University of California at Berkeley
College of Engineering
Department of Electrical Engineering and Computer Sciences
Computer Science Division

CS 150
Spring 2002

J. Wawrzynek
Project Info.

Final Project Specification
MIDI Sound Synthesizer
Version 0.5

1 Introduction

For the final project you are required to use an FPGA board to build a "box" that takes a MIDI signal as input and generates a audio waveform as output. Figure 1 shows a high level view of the synthesizer that you will build. MIDI is an acronym for Musical Instrument Digital Interface, and a MIDI signal is a bit-serial stream of bytes. The audio waveform is a mono (as opposed to stereo) signal. It will be strong enough to drive headphones or small speakers. The MIDI synthesizer is monophonic, meaning that it will translate MIDI signals into sound not more than one note at a time, and single channel, meaning that it will produce the voice of only one instrument. The audio waveforms are generated using a technique called *waveform synthesis*; waveforms from actual musical instruments are stored in ROM and used to generate sound in response to MIDI commands. The sound waveforms are stored and played-back using 16-bit data samples. Output sampling rate is 31.25KHz.

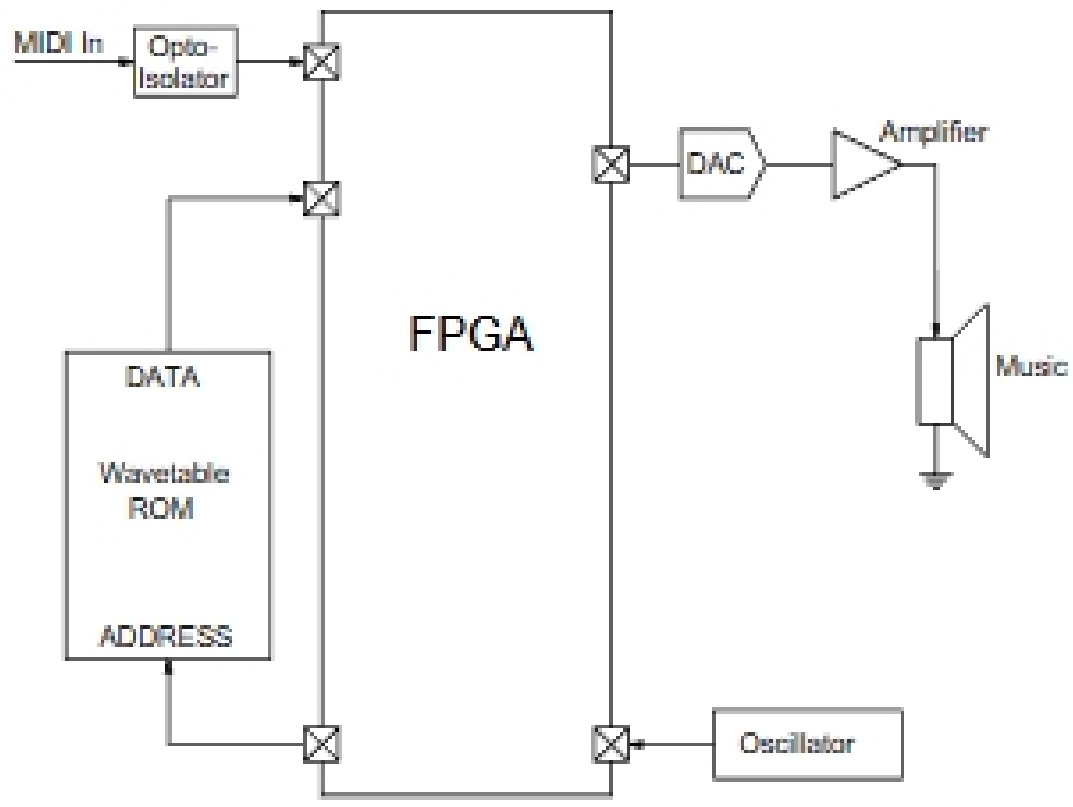

Figure 1: High-level view of the MIDI Synthesizer.

Our project is a simplified version of commercial synthesizers. For comparison, a medium

grade commercial sound box has stereo output, is polyphonic, and is capable of generating 30 notes simultaneously. Note pitch can be varied continuously, and the voices include numerous digital filters for adding special effects to the sound output. Such systems cost around several hundred dollars.

1.1 Sound and Music Theory

Sound is air vibrating at an audible frequency, typically 20Hz – 20KHz for an adult human. The amount of displacement can be sampled and recorded as a sequence of magnitudes over time, and reproduced by speakers or headphones.

Our hearing is quite complex in the way we perceive musical tones. Two of the most important characteristics of a musical tone is its loudness and its pitch. To a first approximation human hearing is logarithmic in perceiving both loudness and pitch. In the case of loudness, this means that a we perceive the loudness as being proportional to the logarithm of the sound wave amplitude.

