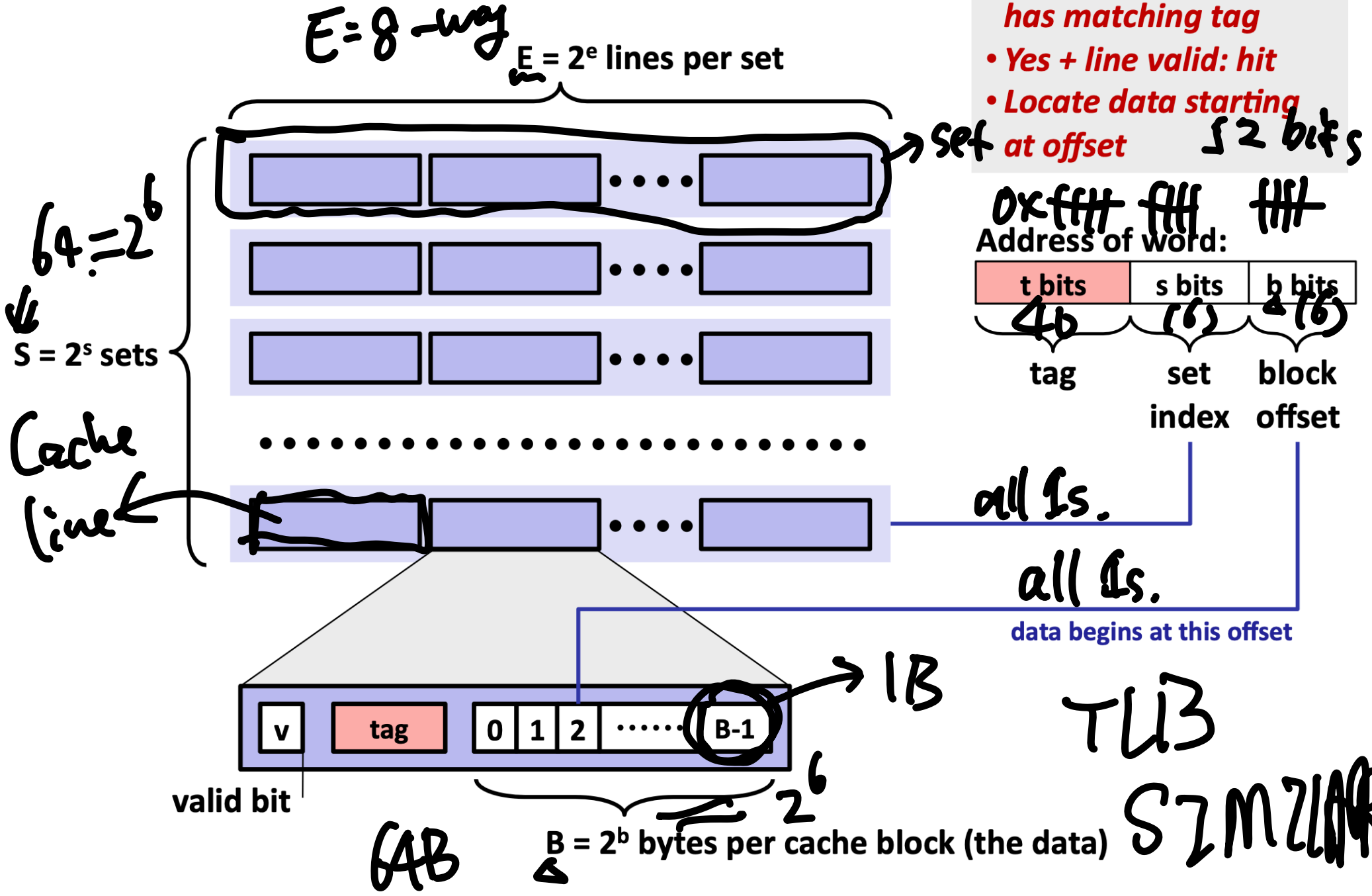
