CASPER

CS773-2022-Autumn: Computer Architecture for Performance and Security

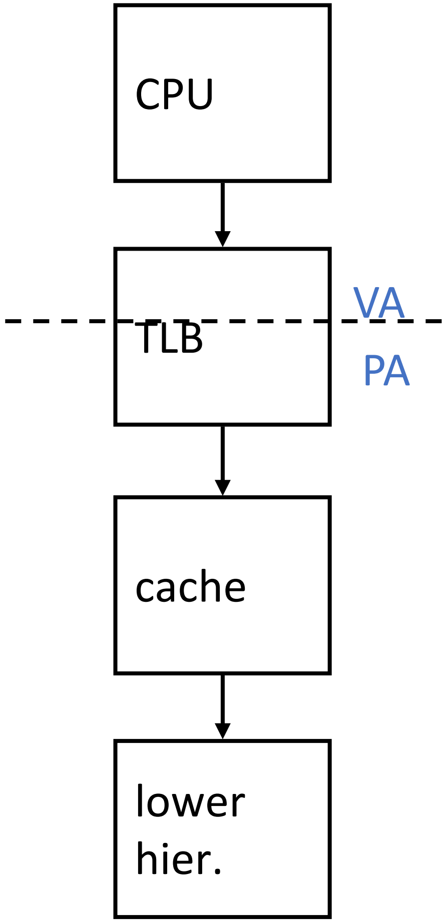
Lecture 10: Virtual Caches and Eviction based cache attacks

