CMOS Technology
The Impact of Device Technology

1943
ENIAC
30 tons, 200KW
~1000 ops/s

Built using
18,000 vacuum tubes
taking 170m²

2023
Macbook Pro
~1kg, 10W
~1 trillion ops/s

Built using
68 billion transistors
packed in a 500mm² chip

Rapid progress in device technology has been the main driver of computer performance
80+ Years of Exponential Improvement!


1947
First transistor
(Bardeen, Brattain, Shockley)

1958
First integrated circuit
(Kilby/Noyce)

1965
Moore’s “Law”
the number of transistors in an integrated circuit doubles every year two years
Moore’s Law

- But Moore’s Law is waning; reaching physical limits!
- Future performance gains must come from systems
A Deep Dive Into a Chip

Packaged chip

Silicon die (100-400mm²)

Transistor (FET)

32 nm

Die cross-section

6-15 metal layers (wires)

Source: Intel

February 23, 2023

MIT 6.191 Spring 2023
Field-Effect Transistors (FETs)

- Nearly all digital systems are built using field-effect transistors, which are voltage-controlled switches.

- FETs come in two varieties; we’ll first see nFETs.

nFET

G (gate)

diffusion terminals

A high voltage at gate (G=1) creates a conducting path between diffusion terminals.
By convention, we label diffusion terminals source (S) or drain (D) depending on their voltages.

- On nFETs, source = diffusion terminal at lower voltage.
- This convention lets us define the behavior of FETs using the voltage between gate and source.

- nFET has a threshold voltage $V_{TH}$.
- nFET is ON if the voltage between gate and source $V_{GS}$ exceeds $V_{TH}$, OFF otherwise.
- Very simplified model, but sufficient to build logic gates.
What Does This Circuit Compute?

NMOS inverter

Assume $V_{TH} \approx V_{DD}/2$

<table>
<thead>
<tr>
<th>IN</th>
<th>OUT</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
What Does This Circuit Compute?

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

NMOS NAND gate
NMOS Logic

- NMOS gates have
  - A pulldown network of nFETs that create a conducting path between ground and \( V_{OUT} \) for some combinations of inputs
  - A pullup resistor that keeps \( V_{OUT} \approx V_{DD} \) when the pulldown network is OFF

- A note on terminology:
  - MOSFETs (metal-oxide-semiconductor field-effect transistors) are a common type of FET
  - nFET was sometimes abbreviated as nMOS; NMOS logic got its name from this abbreviation
A Questionable Gate

- \( V_{\text{IN}} < V_{\text{TH}} \rightarrow \) nFET OFF, \( V_{\text{OUT}} = 0V \)
- \( V_{\text{IN}} > V_{\text{TH}} \rightarrow \) nFET only on until \( V_{\text{OUT}} = V_{\text{IN}} - V_{\text{TH}} \) !
- Gain at most 1, not a valid digital device
- Corollary: nFETs can pull down, but not pull up
MOSFET Physical Structure
Need a bit more info to reason about area, delay, power
With $V_{GS} < V_{TH}$, almost no current flows between source and drain.

As $V_{GS}$ reaches $V_{TH}$, a channel forms between source and drain.

The shape of the channel (and its resistance) also depends on the voltage at the drain. But a low-resistance channel will exist while $V_{GS} > V_{TH}$. 

February 23, 2023

MIT 6.191 Spring 2023
FET First-Order Electrical Model

- Simplest possible model that lets us reason about delay, area, and power. Not very accurate!

\[ \begin{align*}
R_{\text{channel}} &= \begin{cases} 
R_{\text{OFF}} & \text{if } V_{GS} < V_{\text{TH}} \\
R_{\text{ON}} & \text{if } V_{GS} \geq V_{\text{TH}}
\end{cases} \\
R_{\text{ON}} &<< R_{\text{OFF}}
\end{align*} \]
The Problem with NMOS Logic

