P.A.S = 256 MB

L.A.S = 4 GB

Frame Size = 4 KB

Page Table Entry Size = 2 B

Here, frame size = page size = 4 KB = 2^{12} B

offset bit number = $\lceil \log_2 2^{12} \rceil$

$= 12$ bits

Num of frames in 256 MB M.M $= \dfrac{256 \text{ MB}}{4 \text{ KB}}$

$$= \frac{2^8 \times 2^{20}}{2^2 \times 2^{10}} = 2^{28-12}$$

$$= 2^{16}$$

Num of bits required to address 2^{16} frames $= \lceil \log_2 2^{16} \rceil = 16$ bits

16 bits	12 bits

$\longleftrightarrow$ 28 bits $\Rightarrow$ Physical Address

III

Num of Pages $= \dfrac{4\,GB}{4\,KB}$

$$= \frac{4 \times 2^{30}}{4 \times 2^{10}} = \frac{2^{32}}{2^{12}}$$

$$= 2^{20}$$

Bits required to address 2^{20} pages $= \lceil \log_2 20 \rceil$

$$= 20 \text{ bits}$$

offset bit number = 12 bits

20 bits	12 bits

$\longleftrightarrow$ 32 bits $\Rightarrow$ Logical Address

Page Table size $= N * e$

$= 2^{20} * 2B$

$= 2MB$

Frame size $= 4KB$

Now Page Table Size is larger than frame size. We know that page table resides in M.M and M.M is divided into multiple equal sized frame. Here this page table can not be stored in a single frame. We have to divide this page table so that we can store it in multiple frames of M.M

Num of frames required to store 2M Page table =

$$\frac{2\,MB}{4\,KB} = \frac{2 \times 2^{20}}{2^2 \times 2^{10}} = 2^{21-12} = 2^9 = 512$$

Num of Bits for addressing 2^9 frames = 9 bits

outer page table contains 2^9 entries as we need 2^9 frames to store the 2MB page table and 9 bits are required to address each of these 2^9 entries

#

We have divided the main/inner 2MB page table into 2^9 equal sized parts.

We store these 2^9 parts into individual 2^9 frames in main memory.

Now we find the number of entries in each of these 2^9 parts.

