ASIA PACIFIC UNIVERSITY OF TECHNOLOGY & INNOVATION

EE008-3-2-DE
DIGITAL ELECTRONICS

TITLE	INDIVIDUAL ASSIGNMENT
STUDENT NAME	MOHAMMAD FAWZAN ALIM
STUDENT ID	TP064501
INTAKE	APD2F2206CE
LECTURER NAME	DR. CHANDRASEKHARAN NATARAJ
SUBMISSION DATE	09 FEBRUARY 2023

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

Digital Electronics is a field of electronics that deals with digital signals and digital circuits. A digital signal is a signal that has only two discrete values, such as high and low, on and off, 1 and 0 or 5V and GND. Digital Electronics studies on the process of utilizing these digital signals to store, process and transmit information. Digital clock is an example of digital electronics in our day-to-day life.

Digital clock is one of the most widely used devices in our daily life. It is made up of a combination of digital electronic circuits and components that work together to produce accurate timekeeping. A digital clock displays time in numerical format as opposed to an analog clock that display time through hands moving on a clock face. Digital clocks can be more precise and accurate in measuring time.

Objective of this individual assignment is to design and simulate a digital clock with 7 hours, 10 hours or 15 hours. I have discussed with my group and chose 10 hours. As for minutes and seconds in the clock, we can choose any number of minutes per hour and any number of seconds per minute.

2. DESIGN PROCESS

2.1 OVERALL DESIGN PROCESS

Objective of this assignment is to design a clock with a random number of seconds in a minute, random number of minutes in an hour and 10 hours. However, I have gone one step further and decided to give the user greater flexibility in setting the time values. The user will be able to choose the number of seconds in a minute and number of minutes in an hour using switches on the clock. But the hours section will not be configurable. The switches will come with clear instructions on how to configure the clock. This will result in a highly customizable clock that can be tailored to meet specific needs. This will make the clock a versatile and useful tool that can be easily adapted to a wide range of applications.

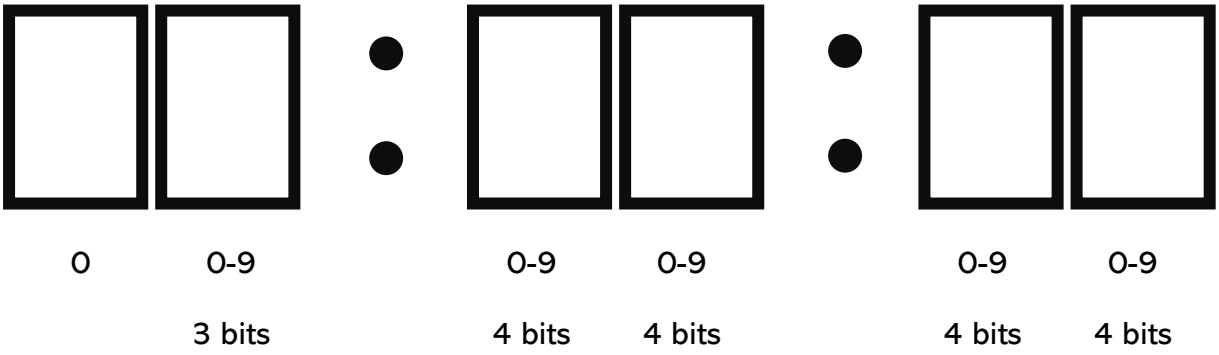


Figure 1: Range and bits of every display

According to the plan above, user can customize the clock to any number of seconds and minutes from 01 to 99. As for hours, it is set to 10 hours. This means the hour displays on the far left can hold the value from 00 to 09. As soon as it reaches 10, the whole clock will reset and go back to 00:00:00.

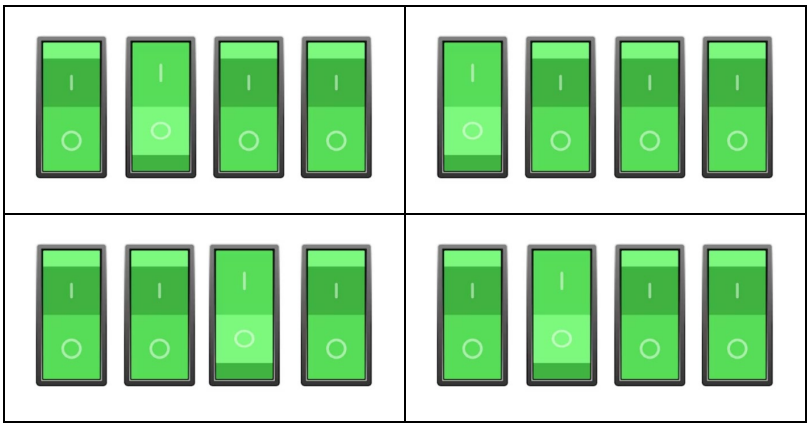


Figure 2: Minutes and Seconds Configuring Switches

There will be four sets of switches on the clock that can be used to configure the number of seconds per minute and number of minutes per hour. Top sets here in Figure 2 are for minutes and bottom sets are for seconds. In each set, there are four switches, representing 4 bits in one display. If we read the values of the switches, we have 0100 and 1000 in binary, which represents 4 and 8. That means 48 minutes. Similarly, we get 0010 and 0100 from the switches in the bottom row, which represents 2 and 4, meaning 24 seconds. Therefore, if we configure the clock using these values, it will have 48 minutes and 24 seconds. Since I am doing the 7-hour clock, therefore my clock has a range from 00:00:00 to 09:47:23. Once the clock reaches 9:47:23, it will reset to 00:00:00 in the next tick by completing 7 complete hours.

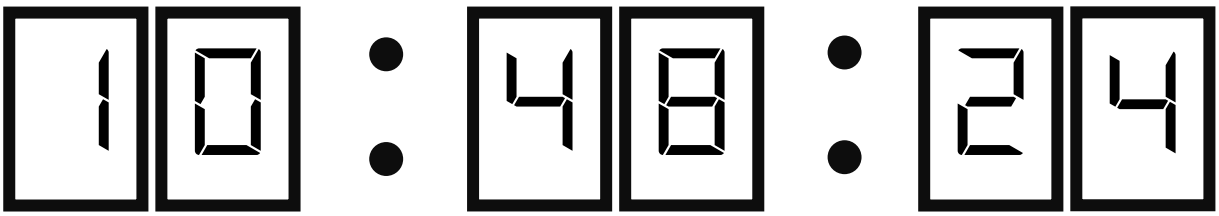


Figure 3: Number of hours, minutes and seconds based on button values.

Because of the configurable design, a new issue will arise. The user can modify the switches while the clock is in operation. This could result in either intentional or accidental malfunctions of the entire clock. To prevent this, we have to freeze the values of the switches once the clock is in operation and ignore the changes until the clock is restarted again. This means we also a restarted button in our clock. To make it organized, we can divide the designing process into two parts. In first part, we will freeze the button values and in the second part, we will design the logic behind the clock.

2.2 FREEZING BUTTON VALUES

When it comes to freezing the values of the switches in the clock, we have a few options to choose from. For example, we have D flip-flop, JK flip-flop, T flip-flop, latch or register. After considering all the options, I have decided to use D flip-flop for this application because of its simplicity.

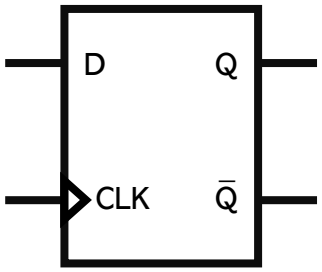


Figure 4: D Flip-Flip

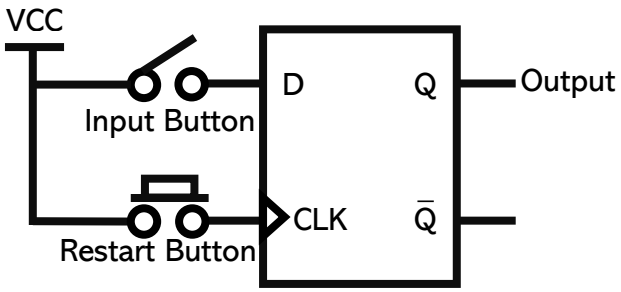


Figure 5: Configuration of D Flip-Flop in the Clock

Table 1: D Flip-Flop Truth Table

Clock Signal CLK	Data Input D	Previous State Q_{n-1}	Current State Q_n
0	X	0	0
0	X	1	1
1	0	X	0
1	1	X	1

A D flip-flop (or Delay flip-flop) is a digital electronic circuit used to delay the change of state of its output signal until the next rising edge of a clock input signal occurs (Analog Devices, n.d.). We can see a symbol of D flip-flop in Figure 4 and how we are going to configure D flip-flop in our clock in Figure 5. As we can see, it has four basic pins. D pin is the data input pin where our switches output will go. CLK pin is the Clock pin, which works like a reset pin. Therefore, we will connect our reset/restart button with it. And lastly, Q is our output pin.