- When nFET is OFF, \( R_{\text{pullup}} \ll R_{\text{channel}} = R_{\text{OFF}} \) pulls up \( V_{\text{OUT}} \) by charging \( C_{\text{gate}} \).
- When nFET is ON, \( R_{\text{channel}} = R_{\text{ON}} \ll R_{\text{pullup}} \) pulls down \( V_{\text{OUT}} \) by discharging \( C_{\text{gate}} \)...
- ... and high current flows from \( V_{\text{DD}} \) to ground!!
CMOS (Complementary MOS) Logic

The key idea

- NMOS logic consumes a lot of energy due to $R_{pullup}$
  - When nFET is ON, $I = V_{DD}/(R_{pullup} + R_{ON}) \approx V_{DD}/R_{pullup}$
  - Can’t make $R_{pullup}$ very large (gate becomes slow)

- Key idea: What if we had a variable $R_{pullup}$?
  - When pulldown network is ON, make $R_{pullup}$ very large (to avoid current and energy waste)
  - When pulldown network is OFF, make $R_{pullup}$ very small (to pull up $V_{OUT}$ quickly)

- CMOS gates have complementary pulldown and pullup FET networks: pulldown is ON when pullup is OFF, and vice versa
  - We can build the pulldown network with nFETs, but what about the pullup?
pFET: nFET’s Evil Twin

- Switching model:
  pFET is ON if the voltage between source and gate $V_{SG}$ exceeds $V_{TH}$, OFF otherwise.

A low voltage at gate (G=0) creates a conducting path between source and drain.

- $V_{SG} < V_{TH}$ OFF
  - $V_{SG} > V_{TH}$ ON

DIAGRAM:

- Source (S) with higher voltage
- Drain (D) with lower voltage
- Gate (G)
What Does This Circuit Compute?

CMOS inverter

Assume $V_{TH} \approx V_{DD}/2$

$V_{IN} < V_{TH}$  $V_{IN} > V_{DD} - V_{TH}$

<table>
<thead>
<tr>
<th>IN</th>
<th>OUT</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
What Does This Circuit Compute?

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

CMOS NAND gate
## CMOS Logic

- CMOS gates have **complementary** pullup and pulldown networks, i.e., the pullup is on where the pulldown is off and vice versa.

<table>
<thead>
<tr>
<th>pullup</th>
<th>pulldown</th>
<th>F(inputs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>on</td>
<td>off</td>
<td>driven “1”</td>
</tr>
<tr>
<td>off</td>
<td>on</td>
<td>driven “0”</td>
</tr>
<tr>
<td>on</td>
<td>on</td>
<td>driven “X”</td>
</tr>
<tr>
<td>off</td>
<td>off</td>
<td>no connection</td>
</tr>
</tbody>
</table>

- CMOS uses pFETs to implement the pullup network and nFETs to implement the pulldown network.
More Questionable Gates

- What can go wrong with the following gates?

- CMOS Rule #1: Complementary pullup and pulldown networks
- CMOS Rule #2: pFETs in pullup, nFETs in pulldown

A = 0 B = 1 or A = 1 B = 0
connect supply and ground

pFET doesn’t pull down
V_{OUT} below V_{TH}
nFET doesn’t pull up
V_{OUT} above V_{DD} - V_{TH}
CMOS Complements

- Conducts when A is high
- Conducts when A is low: \( \overline{A} \)
- Conducts when A is high and B is high: \( A \cdot B \)
- Conducts when A is low or B is low: \( \overline{A} + \overline{B} = \overline{A \cdot B} \)
- Conducts when A is high or B is high: \( A + B \)
- Conducts when A is low and B is low: \( \overline{A} \cdot \overline{B} = \overline{A + B} \)
General CMOS Gate Recipe

Step 1. Derive the pullup network that does what you want, e.g.,

\[ F = \overline{A} + \overline{B} \cdot \overline{C} \]

(Determine what combination of inputs generates a high output)

Step 2. Derive complementary pulldown network: replace pFETs with nFETs, series subnets with parallel subnets, and parallel subnets with series subnets

Step 3. Combine pFET pullup network from Step 1 with nFET pulldown network from Step 2 to form the CMOS gate.

