Down and Dirty Facts About Distortion

Electronic Musician®

PERSONAL STUDIO | RECORDING | PRODUCTION | SOUND DESIGN

DIGITAL CLOCKING
DEMYSTIFIED

What you need to know and why
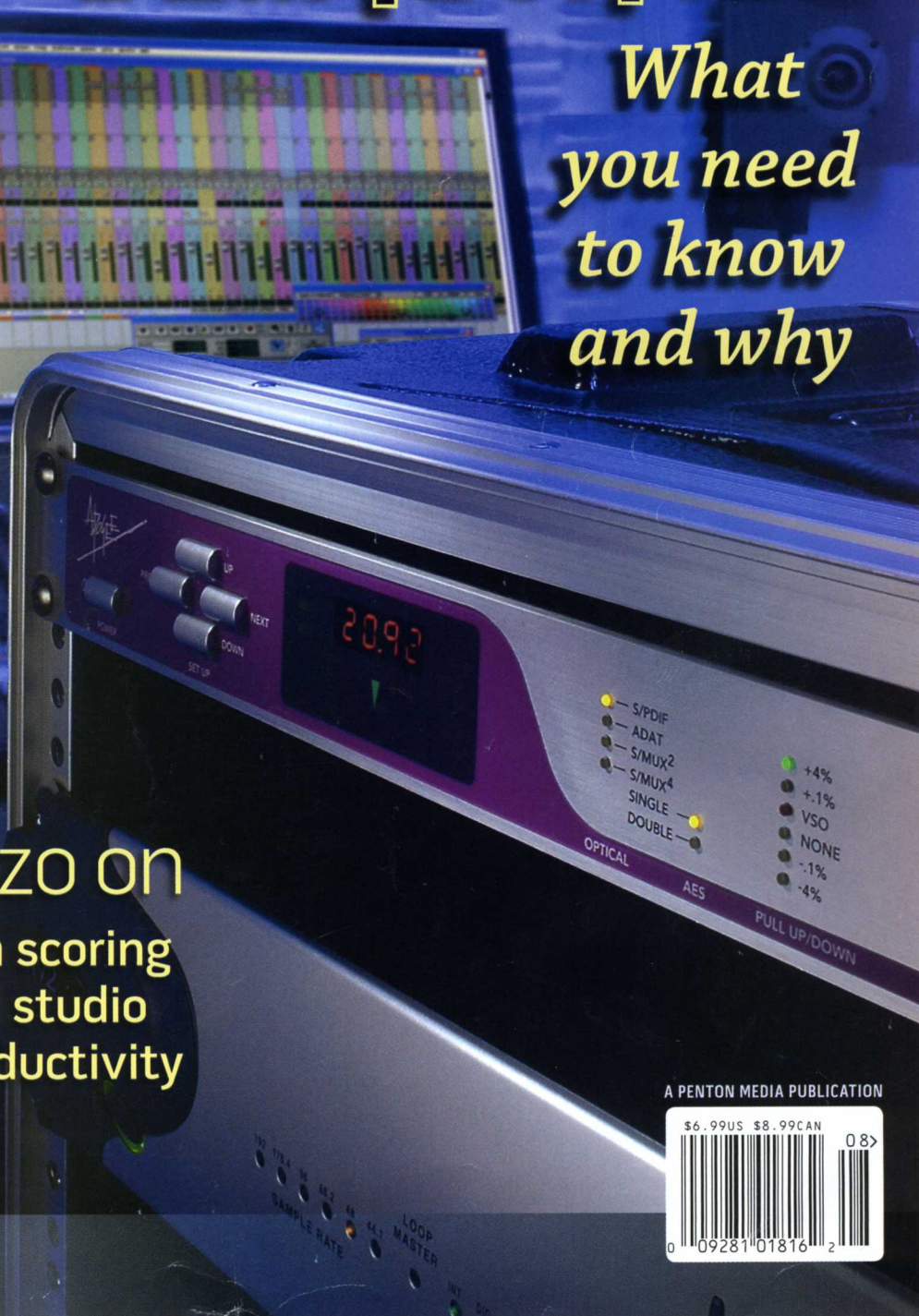
MIX SMARTER AND FASTER IN PRO TOOLS

