

# ADDRESS TRANSLATION AND TLB

Mahdi Nazm Bojnordi

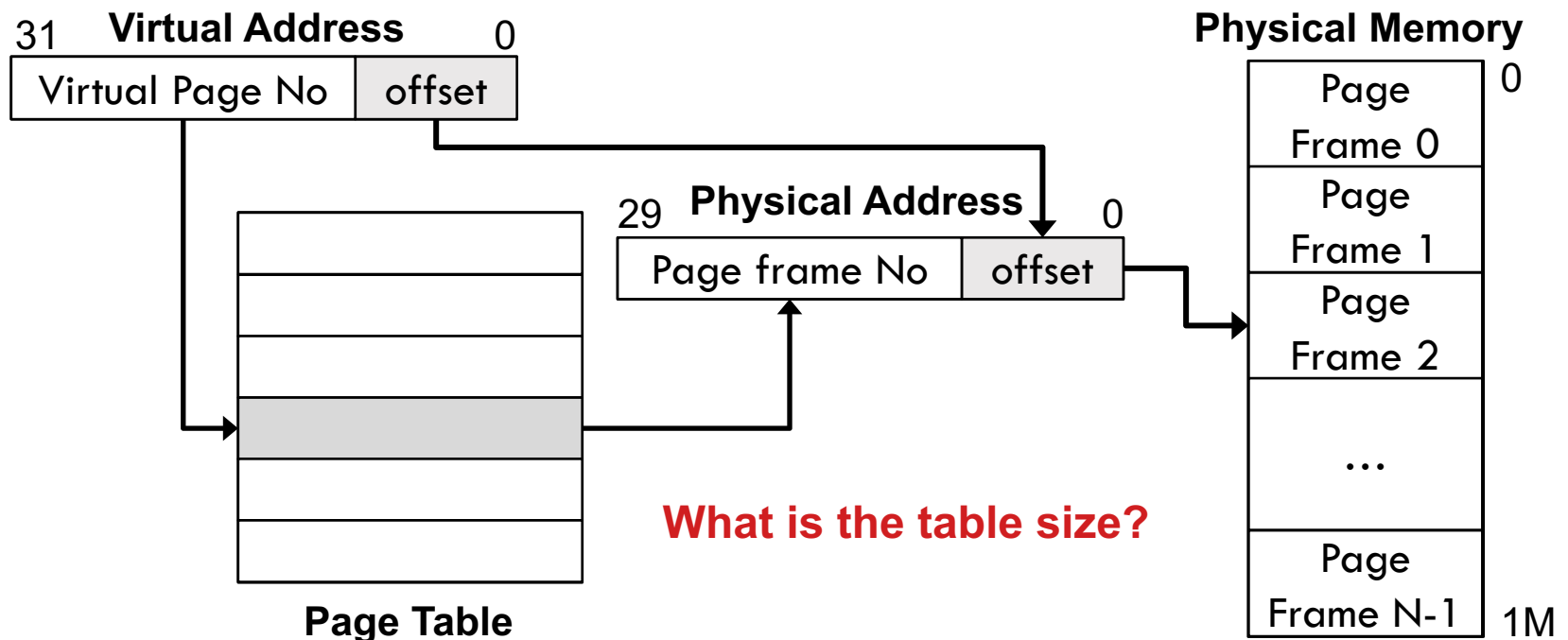
Assistant Professor

School of Computing

University of Utah

# Virtual Address

- Every virtual address is translated to a physical address with the help of hardware
- Data granularity

