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
Boron

Silicon

Phosphorus
Silicon
N-Type: Silicon + Phosphorus

Mobile electrons = D conductor

h. + v. = depleted = D ins.
P-Type: Silicon + Boron

Mobile holes = 0 Cond.  
= depleted = Ins.
p-type (Si+Boron) has mobile holes:
- low voltage (depleted) → insulator
- high voltage (mobile holes) → conductor

n-type (Si+Phosphorus) has mobile electrons:
- low voltage (mobile electrons) → conductor
- high voltage (mobile holes) → insulator
Bipolar Junction

P-Type

N-Type

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

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

P-Type

low v → insulator
high v → conductor

N-Type

low v → conductor
high v → insulator
Forward Bias

P-Type

low v → insulator
high v → conductor

N-Type

low v → conductor
high v → insulator
PN Junction “Diode”

Conventions:
$vdd = vcc = +1.2v = +5v = hi$
$vss = vee = 0v = gnd$
• Solid-state switch: The most amazing invention of the 1900s

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

Vdd \hspace{1cm} \text{in} \hspace{1cm} \text{out} \hspace{1cm} Vss

<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>
\[
\text{NOT} \left( \frac{\text{AND}}{\text{AND}} \right) = \left( \frac{\text{AND} \text{ NOT} \text{ NOT}}{\text{AND} \text{ NOT} \text{ NOT}} \right) \text{ OR } \left( \frac{\text{AND} \text{ NOT} \text{ NOT}}{\text{AND} \text{ NOT} \text{ NOT}} \right) \text{ NOT} \text{ NOT} \text{ B}
\]

- **AND:**
  - Symbol: \( \text{AND} \)
  - Diagram: \( \text{AND} \)

- **OR:**
  - Symbol: \( \text{OR} \)
  - Diagram: \( \text{OR} \)

- **NOT:**
  - Symbol: \( \text{NOT} \)
  - Diagram: \( \text{NOT} \)
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:
  - AND(a, b) = a \land b
  - OR(a, b) = a \lor b
  - NOT(a) = \overline{a}

Truth tables:

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>a &amp; b</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>

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>a \lor b</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>

<table>
<thead>
<tr>
<th>a</th>
<th>a'</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
Identities useful for manipulating logic equations

- For optimization & ease of implementation

\[ a + 0 = a \]
\[ a + 1 = 1 \]
\[ a + \bar{a} = 1 \]
\[ a \ 0 = 0 \]
\[ a \ 1 = a \]
\[ a \ \bar{a} = 0 \]
\[ \overline{(a + b)} = \bar{a} \ \bar{b} \]
\[ \overline{(a \ b)} = \bar{a} + \bar{b} \]
\[ a + a \ b = a \]
\[ a(b+c) = ab + ac \]
\[ a(b+c) = \bar{a} + bc \]
• 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>0</td>
<td>1</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>
\[(a + b)(a + c)\]

\[
a \cdot a + a \cdot c + b \cdot a + b \cdot c
\]

\[
\frac{a \cdot a + a \cdot c + b \cdot a + b \cdot c}{a - 1}
\]

\[
a \left(1 + c + b\right) + b \cdot c
\]

\[
\frac{a \cdot 1 + b \cdot c}{a + b \cdot c}
\]