**PURPOSE**

The purpose of this lab was to implement the Arithmetic Logic Unit (ALU) circuit from Lab 12 on an Altera DE2 Field Programmable Gate Array (FPGA) chip. The input and output pins of the ALU circuit were reassigned to pins on the FPGA board using the existing schematic on the Altera Quartus software, after which the circuit was downloaded to the FPGA board and tested using the same tests executed on the ALU circuit in Lab 12.

After the circuit was tested on the FPGA chip, several components of the circuit were replaced by components described using VHSIC Hardware Description Language (VHDL). After each component was replaced, the circuit was re-compiled, downloaded, and tested using the same operations as those implemented in Lab 12.

**PROCEDURE**

1. The Altera Quartus schematic file from Lab 12 was opened and copied into a new folder, named Lab 13.

2. The 7474 flip-flops used in the four-bit latch component from Lab 12 were replaced with “Latch” components. This new four-bit latch circuit, shown in Figure 1, was tested by reading in several values using the L\_EN input and was added to the high-level schematic. The seven-segment displays were also removed from the ALU schematic.

3. The pins on the FPGA board were assigned to the input and output pins on the Altera Quartus schematic. These pin assignments are shown in Table 1.

4. Each circuit input was assigned to one of the toggle or pushbutton switches on the Altera DE2 board. The read address and write address outputs were connected to one of the LED displays on the board. These pin assignments are also shown in Table 1.

5. The circuit was recompiled and programmed to the FPGA board using the Tools – Programmer menu on the Altera Quartus software. The circuit was tested using the arithmetic and logical operations from Lab 12.

6. The file DispBin.vhd was opened in the Quartus software. This file contained a VHDL description of a seven-segment decoder. This code was compiled, creating a symbol for the seven-segment decoder that was used to replace the same component from the schematic.

7. The revised schematic was recompiled. Step 6 was repeated to replace the other seven-segment decoder with a VHDL description.

8. The schematic was simulated using the same operations as those used for testing in Lab 12. The resulting waveform file was compared to the waveform file generated in Lab 12.

9. The VHDL code for the seven-segment decoder was modified to show the hex character A, b, C, d, E, and F. The schematic was recompiled, and the FPGA board was reprogrammed using this revised description.

10. A VHDL description was created for the 4x2:1 multiplexor. The code was compiled, and the component was tested by itself. After the component was tested, it was placed into the ALU schematic, replacing the original multiplexor.

11. The ALU schematic was recompiled and resimulated using the same operations from Lab 12 to verify that the multiplexor worked correctly.

12. Steps 10 and 11 were repeated to replace the 4-bit latch and the address counters with VHDL-generated equivalent components.

**RESULTS**

Figure 1 below shows the updated schematic for the four-bit latch after each individual component was replaced with the “Latch” component from the Quartus library. Since this component is explicitly designed to be a latch, there was no need to connect the output of each flip-flop to the input of the next, as was done previously. Figure 2 further below shows a simple test of the four-bit latch sub-circuit in which three different data values were read into the latch using the L\_EN input.

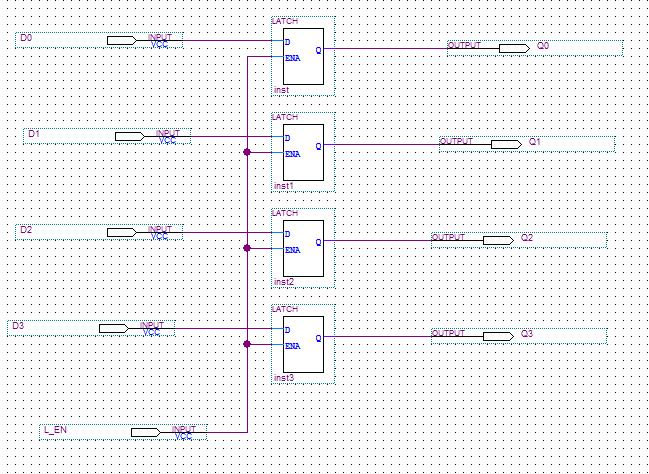


Figure 1: Updated schematic for the 4-bit latch circuit using the LATCH components in Quartus.

