

Multicore Computer Architecture - Storage and Interconnects

Lecture 11

Address Translation Mechanisms



NPTEL Online Course: Multi-Core Computer Architecture and Interconnects, Dr. John Jose, IIT Guwahati

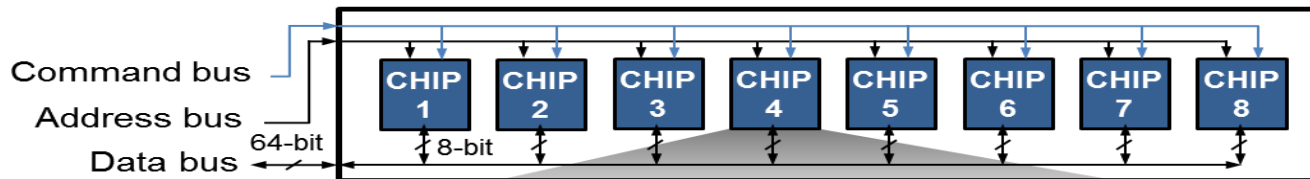
Dr. John Jose

Assistant Professor

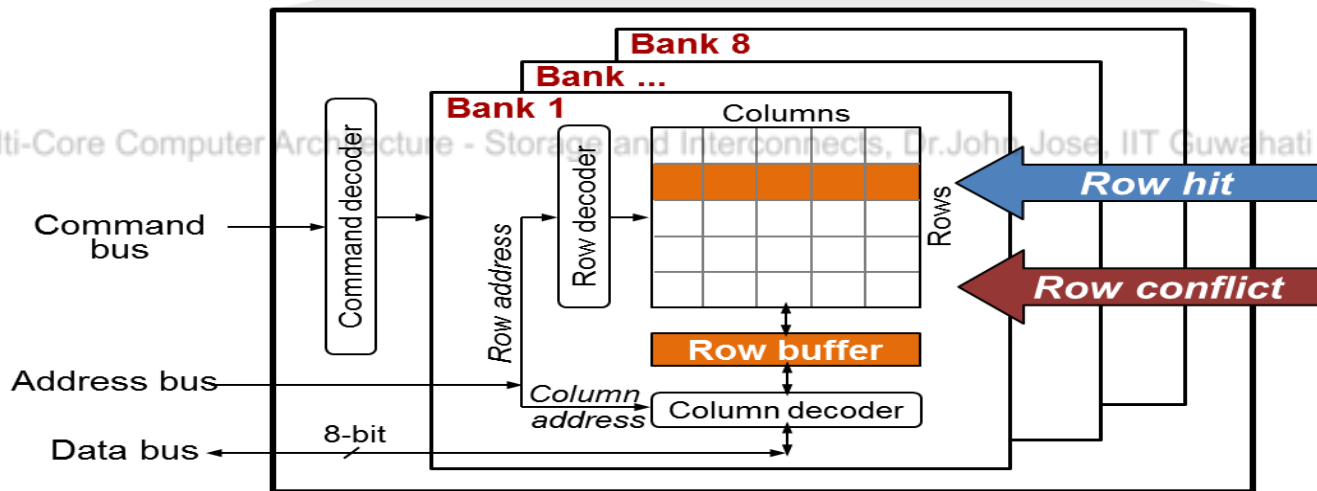
**Department of Computer Science & Engineering
Indian Institute of Technology Guwahati, Assam.**

