CPE 324-01: Advanced Logic Design Laboratory

Lab01

Solderless Breadboard Prototyping and Verification of Small Scale Digital Logic

Submitted by: Esther Shore

Date of Experiment(s): January 11, 2024 and January 23, 2024

Report Deadline: January 25, 2024

1. Introduction

1.1 Assignment #1: Multi-IC Combinational Logic Design

This laboratory introduces the Terasic DE2-115 FPGA board to use as a platform to verify prototype simple hardware designs along with a solderless breadboard. The purpose of this experiment is to understand how to harness the input and outputs present on the DE2-115 board as well as observe a specific output’s behavior based on the construction of combinational logic gates using small scale integration type integrated circuits (SSI type ICs). This is accomplished by implementing a logic network on the solderless breadboard connected to the DE2-115 Trainer using a quad 2-input AND gate and a quad 2-input OR gate.

1.2 Assignment #2: Multi-IC Sequential Logic Design

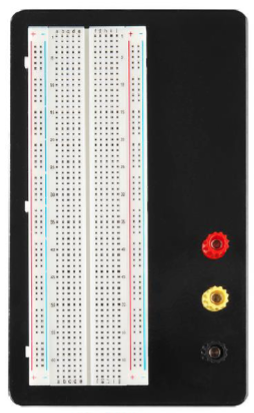
The sequential logic design assignment introduces the idea of cascading ICs to create an 8-bit binary counter from two 4-bit binary counters. It also highlights the basic differences between an asynchronous clear counter and a synchronous one, which is an important concept to understand in order to utilize hardware correctly.

1. Experiment Description

2.1 Assignment #1: Multi-IC Combinational Logic Design

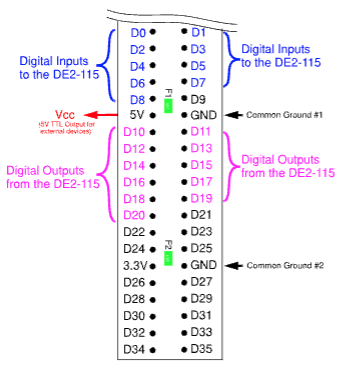
In order to set up for this experiment, we needed to understand how a solderless breadboard works and how to connect to the common ground and VCC to wire external components. It was necessary to understand the electrical connections of the breadboard, using the power rails and interconnected rows surrounding the center channel. The ICs were set up to straddle the center so that the pins on opposing sides would not be electrically connected. See Figure 1 below for a reference photo.

**Figure 1.** Solderless Breadboard

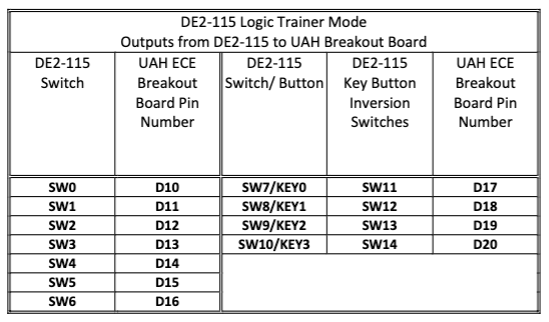


It was also crucial that we understand the arrangement of pins on the breakout board and how they corresponded to the switches on the DE2-115 as seen in Figures 2 and 3.

**Figure 2.** UAH ECE Breakout Board Pinout



**Figure 3.** DE2-115 Logic Trainer Mode Outputs

****

Furthermore, we learned to equate the breakout board pins to the corresponding

LEDs and LED segments for the seven-segment display referring to Figure 4 below.

