**Name:**

Note: Please post your homework to ICS232 D2L on or before the due date.

**Chapter 3 – Boolean Algebra and Digital Logic**

**Essential Terms and Concepts**

2. Which Boolean operation is referred to as a Boolean product?

AND (textbook page 272)

3. Which Boolean operation is referred to as a Boolean sum?

OR (textbook page 273)

12. Describe the operation of a ripple-carry adder. Why are ripple-carry adders not used in most computers today?

Replicating a full 16 times, feeding the carry out of one circuit into the carry in of the circuit immediately to its left (textbook page 324).

18. How are sequential circuits different from combinational circuits?

Combinational circuits have no concept of storage where sequential circuits do. The output depends on past input in a sequential circuit (textbook page 337).

20. What do we mean when we say that a sequential circuit is edge triggered rather than level triggered?

They are allowed to change their states on either the rising or falling edge of the clock signal (textbook page 338).

24. Which flip-flop give a true representation of computer memory?

D (data) flip-flop (textbook page 343)

**Exercises**

2. Construct a truth table for the following:

a) xyz + x(yz)' + x'(y+z) + (xyz)'

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| x | y | z | xyz | yz | (yz)’ | x(yz)’ | y+z | x’(y+z) | (xyz)’ | xyz + x(yz)' + x'(y+z) + (xyz)' |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |

b) (x + y')(x' + z')(y' + z')

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| x | y | z | y’ | z’ | x + y’ | x’ + z’ | y’ + z’ | (x+y’)(x’+z’)(y’+z’) |
| 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |

4. Using DeMorgan's Law, write an expression for the complement of F if

F(x,y,z) = (x'+y)(x+z)(y'+z)'

F’(x, y, z) = (x’ + y)’ + (x+z)’ + ((y’ + z)’)’

10. Show that x = xy + xy'

a) Using truth tables

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| x | y | xy | xy’ | xy + xy’ | x |
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 | 1 |
| 1 | 1 | 1 | 0 | 1 | 1 |

b) Using Boolean identities

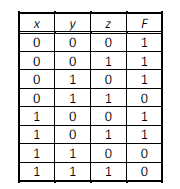
1. Distributive Law: xy + xy’ = x(x + y’)

2. Inverse Law: xy + xy’ = x

17. Simplify the following functional expressions using Boolean algebra and its identities. List the identity used at each step.

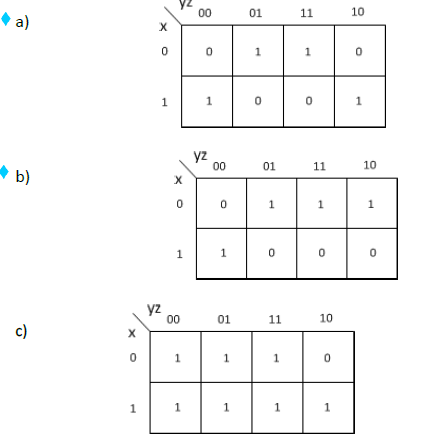
1. x(y + z)(x' + z')-> uses the distributive law twice: 1. x(yx’ + yz’ + zx’) -> distribute and simplify - > xyz’
2. xy + xyz + xy'z + x'y'z -> identity xy + xy’z + x’y’z -> distribute xy’z + x’y’z = zy’ -> combine xy+zy’
3. xy'z + x(y + z')' + xy'z' -> de morgan xy’z + x(y’z) + xy’z’ -> distribute -> xy’ + xy’z -> xy’

23. The truth table for a Boolean expression is shown below. Write the Boolean expression in sum-of-products form.

****

x’y’z’ + x’y’z +x’yz’ + xy’z’ + xy’z

29. Write a simplified expression for the Boolean function defined by each of the following Kmaps.



1. x’y’z + x’yz+ x y’z’ + xyz’
2. xy’z’ + x’y’z + x’yz

3) x’y’z’ + x’y’z + x’yz + xy’z’ + xy’z + xyz + xyz’

46. Draw the combinational circuit that directly implements the following Boolean expression:

F(x,y,z) = x + xy + y'z

A drawing of a person with a mathematical equation

Description automatically generated with medium confidence

51. How many inputs does a decoder have if it has 64 outputs?

6

52. How many control lines does a multiplexer have if it has 32 inputs?

5

**Write a simple C program to sum the entries in an array and print out the sum and the average. The following may be used as a template:**

#include <stdio.h>

#include <string.h>

int main(int argc, char \*\* argv)

{

int average;

int i;

int size;

int sum;

static int numbers[] = {1, -1, 100, 32, 64, -105, 33};

for (i = 0; i < size; i++) {

}

printf("Sum = %d, Average = %d\n", sum, average);

return (0);

}

Below is the code, I also attached the files to the D2L submission

#include <stdio.h>

#include <string.h>

**int** main(**int** argc, **char** \*\* argv) {

**int** average;

**int** i;

**int** size;

**int** sum;

sum = 0;

size = 7;

**static** **int** numbers[] = {1, -1, 100, 32, 64, -105, 33};

**for** (i = 0; i < size; i++) {

sum = sum + numbers[i];

}

average = sum/size;

printf("Sum = %d, Average = %d\n", sum, average);

**return**(0);

}

**Prepare for next class by reading Chapter 4 – MARIE: An Introduction to a Simple Computer**