COS10004 – COMPUTER SYSTEM

Assignment 1: Music Player

Student Name: Vi Luan Dang
Student Number: 103802759

A/ Introduction

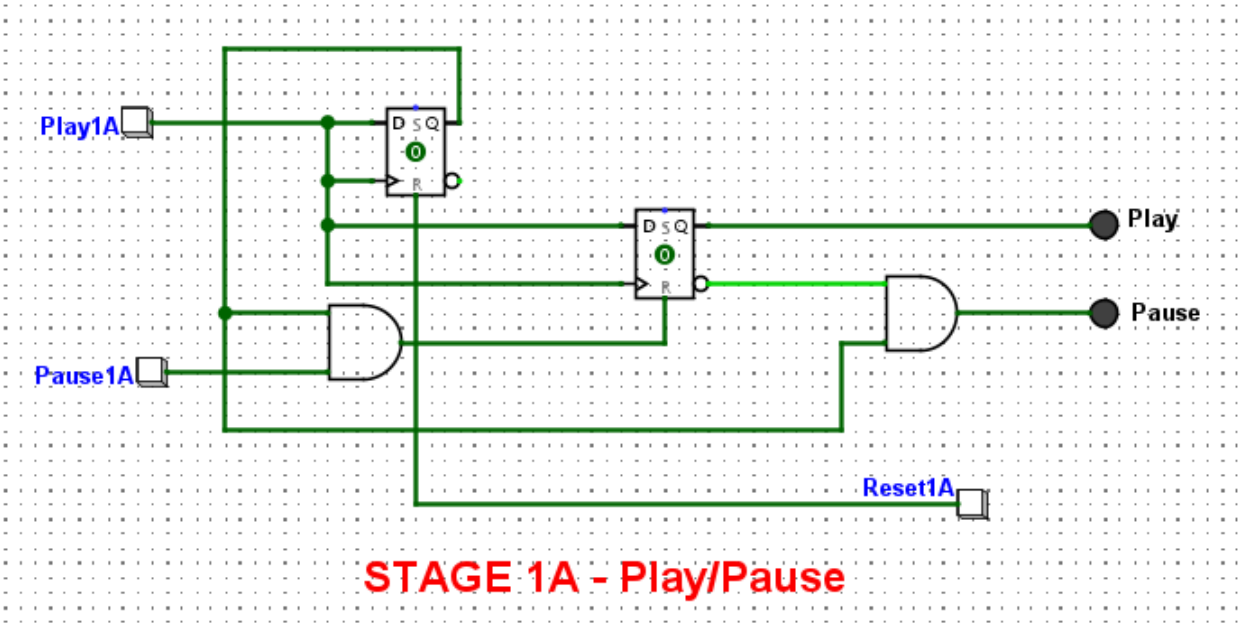
After 5 weeks of designing small and portable circuits as well as building other storage and conditioning implementation, we are tasked with designing a complete project for our Assignment 1 with the functionalities that represent a real-life music player. This document will provide a thorough overview of my process of creating the music player, the outline of my design , as well as several assumptions that I have made and obstacles that I have met.

B/ Description and outline of each circuit

The assignment is divided into stages and with each stage the difficulties rise, and the circuit grow more complex, therefore, in this part I will give a short description of the circuit and discuss them in detail.

Stage 1A

This stage was fairly simple as we only need to create 2 buttons, one for the Play state and the other for the Pause state. When the Play button is set to on the PLAY LED will also be turned on while the PAUSE LED will be turned off and vice versa. The only tricky part of this stage is that the Pause button will have no effect on the system if the Play button has not been pressed.



Picture 1: Stage 1A

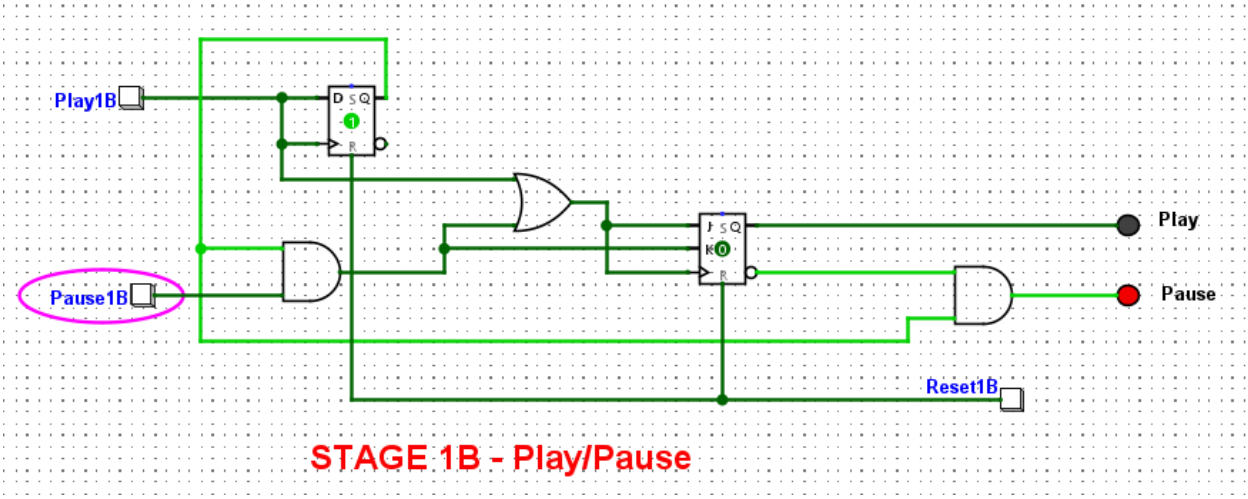
Before the play button is pressed both the LEDs will stay off the Pause button has no effect on the system. For the Play state, I used a D flip flop to store the value whenever the Play button is pressed, and the LED will remain on until the Pause button is pressed, which will reset the D flip flop.

