計節 HW6

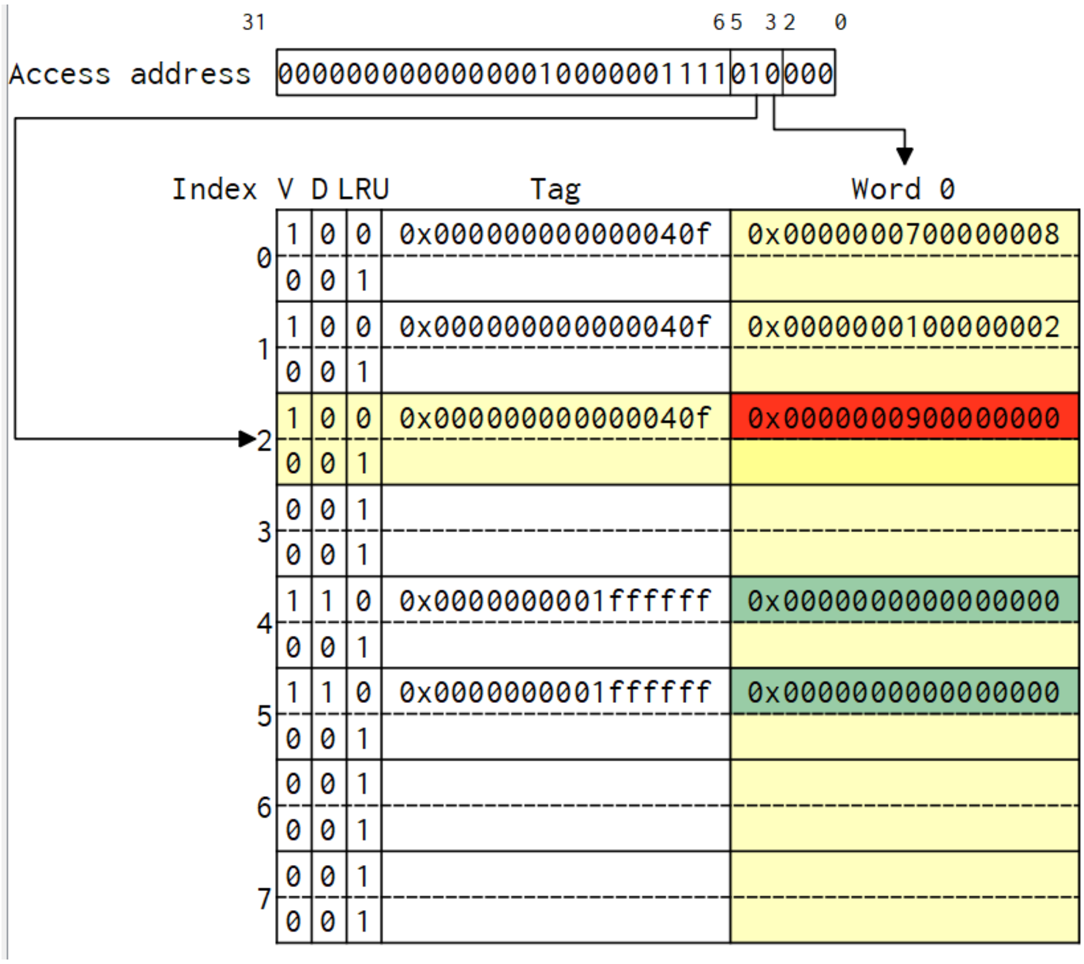
學號：111062107

姓名：鄧弘利

1.

