In recent decades, digital audio (and its related industries) has evolved from being an infant technology that was available only to a select few to its present-day position as a primary driving force in audio production, entertainment, and communication. In fact, digital audio and media production has such an impact upon our lives that it's often an integral part of both the medium and the message within modern-day communication.

Although digital audio is a varied and complex field of study, the basic theory of how the process works isn’t really that difficult to understand. At its most elementary level, it is simply a process by which numeric representations of analog signals (in the form of voltage levels) are encoded, processed, stored, and reproduced over time through the use of a binary number system.

Just as English-speaking humans communicate by combining any of 26 letters together into groupings known as “words” and manipulate numbers using the decimal (base 10) system . . . the system of choice for a digital device is the binary (base 2) system. This numeric system provides a fast and efficient means for manipulating and storing digital data. By translating the alphabet, base 10 numbers, or other form of information into a binary form, a digital device (such as a computer or processor) can perform calculations and tasks that might otherwise be cumbersome, less cost effective, and/or downright impossible to perform in the analog domain. This binary data can be encoded in such media forms as:

- Logical 1 or 0
- On or off
- Voltage or no voltage
After the information has been recorded, stored, and/or processed, the resulting data can be converted back into an analogous form that we humans can readily understand.

Before we delve onto the basic aspects of recording, processing, and reproducing audio in the digital domain, let's take a look at how one form of information can be converted into another equivalent (analogous) form of information. For example, if we type the letters C, A, and T into a word processor, the computer quickly goes about the task of translating these keystrokes into a series of 8-bit digital words that would be represented as [0100 0011], [0100 0001], and [0101 0100]. These “alpha-bits” don’t have much meaning when examined individually; however, when placed together into a group, this data represents a four-legged animal that’s either seldom around or always underfoot (Figure 6.1). From this, we can deduce that whenever binary words are grouped together as a string of data that has an analogous and recognizable pattern, a meaningful message can be conveyed.

In a similar manner, a digital audio system works by sampling (measuring) the instantaneous voltage level of an analog signal at a single point in time . . . and then converting these samples into an encoded word that digitally represents that voltage level. By successively measuring changes in an analog signal’s voltage level (over time) . . . this stream of representative words can be stored in a form that represents the original analog signal. Once stored, the data can then be processed and reproduced ways that have changed the face of audio production forever.

**The Basics of Digital Audio**

In Chapter 2, we learned about the two most basic characteristics of sound:

- Frequency (the component of time)
- Amplitude (the signal-level component)

Digital audio can be likewise broken down into two analogous components:

- Sampling (which represents the component of time)
- Quantization (which represents the signal-level component)
Sampling

In the world of analog audio, signals are passed, recorded, stored, and reproduced as changes in voltage levels that continuously change over time (Figure 6.2). The digital recording process, on the other hand, doesn’t operate in a continuous manner; rather, digital recording takes periodic samples of a changing audio waveform (Figure 6.3) and transforms these sampled signal levels into a representative stream of binary words that can be manipulated or stored for later processing and/or reproduction.

Within a digital audio system, the sampling rate is defined as the number of measurements (samples) that are taken of an analog signal in one second. Its reciprocal (sampling time) is the elapsed time that occurs between each sampling period. For example, a sample rate of 48 kHz corresponds to a sample time of 1/48,000th of a second. Because sampling is tied directly to the component of time, the sampling rate of a system determines its overall bandwidth (Figure 6.4), meaning that a system with higher sample rates is capable of storing more frequencies at its upper limit.

Figure 6.2. An analog signal is continuous in nature.

Figure 6.3. A digital signal makes use of periodic sampling to encode information.
As you might expect, the sampling process can be likened to a photographer who takes a series of shots of an action sequence. As the number of pictures taken in a second increases, the accuracy of the captured event will likewise increase...until the resolution is so great, that you can't tell that the successive pictures have turned into a (hopefully) compelling movie.

During the sampling process (Figure 6.5), an incoming analog signal is sampled at discrete and precisely timed intervals (as determined by the sample rate). At each interval, this analog

**Figure 6.4.** Discrete time sampling. (a) Whenever the sample rate is set too low, important data between sample periods will be lost. (b) As the rate is increased, more frequency-related data can be encoded. (c) Increasing the sampling frequency further can encode the recorded signal with an even higher bandwidth range.
Figure 6.5. The sampling process. (a) The analog signal is momentarily “held” (frozen in time), while the converter goes about the process of determining the voltage level at that point in time and then converting that level into a binary-encoded word that’s numerically equivalent to the sampled level. (b) After the converter has stored the representative word into a memory medium, the sample is released and the next sample is held, as the system again goes about the task of determining the level of the next sampled voltage and so forth, and so forth, and so forth over time.

The Nyquist Theorem

According to the Nyquist theorem, in order for the desired frequency bandwidth to be faithfully encoded in the digital domain, the selected sample rate must be at least twice as high as the highest frequency to be recorded (sample rate ≥ 2 × highest frequency). Thus, an audio signal with a bandwidth of 20 kHz would require that the sampling rate be at least 40 kHz samples/second.
Figure 6.6. Using the "comparator" circuit, during each sample and hold period, the converter compares the input signal level with a given set of reference voltages (which are successively reduced in scale by one-half for each "bit") until an equivalent digital word has been determined. (a) If the signal is greater than the first reference voltage, the first bit gets a "1," and if it's lower than the first reference voltage, the first bit gets a "0." (b) If the signal is greater than the second reference voltage, the second bit gets a "1," and if it's lower than the second reference voltage, the second bit gets a "0." (c) If the signal is greater than the third reference voltage, the second bit gets a "1," and if it's lower than the second reference voltage, the second bit gets a "0." (d) These comparisons continue to zero in on the precise voltage level in smaller and smaller steps until all of the bits in the digital word (in this case, an 8-bit word) have been defined... and the process can begin again to determine the next sample level.
In addition, it’s equally important that no audio signal greater than half the sampling frequency enter into the digital conversion process. If frequencies greater than one-half the sample rate are allowed to enter into the path, erroneous frequencies—known as alias frequencies (Figure 6.8)—could enter into the audible signal as false frequencies, which might be heard as harmonic distortion.

