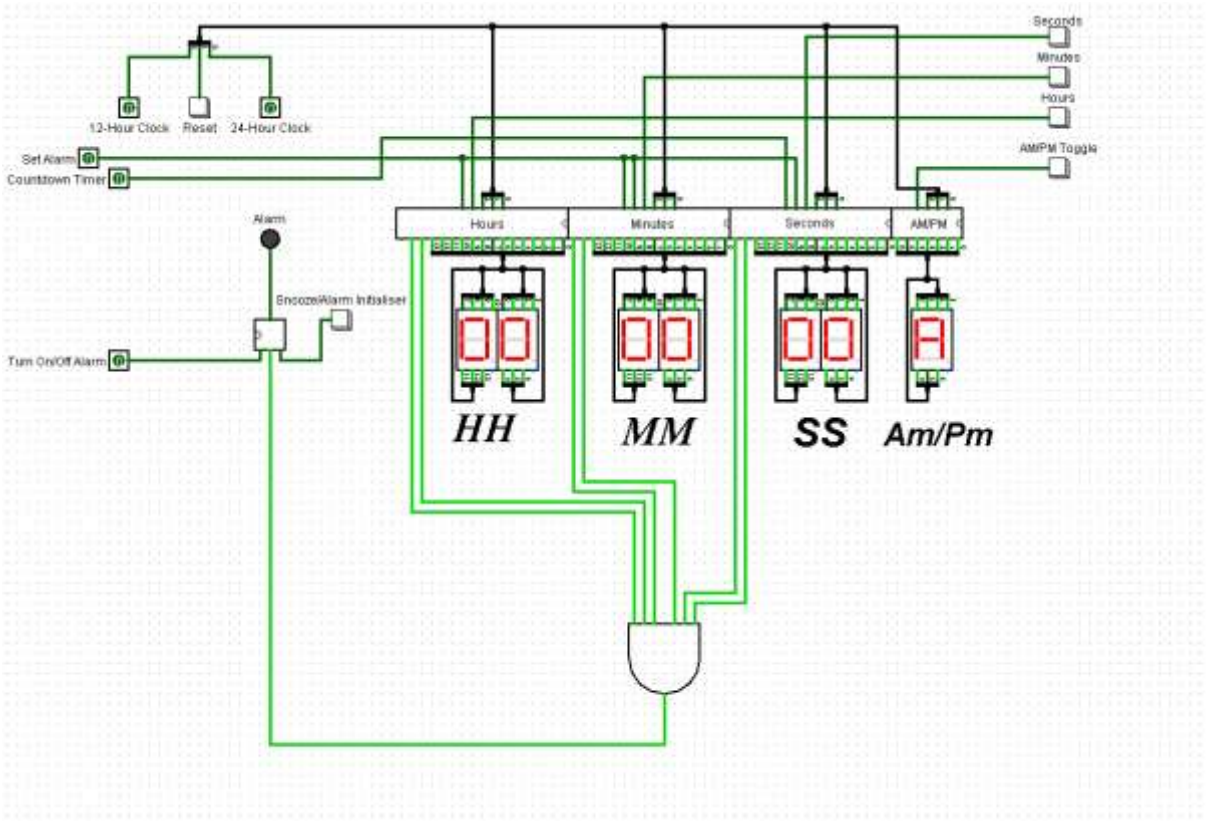
Homework 2-Digital Electronics

SN:19349771
Dami Abagun

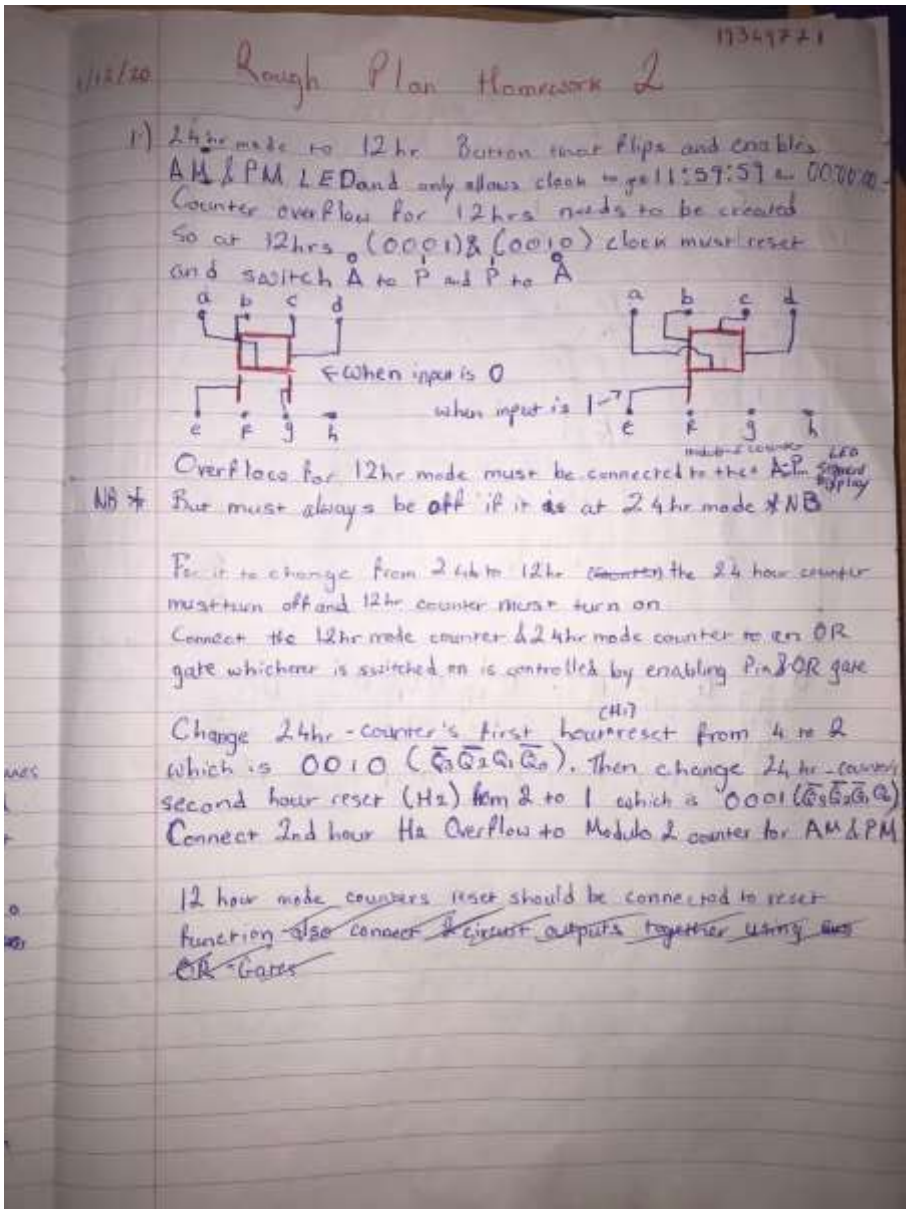
1.) Introduction

Within this lab I had to add extra functionalities within my clock. This was quite a tedious task to do, although it was very enjoyable. I had difficulty with creating the count-up/count-down timer at the end of the homework, but I believe the rest of the functionalities were created to a very good standard. Within this report I will discuss how I underwent creating a 24-hr and 12-hr Mode for the digital clock. I will also discuss the how I created an alarm and all the circuit components that included this. Within this report there are many screenshots of the circuits created in Logisim and also pictures of the main circuit, and some of my extra workings for the extra functionalities. This is a photo of the main Circuit with the fully functioning digital clock. Very similar to the original clock in which I created in lab 4, just with the extra functionalities.

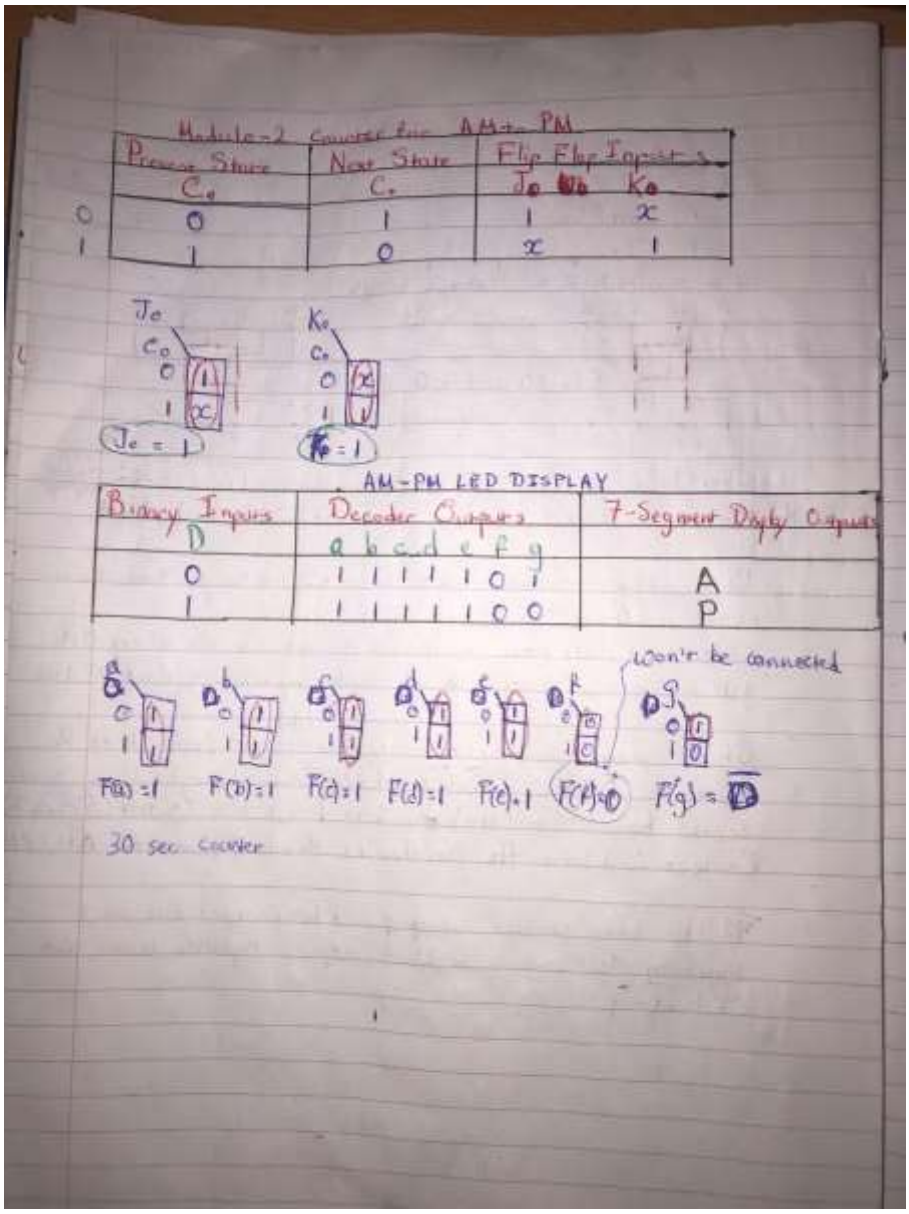


2.) Creating the 12-hour mode and 24-hour mode option.

