CS & IT ENGINEERING

Operating System

Memory Management

Lecture -3

By- Vishvadeep Gothi sir

Recap of Previous Lecture

Topic Non-Contiguous MMT

Topic Paging

Topic Page Table

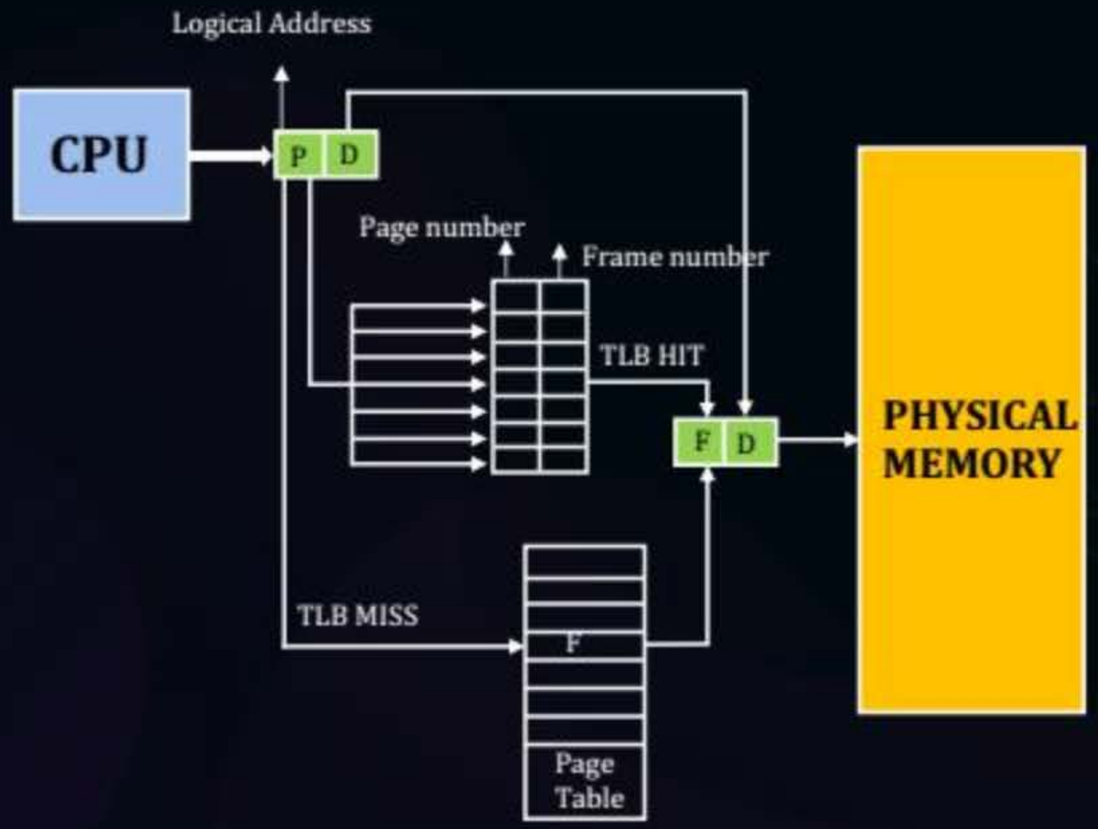
Topics to be Covered

Topic TLB Mapping

Topic Segmentation

Topic : Paging

TLB (Translation Lookaside Buffer)



Topic : Paging

How TLB Stores Entries?

