COMPUTER SCIENCE

Memory Management

Memory management techniques

Part 02

Lecture no -04

Dr. KHALEEL KHAN SIR

1-memory management techniques

Session-11 : 3/12/2022 : Multi-Level Paging / Hierarchical Paging

< Reduce P.T size ovhd + Associate smaller P.T with Process >

L.A = 32 bits ; P.S = 4KB

11K

4MB

chunks

Pages of P.T

4B PTE

N = 1M

P.S of P.T = 1K

PTAS

P0 P1 P2

f

1K

1K

Inn. P.T (outer P.T)

outer P.T

Pages of Process

P.A.S

P0 P5 P10

4KB

P0 P10 P15

Pages of P.T

LAS = 2^{32} = 4GB

P0 P1 P2 P3

$N = \frac{2^{32}}{2^{12}} = 1M$

4KB

Inner P.T's

Ideally the P.T of the Process should fit in one frame ;

P.T will be Contiguous

(Paging on P.T is MLP)

VA = 32 bits; P.S = 4KB ✓ = $\frac{4KB}{4B}$ = 1K → P.S

Process will be Associated with OPT + Pages of IPT in M.M

Let P.S (chunk) of I.P.T = 1K

4KB

e = 4B

1K

0	g
e	k
n	0

O.P.T (2nd level P.T)

P

P.T (1m) — No. of Pages = 1K

0	1	P_0
1	0	
2		
x	a	P_e
y	b	
n	m	P_x

1st level (I.P.T)

PAS

f_0	P_1
f_1	P_0
f_a	P_x
f_b	P_y
j	P_0
h	P_e
o	P_x
f_m	P_n

Pages of I.P.T

4 × 4KB = 16KB

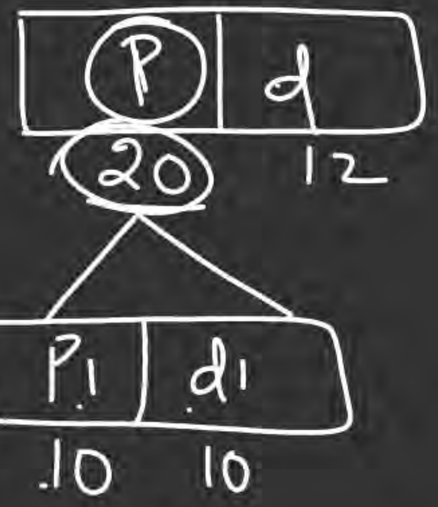
$\frac{2^{32}}{2^{12}}$ = 1m ✓

VAS 1m — 5 Pages

	P_0
	P_1
Code	2
	P_x
	P_y
data	2
	P_n
Stack	1

VA = 32 bits; P.S = 4KB; N = $\frac{2^{32}}{2^{12}} = 2^{20} = 1M$

1 level Paging

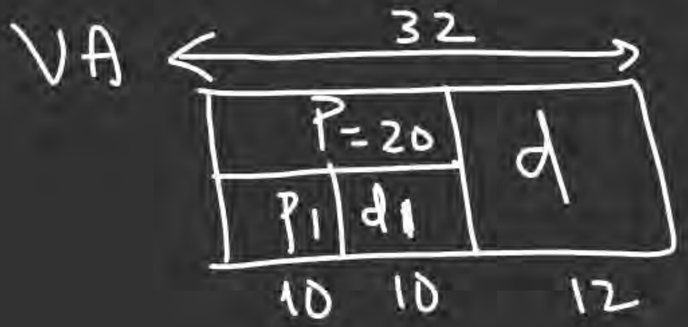


d_1 = Offset within Page (chunk) of I.P.T

P.S of I.P.T = $2^{d_1} = 2^{10} = 1k$

P_1 = No. of Pages (chunks) in I.P.T = 1K

= No. of entries in OPT



(2-level Paging)

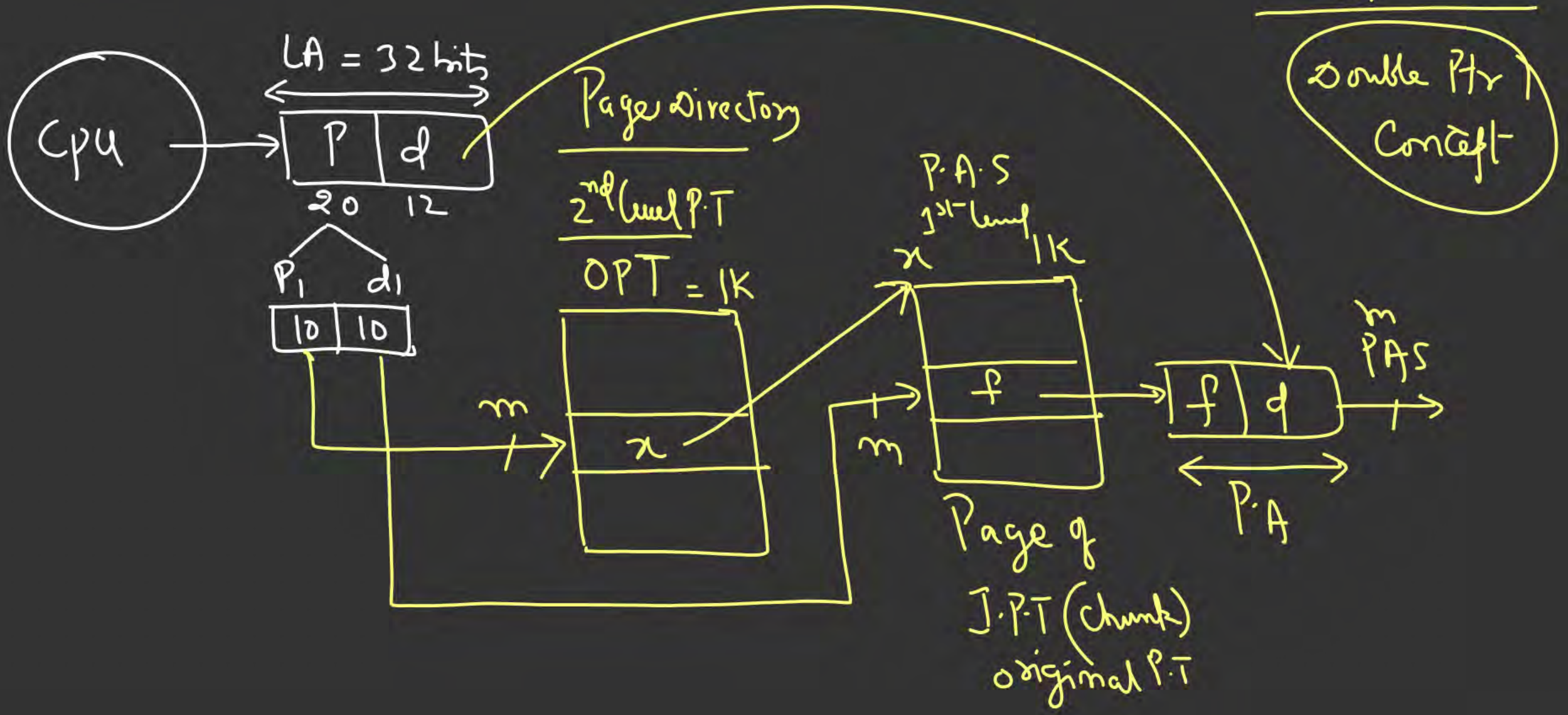
Address Translation with 2-Level Paging Architecture

