EEE102-02 Lab 3 Report:

Combinational Logic Circuit

Kaan Ermertcan - 22202823

09.10.2023

Purpose:

In this lab, we will learn how to build a combinational logic circuit on a breadboard using logic gate ICs. We are going to use a counter to test our circuit with all possible inputs and use an oscilloscope and LEDs to see the output of our circuit and troubleshoot if it is not working.

Design Specifications:

My design is supposed to achieve the same outputs as in my design in lab 2 but on a breadboard instead of on a BASYS 3. There are 3 inputs, each representing a natural condition and two outputs, each representing a status of different part of a house. The truth table my design should achieve is below.

Conditions			**Outcomes**	
Nighttime (A)	Winter (B)	Windy (C)	Window (F)	Heating (G)
0	0	0	1	0
0	0	1	1	0
0	1	0	0	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	1
1	1	1	0	1

Table 1.1: Truth table.

Methodology:

I noticed that I could have made my design in lab 2 simpler. Since then, I used the Karnaugh Map method that we learned this week in the lectures to write simpler expressions for my outputs. First, I converted my truth table (Table 1.2 in Lab 2 Report) to two K-Maps, one for each output:

BC A	Window(F)			
	1	1	0	0
	1	0	0	0

Table 2.1: K-Map of output F.

BC A	Heating(G)			
	0	0	1	0
	0	0	1	1

Table 2.2: K-Map of output G.

From the K-Maps, I got following expressions:

$$F \equiv B'C' + A'B' \ \equiv B'(A' + C') \equiv (B + AC)'$$

$$G \equiv BC + AB \equiv B(A + C)$$

Then, I created an RTL schematic of my improved design:

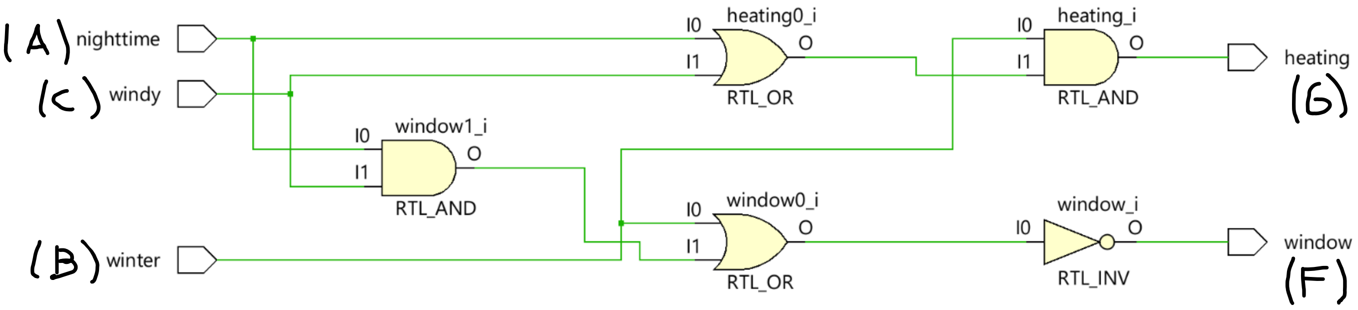


Figure 2.1: RTL Schematic.

The number of logic gates that is required is less compared to my previous design. Thus, this design is easier to implement on a breadboard.

In implementing my circuit, I used 2 gates on a 74HC08 quad 2-input AND gate IC, 2 gates on a 74HC32 quad 2-input OR gate IC, and 1 inverter on a 74HC04 hex inverter IC. Besides the logic gates, I used 3 of the 4 outputs of a 74HC163 counter to test my 3-input logic circuit. I also needed two LEDs for the outputs, but I also decided to add 4 more LEDs to the clock input and outputs of the counter to better visualize what is going on. I also used 1K resistors in series with each LED. Finally, as an extra, I added a reset button for my counter and connected it with a 10K pull-up resistor.

