Concepts Introduced in Chapter 7

- memory hierarchies and the principle of locality
- basic cache organizations
- virtual memory
• GOAL: Keep up with the demand for data from the pipeline(s)
• BANDWIDTH: The pipelines need a range of one word per cycle a few words per cycle depending on the CPU organization. The rate at which data is provided or required is called bandwidth.
• LATENCY: As with the delay of S cycles in an S stage pipeline in the CPU from beginning a particular operation until completing it, the amount of time it takes to access a particular data item is also crucial in a memory system. This delay is called the latency. It plays the role of the memory pipeline startup cost.
• MIX: In more advanced organizations the memory may need to read multiple words and write (typically one word) on each cycle
• BASIC COMPONENTS: There is a cost/size tradeoff of basic memory components
• CODE PROPERTIES: Memory accesses from typical codes tend to have structure that can be assumed and exploited
• USER VIEW: Memory hierarchies can be either transparent to the user or visible (and manageable) by the user
• SYSTEMS DISCUSSED HERE: We will discuss the use of the basic component properties and the code properties to design a transparent memory hierarchy
Memory Hierarchy

- Exploits the principle of locality.
- Use a small fast (expensive) memory.
- Use a large slow (cheap) memory.
- All data found in one level is also found in the level below.
- Goals:
  - Catch most of the references in the fast memory.
  - Have most of the cost per byte be at the slow memory level.
- Will see that memory hierarchy is used to implement protection schemes as well.
Figure 7.1: The basic structure of a memory hierarchy.

- **Speed**: CPU
- **Fastest**: Memory
- **Slowest**: Memory

- **Size**: Smallest
- **Cost ($\text{billion}$)**: Highest
- **Size**: Biggest
- **Cost ($\text{billion}$)**: Lowest
### Table at the Bottom of Page 541

<table>
<thead>
<tr>
<th>Memory technology</th>
<th>Typical access time</th>
<th>$ per MByte in 1997</th>
</tr>
</thead>
<tbody>
<tr>
<td>SRAM</td>
<td>5-25 ns</td>
<td>$100-$250</td>
</tr>
<tr>
<td>DRAM</td>
<td>60-120 ns</td>
<td>$5-$10</td>
</tr>
<tr>
<td>Magnetic Disk</td>
<td>10-20 million ns</td>
<td>$0.10-$0.20</td>
</tr>
</tbody>
</table>
Components of a Memory Hierarchy

- registers (flip flops) - measured in words, managed by the compiler
- cache (SRAMs) - measured in Kbytes, managed by hardware
- main memory (DRAMs) - measured in Mbytes, managed by hardware
- disk - measured in Gbytes, managed by OS
- tape - measured in Gbytes and Tbytes, managed by system administrator
Figure 7.2: Every pair of levels in the memory hierarchy can be thought of as having an upper and lower level.

![Diagram showing data transfer between processor and memory levels]
Principle of Locality

- Temporal Locality - if an item is referenced, then it will tend to be referenced again soon
  - Instructions in loops are repeatedly referenced.
  - Instructions in recursive or utility functions tend to be repeatedly referenced.
  - Data inside loops (e.g. loop counter variables) are repeatedly referenced.
- Spatial Locality - if an item is referenced, then items whose addresses are close by will tend to be referenced soon
  - Instructions are accessed sequentially (unless there is a transfer of control).
  - Data elements of an array tend to be accessed sequentially.
Memory Hierarchy Terms

- Hit - item found in the upper level of the hierarchy
- Miss - item not found in upper level of the hierarchy
- Hit Time - time required to access the desired item in the upper level (includes time to determine if the access is a hit or a miss)
- Miss Penalty - the additional time required to service the miss
- Miss Rate - fraction of accesses that are not in that level
- Block - the amount of information that is retrieved from the next lower level on a miss
Figure 7.3: This diagram shows the structure of a memory hierarchy: as the distance from the CPU increases, so does the size.
Memory Hierarchy Questions

- Where can a block be placed in the current level? (Block placement)
- How is a block found if it is in the upper level? (Block identification)
- Which block should be replaced on a miss? (Block replacement)
- What happens on a write? (Write strategy)