ASSIGNMENT 6

Xinhao Luo

Tuesday 12$^{\text{th}}$ May, 2020

1 Problem 5.2

3, 180, 43, 2, 191, 88, 190, 14, 181, 44, 186, 253

1) Tag size = $32 - log_2(16) - log_2(1) = 28$ bits

decimal address	binary address	tag	index	hit/missed
3	0000 0011	0000	0011	MISS
180	1011 0100	1011	0100	MISS
43	0010 1011	0100	1011	MISS
2	0000 0010	0000	0010	MISS
191	1011 1111	1011	1111	MISS
88	0101 1000	0101	1000	MISS
190	1011 1110	1011	1110	MISS
14	0000 1110	0000	1110	MISS
181	1011 0101	1011	0101	MISS
44	0010 1100	0010	1100	MISS
186	1011 1010	1011	1010	MISS
253	1111 1101	1111	1101	MISS

2) Tag size = $32 - log_2(8) - log_2(1) = 28$ bits

decimal address	binary address	tag	index	offset	hit/missed
3	0000 0011	0000	001	1	MISS
180	1011 0100	1011	010	0	MISS
43	0010 1011	0100	101	1	MISS
2	0000 0010	0000	001	0	HIT
191	1011 1111	1011	111	1	MISS
88	0101 1000	0101	100	0	MISS
190	1011 1110	1011	111	0	HIT
14	0000 1110	0000	111	0	MISS
181	1011 0101	1011	010	1	HIT
44	0010 1100	0010	110	0	MISS
186	1011 1010	1011	101	0	MISS
253	1111 1101	1111	110	1	MISS

3) C1) Tag size = $32 - log_2(8) = 29$ bits

decimal address	binary address	tag	index	hit/missed
3	0000 0011	00000	011	MISS
180	1011 0100	10110	100	MISS
43	0010 1011	01001	011	MISS
2	0000 0010	00000	010	MISS
191	1011 1111	10111	111	MISS
88	0101 1000	01011	000	MISS
190	1011 1110	10111	110	MISS
14	0000 1110	00001	110	MISS
181	1011 0101	10110	101	MISS
44	0010 1100	00101	100	MISS
186	1011 1010	10111	010	MISS
253	1111 1101	11111	101	MISS
MISS RATE	$\frac{12}{12} = 1$			

Cycles $= 2 \times 12 + 12 \times 25 = 324$

C2) Tag size $= 32 - log_2(4) - log_2(2) = 29$ bits

decimal address	binary address	tag	index	offset	hit/missed
3	0000 0011	00000	01	1	MISS
180	1011 0100	10110	10	0	MISS
43	0010 1011	01001	01	1	MISS
2	0000 0010	00000	01	0	MISS
191	1011 1111	10111	11	1	MISS
88	0101 1000	01011	00	0	MISS
190	1011 1110	10111	11	0	HIT
14	0000 1110	00001	11	0	MISS
181	1011 0101	10110	10	1	HIT
44	0010 1100	00101	10	0	MISS
186	1011 1010	10111	01	0	MISS
253	1111 1101	11111	10	1	MISS
MISS RATE	$\frac{10}{12} = 0.833$				

Cycles $= 12 \times 3 + 10 \times 25 = 286$

C3) Tag size $= 32 - log_2(2) - log_2(4) = 29$ bits

decimal address	binary address	tag	index	offset	hit/missed
3	0000 0011	00000	0	11	MISS
180	1011 0100	10110	1	00	MISS
43	0010 1011	01001	0	11	MISS
2	0000 0010	00000	0	10	MISS
191	1011 1111	10111	1	11	MISS
88	0101 1000	01011	0	00	MISS
190	1011 1110	10111	1	10	HIT
14	0000 1110	00001	1	10	MISS
181	1011 0101	10110	1	01	MISS
44	0010 1100	00101	1	00	MISS
186	1011 1010	10111	0	10	MISS
253	1111 1101	11111	1	01	MISS
MISS RATE	$\frac{11}{12} = 0.916$				

Cycles $= 12 \times 5 + 11 \times 25 = 335$

C4) The one with least cycles is the best, so C2 is the best.

4) 1) i) Each block has $2 \times 4 = 8$ bytes

ii) The number of block is $32 \times 1024/8 = 4096$

iii) For index, we need $log_2(4096) = 12$ bits

iv) For offset, we need 1 bits, and the rest will be for tag, which is $32 - 12 - 1 = 19$ bits.

v) For a single block, we will need 32×2 data, 19 tag size, and 1 valid bit. $64 + 19 + 1 = 84$

vi) In Total, we need $4096 \times 84 = 344064$ bits

2) i. $log_2(16) = 4$ bit offsets

ii. Set the number of index bit X, so the data size would be $16 \cdot 4 \cdot 8 \cdot 2^X = 2^{X+9}$

iii. Tag size is $32 - X - 4$ bits.

iv. For a single block, we will have $1 + 32 - X - 4$(offset bit), plus the data size

v. $2^{X+9} + (29 - X) * 2^X \geq 344064$

vi. When $X = 10$, the total size would be 543744

3) The larger cache would possible need longer access time, which decrease its performance.

