# *Virtual Memory*

- To facilitate the use of memory hierarchies, the memory addresses normally generated by modern processors executing application programs are not *physical addresses*, but are rather *virtual addresses* of data items and instructions.
- Physical addresses, of course, are used to reference the available locations in the real physical memory of a system.
- Virtual addresses must be mapped to physical addresses before they can be used.

# *Virtual to Physical Mapping*

- The mapping from virtual to physical addresses can be formally defined as follows:

$$f_t v = \begin{cases} m, & \text{if } m \in M \text{ has been allocated to store} \\ & \text{the data identified by virtual address } m \\ \varnothing & \text{if data } v \textit{ is} \text{ missing in } M \end{cases}$$

- The mapping returns a physical address if a *memory hit* occurs. If there is a *memory miss*, the referenced item has not yet been brought into primary memory.

# *Mapping Efficiency*

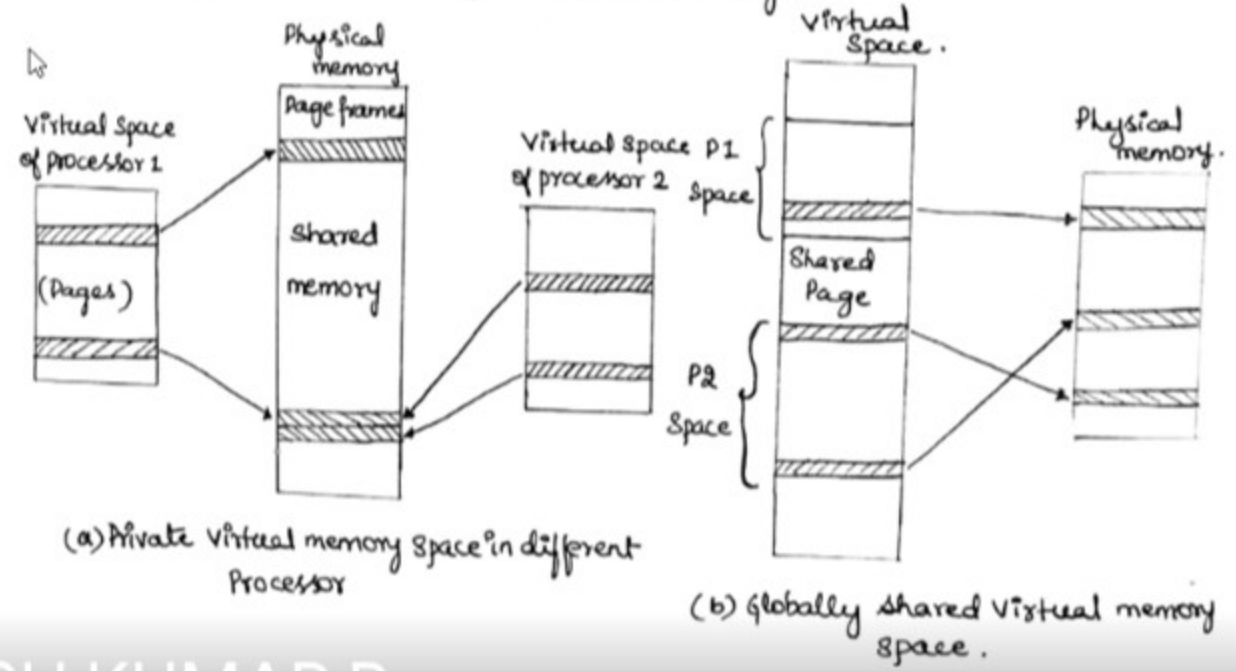
- The efficiency with which the virtual to physical mapping can be accomplished significantly affects the performance of the system.
- Efficient implementations are more difficult in multiprocessor systems where additional problems such as coherence, protection, and consistency must be addressed.