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

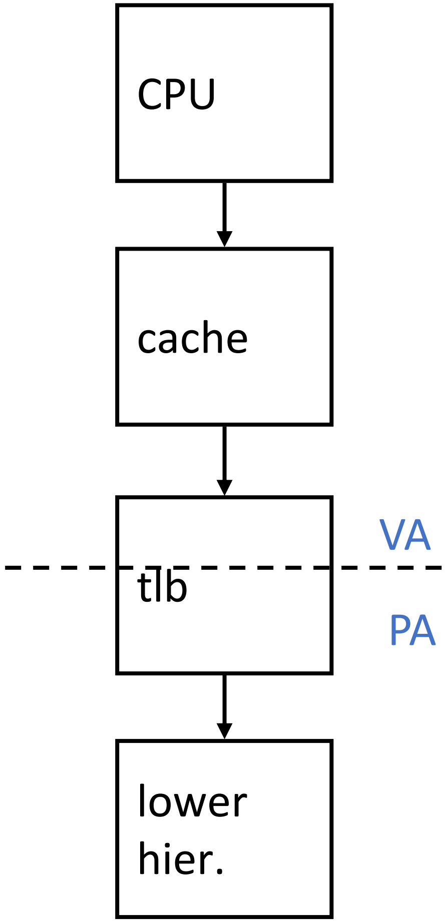
Phones on silence, please

Thank You

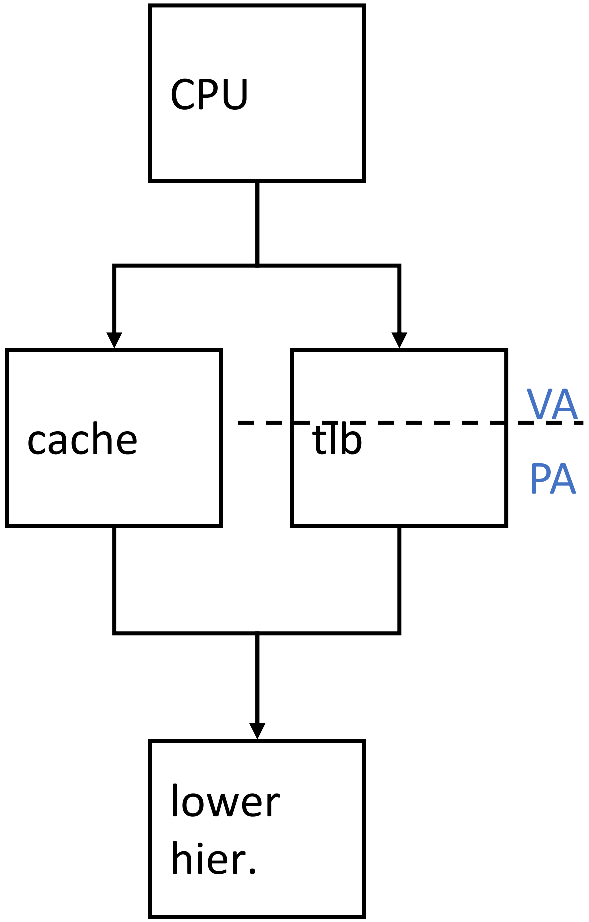
Caches: Virtual or Physical



physical cache

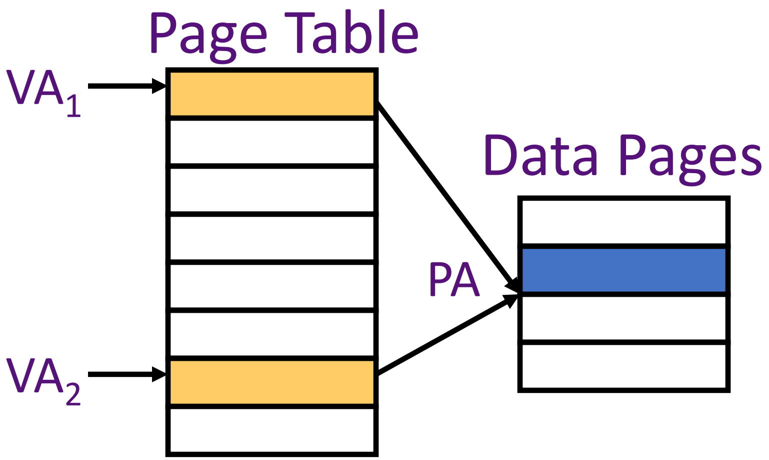


virtual (L1) cache



virtual-physical cache

Synonym Problem



Two virtual pages share one physical page

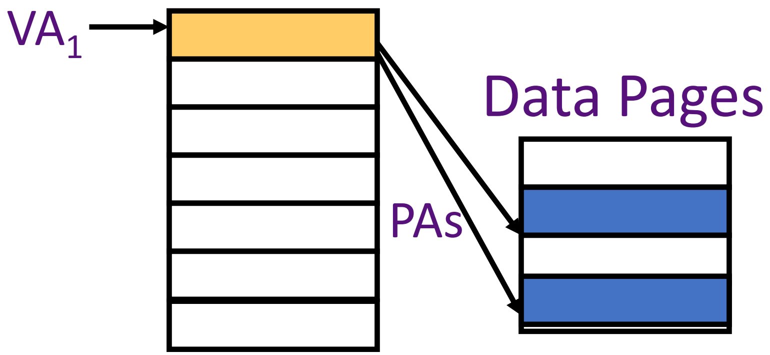
Tag	Data
VA_1	1st Copy of Data at PA
VA_2	2nd Copy of Data at PA

Virtual cache can have two copies of same physical data. Writes to one copy not visible to reads of other!

General Solution: *Prevent aliases coexisting in cache*

Software (i.e., OS) solution for direct-mapped cache

Homonym Problem

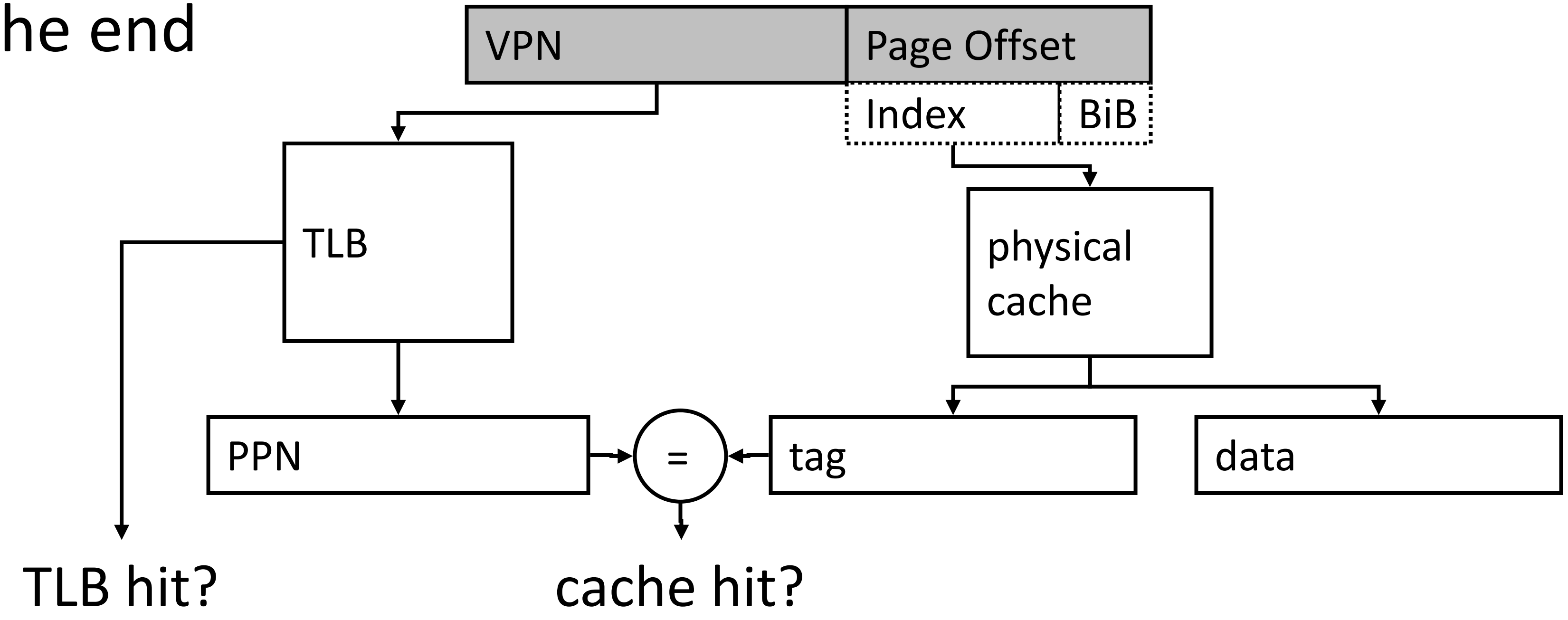


One virtual page maps to two physical pages

Tag may not uniquely identify cache data
Solution: Add ASID with tag
Or
Physical tags
Or
Flush on context switch