In order to eliminate the effects of aliasing, a low-pass filter is placed before the analog-to-digital (A/D) conversion stage. In theory, an ideal filter would pass all frequencies up to the Nyquist cutoff frequency and have an infinite attenuation above this point (Figure 6.9a); however, in the real world, such a “brickwall” filter doesn’t exist. For this reason, a slightly higher sample rate must be chosen in order to account for an attenuation slope that’s required for the filter to be effective (Figure 6.9b). For example, a sample rate of 44.1 kHz is chosen in order to accurately encode an effective bandwidth up to 20 kHz.
Figure 6.8. Frequencies that enter into the digitization process above the Nyquist half-sample frequency limit could introduce harmonic distortion: (a) introduced frequencies above the limit; (b) alias frequencies that are introduced into the audio band.

Figure 6.9. Anti-alias filtering. (a) An ideal filter would have an infinite attenuation at the 20,000-Hz Nyquist cutoff frequency. (b) Real-world filters require an additional frequency "guardband" in order to fully attenuate unwanted frequencies that fall above the half-bandwidth Nyquist limit.
Oversampling

Oversampling is a process that's commonly used in professional and consumer digital audio systems to improve a Nyquist filter’s anti-aliasing characteristics. This process has the effect of further reducing intermodulation and other forms of audible distortion. Whenever oversampling is used, the effective sampling rate within a converter’s filtering block is multiplied by a specific factor (often ranging between 12 and 128 times the original rate). This significant increase in the sample rate is accomplished by interpolating sampled level points between the original sample times. In effect, this technique makes educated guesses as to what the sample levels would be and where they’d fall at the newly generated sample time and then creates an equivalent digital word at that point. This increased sample rate likewise results in a much wider frequency bandwidth (so much so that a simple, less-expensive filter can be used to cut off the frequencies above the Nyquist limit). By down-sampling the rate back to its original value after the filter block, the Nyquist filter’s bandwidth will be narrowed to such a degree that it approximates a much more complex and expensive cutoff filter.

Quantization

Quantization represents the amplitude component of the digital sampling process. It is used to translate the voltage levels of a continuous analog signal (at discrete sample points over time) into binary digits (bits) for the purpose of manipulating and/or storing audio data in the digital domain. By sampling the amplitude of an analog signal at precise intervals over time, it becomes the job of the converter to determine the exact voltage level of the signal (during the sample interval, when the voltage level is momentarily held) . . . and then output an analogous set of binary numbers (as a grouped word of n-bits length) that represents the originally sampled voltage level (Figure 6.10). The resulting word is used to encodes the original voltage level with as high a degree of accuracy as can be permitted by the word’s bit length and the system’s overall design.

Figure 6.10. The instantaneous amplitude of the incoming analog signal is broken down into a series of discrete voltage steps, which are then converted into an equivalent binary-encoded word.
Currently, the most common binary word length for audio is 16-bit (for example, [0110010100101101]); however, professional systems having 20- and 24-bit resolution are also in common use. In addition, computers and signal-processing devices are capable of performing calculations internally at the 32- and 64-bit resolution level. This added internal headroom at the bit level, helps reduce errors in level and performance at low-level resolutions, whenever multiple audio datastreams are mixed and/or processed within a digital signal processing (DSP) system. This greater internal bit resolution is used as errors are most likely to accumulate within the least-significant bits (LSBs, the final and smallest numeric value within a digital word). As multiple signals are mixed together and multiplied (a regular occurrence in gain change and processing functions), lower-bit-resolution numbers become more and more of a component of the final result. . . Since the internal bit depth is higher, these resolutions can be preserved (instead of being dropped within the hard- or software processing function) . . . with a final result being an \( n \)-bit datastream that’s relatively free of errors.

This leads us to the conclusion that greater word lengths will often directly translate into an increased resolution (and thus higher quality) due to the added number of finite steps into which a signal can be digitally encoded. The following details the number of encoding steps that are encountered for the most commonly used bit lengths:

- **8-bit word** = \( \text{nnnnnnnn} \) = 256 steps
- **16-bit word** = \( \text{nnnnnnnn nnnnnnnn} \) = 65,536 steps
- **20-bit word** = \( \text{nnnnnnnn nnnnnnnnn nnnn} \) = 1,048,576 steps
- **24-bit word** = \( \text{nnnnnnnn nnnnnnnnn nnnnnnnn} \) = 16,777,216 steps
- **32-bit word** = \( \text{nnnnnnnn nnnnnnnnn nnnnnnnnn nnnnnnnnn} \) = 4,294,967,296 steps

where \( n \) = binary 0 or 1.

**Signal-to-Error Ratio**

Although analog signals are continuous in nature, as we’ve read, the process of quantizing a signal into an equivalent digital word isn’t. Since the number of discrete steps that can be encoded within a digital word limits the accuracy of the quantization process, the representative digital word can only be an approximation (albeit an extremely close one) of the original analog signal level. A digital system’s *signal-to-error ratio* is closely akin (although not identical) to the analog concept of signal-to-noise (S/N) ratio. Whereas a S/N ratio is used to indicate the overall dynamic range of an analog system, the signal-to-error ratio of a digital audio device indicates the degree of accuracy that’s used with regard to a signal level’s accuracy and its step-related effects of quantization. Given a properly designed system, the signal-to-error ratio for a signal coded with \( n \) bits is:

\[
\text{Signal-to-error ratio} = 6n + 1.8 \text{(dB)}
\]
Therefore, the theoretical signal-to-error ratios for the most common bit rates will yield a dynamic range of:

- 8-bit word = 49.8 dB
- 16-bit word = 97.8 dB
- 20-bit word = 121.8 dB
- 24-bit word = 145.8 dB
- 32-bit word = 193.8 dB

**Dither**

When you come right down to it, the fundamental difference between digital audio and analog audio is one of resolution. Theoretically, the resolution of an analog signal is infinite in both its time and level components, whereas digital audio represents both time (sampling) and level (quantization) as discrete and quantifiable steps. Although these steps are quite small, they add a “squared off” component to a waveform that adds distortions and sonic properties that are similar to that of a square wave. This characteristic (which occurs at the digitized signal’s least significant bit level) has the audible effect of adding low levels of harmonic distortion to the encoded signal; however, by adding a small amount of noise (whose frequency spectrum has been shaped so as to be as unobtrusive and unnoticeable as possible), it’s possible to statistically improve the resolution of the conversion process below LSB level. You heard that right... by adding a small amount of random noise into the A/D path, we can actually:

- Improve the resolution of the conversion process below the least significant bit level.
- Reduce harmonic distortion in a way that greatly improves the signal’s performance.

