Today’s Outline

- **Admin:**
  - HW #5 due Tuesday, Feb 24th at 11:45pm
  - If you are working with a partner, must email Sean by Friday, Feb 20th at 11:45pm
  - Midterm #2 next Friday, Feb 27th

- **Hashing**
- **Memory Hierarchy and Locality**

---

Why do we need to know about the memory hierarchy/locality?

- One of the assumptions that Big-Oh makes is that all operations take the same amount of time.
- Is that really true?

---

Definitions

- **Cycle** – (for our purposes) the time it takes to execute a single simple instruction. (ex. Add 2 registers together)
- **Memory Latency** – time it takes to access memory

---

Moore’s Law

- CPU (has registers)
- SRAM 8KB - 4MB
- DRAM up to 10GB
- Main Memory
- Disk

- Time to access:
  - 1 ns per instruction
  - Cache: 2-10 ns
  - Main Memory: 40-100 ns
  - Disk: a few milliseconds (5-10 Million ns)
Processor-Memory Performance Gap

- x86 CPU speed (100x over 10 years)

What can be done?

- **Goal**: Attempt to reduce the number of accesses to the slower levels.
- **How?**

Locality

**Temporal Locality** (locality in time) – If an item is referenced, it will tend to be referenced again soon.

**Spatial Locality** (locality in space) – If an item is referenced, items whose addresses are close by will tend to be referenced soon.

Caches

- Each level is a **sub-set** of the level below.

  - **Cache Hit** – address requested is in cache
  - **Cache Miss** – address requested is NOT in cache
  - **Cache line size** (chunk size) – the number of contiguous bytes that are moved into the cache at one time

Examples

\[
\begin{align*}
x &= a + 6; \\
y &= a + 5; \\
z &= 8 \times a;
\end{align*}
\]

\[
\begin{align*}
x &= a[0] + 6; \\
y &= a[1] + 5; \\
z &= 8 \times a[2];
\end{align*}
\]

Locality and Data Structures

- Which has (at least the potential for) better spatial locality, arrays or linked lists?