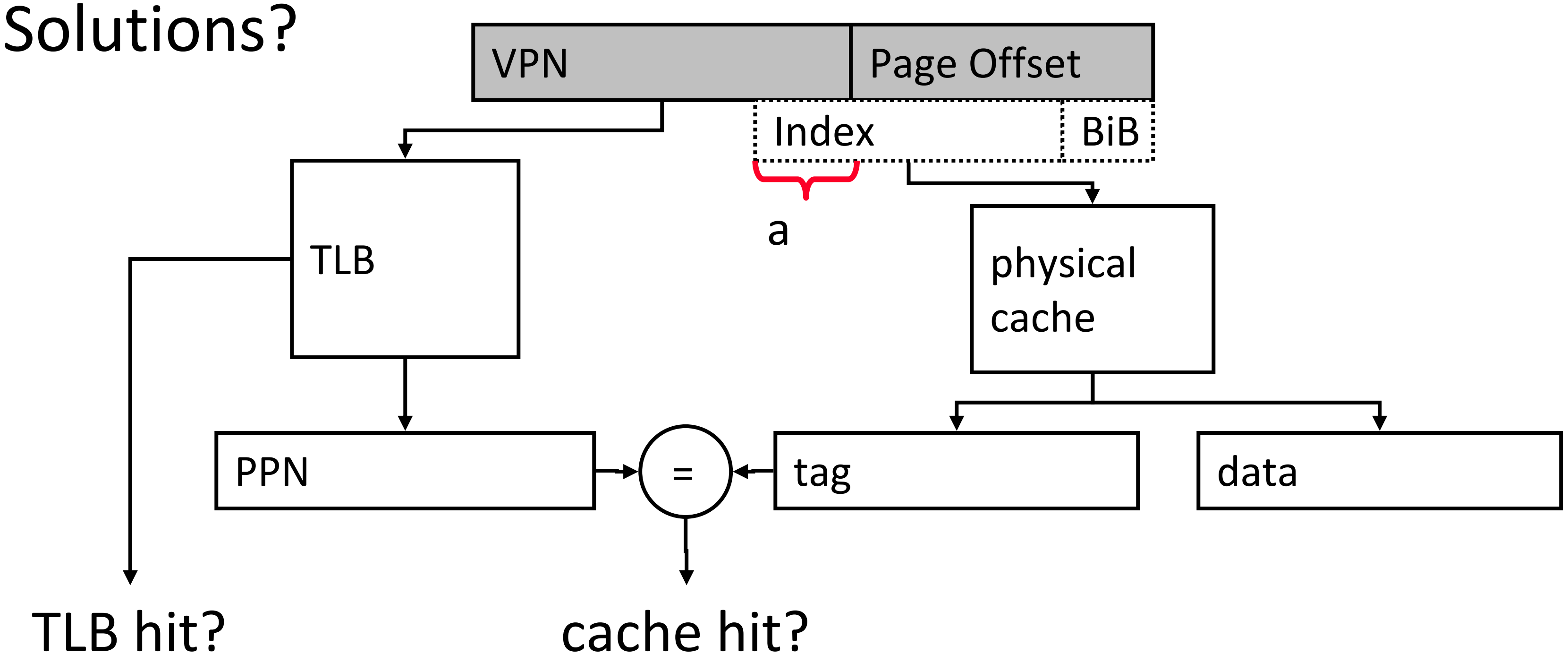
VIPT Caches

- If C≤(page_size × associativity), the cache index bits come only from page offset (same in VA and PA)
- If both cache and TLB are on chip: index both arrays concurrently using VA bits, check cache tag (physical) against TLB output at the end



What if? Think about PIPT, VIPT, PIVT, and VIVT

- If C>(page_size × associativity), the cache index bits include VPN ⇒ Synonyms can cause problems
  - The same physical address can exist in two locations
- Solutions?



Summary

- VIVT (Virtual cache): Fastest, Synonyms, Homonyms,

- VIPT: Good enough, No Homonyms, mostly in L1 caches

- PIVT: ??

- PIPT (Physical cache): You know it

One more layer of virtualization. What?

World of virtual machines (VMs):

A complete compute environment with its own isolated processing capabilities, memory, and communication channels.

Hypervisor: system software that manages VMs.

Example: *Multiple operating system environments on a single system. Windows OS (guest) running on a Linux system (host)*

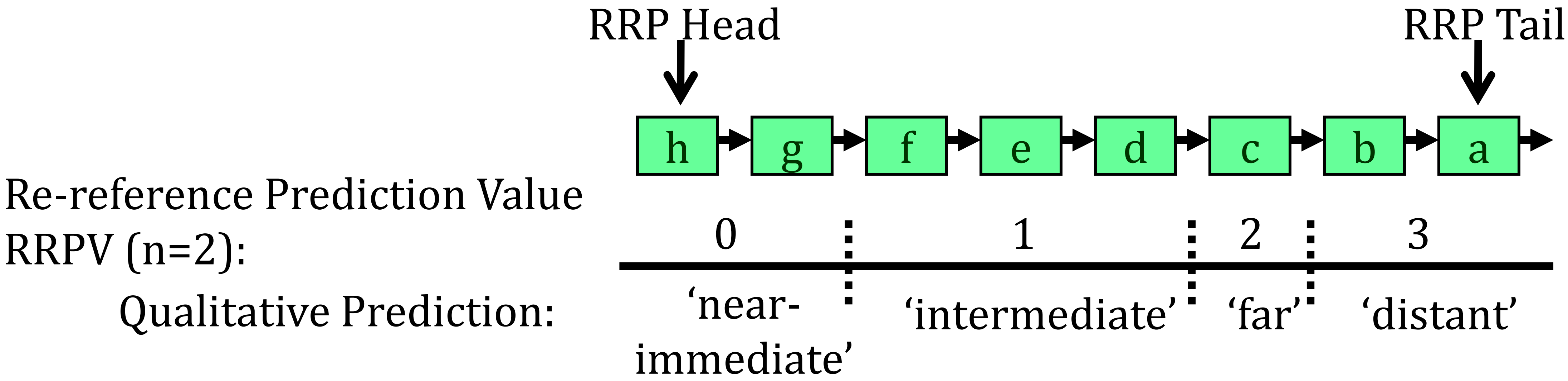
Devil is in the details

Application running on a guest OS generates gVA that should be translated to hPA.

How many page tables to traverse?

