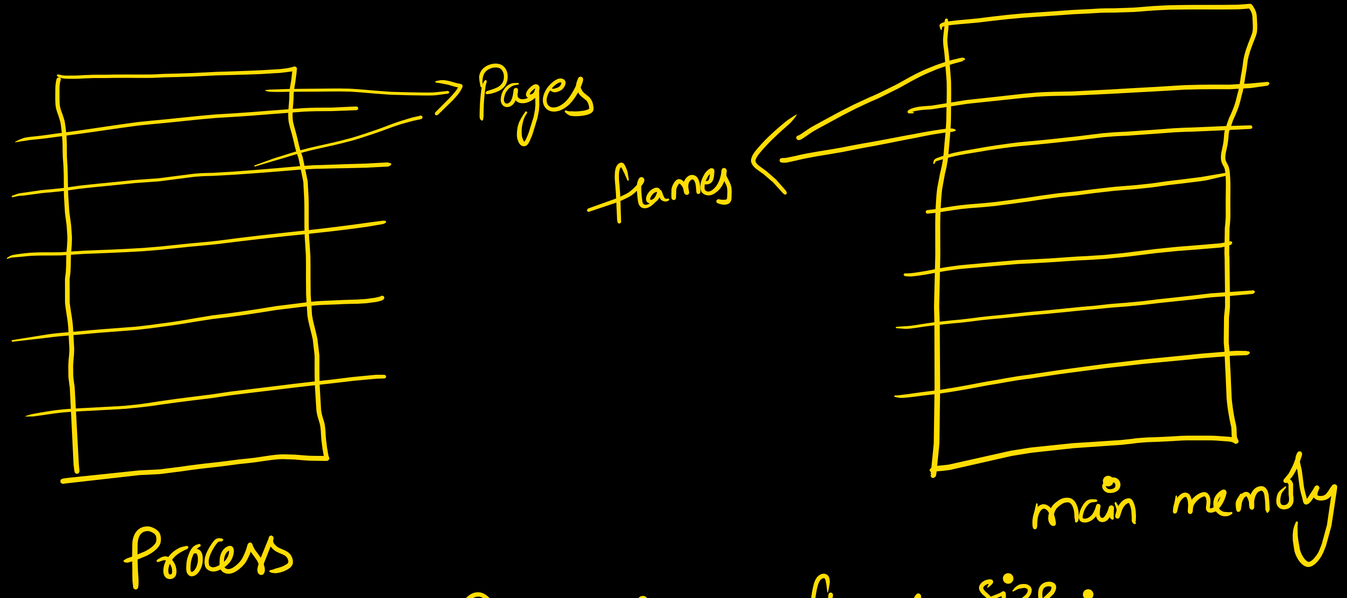
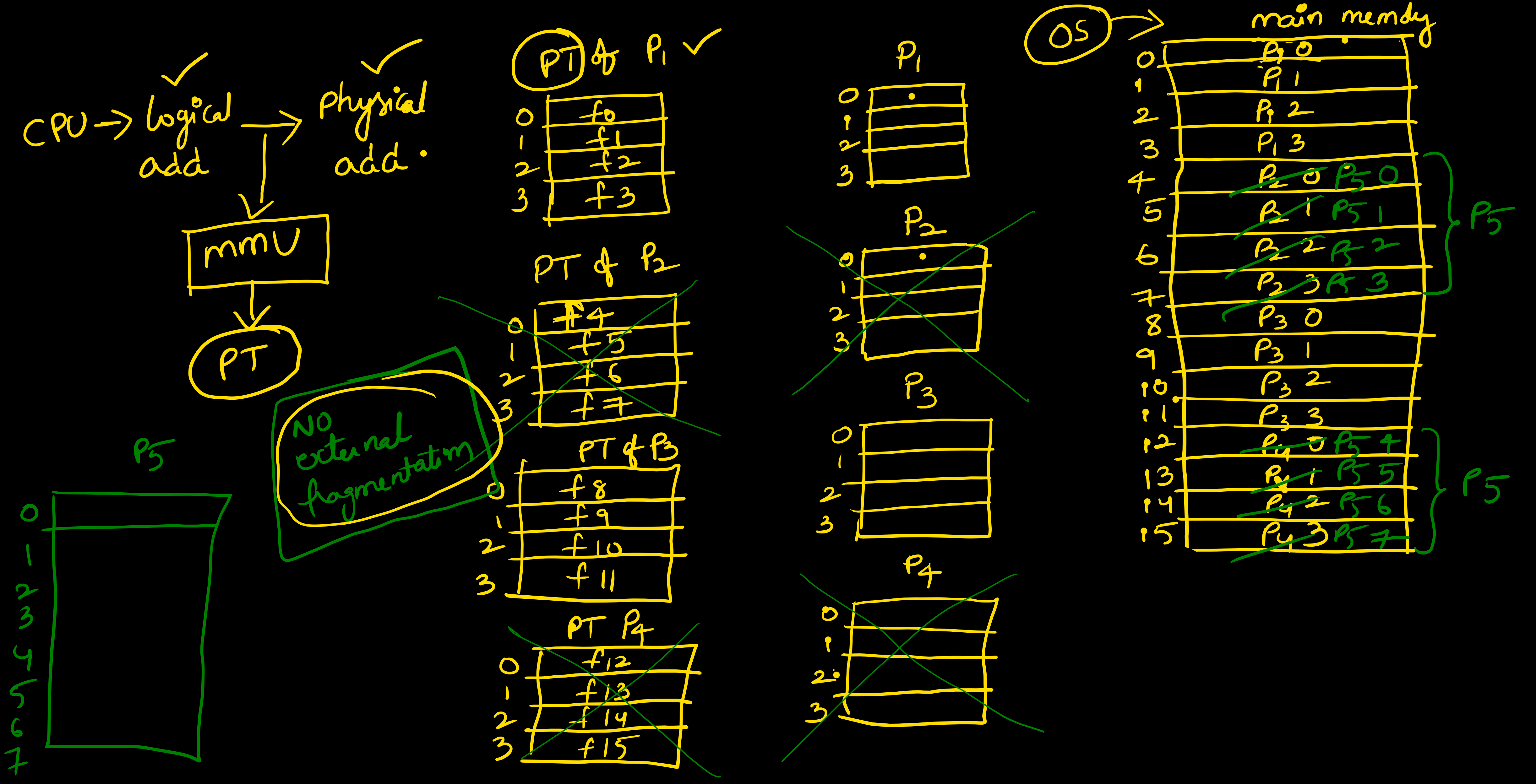
Paging:



Page size = frame size.

Any page can sit in any frame.

A process need not be contiguous


CPU → logical add → physical add.
MMU
PT
P5
0 1 2 3 4 5 6 7
NO external fragmentation
PT of P1
0 f0
1 f1
2 f2
3 f3
PT of P2
0 f4
1 f5
2 f6
3 f7
PT of P3
0 f8
1 f9
2 f10
3 f11
PT P4
0 f12
1 f13
2 f14
3 f15
P1
0 1 2 3
P2
0 1 2 3
P3
0 1 2 3
P4
0 1 2 3
OS
main memory
0 P1 0
1 P1 1
2 P1 2
3 P1 3
4 P2 0 P5 0
5 P2 1 P5 1
6 P2 2 P5 2
7 P2 3 P5 3
8 P3 0
9 P3 1
10 P3 2
11 P3 3
12 P4 0 P5 4
13 P4 1 P5 5
14 P4 2 P5 6
15 P4 3 P5 7
P5
P5

Page Size = 1 KB

Process size → 10 KB

\# Pages → $\frac{10\,KB}{1\,KB}$ = 10 exactly.

\# Frames → 10 → no wastage.

Page size = 1 KB

Process size → 10 KB + 1 B

10 KB → 10 Pages

1 B → 1 Pages

Pages → 11

\# frames → 11

last frame is wasted.

Best case → no wastage

worst case → entire Frame is wasted

Average → $\left(\frac{0+1}{2}\right)$ page/processing

Internal fragmentation

→ avg → ½ of Page

Size of a process is called logical add space (LAS)

No of bits required to address LAS is logical address.