In order to look between the LSB steps, let’s use an analogy to understand how the process works in its basic form. In Figure 6.11, the least significant bit in the encoding process can be

*Figure 6.11. Values falling below the least significant bit level cannot be encoded without the use of dither.*
seen as falling between either the “0” or “1” value. Without dither, if the actual voltage level to be sampled falls at any point between these LSB values, there will be a 50% chance that it will end up being indiscriminately encoded as either a “0” or a “1.” Whenever dither is added, the random element of noise will create a probability curve that allows the A/D circuit to detect whether the lower-level signal is closer to the least significant “0” or “1.” Without going much further into the process, suffice it to say that dither is mathematically determined to provide the most accurate results at the LSB level from a statistical perspective.

In addition to the above, it is also a basic truth that whenever a high-resolution signal is reduced in resolution a quantization error will be introduced into that signal. In short, whenever a signal is converted to a lower-resolution signal, dither should be employed in order to reduce or avoid artifacts that would otherwise be introduced whenever bits are truncated (the simple dropping of lower value bits when converting to a lower-resolution bit rate). This holds true when digitally converting or mixing between any bit-rate structure. For example:

- When internally mixing at one rate and outputting at another lower-resolution rate (hopefully, dithering is provided within the DSP mix processing function)
- When converting 24 to 20 bits, 24 to 16 bits, 20 to 16 bits, etc. (in these cases, dithering can be added via a hardware or plug-in processing module)
- Transferring an analog tape to a 24 or 16 bit rate (in this case, dithering is naturally added via the tape noise as well as the converter’s internal thermal noise)

In the following example, a system could simply drop (truncate) the least significant 8 bits within the conversion process. Alternatively, dither could be added to improve the educational guess that can be made at the new LSB level . . . thereby improving the signal resolution:

<table>
<thead>
<tr>
<th>Original 24-bit word</th>
<th>Upper 16 bits</th>
<th>Lower 8 bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>Truncated bits</td>
<td>nnnnnnnn nnnnnnn</td>
<td>Nnnnnnnn</td>
</tr>
<tr>
<td>Outputed 16-bit word</td>
<td>nnnnnnnn nnnnnnn</td>
<td>Nnnnnnnn</td>
</tr>
<tr>
<td>Original 24-bit word</td>
<td>nnnnnnnn nnnnnnn</td>
<td>Nnnnnnnn</td>
</tr>
<tr>
<td>Random dither</td>
<td>nnnnnnnn nnnnnnn</td>
<td>Dddddddd</td>
</tr>
<tr>
<td>Outputed 16-bit word</td>
<td>nnnnnnnn nnnnnnn</td>
<td></td>
</tr>
</tbody>
</table>

where n = binary 0 or 1; t = truncated bits; d = dithered bits.

The Digital Recording/Reproduction Process

The following sections provide a basic overview of the various stages that are encountered within the process of encoding analog signals into equivalent digital data (Figure 6.12a) and then converting this data back into its original analog form (Figure 6.12b).
The Recording Process

In its most basic form, the *digital recording chain* includes a low-pass filter, a sample-and-hold circuit, an analog-to-digital converter, and the circuitry for signal coding and error correction. At the input of a digital sampling system, the analog signal must be band-limited with a low-pass filter so as not to allow frequencies that are greater than half the sample rate frequency to pass into the A/D conversion circuitry. Such a stop-band (anti-aliasing) filter generally makes use of a roll-off slope at its high-frequency cutoff point (as an immediate or “brickwall” slope would introduce severe signal distortion and phase shifts). In order for the full bandwidth to be accurately encoded, the Nyquist theorem requires that a sampling rate be chosen that’s higher than twice the highest frequency to be recorded. For example, a system with a bandwidth that reaches into the 20-kHz range is often sampled at a rate of at least 44.1K samples per second.

Following the low-pass filter, a *sample-and-hold* (S/H) circuit holds and measures the analog voltage level for the duration of a single sample period, which is determined by the sample rate (i.e., less than 1/44,100th of a second). At this point, computations must be performed in order to translate the sampled level into an equivalent binary word. This step in the A/D conversion is a critical component of the digitization process, as the sampled DC voltage...
level must be accurately quantized into an equivalent digital word (to the nearest step level) in a very short period of time.

Once the sampled signal has been converted into an equivalent digital form, the data must be conditioned for further data processing and storage. This conditioning includes data coding, data modulation, and error correction. In general, the binary digits of a digital bitstream aren’t directly stored onto a recording medium as raw data; rather, data coding is used to translate the data (along with synchronization and address information) into a form that allows the data to be most efficiently and accurately stored. The most common form of digital audio data coding is pulse-code modulation, or PCM (Figure 6.13).

The density of stored information within a PCM recording and playback system is extremely high...so much so that any imperfections (such as dust, fingerprints, or scratches that might adhere to the surface of any magnetic or optical recording medium) will cause severe or irretrievable errors to be generated. In order to keep these errors within acceptable limits, several forms of error correction will be used (depending upon the media type). One method uses redundant data in the form of parity bits and check codes, in order to retrieve and/or reconstruct lost data; a second uses error correction that involves interleaving techniques (whereby data is deliberately scattered across the digital bitstream, according to a complex mathematical pattern). The latter has the effect of spreading the data over a larger surface of the recording media...thereby making the recording media less susceptible to the effects of dropouts (Figure 6.14). In fact, it’s a simple truth that without error correction, the quality of

![Figure 6.13. Pulse-code modulation.](image)

![Figure 6.14. An example of interleaved error correction.](image)
most digital audio media would be greatly reduced or (in the case of the CD) rendered almost useless.

The Reproduction Process

In many respects, the digital reproduction chain works in a manner that is complementary to the digital encoding process. Since most digital media encodes data onto media in the form of highly saturated magnetic transition states or optical reflections, the recorded data must first be reconditioned in a way that restores the digital bitstream back into its original, modulated binary state. Once this is done, the data can then be de-interleaved (reassembled) back into its original form, where it can be converted back into PCM data.

Once the signal has been reconstructed back into its original PCM form, the process of digital-to-analog (D/A) conversion can take place. Often, a stepped resistance network (sometimes called an R/2R network) is used to convert the representative words back into their analogous voltage levels within the playback phase. During a complementary sample-and-hold period, each bit within the word being converted is assigned to a leg in the network (moving from the most-significant to the least-significant bit). Each leg is designed to pass one-half the reference voltage level that can be passed by the previous step (Figure 6.15). The presence or absence of a logical “1” in each step is then used to turn on each successive voltage leg. As you might expect, summing together these voltages will yield a precise level that can be passed on to the converter’s analog output.

