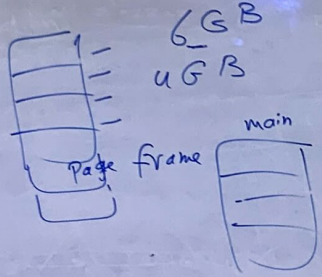
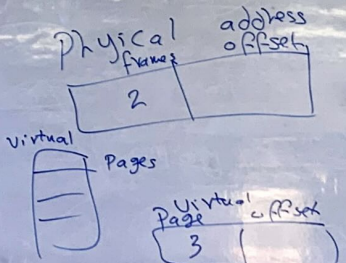
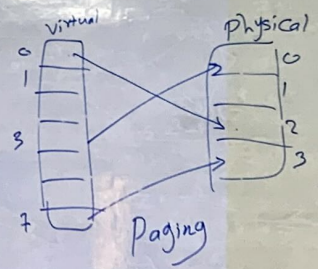
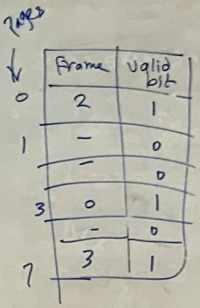
Virtual memory









Virtual address —mapping→ Physical address

v. size

ex.. virtual address space = 8 K , Physical address space = main 4K , Page size = 1K

logical address = 10 0 1 0 0 0 1 1 (13)

Virtual address bit = $2^3 * 2^{10} = 2^{13}$

n. Pages = $\frac{2^3 * 2^{10}}{2^{10}} = 2^3$ (3) = 8 Page.

bit. Pages = 3

Page	offset
3	10

Frame

Physical address 0 1 0 0 0 1 1

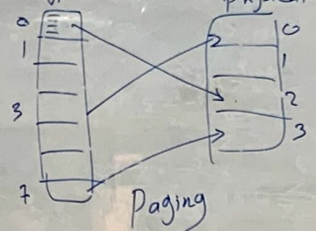
n. frames = $\frac{2^2 * 2^{10}}{2^{10}} = 2^2$ (2)

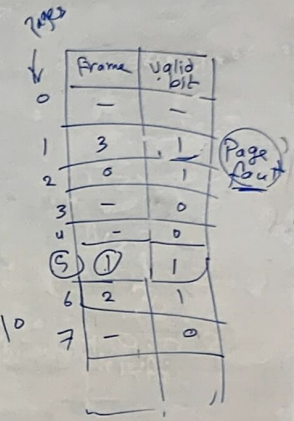
bits. frame

bit. address = 12

Frame	offset
2	10

Page table.





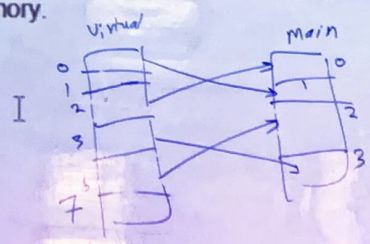
0xE	0xF	0x0FFFFF	0x0FFFFFFE
0x9	0x1	0x234567	0x23456719
0xE	[illegible]	0xCAFEBA	0xCAFEBABE

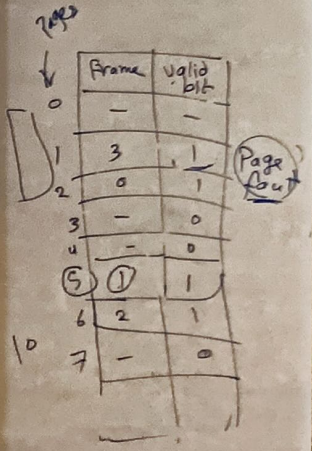
The first and last addresses [illegible] map to cache set 0xAB; the 2nd and 4th addresses both map to set 0x1. However, since the [illegible] multiple blocks in each set, these should not cause a collision. (Since cache was [illegible] empty, we don't have to worry about the set being full.)

18. Suppose a process page table contains the entries shown below. Using the format shown in Figure 6.17a, indicate where the process pages are in memory.

Valid Bit	Frame
1	1
0	–
1	0
1	3
0	–
0	–
1	2
0	–

Ans.





20. Suppose you have a byte-addressable virtual address memory system with 8 virtual pages of 64 bytes each, and 4 page frames. Assuming the following page table, answer the questions below:

Page #	Valid Bit	Frame #
0	1	1
1	0	3
2	0	-
3	1	0
4	1	2
5	0	-
6	0	-
7	0	-

Page | offset
3 | 6
Virtual

Physical
2 | 6

a) How many bits are in a virtual address?

b) How many bits are in a physical address?

c) What physical address corresponds to the following virtual addresses (if the address causes a page fault, simply indicate this is the case)?

i) 0x00 = 0100 0000
ii) 0x44 = 1100 0100
iii) 0xC2 = 0010 0010
iv) 0x80 Page fault.