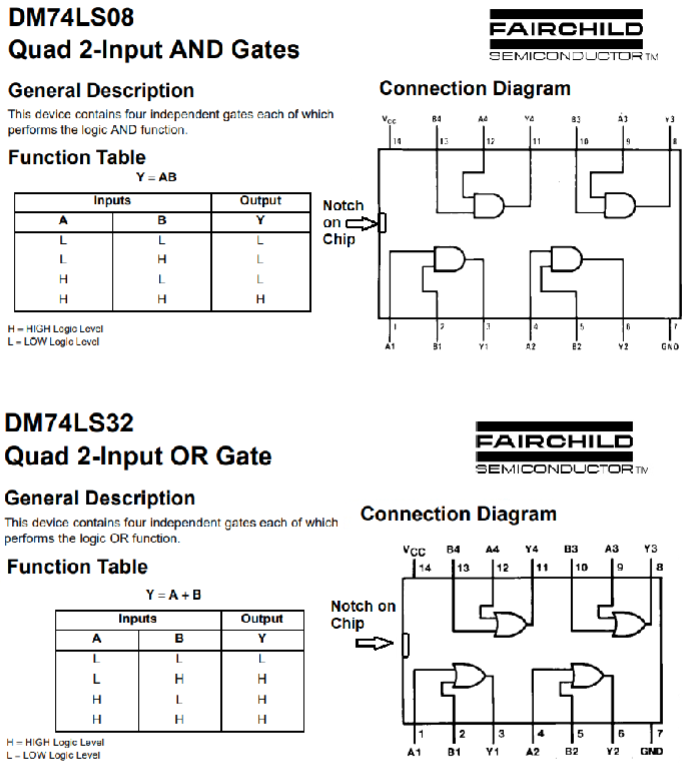
In this experiment, we applied our knowledge of logic gates and the provided hardware

(see Figure 5) to construct a combinational logic network of AND gates and OR gates to specify the behavior of a certain LED.

**Figure 4.** DE2-115 Logic Trainer Mode Inputs.



**Figure 5.** Gates Data Sheet



2.2 Assignment #2: Multi-IC Sequential Logic Design

For this assignment, we had to understand the LOAD and CLEAR operations for each of the ICs. The 74LS163 is synchronous in design while the 74LS161 is asynchronous is design. This means that the 163 functions on the clock cycle while the 161 does not. The desired or active action for both key operations occurs when a Logic 0 is applied. The LOAD loads the counter outputs at logic 0, while the CLEAR erases the previous outputs. The LOAD, CLEAR, and clock inputs were all driven by the DE2-115 board. We also required the basic knowledge of how binary counting works and the concept of significant bits.

3. Demonstration

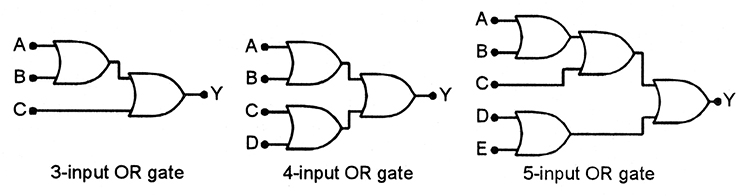
3.1 Assignment #1: Multi-IC Combinational Logic Design

For this assignment, we implemented the logic network seen in the figure below on the solderless breadboard connected to the DE2-115 using a 74LS08 (quad 2-input AND gate) IC and a 74LS32 (quad 2-input OR gate) IC. Each IC contains four two-input gates, meaning we had to implement the three-input OR gate by constructing it using two OR gates (see Figure 7), where the output of one will feed into the output of the other. This setup can be viewed in Figure 6. Following the diagram below and referring to the breakout board pinout to connect to the proper switches, we meticulously wired the circuit as seen in Photo 1.

