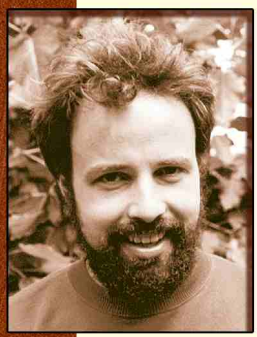
Audio precision®

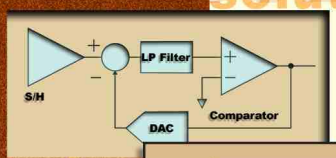
Testing For Optimal Results

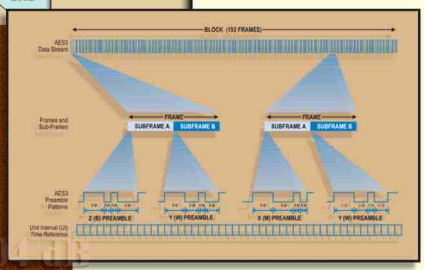
Audio Precision
Application Note #5:

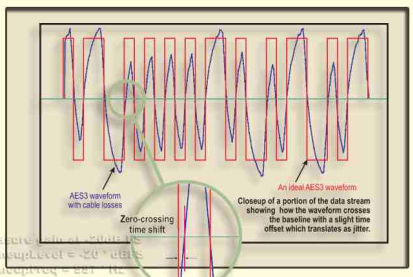
Measurement Techniques for Digital Audio

By Julian Dunn









$R_{thd} = 20\log_{10}(V_n/V) - 10.4$ dB

Measurement Techniques for Digital Audio

by Julian Dunn

8211.0143 Revision 1

Published by:

Audio Precision, Inc.
5750 SW Arctic Drive
Beaverton, Oregon 97005
U.S. Toll Free: 1-800-231-7350
Tel: (503) 627-0832 Fax: (503) 641-8906
email: info@audioprecision.com
web: audioprecision.com

Printed in the United States of America

IV0216084317

Contents

Analog-to-Digital Converter Measurements

Digital-to-Analog Converter Measurements

The Digital Interface

Introduction

Much has been written about digital audio, its defining standards, the ever-changing hardware and software, the various applications in recording and broadcasting and telecommunications and the audibility of this or that configuration or artifact. In this book the late Julian Dunn focused instead on the *measurement* of digital audio signals, and examined in great detail techniques to evaluate the performance of the converters and interface through which the audio passes. Mr. Dunn passed away early in 2003, cutting much too short a brilliant career as one of the world's premier designers and consultants in digital audio.

Chapter One, **Jittery Theory,** studies the causes and effects of the interface timing variations called *jitter* with a number of tests designed to characterize this pervasive malady.

Chapter Two, **Analog-to-Digital Converter Measurements,** looks at key ADC parameters and behavior and includes 15 AP Basic macros to run the necessary tests.

Chapter Three, **Digital-to-Analog Converter Measurements,** does the same for DACs. A sidebar looks at dither. Twenty-five macros are included.

Chapter Four, **The Digital Interface,** discusses the AES3/IES60958 digital interface, examining the basic format and the means of characterizing the signal. Sidebars focus on the international standards and on synchronization considerations.

The macros and tests used in making the measurements discussed in the two converter chapters are listed at the end of the chapters. With the tests and macros on the CD-ROM you'll find two AP Basic menus (a-d menu.apb and d-a menu.apb) to make running the macros easy, along with detailed notes are in the file README.DOC. Note that all these files must be copied to a local folder on your computer to run properly. The tests and macros were written with the Audio Precision System Two Cascade in mind, and although the con-

cepts and techniques are portable the tests would need to be re-written for other instruments.

Check the Audio Precision Web site at audioprecision.com for additional related material, tests, macros and other solutions which may be developed from time to time. We are interested in your comments and suggestions; contact us at techsupport@audioprecision.com.

Jitter Theory

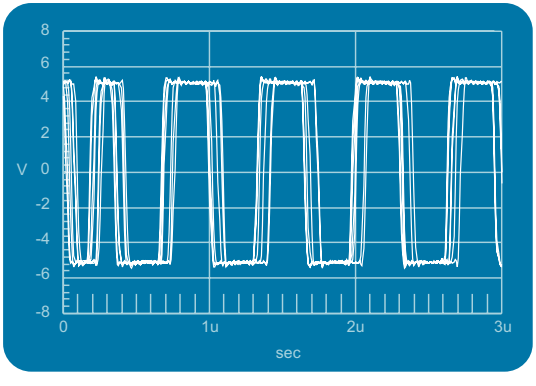
Introduction

Digital audio systems are unlike analog audio systems in two fundamental respects:

- The signal, in its analog state a continuously variable voltage or current, is represented digitally by a limited number of discrete numerical values.
- These numerical values represent the signal only at specific points in time, or *sampling instants,* rather than continuously at every moment in time.

Sampling instants are determined by various devices. The most common are the *analog-to-digital converter* (ADC) and the *digital-to-analog converter* (DAC) which interface between the digital and analog representations of the same signal. These devices will often have *sample clock* to control their *sampling rate* or *sampling frequency.*

Figure 1.
Jittered AES3 waveform.



Sampling instants can also be determined by a *sample rate converter* (SRC) that uses numerical processes to convert a digital signal at one sampling fre-

quency to a digital signal at another sampling frequency. An SRC might not have a physical sample clock at all, but in the numerical process of regenerating signal samples to correspond with new sampling instants is considered to use a *virtual sample clock*.

Digital audio is often thought to be immune to the many plagues of analog recording and transmission: distortion, line noise, tape hiss, flutter, crosstalk; and if not immune, digital audio is certainly highly resistant to most of these maladies. But when practicalities such as oscillator instability, cable losses or noise pickup do intrude, they often affect the digital signal in the time domain as *jitter*.

This jitter can be on the interface carrying the digital signal. Interface jitter can result in data errors or loss of lock, which represent fault conditions; it can also be coupled into equipment to produce jitter in a sample clock, the effect being a (normally) subtle reduction in the accuracy of the sampling process.

What Is Jitter?

Jitter is the variation in the time of an event—such as a regular clock signal—*from nominal.*

For example, the jitter on a regular clock signal is the difference between the actual pulse transition times of the real clock and the transition times that would have occurred had the clock been ideal, that is to say, perfectly regular.

Against this nominal reference, the zero-crossing transitions of many of the pulses in a jittered data stream are seen to vary in time from the ideal clock timing. Expressed another way, jitter is phase modulation of the digital interface signal.

The jitter component can be extracted from the clock or digital interface signal to be analyzed as a signal in its own right. Among the more useful ways of characterizing jitter is by examining its frequency spectrum and identifying the significant frequency components of the jitter itself.

Measuring Jitter

When very little jitter is present, the pulse transitions are moved back or forth by only small measures of time. When the jitter is increased, the transitions move across a larger range of times.

Jitter amplitude, then, is a measure of time displacement and is expressed in units of time, either as fractions of a second or *unit intervals*. For those new to jitter measurement, this can lead to some disconcerting graph labels, with time on the vertical axis versus time on the horizontal axis, for example.

Jitter frequency is the rate at which this phase-shifting is taking place. Like other noise or interference signals, the jitter modulation signal can be a pure

and regular sine wave, a complex waveform or have a completely random character.

The Unit Interval

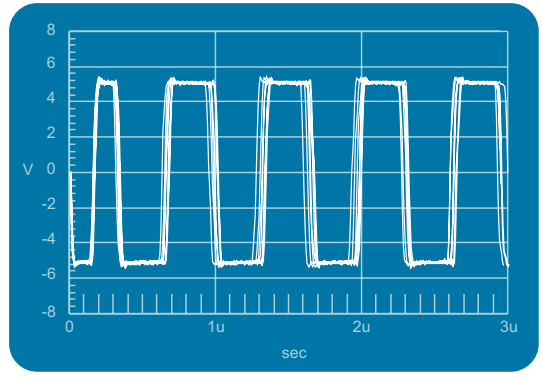
The *unit interval* (UI) is a measure of time that scales with the interface data rate, and is often a convenient term for interface jitter discussions. The UI is defined as the *shortest nominal time interval in the coding scheme.* For an AES3 signal at a 48 kHz frame rate, there are 32 bits per subframe and 64 bits per frame, giving a nominal 128 pulses per frame in the channel after bi-phase mark encoding is applied. So, in this case:

$$1\,\mathrm{UI} / (128 \times 48000) = 163\,\mathrm{ns}$$

The UI is used for several of the jitter specifications in AES3[1] (the Audio Engineering Society's standard for interfacing two-channel linear digital audio), with the result that the specifications scale appropriately with the data and frame rate. As an example, the dimensions in UI for 96 kHz frame rates are exactly half the size, in seconds, as the dimensions in UI for frame rates of 48 kHz. This scaling matches the scaling of the capabilities and requirements of the receivers and transmitters on the interface.

Note: Some specifications in data transmission define the unit interval as the duration of one bit transmission. This produces results incompatible with the AES3 specification and is not used here.

Figure 2. Interval variations on an oscilloscope. Not a valid way to view jitter.



How Can You See Jitter?

Jitter on a digital signal can be observed as pulse transitions that occur slightly before or after the transitions of an ideal clock. Any meaningful measurement, then, must involve a comparison between the jittered signal and an ideal clock.

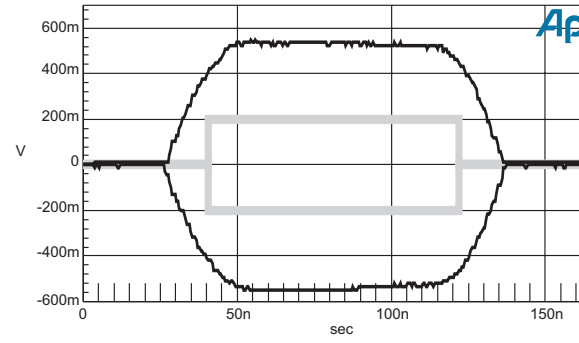


Figure 3. APWIN DSP eye pattern. The black line is the eye formed by the interface signal; the gray rectangle represents the opening that satisfies the minimum input characteristics specified in AES3.

In practice, there are often no ideal clocks to compare with, and real jitter measurements must be *self-referenced*—made relative to the signal itself.

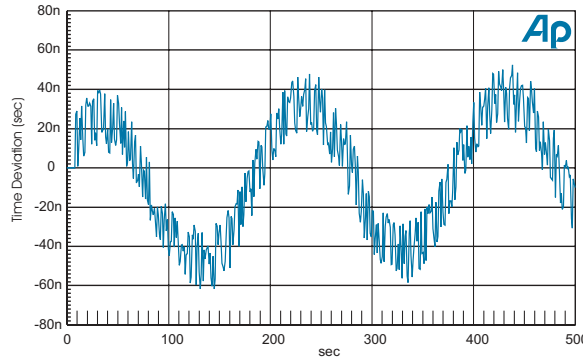
The simplest and most misleading self-referenced technique is "looking at the waveform on an oscilloscope," triggering the oscilloscope on the jittered signal as shown in Figure 2. Unfortunately, you will get deceptive results that depend on the interval between the oscilloscope trigger and the transition being examined, and also on the frequency spectrum of the jitter. Rather than jitter, this technique displays *interval variations.* There is a relationship between the two, but at some frequencies jitter will not be shown at all, and at others the jitter amplitude will appear doubled. In particular, this approach is very insensitive to low-frequency jitter.

Instead, the ideal clock can be simulated by phase-locking a relatively low-jitter oscillator to the jittered signal or real clock, using a phase-locked loop (PLL). (A sidebar on phase-locked loop characteristics is on page 9.) This self-referencing technique will have a high-pass characteristic with a corner frequency that is related to the PLL corner frequency. The PLL provides an ideal clock signal useful as an oscilloscope external trigger or as a reference signal in dual-trace oscilloscope viewing, for example.

If an oscilloscope is triggered by the PLL reference clock and the scope time base is set to the duration of about one UI, a great many sequential pulses will be shown at once, all stacked on top of one another due to the persistence of the screen phosphors. This distinctive display is called an *eye pattern*, a version of which is shown in Figure 3. The opening in an eye pattern is narrowed by the time spread of the pulse transitions. A narrow eye, then, indicates jitter.

Using digital signal processing (DSP) techniques, a DSP analyzer can approximate the ideal clock reference by calculating the clock timing based on an averaging of the incoming signal. The DSP analyzer can then capture the signal (and its jitter) very accurately. From this data the analyzer can display the variation in timing and amplitude of the pulse stream as an eye pattern as in Figure 3; show the jitter waveform in the time domain as in Figure 4, or, us-

Figure 4. 5 kHz jitter vs. time.



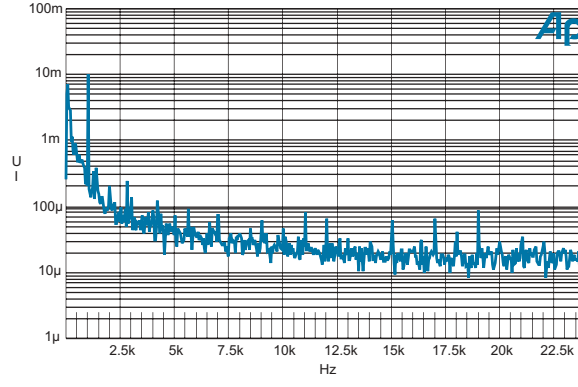
ing FFT spectrum analysis, plot the jitter in the frequency domain, as in Figure 5.

Jitter in Sampling Processes

Jitter can affect a digital audio signal in two broad realms: in the sampling process, and in the digital interface.

Sampling jitter is the term given to errors in the timing of the sampling processes of an ADC, a DAC or an SRC. Larger amounts of sampling jitter may cause an audible degradation to the signal. Sampling jitter is discussed in detail beginning on page 18.

Figure 5. FFT spectrum analysis of jitter signal.



Jitter in the Interface: Data Recovery

Quite apart from the gradual degradation that can result from jitter on sampling clocks, jitter is also an important characteristic to be controlled for reliable data communications. Jitter in digital audio interface signals should be kept within the range that can be tolerated by the data receiver; otherwise, the data may be corrupted. These levels are typically orders of magnitude larger

than the jitter levels that would cause concern in sampling clocks. *Interface jitter* is discussed in detail beginning on this page.

Jitter in Clock Recovery for Synchronization

In many digital audio applications it is important for the signals to be stored, transmitted, or processed together. This requires that the signals be time-aligned. In other applications it is important that the audio sample rate exactly matches a multiple of another rate, such as a video frame rate, so that the video and digital audio signals may be encoded, stored or transmitted together. The action of controlling timing in this way is called *clock synchronization.*

When a clock is synchronized from an external "sync" source, jitter can be coupled from the sampling jitter of the sync source clock. It can also be introduced in the sync interface. Fortunately, it is possible to filter out sync jitter while maintaining the underlying synchronization. The resulting system imposes the characteristics of a low-pass filter on the jitter, resulting in jitter attenuation above the filter corner frequency.

When sample timing is derived from an external synchronization source in this way, the jitter attenuation properties of the sync systems become important for the quality of the audio signal. There are other circumstances where this is not so important.

Digital Interface Jitter

Interface jitter occurs as digital signals are passed from one device to another, where jitter can be introduced, amplified, accumulated and attenuated, depending on the characteristics of the devices in the signal chain. Jitter in data transmitters and receivers, line losses in cabling, and noise and other spurious signals can all cause jitter and degrade the interface signal.

The AES3 digital audio interface format[1] now has specifications for jitter. (The consumer version of the interface, which is described in IEC60958-3:2000[2] also has jitter specifications.) This specification was drawn up to resolve problems that would occur when units that conformed to the interface specification were interconnected and yet the interface did not work reliably.[3]

Intrinsic Jitter

If a unit is either free-running or synchronized with a relatively jitter-free signal, then any output jitter measured at the transmitter is due to the device itself. This is referred to as *intrinsic jitter.*

The level of intrinsic jitter is mainly determined by two characteristics: the phase noise of the oscillator in the clock circuit and, for an externally synchronized device, the characteristics of the clock recovery PLL.

Phase-Locked Loop Characteristics

A mechanical flywheel will slowly follow gradual speed changes but will largely ignore short-term fluctuations. This behavior is similar to that of a phase-locked loop (PLL). The lighter the flywheel the more rapidly it will follow changes and the "cut-off" or corner frequency is higher. The corner frequency of a PLL is determined by its feedback, or loop gain. This feedback falls with frequency, both as a result of the characteristics of the loop filter and from the integration of frequency into phase that is taking place before the phase detector output. At the corner frequency the gain around the loop is unity.

For jitter spectral components below the corner frequency, the negative feedback means that the PLL output will closely follow the PLL input, and the phase noise of the oscillator is attenuated. Above the corner frequency the feedback falls. This means that the jitter of the PLL output will be determined increasingly by the phase noise of the oscillator and less by the input jitter. A key element in the design of a transmitter or receiver PLL is this compromise between intrinsic jitter and jitter attenuation.

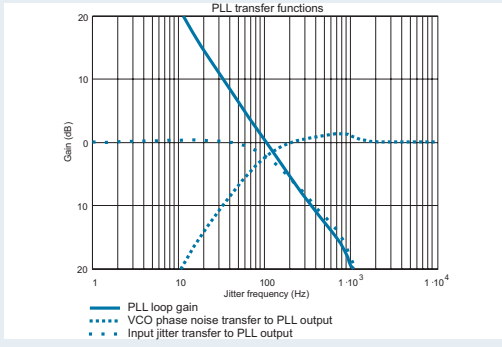


Figure 6. Phase lock loop transfer functions.

For example, consider the quartz clock oscillator in a CD player. Since it is free-running, any jitter at the output is due to the phase noise of the oscillator plus any digital logic delay jitter. Quartz oscillators have low phase noise and the high speed logic devices have very little delay jitter, so the jitter is low—often less than 1 ps rms for jitter frequencies above 700 Hz.

A device designed to lock to external signals with a range of sampling frequencies may have a voltage controlled oscillator (VCO) as a clock. VCOs generally have much higher phase noise than a quartz oscillator; free-running VCOs typically have levels of intrinsic jitter of more than 1 ns rms above 700 Hz. However, in a clock-recovery application the VCO would be within a phase-locked loop (see the sidebar above) in order to synchronize with the external reference, and the intrinsic jitter of the oscillator would be attenuated by the PLL.

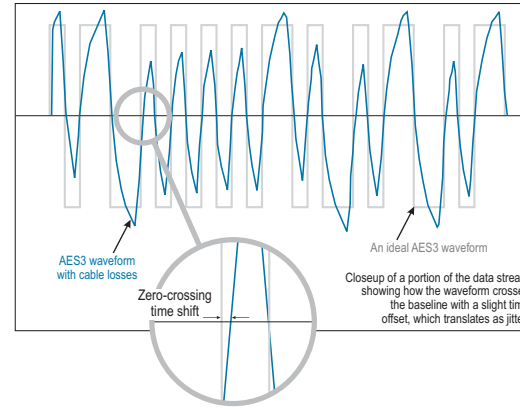
Intrinsic jitter often must be measured in situations where there is no low-jitter reference available, and so the measurements are self-referenced by locking a PLL to the clock signal recovered from the data stream. The characteristics

of this PLL will determine the low-frequency cut-off point of the measurement. AES3 specifies a standard response for this measurement with a 3 dB corner frequency of 700 Hz.

The intrinsic jitter levels in AES3 are specified as a peak measurement, rather than rms. This is because the authors were concerned with the maximum excursion of timing deviations—as it is these that would produce data errors.

Cable-Induced Jitter

Figure 7. AES3 ideal waveform with cable-affected waveform overlaid.



The other source of jitter on the digital interface is as a result of the non-ideal nature of the interconnection. Resistance in the cable or inconsistent impedance can cause high frequency losses which result in a smearing of the pulse transitions, as shown in Figure 7.

This would not be a serious problem if the effect were the same on every transition. That would just result in a small static delay to the signal which could be ignored. However, that would only be the case the pulse stream were perfectly regular—a string of embedded ones or zeros, for example. But real pulse streams consist of bit patterns which are changing from moment to moment, and in the presence of cable losses these give rise to *intersymbol interference*. The proximity and width of data pulses effectively shift the baseline for their neighbors, and with the longer rise and fall times in the cable, the transitions are moved from their ideal zero crossings.

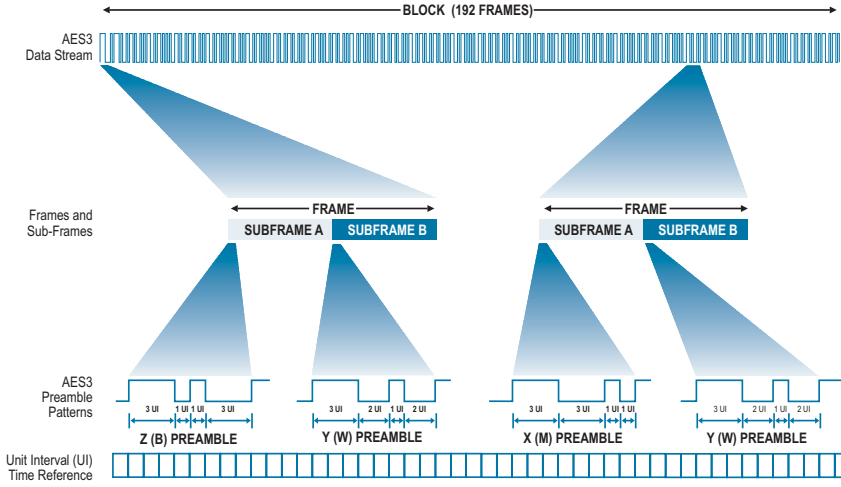


Figure 8. AES3 data pattern. Note that the Y preambles are identical in every frame.

As the AES3 interface uses the same signal to carry both clock and data, it is possible to induce jitter on the clock as a result of the data modulation. This means that care should be taken about mechanisms for interference between the data and the timing of the clock. The smearing of the waveform as a result of cable losses is one such mechanism. See Figure 9 and the *Intersymbol Interference* sidebar.

Data Jitter

Data jitter is a term used to describe the jitter of the transitions in the parts of the AES3 waveform modulated by the data. This form of jitter is often an indicator of intersymbol interference.

Figure 9 in the Intersymbol Interference sidebar illustrates this mechanism inducing data jitter of about 50 ns peak-to-peak in some of the transitions. Data jitter can also be produced by circuit asymmetries where a delay may vary between positive-going and negative-going transitions.

Preamble Jitter

Preamble jitter is a term used to describe the jitter on the transitions in AES3 preambles. The preambles are a set of static patterns which are used to identify the start of the digital audio subframes and blocks. (See Figure 8.) The Y preamble at the start of the second (B) subframe is a completely regular fixed pattern. This unchanging preamble can be used to make jitter measurements that are not sensitive to intersymbol interference, and are therefore a better indicator of either jitter at the transmitter device or noise-induced jitter, rather than jitter due to data modulation.

Intersymbol Interference

Figure 9 shows five AES3 interface signals, each with a different data pattern in the first three bits. The data is encoded by the bi-phase mark encoding scheme (also called Manchester code or FM code), which has a transition between every bit symbol and also a transition in the middle of the symbol if it is "1," but not if it is "0." The top signal represents "1-1-1," the second is "1-1-0," the middle "1-0-0," the next "0-1-0" and the last is "0-0-0."

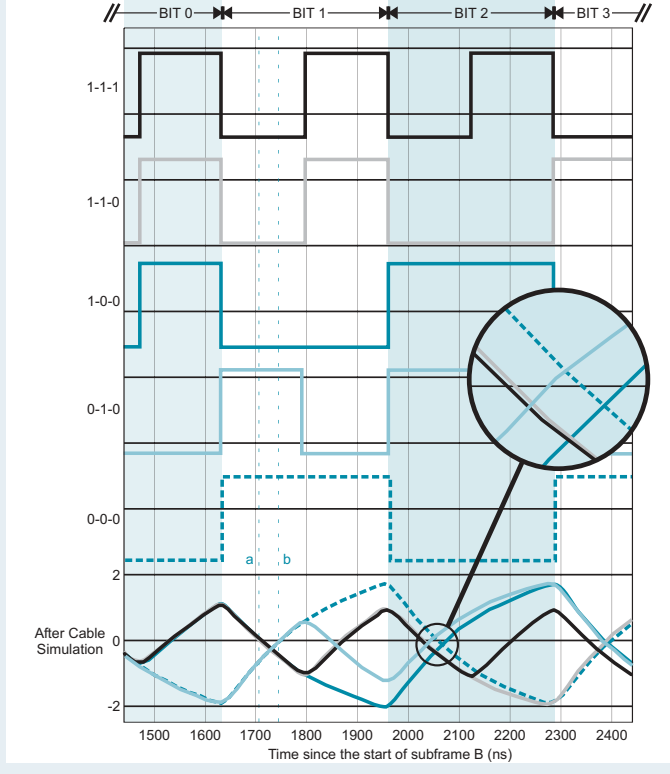


Figure 9. AES3 Intersymbol Interference

At the bottom of the chart, the figure also shows the signals as they may look after transmission down a long length of cable. These cable-affected signals were generated using the Audio Precision System Two cable simulation, and the five results have been overlaid on each other. The losses in a real cable would affect the signals in this manner, rolling off the high frequencies and reshaping the pulses with slower rise and fall times.

In each case the data shown were immediately preceded by the Y preamble, the preamble which begins the B subframe. (See Figure 8.) This preamble is a fixed pattern which lasts for 5 bit periods (10 unit intervals, or UI). A consequence of this is that the traces coming into the left-hand side of the cable simulator plot are at almost exactly the same voltage, since they have all followed the same path for a while. (The preamble is nominally 8 UI long, but the last part of the preceding bit and the first part of the following bit period are fixed to the pattern, resulting in a fixed pattern that is 10 UI long.)

The 1-1-1, 1-1-0 and 1-0-0 traces have a transition starting at 1465 ns (9 UI) from the subframe start because they have an initial "1" in their data. The 0-1-0 and 0-0-0 traces start with an initial "0" so they do not yet show a transition. All five traces then change direction at 1628 ns (10 UI) corresponding with the end of the first bit symbol. (The frame rate of this signal is 48 kHz, so 1 UI is 162.8 ns.)

The markers "a" and "b" indicate that the times of the zero-crossings from those transitions are 1705 ns and 1745 ns. The earlier transitions are those which have a "1" value in the first bit and the later transitions those which have a "0."

As a result of the high-frequency losses in the cable simulation the transition time is quite slow, so the zero crossings are about 100 ns after the inflections that indicate the start of the transitions. This interaction between the value of the first data symbol and the timing of the start of the second data symbol is called intersymbol interference*.*

This interference is more complex after the second bit symbol (about 2050 ns from the start of the subframe, also shown in the magnified view). Here there are four different zero-crossing times corresponding to the four possible bit patterns of the first two bits in the subframe. Most of the timing difference is due to the value of the second bit, but in addition there is a smaller difference relating to the state of the first bit.

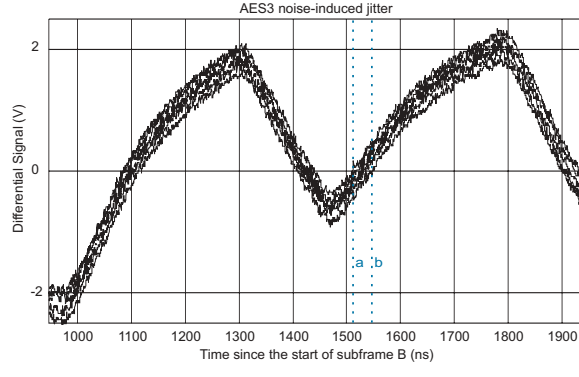
Interfering-Noise-Induced Jitter

If the pulse transitions were not sloped by the cable losses, the rise and fall times of the pulses would be so short that their zero crossings would be relatively unaffected by any added noise. However, the long transition times induced by cable losses allow noise and other spurious signals to "ride" the transitions, resulting in a shift of the zero crossing points of the pulses.

For example, noise on the signal can vary the time at which a transition is detected. The sensitivity to this noise depends on the speed of the transition, which, in turn, depends on the cable losses. This is illustrated in Figure 10.

The five traces on Figure 10 are all of the same part of the B subframe Y preamble. (As mentioned before, this static preamble pattern is chosen because it is not sensitive to data jitter, making the noise-induced jitter mechanism more obvious.) The two markers, "a" and "b" show the range of timings for

Figure 10. AES3 noise-induced jitter.



the zero crossing resulting from the third transition. Their separation is 31 ns. In this example, the noise producing this variation is a low-frequency sine wave of about 300 mV. This type of interference might be induced by coupling from a power line.

The amount of jitter introduced by noise on the cable is directly related to the slope at the zero crossing, as voltage is related to time by that slope. With fast transitions any interfering noise will not produce much jitter: the voltage deviation will cause a smaller time deviation.

> *Note: In this example a long cable was simulated by the Audio Precision System Two Cascade. However, this level of jitter would be reduced by several orders of magnitude for a short interconnection.*

Notice that the direction of the time deviation is related to the direction of the transition. For a transition shifted up by noise the rising transition will be early and the falling transition will be late; for a transition shifted down the opposite is true. Unlike data jitter from intersymbol interference, this form of jitter is more apparent to devices that recover a clock from a particular edge in the preamble pattern. That edge will only have one polarity and so the timing deviation of successive edges will sum together.

However, for systems using many of the edges in the subframe, transitions will be almost evenly matched in both directions and the cancellation will reduce the coupling of low frequency noise-induced jitter into the recovered clock. For noise at high frequencies successive deviations will not correlate and so cancellation will not occur.

Jitter Tolerance

An AES3 digital audio receiver should be able to decode interface signals that have jitter that is small compared with the length of the pulses that it has to decode. As the jitter level is increased the receiver will start to decode the signal incorrectly and then will fail to decode the signal— occasionally muting or sometimes losing “lock” altogether. The maximum level of jitter before the receiver starts to produce errors is called the *jitter tolerance* of the device.

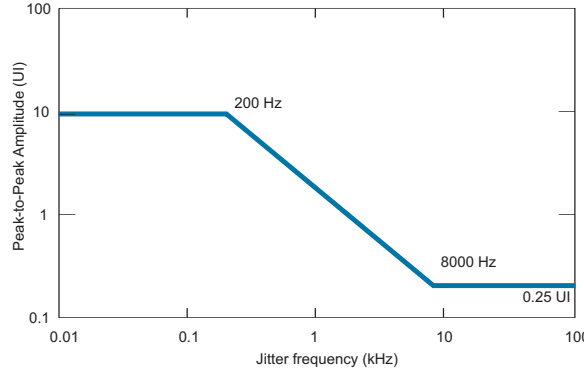
As the PLL characteristics sidebar showed, a clock-recovery PLL has a low-pass characteristic analogous to a mechanical flywheel: it responds or “tracks” to changes slower than the rate of the corner frequency, and it filters out changes that are faster.

Jitter tolerance, then, is independent of frequency for jitter above the corner frequency of the receiver, but as the rate of change of the timing (the jitter frequency) is reduced, the receiver is increasingly able to follow the changes.

This means that at lower jitter rates the receiver will be able to track increasing amounts of jitter, and so jitter tolerance rises.

For jitter frequencies close to the corner frequency it is possible—as a result of a poorly damped design—that the jitter tolerance is significantly reduced. This occurs because the resonance in the receiver is causing the match between the deviation of the incoming data transition timing and the receiver's estimation of the data transition timing to actually be worse than if the receiver was not tracking the jitter at all.

Figure 11. AES3 jitter tolerance template.



The AES3 interface specification defines a *jitter tolerance template*, shown in Figure 11. The tolerance is defined in UI. The line on the graph represents a lower limit for receiver jitter tolerance to sinusoidal jitter of the frequency shown on the X axis. Note that this template implies that receivers should have a corner frequency above about 8 kHz. This means that the receiver PLL will not be able to attenuate jitter below that frequency; instead, it will track the jitter and pass it on. A second PLL with a lower corner frequency must be used if significant jitter attenuation is required.

The Jitter Transfer Function and Jitter Gain

For a device that is synchronized to another clock (such as a digital input, a word clock, or a video sync reference) jitter on the external source could be passed through to the output. The jitter on the output is then a combination of this transferred jitter and the intrinsic jitter of the device.

Although the relation between input and output jitter can be very complex, it is still useful to model the transfer as a simple linear process. The *jitter transfer function* is a measure of the relation between input and output jitter, or *jitter gain*, versus jitter frequency.

Figure 12 shows the calculated jitter transfer function produced by a PLL with a corner frequency of 100 Hz. Notice that below the corner frequency the jitter gain is about 0 dB. Above the corner frequency the PLL attenuates the jit-

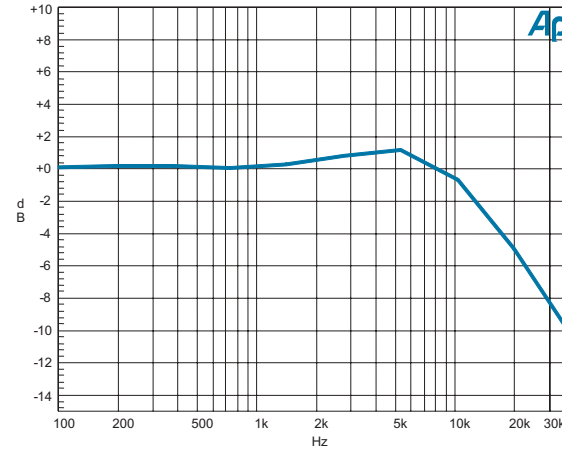


Figure 12. Jitter transfer function.

ter—initially with a slope of 6 dB per octave. This design has a second-order loop filter with a corner at 1 kHz which results in an 18 dB per octave slope above that frequency.

Notice that below the PLL corner, the gain reaches a peak of about 0.5 dB. It is usual for there to be a certain amount of gain just below the corner frequency; this is called *jitter peaking* and it is a consequence of the phase characteristic of the feedback loop in the PLL.

The AES3 standard sets an upper limit of +2 dB for jitter gain.

Non-Linear Jitter Behavior

The linear jitter transfer analysis does not account for non-linear relations between input and output jitter. Phase detectors can often have a “dead” spot where they are not sensitive to small phase deviations. As a result, the PLL output will drift until the phase detector becomes active and applies a correction. This drift will cycle back and forth, producing jitter.

Another non-linear jitter mechanism is the aliasing of high-frequency jitter as it is sampled by a lower-frequency mechanism within the PLL. For example, a 48 kHz frame-rate AES3 signal with a jitter component at 47 kHz could be used to generate an internal clock signal at a 48 kHz rate in order to lock a PLL. This 47 kHz signal would alias to the much lower frequency of 1 kHz where it might not be attenuated. When measuring a jitter transfer function this behavior would make it appear that the gain rises to maxima at multiples of the frame rate.

Jitter Accumulation

In a short chain of digital audio devices, with each device locked to the previous one, there are several contributions to the jitter at the end of the chain. Each device will add its own intrinsic jitter, and each interconnecting cable

will make some contribution with cable-induced jitter. There will also be some jitter gain or loss at each stage.

This process has been called *jitter accumulation.* The effect varies with the individual device jitter characteristics and the data patterns at each stage, but in some circumstances and with some "pathological" signals the jitter mechanisms could all combine in an unfortunate manner.

In a chain of devices with clock recovery systems having similar characteristics a pathological signal will have the same effect at each stage. As Table 1 shows, this can lead to a very large amount of jitter accumulation after only a few similar stages.

For the purposes of this calculation we are looking at jitter at frequencies below the jitter transfer function corner frequencies of all the devices, so jitter attenuation does not occur. Assume—for simplicity—that all the devices contribute the same amount of jitter, *J*, at each stage (this is lumping cable-induced and intrinsic jitter together). Also assume that each device also amplifies the jitter from the previous stage by the same gain—bearing in mind that gain is only possible for jitter near the peak in the jitter transfer function.

Table 1 lists the total output jitter produced at the end of three chains of stages, as a multiple of *J*:

Jitter Gain per Device	Total Jitter *(J)* after 3 Stages	Total Jitter *(J)* after 4 Stages	Total Jitter *(J)* after 5 Stages
0 dB (ideal)	3 *J*	4 *J*	5 *J*
1 dB	3.8 *J*	5.4 *J*	7.1 *J*
3 dB	6.2 *J*	10.2 *J*	15.8 *J*
6 dB	13.9 *J*	29.8 *J*	61.4 *J*

Table 1. Jitter accumulation

This shows that with a gain of 0 dB at each stage the output jitter is simply a sum of the jitter produced at each stage. (These jitter levels are peak values so they will add). Remember that this happens at frequencies *below* the corner frequency; at higher frequencies the input jitter will be attenuated, so the final output jitter will grow more slowly.

The gains of greater than 0 dB show the effect of jitter transfer function peaking. If peaking is present it will only occur near to the PLL corner frequency. Where the jitter is wide-band only a small proportion of it will be amplified and the peaking will have little effect. However, there are mechanisms that can concentrate the jitter in the region of the peak.

First, AES3 data-jitter can have narrow spectral components. With low-level audio signals, for example, the jitter will become coherent with the polar-

ity of the signal. This occurs because for signals close to zero, the more significant bits within the data word change together as an extension of the sign bit. If the interface audio signal is a low-level tone at one frequency, then the cable-induced jitter will tend towards a square wave at that frequency. Occasionally, a spectral peak could coincide with the peak in the jitter transfer function.

In a chain of devices using clock recovery systems with similar characteristics, this signal will have the same effect at each stage. The figure of 6 dB in the table reflects levels of peaking found in equipment that had been designed before this problem was widely understood. As the table shows, this can lead to a very large amount of jitter accumulation after only a few similar stages.

The normal symptom of a pathological level of jitter accumulation is for the equipment towards the end of the chain to very occasionally lose data, or even lock. Unfortunately, the circumstances are such that it is difficult to reproduce when the maintenance engineer is called.

The AES3 specification, since 1997, has two clauses that are intended to address potential jitter accumulation problems. The primary statement specifies that all devices should have a sinusoidal jitter gain of less than 2 dB at any frequency.

In addition, there is a standard jitter attenuation specification that should be met by devices claiming to attenuate interface jitter. This requires attenuation of at least 6 dB above 1 kHz. This frequency is much lower than the jitter tolerance template corner frequency, so these devices need a transmit clock which is separate from the data recovery clock that determines the jitter tolerance.

Sampling Jitter

Sampling jitter is the variation in the timing an audio signal through jitter in an analog to digital (ADC), digital to analog (DAC), or asynchronous sample rate converter (ASRC). In the former two cases this can often be associated with an observable *sample* clock signal but in an ASRC it may be a totally numerical process, as the samples of a signal are regenerated to correspond with new sampling instants: in that case the sample clock is a *virtual sample clock*.

There are many circumstances where a sample clock has to be derived from an external source. In the domestic environment this could be a digital audio recorder or a digital surround processor where the DAC sample clock is derived from the digital input data stream. For professional applications there are also devices with DACs, applications where the sample clocks of ADCs need to be derived from an external sync or where a digital stream needs to be resynchronized to a different reference using an ASRC.

Often this external source will have jitter that can be observed, measured and commented on. However, that is not sampling jitter. The external source might make a contribution to the sample clock jitter but that contribution de-

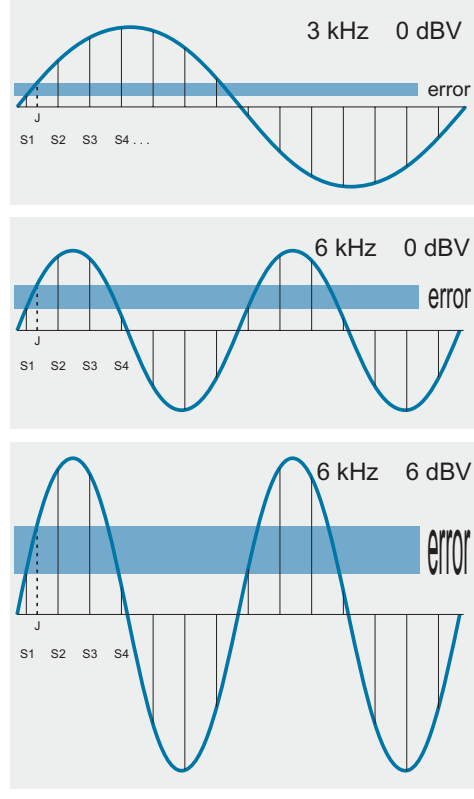
pends on the characteristics of the clock recovery circuit (or numerical algorithm) between the external source connection and the actual sample clock. This will have intrinsic jitter, jitter attenuation, and non-linearities in its behavior.

Sampling Jitter and the External Clock

There are many circumstances in which a sample clock must be derived from an external source. In a digital audio recorder or a digital surround processor, for example, the sample clock controlling the DAC is extracted from the input data stream. In other applications the sample clock of an ADC might need to be locked to an external sync signal, or a digital data stream might need to be resynchronized to a different clock reference using an *asynchronous sample rate converter* (ASRC).

This external clock source may well have jitter, but that, by definition, is *not* sampling jitter. The external source might make a contribution to the sample clock jitter, but that contribution depends on the characteristics of the clock recovery circuit (or numerical algorithm) between the external source connec-

Figure 13. In these examples the sampling rate is constant, but the sampled signal is varied in frequency and amplitude. Note how the amplitude value error for a jittered sample instant (J) increases with signal rate of change.



tion and the actual (or virtual) sample clock. This will have intrinsic jitter, jitter attenuation, and jitter non-linearities in its behavior.

Time-Domain Model

First, we will look at sampling jitter in the time domain.

The effect of a sample being converted at the wrong time can be considered simply in terms of the amplitude error introduced. Any signal that is not DC will change over time, and a wrong sampling instant will produce a wrong amplitude value. As you can see in Figure 13, the amplitude error is proportional to the rate of change, or slope, of the audio signal, which is greatest for high-level high-frequency signals.

Figure 14 illustrates the effect of random sampling jitter on a pure tone. The tone is shown as having an amplitude of 2 V rms and a frequency of 1 kHz. The error signal is calculated using random Gaussian jitter of amplitude 10 ns rms, and the simulation that produced this graph calculates the error of each sample at a sampling frequency of 176.4 kHz, which represents a 4X oversampled DAC in a CD player.

Notice how the error signal and the tone intermodulate. The error is the product of the slope of the tone and the jitter; as a result there are minima in the error at the peaks of the tone where the slope is flat.

The root-mean-square (rms) error computed by the simulation is 124 µV rms, or –84 dB relative to the tone. Assuming that this error is spread fairly evenly throughout the 88.2 kHz bandwidth represented by the sampling frequency of 176.4 kHz, we can estimate that measured over the nominal audio band to 20 kHz, the noise level would be 60 µV rms. This is 90.5 dB below the level of the tone.

This method of analyzing the effect of jitter can be used to make an estimate of the acceptable level of jitter of any given form. It can be simplified to calculate the level of jitter that, if applied to a "worst-case" signal, would produce an error of amplitude equal to the quantization interval. For example, a worst-case full-scale 20 kHz sine wave in a 16-bit system would have a maximum slope of:

$$2 \cdot \pi \cdot F \cdot A = 4.1 \text{LSB} / \text{ns}$$

where

$F = 20$ kHz, the tone frequency

$A = 2^{15} = 32768$ LSB, the tone amplitude (peak).

From this one might conclude that the jitter level should be no more than 244 ps peak, but that limit is fairly arbitrary—there is nothing special about an

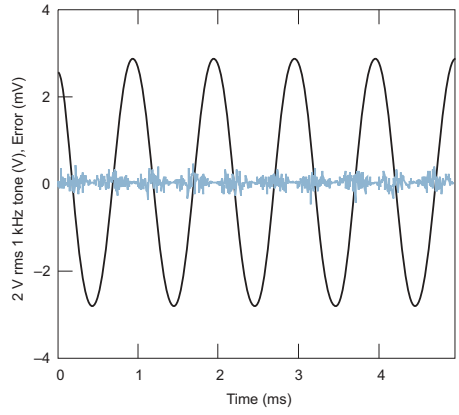


Figure 14. Sampling jitter on a 1 kHz tone. The black line is the signal; the noise-like trace around 0 V is the error introduced by the jitter, shown on a scale enlarged 1000 times.

error of 1 LSB amplitude—and has little relation to the audibility of the error, which will be related to the spectral content of the error.

Frequency-Domain Model

Another method of looking at the effect of jitter is to consider it as a modulation process, and analyze it in terms of frequency components. It can be shown mathematically that a simple relationship exists between a jitter spectral component, an audio signal spectral component and the resulting jitter modulation product.

If a signal is sampled with errors in the sampling instants, the effect is to modulate the signal in time. This is expressed mathematically in (1). The output signal $v'(t)$ is a time-displaced version of the input signal, $v(t)$, and the variation in the displacement is the jitter.

$$v'(t) = v(t - \Delta t). \tag{1}$$

The effect of this can be analyzed by considering sinusoidal jitter of frequency ω_j and peak-to-peak amplitude J.

$$\Delta t = j(t) = \frac{J}{2} \cdot \sin(\omega_j t). \tag{2}$$

The input signal may be a sine wave.

$$v(t) = A\cos(\omega_i t). \tag{3}$$

These equations can be combined and rearranged to:

$$v'(t) = A\cos(\omega_t t)\cos\left(\frac{J\omega_i}{2}\sin(\omega_j t)\right) + A\sin(\omega_t t)\sin\left(\frac{J\omega_i}{2}\sin(\omega_j t)\right). \quad (4)$$

Jitter amplitude (typically less than 10 ns) is generally much smaller than the signal period (typically greater than 40,000 ns). The product of small jitter modulation levels is itself very small, and for such cases we can make the following small-angle approximations:

$$\cos\left(\frac{J\omega_i}{2}\sin(\omega_j t)\right) \approx 1 \quad (5)$$

and

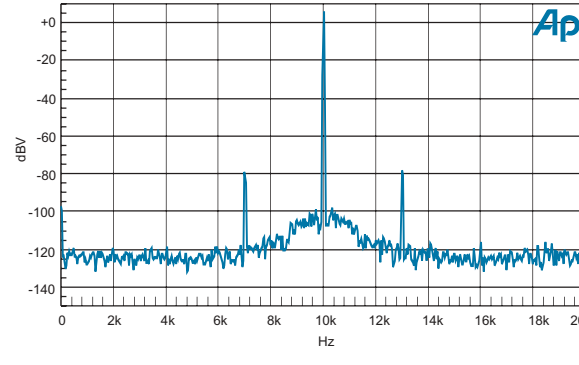
$$\sin\left(\frac{J\omega_i}{2}\sin(\omega_j t)\right) \approx \frac{J\omega_i}{2}\sin(\omega_j t). \quad (6)$$

Using these (4) becomes

$$v'(t) = A\cos(\omega_i t) + A\frac{J\omega_i}{4}\cos\left((\omega_i - \omega_j)t\right) - A\frac{J\omega_i}{4}\cos\left((\omega_i + \omega_j)t\right). \quad (7)$$

The output signal has the input signal with two other components at frequencies offset from the input signal frequency by the jitter frequency, and their amplitude is related to the product of jitter amplitude and signal frequency. This

Figure 15. Jitter-modulated sidebands.



result can be used when estimating the potential audibility of jitter modulation products.

Figure 15 illustrates this effect on a real signal. The input signal is at 10 kHz and the jitter modulation is at 3 kHz. The two components at 3 kHz offset from the input signal are the upper and lower jitter modulation sidebands. (In this figure there are also "skirts" to the spectrum closer to the 10 kHz component. These are due to some low-frequency noise-like jitter in the system).

The ratio of signal to each 'single' sideband, in dB, is:

$$R_{ssb} = 20\log_{10}\left(\frac{J\omega_i}{4}\right)\text{dB}. \quad (8)$$

This result is for sinusoidal jitter components. Using Fourier analysis, more complex waveforms can be broken down into sinusoidal components and the formula can be applied.

For convenience, the formula can be modified by summing the levels in both sidebands to give a total error, and using rms jitter levels, J_n, in nanoseconds and frequency, f_i, in kHz:

$$R_{dsb} = 20\log_{10}\left(J_n f_i\right) - 104\,\text{dB}. \quad (9)$$

Influence of ADC/DAC Architecture

The effect of jitter on converters can be more complex than just the time modulation of the audio signal as discussed above. Other signals (for example, ultrasonic noise created in a noise-shaping low-bit converter) can be sampled with the desired audio signal; in some case another modulation process could taking place as well.

Oversampling Converters

An *oversampling converter* is one that is processing samples at much more than the minimum rate required by the bandwidth of the system. This oversampling rate can typically be from 2X to 256X. The higher rates also use *noise shaping*, an important technique which can help provide low-cost solutions to high-resolution conversion. (Noise shaping can produce a separate side effect that is discussed later.)

Since the jitter bandwidth in a sample clock can extend to half the sampling frequency of the converter, the jitter in an oversampled converter will be spread over a wider spectrum than the jitter in a non-oversampled converter. The error caused by jitter modulation is related to the jitter spectrum, so the er-

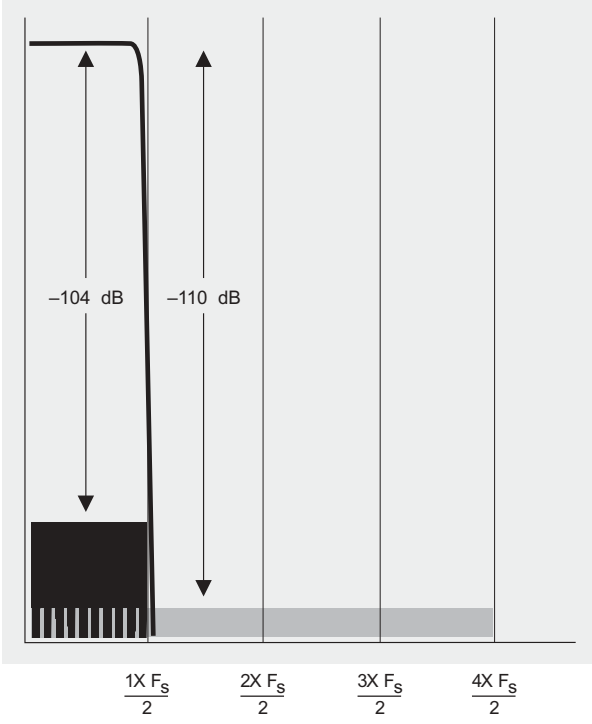


Figure 16. The black square represents the error created by spectrally flat jitter in a 1X converter. The gray rectangle represents the jitter error in a 4X converter, which contains the same power spread over a wider spectrum. The jitter power in the audio passband has been reduced.

ror signal from an oversampled converter is also spread across a wider spectrum.

