Digital Stopwatch

Final Report

Designed by Colin Steed, Eduardo Garcia, David Dzierzawski, Muhammad Imtiaz
Electrical and Computer Engineering Department
School of Engineering and Computer Science
Oakland University, Rochester, MI
e-mails: csteed@oakland.edu, eduardogarcia@oakland.edu, ddzierzawski@oakland.edu, mimtiaz@oakland.edu

***Abstract* — The purpose of this project was to design a digital system that performs a meaningful function. For our project, we designed and created a functioning digital stopwatch with a simple user interface. This project was designed and built in Vivado using VHDL and implemented on a Programmable Logic Board. The components of this final project include several essential VHDL coding techniques. The project could be enhanced by increasing the number lap times capable of being saved or creating a way to display the selected lap time while continuing to display the current timer. However, the current system is an easy-to-use stopwatch for everyday purposes.**

I. INTRODUCTION

As stated previously, the purpose of this project was to design a digital system that performs a meaningful function, in our case, a digital stopwatch system. Our system was designed and built in Vivado, a software suite produced by Xilinx for synthesis and analysis of HDL designs. Specifically, our system utilizes the VHDL programming language and was implemented on a Nexys A7 Programmable Board.

Our goal was to design and create a simple, easy-to-use stopwatch for everyday purposes. Our stopwatch has the essential functions of a standard stopwatch. These functions include Start, Stop, Reset, and Lap. The user can start and stop the watch at any time with the start/stop switch. When the user toggles the lap button, they can record up to four lap times. Each lap time can be displayed when its assigned switch is tripped by the user. The stopwatch uses two four-digit common anode seven-segment LED displays that are configured to behave like one, single eight-digit display. The system displays hours, minutes, seconds, a hundredth of a second, and a tenth of a second.

The components of this final project include several essential VHDL coding techniques we learned in class. These include, but not limited to: BCD Counters, RAM, FSM, Encoders, Decoders, Multiplexors, and Seven Segment Decoders.

Whether it's for sports training, vehicle testing, or cooking, stopwatches prove to be very useful tools for many applications. Most smart devices, such as phones, wrist watches, and PCs have some form of a simple stopwatch. Our goal was to design and program a stopwatch that matches the capabilities of these stopwatches we carry with us every day.

II. METHODOLOGY

A. Designing & Building

The idea behind the stopwatch was relatively simple. The Nexys A7 has an extremely useful hardware layout, making it easy to utilize for our needs. Also, majority of the components we used we had learned previously in class and in the lab. This made planning and designing the stopwatch relatively unpretentious. However, the real challenge came when we began building the components and the system in Vivado. Here, we ran into many complications building and debugging the system. However, even with these setbacks, we were able to design and create a fully functioning stopwatch.

B. Board Layout & Controls

The Nexys A7 programmable logic board has Seven programable pushbuttons and sixteen programable switches for human-machine interfacing. Our Stopwatch utilizes two pushbuttons and five switches for the stopwatch controls.

The power on switch turns the board on or off depending on the position. Switch #15 (SW15) is utilized for the Start/Stop feature. Switch #0 - Switch #3 (SW0-SW3) are utilized to display lap times. When switch #0 is tripped, it will display Lap #1 time. When switch #1 is tripped, it will display Lap #2 time, and so on. When no lap switch is tripped, the system will display the current stopwatch time.

The up button (BNTU) is utilized to toggle and record the lap times. When the lap button is toggled, the system will automatically save the current stopwatch time in the next available lap time slot. The reset button (CPU RESET) is utilized to Reset the Stopwatch. When toggled, this will reset the timer and clear any saved lap times from the RAM. *Figure 1* shows how we laid out the Nexys A7 programmable board and the stopwatch controls.

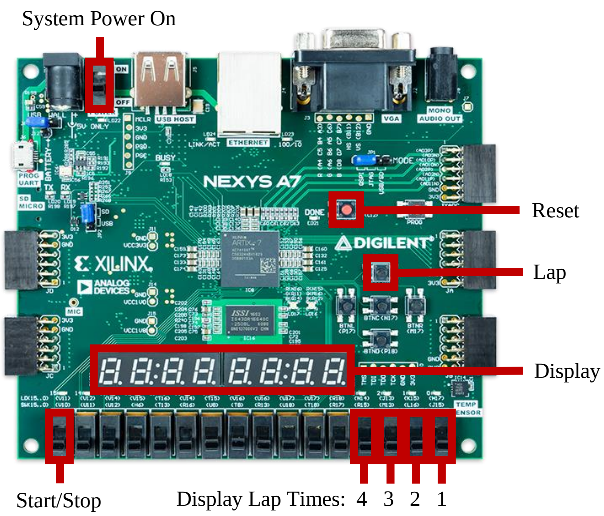


Figure 1 – Board Layout

C. *The Display*

The Nexys A7 board contains two four-digit common anode seven-segment LED displays that are configured to behave like one, single, eight-digit display. Our stopwatch system displays hours, minutes, seconds, a hundredth of a second, and a tenth of a second. The user can choose to display either the current timer status or a previously recorded lap time. *Figure 2* shows the wiring diagram of the eight-digit display. The display is split into five sections. *Figure 3* shows what is displayed when the clock is running.

