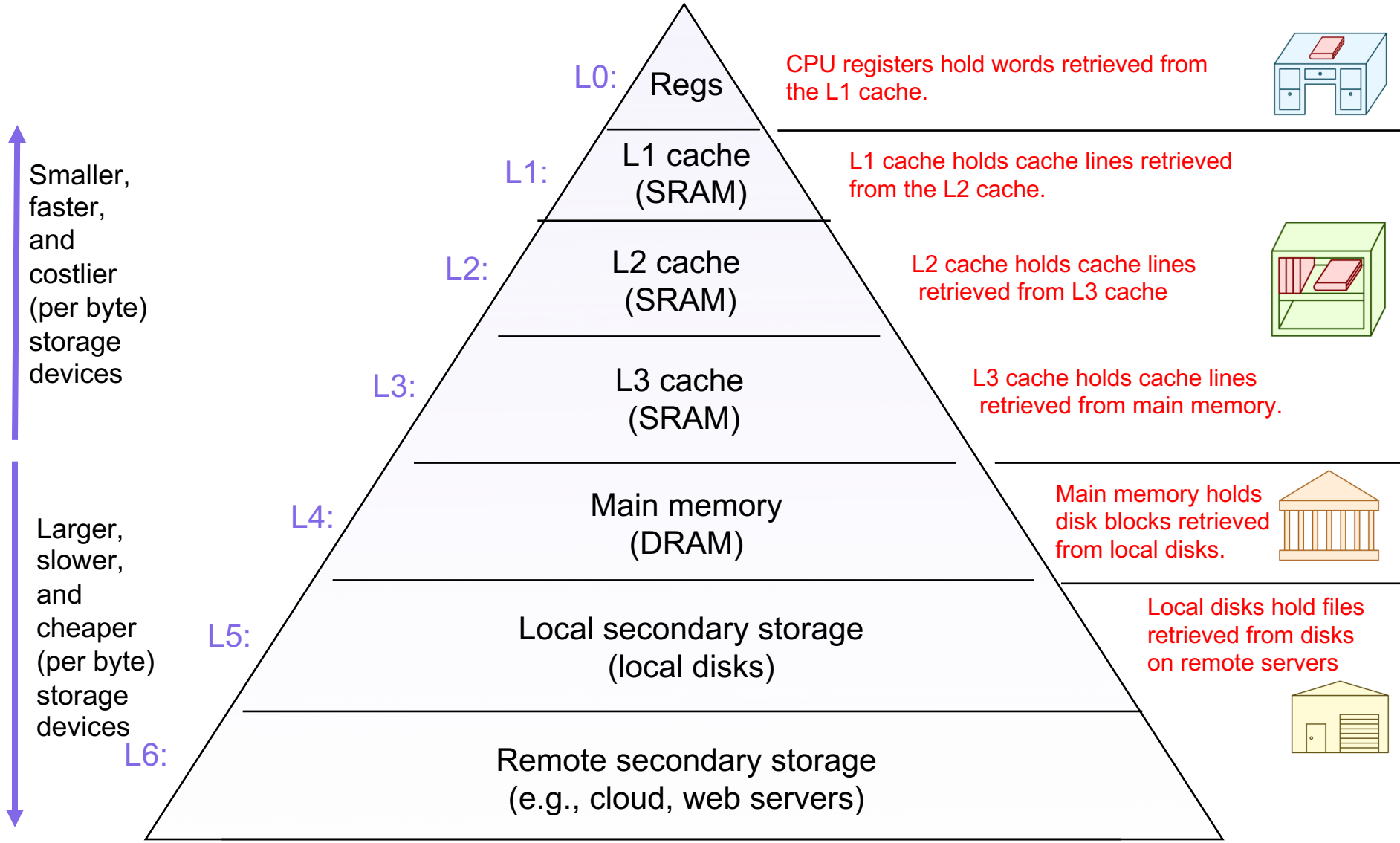
Lecture 15: Virtual Memory

CS 105

March 13, 2019

Review: Memory Hierarchy

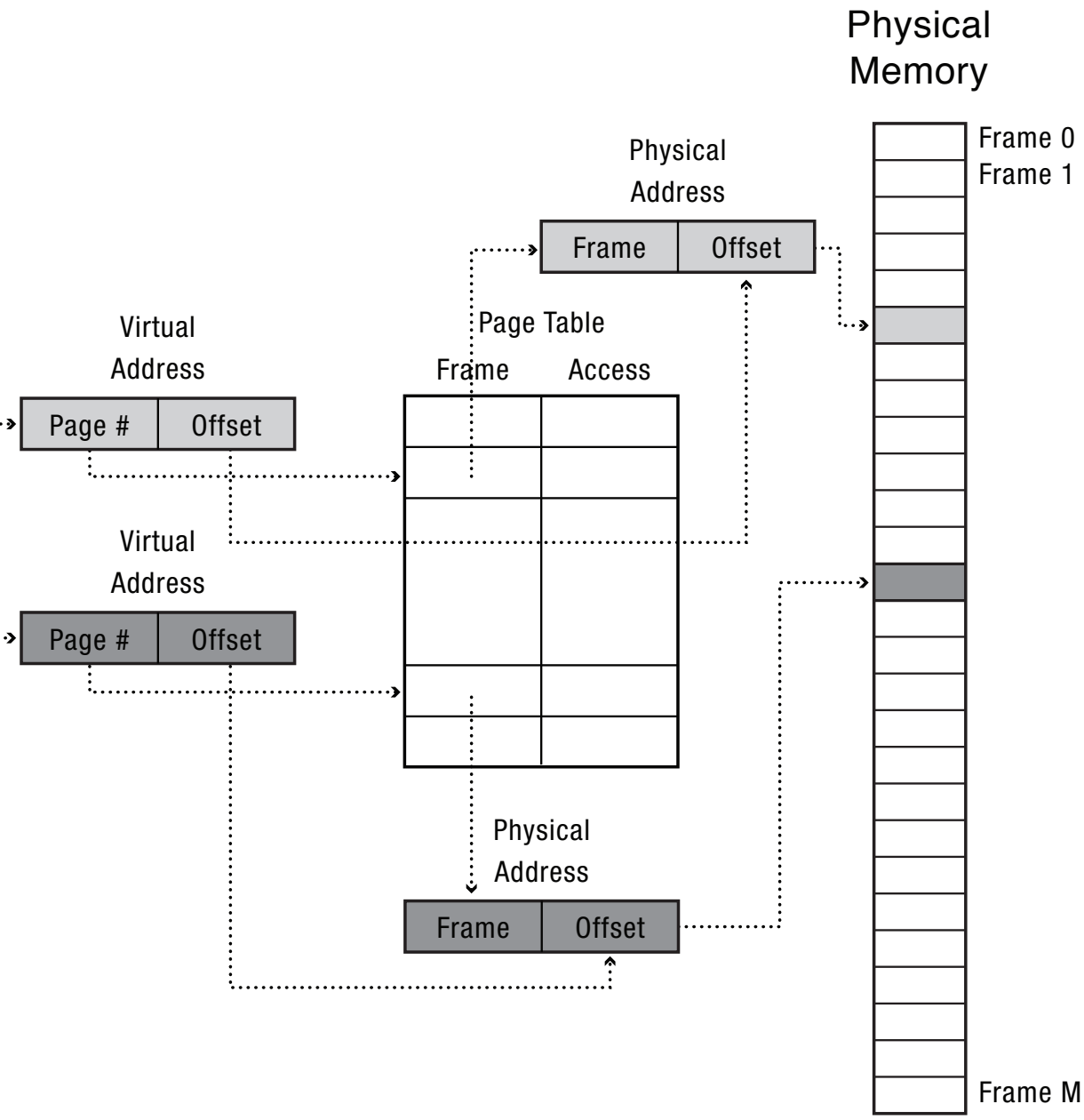


Base-and-Bound

Segmentation

Paging

Paging


Physical Memory
Physical Address
Frame
Offset
Frame 0
Frame 1
Page Table
Frame
Access
Virtual Address
Page #
Offset
Virtual Address
Page #
Offset
Physical Address
Frame
Offset
Frame M