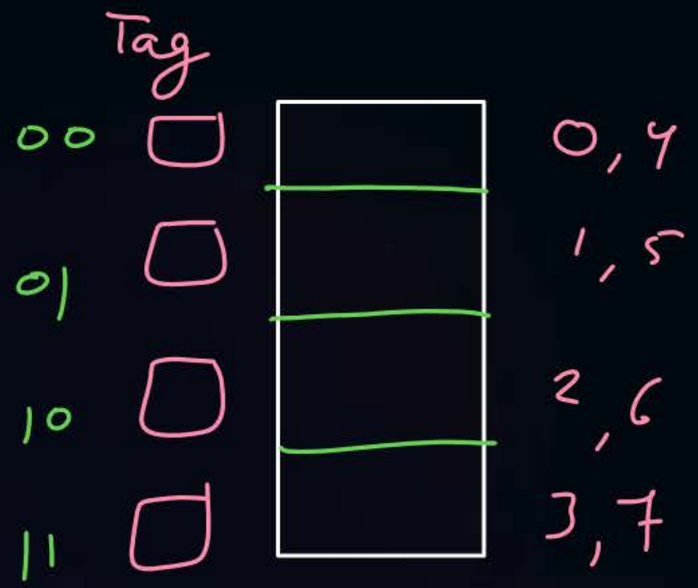
Topic : Paging

TLB Mapping

1. Fully Associative
2. Direct
3. Set-Associative

Topic : Paging

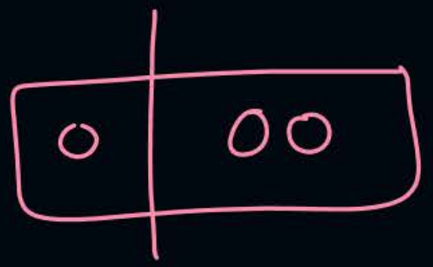
TLB Mapping: Direct



Page no.

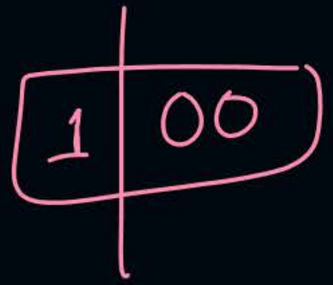
Tag	TLB entry no.

ex :- Page no = 000



no. of bits in TLB entry no. $= \log_2$ (no. of entries in TLB)

Page no. = 100

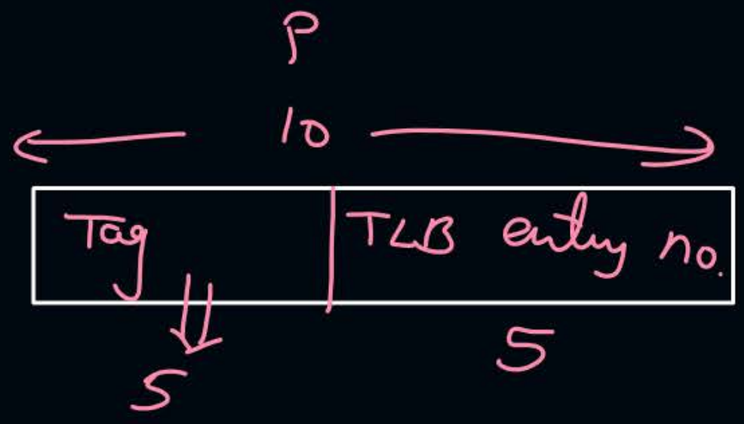


ex:- no. of pages $= 2^{10} \Rightarrow p = 10$ bits

no. of entries in TLB $= 32 = 2^5$

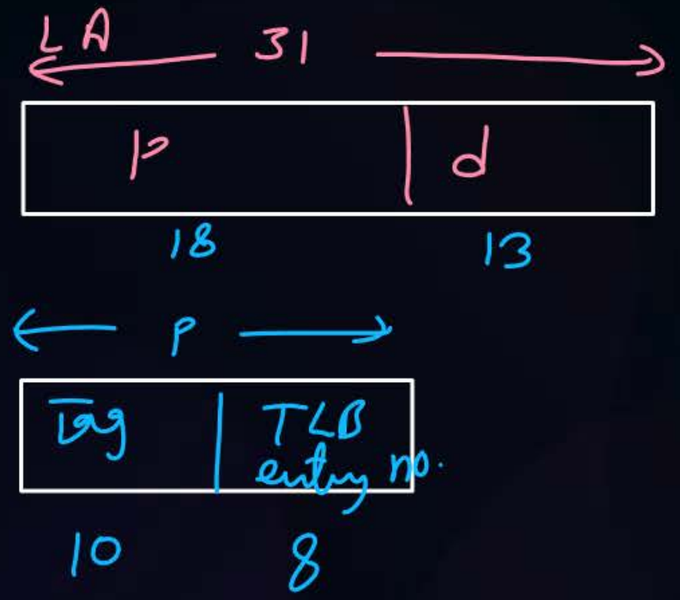
Tag bits = 5 ?

