



The valve DAC

Submicron silicon meets submillimetre vacuum

Marcel van de Gevel

1 Introduction

Despite the increasing popularity of gramophone records [1], many recordings are still only distributed digitally, either on a digital carrier or as a computer file or by a streaming service. This must be frustrating to those audio enthusiasts who prefer valve electronics over transistorized circuitry and would ideally want to remove all transistor stages from their audio equipment, because most so-called valve DACs are actually solid-state DACs followed by a valve buffer.

Although I believe that excellent audio equipment can be made using solid-state technology, I considered it a nice intellectual challenge to see how far I could get with building a DAC that has all critical analogue and mixed-signal functions realized in valve technology. I also saw it as a nice excuse to experiment with valve oscillators and switching circuits, rather than the usual amplifiers. This article summarizes my findings.

1.1 Definitions

Some terms in electronics have acquired many slightly different or even completely contradictory meanings over the years. A good example is the word linear; according to the strict definition inherited from linear algebra, a linear voltage regulator is one that doesn't work, because its output is proportional to its input. The term stability is well-defined for linear systems, but has a somewhat different meaning for sigma-delta modulators. A very peculiar one is the word computer, which used to be a human being doing computations, but gradually changed into a programmable machine, basically a practical implementation of Turing's universal machine.

See **Table 1** for some definitions used in this article. The variable t represents time; for discrete-time systems, t is an integer multiple of the sampling time T .



Term	Definition used in this article
<p>Linear</p>	<p>Suppose a system or a circuit produces an output signal $y_1(t)$ for an input signal $x_1(t)$, while its output signal is $y_2(t)$ when its input signal is $x_2(t)$. The system or circuit is linear when its output signal is $\alpha \cdot y_1(t) + \beta \cdot y_2(t)$ in response to an input signal $\alpha \cdot x_1(t) + \beta \cdot x_2(t)$, for any α and β.</p> <p>As all real-life circuits clip, fold or produce smoke above a certain signal level, no real-life circuit can be entirely linear, but they can be approximately linear over a limited range of signal levels.</p> <p>A phase versus frequency characteristic is called linear when the phase shift is proportional to frequency. This corresponds to a frequency-independent delay.</p> <p>Linear interpolation between two points means interpolation according to a straight line segment between these points.</p>
<p>Affine (as in affine transformation)</p>	<p>Suppose a system or a circuit produces an output signal $y_1(t) + y_7(t)$ for an input signal $x_1(t)$, while its output signal is $y_2(t) + y_7(t)$ when its input signal is $x_2(t)$. The system or circuit is affine when its output signal is $\alpha \cdot y_1(t) + \beta \cdot y_2(t) + y_7(t)$ in response to an input signal $\alpha \cdot x_1(t) + \beta \cdot x_2(t)$, for any α and β. This is often a useful abstraction for circuits with noise, offset or non-zero initial conditions.</p>
<p>Stable</p>	<p>A linear or affine system is stable when any bounded input signal of finite duration produces a bounded output signal.</p> <p>A system that is linear or affine up to a certain clipping level will be called stable when a sufficiently small input signal of finite duration does not make it clip.</p> <p>A chaotic sigma-delta modulator with linear loop filter will be called stable when its state variables remain bounded.</p> <p>A non-chaotic sigma-delta modulator and any sigma-delta modulator of which the loop filter can clip will be called stable when it produces granular cycles and unstable when it produces overload cycles.</p> <p>A limit cycle or limit set will be called stable when it is attracting, meaning that the system converges to the limit cycle or limit set when it is started from an initial condition that is close to the limit cycle or limit set. It is unstable when it is repelling, meaning that when the system is started from an initial condition that is close to but not on the limit cycle or limit set, the distance to the limit cycle or limit set will grow with time.</p>



Granular cycle	A sigma-delta modulator is said to produce granular cycles when it works well: its output switches frequently, its internal state variables remain relatively small and its output signal spectrum is a good approximation of its input signal spectrum over the band of interest.
Overload cycle	A sigma-delta modulator is said to produce overload cycles when it produces low-frequency oscillations, its internal state variables become larger than for granular cycles, and its output signal spectrum does not approximate the input signal well.
Time invariant	Suppose a system or a circuit produces an output signal $y(t)$ for an input signal $x(t)$. If it is time invariant, shifting its input signal in time will result in a time shifted, but otherwise equal output signal. That is, the response to $x(t - \tau)$ will be $y(t - \tau)$ for any τ .
Time variant	A time-variant system or circuit is one that is not time invariant. Typical examples are samplers, decimators and frequency converters (mixer plus local oscillator).
Linear time invariant	A linear time-invariant system is, obviously, one that is both linear and time invariant. These systems have the special property that their output signals can only contain frequency components that are also there at their inputs. They can be characterized completely by measuring the steady-state magnitude and phase responses to sine waves for all frequencies or, alternatively, by measuring their impulse or step response.
Microcontroller	Single-chip computer intended for embedded applications
Computer	Practical implementation of Turing's universal computing machine, the infinite paper tape being replaced by a more practical type of memory
Human computer	Pre-1948 meaning of the word computer: human clerk doing calculations (for whom Turing's computing machines were an abstract model)

Table 1 Definitions



2 General set-up of an audio DAC

Figure 1 shows the set-up of a typical audio DAC. It consists of some sort of digital interface, digital pre-processing, the actual digital to analogue converter, a reconstruction filter, a time reference and a voltage (or current) reference.

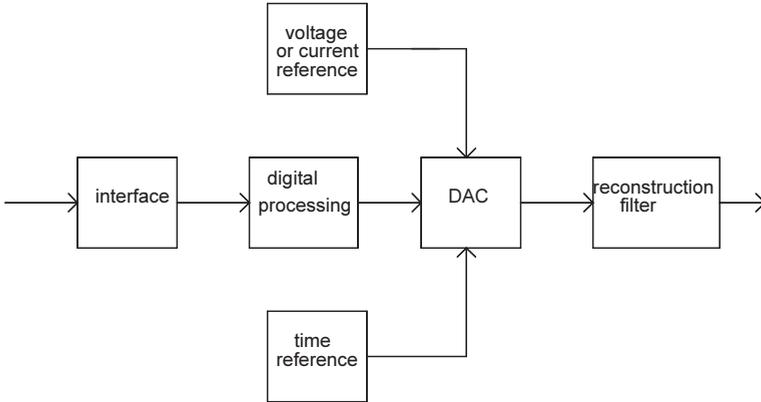


Figure 1 Audio DAC block diagram, only one channel shown.

Digital signal processing is essentially just a bunch of calculations that need to be done. As long as the correct calculations are done and as long as they are done correctly, the result is independent of whether the number crunching is done by an FPGA made of deep submicron CMOS transistors, by a valve computer or by a team of human computers. It would, however, be very hard for normal thermionic valves or for human computers to do the required calculations in real time. Hence, the most practical solution by far is to use deep-submicron CMOS.

The actual digital to analogue conversion, reconstruction filtering, time reference and voltage reference all have a direct impact on the sound quality. This is obvious for the actual digital to analogue conversion and reconstruction filtering. Variations in the voltage reference and time reference modulate the output signal in amplitude and phase, respectively, thereby creating undesired sidebands. If noise shaping is used, fast undesired variations in the voltage and time reference (noise and jitter) frequency-convert out-of-band quantization noise into the audio band.

The actual digital to analogue conversion, reconstruction filtering, time reference and voltage reference must therefore be regarded as critical analogue and mixed-signal functions. Because of the reasons explained in section 1, they must be realized in valve technology.



3 The actual digital to analogue converter

3.1 Reducing DAC requirements by sigma-delta modulation

Digital to analogue converters can roughly be categorized as multibit and single-bit DACs. Accurate multibit DACs are difficult to realize in any technology. Each 1 LSB step must be equally large, but as different components determine the sizes of the steps, that implies that there are tough matching requirements.

Fortunately, noise shaping techniques make it possible to get a good dynamic range out of a DAC with only a few bits or even just a single bit. Single-bit DACs have only one step, which more or less automatically makes the step match itself. There are some second-order effects to take into account, though. For example, when the 0-to-1 and 1-to-0 transition times are different, a 1100 sequence may have a different average value than a 1010 sequence, see **Figure 2**. A very effective way to eliminate this problem is to let the DAC switch back to zero after each bit, essentially changing the sequences to 10100000 and 10001000, respectively. This is known as a return-to-zero DAC.

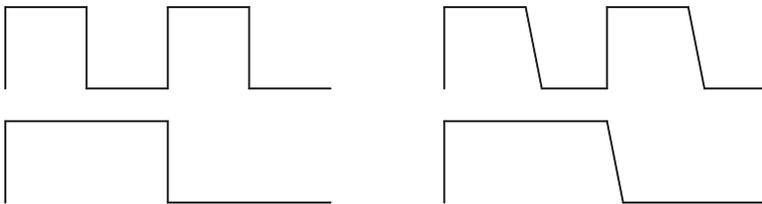


Figure 2 Impact of unequal rise and fall times on a non-return-to-zero DAC. On the left, the ideal waveforms are shown for a 1010 and a 1100 sequence. On the right, a nonzero fall time is taken into account. It is clear that the average value of the 1010 sequence is now greater than the average value of the 1100 sequence due to the different number of falling edges.

A disadvantage of return-to-zero DACs is their greater sensitivity to clock jitter. Essentially, every extra transition of the output is an extra chance for clock jitter to produce noise in the analogue output signal. When the DAC output stays constant, slight variations in timing have no impact on the output signal. Only when the DAC output switches to a different voltage level, slight perturbations of the switching moment translate into perturbations of the output signal. As will be explained later, the jitter sensitivity can be reduced again by making a so-called FIR DAC [2].

A noise shaping loop is traditionally called a noise shaper when it is drawn as an error correction loop and called a sigma-delta modulator or delta-sigma modulator when it is drawn as an ordinary negative feedback loop¹, even though these are essentially equivalent. A noise shaper / sigma-delta modulator with single-bit quantizer converts its input signal into a single-bit output signal with much higher sample rate. This single-bit signal can then be converted to analogue by a single-bit DAC, see **Figure 3**.

¹ I have the impression that the terms sigma-delta modulator and delta-sigma modulator used to be reserved for single-bit systems, until the introduction of the term multi-bit sigma-delta modulator.

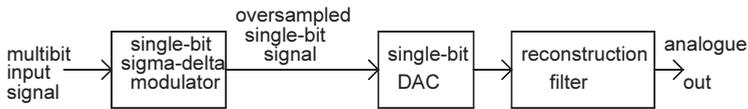


Figure 3 Reducing DAC resolution requirements with a sigma-delta modulator.

The upper part of **Figure 4** shows a simplified block diagram of a sigma-delta modulator, which is simply a negative feedback loop around a coarse quantizer. The loop filter has a large gain inside the band of interest, but its gain rolls off quickly outside this band. Hence, the large rounding error produced by the quantizer is reduced in the band of interest by the negative feedback loop. Outside the band of interest, the error actually gets worse.

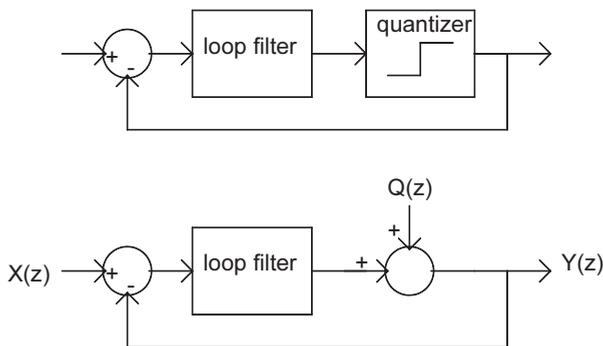


Figure 4 Sigma-delta modulator and its linearized model.

Even though sigma-delta modulators are always nonlinear, they are often designed using a linear model augmented with a rule of thumb. The quantizer is simply replaced with an addition point that adds quantization noise, as shown in the lower part of Figure 4. The transfer from the quantization noise $Q(z)$ to the sigma-delta modulator output signal $Y(z)$ is commonly known as the noise transfer function $NTF(z)$. The feedback loop has a delay, so the impulse response of $NTF(z)$ always starts with a 1. The transfer from the signal input $X(z)$ to $Y(z)$ is called the signal transfer function $STF(z)$.

The obtainable amount of reduction of the quantization error is limited by the Gerzon-Craven noise shaping theorem [3]. Gerzon and Craven have shown that when you plot the magnitude of the noise transfer function on a linear frequency scale (from 0 up to half the sample rate) and a vertical dB scale, the area below the 0 dB line can never be greater than the area above the 0 dB line. For a sigma-delta with stable loop filter, the areas are equal, for a sigma-delta with unstable loop filter, the area below the 0 dB line is smaller than the area above it. This is illustrated in **Figure 5**.



Figure 5 Sketch of a noise transfer function, magnitude in dB versus frequency on a linear scale. According to the Gerzon-Craven noise shaping theorem, the area below the 0 dB line can not be greater than the area above the 0 dB line.

Hence, assuming a stable loop filter, there are three ways to increase the suppression of in-band quantization noise. The first is to increase the sample rate, that is, to choose a high oversampling ratio. This increases the area above the 0 dB line by increasing the number of Hz. The second is to increase the noise gain outside the band of interest. This increases the area above the 0 dB line by increasing the number of dB. The third is to use a very steep filter characteristic between the frequency band where the noise is suppressed and the frequency band where the noise is increased. This ensures that the noise suppression is not wasted on the transition band.

The first method requires fast electronics, much faster than the highest signal frequency. This is a rather normal requirement for any system based on feedback. It turns out that the second method reduces the stability of the sigma-delta modulator; a well-known rule-of-thumb says that the out-of-band noise gain for single-bit sigma-deltas of third or higher order should be about 1.5 times, or 3.52 dB. This is based on Fig. 7 of reference [4]. At a noise gain of 1, no noise shaping is possible because there is no area above the 0 dB line, and at a noise gain of 2, the loop becomes unstable for any input signal level. The factor of 1.5 is a compromise between these extremes. Higher noise gains are possible when more quantization levels are available. The third method means using a high-order loop filter.

Quantization is inherently a non-linear process, and even worse, it is a non-linear process that produces dead zones: variations of the input signal between two decision thresholds have no effect on the quantizer output. Negative feedback can not deal well with dead zones, because the loop gain drops to zero when you are in a dead zone. In a sigma-delta modulator, this typically results in the occurrence of distortion products and idle tones that depend on the signal or offset at the sigma-



delta modulator input. The sigma-delta starts switching between different codes to get the correct average value, and when this is done regularly, it results in a tone.

For those who prefer anti-causal analysis of feedback systems [5]: when you want a quantizer that can only produce integer numbers to produce a fractional output, no amount of predistortion of its input signal will accomplish this. You can only switch between values such that the average is OK, but when you do this in a regular pattern, you get undesired tones.

A way to get rid of the most annoying artefacts of quantization is dithering [6], [7], [8], [9], adding a small noise signal at the input of the quantizer². The total error of a dithered quantizer has a mathematical expectation (ensemble average) and a standard deviation that are independent of the signal when the dither has the following characteristics [6]:

- A. The dither must have a two-LSB peak-to-peak value and a triangular probability density function³;
- B. It must be independent of the signal to be quantized;

For a quantizer that is used inside a feedback loop, requirement B implies that:

- C. The dither samples must be independent of each other.

There are some special cases where requirement C does not apply, such as loops with integer coefficients. See references [8] and [9] for details.

With dither according to the requirements A, B and C, the power spectral density of the rounding error becomes white and independent of the signal. When the quantizer is embedded in a noise shaping loop, the power spectral density at the noise shaper output is shaped according to the noise transfer function, but it is still continuous, without any tones, and independent of the signal. This makes the round-off error sound like (coloured) background noise rather than like a very unpleasant kind of distortion.

The efficacy of dither has been grossly over- and understated in some literature, ranging from 'dither makes quantization essentially linear' to 'dither only works on steady-state signals such as sine waves and not on musical transients'. A non-subtractively dithered quantizer is neither linear nor affine, because the quantization error is statistically dependent on the signal. For example, when the input to the quantizer is 12.25 LSB, for any kind of non-subtractive dither, the quantization error will always

² Scientists believe that dither-like phenomena play a role in processes as diverse as the periodicity of ice ages and the operation of the mammalian auditory system, the inner hair cells being dithered by the Brownian motion of the fluid in the cochlea. See reference [10] for more information.

³ At least this is the option with least noise; in general you have to add two or more random signals with uniform distribution of 1 LSB peak-peak each. Adding precisely two of these signals results in triangular dither.



be $n - 0.25$ LSB with integer n . When the input to the quantizer is 13.31 LSB, the quantization error will be $n - 0.31$ LSB with integer n . Hence, the probability distribution of the quantization error is always dependent on the input signal, even though the average and the standard deviation can be made independent of it.