There is also another D flip flop on the top of the circuit to check whether the Play button has been pressed or not, and the Pause button will only work if the Play button has been pressed previously. The 2 And gates in this circuit work as a logic-controlled switch for the condition above.

A reset button is added as a debug tool and therefore will not affect the functionalities of the circuit.

Stage 1B

Stage 1B is not so different from 1A, except for the fact that since this stage the Pause button will toggle between the two states if it is pressed and if the Play button has been pressed previously.



Picture 2: Stage 1B

The overall circuit of Stage 1A is reserved as it is not so different as Stage 1B. However, The D flip flop in the prior stage has now been replaced with a JK flip flop so as to enable toggling between the two states.

The wire for the JK flip flop is also worth noticing as the Pause button is linked to all the inputs of the flip flop but the Play button is only linked to the first input and the clock. This is implemented in order to provide the Pause button with the ability to toggle between two states and prevent the Play button from changing its state.

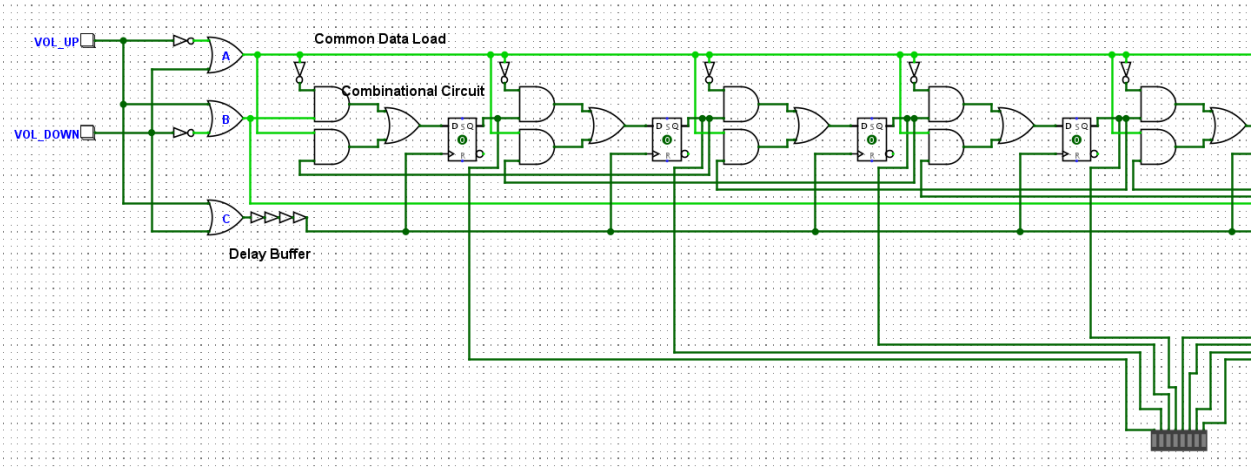
Apart from that, the D flip flop, And gates ,and the reset button still keep their functionalities and has not been changed.

Stage 2

Stage 2 is about implementing a volume control and display system; this system will not be affected by the Play and Pause buttons in stage 1. Because of that I have designed a separate circuit and will wire it to the Play/Pause system in later stages.

The assignment suggests that this circuit will have 8 LED bar showing the current volume level, a decrease and an increase button, and the volume level does not wrap around. Because of that I have planned to design an 8-stage PIPO bidirectional shift register. The circuit has 8 stages as we need 8 LED bar, it is bidirectional as we need an increase and a decrease button, and it is PIPO, which is parallel in/ parallel out, as we need to product parallel output after each button is pressed.

The full circuit will be submitted on canvas; therefore, I will extract a part of the circuit for better explanation and presentation, the circuit are as follows:

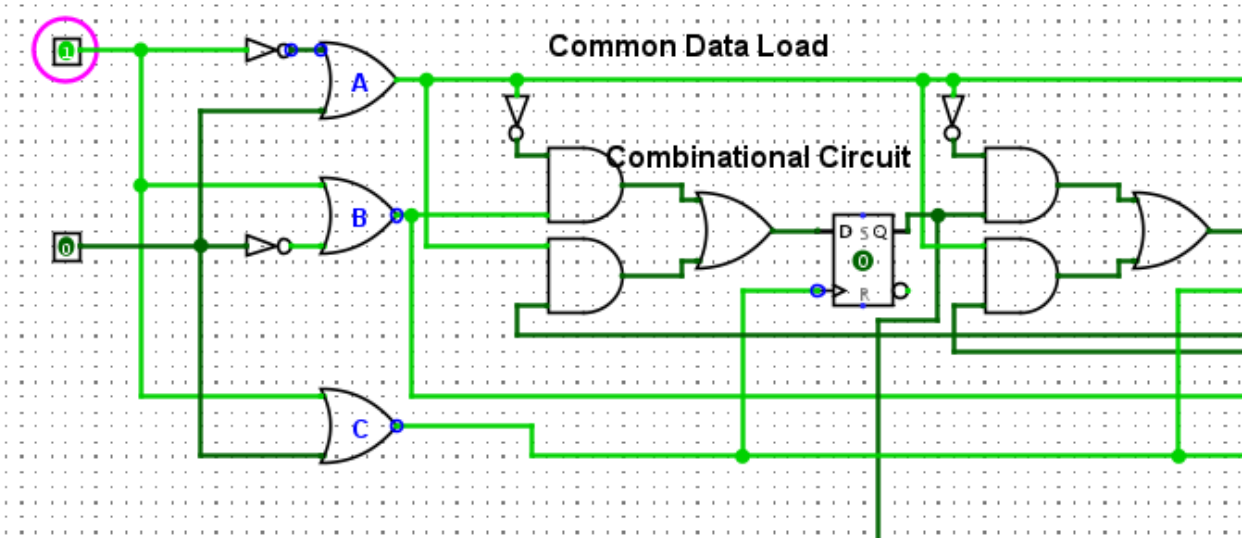


Picture 3: Stage 2B – Volume Control and Display

From the A Or Gate, we can see the Common Data Load, which is the wire that provide data for all D flip flop in our circuit, however, the data load to each flip flop is disable using a Not gate until the Vol_Up button is pressed.

Below the A Or Gate is the B Or Gate which serves the same functionality; however, in an opposite way. This gate is connected to the Combinational Circuit next to it and will disable the data load if the Vol_Down button is pressed.

Before we dive deeper into how the Volume LED bar is display using the flip flop, I would like to draw your attention to the C Or Gate and the Delay Buffers. The depth of this circuit from the Vol_Up button to the input of the first D flip flop is 6; however, the depth from the Vol_Up button to the clock of the first D flip flop is only 2 without the Delay Buffers. This would mean that the clock of the D flip flop would receive the signal before the input, making our intended output unreachable.

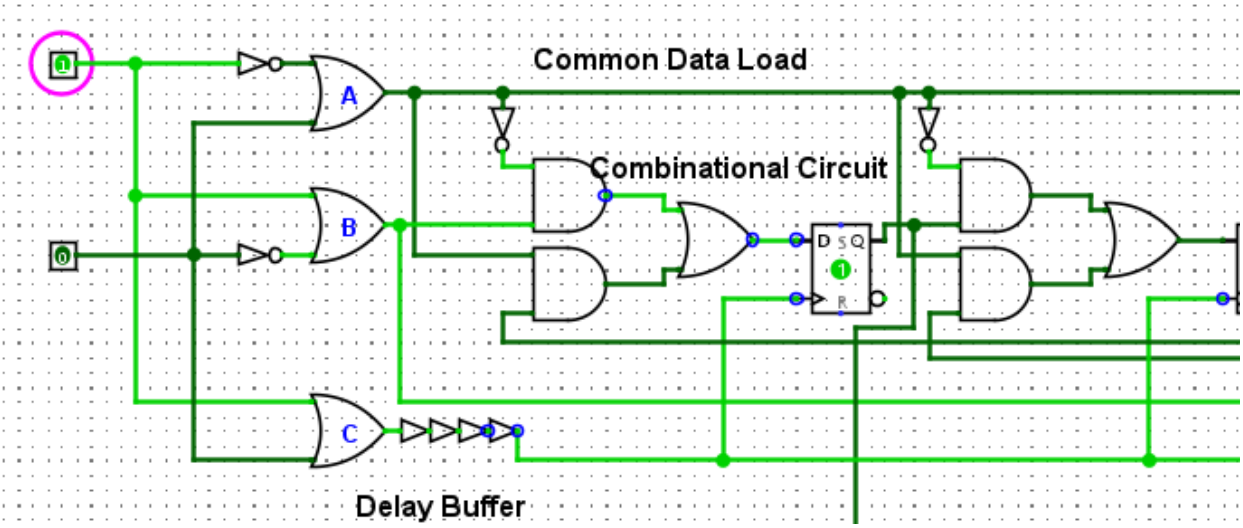


Picture 4: The Volume System without Delay Buffers

According to the D flip flop truth table, the Q output of it can only 1 if both the input and clock are 1; therefore, a Delay Buffer system is required to make our circuit work as it intended to be.

D	Clock	Q	Q’
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

Picture 5: D flip flop truth table



Picture 6: The Volume System with Delay Buffers

With that being presented, once the Vol_Up button is pressed the signal will be transmitted to the combinational circuit and with the Common Data Load is now on, the D flip flop will be turned on. This process stays the same for the rest of the remaining flip flops and that is my implementation of the increase system for the circuit.

