IOWA STATE UNIVERSITY
Department of Electrical and Computer Engineering

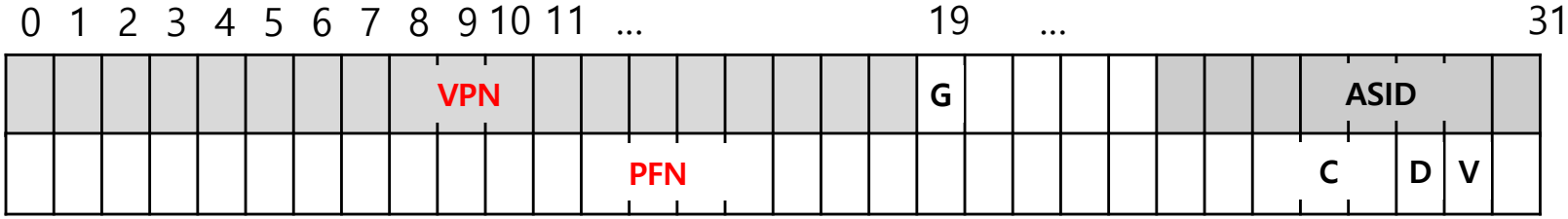
Lecture 23: Swapping

Agenda

- Recap
- Multi-Level Page Tables (cont')
- Swapping

Recap

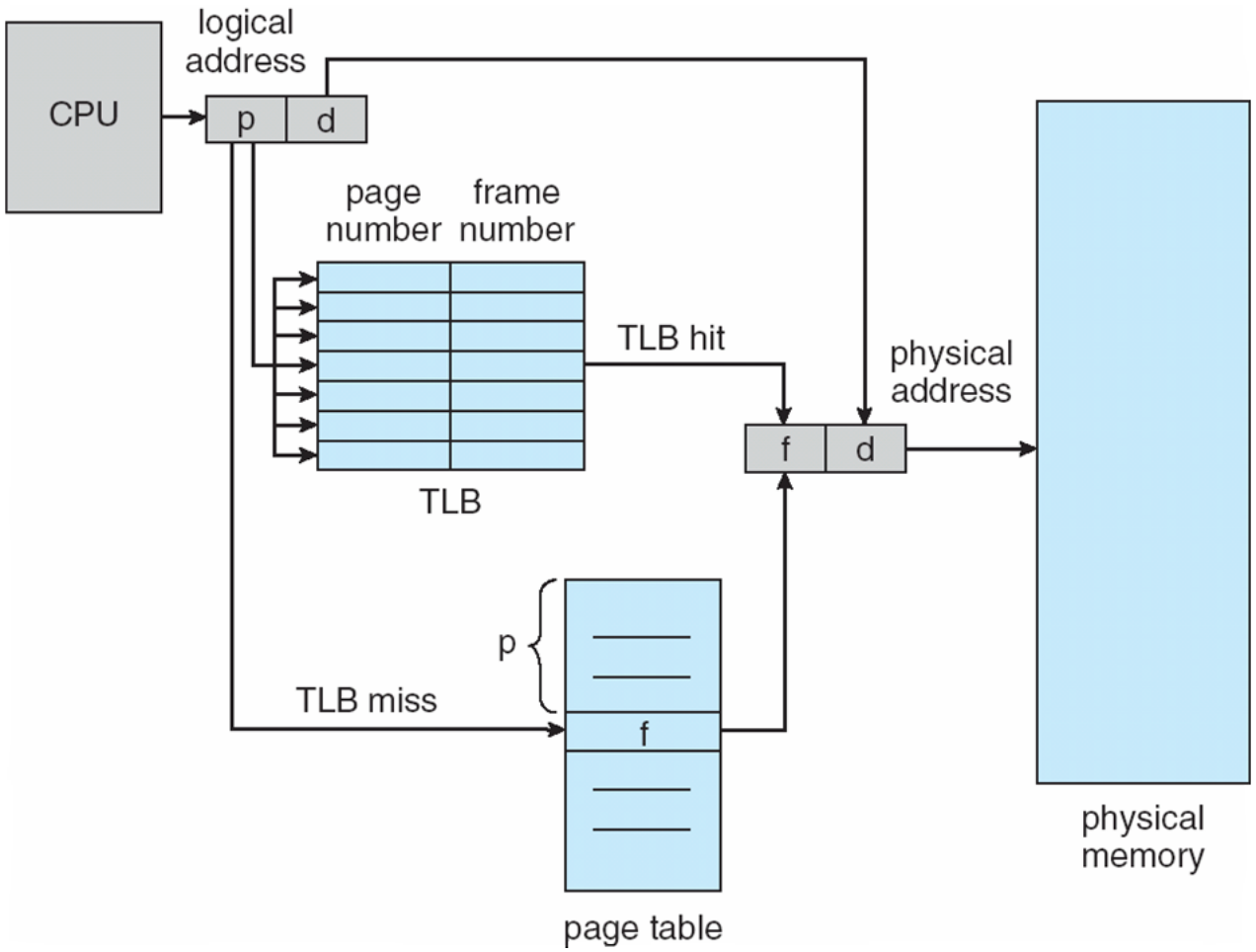
- Translation Lookaside Buffer (TLB)
  - Hardware cache for speeding up paging
  - A real TLB entry (MIPS)



