

# Computer Systems Assignment 1

Daniel Coady (102084174) – 12:30 Wednesday

25/08/2019

## Stage 1

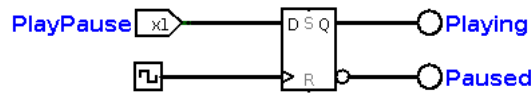


Figure 1: Circuit used for stage one

For this stage I have assumed that we are to use a single pin/button to toggle on or off the music player. Because of this I have used a single D flip-flop with the outputs Q and Q' connected to the playing and paused outputs respectively.

## Stage 2

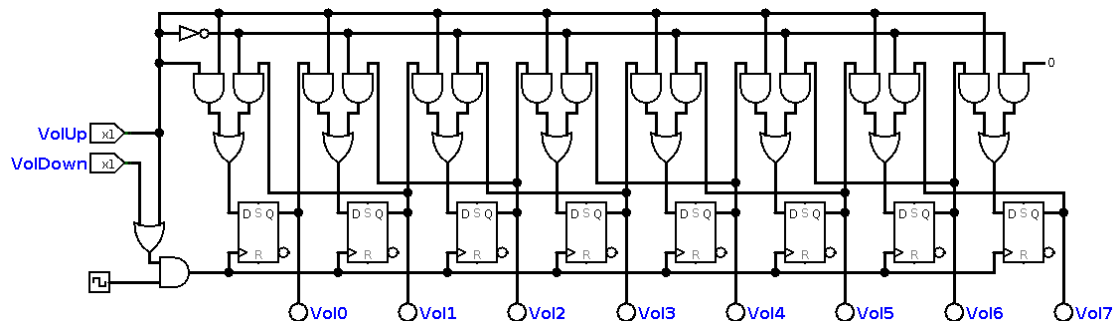


Figure 2: Circuit used for stage two

This stage was a considerable amount more complicated than the last, and it has a circuit to match. For this I assumed that the pins/buttons could continuously make the volume go up or down. With this assumption in mind, I found it to be most appropriate to use a bi-directional shift register to control volume values. This is for two reasons:

- It allows for easy raising or lowering of the required bar graph display
- It inherently does not allow for overflowing or underflowing values

Of course, this isn't just a bi-directional shift register. I've made some modifications to the inputs of circuit so that it may fit our application better. There are two inputs to this circuit: the volume up and volume down pins. Both of these pins are connected to an or gate which feeds into an and gate that also has the clock as the input. This allows us to control when we allow the clock pulse to feed into the circuit so that it can hold state and not cause any unexpected behaviour.

This design is not without flaws however, since if you put both inputs on then it will simply continue increasing the volume. I did attempt to fix this by replacing the or gate

with a xor gate, but this had it's own issue. When the xor gate was in use, occasionally when you go from having both inputs on to just having the volume down input on it would go all the way to zero from any state, which is not behaviour that we want. This may be solvable with a latch to time the inputs to the clock, but I haven't been able to apply this yet. As such, I chose the solution that gives the most consistent and useful output to the user.