EEL 3701C: Digital Logic & Computer Systems
Revision **0**
Lab # Report: **Lab 1**

Poche, Natalie
Class #: 11198
PI Name: Jaiden Magnan
February 6, 2025

REQUIREMENTS NOT MET

N/A

VIDEO FILE LINK

https://youtu.be/a4qPu9fNiyM

PROBLEMS ENCOUNTERED

Had some trouble figuring out the lights but managed to make it work with some outside help.

FUTURE WORK/APPLICATIONS

The lab gives us more practice in Quartus while utilizing different projects. Additionally, through this lab we work with mixed logic and truth tables as well as correlating something we built in Quartus into a working circuit on our breadboards. This is beneficial as we get used to connecting circuits as future labs can have more challenging conecpts.

PRE-LAB QUESTIONS OR EXERCISES

N/A

PRE-LAB REQUIREMENTS (Design, Schematic, ASM Chart, VHDL, etc.)

Each section of the pre-lab requirements should be completed separately, and in order. Include each of the following items in order. Note that some of these items will not apply to every lab. Anything scanned or copied ***must be clear and legible***.

- Logic equations. (Note that logic equations do not contain activation levels.)
- **All** tables and figures should have captions with Figure/Table numbers and a description of for which part of the lab it references, e.g., ***Table 3****: Truth Table for Part B*.
- Truth tables and voltage tables (and/or next-state truth tables).
- When applicable, include Karnaugh Maps (i.e., K-Maps).
- Include hand-drawn circuits (when required). Label all input and output activation-levels and intermediate equations in the circuits.
- Include screenshots of the BDF designs of circuits.
  - Label all input and output activation-levels, i.e., use _L suffix for active-low signals and no suffix for active-high signals. **Add chip and pin numbers to any schematic that will be constructed.**
  - Images should be large enough so that inputs, outputs, labels, and parts are clearly visible and distinguishable to any reader.
  - Each BDF should have the following info on the top left corner (similar to the top right of this page):

    Last Name, First Name
    Lab #, Part #
    Class #
    PI Name:
    Description: (short description of what is to be accomplished in the design; perhaps an equation)

  - In Windows, I use the ***Snipping Tool***, which is now built into Windows. Just type "snip" in the Windows search box and then select ***Snipping Tool***.
- When necessary, include ASM Charts. These can be hand-drawn, but clear and legible. We recommend that you use resources like https://www.draw.io/ to create computer-generated ASMs.
- Truth tables or next-state truth tables should have the following characteristics.
  - Can be either typed or hand-written and scanned (must be clear and legible)
  - Must be in **counting order** (i.e., inputs of 000, 001, 010, 011, …, 111)
  - Clearly distinguish inputs from outputs (see the example below that uses a thick line)
  - If you are designing a state machine or a controller, clearly indicate and separate signal values both before the clock and after the clock (i.e., Q1 and $Q1^+$, respectively)
  - Tip: divide rows into consecutive groups of 4 (or 2 or 8) to make it easier for both you and your PI to read.
  - **Example**

A	B	C	Y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

Table: Caption for above table. It should reference the part of the lab, e.g., "Table 3: Truth Table for Part B."

- Voltage tables should have the following characteristics.
  - Must be in counting order (i.e., inputs of LLL, LLH, LHL, … , HHH)
  - Use similar formatting to truth tables (described above)

- **Example**

A(H)	B(H)	C(L)	Y(L)
L	L	L	H
L	L	H	H
L	H	L	L
L	H	H	H
H	L	L	H
H	L	H	H
H	H	L	H
H	H	H	L

- Include **meaningfully annotated** functional simulations.
  - Using the grouping tool, ***group as many signals together as possible*** (when it makes sense to group them)! If you are simulating a basic logic equation, group all the inputs together. If you are simulating a circuit that includes MSI elements, group signals of the form X_{N-0}. The most-significant bit should appear first, ending with the least-significant bit. If you are simulating an ALU, group the buses together as just described.
  - Not every row of your voltage table must be annotated in the waveform simulation, but your choices of rows that you annotate must be ***encompassing***
  - If you are designing a state machine or a controller, your CLK signal should appear ***at the top*** of your inputs and outputs. The general order is CLK -> Reset -> state bits -> inputs -> outputs.
  - ***Tip***: Use Microsoft Paint to annotate your waveforms. An alternative is to print out the waveforms, annotate them by hand, then scan and upload
  - ***Hint***: Consider a truth table where the output is true in significantly less cases than it is false (or vice versa). If the output signal is active-high, it would be wise to annotate only the cases where the output voltage is HIGH.
- Include every line of VHDL programs, including both ***architecture*** and ***behavior*** sections.
- Include every line of any **MIF** files. If these are associated with assembly language programs, you can either put the assembly code as comments or separately include assembly language programs

Lab 1 Part 1: Introduction to Positive Logic

$V = \overline{(A+\overline{B})\cdot\overline{(C\cdot\overline{D})}}$ Active High: A, B, C, D, V Active Low: None

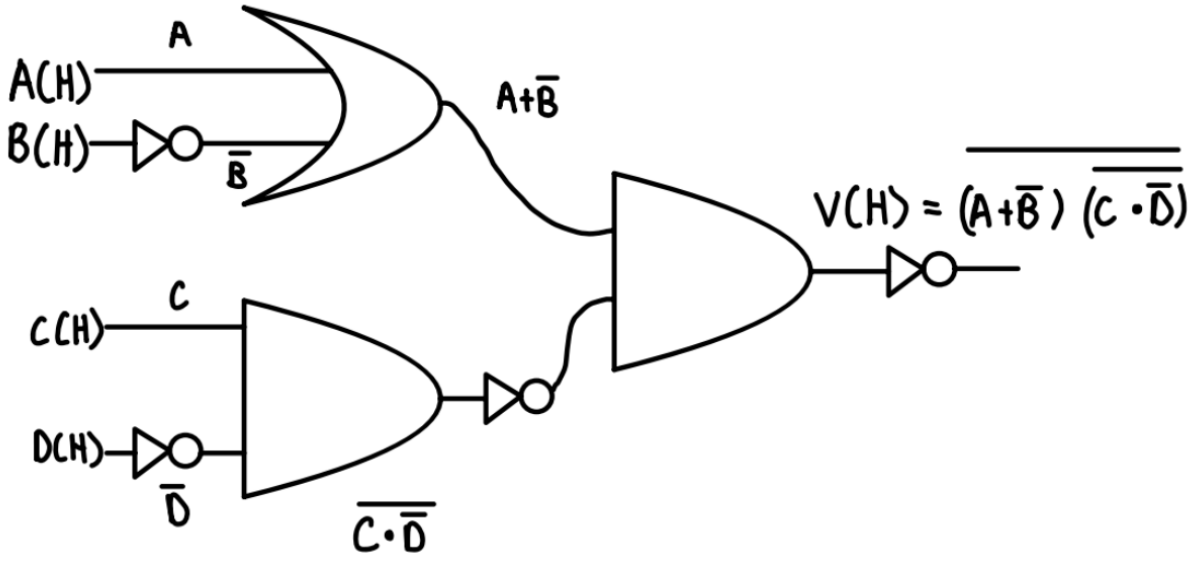


Table 1: Lab 1 Part 1.1 - Circuit Design with no AND or OR gate bubbles

$W = \overline{(A+\overline{B})\cdot\overline{(C\cdot\overline{D})}}$ Active High: A, B, C, D, V Active Low: None