DRAM Rank

DRAM Rank



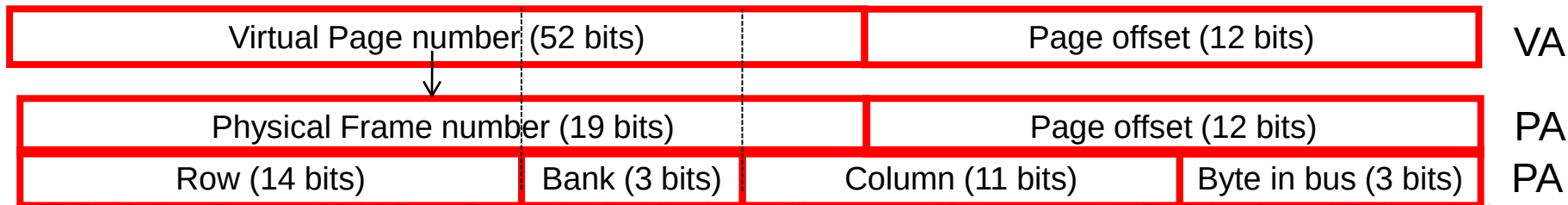
DRAM Chip



DRAM access latency varies depending on which row is stored in the row buffer

Virtual To Physical Address Mapping

- ❖ Operating System influences where an address maps to in DRAM



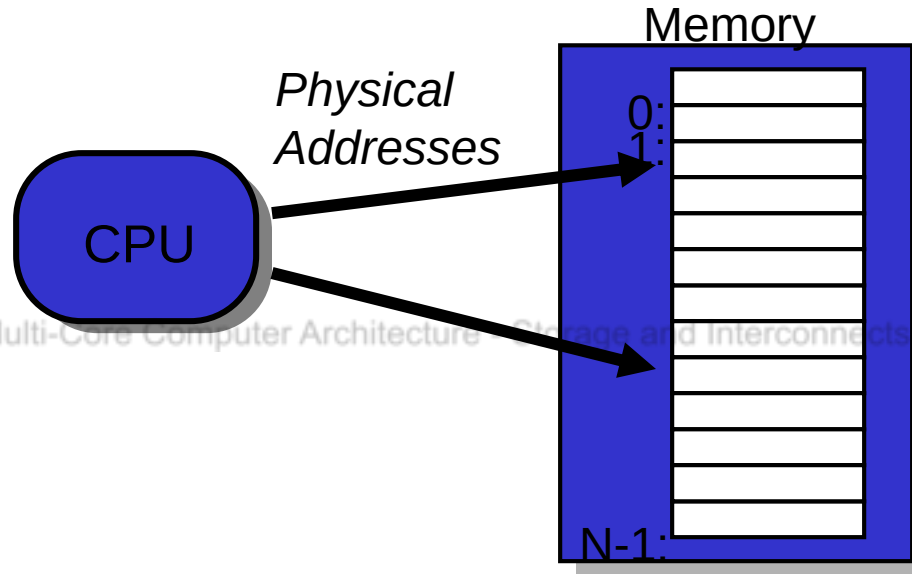
NPTEL Online Course: Multi-Core Computer Architecture - Storage and Interconnects, Dr. John Jose, IIT Guwahati

- ❖ Operating system can influence which bank/channel/rank a virtual page is mapped to.

Virtual Memory Techniques

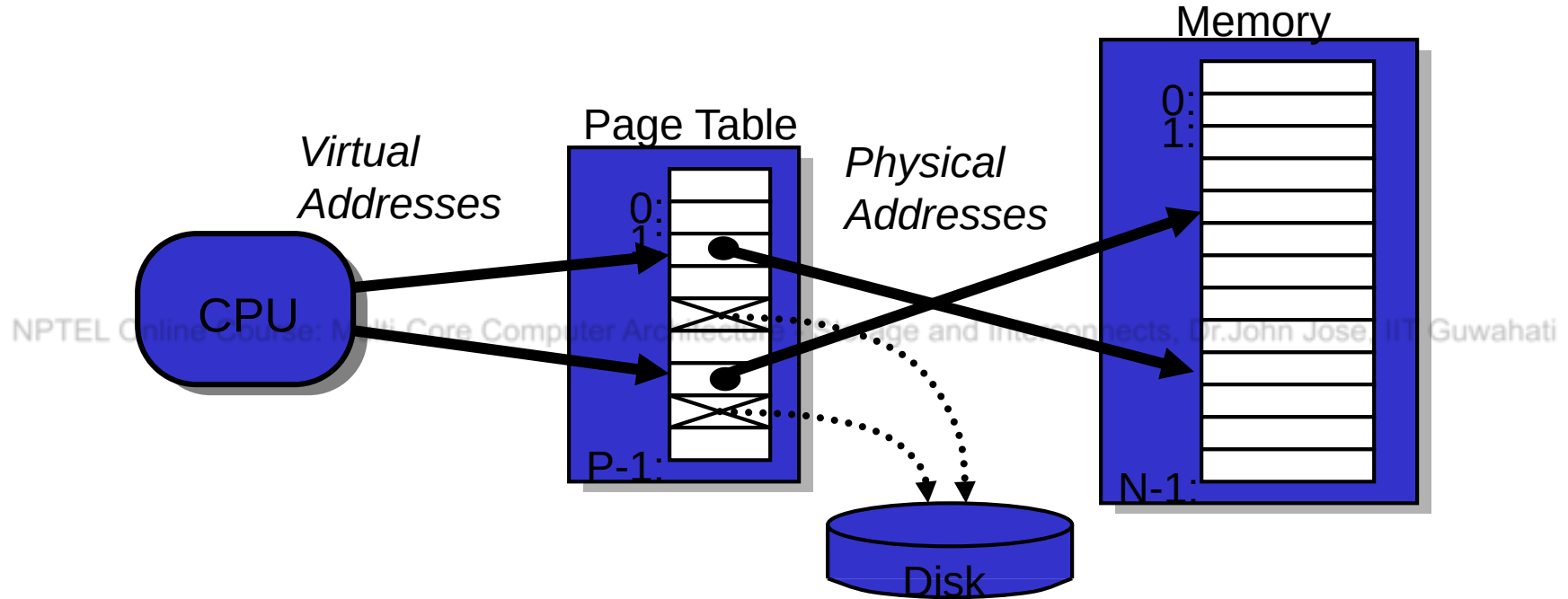
- ❖ Address space of a process can exceed physical memory
- ❖ Sum of address spaces of multiple processes can exceed physical memory
- ❖ Multiple processes resident in main memory each with its own address space
- ❖ One process should not interfere with another because they operate in different address spaces.
- ❖ Different sections of address spaces have different permissions.

