Solutions PDF Generated from: solutions-openai-generated/homeworks/assignment-01-solutions-set-02.json

Question A

Simplify f(x,y,z) = (x(y+z)+z)y into sum-of-products form. Follow example from lecture 3 slides, however, at each step of the simplification process write down the Boolean algebra theorem used in this step. Note lecture slides sometimes used multiple theorems in each step in the slides, your solutions should use only one theorem at each step and may have more steps than shown in the slides. List of Boolean algebra theorems can be found in the supplementary materials on Canvas site.

```
Step 1: (x'(y+z') + z')y

Theorem used: Associative

Step 2: (x'y + x'z' +z'y)

Theorem used: Distributive

Step 3: x'y + (x'z' +z'y)

Theorem used: Commutative

Step 4: x'y + x'z' +z'y

Theorem used: None (already in sum of products form)
```

Question B

Simplify the following Boolean equation into sum-of-products form. Again, justify each simplification step with Boolean algebra theorem used: f(a,b,c,d,e) = (a+b+c)(a+b+d)(a+b+e)(a+d+e)(a+c).

```
Sum-of-products form:
f(a,b,c,d,e) =    acde    +    abcde    +    abcde    +    abde    +    abde

Step 1:
f(a,b,c,d,e) = (a+b+c')(a+b+d)(a+b+e)(a+d'+e)(a'+c)
             = (a+b+c')(a+d'+e)+a(a+b+d)(a+b+e)(a'+c)
             = (a+d'+e)(a+b+c')+a(a+b+d)(a+b+e)(a'+c)        ------------( Commutative property)

Step 2:
             = (a+d'+e)(a+b+c')+a(a+b+d)(a'+e)(a'+c)         ------------( Distributive property)

Step 3:
             = (a+d'+e)(a+b+c')+a(a'+e)(a'+c)                ------------(Complementary property)

Step 4:
```

```
                  = (a+d'+e)(a+b+c')+a'c'd'e'                    ------------(Distributive property)
```

Step 5:
```
                  = a'cd'e' + abc'd'e' + abcd'e' + ab'd'e + ab'de ------------(Distributive property)
```

Question C

Simplify the following Boolean equation into sum-of-products form. Again, justify each simplification step with Boolean algebra theorem used: f(a,b,c,d,e) = (a+b+c)(a+b+d)(a+b+e)(a+d+e)(a+c).

```
f(a,b,c,d,e) = (a+b) (a+d'+e) (a'+c) (Distributive Law)
= (a+b+c')(a+b+d)[a+b(d+e)](a'+c) (Associative Law)
= (a+b+c')(a+b+d)(a+bd+be)(a'+c) (Distributive Law)
= (a+b+c')(a+b+d)(a+b+d'+e)(a'+c) (Complements Law)
= (a+b+c')(a+b+d)(a+b+d'+e)(a+c) (Absorption Law)
= (a+b+c')(a+b+d)(a+d'+e)(a+c) (Commutative Law)
= (a+b+c')(a+d'+e)(a+b+d)(a+c) (Commutative Law)
```

Question D

Simplify f(x,y,z) = (x(y+z)+z)y into most simple form, which may not be sum-of-products. Again, justify each simplification step with Boolean algebra theorem used.

```
= x'y + x'z'y + z'y
= x'y(1+z') + z'y (a+a'b = a + b)
= x'y + z'y (x' y + x' y' = x'y)
= y(x' + z') (a + ab = a)
= y
```

Question E

Simplify f(x,y,z) = (x(y+z)+z)y into most simple form, which may not be sum-of-products. Again, justify each simplification step with Boolean algebra theorem used.

```
f(x,y,z) = ( x (y+ z )+ z )y
```

Using the Associative Law:

```
f(x,y,z) =  xy  +    xz    +   zy
```

Using the Distributive Law:

```
f(x,y,z) =  xy  + y( x  +   z )
```

Using the Identity Law (x + x' = 1):

```
f(x,y,z) = y(1 + x' + z')
```

Using the Identity Law (1.x = x):

```
f(x,y,z) = y(x + x' + z')
```

Finally, using the Commutative Law:

```
f(x,y,z) = y(x' + x + z')
```

2

The most simple form of the expression is: f(x,y,z) = y(x' + x + z')

Question F

Expand f(x,y,z) = xy + yz into canonical sum-of-products. What are the minterms that compose the canonical sum-of-products? Which rows of the truth table do the minterms correspond to? Write the truth table.

The canonical sum-of-products is:
xy'z'+xyz

Minterms: xy'z', xyz

The minterms correspond to the rows in the truth table where the output column is 1.

Truth Table:

```
x  y  z  f(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
```

Question G, Part A

Draw the symbol, Boolean equation, and truth table for a four-input OR gate.