REMOVING VOCAL ARTIFACTS

REVIEWS

Digidesign Digi 003
E-mu Emulator X2
Korg MR-1 & MR-1000
and 5 more
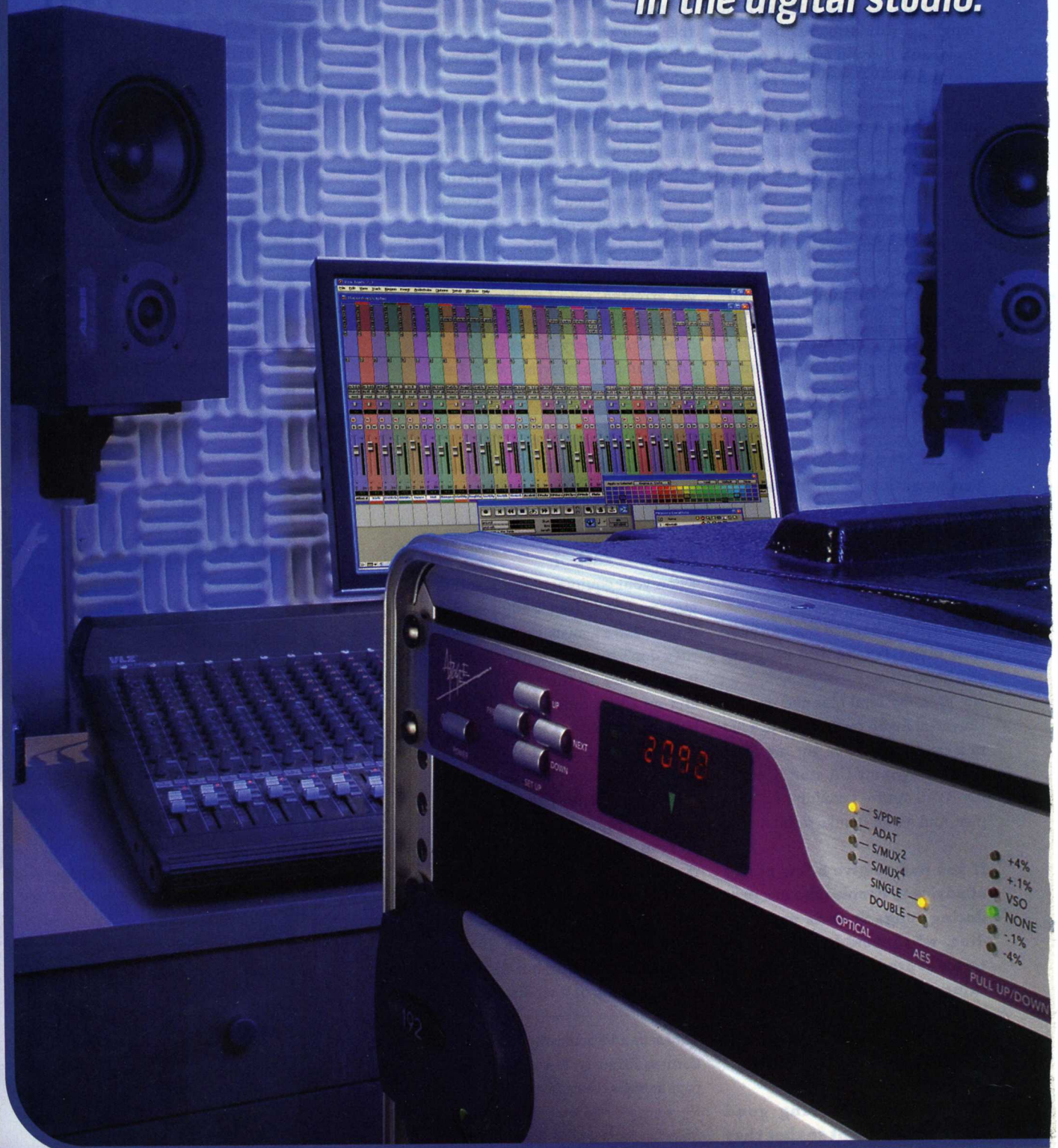
Carmen Rizzo on
film scoring and studio productivity

WWW.EMUSICIAN.COM

Principles of clocking in the digital studio.

2092

- S/PDIF
- ADAT
- S/MUX2
- S/MUX4
- SINGLE
- DOUBLE

OPTICAL

AES

- +4%
- +.1%
- VSO
- NONE
- -.1%
- -4%

PULL UP/DOWN

By Larry the O

Timing is Everything

Although digital clocking is not a particularly sexy subject, it is certainly an important one. Most project studios today are built around a computer running multitrack recording software with audio interfaces and other outboard devices connected to it. Computers operate on a step-by-step basis, and precisely timed execution is an essential element of digital recording. Making sure hardware and software all march in lockstep is crucial, and that's what clocking is about.

fs x 256
fs x 4
fs x 2
fs x 1
fs /2
fs /4

WC OUT 5

Despite clocking's importance in the digital studio, its fundamentals are unclear—or just plain voodoo—to many recordists. One of the main objectives of this article is to convey enough about the subject of clocking that you feel equipped to contact a manufacturer and ask the right questions to get the information you need to maximize your studio setup. Although I probably can't make the entire subject simple for you, I will try to make it at least understandable. I can provide information that is important for you to have but that is not readily available elsewhere.

To help get the lowdown on this broad and tricky subject, I spoke with several experts. I'd like to recognize and thank them here (in alphabetical order): independent consultant Joe Bryan, formerly Universal Audio's head of engineering and architect of UA's 2192 Master Audio Interface; Jim Cooper, director of marketing, and Jon Foley, technical-support supervisor, at MOTU; Doug Ford, engineering supervisor at Skywalker Sound; Gannon Kashiwa, Digidesign's market manager; Igor Levin, sync guru at Antelope Audio; Dan Phillips, product manager at Korg R&D; and Lucas van der Mee, senior design engineer at Apogee Electronics. All contributed great insight and technical detail, without which I could not have formulated a coherent picture of the world of digital clocking.

Time to Begin

As with most digital technology, clocking involves numerous subtleties about which experts have differing opinions. In most cases, though, grasping a handful of fundamental principles is sufficient. I will present these principles here and fill in some other details along the way.

The first and most important principle you need to know about using clocks in digital studios is that there can be only one master, to which every other device must be slaved. Uncompressed digital audio plays at a fixed rate. If two devices in a signal chain are set to the same sampling rate, but each runs off its own internal clock,

they will sample at the same rate, but the odds are infinitesimal that they'll do it at the exact same time. Digital audio streams are delicate; if discontinuities result from two devices having different ideas about when sample time is, the outcome is likely to be a pop or glitch.

If, however, every device in a system is locked to a single clock source, then samples will be taken, passed around, and output in beautiful synchronicity, and audio will play without ugly artifacts. When the kids play nice together, life is sweet, but of course, almost nothing is ever quite that easy. Things can still go wrong, and I'll examine some common problems once I've explained the basics.

