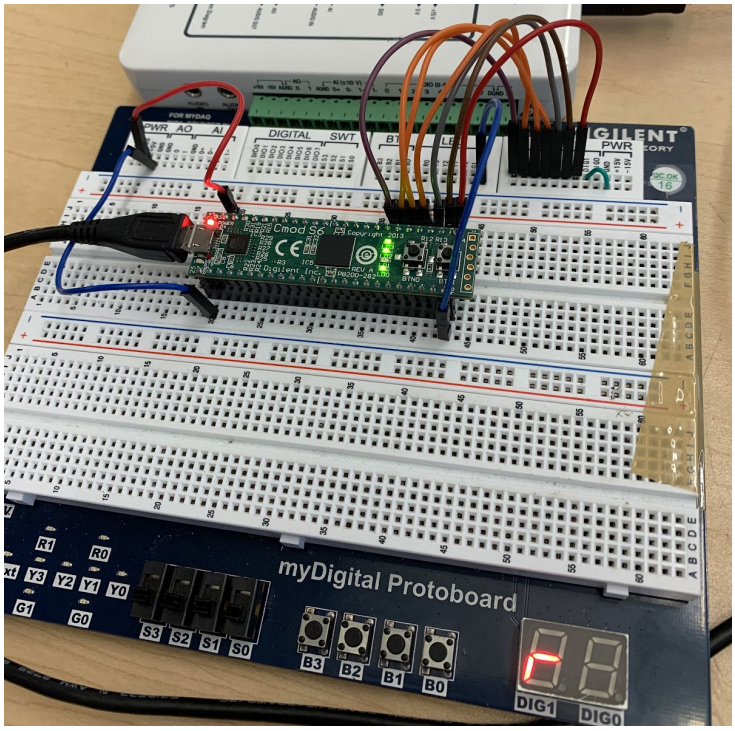



# PLD Design

## Project 2.3.5

Justin Thai

Ms. Chou Period 5

Digital Electronics 2019-'20

Dublin High School

February 25–March 2, 2020

# Table of Contents

# Design Brief & Display Definitions

| | |
|---|---|
| Intended Audience: | Signboard Manufacturers |
| Designer: | Justin Thai |
| Problem Statement: | Some collectors want to protect their more valuable belongings, so they need a way to hide them. |
| Design Statement: | By creating an electronic message with a seven segment display and a programmable logic device, you can make a warning for thieves that the hidden valuables should not be stolen. |
| Constraints: | i. The seven segment display should have four bits for a total of sixteen states.<br>ii. Of these states, no more than five can be blank.<br>iii. Optionally, five bits can be used for a total of 32 states. |
| Deliverables | i. Title Page - name, date, period, project name<br>ii. Table of Contents - corresponding sections and page numbers<br>iii. Design Brief & Display Definitions - design statement, constraints, summary of display (inputs and outputs)<br>iv. Truth Table & Logic Expression - table and algebraic expressions<br>v. Solution - Simulation & Breadboard - images of simulation and breadboard shown clearly, multiple views as necessary, labels and descriptions of final design |

**Display Definitions:**
Internally, the PLD uses a clock voltage with flip flops to create the bit counter used as the inputs into the logic indicated in the truth table. The outputs from the PLD are connected to a common cathode 7-segment display, and send signals such that the SSD displays the message “SECRET TO EVERYONE” as the input from the bit counter iterates sequentially. These outputs are from pins 1 through 7 of the PLD, and correspond to the segments of the 7-segment display. Because there is no external input, there are no pins designated for it.

# Truth Table & Logic Expressions

| Number | Inputs | | | | | 7-Segment Display Segments | | | | | | | SSD Display |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | V | W | X | Y | Z | a | b | c | d | e | f | g | |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 |  |
| 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 |  |
| 2 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 |  |
| 3 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |  |
| 4 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 |  |
| 5 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |  |
| 6 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |  |
| 7 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |  |
| 8 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 |  |
| 9 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |  |
| 10 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 |  |
| 11 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 |  |
| 12 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 |  |
| 13 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |  |
| 14 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 |  |
| 15 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 |  |

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 16 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 |  |
| 17 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 |  |

