Digital Supersaw Synthesizer

Jacob Brown, Isabelle Liu
6.111 Fall 2018

Introduction

In decades past, electronic music was generated in the analog realm. Audio synthesizers were expensive, heavy, and prone to the same problems as in other integrated circuits. With the increase in transistor density and refinement of the processor, more and more complicated tasks are being performed in the digital realm. There are many software programs meant to perform the same functions as their analog predecessors, but few all-digital synthesizers have been created that satisfy the electronic music creators. Indeed, musicians are hard to please.

We propose to design and build a multi-voice, configurable digital synthesizer, allowing a user to play notes/melodies on a USB MIDI keyboard, adjust for desired sound features on a separate control panel, and hear the synthesized 96 kHz, 16-bit stereo result through a standard audio interface. In addition, we plan to display the output waveforms (left and right channels) in real-time over 12-bit VGA to provide visual feedback. We plan to base the overall design on the commercially-available JP6K VST synthesizer plugin, which emulates the analog "Supersaw" waveform known in the EDM industry as the de-facto standard for trance leads. The Supersaw algorithm originated on the Roland JP8000 analog modeling synthesizer and consists of seven detuned and phase-shifted sawtooth waves, where the inter-set mix level is adjustable and the intra-set phase-shift amounts are random [1].The number of voices (simultaneous notes) will be set to 8 as in the original; for two channels, a total of up to 112 concurrent free-running oscillators will be active.

Functions

The project can be broken down into individual modules, to be implemented and tested individually and given priority according to the Goals section. A description of the basic modules, along with an explanation of their means of communicating to other modules, follows.

Deserializer: Converts raw serial data (sequences of 3-byte packets) from the Nexys4 PMOD port into (pitch, velocity, trigger) data to be sent to the **Octave Select** module (or **Arpeggiator** module, if present).

Arpeggiator: Staggers all active notes to produce an arpeggio. Takes input from the **Deserializer** module and **ADC Host** and outputs raw note data (pitch, velocity) to the **Octave Select** module according to an internal sequencing algorithm.

Octave Select: Scales the pitch information up or down by one or two octaves (multiples of 12 semitones). Takes input from the **Deserializer** module (or the **Arpeggiator**, if present) and **ADC Host** and outputs raw note data (pitch, velocity) to the **State Machine** module.

State Machine: Keeps track of how many notes are active based on incoming (pitch, velocity) data and feedback from an ADSR envelope generator. Sends (pitch, velocity) and enable signals to the **Supersaw** modules; enable signals to the **Envelope** module.

Supersaw: Produces a set of seven detuned and phase-shifted sawtooth waves corresponding to a single fundamental frequency. Takes (pitch, velocity) and enable inputs from the **State Machine** module and parameter inputs from the **ADC Host**. Converts MIDI pitch values into periods, computes the necessary detuned periods, and instantiates seven sawtooth **Oscillator** modules with (period, enable) inputs. Mixes the outputs of each **Oscillator** module in relation to the base frequency. Scales the amplitude according to velocity data and parameter input, and finally applies a high-pass filter at the base frequency with an FIR/IIR algorithm, which requires a 127-column memory lookup. Outputs a new 16-bit sample on each sample clock to one of two **Amplifier** modules.

Oscillator: Generates a single 16-bit sawtooth wave. Takes (period, enable) inputs from a **Supersaw** module, computes the slope, and updates the 16-bit sample data every 1,000 clock cycles. Outputs a valid 16-bit sample on every clock cycle.

Noise Generator: Generates white noise and sends it to the **Amplifier** modules, where it is mixed into the output of both **Supersaw** modules. Takes parameter input from the **ADC Host**.

Partial Detune: Detunes the two **Supersaw** modules by up to 10% with respect to each other. Takes parameter input from the **ADC Host** and passes a multiplier value to each **Supersaw** module.

Envelope: Tracks the rise and fall of each note and modifies it according to an ADSR curve. Generates an amplitude envelope, applied at the trigger event of each note. Takes a set of enable signal inputs from the **State Machine** module and ADSR (Attack, Decay, Sustain, Release) inputs from the **ADC Host**. Outputs 1-8 disable signals to the **State Machine** module, along with 1-8 multiplier values to both **Amplifier** modules, corresponding to the output of each **Supersaw** module.

