Documentation for CS 20 Project 2: Stopwatch and Countdown timer

Members:

Nicolas, Jack Vincent -	2020-03124 -	M10-2
Ragunton, Carl David -	2020-04243 -	M10-2

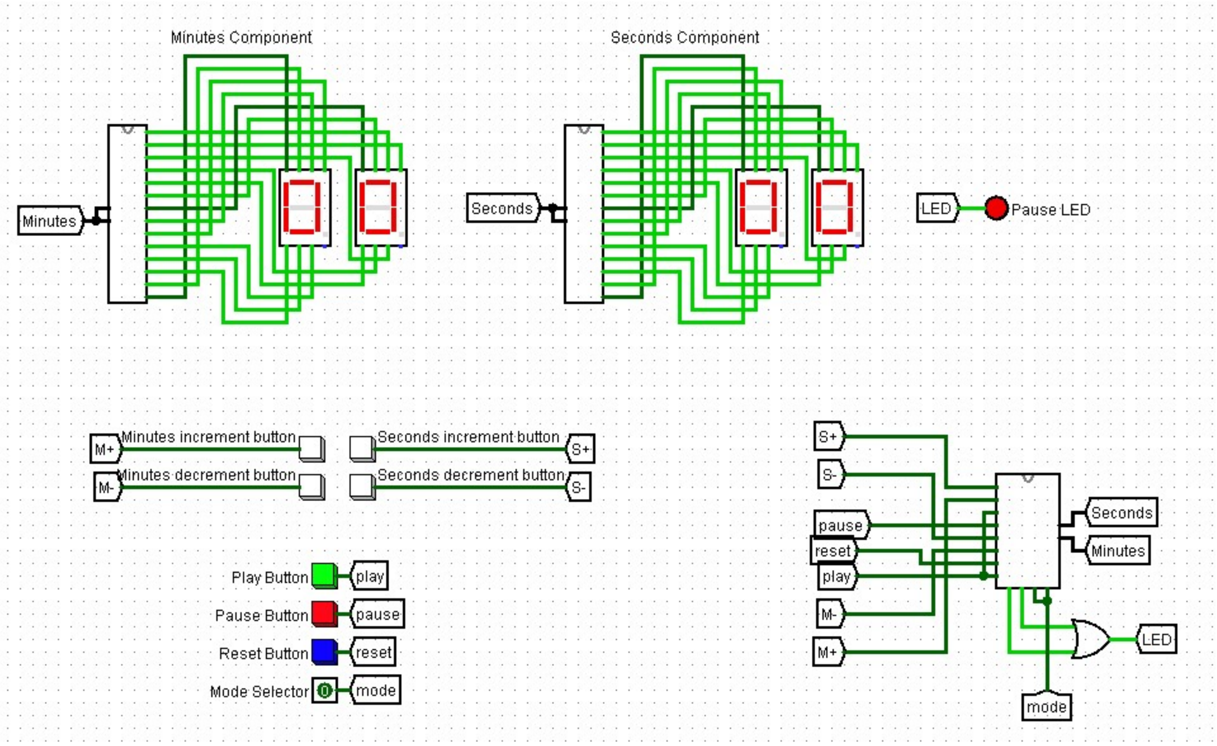
Set the clock frequency to 2Hz instead of 1Hz and enable the clock ticks before starting the circuit

Add the Logisim BCD to 7-segment LED Library when opening the circuit

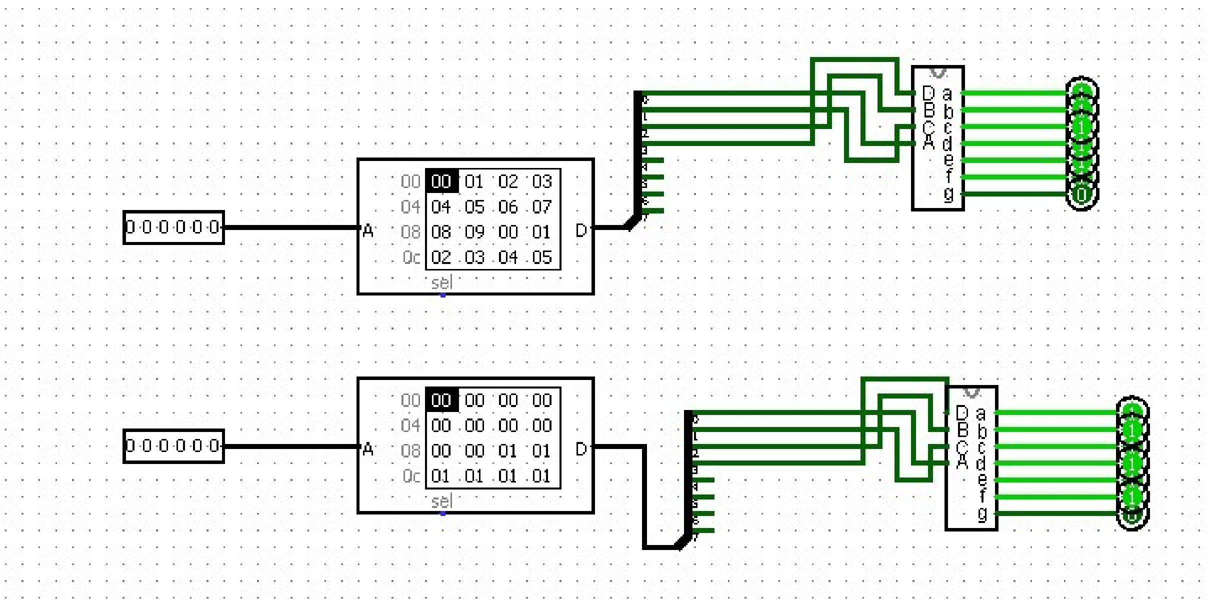
Google Drive to the Documentation Video:

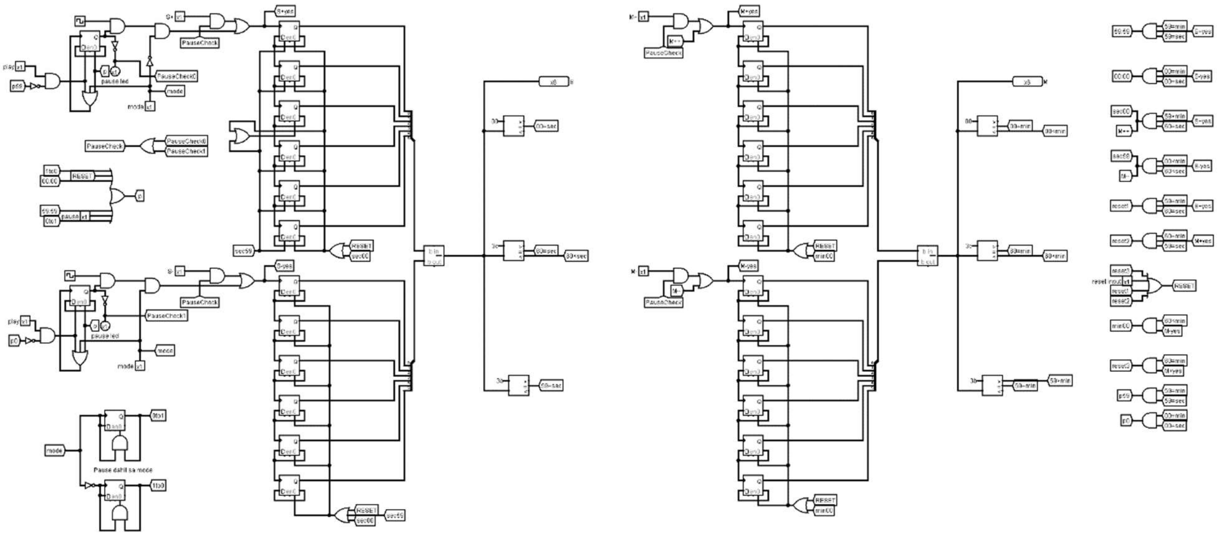
https://drive.google.com/file/d/1raa0pbBBBkm2LIICfnIeZCMlaqGfF6Pv/view?usp=sharing

Circuit:



Sub circuits:





(export image was used for the second sub circuit because a screenshot will not be able to clearly show its contents)

Division of Tasks:

The two of us did not know where to start and how to divide this project. Therefore, we shared our ideas and tried to work on what we know. The division of tasks went like whoever does it first and the other one will complete and correct the errors. That being said, most of the ideas were formulated together.

Here are the contributions made on the circuit:

Carl Ragunton:

- ROMs regarding the 7-segment display

-Timers

- Stopwatch and Countdown process

- Play and Pause Button

- Increment and Decrement

Jack Nicolas:

- Reset Button

- Edge cases of the whole timer (59:59 and 00:00)

- Edge cases between seconds and minutes

- Simplification of the circuit

Short Narrative:

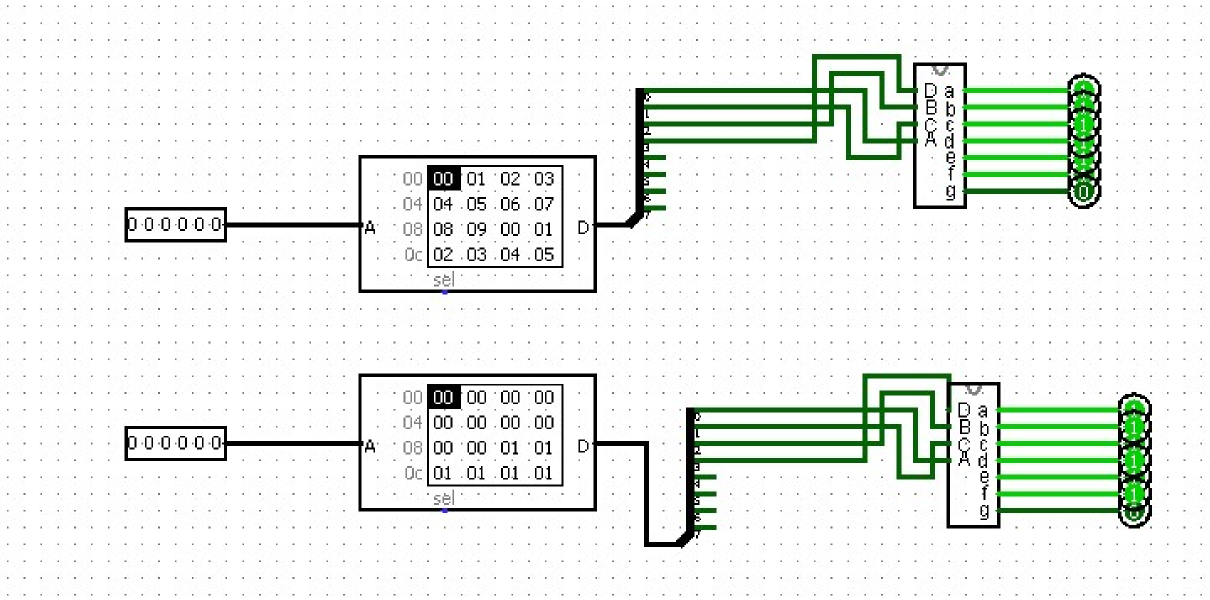
Ideas were shared between the two of us at the start but we did not exactly know how to divide the tasks. Carl Ragunton was the first one to build a successful timer and built the base processes of the circuits. Jack Nicolas, then, continued it and corrected all the possible errors. 6 bits were mostly used in this circuit because it is the smallest bit size that contains values from 00 to 59. The first part that to be made was the sub circuit that contains 2 ROMs for the 7-segment displays. The timers with the process of stopwatch and countdown came next. After that is the play, pause, and reset buttons. Resolving the problems regarding edge cases and some problems was done. Last is the simplification of the whole circuit. Only 2 sub circuits were used because it will be easier to adjust the values of the seconds and minutes if they are done in a single sub circuit. The other sub circuit is for the ROMs.

