CS-UY 2214 — Recitation 10

Introduction

Complete the following exercises. Unless otherwise specified, put your answers in a plain text file named `recitation10.txt`. Number your solution to each question. When you finish, submit your file on Gradescope. Then, in order to receive credit, you must ask your TA to check your work. Your work should be completed and checked during the recitation session.

Problems

1. As we discussed in class, an address can be divided into parts representing the tag, index, and block offset.

   Below, we show how a 16-bit address can be divided for use in a particular direct-mapped cache. Each field is expressed as a range of bits (similar to how we express bit ranges in Verilog).

Tag	Index	Offset
15:10	9:4	3:0

   For these questions, assume each memory cell contains 1 byte.

   (a) What is the blocksize of this cache? Give your answer in bytes.

   (b) What is the size of the entire cache (excluding metadata)? Give your answer in bytes.

2. Assume a direct-mapped cache with 8 rows and a blocksize of 1 cell. The following memory addresses are accessed in sequence. For each of the following memory addresses, indicate (a) its row, (b) its tag, and (c) whether it will result in a hit, a miss, or a miss with eviction.

   ```
   [12, 13, 6, 0, 2, 5, 14, 14, 1, 8]
   ```

3. Repeat the previous exercise, but this time assume a direct-mapped cache with 4 rows and a blocksize of 2 cells.

4. Repeat the previous exercise, but this time assume a direct-mapped cache with 2 rows and a blocksize of 4 cells.

5. Assume a system with 32-bit pointers and 512-byte blocks. Its cache is 2 MB, excluding metadata. Each memory cell is one byte.

   (a) How many blocks can be stored in the cache?

   (b) How many bits is the block offset?

   (c) Assuming a direct-mapped cache, how many bits is the index (i.e. the component of the address that indicates the cache row)?

   (d) Assuming a direct-mapped cache, how many bits is the tag?

6. For the following questions, assume that we have a 4 KB direct-mapped cache with a block size of 32 bytes. Each memory cell stores one byte.

   (a) A particular program performs `lw` instructions on a long sequence of consecutive memory addresses, i.e. 0, 1, 2, 3, 4, 5, 6, ..., etc. Assuming the cache is initially empty, what would be the hit ratio of such a sequence of accesses? Does the total size of the cache affect the hit ratio?

   (b) It takes 5ns to access the cache (regardless of whether it is a hit or miss). It takes 80ns to access main memory.

   What would be the average memory access time (AMAT) for this cache with the workload specified above? That is, how long does each `lw` take on average?

   The AMAT is calculated as the cache access time, plus the main memory access time times the miss ratio.