LFO: Produces an amplitude envelope that varies in time and which is applied to all oscillators simultaneously. Takes input from the **ADC Host** and, preferably using the Nexys4's IP cores, generates multiple envelope waveforms including sine, triangle, square, and sawtooth types; these waves oscillate from 0-20 Hz. Outputs a single multiplier value to both **Amplifier** modules.

Amplifier: Mixes the output of up to eight **Supersaw** modules into a single waveform. Takes input from the **Supersaw** modules, parameter inputs from the **ADC Host**, white noise from the **Noise Generator** module (if present), and scaling information from the **Envelope** and **LFO** modules. Outputs a new 16-bit sample on each sample clock to one of two **Filter** modules.

Filter: Applies an FIR/IIR filter (either high-pass or low-pass) to the signal. The filter order is fixed, but inputs from the **ADC Host** determine the cutoff frequency and resonance of the filter. Samples arrive from one of two **Amplifier** modules and are sent to one of two **Equalizer** modules. The same filter coefficients are used in both instances of this module. Requires a memory lookup from up to 127 * 127 columns of coefficients.

Equalizer: Applies two FIR/IIR shelving filters in series to the output of one of two **Filter** modules. Takes sample input from the **Filter** module and parameter input from the **ADC Host** to approximate bass response and treble response. The coefficients are also shared between both instances of this module. Requires a memory lookup of up to 127 + 127 coefficients. Outputs sample data to the **Offset** and **Display** modules (or **Delay** module, if present).

Delay: Produces an echo effect by delaying the mixed, filtered, and equalized oscillator outputs. Takes sample input from an instance of the **Equalizer** module along with parameter inputs from the **ADC Host** and outputs sample data to the **Offset** and **Display** modules (or **Reverb** module, if present). Stores up to two minutes (~44MiB for two instances) of the incoming sample data to memory and recalls, attenuates, and mixes together the original waveform with its delayed components.

Reverb: Produces a reverberation effect by applying an FIR/IIR model to the mixed, filtered, and equalized oscillator outputs. Takes sample input from an instance of the **Equalizer** module (or **Delay** module, if present) along with parameter inputs from the **ADC Host** and outputs sample data to the **Offset** and **Display** modules. Stores up to 30 seconds (~11MiB for two instances) of the incoming sample data to memory to be convolved by up to 127 * 127 different impulse response coefficients, corresponding to the varying Size and Decay parameters. The Attack parameter determines how many samples to delay the incoming data by and introduces another memory requirement of up to 375KiB for one second of predelay. Finally, the Level parameter sets the relative mix level of processed to unprocessed sound.

Display: Produces video output to be displayed on an attached VGA monitor running at 1024x768 pixels x 60 Hz refresh rate. Takes input from all **ADC Host** channels and sample data to draw in real-time the waveforms corresponding to both channels, as well as to provide visual feedback in the style of the Roland JP-8000 analog modeling synthesizer to show the current state of the parameters. The top half of the screen will show the left and right supersaw outputs; the bottom half will the be the control panel of the synthesizer, as shown below in Figure 1. Additional features may include sections of the screen to display the total number of active oscillators, memory usage, and/or pitch/frequency values.

Offset: Removes the DC component of the output waveforms to allow for proper DAC operation. Takes input from an instance of the **Equalizer** module (or **Delay/Reverb**, if present) and produces a level-shifted sample output centered at zero.

CS4344 DAC: Converts the digital signals from the two channels into analog signals at an appropriate voltage to drive a line-level stereo audio output jack.

ADC Host: Handles SPI communication with two or three **ADS7961 ADC**s through several digital data/clock lines brought out to the PMOD port. Takes input from each **ADS7961** (via the MISO/SDO line) and provides output to all parameter buses.

ADS7961 ADC: Converts the analog signals from the various passive control elements (potentiometers, switches, sliders, and buttons) into 8-bit digital values to be interpreted through the PMOD inputs. Requires external 2.5 V reference (e.g. REF192GPZ). This is an analog component and as such will be built on a breadboard to be attached to the physical control panel. The schematic is shown below as Figure 2 and Figure 3 in the Schematic section.