Vol_Up	Common Data Load	Combinational Circuit	D flip flop
0	1	0	0
1	0	1	1

Picture 7: Increase volume system truth table

The process of decreasing the volume is not so different. The Vol_Down button is connected synchronously to all the clock inputs of the 8 D flip flop and with the Vol_Up button not being pressed the combinational circuit will also be off. Therefore, if we press the Vol_Down button, it will toggle the clock the D flip flop and turning it off.

Vol_Down	Vol_Up	Common Data Load	Combinational Circuit	Clock	D flip flop
0	0	1	0	0	1
1	0	1	0	1	0

Picture 8: Decrease volume system truth table if D flip flop is on

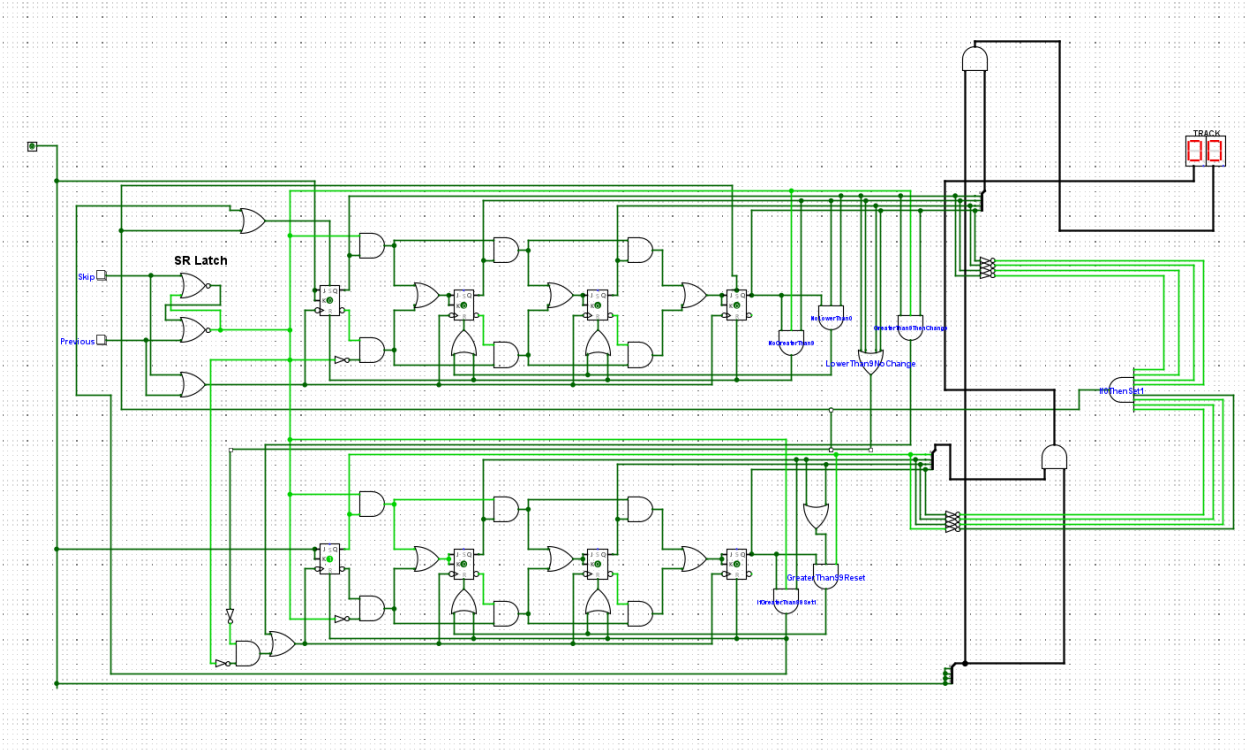
Each output of the D flip flop is then connected to a LED bar to display its state.

Stage 3

Stage 3 is the most difficult stage of this assignment; we are required to design a track skipping and display system for the music player. The system includes a two decimal digit display, a Next button and a Previous button, and the track skipping does wrap around. I have also designed the circuit separately so that it will not be affected by the Play/Pause system or the Volume system. The track will be, however, integrated with the previous system in later stages.

Because of the requirements, I have designed two 4-bit Up/Down Synchronous Counters, each of them will control a hex display and will be linked via various logic gates

