Caches
Processor Performance

• Simple Measure: Millions of instructions per second
  – It takes 1 clock cycle to execute each of 5 stages of executing an instruction (IF, ID, EX, MEM, WB).
  – Your clock is 1 GigaHertz (1 ns clock)
  – 1 billion cycles/sec (divide) 5 cycles/instruction =
  – 200 Million instructions per Second = 200 MIPS

• But your memory requires 40 nanoseconds to access?
  – 1 billion cycles/sec (divide) (5+40) cycles/instruction =
  – 13.3 Millions instructions per Second = 22.2 MIPS
Problem: memory is slower than processor

Fast memory is expensive

Fast memory is small
Memory cost several years ago

Technology:

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>SRAM</td>
<td>0.5-2.5 ns</td>
<td>$2000-$5000</td>
</tr>
<tr>
<td>DRAM</td>
<td>50-70 ns</td>
<td>$20-$75</td>
</tr>
<tr>
<td>Mag. Disk</td>
<td>5<em>10^6 – 20</em>10^6 ns</td>
<td>$0.2-$2</td>
</tr>
</tbody>
</table>

Even if you have the money, size of fast memory is limited by physics.
Solution: memory hierarchy

- CPU
- Cache
- Main Mem.
- Disk

- 1cc
- 20 to 100cc
- $10^4$cc
Where is the cache?
Data that the CPU is likely to need in the future is stored in the cache.
Which data is “likely to be used in the future”?

- **Locality in Time:** If the CPU needed it recently, it is likely to need it again.
  - Loops
  - Variables
  - Top of the Stack

- **Locality in Space:** If the CPU needed something nearby, it is likely to need it again.
  - Subroutines
  - Arrays
Hits and Misses

Processor

hit ↓

Cache

miss

Main Memory

Hit time: memory access time when the data is in the cache.

Miss penalty: additional access time when there is a miss.

Hit rate: fraction of memory accesses found in cache

Miss rate: 1 - hit rate
Processor Performance

- It takes 1 clock cycle to execute each of 5 stages of executing an instruction (IF, ID, EX, MEM, WB) and a 1 GigaHertz (1 ns period) clock

- No cache
  - 1 billion cycles/sec (divide) (5+40) cycles/instruction =
  - 13.3 million instructions per Second = 22.2 MIPS

- 100% hit rate (impossible)
  - 1 billion cycles/sec (divide) 5 cycles/instruction =
  - 200 million instructions per Second = 200 MIPS

- 95% hit rate (not unreasonable)
  - 5 c/instruction * .95 + 45 c/instruction * .05 = (4.75 + 2.25)c/instruction = 7 cycles/instruct on average.
  - $10^9$ c/s / 7 c/instruct. = 143 MIPS
  - 143/22.2 = greater than 6X speedup
Example cache system

- Bottom of line Apple MacBook Pro, 2.3 GHz Intel i5 Dual Core
- On chip
  - L1 cache: 32 KB instruction and 32 KB data
  - L2 cache: 256 KB
  - L3 cache: 3072 KB