Breaking the Sound Barrier: Mastering at 96 kHz and Beyond

James A. Moorer
Sonic Solutions

ABSTRACT:

The possibilities of the forthcoming high-density disks has generated considerable interest in recording and mastering of audio material at rates higher than 48 kHz and wider than 16 bits. One of the proposed high-definition audio formats is 96 kHz using 24-bit PCM. The difference in the processing required is not simply one of bandwidth: the signal processing itself must be carefully analyzed for its suitability as well. Additionally, Sony DSD offers another mastering format with even more technical challenges. Part of the attraction of DSD is that it could be viewed as a Lingua Franca of mastering formats since it can be converted to any release format with no additional degradation. This paper discusses the technical challenges involved in preparing audio material in these formats and proposes some solutions employing today's technology.

DVD[1] has a number of standard audio formats, both compressed and uncompressed. The uncompressed formats are PCM at bit widths of 16, 20, or 24, and sampling rates of 48 kHz and 96 kHz [1][2]. If we wish to take advantage of the possibilities of this new format, we must record and master music at the highest and widest values, which is 24-bits and 96 kHz. Other formats (such as 16-bit, 44.1 kHz) can be derived from any of the DVD PCM audio formats, if sufficient care is taken in the conversion process.

"Direct Stream Digital", or DSD®[3] is another mastering format [2, 3, 4]. There is not a corresponding release format at this time. In DSD, the signal is recorded directly from the 1-bit stream coming out of the delta-sigma converter without going through the corresponding digital decimation to multi-bit PCM. This produces a 1-bit signal at 2.8224 MHz. By tapping into the bit stream at this stage, one avoids any possible degradation from the decimation process (and the required interpolation process at an oversampling D/A converter). One advantage that DSD has is that it has even higher bandwidth than 96 kHz PCM, and it can be converted to any standard format. Again, care must be taken in the conversion process to preserve as much of the original audio quality as possible.

This paper explores some aspects of these formats that relate to the implementation of systems for mastering in these formats.

BANDWIDTH AND STORAGE AT 96 KHZ

---

[1] DVD doesn't stand for anything.
[2] Note that the audio specifications referred to are part of the DVD video specification. The DVD-Audio specification is still under discussion.
[3] DSD is a registered trademark of the Sony Corporation.

A 24-bit, 96 kHz signal produces a data rate of 288 Kbytes/Second. For stereo, it is 576 kB/S. This translates into 34.56 Mbytes per minute of stereo. A typical 4G hard disk holds somewhat over 100 minutes of stereo at this rate. For stereo mastering of an 80-minute release, several 4G hard drives would be required for all the work material. Although this seems like a lot of storage for a single release, it is well within the bounds of current, affordable technology.

Transmission at such rates through a network burdens the network capacity. With the use of the MediaNet™ Digital Media Network with 100 MBit/Second off-the-shelf transport technology, somewhat over 40 channels of 24-bit 96 kHz audio can be exchanged in real-time. This is perfectly adequate for modest-sized workgroups with modest channel widths (say, up to 8 channels of audio at a time) and allows real-time access to any material on the network (that is, it is not necessary to copy files to local disks - they can be accessed through MediaNet wherever they are within the workgroup).

Since the data rate of the USP card used in the SonicStudio™ is about 2.4 Mbytes/Second, playback of up to 8 channels of 24-bit 96 kHz is theoretically possible on a single-card system[4]. Since the bandwidth for recording is somewhat less, up to 6 channels could be recorded at a time. Although the SonicStudio™ stretches the definition of a "portable" system, it could be used as a "field" recorder in some limited circumstances. At the time of this writing, there is no other way to record multi-channel, 24-bit, 96 kHz material. Increasing popularity of this format would assure a blossoming of methods to capture multi-channel audio and would give the audio community a variety of choices that would be compatible with the SonicStudio™.

INTERFACING AT 96 KHZ

At this time, there is no approved standard for digital interfacing at 96 kHz. Any scheme that we use must be considered as temporary, awaiting further standards. We nonetheless need some way to interface digital audio devices at 96 kHz to make the format usable today. The simplest scheme is to modify an existing standard, since there is a well-understood path to implementation.

One manufacturer of a 96 kHz DAT machine has chosen double-speed AES/EBU encoding. This is a logical choice, since AES/EBU is the most widely supported standard for digital audio interfacing. Unfortunately, most of the integrated circuits that are marketed for this format will not function at this clock rate. For most of the receivers, this involves an internal clock speed of 50 MHz which is beyond the capabilities of the phase-locked loops. For the parts with internal PLLs, there is no opportunity to modify the circuit, so there is no possibility of making it work at 96 kHz.

We have found one part that works reliably at 96 kHz, and that is the Motorola 56401. The analog portion of the part is specified with a bandwidth that extends to 50 MHz. Although the limits for the internal clock-doubler is not specified, it seems to operate at 50 MHz without problem over the entire temperature range and across multiple production lots. The part is not tested at the factory for double-speed operation, so special care must be taken to prevent surprises from process changes. Using the internal clock doubler and a 128x oscillator (12.288 MHz), the device receives and transmits double-speed AES/EBU with no apparent problem.

---

[4] Actual capacity may be less than this due to DSP code limitations and data routing limitations. The number given is based simply on the maximum observed SCSI data rate for the USP.

Note that there can be problems in the rest of the interfacing as well: many of the standard transformers that are used for the required isolation of the AES/EBU signal do not perform well at those frequencies. Some exhibit spurious resonances (“spurs”) or other electrical distortions of the signals. The driver and receiver chips must be carefully chosen as well, since there are several manufacturers of the RS-422 drivers and receivers, and the specifications are somewhat different from manufacturer to manufacturer. Care must be taken to assure that the parts have sufficient bandwidth to handle the required data rates and maintain a reasonable “eye-pattern.”

FILTERING AT 96 kHz

In general, the numerical properties of filters are similar at 96 kHz to the properties at 48 kHz. It is important nonetheless to be aware of the differences. Note that there are differences that are helpful in some cases and unhelpful in other cases.

To examine the behavior of filters at differing sampling rates, let us use a simple second-order section in canonical form, as shown in Figure 1.

