**Announcements**

- Homework #6 due today
- Homework #7 due next Thurs.

**Class Material**

- Last lecture
  - Design for speed
  - Logical effort
- Today’s lecture
  - SRAM
  - Register files
- Reading (Chapters 12, 6)

**Random Access Memories (RAM)**

- **STATIC (SRAM)**
  - Data stored as long as supply is applied
  - Larger (6 transistors/cell)
  - Fast
  - Differential (usually)

- **DYNAMIC (DRAM)**
  - Periodic refresh required
  - Smaller (1-3 transistors/cell)
  - Slower
  - Single Ended

**Semiconductor Memory Classification**

<table>
<thead>
<tr>
<th>Random Access</th>
<th>Non-Random Access</th>
<th>Non-Volatile Read-Write Memory</th>
<th>Read-Only Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>SRAM</td>
<td>FIFO</td>
<td>EPROM</td>
<td>Mask-Programmed</td>
</tr>
<tr>
<td>DRAM</td>
<td>LIFO</td>
<td>E²PROM</td>
<td>Programmable</td>
</tr>
<tr>
<td></td>
<td>Shift Register</td>
<td>FLASH</td>
<td>(PROM)</td>
</tr>
<tr>
<td></td>
<td>CAM</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Random Access Chip Architecture

- Conceptual: linear array
  - Each box holds some data
  - But this does not lead to a nice layout shape
  - Too long and skinny

- Create a 2-D array
  - Decode Row and Column address to get data

Basic Memory Array

- **CORE:**
  - Keep square within a 2:1 ratio
  - Rows are word lines
  - Columns are bit lines
  - Data in and out on columns

- **DECODERS:**
  - Needed to reduce total number of pins; \( N+M \) address lines for \( 2^{N+M} \) bits of storage
  - Ex: if \( N+M=20 \), \( 2^{20} = 1 \text{Mb} \)

- **MULTIPLEXING:**
  - Used to select one or more columns for input or output of data

Positive Feedback: Bi-Stability

- \( V_{in} \) to \( V_{out} \) with positive feedback

Basic Static Memory Element

- If \( D \) is high, \( D_b \) will be driven low
- Which makes \( D \) stay high
- Positive feedback

Writing into a Cross-Coupled Pair

- Access transistor must be able to overpower the feedback

Meta-Stability

- Gain should be larger than 1 in the transition region
**Writing a Memory Cell**

**6-transistor CMOS SRAM Cell**

**Writing a “1”**

**SRAM Operation**

**Memory Cell**

**SRAM Operation**

Complementary data values are written (read) from two sides.

- Q_b will get pulled up when WL first goes high
- Reading the cell should not destroy the stored value
CMOS SRAM Analysis (Read)

\[
\Delta V = \frac{W_2/L_2}{W_1/L_1}
\]

Read Static Noise Margin

Obtained by breaking the feedback between the inverters

\[
\text{SNM}
\]

Write Static Noise Margin
6T-SRAM — Layout

Compact cell
Bitlines: M2
Wordline: bootstrapped in M3

65nm SRAM
- ST/Philips/Motorola

Access Transistor
Pull down  Pull up

SRAM Array Layout