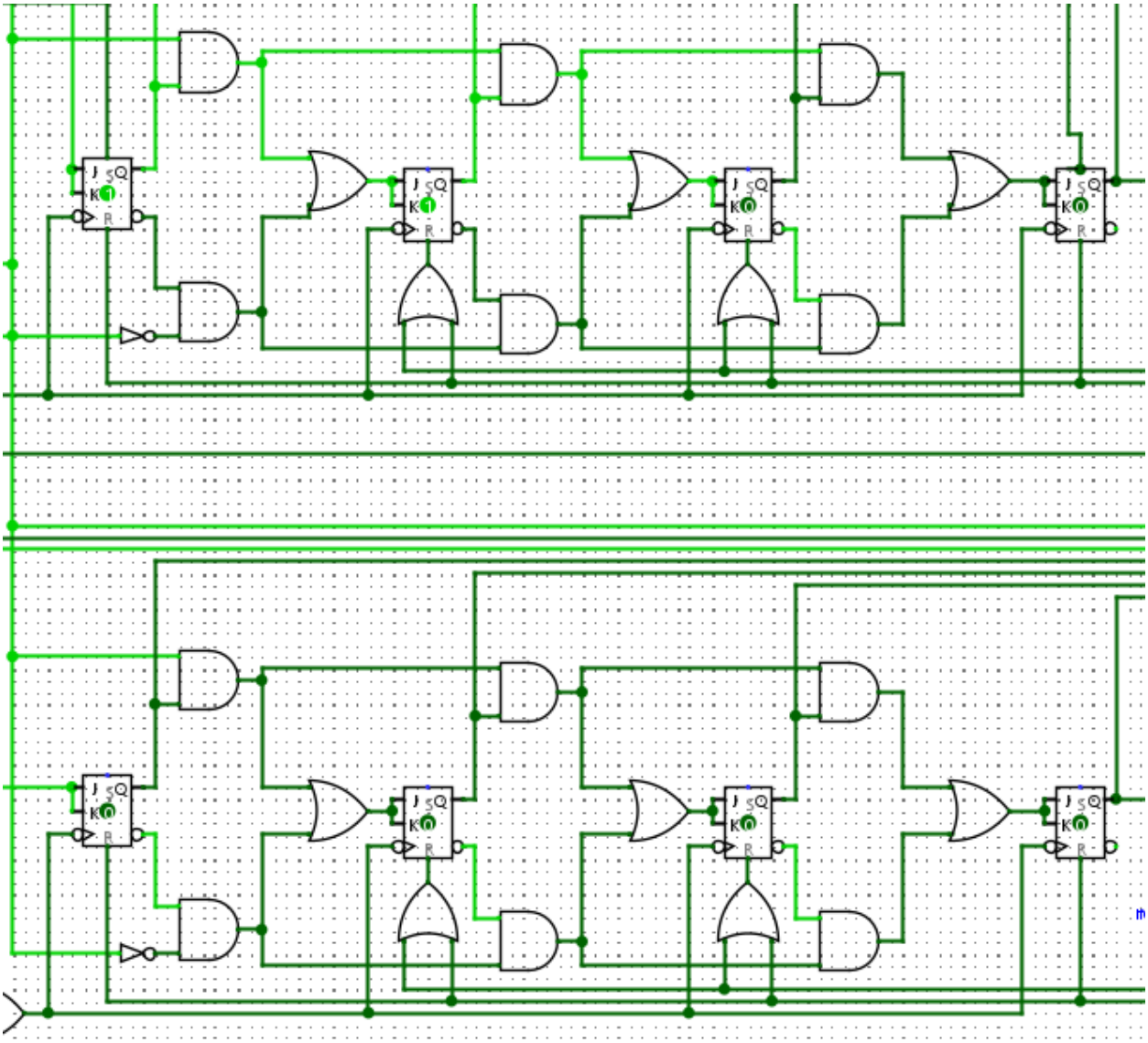
The full circuit will be submitted on canvas; therefore, I will extract a part of the circuit for better explanation and presentation, the circuit are as follows:



Picture 9: Stage 3 – Track skipping and display

The Skip button and Previous button are linked to a SR latch so as to control the up and down counting implementation. The Or gate under the SR latch acts as a synchronous clock for both the Skip and Previous button.

After the initial set up for the buttons, we come to 4 JK flip flop on top and another 4 JK flip flop on the bottom. The 4 JK flip flop on top control the units digit and the other control the tens digit.



Picture 10: 8 JK flip flop for the track counter

The combinational circuits between each flip flop act as a logic-controlled switch that will transmit the signal to the next flip flop only if the previous flip flop has been enable. Because of that these flip flop will toggle one after another. However, if there are no condition for these 8 flip flops then in theory:

- 4 flip flops on the top can represent $2^4 = 16$ digits
- 4 flip flops at the bottom can also represent $2^4 = 16$ digits

Enabling the hex digit display to represent from 0 to F each; therefore, there must be some conditions to restrict this from happening.

From the 4 flip flops on top there are 4 conditions that must be set:

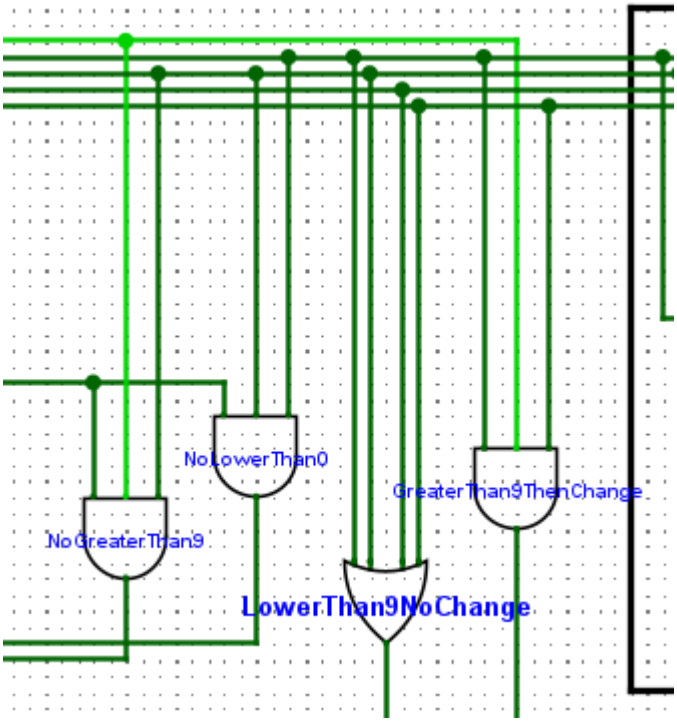
- The digit display cannot be greater than 9, if it does then reset all the flip flops.

