The MXR model 129 Pitch Transposer and model 131 display – The creators perspective. V1.0
Tony Gambacurta December 14, 2021 Revised Jamuary 21, 2022. V1.0

Background
It was the mid 70's, MXR had come out with the Model113 Digital Delay. The DDL was a success. I was in a unique position at MXR, the success of the DDL gave me a bit of a license to do what I wanted to I guess. I wanted to work on a pitch shifter. I thought it would be a powerful tool allowing musicians to play parallel harmonies in real time. No one had really done this. The Eventide 910 came out, but, it was unusable for harmonies due to "glitching". I wanted one guitar player to be able sound like Dickey Betts on Duane Allman playing together on Jessica. An impossible goal to be sure, but it was inspiration none the less.

I worked on the PT for a long time, 2 years around 1977 and 1978. That was a really long time for MXR. Usually once we decided on a path, things moved quickly, not so with the PT. During most of this time, I had very little encouragement, probably because my first attempts sounded so awful. I'll discuss this below when we discuss the Audio to Digital conversion.

Technology at the time was: 8 bit Microprocessors were just coming out and of little use to us except for the display. DRAM had come out. Small scale (SSI) logic was plentiful and cheap because of the computer industry. New medium scale parts were being introduced regularly. LSI was in it infancy. CMOS was coming on strong, but, it was slow. Computers were available as time-sharing services, one had to own or rent a terminal to access it. Mini computers were coming out. Scientific calculators were just out so we stopped using slide rules! No CAD. Pencil on paper drawings on drafting tables, a blue print machine, and hand taped PCBs on a light table.

Product goals:
Real-time pitch shifting that would work with guitar and other instruments to create harmonies.
Presets to allow quick selection of pitch interval.
Auxiliary loop jacks to allow additional processing of the pitch shifted signal, such as a delay.
Audio processing with Mixing, Regeneration (feedback) and Bypass and level matching.
Optional display that would show pitch intervals in musical half steps.

Whats needed to make a Pitch Shifter
Audio IO interface with level matching, Mix, regeneration, aux loop, bypass.
Audio to Digital and Digital to Audio converters.
Memory to store audio.
A digital controller to orchestrate reading and writing to memory.
A scheme to take multiple delay taps and fade between them to avoid hard splicing.
Pitch control circuitry including presets

How a Pitch Transposer works
This is the way I always thought of how a pitch shifter works and how I explained it to others the last 40+ years.

Imagine we have a rotating disk that we can record audio on, say, magnetically. We have a record head, and 2 playback heads and one track. Since its imaginary, the play heads can pass through the record head. The 2 play heads are movable (rotate) and always are positioned 180 degrees apart on the rotating disk.

In operation, the disk rotates and audio is recorded. We can listen to either play head which are time delayed versions of the recorded audio. When we slowly move the play heads, keeping the record head in a fixed position, the output of the play heads will be changing delay time with respect to the record head and the pitch will be shifted as a result. If we listen to a play head as it passes the record head, we potentially get a huge discontinuity in the signal as we are switching the time delay abruptly between the maximum delay and zero delay. So, the goal is to not listen to a play head as it crosses the record head, instead, always listen to the head that is 180 degrees away. Then, as the head we are listening to approaches the record head, we cross fade to the other head which has just crossed.

In the actual unit, we replace the disk with memory to store digital audio samples. Adapting our rotating disk analogy, think of disk as being stationary and the record rotates around the disk as it records. The play heads also rotate around the disk in the same direction as the record head but has the ability to do so at a different rate depending on the desired pitch shift.

There were 3 asynchronous clock domains. There are 2 clocks that represent the sample rates of the of the audio, one for writing to memory, the other for reading from memory. These “sample clocks” do a lot. The ratio of these 2 clock determines the amount of pitch shift. The 3rd clock is a high speed clock that runs a digital state machine that handles the timing to read or write to memory as well as A-D and D-A timing.

The MXR Pitch Transposer circuit highlights.

Audio IO
This section is straight forward. It provides input and output processing, MIX, Regen controls and handles bypass. Four poles of input filtering and output filtering on paths to and from the A-D, D-A.

A-D, D-A
I like referring to A-D as Audio to Digital converters instead of Analog to Digital converters because these are specific to audio signals, not just any analog signal.

This part of the design represents one of my proudest accomplishments. As I mentioned, a lot of time was spent trying to get a decent sounding system. I was trying to make it work with a sigma - delta modulator. I never got that to work satisfactorily. I gave up on it and spent time looking for another way to encode audio efficiently with good fidelity and low cost. I did a lot of research.