MIDI Communication

This section describes the data flow and conversion from MIDI keyboard input to the Nexys4 FPGA. A USB MIDI keyboard is connected to a Windows PC, which performs the necessary handshaking and converts the USB signals into serial using the open source software Hairless MIDI, which is output to a USB-to-Serial adapter as used in Lab 2b. This adapter connects to a Digilent Pmod RS232 adapter that converts serial data to Pmod inputs which is finally fed into the Nexys4 Pmod input ports. A possible alternative is using an embedded USB Host microcontroller such as the CY7C63310, which would require additional programming but would eliminate the PC entirely.

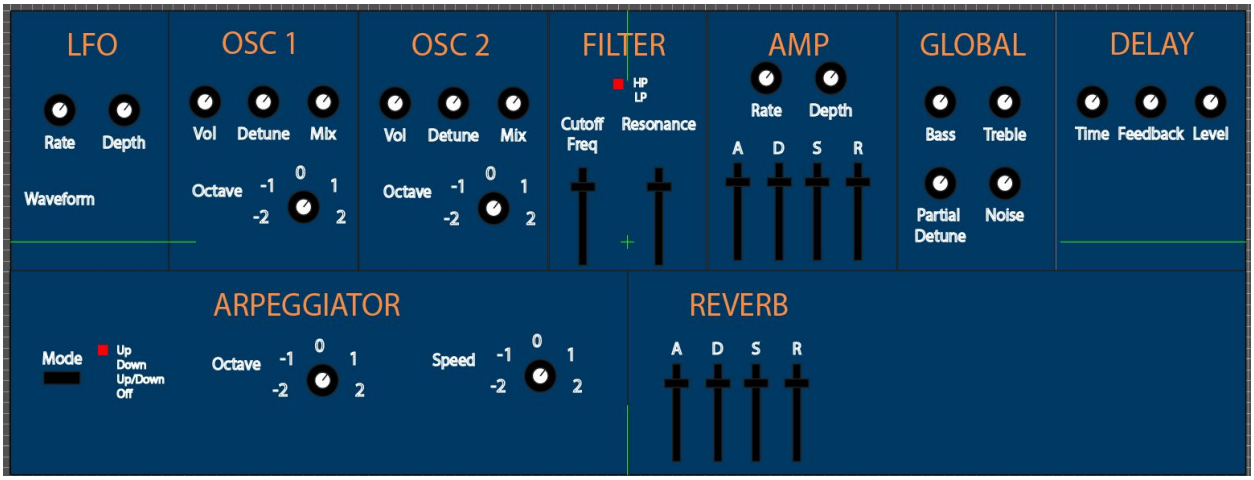


Figure 1 Rough layout of synthesizer control panel
that will be displayed on the bottom half of the screen

Goals

The project is divided into three achievement levels: commitment, goal, and stretch. The description of these levels are as follows.

Commitment

The commitment goal includes everything needed to hear the basic Supersaw waveform in stereo and monitor it with the display. The following modules apply to this level: **ADS7961 ADC**, **ADC Host**, **Deserializer**, **State Machine**, **Oscillator**, **Supersaw** (to be further subdivided), **Amplifier**, **Offset**, **CS4344 DAC**, and **Display**.

Primary

The primary goal includes the commitment modules plus the **Octave Select**, **Equalizer**, **Filter**, **Envelope**, and **LFO** modules. This represents what we believe to be the most useful addons to the core of the project.