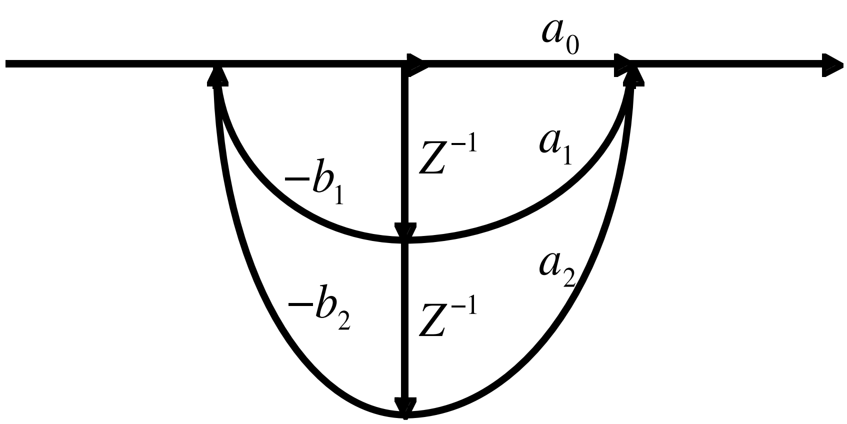


Figure 1: One form of a second-order digital filter

We will assume that the coefficients are implemented in either 24-bit integer or 32-bit IEEE floating point. At frequencies close to zero, the numerical properties of integer and floating point are identical. From the point of view of coefficient accuracy, there is no particular advantage of one over the other for low-frequency filters.

Note that there is an advantage to using integer arithmetic in general, in that most integer arithmetic units have very wide accumulators, such as 48 to 56 bits, whereas 32-bit floating point arithmetic units generally have only 24 or 32 bits of mantissa precision in the accumulator. This can lead to serious signal degradation, especially with low-frequency filters.

Let M be the magnitude of the largest representable integer, which would be hex 800000, or 8388608. The coefficient b_1 can be represented in polar coordinates as follows:

(1) $b_1 = -2r\cos\theta$

where θ is defined as follows:

(2) $\theta = 2\pi f / f_s$

where f is the resonant frequency of the filter and f_s is the sampling rate.

Let $\hat{b}_1$ be the quantized, scaled, 24-bit integer coefficient. Let k be an integer displacement from M. Let the root radius, r, be taken as unity for the purpose of this example. We may then write the scaled coefficient as follows:

(3) $\hat{b}_1 = -(M-k) = -M\cos(2\pi f / f_s)$

If we try to center the pole around, say, 60 Hz, we find that we cannot represent this frequency exactly, even using a floating-point coefficient. Table 1 summarizes the results for the closest integer representations of 60 Hz for both 48 kHz and 96 kHz:

$\hat{b}_1$	k	f	f_s
hex 800102	258	59.9159	48000
hex 800103	259	60.0319	48000
hex 800040	64	59.6831	96000
hex 800041	65	60.1476	96000

Table 1: Example 24-bit integer coefficients of second-order section.

This shows that we can get much more accurate frequencies at 48 kHz sampling rate than at 96 kHz sampling rate. One might wonder why k is about four times the magnitude at 48 kHz than its value at 96 kHz. This is explained simply by looking at the 2-term Taylor expansion of the cosine function:

(4) $\cos(\theta) \approx 1 - \frac{\theta^2}{2}$ (for small values of θ)

The presence of the second order term makes every halving of the sampling rate increase the precision by a factor of four at low frequencies.

The conclusion from this is that *the filter form shown in Figure 1 is not capable of realizing low frequency filters with sufficient precision for professional mastering at 96 kHz, either in 24-bit integer or in 32-bit floating point*. Note that this applies to the transposed form as well, since we are discussing coefficient accuracy, which is identical in all canonical forms of the second-order section. There are a number of other filter forms that are better suited for these purposes (see, for instance, [5][5]).

As noted above, there are some positive benefits on the numerical side from the use of 96kHz sampling rate. For example, if we use the C programs in [6] to design a presence filter, the response at zero and the Nyquist frequency is always constrained to be unity. For very high-frequency filters, this causes a skewing of the response and a gratuitous boost at lower frequencies. Figure 2 shows the responses of a presence filter with Q of 2, boost of 10 dB, and a center frequency of 18 kHz, realized at both 48 kHz and 96 kHz. Since the 48 kHz filter is constrained to be unity at 24 kHz, the response drops sharply above the resonant frequency, and the curve is somewhat elevated below the resonant frequency. With the 96 kHz sampling rate, the shape of the curve more closely resembles the analog equivalent, since the response extends well beyond 24 kHz.