Following the conversion process, a final, complementary low-pass filter is inserted into the signal path. This filter is used to smooth out any nonlinear steps that are introduced by the sampling process...resulting in a waveform that will faithfully represent the originally recorded analog waveform (assuming that the circuit has been properly designed).

Digital Audio Transmission

In this digital age, it’s become increasingly common for audio data to be distributed from one device to another or throughout a connected production system in the digital domain. In this way, digital audio can be transmitted in its original numeric form and (in theory) without any

Figure 6.15. A stepped resistance network is a common device for accomplishing D/A conversion by assigning each word bit to a series of resistors that are scaled by factors of 2.
degradation throughout a connected path or system. When looking at the differences between the distribution of digital and analog audio, it should be kept in mind that, unlike its counterpart, the transmitted bandwidth of digital audio data occurs in the megahertz range; therefore, the transmission of digital audio actually has more in common with video signals than the lower bandwidth range that’s encountered with analog audio. This means that care must be exercised to ensure that impedance is more closely matched and that quick-fix solutions don’t occur (for example, using a Y-cord to split a digital signal between two recorders is a major no-no). Failure to follow these precautions could seriously degrade or deform the digital signal.

Due to these tight restrictions, several digital transmission standards have been adopted that allow digital audio data to be quickly and reliably transmitted between compliant devices. These include such protocols as:

- **AES/EBU**
- **S/PDIF**
- **MADI**
- **ADAT lightpipe**
- **TDIF**
- **mLAN**

## AES/EBU

The AES/EBU (Audio Engineering Society and the European Broadcast Union) protocol has been adopted for the purpose of transmitting digital audio between professional digital audio devices. This standard (which is most often referred to as simply an “AES” digital connection) is used to convey two channels of interleaved digital audio through a single, three-pin XLR microphone cable in a single direction. This balanced configuration connects pin 1 to the signal ground, while pins 2 and 3 are used to carry signal data. AES/EBU transmission data is low impedance in nature (typically 110Ω) and has digital burst amplitudes that range between 3 and 10 V. These combined factors allow for a maximum cable length of up to 328 feet (100 meters) at sample rates of less than 50 kHz, without encountering undue signal degradation.

Digital audio channel data and subcode information are transmitted in blocks of 192 bits that are organized into 24 words (with each being 8 bits long). Within the confines of these data blocks, two subframes are transmitted during each sample period that convey information and digital synchronization codes for both channels in an L–R–L–R... fashion. Because the data is transmitted as a self-clocking biphase code (Figure 6.16), wire polarity can be ignored, and whenever two devices are directly connected the receiving device will usually derive its reference timing clock from the digital source device.

In the late 1990s, the AES protocol was amended to include the “stereo 96k dual AES signal” protocol. This was created to address signal degradations that can occur when running
longer cable runs at sample rates above 50 kHz. In order to address the problem, the dual AES standard allows stereo rates above 50 kHz (such as 96/24) to be transmitted over two, synchronized AES cables (with one cable carrying the L information and the other carrying the R).

**S/PDIF**

The S/PDIF (Sony/Phillips Digital Interface) protocol has been widely adopted for transmitting digital audio between consumer digital audio devices and their professional counterparts (using a similar data structure). Instead of using a balanced 3-pin XLR cable, the popular S/PDIF standard has adopted the single-conductor, unbalanced phono (RCA) connector (Figure 6.17a), which conducts a nominal peak-to-peak voltage levels of 0.5 V between connected devices, with an impedance of 75 Ω. In addition to using RCA wire cable connections, S/PDIF can also be transmitted between devices using Toslink optical connection lines (Figure 6.17b), which are commonly referred to as “lightpipe” connectors.

As with the AES/EBU protocol, S/PDIF channel data and subcode information are transmitted in blocks of 192 bits consisting of 12 words that are 16 bits long. A portion of this information is reserved as a category code that provides the necessary setup information (sample rate, copy protection status, and so on) to the copy device. Another portion is set aside for transmitting audio data used to relay track indexing information (such as start ID and program ID numbers), allowing this relevant information to be digitally transferred from the master to the copy. It should be noted that the professional AES/EBU protocol isn’t capable of digitally transmitting these codes during a copy transfer.

In addition to transmitting two channels in an interleaved L–R–L–R... fashion, S/PDIF is able to communicate multichannel data between devices. Most commonly, this shows up as
a direct digital surround-sound link between a DVD player and audio receiver/amplifier playback system (via either an RCA coax or optical connection.)

**SCMS**

Initially, certain digital recording devices (such as a DAT recorder) were intended to provide consumers with a way to make high-quality recordings for their own personal use. Soon after its inception, however, for better or for worse, the recording industry began to see this new medium as a potential source of lost royalties due to home copying and piracy practices. As a result, the RIAA (Recording Industry Association of America) and the former CBS Technology Center set out to create a “copy inhibitor.” After certain failures and long industry deliberations, the result of these efforts was a process that has come to be known as the *Serial Copy Management System*, or SCMS. SCMS (pronounced “scums”) has been incorporated into many consumer digital devices in order to prohibit the unauthorized copying of digital audio at 44.1 kHz (the standard CD sample rate). This copy inhibitor does not apply to the making of analog copies, to digital copies that are made using the AES/EBU protocol, or to sample rates other than 44.1 kHz.

So, what is SCMS? Technically, it’s a digital protection flag that is encoded in byte 0 (bits 6 and 7) of the S/PDIF subcode area. This flag can have only one of three possible states:

- Status 00: No copy protection, allowing unlimited copying and subsequent dubbing
- Status 10: No more digital copies allowed
- Status 11: Allows a single copy to be made of this product but that copy cannot be copied

Suppose that we have two DAT machines that are equipped with SCMS (with one being used for playback and the other for recording). If we try to digitally copy a DAT that has a 10 SCMS status, we would simply be out of luck; however, suppose that we found a DAT that has an 11 status flag. By definition, the bitstream data would inform the copy machine that it’s OK to record the digital signal; however, the status flag on the subsequent copy tape would then be changed to a 10 flag. If at a later time we wanted to clone this DAT copy, the machine doing the second-generation copy couldn’t be placed into Record. At that point, we have two possible choices: We could record the signal using the analog ports, or a digital format converter could be used that (among other things) allows us to strip the SCMS copy protection flags from the bitstream and continue to make multigenerational copies (it should be noted that some digital audio editors are also able to strip out or reset the protection flag bits).

