Canvas Group Name: Student 1: Student 2:
Student 3:

CMPE110 HA4: Cache Detective

Due Date: Tuesday 02/26/19

1) Consider a 16-bit processor with a directly mapped cache with 16 Byte cache blocks and 4 entries. Fill out the bitfields in the address map below showing how the address bits are used to lookup a value in the cache. **(1 Point)**

15:6 TAG	5:4 Index	3:0 offset

15 0

2) Assume the following address sequence is used to access the cache: 0xA30, 0xA32, 0xA3F, 0xA40, 0xA70, 0xA4D, 0xA30, 0xA0F. Determine whether the accesses hit or miss the cache. Use the cache block size and number of entries as in 1). Fill out the table below. Assume that all cache lines are invalid before the first access. **(8 Points)**

Byte address (hex)	Block address (binary)	Tag (binary)	Cache Index	Hit or Miss?
0xA30	1010 0011 [0000]	1010 00	11	m
0xA32	1010 0011 [0000]	1010 00	11	h
0xA3F	1010 0011 [0000]	1010 00	11	h
0xA40	1010 0100 [0000]	1010 01	00	m
0xA70	1010 0111 [0000]	1010 01	11	m
0xA4D	1010 0100 [0000]	1010 01	00	h
0xA30	1010 0011 [0000]	1010 00	11	m
0xA0F	1010 0000 [0000]	1010 00	00	m

Either the index of the block (TAG+IDX) or the byte start address of the block are acceptable.

3) Consider the access sequence from 2) and draw the state of the cache at the end of that sequence, filling out the table below. In the data column state which data is found at that entry (e.g. Mem[0xA30]). **(4 Points)**

Index	valid	Tag	Data
00	1	101000	Mem[101000 00 0000]
01	0	-	-
10	0	-	-
11	1	101000	Mem[101000 11 0000]

4) The table below shows five different cache organizations utilizing different address width, block size, number of cache entries, the associativity of the cache and number of tag bits. Fill in the five free cells in the table. **(5 Points)**

Address width	Block Size (bytes)	# Entries	Associativity	Tag size (bits)
32	8	32	1	24
16	4	4	4	14
24	16	16	8	19
20	64	128	1	7
64	128	2048	4	48

5) a) Consider a processor with two levels of cache with the following properties:
L1 Hit rate: 80%, L1 Access (hit): 5 cycles, L2 Hit Rate: 70%, L2 Access (hit): 11 cycles, DRAM Hit Rate: 100%, DRAM Access (hit): 100 cycles. Compute the AMAT. The access times are non-inclusive: a miss in L1, then hit in L2 takes 11+5=16 cycles. **(2 Points)**
AMAT = (hit time) + miss rate * miss penalty = hit rate * hit time + miss rate * miss time
= 0.8*5 + 0.2(0.7*16 + 0.3(116)) = 13.2 cycles
Or = 5 + 0.2(11 + 0.3(100)) = 13.2 cycles
-> If students access times are inclusive, then answer is 11.54 cycles.
b) Consider a processor that has a CPI of 1 for all instructions besides loads. Loads are 30% of the instructions and have the AMAT calculated above. What is the average CPI of the processor? **(1 Point)**

Avg CPI = 0.7*1 + 0.3*13.2 = 4.66
= 1 + 0.3 * 12.2 = 4.66