

Lab 1: Basic Logic Design

Overview

- Learn to use the Logisim Evolution tool to visualize logic design.
- Understand the operation of Quartus Prime as a digital design and simulation tool.
- Understand the difference between SOP and POS.
- Simplify logic gate networks to reduce hardware cost using Boolean algebra.

Introduction

Digital logic entails implementing the theory of Boolean algebra into a physical capability in the form of electricity. Knowing this, we can design and simplify circuits using some fundamental Boolean identities:

Name	AND form	OR form
Identity law	$1A = A$	$0 + A = A$
Null law	$0A = 0$	$1 + A = 1$
Idempotent law	$AA = A$	$A + A = A$
Inverse law	$A\bar{A} = 0$	$A + \bar{A} = 1$
Commutative law	$AB = BA$	$A + B = B + A$
Associative law	$(AB)C = A(BC)$	$(A + B) + C = A + (B + C)$
Distributive law	$A + BC = (A + B)(A + C)$	$A(B + C) = AB + AC$
Absorption law	$A(A + B) = A$	$A + AB = A$
De Morgan's law	$\overline{AB} = \bar{A} + \bar{B}$	$\overline{A + B} = \bar{A}\bar{B}$

There are plenty of resources that show how these identities can be derived, such as this [article](#). Boolean logic also has applications in other fields, such as [philosophy](#).

Pre-Lab Procedure

Note: Include pictures of any hand-drawn diagrams in your submission. Show all work.

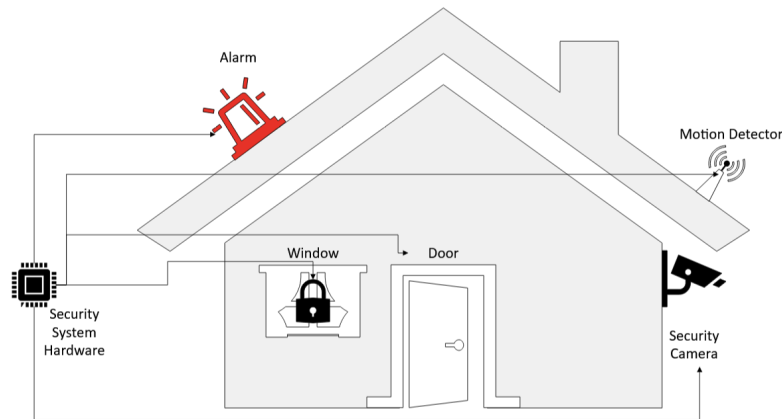
Part 1. Boolean Algebra Simplification: Take the following equations and obtain the Minimum Sum of Products (smallest number of terms/literals possible). Write out all required steps. **(2)**

a. $F = A \cdot B + A \cdot \bar{B} + \bar{A} \cdot B$

b. $Z = \bar{V}W + X + \bar{Y}(X + \bar{V}W)$

Part 2. Implementation: In your **part2** circuit (find it by opening Lab1.circ with Logisim and you will see “part2” in the Design tab under “Lab1”), implement the unsimplified functions from **1a/b** in Logisim. Additionally, you will implement the simplified versions. Name the unsimplified outputs as F and Z, and name the simplified outputs as F_S and Z_S. Include screenshots of these in your submission document. **(4)**

Part 3. Security System Design: You have been hired as an engineer for a home security company. They have asked you to design a *Smart Home Security System*:



The company has asked you to design it such that in one circuit:

- The **Alarm** triggers (1) such that given the motion detector sees something (1), and the door or window are unlocked (0). Remember the boolean algebra definition of ‘or’. Otherwise, it could be a false alarm. Assume the camera can see everything (one big camera network).
 - The **Security Camera** starts recording (1) if the motion detector sees something (1) or if the door and/or window unlock (0).
- a. Draw out the truth table, using the variables:
- D - Door closed (1 = locked, 0 = unlocked)
 - W - Window locked (1 = locked, 0 = unlocked)
 - M - Motion detector detects (1 = detects motion, 0 = does not detect motion)
 - S - Security camera (1 = recording, 0 = off/saving power)
 - A - Alarm (1 = active, 0 = inactive)

Table format, in counting order: **D W M | A S**

(8)

b. Write out the Boolean functions from **2a** using the truth table you created: **(2)**

[Watch this video to learn more about generating an equation from a truth table.](#)

- c. $A(D, W, M)$ using the Sum-Of-Products form.
 $S(D, W, M)$ using the Product-Of-Sums form.
- d. Implement the simplified functions from the previous part in **Logisim** in your **part3c** circuit. Include screenshots of these in your submission document. **(2)**

Part 4. Using Quartus tools to translate our designs into a real digital system

Quartus provides the tools to translate your design into something understandable for the FPGA (the MAX10 chip on your DE10 Lite), which will allow you to see your design work in real life.

See [Appendix A](#) for instructions on how to program the board using the Quartus Toolchain in Logisim.

For this part **you do not need to submit anything**. To get the points for this part you must demo the working design in person to your PI during your lab section. Please see the rubric.

You will model your design on your DE10-Lite by assigning your inputs and outputs of your security system to certain components on the board.

Assign your inputs and outputs as follows ([DE10-Lite Manual](#)):

Door	SW2
Window	SW1
Motion Sensor	SW0
Alarm	LED1
Security Camera	LED0

In-Lab Procedure

1. You will demo your circuit working in **part 4** to your PI. **(20)**
2. Complete the quiz in-lab as specified by your PI. **(40)**

Rubric

<u>Note</u>	<u>Points</u>
Pre-Lab	
P1 - equations fully simplified with steps included (0.5pt for correct answer, 0.5pt for steps)	2
P2 - each circuit has correct behavior (1pt each)	4
P3A - All rows of truth table are correct (1pt each)	8
P3B - Correct SOP and POS form (2pt each)	4
P3C - Functions match P3B (1pt each)	2
Lab report shows work and/or screenshots for all pre-lab parts.	20
In-Lab	
Part 4 demo	20
Quiz	40
Total	100

Appendix A

Programming DE10-Lite with Logisim

1. Before you do anything, ensure that all inputs/outputs are labeled.
2. Follow the instructions in the **Installing Logisim** section of the [Software Installation Guide](#) (integrating Logisim with Quartus/Modelsim) if you have not already.
3. Download [this XML file](#).
4. In Logisim, go to the **FPGA** tab on the top, and select **Synthesize & Download**.
5. Under the dropdown menu for “*Target Board*”, select “*Other*”
6. Find the downloaded XML file from step 2. Select it, and then use this as your target board. You only need to perform this step one time.
7. Map the inputs and outputs to pins, then select **Done**.
8. Ensure the DE10-Lite is plugged in, and select **Execute**. It will ask to confirm that your board is plugged in, and then it will program the DE10.