Memory Hierarchy

CSE 373
Data Structures & Algorithms
Ruth Anderson
Spring 2008

```
public interface PriorityQueue
  <E extends Comparable<? super E>> { ...

public class ThreeHeap
  <E extends Comparable<? super E>>
          implements PriorityQueue<E> { ...

private E[] nodes;
...
nodes = (E[]) new Comparable[DEFAULT_CAPACITY];
```


Today's Outline

- **Admin**:
  - HW #4 due Thursday at 11:59pm
  - Printouts and written problems due at the beginning of class Friday
  - Late Penalty = -25% per 24 hours, submit via email (including electronic version of written problems)
- **Memory Hierarchy and Locality**

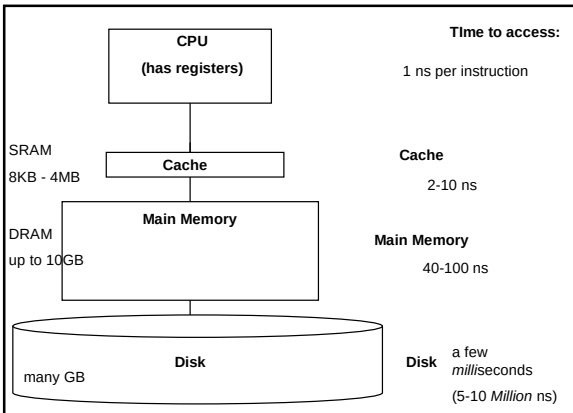
Why do we need to know about the memory hierarchy/locality?

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

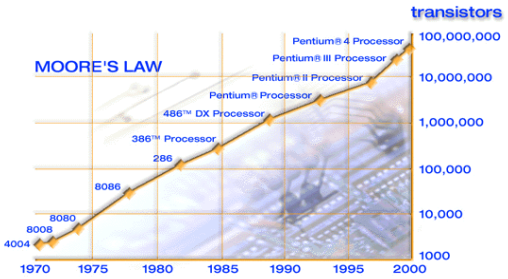
Definitions

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

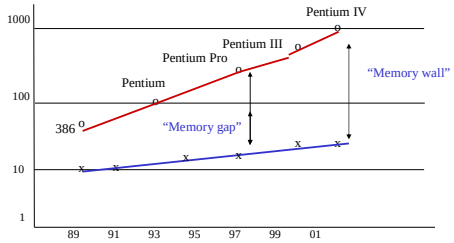


Moore's Law



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

```
x = a + 6;
y = a + 5;
z = 8 * a;
```

```
x = a[0] + 6;
y = a[1] + 5;
z = 8 * a[2];
```

Locality and Data Structures

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