Audio Engineering Society

Convention Paper

Presented at the 146th Convention
2019 March 20–23, Dublin, Ireland

A Custom Integrated Circuit Based Audio-to-CV and Audio-to-MIDI Solution

Brian Kaczynski[1]

[1] *Second Sound, LLC, 999 SW 1st Ave. Apt. 1617, Miami, FL 33130, USA*

Correspondence should be addressed to Brian Kaczynski (briankaz@secondsound.com)

ABSTRACT

A new synthesizer technology is demonstrated which tracks the fundamental frequency of virtually any acoustic or electric instrument played monotonically. This technology relies on a mixed analog-digital application-specific integrated circuit (ASIC), which contains a very fast frequency-locked loop (FLL) that tracks with the minimum physically achievable latency of one audio cycle. The ASIC also contains a novel fundamental frequency detection circuit composed of two switched-capacitor peak detectors with decay time proportional to the fundamental period of the audio signal, and a novel switched-capacitor, zero-ripple envelope follower. This frequency-tracking technology is fast enough to implement an audio-to-CV or even, with the addition of a simple microcontroller, an audio-to-MIDI solution in real time with very high accuracy and negligible latency.

1 Frequency-tracking synthesizers

Music synthesizers which track the frequency and/or dynamics of an instrument are nothing new. As soon as people became accustomed to hearing the new sounds analog synthesizers could generate, it was only natural that they would want to ‘play’ these synths with the instruments they were already used to, such as guitar and vocals. Thus, in the late 70s, products such as the Korg MS-20, and the Roland SPV-355 and GR-300, emerged to bridge this gap between instrumentalists and those analog synthesizers. These products were equipped with dedicated interfaces for converting audio from guitar or other instruments to control voltage (CV) for controlling frequency and/or envelope in analog synthesizers and presented an intriguing alternative to a keyboard for non-keyboard players.

Much later, products such as the Sherman Filterbank (introduced in 1996), and the Schumann PLL pedal (introduced in 2002) made history with their architectures based on phase-locked loops (PLLs). These PLLs required their settling time and damping to be tuned carefully and could easily devolve into chaos (not necessarily a bad thing for music!), although these products did earn scores of fans who were attracted to their unique sounds and possibilities.

What these earlier frequency-tracking synths have in common is an abundance of, to put it euphemistically, ‘character’. Their tracking must be tuned, they can behave idiosyncratically and can produce unwanted results if challenged too much (like with a harmonically ‘difficult’ instrument such as a trumpet or violin), and they simply cannot react with short enough latency to be compatible with MIDI implementation.

2 Fundamental frequency detection

The greatest challenge to tracking the frequency and dynamics of an arbitrary musical instrument is, perhaps not surprisingly, detecting the fundamental frequency itself. These challenges have been amply addressed in academic literature, mostly in the sphere of voice processing. To illustrate the difficulties encountered when tracking the fundamental frequencies of various instruments, it is helpful to examine some time-domain waveforms. Figure 1 shows four cycles of typical waveforms for a bass guitar playing 55 Hz, a trumpet playing 237 Hz and a violin playing the open A string at 196 Hz, which is especially difficult to track because the amplitude of the second and/or third harmonics can be as much as 20 dB greater than the amplitude of the fundamental.

Interestingly, the circuit techniques used in the vast majority of frequency-tracking electronic instruments, including the aforementioned products manufactured by Korg and Roland, come from technology existing since the mid-60s and sometimes even earlier. The Korg MS-20, famous for its 'external signal processor', which converted music from monophonic sources to pitch control voltage (CV), relied on fixed filtering (adjustable by the user to accommodate various instruments) followed by comparators with hysteresis [1]. This technique was categorized by McKinney in 1965 as the threshold analysis basic extractor (TABE) with hysteresis [2].