Ex: Let process size = 128 KB

Then $LAS = 128\,KB = 2^{7} \times 2^{10}\,B = 2^{17}\,B$

$LA = \lceil \log_2 LAS \rceil = \underline{\underline{17 \text{ bits}}}$

LA =

Page number	Page offset
7	10

Let Page size = $1KB = 2^{10}\,B$

Page offset $= \lceil \log_2 \text{Page size} \rceil = 10$ bits

$\underline{\# \text{ Pages}} = \dfrac{\text{Process size}}{\text{Page size}} = \dfrac{128\,KB}{1\,KB} = \underline{128} = 2^{7}$

Size of main memory is called physical address space (PAS)

No of bits used to address main memory is called (PA)

Ex: Let main memory size = 512 KB = (PAS)

$= 2^{19}$ B

$PA = \lceil \log_2 PAS \rceil = 19$ bits. → PA

frame number	frame offset
10	9

Let frame size = 512 B. = 2^9 B

frame offset = 9 bits.

$$\#\text{ frames} = \frac{\text{main memory size}}{\text{frame size}} = \frac{512\text{ KB}}{512\text{ B}} = K = 2^{10}$$

LA →

PNO	Page offset

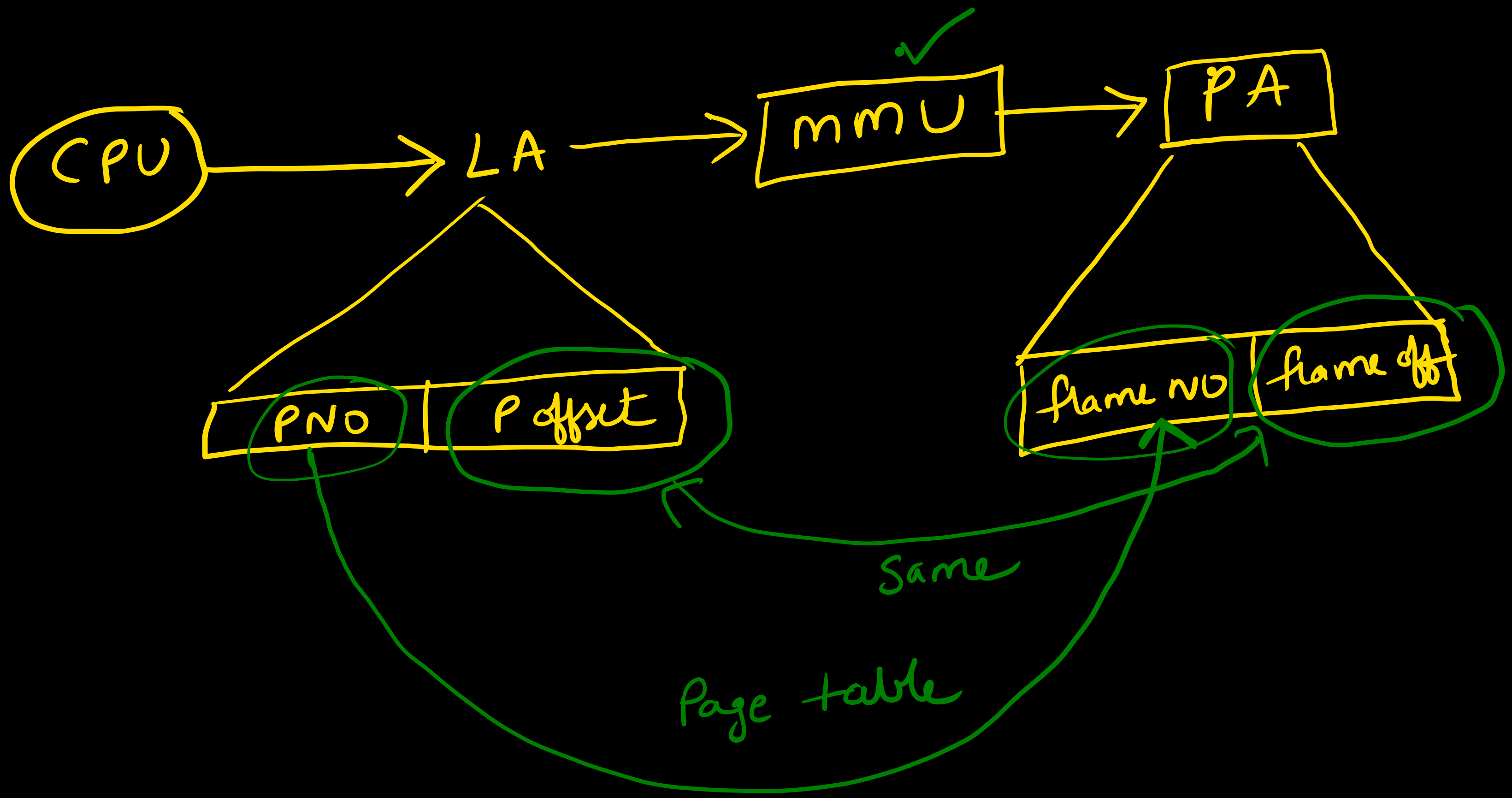
PA →

FNO	frame offset

CPU → LA → PA

LA → PNO → FNO → PA

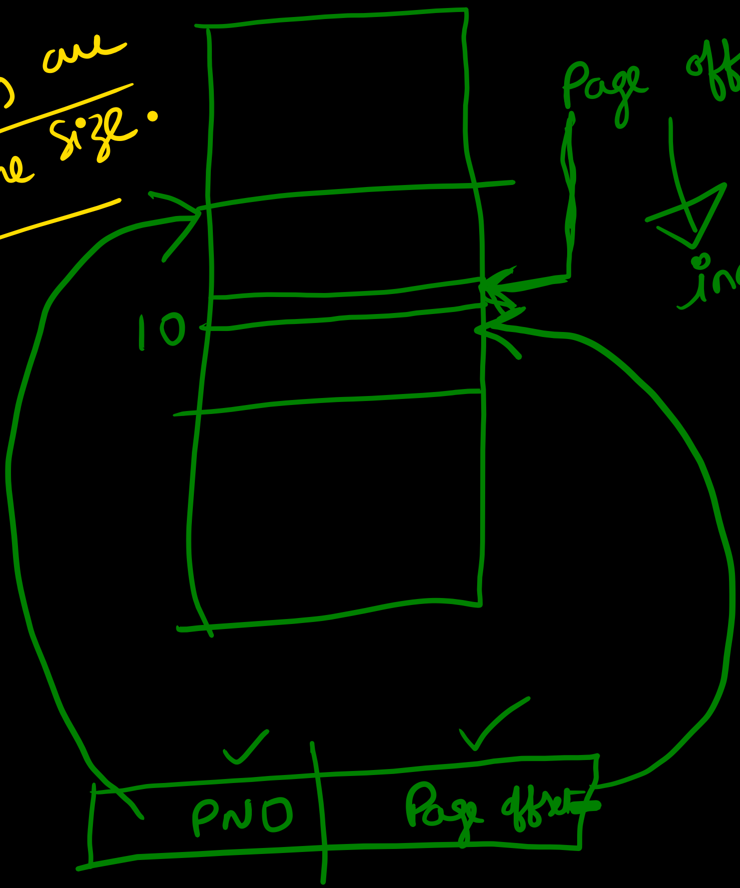
PNO → Page table



CPU
LA
MMU
PA
PNO
P offset
frame NO
frame off
same
Page table

PS = FS

all pages are of same size.