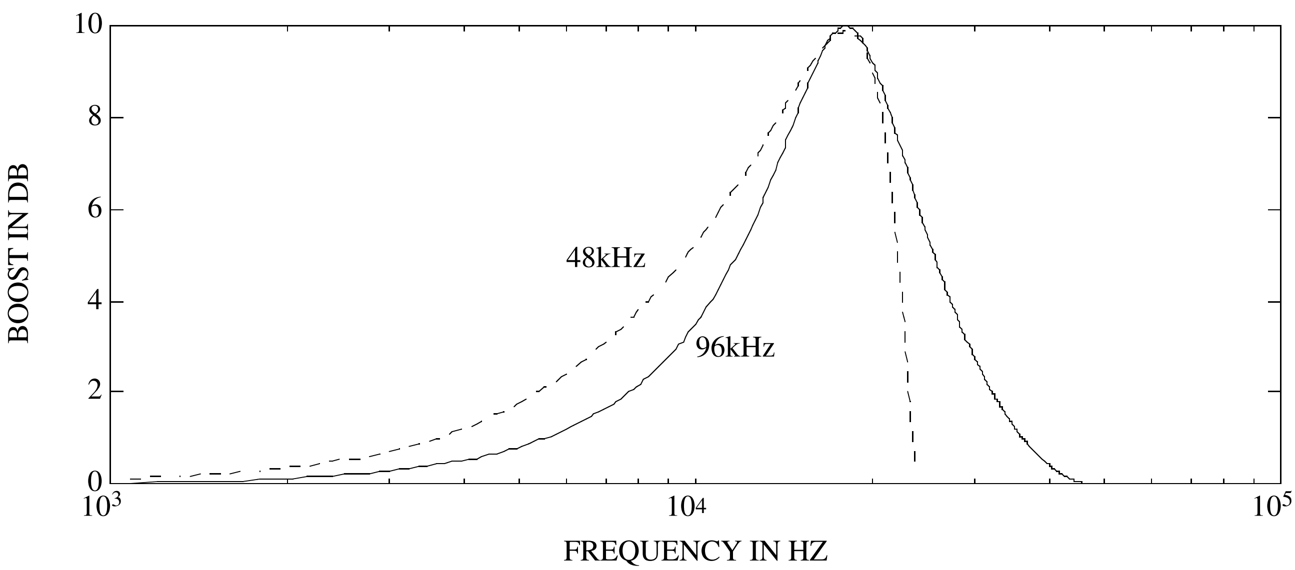


Figure 2: 18kHz presence filter at 48 kHz and 96 kHz sampling rates

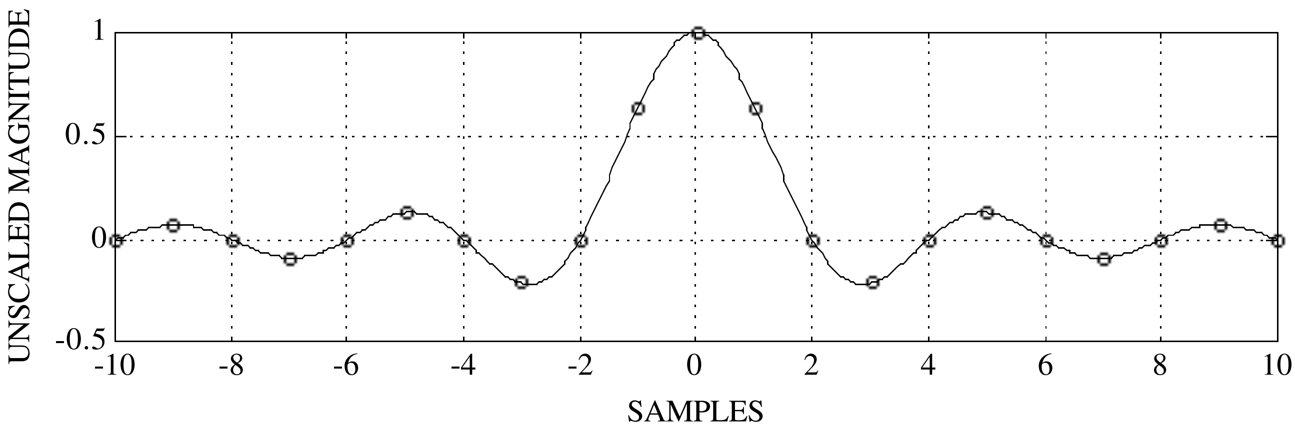
DOWNSAMPLING:

Although DVD-Video supports a 96 kHz, 24-bit release format, there will still be great demand for 44.1 kHz, 16-bit releases. Obviously, great care must be taken in the conversion process to assure that the advantages of mastering at 96 kHz are not lost in the conversion. The theory of sample-rate conversion and dithering is well-established [7, 8, 9], so what remains is some discussion of implementation.

The sample-rate conversion process starts with a low-pass filter to eliminate potential aliasing. This is usually done with a linear-phase FIR filter. Figure 3 shows a filter that downsamples from 96 kHz to 48 kHz. The circles represent the locations of the samples. The height of the circle is proportional to the filter coefficient at that part of the filter. Note that the filter shown was normalized to a magnitude of 1.0. A practical filter might be normalized differently, depending on the filter design method.

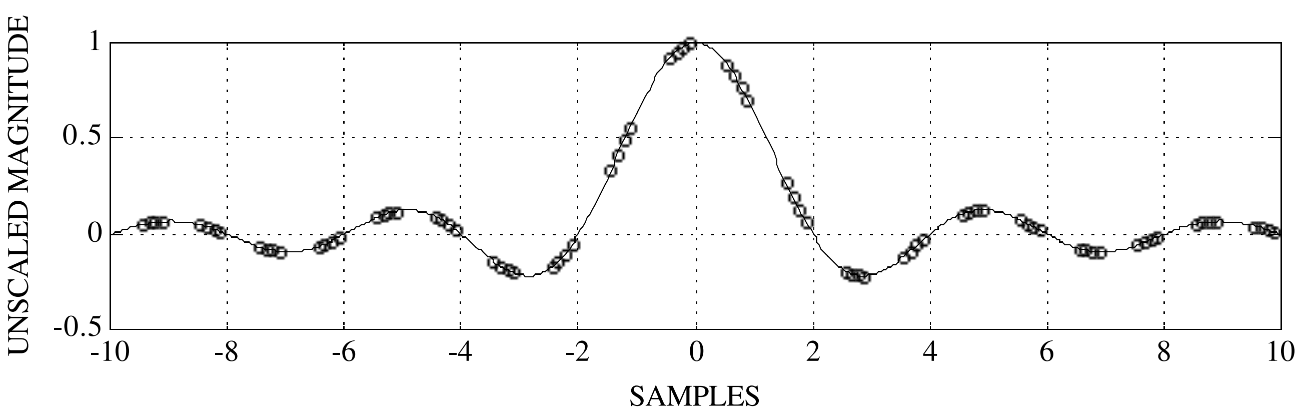
[5] The SonicStudio™ does not use either of these filter forms. It uses a proprietary, low-noise, low-sensitivity structure that does not exhibit the problems described above.

We can see several interesting things just from this picture. The first is that except for the center, the coefficient at the even numbered points is zero, so no multiply need be performed on these points. Next is that for each output point, the filter coefficients are identical. In terms related to polyphase filtering, this means that there is only one phase in the polyphase filter. This means for 2:1 downsampling *that any inaccuracy in the filter coefficients expresses itself as a deviation of the frequency response from the desired response, but not in the production of spurious sidebands*. This is an extremely important point. Any down-sampling with a ratio that has 1 in the denominator will exhibit this behavior.



