CS & IT ENGINEERING

Operating System

Memory Management

Lecture -2

By- Vishvadeep Gothi sir

Recap of Previous Lecture

Topic Memory Management

Topic Memory Management Technique

Topic Contiguous Memory Management Technique

Topics to be Covered

Topic Non-Contiguous MMT

Topic Paging

Topic Page Table

Topic : Paging

- Process is divided in equal size of partitions called as pages
- Physical memory is divided in same size of partitions called as frames
- Pages are scattered in frames
- CPU always generates logical address

Topic : Paging

Consider

- A process has 4 pages
- Main memory has 8 frames

Process

00	
01	
10	
11	

Page table

00	011
01	101
10	010
11	111

mm

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

Topic : Paging

logical add — 4 Pages

Process

logical add	Process	Page
000	a	00
001	b	
010	c	01
011	d	
100	e	10
101	f	
110	g	11
111	h	

Page Table

00	011
01	101
10	010
11	111

addresses — 8 frames

Physical Memory

addresses	Physical Memory	frame
0000		frame 000
0001		
0010		001
0011		
0100	e	010
0101	f	
0110	a	011
0111	b	
1000		100
1001		
1010	c	101
1011	d	
1100		110
1101		
1110	g	111
1111	h	

Page size = 2 bytes

Process size $= 4 * 2B = 8B$

→ logical address space

mm size $= 8 * 2B$

$= 16B = 2^4 B$

$\Downarrow$

address = 4 bits

→ physical add. space

L.A.

p	d
page no.	byte no. (line no. or offset)

$$\text{no. of bits needed for } d = \log_2(\text{page size in bytes})$$

$$\text{no. of bits needed for } p = \log_2(\text{no. of pages in process})$$

$$\text{no. of pages in process} = \frac{L.A.S.}{\text{Page size}}$$

P.A.

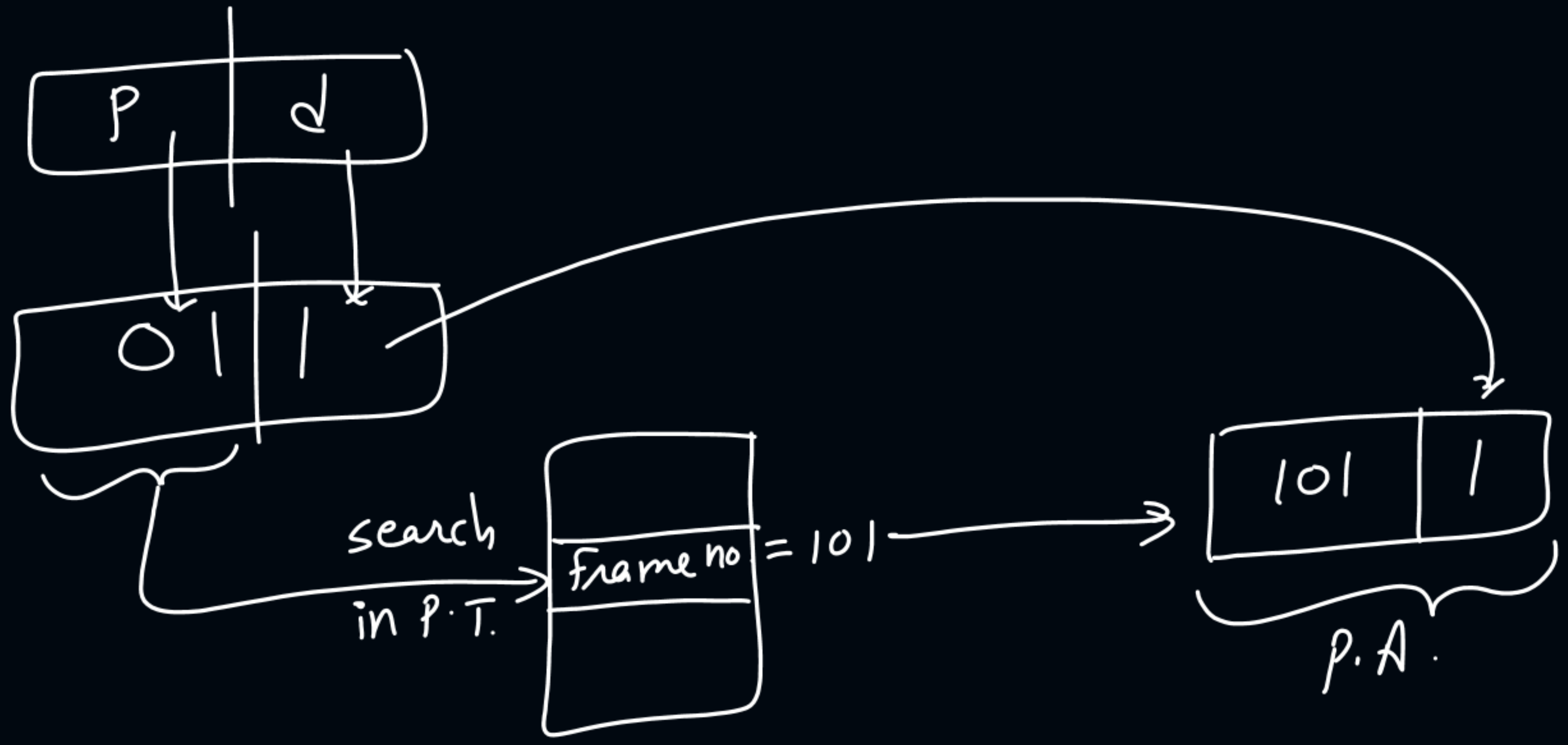
f	d
frame no.	byte no. (line no. or offset)