Time for contention/eviction-based attacks

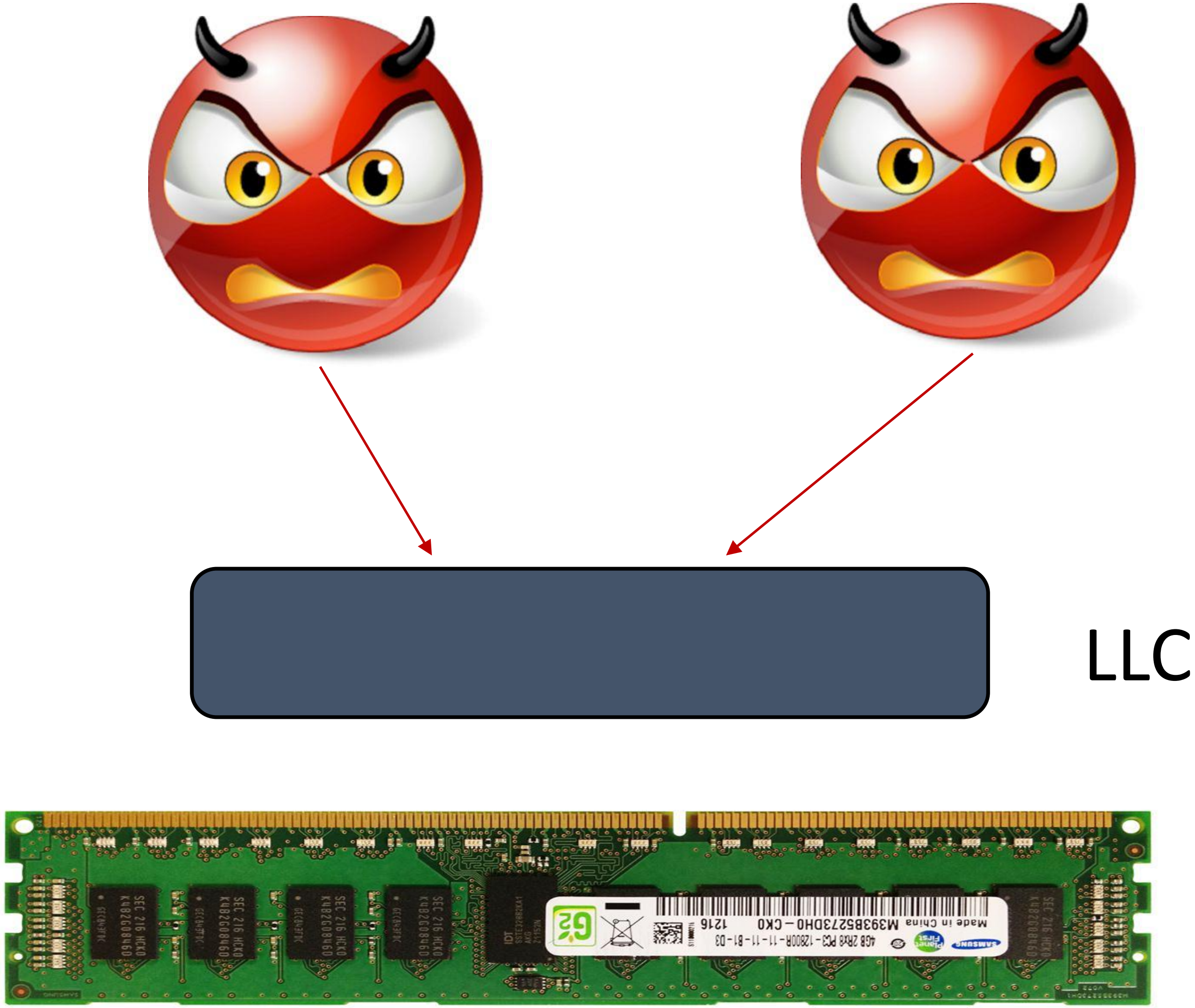
LRU is not effective for Shared Caches



Intuition: New cache block will not be re-referenced soon. Replaces block with distant RRPV. Only two bits per block.

Insert with RRPV=2, Evict with RRPV=3, increment RRPVs till we get a block with RRPV=3, promote blocks with RRPV=0.

Evict + Reload: Covert Channel (sharing but no flush)



