Lecture 21
Virtual Memory

Suggested reading:

(HP Chapter 5.4-5.5)

RELATIVE SIZES

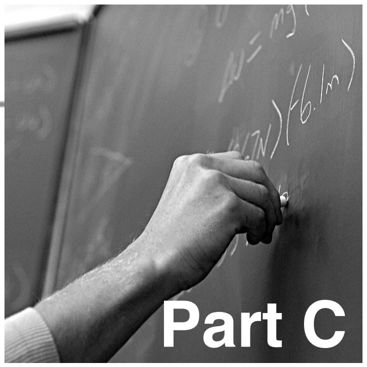
Page Table Size

page table size

- example #1: 32-bit VA, 4KB pages, 4-byte PTE
  - 1M pages (32 bits = 4 GB address space / 4 KB page = 1M pages)
  - 1M pages*4bytes = 4MB page table (bad, but could be worse)
- example #2: 64-bit VA, 4KB pages, 4-byte PTE
  - 4P pages, 16PB page table (not a viable option)
- upshot: can't have page tables of this size in memory

techniques for reducing page table size

- multi-level page tables
- inverted page tables



Block replacement

- Which block should be replaced on a virtual memory miss?
  - Again, we'll stick with the strategy that it's a good thing to eliminate page faults
  - Therefore, we want to replace the LRU block
    - Many machines use a "use" or "reference" bit
    - Periodically reset
    - Gives the OS an estimation of which pages are referenced

Writing a block

- What happens on a write?
  - We don't even want to think about a write through policy!
    - Time with accesses, VM, hard disk, etc. is so great that this is not practical
  - Instead, a write back policy is used with a dirty bit to tell if a block has been written