A System with Physical Memory Only



- ❖ Addresses generated by the CPU correspond directly to bytes in physical memory

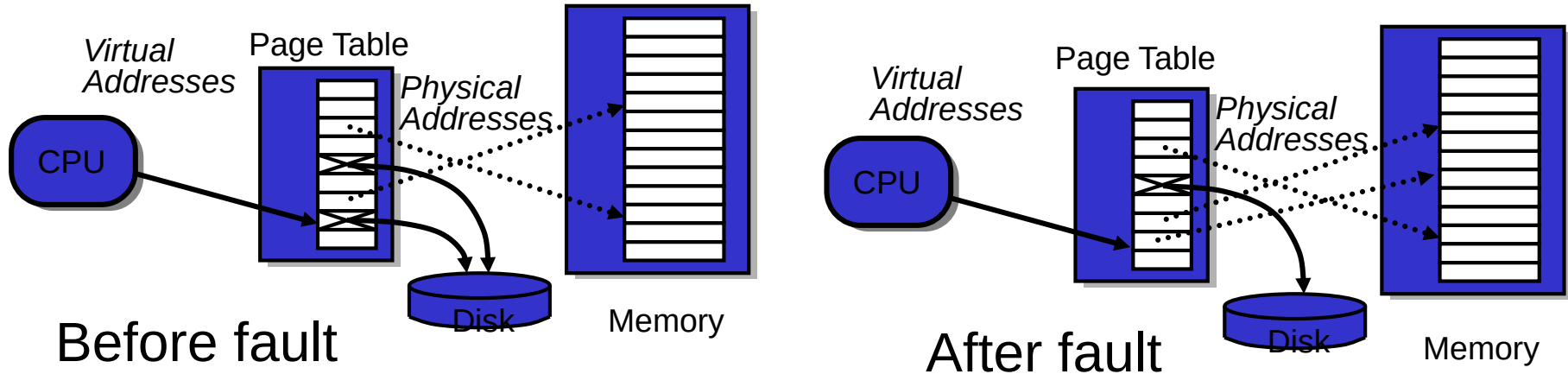
A System with Virtual Memory



- ❖ Address Translation: Hardware converts virtual addresses to physical addresses via OS-managed lookup table (page table)