Claiming that dither only works on steady-state signals is equally incorrect. This impression may have been left by some early articles on dither that for simplicity showed the effect on sine waves or on constant input signals; the later, more general, articles are often difficult to read because of their advanced mathematical content (such as large numbers of high- or even infinite-dimensional Fourier transforms). In any case, it is perfectly possible to calculate the effect of dithering over an ensemble of independently dithered quantizers that all quantize the same musical transient.

Unfortunately, proper dithering is not possible in a single-bit sigma-delta because there are not enough quantization levels available. Practically, there are four solutions to this issue:

A Improper dithering

In practice, single-bit sigma-deltas are often dithered with a dither signal having only a few levels, each level being equally probable, and with an experimentally determined peak-to-peak value [11]. Even though both the probability density function and the peak-to-peak value are not according to theory, this is usually sufficient to eliminate all idle tones and strongly suppress any distortion products without the problem of overdriving the quantizer.

B Chaos

Periodic limit cycles produce periodic output sequences that produce spectral lines, that is, pure tones. In a chaotic attractor, there are many limit cycles, but they are all unstable. Hence, except when they are started exactly at a point of a limit cycle while there is no disturbance at all, chaotic systems do not produce periodic output sequences and therefore, they tend to produce continuous spectra rather than pure tones⁴. A sigma-delta modulator that is made chaotic to get rid of tones should be sufficiently chaotic, though (sufficiently large Lyapunov exponent); if it spends too much time near each unstable limit cycle, it may have narrow bumps in its noise spectrum that still sound like tones [12].

A sigma-delta modulator can easily be made chaotic by using an unstable loop filter. Suppose you have two identical stable sigma-delta modulators with unstable loop filters, almost identical initial states and identical input signals. The input signals to the quantizers will then initially be almost equal and the quantizers will therefore initially produce identical output signals. Because of the unstable loop filters, however, the slight difference between the initial states will grow exponentially until it is big enough to make the quantizers take different decisions. From that moment onwards, the sigma-delta modulator output signals will look quite different. This is typical of chaotic systems:

⁴ Strictly speaking, an autonomous digital system can never be completely chaotic because it only has a finite memory (finite state space), so the output sequence will have to repeat eventually. Practically, this is no problem as long as the repetitions happen so infrequently that no-one hears the difference.



slight differences in initial conditions produce a very different long-term response [13], [14].

Just like dithering, chaos reduces the obtainable signal-to-noise ratio. In the case of dithering, the dither noise just adds to the quantization noise. In the case of chaos, the condition of the Gerzon-Craven noise shaping theorem for getting equal areas above and below the 0 dB-line is not met.

C DAC with a few bits resolution and mismatch shaping

Proper dithering of the quantizer is no problem when it has enough quantization levels, but that implies using a DAC with more than two levels, so special measures are needed to prevent draconic requirements on matching in the DAC. Solid-state implementations often have a DAC consisting of $2^n + 1$ unit cells, n being some small integer, and a logic circuit that ensures that the errors caused by mismatch between the unit cells mainly end up outside the band of interest [15]. I consider this technique far too complicated for my valve DAC.

D Sigma-delta with embedded pulse-width modulator

A simpler alternative is to embed a pulse width modulator inside a sigma-delta loop (**Figure 6**). The number of quantizing levels of the quantizer is increased to make proper dithering possible and the pulse width modulator converts the signal from the quantizer back into a two-level signal, so you can still use a single-bit DAC. The disadvantage is that for a given clock rate, you have a smaller oversampling ratio and, hence, less effective noise shaping.

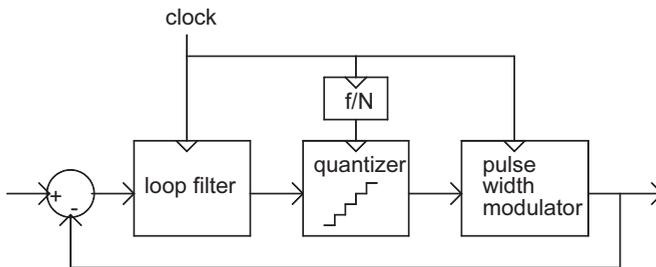


Figure 6 Sigma-delta with embedded pulse-width modulator. The quantizer has more levels, which makes proper dithering possible, but it has to run on a reduced clock rate because the pulse width modulator needs several clock cycles to convert each word coming out of the quantizer. Hence, for a given clock rate, the oversampling ratio is reduced.

Pulse width modulation is by itself a not entirely linear (nor affine) process, see **Figure 7** [16]. Fortunately, it is a nonlinearity that produces only a very small error in the audio band and it is largely suppressed by the sigma-delta feedback loop (provided that the loop filter runs on the PWM clock rate rather than the quantizer sample rate).

Besides, the error can be randomized by rotating the PWM pattern, as shown in **Figure 8**. Here, the pulse width modulator gets eight clock cycles for each quantizer output word. When it has to con-

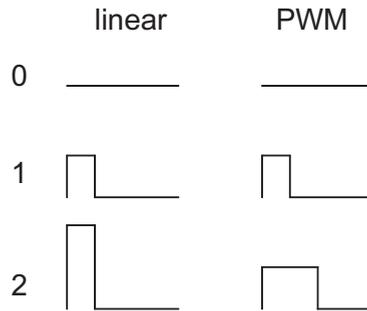


Figure 7 Illustration of the nonlinearity of pulse width modulation. When the patterns for 0 and 1 are as shown in the first two rows, the pattern for 2 should be as shown on the left in the third row, but what you get is shown on the right in the third row [16]. The average value and, hence, the low-frequency content is OK, but it is clearly not the correct waveform.

vert the number three, it randomly picks one out of the eight shown patterns. When each pattern is chosen with equal probability, the mathematical expectation of the output signal is $3/8$ in each PWM clock cycle. This mathematical expectation is linearly related to the converted number; for example, the mathematical expectation of the output signal would be $5/8$ when a five is converted in a simi-

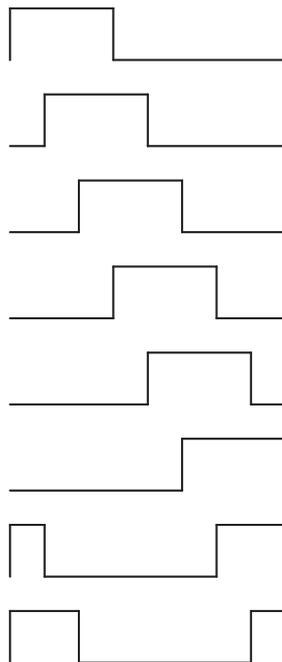


Figure 8 Eight rotated PWM patterns.



lar fashion. Hence, the only error the PWM still makes is a signal-dependent random variation on top of the desired value. This means that the already rather benign PWM distortion is changed into just some signal-dependent noise.

It is easy to see that the noise of the output signal indeed depends on the input code. An input code of zero or eight would always result in the same output pattern, producing no noise, while any other input code does produce some noise. This signal-dependent noise still gets suppressed by the overall sigma-delta feedback loop.

The digital sigma-delta modulator in my DAC can be switched between three modes: single bit with chaos, five-level randomly rotated PWM and nine-level randomly rotated PWM. A knob allows the user to choose what mode is used.

3.2 The DAC circuit

The simplest circuit that can be used as a single-bit return-to-zero DAC is a logic gate with well-defined high and low levels. The logic gate should block the data signal while the data signal is switching. This blocking action results in return-to-zero behaviour. The exact timing of the output signal depends only on the clock and the logic gate, so the circuit that supplies the data need not be made in valve technology.

A switched-capacitor DAC would be less sensitive to jitter, but it would need quite a number of valves because of the required switches. In reference [17], as many as six vacuum diodes were needed for a single switch of a sample-and-hold circuit. A switched-capacitor DAC usually uses six switches, although it might be possible to reduce this by going for a single-ended circuit rather than the usual balanced version. In any case, I haven't pursued this option any further.

Using sigma-delta modulation, speed is traded for accuracy, so the logic gate used as a DAC must be relatively fast. Current steering logic is known to be particularly fast due to the low impedance levels and small signal swings that are used [18]. Hence, we end up with the circuit of **Figure 9**.

The bottom differential pair U_{3A} - U_{3B} is driven by the clock signal and switches a reference current between the upper two differential pairs U_{1A} - U_{1B} and U_{2A} - U_{2B} . Depending on the data signal, the upper differential pairs steer the current to either the positive or the negative output. The grids of the upper differential pairs are switched while they have no tail current, hence the exact timing of the data signal ideally has no impact on the output signal. Together these three differential pairs form a two-tap RTZ FIRDAC [2].

C28 and C29 are the first capacitors of the reconstruction filter, the rest of which is not shown. They keep the impedance at the anodes low for high frequencies. In order to get output voltages referred to ground, the anode resistors are tied to ground and a negative supply voltage is connected to the



tail resistor. Capacitive coupling is used for level shifting, with a Schottky diode clamp for the data signal because its DC level is signal dependent. The voltage reference is referred to the -300 V rail.

In general, a FIRDAC consists of a series of digital-to-analogue converters driven by delayed versions of the same signal. The output signal of the FIRDAC is a (possibly weighted) sum of the output signals of the individual DACs. Hence, the whole thing acts as a mixed-signal FIR filter that has its tapped delay line implemented in a digital fashion and the (possibly weighted) summation implemented in an analogue fashion.

In the case of Figure 9 the two upper differential pairs each act as a return-to-zero DAC. Their combined output signal is similar to the output signal of a single non-return-to-zero DAC, but without the problems that unequal rise and fall times cause in a non-return-to-zero DAC. Hence, we combine the distortion performance of a return-to-zero DAC with the jitter performance of a non-return-to-zero DAC.

4 Voltage reference

Because of its good availability (at least there were plenty of them available at a fair of the NVHR, the Dutch association for the history of the radio) an 85A2 glow discharge reference tube was chosen as the voltage reference. Glow discharge voltage reference tubes have some peculiarities [19]:

A Ignition

Glow discharge lamps always need a higher voltage to start the glow discharge than to maintain the glow discharge. For an 85A2, the maximum ignition voltage is 115 V for a fresh tube, 120 V for an aged one, while the maintaining voltage is $85 \text{ V} \pm 2 \text{ V}$ at the recommended current of 5.5 mA [20]. Even when a voltage exceeding the maximum ignition voltage is applied, it takes some time before the tube ignites. This time depends on the availability of ionized atoms and free electrons in the gas filling, see section 4.1.

B Jump voltage

When the current is swept over the allowed current range, a step may occur in the maintaining voltage. As this is undesired, manufacturers try to keep the step small and try to ensure it doesn't occur at the recommended current level.

C Impedance

Glow discharge tubes have a large and frequency-dependent inductance [21], [22]. This is related to transit time effects of the ionized gas atoms.

D Noise

Like all voltage references, glow discharge voltage reference tubes generate noise. According to reference [22], the open-circuit noise voltage at frequencies that are high compared to the reciprocal



of the ion transit time can be calculated as full shot noise flowing through a resistance equal to the DC voltage across the tube divided by the DC current:

$$S(v_n) = 2qI \left(\frac{V}{I} \right)^2 = \frac{2qV^2}{I}$$

With $q \approx 1.6022 \cdot 10^{-19}$ C, $V \approx 85$ V and $I \approx 5.5$ mA, this boils down to $4.2094 \cdot 10^{-13}$ V²/Hz, or 648.8 nV/ \sqrt Hz. This is not bad at all when you look at the ratio of the noise to the DC reference voltage.

The very same equation applies at lower frequencies, down to about 400 Hz. At these frequencies, avalanche multiplication increases the noise current, but also reduces the impedance. As a result, the noise voltage density stays roughly constant. Below about 400 Hz, non-uniformity of the sputtered cathode surface causes a slight increase of the noise. At very high frequencies, the anode-to-cathode capacitance filters off some of the noise.

4.1 Ignition delay for four brands of 85A2

As the available data on ignition delays was somewhat vague, I bought four 85A2 tubes from different brands and measured the delay myself. The ignition delay consists of the so-called statistical delay and the formative delay [19].

A glow discharge can only build up when there are ionized atoms and free electrons available and when they get accelerated enough to cause an avalanche effect. The statistical delay is essentially the time it takes until this happens, which can vary between ignitions of the same tube. The formative delay is the time it takes until the avalanche has grown to the required current level. This delay is relatively short and quite reproducible.

When the statistical delay dominates, the chance that the tube has not ignited after a time t is $\exp(-t/\tau)$, where τ is the average statistical delay. The tube manufacturer can reduce τ by adding radioactive material such as uranium dioxide or tritium. (Mind you, tritium has a half-life of 12.32 years, so a tube produced in, for example, 1957 only has 3.42 % of the tritium left in 2017. The average statistical delay can then be 29 times the original value.)

Measuring four 85A2 tubes in darkness, I found the results in **Table 2**. The slowest tube is obviously the one from Haltron and the fastest the one from NEC. Presumably the NEC tube is the most radioactive device, while the Haltron device has no radioactive primer or had a radioactive primer with a very short half-life. Looking at the variations between individual measurements, the statistical delay was clearly dominant for all types except the NEC.

The delay of the Haltron tube became somewhat smaller when there was light shining on it, but according to reference [19], this depends on second-order effects such as the sodium content of the glass used in the tube. The 85A2 has molybdenum electrodes. Due to the large work function of molybdenum, only ultraviolet light can directly generate free electrons by photoelectric effect on



the electrodes, and ultraviolet light is largely blocked by the glass envelope. I therefore decided to take the dark values and not count on a reduction of the ignition delay by light.

Brand	Conditions	Number of measurements	Average delay
NEC	Darkness, 170 V, 5 s on, 5 s off	24	25.79 μ s
Pope	Darkness, 170 V, 5 s on, 5 s off	68	358.2 μ s
Philips	Darkness, 170 V, 5 s on, 5 s off	37	1.426 ms
Haltron	Darkness, 170 V, 5 s on, 5 s off	81	1.615 s
Haltron	Light on (CFL) and LED flashlight shining at 5 cm distance, 170 V, 5 s on, 5 s off	20	0.8025 s
Haltron	Darkness, 170 V, 28 s on, 28 s off	28	7.675 s
Haltron	Darkness, 258 V, few seconds on, 30 minutes off	8	3.328 s

Table 2 Measured average ignition delays of 85A2 voltage reference tubes. Although not shown in the table, the delay of the Pope tube gradually reduced during the experiment from a few milliseconds to around 50 μ s.

The statistical delay reduces with increasing voltage, because higher voltages increase the chance that an electron-ion pair gains enough energy to cause an avalanche reaction. Hence, for fast ignition, the voltage at start-up should be made as high as practical.

Because of the statistical nature of the delay, there is no real maximum value. One can only define an acceptable probability of excess and calculate the corresponding delay. Without countermeasures, the reference voltage will rise too far and cause damage when the reference tube ignites too late. This is acceptable when it is not the dominant cause of failure of the complete DAC. According to reference [23], the best failure rate obtainable with ordinary thermionic valves is about 0.5 % in 1000 hours (when you take various precautions I haven't taken, such as slow filament start-up and valve derating). Hence, assuming that after switch-on, the DAC will on average be used for about an hour, a reasonable value for the probability of the reference tube not igniting in time is 1/200 000. The



chances of the circuit getting damaged due to the reference tube igniting too late are then of the same order as the chances of ordinary valve failure. The “maximum” delay is then $\ln(200\,000)$ times the average delay, or 40.62 s at 258 V (see footnote ⁵).

A simple way to ensure that 40.62 seconds of ignition delay of the reference tube do no harm is to put an RC filter with sufficiently large time constant between the reference tube and the parts of the circuit that require the reference voltage. With 258 V of voltage at start-up and an ignition delay of 40.62 seconds, an 82.84 s RC time constant will keep the peak value of the filtered voltage below 100 V. A disadvantage is that it will take minutes for the filtered reference voltage to settle, which will considerably slow down start-up of the DAC and will be quite annoying to the user. Besides, if the filter capacitor is electrolytic, it will need to be a very good low-leakage type.

Another option is to sense whether there is any current flowing through the reference tube and to connect the circuits that need the reference voltage after the reference tube has ignited. This eliminates the need for exceedingly long time constants, results in much faster start-up of the complete DAC and it even prevents damage in the unlikely case that reference tube ignition takes longer than 40.62 seconds. Besides, the voltage across the reference tube before ignition will be larger, which speeds up ignition, see **Figure 10**.

