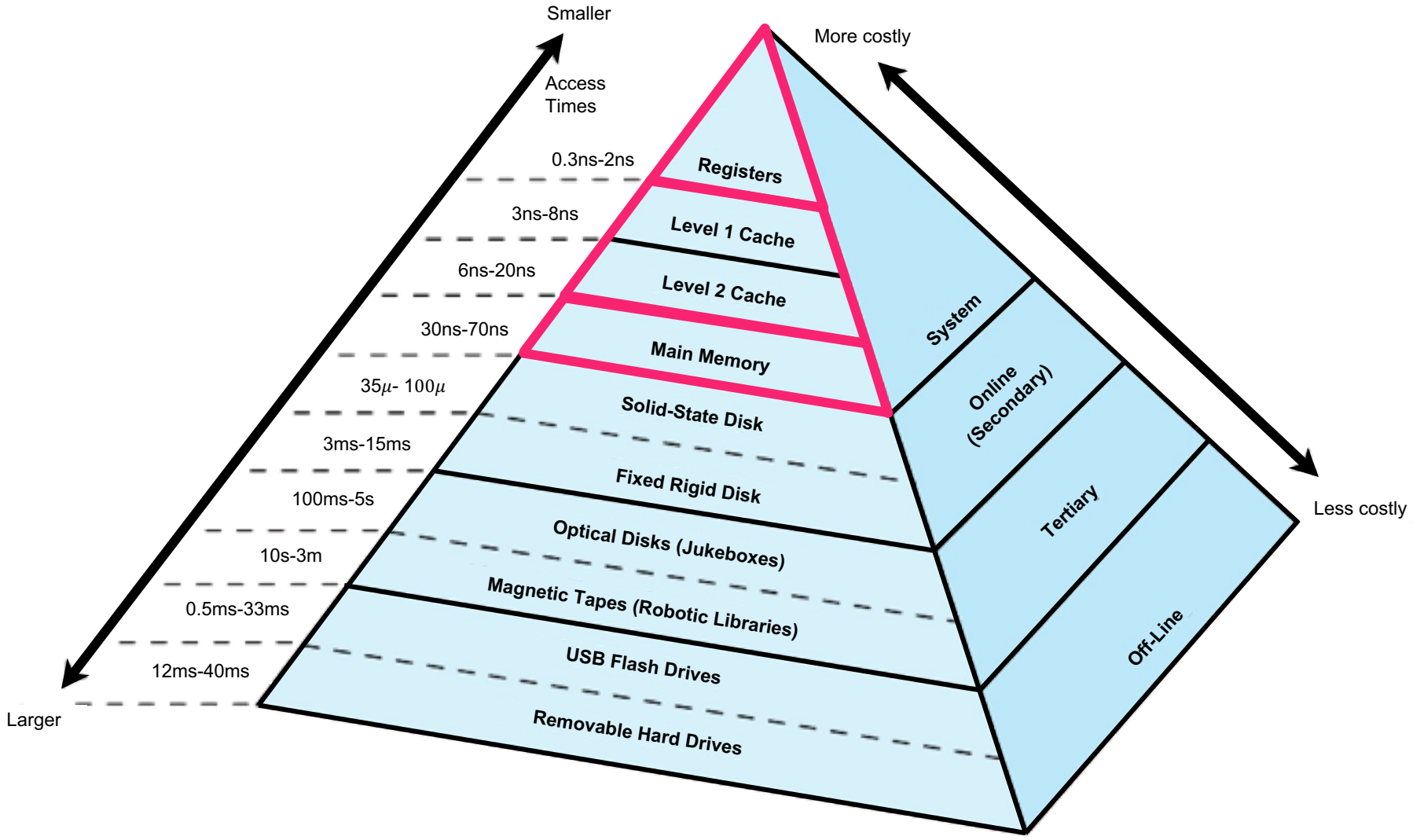
CDA3101

Cache Memory (Part 1)

Memory Hierarchy



Memory Hierarchy

Data movement occurs across memory hierarchy.

Access latencies differ widely.

Register < L1 Cache < L2 Cache < Main Memory < Disk < Archival

Memory performance can be maximized.

