# Assignment 1 Music Player

**Hai Hoang Le**
**SID: 103542974**

Due date: 23 October 2022

—

COS10004

Computer Systems

—

Dr. Minh Quang Nguyen

# Introduction

The following report will describe the development stages of an outline for a music player circuit.

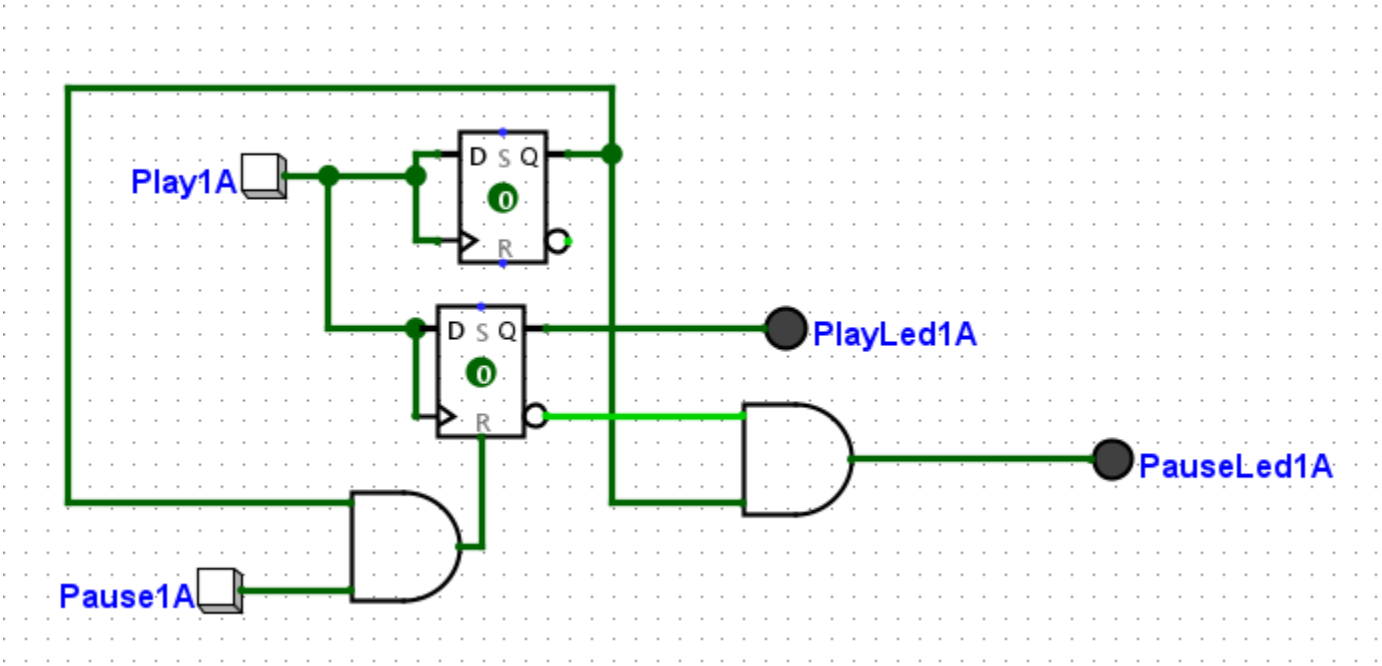
Following the requirement this circuit contains:

- On and Off buttons to control active status of circuit
- 2 led, 1 for On and 1 for Off status.
- Play and Pause buttons to stimulate play pause song
- Volume up and volume down button to stimulate increase, decrease volume
- 8 Led bar graph to indicate levels of volume
- Next Track and Previous Track buttons to stimulate skip songs
- 2 digit decimal display that indicates the track