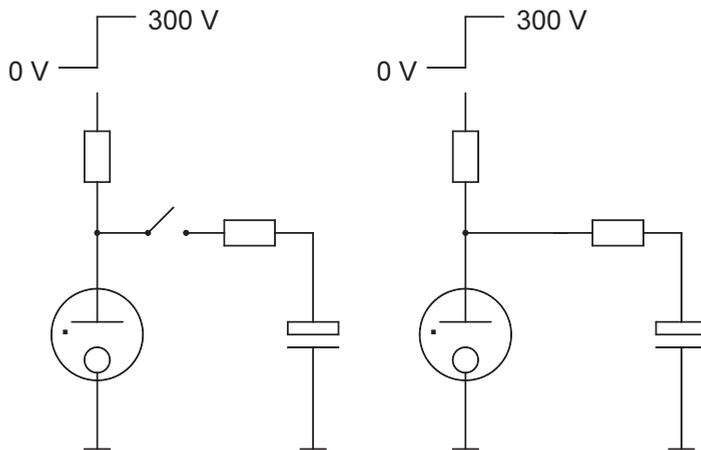


Figure 10 Assume that at start up, the switch is open and the electrolytic capacitors are fully discharged. It is clear from inspection of the circuits that at start up, the full 300 V will appear across the reference tube in the left circuit, while only a part of it will be across the reference tube in the right circuit.

Hence, we arrive at the circuit of **Figure 11**.

⁵ Readers trained in statistics will notice that I’m cheating by taking the average of the measured delays rather than an upper confidence bound. That is, as I have only taken a limited number of measurements, there is a chance that the actual average delay is more than my measured average value.

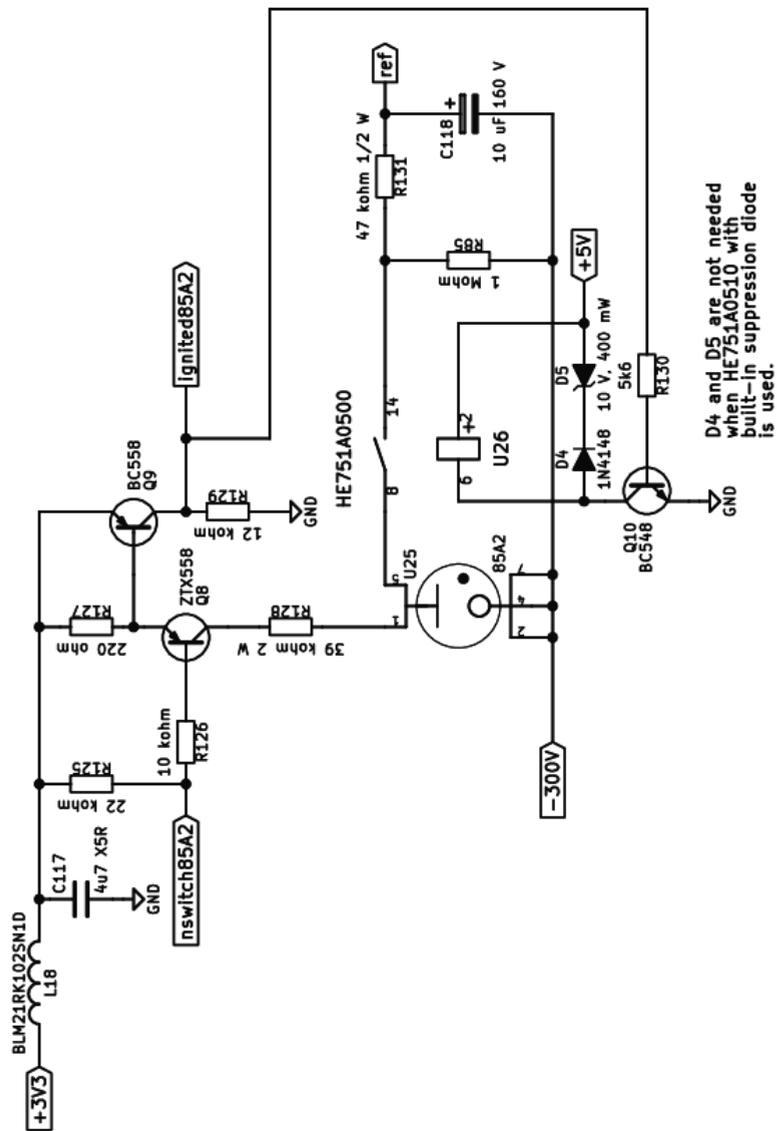


Figure 1.1 The complete voltage reference circuit. Transistor Q_8 is used as a saturated switch that switches on the supply to the 85A2; the use of a transistor seems acceptable here, as its characteristics hardly have any impact on the reference voltage. Q_9 detects whether there is current flowing through the 85A2, that is, whether it has ignited. When this has happened, a high-voltage reed relay HE751A0500 or HE751A0510 is switched on. R_{131} and C_{118} filter off most of the already small reference noise.



5 The time reference

As usual and because of the good phase noise/jitter performance, the time reference is a crystal oscillator.

When the interface to the digital signal source is an SPDIF or AES/EBU interface, the DAC has to synchronize to the bit clock of the signal source while suppressing its jitter. The obvious way to do this is with a PLL with a small bandwidth. For compliance with the grade 2 specifications in the AES11-2003 standard, this requires the crystal oscillator to be tuneable over a range of at least ± 50 ppm around its nominal frequency. For consumer equipment, the IEC 60958-1:2008 standard specifies ± 50 ppm sample rate inaccuracy for “high accuracy” (level 1) equipment, “normal accuracy” (level 2) is specified as ± 1000 ppm.

Tuning valve crystal oscillators can be done with a so-called reactance valve. This is not a special type of valve, but a simple circuit having a capacitance or inductance that depends on the transconductance of the valve and that can therefore be tuned by changing the valve's bias point. My attempts to design a reactance-valve-tuned crystal oscillator with a reasonable tuning range (enough for “high accuracy” consumer equipment) were not successful. Calculations showed that the reactance valve would always have an unpleasantly high loss resistance over some part of its tuning range. Experimental results were even worse; the oscillator didn't start at all due to excessive losses of the reactance valve. It should be possible to mitigate this to some extent by reducing the tuning range of the reactance valve and adding a switchable capacitor bank, using reed relays and fixed capacitors, but it remains a rather inelegant solution, also because the noise of the reactance valve adds jitter.

An alternative that has been available since the 1990's is asynchronous sample rate conversion with a digital sample-frequency-ratio estimator with a small bandwidth [24], [25], [26]. Due to the small bandwidth of the sample-frequency-ratio estimator, the jitter of the input signal is heavily filtered, just like when you use a PLL with a small bandwidth. A disadvantage is the generation of aliasing products in the sample rate converter, but these can be kept arbitrarily small by using a converter with good filters. Using an asynchronous sample rate converter, the crystal oscillator of the DAC need not be tuneable, nor does it need to be accurate.

In a system with PLL, the obvious choice for the crystal frequency is some multiple of all common audio sample rates, such as 28.224 MHz. In a system based on asynchronous sample rate conversion, however, this would be the stupidest possible choice for reasons of interference. As 28.224 MHz is a multiple of all common audio sample rates, the clocks of the source are bound to have harmonics close to 28.224 MHz, but with a frequency difference of a few kilohertz because the clocks are not synchronized. This will result in whistles in the middle of the audible range if both clocks end up in something that acts as a mixer (in the RF sense of the word: non-linear circuit that generates sum and difference frequencies).



The CD sample rate and its multiples will be particularly troublesome, because the ratio of 28.224 MHz to 44.1 kHz is a multiple of 64. An SPDIF or AES/EBU interface typically has spectral peaks at the 32nd and 64th harmonics of the audio sample rate, see figure 3.5 of reference [27]. Similarly, an I²S interface often has a bit clock running at 64 times the sample rate. Relatively low (and therefore strong) harmonics of these frequencies can then cause trouble.

As an example of unintended mixing, a normal single-bit sigma-delta (without offset) has an idle tone at half the sigma-delta clock frequency. With 28.224 MHz sigma-delta clock frequency, if the fifth harmonic of 64 times the CD sample rate somehow leaks into the DAC reference, the DAC will mix them and produce a potentially audible output tone. After all, multiplying its digital input signal with its reference is what a DAC does by definition. Even though the sigma-delta modulators used in this project suppress the idle tone to some extent by dither or chaos, it is best not to ask for trouble and not to make half the sigma-delta clock frequency a multiple of the usual audio sample rates and especially not a multiple of 32 times the usual audio sample rates.

A 27 MHz clock frequency was chosen because multiples of half the clock frequency are conveniently far from all multiples of 32 times the usual audio sample rates, see **Table 3**. The higher audio sample rates such as 88.2 kHz and 192 kHz are multiples of 44.1 kHz or 48 kHz and therefore produce the same or less problems than 44.1 kHz or 48 kHz. Some very high harmonics of the audio sample rates themselves do get close to the clock and to half the clock, but fortunately, they are very high and therefore weak harmonics.

In any case, a disadvantage of the chosen solution with a crystal that has no relation to the common audio sample rates is that all audio signals now have to pass through the asynchronous sample rate converter. There is no option anymore to synchronize the source with the DAC clock, for example using an I²S interface with the DAC as master.

Most 21st century crystal oscillators are some kind of Pierce oscillator, consisting of an inverting transconductance amplifier, two capacitors and a crystal. This topology is, however, not very suitable for valve-based oscillators running above about 10 MHz, because the transconductance of ordinary valves is too low.

Many types of high-frequency valve-based crystal oscillator are discussed in references [28] and [29], but most of them are not directly applicable to 21st century crystals because of their high drive levels (signal power dissipated in the crystal). The maximum drive level of a modern crystal is usually specified as 100 μ W or 500 μ W, while those of the 1950's could easily handle 10 mW. Slightly exceeding the drive level specification just results in a degradation of the frequency accuracy and aging, but exceeding it by orders of magnitude can damage the crystal.



Sample rate f_s	$32 f_s$	Harmonic of $32 f_s$	Multiple of 13.5 MHz	Frequency distance	Harmonic of f_s	Frequency distance
32 kHz	1.024 MHz	13^{th} , 13.312 MHz	13.5 MHz	188 kHz	422^{nd} , 13.504 MHz	4 kHz
		26^{th} , 26.624 MHz	27 MHz	376 kHz	844^{th} , 27.008 MHz	8 kHz
		40^{th} , 40.96 MHz	40.5 MHz	460 kHz	1266^{th} , 40.512 MHz	12 kHz
		53^{rd} , 54.272 MHz	54 MHz	272 kHz	1687^{th} or 1688^{th} , 53.984 or 54.016 MHz	16 kHz
		66^{th} , 67.584 MHz	67.5 MHz	84 kHz	2109^{th} , 67.488 MHz	12 kHz
44.1 kHz	1.4112 MHz	10^{th} , 14.112 MHz	13.5 MHz	612 kHz	306^{th} , 13.4946 MHz	5.4 kHz
		19^{th} , 26.8128 MHz	27 MHz	187.2 kHz	612^{th} , 26.9892 MHz	10.8 kHz
		29^{th} , 40.9248 MHz	40.5 MHz	424.8 kHz	918^{th} , 40.4838 MHz	16.2 kHz
		38^{th} , 53.6256 MHz	54 MHz	374.4 kHz	1224^{th} , 53.9784 MHz	21.6 kHz
		48^{th} , 67.7376 MHz	67.5 MHz	237.6 kHz	1531^{st} , 67.5171 MHz	17.1 kHz
48 kHz	1.536 MHz	9^{th} , 13.824 MHz	13.5 MHz	324 kHz	281^{st} , 13.488 MHz	12 kHz
		18^{th} , 27.648 MHz	27 MHz	648 kHz	562^{nd} or 563^{rd} , 26.976 MHz or 27.024 MHz	24 kHz
		26^{th} , 39.936 MHz	40.5 MHz	564 kHz	844^{th} , 40.512 MHz	12 kHz
		35^{th} , 53.76 MHz	54 MHz	240 kHz	1125^{th} , 54 MHz	0 kHz
		44^{th} , 67.584 MHz	67.5 MHz	84 kHz	1406^{th} , 67.488 MHz	12 kHz

Table 3 Sample rate and bit clock harmonics close to the first five multiples of 13.5 MHz.



A simple and interesting circuit from reference [29] is the tuned-plate, crystal-grid (TPCG) oscillator, see **Figure 12**. It oscillates near the anti-resonance of the crystal, that is, it presents only a very low load capacitance to the crystal. As a result, the frequency is slightly too high, which makes no difference for a DAC with an asynchronous sample rate converter⁶, and the impedance of the crystal is relatively high at the operating frequency, which helps to keep the drive level low.

The valve has three essential functions in the TPCG circuit: it acts as a voltage-controlled current source, its anode-to-grid capacitance couples the anode circuit to the crystal, and it works as an amplitude controller: when the signal gets too large, grid rectification reduces the grid's DC voltage which reduces the transconductance. The LC tank is tuned to have an inductive reactance at the operating frequency. This inductive reactance must be smaller than the absolute value of the reactance of the anode-to-grid capacitance. The advantages of using a tank rather than an inductor are that it prevents unintended oscillations at crystal overtones and it filters off far-off noise of the valve.

The right part of Figure 12 shows a simplified small-signal model that only contains the essential parts and that is only valid near the operating frequency. To the right of the dashed line, there is a controlled current source and a current divider made of an ideal inductor and an ideal capacitor. The ideal inductor is a simplified representation of the LC parallel tank of the complete circuit, which must be inductive at the operating frequency. Capacitor C_{ag} represents the valve's anode-to-grid capacitance and the controlled source g_m represents its transconductance. As long as the reactance X_L of the inductor is smaller than the absolute value of the reactance $X_{C_{ag}}$ of the capacitor, the LC current divider changes the polarity of the current from the controlled source, thereby realizing a negative conductance that can un-damp the crystal. The admittance of everything to the right of the dashed line is:

$$\begin{aligned}
 Y &= g_m \frac{jX_L}{jX_L + jX_{C_{ag}}} + \frac{1}{jX_L + jX_{C_{ag}}} = g_m \frac{X_L}{X_L + X_{C_{ag}}} - \frac{j}{X_L + X_{C_{ag}}} \\
 &= -g_m \frac{X_L}{|X_{C_{ag}}| - X_L} + \frac{j}{|X_{C_{ag}}| - X_L}
 \end{aligned}$$

Hence, the admittance has a negative conductance term and a capacitive susceptance as long as $|X_{C_{ag}}| > X_L$.

⁶ Except that the frequencies calculated in Table 3 will shift when the crystal frequency is somewhat too high. In any case, it helps to prevent audible mixing products between the eight harmonic of 27 MHz and the ninth harmonic of a 24 MHz oscillator on the FPGA module.

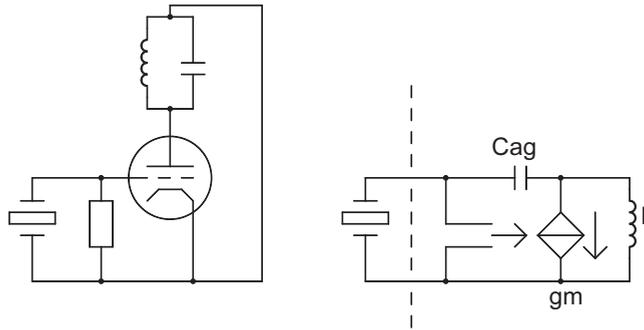


Figure 12 A TPCG oscillator and a simplified model.

Figure 13 shows the crystal oscillator used. It is a balanced version of the TPCG oscillator made with an ECC81. The EF80s are a clock buffer as well as a supply voltage regulator for the actual oscillator. The crystal is a fundamental-mode 27 MHz crystal from Euroquartz in an HC49-4H package. Its specified maximum drive level is 500 μW and its resistance is 40 Ω maximum. I measured its C_0 to be 2.3 pF.

I made a balanced oscillator because it fits in nicely with the DAC, which is also balanced. A disadvantage of making the oscillator balanced is that the voltage across the crystal is twice as large as the voltage on each grid, which again increases the drive level.

6 The reconstruction filter

The reconstruction filter is a simple passive LC filter, using ferrite pot cores with large air gaps and a magnetically very soft ferrite (N48). The advantage of a passive filter is that it can readily handle the steep edges coming out of a DAC. Active filters are less suitable, because sharp edges could drive an active filter employing feedback into slewing or into sub-slewing TIM. It is possible to circumvent this by realizing the first one or two poles passively and the rest actively.

The filter is a 0.05° linear phase filter according to reference [30] with a cut-off frequency of 82 kHz. This cut-off frequency is high enough to reproduce most of the signal frequencies in 192 kHz sample rate material, high enough to cover the human, canine and most of the feline auditory ranges and yet small enough to keep the out-of-band quantization noise down to a reasonable level. Excessive out-of-band quantization noise may cause sub-slewing TIM in audio amplifiers (Figure 14).

