ECE296 PLC Lab 0 - Logic Gates

Chase A. Lotito, *SIUC Undergraduate*

I. Introduction

This experiment was an introduction to the programmable logic controller (PLC). The particular PLC being the Allen-Bradley MicroLogix 1100, which we can program with ladder logic in *RSLogix*.

Using the toggle switches as inputs (SW1 and SW2), the goal is to recreate the functionality of 2-bit AND, OR, and XOR logic gates. They should follow these equations if $A = SW1$ and $B = SW2$:

$$\text{AND} : A \cdot B \tag{1}$$

$$\text{OR} : A + B \tag{2}$$

$$\text{XOR} : A \cdot \bar{B} + \bar{A} \cdot B \tag{3}$$

The blue light represents the output for AND, the green light for OR, and the yellow light for XOR.

II. Assessment of Design

The specific ladder logic used can be seen in *Appendix A*.

In order to check if a switch is "on" or "off", we need to first designate the I/O as normally open (NO) or normally closed (NC). For NO, if "off" we get 0, and if "on" we get 1. For NC, if "off" we get 1, and if "on" we get 0. Essentially, NO preserves the state of the switch, and NC inverts the state of the switch.

A series combination of NO and or NC contacts requires both to be 1 for continuity, so that acts as an AND operation. A parallel combination of NO and or NC contacts only requires a single contact to be 1 for continuity, so that acts as an OR operation.

Figure 5 shows these principles applied to Eq. 1.

Overall, the design works as expected.

Figures 1-4 below show the 4 possible states that the switches can be in (00, 01, 10, 11). The lights from left to right are blue (AND), green (OR), and yellow (XOR).

Fig. 1. State 0 (SW1 = 0, SW2 = 0)

Fig. 2. State 1 (SW1 = 1, SW2 = 0)

III. Conclusion

This lab effectively gives a brief introduction using PLCs and how to use ladder logic with NO

Fig. 3. State 2 (SW1 = 0, SW2 = 1)

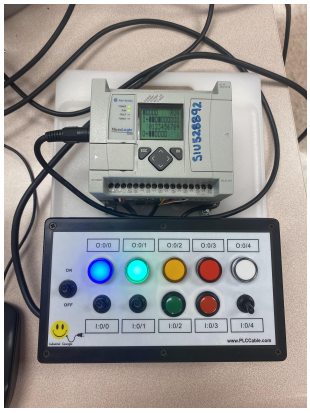

Fig. 4. State 3 (SW1 = 1, SW2 = 1)

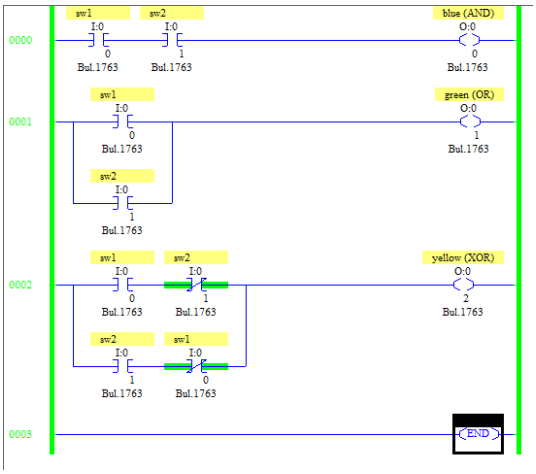

Fig. 5. Ladder Logic for Logic Gates

and NC contacts in order to do logic operations. These skills can now be used further to implement more complex systems, which fundamentally work on bitwise logic operations.

Appendix A: Ladder Logic