*Can a single CMOS gate implement any arbitrary functions?*  
**No**
CMOS Gates are Inverting

- In a CMOS gate, rising inputs (0→1) lead to falling outputs (1→0) and vice versa

- On a rising input,
  - nFETs go OFF→ON, so pulldown may connect output to ground
  - pFETs go ON→OFF, so pullup may disconnect output from $V_{DD}$
  - Output either stays the same or falls

- Corollary: Cannot build non-inverting logic using a single CMOS gate
  - Example: AND

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>A·B</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

rising input   rising output
Analyzing the Delay, Area, and Power of CMOS Gates

NOTE: Demystification, will not be on the quiz
**FET First-Order Electrical Model**
Including pFETs

- Simplest possible model that lets us reason about delay, area, and power. Not very accurate!
CMOS Gate Delay

Consider the following circuit. Given $V_{IN}(t)$, can you derive $V_{OUT}(t)$?

For $t > 0$, $V_{OUT}(t) = V_{OUT}(0)e^{-t/RC}$.
Propagation Delay

Propagation delay ($t_{PD}$): Upper bound on the delay from valid inputs to valid outputs.

$$V_{IN} \leq t_{PD} \leq V_{OUT}$$

To minimize $t_{PD}$, must keep resistances and capacitances low.
MOSFET Sizing

- CMOS gates use MOSFETs with smallest possible $L$ and choose $W$ to set performance
  - Wider FETs drive more current (lower $R$), but their gates are harder to drive (higher $C$) and they take more area

\[ R_{\text{channel}} \propto \frac{L}{W} \]
\[ C_{\text{gate}} \propto L \cdot W \]

How do $C_{\text{gate}}$ and $R_{\text{channel}}$ change with $L$ and $W$?
A standard cell library provides implementations of common gates (NAND, NOR, XOR, etc.) for a specific implementation technology.

Each gate includes:
- Electrical parameters (e.g., Rs and Cs)
- Physical layout

Synthesis tools use gates from the standard library instead of sizing and placing individual transistors.
Wide (High-Fanin) Gates

Most standard cell libraries include 2-, 3- and 4-input devices:

But for a large number of inputs, the series connections of too many MOSFETs can lead to very large effective $R_{\text{pulldown}}$ or $R_{\text{pullup}}$. Instead, use trees of smaller devices...

Example: 8-input NAND

How does $t_{PD}$ grow with the number of inputs $N$?

If we use a single CMOS gate, $t_{PD} \propto N$

If we use a tree of gates, $t_{PD} \propto \log(N)$
CMOS Power Dissipation

- Total power dissipation: $P = P_{\text{dynamic}} + P_{\text{static}}$

- Dynamic power: Caused by $0 \leftrightarrow 1$ transitions of nodes in the circuit
  - Charging/discharging each capacitor consumes $\frac{1}{2}CV_{\text{DD}}^2$ energy
  - If on average $C_S$ capacitance across the chip switches each cycle, and there are $f_{\text{CLK}}$ cycles per second
    $$P_{\text{dynamic}} = \frac{1}{2}C_SV_{\text{DD}}^2f_{\text{CLK}}$$

- Static power: Caused by
  - Subthreshold leakage: Even when the FET is off, a very small current flows from source to drain ($R_{\text{OFF}} < \infty$)
  - Tunneling current: Gate and channel are separated by a very thin (<1nm) dielectric, so some electrons tunnel through
    $$P_{\text{static}} = I_{\text{static}}V_{\text{DD}}$$
  - Static power is typically 10-30% of total power
Summary

- FETs behave as voltage-controlled switches

- NMOS gates are simple, but burn too much power

- CMOS gates:
  - Use complementary pullup and pulldown networks
  - Use pFETs in pullup, nFETs in pulldown network

- CMOS gates are inverting (rising inputs can only cause falling outputs, and vice versa)
Thank you!

Next lecture:
Sequential Circuits