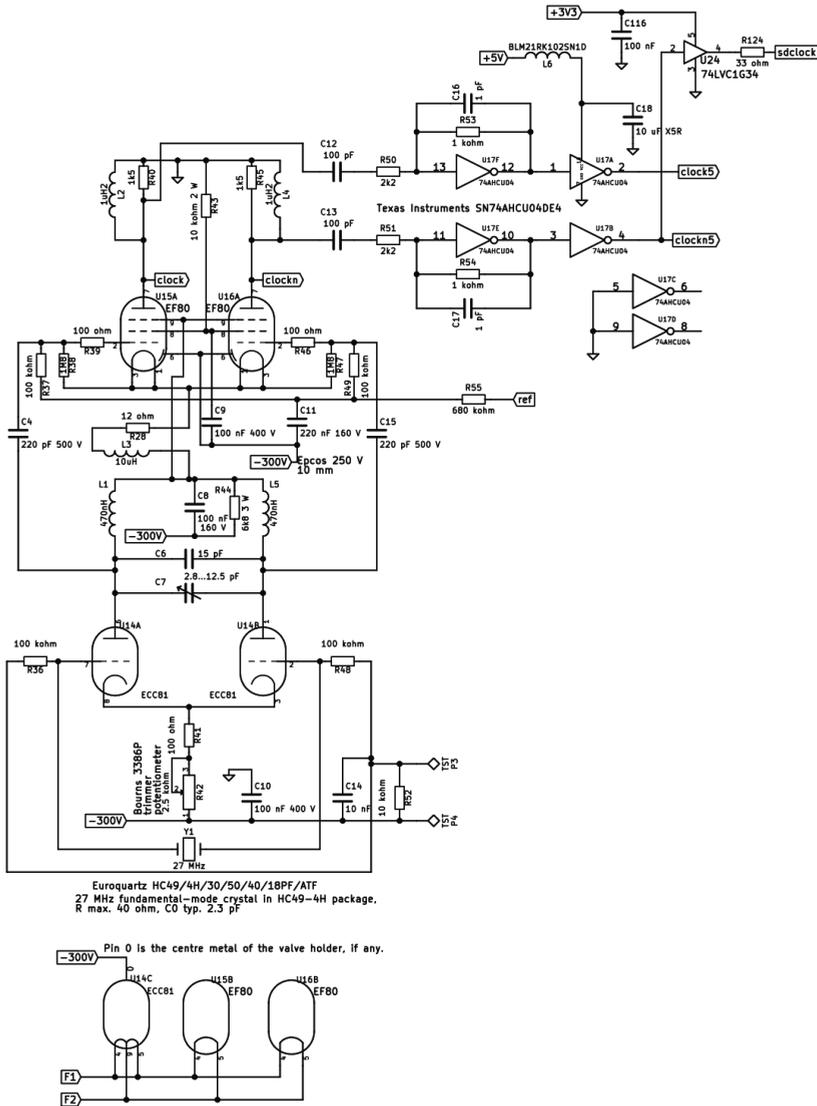


Figure 13 The crystal oscillator

7 Overview of the digital part

An overview of the digital part of the DAC is sketched in **Figure 15**. The signal path is shown at the top and consists of an SPDIF interface, a selectable pre-filter, the asynchronous sample rate converter, an interpolation chain and the actual sigma-delta modulator. (Some blocks are instantiated twice, once for the left and once for the right channel, but this is not shown.) Below the signal chain are some control blocks for initialization and to provide a user interface. The blocks are described in more detail in the next sections.



Filter capacitors: $\pm 1\%$, 160 V, polystyrene
 10 μF DC blockers: $\pm 5\%$, 400 V, polypropylene

Filter inductors:
 56.5, 47.5 and 26 turns, respectively,
 on a 250 nH/turn^2 , N48, P26x16 potcore
 type Siemens/Epcos/TDK B65671-T250-G48
 with no adjusting screw, base plate with clip
 B65675-B5X0 and coil former type
 B65672-B-T1

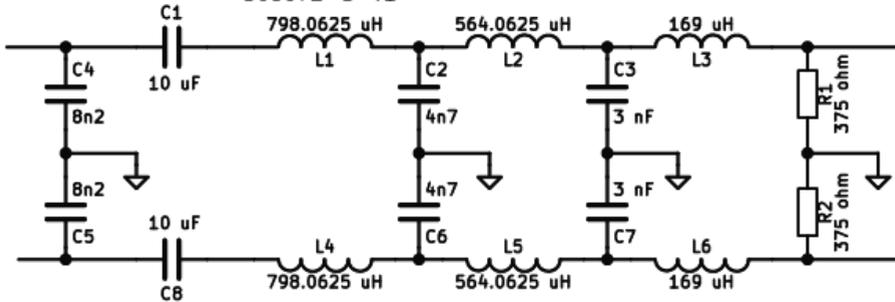


Figure 14 The reconstruction filter, optional output transformer not shown.

The digital signal processing is done with a TE0630 FPGA board from Trenz Electronic, featuring a Xilinx Spartan 6 FPGA called XC6SLX75-3CSG484I, a completely configurable digital chip made in a CMOS technology with 45 nm minimum channel length. The XC6SLX75-3CSG484I is the largest Spartan 6 that is still supported by the free versions of the Xilinx software. Trenz also supplies matching JTAG programming cables, as well as a carrier board that came in handy for the first experiments.

The asynchronous sample rate conversion is done by an SRC4392 from Texas Instruments. As I am not a digital expert, designing my own asynchronous sample rate converter seemed somewhat over-ambitious and besides, judging by its datasheet, the SRC4392 is a very good converter indeed. It does not completely suppress imaging, but this is easily corrected with the pre-filter or the interpolation chain, as explained in section 8. It also has no headroom for overshoots (at least no documented headroom), but that can again be easily fixed. A DIX4192 is used as the SPDIF/AES-EBU interface. Although the SRC4392 contains its own interface, using a separate interface gives us more freedom in inserting filters between the interface and the asynchronous sample rate converter. Judging by their 1.8 V core supply voltage, the DIX4192 and SRC4392 are probably made in 180 nm or 130 nm CMOS.

8 The pre-filters and interpolation filters

8.1 Phase linearity, halfband filtering, apodization

In the early 1980's, Philips successfully used the phase linearity of their CD players as a selling point. Since then, almost all digital filters used for audio sample rate conversions have been long FIR filters

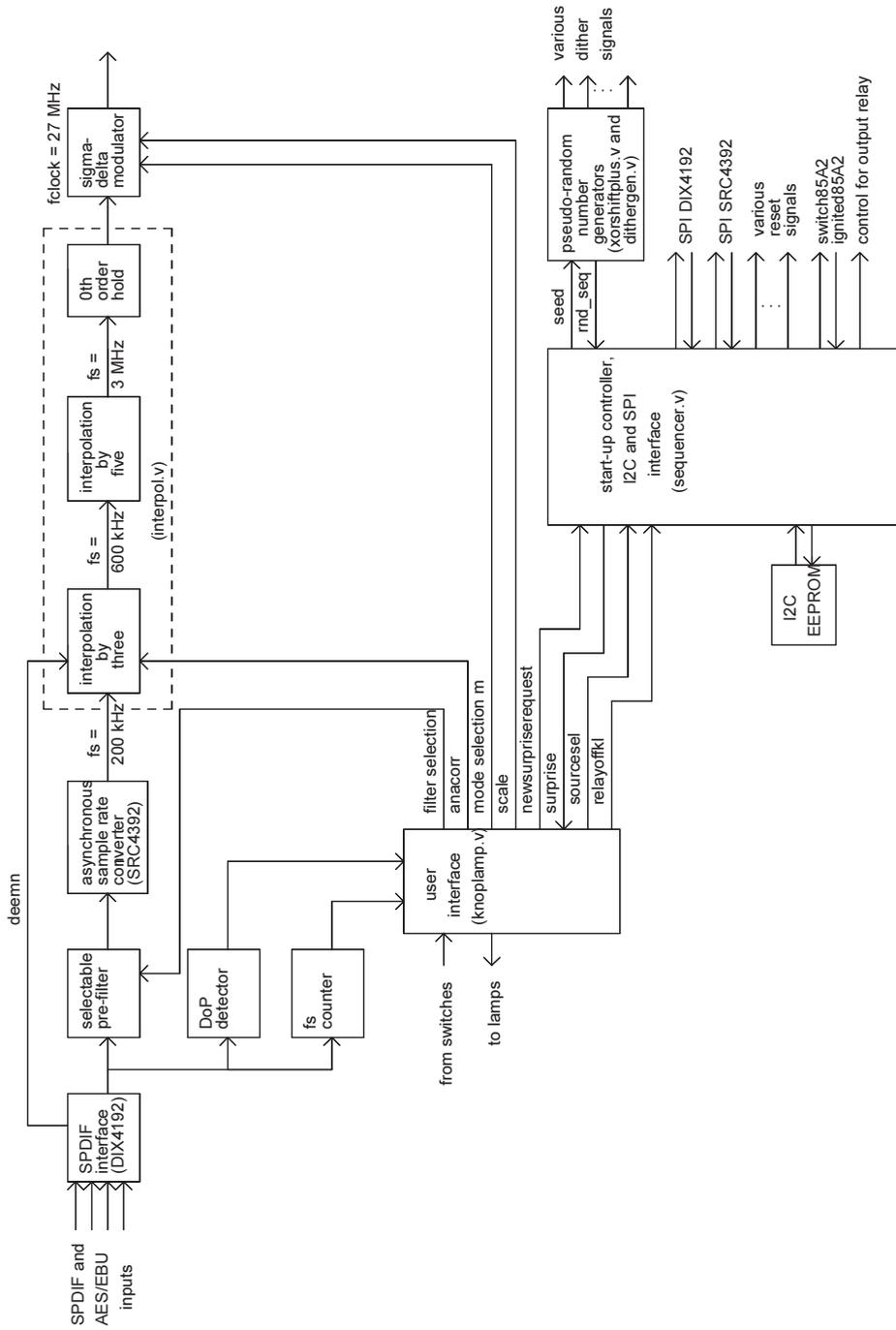


Figure 15 Overview of the digital part.



with linear phase response and quite small passband ripple. It is commonly known that these filters have substantial pre-ringing. Some audio enthusiasts regard this pre-ringing as highly unnatural and likely to cause audible degradation. Others point out [31], [32] that the masking characteristics of human hearing will mask pre-ringing less effectively than post-ringing.

Traditionally, frequencies above 20 kHz are considered inaudible for human listeners. One thing I would like to emphasize is that believing in the audibility of the ringing of an ideal or nearly-ideal 20 kHz low-pass filter implies believing that the frequencies above its 20 kHz cut-off frequency are audible, at least when heard in combination with the rest of the signal. After all, the only difference between the input and the output signal of an ideal low-pass filter is the absence of frequency components above the cut-off frequency from the latter. That is, the cause of the ringing is the absence of frequency components above the cut-off frequency. If you hear the effect of the ringing, you therefore hear a difference between the presence and absence of frequency components above the cut-off frequency.

As Van Maanen pointed out in an earlier edition of *Linear Audio* [33], it is not entirely unthinkable that listening tests made with sine waves are not representative of the audible frequency range with music, because the human auditory system is not linear and time invariant. Experiments with band-limited music have produced very different results depending on the exact methodology. In the 1950's, British researchers from the BBC [34] found that only a few of their most experienced listeners could hear the effect of a sharp low-pass filter at 12 kHz, whereas in the 1990's, Japanese researchers using electroencephalography [35] found evidence that Japanese gamelan players can subliminally hear differences between gamelan music band limited to 26 kHz and not band limited to 26 kHz.

If frequency components above the cut-off frequency are indeed audible when heard in combination with the rest of the signal, then the arguments for using filters with little pre-ringing might be correct. If not, then the only logical choice is a filter that does least harm to the signal components below its cut-off frequency, that is, a filter with linear phase (at least in the passband) and very little passband ripple.

The qualification "very little ripple" requires some explanation. R. Lagadec and T. G. Stockham [36] found out the hard way that a filter that was flat to within ± 0.5 dB from 0 Hz up to the Nyquist frequency was very obviously audible and even annoying to all listeners. The filter had linear phase and it had magnitude peaks with only 25 Hz of distance between one peak and the next. In the time domain, this came down to an impulse response having three peaks, one main peak and a pre- and a post-echo that occurred 40 ms before and after the main peak. These echoes were only 30.8 dB softer than the main peak (see appendix B).



Similarly, a low-pass filter with equidistant passband ripples can be modelled as a cascade of an ideal low-pass filter and a filter with a pre- and a post-echo. Reducing the ripple reduces the pre- and post-echo magnitude, increasing the frequency difference between the peaks reduces the time between the pre-echo, main peak and post-echo, which makes them less audible due to increased (forward and backward) masking.

Hence, filters with smaller ripples and with less ripples are less likely to cause audible problems. If you don't want to rely on masking at all and want to have the echoes down by, say, 120 dB, the ripples have to be as small as ± 0.00001737 dB. The interpolation filter of the SRC4392 has a passband ripple specification of ± 0.007 dB, equivalent to pre- and post-echoes that are 67.9 dB down, clearly requiring some masking to become inaudible. (The SRC4392's decimation filter is switched off in this DAC.)

Many FIR filters used in audio equipment have a transition band from, say, $0.45 f_s$ to $0.55 f_s$ instead of $0.45 f_s$ to $0.5 f_s$, with f_s being the sample rate. They are usually halfband filters, having an attenuation of only a factor of two (6.02 dB) at $0.5 f_s$. These filters have shorter impulse responses than similar filters with a stopband starting at $0.5 f_s$ and almost half their coefficients are zero. They are therefore cheaper and they have shorter ringing, but they do not remove images between $0.5 f_s$ and $0.55 f_s$ (or on the recording side suppress aliasing into the band from $0.45 f_s$ to $0.5 f_s$). The effect of the shorter ringing of a filter with a stopband starting at $0.55 f_s$ can only be audible when frequencies above the cut-off at $0.45 f_s$ are audible. If these are audible, you certainly don't want to allow any images or aliases just above $0.45 f_s$. Hence, filters with a transition band up to $0.55 f_s$ offer only financial advantages and are not preferred for a high-performance DAC.

Peter Craven devised an interesting trick to control the ringing of a complete recording-mastering-reproduction chain [31]. If all filters in the chain are linear-phase filters having a flat response up to $0.45 f_s$ and a transition band from either $0.45 f_s$ to $0.5 f_s$ or from $0.45 f_s$ to $0.55 f_s$, all resulting ringing and aliasing can be suppressed by adding one and only one filter with controlled ringing. This filter needs to have a smaller bandwidth than the other filters, such that $0.45 f_s$ is already suppressed. Craven dubbed this an apodizing filter. It makes no difference where in the chain the apodizing filter is placed, although for practical reasons Peter Craven prefers to have one used during mastering. He makes no claims about the audibility of the effect, he only encourages experimenting with it.

It is difficult to see in the time domain why this helps, but in the frequency domain it is clear. Apart from a constant delay, the phase-linear filters elsewhere in the chain essentially don't change anything up to $0.45 f_s$. Hence, as long as the apodizing filter's stopband begins at a frequency below $0.45 f_s$, the magnitude and phase responses of the entire chain are completely determined by the apodizing filter. The impulse response of the entire chain is simply the inverse Fourier transform of its magnitude and phase responses. Unfortunately, apodization does not suppress the effect of passband ripples, only the ringing due to the transition- and stopbands.



As many digital recordings are made without apodizing filter, it may be useful to have a selectable apodizing filter on an audio DAC. It should be noted that a signal chain with two or more apodizing filters will behave worse than a signal chain having only one such filter, so the apodizing filter should be switched off when playing an already apodized recording.

8.2 Dithering of intermediate results

When the aim is to optimize the sound quality rather than to minimize the amount of chip area spent for a given SINAD value (signal to noise and distortion value), rounding PCM signals must be avoided when possible and must otherwise be done with proper two-LSB peak-to-peak triangular probability density function dither. Unfortunately, commercial digital and mixed-signal audio processing circuits very often round signals with no dithering at all. The resulting distortions, of course, only occur on very-low-level signals and one can argue about how likely they are to be audible on normal programme material, but rather than arguing about that, I prefer to avoid them.

The datasheet of the SRC4392 [25] states that the output signal word length can be reduced if desired and that this is done with triangular probability density dither. This shows that the designers of the SRC4392 have used dither, although it doesn't show that they have done this consistently. At least the output spectrum plots in the SRC4392 datasheet all look very good, also the ones with low-level input signals.

8.3 Interpolation filter overshoot

One thing that is rarely even mentioned in the datasheets of digital audio chips is the impact of overshoots. Stupid as it may be, it is common practice to make commercial digital audio recordings as loud as possible, leaving no headroom at all for overshoots in the digital signal processing. Still, most filters with 0 dB gain for stationary sine waves may produce overshoots on non-stationary or non-sinusoidal waveforms.