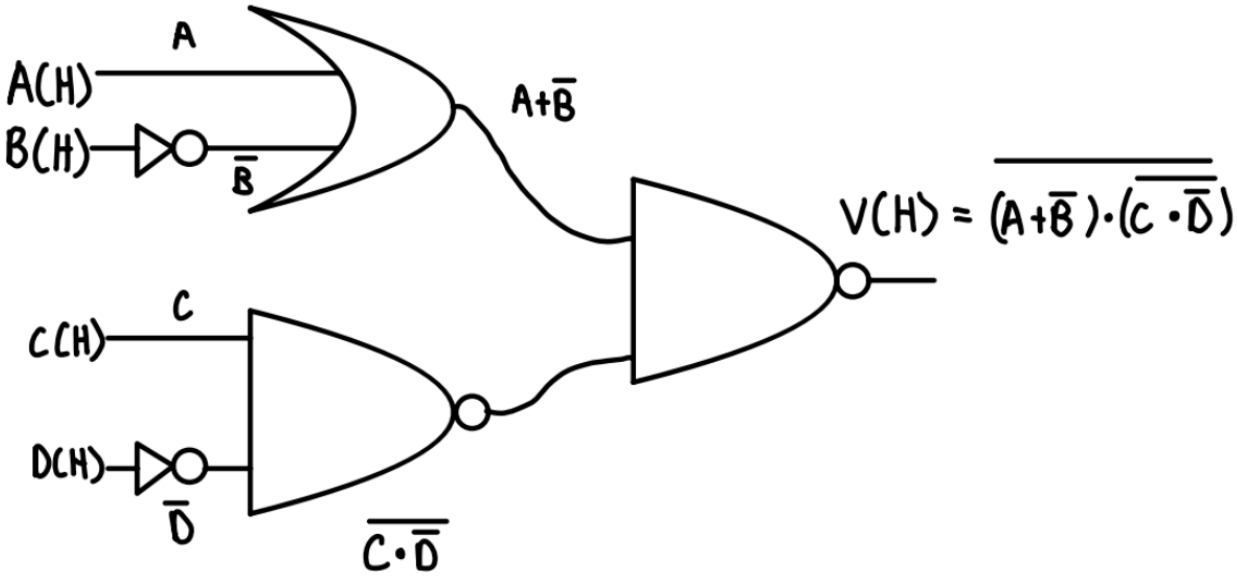


Table 2: Lab 1 Part 1.2 - Circuit Design with NAND Gates

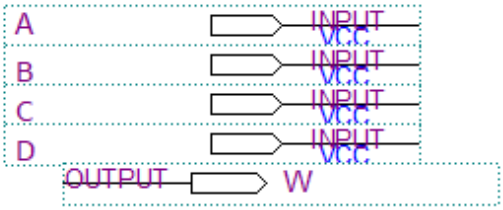
Lab 1, Part A

Name: Natalie Poche

Class #: 11198

PI Name: Jaiden Magnan

Description: W = /[(A + /B) * /(C * /D)]



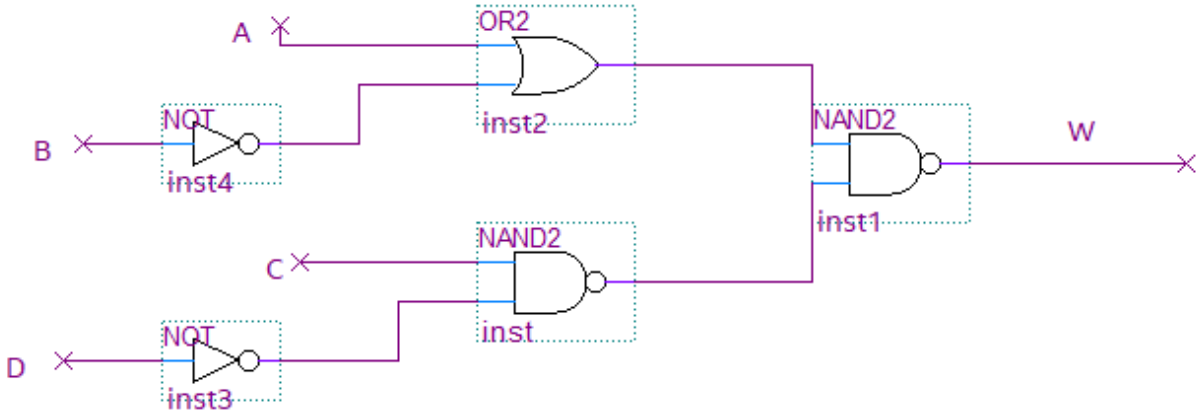


Table 3: Lab 1 Part 1.3 – BDF Circuit Design for W

A	**B**	**C**	**D**	/B	/D	A + /B	C * /D	/(C * /D)	(A + /B) * /(C * /D)	W
0	0	0	0	1	1	1	0	1	1	0
0	0	0	1	1	0	1	0	1	1	0
0	0	1	0	1	1	1	1	0	0	1
0	0	1	1	1	0	1	0	1	1	0
0	1	0	0	0	1	0	0	1	0	1
0	1	1	0	0	1	0	1	0	0	1
0	1	1	1	0	0	0	0	1	0	1
1	0	0	0	1	1	1	0	1	1	0
1	0	0	1	1	0	1	0	1	1	0
1	0	1	0	1	1	1	1	0	0	1
1	0	1	1	1	0	1	0	1	1	0
1	1	0	0	0	1	1	1	0	0	1
1	1	1	0	0	1	1	1	0	0	1
1	1	1	1	0	0	1	0	1	1	0

Table 4: Lab 1 Part 1.4 - Truth Table for W

A(H)	B(H)	C(H)	D(H)	B(L)	D(L)	A + /B	C * /D	/(C * /D)	(A + /B) * /(C * /D)	W(H)
L	L	L	L	H	H	H	L	H	H	L
L	L	L	H	H	L	H	L	H	H	L
L	L	H	L	H	H	H	H	L	L	H
L	L	H	H	H	L	H	L	H	H1	L
L	H	L	L	L	H	L	L	H	L	H
L	H	H	L	L	H	L	H	L	L	H
L	H	H	H	L	L	L	L	H	L	H
H	L	L	L	H	H	H	L	H	H	L
H	L	L	H	H	L	H	L	H	H	L
H	L	H	L	H	H	H	H	L	L	H
H	L	H	H	H	L	H	L	H	H	L
H	H	L	L	L	H	H	H	L	L	H
H	H	H	L	L	H	H	H	L	L	H
H	H	H	H	L	L	H	L	H	H	L

