Processor: Data Path Components

Memory Devices

Small: Register file (group of numbered registers)
Medium: SRAM (Static Random Access Memory)
Large: DRAM (Dynamic Random Access Memory)

SRAM: Static Random Access Memory

SRAM read port: data out

Large register files are impractical.
Big MUX = significant gate delay.
Large memories use a shared output line called a bit line.
No central gates/MUX to choose output!
**Wired ORs** (don't try this at home/in the lab, kids)

Data signal 1

Control signal 1

Data signal 2

Control signal 2

---

**Danger, Will Robinson!**

---

**SRAM cell**

one option

---

**SRAM write port:**

Data in: $D_{in}[1]$  $D_{in}[0]$

Write enable

Address

Select

Write enable

---

Data out: $D_{out}[1]$  $D_{out}[0]$

---

**Noninverting tristate buffers**

Data signal 1

Control signal 1

Data signal 2

Control signal 2

---

<table>
<thead>
<tr>
<th></th>
<th>In</th>
<th>Control</th>
<th>Out</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>Z</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>Z</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>
Organization of a 16 x 4 SRAM

(One option)

Selecting location 1101

But should we build a 32 to 2^10 decoder?

Another organization of a 16 x 4 SRAM

Split-level row/column addressing = physical multidimensional array!

Selecting location 0010

Nibbles "striped" across 4 smaller memories.
Selecting location 1101

Nibbles "striped" across 4 smaller memories.

What value does location 1010 hold?

Organization of a 4M x 8 SRAM

(one option)

= 4 MB memory, size of a large cache for modern laptop

Dynamic RAM = DRAM

DRAM stores bit as charge on capacitor:
- 1 transistor accesses stored charge
- requires periodic refresh = read-write (dynamic power)

SRAM stores bit on pair of inverting gates:
- several transistors
- requires continuous (static) power.
DRAM design
(one option)

Row decoder 11-to-2048

2048 x 2048 array

Column latches

Address[10–0]

Single set of address lines, time-shared for row address, column address.

Accesses entire row, stores in column latches. Mainly used for refreshing entire row at a time. Accessing other columns in same row again cheaper...

64-bit DRAM

3 to 8 row decoder

Column latches

3-bit address

Reading bit at address 101011
1. Select row

3 to 8 row decoder

Column latches

Mux

Data out

Reading bit at address 101011
2. Copy row to latches

3 to 8 row decoder

Row is fading!

Column latches

Mux

Data out
Reading bit at address 101011
3. Refresh row from latches

Reading bit at address 101011
4. Select column from latches