- The digit display cannot be lower than 0, if it does then reset all the flip flops except for the first one (as the requirement does not allow the music play to display 0)

- If we decrease from 0 units digit then the 4 flip flops at the bottom has to be decreased also (if not the circuit cannot go, for example, from 10 to 9)

- If we increase from 9 units digit then the 4 flip flops at the bottom has to be increased also (if not the circuit cannot go, for example, from 19 to 20)

Because of that I have designed 4 logic-controlled switches as follows:



Picture 11: 4 logic-controlled switches

- The NoGreaterThan9 And Gate receives input from the Skip button, the first flip flop, and the fourth flip flop to check if the number is greater than 9 or not, it will reset all the flip flops if it is the case.

- The NoLowerThan0 And Gate receives input from the first, second and fourth flip flops to check if the number is lower than 0 or not (this case is usually from A to F when the number is greater than 9), it will reset all the flip flops except for the first one if it is the case.

- The LowerThan9NoChange Or Gate receives input from all the 4 flip flops on top and will disable the bottom flip flops if the number goes lower than 9.

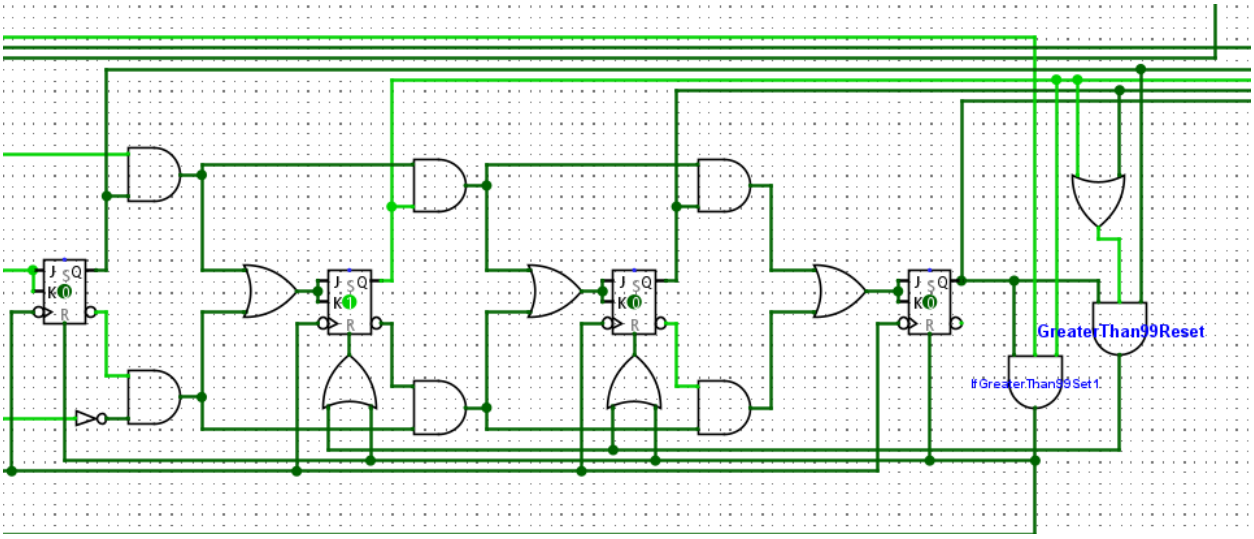
- The GreaterThan9ThenChange And Gate receives input from the Skip button, the first flip flop, and the fourth flip flop to check if the number is greater than 9 or not, it will connect to the 4 flip flops at the bottom to change to tens digit if the number goes greater than 9.

The 4 flip flops at the bottom also need some conditions:

- If the number is greater than 99 then reset all the tens digit.

- If the number is greater than 99 then switch on the first flip flop on top

Because of that I have designed two logic-controlled switch for those conditions.



