Lab 7 - Taylor Cowley and Andrew Okazaki
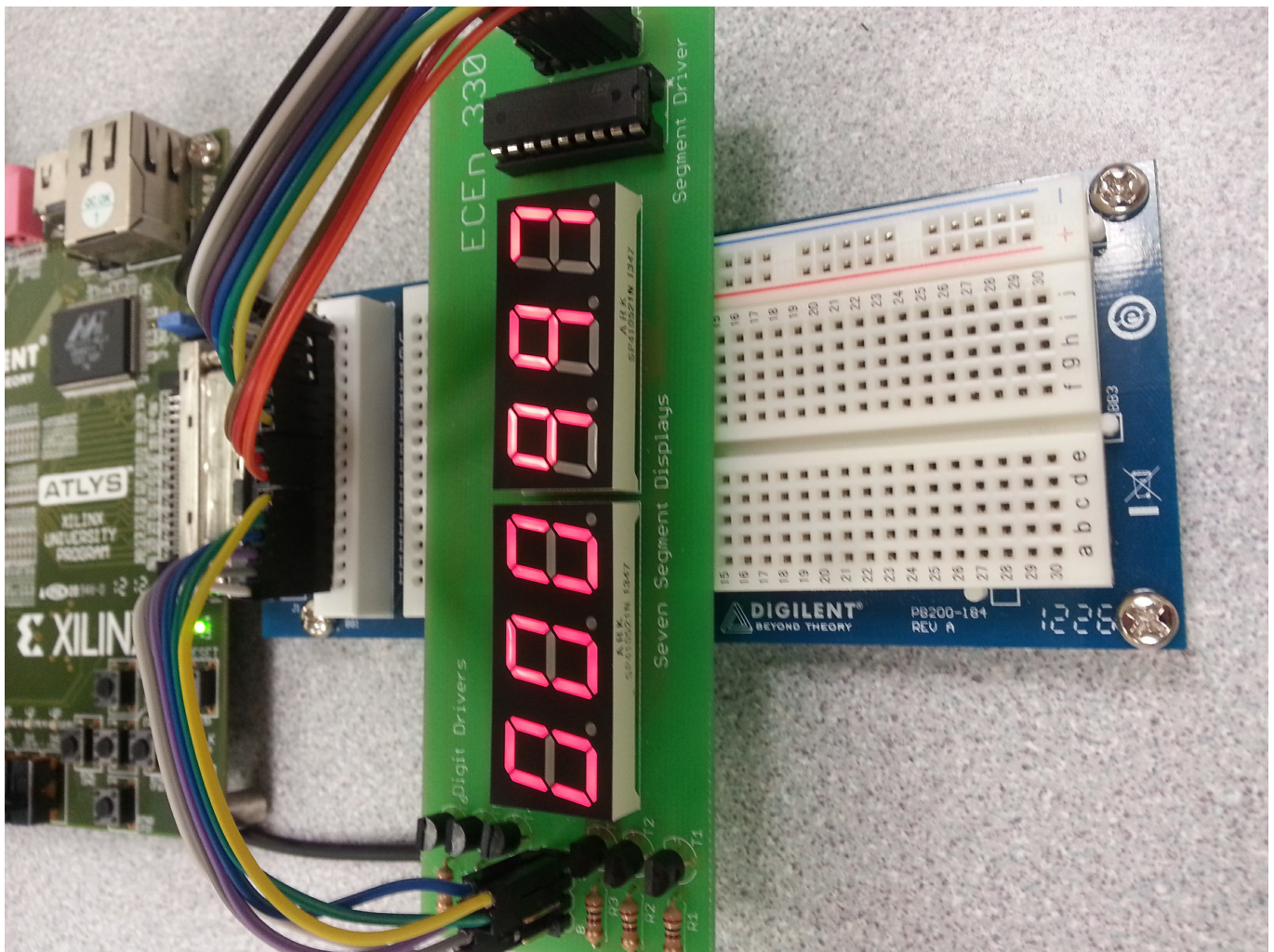
- Adding a Score Display

    - Hardware

    - VHDL

    - Software Interface

    - Bug Report

Adding a Score display

Hardware

An old 6 digit 7-segment LED display was found for former iterations of the ECEN 330 class. It had transistor drivers for the LEDs already on the board, which would make sending signals much easier

