

# ECE Project Proposal: Audio Effects Box

Anton Liakhovitch

27th November 2022

## 1 Background

Music producers employ a variety of effects in order to improve or modify their audio. Effects normally take one of two forms: *effects plugins*, and *effects boxes* (also known as *effects pedals*).

An effect plugin is computer software which integrates with a *digital audio workstation* (DAW) program via one of several standard plugin protocols. The DAW allows the producer to compose and record music, while plugins provide effects and other tools.

An effects box is a physical device which normally includes at least one audio input and one audio output. The box applies a given effect to the incoming audio signal in real time. For example, a delay pedal might add in delayed copies of the original signal for an 'echo' effect. Many software plugins use 'virtual analog' algorithms to emulate well-known physical effects boxes.

Sonic Charge - a music software company - makes an effect plugin called *Permut8*. The user interface is visible in Figure 1. On the left, a virtual analog section allows for basic mixing and filtering operations. On the right side, a digital section allows for manipulation of the signal to create various 'glitch' effects. The plugin is intentionally made to look and behave like an emulation of a physical effects box - however, no physical version of Permut8 has ever been built.

## 2 Goal

This project aims to create a physical effects processing box with similar functionality to Sonic Charge's Permut8 software plugin. The scope of the project covers only the *digital section* of Permut8; the 'virtual analog' section is mostly out of scope and may be added later through updated hardware or software.



Figure 1: Sonic Charge 'Permut8' Effects Plugin

### 3 Functionality

Permut8, the original inspiration for this project, is a form of digital delay effect. The plugin is designed as a software emulation of a fictional piece of hardware. This section provides a brief summary of this fictional hardware. For detailed information, see the *Permut8 User Guide*.

#### 3.1 Buffer

At the core of Permut8 is a memory buffer which stores 128,000 audio samples. The device is clocked at a variable rate, ranging from 0 to 352.8KHz. On each cycle, audio is sampled from the input and written to the buffer. Then, a sample is read from the buffer and sent to the output. The *write address* increments by one every clock cycle and wraps around to zero at the end of the buffer. By default, the *read address* is the same as the write address. Thus, by default, the plugin makes no changes to the incoming audio - it records an input sample into a buffer, then reads and outputs the same sample. In the interface (Figure 1), the read and write positions are visible on the 'LED bar' at the bottom.

The behavior is analogous to a loop of tape, where a write head is placed immediately in front of a read head.

## 3.2 Operations

Permut8 can apply up to two *operations* to the read address in order to make the output more interesting <sup>1</sup>. For example, the "SUB" operator subtracts a fixed value from the read address. This makes the read position lag behind the write position by some amount of time, thereby delaying the output signal. The rows of switches seen in Figure 1 configure input parameters for the operators.

## 3.3 Mixer

While the mixer is part of the 'virtual analog section', it is simple to emulate digitally. The mixer includes several controls:

- *Output level* controls the output audio level.
- *Mix* controls how much of the original, unmodified signal is mixed into the output.
- *Feedback* controls how much of the output signal is mixed back into the input.

## 4 Project Scope

This project aims to implement most of the functionality of Permut8's 'digital' section, and some of the functionality of the 'analog' section. This includes:

- The audio buffer.
- Both operators, with all available operations.
- Variable audio clock frequency <sup>2</sup>.
- Write protect, reverse, and reset switches.
- Stereo audio processing.
- The three mixer knobs (*output level*, *mix*, and *feedback*).
- The 'invert' and 'flip L/R' switches for feedback.

Notable omissions:

- The analog filter and limiter, and the input level knob.
- Tempo sync functionality.
- Preset-saving functionality.

---

<sup>1</sup>The Permut8 manual refers to these operations as "instructions".

<sup>2</sup>This refers to the processing algorithm only - it does not need to correspond to any actual hardware clock frequencies.

- Hexadecimal operand indicator displays.

