The Memory Hierarchy
What is a cache?
What problem do caches solve?
Memory

Abstraction: Big array of bytes

CPU

Memory
Processor vs Memory Performance

- Memory is very slow compared to processors.
How far away is the data?

\[10^9\] Tape / Optical Robot, 2,000 Years
\[10^6\] Disk, 2 Years
100 Memory, 1.5 hr
10 On Board Cache, 10 min
2 On Chip Cache
1 Registers, 1 min

© 2004 Jim Gray, Microsoft Corporation
SRAM and DRAM
Silicon Memories

• Why store things in silicon?
  • It’s fast!!!
  • Compatible with logic devices (mostly)
• The main goal is to be cheap
  • Dense -- The smaller the bits, the less area you need, and the more bits you can fit on a chip/wafer/through your fab.
  • Bit sizes are measured in $F^2$ -- the smallest feature you can create.
  • The number of $F^2$/bit is a function of the memory technology, not the manufacturing technology.
  • i.e. an SRAM in today’s technology will take the same number of $F^2$ in tomorrow’s technology.
Questions

• What physical quantity should represent the bit?
  • Voltage/charge -- SRAMs, DRAMs, Flash memories
  • Magnetic orientation -- MRAMs
  • Crystal structure -- phase change memories
  • The orientation of organic molecules -- various exotic technologies
  • All that’s required is that we can sense it and turn it into a logic one or zero.

• How do we achieve maximum density?
• How do we make them fast?
Anatomy of a Memory

- **Dense**: Build a big array
  - bigger the better
  - less other stuff
  - Bigger -> slower
- **Row decoder**
  - Select the row by raising a “word line”
- **Column decoder**
  - Select a slice of the row
- **Decoders are pretty big.**
The Storage Array

• Density is king.
  • Highly engineered, carefully tuned, automatically generated.
  • The smaller the devices, the better.

• Making them big makes them slow.
  • Bit/word lines are long (millimeters)
  • They have large capacitance, so their RC delay is long
  • For the row decoder, use large transistors to drive them hard.
  • For the bit cells...
    • There are lots of these, so they need to be as small as possible (but not smaller)
Measuring Memory Density

• We use a “technology independent” metric to measure the inherent size of different memory cells.
  • \( F \) = the “feature size” = the smallest dimension a CMOS process can create (e.g., the width of the narrowest wire).
  • In a 22nm process technology, \( F = 22\text{nm} \).
  • \( F^2 \) (F-squared) is the smallest 2D feature we can manufacture.
• A single bit of a given type of memory (e.g., SRAM or DRAM) requires a fixed number of \( F^2 \)
  • This number doesn’t change with process technology.
  • e.g., NAND flash memory is \( 4F^2 \) in 90nm and in 22nm.
• Using this metric is useful because the relative sizes of different memory technologies don’t change much, although absolute densities do.
Sense Amps

• Sense amplifiers take a difference between two signals and amplify it

• Two scenarios
  • Inputs are initially equal (“precharged”) -- they each move in opposite directions
  • One input is a reference -- so only one signal moves

• Frequently used in memories
  • Storage cells are small, so the signals they produce are inherently weak
  • Sense amps can detect these weak, analog signals and convert them into a logic one or logic zero.
Static Random Access Memory (SRAM)

- **Storage**
  - Voltage on a pair of cross-coupled inverters
  - Durable in presence of power

- **To read**
  - Pre-charge two bit lines to Vcc/2
  - Turn on the “word line”
  - Read the output of the sense-amp
SRAM Writes

• To write
  • Turn off the sense-amp
  • Turn on the wordline
  • Drive the bitlines to the correct state
  • Turn off the wordline
Building SRAM

- This is “6T SRAM”
- 6 transistors is pretty big
- SRAMs are not dense
SRAM Density

- At 65nm: 0.52\,\mu\text{m}^2
- 123-140 \, F^2
- [ITRS 2008]

65nm TSMC 6T SRAM
SRAM Ports

• Add word and bit lines
• Read/write multiple things at once
• Density decreases quadratically
• Bandwidth increase linearly
SRAM Performance