In the truth table of D flip-flop in Table 1, $\bar{Q}$ is not there. Instead, we have previous state, Q_{n-1}. Since we are freezing values, we need to know about the previous state of the output. As we can see, when the clock signal has a low input, It does not matter what is the

value of data input, the output remains the same as before. On the other hand, when the clock has a high input, the output is as same as data input. In short, when CLK = 0, Q = No change and when CLK = 1, Q = D.

This is exactly what we want for our clock. When the clock is in operation, reset/restart button is giving low output, meaning CLK = 0. That means the D flip-flops will freeze the values. When we are restarting the clock, the reset/restart button will give high output, meaning CLK = 1. So, the D flip-flops will read the new button values and freeze them again. Therefore, when the clock is in operation, we are not reading the current values of the switches, we are reading the values of the switches when we were starting/restarting the clock.

To accomplish our goal, we will need a button and a D flip-flop for every bit we want to let the user configure. Since we want to make both seconds and minutes configurable, that is 16 bits in total, according to Figure 1. That means we will need 16 toggle switches and 16 D flip-flops in our entire clock. And lastly, a push button to reset/restart the clock.

2.3 CLOCK DESIGN

2.3.1 4-BIT BASIC 0-9 COUNTER DESIGN

The clock is basically a counter underneath the hood. The clock we use in our day-to-day life just counts from zero to fifty-nine for seconds and minutes. Similarly, to achieve that, we have to design a clock that can count from zero to nine for every display. Since 9 in binary is 1001, we basically have to design a 4-bit counter. But we only have access to one bit counter. One of the most popular options for 1 bit counter is JK flip flop. JK Flip-flop is a type of edge-triggered flip-flop that has two inputs, J and K, and a clock input. The J and K inputs determine the state of the output, while the clock input triggers the state transition (Electronics Tutorial, n.d.).

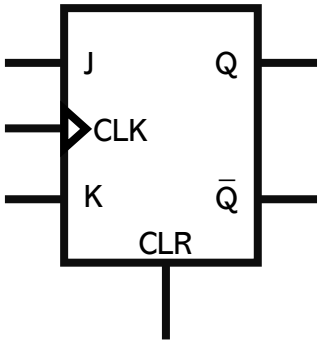


Figure 6: JK Flip-Flop

Table 2: JK Flip-Flop Truth Table

CLR	CLK	J	K	Q_{n-1}	Q_n	State
1	X	X	X	X	0	Clear
0	0	X	X	0	0	Memory
0	0	X	X	1	1	Memory
0	1	0	0	0	0	Memory
0	1	0	0	1	1	Memory
0	1	0	1	X	0	Reset
0	1	1	0	X	1	Set
0	1	1	1	0	1	Toggle
0	1	1	1	1	0	Toggle

As we can see in Figure 6, JK flip-flop has four input pins and two output pins. Based on the values of the pins and previous state of the JK flip-flop, we get the truth table shown in Table 2. We are only interested in Toggle state and Clear state. For the Toggle state to work, CLR = 0, CLK = 1, J = 1, K = 1; then we get the output $Q_n = \overline{Q_{n-1}}$. To make a 4-bit counter using JK flip-flop, we will need 4 JK flip-flips, since one JK flip-flop works with only one bit.

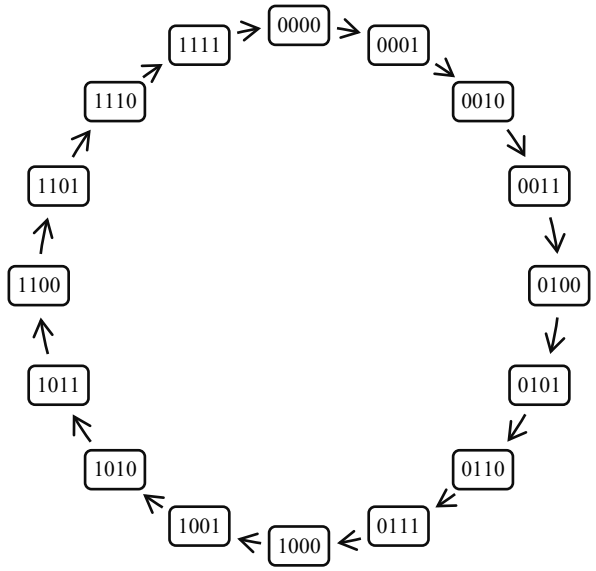


Figure 7: 0-15 4 Bit Counter Value Cycle

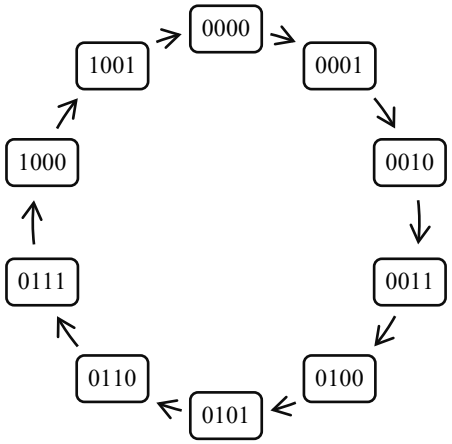


Figure 8: 0-9 4-bit Counter Value Cycle

We need to design a 4-bit counter that can cycle through the value from 0000 to 1111 shown in Figure 7. We have broken down the 4 bits into 4 JK flip-flops in Table 3 below with their current state and next state. This will help us in the planning of the JK flip-flops.

Table 3: States of 4-bit counter

Current State				Next State			
$4Q_n$	$3Q_n$	$2Q_n$	$1Q_n$	$4Q_{n+1}$	$3Q_{n+1}$	$2Q_{n+1}$	$1Q_{n+1}$
0	0	0	0	0	0	0	1
0	0	0	1	0	0	1	0
0	0	1	0	0	0	1	1
0	0	1	1	0	1	0	0
0	1	0	0	0	1	0	1

0	1	0	1	0	1	1	0
0	1	1	0	0	1	1	1
0	1	1	1	1	0	0	0
1	0	0	0	1	0	0	1
1	0	0	1	1	0	1	0
1	0	1	0	1	0	1	1
1	0	1	1	1	1	0	0
1	1	0	0	1	1	0	1
1	1	0	1	1	1	1	0
1	1	1	0	1	1	1	1
1	1	1	1	0	0	0	0

In Table 3, we can see every single scenario of the counter. For every state, we can find out the next stage of the counter. If we study the values carefully, we can see that 1Q is toggling every time the signal generator is changing its value. 2Q is toggling every time 1Q completes a cycle by going from 1 to 0. Similarly, 3Q and 4Q are toggling every time 2Q and 3Q completes a cycle respectively. This means, the output (Q) of the first JK flip-flop will act as the clock (CLK) for the second JK flip-flop and the output (Q) of the second JK flip-flop will act as the clock (CLK) for the third JK flip-flop and so on. But there are two output pins in each JK flip-flop, we have to decide which one to take the CLK connection from.

There are two types of JK flip-flops based on the triggering edge. Positive Edge Triggered flip-flop and Negative Edge Triggered flip-flop. In Positive Edge Triggered JK flip-flop, the state transition is triggered by a positive edge of the clock signal, that is, when the clock signal transitions from low to high. On the other hand, In Negative Edge Triggered JK flip-flop, the state transition is triggered by a negative-going edge of the clock signal, that is, when the clock signal transitions from high to low (Fahad, 2022). I have decided to use Negative Edge Triggered flip-flop in my design. That means, to trigger a JK flip-flop, the clock (CLK) has to go from 1 to 0. We get that from the Q of the previous JK flip-flop. To summarize, we have to connect clock (CLK) of the second, third and fourth JK flip-flops to the output (Q) of the first, second and third JK flip-flops respectively to get the desired result from the

Negative Edge JK flip-flops. This design is called Asynchronous design (Electronics Hub, 2015).