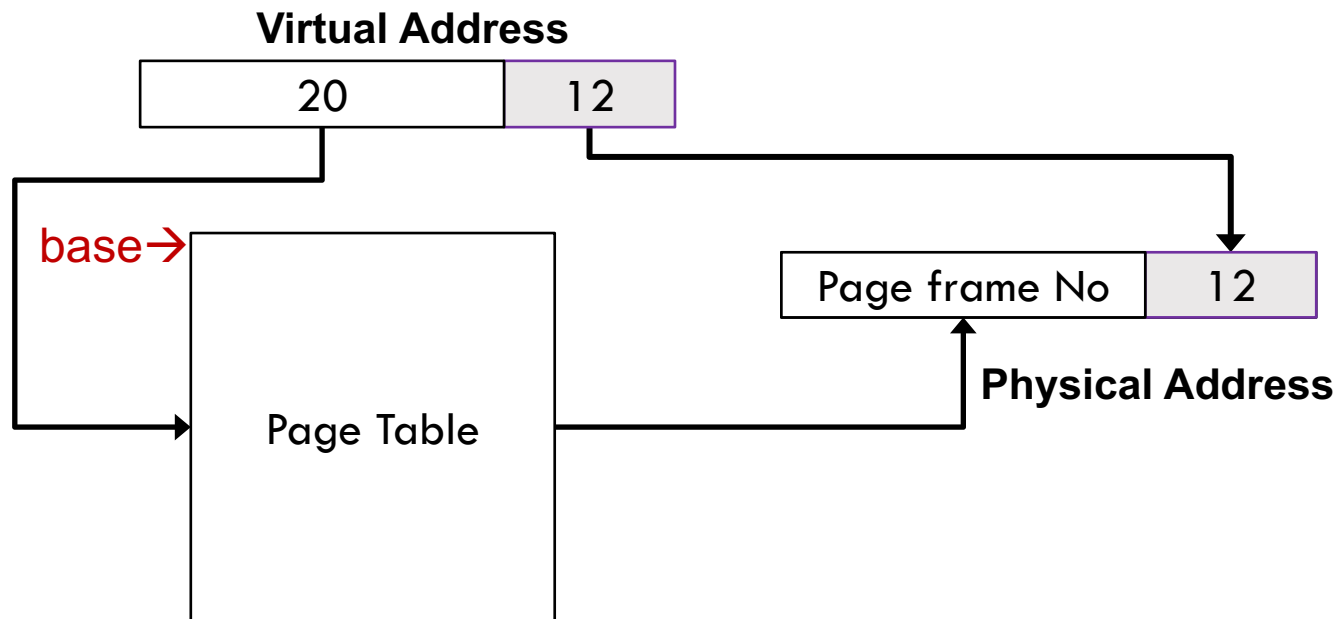


# Address Translation Issues

- Where to store the table?
  - ▣ Too big for on-chip cache
  - ▣ Should be maintained in the main memory
- What to do on a page table miss (page fault)?
  - ▣ No valid frame assigned to the virtual page
  - ▣ OS copies the page from disk to page frame
- What is the cost of address translation?
  - ▣ Additional accesses to main memory per every access
  - ▣ Optimizations?

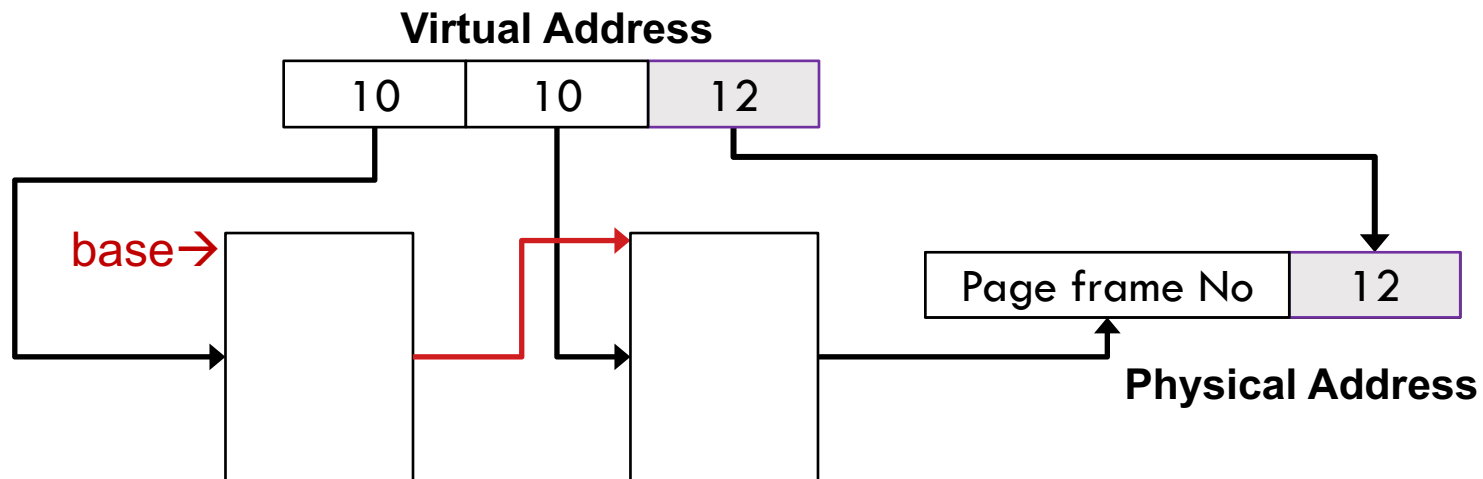
# Address Translation Cost

- Page walk: look up the physical address in the page table
- How many pages to store the page table?



