Montogue

Quiz EL302
Digital Logic

Lucas Monteiro Nogueira

PROBLEMS

Problem 1 (Modified from Vahid, 2011)

Problem 1.1: Evaluate the Boolean equation `F = (a AND b) OR c OR d` for the given values of variables `a`, `b`, `c`, and `d`:

1. `a=1, b=1, c=1, d=0`

2. `a=0, b=1, c=1, d=0`

3. `a=1, b=1, c=0, d=0`

4. `a=1, b=0, c=0, d=0`

Problem 1.2: Evaluate the Boolean equation `F = a AND (b OR c) AND d` for the given values of variables `a`, `b`, `c`, and `d`:

1. `a=1, b=1, c=0, d=1`

2. `a=0, b=0, c=0, d=1`

3. `a=1, b=0, c=0, d=0`

4. `a=1, b=0, c=1, d=1`

Problem 1.3: Evaluate the Boolean equation `F = a AND (b OR (c AND d))` for the given values of variables `a`, `b`, `c`, and `d`:

1. `a=1, b=1, c=0, d=1`

2. `a=0, b=0, c=0, d=1`

3. `a=1, b=0, c=0, d=0`

4. `a=1, b=0, c=1, d=1`

Problem 2 (Agarwal and Lang, 2005, w/ permission)

Write a Boolean expression for the following statement: ”Z is TRUE if either *X* or *Y* is FALSE, otherwise *Z* is FALSE.” Write a truth table for this expression.

Problem 3 (Modified from Sedha, 2004)

12 Basic rules of Boolean algebra

1. $A.0 = 0$	7. $A.A = A$
2. $A.1 = A$	8. $A.\bar{A} = 0$
3. $A + 0 = A$	9. $\bar{\bar{A}} = A$
4. $A + 1 = 1$	10. $A + AB = A$
5. $A + A = A$	11. $A + \bar{A}B = A + B$
6. $A + \bar{A} = 1$	12. $(A + B)(A + C) = A + BC$

Using the rules of Boolean algebra, simplify the following logic equations.

3.1. $Z = (A + \bar{B})(A + B)$

3.2. $Z = AB + A(B + C) + B(B + C)$

3.3. $Z = [A\bar{B}(C + BD) + \bar{A}\bar{B}]C$

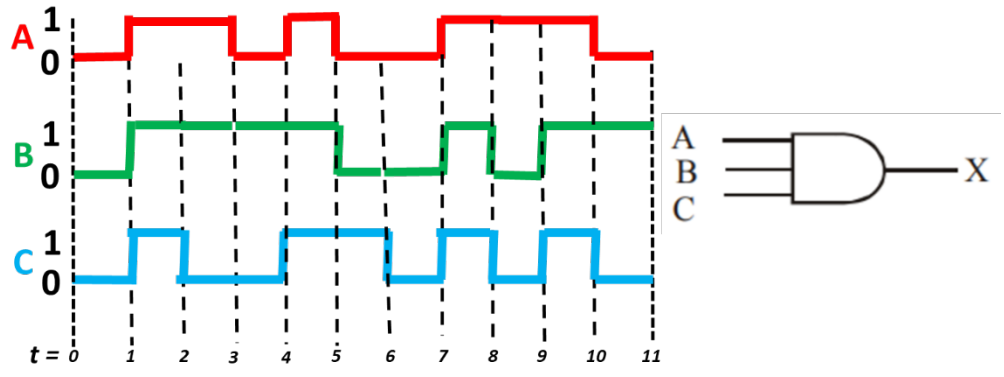
3.4. $Z = \bar{A}BC + A\bar{B}\bar{C} + \bar{A}\bar{B}\bar{C} + A\bar{B}C + ABC$

3.5. $Z = AB + \overline{A\bar{C}} + A\bar{B}C(AB + C)$

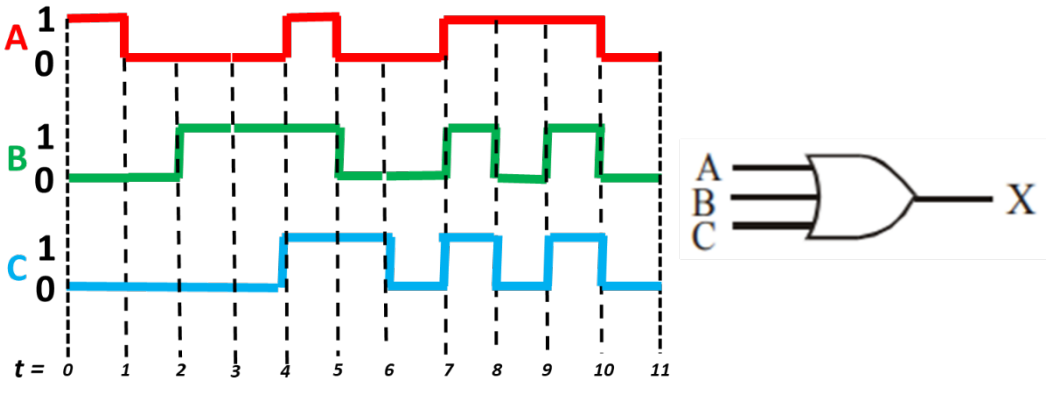
3.6. $Z = (B + BC)(B + \bar{B}C)(B + D)$

Problem 4 (Modified from Sedha, 2004)

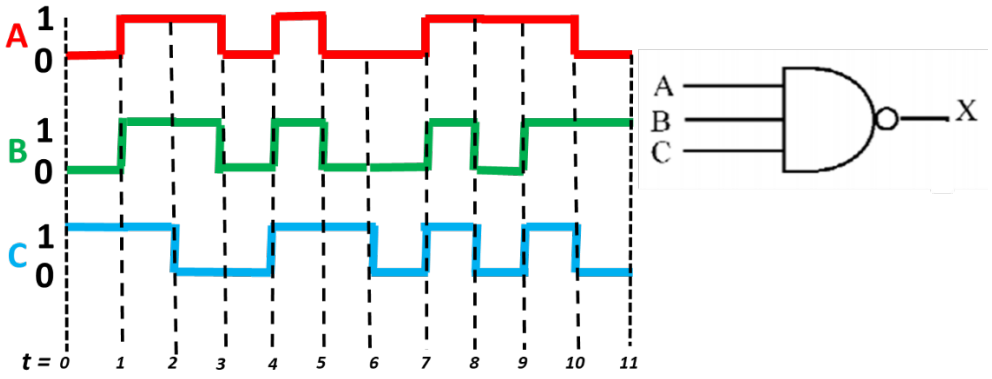
Problem 4.1: The timing diagram shows the three input waveforms *A*, *B*, and *C* applied to an AND gate. Sketch the resulting waveform at *X* from *t* = 0 to *t* = 11.