To illustrate this: consider a 1 kHz signal being sampled with 1 ns of spectrally flat, noise-like jitter. By calculation, this will produce a total error 104 dB below the signal. This total error figure remains the same regardless of the sample rate of the converter.

As you can see in Figure 16, in a 4X oversampled DAC this error signal will be spread over four times the frequency range compared with a 1X converter. For audio purposes, of course, we limit our interest to the 20 Hz to 20 kHz bandwidth, and a measurement made over that range contains only one-quarter of the power of the full spectrum of error noise. One-quarter the power implies one-half the voltage, resulting in an error 6 dB lower than that for the non-oversampled converter.

Jitter sources, however, are normally not spectrally flat. Jitter is usually dominated by lower-frequency components, due both to the typical phase noise spectrum of oscillators and to the low-pass jitter filtering common in clock recovery circuits. Oversampling will not reduce the impact of this lower-frequency jitter.

Jitter-Induced Tones from Noise-Shaped Converters

I have observed an interesting artifact. Very low-level tones were observed on the output of a noise-shaped one-bit converter. This was a DAC that did not use any of the above techniques to reduce sampling jitter sensitivity—just a low-jitter quartz crystal VCO to make sure that any jitter on the sample clock was at a very low level. These tones appeared to be related to the modulation of the VCO control voltage. However, they went away when a higher speed and lower distortion op-amp was used in the post filter. I concluded that the effect was of non-linearity (in the lower performance op-amps) on the jittered ultrasonic noise that demodulated the jitter. (As the modulation would be similar to phase modulation, rather than to amplitude modulation, this requires some asymmetry between the upper and lower sidebands. The jitter modulation produces upper and lower sidebands which have opposite phase. If they had the same amplitude, then on demodulation they would cancel.)

Normally jitter does not produce tones in the absence of signal. Modulation sidebands may be tonal if the jitter is tonal—but they do *have to be sidebands of a modulated signal.*

Noise-Shaping and One-Bit Converters

For high rates of oversampling it is possible to reduce the number of bits while shaping the resultant quantization noise out of the audio band. This technique has many advantages, but it *does* generate ultrasonic noise. The level of this noise is related to the quantization interval. For a one-bit converter the total noise is close to the full-scale level of the converter.

The action of sampling jitter on this ultrasonic noise produces modulation products, just as it does on audio signals. These modulation products can fall in the audio band, and as the ultrasonic noise is present even when the audio signal is at a low level, there will be no benefit from masking. The effect is to raise the noise floor and so reduce the dynamic range of the converter.

To illustrate the scale of the problem: consider the ultrasonic noise produced in a 64X oversampled 48 kHz delta-sigma converter. (ADC or DAC—it does not matter in this example.) The ultrasonic noise is in a band starting above the audio band and going on to half the sample rate, 1.5 MHz. For 1 ns rms jitter the formula would imply modulation effects at levels of the order of:

$$20\log_{10}(1\text{ns}\cdot 1000\text{kHz}) - 104 = -44\text{ dB}.$$

Since this noise is spread over the band to 1.5 MHz, the level within the audio band would be less than this. With spectrally flat jitter it would be about 20 dB less, but it is likely that there would be at least a $1/f$ characteristic to the jitter spectrum, so the reduction may be increased to 40 dB. This leaves a noise floor of about –84 dB FS.

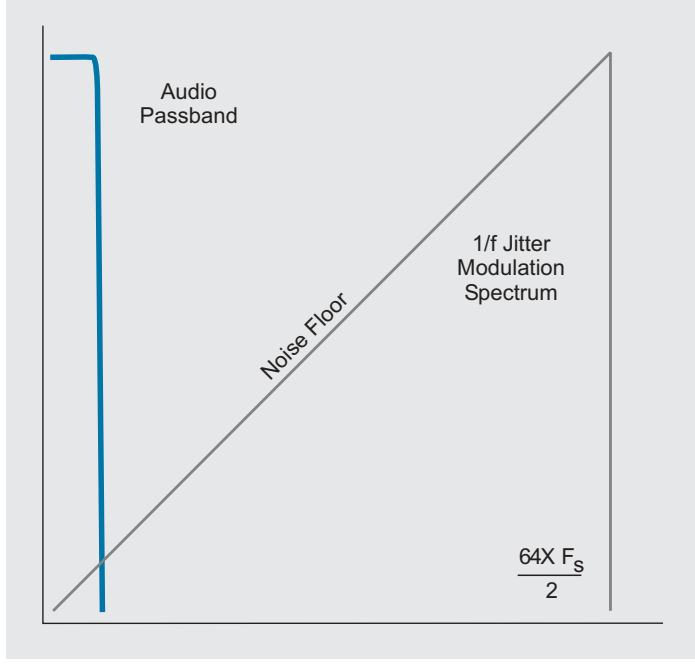


Figure 17. Ultrasonic jitter modulation products.

Reducing Jitter Sensitivity in Delta-Sigma Converters

Most of the commercially available integrated delta-sigma converter devices do not have anything like this sensitivity to jitter. How is that?

We find that it is possible to largely eliminate the effect of jitter modulation of high level ultrasonic quantization noise by *filtering the noise out before it is sampled*. Several techniques are available:

Switched-Capacitor Filters

Sampling or re-sampling occurs at the interface between the sampled signal domain and the continuous-time signal domain, which is not always the same as the interface between the digital and analog domains. A switched-capacitor filter operates on analog signals in the sampled signal domain.

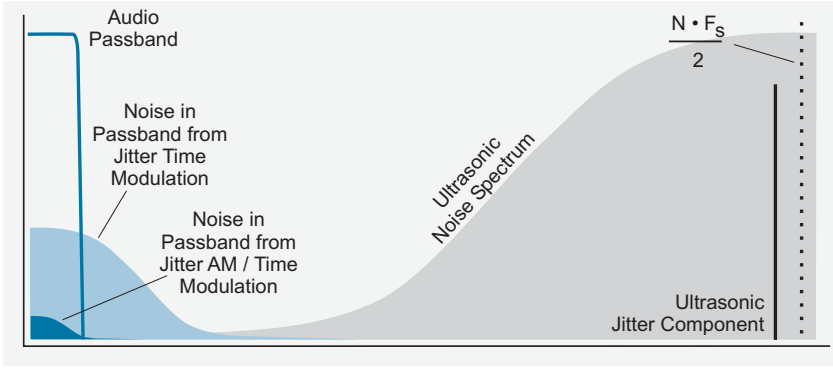


Figure 18. Spectrum of amplitude + time modulation products of ultrasonic jitter with ultrasonic noise.

In a delta-sigma ADC, the ultrasonic quantization noise is used in the delta-sigma modulator for feedback. This noise can be kept in the sampled signal domain if the analog filters within the modulator are implemented as switched-capacitor filters.

In a delta-sigma DAC, the ultrasonic noise on the DAC output can also be attenuated with a switched-capacitor filter.[4]

Multi-Bit Noise-Shaped Converters

Since integrated switched-capacitor filters have to be quite large to have a good noise performance, an alternate solution is finding favor. By increasing the number of levels in the quantizer, the quantization noise in the modulator is reduced. This reduces the ultrasonic noise by the same proportion. The Analog Devices AD1855, for example, has a 64-level modulator and does not use a switched-capacitor filter.[5]

Jitter-Induced Amplitude Modulation

There is another solution to high sensitivity to jitter due to the modulation of the ultrasonic quantizer noise. It is the combination of the jitter-induced time modulation with jitter-induced amplitude modulation.

Ordinarily, a DAC has a *current* or *voltage* output which is independent of sampling frequency. However, a DAC that uses a quantum of *charge* with every sample, rather than current or voltage, will have a current output that *scales with sampling frequency*. In this circumstance sampling jitter produces an amplitude modulation effect, which combines with the pure jitter time modulation in an advantageous manner.

With this kind of DAC, sampling jitter produces an amplitude modulation effect with the following output:[6]

$$v'(t) = A\cos(\omega_i t) - A\frac{J\omega_i}{4}\cos\left(\left(\omega_i - \omega_j\right)t\right) - A\frac{J\omega_i}{4}\cos\left(\left(\omega_i + \omega_j\right)t\right). \quad (10)$$

This amplitude modulation combines with the pure jitter modulation to produce the following:

$$v'(t) = A\cos(\omega_i t) + A\frac{J\left(\omega_i - \omega_j\right)}{4}\cos\left(\left(\omega_i - \omega_j\right)t\right) - A\frac{J\left(\omega_i + \omega_j\right)}{4}\cos\left(\left(\omega_i + \omega_j\right)t\right). \quad (11)$$

The sidebands for this combination now scale with the sideband frequencies, $\omega_i - \omega_j$ and $\omega_i + \omega_j$, rather than the modulated frequency, ω_i. Where ω_i is ultrasonic (and the sideband offset ω_j is very large) this reduces the impact of the jitter on any sideband modulated down into the audio band in approximate proportion to the ratio between the ultrasonic noise frequency component and the audio band frequency.

Where the signal under consideration is at high level and high frequency (a component of the ultrasonic noise) and the sideband offset is very large (due to an ultrasonic jitter signal), sidebands can be modulated down to much lower frequencies. This technique reduces the impact of the jitter modulating the ultrasonic noise down into the audio band in approximate proportion to the oversampling ratio, e.g. 256:1.

Sampling Jitter in Rate Converters

Sample rate converters (SRCs) are used to convert a signal from one sample rate to another. The conversion involves interpolating between the sample points on the input stream to generate values for the new sample points.

Where the two sample rates have an exact integer relationship the new sample points can be determined with no error. In that case it is possible to do the conversion with no sampling jitter, but the input and output streams need to be synchronized. A 44.1 kHz to 96 kHz sample rate conversion, for example, can be done using the mathematical relation of 320/147. The timing of the output stream can be determined from the input stream so that every 147 input samples corresponds with 320 output samples. The interpolation filter coefficients can be computed based on this exact relation. This sort of SRC is called a *synchronous* sample rate converter (SSRC).

Often the output sample frequency cannot be locked to the input. Additionally, some equipment is designed to retain the flexibility to cope with an arbitrary relationship between input and output timing. In these cases the conversion is more complex and includes an algorithm that tries to track the relation between the input and output samples based on their actual time of arrival. This sort of SRC is called an *asynchronous* sample rate converter (ASRC).

Virtual Timing Resolution

The algorithm used to estimate timing relations in an ASRC takes as an input the timing of the sample clock of one of the streams, and measures that with a higher rate clock that is synchronous with the other stream. The jitter in this measurement is determined by the resolution of the measurement clock.

For example, when converting from 48 kHz to 96 kHz the measurement clock may be working at 256 X 96 kHz. This resolution of 40 ns is the amplitude of the time quantization jitter being fed into the time-tracking algorithm.

This potential source of sampling jitter can have strong spectral components: if the 48 kHz clock is 5 ppm low and the 256 X 96 kHz rate is 6 ppm low, then the 40 ns time quantization jitter will be in the form of a sawtooth at a rate of about 25 Hz (256 X 96 kHz X 1 ppm).

Virtual Jitter Attenuation Characteristic

The ASRC timing estimation algorithm will have a jitter attenuation characteristic that can be modeled as a low-pass filter with a corner frequency. However, as this is a numerical process, if the device has enough mathematical resolution the filter corner frequency can be set very low. This means that an ASRC can have a high level of jitter attenuation.

As integrated ASRCs become less expensive, they are seen as a low-cost solution for the effective elimination of sampling jitter for DACs. The output sampling frequency can be fixed to a low-jitter, free-running crystal oscillator and the incoming data stream can be converted to that sampling frequency in the ASRC. A measurement of the clock at the DAC may reveal the low jitter of the crystal oscillator.

However, the re-sampling process within the ASRC needs to be considered as well. As the ASRC jitter is purely a deviation in the numerical value generated by the timing estimation algorithm, it cannot be measured directly. However, it can be evaluated by examining the effect on a high-frequency, high-level digital tone signal passing through the device.

Sampling Jitter Transfer Function

It is often convenient to assess the jitter performance of a device, be it an ADC, DAC or an ASRC, through its transfer function; that is, the effect it has on an audio signal. For the ASRC it may be the only method available.

Figure 19. FFT of DAC stimulated with a 12 kHz tone at –3 dB FS, with (black trace) and without (gray trace) 9.8 ns peak-to-peak wideband jitter.

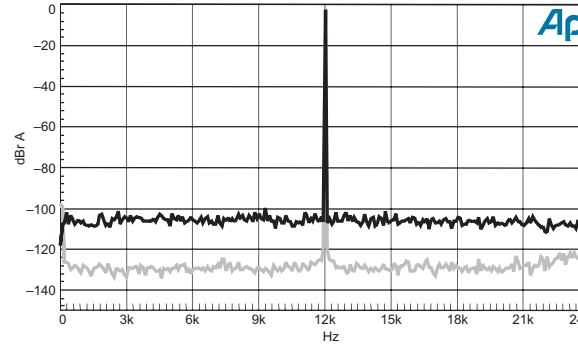
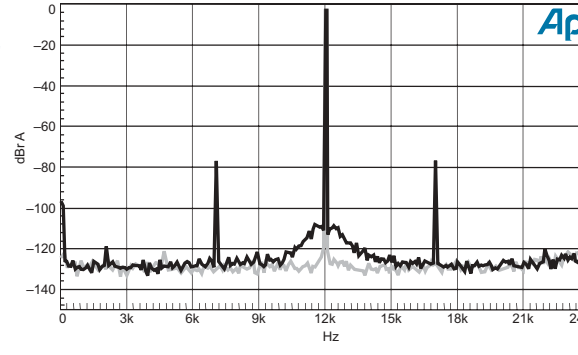


Figure 19 shows the frequency spectrum of a DAC stimulated with a 12 kHz tone at –3 dB FS. The digital input signal to the DAC is used to recover the sample clock, and the effect of the wideband jitter on that input is to raise the noise floor evenly throughout the band.

Jitter attenuation does not have a flat response. Since the jitter stimulation is flat and the modulation effect is also flat, one can conclude that in this case there is no jitter attenuation.

The System Two analog analyzer reports that the 22 kHz unweighted noise is 96 dB below the tone without the jitter stimulus and 80 dB below with the jitter. From this we can calculate that the jitter producing modulation of up to +10 kHz and –12 kHz offsets is 1.32 ns rms, or 12 ps/√Hz.

Figure 20. FFT of DAC stimulated with 12 kHz tone, with (black trace) and without (gray trace) 3.5 ns rms 5 kHz sine wave jitter.



More accurate spot frequency measurements can be made using a sinusoidal jitter stimulus. Figure 20 illustrates this. The error signal—dominated by the

sidebands—is 71.4 dB below the 12 kHz tone. By calculation we can see that this corresponds with sampling jitter of 3.5 ns—the same level as the jitter applied to the interface. This indicates that at 5 kHz there is no jitter attenuation between the applied stimulus jitter on the interface and the sampling clock on the DAC. (The skirts around the 12 kHz tone are probably low-frequency noise in the jitter generation mechanism.)

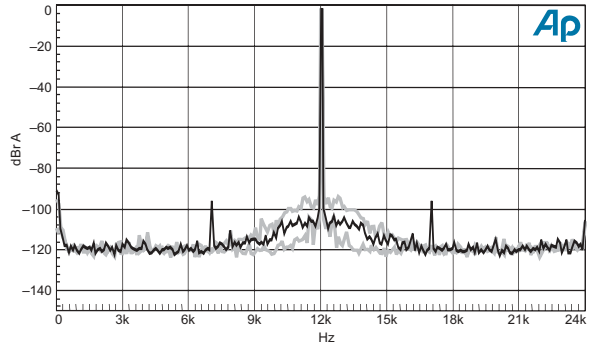


Figure 21. UPPER GRAY: FFT of 12 kHz tone with 9.8 ns peak-to-peak wideband jitter; BLACK: with 3.5 ns rms sine wave jitter; LOWER gray: no jitter.

As an example of how the results could vary, the same tests were repeated with a different device. Figure 21 shows the FFT traces. Notice that the 5 kHz sidebands are attenuated compared with Figure 19, and the higher-frequency components resulting from the wide-band jitter are attenuated relative to Figure 20.

Other Points to Note

The wide-band jitter components shown in Figure 21 are actually at a higher level where the jitter is not attenuated at low frequencies (close to the 12 kHz tone). This apparent increase in sampling jitter compared with the earlier result is not due to jitter gain. Instead, the wide-band jitter is being aliased by the interface receiver and the clock recovery system. If, for example, the sampling clock recovery system is using a 48 kHz clock from the receiver and the jitter has a bandwidth of 200 kHz, then the jitter in the region from 24 kHz to 200 kHz is aliased into the region from 0 Hz to 24 kHz. This increases the jitter noise density within the 24 kHz region by $10\log(200/24) = 9.2\,\text{dB}$.

Another feature to note is that there is significant low-frequency jitter even without the jitter stimulus, and with the sine stimulus the low-frequency jitter increases. These effects are not representative of the linear jitter transfer function but indicate low-frequency intrinsic jitter. The increase when the 5 kHz tone jitter is applied is possibly due to low-frequency noise in the jitter generation mechanism.

Sampling Jitter / Data Jitter Susceptibility.

J-test is an AES3 test signal that was developed to maximize the coherence of data patterns while at the same time providing a basic high-level stimulus tone. This test stimulates worst-case levels of data-jitter. The signal has two components, the first being an un-dithered square wave with a period of 4 samples. A cycle of this is shown here in hexadecimal notation:

```
C00000(-0.5)
C00000(-0.5)
400000(+0.5)
400000(+0.5)
```

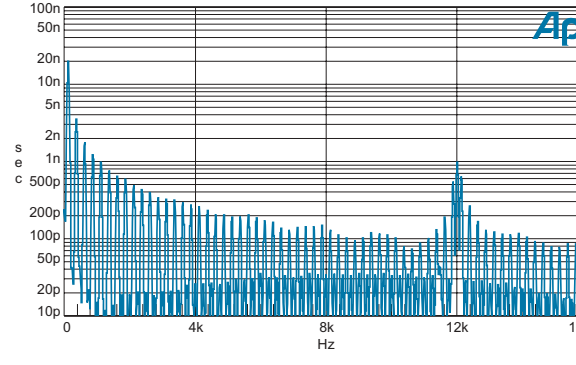
On conversion to analog at a sample rate of 48 kHz this signal would produce a sine wave with an amplitude of –3.01 dB FS at 12 kHz. (It looks like a square wave with a peak amplitude of –6.01 dB FS but in a properly band-limited system this sequence of values represents a sine wave of amplitude –3.01 dB FS.)

This is added to the second component, an undithered 24-bit square wave of amplitude 1 LSB, made by switching between the following:

```
000000(0)
FFFFFF(-1 LSB)
```

This square wave is repeated at a low frequency. The frequency is not critical but, for a sample frequency of 48 kHz, a rate of 250 Hz is normally used, as that makes the signal synchronous with the AES3 channel status block of 192 samples.

Figure 22. J-test jitter spectrum after cable simulation.



The combination of these signals results in the following 192-sample-long cycle of 24-bit data values:

```
C00000 C00000 400000 400000 (x 24)
BFFFFF BFFFFF 3FFFFF 3FFFFF (x 24)
```

The low-frequency coherent alternation in the values of the 22 LSBs produces strong jitter spectral components at 250 Hz and at odd multiples of that frequency. Figure 22, an FFT of the detected jitter signal in the Audio Precision System Two, illustrates this (but using a 384-cycle version of J-test producing a lower rate of 125 Hz). The intersymbol interference has been induced by the cable simulation. Notice that the amplitude axis is calibrated in seconds rms. This test was performed at 48 kHz. The component at 125 Hz has an amplitude of 19.91 ns. The jitter observed on the interface signal was about 35 ns peak-to-peak: this plot is of jitter at a part of the waveform where the amplitude is somewhat reduced from this.

Figure 23. FFT spectrum showing jitter modulation products from J-test after a cable simulation.

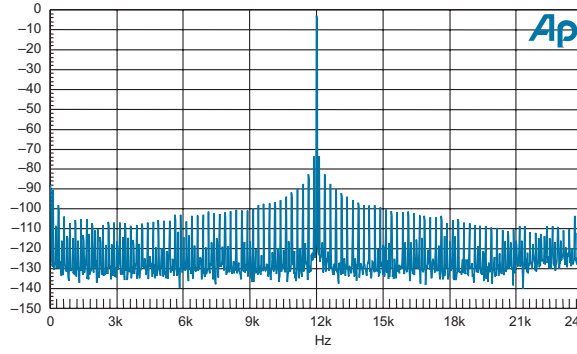


Figure 23 shows an FFT of the analog output of the first test device with J-test applied. Notice that the jitter sidebands follow the interface jitter spectrum reliably, which means that the test device is susceptible to data jitter on the interface.

The shape of each sideband matches the interface jitter spectrum of the previous figure, so we can also conclude that it does not have jitter filtering within the band. The 125 Hz sidebands are each about 70 dB below the stimulus tone (67 dB for both sidebands together). This corresponds with sampling jitter at that frequency of amplitude

$$\text{antilog}-\left((104-67)/20\right)/12=6\ \text{ns rms}.$$

Audibility considerations

It is one thing to be able to identify and measure sampling jitter. But how can we tell if there is too much?

A recent paper by Eric Benjamin and Benjamin Gannon describes practical research that found the lowest jitter level at which the jitter made a noticeable difference was about 10 ns rms. This was with a high level test sine tone at 17 kHz. With music, none of the subjects found jitter below 20 ns rms to be audible.[7]

This author has developed a model for jitter audibility based on worst case audio single tone signals including the effects of masking.[8] *This concluded:*

"Masking theory suggests that the maximum amount of jitter that will not produce an audible effect is dependent on the jitter spectrum. At low frequencies this level is greater than 100 ns, with a sharp cut-off above 100 Hz to a lower limit of approximately 1 ns (peak) at 500 Hz, falling above this frequency at 6 dB per octave to approximately 10 ps (peak) at 24 kHz, for systems where the audio signal is 120 dB above the threshold of hearing."

In the view of the more recent research, this may be considered to be overcautious. However, the consideration that sampling jitter below 100 Hz will probably be less audible by a factor of more than 40 dB when compared with jitter above 500 Hz is useful when determining the likely relative significance of low- and high-frequency sampling jitter.

References

1. AES3-1992—'Recommended Practice for Digital Audio Engineering—Serial Transmission Format for Two-Channel Linearly Represented Digital Audio Data' J. Audio Eng. Soc., vol. 40 No. 3, pp 147-165, June 1992. (The latest version including amendments is available from www.aes.org).

2. IEC60958-3:2000—'Digital audio interface—Part 3 Consumer applications' International Electrotechnical Commission, Geneva. (www.iec.ch).

3. Julian Dunn, Barry McKibben, Roger Taylor and Chris Travis—'Towards Common Specifications for Digital Audio Interface Jitter' Preprint 3705, presented at the 95th AES Convention, New York, October 1993.

4. Nav Sooch, Jeffrey Scott, T. Tanaka, T. Sugimoto, and C. Kubomura—'18 bit Stereo D/A Convertor with Integrated Digital and Analog Filters' Preprint 3113, presented at the 91st AES Convention, October 1991.

5. Robert Adams, Khiem Nguyen and Karl Sweetland, 'A 112 dB SNR Oversampling DAC with Segmented Noise-shaped Scrambling', AES Preprint 4774 presented at 106th AES Convention, San Francisco, September 1998.

6. Julian Dunn—'Jitter and Digital Audio Performance Measurements', Published in 'Managing the Bit Budget', the Proceedings of the AES UK Conference, London, 16-17 May 1994.

7. Eric Benjamin and Benjamin Gannon, 'Theoretical and Audible Effects of Jitter on Digital Audio Quality', Pre-print 4826 of the 105th AES Convention, San Francisco, September 1998.

8. Julian Dunn—'Considerations for Interfacing Digital Audio Equipment to the Standards AES3, AES5, AES11' Published in 'Images of Audio', the Proceedings of the 10th International AES Conference, London, September 1991. pp 115-126.

Analog-to-Digital Converter Measurements

Introduction

The performance of analog-to-digital converters (ADCs) and digital-to-analog converters (DACs) is influenced by complex mechanisms. This makes it difficult to characterize these devices using conventional measurement techniques. In addition, many of the measurements can be made only in the digital domain.

As a result, new measurement techniques have been developed that are sensitive to the error mechanisms within digital audio converters.

While traditional analog measurements have been made by equipment with hardware filters and meters, modern audio test instruments can construct these devices in software using digital signal processing (DSP) techniques. DSP makes other powerful tools readily available, most notably the Fourier transform. The Fourier transform is a mathematical analysis that can reveal a signal in great detail and is used in many of the tests described here.

The dual domain versions of the Audio Precision System One, System Two, Portable One and ATS-1 instruments have both analog and digital test interfaces and support these measurement techniques. The examples and procedures in this Application Note are specifically designed for System Two Cascade.

Level Measurements in the Digital Domain

Digital Full Scale

In the digital domain, signal levels are normally expressed relative to *digital full scale*. This is defined in AES17[1] as the level corresponding to the level of a sine wave that has a peak level equivalent to the maximum positive value. The following table illustrates these values for a 16-bit system.

16-bit Two's Complement Peak Values		
Number Base	Positive Maximum	Negative Maximum
Decimal	32767	–32768
Hexadecimal	$7FFF_H$	8000_H
Binary	$0111\ 1111\ 1111\ 1111_2$	$1000\ 0000\ 0000\ 0000_2$

It is implicit in this definition that the sine wave has no DC component. With perfect alignment of level, no dither, and without any DC offset, the signal will use the positive maximum code but not the negative maximum, which is very slightly further away from zero.

The definition in AES17 also specifies a frequency of 997 Hz for the sine wave. This frequency is selected in relation to the standard sampling frequencies of 44.1 kHz and 48 kHz so that sampling in successive cycles occurs at different phases of the sine. In contrast, a 1 kHz sine wave sampled at 48 kHz will be sampled at the same 48 phases of the sine wave in each cycle.

Decibels, Full Scale: dB FS

IEC 61606:1997,[2] the international standard on digital audio measurement, defines signal levels in *decibels, full scale* (dB FS) as:

$$\textit{Signal level (dB FS)} = 20 \cdot \log\left(A/B\right).$$

Where

- A is the amplitude of the signal whose level is to be determined, and
- B is the amplitude of a sine wave that corresponds to full-scale amplitude.

This definition is primarily intended for application where the measurements use a meter with a root mean square (RMS) detector, but it may also be used to apply to signal amplitudes measured using other detectors, such as quasi-peak, as long as the same meter is used to measure both the signal, A, and the reference, B. To avoid ambiguity the detector, if other than an rms detector, should be specified wherever measurements are quoted.

Note: A digital full-scale amplitude sine wave with no DC component might not be possible at the output of some ADCs. For example, if a device just starts to clip at a level 0.1 dB below digital full scale (perhaps due to a 1 % internal DC offset), the level of that signal is still expressed as –0.1 dB FS, even though it is the largest signal the ADC can convert without clipping.

Using dB FS When Full Scale Is Unattainable

Where a signal is in the digital domain it is normally expressed in a fractional format (two's complement fixed point) where the value for digital full scale is inherent to the numbering scheme.

16 bit value	Fraction of 16-bit full scale	Level
0.111111111111111_2	1.000000	0.000 dB FS
0.000110011001101_2	0.100009	–19.999 dB FS
0.000000101001000_2	0.010010	–39.991 dB FS

In circumstances where it is impossible to pass a sine wave with a level equivalent to digital full scale, analog input levels can still be expressed relative to digital full scale (in dB FS) if the gain of the converter is known. The value of the digital level measured can be compared with the value of digital full scale, and the difference in decibels can be used to describe the analog level in dB FS.

For example, if the gain of an ADC has been determined it is often more convenient to specify an input level as –20 dB FS, rather than "an analog input level that corresponds with a level of –20 dB FS on the digital output when taking into account the known gain of the ADC."

Digital Peak Level Metering Using Sample Values

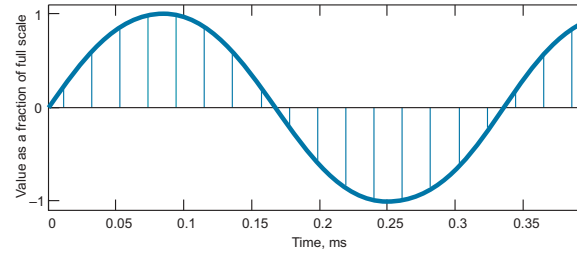
The simplest form of digital audio level meter detects the peak values of the sampled audio data. This is often how level-metering displays on studio and domestic digital audio equipment operate. It is important to recognize the limitations that these meters have.

Figure 24 shows a 3 kHz sine wave that peaks to full scale. It also shows lines corresponding to sampling instants when sampled at 48 kHz. There are exactly 16 sampling intervals in one cycle of the sine wave. The spacing of these points, in phase angles of the sine wave is:

$$\frac{360}{16} = 22.5°.$$

In this case—because there is an exact number of sample periods in the period of the sine wave—these same points would be sampled at every cycle. The peak sample value depends not only on the amplitude and frequency of the sine wave but also the phase with respect to the sample clock.

Figure 24. 3 kHz sine wave sampled at 48 kHz.

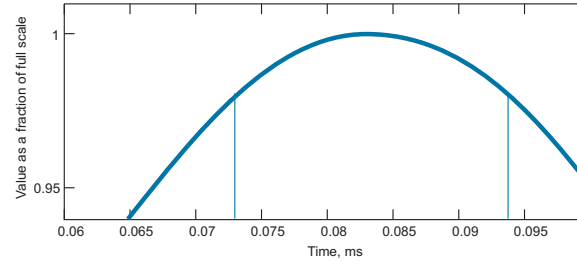


The worst case level error would occur if the peak were halfway between sampling points:

$$20 \cdot \log\left(\cos\frac{22.5^\circ}{2}\right) = -0.17 \text{ dB FS}.$$

This error is shown in Figure 25.

Figure 25. 3 kHz sine wave crest sampled at 48 kHz.



In most cases this behavior is not a problem because during the measurement it is likely that samples would be occurring near the peak in the waveform. The following are cases where the under-reading may be significant:

- a sine wave with a period that has an exact integer ratio with the sampling period. This will cause samples to occur at a very limited number of phases of the sine wave; for example, 3 kHz at a 48 kHz sample rate has 16 samples per cycle. The relationships are not always that obvious: a frequency near 9.14 kHz has the same 21 samples repeated every 4 cycles.
- any signal that has frequency components close to having an integer ratio with the sample frequency, but not actually synchronous. This signal would slowly drift in timing relative to the samples and would cause the sampled peak reading to fluctuate.
- any high bandwidth signal with significant high frequency content and only a short duration. During measurement only a few samples would occur near the crest or crests.

The standard measurement frequency of 997 Hz has a sequence of different phases that will extend for one second when sampled at 48 kHz. In that one second 48000 different phases of the 997 Hz sine wave will be sampled. These will be evenly distributed over the sine wave, with a spacing of

$$\frac{360}{48000} = 0.0075°.$$

That means that a simple signal level meter using a detector that measures only the maximum sample amplitude will see, over a period of one second, a sample which is no further than half this amount from the actual peak of the signal. The level error due to this is vanishingly small:

$$20 \cdot \log\left(\cos\frac{0.0075°}{2}\right) = -1.85 \cdot 10^{-8} \text{ dB}.$$

It is also possible to reduce the peak metering error by interpolating the data by oversampling. The higher density of data points for which there are samples reduces the error in the estimate of signal peak amplitude.

The level meter on the APWIN Digital Input/Output (DIO) panel uses a form of peak sample detection and should not be used for ADC performance measurements. The Digital Analyzer panel, which uses rms and quasi-peak detectors, should be used instead.

RMS Metering

Since rms measurements are much less sensitive to the relative phase of sampling instants, an rms meter is normally used to measure signal levels in ADC testing. This also brings mathematical advantages.

For example, if a signal is made up of several components, the total mean square amplitude of the signal is the sum of the mean square amplitudes of each component, as shown:

$$V_{rms} = \sqrt{VA_{rms}^{\ 2} + VB_{rms}^{\ 2}}.$$

This can be translated to levels in dB so that

$$V(dB) = 10 \cdot \log\left(\text{antilog}\left(\frac{VA(dB)}{10}\right) + \text{antilog}\left(\frac{VB(dB)}{10}\right)\right).$$

Additionally, fast Fourier transform (FFT) amplitude displays show the mean square amplitude in each frequency bin, providing a direct relationship between rms measurements and FFT displays. See **The Fourier Transform**, page 72.

Quasi-Peak Signal Level Metering

Some error behavior in ADCs can produce occasional spikes, and a test using some form of peak detector is appropriate to check for this sort of fault. The actual response of a peak-detecting meter to a momentary stimulus depends on the attack and decay times of the meter, so it is important to either specify these parameters with every measurement or to use a peak meter with a standard characteristic. The standard meter in common use is the quasi-peak detector defined in ITU-R BS.468 (formerly CCIR 468). This detector (called **Q-Peak** on the APWIN analyzer panel detector lists) is supported by Audio Precision test equipment.

Some standards require noise measurements to be made using the quasi-peak detector. This will produce significantly higher readings for typical converter noise than an rms meter, as shown in the section on noise measurement on page 53.

Measurement Techniques

Notes on the APWIN Procedure Examples

In the following sections APWIN procedures are used to illustrate the measurement techniques. The procedure files (*.apb) are supplied on the included CD-ROM and are also available for download from the Audio Precision Web site at audioprecision.com. These procedures have been designed to be used with Audio Precision System Two Cascade but should illustrate the processes for other equipment.

Prior to running the procedures, the ADC to be tested and the APWIN generator and digital I/O panels must configured so that the ADC passes a signal from the analog generator output to the digital analyzer input. The signal being generated is not critical, but the procedures assume that it is within the passband and that it is not clipping the converter. Any other settings that are required by the test will be configured by the procedure.

For any ADC under test the configuration only needs to be performed once, and the procedures should not alter it. The APWIN configuration for setting the interfaces for a specific ADC can, of course, be saved to a test file, which can then be loaded before using the procedures.

Gain

The gain of an ADC is not a unit-free ratio so it cannot be quoted in dB. It is normally quoted indirectly as the analog level corresponding to a digital output level of 0 dB FS. Many practical devices either cannot quite reach 0 dB FS, or have non-linearities that mean that the gain at that level is not rep-

resentative of the gain over most of the range. For this reason, gain is often measured at a digital output level below 0 dB FS, for instance –20 dB FS.

As an example, you may find that an analog level of 3.81 dBV on the input to an ADC generates an output level of –19.87 dB FS. There is not a conventional method of reporting ADC gain, but it may be quoted as the digital output level corresponding to a reference input level, like this:

$$0 \text{ dBV} = -19.87 - 3.81 = -23.68 \text{ dB FS.}$$

As an alternative this can also be described in terms of the analog level corresponding to digital full-scale level:

$$0 \text{ dB FS} = 3.81 + 19.87 = 23.68 \text{ dBV.}$$

Unless otherwise specified (as in a frequency response measurement), the gain is normally quoted at a frequency of 997 Hz.

Setting Stimulus Levels in dB FS

Test signals for the measurement of ADCs, although analog voltages, conventionally have levels specified in dB FS. The code shown in Figure 26 illustrates a simple method to do this in AP Basic. The analog generator output (previously set to a level that does not clip the ADC) is adjusted to correspond with a level of –5 dB FS at the ADC.

```
  ' Estimate and set generator level required to get -20dBFS
 output
       AP.Gen.ChAAmpl("dBV") = AP.Gen.ChAAmpl("dBV") -
 AP.S2Dsp.Analyzer.ChALevelRdg("dBFS") - 20
       Wait 0.5
       AP.Gen.ChAAmpl("dBV") = AP.Gen.ChAAmpl("dBV") -
 AP.S2Dsp.Analyzer.ChALevelRdg("dBFS") - 20
       Wait 2
  ' Estimate gain at -20dBFS, Set Generator to produce
"Level"dBFS
       AP.Gen.ChAAmpl("dBV") = AP.Gen.ChAAmpl("dBV") -
 AP.S2Dsp.Analyzer.ChALevelRdg("dBFS") + ADCOutputLevel
```

Figure 26. Subroutine from within "a-d tech note utilities.apb." Setting stimulus levels in dBFS.

An initial estimate of the gain is made and the input is set to produce an output level of approximately –20 dB FS. The value for gain at that level, `Gain20`, is then used in a second iteration to set up the desired output level to the value of `NewOutputLevel`.

Gain Stability

The gain of an ADC may drift due to instability in the converter reference voltage or in the value of other components. This variation can be monitored over time to determine the gain stability.

AES17 defines an input logarithmic-gain stability test which measures the range of gain seen in an ADC over an hour's time. A brief (typically five minute) warm-up period precedes the test. The measurement is of output level of the ADC, with the input level set to produce a –6 dB FS output initially. The procedure in Figure 27, "a-d input gain stability.apb," performs this test.

Gain-Frequency Response

In sampled systems the bandwidth of the input has to be limited to the *folding frequency*, or half of the sampling frequency, to avoid aliasing. Modern audio ADCs normally have this *anti-alias* filter implemented with a combination of a sharp-cutoff finite impulse response (FIR) digital filter and a simple low-order analog filter. The digital filter operates on a version of the signal after conversion at an oversampled rate, and the analog filter is required to attenuate signals that are close to the oversampling frequency. This analog filter can have a relaxed response, since the oversampling frequency is often many octaves above the passband.

The FIR filter characteristics may be specified very tightly and will have a fine ripple characteristic to the edge of the passband. Above that frequency the response will tail off quite sharply. The key parameters are

- stopband attenuation, or alias rejection;
- attenuation at the folding frequency;
- passband deviation;
- passband ripple amplitude;
- passband ripple periodicity;
- filter dispersion; and
- group delay.

The analog filter, as mentioned, will normally have low-order characteristics. As a result of component tolerances, these low-order characteristics could dominate the deviation of the amplitude response.

```
 Sub Main
'#Uses "a-d tech note utilities.apb"
'*************************************************************************
'APWin procedure developed to illustrate the article
'"Analog to Digital Converter Measurements" written by
'Julian Dunn.  (c) Julian Dunn 2000
'*************************************************************************
'Set level to -6 dB FS and examine gain variation

     Open MacroDir & "\A-D Converter Test Report.LOG" For Append As #1

     Print #1,
     Print #1, "=================================================="
     Print #1, "Input logarithmic-gain stability (AES17-1998 cl 5.6)"
     Print #1, "=================================================="
     AP.S2CDsp.Program = 1
     SetADClevelChA (-6)                        'Apply analogue -6dBFS

     BargraphNumber = AP.BarGraph.New
     AP.BarGraph.AxisLogLin(1) = 0
     AP.BarGraph.Id(1)  = 6005
     AP.BarGraph.AxisLeft(1,"dBFS") = -6.5
     AP.BarGraph.AxisRight(1,"dBFS") = -5.5

     Wait 6               'Wait a short while for readings to stabilise
     AP.BarGraph.Reset BargraphNumber          'and then reset meter

' Note the range of the readings on the meter after an hour
     Close #1
 End Sub
```

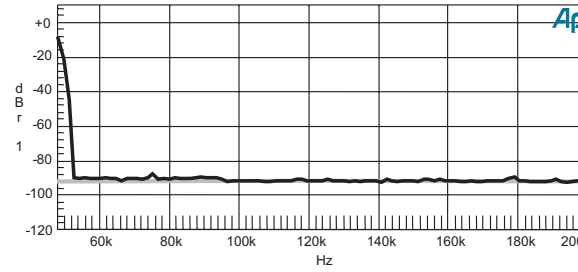
Figure 27. Procedure "a-d input gain stability.apb."

There may also be other factors affecting the frequency response, including components such as transformers, or perhaps a DC blocking filter that is often implemented in the digital and/or analog domains.

The following graphs illustrate some procedures to test frequency response applied to a high-quality ADC operating at a 96 kHz sampling rate.

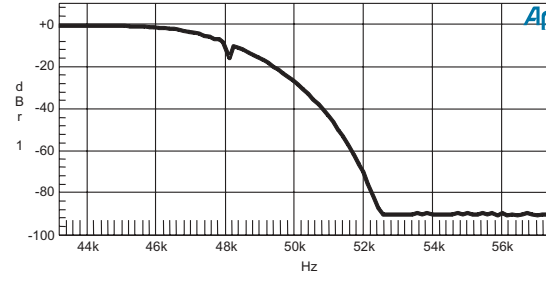
The plot in Figure 28 was generated by "a-d stopband.apb." This procedure measures the signal attenuation on the digital output of the ADC at frequencies from the folding frequency to 200 kHz. The stimulus tone is applied at –20 dB FS. Any signal in this range would appear aliased below the folding frequency into the passband, so suppression of these components is important.

Figure 28. Anti-alias filter stopband attentuation.



This is similar to the alias suppression test described in AES17. The black line on the graph is the response of the ADC. The noise floor with the ADC input muted is shown in gray.

Figure 29. Anti-alias filter transition region.



Notice that the alias suppression above 52 kHz is enough to reduce the –20 dB FS signal to below the noise floor of the measurement. However, there may be problems for signals in the transition region, below 52 kHz but above the folding frequency. The following trace is designed to examine this.

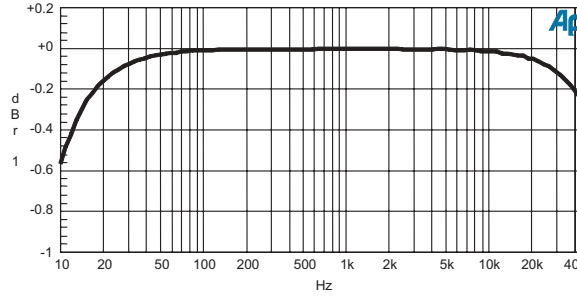
The procedure “a-d antialias corner.apb” has produced the plot in Figure 29. It is similar to the stopband measurement in Figure 28 but focuses on the response in the transition region between the anti-alias filter passband and stopband. This region is of interest as it indicates the potential for this ADC to suffer from aliasing between 48 kHz and 52 kHz, as well as in the margin between the top of the passband and the folding frequency.

The minimum alias suppression is near the folding frequency of 48 kHz, with an attenuation of approximately 10 dB. (The measurement at the folding frequency itself, 48 kHz, is attenuated by an amount that depends on the relative phase of the tone and the ADC sampling, so the notch on this graph is probably not significant.)

The passband frequency response shown in Figure 30 was made using procedure “a-d passband.apb.” The result is displayed on a more magnified Y-axis scale to examine the deviation of the passband response from flat. The zero reference is set at 997 Hz; high- and low-frequency rolloffs are about 0.2 dB at

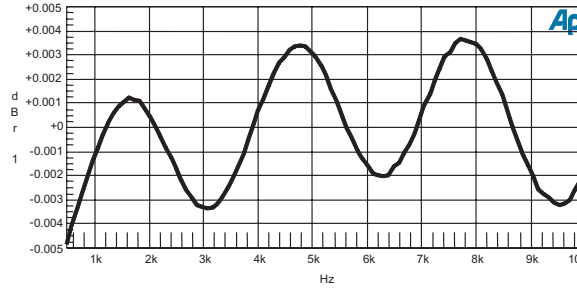
20 Hz and 40 kHz. Even at this magnification the ripple due to the digital filter is not visible.

Figure 30. Frequency response in the ADC passband.



To view the ripple, Figure 31 has been made by concentrating on the middle of the audio band in Figure 30 and then magnifying the Y-axis even further. This shows the sinusoidal nature of the filter ripple and the two components of the response variation. These variations are due to two cascaded stages of equiripple anti-alias filtering in this design.

Figure 31. Anti-alias filter ripple.



This ripple is so small that it could not be audible as a signal level variation. However, it is interesting because it indicates the time dispersion of the passband signal due to the filter. The time-domain equivalent of a sinusoidal gain variation in the frequency response is a pair of attenuated duplicates of the signal, with one duplicate occurring before and one after the main signal. The amplitude and relative timing of these duplicates can be calculated from the ripple periodicity and amplitude.

In this example, the period of the finer of the ripple components is about 3 kHz. (Observe the crests at about 1.6, 4.7 and 7.7 kHz which are 3 kHz apart). The reciprocal of this periodicity indicates the timing offset to be:

$$\frac{\pm 1}{3\ \text{kHz}} = \pm 333\ \mu\text{s}.$$

This is the advance of the pre-echo and the delay of the post-echo relative to the main signal.

The amplitude of these echoes is directly related to the amplitude of the ripple, which, as you can see in Figure 31, is ±0.003 dB. If we assign the main signal a value of 1 and convert the ripple from dB to a linear scale, we can calculate the sum of signal plus ripple, as shown here:

$$10^{\left(\frac{\pm 0.003}{20}\right)} = 1 \pm 0.0003.$$

The rms value of this ripple component, expressed as a ratio to the main component, is therefore:

$$20 \cdot \log\left(0.0003 \div \sqrt{2}\right) = -73\,\text{dB}.$$

This energy is divided between pre- and post-echoes, so each echo is at –76 dB relative to the main signal.

Note: The explicit measurement of group delay is not covered in this Application Note. When comparing the timing of the analog input and the digital output signal in an ADC, the timing instant corresponding to the sample value needs to be defined. For an AES3 or IEC 60958 signal, the sample instant is the first transition of the preamble at the start of the frame corresponding to that data value.

Input for Full-Scale Amplitude and Maximum Input Signal Level

For most ADCs the maximum input signal level is the analog level that corresponds to digital full scale, and right up to the onset of digital clipping the measured distortion is very low. In such a case the determination of maximum input signal level is similar to determining the gain of the ADC. The maximum level is very close to 0 dB FS.

With a small DC offset in the converter, a sine wave would not reach positive and negative full scale at quite the same time, so the maximum level would be slightly less than digital full scale. For example, a DC offset of 1% of the full-scale range of the ADC would cause digital clipping to occur at 99% of full-scale level, which is –0.09 dB FS.

There are two other definitions of maximum input signal level that are useful when digital full scale cannot be attained. One is based on the onset of distortion of a specified amount, and the other is based on a specified amount of signal level compression. AES17 provides specifications for the amounts of distortion or compression. The procedure “a-d input for full-scale.apb” uses all three techniques, compares the results of each and presents the lowest as the result.

```
================================================
Input for full-scale amplitude (AES17-1998 cl 5.4)
and maximum input amplitude (AES17-1998 cl 5.5)
================================================

Gain at -20 dB FS implies input full scale at 22.69 dBV

Maximum input amplitude defined by 1% distortion is 22.82 dBV
 (with output level 0.08 dB FS (RMS))

Input full scale defined by signal just reaching positive full scale
was not measured as positive full scale cannot be reached

Maximum input amplitude defined by 0.3dB compression is 23.45 dBV
(with distortion 3.876% and output level 0.47 dB FS (RMS))

Results of test
Maximum input amplitude is: 22.82 dBV
(defined by the level for 1% distortion)

Input level for full-scale amplitude is: 22.32 dBV
(defined as 0.5 dB below the maximum input level)
```

Figure 32. Results of procedure "a-d input for full-scale.apb." DUT is a high-quality ADC.

This test was run on a high-quality ADC with the results shown in Figure 32, which illustrates the variation produced by the three measurement methods. In this case it was not possible to reach digital full scale due to a small DC offset being subtracted (after clipping) by a DC blocking filter in the digital domain.

The procedure also measured the maximum input amplitude using the two alternate methods, by measuring both the level at which the signal clipped enough to have 1% distortion, and the level at which the signal is compressed by 0.3 dB. The lower of these results is defined as the maximum input amplitude.

AES17 specifies that where the full-scale output amplitude cannot be achieved, a level 0.5 dB below the maximum input amplitude is quoted. In this example the compression result was much higher than the 1% distortion result, so the distortion result was used.

Another device, a popular portable DAT recorder, was tested as shown in Figure 33. The test was performed with the recorder's automatic level control set to "manual" and the record level set to maximum.

```
================================================
Input for full-scale amplitude (AES17-1998 cl 5.4)
and maximum input amplitude (AES17-1998 cl 5.5)
================================================

Gain at -20 dB FS implies input full scale at -12.41 dBV

Maximum input amplitude defined by 1% distortion is -12.16 dBV
 (with output level 0.20 dB FS (RMS))

Input full scale defined by signal just reaching positive full scale
 is -12.40 dBV
 (with distortion, 0.0079 %, and output level 0.00 dB FS RMS)

Maximum input amplitude defined by 0.3dB compression is -11.57 dBV
(with distortion 3.786% and output level 0.55 dB FS (RMS))
Results of test

Maximum input amplitude is: -12.16 dBV
(defined by the level for 1% distortion)

Input level for full-scale amplitude is: -12.40 dBV
(which just reaches positive full scale)
```

Figure 33. Results of procedure "a-d input for full-scale.apb." DUT is a portable DAT recorder.

In this case the full-scale level was reached, at just 0.02 dB below the level predicted from the gain at –20 dB FS.

Maximum Signal Level Vs. Frequency

There may be mechanisms within the operation of an ADC that make the maximum input level vary with frequency. By regulating the generator level in order to achieve a specified distortion level, APWIN can plot the input and output levels that correspond with a distortion reading of 1%. A procedure to set this up is "a-d max input level v freq.apb." The plot of the results of the test using the high-performance ADC is shown in Figure 34.