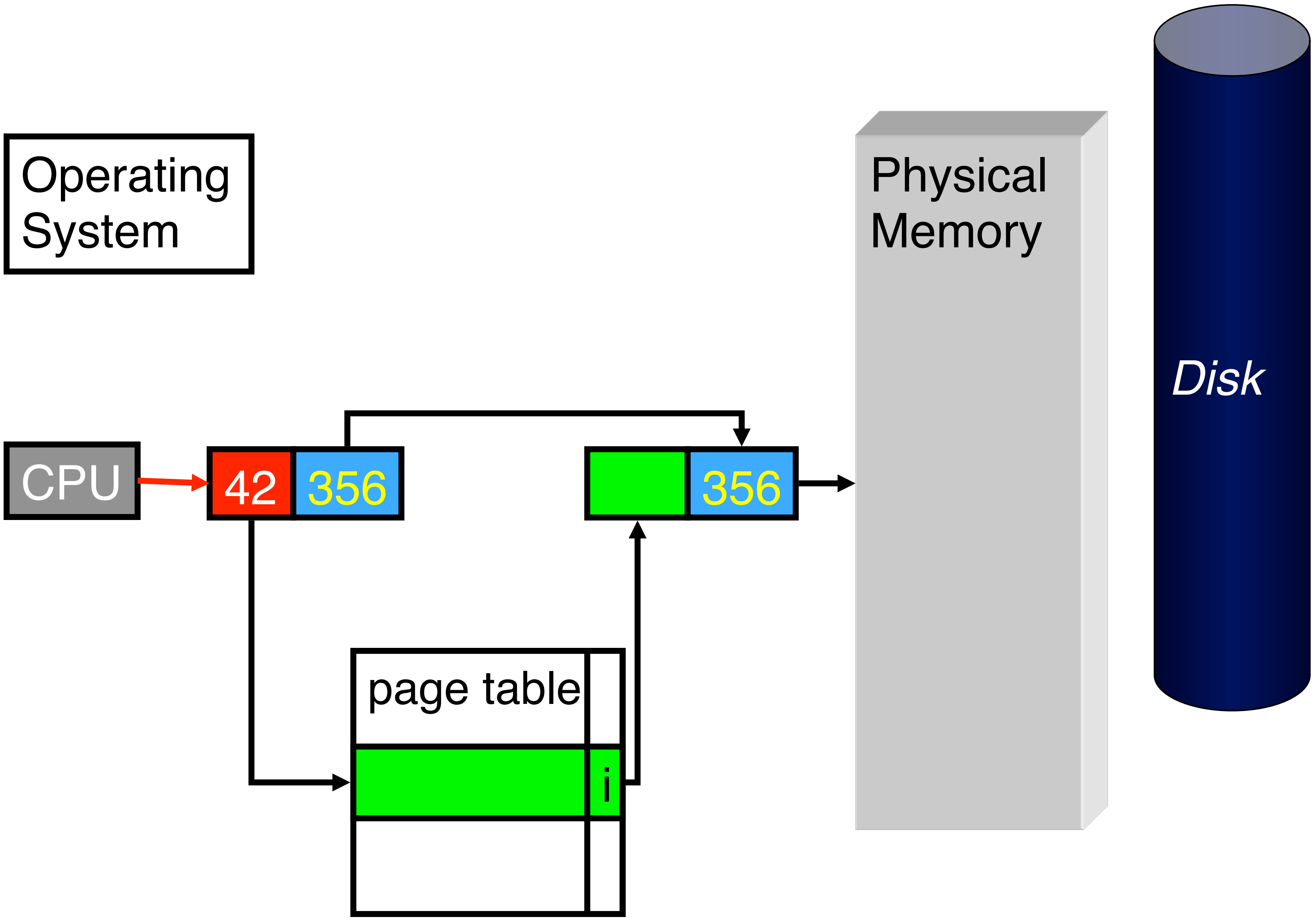
INTRODUCTION TO TLBS

Page tables and lookups…

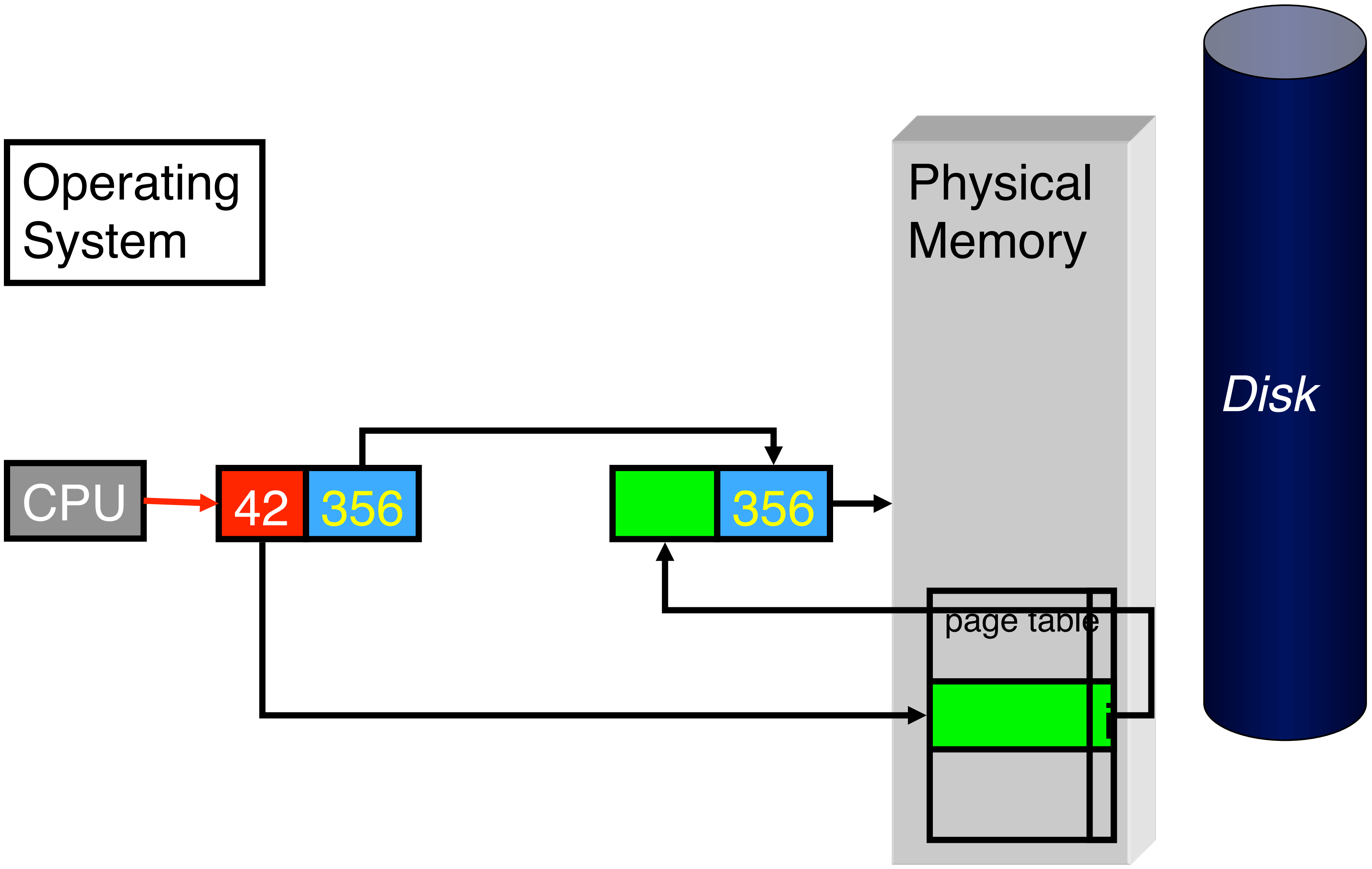
- 1. Its slow! Weve turned every access to memory into two accesses to memory
  - solution: add a specialized “cache” called a “translation lookaside buffer (TLB)” inside the processor