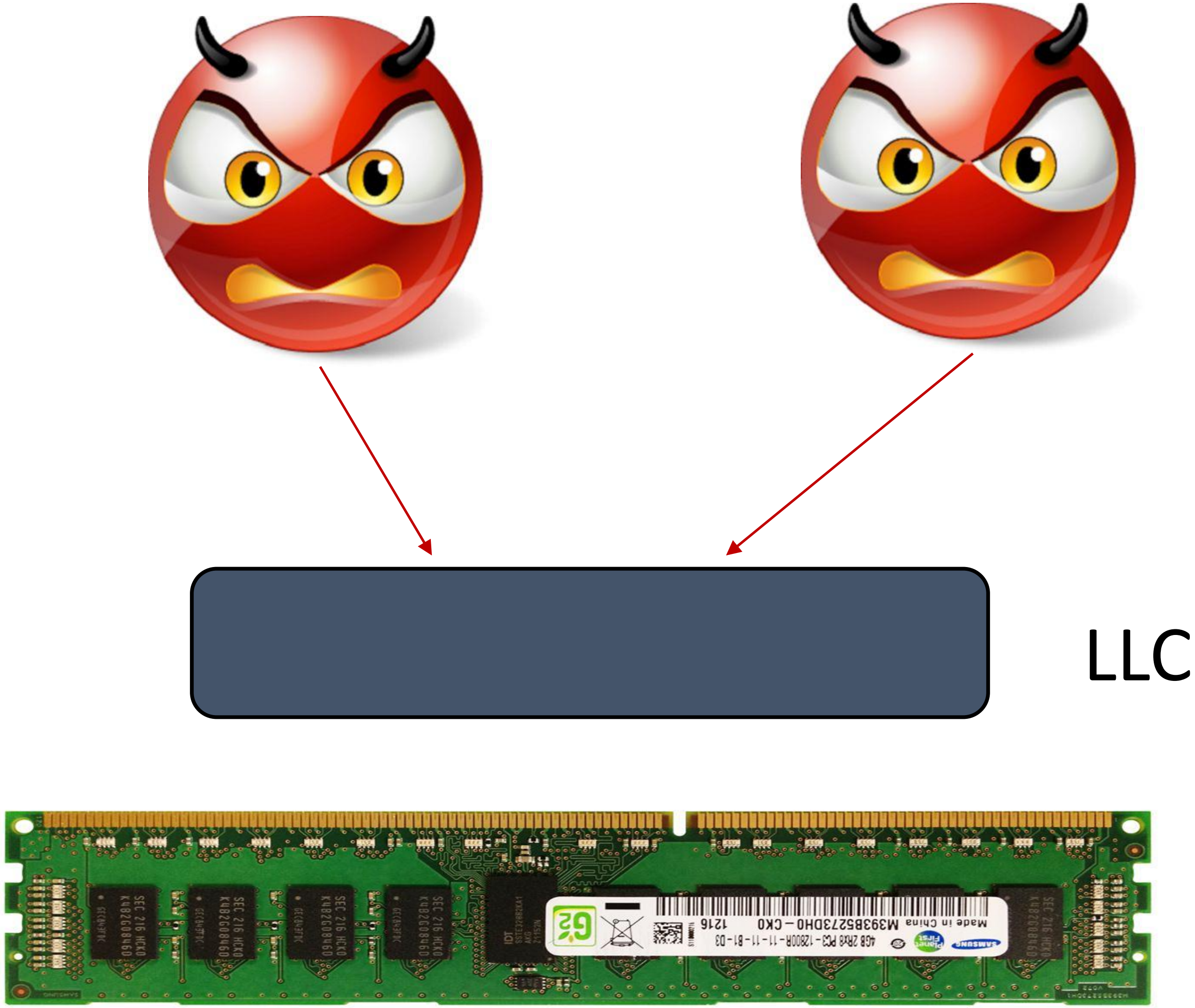
Step 0: Receiver gets data from cache (fast, bit “0”)

Step 1: Sender thrashes cache and evicts the shared address

Step 2: Receiver gets data from DRAM (slow, bit “1”)

Devil is in the details: Discussion time ☺

Prime + Probe: Covert Channel (No sharing)

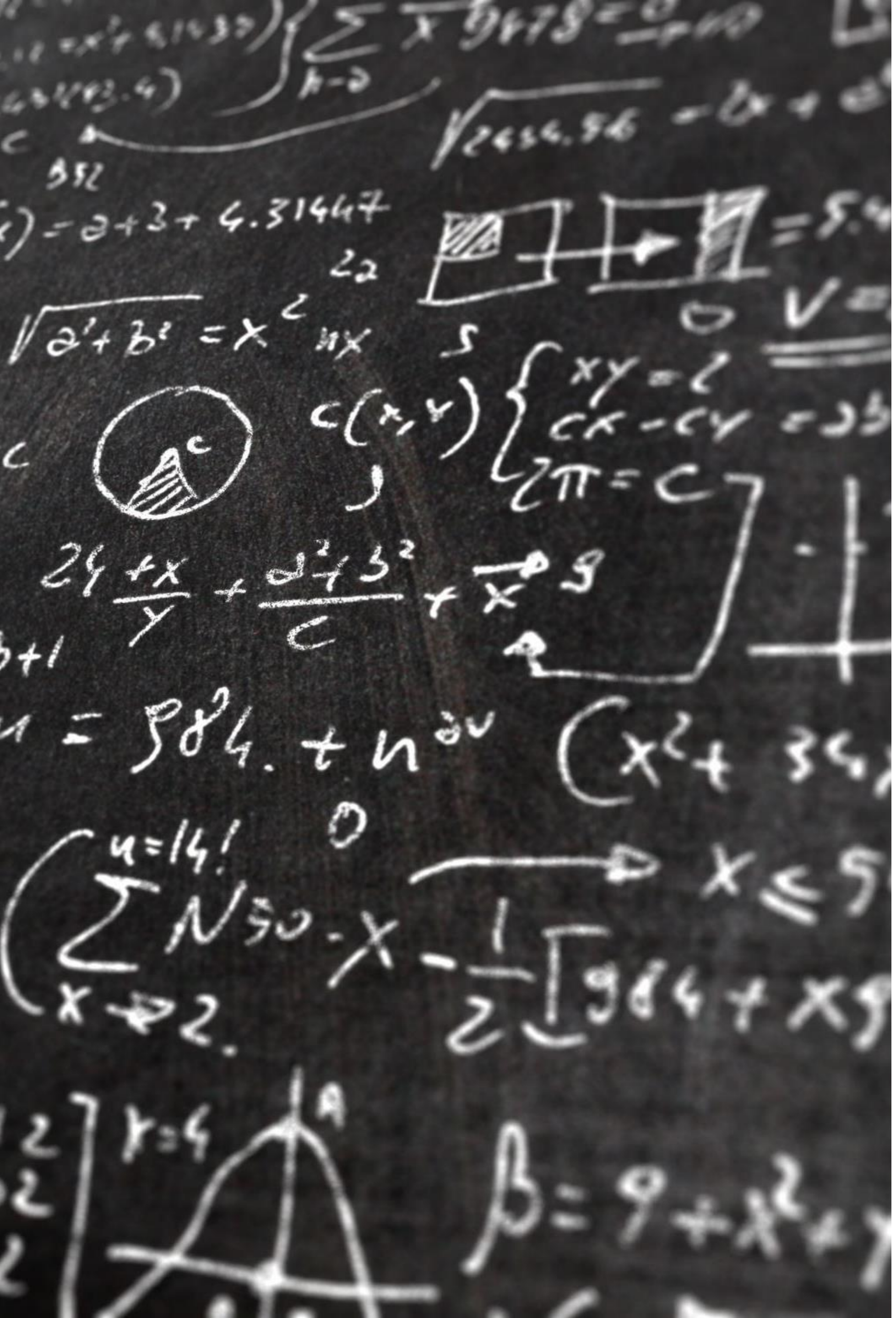


Step 0: Receiver gets data from cache (fast, bit “0”)

