CHAPTER -6

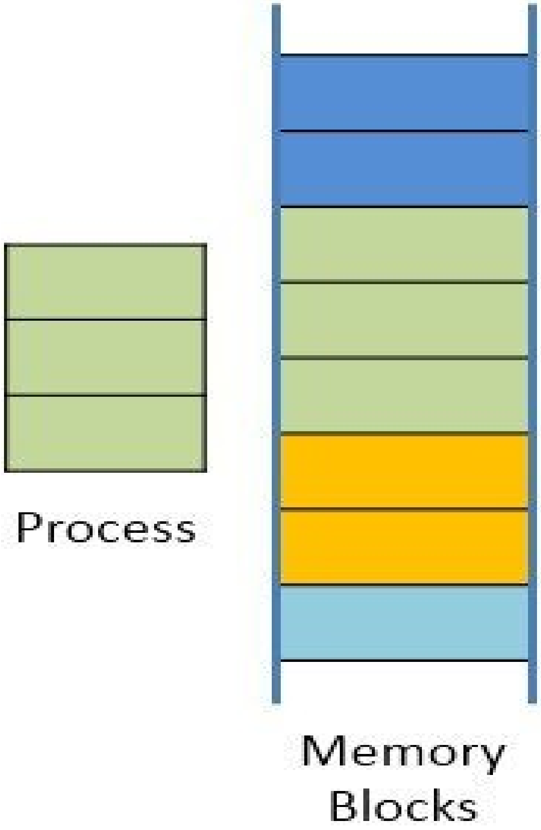
PAGING AND SEGMENTATION

Non-Contiguous Allocation

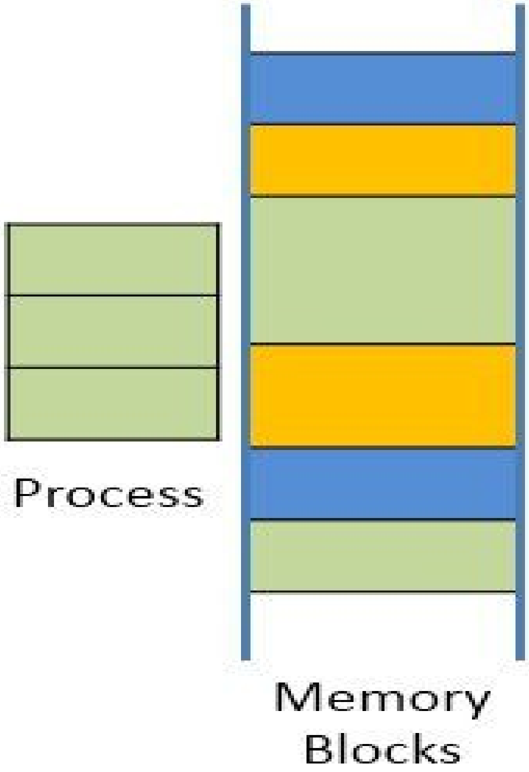
Non-contiguous allocation divides the process into several blocks and place them in the different address space of the memory. The following two schemes are available.

(i)Paging

(ii) Segmentation



Contiguous Memory Allocation



Noncontiguous Memory Allocation

Paging

Paging is a memory management scheme which permits the physical address of a process to non-contiguous. This paging scheme gives solution to

Paging

- Divide logical memory into blocks of same size called **pages.**
- Divide physical memory into fixed-sized blocks called **frames**
- When a process is executed, its pages are loaded into frames in main memory from disk