Flag	Content
19-bit VPN	The rest reserved for the kernel.
24-bit PFN	**Systems can support with up to 64GB of main memory($2^{24} * 4KB$ pages).**
Global bit(G)	Used for pages that are globally-shared among processes.
ASID	**OS can use to distinguish between address spaces.**
Coherence bit(C)	determine how a page is cached by the hardware.
Dirty bit(D)	marking when the page has been written.
Valid bit(V)	tells the hardware if there is a valid translation present in the entry.

Recap

- Address translation with TLB & page table



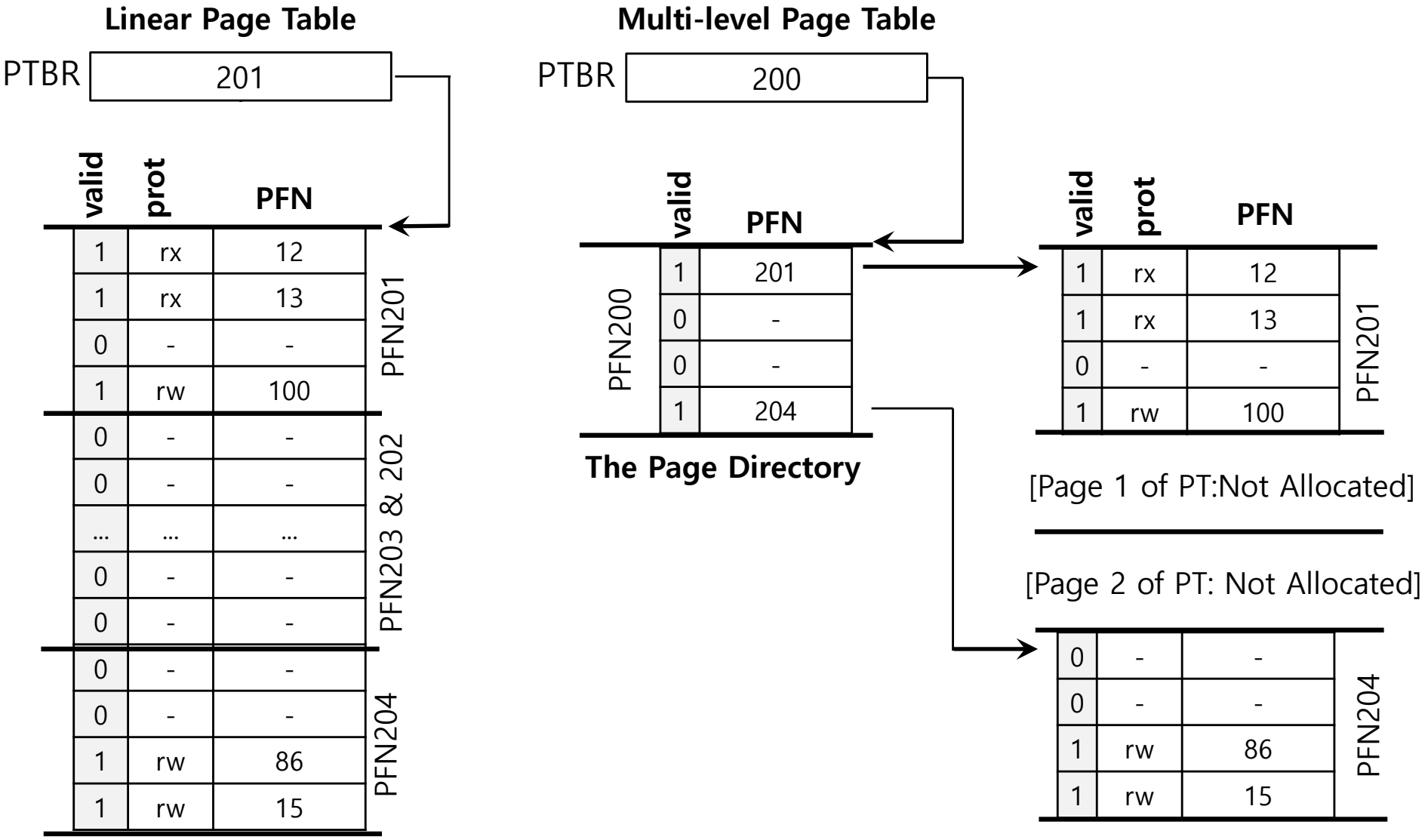
Recap

- TLB may improve performance greatly
  - Effective Access Time (EAT)
    - effective time for accessing memory with TLB
    - TLB Hit ratio = α
      - percentage of times that a page number is found in the TLB
      - TLB hit: one memory access
      - TLB miss: two memory accesses
    - Consider α = 80%, 100ns for each memory access
      - EAT = 0.80 x 100 + 0.20 x 200 = 120ns
    - Consider a more realistic hit ratio α = 99%; still 100ns for each memory access
      - EAT = 0.99 x 100 + 0.01 x 200 = 101ns
  - Temporal & spatial locality

Recap

- Multi-level page table
  - a linear address space is large, but many pages are not used
    - a linear page table often has many empty entries
  - split the page table into page-sized units
    - If an entire page of page-table entries is invalid, don't allocate that page of the page table at all
  - To track whether a page of the page table is valid, use a new structure, called page directory
    - It consists of a number of page directory entries (PDE)
      - one PDE per page of the page table

Recap

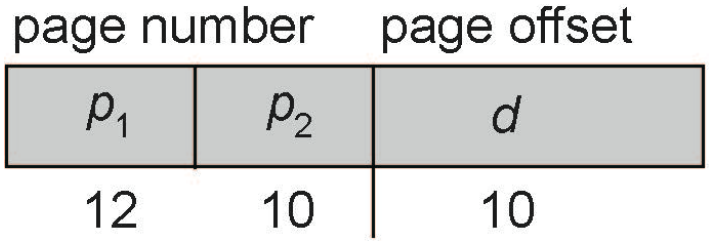

Linear Page Table
PTBR
201
valid
prot
PFN
1 rx 12
1 rx 13
0 - -
1 rw 100
PFN201
0 - -
0 - -
... ...
0 - -
0 - -
PFN203 & 202
0 - -
0 - -
1 rw 86
1 rw 15
PFN204
Multi-level Page Table
PTBR
200
valid
PFN
PFN200
1 201
0 -
0 -
1 204
The Page Directory
valid
prot
PFN
1 rx 12
1 rx 13
0 - -
1 rw 100
PFN201
[Page 1 of PT:Not Allocated]
[Page 2 of PT: Not Allocated]
0 - -
0 - -
1 rw 86
1 rw 15
PFN204