LA

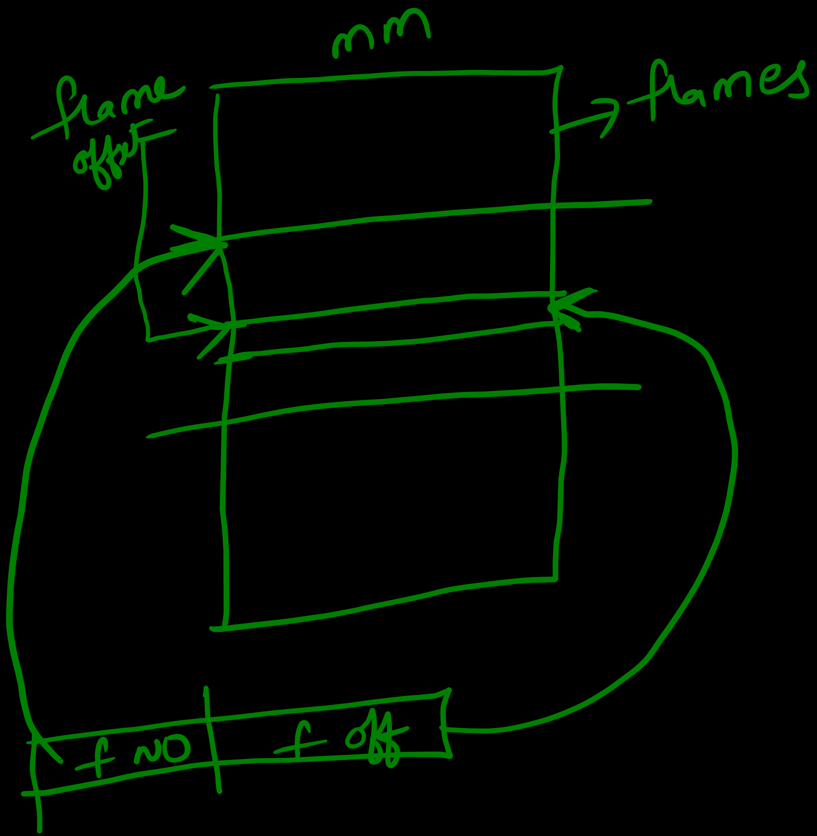
Page offset

index into a page

10

NID HID

PS = FS always



PA

main memory = 64 B = PAS

PA = 6 bits

Frame size = 4 B

frame offset = 2 bits

$$\# \text{frames} = \frac{\text{mm size}}{\text{frame size}} = \frac{64\text{B}}{4\text{B}} = 16 = 2^4$$

PA

frame no	frame off
4	2

Process size = 16 B = LAS

LA = 4 bits

Page size = $\boxed{4\text{ B}}$

Page offset = 2 bits

$$\# \text{Pages} = \frac{\text{LAS}}{\text{Page size}} = \frac{16\text{B}}{4\text{B}} = 4 = 2^2$$