Regarding overshoots, the worst possible input signal is one that switches between positive and negative full scale in a $\text{sgn}(h(-n))$ fashion, $h(n)$ being the filter's impulse response. When this signal shifts into the delay line of a FIR filter, there comes a moment when all positive coefficients get multiplied by positive full scale, while all negative coefficients get multiplied by negative full scale. Taking positive full scale to be +1 and negative full scale to be -1, the resulting output signal peak is the sum of the absolute values of the filter coefficients. The DC gain is simply the sum of the filter coefficients. With respect to the DC gain, the overshoot is therefore the ratio of the sum of the absolute values of the coefficients to the normal sum of the coefficients. Something similar applies to continuous-time filters when you replace the summation by an integral.



For an ideal continuous-time filter, the resulting ratio is infinite, as the integral from $-\infty$ to ∞ of the absolute value of a $\sin(x)/x$ -function diverges [37]. For the long FIR filters used in this DAC, the ratio is typically between 10 dB and 11 dB. Two or three cascaded filters are typically only a few tenths of a dB worse than a single filter.

As the digital interpolation filters have plenty of dynamic range, you can easily keep two bits (12.04 dB) of headroom for filter overshoots in the interpolation chain, but doing so in the actual core of the DAC will significantly reduce the signal-to-noise ratio. The worst-case input signal is essentially a square wave at the cut-off frequency that suddenly makes a 180° phase jump, which is quite unusual as a musical waveform; actual music waveforms are likely to give some overshoot, but much less than 10 to 11 dB.

Hence, I've added three modes of operation. In the loud mode, a full-scale DC input drives the sigma-delta modulator to producing roughly three quarters ones and one quarter zeroes, or vice versa for negative full scale. This still leaves a few dB of headroom, as the sigma-delta modulators can be driven slightly above this level before any integrator clipping occurs. A clipping lamp warns the user when integrator clipping does occur. In the intermediate mode, the signal level driving the sigma-delta is halved. In the paranoid or soft mode, it is again halved, ensuring that even the worst-case input signal causes no integrator clipping.

8.4 Complete digital filter chain

This brings us to the filter chain shown in **Figure 16**. Depending on the choice of the user, the first section can be an apodizing filter, a steep filter or no filter at all; I've also added a "surprise" mode for blind listening. Somewhat arbitrarily, I've chosen an asymmetric Wilkinson filter (see reference [38] and appendix A) as apodizing filter. These filters have flat passband response and nearly linear phase over their passband, while still having less pre- than post-ringing. A disadvantage is that they slightly peak at the start of the transition band. See **Figure 17**, **Figure 18**, **Figure 19** and **Figure 20** for the responses of the apodizing filter.

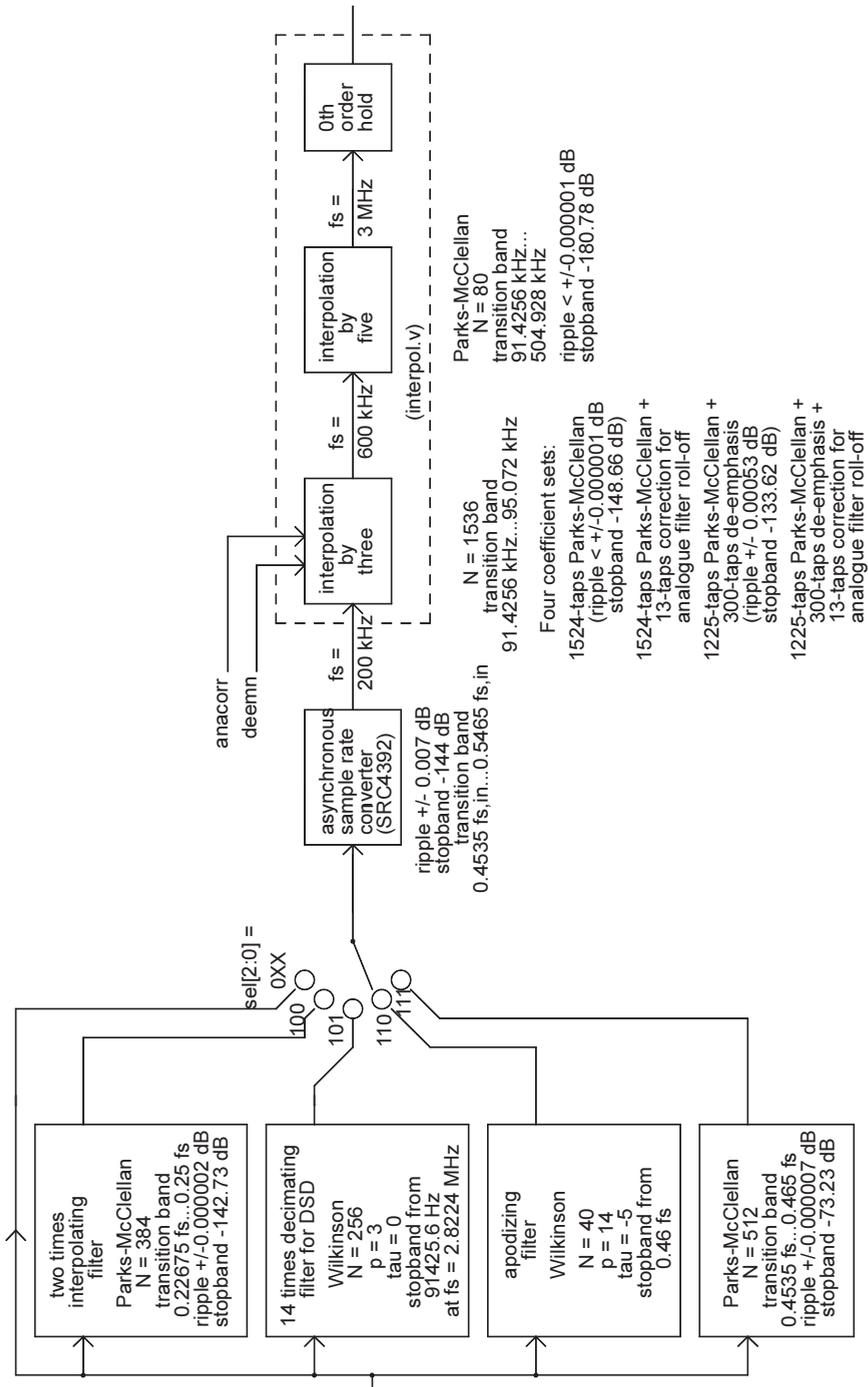


Figure 16 Digital filter chain.

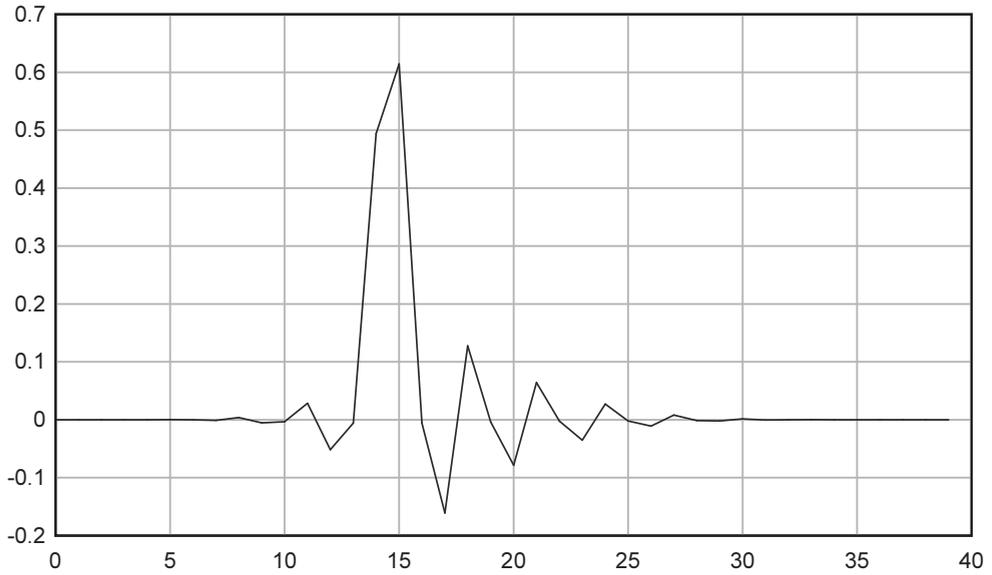


Figure 17 Impulse response of the apodizing filter (asymmetric Wilkinson filter with $N = 40$, $p = 14$ and $\tau = -5$, stopband starting at $0.46 f_s$)

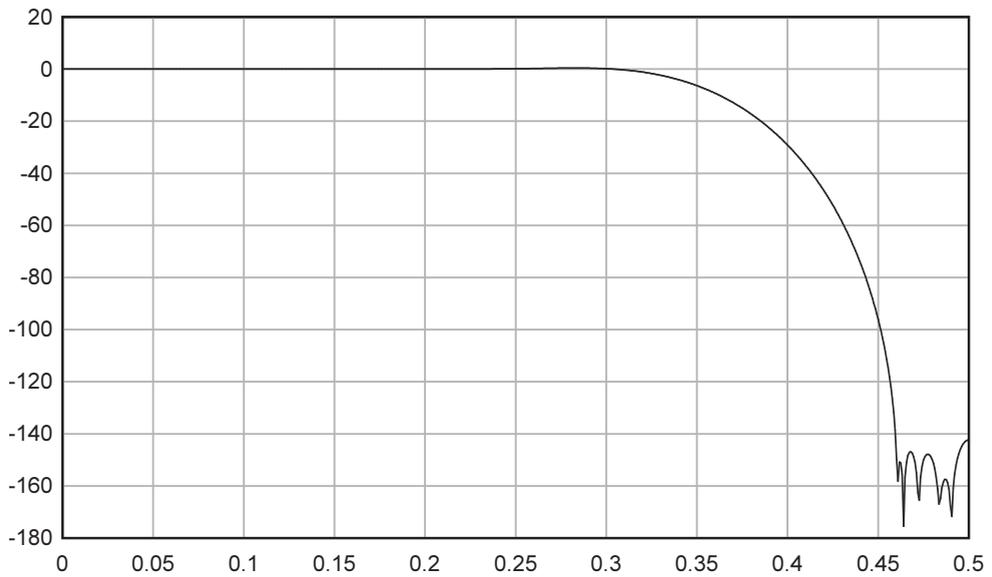


Figure 18 Magnitude of the response of the apodizing filter, horizontal axis: frequency normalized to the sample rate, vertical axis: magnitude in dB.

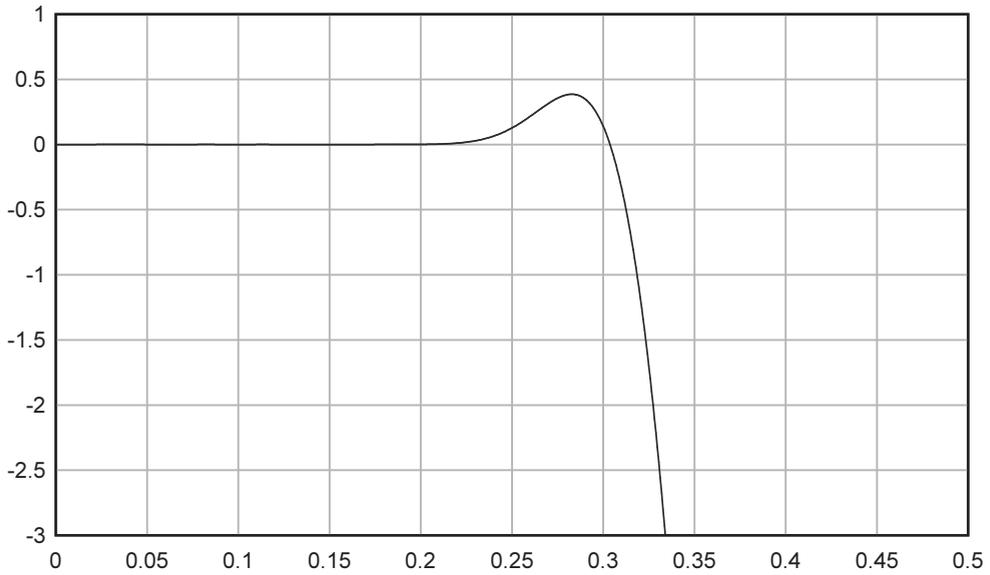


Figure 19 Zoomed-in passband response of the apodizing filter.

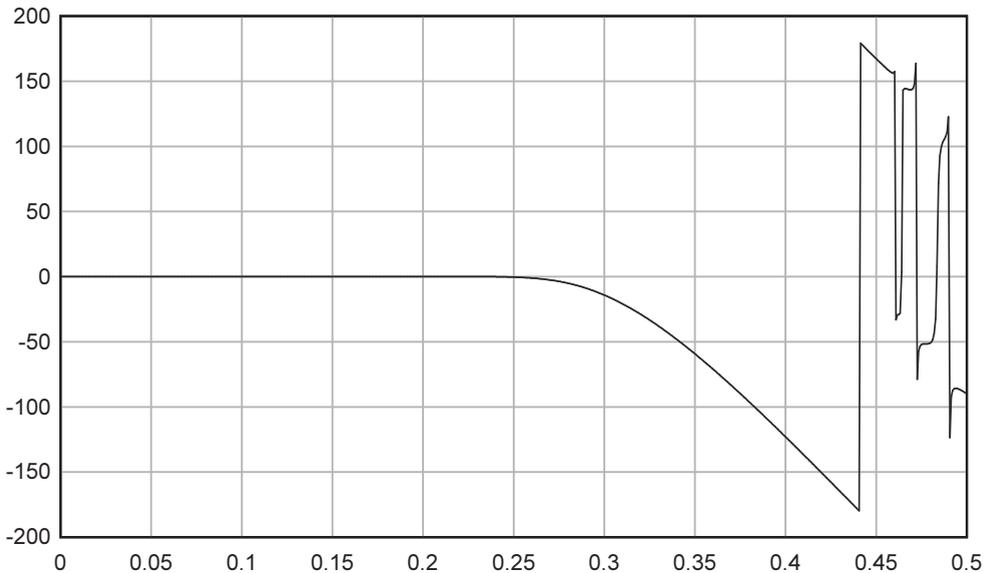


Figure 20 Phase error in degrees of the apodizing filter (difference from the theoretical 14.5 tap delay) versus the frequency normalized to the sample rate. The phase jump from -180° to $+180^\circ$ at 0.44 is an artefact of the phase calculating routine.



Instead of an apodizing filter, one can choose a brick-wall filter to suppress the imaging in the half-band filters of the SRC4392. For sample rates up to 96 kHz, a two-times interpolating filter is used. This suppresses imaging and as a bonus, it doubles the frequency distance between the response peaks of the SRC4392, as the SRC4392's interpolating filter response scales with its input sample rate. For sample rates between 100 kHz and 190 kHz, a non-interpolating filter is used that only partly suppresses imaging. Above 190 kHz, no pre-filtering is needed because the interpolation chain suppresses the images. The third option is not to filter at all between the input and the SRC4392. This reduces the latency of the entire filter chain, which can be useful in professional and in video applications.

The logic design of the FIR filters was done by the Xilinx FIR compiler version 5.0. They run at an eight times multiplied clock rate, 216 MHz, generated with one of the FPGA's DCMs. The FIFOs and double-flip-flop synchronizers used for clock domain transitions are not shown in Figure 16, nor are the dithered rounding stages between the filters or the clock doubler used for generating the prefilter output bit clock when it is in interpolating mode. This clock doubler is a very simple circuit that generates a pulse at both the rising and the falling edges of the incoming bit clock; it was not possible to use the PLLs and DCMs inside the FPGA as bit clock doublers, as they were not compatible with the frequency range and jitter specification, respectively, of the incoming bit clock.

Some audio files are in the so-called DSD (direct-stream digital) format, which is a Sony trade name for a sigma-delta modulate. Because of its excessive ultrasonic noise and because proponents of DSD usually praise its temporal coherence [33], the only logical thing to do with DSD is to smoothly low-pass filter it. Again somewhat arbitrarily, a Wilkinson filter of length 256 with flatness parameter $p = 3$ and time shift parameter $\tau = 0$ (symmetrical response) is chosen to convert DSD into PCM while reducing its ultrasonic noise and while retaining as much as possible of its temporal coherence. The DSD filter decimates its 2.8224 MHz-sample-rate input signal 14 times to make it fit with the input sample rates supported by the SRC4392. The stopband of the DSD filter begins at 91.4256 kHz, which is the end of the passbands of the asynchronous sample rate converter and the interpolation chain. This ensures that the DSD filter apodizes the entire chain, so the transition- and stopbands of the SRC4392 and the interpolation chain have no impact on the shape of the overall impulse response. The responses of the DSD filter are shown in **Figure 21**, **Figure 22** and **Figure 23**.