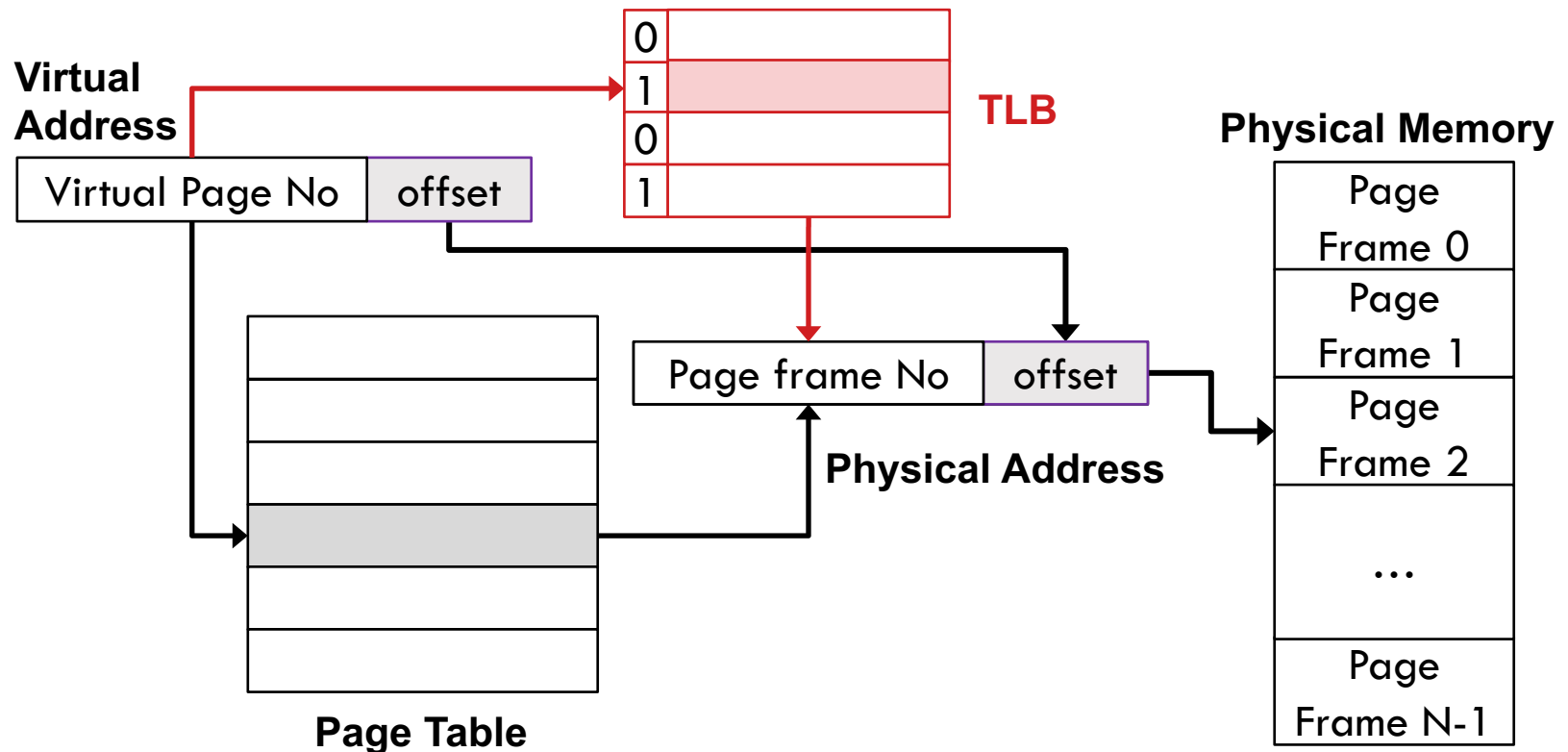
# Multi-Level Page Table

- The virtual (logical) address space is broken down into multiple pages
  - ▣ Example: 4KB pages