Maximize data transfers in fast memory (e.g., Registers ↔ Cache)

Minimize data transfers in slow memory (e.g., Registers ↔ Main Memory)

Place data most likely to be accessed (needed in a register) in fast memory.

Locality of Reference to Improve Performance

Locality can be exploited to improve performance.

Spatial Locality	Temporal Locality
Instructions tend to be accessed sequentially - they are near each other in memory.	Reuse of data or instructions previously accessed
Arrays are together in memory.	

Cache Memory

L1 or **primary cache** is on the same chip as the CPU.

L2 or **secondary cache** can be on the same chip or separate.

Larger than L1

Longer access time than L1

Options:

L1 on CPU, L2 on separate chip

Multicore: Each core has an L1 cache and all cores share L2.

L1 and L2 on CPU, L3 on separate chip

Cache Memory

Unified or integrated caches contain instructions and data.

The alternative is a split cache system.

Instruction (I-cache) and data (D-cache) are separate.

“Harvard cache”

Often see split L1 and unified L2

The separation of data from instructions provides better locality, at the cost of greater complexity.

Cache Memory

1. Blocks containing a desired item is loaded into cache.

Blocks may contain hundreds of bytes.

Blocks take advantage of spatial locality.

3. L1 contains a subset of information in L2.

2. Blocks are copied from main memory into L2 and L1 cache.

4. Cache line holds a block read from main memory.

Blocks and lines are the same size.

Blocks are in main memory, lines in cache.

Cache Writing

Write Through	Write Back
Memory is updated in parallel with each for every write.	Updates only the cache copy of the item
Somewhat defeats the purpose of having a cache	Main memory is updated when the time is removed from the cache.
	Must record whether cache line was written (dirty)
	May cause issues with concurrent access (e.g., shared memory multiprocessors)

Cache Memory

Cache Hit		Cache Miss	
When data is found at a given memory level (e.g. L1, L2)		When data is not found	
Hit Rate	Percent of references found at a given memory level	Miss Rate	100 – Hit Rate
Hit Ratio	Number of hits / total number of references	Miss Ratio	1 – Hit Ratio
Hit Time	Time needed to access data at a given memory level	Miss Penalty	Time required to process a miss

