Autumn 2022, Homework 8 Key

Q1. (12 pts) Cache and Memory mapping
Suppose a computer system's main memory has 2M-byte capacity and the CPU cache consists of 32 blocks, where each block contains 64 bytes.
Q1-1. (4 pts) Direct Mapping
Q1-1-i. Divide the bits in main memory into tag, block and offset bits. Show your progress.

64 bytes / block = 2^6, Offset = 6 bits
32 blocks = 2^5, Blocks = 5 bits
2M bytes = 2^{21} in total. Therefore, tag = 21 - 5 - 6 = 10 bits

Tag	Block	Offset
10 bits	5 bits	6 bits

Grading guide: any wrong value, -0.5. No show of progress, -1.

Q1-1-ii. What is the tag, line and offset for the address $123A63, in hexadecimal? Show your progress.

$123A63 = 1 0010 0011 1010 0110 0011 = (1001000111) (01001) (100011)
= (10 0100 0111)(0 1001) (10 0011) = $247 $9 $23

Tag	Block	Offset
0x 247	0x 9	0x 23

Grading guide: any wrong value, -0.5. No show of progress, -1.

Q1-2. (4 pts) Fully associative mapping
Q1-2-i. Divide the bits in main memory into tag and offset bits. Show your progress.

64 bytes / block = 2^6, Offset = 6 bits
2M bytes = 2^{21} in total. Therefore, tag = 21 - 6 = 15 bits

Tag	Offset
15 bits	6 bits

Grading guide: any wrong value, -0.5. No show of progress, -1.

Q1-2-ii. What is the tag and offset for the address $123A63, in hexadecimal? Show your progress.

$123A63 = 1 0010 0011 1010 0110 0011 = (100100011101001) (100011)
= (100 1000 1110 1001)(10 0011) = $48F9 $23

Tag	Offset
0x 48F9	0x 23

Grading guide: any wrong value, -0.5. No show of progress, -1.

Q1-3. (4 pts) 4-way set associative mapping
Q1-3-i. Divide the bits in main memory into tag, set and offset bits. Show your progress.

64 bytes / block = 2^6, Offset = 6 bits
4-way associativity: 32 blocks / 4 = 8 sets. set bits = log 8 = 3 bits
2M bytes = 2^{21} in total. Therefore, tag = 21 - 6 - 3 = 12 bits

Tag	Set	Offset
12 bits	3 bits	6 bits

Grading guide: any wrong value, -0.5. No show of progress, -1.

Q1-3-ii. What is the tag, set and offset for the address $123A63, in hexadecimal? Show your progress.

$123A63 = 1 0010 0011 1010 0110 0011 = (100100011101) (001) (100011)
= (1001 0001 1101) (001) (10 0011) = $2474 $3 $23

Tag	Set	Offset
0x 91D	0x 1	0x 23

Grading guide: any wrong value, -0.5. No show of progress, -1.

Q2. (8 pts) Cache hit and miss

Suppose a computer system's main memory has 256-byte capacity. The computer has a 16-byte direct-mapped cache with 4 bytes per block. The computer accesses a number of memory locations throughout the course of running a program. Here is the memory addresses in this exact order: **0x91, 0xA8, 0xA9, 0xAB, 0xAD, 0x93, 0x6E, 0xB9, 0x17, 0xE2, 0x4E, 0x4F, 0x50, and 0xA4.** The cache is already filled out as shown below. (The contents of the tag are shown in binary and the cache "contents" are simply the *address* of the main memory byte that was brought into that cache offset.)

Tag (binary)	Block #	offset 0	offset 1	offset 2	offset 3
1110	0	E0	E1	E2	E3
0001	1	14	15	16	17
1011	2	B8	B9	BA	BB
0110	3	6C	6D	6E	6F

Q2-1. What is the hit ratio for the entire memory reference sequence given above (**in bold**)?