The input level (the lower line on the graph) is plotted with the 0 dBr reference set at 997 Hz as the analog level corresponding to 0 dB FS. The upper line is the output level measured at the same time. These results show no significant deviation from perfect performance. The low-frequency rolloff at the output is as a result of the DC blocking filter being implemented in the digital domain. The high-frequency rise on the input level reading is a consequence of the low-order harmonics falling outside the passband of the ADC anti-alias filter, with the result that a higher input level is tolerated before the measured distortion products reach 1%.

Figure 34. Maximum input level for 1 % THD+N vs. frequency. DUT is a high-quality ADC.

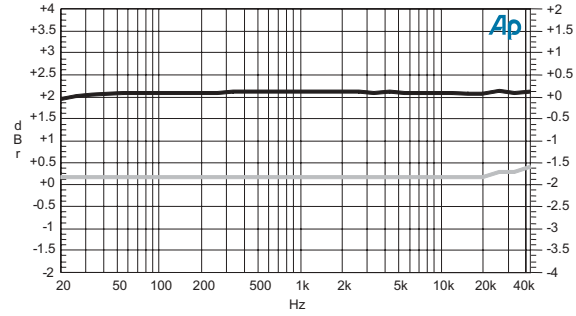
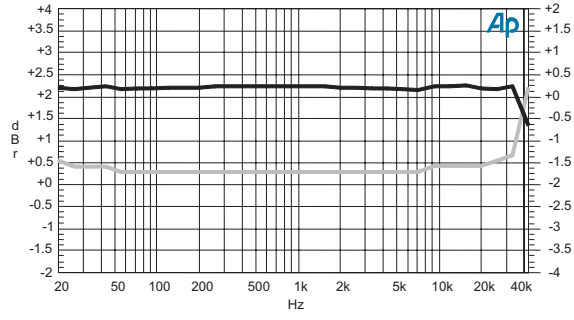


Figure 35 shows the same plot for the portable DAT recorder. This graph shows a rise in the maximum input level (lower line) at both low and high frequencies. As before, the high-frequency rise is due to the elimination of the distortion harmonics from the passband of the converter. The low-frequency rise is a result of the DC blocking filter being implemented in the analog domain, attenuating the signal before the converter so that slightly higher levels are tolerated before full scale is reached. Neither of these characteristics illustrate any problem with the ADC being tested.

Figyre 35 Maximum input level for 1 % THD+N vs. frequency. DUT is a portable DAT recorder.



Noise

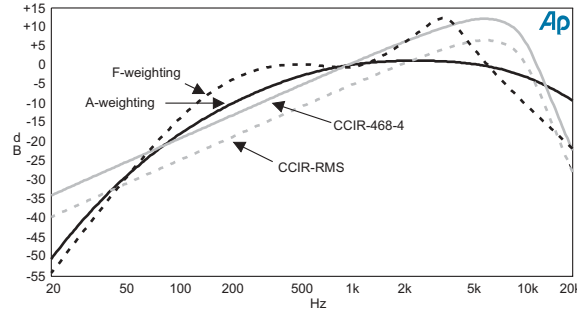
The analog-to-digital conversion process will always produce errors, which in an ideal system should be inaudible. However, in a practical system the conversion errors can often be audible or become so as a result of amplification. These errors are much more acceptable to the listener if they have a random character and are not manifested as distortion, chirping or modulation effects. The error should be noise-like, possessing a spectrum that does not have spurious tonal components and that does not change with the signal.

Errors are more acceptable in the presence of high level signals, which can mask their audibility. For this reason, the errors (which, after dithering, become the noise) of an ADC are examined in the presence of a low-level signal. This signal stimulates the lower-amplitude coding levels of the converter, which would produce the most audible artifacts if any errors were present.

Noise Weighting Filters

Weighting filters, which attempt to mimic some of the characteristics of human hearing, are often used in noise measurements with the intention of making the measurement reflect the audibility of the noise. There are several weighting filters in common use, all of which emphasize the frequencies toward the middle of the band at the expense of those at lower and higher extremes.

Figure 36. Noise measurement weighting curves.



Note: Conventionally, weighting filters are normalized in their overall gain so that they have 0 dB gain at 1 kHz. The CCIR-RMS filter is based on the application of the CCIR-468-4 curve in AES17. This is normalized for 0 dB gain at 2 kHz. (This was originally proposed by Dolby for use with an average responding detector and called CCIR-ARM). The unusual normalization is used to make the results closer to those of an unweighted measurement. The original CCIR-468-4 weighting curve is designed for use with the CCIR-468 quasi-peak detector.

The effect of these weighting filters on a measurement of flat noise is illustrated here:

20 kHz Band-Limited RMS Noise Measurements of a White Noise Source	
Unweighted	–0.07 dB
A-weighted	–2.33 dB
CCIR 468-4 weighted	7.01 dB
CCIR-RMS weighted	1.39 dB
F-weighted	2.46 dB

The noise floor of most converters is fairly flat, so these figures indicate the difference in results that might be quoted. The A-weighting gives the lowest noise figure and is normally the figure quoted on the front page of a data sheet. Where the noise is fairly flat you can add 2.3 dB to an A-weighted noise figure to estimate the unweighted noise over the DC to 20 kHz band.

Noise Measurement Using Quasi-Peak Metering

The CCIR-468 quasi-peak detector reads higher for noise sources than for sine waves of the same rms level. This is because noise sources have a higher crest factor, which is to say a higher peak amplitude, for a given rms level.

The following table illustrates the effect of the APWIN quasi-peak detector on the measurement of a properly dithered word-length reduction, using triangular probability distribution function (TPDF) dither. This is a typical dither used in digital audio systems and is representative of the noise source in many digital systems. The quasi-peak measurements are approximately 4.65 dB higher than the rms measurements. This adds to the effect of the CCIR 468-4 weighting filter to make a difference of 11.6 dB for noise with this property.

Quasi-Peak Measurements of TPDF Dithered Truncation	
Unweighted rms	–0.02 dB
Unweighted Q-peak	4.67 dB
CCIR-RMS weighted rms	1.36 dB
CCIR 468-4 weighted rms	6.99 dB (= CCIR-RMS + 5.629 dB)
CCIR 468-4 weighted Q-peak	11.64 dB

Note that in APWIN the “CCIR” weighting filter selection automatically switches between the standard CCIR 468-4 filter (normalized for 0 dB gain at 1 kHz) and the version normalized at 2 kHz, CCIR-RMS, which has 5.629 dB less gain. When **Q-peak** is selected as the detector the standard CCIR-468-4 filter is used, and when **RMS** is selected the CCIR-RMS filter is used. In other words, APWIN does not allow you to make a rms measurement directly using the standard CCIR-468-4 weighting intended for quasi-peak measurements. The value in the table has been calculated by adding 5.629 dB to the CCIR-RMS reading.

Idle Channel Noise

The idle channel noise of an ADC is measured with the input not driven by any source but connected to an electrical back-termination having the same impedance as a typical source. This would typically be 40 Ω for a balanced input

or 20 Ω for an unbalanced input. A short circuit might also give adequate results.

AES17 specifies measurement of idle channel noise with the CCIR-RMS weighting filter and a 20 kHz (or lower if specified) low-pass filter. Idle channel noise is measured in the procedure "a-d signal to noise.apb" alongside the signal-to-noise ratio. An unweighted result is also produced for comparison. Figure 37 shows the results of this procedure on the ADC within a portable DAT recorder.

```
=======================================
Idle channel and signal to noise ratio
=======================================

 CCIR weighted RMS measurements
 Signal-to-noise ratio          -85.46 dB FS CCIR-RMS
 Idle channel noise             -85.57 dB FS CCIR-RMS

 Unweighted measurements
 Unweighted signal-to-noise ratio       -84.82 dB FS
 Unweighted idle channel noise          -84.08 dB FS
```

Figure 37. Results of "a-d signal to noise.apb."

The idle channel noise measurement is not very useful for testing ADC performance. It is not representative of normal operating conditions and can produce erratic results.

For a successive-approximation converter, idle channel noise measurement does not exercise many of the conversion codes of the converter. The codes that are exercised depend critically on DC offset, and so may offer very inconsistent results.

For a delta-sigma converter this technique is also not very useful. Delta-sigma converters can have idle tones that fall at frequencies determined by the DC offset into the converter. For an ADC with an analog DC blocking stage it is difficult to exercise many DC levels.

However, idle channel DC conditions can be used to study idle tones by taking an FFT spectrum of the ADC output under idle channel conditions.

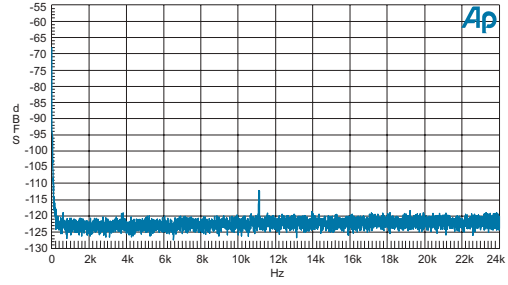
Figure 38, produced by "a-d idle channel FFT.apb," shows an FFT of the output from the DAT recorder in the idle channel state with an idle tone around 11 kHz at –112 dB FS. This disappears from the FFT when a signal is applied.

Note: Some delta-sigma ADC chips incorporate recommended circuits that manipulate the DC on the ADC input to move idle tones out of the audio band in tests like this.

Signal-to-Noise Ratio and Dynamic Range

To avoid the shortcomings of the idle channel noise measurement technique, the conventional approach to ADC noise measurement is to stimulate the converter with a low-level tone (typically –60 dB FS) and then remove the tone from the output with a notch filter. The remaining signal is then low-pass filtered to limit the bandwidth to 20 kHz. The resulting amplitude is measured in dB FS. (Ratios are normally quoted in dB, but as this is with respect to digital full scale dB FS is used.)

Figure 38. FFT of idle channel measurement.



AES17 defines a measurement of signal-to-noise ratio which uses the CCIR-RMS weighting filter in measuring the result. In the case of the portable DAT recorder this gives a result of –85.46 dB FS CCIR-RMS.

This measurement is sometimes called *dynamic range,* and that is the term used for a similar measurement defined in IEC 61606.[2] In that standard, which actually applies only to digital-input / analog-output testing in its current edition, the result can be measured using either an rms detector with A-weighting or with the CCIR 468 detector and CCIR 468-4 weighting.

Unweighted measurements are also often used, although that is not supported by any standard (the unweighted signal-to-noise ratio of the DAT recorder was –84.82 dB FS). Engineers may also have reasons for using the measurement without the low-pass filter.

In summary, take care when using or quoting *signal-to-noise ratio* and *dynamic range* in this context. The measurement is useless without knowledge of the bandwidth, weighting filter, or detector that is used.

The notch filter can be the same as that used for a THD+N measurement (see later) but the result must be quoted as an amplitude relative to full scale, rather than a ratio to the stimulus tone. If the reading is expressed as a ratio (as is often quoted in data sheets) then you should subtract the level of the tone to get the correct result.

For example, the Crystal Semiconductor CS5396 data sheet quotes the following characteristics (at 48 kHz sample rate in 128X oversampling mode):

Crystal CS5396		
Dynamic Range	A-weighted 20 kHz bandwidth	120 dB
Dynamic Range	20 kHz bandwidth	117 dB
THD+N	997 Hz at –60 dB FS and 20 kHz bandwidth	57 dB

Note that the THD+N performance at –60 dB is quoted as a ratio, but if you subtract that ratio of 57 dB from the tone amplitude of –60 dB FS the result of –117 dB FS matches (apart from the sign) the "dynamic range" quoted over a 20 kHz bandwidth.

"Number of Bits"

In the audio industry the discussion about the performance of a product often focuses on the "number of bits" that a product "has." There are multiple meanings being implied for the "number of bits," so that in addition to the word size used for storage or transmission of digital audio data, it is also assumed that it relates to the performance of the equipment. Often the short-form description of a product mentions the "number of bits" rather than any other aspect of performance.

An ideal ADC with a flat noise floor will have the same noise as a dithered quantization at a word-length of that number of bits. (This allows no room for any internal noise, but this is an *ideal* ADC.) The noise is spread over the band from DC to the folding frequency and can be determined using the following equation:

$$Ideal\ \ Noise = 10 \cdot \log\left(2^{(1-2 \cdot N)}\right) \text{dB FS.}$$

This formula is based on an *N*-bit conversion with no errors apart from the noise of a quantization that uses unshaped TPDF dither of 2 LSBs amplitude peak-to-peak. Applying this formula to a 16-bit converter will produce a figure for an unweighted signal-to-noise ratio of 93.32 dB FS, measuring the noise from DC to half the sampling frequency.

The proportion of this noise that falls within a 20 kHz bandwidth will scale with the sampling frequency, F_S:

$$Ideal\ \ Noise(\text{DC to } 20\,\text{kHz}) = 10 \cdot \log\left[\frac{20\,\text{kHz}}{0.5 \cdot F_S}\right] + 3.01 - 6.02 \cdot N\ \text{dB FS}.$$

Unweighted 20 kHz Noise Floor of "Perfect" *N*-bit ADC			
Number of bits	*FS* = 44.1 kHz	*FS* = 48 kHz	*FS* = 96 kHz
16	–93.73 dB FS	–94.10 dB FS	–97.11 dB FS
20	–117.81 dB FS	–118.18 dB FS	–121.19 dB FS
24	–141.89 dB FS	–142.26 dB FS	–145.27 dB FS

Excess Noise for N-Bit Converter

The ideal noise formula could be used for comparison of a real ADC noise floor. The excess of the noise of the converter over the ideal noise for the specified word-length could then be quoted.

For example, an ADC that has a 48 kHz, 24-bit output and a 20 kHz unweighted signal-to-noise ratio of 117 dB could be said to have an excess noise of

$$142 - 117 = 25\ \text{dB}$$

compared with an ideal 48 kHz 24-bit converter.

Significant Bits on Output

However, that does not mean that it is only actually a 20-bit converter. It is likely that the lower four bits of the 24-bit output are required to achieve this performance. If the 24-bit data word on the output was truncated to 20 bits then it is likely that the noise floor would rise further. Quantization distortion would also be produced if the truncation was not dithered.

A white-TPDF-dithered truncation to 20 bits will add noise at –118.18 dB FS, and this would add to the original –117 dB FS noise floor:

$$10 \cdot \log\left[\text{antilog}\left(\frac{-117}{10}\right) + \text{antilog}\left(\frac{-118.18}{10}\right)\right] = -114.5\,\text{dB FS}.$$

It would be possible to reduce this noise penalty by reducing the dither on the basis that the noise of the 24-bit output of the converter would make an adequate dither, or by shaping the re-quantization, but the noise would still in-

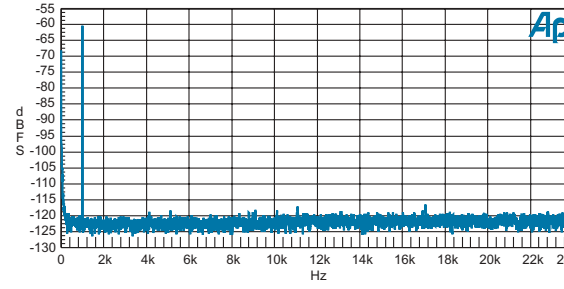
crease to show that these lower four bits were making a significant contribution to the performance.

It is possible to assess the value of the least significant bits by taking a measurement of signal-to-noise ratio and examining it for low-level non-linearities. If the noise rises, or if spurious spectral components appear on the truncated output in the presence of a low level signal, then the bits are significant. See **Low-level non-linear behavior**, page 66.

Noise Spectrum

Figure 39 shows an FFT of the output of the portable DAT recorder, using the same test signal as the signal-to-noise ratio measurement. The FFT was transformed from 16384 points and power-averaged 16 times. The Blackman-Harris 4-term window was used.

Figure 39. FFT of signal-to-noise test output , linear axis.



This figure is an APWIN plot that has been made using a linear frequency scale with the same number of points as FFT bins, which makes it possible to estimate the mean level of the bins in the noise floor at about –122 dB FS. (When the number of plotted points does not equal the number of bins, the APWIN plotting routines plot the highest valued bin for each point where more than one bin was present, and this would skew this visual estimate of bin mean level).

The conversion factor to calculate noise density for this FFT using the Blackman-Harris 4 window is:

$$\begin{aligned}&\text{Noise density correction}\\&= 10\cdot\log\left(\frac{1}{\mathit{Window\,Scaling}}\cdot\frac{\mathit{FFT\,points}}{\mathit{Sampling\,Frequency}}\right)\text{dB}\\&= 10\cdot\log\left(\frac{1}{2.004}\cdot\frac{16384}{48000}\right)\\&= 7.7\,\text{dB}.\end{aligned}$$

The FFT shows low-frequency noise and some discrete components at up to –116 dB FS.

The noise over most of the graph is about in line with the –122 dB FS on the Y-axis. Using the conversion factor this corresponds with:

$$10 \cdot \log\left(Noise\ Density\right) \text{dB FS} = -122 - 7.7 \text{dB FS}$$
$$= -129.7 \text{dB FS}.$$

This noise density, if it were constant over a 20 kHz bandwidth, would correspond with an unweighted noise of:

$$Noise\,(20\,kHz)$$
$$= 10 \cdot \log\left(Bandwidth\right) - 10 \cdot \log\left(Noise\ Density\right) \text{dB FS}$$
$$= 43 - 129.7 \text{dB FS}$$
$$= -86.7 \text{dB FS}.$$

This compares with the –84.82 dB FS measurement reported for the signal-to-noise ratio. The 2 dB difference corresponds with low-frequency noise. As confirmation of this, the difference disappears when the 100 Hz high-pass filter is selected for the signal-to-noise measurement.

The noise floor is basically flat above 200 Hz, but it shows a small increase in the noise density of about 1 dB from 2 kHz to 22 kHz. This could be an effect of the noise-shaping curve of the delta-sigma modulator in the ADC, or it could indicate some shaping of internal dither or quantization noise in the decimation filter after the modulator. The discrete spurious components seen between –119 and –116 dB FS at 5 kHz, 11 kHz, 13 kHz and other frequencies may be idle tones. See **Low-level non-linear behavior**, page 66.

The low-frequency noise contribution is much clearer if graphed on a logarithmic frequency axis. This is shown in Figure 40, which is the same data from Figure 39 re-plotted on a log scale.

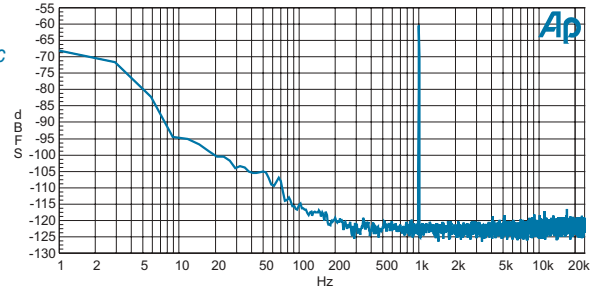


Figure 40. FFT of signal-to-noise test output, logarithmic axis.

In Figure 40 the lower-frequency limit has been selected to be 1 Hz. The FFT bin width for a 48 kHz, 16384 point FFT is 2.93 Hz, and on a logarithmic scale the first three points—DC, 2.93 Hz and 5.86 Hz—occupy a significant proportion of the graph. The high amplitude shown at these points is due to the broadening of the DC bin by the window function, and does not indicate low-

frequency noise. The true low-frequency noise spectrum is begins at about 10 Hz. At that point the noise is about 22 dB above the noise floor at higher frequencies.

DC Offset

The DC offset that is indicated on this FFT can be accurately measured using a DC averaging meter, which is available in APWIN by selecting "DC only" for the coupling on the Digital Analyzer panel. For the portable DAT recorder under test in this illustration, the DC level reads –72.4 dB FS.

In APWIN, the generator settings for DC offset and the analyzer measurements for DC level are relative to the full-scale DC value. Full-scale DC has a level 3 dB higher than a full-scale rms sine wave—which is the defined reference level for dB FS. Consequently, DC settings and readings appear 3 dB lower than the equivalent dB FS (RMS) values. However, DC has the same value as the *peak* level of a full-scale sine wave, so the APWIN values for DC offset are correct for dB FS (peak).

This statement may seem mathematically strange, as the numerical rms value and peak value of a DC level are obviously the same. However, the dB FS (RMS) measurement is defined as the ratio of the rms level of the signal being measured against the rms level of a full-scale sine wave—which is numerically $1/\sqrt{2}$. The rms level of DC at digital full scale is therefore 3 dB above the rms level of a full-scale sine wave, and reads +3 dB FS (RMS). The peak level of full-scale DC is the same as for a sine wave, and so full-scale DC reads 0 dB FS (peak).

```
=====================================================
Total harmonic distortion and noise
=====================================================
Measured as an amplitude

 THD+N at 997.00 Hz and −1.01 dB FS is −105.14 dB FS
```

Figure 41. THD+N Results from "a-d THDandN.apb."

The FFT DC Bin

When considering peak values the FFT over-reads the amplitude of DC components by a factor of two, or 6 dB. If considering rms values, this over-reading is reduced to 3 dB.

High-Level Non-Linear Behavior

Tests for the high level non-linear behavior of an ADC are similar to those for non-linearities in analog electronics, using standardized tests for harmonic distortion and intermodulation distortion.

Harmonic Distortion (THD+N)

Deviation from non-linear behavior can be simply investigated using a pure tone. Any non-linearity in the transfer function of the ADC will result in frequency components in addition to the tone. Static non-linearities (those that depend only on the signal) will result in harmonic products at multiples of the original tone frequency.

Total harmonic distortion plus noise (THD+N) can be measured at various levels and frequencies of input tone. The most conventional measurement uses an input level of –1 dB FS and a frequency of 997 Hz, with the output measured after a series of two filters: a notch filter (to remove the stimulus tone) and a low-pass filter (to limit the bandwidth to 20 kHz). Figure 41 shows the result of applying this measurement to a good quality 24-bit 96 kHz converter.

The procedure "a-d THDandN.apb," made this measurement. The procedure also sweeps against frequency and against level, saving the two plots (Figures 42 and 43) as test files.

Figure 42. THD+N by input tone frequency. black is at –1 dB FS; gray is at –20 dB FS.

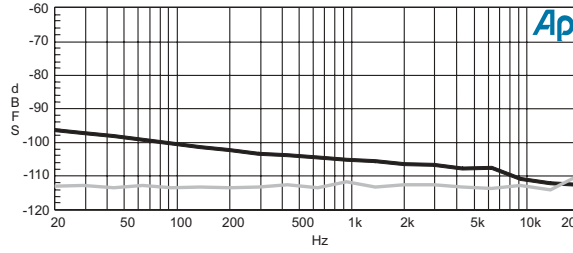
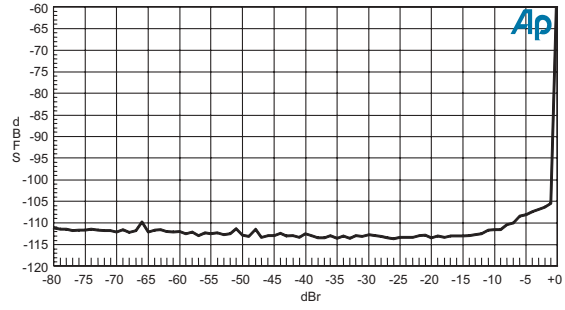


Figure 43. THD+N by level for a 997 kHz tone.



Measurements of THD+N made according to AES17 are to be quoted as a ratio to the unfiltered output signal level. However, typical harmonic distortion

is so low in good quality digital audio systems that the noise level becomes significant and often dominant for all signal levels except close to full scale. As the noise level does not scale with signal level, reporting the THD+N measurements as a ratio to the signal level makes the numerical result vary in inverse proportion to the input tone level.

The alternative shown here plots the results as a level (in dB FS) and not a ratio (in dB), and shows more clearly when the result departs from the noise floor at input levels above –15 dB FS.

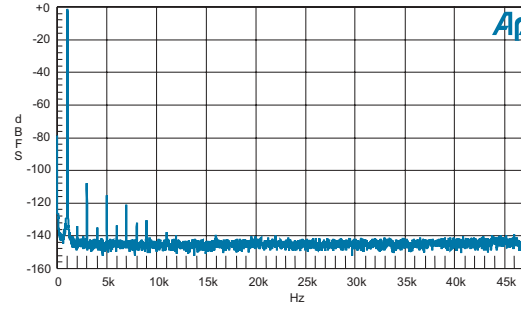
This is particularly important for the plot of THD+N versus level in Figure 43, which shows that below –15 dB FS the measurement is fairly constant. We can conclude that this is the noise floor of the device and that the harmonic distortion components are not significant in the measurement. This plot also reveals that the noise floor rises slightly toward lower input levels. This effect would not be very clear if the plot had a basic 6 dB per octave downward slope.

For many systems the odd harmonics are dominant, and in these cases it is important to measure the third harmonic. For digital audio systems which are band-limited to 20 kHz, this test is not capable of revealing the third harmonic distortion products for tones above 6.7 kHz. This can be observed for the –1 dB FS tone amplitude in Figure 42; and it can be a problem if you wish to measure non-linearity due to slew-rate limiting, for example. In a 20 kHz band-limited system the lowest odd and even harmonics are lost for input tones above 6.7 kHz (for the third harmonic) or 10 kHz (for the second harmonic).

Note: The intermodulation distortion (IMD) measurement, described below, can measure non-linearities with high frequency signals.

It is useful to examine the FFT amplitude spectrum for specific input conditions. The trace in Figure 44 corresponds with the THD+N reading of –105.14 dB FS with a 997.00 Hz stimulus tone at –1.01 dB FS.

Figure 44. FFT of THD+N test output.



This graph was produced using "a-d THD_FFT.apb." The equiripple window was chosen as it has the lowest close side-lobes. The FFT length is 16384 points and the plot is the result of power averaging over eight acquisitions.

The graph reveals that the odd harmonic components at 3, 5 and 7 kHz are a much higher levels than the even harmonic components at 2, 4 and 6 kHz, which leads to the conclusion that the high-level non-linearity is a result of symmetrical mechanisms (which produce odd harmonics). The dominance of the third harmonic confirms the indication inferred from the dip after 6.7 kHz in Figure 42, as the third harmonic of input frequencies above 6.7 kHz will fall outside the 20 kHz measurement band.

This plot can also be used to estimate the uncorrelated noise level as distinct from the discrete harmonic components. The number of points in the plot matches the number of points in the FFT output, so every FFT point has been plotted. This means that an estimate of noise density can be made. The window scaling factor for the equiripple window is 2.63191, and sample rate is 96 kHz, so the factor for conversion to noise density is:

$$\begin{aligned}&\text{Noise density correction}\\&= 10\cdot\log\left(\frac{1}{\textit{Window Scaling}}\cdot\frac{\textit{FFT Points}}{\textit{Sampling Frequency}}\right)\text{dB}\\&= 10\cdot\log\left(\frac{1}{2.63191}\cdot\frac{16384}{96000}\right)\\&= 11.9\,\text{dB}.\end{aligned}$$

The noise floor appears at an average level of –145 dB FS on the Y-axis. Adding the noise density correction, this corresponds to a noise density of –156.9 dB FS / Hz. Multiplied over the 20 kHz bandwidth (by adding 10log(20k) = 43 dB), this noise density translates to an unweighted noise level of –113.9 dB FS.

Intermodulation Distortion (IMD) Tests

Another conventional method of measuring non-linearity is to use two input tones and measure the discrete intermodulation products that are produced. This is a twin-tone intermodulation distortion (IMD) test.

For a pair of frequencies, F_1 and F_2, the effect of non-linearities is to produce harmonic and intermodulation products at the following frequencies:

Order	Harmonic	Intermodulation Difference	Intermodulation Sum
2nd	$2F_1$, $2F_2$	F_1-F_2	F_1+F_2
3rd	$3F_1$, $3F_2$	F_1-2F_2, $2F_1-F_2$	F_1+2F_2, $2F_1+F_2$
4th	$4F_1$, $4F_2$	F_1-3F_2, $2F_1-2F_2$, $3F_1-F_2$	F_1+3F_2, $2F_1+2F_2$, $3F_1+F_2$

For testing bandwidth-limited equipment (such as an ADC) using high-frequency stimulus signals, the intermodulation distortion test offers advantages compared to the harmonic distortion test. The THD test distortion products are always at higher frequencies than the stimulating tone, and for high-frequency stimuli will fall outside the device passband. IMD test distortion products, though, include components at lower frequencies than the stimulus, and these components will fall within the passband and can be measured. The level of these IMD products will reveal problems with non-linearities, such as slew-rate limiting, that are only significant with high-frequency stimuli.

There are several styles of twin-tone signals. The SMPTE RP120-183 and DIN45403 tests each use one high and one low frequency. The AES17 standard IMD test signal uses two high frequencies, one at the "upper band edge" frequency (normally 20 kHz), and another at 2 kHz below that frequency. (For most systems the upper band edge is defined in AES17 as 20 kHz, but it may be lower than this for systems with sample frequencies less than 44.1 kHz.)

The level of the twin-tone is specified for the AES17 test to peak at full scale. This is an rms level of –6.02 dB FS for each tone, with a total rms level of –3.01 dB FS.

20 kHz and 18 kHz input tones will produce the following intermodulation difference frequencies:

Order	Intermodulation Difference	Actual Frequencies, if F_1=20 kHz & F_2=18 kHz
2nd	F_1-F_2	2 kHz
3rd	$F_1-_2F_2$, $2F_1-F_2$	16 kHz, 22 kHz
4th	$F_1-_3F_2$, $2F_1-_2F_2$, $3F_1-F_2$	34 kHz, 4 kHz, 42 kHz

The in-band products up to the fourth order are at 2 kHz, 4 kHz and 16 kHz. AES17 specifies that the measurement is of the ratio of the total output level to the rms sum of the second- and third-order difference frequency components on the output.

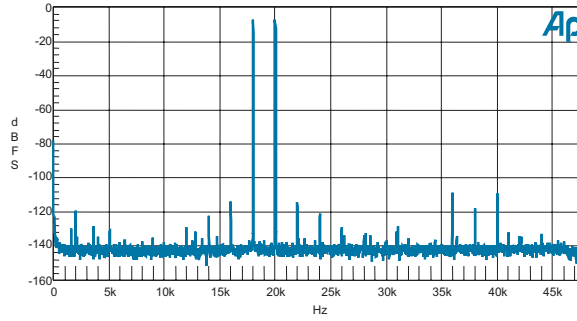


Figure 45. FFT of IMD test output, with 18 kHz and 20 kHz input.

Figure 45 illustrates the output of "a-d IMD_FFT.apb." This procedure calculates the IMD product amplitudes from the FFT by summing the spectral power density around each component.

Note: An alternative method of amplitude estimation from an FFT is to use a "flat-top" window and examine the height of the relevant peak in the spectrum. This method can be used when you do not have access to the raw FFT data. It is not as accurate as the previous method, since the height of the peak depends on the exact relation between the frequency of the component being measured and the frequency corresponding to the FFT bin closest to it. The flat-top window has a lower sensitivity to this than other windows but at the price of higher side-lobes, which may affect the accuracy of measurements of the low-level frequency difference components.

Overload Response

It is important that when an input signal exceeds the full-scale range the errors that are produced are as benign as possible. This is especially true in some audio applications, such as broadcast, when there may be no opportunity to retry at a lower gain setting.

An example of non-benign behavior under overload conditions is inversion in the digital output signal. In the early years of digital audio some systems were liable to do this. The numeric processing would "wrap" from positive full scale to negative full scale as a result of the most significant bit inverting.

In recent times it is more widely understood that numeric processes should be designed to prevent overload behavior by limiting the signal to the full-scale level rather than allowing it to wrap. However, it is still quite possible for a software coding error to produce this problem.

Delta-sigma converters can also have non-benign overload characteristics as they can become unstable in overload conditions.

An overload response test is still useful in examining these conditions. The standard overload test in AES17 is performed by making a THD+N measurement with an input sine wave 3 dB above full scale (+3 dB FS) and another at –3 dB FS and reporting the difference between the two measurements. The sine wave frequency is normally be at 997 Hz, but other frequencies can be used to investigate any frequency dependence. The procedure "a-d overload distorton.apb" performs this test.

Out-of-Band Overload Behavior

Some ADC architectures are prone to non-benign overload modes for signals outside the audio band. In these cases overload behavior for frequencies in the anti-alias filter stopband frequency range should be investigated.

Low-Level Non-Linear Behavior

Noise or distortion that is present with low signal levels is in many ways more objectionable than the harmonic or intermodulation distortion resulting from high-level non-linearities.

Linear digital audio processes, including ADCs and DACs, should have an output signal that is linearly related to the input signal plus a random error term. The error term should be uncorrelated with the input signal.

When low-level signals are quantized the error is highly correlated with the input signal. In delta-sigma converters, the error can have strong discrete frequency components at a frequency related to the instantaneous, or DC, level of the signal.

Dither can be used to de-correlate the quantization error from the signal. Ideally the dithered quantization error is randomized so that it is has the character of white noise at a constant level. The ideal application of dither at all possible stages is not practical, so compromises are made.

Quantization Distortion

A sine test signal at a very low frequency can be used to stimulate most of the levels in an ADC. If the output of the ADC is filtered so that the main tone and principal harmonics are not present, the remnant can give an indication of quantization distortion. A "quantization distortion" measurement following this approach was proposed in the past, using a notch filter to attenuate the main tone by over 80 dB with an additional high-pass filter to take out the harmonics. The low frequency tone was to be 41 Hz and the filter corner frequency was set at 400 Hz.

Although this test is no longer recommended by any measurement standards, it is occasionally referred to and is still honored by the 400 Hz high-pass filters found in some test equipment.

Truncation Artifacts

The error produced by inadequate dither at a quantization—and this can occur at any of several points within an ADC—is correlated with the data bits of lower significance. These bits have a poor correlation with the signal when the signal level is high, but they have a high correlation when the signal is low.

This high correlation will result in artifacts at discrete frequencies that are harmonics, and sometimes aliased harmonics, of the stimulus frequency. There is no standard that specifies the measurement of this effect.

The harmonics resulting from truncation can be observed, for example, in the spectrum of the output of a device stimulated by a low level tone, such as the –60 dB FS tone of the signal-to-noise measurement discussed earlier and illustrated in Figure 39. There are some discrete frequency components shown, but none are harmonically related to the original tone. Sometimes it is necessary to average the FFT spectrum a large number of times to smooth out the representation of the noise floor so that discrete components will be more obvious.

Figure 46. FFT of signal-to-noise test output averaged 256 times.

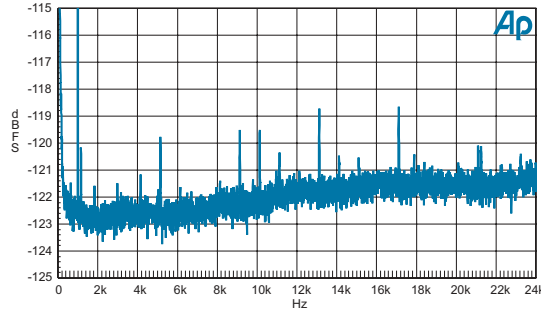


Figure 39 shows the output of the signal-to-noise test averaged 16 times. Figure 46 illustrates the effect of averaging the test of 256 times, with the vertical scale expanded considerably. The spread in the noise-like part of the spectrum has been reduced by the averaging and, as a result, the discrete components have become more obvious. These components are more than 30 dB below the level of the integrated total noise, so they do not represent a significant problem. On inspection, none of the discrete components appears to be harmonically related to the main stimulus tone at 997 Hz. Therefore, even at this magnification, we can still conclude that this converter does not suffer from the artifacts due to simple truncation distortion.

Noise Modulation

It is possible for noise or dither to have decorrelated the truncation error from the input signal, but not decorrelated the truncation error *power* from the input signal. For example, truncation error power might be maximized when

the mean signal level is centered on a quantization step, while it would be minimized if the signal is centered *between* quantization steps.

This correlation of truncation error power with signal is a form of noise modulation.

A simple test for this might be to measure the noise or noise spectrum for various DC levels in the ADC. However, since ADC inputs usually have DC blocking filters, it is normally not possible to control the DC level in the converter.

An alternative to trying to manipulate the DC level in the ADC is to stimulate the ADC with tone signals of various amplitudes. However, this approach will not give results as clear as varying the DC level.

The broadband noise variation is of interest, but the variation of the noise spectrum will often be more revealing. This can be examined using either a swept bandpass filter or an FFT approach.

A third-octave bandwidth is appropriate for the swept bandpass filter measurement, as it scales in bandwidth with frequency and in this respect it is similar to the width of the auditory filter that detects noise. The maximum variation in noise for each third-octave frequency should be noted.

In the FFT approach, if the variation in noise level is small then it may be swamped by the statistical variation of the FFT noise floor. In this case it is possible to use FFT power averaging to reduce the statistical variation.

Jitter Modulation

Jitter is the error in the timing of a regular event, such as a clock. The *intrinsic jitter* of a device is the element of jitter that is independent of any external clock synchronization input, and the *jitter transfer function* indicates the relation between an external synchronization input and the jitter of the device.

The jitter of the clock that determines the ADC sampling instant—which is called *sampling jitter*—is the only jitter that has any effect on analog-to-digital conversion performance. Jitter in other clocks may or may not be indicative of the jitter on the sampling clock.

The direct connection of test probes to a sampling clock inside a particular ADC might be possible, but measurements using this technique are beyond the scope of this article. Instead, we will measure the effect of the jitter on the audio signal.

The theory of sampling jitter in an ADC is discussed in detail in reference.[3]

Synchronization Jitter Susceptibility

An ADC can be prone to jitter that is received on a timing synchronization input. Even though the clock recovery circuits will normally have an element of jitter attenuation, some of this *synchronization jitter* can be transferred to the derived sampling clock. This jitter attenuation characteristic will determine the synchronization jitter susceptibility of the ADC.

> *Note: an ADC that is not capable of external synchronization will not, of course, be susceptible to synchronization jitter.*

Sampling Jitter Transfer Function

The most useful method of measuring the jitter susceptibility is through the *sampling jitter transfer function*.

A procedure for measuring jitter transfer is supplied as "a-d JTF.apb." A 20 kHz tone at –1 dB FS is used as the audio stimulus, while jitter is applied to the ADC synchronization reference using the System Two Cascade sync output BNC. The jitter is in the form of a sine wave with a peak level of 0.125 unit intervals (UI). The frequency of the jitter is swept over a range defined by the following constants defined in the procedure:

```
Const N_frequencies = 10
    Const StartFreq = 100
    Const EndFreq = 39e3
```

At each jitter frequency, the amplitude of the lower-frequency jitter modulation sideband is measured. It is important to have good frequency resolution for this measurement, as the sidebands due to low jitter frequency components will be close to the 20 kHz tone. The measurements are taken from an FFT using a high dynamic range window and the integration technique described for the IMD measurement.

Figure 47. FFT for measurement of jitter transfer function at a jitter frequency of 10357.9 Hz, with gain of –0.7 dB.

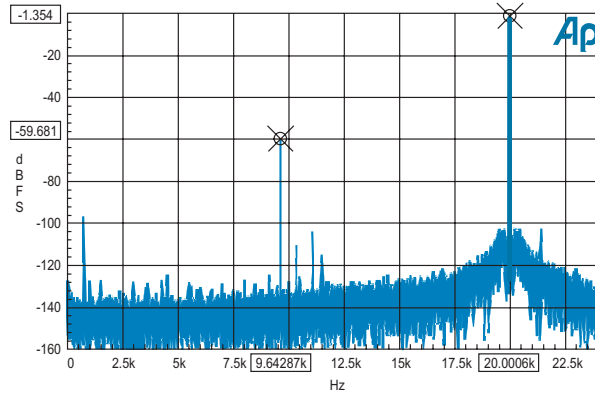


Figure 47 illustrates one of the FFT traces. The cursors highlight the main component at 20 kHz and the jitter sideband at 9.64287 kHz. The sideband amplitude is first calculated from theory using the amplitude of the applied jitter and the stimulus tone frequency. The difference between the calculated level and the actual level is then plotted as the jitter gain.

Note that the "skirts" around the main 20 kHz component are a byproduct of noise in the jitter generation mechanism and do not represent jitter intrinsic to the converter under test. These skirts disappear when the jitter generation is disabled.

Figure 48. Jitter transfer function.

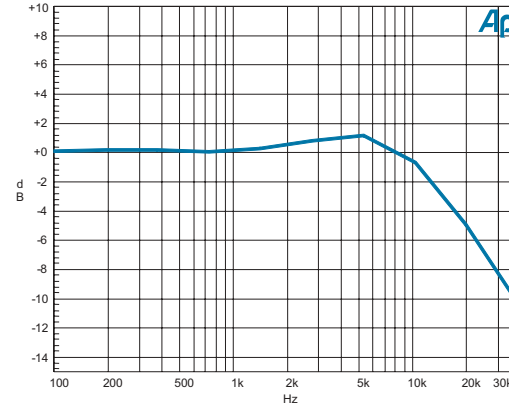


Figure 48 shows the total measured jitter transfer function using this procedure. You can see that there is between 1 and 2 dB of jitter peaking at 5 kHz, and jitter attenuation above 8 kHz. Above 20 kHz the slope is about 6 dB per octave, which indicates a first-order response. More measurements could be made near the 5 kHz point to be assured that the jitter peaking is not much worse than 2 dB, but the main conclusion is clear: this device does not have significant audio-band jitter attenuation. This compares with other converters which have as much as 60 dB attenuation at 500 Hz to ensure that modulation sidebands cannot approach audibility.

For an ADC, the upper jitter frequency limit is set by the maximum sideband offset that can be achieved within the audio band. In the case of a 20 kHz bandwidth system, the maximum frequency offset is just under 40 kHz with the stimulus tone at 20 kHz. The highest frequency plotted by this procedure is 39 kHz. This produces a sideband at

$$|20\,\text{kHz} - 39\,\text{kHz}| = 19\,\text{kHz}.$$

The lower jitter frequency limit is set by the frequency resolution of the FFT. This procedure uses a 32768 point FFT with an equiripple window, which limits the lower jitter frequency measurement at about 15 Hz. The jitter frequency range selected for this measurement has a lower limit at 100 Hz, so

in this case the number of FFT points could be reduced to 8192 with the benefit of increased processing speed.

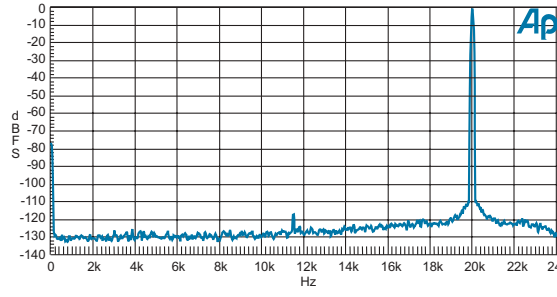
Intrinsic Jitter Artifacts

Intrinsic jitter will produce increased noise in the presence of high-frequency and high-level signals. The procedure "a-d intrinsic jitter.apb" uses this characteristic to estimate the amount of jitter that would produce such a noise floor. It should be used with care, since it does not determine the source of the noise. However, it does provide an upper limit on the amount of intrinsic sampling jitter that is present.

The procedure uses the same high-level stimulus tone as used in the previous measurement of jitter transfer function. An FFT of the converter output is then computed and drawn with one bin per plotted point, as shown in Figure 49.

The high frequency (20 kHz) was selected to maximize the sensitivity to jitter. In some cases it is more useful to choose a frequency in the middle of the band (perhaps 10 kHz); then symmetry in the skirts would be an indicator that it was truly a modulation effect that is being observed.

Figure 49. FFT used for intrinsic jitter calculation.



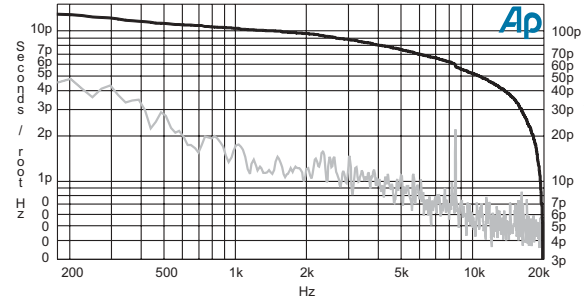
Notice that Figure 49 has some close-in skirts that appear to start from –110 dB FS and go down to –122 dB FS at 20 kHz, ±1 kHz. At higher offsets the slope is more relaxed. This slope could be a shaped noise floor that is *not* due to sampling jitter. To eliminate that possibility, the slope should be compared with the shape of the noise floor produced by a lower-amplitude or lower-frequency stimulus tone.

There is a peak at about 11.6 kHz. This may or may not be due to sampling jitter, so its effect on the total result should be considered as another "unknown" in the measurement.

The procedure measures the amplitude of each bin between DC and the stimulus tone, then calculates the frequency and level of the jitter required to produce this level through modulation of the 20 kHz tone. This is plotted in Figure 50 as (potential) intrinsic jitter versus jitter frequency.

The lower line on Figure 50 is shows jitter density. This line is calibrated in seconds (RMS) of jitter per root hertz on the left axis. This axis covers the range from 300 FS (0.3 ps) to 15 ps.

Figure 50. Calculated intrinsic jitter per root hertz, and integrated to 20 kHz.



The upper line is the integration of this jitter density, representing the total jitter measured from the frequency on the X-axis to the right-hand limit of the graph. This shows, for example, that the total jitter above 1 kHz is just over 100 ps, and above 200 Hz it is about 120 ps.

The amplitude of the discrete component that was noticed earlier—at an offset of 8.5 kHz—is not easy to determine from the noise density curve. The slight step in the integration curve that it produces shows that it is not very significant; so, the uncertainty about the cause of this component does not add a large uncertainty to the total result.

The speculative interpretation of the original FFT into sampling jitter should be treated carefully, but as an indicator of the maximum possible sampling jitter spectral density it is a very sensitive tool.

The Fourier Transform

The Fourier transform is a mathematical technique that converts a data block that represents a signal in the time domain, to a data block that represents the signal in the frequency domain. The most common method of performing the transform is known as the fast Fourier transform, or FFT. As a result, the abbreviation *FFT* is often used when the more general term *Fourier transform* would be applicable.

The frequency domain information is in the form of an amplitude and a phase value for each discrete frequency "bin." The transform preserves all the information about the signal, so it produces the same number of output values as input values. Each frequency bin (with the exception of the first and last bins at each end of the spectrum) has two values: real and imaginary; or, with an alternate representation, magnitude and phase. (The phase information has some specialist applications and will not be discussed here.)

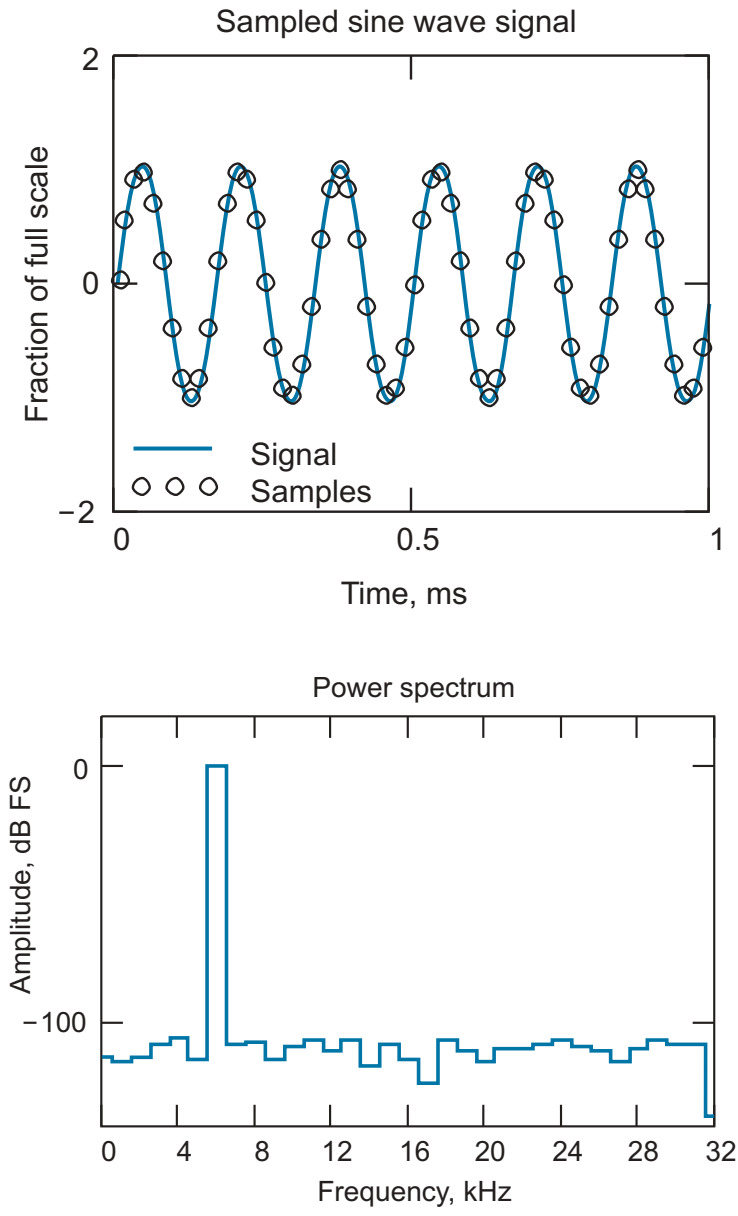


Figure 51. 64-point FFT with a 1 ms block length showing a 6 kHz sine with noise.

Figure 51 shows an example of a Fourier transform. The 64-sample input data block is shown in the top graph. For simplicity, the sample rate F_S is 64 kHz so that the block length is exactly 1 ms. The input signal is a sine wave with a small amount of white noise.

The magnitude of the transform output is shown in the lower graph. There is one bin at 0 dB FS, which corresponds to the input sine wave, and the other bins are less than –100 dB FS, corresponding to the white noise.

