Combinational Circuit Analysis Example

Given this logic circuit we can:

- Find corresponding logic expression from circuit
- Create truth table by applying all input combinations:
  - From truth table find Canonical Sum/Product Representations
  - Manipulate logic expression to other forms using theorems.

Truth Table

<table>
<thead>
<tr>
<th>Row</th>
<th>X</th>
<th>Y</th>
<th>Z</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

From truth table:

- Canonical Sum: $F = \Sigma_{X,Y,Z} (1, 2, 5, 7)$
- Canonical Product: $F = \Pi_{X,Y,Z} (0, 3, 4, 6)$

Corresponding logic expression:

$$F = ((X + Y') \cdot Z) + (X' \cdot Y \cdot Z')$$
Combinational Circuit Analysis Example (continued)

- The previous circuit logic expression $F$ can be transformed into sum of products by multiplying out (Using $T8'$) and written as:

$$F = X \cdot Z + Y'.Z + X'.Y \cdot Z'$$

Realized using a 2-level AND-OR circuit:
Combinational Circuit Analysis Example (continued)

- The logic expression $F$ for the previous circuit can be added out (using T8) and written as:

$$F = ((X+Y').Z) + (X'.Y.Z')$$

$$= (X+Y'+X').(X+Y'+Y).(X+Y'+Z').(Z+X').(Z+Y).(Z+Z')$$

$$= 1.1.(X+Y'+Z').(X'+Z).(Y+Z).1$$

$$F = (X+Y'+Z').(X'+Z).(Y+Z)$$

Realized using 2-level OR-AND circuit.
Equivalent Symbols of NAND, NOR Gates

NAND Symbols

Normal Symbol

\[ X \quad (X \cdot Y)' \]

Alternate NAND Symbol

\[ X \quad Y \quad X' + Y' \]

According to DeMorgan’s theorem T13:

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

NOR Symbols

Normal NOR Symbol

\[ X \quad Y \quad (X + Y)' \]

Alternate NOR Symbol

\[ X \quad Y \quad X' \cdot Y' \]

According to DeMorgan’s theorem T13’:

\[ (X + Y)' = X' \cdot Y' \]
NAND-NAND Logic Circuits for Sum of Products

- A sum of products logic expression can be realized by NAND gates by replacing all AND gates and the OR GATE in the usual realization with NAND gates as follows:

  \[ F = A + B + C + D \ldots \]

  where \( A, B, C, \ldots \) are product terms of the input variables e.g. \( A = x \cdot y \cdot z \)

  \[ F = (A')' + (B')' + (C')' + (D')' + \ldots \quad \text{from T4} \]

  \[ = (A'.B'.C'.D'\ldots)' \quad \text{(from DeMorgan’s theorem T13)} \]

  This is a 2-level NAND representation.
Alternate Sum of Products Realizations
(Applying DeMorgan’s theorem T13 Graphically)

AND-OR

NAND-NAND
NAND-NAND Sum of Products Example

- The sum of products expression

\[
F = X \cdot Z + Y' \cdot Z + X' \cdot Y \cdot Z'
\]

\[
F = ((X \cdot Z)')' + ((Y' \cdot Z)')' + ((X' \cdot Y \cdot Z')')' \quad \text{double negate T4}
\]

\[
F = [(X \cdot Z)' \cdot (Y' \cdot Z)' \cdot (X' \cdot Y \cdot Z')']' \quad \text{DeMorgan’s theorem T13}
\]

Can be realized using the 2-level NAND-NAND circuit:

\[
F = [(X \cdot Z)' + (Y' \cdot Z)' + (X' \cdot Y \cdot Z')']'
\]
NOR-NOR Circuits for Product of Sums

- A product of sums expression can be realized by NOR gates by replacing all the OR gates and the AND gate with NOR gates as follows:

\[ F = A \cdot B \cdot C \cdot D \ldots \]

Where \( A, B, C \) are sum terms of the input variables (e.g. \( A = x+y+z \))

\[ F = (A')' \cdot (B')' \cdot (C')' \cdot (D')' \ldots \quad \text{using T4} \]
\[ = (A' + B' + C' + D' + \ldots) '\]

(using Demorgan’s theorem T13’)

This is a 2-level NOR-NOR representation
Alternate Product of Sums Realizations
(Applying DeMorgan’s theorem T13’ Graphically)

OR-AND

NOR-NOR
Combinational Circuit Synthesis

• An example of a combinational circuit description:
  Create a logic function in 4 input variables $N = N_3N_2N_1N_0$ whose output is 1 only if the input is a prime number.

• This function is 1 when the input $N = 1, 2, 3, 5, 7, 11$ can be written in the canonical sum of products representation as:

$$F = \Sigma_{N_3N_2N_1N_0} (1, 2, 3, 5, 7, 11, 13)$$

$$= N_3'N_2N_1'N_0 + N_3'N_2N_1N_0' + N_3N_2'N_1N_0$$

$$+ N_3'N_2N_1N + N_3N_2N_1N_0 + N_3N_2'N_1N_0 + N_3N_2N_1'N_0$$
A Verbal Synthesis Example:
An Alarm Circuit

• A verbal logic description:
  – The ALARM output is 1 if the panic input is 1, or if the ENABLE input is 1, the EXISTING input is 0, and the house is not secure.
  – The house is secure if the WINDOW, DOOR, GARAGE inputs are all 1

• This can be put in logic expressions as follows:

ALARM = PANIC + ENABLE . EXISTING’ . SECURE’
SECURE = WINDOW . DOOR . GARAGE
ALARM = PANIC + ENABLE . EXISTING’. (WINDOW . DOOR . GARAGE)’

In sum of products form as (by using DeMorgan T13 and multiplying out):
ALARM = PANIC + ENABLE. EXISTING’ . WINDOW’
+ ENABLE . EXISTING’. DOOR’+ ENABLE. EXISTING’. GARAGE’