Ricky Li

CSE140L

Alex Orailoglu

CSE 140L Lab Report

Part 1: In order to test part 1 of the lab, I made sure that my clock was ticking properly. I had to see that the clock's seconds and minutes reset after it hit 60 and that the clock's hours reset after 24. Once I had the clock part down, I started to implement the alarm portion. In the alarm test, I set the alarm time to 6:03 and let the clock run until it hit 6:03. Then I checked the buzzer probe and saw that it was high when the clock time matched the alarm time. I also checked that the probe was low when the requirements were not meant (S5 off or minutes or hours not lining up.

Part 2: In part 2, I changed both ClockDev and AlarmDev and created my own circuits ClockDev_French and AlarmDev_French. In order to do this, I added one more 2 more bits to ClockDev (S6 and M6) and then I removed a bit (H3). Same with alarm except I didn't have to worry about seconds for the alarm so I only had to add M6 and remove H3. Then I calculated $100_{(1100100)}$ and $10_{(1010)}$ in binary and had to configure the bits accordingly and connect them to a 3 input and gate to make sure the clock resets when it hits 100 for the minutes and seconds and 10 for the hour.

There are 86400 seconds in a day for the Western clock and there are 100000 decimal seconds in a French clock. This means that the pulse for a French clock would have to be around 1.157x faster than the pulse of a Western clock. This also means that a decimal minute would be about 1 conventional minute and 26 conventional seconds.

Part 3: I changed ClockDev_French and AlarmDev_French that I created in Part 2 and I just copied and pasted the registers and added other circuit elements in order to recreate the alarm and clock functions while also accounting for Days, Weeks, and Months. Then I rewired the registers and clock portions in order to have Days reset at 10, weeks at 3 and months at 10. I basically just looked over the previous clocks and meticulously made sure I had everything wired to where it was supposed to be. When testing my clock, my Minutes, seconds and hours were working however everything past that wasn't. So I went into the clock ctrl+f to see the values and saw that one of my registers just wasn't connected at all so I reconnected it and everything started to work out perfectly. All that was left was to copy and paste the Mux's and Comparators and LCD's in order to implement the other variables in my clock.

Part 3: I created yet another Clock named ClockDev_Leap which is the same clock as the one in part 3 except I added pins for the year and incrementing the year. I also created a variable named 'FESTIVE' that tracked whenever it was the 12th month. When it was the 12th month, 'FESTIVE' would be 1 and I had an and gate that reset week and incremented month whenever 'FESTIVE' was 1 and it was week 1. Same goes for the

days. However, I also created a variable named ‘LEAP’ whenever the year was 3mod4 which was whenever the first two least significant bits were both 1. This determined whether or not the days would go to 5 or 6 on month 12. So I had an array of gates that determined when to reset the days and increment weeks. (Day10 or ‘FESTIVE” * (Day5*NOT_LEAP or Day6*LEAP)).