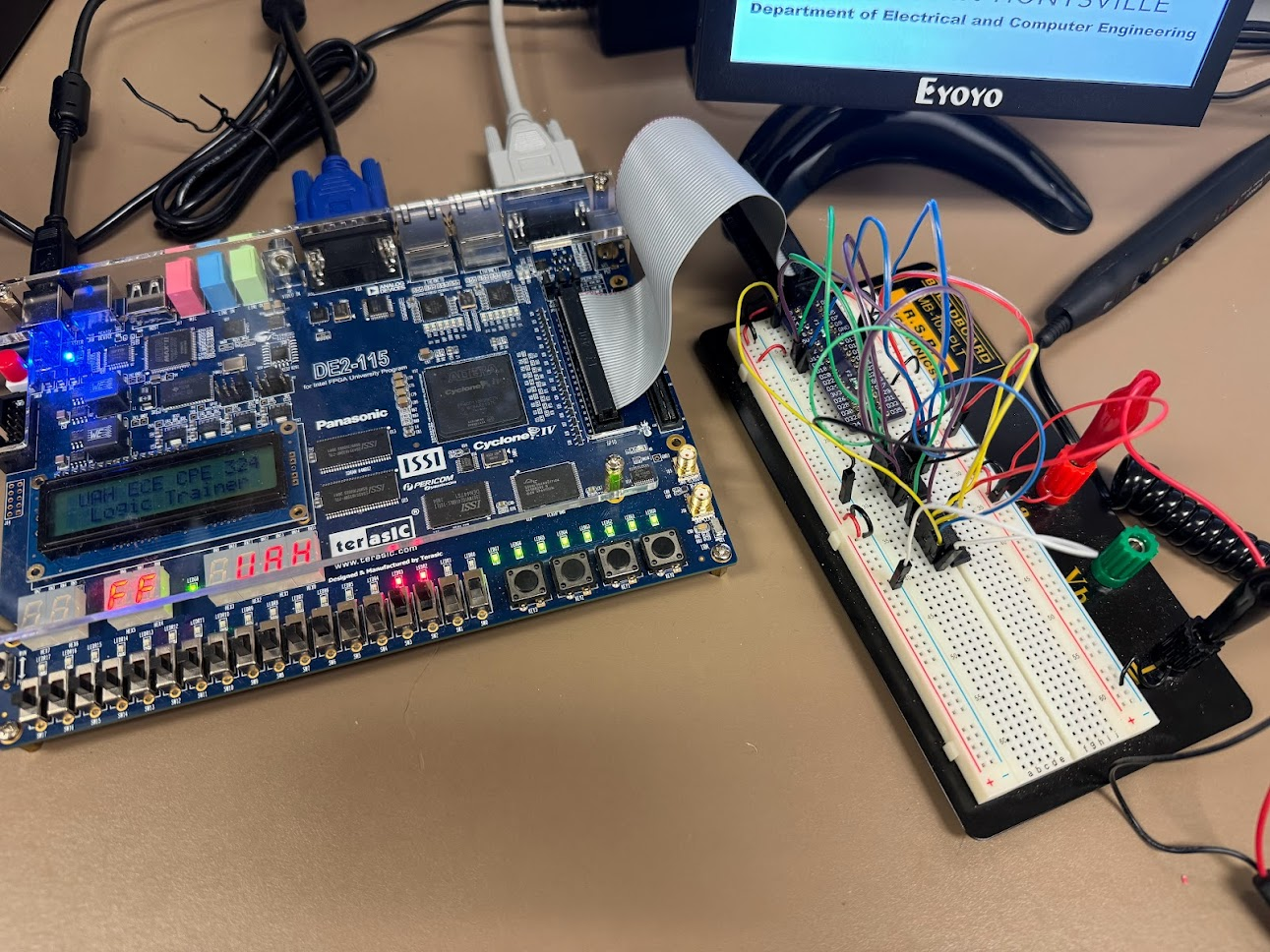
**Figure 6.** Combinational Logic Design



**Figure 7.** 3-Input OR Gate



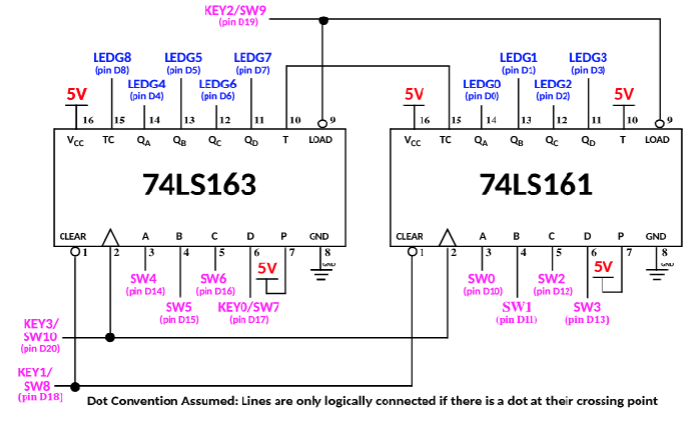
**Photo 1.** Combinational Logic Design Experimental Setup