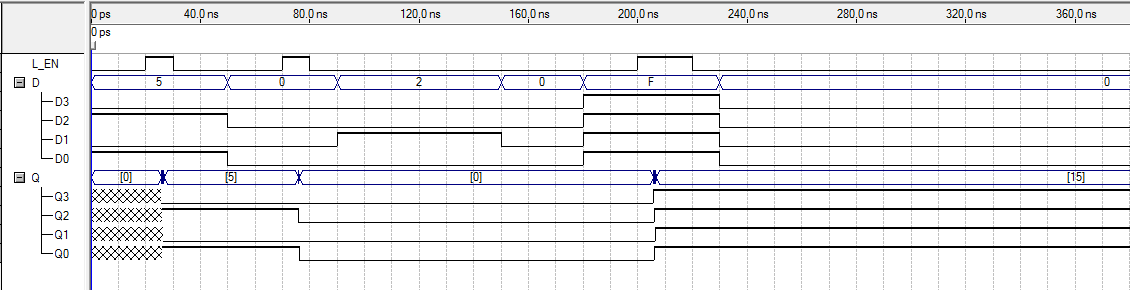
****

Figure 2: Altera Quartus waveform testing the new four-bit latch by reading in 5, 2, and 15.

Table 1 below shows the pin assignments that were made from the Altera Quartus schematic to the Altera DE2 FPGA chip. The Read Address, Write Address, and Write Enable were each assigned to one of the pushbutton switches. Latch Enable, Input Select (for the multiplexor), the four Data bits, the four ALU Function bits, and M were each assigned to one of the toggle switches. When testing the FPGA chip, it was somewhat difficult to physically assert Read Address, Write Address, and Write Enable, as the pushbutton switches were somewhat desensitized from repeated usage prior to this lab.

|  |  |  |
| --- | --- | --- |
| **Node** | **Location** | **Type** |
| LED\_DATAIN\_0 | PIN\_Y23 | HEX3 |
| LED\_DATAIN\_1 | PIN\_AA25 | HEX3 |
| LED\_DATAIN\_2 | PIN\_AA26 | HEX3 |
| LED\_DATAIN\_3 | PIN\_Y26 | HEX3 |
| LED\_DATAIN\_4 | PIN\_Y25 | HEX3 |
| LED\_DATAIN\_5 | PIN\_U22 | HEX3 |
| LED\_DATAIN\_6 | PIN\_W24 | HEX3 |
| LED\_DATAOUT\_0 | PIN\_AF10 | HEX0 |
| LED\_DATAOUT\_1 | PIN\_AB12 | HEX0 |
| LED\_DATAOUT\_2 | PIN\_AC12 | HEX0 |
| LED\_DATAOUT\_3 | PIN\_AD11 | HEX0 |
| LED\_DATAOUT\_4 | PIN\_AE11 | HEX0 |
| LED\_DATAOUT\_5 | PIN\_V14 | HEX0 |
| LED\_DATAOUT\_6 | PIN\_V13 | HEX0 |
| LED\_L\_7 | PIN\_V20 | HEX1 |
| LED\_L\_8 | PIN\_V21 | HEX1 |
| LED\_L\_9 | PIN\_W21 | HEX1 |
| LED\_L\_10 | PIN\_Y22 | HEX1 |
| LED\_L\_11 | PIN\_AA24 | HEX1 |
| LED\_L\_12 | PIN\_AA23 | HEX1 |
| LED\_L\_13 | PIN\_AB24 | HEX1 |
| D0 | PIN\_N25 | Toggle Switch 0 |
| D1 | PIN\_N26 | Toggle Switch 1 |
| D2 | PIN\_P25 | Toggle Switch 2 |
| D3 | PIN\_AE14 | Toggle Switch 3 |
| RA\_0 | PIN\_AE23 | LED Red 0 |
| RA\_1 | PIN\_AF23 | LED Red 1 |
| WA\_0 | PIN\_AA20 | LED Green 6 |
| WA\_1 | PIN\_Y18 | LED Green 7 |
| S0 | PIN\_C13 | Toggle Switch 7 |
| S1 | PIN\_B13 | Toggle Switch 8 |
| S2 | PIN\_A13 | Toggle Switch 9 |
| S3 | PIN\_N1 | Toggle Switch 10 |
| IN\_SEL | PIN\_AF14 | Toggle Switch 4 |
| L\_EN | PIN\_AD13 | Toggle Switch 5 |
| M | PIN\_AC13 | Toggle Switch 6 |
| RA | PIN\_G26 | Pushbutton 0 |
| WA | PIN\_N23 | Pushbutton 1 |
| Write | PIN\_P23 | Pushbutton 2 |

Table 1: Pin Assignments for the FPGA chip.

