1. Textbook Problem 5.2 (all sub-parts)

5.2.1

Decimal address	Binary address	Tag	Index	Hit / Miss
3	0000 0011	0000	0011	Miss
180	1011 0100	1011	0100	Miss
43	0010 1011	0010	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

5.2.2

Decimal address	Binary address	Tag	Index	Offset	Hit / Miss
3	0000 0011	0000	001	1	Miss
180	1011 0100	1011	010	0	Miss
43	0010 1011	0010	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

5.2.3

C1:

Number	Binary	Tag	Index	Hit / Miss
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	100%			

Cycles = 2*12 + 12*25 = 324

C2:

Number	Binary	Tag	Index	offset	Hit / Miss
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	83.3%				

Cycles = 12*3 + 10*25 = 286

C3:

Number	Binary	Tag	Index	offset	Hit / Miss
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	91.6%				

Cycles = 12*5 + 11*25 = 335

C2 is the best.

5.2.4

FIRST CACHE:

Total number of blocks

= Total bytes * (1 block / 2 words) * (1 word / 4 bytes)

= 32 * 1024 * (1 block / 8 bytes)

= 2^{12} blocks

Total number of bits

= 2^{12} * (Block size + tag size + valid bit size)

= 2^{12} * (64 + 19 + 1)

= 344064 bits

SECOND CACHE:

8 KB = 8192 bytes = 2^{13} blocks

Total number of bits

= 2^{13} * (Block size + tag size + valid bit size)

= 2^{13} * (16*32+(32-13-4)+1)

= 2^{13}*528

= 4325376 bits

The second cache contains larger number of bits, we need to access the base address to go through all data.

5.2.5

0x00000

0x00400

0x00800

0x00000

Advantage: Don’t need to alter the address

Disadvantage: It will be difficult to create sequence for different mapping and hard to get a hit from reading.

5.2.6

It is possible. The XOR need to be changed since it may miss some bits from the provided address.

2. Textbook Problem 5.7 (all sub-parts)

5.7.1

2 word blocks:

24/2 = 12 blocks; 12/3 = 4 sets; 2 index bits; 1 offset bit; 29 tag bits

Decimal address	Binary address	Tag	Index	Offset	Hit / Miss
3	0000 0011	0000	001	1	Miss
180	1011 0100	1011	010	0	Miss
43	0010 1011	0010	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-way associative:

	offset 0	offset 1	offset 2	offset 3
0	mem[88]	mem[3]	mem[180]	mem[191]
1		mem[43]	mem[44]	mem[14]
2		mem[186]	mem[253]	

5.7.2

0 index bit, 0 offset bit; 8 sets

Decimal address	Binary address	Tag	Hit / Miss
3	0000 0011	0000 0011	Miss
180	1011 0100	1011 0100	Miss
43	0010 1011	0010 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

Fully associative:

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

5.7.3

4 sets; 1 offset bit

number	binary	tag	offset	LRU	MRU
3	0000 0011	0000 001	1	Miss	Miss
180	1011 0100	1011 010	0	Miss	Miss
43	0010 1011	0010 101	1	Miss	Miss
2	0000 0010	0000 001	0	Hit	Hit
191	1011 1111	1011 111	1	Miss	Miss
88	0101 1000	0101 100	0	Miss	Miss
190	1011 1110	1011 111	0	Hit	Miss
14	0000 1110	0000 111	0	Miss	Miss
181	1011 0101	1011 010	1	Miss	Hit
44	0010 1100	0010 110	0	Miss	Miss
186	1011 1010	1011 101	0	Miss	Miss
253	1111 1101	1111 1110	1	Miss	Miss
Miss Rate				5/6	5/6

- 2 and 3
- 180 and 181
- 190 and 191
- Miss Rate = 9/12 = 3/4

5.7.4

(1) CPI = Base CPI + miss penalty*miss rate
= 1.5 + memory access /((1/speed)*7%)
= 1.5 + 100ns/((1/2GHz)*7%)
= 15.5

Doubled time CPI = Base CPI + miss penalty*miss rate
= 1.5 + memory access /((1/speed)*7%)
= 1.5 + 200ns/((1/2GHz)*7%)
= 29.5

Halfed time CPI = Base CPI + miss penalty*miss rate
= 1.5 + memory access /((1/speed)*7%)
= 1.5 + 50ns/((1/2GHz)*7%)
= 8.5

(2) CPI = Base CPI + miss penalty*2^{nd}miss + 2^{nd}speed*1^{st}miss rate
= 1.5 + 100ns/(1/2GHz) * 3.5% + 12.7%
= 16.34

Doubled CPI = Base CPI + miss penalty*2^{nd}miss + 2^{nd}speed*1^{st}miss rate
= 1.5 + 200ns/(1/2GHz) * 3.5% + 12.7%
= 16.34

Halfed CPI = Base CPI + miss penalty*2^{nd}miss + 2^{nd}speed*1^{st}miss rate
= 1.5 + 50ns/(1/2GHz) * 3.5% + 12.7%
= 5.84

(3) Actual CPI
= Base CPI + miss penalty*second miss rate + second speed*first miss rate
= 1.5 + 100ns/(1/2GHz) * 1.5% + 28.7%
= 6.46

Doubled Actual CPI

= Base CPI + miss penalty*second miss rate + second speed*first miss rate

= 1.5 + 200ns/(1/2GHz) * 1.5% + 28.7%

= 9.46

Halfed Actual CPI

= Base CPI + miss penalty*second miss rate + second speed*first miss rate

= 1.5 + 50ns/(1/2GHz) * 1.5% + 28.7%

= 4.96

5.7.5

CPI = Base CPI + first + second + third

= 1.5 + 100ns/(1/2GHz) * 1.3% + 50*3.5% + 12*7%

= 6.69

Advantages:

- minimal access time
- lower miss rate
- most elements are available in these three levels of cache

Disadvantages:

- Slow access to the third level
- Doesn’t reduce miss rate
- Complex construction

5.7.6

Actual CPI = Base CPI + miss penalty*second miss rate + second speed*first miss rat

= 1.5 * 100ns/(1/2GHz) * (4%-0.7%)n + 50 * 7%

= 13 - 1.4n

Direct Map: 13 - 1.4n = 9.34, n = 2. The size is 1MB

8-way associative: 13 - 1.4n = 6.46, n = 4. The size is 2MB