Solution to HW 06 – CpE 313 – Fall 2004

Question 1. What feature of a set-associative cache makes its miss rate smaller than that for a DMC?

The fact that a memory block is allowed to reside in one of many frames in a set-associative cache lowers the miss rate. In a DMC, a given memory block is allowed to reside only in one frame in the cache.

Question 2. What feature of a DMC makes its hit time smaller than that for any set-associative cache?

The fact that the tag match can be done in parallel with the word transfer to the CPU reduces the hit time.

Question 3. The index size for ANY cache equals log2(set size). Is the preceding statement true? How do you apply this statement for a DMC and for a fully associative cache?

This statement is false. The set size equals the number of frames in one set of the cache. So the set size for a DMC is 1. The statement in question would suggest that index size for the DMC would be log2(1) = 0 bits. This is plainly wrong, as we know that a DMC has the largest index of all cache configurations.

The correct statement is: “The index size for ANY cache equals log2(# of sets).” With a DMC, # of sets is the same as the number of frames in the cache. With a fully associative cache (FAC), # of sets is 1. So for a FAC, index = log2(1) = 0 bits, which agrees with our understanding of a FAC that one does not need an index to retrieve data from a FAC.

Question 4. If you look closely at slides 23 and 24 of Handout 7, you will notice that index bits are being used to pick a set of frames. Although not shown in the diagrams, these index bits are first sent to a decoder which then selects one set out of all sets in the cache. The decoder takes a certain amount of time for its operation. What does this time depend on? With this additional information in mind, give one advantage that a fully associative cache has over a DMC.

The time taken by a decoder to produce an output increases with the number of decoder inputs. Because the decoder in question is being driven by the index from the CPU address, a DMC will have the largest decoder of all cache configurations. By contrast, a FAC will not have such a decoder at all because there are no index bits! Therefore the decoding of index will take the longest time in a DMC. This is a disadvantage of a DMC.

Question 5. Assume that we want to design a cache for a processor that has 32-bit long address bus. The cache must be able to store 128 Kbytes of data. Determine the total number of bits in the cache for:
a) DMC (i.e., a 1-way associative) cache with a block size of one word.
b) a DMC (i.e., a 1-way associative) cache with a block size of 8 words.
c) a fully associative cache with a block size of one word.

Equation Number	Quantity	Formula	Part a	Part B	Part C
1	# of frames	cache size in bytes / block size in bytes	128 * 1024/4 = 32K	128K/32 = 4K	same as part a
2	# of sets	# of frames / associativity	32K/1 = 32K	4K/1 = 4K	1 (or 32K/32K = 1)
3	index size	log2(#of sets)	log2(32K)=15	log2(4K) = 12	log2(1) = 0
4	blk offset size	log2(block size)	log2(4)=2	log2(32) = 5	same as part a
5	tag size	addr size – index size – blk offset size	32 – 15 – 2 = 15	32 – 12 – 5 = 15	32 – 0 – 2 = 30
6	bits/frame	block size in bits + tag size in bits + 1 (for valid bit)	4*8 + 15+1 = 48 bits = 6 bytes	32*8 + 15 + 1 = 272 bits = 34 bytes	4*8 + 30 + 1 = 63 bits
7	total bits/cache	bits/frame * # of frames	6 B * 32K = 192 KBytes	34 B * 4K = 136 Kbytes	63 bits * 32K = 252 Kbytes

What do you conclude about the effect "on total bits for cache" when the associativity is increased?

Equation # 7tells us that the total bits for cache is a product of bits/frame and the # of frames in cache. So finding how an increase in associativity changes these two terms of the product can help us understand how associativity affects the total bits per cache.

Increase in associativity increases the size of the tag, which in turn increases the bits/frame. Since increase in associativity does not decrease the # of frames in cache, the total bits for cache end up increasing significantly with the cache associativity. Note that this calculation does not take into account the additional hardware needed for extra comparators.

What do you conclude about the effect "on total bits for cache" when the block size is increased?

One can conclude from the solutions to parts (a) and (b) that when block size is increased, the total bits per cache decrease.