Direct mapping



Topic : Question

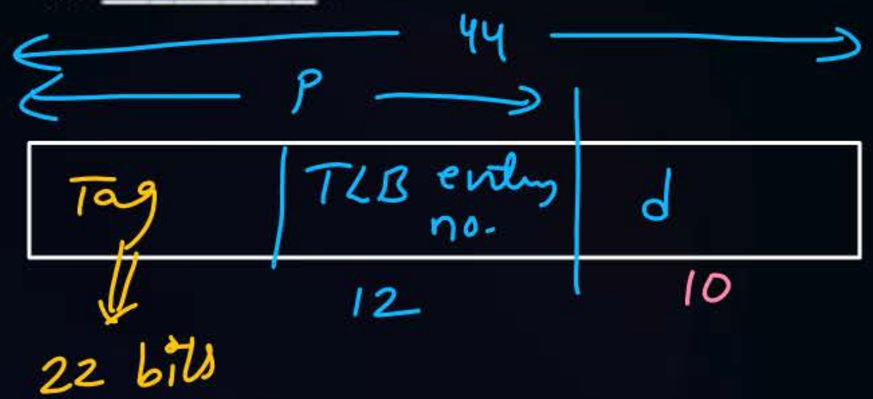
#Q. A computer system implements a 31- bit virtual address, page size of 8 kilobytes, and a 256-entry translation look-aside buffer (TLB) organized as direct mapped. The minimum length of the TLB tag in bits is _____?

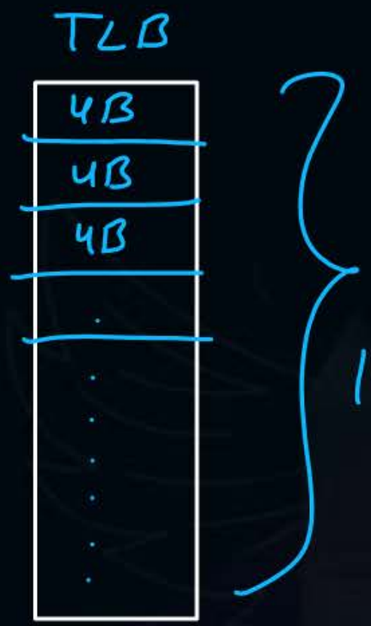


Topic : Question

d = 10 bits ⇐ 2^{10}B ← (1)

#Q. A computer system implements a 44- bit virtual address, page size of (1) kilobytes, and a 16KB look-aside buffer (TLB) organized as direct mapped. Each page table entry is of 4bytes. The minimum length of the TLB tag in bits is 22 ?





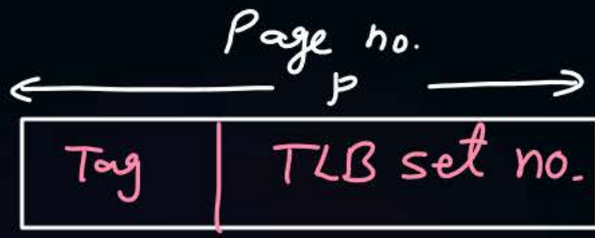
no. of entries in TLB = $\frac{16KB}{4B}$

$= 4K$

$= 2^{12}$

Topic : Paging

TLB Mapping: Set Associative



$$\text{no. of TLB set no.} = \log_2(\text{no. of sets in TLB})$$

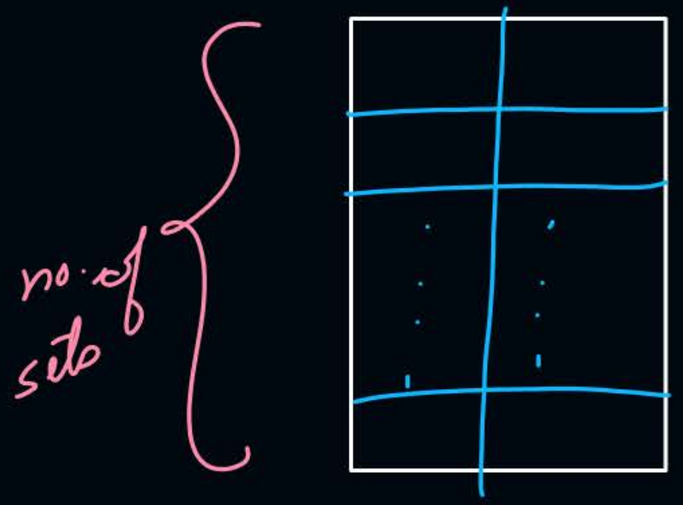
$$\text{no. of sets in TLB} = \frac{\text{no. of entries in TLB}}{\text{associativity}}$$

