Simple Combination Logic Circuit

245 Lab 1

Chris Tark

9/13

**Objective**:

A simple combination logic circuit was designed and translated from an initial function (F) into a physical working circuit. By building this combination logic by hand and through Quartus II I will learn the fundamentals of logic diagram and wiring logic.

Diagram, schematic

Description automatically generated**Design**:

Figure 1: Displays a logic diagram of the function F

Diagram

Description automatically generated

Figure 2: Part of prelab 1 is to design a minimum SOP with inputs 1, 2, 3, 5. This figure demonstrates an K-map and the output A’B + B’C.

A piece of paper with writing on it

Description automatically generatedFigure 3: This is my prelab design. Top right is the initial logic diagram of function F. The middle and bottom pages a wiring diagram. Both diagrams helped created the final product on the breadboard.

The design requirements are simple. I first had to familiarize myself with a breadboard and based off my knowledge be capable of constructing a circuit. Second, derive the minimum sum of product using K-Map simplification, work is shown in figure 2. Lastly, I needed to implement a working circuit with Quartus II software which is presented in figure 1.

**Hardware**:

* BK Precision Triple output DC Supply
* Jumper Wire Kit
* Breadboard
* One pair of alligator clips (one positive and one negative)
* Resistors
* Two 7400 Quad 2-input NAND
* One DIP switch

A piece of paper with writing on it

Description automatically generated**Experiment**:

Figure 4: Displays my wiring diagram to ultimately create the function A’B + CB’. This is also shown in the design section.

This circuits input comes from a DIP switch. A device that allows certain combinations as the input, four switches on the face converts to a binary number. This input indicates all our valid inputs from the original K-Map. To test my output, I set my DIP switch to input all the combinations (1, 2, 3, 4, 5, 6, 7). The LED only had signal when switched to 1, 2, 3, or 5.

Figure 5 (below) clearly indicates a complete wired circuit that works. To set up the circuit I first manipulated wires to create an input A, B, and C originating from the DIP switch. The DIP is connected to ground via low value resistors and wired with an open source (power supply connects). After applying three inputs, I connected two 7400 Quad 2-input NAND gates to the “divot” of the breadboard and followed my wiring diagram draft (figure 4). In the left 7400 NAND gate output 3 is A’, A' relates to B, equivalent first element in function F. B’ is made in a NAND gate in unit 10, again on the left which is finally combined with C in unit 11. Only one gate is used on the right and outputs final the sum of products A’B + CB’.

A picture containing diagram

Description automatically generated

Figure 5: Displays the physical circuit and outputs a valid number 5 causing the light to turn on.

**Analysis**:

The circuit performance is expected to turn on the LED when inputs 1, 2, 3, and 5 are placed on the DIP switch. The circuit shown in Figures 5 and 6 does exactly what is expected 0, 4, 6, and 7 all do not provide a source for the LED to turn on. While constructing the circuit, I was questioning how to incorporate an or gate as it is required in the sum of products equation. I was told by a nearby classmate to not consider that because 7400 takes that into account.

A picture containing diagram

Description automatically generated

Figure 6: Displays a version of output causing no valid response from the LED light**.** The input is 0.

**Conclusion**:

Building a circuit that only works with certain inputs is surprisingly simple. The purpose of this lab was to get comfortable with how a breadboard can be utilized for fascinating outputs of creators choosing. This simple circuit helps visualized what K-Maps can abstractly present and the 7400 NAND gate’s power in a materialized circuit. I learned what a breadboard is and its specific internals underneath the cover. The outer notches are connected horizontally, and the rest are connected vertically. Finally, I learned to create a working circuit ever (this is my first one).

**Appendix**:

n/a