CMCS 411
Computer Architecture

Lecture 23

Virtual Memory

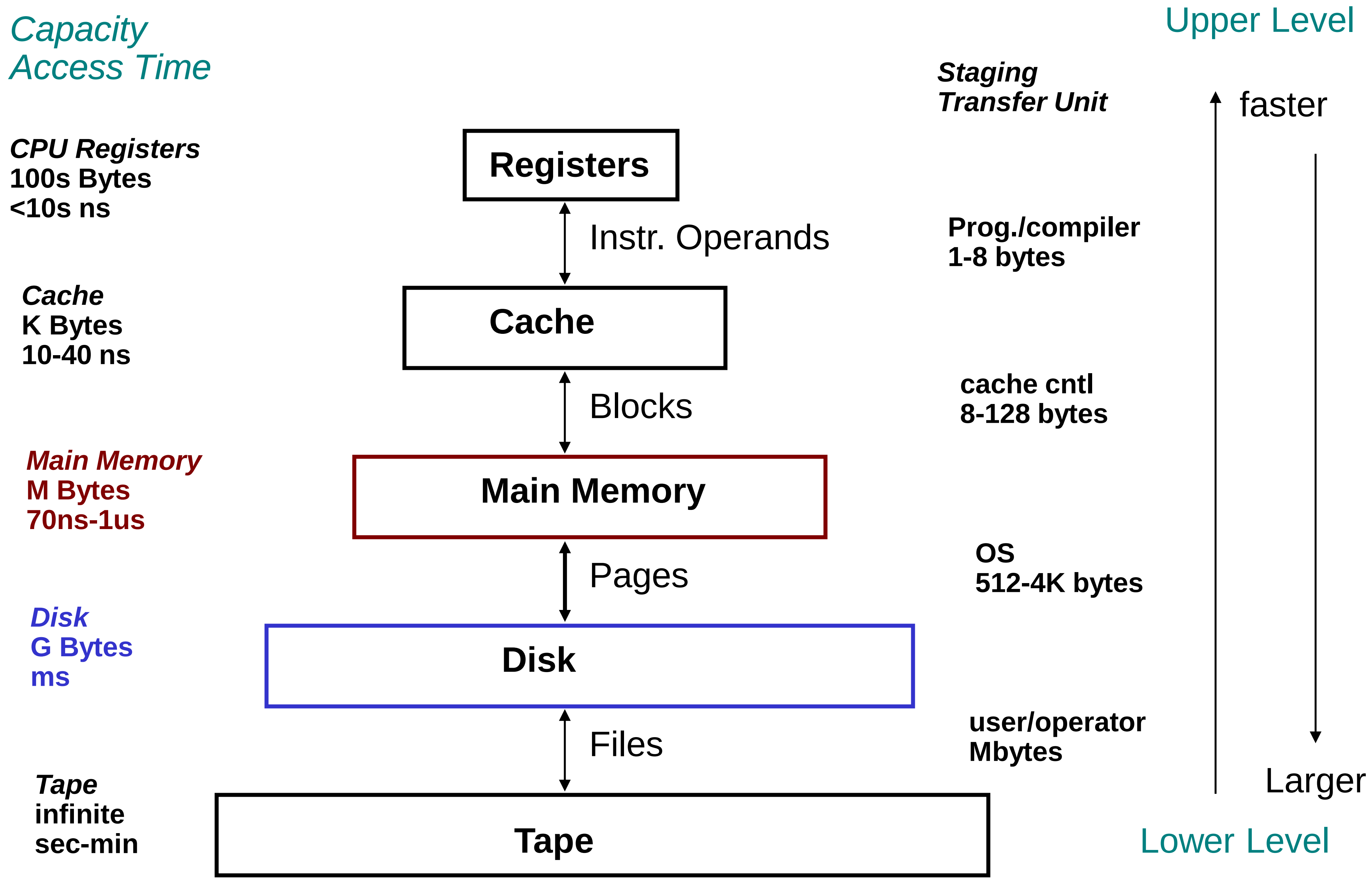
November 14, 2017

www.csee.umbc.edu/~younis/CMPE411/CMPE411.htm

Lecture's Overview

- ***Previous Lecture:***
  - Organization of main memory
    - ➔ Main memory performance issues
    - ➔ Memory interleaving
  - Measuring and improving cache performance
    - ➔ Relationship between computer performance and cache
    - ➔ Factors that affect cache performance
    - ➔ Optimization techniques for cache performance
  - Multi-level caches
    - ➔ N-way set associative cache design
    - ➔ Performance set of associate cache memory
- ***This Lecture:***
  - Virtual Memory
  - Integration of cache and virtual memory