Picture 12: 2 logic-controlled switches

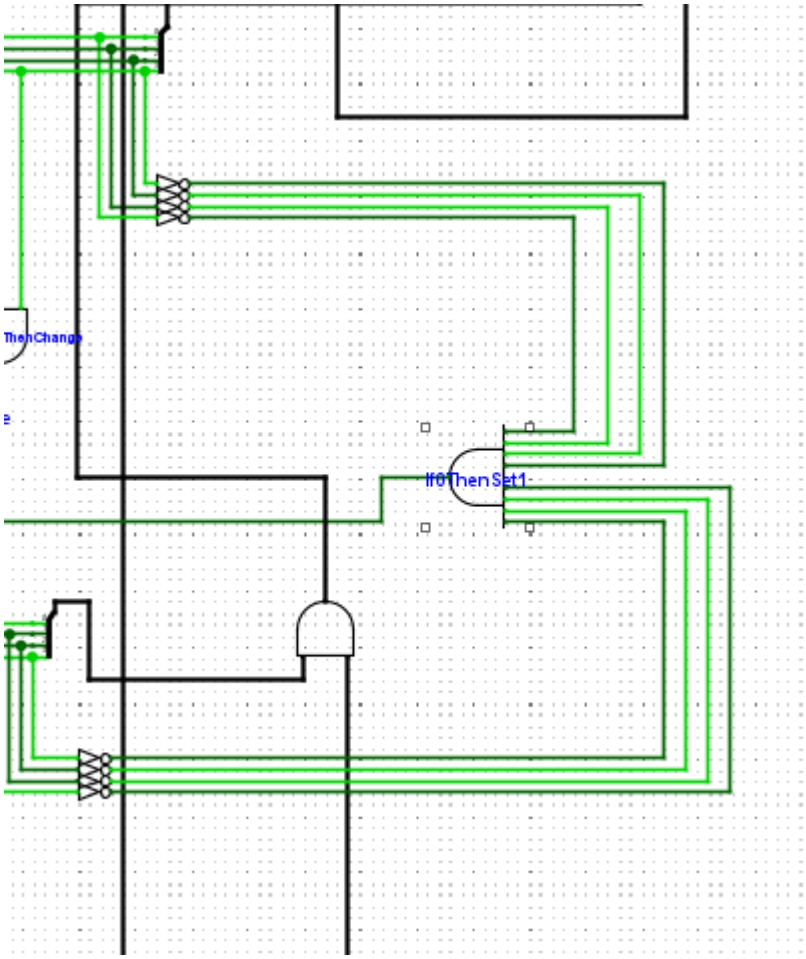
- The IfGreaterThan99Set1 And Gate will switch the first flip flop on the top on if the number is greater than 99.

- The GreaterThan99Reset And Gate will reset all the tens digit if the number is greater than 99.

Apart from that there are also two conditions that must be set, as the requirement of this stage does not allow for number 00 to appear, we need to implement other logic gates to prevent this. There are two situation that the number 00 may appear.

- When we increase from 99 to 00, The circuit will automatically skip one number to 01. This had been achieved via the IfGreaterThan99Set1 And Gate.

- When we decrease from 01 to 00. The circuit will automatically skip the 00 to jump to 99. If this happens then all the flip flops must be disable (because we decrease to 00). Because of that I have also designed a 8-input And Gate.



Picture 13: 8-input And Gate

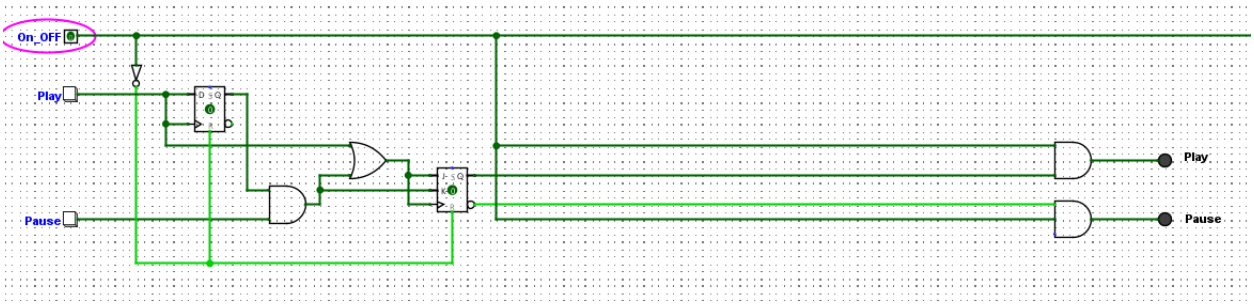
The gate will receive input from all the output of the 8 flip flops, and if they are all disable (meaning that we reach 0) it will be switched on and set the first flip flop on.

All the outputs of the flip flops are then connected to two splitters and the splitters are wired to the hex display enabling the track to move back and forth.

Stage 4 and Stage 5

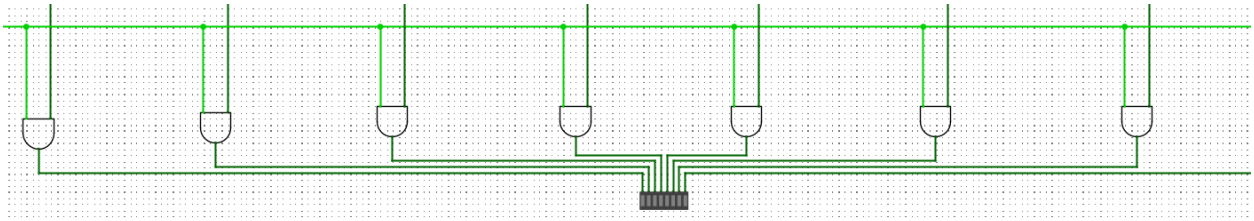
Stage 4 and stage 5 were relatively simple as we only need to connect all the other stages into one complete system and allow them to store information from previous setting and display them.