Table 5: Lab 1 Part 1.5 – Voltage Table for W

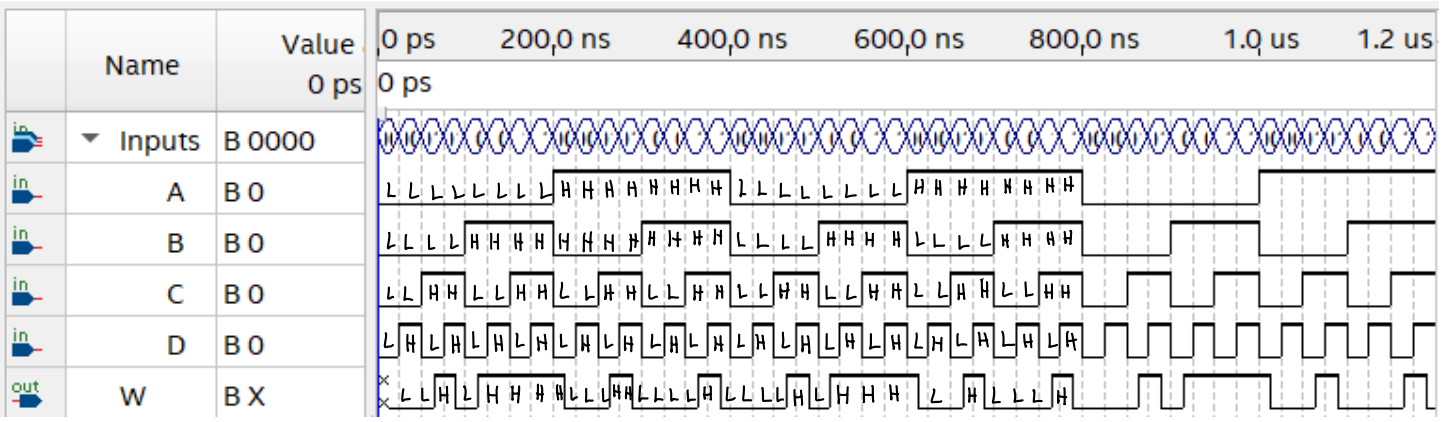


Table 6: Lab 1 Part 1.6 – Time Simulation Results from BDF of W

Lab 1 Part 3: Introduction to Mixed Logic

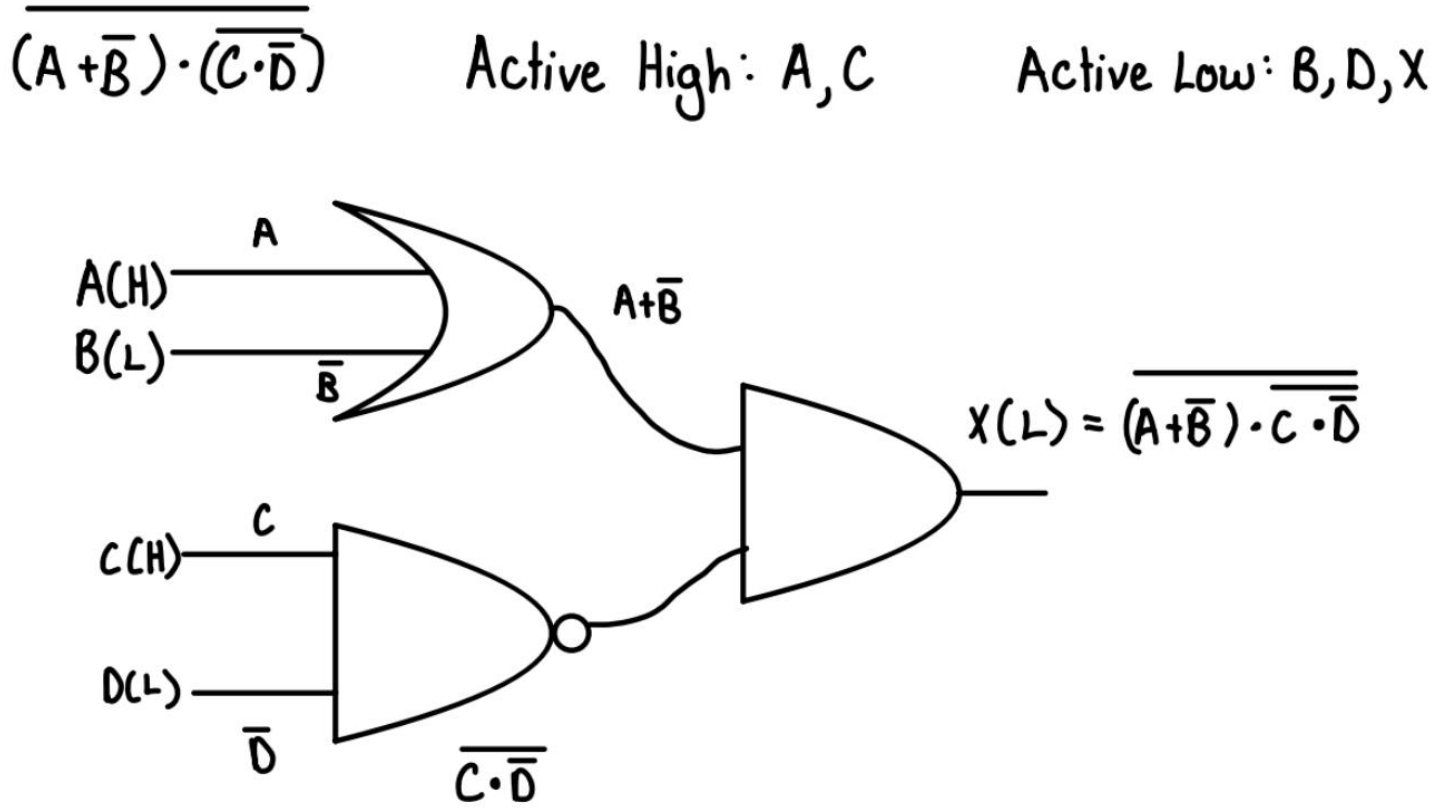


Table 7: Lab 1 Part 3.1 - Circuit Design for X

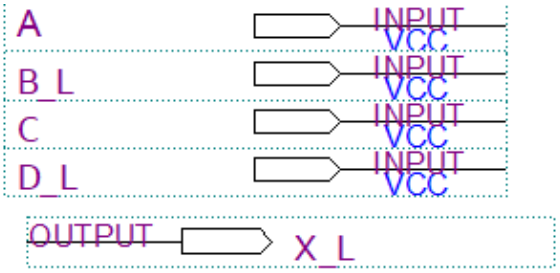
Lab 1, Part B

Name: Natalie Poche

Class #: 11198

PI Name: Jaiden Magnan

Description: X = /[(A +/B) * /(C * /D)]