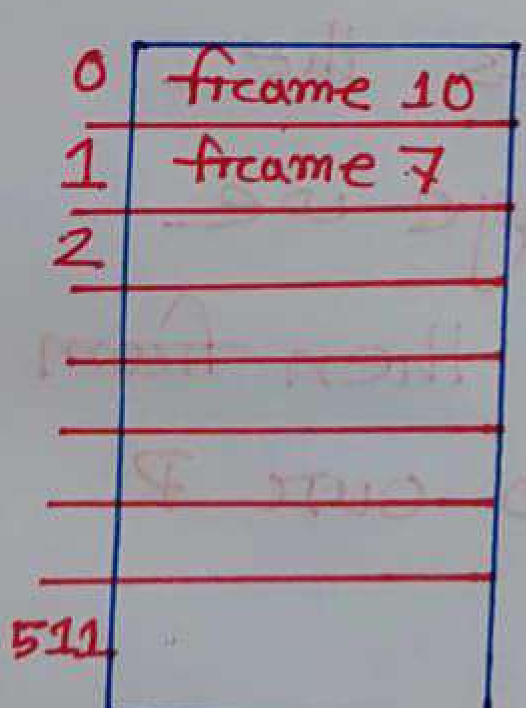
We know frame size = 4KB

Page table entry size = 2B

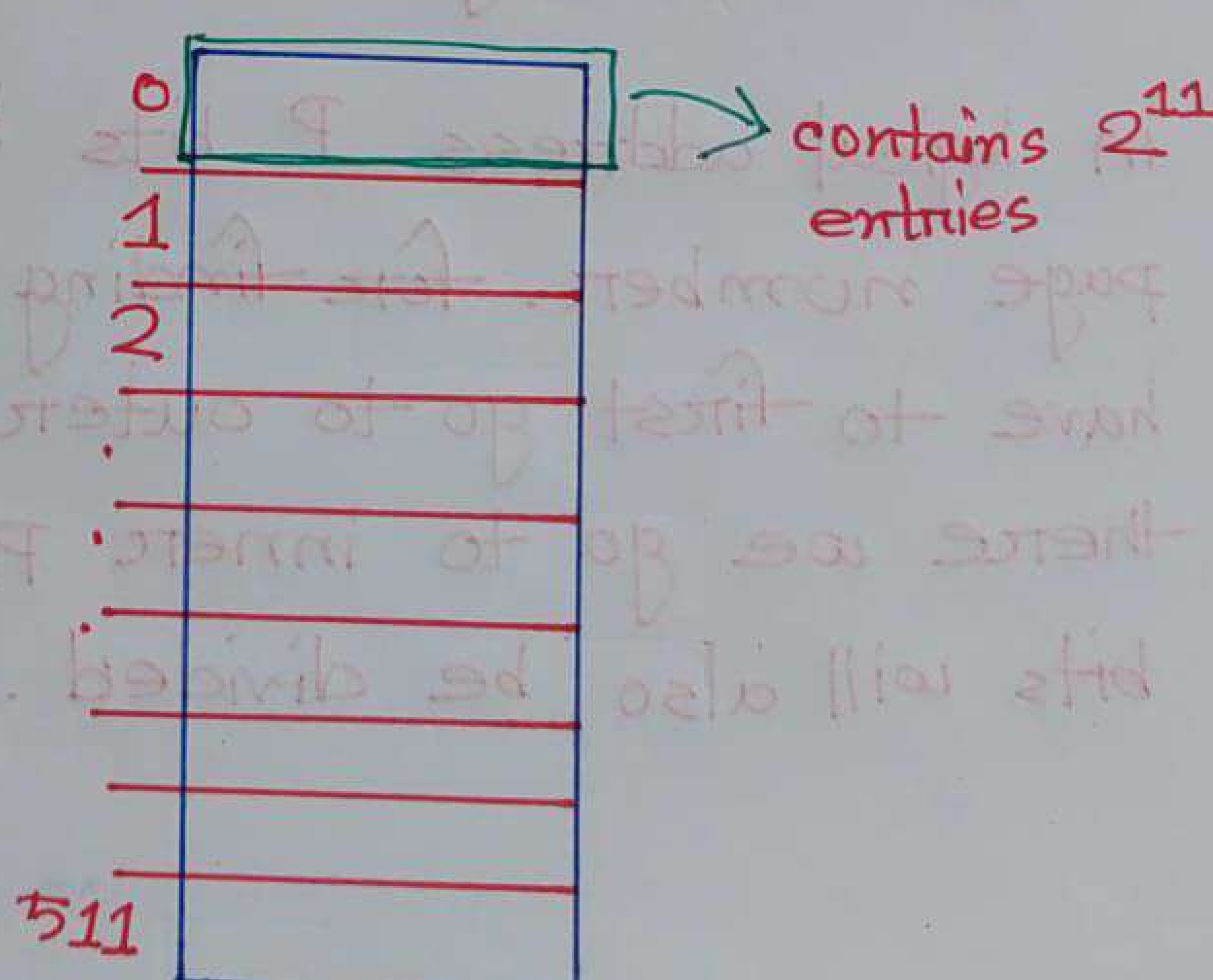
so each partitions stored in individual frame and total entries in each frame $= \frac{4KB}{2B}$

$$= \frac{2^{12}}{2} = 2^{11}$$

To address these 2^{11} entries we need 11 bits.



outer P.T



Inner P.T

So our Logical Address:

20 bits	12 bits

Now we have divided our single page table into two page table. Here, outer page table indicate the frame number where each partitions of inner page table resides. Then After going to that frame, we will get the partitions of inner page table. Then from the entries of that partition we get the frame number of our requested page.

in logical address P bits indicates the page number. ~~for finding~~ a page we have to first go to outer P.T then from there we go to inner P.T. So our P bits will also be divided.

p	d

$\downarrow$

O.P.T bits	I.P.T bits

in the example, we need 9 bits to address outer P.T and 11 bits to address Inner P.T