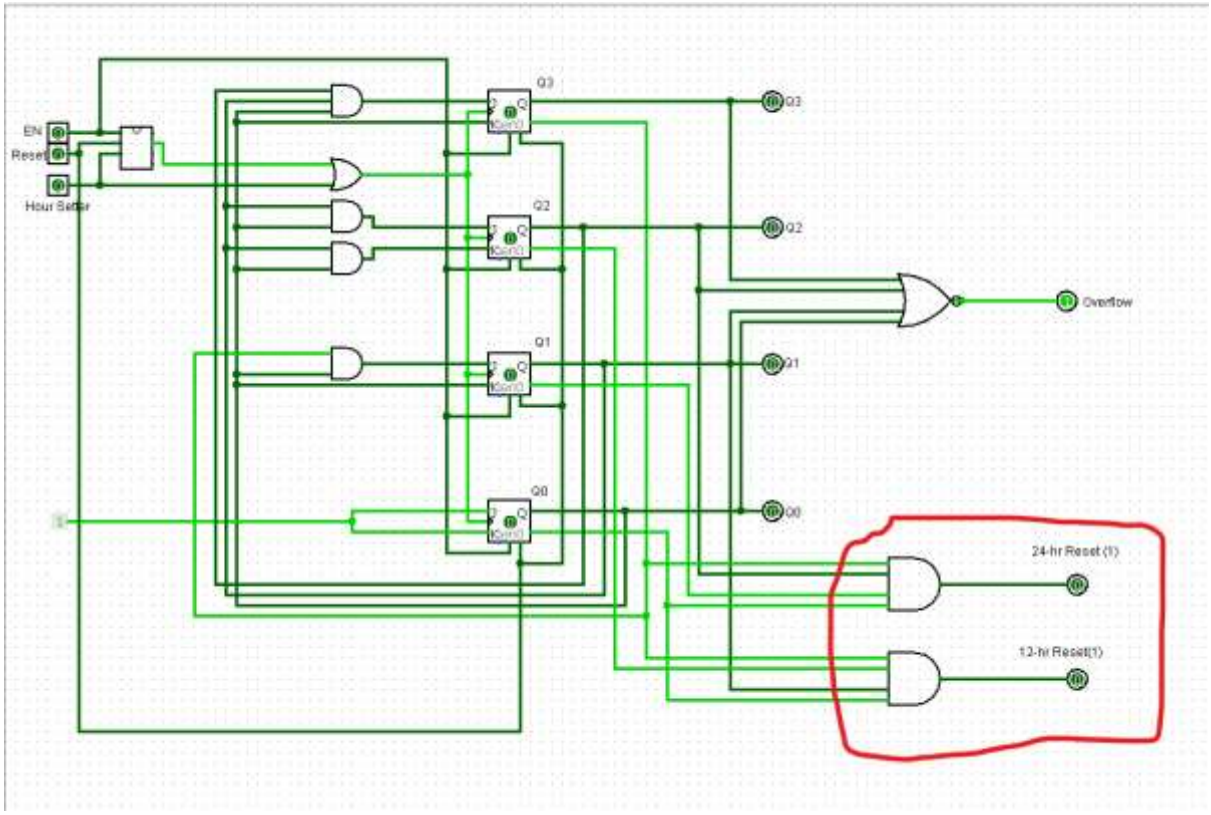
For the first part of the assignment I had to create an option for where the user would be able to choose between using the 24-hour mode and the 12-hour mode on the clock. Firstly before I began this part of the assignment I created a rough plan to have a rough idea of what I was going to do. The image of these plans are located below.

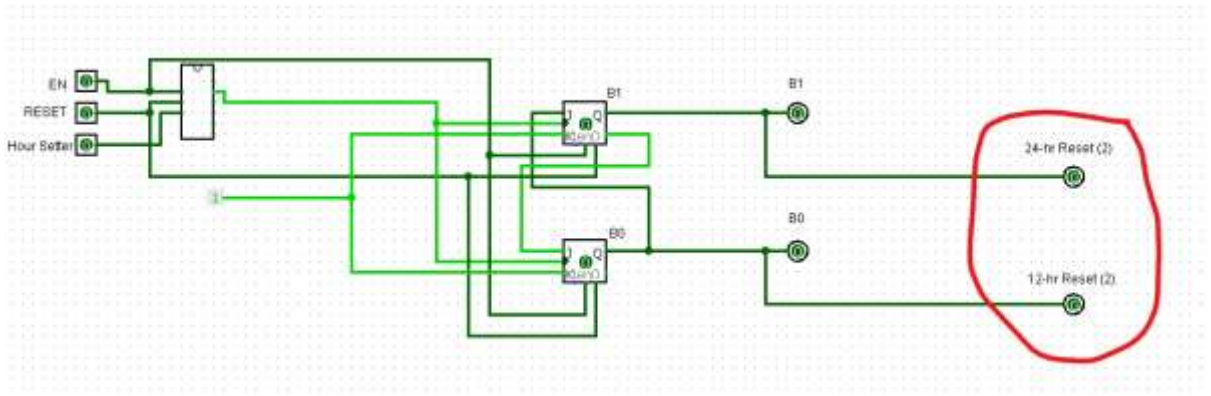


To have the clock stay in the 12-hr Mode I created extra reset functions to my Hour Digit circuits. I also created a function in the circuit that would allow the AM/PM LED to flip between AM and PM whenever it the clock reset after each 12-hour cycle. To create this function I created a modulo-2 counter that went from 0-1. So that anytime the clock reset the clock pulse in this counter would turn it between 0 and 1. 0 was equal to A(AM) and 1 was equal to P(PM). The photo of how I created such a function and its counter is below.

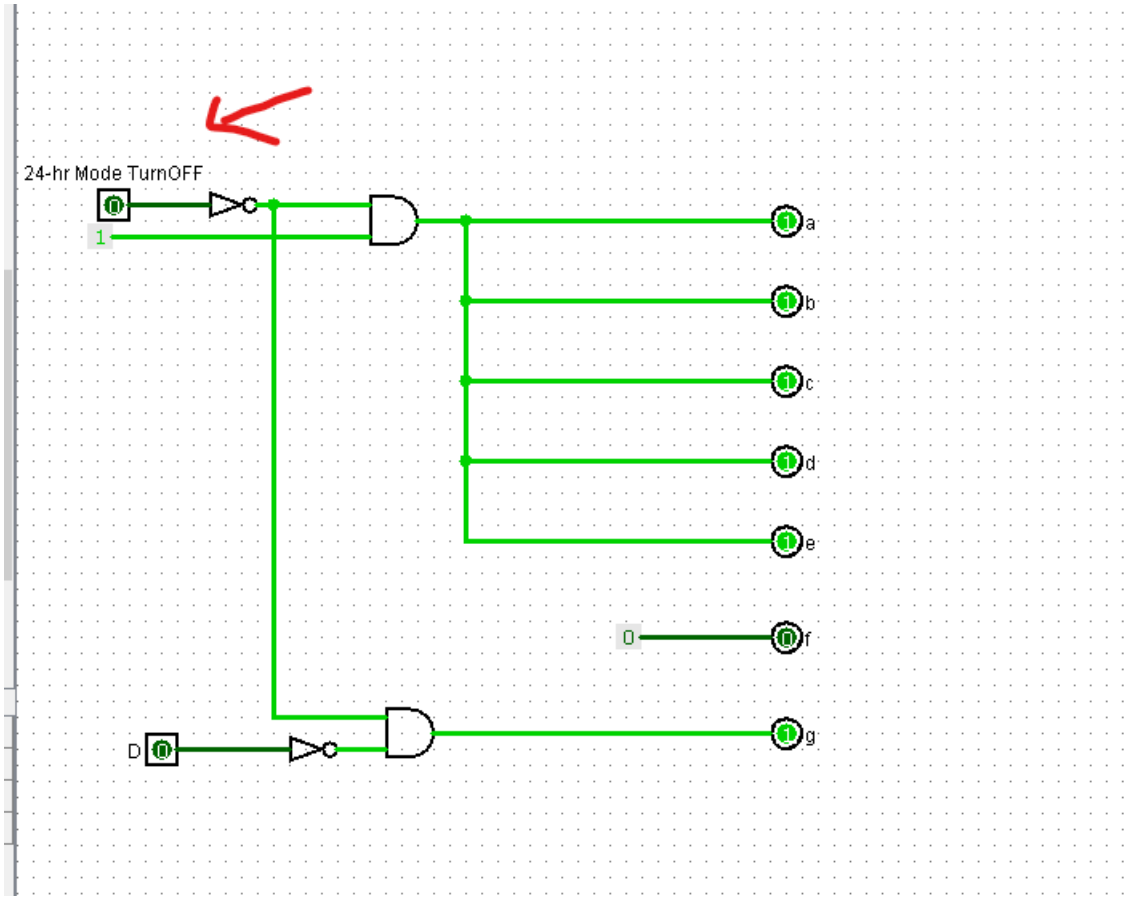


I also made it so that it would be turned off anytime the 24-hour mode would be turned on.

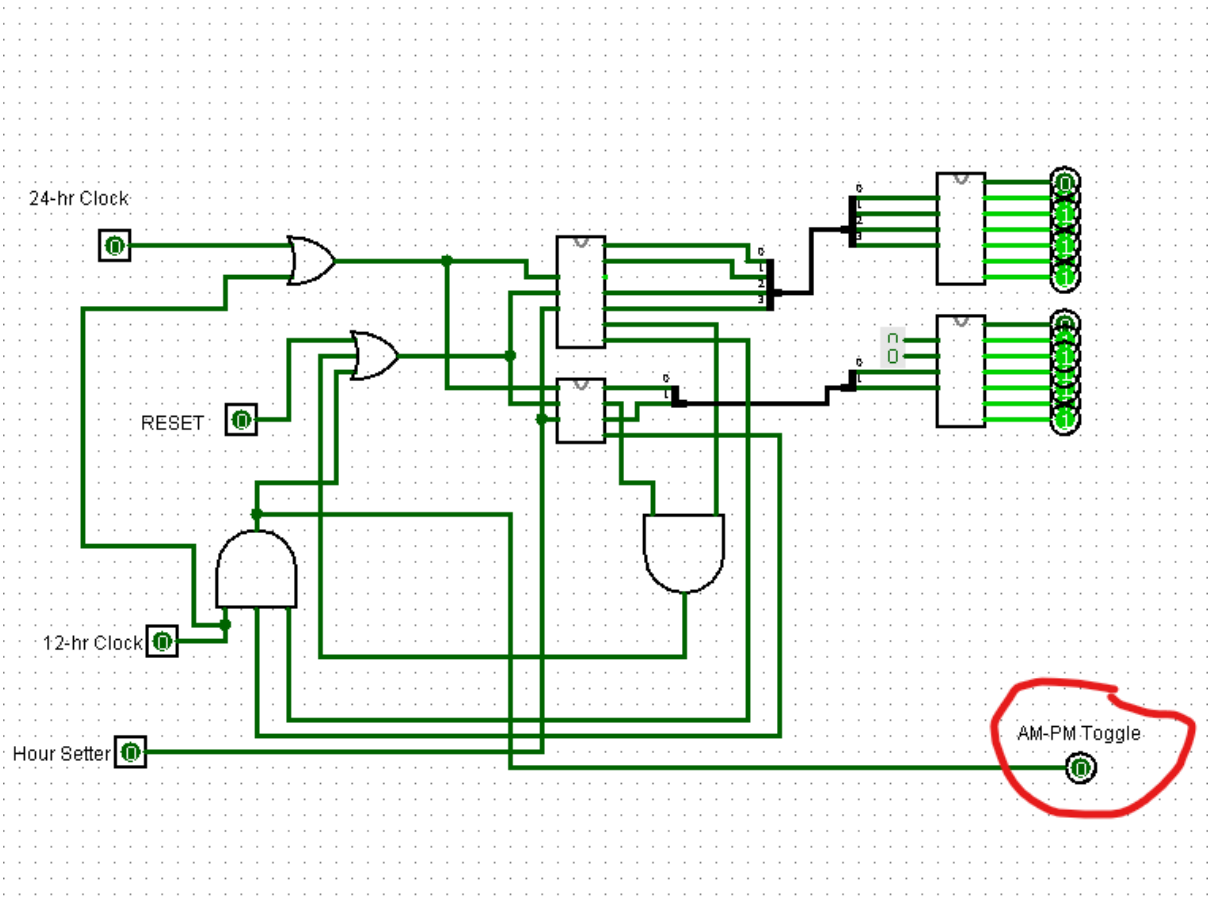




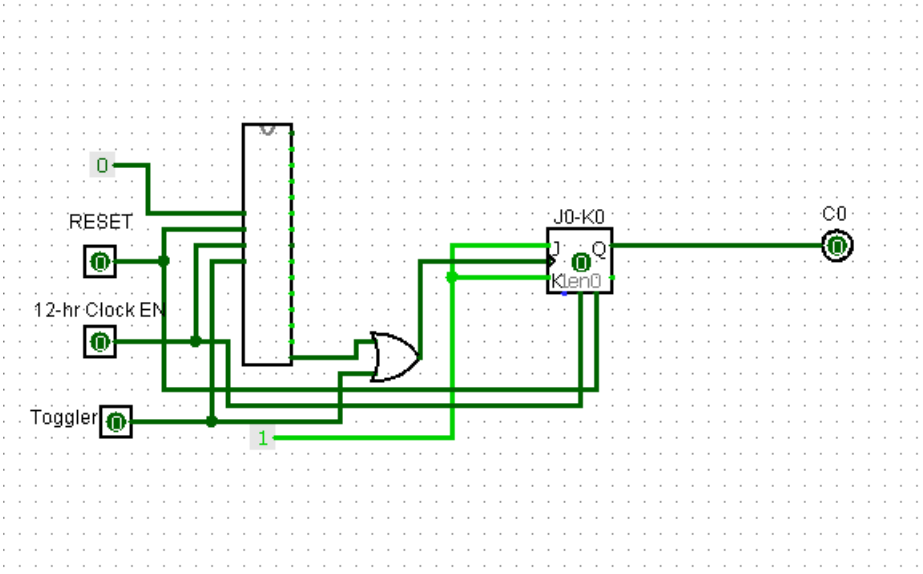
With these functions depending on which pin was turned on at the time the clock would reset. To create the AM/PM LED display I created this from using my Karnaugh map, truth table and logic functions workings. For the LED display the circuit looked like this.



