Homework Assignment 6 Solution

Exercise 5.2

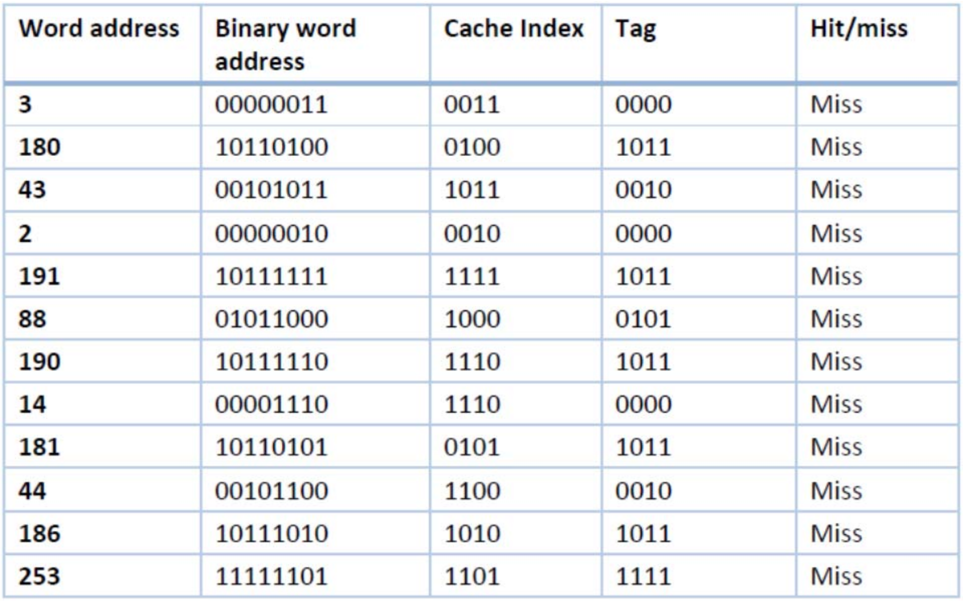
5.2.1 *(10 points)*

Word address and 1-word block size → no offset

16 blocks → 4 bits of index

All remaining higher bits are for tag.

Note: to double check that your index value is right: Index = Word address mod 16



Word address	Binary word address	Cache Index	Tag	Hit/miss
3	00000011	0011	0000	Miss
180	10110100	0100	1011	Miss
43	00101011	1011	0010	Miss
2	00000010	0010	0000	Miss
191	10111111	1111	1011	Miss
88	01011000	1000	0101	Miss
190	10111110	1110	1011	Miss
14	00001110	1110	0000	Miss
181	10110101	0101	1011	Miss
44	00101100	1100	0010	Miss
186	10111010	1010	1011	Miss
253	11111101	1101	1111	Miss