As the whole circuit is too enormous to display, I will only extract part of the modified circuit for better explanation and presentation.



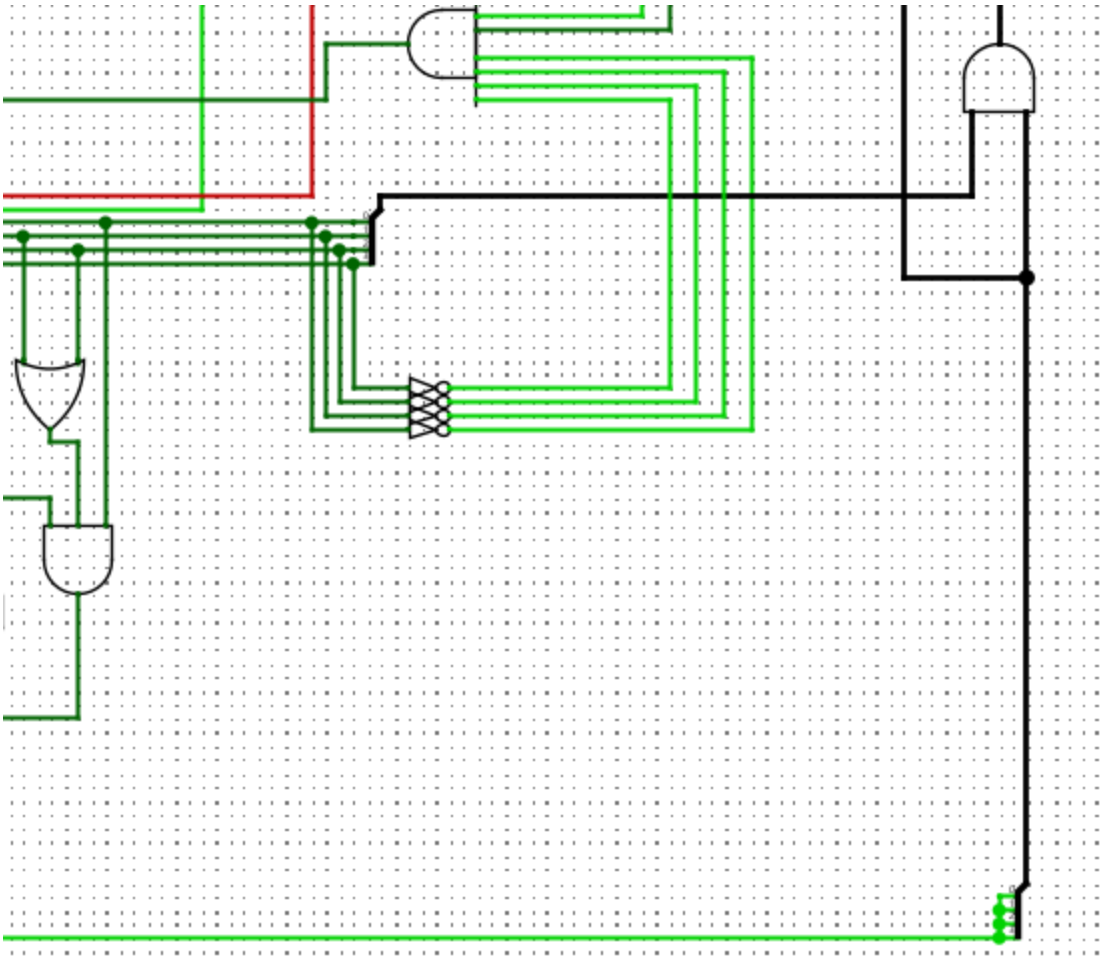
Picture 14: Play/Pause state with On/Off button

The On/Off button is linked to Pause and Play LED to prevent them from being switched on if the On/Off button is still off. Once the On/Off button is switched on the circuit will enter the Pause state, it will remain in the Pause state if the user does not toggle the Play button. Pressing the Pause button without playing beforehand will has no effect on the circuit.



Picture 15: Volume System with On/Off button

The volume system is connected with the On/Off button via And gates, the system will render unactive if the On/Off button is not switched on.



Picture 16: Track skipping system with On/Off button

Before the outputs of the 8 flip flops are wired to the hex displays, it must first connect to an And Gate with the On/Off button to check whether the power is on or not. This has satisfied all the requirements for stage 4.

As all the outputs of the Play/Pause system, Volume system ,and Track Skipping System are stored in flip flops or registers, they will reserve their state even if the power is switched off (only the display will be off, the registers still keep their state). Therefore, the track number and volume will be stored when the power is off and will be displayed again when it is on. This will mean that all the requirements for stage 5 have been satisfied.

C/ Unresolved problems

At the time being I do not see any room for improvement in my circuit, although because of lack of time, I had not been able to challenge myself doing the simulate playing section of the assignment.