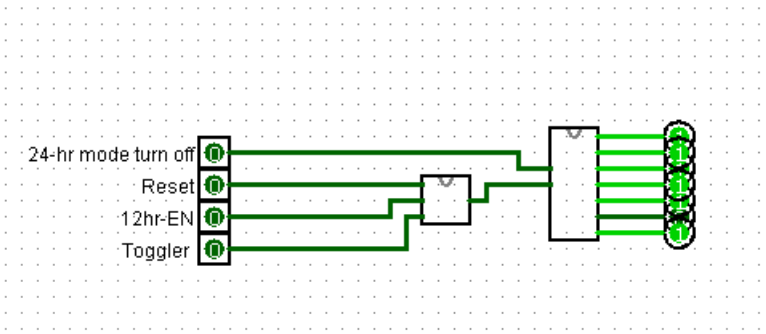
The arrow is pointing towards the function that allowed the 24-hour mode clock turn off the AM/PM LED.



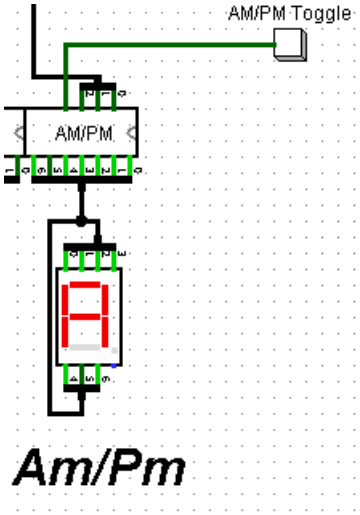
This meant that everytime the hour digits would reset the AM/PM LED would switch its state.



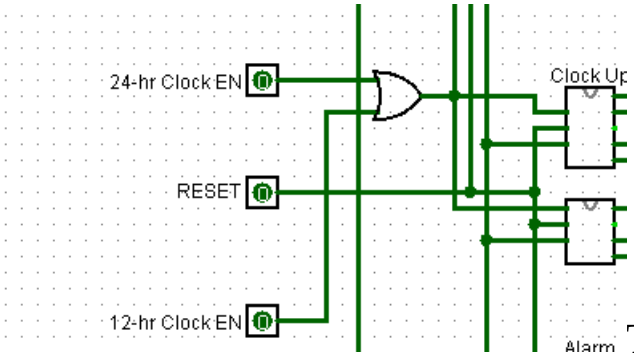
This is the counter circuit for the AM/PM LED. The AM PM toggle is connected to an OR gate and an AM/PM toggler that the user could operate so that the user can change it by themselves if they need to.



Here the counter is connected to the 7-Segment Display that will be connected to the LED display to display A or P. The inputs all correspond to the features they are enabled by.



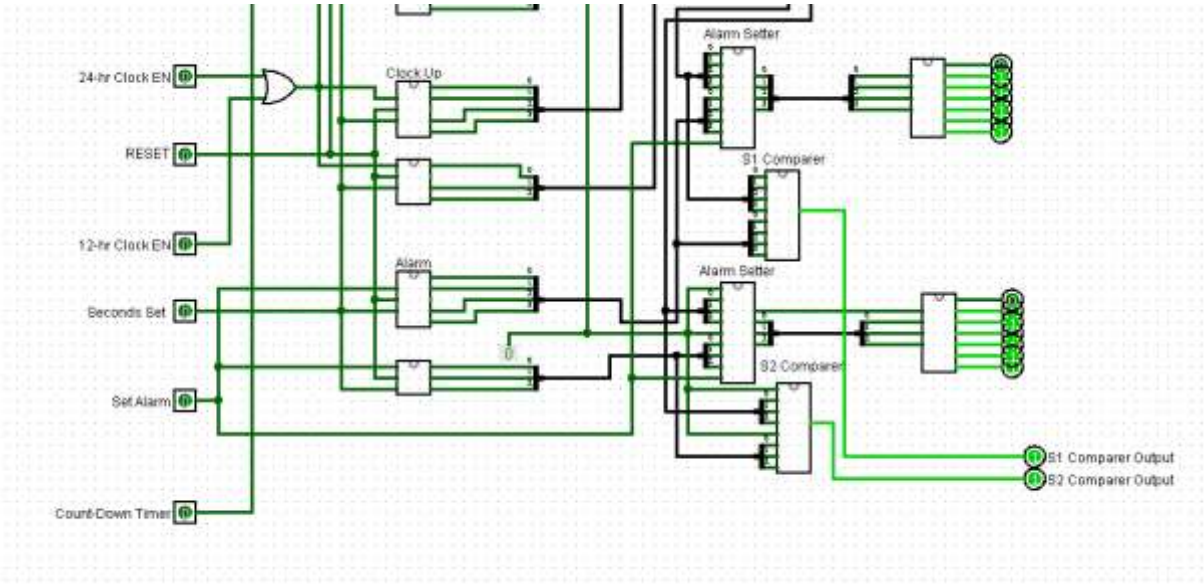
This the final look of the AM/PM function. With this the clock can be at 12-hour mode when the 12 Hour mode pin is clicked on, and the 24-hour mode pin is off. When the 24-Hour mode is on it will not turn on. Also within the seconds minutes and hours subcircuits I added extra input pins so that they would all be able to be affected by the 12-hr mode.



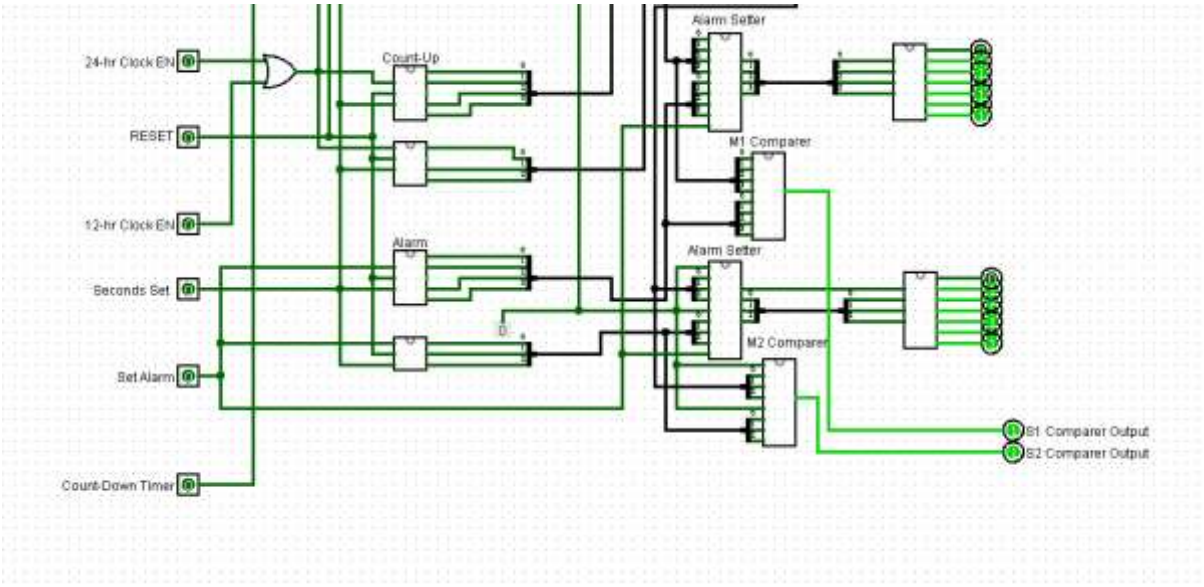
The OR gate that is there makes sure the clock will be enabled when the 24hour mode pin is on, or when the 12hour mode pin is on.

3.)Alarm Feature

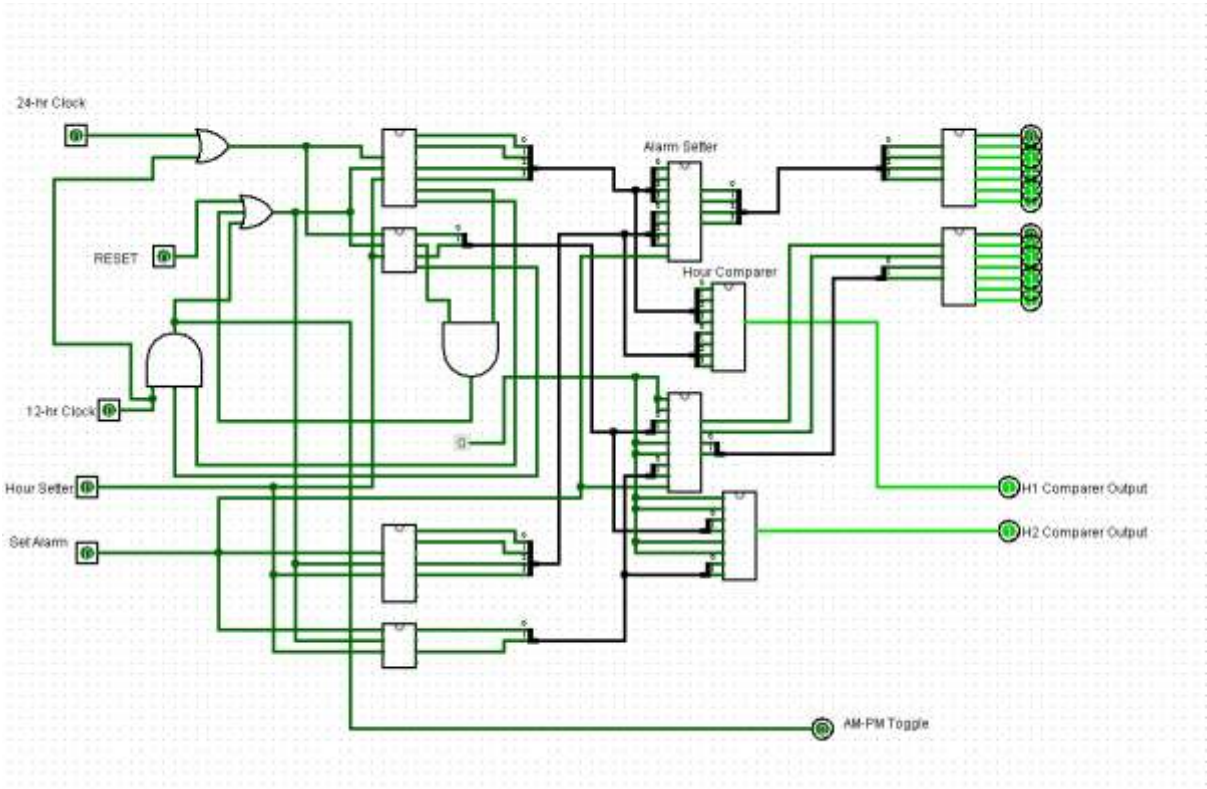
In creating the alarms the circuits and final subcircuits had to be greatly adjusted when compared to the final subcircuits and circuits form the last lab. Firstly I will show you the screenshots of the Seconds, Minutes and Hours subcircuits which include counters corresponding to their right digits (HH:MM:SS).