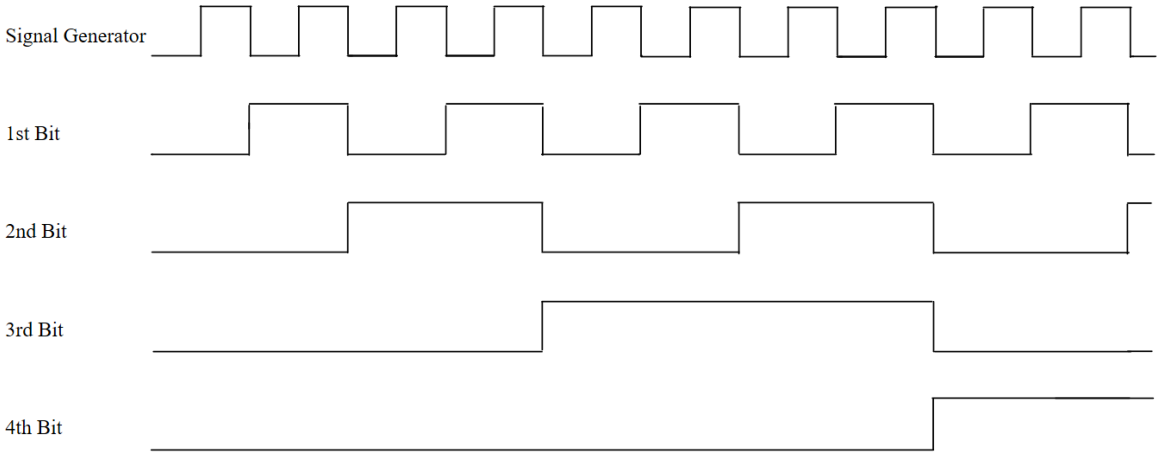


Figure 9: Timing Diagram

As we can see in Figure 9Figure 10, when the signal generator is falling from high to low, the first JK flip-flop gets triggered. Similarly, when the first JK flip-flip is falling from high to low, the second flip-flop is getting triggered and so on. Now we can design our 4-bit counter using this information.

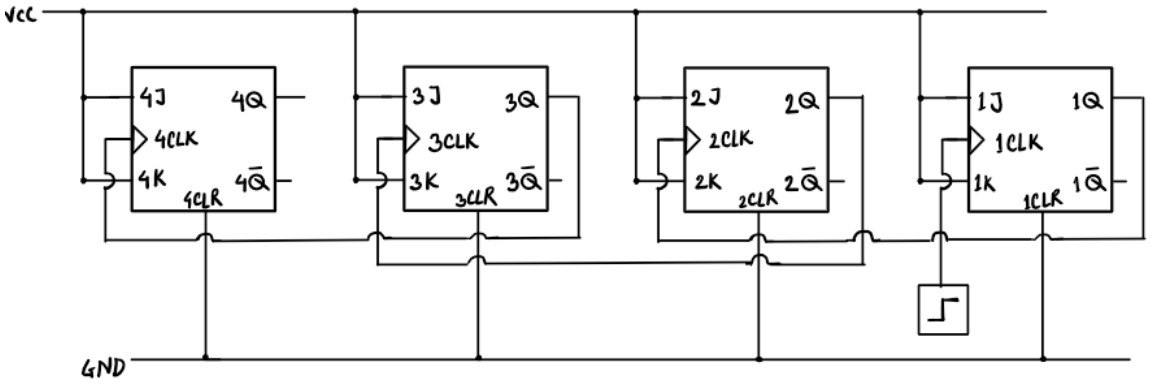


Figure 10: 0-15 4 Bit Counter using JK Flip-Flops

Figure 10 shows the design of the 4-bit counter. This counter counts from 0000 to 1111 as shown in the clock cycle in Figure 7. But there is a problem. We want to count from 0 to 9, not from 0 to 15. Our current design is giving us 0 to 15. So, we have to apply some additional conditions to restrict it to go beyond 9. Basically, creating a counter to count from 0000 to 1001, which is 0 to 9 in decimal. The 0-9 clock cycle is shown in Figure 8. To do that, we have to reset all JK flip-flips to zero once it reaches 10, which is 1010 in binary. Which means when 1Q = 0, 2Q = 1, 3Q = 0 and 4Q = 1, we have to send a high signal into the clear (CLR) pins of all JK flip-flops to reset them to zero (0000). We can do that by applying AND gates and NOT gates. To make it even simpler, we can just ignore the value of 1Q and 3Q because when 2Q

and 4Q have the value 1, it is 10, 11, 15 or 16. Since 10 comes first and we reset the counter back to zero after reaching 10, we never reach 11, 15 or 16. Therefore, it is safe to say, when 2Q = 1 and 4Q = 1, the value of the counter is 1010, which is 10 in decimal. That means, whenever 2Q = 1 and 4Q = 1, we can reset the clock. We can accomplish that by adding 2Q and 4Q into the input pins of an AND gate and connecting the outputs to the clear (CLR) pins of all JK flip-flops.

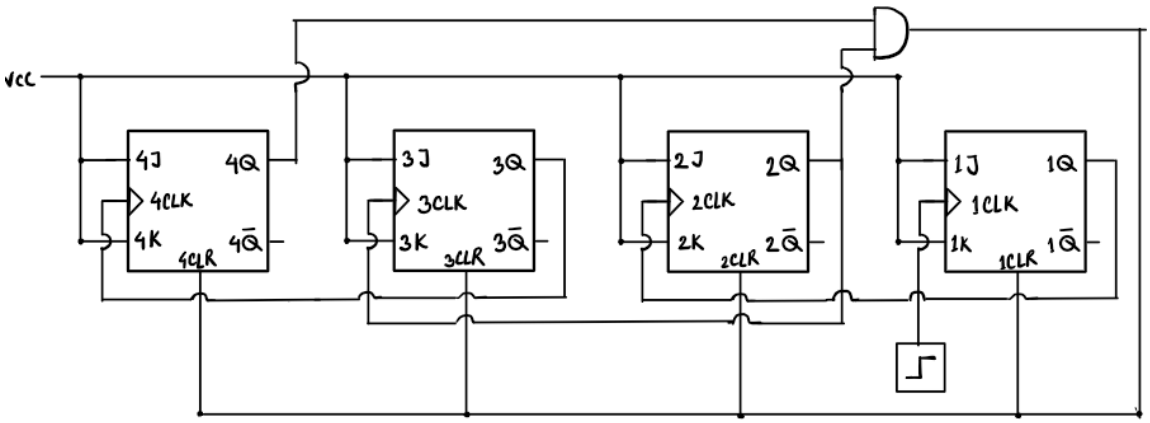


Figure 11: 0-9 4-bit Counter using JK flip-flops

Figure 11 shows our desired 4-bit counter that counts from 0000 to 1001, which is 0 to 9 in decimal. Once it reaches 1010 (10 in decimal), it resets back to 0000 (0 in decimal). We can now use this 4-bit counter to design our seconds and minutes section of the clock.