Agenda

- ~~Recap~~
- Multi-Level Page Tables (cont')
- Swapping

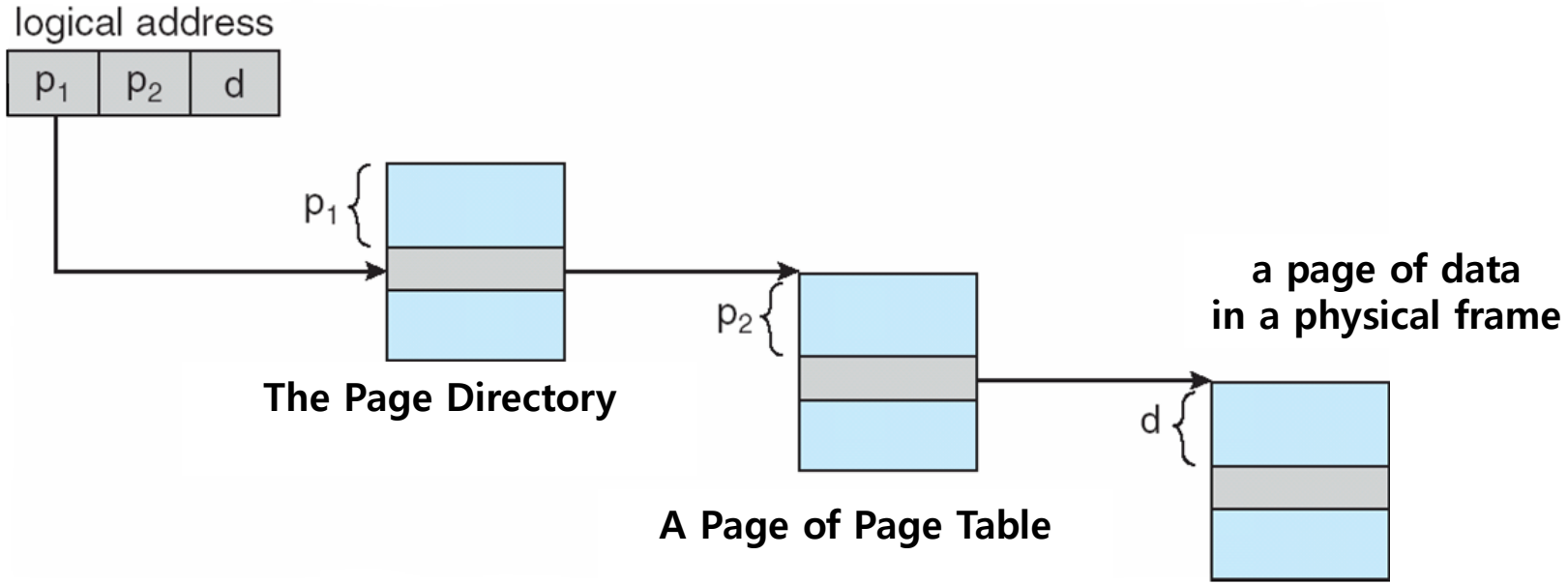
Multi-Level Page Tables (cont')

- Address Translation Example
  - A 32-bit virtual address (with 1KB page size) is divided into:
    - a page number consisting of 22 bits
    - a page offset consisting of 10 bits
  - Since the page table is paged, the page number is further divided into:
    - a 12-bit page directory index (p1)