# Translation Lookaside Buffer

- Exploit locality to reduce address translation time
  - ▣ Keep the translation in a buffer for future references



# Translation Lookaside Buffer

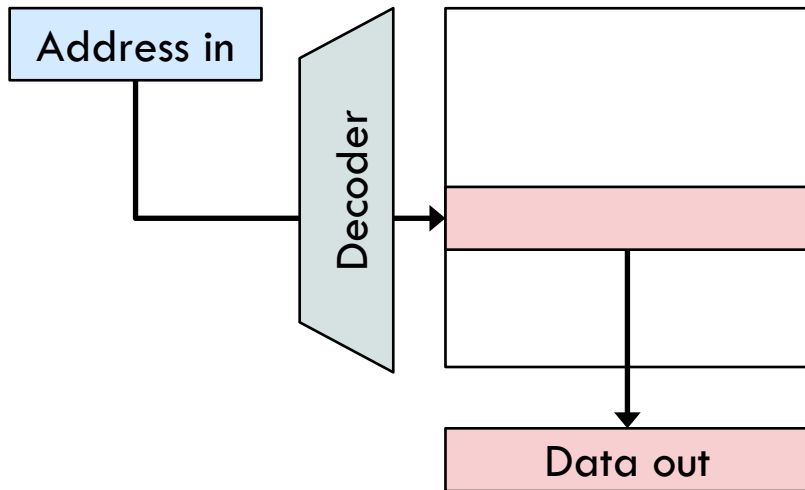
- Just like any other cache, the TLB can be organized as fully associative, set associative, or direct
- TLB access is typically faster than cache access
  - Because TLBs are much smaller than caches
  - TLBs are typically not more than 128 to 256 entries even on high-end machines

| V | Virtual Page # | Physical Page # | Dirty | Status |
|---|----------------|-----------------|-------|--------|
|   |                |                 |       |        |