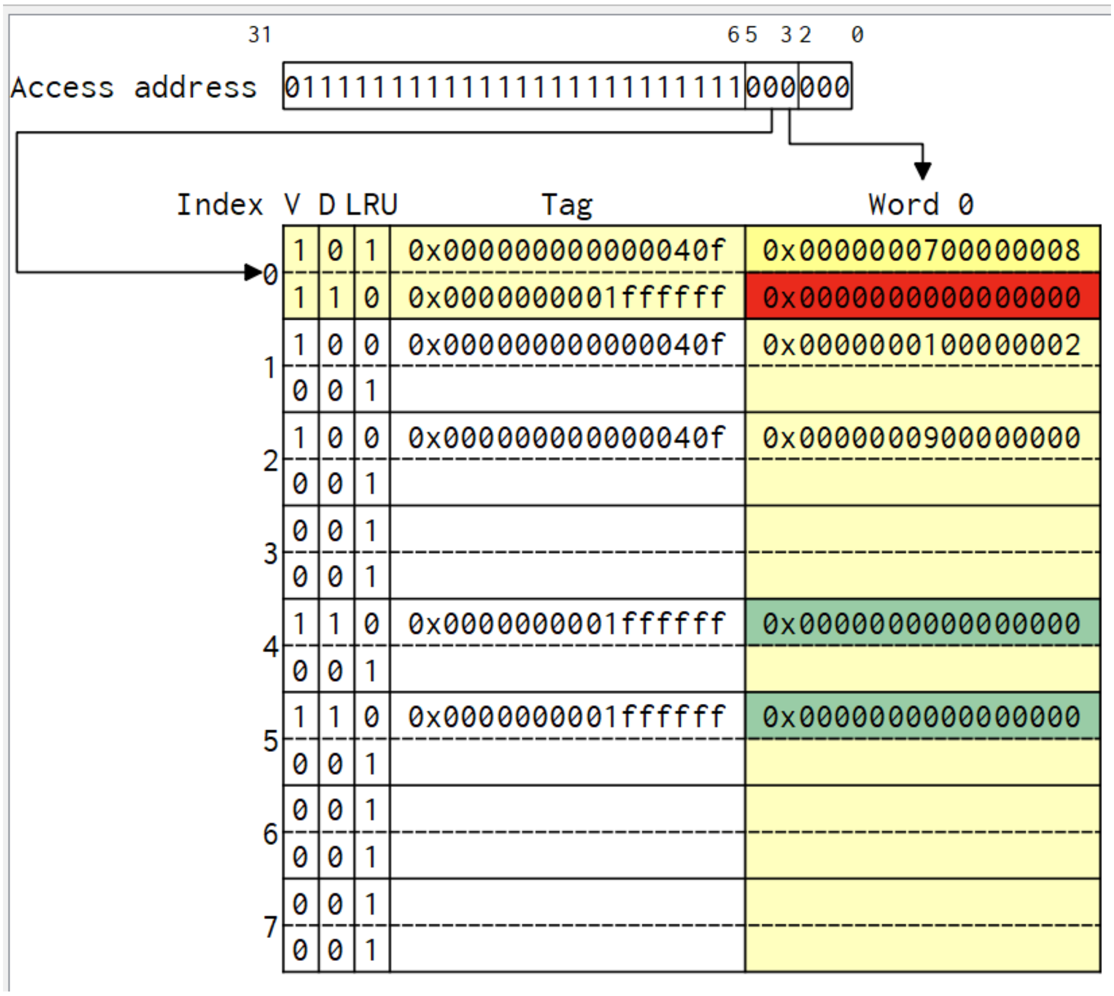
(a)

Before



10344: fcc43823 sd x12 -48 x8 MEM

After



Explanation：

With write allocate, we fetch a block on a write miss.

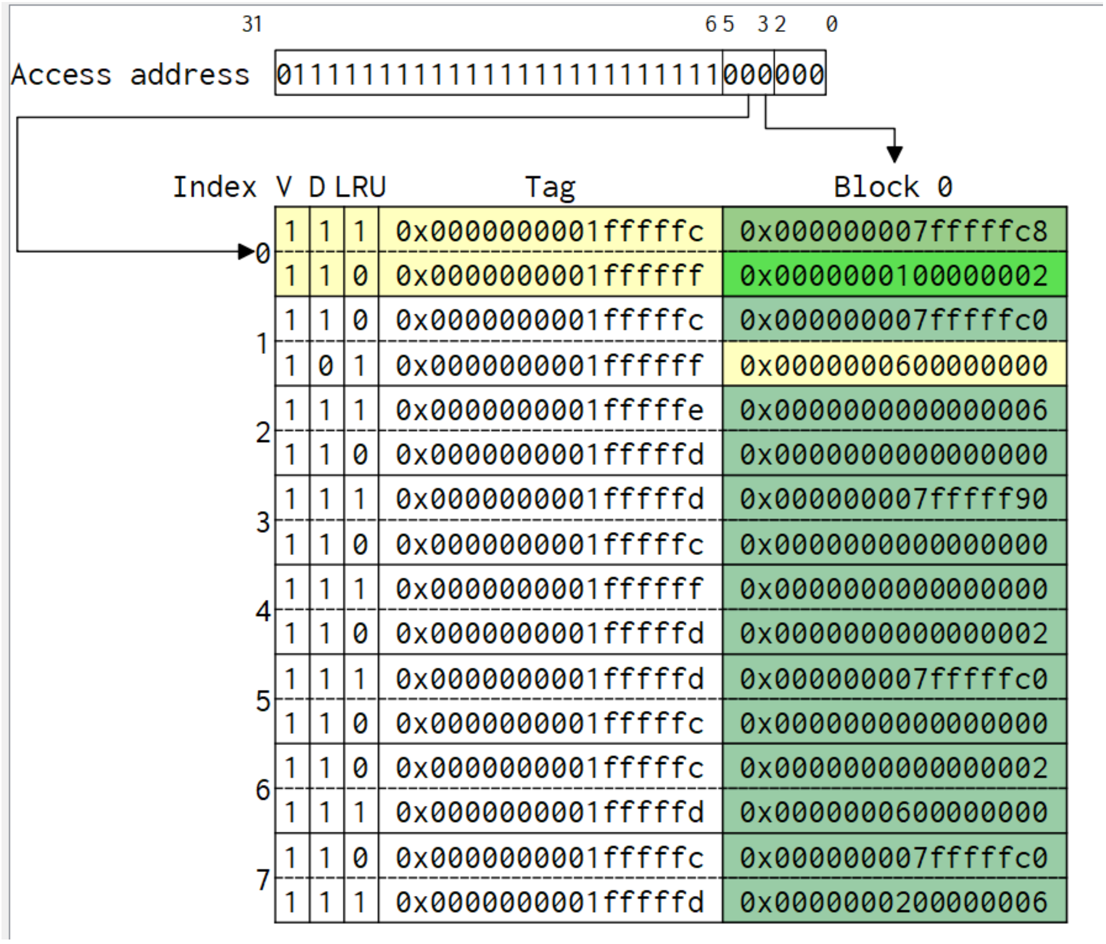
So after the write miss,the V and D bit will be 1, and the LRU bit will become 0 to represent that which block is the most recently used.