MMAT = m

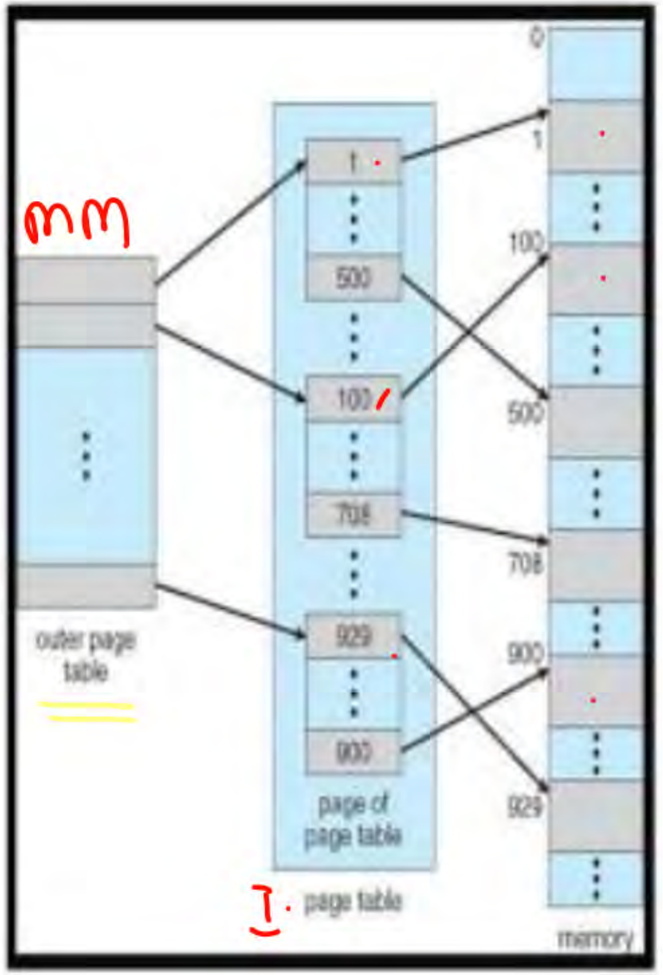
EMAT = 3m

2LP

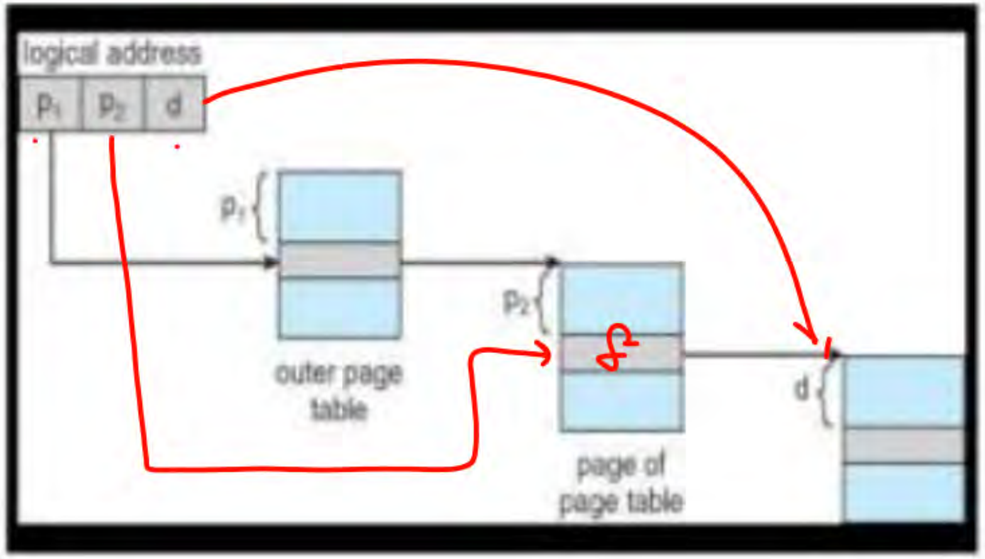
Double Ptr Concept

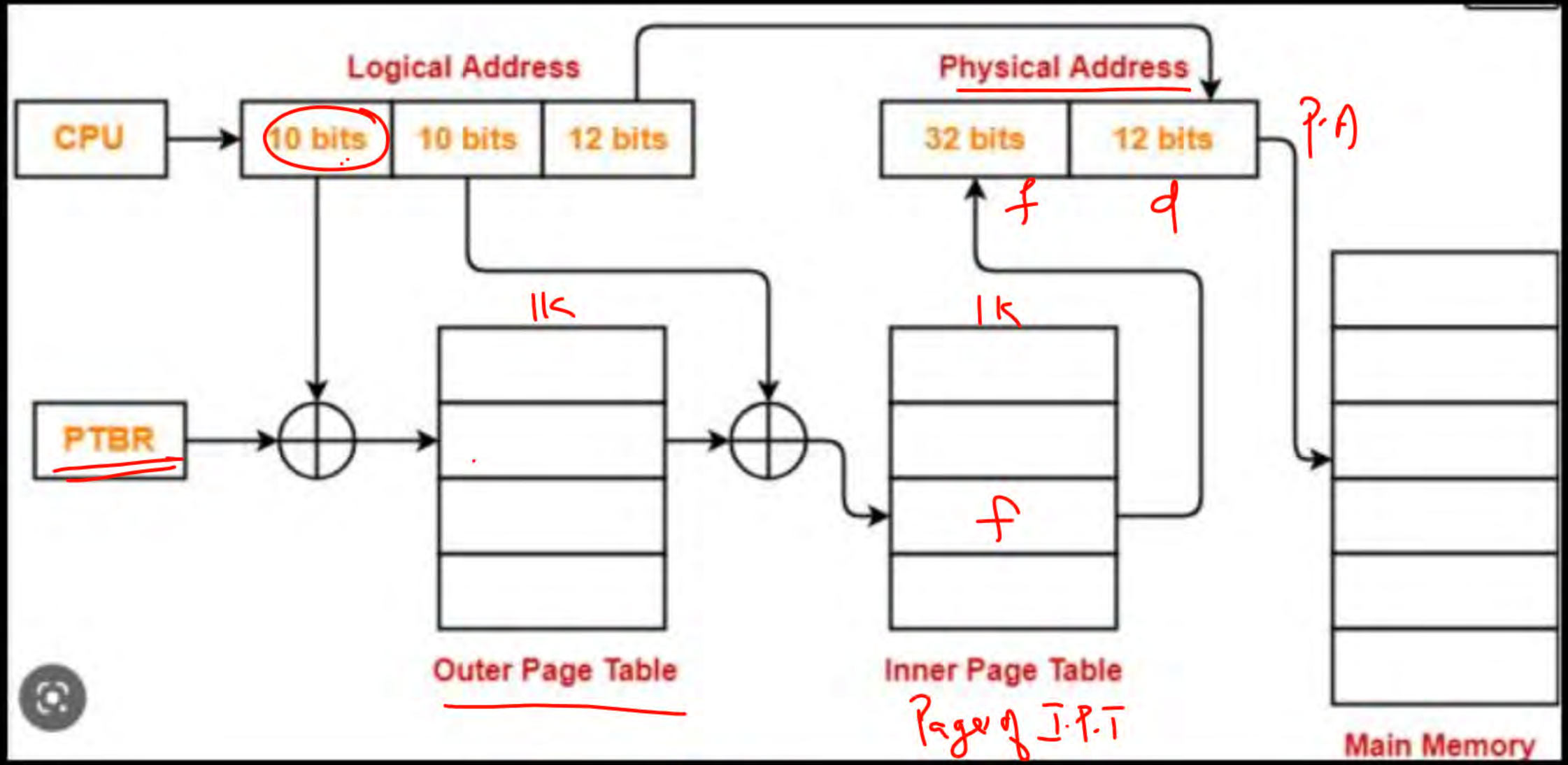


Hierarchical Paging



Hierarchical Paging





2-level Paging

Multi-Level Page Tables

