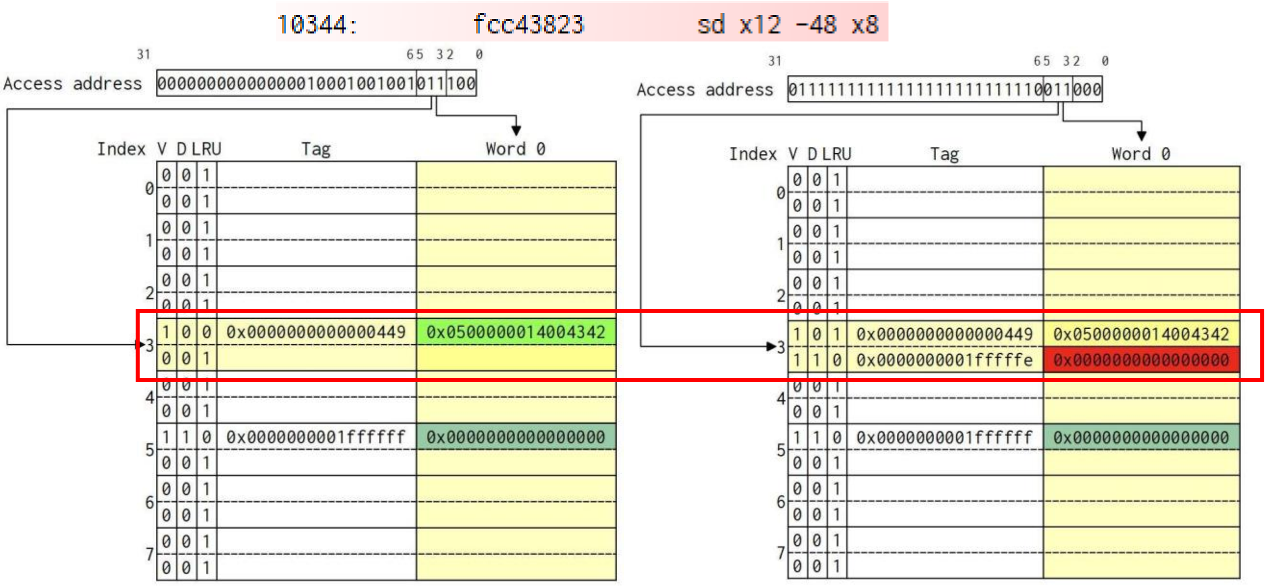
Q1 (18 points) The answer is not unique.

(a) (6 points) One case of cache insertion at an index (a set) with exactly one way already occupied. Don't forget to explain the change of LRU bits in both ways.



The case is a write miss at index 3 which has exactly one way already occupied. We fetch the block since write allocate. And, after we inserted to the vacant way, V and D of that way became 1. For LRU bits of both two ways, LRU of the way we inserted will be 0 while the other will be 1 to represent that the newly inserted way is most-recently used.

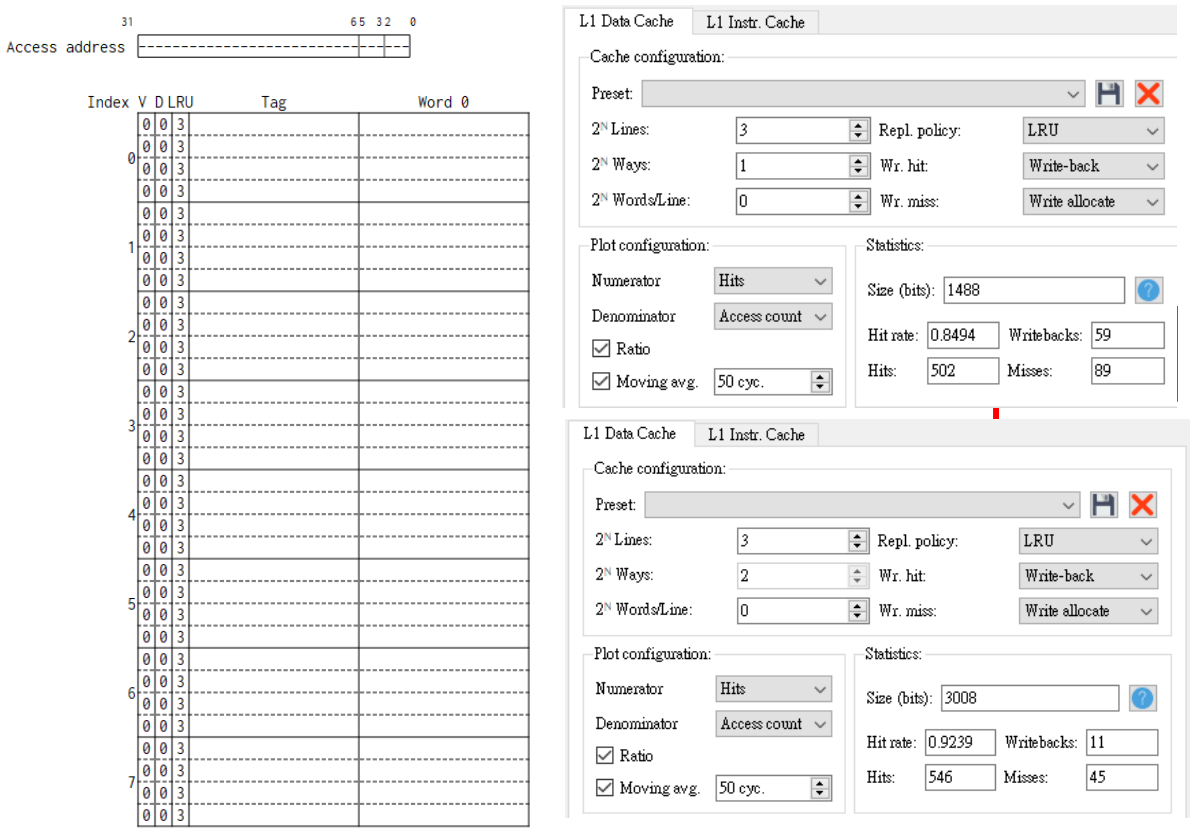
(b) (6 points) One case of a hit that makes a block become dirty. (Not dirty before that hit)