# Circuit Describe

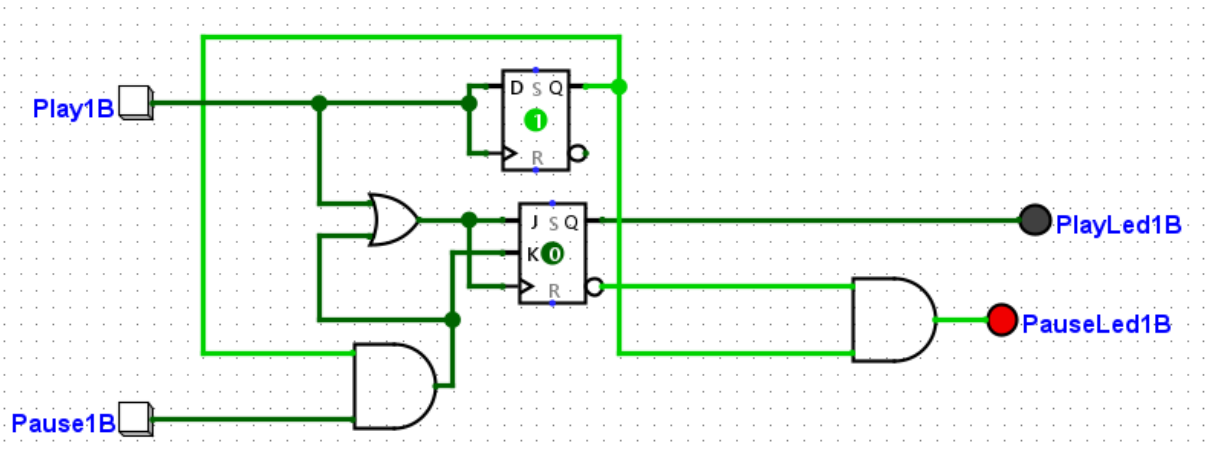
## Stage 1A: Implement the Play/Pause stage

To implement this stage, I used 2 D flip flops and 2 AND gates in order to control the signal. 1 D flip flop will store signal when Play1A is pushed and 1 D flip flop will control the leds. When Play1A is pushed the flip flop goes to 1 for both flips flops and activate the PlayLed1A. When Pause1A is pushed the flip flop which control the leds resets and turns to 0 and activate the Q', otherwise the other flipflop retains 1 for the Q. The PauseLed1A linked to an AND gate with the inputs of Q' and A that when Pause1A is clicked the PauseLed1A activates. Following the requirement, when the PlayLed1A is off, the Pause1A has no effect to the system, because of that I must used an D flip flop and an And gate to follow to statement.