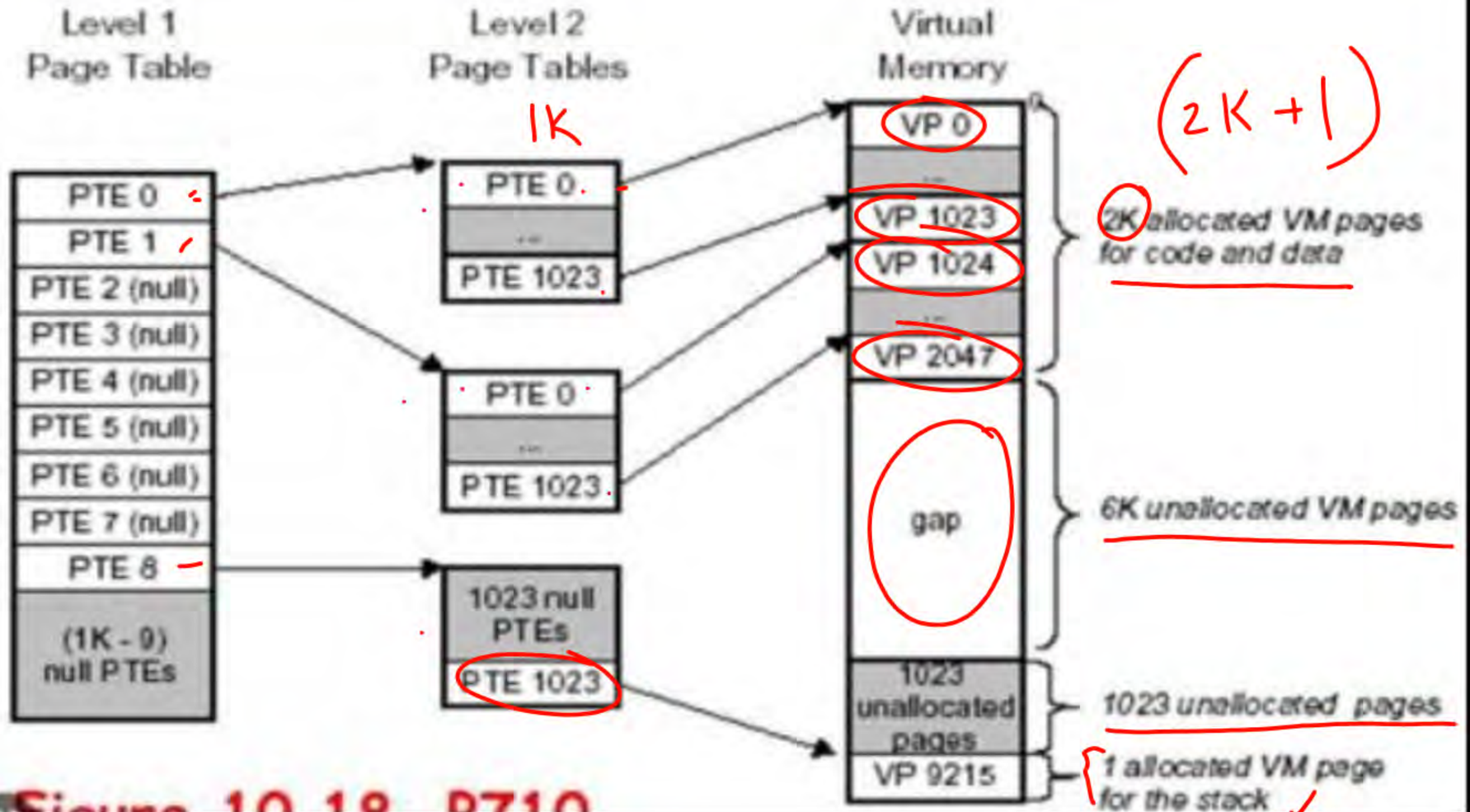


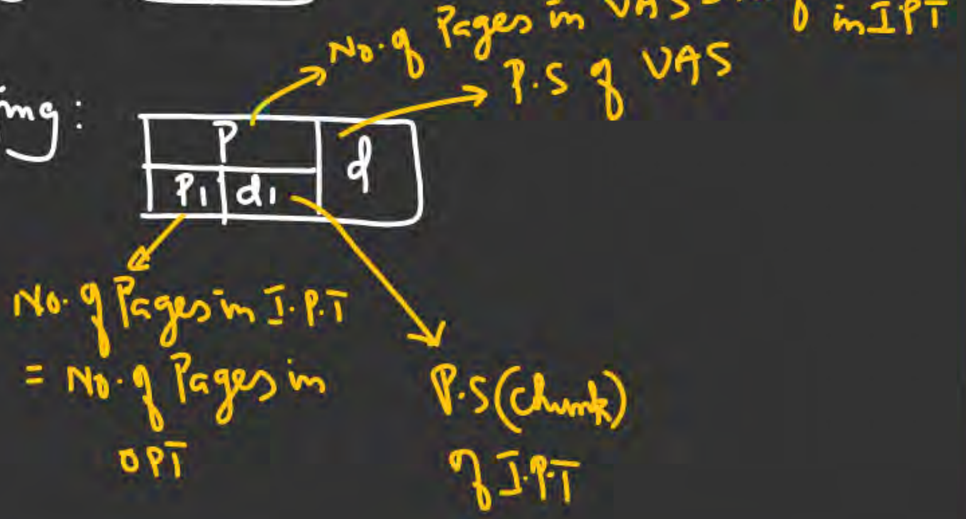
Figure 10.18 P710

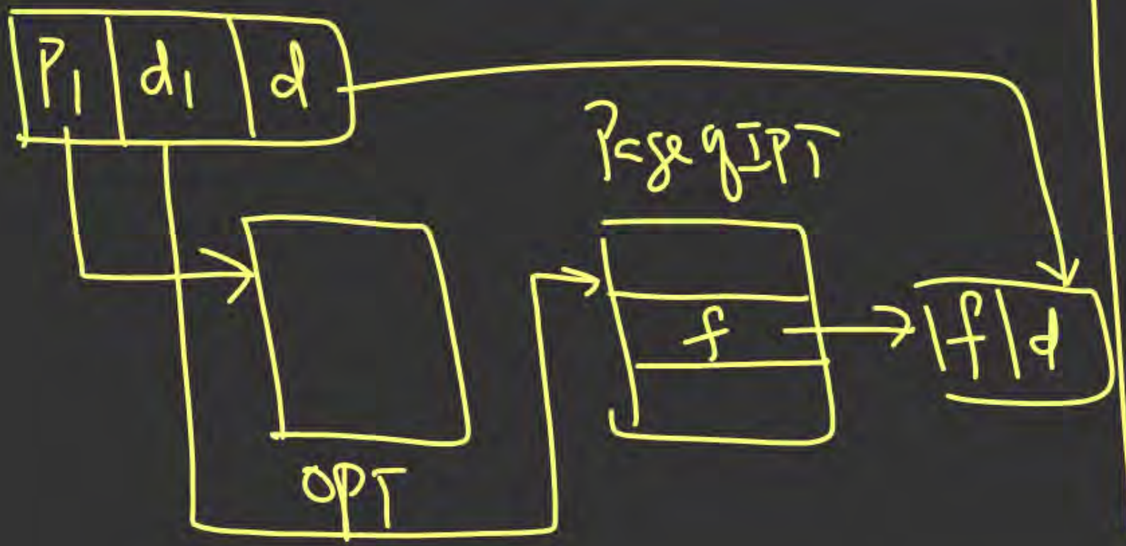
Address Formats:

① 1-level Paging: VA | p | d → Page Size

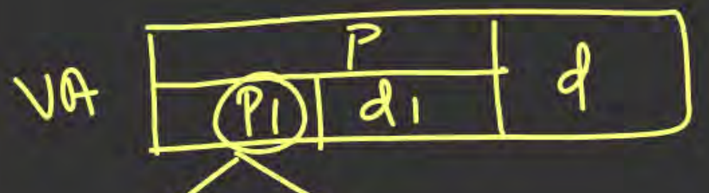
No. of Pages in VAS = No. of entries in P.T