(b)

Before

`100dc:    00e7a023    sw x14 0 x15`

After



Index	V	D	LRU	Tag	Block 0
0	1	1	1	0x0000000001fffffc	0x000000007fffffc8
	1	1	0	0x0000000001ffffff	0x0000000100000002
1	1	1	0	0x0000000001fffffc	0x000000007fffffc0
	1	0	1	0x0000000001ffffff	0x0000000600000000
2	1	1	1	0x0000000001fffffe	0x0000000000000006
	1	1	0	0x0000000001fffffd	0x0000000000000000
3	1	1	1	0x0000000001fffffd	0x000000007ffff90
	1	1	0	0x0000000001fffffc	0x0000000000000000
4	1	1	1	0x0000000001ffffff	0x0000000000000000
	1	1	0	0x0000000001fffffd	0x0000000000000002
5	1	1	1	0x0000000001fffffd	0x000000007fffffc0
	1	1	0	0x0000000001fffffc	0x0000000000000000
6	1	1	0	0x0000000001fffffc	0x0000000000000002
	1	1	1	0x0000000001fffffd	0x0000000600000000
7	1	1	0	0x0000000001fffffc	0x000000007fffffc0
	1	1	1	0x0000000001fffffd	0x0000000200000006

Explanation

This is a write hit.

So the dirty bit becomes 1 ,which means that it is required to write back to memory when it's going to be replaced. Valid remains 1.LRU becomes 0 since this block becomes the most recently one that is used in this set.

(c)

Before

	31	6 5	3 2 0
Access address	011111111111111111111111101	101	000

Index	V	D	LRU	Tag	Block 0
0	1	1	1	0x0000000001ffffe	0x000000007ffffc0
	1	1	0	0x0000000001fffff	0x0000000700000008
1	1	1	1	0x0000000001ffffe	0x0000000000010270
	1	1	0	0x0000000001fffff	0x0000000100000002
2	1	1	0	0x0000000001ffffe	0x0000000000000006
	1	1	1	0x0000000001fffff	0x0000000900000000
3	1	1	1	0x0000000001ffffe	0x000000007ffffc0
	1	1	0	0x0000000001fffff	0x0000000700000006
4	1	1	1	0x0000000001fffff	0x0000000000000000
	1	1	0	0x0000000001ffffd	0x0000000000000006
5	1	1	0	0x0000000001ffffd	0x000000007ffffc0
	1	1	1	0x0000000001ffffe	0x0000000700000000
6	1	1	1	0x0000000001ffffe	0x000000007fffff0
	1	1	0	0x0000000001ffffd	0x0000000600000000
7	1	1	1	0x0000000001ffffe	0x0000000000010394
	1	1	0	0x0000000001ffffd	0x0000000000000002

```
100b4:        02813423        sd x8 40 x2
```

After

Index	V	D	LRU	Tag	Block 0
0	1	1	1	0x0000000001fffffe	0x000000007fffffc0
	1	1	0	0x0000000001ffffff	0x0000000700000008
1	1	1	1	0x0000000001fffffe	0x0000000000010270
	1	1	0	0x0000000001ffffff	0x0000000100000002
2	1	1	0	0x0000000001fffffe	0x0000000000000006
	1	1	1	0x0000000001ffffff	0x0000000900000000
3	1	1	0	0x0000000001fffffd	0x0000000000000000
	1	1	1	0x0000000001ffffff	0x0000000700000006
4	1	1	1	0x0000000001ffffff	0x0000000000000000
	1	1	0	0x0000000001fffffd	0x0000000000000006
5	1	1	0	0x0000000001fffffd	0x000000007fffffc0
	1	1	1	0x0000000001fffffe	0x0000000700000000
6	1	1	1	0x0000000001fffffe	0x000000007ffffff0
	1	1	0	0x0000000001fffffd	0x0000000600000000
7	1	1	1	0x0000000001fffffe	0x0000000000010394
	1	1	0	0x0000000001fffffd	0x0000000000000002

Explanation

There's a new data for index 3, but index 3 has no space for that data , we need to replace the block which LRU bit is 1(least recently used).