The FPGA chip was tested using the operation 3 + 4, shown in Figure 3. HEX3 contains the latch value, 3, while HEX1 contains the register value, 4. HEX2 is displayed as F because no outputs have been mapped to that display. The output, 7, is shown on the right of the display in HEX0.

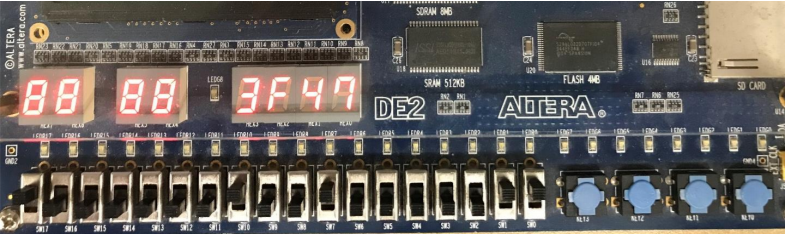


Figure 3: FPGA chip display after executing the test 3 + 4 = 7.

Figure 4 below shows the VHDL code that was used to reprogram the DispBin decoder so that it would show the hex characters A, b, C, d, E, and F to represent the values 10-15. When the decoders were replaced with the VHDL-generated symbols, the ALU circuit was simulated using two arithmetic operations and two logical operations. This simulation is shown in Figure 5. The two arithmetic operations tested were 3 + 4 and 10 – 5 – 1. The two logical operations tested were 3 AND 4’ and 10 OR 5.

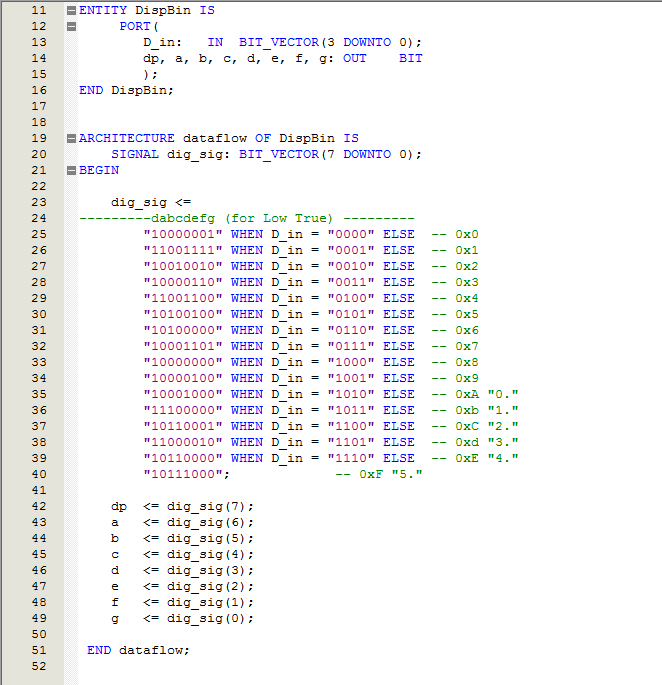


Figure 4: VHDL code used to redesign the decoder component so that it would display the hex values for 10-15.

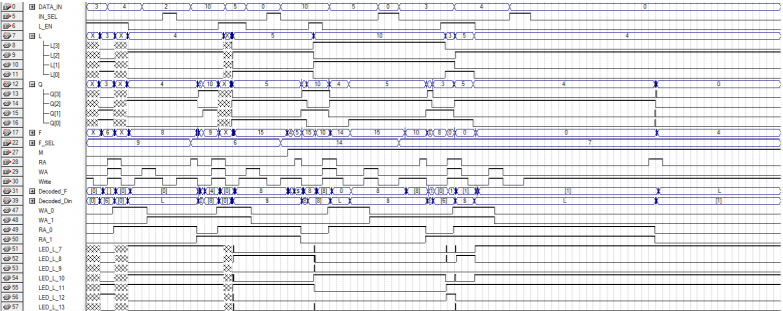


Figure 5: Simulation of two arithmetic operations and two logical operations after the VHDL decoder components were added.