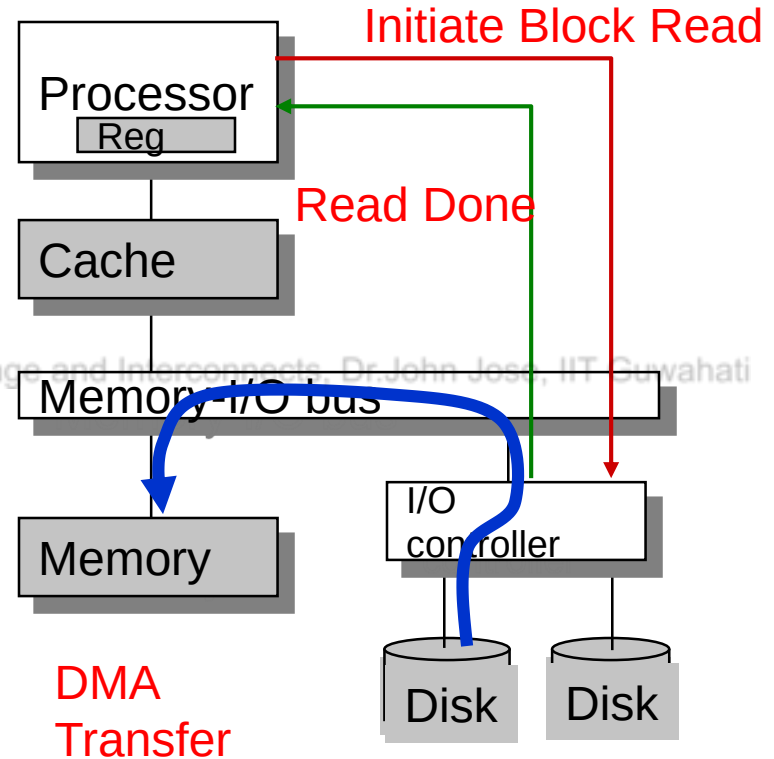
Page Faults

- ❖ What if an object is on disk rather than in memory?
 - ❖ Page table entry indicates virtual address not in memory
 - ❖ OS exception handler move data from disk into memory
 - ❖ OS has full control over this page movement from disk to memory and page table update.



Servicing a Page Fault

- ❖ **Processor Signals Controller**
 - ❖ Read block of length P starting at disk address X and store starting at memory address Y
- ❖ **Read Occurs**
 - ❖ Direct Memory Access (DMA)
 - ❖ Under control of I/O controller
- ❖ **I/O Controller Signals Completion**
 - ❖ Interrupt processor
 - ❖ OS resumes suspended process



Demand Paging

- ❖ Bring a page into memory only when it is needed
 - ❖ Less I/O needed
 - ❖ Less memory needed
 - ❖ Faster response (no need to wait for all pages to load)
 - ❖ More users
- ❖ Page is needed $\Rightarrow$ reference to it
- ❖ Page not-in-memory $\Rightarrow$ bring to memory
- ❖ **Lazy swapper** – never swaps a page into memory unless page will be needed

Page Table –Memory – Disk Interaction

0	A
1	B
2	C
3	D
4	E
5	F
6	G
7	H

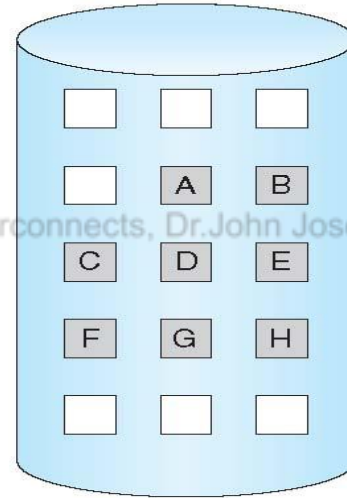
logical
memory

valid-invalid bit		
frame		
0	4	v
1		i
2	6	v
3		i
4		i
5	9	v
6		i
7		i

page table

0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15

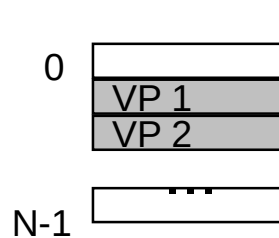
physical memory



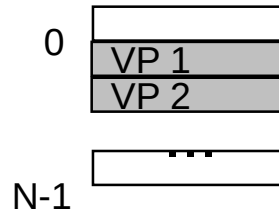
Virtual and Physical Address Spaces

- ❖ Virtual and physical address spaces divided into equal-sized blocks called pages
- ❖ Each process has its own virtual address space
- ❖ Operating system controls how virtual pages are assigned to physical memory