The second fundamental principle of clocking is the distinction between rate synchronization and location synchronization, a frequently misunderstood concept (see **Fig. 1**). The word *clock* usually indicates something that tells the time, but a clock in the digital studio marks only the rate at which samples should occur. Like a metronome, it carries no information—nothing to distinguish one clock cycle from another. With no concept of elapsed time in the signal (beyond the duration of a single sample period), it is impossible to state a location for any clock cycle relative to any other. Clocks in the digital studio are strictly timing related. In contrast, location synchronization systems like timecode embed an address for each frame of data, making it possible to locate to any specified point.

The third important principle is that digital audio clocks are crucial only in real-time contexts. Information being distributed in real time requires timing to be preserved, but by definition, timing is not important in non-real-time transmission. Thus, for example, a simple file copy from one hard disk to another does not entail clocking.

Self-Clocking Systems

There are two kinds of systems for delivering clock signals to digital devices in a studio: self-clocking sys-
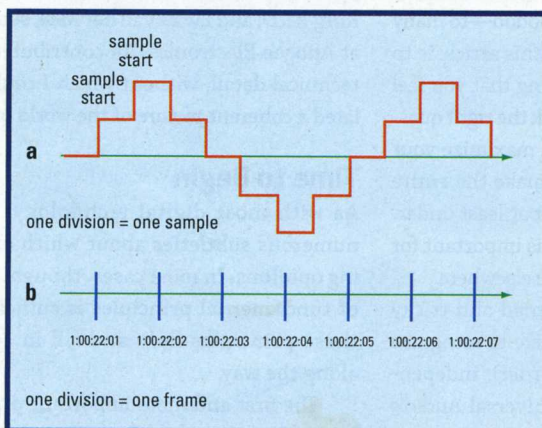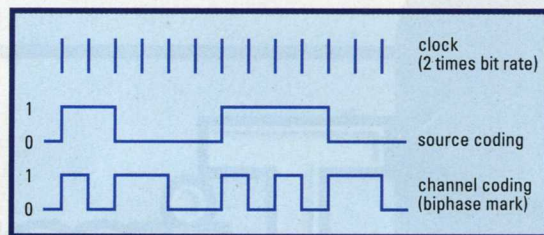

FIG. 2: A chart from the AES3-2003 standard illustrates how data and clock are encoded in a self-clocking interface. Here you see the clock signal on top, the audio data in the middle, and the resulting biphase mark coded data on the bottom.

tems and distributed systems, better known as word clock. Self-clocking schemes embed timing data into a real-time digital audio stream. The self-clocking interfaces most familiar to readers are AES3 (also called AES/EBU), S/PDIF, and ADAT Optical (also called Lightpipe). S/PDIF can use the same Toslink optical ports used by Lightpipe, but the data streams are totally different. In this article, *S/PDIF* refers to transmission over unbalanced coaxial and not optical lines.

In self-clocking systems, the receiving device must extract the embedded clock signal from the digital audio. This is a critically important process with significant impact on the amount of *jitter* (distortion caused by irregular timing, which I'll discuss later in detail) in the system. Self-clocking systems usually run at some multiple of the sampling rate; for example, AES3 runs at 64 times the sampling rate (see **Fig. 2**).

AES3 and S/PDIF incorporate the clock signal into the data stream as part of the encoding scheme, which is called *biphase mark code*. Without going into a full explanation of this coding, the start of every bit in the stream is marked by a state transition. These transitions (it makes no difference whether a transition is from high to low or low to high) are detected and extracted to create the clock. Lightpipe, however, uses NRZI (non-return-to-zero, inverted) coding, which does not have a transition at the beginning of each bit, making clock recovery more difficult.

Some experts describe audio sent over FireWire (aka IEEE 1394) and USB as self-clocking, but there's a major difference between those two systems and the others I've described. In AES3, S/PDIF, and Lightpipe, samples are streaming in real time, with the clock marking the start time of each sample. FireWire and USB, on the other hand, break the stream of samples into packets of data and transmit them sequentially. The packets are received by buffers in the receiving device, reassembled into the original sample stream, and then sent from the buffer at the appropriate time, as determined by the device's sample clock—a process called *reclocking*. Because the samples are reclocked, FireWire and USB are not dependent on embedded clocks for timing.

FIGS. 1a and 1b: In 1a, sample start times (and therefore rate) are defined, but nothing differentiates one sample from another. In 1b, a timecode address identifies each frame; thus, positional clocking allows location to any frame.

However, Levin points out that there are still sources of jitter in these interfaces. For instance, FireWire and USB send packets that are synchronization messages, but the architecture of the interfaces can introduce small variations in the transmission time of these packets. When a phase-locked loop (more on this later) at the receiving end tries to extract the clock from the slightly irregular timing messages, jitter is created.

Word-Clock Systems

Distributed (word-clock) systems deliver, or distribute, a clock signal that is entirely separate from audio data. It most commonly appears as a simple square-wave signal generated by a device designated as the master and routed to each of the other devices. Word clocks usually run at the sampling rate and take the form of unbalanced signals on BNC connectors. That format is a legacy of the first digital audio-clocking system, which was SDIF (Sony Digital Interface, not to be confused with S/PDIF, which came later). SDIF set the mold of using BNC connectors and a square wave running at the sampling rate.

At the time that SDIF appeared, analog-to-digital converters (ADCs) and digital-to-analog converters (DACs) did conversions directly to and from PCM audio. Some early devices that accepted word clock even used that clock directly to drive the sample clock, meaning that any irregularities in the incoming clock were propagated into the device as jitter.