2.3.2 SECONDS DESIGN

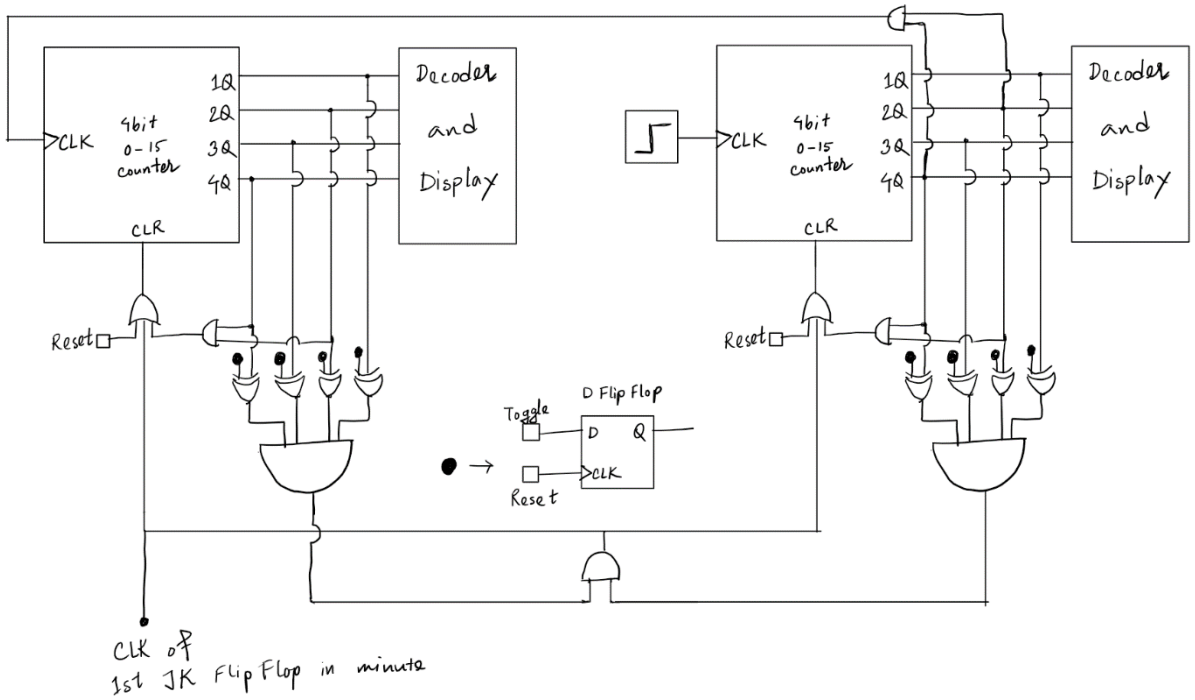


Figure 12: Hand sketch of Seconds Section

Figure 12 shows the hand sketch of the seconds section. I have combined the 4 bit counter that I designed in Figure 11, configurable switches from Figure 5 with some basic gates such as AND gates, OR gates and X-NOR gates to achieve this. The switch values in Figure 12 read 24 seconds, that means the clock has 24 seconds. Therefore, it will go from 0 to 23. Outputs of the counter is going into a 4 to 7 Decoder, which is connected to a seven-segment display, where we can see the output in decimal.

Now, to make the clock fully functional, we have to reset the JK flip-flops under two conditions. First one is when it reaches 10 and the second one is when it reaches 24 seconds. We have already done the first one in the counter. As for the second one, we have to check if the value of the first counter is 4 (0100) when the value of the second counter is 2 (0010). To do that, we are using X-NOR gates and AND gates. We are using X-NOR gates because X-NOR gates gives high output when the inputs are equal (Electronics Tutorial, n.d.), which is shown in Table 4.

Table 4: X-NOR gate Truth Table

A	B	X
0	0	1
0	1	0
1	0	0
1	1	1

4 X-NOR gates on both sides are checking if the counter values are matching with the values from D flip-flops. If all the values match, then we send a signal to the clear (CLR) pin of the JK flip-flops to reset it to zero. At the same time, we also send a signal to first JK flip-flop of the minutes section.

2.3.3 MINUTES DESIGN

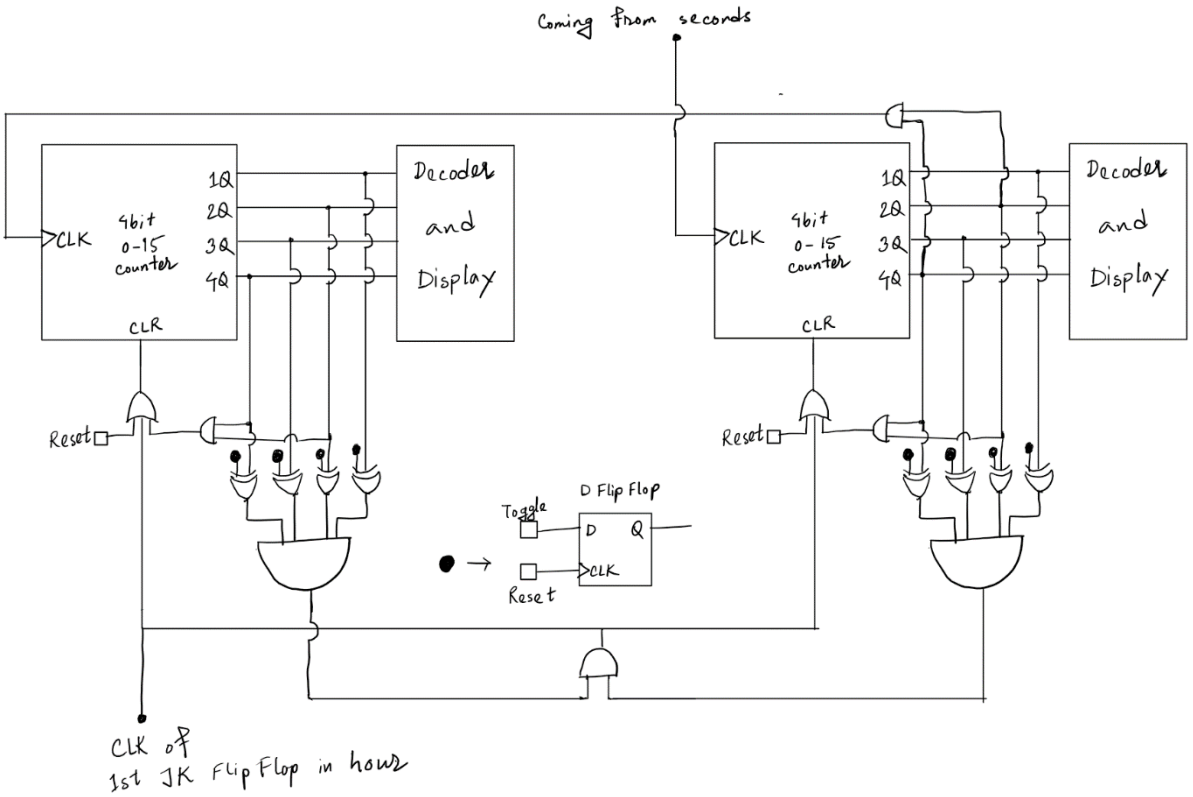


Figure 13: Hand Sketch of Minutes Section

Design process of the minutes section in Figure 13 is almost same as seconds section shown in Figure 12. There are only a few differences in the design. First of all, button values read 48 minutes here. Meaning it will go from 0 to 47. The first JK flip-flop clock (CLK) signal is coming from the seconds section. Similar to seconds section, when we are resetting the minutes to 00, we send a signal to the clock (CLK) pin of the first JK flip-flop of the hours section.

2.3.4 HOURS DESIGN

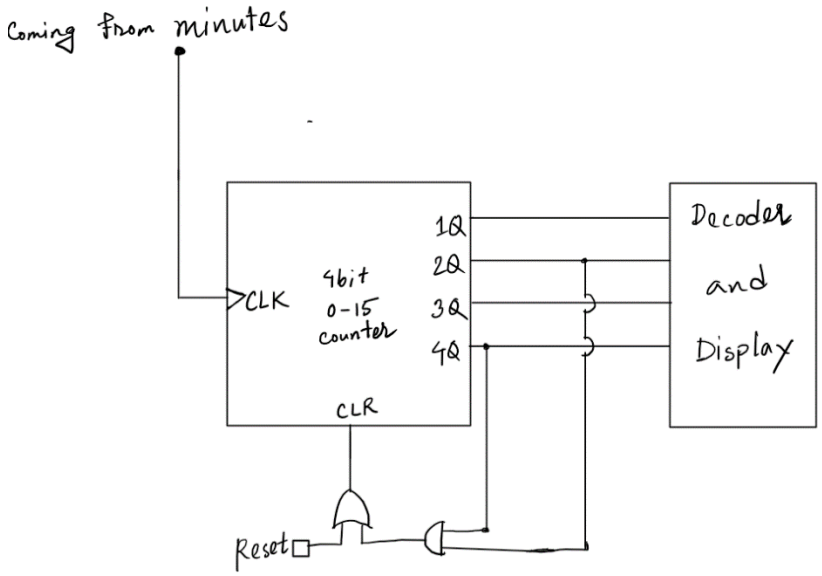


Figure 14: Hand Sketch of Hours Sections

Hour section of the clock is the simplest section of all. It just a basic 4-bit counter that we designed in Figure 11 with no additional components. The clock of the first JK flip-flop gets its signal from the minutes section.

