Final Project

Alarm Clock

List of Authors (Nathaniel C. Dumitrescu, James Fazecas, Grant Parker, Jacob Monks)
Electrical and Computer Engineering Department
School of Engineering and Computer Science
Oakland University, Rochester, MI
e-mails: ncdumitrescu@oakland.edu, jfazecas@oakland.edu, gparker@oakland.edu, jacobmonks@oakland.edu

***Abstract*— The Alarm Clock is a device that is used all around the world. The clock being a tool that is used to indicate, keep track, and measure time, makes it a very necessary instrument to society across the globe. Society is structured around time; work, meetings, school, religious gatherings, and other events are all dependent on time. Time is a means of organizing humanity and their affairs. Therefore the alarm clock is a very important invention as it allows one to sync their life to the time structure of society around them. When comparing a typical alarm clock sitting on your nightstand to this project's alarm clock, the same conceptual application is implemented between the two. The end goal is to set a time for a buzzer or light to go off, indicating that the requested time input has passed. Due to the knowledge we gained from EGR 2800, pertaining to programming, alongside ECE 2700 and VHDL, this project is achievable.**

I. Introduction

The scope of this project is to design and program an alarm clock using VHDL and a Nexus A7-50T board. The reason an alarm clock was chosen for this project was because of the vast use of it. This alarm will indicate that the desired time has come using a light. The clock will be a 24-hour clock in contrast to a 12-hour clock, reason being it will avoid the requirement to implement AM and PM differentiation.

This report will cover the methodology used for designing the VHDL code for the alarm clock, the way the alarm clock works and was tested, and an obstacle encountered through the process of designing.

The main topics learnt in class that this project uses are the design of a counter, concept of an FSM, and the methods of storing bit information.[3]

An additional topic that we were required to learn on our own for the completion of this project was how to control the incrementation of the digits when pressing the button.

II. Methodology

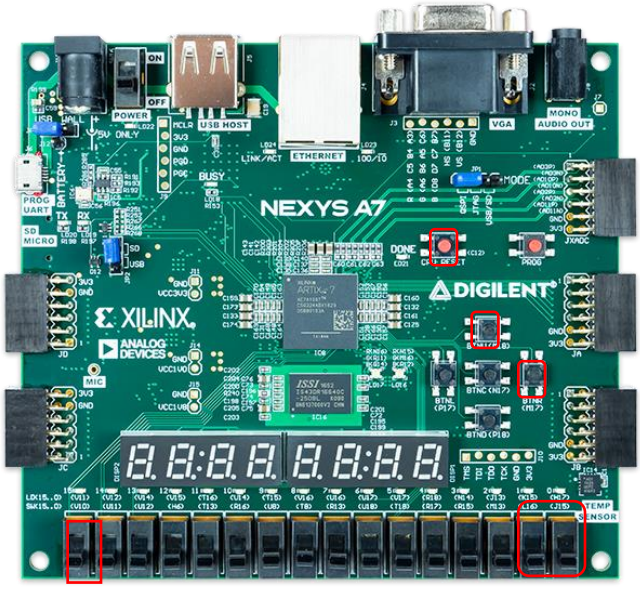
A. Approach

The alarm clock is divided into two basic parts. First part is the actual clock and the second part is the alarm. These parts also outline the approach taken to solve this assignment.

For the clock portion, the major task is setting the time and then having it saved. This will be done through a basic clock design including six counters. One counter will be assigned to each of the four seven-segment displays. Then there will also be a counter for the seconds and one for milliseconds.[3] These will not be displayed and will instead be used for incrementing the minutes and selecting which display is being used.

For the alarm portion, the major task is setting the alarm time and saving it. This will be done using d-flipflops to store the data for the hours, minutes, and seconds of the alarm. The code will then run a comparator that will compare the current time to the alarm time set to cause the alarm to sound when the time matches the set time for the alarm.

