Chapter (3)

Eng. Mai Z. Alyazji

October, 2016
3.1 Simplify the following Boolean functions, using three-variable maps:

(a) \( F(x, y, z) = \sum(0, 2, 4, 5) \)  
(b) \( F(x, y, z) = \sum(1, 2, 3, 6, 7) \)  
(c) \( F(x, y, z) = \sum(0, 2, 3, 4, 6) \)

Answer:

(a) 
\[
\begin{array}{c|c|c|c}
  & 0 & 1 & 11 & 10 \\
\hline 
x \y & 1 & 1 & 1 & 0 \\
\end{array}
\]

\( F = y'z' + xy' + x'z' \)

(b) 
\[
\begin{array}{c|c|c|c|c}
  & 00 & 01 & 11 & 10 \\
\hline 
x \y & 1 & 1 & 1 & 1 \\
\end{array}
\]

\( F = x'z + y \)

(c) 
\[
\begin{array}{c|c|c|c|c}
  & 00 & 01 & 11 & 10 \\
\hline 
x \y & 1 & 1 & 1 & 1 \\
\end{array}
\]

\( F = x'y + z' \)
3.5 Simplify the following Boolean functions, using four-variable maps:

(a) \( F(w, x, y, z) = \sum(1, 4, 5, 6, 12, 14, 15) \)

(b) \( F(A, B, C, D) = \sum(0, 2, 4, 5, 6, 7, 8, 10, 13, 15) \)

Answer:

a)

\[
F = xz' + w'y'z + wxy
\]

b)

\[
F = B'D' + A'B + BD
\]
3.8 Find the minterms of the following Boolean expressions by first plotting each function in a map:

(a) wyz + w'x' + wxz'

Answer:

As we remember in four-literals map:

One square represents one minterm, giving a term with four literals.
Two adjacent squares represent a term with three literals.
Four adjacent squares represent a term with two literals.
Eight adjacent squares represent a term with one literal.
Sixteen adjacent squares produce a function that is always equal to 1.

Thus, three literals wyz are two adjacent squares and so on...

\[
\begin{array}{c|cccc}
  & 00 & 01 & 11 & 10 \\
\hline
w & \\ & 1 & 1 & 1 & 1 \\
 00 & 1 & 1 & 1 & 1 \\
 01 & \text{\color{white}1} & \text{\color{white}1} & \text{\color{white}1} & \text{\color{white}1} \\
11 & 1 & 1 & 1 & 1 \\
10 & \text{\color{white}1} & \text{\color{white}1} & \text{\color{white}1} & \text{\color{white}1} \\
\end{array}
\]
3.9 Find all the prime implicants for the following Boolean functions, and determine which are essential:

(a) \( F(w, x, y, z) = \Sigma(0, 2, 4, 5, 6, 7, 8, 10, 13, 15) \)

Answer:

**Note:** A prime implicant is a product term obtained by combining the maximum possible number of adjacent squares in the map. If a minterm in a square is covered by only one prime implicant, that prime implicant is said to be essential.

3.12 Simplify the following Boolean functions:

(a) \( F(A, B, C, D) = \Sigma(1, 3, 5, 7, 13, 15) \)

\[ F = (A + D')(B' + D') \]
3.15 Simplify the following Boolean function \( F \), together with the don't-care conditions \( d \), and then express the simplified function in sum-of-minterms form:

a) \( F(x, y, z) = \sum(0, 1, 4, 5, 6) \)

\[ D(x, y, z) = \sum(2, 3, 7) \]

b) \( F(x, y, z) = \sum(4, 12, 7, 2, 10) \)

\[ d \text{-} \text{car} \text{e} \text{-} \text{cond} \text{i} \text{t} \text{i} \text{o} \text{n} \text{s} \]

\[ d(A,B,C,D) = \sum(0,6,8) \]

**Answer:**

a) 

\[
\begin{array}{ccc|ccc|cc}
  & y & z & 00 & 01 & 11 & 10 \\
 0 &   &   & 1 & 1 & X & X \\
 1 &   &   & 1 & 1 & X & 1 \\
\end{array}
\]

\( F = 1 \)

b) 

\[
\begin{array}{ccc|ccc|cc}
  & A & B & 00 & 01 & 11 & 10 \\
 00 & X &   &   &   &   & 1 \\
 01 &   &   & 1 & 1 & X &   \\
 11 &   &   & 1 &   &   &   \\
 10 & X &   &   &   &   & 1 \\
\end{array}
\]

\( F = C'D' + B'D' + A'BC \)
3.16 Simplify the following functions, and implement them with two-level NAND gate circuits:

(a) \( F(A, B, C, D) = AC'D' + A'C + AB'C + A'C'D' \)

Answer:

\[
\begin{array}{cccc}
\text{CD} & 00 & 01 & 11 & 10 \\
\text{AB} & 00 & 01 & & & \\
00 & 1 & 1 & & \\
01 & 1 & 1 & & \\
10 & 1 & 1 & & \\
11 & 1 & 1 & & \\
\end{array}
\]

\( F = D' + C = (DC)' \)
3.20 Draw the multiple-level NOR circuit for the following expression:

\[ CD(B + C)A + (BC' + DE') \]

Answer:

\[
(\overline{CD(B+C)A + (BC' + DE')})
\]

\[
( (C' + D')'(B + C) A + (((B' + C)' + (D' + E))') )
\]

\[
( ((C' + D')'' + (B + C)' + A)' + ((B' + C)' + (D' + E)') )
\]

\[
( ((C' + D') + (B + C)' + A)' + (B' + C)' + (D' + E)' )
\]
3.23 Implement the following Boolean function F, together with the don't-care conditions d, using no more than two NOR gates:

\[ F(A, B, C, D) = \sum(2, 4, 10, 12, 14) \]

\[ d(A, B, C, D) = \sum(0, 1, 5, 8) \]

Assume that both the normal and complement inputs are available.

Answer:

\[ F = A'C' + B'D' + AD' \]

\[ F' = D + A'BC \]

\[ F = (D + A'BC)' = (D + (A + B' + C'))' \]
3.24 Implement the following Boolean function $F$, using the two-level forms of logic

(a) NAND-AND, (b) AND-NOR, (c) OR-NAND, and (d) NOR-OR:

$$F(A, B, C, D) = \Sigma(0, 4, 8, 9, 10, 11, 12, 14)$$

Answer:

F = C'D' + AB' + AD'

a) $F = ((C'D')(AB')(AD'))'$
b) $F' = (C'D' + AB' + AD')'$

c) $F = (C + D)' + (A' + B)' + (A' + D)' = ((C+D)(A'+B)(A'+D))'$

d) $F = (C + D)' + (A' + B)' + (A' + D)'$
3.28 Derive the circuits for a three-bit parity generator and four-bit parity checker using an odd parity bit.

Answer:

3.29 Implement the following four Boolean expressions with three half adders:

\[
\begin{align*}
D &= A \oplus B \oplus C \\
E &= A'BC + AB'C \\
F &= ABC' + (A' + B')C \\
G &= ABC
\end{align*}
\]

Answer: