Justin Ngo
Digital Logic Design
Barry Johnson
30 September 2020

Laboratory Assignment 1

Problem Statement

The task of the laboratory assignment is to design a 4-bit even parity code generator. For any 4-bits of information, the circuit must produce a single output P that is the correct parity bit such that when the parity bit is appended to the 4 bits of information, a 5-bit even parity code word is created. For example, if the circuit information is (w x y z) = (1 0 1 1), the circuit should produce an output P of 1 so that the resulting code word is (w x y z P) = (1 0 1 1 1) which has an even parity.

Design Solution

4-Bit Even Parity Generator Truth Table

w	**x**	**y**	**z**	**p**
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

Boolean Equation and Karnaugh Map for p0

wx \ yz	y'z'	y'z	yz	yz'
w'x'	0	1	0	1
w'x	1	0	1	0
wx	0	1	0	1
wx'	1	0	1	0

Boolean Equations

p = (w' ∧ x' ∧ y' ∧ z') ∨ (w' ∧ x' ∧ y' ∧ z) ∨ (w' ∧ x ∧ y' ∧ z') ∨ (w' ∧ x ∧ y ∧ z) ∨ (w ∧ x ∧ y' ∧ z)
∨ (w ∧ x ∧ y ∧ z') ∨ (w ∧ x' ∧ y' ∧ z') ∨ (w ∧ x' ∧ y ∧ z)

p = w' ∧ x'(y ∧ z' + y' ∧ z) + w' ∧ x (y ∧ z + y' ∧ z') + w ∧ x (y' ∧ z + y ∧ z')
+ w ∧ x' (y' ∧ z' + y ∧ z)

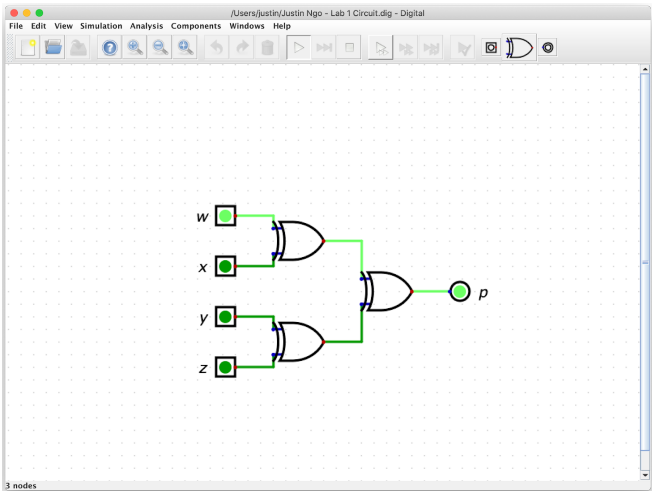
p = w' ∧ x'(y ⊕ z) + w' ∧ x (y ⊙ z) + w ∧ x (y ⊕ z) + w ∧ x' (y ⊙ z)

p = (w ⊕ x) (y'∧z' + y∧z) + (w ⊙ x) (z'y + zy') + (w ⊕ x) (y ⊙ z) + (w ⊙ x) (y ⊕ z)