This design has three switches and three buttons. The left most switch is used as the alarm set switch. When user desires to set the alarm, they must flip the switch up, use the buttons to set the desired time, then return switch position. The two switches on the right are used to turn the clock on and turn the display on. One button is used as a reset which clears out all the alarm and clock data. The other two buttons are used to select the hour (top button) and minute (right button) time.



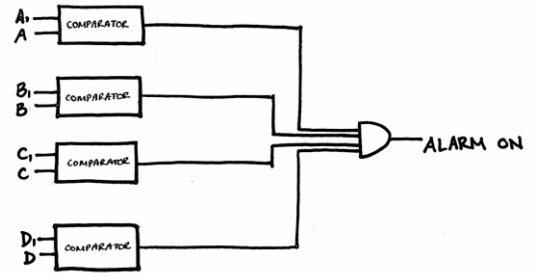
B. Block Diagram and Description

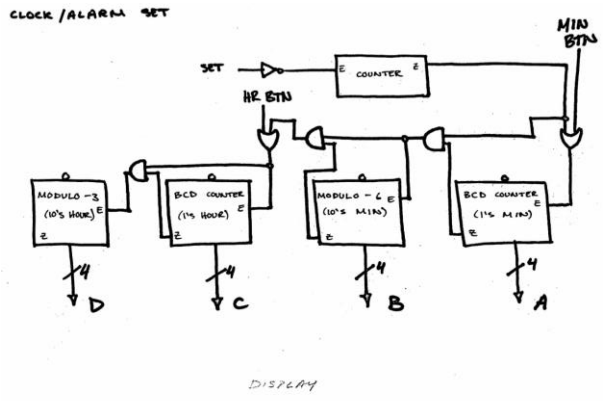
Clock/Alarm Set:

This image shows the main mechanism that drives the clock, 2 BCD counters, a Mod-6 counter, and a Mod-3

counter. A main counter serves as the main input to this system, counting milliseconds. The min_btn and hr_btn are placed accordingly so that they increment the proper counter when pressed.

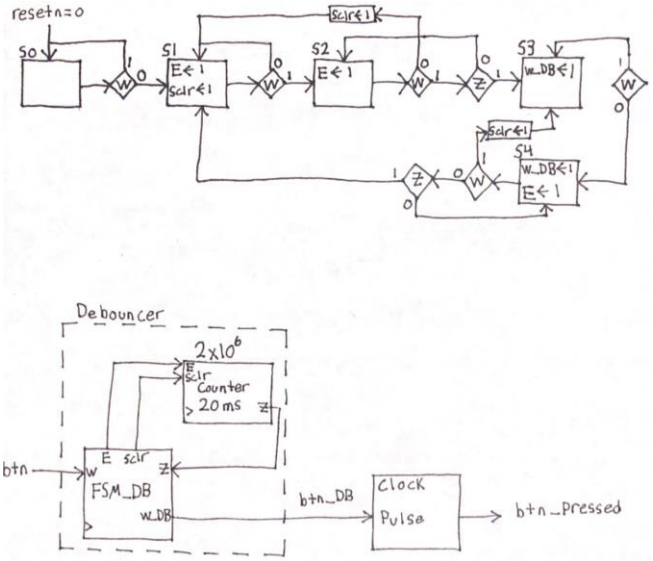
This is a basic display of why the alarm goes off. The time from the counters is compared to the time of the saved alarm through 4 comparators, and then the result of an and gate will produce a 1 when each part is the same.





Button Debounce and Clock Pulse: [1,3]

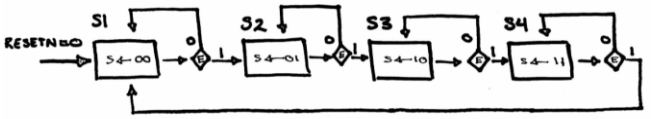
The button debounce state machine checks the input button click so it only reads the button once it has been at the constant state for 20ms, so it does not output the different signals when you first press it for when it oscillates trying to find its high value, and the clock pulse takes the input button and makes the output high for only one clock tick no matter how long button is held. The diagram of the FSM_DB as well as the Debouncer is shown below.