Step 1: Sender thrashes cache

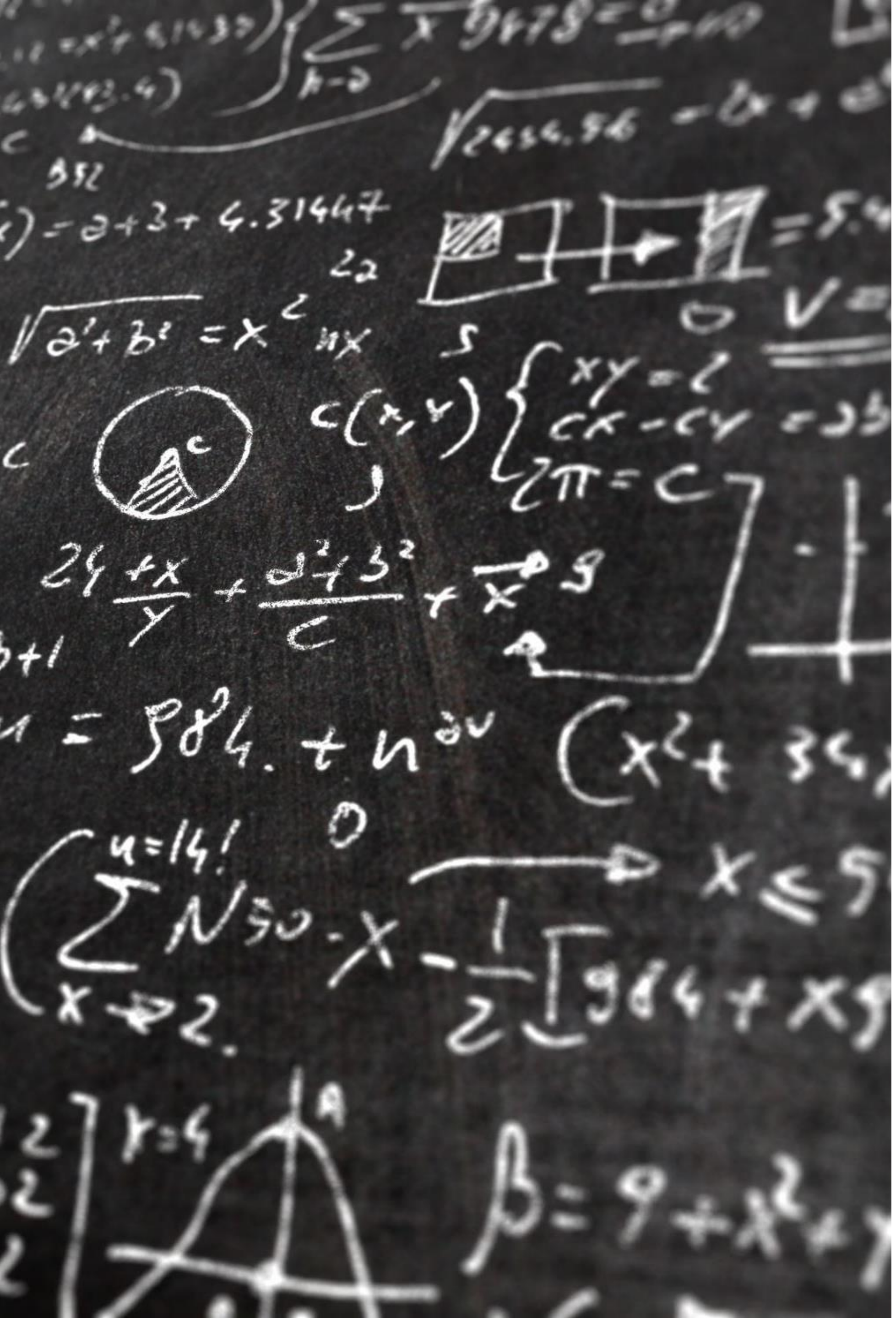
Step 2: Receiver gets data from DRAM (slow, bit “1”)

