DIGITAL STOP WATCH SYSTEM

Derek Battle, Andrew James, Saher Al-Khrayyef, Menglu Zhu

Electrical and Computer Engineering Department
School of Engineering and Computer Science
Oakland University, Rochester, MI
E-mails: debattle@oakland.edu, acjames@oakland.edu, smalkhra@oakland.edu, mzhu2@oakland.edu

Abstract

This stopwatch project is a software and hardware co-design. The time will be shown on the FPGA board and on the LCD. The system will be modeled in VHDL (Very-high-speed integrated circuit Hardware Description Language) and implemented on the Nexys4 FPGA board.

I. Introduction

The digital stopwatch we designed is a time-keeping device that is meant to measure the time elapsed from the start to end of any event. The stopwatch has several different functions including pause (which represents both start and stop), reset, write to the LCD, and is able to clear the hundredth of a second output.

We used the Nexys4 Artix-7 FPGA Board and HD44780 LCD screen. The board was used to implement our digital stopwatch, and LCD screen was used to display the counter time and the elapsed time. The computing language that we used to write the program is VHDL, which works well to program the FPGA we used. For the functions, we used two switches and two push buttons to allow for user interactions. The write to LCD switch will display the elapsed time on the LCD when activated. When deactivated the LCD will show the running time. When the pause switch is activated, the stopwatch will stop running. When the last function reset is activated, all digits change to zeros. We also added the hundredth second (HTSeconds) function that when activated by the push-button causes the rightmost digit on the 7 segments to display nothing.

This project will improve our teamwork, our skills in VHDL programming, and help us gain more experience working with an FPGA. The class lecture was very helpful with this project because it covered important topics such as multiplexers, decoders, counters, finite state machines and 7-segments displays all of which we needed for our project. We also went over a digital stopwatch example in class, which made this project easier to design. Therefore, the only thing we needed to research was the protocol to connect the FPGA to the LCD screen and how to write data from our code to the LCD.

II. Methodology

In this section, we will provide a background on the operation of our stopwatch. The operating principle of each component will be examined.

A. Clock Counter

The Nexys4 board has a 100 MHz internal clock. We needed a clock that would tick every 0.01 seconds so we had to slow down the internal clock by using a clock counter that generates a pulse at this interval. Then, the output of this counter will be connected to the enable of every counter in our data path system.

The fastest digit of our counter is the hundredths of a second, which will run for 100 Hz. When we compared the internal clock speed to what we need to the clock digits to display we found that the internal clock is 10000000 MHZ faster (1 MHZ= 1000000 HZ). So, the first thing we needed to do was slow down the internal clock. The solution was to use a clock counter count every 0.01s.

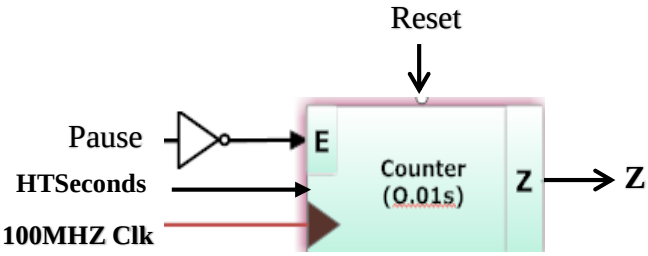

Figure 1 Clock Counter

B. Modulo 6 and Modulo 10 Counter

It was necessary to use both modulo 6 counter and modulo 10 counter in our project because each digit needs to be implemented by a counter (Figure 2). Our stopwatch uses six digits that are displayed on the LED 7-segment display. We used modulo 10 counter for hundredth seconds, tenth seconds, seconds and minutes because their values are only from 0 to 9. We used modulo 6 for the other values of time that range from 0 to 5. Therefore, the range of our stopwatch is from 00:00:00 to 59:59:99 or from 00:00:0 to 59:59:0.

