Gates and Logic

See: P&H Appendix C.2, C.3
• Acts as a conductor or insulator

• Can be used to build amazing things...
• One current controls another (larger) current

• **Static Power:**
  – Keeps consuming power when in the *ON* state

• **Dynamic Power:**
  – Jump in power consumption when switching
Elements

Boron
Silicon
Phosphorus
Silicon
N-Type: Silicon + Phosphorus

Phosphorus Doping
P-Type: Silicon + Boron

Boron Doping
**Semiconductors**

- **n-type** (Si+Phosphorus) has mobile electrons:
  - low voltage (depleted) $\rightarrow$ insulator
  - high voltage (mobile electrons) $\rightarrow$ conductor

- **p-type** (Si+Boron) has mobile holes:
  - low voltage (depleted) $\rightarrow$ insulator
  - high voltage (mobile holes) $\rightarrow$ conductor
P-Type

low $v \rightarrow$ insulator
high $v \rightarrow$ conductor

N-Type

low $v \rightarrow$ conductor
high $v \rightarrow$ insulator
P-Type

low v → insulator
high v → conductor

N-Type

low v → conductor
high v → insulator

Reverse Bias

low v → conductor
high v → insulator
Forward Bias

- Low voltage $\rightarrow$ insulator
- High voltage $\rightarrow$ conductor

P-Type: Low voltage $\rightarrow$ insulator, High voltage $\rightarrow$ conductor

N-Type: Low voltage $\rightarrow$ conductor, High voltage $\rightarrow$ insulator
Conventions:
$v_{dd} = v_{cc} = +1.2\,\text{v} = +5\,\text{v} = \text{hi}$
$v_{ss} = v_{ee} = 0\,\text{v} = \text{gnd}$
PNP Junction

p-type  n-type  p-type
Solid-state switch: The most amazing invention of the 1900s

Emitter = “input”, Base = “switch”, Collector = “output”

PNP Transistor

NPN Transistor

Bipolar Junction Transistors
P-type FET

- Connect Source to Drain when Gate = lo
- Drain must be vdd, or connected to source of another P-type transistor

N-type FET

- Connect Source to Drain when Gate = hi
- Source must be vss, or connected to drain of another N-type transistor
Multiple Transistors

Gate delay
- transistor switching time
- voltage, propagation, fanout, temperature, ...

CMOS design
(complementary-symmetry metal-oxide-semiconductor)
- Power consumption = dynamic + leakage
Conventions:
\[ vdd = vcc = +1.2v = +5v = hi = true = 1 \]
\[ vss = vee = 0v = gnd = false = 0 \]
Function: NOT

- Symbol:

<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>

Truth table
Function: NAND

- Symbol:

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>out</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>
Function: NOR

- Symbol:

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>out</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:**

• **OR:**

• **NOT:**
NAND is universal (so is NOR)

- Can implement any function with just NAND gates
  - De Morgan’s laws are helpful (pushing bubbles)
- Useful for manufacturing

E.g.: XOR (A, B) = A or B but not both (“exclusive or”)

Proof: ?
Some notation:

- **constants**: true = 1, false = 0
- **variables**: a, b, out, ...
- **operators**:
  - $\text{AND}(a, b) = a \land b = a \& b = a \land b$
  - $\text{OR}(a, b) = a \lor b = a \mid b = a \lor b$
  - $\text{NOT}(a) = \bar{a} = !a = \neg a$
Identities useful for manipulating logic equations

- For optimization & ease of implementation

\[
\begin{align*}
  a + 0 &= a \\
  a + 1 &= 1 \\
  a + \bar{a} &= 1 \\
  a 0 &= 0 \\
  a 1 &= a \\
  a \bar{a} &= 0 \\
  (a + b) &= \bar{a} \bar{b} \\
  (a \: b) &= \bar{a} + \bar{b} \\
  a + a \: b &= a \\
  a(b+c) &= ab + ac \\
  a(b+c) &= \bar{a} + \bar{bc}
\end{align*}
\]
• functions: gates ↔ truth tables ↔ equations
• Example: \((a+b)(a+c) = a + bc\)

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>c</th>
<th>a+b</th>
<th>a+c</th>
<th>LHS</th>
<th>bc</th>
<th>RHS</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<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>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

LHS and RHS columns are evaluated for each combination of a, b, and c, showing the truth table for the given logic function.