EE108a Lab 4: Sine waves

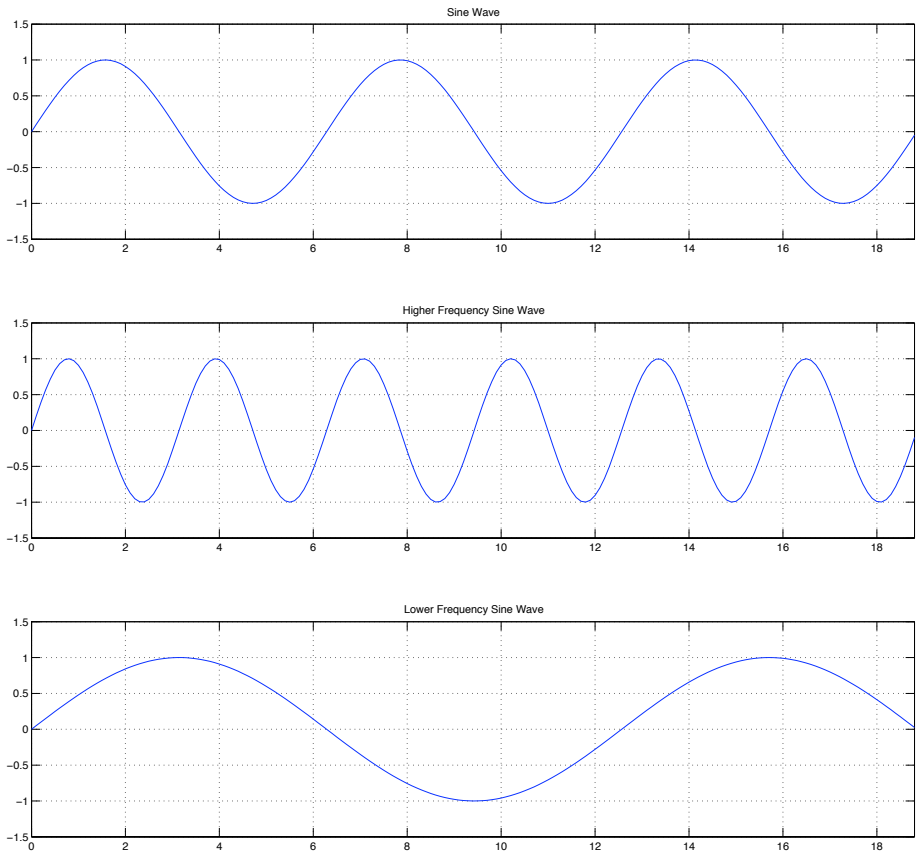
Version 1.0 – David Black-Schaffer

Introduction

In lab 4 you are responsible for generating music by synthesizing sine waves at different frequencies for the audio output codec. If you are not familiar with sine waves this document should give you a brief overview.