Details of Each Parts/Processes:

1. ROMs Regarding the 7-Segment Displays

The first digit and second digit of the numbers to be displayed are separated in different ROMs. This is because values in ROMs will be displayed in octal when connected to the 7-segment display. Also, we used the 7-segment display driver that was given on one of the lab exercises. The method on showing and connecting the ROMs, 7-segment display drivers, and the 7-segment display were also derived from one of the lab reports. All we need to do later is to connect the 6-bit value of the minutes and seconds here and they will be shown properly on the 7-segment displays.

Here is the sub circuit dedicated for this part:



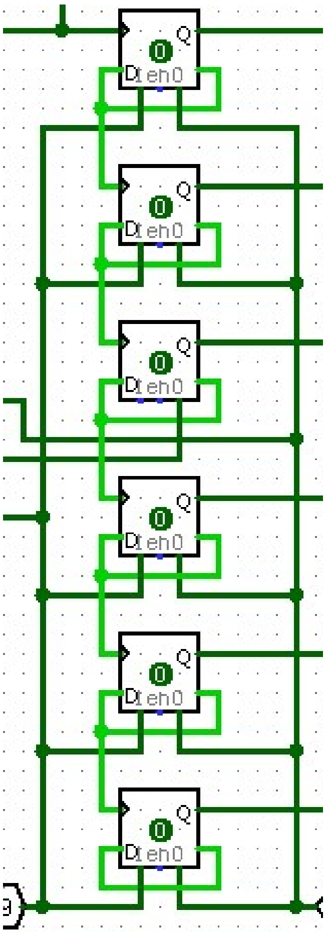
00	00	01	02	03
04	04	05	06	07
08	08	09	00	01
0c	02	03	04	05

00	00	00	00	00
04	00	00	00	00
08	00	00	01	01
0c	01	01	01	01

2. Timers

Again, knowledge from one of the lab activities were utilized here. 6 D-flipflops were used to create a 6-bit value that increases in value every time the first D-flipflop receives a pulse. But 6-bit contains values from 00-63 and not just 00-59. This matter will be explained later.

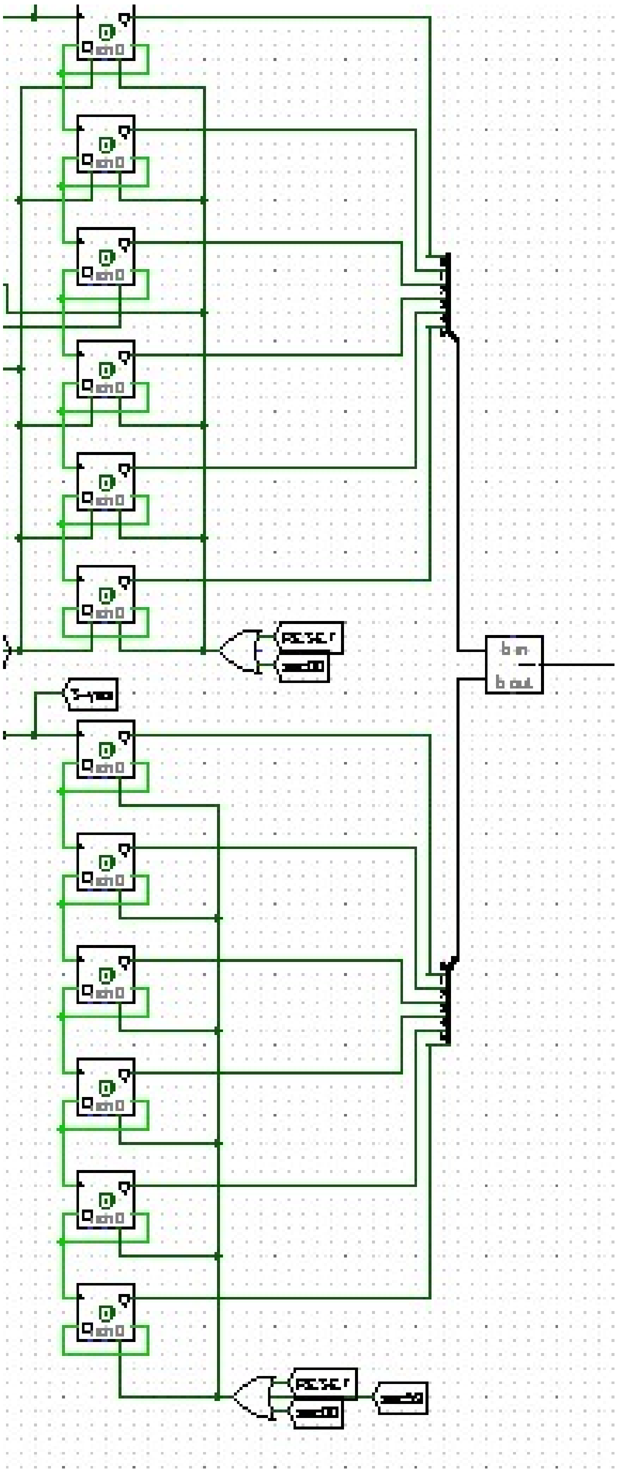
Here is the picture of the D-flipflops that acts as a timer (there are already other parts here for the other processes):