# *Virtual Memory Models (1)*

- Private Virtual Memory
  - In this scheme, each processor has a separate virtual address space, but all processors share the same physical address space.
  - Advantages:
    - Small processor address space
    - Protection on a per-page or per-process basis
    - Private memory maps, which require no locking
  - Disadvantages
    - The synonym problem – different virtual addresses in different/same virtual spaces point to the same physical page
    - The same virtual address in different virtual spaces may point to different pages in physical memory

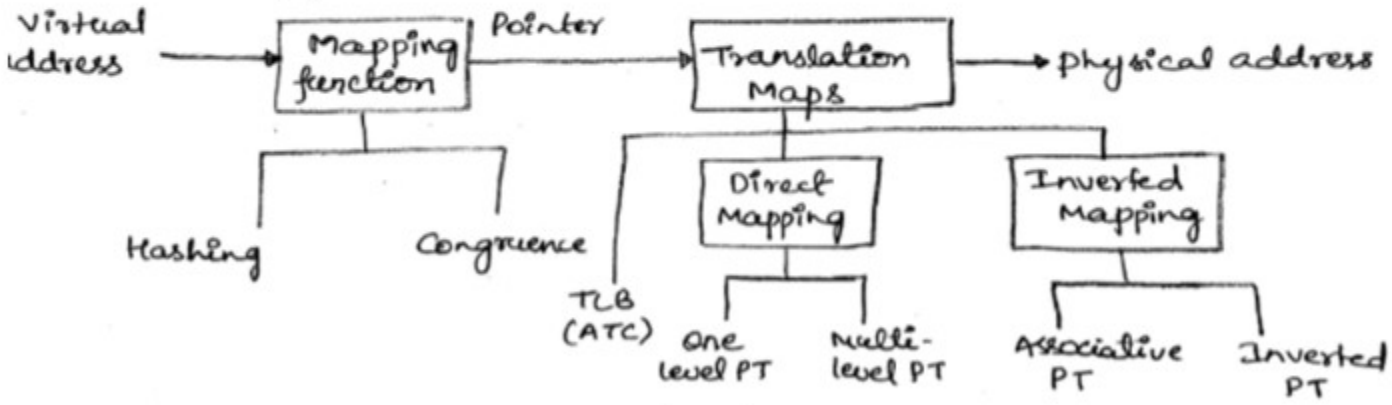
# VIRTUAL MEMORY MODEL

→ Address translation process is longer.



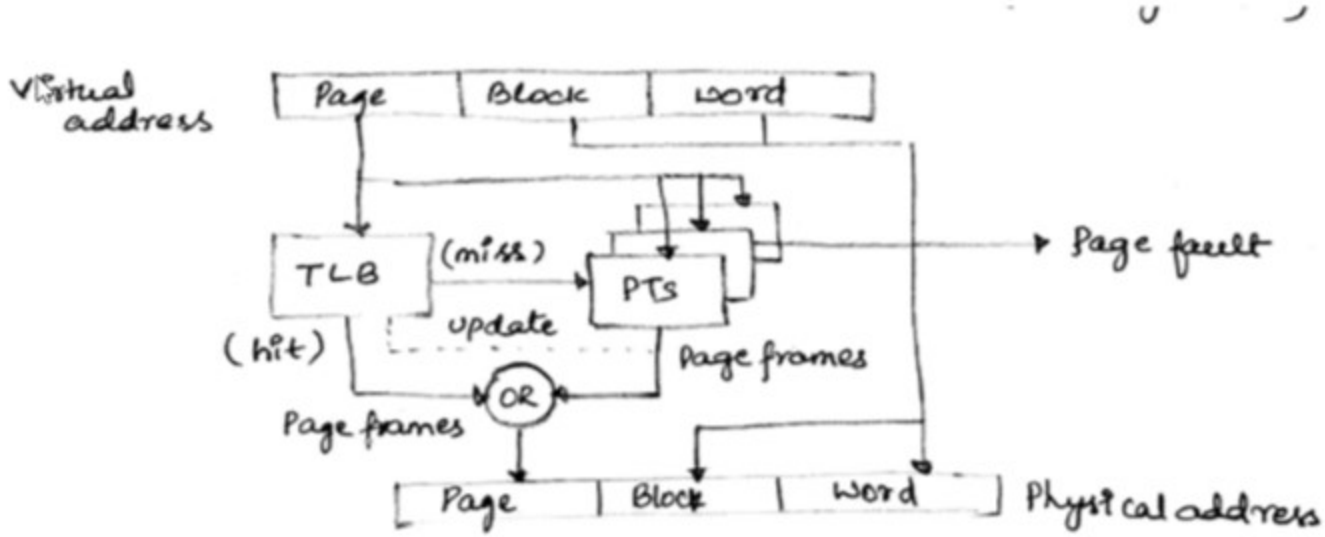
# TLB , PAGING

Various schemes for virtual address translation are summerized below in figure C(a).




(a) Virtual address translation schemes (PT = Page table)

# TLB , PAGING




(b) Use of a TLB and PTs for address translation