After finding all necessary components, I studied all 4 ICs' datasheets to find out how it should be used in a circuit. Then I placed them on a breadboard with the connection order in mind. (Counter->AND & OR gates->Inverter->Output) I connected their VCC and GND connections to corresponding rails on the breadboard. According to the Function Table on the datasheet of the counter, I connected other pins (CEP, CET, PE) to 5V rail for it to run on count mode. I connected the Reset (MR) pin via a 10K pull-up resistor to 5V rail and via a push button to ground. I set the power supply to 5V and connected it to the power rails of the breadboard. Finally, a clock signal was needed for it to count. I set the signal generator to output a square wave at 10Hz with a 2.5 V offset to make it DC. I made sure that the counter is working by monitoring its outputs (Q0-3) on an oscilloscope. When the clock pulse is 1Hz, the counter should be counting one up per second. I will use outputs Q0, Q1, Q2 as I only need 3 inputs. (Counting 0 to 7) After disconnecting power supply and signal generator, I can make the connections for the logic gates. I used the following mapping of the counter's outputs to inputs of my design, achieving the same order as of rows in my truth table.

Q0	Windy (C)
Q1	Winter (B)
Q2	Nighttime (A)

Table 2.3: Counter mapping.

I made logic gate connections while referring to my RTL schematic. Finally, I need to connect my LEDs. Instead of what is described in LEDcircuit.jpg file, I wanted my LEDs to light up when the signal is HIGH. So, I decided to connect my LEDs as shown below.

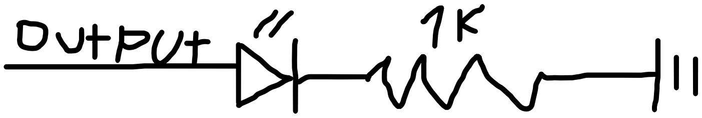
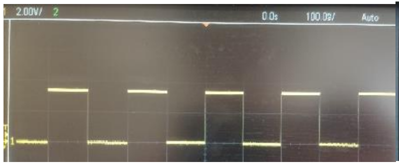


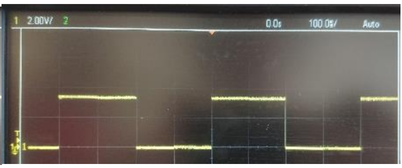
Figure 2.2: LED connection.

Now, my circuit is completed.

Results:

First, pictures of outputs of the counter with a 10Hz clock input displayed on the oscilloscope screen is below.





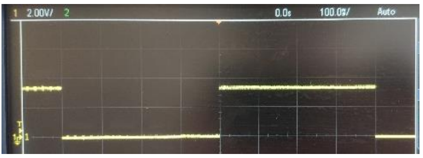


Figure 3.1: Output Q0 Figure 3.2: Output Q1 Figure 3.3: Output Q2

As each output waveform is doubled in length of one on the left, the counter works as expected.

My complete circuit on a breadboard is in the figure below.