We made a custom IP in the Xilinx editor that had 8 registers- 6 to store the input for each digit, and 2 extra for debugging. This was a very similar process to making the PIT in the last lab

VHDL

VHDL code was added to make the proper 7 segment decoder- based on a certain input, the 7 outputs had to arrange to make a number. See Appendix for code. Also, a counter was added, and depending on the counter, a certain digit had its output sent. This is because the board had digit selectors and 7 segment selectors. It can display more than one digit at a time, but they will all display the same thing, depending on the 7 segment drivers. The solution to having different things displayed on each digit is to cycle faster than the human eye can distinguish through each of the 6 digits, outputting the proper data.

One problem we had was this counter was going too quickly, and the LEDs were never receiving enough power to output.

Some simulation was done to verify proper decoding by the VHDL
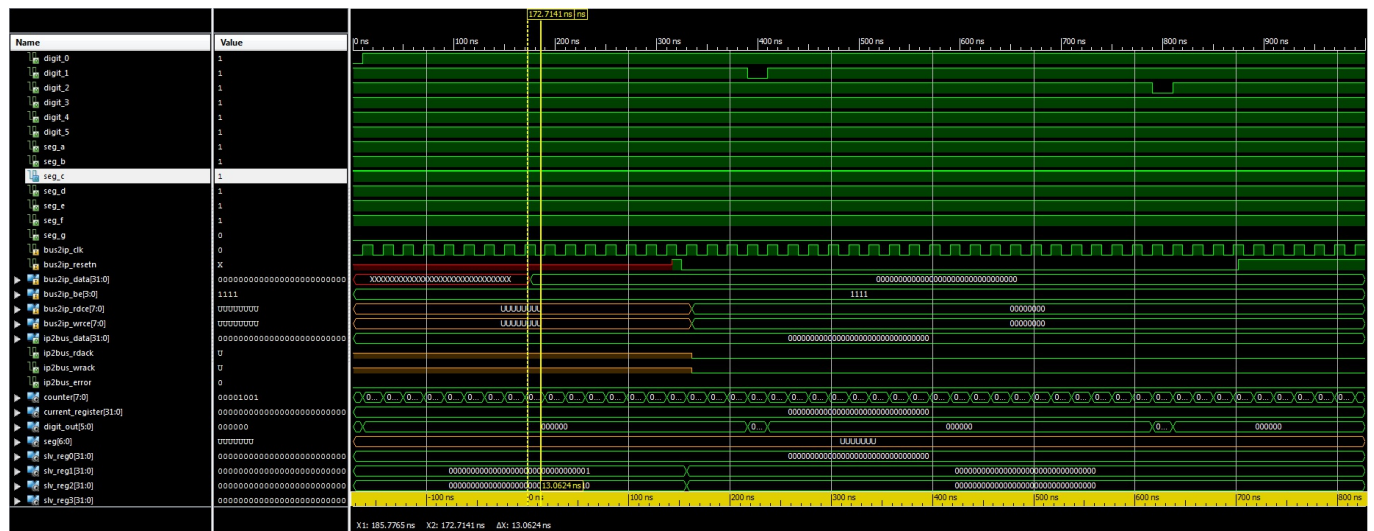
Software Interface

Because all logic for the display is handled in the VHDL, all the software had to do was write the value for each digit into a certain register. With the Xilinx automatically generating the software driver, all it took was

```
SCORE_DISPLAY_2_mWriteReg(XPAR_SCORE_DISPLAY_2_0_BASEADDR,
i*SCORE_DISPLAY_DIGIT_OFFSET, number);
```

to make a number display on the LED display.

Bug Report

In our lab we ran into both hardware errors and software error. In our code it took us a while to find the output pin location in software. The next error we ran into in software was the when outputting a digit then moving to output the next digit we were cycling too fast. The speed of cycling caused artifacts from other digits and did not allow the number

to fully register, in addition to making the display very dim. Once we slowed the timer down it fixed the error that we were experiencing. The hardware error that we ran into was missing connecting the wires. This took a while to debug because at first we thought it was in the software but finally looked at the hardware and found our error.