no. of bits needed for d = $\log_2$(page size in bytes)

no. of bits needed for f = $\log_2$(no. of frames in mm)

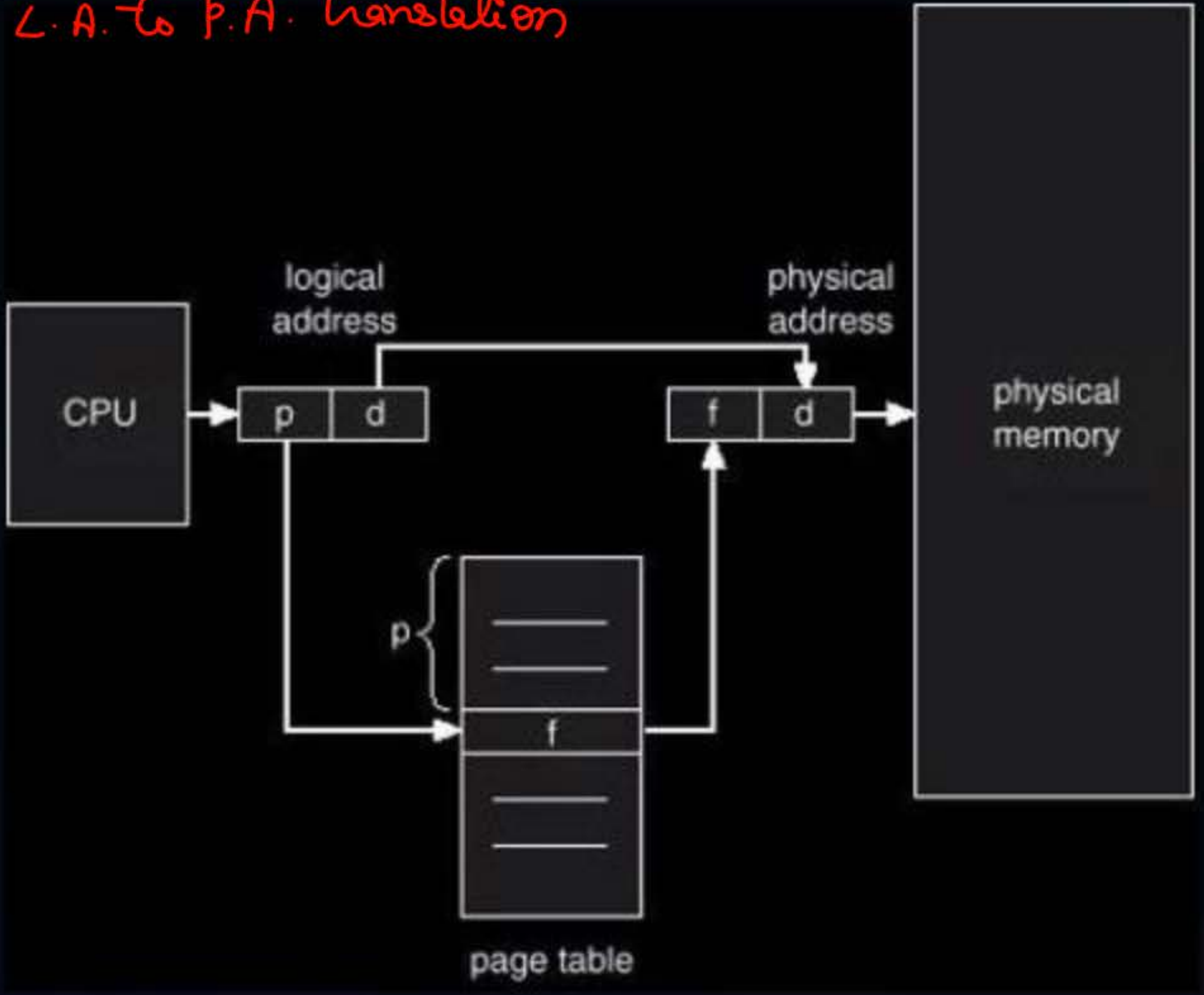
no. of frames in mm $= \dfrac{\text{P.A.S.}}{\text{Page size}}$