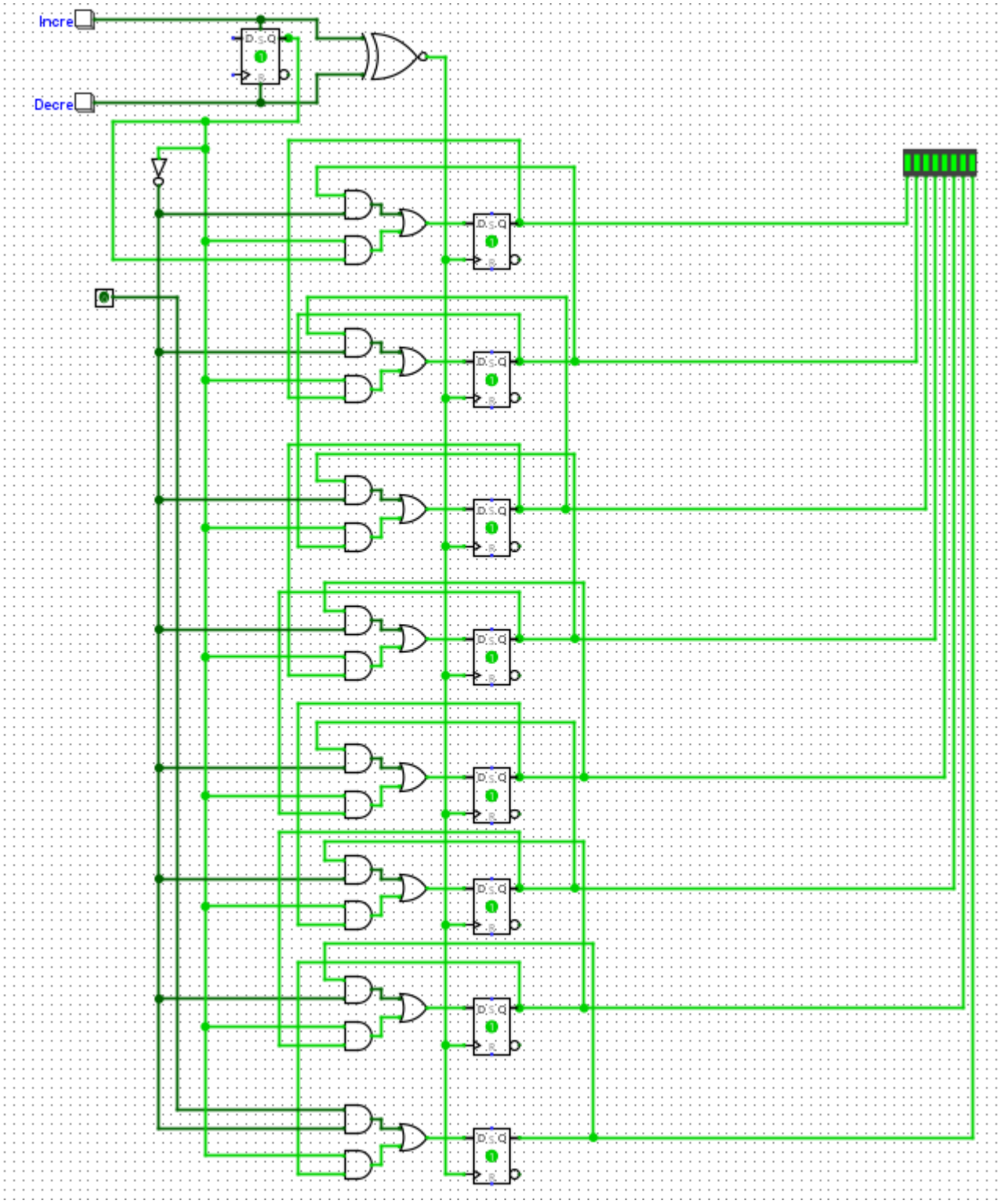

## Stage 1B: Implement the Pause to Play stage transition

As the same as stage 1A, the basic is still the same, however in stage 1B, I used a JK flip flop in order to control the leds and a D flipflop to store signal, an Or gate is added. With the JK flip flop I can control the valid output in my circuit.



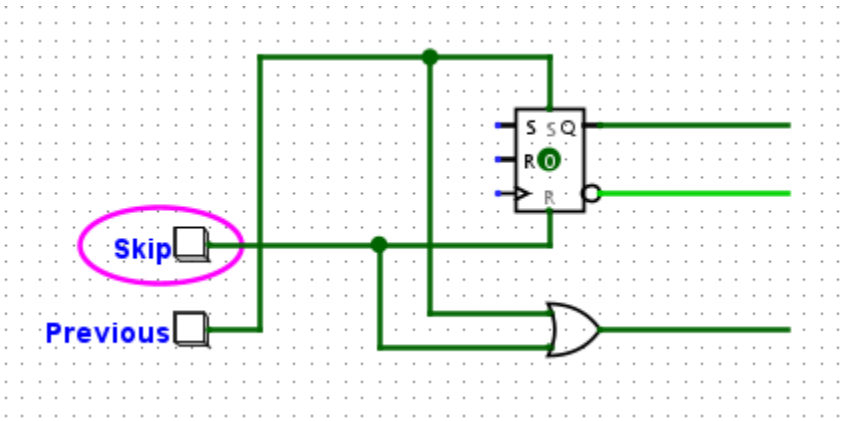
## Stage 2: Implement volume control and display

To implement this stage I used stack that I already learned in the week 4 as well as the lab exercise 5 which I implemented a 5 -bits bi-directional stack which and pop and push which each D flip flop creates a bit. Apply that to this stage which I have to implement a volume control with 8 levels, I used 8 S flip flops to create 8 bits that is the same as 1 volume level. Following to the statement, I have to used a led bar to displays the volume, so I used one and connect each input to each Q of each D flip flop so it can display the volume levels. To control the volumes I use 1 more D flip flop and an XNOR gate and connect them like below. The XNOR gate is used for whenever I click Incre or Decre button it will give signal to all the clocks so that the stack can run.

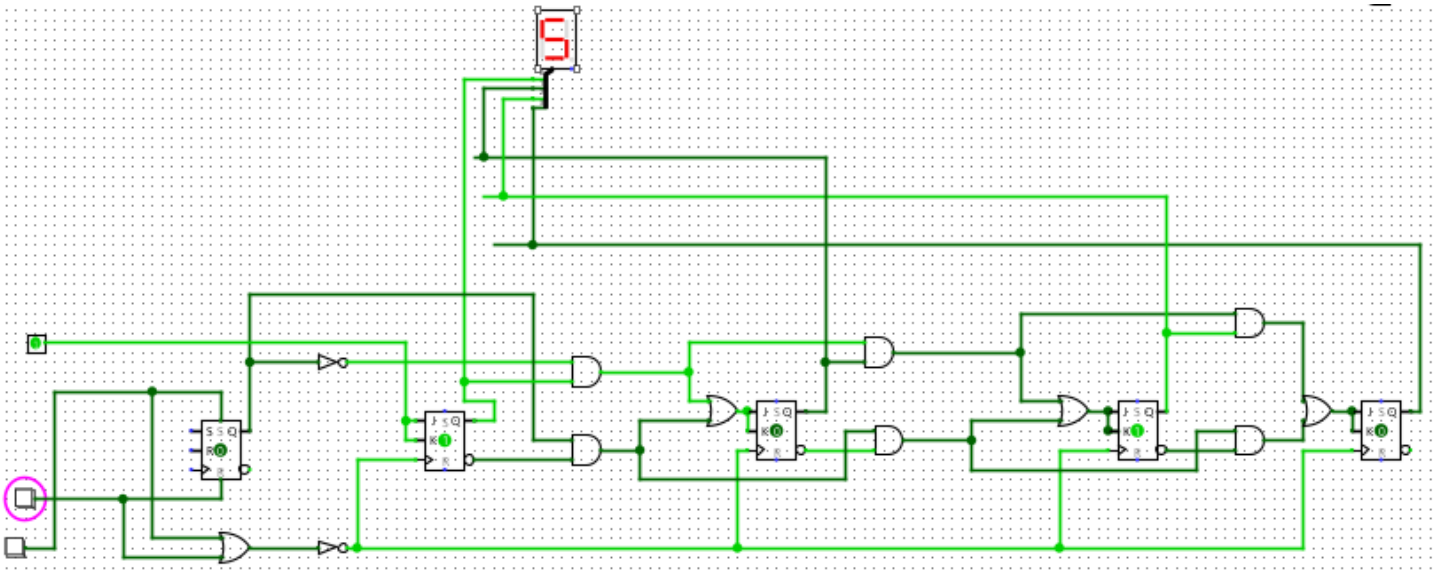



## Stage 3: Implement track skipping and display

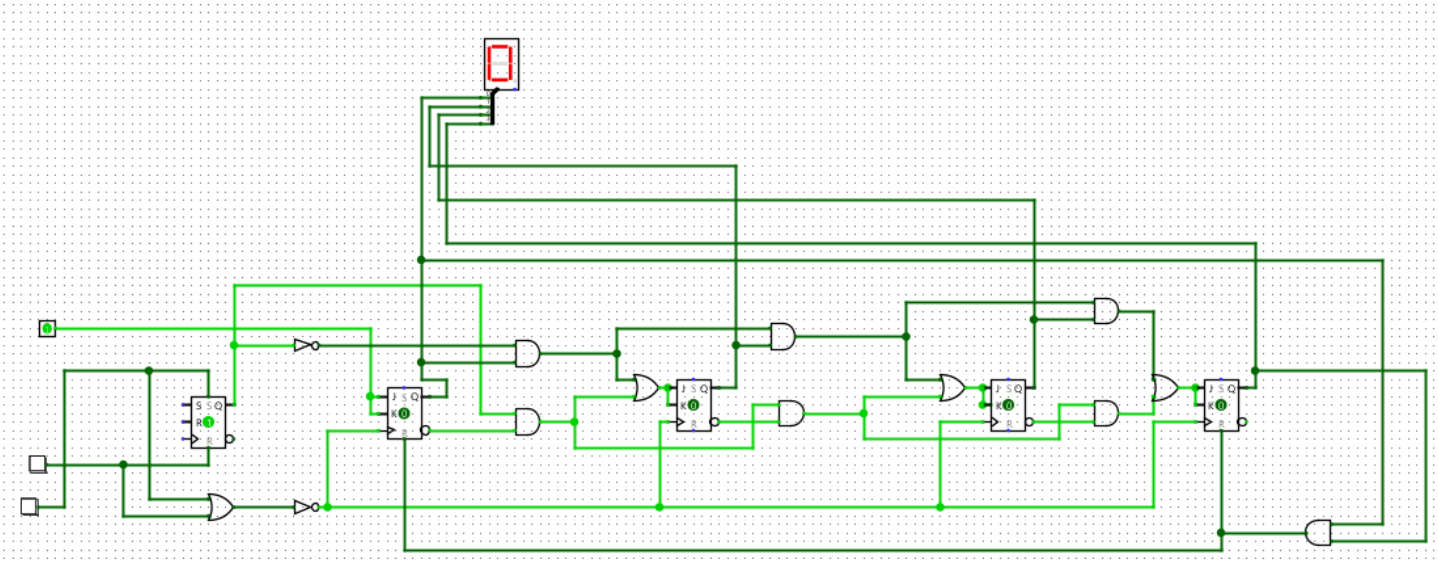
To implement the track skipping I apply the circuit that learned in week number 3 about the ripple counter, however it not enough for this stage 3, the statement requires 2 buttons: next track and precious track



