Q1.

Draw truth tables for each of the following logical functions of 3 Boolean variables:

(a) $z = f(a, b, c) = a + \overline{b} + \overline{c}$

Solution: There should be a 1 in any row where $a = 1$, as well as any row where $b = 0$ or where $c = 0$:

a	b	c	z
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

(b) $z = f(a, b, c) = \overline{a}(b + \overline{c})$

Solution:

a	b	c	z
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

When building a truth table like the one above, it can be helpful to add intermediate columns that show the truth values of individual parts of the overall Boolean expression:

a	b	c	$\overline{a}$	$\overline{c}$	$(b + \overline{c})$	z
0	0	0	1	1	1	1
0	0	1	1	0	0	0
0	1	0	1	1	1	1
0	1	1	1	0	1	1
1	0	0	0	1	1	0
1	0	1	0	0	0	0
1	1	0	0	1	1	0
1	1	1	0	0	1	0

(c) $z = f(a, b, c) = \overline{a}b\overline{c} + \overline{a}\overline{b}\overline{c} + \overline{a}bc$

Solution:

a	b	c	z
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

Note that this is the same truth table for $z = \overline{a}(b + \overline{c})$.

$$
\begin{aligned}
z &= \overline{a}b\overline{c} + \overline{a}\overline{b}\overline{c} + \overline{a}bc \\
&= \overline{a}((b\overline{c}) + (\overline{b}\overline{c}) + (bc)) \\
&= \overline{a}((b + \overline{b})\overline{c} + (bc)) \\
&= \overline{a}(\overline{c} + (bc)) \\
&= \overline{a}((\overline{c} + b)(\overline{c} + c)) \\
&= \overline{a}(\overline{c} + b)
\end{aligned}
$$

(d) $w = f(s, b, i) = i + (s\overline{b})$, where

- w is 1 if a warning light should be on
- s is 1 if the speed is greater than 0
- b is 1 if the seatbelt is buckled
- i is 1 if the ignition key has been turned in the last 5 seconds

Solution:

i	b	s	w
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

This answer is easy to check with a little Python:

```
>>> def bit_not(b):
...   return 1 if b == 0 else 0
...
>>> for i in (0,1):
...   for b in (0,1):
...     for s in (0,1):
...       w = i or (s and (bit_not(b)))
```

```
...         print("{} {} {} | {}".format(i, b, s, w))
...
0 0 0 | 0
0 0 1 | 1
0 1 0 | 0
0 1 1 | 0
1 0 0 | 1
1 0 1 | 1
1 1 0 | 1
1 1 1 | 1
```

Note that the built-in Python `and` and `or` operators work with 1 and 0, but that we need to define our own version of `not` to get a truth table made only of 1's and 0's.

Q2.

Show that the two circuits in Fig. 1 are equivalent by drawing truth tables for each circuit. [Storey, Electronics: A Systems Approach, 5th Ed., Pearson]

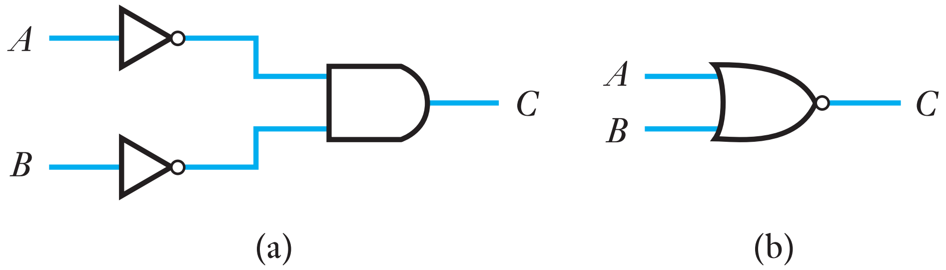


Figure 1: Equivalent logic circuits

Solution: Truth table for Fig. 1(a):

A	B	$\overline{A}$	$\overline{B}$	$C = \overline{A} \cdot \overline{B}$
0	0	1	1	1
0	1	1	0	0
1	0	0	1	0
1	1	0	0	0

Truth table for Fig. 1(b):

A	B	$A + B$	$C = \overline{A + B}$
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

Q3.