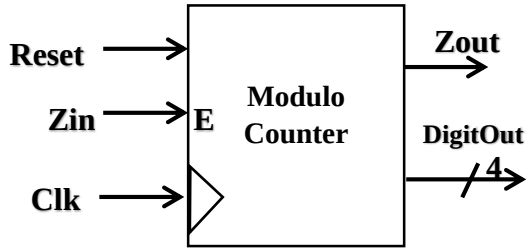

Figure 2 Modulo Conuter

C. Multiplexer

Using a multiplexer allowed us to use only one 7-segment decoder. If we did not use a multiplexer to choose outputs for us we would need to use six separate 7-segment decoders. The multiplexer helped us to simplify our code.

The following is the behavior of the multiplexer.

```
-- Multiplexor
with s select
   omux <= Q_0 when "000",
           Q_1 when "001",
           Q_2 when "010",
           Q_3 when "011",
           Q_4 when "100",
           Q_5 when others;
```

Multiplexer Behavior

D. Seven-Segment Decoder

Each modulo counter will provide the result of its digit in binary format. These digits will go through a multiplexer that selects which digit will be displayed. For this reason, we used a 4 to 7 decoder that will decode each four-bit digit into a seven-bit display.

The following is the 7-segment behavior.

```
with bcd select
   leds <= "1111110" when "0000",
           "0110000" when "0001",
           "1101101" when "0010",
           "1111001" when "0011",
           "0110011" when "0100",
           "1011011" when "0101",
           "1011111" when "0110",
           "1110000" when "0111",
           "1111111" when "1000",
           "1111011" when "1001",
           "-------" when others;
EN <= "01111111";
sevseg <= not(leds);
```

7-Segement Decoder Behavior

E. The Finite State Machine

We used the finite state machine to control the output that would be displayed on the LEDs. The state machine diagram shown below in the design section.

III. Experimental Setup

In our introduction, we mentioned that we used the Nexys4 Artix-7 FPGA Board and LCD_HC44780 screen in order to implement our project design.

1. Design

Figure 3 shows the board that we used to implement our digital stopwatch.

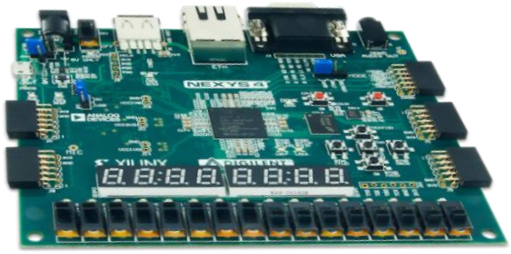

Figure 3 Nexys4 Artix-7 FPGA Board

The FPGA has a 100 MHz internal clock, which drives the entire system. Also, it has a seven-segment display that will be used for displaying the digits of time. We will also use two push buttons and two switches to allow for user interaction.

Figure 4 represents the structural data path of our stopwatch. It has six stages with four total inputs to the system. The internal clock is connected to 100MHz. The hundredth second removal and reset were mapped to push buttons, while pause and write to the LCD were mapped to switches.

