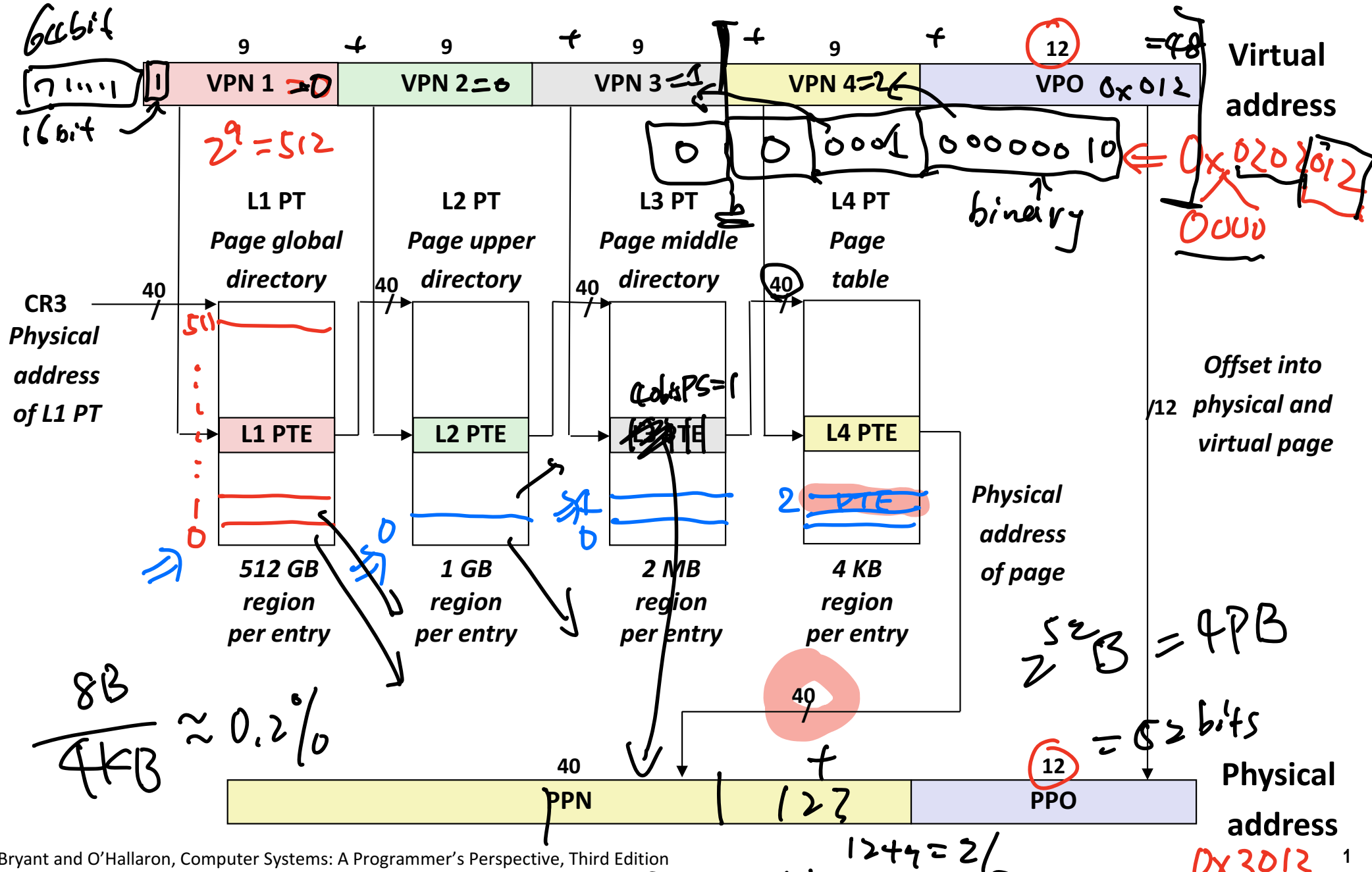


# Core i7 Page Table Translation



1GB  $2^{21} = 2MB$  (1.9MB)

# Review of Symbols

## ■ Basic Parameters

- $N = 2^n$  : Number of addresses in virtual address space
- $M = 2^m$  : Number of addresses in physical address space
- $P = 2^p$  : Page size (bytes)