Given that A is a Boolean variable, evaluate and then simplify the following expressions: [Storey]

- $A \cdot 1$
- $A \cdot \overline{A}$
- $1 + A$
- $A + \overline{A}$
- $1 \cdot 0$
- $1 + 0$

Solution:

- $A \cdot 1 = A$ (Identity)
- $A \cdot \overline{A} = 0$ (And-Complement)
- $1 + A = 1$ (Annihilation)
- $A + \overline{A} = 1$ (Or-Complement)
- $1 \cdot 0 = 0$ (Annihilation)
- $1 + 0 = 1$ (Identity)

Q4.

Your team lead has handed you the logic circuit design (part of a test system for a digital camera) shown in Fig. 2 to analyse and simplify.

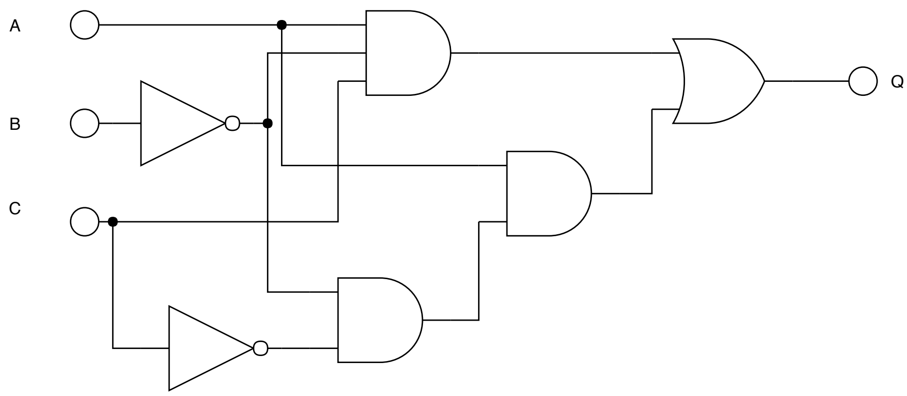


Figure 2: A logic circuit

(a) Write a Boolean expression for the output of this logic circuit in terms of its inputs.

Solution: $Q = A\overline{B}C + A(\overline{B}\,\overline{C})$

You can arrive at this in various ways. I like to work my way back from the output (Q, in this case) towards the inputs:

1. The first gate I hit is an OR, so I know that $Q = ?+?$.
2. Looking at one branch of the OR, I find a 3-input AND gate, with inputs A, $\overline{B}$, and C. So now I know that $Q = A\overline{B}C + ?$.
3. Looking at the other branch of the OR, I see A ANDed with the output of another AND gate that combines $\overline{B}$ and $\overline{C}$. So $Q = A\overline{B}C + A(\overline{B}\,\overline{C}) = A\overline{B}C + A\overline{B}\,\overline{C}$

(b) Apply the rules of Boolean algebra to find a simpler expression for Q.

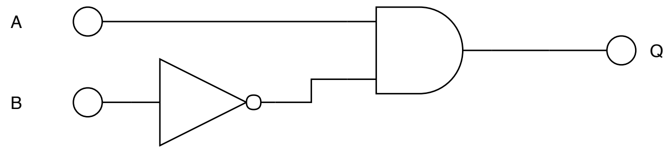
Solution: Starting from the expression found in the previous part, we can do an algebraic simplification (factor out common terms, and use the fact that a logic variable ORed with its inverse is always TRUE):

$$\begin{aligned} Q &= A\overline{B}C + A(\overline{B}\,\overline{C}) \\ &= A(\overline{B}C + \overline{B}\,\overline{C}) \\ &= A(\overline{B}(C + \overline{C})) \\ &= A\overline{B} \end{aligned}$$

(c) Draw a logic circuit which implements the simplified version of the design.

TURN OVER

Solution:



END OF QUESTIONS