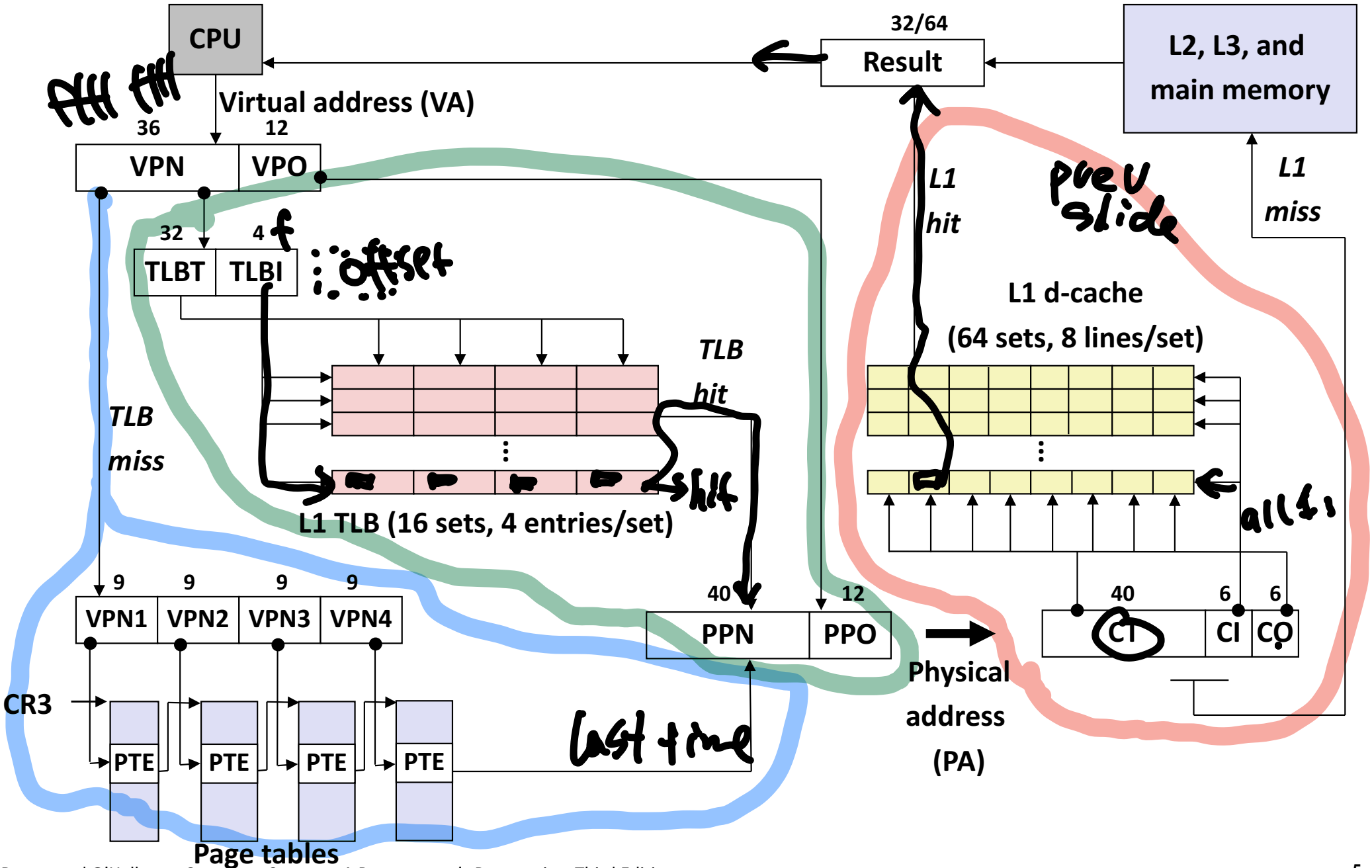
Cache Read

