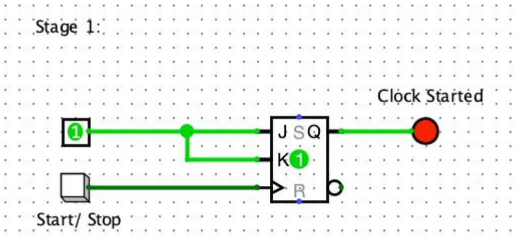
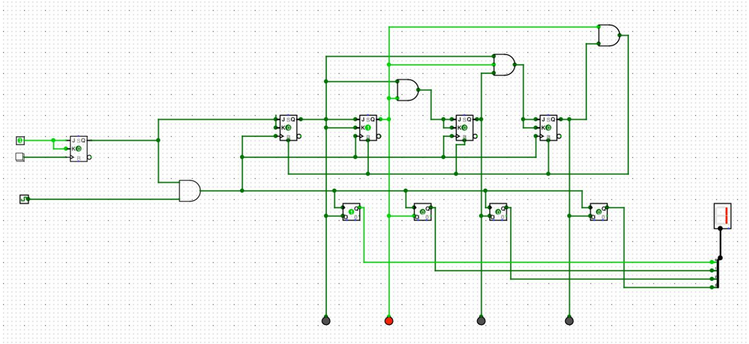
Student's name: Ai Vi Tran
Student's ID: 104209393
Unit code: COS10004
Lab session: 10:30-12:30 Thursday
Tutor's name: MD Kafil Uddin

Stage 1: Implement the Start/Stop button:



I using JK flip flop for the start/stop button because the main advantage of this flip-flop is the ability to toggle between two states. When both of them are high, the output of the flip-flop will toggle between its current state (either high or low) on each clock pulse. Which will make it easier to control the output.

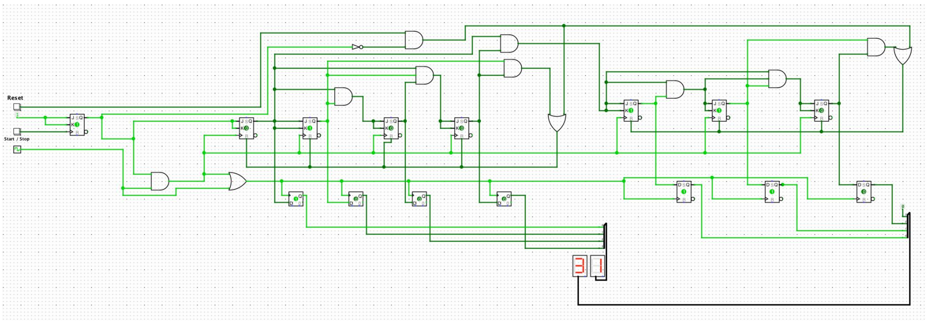
Stage 2: Implement Single Digit "Seconds" Display:



Firstly, as it was mentioned in the instruction of the assignment, I create a 4-bit-synchronous-counter in modulus of 10 that keeps track of the number of "seconds"(in increments of 1s, between 0s and 9s), which is the counter would wrap back to 0 whenever it reaches 9. And an AND gate is used to take the output the 2nd and the 4th bit then results in the output which is connected to the input of reset.

Secondly, to make it possible to stop when I click on the Start/Stop button again, I create an AND gate for the input of clock pulse, which means, automatic clock and the state would be combined to enable the input of the clock pulse. The result that I receive from this is the "Second" display would Stop when the Start/Stop button is pressed in the Start state, also Resume counting when the Start/Stop button is pressed in the Stop state.

Stage 3: Implement the full two-digit "Seconds" display:



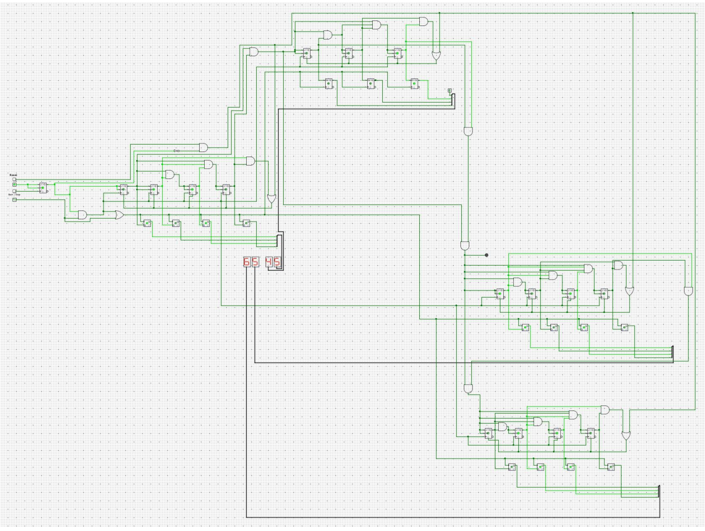
To make the seconds display now shows “ Seconds” in 1s increment using both the units and tens column:

- Firstly, I create another counter in modulo of 6 (3-bit-synchronous-counter) which is similar to what I have done with the 4-bit-synchronous-counter in stage 2. In this stage, the counter will wrap back to 0 when it reaches 6.

