Kurt Aquino Angelo Amadora

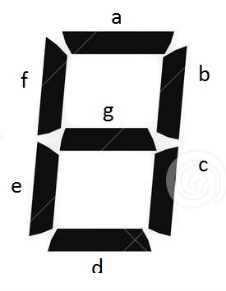
Seaver Choy Luigi Ramos

**Traffic Light with Counter**

As the name suggests, the purpose of this project is to imitate the functionality of a common traffic light into a smaller scale using circuits and breadboards. As with every other stop light, a counting mechanism must be implemented into the machine to sequence the lighting of the LEDs (red, amber, green).

Our system will make use of JK Flip Flops, a BCD to & segment decoder as well as multiple and/or gates for the binary manipulation. A clock circuit as the sequence generator, a 7 segment LED, and a red, amber, and green LED for the outputs.

Each pulse from the clock will only trigger specific segments of the counter to represent the corresponding digit. Given that our system will only have a fixed count, since we are using 4 bits of input, a total of 16 states will be produced, we will integrate the count for each LED light and the LED counter together. As for the output for each sequence there will be a countdown (9-4) for the green LED (0-5th state), another countdown (4-0) for the yellow LED (6th-10th state), and finally one (5-0) for the red LED (11th-15th state). The triggering of LEDs will be represented by the table below:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| I3 | I2 | I1 | I0 | green | yellow | red | a | b | c | d | e | f | g |
| 0 | **0** | **0** | **0** | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |
| 0 | **0** | **0** | **1** | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0 | **0** | **1** | **0** | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 0 | **0** | **1** | **1** | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 |
| 0 | **1** | **0** | **0** | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 |
| 0 | **1** | **0** | **1** | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 0 | **1** | **1** | **0** | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |
| 0 | **1** | **1** | **1** | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 |
| 1 | **0** | **0** | **0** | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | **0** | **0** | **1** | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 1 | **0** | **1** | **0** | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 |
| 1 | **0** | **1** | **1** | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 1 | **1** | **0** | **0** | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |
| 1 | **1** | **0** | **1** | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 |
| 1 | **1** | **1** | **0** | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | **1** | **1** | **1** | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |

The state diagram is as follows:

0000 -> 0001 -> 0010 -> 0011 -> 0100 -> 0101 -> 0110 -> 0111 -> 1000 -> 1001 -> 1010 -> 1011-> 1100 -> 1101 -> 1110

1111