    - a 10-bit page table index (p2)



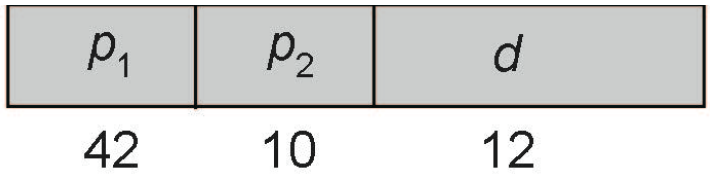
Multi-Level Page Tables (cont')

- Address Translation Example
  - the page directory index (p1) is used to identify a page directory entry (PDE) in the page directory
  - the page table index (p2) is used to identify a page table entry (PTE) in a page of page table



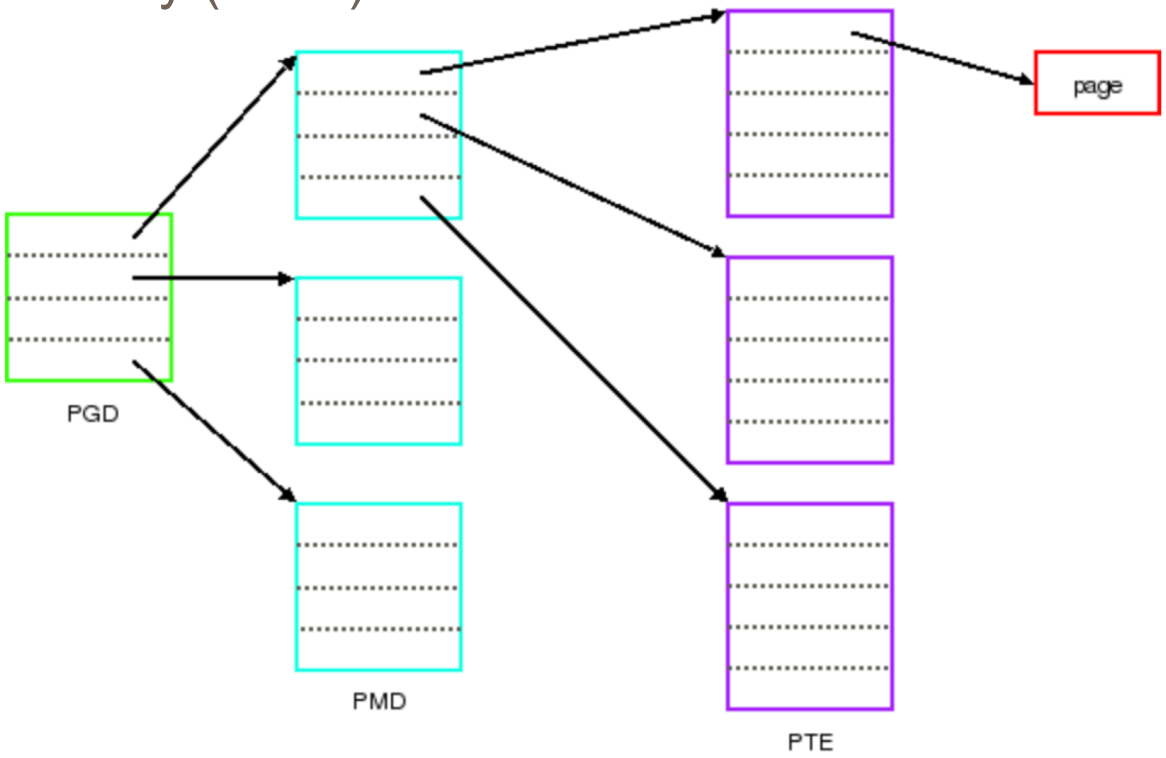
Multi-Level Page Tables (cont')