<u>Figure 3: Possible Downsampling Filter from 96 kHz to 48 kHz</u>

To demonstrate the issue with downsampling from 96 kHz to 44.1 kHz, Figure 4 shows the same FIR filter, but this time a polyphase filter with 147 phases (since 96000/44100 = 320/147). The circles represent four consecutive phases. The first circle of each group of four represents one phase. The second circle of each group of four represents the next consecutive phase, and so on.



<u>Figure 4: Possible Downsampling Filter from 96 kHz to 44.1 kHz</u>

This shows that unlike the 96->48 kHz conversion*, each output point has a different set of coefficients applied to the 96 kHz signal*. Well, not exactly, since there are only 147 phases and there are some symmetries, but nonetheless, any error in the filter coefficients produces not only a deviation in the frequency response but *it also produces harmonic distortion that is semi-periodic with a period of 147 samples*. 147 samples represents a frequency of 300 Hz. This says that the numerical requirements for a 96->44.1 kHz conversion are substantially more severe than for a 96->48 kHz conversion, since we are not talking simply about the frequency response of the filter, but the generation of semi-periodic noise (semi-periodic because the error does not necessarily repeat exactly with that period).

To analyze this, let us write the equation for downsampling:

$$(5) \qquad y(k) = \sum_i w((k)_d + di)\bar{x}(nk + i)$$

where the ratio of the sampling rates is n/d (which, in this case, is 147/320). $y(k)$ represents the output signal at 44.1 kHz. $\bar{x}(k)$ represents the input 96 kHz signal, upsampled by a factor of d by inserting $d-1$ zeros between each sample. $w()$ represents the anti-aliasing filter required for the downsampling operation. The operation $()_d$ represents modulo-*d*.

$w()$ represents the exact coefficients of the anti-aliasing filter. If we let $\widehat{w}()$ represent the quantized coefficients, then we can write the equation for the quantized result as follows:

$$(6) \qquad \widehat{y}(k) = \sum_i \widehat{w}((k)_d + di)\bar{x}(nk + i)$$

Correspondingly, we can define $\delta w()$ as the difference between the exact coefficients and the quantized coefficients, or $w() - \widehat{w}()$. We can then write the equation for the error resulting from the quantization of the coefficients as follows:

$$(7) \qquad \delta y(k) = \sum_i \delta w((k)_d + di)\bar{x}(nk + i)$$

The important thing to notice about this equation is that it is linear and time-varying. This means that we can not simply take the Z-transform of each side, since that is only well-defined in the non-time-varying case. We can simulate this formula and observe the resulting spectrum. Since it is linear, it is sufficient to show the simulation using a single sinusoid. The response to any other signal can then be determined by superposition. Figure 5 shows the spectrum of the result of downsampling a 1 kHz sinusoid using a filter with 257 coefficients in each phase (or 37779 (=147*257) points total). This is a longer filter than would normally be used, so this figure exaggerates the problem. We should take this figure to be a qualitative illustration and not a quantitative one.

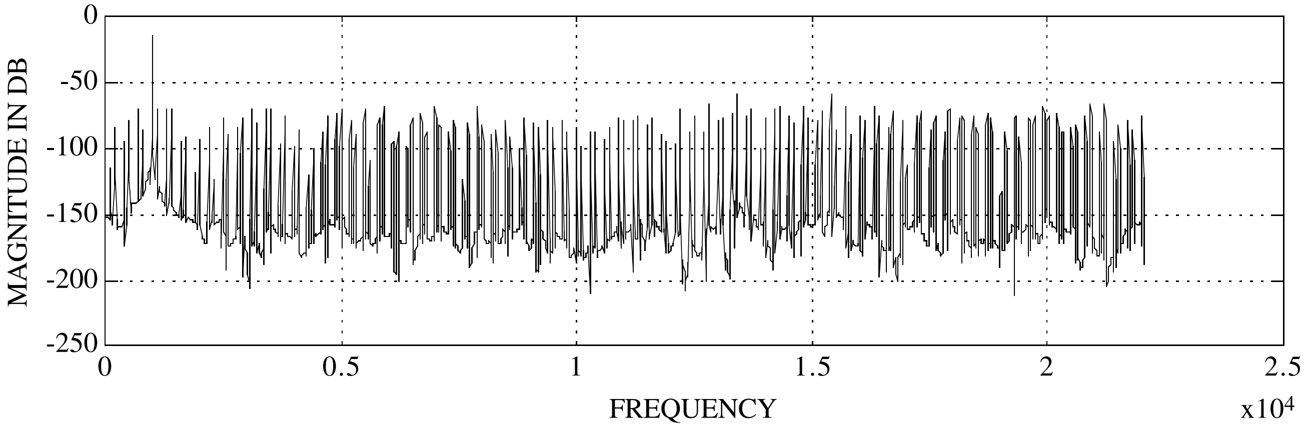


Figure 5: Error spectrum after downsampling from 96 kHz to 44.1 kHz

If we “zoom” in to the 1 kHz region, it is somewhat more clear what is happening:

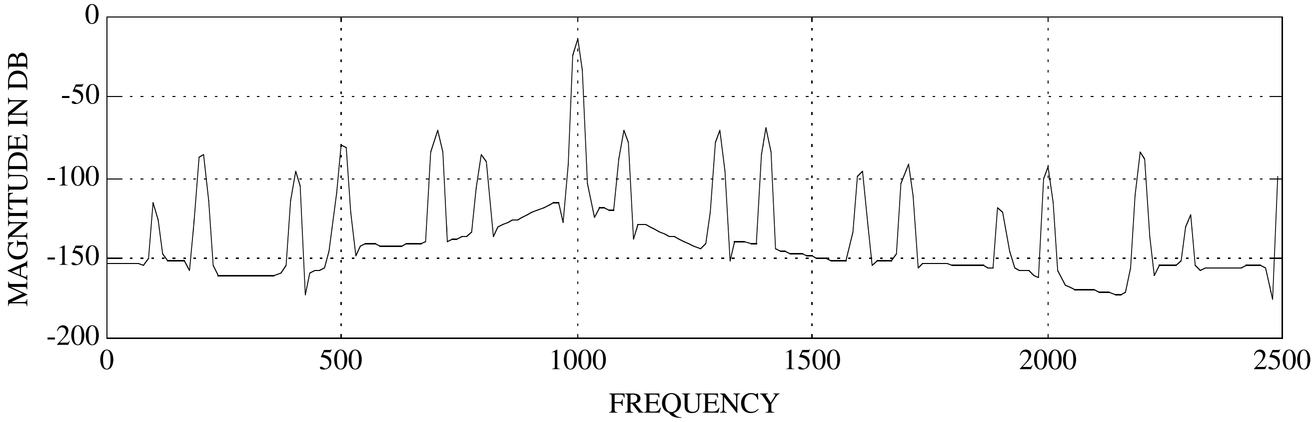


Figure 6: Error spectrum after downsampling from 96 kHz to 44.1 kHz

It is clear that the 1 kHz sinusoid is surrounded by sidebands spaced exactly 300 Hz apart (= 44100/147). The sideband peaks are “doubled” because there is the corresponding sinusoid at -1kHz (not shown) with its sidebands as well. This shows clearly that the error produced by the polyphase filter is qualitatively different from a non-polyphase filter in that sidebands are produced that are spaced by the frequency corresponding to the periodicity of the phases. With a non-polyphase filter (that is, a ratio of sampling rates that reduces to an integer with a denominator of one), the error consists entirely of aliased components, which can always be eliminated by increasing the order of the filter, and coloration of the sound due to inaccuracies of the transfer function of the filter in the audio band. There can be no additional sideband production.

For instance, under worst-case roundoff error, the error could be equal to the number of non-zero multiplications in the filter. With a relatively long filter, such as 60 or 70 multiplies, the worst-case error could be a 6-bit number. If this is done with 24-bit integer arithmetic, or with 32-bit floating point, there might only be 18 significant bits remaining. Note that the expected error, assuming that the roundoff is random with Poisson distribution, will be much smaller - on the order of 3 or 4 bits.

The lesson here is that the phases of the filter must be perfectly matched (that is, they must sum to the exact same number to prevent amplitude modulation), and they must be of sufficient length to minimize error. Note that by the above argument, *24-bit integer coefficients or 32-bit floating point coefficients are not sufficient to produce a 20-bit result*. For best results, longer coefficients must be used, such as 32 significant bits or more. Note also that the computation of the filter coefficients before quantization must be done at even higher precision, such as 64-bit floating point. As noted, the sum of the coefficients of each phase must give the same result. This would be the response at zero frequency, and it must be a constant. We can use the DC response as a kind of measure of the mismatch of the various phases.

Note that since equations (5), (6), and (7) are all linear, it is possible to down-sample the signal using the high-order bits of the coefficients, then compute another down-sampled sequence using the low-order bits of the coefficients as the down-sampling filter. The sum of the two sequences will then represent a downsampled sequence with much greater coefficient accuracy.

There are problems even producing the filter coefficients with sufficient precision. Filters that are produced by optimization must have very conservative stop-band ripple and must operate in high precision (double or long double). In producing a filter by windowing, care must be taken to assure that the transcendental functions used (such as $\sin()$, $\sinh()$, or $I_0()$) are evaluated to a sufficient number of significant bits to avoid error in the resulting coefficients. As an example, the standard approximation for $I_0()$ (equation 9.8.1 in [10]) is good to 0.00000016. Although this seems pretty accurate at first glance, it is only a factor of 8 smaller than the least-significant bit of a 20-bit word. If this approximation is used to generate a function for downsampling (it is part of the common Kaiser $I_0 / \sinh$ window [11]), then the resulting filter will not be accurate enough for a 20-bit result (one should use the series expansion of equation 9.6.12 in [10] and simply take enough terms to get the desired precision. Needless to say, this must be done with careful attention to the numerics of the calculation. For instance, it should be summed starting with the final term first and working backwards to the first term to avoid losing the contribution of the final terms, which will be very small).

If sufficient care is taken with the numerical issues there will be no difference between the conversion from 88.2 kHz to 44.1 kHz and a conversion from 96 kHz to 44.1 kHz.

Of course, if the result must be reduced to, say, 16 bits for CD release, some form of dithering should be used to preserve as much of the dynamic range as possible in the most audible frequency range [8, 9].

BANDWIDTH AND STORAGE OF DSD

A 1-bit signal at 64 times the sampling rate can be thought of as 4 16-bit words. Thus DSD requires four times the data rate and storage capacity of 16-bit, 44.1 kHz audio. One channel of DSD requires 352,800 bytes/Second. One minute of stereo DSD requires 42.336 Mbytes of digital storage. A typical 4G hard disk holds somewhat over 90 minutes of stereo at this rate. With MediaNet, about 35 channels of DSD can be exchanged in real-time over the network.

With the USP card in the SonicStudio, playback of up to 6 channels of DSD is theoretically possible on a single-card system. The recording bandwidth would be up to about 4 channels of DSD.

These figures are derived by using the data rate of a single DSD stream. They do not represent the entire picture in a practical setting. It is more reasonable to carry along a 24-bit, 44.1 kHz PCM stream with each channel of DSD. This provides waveforms for visualization of the signal, and a reasonable "proxy" which can be easily scrubbed and played at varying rates. Of course, final edits must be done to the DSD stream itself so the finished product can be auditioned.

A more realistic rate for 1 channel of DSD, then, is 5.5 times the rate and storage capacity of 16-bit, 44.1 kHz audio. One channel of DSD then requires 485,100 bytes/Second. One minute of stereo DSD, plus PCM proxy, requires 58.212 Mbytes of digital storage. A typical 4G hard disk holds somewhat over 65 minutes of stereo at this rate. With MediaNet, about 25 channels of DSD can be exchanged in real-time.

With the USP card in the SonicStudio, playback of 4 channels of DSD is possible on a single-card system. The recording bandwidth is also about 4 channels (because the playback is really about 4.9 channels, that residual 0.9 channels is enough for the extra recording overhead).

EDITING OF DSD

As noted above, a proxy can be used to view the waveform, locate edit points, and quickly audition a possible edit. To understand what it means to edit DSD, we must explore further the properties of DSD.