CPU generates L.A. 011



Topic : Paging

L.A. to P.A. translation



Topic : Paging

- Processor will have a view of process and its pages
- Page table is used to map a process page to a physical frame
- Number of entries in page table = Number of pages in process
- OS maintains a page table for each process

Topic : Question

#Q. Consider a paged memory system where the logical address of 29 bits and physical address of 35 bits. If each page table entry is of 4 bytes and page size is 1KB then:

Page size $= 1KB = 2^{10}B \Rightarrow d = 10$ bits

1. Number of pages in process? $= 2^{19}$
2. Number of frames in main memory? 2^{25}
3. Number of bits for page number? 19 bits
4. Number of bits for frame number? 25 bits
5. Number of entries in page table? 2^{19}
6. Page table size? $2^{19} * 4B = 2^{21}B = 2MB$

no. of pages in process $= \frac{2^{29} B}{1 KB} = 2^{19} \Rightarrow$ page no. $= 19$ bits

no. of frames in mm $= \frac{2^{35} B}{1 KB} = 2^{25} \Rightarrow$ frame no. $= 25$ bits

---

L.A.

$\longleftarrow 29 \longrightarrow$

p	d
19	10

P.A.

$\longleftarrow 35 \longrightarrow$

f	d
25	10

---

page table entry = frame no + extra bits

(4 bytes) 32 bits = 25 bits + extra bits

extra bits = 7 bits

Topic : Question

[GATE-2001]

#Q. Consider a machine with 64 MB physical memory and a 32-bit virtual address space. If the page size is 4 KB, what is the approximate size of the page table?

P.A. = 26 bits

2^{12} B $\Rightarrow$ d = 12 bits

A. 16 MB

B. 8 MB

C. ✓ 2 MB

D. 24 MB

L.A. ← 32 →

p	d
20	12

P.A. ← 26 →

f	d
14	12

P.T. size $= 2^{20} * 14$ bits

$= 14$ M bits

$\simeq 2$ MB

$L.A. = 24$ bits

#Q. Consider a paged memory system where the process size is 16MB and main memory size is 4GB. The page size is 2KB.

$\hookrightarrow P.A. = 32$ bits

$\hookrightarrow 2^{11} B$

$\hookrightarrow d = 11$ bits

A Number of pages in process? 2^{13}

B Number of frames in main memory? 2^{21}

C Number of bits for page number? 13 bits

D Number of bits for frame number? 21 bits

E Number of entries in page table? 2^{13}

F Page table size? $\rightarrow 2^{13} * 21$ bits

L.A.

$\longleftarrow 24 \longrightarrow$

p	d
13	11

P.A.

$\longleftarrow 32 \longrightarrow$

f	d
21	11

$LA = 27$ bits

#Q. Consider a paged memory system where the process size is 128MB and main memory size is 2GB. The page size is 1KB.