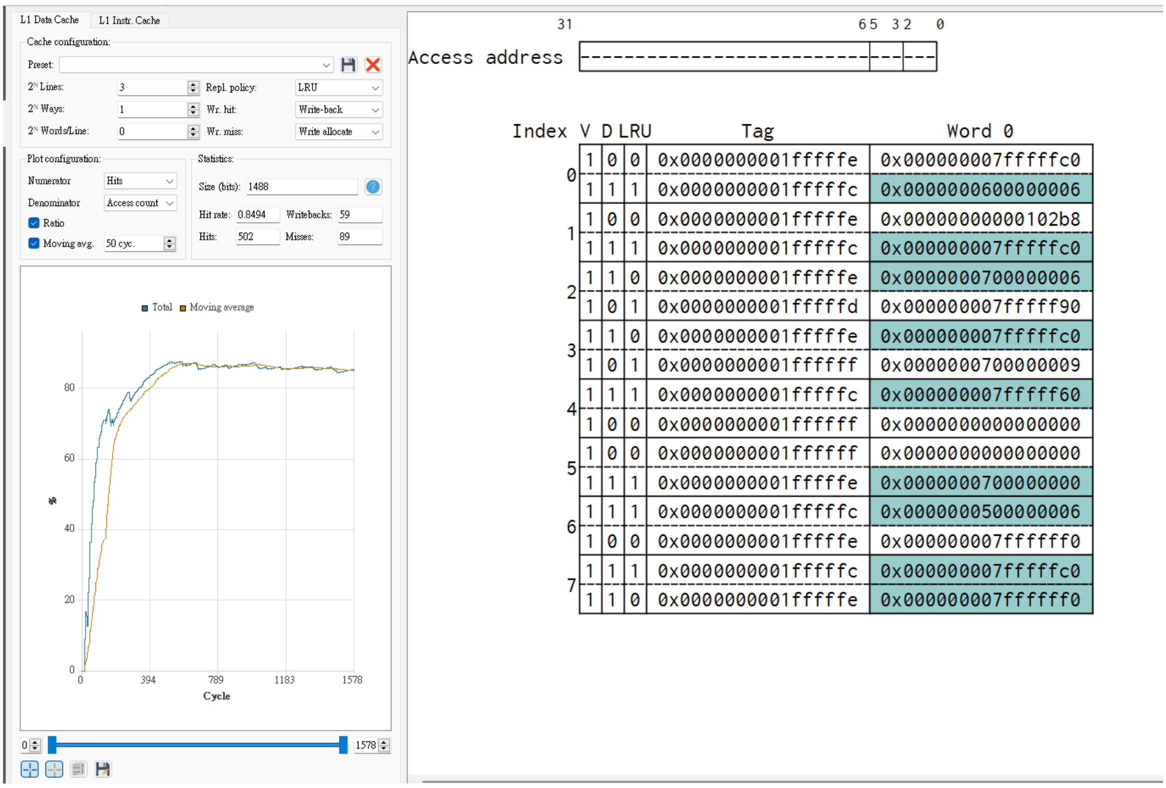
Also we need to check the dirty bit, if it's 1 , we need to write back to memory then replace it with the new one.

The valid bit (V) and dirty bit (D) will be 1, and the LRU bit will become 0 to represent that the block is the most recently used.

(d)