DSD is, in a certain sense, a PCM signal. It is perfectly analogous to 16-bit, 44.1 kHz audio. The 1-bit signal can be thought of as representing +1 and -1 (rather than 0 and 1). To change the gain of a DSD signal, you multiply it by the desired factor, just like with any PCM signal. The difference comes when you try to convert the resulting signal back to DSD: if you multiply the DSD signal by, say, a 16-bit gain, you end up with a sequence of 16-bit numbers at a sampling rate of 2.8224 MHz. Clearly, you then need to convert this back to a 1-bit signal. This is done by imitating what the delta-sigma converter itself does. You need a quantization step and a noise-shaping filter [12]. Figure 7 shows one form of a noise-shaped quantizer that will produce DSD output from a multi-bit (*i.e.,* processed) input [13]. This is similar to a noise-shaped dithering algorithm, except that the quantizer having only 1 bit means that it is always operating in saturation. This makes it much more difficult to analyze the behavior of such a circuit directly.

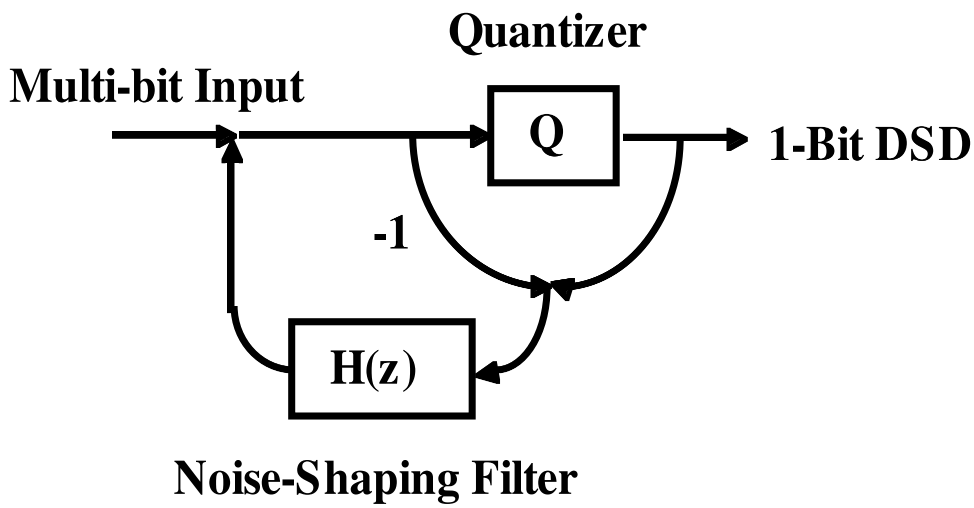


Figure 7: Quantizer with Noise Shaper

The Z-transform of the output of this circuit is $X(z)+(1-H(z))Q(z)$, where $X(z)$ is the Z-transform of the multi-bit input sequence and $Q(z)$ is the Z-transform of the error signal added by the quantizer. We can see from this that the input signal is passed unchanged but for the addition of a filtered version of the quantization noise. The noise-shaping filter can be designed to reduce the quantization noise in the audio band to a very low level.

Note that the traditional delta-sigma converter has the noise-shaping filter in the signal path, meaning that the signal is filtered by the noise shaper along with the quantization noise. There are significant stability considerations that must be addressed in this form. Either form is acceptable and can produce good results. With the filter in the direct signal path, extreme care must be taken to assure that the filter does not produce any coloration throughout the audio band. Ripple of .01 dB or less should be enforced across the audio band, and phase distortion must be negligible.

It is useful to examine the practicality of implementing such a quantizer. A reasonable feedback filter requires a number of multiplications. We may choose a 5th-order filter [12], which would involve at least 11 multiplications. On, say, a 66 MHz 56002, at 2.8224 MHz, there would only be 11 instructions *at most* to perform the operation. We must conclude that *it is not practical to process DSD signals in real-time on a general-purpose DSP chip. If real-time is a requirement, then special-purpose hardware must be built*. Note that we can, for instance, work with the proxy in real-time, then “auto-conform” DSD by performing the same calculations on the 1-bit stream, producing, say, a 24-bit sequence as intermediate data, then re-quantizing to 1-bit as the final stage.

As noted above, real-time processing of DSD signals requires special hardware. Special-purpose hardware has been created specifically for doing real-time edits directly with DSD [14]. An experimental DSD editing board for the SonicStudio was designed and built by Sony that employs a custom signal processing chip that performs cross-fade edits and gain adjustments with requantization in real-time. This is a demonstration that it is practical to edit DSD directly in real-time using today's technology.

IIR FILTERING OF DSD

Equalization of DSD is even more problematic. If we try to do the analysis of Table 1 above, we come to the conclusion that using the direct form of Figure 1, even with 32-bit coefficients, we cannot represent, say, a 60 Hz filter at all, much less with professional precision. As an example, let us look at the first non-zero frequency that can be represented by various bit widths. That is, we will compute the resonant frequency, f, of a filter running at a 2.8224 MHz sampling rate, for a value of $\hat{b}_1 = -(M-k)$ for $k = 1$.

# bits	$\hat{b}_1$	f
24	hex 800001	219.3354
32	hex 80000001	13.708463
40	hex 8000000001	0.856779
48	hex 800000000001	0.0535487

Table 2: Minimum frequency representable at 2.8224 MHz sampling rate.

From this analysis, we can conclude that the filter form shown in Figure 1 is not practical for filtering DSD. Of course, if one has access to, say, double-precision IEEE floating point, then the situation is different. Also, as noted above, other filter forms reduce the precision required for the coefficients. Note that these results differ from the opinions of the authors in [13], who conclude that 32-bit precision is sufficient. I think this difference is due simply to the difference in viewpoint between a university research group and a commercial tool supplier for the professional audio industry[6].

There is another solution to the coefficient accuracy issue, and that is to use multiple-sample delays rather than the single-sample delays. Figure 8 shows one possible filter form with multi-sample delays[7].

---

[6] Having spent 11 years at Stanford University as programmer, student, and research professor, I feel uniquely qualified to comment on the difference between the university research point of view and the commercial tool-provider's point of view.
[7] This idea is not original with the author, but I can not find an original reference to this as a technique for reducing required coefficient precision.

