CASPER

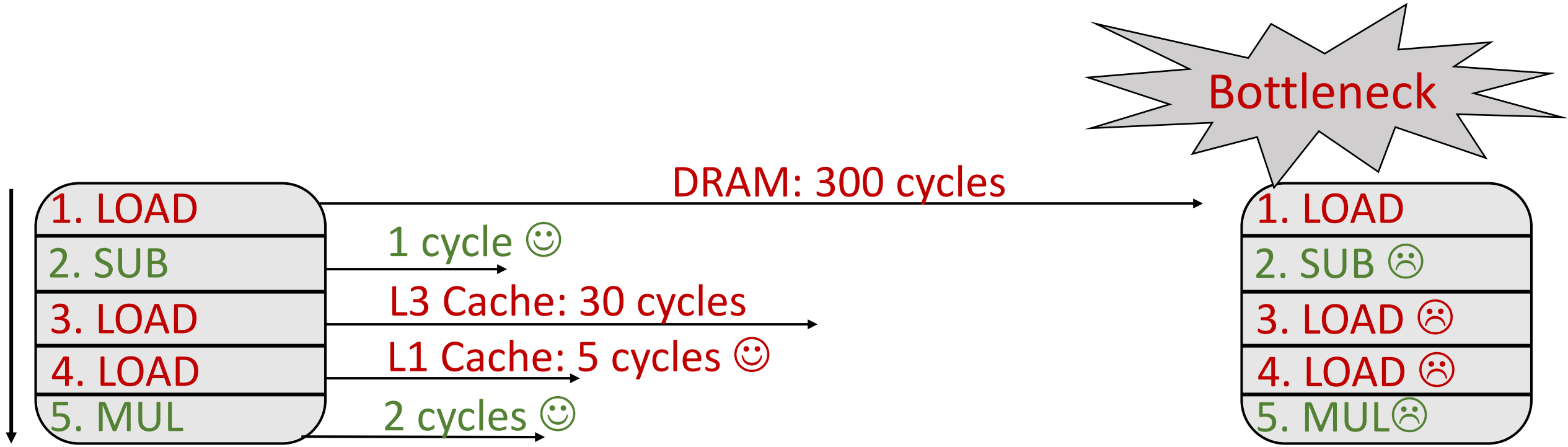
CS305: Computer Architecture

Caches-IV

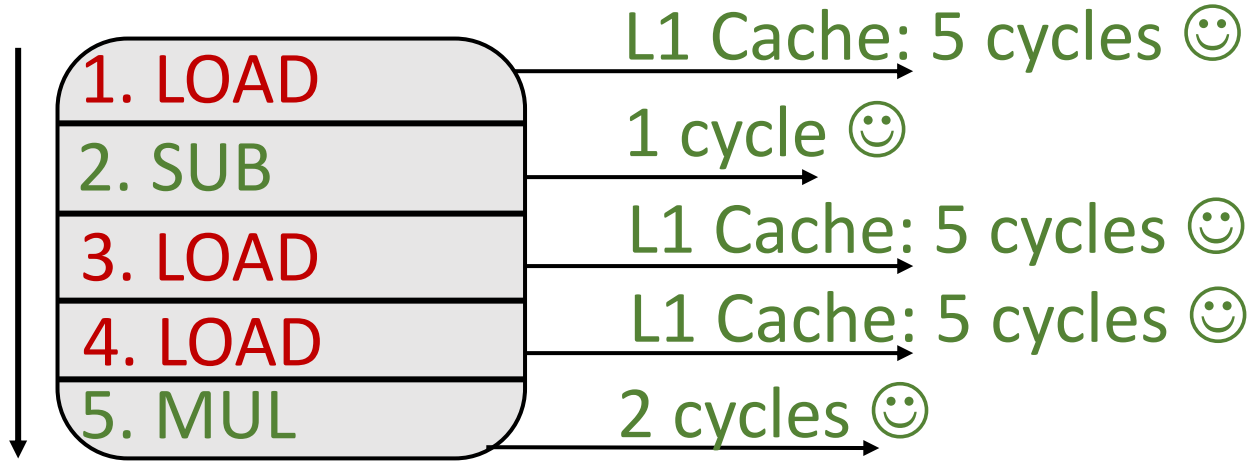
https://www.cse.iitb.ac.in/~biswa/courses/CS305/main.html

