Problem Set – 1 Answers

CS 230, Spring 2023

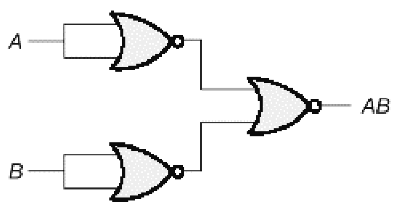
Answers

1. (a), (b), (c)

Can be verified from truth table.

2. (3)

$$f = \overline{\bar{A} + \overline{\left[B + \bar{C}\left(\overline{AB + A\bar{C}}\right)\right]}}$$
$$f = A \cdot [B + \bar{C}(\overline{AB + A\bar{C}})]$$
$$f = A \cdot [B + \bar{C}(\bar{A} + \bar{B})(\bar{A} + C)]$$
$$f = A[B + \bar{C}(\bar{A} + \bar{B}C)]$$
$$f = A[B + \bar{A}\bar{C}]$$
$$f = AB$$



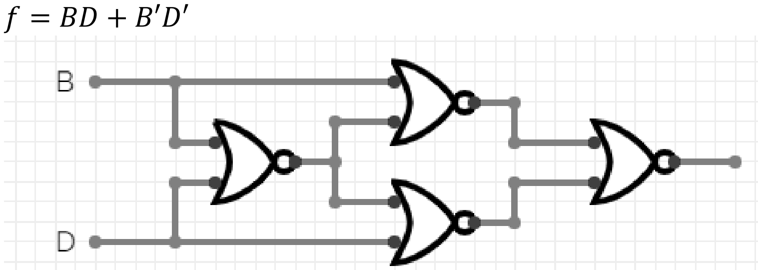
3. (4)

AB \ CD	00	01	11	10
00	1	0	0	1
01	0	1	1	0
11	0	1	1	0
10	1	0	0	1

BD

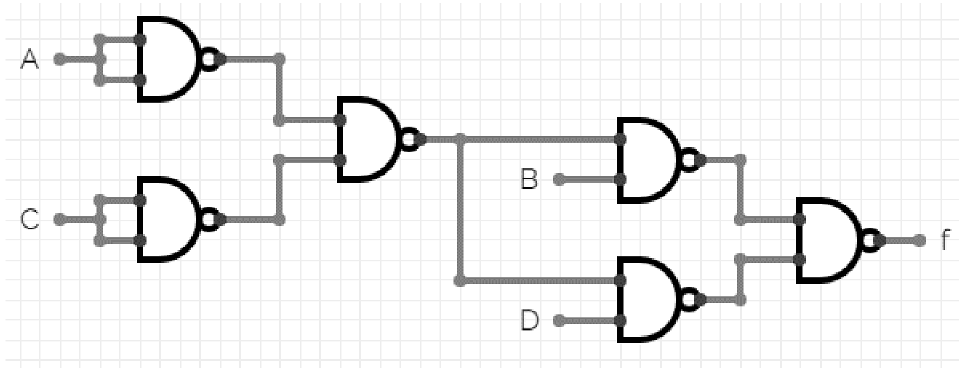
$B'D'$

$$f = BD + B'D'$$



4. (6)

$f = (A + C)(B + D) = B(A + C) + D(A + C) = B \cdot x + D \cdot x$
For x, it takes 3 NAND gates.



5. $2^{2^{n-1}}$

Without the loss in generality say there exists a Boolean function $f = A * B$ where $*$ is an unknown operation.
Now, $f^d = A *^d B$ where $*^d$ is the dual of $*$
And, $f^c = A' *^c B'$ where $*^c$ is the complement of $*$
We know that, $*^d = *^c$ and $A \neq A'$ or $B \neq B'$
Hence, $A = B'$ and $A' = B$ as the Boolean operations are commutative.
And if A and B are further combinations of Boolean operations, we can recursively perform this argument.

So, to have $f^d = f^c$ we need to have mutually exclusive complement pairs of values.
i.e., for 3 literals the pairs will be (0,7), (1,6), (2,5) and (3,4).
So, in n literals system, 2^{n-1} such pairs exist.

We have a choice to keep or discard each such pair. Hence, $2^{2^{n-1}}$.

6. (c)

$M = \overline{XYZ + \bar{X}\bar{Y}} + YZ$
Dual of M,
$M^d = \overline{(X + Y + Z) \cdot (\bar{X} + \bar{Y})} \cdot (Y + Z)$
$M^d = \left[\overline{(X + Y + Z)} + \overline{(\bar{X} + \bar{Y})}\right] \cdot (Y + Z)$
Compliment of M,
$M^c = \overline{\overline{(XYZ + \bar{X}\bar{Y})} + YZ}$
$M^c = \overline{\overline{(XYZ + \bar{X}\bar{Y})}} \cdot \overline{YZ}$
$M^c = (XYZ + \bar{X}\bar{Y}) \cdot \overline{YZ}$

7. (6)

ab \ cd	00	01	11	10
00	1	1	0	0
01	0	0	1	1
11	0	1	1	1
10	1	1	1	0

From the 3 squares, we get $bc, ad, b'c'$
Hence, $f = ad + b'c' + bc$

8. $\overline{\boldsymbol{B}} \cdot \overline{\boldsymbol{D}} + \overline{\boldsymbol{A}} \cdot \overline{\boldsymbol{D}}$

CD \ AB	00	01	11	10
00	1	1		1
01	X			
11	X			
10	1	1		X

From the Kmap, we can see that the blue square encompasses $A'D'$.
Now, as X is a don't care so the value of X doesn't change the outcome of the function. That means, even if X is 0, we can consider it to be 1 and complete our minimization for convenience.
Hence, from the red square we get $B'D'$.
$f = A'D' + B'D'$