Symbol:

Truth Table:

```
A | B | C | D | Y
--|---|---|---|--
0 | 0 | 0 | 0 | 0
0 | 0 | 0 | 1 | 1
0 | 0 | 1 | 0 | 1
0 | 0 | 1 | 1 | 1
0 | 1 | 0 | 0 | 1
0 | 1 | 0 | 1 | 1
0 | 1 | 1 | 0 | 1
0 | 1 | 1 | 1 | 1
1 | 0 | 0 | 0 | 1
1 | 0 | 0 | 1 | 1
1 | 0 | 1 | 0 | 1
1 | 0 | 1 | 1 | 1
1 | 1 | 0 | 0 | 1
1 | 1 | 0 | 1 | 1
1 | 1 | 1 | 0 | 1
1 | 1 | 1 | 1 | 1
```

Boolean Equation:
Y = A + B + C + D

Question G, Part B

Draw the symbol, Boolean equation, and truth table for a three-input XNOR gate.

Symbol:

![XNOR gate symbol](https://constantberge.files.wordpress.com/2018/05/xnor.png)

Boolean equation:

A XNOR B XNOR C = (A B   C) + (A   B   C)

Truth Table:

A	B	C	Output
0	0	0	1
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

Question G, Part C

Draw the symbol, Boolean equation, and truth table for a five-input NAND gate.

Symbol:

![5-input NAND Gate Symbol](5-input-NAND-Gate-Symbol.jpg)

Boolean Equation:

![5-input NAND Gate Boolean Equation](5-input-NAND-Gate-Boolean-Equation.jpg)

Truth Table:

A	B	C	D	E	Y
0	0	0	0	0	1
0	0	0	0	1	1
0	0	0	1	0	1
0	0	0	1	1	1
0	0	1	0	0	1
0	0	1	0	1	1
0	0	1	1	0	1
0	0	1	1	1	1
0	1	0	0	0	1
0	1	0	0	1	1
0	1	0	1	0	1
0	1	0	1	1	1
0	1	1	0	0	1
0	1	1	0	1	1
0	1	1	1	0	1
0	1	1	1	1	1
1	0	0	0	0	1

```
| 1    | 0    | 0    | 0    | 1    | 1    |
| 1    | 0    | 0    | 1    | 0    | 1    |
| 1    | 0    | 0    | 1    | 1    | 1    |
| 1    | 0    | 1    | 0    | 0    | 1    |
| 1    | 0    | 1    | 0    | 1    | 1    |
| 1    | 0    | 1    | 1    | 0    | 1    |
| 1    | 0    | 1    | 1    | 1    | 1    |
| 1    | 1    | 0    | 0    | 0    | 1    |
| 1    | 1    | 0    | 0    | 1    | 1    |
| 1    | 1    | 0    | 1    | 0    | 1    |
| 1    | 1    | 0    | 1    | 1    | 1    |
| 1    | 1    | 1    | 0    | 0    | 0    |
| 1    | 1    | 1    | 0    | 1    | 0    |
| 1    | 1    | 1    | 1    | 0    | 0    |
| 1    | 1    | 1    | 1    | 1    | 0    |
```

Question I

There are 16 different truth tables for Boolean functions of two variables. List each truth table. Give each one a short descriptive name (such as OR, NAND, and so on).

```
1.  OR - all truth values in either column yield true; falses only in both columns
2.  NOR - all truth values in either column yield false; trues only in both columns
3.  XOR - only one true value in either column yields true; all other yield false
4.  XNOR - only one false value in either column yields false; all other yield true
5.  AND - all truth values in both columns yield true; falses only in either column
6.  NAND - all truth values in both columns yield false; trues only in either column
7.  A only - all truth values in A yield true, all in B yield false
8.  B only - all truth values in B yield true, all in A yield false
9.  A then B - all truth values in A yield true, all in B yield true only if A is true
10. B then A - all truth values in B yield true, all in A yield true only if B is true
11. A and not B - all truth values in A yield true, all in B yield false
12. B and not A - all truth values in B yield true, all in A yield false
13. Not A and Not B - all truth values in both columns yield false
14. Not A then Not B - all truth values in A yield false, all in B yield false only if A is false
15. Not B then Not A - all truth values in B yield false, all in B yield false only if B is false
16. Const ants - all truth values in both columns yield the same truth value (either true or false)
```

Question J

Write a truth table for the function performed by the gate shown in the figure. The truth table should have two inputs, A and B. What is the name of this function?

```
Truth Table:
A | B | Output
0 | 0 | 0
0 | 1 | 1
1 | 0 | 1
1 | 1 | 0

This function is an XOR Gate.
```

Question K

Write a truth table for the function performed by the gate in the figure. The truth table should have three inputs, A, B, and C.

```
A | B | C | Output
-- | -- | -- | ------
0 | 0 | 0 | 0
0 | 0 | 1 | 0
0 | 1 | 0 | 0
0 | 1 | 1 | 0
1 | 0 | 0 | 1
1 | 0 | 1 | 1
1 | 1 | 0 | 1
1 | 1 | 1 | 0
```

Question L

Sketch a transistor-level circuit for a CMOS four-input NOR gate.

![Transistor-level CMOS four-input NOR gate](https://i.imgur.com/jmXijm8.png)

Execution Time

```
0:02:50.257665
```

OpenAI Parameters

```
Model: text-davinci-003, Max. Tokens: 1024, Temperature: 1, N: 1
```