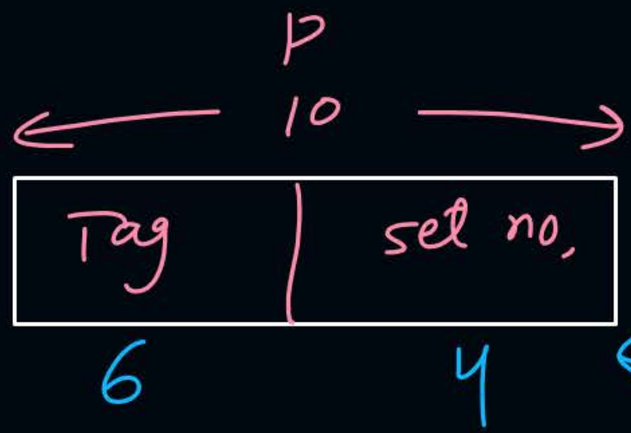
Ex:- no. of pages = 2^{10} $\Rightarrow$ p = 10 bits

no. of entries in TLB = 32

2 - way set associative TLB

no. of sets = $\frac{32}{2} = 16 = 2^4$

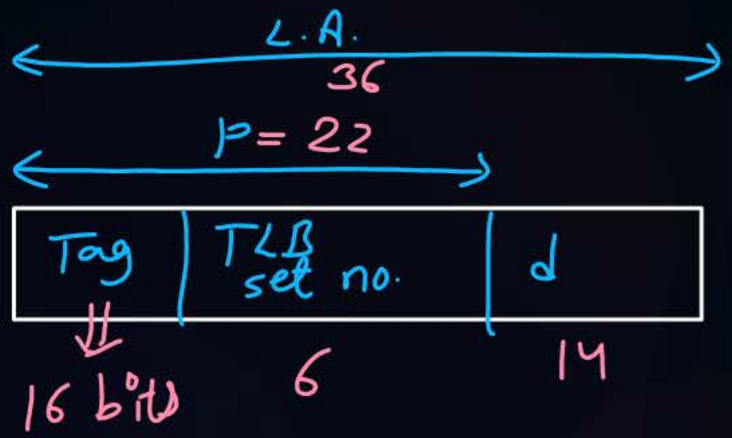




Topic : Question

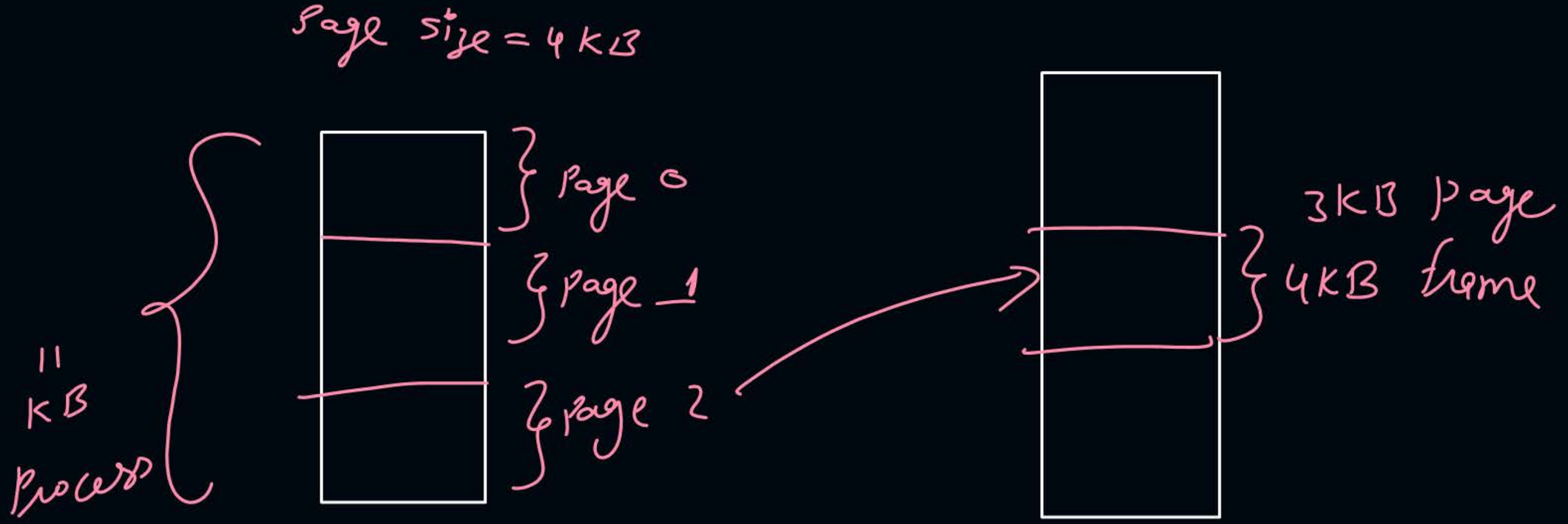
2^{14} B

#Q. A Computer system implements a 36-bit virtual address, page size of 16 Kbytes and a 256 – entry translation look-aside buffer (TLB) organized into 64 sets each having four ways. Assume that the TLB tag does not store any process id. The minimum length of the TLB tag in bits is __16__.



