Solutions to Week 3 Digital Logic & Boolean Algebra

Question 1

Using Truth Tables prove whether the following Boolean equalities are true:

1. /(A + B) = /A . /B

   Answer:

   Example Using a Truth Table approach, note this is one of the very important **de Morgan's theorems:**

A	B	A + B	**/(A+B)**	/A	/B	**/A . /B**
0	0	0	**1**	1	1	**1**
0	1	1	**0**	1	0	**0**
1	0	1	**0**	0	1	**0**
1	1	1	**0**	0	0	**0**

We can see it clearly that their last columns are the **SAME** so the original equation is **true.**

2. /(A.B) = /A . /B

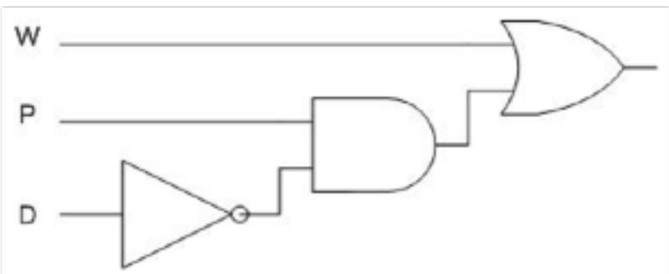
   Answer:

A	B	A · B	**/(A·B)**	/A	/B	**/A . /B**
0	0	0	**1**	1	1	**1**
0	1	0	**1**	1	0	**0**
1	0	0	**1**	0	1	**0**
1	1	1	**0**	0	0	**0**

We can see it clearly that their last columns are **DIFFERENT** so the original equation is **false.** But note that the other de Morgan Law is **/(A . B) = /A + /B**

Question 2

Consider the following logic diagram with switches W, P and D:



1. Identify the three gates being used in the diagram.
   Answer: NOT, AND, OR gates

2. Write the logic expression that is equivalent to the diagram. (It need not be simplified.)
   Answer: The expression is /D. P + W

3. Complete a truth table that shows the output (X) for inputs (W, P and D).
   Answer:

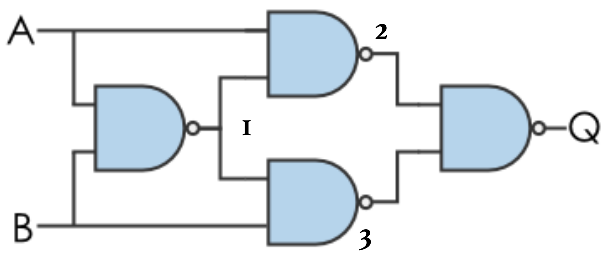
W	P	D	/D	/D . P	X = /D . P + W
0	0	0	1	0	0
0	0	1	0	0	0
0	1	0	1	1	1
0	1	1	0	0	0
1	0	0	1	0	1
1	0	1	0	0	1
1	1	0	1	1	1
1	1	1	0	0	1

Note that as long as switch W is set to 1, the output will always be 1, because of the property of the OR gate. This is sometimes referred to as "do not cares", but it is always wise to complete the truth table none the less.

4. Using LogiSim (or DigitalWorks or alternative software) draw the logic diagram and verify the truth table in 3 by placing a lamp at the final output.

Question 3

Consider the following logic diagram composed of NAND gates.



1. Write a truth table that shows the output (Q) for inputs (A and B).

Answer:

A	B	1 = /(A.B)	2 = /(A.1)	3=/(1.B)	Q=/(2.3)
0	0	1	1	1	0
0	1	1	1	0	1
1	0	1	0	1	1
1	1	0	1	1	0

2. Which logic gate is this equivalent to?

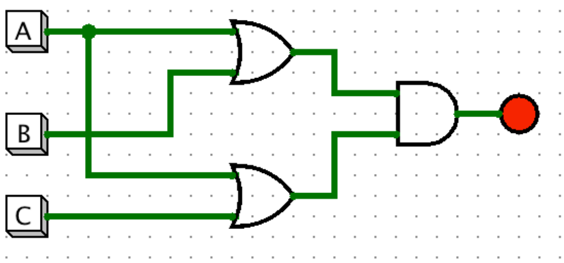
   Answer: The logic is the same as for an XOR gate. The output of an XOR gate is true only when **exactly one** of its inputs is true.

Question 4

Draw the logic diagrams for each of the following Boolean expressions.