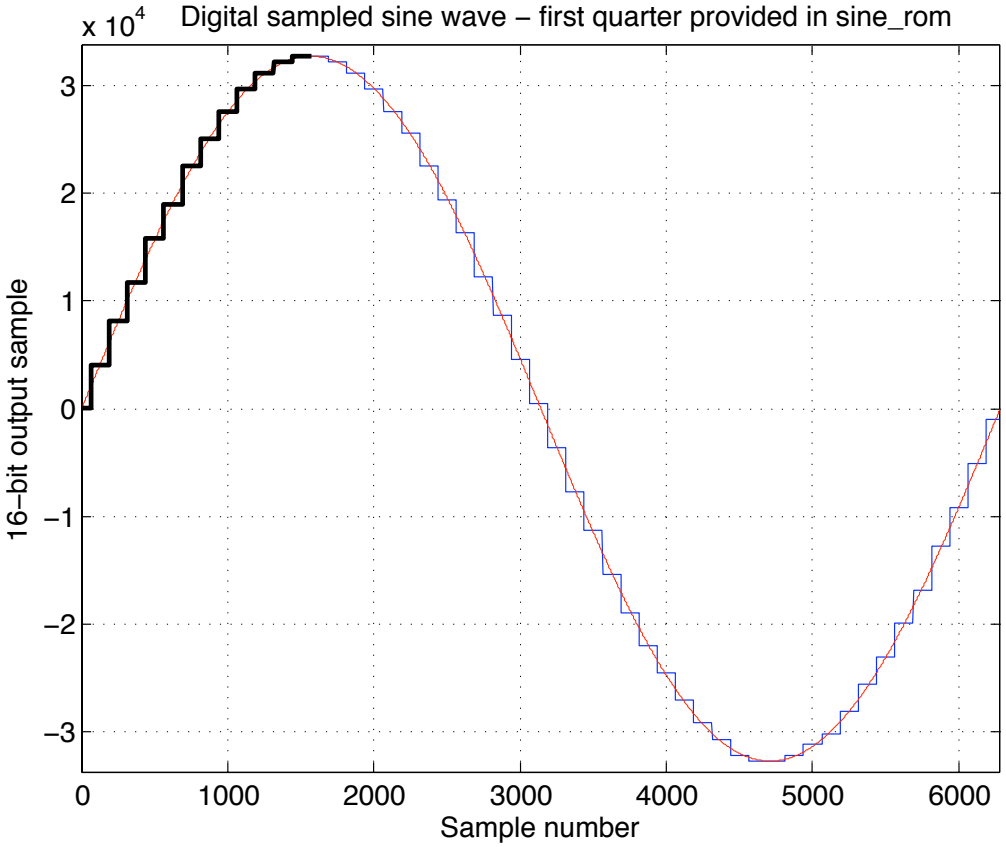
Generating Tones

The FPGA board produces sound by sending numbers (samples) to the audio codec at a fixed rate (the sampling frequency). The audio codec converts these numbers into a voltage, which then goes to the speaker and moves the physical speaker element back and forth with a magnet to follow the signal. If the numbers you send to the codec follow a sine wave, the output will be a pure tone. If the sine wave is slower the output will be lower frequency and if it is faster the output will be higher frequency.