3. INDIVIDUAL SIMULATION

3.0 COMPLETE SIMULATION DESIGN

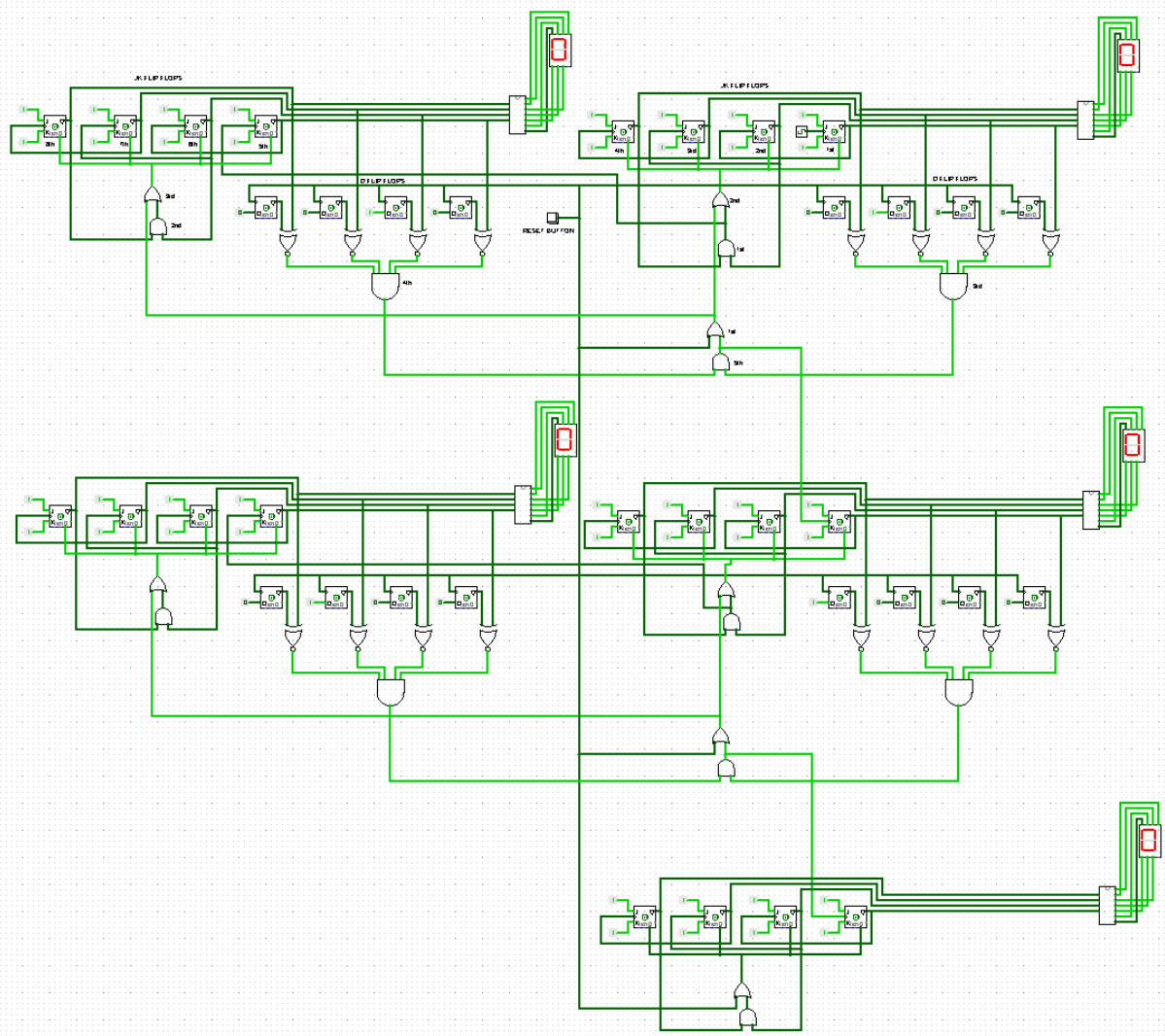


Figure 15: Complete Simulation Design

Figure 15 shows the complete design of the clock. The top two displays are for seconds, the displays in the middle are for minutes and the bottom display is for hour. We can see that the top four displays are connected to some D flip-flops, which are connected to some switches. The values from these switches decide the number of seconds per minute and number of minutes per hour in the clock. As oof the screenshot in figure 15, the values of the screenshot read 0010 and 0100 in the seconds section, which is 2 and 4 in decimal. That means the clock has 24 seconds per minute and it will go from 00 seconds to 23 seconds before resetting back to 00. Similarly, the values of the switches in minutes section read 0100 and 1000, which is 4 and 8 in decimal. That means the clock has 48 minutes per hour. Meaning it will go from 00 to 47 before turning back to 00.

3.1 SECONDS SIMULATION DESIGN

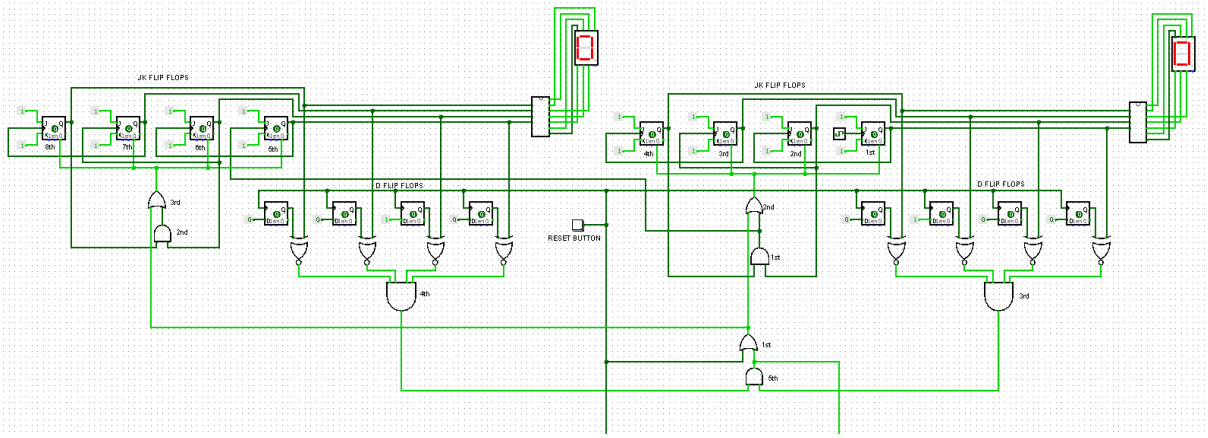


Figure 16: Simulation of Seconds Section

Simulation of seconds section in Figure 16 looks very similar to the hand sketch shown previously. On the top right of the figure, we see a set of 4 JK flip-flops. All the JK flip-flops has value 1 in their J and K pin. The first JK flip-flop on the top-right corner represents the LSB of the first digit of the seconds. The clock (CLK) of this JK flip-flop is connected to a signal generator. Then the output (Q) of this JK flip-flop is fed into the clock (CLK) pin of the 2nd JK flip-flop. The sequence goes on until the fourth JK flip-flop. The output of these JK flip-flops represent a 4-bit binary number which is fed into a 4 to 7 decoder. The decoder converts the data into 7-bit binary coded decimal (BCD) format, which then gets fed into a seven-segment display. We have a similar setup on the top left corner as well. These two setups give us a sequence.

On the bottom right corner of the JK flip-flops on both sides, we have 2 sets of 4 switches connected to 2 sets of four D flip-flops. These D flip-flops freeze the values of the switches until the reset button in the center is pressed. To make the clock fully functional, we have to reset every JK flip-flop under certain conditions. The JK flip-flops will reset under three conditions. First of all, when the reset button is pressed, it resets all the JK flip-flops in the clock. Secondly, when the value on each side reaches 10, we reset the JK flip-flops on that side only and send a signal to the next sets of JK flip-flops if necessary. 10 in binary is represented as 1010. That means when the output of the 2^{nd} and 4^{th} JK flip-flops are both high, we reset 1^{st} to 4^{th} JK flip-flops. That is demonstrated in the 1^{st} AND gate. At the same time, we send a signal from the output of this AND gate to the 5^{th} JK flip-flop. We see a similar reset mechanism on the left side, except it does not send a signal to the minutes section. Lastly, we reset the JK flip-flops when the clock reaches maximum number of seconds. In Figure 16, the

switches values read 24 seconds. That means when the clock reaches 23 seconds, in the next tick, it will reset back to 00. We accomplish this by the help of X-NOR gates and AND gates. When the values of the JK flip-flops becomes same as switches on both sides, we reset both sets of JK flip-flops and send a signal to the minutes section.

