Boolean Algebra and Combinatorial Circuits
Introduction

• Combinatorial circuit
  • The output is uniquely defined for every combination of inputs
  • No memory: previous inputs and the state of the system do not affect
  • Only combination of the logic gates

• Sequential circuit
  • Output is a function, not only of the input but also of the state of the system
Basis Gate type

AND

\[ x_1 \land x_2 = \begin{cases} 
1 & \text{if } x_1 = 1 \text{ and } x_2 = 1 \\
0 & \text{otherwise} 
\end{cases} \]

OR

\[ x_1 \lor x_2 = \begin{cases} 
1 & \text{if } x_1 = 1 \text{ or } x_2 = 1 \\
0 & \text{otherwise} 
\end{cases} \]

NOT

\[ \bar{x} = \begin{cases} 
1 & \text{if } x = 0 \\
0 & \text{otherwise} 
\end{cases} \]
Example of combinatorial circuit

\[ y = (x_1 \land x_2) \lor x_3 \]
Properties

Associative Laws
\[(a \lor b) \lor c = a \lor (b \lor c) \quad (a \land b) \land c = a \land (c \land b) \quad \forall (a, b, c) \in Z_2\]

Commutative laws
\[a \lor b = b \lor a \quad a \land b = b \land a \quad \forall (a, b, c) \in Z_2\]

Distributive laws
\[a \land (b \lor c) = (a \land b) \lor (a \land c)\]
\[a \lor (b \land c) = (a \lor b) \land (a \lor c) \quad \forall a, b, c \in Z_2\]

Identity Laws
\[a \lor 0 = a \quad a \land 1 = a \quad \forall a \in Z_2\]

Complement Laws
\[a \lor \bar{a} = 1 \quad a \land \bar{a} = 0 \quad \forall a \in Z_2\]
Boolean Algebra
Laws of Boolean algebra

• Associative Law
  \[(x + y) + z = x + (y + z)\]
  \[(x \cdot y) \cdot z = x \cdot (y \cdot z)\]

• Communicative Law
  \[x + y = y + x, \quad x \cdot y = y \cdot x \quad \forall x \in S\]

• Distributive law
  \[x \cdot (y + z) = (x \cdot y) + (x \cdot z)\]
  \[x + (y \cdot z) = (x + y) \cdot (x + z) \quad \forall x \in S\]

• Identity Law
  \[x + 0 = x, \quad x \cdot 1 = x \quad \forall x \in S\]

• Complement Law
  \[x + x' = 1 \quad x \cdot x' = 0 \quad \forall x \in S\]
Theorem

Let $B = (s, +, \cdot, ', 0, 1)$ be a Boolean algebra. Here are the properties
(a) Idempotent Law:
    \[ x + x = x, \quad xx = x \quad \forall x \in S \]
(b) Bound Law
    \[ x + 1 = 1, \quad x \cdot 0 = 0 \quad \forall x \in S \]
(c) Absorption Laws
    \[ x + xy = x, \quad x(x + y) = x \quad \forall x, y \in S \]
(d) Involution Laws
    \[ (x')' = x \quad \forall x \in S \]
(e) 0 and 1 laws
    \[ 0' = 1, \quad 1' = 0 \]
(f) De Morgan’s law for Boolean algebras
    \[ (x + y)' = x'y', \quad (xy)' = x' + y' \quad \forall x, y \in S \]
Boolean Functions and Synthesis of Circuits

• Sometime we need a circuit for a specified task
• Ex. Exclusive or from basis Boolean type

\[ x_1 \oplus x_2 \]

Input set = \{(1,1),(1,0),(0,1),(0,0)\}
Output range \( z_2 = \{0,1\} \)
Writing

\[ x_1 \oplus x_2 = X(x_1, x_2) \]

Solution: Use the combination of basis logic gate to solve the equation
Definition

Let $X(x_1, x_2, ..., x_n)$ be a Boolean expression

A Boolean function has a form

$$f(x_1, x_2, ..., x_n) = X(x_1, x_2, ..., x_n)$$

Ex $f: z_2^3 \rightarrow z_2$

$$f(x_1, x_2, x_3) = x_1 \land (\bar{x}_2 \lor x_3)$$

<table>
<thead>
<tr>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
<th>$f(x_1, x_2, x_3)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</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>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</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>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Ex.

