State is coded as the amount of energy stored by a device.

State is read by sensing the amount of energy

Problems: noise changes $Q$ (up or down), parasitics leak or source $Q$. Fortunately, $Q$ cannot change instantaneously, but that only gets us in the ballpark.
Last Time: Storing Bits Reliably

Store more energy than we expect from the noise.

\[ Q = CV. \] To store more charge, use a bigger \( V \) or make a bigger \( C \).

Cost: Power, chip size.

Represent state as charge in ways that are robust to noise.

Example: 1 bit per capacitor. Write 1.5 volts on \( C \). To read \( C \), measure \( V \).

\[ V > 0.75 \text{ volts is a "1".} \]
\[ V < 0.75 \text{ volts is a "0".} \]

Cost: Could have stored many bits on that capacitor.

Correct small state errors that are introduced by noise.

Ex: read \( C \) every 1 ms

Is \( V > 0.75 \) volts?
Write back 1.5V (yes) or 0V (no).

Cost: Complexity.
Last Time: 1-T DRAM cells

"Bit Line"  "Word Line"  Capacitor  Vdd

Word Line and Vdd run on "z-axis"

Why Vcap values start out at ground.

Diode leakage current.
Today: Memory Technology Wrap-Up

Static Memory Circuits: For SRAM memory cells and for flip-flops.

Memory Arrays: Row decoders, column sense amps, array sizing.

DRAM Interfaces: How the SDRAM chips on the Calinx board work.
Last Time: Model for "off" transistor...

\[ V_d = 1\, \text{V} \quad V_g = 0.2\, \text{V} \quad V_s = V_{\text{sub}} = 0\, \text{V} \]

\[ I_{\text{ds}} = I_0 \left[ \exp\left(\kappa V_g - V_s\right)/V_0\right] \left[1 - \exp\left(-V_d/V_0\right)\right] \]

- \( I_0 \approx 100\, \text{fA} \)
- \( V_0 = kT/q = 25\, \text{mV} \)
- \( \kappa = 0.7 \)

Current flows when electrons diffuse to the "gate wall" top

\# electrons that reach top goes up as wall comes down, implies \( I_{\text{ds}} \sim \exp(V_g) \)
Last Time: Transistor Off Current

\[ I_{ds} = I_{o} \left[ \exp\left(\frac{\kappa V_g - V_s}{V_o}\right) \right] \left[ 1 - \exp\left(-\frac{V_{ds}}{V_o}\right) \right] \]

\[ I_{off} = 10 \text{ nA} \]

1.2 mA = \(I_{on}\)

0.25 = \(V_t\)

0.7 = \(V_{dd}\)
Last Time: Model for “on” transistor ...

\[
I_{ds} = \frac{\text{carriers in channel}}{(\text{transit time})} \quad Q = CV \quad f(\text{length, velocity})
\]

\[
I_{ds} = \left[\frac{\mu \epsilon W}{(LD)}\right] [V_{gs} - V_{th}] [V_{ds}]
\]

If \( V_{ds} > V_{gs} - V_{th} \), channel physics change:

\[
I_{ds} = \left[\frac{\mu \epsilon W}{(2LD)}\right] [V_{gs} - V_{th}]^2
\]

\( W = \text{transistor width, } L = \text{length, } D = \text{capacitor plate distance} \)

\( \mu \) is velocity, \( \epsilon \) is dielectric constant
Inverters: Circuits and Layout

Vdd symbol

\[ V_{\text{in}} \rightarrow V_{\text{out}} \]

\[ V_{\text{in}} \]

\[ V_{\text{out}} \]
Inverter: Die Cross Section

V_{in} \quad V_{out} \quad V_{out} \quad V_{in}

\text{n}\text{-well} \quad \text{oxide} \quad \text{oxide}
Inverters: n-fet Transistor Equation

If $V_{gs} > V_t$ and $V_{ds} > V_{gs} - V_t$:

$$I_{ds} = \left(\frac{k}{2}\right) \left(\frac{W}{L}\right) [V_{gs} - V_t]^2$$

Otherwise, if $V_{gs} > V_t$:

$$I_{ds} = k \left(\frac{W}{L}\right) [V_{gs} - V_t] [V_{ds}]$$

Otherwise:

$$I_{ds} = 0, \text{ but really } = I_0 \left[\exp\left(\frac{V_{gs} - V_s}{V_o}\right)\right] \left[1 - \exp\left(-\frac{V_{ds}}{V_o}\right)\right]$$

Note: $V_t$ is transistor threshold, was formerly $V_{th}$. Also, $V_t$ is actually $V_t(V_s) \sim \sqrt{V_s}$. 
Inverters: p-fet Transistor Equation

\[
I_{sd} = \begin{cases} 
\frac{k}{2} \left( \frac{W}{L} \right) (V_{sg} - V_t)^2 & \text{If } V_{sg} > V_t \text{ and } V_{sd} > V_{sg} - V_t \\
k \left( \frac{W}{L} \right) (V_{sg} - V_t) V_{sd} & \text{Otherwise, if } V_{sg} > V_t \\
0, \text{ but again, in reality there is a “leakage” current.} & \text{Otherwise:}
\end{cases}
\]

Note: \( V_t \) for p-Fet and n-Fet are different. Also true for “k” (fab constant). \( k_p < k_n \), due to electrons being faster than holes.
Inverters with \( \text{Vin} = \text{Gnd} \), \( \text{Vout} = \text{Vdd} \)

\[
I_{sd} = k \left( \frac{W}{L} \right) \left[ V_{sg} - V_t \right] \left[ V_{sd} \right]
\]

This goes as close to 0 as it can while still supplying the leakage current.

