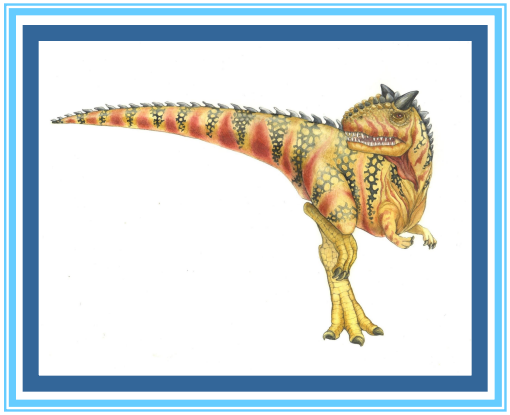
Chapter 8: Main Memory
Part 4



Recap

- Memory Addressing
- Physical and logical addresses
- Hardware address protection
- Memory management Unit
- Dynamic memory allocation
- Memory Management Technique : Segmentation

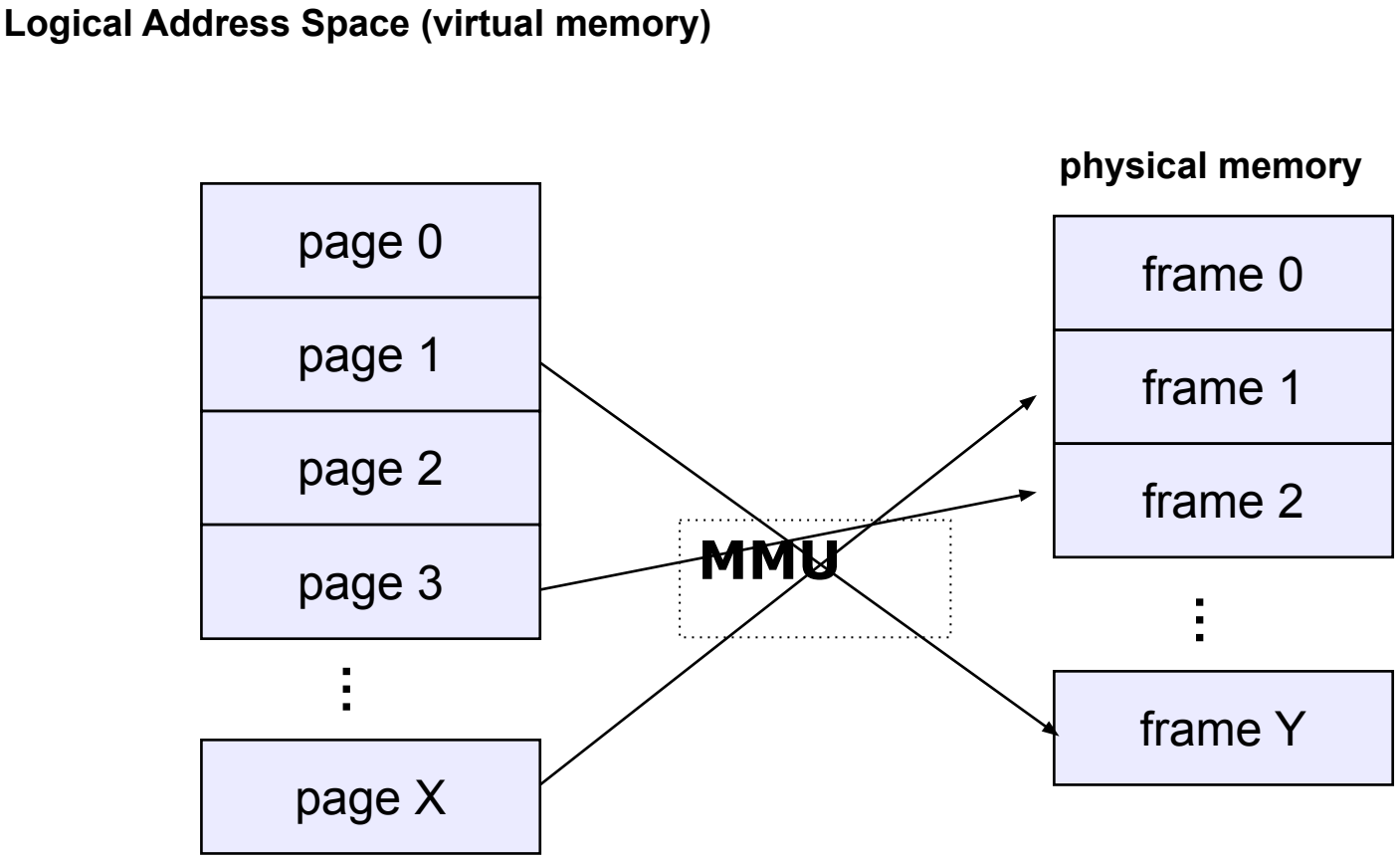
Objectives

- Non-Contiguous Memory Management Techniques:
  - **Paging**

Paging

- Physical address space of a process can be noncontiguous; process is allocated physical memory whenever the latter is available
- Memory partitions are same fixed size.
- The mapping from virtual logical to physical address is done by the memory management unit (MMU) which is a hardware device and this mapping is known as paging technique.
- The Physical Address Space is conceptually divided into a number of fixed-size blocks, called **frames**.
- The Logical address Space is also splitted into fixed-size blocks, called **pages**.
- Page Size = Frame Size