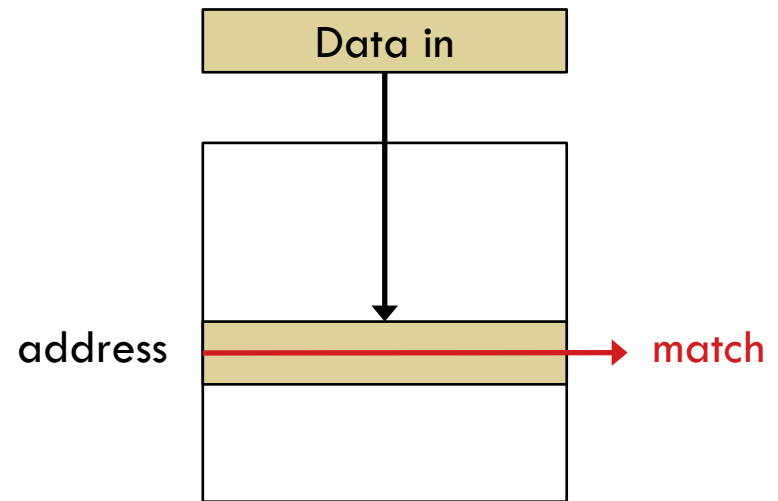
# CAM Based TLB

- Content addressable memory (CAM)
- ▣ Unlike RAM, data in address out

**RAM: Read Operation**



**CAM: Search Operation**

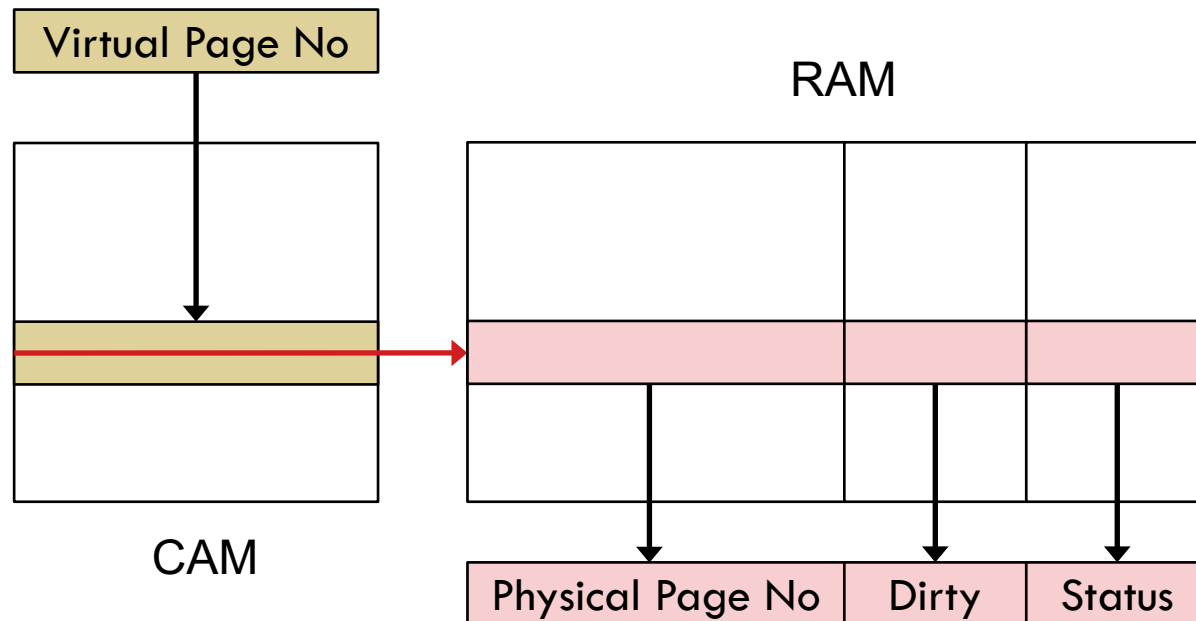




# CAM Based TLB

- Content addressable memory (CAM)
  - ▣ Unlike RAM, data in address out
- CAM based TLB
  - ▣ Both CAM and RAM are used

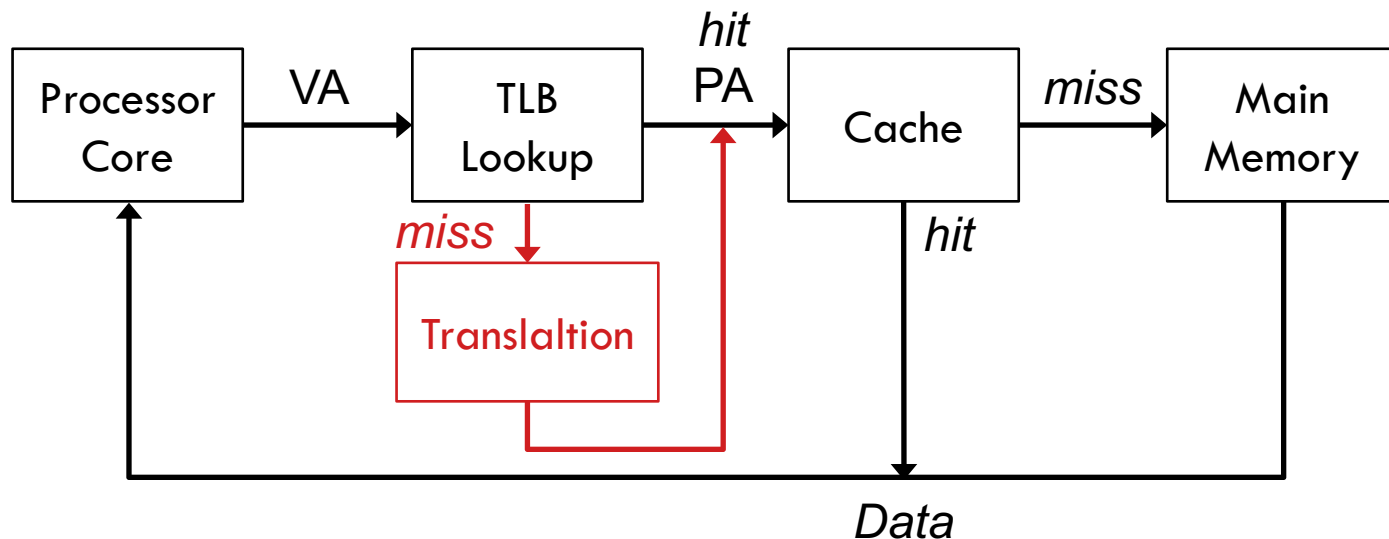
What if multiple rows match?