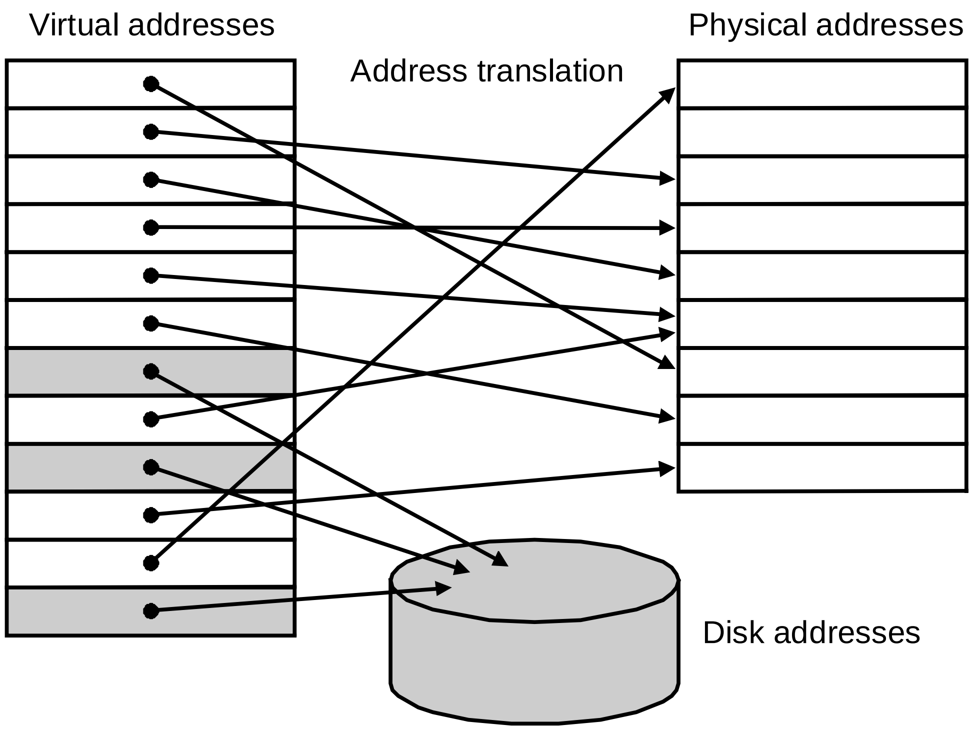
Memory Hierarchy



* Slide is courtesy of Dave Patterson

Virtual Memory

- ❑ Using virtual addressing, main memory plays the role of cache for disks
- ❑ The virtual space is much larger than the physical memory space
- ❑ Physical main memory contains only the active portion of the virtual space
- ❑ Address space can be divided into fixed size (pages) or variable size (segments) blocks



<u>Advantages</u>

➔ Allows efficient and safe data sharing of memory among multiple programs

➔ Moves programming burdens of a small, limited amount of main memory

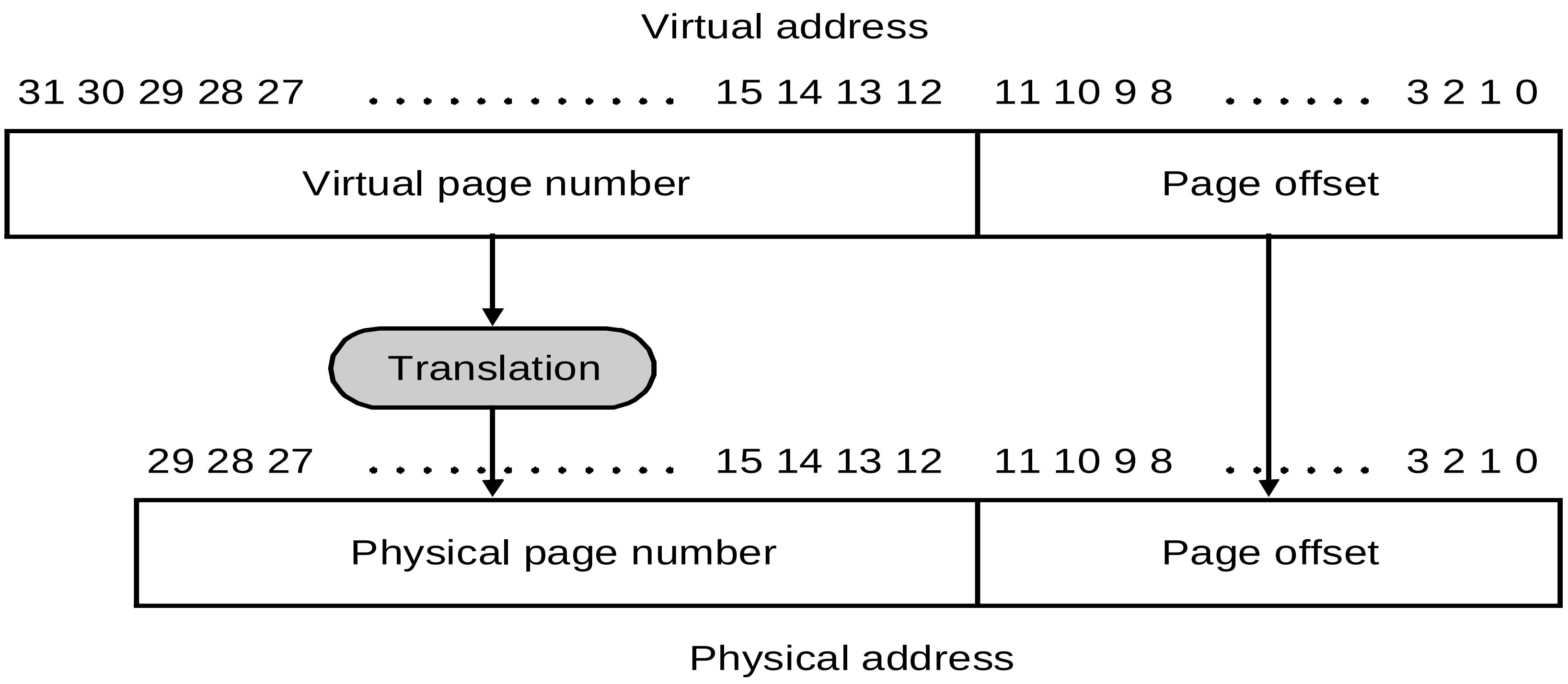
➔ Simplifies program loading and avoid the need for contiguous memory block