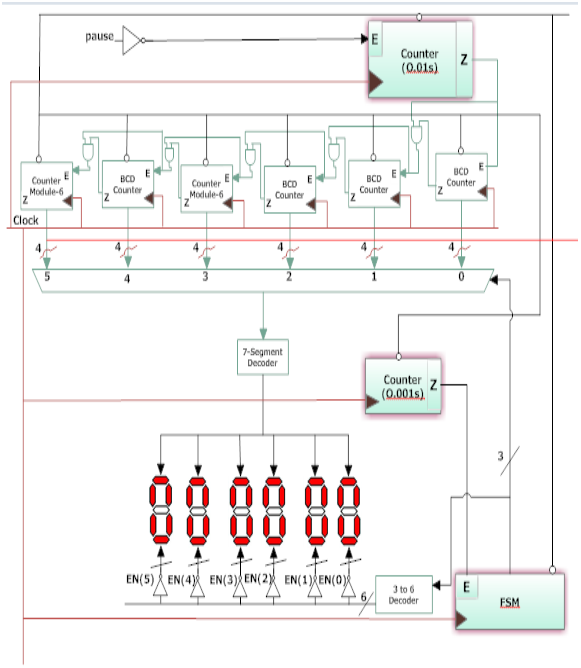

Figure 4 Data Path Structural

The output (Z) is connected to the enable of the hundredth second counters. We used an AND gate to connect the output (Z) with the output from previous counter and the output from the AND gate will be input to the next counter. The same idea was used with all counters. Also, each stage provides one digit in 4-bit format, so we needed a seven-segment decoder to decode the value into display format. Then, the output from the decoder can be implemented to their segment on the board.

A finite state machine was used to control which output would be displayed. Figure 5 shows the structural state machine. It is a Mealy Machine because the output depends on the current state and on the input.

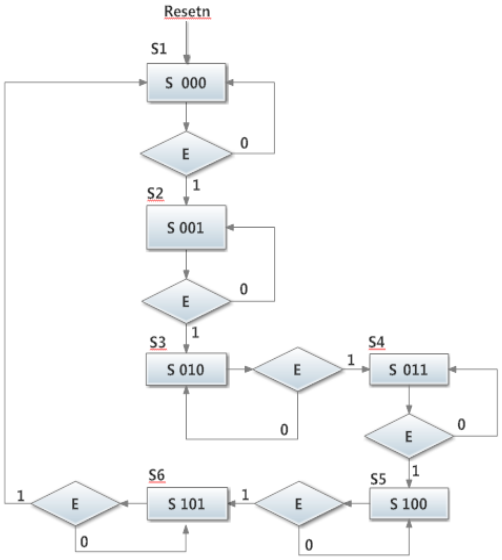

Figure 5 Finite State Machine Structural

2. Procedure and Implementations

We started the project by testing the code that we were provided to confirm that it works properly. The given code worked with four of the seven-segment displays. We modified the code to work on six seven-segment displays to allow the stopwatch to count to a higher time. Our modifications worked correctly with no issues.

The next step was to connect the LCD with the board by developing a protocol to allow the two components to interact with each other. We connected the two components using wires fed to a breadboard and back to the LCD using 16 bins.

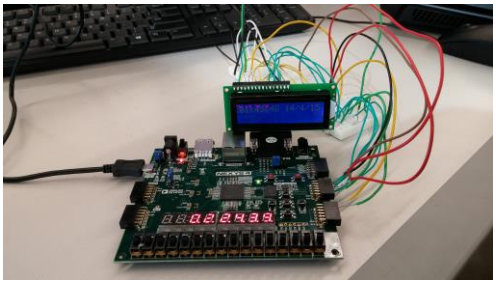

Figure 6 LCD HD44780 connected to FPGA

Our goal was to display the running time and the elapsed time on the LCD screen. The LCD that we used is shown in (Figure 7). We started out

writing to the LCD by testing if the word "HELLO" could be written to the LCD. Once we were able to show this word on the LCD screen, we were ready to move forward with our design. The next step was to show the elapsed time on the LCD screen. We modified the code to display this time on the LCD along with the date. To control this write function we decided to use a switch. When the switch is enabled the elapsed time will show on the LCD screen. Since we got this successfully working we began working on showing the running time on the LCD.

Figure 7 LCD HD44780

The following figure 8 is the structural block diagram for the LCD.