Address Translation Scheme

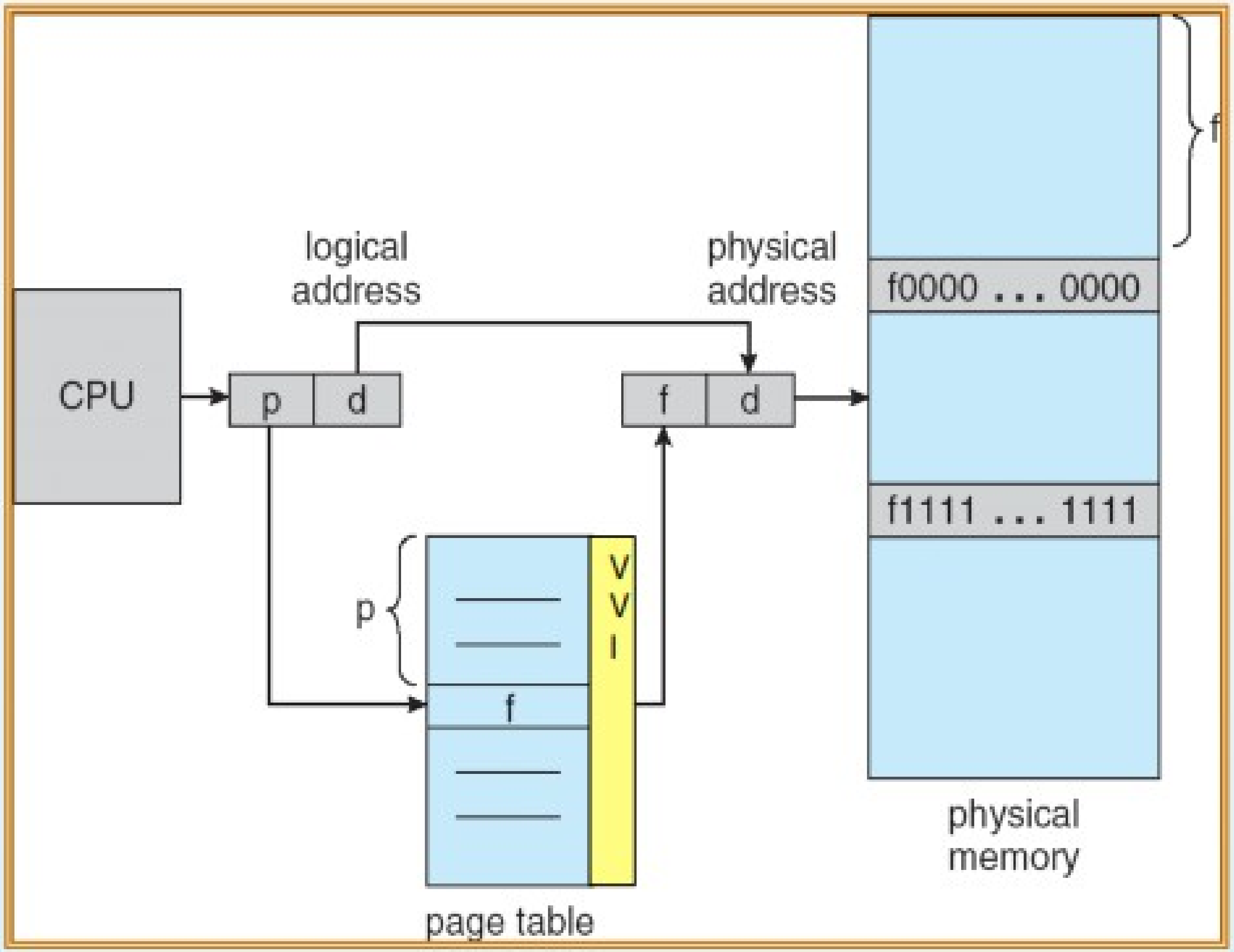
Address generated by CPU is divided into:

 Page number (p) – used as an index into a page table

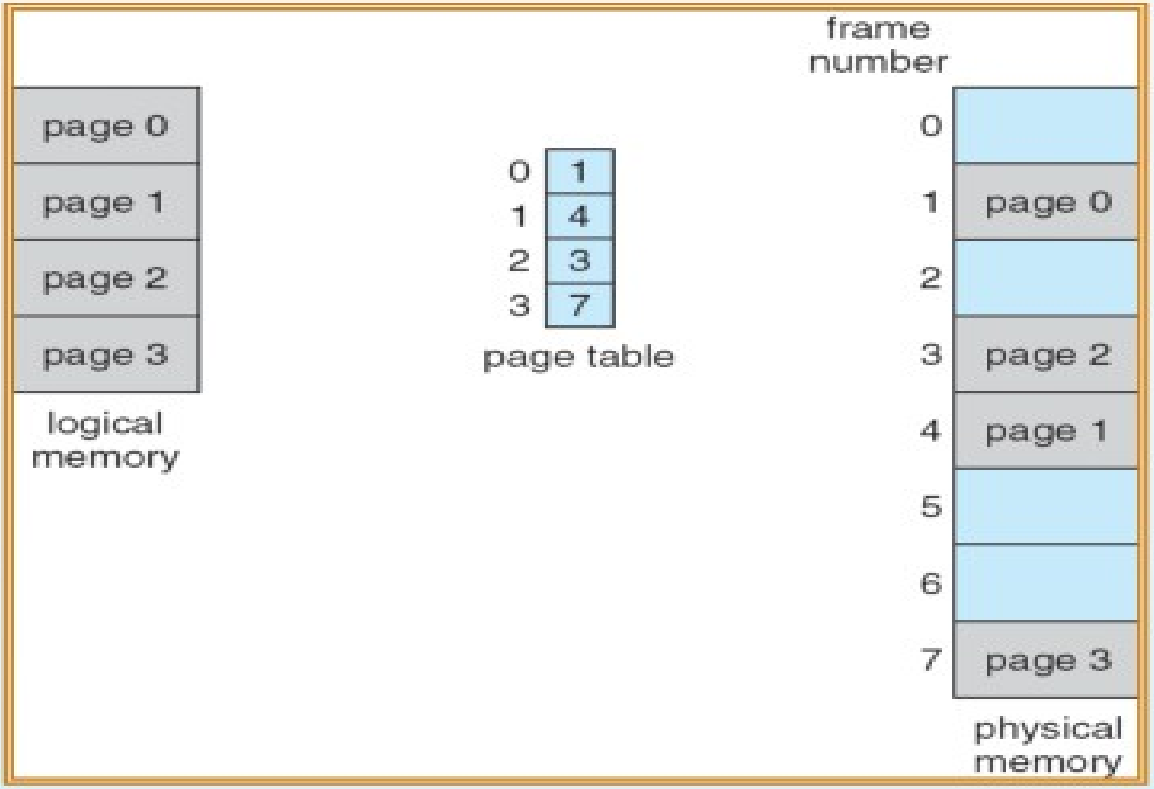
 Page offset (d) – combined with base address to define

