ECEN 427 Lab 6
Matt Heydorn
Zach Halvorsen

Lab 7: Creative Project

Section 7.1: Short Description of Project

We integrated a classic NES controller with Space Invaders. We created a communication driver for the controller in VHDL and then created a software driver to be able to use the VHDL in our C code.

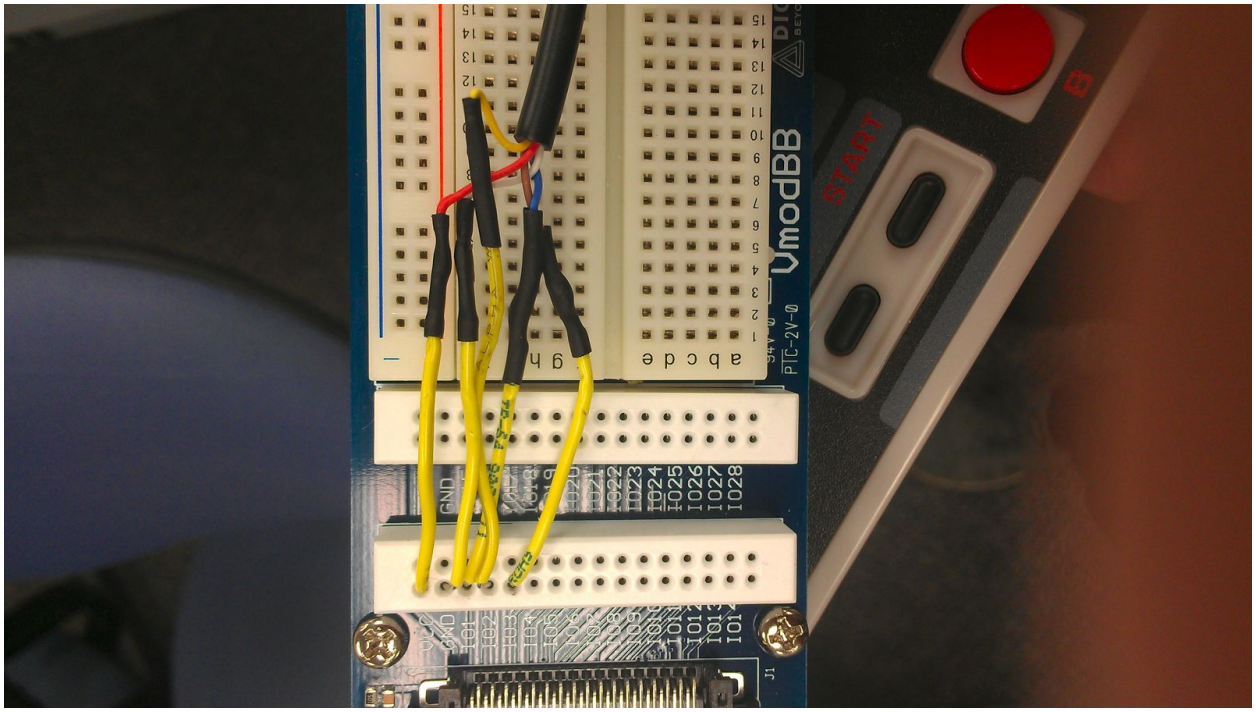
Section 7.1A: VHDL Communication Driver

The NES controller uses a UART similar form to tell what buttons are being pushed. It is a little more simple than a UART because there is no timing involved. The system driving the controller has two signals that control the output of the controller. It polls the buttons by sending a latch signal. This lets the controller know we want to read the value of the buttons so it gets it ready. We could then read the first button value after waiting a short period of time. The only constraint on the timing is that it has to be at least six microseconds. Then the VHDL sends another signal called pulse that lets the controller know we are ready for the value of the next button. This cycles through until all of the buttons have been read. We would send the latch at a frequency of 60 Hz and pulse every eight microseconds. We only sent an interrupt to our C code when a button changed a value.

Section 7.1B: Software Driver

Integrating the NES controller into our space invaders software was fairly easy. We made a driver API with a function that returns the button states. This was a simple read to the slave register in our IP. Also, our IP uses an interrupt that fires whenever the button state changes. In our handler for this interrupt we just check each button bit and mask it with out variable for buttons debounced. This way we did not have to create a new global variable for the NES controller button states. This was convenient because we didn't have to change any existing code, just add the new NES controler driver code and interrupt handler.

Section 4.2: Picture



From left to right the pins are VCC, Ground, Pulse, Latch, and Data

Bug Report

The biggest bug we had was that our cheap chinese NES controllers from Amazon has the same colored wires as a normal NES controller, but the wires all go to different places. We didn't realize this at first (and weren't expecting that either) so we had problems. It turns out that Red is vcc, white ground, blue pulse, yellow latch, and brown is data. Fortunately we had a second NES controller which we broke open to see which sockets on the D connector each wire went to.