In the case of pitch, the human auditory system is very keen at detecting the logarithmic relationships between frequency, and "musical" pitch intervals. The simplest interval to detect is the octave, where the higher pitch has double the frequency of the lower pitch. Given those two frequencies, it is possible to subdivide the interval (which is the multiple 2) into twelve parts, producing eleven *semi-tones* in between. These twelve semi-tones form the chromatic scale of the traditional western *twelve-tone scale*.

These twelve intervals form a geometric series. As we move from one up to the next we can determine its frequency by multiplying by $\sqrt[12]{2}$. After 12 such multiplications we will have doubled the frequency and reached the octave. Most people can detect a pitch difference much smaller than these 12 divisions, in fact as small as a few hundredths of a semi-tone. In musical terms a *cent* is 1/100 of a *semi-tone*. To increase the pitch of a note by one cent, multiply its frequency by $\sqrt[1200]{2}$.

The note called middle C has a frequency of 261.63Hz. The MIDI encoding for that note is key 60. The note called high C, which is an octave higher, has a double the frequency—523.25Hz, and it has a MIDI encoding of 72. Others tones can be produced by multiplying and dividing the frequency by factors of $\sqrt[12]{2}$. For example, MIDI note number 61 has a frequency of 261.63kHz$\times \sqrt[12]{2}$.

There is one more important aspect to a musical sound, called *timbre*. Whereas pitch is indication of how often a sound wave repeats, and loudness is the way we perceive the amplitude of a sound, timbre is a perception of the *shape* of the sound wave. Sometimes the timbre of a musical tone is called its tone *quality* or *color*. Different musical instruments have different, and readily identifiable, timbres. Like our perception of other characteristics of a musical tone, timbre is complex, but it is deeply related to the way a musical note starts, the shape the waveform takes as it repeats itself, and how it dies out.

For many musical instruments, a simple model can be used to describe the shape of the waveform. The waveform can be split into three parts: an attack, a sustain, and a release. The attack is the most interesting part of a musical sound, corresponding to when a note is first played. It typically lasts for no more than 300ms, and during this time, the waveform may be very irregular. Most of the rest of the musical sound is very repetitive,

and it is called the sustain. In the sustain section, not much would be missing to the ear if just one cycle were played over and over again for the duration of the rest of the note. At the end of the note, there may also be irregularities in the waveform, and the release consists of the dying away of the sound. The release is typically shorter than 100ms.

This simple model matches some musical instrument better than others. Notes from flutes and other woodwind instruments and bowed sounds such as from the violin family, fit nicely into this model. However, sounds from instruments that are struck, such as a marimba or even a piano don't really have a sustain part to their notes. In these cases we could simply omit the sustain part of the model, or probably more simply represent the entire note as an attack only.

1.2 Sampling and Storing

A 512KB EPROM will be used to store the sound information. At a sample rate of 31.25KHz, and at 2 bytes per sample, the EPROM can store roughly 8 seconds of raw sound. However, it must be capable of reproducing notes of maybe 100 different pitches at varying durations.

The simplest way of getting arbitrary duration is to store an attack, a short section of the sustain, and a release in a *note template*. Then, to generate a note, the synthesizer will play the attack and continue playing into the sustain. While the note is held, it will continue to play the sustain part by *looping*. When the key is released, to signal the end of the note, the synthesizer will continue playing the sustain to the end of the current loop iteration and then play the release portion of the stored note. This process for playing a note is illustrated in figure 2.

One way of generating an arbitrary pitch is to store just one note template of an instrument at a known pitch, and varying the playback frequency. Early inexpensive synthesizers varied the play back frequency of a note by adjusting the sampling frequency. This creates problems in the system design and will not be used here. Instead we will output all notes with a constant sampling frequency and will vary the *stride* with which we step through the samples in the template. For example, if we stored the note middle C (MIDI key 60) sampled at a rate of 31.25KHz, playing the note back by taking every sample from the template would result in another middle C. However, if we take every other sample from the template (stride = 2), the frequency, and thus the pitch, of the note would double, resulting in high C (MIDI key 72). Generating notes with a frequency in between these two requires the use of non-integer stride and thus a non-integer index into the template. In our design, we will ignore the fractional part of the index when looking up samples stored in the template. However, the index must be maintained as a non-integer number internal to the synthesizer logic. Truncating the index when looking up samples in the template, results in a slight distortion of the waveform, but greatly simplifies the logic. You may choose to earn extra credit by modifying your design to account for the fractional part of the index. As described later in this document, linear interpolation performed on the closest two samples stored in the template results in a more accurate waveform.

An interesting property of musical instruments is that the timbre of the sound varies from note to note over the range of the instrument. Notes played in the high register of a