

CS311 - COMPUTER SYSTEMS, I

Homework #7 - Due: Wednesday, November 28, at the start of class

Emphasis: Memory Devices and Systems; Error Detecting and Correcting Codes

1. Suppose you are building a one-level memory system from DRAM chips, having the following parameters:

Overall size: 2 GB (giga-byte) [Note $G = 2^{30}$]
Word size 64 bits
Addressability byte
Chips used: 256 Mbit, (note bit - not byte!; $M = 2^{20}$) organized 64 M x 4 bits

- What is the size (in bits) of a physical address?
- How many chips (total) are needed for this memory?
- How many chips will be accessed for any one read or write operation on this memory?
- How many address bits will each chip have?
- How many physical address bits will be decoded for chip select?
- Show - by using a diagram similar to the ones presented in class - how the bits of a physical address sent to the memory by the CPU will be interpreted by the memory as word in chip, and bank (module) number
 - If interleaving is not used
 - If interleaving is is used.

Show specific bit numbers for both cases. (Number bits 0, 1 ... starting at the right)

- Draw a block diagram - in the style of the examples done in class - for this system.
- Indicate on the diagram what group of chips will be accessed for address `0x70000008`
 - If interleaving is not used
 - If interleaving is is used.

2. Assume a two-level memory system (consisting of main memory plus a cache) has the following parameters:

Main memory: 4 GB - byte addressable
Cache: 1 MB - organized as 128 K blocks of 8 bytes (note $K = 2^{10}$)
[Note: this is the total amount of data; tags are additional]

- Show - by using a diagram similar to the ones presented in class - how the bits of a logical address will be interpreted as tag, cache block number and byte in block if the cache is direct mapped. Show specific bit numbers.
- How many bits (total) will be needed for the cache - including storage for the data items and their tags? (Note that part of the tag for a given entry is implicit in its position in the cache, so it is only need to store the part that is not implied)
- What block number in the cache will contain memory address `0x87654321`, and what will the tag be, for this case?
- Suppose that a naive cache structure were used, in which each entry in the cache stored a tag and just a single byte of data. How many bits (total) would be needed for the cache in this case, assuming that the cache can still hold 1 MB of data.
- e-g. Repeat parts a-c for a case where the cache is 4-way set associative, but has the same total size (i.e. it consists of 32K sets of 4 blocks each holding 8 bytes of data. (For the last part, give set number) Note that you are not asked to repeat d.

3. Assume a two-level memory system (consisting of main memory plus virtual memory) has the following parameters:

Logical address space: 4 GB - byte addressable
 Physical memory: 1 GB
 Virtual memory: Paged, using a page size of 4096 bytes

- Show - by using a diagram similar to the ones presented in class - how the bits of a virtual address will be interpreted as page number and byte in page. Show specific bit numbers.
- How many bits will be needed in a page table entry to specify a frame number?
- Suppose each page table entry is 4 bytes and holds a frame number and a valid bit. (This will waste some bits). How much memory (in bytes) will be needed to store the page table for a process that is allocated 128 MB of virtual memory?
- Suppose that, at some time, the page table for a certain process has a size of 6 entries, and looks like this:

Page	Valid	Protection*	Frame (hexadecimal)
0	0	(not applicable)	
1	1	RW	12345
2	1	RO	0002A
3	1	RW	11111
4	1	RW	22222
5	0	(not applicable)	
6	1	RO	33333

* RW = read/write; RO = read only

Give the physical address that will correspond to each of the following virtual addresses if the virtual address is valid. If the virtual address is invalid, explain why it is invalid.

- Read 00002123
- Write 00006789
- Read 00005174
- Read 00007211
- Write 00001743

4. Assume a three-level memory system (consisting of a cache, main memory, and virtual memory with the MMU between the cache and the main memory) has the following parameters:

cache hit time 1 ns
 cache hit rate 90 %
 TLB hit rate 95 % * [of pages where the cache misses]
 page fault rate 0.01 % [of pages where the TLB misses]
 main memory access time 100 ns [includes time to update TLB and/or cache]
 disk access time 10 ms [includes time to update page table, TLB, and cache]
 if there is a page fault, 30% of the time a dirty page has to be replaced

(* Assume that the page table is not itself subject to mapping so looking up a translation in the page table can always be done with a single address translation)

Calculate the average memory access time (AMAT) for this system. Show your work carefully - do not just give an answer!

5. Assume a system has word-addressable memory with a direct-mapped cache with 16 one-word blocks that are initially empty. Consider the following series of memory references (where the values given are word numbers). Label each as a cache “hit” or “miss”, and indicate the final contents of the cache:

1 4 8 5 20 17 19 56 9 11 4 43 5 6 9 17

6. Repeat the preceding problem, but assume that the cache is organized as 4 four-word blocks, with a complete block of four words being read into the cache from main memory whenever any word in a block is needed

7. Repeat problem 5, but assume that the cache is two-way set associative, organized as 8 sets, each consisting of a pair of one-word blocks. LRU (least recently used) is used to select which element of a set to replace if both are valid and another reference collides.

8. Consider a system that uses a Hamming code for error detection/correction. Each data item is 11 bits long - which means that 4 correcting bits are used. Odd parity is used for the correcting bits, but no additional parity bit is used.

- a. Give the word that would actually be stored for the following data word. (Remember that the error correcting bits are interspersed with the data bits)

10010110101

- b. Suppose the following word is retrieved from memory. What is the correct value of the data word?

100101100100111