The case is a write hit at index 3 that makes the corresponding block become dirty. (D changes from 0 to 1) LRU bit also changes from 1 to 0, which means the block is just used. After the block becomes dirty, write-back will be needed if the block is going to be replaced.

(c) (6 points) One case of replacement with write-back needed.

The case is a write miss at index 4, and the replaced block needs to be written back to the memory since it is dirty. Also, you can find that the LRU of the replaced block is 1, which means it is least-recently used among two ways of index 4. And that is the reason why the block is chosen to be replaced.

Q2 (6 points) The answer is not unique



As the pictures show, I changed the cache to be a 4-way set associative and left other settings unchanged. As a result, you can see that the hit rate has improved from 0.84 to 0.92. The reason why increasing the associativity can improve the hit rate is that we will have more choices (blocks) when accessing the same index with higher associativity. Therefore, the number of write-backs and misses are both less than the original ones, and the hit rate goes up.

Q3 (36 points)

(a) (10 points)

	Address format			Cache size
Cache Type	Tag (bits)	Index (bits)	Block offset (bits)	Total size (bits)
Cache 1	20 – 4 = 16	4	0	$(1+16+32)\times 16=784$
Cache 2	20 – 2 – 2 = 16	2	2	$(1+16+128)\times 4=580$
Cache 3	20 – 2 – 1 = 17	2	1	$(1+17+64)\times 2\times 4=656$
Cache 4	20 – 2 = 18	0	2	$(1+18+128)\times 4=588$

(b) (26 points)

(i) (18 points)

- Cache 2 (6 points)

Word address	Block address	Tag	Index	Hit or Miss	Miss
16	4	1	0	Miss	Compulsory
17	4	1	0	Hit	
18	4	1	0	Hit	
19	4	1	0	Hit	
20	5	1	1	Miss	Compulsory
48	12	3	0	Miss	Compulsory
49	12	3	0	Hit	
17	4	1	0	Miss	Conflict
48	12	3	0	Miss	Conflict
49	12	3	0	Hit	
17	4	1	0	Miss	Conflict
5	1	0	1	Miss	Compulsory
6	1	0	1	Hit	
7	1	0	1	Hit	

- Cache 3 (6 points)

Word address	Block address	Tag	Index	Hit or Miss	Miss
16	8	2	0	Miss	Compulsory
17	8	2	0	Hit	
18	9	2	1	Miss	Compulsory
19	9	2	1	Hit	
20	10	2	2	Miss	Compulsory
48	24	6	0	Miss	Compulsory
49	24	6	0	Hit	
17	8	2	0	Hit	

48	24	6	0	Hit	
49	24	6	0	Hit	
17	8	2	0	Hit	
5	2	0	2	Miss	Compulsory
6	3	0	3	Miss	Compulsory
7	3	0	3	Hit	

- Cache 4 (6 points)

Word address	Block address	Tag	Index	Hit or Miss	Miss
16	4	4	*X*	Miss	Compulsory
17	4	4	*X*	Hit	
18	4	4	*X*	Hit	
19	4	4	*X*	Hit	
20	5	5	*X*	Miss	Compulsory
48	12	12	*X*	Miss	Compulsory
49	12	12	*X*	Hit	
17	4	4	*X*	Hit	
48	12	12	*X*	Hit	
49	12	12	*X*	Hit	
17	4	4	*X*	Hit	
5	1	1	*X*	Miss	Compulsory
6	1	1	*X*	Hit	
7	1	1	*X*	Hit	