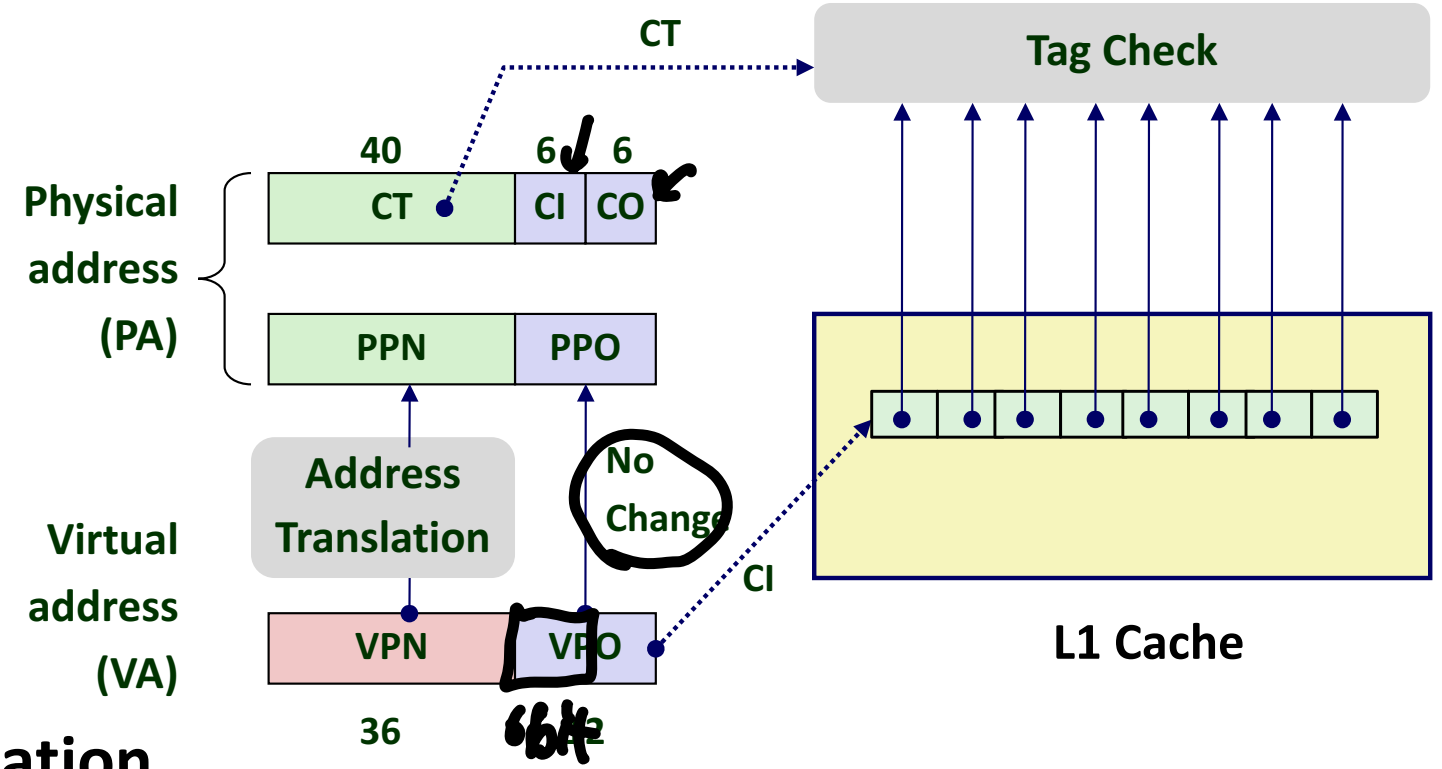
- *Locate set*
- *Check if any line in set has matching tag*
- *Yes + line valid: hit*
- *Locate data starting at offset*