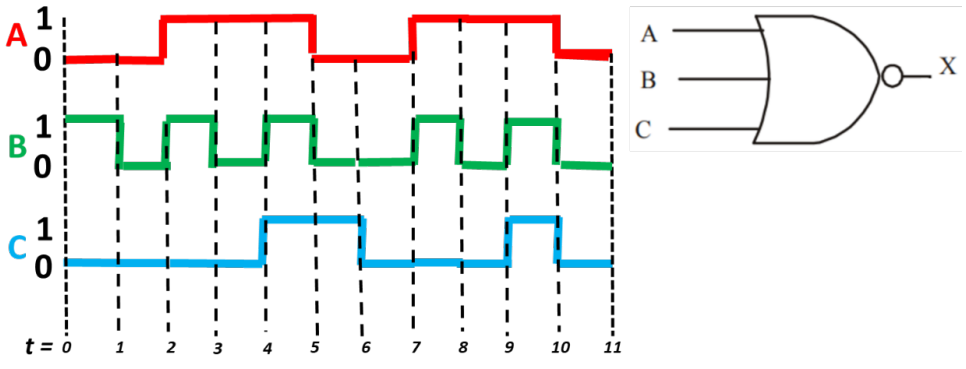
Problem 4.2: The timing diagram shows the three input waveforms *A*, *B*, and *C* applied to an OR gate. Sketch the resulting waveform at *X* from *t* = 0 to *t* = 11.



Problem 4.3: The timing diagram shows the three input waveforms *A*, *B*, and *C* applied to an NAND gate. Sketch the resulting waveform at *X* from *t* = 0 to *t* = 11.



Problem 4.4: The timing diagram shows the three input waveforms *A*, *B*, and *C* applied to an NOR gate. Sketch the resulting waveform at *X* from *t* = 0 to *t* = 11.



Problem 5 (Agarwal and Lang, 2005, w/ permission)

Consider the following four logic expressions.

1. $(A + \bar{B})(\bar{A} \cdot \bar{B} + C) + \overline{CD}$

2. $(A \cdot \bar{C} + \overline{BD})\overline{(D + \bar{B} + A)}$

3. $A + \overline{\bar{B} \cdot D} + A \cdot C \cdot \bar{D}$

4. $\overline{\left(A + \bar{\bar{C}} + B + \bar{D}\right)} + A \cdot \bar{C} \cdot D$

Problem 5.1: Give an implementation using gates for each of the logic expressions above.

Problem 5.2: Write the truth table for each of the four expressions.

Problem 5.3: Suppose you know that $A = 0$. Simplify the four expressions under this constant.

Problem 5.4: Simplify the four expressions assuming that A and B are related as $A = \bar{B}$.

Problem 6 (Modified from Vahid, 2011)

Convert each of the following equations directly to gate-level circuits.

Problem 6.1: `F = ab'+ bc + c'`

Problem 6.2: `F = ab + b'c'd'`

Problem 6.3: `F = ((a + b') * (c' + d)) + (c + d + e')`

Problem 6.4: `F = a'b' + b'c`

Problem 6.5: `F = ab + bc + cd + de`

Problem 6.6: `F = ((ab)' + c) + (d + ef)'`

Problem 7

Convert the following Boolean functions to a digital circuit:

Problem 7.1: `F(a,b,c)= c'`

Problem 7.2: `F(a,b,c)= a'b`

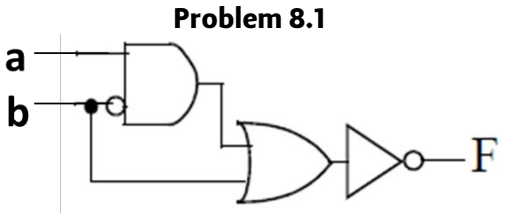
Problem 7.3: `F(a,b,c)= a'bc + ab`

Problem 7.4: `F(a,b,c)= abc + ab + a + b + c`

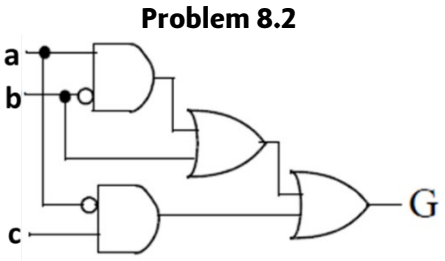
Problem 8

Create Boolean equation representations for the two logic gate diagrams illustrated below.

Problem 8.1



Problem 8.2



Problem 9 (Modified from Vahid, 2011)

Problems 9.1 to 9.4: Draw up truth tables for the four Boolean functions introduced in Problem 7.

Convert each of the following Boolean equations to a truth table.

Problem 9.5: `F(a,b,c) = a' + bc'`

Problem 9.6: `F(a,b,c) = (ab)' + ac' + bc`

Problem 9.7: `F(a,b,c) = ab + ac + ab'c' + c'`

Problem 9.8: `F(a,b,c,d) = a'bc + d'`

Problem 9.9: Draw up a truth table for the circuit introduced in Problem 8.1.

Problem 9.10: Draw up a truth table for the circuit introduced in Problem 8.2.

Problem 10 (Modified from Vahid, 2011)

Problem 10.1: Boolean function `F` corresponds to the following truth table. Draw a logic circuit for `F`.

Inputs			Output
a	b	c	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Problem 10.2: Boolean function `F` corresponds to the following truth table. Draw a logic circuit for `F`.

Inputs			Output
a	b	c	F
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

Problem 10.3: Boolean function `F` corresponds to the following truth table. Draw a logic circuit for `F`.

Inputs			Output
a	b	c	F
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

Problem 11 (Agarwal and Lang, 2005, w/ permission)

Consider the statement: "*Z* is TRUE if at least two of *W*, *X*, and *Y* are TRUE, otherwise *Z* is FALSE."

Problem 11.1: Write a Boolean expression for the above statement.

Problem 11.2: Write a truth table for the function *Z*.

Problem 11.3: Implement *Z* using only AND, OR, and NOT gates. The inputs *W*, *X*, and *Y* are available. Each gate may have an arbitrary number of inputs.

> *Hint*: A sum-of-products representation of the Boolean expression will facilitate this implementation.

Problem 11.4: Implement *Z* using only AND, OR, and NOT gates. Each gate may have no more than two inputs. As before, the inputs *W*, *X*, and *Y* are available.

Problem 11.5: Implement *Z* using only NAND and NOR gates.

> *Hint*: A NAND gate or a NOR gate with its inputs tied together behaves like an inverter.

Problem 11.6: Implement *Z* using only NAND gates.

> *Hint*: Use De Morgan's laws.

Problem 11.7: Implement *Z* using only NOR gates.

> *Hint*: Use De Morgan's laws.

Problem 11.8: Repeat Part 4 and attempt to minimize the number of gates used.

Problem 12 (Agarwal and Lang, 2005, w/ permission)

Consider the Boolean functions $F(A, B, C)$ and $G(A, B, C)$ specified in the following truth table.

A	B	C	$F(A,B,C)$	$G(A,B,C)$
0	0	0	1	0
0	0	1	0	0
0	1	0	0	0
0	1	1	0	1
1	0	0	1	0
1	0	1	1	1
1	1	0	0	1
1	1	1	1	1

Problem 12.1: Write a logic expression corresponding to the functions $F(A, B, C)$ and $G(A, B, C)$.

Problem 12.2: Implement $F(A, B, C)$ with logic gates.

Problem 12.3: Implement $F(A, B, C)$ using only 2-input gates.

Problem 12.4: Implement $F(A, B, C)$ using only 2-input NAND gates.