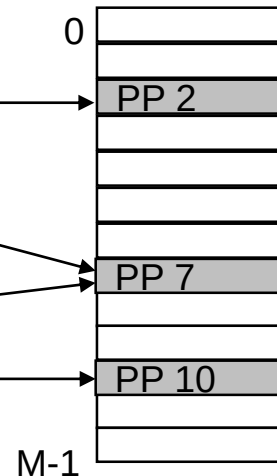
Virtual Address Space for P1:



Virtual Address Space for P2:



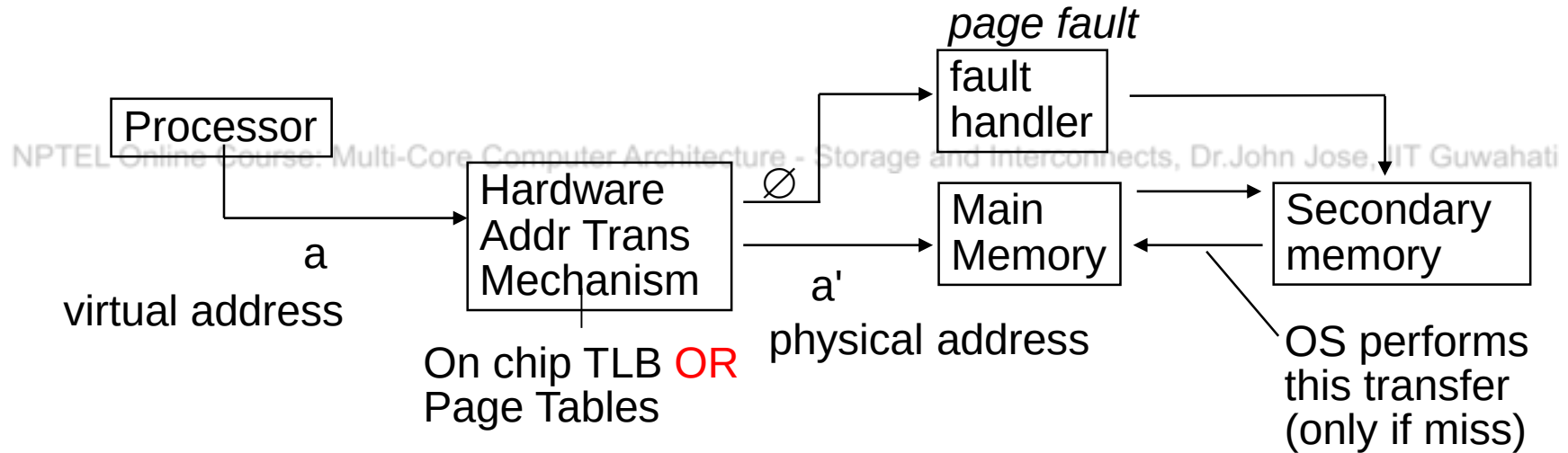
Address Translation



Physical Address Space (DRAM)