➔ allows programs to be loaded at any physical memory location

Cache		***Virtual memory***
Block	⇒	Page
Cache miss	⇒	page fault
Block addressing	⇒	Address translation

Virtual Addressing

- Page faults are costly and take millions of cycles to process (disks are slow)
- Optimization Strategies:
  - ➔ Pages should be large enough to amortize the high access time
  - ➔ Fully associative placement of pages reduces page fault rate
  - ➔ Software-based handling of page faults allows using clever page placement
  - ➔ Write-through technique can make writing very time consuming (use copy back)



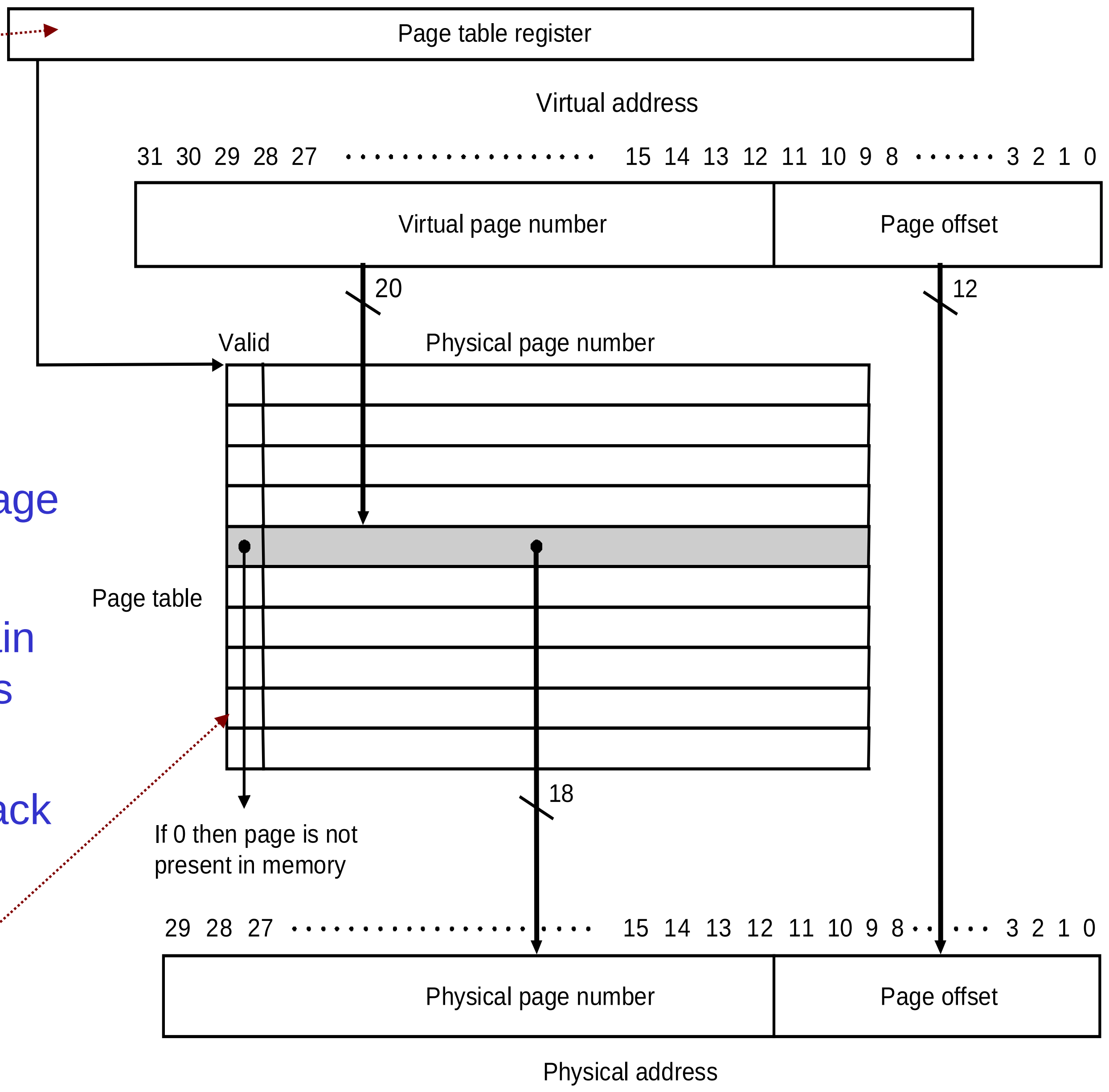
Page Table

Hardware supported

Page table:

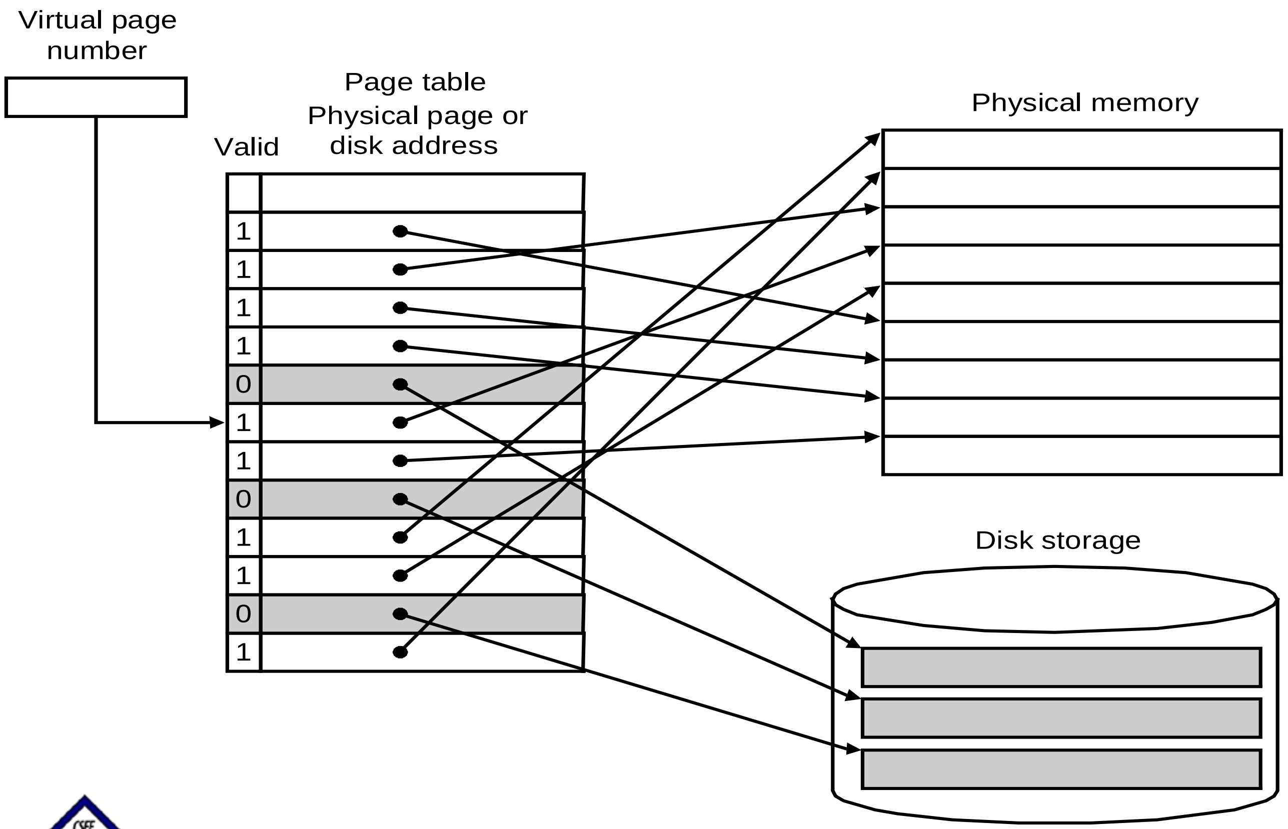
- Resides in main memory
- One entry per virtual page
- No tag is required since it covers all virtual pages
- Point directly to physical page
- Table can be very large
- Operating sys. may maintain one page table per process
- A dirty bit is used to track modified pages for copy back

Indicates whether the virtual page is in main memory or not



Page Faults

- A page fault happens when the valid bit of a virtual page is off
- A page fault generates an exception to be handled by the operating system to bring the page to main memory from a disk
- The operating system creates space for all pages on disk and keeps track of the location of pages in main memory and disk
- Page location on disk can be stored in page table or in an auxiliary structure



- LRU page replacement strategy is the most commonly used
- Simplest LRU implementation uses a reference bit per page and periodically reset reference bits

Optimizing Page Table Size

With a 32-bit virtual address, 4-KB pages, and 4 bytes per page table entry:

$$\text{Number of page table entries} = \frac{2^{32}}{2^{12}} = 2^{20}$$

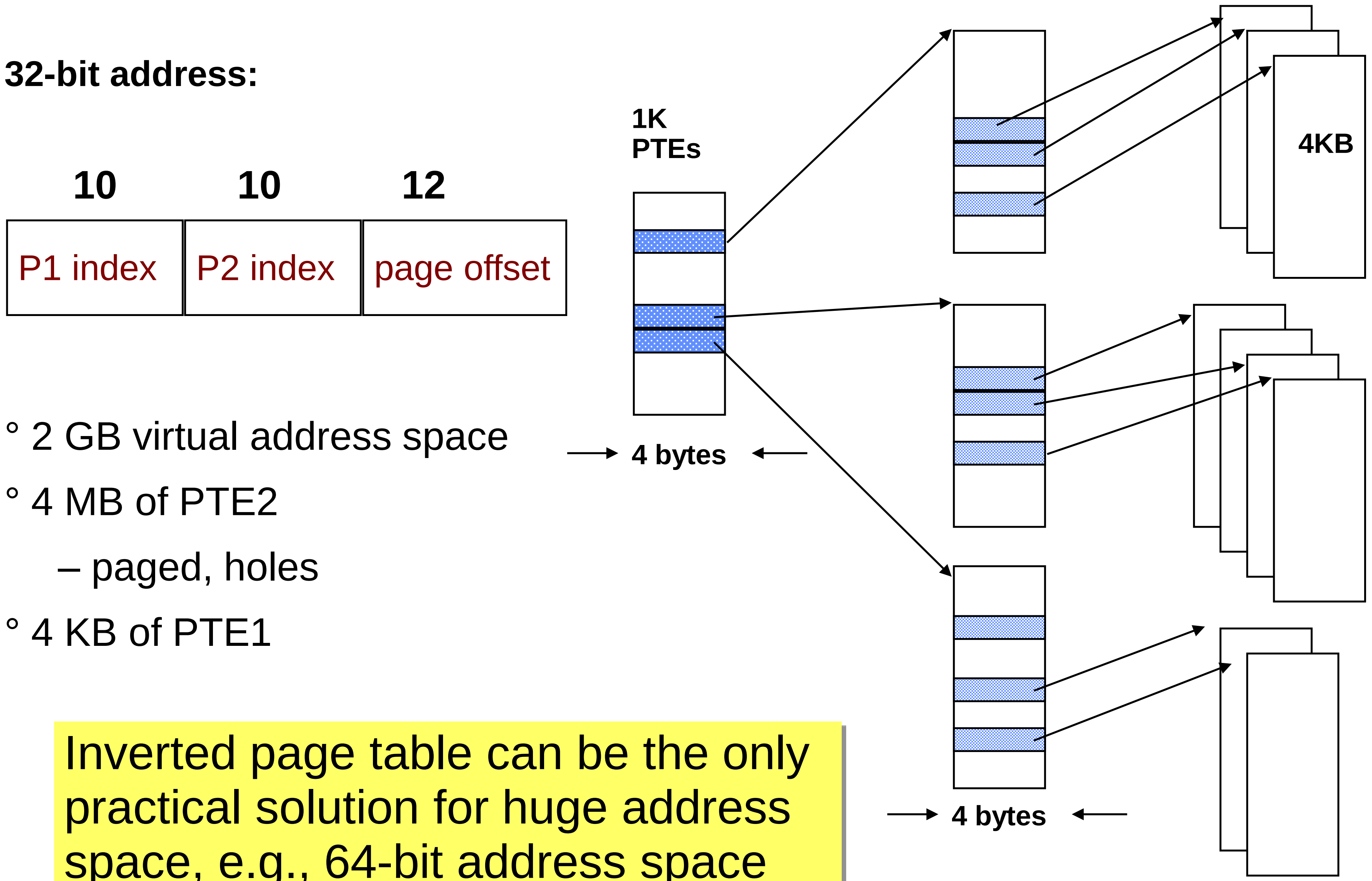
$$\text{Size of page table} = 2^{20} \text{ page table entries} \times 2^{2} \frac{\text{bytes}}{\text{page table entry}} = 4\,\text{MB}$$