② 2-level Paging:



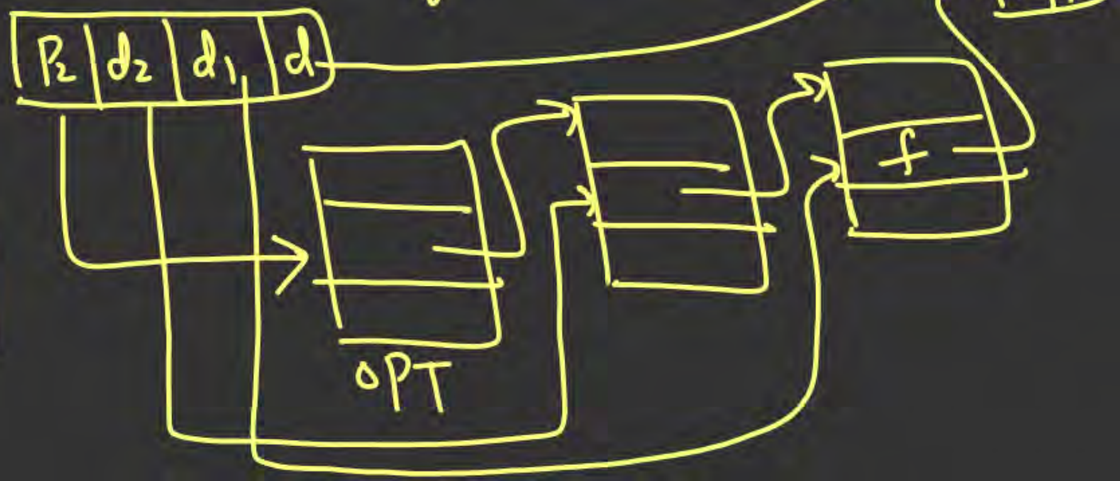


③ 3-level Paging



p2 → No. of Pages of 2nd level P.T = No. of entries in O.P.T

d2 → Page-Size of 2nd level P.T



② Paging with Hashing (Hashed Page Table)

f (Key) = index → H.T for getting the value;