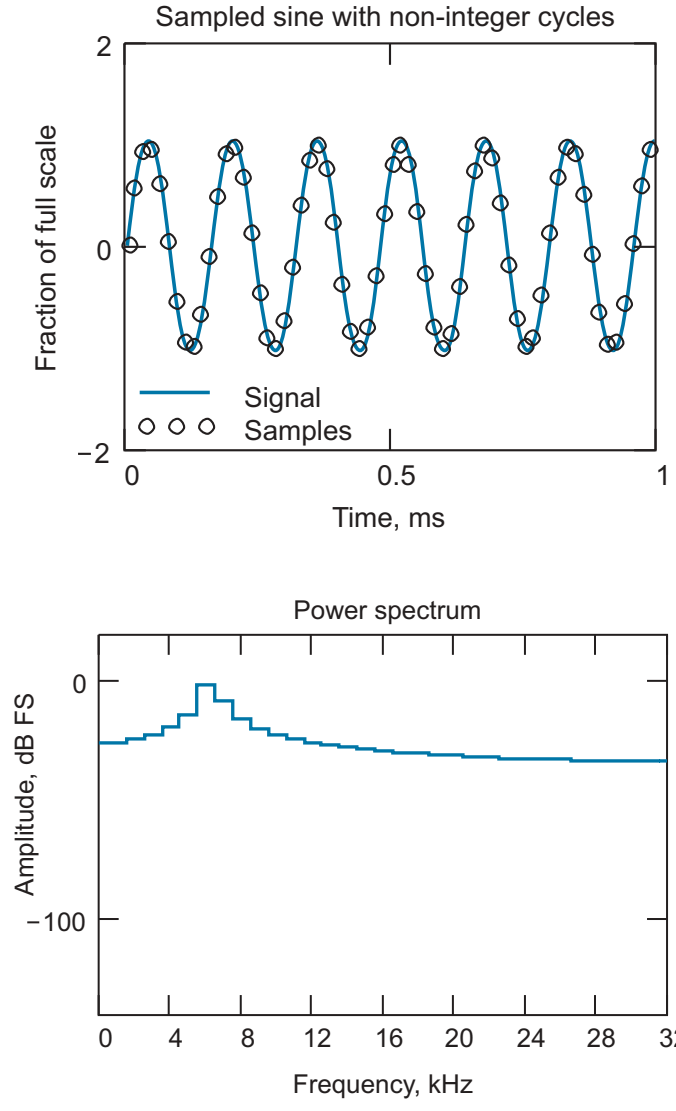


Figure 52. 64-point FFT with a 1 ms block length showing the leakage from a 6.3 kHz sine that does not repeat over 1 ms (no window).

Note that the frequency axis consists of 33 bins spread from DC (0 Hz) to $F_S/2$ (32 kHz). The DC and $F_S/2$ bins are at the end points of the spectrum and

consequently are half the width of the other bins, which are 32 kHz/32=1 kHz wide. (The $F_S/2$ bin is not very useful and is often ignored).

This example represents a special case where a integer number of cycles of the waveform fit exactly into the input data block. In the frequency domain, this means that the fundamental frequency is exactly centered on the bin corresponding to the number of cycles that the waveform has completed in the input data block length. Hence, the peak at bin number 6 in the previous figure.

The Fourier transform can correctly represent only a static signal. The 64-sample data block transforms to a frequency-domain representation of a static signal made by repeating the data block forever. In the audio measurements that use FFTs the signals normally used are fairly static: they do not last forever but they are stable for the duration of the measurement. In cases where the signal exactly repeats over the length of the data block, as in the example just illustrated, the transform will produce a good representation.

Windowing

Normally, the signal does not exactly repeat over the FFT block, and a discontinuity appears in the signal at the point where the data at the end of the buffer wraps into the data at the start of the buffer. This discontinuity transforms into the frequency domain and is likely to swamp the features of interest, as shown in Figure 52.

In this case there are 6.3 cycles of the sine wave in the 64-sample block. At the point where the end of the block wraps to the beginning, there is a large discontinuity. This discontinuity distorts the power spectrum so that the noise floor is swamped by wide skirts to the main spectral peak; this mechanism is called *leakage*.

Of the two techniques available, *windowing* is the most commonly used.

Windowing multiplies the input data block by one of several window functions that tapers the signal at both ends of the block and minimizes the discontinuity.

Figure 53 illustrates the use of a Hann window on the same signal as used in the Figure 52. This window function is one cycle of an inverted raised cosine, and, apart from a rectangular window (which is effectively *no* window) it is the simplest used. In this example, the Hann window is scaled so that it has a mean square value of 1, which preserves approximately the same power in the data block.

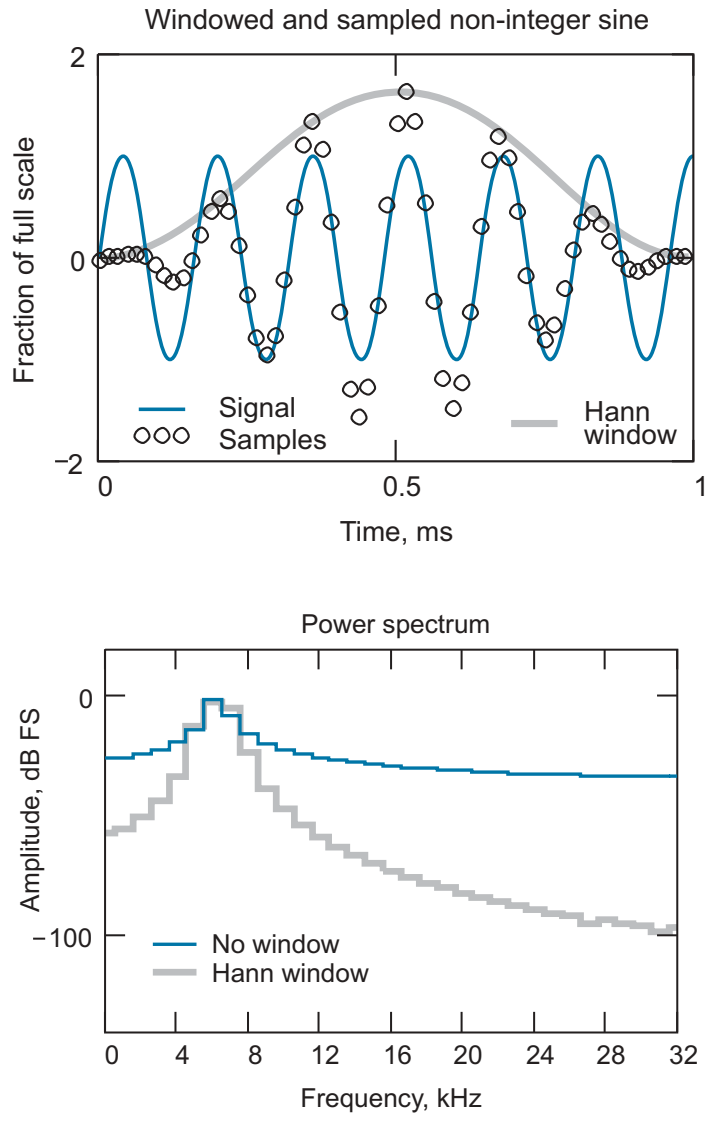


Figure 53. Non-synchronous, FFT of 1 kHz signal. Non-windowed (blue) compared with Hann window (black).

The power spectrum shown in the lower graph of Figure 53 displays the benefit of the Hann window in the much lower skirts. However, when compared with the synchronous FFT, the effect has been to broadening the spectral peak and add skirts where the power in the main lobe has "leaked" to nearby bins.

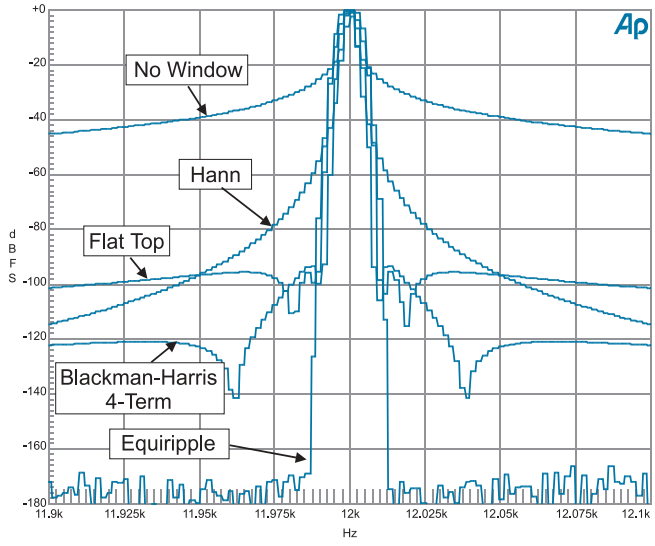


Figure 54. Comparison of FFT windowing functions supplied with System Two Cascade.

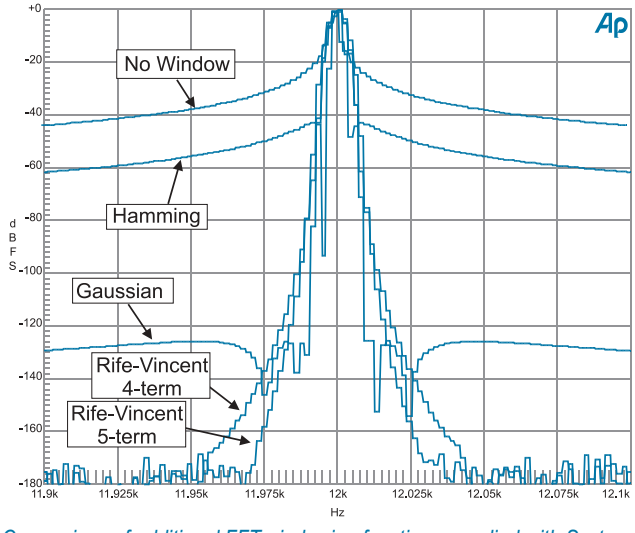


Figure 55. Comparison of additional FFT windowing functions supplied with System Two Cascade.

Several window functions are in common use, each representing a different compromise between frequency resolution and leakage. Figures and show examples of the windows supplied with the Audio Precision System Two Cascade.

Signal Frequency Post-Acquisition Scaling

Another technique to reduce leakage in an FFT is to modify the acquired signal so that it is stretched or compressed until it repeats an integer number of times over the FFT block length. This avoids the requirement for a window and so maintains a one-bin-wide resolution in the FFT. The function has been implemented in the Audio Precision DSP FFT program and is selected by choosing a window of **None, move to bin center**. The effect is to move the observed frequency of the sine wave (and its harmonics) to exactly align with the FFT frequency bins. Leakage into neighboring frequency bins is then almost entirely eliminated. The disadvantage of this technique is that it only works for sine waves.

Interpretation of Noise in FFT Power Spectra

The Audio Precision FFT spectrum analyzer is calibrated so that the amplitude axis gives the correct reading for sine waves. It is important to note that it cannot be used as an indicator of the level of spectrally non-discrete signals—such as noise—without applying a conversion factor that depends on the bin width and on the window used.

Conventionally, an audio FFT amplitude spectrum is displayed by scaling the vertical axis so that a bin peak indicates a value that corresponds to the amplitude of any discrete frequency components within the bin. This calibration is not appropriate for measuring broadband signals, such as noise power.

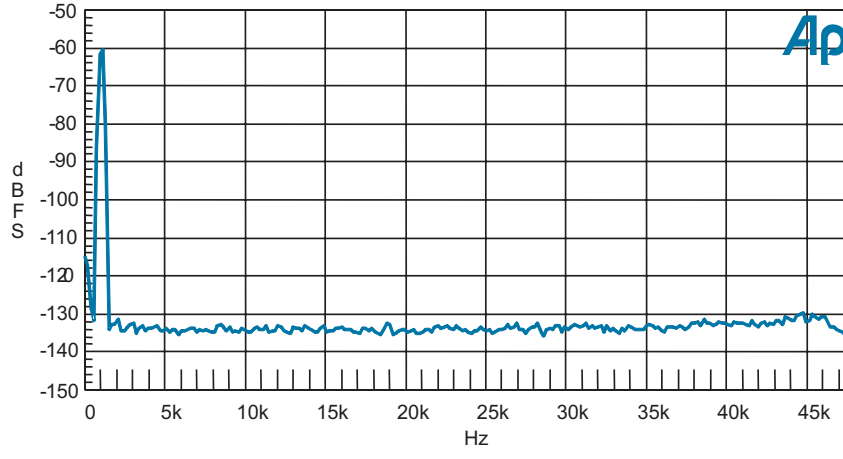


Figure 56. FFT spectrum with noise floor at –109 dB FS.

Figure 56 illustrates this for a high-performance ADC. This is a 1024-point FFT using a Blackman-Harris window, then power averaged (see below) so that it represents the mean result from 32 FFTs.

The window spreads the energy from the signal component at any discrete frequency, and the Y-axis calibration takes this windowing into account. For the Blackman-Harris window used here, the calibration compensates for the power being spread over a bandwidth 2.004 bins wide.

This can be converted to the power in a 1 Hz bandwidth, or the *power density*, by adding a scaling factor in dB that can be calculated as follows:

$$\begin{aligned}&\text{Conversion factor}\\&= 10 \cdot \log\left(\frac{1}{\mathit{Window\ Scaling} \cdot \mathit{Bin\ Width}}\right)\\&= 10 \cdot \log\left(\frac{1}{\mathit{Window\ Scaling}} \cdot \frac{\mathit{FFT\ Points}}{\mathit{Sampling\ Frequency}}\right)\\&= 10 \cdot \log\left(\frac{1}{2.004} \cdot \frac{1024}{96000}\right)\\&= 22.73\,\text{dB}.\end{aligned}$$

This scaling factor is for the FFT used in Figure 56, which uses a Blackman-Harris window, a 1024-point FFT and a sampling frequency of 96 kHz. Note that the calculation is in power terms so the ratio in dB is 10 times the logarithm of the ratios.

For some of the other windows used in the Audio Precision Systems FFT analysis the figures are:

Window	Scaling Bandwidth	Scaling Factor
None (rectangular)	1.00000 bins	0 dB
Hamming	1.36283 bins	1.34 dB
Hann	1.50000 bins	1.76 dB
Blackman-Harris-4	2.00435 bins	3.02 dB
Gaussian	2.21535 bins	3.45 dB
Rife-Vincent-4	2.31000 bins	3.63 dB
Rife-Vincent-5	2.62653 bins	4.19 dB
Equiripple	2.63191 bins	4.20 dB
Flat-top	3.82211 bins	5.82 dB

To estimate the noise from a device based on an FFT spectrum you can integrate the power density over the frequency range of interest. For an approximately flat total noise (where the noise power density is roughly constant) it is possible to estimate the sum of the power in each bin within reasonable accu-

racy, by estimating the average noise power density and multiplying by the bandwidth.

Figure 56, for example has a noise floor that is approximately in line with about –134 dB FS on the Y-axis. The conversion factor for this FFT was previously calculated as –22.7 dB, so the noise power density is:

$$-134\,\text{dB FS} - 22.7\,\text{dB} = -156.7\,\text{dB FS per Hz.}$$

The integration to figure the total noise over a given bandwidth is simple if the noise is spectrally flat. Multiply the noise power density by the bandwidth, which in this case is 20 kHz. For dB power (dB=10logX), this is the same as adding 43 dB, as follows:

$$\begin{aligned}\text{Noise} &= \textit{NoiseDensity}\,(\text{dB}) + 10\cdot\log(\textit{Bandwidth})\ \text{dB FS}\\ &= -156.7 + 10\cdot\log(20\,000)\\ &= -156.7 + 43\,\text{dB FS}\\ &= -113.7\,\text{dB FS.}\end{aligned}$$

Power Averaging

Power averaging is normally used to reduce the statistical variation of a noise floor. This is achieved by acquiring a number of FFT power spectra and computing the mean result for each bin. The noise in each bin is reduced to a statistical mean, and any spectrally discrete components (often called *spuriae*) will become more obvious.

This also makes it easier to visually estimate the amplitude of the noise floor using the technique described above.

Synchronous Averaging

It is possible to average the signal in the time domain before applying the transform. This *synchronous averaging* technique requires that successively acquired data blocks have their signals aligned in time before averaging. This can be done with a trigger, or by adjusting the timing of each acquired data block to match the previously acquired data. Either way, this technique will reduce the noise level *below* the statistical mean value, while preserving the level of components that are synchronized with the main (or trigger) signal. Synchronous averaging is used to find spectral features that are below the level of the noise. The indicated level of non-synchronous components, such as noise, is not significant.

List of Procedure Files

The following APWIN Basic procedures are referred to or used in this chapter:

- a-d tech note utilities.apb
- a-d input gain stability.apb
- a-d stopband.apb
- a-d antialias corner.apb
- a-d passband.apb
- a-d Input for full-scale.apb
- a-d max input level v freq.apb
- a-d signal to noise.apb
- a-d idle channel FFT.apb
- a-d THDandN.apb
- a-d THD_FFT.apb
- a-d IMD_FFT.apb
- a-d Overload distortion.apb
- a-d JTF.apb
- a-d intrinsic jitter.apb

Two additional files have been provided for consistency and ease of use:

- a-d Menu.apb
- a-d Setup.at2c

These files are provided on the companion CD-ROM. You may also download the files from the Audio Precision Web site at audioprecision.com. These procedures and tests are designed for use with System Two Cascade, but with minor changes can be modified to work with System Two as well.

Please check the README.DOC file in the same folder for further information.

References

1. AES17, “AES standard method for digital audio engineering—Measurement of digital audio equipment,” *J. Audio Eng. Soc.*, vol. 46 No. 5, pp. 428-447 (May 1998). [The latest version is available at aes.org.]

2. IEC 61606, “Audio and audiovisual equipment—Digital audio parts—Basic methods of measurement of audio characteristics,” Geneva, Switzerland: International Electrotechnical Commission (1997).

3. See the chapter **Jitter Theory** beginning on page 3 of this book.

Digital-to-Analog Converter Measurements

Introduction

The complexity of the mechanisms that affect modern audio digital-to-analog converter (DAC) performance means that the conventional measurement techniques developed for testing analog systems are sometimes inappropriate. Often, conventional approaches either are insensitive to the errors produced by these mechanisms, or they do not provide adequate information for diagnosing the cause of the errors. As a result, new measurement techniques have been developed, some of which have been standardized by the AES[1] and the IEC.[2]

This chapter shares much with the Analog-to-Digital Converter Measurements chapter,[3] and parts of that article should be used for reference. In particular, the sections **Level Measurements in the Digital Domain** (page 37) and **The Fourier Transform** (page 72) provide important background information to the measurement techniques described here.

Not surprisingly, many of the measurement techniques used for D-to-A testing are similar to those used for A-to-D testing. They require test signal generation and analysis in complementary domains, of course, but the results produced are of similar significance.

Some measurement techniques, though, are specific to D-to-A testing:

- In particular, the most common source of clock synchronization in a DAC is the clock embedded in the input signal, and the interaction between the data and the recovered sample clock jitter becomes significant.
- In other situations the test signal needs to be on a pre-recorded medium, such as a CD, DAT or DVD, so a new requirement may be that the test stimulus signal needs to be recorded in advance.
- The standard digital audio interface and pre-recorded media formats carry data about the audio signal. This data may define how the signal is

converted to the analog domain; the response to this data is also an aspect of testing that is new for this article.

Measurement Techniques

Notes on the APWIN Procedure Examples

In the following sections APWIN procedures are used to illustrate the measurement techniques. The procedure files (*.apb) are supplied on the CD-ROM included with this Application Note and are also available for download from the Audio Precision Web site at audioprecision.com. These procedures have been designed to be used with Audio Precision System Two Cascade but should illustrate the processes for other equipment.

Prior to running the procedures, the DAC to be tested and the APWIN Analog Analyzer and Digital I/O panels must be configured so that the DAC passes a signal from the Digital Generator output to the Analog Analyzer input. Any other settings that are required by the test will be configured by the procedure.

For any DAC under test the configuration only needs to be performed once, and the procedures should not alter it. The APWIN configuration for setting the interfaces for a specific DAC can, of course, be saved to a test file, which can then be loaded before using the procedures.

See the README.DOC file which accompanies the procedure files.

```
'Measure gain at -20dB FS
     LineupLevel = -20              ' dBFS
     LineupFreq = 997               ' Hz

'Set analogue reference for full scale level (based on digital output level)
     AP.S2CDsp.Analyzer.ChALevelTrig 'Reset ready count for new reading
     AP.S2CDsp.Analyzer.ChBLevelTrig 'Reset ready count for new reading
     LevelA = AP.S2CDsp.Analyzer.ChALevelRdg("dBV")
     LevelB = AP.S2CDsp.Analyzer.ChBLevelRdg("dBV")
     AP.Anlr.RefChAdBr("dBV") = LevelA-AP.DGen.ChAAmpl("dBFS")
     AP.Anlr.RefChBdBr("dBV") = LevelB-AP.DGen.ChBAmpl("dBFS")
```

Figure 57. Procedure script to calibrate analog analyzer dBr reference to be equivalent to dB FS at the DAC output (extracted from "d-a gain.apb").

Setting Stimulus Levels in dB FS

Digital test signals for the measurement of DACs conventionally have levels specified in dB FS, which can be controlled directly by the digital signal generator. This is much simpler than for ADC testing where the analog stimulus level has to be matched to the device gain.

Gain

The gain of a DAC is normally quoted as the analog output level resulting from a digital input level of 0 dB FS. Practical devices may have non-linearities that mean that the gain at that level is not representative of the gain over most of the range, so gain is often measured at a digital output level below 0 dB FS, often –20 dB FS.

As an example, you may find that a level of –20.00 dB FS on the input to a DAC generates an output level of 3.68 dBV. The gain of the DAC, then, expressed as the output level corresponding to an input level of 0 dB FS, is:

$$0\,\text{dB FS} = 3.68 + 20.00 = 23.68\,\text{dBV}.$$

Unless otherwise specified, the gain is quoted at a frequency of 997 Hz.

The procedure “d-a gain.apb” measures the gain as described above and also sets the user-defined analog output reference levels, (dBr A, and dBr B) to correspond to dB FS based on this gain value.

An output of this procedure can be viewed in the APWIN Log File:

```
========================================
D-A converter gain
========================================
 Output level for at -20dB FS input
  Channel A: -14.023 dBV
  Channel B: -13.937 dBV
 Equivalent to gain of:
  Channel A: 5.977 dB(V/FS)
  Channel B: 6.063 dB(V/FS)
```

Analog levels expressed in dB FS

The analog output level of a DAC can be expressed with respect to the corresponding digital level in dB FS. This is convenient for many measurements and is required in AES17 for quoting some results. Test equipment often allows a dB scale with a user-set reference, such as dBr A and dBr B in APWIN. In many of the test procedures used here, these reference levels are configured to correspond with digital input levels using the DAC gain determined at the reference level of –20 dB FS and frequency of 997 Hz, as shown in the script extract in Figure 57.

This allows the use of the appropriate dBr (dBr A or dBr B) to be equivalent to dB FS for the analog measurements of that channel. This is ideal for those measurements that need to be quoted with respect to full scale, such as idle channel noise and signal-to-noise ratio.

Gain stability

The gain of a DAC may drift due to instability in the converter reference voltage or the value of other components. This variation can be monitored over time to determine the gain stability.

The output level stability test defined in AES17 is a measurement of the variation in the DAC output level with a –6 dB FS input, over a period of at least an hour. The DAC is first given a brief (typically 5 minute) warm-up. The APWIN procedure “d-a output gain stability.apb” illustrates the various settings that are required to perform this test accurately. The key parameters are set near the top of the procedure:

```
'Test conditions
 TestLevel = -6                    ' dBFS
 TestFreq = 997                    ' Hz
 DeviceWarmUpInterval = 0.1        ' (minutes)
 StabilityTestDuration = 1 ' minutes (>= 60 for AES17)
```

A typical output of this procedure is shown below. This can be viewed in the APWIN Log File.

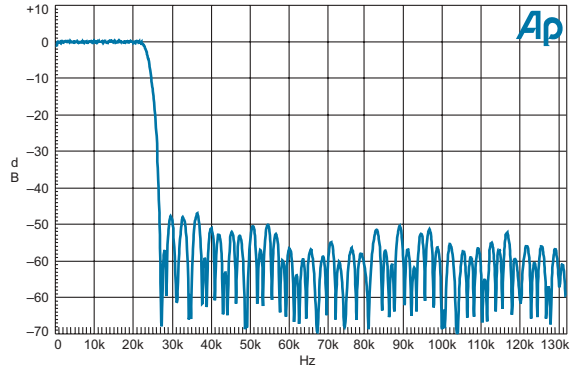
```
 ========================================
 Output-level stability (AES17-1998 cl 6.5)
 ========================================
  Level variation over 1.0 minutes
   Channel A 0.0012 dB
   Channel B 0.0012 dB
```

Gain-frequency response

Digital audio signals can only represent a selected bandwidth. When constructing an analog signal from a digital audio data stream, a direct conversion of sample data values to analog voltages will produce images of the audio band spectrum at multiples of the sampling frequency. Normally, these images are removed by an anti-imaging filter. This filter has a stopband that starts at half of the sampling frequency—the folding frequency.

Modern audio DACs usually have this anti-imaging filter implemented with a combination of two filters: a sharp cut-off digital finite impulse response (FIR) filter, followed by a relatively simple low-order analog filter. The digital filter is operating on an oversampled version of the input signal, and the analog filter is required to attenuate signals that are close to the oversampling frequency.

Figure 58. Anti-image filter.



This figure shows an anti-image filter frequency response for one DAC operating at a sampling frequency of 48 kHz. The response is "normalized" to 1 kHz (in practice this may be 997 Hz for reasons discussed in the chapter **Analog-to-Digital Converter Measurements** beginning on page 37). This means that the y-axis is adjusted for the response to read 0 dB at 1 kHz. The passband shows little variation up to a edge where the gain falls rapidly into the stopband. The region between the passband and stopband, in this case from 22 kHz to 26 kHz, is the filter transition region.

The key parameters of the transition region and stopband are:

- Minimum stopband attenuation.
  This is given by the height of the highest lobe in the stopband. For the device of Figure 58 the highest lobe shown is at a frequency of 36 kHz, and the lobe peaks to a minimum attenuation of 47 dB.
- Stopband lower edge frequency.
  This is often specified by the manufacturer and defines the range over which the minimum stopband attenuation applies. (If it is not specified then it can be defined as the lowest frequency where the attenuation is equivalent to the minimum stopband attenuation. In this case that frequency is approximately 26.5 kHz, on the falling curve in the transition region.)
- Attenuation at the folding frequency.
  For this device, it is 6 dB at 24 kHz.

The key parameters of the passband are better illustrated in Figures 59 and 60, which focus on this area of the response:

- Passband upper edge frequency.
  If this is specified, then it is used to define the applicable passband range for measurement of passband deviation. Alternatively, it may be defined as the highest frequency where the attenuation is within the specified

Figure 59. Anti-image filter passband, linear plot.

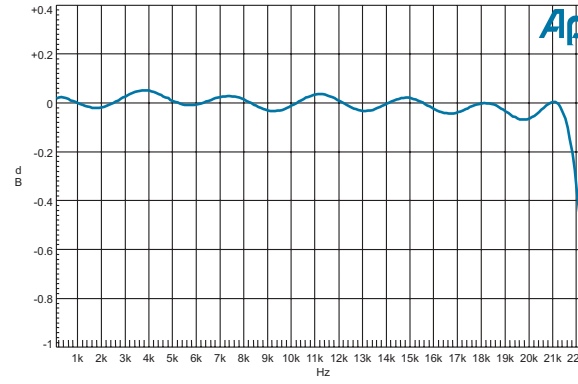
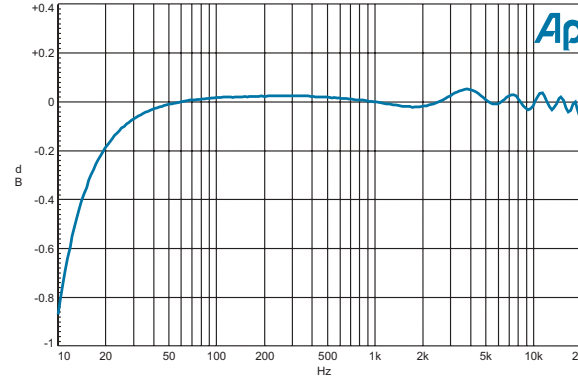


Figure 60. Anti-image filter passband, logarithmic plot.



passband deviation. In this case, a figure of 21.5 kHz could be quoted for a deviation of 0.08 dB.

- Passband deviation.
  This is the maximum deviation of gain over the passband when compared with the gain at 1 kHz (the graph in Figure 59 is normalized to 1 kHz). This deviation normally has more than one component. In the case of Figure 59 there is a regular sinusoidal ripple (caused by the high-order digital FIR filter) superimposed on more gradual gain changes (caused by low-order effects) which slope from the peak at 3.8 kHz. Over the range of 100 Hz to 22 kHz the maximum deviations from the 1 kHz gain are +0.07 dB at 3.8 kHz and –0.08 dB at 19.8 kHz.

- Passband ripple amplitude.
  If the overall gain slopes are ignored, then the gain fluctuation due to the sinusoidal ripple component alone appears to be about 0.8 dB peak-to-peak and accounts for about half the total passband deviation.

- Passband ripple periodicity.
  This is the separation of the cycles in the passband ripple. Figure 59

shows a passband ripple that has a periodicity of 3.6 kHz. The passband ripple periodicity can be related to the time dispersion of signals in the passband of the filter.[4]

The logarithmic frequency plot, shown in Figure 60, is more useful for recognizing the low-frequency roll-off due to DC blocking filters. (There could also be other components, such as transformers, that could be responsible for this.) Figure 60 shows that the 20 Hz response is about 0.2 dB down (from the 1 kHz reference).

Measuring Stopband Response Using a Wide-Band Signal and an FFT

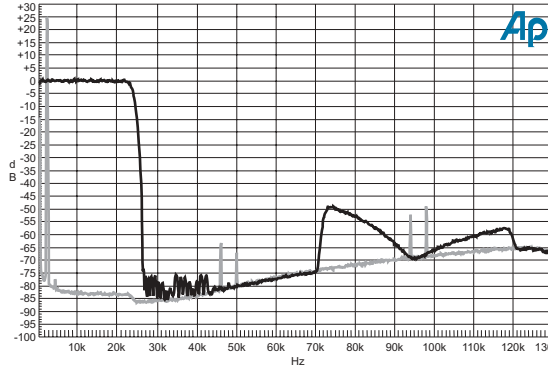
The stopband normally starts at or above the folding frequency ($F_S/2$), so this response cannot be measured by direct stimulation with a test tone at that frequency. It is necessary to stimulate the DAC with a signal below the folding frequency and to measure the amplitude of the "images." One convenient way of doing this is to use a spectrally flat pseudo-random signal as the stimulus and measure the spectrum of the reconstructed output with an FFT.

The APWIN procedure "d-a stopband fft.apb" performs this operation and was used for the first anti-image filter response in Figure 58.

This measurement technique shows the passband and the transition region as well as the stopband. The noise-like variation of the pseudo-random signal results in fluctuations in the results. With a moderate amount of averaging these fluctuations can be reduced to insignificance in respect to the stopband and transition region performance. However, even with a large amount of averaging, the fluctuations are still significant compared with the typical deviations being measured in the passband.

Figures 61 through 64 are some examples of more stopband measurements using this technique.

Figure 61. Stopband FFT, DAC "A."



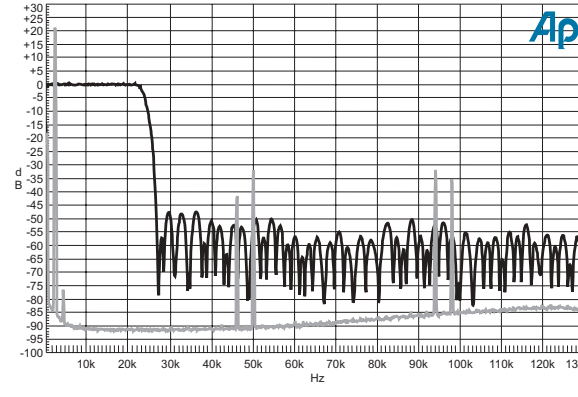
In all the four stopband FFT plots the test conditions are similar: The black trace shows the output spectrum when stimulated by the white pseudo-random MLS sequence at F_S = 48 kHz. The gray trace is a reference trace which is of the DAC stimulated with a tone that has the same peak amplitude. This reference trace is plotted in order to show the measurement noise floor so that it can be distinguished from the images of the MLS stimulus. The black trace is normalized for 0 dB at 1 kHz. The gray trace is scaled by the same amount.

In Figure 61 for DAC "A," at some frequencies the black and gray traces are at the same level. At those frequencies noise dominates so we can only observe that the attenuation must exceed this level. (The gray trace shows attenuated spectral images of the 2 kHz tone and we shall see later how with a tone stimulus we can explore the points of the frequency response much more slowly but with a greater sensitivity.)

This plot indicates that the minimum stopband attenuation over the band to 130 kHz is 49 dB. Defining the stopband lower edge frequency by the lowest frequency with that attenuation (a definition for convenience) we get a figure of 26 kHz. The attenuation at the folding frequency (24 kHz) is 6 dB.

The manufacturer of this part quotes a minimum stopband attenuation of 72 dB. This appears to be true for the attenuation of the spectral images either side of F_S (48 ±24 kHz) but this plot shows that this is not true of the images either side of 2 · F_S (96 ±24 kHz). The response at the 2 · F_S images is indicative of a zero-order hold function operating on 96 kHz data (rather than a more complex FIR filter that may be expected). Perhaps the manufacturer had forgotten about this characteristic when producing the specification?

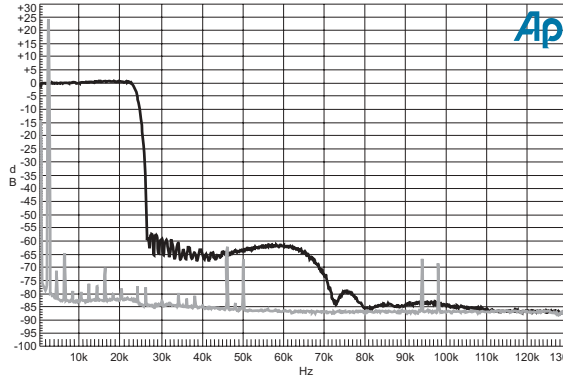
Figure 62. Stopband FFT, DAC "B."



The DAC "B" stopband attenuation shown in Figure 62 produces image components that are clearly much higher—at all frequencies—than the noise floor of the measurement. This indicates (with confidence) a minimum stopband attenuation of 48 dB for frequencies up to 130 kHz.

The passband lower edge is at 26.5 kHz and the attenuation at the folding frequency is 7 dB.

Figure 63. Stopband FFT, DAC "C."



The stopband of DAC "C" in Figure 63 shows an attenuation characteristic that increases with frequency. In the case of this shape of response the minimum stopband attenuation depends on the choice of value for the stopband lower edge. (As the attenuation tends to increase as the frequency rises the convenient definition we used for DAC "A" does not work as well.)

It is possible to choose the lowest frequency where the attenuation is typical. In this case I am quoting the specification with the start of the stopband to be on the part of the steep slope before it reaches a local minimum and rises again due to the stopband ripple.

Using that definition, the stopband lower edge is 26.3 kHz and minimum attenuation 57 dB.

For DAC "C" the attenuation at the folding frequency is 6 dB.

The graph of the stopband response of DAC "D" in Figure 64 illustrates the disadvantage of this technique for plotting stopband attenuation. The stopband attenuation is completely below the noise floor and so while we can observe that the attenuation must be greater than 82 dB, we cannot measure it.

In the absence of a minimum stopband attenuation figure the stopband lower edge is also not defined. However, we can tell that it is no lower than 26 kHz, and, as the slope of the response in the transition region is so steep, it can be estimated to be very close to that figure.

The attenuation at the folding frequency is 6 dB.

A more sensitive but much more laborious technique for measuring stopband attenuation is to use a sine wave stimulus and sweep this over the frequency band. At each stimulus frequency, the amplitude of each of the images

can be observed using an FFT. The procedure “d-a stopband sweep.apb” does this for a small range of stimulus frequencies to illustrate the method.

Figure 64. Stopband FFT, DAC “D.”

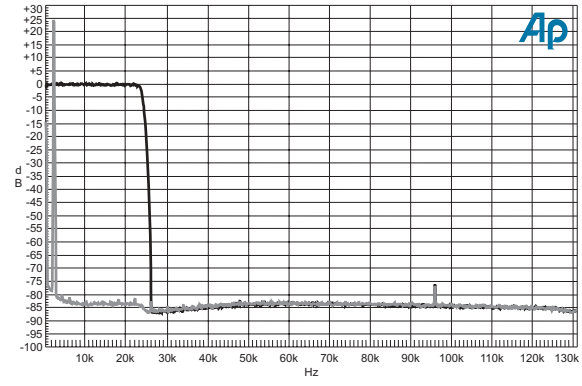


Figure 64 shows the output of that procedure for DAC “D.” There are seven FFT spectra laid over each other. The FFTs are taken with test frequencies from 18.2 kHz to 22.5 kHz. This band of test frequencies was chosen for this example so that the harmonic distortion products do not overlap with the images to make it even more confusing!

The spikes in the ranges from 36 kHz to 45 kHz and from 54 kHz to 68 kHz are second and third harmonics of the test frequencies, and do not tell us anything about the filter (except that the harmonic distortion is probably occurring after the main filtering action has occurred). The spikes from 25.5 kHz to 30 kHz are the lowest set of spectral images of the stimulus tones, and are at a frequency corresponding to the difference between the input and sampling frequencies. These images are near to the lower stopband edge frequency that we observed in Figure 64. The results appear to indicate a minimum attenuation of around 107 dB. (Measurements at more stimulation frequencies would be required to confirm that this result is typical of the whole band.)

The next-higher set of spectral images is at the sum of the input and sampling frequencies. These would fall in the range 66.2 kHz to 70.5 kHz. They appear to be very close to the noise floor at below –113 dB.

The difference between the stopband attenuation of DAC “D,” at greater than 100 dB, and the attenuation of the first two DACs, at less than 50 dB, is interesting.

Measuring Passband Deviation

This measurement is similar to that for ADCs. Every frequency in the passband can be uniquely stimulated, and that means that direct techniques

such as a sine wave sweep can be used. (More sophisticated methods such as multi-tone could also be used but will not be described here.)

In the procedure "d-a passband.apb" the Analog Analyzer level meters are used to measure the level on the output of the DAC being tested. The Digital Generator is set for an output of –20 dB FS and the frequency swept over the complete range supported by the generator (10 Hz to 0.47 · F_S). During the sweep the results are averaged to improve the accuracy of the results.

As the passband ripple is an indicator of time dispersion, the test has particular significance. Since passband ripple levels from modern DACs are small the test has to be as precise as possible. That is why the APWIN Digital Analyzer measurement mode (which is capable of faster measurements using the ADC with DSP measurement techniques) is not used.

Use of the Analog Analyzer avoids confusion between the ripple of the DAC being measured and the ripple in the test equipment ADC filter. However, this procedure could be adapted to use the Digital Analyzer as long as a suitable equalizing correction was incorporated into the test.

Figure 65 is a graph showing the passband frequency response of DAC "B" with a logarithmic frequency scale. This shows that within the range of the digital sine generator the frequency response of the DAC deviates by less than 0.8 dB at the low-frequency limit of 10 Hz, and about 1.1 dB at the high-frequency limit of 0.47 · F_S. Over the conventional audio frequency range from 20 Hz to 20 kHz the deviation is determined by the low-frequency attenuation of 0.2 dB.

Figure 65. DAC "B" passband, logarithmic frequency plot.

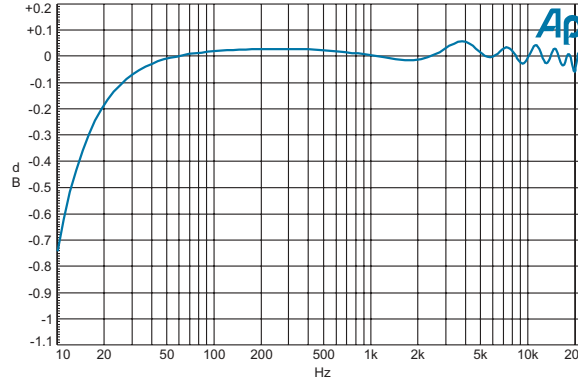


Figure 66 is a graph showing the passband frequency response of DAC "B" with a linear frequency scale.

The passband upper edge frequency for the digital filter used by the DAC is specified as 0.448 · F_S, or 21.5 kHz at this sampling frequency. This plot

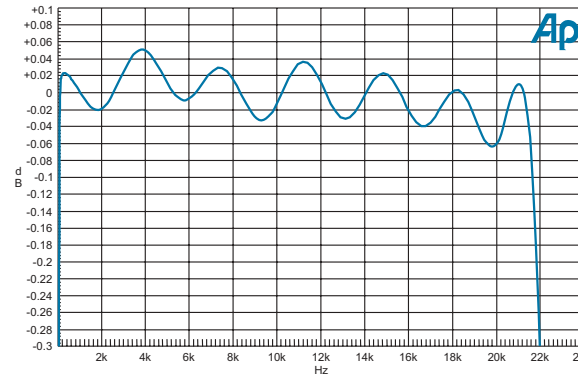


Figure 66. DAC "B" passband, linear frequency plot.

shows that above that frequency the response falls rapidly beyond the range of the gain fluctuations lower in the passband.

Passband Ripple and Dispersion

On the linear frequency scale the passband has an obvious sinusoidal component. It has a periodicity of about 3.5 kHz with six cycles from 300 Hz to 21 kHz. In the time domain, this corresponds to time-dispersion components preceding and following the main signal. The time separation of these pre- and post-echoes from the main signal is simply the reciprocal of the periodicity, or 290 µs. (See AES Preprint 4764 for more information on this.)[7]

The amplitude of these echoes can be estimated from the peak-to-peak amplitude of the ripple component—bearing in mind there can be other patterns in the gain variation. This particular component appears to have an amplitude of about 0.05 dB after allowing for slower variations.

$$\textit{Echo amplitude} = 20 \cdot \log\left(10^{\frac{\textit{PeakToPeakRipple}}{80}} - 1\right)\text{dB}$$
$$= -57\,\text{dB}.$$

This indicates that the primary passband dispersion of the filter is producing echoes 56 dB below the main signal and separated by 290 µs before and after.

Output amplitude for full scale input

This measurement is quite simple for a DAC. It requires the output level to be measured when a 997 Hz 0 dB FS tone is applied. If the device has gain settings then, according to AES17 [3], they should be at their normal positions.

One would normally expect that this level would be very close to the level implied by the gain measurement taken at –20 dB FS. Any significant difference would indicate high-level non-linearity.

Maximum Output Amplitude

The maximum output amplitude is a measurement that applies when the DAC has gain controls. It is intended to determine the level at which clipping starts to affect the output signal when the gain setting is adjusted above normal.

```
  ================================================
  Output amplitude at full scale (AES17-1998 cl 6.3)
  and maximum output amplitude (AES17-1998 cl 6.4)
  ================================================
  For the current settings of the device under test:-
   The gain (measured at -20dB FS) is
    Channel A 7.083 dBV/FS
    Channel B 7.040 dBV/FS
  Output amplitude at full scale:
    Channel A 7.083 dBV, with THD+N of 0.0043% and compression of -
0.001 dB
    Channel B 7.040 dBV, with THD+N of 0.0042% and compression of -
0.001 dB

  Maximum output amplitude:
   For devices without controls that can affect the output level then
   the maximum output amplitude is equivalent to the output amplitude
   at full scale.

   If the device under test has controls that can alter the output
   level then the maximum output amplitude is determined by adjustment
   of the controls of the device under test to the maximum level.
```

Figure 67. Results of procedure "d-a output at full-scale.apb."

The procedure "d-a output at full-scale.apb" (similar to the procedure "a-d input for full-scale.apb" described beginning page 49) performs this measurement.

The output level is measured with a 0 dB FS input. The gain controls are adjusted to maximize the signal level that can be achieved without the onset of significant distortion. For this test "significant distortion" means with no more than 1% THD+N, or 0.3 dB compression.

It is unusual to find a DAC that suffers from this amount of THD+N or compression, so the procedure is designed with the presumption that the compression and distortion are not significant. It will measure the output amplitude at digital full scale input and also note the THD+N and compression in order to verify this presumption.

The procedure makes up to four measurements with a 997 Hz sine test signal.

- With the gain control set to maximum and an the input signal set to 0 dB FS, the output amplitude is measured. If the THD+N or compression are below the 1% and 0.3 dB targets of the following two measurements, then this is the maximum output amplitude.
- If the THD+N is greater than 1%, then the gain setting is adjusted until THD+N measures 1%. Then the output level is measured.
- The level at which the compression is 0.3 dB is determined. This is done by making two measurements, first applying the test signal at –20 dB FS with the DAC level set at maximum and measuring the output level as *A*. Then the applied test signal level is set to 0 dB FS and the output level is measured as *B*. The compression is given by $(A + 20\,\mathrm{dB}) - B$. If the compression is greater then 0.3 dB, then the gain setting is reduced until the maximum level where the compression is not greater than 0.3 dB is found.

This test was run on a consumer DAT recorder in D-to-A monitor mode with the results shown in Figure 67.

In this test the gain of the device is first measured at –20 dB FS and noted in a form, dB V/FS, that corresponds to the output level for 0 dB FS input if there were no compression. The actual amplitudes at digital full scale match these to within the accuracy of the measurement, so this DAC exhibits no measurable compression. (The 0.001 dB value is less than the measurement error tolerance.)

Maximum Signal Level versus Sine Frequency

There may be mechanisms within the operation of a DAC that make the maximum output level vary with frequency. The procedure documented for the chapter on ADC testing (**Analog-to-Digital Converter Measurements**) could be adapted for this application.

On the other hand, as the full-scale input for a DAC is well-defined it is more appropriate, for many applications, to take measurements to confirm that compression and harmonic distortion do not affect the linear response at that signal level.

A method of doing this is to measure distortion and compression for a full scale input being swept in frequency.

Figures 69 and 70 illustrate the measurements used in the APWIN procedure “d-a full scale compression v freq.apb.” In this procedure the output level is plotted against frequency with a sine wave digital input level at –20 dB FS. These measurements are then scaled up by 20 dB and compared with the output level sweep measured with a full scale digital input signal. The trace

shown in Figure 69 is the output level sweep at 0 dB. This has signs of high-frequency roll-off above 15 kHz.

The plot of compression versus frequency in Figure 70—taken as the difference between the previous plot and the –20 dB FS reference plot—shows that if there is any frequency dependent compression it is below the normal variation of the measurement. The high-frequency roll-off is therefore not due to compression. It is part of the linear frequency response of the DAC.

Figure 68. THD+N versus frequency at 0 dB FS.

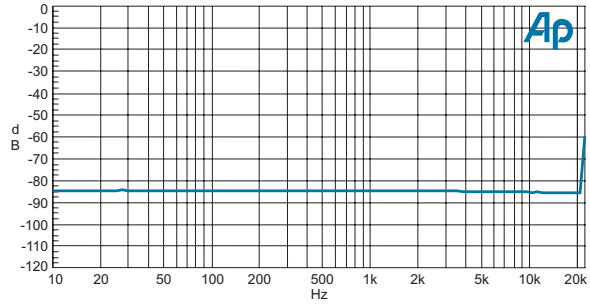


Figure 69. Output level versus frequency at 0 dB FS.

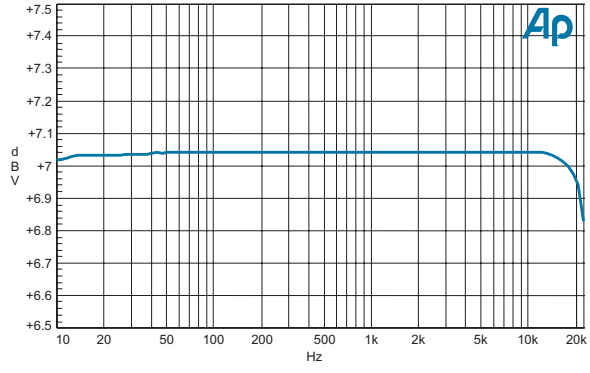
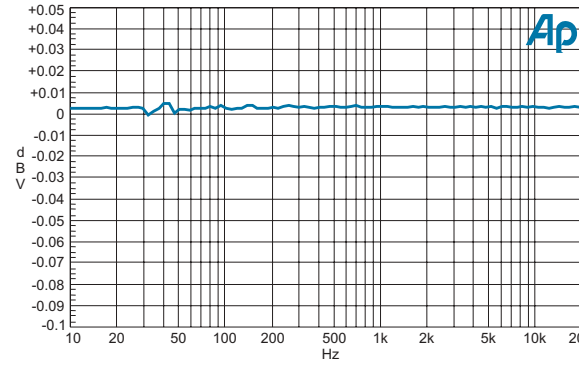


Figure 68 shows the output graph from the APWIN procedure “d-a full scale thd v freq.apb.” This procedure makes a sweep of THD+N against frequency for a sine wave at digital full scale.

The measurement is band-limited to 80 kHz, rather than the AES17 standard for THD+N of 20 kHz or less. This change has been made so that it remains sensitive to harmonics when the sine wave is above 10 kHz. Otherwise, for example, distortion due to clipping that started at 15 kHz and produced harmonic distortion products at 30 kHz, 45 kHz, 60 kHz and 75 kHz would not show up as a significant change to the line.

Figure 70. Compression level versus frequency at 0 dB FS.



The resulting plot in Figure 68 has a flat line at –85 dB over most of the band, and indicates that this device does not suffer from any artifacts that reduce the working maximum output level to below the level corresponding to full scale on the input.

Even at 22.5 kHz, where the measurement rises to about –60 dB (or 0.1%), the reading is still significantly below the 1% threshold. It should be noted that this reading is probably not due to a harmonic distortion product but from a sampling image that has been inadequately attenuated by the reconstruction filter in the DAC. The first image of a 22.5 kHz tone when sampled at the 48 kHz sampling frequency used here would appear at 48–22.5=25.5 kHz, which is below the starting frequency of the stopband for this DAC.

