SYSC2310 A and B Introduction to Digital Systems
Fall 2022

Lab 3 Report

Using and Simplifying Boolean Functions

Student Name: _Uchenna Obikwelu_

Student ID: _101241887_

TA Name: _____Mona_____

Lab Section: _SYSC 2310 L20__

Lab Date: 14th October 2022

Date Completed: 14th October 2022

Exercise 1: Simplify a Boolean Logic Function

Derive the Simplified Function (F_2)

$F_1 = xy'z + x'y'z + w'xy + wx'y + wxy$ Original

Simplified: $F_2 = y'z + yx + yw$

Show your work here:

$F_1 = xy'z + x'y'z + w'xy + wx'y + wxy$

F1=x(y'z+w'y+wy)+x'y'z+wx'y

F1= x(y'z+y(w+w'))+x'y'z+wx'y

F1= x(y'z+y(1))+x'y'z+wx'y

F1= xy'z+xy+x'y'z+wx'y

F1= y'(xz+x'z) +xy+wx'y

F1= y'(z(x+x'))+y(x+wx')

F1= y'(z(1))+y(x+wx')

F1= y'z+y(x+wx')

F1= y'z +y[(x+w).(x+x')]

F1= y'z +y[(x+w).(1)]

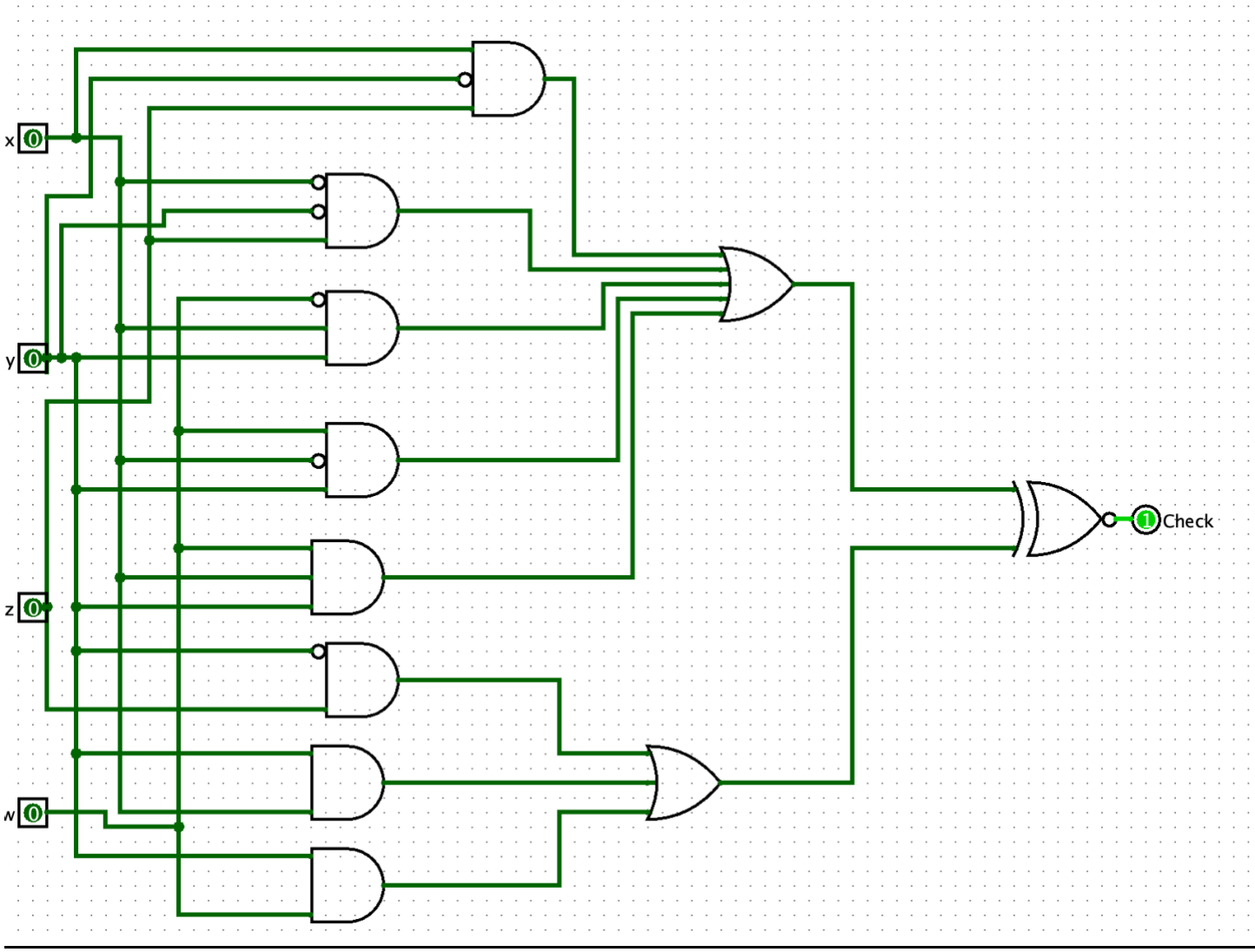
F1= y'z +y[(x+w)]