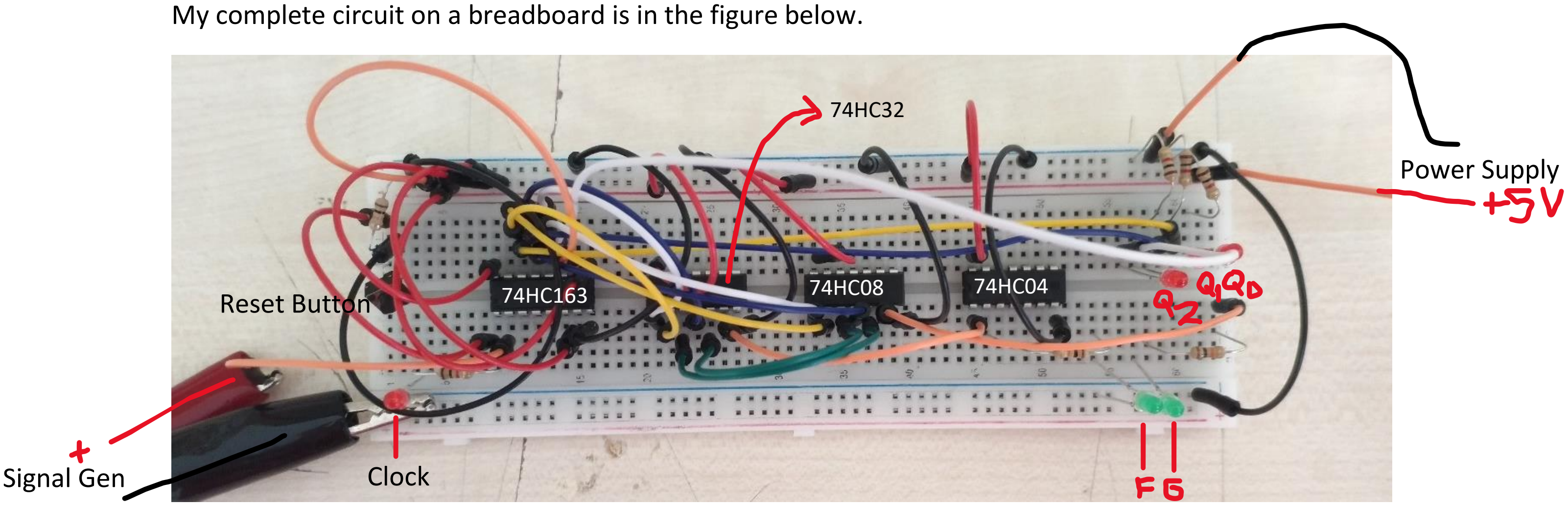


Figure 3.4: My circuit

As it can be seen from the figure, I used (mostly) red jumpers for +5V connections and black jumpers for ground connections. I used white for Q0, blue for Q1, yellow for Q2, green for connections between AND and OR gates, and orange for remaining connections. I connected my inputs Q0 and Q2 to the inputs of the first gates on 74HC32 (OR) and 74HC08 (AND). Then, I connected the output of the first AND to second OR, the output of the first OR to second AND. I connected the input Q1 to the remaining inputs of the second gates on both 74HC32 and 74HC08. Finally, the output of the second gate on 74HC32 is connected to the input of the first inverter on 74HC04 and the output of that inverter is our output G. The output of the second gate on 74HC08 is our output F.

Now, I set the signal generator to 1 Hz and connected it, and the power supply to the breadboard. At first the outputs seemed to be incorrect, but I quickly noticed that I mixed up two input signals and quickly swapped them. After that, my outputs exactly matched my truth table in Table 1.1 as can be seen from the figures below.