> *Hint*: Use De Morgan's laws.

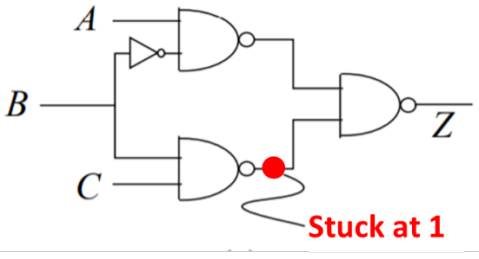
Problem 12.5: Repeat Problems 12.2 through 12.4 for the function $G(A, B, C)$.

Problem 13 (Modified from Agarwal and Lang, 2005, w/ permission)

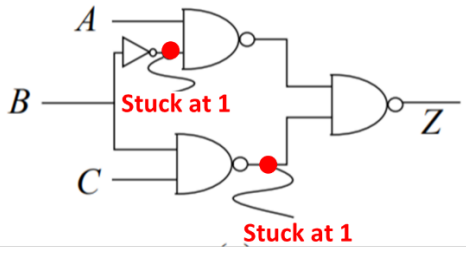
Many manufacturing flaws in digital circuits can be modeled as stuck-at faults. The output of a gate is said to suffer from a stuck-at 1 fault if the output is a 1 irrespective of its input values. Similarly, a stuck-at 0 fault at an output causes the output to produce a 0 at all times. Consider the circuits illustrated below with one or more faults. Write an expression for each of the circuits in terms of the input variables for the given faults.

Hint: As an example, the output of the faulty circuit in Problem 13.1 will be independent of the variable *C*.

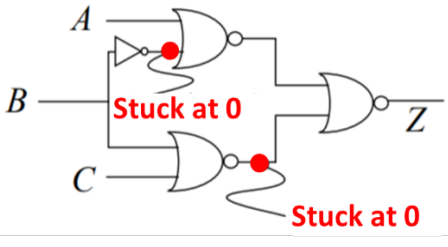
Problem 13.1



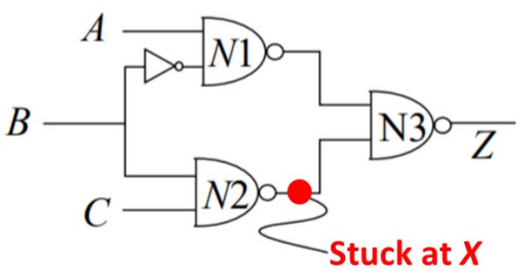
Problem 13.2



Problem 13.3



Problem 13.4: Suppose we were given the faulty circuit illustrated below where the input of NAND gate *N2* is known to have a stuck-at fault. However, we do not know whether it is a stuck-at-1 fault or a stuck-at-0 fault. Further, suppose that we have access to inputs *A*, *B*, and *C*, and the output *Z*. In other words, we are unable to directly observe the output *X* of the faulty NAND gate *N2*. How would you go about determining whether *N2* suffers from a stuck-at 1 fault or a stuck-at 0 fault?



SOLUTIONS

P.1 ➡ Solution

Problem 1.1: The Boolean equations can be developed as follows. Don't forget to begin the solution with the operation in parentheses.

1. F = (1 AND 1) OR 1 OR 0 = 1 OR 1 OR 0 = 1
2. F = (0 AND 1) OR 1 OR 0 = 0 OR 1 OR 0 = 1
3. F = (1 AND 1) OR 0 OR 0 = 1 OR 0 OR 0 = 1
4. F = (1 AND 0) OR 0 OR 0 = 0 OR 0 OR 0 = 0

Problem 1.2: The Boolean equations can be developed as follows.

1. F = 1 AND (1 OR 0) AND 1 = 1 AND 1 AND 1 = 1
2. F = 0 AND (0 OR 0) AND 1 = 0 AND 0 AND 1 = 0
3. F = 1 AND (0 OR 0) AND 0 = 1 AND 0 AND 0 = 0
4. F = 1 AND (0 OR 1) AND 1 = 1 AND 1 AND 1 = 1

Problem 1.3: The Boolean equations can be developed as follows.

1. F = 1 AND (1 OR (0 AND 1)) = 1 AND (1 OR 0) = 1 AND 1 = 1
2. F = 0 AND (0 OR (0 AND 1)) = 0 AND (0 OR 0) = 0 AND 0 = 0
3. F = 1 AND (0 OR (0 AND 0)) = 1 AND (0 OR 0) = 1 AND 0 = 0
4. F = 1 AND (0 OR (1 AND 1)) = 1 AND (0 OR 1) = 1 AND 1 = 1

P.2 ➡ Solution

The statement can be represented with the simple Boolean expression

$$Z = \overline{X} + \overline{Y}$$

The truth table is shown below.

X	Y	Z
0	0	0
0	1	0
1	0	0
1	1	1

P.3 ➡ *Solution*

Problem 3.1: We first expand the expression with the distributive property,

$$Z = \left(A + \overline{B}\right)\left(A + B\right) = AA + AB + \overline{B}A + \overline{B}B$$

By rule 7, the term in blue can be simplified as A. By rule 8, the term in red yields zero. Accordingly,

$$Z = A + AB + \overline{B}A + 0$$
$$\therefore Z = A\left(1 + B + \overline{B}\right)$$

By rule 6, the term in green equals 1. It follows that

$$Z = A\left(1 + 1\right) = A.1$$

By rule 2, $A.1 = A$, which leads to the final simplification

$$\boxed{Z = A}$$

Problem 3.2: We first expand the equations using the distributive property,

$$Z = AB + A\left(B + C\right) + B\left(B + C\right) = AB + AB + AC + BB + BC$$

Rule 5 states that a Boolean sum of two identical terms, as highlighted in blue, equals the term itself,

$$Z = AB + AC + BB + BC$$

By rule 7, the term in red is such that $B.B = B$, giving

$$Z = AB + AC + B + BC$$

Restating the blue term as $AB = BA$ and adjusting, we write

$$Z = B + BA + AC + BC$$

By rule 10, $B + BA = B$, so that

$$Z = B + AC + BC$$

Adjusting again,

$$Z = B + BC + AC$$

Applying rule 10 to the purple term,

$$\boxed{Z = B + AC}$$

Problem 3.3: We first expand the term in parentheses,

$$Z = \left[A\overline{B}\left(C + BD\right) + \overline{A}\overline{B}\right]C = \left(A\overline{B}C + A\overline{B}BD + \overline{A}\overline{B}\right)C$$

By rule 8, the operation in blue yields zero. Further, ANDing A or D with zero also leads to logic-0. Thus,

$$Z = \left(A\overline{B}C + \underbrace{A0D}_{=0} + \overline{A}\overline{B}\right)C = \left(A\overline{B}C + \overline{A}\overline{B}\right)C$$

Expanding further,

$$Z = \left(A\overline{B}C + \overline{A}\overline{B}\right)C = A\overline{B}CC + \overline{A}\overline{B}C$$

By rule 7, the term in red $C.C = C$, giving

$$Z = A\overline{B}C + \overline{A}\overline{B}C = \overline{B}C\left(A + \overline{A}\right)$$