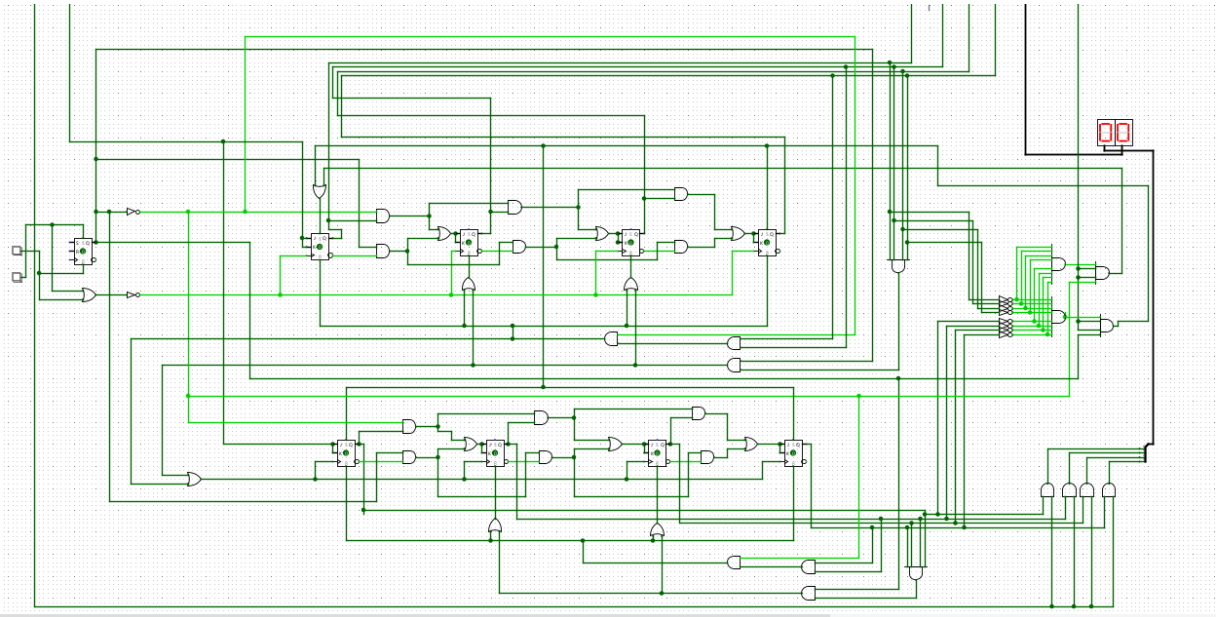

The control circuit of the counter created by an s-R flipflop and an OR gate which are linked like above. The Q output is responsible for the signal while the OR gate output indicates for all clocks pulse for the ripple.



Because we need 2 buttons: skip track and previous track that means the circuit needs to be bi-directional. Those AND and OR gates used for the bi-directional work smoothly. In the lab exercise 3, we implement a 3 bit counter (3 JK flip flops) that can count up to 8 (2^3) , however in this music player I have to count up to 9 which I must add one more JK flip flop ( 2^4 =16). But I don't need to count to 16, just 9 so I need to add some gate that when it counts up to 9 (1001) and will return back to 0.