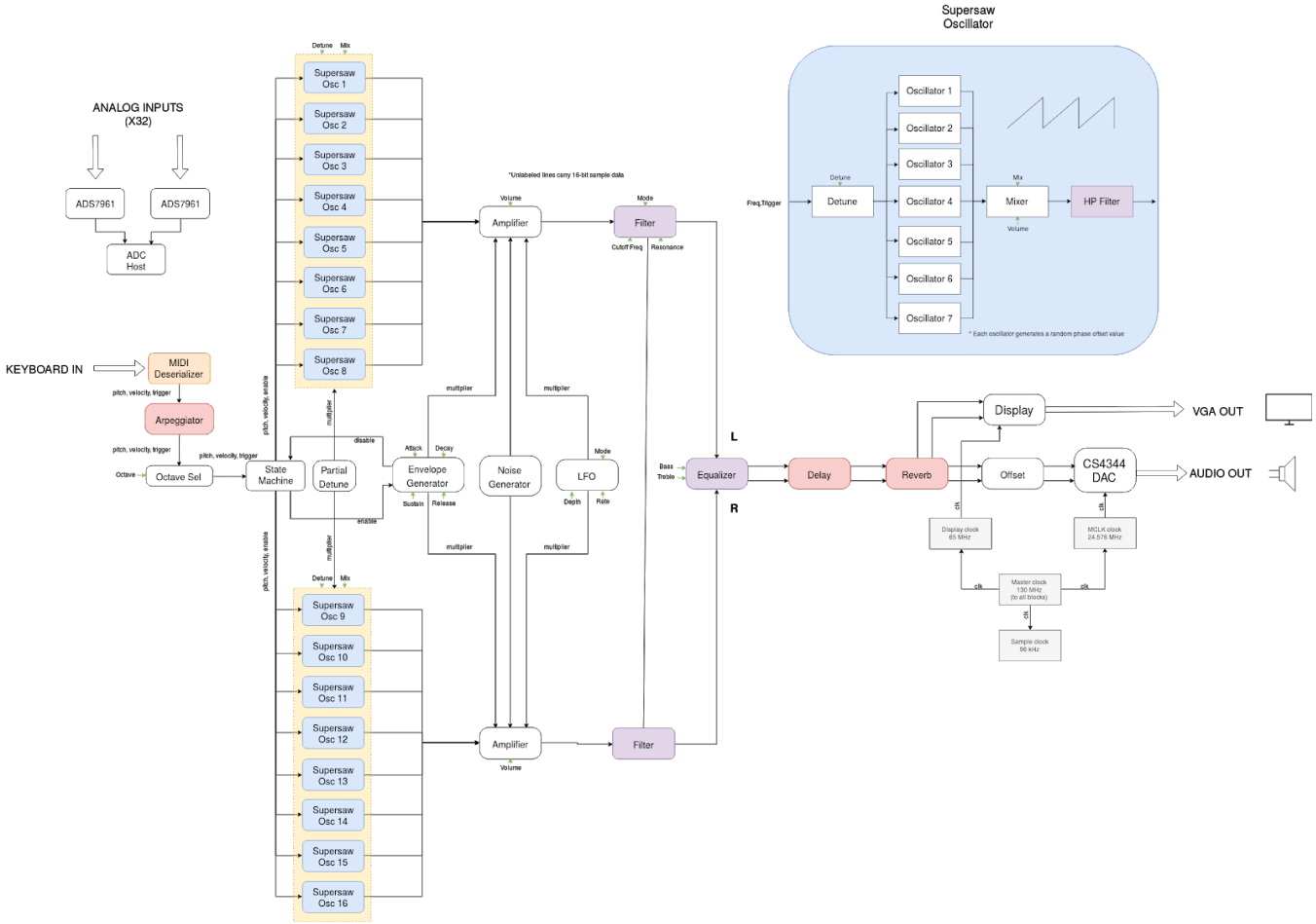
Stretch

The stretch goal includes all of the above plus the **Arpeggiator, Delay, Reverb, Noise Generator**, and **Partial Detune** modules. These are features that have the potential to create the richest, most pleasing musical palette but which have an uncertain requirement on time investment.

Responsibilities

We plan to subdivide the labor for the project according to our individual skill sets. Isabelle has experience with mechanical layout and drafting, serial data encoding/decoding, and graphics, so she will spearhead the construction of the analog input control panel, code the **Deserializer** module and **ADC Host** module, and the **Display** module. Jacob will apply his understanding of analog circuits and music theory to implement as many of the other modules as possible. We foresee sharing the coding burden as much as possible.

Block Diagram



Memory Requirements

We anticipate that we will be able to accomplish everything within the Nexys4's 128 MiB of DDR2 SRAM. The display portion would require a few dozen ROMs to represent various control elements, but we anticipate the optional **Delay** and **Reverb** modules to use the most, since recall is needed of up to a minute or more of previous sample data for these modules to function as in modern synthesizers. Digilent provides code to enable quick access to the SRAM (https://reference.digilentinc.com/learn/programmable-logic/tutorials/nexys-4-ddr-sram-to-ddr-component/start).