By rule 6, the term in green yields 1, so that

$$Z = \overline{B}C1 = \boxed{\overline{B}C}$$

Problem 3.4: Since the expression contains A and its complement, B and its complement, and C and its complement, a number of simplifications are immediately apparent, and none is particularly advantageous to use first. Grouping terms, we write

$$Z = \overline{A}BC + A\overline{B}\overline{C} + \overline{A}\overline{B}\overline{C} + A\overline{B}C + ABC = BC\left(\overline{A} + A\right) + A\overline{B}\overline{C} + \overline{A}\overline{B}\overline{C} + A\overline{B}C$$

By rule 6, the blue term in parentheses equals 1, giving

$$Z = BC.1 + A\overline{B}\,\overline{C} + \overline{A}\overline{B}\,\overline{C} + A\overline{B}C = BC + A\overline{B}\,\overline{C} + \overline{A}\overline{B}\,\overline{C} + A\overline{B}C$$

Next, factoring the terms in red,

$$Z = BC + A\overline{B}\left(C + \overline{C}\right) + \overline{A}\overline{B}\,\overline{C} = BC + A\overline{B}.1 + \overline{A}\overline{B}\,\overline{C}$$

$$\therefore Z = BC + \overline{B}\left(A + \overline{A}\,\overline{C}\right)$$

By rule 11, $A + \bar{A}\bar{C} = A + \bar{C}$, so the term in green can be simplified to give

$$\therefore Z = BC + \overline{B}\left(A + \overline{C}\right) = \boxed{BC + \overline{B}A + \overline{B}\,\overline{C}}$$

Problem 3.5: We begin by expanding the term in parentheses,

$$Z = AB + \overline{AC} + A\overline{B}C\left(AB + C\right) = AB + \overline{AC} + A\overline{B}CAB + A\overline{B}CC$$

Rearranging and using rules 7 and 8, we get

$$Z = AB + \overline{AC} + \underbrace{AA}_{=A}\,\underbrace{\overline{B}\overline{B}}_{=1}C + A\overline{B}\,\underbrace{CC}_{=C} = AB + \overline{AC} + A0C + A\overline{B}C$$

$$\therefore Z = AB + \overline{AC} + A\overline{B}C$$

$$\therefore Z = A\left(B + \overline{B}C\right) + \overline{AC}$$

By rule 11, $B + \bar{B}C = B + C$, so that

$$Z = A\left(B + C\right) + \overline{AC} = AB + AC + \overline{AC}$$

Lastly, because $AC + \overline{AC} = 1$ (rule 6) and $AB + 1 = 1$ (rule 4),

$$Z = AB + \underbrace{AC + \overline{AC}}_{=1} = \underbrace{AB + 1}_{=1}$$

$$\therefore \boxed{Z = 1}$$

Problem 3.6: At the outset, we apply the distributive law to the expressions in the first and second parentheses, giving

$$Z = \left(B + BC\right)\left(B + \overline{B}C\right)\left(B + D\right) = \left(BB + B\overline{B}C + BBC + B\overline{B}CC\right)\left(B + D\right)$$

Using rules 1, 7, and 8, we simplify to obtain

$$Z = \left(\underbrace{BB}_{=B} + \underbrace{B\overline{B}}_{=0}C + \underbrace{BB}_{=B}C + \underbrace{B\overline{B}}_{=0}CC\right)\left(B + D\right)$$

$$\therefore Z = \left(B + \underbrace{0C}_{=0} + BC + \underbrace{0CC}_{=0}\right)\left(B + D\right)$$

$$\therefore Z = \left(B + BC\right)\left(B + D\right)$$

Applying the distributive law a second time, we get

$$Z = \left(B + BC\right)\left(B + D\right) = \underbrace{BB}_{=B} + BD + \underbrace{BB}_{=B}C + BCD$$

$$\therefore Z = B + BD + BC + BCD$$

$$\therefore Z = B\left(D + 1\right) + BC\left(D + 1\right)$$

By rule 4, $D + 1 = 1$, so that

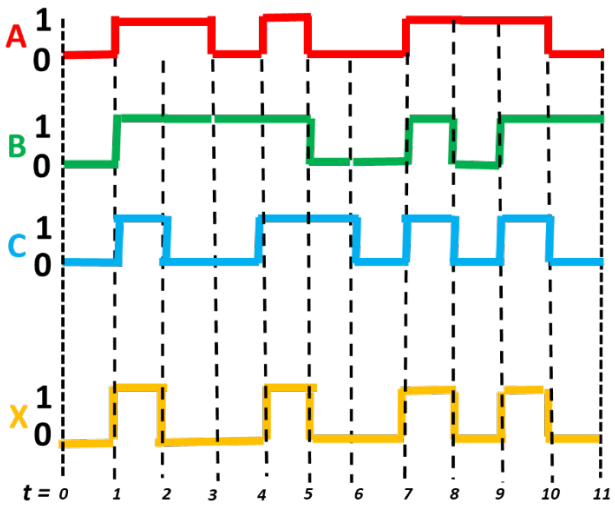
$$Z = B + BC = B\left(C + 1\right)$$

Applying rule 4 one last time, $C + 1 = 1$ and

$$Z = B\underbrace{\left(C + 1\right)}_{=1} = \boxed{B}$$

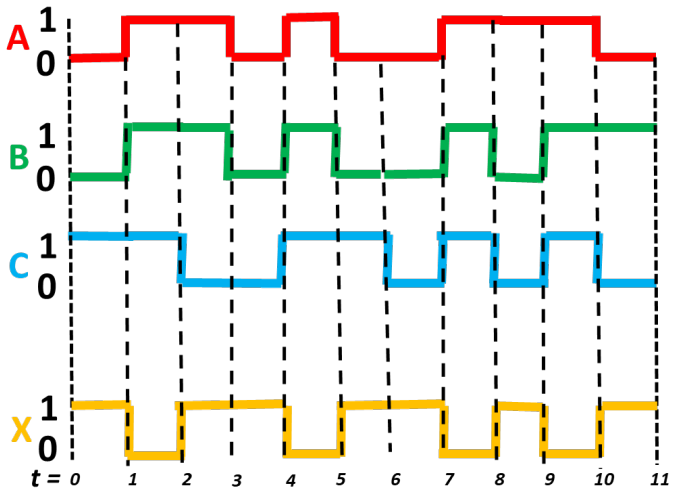
P.4 ➡ *Solution*

Problem 4.1: The functioning of an AND gate is straightforward: the output is 1 only when all inputs are also 1, otherwise the output is 0. To apply this rule to the waveforms in question, take the interval from $t = 0$ to $t = 1$. In this interval, input A is 1, input B is 0, and input C is 0; since at least one input is zero, the output $X = 0$. From $t = 1$ to $t = 2$, inputs A, B, and C are all 1, therefore $X = 1$. The output waveform is sketched in the next page.