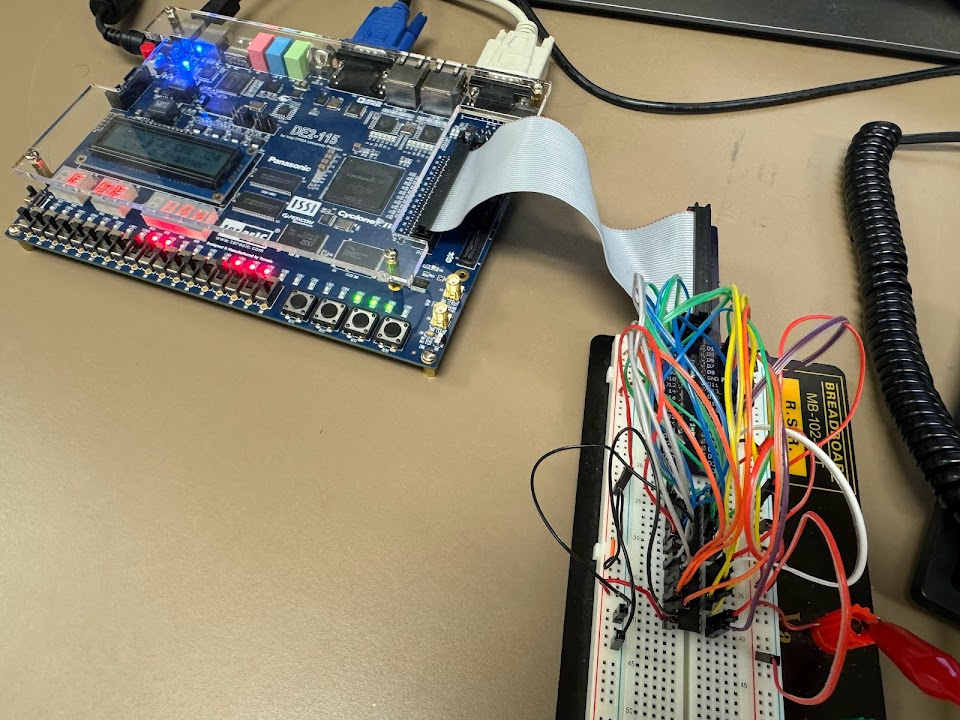
3.2 Assignment #2: Multi-IC Sequential Logic Design

Using Figure 8 below, we created an 8-bit counter using a single 74LS161 and a 74LS163 IC. We following the wiring and power guidelines as seen in the diagram.

**Figure 8.** Counter Design Circuit



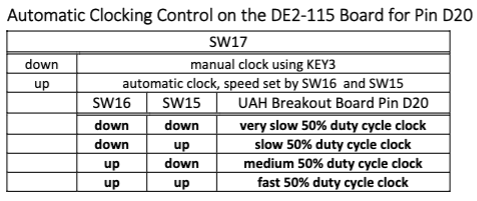
**Photo 2.** Sequential Logic Design Experimental Setup



To load an initial count value of 00001110, we set the switches for LOAD and CLEAR high using switches 7 and 8 to set inactive to begin, then setting the target value by setting SW1-2-3 high. Then we toggle the LOAD switch to logic 0 active and press KEY3 to generate a singular clock cycle. This is where we see the loading occur and the corresponding LEDs turn on. We then toggle the LOAD switch back to logic 1 to set it back to its inactive state.

We also automatically generated forward count starting from 00001100 to at least 00011111 using a slow speed clock by switching SW17 high, SW16 low, and SW15 either high or low depending on if very slow or slow speed based on the Figure 9 table. The counting is barely visible even at the “slow” speed, while at the medium and fast speeds, the change is no longer visible to the human eye.

**Figure 9.** Clock Speed Settings



4. Experimental Results

4.1 Assignment #1: Multi-IC Combinational Logic Design

**Table 1: Truth Table**

| Inputs (Test vectors) | | | | | | Output, O | |
| --- | --- | --- | --- | --- | --- | --- | --- |
| I0 | I1 | I2 | I3 | I4 | I5 | Expect | Actual |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 |

