Digital Stopwatch/Timer

ECE 2700 Final Project Report

Everardo Mejia, Sara Jamu, Tom Tooma, Hansen Shamoon
Electrical and Computer Engineering Department
School of Engineering and Computer Science
Oakland University, Rochester, MI
E-mails: emejia@oakland.edu, sjamu@oakland.edu, ttooma@oakland.edu, hansenshamoon@oakland.edu

Abstract— **The purpose of this project was to create a stopwatch that tracks laps and time intervals according to the user's preferences. Today, most people have a stopwatch feature installed onto their personal devices, mobiles, computers, etc. Through this project, we hope to successfully program a user-friendly stopwatch that matches the functionality of stopwatches we use daily. The system of this stopwatch will be built in a coding software called Vivado, using VHDL language.**

I. Introduction

With the use of a Nexys A100T board, this project will be using 6 seven-segment displays as a digital stopwatch with the use of a switch to act as start and pause, two buttons that will reset, lap, and 4 switches to allow the user to browse between the logged lap data. Enabling J15, the designated start/stop switch, allows the stopwatch to begin counting with an accuracy of a hundredth of a second. With the use of the BTNC button, the user will be able to use the lap feature that is included in most digital stopwatches and logs the data into pre-existing memory slots. Additionally, switches J11 - J14 will be used for the user to review the logged lap data. Furthermore, while displaying logged lap data the program will occupy two more seven-segment displays to show the user which lap the stopwatch is displaying (e.g. "L2 ##:##:##"). Finally, the BTNC button will be used as a reset button and will clear all data including the logged lap data. The CPU Reset button will restore the time depicted in the seven-segments to zero. For this particular Nexys board, by utilizing 2-bits, the system will be able to provide for four different lap times. Four lap times are efficient in displaying the abilities of the stopwatch. Some of the skills that this project will need include coding methods that were taught in class and a good understanding of the Nexys board. Some of the main components that will be programmed into the Nexys board include an FSM, 6 counters, seven-segment displays, and a few multiplexores.

II. Methodology

For our system layout, we decided to utilize multiple counters, multiplexores, decoders, seven-segment displays, registers, encoders, and finite state machines. We decided that the FSM and registers will act as random access memory for the stopwatch with the ability to store the different lap times recorded. As mandatory, the utilization of a 7-segment decoder is necessary for the displays to work properly. By connecting the acting memory to the multiplexor, the user can toggle between whether they want to display the recorded lap times or the stopwatch time using the switches. As for the counters, they will be utilized in having the stopwatch time increment. They are essential for the FSM as a component. The design will have six different states, one state for each display. Reset will simply reset the entire stopwatch. All displays will start at zero, and by toggling a switch, the stopwatch will start incrementing hundredths of a second upwards rapidly.

A. Board Layout

For the board layout, we decided to utilize switches 14 through 11 (SW14-11) to determine each lap time, as there are four different switches. The power switch turns on the Artix A7 board, and the CPU Reset is used to reset the board or stopwatch. We also decided to utilize the center button (BTNC) in order to record the laps. The display is used to display either the laps or the stopwatch time. According to Figure 1, switch 14 corresponds to lap 1, switch 13 corresponds to lap 2, switch 12 corresponds to lap 3, and switch 11 corresponds to lap 4. Switch 15 (SW15) plays two key roles as a switch, it can be used to either start or stop the stopwatch. When switch 15 is toggled on, the stopwatch stops and when it is toggled off, it continues to tick.

