Modular Audio Effects Box (aka modFXbox)

Andrew B. Shapiro, Marc P. Resnick

November 2, 2010

Abstract

This reports presents the design and implementation for a Field Programmable Digital Effects Rack using the Xilinx 2 series FPGA. Using an 18-bit AC'97-compatible codec (audio) and a 24-bit VGA output (video), the FPGA can be used to synthesize the large, even cumbersome devices normally used by musicians and sound engineers to modify sound in real-time. Industry standard effects such as delay, pitch modulation, convolution, filters, and others will be available to the user. Modular effects blocks and configurable dataflow allow for an extensible library and numerous arrangments of effects. The user interface for constructing an ordering through the effects will be displayed on a monitor and interactively configurable via a dummy patchbay or keyboard and mouse. The final result is a modular system for emulation of audio effect devices many times the size of an FPGA, which is compatible and interchangeable with the actual devices that it emulates.

1 Overview

The Modular Digital Effects Box is a medium and interface for the routing of an audio signal through an array of digital audio effects. The user controls the parameters of each modularized effect block using an onscreen interface and arranges the routing of the signal through the blocks using a physical patch bay. The GUI will update to reflect the current signal path. The end result is a reconfigurable array of audio effects that can be used to modify sound in real time.

The labkit's onboard AC'97-compatible codec reads 18-bit samples of audio input at a sampling rate of 44.1kHz. This input wire will have a designated port on the patch bay. The samples are then passed into the effects block. This block contains each instantiated effect module, along with their inputs, outputs, and parameters. In order to configure the routing of the effects block, the patch bay is scanned for the connections that have been made. To accomplished this, the control module instructs the patch bay's synthesized outputs to sequentially output constant a constant test signal while each input is probed for that signal to determine the connection. The labkit's general I/O connections will be used to wire and probe the patch bay. After the routing has been determined, the input of each module is set either to the appropriate output, if it is connected, or ground, if it is not. The control block also passes the parameters of each effect module, which are determined by the interface.

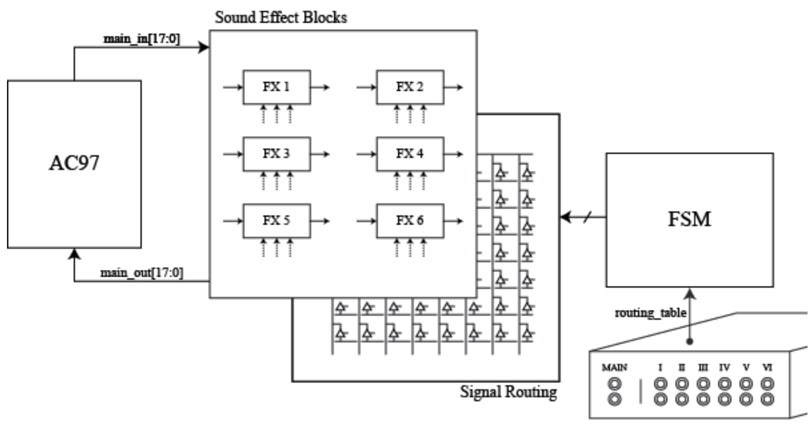

Figure 1: System Block Diagram

The effects block is modular and extensible, made up of many submodules, each which modifies an audio signal in some way. Also included is a signal generator, useful for both testing and creating new effects by composition of modules. Effect prmitives include a delay, pitch shifter, and equalizer; other modules can be built from these primitices and signalgenerators, such as vibrato and chorus. A list of desired blocks will be used to instantiate the necessary modules before compilation, and new modules can be easily integrated into the system. When the output module receives the digital signal, it passes samples back to the AC'97, which converts them to analog and outputs audio.

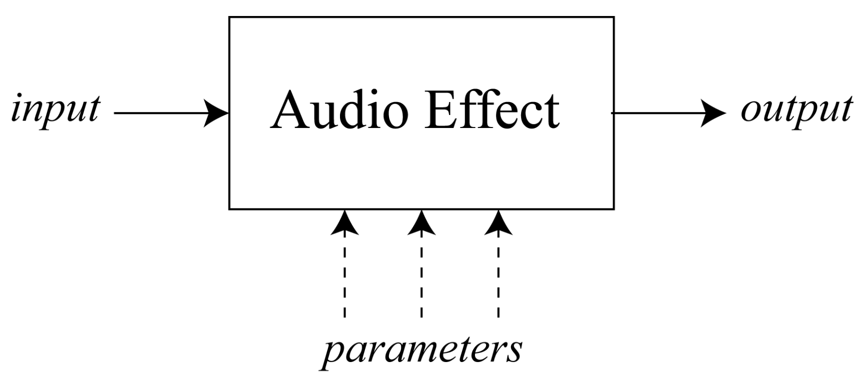

Figure 2: Effect Primitive. System function from input signal to output signal is dependent upon the effect type and paramter values.

2 Audio Effects Modules

2.1 Filter Effects (A. Shapiro)

Description A filter effect block has the ability to apply Finite Input Response (FIR) filter using convolution methods developed in Lab 4A (recorder). Each filter is represented in ZBT memory as a 64-sample 10-bit resolution FIR.

Inputs Clock, Original Signal (18-bit), FIR Address

Outputs Filtered Signal (18-bit)

Memory Required (number of filters) * 64-samples * 10-bits

Testing Signal generator will be used in combination with oscilloscope to visually inspect the output of this module for intended time- and frequency-domain effects of a given filter. For example, high- and low-pass filters can be verified by analyzing the FFT spectrum of input and output signals.