the physical memory address that is sent to the memory

Address Translation Architecture



Paging Example



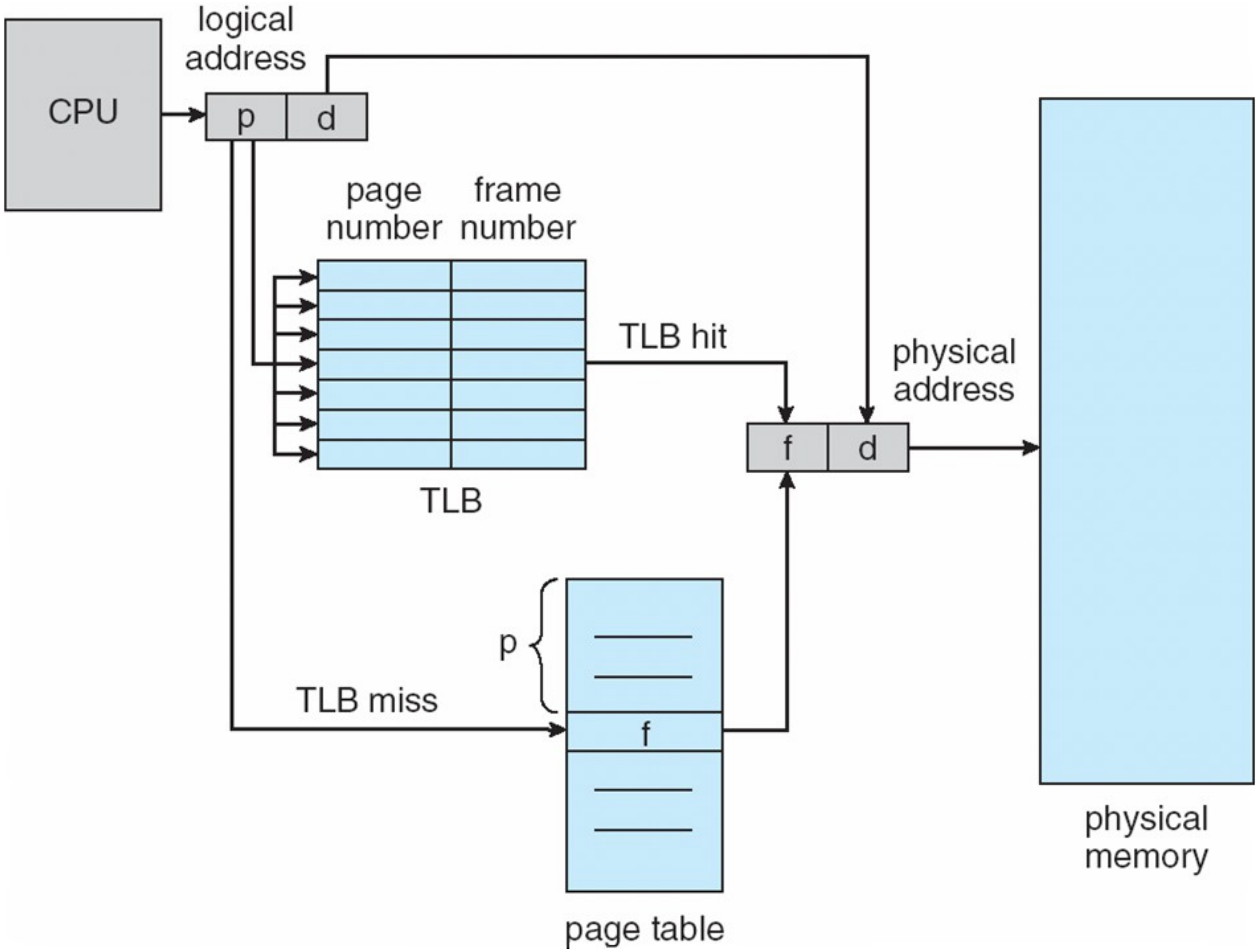
Implementation of Page Table

- Page table is kept in main memory
- *Page-table base register* (PTBR) points to the page table
- *Page-table length register* (PRLR) 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 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)**

TLB

- A translation lookaside buffer (**TLB**) is a memory cache that is used to reduce the time taken to access a user memory location. It is a part of the chip's memory-management unit (MMU).
- It stores the page number and corresponding frame no which are frequently used.

Paging Hardware With TLB



Hit Ratio