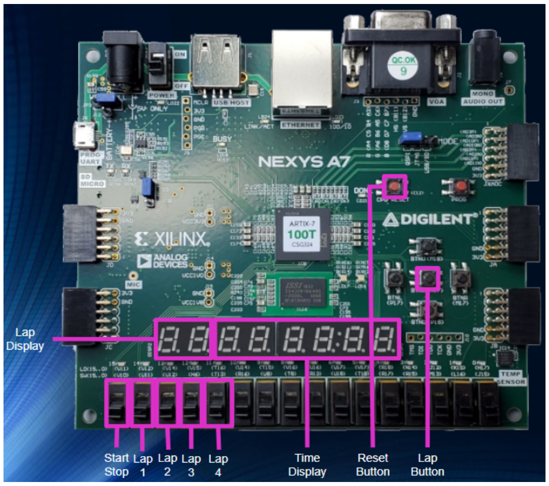

Figure 1: Board Layout

B. Wiring Diagram

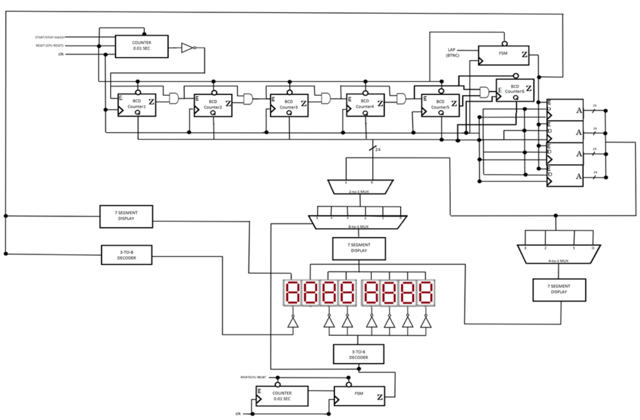

Figure 2: Wiring Diagram

For the wiring diagram, in order to get the stopwatch running, we utilized multiple components and wired them together. For the timer, we utilized BCD counters, FSM counters, registers, finite state machines, multiplexores, encoders, decoders, and the seven-segment displays. The counters are wired to the registers that record lap time, which get wired to a multiplexor to act as a toggle between the stopwatch time and the lap times. As shown, there are four registers, one for each lap time. The wiring diagram depicts all necessary wiring for each particular system and how they work together to allow the stopwatch to properly function.

C. Counters

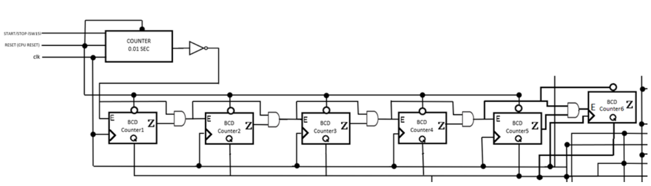

Figure 3: Counters

Our stopwatch for our circuit diagram utilizes six counters. Five BCD counters and one incrementing counter that feeds into the other counters. These counters all reflect into the display for the stopwatch, with the maximum time allowed on the display being 59:99.99. As for the registers, the counter system is also connected to them for the lap times. In connection, the lap times get recorded by using the center button.

D. Seven-Segment Displays

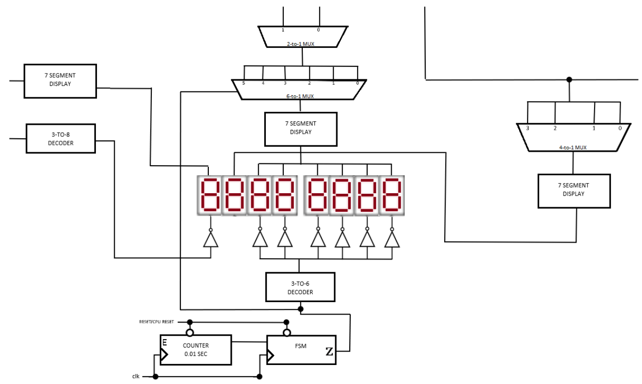

Figure 4: 7-Segment Displays

The A7-100T board contains eight seven segment displays, with them being conjoined in fours. For our stopwatch, we utilized the last two in the first portion of the four seven-segments, and all of the last portion. Thus, having us use six seven-segment displays for our project. We initially planned on utilizing all eight, but had trouble incorporating the first two into the stopwatch system. Figure 4 shows the wiring diagram for the seven-segment displays, and what they are connected to.