Devil is in the details: Discussion time ☺



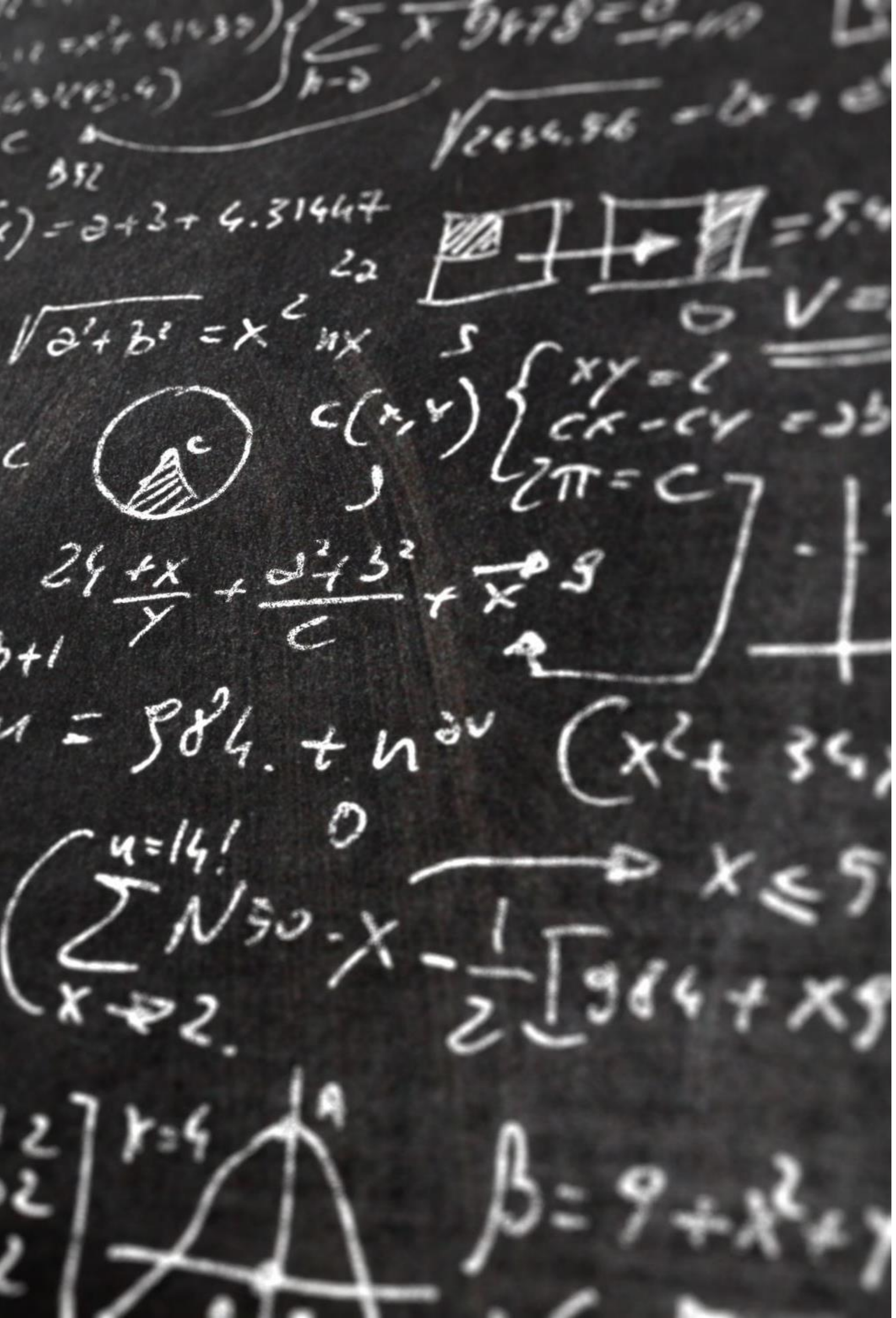
Summary so far

- Flush + Reload: Demands Page Sharing, Fine-grained, low-noise (variant is flush+flush)
- Evict + Reload: An alternative for Flush + Reload, Almost equally effective
- Prime + Probe: Does not demand page sharing, coarse grained, high-noise, need to find out the eviction set