- Two levels may not be sufficient
  - E.g., assume a 64-bit address space
    - If page size is 4 KB (2^{12})
      - then page table has 2^{52} entries
    - In a two-level page table
      - a page (4KB) of page table could be 2^{10} 4-byte entries
        - 10 bits for indexing into the page of page table
      - the page directory has 2^{42} 4-byte entries (2^{44} bytes)
        - the page directory itself is huge!



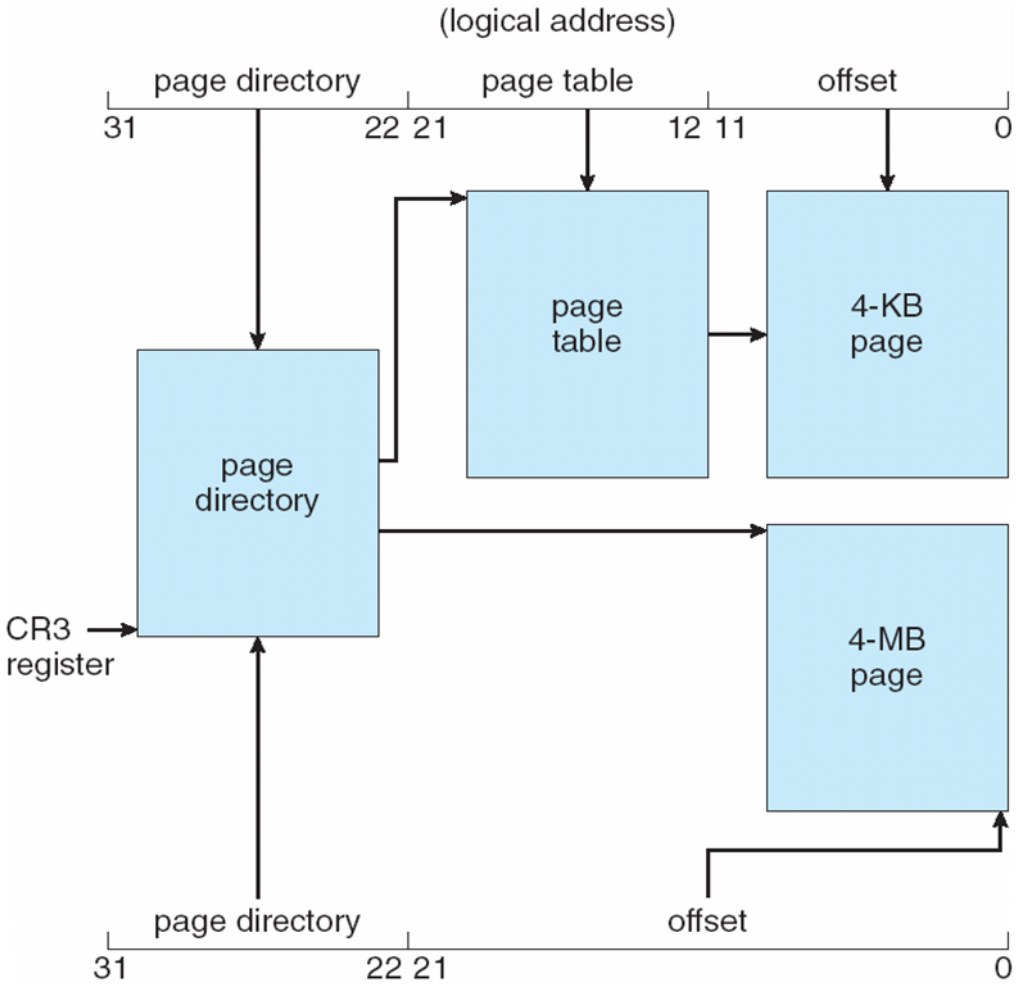
Multi-Level Page Tables (cont')