For your reference, below is the main memory map:
Tag: 4bits, Block: 2 bits, Offset: 2 bits

Tag (binary)	Block 0	Block 1	Block 2	Block 3
0000	00, 01, 02, 03	04, 05, 06, 07	08, 09, 0A, 0B	0C, 0D, 0E, 0F
0001	10, 11, 12, 13	14, 15, 16, 17	18, 19, 1A, 1B	1C, 1D, 1E, 1F
0010	20, 21, 22, 23	24, 25, 26, 27	28, 29, 2A, 3B	2C, 2D, 2E, 2F
0011	30, 31, 32, 33	34, 35, 36, 37	38, 39, 3A, 3B	3C, 3D, 3E, 3F
0100	40, 41, 42, 43	44, 45, 46, 47	48, 49, 4A, 4B	4C, 4D, 4E, 4F
0101	50, 51, 52, 53	54, 55, 56, 57	58, 59, 5A, 5B	5C, 5D, 5E, 5F
0110	60, 61, 62, 63	64, 65, 66, 67	68, 69, 6A, 6B	6C, 6D, 6E, 6F
0111	70, 71, 72, 73	74, 75, 76, 77	78, 79, 7A, 7B	7C, 7D, 7E, 7F
1000	80, 81, 82, 83	84, 85, 86, 87	88, 89, 8A, 8B	8C, 8D, 8E, 8F
1001	90, 91, 92, 93	94, 95, 96, 97	98, 99, 9A, 9B	9C, 9D, 9E, 9F
1010	A0, A1, A2, A3	A4, A5, A6, A7	A8, A9, AA, AB	AC, AD, AE, AF
1011	B0, B1, B2, B3	B4, B5, B6, B7	B8, B9, BA, BB	BC, BD, BE, BF
1100	C0, C1, C2, C3	C4, C5, C6, C7	C8, C9, CA, CB	CC, CD, CE, CF
1101	D0, D1, D2, D3	D4, D5, D6, D7	D8, D9, DA, DB	DC, DD, DE, DF
1110	E0, E1, E2, E3	E4, E5, E6, E7	E8, E9, EA, EB	EC, ED, EE, EF
1111	F0, F1, F2, F3	F4, F5, F6, F7	F8, F9, FA, FB	FC, FD, FE, FF

Access sequence:

Accesses	Tag	Block	Hit/Miss	B0 Tag	B1 Tag	B2 Tag	B3 Tag
91	1001	0	Miss	1110 -> 1001	0001	1011	0110
A8	1010	2	Miss	1001	0001	1011 -> 1010	0110
A9	1010	2	Hit	1001	0001	**1010**	0110
AB	1010	2	Hit	1001	0001	**1010**	0110
AD	1010	3	Miss	1001	0001	1010	0110 -> 1010
93	1001	0	Hit	**1001**	0001	1010	1010
6E	0110	3	Miss	1001	0001	1010	1010 -> 0110
B9	1011	2	Miss	1001	0001	1010 -> 1011	0110
17	0001	1	Hit	1001	**0001**	1011	0110
E2	1110	0	Miss	1001 -> 1110	0001	1011	0110
4E	0100	3	Miss	1110	0001	1011	0110 -> 0100
4F	0100	3	Hit	1110	0001	1011	**0100**
50	0101	0	Miss	1110 -> 0101	0001	1011	0100
A4	1010	1	Miss	0101	0001 -> 1010	1011	0100

From the above sequence table, 5 hits out of 14 accesses = 5/14 = 35.7%

Grading guide: if the result is wrong, -2. If there is no showing of progress, -2.

Q4-2. What memory blocks will be in the cache after the last address has been assessed?

Tag (binary)	Block #	offset 0	offset 1	offset 2	offset 3
0101	0	50	51	52	53
1010	1	A4	A5	A6	A7
1011	2	B8	B9	BA	BB
0100	3	4C	4D	4E	4F

Grading guide: any wrong value, -0.5.