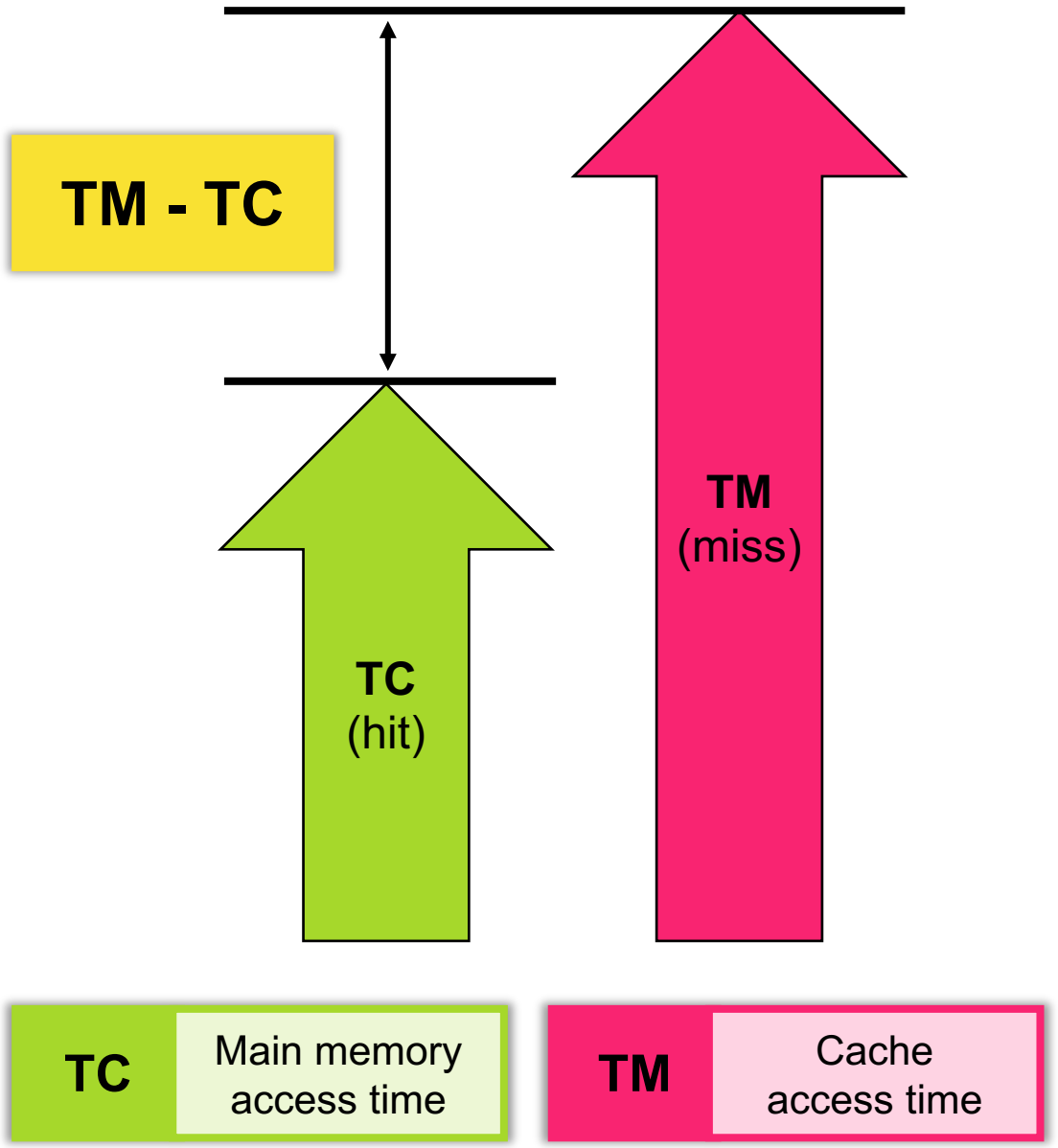
Lookaside Cache

Accesses to cache and to main memory occur in parallel

Main memory access is cancelled if hit in cache occurs.