- Introducing more levels
  - E.g., a three-level page table
    - page global directory (PGD)
    - page middle directory (PMD)
    - page table



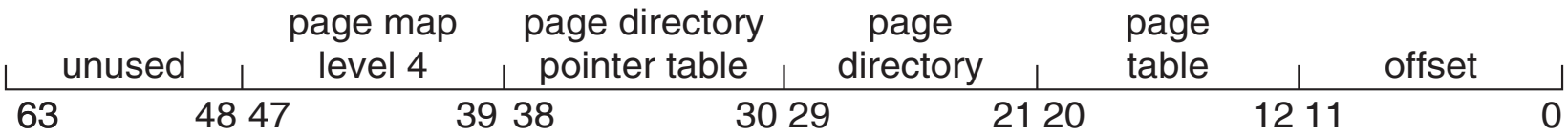
Multi-Level Page Tables (cont')

- Intel IA-32 Paging Architecture



Multi-Level Page Tables (cont')

- Intel x86-64
  - 64 bits is ginormous (> 16 exabytes)
  - In practice only implement 48 bit addressing
    - Page sizes of 4 KB, 2 MB, 1 GB
    - Four levels of paging hierarchy
  - Can use page address extension (PAE) to extend physical address bits



Multi-Level Page Tables

- Advantages
  - Only allocates page-table space in proportion to the amount of address space you are using.
  - The OS can grab the next free page when it needs to allocate or grow a page table.

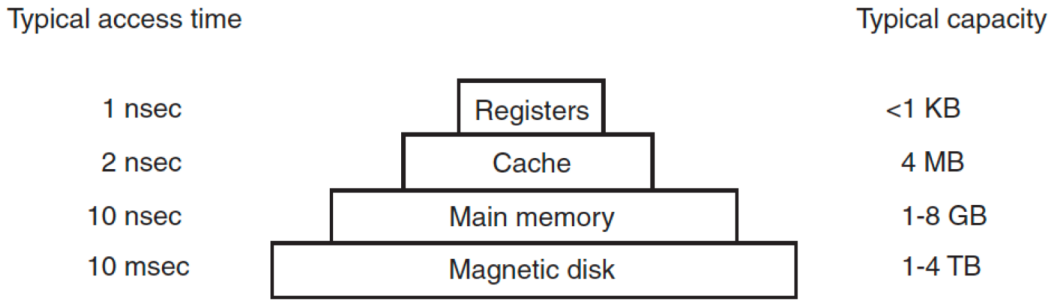
- Disadvantages
  - may be slower for some memory accesses
    - time-space tradeoff
  - complexity