Page Tables

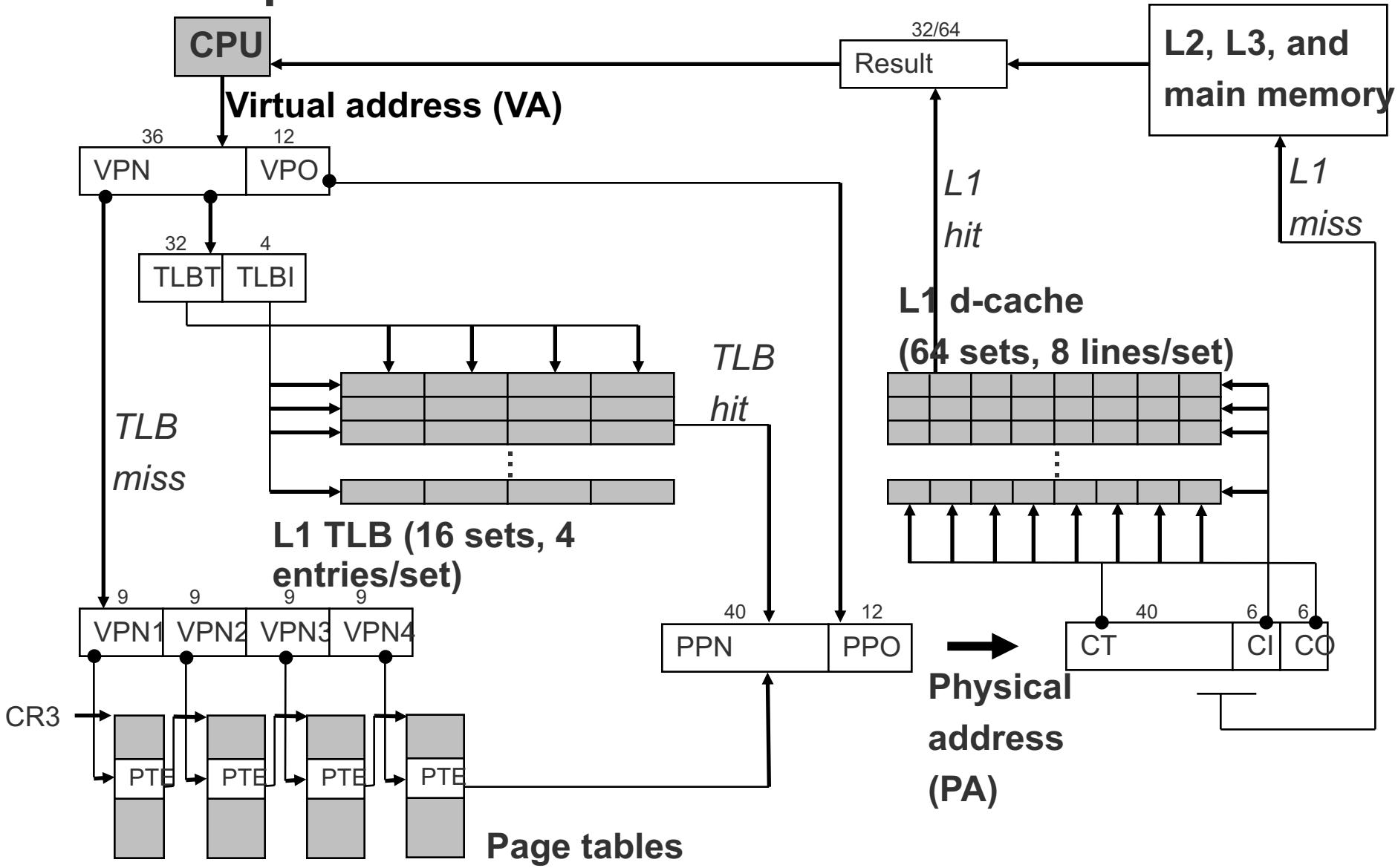
- each page table entry has a valid bit
- for valid entries, frame indicates physical address of page in memory
- Access bits indicate permissions (r,w,x, os vs user space)