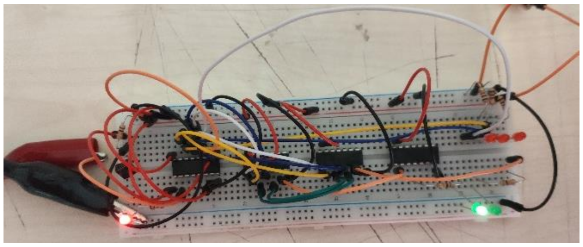


Figure 3.5: Case 000

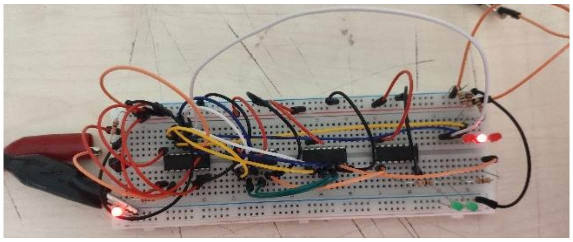


Figure 3.7: Case 010

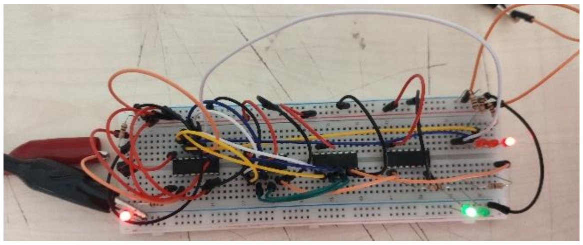


Figure 3.6: Case 001

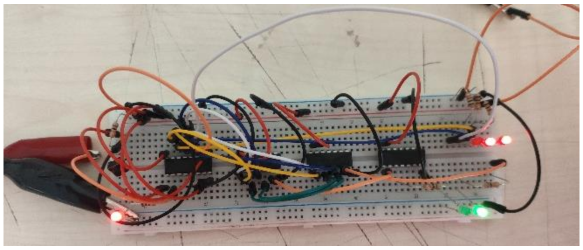


Figure 3.8: Case 011

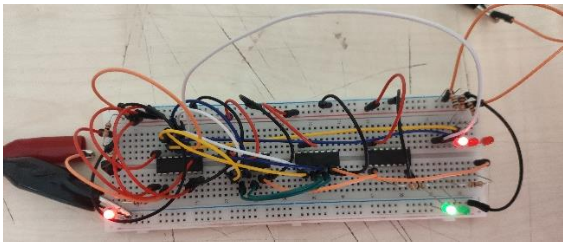


Figure 3.9: Case 100

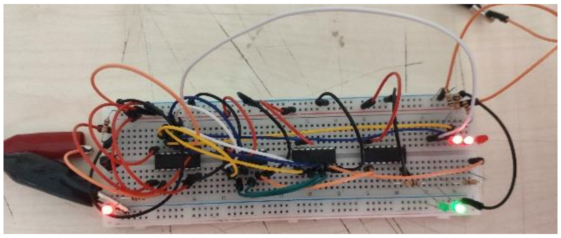


Figure 3.11: Case 110

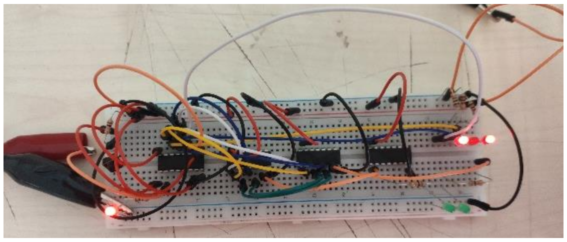


Figure 3.10: Case 101

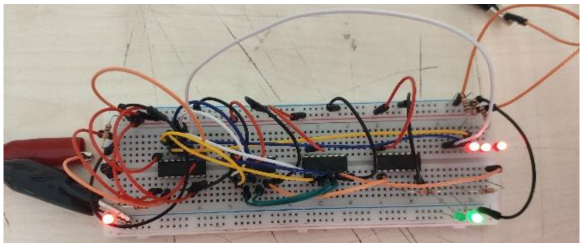


Figure 3.12: Case 111

After that, I also investigated the outputs under an oscilloscope with two probes, first comparing each output (Channel 1) with the clock signal (Channel 2) and then comparing outputs with each other (F on channel 1, G on channel 2).

