ECE/ENGRD 2300
Digital Logic & Computer Organization
Spring 2018

Boolean Algebra
Announcements

- **Additional handouts posted on course website**
  - Weekly calendar
  - TA office hours start next week
  - Regrade request form

- **Lab 2 will be released tonight**
Fun Facts: Number Bases

- Digital computers represent numbers in binary
- We count in 10
- Mayans used positional notion based on 20

Maya numerals
Review: Digital Abstraction

- Typically digital implies a *binary* system (base 2)
  - Only 2 recognized values (*logic levels*)
    - High / Low
    - True / False
    - On / Off
    - ...

- Logic levels are particular voltage ranges

- Key idea: don’t allow “0” to be easily mistaken for a “1” or vice versa
Review: Logic Gates

- Logic gates are functions: take one or more binary inputs and produce a binary output.

<table>
<thead>
<tr>
<th>A</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Truth Table

**NOT Gate**

- Gate Diagram
- Truth Table:
  - A: 0, Y: 1
  - A: 1, Y: 0

**AND Gate**

- Gate Diagram
- Truth Table:
  - A: 0, B: 0, Y: 0
  - A: 0, B: 1, Y: 0
  - A: 1, B: 0, Y: 0
  - A: 1, B: 1, Y: 1

**OR Gate**

- Gate Diagram
- Truth Table:
  - A: 0, B: 0, Y: 0
  - A: 0, B: 1, Y: 1
  - A: 1, B: 0, Y: 1
  - A: 1, B: 1, Y: 1
Exercise: Truth Table

- Write down the truth table for a three-input Boolean function $Y = f(A,B,C)$ that outputs
  - $Y = 1$ if $A \cdot B$ is not equal to $B' + C$
  - $Y = 0$ otherwise

<table>
<thead>
<tr>
<th>ABC</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td></td>
</tr>
<tr>
<td>001</td>
<td></td>
</tr>
<tr>
<td>010</td>
<td></td>
</tr>
<tr>
<td>011</td>
<td></td>
</tr>
<tr>
<td>100</td>
<td></td>
</tr>
<tr>
<td>101</td>
<td></td>
</tr>
<tr>
<td>110</td>
<td></td>
</tr>
<tr>
<td>111</td>
<td></td>
</tr>
</tbody>
</table>
Boolean Algebra

• Mathematical foundation for analyzing and simplifying digital circuits

• Boolean algebra (George Boole, 1854)
  – Two-valued algebraic system
  – Used to formulate true or false postulations

• Switching algebra (Claude Shannon, 1938)
  – Adopted Boolean algebra for digital circuits
  – Terms “Boolean algebra” and “switching algebra” are used interchangeably
Basic “Ingredients”

• Variables that have values of either 1 or 0 (True or False)

• Basic operators are AND, OR, and NOT
Some Important Definitions

- **Literal**: a single variable or its complement
  - e.g., $X$ (positive literal), $X'$ (negative literal)

- **Product term**: AND of (more than one) literals
  - e.g., $X' \cdot Y$

- **Sum term**: OR of literals
  - e.g., $X + Y + Z'$
Operator Precedence

• What does $W \cdot X' + Y \cdot Z$ mean?

• Operator precedence rules
  1. NOT (highest priority)
  2. AND
  3. OR (lowest)
Some More Definitions

• **Normal term**: Product or sum term in which every variable appears, and exactly once

• **Minterm**: Normal product
  – e.g., \((X \cdot Y' \cdot Z)\) for a 3-input Boolean function