3.2 SECONDS SIMULATION RESULTS

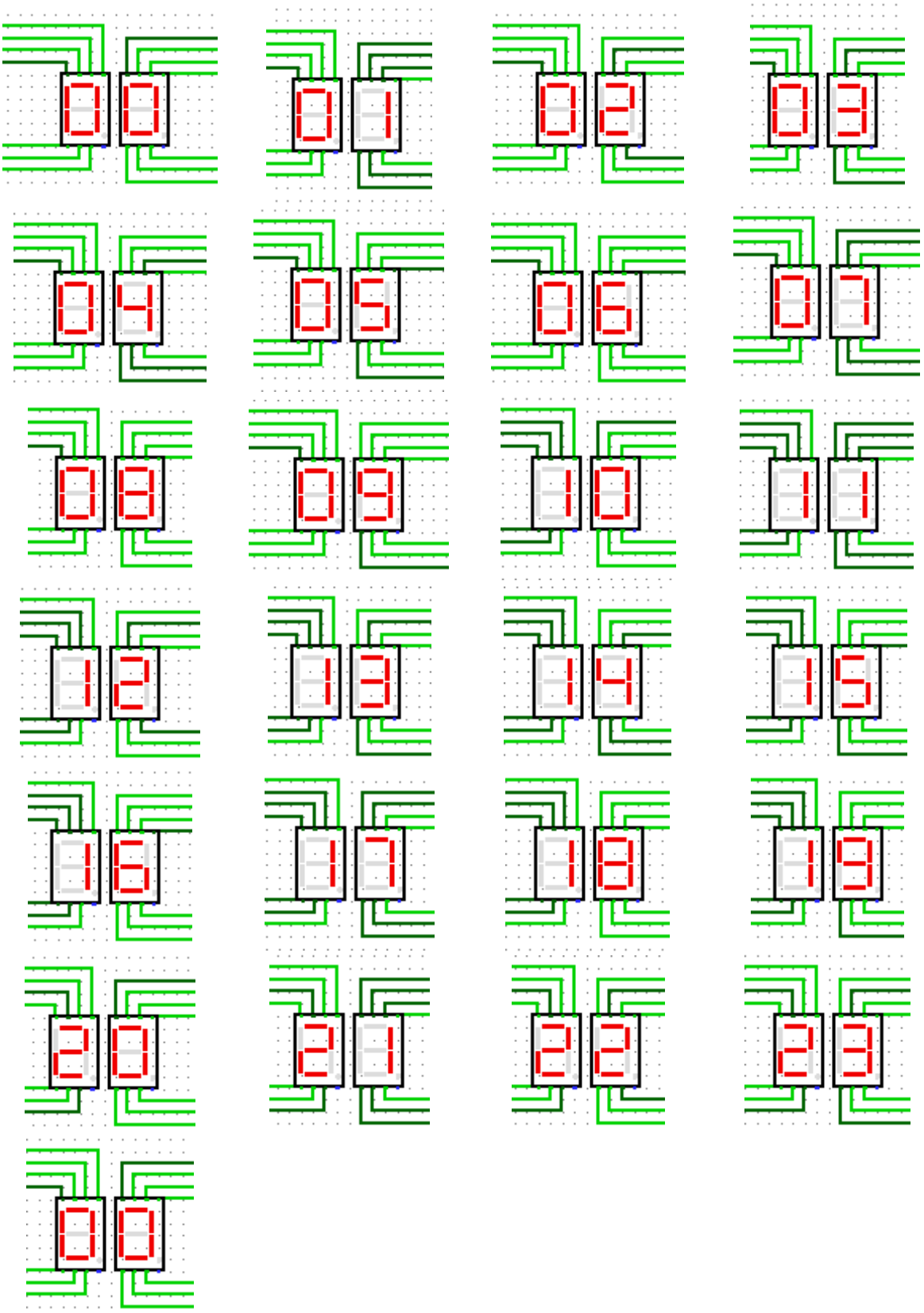


Figure 17: Simulation Results of Seconds

The figure 16 shows the simulation of the seconds. According to the button setup shown in the figure, the seconds is set to 24. That means it will go from 00 to 23. We can see it resetting back 00 after 23. At the same time, it sends a signal to the minutes section and increases the value of the minute by one.

3.3 MINUTES SIMULATION DESIGN

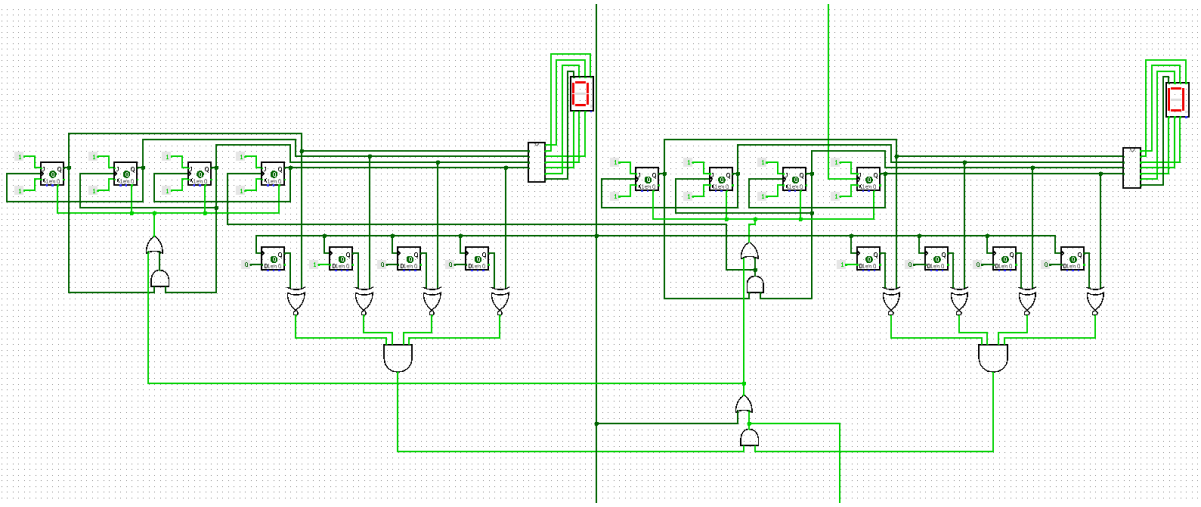
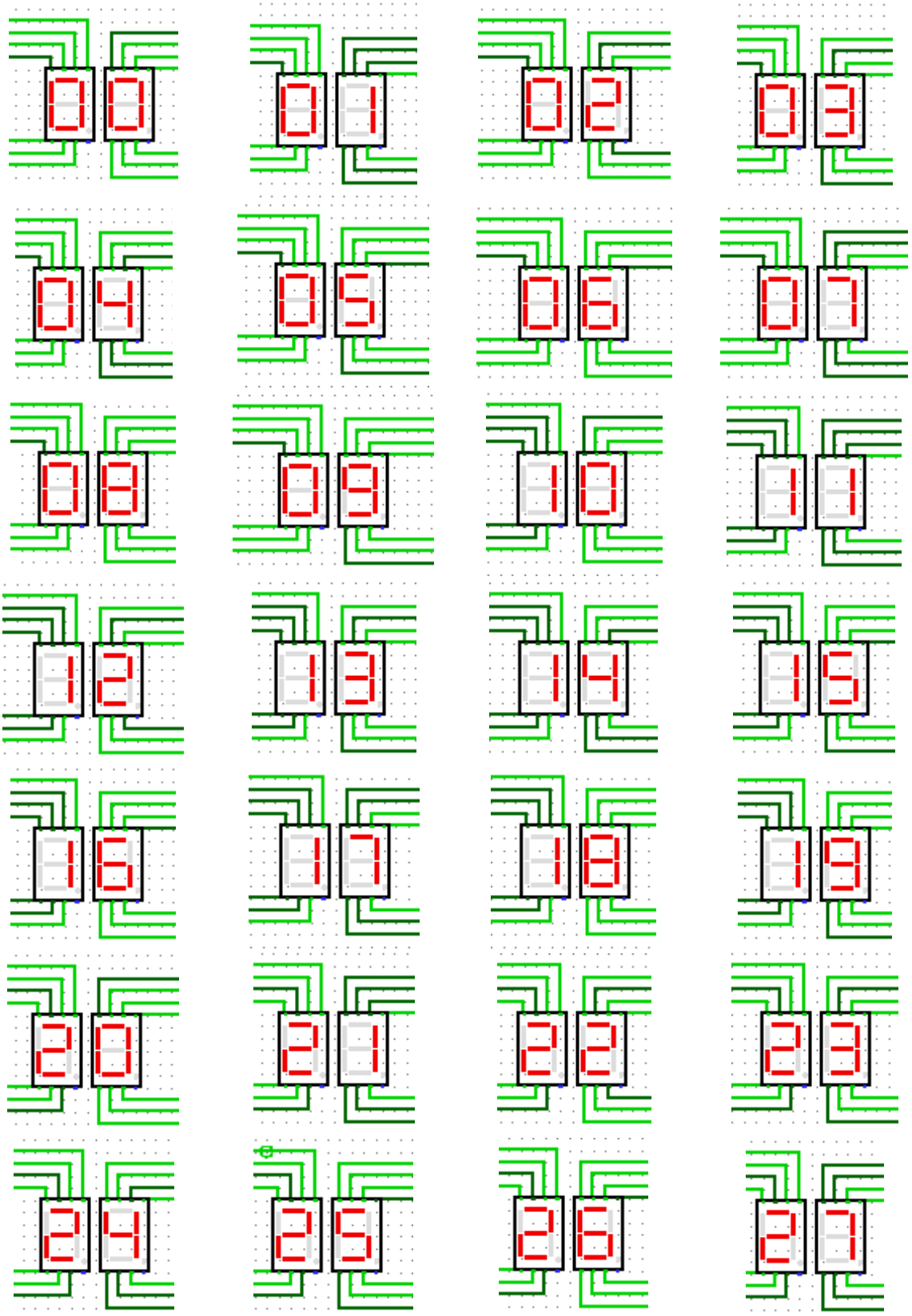


Figure 18: Simulation of Minutes Section

Figure 18 shows the minutes section of the clock. It is exactly the same as seconds section shown in Figure 16. The only difference is the clock signal in the clock (CLK) pin in the first JK flip-flop is coming from the seconds section instead of a signal generator.