• Read and write times
  • 10s-100s of ps

• Bandwidth
  • Registers -- 324GB/s
  • L1 cache -- 128GB/s
DRAM
Dynamic Random Access Memory (DRAM)

- **Storage**
  - Charge on a capacitor
  - Decays over time (us-scale)
  - This is the “dynamic” part.
  - About $6F^2$: 20x better than SRAM

- **Reading**
  - Precharge
  - Assert word line
  - Sense output
  - Refresh data

Only one bit line is read at a time.
The other bit line serves as a reference.
The bit cells attached to Wordline 1 are not shown.
DRAM: Write and Refresh

- **Writing**
  - Turn on the wordline
  - Override the sense amp.

- **Refresh**
  - Every few milli-seconds, read and re-write every bit.
  - Consumes power
  - Takes time
DRAM Lithography:

How do you get a big capacitor?

\[ C \sim \frac{\text{Area}}{\text{dielectric-thickness}} \]

Stacked Capacitors
DRAM Lithography

Trench Capacitors
Accessing DRAM

- Apply the row address
  - “opens a page”
  - Slow (~12ns read + 24 ns precharge)
  - Contents in a “row buffer”

- Apply one or more column addr
  - fast (~3ns)
  - Reads and/or writes
DRAM Devices

• There are many banks per die (16 at left)
  • Multiple pages can be open at once.
  • Can keep pages open longer
  • Parallelism

• Example
  • open bank 1, row 4
  • open bank 2, row 7
  • open bank 3, row 10
  • read bank 1, column 8
  • read bank 2, column 32
  • ...

Micron 78nm 1Gb DDR3
DRAM: Micron MT47H512M4
DRAM: Micron MT47H512M4
DRAM Variants

• The basic DRAM technology has been wrapped in several different interfaces.
• SDRAM (synchronous)
• DDR SDRAM (double data-rate)
  • Data clocked on rising and falling edge of the clock.
• DDR2 -- faster, lower voltage DDR
• DDR3 -- even faster, even lower-voltage
• GDDR2-5 -- For graphics cards.
Current State-of-the-art: DDR3 SDRAM

- DIMM data path is 64bits (72 with ECC)
- Data rate: up to 1066Mhz DDR (2133Mhz effective)
- Bandwidth per DIMM GTNE: 16GB/s
  - guaranteed not to exceed
- Multiple DIMMs can attach to a bus
  - Reduces bandwidth/GB (a good idea?)

Each chip provides one 8-bit slice.
The chips are all synchronized and received the same commands
DRAM Scaling

- Long term need for performance has driven DRAM hard
  - complex interface.
  - High performance
  - High power.
- DRAM used to be the main driver for process scaling, now it’s flash.
- Power is now a major concern.
- Scaling is expected to match CMOS tech scaling
- $F^2$ cell size will probably not decrease
- Historical foot note: Intel got its start as a DRAM company, but got out of it when it became a commodity.
Back to Caching
## A Typical Hierarchy: Costs and Speeds

<table>
<thead>
<tr>
<th></th>
<th>Cost</th>
<th>Access time</th>
</tr>
</thead>
<tbody>
<tr>
<td>On-chip L1 cache SRAM KBs</td>
<td>???</td>
<td>&lt; 1ns</td>
</tr>
<tr>
<td>On-chip L2 cache SRAM KBs</td>
<td>???</td>
<td>&lt; 2-3ns</td>
</tr>
<tr>
<td>On-chip L3 cache SRAM MBs</td>
<td>???</td>
<td>&lt; 10ns</td>
</tr>
<tr>
<td>main memory GBs</td>
<td>0.009 $/MB</td>
<td>60ns</td>
</tr>
<tr>
<td>SSDs GB</td>
<td>0.0006 $/MB</td>
<td>20,000ns</td>
</tr>
<tr>
<td>Disk TBs</td>
<td>0.00004 $/MB</td>
<td>10,000,000ns</td>
</tr>
</tbody>
</table>
Typical Hierarchy: Architecture