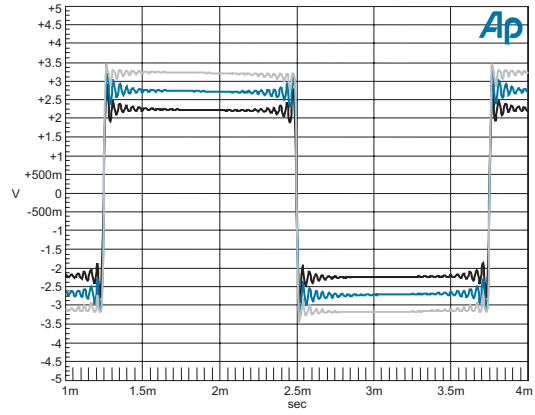
Digital Filter Overshoot and Headroom

The tests for maximum output amplitude and maximum signal level versus frequency that were described earlier use sine wave test signals. The digital anti-imaging filters within DACs will normally be designed so that they will not clip sine signals (and if they do it may only be by the tiny fraction of a dB due to the passband ripple). However, complex test signals can reveal problems of a much higher amplitude.

For example, a square wave will produce ringing that is a natural consequence of the removal of the higher harmonics by the filter. Figure 71 illustrates this for DAC “D.”

The output of a DAC is shown with three square wave traces at different amplitudes (and colors). The lowest amplitude trace (shown in black) has symmetrical ringing that overshoots by approximately 650 mV. However the largest amplitude square wave (in black) shows significantly lower overshoot. This is because the numerical representation of the correct overshoot value is outside the range of the digital filter. It has clipped in the digital domain. This is an imperfection—but not a serious one. A far worse behavior would be if the digital

Figure 71. D-A converter output clipping.



filters wrapped and the overshoot caused the sign of the signal to change—so a properly limited clip is a fairly good sign.

As can be seen in Figure 71, this particular converter shows some asymmetry in the clipping between the overshoot that proceeds the transition and the overshoot that trails it. The filtering function of this DAC is designed to be linear phase between the digital and analog domains, with phase and amplitude errors due to the analog filter compensated by the digital filter. However, the clipping here occurs at the output of the digital filter, so the phase compensation is not applied to the result of the clipping. The overshoots that follow the transition have a component due to the analog filter and, in the clipped condition, are not symmetrical with the overshoots preceding the transition, which have no component due to the analog filter.

Figure 72. DAC "D" output clipping.

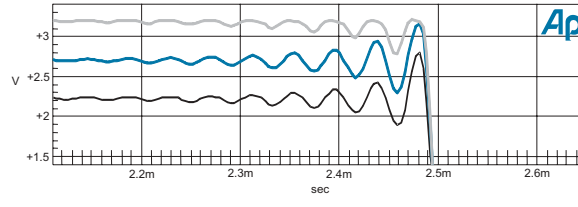
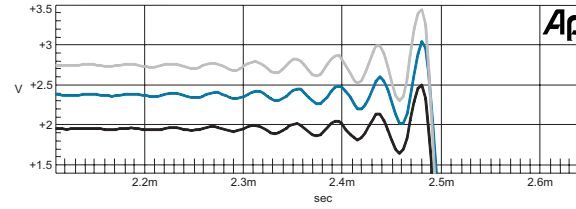


Figure 72 shows a magnified view of the ringing in advance of the square wave falling edge.

The highest amplitude square wave is the result of stimulating the DAC with a square wave peaking to digital full scale, so this helps us determine the analog level corresponding to digital full scale—about 3.2 V. This has excursions due to the filter ringing limited to the same level. One can therefore conclude that there is little or no headroom within DAC "D" beyond the range corresponding to the maximum input level.

Figure 73 . DAC "A" output clipping.



In Figure 73 the same plot for DAC "A" shows a different result. An overshoot of 0.7 V is showing no obvious signs of clipping. This indicates a headroom of at least:

$$20 \cdot \log \frac{3.4}{2.75} = 1.8\,\text{dB}.$$

It is possible to measure headroom beyond that shown by the clipping of a full scale square wave. Digital filters can be overdriven further by even more complex signals. The simplest method of generating an arbitrary near-worst case signal is to use a maximum length sequence (MLS). The MLS generator in System Two Cascade has an output consisting of samples of a constant amplitude and with a sign that varies pseudo-randomly according to the sequence. In that sequence some of the patterns of sample values will produce exceptionally high output values, and these high points can be used to probe the clipping behavior.

Figure 74. DAC "A" output with MLS at 50 % of full scale input.

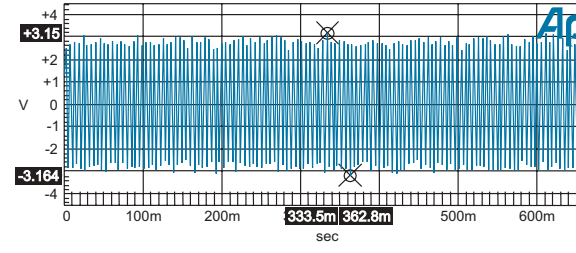
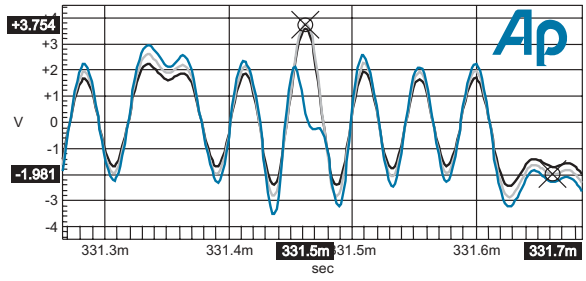


Figure 74 illustrates how the MLS can produce output peaks that far exceed the levels presented at the input. For DAC "A," digital full scale sine wave produces a peak output voltage of about 2.7 V, so an amplitude of 50% would correspond with 1.35 V. As the 50% MLS here is producing peaks of 3.1 V that represents an increase of 130%.

The values highlighted by the cursor can be used to probe the internal headroom of the converter. This is shown in Figure 75.

Figure 75. DAC "A" output with MLS at 80 % of full scale input.



This figure has expanded the trace in the region near the positive peak highlighted in Figure 74 and repeats the measurement for various generator amplitudes. The trace shows the results taken for amplitudes of 60%,70% and 80% of full scale (more than these were originally measured in order to find the amplitude where clipping occurs). The trace at 80% of full scale is showing signs of overload at the selected peak. The previous measurement showed that this peak is 130% above (2.3 times) the nominal input signal level. Therefore this indicates clipping at 2.3 · 0.8 = 1.84 times full scale, or +5.3 dB FS.

Another method for examining the overload characteristic of a DAC uses the Analog Analyzer peak meter to measure the peak level compression of the MLS at high levels. The procedure "d-a output clipping.apb" illustrates this technique, again with DAC "A." The results are tabulated below:

```
================================================
D-A filter MLS overshoot compression
================================================
Channel A
 Gain:  8.43 dBV/FS
 Peak overshoot:  7.37 dB
    I/P       I/P+Overshoot O/P-Gain Compression
 -7.00 dB FS  0.37 dB FS    0.45 dB FS    0.09 dB
 -6.00 dB FS  1.37 dB FS    1.44 dB FS    0.08 dB
 -5.00 dB FS  2.37 dB FS    2.10 dB FS   -0.24 dB
 -4.00 dB FS  3.37 dB FS    2.39 dB FS   -0.94 dB
 -3.00 dB FS  4.37 dB FS    2.56 dB FS   -1.82 dB
 -2.00 dB FS  5.37 dB FS    2.69 dB FS   -2.67 dB
 -1.00 dB FS  6.37 dB FS    2.78 dB FS   -3.58 dB
  0.00 dB FS  7.37 dB FS    2.82 dB FS   -4.53 dB
================================================
```

This indicates that peak level compression has reached 1 dB with a –4 dB FS input, with the peak output voltage at the analog level corresponding with approximately +2.4 dB FS, or 3.5 V.

The results of the same test with DAC "D" are:

```
================================================
D-A filter MLS overshoot compression
================================================
Channel A
 Gain:  9.49 dBV/FS
 Peak overshoot:  7.55 dB
    I/P       I/P+Overshoot O/P-Gain Compression
 -9.00 dB FS -1.45 dB FS  -1.35 dB FS   0.05 dB
 -8.00 dB FS -0.45 dB FS  -0.41 dB FS   0.05 dB
 -7.00 dB FS  0.55 dB FS   0.33 dB FS  -0.21 dB
 -6.00 dB FS  1.55 dB FS   0.61 dB FS  -0.93 dB
 -5.00 dB FS  2.55 dB FS   0.78 dB FS  -1.75 dB
 -4.00 dB FS  3.55 dB FS   0.97 dB FS  -2.56 dB
 -3.00 dB FS  4.55 dB FS   1.22 dB FS  -3.37 dB
 -2.00 dB FS  5.55 dB FS   1.46 dB FS  -4.16 dB
 -1.00 dB FS  6.55 dB FS   1.58 dB FS  -4.95 dB
  0.00 dB FS  7.55 dB FS   1.73 dB FS  -5.78 dB
================================================
```

This shows the compression reaches 1 dB with the input level –6 dB FS. The peak output level is then approximately +0.6 dB FS, or 3.2 V. This small amount of headroom confirms the result shown with the square wave earlier.

Is this measurement useful?

Most signals driving into a DAC are not likely to cause any overshoot, and so the amount of headroom beyond 0 dB FS is not relevant to the faithful reproduction of those signals. However, some signals may cause overloads in DACs. The MLS signal is not meant to be a representative signal. It is being used as a (nearly) worst-case signal in order to measure other effects. For example, if—in another device—the kind of signal inversion that occurs in the trace of DAC "A" were to occur only just above full scale (rather than at +5.3 dB FS) it may then produce audible artifacts in the presence of some high-level material.

Noise

The digital-to-analog conversion process will always have errors, and in an ideal system these errors should be inaudible. However, if they are audible, or

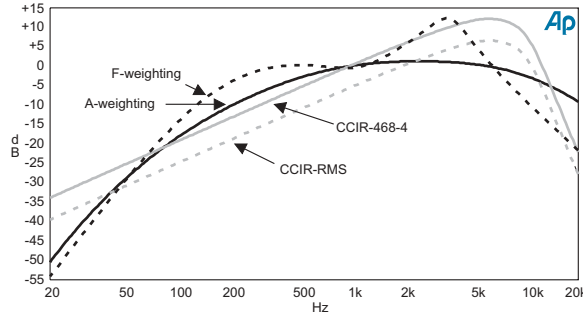
could become audible as a result of amplification, then the errors should be noise-like. This means that the error should have a spectrum that does not have spurious tonal components (from quantization distortion or idle tones) and that does not change with the signal (as caused by noise modulation).

Errors are more acceptable if they are only exhibited in the presence of high-level signals, as these signals have a masking effect that reduces their audibility. Conventionally, the noise of a DAC, like an ADC, is examined in the presence of a low-level signal. This stimulates the lower-amplitude coding levels in the converter, which, if any errors were present, would produce the most audible artifacts.

Noise Weighting Filters

Weighting filters, which attempt to mimic some of the characteristics of human hearing, are often used in noise measurements with the intention of making the measurement reflect the audibility of the noise. There are several weighting filters in common use, all of which emphasize the frequencies toward the middle of the band at the expense of those at lower and higher extremes.

Figure 76. Noise measurement weighting curves.



Note: Conventionally, weighting filters are normalized in their overall gain so that they have 0 dB gain at 1 kHz. The CCIR-RMS filter is based on the application of the CCIR-468-4 curve in AES17. This is normalized for 0 dB gain at 2 kHz. (This was originally proposed by Dolby for use with an average responding detector and called CCIR-ARM). The unusual normalization is used to make the results closer to those of an unweighted measurement. The original CCIR-468-4 weighting curve is designed for use with the CCIR-468 quasi-peak detector.

The effect of these weighting filters on a measurement of flat noise is illustrated in the following table:

20 kHz Band-Limited RMS Noise Measurements of a White Noise Source	
Unweighted	–0.07 dB
A-weighted	–2.33 dB
CCIR 468-4 weighted	7.01 dB
CCIR-RMS weighted	1.39 dB
F-weighted	2.46 dB

The noise floor of most converters is fairly flat, so these figures indicate the difference in results that might be quoted. The A-weighting gives the lowest noise figure and is normally the figure quoted on the front page of a data sheet. Where the noise is fairly flat you can add 2.3 dB to an A-weighted noise figure to estimate the unweighted noise over the DC to 20 kHz band.

Noise Measurement Using Quasi-Peak Metering

The CCIR-468 quasi-peak detector reads higher for noise sources than for sine waves of the same rms level. This is because noise sources have a higher crest factor, which is to say a higher peak amplitude, for a given rms level.

The following table illustrates the effect of the APWIN quasi-peak detector on the measurement of a properly dithered word-length reduction, using triangular probability distribution function (TPDF) dither. This is a typical dither used in digital audio systems and is representative of the noise source in many digital systems. The quasi-peak measurements are approximately 4.65 dB higher than the rms measurements. This adds to the effect of the CCIR 468-4 weighting filter to make a difference of 11.6 dB for noise with this property.

Quasi-Peak Measurements of TPDF Dithered Truncation	
Unweighted rms	–0.02 dB
Unweighted Q-peak	4.67 dB
CCIR-RMS weighted rms	1.36 dB
CCIR 468-4 weighted rms	6.99 dB (= CCIR-RM + 5.629 dB)
CCIR 468-4 weighted Q-peak	11.64 dB

Note that the APWIN “CCIR” weighting filter selection automatically switches between the standard CCIR 468-4 filter (normalized for 0 dB gain at 1 kHz) and the version normalized at 2 kHz, CCIR-RMS, which has 5.629 dB less gain. When Q-peak is selected as the detector then the standard CCIR-468-4 filter is used while for the rms detector the CCIR-RMS filter is used. In other words APWIN does not allow you to make a rms measurement directly using the standard CCIR-468-4 weighting intended for Quasi-peak measurements. The value in the table has been calculated by adding 5.629 dB to the CCIR-RMS reading.

Idle Channel Noise

The idle channel noise of a DAC is measured with the digital input driven by a signal representing zero. For two’s complement linear PCM (as used with AES3 and IEC60958) this is coded with all bits “zero.”

AES17 specifies measurement of idle channel noise with the CCIR-RMS weighting filter and a 20 kHz lowpass filter (a lower frequency than 20 kHz can be used if specified). Idle channel noise is measured in “d-a idle channel noise.apb” alongside the signal-to-noise ratio (more about that later). An unweighted result is also produced for comparison. Figure 77 shows the results gathered by the procedure when measuring DAC “A,” which has a 24-bit input.

```
================================================================
Idle channel and signal to noise ratio (AES17-1998 cl 9.1,9.3)
================================================================

 CCIR weighted RMS measurements
 Signal-to-noise ratio
  Channel A: -101.40 dB FS CCIR-RMS
  Channel B: -101.38 dB FS CCIR-RMS
 Idle channel noise
  Channel A: -101.50 dB FS CCIR-RMS
  Channel B: -101.70 dB FS CCIR-RMS

 Un-weighted measurements
 Un-weighted signal-to-noise ratio
  Channel A: -102.89 dB FS
  Channel B: -102.99 dB FS
 Un-weighted idle channel noise
  Channel A: -102.72 dB FS
  Channel B: -102.91 dB FS
```

Figure 77. Results from DAC “A” gathered by “d-a idle channel noise.apb.”

The idle channel noise measurement is not as useful for testing DAC performance as the signal-to-noise ratio measurement, discussed on page 108. The

signal-to-noise test measures noise in the presence of signal, while the idle channel noise test uses the digital zero signal which is not representative of normal operating conditions and, as a result, can produce misleading results.

For DAC architectures that use a multi-level conversion, the main noise mechanisms can be a result of level mismatches. For those types of devices a lack of modulation in the input data (such as for the idle channel test) will produce a much "better" measurement, one with an unrealistically low noise reading.

Perhaps because of this, the manufacturers of DACs with other architectures have sometimes incorporated circuits that modify the converter's operation in order to measure well for this test. These circuits may disable the conversion function when the number of samples of digital zero received exceeds a defined number. This is particularly true for delta-sigma converters, which are not sensitive to internal level mismatches but have other noise sources that do not vary as significantly with modulation. They can produce a higher noise reading for the idle channel measurement and could benefit, on paper at least, from such circuits.

```
============================================================
Idle channel and signal to noise ratio (AES17-1998 cl 9.1,9.3)
============================================================

 CCIR weighted RMS measurements
 Signal-to-noise ratio
  Channel A: -101.44 dB FS CCIR-RMS
  Channel B: -101.57 dB FS CCIR-RMS
 Idle channel noise
  Channel A: -108.85 dB FS CCIR-RMS
  Channel B: -109.08 dB FS CCIR-RMS

 Un-weighted measurements
 Un-weighted signal-to-noise ratio
  Channel A: -103.00 dB FS
  Channel B: -103.01 dB FS
 Un-weighted idle channel noise
  Channel A: -110.03 dB FS
  Channel B: -110.01 dB FS
```

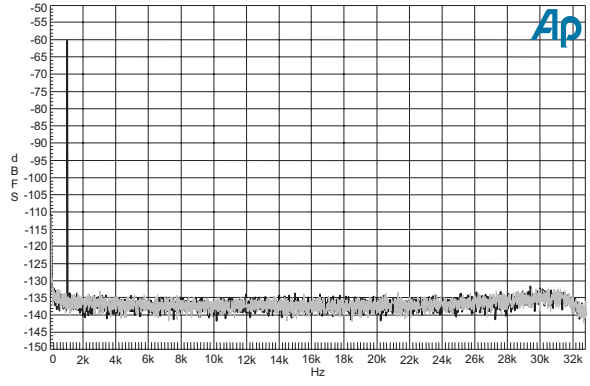
Figure 78. Results from DAC "A" with "cheat" switch on, gathered by "d-a idle channel noise.apb."

This behavior is not occurring in the measurements of DAC "A" shown in Figure 78, but this device can be switched into a mode that mutes the converter when presented with an input of digital zero. Note the improvement in the idle channel noise measurements in Figure 78. This is the same DAC—but with the "cheat" switch on. The idle channel noise is now over 7 dB better while the signal-to-noise measurement is no different.

Idle channel FFT spectra

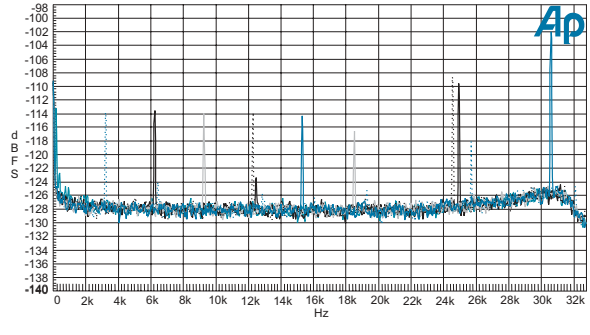
Some conversion architectures, such as delta-sigma devices, are prone to have an idling behavior that produces low-level tones. These "idle tones" can be modulated in frequency by the applied signal, which means that they are difficult to identify if a signal is present. An FFT of the idle channel test output can be used to find these tones. The procedure "d-a idle channel fft.apb" can be used to do this. This records the spectrum both with digital zero and with a properly dithered –60 dB FS tone. Figure 79 shows the result with DAC "A."

Figure 79. FFTs of idle channel noise and of –60 dB FS.



The two traces show a very similar noise floor, apart from a component near 11 kHz at a level of –132 dB FS in the gray trace. The black trace with the –60 dB FS signal does not show this component, which suggests that it could be an idle tone. As idle tones critically depend on the applied signal, a more thorough investigation would examine the spectra at various levels of DC. The procedure "d-a idle channel fft v level.apb" illustrates such an investigation, with the result shown in Figure 80. It is not as easy as simply running this pro-

Figure 80. FFT spectra for various DC levels.



cedure: the range of DC values for the sweep to produce this graph was selected after investigations with many more traces over a larger span of levels.

The DC values were selected to illustrate how a DC level swept over the range of 5.6% to 6.6% of full scale causes an idle tone at about –114 dB FS to sweep from 200 Hz to 15.3 kHz. There are also other components that appear to be at multiples of these frequencies. These vary in amplitude up to –102 dB FS at 30.6 kHz.

Signal-to-Noise Ratio and Dynamic Range

To avoid the shortcomings of the idle channel noise measurement, the conventional measurement of DAC noise is to measure the noise in the presence of a signal. The signal is a properly dithered 997 Hz tone at –60 dB FS which is then removed from the DAC output with a notch filter. The remaining signal is low-pass filtered to limit the bandwidth and the amplitude is measured in various ways and expressed in dB FS.

The signal-to-noise ratio defined in AES17 uses the CCIR-RMS weighting filter to measure the result. To identify this the result should be quoted as, for example: –91.76 dB FS CCIR-RMS.

This measurement is sometimes called *dynamic range,* which is the term used for a similar measurement defined in IEC61606. In that standard the result can be measured using an rms detector with A-weighting or with the CCIR 468 detector and CCIR 468-4 weighting.

Unweighted measurements are also often used, although that is not supported by any standard. (The unweighted signal-to-noise ratio of the DAT recorder was –84.82 dB FS.) Engineers may have reasons for making the signal-to-noise measurement without the low-pass filter as well.

The cocktail of signal-to-noise ratio results shown in Figure 81 were recorded using the macro “d-a signal to noise.apb” on the same DAC.

In summary, take care when using and quoting a signal-to-noise measurement: it is useless without knowledge of the bandwidth, weighting filter, or the detector that is used.

The notch filter to remove the 997 Hz tone can be the same as that used for a THD+N measurement (see **Harmonic Distortion**, page 115) but the result has to be quoted as an amplitude relative to full scale, rather than a ratio to the stimulus tone. If you have the reading as a ratio (as it is often quoted in data sheets), subtract the level of the tone to get the correct result.

```
================================================================
Signal to noise ratio (AES17-1998 cl 9.3)
================================================================
AES17 CCIR weighted RMS signal-to-noise ratio
  Channel A: -101.44 dB FS CCIR-RMS
  Channel B: -101.52 dB FS CCIR-RMS
IEC61606 CCIR-468 (ITU-R BS 468-4) signal-to-noise ratio
  Channel A: -91.19 dB FS CCIR Q-Peak
  Channel B: -91.33 dB FS CCIR Q-Peak
IEC61606 A-weighted RMS signal-to-noise ratio
  Channel A: -105.08 dB FS (A-weight RMS)
  Channel B: -105.25 dB FS (A-weight RMS)
Un-weighted RMS signal-to-noise ratio (20kHz band-limited)
  Channel A: -102.92 dB FS (Unweighted)
  Channel B: -102.95 dB FS (Unweighted)
```

Figure 81. Results of "d-a signal to noise.apb."

Noise on Digital Test Signals Due to Dither

The test signal for signal-to-noise measurement has to have a word length to match the capabilities of the DAC under test. This signal will therefore include noise from the dithered quantization to that word length.

This noise is spread over the band from DC to the folding frequency (one half the sampling frequency). For the recommended Triangular Probability Density Function (TPDF) dither of amplitude 2 LSBs, the rms level of the noise from the dither and the quantization can be determined using the following equation:

$$TPDF\ \&\ Q\ Noise = 10 \cdot \log\left(2^{(1-2 \cdot N)}\right) \text{dB FS}$$
$$= 3.01 - 6.02 \cdot N \text{ dB FS}$$

where N is the word length.

Applying this formula to a 16-bit converter will produce a figure for an unweighted signal-to-noise ratio, measuring the noise from DC to half the sampling frequency, of –93.32 dB FS.

The proportion of this noise that falls within a 20 kHz bandwidth will scale with the sampling frequency, F_S:

$$TPDF\ \&\ Q\ Noise\ (DC\ to\ 20\,kHz) = 10 \cdot \log\left[\frac{20\,\text{kHz}}{0.5 \cdot F_s}\right]$$
$$+ 3.01 - 6.02 \cdot N \text{ dB FS.}$$

The values produced by this equation for some common word-lengths and sampling frequencies are tabulated on the next page:

Unweighted 20 kHz noise floor of TPDF-dithered quantization			
Number of bits	F_S = 44.1 kHz	F_S = 48 kHz	F_S = 96 kHz
16	–93.73 dB FS	–94.10 dB FS	–97.11 dB FS
20	–117.81 dB FS	–118.18 dB FS	–121.19 dB FS
24	–141.89 dB FS	–142.26 dB FS	–145.27 dB FS

Equivalent "Number of Bits"

In the audio industry the discussion about the performance of a product often focuses on the "number of bits." There are multiple meanings being implied for this term; in addition to defining the word size used for storage or transmission of digital audio data, it is also assumed that it relates to the performance of the equipment.

Often the short-form description of a product mentions the "number of bits" rather than any other aspect of performance. When the term is used for a device that supports 24 bits but does not have the noise floor of a perfect 24-bit quantization it is sometimes said that the device is "not a *true* 24-bit converter."

A perfect DAC will not add any noise to the input signal. However, a digital input signal would need to be quantized (preferably with dither) to the input word-length of the DAC, and that process will have noise.

The noise level due to the dithered quantization alone can be seen as the target that a DAC needs to achieve to be a "true N-bit DAC." For example, if a 24-bit DAC had a signal-to-noise measurement of –118 dB FS (at a sample rate of 48 kHz) it might be said that it was "equivalent to 20 bits."

To illustrate how misleading this statement would be, just consider how the noise would increase if the converter were then fed with properly dithered 20-bit data. The –118 dB FS noise from the DAC and the TPDF dithered quantization noise from the input signal at –118 dB FS would then add together to produce a result about 3 dB worse. This is half the loss in noise performance that a one-bit reduction in word-length would produce.

Put another way, when fed 24-bit data the DAC has a "20-bit" performance, but if the user thought it was a 20-bit converter and fed it with 20-bit data the performance would then degrade to "19.5 bits."

DAC Intrinsic Noise

In some circumstances it may be useful to know how much noise is being added by a DAC. It is possible to evaluate this by subtracting the noise power

in the original signal from the noise power measured in the signal-to-noise measurement.

For example, the DAC in the 16-bit DAT recorder (DAC "D") has a signal-to-noise ratio of –93.6 dB FS (unweighted) measured in a 20 kHz bandwidth. This is close to the signal-to-noise ratio of the applied test signal at –94.10 dB FS. How do we work out how much noise is added by the DAC?

Uncorrelated noise has the property that the mean square level of the total noise is the sum of the mean square level of the noise components that are contributing to it. Therefore the relation is:

$$Output\_Noise^2 = Input\_Noise^2 + Added\_Noise^2.$$

Note: This relation applies if all the noise terms are referred back to the same point; in other words, the output noise and the device noise should be scaled to the level that they would have had at the input to produce the level of noise that is being measured at the output. In the case of DAC output level measurements in dB FS, the output levels can be referred to the corresponding digital input level, as the gain scaling is implicit to dB FS.

For noise levels in a decibel scale, the sum of squares relation is:

$$10^{\frac{OutputNoiseLevel\ (\mathrm{dB\ FS})}{10}} = 10^{\frac{InputNoiseLevel\ (\mathrm{dB\ FS})}{10}} + 10^{\frac{AddedNoiseLevel\ (\mathrm{dB\ FS})}{10}}.$$

Given a measured output noise level, and a known input noise level in the applied test signal, this equation can be used to determine the added noise level:

$$AddedNoiseLevel(\mathrm{dB\,FS}) = 10 \cdot \log\left(10^{\frac{OutputNoiseLevel\ (\mathrm{dBFS})}{10}} - 10^{\frac{InputNoiseLevel\ (\mathrm{dBFS})}{10}}\right).$$

The procedure "d-a subtracting test signal noise.apb" performs this calculation with the measured noise, and Figure 82 shows the results for the 16-bit DAC (DAC "C") in the DAT recorder. You can see that the intrinsic DAC noise is at about –102 dB FS. This is significantly less than the total output noise of –93.6 dB FS measured for the unweighted signal-to-noise measurement.

At the end of the procedure the unweighted idle channel noise (the DAC output noise with a digital zero input) is measured. This is effectively a measure of the DAC intrinsic noise, since the input is digital zero without any noise (from dither or truncation). The result can be directly compared with the DAC intrinsic noise calculated using the signal-to-noise ratio test signal, listed just

above it on Figure 82. The difference of 6.5 dB between the two intrinsic noise levels is an example of a noise modulation effect that might not be desired.

```
================================================================
Intrinsic DAC noise
================================================================
 Noise measurements (Un-weighted signal-to-noise ratio)
  Test signal noise on DAC input: -94.23 dB FS
  Channel A total DAC output noise: -93.60 dB FS
  Channel B total DAC output noise: -93.64 dB FS

 Calculated un-weighted intrinsic DAC noise in presence of signal
  Channel A intrinsic DAC noise: -102.32 dB FS
  Channel B intrinsic DAC noise: -102.57 dB FS

 Un-weighted intrinsic DAC noise with idle channel
  Channel A idle channel noise: -108.96 dB FS
  Channel B idle channel noise: -109.01 dB FS
```

Figure 82. Results of procedure "d-a subtracting test signal noise.apb," showing intrinsic DAC noise for a 16-bit DAT recorder (DAC "C") .

Noise spectrum

An FFT of the output of the portable DAT recorder (DAC "C") with the test signal used for the signal-to-noise ratio measurement is shown in Figure 83. This FFT was transformed from 16384 points and power averaged 16 times. The Blackman-Harris 4 term window was used.

The conversion factor to calculate the noise density scale from the discrete spectral line amplitude scale is, for this FFT:

$$\begin{aligned} &\textit{Noise density correction} \\ &= 10 \cdot \log\left(\frac{1}{\textit{WindowScaling}} \cdot \frac{\textit{FFTPoints}}{\textit{SamplingFrequency}}\right) \text{dB} \\ &= 10 \cdot \log\left(\frac{1}{2.004} \cdot \frac{16384}{262144}\right) \text{dB} \\ &= 15.06\,\text{dB}. \end{aligned}$$

The value *WindowScaling* is a property of the window that is related to the amount of spectral broadening produced by the window. There is more information on this in the section on the Fourier transform beginning on page 72 in the **Analog-to-Digital Converter Measurements** chapter.

The procedure "d-a noise floor FFT.apb" includes this correction when presenting the FFT of the noise floor in the presence of a signal used in the signal-to-noise ratio measurement.

The plot in Figure 83 has been made using a linear frequency scale with the same number of plotted points as FFT bins. This means that every FFT point is plotted. It is therefore possible to estimate the noise density by taking the mean level of the noise floor by eye.

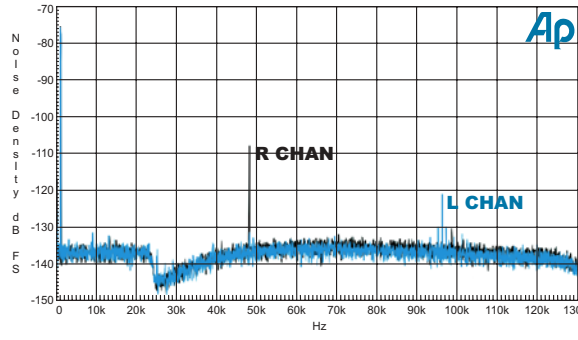


Figure 83. FFT of DAC "C" signal-to-noise test output, linear scale, Left and Right Channels.

Note: It is important to plot every point. The APWIN plotting routines would otherwise plot the highest valued bin for each frequency point where more than one bin was present and this would skew this visual estimate of bin mean level.

The mean level of the bins in the noise floor is at about –136 dB FS per hertz over the frequency range up to 20 kHz. This is related to a total noise level. We can calculate this noise over this bandwidth by integrating the noise density over that frequency range. In this case it is approximately flat over that range, so it is possible to calculate this by multiplying the mean density by the square root of the bandwidth. In logarithmic terms using dB FS this is expressed as:

$$\begin{aligned} Unweighted\ Noise &= MeanNoiseDensity + 10\log(Bandwidth) \\ &= -136 + 10 \cdot \log(20{,}000)\ \text{dB FS} \\ &= -136 + 43\ \text{dB FS} \\ &= -93\ \text{dB FS}. \end{aligned}$$

This result compares to within a decibel of the more accurate direct measurement of –93.6 dB FS for the unweighted signal-to-noise ratio made earlier.

Take care that this calculation can be made, because the FFT has been scaled by the –15 dB calculated earlier in order to represent the signal amplitude in terms of a spectral density per 1 Hz bandwidth. (More information on

Fourier transform scaling is found beginning on page 72)[6] For an FFT that has not been so scaled the calculation is:

$$\begin{aligned} &\textit{Unweighted Noise} \\ &= 10 \cdot \log\left(\frac{1}{\textit{WindowScaling}} \cdot \frac{\textit{FFTpoints}}{\textit{SamplingFrequency}}\right) \\ &+ 10 \cdot \log\left(\textit{Bandwidth}\right) + \textit{Mean amplitude per bin}\,(\text{dB FS}). \end{aligned}$$

The FFT is a high-bandwidth measurement using a 262 kHz sample rate with the 16-bit "HiBW" ADC in the Audio Precision System Two Cascade. This rate was chosen because it is useful in order to examine the ultrasonic noise floor.

Audio Precision "HiBW" and "HiRes" converters

The 16-bit ADC in the analyzer has a wider bandwidth but poorer dynamic range than the analyzer "HiRes" precision ADC. The lower dynamic range is not a problem because the low test signal level allows analog gain to be applied in front of the test equipment ADC, so that it can be driven with a signal that is up to 60 dB closer to full scale than for the DAC under test. The output of the analyzer ADC is then scaled down in the digital domain by the same ratio.

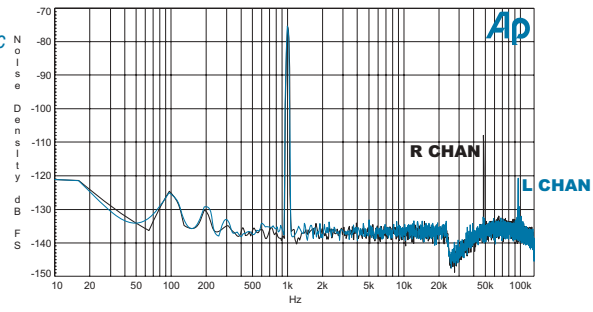
DAC "C" has a delta-sigma modulator. The rising noise density from 25 kHz is a characteristic of the noise shaping filter used in the modulator. At the modulator output this noise floor rises further but there is an analog lowpass filter after the modulator to reduce the amount of ultrasonic noise at the DAC output. In this case the noise floor appears to be well controlled so that the noise density does not rise significantly above the in-band noise level.

There are some spurious components shown on the plot in Figure 83. When looking at an FFT like this—one that has been scaled to show noise density—the height of the peaks due to discrete, or single frequency, spectral components does not correspond to the amplitude of that component. The correction used earlier needs to be subtracted. In this case 15 dB needs to be added to the noise density figure at the peak in order to estimate the amplitude of that component.

It is interesting that the two channels have different spectral components. The left channel has a –106 dB FS (–121 dB FS + 15 dB FS) component at 96 kHz, which is twice the sample rate, along with sidebands at 1 kHz and odd harmonics of 1 kHz offsets. The right channel has a fairly strong sample rate component at –93 dB FS (–108 dB FS + 15 dB FS). These components possibly indicate crosstalk from the respective clocks.

The low-frequency noise contribution is much clearer if shown with a logarithmic frequency axis. This is shown in Figure 84, which is a re-plot of the data in Figure 83 but to a log scale.

Figure 84. FFT of signal-to-noise test output, logarithmic scale.



In Figure 84, the lower frequency limit has been selected to be 10 Hz. The bin width for a 262.144 kHz, 16384 point FFT is 16 Hz. The amplitude at the first three points is due to the broadening of the DC bin by the window function and does not indicate low-frequency noise. (There is a longer discussion of the DC bin in the Analog-to-Digital Converter Measurements chapter.)[6] The effect of any non-DC components in the low-frequency noise spectrum is not apparent until about 64 Hz. and above. The measurement was made in England where the power line frequency is 50 Hz. The components at 100 Hz, 200 Hz, and 300 Hz are at even multiples of this power line rate and so are probably related to power supply ripple or some power line interference.

High-level non-linear behavior

Tests for the high-level non-linear behavior of a DAC are similar to those for non-linearities in analog electronics. Strictly speaking, the distortion and compression of signals at maximum level could come under the category of high-level non-linear behavior, but they are considered earlier in the discussion of the measurement of maximum levels.

Harmonic distortion and intermodulation distortion are the standardized tests for non-linearity measurement below full scale level.

Harmonic distortion

Deviation from non-linear behavior can be simply investigated using a pure tone. Any non-linearity in the transfer function of the DAC will result in frequency components in addition to the tone. Static non-linearities (those that depend only on the signal) will result in harmonic products at multiples of the original tone frequency. The most significant individual harmonics are normally at low multiples, such as the 2nd and 3rd harmonic at twice and three times the original (fundamental) frequency.

Conventionally, harmonic distortion is measured along with noise, and the measurement is called Total Harmonic Distortion and Noise (THD+N). This is most often measured with a test signal at 1000 Hz or 997 Hz and at a level of 0 dB FS or –1 dB FS, but it can be measured at various levels and frequencies of input tone. The measurement of THD+N on the output of a device is of the level of the residual left after the main tone is removed with a notch filter, and passed through a low-pass filter that limits the bandwidth to 20 kHz. The level of the residual is measured with an rms meter. When AES17:1998 is strictly adhered to, the result should then be quoted as a ratio between the level of the residual and the unfiltered signal level.

```
=======================================================
Total harmonic distortion and noise AES17:1998 cl 8.5
=======================================================

 THD+N at 997.00 Hz and -1.00 dB FS
  Channel A: -98.33 dB or -99.33 dB FS (Unweighted)
  Channel B: -98.92 dB or -99.92 dB FS (Unweighted)
```

Figure 85. Results of procedure "d-a thdandn.apb."

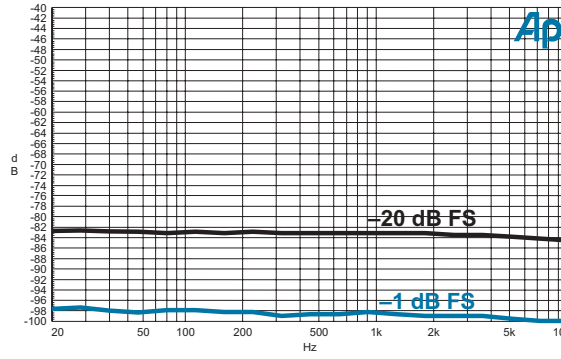
The procedure "d-a thdandn.apb" performs this measurement, displaying a reading for an input signal at a frequency of 997 Hz and a level of –1 dB FS. It also generates sweeps against frequency and against level, saving the plots as test files. Figure 85 shows the results of this procedure quoted as a ratio in dB and as an amplitude in dB FS for DAC "A," a 24-bit 48 kHz converter. Figures 86 and 87 are also produced by the same procedure.

Figure 86 is a graph (made according to AES17:1998 cl 8.5.1) of the THD+N ratio for a sweep of frequencies from 20 Hz to half the nominal upper band edge frequency of 20 kHz. The sweep does not extend beyond this point as any harmonic products would not be captured within the 20 kHz bandwidth of the filter being used. The reduction at the high-frequency end of the sweep is likely to be due to the reduction in the number of harmonic components that fall within the band as the test frequency rises. The 3rd harmonic will fall beyond 20 kHz for fundamental frequencies above 6.7 kHz, as will the 4th harmonic of frequencies above 5 kHz.

The upper trace on Figure 86 reads about 15 dB higher than the lower trace, yet it was made at a lower stimulus level, –20 dB FS. The lower trace was made at a higher stimulus level of –1 dB FS.

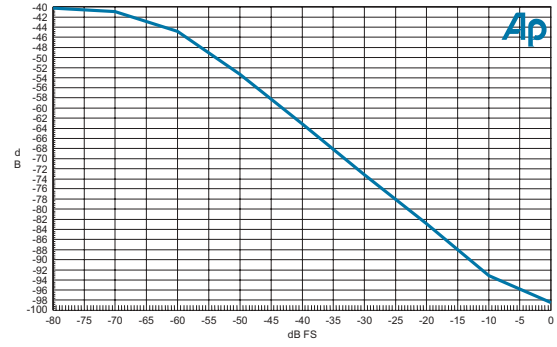
This display of a higher distortion reading for a lower stimulus level is not due to an increase in distortion but to the fact that the noise component of the measurement has not fallen with the applied signal. This is normal behavior, indicating that for all but the highest level the harmonic distortion products are

Figure 86. THD+N as a ratio, with total DAC output signal vs. frequency.



at a much lower level than the noise. The effect is shown more clearly in a trace of THD+N against test signal level, shown in Figure 87.

Figure 87. THD+N as a ratio, with total DAC output signal vs. applied signal level.



This trace was also generated by the same procedure. The main part of the trace between the coordinates (–50 dB FS,–53 dB) and (–10 dB FS,–93 dB) is close to being a straight line. This is the measurement of the constant noise floor at –103 dB FS as a proportion of the total signal amplitude as the test signal falls.

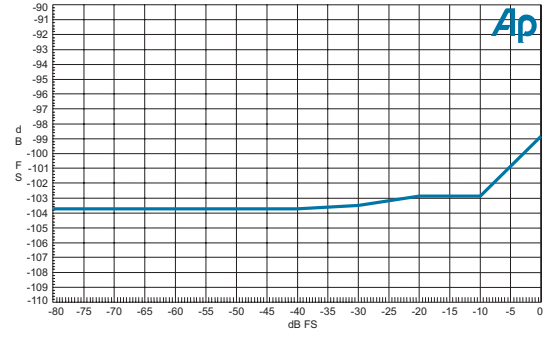
Below –50 dB FS the line deviates from this straight relationship. This deviation downward is not because the noise level measured within the THD+N measurement is falling for signal levels below –50 dB FS but because the contribution to the total signal level from wide-band noise has increased. Without the signal present the wide-band noise on the DAC output is at –63 dB FS, so a reading of the level of the DAC output signal with a tone at –60 dB FS will be the sum of that tone level and the wide-band noise. This can be calculated as:

$$\begin{aligned} \textit{DAC output level} &= 10 \cdot \log\left(10^{-63/10} + 10^{-60/10}\right) \text{dB FS} \\ &= -58.2 \text{dB FS}. \end{aligned}$$

The increase of 1.8 dB in the total level will cause a reduction by the same amount in the dB ratio (representing THD+N amplitude divided by total output amplitude) and will account for the deviation from linearity at input level (–60 dB FS,–45.8 dB). This error bottoms out at –40 dB, which represents the ratio of noise in the THD+N reading—which is band-limited to 20 kHz—to the wide-band noise (measured by the level meters prior to the notch filter) in the test equipment. (The Audio Precision S-AES17 filter option can be used to limit the level meter bandwidth to 20 kHz. If that filter is used then the ratio will "bottom out" at a much lower signal level.)

Another method of showing the effect being measured avoids this confusion. It is also more useful in that deviations in the THD+N level are more obvious. This is shown in Figure 88.

Figure 88. THD+N measured as amplitude vs. generator amplitude.



The measurement technique is the same but the Y-axis is plotted as a level, rather than a ratio. As the range of Y-axis values is reduced this amplitude scale can be more expanded than the ratio scale.

For most of the graph the THD+N amplitude measurement is flat at –103.7 dB FS. This indicates that noise components are independent of signal level over this range and that the harmonic components are insignificant. For X-axis values (test signal generator amplitudes) above –30 dB FS the THD+N amplitude starts to increase. This is a consequence of the high-level nonlinearities of the DAC being tested beginning to contribute harmonic distortion to the total reading.

It is useful to examine the FFT amplitude spectrum for specific input conditions. The traces in Figures 89, 90, and 91 were all made with the procedure "d-a THDN output fft.apb." They examine the spectrum of the same DAC output, applying a 997 Hz sine wave at –1 dB FS as a test signal. The THD+N under these conditions is –99.33 dB FS.

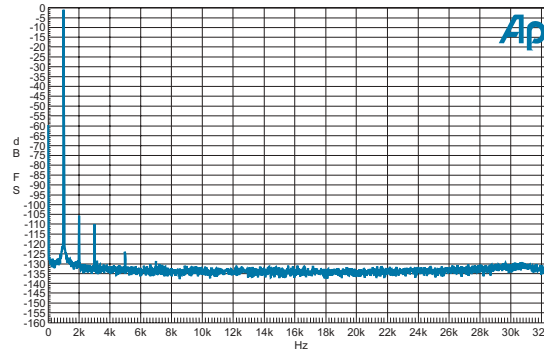


Figure 89. FFT of THD+N test output, linear scale.

Figure 89 shows two main distortion components at the 2nd and 3rd harmonic frequencies at –104 dB FS and –110 dB FS. There are also signs of 5th and 6th harmonics at –125 dB FS and –128 dB FS. These measurements represent a very high linearity in the DAC being tested, which is possible in even fairly inexpensive devices.

This measurement is also testing the linearity of the Audio Precision measurement ADC being used to digitize the DAC output for the FFT. In the case of System Two Cascade the distortion specification for the highest-performance ADC is –105 dB, which is more accurate than—but still comparable with—the device being tested, so there is some uncertainty about the results.

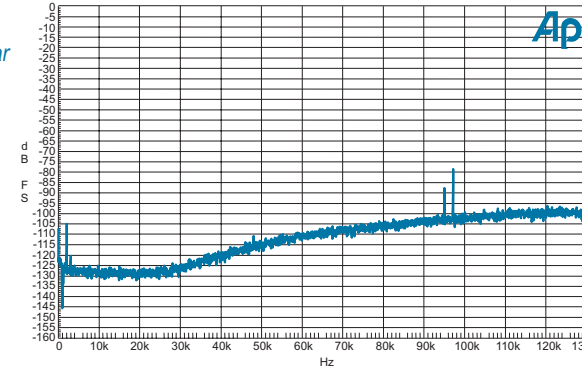
Another approach is to use the output of the analog notch filter as the input for the measurement ADC. This notch reduces the peak level of the remaining signal so significant gain can be applied in front of the ADC without clipping the signal. Test equipment can auto-range to take advantage of this directly.

The graphs in Figures 90 and 91 were produced by taking the signal on the output of the notch filter.

In the absence of the main tone component the Cascade's higher-bandwidth ADC can be used to give a picture of more of the frequency spectrum. This is now possible (even though the higher bandwidth measurement ADC has a poorer linearity than the device under test) because the removal of the high-level tone by the notch filter allows the residual signal to be presented at a much higher level to the ADC input. This is done so that errors due to ADC non-linearity will be at a much lower level with respect to the residual. Also, in the absence of the original tone, no harmonics will be produced by the measurement converter.

The higher bandwidth of this measurement reveals a rising noise floor and images of the test tone at a distance of 1 kHz on either side of 96 kHz (These are "images" of the test tone frequency.) The rising noise floor is a characteristic of the delta-sigma converter architecture and the images are indicative of a

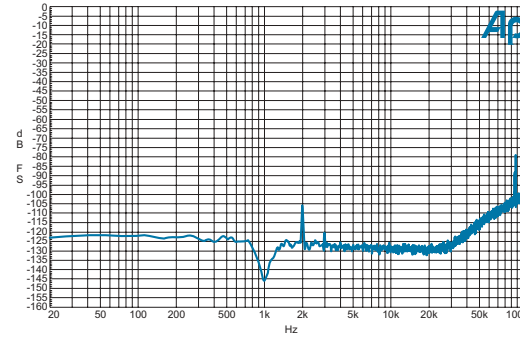
Figure 90. FFT of THD+N test notch filter output, linear scale.



weak anti-image filter rejection characteristic. (Refer to the stopband FFT of DAC "A" presented in Figure 61.)

The linear frequency axis of this plot shrinks the important audible region below 20 kHz to a small section of the graph. If you wish to show an expanded view of the audible range, a better presentation uses a logarithmic frequency scale, as shown in Figure 91.

Figure 91. FFT of THD+N test notch filter output, logarithmic scale.



This logarithmic graph clarifies identification of the harmonic components even while still showing the rising noise to over 100 kHz. The 2nd and 3rd harmonics are now at –105 dB FS and –120 dB FS. The 3rd harmonic has been significantly reduced (compared with Figure 89) showing that non-linearities in the measurement ADC were contributing to the earlier result.

> *Note: You may notice that the level of the underlying noise of the higher bandwidth plots is 6 dB higher than from the FFT taken by the lower bandwidth high-precision converter. This is not caused by the difference in converter precision but is due to the four-times-higher sample rate. Each FFT bin represents four times the bandwidth, and therefore has* $10 \cdot \log(4) = 6$ dB *more noise.*

This plot can be used to estimate the noise level as distinct from the discrete harmonic components. The number of points in the plot matches the number of points in the FFT output, so every FFT point has been plotted. This means that an estimate of noise density can be made. The window scaling factor for the equiripple window is 2.63191, and sample rate is 262144 Hz, so the factor for conversion to noise density is:

$$\begin{aligned} &\textit{Noise density correction} \\ &= 10 \cdot \log\left(\frac{1}{WindowScaling} \cdot \frac{FFTpoints}{SamplingFrequency}\right) \text{dB} \\ &= 10 \cdot \log\left(\frac{1}{2.63191} \cdot \frac{16384}{262144}\right) \\ &= 16.24 \text{ dB}. \end{aligned}$$

The noise floor appears at an average level of –128 dB FS within the 20 kHz band on the Y-axis. After applying the noise density correction this corresponds with a noise density of –144 dB F_S/Hz. Multiplied over a 20 kHz bandwidth (by adding $10 \cdot \log(20{,}000) = 43$ dB) this noise density translates to an unweighted noise level of –101 dB FS. This method of deducing noise level is not ideal, but it is often a useful calculation in the absence of other information. (Figure 88 indicated a noise floor bottoming out at –103.7 dB FS [unweighted], so this reading is a few decibels high.)