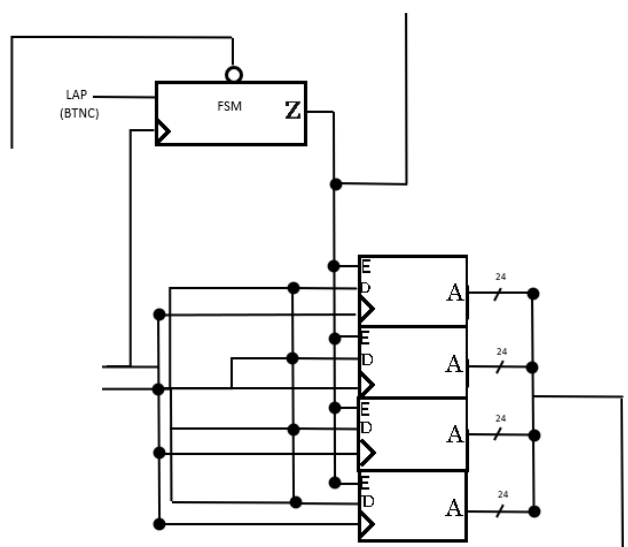

Figure 5: Lap Registers

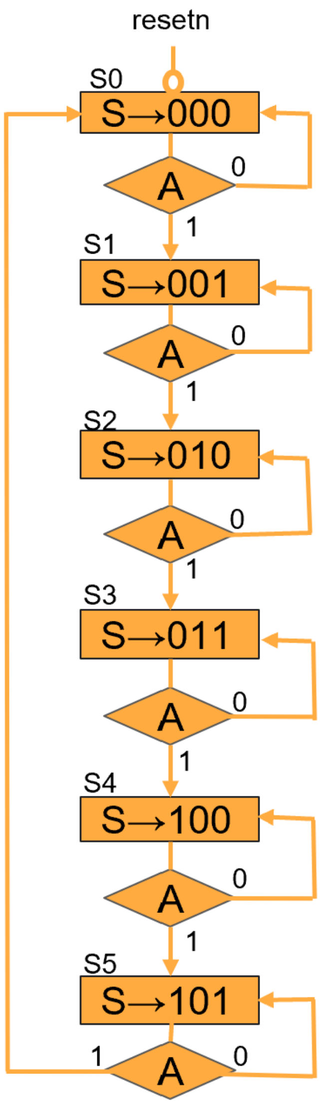

Figure 6: ASM of 7-segment State Machine

Figure 5 depicts the four registers, one for each lap time. Figure 6 depicts the states for the state machine that is used in connection to the seven-segment displays. Because there are six seven-segment displays being utilized, the finite state machine has six different states. Each state controls a seven-segment display. In connection with the counter, the finite state machine switches between states. The ASM displayed in Figure 6 goes to the next state as the timer increases and begins to require another digit. The four registers, as mentioned before, are the acting memory the circuit utilizes for recording each lap time. These registers are essential to being able to keep stopwatch values recorded for each lap.

III. Experimental Setup

For the set up of this project, we used the following equipment:

- Vivado 2021.1 Software
- Nexys A100T Board
- 2 buttons
    - One button to reset the stopwatch/timer
    - One button to lap the stopwatch time
- 5 switches (acts as start and stop)
    - One switch for start and stop
    - four switches for the user to browse through selected laps

Using the Vivado Software, our group will be creating the following design, simulation, and constraint files:

1. Design
    - TopFile
    - Lap Top File
    - Lap State FSM
    - Lap Encoder
    - FSM
    - 3 to 8 Decoder
    - Counter Top File, along with:
        - Counter for the 100th decimal
        - Counter 9
        - Counter 6
        - Counter 7 Segment
    - Seven Segment Decoder
    - 32 bit Registers
    - MUX
        - 8 to 1 MUX
        - 2 TO 1 MUX
        - 4 TO 1 MUX for Lap

With these files, the group utilizes the following steps to ensure the code is fully functional and produces the expected output using the Vivado Software:

1. Run Synthesis

Synthesis would show if there were any coding or syntax errors

1. Run Behavioral Function Simulation

Based on test values set in the testbench files, the group can confirm whether the resulting values match with the groups expected values

1. Run Implementation
2. Generated Bitstream file

Ensures that .xdc/constraints file is properly designed

1. Tested code on Nexys FPGA Board

Stopwatch runs as expected using the start/stop switch, reset and lap buttons, and lap data display switches.

IV. Results

The overall performance of the digital stopwatch went successfully. With the seven-segment displays showing an accuracy of a hundredth of a second and a max time of 59 minutes and 59.99 seconds. The use of the reset button properly reset the stopwatch whether the stopwatch was on or off and reset all lap memory data. In addition, the lap memory switches, that allow the user to toggle between lap 1 through 4, was able to display the data that was recorded at the times in which the lap button was pushed and successfully stores timing data until all four lap memories are filled. The Lap display feature that the group had hoped to integrate into the project was unsuccessful. Due to the creation of a Seven-segment display and 3 to 2 mux, it made it difficult for the group to designate the last two seven-segment displays to properly display the Lap number in addition to the lap data when it is that the user is viewing the lap data (e.g. “L# ##:##.##). When our group attempted to run the stopwatch after implementing the additional Lap Number Label system, there were major errors. These errors included all leds for the seven-segment display to light up and no change in values to occur, all seven-segment displays displayed a backwards “F”, and prevented the user from viewing the lap data. Because of these issue, our group decided to exclude this additional feature from the project and instead replace the “L#” display on the seven-segment display with leds above each switch to light up whenever each respective lap data (1 through 4) was displayed on the seven-segment display. Despite this falling short of our group's expectations, the digital stopwatch operated as necessary and provided the lap data as initially desired.

V. Conclusions

Designing the stopwatch system composed of a counter system, a lap system, and few buttons for other features. Even though programming a stopwatch sounds straightforward, there were many challenges we faced throughout the process, especially when creating the lap system and trying best to eliminate crossovers with the counter system. There was one extra feature we wanted to have, but did not have the time to fully configure, which was displaying a lap number (ex: L1, L2, L3, . . . etc) for when the user is browsing through selected laps. All in all, we were able to successfully debug all our issues as a team and programmed the stopwatch into the FPGA board so that it displayed all required features. To conclude, the stopwatch we created was user-friendly and had all the functions of any other stopwatch we may find on an everyday basis.

References

[1] D. Llamocca, *Reconfigurable Computing Research Laboratory*. [Online]. Available: http://www.secs.oakland.edu/~llamocca/index.html. [Accessed: 14-Apr-2022].

[2] Xilinx, “Vivado ML Overview,” *Xilinx*. [Online]. Available: https://www.xilinx.com/products/design-tools/vivado.html. [Accessed: 14-Apr-2022].