=F2

How many gates did you save?

14 gates were used for F1 while 4 gates were used for F2. I saved 10 gates.

Screen Shot of Exercise 1 Circuit

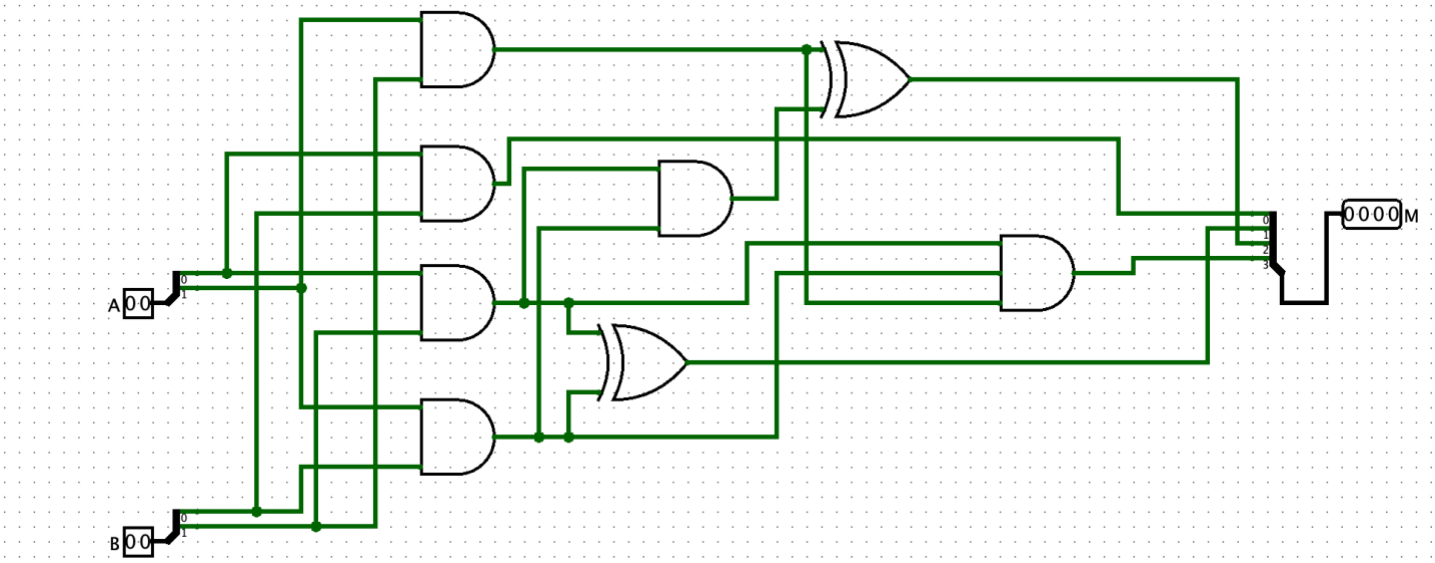

x
y
z
w
Check

Copy and Paste Exercise 1 Simulation Logging File

x	y	z	w	Check
0	0	0	0	1
1	0	0	0	1
1	1	0	0	1
1	1	1	0	1
1	1	1	1	1
1	1	1	0	1
1	1	0	0	1
1	0	0	0	1
0	0	0	0	1
0	1	0	0	1
0	1	1	0	1
0	1	1	1	1
0	0	1	1	1
0	0	0	1	1
0	0	0	0	1