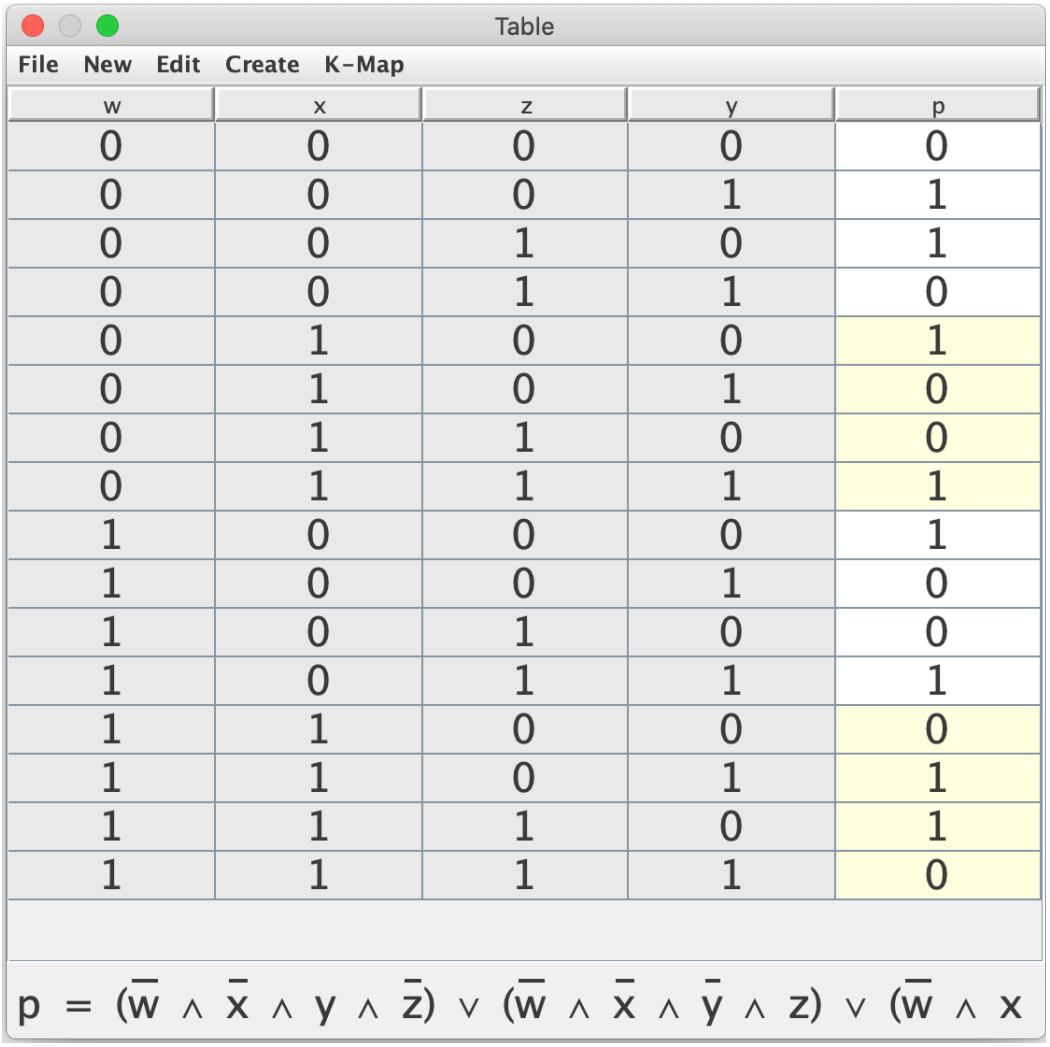
p = (w ⊕ x) (y ⊕ z)' + (w ⊕ x)' (y ⊕ z)

p = w ⊕ x ⊕ y ⊕ z

Logic Circuit Diagram



Simulator Truth Table



w	x	z	y	p
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

Summary and Conclusions

I began the laboratory assignment by researching and studying parity bits from the online textbook. Following the instructions from the assignment, I created a truth table with all the 16 possible input combinations and output values. The output value was 1 if there were an odd number of input variables with the value 1, and 0 if there were an even number of input variables with the value 0. I then created a Karnaugh map which produced a checkerboard value and came up with the long boolean equation for the values. I then simplified the boolean original equation using the distributive property and rules to transform over to XNOR and XOR. I built the logic function using the minimal number of gates and combinations of gates, using only three XOR. The simulator works because for example when (w x y z) = (1 1 1 0), the circuit should produce an output P of 1 so that the resulting code word is (w x y z P) = (1 0 1 1 1) which has an even parity. In contrast, the simulator also works because for example when (w x y z) = (0 0 1 1 0), the circuit produces an output P of 0 so that the resulting code word is (w x y z P) = (0 0 1 1 0) which has an even parity. I then finished by checking the output of my program with the built in truth table for my XOR functions.