Answers to Exercises

1. Suppose a computer using direct mapped cache has 2^{20} bytes of byte-addressable main memory, and a cache of 32 blocks, where each cache block contains 16 bytes.

   a) How many blocks of main memory are there?
   b) What is the format of a memory address as seen by the cache, i.e., what are the sizes of the tag, block, and offset fields?
   c) To which cache block will the memory address 0x0DB63 map?

2. Suppose a computer using direct mapped cache has 2^{32} byte of byte-addressable main memory, and a cache of 1024 blocks, where each cache block contains 32 bytes.
   a) How many blocks of main memory are there?
   b) What is the format of a memory address as seen by the cache, i.e., what are the sizes of the tag, block, and offset fields?
   c) To which cache block will the memory address 0x000063FA map?

3. Suppose a computer using direct mapped cache has 2^{32} bytes of byte-addressable main memory and a cache size of 512 bytes, and each cache block contains 64 bytes.
   a) How many blocks of main memory are there?
   b) What is the format of a memory address as seen by cache, i.e., what are the sizes of the tag, block, and offset fields?
   c) To which cache block will the memory address 0x13A4498A map?

4. Suppose a computer using fully associative cache has 2^{16} bytes of byte-addressable main memory and a cache of 64 blocks, where each cache block contains 32 bytes.

   a) How many blocks of main memory are there?
   b) What is the format of a memory address as seen by the cache, i.e., what are the sizes of the tag and offset fields?
   c) To which cache block will the memory address 0xF8C9 map?

5. Suppose a computer using fully associative cache has 2^{24} bytes of byte-addressable main memory and a cache of 128 blocks, where each cache block contains 64 bytes.

   a) How many blocks of main memory are there?
   b) What is the format of a memory address as seen by the cache, i.e., what are the sizes of the tag and offset fields?
   c) To which cache block will the memory address 0x01D872 map?

6. Suppose a computer using fully associative cache has 2^{24} bytes of byte-addressable main memory and a cache of 128 blocks, where each block contains 64 bytes.

   a) How many blocks of main memory are there?
   b) What is the format of a memory address as seen by the cache, i.e., what are the sizes of the tag and offset fields?
   c) To which cache block will the memory address 0x01D872 map?

♦ 7. Assume a system's memory has 128M bytes. Blocks are 64 bytes in length and the cache consists of 32K blocks. Show the format for a main memory address assuming a 2-way set associative cache mapping scheme and byte addressing. Be sure to include the fields as well as their sizes.

8. A 2-way set-associative cache consists of four sets. Main memory contains 2K blocks of eight bytes each and byte addressing is used.

   a) Show the main memory address format that allows us to map addresses from main memory to cache. Be sure to include the fields as well as their sizes.

9. Suppose a byte-addressable computer using set associative cache has 2^{16} bytes of main memory and a cache of 32 blocks, and each cache block contains 8 bytes.

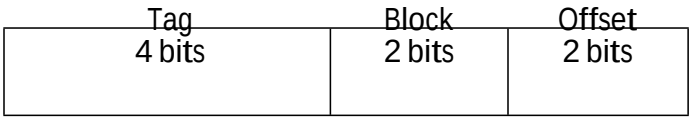
   a) If this cache is 2-way set associative, what is the format of a memory address as seen by the cache, that is, what are the sizes of the tag, set, and offset fields?
   b) If this cache is 4-way set associative, what is the format of a memory address as seen by the cache?

10. Suppose a byte-addressable computer using set associative cache has 2^{21} byes of main memory and a cache of 64 blocks, where each cache block contains 4 bytes.

    a) If this cache is 2-way set associative, what is the format of a memory address as seen by the cache, that is, what are the sizes of the tag, set, and offset fields?
    b) If this cache is 4-way set associative, what is the format of a memory address as seen by the cache?

*11. Suppose we have a computer that uses a memory address word size of 8 bits. This computer has a 16-byte cache with 4 bytes per block. The computer accesses a number of memory locations throughout the course of running a program.