(ii) (8 points)

- Cache 2 (3 points)

Cache	
Block address	Content
0	Mem[16], Mem[17], Mem[18], Mem[19]
1	Mem[4], Mem[5], Mem[6], Mem[7]
2	
3	

- Cache 3 (5 points)

Cache		
Set	Block 0 Content	Block 1 Content
0	Mem[16], Mem[17]	Mem[48], Mem[49]
1	Mem[18], Mem[19]	Mem[4], Mem[5]
2	Mem[20], Mem[21]	
3	Mem[6], Mem[7]	

Q4 (12 points)

(a) (4 points)

p1	p2	d1	p4	d2	d3	d4	p8
1	1	0	1	1	0	0	0

$h1 = p1 \oplus d1 \oplus d2 \oplus d4 = 0$

$h2 = p2 \oplus d1 \oplus d3 \oplus d4 = 1$

$h4 = p4 \oplus d2 \oplus d3 \oplus d4 = 0$

(h4,h2,h1) = 010

The 2^{th} bit from the left is in error.

(b) (4 points)

p1	p2	d1	p4	d2	d3	d4	p8
1	0	1	1	0	1	0	1

$h1 = p1 \oplus d1 \oplus d2 \oplus d4 = 0$

$h2 = p2 \oplus d1 \oplus d3 \oplus d4 = 0$

$h4 = p4 \oplus d2 \oplus d3 \oplus d4 = 0$

$h8 = p8 \oplus p1 \oplus p2 \oplus p4 \oplus d1 \oplus d2 \oplus d3 \oplus d4 = 1$

(h8,h4,h2,h1) = 1000

The 8^{th} bit from the left is in error.

(c) (4 points)

p1	p2	d1	p4	d2	d3	d4	p8
1	0	0	0	1	0	1	1

$h1 = p1 \oplus d1 \oplus d2 \oplus d4 = 1$

$h2 = p2 \oplus d1 \oplus d3 \oplus d4 = 1$

$h4 = p4 \oplus d2 \oplus d3 \oplus d4 = 0$

$h8 = p8 \oplus p1 \oplus p2 \oplus p4 \oplus d1 \oplus d2 \oplus d3 \oplus d4 = 0$

The data has double errors.

Q5 (10 points)

(a) (6 points)

	Miss penalty	Total miss cycles
Cache 1	5+4 = 9	70000*(0.04*9+1/3*0.03*9) = 31500
Cache 2	5+2 = 7	70000*(0.04*7+1/3*0.03*7) = 24500

(b) (4 points)

(c)	Miss penalty	Total miss cycles
Cache 3	5+1 = 6	70000*(0.04*6+1/3*0.03*6) = 21000

	Stall cycle/instruction
Cache 1	31500/70000 = 0.45
Cache 2	24500/70000 = 0.35
Cache 3	21000/70000 = 0.3

CPI_{base} = CPI - $CPI_{misses\ with\ cache\ 3}$ = 1.7 – 0.3 = 1.4

$CPI_{with\ cache\ 1}$ = 1.4 + 0.45 = 1.85

$CPI_{with\ cache\ 2}$ = 1.4 + 0.35 = 1.75

Q6 (18 points) TLB hit check the page number of the virtual address, cache hit check the index of the physical address, and then compares the tag

Virtual address	Physical address	Cache Hit/Miss	TLB Hit/Miss
0x954a16c2	0x3b9416c2	Miss	Miss
0x6542c746	0x0ae6c746	Miss	Miss
0x954a1647	0x3b941647	Hit	Hit
0x6542c412	0x0ae6c412	Miss	Hit
0x2b74c4d3	0x14acc4d3	Miss	Miss
0x6542c46a	0x0ae6c46a	Miss	Hit
0x954a16dd	0x3b9416dd	Hit	Miss
0x6542c417	0x0ae6c417	Hit	Hit
0x2b74c723	0x14acc723	Miss	Miss

TLB

Tag	Physical page addr.
~~954a~~ ~~2b74~~ ~~954a~~ 2b74	~~3b94~~ ~~14ac~~ ~~3b94~~ 14ac
6542	0ae6

Cache

	Tag	cache index	H/M	cache content					
				16	…	C7	…	C4	…
(1)	3b94	16	M	(1)					
(2)	0ae6	C7	M	(1)		(2)			
(3)	3b94	16	H	(1)		(2)			
(4)	0ae6	C4	M	(1)		(2)		(4)	
(5)	14ac	C4	M	(1)		(2)		(5)	
(6)	0ae6	C4	M	(1)		(2)		(6)	
(7)	3b94	16	H	(1)		(2)		(6)	
(8)	0ae6	C4	H	(1)		(2)		(6)	
(9)	14ac	C7	M	(1)		(9)		(6)	