**MADI**

The MADI (Multichannel Audio Digital Interface) standard was jointly proposed as an AES standard by representatives of Neve, Sony, and SSL as a straightforward, clutter-free digital connection interface between multitrack devices (such as a digital tape recorder or mixing console). The format allows up to 56 channels of linearly encoded digital audio to be connected via a single 75-Ω, video-grade coaxial cable at distances of up to 120 feet (50 meters)
or at greater distances whenever a fiberoptic interface/cable is used. MADI makes use of a serial data transmission format that’s compatible with the AES/ EBU twin-channel protocol (whereby the data, Status, User, and parity bit structure is preserved) . . . and sequentially cycles through each channel (starting with Ch. 0 and ending with Ch. 55). The transmission rate of 100 Mbit/second provides for an overall bandwidth that’s capable of handling audio data and numerous sync codes at various sample rate speeds (including allowances for changing pitch either up or down by 12.5% at rates between 32 and 48 kHz).

**ADAT Lightpipe**

A wide range of modular digital multitrack recorders, soundcards, and hardware devices use the Alesis lightpipe system for transmitting multichannel audio via a standardized optical cable link. Lightpipe connections make use of standard Toslink connectors and cables to transmit up to eight digital audio channels over a sequential, optical bitstream. Although these connectors are identical to those that are used to optically transmit S/PDIF stereo digital audio, the datastreams are incompatible with each other. Lightpipe data is not bidirectional in that it can only travel from a single source to a destination in one direction; therefore, two cables are needed to distribute data both to and from a device. Only digital audio data is transmitted over the serial bitstream, meaning that sync data will not be passed. Should sync be needed to lock one or more devices together (as in a multiple ADAT system or ADAT/DAW setup), separate sync cabling (often 9-pin serial connectors) will be required.

**TDIF**

The TDIF (Tascam Digital InterFace) is a proprietary format that uses a 25-pin D-sub cable to transmit and/or receive up to eight channels of digital audio between compatible devices. Unlike the lightpipe connection, TDIF is a bidirectional connection, meaning that only one cable is required to connect the 8 ins and outs of one device to another. Although systems that support TDIF-1 cannot send and receive sync information (a separate wordclock connection is required for that), the newer TDIF-2 protocol is capable of receiving and transmitting sync through the existing connection, without any additional cabling.

**mLAN**

One of the major problems that faces the professional and project studio is the vast number of interconnections that must be made between devices within a connected setup. Given that audio, MIDI, and sync are often routed separately within a production system, the mess can often result in a complex pile of spaghetti that’s not only cumbersome and complex . . . but can also be unreliable and extremely difficult to reconfigure. Although several proprietary network interconnections are beginning to solve this problem in the digital domain, an innovative system from Yamaha (known as mLAN) seems to be leading the pack by allowing multichannel digital audio and MIDI music data to be transferred via a single IEEE
1394/FireWire® or iLINK® cable (Figure 6.18). Theoretically, mLAN can transfer approximately 100 channels of digital audio data and up to 256 ports of MIDI data (16 channels × 256 connections) between digital audio workstations (DAWs), digital mixers, synths, and effects processors over a single cable in a bidirectional fashion... at speeds of 100, 200, or 400 Mbps!

All of this means that a mLAN-based music system could be quickly and easily configured, saved, and recalled with none of the frustration and down time that occur when reconfiguring a conventional system. Using the patch bay application that’s provided with all mLAN products, devices can be routed and reconfigured within software. Since it’s “hot pluggable,” devices (called “nodes”) can be plugged and unplugged without having to power-down or reset the system. In addition, mLAN is able to transmit and resolve world clock issues... even allowing devices to run at differing sample rates on the same network. Finally, this format can run with or without a computer. Should multiple mLAN hardware devices be connected, the system will establish a network link for easy connectivity.

Although Yamaha developed mLAN, it is available to other manufacturers on a royalty-free (no-cost) basis. As of this writing, over 40 manufacturers have signed on as mLAN licensees, and eight manufacturers have developed first-generation products.

### Signal Distribution

If copies are to be made from a single, digital audio source or if data is to be distributed throughout a connected network using AES/EBU, S/PDIF, and MADI digital transmission cables... it is possible to distribute the data from one device to the next in a straightforward, daisy-chain fashion (Figure 6.19). This method works well only if a few devices are to be...
chained together. If several devices are to be connected together in the system, time-base errors (known as jitter) might be introduced into the path, with the possible side effects being added noise and a slightly “blurred” signal image. One way to reduce such potential time-base errors is to use a digital audio distribution device that can route the data from a single digital audio source to a number of individual device destinations (Figure 6.20).

What Is Jitter?

Jitter is a controversial and widely misunderstood phenomenon. To my knowledge, it’s been explained best by Bob Katz of Digital Domain (www.digido.com; Orlando, Florida). The following is a brief excerpt of his article “Everything You Always Wanted To Know About Jitter But Were Afraid To Ask.” Further reading on digital audio and mastering techniques can be found in Bob’s excellent book, Mastering Audio: The Art and the Science, from Focal Press (www.focalpress.com):

Jitter is time-base error. . . . It is caused by varying time delays in the circuit paths from component to component in the signal path. The two most common causes of jitter are poorly designed Phase Locked Loops (PLLs) and waveform distortion due to mismatched impedances and/or reflections in the signal path.
Here is how waveform distortion can cause time-base distortion: The top waveform (Figure 6.21a) represents a theoretically perfect digital signal. Its value is 101010, occurring at equal slices of time, represented by the equally spaced dashed vertical lines. When the first waveform passes through long cables of incorrect impedance, or when a source impedance is incorrectly matched at the load, the square wave can become rounded, fast rise times become slow, also reflections in the cable can cause misinterpretation of the actual zero crossing point of the waveform. The second waveform (Figure 6.21b) shows some of the ways the first might change; depending on the severity of the mismatch you might see a triangle wave, a squarewave with ringing, or simply rounded edges. Note that the new transitions (measured at the Zero Line) in the second waveform occur at unequal slices of time. Even so, the numeric interpretation of the second waveform is still 101010! There would have to be very severe waveform distortion for the value of the new waveform to be misinterpreted, which usually shows up as audible errors—clicks or tics in the sound. If you hear tics, then you really have something to worry about.