Intermodulation Distortion

Another conventional method of measuring non-linearity is to use two input tones and measure the discrete intermodulation products that are produced. This is a twin-tone intermodulation test.

For a pair of frequencies F_1 and F_2, the effect of non-linearities is to produce harmonic and intermodulation products at the following frequencies:

Order	Harmonic	Intermodulation Difference	Intermodulation Sum
2nd	$2F_1$, $2F_2$	F_1-F_2	F_1+F_2
3rd	$3F_1$, $3F_2$	F_1-2F_2, $2F_1-F_2$	F_1+2F_2, $2F_1+F_2$
4th	$4F_1$, $4F_2$	F_1-3F_2, $2F_1-2F_2$, $3F_1-F_2$	F_1+3F_2, $2F_1+2F_2$, $3F_1+F_2$

The advantage of this test for bandwidth-limited equipment such as a DAC, is that it is possible to arrange that the device is stimulated by tones at the higher in-band frequencies to stress the device. However, unlike harmonic distortion, some of the IMD products produced by non-linearities can be mea-

sured within the passband of the system; some of the difference frequency products are at lower frequencies than the stimulating tones.

There are several styles of twin-tone signals. The SMPTE RP120-183 and DIN45403 tests each use one high and one low frequency. The AES17 standard IMD test signal uses two high frequencies, one at the "upper band edge" frequency (normally 20 kHz), and another at 2 kHz below that frequency. (For most systems the upper band edge is defined in AES17 as 20 kHz, but it may be lower than this for systems with sample frequencies less than 44.1 kHz.)

The level of the twin-tone is specified for the AES17 test to peak at full scale. This is an rms level of –6.02 dB FS for each tone, with a total rms level of –3.01 dB FS.

20 kHz and 18 kHz input tones will produce the following intermodulation difference frequencies:

Order	Intermodulation Difference	Actual Frequencies if F_1 = 20 kHz & F_2 = 18 kHz
2nd	$F_1–F_2$	2 kHz
3rd	$F_1–_2F_2$, $2F_1–F_2$	16 kHz, 22 kHz
4th	$F_1–_3F_2$, $2F_1–_2F_2$, $3F_1–F_2$	34 kHz, 4 kHz, 42 kHz

The in-band products up to the 4th order are at 2 kHz, 4 kHz and 16 kHz. AES17 specifies that the measurement is of the ratio of the total output level to the rms sum of the 2nd- and 3rd-order difference frequency components on the output.

Figure 92. IMD test using FFT.

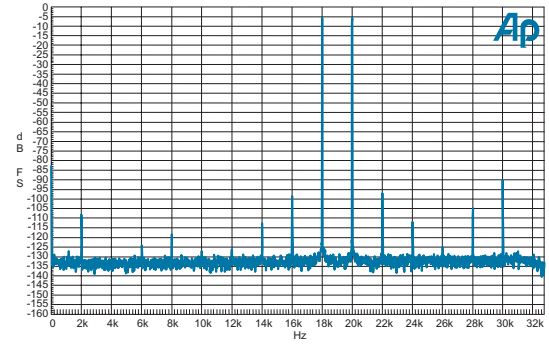


Figure 92 shows the spectrum of the DAC output when stimulated by the twin-tone. The Y-axis has been calibrated to the level of a discrete sine wave in dB FS. The procedure "d-a imd_fft.apb" produces the graph and calculates the IMD product amplitudes from the FFT by summing the spectral power den-

sity around each spectral component. The levels of each component and the IMD ratio are tabulated in Figure 93.

```
 ======================================================
 FFT of intermodulation distortion output
 ======================================================
-108.24 dB FS 2nd Order Difference product
-98.73 dB FS 3rd Order Difference product
-98.27 dB FS Sum of 2nd and lower 3rd order difference products
-3.07 dB FS total signal level
-95.20 dB IMD ratio
-107.41 dB analog analyzer DFD 2nd order component
```

Figure 93. Results from IMD DFD Measurement.

It is also possible to determine the level of each spectral component by examining the height of each relevant peak in the spectrum. That method can be used when you do not have access to the raw FFT data for calculation, as above. It is not as accurate, since the height of the peak depends on the exact relation between the frequency of the component being measured and the frequency corresponding to the FFT bin closest to it. Windows are available that reduce this sensitivity (such as the “flat-top” window) but the side-lobes of these windows are higher and may affect the accuracy of the measurements of the low-level frequency difference components.

The result from the System Two Cascade Analog Analyzer IMD difference frequency distortion (DFD) measurement is also shown in Figure 93. DFD is defined in IEC60268-2:1993 as the ratio of the amplitude of the 2nd order difference component to the arithmetic sum of the amplitude of the two components in the test signal. In this case the sum of the two original components is 0 dB FS, so the ratio used by the IEC60268 definition test has the same numerical value as the amplitude of the component in dB FS. (The reading using the analog meter is slightly higher, but at these low levels of distortion that difference may be a property of the measuring equipment.)

Low-level non-linear behavior

Noise or distortion that is present with low signal levels is in many ways more objectionable than the harmonic or intermodulation distortion resulting from high-level non-linearities.

Linear digital audio processes, including ADCs and DACs, should have an output signal that is linearly related to the input signal plus a random error term. The error term should be uncorrelated with the input signal.

However, when low-level signals are quantized there is a significant amount of non-random error. With conventional converters, the error is highly corre-

lated with the input signal. For delta-sigma converters, the error can have strong discrete frequency components at a frequency related to the instantaneous, or DC, level of the signal. The solution is to add dither.

Ideally, dither randomizes the quantization error so that it is has the character of white noise at a constant level. The ideal application of dither at all possible stages, however, is often not practical. Any compromises that must be made can be evaluated by the measurement of the amplitude of low-level distortion products.

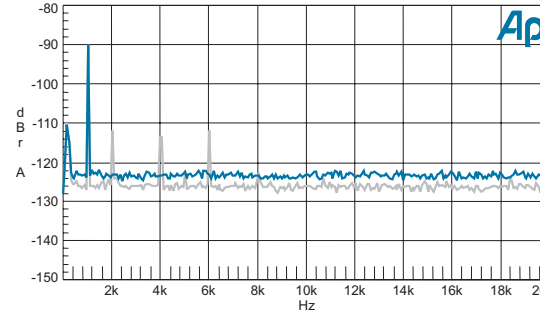
The procedure “d-a low level distortion fft.apb” illustrates how these products can be investigated. The method is similar to making a measurement of the noise floor spectrum described earlier (“d-a noise floor FFT.apb”), but as we are now interested in the amplitude of specific components rather than noise density, the following changes are made:

- The amplitude axis is calibrated for the height of discrete components (the APWIN default), rather than for noise density.
- Since we are not now interested in the noise, it is unnecessary to plot each FFT bin individually. To save time only 256 points are plotted, rather than the 8192 points (one for each FFT bin) plotted for the noise floor graph.
- The “flat-top” window is selected in order to optimize the accuracy of the amplitude measurement from the FFT plot.
- Only the frequency range to 20 kHz is of interest, so the high-resolution ADC is used at 65536 Hz sample rate.

The spectrum in Figure 94 is a measurement of the output of DAC “D.” The procedure applies a 997 Hz test signal at –90 dB FS. The lower trace shows the result when dithered for a 17-bit word length. DAC “D” truncates data below the 16th bit, so the dither in this case is inadequate.

The harmonics of the 997 Hz tone (at 1994, 3988, 5982 and 7976 Hz) have been produced by the low-level non-linearity introduced by the inadequate

Figure 94. FFT to examine low-level distortion.



level of the dither. These components are each at least 15 dB below the total unweighted noise floor, but, not being masked by the uncorrelated noise or by the low-level signal, they are audible.

In contrast, the upper trace was measured with the correct amplitude of dither (16-bit) and shows no harmonics. The amplitude of the noise floor is, of course, higher.

The sensitivity of this test can be increased by expanding the vertical scale, and increasing the number of averages.

Noise Modulation

It is possible for noise or dither to have decorrelated the truncation error from the input signal, but not to have decorrelated the truncation error *power* from the input signal. For a simple example, truncation error power might be minimized if the mean signal level is centered between quantization decision points, while it would be maximized when the signal is closer to a decision point. This is illustrated in Figure 118 in the annex on dither. The positive half of the waveform approaches the decision point between the 0 and 1 level (at 0.5 LSB), and the dither causes the quantizer to switch frequently between those two levels. This is in contrast with the negative half of the waveform, which is close to midway between decision points where the dither is much less likely to cause the output to change.

This correlation of truncation error power with signal is a form of noise modulation.

A simple test for this would be to measure the noise or noise spectrum for a low-level tone with various DC levels. The idle channel FFT spectra measurement discussed on page 107 would also reveal broad-band noise fluctuations.

When performing an FFT, if the variation in noise level is small it may be swamped by the statistical variation of the FFT noise floor. In this case, it is possible to use FFT power averaging to reduce the statistical variation.

A swept bandpass filter measurement may also be used. AES17 recommends a using 41 Hz stimulus at –40 dB FS, notching the stimulus out of the results, applying a series of one-third octave bandpass filters and measuring the noise in each band. The stimulus is then dropped by 10 dB and a new set of measurements is taken. This process is repeated until a family of measurements is completed.

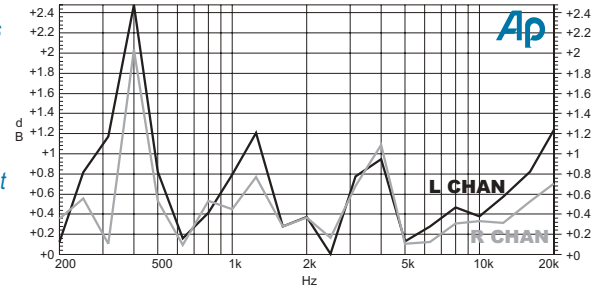


Figure 95. AES17 modulation noise. Black is the left channel, gray is the right. Each point plotted is the maximum variation in noise over a series of measurements at different stimulus levels.

Figure 95 shows the results of noise measurements acquired in this way, generated by the procedure "d-a noise modulation.apb." After the family of curves is generated as described above, the difference between the lowest and highest noise reading in each range is measured. This is plotted as the maximum noise variation for each one-third octave frequency band. The noise modulation as defined in AES17 is the greatest variation across the audible spectrum, which in this case is about 2.4 dB for the left channel and 1.9 dB for the right channel, both occurring at 400 Hz.

The one-third octave width is appropriate for this sort of measurement, since it scales in bandwidth with frequency. In this respect it is similar to the width of the auditory filter that detects noise.

Jitter Modulation

The theory of sampling jitter in a DAC is described in the *Jitter Theory* chapter.

Jitter is the error in the timing of a regular event, such as a clock. The *jitter transfer function* indicates the relation between the jitter of an external synchronization input and the jitter of a device. The *intrinsic jitter* of the device is that element of jitter that is independent of any external clock synchronization input.

The jitter of the clock that determines the DAC reconstruction timing is called *sampling jitter*. It is the only jitter that has any effect on digital-to-analog conversion (DAC) performance. Jitter in other clocks may, or may not, be indicative of the jitter on the sampling clock.

The direct connection of test probes to a sampling clock inside an DAC may be possible, but measurements using this technique are beyond the scope of this article. Measurements of the *effect* of this jitter on the audio signal are considered instead.

Interface Jitter Susceptibility

A DAC can be prone to jitter that is received on the digital interface. Some of this *interface jitter* can be transferred to the derived sampling clock; this depends on the degree of jitter attenuation in the clock recovery circuits prior to the sampling clock. This jitter attenuation characteristic will determine the jitter susceptibility of the DAC.

Sampling Jitter Transfer Function

The most useful method of measuring the jitter susceptibility is through the sampling jitter transfer function.

A procedure for measuring the jitter transfer function is supplied as "d-a JTF.apb." A 20 kHz tone at –1 dB FS is used as the audio stimulus. Jitter is applied to the interface signal carrying the tone. The jitter is in the form of a sine wave with a peak level of 0.125 UI (though this may need to be reduced if the DAC cannot maintain lock at that level of jitter). The jitter frequency is swept over a range defined by the following constants defined in the procedure:

```
Const N_frequencies = 10
Const StartFreq = 100
Const EndFreq = 39e3
```

The jitter will produce modulation sidebands above and below the audio signal frequency. At each jitter frequency, the amplitude of the lower (frequency) sideband is measured. (It is important to have good frequency resolution for this measurement as the sidebands for low jitter frequencies will be close to the 20 kHz tone.) The measurements are taken from an FFT using a high dynamic range window (specifically, Equiripple) and applying integration over nearby bins.

Figure 96. FFT for DAC "D" jitter transfer function measurement.

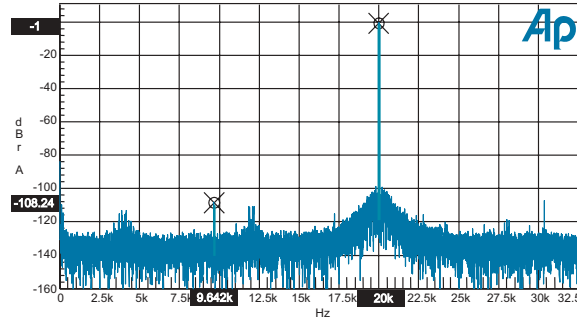


Figure 96 illustrates one of the FFT traces. The cursors are highlighting the main component at 20 kHz and the lower sideband at 9.642 kHz. The sideband amplitude is first calculated from theory, for amplitude of the applied jitter and the main tone frequency. The difference between this calculated level and the

actual measured level is then plotted as the jitter gain. (The interface jitter level used for this measurement was reduced to 0.05 UI, as the 0.125 UI setting defined in the test causes the device to temporarily lose lock. This reduced level of jitter stimulation means that the measurement is 8 dB less sensitive, since the sidebands' amplitudes are closer to the noise floor.)

Note that the "skirts" around the main 20 kHz component in Figure 96 are a result of the jitter generation mechanism and are not jitter intrinsic to the converter under test. These skirts disappear when the jitter generation is disabled.

Figure 97. DAC "D" jitter transfer function.

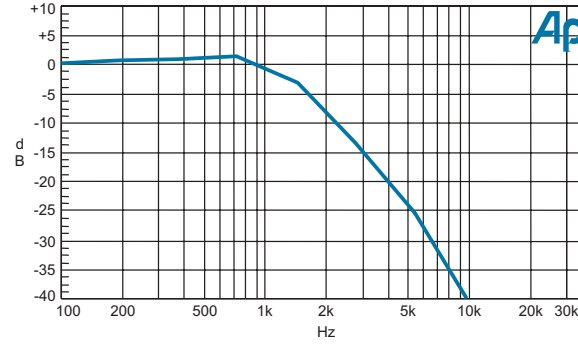
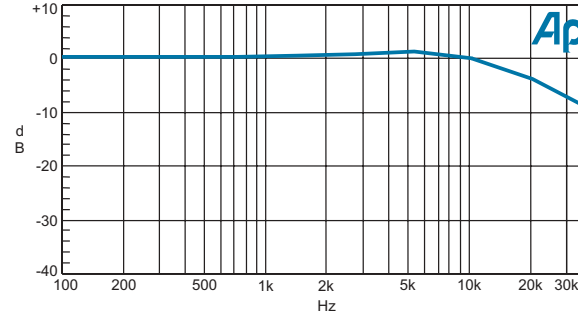


Figure 97 shows the total measured jitter transfer function using this procedure. This shows that there is approximately 1 dB of jitter peaking at around 700 Hz with jitter attenuation above 800 Hz. The slope above 2 kHz is about 40 dB per decade, indicating a second-order response. This device has significant audio-band jitter attenuation, but as this is only true for higher jitter frequencies it remains susceptible to lower-frequency jitter. (This compares with some converters which have 60 dB attenuation at 500 Hz in order to ensure that modulation sidebands cannot approach audibility.)

The plot in Figure 98 shows the measurement of another DAC. In this case there is no significant jitter attenuation in the audio band.

Figure 98. DAC "A" jitter transfer function.



The upper frequency limit for this measurement is set by the maximum sideband offset that can be achieved within the measurement band. Any analog band-limit filter after the converter may affect measurement bandwidth. In the case of a 20 kHz bandwidth DAC with an analog anti-image filter, the maximum frequency offset is just under 40 kHz with the stimulus tone at 20 kHz. The highest jitter frequency plotted by this procedure is 39 kHz. This produces a sideband at:

$$|20\ \text{kHz} - 39\ \text{kHz}| = 19\ \text{kHz}.$$

The lower jitter frequency limit for the measurement is set by the frequency resolution of the FFT. This procedure can use a 32768 point FFT with an equiripple window, which sets the lower jitter frequency measurement limit to about 15 Hz.

The lower jitter frequency limit selected for this particular measurement has been set at 100 Hz. As this does not require such high frequency resolution, this allows a reduction in the number of FFT points to 8192 with the benefit of increased processing speed.

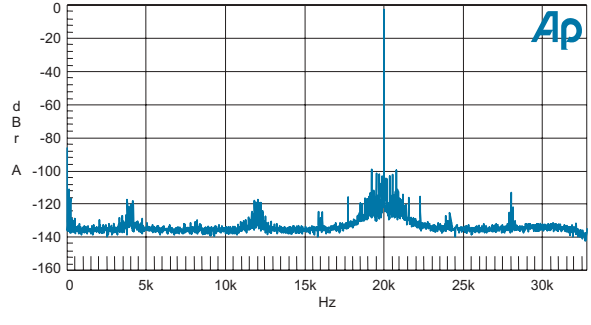
Intrinsic Jitter Artifacts

Intrinsic jitter will produce increased noise in the presence of signals which are both high-frequency and high-level. The procedure "d-a intrinsic jitter.apb" uses this characteristic to estimate the amount of jitter that would produce such a noise floor. As it does not determine the *source* of the noise it should be used with care, as the noise could be from other sources. However, it will provide an upper limit on the amount of intrinsic sampling jitter that could be present.

The same high-level tone is used as the stimulus as for the previous measurement of jitter transfer function. An FFT of the converter output is then computed and plotted with one bin per plotted point. Figure 99 is an FFT of the DAC "D" output that is used for the intrinsic jitter calculation later.

The high frequency is selected in order to maximize the sensitivity of the measurement to jitter. In some cases, it is more useful to use a lower frequency

Figure 99. FFT used for intrinsic jitter calculation.

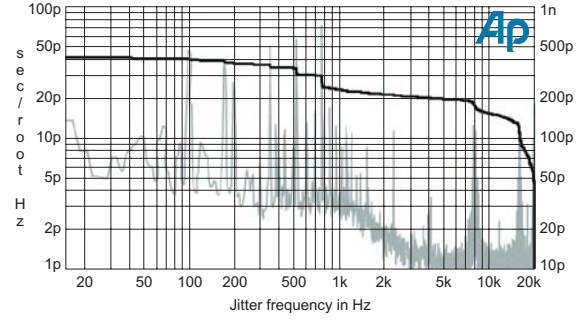


in the middle of the band (such as 10 kHz) to look for symmetry in the skirts, which is an indicator that modulation effects are being observed.

The procedure measures the amplitude of each bin between DC and the stimulus tone, and calculates the frequency and level of jitter that would be required to produce this bin amplitude through jitter modulation of the 20 kHz tone. This produces a plot of *potential* intrinsic jitter versus jitter frequency.

This is shown on Figure 100 as a jitter density, plotted as the lower line, in gray. This line is calibrated in rms seconds of jitter per root hertz on the left axis.

Figure 100. Calculated intrinsic jitter per root Hz, integrated to 20 kHz.



The integration of this jitter density to determine the total jitter is not a simple task to do from the graph. The integrated jitter curve is shown to simplify this task. This upper curve in black represents the total jitter measured from the frequency on the X axis to the right-hand limit of the graph. This shows, for example, that the total jitter above 1 kHz is just over 350 ps, and above 200 Hz it is about 225 ps.

The interpretation of the original FFT into sampling jitter should be treated carefully. However, as an indicator of the upper limit of possible sampling jitter spectral density, it is a very sensitive tool.

In this example an examination of the original FFT is useful in judging the reliability of the result. Components within 5 kHz of the main 20 kHz tone appear to be symmetrical and so are likely to be caused by modulation, such as jitter. On the other hand, the components around 4 kHz and 12 kHz are less likely to be jitter. These are offset by 8 kHz and 16 kHz from the main tone. The component 8 kHz above the main tone, at 28 kHz, does not have the same shape as that at 12 kHz and this lack of symmetry suggests that they are not due to jitter modulation. Another source for these components should be investigated.

Most of the jitter is in the region below 1 kHz. We know from an earlier measurement of this DAC (DAC “D”) that the sampling jitter transfer function of this part shows little or no attenuation below 1 kHz. It is therefore quite pos-

sible that the jitter being observed is sourced prior to the clock recovery circuit. It may be from the "data-jitter" on the interface signal or from interference within the unit. "Data-jitter" from the interface could be investigated using J-test (see the next section).

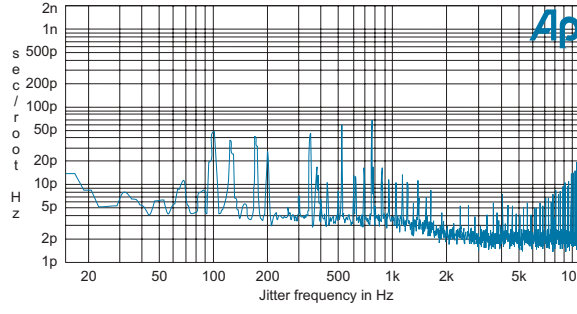
This analysis of this intrinsic jitter measurement could be compared with the similar analysis in **Analog-to-Digital Converter Measurements** chapter.

Jitter Induced by J-test

The J-test signal was designed to investigate jitter induced from the data modulation of the interface. It carries a tone at a quarter of the sample rate while almost all the data is modulated at 1/192 times the sample rate. For a 48 kHz system these rates are 12 kHz and 250 Hz, and the 250 Hz is effectively a square wave. Data-jitter sensitivity would appear as jitter sidebands to the 12 kHz tone at ±250 Hz and odd harmonics (±750, ±1250 and so on).

The procedure "d-a jtest jitter.apb" is a modification of the previous test ("d-a intrinsic jitter.apb") that uses the J-test signal. (See the **Jittler Theory** chapter for more information on J-test.) Figure 101 is the result achieved from this measurement.

Figure 101. DAC "D" J-test "jitter" measurement.



The integrated jitter result is not useful for this test because the components appearing at the high-frequency part of the graph would dominate the result. These are not due to jitter but a direct measurement of the low-level 250 Hz square wave.

From this graph we can observe that the low-frequency jitter components are identical in amplitude and frequency to the previous result in Figure 100, and there is no sign of components at 250 Hz. Therefore, we can conclude that in this test situation the unit is not showing jitter components due to data-jitter, and that the low-frequency jitter components are originating elsewhere.

Jitter Tolerance

Another jitter test to perform on a DAC is to verify jitter tolerance. Once again, refer to the chapter on **Jitter Theory** for more details.[5]

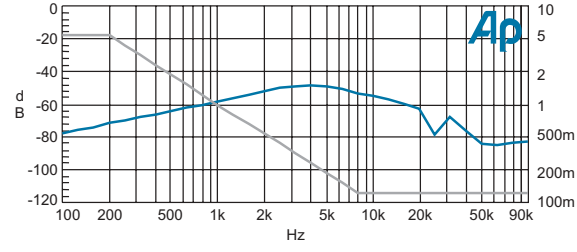
This can be performed in two ways.

- The actual jitter tolerance can be measured.
  At each frequency of interest, adjust the jitter being applied until errors just start to occur, and plot these points.
  Developing an APWIN procedure to perform this task is left as an exercise for the reader!
- The conformance with the AES3 or IEC60958 tolerance templates can be verified.

For this second method, the correct operation of the DAC can be monitored while applying jitter over the range of frequencies and levels from the templates. The monitoring could be by listening to the output of the converter while it is converting a high-level tone at a high frequency. (This is easier to do by listening to the residual after a THD+N notch filter.) Any data errors would result in either bad data, mutes or repeated samples which are likely to be quite audible in these circumstances.

Among the files supplied with System Two Cascade is a test called "DIO D-A JITTER TOLERANCE.at2c." A version of this file with minor modifications has been used to produce the trace in Figure 102.

Figure 102. DAC "A" jitter tolerance verification.



This test and result are in the file "DAC A jitter tolerance.at2c." This shows two traces against the sweep of jitter frequency. The DIO jitter amplitude trace in gray is showing the amplitude of applied jitter which falls from 5 UI (or 10 UI peak to peak) at 200 Hz down to 0.125 UI (0.25 UI pk-pk) at 8 kHz which is the tolerance template required by the specification in AES3 and in IEC60958-4. The EQ curve used by the jitter generator to follow this template is installed with APWIN as "APWIN\EQ\jittol.adq."

The black THD+N trace, which uses peak detection and a 2 second wait for settling, reveals if any errors have been generated. Though the THD+N trace varies (it rises as the jitter-modulated sidebands come out of the THD+N notch

close to the stimulus tone frequency, and then falls as the jitter attenuation starts to reduce the sidebands above 5 kHz), there are no signs of errors. So the device—DAC "A"—has passed the test.

Sampling Frequency Tolerance

The sampling frequency range supported by digital audio equipment can vary quite significantly between models. Different clock recovery techniques may be used. Some may only be able to match to an incoming clock that is within a narrow range (20 parts per million, for example), while others can match to frame rates anywhere between 30 kHz and 108 kHz.

Tests of sampling frequency tolerance require that the incoming frame rate is adjusted while monitoring the correct operation of the device. This can be done in a similar manner to the jitter tolerance test by using the THD+N result to show correct operation.

AES3 / IEC60958 Digital Interface Metadata

For a DAC using the digital audio interface specified in IEC60958 and AES3 there are some aspects of behavior that may depend on information carried in the interface metadata.

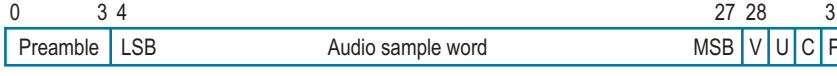


Figure 103. AES3 Subframe

The interface carries the data assocsiated with each audio sample in a 32-bit subframe. Each subframe begins with a synchronization pattern called a preamble that has a duration of 4 bits. The preambles are followed by the audio data,

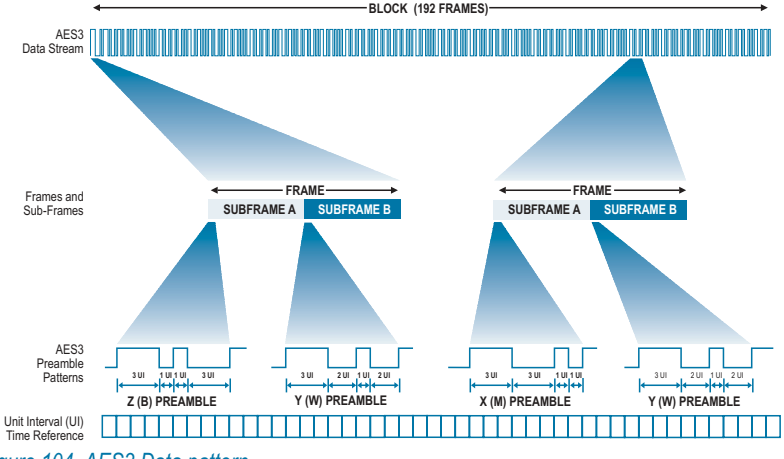
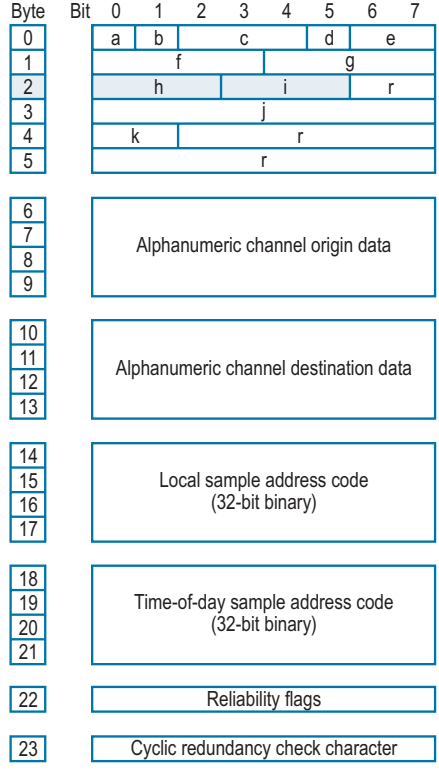


Figure 104. AES3 Data pattern

which is in turn followed by four bits of metadata at the end of each subframe. The first bit of metadata is bit 28, the validity bit; next is the user data bit; then the channel status bit; finally, the parity bit, bit 31.

A frame consists of two subsequent, associated subframes; the frame rate normally corresponds to the source sampling frequency. In the most common implementations, subframe 1 carries the information for audio channel 1, and subframe 2 carries audio channel 2.



a Use of channel status character
b Audio / non-audio use
c Audio signal emphasis
d Locking of source sample frequency
e Sampling frequency
f Channel mode
g User bit management
h Use of auxiliary sample bits
i Source word length and source encoding history
j Future multichannel function description
k Digital audio reference signal
r Reserved

Figure 105. Channel Status Block

There are three unique synchronization preambles. The first subframe of each frame begins with the X preamble, except that every 192 frames this is replaced with the Z preamble. The second subframe always begins with the Y preamble.

The information in each channel status bit is 1 / 192 of the entire channel status data. Each Z preamble indicates the beginning of a new data block, and 192 status bits are collected and arranged into a 24-byte (192-bit) array called the channel status block. The meaning of most of the bits within the channel status block differs between the professional and consumer applications. The professional interpretation is indicated by the first bit of the block (bit 0 of byte 0) being set to one.

Word Length Indication

Each subframe has two fields for the audio data. The fields can be used together to make up a 24-bit word or can be split into a 20-bit field for the main audio—made from the 20 most significant bits of the 24-bit main audio word—with the 4 least significant bits available for other applications. The 4-bit field is called the *auxiliary sample word*. When the interface was defined, it was envisaged that there would rarely be a need for all 24 bits, so unless otherwise indicated the channel is set to use only the 20-bit field for the main audio.

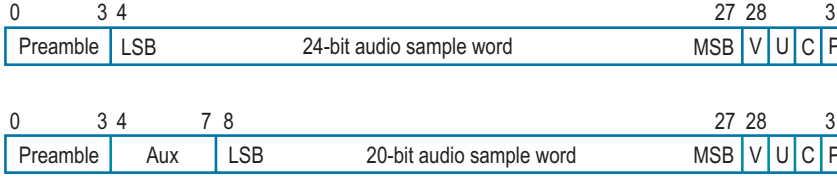


Figure 106. AES3 24-bit audio word, and 20-bit audio word with 4-bit auxiliary sample word.

In the professional version of the interface (AES3 and IEC60958-4) there are control flags in byte two of the channel status block that can be used to determine which mode is being used. The default indication is for the audio word to be 20 bits long, with the application of the auxiliary audio data undefined. There is also a channel status field to indicate the active word length within either the 20 or 24-bit audio data word. A table showing the status flags is in Figure 107.

In the consumer version of the interface (IEC60958-3) the default is also for the audio sample word to be 20 bits. Some specific applications are defined by category codes (in the channel status) and some of the annexes defining the use of the codes state the audio sample word length; for example, the DAT and CD category codes specify a word length of 16 bits. The most recent revision (1999) has also defined a channel status flag to identify which mode is being used, and a channel status field that can indicate the number of active bits

within the 20- or 24-bit audio data word. A table showing the status flags is in Figure 108.

However, as there are very few applications for using the lower 4 bits of auxiliary audio data for anything else, it is quite common for receiving devices to roll up those bits into the audio no matter what the indication in the channel status. In case that field is actually to be used for any other purpose, it is useful to be able to verify how the DAC reacts to the status controls.

Byte Two Bits 0-1-2			Professional Channel Word Length Status Indication
0	1	2	Function:
0	0	0	Default; maximum audio sample word length = 20 bits. Use of auxiliary sample bits not defined.
0	0	1	Maximum audio sample word length = 24 bits. Auxiliary sample bits used for main audio data.
0	1	0	Maximum audio sample word length = 20 bits. Auxiliary sample bits used for coordination channel.
0	1	1	Reserved for user-defined applications.
x	x	x	All other states of bits 0–2 are reserved and are not to be used until further defined.

Byte Two Bits 3-4-5			Professional Channel Word Length Status Indication
3	4	5	Function (with max 24-bit word set in bits 0–2):
0	0	0	Default; word length not indicated.
0	0	1	23 bits
0	1	0	22 bits
0	1	1	21 bits
1	0	0	20 bits
1	0	1	24 bits
x	x	x	All other states of bits 3–5 are reserved and are not to be used until further defined.

Byte Two Bits 3-4-5			Professional Channel Word Length Status Indication
3	4	5	Function (with max 20-bit word set in bits 0–2):
0	0	0	Default; word length not indicated.
0	0	1	19 bits
0	1	0	18 bits
0	1	1	17 bits
1	0	0	16 bits
1	0	1	20 bits
x	x	x	All other states of bits 3–5 are reserved and are not to be used until further defined.

Figure 107. Professional Channel Word Length Status Indication.

Byte Four Bit 0	Consumer Channel Word Length Status Indication
	Function:
0	Default; maximum audio sample word length = 20 bits.
1	Maximum audio sample word length = 24 bits.

Byte Four Bits 1-2-3			Consumer Channel Word Length Status Indication
1	2	3	Function (with max 24-bit word set in bit 0):
0	0	0	Default; word length not indicated.
0	0	1	23 bits
0	1	0	22 bits
0	1	1	21 bits
1	0	0	20 bits
1	0	1	24 bits
x	x	x	All other states of bits 1–3 are reserved and are not to be used until further defined.

Byte Four Bits 1-2-3			Consumer Channel Word Length Status Indication
3	4	5	Function (with max 20-bit word set in bit 0):
0	0	0	Default; word length not indicated.
0	0	1	19 bits
0	1	0	18 bits
0	1	1	17 bits
1	0	0	16 bits
1	0	1	20 bits
x	x	x	All other states of bits 1–3 are reserved and are not to be used until further defined.

Figure 108. Consumer Channel Word Length Status Indication.

This can be done by manipulating the channel status pattern being sent to the DAC for indicating that the field is not part of the main audio word and then measuring to see if, despite this, the analog output is responding to the data in the auxiliary audio field.

An audio signal can be set to modulate only the auxiliary audio field by setting a sine wave of amplitude –121 dB FS (±7.5 LSBs) and with a positive offset of –120.6 dB FS (+7.8 LSBs) with no dither. This covers the range of values from 0.3 LSBs to 15.3 LSBs which will keep the most significant 20 bits at zero. The output level at the sine wave frequency can then be measured and compared with the level with a muted output, and with the output without the DC offset. These measurements are performed by the procedure "d-a aux truncation test.apb."

The results shown in Figure 109 do not indicate truncation. They show that there is a small change in level with the DC offset from –118.6 dB FS to –120.6 dB FS. This indicates that there is a small problem with non-linearity that should be investigated, but the variation is not enough to suggest that the modulation in the auxiliary data is being ignored. The level measured with a muted input is significantly below both these readings.

```
============================================================
Test for truncation of auxiliary audio
============================================================
Test signal level: -121.0 dB FS

Test signal level: -121.0 dB FS

Output bandpass level with muted input
 Ch A level: -127.9 dB FS
 Ch B level: -127.8 dB FS

Output bandpass level with offset. Modulation only in aux bits
 Ch A level: -118.6 dB FS
 Ch B level: -117.8 dB FS

Output bandpass level without offset. Modulation in all bits
 Ch A level: -120.6 dB FS
 Ch B level: -119.3 dB FS
============================================================
```

Figure 109. Results from d-a aux truncation test.apb

If the data in the four LSBs were being ignored, the readings would show a significantly lower level (probably the same level as for the muted input) for the signal with offset, which is only carried in the aux bits.

Sample Rate Indication

The AES3 / IEC60958 digital audio interfaces indicate the sample rate in the channel status. Some equipment requires this indication to match the incoming frame rate for correct operation. This means that the device may mute

if the incoming status pattern is in a “sample rate not indicated” state. This may be inappropriate in many applications and should be checked.

There may also be situations where the channel status indication of sample rate is incorrect but correct operation of the converter is desirable. This can be verified by setting the indication to an incorrect value and observing operation.

NOTE: There are no fundamental reasons why a device should require the sample rate indication in order to function. However, some devices may use it to control part of the clock recovery system. This is not ideal, as such a device would not be compatible with perfectly correct interface signals that do not indicate the sample rate.

Non-Audio and Validity Flags

The digital audio interface stream may be carrying data that is not suitable for conversion to analog through a conventional linear PCM DAC. This data may be multi-channel data-compressed audio, such as Dolby AC-3 or MPEG audio. If this is the case the channel status indicates this by setting the “non-audio” flag to 1.

In other circumstances the data in the audio word may be corrupted. This can be indicated by setting the “validity” flag in the channel status. This flag is often also used to indicate that an error has occurred. There may be reasons for a DAC to ignore the validity flag. Some DACs will mute when it is received and others will not.

Manipulation of the non-audio and validity flags while listening to a signal passing through a DAC output can reveal how it responds to these indications.

Emphasis Flags

Some digital audio recordings and systems use emphasis and deemphasis together at the encoder and decoder (ADC and DAC).

Emphasis amplifies the higher frequencies in a signal prior to encoding. When the signal is decoded the inverse response is applied. This attenuates the higher frequencies in the signal so that the total response is flat.

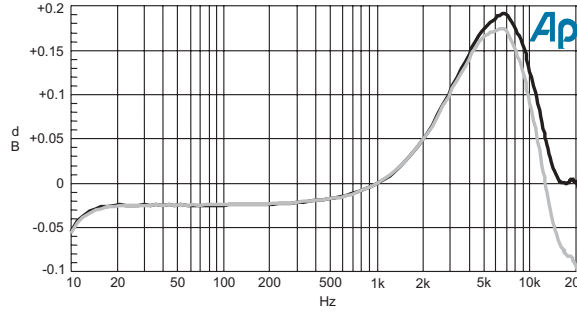
There are two standard emphasis curves:

- In the broadcast environment a characteristic called ITU-T Recommendation J-17:1988 (or *J-17*) is sometimes used. The largest application of this is with NICAM digital stereo sound for television; NICAM digital stereo is not a linear PCM format, so it would not be fed directly into a DAC. However, there may be applications where the emphasized signal is exchanged in linear PCM format, and in those circumstances it could be fed into a linear PCM DAC.

- The Compact Disc format permits the use of emphasis, which has a different frequency characteristic from J-17. There are, however, very few CDs recorded with emphasis. Recording engineers have problems managing the frequency-dependent headroom that results from using an emphasized recording chain. Also, some CD players and DACs apparently do not correctly deemphasize the signal on replay, and on those devices an emphasized recording would replay incorrectly.

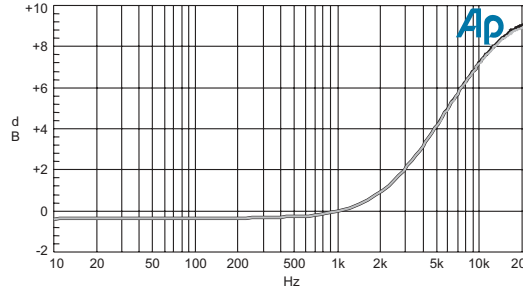
The deemphasis characteristic of the DAC can be verified using the technique described in the section on passband deviation on page 92. This uses the procedure "d-a passband.apb." Before running the procedure select the emphasis characteristic required. This can be done in APWIN using the Digital I/O panel "PreEmphasis" selection. The transmitted channel status pattern should also be configured to indicate the emphasis selected, in order to activate the deemphasis circuit in the DAC.

Figure 110. DAC "D" deemphasis response deviation.



An example of an emphasis-deemphasis passband response sweep is shown in Figure 110. CD-type preemphasis has been selected on the generator output, and the DAC deemphasis has been activated through the channel status data. Any mismatch between the emphasis and deemphasis causes a gain variation in the frequency sweep.

Figure 111. Measurement of emphasized signal without deemphasis.



This graph has a small deviation which confirms that the characteristic has been selected, but the deephasis has a response error of almost 0.2 dB at 7 kHz.

If the DAC was not performing deemphasis, the plot would show the preemphasis characteristic for CD as shown in Figure 111.

The J-17 emphasis characteristic has more high-frequency boost starting at a lower frequency, but the same technique can be applied.

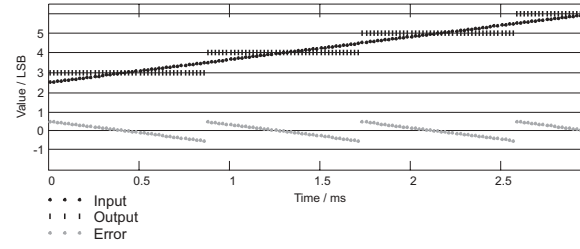
DITHER ANNEX

Dither is used to make a digital signal behave more like an analog signal.

Without dither, the quantization error which results from sampling a low-level signal varies with the signal. This variation is very unnatural, and it changes the nature of low-level signals in an obvious manner. A good example of this is a decaying piano note, which sounds very distorted before it disappears—possibly very suddenly. It is preferable for the error to be noise-like, uncorrelated to the audio signal. Dither is used to achieve this.

Dither is a small noise-like signal added to the input. This presents a random component to the quantizer. When the signal is between two quantization levels, the quantizer then selects either of the two levels in proportion to how close they are to the (dithered) input level, which results in the average output level matching the input level. There is still quantization error, but now it is random. It is decorrelated from the signal, and the distortion and modulation effects of the correlation can be minimized.

Figure 112. Quantization of a ramp without dither.



When a signal is quantized with dither, it has the effect of making the resultant signal sound like the original, at the price of added noise.

The mechanism is illustrated in Figure 112. The input signal is a simple ramp, and the input samples are shown as black dots. The quantization process, in this example, takes the nearest integer value of least significant bits (LSBs). For most of the first millisecond the output of the quantizer (shown in black) remains constant at 3 LSB, even though the input signal is rising. At about 0.9 ms the input signal crosses the quantizing decision threshold at

3.5 LSB, and the output jumps to the next level, 4 LSB. (Other quantizers may simply take the largest integer that is not more positive than the input, but the difference is only a DC offset of 0.5 LSB.)

Notice the pattern of the short black lines, which represent the output of the quantizer. The ramp has been converted into a staircase. The quantization error—the difference between input and output—is displayed in gray. This shows a regular sawtooth of amplitude 1 LSB. The error is highly correlated with the signal, with a slope equal and opposite to the slope of the signal. This correlation is undesirable as it produces strange tonal components to the background noise floor.

Dither probability density

Thermal noise, the type of noise encountered in analog signals, is Gaussian; that is, a graph of the distribution of its probable values follows the familiar bell-shaped Gaussian curve. Gaussian noise, however, is not the most effective dither for digital audio signals.

Figure 113. Rectangular and Triangular dither.

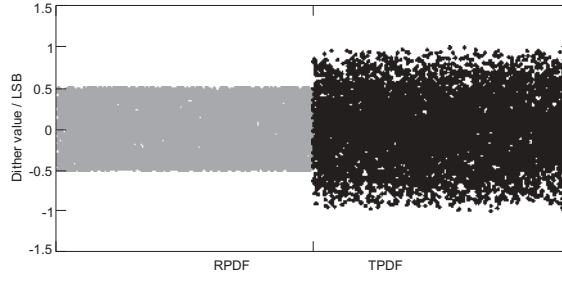
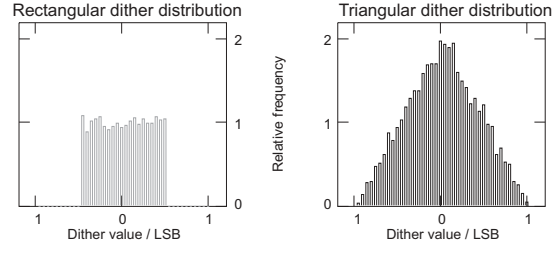


Figure 114. Dither distribution.



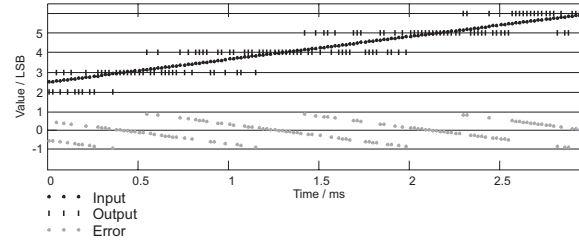
The two forms of dither that are commonly used are called *Rectangular Probability Density Function* (RPDF) dither and *Triangular Probability Density Function* (TPDF) dither. The distributions of their probable values are illustrated in Figures 113 and 114.

RPDF Dither

The RPDF dither value has an equal chance of falling anywhere in a range that is 1 LSB wide. (The histograms shown in Figure 114 are uneven because they are derived from 8192 samples of actual dither.)

The effect of the RPDF dither is shown in the following figure.

Figure 115. Quantization of a ramp with RPDF dither.



The extra uncertainty added by the dither means that for input values between two quantization levels the output is a mix of both levels. The dither has the effect of making the relative number of output values from the quantization levels above and below the input signal such that their mean value is the same as that of the input. This means that the mean value of the error is zero (or, for some quantizers, a DC level).

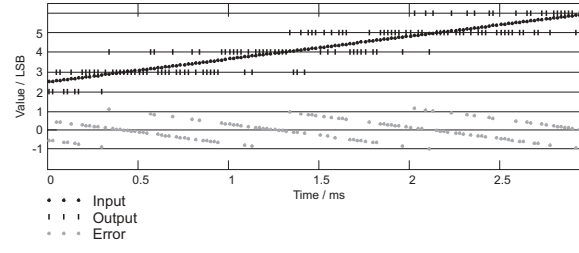
The average noise penalty for adding RPDF dither (at an otherwise perfect quantization) is to double the noise "power," a 3.01 dB increase.

Note that when the input signal is close to a quantization level, RPDF dither has little or no effect. This means that the rms error is very much lower at these points. The effect of this is signal-dependent noise modulation—which is not ideal—so TPDF dither is often preferred.

TPDF Dither

TPDF dither is generated by adding two RPDF values together. This has the effect of doubling the peak-to-peak variation but changing the shape of the distribution in such a way that the dither value is more likely to be in the middle of the range.

Figure 116. Quantization of a ramp with TPDF dither.



The TPDF dither has the effect of making the rms error independent of signal level (as well as maintaining the property of the RPDF to make the mean error zero). This is the form of dither recommended for most measurement applications. The average noise penalty for adding TPDF dither (at an otherwise perfect quantization) is to triple the noise "power," a 4.77 dB increase. This is 1.76 dB more than RPDF dither.

Shaped dithers

An important variation of TPDF dither is called *high-pass dither*. High-pass dither is not spectrally flat but is weighted towards high frequencies. This has the advantage that the resulting quantization noise has a slightly lower audibility; also, the generation of high-pass TPDF dither is slightly simpler than flat TPDF dither. It is very commonly used.

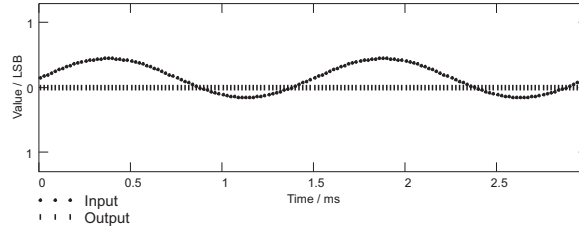
The mathematical properties of this dither do not completely decorrelate the rms error from the signal level, leaving a very small correlation which may be important in some applications, such as within a recursive filter. For this reason it is not used in all applications.

Shaped dithers, such as high-pass TPDF dither, are not to be confused with *noise shaping*. Noise shaping affects the quantization noise of the data truncation as well as the dither, while shaped dithers cannot reduce the noise contribution from the quantizer truncation itself. Noise shaping also has a higher total noise penalty than shaped dither, so that even though the noise density is lowered at some frequencies the total unweighted noise power in the frequency range from DC to half the sampling frequency is increased.

Dithering a low-level tone

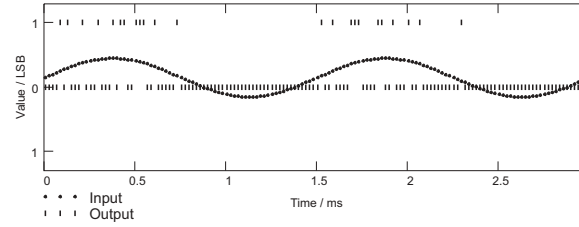
Figures 117, 118 and 119 illustrate the most startling benefit of dither; or, put another way, the most obvious deficiency of working without dither. They show the effect of a quantizer on a low-level tone. The input tone has a peak-to-peak amplitude of 0.6 times the quantization step size so it is "below the LSB" and it has been given a DC offset of +0.15 LSB.

Figure 117. Quantization of a tone without dither.



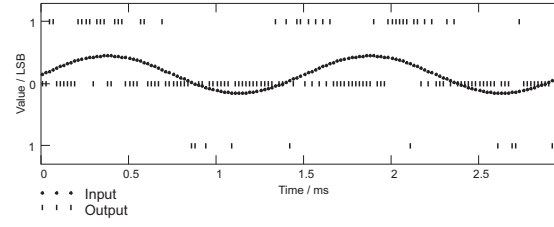
Without dither there is no modulation on the output. The input signal does not cross any quantization decision levels (for this quantizer the nearest are at –0.5 and +0.5 LSB) so it has disappeared altogether. There is no noise.

Figure 118. Quantization of a tone with RPDF dither.



The addition of RPDF dither has produced an output signal. If it is examined in the frequency domain there is white noise and a tone. The level, phase and frequency of the tone in the output signal can be shown to match that of the input. The rms level of the quantization noise is correlated with the signal as the negative half of the cycle (which is closer to the mid value for 0) is less noisy than the positive half; the noise is modulated by the signal.