- 2. its still huge!
  - even worse: we’re ultimately going to have a page table for every *process*. Suppose 1024 processes, that’s 4GB of page tables!

Paging/VM

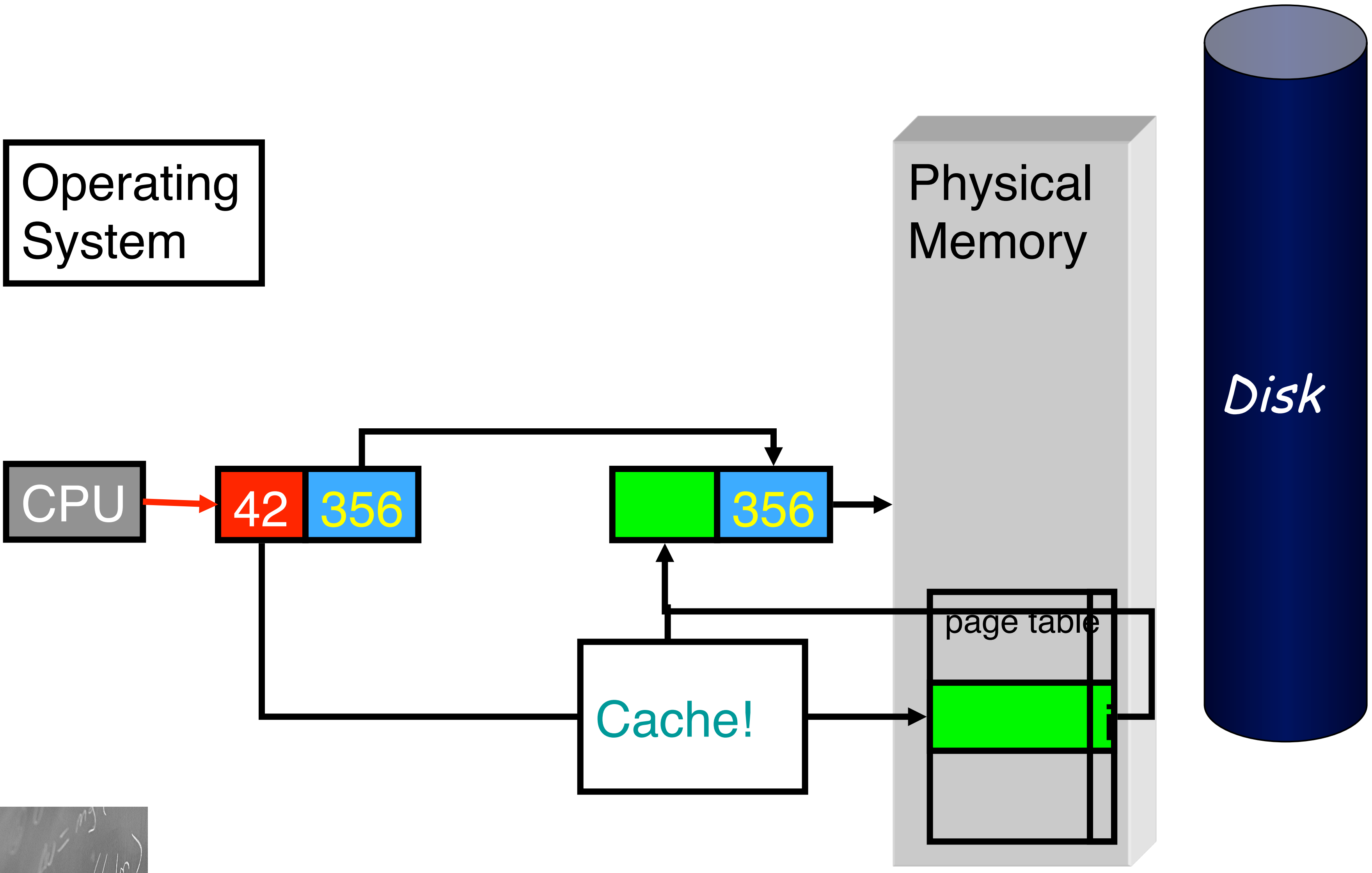

Operating System
Physical Memory
Disk
CPU
42
356
356
page table
i