If the numeric value of the waveform is unchanged, why should we be concerned? Let’s rephrase the question: “When (not why) should we become concerned?” The answer is “hardly ever.” The only effect of time-base distortion is in the listening; as far as it can be proved, it has no effect on the dubbing of tapes or any digital-to-digital transfer (as long as the jitter is low enough to permit the data to be read; high jitter may result in clicks or glitches as the circuit cuts in and out). A typical D to A converter derives its system clock (the clock that controls the sample and hold circuit) from the incoming digital signal. If that clock is not stable, then the conversions from digital to analog will not occur at the correct moments in time. The audible effect of this jitter is a possible loss of low-level resolution caused by added noise, spurious (phantom) tones, or distortion added to the signal.

A properly dithered 16-bit recording can have over 120 dB of dynamic range; a D to A converter with a jittery clock can deteriorate the audible dynamic range to 100 dB or less, depending on the severity of the jitter. I have performed listening experiments on purist, audiophile-quality musical source material recorded with a 20-bit accurate A/D converter (dithered to 16 bits within the A/D). The sonic results of passing this signal through
processors that truncate the signal at $-110$, $-105$, or $-96$ dB are increased “grain” in the image; instruments losing their sharp edges and focus; reduced sound-stage width; apparent loss of level causing the listener to want to turn up the monitor level, even though high-level signals are reproduced at unity gain. Contrary to intuition, you can hear these effects without having to turn up the listening volume beyond normal (illustrating that low-level ambience cues are very important to the quality of reproduction). Similar degradation has been observed when jitter is present. Nevertheless, the loss due to jitter is subtle and primarily audible with the highest-grade audiophile D/A converters.

**Wordclock**

One aspect of digital audio recording that never seems to get enough attention is the need for synchronization at the sample level within a series of interconnected digital audio devices. In order to reduce such gremlins as clicks, pops, and jitter (oh my!), it’s often necessary to lock the overall sample rate timing to a single master clock signal (so that the conversion sample and hold states for all digital audio channels and devices will occur at exactly the same point in time) ...through the use of a single timing reference known as *wordclock*.

As an example, let’s assume that we’re in a room that has four or five clocks, and none of them reads the same time! In places like this, you never quite know what the time really is... the clocks could be running at different speeds or at the same speed but are ticking at different times. Basically, trying to accurately keep track of the time while simultaneously looking at all of the clocks would end up being a jumbled nightmare. On the other hand, if all of these clocks were locked to a single, master clock (remember those self-correcting clocks that have been installed in most schools?)... keeping track of the time (even when moving from room to room) would be much simpler.

In effect, wordclock works in a similar fashion. If the sample clock (the timing reference that determines the sample rate and DSP traffic control) for each device was set to operate in a freewheeling, internal fashion... the timing references of each device within the connected digital audio chain wouldn’t accurately match up. Even though the devices are all running at the same sample rate, these resulting mismatches in time will often result in clicks, ticks, excessive jitter, and other unwanted grunge. In order to correct for this, the internal clocks of all the digital devices within a connected chain must be referenced to a single “master” wordclock timing element (Figure 6.22).

Similar to the distribution of time code, there can only be one master wordclock reference within a connected digital distribution network. This reference source can be derived from a digital mixer, soundcard... or any desired source that can transmit wordclock. Often, this reference pulse is chained between the involved devices through the use of BNC and/or RCA connectors, using low-capacitance cables (often 75-$\Omega$, video-grade coax cable is used, although this cable grade isn’t always necessary on shorter cable runs).

It’s interesting to note that wordclock isn’t generally needed when making a digital copy from one device to another (via such protocols as AES, S/PDIF, MADI, or TDIF2), as the
Timing information is actually embedded within the data bitstream itself. Only when we begin to connect devices that share and communicate digital data will we see the immediate need for wordclock.

It almost goes without saying that there will often be differences in connections and parameter setups, from one system to the next. In addition to proper cabling and impedance termination considerations throughout the network, specific hardware and software setups may be required in order to get all the device blocks to communicate. In order to better understand your particular system’s setup (and to keep frustration to a minimum), it’s always a good idea to keep all of your device manuals close at hand.

### Digital Audio Recording Systems

For the remainder of this section, we’ll be looking at the various types of digital audio recording devices that are currently available on the market. From my own personal viewpoint, not only do I find the here and now of recording technology to be exciting and full of cost-effective possibilities... I also love the fact that there are lots of recording media and device-type options. In other words, a digital hard- or software system that works really well for me might not be the best and easiest solution for you! As we take an in-depth look at many of these device and system choices, I hope that you’ll take the time to learn about each one (and possibly even try your hand at listening to and/or working with each system type). In the earlier years of recording, there were only a few ways that a successful recording could be made. Now, in the age of technological options... your mission (should you decide to accept it) is...
to research and test-drive devices and/or production systems to find the one that best suits your needs, budget, and personal working style.

The Fixed-Head Digital Audio Recorder

The fixed-head digital audio recorder is a reel-to-reel system that often emulates its analog counterpart in form and function, although most other similarities end there. These recording systems use digital audio conversion and special data encoding structures to store digital audio data onto specially formulated digital audio tape, using state-of-the-art, thin-film heads (Figure 6.23). Although two-channel fixed-head recorders do exist, the vast majority of these systems are multitrack. Of these, the most commonly found recorders are 24- and 48-track recorders that use the Digital Audio Stationary Head (DASH) format, which was ratified and jointly established by Sony, Willi Studer AG, and Matsushita Electric Industries.

The DASH Format

DASH was developed to help ensure standardization between different generations and manufacturers of digital, fixed-head recorders. The standard provides for three data storage densities (fast, medium, and slow), with the choice being determined by the tape speed of the recorder. Using this system, data isn’t encoded onto a single track but actually spread over several interleaved data tracks in order to achieve the high data densities that are required to record digital audio onto longitudinal tape.

