Date: 6-12-21 :

PAGING

⇒ Allmost all modern operating systems are using virtual memory to utilize the RAM efficiently

→ Virtual memory is part of the secondary storage but acts as primary storage

→ When a process ready for execution it is not directly loaded into RAM, first the process loaded into virtual memory.

→ When process loaded into virtual memory it is divided into small units called as pages

→ physical memory (RAM) is divided into small units called as frames

→ page size must be equal to frame size

→ Logical address (or) virtual address = pageno + offset

↳ (location of instruction) with in the page

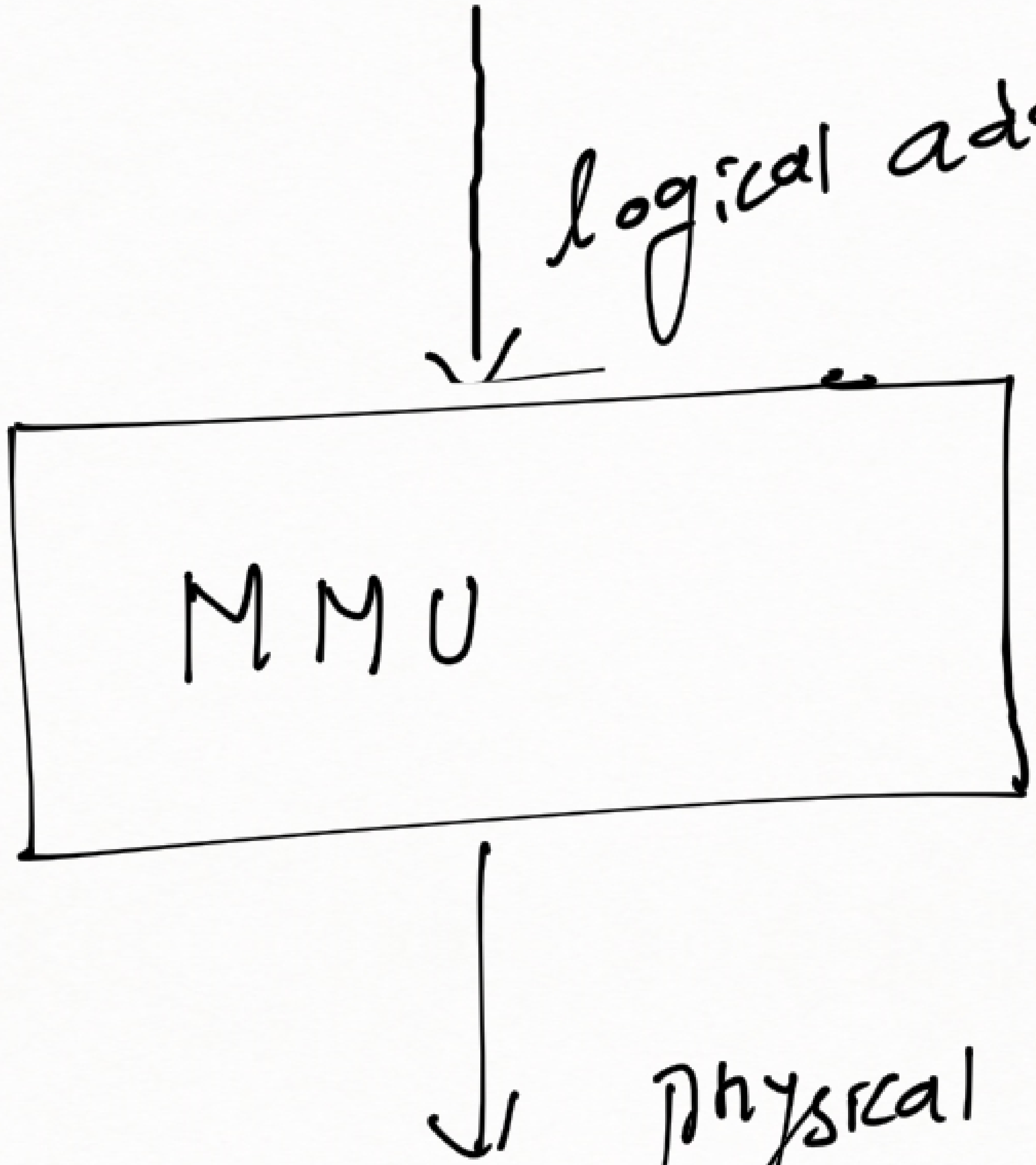
physical address (or) RAM address = frame number + offset