**Logic Expressions Unsimplified:**

a = W’X’Y’Z’+W’X’Y’Z+W’X’YZ’+W’XY’Z’+WX’YZ’+WXY’Z’+VW’X’Y’Z’

b = V’WXYZ’

c = V’W’X’Y’Z’+V’WX’Y’Z’+V’WX’YZ+V’WXYZ’+V’WXYZ+VW’X’Y’Z’

d = V’W’X’Y’Z’+V’W’X’Y’Z+V’W’X’YZ’+V’W’XY’Z’+V’W’XY’Z+V’W’XYZ+V’WX’Y’Z’+
V’WX’YZ’+V’WX’YZ+V’WXY’Z’+V’WXYZ’+V’WXYZ+VW’X’Y’Z’

e = V’W’X’Y’Z+V’W’X’YZ’+V’W’X’YZ+V’W’XY’Z’+V’W’XY’Z+V’W’XYZ+V’WX’Y’Z’+V’WX’YZ’+
V’WX’YZ+V’WXY’Z’++V’WXY’Z+V’WXYZ+VW’X’Y’Z’+VW’X’Y’Z

f = V’W’X’Y’Z’+V’W’X’Y’Z+V’W’X’YZ’+V’W’XY’Z’+V’W’XY’Z+V’W’XYZ+V’WX’YZ’+
V’WXY’Z’+V’WXYZ’+VW’X’Y’Z’

g = V’W’X’Y’Z’+V’W’X’Y’Z+V’W’X’YZ+V’W’XY’Z’+V’W’XY’Z+V’W’XYZ+V’WX’Y’Z’+
V’WX’YZ’+V’WXY’Z’+V’WXY’Z+V’WXYZ’+V’WXYZ+VW’X’Y’Z’+VW’X’Y’Z

**Logic Expressions Simplified:**

a = V’W’X’Y’+X’YZ’+XY’Z’+VZ

b = WXYZ’

c = WXY+WYZ+X’Y’Z’

d = V’X’Z’+V’W’Y’+W’XZ+WZ’+WY+VZ

e = W’Z+WX’Z’+X’Z+XZ’+YZ+V

f = V’W’Y’+W’XZ+X’YZ’+WXZ’+VZ

g = WZ’+XZ+W’Y’+W’Z

# Final Solution

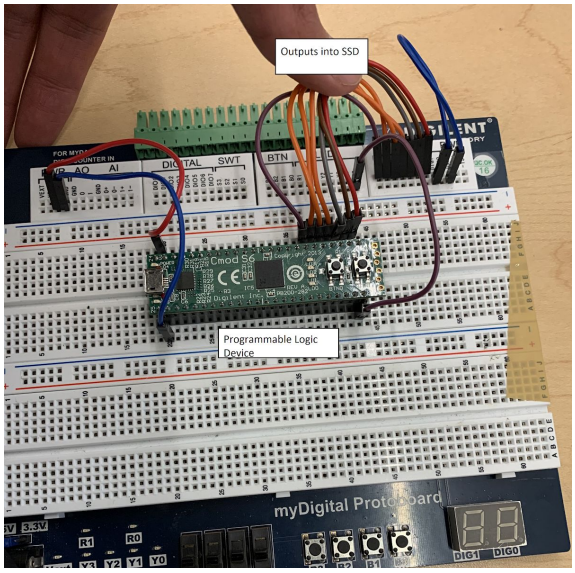
**Final Solution Description:**

In order to make the SSD display the phrase “secret to everyone,” logic needed to be created for the 18 different characters. I used five bits as inputs for my bit counter, since only four bits wouldn’t iterate high enough; However, the increase to 32 possible characters leaves me with a significant amount of unused states. To resolve this, I created a logic circuit that would return the bit counter back to zero prematurely to truncate the unused states. When the bit counter reaches a certain number, the logic outputs high into the reset of the flip flops that make up the bit counter, returning the bit counter to zero.

To make the SSD form the characters I want, the individual segments need their own logic circuits to indicate when they should light up during the cycle of the bit counter. For each of them I created a truth table, with the five bits of the bit counter as inputs and the segment as the output, and inserted the table into a logic converter. When making the table I made sure that the states I truncated were designated as “don’t care” conditions. I then used the logic converter to create simplified logic expressions from the truth tables, which were then converted into circuits, which I connected to the bit counter inside the multisim file.