Figure 119. Quantization of a tone with TPDF dither.



With TPDF dither, the output noise has increased so that now 3 different levels are being used. Though not very obvious from the graph, the rms quantization noise amplitude is no longer correlated with the signal. The output of the quantizer sounds like a tone in a steady background of white noise.

List of Procedure Files

The following APWIN Basic procedures are referred to or used in this Application Note:

- d-a aux truncation test.apb
- d-a full scale compression v freq.apb
- d-a full scale thd v freq.apb
- d-a gain.apb
- d-a idle channel fft v level.apb
- d-a idle channel fft.apb
- d-a idle channel noise.apb
- d-a imd_fft.apb
- d-a intrinsic jitter.apb
- d-a jtest jitter.apb
- d-a jtf_fft.apb
- d-a low level distortion fft.apb
- d-a noise floor FFT.apb
- d-a noise modulation.apb
- d-a output at full-scale.apb
- d-a output clipping.apb
- d-a output gain stability.apb
- d-a passband.apb
- d-a signal to noise.apb
- d-a stopband fft.apb
- d-a stopband sweep.apb
- d-a subtracting test signal noise.apb
- d-a tech note utilities.apb
- d-a thdandn.apb
- d-a THDN output fft.apb

Two additional files have been provided for consistency and ease of use:

- d-a Menu.apb
- d-a Setup.at2c

These files are on the companion CD-ROM. Please check the README.DOC file in the same folder for further information.

You may also download the files from the Audio Precision Web site at audioprecision.com. Check the **What's New** link for updated procedures.

These procedures and tests are designed for use with System Two Cascade, but with minor changes can be modified to work with System Two as well.

References

1. AES3-1992—"Recommended Practice for Digital Audio Engineering—Serial Transmission Format for Two-Channel Linearly Represented Digital Audio Data," J. Audio Eng. Soc., vol. 40 No. 3, pp 147-165, June 1992.
   [The latest version including amendments is available from www.aes.org.]
2. IEC-60958—"Digital Audio Interface" Second Edition, International Electrotechnical Commission, Geneva, December 1999.
3. AES17-1998—"AES standard method for digital audio engineering—Measurement of digital audio equipment" J. Audio Eng. Soc., vol. 46 No. 5, pp. 428-447, May 1998.
   [The latest version is available from www.aes.org].
4. IEC 61606: 1997—"Audio and audiovisual equipment—Digital audio parts—Basic methods of measurement of audio characteristics," International Electrotechnical Commission, Geneva.
5. See the chapter **Jitter Theory** beginning on page 3 of this book.
6. See the chapter **Analog-to-Digital Converter Measurements** beginning on page 37 of this book.
7. Julian Dunn—"The benefits of 96 kHz sampling rate formats for those who cannot hear above 20 kHz," Preprint 4734, presented at the 104th AES Convention, Amsterdam, May 1998. [This is available from www.nanophon.com/audio.]

The Digital Interface

Introduction

The AES3[1] and IEC60958[2,3,4] standards provide a common interface for digital audio signals. This chapter describes the interface and highlights some of the aspects that may require measurement to verify conformance.

The interface defined in AES3 and IEC60958-4 is commonly called the "professional standard" interface; IEC60958-3 defines the "consumer standard" interface.

There are a number of differences between the professional and consumer standards which in some cases can render them completely incompatible. For proper performance, the consumer and professional interfaces should not be mixed. However, they are similar enough that in many situations, given the right electrical connections, the embedded audio can carried from one standard to the other.

By requiring conformance with these standards, a user of digital audio equipment rightfully expects compatibility within devices adhering to a standard. Compatibility allows interconnecting the equipment without suffering loss of performance or functionality—which is, after all, the aim of interface standardization.

The digital audio interface carries three types of information:

- timing information,
- audio data, and
- non-audio data.

Some of this information can be degraded by implementations of the interface that conform to the standard but are not ideal. We shall consider aspects

of interface behavior and performance that may make one implementation more useful than another, such as the ability of a receiver to tolerate incoming jitter or a wide range of frame rates, or the precision with which a transmitter maintains synchronization.

This chapter also discusses synchronization. For real-time applications, such as recording, replay, or transmission it is important to have sample synchronization between equipment. The AES11[6] specification is a useful basis for defining good practice in this area, and this chapter describes the principle by which AES11 can produce a form of synchronization.

Basic Interface Format

Bi-phase coding

The simplest coding of binary pulse code modulation (PCM) audio data is to code a “one” as a logic high, and a “zero” as a logic low. This is not an ideal format electrically. Consider the case where all the bits are set to ones (or zeros) for a period of time. Another signal—a bit clock—would be required to identify the individual bits.

The coding used in this interface format is more sophisticated. This bi-phase coding scheme has an embedded “bit clock” which can also be used to recover the sampling frequency. Bi-phase coded PCM has a mean voltage of zero, eliminating DC on the interface, with the result that the data can be AC-coupled through a transformer or series capacitor. The coding works like this:

Each data bit has a *time slot* that begins with a transition and ends with a second transition, which is also the beginning transition for the next time slot. If the data bit is a “one,” an additional transition is made in the middle of the time slot; a data “zero” has no additional transition.

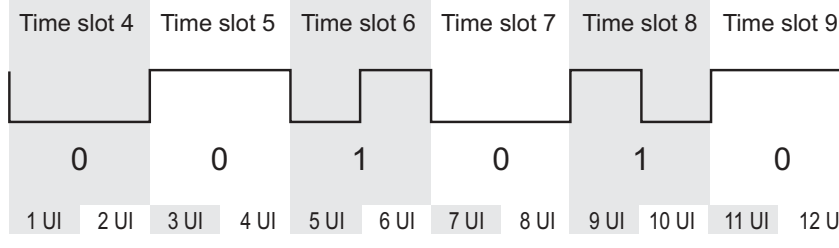


Figure 120. Bi-phase coding.

Figure 120 illustrates this bi-phase coding with 6 bits of data.

As you can see, even with a digital DC signal of continuous data zeros, there are still transitions at each time slot (or bit). The clock is always carried

by these regular transitions, the interface signal is now clearly AC, and the direction of the transitions (or signal polarity) becomes irrelevant.

Unit interval

Many of the timing parameters on the interface are defined in terms of the *unit interval*, or UI. This is the shortest nominal interval between transitions. The bi-phase coding introduces a second transition (indicating data "one") into the time slot, which means that a time slot is defined as 2 UI wide, as shown in Figure 120.

Framing

The data carried by the interface is transmitted serially. In order to identify the assorted bits of information the data stream is divided into *frames*, each of which are 64 time slots (or 128 UI) in length. Since the time slots correspond with the data bits, the frame is often described as being 64 bits in length, but the preamble sections (see below) break this correspondence.

Each frame consists of two *subframes*. Figure 121 shows an illustration of a subframe, which consists of 32 time slots numbered 0 to 31. A subframe is 64 UI in length.

The first four time slots of each subframe carry the *preamble* information. The preamble marks the subframe start and identifies the subframe type.

The next 24 time slots carry the audio sample data, which is transmitted in a 24-bit word with the *least significant bit* (LSB) first.

Figure 121. The AES3 subframe (24-bit audio data).

After the audio sample word there are four final time slots, which carry:

- the **validity** bit
- the **user data** bit
- the **channel status** bit, and
- the **parity** bit.

The two subframes in a frame can be used to transmit two *channels* of data (Channel 1 in subframe 1, Channel 2 in subframe 2) with a sample rate equal to the frame rate; or, instead the two subframes can carry successive samples of the same channel of data, but at a sample rate that is twice the frame rate.

Preambles

A preamble is a distinctive data pattern carried in the first 4 time slots of a subframe to mark subframe and block starts. There are three preambles, all of which break the bi-phase coding rule by containing one or two pulses which have a duration of 3 UI. This rule-breaking means that the pattern cannot occur anywhere else in the pulse stream.

Subframe 2 always begins with a Y preamble. Subframe 1 *almost* always begins with an X preamble, with this exception: every 192 frames the X preamble in subframe 1 is replaced with a Z preamble, which indicates a *block start*. This provides framing for the information carried in the channel status fields—the *channel status block*.

The interface signal is insensitive to polarity so the preambles can be found to start with a falling transition:

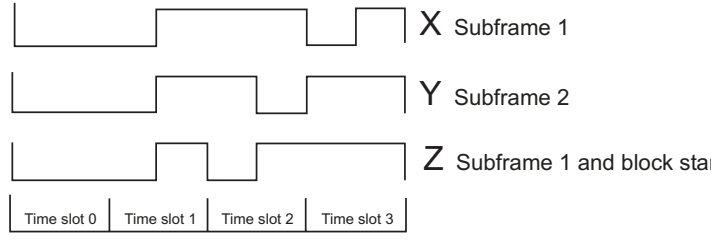


Figure 122. Preamble patterns with a falling first transition.

or with a rising transition:

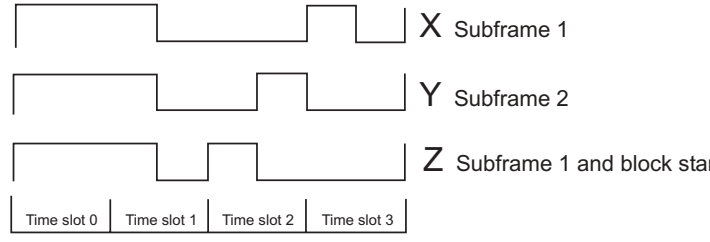


Figure 123. Preamble patterns with a rising first transition.

Under the bi-phase coding rules there should be a transition between each time slot; but the preambles, of course, each have two three-bit pulses, so for each preamble there are two time slot boundaries without transitions. The first of these bi-phase coding violations is in the same place for each preamble—after time slot 0. This identifies that a new subframe has started. The pattern that follows then identifies the type of subframe.

The time slot numbers in Figures 122 and 123 correspond with the numbers shown in Figure 121 and are 2 UI wide. The preambles are 8 UI wide and so take the same amount of time as 4 bits.

Audio data

After the preamble the audio data is transmitted with the LSB first. For audio word lengths less than 24 bits the data is justified to the *most significant bit* (MSB) and zero-filled below the LSB, as shown in Figure 124.

Figure 124. The AES3 subframe (16-bit audio data).

In some of the audio modes—those that transmit 20 or fewer bits of main audio data—the first four bits after the preamble can be used for another signal known as *auxiliary audio data*. This mode has the subframe structure of Figure 125. If this auxiliary data is used, then the channel status (see page 7) should indicate that the maximum word length is 20 bits, and the receiver should mask off the auxiliary audio field so that any values there are not added to the main audio sample values. Unfortunately, many decoders are not that sophisticated.

Figure 125. The AES3 subframe (20-bit audio data with auxiliary data).

The use of auxiliary audio is very rare. One application is for voice communications, and AES3 suggests the auxiliary bits can be used for coordination or talk-back purposes. One way of doing this is to transmit a 12-bit channel at a sample rate of one-third of the frame rate. Other applications, like the use of the auxiliary audio field for transmission of a data-compressed version of the main audio signal may also be possible.

Validity bit

The **validity** bit was originally intended to somehow qualify the transmitted data. If the bit is set then the data is identified as *not suitable for conversion* to analog audio. However, there are some applications that set the validity bit if an error has been found and concealed. This behavior is quite common for Compact Disc players, for example.

This confusion as to the function of this bit means that it is not easy to decide how a receiver should behave when a sample is marked as invalid.

When the IEC60958 or AES3 stream is used to transmit data that does not represent linear PCM audio, then the bit should certainly be set. This has at least a chance of causing linear PCM replay equipment to mute, which is preferable to an attempt to reproduce the data as an audio signal.

The specifications for carrying data-compressed audio on AES3 or IEC60958 require this bit to be set, so that linear PCM receiver devices will recognize the need to mute. This has the potential benefit of stopping the receiver from producing a burst of high-level noise from the data before the channel status pattern (which is only updated every 192 frames) can identify the signal as not being linear PCM audio. This is indicated in channel status by the **non-audio** bit.

User bit

The **user** bit can be used to carry user-specific information. In practice this means application-specific information for consumer devices such as CD or DCC.

The consumer specification, IEC60958-3, has defined a *packet-based* format for carrying program-related information in the user data stream and defines rules for the preservation of the user data by various classes of equipment.

Consumer format user data transmission classes and behavior		
Class	Equipment Behavior	Examples
I	Generating original user data	CD, DAT, DCC, mini disc
II	User data transparent (or no user data output)	Sound processor
III	Mixed mode user data (transparent but with some exceptions)	Mixer, sample rate converter, sound sampler

Table 1. Consumer format user data transmission classes and behavior.

In the consumer format the user data streams from subframe 1 and subframe 2 are combined to form one stream at 2 bits per frame. This means that for a frame rate of 44.1 kHz the user data rate is 88200 bits/sec.

The professional specification, AES3 (and IEC60958-4) has channel status patterns that allow the indication of various formats of user data, specifically:

- 192-bit block with same block start as channel status;

- AES18 (packet based);
- IEC60958-3 user-data format.

In the professional format (except for the IEC60958-3 user-data format mode) the user data streams from subframe 1 and subframe 2 are associated with the audio channel being carried in that subframe. Therefore there are two streams, each with one bit per frame. (The case of user data transmission with the single-channel double-sample- rate mode is not defined explicitly. It may be logical to combine the user data stream into one with 2 bits per frame, since both subframes carry the same channel.)

At the time of publication there were few applications of the professional format that use the user-data channel—so subtleties of user data implementation may not be useful.

Channel status bit

The **channel status** information is transmitted in a block of 192 bits. A frame starting with preamble "Z" (see above) identifies the first bit of the block. The Z preamble is sometimes called "block start."

There are independent **channel status** bits for both subframe 1 and subframe 2, so there are actually two blocks. Quite often these two blocks carry identical data, and many receivers only examine the data from one of the subframes.

Some of the **channel status** bits affect how equipment should treat the data in the audio sample word. In particular the **non-audio** and **emphasis** fields make significant differences to the way the data needs to be interpreted.

If the **non-audio** bit is set then the audio sample word is not suitable for decoding as linear PCM data. The name "non-audio" is a bit of a misnomer, as audio using data-compressed formats, such as MPEG, DTS, Dolby AC-3 and Dolby E are flagged as **non-audio** because treatment of their raw data as if it was linear PCM would be inappropriate and would result in the generation of high level noise. The standard for carrying these data-compressed formats is IEC61937[9], for consumer applications, or SMPTE 337M[10], for professional applications.

If the **emphasis** field indicates that the signal has emphasis, then deemphasis should be applied in any conversion to analog. The only emphasis supported by the consumer format is the CD type. This has a high-frequency boost shelf with time constants of 50 µs and 15 µs for the zero and pole. The professional format supports this format as well as J-17[7] emphasis, which has time constants at approximately 333 µs and 38.5 µs.

Apart from the first two bits, the meaning of the bits within the block is defined differently for the consumer and professional formats. More detailed in-

formation on status bits for both the professional and consumer implementations can be found in the **Channel Status Annex** beginning on page 190.

Parity bit

The **parity** bit is used to maintain *even parity* for the data as a means of error detection. Specifically, even parity in the interface signal means that there is an even number of mid-cell transitions in the data area, which spans time slots 4 to 31. Since there is an even number of all other transitions, even parity means that there is an even number of transitions in every frame.

Even parity has the effect of starting each subframe with a transition in the same direction all the time. As a consequence, the transmitter of an AES3 or IEC60958 stream does not need to calculate parity, and the receiver needs only to verify (since the parity bit is the last bit of the subframe) that the state of the second half of the parity bit is always the same as its state in the previous subframe.

If an error occurs, it is most likely to be a *pair* of missing transitions, that is, both edges of an individual pulse that was not detected. If a pair of transitions are missing, the parity will not change, even though there was an error. In fact, in many schemes for decoding the bitstream, a genuine parity error is impossible.

However, a violation of the bi-phase coding could be detected at such a point, since at least one of the missing transitions would be on the time slot boundary. For this reason it is much more useful to check bi-phase coding violations in identifying errors than to use the parity bit.

Electrical properties

There are three basic electrical formats:

- The balanced format. This is the primary professional format and is defined in AES3.
- The consumer coaxial format. This is defined in IEC60958-3.
- The professional coaxial format. This is defined in AES-3id[11] and in SMPTE276M[12]. This format was developed to use analog video transmission systems for digital audio transmission.

Balanced format

This uses a *shielded twisted pair* cable to carry the interface signal differentially and is normally coupled to equipment with a standard XLR connector. (See IEC60268-12). It has the advantage that we can use cabling that is in com-

mon with analog interfaces. However, it can also result in confusion between the two types of connections.

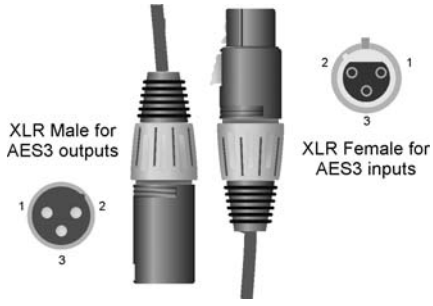


Figure 126. XLR connectors for balanced format AES3 interface.

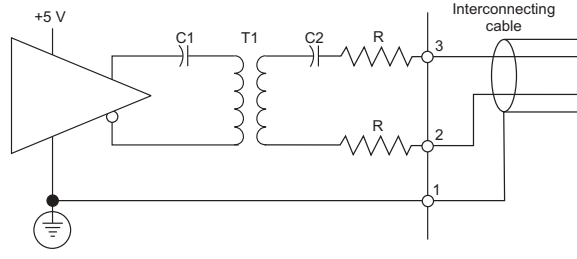
Though not required by AES3, many designs use small *pulse transformers* at the receiver and transmitter. In the same way as the balanced interfacing application for analog audio, the transformers offer advantages for reducing emissions and susceptibility to inductive coupling as a consequence of the improved current balance in the line.

Transformers are required by the EBU version of the specification, EBU 3250[5]. This is motivated by the need to maintain a high common-mode impedance at the cable terminations so that crosstalk is minimized. Crosstalk is of particular concern for EBU members because of the large amount of cabling run in parallel at broadcast installations.

Like the other electrical formats there is a requirement for the cable impedance and the transmitter and receiver termination impedances to be matched. In this case the nominal impedance is 110 Ω.

At the transmitter, the amplitude of the signal should be between 2 V and 7 V peak-to-peak with the output terminated. Without termination (assuming conventional implementation of source impedance) the generator voltage would be twice that. This can be driven from complementary outputs with logic operating from 3.3 V or 5 V rails and using a 1:1 transformer. A line driver circuit is shown here:

Figure 127. AES3 balanced format line driver.

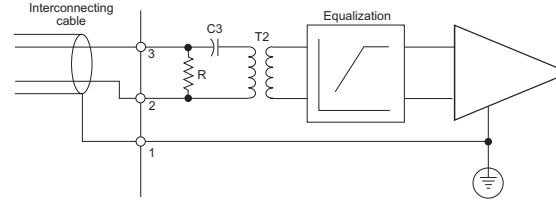


At the receiver, the amplitude of the signal may be significantly reduced through cable losses; or, there may be no loss at all. As a consequence the range of possible receiver interface voltage levels is much greater than at the transmitter.

These losses have a greater effect on the high-frequency component of the signal, with the result that the heights of the shorter pulses will become lower than that of the longer pulses. This distorting effect on the signal means that it is not adequate to refer to the "peak-to-peak amplitude" of the signal but instead to measure the size of the "eye" in an eye diagram, which is described on page 173.

It is possible to have an equalization circuit to compensate for some of the distortion, and this would be fitted prior to the differential-to- single-ended data slicer, as shown in Figure 128.

Figure 128. AES3 balanced format receiver with equalizer.



Though they were popular in the early applications for AES3, equalizers are not used very often in modern designs. This may be because there is an expectation that cable losses will not be as significant, perhaps because lower-loss cable is used; and also because in most applications the cable length is quite short—significantly less than 100 meters. In the early 1980s I found that with the standard BBC-specified shielded twisted pair cable used for analog audio signal distribution, it was possible to get reliable operation over 100 meters without an equalizer, and that this could be extended to 250 meters with an equalizer. Moreover, with short cable lengths, the equalizer can be a liability, increasing the sensitivity to errors from cable reflections due to impedance mismatch.

Unbalanced Format

The two unbalanced formats use a 75 Ω impedance- matched coaxial cable for transmission.

The consumer version has a transmitted level of 0.5 V peak-to-peak and uses the same kind of coaxial connecter (the RCA "phono"connector, defined in 8.6 of Table IV of IEC 60268-11) that is used for consumer analog connections.

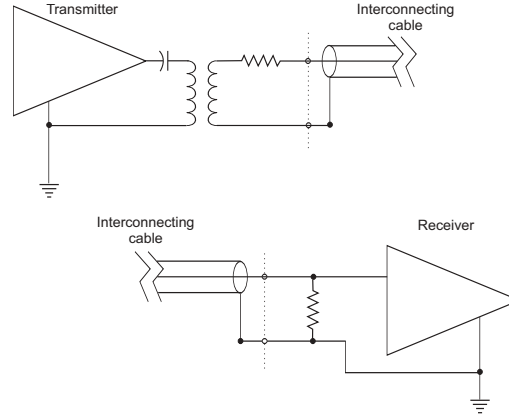
Figure 129. RCA "phono" connector for consumer interface.

The professional version has a level of 1 V peak-to-peak and uses a BNC connector (see IEC 60169-8).

Figure 130. BNC connector for unbalanced professional interface.

The same kind of interface signal distortion occurs in the unbalanced version as for the balanced version of the interface, so the eye diagram is also used here in to assess and define signal levels and receiver characteristics.

Figure 131. Unbalanced format transmitter and receiver.



In the consumer application (IEC60958-3), short lengths (less than 1 meter or so) of cable designed for analog audio interconnections will work quite adequately, even though the cable transmission characteristics are poor.

The professional specification (AES-3id and SMPTE276M) is intended for use over much longer distances, and uses professional analog video cable. 75 Ω video coax cable has an appropriate frequency characteristic for this application, and long transmission distances are possible. AES-3id illustrates that transmission distances of more than 2 km can be achieved with sophisticated equalization schemes. (The two specifications are similar and interoperate. However, the SMPTE version requires tighter tolerances.)

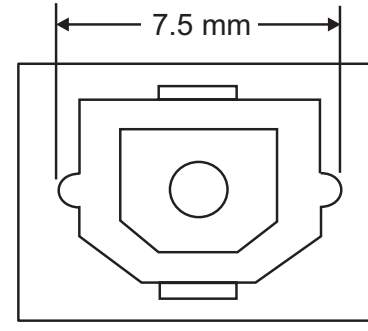
Optical Format

In common use for the consumer format is an optical interface called TOSLink®, after the version sold by Toshiba. This uses plastic multi-mode optical fiber with a red light-emitting diode (LED) transmitter and a photo diode

receiver. The transmission distance is limited to less than a few yards (or meters). IEC60958-3 has a section for defining this format but it is still "under consideration." As a result, methods of defining receiver and transmitter performance do not have a benchmark to evaluate against.

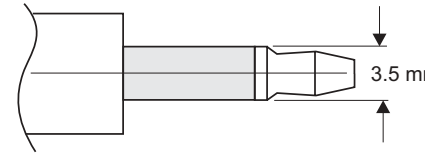
There are two connector formats for the optical fiber. The older and more widely used uses a friction lock connector type F-05 specified in IEC60874-17, shown in Figure 132.

Figure 132. TOSLink® optical connector.



This connector is too large for portable audio equipment, so a coaxial connector has been developed that appears quite similar to the electrical 3.5 mm mini-jack plug used for personal stereo headphones. This is shown in Figure 133.

Figure 133. 3.5 mm optical connector.



The socket for this connector has the advantage that it can double-up as the analog headphones jack and hence use no extra space on the equipment surface.

Synchronization

The embedded clock defined by the interface bit-cell transitions, the subframe and the frame boundaries can be used as a timing reference by equipment to derive timing for converters, processors, and digital outputs. For digital outputs AES11 defines limits for the timing offset between the frames of the reference input signal and the frames of the outputs.

In some cases the timing reference is provided by another signal or clock, and the incoming signal needs to have been already synchronized to that clock. AES11 defines a specification for this sort of synchronization, and It also cov-

ers synchronization of the digital audio interface with video signals. For further details see the **Synchronization** sidebar, which begins on page 181.

Output Port Measurements

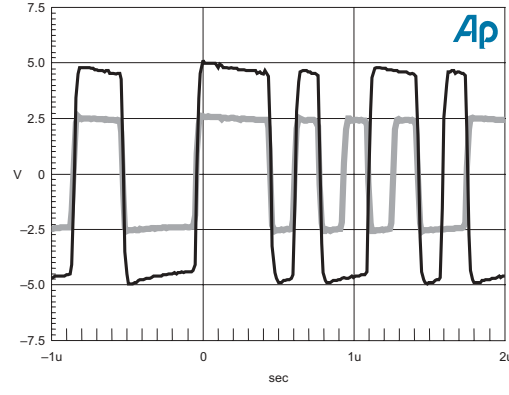
Output port impedance

A simple method of checking an output port impedance is to measure the level without a termination, and then measure it again terminated with the nominal line impedance (110 Ω for AES3 and 75 Ω for coaxial signals). A conventional design with the correct source impedance should show a ratio between the two levels of 2:1.

This measurement is best done by observing the waveform as an amplitude-versus-time trace. The reason for this will be made clear later.

An amplitude-time trace can be measured with an oscilloscope or with the INTERVU package of APWIN. The trace in Figure 134 shows the result from a back-to-back INTERVU test of the System Two Cascade digital output.

Figure 134. System Two Cascade interface waveform, as measured by INTERVU. Black is unterminated, gray is terminated.

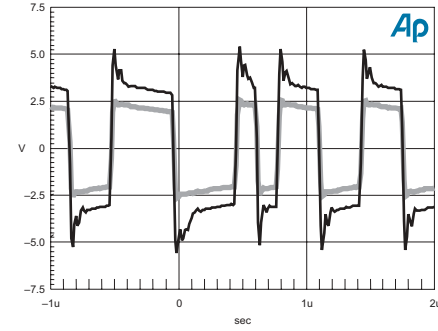


This test result and setup is in the file “output term test back to back.at2c.” This shows very close to a 2:1 ratio between the two traces, as expected with correct source termination. The higher amplitude waveform, shown here in black, is the unterminated one.

I have chosen to look at the part of the waveform near a preamble. The static pattern around the preamble makes it easy to make direct comparisons. In APWIN this is determined by selecting a preamble as the trigger.

In comparison, Figure 135 (from “output term test DAT.at2c”) shows the same measurement on the AES3 output of a DAT machine. The unterminated and terminated waveforms are different shapes. There is not a consistent 2:1 ratio in amplitude.

Figure 135. Interface waveform output from DAT machine. black is unterminated, gray is terminated.



STANDARDS

There are several published standards documents that either define basically the same digital audio interface as AES3, or the similar consumer-targeted equivalent, in IEC60958-3. There are also standards that are used in conjunction with these interfaces.

IEC60958:1989 (previously known as IEC958:1989)

This has been replaced by the multi-part document, IEC60958-n. It defined both the professional and consumer applications and the two-connector types for electrical connection. By accident it did not require that the professional format used the XLR and the consumer format used the coaxial connection.

IEC60958-1, -3, and -4

The revision of IEC958:1989 involved splitting the standard into three parts. Part 1 covers the aspects common to both consumer (which is in part 3) and professional (part 4) applications. As the document has a different reference it is also Edition 1—which may be confusing.

AES3-1992[1]

The primary definition of the professional format is in this document. This undergoes regular revision by amendment or new edition. It is possible for interested parties to contribute to this process by joining the working group on digital input/output interfacing, SC-02-02. Further information on joining AES standards working groups is available at http://www.aessc.aes.org.

EBU 3250 (Ed. 2, 1992)[5]

This document has been produced by the European Broadcasting Union (EBU). It is similar to AES3-1992 (without the amendments) apart from one key difference—the EBU document specifies that transformers shall be used between the cable connection and the receiver and transmitter electronics. Transformers are optional for AES3.

ITU-R BS647-2 (1992)[8]

This is very similar to EBU3250. The International Telecommunications Union (ITU) is an intergovernmental organization that is part of the UN.

IEC60958-4 (Ed. 1, 1999)[4]

This part of IEC60958 defines the professional interface. At the time of writing (early 2001) it is similar to AES3-1992 with amendments 1 (1997) and 2 (1998) but not amendments 3 or 4 (both 1999); the key difference is that it does not support sampling frequencies above 48 kHz. There is an amendment in process within IEC to rectify this.

Technical Report IEC60958-2:1994 (or IEC958-2:1994)

This document is not a standard. The specification describes a method of carrying software information in the channel status stream of the consumer application of IEC60958:1989. It uses the setting of the channel status mode field in byte 0 of the channel status block to distinguish this use of the channel status block. Originally it was proposed as an amendment to the IEC958:1989 standard but was rejected. With the conversion of that standard to a three-part standard, with parts called IEC60958-1, IEC60958-3, and IEC60958-4, this document appears—at first sight—to be part 2 of a 4-part standard. That is not the case.

IEC60958-3 (Ed. 1, 1999)[3]

This part of IEC60958 defines the consumer interface in all respects except the optical interface. At the time of writing (early 2001) it does not support sampling frequencies above 48 kHz. There is an amendment in process within IEC to rectify this.

AES-3id 1995[11] and SMPTE 276M-1995[12]

These two documents both define a variant of AES3 that is transmitted over 75 Ω coaxial cable at a level of 1 V (peak to peak). The impedance and level are chosen to be compatible with broadcast analog video interfacing and allow the use of some of the same cabling and routing infrastructure.

The two specifications are different. The SMPTE specification has tighter tolerances for some parameters and intended for use with dedicated interfaces on equipment for high performance. AES3-id has more relaxed specifications that permit use with passive converters between the 110 Ω balanced and the 75 Ω coaxial formats.

AES11-1991[6] (Synchronization)

This standard defines rules to be followed to ensure synchronization of digital audio equipment together and with video. A special AES3 signal is used to distribute a timing reference from the synchronization (clock) master to all the other (slave) devices in the synchronized system. This timing reference is called a Digital Audio Reference Signal (DARS).

When making this simple assessment it is important to use a short cable from the device under test (or *DUT*) in order to keep the effect of reflections to a minimum.

A single-valued measurement of the waveform amplitude such as a peak-to-peak measurement under APWIN, for example, will not reveal the pulse distortion of the unterminated case. In fact, the peak-to-peak measurements on this waveform are 9.8 V and 5.1 V, which is a ratio of 2.08:1, and similar to the back-to-back measurement of the System Two Cascade.

Output port amplitude

In the previous section the assessment of output impedance used amplitude measurement in order to evaluate the source impedance from the output port.

Nominally, the interface waveform is made up of pulses that switch between two levels, so it may be expected to measure the amplitude of the waveform as the difference between these levels; indeed, the most direct measurement uses this technique, expressing the amplitude in volts, peak-to-peak. In APWIN the interface waveform amplitude is described using this measurement.

The traces of Figure 134 and Figure 135 show that it is not as simple as that. The horizontal part of the waveform tends to droop towards zero volts as a result of DC blocking on the transmitter output. For reference, Figure 127 shows a typical AES3 transmitter circuit with two capacitors and a transformer that each have a DC blocking effect.

Useful measurements of the amplitude would ignore the droop, which is not relevant for maintaining data integrity but only serves to increase the peak-to-peak measurement value.

You can eliminate the contribution of the droop to the peak-to-peak amplitude by subtracting the magnitude of the droop from the peak-to-peak value. For example, in Figure 135 the peak-to-peak amplitude for the trace of the terminated signal (looking at the two broadest pulses) is 5.1 V with a droop of 0.5 V (on those pulses); the corrected amplitude would be 4.5 V.

However, please note that this type of measurement whereby the droop is subtracted is not standardized. It is useful for internal use but if it is quoted in a publication then the measurement calculation should be made clear.

Output port balance

Balance is a property only relevant to the twisted-pair version of the interface. The specification for balance in AES3 states, “...any common-mode component of the signal shall be more than 30 dB below the signal at frequencies from DC to 128 times the maximum frame rate...” The test conditions are not described.

This specification is not ideal, as it is a statement of signal symmetry rather than balance.

The purpose of output port balance

The reason for using the balanced twisted pair in an output circuit is to minimize crosstalk between cables run together. Since the cables are shielded (which minimizes electrostatic crosstalk), the crosstalk mechanism would be primarily through induction.

For balanced interconnects, inductive susceptibility is determined entirely by the *impedance balance*. In a balanced circuit there is an impedance between the two conductors, an impedance from each conductor to ground, and a common-mode impedance, which is the impedance to ground measured from the two conductors in parallel.

The impedance balance is determined by the ratio of the *mismatch* in the impedance to ground from each conductor, with respect to the common-mode impedance.

Balance can therefore be improved by either high-precision matching in the impedance of each leg, which is not easy to achieve; or, by making the common-mode impedance very high compared to the impedance between the conductors, which is quite achievable with the use of a transformer.

Inductive emissions in an output circuit are determined by asymmetry in the current carried by the two conductors. Current asymmetry is only directly linked to voltage asymmetry if both the source and destination have an alternate path for significant return currents—otherwise any current on one conductor has to return via the other.

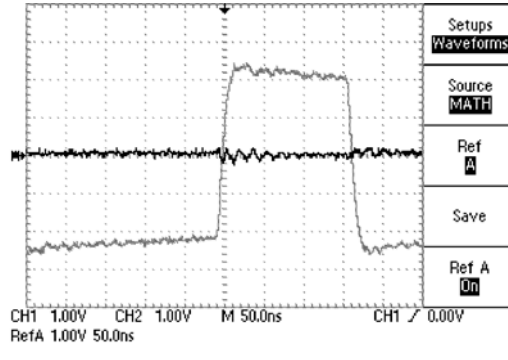
If the common-mode impedance is much higher than the differential (interconductor) impedance, then this alternate path will not carry significant currents.

If the common-mode impedance at the source and destination is low, then a significant alternate return current path is created. In that case, both voltage symmetry as well as the symmetry in the impedance of each leg become important for equalizing the current in each conductor and so minimizing crosstalk.

Measuring output port balance

For output ports without transformers the common-mode impedance may be low, and a measurement of common-mode voltage could be simply made by summing the voltages of the two output conductors with respect to a common-

Figure 136. Oscilloscope view of sum (black) and difference (gray) of balanced interface signal legs.



mode reference point, such as the cable shield or chassis, and dividing by two. This summing can be done quite easily in most two-channel oscilloscopes.

Where the output is coupled via transformers (as illustrated in Figure 127) the common-mode impedance will be high, and a measurement of common-mode voltage at the output terminals will be sensitive to the impedance balance of the connected measurement device. Any such measurement will have limited accuracy and a limited relationship to the impedance balance, which is the relevant factor for crosstalk.

It is recommended that a matching pair of high-impedance oscilloscope probes are used to make a measurement of the output port balance in the manner defined by AES3. This should minimize the effect of their load on the measurement. Figure 136 shows a measurement of the differential and common-mode parts of a signal made in the way. The traces show the sum and difference signals derived from the two input channels. The sum signal, shown in black, is twice the common-mode voltage. The difference signal, shown in gray , is the differential-mode voltage.

By inspection of this measurement we see that transitions in the differential signal are accompanied by high-frequency disturbances on the trace of the common-mode signal, shown as the black trace. The main spectral components of these disturbances have an amplitude of up to 0.4 V peak-to-peak (representing a common-mode amplitude of 0.2 V peak-to-peak) and a period of less than 20 ns (a frequency of greater than 50 MHz). This is close to 30 dB below the differential signal (at 4.5 V peak-to-peak) but the significant frequency components fall above the DC-to-128 *F*s (6 MHz) range in the AES specification, so this output passes the balance specification.

Another balance measurement for output ports

A useful balance measurement could be made that gives comparable results even for the transformer-coupled output case. The test can be modified from that specified in the standard so that it is made with a controlled differential and common-mode load to the output port. The 110 Ω nominal load impedance could be used with a center tap (between two 55 Ω resistors that make up the 110 Ω) connected via an 82.5 Ω resistor to ground. This does not reflect a typical load but it is a useful measure for the following reasons:

- The relatively low common-mode load impedance makes the measurement less sensitive to very high-impedance mismatches that are not significant.
- The common-mode and differential impedances are the same, so the ratio of common-mode to differential-mode signals is the same for voltage and for current.

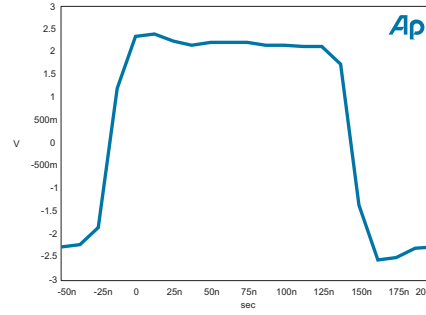
- The significance of the result is comparable for output ports with high or low common-mode source impedances, i.e., ports with or without transformers.

The accuracy of this measurement depends on the matching of the two 55 Ω resistances to much better than the balance ratio that is being measured. Apart from that matching, the precision of the resistors is not critical to the measurement accuracy and need be no better than 2%.

Transition times

The speed of the transitions on the interface can be measured using an oscilloscope. Digital audio test sets, such as System Two Cascade, can also be used. System Two Cascade has a sampling frequency of 80 MHz and a bandwidth of 30 MHz when using the INTERVU software. This is fast enough to get a reasonably accurate measurement for typical AES3 waveforms, and the result is illustrated in Figure 137.

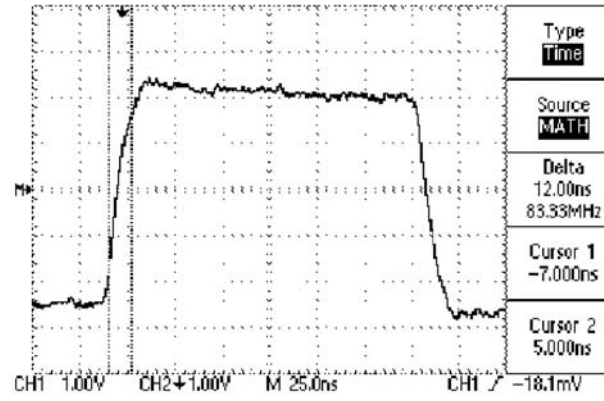
Figure 137. System Two Cascade view of the interface waveform at the output of a DAT machine.



The rise and fall transition times are defined as the time between the 10% and 90% amplitude points. In the case of this figure, with an amplitude of approximately 5 V, the 10% and 90% points are at about 0.5 V away from the low and high state values. The transition times on this trace appear to be between 15 ns and 20 ns. This is slightly faster than we can measure reliably with the Cascade.

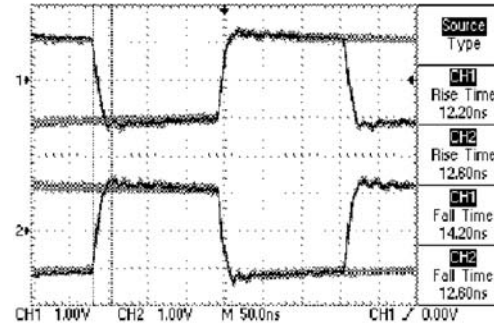
An oscilloscope trace is shown in Figure 138. The oscilloscope used for this trace uses digital sampling at a rate of 1 GS/s with a signal bandwidth of 60 MHz. The two channels of the oscilloscope are used together, with one trace displaying the differential signal (with channel two inverted and summed with channel one).

Figure 138. Oscilloscope view of the interface waveform at the output of a DAT machine. Channel two has been inverted and summed with channel one.



The cursors have been set manually at the 10% and 90% points of the rising transition and the time separation of the cursors (delta) is 12 ns.

Figure 139. Oscilloscope view of the interface waveform at the output of a DAT machine. The two oscilloscope channels are displayed independently.



This oscilloscope also gives direct readings of rise and fall time on the individual channels. Those are consistent with this result, but fluctuate with individual traces. This is shown in Figure 139.

Intrinsic Jitter

The theory behind jitter is explained in *Jitter Theory*, beginning page 3. There it is explained that the jitter on a digital interface output port is specified through two distinct measurements: the measurement of jitter produced by the device (the *intrinsic jitter*) and the conformance of the output signal with the *jitter transfer function* (which specifies the amount of jitter being passed through from an external synchronization source).

The intrinsic jitter of a device may well depend on the synchronization mode of the device. If selected as a clock master, the device may use an internal clock with one level of intrinsic jitter. If the device is selected as a clock slave and is locked to an external source, a different circuit will be used, and that circuit may have a different intrinsic jitter measurement. In addition, the

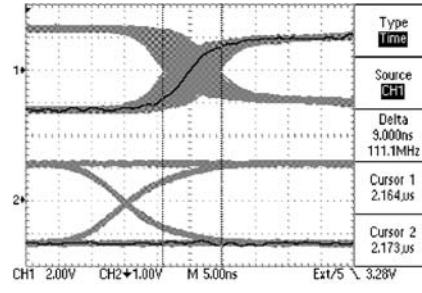
clock system may change jitter characteristics significantly at different sample rates.

The most basic measurement of jitter is by oscilloscope. This is only possible if the oscilloscope is triggered from a known low-jitter clock that is synchronous to the frame rate of the output under test.

An example of this measurement is shown in Figure 140. This was made using the **TRANSMIT FRAME SYNC** output of System Two Cascade to trigger the oscilloscope. The signal on the output port of the DUT is shown on the top trace. The persistence of the oscilloscope has been set to "infinite" and data has been collected for a few seconds to capture the range of timing deviations in that period. The cursors are aligned to show that the range of movement of the zero crossing point is 9 ns.

The signal on the input port of the DUT is shown on the lower trace and is broadened to a width of about 1.5 ns. This indicates the residual jitter in the measurement, since the input port is driven from the digital output of the System Two Cascade with the jitter generator set to **OFF**. This residual jitter may be due to jitter in the signal generator or in the oscilloscope. (Because of timing offsets between input and output, the lower trace shows the transition in the middle of the bit cells—which is sometimes not present—while the upper trace shows the transition between two bit cells—which is always present.)

Figure 140. Oscilloscope view of interface jitter using an external frame sync trigger.



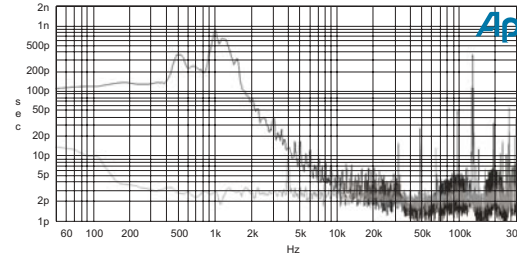
Taking into account this residual jitter, we can conclude that the output jitter of the DUT is 9±1.5 ns peak-to-peak. This simple and direct measurement is useful indicator of the jitter level, but has disadvantages:

- The measurement can only be made when the output port timing is slaved from a known low-jitter reference.
- Low-frequency and very-low-frequency jitter components will have exactly the same weighting on the jitter measurement result as the higher-frequency jitter components will have.

- It is not possible to capture every single transition in a sequence. This could cause an insensitivity in the measurement to jitter at some frequencies.
- The deviation of the transitions from the mean is not clear. If the jitter is an asymmetric, then the peak deviation from the mean will be not be simply half the peak-to-peak deviation. We need to evaluate the maximum excursion of timing deviation from the mean, as that is what relates to interface error mechanisms.

The AES3 intrinsic jitter specification (also in IEC60958-3 and IEC60958-4) is written for a different measurement method. This specification uses a jitter meter that compares the timing of the input transitions with a clock derived from the same signal, but with a defined jitter attenuation characteristic. This combination has the effect of producing a measurement that can meet the defined high-pass characteristic with a 3 dB corner frequency at 700 Hz.

Figure 141. Spectrum of jitter from interface signal shown in Figure 140.



These jitter meters are becoming available on digital audio test equipment and are provided in the Audio Precision range of instruments.

The same signal that is shown in Figure 140 produces a jitter measurement of 3.3 ns peak with the APWIN meter set to a frequency range of 700 Hz to 100 kHz. These two results are consistent, given that a peak-to-peak reading can be up to twice the peak reading and the lower-frequency limit to the first result is much lower.

Figure 141 illustrates the spectrum of this jitter gathered using the INTERVU package of APWIN. This test is stored as “Intrinsic jitter spectrum.at2c.”

The graph shows that the jitter spectrum has a significant peak around 1.2 kHz. This may be indicative of the corner frequency of the clock recovery phase-locked loop (PLL) in the DUT. A high peak such as this can be a consequence of inadequate damping of the PLL. The jitter transfer function plot may be able to confirm this.

This spectrum has not been corrected to show jitter spectral density. The correction factor required to convert the vertical

axis units to show jitter spectral density is approximately 1/10. (This is based on the formula from TECHNOTE 24: A-to-D Converter Measurements*, page 20, but modified with the INTERVU data capture duration of 19.6608 ms replacing FFTPoints/ SamplingFrequency.)*

Jitter transfer function

The jitter transfer function is measured using an interface signal with a deliberate and controlled level of sinusoidal jitter. The frequency of this jitter is swept over the range of interest, and the jitter level at the output of the DUT is measured.

This measurement can use the oscilloscope method (described above) to measure the output jitter, but only if a suitable trigger clock is available *without* the stimulus jitter. (For example, with System Two Cascade the **TRANSMIT FRAME SYNC** output signal can be used as a jitter-free trigger. For that use, clear the **Jitter Clock Outputs** checkbox on the APWIN Sync/Ref Input/Output panel).

In most circumstances System Two Cascade can also perform the complete measurement. However, the oscilloscope method may be preferred for measurements below the low-frequency corner frequency of the jitter measurement meter.

Figure 142. Jitter transfer function of an AES3 transmitter-receiver evaluation board as measured by System Two Cascade.

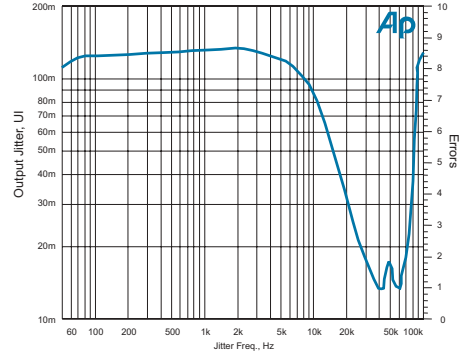


Figure 142 shows a measurement using System Two Cascade. The DUT is an evaluation board for an AES3 receiver and transmitter. The test file is “JTF eval board.at2c.”

The measurement was made with a sinusoidal jitter input of 0.25 UI peak-to-peak (0.125 UI peak). This amplitude was selected as the highest amplitude of jitter that the jitter tolerance specification of AES3 requires for receivers to be able to decode at all frequencies. See **Receiver Jitter Tolerance**, page 176. For equipment that does not meet this tolerance level the applied jitter amplitude may need to be reduced.

The measured level of output jitter is shown in terms of peak jitter, rather than peak-to-peak, so a reading of 125 mUI corresponds to the same level as the applied jitter level. The trace shows a slight peak at around 2 kHz, followed by attenuation that is at –3 dB at around 10 kHz. This then falls to a reading of around 13 mUI at 40 kHz.

Above the 40 kHz point, the measurement rises again to a small peak of 17 mUI at around 48 kHz. Above that frequency, the response is a mirror image of the response below 48 kHz. This is indicative of aliasing at the sub-frame rate of 96 kHz. This occurs if the phase detector in the clock recovery system uses interface transitions in the preamble, not in the modulated part of the data stream. That jitter is effectively being sampled at a rate of 96 kHz, so jitter above half that rate (48 kHz) becomes equivalent to jitter below half that rate. This hypothesis was confirmed by setting the input jitter frequency to 95.999 kHz, which is just 1 Hz below the 96 kHz sub-frame rate, and observing on an oscilloscope that the output jitter was a slowly-moving 1 Hz.

When performing this observation on an oscilloscope, it is important to trigger the oscilloscope so that it is not just presenting transitions at the sub-frame or frame rate. If that were the case then the oscilloscope trigger is performing the same aliasing as we are trying to observe, and jitter occurring close to those rates will appear to be at a low frequency. To avoid this problem, trigger the oscilloscope using a jitter-free reference clock running at a higher rate. The **MASTER CLOCK OUTPUT** on the back of the System Two Cascade provides a clock that has a period of 0.5 UI. This can be used to observe jitter in interface transitions. The oscilloscope trigger hold-off can be adjusted until the transitions are observed to be 1 UI apart; however, this is not essential in making the observation if the jitter amplitude is significantly less than the trigger signal period.

The small peak at 48 kHz in the measurement is another aliasing effect, and is probably indicative of a non-linearity in the phase detector that is causing a modulation of the incoming jitter at the frame rate. This was confirmed by setting the jitter frequency to 1 Hz below 48 kHz. A major component of the jitter was then seen to be at 1 Hz.

The measured jitter transfer function conforms to the AES3 specification for jitter gain, as is detailed in the **Jitter Theory** chapter. That specification requires that for any frequency there should not be more than 2 dB of jitter amplification, measured from input to output. The key point to look for is the amount of "jitter peaking." In the case of the measurement shown in Figure 142, the peak at 2 kHz is at 133 mUI, representing a gain of 0.54 dB over the input level of 125 mUI (both measurements peak, rather than peak-to-peak). This is well within the specification.

The overall measurement indicates that this circuit does not provide significant levels of jitter attenuation, such as the 6 dB at 1 kHz described in the op-

tional AES3 jitter attenuation specification. This is normal for a device that uses the same clock to both perform data recovery and provide the output clock. This is a single-PLL configuration of a receiver/transmitter system. In a dual-PLL system, further jitter attenuation can be provided in the second PLL, which is not used to decode the incoming data stream. More information on this topic can be found in the **Jitter Theory** chapter.

Input Port Characterization

Measurements on an input port are divided into two forms:

- direct measurements of input port impedance characteristics;
- determination of the decoding capabilities of the input when presented with deliberately degraded signals.