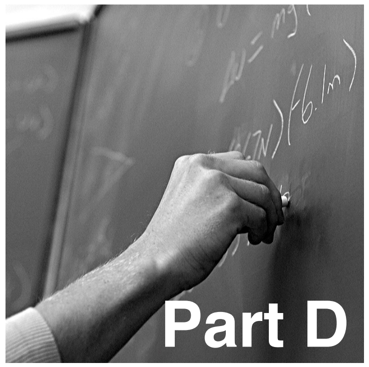
Paging/VM



Place page table in physical memory
However: this doubles the time per memory access!!

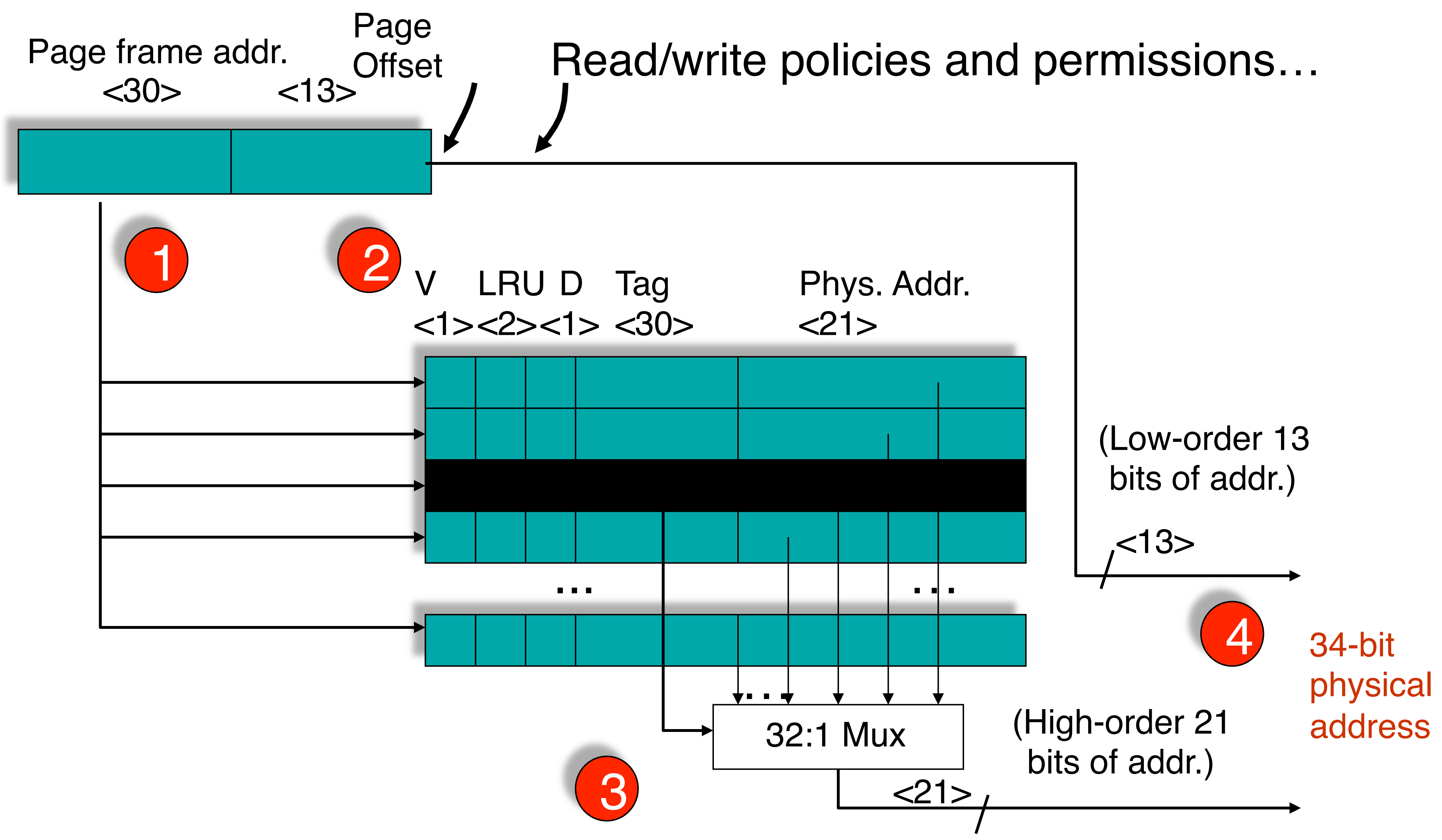
Paging/VM





Special-purpose cache for translations
Historically called the TLB: Translation Lookaside Buffer

