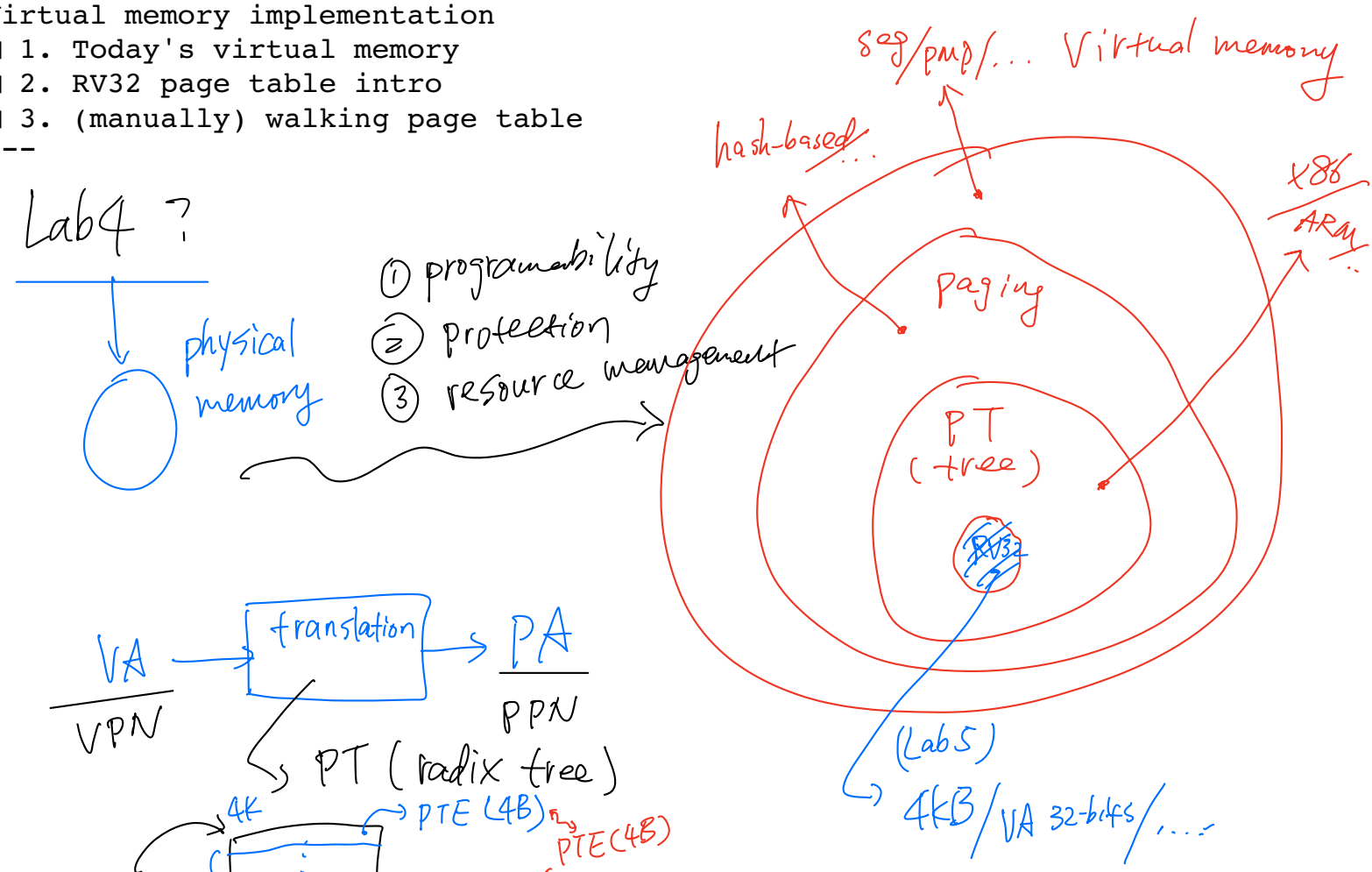
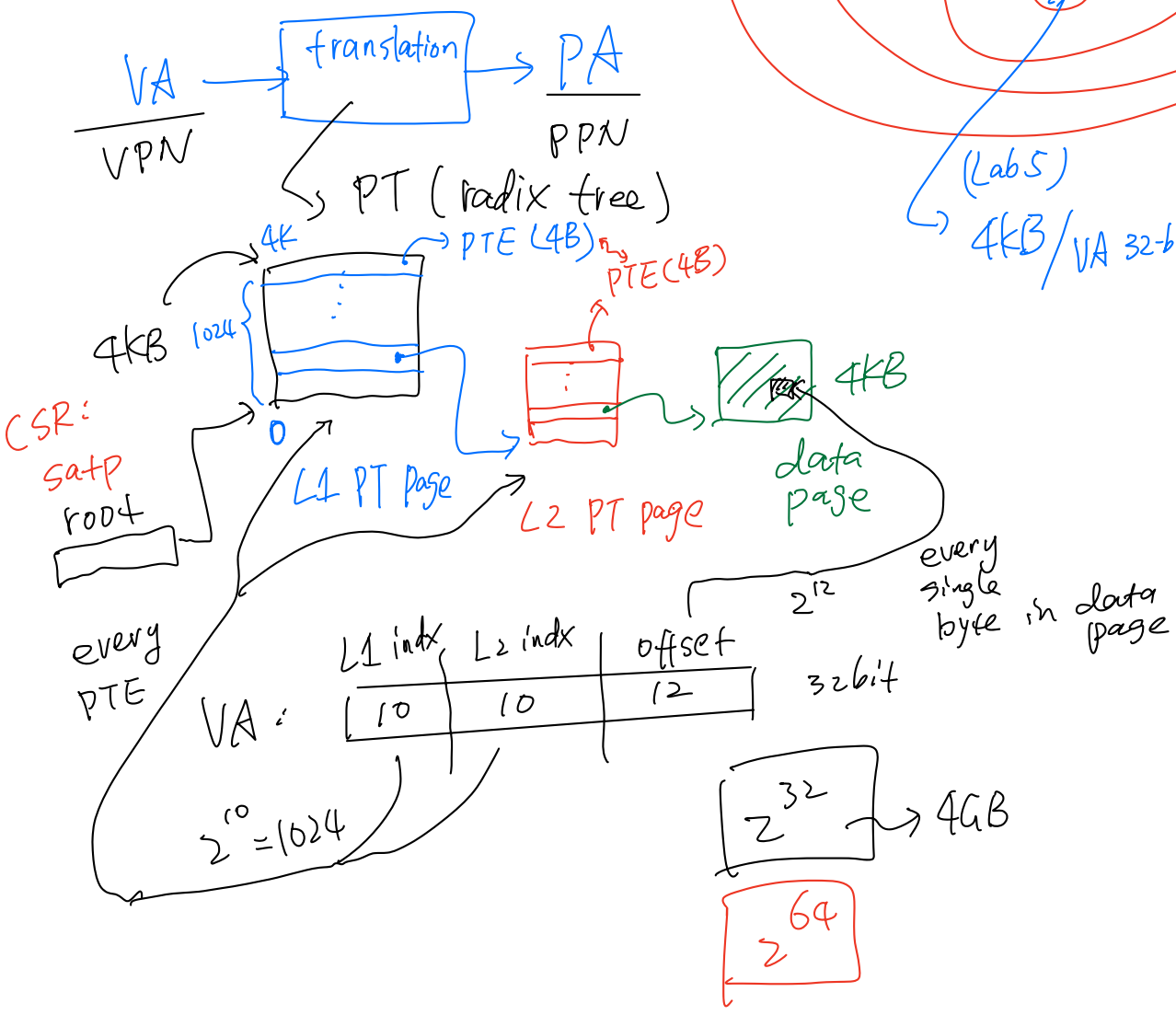
Week 7.a
CS6640
02/17 2026
https://naizhengtan.github.io/26spring/

