Before we get started

Homework 8 due at beginning of lecture
Exam 4 on Monday, December 14th, during class
Today: Exam 4 Review
(Extra) office hours for HW8 and Exam4:
- Dec 9th: Wednesday 12PM - 1PM in CS6367
- Dec 9th: Wednesday 2:30PM - 3:30PM in CS7367
- Dec 10th: Thursday 11AM - 4PM (possibly 5PM) in CS7367
- Dec 10th: Thursday 12:45PM - 2:15PM in CS1308
- Dec 11th: Friday 8AM - 8PM (minus class time) in CS1308
- Dec 11th: Friday 12PM - 1PM in CS6367
- Dec 11th: Friday 1PM - 2:30 in CS6352
- Dec 11th: Friday 2:30PM - 3:30PM in CS7367
- Dec 14th: Monday 8AM - 10AM in CS1308
- Dec 14th: Monday 9:30AM - 11AM in CS1240

Quote of the day:
“Inspiration exists, but it has to find you working”
-- Pablo Picasso 1881 - 1973
Outline

- Chapter 8 Topics
  - Simple logic gates (AND, OR, NOT, XOR, NAND, NOR, XNOR)
  - Multiple inputs on any simple logic gates
  - Sum-of-products design for circuits
    - Take any truth table and design a circuit using AND, OR, and NOT gates
  - DeMorgan’s Law
  - Simple circuits, such as multiplexers, full adders, half adders, decoders, encoders, and incrementors.
  - Principle behind control signal and next state circuits
  - R-S Latch, Gated D-Latch, Master-Slave flip-flop.
  - Derive a truth table from a circuit of simple logic gates
  - Transistors
    - N-type and P-type, and their operation
    - Understand and derive truth tables for transistor circuits.
    - Components of the transistor and how they affect its function
    - Moore’s Law
Find the boolean expression for Output from following truth table. The boolean expression should be in terms of A and B and in sum-of-products (SOP) form.

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

output = A'B'C+A'BC'+AB'C'+AB'C
Fill out the truth table for the following boolean equation.

\[
\text{output} = ABC' + AB'C + A'B'C + A'BC' + A'B'C'
\]

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>output</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
Implement the following boolean equation with logic gates.

$$\text{output} = \overline{A}BC' + AB'C + \overline{A}B'C + A'BC' + \overline{A}B'C'$$
Implement the following boolean equation with logic gates.

output = AB + A'B' + C
Fill out the truth table for the logic gate level circuit below.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th>output</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
<td>C</td>
<td>D</td>
<td>E</td>
<td>F</td>
<td>G</td>
<td>H</td>
<td>I</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>
De Morgan’s Laws

\(\neg(AB) = \neg A + \neg B\) \hspace{1cm} \text{given by cheatsheet}

\(\neg(XY) = \neg X + \neg Y\) \hspace{1cm} \text{by rewriting and changing variable name}

\((XY)' = X' + Y'\) \hspace{1cm} \text{by rewriting and changing negation notation for better visualization}

\((XYZ)' = X' + Y' + Z'\) \hspace{1cm} \text{by expanding previous line to 3 product terms}

\(\neg(A+B) = (\neg A)(\neg B)\) \hspace{1cm} \text{given by cheatsheet}

\((X+Y+Z)' = X'Y'Z'\) \hspace{1cm} \text{by applying same algorithm}
Compute the negation of the following expression using DeMorgan’s Law

\[ AB'C+D'E+(A'+B)(C'+D+E') \]

Negation

\[
= (AB'C+D'E+(A'+B)(C'+D+E'))' \text{ by negating entire expression}
\]

\[
= (AB'C)'(D'E)'((A'+B)(C'+D+E'))' \text{ by applying DeMorgan’s Law on each of the 3 products}
\]

\[
= (AB'C)'(D'E')'((A'+B)'+(C'+D+E')') \text{ by applying DeMorgan’s Law on the rightmost product}
\]

\[
= (A'+B+C')(D+E')(AB'+CD'E) \text{ by applying DeMorgan’s law on each sum on the rightmost product}
\]

\[
= (A'+B+C')(D+E')(AB'+CD'E) \text{ by removing unnecessary parentheses}
\]
Compute the negation of the following expression using DeMorgan’s Law

\((A'+B+C')(D+E')(AB'+CD'E)\)

Negation

\[
\begin{align*}
= & \quad ((A'+B+C')(D+E')(AB'+CD'E))' \quad \text{by negating entire expression} \\
= & \quad (A'+B+C')' + (D+E')' + (AB'+CD'E)' \quad \text{by applying DeMorgan’s Law on each of the 3 sums} \\
= & \quad (AB'C) + (D'E) + ((AB')(CD'E)') \quad \text{by applying DeMorgan’s Law on each sum} \\
= & \quad (AB'C) + (D'E) + ((A'+B)(C'+D+E')) \quad \text{by applying DeMorgan’s law on each product on the rightmost sum} \\
= & \quad AB'C + D'E + (A'+B)(C'+D+E') \quad \text{by removing unnecessary parentheses,} \\
\end{align*}
\]

which matches our original expression
What is the difference between a latch and a flip-flop?

A flip-flop is edge triggered while latch isn’t.
If the figure on the left is a (musical) conductor, then what is the figure on the right?

semiconductor
Multiple Choice Question

What does 22nm transistors mean?

a. the channel length is 22nm
b. the distance between source and drain is 22nm
c. the technology node is 22nm
d. all of the above (correct answer)
e. none of the above
Transistor Behavior

What are the three terminals on a transistor?
Gate, Source, Drain

How does a transistor (pmos) function when it is ON?
Voltage (Vdd) is applied to the gate. An electric field is created so that electrons on the other side of the insulator create a channel, so that electrons can flow from the source to the drain.
Fill out the truth table for the following circuit

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>Q</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
What is Dennard Scaling?

Dennard scaling states that voltage scales down as length/size decreases.

Supply voltage is also acceptable.

Voltage and/or current is also acceptable.

Power density stays constant as transistor size scales is also acceptable.

Power is proportional to area is also acceptable.