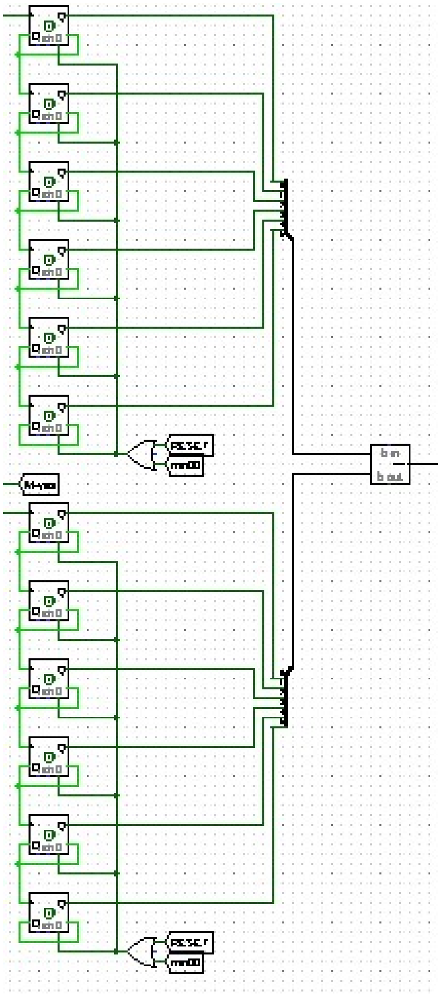
3. Stopwatch and Countdown process

Notice that the timer itself can act as a stopwatch. Now, we just need a countdown process. This can be easily done by creating another timer whose value will be subtracted from the value of the timer before. Just to be clear, there are 4 timers in this circuit. 2 for the seconds and 2 for the minutes. The conditions of when will the seconds affect the minutes will be dealt with later on.

These are the 2 timers for seconds:



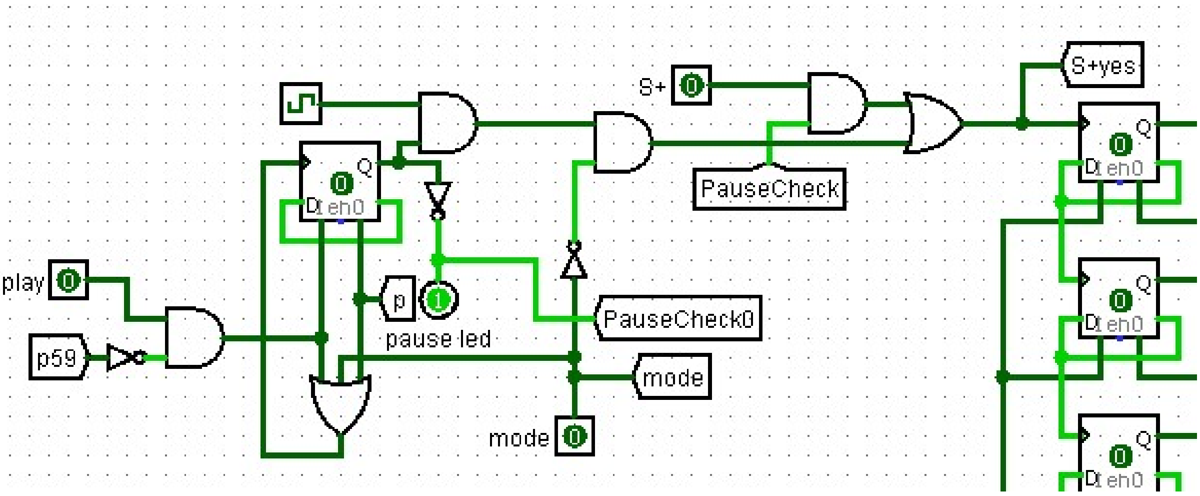
The timers for minutes are shown below:



4. Play and Pause Buttons

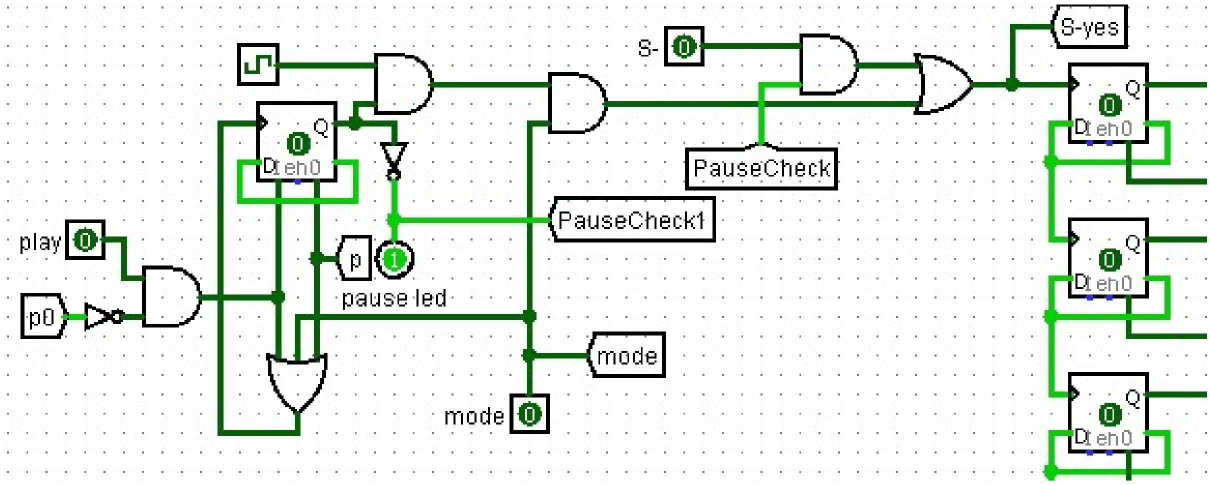
This part was basically done with literally with trial and error. Of course, there is a clock to send pulse on the timers when needed to play. Basically, in order for the stopwatch to play, the mode needs to be 0. The focus of this part is the D-flipflop. When the play button is pressed, it will force the D flipflop to output 1 which will then let the clock pulse reach the timer flipflops. On the other hand, when the pause button is pressed, the D flip flop's output value will become 0 immediately. The p59 here means that it will play until the value becomes 59:59. The pause LED as its name suggests will become 1 whenever the circuit is paused. The "pause" part here is specifically for the pause button. Other conditions will be explained later.

