ECE 2300
Digital Logic & Computer Organization
Spring 2021

CMOS Logic
Announcements

- Lab sessions start next week

- Prelims will be on zoom
  - More instructions to come in early March
**NAND Logic Gate**

Using De Morgan’s Law: \((X \cdot Y)' = X' + Y'\)

We can build circuits from NAND only!
We Can Build Circuits from NAND Only!

• NOT

• AND

• OR
Sum-of-Products Revisited

AND-OR

NAND-NAND
**NOR Logic Gate**

Using De Morgan’s Law: \((X+Y)’ = X’\cdot Y’\)

We can build circuits from NOR only!
Product-of-Sums Revisited

OR-AND

NOR-NOR
A Little Bit of History

- **Transistors**
  - Invented by John Bardeen, Walter Brattain, and William Shockley at Bell Labs in 1947

- **Integrated circuits (IC)**
  - Independently developed by Jack Kilby (at TI) and Robert Noyce (at Fairchild) in the 1950s
  - Noyce and Gordon Moore founded Intel in 1968


<table>
<thead>
<tr>
<th>PROCESSOR</th>
<th>INTRO DATE</th>
<th>PROCESS</th>
<th>TRANSISTORS</th>
<th>FREQUENCY</th>
</tr>
</thead>
<tbody>
<tr>
<td>4004</td>
<td>1971</td>
<td>10 μm</td>
<td>2,300</td>
<td>108 KHz</td>
</tr>
<tr>
<td>8080</td>
<td>1974</td>
<td>6 μm</td>
<td>6,000</td>
<td>2 MHz</td>
</tr>
<tr>
<td>8086</td>
<td>1978</td>
<td>3 μm</td>
<td>29,000</td>
<td>10 MHz</td>
</tr>
<tr>
<td>80286</td>
<td>1982</td>
<td>1.5 μm</td>
<td>134,000</td>
<td>12 MHz</td>
</tr>
<tr>
<td>80386</td>
<td>1985</td>
<td>1.5 μm</td>
<td>275,000</td>
<td>16 MHz</td>
</tr>
<tr>
<td>Intel 486 DX</td>
<td>1989</td>
<td>1 μm</td>
<td>1.2 M</td>
<td>33 MHz</td>
</tr>
<tr>
<td>Pentium</td>
<td>1993</td>
<td>0.8 μm</td>
<td>3.1 M</td>
<td>60 MHz</td>
</tr>
</tbody>
</table>

MOS Transistors

- **Metal-Oxide Semiconductor Field-Effect Transistors (MOSFETs)**
  - MOS for short

  A 3-terminal device controlled by the **gate voltage** that acts like a **switch**

- **Extreme changes in resistance (0 to \( \infty \)) make transistors act like switches**

Lecture 4: 9
NOT Gate Input & Output Voltages

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

<table>
<thead>
<tr>
<th>A</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>L</td>
<td>H</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>A</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0V</td>
<td>5V</td>
</tr>
<tr>
<td>5V</td>
<td>0V</td>
</tr>
</tbody>
</table>

- When the input voltage is *low*, the output should be connected to the voltage supply (e.g., $V_{DD}$, $V_{CC}$)
- When the input voltage is *high*, the output should be connected to ground (i.e., GND)
Lecture 4

NOT Using Switches

- Can build a NOT using two types of complementary switches
  - Type 1: Closed when input = 0, open when input = 1
  - Type 2: Closed when input = 1, open when input = 0

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

<table>
<thead>
<tr>
<th>A</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>L</td>
<td>H</td>
</tr>
<tr>
<td>H</td>
<td>L</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>A</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0V</td>
<td>5V</td>
</tr>
<tr>
<td>5V</td>
<td>0V</td>
</tr>
</tbody>
</table>
MOS Transistors

- Current flows when ON (conducting)
- No current flows when OFF (not conducting)
- Type 1 and Type 2 switches

Type 1: PMOS or p-channel
Type 2: NMOS or n-channel

G: Gate; S: Source; D: Drain
Bubble: LOW closes the switch
MOS Transistors

- **PMOS**
  - Closed when input is low \[1\]
  - Open when input is high
  - Passes a good one (but a poor zero) \[2\]

- **NMOS**
  - Closed when input is high \[1\]
  - Open when input is low
  - Passes a good zero (but a poor one) \[2\]

\[1\] In both cases, the voltage difference between the gate and source must exceed certain threshold voltage before the transistor starts having any effect
\[2\] Optional reading: [vlsimsee.blogspot.com/2013/05/why-cant-nmos-pass-1-and-pmos-pass-0.html](http://vlsimsee.blogspot.com/2013/05/why-cant-nmos-pass-1-and-pmos-pass-0.html)
CMOS Logic Gates

• Complementary MOS (CMOS)
  – CMOS dominates the digital IC market

• Uses both NMOS and PMOS devices such that there is no direct supply-ground path
  – Dissipates little power when the inputs don’t change

• Our focus: Static CMOS gates
  – Other types exist as well (pseudo-NMOS, domino, ...)

\[ \text{Diagram showing CMOS gate with inputs, PMOS, NMOS, V}_{\text{DD}}, \text{and GND}} \]
CMOS Inverter

- A is high: Q1 is on, Q2 is off, Z is low
- A is low: Q1 is off, Q2 is on, Z is high
CMOS NAND Gate

The CMOS NAND gate is shown in the diagram with transistors Q1, Q2, Q3, and Q4. The gate operates based on the logic levels A and B, and the output Z.