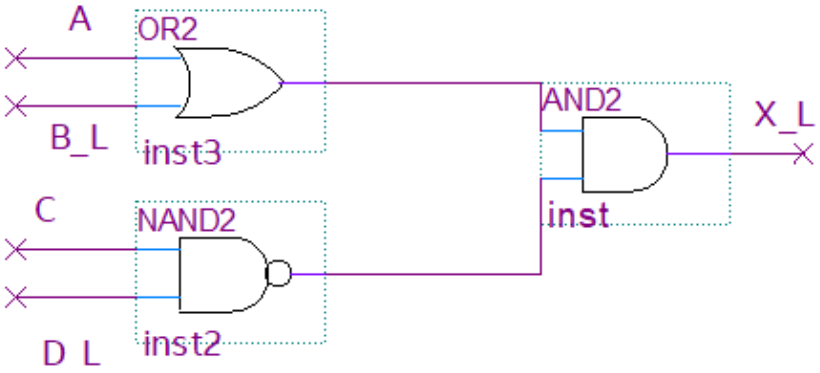


Table 8: Lab 1 Part 3.2 – BDF Circuit Design for X

A	**B**	**C**	**D**	/B	/D	A + /B	C * /D	/(C * /D)	X
0	0	0	0	1	1	1	0	1	1
0	0	0	1	1	0	1	0	1	1
0	0	1	0	1	1	1	1	0	0
0	0	1	1	1	0	1	0	1	1
0	1	0	0	0	1	0	0	1	0
0	1	1	0	0	1	0	1	0	0
0	1	1	1	0	0	0	0	1	0
1	0	0	0	1	1	1	0	1	1
1	0	0	1	1	0	1	0	1	1
1	0	1	0	1	1	1	1	0	0
1	0	1	1	1	0	1	0	1	1
1	1	0	0	0	1	1	1	0	0
1	1	1	0	0	1	1	1	0	0
1	1	1	1	0	0	1	0	1	1

Table 9: Lab 1 Part 3.3 – Truth Table for X

A(H)	B(L)	**C(H)**	D(L)	X(L)
L	H	L	H	H
L	H	L	L	H
L	H	H	H	L
L	H	H	L	H
L	L	L	H	L
L	L	H	H	L
L	L	H	L	L
H	H	L	H	H
H	H	L	L	H
H	H	H	H	L
H	H	H	L	H
H	L	L	H	L
H	L	H	H	L
H	L	H	L	H

Table 10: Lab 1 Part 3.3 – Voltage Table for X

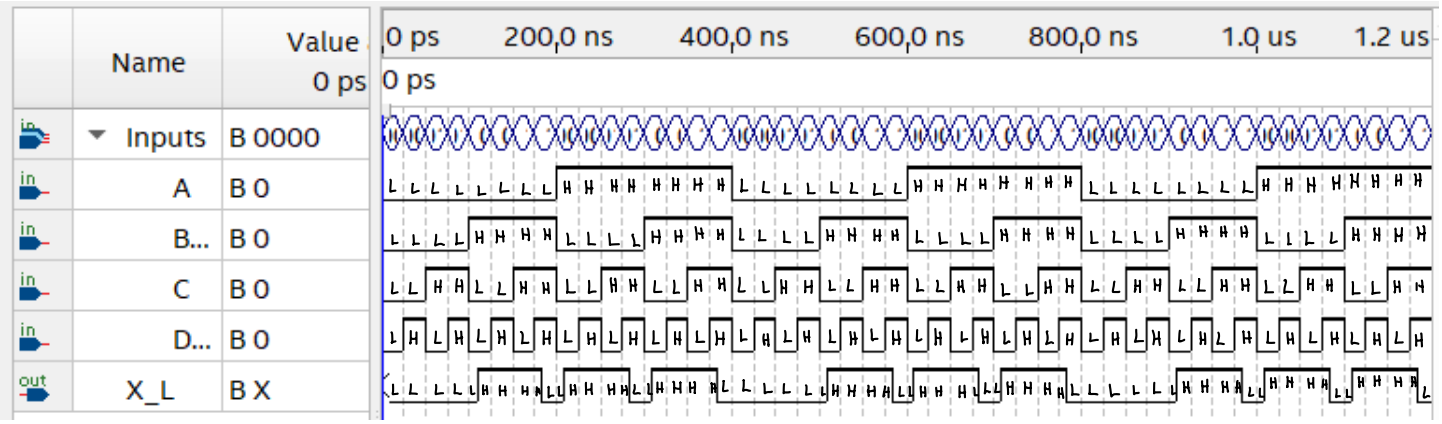
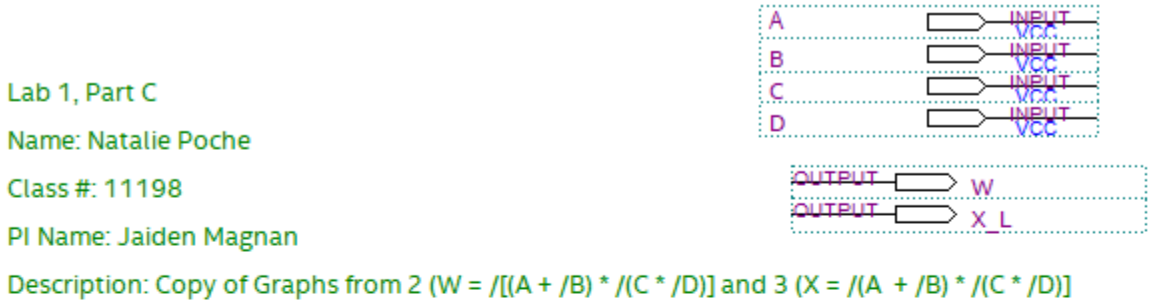