Virtual memory implementation
- [ ] 1. Today's virtual memory
- [ ] 2. RV32 page table intro
- [ ] 3. (manually) walking page table

---




(20bits)

VA: 0x803fffec

0x803ff

80 | 0011 | ff

0 3

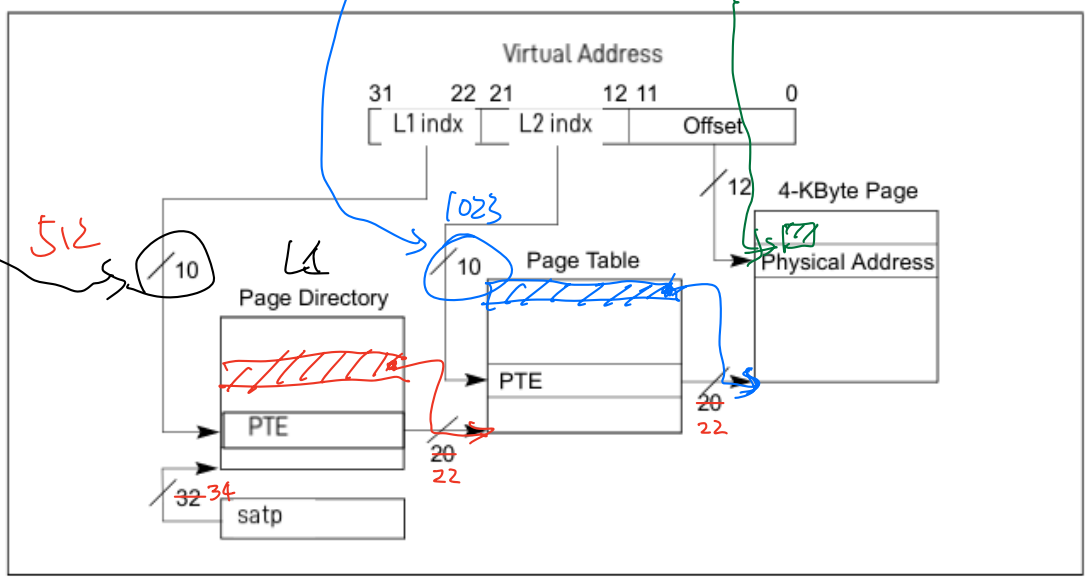
0x600 >> 2

= 0x200

L1: 0x200 → 512

L2: 0x3ff → 1023

offset: 0xfec



OSI Week7a

**1. CSR satp (page table root)**

physical page number

$2^{22} \cdot 2^{12} \rightarrow 2^{34}$

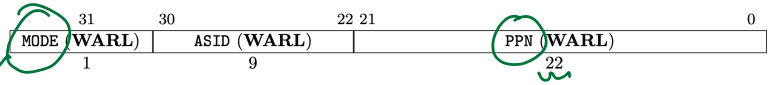


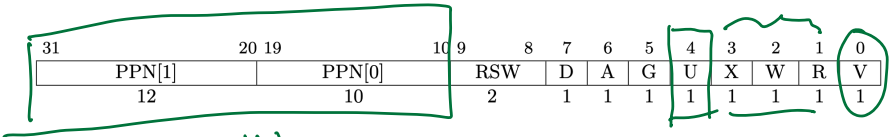

Figure 4.14: Supervisor address translation and protection register **satp** when SXLEN=32.

(1-bit MODE)

| SXLEN=32 | | |
|---|---|---|
| Value | Name | Description |
| 0 | Bare | No translation or protection. |
| 1 | Sv32 | Page-based 32-bit virtual addressing (see Section 4.3). |

4B

**2. Page table entry (PTE)**




PPN (22 bits)

Figure 4.18: Sv32 page table entry.

addressing PTE

**3. Virtual address (VA)**

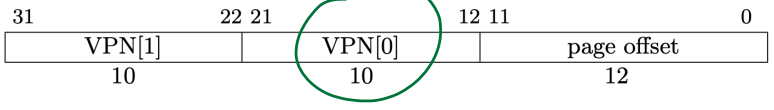
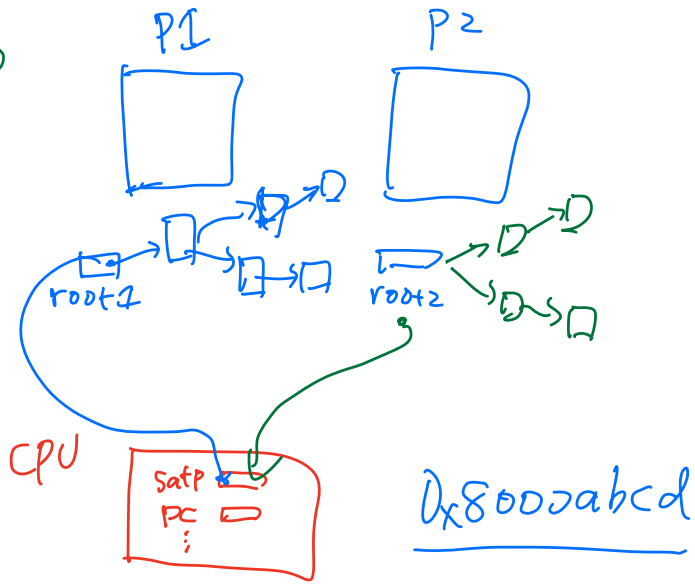



Figure 4.16: Sv32 virtual address.



* simulate CPU: manual page walk
  (1) split the VA to l1/l2 indexes and offest
  (2) get the root of page table
  (3) calc the L1 page
  (4) calc the L2 page
  (5) calc the physical address