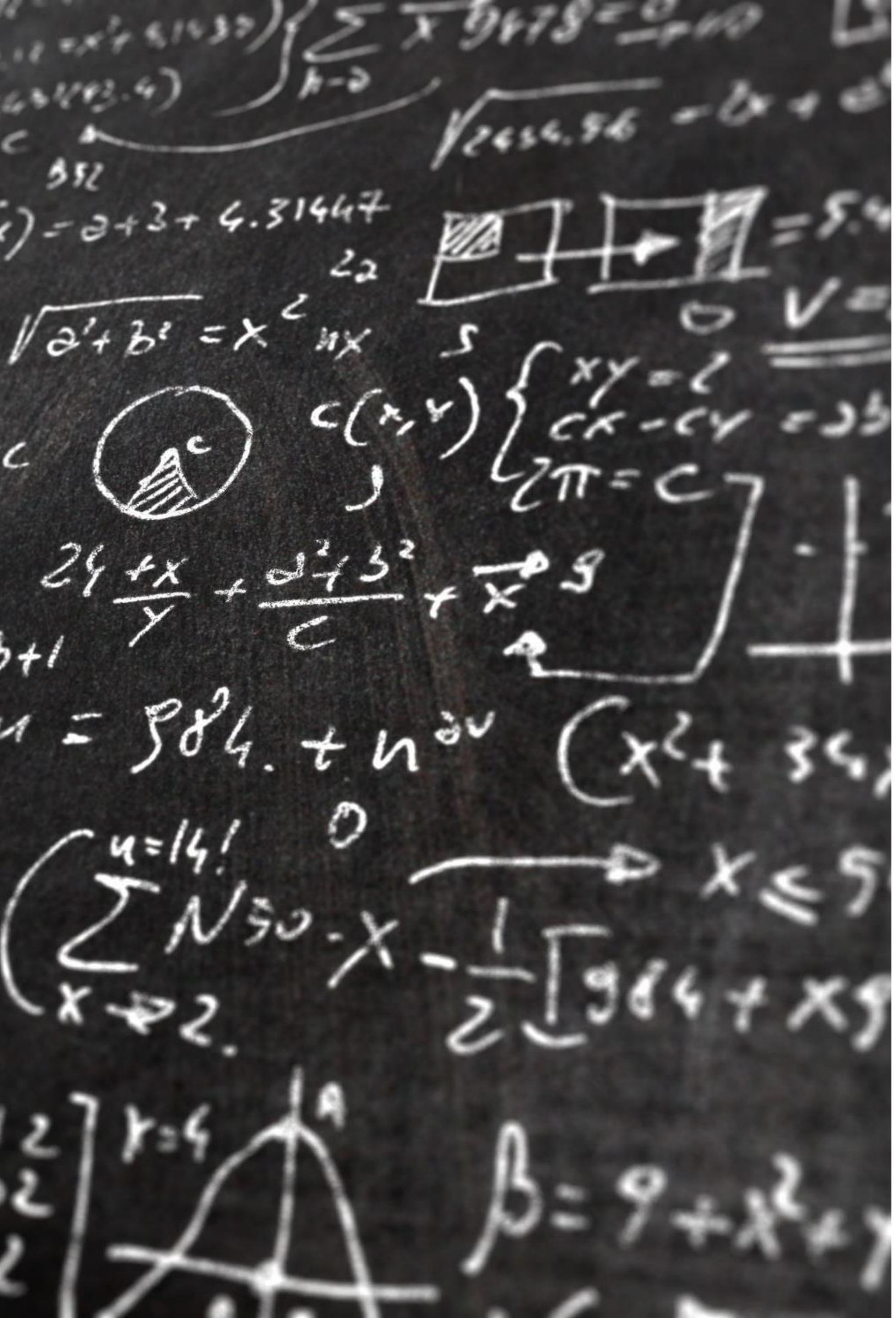
More on Flush based …

- Works across CPU sockets
- Works on non-inclusive caches
- However, it can only recover statically allocated data



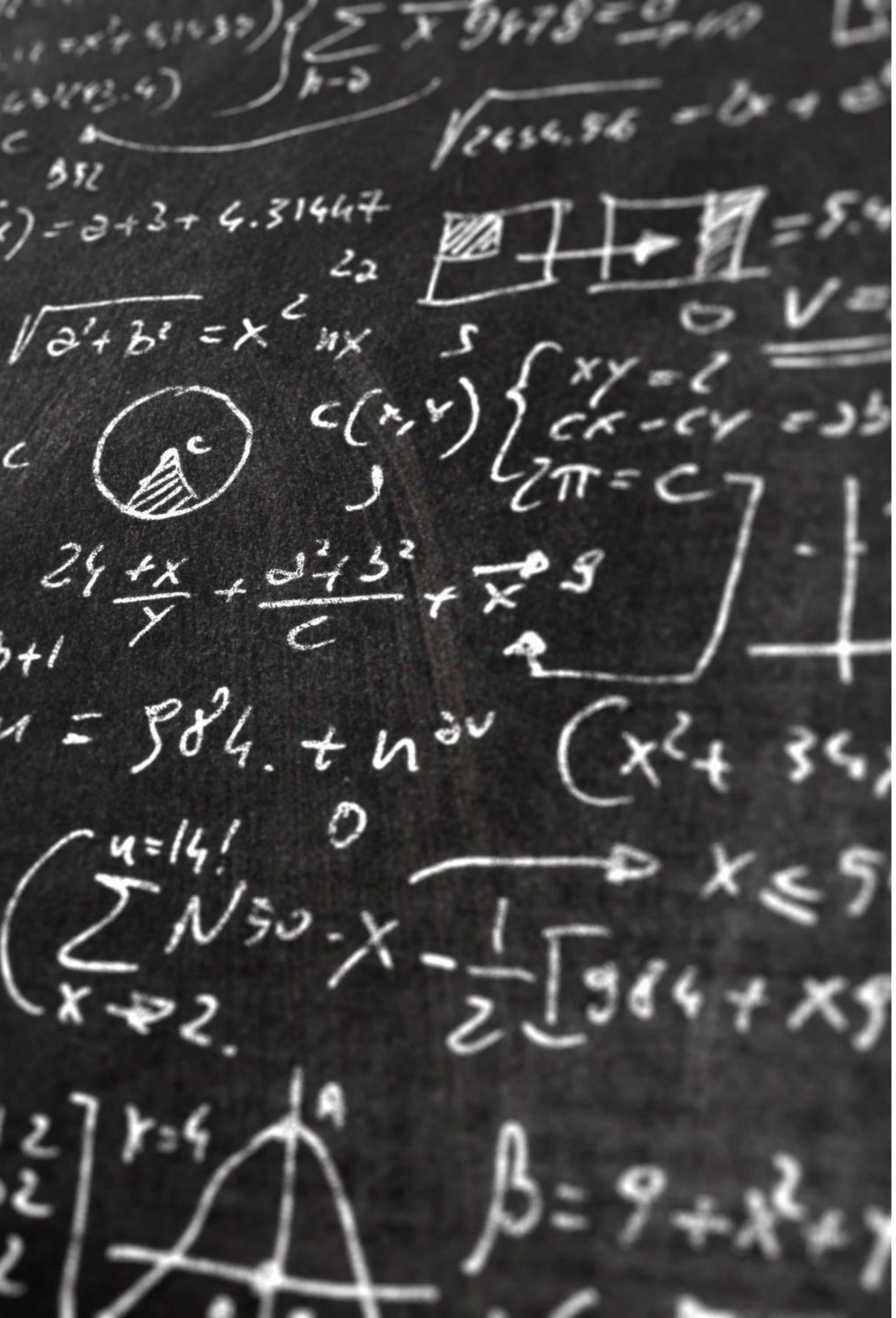
More on Flush based …

- Works across CPU sockets
- Works on non-inclusive caches
- However, it can only recover statically allocated data



More on Eviction based …

- No flush instruction
- Attacker can use huge pages
- Applicable to processors without clflush instruction
- Only work with inclusive caches in the same CPU socket
- Need information about LLC slices



Paper to read/mock-review

http://palms.ee.princeton.edu/system/files/SP_vfinal.pdf

Last-Level Cache Side-Channel Attacks are Practical (S&P 2015)