Address Mapping



Paging

- **Frames:**

  Divide physical memory into fixed-sized blocks called **frames**

- **Pages:**

  Divide logical memory into blocks of same size called **pages**

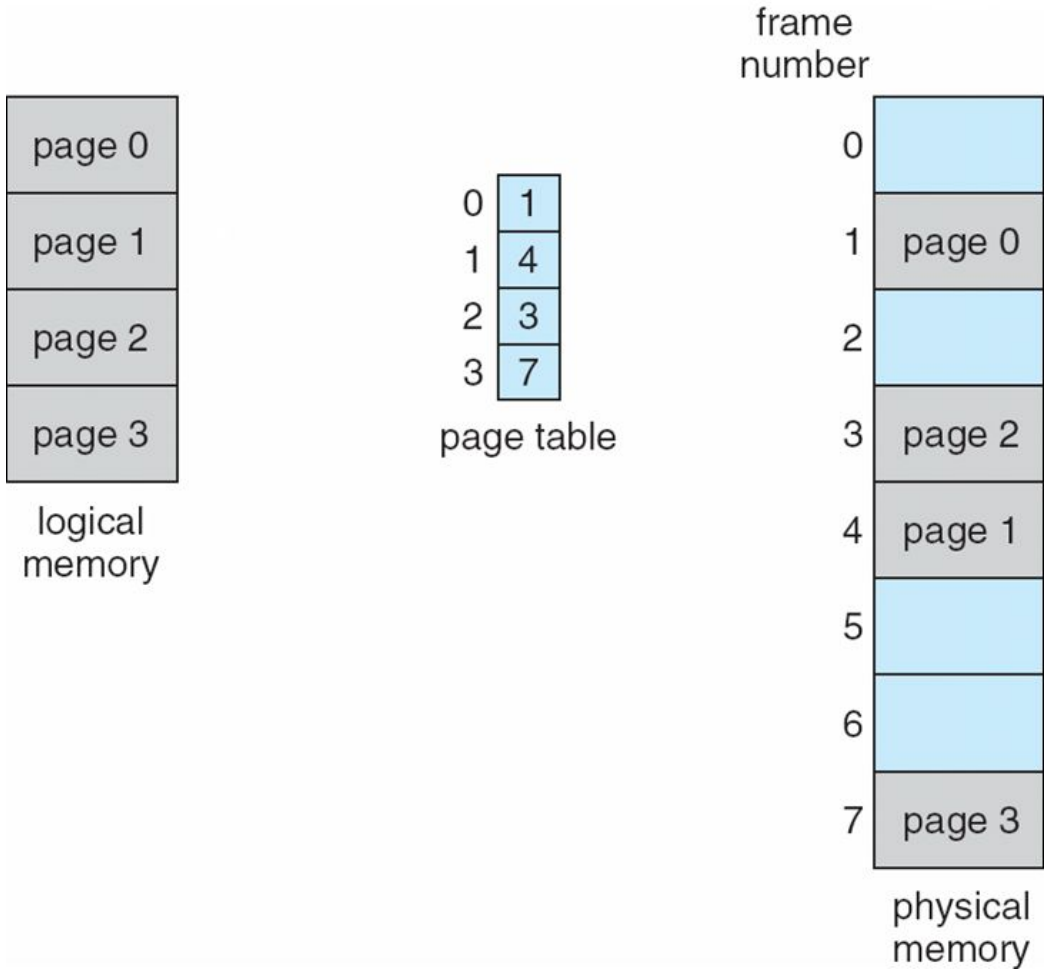
- **Page Table:**

  Set up a **page table** to translate logical to physical addresses

- *To run a program of size **N** pages, need to find **N** free frames and load program*

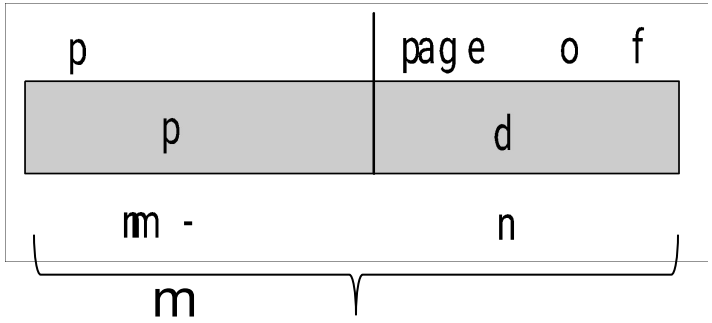
- **Advantage of Paging:**
  - Easy to use memory management algorithm
  - No need for external Fragmentation
  - Swapping is easy between equal-sized pages and page frames.
- **Disadvantages:**
  - Still have Internal fragmentation
  - Overhead to maintain data structure of page table

