3. CMOS Technology

6.004x Computation Structures
Part 1 – Digital Circuits

Copyright © 2015 MIT EECS
Combinational Device Wish List

- Design our system to tolerate some amount of error
  ⇒ Add positive noise margins
  ⇒ VTC: gain > 1 & nonlinearity
- Lots of gain ⇒ big noise margin
- Cheap, small
- Changing voltages will require us to dissipate power, but if no voltages are changing, we’d like zero power dissipation
- Want to build devices with useful functionality (what sort of operations do we want to perform?)
MOSFETs (metal-oxide-semiconductor field-effect transistors) are four-terminal voltage-controlled switches. Current flows between the diffusion terminals if the voltage on the gate terminal is large enough to create a conducting “channel”, otherwise the mosfet is off and the diffusion terminals are not connected.

\[ I_{DS} \propto \frac{W}{L} \]
N-Channel MOSFET: Electrical View

The four terminals of a Field Effect Transistor (gate, source, drain and bulk) connect to conductors that generate a set of electric fields in the channel region which depend on the relative voltages of each terminal.

![Diagram of N-Channel MOSFET]

Want \( V_p \leq V_N \)

Linear operating region (ohmic mode)

Saturation mode at point of pinch-off

Saturation mode

Olivier Delege and Peter Scott (CC BY-SA 3.0)
N-channel MOSFET $I_{DS}$ vs. $V_{DS}$

Ohmic: $I_{DS} = V_{DS}/R$

$saturated$

Increasing $V_{GS}$

$V_{TH} = 0.5V$
FETs Come in Two Flavors

NFET: n-type source/drain diffusions in a p-type substrate. Positive threshold voltage; inversion forms n-type channel.

Connect B to GND to keep PN reverse-biased ($V_p \leq V_n$); keeps D and S insulated from B

PFET: p-type source/drain diffusions in a n-type substrate. Negative threshold voltage; inversion forms p-type channel.

Connect B to VDD to keep PN reverse-biased

The use of both NFETs and PFETs – complimentary transistor types – is a key to CMOS (complementary MOS) logic families.
CMOS Recipe

If we follow two rules when constructing CMOS circuits, we can model the behavior of the mosfets as simple *voltage-controlled switches*:

**Rule #1:** only use NFETs in pulldown circuits

**Rule #2:** only use PFETs in pullup circuits

NFET Operating regions:

- **“off”:** $V_{GS} < V_{TH,NFET}$
  - $V_{GS} \uparrow \Rightarrow \text{“R”}\downarrow$
  - NFET threshold = ~0.5V

- **“on”:** $V_{GS} > V_{TH,NFET}$
  - $V_{GS} \downarrow \Rightarrow \text{““}\uparrow$

PFET Operating regions:

- **“off”:** $V_{GS} > V_{TH,PFET}$
  - $V_{GS} \uparrow \Rightarrow \text{““}\downarrow$

- **“on”:** $V_{GS} < V_{TH,PFET}$
  - $V_{GS} \downarrow \Rightarrow \text{““}\uparrow$

PFET threshold = ~ −0.5V
When \( V_{IN} \) is low, the nfet is off and the pfet is on, so current flows into the output node and \( V_{OUT} \) eventually reaches \( V_{DD} (> V_{OH}) \) at which point no more current will flow.

When \( V_{IN} \) is high, the pfet is off and the nfet is on, so current flows out of the output node and \( V_{OUT} \) eventually reaches GND (< \( V_{OL} \)) at which point no more current will flow.

When \( V_{IN} \) is in the middle, both the pfet and nfet are “on” and the shape of the VTC depends on the details of the devices’ characteristics. CMOS gates have very high gain in this region (small changes in \( V_{IN} \) produce large changes in \( V_{OUT} \)) and the VTC is almost a step function.
Beyond Inverters:
Complementary pullups and pulldowns

We want *complementary* pullup and pulldown logic, i.e., the pulldown should be “on” when the pullup 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>

Now you know what the “C” in CMOS stands for!

Since there’s plenty of capacitance on the output node, when the output becomes disconnected it “remembers” its previous voltage -- at least for a while. The “memory” is the load capacitor’s charge. Leakage currents will cause eventual decay of the charge (that’s why DRAMs need to be refreshed!).
CMOS Complements

What a nice $V_{OH}$ you have...

Thanks. It runs in the family...

- A conducts when A is high
- $\bar{A}$ conducts when A is low
- B conducts when A is high and B is high: $A \cdot B$
- B conducts when A is low or B is low: $A + B = A \cdot B$
- $\bar{A}$ conducts when A is high or B is high
- $A \cdot \bar{B} = A + B$
A Pop Quiz!