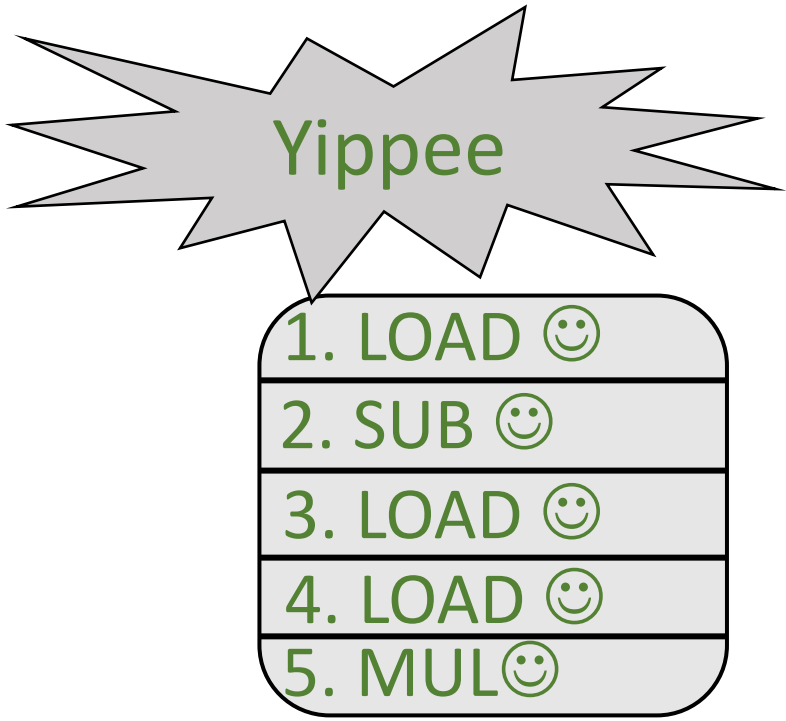
https://www.cse.iitb.ac.in/~biswa/

The Implications of L1-D Hit rate of 100%



A Dreamy World

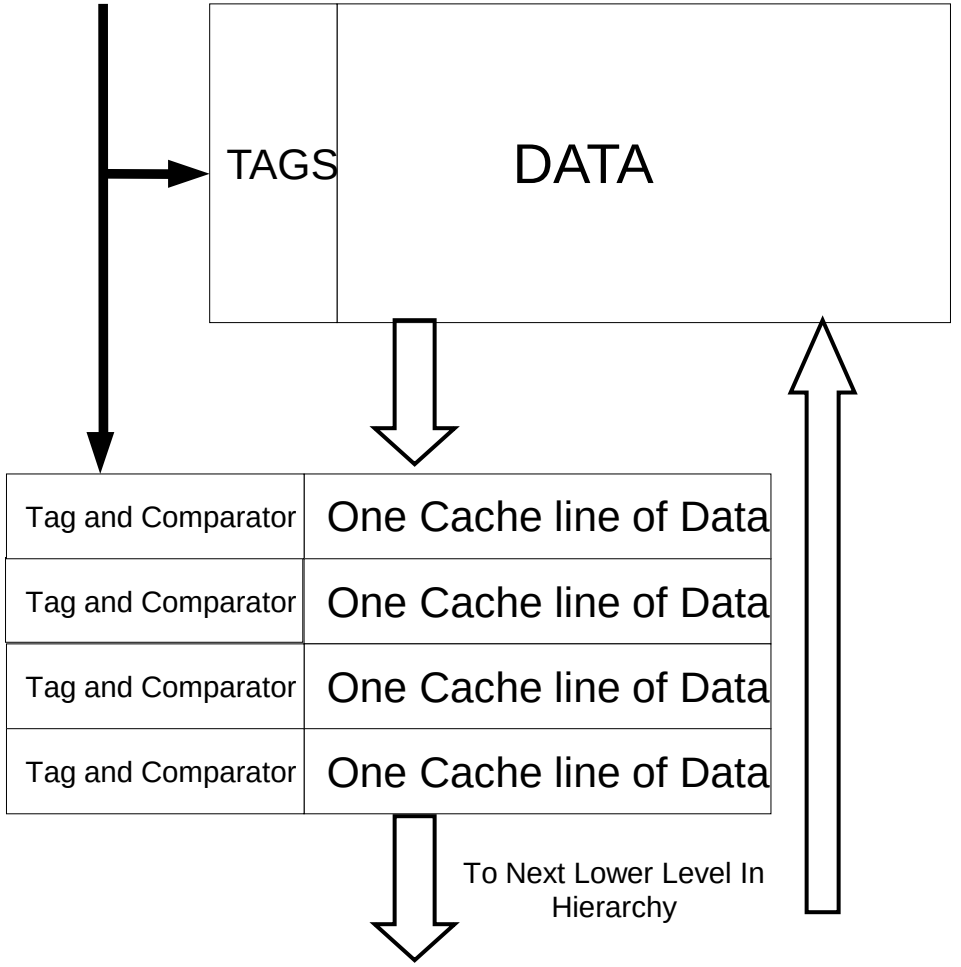




L1 Data Cache

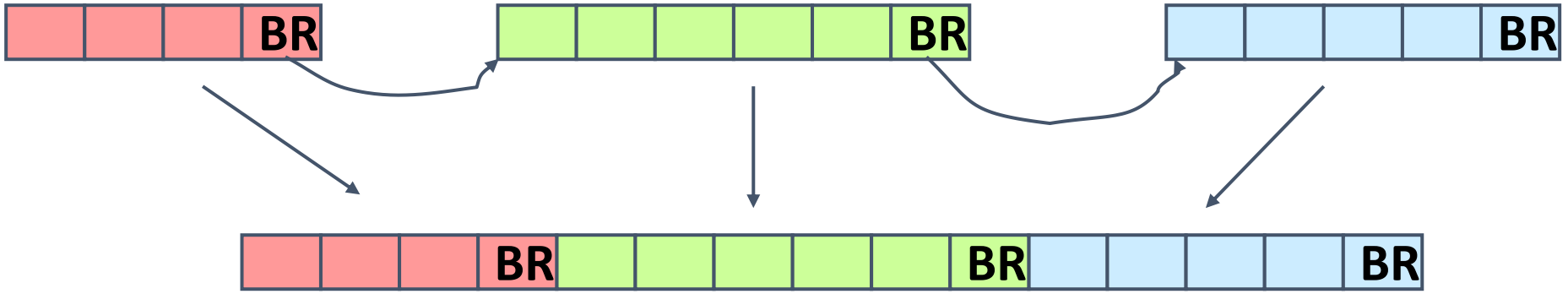
Pipelined : For high bandwidth
Simple (low-associative):
For fast hit time

Victim Cache:
How to combine fast hit time
yet still avoid conflict misses?
Keep the discarded data