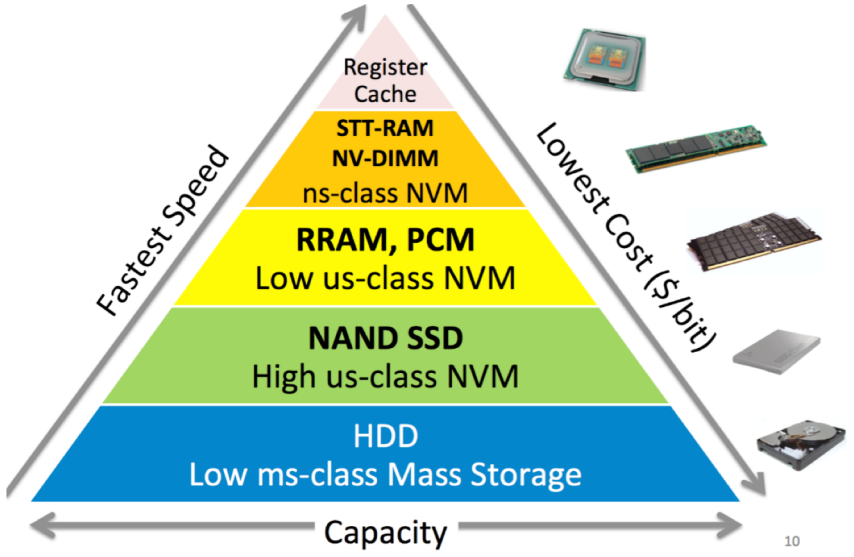
Agenda

- ~~Recap~~
- ~~Multi-Level Page Tables (cont')~~
- Swapping

Swapping

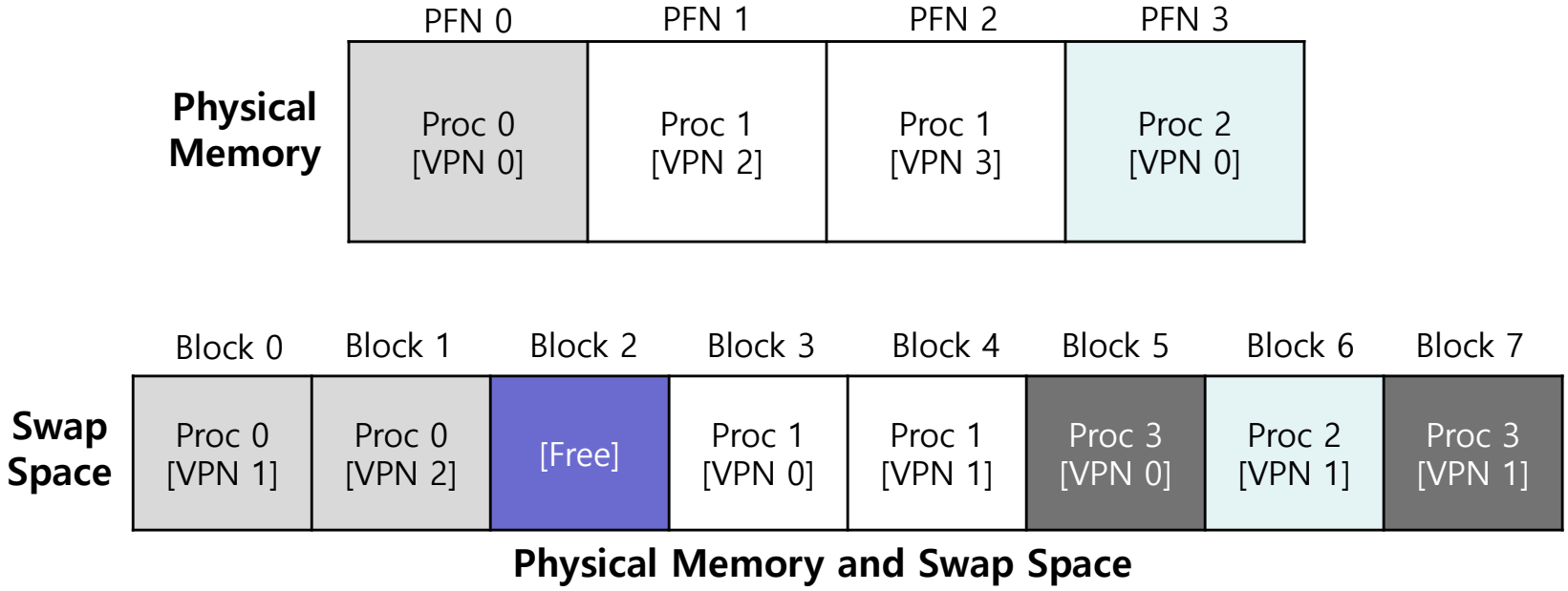
- Extend physical memory
  - stash away portions of address space that currently aren't in great demand
  - the unpopular pages are placed in the **swap space** at the next layer of memory/storage hierarchy
    - e.g., hard disk drives (HDDs)





Swapping

- Swap Space
  - Reserved space on the disk for moving pages back and forth
  - OS manages the swap space, in page-sized unit



