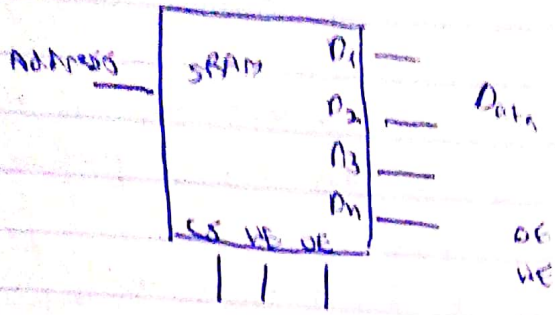
Comp Arch hw #1

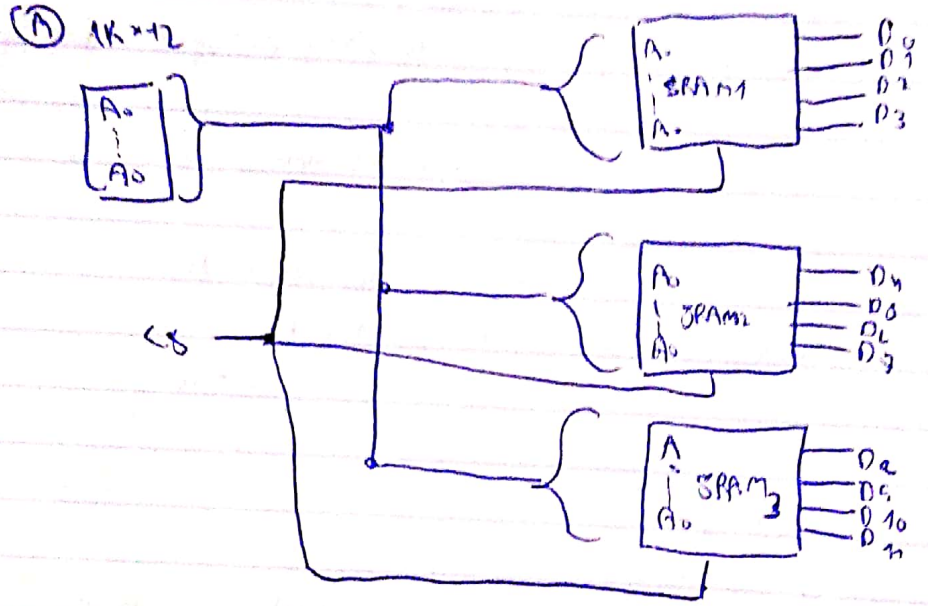
1) Because there is 1K address, we need 10 bits for an address



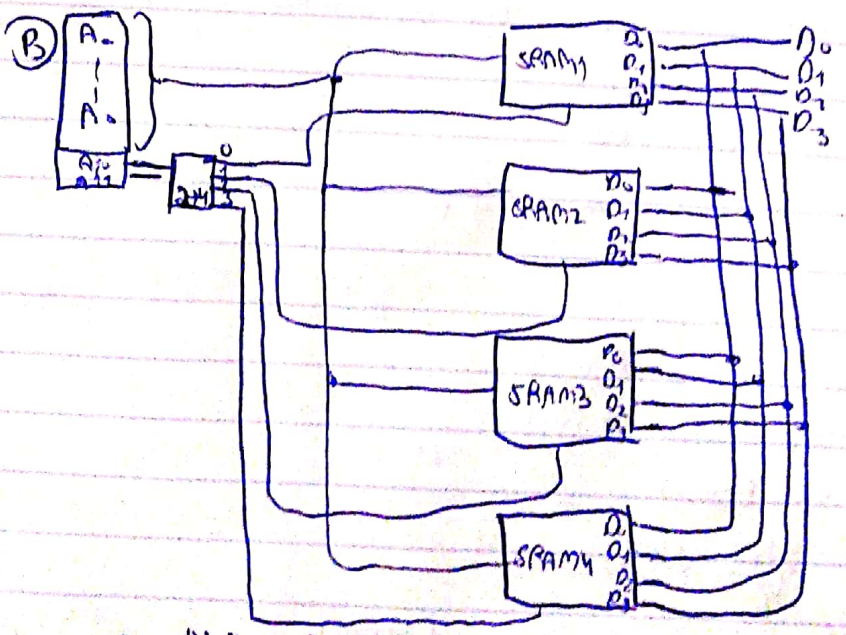
OE - Read
WE - Write

CS=0 - Data is stored
CS=1 - Data won't.

A) 1K×12



B)



didn't write OE & WE but they still exist

Size:

② Cache 1024

main memory 16mB

block 2 data

data 16 bits

1 way: because there is 2 data in each block, offset is one bit.

cache has 2^{10} blocks = 2^{10} lines in cache, therefore the size of the set will be 10 blocks.

Tag is 24 (size of the address) less, meaning 24-11 = 13

Tag 13bit, set - 10bit, offset - 1bit.

another way: size of the offset here is also 1

cache has 2^{10} blocks, and thats why there is 2^9 lines in cache

size of set = 9 bits

tag = bits that are left in the address

24 - 10 = 14

Tag - 14bit, set - 9bit, offset - 1bit

each block in cache: 2x16 bits for 1.

tag 14 bits valid bit + dirty bit = 48 bits

③ Block size: 8 bytes

main memory 256 bytes

cache 64 bytes write LRU yes write allocate write back

Miss penalty = 4 clock cycles

wanna write to address 232

$\frac{232}{8} = 29$ getting rid of A.

address size = 8 bits

each block 8 bytes so offset = 3

each way $\frac{8}{2}$ blocks. 4 rows

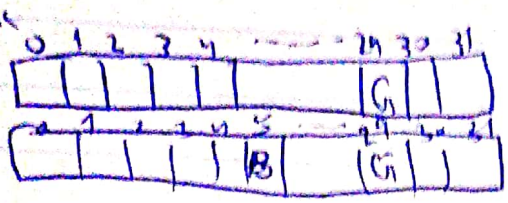
set: 2 tag: 3

$232_{10} = 1110\ 1000$

232 is block number 29_{10}. $(11101)_2$

(A) before

after



(B)

	D	V	Tag	Data
00	0	1	0	A
01	1	1	7	G
10	1	1	5	D
11	1	1	4	F

Way 0

D	V	tag	data
0	0	x	x
0	1	2	C
1	1	3	E
0	0	x	x

Way 1

From what were shown we need 4 clock cycles 4 to bring B to the main cache, 4 to bring 29 from main to cache, and one more to bring write CPU to cache 252

1	5	0	1	m	0	From what's given the line doesn't exist.
2	7	0	1	H	0	In the previous command we brought a line containing both address 7 (x 4 x 6)
3	1	0	0	m	0	From what's given the line doesn't exist
4	4	0	1	H	0	with the data is given in 1
5	36	1	1	m	1	data is not found. It's already given to us Set: 1 way: 1 (in 1)
6	8	0	2	m	0	from what's given the line doesn't exist
7	100	3	1	m	0	by the last time that we got Set: 1, the line was brought to way1 therefore, according to LRU we need to go to way0
8	6	0	1	m	1	even though the line is ~~containing~~ containing address 6, it's shown in (2), but in (7) we switched it with a new line, therefore we must in the last line that we got set: 1, the line was brought to way0, therefore according to LRU we need to go to way1