

SYSC2310 A and B Introduction to Digital Systems Fall 2022

Lab 2 Report Basic Combinational Circuits

Student Name: _Uchenna Obikwelu_

Student ID: _101241887_

TA Name: _____Ethan_____

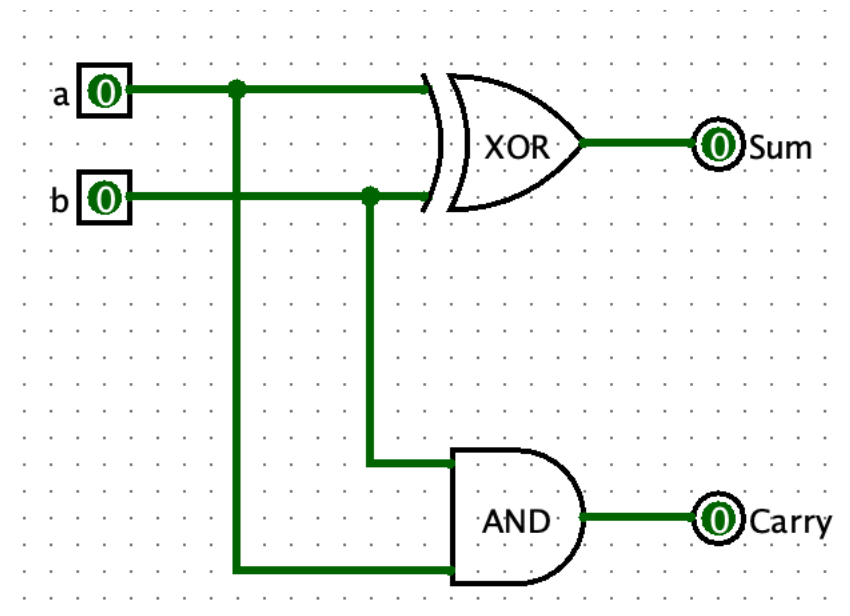
Lab Section: _SYSC 2310 L20__

Lab Date: 3rd October 2022

Date Completed: 3rd October 2022

Exercise 1: Build a 1-bit half adder

Screen Shot of Exercise 1 Circuit

