Memory & Caches II
CSE 351 Winter 2024

Instructor:
Justin Hsia

Teaching Assistants:
Adithi Raghavan
Aman Mohammed
Connie Chen
Eyoel Gebre
Jiawei Huang
Malak Zaki
Naama Amiel
Nathan Khuat
Nikolas McNamee
Pedro Amarante
Will Robertson

https://what-if.xkcd.com/111/
Relevant Course Information

- HW14 due tonight, HW15 due Wednesday, HW16 due Friday

- Lab 3 due Friday (2/16), late deadline is Monday (2/19)
  - President’s Day: no lecture, but some support hours (see Ed)

- Midterm grades will be released when we can
  - Regrade requests will be available afterward
Mid-Quarter Survey Summary

❖ Lessons:
  ▪ There’s a lot of content… some lessons (videos) are too long
  ▪ Wish there were more practice problems and worked examples
  ▪ Would like if the slides used in the videos were posted (will do)
  ▪ Organization of Ed Lessons could be improved (FYI, website schedule by due dates)

❖ Lecture:
  ▪ Would like a different balance of content review & work time (but no agreement)
  ▪ Would like more practice problems and worked examples

❖ Section: want more practice problems (?)

❖ Support hours: want more total, more later in day, more on Zoom
Caches II
Lesson Summary (1/2)

❖ Cache parameters define the cache geometry:
  - **Block size** is number of bytes per block
  - **Cache size** is number of bytes (or blocks) of data the cache can hold

❖ Finding a byte in the cache:
  - **Offset** refer to which byte in block
  - **Index** refers to which block in cache

❖ **Example:**
  - \( K = 4 \text{ B}, C = 16 \text{ B} = 4 \text{ blocks} \)
Lesson Summary (2/2)

❖ **Direct-mapped cache:** each block in cache is assigned a unique index
  - Uses hash function of (block number) mod (# of cache indices)
    - Deterministic placement of each block, with many blocks mapping into the same index
    - Tag bits stored in cache and used to distinguish between blocks that map to same index

❖ **Accessing the cache:**
  (TIO address breakdown)
  1) **Index** field tells you where to look in cache (width $s = \log_2 S$)
  2) **Tag** field lets you check that data is the block you want (width $t = m - s - k$)
  3) **Offset** field selects specified start byte within block (width $k = \log_2 K$)
Lesson Q&A

❖ Learning Objectives:
   ▪ Determine how memory addresses and data interact with the cache (i.e., cache lookups, data movement).
   ▪ Analyze how changes to cache parameters [and policies] affect performance metrics such as AMAT.

❖ What lingering questions do you have from the lesson?
   ▪ Chat with your neighbors about the lesson for a few minutes to come up with questions
Caches II – Practice
Polling Questions (1/2)

❖ We have a direct-mapped cache with the following parameters:
  ▪ Block size of 8 bytes
  ▪ Cache size of 4 KiB

❖ How many blocks can the cache hold?
❖ How many bits wide is the block offset field?
❖ Which of the following addresses would fall under block number 3?
  A. 0x3       B. 0x1F       C. 0x30       D. 0x38
Polling Questions (2/2)

Based on the following behavior, which of the following block sizes is NOT possible for our cache?

- Cache starts *empty*, also known as a *cold cache*
- Access (addr: hit/miss) stream:
  - (0xE: miss), (0xF: hit), (0x10: miss)

A. 4 bytes  
B. 8 bytes  
C. 16 bytes  
D. 32 bytes
Homework Setup (1/2)

```c
struct WolfPos {
    float x;
    float y;
    float z;
    int id;
};
struct WolfPos grid[16][16];
```

- Assume `&grid = 0`

- What are the addresses of the following pieces of data?
  - `&(grid[0][0].id) =`
  - `&(grid[1][0].y) =`
  - `&(grid[3][4].x) =`
struct WolfPos {
    float x;
    float y;
    float z;
    int id;
};

struct WolfPos grid[16][16];

- Assume &grid = 0

- Cold direct-mapped cache with $C = 1024$ B and $K = 16$ B
  - What happens if we access grid[0][0].x and then grid[4][0].x?
Group Work Time

❖ During this time, you are encouraged to work on the following:
  1) If desired, continue your discussion
  2) Work on the homework problems
  3) Work on the lab (if applicable)

❖ Resources:
  ▪ You can revisit the lesson material
  ▪ Work together in groups and help each other out
  ▪ Course staff will circle around to provide support