Figures 6-8 show the VHDL code that was used to redesign the two-bit counters, multiplexor, and four-bit latch, respectively. After each component was redesigned using VHDL, it was implemented in the ALU schematic on the Altera Quartus software and tested.

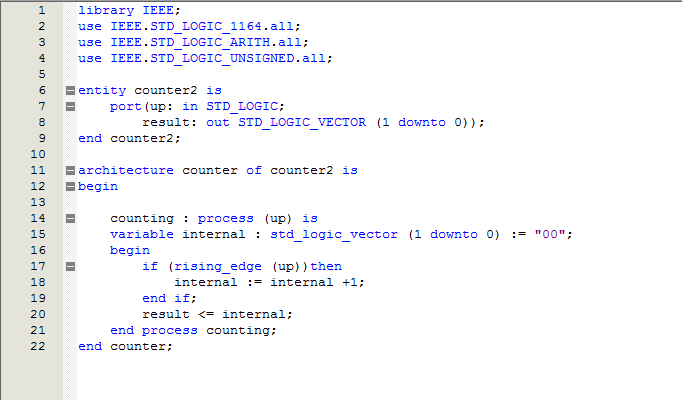


Figure 6: VHDL code used to redesign the two-bit counters used for the Read Address and Write Address signals.

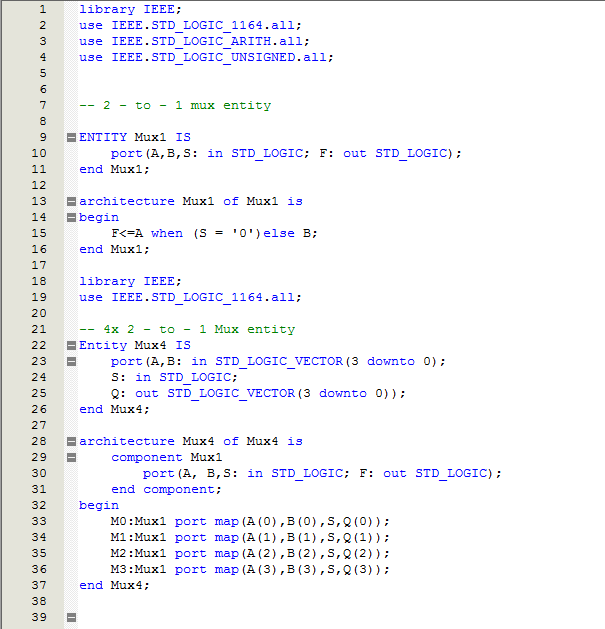


Figure 7: VHDL code used to redesign the multiplexor component for the ALU circuit.

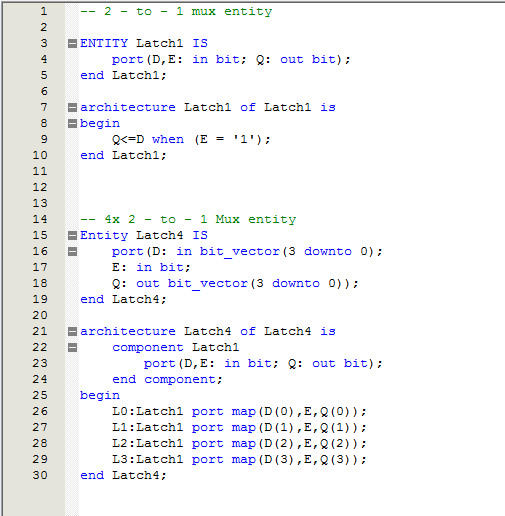


Figure 8: VHDL code used to redesign the four-bit latch component for the ALU circuit.