For verification of decoding ability a known signal can be applied, and the output can be compared to this signal and examined for degradation. If the output of the DUT can be configured so that it transmits the decoded data unmodified, then a pseudo-random sequence can be used as a source. The output sequence can be checked against the input sequence to show that the data has not been corrupted. See the discussion of Bittest in **Data transparency**, page 184.

If the only output that can be monitored is analog, or if the output is digital but has been subjected to some processing which produces an output that is not a bit-exact copy, then the output data cannot simply be verified to be exactly the same as the input.

In this case, an alternative measurement can be made. A high-frequency sine wave is applied as the input signal. A THD+N test can then used to determine the point of receiver failure, as the sine signal will either disappear, or will become distorted with decoding errors. Listening to the THD+N residual after the notch filter can provide a convenient audible indicator of receiver failure.

Input port impedance

For the AES3 specification, the input port should have an essentially resistive impedance of 110±22 Ω from 100 kHz up to 128 times the maximum frame rate that it supports. This can be measured precisely, but for most purposes an oscilloscope can make an evaluation in a similar manner to that for the output port impedance. A two-channel oscilloscope is used in differential mode with one channel subtracted from the other. High impedance scope probes are also required in order to avoid loading the circuit.

The technique for evaluating the impedance is similar for the unbalanced and balanced formats. In the balanced format specification, the termination im-

pedance refers to the differential mode impedance between the two signal lines.

A reference output port that has a reliable impedance is used to drive the input port under test. An oscilloscope can be used to observe the voltage waveform on the cable at the input port, and then this can be compared with the waveform viewed on the cable when the input port has been removed and replaced with a resistor of the correct impedance.

System Two Cascade provided the source for the oscilloscope traces of Figure 143. The traces are of the difference between the two oscilloscope channels, and the scales are 1 V/div and 100 ns/div. The reference trace with the 110 Ω 1% resistor is shown in gray. The measurement trace, shown in black, is a close match to the reference.

There is some overshoot after the transition. The overshoot indicates that for the highest-frequency signal components the impedance may be slightly higher than the reference. The small amount of extra droop indicates that the low-frequency input impedance may be slightly lower than the reference. The maximum voltage difference caused by the overshoot is around 0.2 V, or 8% of the signal voltage at that point, and it lasts for about 30 ns. The difference in droop is less than 0.1 V over the 480 ns (3 UI) of the first pulse of the preamble. The overshoot and droop effects, then, are very small compared with the 20% tolerance of the AES3 input impedance specification. Both of these effects could be a consequence of the limited bandwidth of the transformer on the input port being tested.

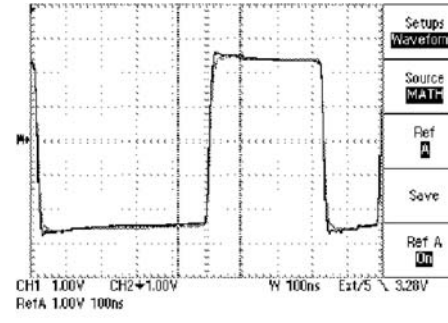


Figure 143. Evaluating AES3 input receiver input port impedance using an oscilloscope.

The trace of Figure 144 shows the same test on the input port of another device. In this case the stimulus interface waveform carries no embedded information—no audio, no user data or channel status bits—so the whole waveform is stable. The time axis has been extended to the left to include the 3 bits (U, C and P) preceding the preamble and the two 3 UI pulses in the preamble.

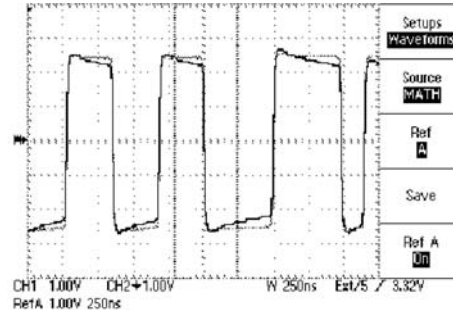


Figure 144. This is the same evaluation shown in Figure 143, performed on the input port of a different device.

The droop in this case is more significant. It appears to be 0.4 V over the 3 UI preamble pulse, and this may indicate a problem with the impedance match at low frequencies. Taking into account the droop—which changes the starting voltage of the transitions—the amplitude step of the transitions is also reduced by about 0.25 V, or 5%. The shape of the curve after the transition does not show a significant overshoot, so we are not seeing as much impedance change for the higher frequencies in the signal as we saw in Figure 143.

In conclusion, the impedance is not a good match, but is likely to be within the 20% tolerance of AES3. If in doubt, then use a dedicated impedance analyzer to make a more precise measurement.

Maximum input amplitude

Input receivers may fail with input voltage amplitudes above a characteristic maximum. This maximum level could be determined by trial and error.

Alternatively, applying a signal at a specified maximum level and confirming correct operation can show conformance with that performance specification.

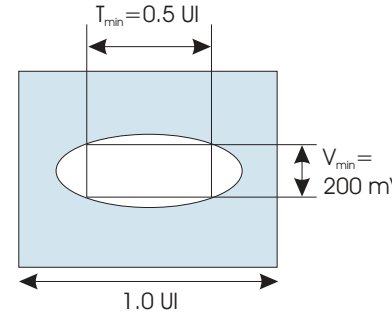
Minimum input signal amplitude and the eye diagram

The minimum input signal level is defined in relation to an eye diagram. This defines a minimum height, and a minimum duration of that minimum pulse height for a signal that should be correctly decoded.

Figure 145 shows the eye diagram used for AES3-1992[1] and for IEC60958-3:1999[3] and IEC60958-4:1999[4]. The box inside the eye, which is 200 mV by 0.5 UI, defines the minimum signal that the receiver should be able to correctly decode.

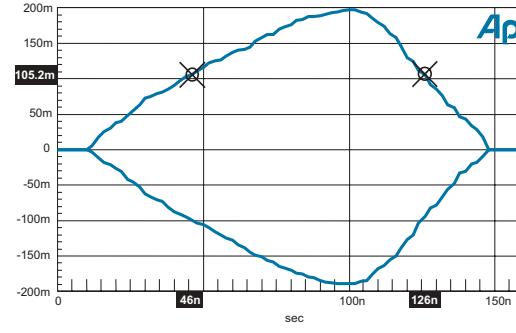
In practice, different receiver designs use different decoding techniques, and as the signal eye size falls there can be many reasons that a receiver might fail. In a given receiver, then, some signals may be more difficult to decode than others, even those that display smaller “eyes.”

Figure 145. Eye diagram used for AES3-1992, IEC60958-3:1999 and IEC60958-4:1999



Even so, the verification that a receiver can decode signals with eye sizes at least as small as the AES/IEC minimum remains useful.

Figure 146. APWIN view of an eye pattern.



Receiver performance can be evaluated by impairing or degrading the interface signal while monitoring the receiver and the eye pattern. Three approaches using degraded signals follow:

Lengthening the rise and fall times

System Two Cascade can produce a degraded digital interface signal with configurable rise/fall time and amplitude. It is possible to degrade the signal to one that has an eye that equals these minimum limits. This can almost be achieved for a 48 kHz frame rate AES3 signal by setting the amplitude to 740 mV and the rise/fall time to 200 ns. The oscilloscope trace of such a signal—modulated with a pseudo-random sequence—is shown in Figure 147. See the discussion of Bittest in **Data transparency**, page 184.

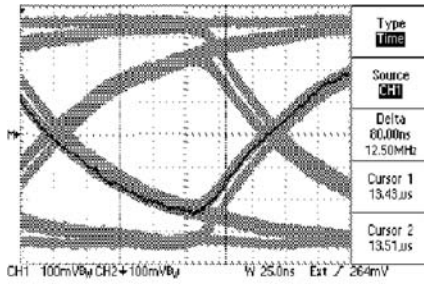


Figure 147. Oscilloscope view of an eye pattern.

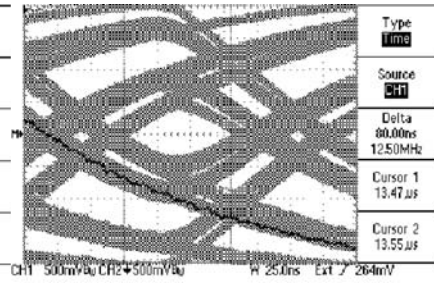


Figure 148. Oscilloscope view of an eye pattern, showing a smaller "opening."

For this figure the **MASTER CLOCK OUTPUT** on the System Two Cascade rear panel provided the oscilloscope trigger, with the oscilloscope hold-off adjusted to align with the data transitions. Many traces are shown on top of each other through the use of several seconds of display persistence.

The eye diagram can also be directly measured with System Two Cascade, shown in Figure 146. (See test "DIF eye.at2c" as well.)

Adding noise and jitter to the signal

It is also possible to modify the eye height and width by adding noise and jitter to the signal, which can be accomplished in System Two Cascade on the DIO panel. You can easily experiment with different degrees and combinations of jitter and noise when investigating receiver performance.

Using the cable simulation

Another approach is to degrade the interface signal with the same effect as that of a long cable between the signal source and the receiver. Of course, you can always do this with a long cable, but a lower cost and less bulky approach is to use a circuit that approximates to a simulation of a long cable, such as System Two Cascade's cable simulation function on the DIO panel. The oscilloscope trace of the signal is shown in Figure 148, with the 0.5 UI eye width specification marked by cursors. (Also see "DIF cable sim eye.at2c.")

The System Two Cascade cable simulation presents a signal that does not meet the eye diagram specified by the standards. Despite this, many receivers can decode it, and it forms a useful benchmark of what a good receiver is capable of. Some AES3 receivers may have an optional equalization to facilitate the use of very long cable lengths, and the cable simulation will help with assessing these receivers.

The disadvantage with the simulation approach is that it can only give a pass or fail result. The difference between threshold of errors and total failure

is quite small, so the error rate does not indicate the margin by which the receiver has failed the test.

Common-mode rejection

The common-mode rejection specification for the balanced AES3 format requires that a receiver should remain functional even with a common-mode signal of up to 7 V peak at frequencies from DC to 20 kHz. For testing against this specification, this common-mode signal can be imposed using a center-tapped transformer on the interface signal generator output. Some digital audio test equipment can generate this common-mode component, including System Two Cascade.

This specification is not sensitive to the impedance balance of the input port. Any imbalance can result in the production of common-mode currents and introduces a mode conversion mechanism whereby induced common-mode signals produce differential voltages.

There is not a direct specification for this performance aspect in AES3. The use of transformers (as mentioned in AES3) should ensure it is not an issue; without transformers, however, this crosstalk mechanism may be significant—particularly in conjunction with long cable runs with many different signals bundled together.

Receiver jitter tolerance

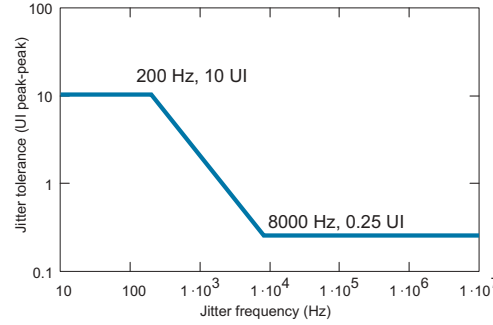
The digital audio interface input port will have some ability to decode signals correctly, even in the presence of jitter. The jitter tolerance is a measure of how much jitter can be present before the receiver fails.

At high jitter frequencies the tolerance to jitter is fixed, but below a characteristic frequency the tolerance increases. Near this characteristic frequency increase—the jitter tolerance corner frequency—the jitter tolerance may have a minimum. This is explained in the chapter **Jitter Theory**[13] and in *AES Preprint 3705*[14].

The specifications for receiver jitter tolerance in professional (AES3) and consumer (IEC60958-3) applications differ. The professional specification requires the jitter tolerance corner frequency to be around 8 kHz or above. This is in order to provide a rugged system, performing well even when many devices are connected in a chain and low-frequency jitter has built up.

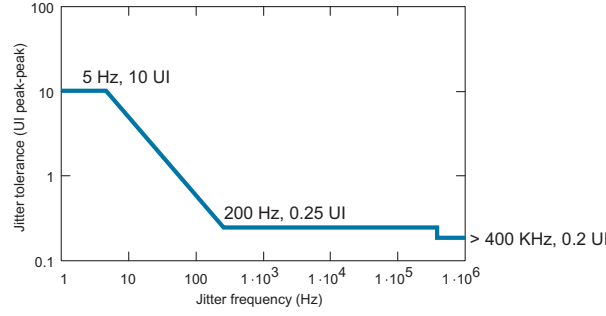
The professional template is shown in Figure 149. Above the corner at 8 kHz is a frequency-independent high- frequency shelf with a tolerance of 0.25 UI peak-to-peak. Below 8 kHz there is a 6 dB per octave slope where the tolerance rises to 10 UI peak-to-peak at 200 Hz.

Figure 149. Jitter tolerance template for the professional interface.



The consumer specification is written to allow lower-cost clock-recovery systems and is more relaxed in comparison to the professional format, as can be seen in Figure 150.

Figure 150. Jitter tolerance template for the consumer interface.



The jitter tolerance corner frequency is lowered to around 200 Hz. This allows a single-stage clock recovery system to be used, which can also provide jitter attenuation above 200 Hz. This would be useful for generating a sample clock used by a digital-to-analog converter where sidebands due to jitter much above 200 Hz are increasingly likely to be audible.

In both templates the maximum jitter tolerance is set at 10 UI. This is primarily to simplify the task of generating the test signal. In receivers the tolerance will continue to increase as the jitter frequency falls.

The consumer template also has a curious step at 400 kHz. Above this frequency the required tolerance level is reduced slightly, but apart from that the flat high-frequency part of the template is at the same level as for the professional format.

Any receiver that meets the professional tolerance specification will meet the consumer specification.

Measuring receiver jitter tolerance

Testing the jitter tolerance of an input port is similar to some of the other tests in this section. An appropriate signal from the DUT is monitored in a way

that will reveal when errors start to occur at the input port receiver. Meanwhile, sinusoidal jitter of variable frequency and level is applied to the input port.

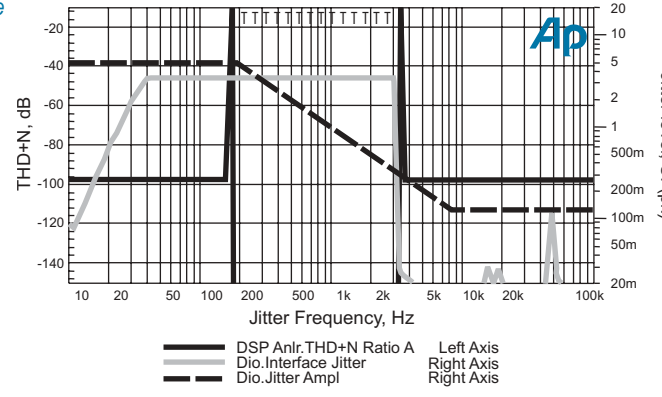
At each test frequency the level of jitter is increased until errors are detected. The highest level before errors occur at each frequency defines the jitter tolerance at that frequency. This technique provides a measurement of the actual jitter tolerance of the input port.

Measuring conformance to the specification

To merely verify conformance with the specification, it is simpler to just use the values of jitter level and frequency from the template and verify correct receiver operation.

System Two Cascade can test against the jitter tolerance template curves automatically by attaching a template data file to the jitter generator in **Jitter Generation: EQ Sine** mode. Included with this Application Note is the test "DIF jitter tolerance.at2c" (based on a test supplied by Audio Precision) that does this. Two template data files are supplied: "IEC60958-3jittertolerance.adq" and "AES3jittertolerance.adq." One of these two files (the consumer or the professional template) needs to be selected as the **EQ Curve** in the **Jitter Generation** section of the DIO panel. The template EQ curves provide the correct amplitude when the jitter amplitude selected on the DIO panel is **1 UI,** and **EQ Sine** is selected as the type of jitter. The test monitors the distortion on a sine wave to identify when receiver errors begin to occur.

Figure 151. Device THD+N plotted against jitter.



The traces in Figure 151 are taken from the results stored in this test (DIF jitter tolerance.at2c). The failure of the input port is determined by the THD+N reading on the signal coming back from the DUT, which is plotted against the left-hand scale in black. The device under test is a professional DAT recorder

with a normal THD+N reading of 97.5 dB (measured digital-to-digital, in "input monitor" mode). For jitter frequencies of 160 Hz to 3.6 kHz the THD+N reading is off the scale, since the receiver is unable to lock to the signal. (The repeated "T" characters across the top of the graph for this range show that the THD+N measurement was timed-out since the reading was unstable).

The other two curves show the applied jitter level as a dotted line, and the measured jitter level as received on the signal coming back from the output port of the DUT as a gray line. The scale is in UI but shows peak, rather than peak-to-peak readings, so the values of applied jitter are half those shown on the template. (The jitter measurement is saturated at levels above 3.4 UI. The graph also shows this reading when the DUT is unlocked.)

Signal Characterization

Sometimes there is a requirement to measure a signal "in situ." This could be as part of diagnosis of a system problem; for example, the occurrence of occasional data errors on a device that when tested conforms to specifications.

Many of the characterization methods have been mentioned already in connection with measuring output ports. The following is a review with specific application to measuring a signal in isolation.

Signal Amplitude

The peak amplitude is a characteristic that is easy to measure, but it is not normally a direct indicator of signal quality. If the peak signal level is very much lower than the specified level due to cable losses, then it is likely that the pulse distortion due to the frequency-dependent nature of those losses will have caused an even more significant reduction in the eye opening.

In any case, an estimate of the size of the eye opening should be made; this can be compared with the eye diagram associated with the input port minimum input signal amplitude.

Signal Interface jitter

The measurement method of output port intrinsic jitter can be applied to measuring jitter on the interface signal at any point. The low-frequency rolloff setting of 700 Hz used for the intrinsic jitter specification can also be used in this case. A jitter spectrum can also be used to look for any specific problems (see the test "Intrinsic jitter spectrum.at2c," referred to on page 168). If there is a large jitter peak this may correspond to a poorly-damped PLL earlier in the system.

It may be possible to use a particular test signal to modulate the interface signal being examined. In this case the J-test signal described on page 32 is

useful as a method of presenting a worst-case form of data-induced jitter onto the line.

Signal symmetry and DC offset

The digital audio interface signal should not have any significant DC component. This can be measured using a high-impedance voltmeter. If the presence of the AC components of the interface signal confuses the meter, a passive RC low-pass filter can precede it.

For the balanced (AES3) interface it is possible for one leg of the interface to become an open circuit without causing a failure. This is because the signal return can be completed through other paths; for example, by capacitance to the cable shield.

In some circumstances this fault condition can apparently improve signals. The capacitive coupling that bypasses the open circuit will couple higher-frequency better than low-frequency components. This can produce some degree of equalization when a signal has had considerable attenuation of higher-frequency components through cable losses.

Signal reflections

An oscilloscope can be used to look for signs of signal reflections. These are produced by impedance discontinuities in the transmission line formed by the interconnect cable and connectors. Discontinuities occur if:

- Either termination is incorrect or not fitted.
- There is a stub of cable—terminated or not—connected to some midpoint of the cable. This may occur at a patch bay in a studio.
- The use of a BNC T-fitting to parallel two inputs. (This can be done, but only if the terminations that are not at the end of the cable are switched off and the T-fitting is mounted directly on the non-terminated inputs. However, it is not normal for terminations to be switchable at all.)
- There is a length of cable with incorrect impedance. It is quite easy to accidentally use 50 Ω coaxial cable (coax) instead of 75 Ω by mistake. With the balanced interface some types of twisted pair cable may produce a serious mismatch. In particular star-quad cable has a very low transmission line impedance (not to be confused with resistance) and will cause bad reflections if mixed with simple twisted pair cable for an AES3 connection.

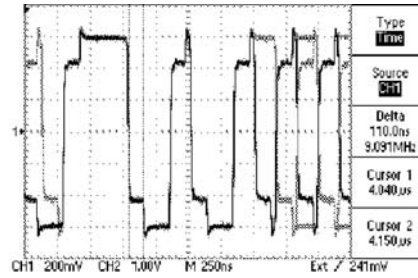


Figure 152. Oscilloscope view of reflections in an improperly terminated AES3 interface line.

An example of an oscilloscope view of a signal with reflections is shown in Figure 152. This is created by a length of 75 Ω coax with a 110 Ω resistor forming an incorrectly valued termination.

The cursors highlight the delay between the transition after the first 3 UI pulse in the preamble and the reflected pulse due to a mis-termination. The delay is 110 ns, which corresponds to the path of the pulse to the mis-termination and back, at the propagation velocity down the cable.

The reflected signal has the same polarity as the non-reflected signal, so it adds to the amplitude. This is because the cable is terminated in a higher resistance than the cable impedance. If the discontinuity is a impedance reduction, then the reflection would have opposite an polarity and would subtract from the amplitude.

In this example the discontinuity is not enough to cause the eye height to be reduced significantly, so it should not affect the decoding of the signal.

SYNCHRONIZATION

The embedded clock defined by the interface bit-cell transitions, the subframe and the frame boundaries can be used as a timing reference by equipment to derive timing for converters, processors, and digital outputs. For digital outputs AES11 defines limits for the timing offset between the frames of the reference input signal and the frames of the outputs.

In some cases the timing reference is provided by another signal or clock, and the incoming signal needs to have been already synchronized to that clock. AES11 defines a specification for this sort of synchronization. It also covers synchronization of the digital audio interface with video.

Synchronization by embedded clock

The simplest form of synchronization of a single device is when there is only one input signal and that signal is used as the timing reference. This is often not thought of as synchronization because it is implicit to the operation of such equipment, such as outboard stereo DACs or stereo digital recorders (DAT, DCC, or CD-Audio).

If this input also has an output associated with it, then according to AES11 that output should be aligned so that the time difference between frame starts at input and out-

put is less than 5% of a frame period, or 6.4 UI. This is shown in Figure 153, with the circle representing the possible phases of relative input and output frame timing. In this picture the distance around the perimeter of the circle corresponds to one frame.

DARS

The digital audio reference signal, or DARS, is an AES3 signal that is used for timing purposes rather than for carrying audio data. This signal can be fed from the clock master device to other devices—which would be synchronization slave devices—which need to be synchronized to each other or to the clock master. For example, the clock master may be a digital mixer. The slaves may be the various source devices that feed the inputs of the mixer, such as tape and hard disk recorders and outboard analog-to-digital converters.

These slave devices also need to meet the AES11 output tolerance alignment specification of ±5% of a frame. As a result, the signals from the slave devices are appropriately aligned to the internal timing of the mixer so that there is no ambiguity about which frames are associated with the same sample time.

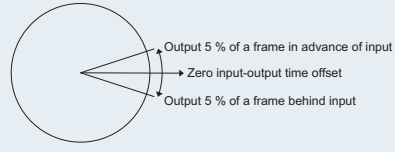


Figure 153. Synchronization to the input signal.

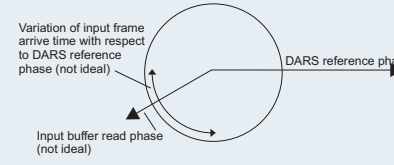


Figure 154. Example of input timing ambiguity with a DARS reference.

Input data alignment

Consider the case where an input signal is used, but it is not the synchronization source. For example, we might have a digital mixer using a DARS reference from a master synchronization clock, and several other input signals that need to be timed together. The data from the input signals need to be processed in synchronization with each other and with the DARS signal so that sample data corresponding to the same sample time are processed together.

> *In this situation it is assumed that the input signals are all at the same sample rate and have been synchronized by a DARS. If any are at slightly different rates then a re-synchronizing sample rate converter would be required.*

The timing of the arrival of the data frames from each input signal will determine which frames are aligned together when processed. If the timing is closely matched there is no ambiguity, but if one of the input signals is slightly misaligned that produces a problem.

For this example, consider that the data from each input signal is received and decoded and briefly held in a buffer store. At a time determined by the mixer's own clock (which is derived from the DARS) this buffer store is transferred to another store, or "read." This defines the boundary between times when an input data word corresponds with one sample or the next.

An ambiguity in frame alignment can occur if the input signal arrives just at the time when the mixer is reading the input data buffer. If the new frame data for input sample N has been decoded and then loaded into the mixer input data buffer just before mixer sample M is read, then the input sample number and mixer sample number are the same. However, if input sample N arrives a few microseconds later, then input sample N–1 is used for mixer sample M. This produces a time error of one sample for that input.

Even worse is the situation where the input sample arrives so close to the moment that the input buffer is read, that a small amount of jitter causes a fluctuation of states between a delay of one sample, and no delay at all. This is shown in Figure 154. This could result in the missing and repeating of input samples each time the data arrival phase crossed the buffer "read" phase.

The AES11 rules address this problem with the combination of input and output alignment tolerance. The output tolerance has already been mentioned. The input tolerance requires that the receiver should correctly process an input that has arrived with a timing that is within 25% of a frame period to the timing of the reference. This range is shown in Figure 155.

A receiver that needs to support the DARS synchronization mode should be designed with the input buffer read time opposite in phase to the ideal phase alignment determined by the timing of the DARS.

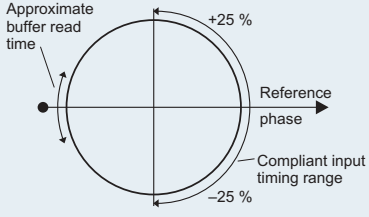


Figure 155. AES11 input alignment tolerance.

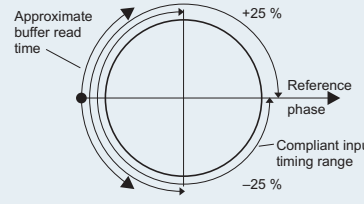


Figure 156. Provision of input buffer hysteresis to improve response to "wander."

AES11 requires that a receiver should treat synchronized input data as being sampled at the same instant, if the frame start is aligned to the DARS frame start with an error of less than 25% of a frame period. This timing offset tolerance allows for a chain of devices that are synchronized using the signal embedded clock (rather than a DARS) and therefore adding up to 5% of a frame of error for each device, and also for other timing errors.

A good receiver design can go further than this. It could use hysteresis in the region of non-compliant input timing and take away the risk of any particular timing relationship resulting in the dropping and repeating of samples. The ±25% rule mentioned above allows for hysteresis in the other 50% of the timing circle. This could be implemented to ensure that if the relative input alignment drifts past the critical phase, a sample of input data is not lost or repeated until the timing is up to 75% of a frame away from the nominal ideal, as is illustrated in Figure 156. If that occurs and the alignment drifts in the other direction, then the correction in the other direction would not occur until the error had reduced down to 25% from the nominally ideal timing. This will then give a tolerance-to-timing wander of as much as 50% of a sample frame, even if the source has a worst-case misalignment of 180 degrees to the correct (reference) phase.

Determining data handling characteristics

The ability of the digital audio interface ports to conform to the relevant standards does not ensure that the equipment interfaces will behave as expected.

It is possible, for example, for the interfaces to truncate part of the audio data word, or to require a particular channel status pattern before they can decode the audio data. Characteristics like these may be a consequence of the internal data word size, for example; or of a method of selecting an operating frequency by reading the channel status indication of the interface signal sampling frequency.

In such cases the interface ports can be functioning properly, yet there is a failure to decode the audio.

Audio data

The measurement of the performance of audio processing is beyond the scope of this Application Note. However—short of that—there are some important tests to assess how the digital interfaces are processing the data.

Data transparency

Many devices, such as digital recorders, routing devices and format converters are totally transparent to the audio data; that is, the audio data is passed through as a perfect bit-for-bit image. In some modes other equipment, such as digital mixers or outboard processing boxes, can also be operated in a data-transparent manner.

For these examples a pseudo-random data test signal can be very useful. This type of data stream can follow a defined sequence of bits over an extended period of time. The stream can then be recognized (even if time-shifted in a recorder) and the transparency of the image evaluated on a bit-for-bit basis.

The digital generator in APWIN can produce such a stream, which is called *Bittest*. Bittest can be selected as a generator special waveform by choosing **Wfm: Special: Bittest Random** on the Digital Generator panel.

The Bittest pattern can be recognized by choosing **Analyzer: Digital Data Analyzer (Bittest)** and **Waveform: Random** on the Digital Analyzer panel. The word-length to be tested should be selected on the DIO panel under **Input: Resolution**.

If any change in the data, including dither, has been applied to the signal, then the pseudo-random technique will not work. In that case the equipment is

not data-transparent and other measures of signal accuracy are appropriate, such as measuring the THD+N of the embedded audio.

Channel Status

The earlier description of the channel status bit function (page 153), and the annex on channel status that defines all the bit states (page 190), indicate how channel status can be used. Some of the more basic functions can be tested in a straightforward manner.

Status transparency

For equipment with any kind of pass-through function for the channel status information, it is useful to identify which data is actually being passed through. This can be discovered by sending various channel status patterns to the equipment and recording the pattern that is returned.

For the AES3 channel status data there is a check code, the Cyclic Redundancy Check Code (CRCC). If AES3 channel status data is modified in any way, then this code needs to be regenerated. If all the channel status remains the same, then the code does not need to be changed.

There is some information in the CRCC code. If an incoming channel status pattern has a CRCC error, that indicates the channel status is unreliable. The special case of CRCC=0 may indicate that CRCC is not implemented. If it is consistently zero, then it may make sense for the equipment to ignore the CRCC error.

Apart from the zero case, there are two methods of handling CRCC errors.

- The first is to ignore the new block and repeat the old channel status block.
- The second is to force a CRCC error in the reconstructed channel status block. This method is more appropriate if there is real-time data present in the channel status such as sample address code, since it does not require a delay of 192 samples to determine if the CRCC is correct before the whole channel status block can be transmitted. (As maintaining time-alignment of channel status and audio may be important, this would also involve an equivalent delay for the audio).

Using APWIN and the status page of the DIO panel it is possible to force incorrect channel status and observe the result on the output.

Actions in response to channel status information

Quite apart from the passing through of channel status data, there are several actions that can be triggered based on the channel status bit pattern. These can all be investigated by manual adjustment, and include:

- **Muting of non-audio data.** Setting bit 1 in the channel status stream should cause a mute of any DAC. If the audio signal is passed through digitally it should also be muted, unless the non-audio flag is carried with it.
- **Deemphasis.** For both professional and consumer formats there are emphasis flags. These should enable deemphasis filters on any signals, such as analog, that do not carry through the deemphasis flag information. In a format converter, the (rarely used) J17 emphasis flag in the AES3 stream cannot be converted to a consumer format equivalent. The DUT's reaction to the presence of this flag can be noted.
- **Sample rate selection.** Many devices require the sample rate to be indicated in order to function correctly. Some devices do not operate when the "non-indicated" state is used. This behavior can be noted.
- **Word length manipulation.** Where the audio word length is being reduced, then dither may be applied before the truncation in order to avoid signal-correlated errors in the resultant signal. Note whether or not the dither is disabled if the word-length indication shows that the word length does not require truncation.
- **Auxiliary audio masking.** The bottom 4 bits in the 24-bit audio word might (rarely) be used to carry other data. If that is the case it is indicated in the channel status that the maximum audio word length is 20 bits. Note whether or not the DUT masks off the lower bits in this condition. (See the chapter **Digital-to-Analog Converter Measurements**[16] and "d-a aux truncation test.apb.")
- **Copy inhibit.** In the consumer mode there are various combinations of the copyright bit and category code that should control if a device can record the signal or not. If the device is a recorder and is intended to support copyright rules then this can be verified. See **Serial Copy Management System,** below.

Channel status pattern generation

When an output device is required to generate a channel status pattern there may be a need to verify that this pattern is correct. This verification can be performed by checking that the channel status field indication correctly indicates the state of the channel. This can be achieved by comparing the generated status pattern with the field interpretations shown in the standard. See the **Channel Status Annex,** page 190.

For digital audio equipment that can have various operating modes, this status pattern needs to be verified in each mode. For a replay device or a format converter, the mode may be determined by the input data. For example, the following conditions can affect the generated channel status pattern:

- **Selected sampling frequency (internal sync).** Note if non-standard rates supported, how they are indicated, and how this affects the receivers the equipment is to be used with.
- **Synchronization sampling frequency (external sync).** Note what happens to the rate indication when the synchronization sampling frequency deviates from a standard rate.
- **Pre-emphasis selection.**
- **Copyright status.** This is subject to the serial copy management system and affects the consumer format bits 2 and 15.
- The channel is carrying linear PCM audio, data compressed audio or even a data stream that does not represent audio.
- Monophonic or stereophonic program material, or two independent channels.
- **Data word length.**

In addition, where there is more than one correct channel status indication that could be used, it may be useful to determine if it is the most appropriate or informative. This last determination needs to consider the application, the likely capability of receiving devices, and, where copyright control is involved, the preferred response of the receiver device.

The professional standard, AES3, defines three levels of support that a device may have. These are called channel status implementation levels.

- **Standard**. This requires that the first three bytes and the cyclic redundancy check code (CRCC) in byte 23 are correctly implemented.
- **Minimum**. This level corresponds to implementation not including the bytes required by the standard implementation level, but requires that at least bit 0 of byte 0 is correctly implemented.
- **Enhanced**. This name is given to any implementation that excess the standard implementation level.

Serial Copy Management System (SCMS)

For the consumer format a set of rules have been developed that are intended to restrict the proliferation of digital copies (made at home) of pre-recorded material with copyright. This Serial Copy Management System (SCMS) requires that status indication follows rules that assist in limiting copying of such material to one or two generations.

As stated earlier, bit 2 indicates by being zero that copyright is asserted. If that is the case, then bit 15, the “L-bit,” is used to indicate the generation status of the channel (bit 15 is part of the Category Code field).

Since this system was introduced after IEC60958 (then called IEC958) was first published, there are some apparently complex rules in order to retain compatibility with the Compact Disc format, which pre-dates SCMS.

Generally, the L-bit is set to "one" to indicate that the signal is from pre-recorded material, and is cleared to "zero" to indicate that a "home-copy" has been made. The SCMS rules require that a home-copy with copyright asserted cannot be copied again.

With the category codes for laser-optical products (such as CD) and digital broadcast receivers, the sense of the L-bit is reversed. In these cases the L-bit is set to "zero" to indicate that the signal is from pre-recorded material.

There are two category codes for which the devices are deemed to be without knowledge of copyright status. These are the "general" category code, 00000000_h; and the code for converters for analog signals without copyright information, $0110000L_h$ (the sense of the L-bit is determined by the product category). An SCMS compliant recorder, such as a consumer-mode DAT recorder, will record a signal with these codes and ignore the copyright flag. On replay it will indicate that the material is the equivalent of "pre-recorded." This has the effect that one further home-copy generation is allowed—giving two generations of copying in all.

Validity bit

Since the validity bit was poorly defined, it is not clear how equipment should behave on receipt of a signal with the bit set to "one"—indicating an invalid signal.

Strictly to the specification, the audio data word associated with an invalid data status indication should not be converted to analog. This would infer that any equipment that is not simply passing the validity flag through with the audio data should mute or interpolate the associated audio word, so that any following equipment does not convert it to analog. This behavior can be verified in APWIN by setting the valid flag manually in the DIO panel and observing the effect on a DUT.

If the validity bit is being passed through with the audio, then it is important that there they remain exactly aligned. If there were a time offset between the flag and the audio, the flag could align with a different audio word, with the result that the originally incorrect word would be wrongly marked as valid, and a correct word would be wrongly marked as invalid. Correct alignment can be verified with dedicated test equipment. The author is unaware of any that is commercially available.

Some equipment uses the validity bit to indicate that error concealment has taken place. This non-compliant behavior is common to a large number of CD players. If the response of the DUT is to replace invalid samples with a mute

or concealment that is more noticeable than the original concealment, then that may be seen as a disadvantage.

Because of this confusion it is common for equipment to be required to ignore the validity flag.

User data

For the professional format, the user data stream can be used for a variety of applications, which can have different formatting and relations to the other interface data.

If a device under test is aiming to be completely transparent to all defined and future formats, this can be verified by passing through a known pseudo-random data stream (such as Bittest) and confirming that it has not been corrupted.

If the DUT is not transparent, the test can be repeated for the three standard formats that have already been defined in case the device only supports one of those subsets. This could be a large exercise without a protocol analyzer. Unfortunately, the author is not aware of any commercially available AES3 or IEC60958 user-data protocol analyzers.

Simpler than that, it may be possible to put the DUT in the path between equipment that is communicating in the appropriate format and then to confirm that the user data messages are still getting through.

For the consumer format the general user data format is the only one specified. The technique described in the previous paragraph could also be used (with appropriate consumer equipment) to test for transparency to consumer-format user data messages.

Channel Status Annex

The annex lists these fields for IEC60958-3:2000 and AES3-1997 but is not authoritative. A copy of the latest revision of the appropriate standard should be used if possible.

Consumer format channel status

The following tables apply if bit 0 is set to "zero" indicating consumer application. The definitions for bytes 1 to 23 only apply if bits 1, 6, and 7 are set to "zero" indicating linear PCM audio and channel status mode 0. In practice, the standard requires that the value of bits 6 and 7 are always set to "zero" until any future revision defines them.

Byte	Consumer format channel status fields							
0	Pro/con = 0	Non-audio = 0	Copyright	Emphasis			Channel status mode = 00	
	bit 0	1	2	3	4	5	6	7
1	Category code							
	bit 8	9	10	11	12	13	14	15
2	Source number				Channel number			
	bit 16	17	18	19	20	21	22	23
3	Sampling frequency				Clock accuracy			
	bit 24	25	26	27	28	29	30	31
4	Word length				(Future original sampling frequency?)			
	bit 32	33	34	35	36	37	38	39
5–23	Reserved							
	bits 40–191							

Table 2. Consumer format channel status fields.

Consumer format channel status field interpretations			
Bits	**label**	**interpretation**	
0	pro/con	0: consumer; 1: professional format	
1	non-audio	0: suitable for conversion to analog audio using linear PCM 1: not suitable	
2	copyright	0: asserted; 1: not asserted	
3–5	emphasis	000: Emphasis not indicated 100: emphasis—CD-type	
6–7	channel status mode	00: mode zero; other values reserved	
8–15	category code		
16–19	source number	(bit 16 is LSB)	
20–23	channel number	(bit 20 is LSB)	
24–27	sampling frequency	0000: 44.1 kHz 0100: 48 kHz 1100: 32 kHz	
28–29	clock accuracy	10: Level I, ±50 ppm 00: Level II, ±1000 ppm 01: Level III, variable pitch shifted	
30–31	reserved		
32	word length (field size)	0: Maximum length 20 bits 1: Maximum length 24 bits	
33–35	word length		if bit 32 = 1 / if bit 32 = 0
		000:	not indicated / not indicated
		101:	24 bits / 20 bits
		001:	23 bits / 19 bits
		010:	22 bits / 18 bits
		011:	21 bits / 17 bits
		100:	20 bits / 16 bits
36–39	reserved		
40–191	reserved		

Table 3. Consumer format channel status interpretations.

Note that the bit fields are shown with the earliest, or lowest-numbered bit, to the left. As the format is LSB-first, this notation this is opposite to the conventional binary notation, which would show the MSB to the left.

Professional format channel status

The following tables apply if bit 0 is set to “one” indicating professional application.

Byte	Professional format channel status fields							
0	Pro/con = 1	Non-audio	Emphasis			Lock	Sample frequency	
	bit 0	1	2	3	4	5	6	7
1	Channel mode				User bit management			
	bit 8	9	10	11	12	13	14	15
2	Use of auxiliary mode sample bits			Source word length			Alignment level	
	bit 16	17	18	19	20	21	22	23
3	Channel Identification for multichannel application							
	bit 24	25	26	27	28	29	30	31
4	DARS			Sample frequency (fs)				fs scaling
	bit 32	33	34	35	36	37	38	39
5	Reserved							
	bit 40	41	42	43	44	45	46	47
6	Alphanumeric channel origin data (first character)							= 0
	bit 48	49	50	51	52	53	54	55
7	Alphanumeric channel origin data							= 0
	bit 56	57	58	59	60	61	62	63
8	Alphanumeric channel origin data							= 0
	bit 64	65	66	67	68	69	70	71
9	Alphanumeric channel origin data (last character)							= 0
	bit 72	73	74	75	76	77	78	79
10	Alphanumeric channel destination data (first character)							= 0
	bit 80	81	82	83	84	85	86	87
11	Alphanumeric channel destination data							= 0
	bit 88	89	90	91	92	93	94	95
12	Alphanumeric channel destination data							= 0
	bit 96	97	98	99	100	101	102	103
13	Alphanumeric channel destination data (last character)							= 0
	bit 104	105	106	107	108	109	110	111
14	Local sample address code (32-bit binary, LSW)							
	bit 112	113	114	115	116	117	118	119
15	Local sample address code (32-bit binary)							
	bit 120	121	122	123	124	125	126	127
16	Local sample address code (32-bit binary)							
	bit 128	129	130	131	132	133	134	135
17	Local sample address code (32-bit binary, MSW)							
	bit 136	137	138	139	140	141	142	143
18	Time of day code (32-bit binary, LSW)							
	bit 144	145	146	147	148	149	150	151
19	Time of day code (32-bit binary)							
	bit 152	153	154	155	156	157	158	159
20	Time of day code (32-bit binary)							
	bit 160	161	162	163	164	165	166	167
21	Time of day code (32-bit binary, MSW)							
	bit 168	169	170	171	172	173	174	175
22	reserved				Reliability flags			
					bytes 0-5	bytes 6-13	bytes 14-17	bytes 18-21
	bit 176	177	178	179	180	181	182	183
23	Cyclic redundancy check character (CRCC)							
	bit 184	185	186	187	188	189	190	191

Table 4. Professional format channel status fields.

Professional format channel status field interpretations			
Bits	**label**	**interpretation**	
0	pro/con	0: consumer; 1: professional format	
1	non-audio (or, more accurately, "not linear PCM")	0: audio data is linear PCM samples 1: other than linear PCM samples	
2–4	emphasis	000: Emphasis not indicated 100: No emphasis 110: CD-type emphasis 111: J-17 emphasis	
5	lock	0: not indicated 1: unlocked	
6–7	sampling frequency	00: not indicated (or see byte 4) 10: 48 kHz 01: 44.1 kHz 11: 32 kHz	
8–11	Channel mode (SCDSR = single channel double sample rate)	0000: not indicated (default to 2 ch) 0001: 2 channel 0010: 1 channel (monophonic) 0011: primary / secondary 0100: stereo 0101: reserved for user applications 0110: reserved for user applications 0111: SCDSR (see byte 3 for ID) 1000: SCDSR (stereo left) 1001: SCDSR (stereo right) 1111: Multichannel (see byte 3 for ID)	
12–15	user bit management	0000: no indication 0001: 192-bit block as channel status 0010: As defined in AES18 0011: user-defined 0100: As in IEC60958-3 (consumer)	
16–18	use of aux sample word	0000: not defined, audio max 20 bits 0001: used for main audio, max 24 bits 0010: used for *coord*, audio max 20 bits 0011: user-defined	
19–21	source word length	000: 001: 010: 011: 100: 101:	if max = 20 bits not indicated 23 bits 22 bits 21 bits 20 bits 24 bits
			if max = 24 bits not indicated 19 bits 18 bits 17 bits 16 bits 20 bits

Table 5. Professional format channel status interpretations.

Professional format channel status field interpretations (Cont.)		
Bits	**label**	**interpretation**
22–23	alignment level	00: not indicated 01: –20 dB FS 10: –18.06 dB FS
24–31	channel identification	if bit 31 = 0 then channel number is 1 plus the numeric value of bits 24-30. if bit 31 = 1 then bits 4–6 define a multichannel mode and bits 0–3 give the channel number within that mode.
32–33	digital audio reference signal (DARS)	00: not a DARS 10: DARS grade 2 (+ / –10 ppm) 01: DARS grade 1 (+ / –1 ppm)
35–38	sampling frequency	0000: not indicated 1000: 24 kHz 0100: 96 kHz 1001: 22.05 kHz 0101: 88.2 kHz 1101: 176.4 kHz 1111: User defined
39	sampling frequency scaling	0: no scaling 1: apply factor of 1 / 1.001 to value
48–79	alphanumerical channel origin	four-character label using 7-bit ASCII with no parity. Bits 55, 63, 71, 79 = 0.
80–111	alphanumerical channel destination	four-character label using 7-bit ASCII with no parity. Bits 87, 95, 103, 111 = 0.
112–143	local sample address code	32-bit binary number representing the sample count of the first sample of the channel status block.
144–175	time of day code	32-bit binary number representing time of source encoding in samples since midnight
176–183	reliability flags	0: data in byte range is reliable 1: data in byte range is unreliable
184–191	CRCC	00000000: not implemented X: error check code for bits 0–183

Table 5 (cont.). Professional format channel status interpretations. (cont.)

Note that the bit fields are shown with the earliest, or lowest-numbered bit, to the left. As the format is LSB-first, this notation this is opposite to the conventional binary notation, which would show the MSB to the left.

List of Files

The following APWIN files are referred to or used in this Application Note:

- DIF cable sim eye.at2c
- DIF eye test.at2c
- DIF jitter tolerance.at2c
- Intrinsic jitter spectrum.at2c
- JTF eval board.at2c
- output term test back to back.at2C
- output term test DAT.at2C
- AES3jittertolerance.adq
- IEC60958-3jittertolerance.adq

These files are on the companion CD-ROM. You may also download the files from the Audio Precision Web site at audioprecision.com. These tests and data files are designed for use with System Two Cascade, but with minor changes can be modified to work with System Two as well. Please check the README.DOC file in the same folder for further information.

References

1. AES3-1992—'Recommended Practice for Digital Audio Engineering—Serial Transmission Format for Two-Channel Linearly Represented Digital Audio Data,' J. Audio Eng. Soc., vol. 40 No. 3, pp 147–165 (June 1992). (The latest version including amendments 1–4 is available from http://www.aes.org)

2. IEC-60958-1:1999, 'Digital audio interface—Part 1: General,' International Electrotechnical Commission, Geneva. (December 1999). http://www.iec.ch

3. IEC-60958-3:1999, 'Digital audio interface—Part 3: Consumer applications,' International Electrotechnical Commission, Geneva. (December 1999). http://www.iec.ch

4. IEC-60958-4:1999, 'Digital audio interface—Part 4: Professional applications,' International Electrotechnical Commission, Geneva. (December 1999). http://www.iec.ch

5. EBU Tech. 3250-E 'Specification Of The Digital Audio Interface (The AES/EBU Interface)'—Second Edition August 1992. European Broadcasting Union, Geneva. http://www.ebu.ch

6. AES11-1997—'AES Recommended Practice for Digital Audio Engineering—Synchronization of Digital Audio Equipment in Studio Operations,' J. Audio Eng. Soc., Vol. 45 No. 4, pp 260–269 (April 1997). (The latest version is available from http://www.aes.org)

7. ITU-T Recommendation J.17—'Pre-emphasis used on sound-programme circuits,' International Telecommunication Union, Geneva. (November 1988). http://www.itu.int

8. ITU-R Recommendation BS.647-2—'A digital audio interface for broadcasting studios,' International Telecommunication Union, Geneva. (March 1992). http://www.itu.int

9. IEC-61937—'Digital audio—Interface for non-linear PCM encoded audio bitstreams applying IEC 60958,' First Edition, International Electrotechnical Commission, Geneva. (April 2000). http://www.iec.ch

10. SMPTE 337M-2000: for Television—'Format for Non-PCM Audio and Data in an AES3 Serial Digital Audio Interface,' Society of Motion Picture and Television Engineers, White Plains, NY, USA. http://www.smpte.org

11. AES-3id-1995—'AES Information Document for Digital Audio Engineering—Transmission of AES3 Formatted Data by Unbalanced Coaxial Cable,' J. Audio Eng. Soc., vol. 43 No. 10, pp. 827–844 (October 1995). (The latest version is available from http://www.aes.org)

12. SMPTE 276M-1995: for Television—'Transmission of AES/EBU Digital Audio Signals Over Coaxial Cable,' Society of Motion Picture and Television Engineers, White Plains, NY, USA. http://www.smpte.org

13. See the chapter **Jitter Theory**, beginning on page 3 of this book.

14. Julian Dunn, Barry McKibben, Roger Taylor and Chris Travis—'Towards Common Specifications for Digital Audio Interface Jitter,' Preprint 3705, presented at the 95th AES Convention, New York, (October 1993). http://www.nanophon.com/audio

15. See the chapter **Analog-to-Digital Converter Measurements** beginning on page 37 of this book.

16. See the chapter **Digital-to-Analog Converter Measurements** beginning on page 83 of this book.

Julian Dunn 1961–2003

Julian Dunn took degrees in Astronomy and then Medical Electronics at London University, where he first became interested in signal processing. After graduating in 1984 he joined the BBC Designs Department, also in London. There he started to design digital audio equipment, as part of work for BBC Radio in prototyping equipment for use with the new digital audio recorders, mixing consoles and transmission systems.

After a year working at the Mullard Radio Astronomy Observatory in Cambridge, England, Julian joined Prism Sound as a consultant, where he returned to designing digital audio equipment. In 1998 he formed his own digital audio design company, Nanophon. Nanophon provided specialist consultancy in digital audio conversion, DSP software and hardware, digital audio interfacing and clock recovery systems. The Nanophon Web site is currently maintained at www.nanophon.com.

Julian presented technical papers to AES conferences and conventions on various topics in digital audio, and was a contributor to the work of the AES standards digital audio subcommittee since 1991. He served as the chairman of the AES working group for digital input/output interfacing which is responsible for AES3, and was a joint project leader for the IEC team revising IEC60958.

Among his leisure interests Julian enjoyed watching cricket, traveling the Caribbean and the maintenance and repair of old equipment of various sorts.

He was a treasured colleague and a dear friend to many of us, and a brilliant light in the world of digital audio engineering. On January 23, 2003 Julian succumbed to the leukemia he had been battling throughout the previous year. He will be sorely missed.