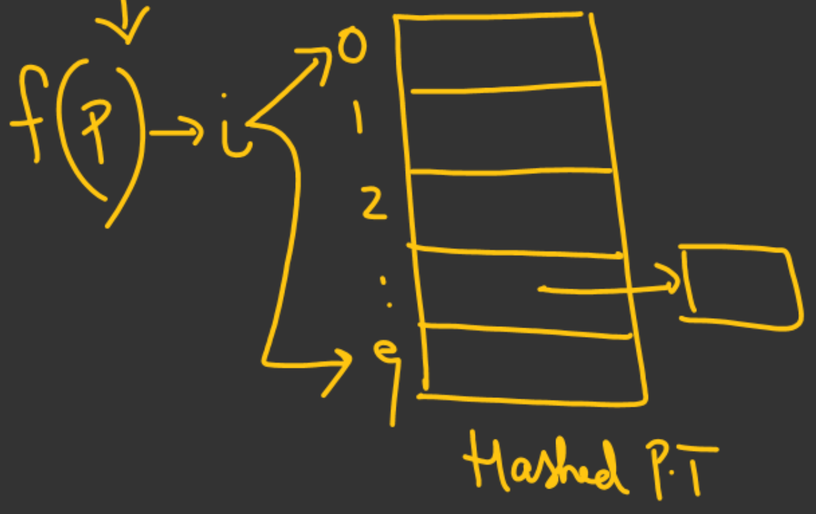
Hashing : Collission

$(f(x_1) = f(x_2) = i$
Collission$)$

- Probing
- Chaining ✓

$f = (\%\ 10) == 10$

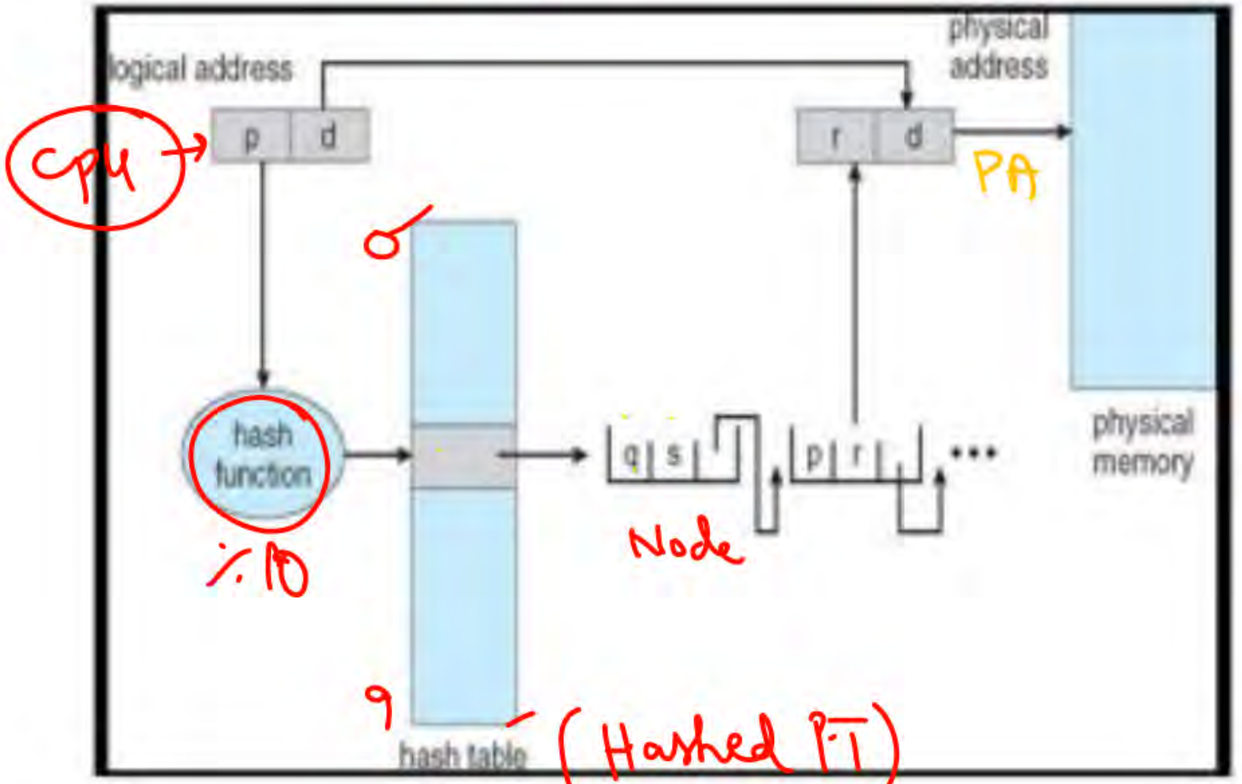
5 % 10 = 5
15 % 10 = 5
25 % 10 = 5



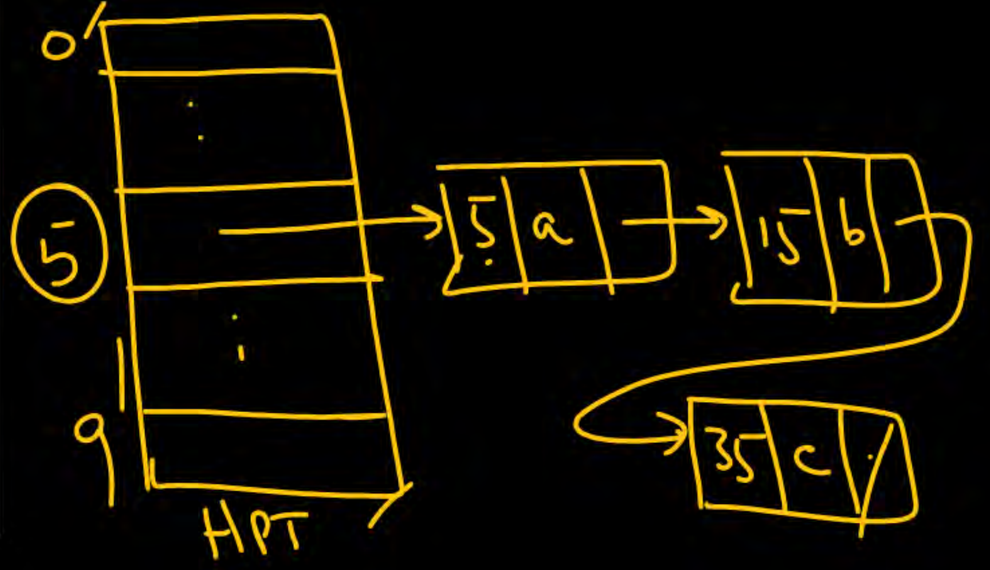