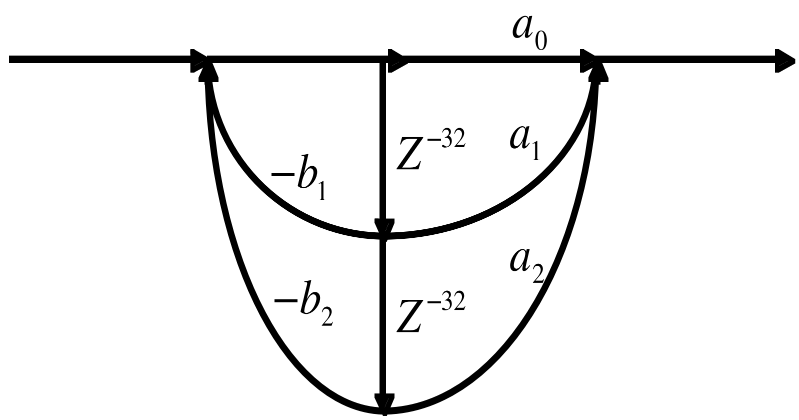


Figure 8: Possible second-order digital filter for DSD

The effect of the multi-sample delays is to replicate the frequency response from 0-44100 Hz in the range 44100-88200, 88200-132300, 132300-176400, and so on. There will be a total of 32 copies of the frequency response. Figure 9 shows the response of the 18 kHz presence filter used in the examples above. The response above about 700 kHz is not shown.

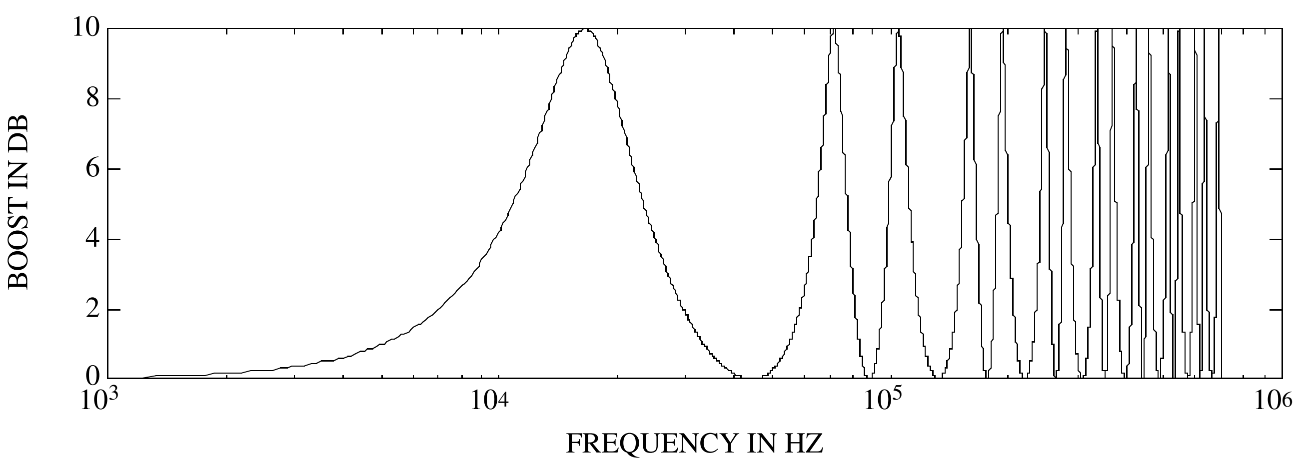


Figure 9: Response of 18 kHz Presence Filter with 32-sample Delays

When this signal is then re-quantized to get 1-bit DSD, the signal above the audio band becomes part of the “noise” that is shaped by the noise-shaping filter, so the duplicated responses will not have any effect in the audio band. Note that care must be taken so that the noise-shaping filter itself does not saturate when the signal amplitude is increased, as shown above. A poorly-designed noise-shaping filter can cause a substantial increase in the noise floor in the audio band when presented with this signal.

Note that the *numerical properties of the DSD filter with 32-sample delays are identical to the numerical properties of a filter with unit delays operating at 88.2 kHz*, and so all the considerations noted above about the numerical behavior of the second-order section apply without modification to the filter of figure 8.

FIR FILTERING OF DSD

One might think that since the DSD signal consists entirely of +1 and -1 some advantage could be gained by the use of FIR filters, since the operations would be reduced entirely to additions. The problem is in the length of the filter required. We can get a rough idea of the filter length by looking at the spacing of the frequency bins of an FFT of the given length. The following table gives some selected lengths at various sampling rates.

LENGTH	48,000 Hz	96,000 Hz	2,822,400 Hz
4,096	11.71875 Hz	23.4375 Hz	689.0625 Hz
32,768	1.464844 Hz	2.9296875 Hz	86.1328 Hz
262,144	.1831055 Hz	.3662109 Hz	10.7666 Hz
2,097,152	.022888 Hz	.04577637 Hz	1.345825 Hz

Table 3: Frequency Bin Separation for FIR Filters.

Thus we see that to get frequency resolution for arbitrary FIR filters, we end up with millions of coefficients. Even though all the computations can be made with adds and subtracts, this is still not terribly practical.

As with editing and gain adjust, special hardware will be required (at this time) to perform equalization on DSD signals in real-time. Note that all operations could be previewed using a 24-bit, 44.1 kHz proxy, and then the DSD could be "auto-conformed" out of real-time.

NON-UNIQUENESS OF DSD

One curious aspect of DSD is that it is not unique. Two DSD signals can have identical components in the audio band and still appear completely different. For example, we can represent zero by the sequence +1, -1, +1, -1, . . . or by the sequence -1, +1, -1, +1, . . . Both these sequences have only one component at 1/2 the sampling rate, and that component is 180° out of phase in the two examples. The signal in the audio band is exactly zero. This has a number of significant implications. For example, after re-quantizing following a cross-fade, unless we do something special, even when the cross-fade is done and no further change to the signal is applied, the sequence coming out of the re-quantizing operation will be different from the one going into the re-quantizing operation. For instance, with the quantizer shown in Figure 7, the exact output will depend on the state of the filter, which depends not only on the input sequence, but on the resulting output sequence.

It is desirable to be able to return to the original sequence after modification of the signal is finished, such as after a fade. One solution to the issue of returning to the original sequence is embodied in the special-purpose DSD hardware mentioned above.