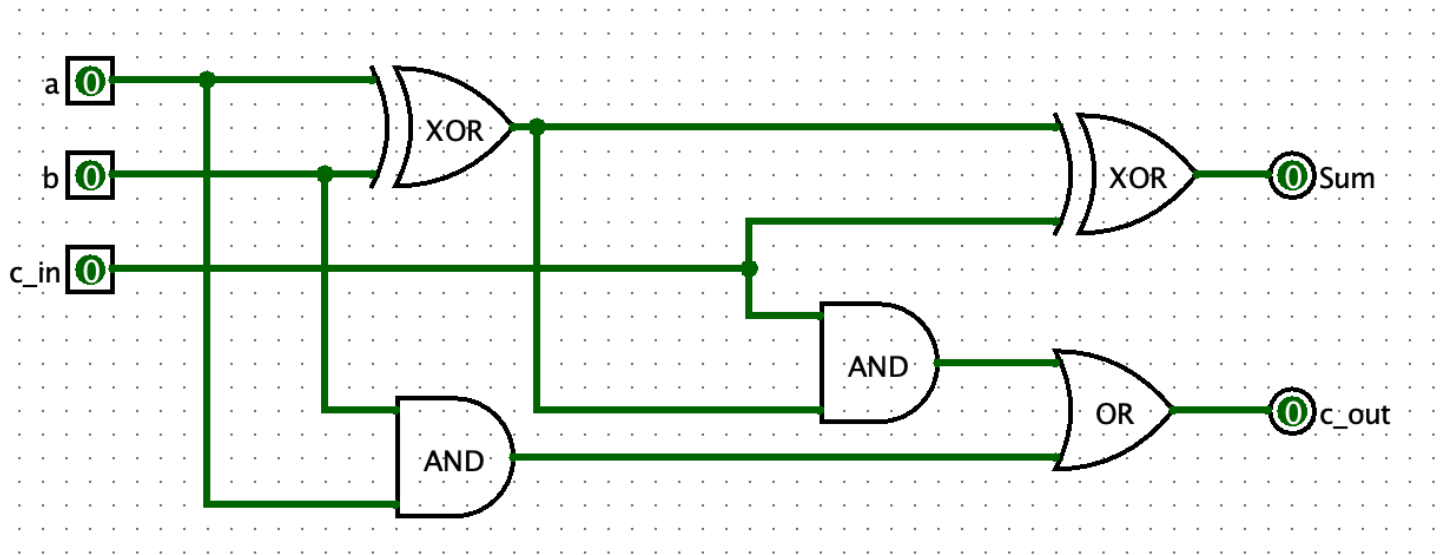


Copy and Paste Exercise 1 Simulation Logging File

a	b	Sum	Carry
0	0	0	0
0	1	1	0
0	0	0	0
1	0	1	0
1	1	0	1

Exercise 2: Build a 1-bit full adder

Screen Shot of Exercise 2 Circuit

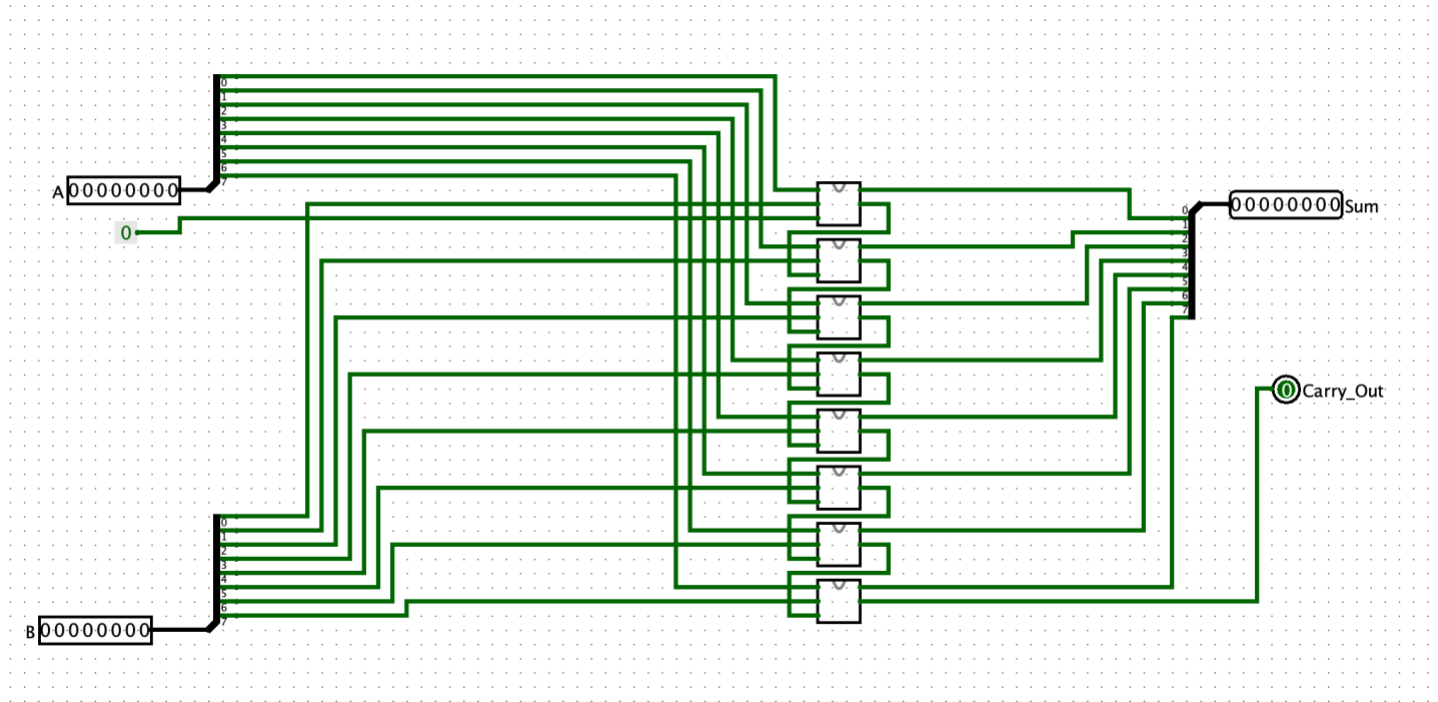


Copy and Paste Exercise 2 Simulation Logging File

a	b	c_in	c_out	Sum
1	1	1	1	1
0	1	1	1	0
0	1	0	0	1
1	1	0	1	0
1	0	0	0	1
1	0	1	1	0
0	0	1	0	1
0	1	1	1	0
0	1	0	0	1
0	0	0	0	0