no. of sets = $64 = 2^6$

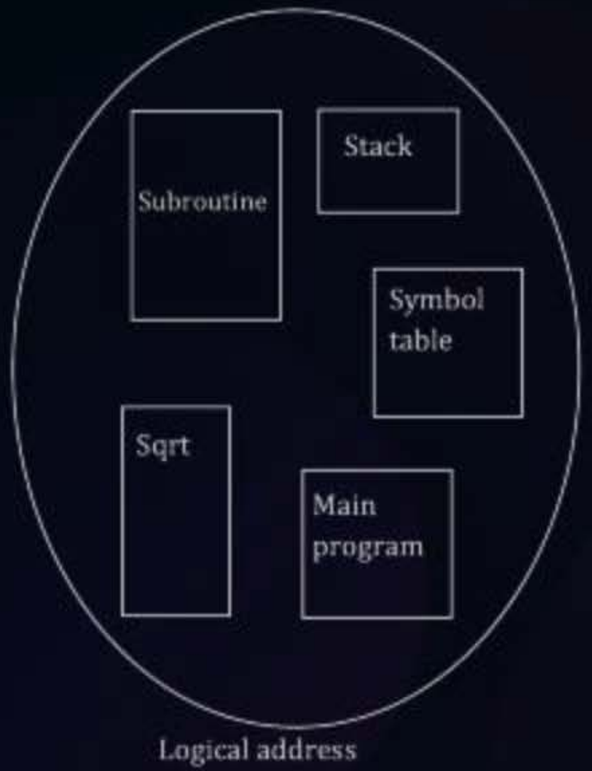
Paging suffers from internal Fragmentation



Topic : Segmentation

- Divide Process in logically related partitions (Segments)
- Segments are scattered in physical memory

(Segments) → variable sizes



Topic : Segmentation

Process

seg 0
seg 1
seg 2
seg 3

Segment Table

	Base address	limit
00	9000	500
01	1000	300
10	12000	400
11	5000	800

addresses

Physical Memory

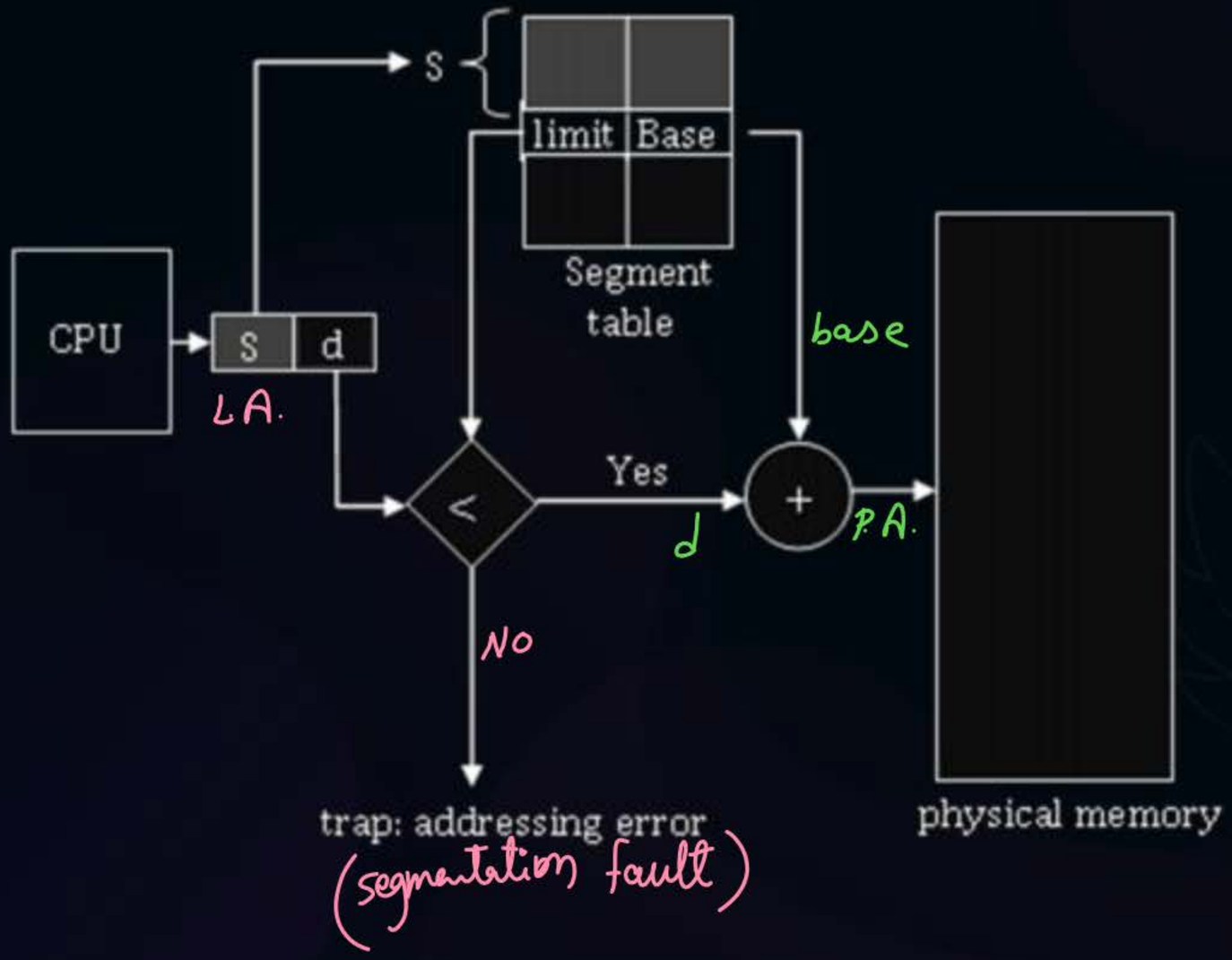
S = 2

d = 249

P.A. = 12000 + 249

12249

Topic : Segmentation



Topic : Segmentation

Segment Table

00	9000	500
01	1000	300
10	12000	400
11	5000	800

Segment number	Offset	Physical Address
1	204	1000 + 204 = 1204
0	449	9000 + 449 = 9449
2	436	fault
3	737	5000 + 737 = 5737

Topic : Segmentation

- Size of segment can vary, so along with base, keep limit information also
- Limit defines max number of words within the segment

or bytes

---

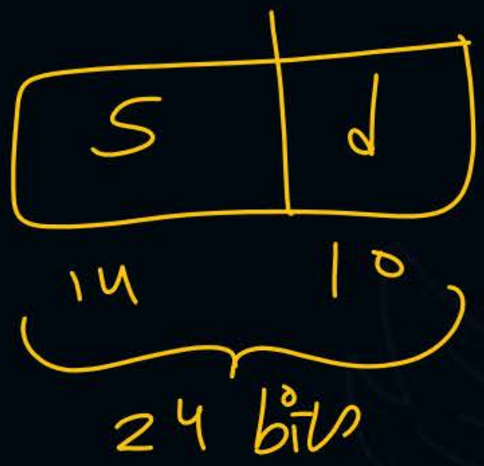
no. of bits in d decided based on size of max possible segment.

[NAT]

#Q. Maximum segment size = 16KB $= 2^{14}$ B $\Rightarrow$ d = 14 bits

Number of segments in process = 2^{10} = s = 10 bits

Logical address = 24 bits ??