Table 11: Lab 1 Part 3.3 – Time Simulation Results for X

Lab 1 Part 4: Conversion Between Activation Levels



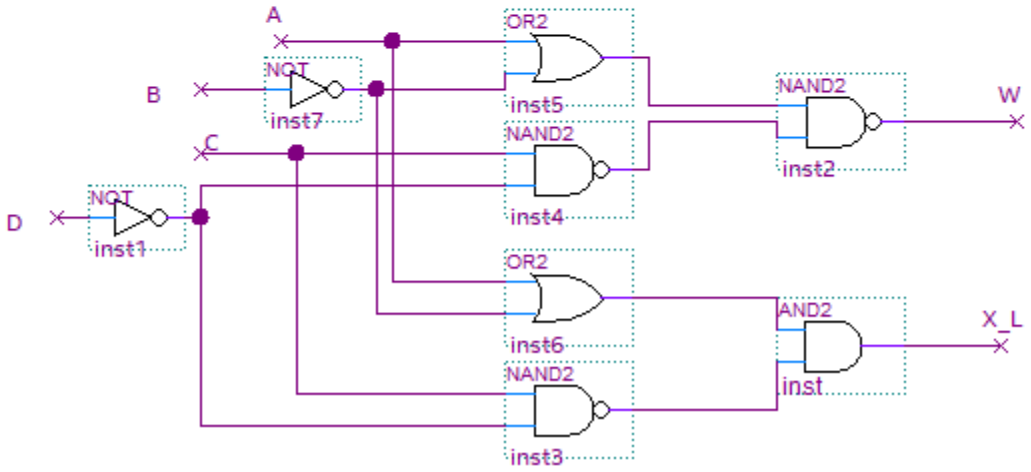


Table 12: Lab 1 Part 4.1 – BDF of W and X

A	B	C	D	/B	/D	A + /B	C * /D	/(C * /D)	X	W
0	0	0	0	1	1	1	0	1	1	0
0	0	0	1	1	0	1	0	1	1	0
0	0	1	0	1	1	1	1	0	0	1
0	0	1	1	1	0	1	0	1	1	0
0	1	0	0	0	1	0	0	1	0	1
0	1	1	0	0	1	0	1	0	0	1
0	1	1	1	0	0	0	0	1	0	1
1	0	0	0	1	1	1	0	1	1	0
1	0	0	1	1	0	1	0	1	1	0
1	0	1	0	1	1	1	1	0	0	1
1	0	1	1	1	0	1	0	1	1	0
1	1	0	0	0	1	1	1	0	0	1
1	1	1	0	0	1	1	1	0	0	1
1	1	1	1	0	0	1	0	1	1	0

Table 13: Lab 1 Part 4.2 – Truth Table for W and X

A(H)	B(H)	C(H)	D(H)	X(L)	W(H)
L	L	L	L	H	L
L	L	L	H	H	L
L	L	H	L	L	H
L	L	H	H	H1	L
L	H	L	L	L	H
L	H	H	L	L	H
L	H	H	H	L	H
H	L	L	L	H	L
H	L	L	H	H	L
H	L	H	L	L	H
H	L	H	H	H	L
H	H	L	L	L	H
H	H	H	L	L	H
H	H	H	H	H	L