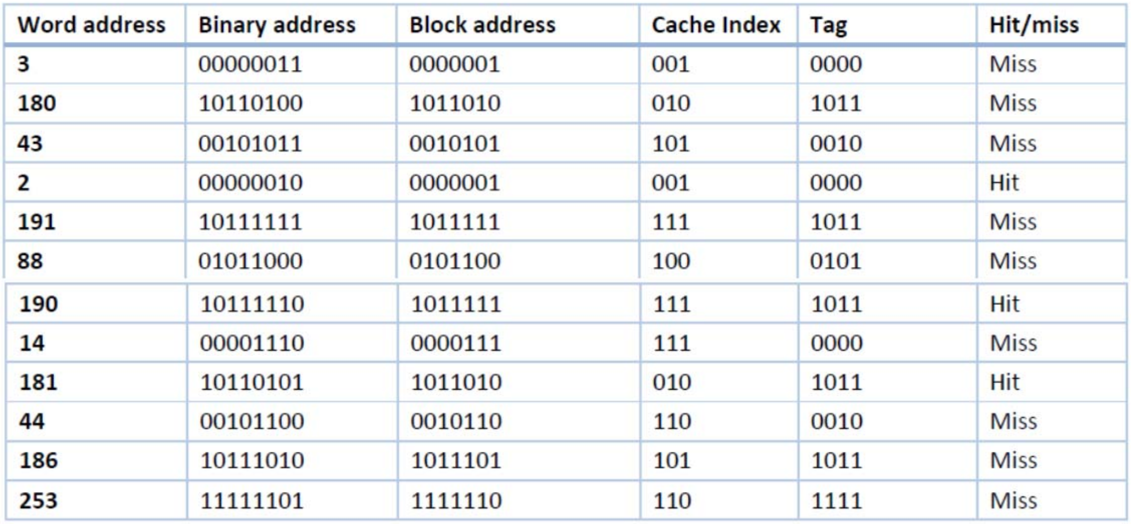
5.2.2 *(10 points)*

Word address and 2 word block → 1 bit offset

8 blocks → 3 bits of index

All remaining higher bits are for tag.

Note: to double check that your index value is right: Index = Word address/2 mod 8



Word address	Binary address	Block address	Cache Index	Tag	Hit/miss
3	00000011	0000001	001	0000	Miss
180	10110100	1011010	010	1011	Miss
43	00101011	0010101	101	0010	Miss
2	00000010	0000001	001	0000	Hit
191	10111111	1011111	111	1011	Miss
88	01011000	0101100	100	0101	Miss
190	10111110	1011111	111	1011	Hit
14	00001110	0000111	111	0000	Miss
181	10110101	1011010	010	1011	Hit
44	00101100	0010110	110	0010	Miss
186	10111010	1011101	101	1011	Miss
253	11111101	1111110	110	1111	Miss

Exercise 5.6

5.6.1 ***(6 points)***

Clock rate must be equal to the inverse of Hit time.
P1: 1.52 GHz
P2: 1.11 GHz

5.6.2 ***(6 points)***

AMAT = Hit time + Miss Rate × Miss Penalty

P1 AMAT = 0.66 + 8% × 70 = 6.26 ns = 9.48 cycles
P2 AMAT = 0.9 + 6% × 70 = 5.1 ns = 5.67 cycles

5.6.3 ***(6 points)***

Add extra cycle due to L1 cache misses. Every instruction needs to be fetched and 36% of instructions access memory
P1 total CPI = 1.0 + 8%×70/0.66 + 36%×8%×70/0.66 = 1.0 + (1+36%)×8%×70/0.66 = 12.54
P2 total CPI = 1.0 + 6%×70/0.9 + 36%×6%×70/0.9 = 7.35

P1 execution time = I × 12.54× 0.66 = 8.28 ×I ns,
P2 execution time = I × 7.35 × 0.9 = 6.62 ×I ns, where I is the number of instructions.

→P2 is faster

5.6.4 ***(7 points)***

The miss rate of L1 cache is 8%, which goes to the L2 cache. So the overall miss rate of L2 cache to the main memory is 8% × 95%
P1 AMAT = 0.66 + 8% × 5.62 + 8%×95% × 70 = 0.66 + 0.45 + 5.32 = 6.43 ns = 9.74 cycles
It is worse with L2 cache.

