**PURPOSE**

The purpose of this lab was to design and build a register file with four addresses, each capable of storing four bits (4x4). The circuit was able to write data to any of the four available memory addresses using a write enable signal in conjunction with a write address signal. A read address input signal was then used to read data from any of the addresses. The data being written to the register and the data read from the register were each displayed using a seven-segment display.

The Altera Quartus II software was used to design and test initial implementations of the circuit, after which the circuit was designed on a protoboard and tested using the Intronix LogicPort Logic Analyzer software. An extensive set of tests was generated and carried out on the circuit to verify that the correct data values would be written to and read from the register file.

**PROCEDURE**

1. A new project was created in the Altera Quartus II simulation software. A new schematic file was added to the project.

2. The following components were added to the schematic file: one 74LS670 4x4 register file, two 74LS193 synchronous counters, and two 74247 seven-segment display decoders. One of the 74LS193 counters was to be used to generate the write address signal, while the other was to be used to generate the read address signal. One of the 74247 decoders was to be used to display the inputs written to the register file, while the other was to be used to display the outputs read from the register file.

3. The GWN pin on the 74670 was tied to the Write Enable input. Since this is an active-low signal, the Write Enable input was set to 1 until a value was to be written to the register file. The WA and WB pins on the 74670 were connected to the two least-significant bits from the 74193 counter for the write address, while the RA and RB pins on the 74670 were connected to the two least-significant bits from the 74193 counter for the read address. This let each counter control the register to which or from which data was to be written or read. Input pins D1-D4 on the 74670 were connected to four external data inputs.

4. The external data inputs D0-D3 were connected to input pins A-D on the first 74247 decoder. This decoder was to display the data that was being written into the register file. For the second decoder, which was to display the data being read from the register file, input pins A-D were connected to the output pins from the register file. The register file outputs were also connected to four separate output pins Q0-Q3. The full schematic is shown in Figure 1.

5. A new waveform file was added to the project. Using the Node Finder option, the following signals were added to the waveform: Write Address, Read Address, Write Enable, the data inputs D0-D3, the register file outputs Q0-Q3, the decoded outputs of the first 74247 decoder OA\_D – OG\_D, and the decoded outputs of the second 74247 decoder OA\_Q – OG\_Q.

6. A set of four write instructions was generated and implemented using the input pins in the waveform. These four instructions each wrote a value into a different register. Altogether, these values encompassed a wide range of values that the register would have to handle. Four read instructions were then added to the waveform to verify that each register was holding the value that had been written to it.

7. The inputs of each waveform had to be altered to write in the data inputs in the instruction. The procedure for writing data to a register was as follows:

1. Set write address to correct register
2. Set data to value that was to be written
3. Assert the Write Enable signal (Write Enable is active low so it must be set to 0)
4. Deassert the Write Enable signal (Set to 1)

8. The file was compiled and simulated. The values of the output decoder were compared with the instructions and the register file was verified to be correctly loading and storing data.

9. The circuit was implemented on a protoboard using an 8-pin DIP switch to provide the data inputs D0-D3. The circuit also contained three SPDT debounced logic sources to be used for Write Enable, Write Address, and Read Address.

10. A new project was created in the LogicPort Logic Analyzer software to observe the circuit. Since the circuit would be tested by manually setting the values of the data, Read Address, Write Address, and Write Enable, some settings in the LogicPort software had to be adjusted so it could sample data from a longer time period. The Voltage Threshold value was decreased to 0.85 V. The clock frequency was decreased to 10 kHz. The pre-fill time limit was set to -0.5 seconds and the post-fill time limit was set to one minute.

11. The Read Address and Write Address inputs were set to be the clock signals of this waveform, as it was necessary for the other signals to change only in response to these two signals. The other signals were to trigger on the rising edge of the Read Address and Write Address inputs. The same instructions used to test the Altera Quartus schematic were used to test the physical circuit.