Seconds with Alarm functions.



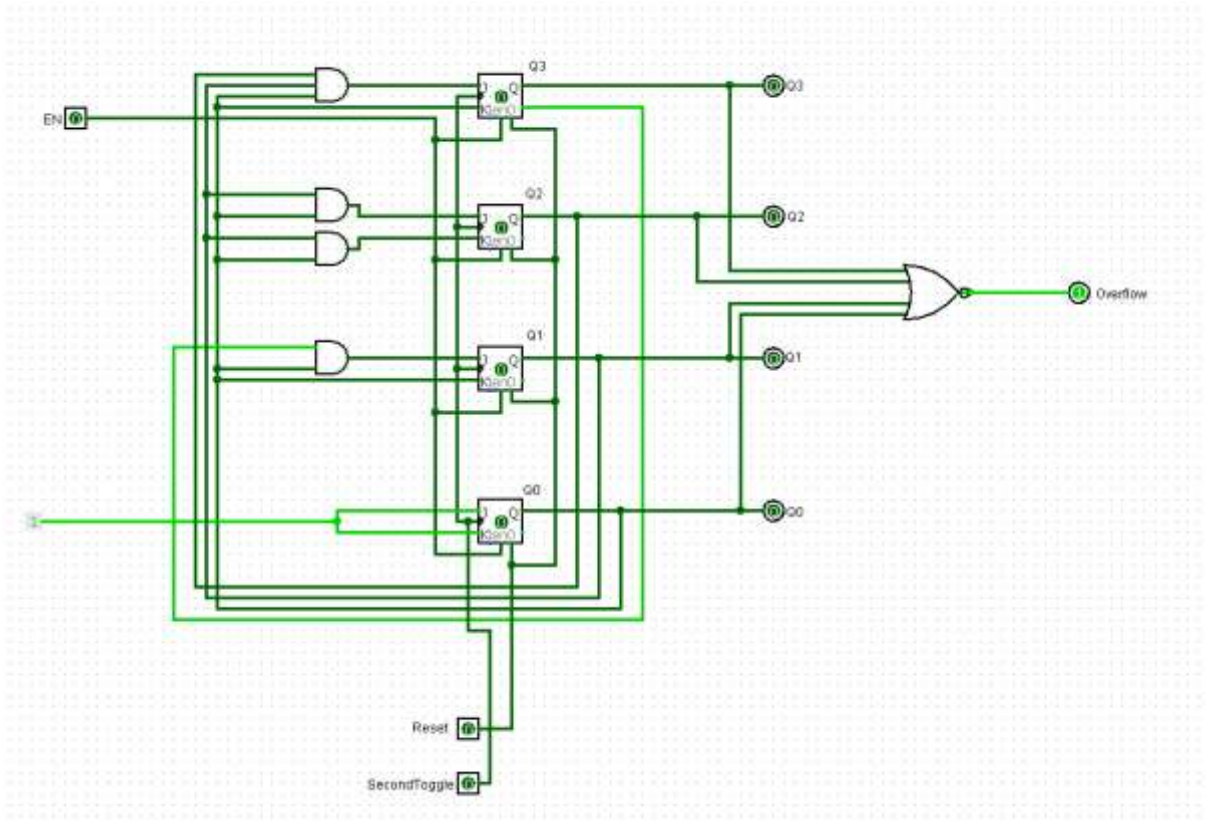
Minutes with Alarm functions.



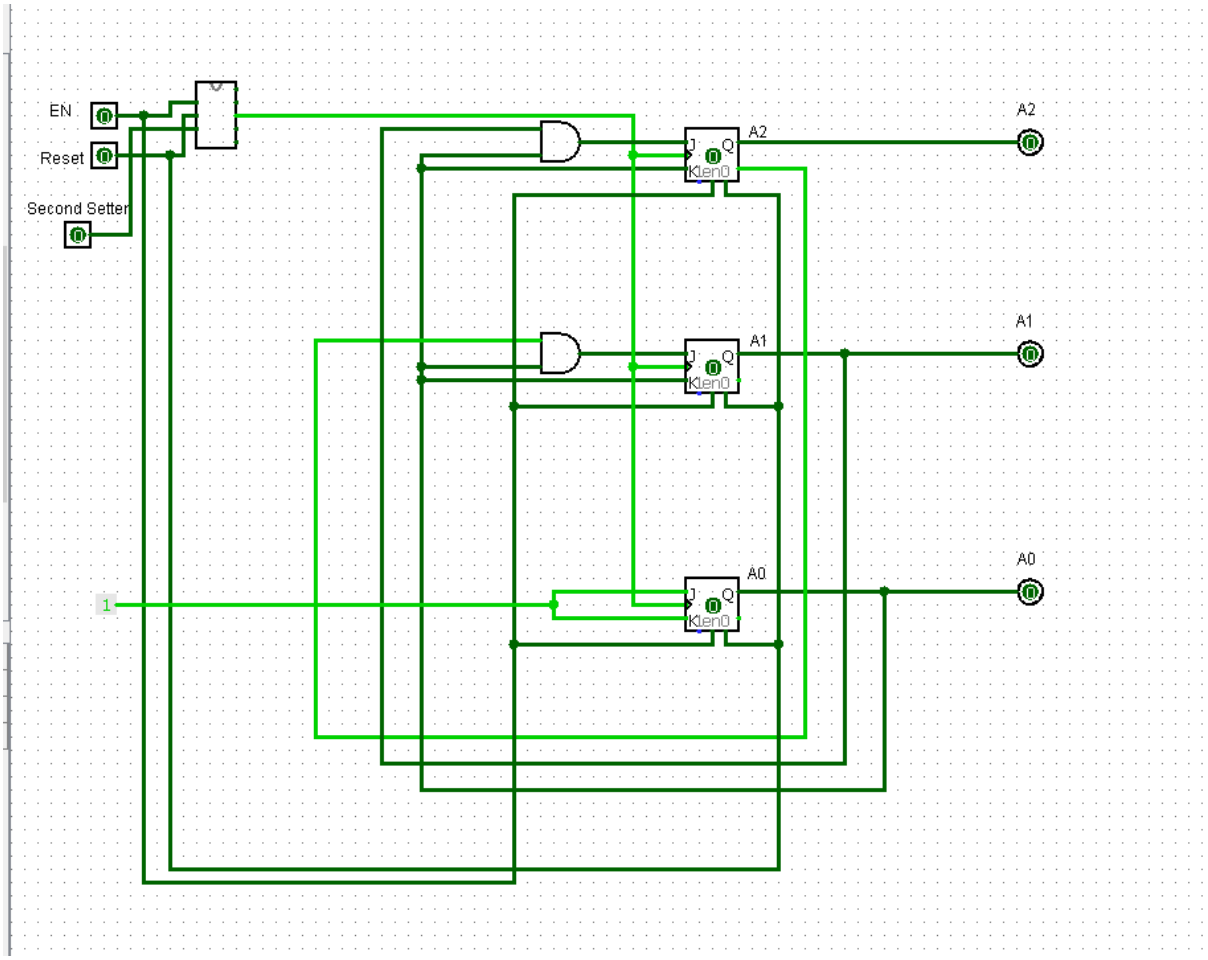
Hours with Alarm functions.

Firstly I created new counters for Seconds, Minutes and Hours for the alarm setting. With these counters the second digit of the Hours, Minutes and Seconds did not have an overflow in order for the user to

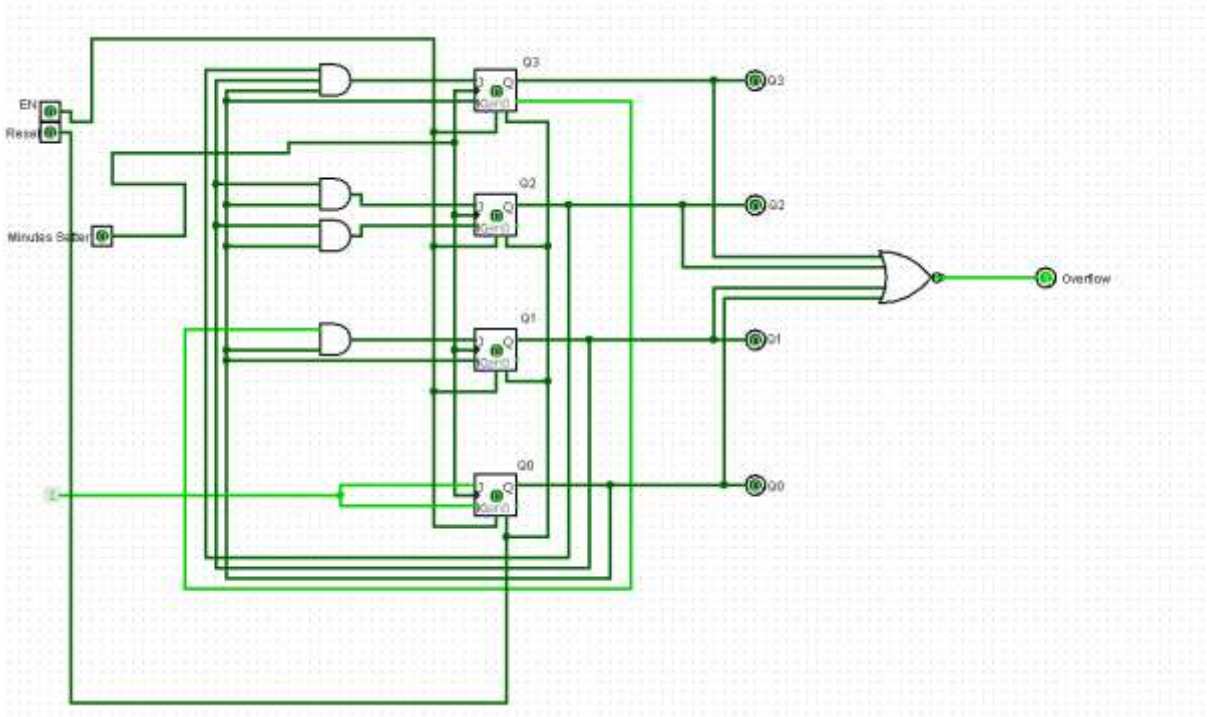
not experience any fault when inputting the time they want the alarm to flash the LED for. Screenshots of these circuits are below.