- The percentage value of the number of times a page is found in the TLB is called the hit ration.

Memory protection

- Memory protection implemented by associating protection bit with each frame to indicate if read-only or read-write access is allowed
- **Valid-invalid** bit attached to each entry in the page table:
  - “valid” indicates that the associated page is in the logical address space.
  - “invalid” indicates that the page is not in the logical address space.

Advantages of paging

- Paging supports time sharing system.
- It avoids external fragmentation.
- Compaction overheads are eliminated.
- Sharing of common code are possible.

Disadvantages of paging

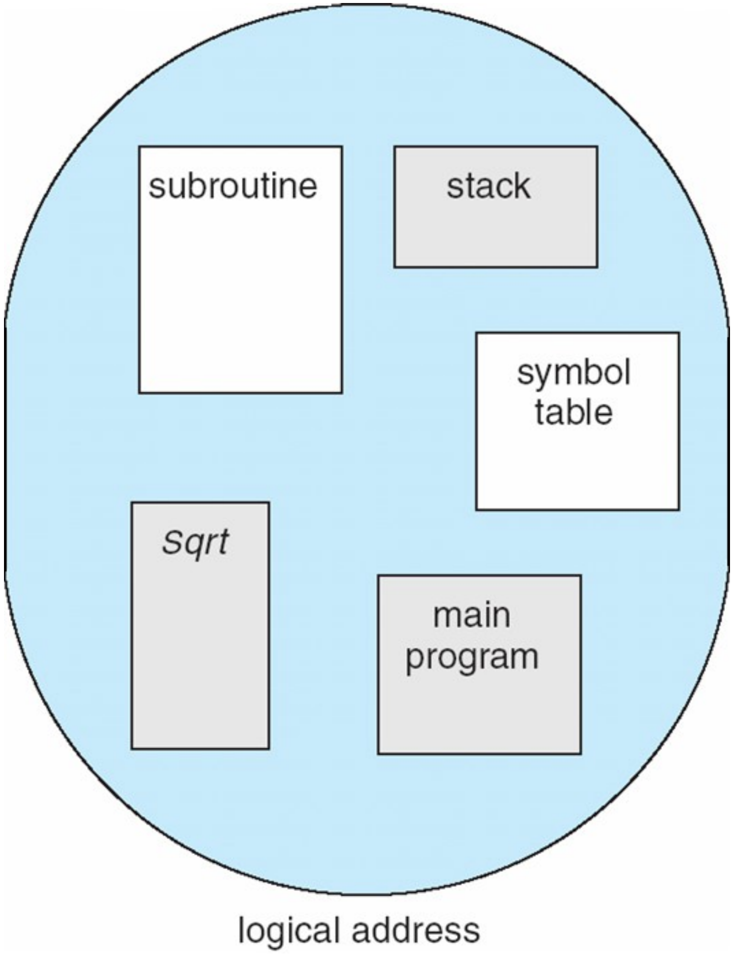
- When number of pages are large, it is difficult to maintain page tables.
- Hardware cost is high.