The Roland SPV-355 added an automatic gain control (AGC) to the basic TABE approach followed by a tunable bandpass filter and a clever four-phase envelope detector (four phases to reduce ripple when processing low frequency signals) and feedback loop which attempts to servo the filter cut-off frequency to a sweet spot to remove higher harmonics as well as low-frequency noise [3]. The Roland GR-300, made famous by Pat Metheny, is considered by many to be the epitome of analog guitar synthesizers. It contains dedicated processing electronics for each string featuring a filter tuned to its frequency range, with either a low-range or a high-range bandpass setting that gets set in a clever feedback configuration depending on the detected frequency [4]. These methods all fall into the category of fundamental frequency detectors with tunable filters that apply a closed-loop control system, described in various incarnations from 1948 [5] to 1976 [6].

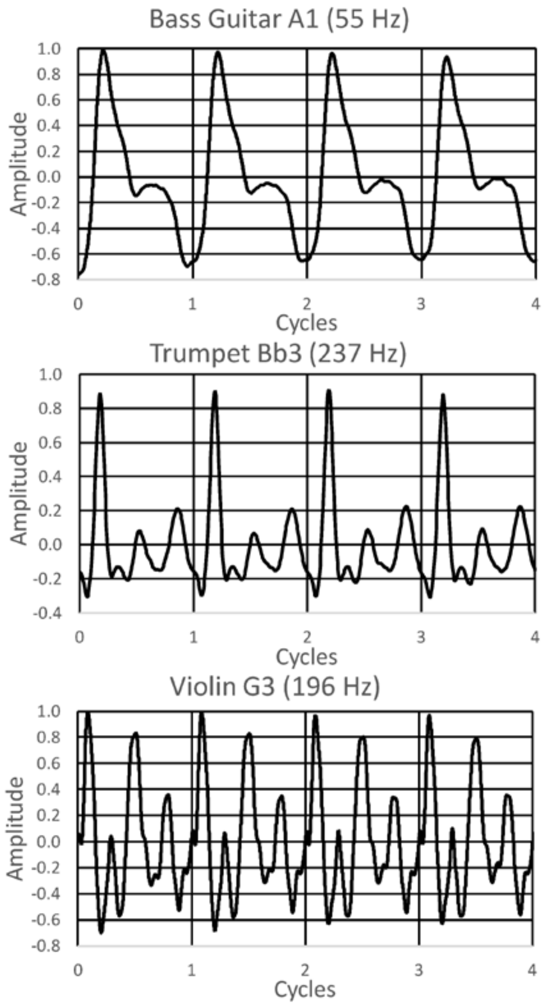

Figure 1. Waveforms of selected instruments

There are several problems with these early attempts at fundamental frequency detection for music synthesis including the following: (1) the impossibility of filtering the second and higher harmonics sufficiently for instruments with complex spectra, such as violins; (2) the susceptibility of any 'tuned filter in servo loop' approach, such as Roland's, to perceptible delays in its transient response, especially for bass instruments in the 30–100 Hz range; (3) the fact that these circuits often contain tuned elements which make working over a range of greater than 2-3 octaves problematic if not impossible; (4) the tendency of the auto-tunable bandpass approach in particular to exhibit tracking errors upon playing a high note followed by a low note that momentarily falls into the high-pass stopband — in this case the mechanism is very likely to lock to a higher harmonic of the fundamental. This last problem was even described as early as 1956 by Edson and Feldman [7].

Note that this author does not consider it necessary to 'prove' that these late 1970s approaches to frequency-tracking music synthesis do not meet the latency requirements to perform a satisfactory conversion of audio to MIDI. Discussion of the topic with a sufficient number of professional musicians who have experimented with the products has proven more than convincing.

3 Analog vs. Digital Techniques

The elephant in the room for fundamental frequency detection technologies is digital techniques using autocorrelation and/or spectral analysis. The vast majority of the contemporary literature on frequency tracking deals with DSP algorithms such as the YIN algorithm [8]. However, despite the dominance of digital techniques in frequency tracking literature, analog methods can still offer some advantages.

Digital frequency detection algorithms generally operate on a fixed window of the audio signal. Consider a digital fundamental frequency extractor based on autocorrelation which must detect frequencies down to 25 Hz. It must operate on a time window of at least 40 ms in order to capture the lowest frequency of interest. It is well known that the lowest detectable frequency sets the minimum achievable latency for an autocorrelation-based digital frequency tracking method such as YIN or ESPRIT [9]. In contrast, an analog approach can theoretically exhibit a latency very close to one audio cycle, regardless of signal frequency.