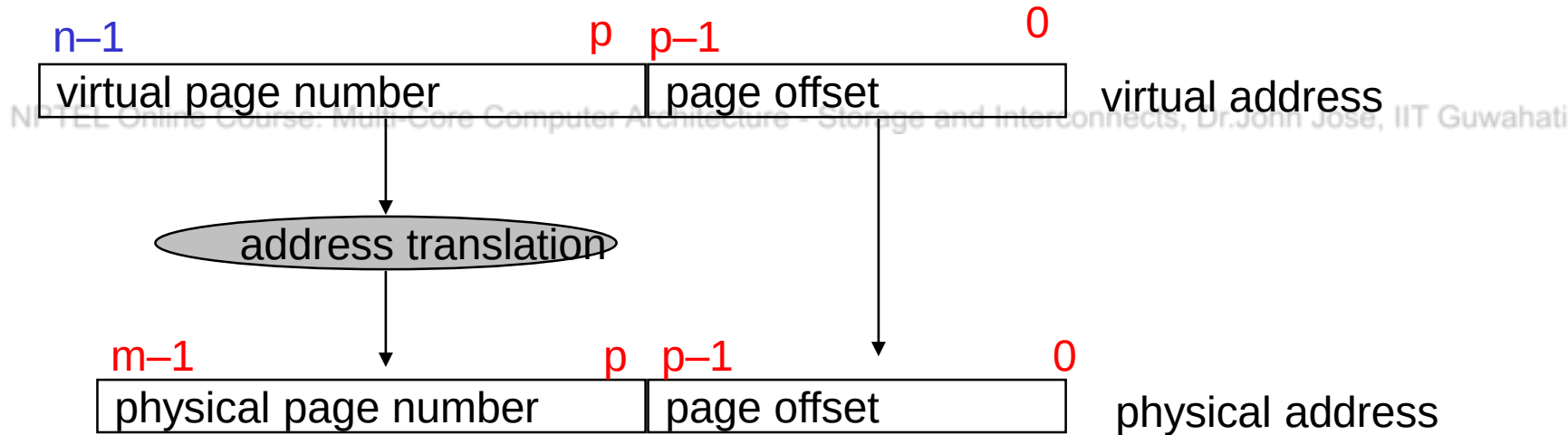
(e.g., read/only library code)

VM Address Translation



VM Address Translation

- ❖ $P = 2^p =$ page size (bytes).
- ❖ $N = 2^n =$ Virtual address limit
- ❖ $M = 2^m =$ Physical address limit



- ❖ Page offset bits don't change as a result of translation

Page Tables

Virtual Page Number

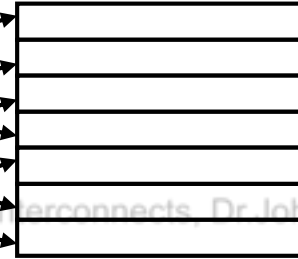


Memory resident
page table

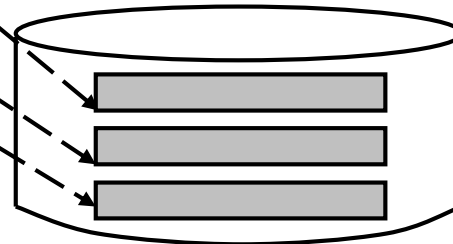
valid (physical page or disk address)

1	●
1	●
0	●
1	●
1	●
1	●
0	●
1	●
0	●
1	●

Physical Memory



Disk Storage



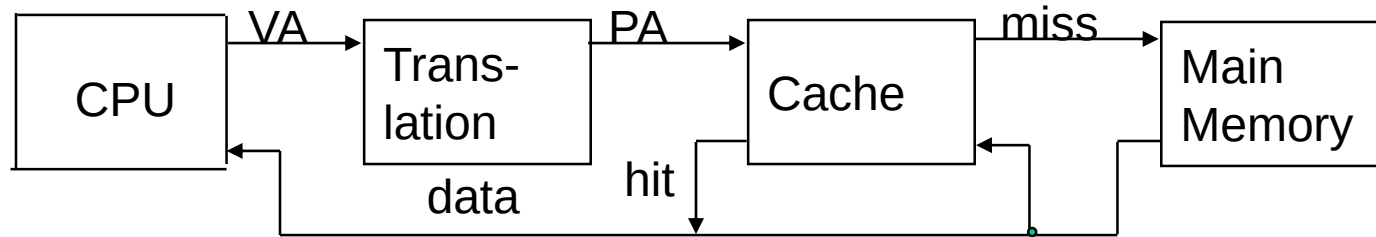
Protection via Page Table

- ❖ Page Table access also allows protection / access rights
- ❖ Access rights field indicate allowable access (e.g., read-only, read-write, execute-only)
- ❖ Page tables typically support multiple protection modes (e.g., kernel vs. user)
- ❖ Protection violation fault if user doesn't have necessary permission

Page Size

- ❖ The size of the page table is inversely proportional to the page size
- ❖ Transferring larger pages to or from secondary storage, possibly over a network, is more efficient than transferring smaller pages.
- ❖ The number of TLB entries is restricted, so a larger page size means that more memory can be mapped efficiently, thereby reducing the number of TLB misses.
- ❖ Unused memory within a page is internal fragmentation
- ❖ A small page size will result in less wasted storage

