Lab 10: Cache
CSE-031-01 **Points: 20**

Overview

In this lab, we will investigate how cache is organized in a CPU.

Getting Started

Before we begin any activities, create a directory (`Lab_10`) inside the `CSE31` directory we created in the first lab. You will save all your work from this lab here. **Note that all the files shown in green below are the ones you will be submitting for this assignment.**

You must have a clear idea of how to answer the TPS questions before leaving lab to receive participation score.

Direct-mapped Cache

TPS (Think-Pair-Share) activity 1: Discuss questions 1 – 8 (30 minutes) while paired with your classmates assigned by your TA (you will be assigned to groups of 3-4 students) and record your answers in a text file named **`tpsAnswers.txt`** under a section labelled "TPS 1" (***you will continue to use this file to record your answers to all the TPS questions that follow in the lab handout***):

1. Study the lecture slides on cache included with the assignment. What is cache? Why do we need cache?
2. There are generally 2 practical ways to organize a cache: Direct-mapped cache and N-way set associative cache. In both types of cache, data at a specific address of the main memory (RAM) are mapped to a pre-defined location in the cache. A "**Block**" is the basic unit of data being mapped between the main memory and cache. The size of a block depends on the specification of a cache. Every time data is transferred between cache and the main memory, it is **a block of data** being transferred. In this exercise, we will explore the Direct-mapped cache.
3. In a Direct-mapped cache, the cache is organized as a hash table. Addresses of the main memory are mapped to the indices of the cache (block numbers) using a modulo operator (%) as the hash function. As a result, we can divide a memory address into 3 fields: tag, index, offset.
4. Offset bits tell us how many bytes of data are in a block. These bits are the ***right-most bits of the memory address***. You can consider this as the **number of columns of data** in a cache. With a specific value of the offset bits from an address, we know which column of a block we are trying to access. Given the block size of a cache is 16B (bytes), how many bits do we need for offset? What is the number of bits in offset as a function of block size? Is it practical to have a cache of block size = 1 byte?
5. Index bits tell us how many blocks there are in a cache. These bits are ***the next right-most bits of the memory address after the offset bits***. You can consider this as the **number of blocks (rows) of data** in a cache. With a specific value of the index bits from an address, we know which block (row) we are trying to access. Given there are 64 blocks in a cache, how many index bits do we need? What is the number of bits in index as a function of number of blocks?
6. Once you know the number of blocks and the block size of a cache, do you know the total size of the cache? How?
7. Since the size of cache is always smaller than the size of the main memory, the sum of bits of the offset and index of a cache will be less than the number of bits in an address of the main memory. What do we do to the left-over bits from the address? Why are they important?
8. Given a memory address of 20 bits (during Intel 8086 era), **128B** of **direct-mapped** cache, and **8B block size**, answer the following questions:
   a. How big is this main memory?
   b. How many offset bits?
   c. How many blocks are there in the cache?
   d. How many index bits?
   e. How many tag bits?

f. Draw the layout of the cache: including tags, valid bits, dirty bits, and data blocks.
g. What is the number of bits per row of the cache (number of bits being used in a row: tag, valid bit, dirty bits, and data block)?

Your TA will "invite" one of you randomly after the activity to share what you have discussed.

N-way Set Associative Cache

TPS activity 2: Discuss questions 1 – 4 (30 minutes) with your TPS partners in your assigned breakout room and record your answers in **`tpsAnswers.txt`** under a section labelled "TPS 2" (use the lecture slides on cache included with the assignment to answer these questions):

1. What is the disadvantage of a Direct-mapped cache? What kind of cache miss will it introduce?
2. To overcome this problem, we can allow multiple blocks of data to occupy the same ***set*** of a cache. Note that we use "**set**" here instead of index of cache. In this organization, we group N blocks (rows) of cache into a set and allow more than one block of data to stay within a set. The layout of the cache remains the same as its direct-mapped version, but the difference is that every N blocks are now being grouped into a set.
3. The memory address is still partitioned into the same 3 fields, but the index bits now refer to the set number. Given a cache with 1024 blocks and the associativity is 4 (4 blocks per set), how many index bits do we need? What is the number of bits in index as a function of number of blocks and associativity?
4. Given a memory address of 20 bits (during Intel 8086 era), **128B** of **2-way** cache, and **8B block size**, answer the following questions:
   a. How big is this main memory?
   b. How many offset bits?
   c. How many blocks are there in the cache?
   d. How many sets are there in the cache?
   e. How many index bits?
   f. How many tag bits?
   g. Draw the layout of the cache: including tags, valid bits, dirty bits, and data blocks. Indicate the sets with a different color (or a thicker) boarder.
   h. What is the number of bits per row of the cache (number of bits being used in a row: tag, valid bit, dirty bits, and data block)?

Your TA will "invite" one of you randomly after the activity to share what you have discussed.

Individual Assignment 1: Cache in Your Computer

Perform the following:

- Visit https://www.cpu-world.com/cgi-bin/CPUID.pl.
- In the **Filter submissions:** window on the left, select *Intel* under **Manufacturer:** and select *Core i5* or *Core i7* or *Core i9* (your choice) under **Family:** and click on **Filter**.
- In the filtered results randomly select a processor.

Answer the following questions in a text file named **individualAssign1.txt**:

1. What is the full name of the processor you selected?
2. How many levels of caches does your CPU have (L1, L2, L3, etc.)? Is there a separate L1 cache for data and instructions?
3. How big is each level of cache?
4. What is the block size (sometimes it is called line size)?
5. Are the caches direct-mapped or set associative? If set associative, how many ways?
6. With L1 data cache, how many tag bits, index bits, and offset bits?

Collaboration

You must credit anyone you worked with in any of the following three different ways:

1. Given help to
2. Gotten help from
3. Collaborated with and worked together

What to hand in

When you are done with this lab assignment, submit all your work through CatCourses.

Before you submit, make sure you have done the following:

- Attached your `individualAssign1.txt` and `tpsAnswers.txt`.
- Filled in your collaborator's name (if any) in the "Comments…" textbox at the submission page.

Also, remember you **DO NOT NEED TO demonstrate** to the TA or instructor for this assignment.

Scoring:

- TPS activity 1: 6pts (total).
- TPS activity 2: 5pts (total).
- Individual Assignment 1: 9pts (demo is not required for this assignment).