I came up with what I call **non-linear differential encoding**. Differential encoding takes the previous approximation of the audio signal, compares it to the current audio sample, and digitizes the difference. It both encodes it to digital, and grabs the version of the signal, with its error, from the D-A converter that was used to digitize the difference. Decoding takes each sample, puts it through a D-A, and accumulates it with all the previous samples in an integrator. The integrator has a switch to act as a sample and hold and allow the amount of integration per sample to be independent of sample rate. Differential encoding allows

encoding signals larger than the D-As maximum output, it can take a few samples to get to full scale if needed. Most importantly, it corrects for errors in the previous samples due to the feedback from the D-A converter.
Non-linear encoding is interesting, but, doesn't sound very good by itself. My theory about that is that our ear hears the worst encoding, the course encoding as the signal nears full scale. However, non-linear when combined with differential encoding is a marriage made in heaven. The D-A can put out large signals when needed to help large signals and transients, but still have good resolution near zero as it acts like a multi-bit sigma-delta converter. The best of all worlds. Subjectively, with musical instruments, I felt we got about the equivalence of 12 bit performance storing only 8 bits per sample.
What made this easier to do was an 8 bit, mu-law, D-A from AMD, the AD6072 and the one chip successive approximation register (AM2502). This A-D/D-A system was also used in MXR digital delays after the 113.

Sigma delta sidebar:
Sigma delta is a one bit at a time encoding scheme. Decoding can be as simple taking the bit stream and running it through a low pass filter.
Most modern (in the 2000's) A-D converter systems are based on sigma delta technology using IC technology. Back in the day, these were unavailable. A basic A-D would be a comparator feeding a flip-flop D input clocked with a high speed sample clock, assume around a MHz. An RC filter from the output of the flip-flop the feeds one side of the comparator input, the other being fed by the incoming audio. This simple scheme can encode analog signals. It requires a lot of bits, and/or clever circuits to replace the RC. One problem is that in the Pitch Transposer, when the read crosses the write, we get a big discontinuity that Sigma Delta takes a long to to recover from. I couldn't get it to recover before we started to fade to that output. Faster recovery filters were poor fidelity.

Memory
In the MXR Model113 Digital Delay, I had used the 22 pin 4k DRAMs. These were awful parts to work with. I stuck with DRAMs, but the 16 pin which were much easier parts to work with even though they also required +12V and -5V supplies. I was not running these anywhere near their timing limits and was able to use RC timing circuits for the RAS, CAS and WE signals. 16 Kbits of DRAM total. No extra refresh cycles were needed.

Digital control
The digital section is interesting. It is a state machine that orchestrates memory read and write cycles. It runs on its own high speed clock generated by a74LS14 with RC timing. Frequency wasn't critical as long as was fast enough to service the read and write cycles worst case which is when there is near 0 pitch shift.
Handling the triggering of cycles from the read and write sample clocks, was tricky. Using a JK flip flop to hold the state of a read or write cycle was used so that when both a read and write cycle were requested within the same HS clock cycle, it would do the opposite from what it did the previous cycle. I, and many other engineers, had not heard of meta-stability at this time, maybe ignorance is bliss?
The state machine was built using a counter and a small bipolar PROM, 32 x 8 bits. These PROMs were powerful for logic replacement. Like early FPGAs. We programmed these, but, they are not erasable.

This section includes separate read and write address counters and circuitry to detect when they are equal (heads crossing).

Analog Control
A scheme to take multiple delay taps and fade between them to avoid hard splicing (glitching). In this section, its the right side of the upper board, a lot is going on. Audio sampling is triggered by the sample clocks. The digital section including A-D and D-A operate on the HS clock time domain. This is done with sample and holds.
This section also handles mixing audio from the 2 read heads to splice audio. The rate of fade between taps is determined by a signal from the lower board called VC.
A “correlater” watches the correlation between the 2 read taps. If they correlate better over time in phase then it splices them that way, if it looks better out of phase then it splices out of phase. Crazy stuff. I could have patent that. Sounded much better when playing single notes but didn’t seem to interfere with complex material. This algorithm was put into the pitch shifting in an ART product called the DMV-Pro. In the Pitch Transposer, this feature probably should have been defeatable.