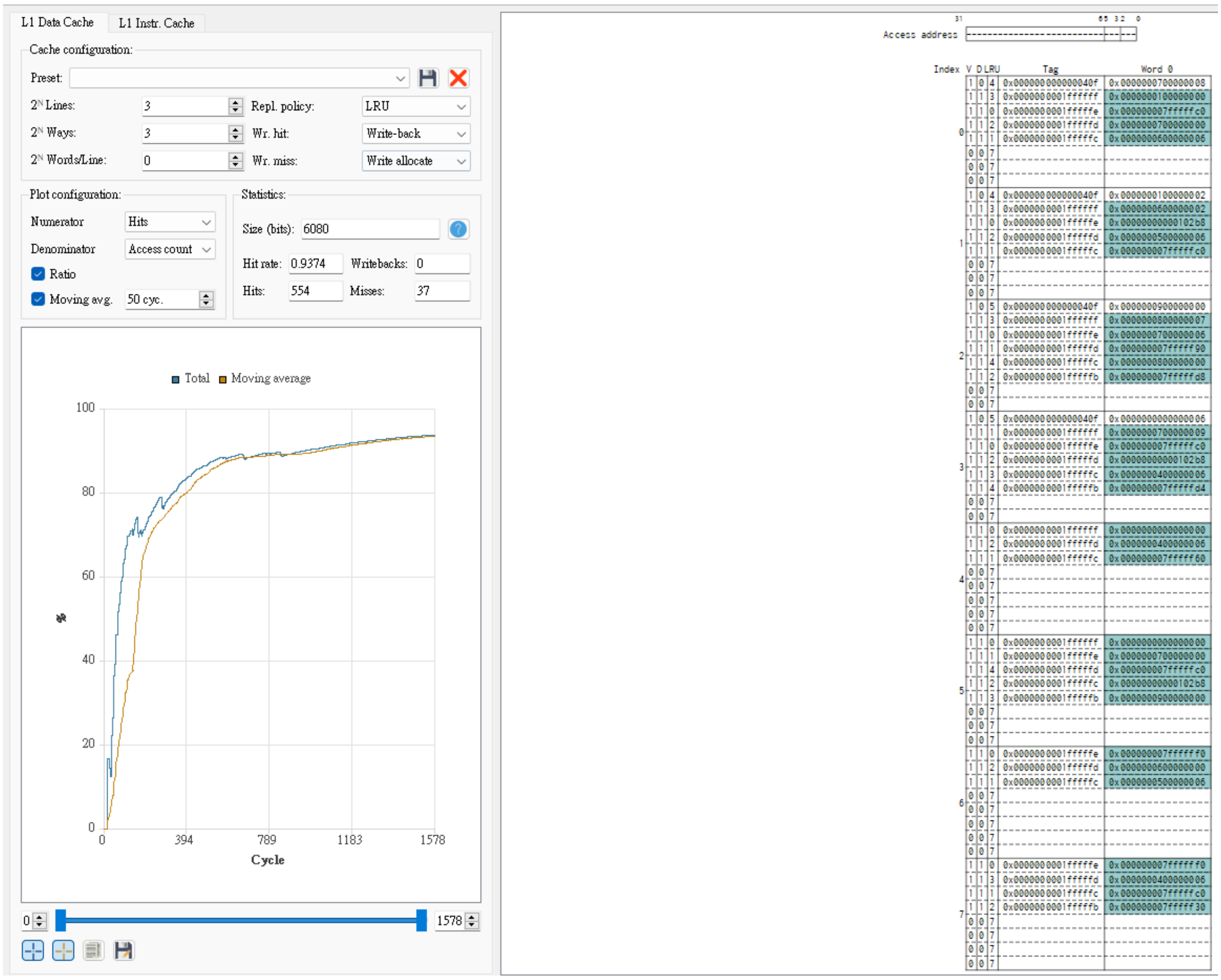
Before

Hit rate：0.8494



After

Hit rate：0.9374



I let it becomes 8 way.

Because this program does not need too many memory,so if we adjust it to 8 ways, the main miss is then becomes compulsory miss.

2.

(a)

1 1 1 0 0 0 1 1 0 1 1 0 0 1 P_{11}

Bit position		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Encoded data bits		p1	p2	d1	p4	d2	d3	d4	p8	d5	d6	d7	d8	d9	d10	d11	~~p16~~
Parity bit coverage	p1	✓		✓		✓		✓		✓		✓		✓		✓	
	p2		✓	✓			✓	✓			✓	✓			✓	✓	
	p4				✓	✓	✓	✓					✓	✓	✓	✓	
	p8								✓	✓	✓	✓	✓	✓	✓	✓	
	~~p16~~																~~✓~~

$p_1 \Rightarrow p_1, 1, 0, 1, 0, 1, 0$

$\Rightarrow p_1 = 1$

$p_2 \Rightarrow p_2, 1, 0, 1, 1, 1, 1$

$\Rightarrow p_2 = 1$

$p_4 \Rightarrow p_4, 0, 0, 1, 0, 0, 1$

$\Rightarrow p_4 = 0$

$p_8 \Rightarrow p_8, 0, 1, 1, 0, 0, 1$

$\Rightarrow p_8 = 1$

$\therefore C = \underbrace{1\,1\,1\,0\,0\,0\,1\,1\,0\,1\,1\,0\,0\,1}_{\text{even}}\ 0$ (P_{11})

(b) $h_1 \Rightarrow$ check the the position of 1, 3, 5, 7, 9, 11, 13 bit positions of C

$\Rightarrow$ the number of 1 in these positions is 4 which is even.