## ■ Components of the virtual address (VA)

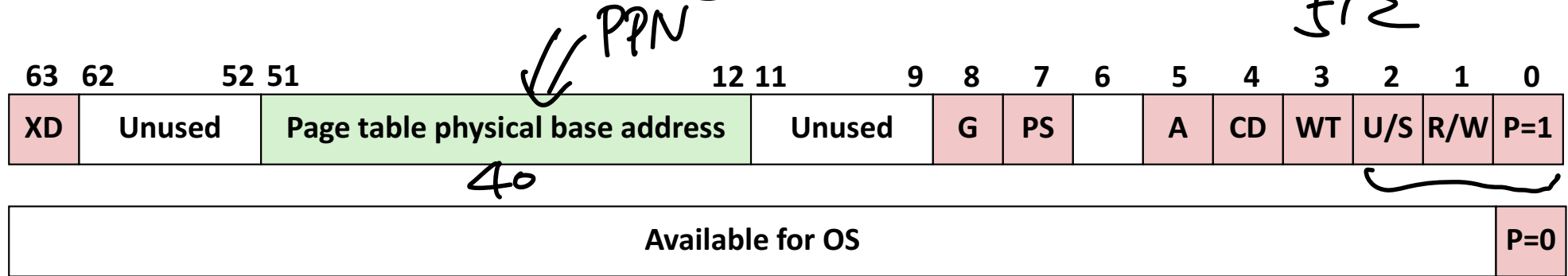
- TLBI: TLB index
- TLBT: TLB tag
- VPO: Virtual page offset
- VPN: Virtual page number

## ■ Components of the physical address (PA)

- PPO: Physical page offset (same as VPO)
- PPN: Physical page number
- CO: Byte offset within cache line
- CI: Cache index
- CT: Cache tag

# Core i7 Level 1-3 Page Table Entries

$$\frac{4KB}{512} = 8B$$



Each entry references a 4K child page table. Significant fields:

**P:** Child page table present in physical memory (1) or not (0).

**R/W:** Read-only or read-write access access permission for all reachable pages.

**U/S:** user or supervisor (kernel) mode access permission for all reachable pages.

**WT:** Write-through or write-back cache policy for the child page table.

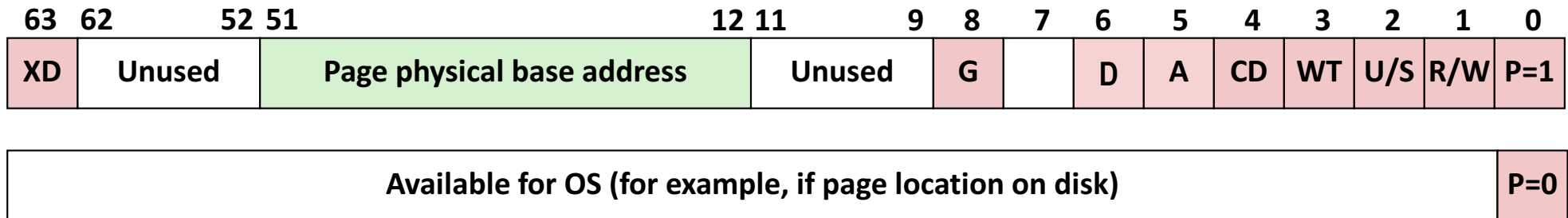
**A:** Reference bit (set by MMU on reads and writes, cleared by software).

**PS:** <sup>for L3</sup> Page size: if bit set, we have 2 MB or 1 GB pages (bit can be set in Level 2 and 3 PTEs only).

**Page table physical base address:** 40 most significant bits of physical page table address (forces page tables to be 4KB aligned)