$\to P.A = 31$ bits $\to d = 10$ bits

A. Number of pages in process? 2^{17}

B. Number of frames in main memory? 2^{21}

C. Number of bits for page number? 17

D. Number of bits for frame number? 21

E. Number of entries in page table? 2^{17}

F. Page table size? $2^{17} * 21$ bits

L.A. 27

p	d
17	10

P.A. 31

f	d
21	10

#Q. Consider a paged memory system where the logical address is 25 bits and physical address is 33 bits. The page size is 4KB.

$2^{12}B \Rightarrow d = 12$ bits

A Number of pages in process? 2^{13}

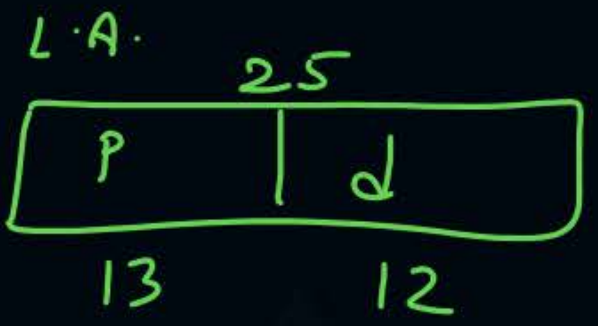
B Number of frames in main memory? 2^{21}

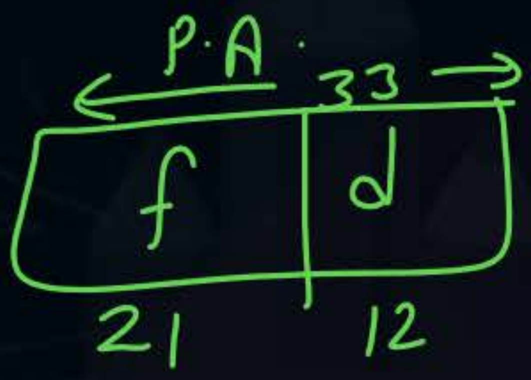
C Number of bits for page number? 13

D Number of bits for frame number? 21 bits

E Number of entries in page table? 2^{13}

F Page table size? $2^{13} * 21$ bits





Ques) P.T. size = 4 KBytes

P.T. entry = 4 B

no. of pages in process = ____ ?

Sol^n

4 KB = no. of pages * 4 B

no. of pages = 1K = 2^{10}

Ques) In prev. Quest^n if page size = 256 bytes → 2^8 B

d = 8 bits

then L.A. __18__ bits ?

Sol^n

L.A. 18

p	d
10	8

p = 10 bits

[MCQ] [GATE-2015]

#Q. A computer system implements 8 kilobyte pages and a 32–bit physical address space. Each page table entry contains a valid bit, a dirty bit, three permission bits, and the translation. If the maximum size of the page table of a process is 24 megabytes, the length of the logical address supported by the system is __36__ bits?

$\rightarrow d = 13$ bits

$24M\cancel{B} = \text{no. of pages} * 3\cancel{B}$

$\frac{24M}{3} = \text{no. of pages}$

no. of pages $= 8M$

$= 2^{23}$

$\Downarrow$

$p = 23$ bits

P.A. 32-bits

f	d
19	13

L.A.

p	d
23	13

36 bits

P.T. entry size $= 19 + 5$ bits

$= 24$ bits

$= 3$ bytes

Topic : Paging

Where the Page table Stored?