Find combinatorial circuit for this truth table

Consider the first row get

\[ f_1(x_1, x_2, x_3) = x_1 \land x_2 \land x_3 \]

4\(^{th}\) row

\[ f_4(x_1, x_2, x_3) = x_1 \land \bar{x}_2 \land \bar{x}_3 \]

6\(^{th}\) row

\[ f_6(x_1, x_2, x_3) = \bar{x}_1 \land x_2 \land \bar{x}_3 \]

From the theory OR all the term

\[ f(x_1, x_2, x_3) = (x_1 \land x_2 \land x_3) \lor (x_1 \land \bar{x}_2 \land \bar{x}_3) \lor (\bar{x}_1 \land x_2 \land \bar{x}_3) \]

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</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>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</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>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
minterm

Let $x_1, x_2, \ldots, x_n$ is a Boolean expression of the form
$$y_1 \land y_2 \land \ldots \land y_n$$

Where $y_i$ is either $x_i$ or $\bar{x}_i$

Theorem:
If $f : z_2^n \to z_2$, then $f$ is a Boolean function. If $f$ is not identically zero, let $A_1, \ldots, A_k :$ elements $A_i$ of $Z_2^n$ for which $f(A_i) = (a_1, \ldots, a_n)$, set
$$m_i = y_1 \land \ldots \land y_n$$

Where
$$y_j = \begin{cases} x_j & \text{if } a_j = 1 \\ \bar{x}_j & \text{if } a_j = 0 \end{cases}$$

Then
$$f(x_1, \ldots, x_n) = m_1 \lor \ldots \lor m_k$$

Called: Disjunctive normal form of function $f$
Applications

A design for Boolean combinatorial circuit from some combinatorial of gate AND, OR and not

A gate is a function form $Z_2^n$ into $Z_2$

AND gate $Z_2^2$ into $Z_2$

NOT gate $Z_2$ into $Z_2$
NAND gate

- **Denotation**

\[ x \uparrow y = \overline{xy} \]

Let \( x_1, x_2 \) as input

\[ x_1 \uparrow x_2 = \begin{cases} 
0 & \text{if } x_1 = 1 \text{ and } x_2 = 1 \\
1 & \text{otherwise}
\end{cases} \]

Additional

\[ \overline{x} = \overline{xx} = x \uparrow x \]

\[ x \lor y = \overline{x\overline{y}} = \overline{x} \uparrow \overline{y} = (x \uparrow x) \uparrow (y \uparrow y) \]
Ex1

Design combinatorial circuits using NAND gate to compute the functions
\( f_1(x) = \bar{x} \) and \( f_2(x, y) = x \vee y \)
Ex2.

Consider this table

The disjunctive normal is

\[ f(x, y, z) = xyz \lor xy\bar{z} \lor x\bar{y}\bar{z} \]

The combinatorial circuit is
The half adder

Accept as input two bits $x$ and $y$ and produces as output binary sum $cs$ of $x$ and $y$. The term $cs$ is a two-bit binary number: $s$ is sum and $c$ is carry

By Observation

$$c = xy, \quad s = x \oplus y$$

<table>
<thead>
<tr>
<th>$x$</th>
<th>$y$</th>
<th>$c$</th>
<th>$s$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
The full adder

Accept input three bits \( x, y \) and \( z \)

Outputs \( c, s \)

Binary equations

\[
s = x \oplus y \oplus z
\]

\[
c = xyz \lor xy\bar{z} \lor x\bar{y}z \lor \bar{x}yz
\]

\[= xy \lor x\bar{y}z \lor \bar{x}yz
\]

\[= xy \lor xyz \lor \bar{x}yz
\]

\[= xy \lor xz \lor \bar{x}yz
\]

\[= xy \lor xz \lor xyz \lor \bar{x}yz
\]

\[= xy \lor xz \lor yz
\]

\[= xy \lor z(x \lor y)
\]

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>
3 bits adder