ECE4250 Lab 6 Report

Sterling LaBarbera

14096504

# Objective

The objective of this lab was to gain experience using Vivado for transferring VHDL designs onto a real FPGA board. For this lab we created a 4-bit full adder that displayed on a 7-segment display, and used switches to represent the bits of each value.

# Lab Work

For this lab, seven files were provided, six VHDL code files and a .XDC file for use in the Vivado setup. We filled in missing segments of each code file, and after it compiled, we used Vivado to create a bit stream that would program the Nexus4 FPGA board in the lab.

## Implementation

To start, we completed a 4-bit full adder VHDL file for the basic computations in the same way as Lab 2, with a full adder component, port mapped to the 4 input bits and carries of the 4-bit full adder.

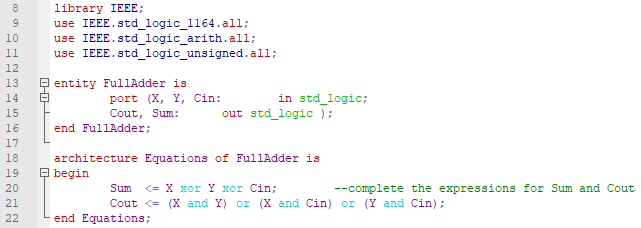


Figure . Full adder code.

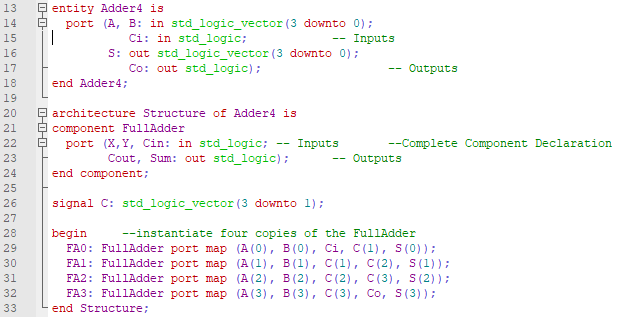


Figure . 4-bit full adder code

Three more Components were needed. One for controlling the LED display, which did hexadecimal to 7-segment logic and inverted the output since the displays were active low. A second for managing the overall LED ouputs since it is displaying the sum, though I did not fully understand how it was working. The third one was called AnodeControl, but that one I also did not really understand, but it seems to be updating the counter needed for the second of these three files. The images below are in the order just described.



Figure . LED display code.

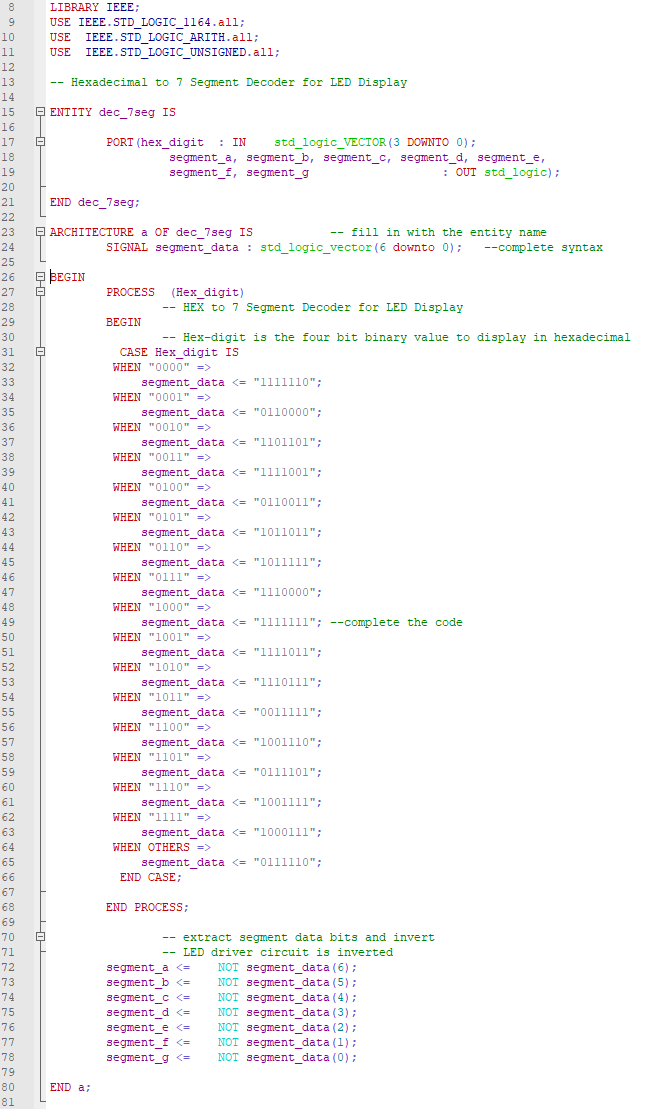


Figure . dec-7seg code.