Another difficult to quantify yet important advantage of an analog solution to the fundamental pitch identification problem is that when it gets challenged by corner cases (which can occur all too easily in signals from musical instruments) it tends to fail much more gracefully than its digital cousins. For example, it is sufficient to play polyphonically through an analog pitch-tracking synthesizer meant for monophonic signals and this advantage is readily apparent. It is generally easier to find examples which are still 'musical' in these cases than with digital versions.

4 Peak vs. Zero-Crossing Detectors

Since filtering and zero crossing or threshold detection (with or without hysteresis) do not achieve the desired low latency for a satisfactory audio-to-MIDI solution, it is worth exploring other fundamental frequency detection methods. A cursory look at the waveforms in Fig. 1 might lead one to believe that detecting peaks instead of zero (or offset from zero) crossings would be a useful method to try. Schematics available online demonstrate that the EBS OctaBass, a pedal introduced in 2002 which synthesizes a sub-octave, uses dual peak detectors and a flip-flop that can be set when the positive peak exceeds the positive threshold, and reset when the negative peak falls below the negative threshold [10]. The approach used in the OctaBass was first described by Angus, Garner and Howard in 1998 [11]; interestingly, this technique was also applied there in the context of speech processing.

The problem with the OctaBass implementation is that the audio source is thoroughly filtered to bass guitar range and the peak detector thresholds and decay are also 'tuned' to fixed time constants, thus it only works well over a limited range of frequencies (according to the OctaBass Users Manual the octaver -3dB bandwidth is 10–110 Hz).

5 PLL-based music synthesizers

To track to the fundamental frequency of an audio input within one fundamental cycle of time, it is necessary to sample that audio signal many times per cycle. With this in mind, it is useful to take a brief detour and explore clock multiplier architectures. Two music synthesizer products based on phase-locked loops (PLLs) are especially interesting: the Sherman Filterbank and the Schumann PLL pedal. The classic PLL integrated circuit is the 4046 with all its variations, which is used almost exclusively in these applications. As illustrated in Fig. 2, PLLs work by tuning a slave oscillator using a feedback loop to exactly match the frequency of a reference clock (or multiple thereof).

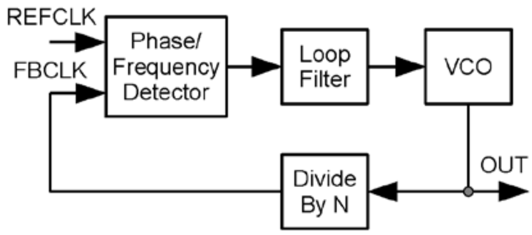

Figure 2. PLL Block Diagram

PLL-based music synthesizers face the same challenge of fundamental frequency detection as that discussed above, as the reference clock must be a square wave running at the fundamental frequency of the audio signal. Both the Sherman Filterbank and the Schumann PLL pedal utilize fixed filtering followed by a hard-clipping amplifier [12]. In the case of the Schumann PLL, the clipping amplifier threshold can be adjusted within approximately ±0.5V. In other words, both approaches fall into the category of the TABE discussed above.

Despite the points made above, PLL-based frequency-tracking synthesizers face what may be an even bigger problem than fundamental frequency detection. Because they are based on a feedback loop, this creates a trade-off between frequency tracking speed and stability. A good rule of thumb is that the loop bandwidth cannot exceed around 1/5 of the reference frequency [13]. This means, for example, that a stable PLL can never be made to track bass notes in the 30–100 Hz frequency range with latency shorter than than five or so cycles, which will most certainly be audible as latency. Additionally, since PLLs strive to drive phase error to zero over time, phase error can accumulate when the input frequency jumps over a large range after a PLL. This causes slewing, cycle skipping, and even longer delays in the transient response than the loop bandwidth derived from the linear feedback system model would suggest.

6 Frequency-Locked Loop

For fastest frequency tracking, the only option is a frequency-locked loop (FLL), as it is not a difficult task to design an FLL that can lock to the reference frequency within one reference cycle.

