Computation Structures — Tutorial 9

November 28, 2017

Cache Memory

1. Given the following sequence of memory accesses:

   2, 3, 11, 16, 21, 13, 64, 48, 14, 11, 3, 22, 11

   (a) Give cache misses and cache hits as well as the final content of cache memory for a system featuring a **totally associative cache** that can contain up to **8 32-bit words** and that applies an **LRU** replacement policy.

   (b) Give cache misses and cache hits as well as the final content of cache memory for a system featuring a **totally associative cache in blocks of 4 32-bit words** that can contain up to **16 32-bit words** and that applies an **LRU**. How long are keys in this cache?

2. If an access to cache memory takes 1 clock cycle, and if a cache miss takes 5 more cycles, which cache hit rate is required in order to get an average access time to memory of 2 cycles?

3. Consider the following program:

   ```
   int array[1000];
   for (i = 0; i < 1000; i++)
       for(j = 0; j < 1000; j++)
           A[i] = A[i] + 1;
   ```

   This program is compiled without any optimization and run on a processor featuring a **totally associative cache in blocks of 4 32-bit words** with **delayed modification** ("write-back"). The cache size is **1KB**. What is the cache miss rate for data?

4. The following schema represents a **block direct mapped cache** for a machine handling 32-bit words and addresses.

   (a) How many words from main memory can be stored simultaneously in this cache?

   (b) How many address bits are used to select the cache line to access?

   (c) What size is the `TAG` field?

   (d) Suppose that the word stored at address `0x0002045C` in main memory is currently cached. Where can we find this word in the cache? What should be the value of the corresponding `TAG` field?

   (e) Can the words located at addresses `0x00012368` and `0x00322FF8` in main memory be simultaneously cached?

   (f) What about the words at `0x02536038` et `0x00001034`?