L1 Instruction Cache

Key Idea: Special instruction cache that packs multiple non-contiguous basic blocks into one contiguous trace cache line

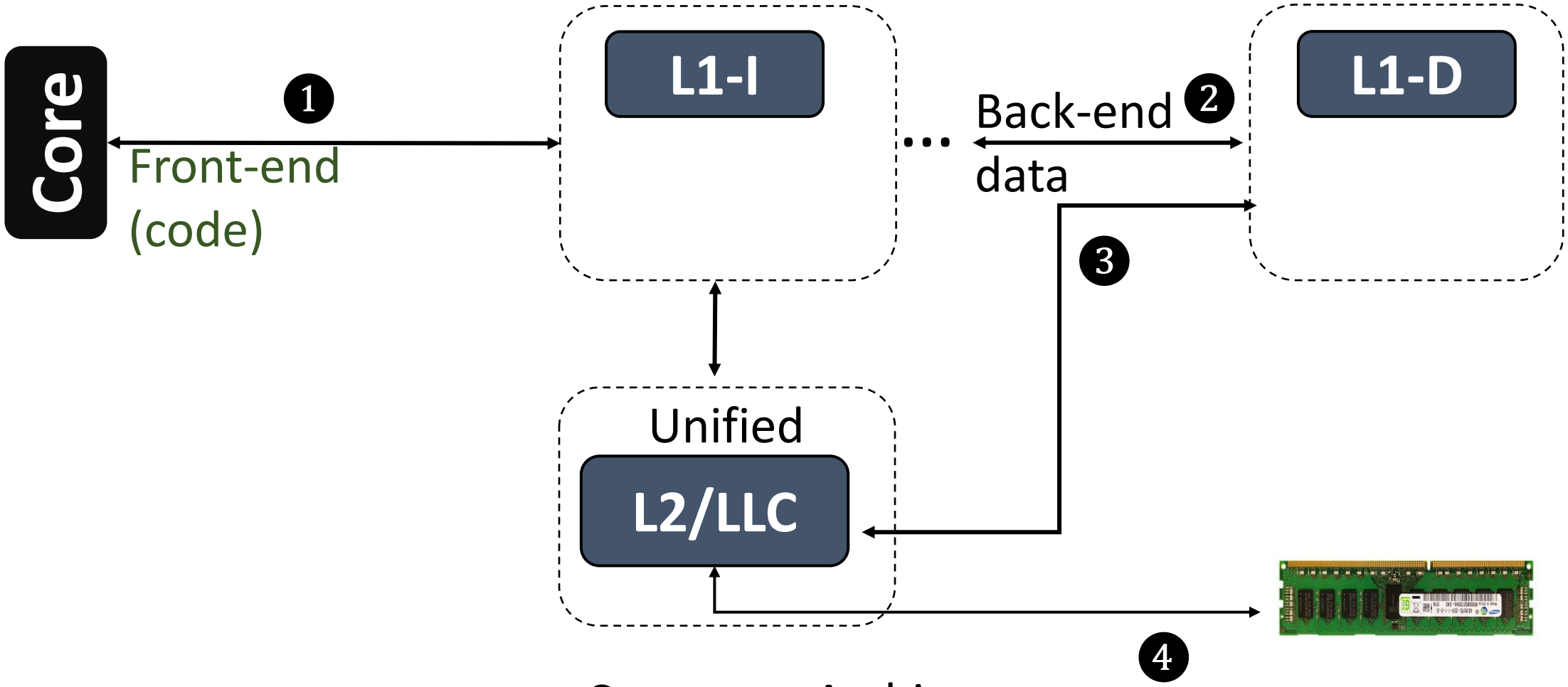


Single fetch brings in multiple basic blocks

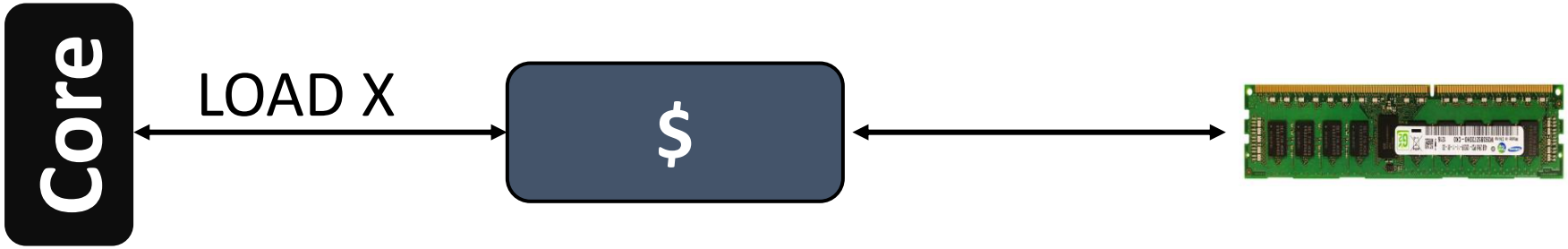
Trace cache indexed by start address *and* next *n* branch predictions

BTW, BTB is nothing but a cache of target addresses.