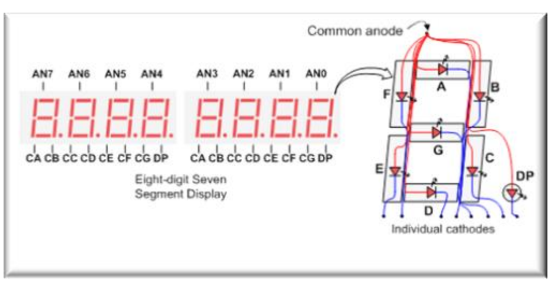


Figure 2 – Wiring Diagram

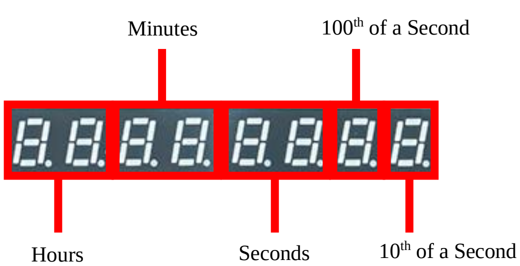


Figure 3 – Display Layout

D. *System Layout*

Our system includes several essential VHDL coding techniques and components we learned during the semester in class and in the lab. We utilized BCD Counters, RAM, FSM, Encoders, Decoders, Multiplexors, and Seven Segment Decoders. All the components are tied into the same clock and reset input. The counters feed directly into the both the RAM system and the seven-segment display system. What gets displayed is determined by what switches the user trips. *Figure 4* shows an AutoCAD drawing of our overall layout for the stopwatch system.

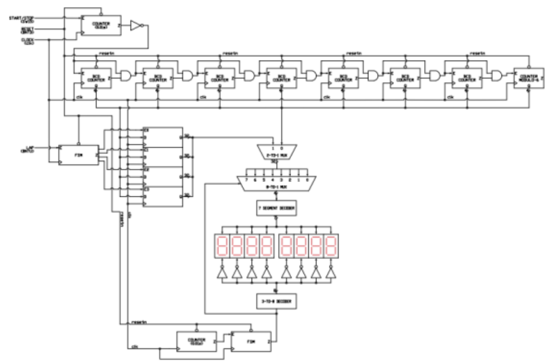
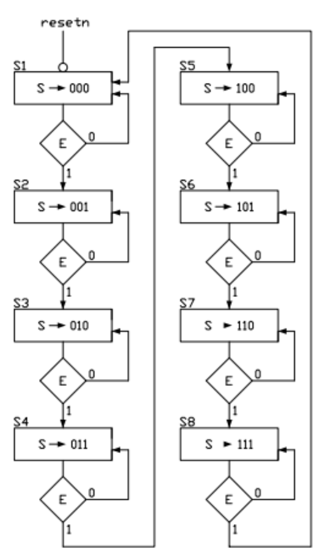


Figure 4 – System Layout

E. *Finite State Machine*

Our FSM has eight states to control the seven-segment display. The input of this FSM will be controlled by the output pulse "z" from the counter inside the serializer. The pulse "z" from the counter feeds directly into the enable for the FSM so that when the counter reaches one millisecond, the machine will move to the next state. The FSM shifts between states in coordination with the counter and sends its values to the output "S". The FSM then feeds its output "z" directly into the 3-to-8 decoder and the 8-to-1 multiplexor. *Figure 5* shows the design of our FSM.

Figure 5 - FSM



F. BCB Counters

The Nexys A7 board has eight, seven-segment displays. By utilizing all eight displays, our system can display a maximum time of ninety-nine hours, fifty-nine minutes, fifty-nine seconds, nine hundredths of a second, and nine tenths of a second. The main counting system utilizes nine counters to increment the stopwatch timer. The lead counter increments by a hundredth of a second (0.01s) and feeds into the rest of the counters. The counters feed directly into the both the RAM system and the seven-segment display system. *Figure 6* shows the drawing design of our main counting system.

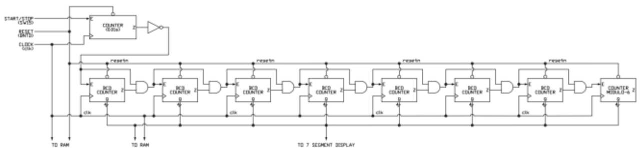


Figure 6 – BCB Counting

G. Random Access Memory

A common feature on stopwatches is the ability to record lap times while the timer is running. This is especially important for sports and vehicle testing. We implemented into our system a way for the user to record lap times and view them at any time. For this, we utilized a RAM emulator to store this data. Our RAM is capable of storing up to four lap times. The lap input is connected to the up button (BNTU) on the Nexys A7. When the button is pressed, the system automatically stores the exact time in the next available time slot. The user can view these lap times by tripping one of the switches (SW0-SW3) assigned to the lap times. When switch #0 is tripped, it will display Lap #1 time. When switch #1 is tripped, it will display Lap #2 time, and so on. *Figure 7* shows the drawing design of our RAM emulator.