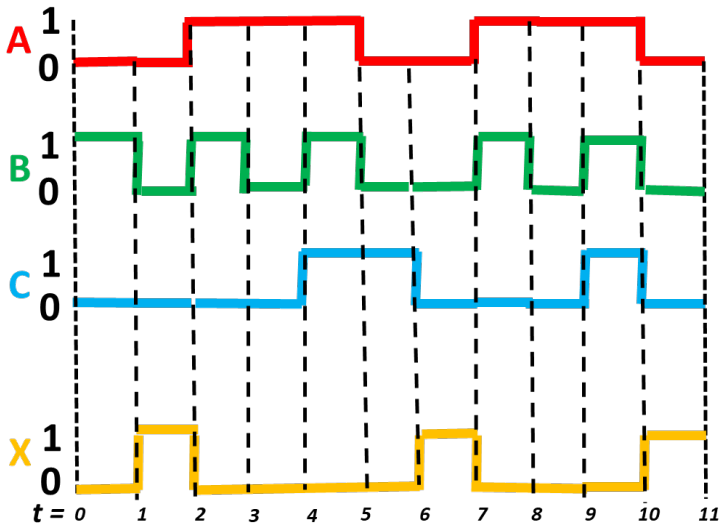


Problem 4.2: The functioning of an OR gate is simple: whenever one or more of the inputs is 1, the output is 1; if all inputs are 0, the output is 0. Applying this rule to the gate in question is straightforward. For example, from $t = 0$ to $t = 1$, input A is 1, input B is 0, and input C is 0; it follows that $X = 1$. From $t = 1$ to $t = 2$, input A is 0, input B is 0, and input C is 0; accordingly, $X = 0$. The output waveform is sketched below.

Problem 4.3: In a NAND gate, the output is 0 only when all inputs are 1; otherwise, the output is 1. Applying this rule to the waveforms in question, consider the interval from $t = 0$ to $t = 1$. In this interval, input A is 0, input B is 0, and input C is 1; since at least one input is zero, the output $X = 1$. From $t = 1$ to $t = 2$, inputs A, B, and C are all 1, therefore $X = 0$. The output waveform is sketched below.



Problem 4.4: In a NOR gate, the output is 0 whenever any of the inputs is 1; if all inputs are 0, the output is 1. Applying this rule to the waveforms in question, consider the interval from $t = 0$ to $t = 1$. In this interval, input A is 0, input B is 1, and input C is 0; since at least one input is 1, the output $X = 0$. From $t = 1$ to $t = 2$, inputs A, B, and C are all 0, therefore $X = 1$. The output waveform is sketched in the next page.

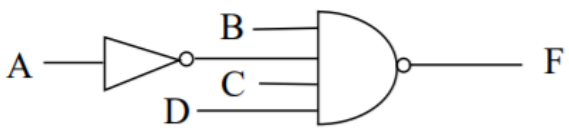


P.5 ➡ *Solution*

Problem 5.1, equation 1: The first equation can be simplified to

$$F_1 = \left(A + \overline{B}\right)\left(\overline{A}\,\overline{B} + C\right) + \overline{CD} = \left(\overline{\overline{A}BCD}\right)$$

and is implemented below.



Problem 5.2, equation 1: The truth table for all four equations is given at the end of this solution.

Problem 5.3, equation 1: With $A = 0$, function F_1 simplifies to

$$F_1 = \left(\overline{\overline{A}BCD}\right) = \left(\overline{\overline{0} \cdot B \cdot C \cdot D}\right) = \boxed{\overline{BCD}}$$

Problem 5.4, equation 1: Having $A = \overline{B}$ (or $B = \overline{A}$) leads to

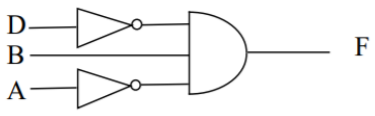
$$F_1 = \overline{\overline{A}BCD} = \overline{BBCD} = \boxed{\overline{BCD}}$$

Problem 5.1, equation 2: Using De Morgan's laws, the fact that $X \cdot \overline{X} = 0$, the fact that $X \cdot X = X$, and the distributive law, we obtain

$$F_2 = \left(A \cdot \overline{C} + \overline{BD}\right)\left(\overline{D + B + A}\right) = \left(A \cdot \overline{C} + \overline{B} + \overline{D}\right)\left(\overline{D} \cdot B \cdot \overline{A}\right)$$

$$\therefore F_2 = \overline{D}B\overline{A}$$

The simplified function is implemented below.



Problem 5.2, equation 2: The truth table for all four equations is given at the end of this solution.

Problem 5.3, equation 2: With $A = 0$, it follows that $\overline{A} = 1$ and

$$F_2 = \overline{D}B\overline{A} = \overline{D}B1 = \boxed{B\overline{D}}$$

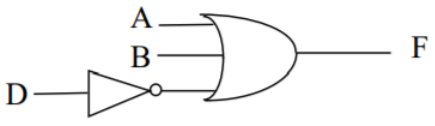
Problem 5.4, equation 2: With $A = \overline{B}$ or, equivalently, $B = \overline{A}$, F_2 becomes

$$F_2 = \overline{D}B\overline{A} = \boxed{0}$$

Problem 5.1, equation 3: Using the fact that $X + XY = X$ and De Morgan's laws, we get

$$F_3 = A + \overline{BD} + ACD = A + B + \overline{D}$$

The simplified function is implemented with the following logic diagram,



Problem 5.2, equation 3: The truth table for all four logic equations is given at the end of this solution.

Problem 5.3, equation 3: With $A = 0$, we straightforwardly obtain

$$F_3 = 0 + B + \overline{D} = \boxed{B + \overline{D}}$$

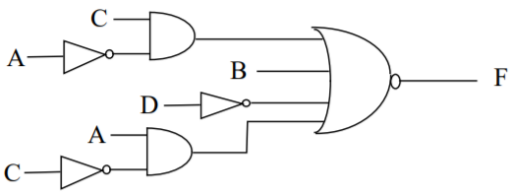
Problem 5.4, equation 3: With $A = \overline{B}$ and the rule $X + \overline{X} = 1$, we ultimately obtain

$$F_3 = A + B + \overline{D} = \boxed{1}$$

Problem 5.1, equation 4: Using the fact that $\overline{\overline{X}} + XY = \overline{\overline{X}} + Y$ and De Morgan's, the function simplifies to

$$F_4 = \overline{\overline{A}C + B + \overline{D} + A\overline{C}}$$

The function is implemented below.



Problem 5.2, equation 4: The truth table for all four equations is given at the end of this solution.