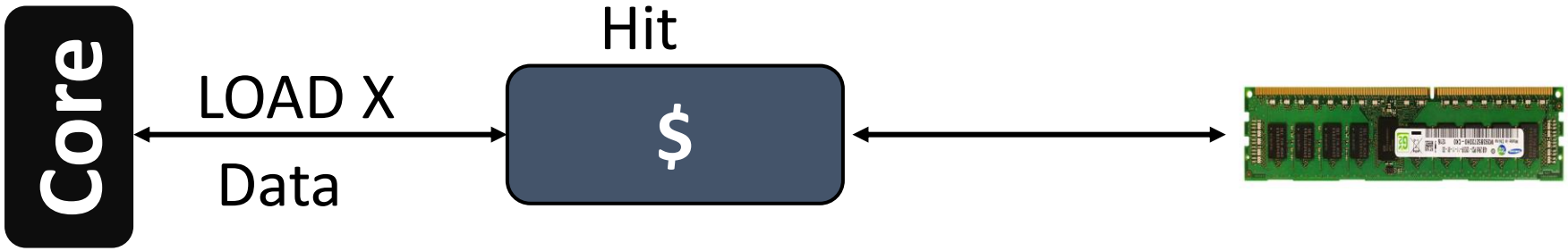
Core, cache, DRAM interaction



Core, cache, DRAM interaction

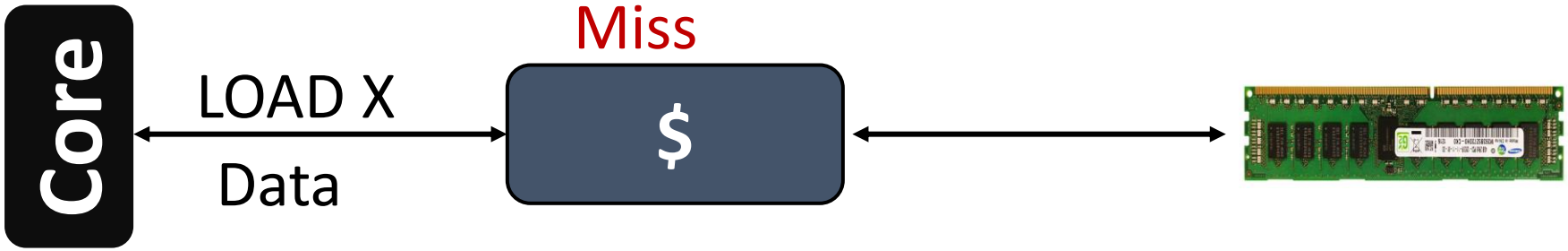


Core, cache, DRAM interaction



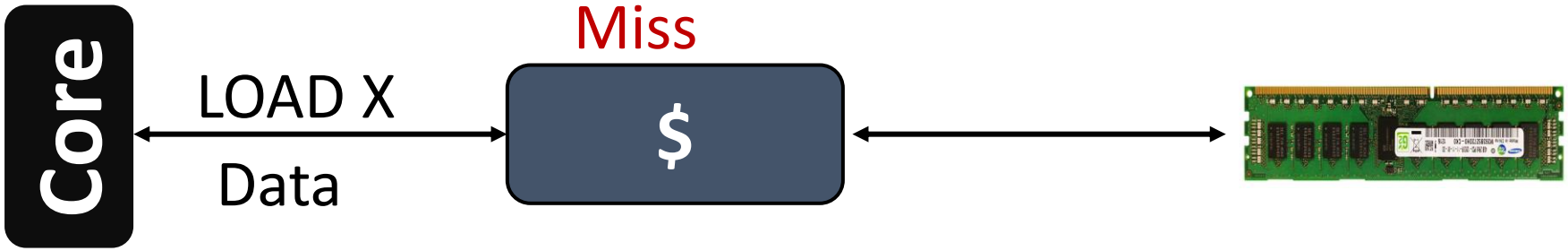
Few cycles

On a miss: Critical Word first



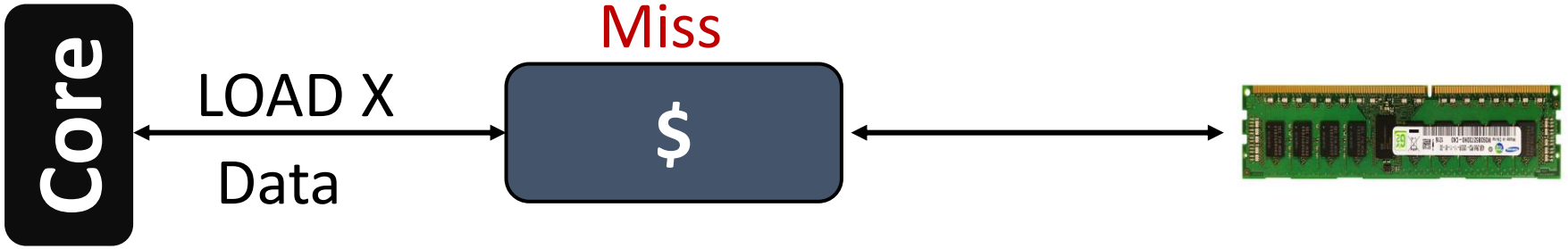
On a miss, respond with the word/byte requested to the core so that core can continue while fetching the rest of the block

On a miss: Early Restart



On a miss, fetch the words/bytes in normal order, but as soon as the requested word/byte of the block arrives, send it to the core.