<u>Optimization techniques:</u>

- Keep bound registers to limit the size of page table for given process in order to avoid empty slots
- Store only physical pages and apply hashing function of the virtual address (inverted page table)
- Allow paging of the page table, i.e. apply virtual addressing recursively
- Use multi-level page table to limit size of the table residing in main memory
- Cache the most used pages $\Rightarrow$ Translation Look-aside Buffer

Multi-Level Page Table

32-bit address:



° 2 GB virtual address space

° 4 MB of PTE2

– paged, holes

° 4 KB of PTE1

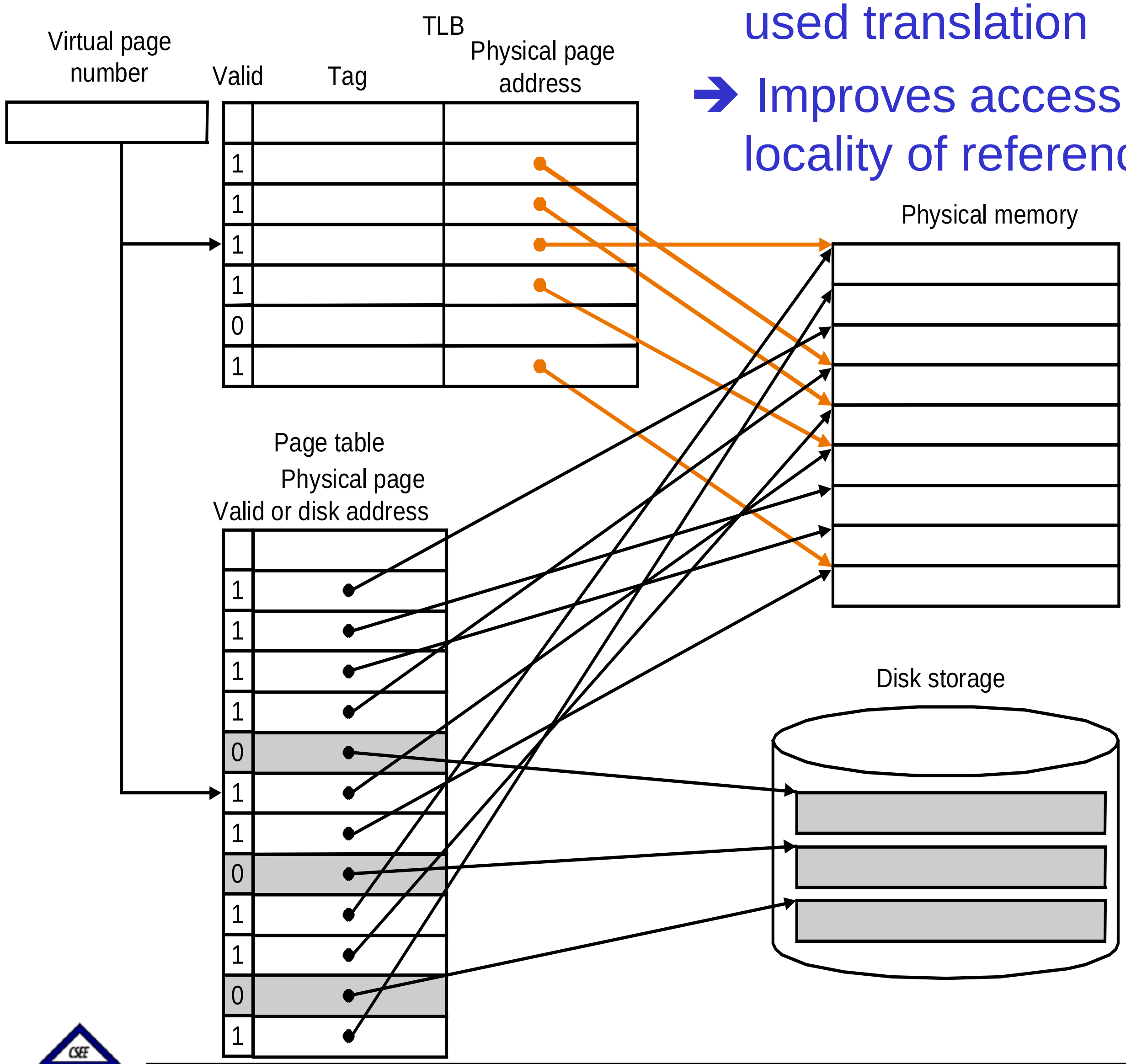
Inverted page table can be the only practical solution for huge address space, e.g., 64-bit address space