3.4 MINUTES SIMULATION RESULTS



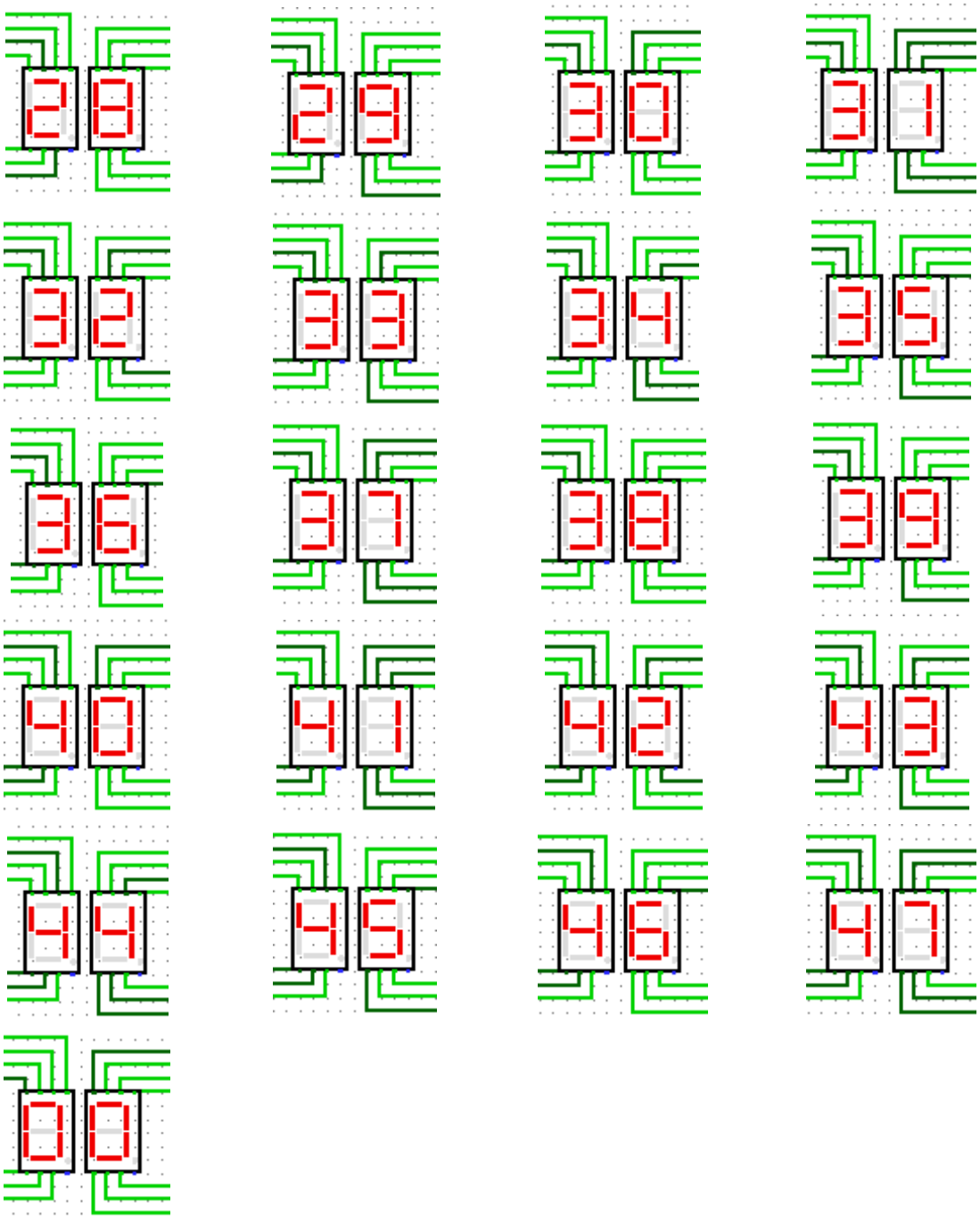


Figure 19: Simulation Results of Minutes

The Figure 19 shows the simulation of the minutes. According to the button setup shown in the figure, the seconds is set to 48. That means it will go from 00 to 47. We can see it resetting back 00 after 47. At the same time, it sends a signal to the hour section and increases the value of the hour by one.

3.5 HOURS SIMULATION DESIGN

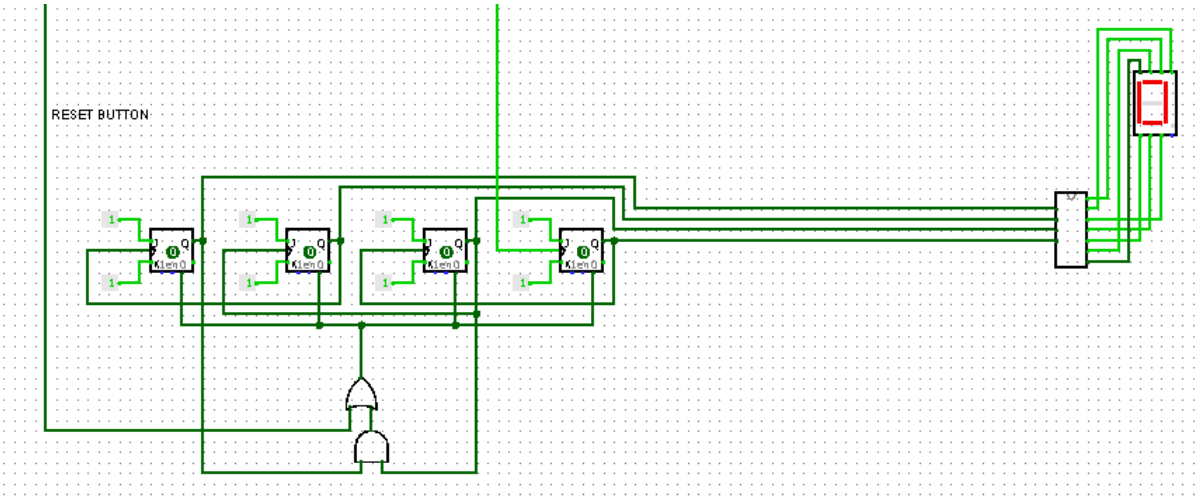


Figure 20: Simulation of Hours Section

Figure 20 shows the hours section of the clock. It is similar to the seconds design we have seen earlier. Seconds section had three reset conditions. But this section has only two. They are reset button and value becoming 10. We have seen both of them working in the seconds section.

