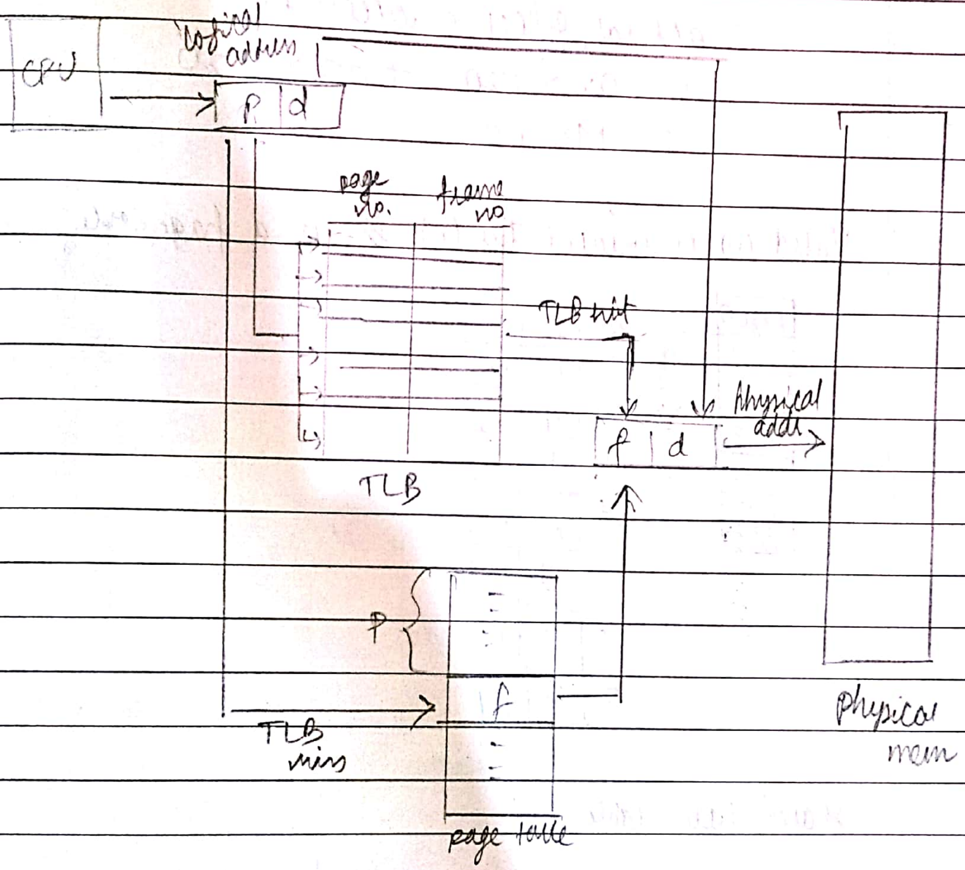
Lecture-22
BDu

Date 22/03/18
Page

Implementation of page table

- Page table is kept in the 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.



Q. Consider a paging system with the page table stored in memory. If memory access (reference) is 100 ns. a) How long does a paged memory reference takes b) If we add TLB and 80% of all page table references are found in the TLB. What is the effective memory reference time. Assume that finding a ~~the~~ page table entry in the TLB takes 20 ~~unit~~ ns.

a)
Memory access time = 2×100 ns = 200 ns (as we need two memory reference)

~~b) Memory access time~~

~~$= 100 + \frac{80}{100} \times 20 + \frac{20}{100} \times 100$~~

~~$= 100 + 16 + 20 = 136$ ns~~

Alternatively

effective access time with TLB hit

100 + 20 = 120 ns → 80%

effective access time with TLB miss

100 + 100 + 20 = 220ns → 80%.

(average)

∴ actual effective access time

= 0.8 × 120 + 0.2 × 220

= 140 ns

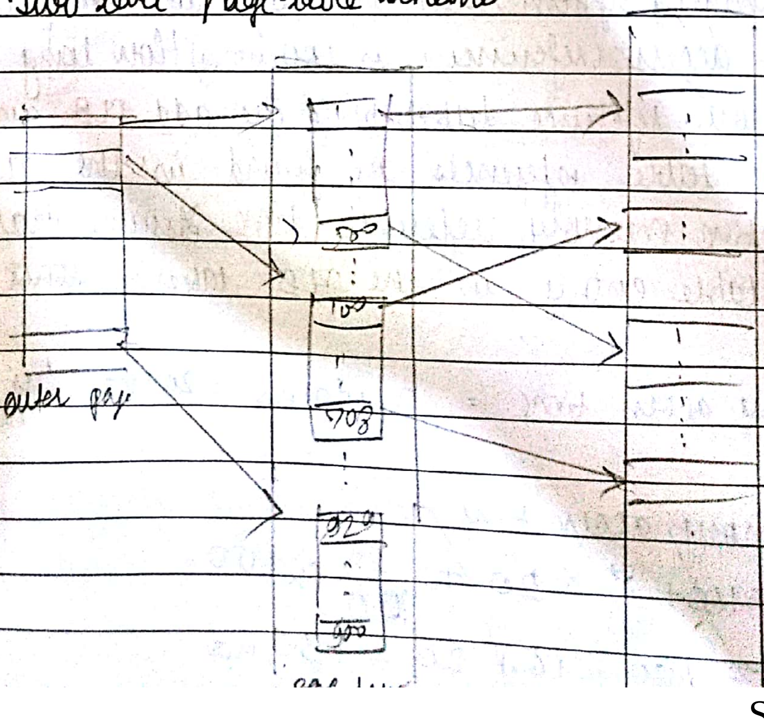
Valid (v) or invalid bit (i) ~~b~~ in a page table