Integrating VM and Cache

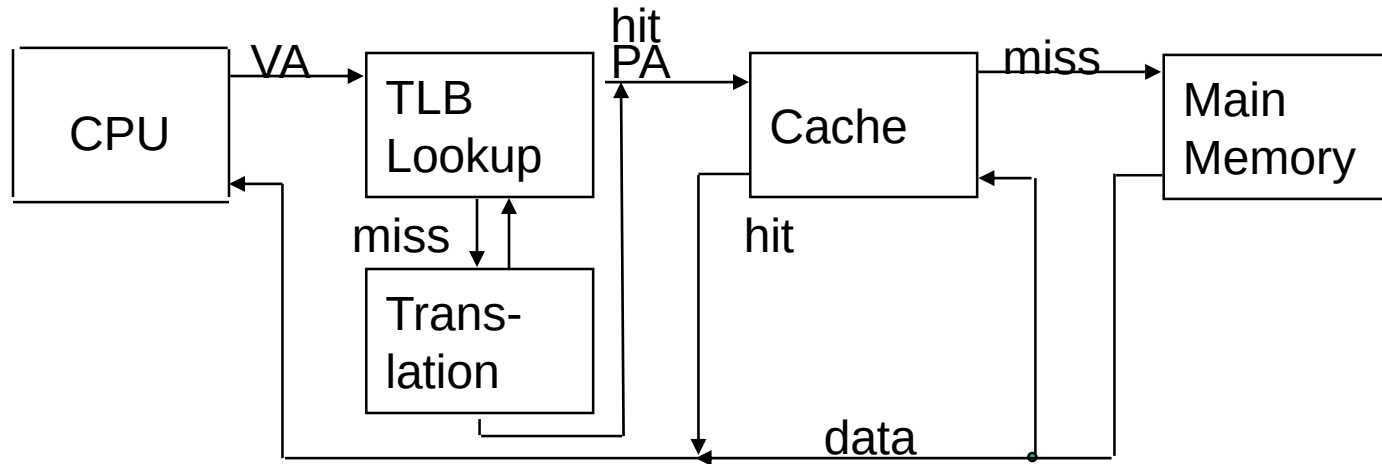


- ❖ Caches are accessed by physical addresses
- ❖ Multiple processes can have blocks in cache at same time
- ❖ Cache doesn't need to be concerned with protection issues
- ❖ Access rights checked as part of address translation
- ❖ Perform Address Translation Before Cache Lookup - TLB

Address Translation with a TLB

- ❖ Translation Lookaside Buffer (TLB) is a small hardware cache
- ❖ Maps virtual page numbers to physical page numbers
- ❖ Contains complete page table entries for small number of pages (recent pages)

NPTEL Online Course: Multi-Core Computer Architecture - Storage and Interconnects, Dr. John Jose, IIT Guwahati



Multi-Level Page Tables

- ❖ Consider 4KB (2^{12}) page size, 32-bit virtual address space, 4-byte PTE :
What is the size of page table ?

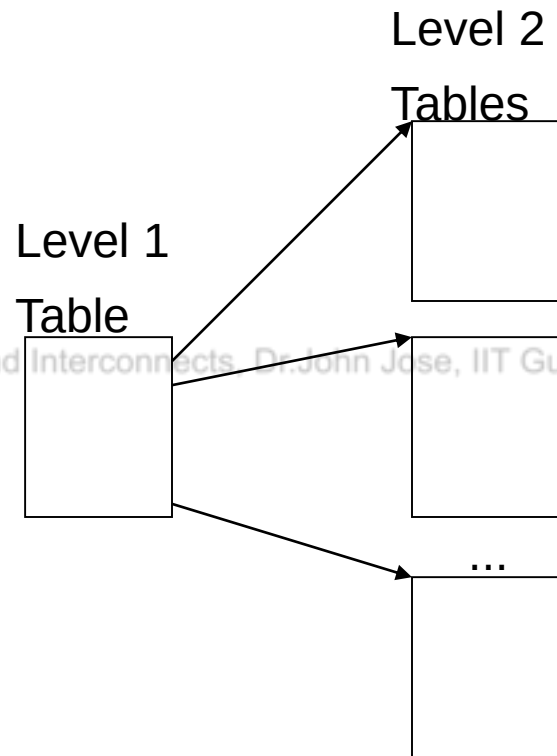
- ❖ Would need a 4 MB page table!