There are also additional requirements based on practical considerations of physical hardware development:

- Standard connectors (1/4 inch or 1/8 inch TRS) for audio in/out)
- 16-bit or 32-bit DAC and ADC <sup>3</sup>.

## 5 Basic Design

The draft design described here is subject to change, and should only be used to gain an understanding of the technologies and challenges involved in the project.

The device will be based on a readily available 32-bit ARM MCU - for example, the STM32-F411 or the RP2040. The MCU will communicate with a stereo ADC/DAC IC over an I2S (or similar) interface. The audio buffer will be stored in an external RAM module, connected via either a parallel or SPI interface. Buttons, knobs, and LEDs will be accessed via a chain of shift registers. Firmware will be written in Rust.

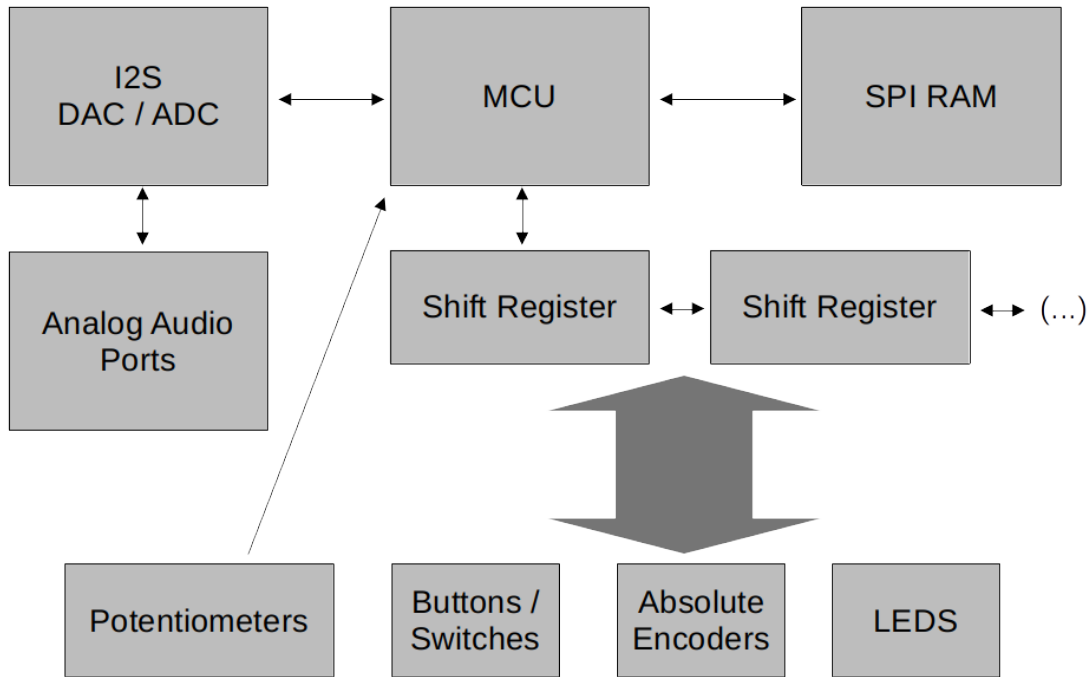


Figure 2: High-level Hardware Diagram

<sup>3</sup>The original Permut8 uses 12-bit audio. 24-bit is generally considered the highest resolution detectable by humans. However, extra resolution is useful in a hardware context because it allows for a wide range of input levels.

## 6 Optional Features

Additional features may be added once the rest of the project is complete. These features are optional, and no time should be spent on them until the rest of the project is done - however, some consideration may be put into forwards compatibility so that they can be easily added later. Potential optional features include:

- A display, encoder, and button which may be used to control menus.
- Tempo sync functionality (MIDI, MIDI-over-USB, or clock-pulse-over-TRS).
- The filter and limiter, as software or as analog circuits.
- A headphone amplifier <sup>4</sup>.

---

<sup>4</sup>This may turn out to be trivial, depending on the capabilities of the chosen DAC.