3.6 HOURS SIMULATION RESULTS

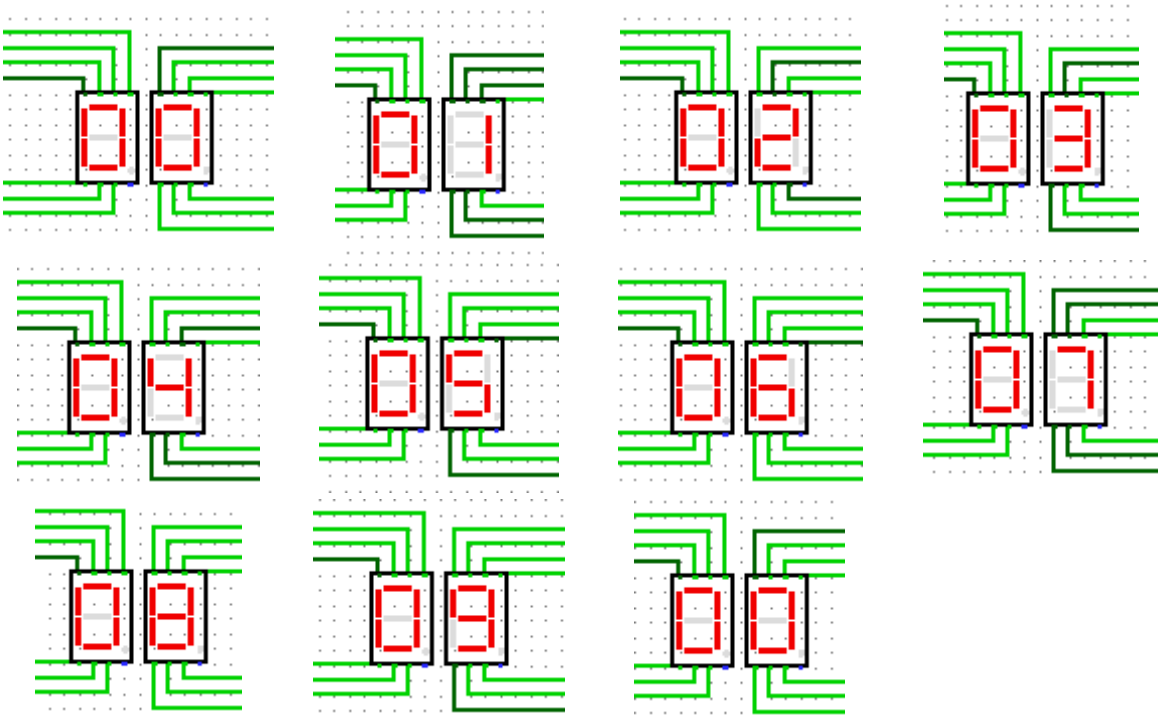


Figure 21: Simulation Results of Hours

The Figure 21 shows the simulation of the hours. Since I am designing a 10-hour clock according to the objective, the clock will go from 00 hours to 09 hours. We can see it resetting after 09 hours.

4. ADDITIONAL FEATURE

Objective of this assignment was to design and simulate a clock with 10 hours, any number of minutes and seconds. I went one step further and added the ability to configure the clock to any number of minutes and seconds as the user desire. In here, we will demonstrate how it works.

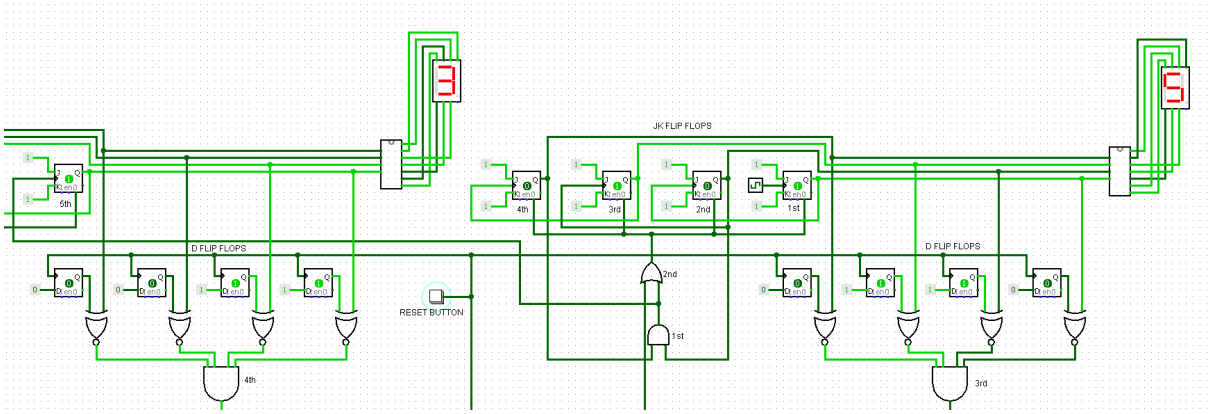


Figure 22: Additional Feature (1)

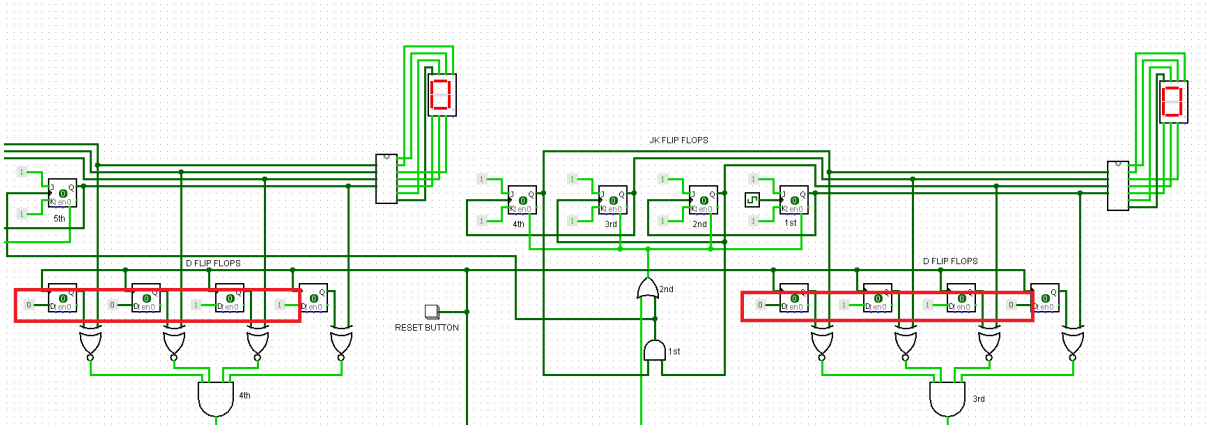


Figure 23: Additional Feature (2)

In the entire report, we have worked with 24 seconds per minute for this clock. But In Figure 23, we can see the switches highlighted. The values have been changed. The values of the switches read 0011 on the left which is 3 and 0110, which is 6. That means the number of seconds per minute will be 36. And we see it in action as the seconds resets back to 00 after 35. Just like that, user can configure the clock to any number of minutes and seconds. The changes will not apply until the user presses the reset button. Until they press it, it will just follow the old values. The D flip-flops are in positions just for this. Once they press the reset button, the clock will apply the changes and start from zero.

5. LIMITATIONS

The clock has an amazing additional feature. It allows the user to configure the number of minutes and seconds based on their need. They will use the switches to configure. But there is a problem. They can go beyond 9 for every display if they want to. 10 in binary is 1010. If they set the switches to any value above that, the clock will malfunction as it cannot display 2 digits in one seven segment display. The user has to be cautious of this issue and be careful when they are configuring the clock.

6. CONCLUSION

Working on this digital clock was a valuable learning opportunity for me. I am very grateful to the lecturer for giving us the opportunity work on such interesting and relatable project. During the course of this project, I gained a deeper understanding of sequential circuits and their importance in digital design. I now have a basic understanding of the logic behind the advanced digital machines and how they store data, gaining insight into the foundation of these digital devices that surround us.

During the process, I faced a few challenges. One of the biggest challenges I faced was to figure out how to freeze the values of the switches efficiently. But I was able to overcome this difficulty by learning about D flip-flop. The project not only enhanced my technical skills, but also helped me to appreciate the importance of persistence and perseverance when faced with technical difficulties.

7. REFERENCES

Analog Devices. (n.d.). *D Flip-Flop – What is a D Flip-Flop?* Retrieved From: https://www.analog.com/en/design-center/glossary/d-flip-flop.html. [Accessed Date: 5 February 2023]

Electronics Tutorials. (n.d.). *The JK Flip Flop.* Retrieved From: https://www.electronics-tutorials.ws/sequential/seq_2.html. [Accessed Date: 5 February 2023]

Fahad, S. (2022). *JK Flip-flop: Positive Edge Triggered and Negative Edge-Triggered Flip-Flop.* Retrieved From: https://www.electroniclinic.com/jk-flip-flop-positive-edge-triggered-and-negative-edge-triggered-flip-flop/. [Accessed Date: 5 February 2023]

Electronics Hub. (2015). Asynchronous Counter. Retrieved From: https://www.electronicshub.org/asynchronous-counter/ [Accessed Date: 5 February 2023]

Electronics Tutorial. (n.d.). *Exclusive-NOR Gate Tutorial*. Retrieved from: https://www.electronics-tutorials.ws/logic/logic_8.html. [Accessed Date: 5 February 2023]