Trac Duc Anh Luong - 103488117

Assignment 1

Circuit Description

This report is the documentation of my logic circuit for a music player's user interface, which includes:

- An ON_OFF button
- 2 PLAY and PAUSE buttons
- 2 VOLUME_UP and VOLUME_DOWN buttons
- 2 track skip buttons: NEXT_TRACK and PREV_TRACK
- An 8-LED bar graph indicating 8-volume settings
- A 2-digit decimal display, using the Logisim Hex display, that indicates the track number
- 2 LEDs displaying PLAY and PAUSE modes

Components that the circuit used:

- Gates
- Wires
- LEDs
- Hex displays
- Buttons
- Splitters
- Flip-flops
- Pins

Design Outline

1. ON/OFF

The ON_OFF button is located at the top of the circuit. It is wired to other LEDs and Hex displays through AND gates to ensure that those components are turned off when the music player enters the OFF state (for the Hex displays, the data bits of the AND gate are adjusted to 4 and the number will be set to 00).

2. PLAY/PAUSE

There are 2 PLAY and PAUSE buttons, as well as 2 PLAY and PAUSE LEDs.

- When the PLAY button is pressed, the system enters the PLAY state:
    - the PLAY LED is turned on
    - the PAUSED LED is turned off
- When the PAUSE button is pressed, the system enters the PAUSED state:
    - the PLAY LED is turned off
    - the PAUSED LED is turned on

- When the music player is in the PAUSED state if the PAUSE button is pressed it will return to the PLAY state if it was in a PLAY state prior, otherwise, it does not affect the PLAY state.

The PLAY and PAUSE buttons use a JK flip-flop and a D flip-flop. The JK flip-flop is there to toggle when the PAUSE button is pressed while keeping the PLAY state unchanged as further PLAY button presses will be ignored. The D flip-flop is there to make sure the music player enters a PAUSE state when switching from ON to OFF, as well as ignore PAUSE button presses if the music player was not in a PLAY state prior.

3. VOLUME UP/VOLUME DOWN

The VOL_UP and VOL_DOWN buttons are used to increase and decrease the volume of the music player. These buttons used an 8-bit bi-directional shifting stack to sequentially “pop” and “push” the stack. When VOL_UP is pressed and the clock is running, the stack will increment until it reaches a maximum of 8. When VOL_DOWN is pressed and the clock is running, the stack will increment until it reaches a minimum of 0. To control the volume so that we can either increase or decrease it by one level, we will use a button to serve as our clock. The buffer is there to make sure that we can delay the signal as our flip-flops store the data from the previous press. If there is no buffer, the volume will be unstable as it can increase or decrease several levels with each button press. As a result, the LED lights will display 8 volume settings. As the volume does not wrap around, pressing VOL_UP when the maximum is reached will not affect the state of the music player. Similarly, if minimum (zero) volume is reached, further VOL_DOWN button presses are ignored.

4. NEXT TRACK/PREVIOUS TRACK

The NEXT_TRACK and PREV_TRACK are used to skip tracks and display the current track number. This action can be done both forward using the NEXT_TRACK button and backward using the PRE_TRACK button. For this stage, I used 2 synchronous 4-bit counters to display each of the digits from 0 to 9 (as the track number is between 01 and 99). I used JK flip-flops configured to operate as toggle, giving a maximum count of zero (0000) to fifteen (1111) and back to zero again. Similar to the volume settings, I implemented a custom clock using a button and buffers to delay the signal and keep our track counter in a stable state. Then the 4-Bit counter advances upward in sequence (0, 1, 2, 3, …, E, F) or downwards in reverse sequence (F, E, …, 3, 2, 1, 0). To produce a count sequence from 0 to 9, after reaching the count of “1001”, the counter recycles back to “0000” by resetting the flip-flops through the “next” AND gate. To produce a count sequence from 9 to 0, after reaching the count of “0000”, the counter recycles back to “1001” by resetting the 2 middle flip-flops through the “prev” AND gate, therefore, instead of “1111”, we get “1001”. For the track to wrap around (pressing NEXT_TRACK when the display is at 99 brings the counter back to 01 and pressing PREV_TRACK when the display is at 01 brings the counter to track 99), we need to cancel out the 00 display case.

a. 99 to 01:

I wired the second and fourth bit of the tens hex display to the AND “one” gate, with a pulse from our NEXT_TRACK button. Therefore, when the value is 1010 (or 10 in decimal), we will

reset all the flip-flops wired to the tens display (so the display for the tens digit will be 0000) and set the first bit of the ones hex display, making it 0001 instead of 0000.

b. 01 to 99:

I wired all the outputs of 8 flip-flops and the pulse from the PREV_TRACK button through a NOT gate to a 9-input AND gate. If all the inputs are 0, the output of the AND gate will be 1, setting the first and last bits of the ones hex display, making it toggle from 0000 to 1001, or 9 in decimal value, skipping the value of 00 and displaying 99.