External Components

Item Name	Qty	Part No	Unit Price	Total price	Notes
MIDI Keyboard	1			(N/A)	Needs a USB Host
Pmod I2S Stereo Audio Output	1	Digilent 410-191	13.99	13.99	CS4344 DAC
8-bit 16-channel ADC IC	2	Digikey ADS7961SDBT	6.16	12.32	28-TSSOP (SMT)
Breakout board	2	Digikey PA0039-ND	6.89	13.78	28-TSSOP (SMT) to DIP
Sliders	6	Digikey PTA3043-2010CIB103-ND	1.32	7.92	More precise adjustment (10k)
Rotary Pots	20	Digikey 987-1661-ND	0.64	12.80	(10k) (purchased separately)
5-way Rotary Switch	2	Digikey CKN11043-ND	6.14	12.28	SP5T
4-way Rotary Switch	1	Digikey CKN11777-ND	4.56	4.56	SP4T
Digilent Pmod RS232	1	Digilent 410-068	14.99	14.99	Serial to Pmod interface (purchased separately)
Subtotal				$92.64	

Schematic

Shown below in Figure 2 is the ADC circuit that converts the numerous analog inputs that control our synthesizer's various features into digital signals. Figure 3 is an enlarged view of the left half of the ADC, showing the chosen component values and digital data lines in greater detail.

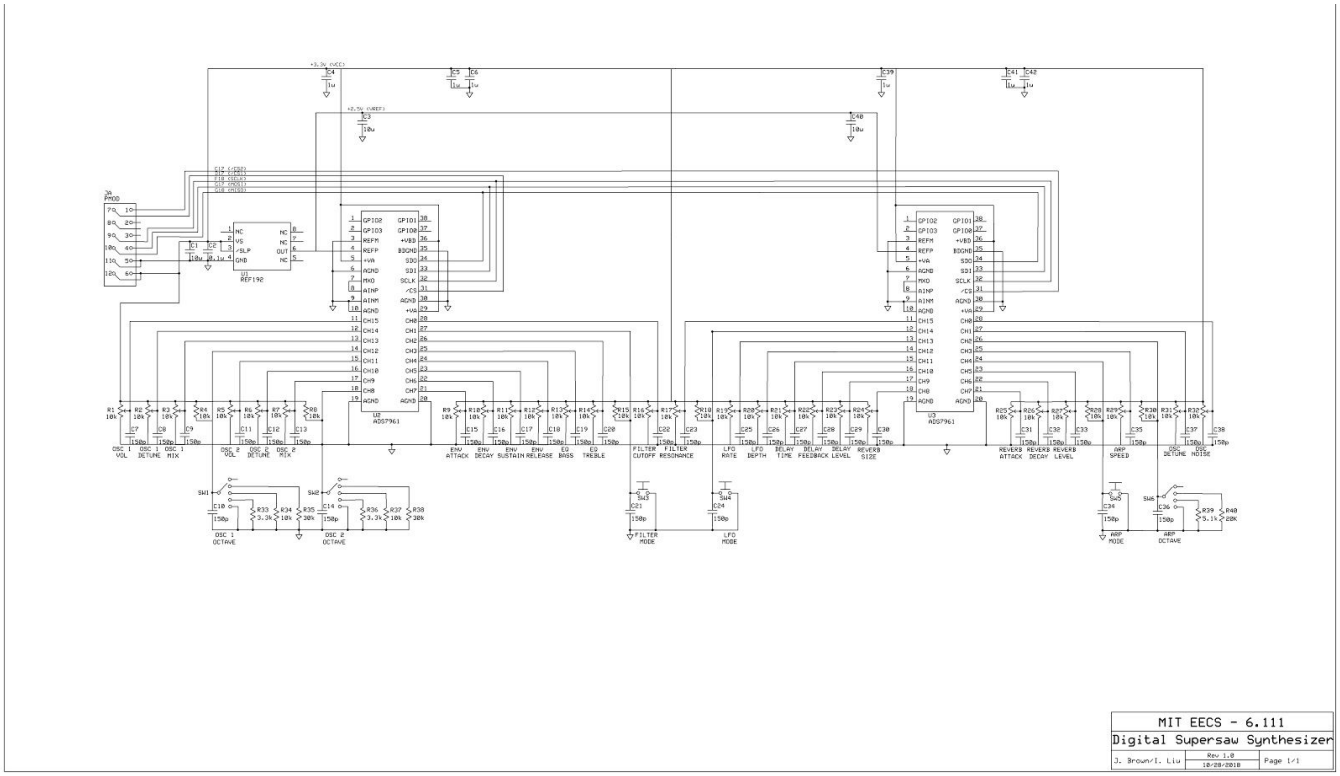


Figure 2 ADC for the analog inputs that control the synthesizer's various features.