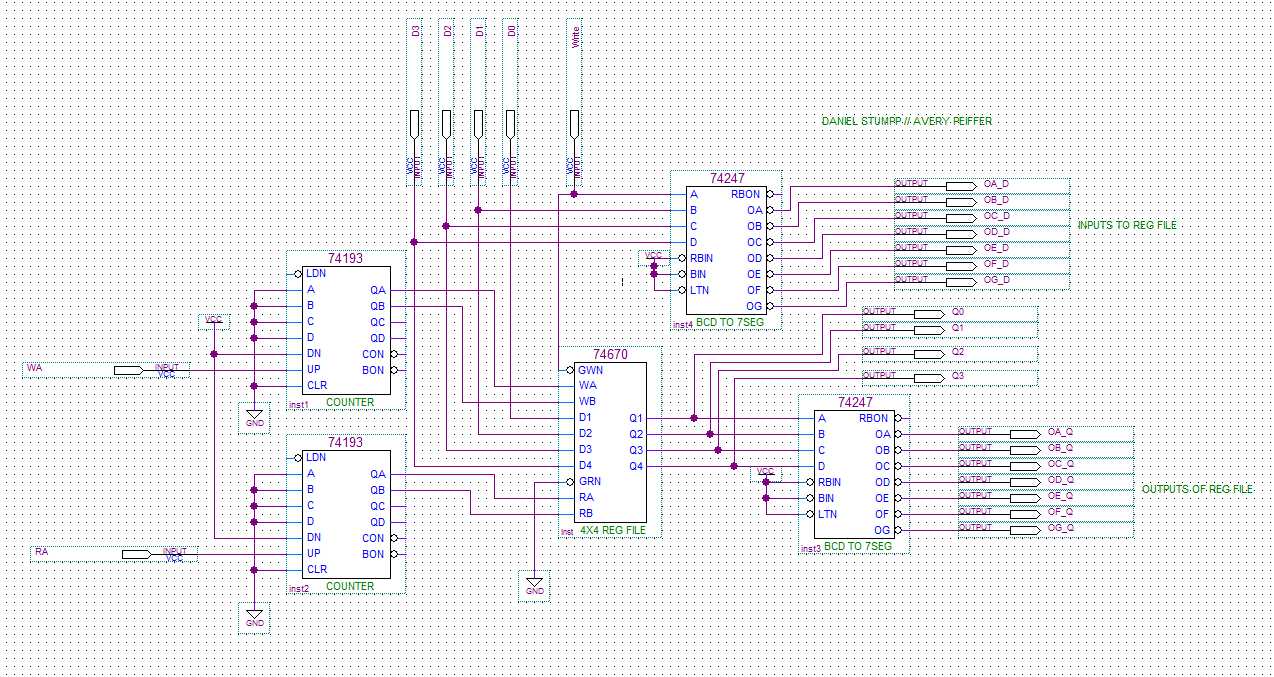
**RESULTS**

Figure 1: The 4x4 register file circuit designed in an Altera Quartus schematic

Figure 1 above shows the 4x4 register file schematic designed in the Altera Quartus software. To test the schematic, the following instructions were implemented by toggling the inputs in the Waveform Editor (in order):