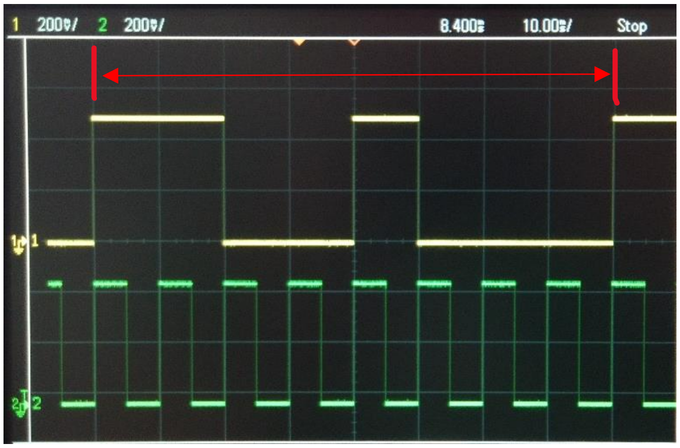


Figure 3.13: Output F on oscilloscope

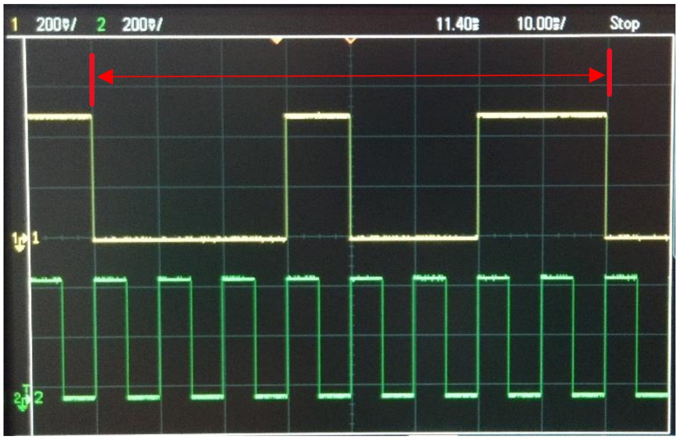


Figure 3.14: Output G on oscilloscope

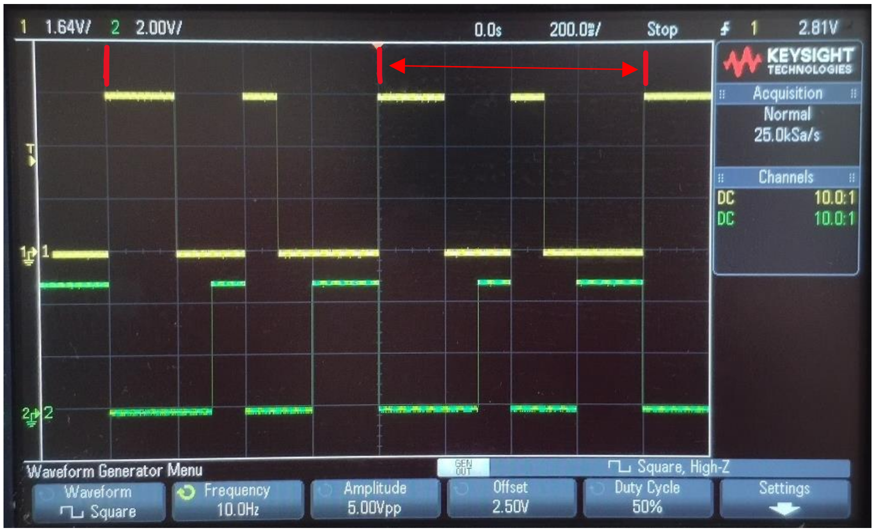


Figure 3.15: F on channel 1, G on channel 2

When taking oscilloscope measurements, I had to press Run/Stop as the oscilloscope was triggering each time when the output rises thus, triggering at two different points of the waveform. As can be seen from the figures, oscilloscope measurements also represent the truth table.

Finally, out of curiosity, I tried increasing the clock signal up to 10 MHz and inspected my output F and clock signal on the oscilloscope again.

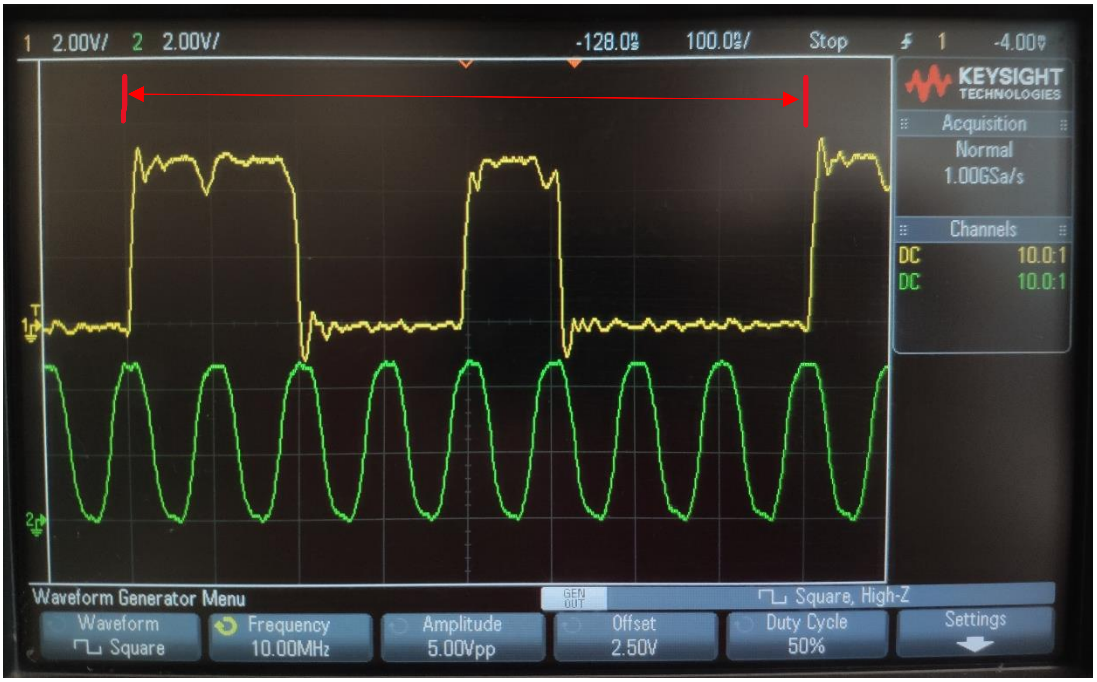


Figure 3.16: F on channel 1

While the voltage spikes are more significant at 10 MHz, my logic gates still produced the intended output. According to 74HC163's datasheet, clock frequency can be as high as 51 MHz. Thus, I have not exceeded the IC's operating limits.

Conclusion:

In this lab, I learned to build a combinational logic circuit on a breadboard and test it using oscilloscope, counter and LEDs. I also tried the circuit at different clock frequencies and verified that it works. I was lucky that the first chips I picked up worked and I did not need to swap chips. Learning the K-Map method in the lectures enabled me to simplify my design. As my results match my truth table, the experiment was successful.