2.2 Volume and Stereo Effects

2.2.1 Mixer (M. Resnick)

Description In audio applications, the "mix" of two signals describes their weighted sum. This module will be instantiated by other effects blocks to control how "wet" or "dry" the output is. A "wet" output implies that only the modified signal is passed as output, whereas a "dry" output includes only the original signal.

Inputs Clock, Original Signal (18-bit), Modified Signal (18-bit), Mix (signed 5-bit)

Outputs Weighted sum of input signals (18-bit)

Testing Signal generator will be used in combination with oscilloscope to mix two signals and compare with a known desired output.

2.2.2 Pan (A. Shapiro)

Description Panning, or distributing signal strength between left and right channels, is a simple way of using stereo audio to create the impression of moving sound. A weighted average which preserves total power of the output signal (i.e. average of left and right channel sample is equal to mono sample) distributes the input signal between the left and right channels. The degree to which the signal is panned is determined by the Pan paramter.

Inputs Input Signal (18-bit) , Pan (signed 5-bit)

Outputs Panned Signal (18-bit)

Applications Ping-pong

Testing Signal generator will be used in combination with oscilloscope to pan stereo from full left channel to full right channel in 32 steps.

2.3 Delay Effects (A. Shapiro)

Description Delay will intentionally introduce latency into the data path of the audio signal. Many parameters control the behavior of the delay, and playback can even be looped over the length of the delay buffer.

Inputs Input Signal (18-bit), Delay Length, Playback Speed, Buffer Length, Enable Loop

Outputs Delayed Signal

Applications Echo, Chorus, Reverb

2.4 Pitch Shifter (M. Resnick)

Description Pitch shifting is the process of raising or lowering the perceived fundamental frequency of a sound without altering its speed. The FFT of an input signal is analyzed to

This is accomplished by first taking the FFT of window of samples, and frequency domain analysis to find the magnitude and true frequency. The frequency of the sample is then scaled by a user-input amount, after which is is synthesized and converted back to the time domain with an IFFT.

Inputs Signal (18-bit), Scale (signed 5-bit)

Outputs Signal (18-bit)

Testing Signal generator will be used to create a sine wave of a known frequency, and the output will be observed with the oscilloscope to see if the frequency shifts properly in the desired range and maintains its shape otherwise. Testing on actual audio input will also be done to verify that the output does not include any unexpected artifacts.

2.5 Composition of Modules

Additional effects blocks will be created by the composition of instances of the above modules. Such blocks include a flanger, chorus, slapback, echo, and numerous other effects.

3 Supporting Modules

3.1 Signal Generators (M. Resnick)

Description In order to create sound when no live audio source exists, the Modular Digital Effects Box uses look-up tables to produce periodic signals of varying shape. These signals are also used as paramter inputs to other effects (e.g. time varying delay will produce a reverb effect). Sine, Square, Sawtooth, and Triangle waves will be stored in labkit memory, leaving more room for volatile memories on the FPGA.

Inputs Clock, Frequency (15-bit), Amplitude (18-bit), Shape (2-bit selector)

Outputs Generated-Signal (18-bits, updating at *Frequency*)

Memory Required 4 1024-sample, 18-bit full-wave LUTs (Square, Sawtooth, Triangle, Sine), totaling about 74KB

Testing Output signal will be observed using oscilloscope and compared to expected waveform

3.2 User Interface (A. Shaprio and M. Resnick)

Description Users interact with the Modular Effects box via the mouse and keyboard (as well as the patchbay). The visual display of system information including enabled effects, routing, and block paramater constants is controlled by the UI module. Managment of screen real-estate is facilitated by a click-and-drag workflow. Interactions that directly effect audio effect settings are passed to the Control module using a protocol that frames the relevant information in a User-issued Command word.

Inputs System Clock, Pixel Clock, Mouse, Keyboard

Outputs VGA data signals, User-issued Command word

Memory Required Double-buffering may be used to reduce flicker.

Testing Functionality of the display and user-inputs will be testing incrementally as they develop.

3.3 Control Logic (A. Shaprio and M. Resnick)

Description The control logic will be responsible for scanning the patchbay, updating the routing table, and handling user commands issued from UI module. The information stored in the routing table is used to configure the Signal Routing Mesh.

Inputs Patchbay connectivity, User-issued Command word

Memory Required Routing Table (unknown size)

Testing Functionality of the control module will be tested using scripted test fixtures.

3.4 Signal Routing (A. Shaprio)

Description The control logic will be responsible for scanning the patchbay, updating the routing table, and handling user commands issued from UI module.

Inputs Routing Table

Outputs Enable signals for matrix signal taps.

Testing Following the signal path through numerous routing configurations will reveal any problems or artifacts introduced by this module's implementation.

3.5 Convolution (M. Resnick)

Description This module is taken in part from Lab 4A of the 6.111 course material. Its function is to apply a convolution of of two input signals.

Inputs: Signal A (18-bits), Signal B (18-bits)

Outputs: Convolved Signal (18-bits)

Memory Required: pipeling of multiply-add operations will require additional registers.

3.6 Fast Fourier Transform

Description The Fast Fourier Transform is an algorithm that quickly determines the frequency domain representation of sequence of samples. This representation of the sound can then be modified and resynthesized into the time domain, allowing for any effects that rely on pitch modulation to function.

Inputs: n-sample window of 18-bit samples

Outputs: Frequency domain representation of samples