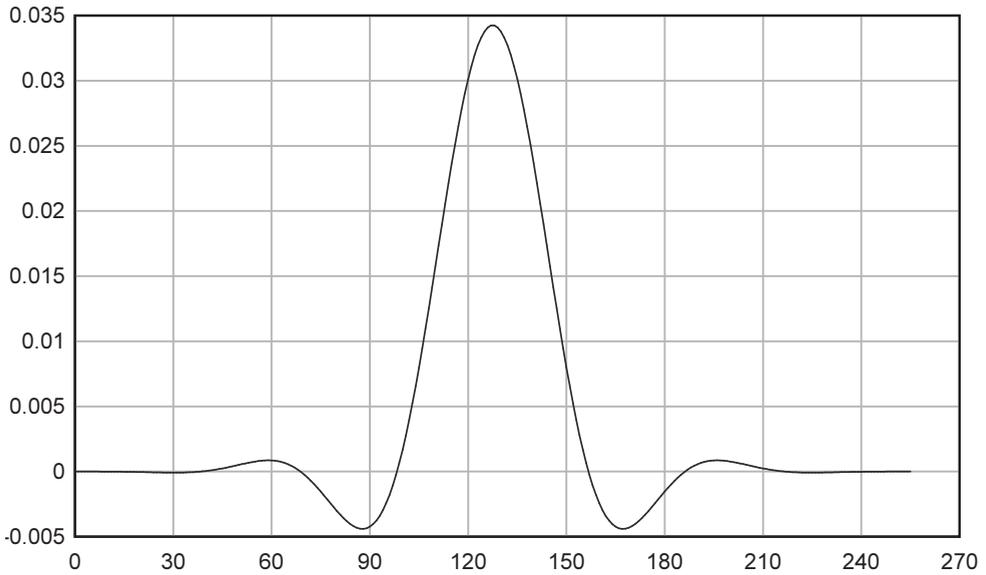


Figure 21 Impulse response of the DSD filter (Wilkinson $N = 256, p = 3, \tau = 0$, stopband starts at $0.03239285714 f_s$). At 2.8224 MHz input sample rate, the whole response is only $90.35 \mu\text{s}$ long.

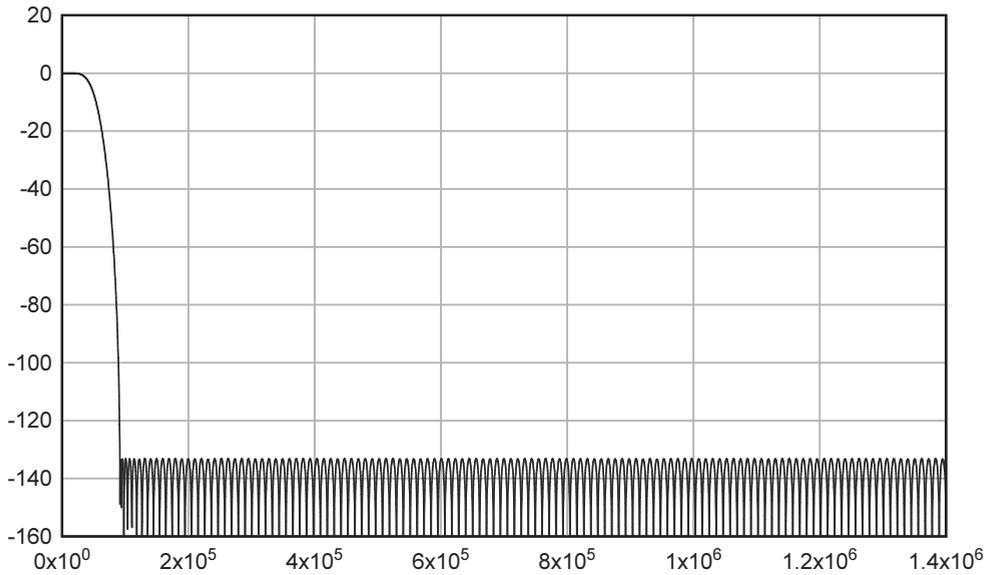


Figure 22 Magnitude response of the DSD filter, vertical scale in dB, horizontal scale in Hz (for 2.8224 MHz input sample rate).

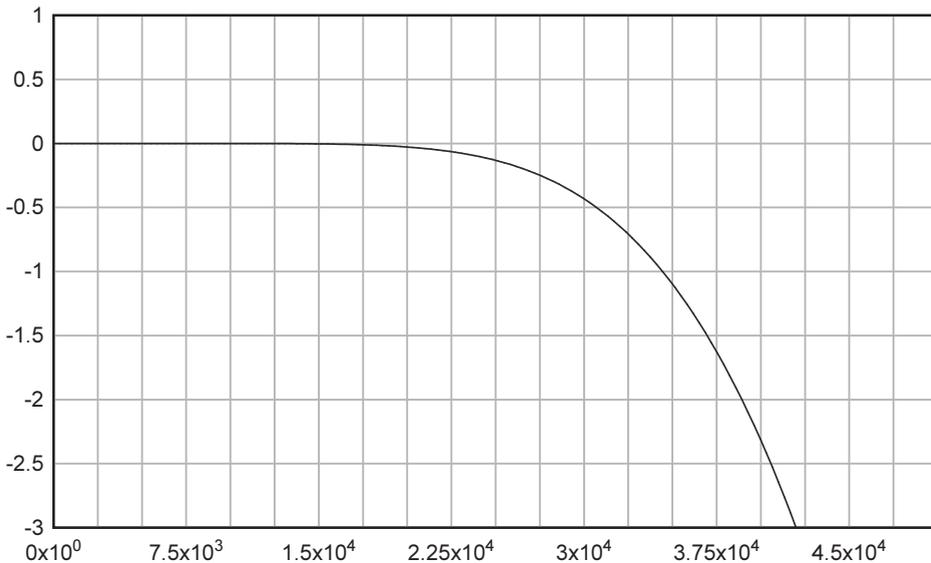


Figure 23 Zoomed-in magnitude response of the DSD filter, vertical scale in dB, horizontal scale in Hz (for 2.8224 MHz input sample rate).

The next stage is the SRC4392 asynchronous sample rate converter. It converts whatever comes in to a 200 kHz sample rate⁷. As its output sample rate will normally be greater than its input sample rate, its output decimation filter is not needed. It is switched off, eliminating its ripples and any other artefacts. In the case of DSD, the sample rate of the signal going from the DSD filter to the SRC4392 is slightly higher than the SRC4392's output sample rate (201.6 kHz versus 200 kHz). Nevertheless, as the DSD filter suppresses anything above 91.4256 kHz, no aliasing can occur.

Another special case is 192 kHz PCM. The transition band of the interpolation chain inside the SRC4392 extends up to 0.5465 times the input sample rate, which is 104.928 kHz at 192 kHz input sample rate. At 200 kHz output sample rate, 104.928 kHz aliases to 95.072 kHz. This alias is blocked by the interpolation chain between the SRC4392 and the sigma-delta modulator, the stopband of which starts at 95.072 kHz.

After the SRC4392, two interpolating FIR filters bring the sample rate up to 3 MHz. The signal then goes into a zero-order hold filter, that is, it is simply kept constant until the next sample is calculated. The first of the two interpolating filters includes compensation for the analogue post filter; the compensation is switched off when the apodizing filter peaking already more than compensates for analogue filter roll-off.

⁷ The 200 kHz SRC4392 output sample rate is high enough not to be the signal bandwidth bottleneck, it allows convenient multiplication factors in the interpolation chain and it is compatible with clock frequencies for the SRC4392's output I2S interface that cause no interference at odd multiples of 13.5 MHz. The SRC4392 output I2S interface is run in slave mode at 10.4 MHz bit clock and 200 kHz word clock.



The coefficients for the interpolating filters were calculated with the equiripple FIR filter design program of which McClellan, Parks and Rabiner published the complete FORTRAN code in 1973 [39]. In fact, most FIR filters are designed using some variant of their program. The original code needs only a few modifications to run under gfortran, a modern open-source Fortran compiler; there is one ARCOS that needs to be changed into an ACOS, the double-precision functions now also need to be declared as double-precision in the part of the program from which they are called, numerical values (literals) intended as double-precision numbers need to get an exponent D0 to prevent unintended rounding to single precision and the option to store the impulse response on punched cards is not supported anymore. The program supports filters with multiple pass-, stop- and transition bands, and it can easily be modified to include non-flat passbands and frequency-dependent ripple magnitudes.

In principle a modified version of McClellan's program can design FIR filters that compensate for analogue filter roll-off and for the ripple of the interpolation filter of the SRC4392. I soon found out, however, that the filters quickly become excessively long if you want their responses to follow some funny shape with extreme accuracy. I therefore decided not to compensate for SRC4392 ripple and to use a different method for analogue filter roll-off compensation.

By manually tweaking coefficients in an Excel spreadsheet, I quickly found a 13-taps FIR filter that compensates the roll-off of my analogue filter to +0 dB / -0.02 dB up to 20 kHz, +0 dB / -1 dB up to 61.22 kHz (just monotonic roll-off, no ripple, non-idealities of the components of the analogue filter and the optional transformer not included). This compensating filter doesn't boost any frequency by more than 0.65 dB. I convolved its impulse response with the impulse response of a normal, very-low-ripple Parks and McClellan filter⁸. The combined impulse response is used as one of the coefficient sets of the three-times interpolating filter in the interpolation chain.

The CD standard features a 50 μ s/15 μ s pre-emphasis/de-emphasis option that is rarely used, but needs to be supported anyway. To mimic this with a FIR filter, I have simply calculated the step response of a continuous-time 50 μ s/15 μ s de-emphasis filter and sampled it at 600 kHz, the output sample rate of the first interpolation stage. The differences between adjacent samples were used as the weights for the de-emphasis FIR filter. To get a finite length, I have tapered off the impulse response. By tweaking the 15 μ s time constant, I could get a bit better match between the magnitude response of the FIR and the magnitude response of an ideal continuous-time 50 μ s/15 μ s de-emphasis filter.

The calculated impulse response was again convolved with the impulse response of a low-pass filter to get an alternative coefficient set for the first interpolating filter. The total number of coefficient sets is four: plain low-pass, low-pass with correction for the analogue reconstruction filter, low-pass with de-emphasis, low-pass with correction for the analogue reconstruction filter and with de-emphasis.

⁸ I generally set the weight for passband ripple to 1 and the weight for stopband ripple to 2, which in McClellan's program means that the pre- and post-echoes are each suppressed as much as the stopband.



9 The sigma-delta modulator

I've implemented a sigma-delta modulator that can work in three different modes. In two modes it is a properly dithered sigma-delta that incorporates a multibit quantizer and a kind of pulse width modulator with randomly rotated pattern (as explained in section 3.1), in the third mode it is a chaotic single-bit sigma-delta with undithered quantizer⁹. The PWM modes have the advantage of being theoretically free of tones and almost free of noise modulation artefacts, but their disadvantage is that the sample rate of the quantizer can only be a fraction of the clock rate. One of the two PWM modes has a quantizer sample rate of a quarter of the 27 MHz clock with five quantization levels, the other has a quantizer sample rate of one eighth part of the 27 MHz clock with nine quantization levels. The chaotic single-bit sigma-delta mode has a quantizer sample rate of the full 27 MHz, hence it has a higher oversampling ratio than the PWM versions. See **Figure 24** for a block diagram.

The sigma-delta modulator has been designed using the noise transfer function method with an out-of-band noise gain of about 1.5 [4], as explained in section 3.1. Essentially, one models the quantizer as an addition point for quantization noise and pretends that everything is linear to calculate the loop filter coefficients. This reduces the problem to a filter synthesis problem. Extensive transient simulations then have to show what impact non-linear effects have. The filter synthesis calculation is conceptually straightforward but quite long due to the number of terms; it took me seven pages of A4 paper to write out the equations. There is no need to solve any seventh-order equations, you just equate the polynomial coefficients to the values you want them to have and end up with a system of first-order equations.

A complicating factor in the PWM case is the fact that the quantizer runs at a different sample rate than the loop filter. The correct way to calculate the coefficients probably entails impulse-invariant transforms, but I've simply calculated the coefficients pretending that everything runs at the quantizer sample rate, and then scaled down the signal going into each accumulator by the ratio of the sample rates, see **Figure 25**. As long as all signals vary slowly compared to the sigma-delta clock rate, adding a signal 6.75 million times per second has the same effect as adding a quarter of the signal 27 million times per second.

Rather than using noise transfer synthesis, some authors advocate the use of a more scientific bounded-input, bounded-output method for multibit converters, see reference [40]. I haven't used their method because the required number of quantization levels is too large for a PWM-based modulator; with a given clock rate, this would result in a rather low quantizer sample rate and, hence, in a low oversampling factor.

The chosen noise transfer functions for the loops with a pulse width modulator are fifth order, having zeroes at DC, in the upper part of the human auditory range (11869 Hz) and in the upper part of the feline auditory range (37533 Hz). The analogue reconstruction filter is sixth order; although I

⁹ In fact the PWM modes are also slightly chaotic because the poles of the resonators are just outside the unit circle.

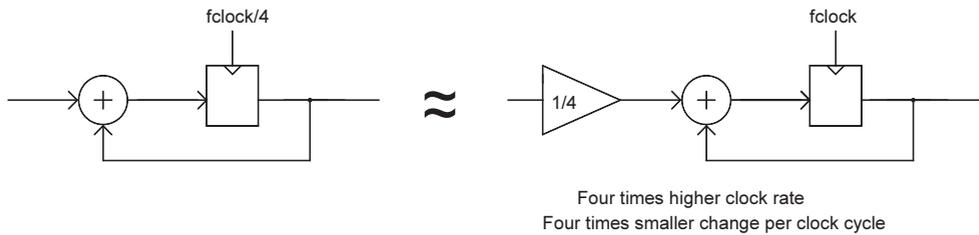


Figure 25 Illustration of the approximation used to handle the multirate nature of the PWM cases.

haven't checked if this is truly needed, it seemed reasonable to use a reconstruction filter order that is slightly higher than the order of the sigma-delta.

The chaotic version has two all-pass sections included in its noise transfer function, one to destabilize low-frequency limit cycles and one to destabilize limit cycles near half the sample rate (following Lars Risbo's recommendation [13]). This increases the order of the chaotic version to seven, but as the allpass sections have little influence on the magnitude of the noise transfer function, the noise outside the signal band still only increases with a fifth-order slope.

Single-loop sigma-delta modulators of order greater than two and chaotic modulators of any order are conditionally stable and may burst into uncontrollable low-frequency oscillations (overload cycles) when overloaded. State variable limiting has been used to ensure proper recovery from overload; at abnormally high signal levels, the outputs of all integrators except the last one are clipped, which basically reduces the loop to a non-chaotic first-order system.

10 Random number generation for dithering

For various practical reasons, the random numbers used for dithering come from pseudorandom number generators rather than true random number generators. The used pseudorandom number generation algorithm is the so-called XORSHIFT128+ algorithm [41]. Its period is $2^{128} - 1$ cycles, which comes down to a period time of almost $4 \cdot 10^{23}$ years when running at 27 MHz clock rate. This means that instead of being continuous, its output spectrum consists of discrete peaks at all multiples of $7.93459 \cdot 10^{-32}$ Hz, but one would have to listen for almost $4 \cdot 10^{23}$ years and to have an exceptionally good auditory memory to notice this. The complete signal path uses 11 instances of XORSHIFT128+, all with a different seed (starting number).

Each XORSHIFT128+ produces 64 bit pseudorandom words. These are converted into 33 bit triangular-pdf dither by splitting them into two 32 bit words, one made of the odd and one made of the even bits. These 32 bit words are added and the MSB is inverted to get rid of the DC component. The 32 bit words are also subtracted; as shown in reference [42], one can use the sum of two random words for one channel and the difference for the other channel of a stereo system. To bring some true



randomness into the circuit and to avoid discussions about whether the noise is close enough to being ergodic, I have made a part of the seeds of the pseudorandom number generators dependent on the ignition delay of the voltage reference tube and on the crystal oscillator start-up time. The ignition delay depends on radioactive decay or on cosmic radiation and is truly random according to the present physics theories.

11 Start up and user interface

There are several housekeeping functions to be performed, such as writing the correct settings into the DIX4192 and the SRC4392, starting up circuits in the right order, measuring the start-up times of the 85A2 and the crystal oscillator and supplying the numbers as part of the seeds to the pseudorandom number generators, controlling the output relay and providing some sort of user interface. These tasks can be performed by specially designed hardware state machines or by firmware running on some kind of embedded computer/microcontroller [43]. A microcontroller can itself be implemented on the FPGA, if desired.

To keep things simple, I chose a very simple user interface (just a few knobs and lights) and implemented everything in hardware. To minimize interference, all required circuits either run on the sigma-delta modulator clock, or on a multiple of the sigma-delta modulator clock, or are shut down before the audio outputs are activated.