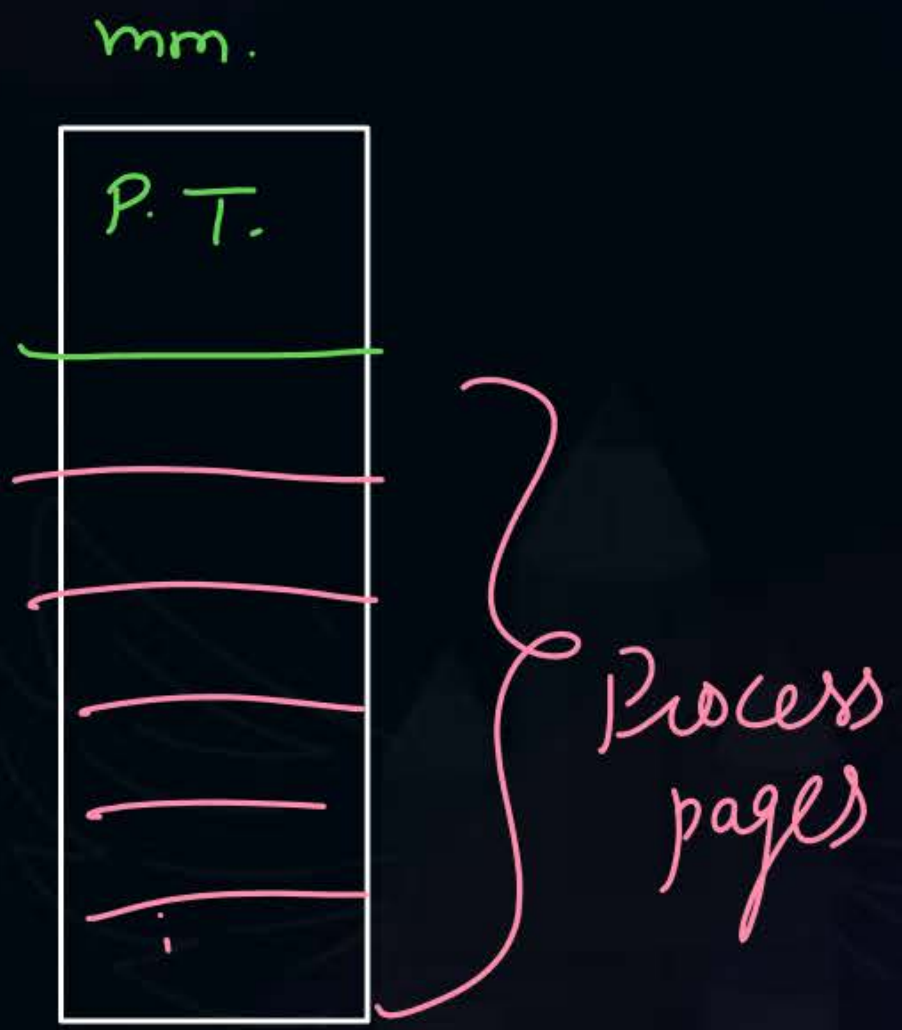
⇓

in mm.

→ L.A. → P.A. → Content

mm access for P.T.

mm access for Content



Performance of Paging

Effective mem. access time $= 2 * t_{mm}$

if P.T. is very small and can be stored in CPU registers

$$E.M.A.T. = t_{mm}$$

TLB (Translation Lookaside Buffer)