    Suppose this computer uses direct-mapped cache. The format of a memory address as seen by the cache is shown below:



    The system accesses memory addresses in this exact order: 0x6E, 0xB9, 0x17, 0xE0, 0x4E, 0x4F, 0x50, 0x91, 0xA8, 0xA9, 0xAB, 0xAD, 0x93, and 0x94. The memory addresses of the first four accesses have been loaded into the cache blocks as shown below. (The contents of the tag are shown in binary and the cache "contents" are simply the address stored at that cache location.)

	Tag Contents	Cache Contents (represented by address)
Block 0	1110	0xE0
		0xE1
		0xE2
		0xE3

	Tag Contents	Cache Contents (represented by address)
Block 1	0001	0x14
		0x15
		0x16
		0x17

Block 2	1011	0xB8
		0xB9
		0xBA
		0xBB

Block 3		
	0110	0x6C
		0x6D
		0x6E
		0x6F

a) What is the hit ratio for the entire memory reference sequence given above, assuming we count the first four accesses as misses?
b) What memory blocks will be in the cache after the last address has been accessed?

13. A direct-mapped cache consists of eight blocks. A byte-addressable main memory contains 4K blocks of eight bytes each. Access time for the cache is 22 ns and the time required to fill a cache slot from main memory is 300 ns (this time will allow us to determine the block is missing and bring it into cache). Assume a request is always started in parallel to both cache and to main memory (so if it is not found in cache, we do not have to add this cache search time to the memory access). If a block is missing from cache, the entire block is brought into the cache and the access is restarted. Initially, the cache is empty.
    a) Show the main memory address format that allows us to map addresses from main memory to cache. Be sure to include the fields as well as their sizes.
    b) Compute the hit ratio for a program that loops 4 times from locations 0 to 67_{10} in memory.
    c) Compute the effective access time for this program.

14. Consider a byte-addressable computer with 24-bit addresses, a cache capable of storing a total of 64K bytes of data and blocks of 32 bytes. Show the format of a 24-bit memory address for:

    a) direct mapped b) associative
    c) 4-way set associative

---

*15. Suppose a byte-addressable computer using 4-way set associative cache has 2^{16} words of main memory (where each word is 32 bits) and a cache of 32 blocks, where each block is 4 words. Show the main memory address format for this machine. (Hint: Because this architecture is byte-addressable, and the number of addresses is critical in determining the address format, you must convert everything to bytes.)

---

18. Suppose a process page table contains the entries shown below. Using the format shown in Figure 6.17a, indicate where the process pages are located in memory.

Frame	Valid Bit
1	1
--	0
0	1
3	1
--	0
--	0
2	1
--	0

♦19. Suppose a process page table contains the entries shown below. Using the format shown in Figure 6.22a, indicate where the process pages are located in memory.

Frame	Valid Bit
--	0
3	1
--	0
--	0
2	1
0	1
--	0
1	1

---

20. Suppose you have a byte-addressable virtual address memory system with 8 virtual pages of 64 bytes each, and 4 page frames. Assuming the following page table, answer the questions below:

Page #	Frame #	Valid Bit
0	1	1
1	3	0
2	-	0
3	0	1
4	2	1
5	-	0
6	-	0
7	-	0

a) How many bits are in a virtual address?

b) How many bits are in a physical address?

c) What physical address corresponds to the following virtual addresses (if the address causes a page fault, simply indicate this is the case)?

i) 0x00
ii) 0x44
iii) 0xC2
iv) 0x80

21. Suppose we have 2^{10} bytes of virtual memory and 2^8 bytes of physical main memory. Suppose the page size is 2^4 bytes.

a) How many pages are there in virtual memory?
b) How many page frames are there in main memory?
c) How many entries are in the page table for a process that uses all of virtual memory?

Ans.

a) $2^{10}/2^4 = 2^6$, so there are 64 pages in virtual memory
b) $2^8/2^4=2^4$, so there are 16 page frames in virtual memory
c) The page table must have an entry for each virtual page, so it must have 64 entries.