# TLB in Memory Hierarchy

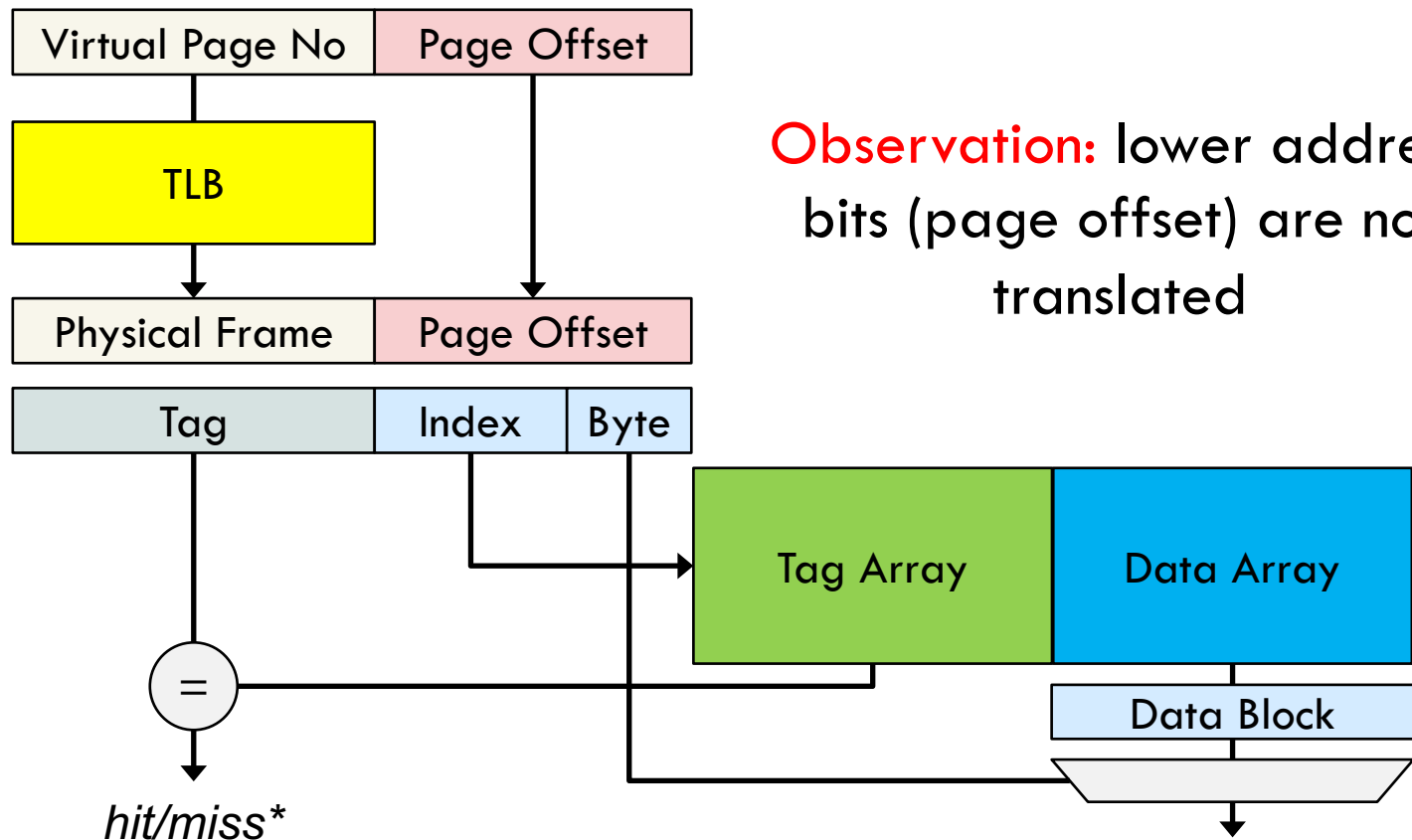
- On a TLB miss, is the page loaded in memory?
  - ▣ Yes: takes 10's cycles to update the TLB
  - ▣ No: page fault
    - Takes 1,000,000's cycles to load the page and update TLB

**Physically indexed, physically tagged: TLB on critical path!**



# Physically Indexed Caches

- **Problem:** increased critical path due to sequential access to TLB and cache



# Virtually Indexed Caches

- **Idea:** Index into cache in parallel with page number translation in TLB

Question for the final exam:  
what if the page offset is not  
equal to index+byte?

