EE-281 Logic Design Lab

Lab #2

Other Logical Building Blocks and Design Process

## Name: Ajin Sunny and Derek King

## Date of Experiment: January 26th 2015

## Date of Completion of Report: January 31st, 2015

**Introduction**

In this lab we will be designing a redundant fault detector circuit for an airplane that monitors serious faults which the aircraft encounters such as engine failure, fire inside the aircraft, cabin pressure variation, and bird impact on the aircraft. We will use a 7 segment display to indicate the pilot about the status of the sensors which detects the fault in the aircraft. We will also use a priority encoder to ensure that the most important error is displayed regardless of any other errors that might be present.

**Experiment and Description**

We will design a logic circuit by using TTL arts from output from the sensors to a two bit code, C1 C0. The circuit will output an overall status signal of ERR\_L as logic level 1 when there is nothing is wrong and 0 when something is wrong.

Priority encoders in the circuit will output the signal based on the most important error and the priority of the errors will as follows:

1. Engine Failure
2. Fire
3. Cabin Pressure
4. Bird Impact

First we will draw the truth table of the inputs and then the expanded truth table including which segments of the display should be turned on based on the input (negative logic).

Truth Table:

|  |  |  |  |
| --- | --- | --- | --- |
| ERR\_L | C0 | C1 | F(ERR\_L, C0, C1) |
| 0 | 0 | 0 | E |
| 0 | 0 | 1 | F |
| 0 | 1 | 0 | C |
| 0 | 1 | 1 | B |
| 1 | d | d | None |

Expanded Truth Table:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ERR\_L | C0 | C1 | A | B | C | D | E | F | G |
| 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 1 | d | d | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

Looking at these truth tables, we can notice a few patterns. First of all, segments A and C are opposites, with A being an AND relation with C1 and C0 and C being a NAND relation between C1 and C0. In addition, E and F are on if there is an error, because of this we will not need a MUX in order to control the logic behind those segments. Also, segment B is not used; therefore there is no need to wire it in our design. The following equations that we determined to be the logic we needed to implement.

**Output Display**

The output display works with negative logic. The segments are given 5V and the outputs are controlled be giving a particular segment 0V (5V drop leads to segment on, negative logic) or 5V (0V drop leads to segment off). Therefore, we must design logic with our C0 and C1 inputs in order to output a digital 0, or ground, in order to turn on the LED’s we want. The following are the digital logic gates that we designed for each input in order to get the output results we wanted. Please note that the logic going into the MUX corresponds with ERR\_L being 0 as indicated by the truth table. If ERR\_L happened to be a 1, the input to the MUX would just be a 1. Also note that B is excluded because it isn’t used (that input to the 7-segment display is just tied to 5V rather than a MUX), and E and F are excluded because the input to the MUX would just be ground.

Fig. 1 (A)

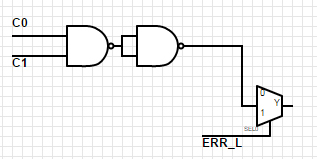


Fig. 2 (C)

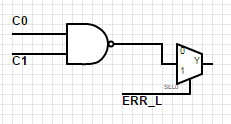


Fig. 3 (D)

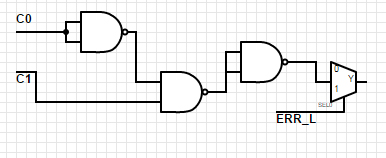
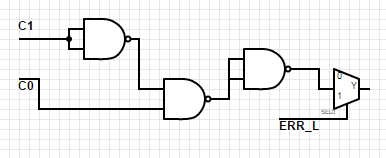


Fig. 4 (G)



**Results**

With the combination of the MUX and the priority encoder, we were able to successfully implement a fault detector using a single 7-segment display. The priority encoder determines which faults are more important that others and sends that information to the MUX which then outputs to the display the most critical error at any given time. Besides a few minor wiring mishaps, our experiment worked flawlessly the first time we wired it thanks to the time we spend designing the inputs to the MUX as well as the attention to wiring all of the chips together correctly.

**Conclusions**

In conclusion, this experiment taught us how to use the priority encoder as well as the MUX. With these two circuit elements, we were able to build a practical fault detector. We gained even more experience with reading data sheets as well as diagnosing faults when we had a few wires misplaced. This lab will help us in future experiments if we have to use priority encoders or MUXs. In addition this lab prepared us to better diagnose a circuit that is not working properly.