Since those earliest days, word clock has been a common method of clocking, even in some early multichannel interfaces that didn't use BNC connectors. For example, the TDIF format used by Tascam's digital recorders carries a word clock on one of the conductors in the 25-pin D-sub connector employed by the interface. And though Lightpipe is self-clocking, Alesis also released ADAT Sync at the same time. ADAT Sync is a companion interface that carries a word-clock signal on a 9-pin D-sub connector, thus eliminating the need to do the clock extraction required when syncing to Lightpipe and avoiding the jitter that could incur.
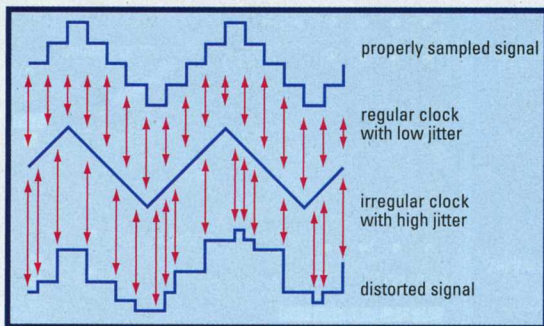

FIG. 3: This is an example of jitter-induced waveform distortion in A/D conversion. Note how timing irregularities in the jittery clock create distortion, which is exaggerated to illustrate the point.

Digidesign Pro Tools systems have always been able to slave to an external clock. For many years, however, Digidesign used its own form of word clock called Superclock, which runs at 256 times the sampling rate. Current Pro Tools systems use standard sampling-rate word clock, and several connection schemes are possible for clocking them, but the most common (when using more than one interface) is Digidesign's Loop Sync system, which I'll describe later.

Once AES3 came into widespread use, engineers realized that AES3 streams containing audio data do not result in as clean a clock as those with no audio data. In a stream with no audio data, the waveform is quite regular. Therefore, waveform distortion from capacitance or termination problems should affect each clock the same, and the waveform, while distorted, should still be regular. A stream containing audio has many more irregular transitions (a characteristic of biphase mark code), which means more noise in the system and ultimately a less stable clock.

The AES11 standard specifies the use of an AES3 stream containing no audio data as a reference clock that can be distributed to all devices in a system. In effect, AES11, also called *AES black* (a reference to *black burst,* a composite video signal without picture information used as a master timing reference in broadcast and video-production facilities), acts as a word clock, turning a self-clocking interface into a distributed clock interface. AES11 is used by some high-end and multiroom production facilities.

A Clock to the System

Project studios working to picture use QuickTime or AVI movies as their video source, but professional postproduction and video facilities still use black burst as the house-sync source to which everything is locked. Their reliance on black burst creates one of several clocking challenges that post houses deal with. Word clock is commonly derived from the horizontal sync pulse in the black-burst signal, at a rate of the number of video lines per frame multiplied by the frame rate. For NTSC color, that works out to 15.734 kHz. The mathematical relationship of this frequency to typical audio sampling rates is just plain ugly and difficult to properly resolve, but most master clock devices and many professional digital audio devices can do it.

Whenever film and video are both used in a production, the issue of *pull-up* and *pull-down* crops up. Explaining these is beyond the scope of this article, but I'll just say that relating pull-up and pull-down rates to digital audio clocking is yet another hurdle to overcome. Because problems occur in the types of facilities that deal with black burst, it is not surprising that professional audio-for-video devices and master clock devices are usually equipped to resolve such oddities against word clock, AES3, and S/PDIF. Such devices sometimes handle Lightpipe and other references too.

Clocking can become a real issue in audio post-production for film and video. Problems often arise in audio post because of decisions made during production without coordinating with the post house. Agreeing on clocking systems before production gets under way will avoid off-speed clocking issues that can cause sound and picture to slip out of sync.

High-end professional situations can also involve audio networking systems such as MADI—the AES multichannel standard—or CobraNet, EtherSound, and Optocore. MADI is structured similarly to AES3, except that it carries a separate distributed clock signal in addition to embedded clock information, so either can be used. CobraNet, EtherSound, and Optocore all transmit audio in packets, accompanied by a real-time word clock in the form of timed network messages. While it may seem like a self-clocking interface because the clock travels with the data, it is carried by time-stamp messages and does not have to be extracted from the data. Individual nodes synchronize their local clocks with the master clock.

Jitter: Too Much Caffeine?

By now, you may be wondering how clocking affects your sound. When clocking is not done well, the audi-
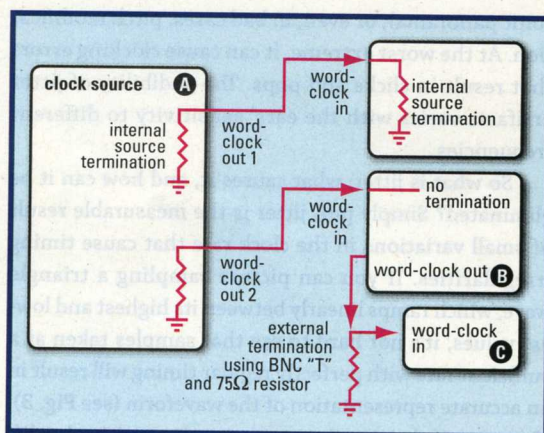

FIG. 4: In this transmission line, the clock source (device A) is internally terminated. Device B has no termination and daisy-chains the clock to device C, which also has no termination but uses a BNC T-connector with termination on one leg.

ble effects can range from very subtle to infuriatingly blatant. You've probably at least heard of jitter, and it is a significant topic. The effects of jitter can take the form of loss of clarity and definition, a reduction in the soundstage (that is, less width and depth in the

sonic panorama), or even, in bad cases, pitch modification. At the worst extreme, it can cause clocking errors that result in clicks and pops. The audibility of jitter artifacts varies with the ears' sensitivity to different frequencies.

So what is jitter, what causes it, and how can it be eliminated? Simply put, jitter is the measurable result of small variations in the clock rate that cause timing irregularities. If you can picture sampling a triangle wave, which ramps linearly between its highest and lowest values, it's not hard to see that samples taken at a sufficient rate with perfectly regular timing will result in an accurate representation of the waveform (see **Fig. 3**). However, if the time between samples varies, the difference between two successive samples will result in waveform distortion. Because such timing variations are quite small in most cases, the resulting distortion may be subtle, but it is still there.