Core, cache, DRAM interaction

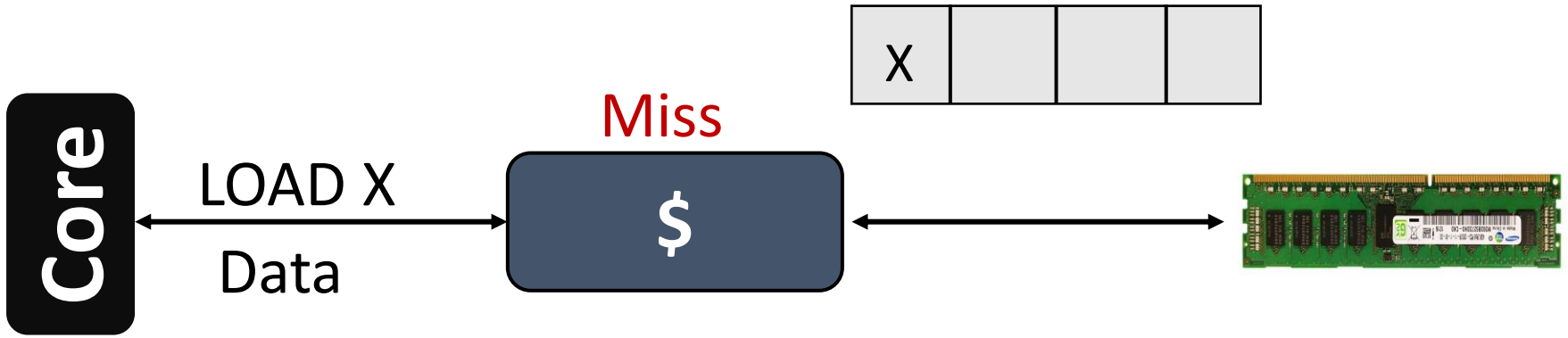


100s of cycles

I am an out-of-order core

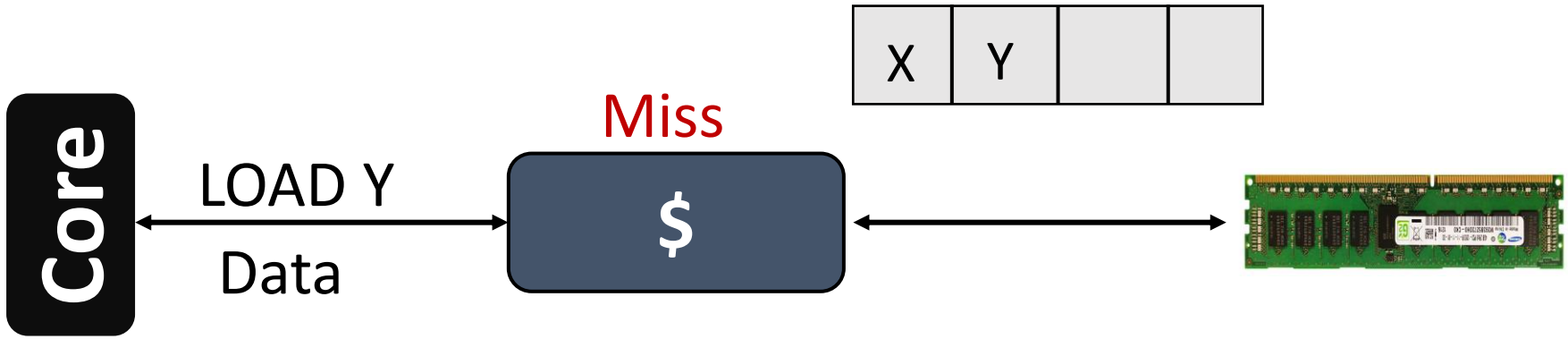
One cache miss and can't handle anymore misses

MSHRS (Miss-status holding registers)

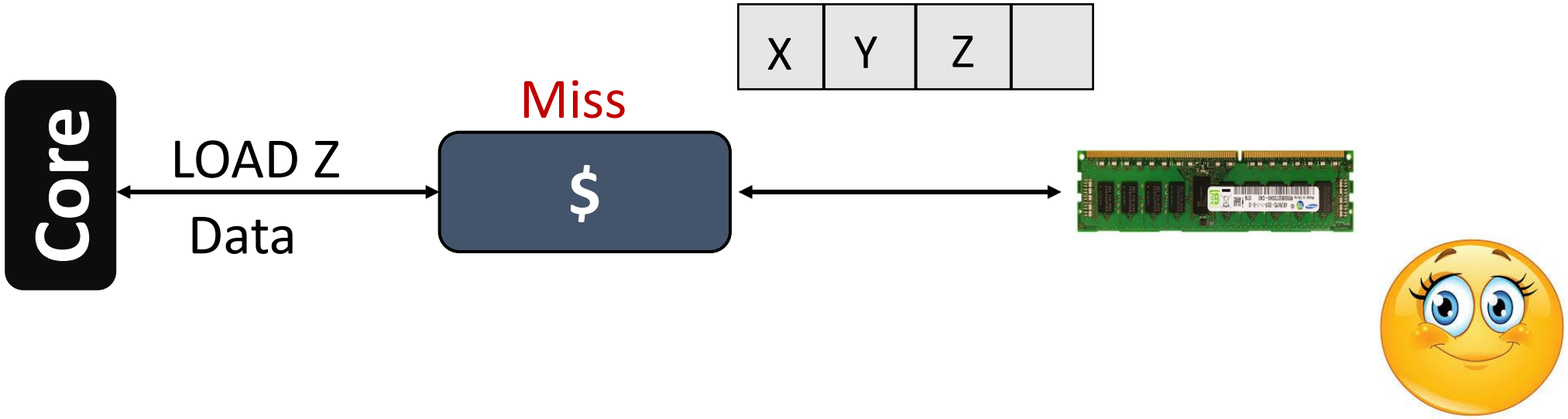


Non-blocking cache

MSHRS (Miss-status holding registers)

