Memory & Caches I
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
Relevant Course Information

❖ HW14 due Monday, HW15 due Wednesday

❖ Lab 3 due next Friday (2/16)
  ▪ Make sure to look at HW14 before starting

❖ Midterm starts tomorrow (2/8-10)
  ▪ Only private posts on Ed Discussion
  ▪ Staff cannot help you study during the exam window – only point you to resources and clarify the questions
  ▪ We will post clarifications and corrections about the exam on Ed as we go
Caches I
Lesson Summary (1/3)

- IEC prefixes are unambiguously powers of 2:

<table>
<thead>
<tr>
<th>SI Size</th>
<th>Prefix</th>
<th>Symbol</th>
<th>IEC Size</th>
<th>Prefix</th>
<th>Symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>10^3</td>
<td>Kilo-</td>
<td>K</td>
<td>2^10</td>
<td>Kibib-</td>
<td>Ki</td>
</tr>
<tr>
<td>10^6</td>
<td>Mega-</td>
<td>M</td>
<td>2^20</td>
<td>Mebib-</td>
<td>Mi</td>
</tr>
<tr>
<td>10^9</td>
<td>Giga-</td>
<td>G</td>
<td>2^30</td>
<td>Gibib-</td>
<td>Gi</td>
</tr>
<tr>
<td>10^{12}</td>
<td>Tera-</td>
<td>T</td>
<td>2^40</td>
<td>Tebib-</td>
<td>Ti</td>
</tr>
<tr>
<td>10^{15}</td>
<td>Peta-</td>
<td>P</td>
<td>2^50</td>
<td>Pebib-</td>
<td>Pi</td>
</tr>
<tr>
<td>10^{18}</td>
<td>Exa-</td>
<td>E</td>
<td>2^60</td>
<td>Ebibib-</td>
<td>Ei</td>
</tr>
<tr>
<td>10^{21}</td>
<td>Zetta-</td>
<td>Z</td>
<td>2^70</td>
<td>Zebib-</td>
<td>Zi</td>
</tr>
<tr>
<td>10^{24}</td>
<td>Yotta-</td>
<td>Y</td>
<td>2^80</td>
<td>Yobib-</td>
<td>Yi</td>
</tr>
</tbody>
</table>

\[ 2^{XY} \text{ “things”} = 2^X \text{ “things”} + 2^Y \text{ “things”} \]
Lesson Summary (2/3)

❖ Memory Hierarchy

- Successively higher levels contain “most used” data from lower levels
- Caches are intermediate storage levels used to optimize data transfers between any system elements with different characteristics
- Exploits *temporal and spatial locality*:

![Memory Hierarchy Diagram](image)
Lesson Summary (3/3)

- **Cache Performance**
  - Ideal case: found in cache (*cache hit*), return requested data immediately
  - Bad case: not found in cache (*cache miss*), search in next level
    - Bring entire *cache block* containing requested data into this cache once found
  - **Average Memory Access Time** (AMAT) = HT + MR × MP
    - Hurt by Miss Rate and Miss Penalty

\[
\begin{align*}
\text{Hit} & \text{ takes } HT \\
\text{Miss} & \text{ takes } HT + MP
\end{align*}
\]
Lesson Q&A

❖ Learning Objectives:

- Describe the memory hierarchy and explain the relationship between cost, size, and access speed of its layers.
- 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 I – Practice
Polling Questions (1/2)

❖ Convert the following to or from IEC:
  ▪ 512 Ki-books
  ▪ $2^{27}$ caches

❖ Compute the average memory access time (AMAT) for the following system properties:
  ▪ Hit time of 1 ns
  ▪ Miss rate of 1%
  ▪ Miss penalty of 100 ns
Polling Questions (2/2)

❖ Processor specs: 200 ps clock, MP of 50 clock cycles, MR of 0.02 misses/instruction, and HT of 1 clock cycle

AMAT =

❖ Which improvement would be best?

A. 190 ps clock

B. Miss penalty of 40 clock cycles

C. MR of 0.015 misses/instruction
Caches I – Context
AMAT, Revisited

- *Average Memory Access Time (AMAT)*: average time to access memory considering both hits and misses
  
  \[
  \text{AMAT} = \text{Hit time} + \text{Miss rate} \times \text{Miss penalty}
  \]
  
  (abbreviated AMAT = HT + MR × MP)

- We called this a *cache performance metric*
  
  - This isn’t the only metric we could have used!
Metrics in Computing

❖ Generally, folks care most about performance
  ▪ Energy-efficiency is more important now since the plateau in 2004/2005
  ▪ This is why we have so many specialized chips nowadays

❖ Really, this is just efficiency – making efficient use of the resources that we have
  ▪ Performance: cycles/instruction, seconds/program
  ▪ Energy efficiency: performance/watt
  ▪ Memory: bytes/program, bytes/data structure
Metrics

❖ What do we do with metrics?
  ▪ We tend to optimize along them!
  ▪ Especially when jobs/funding depend on better performance along some metric
    • See all of Intel under “Moore’s Law”

❖ Sometimes, strange incentives emerge
  ▪ “Minimize the number of bugs on our dashboard”
    • Does it count if we make the bugs invisible?
  ▪ “Make this faster for our demo in a week”
    • Shortcuts might hurt performance at scale
  ▪ “Minimize our average memory access time”
    • What if we add more memory accesses that we know will hit?
Metrics and Success

- Success is *defined along metrics*
  - This affects how we measure and optimize

- Let’s say that we choose *performance/program* or *performance/program set* (*i.e.*, benchmarks):
  1. Measure existing performance
  2. Come up with a bunch of optimizations that would improve performance
  3. Select a few to build into the “next version”
Metrics and Success

- Success is *defined along metrics*
  - This affects how we measure and optimize

- Let’s say that we choose **profit/year** or **stock price**:
  - Success means earning more profit than last year
  - Improvement or optimizations might include:
    - Reduce expenses, cut staff
    - Sell more things or fancier things (e.g., in-app purchases)
    - Make people pay monthly for things they could get for free
    - Increase advertising revenue:

---

*Whistle-Blower Says Facebook ‘Chooses Profits Over Safety’*

Frances Haugen, a Facebook product manager who left the company in May, revealed that she had provided internal documents to journalists and others.
Metrics and Success

❖ Success is *defined along metrics*
  ▪ This affects how we measure and optimize

❖ Let’s say that we choose *minoritized participation in computing*:
  ▪ What does success/participation mean (and dangers)?
    • Women? BIPOC? All minoritized lumped together?
      – Might optimize for one group at the expense of others
    • Taking intro? Passing intro? Getting a degree? Getting a job?
      – Says nothing about retention or participation/decision-making level
Design Considerations

❖ Regardless of what we build, the way that we define success shapes the systems we build

  ▪ Choose your metrics carefully
  ▪ There’s more to choose from than performance (e.g., usability, access, simplicity, agency)

❖ Metrics are a “heading” (in the navigational sense)

  ▪ Best to reevaluate from time to time in case you’re off course or your destination changes
Discussion Questions

❖ Discuss the following question(s) in groups of 3-4 students
  ▪ I will call on a few groups afterwards so please be prepared to share out
  ▪ Be respectful of others’ opinions and experiences

❖ Let’s say your (main) metric for college is to get a 4.0 overall GPA.
  ▪ What are some potential unintended consequences of this metric?
  ▪ What are some other potential metrics you could use for college?
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