Lecture 6: Universal Gates

CSE 140: Components and Design Techniques for Digital Systems
Fall 2014

CK Cheng
Dept. of Computer Science and Engineering
University of California, San Diego
Combinational Logic: Other Types of Gates

- Universal Set of Gates
- Other Types of Gates
  1) XOR
  2) NAND / NOR
  3) Block Diagram Transfers
Universal Set of Gates: Motivation

- AND, OR, NOT: Logic gates related to reasoning from Aristotle (384-322BCE)
- NAND, NOR: Inverted AND, Inverted OR gates. VLSI technologies. All gates are inverted.
- Multiplexer + input table: FPGA technology. Table based logic for programmability.
- XOR: Parity check

In the future, we may have new sets of gates due to new technologies. Given a set of gates, can this set of gates cover all possible switching functions?
Universal Set

Universal Set: A set of gates such that every switching function can be implemented with the gates in this set.

Examples

\{\text{AND, OR, NOT}\}

\{\text{AND, NOT}\}

\{\text{OR, NOT}\}
Universal Set

Universal set is a powerful concept to identify the coverage of a set of gates afforded by a given technology.

If the set of gates can implement AND, OR, and NOT gates, the set is universal.
Universal Set

Universal Set: A set of gates such that every Boolean function can be implemented with the gates in this set.

Examples

\{\text{AND, OR, NOT}\}

\{\text{AND, NOT}\} OR can be implemented with \text{AND} \& NOT gates \hspace{1cm} a + b = (a' \ b')'

\{\text{OR, NOT}\} AND can be implemented with \text{OR} \& NOT gates \hspace{1cm} ab = (a' +b')'

\{\text{XOR}\} is not universal

\{\text{XOR, AND}\} is universal
Is the set \{AND, OR\} (but no NOT gate) universal?
A. Yes
B. No
Is the set \{f(x,y)=xy'\} universal?
A. Yes
B. No
\{\text{NAND, NOR}\}

\{\text{NOR}\}

\{\text{XOR, AND}\}

\[ X \oplus 1 = X \cdot 1' + X' \cdot 1 = X' \] if constant “1” is available.
Other Types of Gates: Properties and Usage

1) XOR \( X \oplus Y = XY' + X'Y \)
It is a parity function (examples)
Useful for testing because the flipping of a single input changes the output.

<table>
<thead>
<tr>
<th>id</th>
<th>x</th>
<th>y</th>
<th>x ( \oplus ) y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</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>2</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>x=0</th>
<th>x=1</th>
</tr>
</thead>
<tbody>
<tr>
<td>y=0</td>
<td>0</td>
</tr>
<tr>
<td>y=1</td>
<td>1</td>
</tr>
</tbody>
</table>
Other Types of Gates: Properties and Usage

1) XOR \( X \oplus Y = XY' + X'Y \)

(a) Commutative \( X \oplus Y = Y \oplus X \)
(b) Associative \((X \oplus Y) \oplus Z = X \oplus (Y \oplus Z)\)
(c) \(1 \oplus X = X', \quad 0 \oplus X = 0X' + 0'X = X\)
(d) \(X \oplus X = 0, \quad X \oplus X' = 1\)
e) If $ab = 0$, then $a \oplus b = a + b$
Proof: If $ab = 0$, then $a = a (b + b') = ab + ab' = ab'$
$b = b (a + a') = ba + ba' = a'b$
$a + b = ab' + a'b = a \oplus b$

f) $f(x,y) = x \oplus xy' \oplus x'y \oplus (x + y) \oplus x =$ ?
(Priority of operations: AND, $\oplus$, OR)

Hint: We apply Shannon’s Expansion.
Shannon’s Expansion (for switching functions)

Formula: \( f(x, Y) = x \cdot f(1, Y) + x' \cdot f(0, Y) \)

Proof by enumeration:
If \( x = 1 \), \( f(x, Y) = f(1, Y) : 1 \cdot f(1, Y) + 1' \cdot f(0, Y) = f(1, Y) \)

If \( x = 0 \), \( f(x, Y) = f(0, Y) : 0 \cdot f(1, Y) + 0' \cdot f(0, Y) = f(0, Y) \)
Back to our problem...

Simplify the function
\[ f(X,Y) = X \oplus XY' \oplus X' Y \oplus (X+Y) \oplus X \]

Case \( X = 1 \): \( f(1,Y) = 1 \oplus Y' \oplus 0 \oplus 1 \oplus 1 = Y \)
Case \( X = 0 \): \( f(0,Y) = 0 \oplus 0 \oplus Y \oplus Y \oplus 0 = 0 \)

Thus, using Shannon’s expansion, we have
\[ f(X,Y) = Xf(1,Y) + X'f(0,Y) = XY \]
XOR gates

iClicker: $a + (b \oplus c) = (a + b) \oplus (a + c)$?
A. Yes
B. No
2) NAND, NOR gates

NAND, NOR gates are not associative

Let $a | b = (ab)'$

$$(a | b) | c \neq a | (b | c)$$
3) Block Diagram Transformation

a) Reduce # of inputs.

\[ \begin{align*}
\text{original} & \iff \text{new} \\
\text{original} & \iff \text{new}
\end{align*} \]
b. DeMorgan’s Law

\[(a+b)’ = a’ b’\]

\[(ab)’ = a’ + b’\]
c. Sum of Products (Using only NAND gates)

Sum of Products (We create many bubbles with NOR gates)
d. Product of Sums (NOR gates only)

We will create many bubbles with NAND gates.
NAND, NOR gates

Remark:
Two level NAND gates: Sum of Products
Two level NOR gates: Product of Sums
Part II. Sequential Networks

Flip flops
Specification
Implementation
Reading

[Harris] Chapter 3, 3.1, 3.2