1) Load 3 to Register 1

2) Load 15 to Register 3

3) Load 7 to Register 2

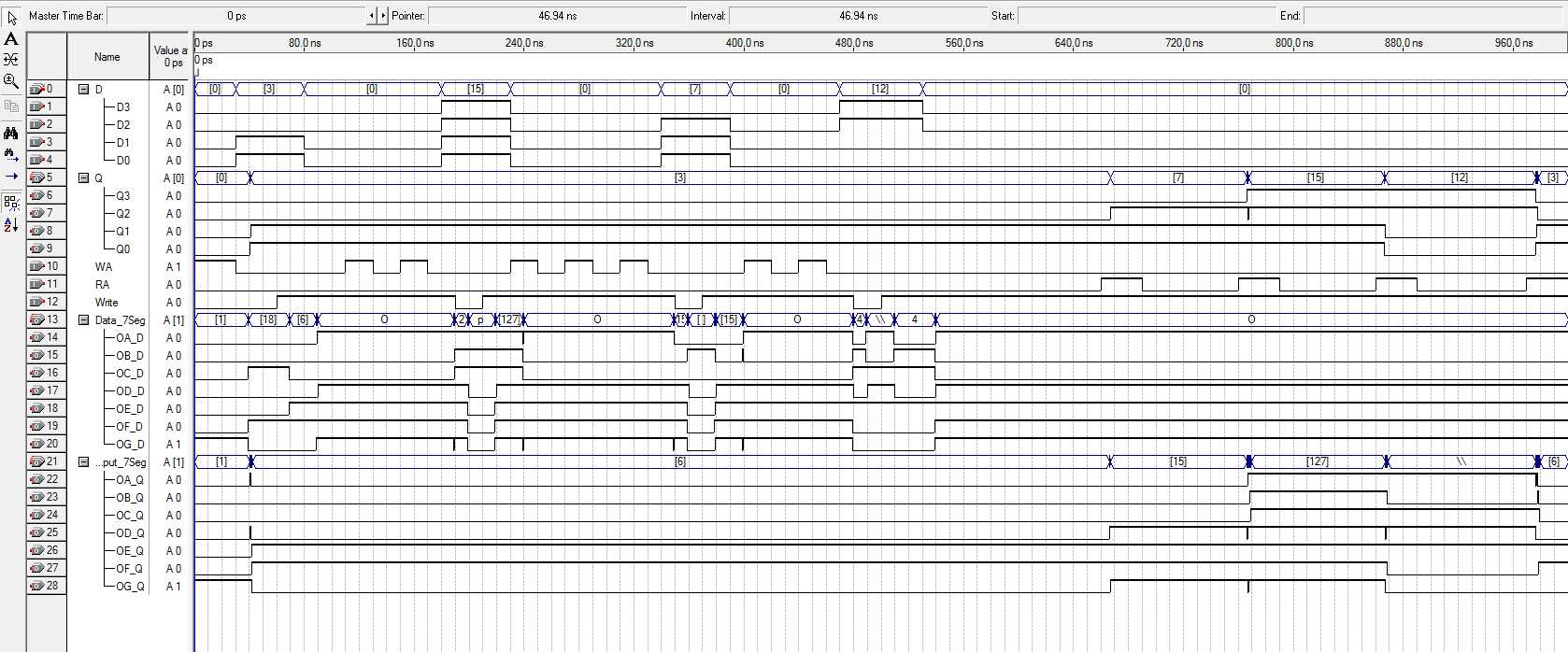
4) Load 12 to Register 0

5) Read Register 2

6) Read Register 3

7) Read Register 0

8) Read Register 1

These tests encompassed the range of values (0-15) that would be read from and written to the register file. The first four instructions were done out of order to confirm that the two 74193 counters correctly reset after counting to 11. The output of each read instruction was observed in the Waveform Editor and is shown in Figure 2.

**2 3 0 1**

Figure 2: The Altera Quartus waveform of the register file circuit. From 0 ns to approximately 560 ns are the four write instructions. From approximately 640 ns to 1000 ns are the four read instructions. The numbers under each assertion of the read address signal represent the register from which data is being read.

The 74247 seven-segment decoders have active-low outputs, so for each read instruction, the values that went to 0 among the Output\_7Seg group (Nodes 22-28) were used in conjunction with the seven-segment display diagrams, shown below in Figures 3 and 4, to verify the contents of the register. At approximately 660 ns, the data from Register 2 is read. The signals from the Output\_7seg group observed to be low are A, B, and C. A, B, and C being turned on in the seven-segment display corresponds to the number 7, confirming that Register 2 held the correct value. For Register 3, no values are low, corresponding to the hexadecimal F. For Register 0, the signals B, F, and G are low, corresponding to the hexadecimal value C and the decimal value 12. For Register 1, the signals A, B, C, D, and G are low, corresponding to 3.

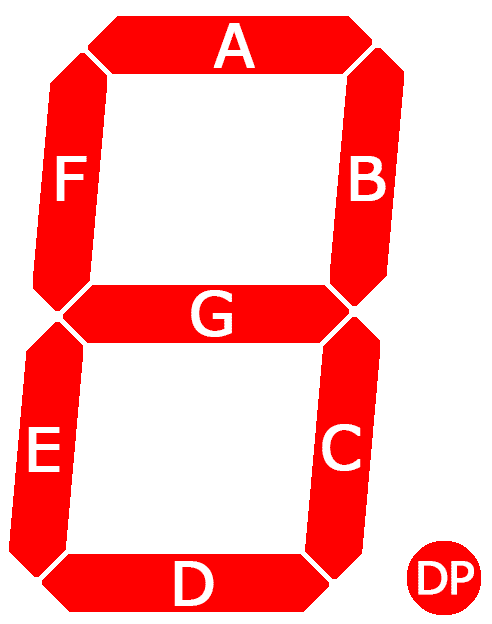


Figure 3: The decimal layout for the seven-segment display used for the values of 0-9.[1].