$\Rightarrow h_1 = 0$

$h_2 \Rightarrow$ check 2, 3, 6, 7, 10, 11, 14

$\Rightarrow$ number of 1 is 6

$\Rightarrow h_2 = 0$

$h_4 \Rightarrow$ check 4, 5, 6, 7, 12, 13, 14

$\Rightarrow$ number of 1 is 2

$\Rightarrow h_4 = 0$

- Decoding: let **H = $h_8h_4h_2h_1$** and **h_n** = parity check of whole word
  - **H = 0000, h_n even** → no error
  - **H = 0000, h_n odd** → error in p_n bit
  - **H ≠ 0000, h_n odd** → correctable single bit error (as in SEC)
  - **H ≠ 0000, h_n even** → double error occurred

$h_8 \Rightarrow$ check 8, 9, 10, 11, 12, 13, 14

$\Rightarrow$ number of 1 is 4

$\Rightarrow h_8 = 0$

$\therefore H = h_8h_4h_2h_1 = 0000_2 = 0_{10}$

$h_n = 0 \Rightarrow h_n$ even

$\Rightarrow$ we can know there's no error #

$d_5 \Rightarrow$ position 9 $\quad 0 \to 1$

(c) $h_1 \Rightarrow$ check the the position of 1, 3, 5, 7, 9, 11, 13 bit positions of C

$\Rightarrow$ the number of 1 in these positions is $\cancel{4}$ 5 which is even.

$\Rightarrow h_1 = \cancel{0}\ 1$

$h_2 \Rightarrow$ check 2, 3, 6, 7, 10, 11, 14

$\Rightarrow$ number of 1 is 6

$\Rightarrow h_2 = 0$

$h_4 \Rightarrow$ check 4, 5, 6, 7, 12, 13, 14

$\Rightarrow$ number of 1 is 2

$\Rightarrow h_4 = 0$

$h_8 \Rightarrow$ check 8, 9, 10, 11, 12, 13, 14

$\Rightarrow$ number of 1 is $\cancel{4}$ 5

$\Rightarrow h_8 = \cancel{0}\ 1$

position 9

$\therefore H = h_8 h_4 h_2 h_1 = 1001_2 = 9_{10} \neq 0_{10}$

$h_n = \cancel{0}\ 1 \Rightarrow h_n$ ~~even~~ odd

$\Rightarrow$ correctable single bit error #

(d) $\boxed{P_1, d_8 \Rightarrow \text{position } 1, 12 \text{ of } C \quad P_1 \Rightarrow 1 \to 0 \quad d_8 \Rightarrow 0 \to 1}$

$h_1 \Rightarrow$ check the the position of 1, 3, 5, 7, 9, 11, 13 bit positions of C

$\Rightarrow$ the number of 1 in these positions is $\cancel{4}$ 3 which is even.

$\Rightarrow h_1 = \cancel{0}\ 1$

$h_2 \Rightarrow$ check 2, 3, 6, 7, 10, 11, 14

$\Rightarrow$ number of 1 is 6

$\Rightarrow h_2 = 0$

$h_4 \Rightarrow$ check 4, 5, 6, 7, 12, 13, 14

$\Rightarrow$ number of 1 is $\cancel{2}$ 3

$\Rightarrow h_4 = \cancel{0}\ 1$

$h_8 \Rightarrow$ check 8, 9, 10, 11, 12, 13, 14

$\Rightarrow$ number of 1 is $\cancel{4}$ 5

$\Rightarrow h_8 = \cancel{0}\ 1$

$\therefore H = h_8 h_4 h_2 h_1 = 1101_2 \neq 0_{10}$

$h_n = 0 \Rightarrow h_n$ even

$\Rightarrow$ double error occured #

e. p_1, d_8, p_{11} ⇒ position 1, 12, 15

$1 \to 0$ $0 \to 1$ $0 \to 1$

h_1 ⇒ check the the position of 1, 3, 5, 7, 9, 11, 13 bit positions of C

⇒ the number of 1 in these positions is ~~4~~ 3 which is even.

⇒ $h_1 = \cancel{0}\ 1$

h_2 ⇒ check 2, 3, 6, 7, 10, 11, 14

⇒ number of 1 is 6

⇒ $h_2 = 0$

h_4 ⇒ check 4, 5, 6, 7, 12, 13, 14

⇒ number of 1 is ~~2~~ 3

⇒ $h_4 = \cancel{0}\ 1$

h_8 ⇒ check 8, 9, 10, 11, 12, 13, 14

⇒ number of 1 is ~~4~~ 5

⇒ $h_8 = \cancel{0}\ 1$

∴ $H = h_8h_4h_2h_1 = 1101_2 = 13_{10}$ ⇒ we think that this is correctable single bit error at position 13, but actually 3 bit errors.

$h_n = 1$ ⇒ h_n odd

f. $2^p \geq p + d + 1$ # parity : $p = 8$

$p \geq \log(p+d+1)$ # data : $d = 128$

$\Rightarrow p \geq \log(p+129)$

$p \geq 8$

parity bit = 8

but we need 1 more bit for SECDED.

so 9 bits #

3.