Exercise 2: Convert a set of Boolean functions to a circuit

Screen Shot of Exercise 2 Circuit



Copy and Paste Exercise 2 Simulation Logging File

A	B	M
0	0	0
1	0	0
0	0	0
2	0	0
3	0	0
3	1	3
3	3	9
3	2	6
3	3	9
3	1	3
1	1	1
1	3	3
0	3	0
2	3	6
2	2	4
3	2	6
2	2	4
0	2	0
0	0	0

Exercise 3: Design a 3-bit by 2-bit binary multiplier

Derive the function for the full adder

Equations for a half adder are:

$S = x \oplus y$

$C = x.y$

Equations for a full adder are:

S= (x⊕y) ⊕(carryin)

C= (carryin . (x⊕y)) +(x.y)

Derive a closed form formulation to the multiplication of a 3-bit input $[A_2, A_1, A_0]$ by a 2-bit input $[B_1, B_0]$. The output would be $[M_4, M_3, M_2, M_1, M_0]$:

A_2 A_1 A_0

B_1 B_0

A_2B_0 A_1B_0 A_0B_0

A_2B_1 A_1B_1 A_0B_1

$[C_3\ S_3]$ $[C_2\ S_2]$ $[C_1\ S_1]$ M_0

M_4 M_3 M_2 M_1

$M_0 = A_0B_0$

$[C_1\ S_1] = [\ A_1B_0 . A_0B_1, A_1B_0 \oplus A_0B_1]$

$M_1 = A_1B_0 \oplus A_0B_1$

$[C_2\ S_2] = (\ (C1 . (A_2B_0 \oplus A_1B_1)\ ) + (A_2B_0 . A_1B_1), (A_2B_0 \oplus A_1B_1) \oplus (C1))$