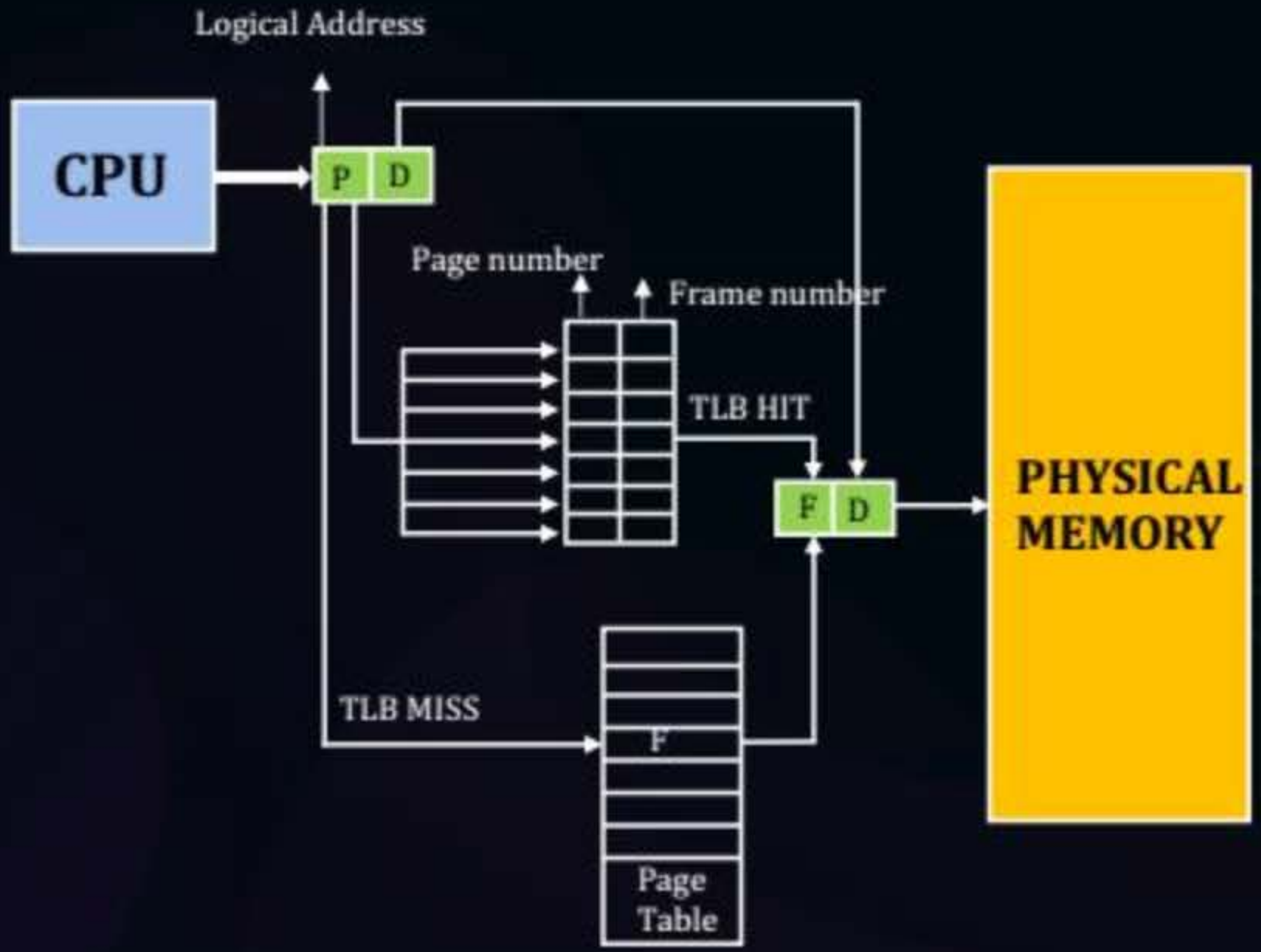
↓

a hardware used to improve performance of paging.

It stores recent and more frequently accessed P.T. entries

Topic : Paging

TLB (Translation Lookaside Buffer)



CPU access TLB with L.A.

hit

P.A.

↓

access mm

↓

content

miss

access mm for P.T.

↓

P.A.

↓

access mm

↓

content

using TLB

$$E.M.A.T. = H * (t_{TLB} + t_{mm}) + (1-H)\left[t_{TLB} + 2t_{mm}\right]$$

or

$$= t_{TLB} + t_{mm} + (1-H)\, t_{mm}$$

Ques)

$t_{TLB} = 20ns$

$t_{mm} = 500ns$

$H = 80\%$

E.M.A.T. = 620 ns?

Sol^n

$$\text{E.M.A.T. with TLB} = 20 + 500 + 0.2 * 500$$
$$= 620 \text{ ns}$$

$$\text{E.M.A.T. without TLB} = 2 * 500$$
$$= 1000 \text{ ns}$$

Ques)

In prev. quest^n, performance gain by using TLB as compared to not using TLB ______ ?

Sol^n

$$= \frac{1000 \text{ ns}}{620 \text{ ns}}$$

$$= 1.61$$

Ques) $t_{TLB} = 10ns$

EMAT with TLB = 500 ns

— " — w/o TLB = 900 ns

$H = \_\_\_\_\_\_$?