5.6.5 ***(5 points)***

With L2 cache, P1 total CPI = 1.0 + (1+36%)×8%×(5.62 + 95%×70)/0.66= 12.89

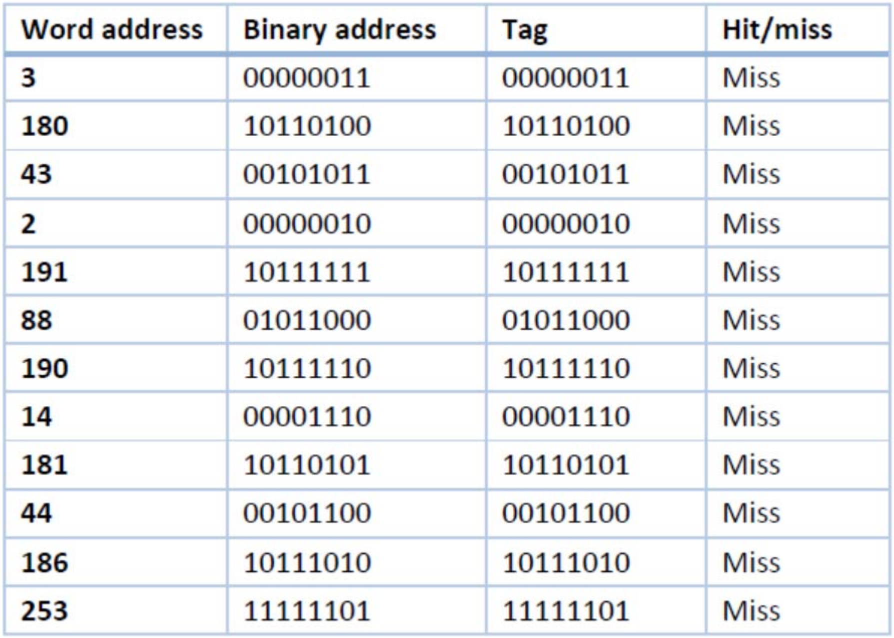
5.6.6 ***(10 points)***

Still P2 is faster because the run time of P1 with L2 cache is I × 12.89 × 0.66 = 8.51 × I ns
To match P2's performance of 6.62×I ns, we need P1's total CPI to target 6.62 / 0.66 = 10.03
Assuming the target L1 cache miss rate as x, then
1.0 + (1+36%) × x × (5.62 + 95%×70)/0.66 = 10.03
Solve it to get targeted miss rate x = 6.1%

Exercise 5.7

5.7.2 ***(10 points)***

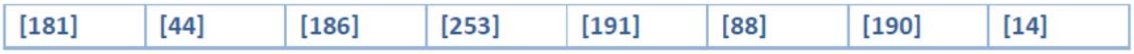
For a fully associative cache of 1-word block and a total size of 8 words, block address is same as word address
Tag: same as binary address
Index: None (only one set)



Word address	Binary address	Tag	Hit/miss
3	00000011	00000011	Miss
180	10110100	10110100	Miss
43	00101011	00101011	Miss
2	00000010	00000010	Miss
191	10111111	10111111	Miss
88	01011000	01011000	Miss
190	10111110	10111110	Miss
14	00001110	00001110	Miss
181	10110101	10110101	Miss
44	00101100	00101100	Miss
186	10111010	10111010	Miss
253	11111101	11111101	Miss

The only possible way for there to be a hit is a repeated reference to the same word, which doesn't occur for this sequence.

The final cache contents:



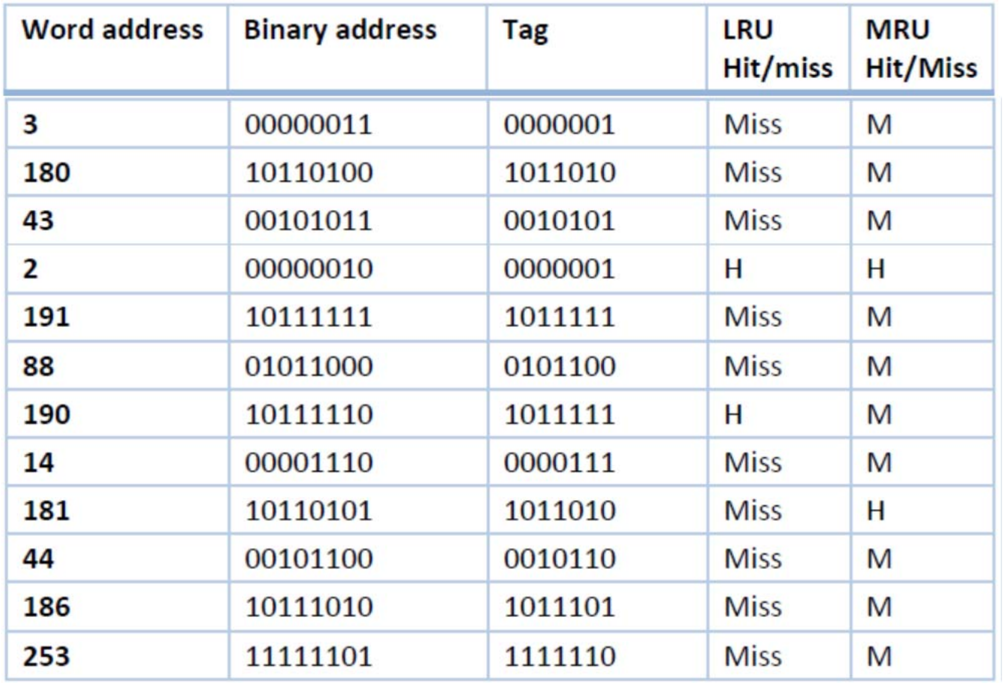
[181]	[44]	[186]	[253]	[191]	[88]	[190]	[14]