Figure 3 shows a block diagram of the basic digitally-controlled FLL integrated into the present work and described in detail in US patent #9685964, 'Fast-locking frequency synthesizer' [14]. It contains: (a) a digitally-controlled oscillator (a DCO) implemented as an analog RC relaxation oscillator with a digitally-controlled resistor value in exponential steps over an almost eight-octave range; (b) a 22-bit counter which counts N_{DCO} — the number of DCO cycles that elapse between successive reference cycles; (c) a digital frequency-iteration engine which computes the next digital code, based on the number of DCO cycles counted by the 22-bit counter, as a 16-bit number with an 8-bit integer part 'Nint' and 8-bit fractional part 'Nfrac'; (d) a sigma-delta modulator which modulates the 8-bit frequency control word of the DCO every cycle to give the DCO eight more bits of effective frequency resolution.

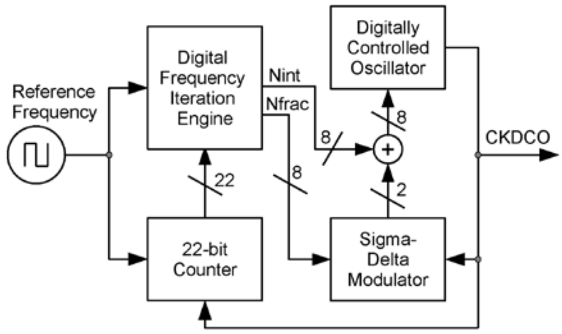

Figure 3. Frequency-Locked Loop

The frequency F_{DCO} of the DCO depends on the input code X_{DCO} approximately as follows:

$$F_{DCO} = F_{min} 2^{\frac{X_{DCO}}{32}} \tag{1}$$

where Fmin is the minimum DCO frequency and is equal to about 25 Hz * 8,192 = 204.8 kHz (the relationship becomes slightly nonlinear at the highest frequencies due to finite comparator delay in the relaxation oscillator, but this does not have a significant impact the analysis). To drive N_{DCO} to 8,192, the digital frequency iteration engine estimates the base-2 logarithm of the fraction N_{DCO}/8,192 using a continuous piecewise linear approximation. Because of the exponential nature of the F_{DCO}-vs-X_{DCO} curve, it can be seen that the new X'_{DCO} should be chosen as follows:

$$X'_{DCO} = X_{DCO} - 32 \log_2 \frac{N_{DCO}}{8{,}192} \qquad (2)$$

Figure 4 shows this FLL settling behaviour for a trumpet playing F4 (352 Hz) and a bass guitar playing A1 (55 Hz). The REF_CLK signal is the detected fundamental and the SINE is a sinusoidal fixed-envelope output derived from the DCO frequency. The frequency settles to the correct value within one cycle in both cases. Verilog is used to simulate this system because its event-driven nature makes it particularly amenable for simulating a mixed analog-digital system with variable sample rates.

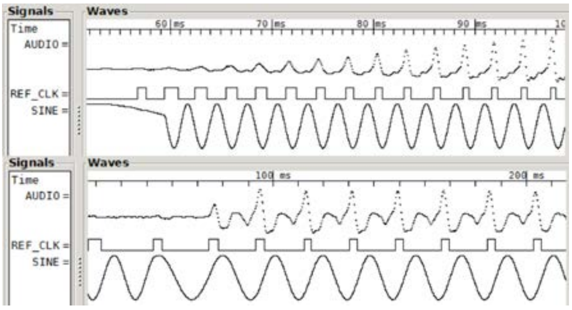

Figure 4. FLL Settling Simulations

These plots illustrate the sine waveform instead of a pitch control voltage (CV) because it is much easier to see the frequency locking time with sine waves. The pitch CV is generated by simply converting the digital frequency code to analog with a DAC, which introduces no additional latency.

Using such a high DCO frequency serves the dual purpose of (1) making the noise from jitter caused by the 8-bit sigma-delta modulator inaudible; and (2) enabling this work to employ a unique fundamental frequency detector, based on switched-capacitor peak detectors, which avoids all of the shortcomings of the solutions discussed above.

7 Switched-Capacitor Peak Detector

The reference frequency shown in Fig. 3 must of course be derived from the audio input using some kind of fundamental frequency detection circuit. Fortunately, it is possible to make a very wideband peak-detector-based fundamental frequency detection circuit using switched-capacitor circuits, which are well understood to be compatible with modern ASIC technologies.