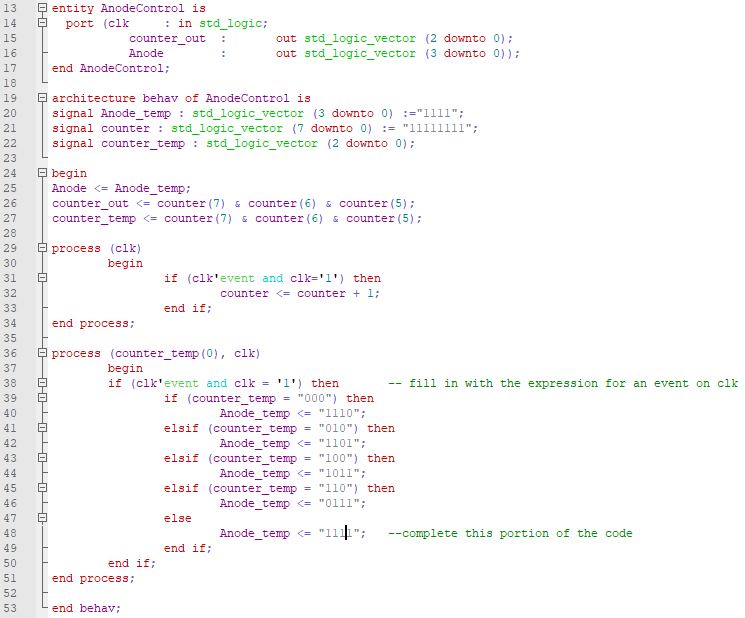


Figure . Anode control code

The final VHDL file contained all the components and managed the outputs to the LED displays using the switch as inputs, and the 4-bit adder to produce the output.

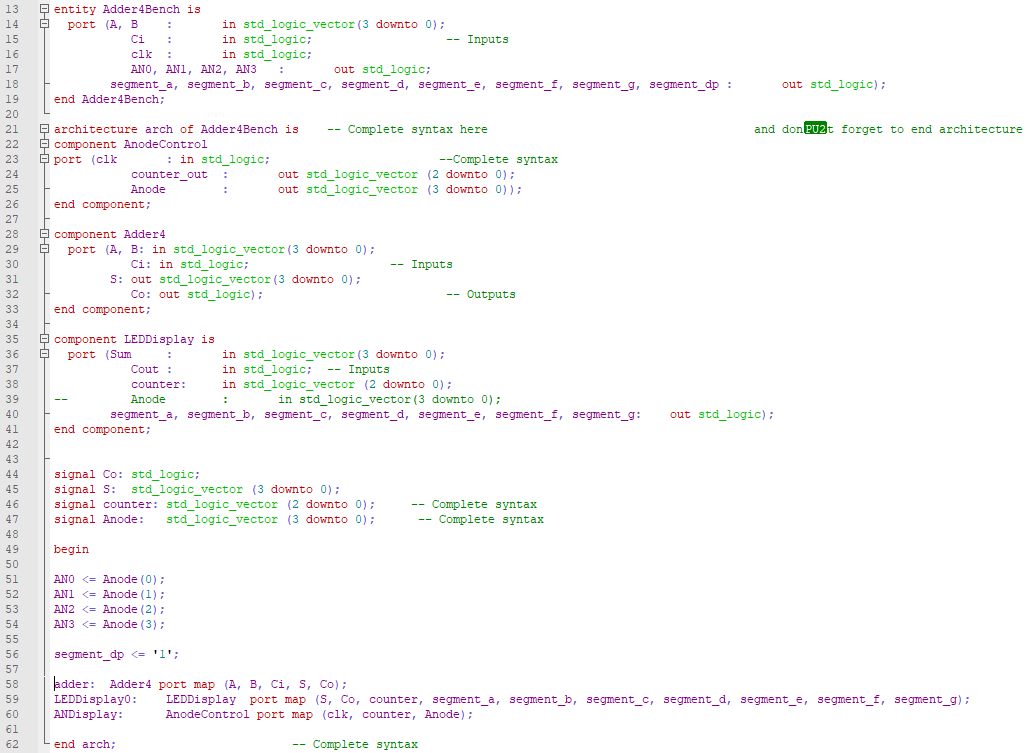


Figure . Adder4Bench code.