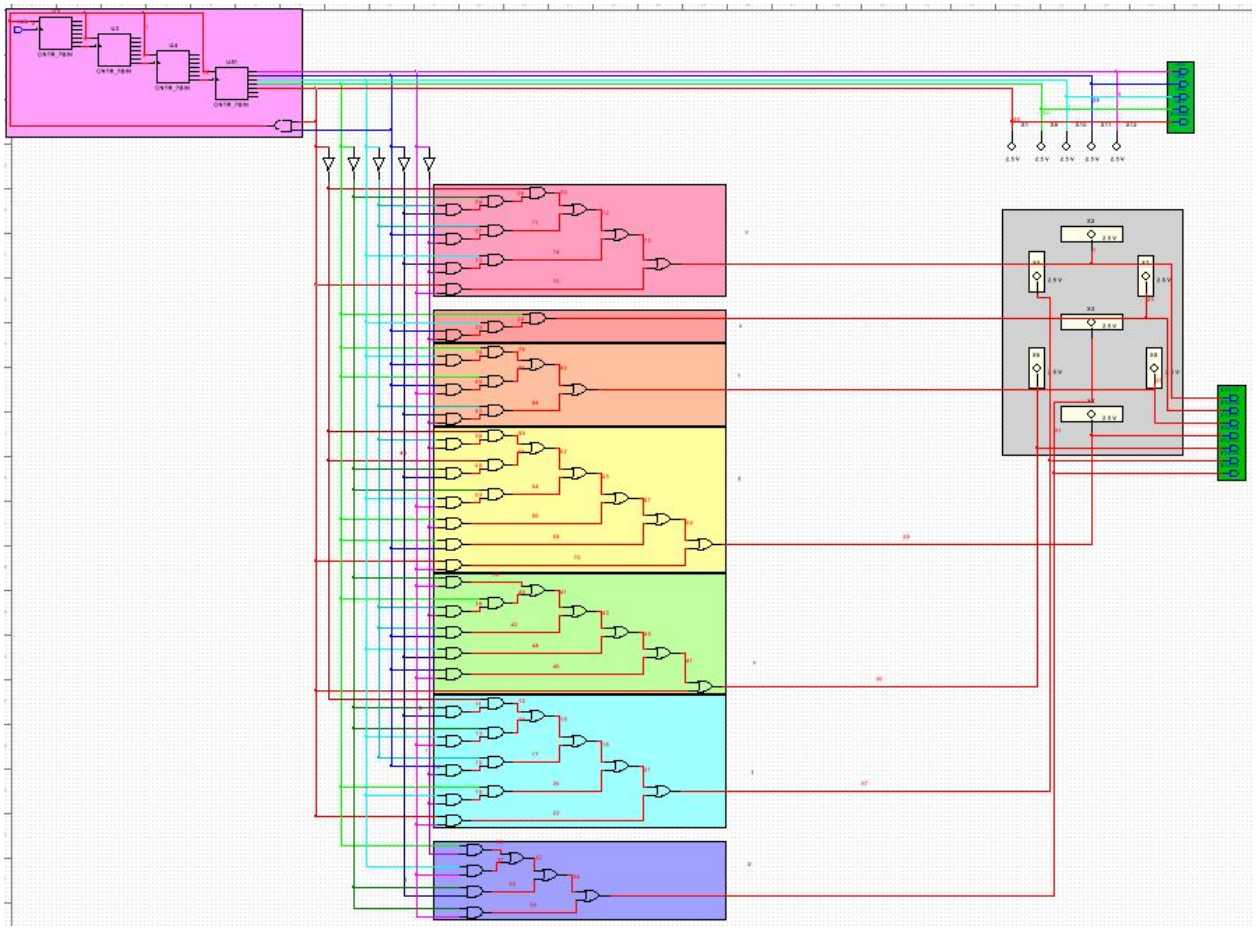
The completed circuit requires many logic gates. Making this electronic message using ICs would be very time consuming and creates the potential for errors during the production process. Instead, by using a programmable logic device, the process of physically creating the logic circuit can be overcome. By setting up the PLD file such that its clock voltage is the input and its pins output the logic from the circuits, the entire circuit is complete.

**Final Solution Breadboard:**




Above is a picture of the physical breadboard of my final design. It is clear to see the lack of complexity in its physical components, as all the logic required to create the proper outputs is done internally using the PLD’s transistors.

**Final Solution Simulation:**



| Color | Label | Description |
|---|---|---|
| | Segment a | These are the logic circuits used to make the corresponding segments output high properly, to create the letters appear on the 7-segment display. The inputs for these come from the bit counter, and their outputs are sent from the PLD to the 7-segment display externally. |
| | Segment b | |
| | Segment c | |
| | Segment d | |
| | Segment e | |
| | Segment f | |
| | Segment g | |
| | Bit Counter | To create a bit counter that iterates automatically, the clock voltage built into the PLD was used. However, this CLK is locked at a frequency of 8MHz. To slow this frequency so that it is visible to the human eye, it is wired through multiple divide-by-two counters. In the |

| | | |
|---|---|---|
| | | simulation a special component was used that is the equivalent of 7 flip flops chained together. The last few flip flops were used to output into the logic circuits for the 7 segments. The slowest of the flip flops chosen was designated as the most significant bit, and the fastest was designated as the least significant bit.<br><br>With 5 bits used as inputs, there are a total of 32 possible combinations. Of these only 18 were used to actually display characters or intentional blanks in the phrase, “secret to everyone”. In order to resolve the issue of unused numbers, logic was used to truncate the range of values. When the bit counter reached 18, it would return to 0 instead of continuing to iterate up to 31. |
| | PLD Outputs | These output connectors connect the output of the logic circuits to the external 7-segment display. The pins 1-7 of the PLD correspond to inputs g, f, e, d, c, b, and a, respectively.<br>Additionally there are five other output connectors for the LEDs built into the PLD and another pin so that the number of the bit counter can be seen. |

Justin Thai
3/2/2020