Error Correction

The operation of a DASH encoder is based on the cross interleave code (CIC), with increased interleaving being made between the even- and odd-numbered words (allowing up to three
consecutive words to be corrected). This interleaving makes it possible for the tape to be spliced, thereby allowing physical tape edits to be made. The correctability of burst (large-scale) errors is determined by the encoders and is the same for all three-speed versions. Error correction encoding and decoding are done independently for each track. For example, if excessive errors appear on one of the recorded tracks (as might occur with a dropout), the correction capabilities on other tracks won’t be affected (a feature that safeguards the program material under adverse conditions). In addition to allowing the tape to be physically spliced (using a corrective cross-fade function), the tracks on a DASH tape can be punched in and out and electronically edited (using a special edit control system).

**DASH Recording Systems**

Recorders using the DASH format are commonly available in 24- and 48-track configurations. With the development of new large-scale integrated circuits (LSIs), smaller and lighter recorders are available that are lower in cost and power consumption than their first-generation models. Both Sony and Studer have developed a series of popular multichannel DASH recorders. The Sony 24-track PCM-3324S and the Studer D 827 Mark II MCH use 1/2-inch tape that runs at a speed of 30 ips and operate using 16- or 24-bit wordlengths at sample rates of 48, 44.1, and 44.056 kHz. Using thin-film head technology (a technology borrowed from integrated circuit fabrication, which is detailed in Figure 6.24), the digital tracks are longitudinally spread across the width of the tape, along with two outside analog tracks and an additional control or external data track. The error detection circuitry of both of these recorders allows splice edits to be made by creating a seamless 90° cross-fade that interpolates the data before and after the splice.

*Figure 6.24. Headblock mechanism of the Sony 3324A. (Courtesy of Sony Professional Audio, www.sony.com/proaudio.)*
Both the Sony PCM-3348 (Figure 6.25) and the Studer D 827 Mark II MCH (Figure 6.26) DASH recorders have the added bonus of being able to record 48 tracks of digital audio that's fully compatible with the 24-track DASH format. Using a unique system, tapes that were previously recorded on a 24-track DASH machine can be recorded and reproduced with no signal degradation using tracks 1 to 24, with tracks 25 to 48 being available on the same tape with no problems in compatibility.

**The Rotating-Head Digital Audio Recorder**

*Rotating-head digital recorders* fall into several categories, including DAT (digital audio tape) systems and MDM (modular digital multitrack) recording systems.
The Rotary Head

Because of the tremendous amount of data density that’s required to record/reproduce PCM digital audio (approximately 2.77 million bits/second), the recording of data onto a single, linear tape track is extremely impractical. In order to get around this bandwidth limitation, when using a narrow-width tape, a rotating-head helical scan path is used to effectively increase the overall head-to-tape contact area at a given tape speed. This process (which is also used in VHS videotape transport technology) creates a slanted tape path that wraps around a rotating drum that has two or more magnetic heads built into its scan surface. As the head rotates... numerous closely spaced tape “scan” paths are consecutively recorded along the tape path at a slanted angle. These repeated scans result in an effective recorded track that’s much longer in length than would be possible with a linear track... resulting in a system that can record wider frequency bandwidths at a low physical tape speed. Examples of a helical transport that employ rotary head technology are shown in Figure 6.27.

Digital Audio Tape (DAT) System

The digital audio tape, or DAT recorder, is a compact, dedicated PCM digital audio format (Figure 6.28) that displays a wide dynamic range, low distortion, and virtually immeasurable amounts of wow and flutter. Given the fact that the DAT recorder was initially designed for the consumer market, its professional specs have resulted in its being widely adopted in

---

Figure 6.27. Helical scan path: (a) tape path around drum mechanism; (b) recording of slanted tracks.
numerous studio and on-location applications. DAT technology makes use of an enclosed compact cassette that’s even smaller than a compact audio cassette. Equipped with both analog and digital input/outputs, the DAT format can record and play back at three standard sampling frequencies: 32, 44.1, and 48 kHz (although sample-rate capabilities and system features may vary from one recorder to the next). Current DAT tapes offer running times of up to 2 hours when sampling at 44.1 and 48 kHz, and the standard is capable of three record/reproduce modes for the 32-kHz (broadcast) sampling rate:

- Option 1 provides 2 hours of maximum recording time with 16-bit linear quantization.
- Option 2 provides up to 4 hours of recording time with 12-bit nonlinear quantization.
- Option 3 (which is rarely, if ever, used) allows for the recording of four-channel, nonlinear, 12-bit audio.

**DAT Tape/Transport Format**

The actual track width of the DAT format’s helical scan can range downward to about 1/10th the thickness of a human hair (which allows for a recording density of 114 million bits per square inch . . . the first time such a density has been achieved using magnetic media). To assist with tape tracking and to maintain the highest quality playback signal, a sophisticated tracking correction system is used to center the heads directly over the tape’s scan path. The head assembly of a DAT recorder uses a 90° half-wrap tape path. Figure 6.29 shows a two-head assembly in which each head is in direct contact with the tape 25% of the time. Such a discontinuous signal requires that the digital data be communicated to the heads in digital “bursts,” which necessitates the use of a digital buffer. On playback, these bursts are again smoothed out into a continuous datastream. This encoding method has the following advantages:

- Only a short length of tape is in contact with the drum at one time. This reduces tape damage and allows high-speed searches to be performed while the tape is in contact with the head.
- A low tape tension can be used to ensure a longer head life.
The high-speed search function (approaching 300 times normal speed) is a key feature of the DAT format. In addition to this, the format makes provisions for non-audio information to be written directly into the digital stream’s subcode area. This subcode area serves as a digital identifier (just as a compact disc uses subcodes for selection and timing information) and can be written as any one of three data types:

- Start ID, which is used to indicate the beginning of a selection
- Skip ID, which indicates that a selection should be skipped over
- Program number, which makes it easier to identify selections when searching at high speeds

This subcode information can be written, erased, and rewritten at any time without altering the audio program. In addition, the subcode area can be used to encode SMPTE time code for use with audio for film or video, as well as for synchronous music production.

The Modular Digital Multitrack (MDM)

The device that single-handedly ushered professional and project studio production out of the analog age and into affordable digital multitrack audio is the modular digital multitrack system, or MDM (Figure 6.30). MDMs are compact multitrack digital audio recorders that are capable of recording eight tracks of digital audio onto videotape-grade cassette cartridges that can often be bought at your favorite neighborhood drugstore. These recorders are said to be “modular” because several of them can be linked together (in groups of eight) in a synchronous fashion that allows them to work in tandem as a large-scale multitrack recording system (often being configured in the 24- and 32-track range). Another important aspect of the MDM revolution was their price. Given the fact that a tape-based digital multitrack (i.e., DASH) could easily sell for more than $60,000, it’s not hard to understand why the (then) average price of $2000 sparked a revolution in production technology.