LA

Page no	P off
2	2

MMU → H/W → PT

P_1

0	0	1	2	3
1	4	5	6	7
2	8	9	10	11
3	12	13	14	15

LA 6 = PA 14

Page table of P_1

0	0010 → frame number
1	0011
2	0100
3	0101

LA = 6

PNO	Poff
01	10

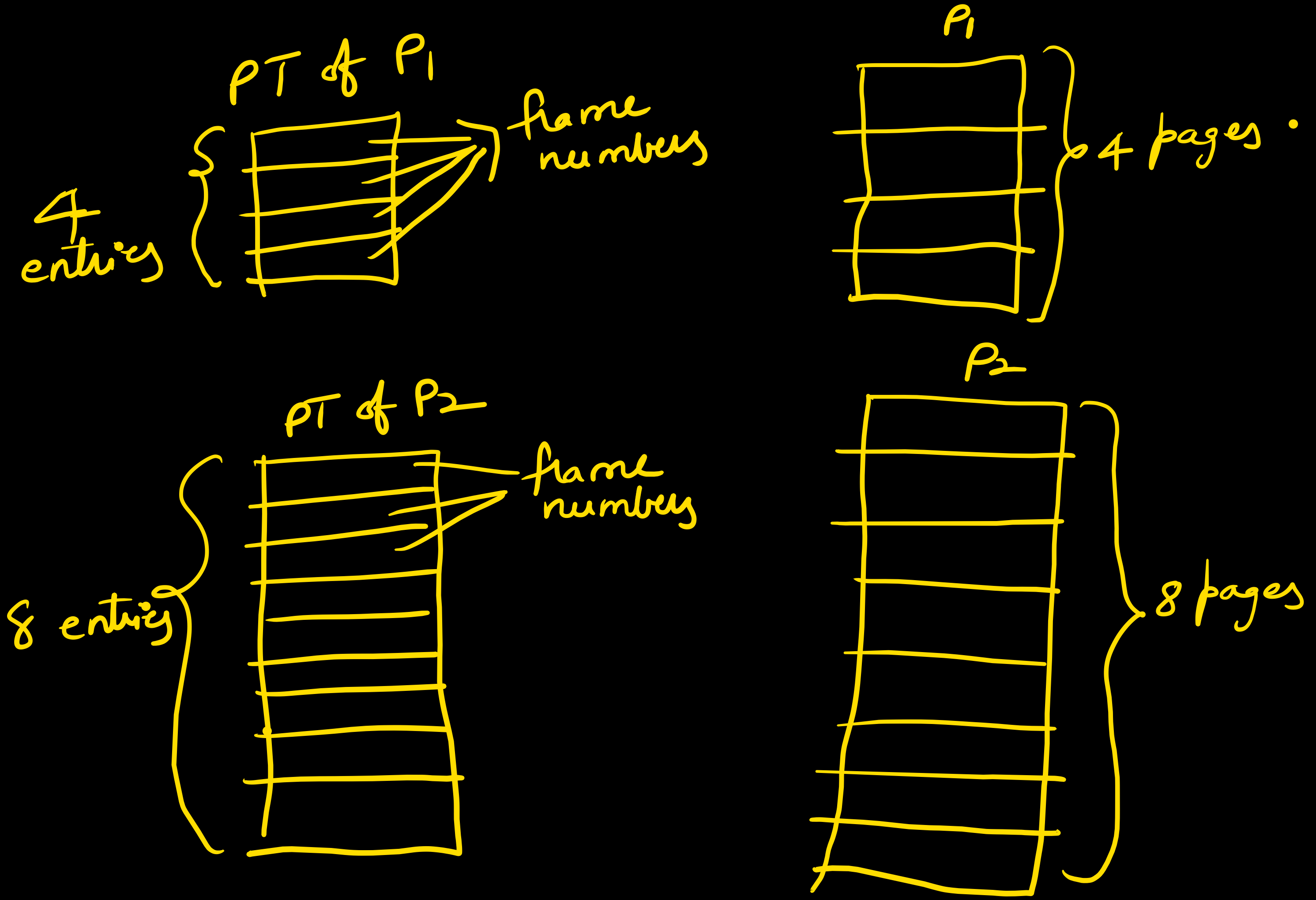
fNO	
0011	10

= 14

OS	0	0	1	2	3
OS	1	4	5	6	7
0. P_1	2	8	9	10	11
1. P_1	3	12	13	14	15
2. P_1	4	16	17	18	19
3. P_1	5	20	21	22	23
	6	24	25	26	27
	⋮				
	15				

Every process will have a seperate page table.

Ex:



The number of entries in a PT are equal to no of pages in the process.

For each page of a process, there will be an entry in the page table.

mm = 64 B

Page size = Frame size = 4 B

Process size = 16 B

Size of PT in ?

$$\# \text{pages} = \frac{\text{Process size}}{\text{Page size}} = \frac{16 B}{4 B} = 4.$$

$$\# \text{frame} = \frac{\text{mms}}{\text{Frame size}} = \frac{64 B}{4 B} = 16 = 2^4$$

Page table size = No of entries in PT * Size of each entry

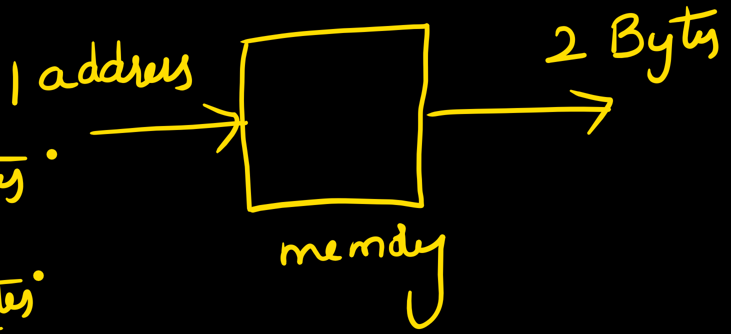
= No of pages in process * bits required for frame numb[er]

= 4 * 4 bits = 16 bits.

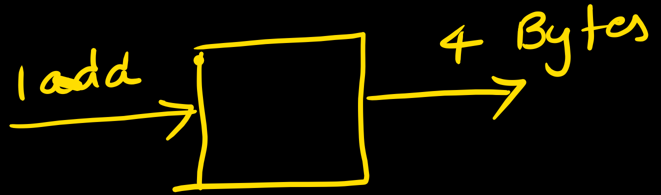
= 2 Bytes.

Generally computers are word addressable.

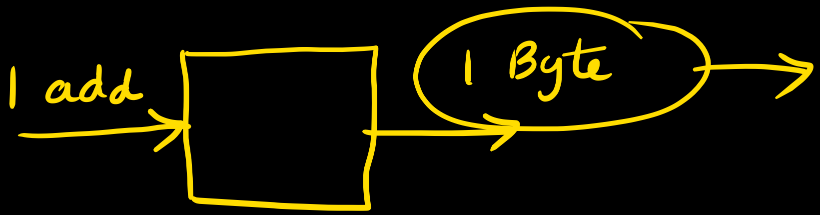
add:
100 → 2 Bytes.
100 → 4 Bytes.



Every 2 Bytes has one address. and so word size = 2 Bytes



Every 4 B has one address and so word size = 4 Bytes.



It is Byte addressable.