Hashed P.T

Hashed Page Tables

Paging with Hashing

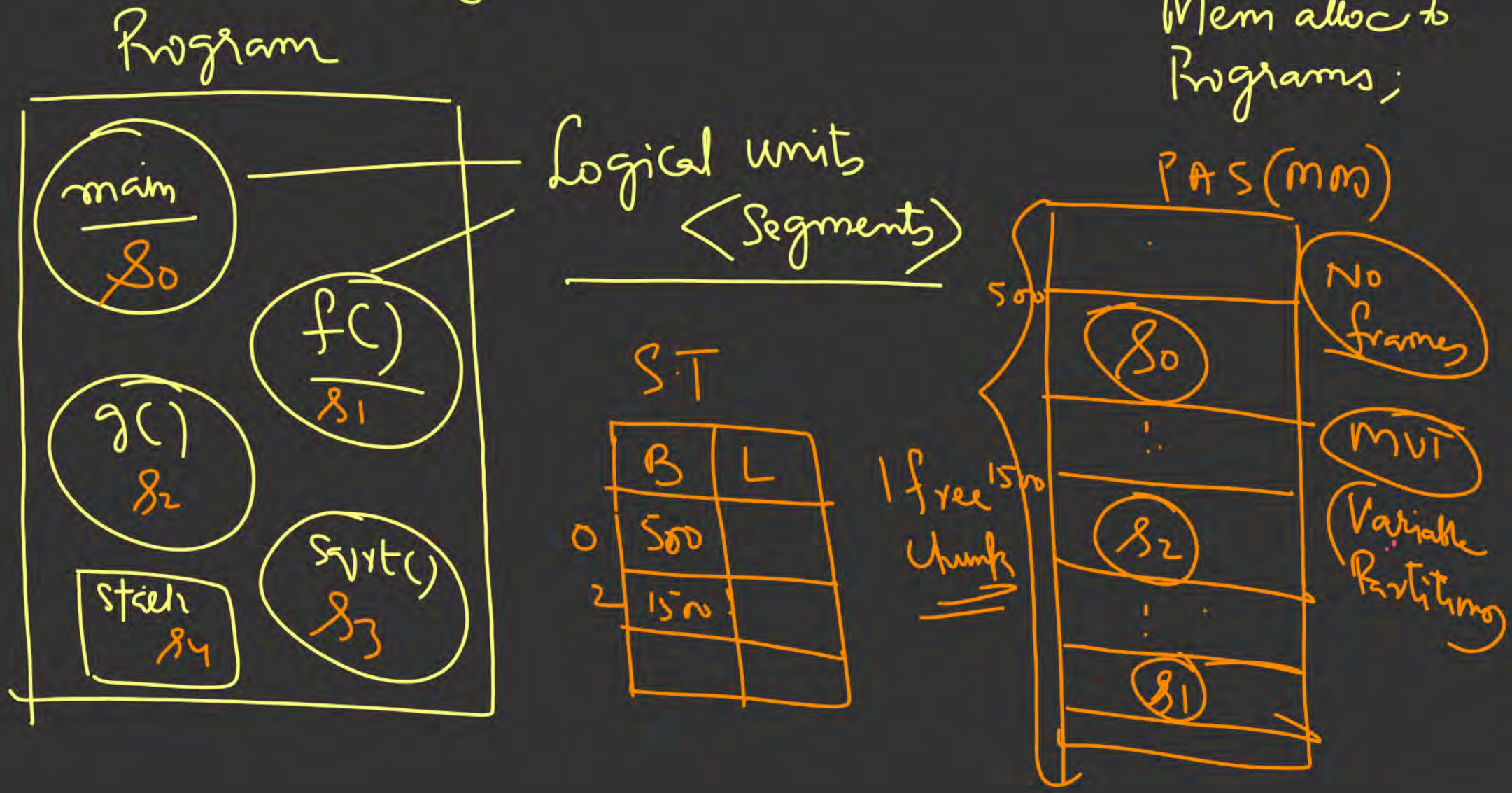


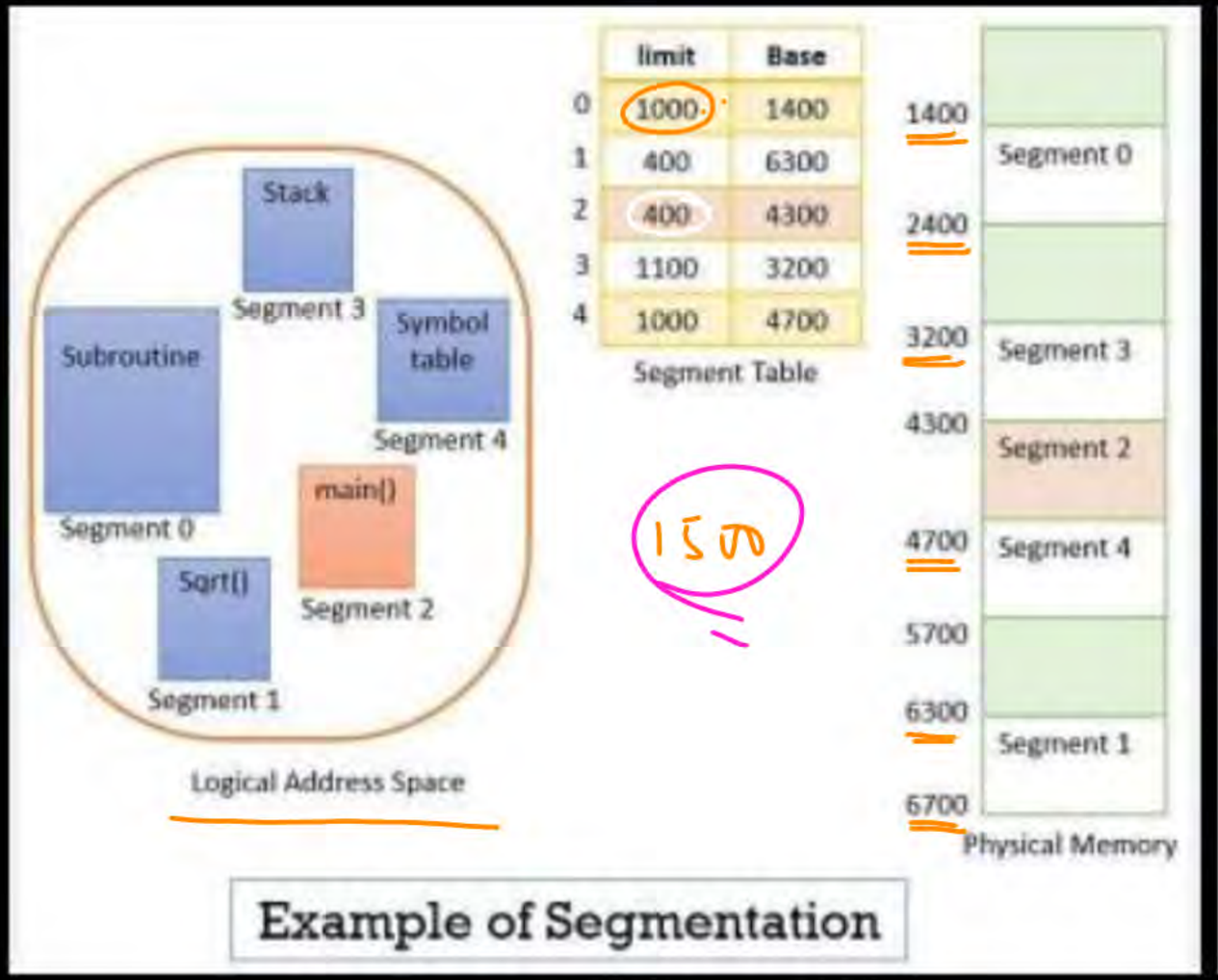
Hashed P.Ts relatively has less no. of entries in comparison to original (Traditional) PTs