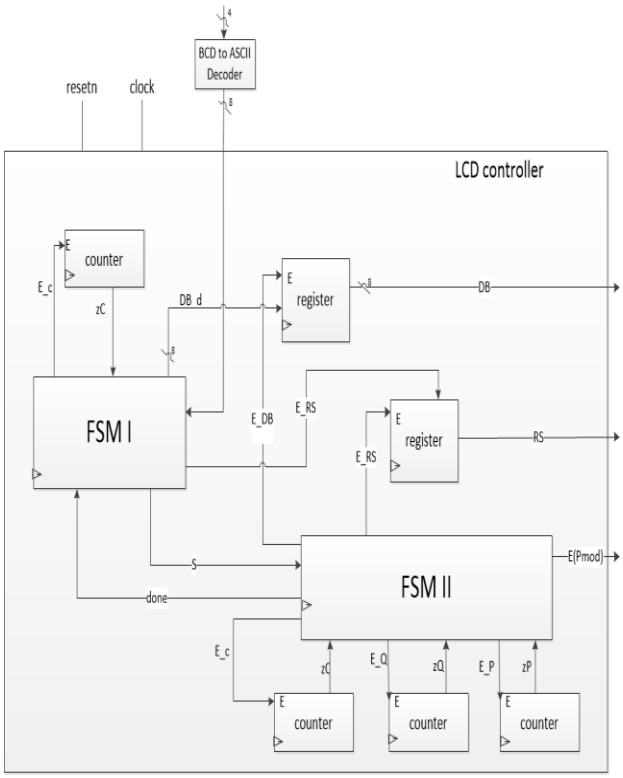

Figure 8 LCD Structural Block Diagram

We needed two state machines to implement our design successfully. Finite state machine 1 is used to set the data we will write in states 1-6. States 7- 25 are used to write this data to the LCD. See figure 9 below.

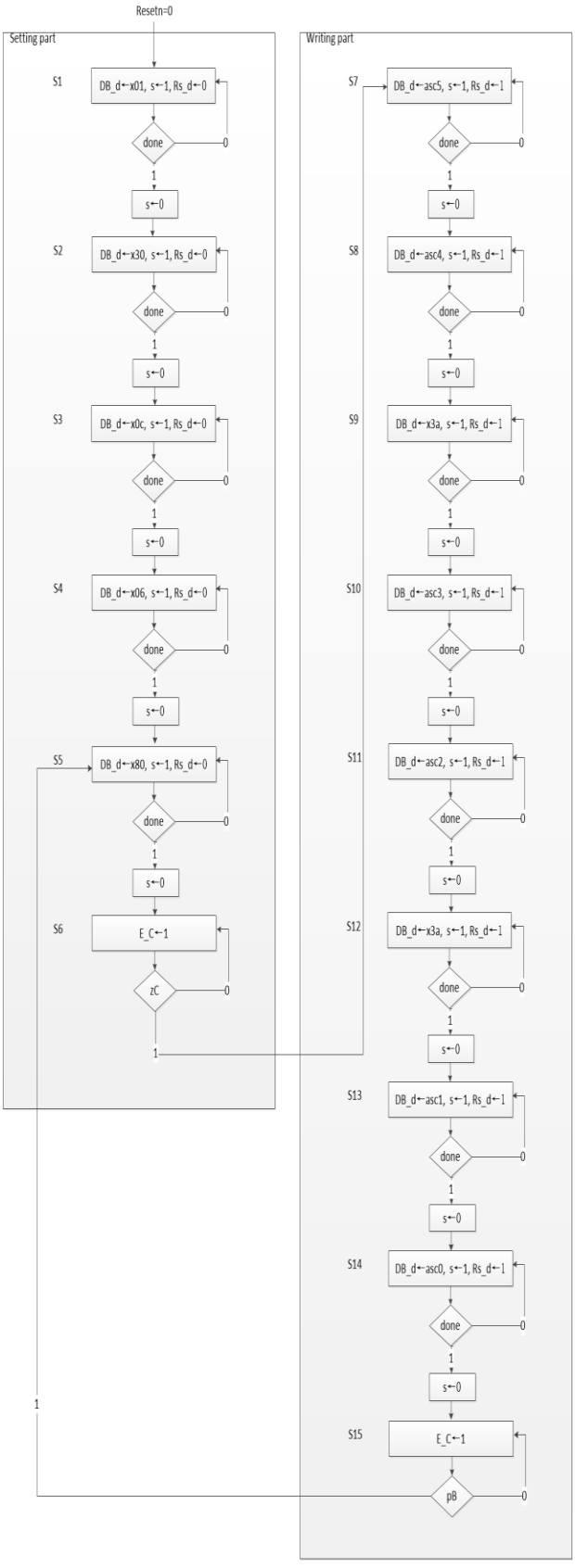

Figure 9 State Machine 1

Finite state machine 2 was designed to control the enable of the two registers rS and rW which set the data and then writes it to the LCD. See figure 10 below.

