**COMP4300 Spring 2019**

Homework 2 Solutions

1. Suppose you are designing a cache for a machine with 40-bit addresses. The cache is 1MB in size. Cache blocks are 64 bytes.
   1. How many blocks can be held in the cache
   2. How many bits of the address are devoted to the offset?
   3. If the cache is direct-mapped, how many bits are devoted to the tag and index?
   4. If the cache is 4-way set associative, how many bits are devoted to the tag and index? How many sets are there?
   5. If the cache is fully associative, how many bits are devoted to the tag and index?

*Solution:*

a. 1MB  *= 220*

64 bytes = 26

So there are 220-6 = 214 = 16k of cache blocks

b. A cache line has 64 bytes, so 6 bits are needed for the offset

c. In a direct mapped cache, only one block can go in each cache block, so 14 bits are needed to index the 16K of caches blocks, leaving 40-14-6 = 20 bits for the tag tag.

d. If the cache is 4-way set associative, there are 4 blocks per set, and hence 4K of sets, so 12 bits needed for the index, leaving 22 bits for the tag.

e. If the cache is fully associative, there is no index (0 bits), leaving 36 bits for the tag.

1. Suppose you have a machine with separate I- and D- caches. The miss rate on the I-cache is 2%, and on the D-cache 3%. On an I-cache hit, the value can be read in the same cycle the data is requested. On a D-cache hit, one additional cycle is required to read the value. The miss penalty is 100 cycles for either cache. 35% of the instructions on this RISC machine are LW or SW instructions, the only instructions that access data memory. A cycle is 2ns. What is the average memory access time?

*Solution:*

For 100 instructions, there will be 100 instruction reads and 35 data reads. Thus 100/135 or 74.07% of memory accesses are instruction memory accesses, and the remainder, or 25.93% are data memory accesses. The avg memory access number of cycles for instruction memory is : 1 + 100\*0.02 = 1 + 2 = 3

For data memory it is: 2 + 100\*0.03 = 2 + 3 = 5

For all memory accesses the average cycles are 0.7407\*3 + 0.2593\*5 = 2.22 + 1.30 = 3.52.

The average memory access time is 3.52 cycles \*2 ns/cycle = 7.04ns