Pitch control circuitry including presets
This is the fun part of the design.
Preset selection, which simply selects one of 4 front panel pots determine the pitch. On the front panel, the pitch control knobs are touch sensitive. Touching the knob selects that preset. I got the idea for this from Jim Cooper (J.L. Cooper Electronics) while he was working for Tom Oberheim. He told me he got it from Don Lancaster’s CMOS Cookbook. This is a very robust way to do touch sensitive “buttons”. A foot switch can also be used to select presets. The foot-switch box has 5 push buttons, each with a resistor going to a mono phone jack which connects with a guitar chord to the main unit. The 5 resistors are a sequence something like 1k, 2k, 4k, 8k 16k. This allows the receiving circuit to determine which button, with the lowest resistance is being pressed even if more than 1 button is pressed. So users can hold one button while rocking their foot onto another button with a higher priority. Hold down the “third” and rocker to the minor 3rd as needed.
The footswitch is “or’d” with the front panel touch knobs. This feeds a circuit that prioritizes the selections and latches the latest selection. This circuit using a CD4532 priority encoder is very clever and was probably inspired by Don Lancaster’s CMOS Cookbook. It also uses the other half of the CD4052 analog mux that is used to select which pot is determining pitch. 2 packages discriminate which preset is selected and latch it with no clocks!
The rest of the sample clock generation takes a 0 to 10V control voltage and drives the sample clocks which are 2 555 timers. Sample clocks are at their maximum when there is zero pitch shift. When want to shift pitch up, we slow the write clock, when we want to lower pitch we slow the read clock. We also generate a VC to determine the fade rate of the splicing. There is a cute 2 transistor rectifier that helps with this.

Pitch Transposer Display Model 131.
This was my first product using an embedded micro-controller, before they even called it “embedded”. I felt that if we were to provide a numerical indication of the amount of pitch shift it should show it in musical half steps rather than as a frequency ratio. An octave down displays as -12.0 rather than 0.50, an octave up is +12.0 rather than 2.00. The way I chose to do this was with a triple precision (24 bit) base 2 log routine. I got the algorithm from Donald Knuth’s “The Art of Computer Programming”.
The display works by taking the 2 sample clocks from the main unit and measuring their ratio. It uses the micro’s internal counter/timer for one sample clock and an external counter for the other. The chip was a Mostek 3870 based on the Fairchild F8 family. It was programmed at

the factory in an on chip non-programmable ROM. We had to order something like 2000 parts on the first order.
For development, an electronic distributor loaned me a development system (thanks Doug Pucci). It was a beast. For prototyping they had versions of the IC that were packaged with a socket on top for a UV EPROM. These development parts were way too expensive for putting in the production displays but allowed us fully functional prototypes executing the exact same program the ROM versions would. BTW, somewhere out there are displays that have these prototyping parts with the UVPROM plugged in to a socket on the microprocessors. Somehow these engineering protos got shipped!

Thoughts looking back.
The Pitch Transposer and Display are one of my proudest accomplishments. It pushed me and it pushed MXR. Hopefully it allowed musicians and producers to push their art also.
It was used most famously by Trevor Rabin on Yes “Owner of a Lonely Heart”.

My favorite use was going out to see my friend Rob Storms with “Rob and John”. Rob used it on vocals with the footswitch. He could select 3rds, minor 3rds, 4ths and 5ths. I remember being impressed with how well he used it.

Circuitry wise, there is lot of R-C timing delays and pulse shapers give me the feeling of “I would never do it that way now”, however, I am not sure any other way would have been better given the technology at the time. Any other way would probably not be cheaper. There are too many trim pots.

Technology has come a long way since this product’s time. I think that is a good thing. I was using the best available low cost technology to achieve something new to the world. In the 70’s it was a LOT of work to do pitch shifting. 2 years of tinkering with variable digital delays made of experimental A-D and D-A converters trying to get something that would sound good enough and be low enough cost for musicians to be able to afford it. It was not easy.

With today's technology it’s trivial, just write the code. I think that is awesome. Doing audio processing in the digital domain is much easier and better than the analog way. I have had a lot of experience with both. Digital sounds better too, although it may not be as organic.

People that helped: (that I can recall 40 years later)
Richard Neatrour
Jim Stachoski
Steve Donaldson
John Porubeck
Jeff Pokrant was a tech that worked on PTs for many years. If you own a PT, chances are Jeff was the technician that set it up.
Also Keith Barr, Terry Sherwood, Phil Betette, John Langlois, Ron Wilkerson.