(a) 16 bytes # (block offset + byte offset)

(b) 32 blocks # (index field)

(c) $16 \times 32 = 512$ bytes #

(d)

Valid	dirty	tag	data
(1-bit)	(1-bit)	(5-bit)	(16 x 8-bit)
			16 bytes

$32 \times (1+1+5+128) = 4320$ bits $= 540$ bytes #

(e)

Tag	Index	Block offset	Byte offset
13:8	7:4	3:2	1:0

(f)

Valid	dirty	Tag	data	reference
(1-bit)	(1-bit)	(6-bit)	(128-bit)	(1-bit)

2 blocks + 1 reference $= 2 \times (136) + 1 = 273$

$16 \times 273 = 4368$ bits $= 546$ bytes #

4.

(a) L2-DM $\Rightarrow 1 + 5\% \times 16 + 5\% \times 4\% \times 200 = 2.2$

L2-4WAY $\Rightarrow 1 + 5\% \times 20 + 5\% \times 3.5\% \times 200 = 2.35$

this provides the best ✓ L2-DM + L3

$\Rightarrow 1 + 5\% \times 16 + \underbrace{5\% \times 4\%}_{0.1} \times 50 + \underbrace{5\% \times 4\% \times 2\%}_{0.008} \times 200 = 1.908$

L2-4WAY + L3

2.5 × 3.5 = 125, 75, 8.75

$\Rightarrow 1 + 5\% \times 20 + \underbrace{5\% \times 3.5\%}_{0.0875} \times 50 + \underbrace{5\% \times 3.5\% \times 2\%}_{0.007} \times 200 = 2.0945$

L1 - L2DM - L38WAY provides the best performance.

(b) Let x be the new L2 access cycles

L2-DM

$\Rightarrow \underbrace{1 + 5\% \times x + \underbrace{5\% \times 4\% \times 50}_{0.1} + \underbrace{5\% \times 4\% \times 2\% \times 200}_{0.008}}_{1.108} \le 1.8$

$0.05x \le 0.692$

$x \le 13.84$ $\therefore x$ 取 13 #

5.

XXXXXX tag | XX index | XXXX offset

No1 Read 0x340

00110100 0000₂ → $00110100\ 0000_2$

Miss/hit : Miss

block replacement : No

write allocate : No

write back : No

	R	V	D	Tag	Data	V	D	Tag	Data
		Way 0				Way 1			
Set 0	0	1	0	000000_2	Mem[0x000-00f]	1	0	001101_2	MEM[0x340 -34f]
Set 1									
Set 2									
Set 3									

No2 Read 0x000

$0000\,0000\ 0000_2$

Miss/hit : hit

block replacement : No

write allocate : No

write back : No

	R	V	D	Tag	Data	V	D	Tag	Data
		Way 0				Way 1			
Set 0	1	1	0	000000_2	Mem[0x000-00f]	1	0	001101_2	Mem[0x340 -34f]
Set 1									
Set 2									
Set 3									

No3 Read 0x1d8

0001 1101 1000

Miss/hit : Miss

block replacement : No

write allocate : No

write back : No

	R	V	D	Tag	Data	V	D	Tag	Data
		Way 0				Way 1			
Set 0	1	1	0	000000_2	Mem[0x000-00f]	1	0	001101_2	Mem[0x340 -34f]
Set 1	1	1	0	000111_2	Mem[0x 1d0 - 1df]				
Set 2									
Set 3									

No4 Write 0x354

0011 0101 0100

Miss/hit : Miss

block replacement : No

write allocate : Yes

write back : No

	R	V	D	Tag	Data	V	D	Tag	Data
		Way 0				Way 1			
Set 0	1	1	0	000000_2	Mem[0x000-00f]	1	0	001101_2	Mem[0x340 -34f]
Set 1	0	1	0	000111_2	Mem[0x 1d0 - 1df]	1	1	001101_2	Mem[0x350 - 0x35f]
Set 2									
Set 3									

No5 Read 0xa61

1010 0110 0001

Miss/hit : Miss

block replacement : No

write allocate : No

write back : No

	R	Way 0 V	D	Tag	Data	Way 1 V	D	Tag	Data
Set 0	1	1	0	000000_2	Mem[0x000-00f]	1	0	001101_2	Mem[0x340 -34f]
Set 1	0	1	0	000111_2	Mem[0x1d0 - 1df]	1	1	001101_2	Mem[0x350 - 0x35f]
Set 2	1	1	0	101001_2	Mem[0xa60 - a6f]				
Set 3									

No6 Write 0xa61

1010 0110 0001

Miss/hit : Hit

block replacement : No

write allocate : No

write back : No

	R	Way 0 V	D	Tag	Data	Way 1 V	D	Tag	Data
Set 0	1	1	0	000000_2	Mem[0x000-00f]	1	0	001101_2	Mem[0x340 -34f]
Set 1	0	1	0	000111_2	Mem[0x1d0 - 1df]	1	1	001101_2	Mem[0x350 - 0x35f]
Set 2	1	1	1	101001_2	Mem[0xa60 - a6f]				
Set 3									