\( I_{ds} \approx 0 \), but really a small leakage current

\( \text{Isd > Vsg - Vt \ once \ Vout \ is \ Vdd?} \)

\( \text{Is Vsg > Vt?} \)
Inverters with \( \text{Vin} = \text{Vdd}, \text{Vout} = \text{Gnd} \)

\[ I_{sd} \approx 0, \text{ but really a small leakage current} \]

\[ I_{sd} = k \left( \frac{W}{L} \right) (V_{gs} - V_t) [V_{ds}] \]

\( \text{This goes as close to 0 as it can while still supplying the leakage current.} \)

\( \text{Is} \ V_{ds} > V_{gs} - V_t \text{ once Vout is Gnd?} \)

\( \text{Is} \ V_{gs} > V_t ? \)
Calculating the inverter threshold (Vth)

Tie output to input. \( V_{th} \)

Assume voltage is "somewhere near the middle"

For nfet, is \( V_{ds} > V_{gs} - V_{t} \)?

For pfet, is \( V_{sd} > V_{sg} - V_{t} \)  ?

No, by definition! Use:

\[
I_{ds} = k_n (W/L) \left[ V_{th} - V_{tn} \right] \left[ V_{th} \right]
\]

\[
I_{sd} = k_p (W/L) \left[ V_{dd} - V_{th} - V_{tp} \right] \left[ V_{dd} - V_{th} \right]
\]

To compute the exact "voltage in the middle".
Question: What happens when ...

Stays at Vth until a tiny amount of Vin noise appears.

Then output goes to Vdd or Gnd until ...

... Vin noise flips it back the other way.

Lesson: at Vth, small dVin make big dVout
Dynamic Memory: Circuit remembers for a fraction of a second.

Static Memory: Circuit remembers as long as the power is on.

Non-volatile Memory: Circuit remembers for many years, even if power is off.
Recall DRAM cell: 1 T + 1 C

- Bit Line
- "Column"
- "Word Line"
- "Row"
- Vdd
- "Column"
- Word Line
- "Bit Line"
Idea: Store each bit with its complement

We can use the redundant representation to compensate for noise and leakage.
Case #1: y = Gnd, \( \overline{y} = \text{Vdd} \ldots \)
Case #2: \( y = Vdd, \overline{y} = Gnd \) ...
Combine both cases to complete circuit

"Cross-coupled inverters"
SRAM Challenge #1: It’s so big!

SRAM area is 6X-10X DRAM area, same generation ...

Capacitors are usually “parasitic” capacitance of wires and transistors.

Cell has both transistor types

Vdd AND Gnd

More contacts, more devices, two bit lines ...
Challenge #2: Writing is a “fight”

When word line goes high, bitlines “fight” with cell inverters to “flip the bit” -- must win quickly!

Solution: tune W/L of cell & driver transistors

Bitline drives Gnd

Initial state Vdd

Initial state Gnd

Bitline drives Vdd

Initial state Vdd
Challenge #3: Preserving state on read

When word line goes high on read, cell inverters must drive large bitline capacitance quickly, to preserve state on its small cell capacitances.
SRAM vs DRAM, pros and cons

**DRAM** has a 6-10X density advantage at the same technology generation.

**SRAM advantages**

- **SRAM** has deterministic latency: its cells do not need to be refreshed.
- **SRAM** is much faster: transistors drive bitlines on reads.
- **SRAM** easy to design in logic fabrication process (and premium logic processes have SRAM add-ons).

Big win for DRAM
Flip Flops Revisited
Recall: Static RAM cell (6 Transistors)

"Cross-coupled inverters"
Recall: Positive edge-triggered flip-flop

A flip-flop “samples” right before the edge, and then “holds” value.

Sampling circuit

Holds value

16 Transistors: Makes an SRAM look compact!

What do we get for the 10 extra transistors?

Clocked logic semantics.
Sensing: When clock is low

A flip-flop “samples” right before the edge, and then “holds” value.

**Sampling circuit**

**Holds value**

\[
\begin{align*}
\text{clk} &= 0 \\
\text{clk}’ &= 1
\end{align*}
\]

Will capture new value on posedge.

Outputs last value captured.
Capture: When clock goes high

A flip-flop “samples” right before the edge, and then “holds” value.

D to Q
Sampling circuit
Holds value

clock = 1

clock' = 0

Remembers value just captured.

Outputs value just captured.
Admin: Final Xilinx Checkoff Friday ...