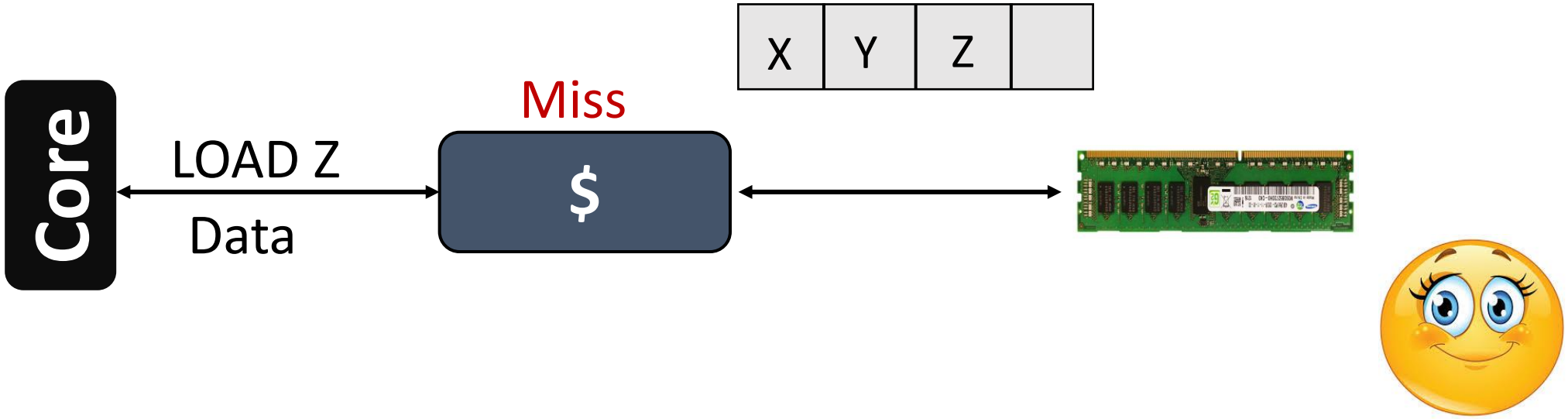


MSHRS (Miss-status holding registers)



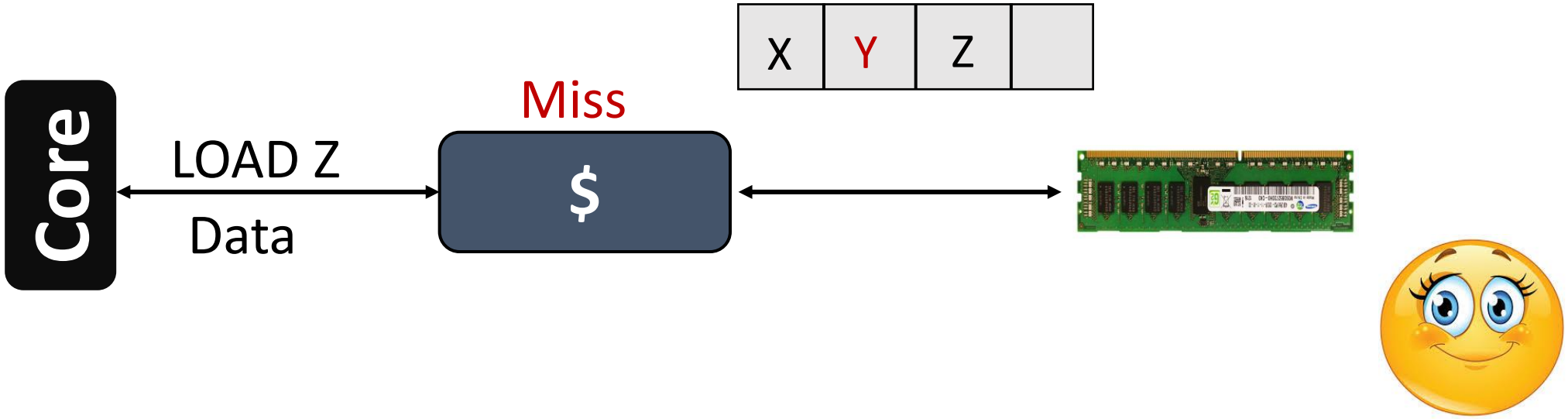
K-entry MSHR allows K outstanding misses: provides memory-level parallelism

MSHRS (Miss-status holding registers)



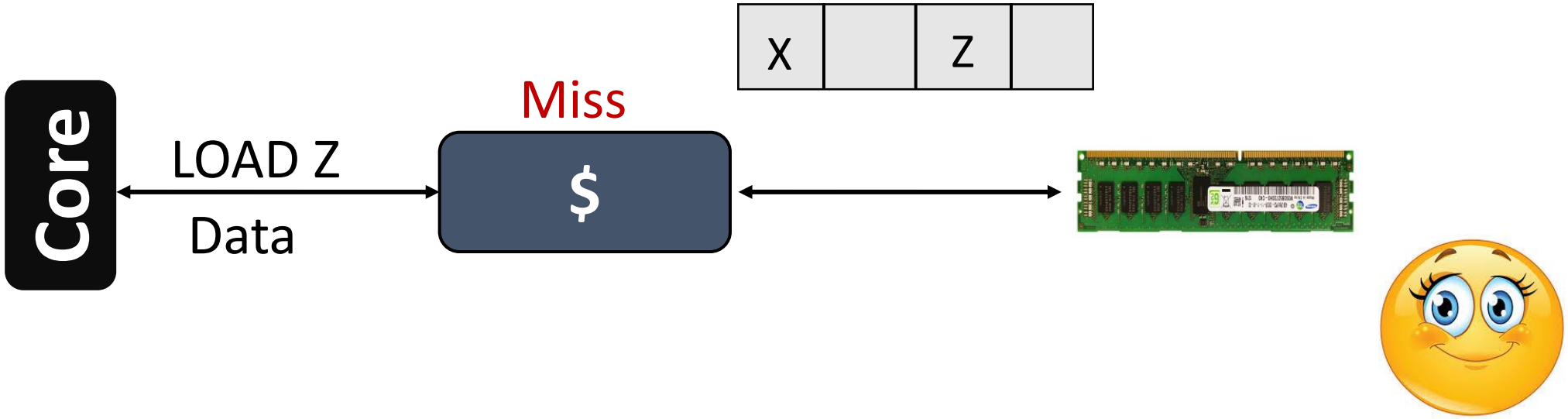
DRAM response time is not constant: can take from 60 cycles to 1000s of cycles (on a multi-core system).

