Musical Feet:
A Step-by-Step Approach to Music Generation

Harley Zhang and Rajeev Nayak

1 Introduction

The proposed digital system consists of several different modules that can be divided into three main groups: input modules, audio output modules, and video output modules. All of these modules will be clocked off a 27MHz clock signal provided by the 6.111 Labkit. The system's input comes from an analog signal produced by a pedometer, and it is then processed by the input modules to produce relevant data pertaining to the tempo and tonality of the generated music. These data signals are then sent to both the video and the audio output modules. The video output modules create a visualization pattern consisting of ripples that occur whenever a footstep is taken. The audio output modules use the data produced by the input modules to generate notes to be synthesized. These notes are produced from samples stored in BRAMs and then sent to the AC97 to be converted into analog signals, which can then be played by a speaker. The following sections outline the general function and implementation of the various modules. Figure 1, attached at the end of this proposal, is a high-level block diagram showing the interconnections between the modules.

2 Input Modules (Harley)

The input modules process the pedometer signal and use it to determine the tempo and tonality of the music to be generated. In this section, these modules are described in detail.

2.1 Pedometer and ADC

The input to the system comes from an analog pedometer PCB, which uses a magnet and coil to detect a user's steps. Normally, the pedometer's output is about 0V. When a step is taken, the magnet induces a voltage, producing about 250mV at the pedometer's output. This analog output is sent to an external ADC chip, where it is then converted into digital data (ped_dig_in). The signal is then sent to the tempo generator module.

2.2 Tempo Generator Module

The tempo generator module takes in the digitized pedometer input and uses it to generate a tempo for the music. A counter within the module counts each clock cycle so that each time a new footstep is taken, the tempo generator stores the number of clock cycles elapsed since the previous footstep and resets the counter. This number is stored in a circular array of ten registers.

Once the new number is stored in the array, the module computes the new tempo of the music. The tempo is determined by the time elapsed between each of the previous ten footsteps, as stored in the register array. The module takes a weighted average of these ten values and outputs it as the tempo_period. When the module is done calculating this weighted average, the tempo_ready signal is asserted.

2.3 Tonality Generator Module

The tonality generator module takes in the tempo_period input and uses it to generate a tonality for the music. Just like the tempo generator, the tonality generator employs an array of ten registers. In this case, the past ten tempo_period's are stored in the array. When the tempo_ready signal is enabled, the tonality generator stores the new tempo_period in the register array. It calculates the differences of successive tempo_period's in the registers and takes a weighted average of them. This approximates the rate of change of the tempo, which is compared to a threshold value in order to determine the tonality. If the tempo is changing significantly, the tonality is set to be minor, evoking instability. If the rate of change is below the threshold, the tonality is set to be major. The tonality is indicated by a 1-bit output, and the tempo_period is outputted as well. When these outputs are ready, the tonality_ready signal is enabled.

3 Audio Output Modules (Rajeev)

The audio output modules improvise and generate string quartet music to be played on an external speaker. These modules are described in detail below.

3.1 Chord Generator

The chord generator uses the tonality information to improvise the next chord for the string quartet to play. It internally stores the current key of the music and the previous three chords. Based on the tonality generated by the input modules, the key, the previous chords, and a random number generator, the next chord is selected. If the tonality changes, the chord generator will modulate to a different key, changing its internal key variable and computing a new chord accordingly. When the module is done calculating, the chord and tempo_period are sent to the next module, and the chord_ready signal is enabled.

3.2 Note Generator

The note generator reads the current chord when the chord_ready signal is high. Based on the chord, it generates notes for four instruments: two violins, a viola, and a cello. It also stores the previously played notes and takes them into account when it calculates the next notes. The notes are recalculated at every beat, which is determined by the tempo_period. If the chord remains the same during successive beats the note generator still assigns new notes to the instruments, keeping the same chord. When the chord changes, the note generator figures out a clean way to alter the notes from the current chord into the new one. Once computed, the four note values are sent to four different oscillators, one for each instrument.

3.3 Oscillators

Each of the four instruments has an oscillator that generates a waveform for a particular note. Based on the input note value, the oscillator accesses the instrument samples in a BRAM and specifies the correct samples in order to generate a pitch at the given frequency. These samples are read later in order to produce the final audio output.

3.4 BRAMs

The four oscillators access the instrument sample data in each of four respective BRAMs. These BRAMs will be initialized with samples of each of the four instruments beforehand, using a USB flash drive. The output of the BRAMs will be accessed by the envelope generators.

3.5 Envelope Generators

As the oscillators specify BRAM addresses, the envelope generators read the output data and process them using an amplitude envelope. Once the oscillators finish specifying all of the addresses for a single pitch, they send ready signals to each of the envelope generators. At this point, the envelope generators take the samples and pass them through an ADSR (attack, decay, sustain, release) envelope. The amplitude of the pitch is increased drastically (the attack), then decreased slightly (the decay), and then held constant (the sustain). When the oscillators indicate that new pitches are ready, the envelope generators reduce the amplitude of the current pitch to zero (the release) and immediately attack the next pitch. The volume is determined by the tempo of the music; as it gets faster, the music gets louder. The post-processed data from each envelope generator is sent along to the mixer.

3.6 Mixer

The mixer takes the data from each of the four envelope generators and combines them into one stereo signal. Once all four pitches are ready, the mixer adds their output signals together. This data is sent to the AC97, for digital-to-analog conversion. Once converted, the AC97 output will be played on a set of external speakers.