Assumptions

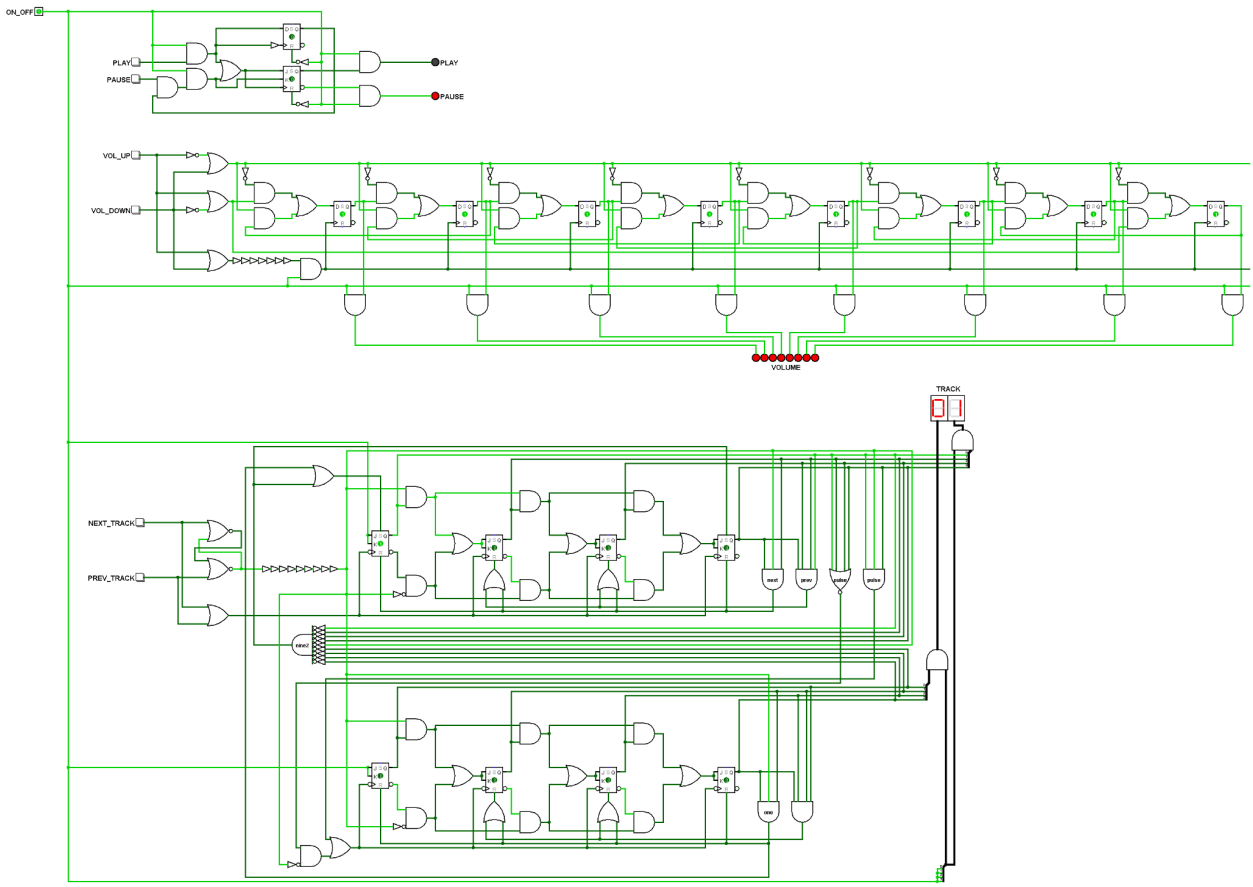
No assumption was made.

Unresolved Problems

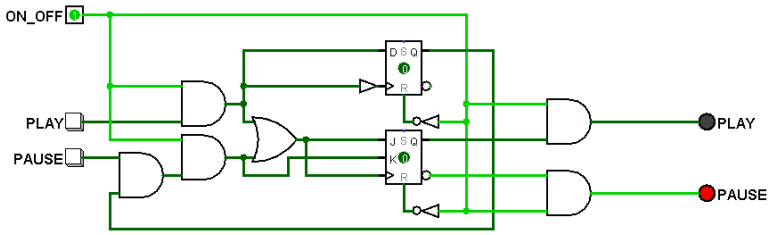
No unresolved problem.