**XD:** Disable or enable instruction fetches from all pages reachable from this PTE.

# Core i7 Level 4 Page Table Entries



**Each entry references a 4K child page. Significant fields:**

**P:** Child page is present in memory (1) or not (0)

**R/W:** Read-only or read-write access permission for this page

**U/S:** User or supervisor mode access

**WT:** Write-through or write-back cache policy for this page

**A:** Reference bit (set by MMU on reads and writes, cleared by software)

**D:** Dirty bit (set by MMU on writes, cleared by software)

**Page physical base address:** 40 most significant bits of physical page address  
(forces pages to be 4KB aligned)

**XD:** Disable or enable instruction fetches from this page.

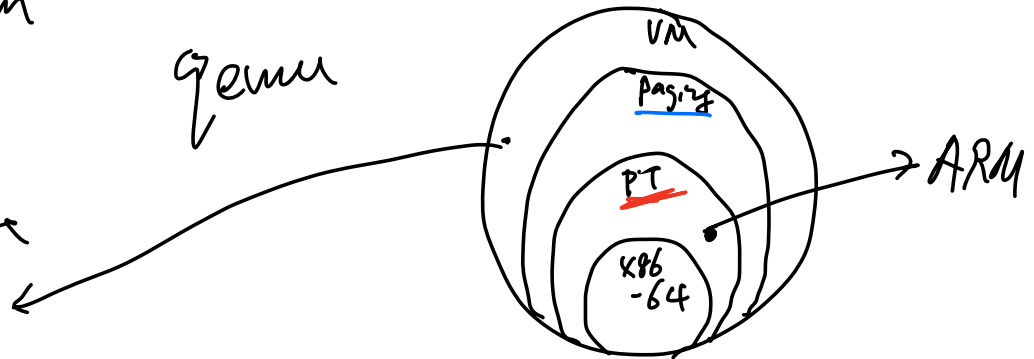
Week 10.a  
 CS 5600  
 03/13 2023

1. Last time
  2. x86-64: addresses
    - virtual
    - physical
  3. x86-64: page table structures
  4. Practice
- 

## Admin

- midterm
- Lab4.
- Last time
  - VM

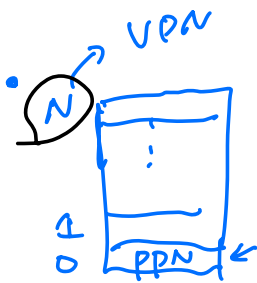
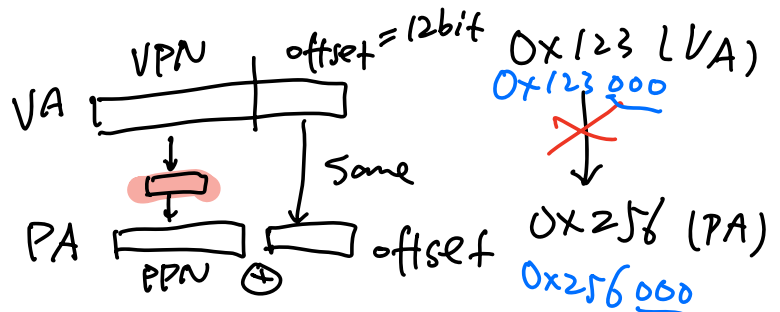
Genus



VA → PA

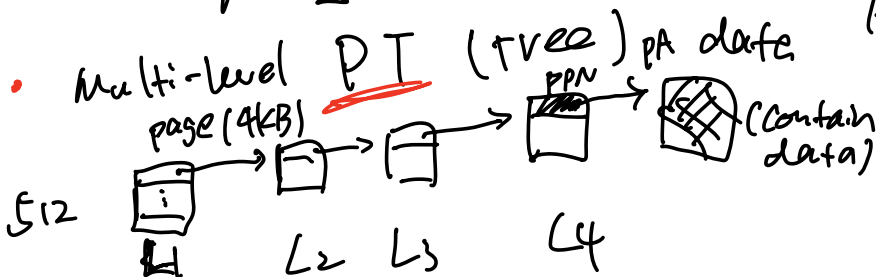
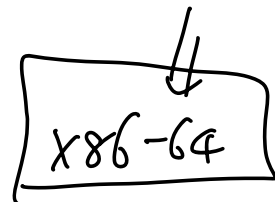
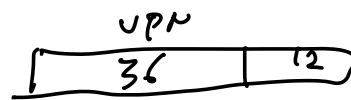
• Paging