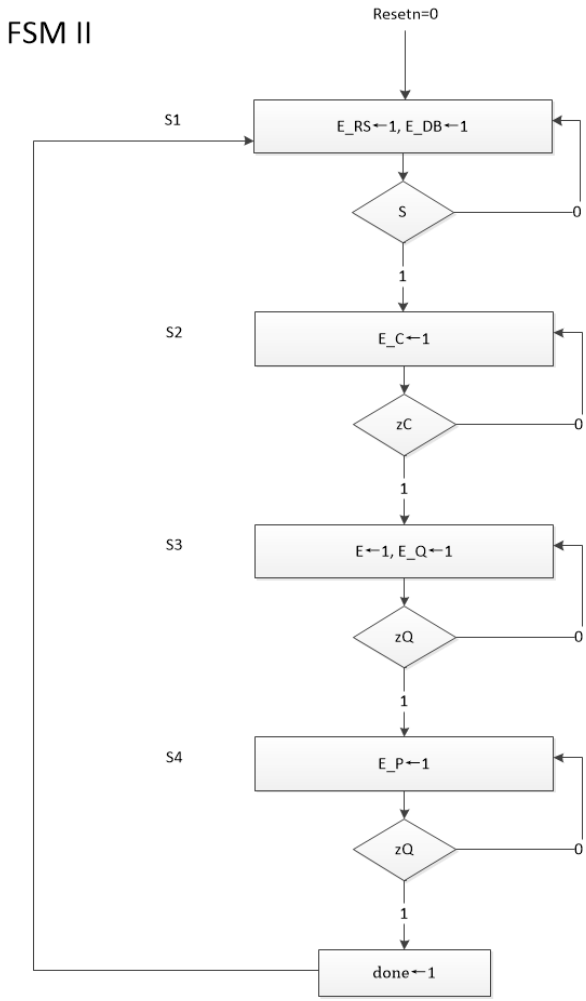

Figure 10 State Machine 2

IV. Results

We were able to accomplish the goals that we set for ourselves in this project. Our code worked properly and was able to simulate a digital stopwatch. Also, the added functions that we added for user interaction all worked how we initially hoped. Each step along the way we tested our code to make sure everything was properly mapped.

The pause and reset functions were implemented using a finite state machine. We needed a state machine because we wanted to change state only when the buttons and switches were pressed. The finite state machine we implemented has three states: idle (S1), run (S2) and pause (S3). The state transition is triggered when the button or switch is pressed. Essentially, it is a Mealy Machine implementation, since the next state is determined by the current state as well as the input.

Conclusion

In this project, we gained a lot of knowledge and experience working with a FPGA board. We learned how to connect a LCD screen to an FPGA board and develop protocol for the two to interact. Furthermore, we learned more about the fundamental principles of designing digital systems. Our group members have all improved in several areas such as building a structural data path for a digital system and understanding more about state machine. Our project was a great real-life experience in building a software/hardware co-design and our stopwatch could be used regularly in sports, experiments, etc.

References

1- VHDL Coding Tutorial- Daniel Llamocca http://www.secs.oakland.edu/~llamocca/VHDLforFPGAs.html
2- Intro to Digital Design- Darrin M. Hanna http://www.digilentinc.com/data/textbooks/intro_digital_design-digilent-vhdl_online.pdf
3- An Introduction to Software and Hardware Interfacing 2nd Edition- Han-Way Huang