3. (15 points) Associative Caches

Here is a series of address references given as 4-bit word addresses in both decimal and binary; we also list the relative time at which these references occur:

Addr	8	9	8	5	14	10	1	9	6	12	5	1	8
Binary	1000	1001	1000	0101	1110	1010	0001	1001	0110	1100	0101	0001	1000
Time	1	2	3	4	5	6	7	8	9	10	11	12	13

Below are four different 8-word caches (similar to Figure 5.14 of the text). For each cache type, assuming the cache is initially empty, show the final contents of the cache, and in the table at the bottom, show how many cache hits and misses there are for each type of cache. Write your solution in the tables below, assuming the above word address are 4-bit binary numbers. You should write the binary form of the tag in the tables below, except for the fully associative cache, where you may write the decimal form of the tag. In the data column, write M[3] for data at memory address 3, M[8] for data at memory address 8.

Assume a LRU replacement scheme. When inserting an element into the cache, if there are multiple empty slots for that index, you should put the new element in the left-most empty slot.

Direct mapped

Set	Tag	Data
0	1	M[8]
1	0	M[1]
2	1	M[10]
3		
4	1	M[12]
5	0	M[5]
6	0	M[6]
7		

Two-way set associative

Set	Tag	Data	Tag	Data
0	10	M[8]	11	M[12]
1	01	M[5]	00	M[1]
2	01	M[6]	10	M[10]
3				

Four-way set associative

Set	Tag	Data	Tag	Data	Tag	Data	Tag	Data
0	110	M[12]	100	M[8]	101	M[10]	011	M[6]
1	100	M[9]	010	M[5]	000	M[1]		

Fully associative

Tag	Data	Tag	Data	Tag	Data	Tag	Data	Tag	Data	Tag	Data	Tag	Data	Tag	Data
1000	M[8]	1001	M[9]	0101	M[5]	1110	M[14]	1010	M[10]	0001	M[1]	0110	M[6]	1100	M[12]

Question continued on next page.

Write the number of cache hits and misses for each scheme in the table below:

	Hits	Misses
Direct Mapped	3	10
2-way Set Associative	2	11
4-way Set Associative	4	9
Fully Associative	5	8

4. (4 points) Associative Caches

   Suppose we have a 512 KB, 4-way set associative cache with a 32-byte line size. For 32-bit addresses,

   (a) (1 point) Which bits would be used to specify the byte offset? Give your answer in either the form b_8–b_4 or 8:4.

   (b) (1 point) Which bits would be used to specify the block offset?

   (c) (1 point) Which bits would be used to specify the index?

   (d) (1 point) Which bits would be used to specify the tag?