$M_2 = (A_2B_0 \oplus A_1B_1) \oplus (C1)$

$[C_3\ S_3] = [A_2B_1 . C2, A_2B_1 \oplus C2]$

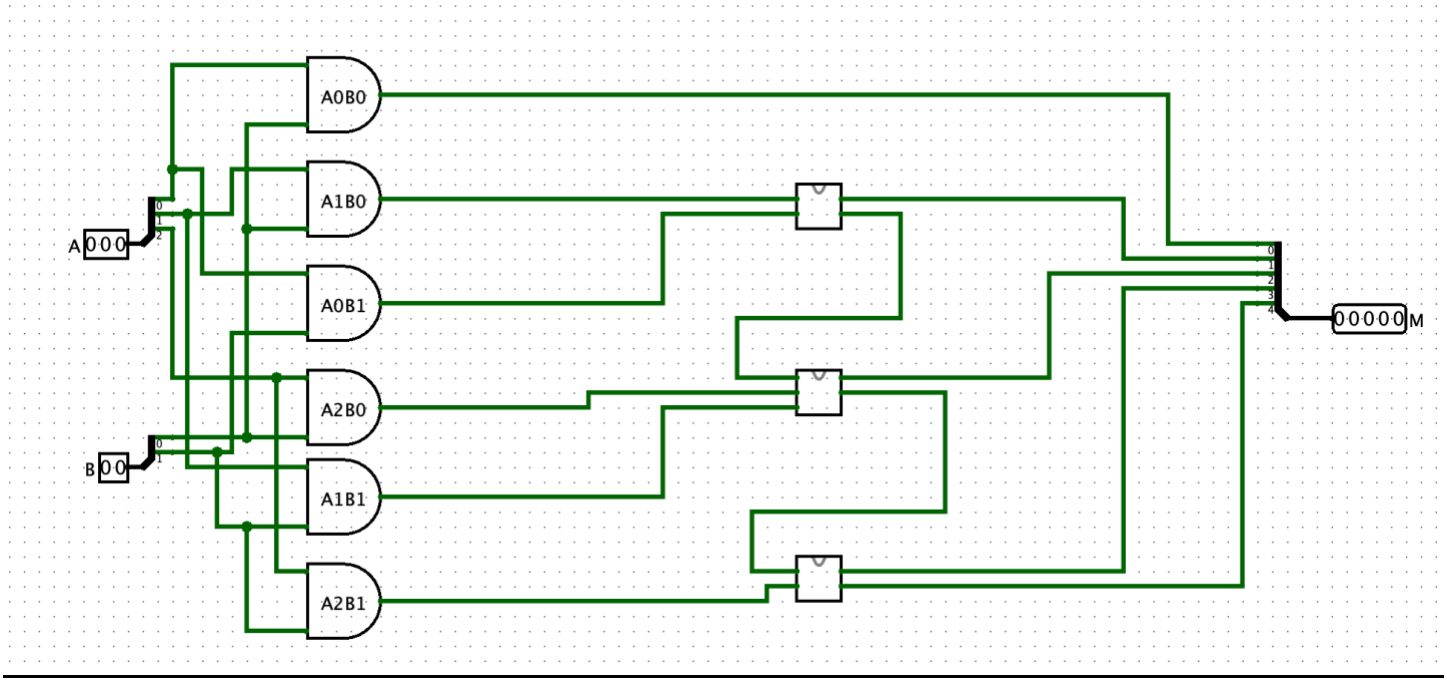
$M_3 = A_2B_1 \oplus C2$

$M_4 = A_2B_1 . C2$

Screen Shot of Exercise 3 Circuit

A	B	M
0	0	0
2	0	0
2	1	2
2	3	6
3	3	9
7	3	21
5	3	15
5	2	10
5	0	0
4	0	0
6	0	0
7	0	0
7	1	7
7	3	21
3	3	9
2	3	6
2	1	2
2	0	0
2	2	4
0	2	0
1	2	2
3	2	6
7	2	14
7	3	21
7	1	7
6	1	6
2	1	2
0	1	0
0	0	0

Copy and Paste Exercise 3 Simulation Logging File



INTRODUCTION TO DIGITAL SYSTEMS

SYSC2310 (Fall 2022)

Lab 3: Using and Simplifying Boolean Functions

Objectives

- Get familiar with simplifying Boolean functions.
- Convert Boolean functions to actual circuits.
- Build a 2-bit x 2-bit binary multiplier.

Relation to course outcomes

The work in this lab is related to the two following course outcomes:

- Manipulate or design basic combinational operators.
- Apply analysis skills to correctly describe the behavior of given combinational logic circuits.

Home Preparation

- Take the time to read this document prior to arriving to the lab in order to understand the essence of the work to be performed.

Preparation for the lab

1. Under your 'sysc2310lab/lab3' folder, create three folders 'Ex1', 'Ex2' and 'Ex3', to save your work during this lab. Remember, by the end of the term, you are required to have the files for all the exercises in every lab. TAs and lab technicians will not be able to recover any deleted file, or any file that you have overwritten by mistake. Local Windows recycle bin doesn't work with the network M drive.
2. Run Logisim.exe following instructions in the previous lab or find it by searching for Logisim in the Windows start menu.
3. Download 'Lab 3 Report Template.docx' from the "Labs" Module, rename the file to 'Lab 3 Report `Jane Doe`.docx' where `Jane Doe` is your name, and save to your 'lab3' folder.
4. Open your lab report document and complete the first page. Read through the rest of the document so that you are aware of what needs to be recorded.

Exercise 1: Simplify a Boolean Logic Function

1. Simply the following function (F_1) to a new function (F_2) with at most 5 literals. You can review the examples discussed in the lecture to help you in solving this problem. The number of literals is the total number of inputs to a function, considering both the variables and their complements. For instance, currently F_1 has 7 literals (x, x', y, y', w, w', z).

$$F_1 = xy'z + x'y'z + w'xy + wx'y + wxy$$

2. Record and compare, in your lab notes, the number of gates that should be used to implement F_1 vs F_2. How many gates did you manage to save?
3. Build a circuit in Logisim to implement function F_1.
4. In the same file, and using the same inputs, build an independent circuit to implement F_2.
5. Connect F_1 and F_2 to a simple comparator to confirm that they are equivalent. For this design, we will use an XNOR gate, where the output of the XNOR gate is a '1' if its inputs are equal, and '0' if its inputs are different.
6. Connect a LED to the output of the XNOR gate.
7. Enable logging to 'Lab3_Ex1_log.txt', and use the poke tool () to change the values of the input bits. Notice output LED.
8. Save the circuit as 'Ex1.circ'.