Because we live in an imperfect world and every clock has some amount of jitter, totally eliminating jitter just isn't possible. The goal is to minimize the amount of jitter that occurs in your system. Random jitter is caused by crosstalk and other induced sources, whereas periodic jitter can be signal related, or the result of two clocks in the system beating against each other.

Periodic-jitter-induced distortion occurs because jitter modulates the audio data. The jitter's amplitude and the signal's rate of change determine the degree of modulation. Periodic jitter is a form of frequency modulation and therefore produces sidebands. Thus, the more high-frequency content in the signal, the more susceptible it is to jitter-induced distortion. Jitter can also be cumulative, so a clock that passes through several devices can end up with considerably more jitter than it began with. For example, cumulative jitter could come into play if a signal passes through a poorly designed digital router (an electronic, not a physical, patch bay), which can add considerable jitter.

Another fundamental principle, and the most important fact to know about jitter, is that it is of consequence only in devices in which conversions occur—either ADCs, DACs, or real-time asynchronous sampling-rate converters (ASRCs). An ADC or ASRC encodes audio for

storage or processing by converting it from an analog to a digital signal. Remember, analog audio is a continuous waveform, and a sample is a snapshot of that signal at the moment of sampling. Variation in the regularity with which the signal is sampled results in a distorted capture of the waveform. Distortion introduced by jitter on an ADC's clock is recorded as part of the signal and can never be removed; the same applies to poorly clocked ASRCs. Once jitter occurs in the recording process, you're stuck with it, which is one reason why high-quality ADCs are important.

On the other hand, jitter in a DAC creates distortion in playback: you will hear it, but it has no effect on the source data. In that situation, reducing jitter will improve the sound. In fully digital transfers, data is reclocked after it has been received; consequently, as long as none of the data is corrupted during transfer, jitter in the clocking will have no effect. Therefore, the rest of this discussion will not pertain to digital-to-digital transfers.

All That Jitters

The three most common causes of jitter are a poor-quality clock source, a bad transmission line (that is, the clock connections between devices), and a poor-quality receiver in the slave device. I'll discuss these causes one at a time.

Assuming that a clock is able to generate a good square wave, its stability determines its quality. If the source clock does not have precise and regular timing, nothing down the line can improve the situation. In earlier days of digital audio, clock quality was quite variable and many devices had very poor clocks. Current digital audio devices generally have much-better-quality clocks, though the operative word here is *generally*; some devices still do not. Mostly, they are inexpensive devices for which the manufacturer can't or won't spend the money to develop a high-quality clock or purchase more-expensive parts.

So how can you tell which devices have good clocks and which don't? Several common oscillator architectures are commonly used for digital audio clocks, each with benefits and drawbacks that include their jitter characteristics. But without specialized test equipment and sufficient knowledge, a user has no easy way to know for sure how good a clock is.

When possible, making A/B/C comparison auditions in your own studio can reveal what is most important: the differences in how devices sound. But it can be difficult to acquire numerous pieces of equipment to audition; often, the best you can do is approach companies you feel are technically competent and concerned with quality, and question them closely based on the material in this article to get a feel for the quality of their clocking.
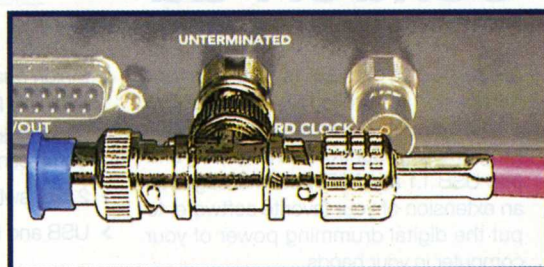

FIG. 5: This is properly terminated word-clock input using a BNC T-connector and a 75Ω terminator.

Playing by the Rules

Assuming that the device providing the master clock for your studio contains a stable clock, your next concern is the connection that conveys that clock to the various devices in the studio. The next principle I'll discuss is the concept of a transmission line. A transmission line must be properly terminated at each end but nowhere in the middle (see **Fig. 4**). Ideally, the transmission will degrade the signal as little as possible. Termination and length both matter. I will attempt to explain these ideas with a colorful analogy.

Imagine that you're playing a game of catch with a friend. If your friend throws the ball too hard or too soft, it will be difficult for you to catch correctly. If it is windy, rainy, or in some other way nasty outside, it will affect how the ball travels, again making it difficult to catch correctly. If the ball is wet, it will be tougher to catch than if it were dry. In addition, the person catching the ball has to be properly equipped. If you're trying to catch a baseball without a mitt, it could be painful and you might drop the ball. If you try to catch it with something made of metal or some other rigid material, the ball will bounce away from you. If, however, you have a proper mitt, the environmental conditions are reasonable, and the ball is thrown well, then everything will probably go

fine—unless either of you is simply an idiot.

Now let's translate that analogy into a clocking situation. Throwing the ball correctly and using a suitable mitt translate into proper termination—that is, having the correct impedance at each end of the transmission line. For BNC word clocks, 75Ω is the proper termination, and a resistor of that value can provide it quite simply. S/PDIF also requires 75Ω termination, and AES3 uses 110Ω termination. Ensuring that your clock lines are properly terminated at the ends takes some thought and care, but it is key to minimizing jitter, especially for AES3 and S/PDIF, which operate at much higher rates than word clock. You have to assume that a clock output has the proper impedance termination at the source, so your real concern will be with terminating the far end of the line.