<table>
<thead>
<tr>
<th>Date</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>F 10/13</td>
<td>Lab 3: Final Xilinx Checkoff, 12-2PM or 3-5PM, 125 Cory</td>
</tr>
<tr>
<td>M 10/16</td>
<td>Lab 3: Final Report Due, 11:59 PM via the submit program</td>
</tr>
</tbody>
</table>

4. **On Friday 10/14** in lab section, you will demonstrate the processor running on the Calinx board. Unlike the 10/7 checkoff, this demo will test all facets of the processor (including the issues discussed in Problem 4). During this demo, the TA will provide you with secret test code. If you are able to pass these tests on your first try, you will receive bonus points. You can also receive bonus points if you fix your processor to pass the tests within your section time. If your processor is not fixed by the end of section, your TA will provide source for the secret test code, for use in your weekend debugging sessions.

Lab report due **Monday, 11:59 PM.**
Memory Arrays

Calinx DRAM: 133 Mhz, 128 Mb

128Mb: x4, x8, x16 SDRAM

SYNCHRONOUS DRAM

MT48LC32M4A2 – 8 Meg x 4  x 4 banks
MT48LC16M8A2 – 4 Meg x 8  x 4 banks
MT48LC8M16A2 – 2 Meg x 16 x 4 banks

For the latest data sheet, please refer to the Micron Web site: www.micron.com/dramds

Data sheet on “resources” page. Will need to understand for final project!
Last Time: 1-T DRAM cell

Word Line and Vdd run on "z-axis"

Why Vcap values start out at ground.
Last Time: DRAM Read is Destructive

Raising the word line removes the charge from every cell it connects too!

Must write back after each read.
Last Time: DRAM Refresh...

Parasitic currents leak away charge.

Solution: “Refresh”, by reading cells at regular intervals (tens of milliseconds).
People buy DRAM for the bits. “Edge” circuits are overhead.

So, we amortize the edge circuits over big arrays.
A “bank” of 32 Mb (128Mb chip -> 4 banks)

12-bit row address input

1 of 4096 decoder

4096 rows (tester found good bits in bigger array)

33,554,432 usable bits

2048 columns

2048 bits delivered by sense amps

Select requested bits, send off the chip
Recall DRAM Challenge #3b: Sensing

How do we reliably sense a 60mV signal? Compare the word line against the voltage on a "dummy" world line.

"Dummy" word line. Cells hold no charge.
Corresponds to row read into sense amps

12-bit row address input

1 of 4096 decoder

4096 rows

33,554,432 usable bits (tester found good bits in bigger array)

2048 columns

2048 bits delivered by sense amps

Select requested bits, send off the chip

Slow! This 7.5 ns period DRAM (133 MHz) can do row reads at only 75 ns (13 MHz). Plus, need to add selection time.

DRAM has high latency to first bit out. A fact of life.
An ill-timed refresh may add to latency

Parasitic currents leak away charge.

Solution: “Refresh”, by reading cells at regular intervals (tens of milliseconds)

Diode leakage...
Latency is not the same as bandwidth!

Thus, push to faster DRAM interfaces

What if we want all of the 2048 bits?
In row access time (75 ns) we can do 10 transfers at 133 MHz.
8-bit chip bus -> 10 x 8 = 80 bits << 2048
Now the row access time looks fast!

4096 rows
33,554,432 usable bits (tester found good bits in bigger array)

2048 columns

2048 bits delivered by sense amps
Select requested bits, send off the chip
Sadly, it’s rarely this good ...

What if we want all of the 2048 bits? The “we” for a CPU would be the program running on the CPU.

Recall Amdal’s law: If 20% of the memory accesses need a new row access ... not good.

33,554,432 usable bits (tester found good bits in bigger array)

2048 bits delivered by sense amps

Select requested bits, send off the chip
DRAM latency/bandwidth chip features

Columns: Design the right interface for CPUs to request the subset of a column of data it wishes:

- 2048 bits delivered by sense amps
- Select requested bits, send off the chip

Interleaving: Design the right interface to the 4 memory banks on the chip, so several row requests run in parallel.
Off-chip interface for the Micron part ... 

A clocked bus protocol (133 MHz)

From Micron 128 Mb SDRAM data sheet (on “resources” web page)

DRAM is controlled via commands (READ, WRITE, REFRESH, ...)

Synchronous data output.

Note! This example is best-case!
To access a new row, a slow ACTIVE command must run before the READ.
Opening a row before reading ...

70 ns between row opens

44 ns
Interleave: Access all 4 banks in parallel

NOTE: Each READ command may be to any bank. DQM is LOW.

Figure 8
Random READ Accesses
Essential tools for the final project.

<table>
<thead>
<tr>
<th>Day</th>
<th>Date</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Th</td>
<td>10/12</td>
<td>Cache I</td>
</tr>
<tr>
<td>F</td>
<td>10/13</td>
<td></td>
</tr>
<tr>
<td>Sa</td>
<td>10/14</td>
<td></td>
</tr>
<tr>
<td>Su</td>
<td>10/15</td>
<td></td>
</tr>
<tr>
<td>M</td>
<td>10/16</td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>10/17</td>
<td>Cache II</td>
</tr>
</tbody>
</table>