Virginia Tech - ECE 5484 - Summer 2020

Homework 5

Before starting this homework assignment, please be sure that you have completed all of the following activities.

- View the relevant online lectures and read associated sections in the textbook before or while you work on this homework assignment.
- Review the course syllabus. Note the grading policies, including policies for submitting assignments.
- Review the course schedule. Note the due dates for course assignments, including this one.
- Review the Graduate Honor System at https://graduateschool.vt.edu/academics/expectations/graduate-honor-system.html. Review the Graduate Honor System Constitution, especially Articles I (Sections 1, 2, and 3), V, VI, VII, VIII, and IX.

Please note the following.

- Solutions must be clear and presented in the order assigned. Solutions must show work needed, as appropriate, to derive your answers. Written answers should be concise, but sufficiently complete to answer the question. Neat hand drawings, where needed, are acceptable. Your final solution for each problem must be easily identified.
- At the top of the first page, include: your name (as recorded by the university); your email address; and the assignment name ("ECE 5484, Homework 5"). Do *not* include your Virginia Tech ID number or your social security number.
- Homework must be submitted as a PDF (.pdf) file with the file name *lastname_firstname*_HW5.pdf, where *lastname* is your last or family name and *firstname* is your first or given name. Submit a single file.
- Submit your assignment using the Assignments area of the class website. You must submit your assignment by 11:55 p.m. on the due date.

---

1. Suppose a computer using direct mapped cache has 2^{64} bytes of byte-addressable main memory, and a cache of 2048 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, block, and offset fields?
   c) To which cache block will the memory address 0x00000000000163FA map?

2. 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 can the memory address 0x01D872 map?

3. 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?

4. Suppose we have 2^{20} bytes of virtual memory, 2^{16} bytes of physical main memory and the page size is 2^8 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?

5. For the system in problem 4, suppose a main memory access requires 30ns, the page fault rate is .01%, it costs 12ms to access a page not in memory (this time includes the time necessary to transfer the page into memory, update the page table, and access the data). Also suppose a TLB hit requires 7ns, the cache miss rate is 3%, the TLB hit rate is 95%, a cache hit requires 15 ns. On a TLB or cache miss, the time required for access includes a TLB and/or cache update, but the access is not restarted. On a page fault, the page is fetched from disk, all updates are performed but the access is restarted. All references are sequential (no overlap, nothing done in parallel)

   a) Calculate the time for a TLB hit and a cache hit.
   b) Calculate the EAT (effective access time) for a TLB hit.

6. A system implements a paged virtual address space for each process using a one-level page table. The maximum size of virtual address space is 32MB. The page table for the running process includes the following valid entries (the $\rightarrow$ notation indicates that a virtual page maps to the given page frame, that is, it is located in that frame):

   Virtual page 3 $\rightarrow$ page frame 6
   Virtual page 5 $\rightarrow$ page frame 10
   Virtual page 8 $\rightarrow$ page frame 5
   Virtual page 4 $\rightarrow$ page frame 2
   Virtual page 0 $\rightarrow$ page frame 1

The page size is 2048 bytes and the maximum physical memory size of machine is 4MB.

a) How many bits are required for each virtual address?
b) How many bits are required for each physical address?
c) What is the maximum number of entries in a page table?
d) To which physical address will the virtual address 0x37F translate?
e) Which virtual address will translate to physical address 0x1203?