• **Maxterm**: Normal sum
  – e.g., \((X' + Y + Z')\)
Minterms & Maxterms

<table>
<thead>
<tr>
<th>XYZ</th>
<th>minterm</th>
<th>Minterm name</th>
<th>maxterm</th>
<th>Maxterm name</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>X’•Y’•Z’</td>
<td>m₀</td>
<td>X+Y+Z</td>
<td>M₀</td>
</tr>
<tr>
<td>001</td>
<td>X’•Y’•Z</td>
<td>m₁</td>
<td>X+Y+Z’</td>
<td>M₁</td>
</tr>
<tr>
<td>010</td>
<td>X’•Y•Z’</td>
<td>m₂</td>
<td>X+Y’+Z</td>
<td>M₂</td>
</tr>
<tr>
<td>011</td>
<td>X’•Y•Z</td>
<td>m₃</td>
<td>X+Y’+Z’</td>
<td>M₃</td>
</tr>
<tr>
<td>100</td>
<td>X•Y’•Z’</td>
<td>m₄</td>
<td>X’+Y+Z</td>
<td>M₄</td>
</tr>
<tr>
<td>101</td>
<td>X•Y’•Z</td>
<td>m₅</td>
<td>X’+Y+Z’</td>
<td>M₅</td>
</tr>
<tr>
<td>110</td>
<td>X•Y•Z’</td>
<td>m₆</td>
<td>X’+Y’+Z</td>
<td>M₆</td>
</tr>
<tr>
<td>111</td>
<td>X•Y•Z</td>
<td>m₇</td>
<td>X’+Y’+Z’</td>
<td>M₇</td>
</tr>
</tbody>
</table>
Canonical Form

• A canonical form is a unique representation for a Boolean function
  – Given a fixed ordering of the input variables
  – Two equivalent functions have the same canonical form

• Truth table is a canonical form
  – e.g., (A’+B’) and (AB)’ have the same truth table
  – \(2^n\) rows always required for an n-input function

• Other canonical representations
  – Canonical sum
  – Canonical product
  – Even more compact forms exist (outside the scope of this course)
Canonical Sum of a Logic Function

• **Sum of minterms** that correspond to the on-set of a function
  - On-set: Input combinations that make $F=1$
  - $F = X'\cdot Y'\cdot Z' + X'\cdot Y\cdot Z + X\cdot Y'\cdot Z' + X\cdot Y\cdot Z$
    $= \sum_{X,Y,Z}(0,3,4,7)$
Canonical Product of a Logic Function

- **Product of maxterms** that correspond to the off-set of a function
  - Off-set: Input combinations that make $F=0$
  - $F = (X+Y+Z')(X+Y'+Z)(X'+Y+Z')(X'+Y'+Z) = \Pi_{X,Y,Z}(1,2,5,6)$
  - $F = \Pi_{X,Y,Z}(1,2,5,6) = \Sigma_{X,Y,Z}(0,3,4,7)$
Exercise: Canonical Forms

- Write down the canonical sum & product forms for basic operators NOT, AND, OR

<table>
<thead>
<tr>
<th>A</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>Y</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>Y</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>
Axioms of Boolean Algebra

- **Statements that are assumed true**
- **Obey the principle of duality**
  - Interchange 1 and 0, AND and OR, still correct
  - Many axioms come in pairs
Axioms of Boolean Algebra

• Binary
  
  \[(A1)\quad X = 0 \text{ if } X \neq 1\quad (A1')\quad X = 1 \text{ if } X \neq 0\]

• Complement
  
  \[(A2)\quad \text{If } X = 0, \text{ then } X' = 1\quad (A2')\quad \text{If } X = 1, \text{ then } X' = 0\]
# Axioms of Boolean Algebra

- **AND** and **OR**

<table>
<thead>
<tr>
<th>Axiom</th>
<th>Expression</th>
<th>Axiom</th>
<th>Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>(A3)</td>
<td>0•0=0</td>
<td>(A3′)</td>
<td>1+1=1</td>
</tr>
<tr>
<td>(A4)</td>
<td>1•1=1</td>
<td>(A4′)</td>
<td>0+0=0</td>
</tr>
<tr>
<td>(A5)</td>
<td>0•1=1•0=0</td>
<td>(A5′)</td>
<td>1+0=0+1=1</td>
</tr>
</tbody>
</table>