Paging Model of Logical and Physical Memory



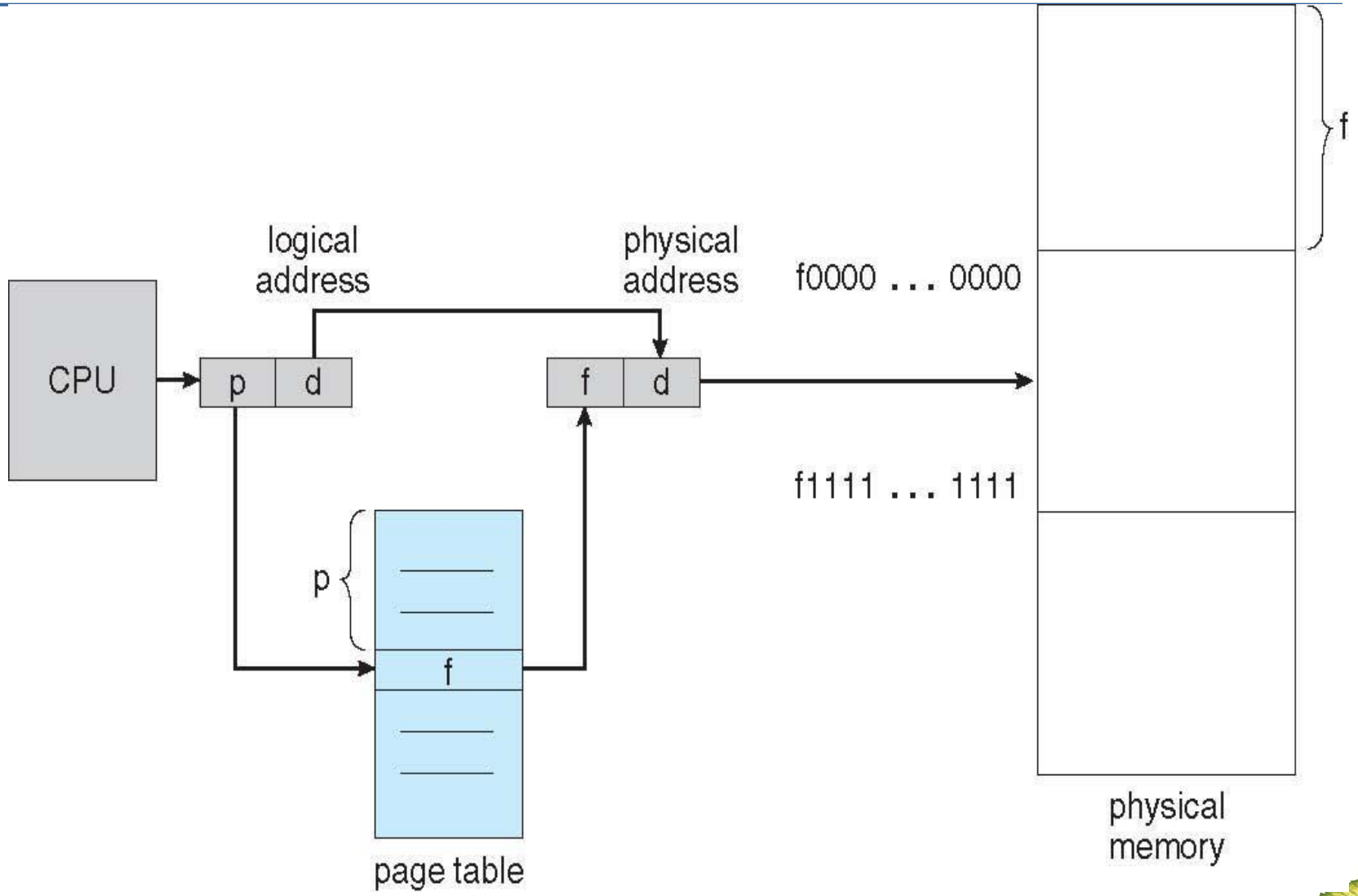
Address Translation Scheme

- Address generated by CPU is divided into:
  - **Page number (*p*) –** used as an index into a **page table** which contains base address of each page in physical memory
  - **Page offset (*d*)** – combined with base address to define the physical memory address that is sent to the memory unit