What is required is a peak detector with a decay time constant that scales with the detected fundamental frequency of the audio signal. Low-frequency signals should have a long time constant and, conversely, high-frequency signals should have a short time constant, with the net result being that the peak waveform decays by an amount between peaks which is independent of signal frequency. Such a peak detector is shown in Fig. 5.

To achieve the required signal frequency dependent decay time in the peak detectors, it is sufficient to replace the resistor commonly used to bleed charge off the peak detector 'hold' capacitor with a switched capacitor clocked at rate generated by dividing down the CKDCO clock running at 8,192 times the audio fundamental. Let us examine how this works based on the components illustrated in Fig. 5.

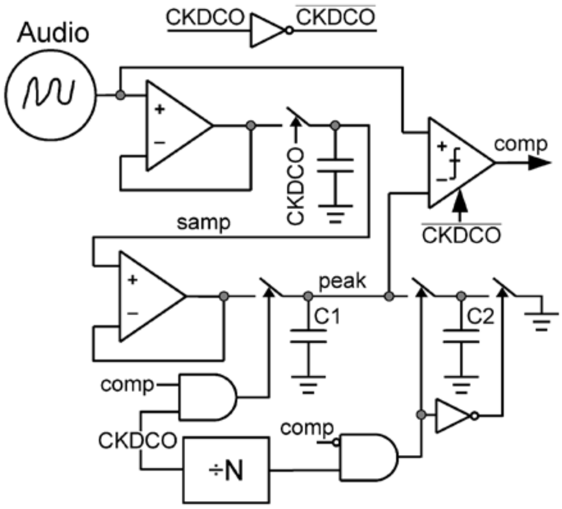

Figure 5. Switched-Capacitor Peak Detector

First, we have a sample-and-hold amplifier clocked by CKDCO and generating the 'samp' signal. This is followed by another sample-and-hold amplifier which is enabled only when the audio input is greater than the stored value 'peak' on peak hold capacitor C1.

Generally, the 'peak' signal retains its value whenever the audio input voltage is lower than this peak voltage. However, when the audio signal is below the peak, making the 'comp' output of the comparator low, the divide-by-N counter followed by the AND gate are active, which allow capacitor C2, initially discharged to zero volts, to be shorted across the peak hold capacitor C1 periodically. If the voltage across C1 is Vpeak[0] before any shorting event occurs, the voltage will subsequently be reduced due to charge sharing as follows:

$$Vpeak[1] = \frac{C1}{C1+C2} Vpeak[0] \tag{3}$$

After such charge sharing events have occurred n times, the peak voltage will be reduced to Vpeak[n]:

$$Vpeak[n] = \left(\frac{C1}{C1+C2}\right)^n Vpeak[0] \tag{4}$$

Since the CKDCO signal is a clock running at 8,192 times the fundamental frequency, roughly 8,192/N charge sharing events will occur per audio cycle and the peak will therefore decay by a constant amount per cycle, independently of the audio frequency.

The amount of decay can be adjusted by either changing the ratio of C1 to C2 or the division factor N or a combination of both. The ASIC implements eight possible decay factors by allowing N to take the values 128, 256, 512 or 1024 and the ratio of C1 to C2 to take the values 55 or 79. Making the decay longer gives more immunity to harmonics at the expense of tracking speed for fast envelopes and is thus appropriate for bowed strings, whereas shorter decays are appropriate for plucked stringed instruments such as the bass guitar.

In practice, the sample-and-hold amplifiers are implemented with fully differential operational amplifiers, bottom-plate sampling, non-overlapping clock generators and other techniques that would be expected in an integrated implementation for maximum performance. The schematic shown in Fig. 3 should therefore be understood as a simplified version and not what is literally present on the ASIC.