Table 14: Lab 1 Part 4.2 – Voltage Table for W and X

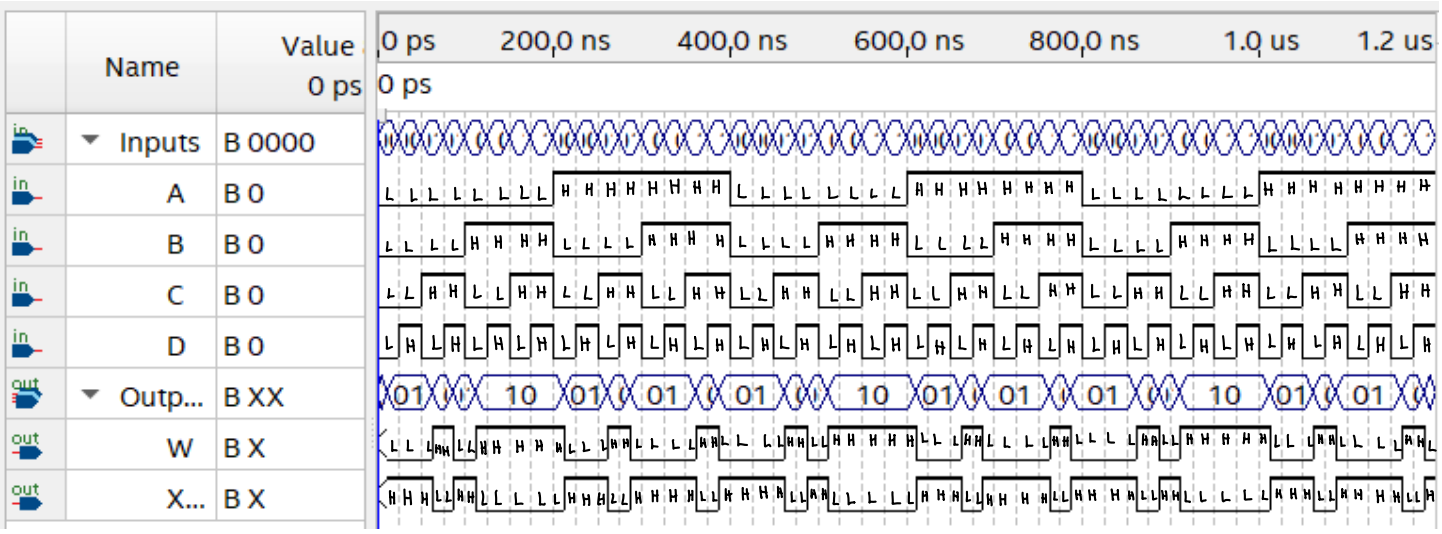


Table 15: Lab 1 Part 4.2 – Wave Simulation Result for W and X

Lab 1 Part 5: Logic Minimization Using Mixed Logic

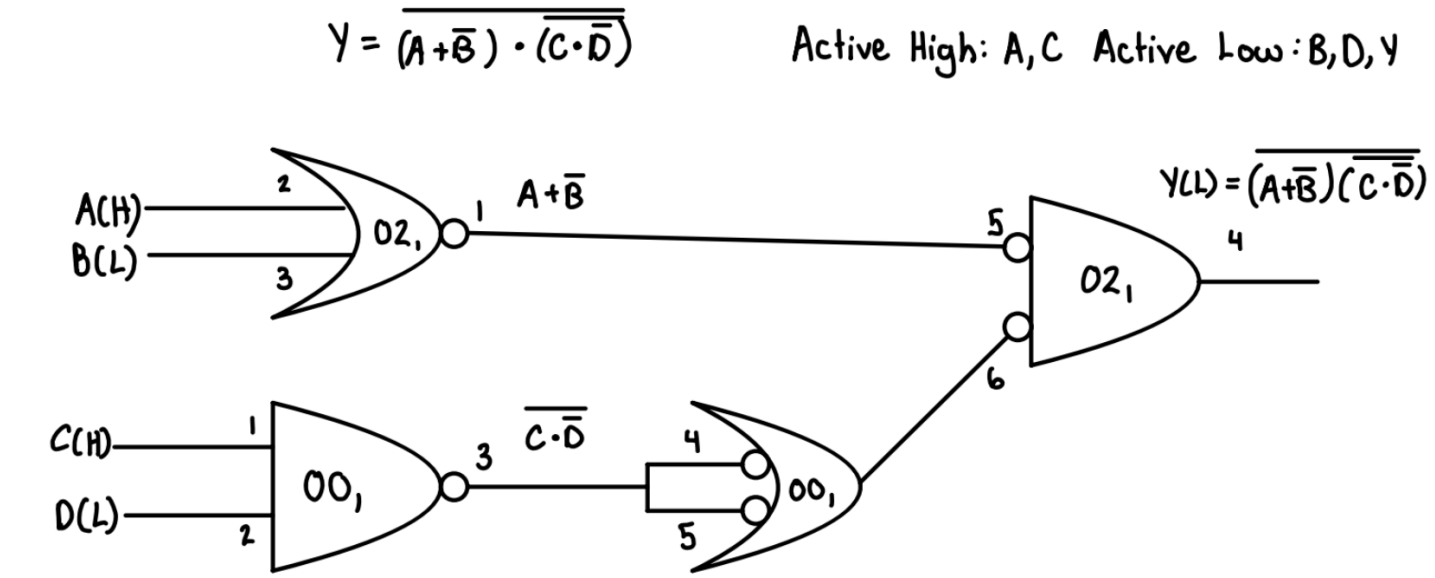


Table 16: Lab 1 Part 5.1 - Circuit Diagram for Y

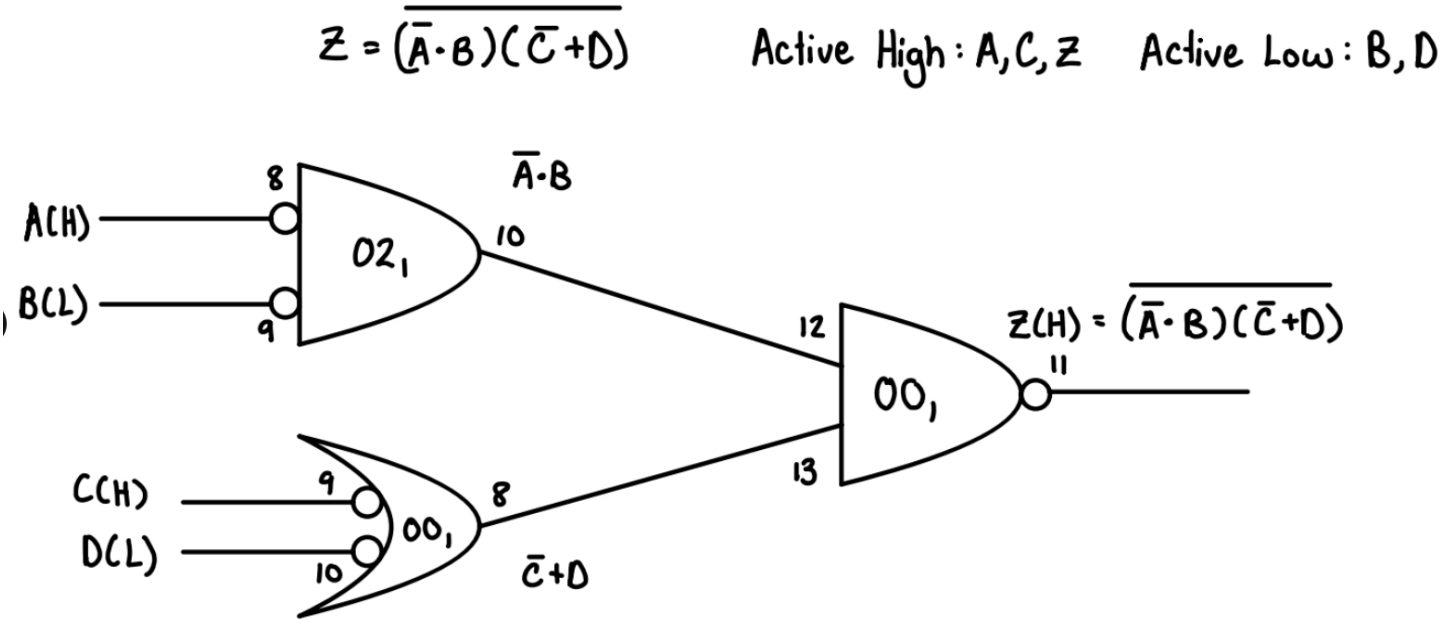


Table 17: Lab 1 Part 5.1 - Circuit Diagram for Z

A	**B**	**C**	**D**	/B	/D	A + /B	C * /D	/(C * /D)	(A + /B) * /(C * /D)	Y
0	0	0	0	1	1	1	0	1	1	0
0	0	0	1	1	0	1	0	1	1	0
0	0	1	0	1	1	1	1	0	0	1
0	0	1	1	1	0	1	0	1	1	0
0	1	0	0	0	1	0	0	1	0	1
0	1	1	0	0	1	0	1	0	0	1
0	1	1	1	0	0	0	0	1	0	1
1	0	0	0	1	1	1	0	1	1	0
1	0	0	1	1	0	1	0	1	1	0
1	0	1	0	1	1	1	1	0	0	1
1	0	1	1	1	0	1	0	1	1	0
1	1	0	0	0	1	1	1	0	0	1
1	1	1	0	0	1	1	1	0	0	1
1	1	1	1	0	0	1	0	1	1	0

Table 18: Lab 1 Part 5.2 - Truth Table for Y

A	B	C	D	/A	/C	/A * B	/C + D	(/A * B) * (/C + D)	Z
0	0	0	0	1	1	0	1	0	1
0	0	0	1	1	1	0	1	0	1
0	0	1	0	1	0	0	0	0	1
0	0	1	1	1	0	0	1	0	1
0	1	0	0	1	1	1	1	1	0
0	1	1	0	1	0	1	0	0	1
0	1	1	1	1	0	1	1	1	0
1	0	0	0	0	1	0	1	0	1
1	0	0	1	0	1	0	1	0	1
1	0	1	0	0	1	0	1	0	1
1	0	1	1	0	1	0	1	0	1
1	1	0	0	0	0	0	0	0	1
1	1	1	0	0	0	0	0	0	1
1	1	1	1	0	0	0	1	0	1

Table 19: Lab 1 Part 5.2 - Truth Table for Z

A(H)	B(L)	**C(H)**	D(L)	Y(L)
L	H	L	H	H
L	H	L	L	H
L	H	H	H	L
L	H	H	L	H1
L	L	L	H	L
L	L	H	H	L
L	L	H	L	L
H	H	L	H	H
H	H	L	L	H
H	H	H	H	L
H	H	H	L	H
H	L	L	H	L
H	L	H	H	L
H	L	H	L	H

Table 20: Lab 1 Part 5.3 - Voltage Table for Y

A(L)	B(H)	C(L)	D(H)	Z(H)
H	L	H	L	H
H	L	H	H	H
H	L	L	L	H
H	L	L	H	H
H	H	H	L	L
H	H	L	L	H
H	H	L	H	L
L	L	H	L	H
L	L	H	H	H
L	L	H	L	H
L	L	H	H	H
L	H	L	L	H
L	H	L	L	H
L	H	L	H	H