An example of a TLB



Review: Translation Cache

A way to speed up translation is to use a special cache of recently used page table entries -- this has many names, but the most frequently used is *Translation Lookaside Buffer* or *TLB*

Virtual Page #	Physical Frame #	Dirty	Ref	Valid	Access

$\underbrace{\hspace{6em}}_{\text{tag}}$

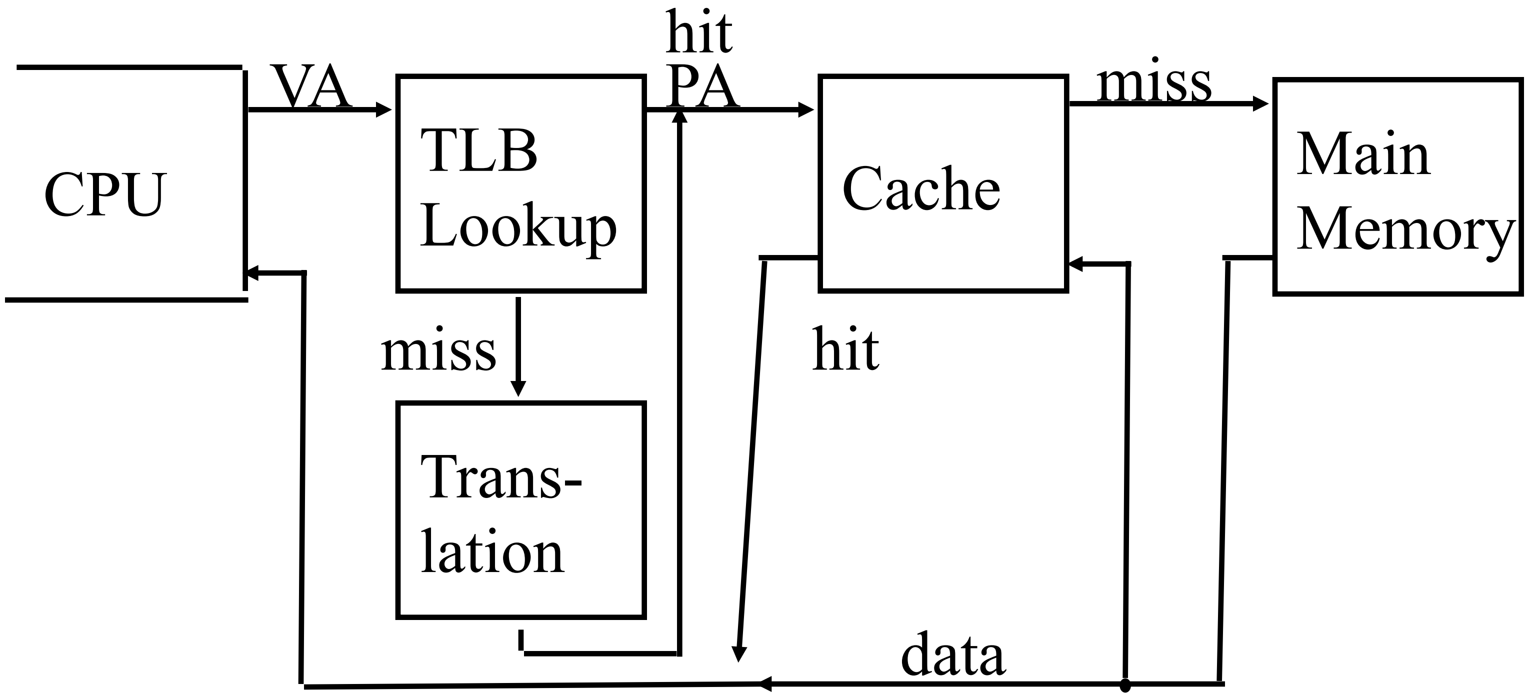
Really just a cache (a special-purpose cache) on the page table mappings

TLB access time comparable to cache access time
(much less than main memory access time)

Review: Translation Cache

Just like any other cache, the TLB can be organized as fully associative, set associative, or direct mapped

TLBs are usually small, typically not more than 128 - 256 entries even on high end machines. This permits fully associative lookup on these machines. Most mid-range machines use small n-way set associative organizations.