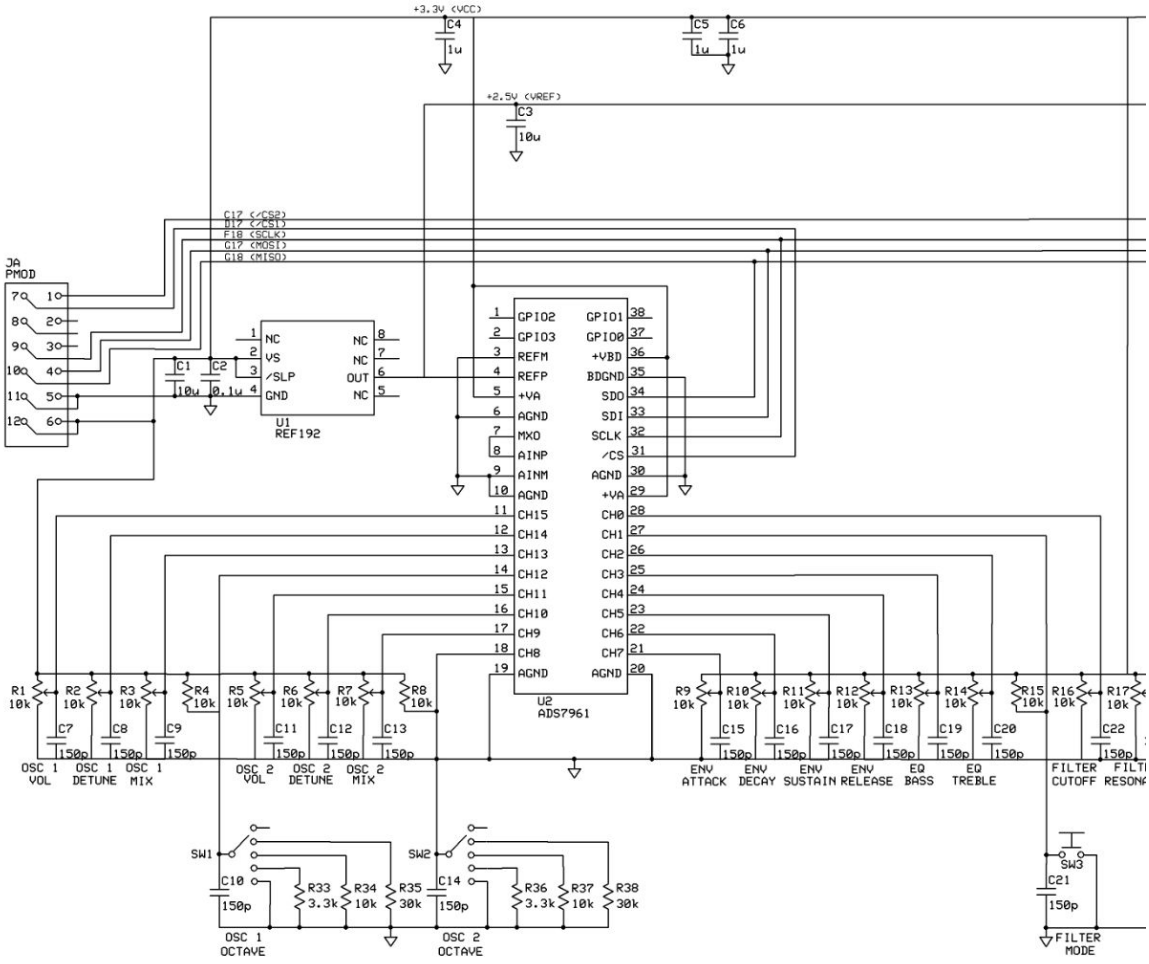


Figure 3 Enlarged view of left half of ADC.

References

1. Szabo, A. (2010) *How to Emulate the Super Saw* https://www.nada.kth.se/utbildning/grukth/exjobb/rapportlistor/2010/rapporter10/szabo_adam_10131.pdf
2. Roland Corp. (1996) *JP8000 Owner's Manual* *https://www.roland.com/us/support/by_product/jp-8000/owners_manuals/*
3. KVR Audio, Inc. (2009-2018) *Roland Supersaw* (forum discussion) https://www.kvraudio.com/forum/viewtopic.php?t=258924