This is the part that deals whether the circuit will play or pause for mode 0 or the stopwatch:



For the countdown play and pause, it is basically the same except that it will only work when the mode is set to 0. Also, it will not work when the value is already 00:00.

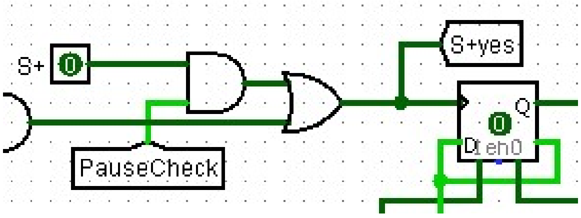
Here is the part for the countdown play and pause:

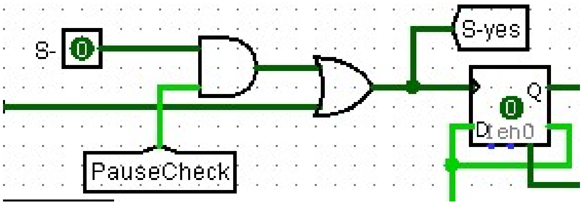


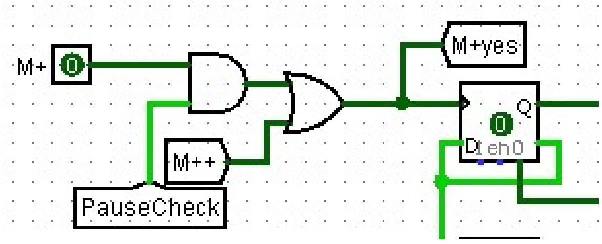
5. Increment and Decrement

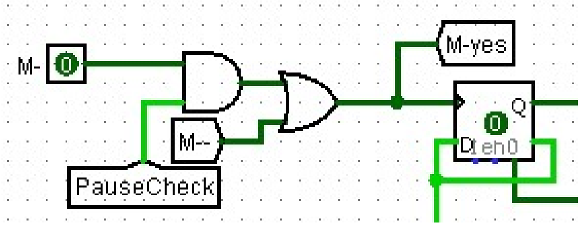
Simple enough, the increment and decrement will work when the respective buttons are clicked while the circuit is paused.

Here are all the increment and decrement processes for the seconds and minutes:



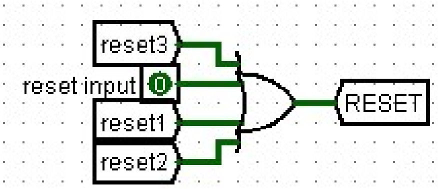




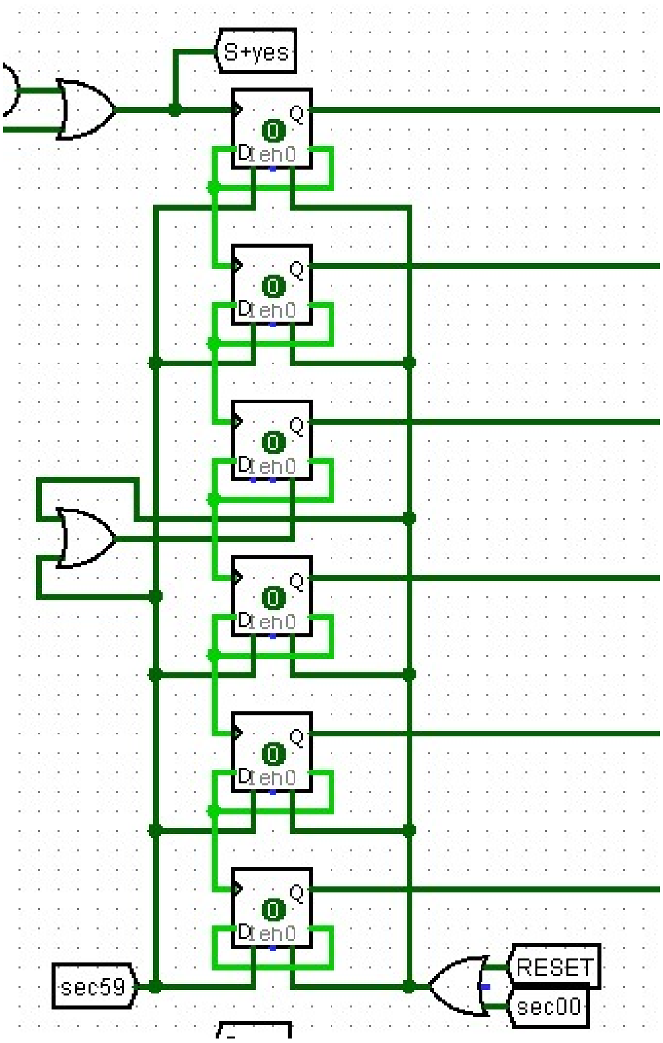


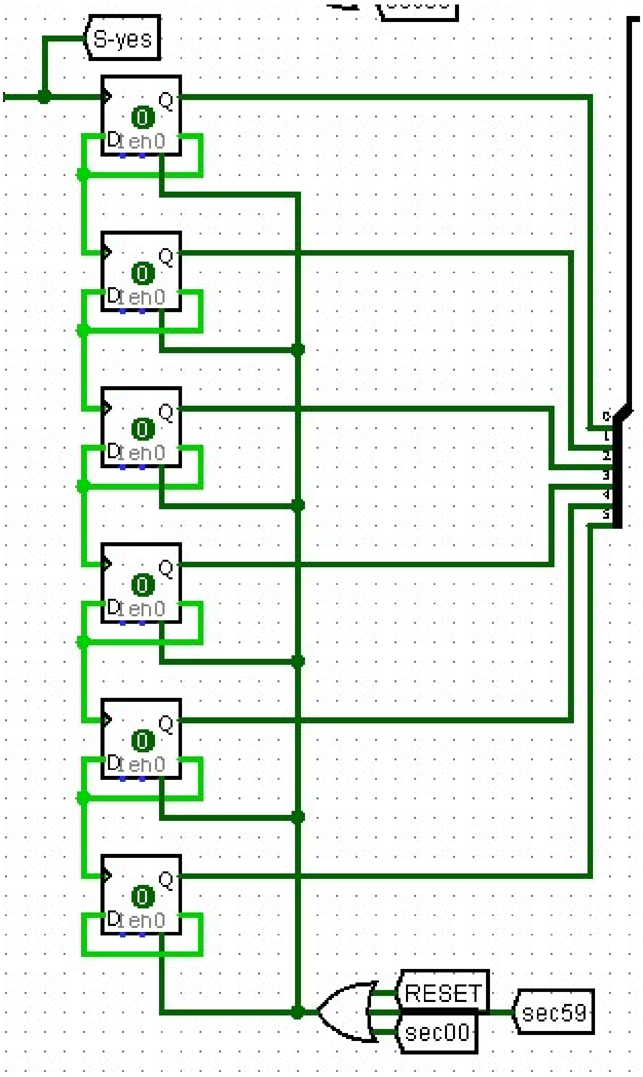
6. Reset Button

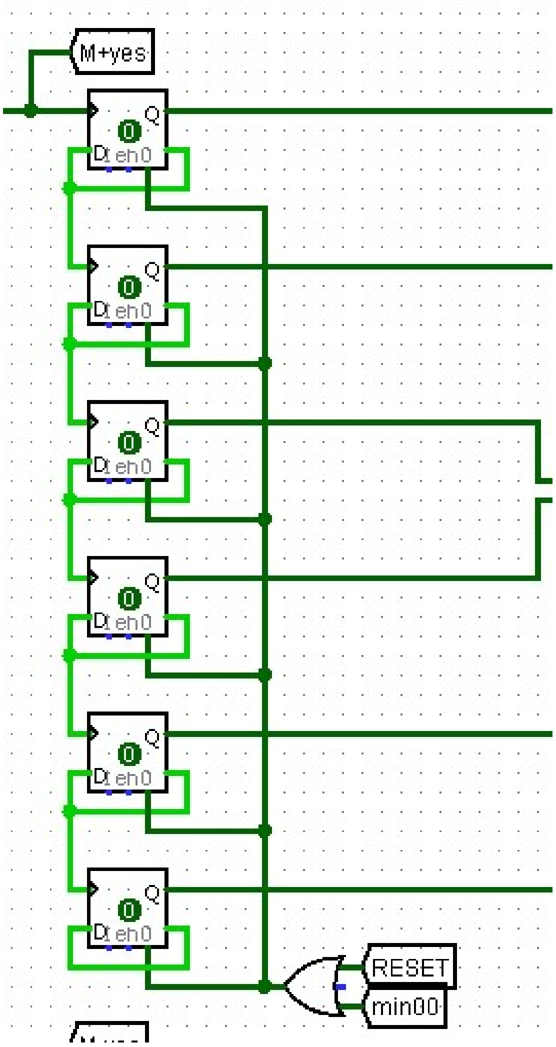
The reset button is connected to the reset input below:

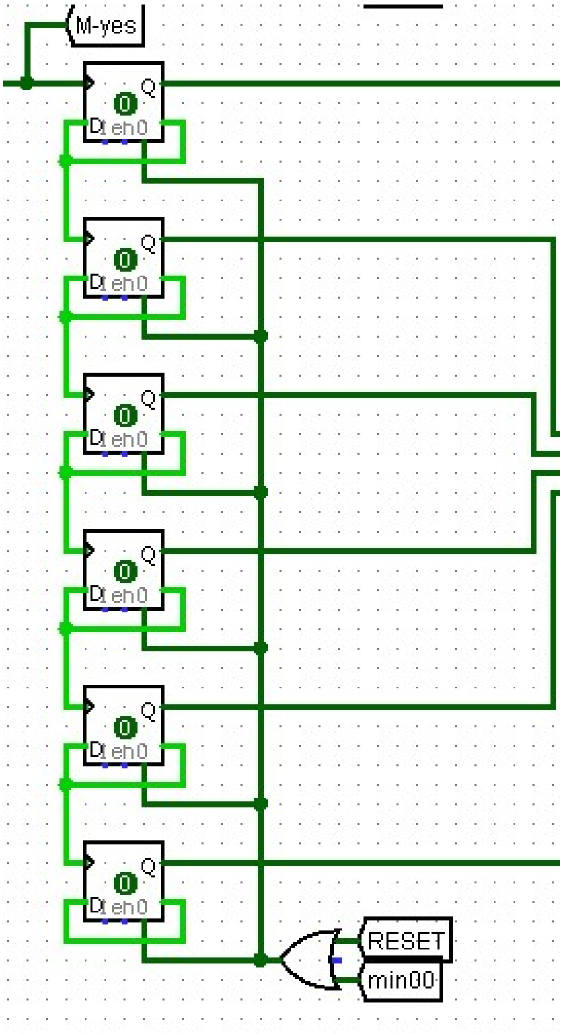


It connects to all 4 timers and resets them to their initial state as shown in these screenshots:



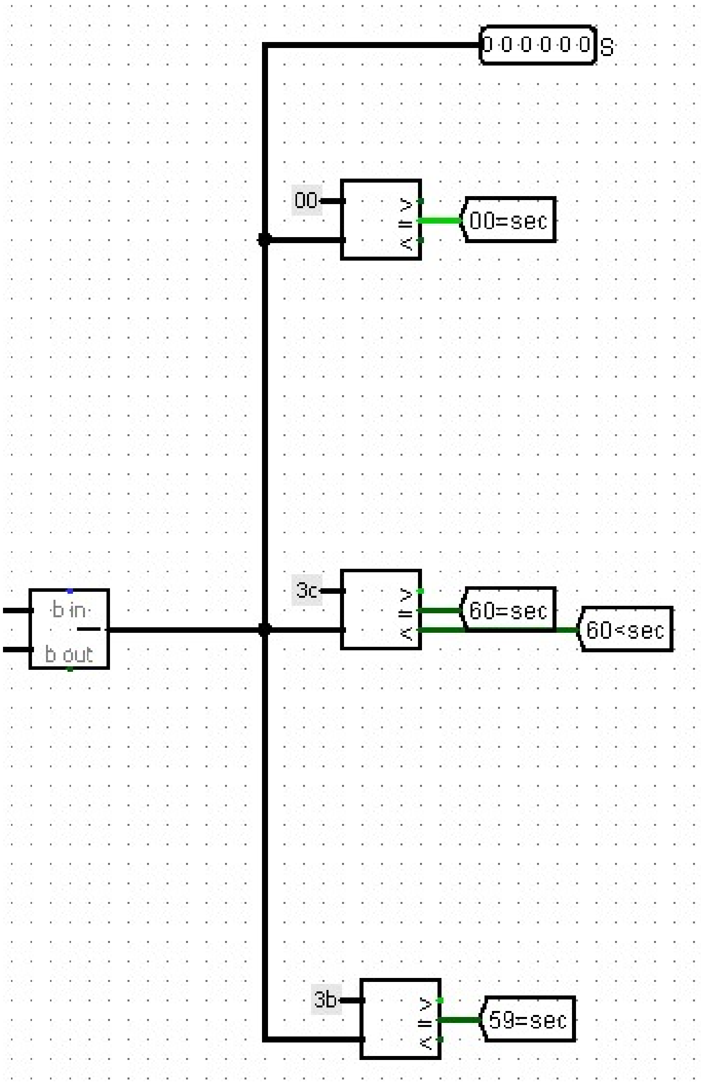

S-yes
Q
D
0
RESET
sec00
sec59
0
1
2
3
4
5

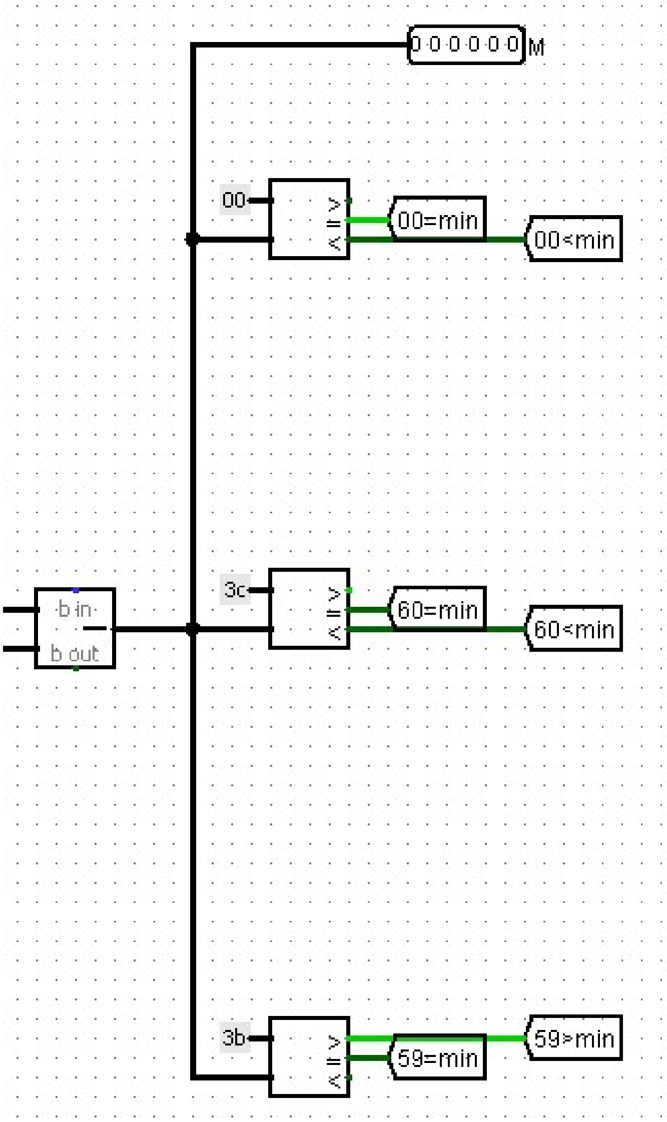

M+yes
Q
0
D
en
0
RESET
min00



Before proceeding to the other parts/processes, we need to explain how we will tackle edge cases in this circuit. We used comparators to know what the current value of the seconds and minutes are. In total, there are 6 comparators; 3 for seconds and 3 for minutes. The values being compared are already the difference between the timers as explained before. Most of the remaining explanations heavily rely on these compared values. Also, it is because of these comparators that we don't need to worry that 6bit can have values from 00 to 63 instead of 00-59.

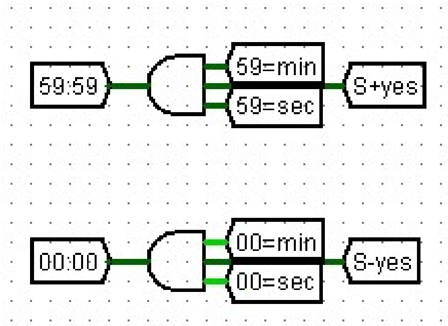
These are the comparators:





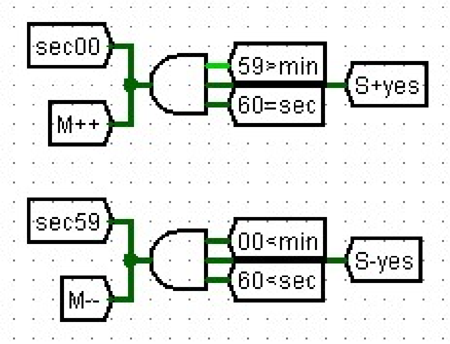
7. Edge cases of the whole timer (59:59 and 00:00)

When the circuit is 59:59 as the second was increased, the circuit will immediately pause. The same thing will happen when 00:00 was reached as the second was decreased.