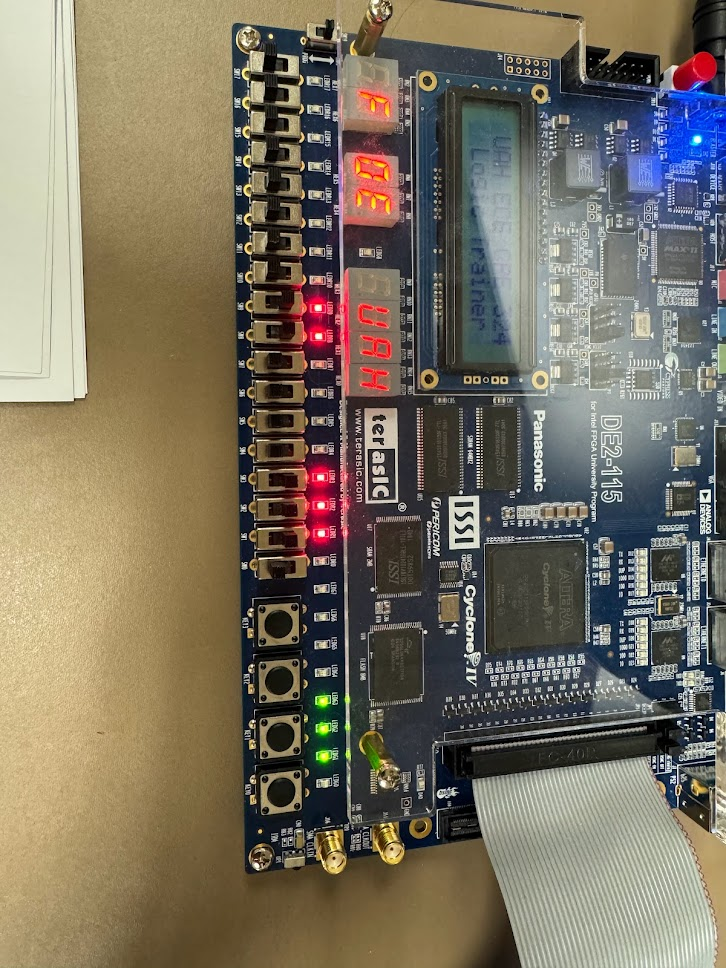
As observed in Photo 1, when switches 2 and 3 were set to high logic 1, LED0 is

also high logic 1 as expected using the truth table in Table 1. All cases from the table were tested by going through each switch combination and confirmed the output for the LED that we expected. The corresponding segment on the seven-segment display also followed the expected behavior of being inactive when the input is set to logic high and on when set to logic low 0.

4.2 Assignment #2: Multi-IC Sequential Logic Design

The experimental setup can be observed in Photo 2. Loaded data 00001110 example can be seen below in Photo 3. We were able to correctly load the counter as well as clear at different speeds and using both manual and automatic modes.

**Photo 3.** Loaded Data 00001110



Lab Questions

4.2.1 With the count value of the counter set initialized to 00001110, generating a manual clock by pressing the KEY3 button for three clock cycles increments the count value by 1 for each clock cycle. The ripple counter, also called the Terminal Count TC, remains unchanged because the 8-bit counter has not reached overflow three clock cycles after 00001110, or 00010001.

4.2.2 When we manually advanced the clock signal five clock cycles after temporarily rewriting the input P on Pin 7 o the 74LS161 IC so that it is connected to ground instead of 5V, the counter does not advance because both countable inputs (P and T) must be high to count and the cascading effect was disrupted. Returning the input P to 5V resolves the issue and continues counting up one for each clock cycle when manually advanding the clock signal an additional five clock cycles.

4.2.3 Running the clock with IC in automatically generated clocking mode with clock set to a high speed (SW17, SW16, SW15 all in the up position), the output appears statically set high for all LED outputs because the speed is so fast that we cannot detect the changes, but it is counting up very quickly. TC also appears solid, indicating the overflow from counting up to the maximum value is occurring at a rapid rate as well.

4.2.4 Activating the CLEAR input without advancing the clock with the count value initialized to 11111111 clears only the four least significant bits, resulting in 11110000. Then, with the CLEAR input still activated and advancing the clock one cycle, the remaining four bits are cleared to 0. This is because the 163 IC, which controls the four most significant bits, is synchronous and executes based on the clock, whereas the 161 IC is asynchronous and does not rely on the clock.

4.2.5 If the SW10 key was used to manually clock the design instead of KEY3, the design would maintain the same functionality since the two are internally electrically connected on the DE2-115, feeding into the same D20 output pin on the breakout board.

5. Conclusion

5.1 Assignment #1: Multi-IC Combinational Logic Design

In this experiment, we learned how to use and implement logic gates to control an LED output utilizing the DE2-115, 74LS08, 74LS32, breakout board, and solderless breadboard. We learned how to interface with the switches on the FPGA and wire elements together, understanding the need for a common ground and VCC. No significant errors were faced.

5.2 Assignment #2: Multi-IC Sequential Logic Design

Through this Multi-IC sequential logic design activity, we were able to cascade two simple 4-bit binary counter ICs to create an 8-bit binary counter and understand the basic differences between asynchronous and synchronous control signals through the behavior of each ICs’ clear counter. It was difficult at first to understand how the loading worked, but by understanding the underlying concepts of synchronous control signals, everything else fell into place.