Problem 5.3, equation 4: With $A = 0$, we have

$$F_4 = \overline{\overline{A}C + B + \overline{D} + A\overline{C}} = \overline{1C + B + \overline{D} + 0\overline{C}}$$

$$\therefore F_4 = \overline{C + B + \overline{D}}$$

$$\therefore \boxed{F_4 = \overline{B}\overline{C}D}$$

Problem 5.4, equation 4: With $A = \overline{B}$, F_4 ultimately becomes

$$F_4 = \overline{\overline{A}C + B + \overline{D} + A\overline{C}} = \boxed{\overline{B}CD}$$

The truth table is drawn next. The simplified Boolean equations obtained in the calculations above are also listed.

$$F_1 = \overline{\overline{A}BCD}$$

$$F_2 = \overline{\overline{D}B\overline{A}}$$

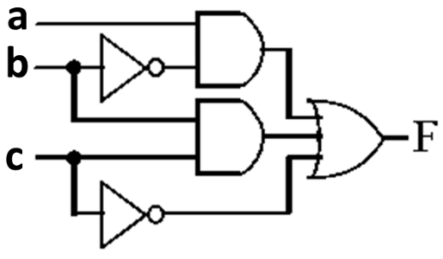
$$F_3 = A + B + \overline{D}$$

$$F_4 = \overline{\overline{A}C + B + \overline{D} + A\overline{C}}$$

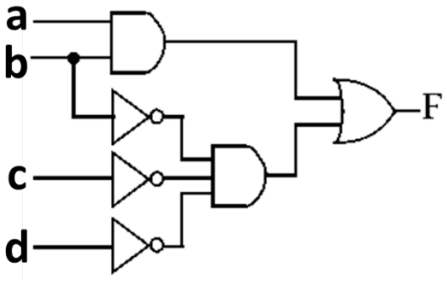
A	B	C	D	F_1	F_2	F_3	F_4
0	0	0	0	1	0	1	0
0	0	0	1	1	0	0	1
0	0	1	0	1	0	1	0
0	0	1	1	1	0	0	0
0	1	0	0	1	1	1	0
0	1	0	1	1	0	1	0
0	1	1	0	1	1	1	0
0	1	1	1	0	0	1	0
1	0	0	0	1	0	1	0
1	0	0	1	1	0	1	0
1	0	1	0	1	0	1	0
1	0	1	1	1	0	1	1
1	1	0	0	1	0	1	0
1	1	0	1	1	0	1	0
1	1	1	0	1	0	1	0
1	1	1	1	1	0	1	0

P.6 ➡ *Solution*

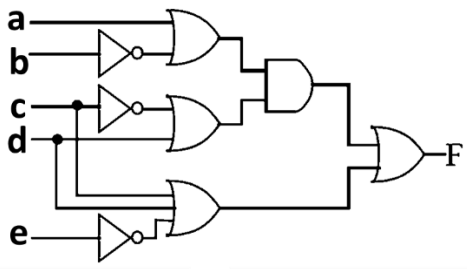
Problem 6.1: The gate-level circuit is illustrated below.



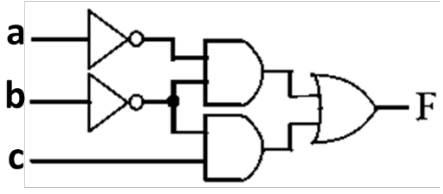
Problem 6.2: The gate-level circuit is illustrated below.



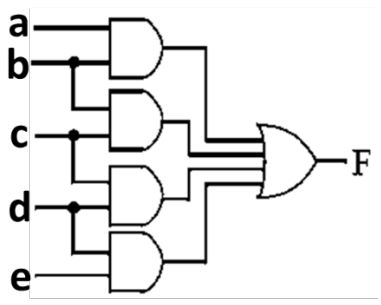
Problem 6.3: The gate-level circuit is illustrated below.



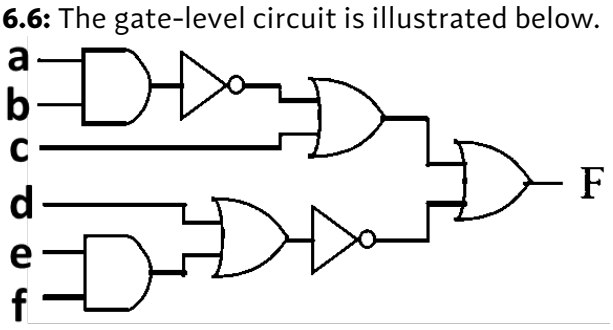
Problem 6.4: The gate-level circuit is illustrated below.



Problem 6.5: The gate-level circuit is illustrated below.

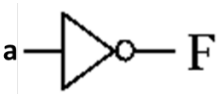


Problem 6.6: The gate-level circuit is illustrated below.

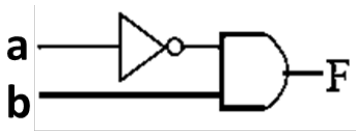


P.7 ➡ *Solution*

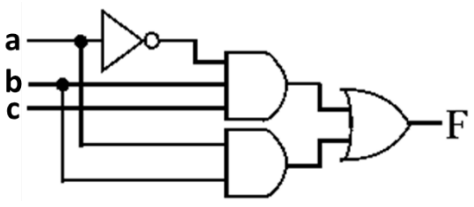
Problem 7.1: The circuit for function 7.1 is shown below.



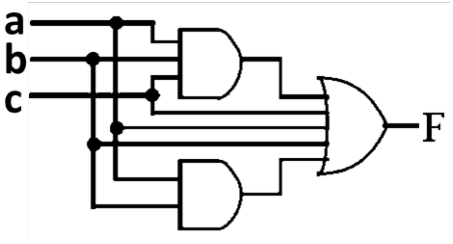
Problem 7.2: The circuit for function 7.2 is shown below.



Problem 7.3: The circuit for function 7.3 is shown below.



Problem 7.4: The circuit for function 7.4 is shown below.



P.8 ➡ *Solution*

Problem 8.1: The Boolean equation that represents the circuit is

$$F(a,b) = (ab' + b)'$$

Problem 8.2: The Boolean equation that represents the circuit is

$$G(a,b,c) = (ab' + b) + a'c$$

P.9 ➡ *Solution*

Problem 9.1: The first function is `F(a,b,c) = c'`.

Inputs			Output
a	b	c	F
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

Problem 9.2: The second function is `F(a,b,c) = a'b`.

Inputs			Output
a	b	c	F
0	0	0	0
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

Problem 9.3: The third function is `F(a,b,c) = a'bc + ab`.

Inputs			Output
a	b	c	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

Problem 9.4: The fourth function is `F(a,b,c) = abc + ab + a + b + c`.

Inputs			Output
a	b	c	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Problem 9.5: The truth table is shown below.

Inputs			Output
a	b	c	F
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

Problem 9.6: The truth table is shown below.

Inputs			Output
a	b	c	F
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Problem 9.7: The truth table is shown below.

Inputs			Output
a	b	c	F
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Problem 9.8: The truth table is shown below.

Inputs				Output
a	b	c	d	F
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	1
1	1	1	1	0

Problem 9.9: In Problem 8.1, we established that the circuit in question is represented by the Boolean `F(a,b) = (ab'+ b)'`. The corresponding truth table is shown below.

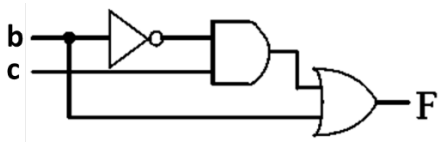
Inputs		Output
a	b	F
0	0	1
0	1	0
1	0	0
1	1	0

Problem 9.10: In Problem 8.2, we established that the circuit in question is represented by the Boolean `G(a,b,c) = (ab'+ b) + a'c`. The corresponding truth table is shown below.