* Slide is courtesy of Dave Patterson

Translation Look-aside Buffer

➔ is a special cache that keeps track of recently used translation

➔ Improves access performance relying on locality of reference principle



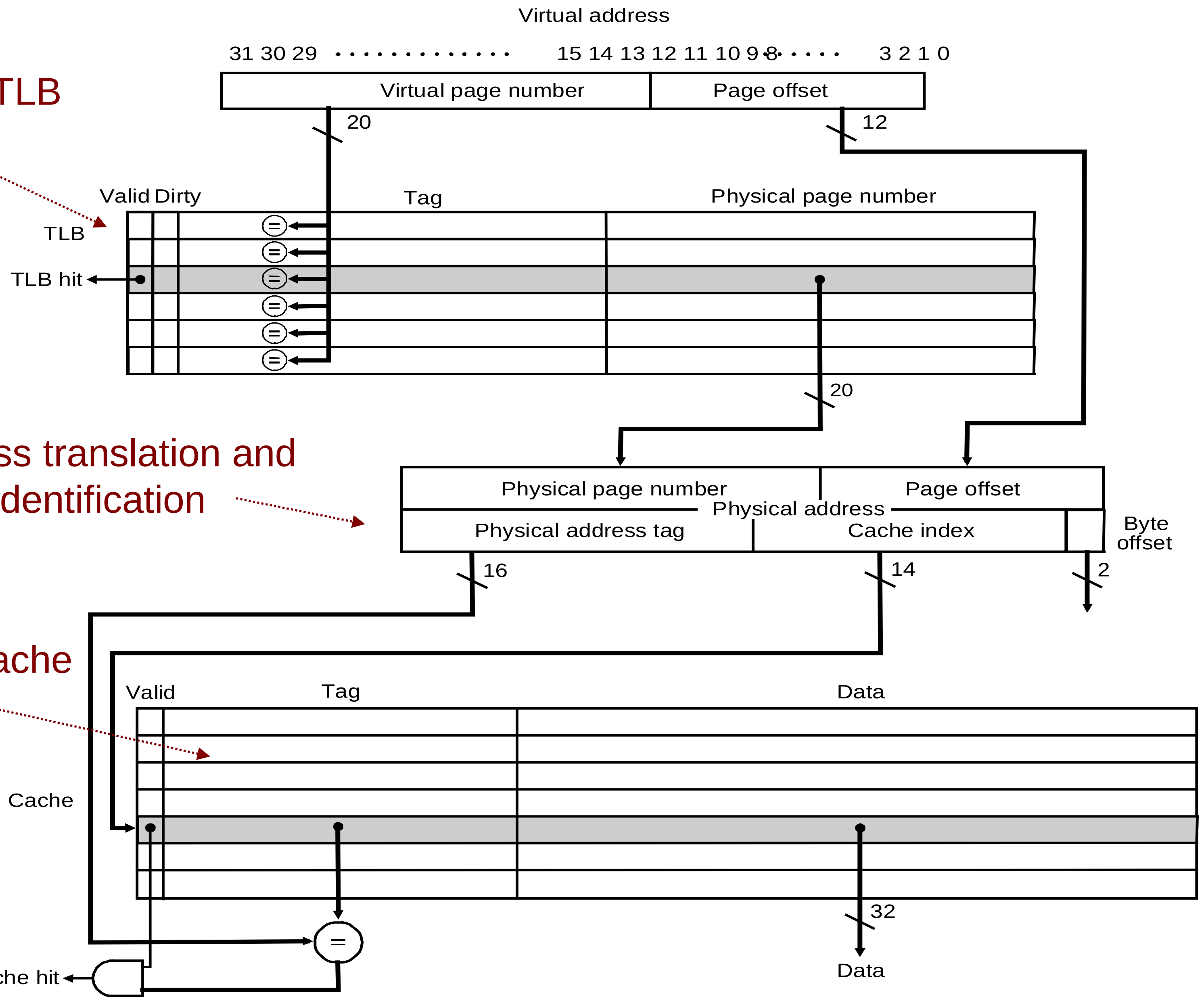
- TLB misses are exceptions that are typically handled at the hardware level
- Simple replacement strategy is applied to TLB misses since it happens frequently