Figure 9 below shows the final schematic after the four-bit latch, the multiplexor, the decoders, and the two-bit counters were redesigned using VHDL. The final schematic was tested using the same arithmetic and logical operations that were used to test the circuit with only the VHDL decoder components added. This waveform is shown in Figure 10.

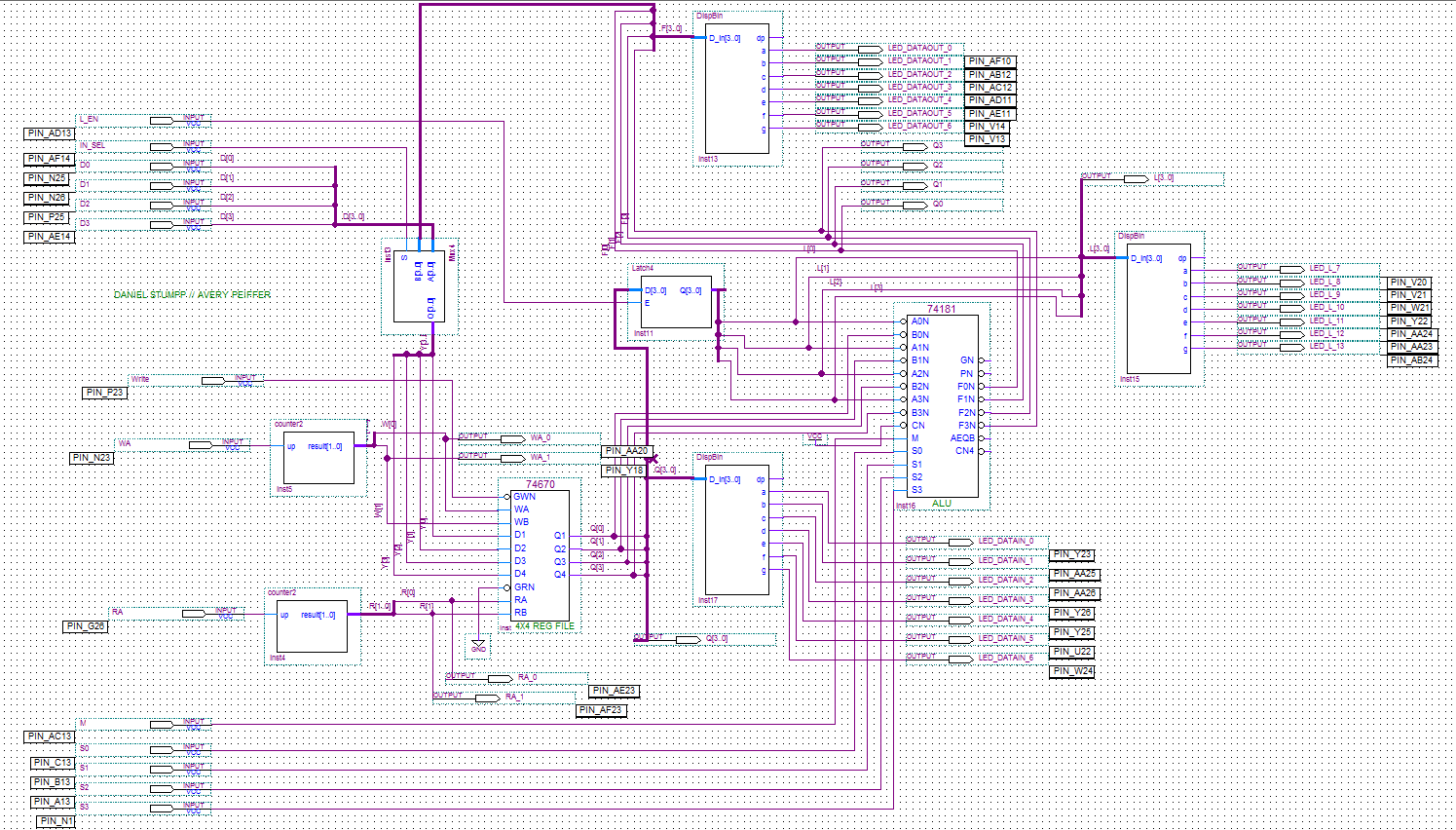


Figure 9: Final ALU schematic after several components were replaced with VHDL-coded symbols.