Table 21: Lab 1 Part 5.3 - Voltage Table for Z

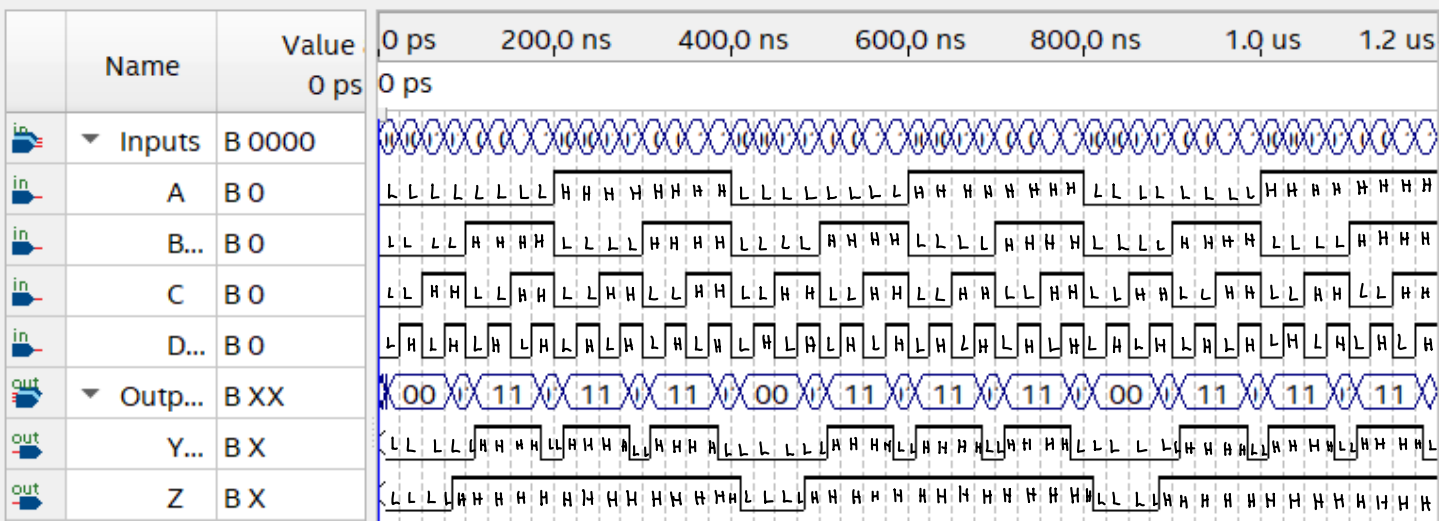
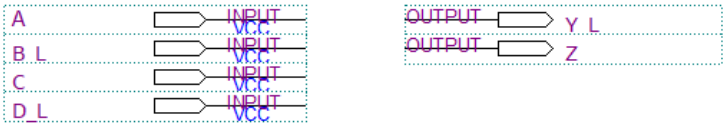


Table 22: Lab 1 Part 5.3 - Time Simulation Result for Y and Z

Lab 1, Part D
Name: Natalie Poche
Class #: 11198
PI Name: Jaiden Magnan
Description: Implementation of Y = /[(A + /B) * /(C * /D)] and Z = /[(/A * B) * (/C * D)]