- ❖ Common solution

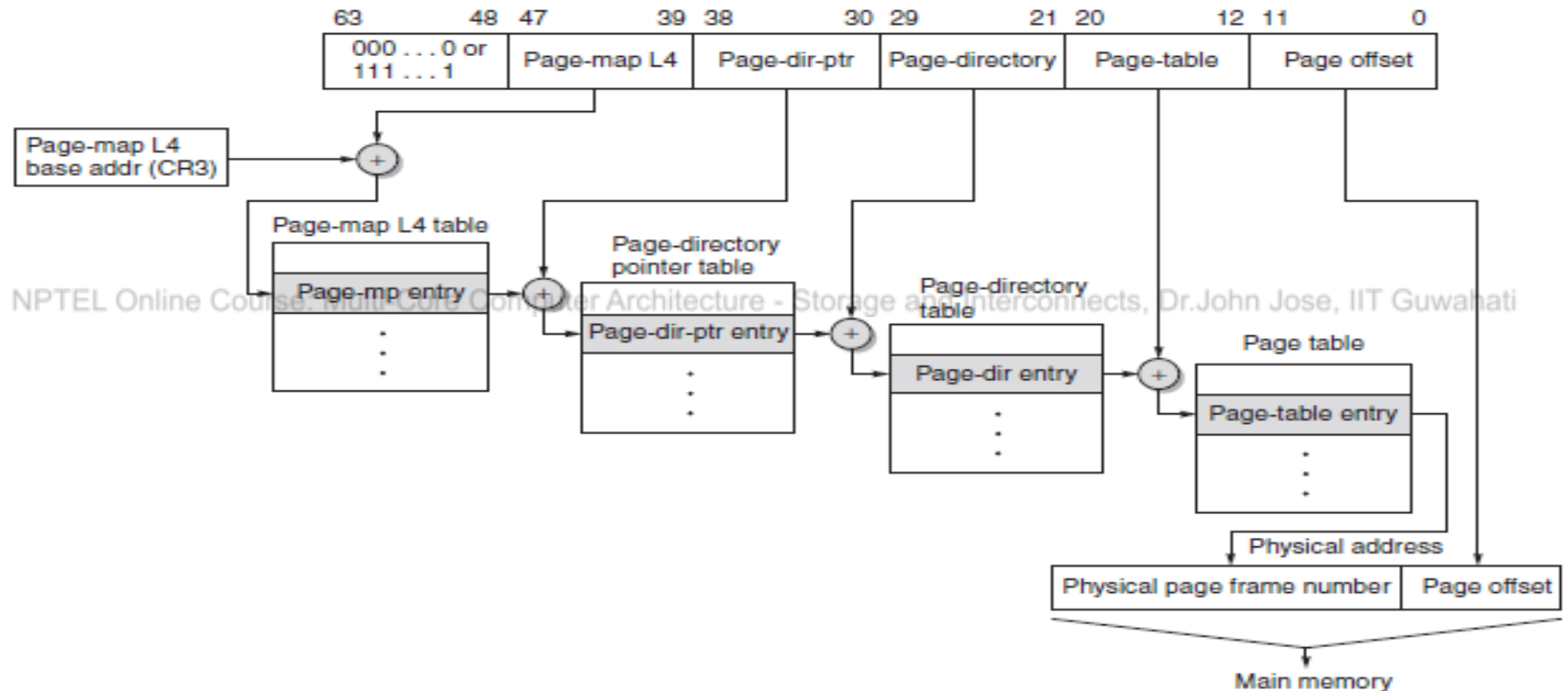
- ❖ Multi-level page tables

- ❖ Level 1 table: 1024 entries, each of which points to a Level 2 page table.

- ❖ Level 2 table: 1024 entries, each of which points to a page



Multi-Level Page Tables



**THANK
YOU!**

FOR

YOUR

PRESENCE

NPTEL Online Course: Multi-Core Computer Architecture, Storage and Interconnects, Dr. John Jose, IIT Guwahati

johnjose@iitg.ac.in

<http://www.iitg.ac.in/johnjose/>