End-to-end Core i7 Address Translation



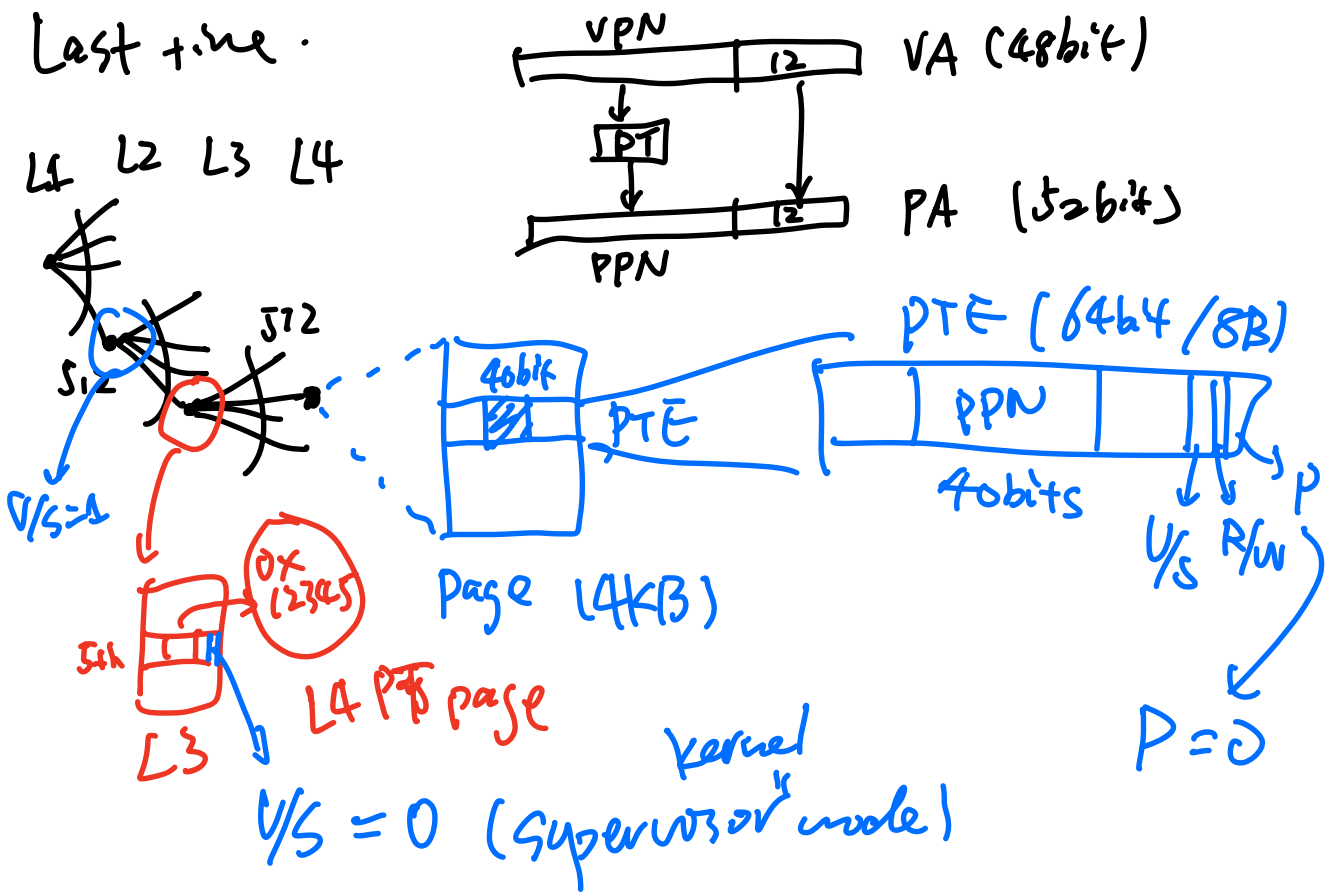
Cute Trick for Speeding Up L1 Access



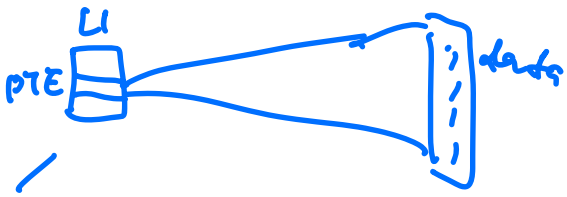
- **Observation**
  - Bits that determine CI identical in virtual and physical address
  - Can index into cache while address translation taking place
  - Cache carefully sized to make this possible: 64 sets, 64-byte cache blocks
  - Means 6 bits for cache index, 6 for *cache* offset
  - That's 12 bits; matches *VPO, PPO* → One reason pages are 2^{12} bits = 4 KB

Week 10.b
CS 5600
03/15 2023

1. page table practice
2. TLBs
3. Where does the OS live?
4. page fault



Practice



A: memory that different PTEs can address

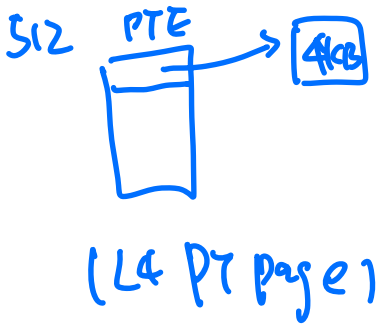
--Question: how much memory can one L1 page entry address?

64GB. L1 PTE → L2 → L3 → L4 → data page (4KB)

40bit

512 * 512 * 512 * 4KB = 2^{27} KB

--Question: so how much virtual memory is each L4 page *table* responsible for translating? 4KB? 2MB? 1GB?