- **A1-A5 completely define Boolean algebra**
  - Everything else derived from these axioms
Single Variable Theorems

• Identity: \((T1)\) \(X \cdot 1 = X\) \((T1')\) \(X + 0 = X\)

• Null Element: \((T2)\) \(X \cdot 0 = 0\) \((T2')\) \(X + 1 = 1\)

• Idempotency: \((T3)\) \(X \cdot X = X\) \((T3')\) \(X + X = X\)

• Involution: \((T4)\) \((X')' = X\)

• Complements: \((T5)\) \(X \cdot X' = 0\) \((T5')\) \(X + X' = 1\)

• Can prove by *perfect induction*
  – Show that all possible inputs meet the theorem
Proof by Perfect Induction

(T3) \( X \cdot X = X \)

\[
\begin{align*}
X = 0 & \Rightarrow 0 \cdot 0 = 0 \\
X = 1 & \Rightarrow 1 \cdot 1 = 1
\end{align*}
\]

(T3') \( X + X = X \)

\[
\begin{align*}
X = 0 & \Rightarrow 0 + 0 = 0 \\
X = 1 & \Rightarrow 1 + 1 = 1
\end{align*}
\]

(T5) \( X \cdot X' = 0 \)

(T5') \( X + X' = 1 \)
Two and Three Variable Theorems

• **Commutativity**
  
  \[(T6)\] \(X \cdot Y = Y \cdot X\) \hspace{1cm} \[(T6')\] \(X + Y = Y + X\)

• **Associativity**
  
  \[(T7)\] \((X \cdot Y) \cdot Z = X \cdot (Y \cdot Z)\) \hspace{1cm} \[(T7')\] \((X + Y) + Z = X + (Y + Z)\)

• **Distributivity**
  
  \[(T8)\] \(X \cdot Y + X \cdot Z = X \cdot (Y + Z)\) \hspace{1cm} \[(T8')\] \((X + Y) \cdot (X + Z) = X + (Y \cdot Z)\)

  AND distributes over OR \hspace{1cm} OR distributes over AND
Two and Three Variable Theorems

• **Covering**
  
  \[(T9) \quad X \cdot (X+Y) = X \quad (T9') \quad X + X \cdot Y = X \]

• **Combining**
  
  \[(T10) \quad X \cdot Y + X \cdot Y' = X \quad (T10') \quad (X+Y) \cdot (X+Y') = X \]

• **Consensus**
  
  \[(T11) \quad X \cdot Y + X' \cdot Z + Y \cdot Z = X \cdot Y + X' \cdot Z \]
  
  \[(T11') \quad (X+Y) \cdot (X'+Z) \cdot (Y+Z) = (X+Y) \cdot (X'+Z) \]
Exercise

• Prove algebraically: \((T9')\) \(X + X \cdot Y = X\)

• Solution

\[
X + X \cdot Y = X \cdot 1 + X \cdot Y = X \cdot (1 + Y) = X \cdot 1 = X
\]

- \(T1\) (identity)
- \(T8\) (distributivity)
- \(T2'\) (null elements)
- \(T1\) (identity)
De Morgan’s Theorem

- So important, also known as De Morgan’s Law

\[(T12)\] \((X_1 \cdot X_2 \cdot \ldots \cdot X_n)' = X_1' + X_2' + \ldots + X_n'\]

\[(T12')\] \((X_1 + X_2 + \ldots + X_n)' = X_1' \cdot X_2' \cdot \ldots \cdot X_n'\]
De Morgan Example

- By DeMorgan’s Law
  \((X \cdot Y \cdot Z)' = X' + Y' + Z'\)

- Proof by perfect induction

<table>
<thead>
<tr>
<th>XYZ</th>
<th>(X \cdot Y \cdot Z)'</th>
<th>X' + Y' + Z'</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>001</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>010</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>011</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>100</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>101</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>110</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>111</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Before Next Class

• H&H 2.4-2.7

Next Time

Combinational Logic Minimization