CSCI 2330 – Direct-Mapped Caching Exercises

Consider a system with an **8-bit word size** that contains a direct-mapped cache comprised of **8 cache lines** with a **block size of 4 bytes** (i.e., a total cache size of 32 bytes). Assume write-back, write-allocate cache behavior.

1. Assuming that the system has the maximum possible amount of memory (given the word size), how many **blocks** of memory exist?

2. How many different memory blocks are mapped to each cache line?

3. When using a memory address to locate data in the cache, how many **index bits**, **offset bits**, and **tag bits** will there be?

4. Suppose that the cache has the contents pictured below.

Consider each of the memory operations listed below, assuming that each operation is either reading or writing a single byte at the given address (address specified in binary). **Assuming that the cache resets to the contents below in between each operation,** indicate the following for each operation:

(i) whether the operation results in a cache **hit** or cache **miss**,
(ii) how many **trips to memory** are required by the operation, and
(iii) the **affected cache line with any updates** resulting from the operation.

For read operations, the "result" specified is the byte value ultimately retrieved by the read (either from the cache or from memory). For write operations, the "result" is the updated value of the affected block. These results would not be known prior to actually executing the operation, but may affect how the relevant cache line should be updated.

(a) **Read 01000100** (result: 5)

(b) **Read 11100000** (result: 17)

(c) **Write 01110000** (result: 7)

(d) **Read 10101000** (result: 12)

(e) **Read 01101100** (result: 2)

(f) **Write 11111100** (result: 3)
(assume write-misses fetch the block from memory but only modify the cached copy)

Line	V	D	Tag	Data (4 Bytes)
0	1	0	111	17
1	1	0	011	9
2	0	0	101	15
3	1	1	001	8
4	1	0	011	4
5	0	0	111	6
6	0	0	101	32
7	1	0	110	3