Computer Organisation and Program Execution
Assignment 2 Part 2 Design Document

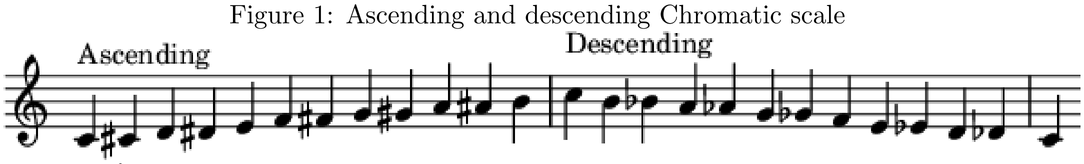
Kathia Anyosa - u6414938

15 May 2020

Overview

For part two of the assignment, I decided to modify the code in part one so as to generate a looping Chromatic scale as shown in Figure 1 (Benward and Saker, 2003). I chose this as I realized that it would involve a similar approach as that taken in part one. The approach involved generating a waveform for each of the 12 notes in the scale, and having them last for the same amount of time, with silence in between. The waveforms all have an amplitude of approximately `fixme`, so as to maintain roughly the same volume for all notes. The frequencies of each waveform were calculated from an initial note A at 440Hz and the scale played by the sequencer starts and ends on the note C.

Figure 1: Ascending and descending Chromatic scale



Implementation

The frequencies of each note were calculated from A at 440Hz, by following the same approach taken from *Chromatic Scale Frequency Chart and Definition*, in which notes below A are calculated by multiplying each one by 1.0594 and those above by dividing by the same number. This produced the following frequency table:

$Note$	$Pitch\ (Hz)$	*Ratio*
C	261.77	1.0594
$C\sharp/D\flat$	277.31	1.0594
D	293.79	1.0594
$D\sharp/E\flat$	311.24	1.0594
E	329.73	1.0594
F	349.31	1.0594
$F\sharp/G\flat$	370.06	1.0594
G	392.04	1.0594
$G\sharp/A\flat$	415.33	1.0594
A	440	1.0594
$A\sharp/B\flat$	466.14	1.0594
B	493.82	1.0594
C	523.16	1.0594

In order to calculate the period of each oscillation, the sample rate of 48kHz was divided the frequency and the result was rounded to the nearest integer, as per the following equation:

$$\frac{48kHz}{Frequency} \approx Period$$

From this the increase per step was calculated by dividing the range by the period and rounding to the nearest integer, I chose the following value as I wanted to achieve the loudest sound possible using the full dynamic range:

$$\frac{65535}{Period} \approx Increase\ per\ step$$

I wanted all notes to last for the same amount of time, so I set up counters that would stop the oscillation after 12000 steps, or 0.25 seconds. I also wanted there to be silence in between the notes, so that the change in pitch was more evident to listeners, and so I set a counter that would generate an oscillation with no amplitude for the same amount of time. However, I decided to double the length of each silence as I felt that it made the transition between notes more clear.

I initially wanted to follow the same approach taken in Assignment 1, where a function initializes the wave by setting the counter and the minimum value then branches to a loop to produce the waveform, but this proved to be more complicated as there were multiple waveforms to generate. Therefore, I decided to simply set those values without a label and branch to a loop that would store the address and return to it once the counters were zero, proceeding to generate the next note or silence. The waveform generators are labelled `pitch` and all work by continuously adding the increase per step and producing a sound while the counter is not zero. The silence generator `silence` works in the same way.

Reflection

After running the sample plotter I noticed that the waveforms did not have the range expected, which I believe is due to the minimum value being set only once and the loops continuously adding to this value, causing it to wrap-around and not reaching its actual minimum and maximum values. I also learnt how to properly `push` and `pop` the returning address. Initially I popped the address inside the `return` function, causing the sequencer to play only the first note. After this I moved it to the `pitch` loop, placing it before branching to `return`, which led to the same error. Finally, I placed it before both conditions and produced the whole sequence. Furthermore, I believe there must be a way to make the code more concise by having only one `pitch` loop that generates all waveforms, but I have not been able to figure out how to solve this as of the time of submission.

References

[1] Benward, B. and Saker, M., 2003. *Music: In Theory and Practice*. 7th ed. Boston: McGraw-Hill, p.37.

[2] Scaleandmodes.com. 2018. *Chromatic Scale Frequency Chart and Definition.*[online] Available at: `https://www.scalesandmodes.com/Major_Scales/Major_Scales/Chromatic_Scale_Definition.html`