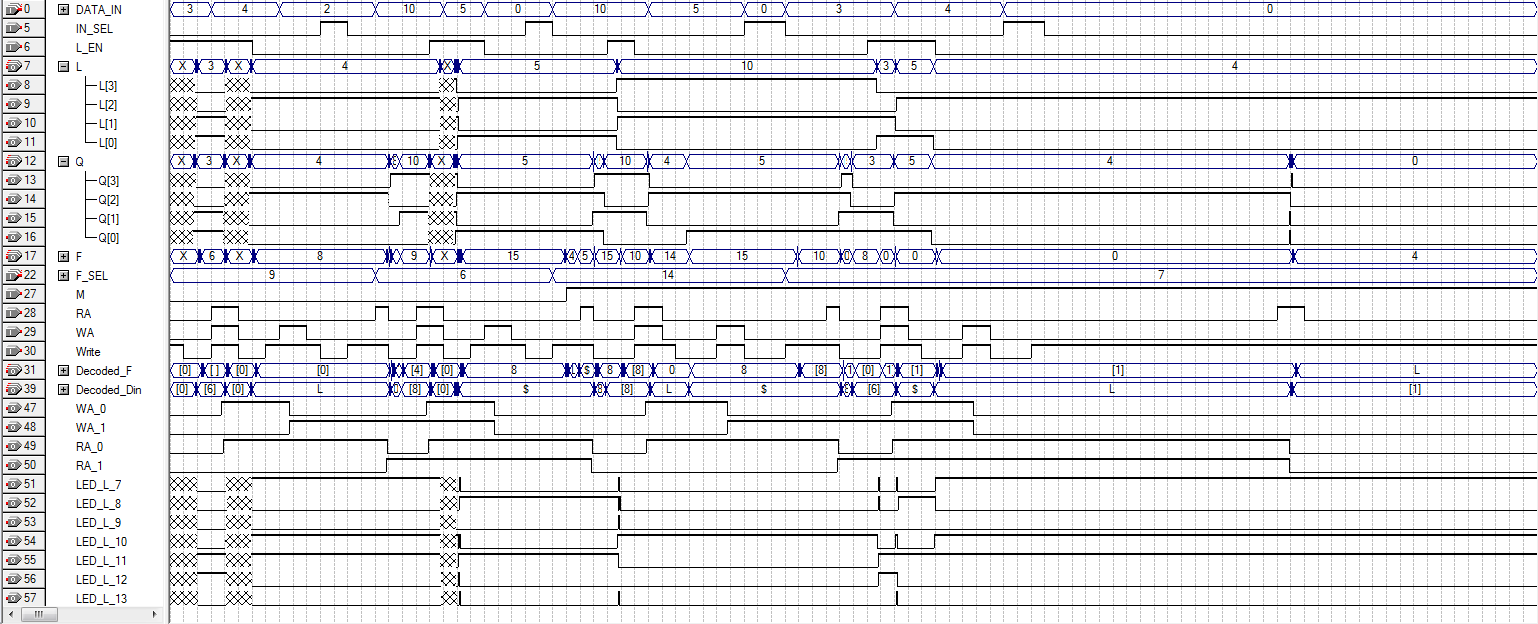


Figure 10: Testing the final schematic using two arithmetic and two logical operations.

**CONCLUSIONS**

In this lab, an FPGA chip was used to implement the ALU circuit from Lab 12. The input and output pins of the design were assigned to the FPGA chip, after which it was programmed by importing the schematic used in the Altera Quartus software. The FPGA chip was tested using a subset of the operations used to test the ALU circuit in Lab 12. Using the FPGA chip to test the schematic was an extreme increase in efficiency and convenience, though the pushbutton switches on the chip are worn down from repeated use and so are difficult to use. Several components of the ALU circuit were replaced by VHDL-generated symbols. Doing so showed that VHDL can be used to generate components that are explicitly needed for a circuit and therefore modularize the process of digital design.

**REFERENCES**

[1] Dr. Alex Jones’ Laboratory Manual

[2] Partner Daniel Stumpp