Using all these files, we loaded them into Vivado along with a special XDC file that contains constraints for the FPGA elements. In this case, it was set to use the switches for inputs, and the LED display for outputs, as well as tying a pin for the counter. Most of the properties are commented out as this is a generic file, and edited to suit the current program.

# Conclusion

I did not have any trouble with this lab since we mostly had to copy segments of code from other files and make sure the names matched up. the only other fill-ins followed patterns already set in the code. Below are some output example images.

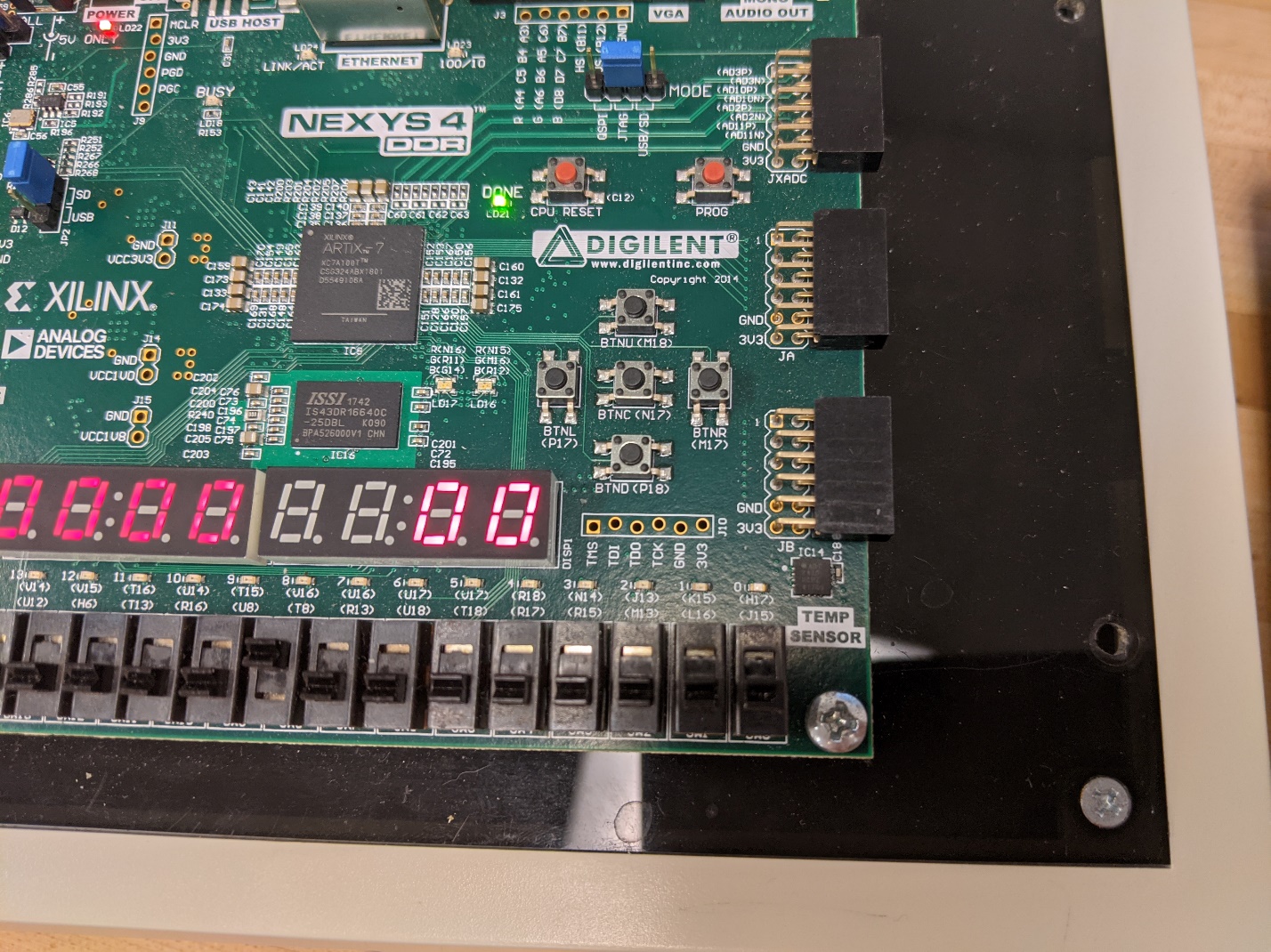