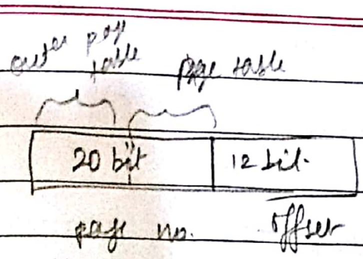
|  |
|---|
| page 0 |
| " 1 |
| " 2 |
| " 3 |
| " 4 |
| " 5 |
| [illegible] |

|  |  |  |
|---|---|---|
| 0 | 2 | v |
| 1 | 3 | v |
| 2 | 4 | v |
| 3 | 7 | v |
| 4 | 8 | v |
| 5 | 9 | v |
| 6 | 0 | i |
| 7 | 0 | i |

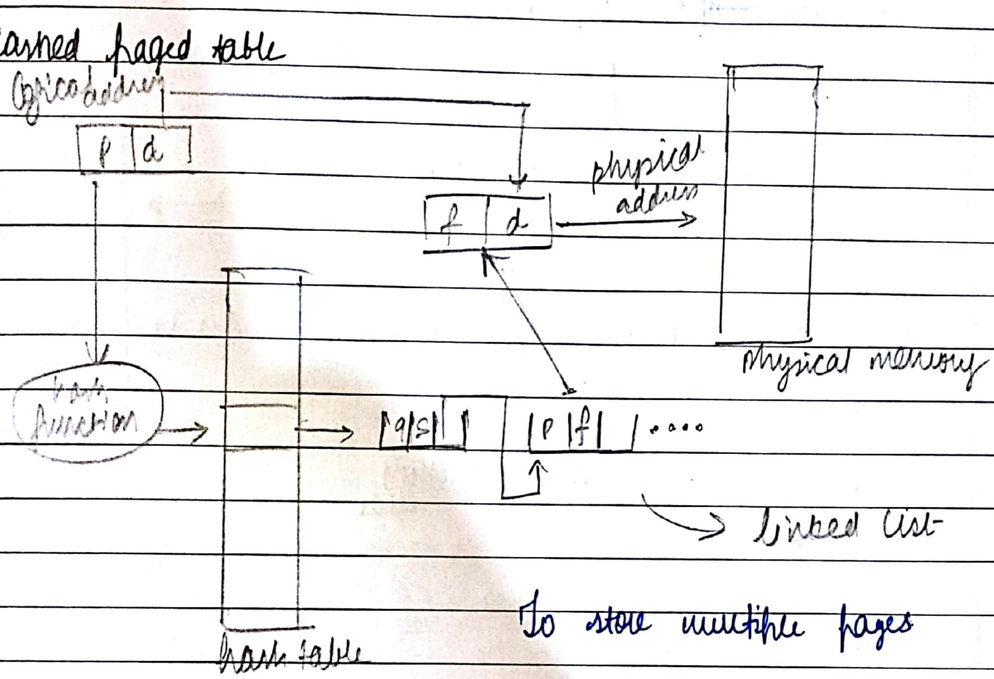
shared page table