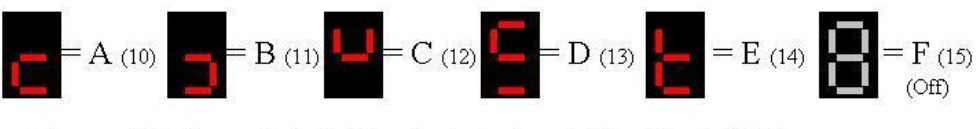


Figure 4: The hexadecimal layout for the seven-segment display used for the decimal values of 10-15 [2].

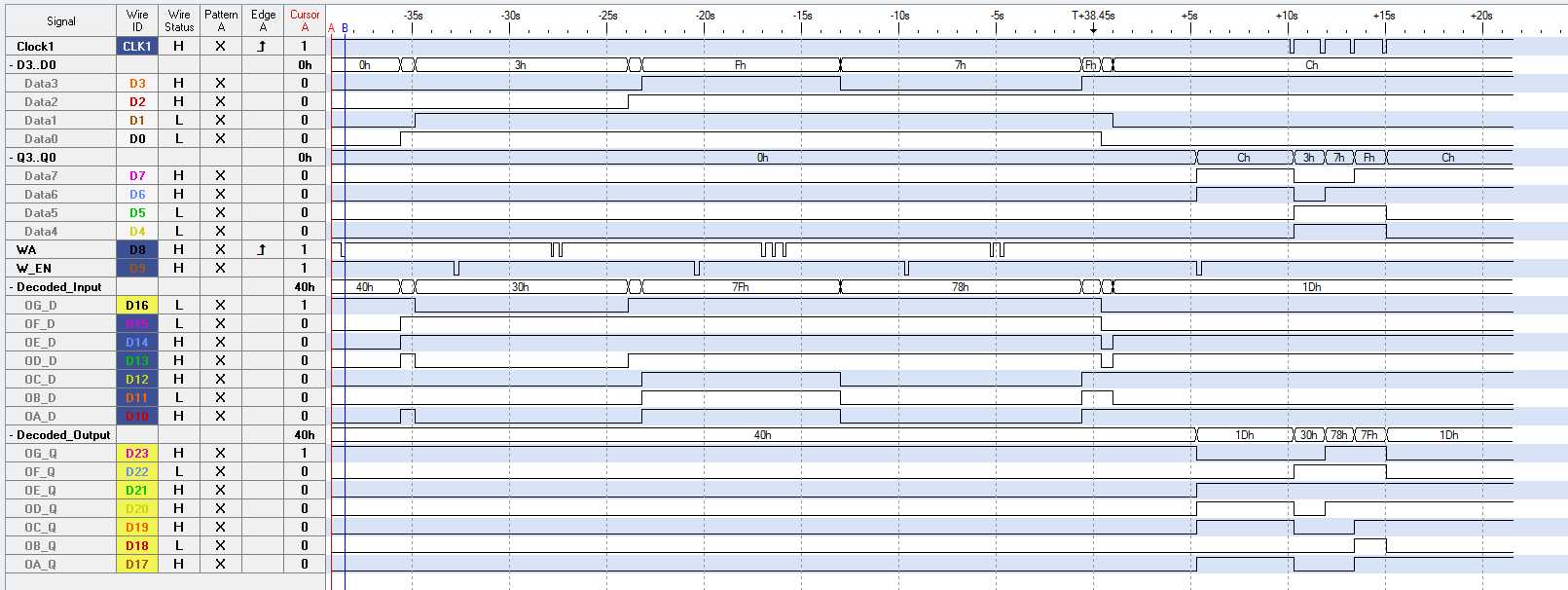
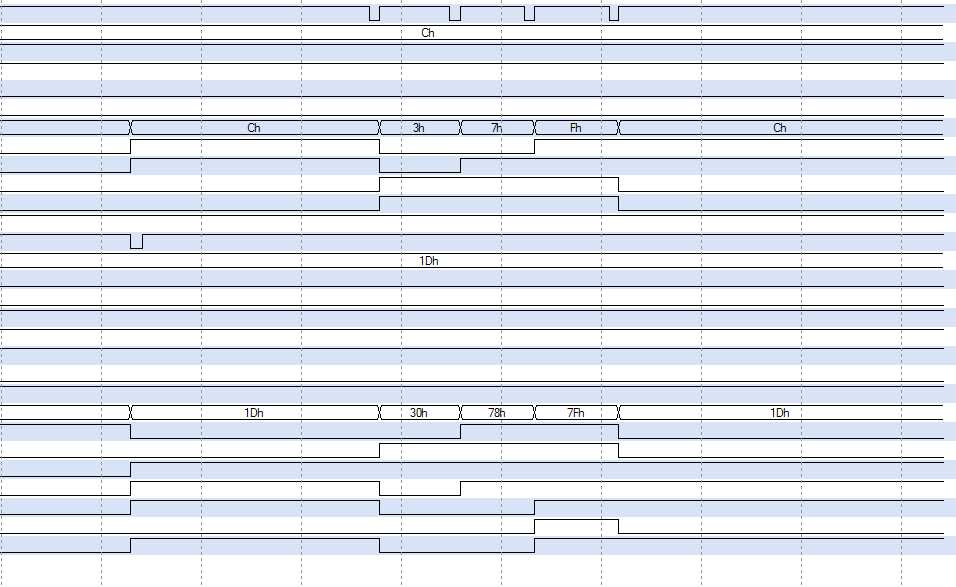
When the physical circuit was assembled, it was connected to the LogicPort Logic Analyzer. As detailed in the Procedure section, the timing settings were changed so the LogicPort software would be able to capture data from the entire set of tests. The entire Logic Analyzer waveform is shown below in Figure 5. The latter part of this waveform, in which the data from each register is read, is shown in more detail in Figure 6.