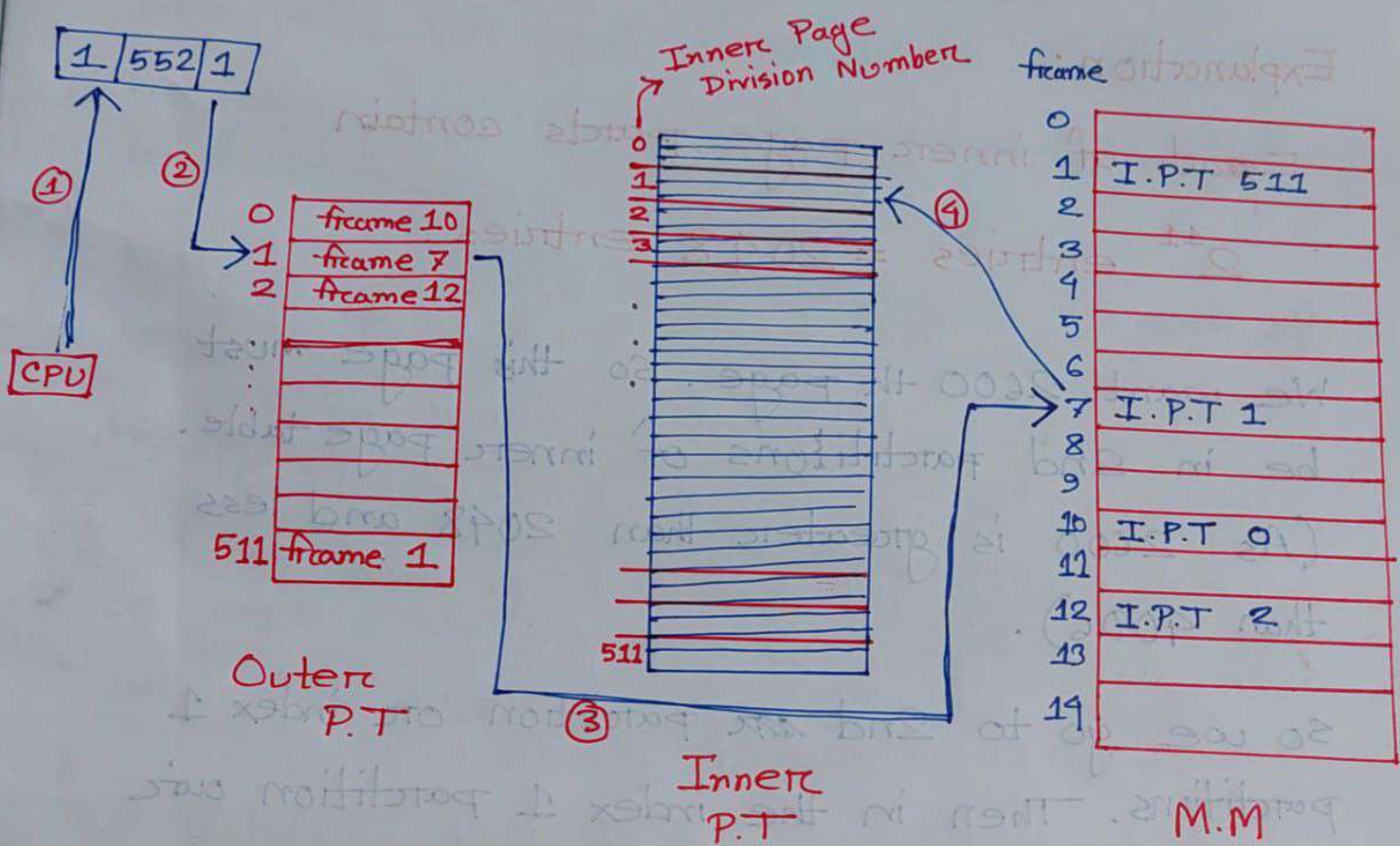
so

20	12

$\downarrow$

9	11	12

$\Rightarrow$ for 2 level paging



If CPU wants 2600th page and 1st Byte of that page, then CPU generates L.A :

20 bits	12 bits
101000101000	1

9 bits	11 bits	12 bits
1	01000101000	1

1	552	1

Explanation :

Each of inner page parts contain 2^{11} entries $= 2048$ entries.

We want 2600 th page. So this page must be in 2nd partitions of inner page table. (As 2600 is greater than 2048 and less than 4096).

So we go to 2nd ~~or~~ partition or index 1 partitions. Then in the index 1 partition our requested page (2600 th) serial will be $2600 - 2048 = 552$.

from our L.A $\Rightarrow 01000101000_2 = 552_{10}$