As far as the clock pulse goes, its purpose was for when the board reads the input value as high, a high output comes out for only one clock tick. In our case no matter how long you hold the button down it will only output a high signal for one clock tick until you let go of the button and press it again.

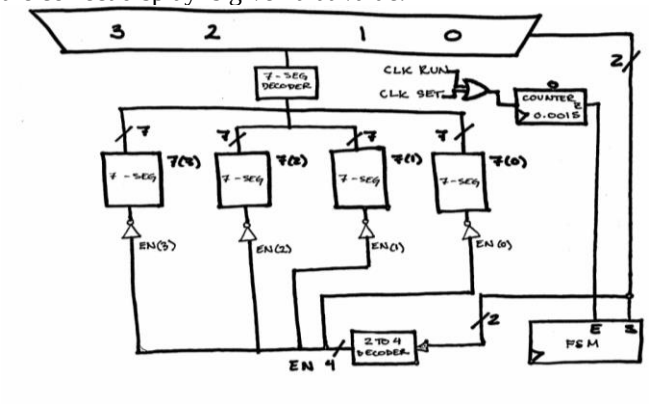
FSM: [1]

The Finite state machine shown here is very simple. It cycles through each state at a regular pace so that the output can determine which counter gets incremented, whether it's minutes or hours, and the 1s place or 10s place.



7seg Display: [1]

This is a representation of the components that turn on the 7-segment displays. The MUX determines which counter value gets passed, and the 2-4 Decoder ensures that the correct display is given that value.



III. Experimental Setup

The hardware will consist of a Nexys A7-50T board. We couldn't figure out how to slow down the clock counter enough to see in the simulation, so our program had to be tested by simply plugging in the board, programming it, and seeing if it works.[3] It was very difficult to set the alarm and the clock within a testbench so a testbench was not included.

We also had trouble figuring out how to hook up a buzzer and cause it to sound so instead we used an LED to indicate that the alarm time desired has come. We found this a way to still show that alarm clock is working.[1]

For the sake of easy demonstration and experimentation, we created the clock to increase the minute digit at ten seconds instead of sixty seconds. To have the clock indeed increasing at sixty seconds would have just created a tedious experimentation and demonstration process.

IV. Results

Attached is a link to a demonstration of how the Alarm Clock we designed works. We were able to use and implement counters to create a 24-hour clock. The results were as expected except for one situation. When trying to use the buttons to set the time, the numbers would increment very sporadically due to the sensitivity of the button. The solution was a button debounced function and a clock pulse [2] function that allowed for the digits to increase by one value each button press. This solution was something that we needed to look for outside of the class material.

https://www.youtube.com/channel/UCHDZTgKk2la0lyQ2if9fDGA

Conclusions

Taking what we have learned from Dr. Llamocca's lectures and applying the knowledge as a team we were able to complete the final project task. However, the clock pulse[2] function was an unfamiliar aspect that resulted with the group figuring its implementation as a team. There are many takeaways from this project, not just the VDHL coding that was mentioned in this report, but also each member in the group learned that teamwork and cooperation through proper communication is key in reaching full potential when it comes to success and time efficiency. This group project gives a glimpse of how working with others will be in the future and gives each group member a better appreciation for the alarm clock sitting on their nightstands.

References

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

[2] "Counters in Digital Logic." GeeksforGeeks, 13 Feb. 2018. https://www.geeksforgeeks.org/counters-in-digital-logic/

[3] Kester, David KesterDavid. "ONE Clock Period Pulse Based on Trigger Signal." *Stack Overflow,* stackoverflow.com/questions/20174889/one-clock-period-pulse-based-on-trigger-signal.