4 Video Output Modules (Harley)

The video output will be controlled by the tempo_ready, tempo_period, chord, and tonality signals. There will be two different modes of video output, each of which is summarized in this section. The current output mode will be selected by a switch on the Labkit.

4.1 Ripple Visualization

The ripple visualization depends on the tempo_ready and tonality signals. When the tempo_ready signal goes high, it indicates that a new footstep has been detected by the pedometer. This footstep will be represented on the monitor as a ripple emanating from the middle of the screen. At each new footstep, a new ripple will emanate outwards. This gives the user a sense of the tempo of the output music. The background color of the pattern will depend on the tonality of the music. When the music is in a major key, the background will be blue, and when it is in a minor key, it will be red.

4.2 Music Information

The music information mode displays the current chord and tempo of the music as text on the screen.

5 Testing and Debugging

Each of the separate module groups will be tested independently. The input ADC will be tested by displaying the ped_dig_in signal on the 6.111 Labkit's hex display and repeatedly activating the pedometer. The tempo generator, tonality generator, chord generator, and note generator will be tested by tying their inputs to switches on the Labkit and displaying their outputs on the hex display. The oscillators and BRAMs will be tested individually, using switches to specify notes and sending the BRAM outputs directly to the AC97 output. The envelope generators and the mixer can be tested in this same way, by incrementally inserting them between the output and the AC97 output. Finally, the VGA module can be tested by specifying its inputs with switches and viewing the output on the monitor.

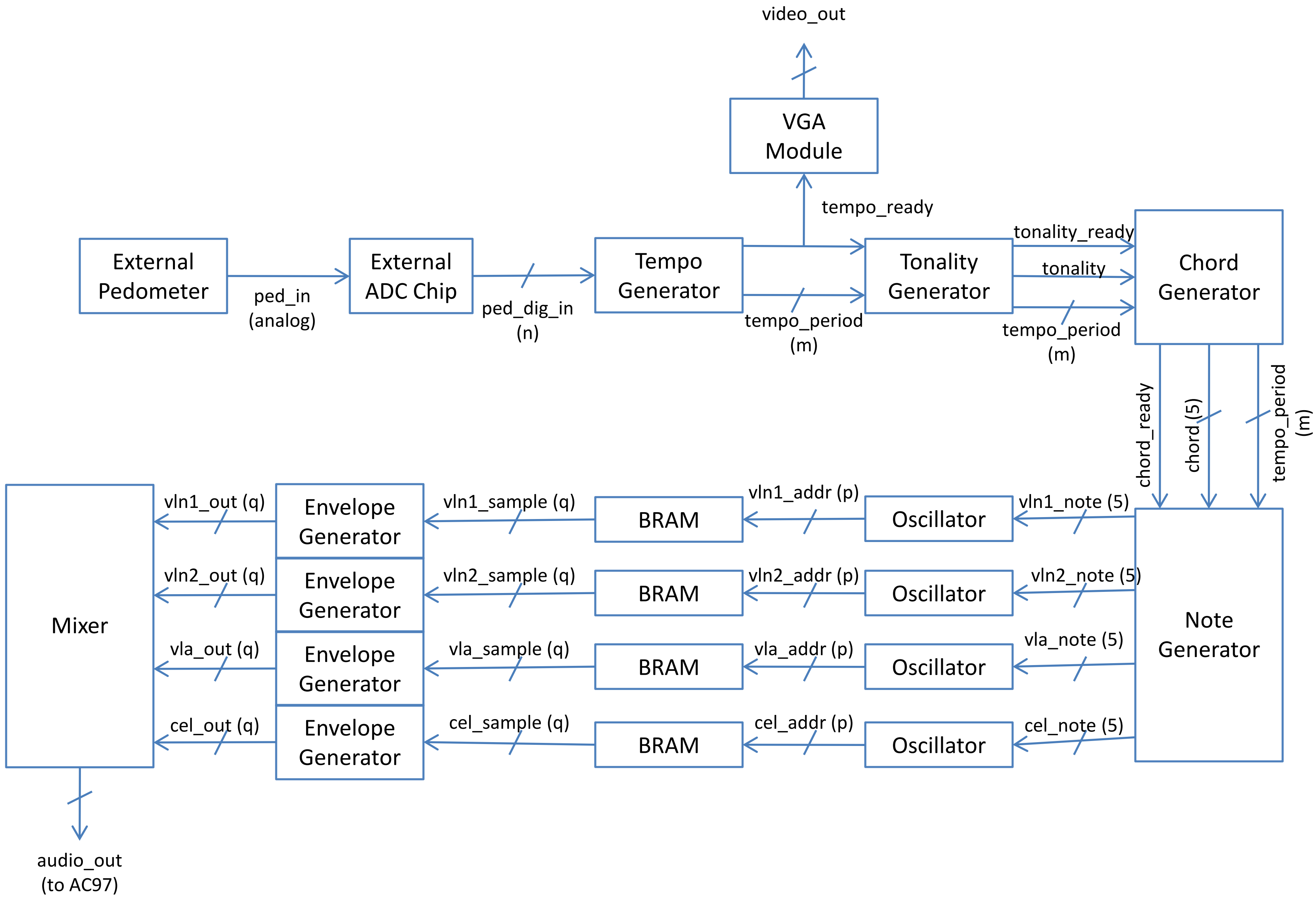

Figure 1. High-level block diagram of the Musical Feet digital system.