No7 Read 0x3ec

0011 1110 1100

Miss/hit : Miss

block replacement : No

write allocate : No

write back : No

	R	Way 0 V	D	Tag	Data	Way 1 V	D	Tag	Data
Set 0	1	1	0	000000_2	Mem[0x000-00f]	1	0	001101_2	Mem[0x340 -34f]
Set 1	0	1	0	000111_2	Mem[0x1d0 - 1df]	1	1	001101_2	Mem[0x350 - 35f]
Set 2	0	1	1	101001_2	Mem[0xa60 - a6f]	1	0	001111_2	Mem[0x3e0 - 3ef]
Set 3									

No8 Read 0xa62

1010 0110 0010

Miss/hit : Hit

block replacement : No

write allocate : No

write back : No

	R	Way 0 V	D	Tag	Data	Way 1 V	D	Tag	Data
Set 0	1	1	0	000000_2	Mem[0x000-00f]	1	0	001101_2	Mem[0x340 -34f]
Set 1	0	1	0	000111_2	Mem[0x1d0 - 1df]	1	1	001101_2	Mem[0x350 - 35f]
Set 2	1	1	1	101001_2	Mem[0xa60 - a6f]	1	0	001111_2	Mem[0x3e0 - 3ef]
Set 3									

No9 Read 0x3ea

0011 1110 1010

Miss/hit : Hit

block replacement : No

write allocate : No

write back : No

	Way 0					Way 1			
	R	V	D	Tag	Data	V	D	Tag	Data
Set 0	1	1	0	000000_2	Mem[0x000-00f]	1	0	001101_2	Mem[0x340 - 34f]
Set 1	0	1	0	000111_2	Mem[0x1d0 - 1df]	1	1	001101_2	Mem[0x350 - 35f]
Set 2	0	1	1	101001_2	Mem[0xa60 - a6f]	1	0	001111_2	Mem[0x3e0 - 3ef]
Set 3									

No10 Read 0x422

0100 0010 0010

Miss/hit : Miss

block replacement : Yes

write allocate : No

write back : Yes (set 2 way 0)

	Way 0					Way 1			
	R	V	D	Tag	Data	V	D	Tag	Data
Set 0	1	1	0	000000_2	Mem[0x000-00f]	1	0	001101_2	Mem[0x340 - 34f]
Set 1	0	1	0	000111_2	Mem[0x1d0 - 1df]	1	1	001101_2	Mem[0x350 - 35f]
Set 2	1	1	0	010000_2	Mem[0x420 - 42f]	1	0	001111_2	Mem[0x3e0 - 3ef]
Set 3									

6.

(a) $512 = 2^9$ $\therefore$ 9 bits

(b) $16 - 9 = 7$

$\therefore 2^7 = 128$ (pages) #

(c) $14 - 9 = 5$

$\therefore 2^5 = 32$ (pages) #

(d) since it could have 2^7 pages and each page is 4 bytes,

so $2^7 \times 4 = 512$ bytes #

(e) $\underbrace{1}_{\text{valid}} + \underbrace{1}_{\text{reference}} + \underbrace{1}_{\text{dirty}} + \underbrace{(7-4)}_{\text{tag}} + \underbrace{5}_{\text{bit number of physical page number}} = 11$

$\Rightarrow 16 \times 11 = 176$ bits #

7.

0x5368

tag : 0x5

Hit/Miss TLB : Miss

page fault : No

Valid	Tag	Physical Page Number	Reference (Used)
1	0x4	6	1
1	0x1	2	0
1	0xa	3	1
1	0x5	11	1

0x02c3

tag : 0x0

H/M TLB : Miss

page fault : Yes

Valid	Tag	Physical Page Number	Reference (Used)
1	0x4	6	~~1~~ 0
1	0x0	20	~~0~~ 1
1	0xa	3	1
1	0x5	11	1

0x434b

tag : 0x4

H/M TLB : Hit

page fault : No

Valid	Tag	Physical Page Number	Reference (Used)
1	0x4	6	~~0~~ 1
1	0x0	20	1
1	0xa	3	1
1	0x5	11	1

0x6812

0x6

H/M TLB : Miss

page fault : No

Valid	Tag	Physical Page Number	Reference (Used)
1	0x4	6	0
1	0x0	20	0
1	0x6	7	1
1	0x5	11	0

0xaf50

0xa

H/M TLB : Miss

page fault : No

Valid	Tag	Physical Page Number	Reference (Used)
1	0x4	6	0
1	0x0	20	0
1	0x6	7	1
1	0xa	3	1