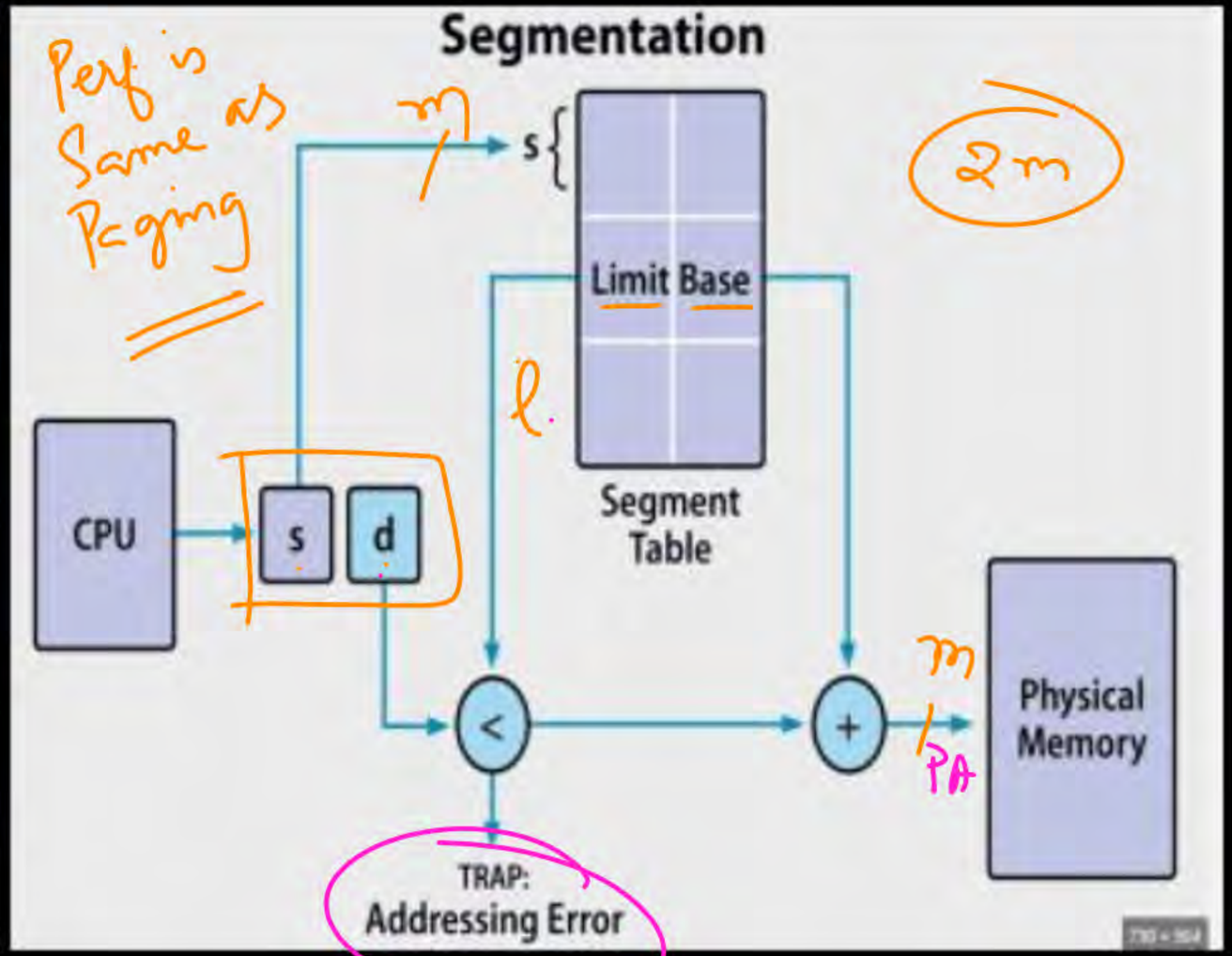


II. Segmentation:

Paging does not guarantee user's view of Mem alloc to Programs;







LA PA

1) $\langle 2, 333 \rangle \rightarrow 4300 + 333 = 4633$

2) $\langle 4, 1500 \rangle \rightarrow$ Trap

3) $\langle 1, 33 \rangle \rightarrow 6300 + 33 = 6333$

Paging vs Segmentation

Paging +
Paging with TLB
+
Hashed Paging
+
M.L-Paging
+
Segmentation

	I. Frag	E. Frag m.
Paging	✓ Last Page	✗
Segment	✗	✓ <in PAS> Variable Part.

Paging ⇒ Segmented - Paging Architecture

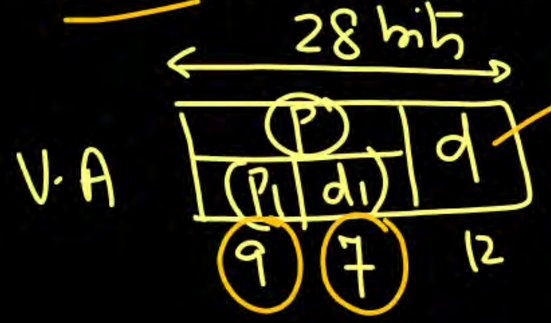
Q.7

Consider a System using 2 Level Paging Architecture. The Top level 9 bits of the Virtual Address are used to index into the outer Page Table. The next 7 bits of the Address are used to index into next level Page Table. If the size of Virtual Address is 28 bits. Then

(i) How large are the Pages and How many are there in Virtual Address Space?

(ii) If P.T.E at both levels is 32 bits in size then what is the Space Overhead needed to translate Virtual Address to Physical Address of an Instruction or Data Unit?