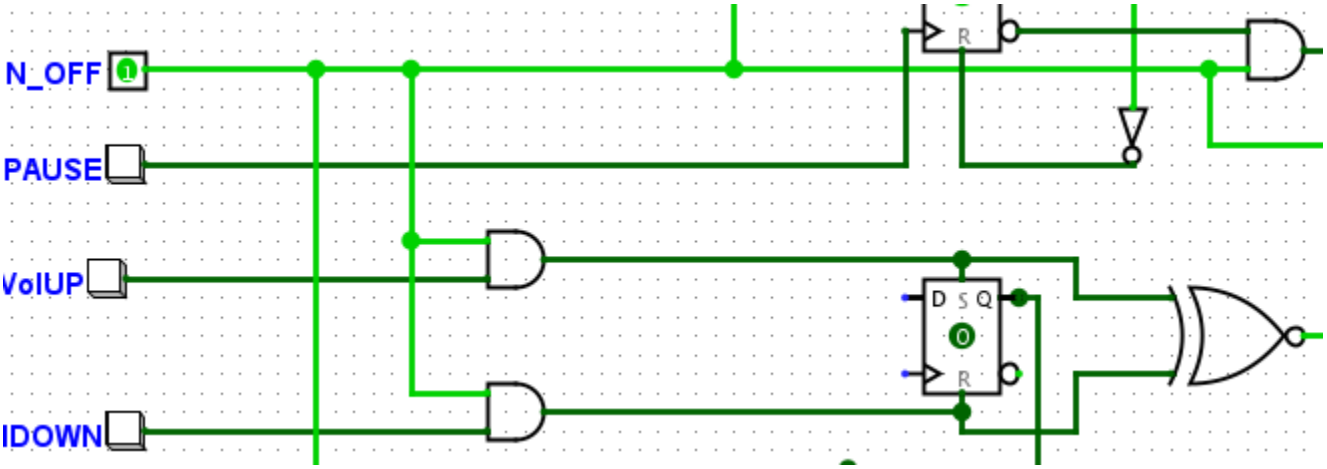


In order to limit the counter to 9 (1001) I need to add a trigger which in case is an AND gate connect with output Q for the first and last flop flops. When it goes to 9 and click the skip button, it will trigger the reset and counter will go to 0. To count to 99, I will add one more counter and add more gates so when the it counts to 9 it and it will trigger the second counter to count to 1, the process will continue until it goes to 99. In addition, some implement should added so when the counter goes to 99 it will trigger an reset back to 1.

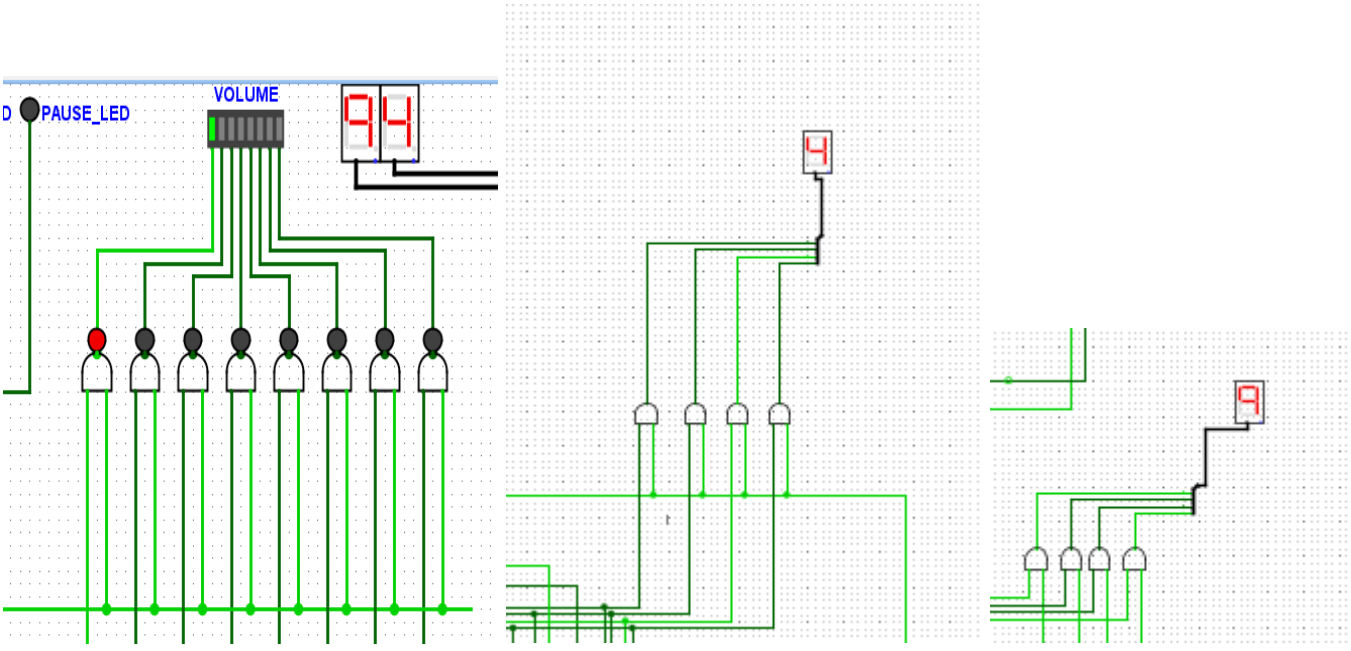


## Stage 4: Implement OFF state of the music player

Any device always having an off and on button which can indicate the whole device off or on state. In this music player circuit, I created a pin which has the role of controlling the state of this music player. When it in the off state, the leds display for Play and Pause buttons stop as well as the volume controller will not work, cant not increase or decrease the levels. The gate that I used here is adding AND gates which a input wire up with the On\Off pin, and the output link to the flip flop, and the displays devices.