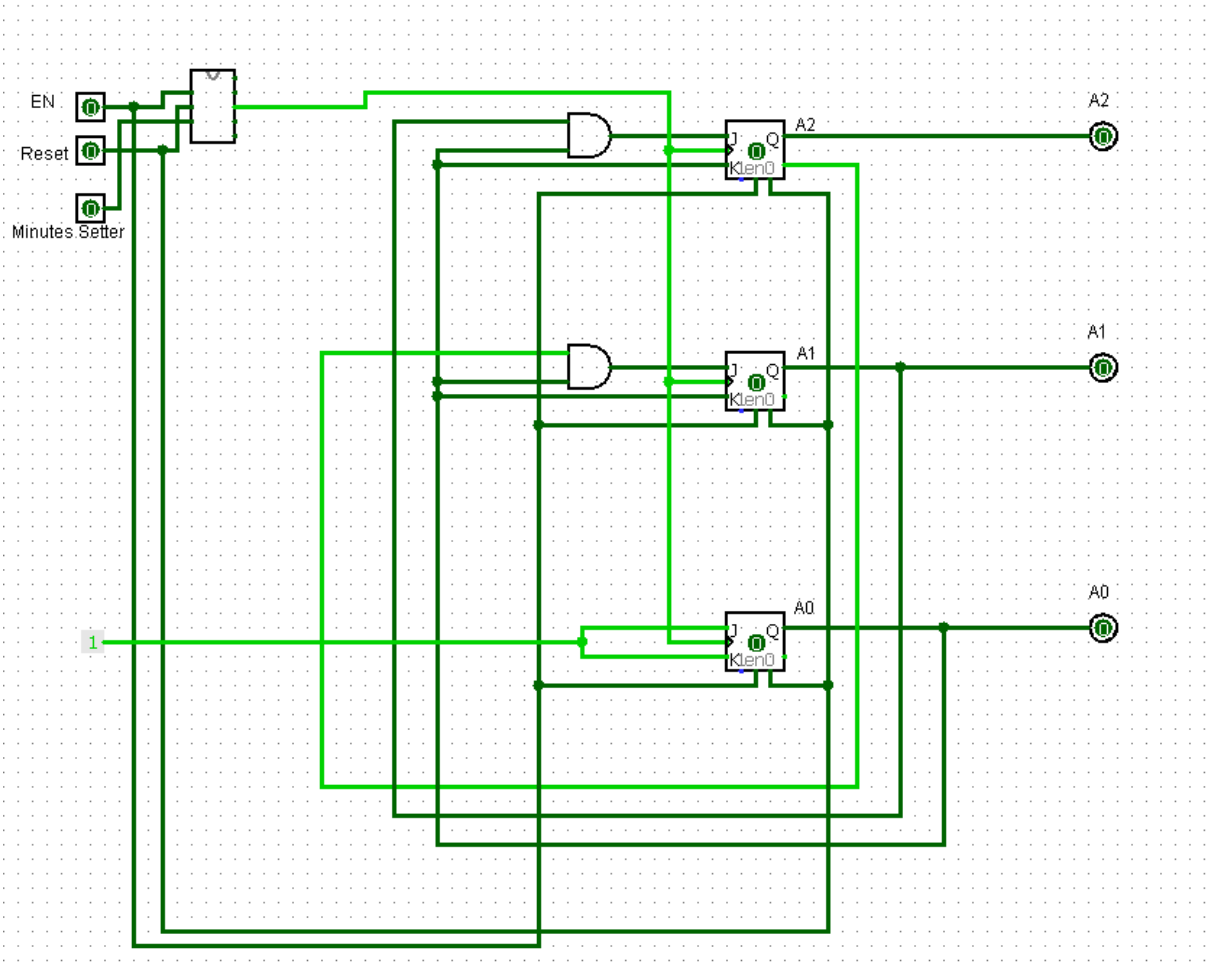
First Second.



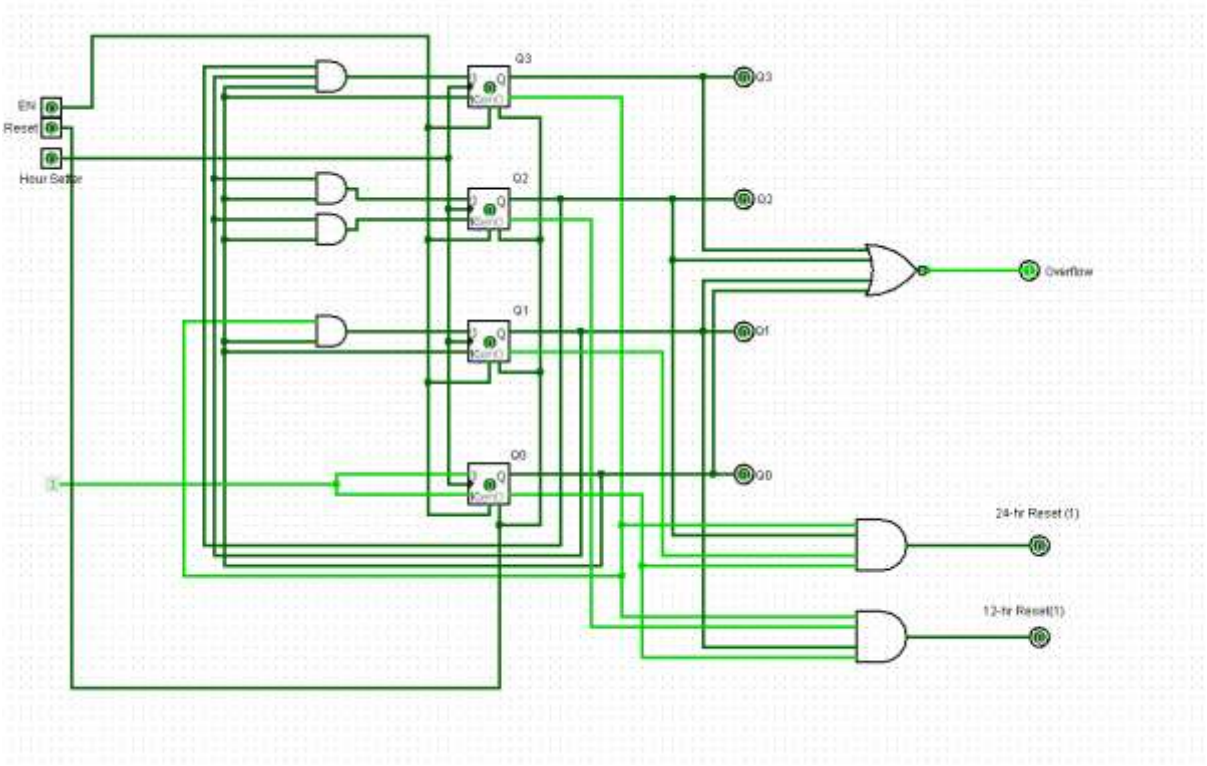
Second alarm seconds digit.



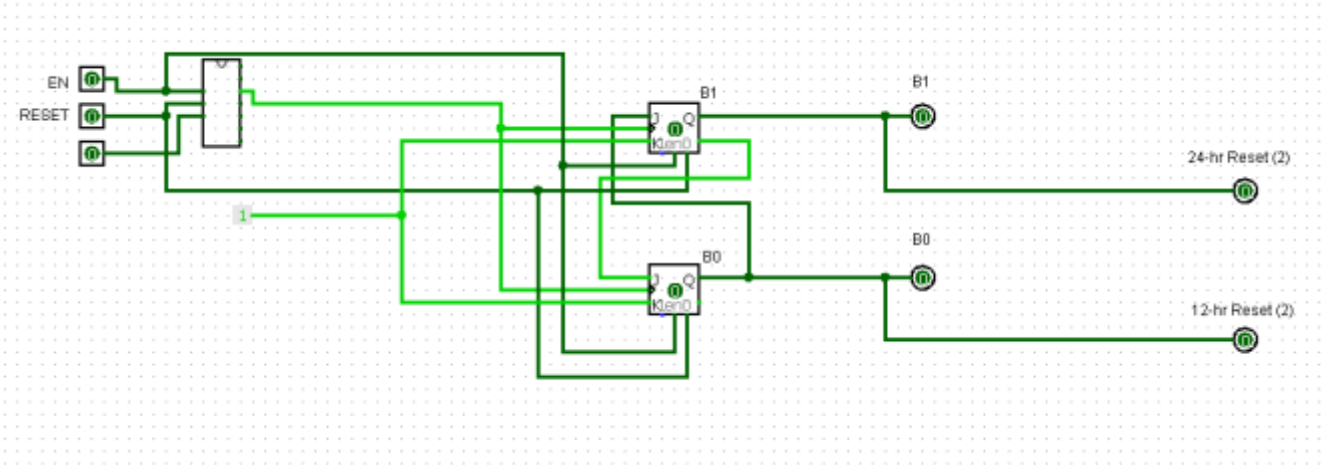
First Minute.



Second Minute.



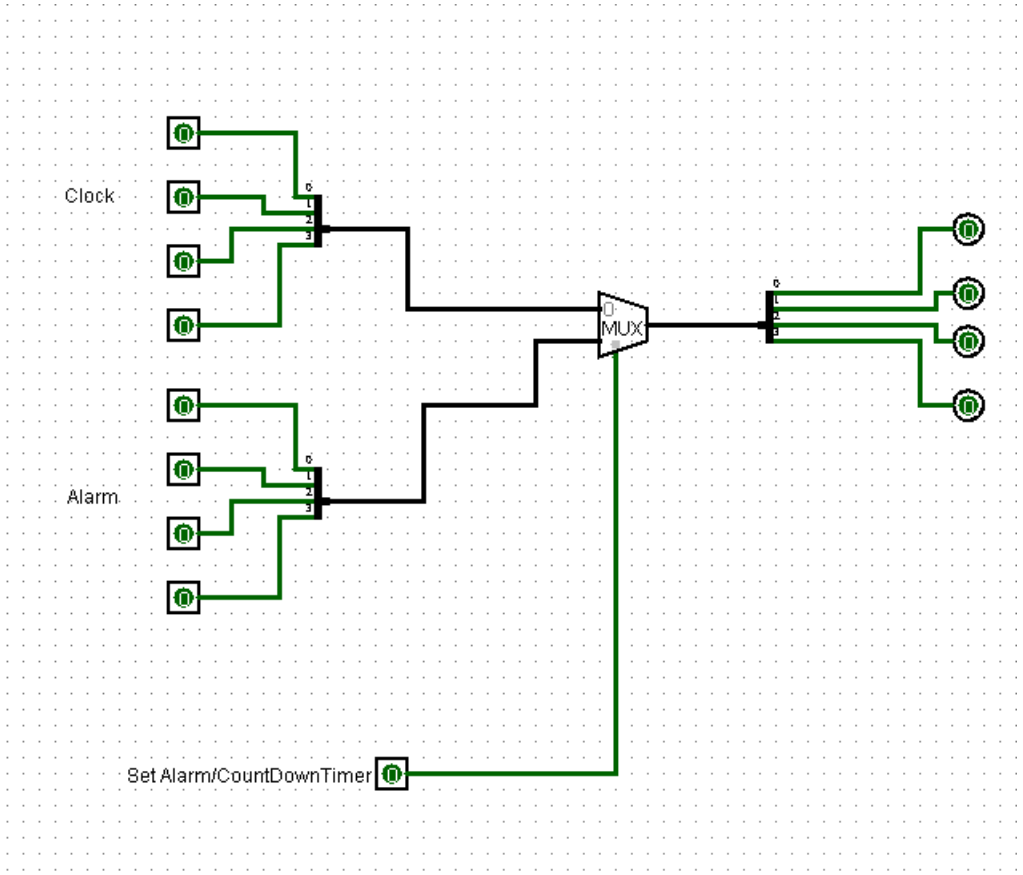
First Hour.



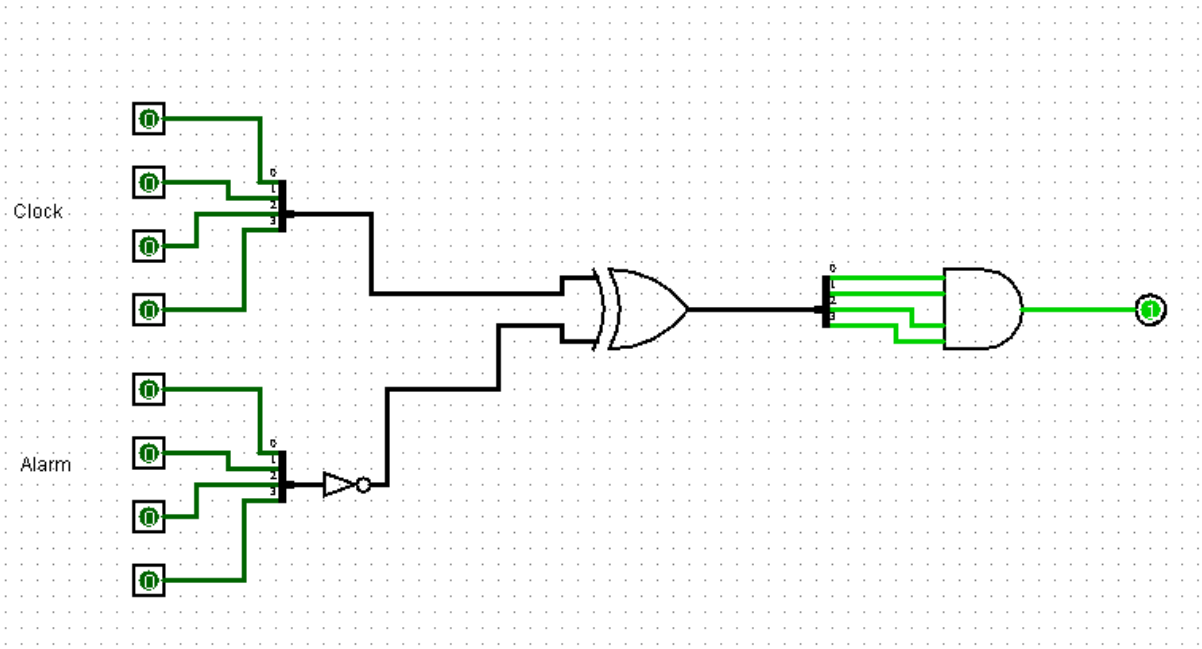
Second Hour.