4B



P.S of VAS $= 2^{12} = 4KB$

P = No. of Pages in VAS $= 2^{P}$

P = 16

$\therefore$ No. of Pages $= 2^{16} = 64K$

$(\overline{OPT} + \overline{IPT})$

$(2^{9} \times 4B + 2^{7} \times 4B) = 512 + 128 = 640$

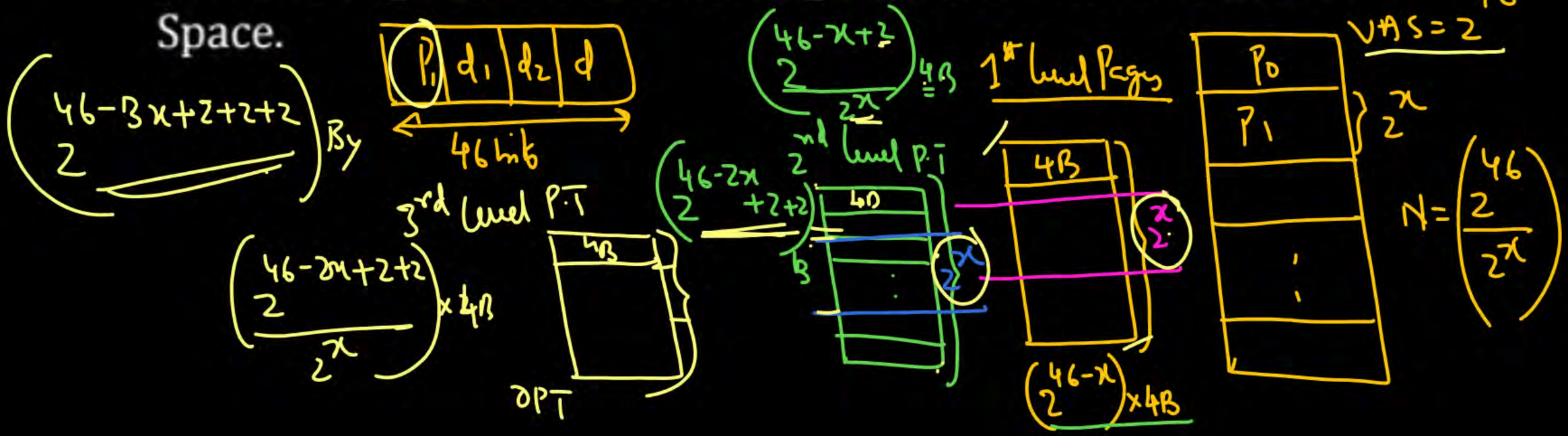
$(640 \times 4B)$

Q.8

PYQ

Consider a Computer System using 3 Level Paging Architecture with a uniform Page Size at all levels of Paging. The size of Virtual Address is 46 bits. Page Table Entries at all levels of Paging is 32 bits. What must be the Page Size in Bytes such that the Outer Page Table exactly fits in one frame of Memory. Assume Page Size is power of 2 in Bytes. Show the Virtual Address format indicating the number of bits required to access all the three levels of Page Tables and the Page offset of Virtual Address Space.

Let $PS = 2^x$ B



OPT · Should fit in one frame of Memory;

$$\left(2^{46-3x+2+2+2}\right)B = 2^{x}B$$

frame size = Page size

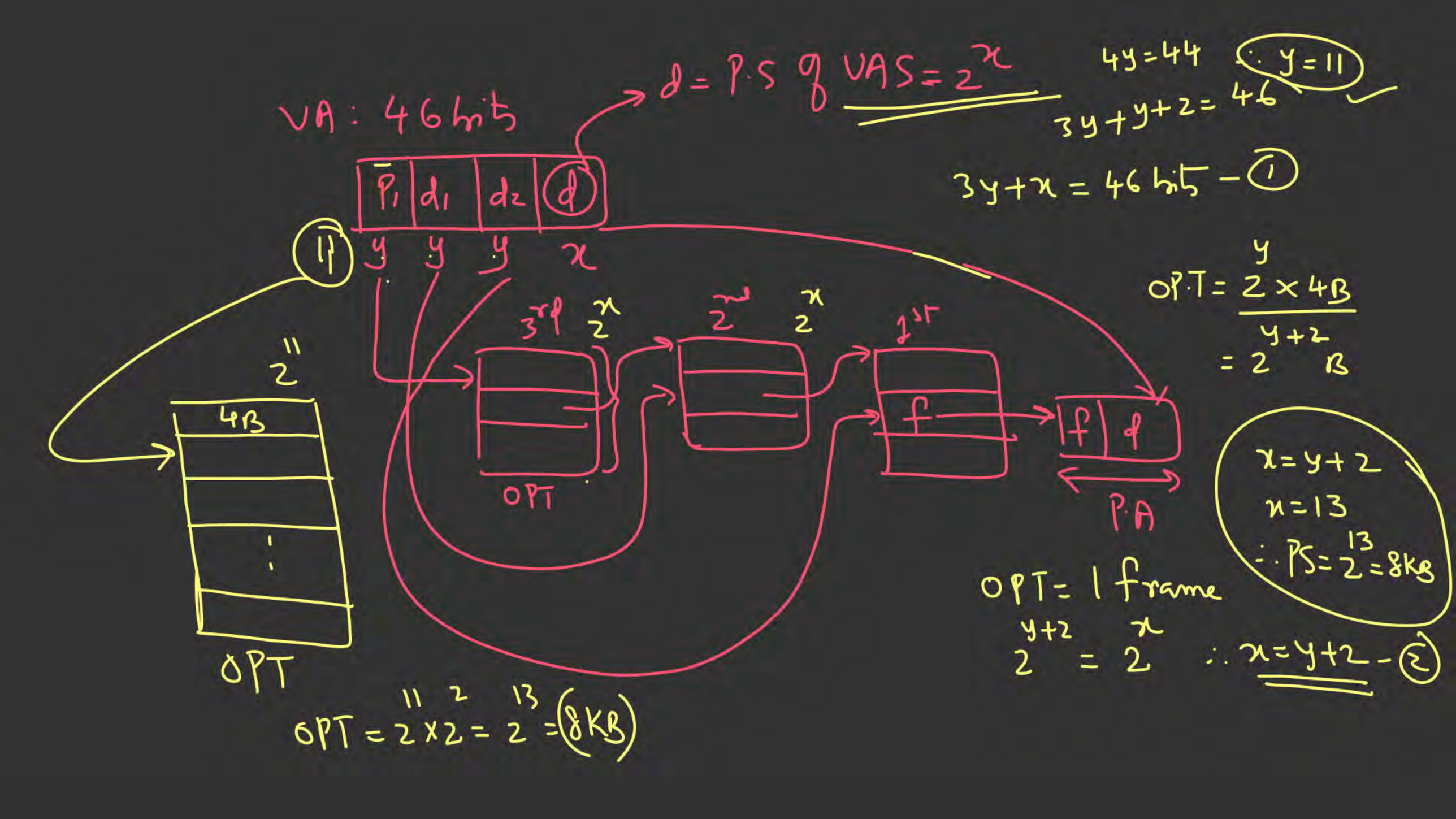
$2^{x} = FS$

$46 - 3x + 6 = x$

$52 = 4x$

$x = 13$

$\therefore P.S = 2^{x} = 2^{13} = 8KB$


VA: 46 bits
d = PS of VAS = 2^x
$\bar{P}_1$ | d_1 | d_2 | d
11
y y y x
2^{11}
4B
OPT
3rd 2^x
2nd 2^x
1st
OPT
f
f | f
P.A
$4y = 44$
$\therefore y = 11$
$3y + y + 2 = 46$
$3y + x = 46$ bits - (1)
OPT = $\frac{2 \times 4B}{y+2}$ = 2^y B
OPT = 1 frame
$2^{y+2} = 2^x$
$\therefore x = y + 2$ - (2)
$x = y + 2$
$x = 13$
$\therefore PS = 2^{13} = 8KB$
OPT = $2 \times 2 = 2^{13} = (8KB)$

THANK YOU!