These 2 AND gates will prevent user to volume up or down whenever the ON/OFF pins is off, they can not change the state of each flip flops of the stack in off state.

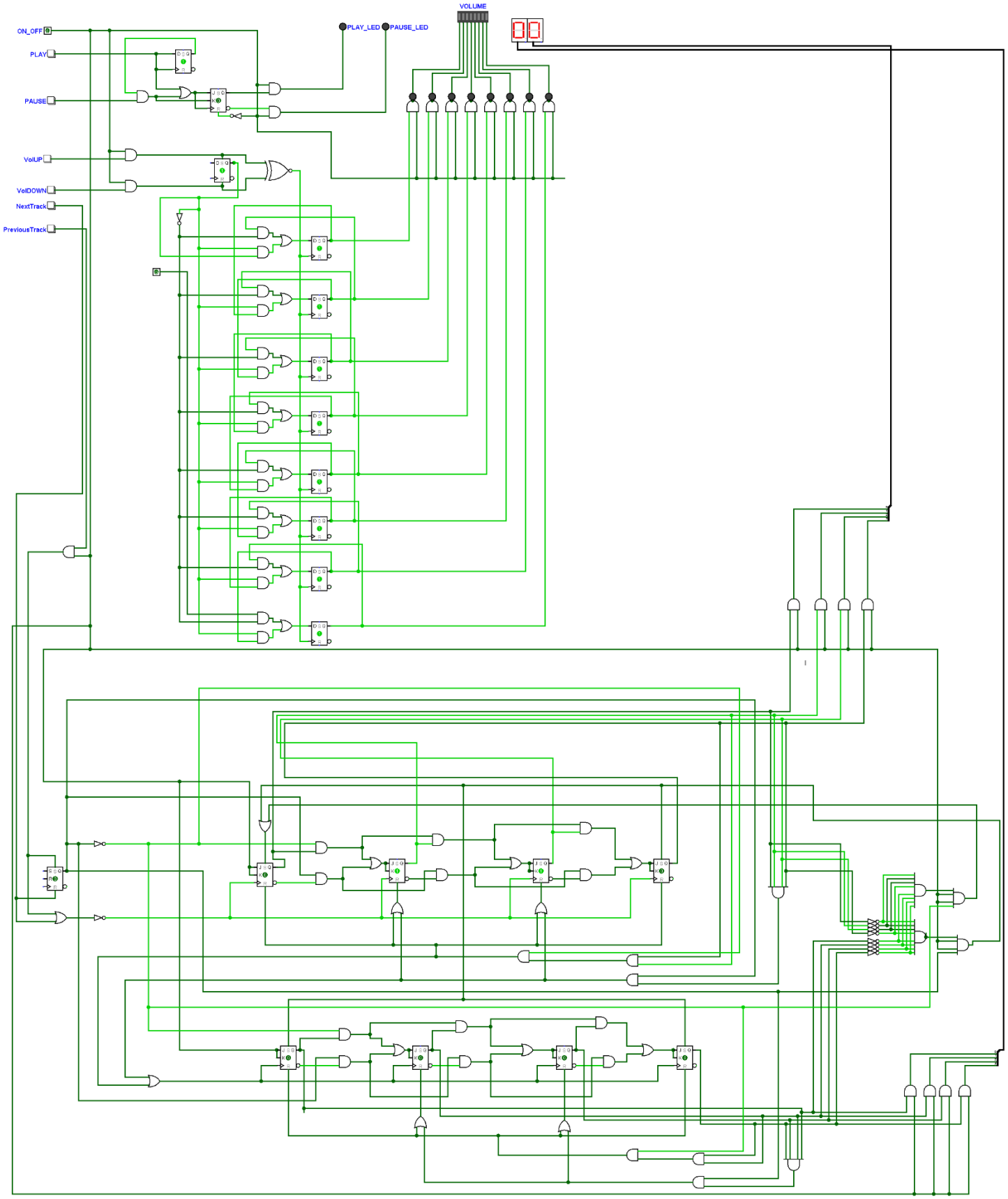


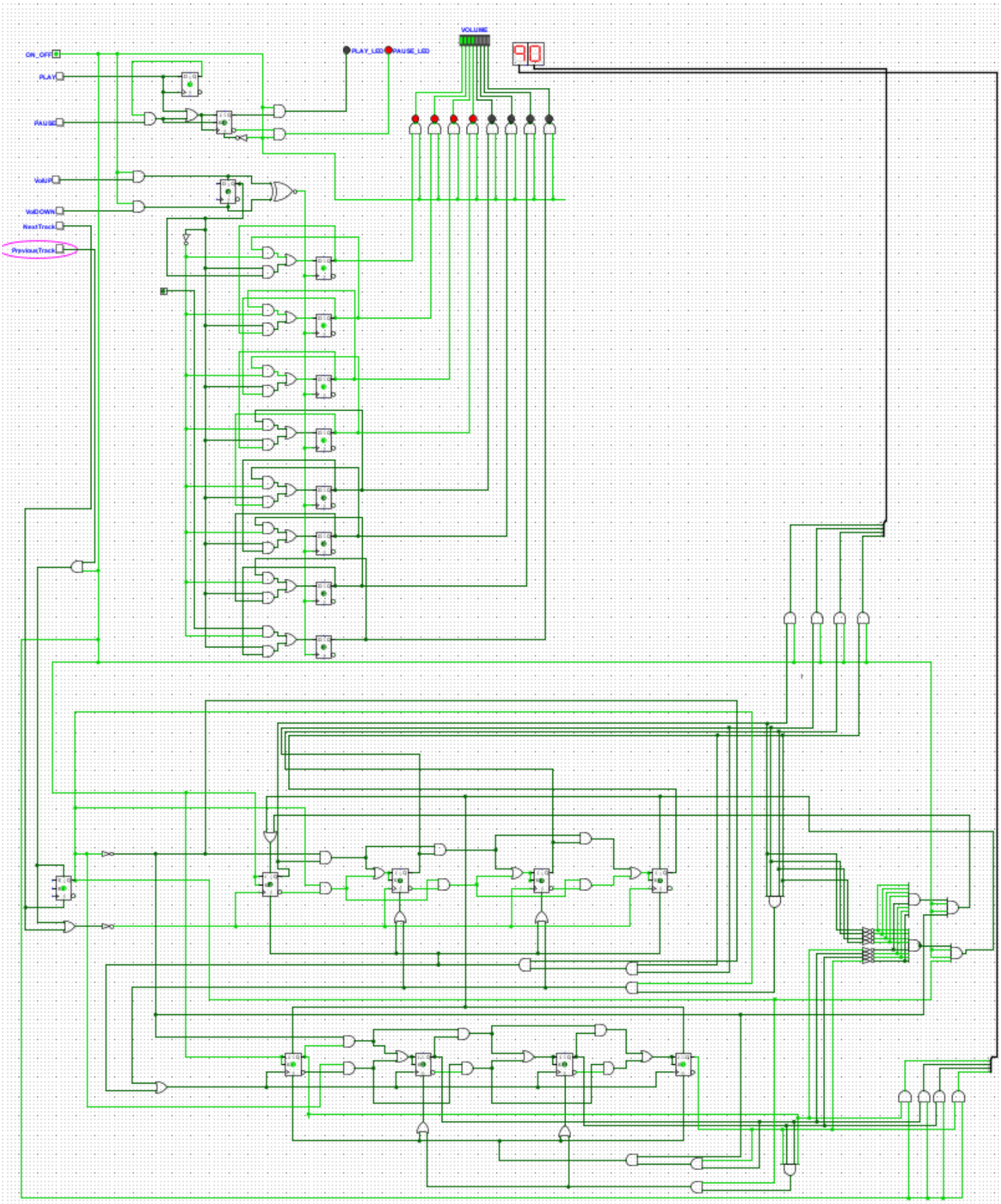
These AND gates will deny these leds to be light on when the circuit in off state.

## Stage 5A/B: Implement storage of previous settings for your player

By using the method of adding AND gates in the circuit. It prevents user to change the volume levels or track in when the player is off. The values of all the flip flops remain so when the turn on the player again, the circuit can keep the previous setting.

# Screenshots




ON_OFF
PLAY
PAUSE
VolUP
VolDOWN
NextTrack
PreviousTrack
PLAY_LED
PAUSE_LED
VOLUME
90