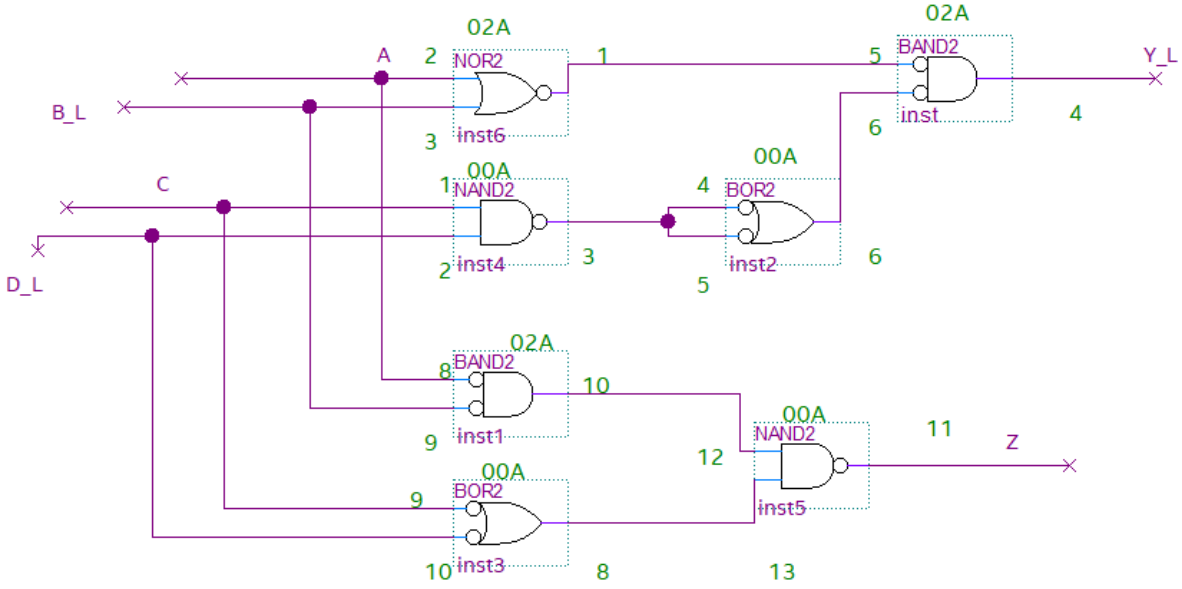


Table 23: Lab 1 Part 5.4 - BDF of Y and Z with Pin Numbers and Chip Labels

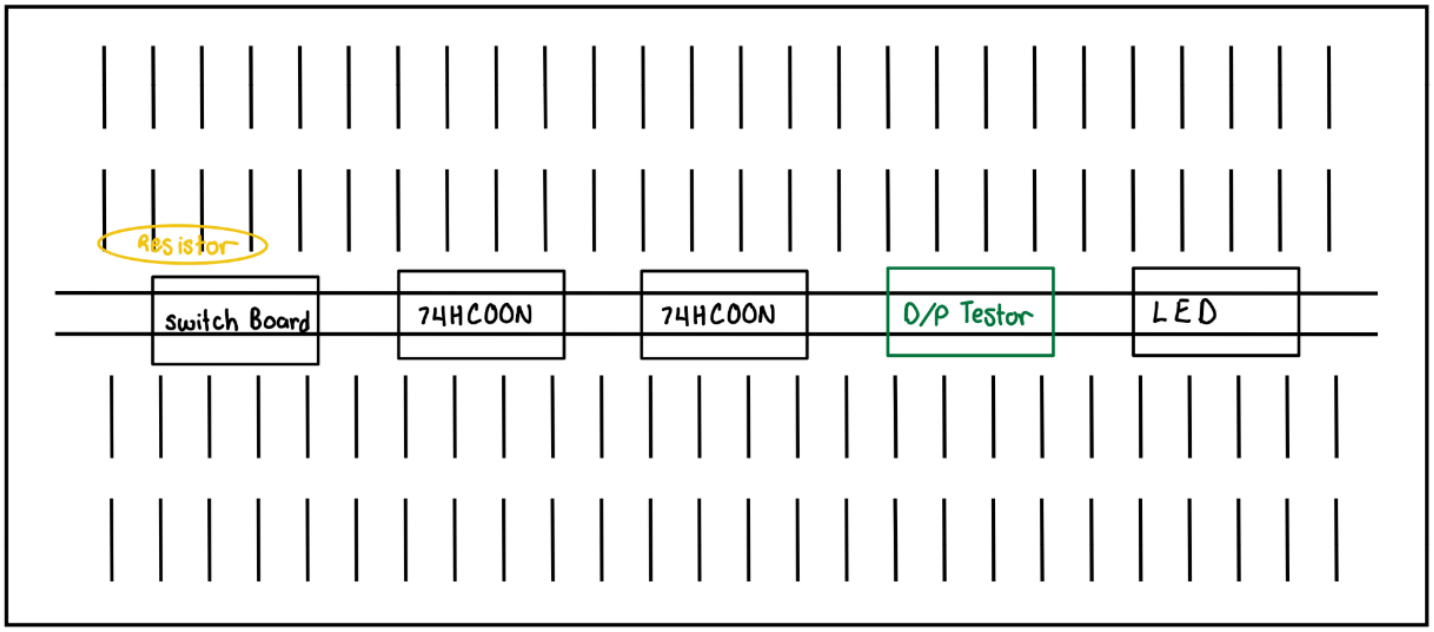


Table 24: Lab 1 Part 5.6 - Wiring Diagram for Y and Z

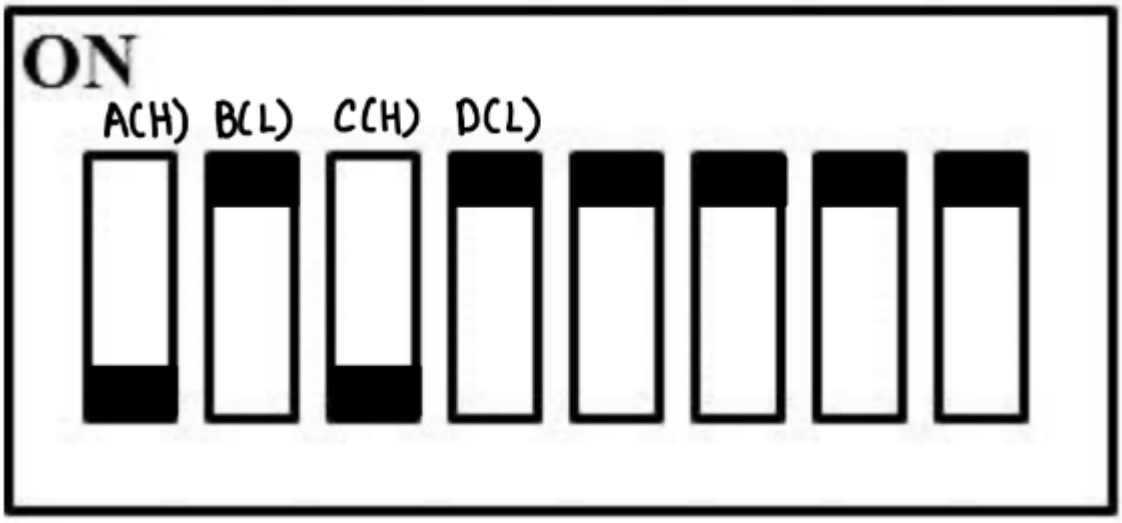


Table 25: Lab 1 Part 5.7 - Switch Legend

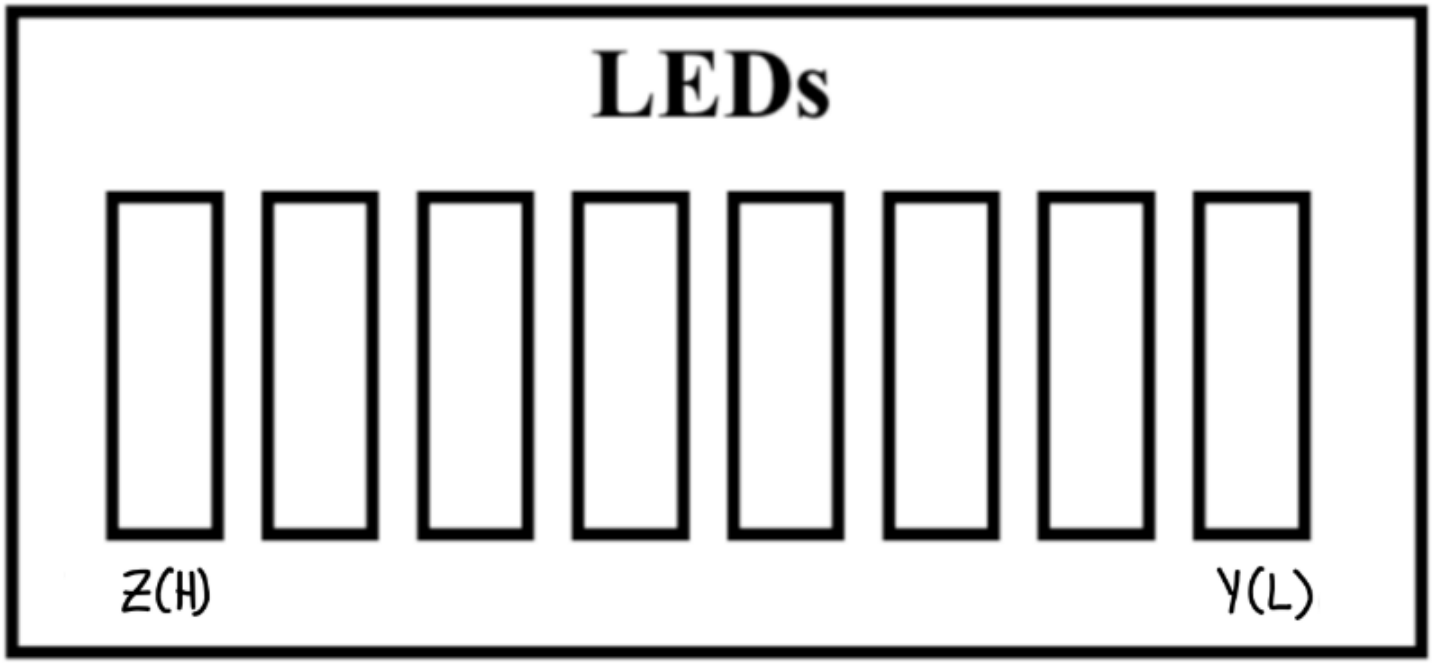


Table 26: Lab1 Part 5.8 - LED Legend