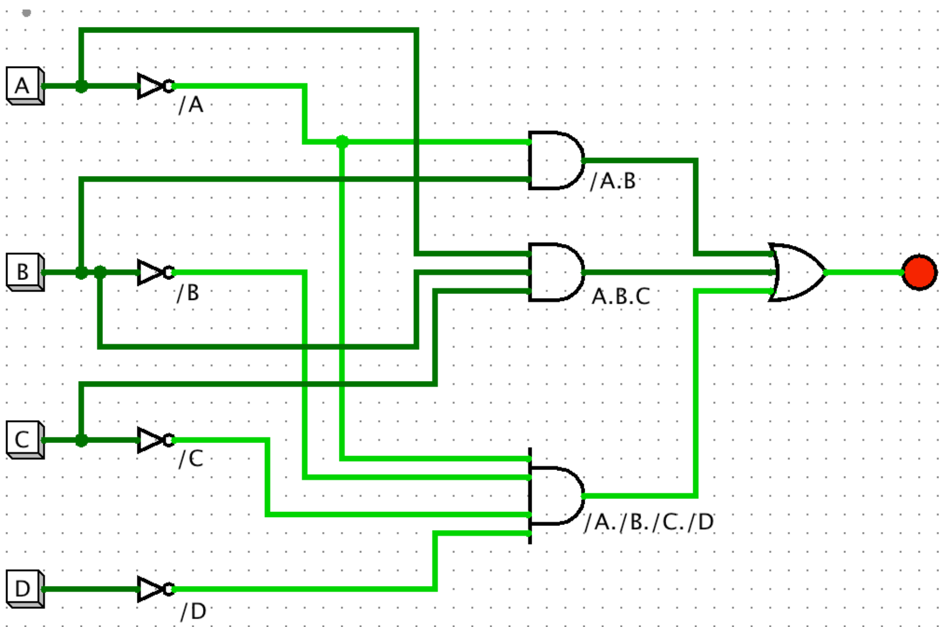
1. (A+B).(A+C)

   Answer:

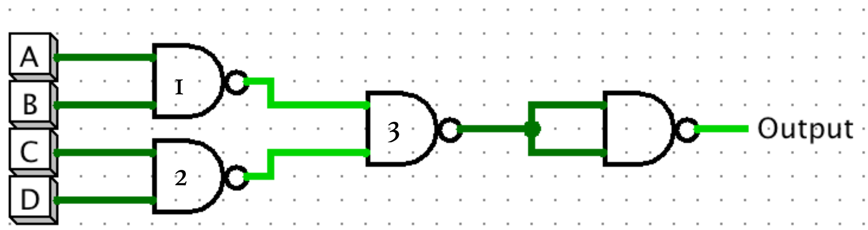


2. /A.B +A.B.C + /A./B./C./D

   Answer:



Question 5



By hand write down

1. the output of each gate, and
2. the final output for the given digital logic diagram.

Boolean Expression	Description
A + 1 = 1	A in parallel with closed = "CLOSED"
A + 0 = A	A in parallel with open = "A"
A . 1 = A	A in series with closed = "A"
A . 0 = 0	A in series with open = "OPEN"
A + A = A	A in parallel with A = "A"
A . A = A	A in series with A = "A"
NOT $\overline{A}$ = A	NOT NOT A (double negative) = "A"
A + $\overline{A}$ = 1	A in parallel with NOT A = "CLOSED"
A . $\overline{A}$ = 0	A in series with NOT A = "OPEN"
A+B = B+A	A in parallel with B = B in parallel with A
A.B = B.A	A in series with B = B in series with A
$\overline{A+B}$ = $\overline{A}$.$\overline{B}$	invert and replace OR with AND
$\overline{A.B}$ = $\overline{A}$+$\overline{B}$	invert and replace AND with OR

Answer:
1 = /(A.B)
2 = /(C.D)
3 = /(1.2) = /(/(A.B) . /(C.D))
Output = /(3.3) = /(/(/(A.B) . /(C.D)) . /(/(A.B) . /(C.D)))

According to the Laws of Boolean

- X.X = X (A variable AND'ed with itself is always equal to the variable.)
- //X = X (A double complement of a variable is always equal to the variable.)

Set X = /(/(A.B) . /(C.D))

Final answer
= /(/(/(A.B)./(C.D)) . /(/(A.B)./(C.D)))
= /(/(/(A.B) . /(C.D)))
= /(A.B) . /(C.D)