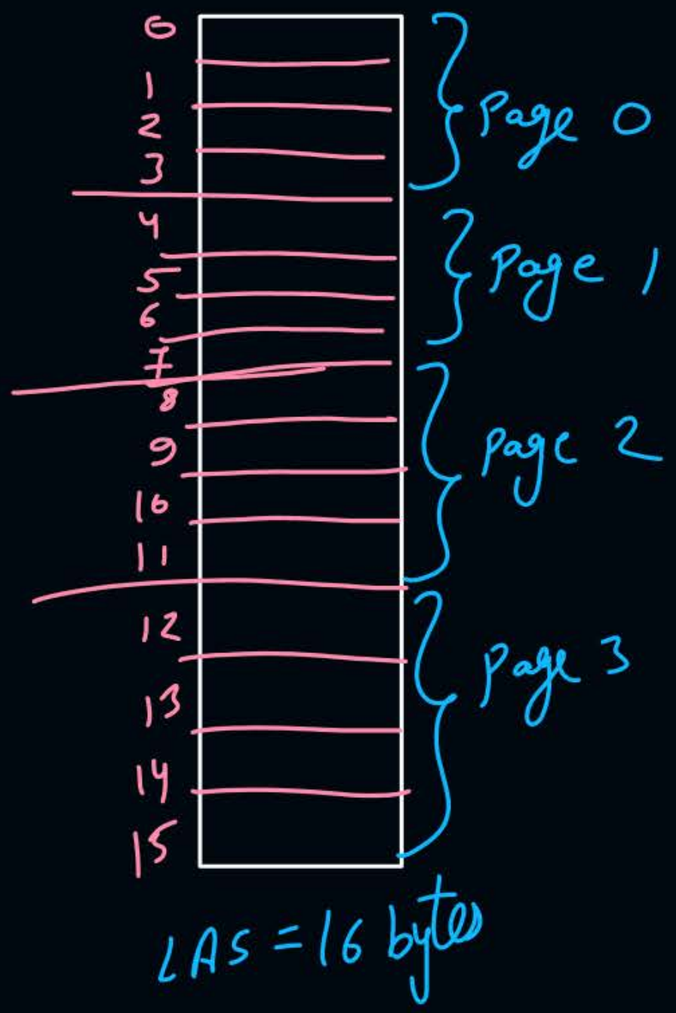


Topic : Segmentation

- Segmentation suffers from external fragmentation

L.A. to P.A. translation for decimal in paging :-

ex: Page size = 4 bytes



$$\text{Page no.} = \left\lfloor \frac{LA}{\text{Page size}} \right\rfloor = \left\lfloor \frac{14}{4} \right\rfloor = 3$$

$$d = LA \% \text{ page size} = 14 \% 4 = 2$$

$$LA = (\text{Page no} * \text{page size}) + d$$

$$(3 * 4) + 2$$

$$= 14$$

$$f = \left\lfloor \frac{P.A.}{\text{Page size}} \right\rfloor$$

$$d = P.A. \ \% \ \text{page size}$$

$$P.A. = (f * \text{page size}) + d$$

Ques) Page size = 64 B

L.A.S. = 256 B = 2^8

L.A. = 147

P.A. = ?

Page Table

0	5
1	3
2	1
3	7

$f = 1$

Soln

$p = \frac{147}{64} = 2$

$d = 147 \% 64 = 19$

$P.A. = (1 * 64) + 19$

$= 83$

8 bits

p	d
2	6

$(147)_{10} =$ | 10 | 010011 |

p d

p. 1. $\Rightarrow f = 1$

f

1	010011	$\rightarrow (83)_{10}$

(GATE-2024)

Ques) page size = 2kB

Pages 0, 1, 2, 3 are stored on 1, 3, 2, 0 frames in main memory.

$L.A. = (2500)_{10}$

$P.A. = (6596)_{10}$?

	PT
0	1
1	3
2	2
3	0

Soln

$$p = \left\lfloor \frac{2500}{2048} \right\rfloor = 1 \Rightarrow P.T. \Rightarrow f = 3$$