Two level page-table scheme



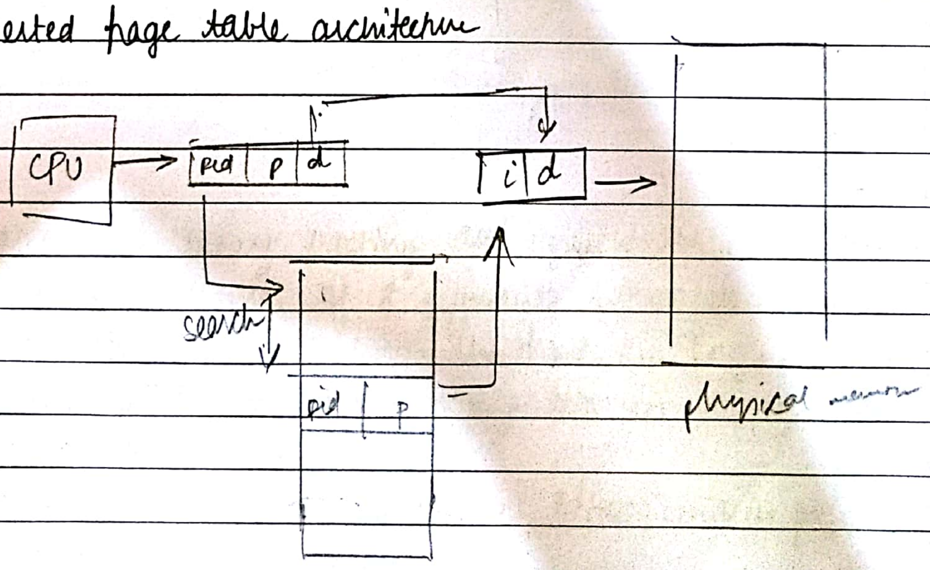


## Hashed paged table

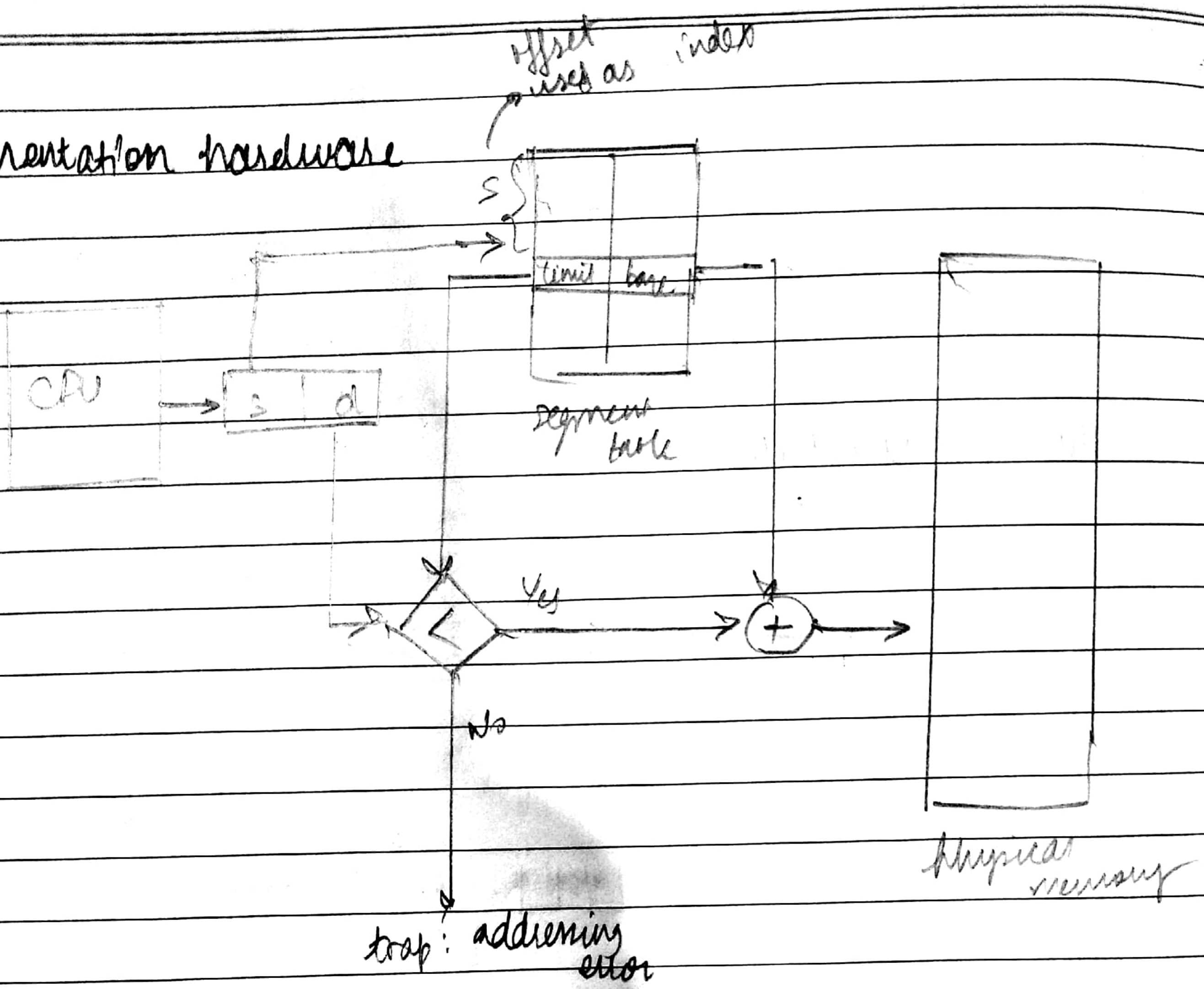


To store multiple pages

## Inverted page table architecture



## Segmentation hardware


offset used as index
s
limit
base
segment table
CPU
s
d
Yes
No
trap: addressing error
physical memory