Many devices provide internal termination; others, like the MOTU interfaces in my studio, do not. The device's user manual should tell you whether word-clock inputs are terminated, but sometimes it is necessary to contact the manufacturer to find out. Some devices have internal terminators that can be switched in or out, to allow the device to be either at the end (terminated) or the middle (unterminated) of a clock chain. If a device has such a switch, it may appear on the back panel next to the BNC
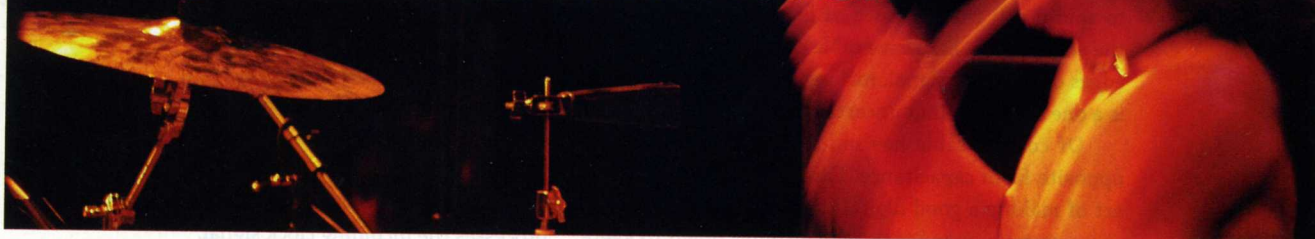
connector, as a software setting, or as a jumper that must be changed inside the unit. Again, consult the manual.

If a device has no internal termination, there are simple options for proper external termination. The simplest is to get a BNC T-connector, which has two female and one male connection. The male goes into the receiving equipment, the clock connects to one female connector, and the other female connector should get a BNC connector with a 75Ω resistor soldered between the middle (signal) pin and ground. If you can use a soldering iron, you can do this yourself (see **Fig. 5**).

If the transmission line is not properly terminated at the receiving end, it is analogous to a metal mitt: the signal can reflect back into the transmission line, corrupting the clock signal and introducing jitter. It is important that clock lines not be overterminated or underterminated; either causes waveform distortion that can bring problems. A few clock manufacturers provide diagnostic LEDs to indicate whether clock outputs are properly terminated at the far end.

Stable Cables

The rain and wind in my analogy refer to the principle of cable capacitance. Now you'll learn why you shouldn't use mic cables for AES digital connections, hi-fi cables for S/PDIF, or even cheap BNC cables for word clock.

Every cable has some capacitance, but the amount varies greatly with the cable's materials and construction. Capacitance occurs between the cable's signal line and ground and acts as a lowpass filter. If you've ever connected a synthesizer to an oscilloscope and looked at a square wave as you lowered the lowpass filter's cutoff frequency, you've seen the waveform's edges start to round off; the presence of high frequencies make the edges sharp. Excessive cable capacitance rounds the edges of the clock signal, making it less clear whether it is a one or a zero. That uncertainty results in jitter.

Of course, cable has inductance too, which acts opposite from capacitance. Van der Mee says that the art of making a good digital cable is in controlling the characteristic impedance, which comes down to keeping a proper ratio between capacitance and inductance. When the two are in proper balance in a cable, the artifacts created by it approach simple delay, with less phase distortion than when one or the other predominates.

The solution to capacitance and inductance problems is to use the proper impedance cable. Even the best-quality mic cables are designed for frequencies of 1 MHz or less. That's plainly sufficient for word clock, but an AES3 signal is about 3 MHz at a 48 kHz sampling rate. At audio frequencies, you are contending more with the resistive component of cable impedance, but in the megahertz range, the reactive component comes more into play, and even supercool mic cables could have enough reactance to cause the clock signal harm.
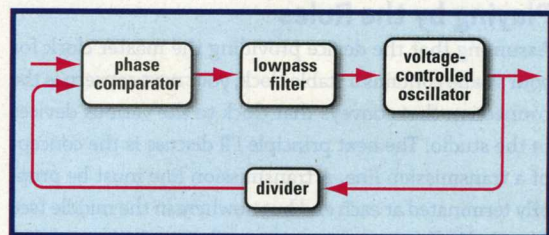

FIG. 6: Here you see the basic architecture of a phase-locked loop.

Regular hi-fi cable with RCA connectors can't handle the bandwidth of S/PDIF for the same reason. Cable aside, the RCA connectors themselves are not made to work with 75Ω lines. Also remember that good shielding is more crucial for word-clock connections, which are unbalanced, than for AES3, which is a balanced line feeding a differential receiver.

Even with the proper cable, cable runs of excessive length can result in enough losses to cause problems. Looking at an AES3 signal on an overly long cable run, for instance, you will see a flattening of the waveform and unclear transitions. The AES3 specification states that you can run cables as long as 100 meters without any equalization or special treatment, whereas S/PDIF cables have a maximum length of 10 meters, and you should generally keep BNC word-clock cables to no more than 5 meters. (However, van der Mee reports good results using cable as long as 300 meters.) Optical fiber cables should be kept to about 15 meters if they're plastic, but glass cables can run much farther—over a mile, in fact. Note that optical connections such as Lightpipe are vulnerable to jitter resulting from dispersion of light in the optical fiber, which smears the waveform edges similarly to capacitance.

Phasers Locked

Now I'll address the idiot receiver in my analogy. Even if you use a low-jitter clock source, a low-capacitance cable is in place, and the line is properly terminated at both ends, there is still the matter of how the receiving device processes the incoming clock signal.

Each device already has its own internal clock, which must be synchronized to the incoming clock in order to slave the device. In the old days, when slaved to an external word clock, some devices clocked directly off the word clock appearing at the BNC word-clock input jack. Those devices were extremely sensitive to jitter on the external clock. Today, internal clocks are synchronized to an external clock through the use of another clocking fundamental, the *phase-locked loop (PLL)*. A PLL is a servo system that compares the incoming clock to the internal clock and applies a correction signal as needed to the internal clock to match its frequency and phase to that of the incoming clock (see **Fig. 6**).

