Today's learning goals

• Relate algorithms for integer operations to bitwise boolean operations
• Correctly use XOR and bit shifts
• List the truth tables and meanings for negation, conjunction, disjunction, exclusive or, implication.
• Relate boolean operations to applications in combinatorial circuits.
Arithmetic in hardware

**Other models are possible**

**Inputs** e.g. coefficients in fixed-width binary representation

Combinatorial (Logic) Circuit

**Outputs** e.g. coefficients in fixed-width binary representation

Values flow left to right: possible values on a wire are 0 (low) or 1 (high)

Circuit elements: wires, gates

Gates may share input; outputs of gate can become inputs to other gates
## Definition tables

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>y</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>y</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>NOT x</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

[Diagrams for AND, XOR, and NOT gates]
Example: logical circuit

Output when \( x = 1, y = 0, z = 0, w = 1 \) is ____
Output when \( x = 1, y = 1, z = 1, w = 1 \) is ____
Output when \( x = 0, y = 0, z = 0, w = 1 \) is ____
Example: logical circuit

Which of the following is true about all possible input values x,y,z,w? “The output out is set to 1 exactly when
A. x is 0, and is set to 0 otherwise”
B. \((xyzw)_2\) is less than 8, and is set to 0 otherwise”
C. \((wzyx)_{2,4}\) is an even integer, and is set to 0 otherwise”
D. All of the above
E. None of the above
Circuits

Draw a logic circuit with inputs x and y whose output is always 0. Can you use exactly 1 of the gates we’ve seen so far?
Fixed-width addition: adding one bit at a time, using the usual column-by-column and carry arithmetic, and dropping the carry from the leftmost column so the result is the same width as the summands. In many cases, this gives representation of the correct value for the sum when we interpret the summands in fixed-width binary or in 2s complement.

- Inputs \(x_0, y_0, x_1, y_1\) represent \((x_1x_0)_2,2\) and \((y_1y_0)_2,2\)
- Outputs \((x_1x_0)_2,2 + (y_1y_0)_2,2\)

How many bits of output should we allow for?
A. 2
B. 4
C. 6
D. 8
E. I don’t know
Fixed-width w addition

Rosen p. 251, 826

Translate one symbol sum, carry to circuit

\[
\begin{array}{c}
x_{w-1} \cdots x_1 x_0 \\
+ y_{w-1} \cdots y_1 y_0 \\
\hline
1 1 1 0 1 0 1
\end{array}
\]

\[
\begin{array}{cccc}
\text{Input} & \text{Output} & \text{Input} & \text{Output} \\
x_0 & y_0 & s_0 & c_0 \\
1 & 1 & 1 & 1 \\
1 & 0 & 1 & 0 \\
0 & 1 & 0 & 1 \\
0 & 0 & 0 & 0
\end{array}
\]
Fixed-width w addition

\[
\begin{array}{cccc}
1 & 1 & 0 & 1 \\
0 & 0 & 0 & 1
\end{array}
\]

\[
+ \begin{array}{cccc}
1 & 0 & 0 & 0 \\
1 & 0 & 1 & 1
\end{array}
\]

\[
\begin{array}{cccc}
1 & 1 & 1 & 0 \\
1 & 1 & 0 & 1
\end{array}
\]

Translate one symbol sum, carry to circuit

<table>
<thead>
<tr>
<th>\text{Input}</th>
<th>\text{Output}</th>
</tr>
</thead>
</table>
| \begin{array}{cc}
x_0 & y_0 \\
1 & 1 \\
1 & 0 \\
0 & 1 \\
0 & 0
\end{array} | \begin{array}{c}s_0 \\
1 \\
1 \\
1 \\
0 \\
0
\end{array} |

<table>
<thead>
<tr>
<th>\text{Input}</th>
<th>\text{Output}</th>
</tr>
</thead>
</table>
| \begin{array}{cc}
x_0 & y_0 \\
1 & 1 \\
1 & 0 \\
0 & 1 \\
0 & 0
\end{array} | \begin{array}{c}c_0 \\
1 \\
1 \\
0 \\
0
\end{array} |
Fixed-width 2 binary addition

- Inputs $x_0, y_0, x_1, y_1$ represent $(x_1x_0)_{2,2}$ and $(y_1y_0)_{2,2}$
- Outputs $z_0, z_1, z_2$ represent $(z_2z_1z_0)_{2,3} = (x_1x_0)_{2,2} + (y_1y_0)_{2,2}$ (may require up to width 3)
Fixed-width 2 binary addition

\[ x_0 \quad y_0 \quad \text{XOR} \quad \text{AND} \quad x_1 \quad y_1 \quad \text{XOR} \quad \text{AND} \]
Fixed-width 2 binary addition

\[ x_0 \quad \text{XOR} \quad x_1 \quad \text{AND} \quad y_0 \quad \text{AND} \quad y_1 \]
Logic

- Use gates and circuits to express arithmetic.
- Precisely express true facts and invariant statements.
- Identify valid arguments (patterns of reasoning) that could be used in proofs.
Definitions

- **Proposition**: declarative sentence that is T or F (not both)
- **Propositional variable**: variables that represent propositions.
- **Compound proposition**: new propositions formed from existing propositions using logical operators.
- **Truth table**: table with 1 row for each of the possible combinations of truth values of the input and an additional column that shows the truth value of the result of the operation corresponding to a particular row.
Circuits ~ Propositions

- 0 (off) ~ False
- 1 (on) ~ True

### Conjunction

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>p</td>
<td>q</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
</tr>
</tbody>
</table>

### Exclusive or

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>p</td>
<td>q</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
</tr>
</tbody>
</table>

For next time

• Read website carefully
  http://cseweb.ucsd.edu/classes/fa20/cse20-a/

• Next pre-class reading:
  • Section 1.3 Definitions 1 and 2