TLB and Cache in MIPS

Fully associative TLB

Address translation and block identification

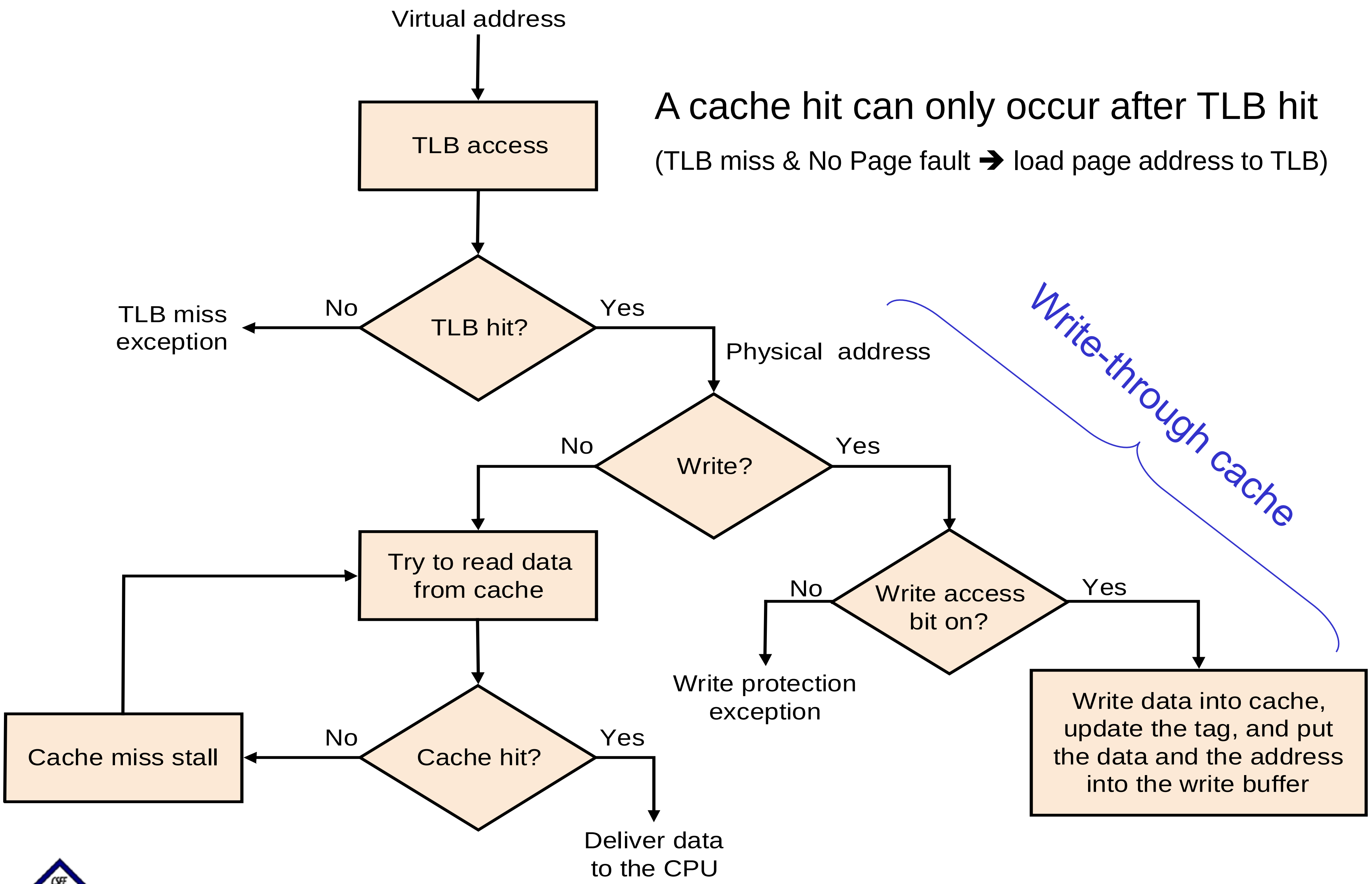
Direct-mapped Cache



TLB and Cache in MIPS

A cache hit can only occur after TLB hit

(TLB miss & No Page fault ➔ load page address to TLB)



Memory Related Exceptions

Possible exceptions:

Cache miss: referenced block not in cache and needs to be fetched from main memory

TLB miss: referenced page of virtual address needs to be checked in the page table

Page fault: referenced page is not in main memory and needs to be copied from disk

Cache	TLB	Page Table	Possible? If so, under what condition
miss	hit	hit	
hit	miss	hit	
miss	miss	hit	
miss	miss	miss	
miss	hit	miss	
hit	hit	miss	
hit	miss	miss	

Memory Related Exceptions

Possible exceptions:

Cache miss: referenced block not in cache and needs to be fetched from main memory

TLB miss: referenced page of virtual address needs to be checked in the page table

Page fault: referenced page is not in main memory and needs to be copied from disk

Cache	TLB	Page Table	Possible? If so, under what condition
miss	hit	hit	Possible, although the page table is never really checked if TLB hits
hit	miss	hit	
miss	miss	hit	
miss	miss	miss	
miss	hit	miss	
hit	hit	miss	
hit	miss	miss	

Memory Related Exceptions

Possible exceptions:

Cache miss: referenced block not in cache and needs to be fetched from main memory

TLB miss: referenced page of virtual address needs to be checked in the page table

Page fault: referenced page is not in main memory and needs to be copied from disk

Cache	TLB	Page Table	Possible? If so, under what condition
miss	hit	hit	Possible, although the page table is never really checked if TLB hits
hit	miss	hit	TLB misses, but entry found in page table and data found in cache
miss	miss	hit	
miss	miss	miss	
miss	hit	miss	
hit	hit	miss	
hit	miss	miss	

Memory Related Exceptions

Possible exceptions:

Cache miss: referenced block not in cache and needs to be fetched from main memory

TLB miss: referenced page of virtual address needs to be checked in the page table

Page fault: referenced page is not in main memory and needs to be copied from disk