Figure 3: Decimal layout for (values 1-10) for the seven-segment display used [1].

Figure 4: The hexadecimal output diagram for the seven-segment display used [2].

Figure 5: The entire waveform output for the Logic Analyzer software. Data is being written to the registers at the points in which the W\_EN signal is asserted low, in the first half of the waveform. Data is being read from the registers at the points in which Clock1, or Read Address, is being increased, in the latter third of the waveform. Note that OA\_Q is at the bottom of the waveform and OG\_Q is above it.

 In this test, the registers were read in order, starting with Register 0 and ending at Register 3. There is no change of the Read Address signal needed to read the contents of Register 0 because the Read Enable signal is always asserted, meaning one of the registers is continuously being read. Since Read Address is 0 by default, Register 0 is read automatically.. On each subsequent change of the Read Address signal (shown at the top of Figure 6), the signals that are asserted low correspond to the value held in that register. These signals match those observed in the Altera Quartus simulation waveform.

**R0 R1 R2 R3**

**G**

**F**

**E**

**D**

**C**

**B**

**A**

Figure 6: The latter third of the waveform, in which values are being read from each register. The signals A-G are labeled on the left side, while the registers from which data is being read are labeled above signals A-G.

**CONCLUSIONS**

In this lab, a 4x4 register file was designed and built. The register file was capable of reading and storing four bits of data in one of four separate registers. Two counters were used to access the different registers, while two seven-segment displays were used to display the values being stored to and read from the register. The circuit schematic was tested on the Altera Quartus software, while the physical circuit was tested using the LogicPort Logic Analyzer. A set of instructions were generated to verify that the register file was correctly reading and storing values. These instructions were carried out on both the Altera Quartus simulation and the physical circuit to verify that the register could read and write values correctly.

**REFERENCES**

[1] “How to Set up 7-Segment Displays on the Arduino,” *Circuit Basics*, 13-Aug-2018. [Online]. Available: http://www.circuitbasics.com/arduino-7-segment-display-tutorial/. [Accessed: 31-Mar-2019].

[2] Seven Segment LED Display Layout Sheet

[3] Dr. Alex Jones’ Laboratory Manual

[4] Data Sheets for the 74LS193, 74LS247, and 74670 devices

[5] Partner Daniel Stumpp

**ACKNOWLEDGEMENTS**

I would like to acknowledge Dr. Alex Jones and the ECE 0501 course TAs for all of the work they do every class period to help students. It must be extremely difficult and stressful to field questions from 50 groups, each of which may have a different error in their circuit. It is very relaxing to know that there is a team of individuals who are willing to perform tests themselves in an effort to help us identify a bug if we have not been able to do so.