Figure . 0 + 0

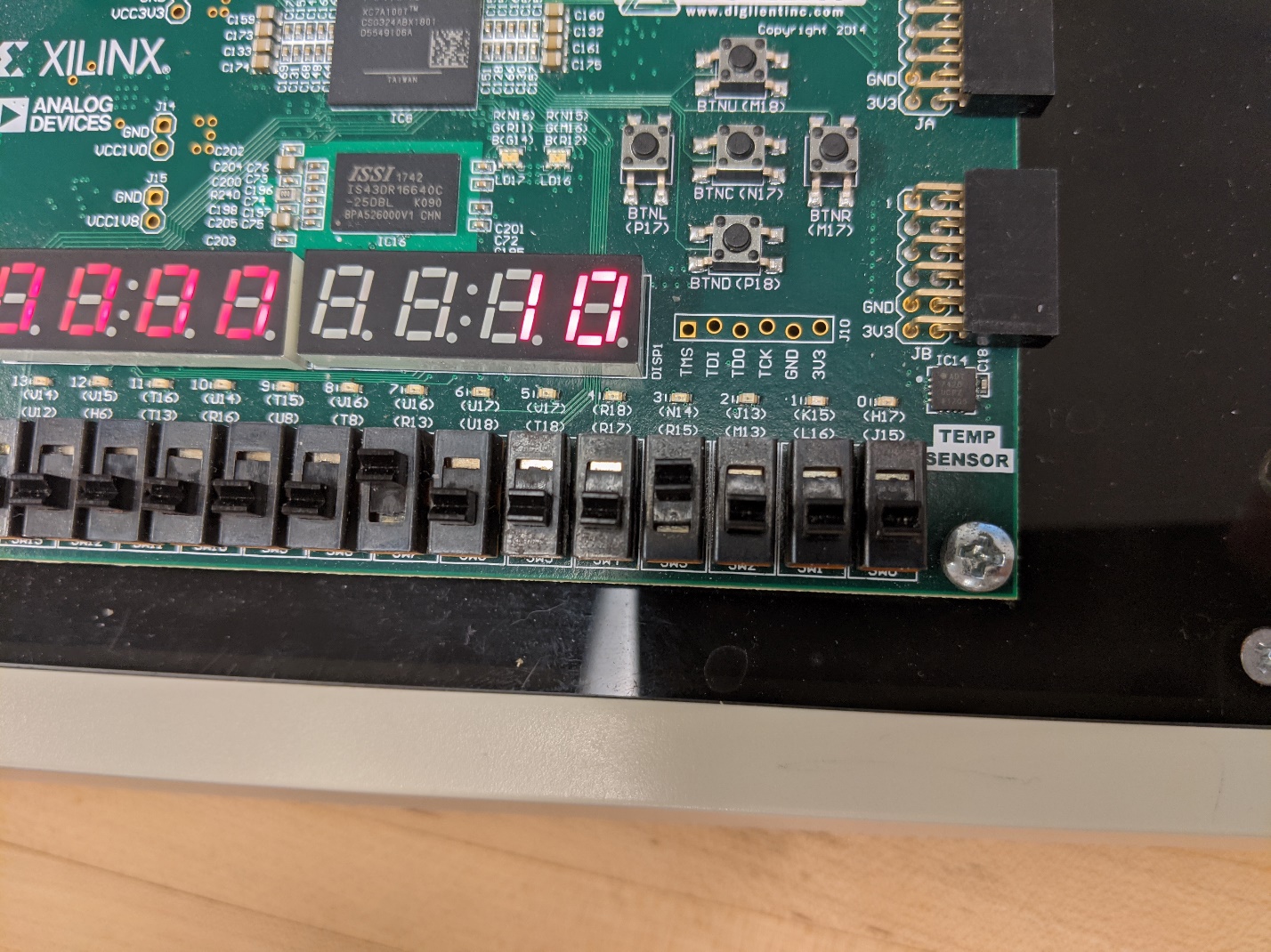


Figure . 8 + 8 = 0x10

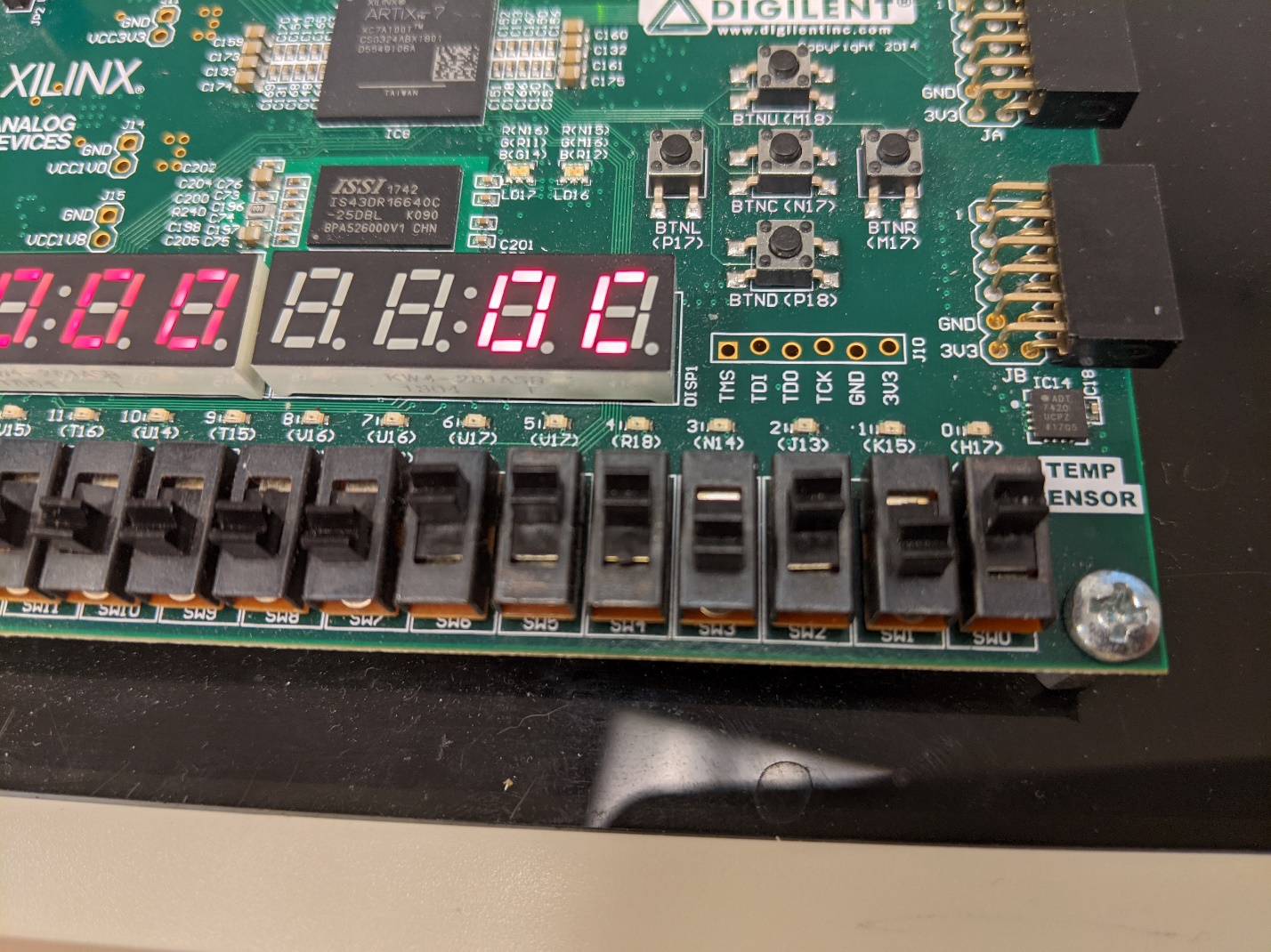


Figure . 7 + 5 = 0xC

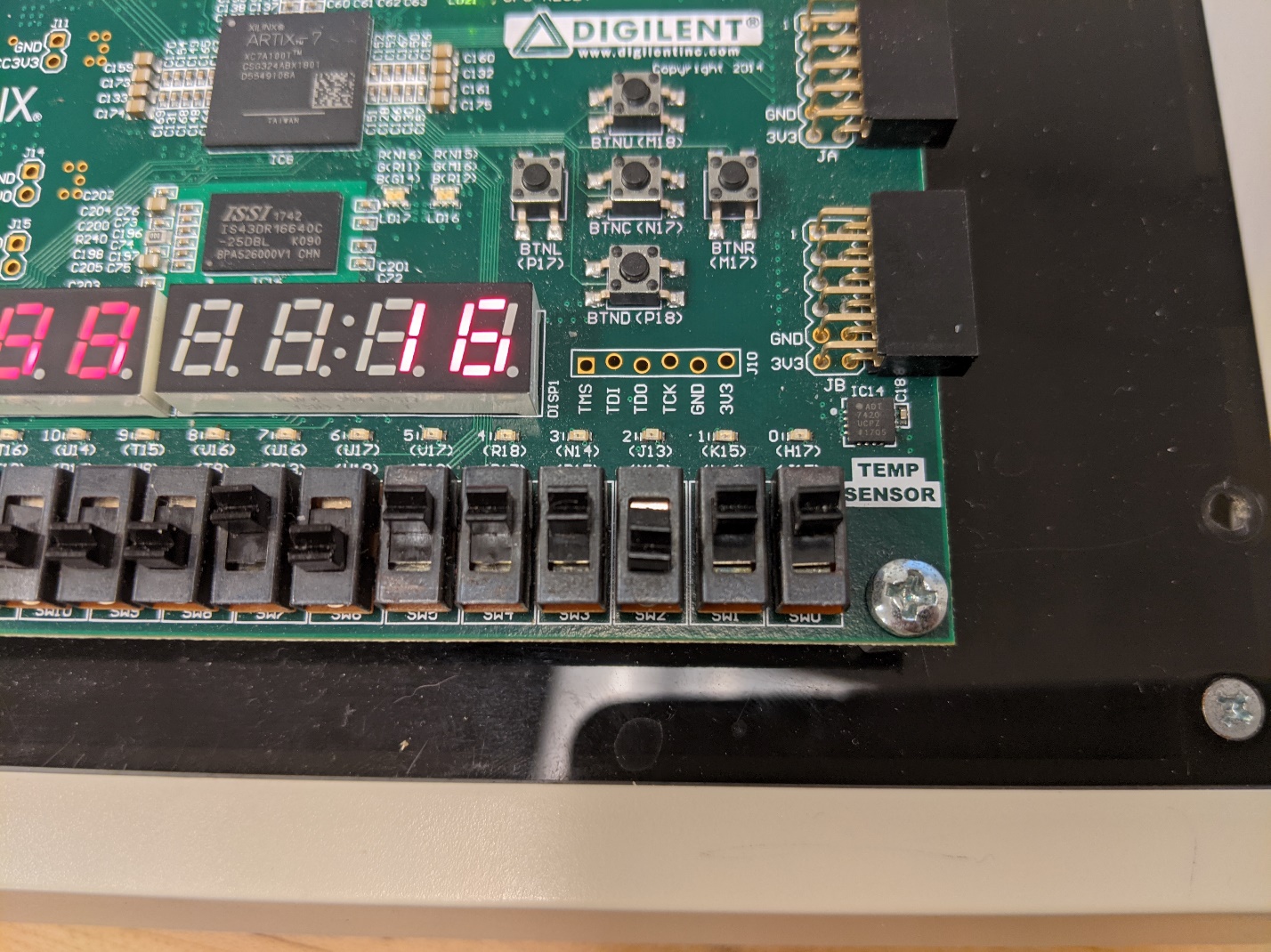


Figure . B + B = 0x16