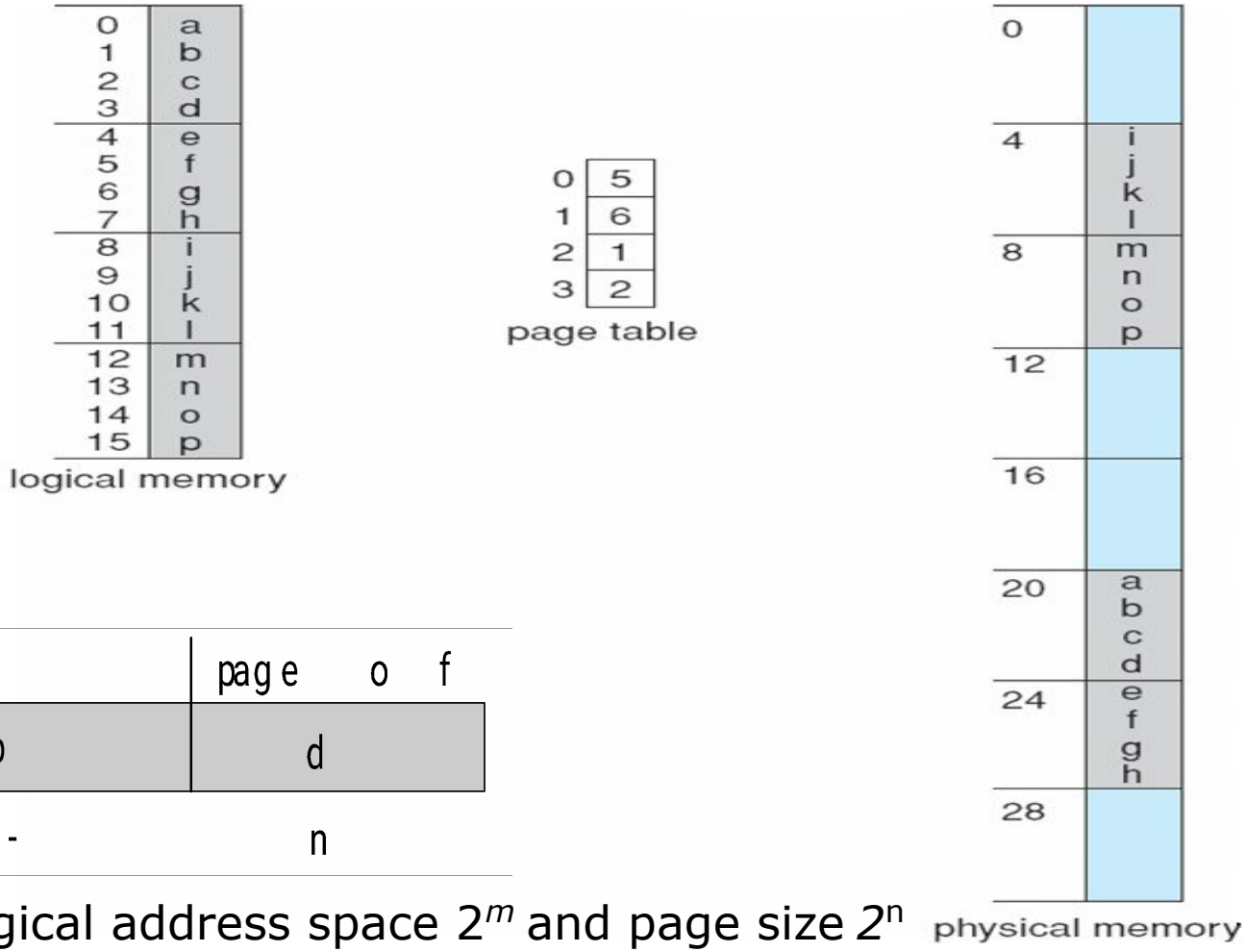


  - For given logical address space 2^m and page size 2^n

Paging Hardware



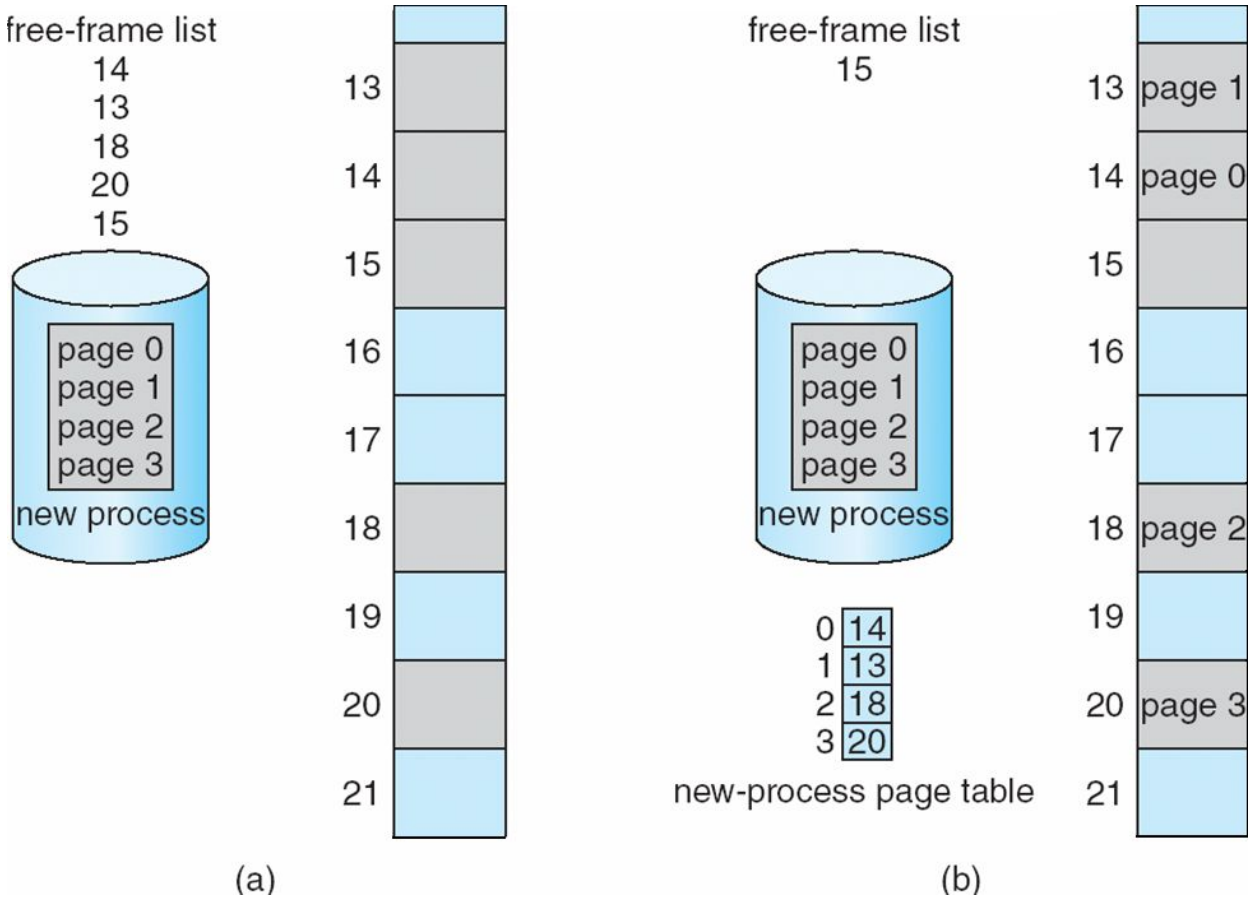
Paging Example



logical address space 2^m and page size 2^n

n=2 and m=4 16-byte logical memory and page size 4-byte

Free Frames



Before allocation

After allocation

Implementation of Page Table

- Page table (data structure) is kept in main memory
- **Page-table base register** (**PTBR**) points to the page table
- **Page-table length register** (**PTLR**) indicates size of the page table
- In this scheme every data/instruction access requires two memory accesses
  - One for the page table and one for the data / instruction
- The hardware implementation of page table can be done by using dedicated registers. But the usage of register for the page table is satisfactory only if page table is small. If page table contain large number of entries then we can use TLB(translation Look-aside buffer), a special, small, fast look up hardware cache.
- The two-memory access problem can be solved by the use of a special fast-lookup hardware cache called **associative memory** or **translation look-aside buffers** (**TLBs**)

Implementation of Page Table (Cont.)

- Some TLBs store **address-space identifiers** (**ASIDs**) in each TLB entry – uniquely identifies each process to provide address-space protection for that process
  - Otherwise need to flush at every context switch
- TLBs typically small (64 to 1,024 entries)
- On a TLB miss, value is loaded into the TLB for faster access next time
  - Replacement policies must be considered
  - Some entries can be **wired down** for permanent fast access