MSHRS (Miss-status holding registers)



DRAM response time is not constant: can take from 60 cycles to 1000s of cycles (on a multi-core system).

MSHRS (Miss-status holding registers)



DRAM response time is not constant: can take from 60 cycles to 1000s of cycles (on a multi-core system).

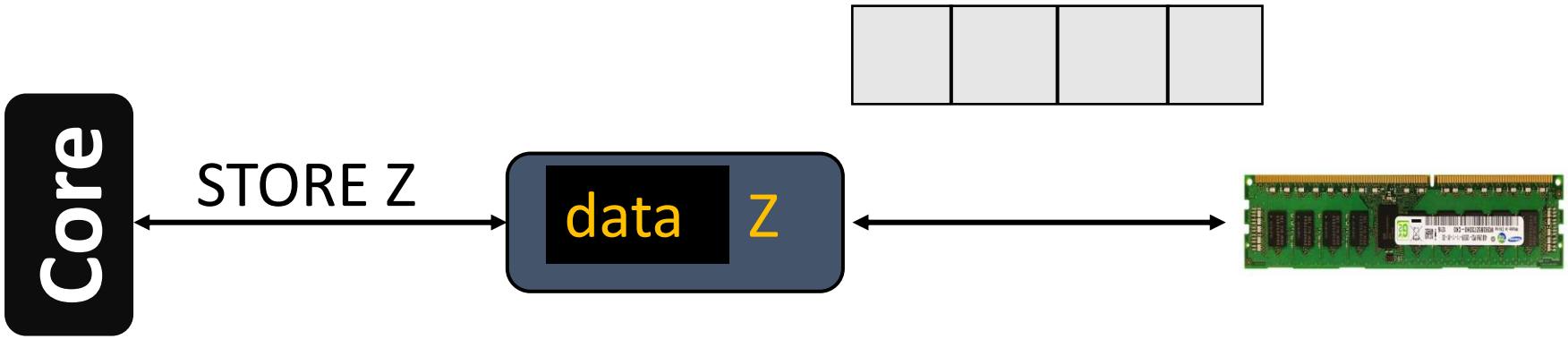
What about writes (stores)

On a hit: Update the cache block. We need an additional bit in tag-store, named *dirty bit along with the valid bit and replacement priority bits*.

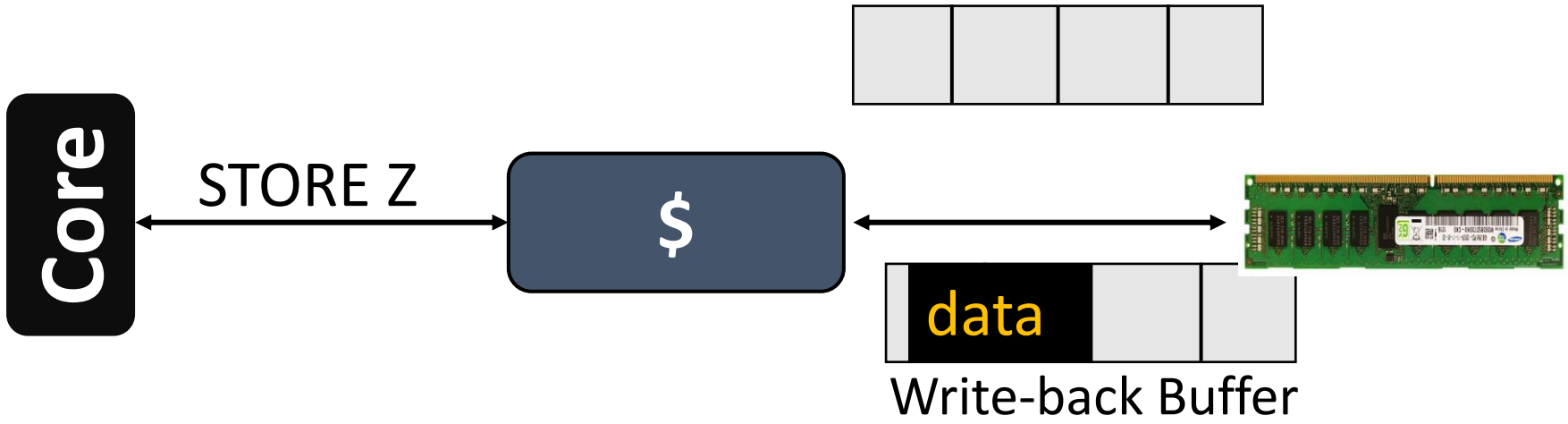
Write-through cache: On a hit, write into a cache block and also into the next-level in the memory hierarchy

Write-back cache: On a hit, write into a cache block only, and during replacement update the next-level