Tends to lower average memory access time

Increases CPU to memory traffic



Lookaside Cache

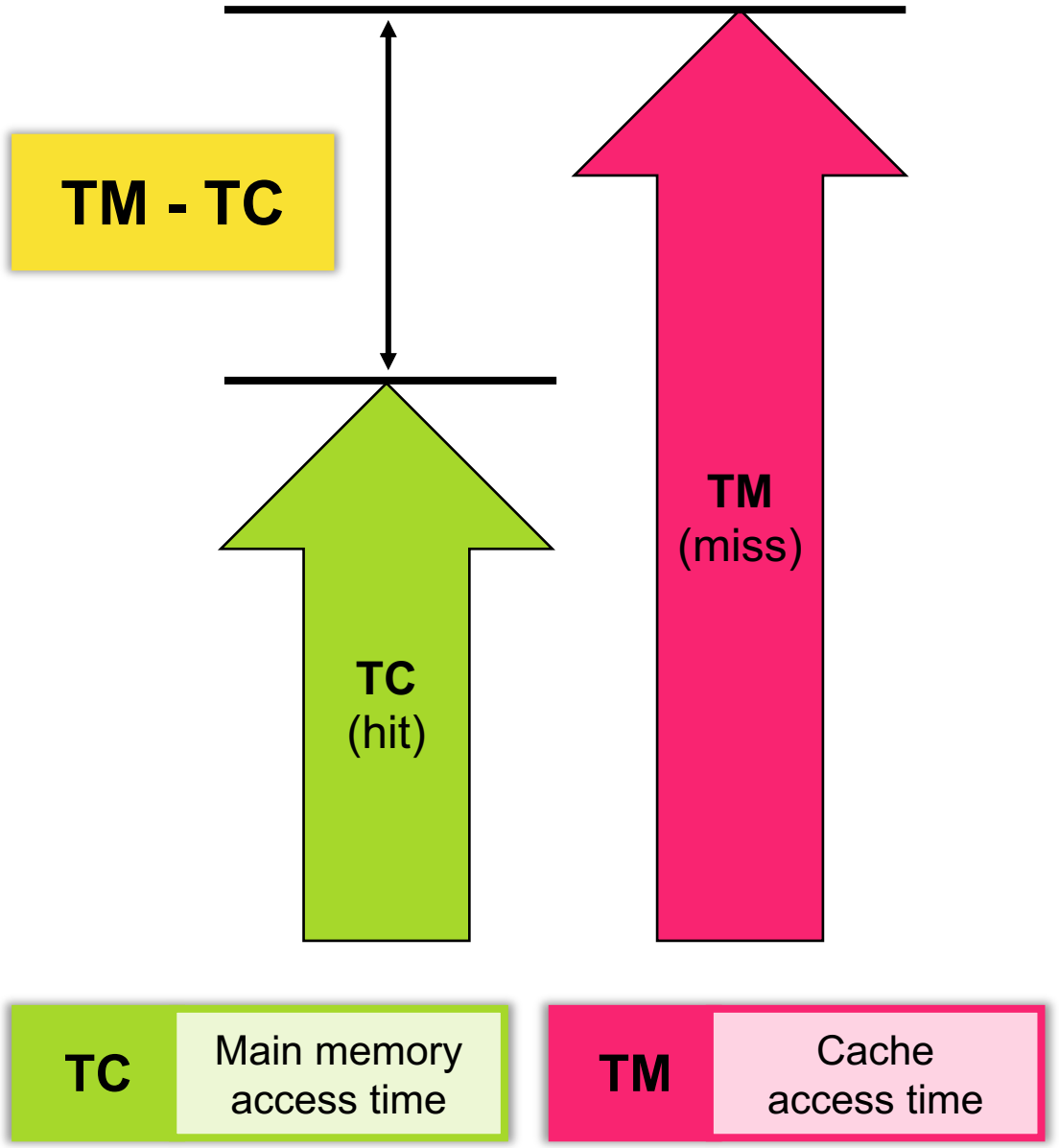
Example

Consider a system with a main memory access time of 200 ns supported by a cache having a 10 ns access time and a hit rate of 99%.

The **average memory access time** (AMAT) is:

$$0.99\times(10ns) + 0.01\times(200ns)$$
$$= 9.9ns + 2ns$$
$$= 11.9ns$$

$$AMAT = Hit\ Ratio\times TC + (1 - Hit\ Ratio)\times TM$$



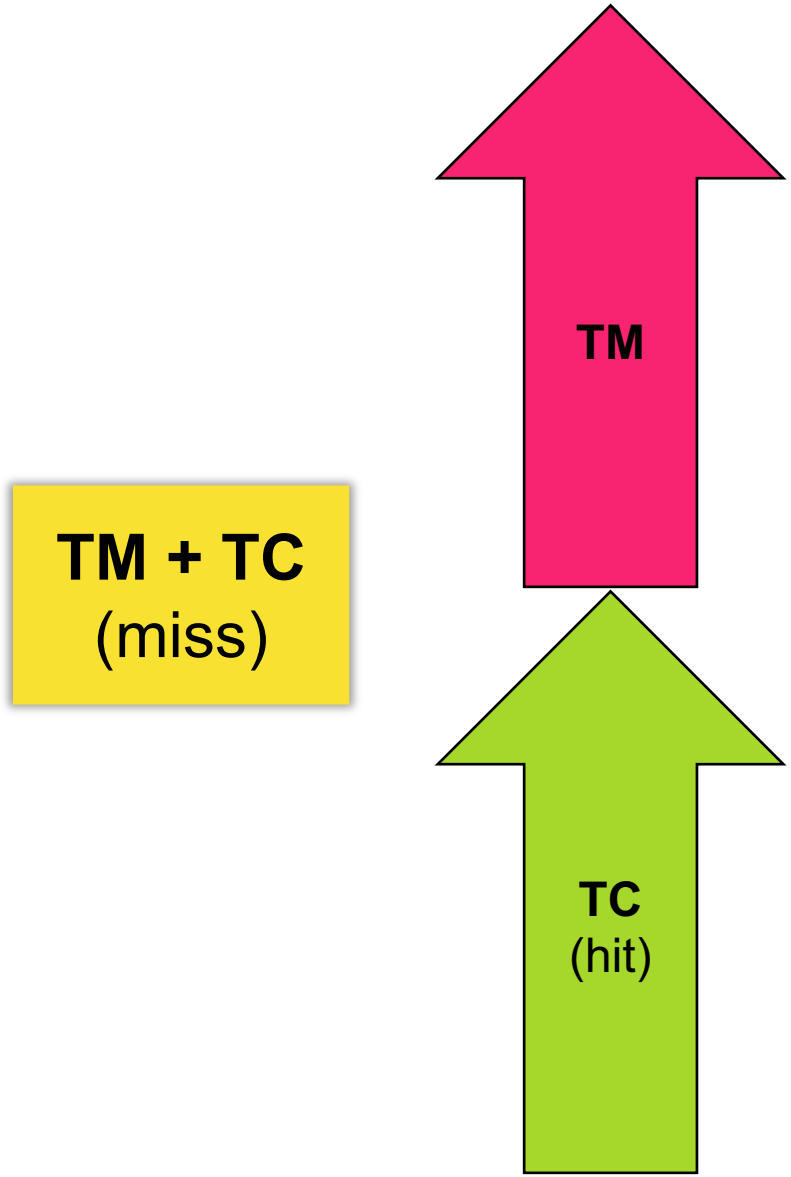
Look Through Cache

First level cache is checked first.

Next level is only checked if miss occurs.

Avoids unneeded CPU-to-memory traffic

Tends to increase average memory access time



Look Through Cache

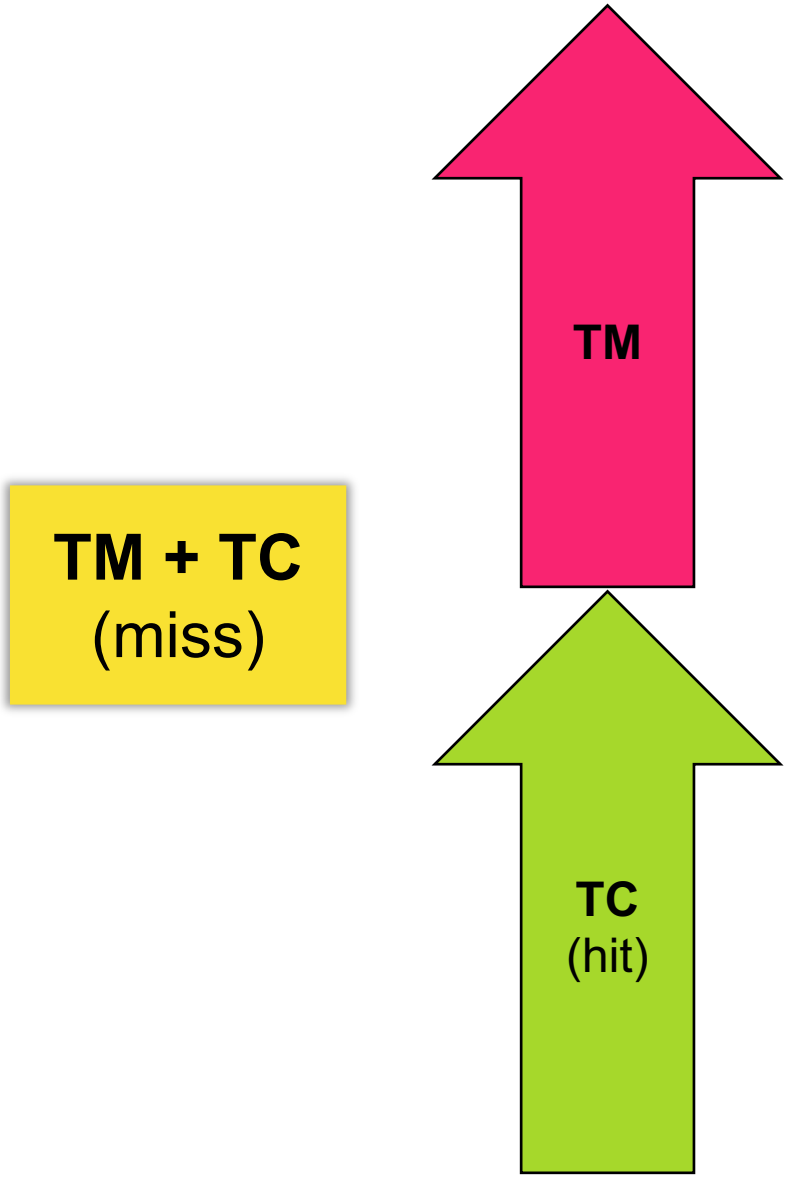
Example

Consider a system with a main memory access time of 200 ns supported by a cache having a 10 ns access time and a hit rate of 99%.

