Combinatorial Logic

COMP370
Intro to Computer Architecture

VLSI Production Steps

• Cover the chip with photoresist
• Expose a pattern by shining light through a mask
• Dissolve the exposed photoresist
• Expose the exposed chip to a chemical
• Wash off the photoresist

Creation of a Transistor

Start with lightly P doped silicon

Cover the silicon with photoresist
Creation of a Transistor

Expose portions of the photoresist to light by projecting an image onto the chip.

Creation of a Transistor

Wash away the exposed photoresist

Creation of a Transistor

Embed lots of N dopant in the exposed silicon

Creation of a Transistor

Wash away the photoresist
Creation of a Transistor

Cover the chip with photoresist

Expose portions of the photoresist to light

Wash away the exposed photoresist

Grow silicon dioxide in the exposed areas
Creation of a Transistor

Wash off the photoresist

Cover the chip with photoresist

Expose portions of the photoresist to light

Wash off the exposed photoresist
Creation of a Transistor

- **Top View**: Cover the chip with conductive polysilicon.
- **Side View**: Wash off the photoresist. The polysilicon on top of the photoresist is also removed.

NPN Transistor Stick Diagram

- **Gate**
- **Source**
- **Drain**

Conducts when the gate has current.

PNP Transistor Stick Diagram

- **Gate**
- **Source**
- **Drain**

Conducts when the gate does not have current.
**Not Gate**

- When the input is 1 (current) the upper PNP transistor does not conduct. Power cannot flow to the output.
- The lower NPN transistor does conduct. The output is connected to ground.
- When the input is 0 (no current) the upper PNP transistor can conduct power to the output.
- The lower NPN transistor does not connect the output to ground.

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>output</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>

**NAND Gate**

- When either A or B is 0, the output is connected to power.
- When both A and B are one, the output is connected to ground.

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>output</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>

**NOR Gate**

- When A & B are 0, the output is connected to power.
- When either A or B are one, the output is connected to ground.

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>output</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>
**AND Gate**

- AND gate is a NAND followed by a NOT gate

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</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>

**deMorgan’s Law**

- \((X \cdot Y)’\) is a NAND
- \((X + Y)’\) is a NOR
- Frequently logic equations with AND and OR can be converted to NAND and NOR
- \(F = AB + CD\) is the same as \(F = ((AB)’ \cdot (CD)’)’\)

**How many transistors does it take to make AND and NAND gates?**

1. 4 for AND, 2 for NAND
2. 6 for AND, 4 for NAND
3. 4 for AND, 6 for NAND
4. 8 for AND, 6 for NAND

**AND using NOR**

\[F = AB\] is equivalent to \[F = (A’ + B’)’\]
**OR using NAND**

\[ F = A+B \text{ is equivalent to } F = (A' \cdot B')' \]

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>OR</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</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>1</td>
</tr>
</tbody>
</table>

**Circuit with AND and OR gates**

\[ F = AB + CD \]

**Circuit with NAND gates**

\[ F = AB \cdot CD \]

equivalent to

\[ F = AB + CD \]

**Convert to NAND or NOR gates**

- Convert this circuit to use only NOT, NAND or NOR gates.
**NOR Solution**

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

**Alternate Notation**

- A circle on the input to a gate represents the NOT of the input.

**What equation is equivalent to \( F = A'B + C' \)**

1. \( F = (A + B') \cdot C' \)
2. \( F = A'B \cdot C \)
3. \( F = ((A' + B) \cdot C')' \)
4. \( F = (A' + B) \cdot C' \)

**Useful Circuits**

- Some circuits are frequently used in creating larger circuits.
- **Multiplexor** (mux) copies one of \( n \) inputs to the output based on \( \log_2 n \) selector inputs
- **Demultiplexor** (demux) copies input to one of \( n \) outputs based on \( \log_2 n \) selector inputs
**Multiplexor**

- One of the four inputs ($I_0 – I_3$) is copied to the output based on the two selectors ($S_0 – S_1$)

**NAND gate Multiplexor**

- Multiplexors can be created from NAND gates

**Demultiplexor**

- Copies input to one of n outputs ($O_0 – O_3$) based on $\log_2 n$ selector inputs ($S_0 – S_1$)

**Encoders and Decoders**

- An encoder is just like a multiplexor with only one inputs being a 1.
- An encoder has n inputs and $\log_2 n$ outputs.
- The encoder output bits contain the binary number of the one input line that is true.
- A decoder is just like a demultiplexor whose input is always a 1.
- A decoder has n outputs and $\log_2 n$ inputs.
• The decoder implements $F = A'B' + AB$