The complete switched-capacitor peak detector consists of two circuits as depicted in Fig. 5 — one operating on the audio signal and the other operating on an inverted version of the audio signal. Because differential signaling is used, the inversion can be obtained simply by swapping the positive and negative polarity signals. This creates a positive peak signal ('ppeak') and negative peak signal ('npeak'), which are shown in Fig. 6 using the violin playing the bowed open G string (196 Hz) for a C1 to C2 ratio of 55 and three different rate settings (rate = 0 corresponds to setting N = 128, rate = 1 corresponds to setting N = 256 and rate = 2 corresponds to setting N = 512).

The comparator outputs ('comp') from both peak detectors are fed into the inputs of an SR-latch and its output serves as the fundamental frequency for the FLL.

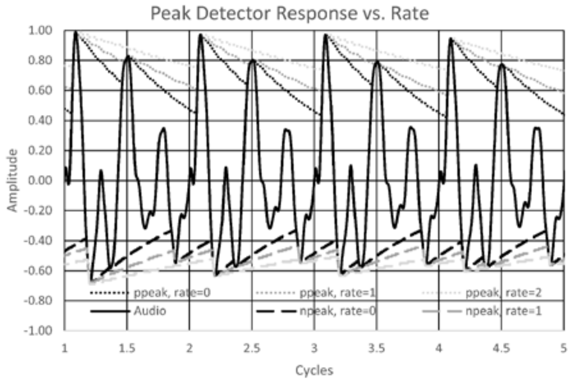

Figure 6. Peak Detector Response vs. Decay Rate

It is evident from the plots that the FLL will lock to the second harmonic of 196 Hz if the rate setting is 0 or 1 but will correctly lock to the fundamental if the rate is set to 2 or higher. Instruments with a simpler harmonic profile, such as the bass guitar, can benefit from the increased agility of the lower rate setting as they do not contain as much energy to reject in higher harmonics.

8 Zero-Ripple Envelope Follower

Before addressing audio to MIDI conversion, let us examine another basic analog synth building block that comes almost for free with a fundamental frequency detector based on dual switched-capacitor peak detectors: a zero-ripple envelope follower. Because we have a fast sample clock and a comparator telling us when the input audio is instantaneously greater than the peak, we can add one more sample-and-hold amplifier which transfers the 'peak' signal to the output 'env' at specially-chosen times.

As illustrated in Fig. 7, the 'peak' voltage is sampled onto the sampling capacitor C whenever one of the following two events occur: (1) the 'comp' signal is low and was high on the previous CKDCO cycle; or (2) the output of a 14-bit counter, which is held in reset while 'comp' is high, overflows. This can be understood as follows: When the 'comp' signal transitions high to low, this means the 'peak' signal represents the maximum value it had over the last audio cycle — so naturally, that value is an accurate instantaneous estimate for the envelope.

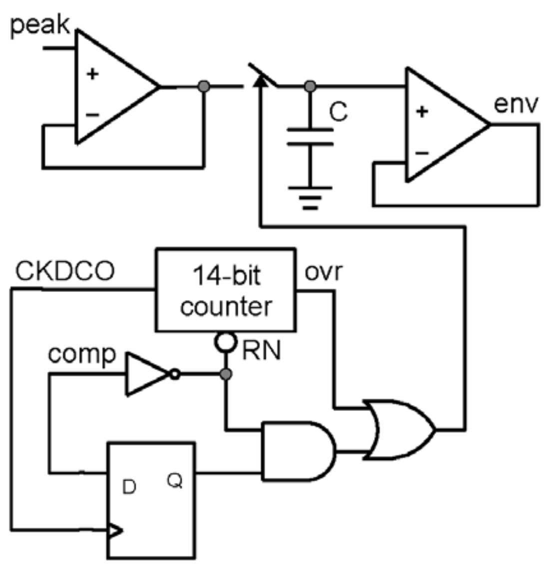

Figure 7. Zero-Ripple Envelope Follower

Let us consider what happens when the 14-bit counter overflows. Because there are 8,192 CKDCO cycles in an audio cycle, and 16,384 CKDCO cycles until the 14-bit counter overflows, the envelope follower will be forced to sample the decaying peak signal every TWO audio cycles if a peak is not detected. This allows the envelope to decay even when no peaks are detected.

This 'env' output can be further filtered with a standard switched-capacitor filter clocked at a frequency derived from CKDCO to smooth out the step transitions so they are not audible. The resulting filtered envelope is a true 'zero-ripple' envelope as it remains flat regardless of the input frequency and its decay time always remains proportional to the fundamental period of the audio signal.