Cache	**TLB**	**Page Table**	**Possible? If so, under what condition**
miss	hit	hit	Possible, although the page table is never really checked if TLB hits
hit	miss	hit	TLB misses, but entry found in page table and data found in cache
miss	miss	hit	TLB misses, but entry found in page table and data misses in cache
miss	miss	miss	
miss	hit	miss	
hit	hit	miss	
hit	miss	miss	

Memory Related Exceptions

Possible exceptions:

Cache miss: referenced block not in cache and needs to be fetched from main memory

TLB miss: referenced page of virtual address needs to be checked in the page table

Page fault: referenced page is not in main memory and needs to be copied from disk

Cache	TLB	Page Table	Possible? If so, under what condition
miss	hit	hit	Possible, although the page table is never really checked if TLB hits
hit	miss	hit	TLB misses, but entry found in page table and data found in cache
miss	miss	hit	TLB misses, but entry found in page table and data misses in cache
miss	miss	miss	TLB misses and followed by page fault. Data must miss in cache
miss	hit	miss	
hit	hit	miss	
hit	miss	miss	

Memory Related Exceptions

Possible exceptions:

Cache miss: referenced block not in cache and needs to be fetched from main memory

TLB miss: referenced page of virtual address needs to be checked in the page table

Page fault: referenced page is not in main memory and needs to be copied from disk

Cache	TLB	Page Table	Possible? If so, under what condition
miss	hit	hit	Possible, although the page table is never really checked if TLB hits
hit	miss	hit	TLB misses, but entry found in page table and data found in cache
miss	miss	hit	TLB misses, but entry found in page table and data misses in cache
miss	miss	miss	TLB misses and followed by page fault. Data must miss in cache
miss	hit	miss	Impossible: cannot have a translation in TLB if page is not in memory
hit	hit	miss	
hit	miss	miss	

Memory Related Exceptions

Possible exceptions:

Cache miss: referenced block not in cache and needs to be fetched from main memory

TLB miss: referenced page of virtual address needs to be checked in the page table

Page fault: referenced page is not in main memory and needs to be copied from disk

Cache	TLB	Page Table	Possible? If so, under what condition
miss	hit	hit	Possible, although the page table is never really checked if TLB hits
hit	miss	hit	TLB misses, but entry found in page table and data found in cache
miss	miss	hit	TLB misses, but entry found in page table and data misses in cache
miss	miss	miss	TLB misses and followed by page fault. Data must miss in cache
miss	hit	miss	Impossible: cannot have a translation in TLB if page is not in memory
hit	hit	miss	Impossible: cannot have a translation in TLB if page is not in memory
hit	miss	miss	

Memory Related Exceptions

Possible exceptions:

Cache miss: referenced block not in cache and needs to be fetched from main memory

TLB miss: referenced page of virtual address needs to be checked in the page table

Page fault: referenced page is not in main memory and needs to be copied from disk

Cache	**TLB**	**Page Table**	**Possible? If so, under what condition**
miss	hit	hit	Possible, although the page table is never really checked if TLB hits
hit	miss	hit	TLB misses, but entry found in page table and data found in cache
miss	miss	hit	TLB misses, but entry found in page table and data misses in cache
miss	miss	miss	TLB misses and followed by page fault. Data must miss in cache
miss	hit	miss	Impossible: cannot have a translation in TLB if page is not in memory
hit	hit	miss	Impossible: cannot have a translation in TLB if page is not in memory
hit	miss	miss	Impossible: data is not allowed in cache if page is not in memory

Memory Protection

- ❑ It is always desirable to prevent a process from corrupting allocated memory space of other processes
- ❑ The processor must support processes in a non-privileged mode to avoid messing up memory protection
- ❑ Implementation can be by mapping independent virtual pages to separate physical pages
- ❑ Write protection bits would be included in the page table for authentication
- ❑ Sharing pages can be facilitated by the operating system through mapping virtual pages of different processes to same physical pages
- ❑ To enable the operating system to implement protection, the hardware must provide at least the following capabilities:
  - ➔ Support at least two modes of operations, one of them is a user mode
  - ➔ Provide a portion of CPU state that a user process can read but not write, e.g. page pointer
  - ➔ Enable change of operation modes through special instructions

Handling TLB Misses & Page Faults

- **<u>TLB Miss</u>**: *(hardware-based handling)*
  - ➔ Check if the page is in memory (valid bit) --> update the TLB
  - ➔ Generate page fault exception if page is not in memory
- **<u>Page Fault</u>**: *(handled by operating system)*
  - ➔ Transfer control to the operating system
  - ➔ Save processor status: registers, program counter, page table pointer, etc.
  - ➔ Lookup the page table and find the location of the reference page on disk
  - ➔ Choose a physical page to be replaced by the referenced page, if the candidate physical page is modified (dirty bit is set) the page needs to be written back
  - ➔ Start reading the referenced page from disk to the assigned physical page
- The processor needs to support "restarting" instructions in order to guarantee correct execution (easily supported in MIPS)
- The user process causing the page fault will be suspended by the operating system until the page is readily available in main memory
- Protection violations are handled by the operating system similarly but without automatic instruction restarting

Conclusion

- ***Summary***
  - ➔ Virtual Memory
    - Virtual addressing
    - Address translation
  - ➔ Memory paging
    - Page table
    - Page faults
    - Translation look-aside buffer
  - ➔ Memory-related exceptions
    - Relationship between TLB, cache miss and page fault exceptions
    - Handling of memory-related exceptions
- ***Next Lecture***
  - ➔ Input/Output system

Read sections 5.4 in 4th Ed. Or 5th Ed. of the textbook