VPN → PPN

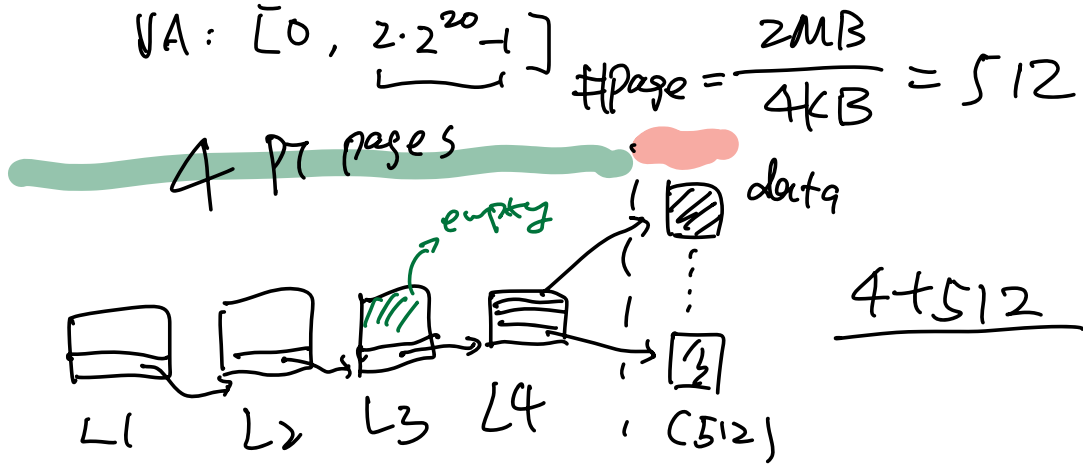


$$N = 2^{36}$$

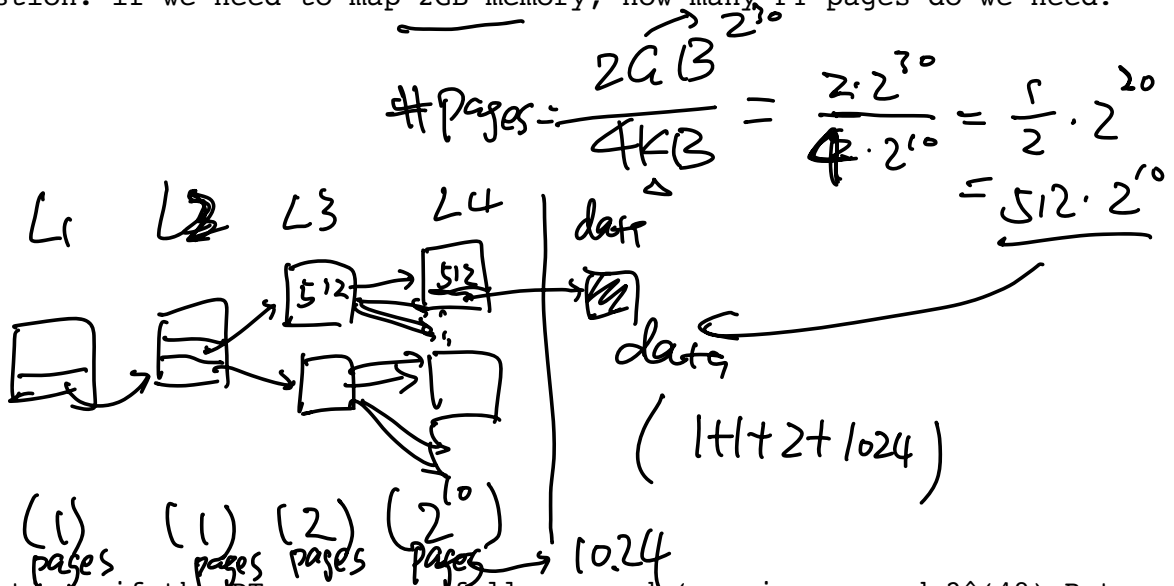
48 bit VA



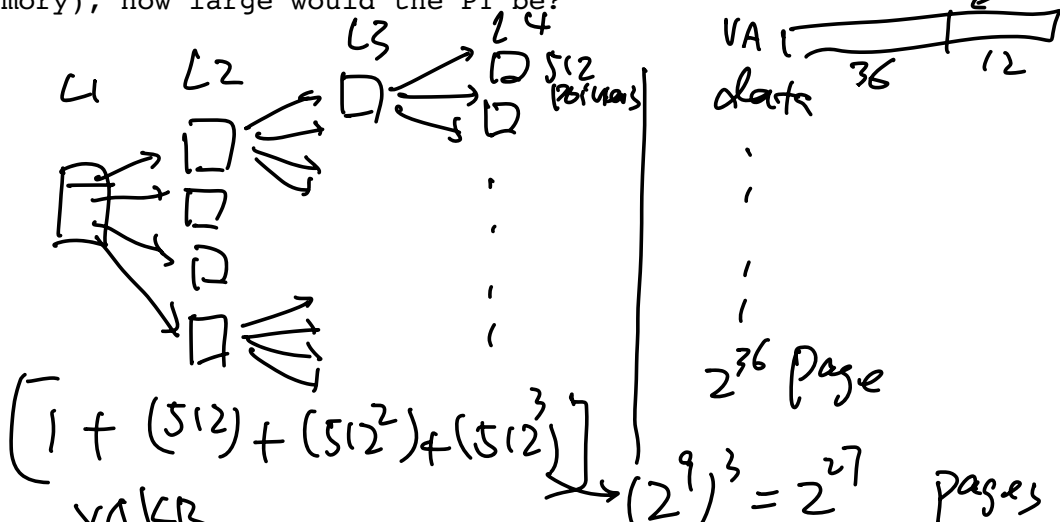
-- Question: Given one page can have 512 pointers, if we need to map 2MB memory, how would the tree look like?



-- Question: if we need to map 2GB memory, how many PT pages do we need?



-- Question: if the PT pages are fully mapped (meaning mapped  $2^{48}$  Bytes memory), how large would the PT be?



1417

$$\begin{aligned} &= 2^{27} \cdot 2^9 \text{ pointers} \\ &= 2^{36} \end{aligned}$$