Kostadin Pendev, Madeline Flynn

ECE222

5-10-18

**Final Project: Lab Report**

*Objective*

The objective of our final project was to create a controller for a braille digital watch under the brand name Brailleguet. Braille digital watches are not common yet as their analog counterparts are easier to produce and program. We will implement a controller for a braille digital watch using VHDL code and the Altera DE-112 Education Board. The Braille portion will be represented using the LCD screen and the many switches available will allow for tuning and control of the watch.

*Procedure*

Our code was split into four parts - the LCD code, the counter, a slow clock, and a overarching program to contain all of those components.

The LCD code requires 8-bit binary digits to represent the code for an ASCII character to be displayed on the LCD screen. Due to braille numbers being able to fit in a two by two grid, we could create individual signals for each dot in the grid for each digit of time. When the dots weren’t needed, the signal was set to the [space] character in ASCII. Then, for each numerical value of the braille number, the dots were replaced with the number for ease of reading. Once the signal for each dot was assigned, they were added in their respective place into the LCD output value. Between each braille number there was a space, and between the hours and minutes was an asterisk.

The slow clock code is based on the internal clock of 50MHz. The code takes the internal clock as an input and outputs a clock signal that is 1Hz. Thus, the output clock signal goes high every second. The internal clock is slowed down by counting the amount of times that the internal clock goes high. The code outputs a 1 only after the internal clock goes high 50 million times. For the demonstration, we made the slow clock 1 second, which controls the minutes. In order to make the minutes act like minutes we would need to modify our code to count to 3 billion before outputting a high signal.

The counter has choice, clk, tune, incMin, incHr as inputs and AM\_PM, minutes\_ones, minutes\_tens, hour\_ones, hour\_tens as outputs. The clk input is connected to the component second\_clk output and controls when the minutes are incremented. The tune mode is the mode where the user can increment the minutes by sending a high signal to incMin, can increment the hours by sending a high signal to incHr, and can choose normal time by sending a low signal to choice or military time by sending a high signal to choice. In the code we created variables for each digit and created a process where the minutes would increment only if the user is not in tune mode and if the clock is high. Additionally, we had rollover conditions and reset conditions for the hours and minutes. For example, if the minute ones reached 10, the ones would reset and the minute tens would increment. If the minute tens reached 6 and the minute ones reached 10, the hours ones would increment. Similarly, if hour ones reached 10, the hour ones would reset and the hour tens would increment. If the military time number components reached 23:59, the components would rollover to 00:00. Likewise, if the normal time reached 12:59, the components would rollover to 01:00. The AM\_PM indicator is based on the military time; thus, if military time is between 00:00 and 11:59 the output is AM and if it is between 12:00 and 23:59 the output is PM. At the end, the code converts all of the integers to std\_logic\_vectors and assigns them to outputs. Lastly, the code outputs either military or normal time based on the user’s choice.

The lop level entity, Watch, contained all of the above VHDL in the form of components and included the necessary inputs and outputs from the board. The inputs included a reset for the LCD screen, 12 hour or 24 hour choice, tune mode, clock, incrementing minutes, and incrementing hours. The outputs included peripherals for the LCD screen, the display for the LCD screen, and a signal indicating whether it was AM or PM. From this top level entity, we compiled all the code; planned the necessary switches, LEDs, and pins; and uploaded the code to the board.

*Results*

The final result was a functioning braille digital watch with a tuning function to change the hours and minutes. The LCD screen displayed the time in the format of H H : M M with the H’s and M’s replaced with their respective braille number. The watch counted up on a faster time scale for demonstration purposes, but counted correctly for time. To switch between 12 hour and 24 hour time or to tune the time to what you need, it is necessary to put it in tune mode using a switch. A green LED was used to indicate AM or PM on the side of the board.

*Conclusion*

In the end, we had a watch controller that represents time using braille with different formats and options on an LCD screen. We were able to create this watch with the VHDL that we learned and because of our experience with the FPGA board in lab. In order to fully complete the watch, the braille in the controller needs to be mapped to mechanical spheres that move up and down on the body of the watch so that the user is able to read what time it is by running their finger on the body of the watch.