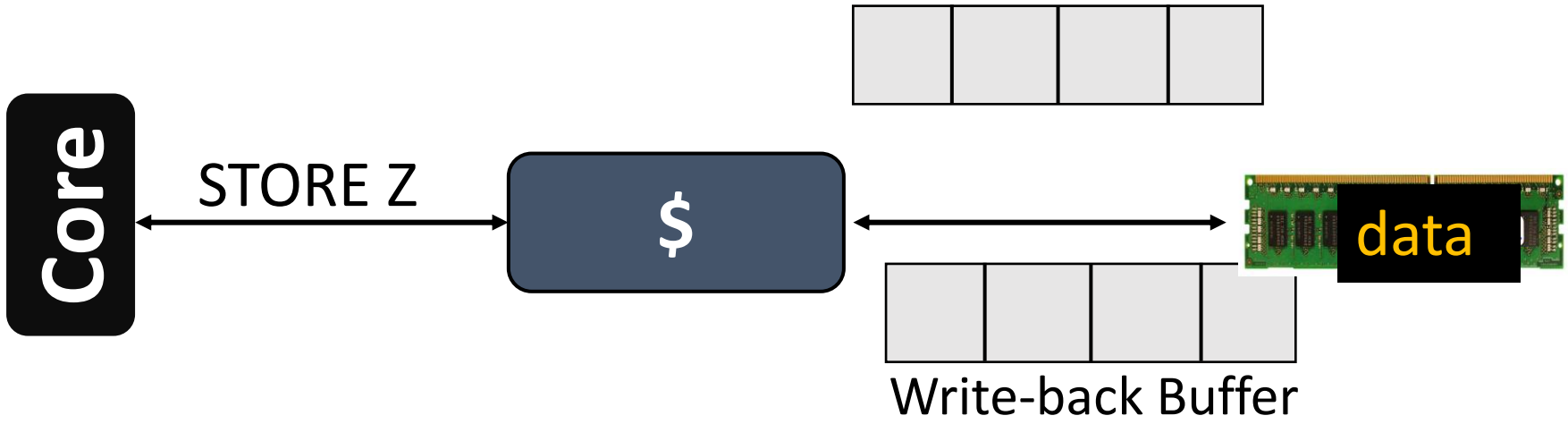
What about writes: Writeback cache



What about writes: On replacement



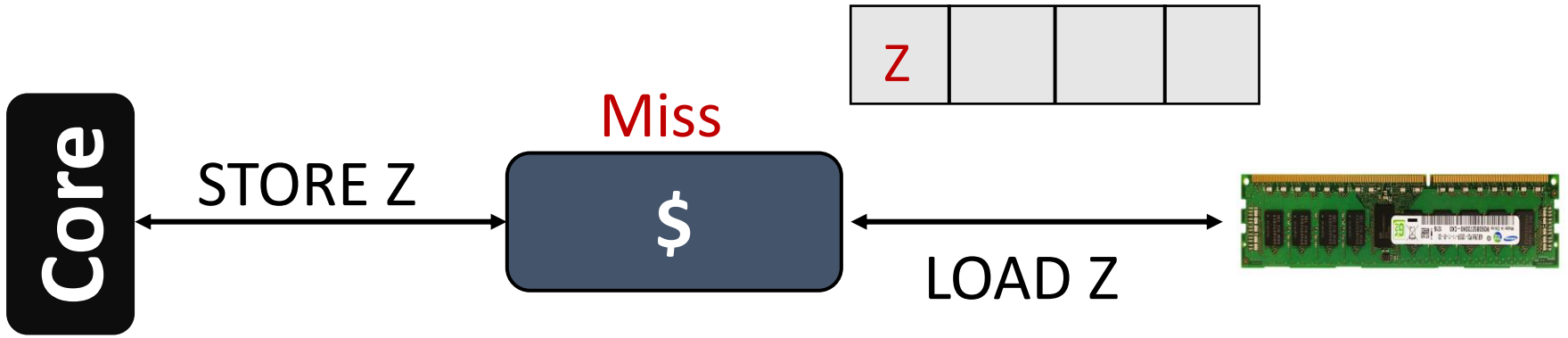
Write-back cache



Write-back cache

In general, STOREs are not critical for performance. But why?

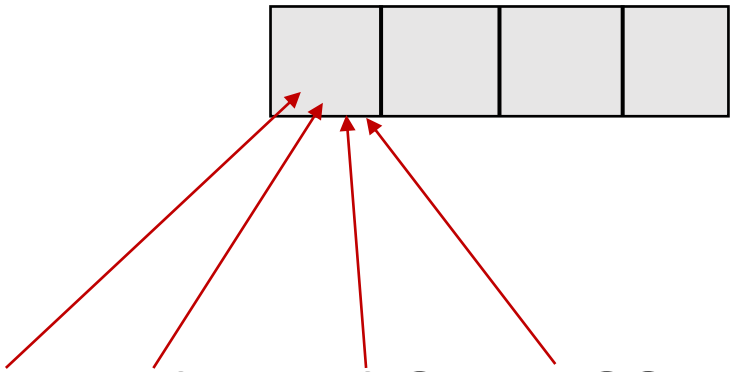
What about a write miss?



STORE gets converted to a LOAD, and data is allocated into the cache (write-allocate policy).

Usually write-allocate is used with the write-back caches.

Write Merging



STORE to address Z, Z+1, Z+16, Z+63 merged as all belong to one cache line of 64 bytes.

The Bigger Picture

CPU time = CPU execution cycles + Memory-stall cycles

Clock cycle time may be different

Memory-stall cycles = Read-stalls + Write-stalls

Read/Write stalls =

#Reads/writes X Read/Write miss-rate X Read/write miss-penalty

Met Dank