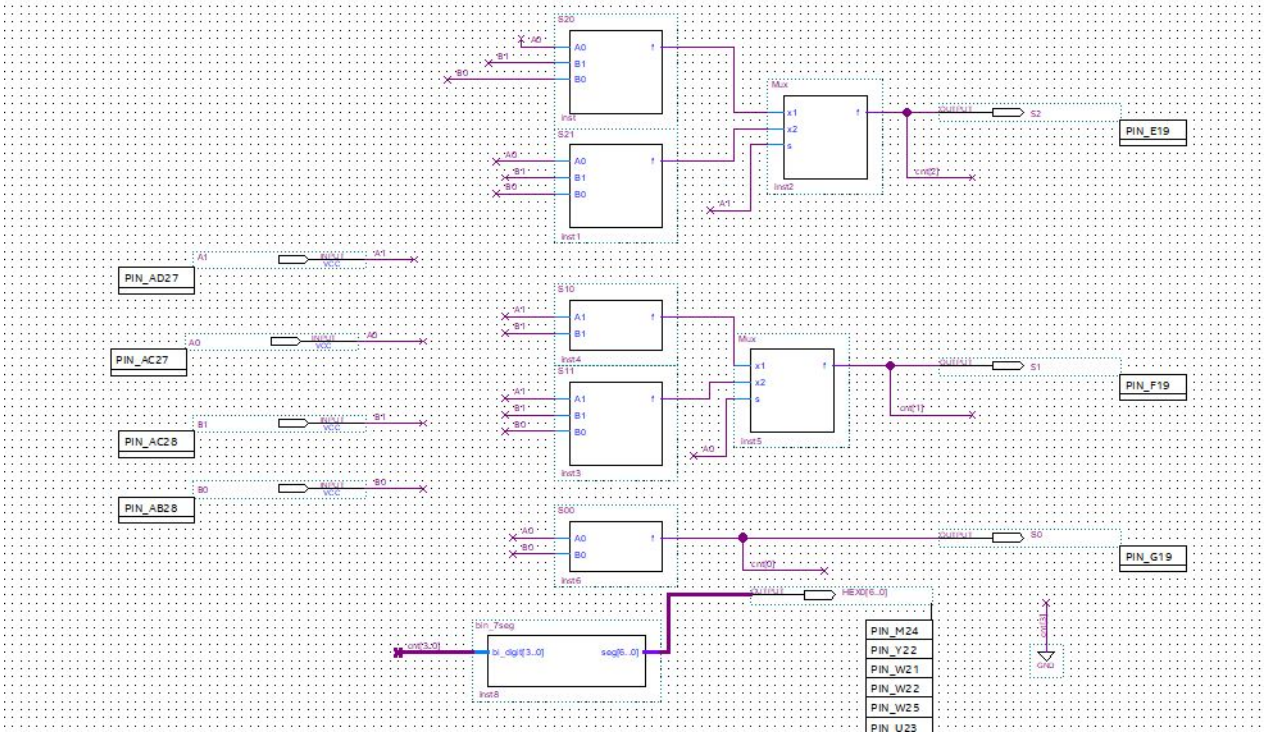
Lab 4 Report
Connor Callahan
Dillon Kanai

Write an introduction describing the behavior of your circuit and what design and implementation decisions you made.

Introduction:

In this lab, we simulated the addition of two two-bit numbers. The first number consisted of two bits: A1 and A0. Likewise, the second number consisted of B1 and B0. We first used k-maps to determine the behavior of the circuit and verilog code to program the addition part of this circuit (code shown below). Then, we put the verilog code into a higher priority circuit that involved using muxes and a 7 segment display. We simply copy and pasted the 7 segment display from lab 3. After that we loaded it onto the FPGA. The final result should show LEDs lighting up (which represented the sum in binary form) and the sum in decimal on the 7 segment display. We didn't use a third mux for the third part of the circuit because the end function was the same (regardless of what the select value was).

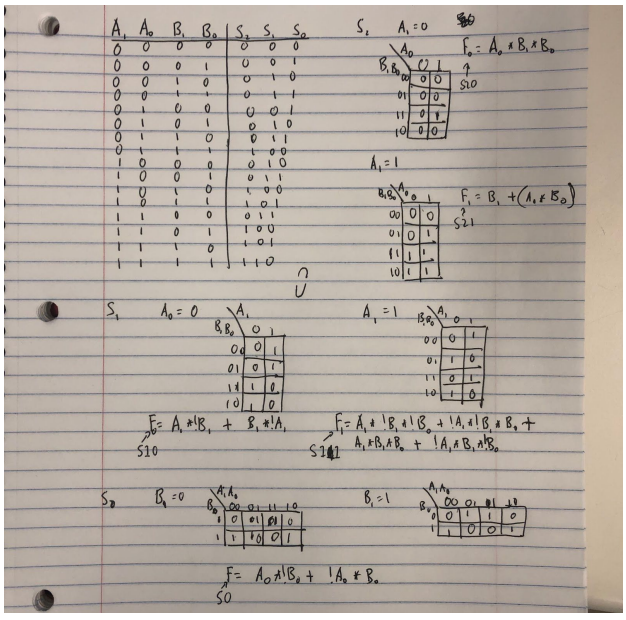
Circuit with Verilog code, muxes, and 7 segment display:



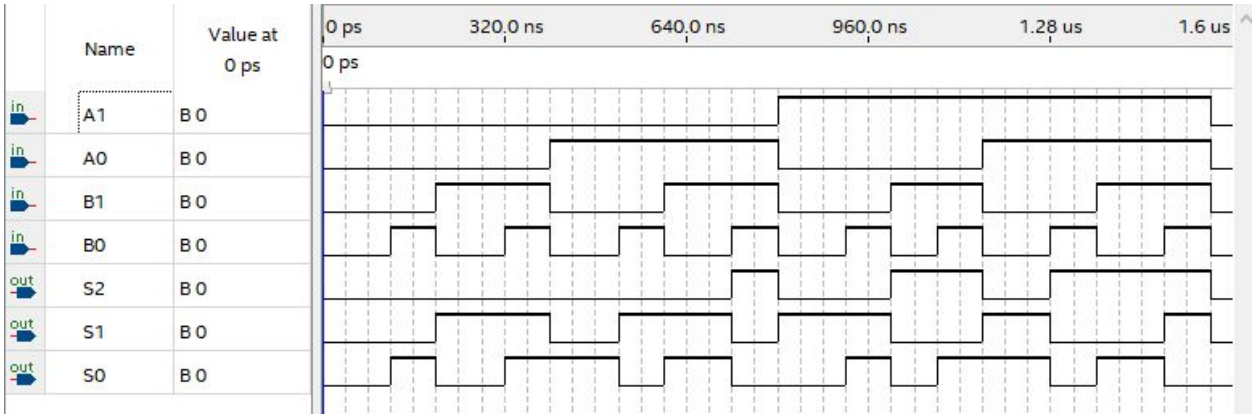
Verilog code for the addition functions:

```
module S20 (A0, B1, B0, f);
   input A0, B1, B0;
   output f;

   assign f = A0&B1&B0;
endmodule

module S21 (A0, B1, B0, f);
   input A0, B1, B0;
   output f;

   assign f = B1|(A0&B0);
endmodule

module S10 (A1, B1, f);
   input A1, B1;
   output f;

   assign f = (A1&~B1)|(~A1&B1);
endmodule

module S11 (A1, B1, B0, f);
   input A1, B1, B0;
   output f;

   assign f = (A1&~B1&~B0)|(~A1&~B1&B0)|(A1&B1&B0)|(~A1&B1&~B0);
endmodule

module S00 (A0, B0, f);
   input A0, B0;
   output f;

   assign f = (A0&~B0)|(~A0&B0);
endmodule

module Mux(x1, x2, s, f);
   input x1, x2, s;
   output f;

   assign f = (x1&~s)|(x2&s);
endmodule
```

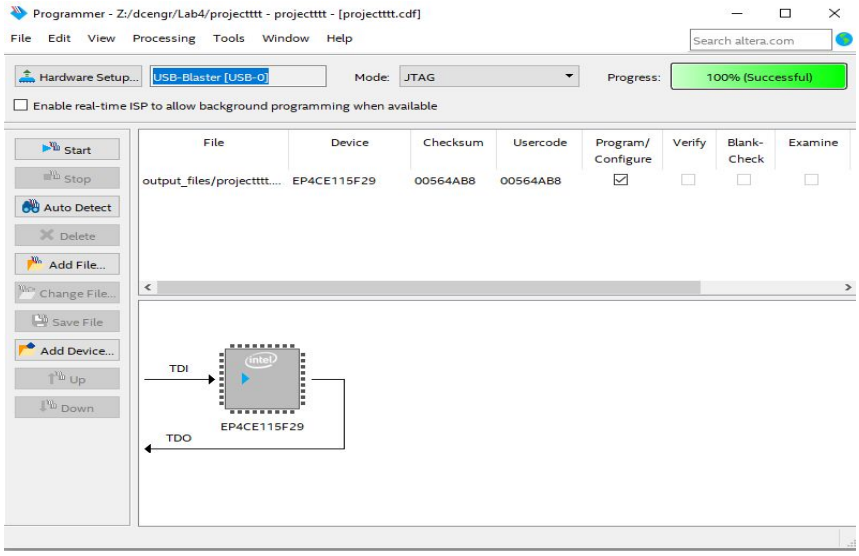
K-maps of the addition functions:



Waveform:



Proof of successful load onto FPGA:



Include the K-maps and the logic expressions for the functions you used. Explain how you chose these based on your prelab work.

See image of my notebook above for k-maps and functions that we used. We used the same set up that we used in the prelab because we had no particular reason to change it.

Explain the advantages and disadvantages of the different implementation methodologies, from your perspective. Consider both ease and efficiency of design and ease and efficiency of making modifications.

It helps us realize that we can do less complicated/easier ways to build a circuit. For example, one of us originally wanted to connect each wire, while all along he could have just labeled each wire (less time consuming and easier to do). Another example was removing the third mux. Because one of us didn't include a third mux, we realized that it wasn't necessary there were cancellations in the function equation. In general, it allows each individual to see other opinions of how to plan out a circuit, and most of the time it will shine light on a better way to do the lab.

If you found problems with the circuit that required correction and a new download, explain what you observed that demonstrated the problem and describe determined the cause of the problem and how you fixed it.
There weren't any major problems with the circuit. The only changes we needed to make were to the 7 segment display verilog code because there were some errors in the Verilog code. Also one of the segments on our 7 segment displays didn't work so we had to change the pin assignments to another display on the FPGA.

Write a conclusion about the challenges of this lab and what you learned in this lab.
Overall, from this lab we learned how implement verilog code of many different functions including a 2 to 1 mux and the functions that we created in the pre lab. The main challenge of the lab was k-maps because they were so big and complicated. However, it wasn't too big of a problem because we were both working together to find the correct logic.