I added an extra pin in the every circuit. This pin's function is to enable the user to increase the digits in the digital clock and pick a time. This pin was connected to an OR gate as an input with a clock being connected to this OR gate also. This meant that the clock of the circuits would toggle if the user pressed the button connected to the toggling pin, increasing the time enabling the user to input a specific time. I also added this pin to my original circuits from the last lab.

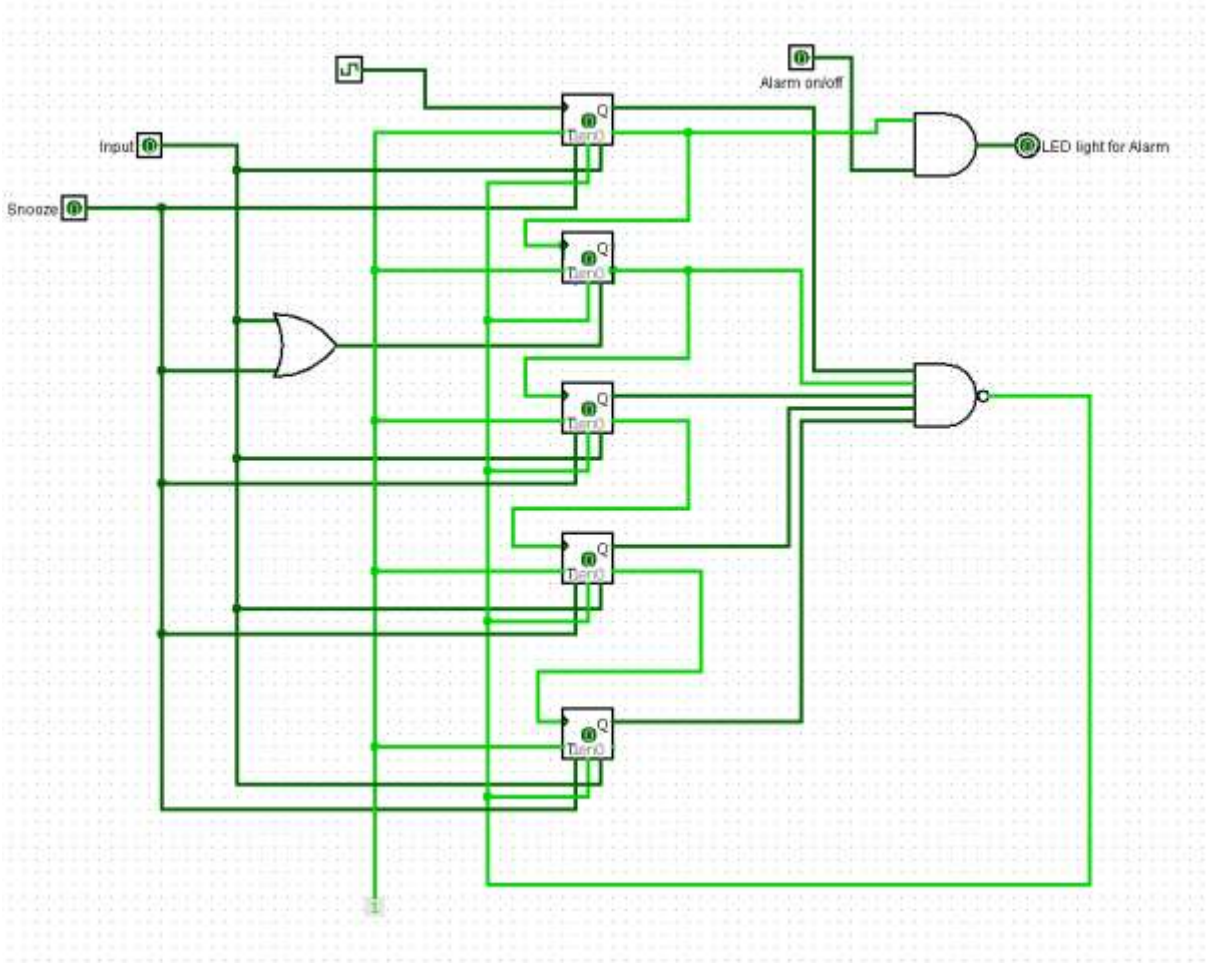
For the alarm function I used these circuits and a combination of multiplexer subcircuits and a digit comparing circuit that consisted of XOR gates and inputs from the counters for the digital clock. The way this worked is that a set alarm switch was connected to the multiplexer switch in the subcircuit containing the user alarm inputs and the digital clock inputs. In order to switch the display between the setting of the alarm time and the users input alarm time, the set alarm pin had to be turned on in the main circuit. The with the buttons connected to the digit togglers that would increment the time each time they are pressed the user would be able to input a time that would be stored there after the set the alarm and turn off the set alarm pin. The comparing circuit used an XOR gate and a NOT gate in order to compare the digits of the users input time and the current time which is flowing. If the 4 bits inputs/number matched with the one inputted by the user then the Alarm LED would flash for thirty seconds. This happens as the output of the comparer turns to and enters as the input for the 30-second counter which I made for the alarm. When this occurs the counter works and produces an output towards the LED allowing it to flash momentarily for thirty seconds. All circuits used within the creation of the alarm are located within the screenshots down below.



(If set alarm switch is at 0 normal clock inputs will be displayed on LED display, but if it was at 1 the multiplexer makes sure to display the clock outputs on the LED display, mainly for the user to input their choice of time.)



Comparer compares the bits of the clock and the alarm if they are equal to each other comparer output will produce a 1. The comparer outputs are put together in one AND meaning only when all outputs are true/1 then the alarm will flash as the desired time has been passed.



30-second counter for the alarm LED to flash for thirty seconds input comes from the AND gate that joins together the comparer outputs. There is also a snooze/alarm initialiser button that can be turned on to turn of the alarm manually. To prevent the alarm from turning on when both clocks are at 00:00:00 the user must set the alarm without turning it on first, after their time is set they must switch on the alarm and press the initialise/snooze button before turning on their clock.

4.) Countdown Timer

To create the count up/count down timer, I believed that my original clock was already a count-up timer that could be set to any time and be changed accordingly. In my attempt I decide to make a count-down clock with the counters I originally had all counting down. I then decided to join these counters with my clock that counted up where the user could choose whether or not to use the Count-up or Count-down clock. I put the outputs of each clock's digit into a circuit with a multiplexer and a switch. I believed that the switch would allow the user to switch between both clocks/timers. However it only allows the user to switch to a count-down timer that will countdown with no regards or relation to the Count-up timer/clock. I didn't know how to fix this error so instead of wasting time I just stopped including the function at the minutes part of my circuit. I created the Count-down counters for each digit of the clock by creating Modulo-10 9-0 synchronous counters, modulo-6 5-0 synchronous counters and a modulo-3 2-0 synchronous counters.

All corresponding attempts information and configuration of circuits using this information are all in screenshots below with the final subcircuit of Seconds, Minutes and Hours showing the attempt to make this timer with thorough care.

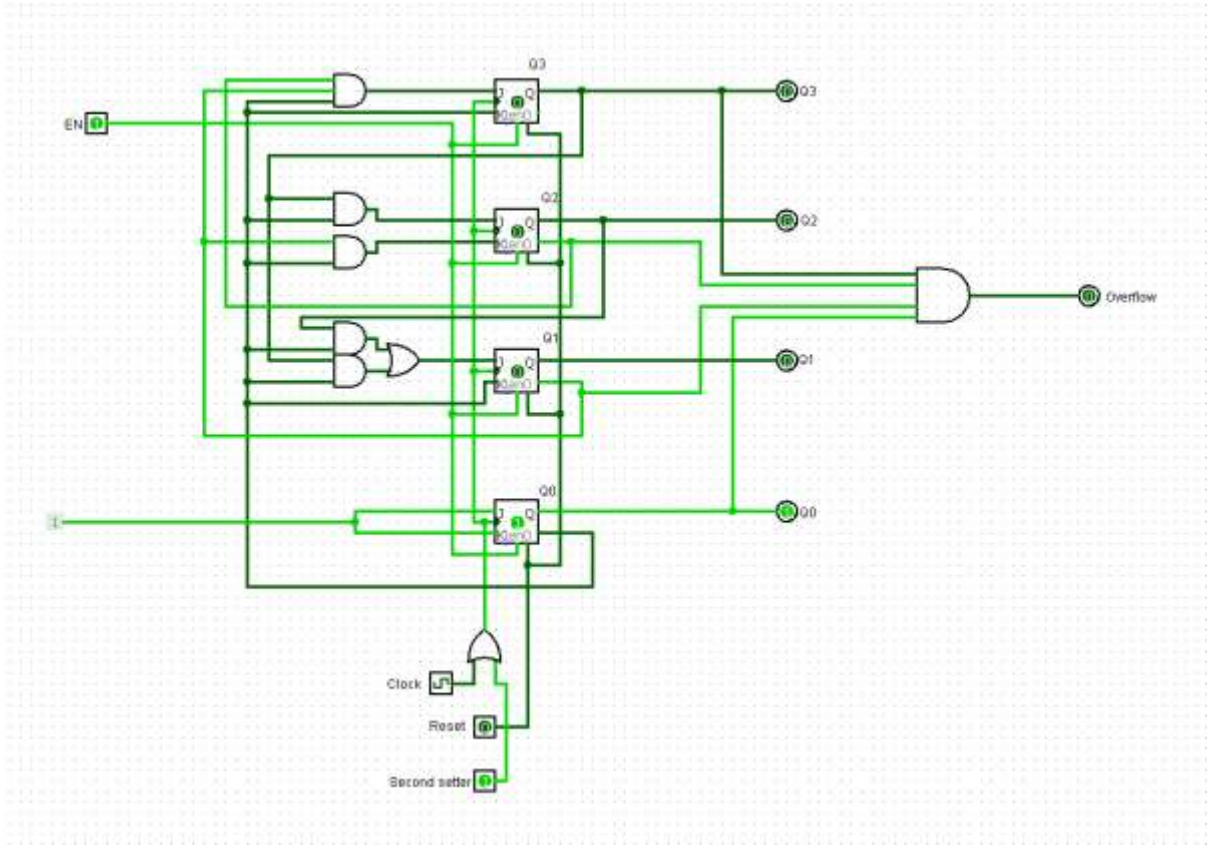

For Countdown Timer
Module 10- Counter Timer
Present State
Next State
Flip-Flop Inputs
State Diagram
$J_3 = Q_2\bar{Q}_1\bar{Q}_0$
$J_2 = Q_3\bar{Q}_0$
$J_1 = Q_2\bar{Q}_0 + Q_3\bar{Q}_1$
$J_0 = 1$
$K_3 = \bar{Q}_0$
$K_2 = Q_1\bar{Q}_0$
$K_1 = Q_0$
$K_0 = 1$