↳ (location of instruction within the frame)

→ PC (program counter) contains logical address but CPU needs physical address

→ logical address is converted into physical address by MMU (memory management unit)

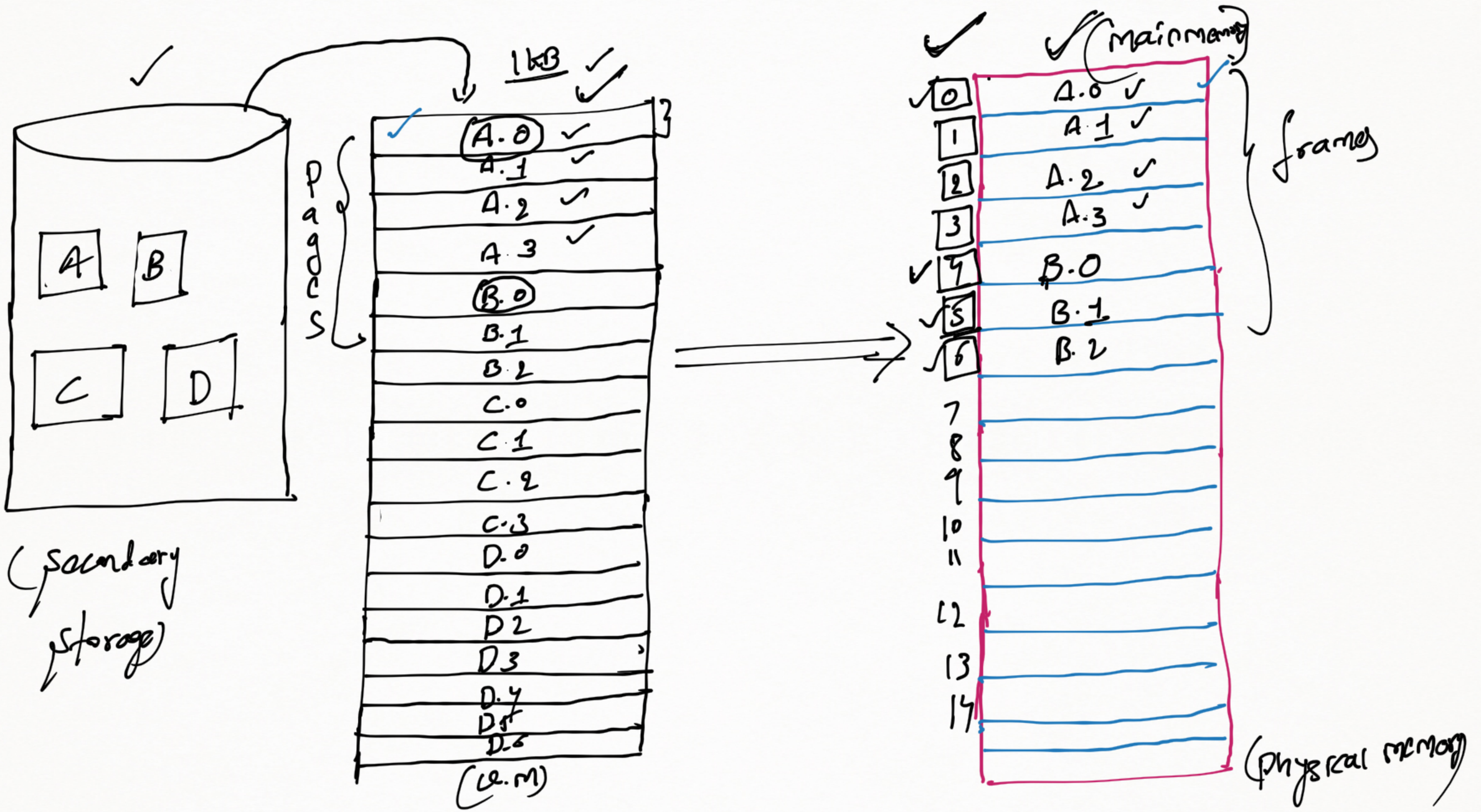
logical address → for every process page table are created