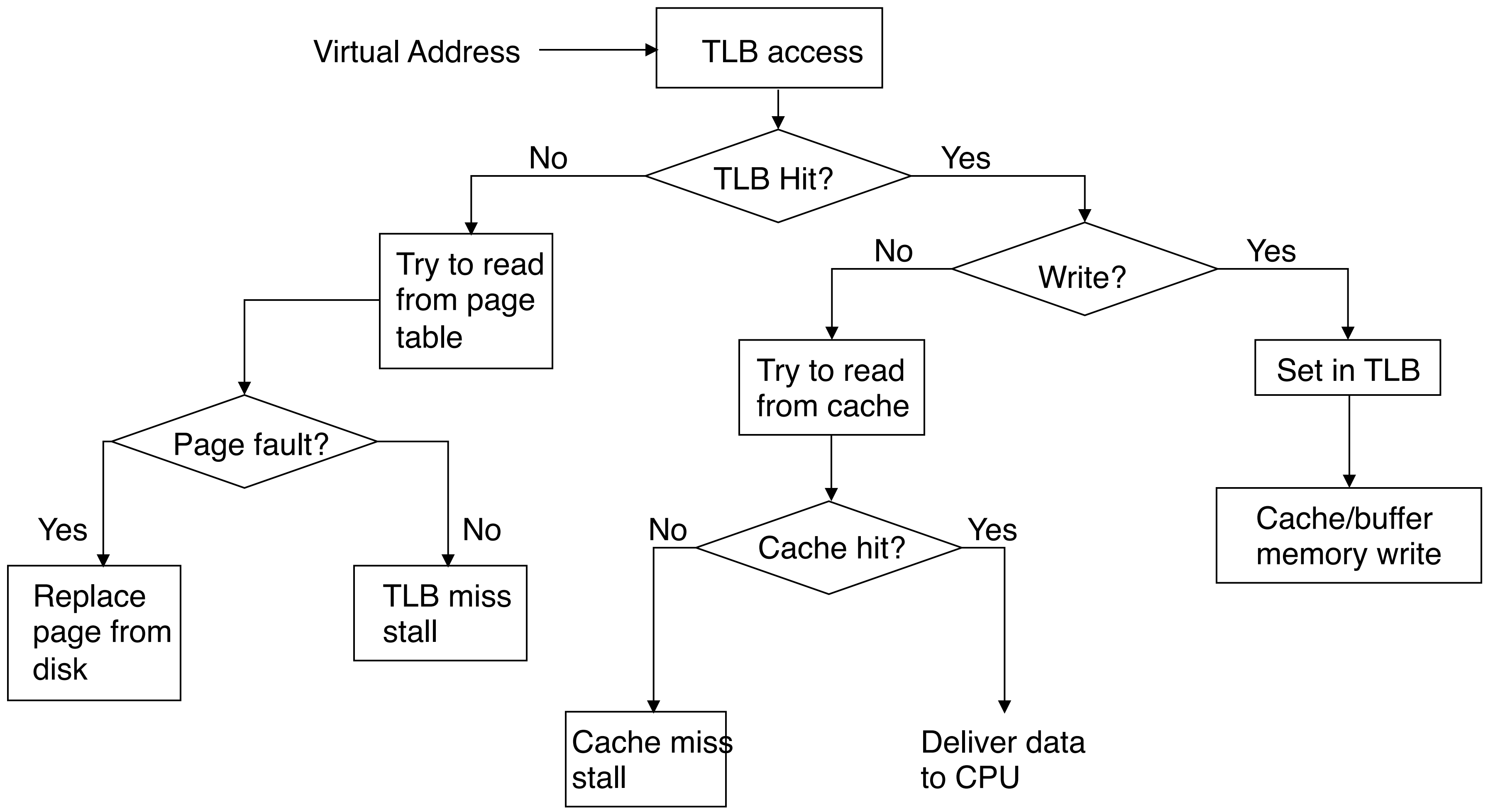
Translation with a TLB

A FULL ADDRESS TRANSLATION

The “big picture” and TLBs

- Address translation is usually on the critical path…
  - …which determines the clock cycle time of the μP
- Even in the simplest cache, TLB values must be read and compared

The "big picture" and TLBs



Examples