BRADLEY
UNIVERSITY

ECE 102: Introduction to EE: Digital Systems

Digital Alarm Clock

Grant Abella, Connor Mandli

Contact Information

Grant Abella
Electrical and Computer Engineering Student
Caterpillar College of Engineering and Technology
Bradley University

Williams Hall 340
821 N. Duryea Place
Peoria, IL, 61606, USA

Phone: +1 (630) 776-4026
e-Mail: gabella@bradley.edu

Table of Contents

Project Confirmation Letter

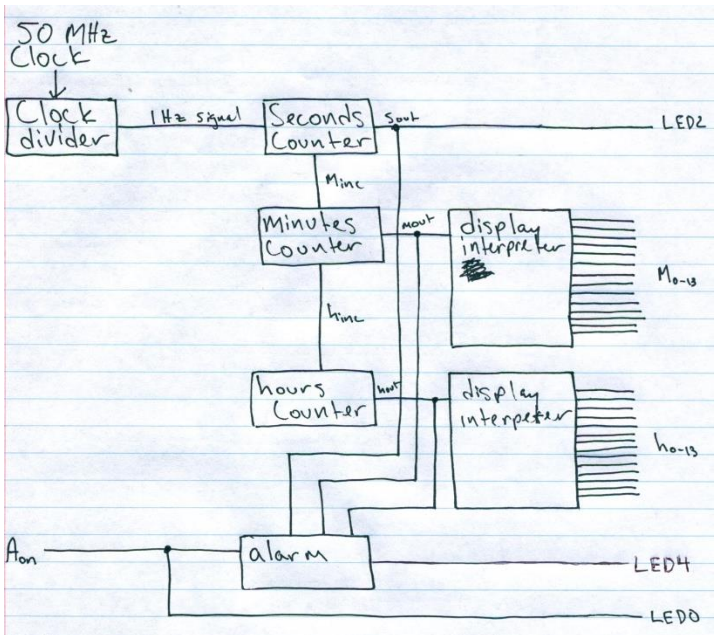
The group decided together to do the alarm clock project for a number of reasons, first and foremost being experience. Having used the Xilinx software to implement synchronous counters, clock dividers, resets, and discrete logic, we have built up a strong foundation in each of the fundamental components of the clock. It provides us with challenges as well; we will be directing the outputs into multiple seven-segment displays. We have decided to implement an additional alarm section into our design, which can be set to go off a specific time. The time we've spent in the lab has prepared us well for our project; it will not be a challenge for us to get a timing diagram and to upload the circuit to the Spartan 3E board. The lab time has also given us vital experience with VHDL programming, which takes some time to get used to; instead, we will be starting this project well beyond the learning curve. Overall, we chose this project because we knew it is doable, and that we will have ample opportunity to add in something extra.

1 Project proposal

1.1 Specifications

- Seconds counter – increments by one on the rising edge of the 1Hz clock signal. This counter resets to zero one second after 59 seconds, and sends a signal to the minutes counter, telling it to increment by one.
- Minutes counter – increments by one upon receiving a signal from the seconds counter. This counter resets to zero one second after 59 seconds and 59 minutes. As it resets, it outputs a signal that is sent to the hours counter, telling it to increment by one.
- Hours counter – increments by one after receiving a signal from the minutes counter. This counter resets to zero one second after 59 seconds, 59 minutes, and 23 hours.
- Alarm – this component is only active if SW0 is switched on. LED0 will be activated in order to notate that the alarm is active. The component then reads the status of each counter upon the rising edge of the 1Hz clock signal and compares their values to its pre-programmed alarm time. If the times on the counters match up with the programmed alarm time, the alarm will be activated, and LED4 will be turned on.
- Display interpreter – the minutes and hours counters will output to display interpreters, which will determine which segments will be activated on the displays in order to show the value contained in each counter. Since we will not be displaying the seconds on a seven-segment display, the seconds counter will output to LED2.

1.2 Block Diagram



- The clock divider is what converts the Spartan 3E's 50MHz clock into a 1Hz signal. This signal is passed into the seconds counter as an input.
- The seconds counter has one output called s_{out}, which goes to LED2 on the Spartan board, and this LED will blink every second that passes. The other output associated with the seconds counter is m_{inc}, which is activated one second after the seconds counter hits 59 seconds. This output is sent into the minutes counter, signaling that the minutes count should increment by one.

- The minutes counter has one output called m_{out}, which carries the value contained in the minutes counter at that moment. This output is sent into the first display interpreter. The other output of the minutes counter is h_{inc}, which is activated one second after the seconds counter hits 59 seconds and the minutes counter hits 59 minutes. This output is sent into the hours counter, signaling that the hours count should increment by one.

- The hours counter has only one output, h_{out}, which is sent into the second display interpreter.