DOWNSAMPLING OF DSD:

The problems of downsampling are made worse by DSD. For conversion to 44.1 kHz, the 64:1 downsampling does have the property that limited coefficient accuracy only means distortion in the frequency response of the downsampling filter and thus the possibility of aliasing, but it cannot create harmonic distortion or raise the noise in the audio band (except for possible aliased components). If we require the transition band of the filter to go from 20,000 Hz to 24,100 Hz, this is a transition ratio of 4,200/2,822,400 = 0.0014881. If we take as a rule of thumb, say, 30 multiplies divided by the transition width for a professional-quality filter, we can conclude that the filter itself would have to be more than 20,000 points in length. This results in over 500 *billion* additions per second of sound. It will be sometime before our desktop computers are capable of this rate of calculation in real time. In fact, using a 66 MHz 56002 as a benchmark, it would require about 30 minutes *minimum* for each second of sound to perform the calculation. For an 80-minute album, this amounts to a *minimum* of about 40 hours. Note also that a 20,000 point filter would require more than 24 bits for the coefficients, since the deviation of the frequency response by truncating the coefficients to 24 bits might well introduce either coloration of the sound in the audio band or objectionable aliasing components.

This problem concerned the engineers at Sony. As a result, an experimental real-time down conversion processor was built using a special 1-bit convolution circuit and fixed point digital signal processors (DSP). A 32767-tap convolution is processed by 8 paralleled 4096-tap convolution processor cards per channel. The processor card outputs are added and then processed through extended precision Super Bit Mapping (SBM®[8]) on a DSP card. Via a digital I/O card, word sync and the DSD signal are input and the 44.1kHz/16bit signal is output by SDIF-2 format. This demonstrates that the difficult task of preserving the quality of DSD against the numerical problems inherent in the format is possible in real-time using today's technology.

CONCLUSIONS:

Mastering at 88.2 kHz or at 96 kHz is possible using today's technology for storage, transmission, networking, processing and editing. Although the storage requirements are significantly greater than when working at lower sampling rates, they are easily within the bounds of modern digital audio equipment.

There are a number of considerations both in the numerical aspects equalization and in downsampling that must be taken into account to assure a professional-quality result. Care must be taken in the filter design for equalization to avoid problems of coefficient error and roundoff-noise buildup. Sufficient precision must be used in downsampling filters to assure that no audible coloration of the audio band is introduced. The conversion from 96 kHz to 44.1 kHz requires the most careful implementation, since coefficient error in the downsampling filter can produce harmonic distortion as well as coloration.

It is possible with today's equipment to record and playback DSD. Modern storage and networking techniques are easily adequate to deal with production quantities of DSD signals. Real-time editing, fading, mixing, and equalization of DSD requires special-purpose hardware, since it is beyond the capabilities of general-purpose DSPs (at the time of this writing). Similarly, downsampling at anywhere near real-time rates requires computation capacity and bit widths that are not available on general-purpose DSPs today, but are available in special-purpose hardware. It is practical today, using hardware

[8] SBM is a registered trademark of the Sony Corporation.

designed specifically to process DSD, to record and edit a DSD master. It can then be converted at high quality to any release format that exists today.

<u>REFERENCES:</u>

[1] "DVD Video Book, Version 0.9 (Draft)" DVD Products Division, Toshiba Corporation, April 1996 "A New CD Mastering Process Using Direct Stream Digital" by Ayataka Nishio, Makoto Akune, Gen Ichimura, Yasuhiro Ogura, Yuji Tsuchida

[2] Y. Yamazaki, "High-speed 1-bit Signal Processing Considering Spectrum of Quantization Noise," AES 5th Regional Convention, Tokyo, July, 1991.

[3] Y. Yamazaki, and H. Ohta, "Control of Quantization Noise Spectrum with High-speed 1-bit Coding System," AES 6th Regional Convention, Tokyo, June, 1993.

[4] A. Nishio, G. Ichimura, Y. Inazawa , N. Horikawa, and T. Suzuki "Direct Stream Digital Audio System" Proceedings of the AES 100th Convention, Copenhagen, Denmark, May 1996

[5] R. C., Agarwal, and C. S. Burrus, "New Recursive Digital Filter Structures Having Very Low Sensitivity and Roundoff Noise" IEEE Trans. Circuits and Systems, CAS-22, #12, pp 921-927, December 1975.

[6] J. A. Moorer, "The Manifold Joys of Conformal Mapping," J. Audio Eng. Soc., Vol. 31, #11, pp826-841, 1983 November

[7] R.W. Schafer and L.R. Rabiner, "A Digital Signal Processing Approach to Interpolation," Proc. IEEE, Vol. 61, pp692-702, June 1973

[8] R.A. Wannamaker, "Psychoacoustically Optimal Noise Shaping," J. Audio Eng. Soc., Vol. 40, 1992 July/Aug., pp611-620

[9] S.P. Lipshitz, J. Vanderkooy and R.A. Wannamaker, "Minimally Audible Noise Shaping," J. Audio Eng. Soc., Vol 39, 1991 Nov., pp836-852

[10] M. Abramowitz and I.A. Stegun, *Handbook of Mathematical Functions*, Dover Publications, New York, 1965

[11] J.F. Kaiser, "Digital Filters," in *System Analysis by Digital Computer*, edited by F.F. Kuo and J.F. Kaiser, John Wiley & Sons, New York, 1966, pp232-233

[12] R.W. Adams, P.F. Ferguson, Jr., A. Ganesan, S. Vincelette, A. Volpe, and R. Libert "Theory and Practical Implementation of a Fifth-Order Sigma-Delta A/D Converter" J. Audio Eng. Soc., Vol. 39, No. 7/8, 1991 July/August, pp515-528

[13] N.M. Casey and J.A.S. Angus, “One-Bit Digital Processing of Audio Signals,” Audio Eng. Soc. 95th Convention, 7-10 October 1993, New York, preprint #3717

[14] A. Nishio, M. Akune, G. Ichimura, Y. Ogura, and Y. Tsuchida, "A New CD Mastering Processing Using Direct Stream Digital" Audio Eng. Soc. 101st Convention, 8-11 November 1993, Los Angeles