A silly feature of the DIX4192 is that one cannot simply set it to use the same clock frequencies for its I²S output as it finds on its active SPDIF/AES3 input. Instead, one needs to read out the PLL2 multiplication factor RXCKR that it automatically selects and update its PLL2 output division ratio RXCKOD accordingly. This needs to be done whenever the input sample rate changes so much that the DIX4192 changes its RXCKR value. Instead of making some interrupt handling thing, I made a state machine that frequently checks the RXCKR value and adjusts the RXCKOD value accordingly. It also changes the input selector value when the user wants to switch to another input. To minimize interference, this is done with an SPI clock frequency equal to the sigma-delta clock frequency. To prevent periodic data patterns having strong spectral lines, there are randomized wait times between the SPI read and write actions.

To make blind listening tests easier, the switches that control the pre-filter and the sigma-delta method have a “surprise” setting. When this setting is chosen, you randomly get one of the pre-filters or one of the three sigma-delta methods. The random number is stored in an EEPROM, so you can always switch off the DAC and continue listening some other time; my late colleague Henk ten Pierick used to say that you have to listen to audio equipment for a week or two before you can really decide if it is any good. You can also do ABX testing (or rather ABCX testing) if you like, by comparing the “surprise”-setting with the other, known, settings. When the “surprise” pushbutton is pressed, neon lamps show what the randomly chosen filter and sigma-delta method were. When the “surprise” pushbutton is released, new random values are chosen.



It should be noted that the various filter options have different delays. No attempt has been made to equalize delays in surprise mode. Also keep in mind that at low sample rates, the 0.386 dB gain peak and the roll-off of the apodizing filter lie within the frequency range traditionally considered audible for humans, as shown in Figure 19. Otherwise the gain matching between the various modes should be good enough for a blind test; the biggest difference is a 0.017 dB difference in gain between sigma-delta methods 1 and 2.

As there is only one DSD filter, the filter “surprise” setting doesn’t do anything for DSD. Depending on the setting of switch S1D on the FPGA module, the clipping lamp can be suppressed during the “surprise” mode to ensure that you don’t subconsciously recognize the mode by looking for differences in the clipping light.

12 PCB design

Much as I like point-to-point wiring, perfboard and chassis-mounted electronics, they are not particularly suitable for a critical mixed-signal circuit like the valve DAC. In a sigma-delta DAC, any crosstalk from the sigma-delta modulate to the voltage reference or to the clock increases the noise floor. On the very first prototype, built with point-to-point wiring on a copper-clad board, touching a thin coaxial cable that carried the sigma-delta modulate was enough to raise the noise floor by 10 dB or so.

Hence, I decided to design a multilayer PCB for the valve DAC. A four-layer board provides two layers of shielding when you use the inner layers as supply and ground planes and put all the digital stuff on the back side and all the analogue stuff on the front.

Ideally, everything should be made in surface mount technology to minimize the number of wires sticking out of the wrong side of the board. This seemed inappropriate for the valve circuits, though¹⁰. As a compromise, I used SMD technology for the active digital circuitry and through-hole components for the analogue and mixed-signal circuitry, and kept some distance between the FPGA module and the actual DACs. To keep the board costs down (it is still the most expensive part of the DAC), I used a fairly straightforward PCB technology without buried vias [44]. This made it necessary to keep some distance between analogue and digital anyway, as all vias extend to the other side of the board. Where possible I used relatively large SMD components (like 1206 or 0805 size) to make soldering them a bit easier for people over 40, such as myself.

The point-to-point wired prototype had several problems related to transmission line reflections. The weirdest one was a gross distortion caused by random switching between two filters caused by the sample frequency counter getting confused by reflections on the word clock line. Terminating the lines at the source side with roughly the right impedance was enough to solve these issues. The PCB

¹⁰ People who have seen the rebuilt Colossus in the National Museum of Computing, block H, Bletchley Park, England, know that Colossus was actually made with surface mounted valve holders so valves could be mounted on both sides of the huge mounting panel. The resistors and capacitors were normal wired components, though.



also contains several resistors meant as transmission line terminators. The transmission lines were designed assuming that the dielectric between the outer and inner metal layers is 360 μm thick and has a relative dielectric constant of 4.35, which should more or less match the Eurocircuits PCB Proto technology¹¹. The dielectric thickness between the two inner layers is not critical, as long as the dielectric can easily handle 300 V.

Except for the first filter capacitors and the DC blocking capacitors, the reconstruction filters are mounted on a separate single-layer board. As the filter components are quite large, have only few connections between them, carry no high voltages and are not exceedingly sensitive to board leakage currents, a simple single-layer board with no solder mask works fine and is much cheaper than using a larger four-layer board.

Gerber files, the complete KiCAD database and some additional information are available on the *Linear Audio* website (www.linearaudio.net/downloads).

13 Measured results

Unless otherwise noted, all measured results are with a balanced output filter as shown in Figure 14. The filter board has space reserved for an optional output transformer, but it was not populated.

13.1 In-band noise

The four-layer-board version of the valve DAC was subjected to a series of measurements, experiments and small improvements. These showed that:

- The PWM8 mode gives the best in-band noise performance. Simulations show that it also has the largest headroom and is therefore the least likely to run into clipping when used with the loud setting. All in all, PWM8 has the best dynamic range.
- Changes that made the clock signals coming out of the EF80s stronger reduced the noise floor.
- The noise floor also dropped when C5 and C24 were added to the DAC core (see Figure 9) to give the sigma-delta modulate a better return path. No further improvement was obtained by trying to compensate for the current through the *sdin* pin by coupling its inverse to the -137 V line through a 4.7 pF capacitor. The changes that did work are all included in the board layout on the Linear Audio website; on my board they were made in a rather improvised manner.
- Differences between the anode voltage of U1A/U2B and the anode voltage of U1B/U2A cause a noise increase and the signal swing between the anodes causes noise modulation. Hence, the anode resistors must be trimmed for minimum idle channel noise. This effect is slightly smaller in PWM8-mode than in PWM4-mode.

¹¹ Eurocircuits uses two 180 μm -thick layers of PR7628 on top of each other [44]. According to [45], this type of prepreg has a relative dielectric constant between 4.1 and 4.6.



These observations can be explained by crosstalk from the sigma-delta modulate to the clock signals and by considering what happens to the cathode potential of U_{1A}/U_{1B} or U_{2A}/U_{2B} during the clock phase when the tail current is off. For example, consider U_{1A}/U_{1B} in Figure 9 and assume that U_{1A} has a higher anode potential than U_{1B} . Due to the finite voltage gain of an E88CC triode, the voltage at which the cathode will settle when clockn is low will then be higher when U_{1A} has a high level at its grid than when U_{1B} has a high level at its grid. Some intermediate voltage level will occur when there is a transition on the grids of U_{1A} and U_{1B} while clockn is low.

Hence, depending on the previous and the present bit value, there can be four different values for the cathode potential of U_{1A}/U_{1B} at the time when clockn starts going high again. The lower the cathode potential, the faster U_{1A} or U_{1B} starts to conduct significant current. As a result, the weight of the present bit depends on previous bits. As this is a multiplicative rather than an additive effect, out-of-band quantization noise will intermodulate with out-of-band quantization noise, and a part of the resulting intermodulation product will lie at low frequencies and degrade the in-band noise floor.

In fact the main problem occurs after transitions of the sigma-delta modulate. When the grid of U_{1A} is high continuously, the cathode settles to a higher voltage than if the grid of U_{1B} had been high continuously, but due to its higher anode voltage, U_{1A} also needs less grid-to-cathode voltage to turn on, which compensates the effect. Hence the smaller noise modulation in PWM8-mode, which has less transitions than the PWM4-mode.

Anyway, when U_{1A} and U_{1B} have the same anode voltage (or, more precisely, an offset between their anode voltages that exactly compensates for the mismatch between the triodes), the cathode will always settle to the same voltage and the pattern sensitivity will be gone.

The final noise floor in PWM8 mode and with the loud setting was -85.76 dB(A) on the left and -91.29 dB(A) on the right channel with respect to a full-scale sine wave. These are very decent values for such a simple circuit. The noise with a full-scale 1 kHz signal present was -75.37 dB(A) left and -76.74 dB(A) right, with 20 Hz full-scale it becomes -69.35 dB(A) and -69.39 dB(A) due to the reactances of the DC blocking capacitors. The noise gradually increases with the signal amplitude, which is more similar to the type of noise modulation found in an instantaneous companding system than to the periodic noise modulation found in an undithered or inappropriately dithered digital system. If you wish to reduce noise modulation, you can reduce the impedance loading U_1/U_2 or try devices with higher voltage gain, such as pentodes (partition noise will be negligible because of the high signal levels).

13.2 Out-of-band noise

The out-of-band noise is about 10 mV RMS in PWM8 mode, 4 mV RMS in PWM4 mode and 400 μ V RMS in chaotic mode. Although the 10 mV RMS of the PWM8 mode might cause sub-slewing TIM in an amplifier with a bipolar input stage, no input filter and a relatively small loop bandwidth, none of these values are likely to cause significant sub-slewing TIM in a valve amplifier because of the better



open-loop linearity of valves.

13.3 Harmonic distortion

The distortion consists mainly of the third harmonic. In PWM8 mode, which is the mode with least distortion, its level is around -76 dB (0.016 %) at 1 kHz and at 10 kHz, with no significant difference between the channels. It increases to -63 dB (0.07 %) at 20 Hz.

13.4 Frequency response

The output level at 0 dBFS in loud mode was measured to be 1.2 V RMS at 1 kHz. At other frequencies between 20 Hz and 20 kHz, it remained within +0/-0.21 dB from the 1 kHz value using the steep (non-apodizing) filter setting. It is unclear how much of the variation was due to the measuring equipment.

13.5 Single-ended variant

A simple way to make a single-ended variant is to ground the DAC_out_n pin (leaving C₃₂ in place) and to only filter DAC_out_p. This saves half the potcores and some polystyrene capacitors. I did some quick measurements on this variant and found no major performance differences except the halved output signal voltage and a reduction of about 3 dB of the ratio of a full-scale signal to the idle-channel noise.

14 Conclusion

Having demonstrated that it is quite possible to use valves for the actual digital to analogue conversion, I don't want to hear anyone call a solid-state DAC followed by a valve-based output buffer a valve DAC anymore!

15 Acknowledgements

Hans Rosenberg helped me enormously when I was struggling with FPGA boards for the first time in my life. His help is greatly appreciated.

I also want to thank the late Henk ten Pierick for the many interesting discussions we had when we were both getting to understand sigma-delta modulators.

A big thanks to Jan Didden for publishing a most interesting audio bookazine the past seven years, it is a pity that he stops.

Hein van den Heuvel lent me money when I went to an NVHR fair to buy reference valves, but absent-mindedly forgot to take cash with me.

Last but not least, I want to thank Connie, the late Harvey, the late Truffel and Donny for occasionally reminding me that there are other things you can do in your spare time than just designing DACs.



16 References

- [1] Robert Haagsma, *Passion for vinyl – A tribute to all who dig the groove*, Record Industry BV, Haarlem, 2nd edition February 2014, ISBN 978-94-6228-264-3
- [2] Heinrich Pfeifer, Werner Reich and Ulrich Theus, *Circuit arrangements for averaging signals during pulse-density D/A or A/D conversion*, US patent 4947171, 7 August 1990
- [3] Michael A. Gerzon and Peter G. Craven, "Optimal noise shaping and dither of digital signals", Audio Engineering Society preprint 2822, presented at the 87th convention, October 1989
- [4] Kirk C.-H. Chao, Shujaat Nadeem, Wai L. Lee and Charles G. Sodini, "A high order topology for interpolative modulators for oversampling A/D converters", *IEEE Transactions on Circuits and Systems*, vol. 37, no. 3, March 1990, pages 309...318
- [5] Fred D. Waldhauer, *Feedback*, John Wiley & Sons, 1982, ISBN 0-471-05319-8
- [6] Robert A. Wannamaker, Stanley P. Lipshitz, John Vanderkooy and J. Nelson Wright, "A theory of nonsubtractive dither", *IEEE Transactions on Signal Processing*, vol. 48, no. 2, February 2000, pages 499...516
- [7] Stanley P. Lipshitz, Robert A. Wannamaker and John Vanderkooy, "Quantization and dither: a theoretical survey", *Journal of the Audio Engineering Society*, vol. 40, no. 5, May 1992, pages 355...375
- [8] Michael A. Gerzon, Peter G. Craven, J. Robert Stuart and Rhonda J. Wilson, "Psychoacoustic noise shaped improvements in CD and other linear digital media", Audio Engineering Society preprint 3501, presented at the 94th convention, March 1993
- [9] Stanley P. Lipshitz, Robert A. Wannamaker and John Vanderkooy, "Dithered noise shapers and recursive digital filters", Audio Engineering Society preprint 3515, presented at the 94th convention, March 1993
- [10] Kurt Wiesenfeld and Fernan Jaramillo, "Minireview of stochastic resonance", *Chaos*, vol. 8, no. 3, September 1998, pages 539...548
- [11] Steven R. Norsworthy and David A. Rich, "Idle channel tones and dithering in delta-sigma modulators", *Preprints of the 95th Audio Engineering Society convention*, preprint 3711, October 1993
- [12] Richard Schreier, "On the use of chaos to reduce idle-channel tones in delta-sigma modulators", *IEEE Transactions on Circuits and Systems-I: Fundamental theory and applications*, vol. 41, no. 8, August 1994, pages 539...547



-
- [13] Lars Risbo, *Σ - Δ modulators - Stability analysis and optimization*, Ph. D. thesis, Technical University of Denmark, 16 June 1994
- [14] Robert C. Hilborn, *Chaos and nonlinear dynamics - An introduction for scientists and engineers*, second edition, Oxford University Press, New York, 2000, ISBN 0 19 850723 2
- [15] Kuan-Dar Chen and Tai-Haur Kuo, "An improved technique for reducing baseband tones in sigma-delta modulators employing data weighted averaging algorithm without adding dither", *IEEE Transactions on Circuits and Systems-II: analog and digital signal processing*, vol. 46, no. 1, January 1999, pages 63...68
- [16] Peter G. Craven, *Analogue and digital converters using pulse edge modulators with non-linearity error correction*, US patent 5548286, 20 August 1996
- [17] T. A. Brubaker and G. A. Korn, "Accurate amplitude distribution analyzer combining analog and digital logic", *The review of scientific instruments*, vol. 32, no. 3, March 1961, pages 317...322
- [18] David O. Clayden, "Circuit design of the Pilot ACE and the Big ACE", chapter 19 of B. Jack Copeland (Editor), *Alan Turing's Automatic Computing Engine - The master codebreaker's struggle to build the modern computer*, Oxford University Press, 2005, ISBN 0 19 856593 3
- [19] G. F. Weston, *Cold cathode glow discharge tubes*, London ILIFFE Books, Ltd., 1968
- [20] Philips data handbook ET7 (electron tubes part 7), *Gas-filled tubes*, August 1975
- [21] A. van der Ziel and E. R. Chenette, "Noise and impedance measurements in voltage regulator tubes", *Physica*, vol. XXIII, 1957, pages 943...952
- [22] J. F. Dix and K. B. Reed, "Cold cathode discharge tubes - Impedance and noise properties", *Electronic Technology*, vol. 39, no. 1, January 1962, pages 31...37
- [23] L. Knight, "Valve reliability in digital calculating machines", *Electronic Engineering*, January 1954, pages 9...13
- [24] Robert Adams and Tom Kwan, "Theory and VLSI architectures for asynchronous sample-rate converters", *Journal of the Audio Engineering Society*, vol. 41, no. 7/8, July/August 1993, pages 539...555
- [25] Texas Instruments, *SRC4392 two-channel, asynchronous sample rate converter with integrated digital audio interface receiver and transmitter*, December 2005, revised December 2012