- Both display interpreters read the input that is sent into them and convert that information into 14 signals that will be wired to the seven segment displays. Each display interpreter needs to have 14 outputs because the values contained in both the seconds and hours counters will be two digits.

- The alarm takes four inputs. The first input is A_{on}, which will be bound to SW0 on the Spartan 3E. This input denotes whether or not the alarm will be active. This signal is also sent to LED0 in order to display that the alarm is on. The other three inputs are s_{out}, m_{out}, and h_{out}. The alarm uses these to tell what time the clock is displaying at that moment, and it compares these values to its pre-programmed alarm time. If the time on the clock matches the alarm time, a signal is sent to LED4 to denote that the alarm has been activated.

1.3 Division of Labor

We plan to split up the workload evenly between the two of us. The main counter (using synchronized flip-flops), circuit diagram, project introduction, and background research will all be handled by Grant. The output section (with the seven-segment displays), the additional reset logic, design procedure, project results, and final conclusion will be taken care of by Connor.

1.4 Timeline and milestones

- Have clock code written by November 30th

- Implement seven-segment display by December 2nd

- Complete any remaining tasks before December 4th

2 Project Report

2.1 Introduction

- Motivation:
  We were motivated to complete this project mostly because we wanted to gain an in-depth understanding of how the project design process works from beginning to end. This is the first major project of our ECE program, and we were eager to see it progress from beginning to end Also, we wanted to gain further understanding of how seven segment displays work because neither of us had worked with them before. This proved challenging, but once we got it working it felt very rewarding and further drove our progress as we finished up the project.

- Problem description:
  The alarm clock has to be able to count seconds, minutes, and hours, and must be able to display 4 digits of base-10 numbers for minutes and hours. The numbers must count up in the correct order and overflow into the next bit after the correct amount of time (i.e. 09:09 will go to 10:10 after 60 seconds). The numbers will be displayed on four 7-segment displays (one for each bit). The seconds counter will be output to an LED (meaning the light will blink on and off every second). The alarm section will be enabled by a switch; an indicator LED will turn on when the alarm is enabled. When the alarm time is reached, an LED will light up and a small sound will be generated.

- Report outline:
  The following report details the ins and outs of the digital alarm clock project, including background research, the design procedure, a demonstration of the complete project, finalized results, and a final summary.

2.2 Background

The clock we designed uses four seven segment displays to show minutes and hours, and includes a programmable alarm. The clock is based off of a series of counters that add multiple bits for seconds, minutes, and hours. We utilized the clock divider circuit to provide the seconds counter its own clock signal, and had the minutes counter detect the seconds count and increment based on that value. This was the same for our hours counter. The essentially made the entire system an asynchronous counter. The alarm is enabled by a slide switch; when an alarm is set the indicator LED will be on, and when the clock reaches the programmed alarm, an LED on the board lights up and the alarm will sound.

2.3 Design Procedure

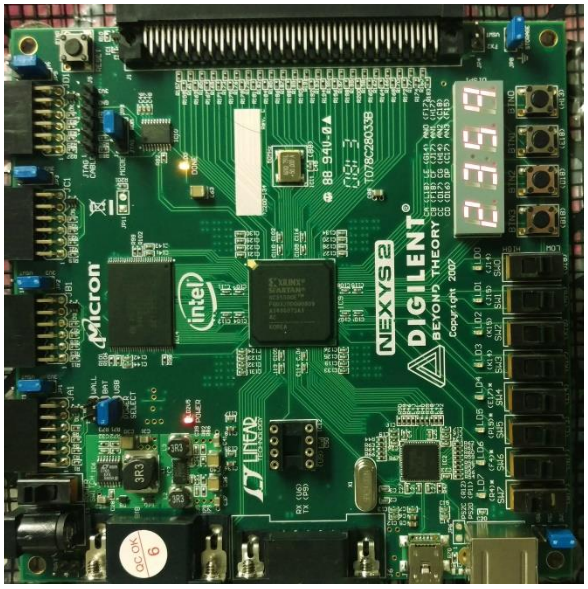
Our general procedure was as follows:

1. Implement clock divider
   a. This was not difficult as we already had the clock divider file. Also, we had already incorporated the clock divider in a previous assignment, so we new how to work with it. The only thing we had to change was that we needed a second output from it to run the display clock. We needed a signal much faster than the 1 Hz signal that it was already written to output.