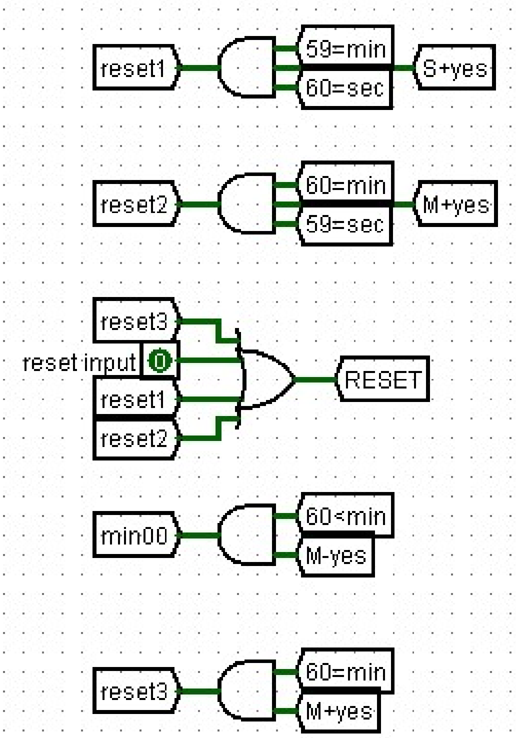
8. Edge cases between seconds and minutes

As shown below, when 59 is greater than the minute value and second value is equal to 60, the minutes will be increased by one and the seconds will be forced to be 00. In the same manner, whenever minutes is greater than 0 and seconds value is greater than 60, the minutes will decrease by 1 and the seconds will be given the value of 59. Let us clarify that the condition 60<sec does not literally mean that seconds is greater than 60. Take this as an example: 000000 – 000001 = 111111. Therefore, whenever 60<sec is true, it means that the second's value is already 00 and you are trying to decrease it even more.

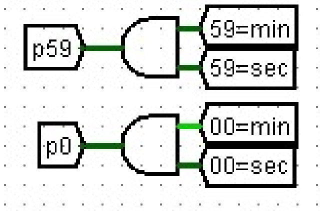


9. Other conditions

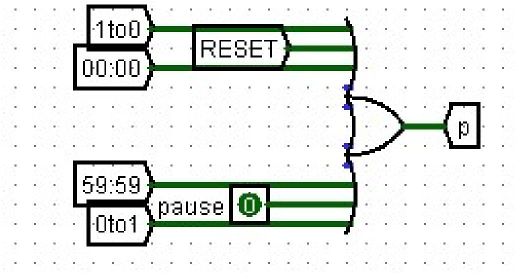
There are other several conditions needed to complete this circuit. These are relatively easy to understand. The "resets" are connected to RESET which will just force the entire circuit to have the value of 00:00. Reset 1 turns the minute and second components to 00:00 when it reaches 59:59 and the increment second button is used. Reset 2 turns the minute and second components to 00:00 when it reaches 59:59 and the increment minute button is used. Reset 3 turns the minute and second components to 00:00 when it reaches 59:XX, when the increment minute button is used. The min00 will just reset the minutes to 00 to stop a decrement to a minute that has value of 00.

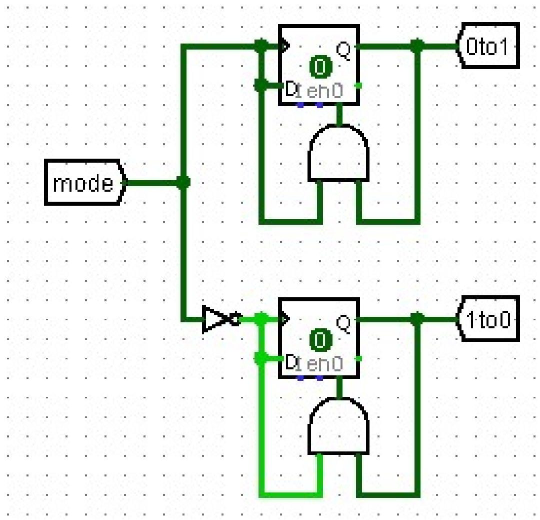


Remember that in the play and pause buttons, there are p0 and p59. These are the conditions for those. This disables the play button from being activated when the minute and second component has a value of 59:59 and the mode is 0, as well as when the minute and second component has a value of 00:00 and the mode is 1.



As said before, other conditions that will make the circuit paused will be explained. Almost all except for 2 are already discussed. The remaining conditions for pause are the changes in mode which are included in the 2nd picture below. Again, this was done by trial and error. The D flipflop needs its value to immediately become zero again after it becomes 1. Just the slightest change in the flip flop's value is enough to pause the circuit but it needs to be 0 again or else the circuit will always stay stopped. Simplifying things, whenever the mode is changed, it makes one of the flip flop's state to become 1. Its output will go to pause the circuit and to force the flipflop, itself to have a 0 state.





How the whole circuit works:

The circuit is made up off a play button, pause button, LED which indicates when the circuit is paused, reset button, mode selector, increment/decrement buttons for either the minute component or second component and four 7-segment displays to showcase the minute and second components.

When the mode selector is in 0 or the stopwatch mode, you can press the play button and then the circuit will begin to count up.

When the mode selector is in 1 or timer mode, you can press the play button then the circuit will count down, just make sure that the minute or second components have a value, since it would not work when its value is 00:00.

It can be seen that the pause LED is off when the circuit is playing. During this period, you are unable to utilize the increment/decrement buttons on either of the modes. The reset button can also be pressed anytime and it will pause the circuit and change the value of the minute and second components to 00:00. You can also press the pause button when the circuit is playing to stop the circuit from playing.

On another note, when the circuit is playing and the mode is changed, the circuit will come into a pause and when the play button is pressed again, it would either count up or count down depending on the mode change.