~~Theorem 3: Unused States~~

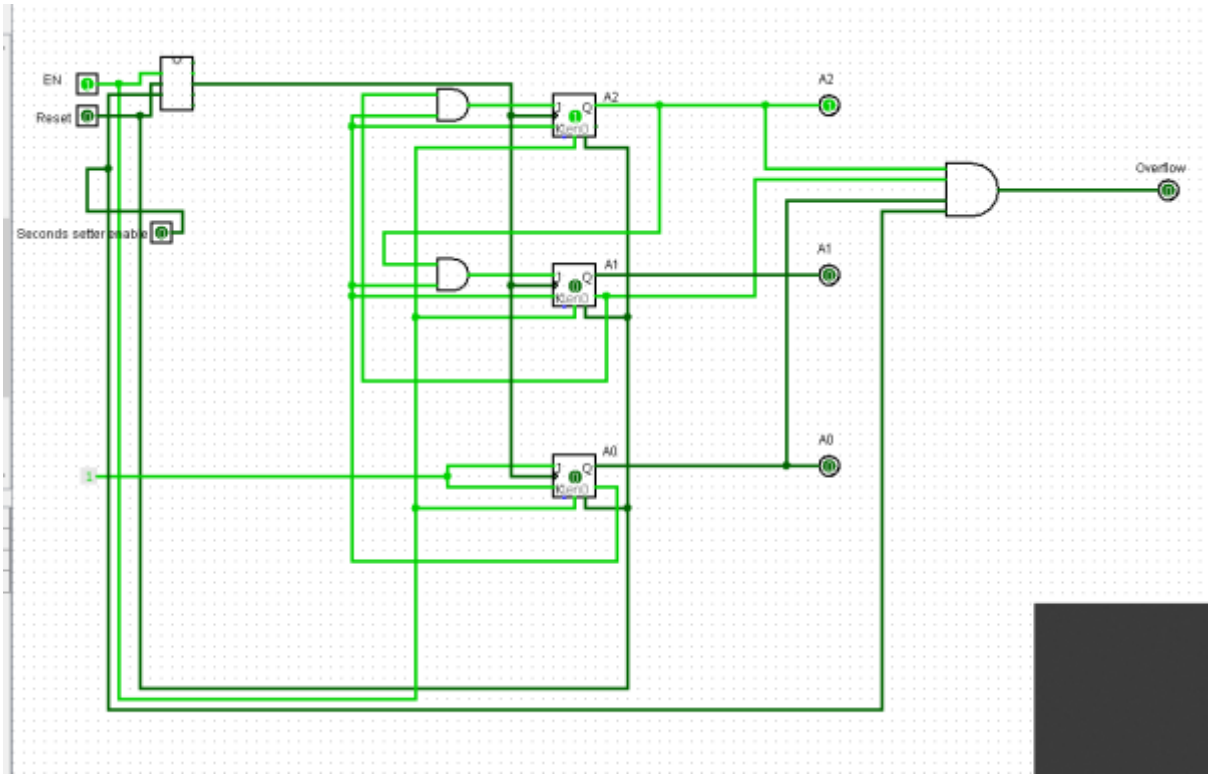
Checking Unused States:

	Present State				Next State				Flip-Flop Inputs					
	Q_3	Q_2	Q_1	Q_0	Q_3	Q_2	Q_1	Q_0	J_3 K_3	J_2 K_2	J_1	K_1	J_0	K_0
15	1	1	1	1	1	1	1	0	0 0	0 0	0	0	1	1
14	1	1	1	0	0	1	0	1	0 1	1 0	1	1	1	1
13	1	1	0	1	1	1	0	0	0 0	0 0	0	0	1	1
12	1	1	0	0	0	0	1	1	0 1	1 1	1	1	1	1
11	1	0	1	1	1	0	1	0	0 0	0 0	0	0	1	1
10	1	0	1	0	0	1	0	1	0 1	1 0	1	1	1	1

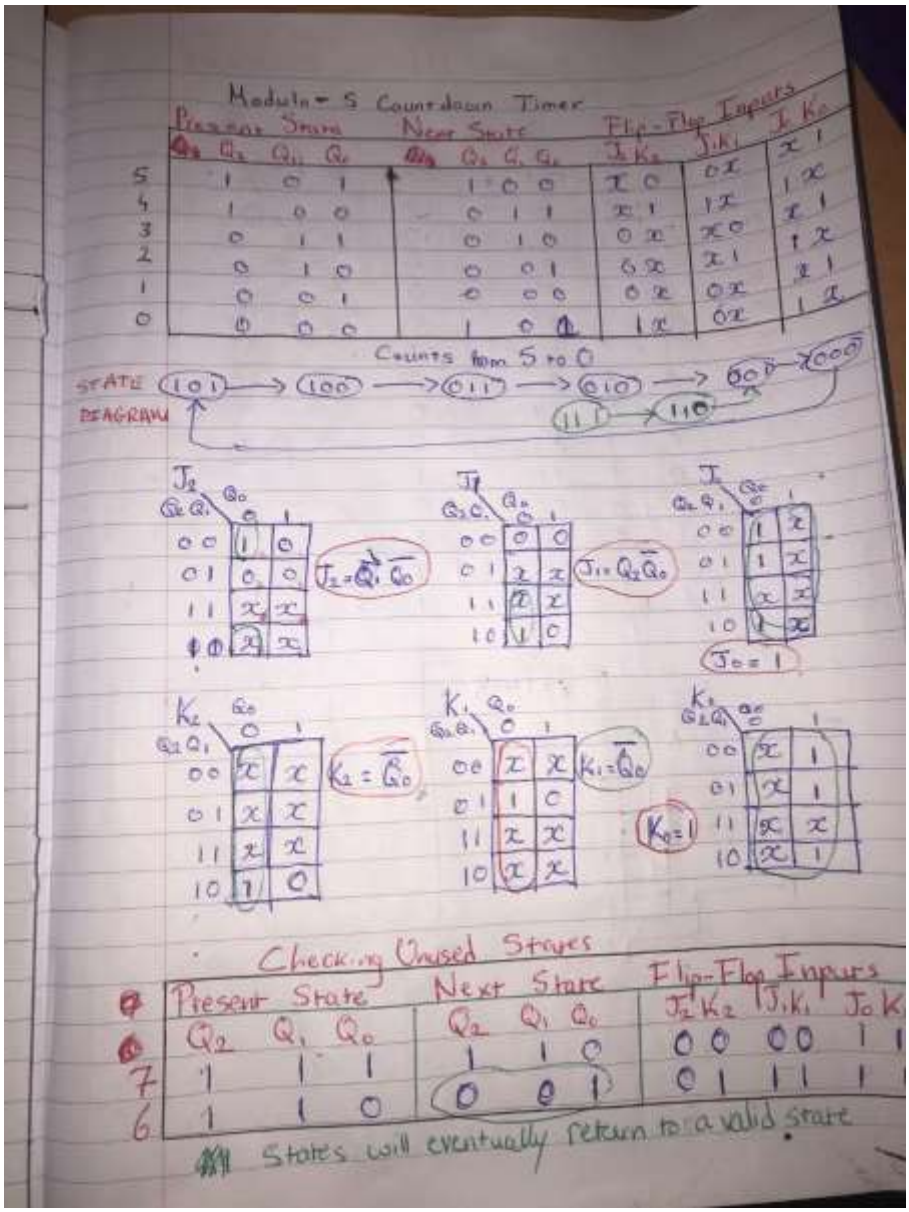
All states return to a valid state *Showed in Green



S1 Countdown



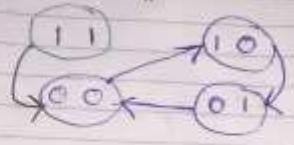
S2 Countdown


Modulo - 5 Countdown Timer
Present State
Next State
Flip-Flop Inputs
Counts from 5 to 0
STATE DIAGRAM
Checking Unused States
Present State
Next State
Flip-Flop Inputs
States will eventually return to a valid state

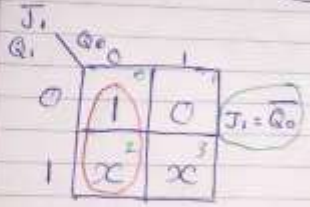
Modulo-3 Countdown Counter

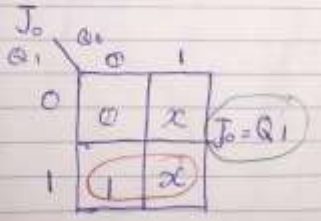
Using 2 bits to Count from 2 to 0 (for 10 to 0)

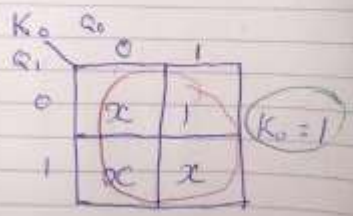
	Present State Q_1	Q_0	Next State Q_1	Q_0	Flip-Flop Inputs J_1, K_1	J_0, K_0
3	1	1	x	x	x x	x x
2	1	0	0	1	x 1	1 x
1	0	1	0	0	0 x	x 1
0	0	0	1	0	1 x	0 x



State Diagram



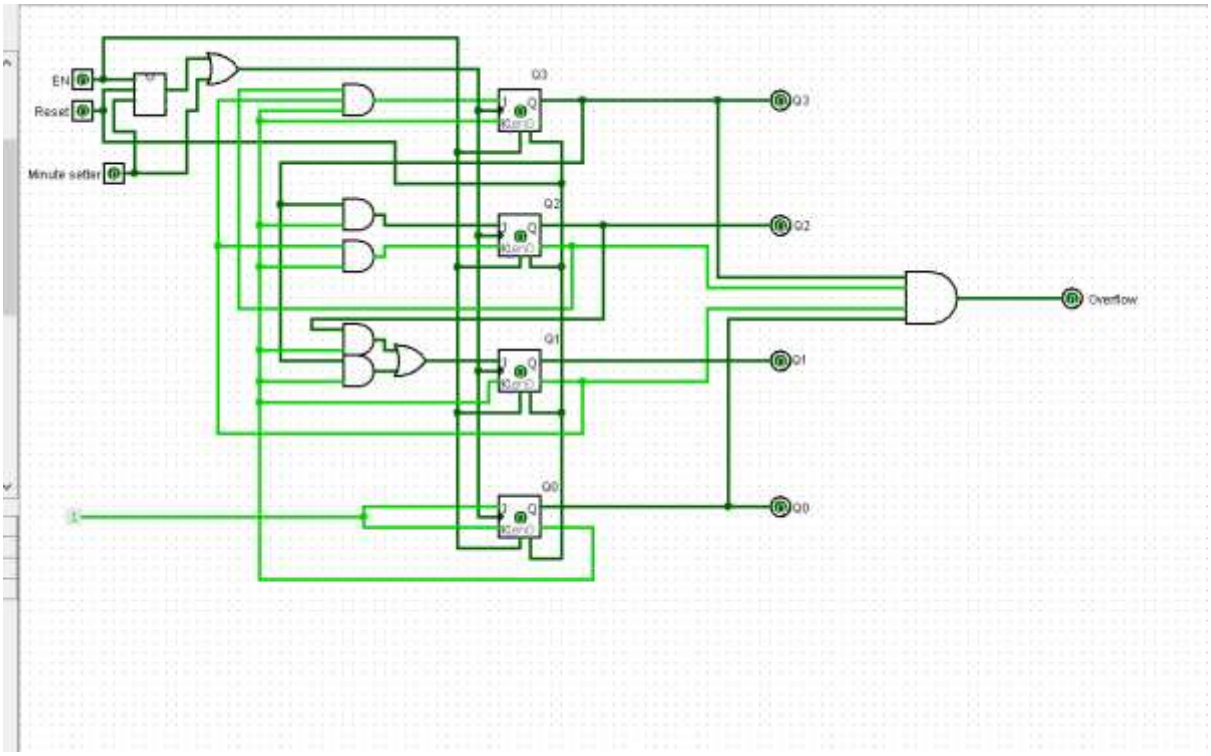




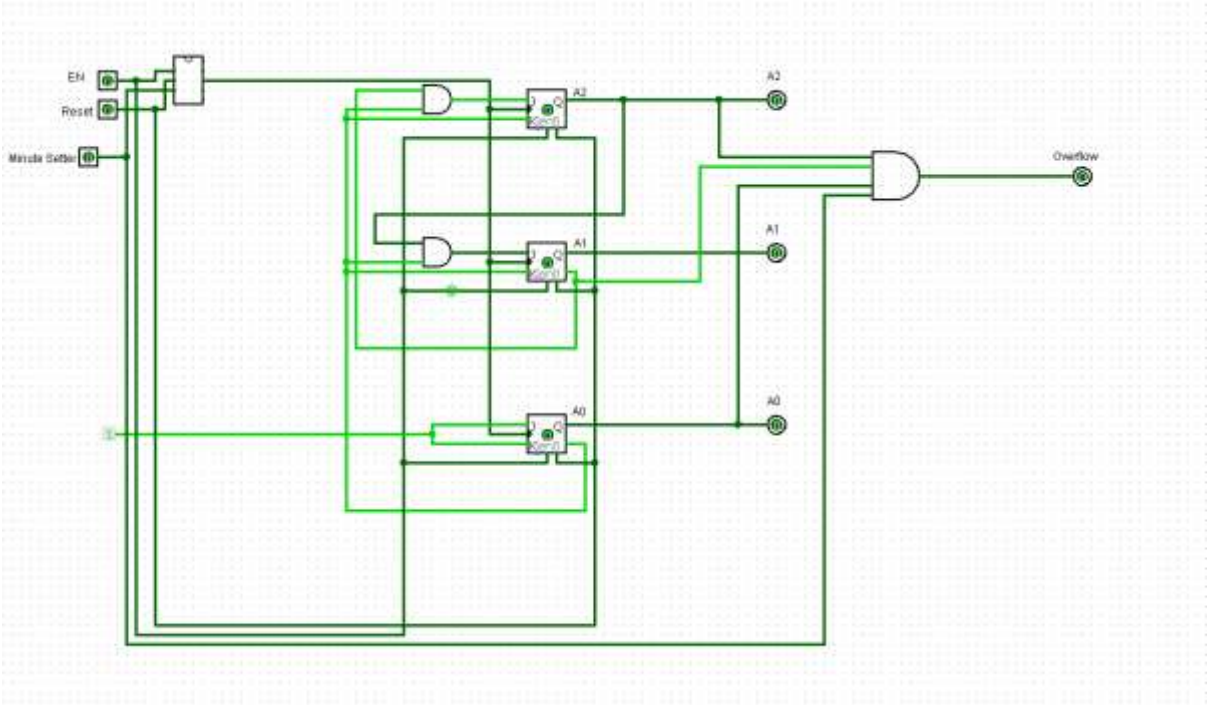
Checking Unused States

	Present State Q_1	Q_0	Next State Q_1	Q_0	Flip-Flop Inputs J_1, K_1	J_0, K_0
3	1	1	0	0	0 1	1 1