Note that the actual sample-and-hold amplifier implemented in the ASIC uses only a single differential op amp (not two as shown in Fig. 7), differential signaling and bottom-plate sampling; therefore, the schematic shown should again be considered to be a simplified version.

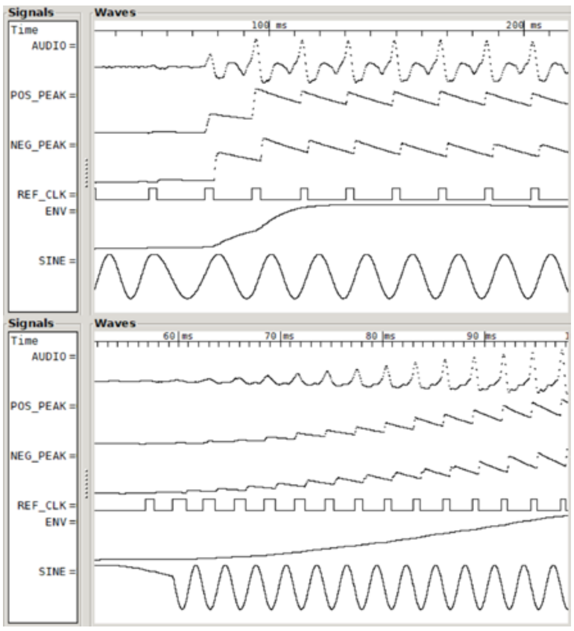

Figure 8. FLL Settling with Additional Waveforms

Figure 8 shows the FLL settling simulation with the positive and negative peak and zero-ripple envelope signals added, illustrating that the circuits can handle both the fast-attack low-frequency bass guitar signal and the slow-attack trumpet signal equally well. It also shows the envelope signal 'ENV', which is generated as the average of the filtered positive and negative zero-ripple envelopes. These individual positive and negative envelopes are not shown because they are difficult to distinguish from the composite ENV signal at the level of scale of Fig. 8.

9 Adaptive Polarity Selection

Much has been written in literature about the importance of selecting the proper polarity or phase of the audio signal for fundamental frequency detection. Common sense tells us that the polarity with larger peaks should be chosen to maximize the immunity to jitter noise at the cycle detection times.

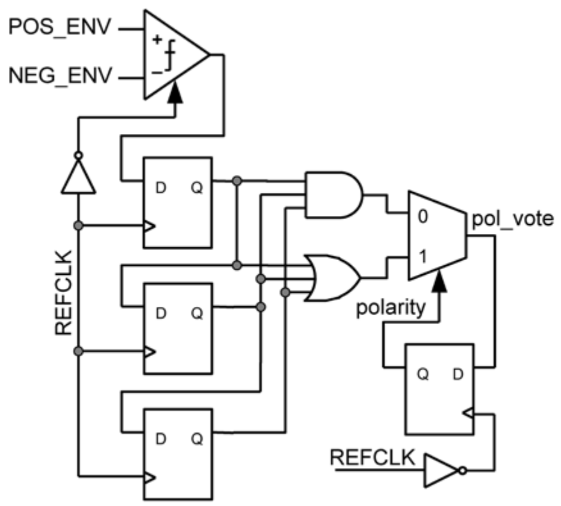

Figure 9. Adaptive Polarity Selection Circuit

This work implements an adaptive polarity-selection scheme by comparing the filtered positive and negative zero-ripple envelope signals and selecting the polarity with the larger envelope. To avoid limit-cycle oscillations, there must be three consecutive larger amplitude samples of a given polarity before that polarity is selected. This is illustrated in Fig. 9. If the POS_ENV signal is larger than the NEG_ENV signal for three consecutive reference clock cycles, the outputs of the D flip flops are all high and the outputs of both the 3-input AND and OR gates, as well as the 'pol_vote' and 'polarity' signals, are therefore also high. It is evident from that state onward that three consecutive cycles in which NEG_ENV is larger than POS_ENV will have to be registered in order for the polarity to switch to low. Conversely, when the polarity is low, it takes three consecutive cycles for the polarity to switch back to high.