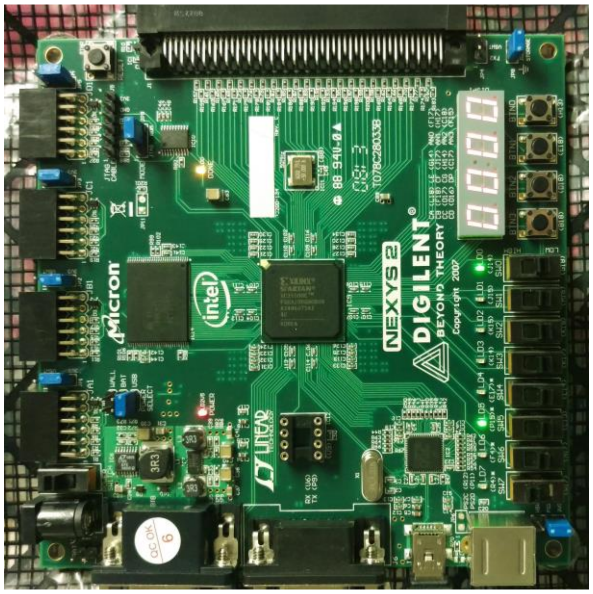
For the display, we used a 100 kHz signal since it needed to be fast enough so that the switching of each display would not be noticeable to the human eye.

2. Design clock circuit
    a. 3 counters (seconds, minutes, hours) and overflow / reset for each bit
        i. The seconds counter increments on the rising edge of the 1 Hz clock signal. The minutes counter checks the value of the seconds counter. If the seconds counter equals 59 and another rising edge occurs, the minutes will increment by one and the seconds counter resets back to zero. The hours counter checks the value of the minutes counter. If the minutes and seconds counters both equal 59 and a rising edge occurs, both the minutes and seconds counters reset back to zero and the hours are incremented by one. If the hours counter equals 24, it is reset to zero.
3. BCD converter
    a. The clock outputs 3 vectors: the binary representations of the hours, minutes, and seconds counters. The BCD converter takes these values and implements an algorithm to convert them into binary coded decimal. This is necessary because we needed to be able to splice the values into digits so that we could show them on the display.
4. Output to 7-segment display
    a. This task is handled by the display interpreter, which takes in the 100 kHz sinal and runs a process at that speed that counts up to 5. On each count, the anode value for the display is changed, and the corresponding digit of the selected counter is displayed. This is done four times, and on the fifth count, the anode vector is reset to "1111" which turns all of the seven segment displays off momentarily, and then the count is reset back to zero and the process repeats. This anode rest is necessary because the display doesn't appear correctly without it.
5. Alarm section
    a. Set/enable alarm
        i. This is handled in the clock file. The file takes a binary input signal that is bound to SW7 of the board. If this value is determined to be 1 (meaning the switch is activated), then the alarm is set and every second, the values of each of the three counters of the clock are compared with the three alarm values for seconds, minutes, and hours.
    b. Output sound/alarm out
        i. If all three values of the clock counters are determined to be the same as the pre-programmed alarm times, a signal is output that denotes that the alarm has been triggered.

2.4 Project Results and Demonstration



Here you can see that the clock is at 23 hours and 59 minutes. SW7 is on, which sets the alarm to active.



As the time switches to 0 hours and 0 minutes, the alarm is triggered as this is the pre-programmed alarm time. LD5 is illuminated, denoting that the alarm has gone off.

2.5 Summary and Conclusion

Despite all of the background knowledge on programming and sequential circuit design our group had, it was necessary to further this using outside sources. The most significant part of our research all revolved around our use of 7-segment displays and the Nexys2.

Even having gone over the implementation of 7-segment displays in class, we had some difficulties with our implementation. We quickly realized that the example in class had been using a common cathode display, and the board used a common anode orientation which made the display show nonsensical symbols. Even inverting our output section we still weren't seeing the numbers we wanted; we discovered that, counterintuitively, the signals going to the board had to be ordered from "g" to "a" (i.e. if we sent

"1000111" to the board, the first bit would be the "g" segment). With our outputs inverted and reversed in order, we were able to see the numbers we wanted.

After this, we encountered a very significant problem: the logic we had been planning out had been based around each display having seven of its own inputs--but the displays on the board shared seven input lines, with a four selection bits to designate which display was showing each output. This was simultaneously more and less complicated; with 27 unique segments, it would have been a nightmare to route signals to, yet there were new pieces of code needed to put to utilize the selection bits.

The four displays on the Nexys2 board are encoded in a unique fashion; they all share the standard 7-segment "a-g" inputs, with each of the four displays being selected with a selection bit. The catch is that only one display can be on at a time, meaning that to show digits at once it has to switch the first one on, turn off, and then turn on the next digit. A digital clock will normally have to have 3-4 digits on at a time; to show a four digit number like "11:35," the displays have to be switching on and off at a rate faster than we can see so that all four digits will look like they are on continuously. Understanding this concept took some time, and coding it was difficult because with a low refresh rate, the display was very hard to look at as it flickered and stuttered.