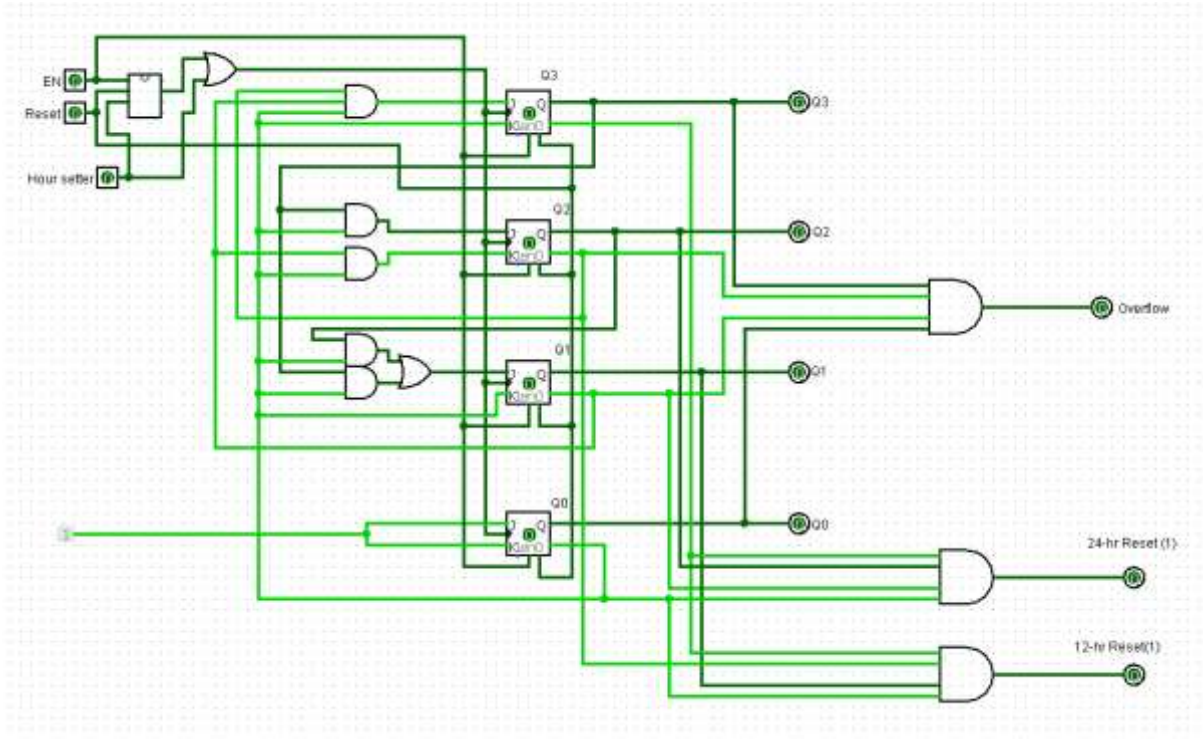
Returns to a valid state so circuit will be valid



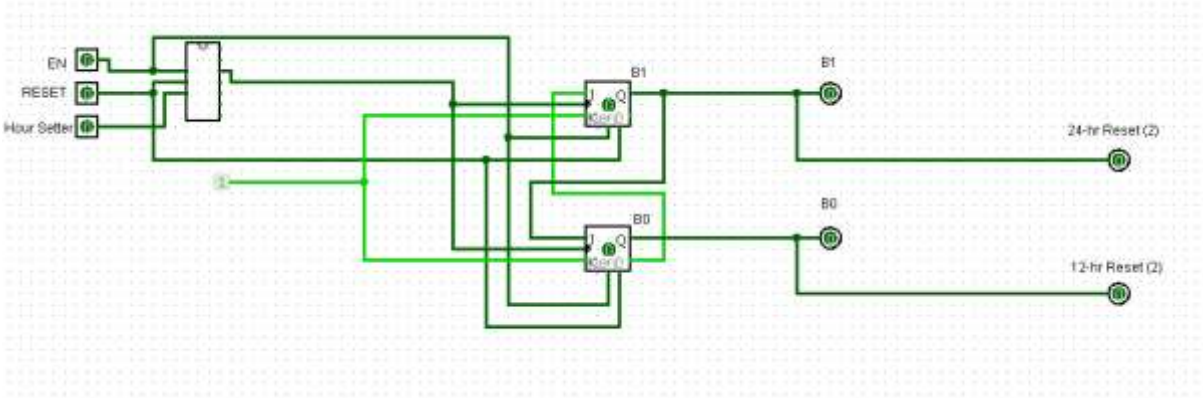
M1 Countdown



M2 Countdown



H1 Countdown

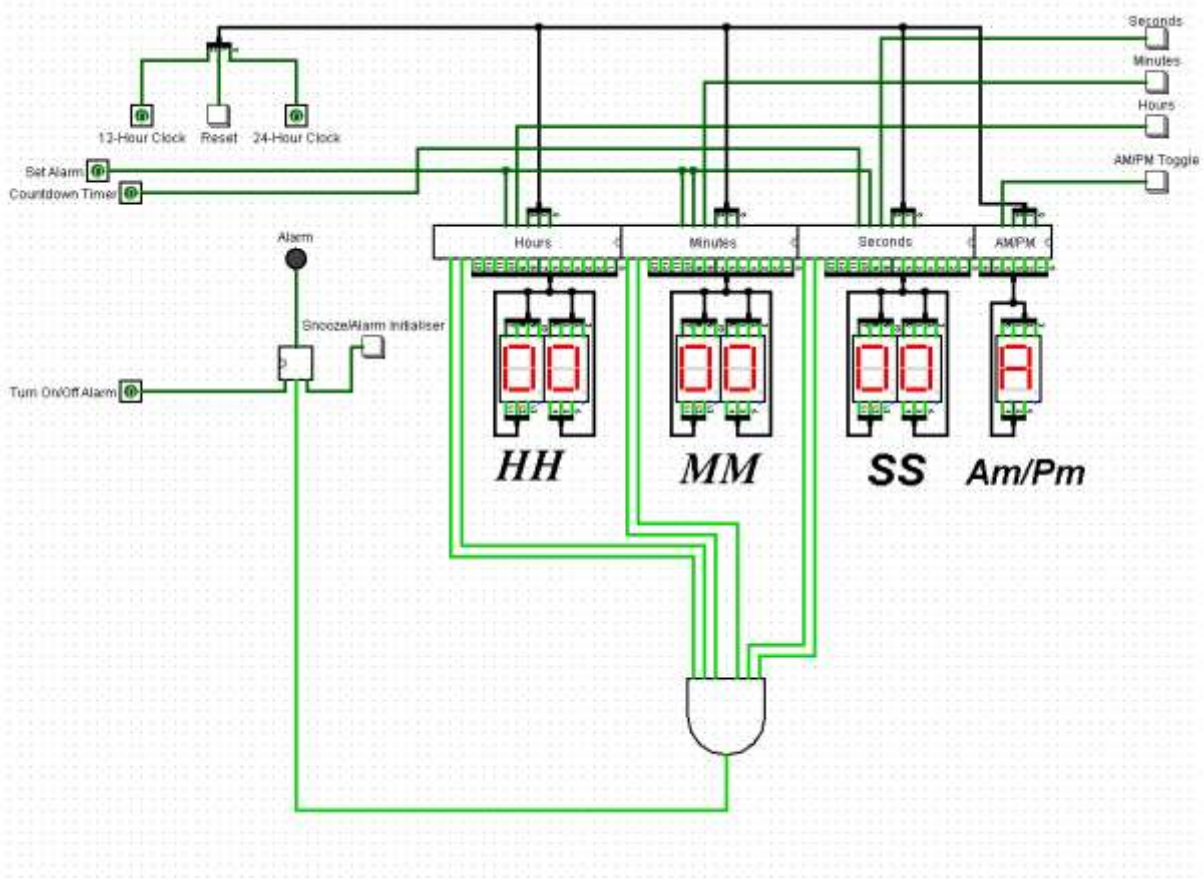


H2 Countdown

These are all the circuits involved in making the countdown clock.

Instruction Manual/Leaflet

This contains every piece of information you need to know about running this digital clock.



Firstly before the user does anything with the clock please press the reset button in order to clear the clock and avoid any mishaps using the clock.

Setting the clock

1.) In order for the user to set the clock they must first reset the clock using the reset button and make sure that the ticks are not enabled.

2.) If the user decides that they want to use the clock in 24-hour format. Please click the 24-hour mode enabling pin. If they decide to use it in 12-hour mode, click the 12-hour mode pin. (BOTH PINS CAN NOT

BE ON OR THE CLOCK WON'T FUNCTION PROPERLY.)

3.) If they wish to set a time on the clock they must use the buttons to set a time withing the range of 23:59:59 (24-hour mode) and 11:59:59 (12-hour mode).

4.) The buttons are labelled accordingly and the time can be adjusted by pressing these buttons until they obtain their desired time.

5.) When the user has selected their time and set it the enabling pin of the desired mode must be on and the ticks must be enabled.

6.) In order to pause the time switch of the enabling pin. In order to stop the clock completely switch off the enabling pin and disable the ticks. Press the reset button to completely reset the whole clock back to 0.

7.) To toggle the AM and PM when setting time in 12-hour mode click the AM/PM toggle button. (THE 12-HOUR MODE ENABLING PIN MUST BE ON FOR THIS TO BE TOGGLED.)

Using the Alarm

1.) In order to use the alarm first make sure that the clock has been reset to avoid any problems with the clock.

2.) To set a time on the alarm switch on the Set Alarm enabling pin, then use the buttons to input the desired time. (MAKE SURE NO OTHER PIN IS TURNED ON.)

3.) When the user has input the desired time the user must then switch of the Set Alarm enabling pin, and choose their desired mode (24-hour or 12-hour.).

4.) When the user has done so the user must turn on the alarm and press the Alarm initialiser button.
5.) The user must then enable the ticks and allow the clock to run/count.
6.) The LED must flash red if the time set on the alarm is reached. It shall flash red momentarily for thirty seconds if the user wishes to turn it off manually they can press the snooze button or turn off the alarm completely, by disabling the enabling pin.
7.) If the user would like to input another time for the alarm they should press the reset button and follow steps 1-6 again for their desired time.

Countdown

1.) In this clock the countdown timer is not functioning correctly but if the user wishes to turn it on and see the seconds count down from 59-0, they must enable the countdown timer pin, enable a mode, and enable the ticks.