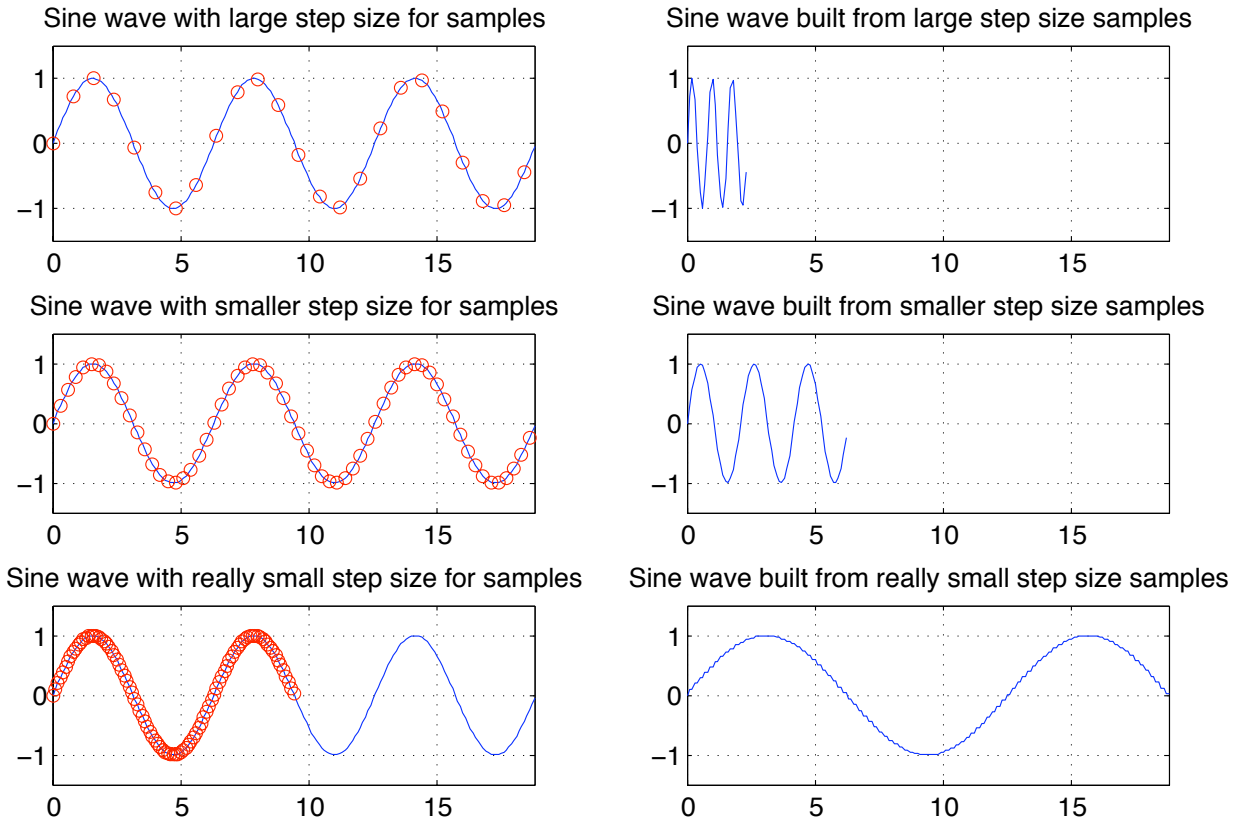


The audio codec on the board expects samples at the rate of 48,000 per second, or 48kHz. Figuring out what frequency you are generating is easy if you know this number. If it takes you 48,000 samples to get through a full sine wave, then you are producing 1 sine per second, and your frequency is 1Hz. (The limit of human hearing is around 20Hz, so you won't be able to hear anything.) Similarly, if you go through the full sine wave in 2 cycles then you are generating a frequency at 24kHz, which is the highest frequency you can generate with a 48kHz sampling rate. From this simple relationship you can figure out how many samples you have to go through the full sine wave to generate any frequency.

So how do you generate your notes for lab 4? Well, in lab 4 we are providing you with a read-only-memory (ROM) which contains the first quarter of a sine wave. The ROM has 10 address bits, so there are 2^10=1024 samples in the first quarter of the sine wave. (Which means the full sine wave would have 4096 samples.) To use the ROM you instantiate it in your design, put in the address of the sample you want (address 0 will give you the first sample of the sine wave, address 1023 will give you the last sample of the first quarter) and one cycle later you get the value for that position in the sine wave. When you generate tones for lab 4 you need to make sure you get the address right and flip the output value correctly to generate the full sine wave from the quarter we provide.



This is enough to get you to output a full sine wave, but if the sine wave we provided has 4096 samples in total, you will be producing 48,000/4096 = 11.7 sine waves per second, or a tone at 11.7Hz. Not very exciting. So how do you generate other notes? Well, the answer is simple: you skip samples. If we were to go through the sine_rom addresses with a step size of 2, we'd have a sine wave that only took 2048 samples, and our frequency would be 23.4Hz. Make sense? Here's a picture. Note that we have a sine wave on the left and we are walking through it with different step sizes. The samples we choose are then output at a fixed rate (48kHz) to produce the frequencies on the right.



Now the hard part is that the song_rom specifies a note, not a frequency. Somewhere we have to convert each note into a frequency and then into a step size for our particular sine wave. This is done for you in the provided excel spreadsheet, and it is exactly this calculation (and the knowledge of the frequency of middle C) that is used to generate the frequency_rom. So all you have to do is take the note and look up the step size in the frequency_rom. Then you walk through the sine_rom by this step size and output the samples. The only tricky part is to remember that you only have a quarter sine wave.