Exercise 3: Build an 8-bit full adder by using 8 1-bit full adders

Screen Shot of Exercise 3 Circuit

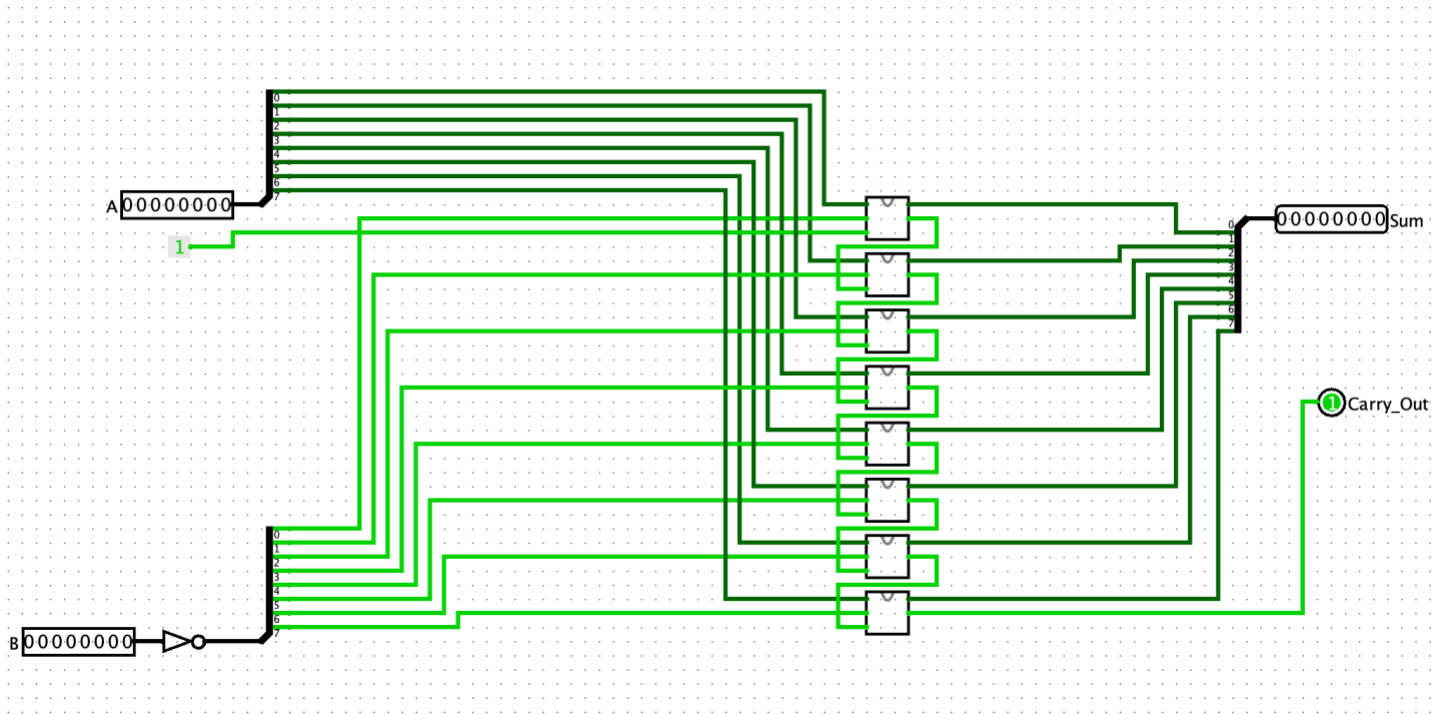


Copy and Paste Exercise 3 Simulation Logging File

A	B	Carry_Out	Sum
0	0	0	0
16	0	0	16
16	16	0	32
16	48	0	64
48	48	0	96
48	56	0	104
52	56	0	108
60	56	0	116
61	56	0	117
63	56	0	119
63	57	0	120
63	59	0	122
63	63	0	126
191	63	0	254
191	127	1	62
191	255	1	190
183	255	1	182
182	255	1	181
190	255	1	189
188	255	1	187

Exercise 4: Build an 8-bit subtractor (A-B) by using an 8-bit full adder

Screen Shot of Exercise 4 Circuit



Copy and Paste Exercise 4 Simulation Logging File

A	B	Carry_Out	Sum
0	0	1	0
1	0	1	1
3	0	1	3
7	0	1	7
23	0	1	23
55	0	1	55
55	1	1	54
55	3	1	52
55	7	1	48
119	7	1	112
119	5	1	114
119	13	1	106
119	12	1	107
119	8	1	111
117	8	1	109
101	8	1	93
69	8	1	61
197	8	1	189
197	40	1	157
197	56	1	141
197	48	1	149
197	32	1	165
197	33	1	164

INTRODUCTION TO DIGITAL SYSTEMS

SYSC2310 (Fall 2022)

Lab 2: Basic Combinational Circuits

Objectives

- Build a 1-bit half adder and 1-bit full adder.
- Introduce the concept of hierarchy by building an 8-bit adder, using the 1-bit full adder.
- Build an 8-bit subtractor, using the 8-bit full adder.

Relation to course outcomes

