In a hypothetical computer system, the main memory is augmented with a 2-way set-associative cache to boost performance. The memory is byte-addressable. The address consists of 12 bits divided into three fields: 7-bit tag, 3-bit set, and 2-bit word. The cache implements a first-in-first-out (FIFO) replacement algorithm and follows a write-back policy.

1. What is the size of the main memory?

   MM size = 2^{12} words = 4 KB

2. How long is the cache line?

   Line size = 2^{2} words = 4 B

3. How much data can be stored in the cache?

   Cache size = (2^{3} sets) * (2 lines per set) * (4 words per line) = 64 B

4. The table below captures a sequence of memory reads/writes made during the execution of a specific program. The first two rows show, for each memory operation, the address referenced during that operation and the type of the operation. Fill up the rest of table assuming that the cache is initially empty.

Address (decimal)	118	121	116	53	117	278	119	276	116	123
Read/**W**rite (R/W)	W	W	R	R	R	R	R	R	W	R
Set (decimal)	5	6	5	5	5	5	5	5	5	6
Tag (decimal)	3	3	3	1	3	8	3	8	3	3
Hit/**M**iss (H/M)	M	M	H	M	H	M	M	H	H	H
Write-back? (Y/N)	N	N	N	N	N	Y	N	N	N	N

5. Suppose the average access time of the memory system (i.e., the cache combined with the main memory) measured during the execution of this program is 110 ns. Calculate the access time of the cache given that the access time of the main memory is 200 ns.

   Average access time (T_{av1}) = 110 ns
   Memory access time (T_m) = 200 ns
   Cache access time (T_{c1}) is unknown
   Hit ratio (H_1) = number of hits / number of accesses = 5 / 10 = 0.5
   $T_{av1} = T_{c1} + (1 - H_1) * T_m$
   ⇨ $110 = T_{c1} + (1 - 0.5) * 200$
   ⇨ $T_{c1} = 10$ ns

6. Would it possible to use an L2 cache, whose average hit ratio is 0.65, to reduce the average access time of the memory system by 60%. If yes, what would be the access time of that L2 cache? Otherwise, suggest what could be done to achieve that reduction. In either case, justify your answer quantitatively.

   Hit ratio of L2 cache (H_2) = 0.65
   L2 cache access time (T_{c2}) is unknown
   Average access time (T_{av2}) = 0.4 * T_{av1} = 44 ns
   $T_{av2} = T_{c1} + (1 - H_1) * (T_{c2} + (1 - H_2) * T_m)$
   ⇨ $44 = 10 + (1 - 0.5) * (T_{c2} + (1 - 0.65) * 200)$
   ⇨ $T_{c2} = -2$ ns
   ⇨ $T_{c2} < 0$
   ⇨ The 60% reduction in access time of the memory system can't be achieved using this L2 cache!

   To achieve the target reduction: use an L2 cache with a higher hit ratio.

   Notice that using a single cache with better replacement algorithm and/or higher set-associativity might help a little bit, but in this case, it wouldn't be sufficient to achieve the 60% reduction.