$900\ ns = 2\ t_{mm}$

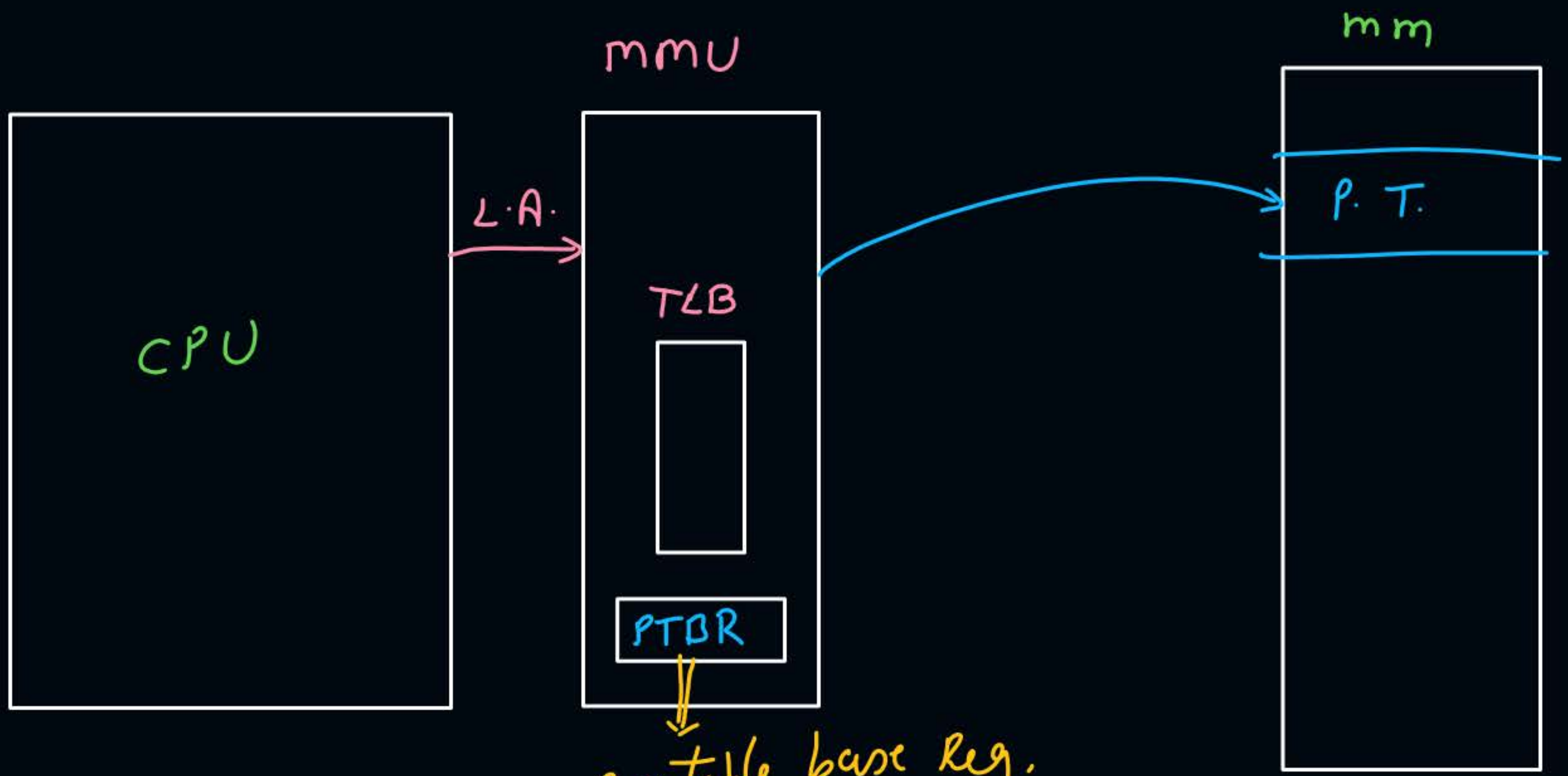
$t_{mm} = 450\ ns$

$500 = 10 + 450 + (1-H)\ 450$

$H = 0.9111$

$= 91.11\ \%$

Memory Management Unit (MMU) :- it Translates L.A. into P.A.



Page table base Reg.

⇓

stores starting add. of P.T. of current running process

TLB implementatⁿ

TLB stores only P.T. entry

⇓

when context swith happens all TLB entries are made invalid.

TLB stores process id with each P.T. entry

⇓

multiple processes' P.T. entries can be present in TLB at a time.

2 mins Summary

Topic Non-Contiguous MMT

Topic Paging

Topic Page Table

Happy Learning

THANK - YOU