-
- [26] Xianggang Yu, Terry L. Sculley and Jung-Kuei Chang, *Asynchronous sample rate converter and method*, United States patent US 7262716 B2, 28 August 2007
- [27] John Emmett, *Engineering Guidelines The AES/EBU digital audio interface*, European Broadcasting Union, 1995
- [28] Armour Research Foundation of Illinois Institute of Technology project E-050, *A study of crystal oscillator circuits*, Signal Corps Contract No. DA36-059-ac-64609, 14 August 1957, unclassified 149085
- [29] Chapter 7, "Transmitter theory", of W. W. Smith and Ray L. Dawley (Editors), *The "Radio" handbook*, Editors and Engineers limited, seventh edition, California, October 1940
- [30] Anatol I. Zverev, *Handbook of filter synthesis*, Wiley, New York, 1967
- [31] Peter G. Craven, "Antialias filters and system transient response at high sample rates", *Journal of the Audio Engineering Society*, vol. 52, no. 3, March 2004, pages 216...242
- [32] Paul Lesso and Anthony Magrath, "An ultra high performance DAC with controlled time domain response", Audio Engineering Society convention paper 6577, October 2005
- [33] Hans van Maanen, "On the audibility of "high resolution" digital audio formats and how to test this", *Linear Audio*, vol. 5, April 2013, pages 57...76
- [34] D. E. L. Shorter, W. I. Manson and E. R. Wigan, *Some experiments on the subjective effect of limiting the upper frequency range of programmes*, BBC Research Department report No. L-034 (1957/ 10), 1957
- [35] Tsutomu Oohashi, Emi Nishina, Norie Kawai, Yoshitaka Fuwamoto and Hiroshi Imai, "High-frequency sound above the audible range affects brain electric activity and sound perception", Audio Engineering Society preprint 3207, presented at the 91st Convention, October 1991
- [36] R. Lagadec and T. G. Stockham, "Dispersive models for A-to-D and D-to-A conversion systems", Audio Engineering Society preprint 2097, presented at the 75th convention, March 1984
- [37] math.stackexchange.com/questions/390810/improper-integral-sinx-x-converges-absolutely-conditionally-or-diverges
- [38] R. H. Wilkinson, "High-fidelity FIR filters based on central-difference operators", *IEE Proceedings on*



Circuits, Devices and Systems, vol. 141, no. 2, April 1994, pages 111...120

- [39] James H. McClellan, Thomas W. Parks and Lawrence R. Rabiner, "A computer program for designing optimum FIR linear phase digital filters", *IEEE Transactions on Audio and Electroacoustics*, vol. AU-21, no. 6, December 1973, pages 506...526
- [40] John G. Kenney and L. Richard Carley, "Design of multibit noise-shaping data converters", *Analog Integrated Circuits and Signal Processing*, 1993, volume 3, issue 3, pages 259...271
- [41] <http://xorshift.di.unimi.it/xorshift128plus.c>
- [42] Robert A. Wannamaker, "Efficient generation of multichannel dither signals", *Journal of the Audio Engineering Society*, vol. 52, no. 6, June 2004, pages 579...586
- [43] A. M. Turing, "On computable numbers, with an application to the Entscheidungsproblem", *Proceedings of the London Mathematical Society*, second series, volume 42, 1936/1937, pages 230...265
- [44] <http://be.eurocircuits.com/shop/orders/configurator.aspx>
- [45] Würth Elektronik, *Webinar 2013: Verbesserte Signalintegrität durch impedanzangepasste Leiterplatten*
- [46] Thomas W. Parks and James H. McClellan, "Chebyshev approximation for nonrecursive digital filters with linear phase", *IEEE Transactions on Circuit Theory*, vol. CT-19, no. 2, March 1972, pages 189...194
- [47] R. H. Wilkinson, "High-fidelity finite-impulse-response filters with optimal stopbands", *IEE Proceedings-G, Circuits, Devices and Systems*, vol. 138, no. 2, April 1991, pages 264...272



Appendix A. Parks-McClellan and Wilkinson filters

The z-domain transfer function of a FIR filter is

$$H(z) = \sum_{k=0}^{N-1} h(k)z^{-k}$$

In this equation, $h(k)$ is the filter's impulse response and z^k represents k sample periods of delay. For filters with symmetrical impulse response (meaning $h(k) = h(N - k - 1)$), it is convenient to split $H(z)$ into a part that represents a constant delay and a part that represents the magnitude of the transfer. Assuming an odd filter length,

$$H(z) = z^{-\frac{N-1}{2}} \sum_{k=-\frac{N-1}{2}}^{\frac{N-1}{2}} h\left(k + \frac{N-1}{2}\right) z^{-k}$$

The response under stationary sine wave excitation can be found by substituting

$$z = e^{j\omega T}$$

with $\omega = 2\pi f$, while T is the sample period. The constant delay of $(N - 1)/2$ sample periods has no impact on the magnitude of the response, so the magnitude of the response is given by

$$\sum_{k=-\frac{N-1}{2}}^{\frac{N-1}{2}} h\left(k + \frac{N-1}{2}\right) e^{-jk\omega T} = \sum_{k=1}^{\frac{N-1}{2}} 2h\left(k + \frac{N-1}{2}\right) \cos(k\omega T) + h\left(\frac{N-1}{2}\right)$$

Hence, the magnitude of the response of an odd-length symmetrical FIR filter is the sum of a bunch of weighted cosines of integer multiples of ωT . A similar expression can be obtained for even-length symmetrical FIR filters, except that an extra $\cos(\omega T/2)$ needs to be factored out.

The trick is to find the correct weights for the cosines. Once a frequency response is found that is realizable and optimal in whatever sense you want to optimize the filter, the corresponding impulse response and, hence, the corresponding filter coefficients follow by inverse discrete Fourier transform.

The Parks-McClellan method finds optimal weights in the Chebyshev sense, that is, it minimizes the largest error between the actual response and the desired response. The error is weighted by weight factors that the user can specify for each pass- and stopband, and with small changes to the program the weight within each band can be made to vary with any desired smooth function of fre-



quency. It is known from theory that an optimal design has $N/2 + 1$ (even N) or $N/2 + 1.5$ (odd N) extremes for a filter with length N . The algorithm starts with a guess for the frequencies of the extremes (equally distributed over the bands). It uses a closed-form expression for the magnitude of the ripples and assumes that the magnitude is at the peaks of the ripples at the guessed frequencies.

A sum of cosines of integer multiples of ωT can also be written as a polynomial of $\cos(\omega T)$. Therefore, polynomial interpolation (barycentric Lagrange interpolation) on $\cos(\omega T)$ can be used to find the magnitude response that passes through the guessed extremes. If the actual extremes of the interpolated transfer function are at different frequencies than the initial guess, these new frequencies are used as the new guess for the extremal frequencies and the whole procedure is repeated until it converges. This procedure is known as the Remez exchange algorithm. A more extensive explanation and the complete FORTRAN code can be found in references [46] and [39], respectively.

Wilkinson [47] described a method to make linear-phase low-pass filters with ripple-free passband and equiripple stopband. Instead of using cosines, the transfer can be written as a polynomial of $\sin^2(\omega T/2)$. Forcing the first-order up to the p -th order term to be zero ensures that the first $2p$ derivatives of the magnitude to frequency at $f = 0$ are 0, similar to a Butterworth filter. Wilkinson uses a Parks-McClellan-like algorithm to approximate $-1/\sin^{2p+2}(\omega T/2)$ over the stopband and then makes the total response $1 + \sin^{2p+2}(\omega T/2) \cdot Z$, where Z is the transfer found by the Parks-McClellan-like algorithm.

I have tweaked a copy of the McClellan program to change it into a symmetrical Wilkinson filter design program; it works well as long as p is not too large and the cut-off frequency is not too low, otherwise it explodes numerically despite the use of quadruple-precision numbers. Wilkinson's derivation only applies to odd-order filters, but I've used the program on even-order filters without any problems.

Wilkinson later extended his method to asymmetrical filters having approximately linear phase over their passband [38], using the fact that any finite impulse response can be written as the sum of a symmetrical impulse response and the difference between adjacent weights of another symmetrical impulse response. When designed for a delay corresponding to less than $(N - 1)/2$ sample periods, an asymmetrical Wilkinson filter has less pre- than post-ringing. The disadvantage is that when this is done, the magnitude response peaks to some extent. In fact you have to make a compromise between the reduction in pre-ringing and the amount of peaking.

Suppose you define the desired delay and suppose that this is an integer number of samples. If you forget for the moment that you need a stopband, it is clear what the filter coefficients should be: one for the tap corresponding to the desired delay and zero everywhere else. This response can then be split into two symmetrical impulse responses and from that the coefficients for the polynomials of $\sin^2(\omega T/2)$ can be calculated. Wilkinson actually does this in a smarter way that also works for non-integer delays.



To add a stopband and depending on the flatness parameter p , the low-order terms of the polynomials of $\sin^2(\omega T/2)$ are then kept as is, but a Parks-McClellan-like algorithm is used to change the weights of the higher-order terms such that they nearly cancel the lower-order terms over the stopband. This is rather similar to the procedure for symmetrical Wilkinson filters, except that it needs to be done twice, once for each of the two symmetrical impulse responses that together define the complete filter's asymmetrical impulse response.

I have also tweaked a copy of the McClellan program to change it into an asymmetrical Wilkinson filter design program. Again, it works well as long as p is not too large and the cut-off frequency is not too low, otherwise it explodes numerically despite the use of quadruple-precision numbers.



Appendix B. Passband ripples and pre- and post-echoes of digital low-pass filters

Practical digital low-pass filters can have a perfectly linear phase characteristic, but they never have a perfectly flat magnitude response over their passband. Very often, they have ripples in their passband. Under some assumptions, these ripples can be described in terms of pre- and post-echoes. The working assumption in this appendix is that there is some upper frequency f_{\max} above which frequency components of sounds are inaudible, also when heard in combination with other frequency components, and that the cut-off frequency of the filter is greater than or equal to f_{\max} . Hence, when an audio signal is passed through the filter, what happens above f_{\max} has no impact on the sound of the output signal.

B.1. Simplification by assuming equal frequency distance between the ripples

Following R. Lagadec and T. G. Stockham [36], assume that the filter has linear phase and that its passband ripples have equal frequency distances. The filter can then be regarded as equivalent to a cascade of two linear-phase filters, a low-pass with perfectly flat passband and a filter that only has equidistant ripples all the way up to the Nyquist frequency. This second filter is the one that does the damage.

A filter with an impulse response consisting of a pre-echo, a main response and a post-echo produces exactly this kind of equidistant ripple response. That is, when the echoes occur a time T before and after the main signal and have a magnitude α , the transfer function (in the s domain) is

$$H(s) = 1 + \alpha e^{-sT} + \alpha e^{sT} \quad (\text{B.1})$$

neglecting the filter's constant delay for simplicity. Substituting $s = j\omega$, the corresponding magnitude response is

$$H(j\omega) = 1 + 2\alpha \cos(\omega T)$$

This produces ripples with a frequency distance of $1/T$.

It is interesting to see some numerical examples of the relation between the ripple magnitude and the magnitude of the pre- and post-echoes, see **Table 4**. Quite small ripples are required if you want to get the echoes down to a level that doesn't require temporal masking to make them inaudible. Fortunately, McClellan's FORTRAN program can readily produce filters with extremely small passband ripples.



α	Echo magnitude	Ripple
0.03	-30.46 dB	+0.5061 dB / -0.5374 dB
0.005789	-44.75 dB	+0.09999 dB / -0.1012 dB
0.000403	-67.89 dB	+0.006998 dB / -0.007004 dB
0.000001	-120 dB	+0.00001737 dB / -0.00001737 dB

Table 4 Some examples of ripple and echo magnitudes according to the theory of reference [36].

B.2. More realistic filter responses

The passband ripples of practical digital low-pass filters are not entirely evenly spaced in the frequency domain. Hence, it is unclear to what extent the theory of R. Lagadec and T. G. Stockham applies to realistic filters. The pre- and post-echoes of a filter are, of course, determined entirely by its impulse response. Nonetheless, staring at the impulse response doesn't give us any idea of the echoes due to passband ripples, because the shape of the impulse response is mostly determined by what happens above the cut-off frequency.

The following method can make the impulse response aberrations due to passband ripples more visible:

1. Cascade the filter under test with an ideal continuous-time low-pass filter with cut-off frequency f_{\max} . As we are interested in what happens below f_{\max} , this doesn't affect the part of interest. When the passband is perfectly flat, the impulse response of the combined filter now has a $\sin(2\pi f_{\max}t)/(2\pi f_{\max}t)$ shape.
2. Sample the impulse response of the combined filter with a sample rate of $2 f_{\max}$ and ensure that the samples fall on the zero crossings of the $\sin(2\pi f_{\max}t)/(2\pi f_{\max}t)$ function. Ideally, this produces only one nonzero sample, the one related to the main response. Any other peaks or $\sin(x)/x$ shaped ripples are due to the imperfect passband response of the filter under test.

Step 1 can be done mathematically by calculating the convolution of the impulse responses of the filter under test and the ideal filter. Step 2 means that you only need to calculate this convolution at a countable number of points. Overall, the calculation to be done is:

$$\sum_{k=0}^{N-1} \frac{\sin(2\pi f_{\max}(t - kT))}{2\pi f_{\max}(t - kT)} h(k) \tag{B.2}$$

for

$$t = \frac{n}{2f_{\max}} + \left(\frac{N-1}{2} + \tau \right) T \tag{B.3}$$



In these equations, N is the length and T is the sample period of the filter under test, $h(k)$ is its impulse response, f_{\max} is the highest frequency of interest (as explained at the start of this appendix), t represents time and k and n are integers. τ is the number of sample periods that the main peak of the filter's impulse response is displaced from the centre; $\tau = 0$ for symmetrical filters.

As a typical example, I took a 128-tap, two-times-interpolating Parks-McClellan filter with 96 kHz output sample rate and a transition band from 21.768 kHz to 26.232 kHz. The weight was 1 in the passband and 12778 in the stopband, which resulted in a passband ripple of +0.007786 dB / -0.007793 dB and a stopband suppression of 143.08 dB (same order of magnitude as the filters used in the SRC4392). The echoes should theoretically have a level of -66.97 dB.

Using equations (B.2) and (B.3) with $f_{\max} = 21768$ Hz, the plot of **Figure 26** results. The largest echoes in the plot have a level of -69.22 dB. The difference with the theory of R. Lagadec and T. G. Stockham is less than 2.3 dB.

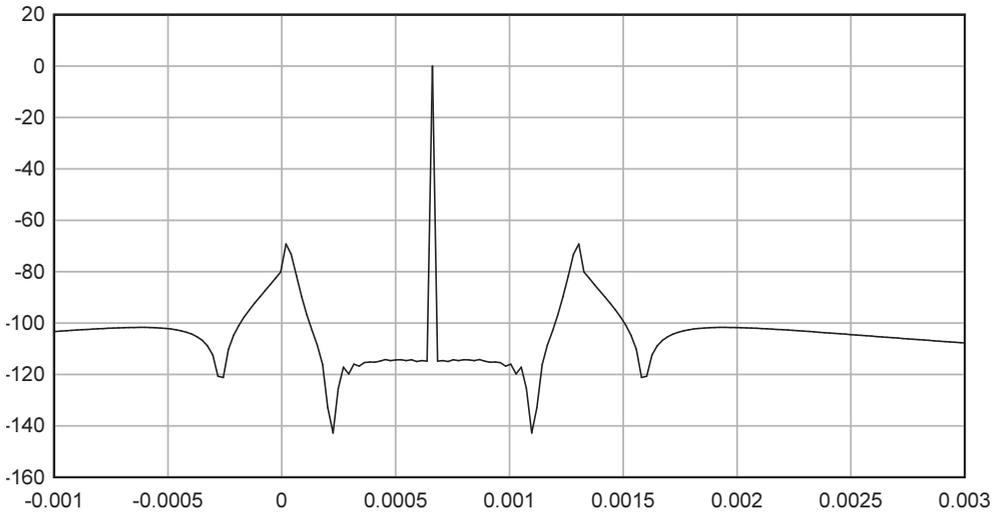


Figure 26 Impulse response aberrations due to the +0.007786 dB / -0.007793 dB passband ripple of a typical 128-tap filter. Horizontal scale: time points of equation (B.3) in seconds, vertical scale: result of equation (B.2) expressed in dB.



The AES Student Delegate Assembly is pleased to announce the Student Competitions at the upcoming **142nd Convention of the Audio Engineering Society in Berlin from 20 May to 23 May 2017.**

AES Student Recording Competition - Deadline 17th April 2017

An opportunity for students to present their recording works to a panel of renowned industry professionals to critique and judge, in four categories: Traditional Acoustic Recording, Traditional Studio Recording, Modern Studio Recording and Electronic Music and Sound for Visual Media.

AES Student Design Competition - Deadline 8th May 2017

A chance for aspiring hardware and software engineers to gain recognition for their hard work, technical creativity, and ingenuity. The only restriction is that the project must be based around audio. Participants must present a poster for the AES convention Student Design Exhibition, in either the Undergraduate or Postgraduate categories.

Both competitions are excellent learning opportunity for all students through feedback, constructive criticism, and conversation between those who are entered in the competitions as well as the experienced judges and those attending the exhibition.

The **143rd Convention of the Audio Engineering Society** will take a place in **New York, USA** from **18 October to 21 October 2017.**

The Student Recording Competition submission deadline for the 143rd is **18th September 2017.** The Student Design Competition submission deadline for the 143rd is **2nd October 2017.**

Further details, the complete rules and how to enter can be found at <http://www.aes.org/students/awards/>

Follow the Student Delegate Assembly (the AES's student ambassadors) at www.aes.org/students/blog, <http://facebook.com/AESsda> and <http://twitter.com/AESsda>.