LAS = 256 MB , word size = 4 B , LA = ?

$$LAS = 256\,M B = \frac{256M}{4} W$$

$$= 64\,M \text{ words}$$

$$= 2^6 \times 2^{20} \text{ words}$$

$$= 2^{26} \text{ words}$$

$$\therefore LA = 26 \text{ bits}$$

In exam, if they dont mention words, then it is byte addressable i.e 1 word = 1 Byte.

PAS = 512 KB , words = 4 B , PA = ?

$$PAS = 512\text{ KB} = \frac{512\text{ K}}{4}\text{ words}$$

$$= 2^{17}\text{ words}$$

$\therefore$ PA = 17 bits.

PS = 512 B , $\underline{\underline{WS}}$ = 4 B , $\underline{\text{Page offset}}$

$$PS = 512\text{ B} = \frac{512}{4}\,w = \underline{\underline{2^{7}}}\,w$$

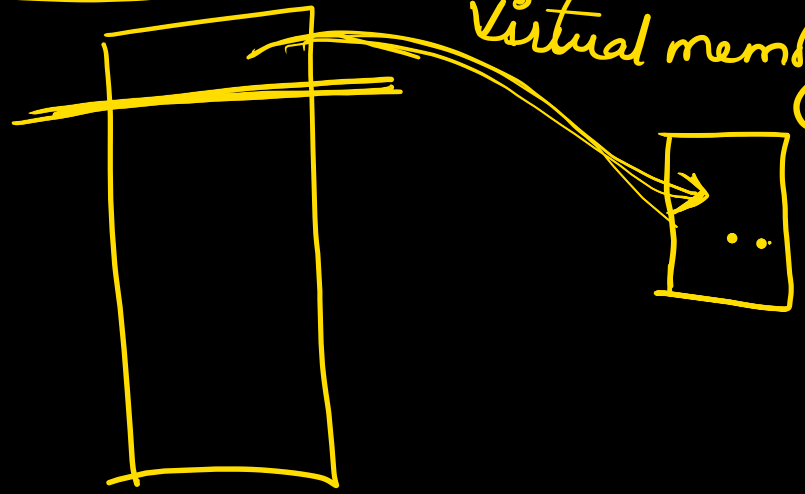
P offset = 7 bits

Virtual address space :

if Process size > main memory size

128 MB → 1 MB.

LAS is called "virtual address space"



Virtual memory. ✓

$LAS = M = 2^m$

$LA = \lceil \log_2 M \rceil = m$ bits

$PAS = P = 2^p$

$PA = \lceil \log_2 P \rceil = p$ bits

Page size $= 2^k =$ frame size.

$\# \text{Pages} = \dfrac{LAS}{\text{Page size}} = \dfrac{2^m}{2^k} = 2^{m-k}$

LA

m-k	k

$\# \text{frames} = \dfrac{PAS}{\text{frame size}} = \dfrac{2^p}{2^k} = 2^{p-k}$

PA

p-k	k

$LAS = \# \text{pages} \times ps$
$= 2^{m-k} \times 2^k = 2^m$

$PAS = \# \text{frame} \times \text{frame size}$
$= 2^{p-k} \times 2^k = 2^p$

Page table size $= \underbrace{\text{No of entries}} \times$ frame no bits.
$= 2^{m-k} \times (p-k)$

VAS — LAS = 128 MB = 2^{27} B   PTS = ?

PAS = 1 MB = 2^{20} B

Page size = 4 KB = 2^{12} B

LA = $\overbrace{[\ 15\ \mid\ 12\ ]}^{27}$

Page numb   Page offset

PA = $\overbrace{[\ 8\ \mid\ 12\ ]}^{20}$

frame numb.   frame off

PTS = # Pages * bits for frame number

= 2^{15} * 8 bits =

Virtual memory

many ways.

LAS = 128 KB , PAS = 128 KB , Page size = 4 KB , PTS = ?

$= 2^{17}$ B $\qquad = 2^{17}$ B $\qquad = 2^{12}$ B.

LA = (17)

Page no	P off
5.	12

PA = (17)

frame no	P off
5 ✓	12

PTS = # Pages * bit for frame number.

$= 2^5 * \underline{5}$ bits.

$LAS = 256\,KB$, $PAS = 1\,MB$, Page Size $= 4\,KB$

Page table entry size $= 2B.$ ✓

PTS ?

~~PS~~ Page table

$$\# \text{pages} = \frac{256\,KB}{4\,KB} = 64.$$

$$PTS = \#\text{pages} * PTE = 64 * 2B = 128\,B.$$