Exercise 2: Convert a set of Boolean functions to a circuit

1- Implement the following four functions:

$$M_0 = A_0B_0$$

$$M_1 = A_0B_1 \oplus A_1B_0$$

$$M_2 = (A_0B_1 \,.\, A_1B_0) \oplus A_1B_1$$

$$M_3 = A_0B_1 \,.\, A_1B_0 \,.\, A_1B_1$$

Each of the inputs A and B as well as the output M should be implemented as input 'Pins' and output Pins, with data bits of 2, 2 and 4 respectively. Splitters should be used to divide the two 2-bit line inputs and the 4-bit output line to individual lines for the internal gates.

2- Enable logging of the inputs (A and B) and output M to a file name 'Lab3_Ex2_log.txt', while using radix-10.
3- Simulate the circuit by using the poke tool, to test all the possible inputs.
4- Save the circuit as 'Ex2.circ'.

Derivation of the previous equations:
As you should have observed, the above circuit implements a 2-bit × 2-bit binary multiplier. In order to find the above equations, we perform a regular multiplication as follows:

$$
\begin{array}{cccc}
 & & A_1 & A_0 \\
 & \times & B_1 & B_0 \\
\hline
 & & A_1B_0 & A_0B_0 \\
+ & A_1B_1 & A_0B_1 & \\
\hline
M_3 & M_2 & M_1 & M_0
\end{array}
$$

- Multiplication between two 1-bit inputs is just an AND gate. Hence, equation for the LSB is $M_0 = A_0B_0$ (as given above).
- A_1B_0 and A_0B_1 are added using the 'half-adder' from the previous lab since we are adding only no terms (no Carry_in). Let's denote

  $$[C_1, S_1] = \text{Half-Adder } (A_1B_0 \text{ , } A_0B_1)$$

  Please note that, for $[C, S]$ = Half-Adder $(x \text{ , } y)$, the half-adder equations would be:

  $$C = x.y,$$
  $$S = x \oplus y.$$

  Hence, in our case, we would have $C_1 = A_1B_0 \,.\, A_0B_1$ and $S_1 = A_1B_0 \oplus A_0B_1$.
  So, S_1 will be the output M_1. Hence, $M_1 = A_1B_0 \oplus A_0B_1$ (as given above).
- C_1 from the previous step would be added to A_1B_1, using another half-adder circuit since we have only two inputs to this stage of addition.

  $$[C_2, S_2] = \text{Half-Adder } (C_1, A_1B_1)$$

  S_2 will be the output M_2. Hence, $M_2 = S_2 = (A_1B_0 \,.\, A_0B_1) \oplus A_1B_1$ (as given above).
- Finally, M_3 would be the carry out from the previous addition stage:

  $$M_3 = C_2 = A_1B_0 \,.\, A_0B_1 \,.\, A_1B_1 \text{ (as given above).}$$

Exercise 3: Design a 3-bit by 2-bit binary multiplier

1. Derive the function for the full adder studied in the previous lab.
2. Derive a closed form formulation to the multiplication of a 3-bit input $[A_2, A_1, A_0]$ by a 2-bit input $[B_1, B_0]$. The output would be $[M_4, M_3, M_2, M_1, M_0]$
3. Build the corresponding circuit.
   Please note, in Logisim, they used a not very common definition for the XOR gates with 3 or more inputs instead of the industry common standard of using it as a parity (equals to cascading two XOR gates, each with 2 inputs). Therefore, in Logisim, use only XOR gates with two inputs. Do not use XOR gates with more than 2 inputs in Logisim.
4. Enable logging of the inputs (A and B) and output M to a file name 'Lab3_Ex3_log.txt', while using radix-10
5. Simulate the circuit by using the poke tool, to test all the possible inputs.
6. Save the circuit as 'Ex3.circ'.

Congratulations. Now you can design a fully functioning MUL instruction with arbitrary size inputs/outputs.

Demonstration:

- In order to get full credit, complete all exercises, then show the TA your work and your completed lab report, and be prepared to answer questions asked by the TA.
- Save your lab report as pdf with the same file name except the file type will now be 'pdf'. **Submit your report in Brightspace right after the TA checks your work.**
- If you are not able to complete the lab during the scheduled lab time, show the TA the work that you have completed and submit your lab.
    - Your mark will be based on the work completed during the lab.
    - After the lab, finish up the lab on your own time.
        - You are encouraged to submit an updated lab report, but your mark will not increase.
- **Note:** If a student does not submit their work, 0 marks are given for the lab.
- **Note:** If a student does not show their work to a TA, 0 marks are given for the lab.