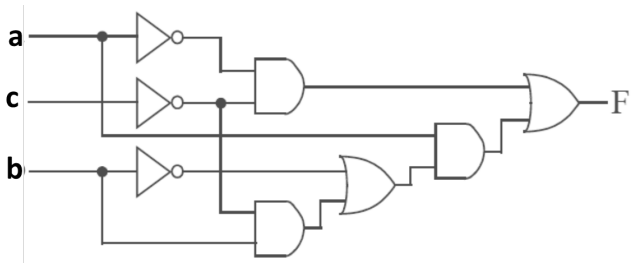
Inputs			Output
a	b	c	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

P.10 ➡ *Solution*

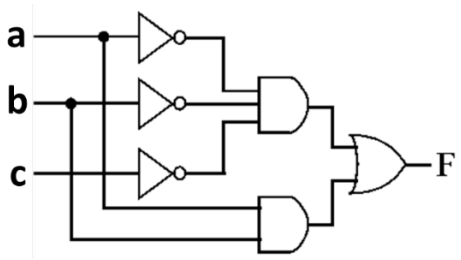
Problem 10.1: The circuit is illustrated below.



Problem 10.2: The circuit is illustrated below.



Problem 10.3: The circuit is illustrated below.



P.11 ➡ *Solution*

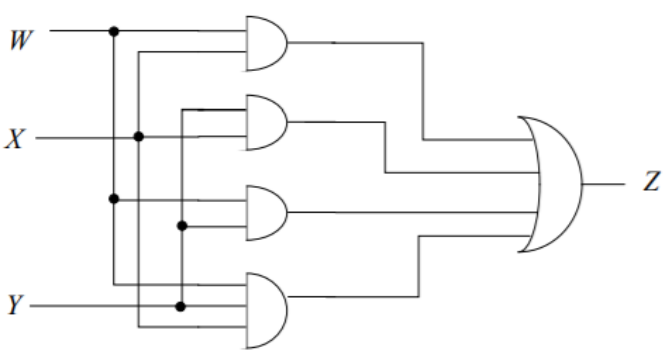
Problem 11.1: The Boolean equation we aim for is

$$Z = WX + WY + XY + WXY$$

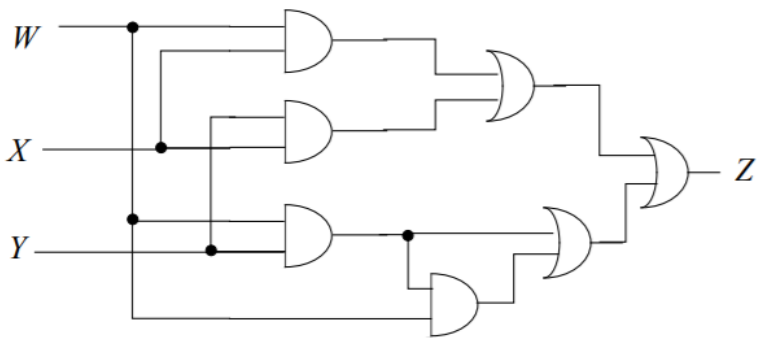
Problem 11.2: The logic table is shown in continuation.

W	X	Y	Z
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

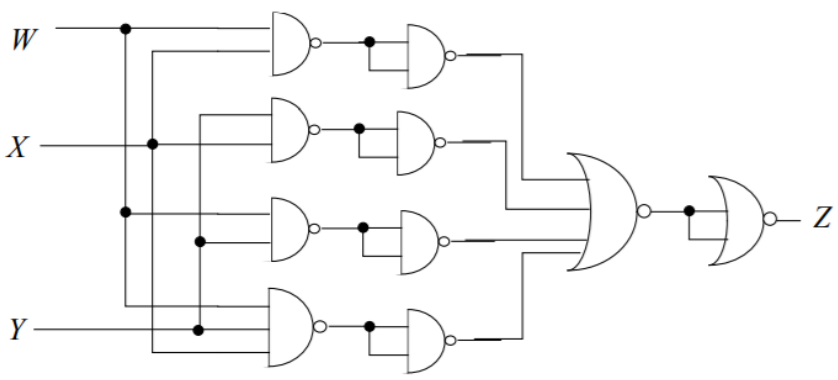
Problem 11.3: The logic diagram desired is shown below.



Problem 11.4: The logic diagram desired is shown below. The main difference between this diagram and the previous one is the use of more OR gates, but with less inputs in each one.



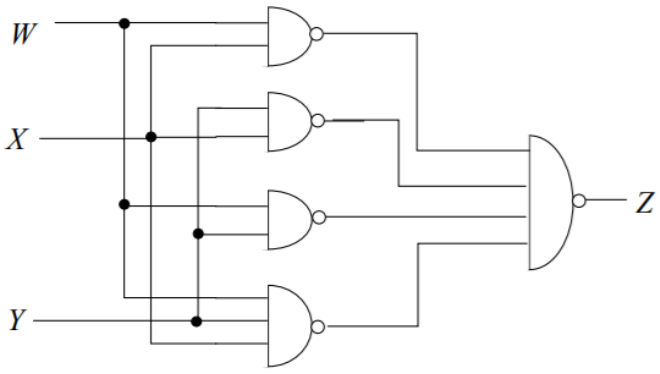
Problem 11.5: The logic diagram is shown below.



Problem 11.6: To implement function Z with NAND gates only, we first appeal to De Morgan's laws and restate the logic equation as

$$Z = \overline{\left(\overline{WX}\right)\left(\overline{WY}\right)\left(\overline{XY}\right)\left(\overline{WXY}\right)}$$

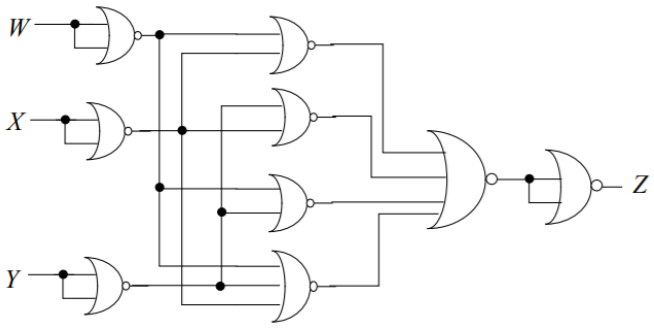
The corresponding logic diagram is shown below.



Problem 11.7: To implement Z with NOR gates only, we apply De Morgan's laws and write

$$Z = \overline{\left(\overline{W}+\overline{X}\right)+\left(\overline{W}+\overline{Y}\right)+\left(\overline{Y}+\overline{X}\right)+\left(\overline{W}+\overline{X}+\overline{Y}\right)}$$

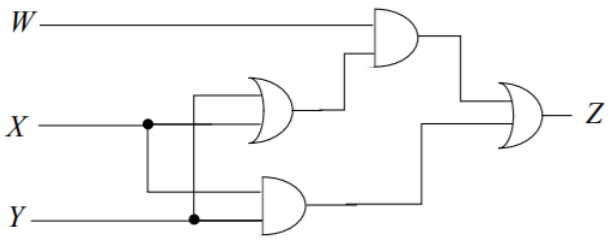
The corresponding logic diagram is shown below.