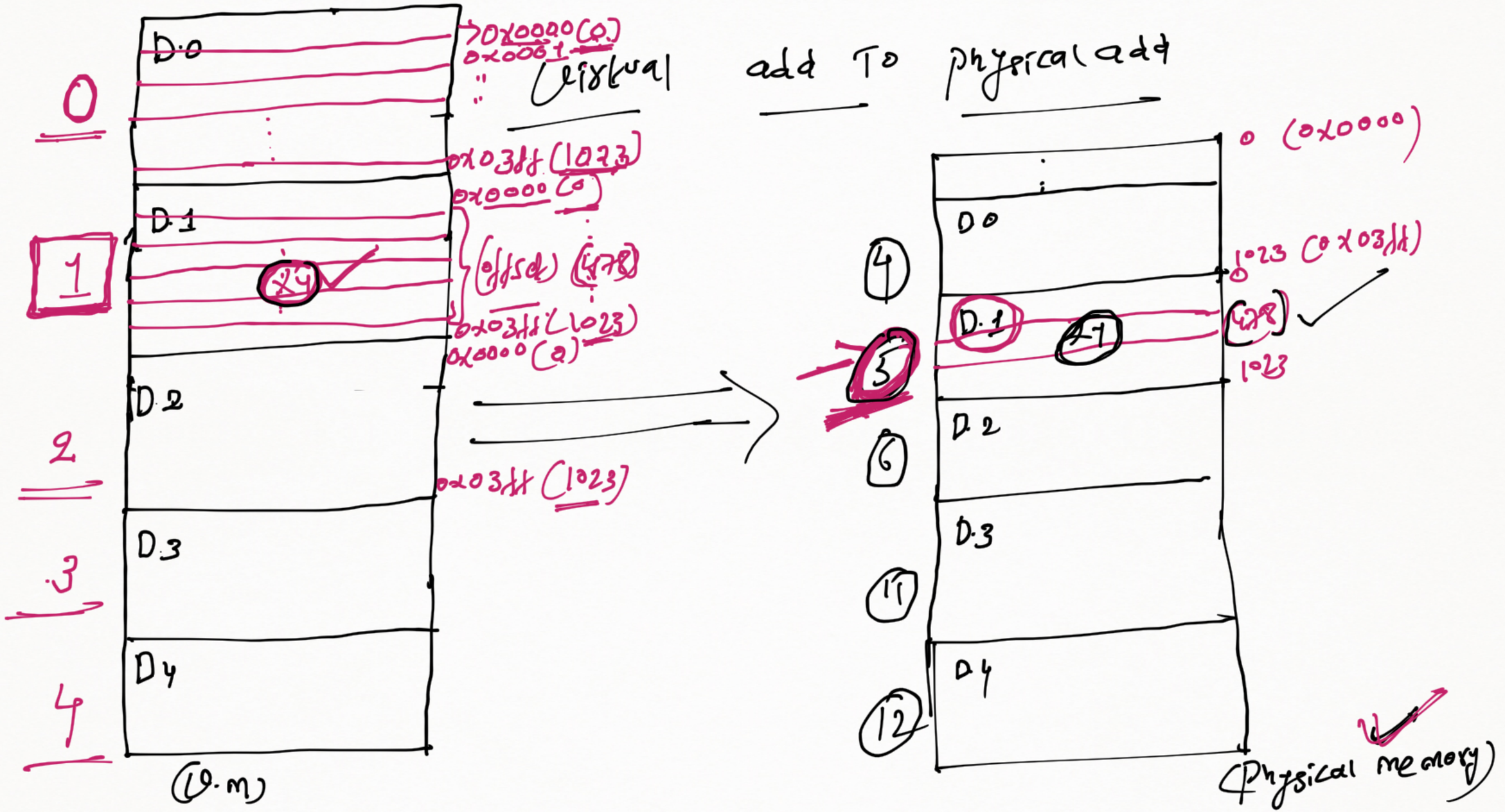


mmu will take the help of page table to convert logical add into physical add

page table contains information that which process which page load into

physical address which frame number of RAM


A
B
C
D
(secondary storage)
1kB
pages
A.0
A.1
A.2
A.3
B.0
B.1
B.2
C.0
C.1
C.2
C.3
D.0
D.1
D2
D3
D.4
D.5
D.6
(R.M)
main memory
0
1
2
3
4
5
6
A.0
A.1
A.2
A.3
B.0
B.1
B.2
7
8
9
10
11
12
13
14
frames
(physical memory)


D.0
D.1
D.2
D.3
D.4
0
1
2
3
4
0x0000 (0)
0x0001
0x03ff (1023)
0x0000 (0)
0x03ff (1023)
0x0000 (0)
0x03ff (1023)
(D.m)
add To physical add
0 (0x0000)
1023 (0x03ff)
1023
4
5
6
11
12
(Physical memory)

page size is 1KB

0 → 0000

pc reg - 16 bit