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

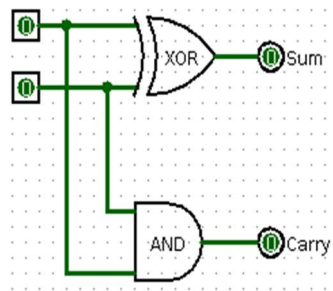
- Understand the basic building blocks of digital systems.
- Apply analysis skills to correctly describe the behavior of given combinational logic circuits.

Preparation

1. Under your 'sysc2310lab/lab2' folder, create four folders 'Ex1', 'Ex2', 'Ex3' and 'Ex4', 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 2 Report Template.docx' from the "Labs" Module, rename the file to 'Lab 2 Report Jane Doe.docx' where Jane Doe is your name, and save to your 'lab2' 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: Build a 1-bit half adder

Build the following circuit, which represents a 1-bit half adder:

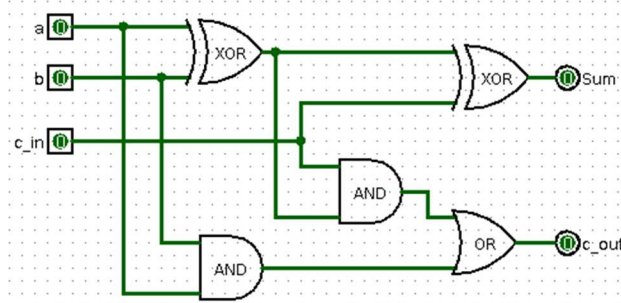


- 1- Simulate the circuit and log the results in a file by logging to a file name 'Lab2_Ex1_log.txt'
- 2- Simulate the circuit by using the poke tool, to test all the possible inputs.
- 3- Save the circuit as 'Ex1.circ'.

A half adder is an adder that accepts 2 input bits and generates the sum and the carry. The half adder does not accept an input carry. I.e., It can be used to add the LSB of two binary numbers where there is no carry_in. Next, we will build a full adder, which accepts two inputs and a carry_in.