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>Q1</th>
<th>Q2</th>
<th>Q3</th>
<th>Q4</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>L</td>
<td>L</td>
<td>off</td>
<td>on</td>
<td>off</td>
<td>on</td>
<td>H</td>
</tr>
<tr>
<td>L</td>
<td>H</td>
<td>off</td>
<td>on</td>
<td>on</td>
<td>off</td>
<td>H</td>
</tr>
<tr>
<td>H</td>
<td>L</td>
<td>on</td>
<td>off</td>
<td>off</td>
<td>on</td>
<td>H</td>
</tr>
<tr>
<td>H</td>
<td>H</td>
<td>on</td>
<td>off</td>
<td>on</td>
<td>off</td>
<td>L</td>
</tr>
</tbody>
</table>
An $n$-input NAND uses $2n$ transistors
A “Mystery” Gate

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>Q1</th>
<th>Q2</th>
<th>Q3</th>
<th>Q4</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>L</td>
<td>L</td>
<td>L</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>L</td>
<td>H</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H</td>
<td>L</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H</td>
<td>H</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[ V_{DD} \]

A: 1
B: 0
Q1: 0
Q2: 1
Q3: 1
Q4: 0
Z: 1

Lecture 4: 18
Structure of Transistor Networks

- Two complementary networks:
  - A pull-up network composed of PMOS, with sources tied to voltage supply
  - A pull-down network composed of NMOS, with sources tied to ground
  - Equal number of NMOS and PMOS transistors
Structure of Transistor Networks

• The pull-up and pull-down networks are always duals

• To construct the dual of a network:
  – Exchange NMOS for PMOS (and vice versa)
  – Exchange series subnets for parallel subnets (and vice versa)
    • This transformation applies to hierarchical structures
Series and Parallel Subnets

• **Series**
  – *All* inputs must be 1 (n-type) or 0 (p-type) to make the connection

• **Parallel**
  – *At least one* input must be 1 (n-type) or 0 (p-type) to make the connection
Duality of Parallel/Series Subnets

n-type series subnet
F pulls down to 0 when A and B are high =>
F = (A ∙ B)’

p-type parallel subnet
F pulls up to 1 when A or B is low =>
F = A’+B’ = (A ∙ B)’

n-type parallel subnet
F pulls down to 0 when A or B is high =>
F = (A+B)’

p-type series subnet
F pulls up to 1 when A and B are low =>
F = A’B’ = (A+B)’
Analysis of Transistor Networks

• Transistor states
  – Determine all possible input combinations
  – Figure out the state of each transistor
  – Determine final output

• or by inspection
  – Figure out what input combinations cause a 1 (or a 0) output
Analysis of Transistor Networks

- Build the truth table

<table>
<thead>
<tr>
<th>INPUTS</th>
<th>TRANSISTORS</th>
<th>OUTPUT</th>
</tr>
</thead>
<tbody>
<tr>
<td>A B C</td>
<td>Q1 Q2 Q3 Q4 Q5 Q6</td>
<td>Z</td>
</tr>
<tr>
<td>0 0 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0 0 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0 1 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0 1 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 0 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 0 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 1 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 1 1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Analysis of Transistor Networks

• By inspection
  – Inspect either pull-up (PMOS) or pull-down (NMOS) network
  – Translate the series (parallel) subnets into product (sum) terms
  – For pull-down network, negate the combined expression

Pull-up: \((A' + B')C'\)
Pull-down: \((A \cdot B + C)'\)
Recipe for Constructing CMOS Gate

F = (A(B+C))'

**Step 1.** Figure out pull-down network that does what you want (e.g., what combination of inputs generates a low output)

**Step 2.** Walk the hierarchy replacing NMOS with PMOS, series subnets with parallel subnets, and parallel subnets with series subnets

**Step 3.** Combine PMOS pull-up network (from Step 2) with NMOS pull-down network (from Step 1) to form fully-complementary CMOS gate
CMOS Sanity Checks

- Equal number of NMOS and PMOS
- NMOS sources tied to ground or to drain of another NMOS
- PMOS sources tied to Vdd or drain of another PMOS
- Inputs tied to pairs of PMOS and NMOS transistors
Next Time

Combinational Building Blocks