Direct Map Cache and Set Associative Cache (Revision)

Lecture 14

CDA 3103

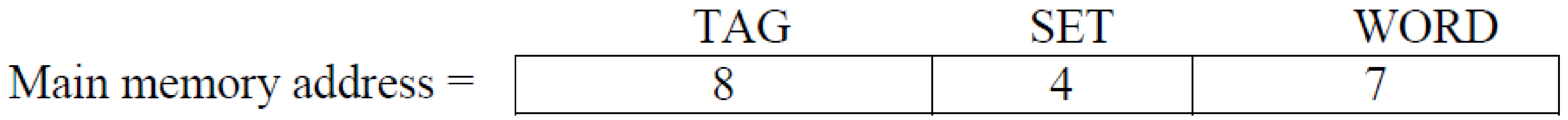
07-07-2014

Example 1

- A set-associative cache consists of 64 lines, or slots, divided into four-line sets. Main memory contains 4K blocks of 128 words each. Show the format of main memory addresses.

Solution

- The cache is divided into 16 sets of 4 lines each. Therefore, 4 bits are needed to identify the set number. Main memory consists of 4K = 2^{12} blocks. Therefore, the set plus tag lengths must be 12 bits and therefore the tag length is 8 bits. Each block contains 128 words. Therefore, 7 bits are needed to specify the word.



	TAG	SET	WORD
Main memory address =	8	4	7

Example 2

- A two-way set-associative cache has lines of 16 bytes and a total size of 8 kbytes. The 64-Mbyte main memory is byte addressable. Show the format of main memory addresses.

Solution

- There are a total of 8 kbytes/16 bytes = 512 lines in the cache. Thus the cache consists of 256 sets of 2 lines each. Therefore 8 bits are needed to identify the set number. For the 64-Mbyte main memory, a 26-bit address is needed. Main memory consists of 64-Mbyte/16 bytes = 2^{22} blocks. Therefore, the set plus tag lengths must be 22 bits, so the tag length is 14 bits and the word field length is 4 bits.

	TAG	SET	WORD
Main memory address =	14	8	4

Example 3a

- 3. Consider a machine with a byte addressable main memory of 2^{16} bytes and block size of 8 bytes. Assume that a direct mapped cache consisting of 32 lines is used with this machine.

  - a. How is a 16-bit memory address divided into tag, line number, and byte number?

Solution 3a

8 leftmost bits = tag; 5 middle bits = line number; 3 rightmost bits = byte number.

TAG	LINE	WORD
8	5	3

Example 3 (continued)

- **b. Into what line would bytes with each of the following addresses be stored?**
- 0001 0001 0001 1011
- 1100 0011 0011 0100
- 1101 0000 0001 1101
- 1010 1010 1010 1010

Solution 3b (continued)

- Line 3.
- Line 6.
- Line 3.
- Line 21.

Example 3 (continued)

- c. Suppose the byte with address 0001 1010 0001 1010 is stored in the cache. What are the addresses of the other bytes stored along with it?
- d. How many total bytes of memory can be stored in the cache?
- e. Why is the tag also stored in the cache?

Solution 3 (continued)

- **c.** Bytes with addresses 0001 1010 0001 1000 through 0001 1010 0001 1111 are stored in the cache.
- **d.** 256 bytes.
- **e.** Because two items with two different memory addresses can be stored in the same place in the cache**.** The tag is used to distinguish between them.

Question 5

- 5. Consider a memory system that uses a 32-bit address to address at the byte level, plus a cache that uses a 64-byte line size.
- a. Assume a direct mapped cache with a tag field in the address of 20 bits. Show the address format and determine the following parameters: number of addressable units, number of blocks in main memory, number of lines in cache, size of tag.
- b. Assume an associative cache. Show the address format and determine the following parameters: number of addressable units, number of blocks in main memory, number of lines in cache, size of tag.
- c. Assume a four-way set-associative cache with a tag field in the address of 9 bits. Show the address format and determine the following parameters: number of addressable units, number of blocks in main memory, number of lines in set, number of sets in cache, number of lines in cache, size of tag.

Solution 5

- a. Address format: Tag = 20 bits; Line = 6 bits; Word = 6 bits.

  Number of addressable units = 2^{32} bytes; number of blocks in main memory = 2^{26};

  Number of lines in cache = 2^{6} = 64; size of tag = 20 bits.

