The Five Classic Components of a Computer

Today's Topics:
- Memory technologies
- Technology trends
- Impact on performance
- Memory Hierarchy
- The principle of locality
- Memory hierarchy terminology

Memory Hierarchy Technology

Random Access:
- "Random" is good: access time is the same for all locations
- DRAM: Dynamic Random Access Memory
  - High density, low power, cheap, slow
  - Dynamic: need to be "refreshed" regularly
- SRAM: Static Random Access Memory
  - Low density, high power, expensive, fast
  - Static: content will last "forever" (until lose power)

"Non-so-random" Access Technology:
- Access time varies from location to location and from time to time
- Examples: Disk, CDROM

Sequential Access Technology: access time linear in location (e.g., Tape)

Main memory: DRAMs + Caches: SRAMs
Technology Trends

<table>
<thead>
<tr>
<th>Capacity</th>
<th>Speed (latency)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logic: 2x in 3 years</td>
<td>2x in 3 years</td>
</tr>
<tr>
<td>DRAM: 4x in 3 years</td>
<td>2x in 10 years</td>
</tr>
<tr>
<td>Disk: 4x in 3 years</td>
<td>2x in 10 years</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Year</th>
<th>Size</th>
<th>Cycle Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1980</td>
<td>64 Kb</td>
<td>250 ns</td>
</tr>
<tr>
<td>1983</td>
<td>256 Kb</td>
<td>220 ns</td>
</tr>
<tr>
<td>1986</td>
<td>1 Mb</td>
<td>190 ns</td>
</tr>
<tr>
<td>1989</td>
<td>4 Mb</td>
<td>165 ns</td>
</tr>
<tr>
<td>1992</td>
<td>16 Mb</td>
<td>145 ns</td>
</tr>
<tr>
<td>1995</td>
<td>64 Mb</td>
<td>120 ns</td>
</tr>
</tbody>
</table>

Who Cares About the Memory Hierarchy?

Processor-DRAM Memory Gap (latency)

- Processor: 60%/yr. (2X/1.5yr)
- DRAM: 9%/yr. (2X/10 yrs)

- "Moore's Law": grows 50% / year
Impact on Performance

° Suppose a processor executes at
  • Clock Rate = 200 MHz (5 ns per cycle)
  • CPI = 1.1
  • 50% arith/logic, 30% ld/st, 20% control
° Suppose that 10% of memory operations get 50 cycle miss penalty
° CPI = ideal CPI + average stalls per instruction
  = 1.1 + (0.30 (memory access/ins) x 0.10 (miss/memory access) x 50 (cycle/miss))
  = 1.1 cycle + 1.5 cycle
  = 2.6
° 58% of the time the processor is stalled waiting for memory!
° a 1% instruction miss rate would add an additional 0.5 cycles to the CPI!

The Goal: illusion of large, fast, cheap memory

° Fact: Large memories are slow, fast memories are small
° How do we create a memory that is large, cheap and fast (most of the time)?
  • Hierarchy
  • Parallelism
An Expanded View of the Memory System

- Processor
- Control
- Datapath
- Memory

- Speed: Fastest
- Size: Smallest
- Cost: Highest

- Memory
- Slowest
- Biggest
- Lowest

Why hierarchy works

- The Principle of Locality:
  - Program access a relatively small portion of the address space at any instant of time.

- Probability of reference

  0 2^n - 1

Address Space
Memory Hierarchy: How Does it Work?

- **Temporal Locality (Locality in Time):**
  - Keep most recently accessed data items closer to the processor

- **Spatial Locality (Locality in Space):**
  - Move blocks consists of contiguous words to the upper levels

**Memory Hierarchy: Terminology**

- **Hit:** data appears in some block in the upper level (example: Block X)
  - Hit Rate: the fraction of memory access found in the upper level
  - Hit Time: Time to access the upper level which consists of RAM access time + Time to determine hit/miss

- **Miss:** data needs to be retrieve from a block in the lower level (Block Y)
  - Miss Rate = 1 - (Hit Rate)
  - Miss Penalty: Time to replace a block in the upper level + Time to deliver the block the processor

- **Hit Time << Miss Penalty**
Memory Hierarchy of a Modern Computer System

- By taking advantage of the principle of locality:
  - Present the user with as much memory as is available in the cheapest technology.
  - Provide access at the speed offered by the fastest technology.

How is the hierarchy managed?

- Registers <-> Memory
  - by compiler (programmer?)

- cache <-> memory
  - by the hardware

- memory <-> disks
  - by the hardware and operating system (virtual memory)
  - by the programmer (files)
### Static RAM (SRAM)

**D latch**

- Address
- Chip select
- Output enable
- Write enable
- Din[7–0]

**SRAM**

- 32K x 8

**Three-state buffers**

- Select 0
- Data 0
- Enable
- Select 1
- Data 1
- Enable
- Select 2
- Data 2
- Enable
- Select 3
- Data 3
- Enable

**Output**

### 4 X 2 SRAM

**4-to-1 MUX**

- Write enable
- Address
- Din[1]
- Din[0]

**2-to-4 Decoder**

- Address
- Dout[1]
- Dout[0]
32K X 8 SRAM

Dynamic RAM (DRAM)

Dynamic RAM (DRAM)

DRAM cell

4M X 1 DRAM