What function does this gate 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>

NAND

COST for an older 45nm process:
- $3500 per 300mm wafer
- 300mm round wafer = $\pi (150e^{-3})^2 = .07m^2$
- NAND gate = $(82)(16)(45e^{-9})^2 = 2.66e^{-12}m^2$
- $2.6e^{10}$ NAND gates/wafer (= 100 billion FETS!)
- marginal cost of NAND gate: $132n\$
General CMOS Gate Recipe

Step 1. Figure out the pullup network that does what you want, \textit{e.g.,}

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

(Determine what combination of inputs generates a high output)

Step 2. Walk the hierarchy replacing \textit{nfets} with \textit{pfets}, series subnets with parallel subnets, and parallel subnets with series subnets

Step 3. Combine \textit{pfet} pullup network from Step 1 with \textit{nfet} pulldown network from Step 2 to form a fully-complementary CMOS gate.

Does this recipe work for all logic functions?
CMOS Gates Are Naturally Inverting

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

- NFETs go from “off” to “on”
  → pulldown paths connected
  → output may be connected to ground

- PFETs go from “on” to “off”
  → pullup paths disconnected
  → output may be disconnected from V_{DD}

Corollary: you can’t build positive logic, e.g., AND, with one CMOS gate

\[
\begin{array}{c|c|c}
A & B & A \cdot B \\
0 & 0 & 0 \\
0 & 1 & 0 \\
1 & 0 & 0 \\
1 & 1 & 1 \\
\end{array}
\]

For CMOS gate:
All inputs 0
→ nfets off, pfets on
→ output must be 1
All inputs 1
→ nfets on, pfets off
→ output must be 0

Oops, output is also rising!
A=1, B rising...
CMOS Timing Specifications

Circuit:

Electrical model:

Waveforms:

\[ \tau = R_{PD} \cdot C_L \]

\[ \tau = R_{PU} \cdot C_L \]
Propagación de la demora (\(t_{PD}\)): Un LÍMITE SUPERIOR en la demora desde inputs válidos a outputs válidos.

**GOAL:**
minimizar propagación de la demora!

**ISSUE:**
mantén capacitancias bajas y transistores rápidos.
Contamination Delay

Contamination delay ($t_{CD}$): A LOWER BOUND on the delay from any invalid input to an invalid output.

Do we really need $t_{CD}$?
Usually not... it’ll be important when we design circuits with registers (coming soon!)

If $t_{CD}$ is not specified, safe to assume it’s 0.
The Combinational Contract

\[ A \rightarrow B \]

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

\( t_{PD} \) propagation delay
\( t_{CD} \) contamination delay

Notes:
1. *No Promises* during \( \text{xxxx} \)
2. Default (conservative) spec: \( t_{CD} = 0 \)
Acyclic Combinational Circuits

If NAND gates have a $t_{PD} = 4\text{nS}$ and $t_{CD} = 1\text{nS}$

$t_{CD}$ is the *minimum* cumulative contamination delay over all paths from inputs to outputs

$t_{PD}$ is the *maximum* cumulative propagation delay over all paths from inputs to outputs

$t_{PD} = \underline{12} \text{nS}$

$t_{CD} = \underline{2} \text{nS}$
One Last Timing Issue...

Recall the rules for *combinational devices*:

Output guaranteed to be valid when *all* inputs have been valid for at least $t_{PD}$, and, outputs may become invalid no earlier than $t_{CD}$ after an input changes!

Many gate implementations—e.g., CMOS—adhere to even tighter restrictions.
What Happens In This Case?

CMOS NOR: B

A

Z

NOR:

<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>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Lenient NOR:

<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>X</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>X</td>
<td>0</td>
</tr>
</tbody>
</table>

LENIENT Combinational Device:

Output guaranteed to be valid when any combination of inputs sufficient to determine the output value has been valid for at least \( t_{PD} \). Tolerates transitions -- and invalid levels -- on irrelevant inputs!

Input A=1 is sufficient to determine the output.
Summary

• CMOS
  • Only use NFETs in pulldowns, PFETs in pullups → mosfets behave as voltage-controlled switches
  • Series/parallel Pullup and pulldown switch circuits are complementary
  • CMOS gates are naturally inverting (rising input transition can only cause falling output transition, and vice versa).
  • “Perfect” VTC (high gain, $V_{OH} = V_{DD}$, $V_{OL} = GND$) means large noise margins and no static power dissipation.

• Timing specs
  • $t_{PD}$: upper bound on time from valid inputs to valid outputs
  • $t_{CD}$: lower bound on time from invalid inputs to invalid outputs
  • If not specified, assume $t_{CD} = 0$
  • Lenient gates: output unaffected by some input transitions

• Next time: logic simplification, other canonical forms