$$d = 2500 \% 2048 = 452$$

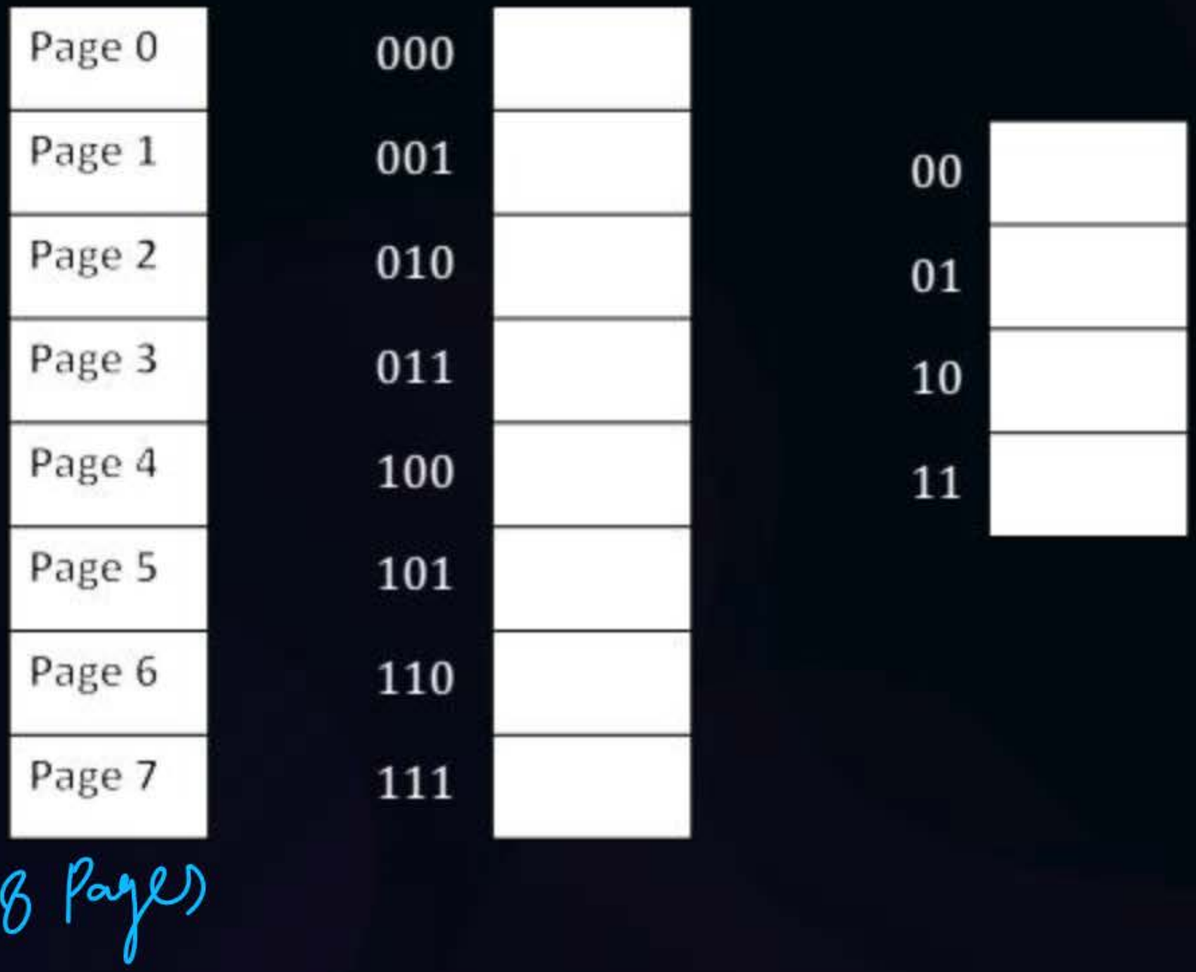
$$P.A. = (3 * 2048) + 452$$

$$= \underline{6596}$$

Topic : Virtual Memory

- Feature of OS
- Enables to run larger process with smaller available memory

Topic : Virtual Memory



Topic : Virtual Memory

Process

Page 0	
Page 1	
Page 2	
Page 3	
Page 4	
Page 5	
Page 6	
Page 7	

8 Pages

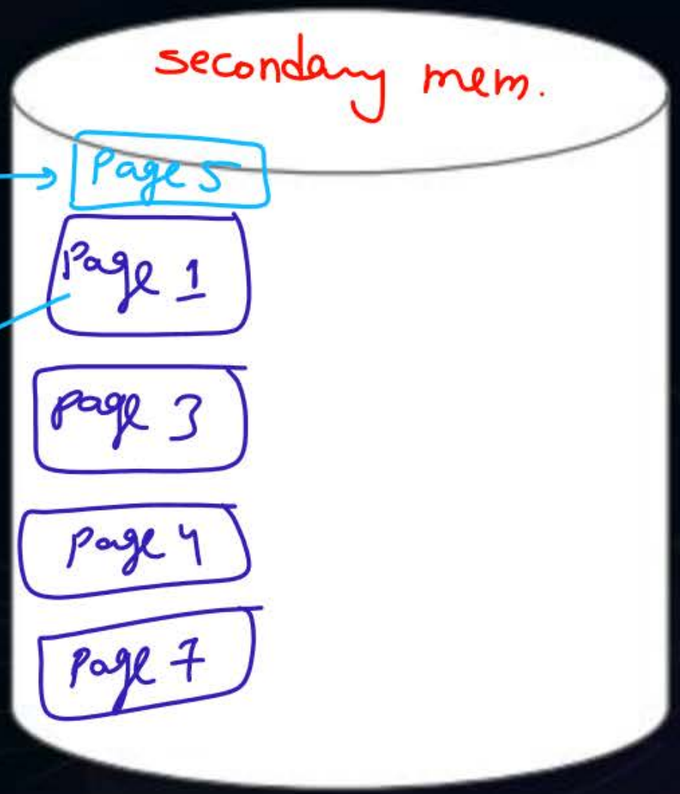
Page table

000	01
001	10
010	00
011	
100	
101	~~10~~
110	11
111	

mm.

00	Page 2
01	Page 0
10	Page 5
11	Page 6

4 frames



Topic : Demand Paging

- **Demand Paging:**

  Bring pages in memory when CPU demands

- **Page Fault:**

  When the demanded page is not available in physical memory

↳ OS provides page fault servica by bringing demanded page into mm.

↳ After servica the instn which caused page fault will start.

Pure demand paging :-

mm frames are empty when process starts.
and bring pages in mm only on demand.

2 mins Summary

Topic TLB Mapping

Topic Segmentation

Happy Learning

THANK - YOU