Of course, a few additional precautionary measures must be taken which are not shown. Firstly, care must be taken so that the reference clock does not glitch when switching from one polarity to the other. Secondly, the fundamental frequency must be ignored for one cycle when the polarity switches so that the digital frequency iteration engine will not process a cycle of incorrect length.

10 Audio to MIDI

Audio to MIDI is now possible with the chip described here with satisfactory latency even for live performance situations. As described above, the chip stores its current frequency in a 16-bit word. The chip has a 'MIDI out' feature which shifts the 16-bit frequency word out every time it is updated via two 8-bit writes out of a standard SPI interface. This, together with the fact that the 16-bit frequency varies logarithmically with the frequency of the audio signal, improves the ease with which the frequency can be converted to the appropriate MIDI, including the option of a pitch bend.

Combined with measurements of the zero-ripple envelope via a crude A/D converter (the 12-bit A/D converter which is standard on most micro-controllers is more than sufficient), a micro-controller can read the 16-bit frequency word and convert all of this information to a MIDI note on (with optional pitch bend) and note off command.

Because there are so many use cases for audio to MIDI, the results should be rather heard first hand in a real-time performance situation than presented in graphical form. The author is presenting a live demonstration at the 2019 AES Convention venue and invites interested parties to make contact for demonstration opportunities.

11 Conclusions

The audio-to-CV/MIDI chip is demonstrated in a prototype PCB designed for this purpose. It is expected that this chip will find use in many sound processing applications such as the conversion of audio to music notation, digital and analog synthesizers, music education, experimental performance situations, and perhaps others which the author hasn't yet imagined.

The journey to track the pitch of monophonic audio sources really began with the development of the vocoder during the 1930s (in the case of human speech as the audio source). Throughout the 20th Century, improvements such as closed-loop feedback were introduced, which eventually found their way into synthesizer products such as the Korg MS-20 and Roland SPV-355 and GR-300 in the late 1970s.

It is hoped that the developments illustrated in this paper will enable a dramatic leap in performance for frequency-tracking synthesizers which should satisfy even the most demanding instrumentalists searching for an audio to synthesizer solution — whether analog or digital. Integrating all of these functions into one chip should also make these types of synthesizers more readily available and certainly more economical.

References

[1] Keio Electronic Laboratory Corporation, *MS-20 Monophonic Synthesizer Service Manual* (1978).

[2] McKinney, N.P. "Laryngeal frequency analysis for linguistic research" (1965).

[3] Roland Corporation, *SPV-355 Service Notes* (1980).

[4] Roland Corporation, *GR-300, G-303/G-808 Service Notes* (1980).

[5] Riesz, R.R., "Apparatus for determining pitch frequency in a complex wave," *U.S. Patent No. 2,593,698* (filed 1948).

[6] Niedźwiecki, A. and Mikiel, W. „Digital measurement of the fundamental frequency in a harmonic speech signal," *Speech analysis and synthesis* (1976).

[7] Edson, J.O. and Feldman, C.B.H. "Derivation of vocoder pitch signals," *U.S. Patent No. 2,906,955* (filed 1956).

[8] De Cheveigné, A. and Kawahara, H., "YIN, a fundamental frequency estimator for speech and music," *The Journal of the Acoustical Society of America* (2002).

[9] Derrien, O., "A Very Low Latency Pitch Tracker for Audio to MIDI Conversion," *Proceedings of the 17th Int. Conference on Digital Audio Effects (DAFx-14)* (2014).

[10] EBS Sweden AB, *Assembly Specification EBS Black Label OctaBass Pedal* (2002).

[11] Angus, J., Garner, P. and Howard, D. "Fundamental frequency estimation for use with a singing pitching development system for primary school children," *Proceedings of the Institute of Acoustics* (1998).

[12] Conversation with Herman Gillis, CEO of Sherman Productions BVBA (2018).

[13] Gardner, F.M., "Charge-Pump Phase-lock Loops," *IEEE Transactions on Communications* (1980).

[14] Kaczynski, B., "Fast-locking frequency synthesizer," *U.S. Patent No. 9,685,964* (filed 2015).