That may sound like a tricky process, and it is. There

are many ways to implement PLLs, from cheap and dirty to expensive and clean. If the receiving device has a low-quality PLL, it is an idiot and all the trouble you've gone to in getting a stable source and a low-capacitance, properly terminated line is for naught. The quality of the PLL in a digital device is hugely important.

Though PLLs can be weak links in the chain, they also have some very powerful benefits. One is that a well-designed PLL can do an excellent job of cleaning up even substantial amounts of jitter from an incoming clock (within limits, of course). Further, almost all of today's digital audio devices use sigma-delta ADCs, which generate high-bandwidth 1-bit streams that are later downsampled and converted into multibit PCM samples. Because word clock runs at the sampling rate, in order for a sigma-delta converter to sync to word clock, the frequency of the word clock must be multiplied up to the 24 MHz frequency of the sigma-delta converter's clock, called the *M clock*. This multiplication is performed by a PLL.

Given all of that, you obviously want to know how good the PLL is in any device. As with determining clock quality, a thorough analysis of jitter in a PLL is probably beyond the scope of most EM readers; however, Bryan suggests a way to get at least a reasonable take on PLL performance using regular audio-editing software with frequency analysis.

Simply record a 10 kHz sine wave at 0 dB into your ADC and perform a frequency analysis. A perfect sine wave should show energy at only one frequency, so measuring the amplitude of any sidebands you see will give you a good indication of how much jitter-induced frequency modulation is happening. In general, says Bryan, no sideband should exceed the ADC's stated signal-to-noise ratio.

Watching the Clock

Beyond that, the best strategy is to buy good equipment from companies that appear to know what they're doing.
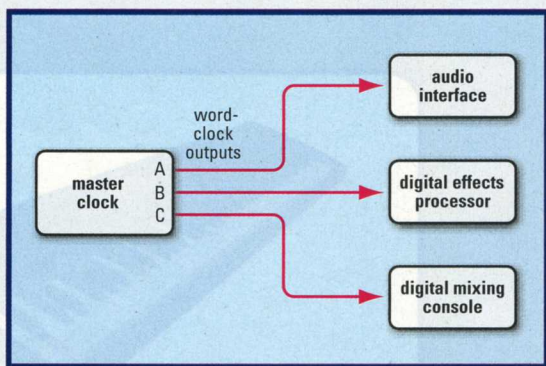

FIG. 7: To minimize transmission-line complications, a star configuration sources all clocks from a single high-quality master clock with multiple outputs. Note that proper termination must be applied at each slave device's clock input.

All the companies I spoke with for this article care about doing clocking well, and some, like Apogee Electronics and Antelope Audio, specialize in clocking devices. Many other companies, of course, are striving to make high-quality equipment with stable clocks; Mytek Digital, Benchmark Media, Lucid Audio, Rosendahl Studiotechnik, Brainstorm Electronics, and Lavry Engineering are only a few of them, and there are lots more.

The fact that such critical information is unavailable to users poses a real problem in knowing whether you're assembling a well-clocked system. But it gets even worse: there is no standard defining the voltage level or phase of a BNC word clock, and manufacturers have different ideas about how devices should behave. This lack of standards can lead to all sorts of troubleshooting nightmares that will reduce your method to just trying every configuration you can think of until, hopefully, you find one that works. It's not very scientific, but that's how it is if things don't work right. Fortunately, they more often do.

Even with the entire transmission line in place, you're not done yet. You still have to make sure that every device is "listening" for the external clock. Some devices will automatically switch to an external clock when they detect one present; they will also detect the clock's sampling rate and switch to that. Other devices require the user to change a setting manually to enable external clocking. Audio interfaces for computers often have software driver settings that must be changed before they will clock from an external source. If you change the application you are running, you may need to recheck the clocking to make sure the application did not set the driver to sync to a different clock source.

Finally, although a device set to receive an external clock should detect the sampling rate and switch its own sampling rate accordingly, not all of them do—again, including some computer audio interfaces. Assumptions will lead to trouble; be sure you know that each device is not only enabled to receive external clock, but also is set to the correct rate.

Most devices will simply mute if they cannot auto-switch to an incoming clock at a different rate than the one set internally, but some may switch to their internal clocks and not resync when the external clock reappears. As with any situation in which devices are not locked to the same clock, ugly artifacts can occur. For example, when using a master clock device, you need to be vigilant in explicitly switching its frequency every time you use a different sampling rate.

To paraphrase 19th-century abolitionist Wendell Phillips, eternal vigilance is the price of good clocking. Often, 48 kHz and 44.1 kHz are close enough in frequency that some devices set to one rate will accept a clock and run at the other without complaint. For

instance, consider a file playing at 48 kHz and recorded to an external device inadvertently set to 44.1 kHz. When played back at 44.1 kHz, all the samples would be present, but they'd play back too slowly, resulting in the pitch being lowered.

Who's in Charge?