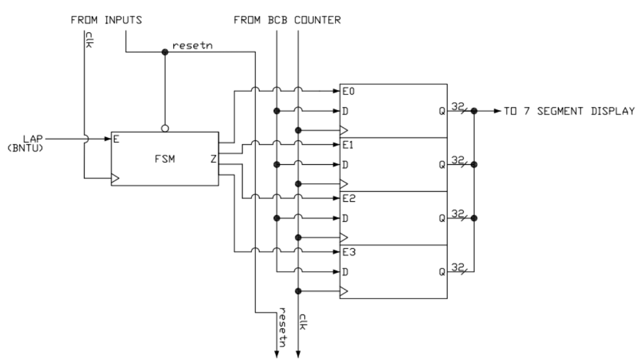


Figure 7 – RAM

H. Seven Segment Display

As stated previously, The Nexys A7 board contains two four-digit, seven-segment LED displays that are configured to behave like one, single, eight-digit display. Our stopwatch system displays hours, minutes, seconds, a hundredth of a second, and a tenth of a second. The user can choose to display either the current timer status or a previously recorded lap time. The display is controlled by a 2-to-1 multiplexor to determine whether a lap time will be displayed, or the current watch time is displayed. Following the 2-to-1 multiplexor is an 8-to-1 multiplexor and the seven-segment decoder. The Up Button (BNTU) is utilized to toggle and record the lap times. Switch #15 (SW15) is utilized for the Start/Stop feature. *Figure 8* shows the seven segment display layout.

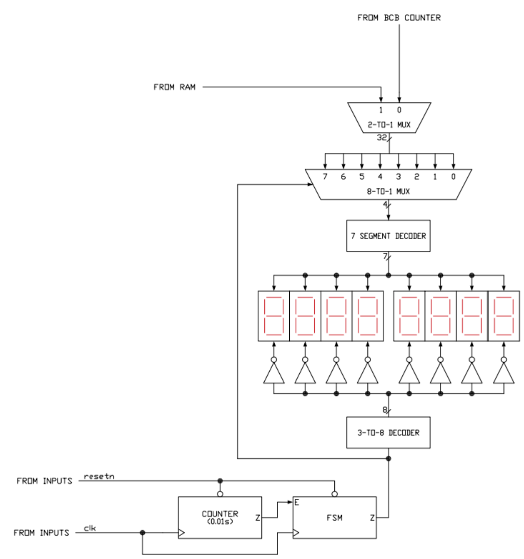


Figure 8 – Seven Segment Display

III. Experimental Setup

As stated previously, we used the Nexys A7 programmable logic board to run our stopwatch. We worked in the Viviado VHDL programming environment to build and code our stopwatch design. We created source files for all of our main competes such as the seven-segment decoder, the BCD counter, the BCD counter modulo 6, the adder counter, the multiplexor, the top file, and the testbench. We utilized the provided constraints file for our specific board. In our case, we have the 50T version of the Nexys A7.

IV. Results

Once all of the code was written, we combined all of the code to one Viviado project and downloaded the program the Nexys A7. We then began debugging the system. This is where we ran into many setbacks and slowdowns for our project development. Immediately we were fixing issues that was preventing the stopwatch from performing and intended.

We slowly buffed out all of the bugs and made progress towards a functioning system. Once we finished debugging the system, we tested the stopwatch to ensure all of its functions were fully operational. As intended, the system displayed hours, minutes, seconds, a hundredth of a second, and a tenth of a second. The clock was able to start and stop at any time with the start/stop switch. When the lap button was toggled, the system successfully recorded up to four lap times. Each lap time was displayed when its assigned switch is tripped.

CONCLUSIONS

Designing and building the stopwatch system was easier said than done. For a device that is relatively simple as an idea, it proved to be very challenging when attempting to implement the idea in the real world. We ran into many setbacks and bugs that practically brought the development to a halt at times. Ultimately, in the end, we solved all our issues and were able to create a stopwatch that performed as we desired. Our stopwatch design is a simple, easy-to-use stopwatch for everyday purposes. Our stopwatch has the essential functions of an everyday standard stopwatch that anyone can use.

REFERENCES

[1] Llamocca, Daniel. *Reconfigurable Computing Research Laboratory*, http://www.secs.oakland.edu/~llamocca/index.html.

[2] Brown, Arthur. "Nexys A7 Reference Manual." *Nexys A7 Reference Manual-Digilent Reference*, https://digilent.com/reference/programmable-logic/nexys-a7/reference-manual.

[3] "Xilinx Vivado." *Wikipedia*, Wikimedia Foundation, 30 July 2021, https://en.wikipedia.org/wiki/Xilinx_Vivado.

Link to our video Demonstration:

https://dai.ly/k2KDsaBwrpGfJQxr89G