5) (a) Example addresses here will be

- 00000000000000000000000000000000
- 10000000000000000000000000000000
- 00000000000000000000000000000000

(b) Example addresses here will be

- 00000000000000000000000000000000
- 00000000100000000000000000000000
- 00000000100000000000000000000000
- 00000000000000000000000000000000

All of them will MISS in both cache

(c) Adv & disAdv

Advantage	Direct mapping won't need to alter the address before any actions
Disadvantage	Generating sequence for different mapping bring extra complexity and maybe harder to achieve hits for any read operations

6) It is possible, and the tag bits will need to be entered again once the bits are XOR for the direct-mapped cache

2 Problem 5.7

1) (a) Block size = 24/2 = 12; Set size = 12/3 = 4

(b) # of index bit = 2; # of offset bit = 1; # of tag bit = 29

(c)

decimal address	binary address	tag	index	offset	hit/missed
3	0000 0011	00000	01	1	MISS
180	1011 0100	10110	10	0	MISS
43	0010 1011	01001	01	1	MISS
2	0000 0010	00000	01	0	HIT
191	1011 1111	10111	11	1	MISS
88	0101 1000	01011	00	0	MISS
190	1011 1110	10111	11	0	HIT
14	0000 1110	00001	11	0	MISS
181	1011 0101	10110	10	1	HIT
44	0010 1100	00101	10	0	MISS
186	1011 1010	10111	01	0	MISS
253	1111 1101	11111	10	1	MISS

(d)

	Set 0	Set 1	Set 2	Set 3
Content 0	men[88]	mem[3]	mem[180]	mem[191]
Content 1		mem[43]	mem[44]	mem[14]
Content 2		mem[186]	mem[253]	

2) (a) No index bit, No offset bit

(b) # of set = 8 / 1 = 8

(c)

decimal address	binary address	tag	hit/missed
3	0000 0011	00000011	MISS
180	1011 0100	10110100	MISS
43	0010 1011	01001011	MISS
2	0000 0010	00000010	MISS
191	1011 1111	10111111	MISS
88	0101 1000	01011000	MISS
190	1011 1110	10111110	MISS
14	0000 1110	00001110	MISS
181	1011 0101	10110101	MISS
44	0010 1100	00101100	MISS
186	1011 1010	10111010	MISS
253	1111 1101	11111101	MISS

(d)

	Set 0
Content 0	mem[181]
Content 1	mem[44]
Content 2	mem[186]
Content 3	mem[253]
Content 4	mem[191]
Content 5	mem[88]
Content 6	mem[190]
Content 7	mem[14]

3) # of sets = 8 / 2 = 4 ; # of offset bit = 1

(a)

decimal address	binary address	tag	offset	LRU	MRU
3	0000 0011	0000001	1	MISS	MISS
180	1011 0100	1011010	0	MISS	MISS
43	0010 1011	0010101	1	MISS	MISS
2	0000 0010	0000001	0	HIT	HIT
191	1011 1111	1011111	1	MISS	MISS
88	0101 1000	0101100	0	MISS	MISS
190	1011 1110	1011111	0	HIT	MISS
14	0000 1110	0000111	0	MISS	MISS
181	1011 0101	1011010	1	MISS	HIT
44	0010 1100	0010110	0	MISS	MISS
186	1011 1010	1011101	0	MISS	MISS
253	1111 1101	1111110	1	MISS	MISS
MISS RATE				$\frac{10}{12} = \frac{5}{6}$	$\frac{10}{12} = \frac{5}{6}$

(b) The best situation is

- 2 and 3
- 180 and 181
- 190 and 191

so we have MISS RATE = $\frac{9}{12} = \frac{3}{4}$

4) (a) •

$$\begin{aligned} Actual\,CPI &= Base\,CPI + miss\,penalty \cdot miss\,rate \\ &= 1.5 + \frac{Main\,Memory\,Access\,Time}{\frac{1}{Processor\,Speed}} \cdot 7\% \\ &= 1.5 + \frac{100ns}{\frac{1}{2Ghz}} \cdot 7\% \\ &= 15.5 \end{aligned} \tag{1}$$

Double

$$\begin{aligned} Actual\,CPI &= Base\,CPI + miss\,penalty \cdot miss\,rate \\ &= 1.5 + \frac{Main\,Memory\,Access\,Time}{\frac{1}{Processor\,Speed}} \cdot 7\% \\ &= 1.5 + \frac{100ns \cdot 2}{\frac{1}{2Ghz}} \cdot 7\% \\ &= 29.5 \end{aligned} \tag{2}$$