512 x 4KB = 2MB

PTE

$= 2^{29}$ KB

$= 2^{19}$ MB

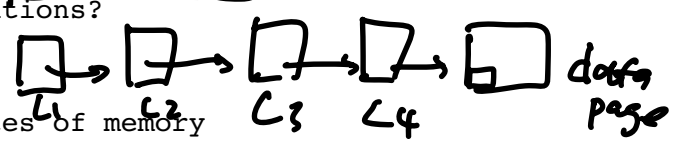
$= 2^{9}$ GB

$= 512$ GB

B. Allocating memory

What is the minimum number of physical pages required on x86-64 to allocate the following allocations?

1 byte of memory ⇒ 5



1 allocation of size 2^12 bytes of memory

(b) 2^9 allocations of size of 2^12 bytes of memory each ⇒

(c) 2^9 + 1 allocations of size of 2^12 bytes of memory each

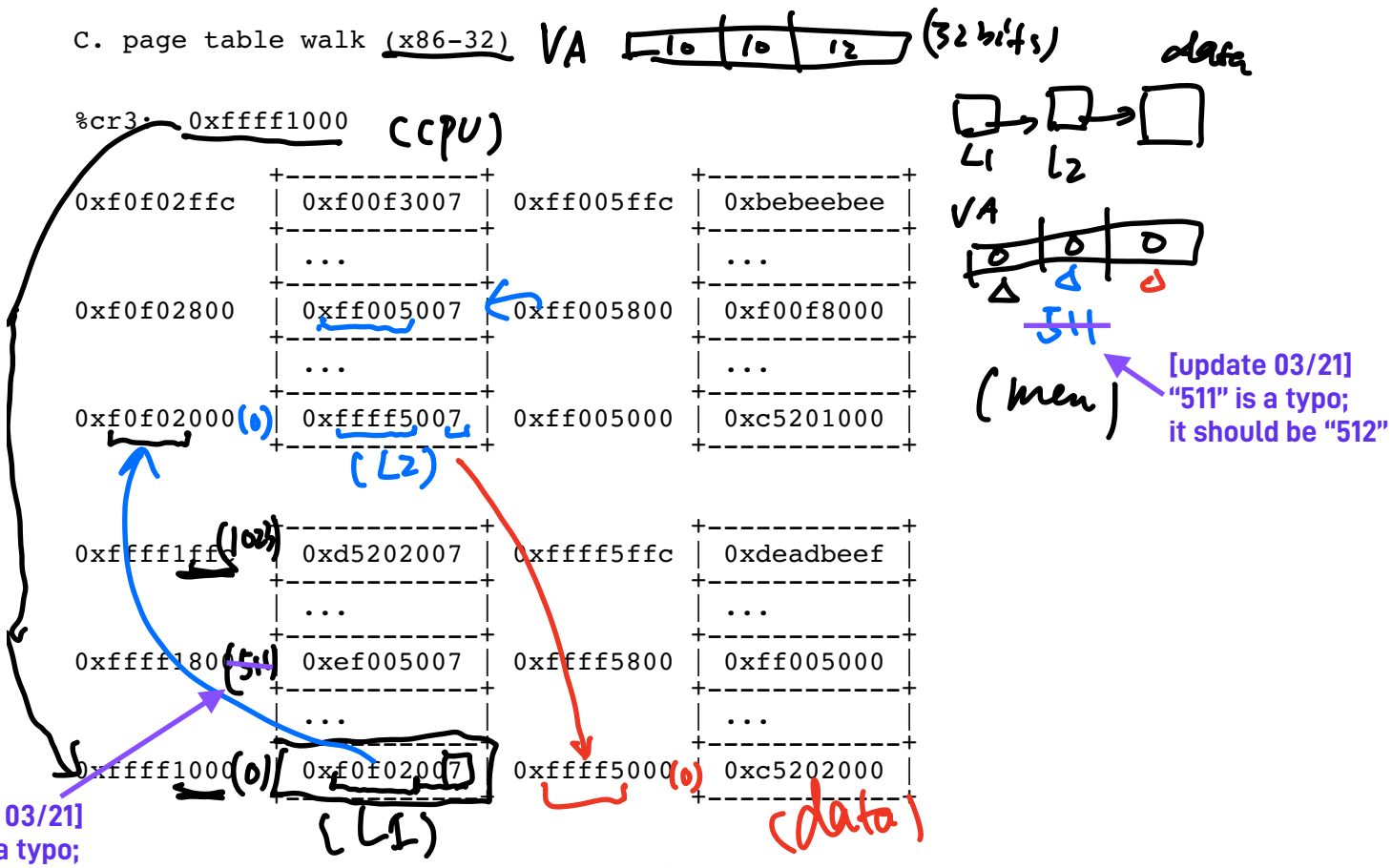
2^18 + 1 allocations of size 2^12 bytes of memory each

