CSE 140L Lab 2

(Amy Nguyen, A16125627); (Mark Lorenzo, A15955698); (Jared Villanueva, A15821317);

Academic Integrity

Your work will not be graded unless the signatures of all members of the group are present beneath the honor code.

To uphold academic integrity, students shall:

- Complete and submit academic work that is their own and that is an honest and fair representation of their knowledge and abilities at the time of submission.
- Know and follow the standards of CSE 140L and UCSD.

Please sign (type) your name(s) below the following statement:

I pledge to be fair to my classmates and instructors by completing all of my academic work with integrity. This means that I will respect the standards set by the instructor and institution, be responsible for the consequences of my choices, honestly represent my knowledge and abilities, and be a community member that others can trust to do the right thing even when no one is watching. I will always put learning before grades, and integrity before performance. I pledge to excel with integrity.

(Amy Nguyen)
(Mark Lorenzo)
(Jared Villanueva)

Free Response

Please answer the following questions.

1. Please write a summary paragraph explaining how you tested your alarm clock in part 1. (4 pts)

It is insufficient to say “I ran the testbench and it worked.” Please tell us what is happening in the testbench and other testing methods that you may have implemented. Word limit: 200 words.

Part 1: I first tried to ensure that my files compiled on ModelSim with a transcript output of any kind. Afterwards, I implemented normal clock behavior, making sure to incrementally compile and run to verify that seconds, minutes, and then hours were displaying something at least. I then went back and honed my logic to account for Timeset and re-tested to see if the correct normal clock functionality and time setting functionality were in the ballpark of around 8 o’clock. I finally implemented the alarm, following a similar structure to the Timeset and it worked. I verified that each instance of output in the testbench was actually being outputted in the transcript before modifying the testbench to output to a file.

2. Please write a summary paragraph explaining the day of the week enhancement and how it was implemented. (4 pts)

Word limit: 200 words.

The day of the week enhancement simply adds a display to our clock in which it displays the day of the week in numerical form. The days are the week are represented by integers 0-6, in which 0 represents Monday and 6 represents Sunday. Essentially, we’ve kept all of our previous functionality from Part 1 of the lab, keeping the ability to set the time, and set the alarm, etc, and now we have added functionality to set the day of the week. We have also added the ability to increment the date based on the hours, so if it is 1 (Tuesday) at 23:59, it would update to be 2 (Wednesday) 00:00. We implemented this by creating the variables TDays (to keep track of the days) and TDen (to keep track of when the day counter would be enabled in order for it to increment), created a Day counter, and created a display.

3. Please write a summary paragraph explaining the date enhancement and how it was implemented. (4 pts)

Word limit: 200 words.

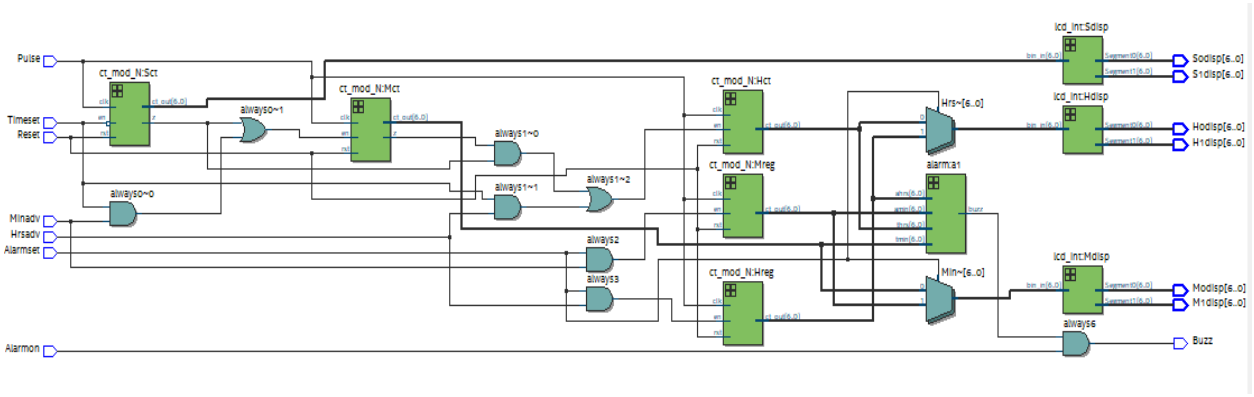
For the date enhancement, we first copied the code from part 2 over. In order to increment the date, we needed to know when the hours were reaching their maximum, just like how in part 2 for implementing the day of the week counter. We also needed to know what month it was so

that date knew when to signal that it was about to increment when date reaches either 28, 30, or 31. The Month counter also needed to know what date it was so that when the date was about to roll over, Month would increment. Thus these two counters relied on each other's time output value. The file ct_mod_D helped with identifying when the days should roll over (28, 30, 31) by taking in a month number input. Create new variables that are analogous to the ones already present from part 1 and part 2 and display the outputs of the Month and Date Plus 1 for each since the Month and Date are 1-index, unlike the day of the week counter.

