Digital logic: Functional completeness, logic simplification

Yipeng Huang

Rutgers University

April 22, 2021
Table of contents

Announcements

Combinational logic
  Definitions for more-than-2-input gates

Functional completeness
  The set of logic gates \{\text{NOT, AND, OR}\} is universal
  The NAND gate is universal
  The NOR gate is universal

Basic algorithms for logic simplification
Looking ahead

Class plan

1. PA5 due Monday, 4/26.
Table of contents

Announcements

Combinational logic
   Definitions for more-than-2-input gates

Functional completeness
   The set of logic gates \{\text{NOT, AND, OR}\} is universal
   The NAND gate is universal
   The NOR gate is universal

Basic algorithms for logic simplification
Combinational vs. sequential logic

**Combinational logic**
- No internal state nor memory
- Output depends entirely on input
- Examples: NOT, AND, NAND, OR, NOR, XOR, XNOR gates, decoders, multiplexers.

**Sequential logic**
- Has internal state (memory)
- Output depends on the inputs and also internal state
- Examples: latches, flip-flops, Mealy and Moore machines, registers, pipelines, SRAMs.
More-than-2-input AND gate

Table: Truth table for three-input AND gate

<table>
<thead>
<tr>
<th>$A$</th>
<th>$B$</th>
<th>$C$</th>
<th>$ABC$</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>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</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>0</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>1</td>
</tr>
</tbody>
</table>
More-than-2-input OR gate

A \lor B

\begin{array}{ccc|c}
A & B & C & A + B + C \\
0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 \\
0 & 1 & 0 & 1 \\
0 & 1 & 1 & 1 \\
1 & 0 & 0 & 1 \\
1 & 0 & 1 & 1 \\
1 & 1 & 0 & 1 \\
1 & 1 & 1 & 1 \\
\end{array}

Table: Truth table for three-input AND gate
Table of contents

Announcements

Combinational logic
   Definitions for more-than-2-input gates

Functional completeness
   The set of logic gates \{NOT, AND, OR\} is universal
   The NAND gate is universal
   The NOR gate is universal

Basic algorithms for logic simplification
The set of logic gates \{\textsc{NOT, AND, OR}\} is universal

And

\[
\begin{align*}
a & \quad \text{out} \\
b & \quad \text{out} = a \land b
\end{align*}
\]

Or

\[
\begin{align*}
a & \quad \text{out} \\
b & \quad \text{out} = a \lor b
\end{align*}
\]

Not

\[
\begin{align*}
a & \quad \text{out} \\
& \quad \text{out} = \neg a
\end{align*}
\]

Figure: Source: CS:APP
The set of logic gates \{NOT, AND, OR\} is universal

- Any truth table can be expressed as sum of products form.
- Write each row with output 1 as a product (minterm).
- Sum the products (minterm).
- Forms a disjunctive normal form (DNF).
- \( D = \overline{A}BC + A\overline{B}C \)
- Always only needs NOT, AND, OR gates.
- Supplementary slides example...

Logical Completeness

Can implement ANY truth table with AND, OR, NOT.

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

1. AND combinations that yield a "1" in the truth table.
2. OR the results of the AND gates.

Sum of products OR of AND clauses
The NAND gate is universal

NOT gate as a single NAND gate

\[ A \rightarrow \overline{A} = A \rightarrow \overline{A} \]

<table>
<thead>
<tr>
<th>A</th>
<th>\overline{A}</th>
<th>AA</th>
<th>\overline{AA}</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Table: \( \overline{A} = \overline{AA} \)

AND gate as two NAND gates

\[ A \rightarrow \overline{A} = A \rightarrow \overline{A} \]

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>AB</th>
<th>\overline{AB}</th>
<th>\overline{AB}</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Table: \( AB = \overline{AB} \)
The NAND gate is universal

De Morgan’s Law

<table>
<thead>
<tr>
<th>$A$</th>
<th>$B$</th>
<th>$\overline{A}$</th>
<th>$\overline{B}$</th>
<th>$\overline{A} \overline{B}$</th>
<th>$A + B$</th>
<th>$\overline{A + B}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Table: $\overline{A \overline{B}} = \overline{A + B}$

OR gate as three NAND gates

$A \overline{B} = A + B$
The NOR gate is universal

NOT gate as a single NOR gate

\[ A \rightarrow \overline{A} = \overline{A} \]

\[ \begin{array}{c|c|c|c}
 A & \overline{A} & A + A & \overline{A} + \overline{A} \\
 \hline
0 & 1 & 0 & 1 \\
1 & 0 & 1 & 0 \\
\end{array} \]

Table: \( \overline{A} = A + A \)

OR gate as two NOR gates

\[ A \rightarrow \overline{A} \rightarrow AB \]

\[ A \rightarrow \overline{A + B} \rightarrow A + B \]

\[ \begin{array}{c|c|c|c|c|c}
 A \ & B \ & A + B \ & \overline{A + B} \ & \overline{A + B} \\
 \hline
0 \ & 0 \ & 0 \ & 1 \ & 0 \\
0 \ & 1 \ & 1 \ & 0 \ & 1 \\
1 \ & 0 \ & 1 \ & 0 \ & 1 \\
1 \ & 1 \ & 1 \ & 0 \ & 1 \\
\end{array} \]

Table: \( AB = \overline{AB} \)
The NOR gate is universal

De Morgan’s Law

<table>
<thead>
<tr>
<th>$A$</th>
<th>$B$</th>
<th>$\overline{A}$</th>
<th>$\overline{B}$</th>
<th>$\overline{A} + \overline{B}$</th>
<th>$AB$</th>
<th>$\overline{AB}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</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>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Table: $\overline{A} + \overline{B} = \overline{AB}$

AND gate as three NOR gates

\[
\begin{align*}
A & \Rightarrow \overline{A} \\
B & \Rightarrow \overline{B} \\
\overline{A} + \overline{B} & = AB
\end{align*}
\]
Table of contents

Announcements

Combinational logic
   Definitions for more-than-2-input gates

Functional completeness
   The set of logic gates \{\text{NOT}, \text{AND}, \text{OR}\} is universal
   The \text{NAND} gate is universal
   The \text{NOR} gate is universal

Basic algorithms for logic simplification