(b) L4 : 1 + L2 : 1 L3 : 1 + L4 : 512 PTEs (1 page) + 512 data = 4 + 512

(c) :

1 + 1 + 1 + 2 + 513 (data) = 5 + 513.

C. page table walk (x86-32)



[update 03/21] "511" is a typo; it should be "512"

[update 03/21] "511" is a typo; it should be "512"

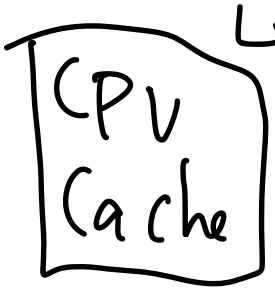
- What's the output of the following C excerpt?

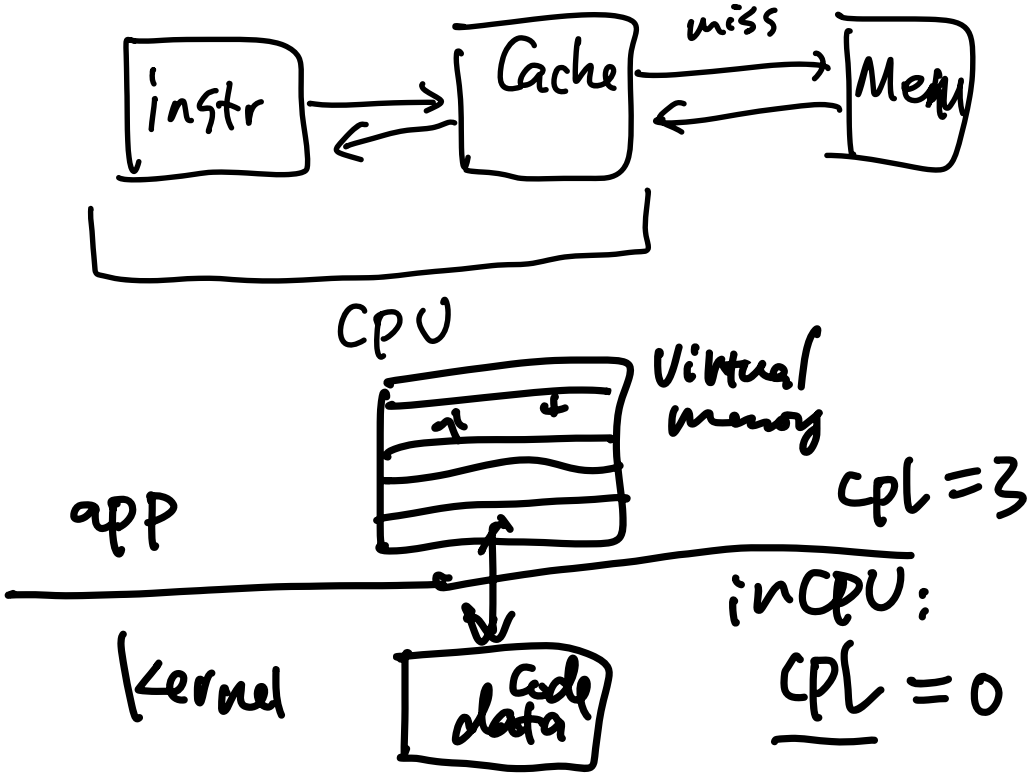
```
int *ptr1 = (int *) 0x0;
printf("%x\n", *ptr1);
```

0xC5202000

TLB: translation lookaside buffer

↳ Cache. mapping: VPN → PPN





Option 1: kernel own space

Lab4. (%cr3)

⇒ Option 2: kernel app share the same space

Virtual Address Space of a Linux Process