Half

$$\begin{aligned} Actual\,CPI &= Base\,CPI + miss\,penalty \cdot miss\,rate \\ &= 1.5 + \frac{Main\,Memory\,Access\,Time}{\frac{1}{Processor\,Speed}} \cdot 7\% \\ &= 1.5 + \frac{100ns \cdot 0.5}{\frac{1}{2Ghz}} \cdot 7\% \\ &= 8.5 \end{aligned} \tag{3}$$

(b) •

$$\begin{aligned} Actual\,CPI &= Base\,CPI + miss\,penalty \cdot 2^{nd} miss\,rate(direct) + 2^{nd} speed(direct) \cdot 1^{st} miss\,rate \\ &= 1.5 + \frac{100ns}{\frac{1}{2Ghz}} \cdot 3.5\% + 12 \cdot 7\% \\ &= 9.34 \end{aligned} \tag{4}$$

Double

$$\begin{aligned} Actual\,CPI &= Base\,CPI + miss\,penalty \cdot 2^{nd} miss\,rate(direct) + 2^{nd} speed(direct) \cdot 1^{st} miss\,rate \\ &= 1.5 + \frac{100ns \cdot 2}{\frac{1}{2Ghz}} \cdot 3.5\% + 12 \cdot 7\% \\ &= 16.34 \end{aligned} \tag{5}$$

Half

$$\begin{aligned} Actual\,CPI &= Base\,CPI + miss\,penalty \cdot 2^{nd} miss\,rate(direct) + 2^{nd} speed(direct) \cdot 1^{st} miss\,rate \\ &= 1.5 + \frac{100ns \cdot 0.5}{\frac{1}{2Ghz}} \cdot 3.5\% + 12 \cdot 7\% \\ &= 5.84 \end{aligned} \tag{6}$$

(c) •

$$\begin{aligned} Actual\,CPI &= Base\,CPI + miss\,penalty \cdot 2^{nd} miss\,rate(8way) + 2^{nd} speed(8way) \cdot 1^{st} miss\,rate \\ &= 1.5 + \frac{100ns}{\frac{1}{2Ghz}} \cdot 1.5\% + 28 \cdot 7\% \\ &= 6.46 \end{aligned} \tag{7}$$

Double

$$\begin{aligned} Actual\,CPI &= Base\,CPI + miss\,penalty \cdot 2^{nd} miss\,rate(8way) + 2^{nd} speed(8way) \cdot 1^{st} miss\,rate \\ &= 1.5 + \frac{100ns \cdot 2}{\frac{1}{2Ghz}} \cdot 1.5\% + 28 \cdot 7\% \\ &= 9.46 \end{aligned} \tag{8}$$

Half

$$\begin{aligned} Actual\,CPI &= Base\,CPI + miss\,penalty \cdot 2^{nd} miss\,rate(8way) + 2^{nd} speed(8way) \cdot 1^{st} miss\,rate \\ &= 1.5 + \frac{100ns \cdot 0.5}{\frac{1}{2Ghz}} \cdot 1.5\% + 28 \cdot 7\% \\ &= 4.96 \end{aligned} \tag{9}$$

5)

$$\begin{aligned} Actual\,CPI &= Base\,CPI + 1^{st} + 2^{nd} + 3^{rd} Miss\,Rate\,Per\,Instruction \\ &= 1.5 + \frac{100ns}{\frac{1}{2Ghz}} \cdot 1.3\% + 50 \cdot 3.5\% + 12 \cdot 7\% \\ &= 6.69, \text{and it does not provide better performance.} \end{aligned} \tag{10}$$

Advs
- Reduced global miss rate
- Smaller access time compared to main memory
- More elements would be available

DisAdvs
- The reduced global miss rate could be ignored compared with the secondary cache memory
- Access is slower compared with the primary cache memory
- Brings more complicated design and increase costs

6) Let X as the number of off-chip cache we need

$$\begin{aligned} Actual\,CPI &= Base\,CPI + miss\,penalty \cdot 2^{nd} miss\,rate + 2^{nd} speed \cdot 1^{st} miss\,rate \\ &= 1.5 + \frac{100ns}{\frac{1}{2Ghz}} \cdot (4\% - 0.7\%X) + 50 \cdot 7\% \\ &= 13 - 1.4X \end{aligned} \tag{11}$$

Direct Map $13 - 1.4X = 9.34$; $X \approx 2$. The size should be $1MiB + 0.5MiB = 1.5MiB$, including the original cache

8-way $13 - 1.4X = 6.46$; $X \approx 4$. The size should be $2MiB + 0.5MiB = 2.5MiB$, including the original cache