The **average memory access time** (AMAT) is:

$$0.99\times(10ns) + 0.01\times(10ns + 200ns)$$
$$= 9.9ns + 2.1ns = 12ns$$
$$= 10 + 0.01(200) = 10 + 2 = 12ns$$

$$AMAT =$$
$$TC + (1 - Hit\ Ratio)\times TM$$



Multi-Level Cache

Example

Consider a system with a main memory access time of 200 ns supported by a L1 cache having a 10 ns access time and a hit rate of 90%, and an L2 cache having a 20 ns access time and a hit rate of 95%. **What is the AMAT if for a look aside cache?**

AMAT

$$0.9\times(10ns) + 0.1\times(0.95\times 20ns + 0.05\times 200ns)$$
$$= 9 + 0.1\times(19 + 10)$$
$$= 9 + 1.9 + 1$$
$$= 11.9ns$$

Multi-Level Cache

Example

Consider a system with a main memory access time of 200 ns supported by a L1 cache having a 10 ns access time and a hit rate of 90%, and an L2 cache having a 20 ns access time and a hit rate of 95%. **What is the AMAT if for a look through cache?**

AMAT

$$10ns + 0.1\times(20ns + 0.05\times200ns)$$
$$= 10 + 0.1\times(20 + 10)$$
$$= 10 + 3$$
$$= 13ns$$

Sources of Misses

Compulsory Misses

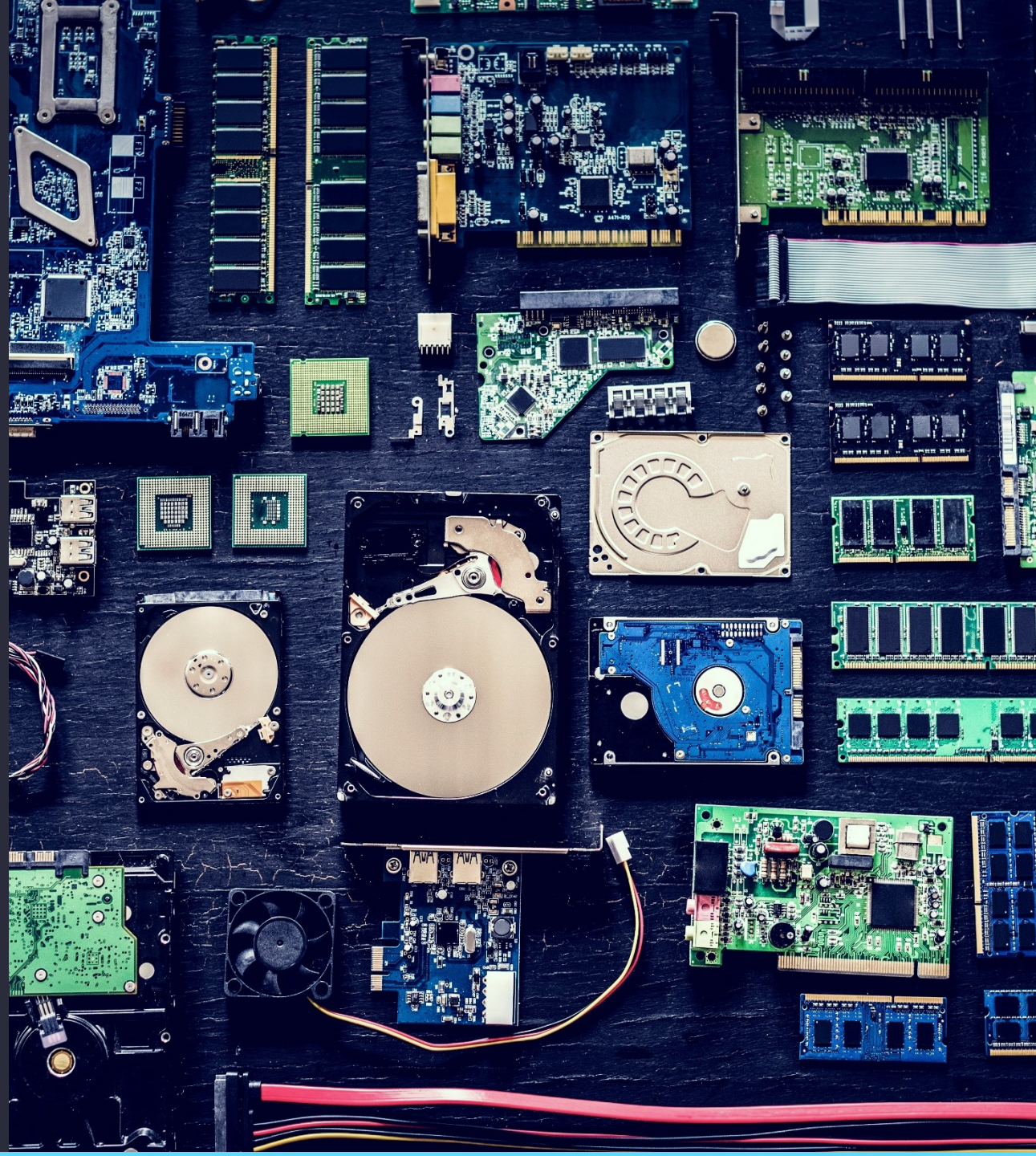
Unavoidable due to initially empty cache