Pasted Screenshots

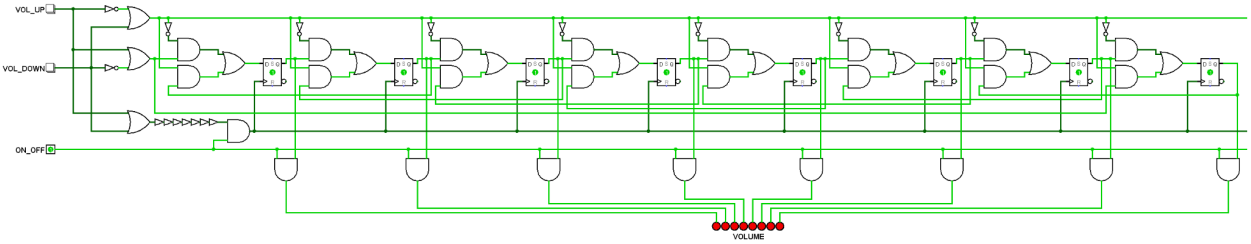
THE ENTIRE OUTLINE



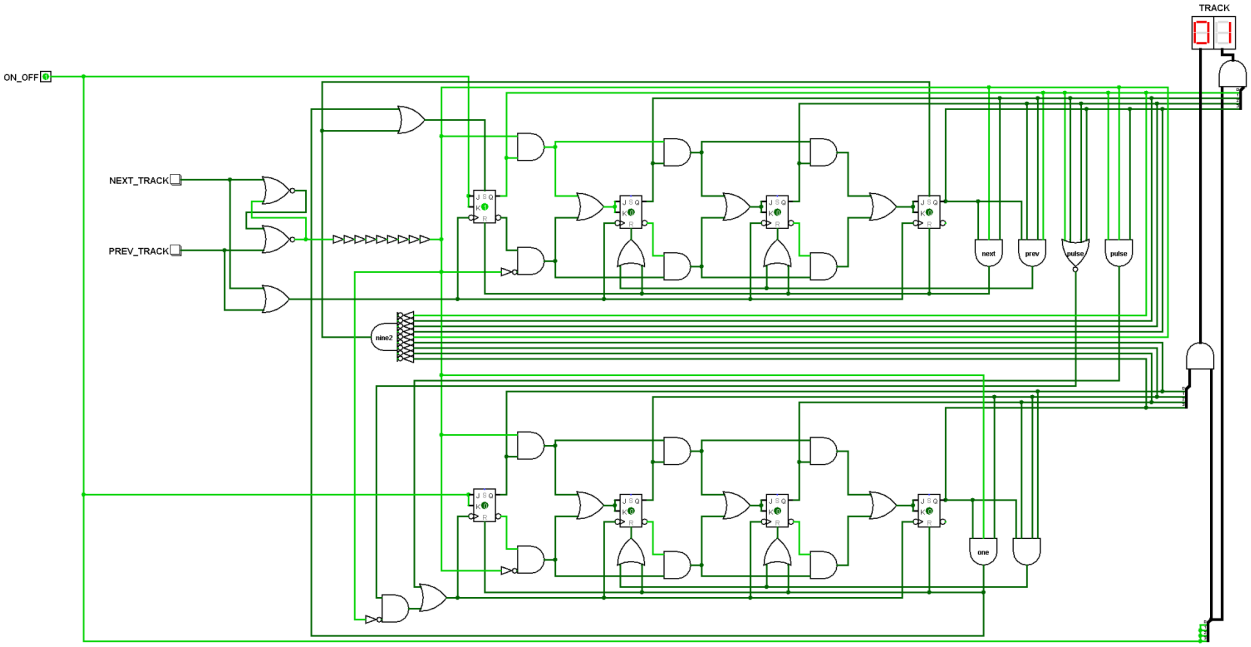
PLAY/PAUSE



VOLUME UP/VOLUME DOWN



NEXT TRACK/PREVIOUS TRACK



References

Neso Academy 2015, ‘3-Bit & 4-bit Up/Down Synchronous Counter’, www.youtube.com, viewed 14 October 2022, <https://www.youtube.com/watch?v=svFUEJkoeVY>.

‘Synchronous Counters’ 2013, Basic Electronics Tutorials, viewed 14 October 2022, <https://www.electronics-tutorials.ws/counter/count_3.html>.

‘Bidirectional Counters’ 2013, Basic Electronics Tutorials, viewed 14 October 2022, <https://www.electronics-tutorials.ws/counter/count_4.html>.

‘MOD Counters’ 2016, Basic Electronics Tutorials, viewed 14 October 2022, <https://www.electronics-tutorials.ws/counter/mod-counters.html>.