Once you understand that one device must be the clock master, which one should it be? The answer probably depends on the context, but in all cases, it should be a good-quality, stable clock. Devices like CD or DVD players make very jittery clock sources.
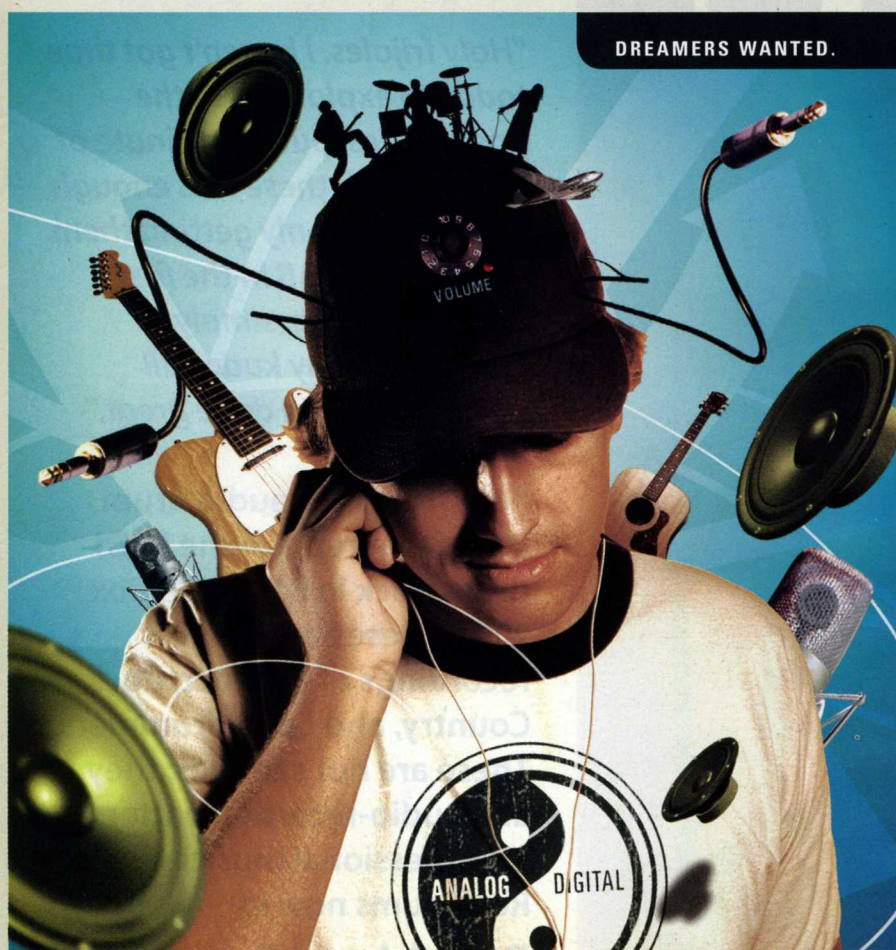
Because the most critical point for clocking is the ADC, an ideal scheme is to use a high-quality ADC's clock as the master. One of the features that make such a device high quality (and more costly) is having a good clock. In simple systems, you should be able to clock from the ADC, and that's usually a good idea.

Studios with digital mixers often use them as clock masters for a couple of reasons. When you load a project and the mixer resets to the session's sampling rate, any devices being clocked from the mixer that autodetect sampling rate will automatically change. Another reason is that some mixers just don't seem to like being clocked externally. In other studios, the DAW audio interface is used as the master to attain the same level of convenience.

Pro Tools HD systems can use a scheme called Loop Sync, in which the word-clock output from each interface in the system is connected to the word-clock input of the next, with the output of the last in the chain looping back to the input of the first. In software, any of the interfaces can be designated as the master, and the rest will slave to it, making it easier to slave to digital inputs in different formats that you have connected to the different interfaces without physically repatching.

Another very popular solution is to use a dedicated master clock device. These devices are designed to have very stable clocks, but they generally also provide multiple outputs, usually including a few different formats (AES3, word clock, S/PDIF, and so on). Further, they provide functions needed in postproduction situations, where it is necessary to resolve word clock to other incoming signals, such as timecode, AES3, or black burst.

Regardless of what device provides the clock, there is the question of how to distribute it to all the other devices. Most devices that have a word-clock input also have a word-clock output or thru. A word-clock-thru connector makes it easy to daisy-chain the clock from one device to another. Though a very simple method of clock distribution, it is potentially problematic. The first issue is that the length of the transmission line is increased. Even though you're using multiple cables, they act as one long cable as far as capacitance is concerned. Another issue is that daisy-chaining makes it critical to know whether

the word-clock input of a device in the middle of the chain is terminated; if it is, havoc will ensue.

High-quality devices from some manufacturers, such as Apogee, do not simply hardwire the clock input to a thru connector, but actually regenerate the incoming clock before sending it out. This is exemplary behavior that really improves system clocking. However, Kashiwa and van der Mee both caution that every PLL has its own jitter signature, which is added to the clock; consequently, passing a signal through several devices that each reclock could result in a troublesome accumulation of jitter.

The perils of daisy chaining are best avoided by wiring in a star configuration whenever possible (see **Fig. 7**). Master clock devices are useful in larger systems with a number of components that need clocking, because they have multiple outputs that can usually drive several units directly. If you are clocking instead from an ADC or other device, a word-clock distribution amplifier can take the master clock and generate a number of outputs to fan out to your studio's devices.

Clocking Out

Although analog audio is still alive and kicking, digital audio dominates most production today, bringing with it the need to understand a different set of basics than that in the analog world. Clocking is absolutely one of the most important elements in a digital audio system. At times the concepts can be a little slippery, but clocking is a technical underpinning of modern studios that is ignored or misunderstood at the user's peril. (For more information about clocking and a recommended reading list, see the **online bonus material** at www.emusician.com.)

The basic foundations of digital clocking are not hard to understand or implement. Once you have a good system set up and know how it works, hopefully you will notice a difference in the sound of your studio. At the very least, you should be able to eliminate (or greatly reduce) ugly clicks and pops. You may also notice greater definition in your sound.

Always keeping fundamental principles in mind will guide you through most clocking questions. The bright side is that once a well-thought-out clocking scheme is in place, it should remain stable for as long as your system setup is consistent, leaving you free to concentrate on the project at hand. **EM**

Larry the O will soon embark on a studio-building project that you may well be reading about this time next year.