Capacity

Due to limited size of cache

Conflict

Due to different blocks mapping to the same cache line



Cache

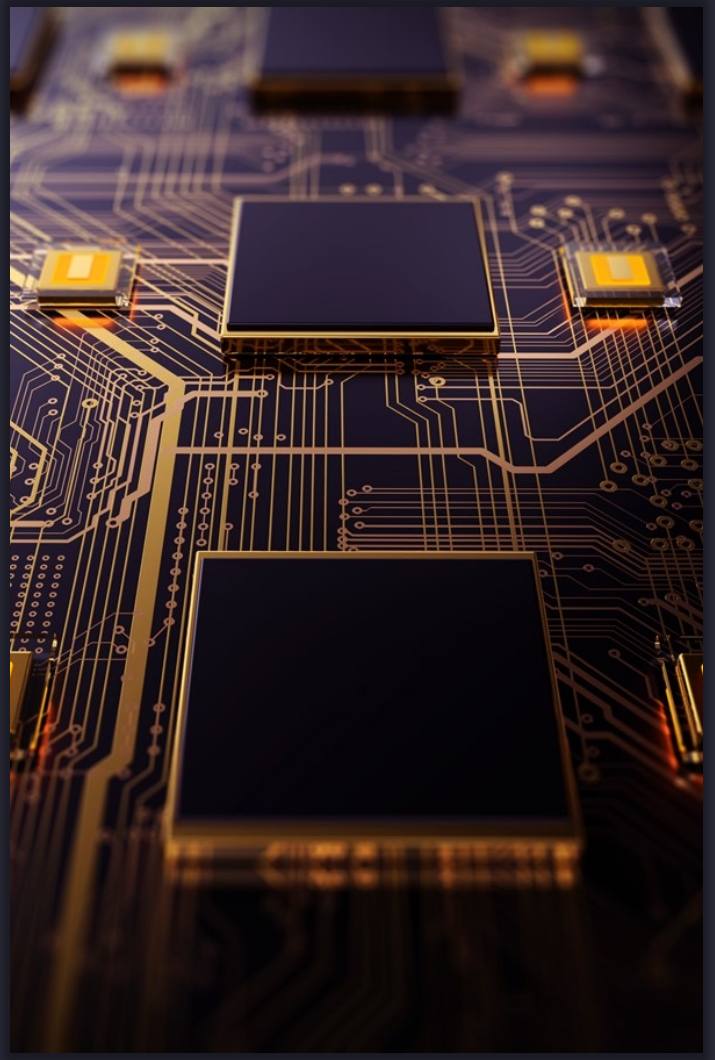
Is it the item I want in the cache?

How do I find it?

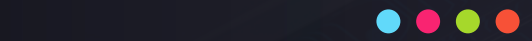
Three cache organization methods:

- Fully associative
- Set associative
- Direct mapped

Wrap Up



Thank you for watching.



UF | UNIVERSITY of FLORIDA