Physical Memory and Swap Space

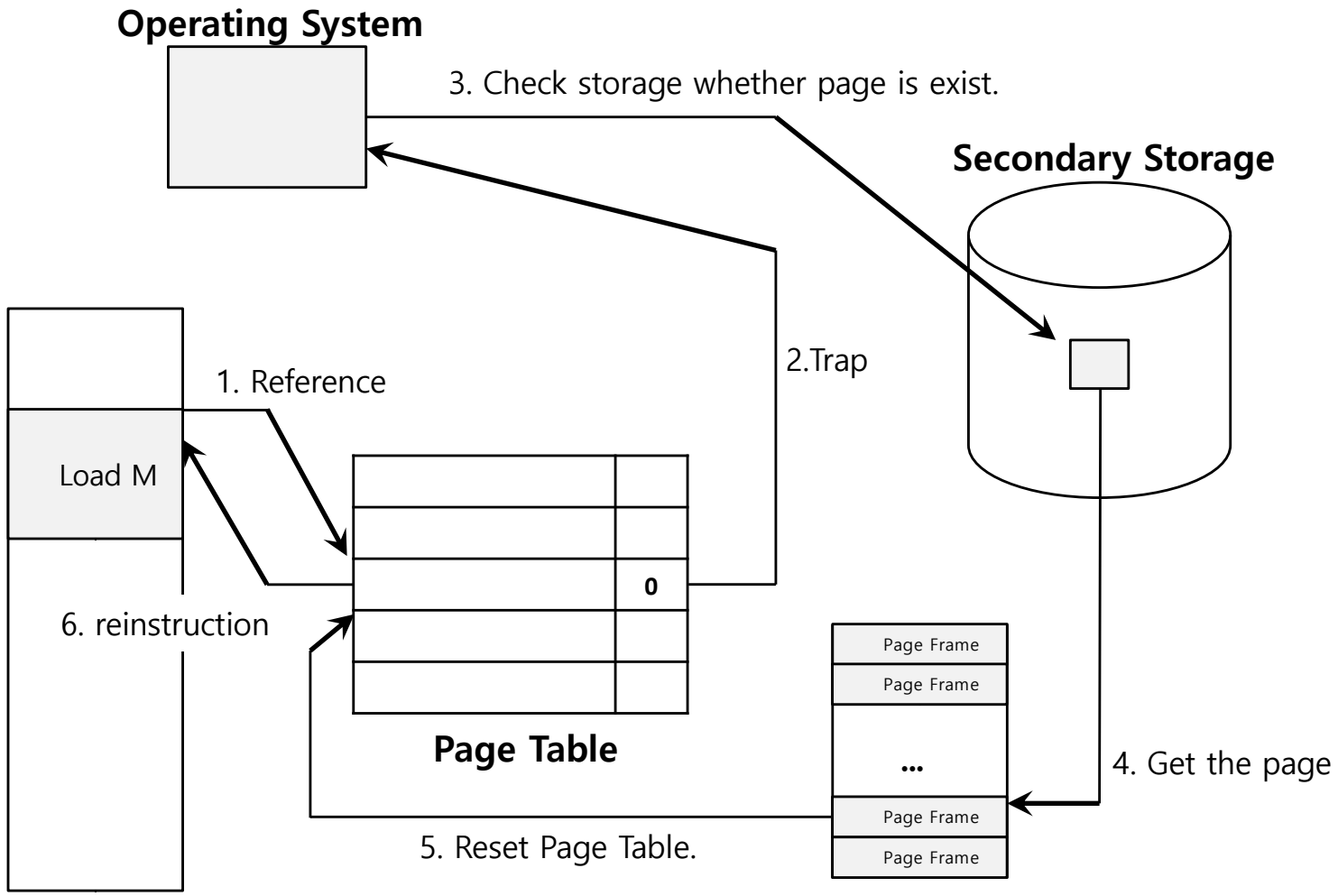
Swapping

- Present Bit
  - One bit in the page table entry (PTE) to support swapping
    - When the hardware checks the PTE, it may find that the page is not present in physical memory

Value	Meaning
1	page is present in physical memory
0	The page is not in memory but rather on disk.

Accessing a page not in physical memory

- Page fault handler swap in a page from disk



Agenda

- ~~Recap~~
- ~~Multi-Level Page Tables (cont')~~
- ~~Swapping~~

Questions?

*acknowledgement: slides include content from "Modern Operating Systems" by A. Tanenbaum, "Operating Systems Concepts" by A. Silberschatz etc., "Operating Systems: Three Easy Pieces" by R. Arpaci-Dusseau etc., and anonymous pictures from internet.