Problem 11.8: We first use some Boolean algebra,

$$Z = WX + WY + XY + WXY$$
$$\therefore Z = WX\underbrace{\left(1+Y\right)}_{=1} + WY + XY$$
$$\therefore Z = WX + WY + XY$$
$$\therefore Z = W\left(X+Y\right) + XY$$

This simplified relation can be represented by a simple 4-gate diagram, as shown.



P.12 ➡ *Solution*

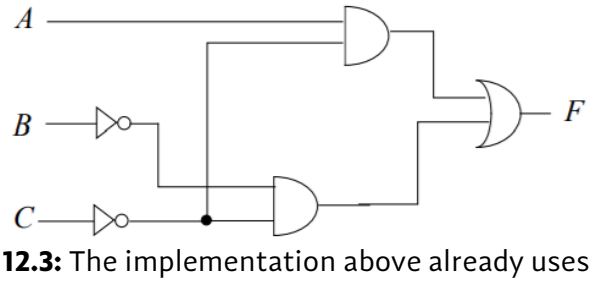
Problem 12.1: F can be represented with the Boolean equation

$$F = \overline{A}\overline{B}\overline{C} + A\overline{B}\overline{C} + A\overline{B}C + ABC$$

which can be simplified to

$$F = \overline{B}\overline{C}\left(A+\overline{A}\right) + AC\left(B+\overline{B}\right) = \overline{B}\overline{C} + AC$$

Problem 12.2: The simplified function can be implemented with three gates and two inverters, as shown.

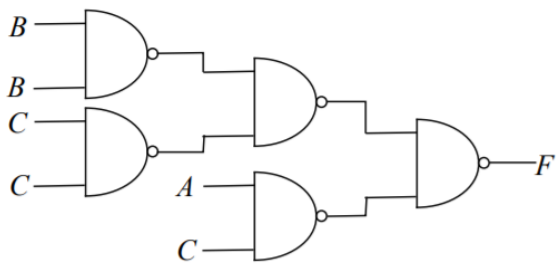


Problem 12.3: The implementation above already uses gates with two or less inputs.

Problem 12.4: Using our simplified version of F, De Morgan's laws, and the fact that a NAND gate with logical signal X tied into both inputs produces $\overline{X}$, we write

$$F = \overline{\left(\overline{B}\cdot\overline{C}\right)\cdot\left(\overline{A\cdot C}\right)}$$

This function is implemented with the following logic diagram.



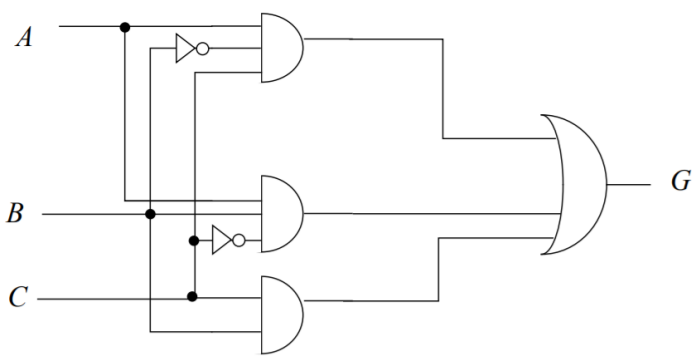
Problem 12.5: Function G can be represented with the Boolean relation

$$G = \overline{A}BC + A\overline{B}C + AB\overline{C} + ABC$$

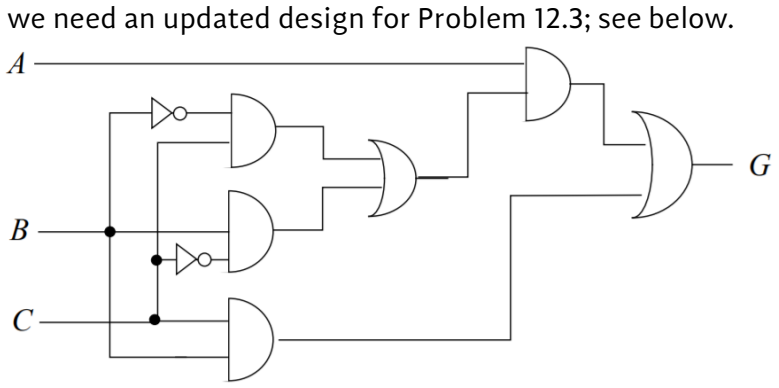
Grouping the first and last terms simplifies things a little,

$$G = BC\left(A + \overline{A}\right) + A\overline{B}C + AB\overline{C} = A\overline{B}C + AB\overline{C} + BC$$

The simplified function is implemented below.



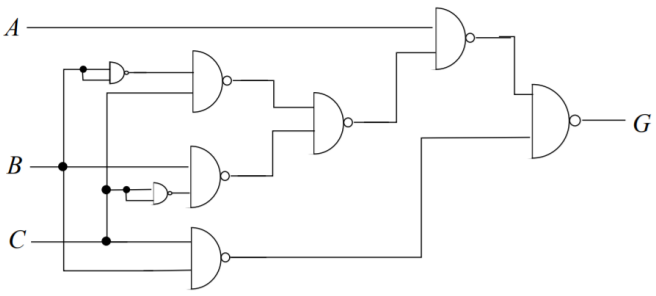
The implementation of G given above uses gates with more than 2 inputs, so we need an updated design for Problem 12.3; see below.



To implement G with NAND gates only, we first restate it as

$$G = \overline{BC\left(\overline{A\left(\overline{\overline{B}C}\right)\left(\overline{BC}\right)}\right)}$$

The corresponding circuit is shown below.



P.13 ➡ *Solution*

Problem 13.1: If the lower NAND gate that ties inputs from B and C has its output stuck at 1, we can altogether ignore its contribution to the circuit's logic behavior. The remaining AND gate in the first stage of the circuit accepts an input from A and an inverted input from B, so that

$$\boxed{Z = A\overline{B}}$$

Problem 13.2: In this diagram, we can neglect not only the output from the lower NAND gate in the first stage, but also the inverted input from B that enters the upper NAND gate. As a result, output Z is further simplified to

$$Z = A\cancel{\overline{B}} = \boxed{A}$$

Problem 13.3: This diagram is similar to the one considered in the two previous problems, but NOR gates are employed instead of NAND gates, and

the faults are of stuck-at-0 type instead of stuck-at-1 type. Much like the diagram in the previous problem, we can neglect the contribution of the lower gate's (in this case, a NOR gate) output and the inverted *B* input just before the upper gate in the first stage. As a result, output *Z* can be described by the simple Boolean relation

$$\boxed{Z = A}$$

Problem 13.4: The Boolean for a stuck-at-1 is $Z = A\bar{B}$. The Boolean for a stuck-at-0 is $Z = 1$. One possible test is $A = 1, B = 1$. If the output $Z = 1$, then it is a stuck-at-0 fault. If the output $Z = 0$, then it is a stuck-at-1 fault.

REFERENCES

- AGARWAL, A. and LANG, J.H. (2005). *Foundations of Analog and Digital Electronic Circuits*. Amsterdam: Elsevier.
- SEDHA, R.S. (2004). *A Textbook of Digital Electronics*. New Delhi: S. Chand and Company.
- VAHID, F. (2011). *Digital Design*. 2nd edition. Hoboken: John Wiley and Sons.