- b. Address format: Tag = 26 bits; Word = 6 bits.

  Number of addressable units = 2^{32} bytes; number of blocks in main memory = 2^{26};
  Number of lines in cache = undetermined; size of tag = 26 bits.

- c. Address format: Tag = 9 bits; Set = 17 bits; Word = 6 bits.

  Number of addressable units = 2^{32} bytes; Number of blocks in main memory = 2^{26};

  Number of lines in set = k = 4; Number of sets in cache = 2^{17}; Number of lines in cache =2^{19};

  Size of tag = 9 bits.

Question 6

- 6. Consider a computer with the following characteristics: total of 1Mbyte of main memory; word size of 1 byte; block size of 16 bytes; and cache size of 64 Kbytes.
- a. For the main memory addresses of F0010, 01234, and CABBE, give the corresponding tag, cache line address, and word offsets for a direct-mapped cache.
- b. Give any two main memory addresses with different tags that map to the same cache slot for a direct-mapped cache.
- c. For the main memory addresses of F0010 and CABBE, give the corresponding tag and offset values for a fully-associative cache.
- d. For the main memory addresses of F0010 and CABBE, give the corresponding tag, cache set, and offset values for a two-way set-associative cache.

Solution 6a

- Because the block size is 16 bytes and the word size is 1 byte, this means there are 16 words per block. We will need 4 bits to indicate which word we want out of a block. Each cache line/slot matches a memory block. That means each cache line contains 16 bytes. If the cache is 64Kbytes then 64Kbytes/16 = 4096 cache lines. To address these 4096 cache lines, we need 12 bits (2^{12} = 4096).

- Consequently, given a 20 bit (1 MByte) main memory address:

  Bits 0-3 indicate the word offset (4 bits).

  Bits 4-15 indicate the cache line/slot (12 bits).

  Bits 16-19 indicate the tag (remaining bits).

Solution 6a (continued)

- **F0010** = 1111 0000 0000 0001 0000

  Word offset = 0000 = 0

  Line = 0000 0000 0001 = 001

  Tag = 1111 = F
- **01234** = 0000 0001 0010 0011 0100

  Word offset = 0100 = 4

  Line = 0001 0010 0011 = 123

  Tag = 0000 = 0
- **CABBE** = 1100 1010 1011 1011 1110

  Word offset = 1110 = E

  Line = 1010 1011 1011 = ABB

  Tag = 1100 = C

Solution 6b

- b. We need to pick any address where the line is the same, but the tag (and optionally, the word offset) is different. Here are two examples where the line is 1111 1111 1111

- Address 1:
  - Word offset = 1111
  - Line = 1111 1111 1111
  - Tag = 0000
  - Address = 0FFFF

- Address 2:
  - Word offset = 0001
  - Line = 1111 1111 1111
  - Tag = 0011
  - Address = 3FFF1

Solution 6c

- With a fully associative cache, the cache is split up into a TAG and a WORDOFFSET field. We no longer need to identify which line a memory block might map to, because a block can be in any line and we will search each cache line in parallel. The word-offset must be 4 bits to address each individual word in the 16-word block. This leaves 16 bits leftover for the tag.

- F0010

  Word offset = 0h

  Tag = F001h
- CABBE

  Word offset = Eh

  Tag = CABBh

Solution 6d

- d. As computed in part a, we have 4096 cache lines. If we implement a two –way set associative cache, then it means that we put two cache lines into one set. Our cache now holds 4096/2 = 2048 sets, where each set has two lines. To address these 2048 sets we need 11 bits (2^{11} = 2048). Once we address a set, we will simultaneously search both cache lines to see if one has a tag that matches the target. Our 20-bit address is now broken up as follows:
- Bits 0-3 indicate the word offset
- Bits 4-14 indicate the cache set
- Bits 15-20 indicate the tag
- **F0010** = 1111 0000 0000 0001 0000

  Word offset = 0000 = 0

  Cache Set = 000 0000 0001 = 001

  Tag = 11110 = 1 1110 = 1E
- **CABBE** = 1100 1010 1011 1011 1110

  Word offset = 1110 = E

  Cache Set = 010 1011 1011 = 2BB

  Tag = 11001 = 1 1001 = 19