**SWINBURNE UNIVERSITY OF TECHNOLOGY**

COS10004 – Computer Systems

Logo

Description automatically generated with medium confidence

ASSIGNMENT 1: MUSIC PLAYER

Name: Lê Xuân Nhật - SWH00994

Class: Monday morning

# Design outline:

The circuit has a pin that acts as a power button, which controls all others button and output behavior. Without the power all displays and buttons are disabled. There are two other pins that represent the play button and the pause button, with two separate LEDs for each state.

Next, the volume controls include “++” and “—” buttons, implementing R-S flipflop as the control to increase and decrease.

Finally, the track counter utilizes the track counter design which skips “00” state whether it’s 99 wraps back to 01 and vice versa.

# Circuit description:

First, we look at the power pin. It connects to almost every input and output with an AND gate to ensure that no signal will be sent or LEDs will be turned on during the OFF state.

The PLAY/PAUSE control use the programmable gates in Week 4 lecture for the bi-directional stack, as stage 1 require the PLAY pin as the input pin and PAUSE as the selection pin. Without the PLAY pin, the PAUSE pin has no effect. However, moving on to stage 4, the assignment stated that the default stage when turning on without PLAY/PAUSE input should be PAUSED, so I put several more programmable gates to ensure the functionality.

The VOLUME control behaves similarly like the bi-directional stack in Week 4. However, the control changed to buttons connecting to R-S flipflops used in Week 2, with added buffer to delay the signal so that several input will not be received at once, making the circuit jump bits. The two buttons also connected to clocks of D flipflops, with even more buffer so that the VOLUME control increase/decrease one by one.

Perhaps the most challenging task is meeting all the requirements of stage 3 – Track Counter. First of all, I searched for a bi-directional counter. I found this page depicting the synchronous 3-bit Up/Down Counter (link: [Bidirectional Counter - Up Down Binary Counter (electronics-tutorials.ws)](https://www.electronics-tutorials.ws/counter/count_4.html)). After that, I upgrade it to a 4-bit bi-directional counter, with MOD A(10) on the count up and MOD F - > A on the count down. The MOD A method is similar to the MOD 6 on the week 3 counter. The MOD F to A, however, require a little bit more critical thinking. The 4-bit counter with HEX display will display the count down like the following order: 0, F, E, D…, A, 9, … So, we have to skip from F to A. F (hexadecimal) in binary is 1111. So in order to skip from F -> A (or jump straight to 9) we should reset the two middle JK flipflops when it is in the stage of 1111 to 1001 (using AND gate, link it to the Rs of flipflops) so that it will be a 9. After finishing one 4 – bit bi-directional counter with proper MODs, I attempted to connect 2 counters together: one for the first digit and one for the second digit. My initial thought was that the number occupying the ones jump 10 times up/down, the number in the tens also moves (by 1 only). So the only thing I have to connect is the MOD of the first counter to the clock of the second counter. However, this has proven to be dysfunctional due to the fact that there are several errors when testing the circuit. Additional components had been added: I have to create seperate gates to update the clock for the second counter to ensure the syncronization of the system. Also, I added an OR gate to the MOD F to A system due to various cases the counter jump to A (1010) or B (1011), so the condition of every single bit equals one (1111 = F) did not occur so the AND gates of the MOD F did not work. After finishing the first initial design, I remove the use of pin and clock in favor of buttons. Once again I implement the use of R-S flipflop (mentioned in the volume section of this report) so that each button at the same time send a clock signal and decide the input for the count up/count down wire. Nevertheless, there is one final hurdle: the assignment requires when the track number wrap around, there is no “00” state. The transition of 99 straight to 01 is quite simple: connect the MOD A of the second counter (responsible for the tens) to the S of the first flip flop of the first counter (responsible for the ones). So that when 99 moves to 00, the circuit dectects and it jumps one bit. The hard part is how to create the transition from 01 straight to 99. At first, the thought process was using an AND gate to detects when the circuit enter “00” states it set both counters to 99 (connect the output of that AND to S of the flipflops so that both counters jump from 0000 to 1001). The reality is if implemented mentioned circuit, the default state when opening the file is not “00” but “09”, and no “music player” should work like that. Therefore I use a D flipflop to store the bit when entering “01” state and activate its when it goes down (pressing the PREV button), and reset it (connecting the NEXT button to R) when it goes up and enter the “01” state to ensure there is no error when counting up.

The use of stack in volume control and the use of 4-bit bi-directional counter in track counter cover stage 5A already, so turning on/off will not affect the memory. To make sure this circuit also fullfill the requirements of stage 5B, I added AND gates to every output so that it will not display anything when turn off.

Finally, I want to mention the simple display for minutes and seconds that play when PLAY is pressed, and pauses when PAUSED is pressed, and resets when the next or previous track is selected. The counter can go only up to 5 minutes and can only play when you enabled tick in the logisim application, with the frequency set to 2HZ.

Overall, the task has been very challenging at times, but with all the help of my associates, including both from K3 and K4 students as well as the internet, I have solved all of the problems as well as fullfilling all of the requirements that the assignment demands. This has been an enjoyable experience for me as well and I am grateful for Swinburne for my time as a student here.

# Screenshot:

Diagram

Description automatically generated