Screenshots

Part 1

Screenshot of the RTL viewer top-level schematic/block diagram in Quartus (3 pts)

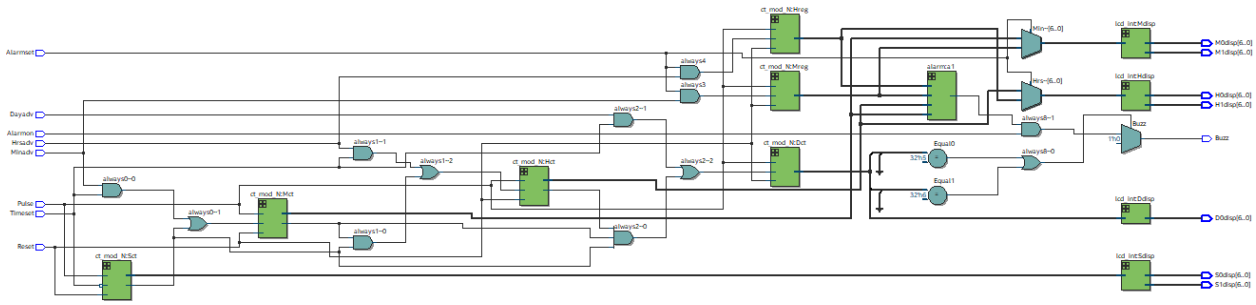


Please include the output file of part 1 testbench in your submission, and name it "output1.txt" (3 pts)

We will be looking for a text file with that name specifically, so be sure to rename it from "list.txt". Nothing is required in the writeup for this question.

Part 2

Screenshot of the RTL viewer top-level schematic/block diagram in Quartus (3 pts)

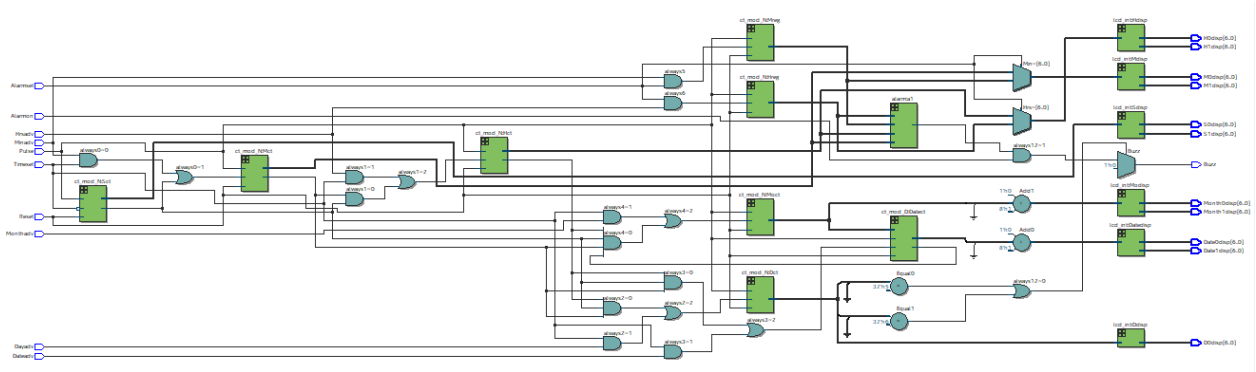


Please include the output file of part 2 testbench in your submission, and name it “output2.txt” (3 pts)

We will be looking for a text file with that name specifically, so be sure to rename it from “list.txt”. Nothing is required in the writeup for this question.

Part 3

Screenshot of the RTL viewer top-level schematic/block diagram in Quartus (3 pts)



Please include the output file of part 3 testbench in your submission, and name it “output3.txt” (3 pts)

We will be looking for a text file with that name specifically, so be sure to rename it from “list.txt”. Nothing is required in the writeup for this question.