Exercise 2: Build a 1-bit full adder

- 1- Copy the circuit 'Ex1.circ' from 'Ex1' folder to the 'Ex2' folder. Rename the file to 'Ex2.circ'. From Logisim, File, Open, 'Ex2.circ'
- 2- Modify the circuit to be as follows, which represents a 1-bit full adder:

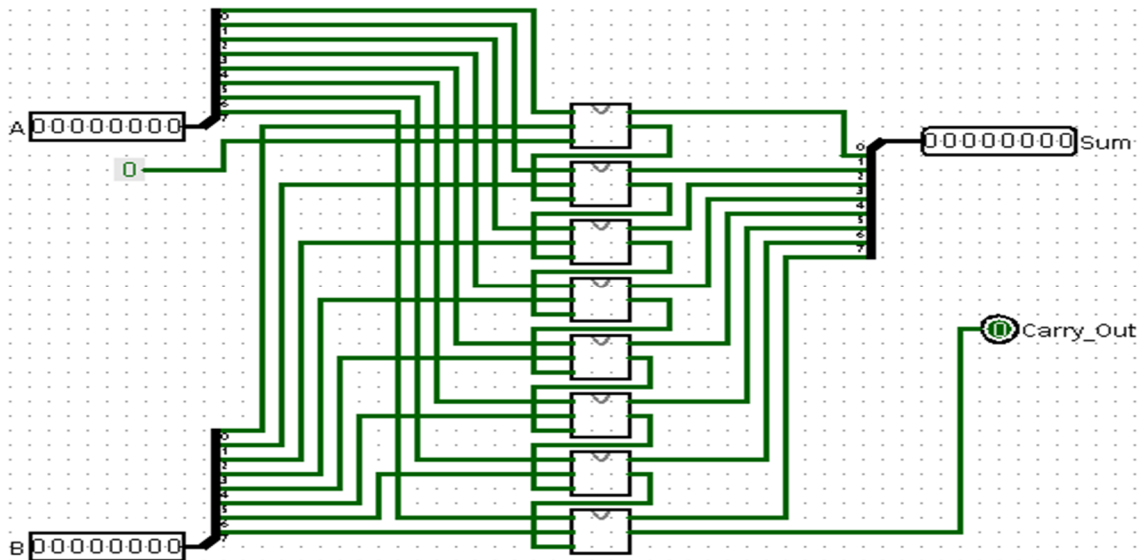


- 3- Enable logging to a file name 'Lab2_Ex2_log.txt'
- 4- Simulate the circuit by using the poke tool, to test all the possible inputs.
- 5- Save the circuit as 'Ex2.circ'.

A full adder is an adder that accepts 3 input bits (a, b and a carry_in) and generates the sum and the carry_out.

Exercise 3: Build an 8-bit full adder by using 8 1-bit full adders

- 1- Again, copy the 'Ex2.circ' from the 'Ex2' folder to the 'Ex3' folder. Rename it to 'Ex3.circ'. Note that, we need to use the previous 1-bit full adder as a module inside a bigger circuit. Here, we will try to build the following circuit:



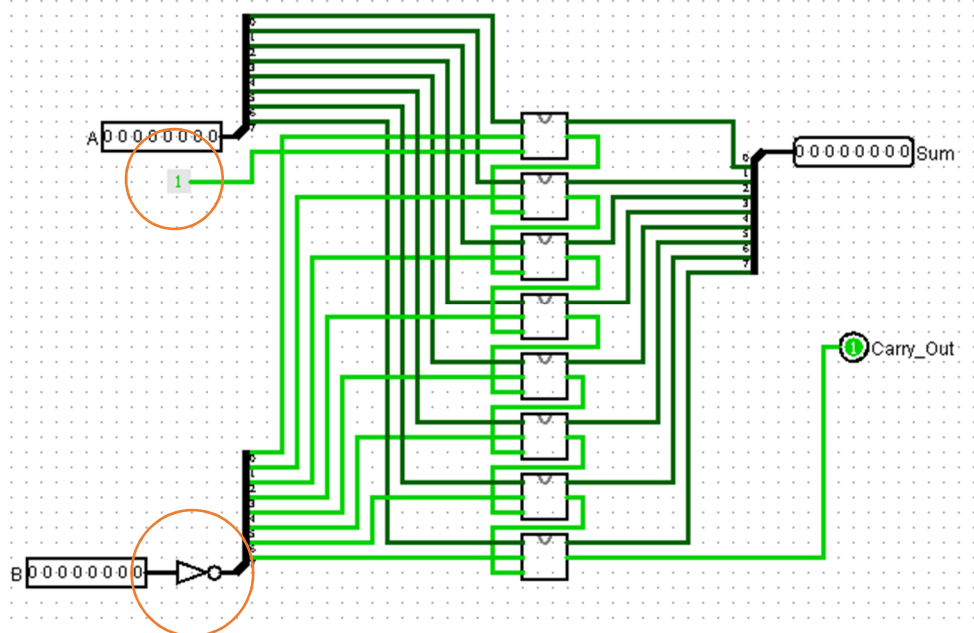
- 2- From the Explorer Pane, select the main circuit to show its attributes. Change the 'Circuit Name' to '1b_full_adder'.
- 3- From the Menu bar, click 'Project', 'Add Circuit', and name it 'main'. Now, we have an empty 'main' circuit and a module that is called '1b_full_adder'.
- 4- Add 8 copies of the module '1b_full_adder'.
- 5- Add 2 input Pins (Ctrl-4), and change their 'Data Bits' attribute to 8 bits, name them A and B.
- 6- Add a 'Constant' from the wiring library. Change its value to 0x0. This should work as a carry_in to the LSB.
- 7- Also, add 1 output Pin (Ctrl-5), and change its 'Data Bits' attribute to 8 bits, with name Sum.
- 8- Add 1 output Pin (Ctrl-5), with name 'Carry_Out'.
- 9- Add 2 Splitters (Input) from the wiring library. Change its attributes as follows: 'Bit Width in': '8' and Fan Out to '8'.
- 10- Add 1 Splitters (output) from the wiring library. Change its attributes as follows: 'Bit Width in': '8' and Fan Out to '8' and facing 'West'.
- 11- Connect all the wires. Essentially, we connect bit 0 from input A (a0) and bit 0 from input B (b0) the inputs of the first 1-bit full adder module (the top one). a1 and b1 to the second module, and so on. The carry_in of the first module is the constant 0. The carry_in for all the other modules is the carry_out of the previous module. Carry_out of the last module is the overall carry_out. Also connect the output sum. Make sure that all the wires are dark green and there are no green dots (→), which indicates two wires are cross connected to each other.
- 12- Enable logging to a file name 'Lab2_Ex3_log.txt', and add 'A', 'B', 'Sum' and the 'Carry_out' change the radix of all of them to 10, to make it easy for the TA to review your work.

- 13- Simulate the circuit by using the poke tool, to test some of the possible inputs.
- 14- Save the circuit as 'Ex3.circ'.

You have just designed a fully functioning ADD instruction that could be used in a simple 8-bit CPU.

Exercise 4: Build an 8-bit subtractor (A-B) by using an 8-bit full adder

- 1- Again, copy the 'Ex3.circ' from the 'Ex3' folder to the 'Ex4' folder. Rename it to 'Ex4.circ'.
- 2- Modify the circuit to be similar to the following figure by the following two changes:
 - a. Add a NOT Gate after the input B. Get a Not gate from the library of gates and change its 'Data bits' attribute to 8.
 - b. Change the 'value' attribute of the constant connected to the Carry_in to '0x1'.



- 3- Enable logging to a file name 'Lab2_Ex4_log.txt', and add 'A', 'B', 'Sum' and the 'Carry_out' change the radix of all of them to 10, to make it easy for the TA to review your work.
- 4- Simulate the circuit by using the poke tool, to test some of the possible inputs. In your testing, try to keep the input A greater than the input B so the output would be correct by discarding the carry out. Otherwise (if B is greater than A), the output would be represented by its 2's complement.
- 5- Save the circuit as 'Ex4.circ'.

Congratulations. You have just designed a fully functioning subtraction instruction that could be used in a simple 8-bit CPU.

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.