Lab 5
Computer Organization
ECSE 324

Introduction

In this lab, you will combine the low-level ARM programming techniques you have acquired in the course by implementing a musical synthesizer (or "synth").

0 Audio

For this part, it is necessary to refer to section 4.1 (pp. 39-40) of the De 1-SoC Computer Manual.

Write a driver for the audio port following the approach in Lab4. The driver should only have one subroutine. The subroutine should take one integer argument and write it to bot the left and the right FIFO only if there is space in both FIFOs (Hint: Use the value of WSLC and WSRC in the subroutine). The subroutine should return an integer value 1 if the data was written to the FIFOs, ad return 0 otherwise.

Use the driver in an application that plays a 100 Hz square wave on the audio out port. The frequency can be achieved by knowing the sampling rate of of the audio DAC. For example, if the sampling rate is 100 samples per second and a 2 Hz square wave is to be played, that means there are two complete cycles of the wave contained in 100 samples, so for 25 samples a '1' should be written to the FIFOs, and for 25 samples a '0' should be written to the FIFOs.

For this lab, find the sampling rate from the manual and calculate the number of samples for each half cycle of the square wave. Finally, write 0x00FFFFFF and 0x00000000 to the FIFO instead of '1' and '0'.

1 Make Waves

We are providing you with drivers for writing to the audio codec, as well as a ".s" file with a wavetable containing one period of a 1 Hz sine wave at a sampling frequency of 48000 Hz.

To play a note with frequency f using the wavetable, generate a signal using the following equations for every sampling instant t:

$$\text{index} = (f * t) \bmod 48000,$$

$$\text{signal}[t] = \text{amplitude} * \text{table}[\text{index}].$$

If more than one note is played at the same time, compute the sample for each note and add them together. If the index is not an integer, you can calculate table[index] by linear interpolation using the two nearest samples. For example, table[10.73] := (1-0.73)*table[10] + 0.73*table[11].

You should write a function which takes as input f and t and returns signal[t]. Use a timer to feed the generated samples to the audio codec periodically.

2 Control Waves

It should be possible for the user to play the synth using a PS/2 keyboard. Table 1 shows the notes your synth should play, the key of the PS/2 keyboard each note should map to, and the frequency of the note in Hz. You should also implement a volume control with the keyboard so that the user can turn the volume (amplitude) up or down. This is the minimum functionality you must implement; you can also implement other notes and features, such as different voices.

Note	Key	Frequency
C	A	130.813 Hz
D	S	146.832 Hz
E	D	164.814 Hz
F	F	174.614 Hz
G	J	195.998 Hz
A	K	220.000 Hz
B	L	246.942 Hz
C	;	261.626 Hz

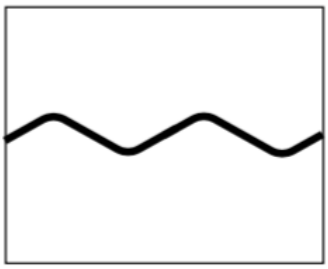

Table 1: Note Mapping

3 Grading

You will be evaluated according to the following criteria:

- Audio 20%
- Audio sound correct 20%
- Note control work 20%
- Volume control work 20%
- Does the code follow good software development principles (style, efficiency, clear commenting, appropriate use of source files, code reuse, etc.)? 20%