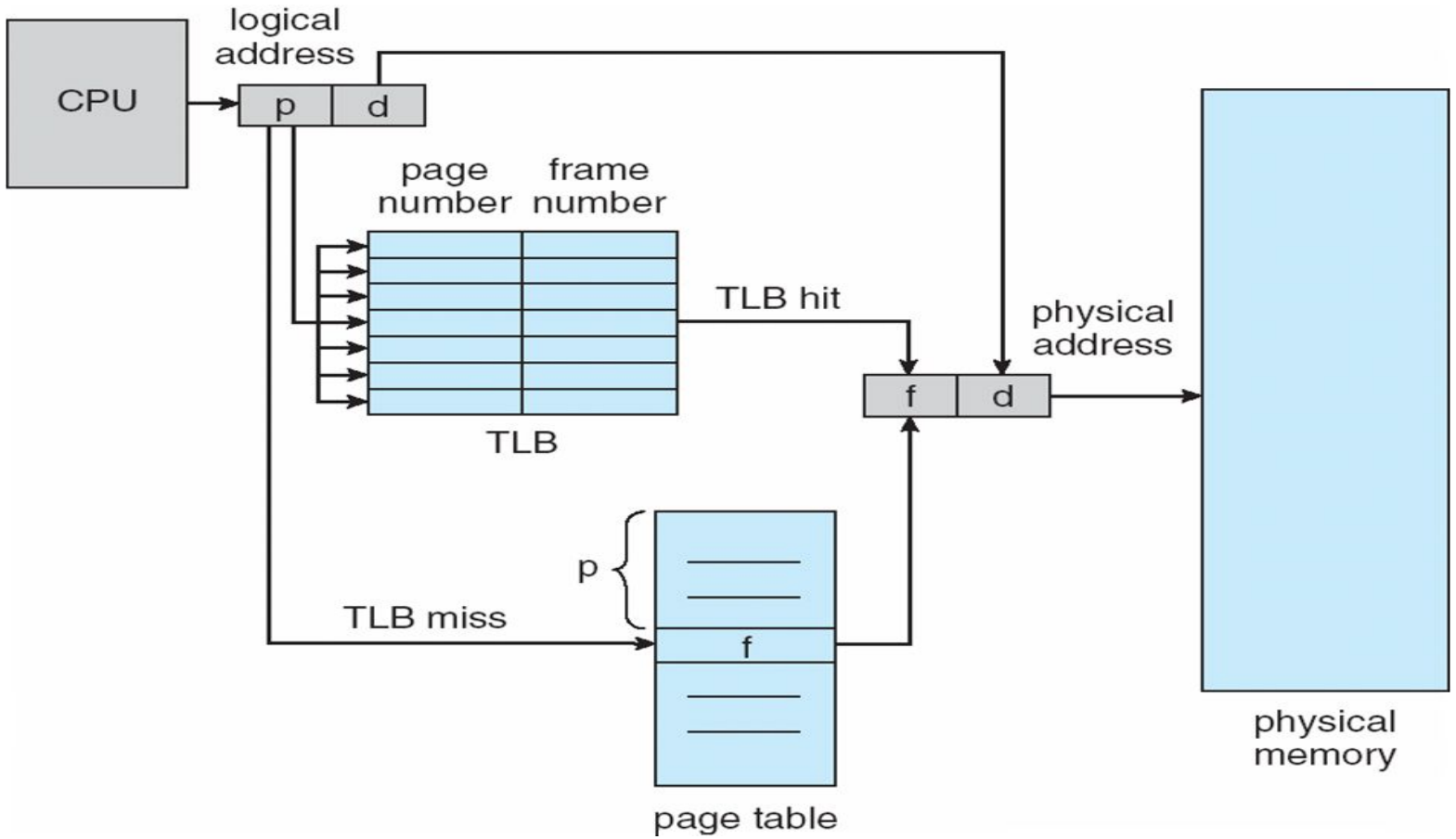
Associative Memory

- Associative memory – parallel search

Page #	Frame #

- Address translation (p, d)
  - If p is in associative register, get frame # out
  - Otherwise get frame # from page table in memory

Paging Hardware With TLB



Paging Protection

- The paging process should be protected by using the concept of insertion of an additional bit called Valid/Invalid bit. Paging Memory protection in paging is achieved by associating protection bits with each page.
- These bits are associated with each page table entry and specify protection on the corresponding page.