- Secondly, to make it turn 1 when the 4-bit-synchronous-counter reaches 10, the input of the 1st flip-flop will receive the pulse from the AND gate of 9, and then whenever the 1st “Second” display reaches 9, it’ll receive the pulse and display it in a second after to avoid the illegal state, which means when the 1st “Second” display wraps back to 0, the 2nd “Second” display will turn 1 at the same time.

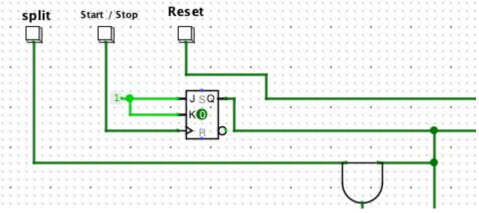
- Thirdly, to make the Reset button work, I used a NOT gate, an AND gate. whenever the it is in the Stop state, the NOT gate make it possible for the input of the AND gate, another input is the pulse of Reset button. I also use an OR gate for the input of reset, so that it won’t disturbed the modulo. Them all result in the display reset to all 0 whenever the Reset button is clicked and enters the Stop state.

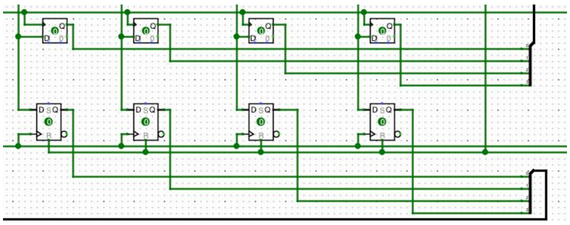
Stage 4: Implement the “Minutes” display:



The concept of this stage is the combination of stage 2 and 3, o make it wrap back to 00 after reaching 99, I use the counter with the modulus of 10 for each “Minutes” display. The combination of both 2 “Minutes” displays, when both of them reach the highest number of its modulus, the AND gate will send the pulse to the 1st “Minutes” display.

Stage 5: Implement the “Split” button:



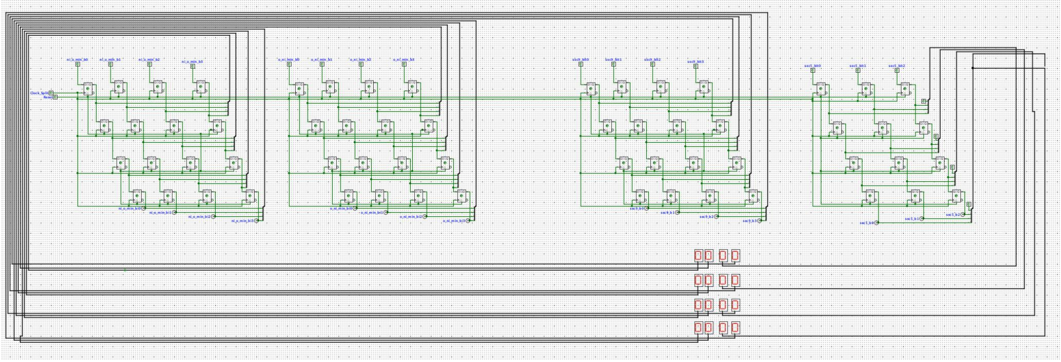


In this stage, I used an AND gate :

- Input: 1 input is from the Split button, the other one is the state of the digital stopwatch.
- Output: is connected to the clock input of the shift register.

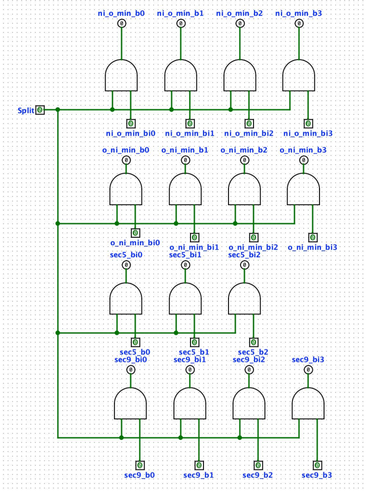
Stage6: Implement “Split” time recording and multi-”Split” display:

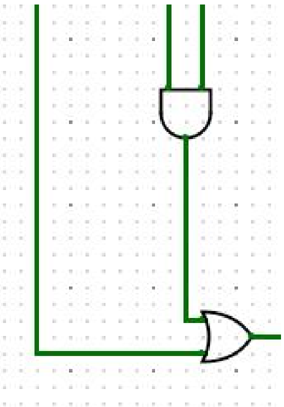
6A: When the stopwatch is active:

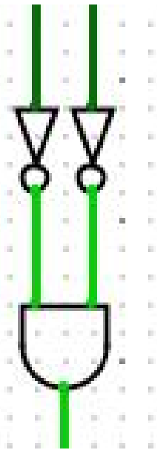


To store 5 times, because I have had one of them are stored and displayed already, so I added another 4-shift-register, their function is to record times. In 6A, I used D flip-flop for my storage. The output of previous shift register will be the input of the next one and so on. To make it operate smoothly, their clocks are connected to the same clock as stage 5, when the split button is press.

6B: When the stopwatch is in the Stop state:







Part 6B is quite complicated, the solution i came up with is divide them into 2 option (using OR gate at the input of “Split” clock

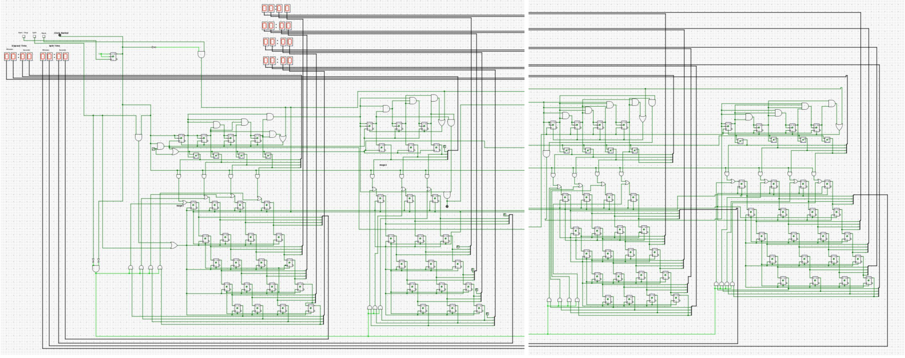
- Option1: When the stopwatch is active, the Split display will only show the Slip time, whenever the “Slpit” button is pressed (AND gate, input1: Start state, input2: Split button) to ensure it always display the Split time when the split button is pressed.

- ➢ Option2: When the stopwatch is in the Stop state, another input of the OR gate is only the Split button to show each split time in turn on the “Split” Time display, when the “Split” button is pressed (the “Split” display would move to the next split time record and the “Split” time display will wrap back to the earliest recorded split time after showing the most recent split time)

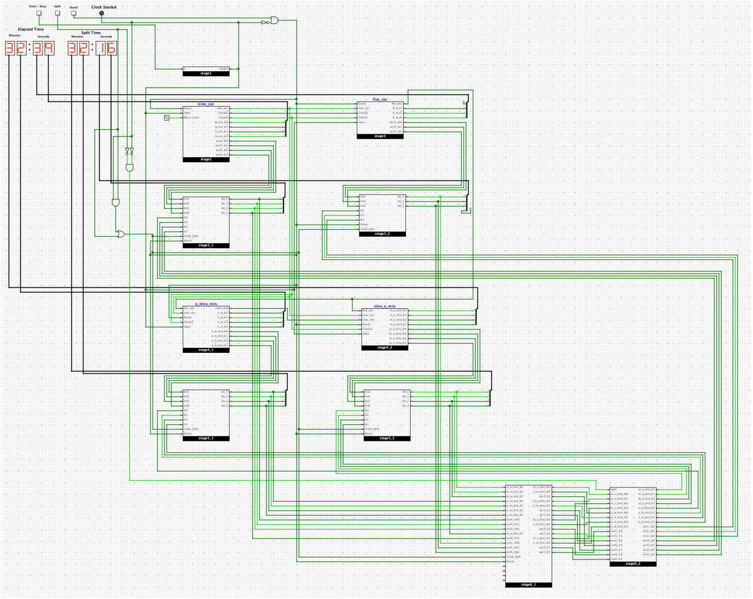
P/S

- ➢ The left hand side picture is to gather the condition of showing the split time display when it is in the Stop state or the Active state.
- ➢ The picture in the middle is the 2 option to activate the split clock
- ➢ The right had side picture is the input to activate the “Split” display when it is in the Stop state, it will start enable the “Split” display to display the earliest split time after showing the most recent time in the Stop state, simultaneously with the Split butotn is pressed.

Here is my sketch before rearrange them into sub circuits:



Here is my work after rearrange and put them all into sub circuit:


Start / Stop
Split
Reset
Clock Started
Elapsed Time
Minutes
Seconds
32:39
Split Time
Minutes
Seconds
32:16