- **translation lookaside buffer (TLB):** special cache for page table entries
- how big should you make the pages?
- how big would a page table be
  - in practice, use multi-level page tables

Memory as a Cache

- each page table entry has a valid bit
- a **page fault** occurs when a program requests a page that is not currently in memory

Example: Core i7 Address Translation



Meltdown and Spectre

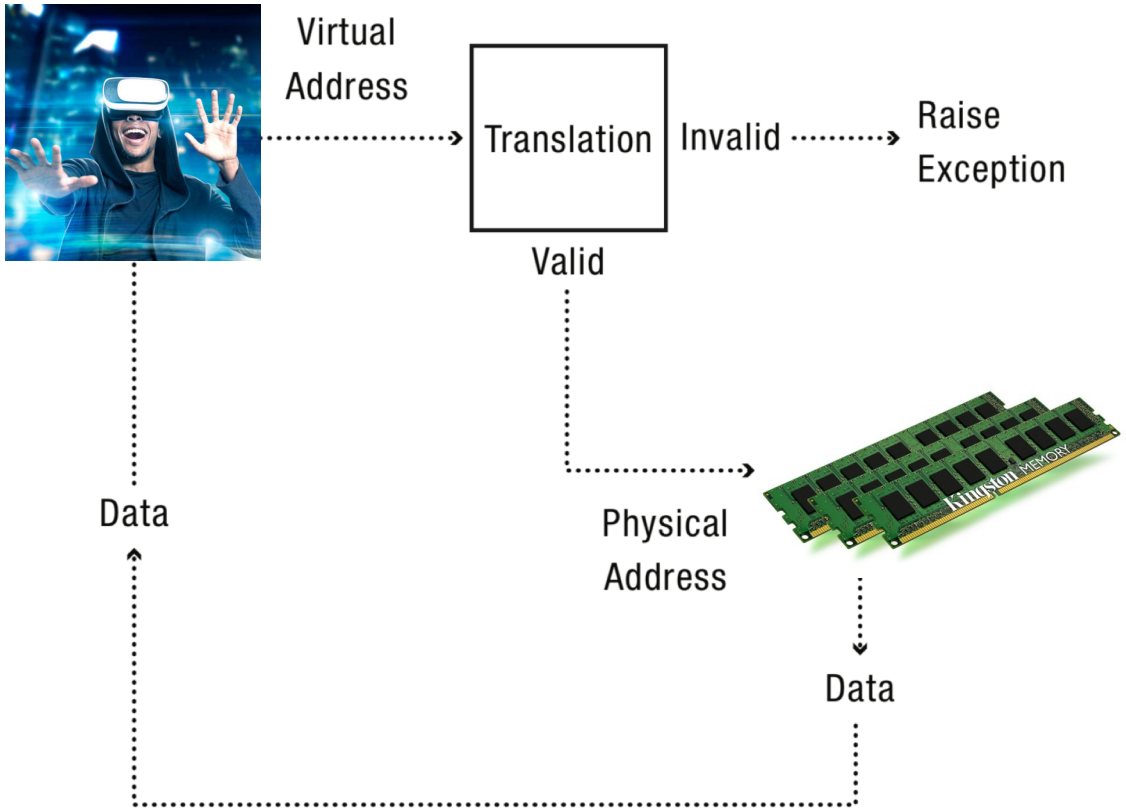
Spectre

```
int x, len1, len2;
int a1[len1];
int a2[len2];

if (x < len1){
    int ival= a1[x];

    if(ival < len2){
        b2 = a2[ival];
    }
}
```

Virtual Memory



And that's the end of Part 1