SEGMENTATION

The concept of Segmentation

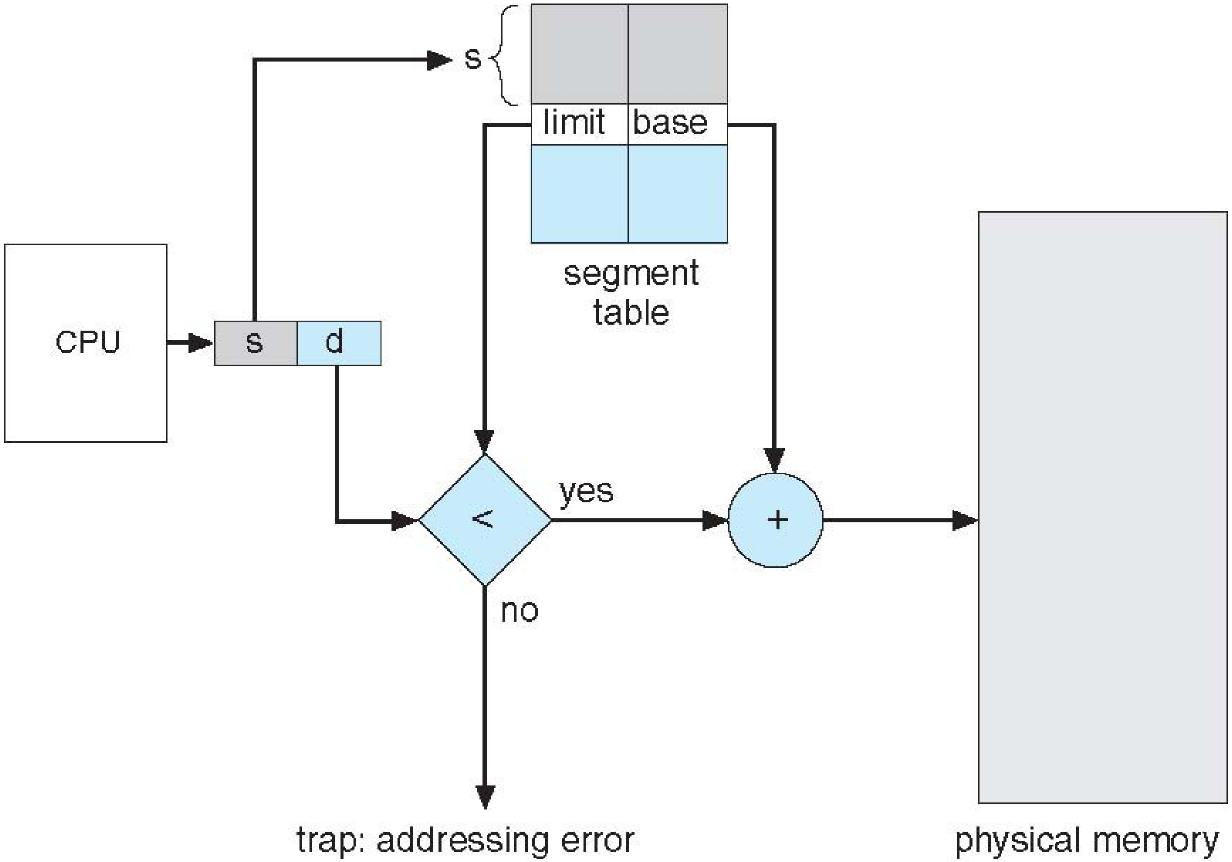
- Memory-management scheme that supports user view of memory
- A program is a collection of segments
  - A segment is a

User view of the program



- Each of the segments are of different length.
- The segments are numbered and are referred to by a segment number.
- The logical address of the segment consists of segment number and an offset.
- Each process have separate segment table.
- Each segment table consists of segment no, its length, base address and access rights.

Segmentation hardware



Segmentation Hardware

- Logical address consists of a two tuple:

  <segment-number, offset>,

- Each segment table consists of segment no, its length, base address and access rights.

- **Segment-table base register (STBR)** points to the segment table’s location in memory

Protection for segmentation

- Protection
  - With each entry in segment table associate:
    - validation bit = 0 ⇒ illegal segment
    - read/write/execute privileges
- Protection bits associated with segments; code sharing occurs at segment level

Advantages of Segmentation

- Eliminates Fragmentation
- Protection mechanism is good
- Allows shared segments among users
- Dynamic loading and linking of segments is possible
- Allow dynamic growing of segments
- Supports modular programming

Explain the Difference between paging and segmentation

Paging	segmentation
The main memory is divided into fixed size partitions called pages	The main memory is divided into variable sized partitions called segments
OS maintains a page map table	OS maintains a segment table
Page table consists of two parts - page number and frame number	Segment table consists of three parts segment number , base address and limit
Processes use the page number and offset to calculate physical address	Processes use segment number and offset to calculate physical address
It does not support modular programming	It supports modular programming
It does not support dynamic growth of pages	It supports dynamic growth of segments