5.7.3 ***(16 points)***

For a fully associative cache of 2-word block and a total size of 8 words, block address = word address / 2

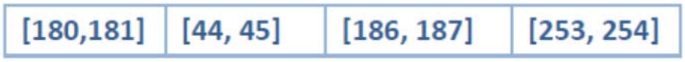
Tag: same as block address

Index: None (only one set)



Word address	Binary address	Tag	LRU Hit/miss	MRU Hit/Miss
3	00000011	0000001	Miss	M
180	10110100	1011010	Miss	M
43	00101011	0010101	Miss	M
2	00000010	0000001	H	H
191	10111111	1011111	Miss	M
88	01011000	0101100	Miss	M
190	10111110	1011111	H	M
14	00001110	0000111	Miss	M
181	10110101	1011010	Miss	H
44	00101100	0010110	Miss	M
186	10111010	1011101	Miss	M
253	11111101	1111110	Miss	M

The final cache contents:



[180,181]	[44, 45]	[186, 187]	[253, 254]

Miss rate: 10/12

5.7.4 *(18 points)*
CPI = base CPI + Mem stall CPI
Main memory access time 100 ns = 200 cycles

L1 only:
MEM stall cycles per instruction (CPI): 0.07 × 200 = 14 cycles
CPI = 1.5 + 14 = 15.5

Direct mapped L2:
Mem stall CPI = .07 × 12 + 0.035 × 200 = 7.84 cycles
CPI = 1.5 + 7.84 = 9.34

8-way set associated L2:
Mem stall per CPI = .07 × 28 + 0.015 × 200 = 4.96 cycles
CPI = 1.5 + 4.96 = 6.46

Doubled memory access time, Main memory access time 200 ns = 400 cycles
L1 only: MEM stall CPI = 0.07 × 400 = 28
CPI = 1.5 + 28 = 29.5

Doubled memory access time, direct mapped L2:
Mem stall CPI = .07 × 12 + 0.035 × 400 = 14.84
CPI = 1.5 + 14.84 = 16.34

Doubled memory access time, 8-way set associated L2:
Mem stall CPI = .07 × 28 + 0.015 × 400 = 7.96
CPI = 1.5 + 7.96 = 9.46

Halved memory access time, Main memory access time 50 ns = 100 cycles
L1 only:
MEM stall per instruction = .07× 100 = 7
CPI = 1.5 + 7 = 8.5

Halved memory access time, direct mapped L2:
MEM stall per instruction = .07 × 12 + 0.035 × 100 = 4.34
CPI = 1.5 + 4.34 = 5.84

Halved memory access time, 8-way set associated L2:
MEM stall per instruction = .07 × 28 + 0.015 × 10 = 3.46
CPI = 1.5 + 3.46 = 4.96

5.7.5 *(6 points)*
With a L2 direct-mapped cache and a L3 cache of 50 cycle access time,
MEM stall CPI = 0.07 × 12 + 0.035 × 50 + 0.013 × 200 = 0.84 + 1.75 + 2.6 = 5.19
CPI = 1.5 + 5.19 = 6.69

It’s better than the L2 direct-mapped cache only CPI of 9.34

Adding the L3 cache does reduce the overall memory access time, which is the main advantage of having a L3 cache. The disadvantage is that the L3 cache takes real estate away from CPU to having other types of functional units.