(Note: Just as video killed the radio star . . . DAWs have put a serious dent in the MDM market. As a result, these devices can be had for a virtual song on the used market. Be wary when...
considering a used MDM, as parts are now harder to get for certain machines and replacing a worn head will often cost more than what you paid for the unit.)

**ADAT MDM Format**

The ADAT standard, which was created by the Alesis Corporation (Figure 6.31) and is no longer available on the new market, is a family of rotary-head, eight-track modular digital multitrack recorders that use standard S-VHS videotape. This format includes such features as 16-/20-bit wordlength capabilities, professional sample-rate standards (including varispeed over a wide range), autolocate functions, and autoloop/rehearse functions (that work in both the play and record punch modes). When using a standard 120-minute S-VHS tape at the highest sampling rate, a recorder that conforms to the ADAT standard can yield recording times of slightly over 40 minutes (or 53 minutes when a 160-minute tape is used).

Analog input/output (I/O) connections vary between models and can include 1/4-inch unbalanced/ balanced jacks and RCA or XLR connections (in either −10 or +4 dBm configurations). Many models also allow multiple I/O connections to be made via a single, standard 56-pin Elco connector at professional +4-dBm reference levels.

Remote ADAT functions are carried out through the use of the LRC (little remote control), BRC (big remote control), or (in certain cases) MIDI machine control. The LRC was shipped...
with every new ADAT and contains all the device’s basic front panel transport controls on a single, palm-sized remote. The optional BRC (Figure 6.32) is capable of acting as a full-featured remote control, digital editor, expanded-feature autolocator, and synchronizer on a single tabletop or free-standing surface. It’s capable of controlling up to 16 remote ADAT units and can bounce data from one track to another in the digital domain (while also allowing tracks to be shifted in location). An example of track location shifting is its ability to copy a set of background vocals in a song’s bridge to another point near the end of the song, without the use of a DAW. All that’s needed is to define a track and location source, as well as its destination, and the BRC will take care of all transport, record, and audio functions.

Digital I/O is transmitted (in a single direction) through the use of a Toslink fiberoptic cable line that’s capable of simultaneously carrying all eight channels. This link is used to connect an ADAT to a growing number of digital peripheral I/O devices (such as a digital audio interface, mixer, effects processor, and even certain synthesizers).

On the sync front, since only audio data passes through the optical Toslink cable, it’s necessary to interconnect all of the devices together by chaining the sync ports in an in/thru fashion using a 9-pin sync cable. These synchronization ports are used to transmit a proprietary code for interlocking the devices together, with near sample accuracy. From a user’s perspective, this code will be displayed as SMPTE and, through the use of a special conversion device or system that includes an ADAT sync interface, this code can often be easily converted to/from SMPTE and/or MIDI time code.

**DTRS MDM Format**

Although the rotary-head Digital Tape Recording System (DTRS) was created by Tascam (Figure 6.33), several manufacturers offer MDM recorder systems that adhere to this standard. These 8-track modular recorders are capable of recording up to 108 minutes of digital audio onto a standard 120-minute Hi-8-mm video tape, and (like the ADAT format) can be combined with other DTRS recorders to create a system that has 24, 32, or more tracks. Digital I/O is made through the proprietary TDIF connection, which uses a special...
25-pin D-sub connector to link to other DRTS recorders, digital mixers, hard disk recorders, or external accessories. The TDIF digital interconnection comes in two flavors:

- **TDIF-1**, which is capable of transmitting and receiving all eight channels of digital audio (in a bidirectional I/O fashion) over a single cable
- **TDIF-2**, which, in addition to the features of TDIF-1, is also capable of transmitting sync without the need for an external sync connection

As with all MDMs, the sync capabilities of a DRTS recorder will vary between models and manufacturers. Like the ADAT, systems with TDIF-1 must be synced together through the use of a 15-pin D-sub cable, which will need to be chained between the various devices. This proprietary sync code can also be converted to MTC or SMPTE time code, when using conversion boxes or other special interface options. Autolocators (such as the Tascam RD-848) can be used to remote control many of the basic transport, track arming, and autolocation functions from a single desktop surface.

### Third-Party Developments and Accessories for the MDM

Years after MDMs were introduced, an ever-growing number of accessories still pop up on the market for both the ADAT and DTRS MDM formats. These accessories can be used to link the digital audio and possibly the synchronization capabilities to an external device in order to interface and/or synchronize an MDM system to a host of possible applications. For example, it’s not uncommon to find a computer-based, multichannel digital audio interface that has ADAT and/or TDIF digital I/O ports. Often, these ports are a cost-effective way to add on extra I/O channels to a host DAW system. When using a system in this manner, be sure that the MDM’s wordclock is set to follow the interface (or whatever device is set to generate the master wordclock).

The list of manufacturers and third-party companies is far too large and varied to provide here, as the number of new and innovative applications and accessories for integrating these devices into present-day systems continues to grow. The best way to keep on top of recent developments is to contact the manufacturers, search the Web, and keep reading the trade magazines.
MiniDiscs

Since its introduction by Sony as a consumer device in 1992, the MiniDisc (MD) has actually grown in popularity as a medium for recording and storing CD, MP3 tracks, and original recordings (Figure 6.34). Based on established rewritable magneto-optical (MO) technology, the 64-mm disc system has a standard recording capacity of up to 74 minutes (effectively the same record/play time as the CD, at one-quarter the size). This extended record/play time is due to a compression codec known as Adaptive Transform Acoustic Coding (ATRAC), which makes use of an encoding scheme that reduces data by eliminating or reducing data that has been deemed to be imperceptible by the average listener (in much the manner as the MP3 and WMA). Probably the biggest selling point of the MiniDisc recording medium is its portability (Figure 6.35). Units that are small enough to fit in the palm of your hand are being used to capture song ideas, lectures, instrumental samplefiles, and even live bootleg concerts. These and other advantages help to keep the MiniDisc in the pockets of many music and recording professionals.

Figure 6.34. Tascam MD 801R mkII MiniDisc recorder. (Courtesy of Tascam, www.tascam.com.)

Figure 6.35. Portable MiniDisc recorder.
Hard-Disk Recording

Over the history of disk- and disc-based systems, the style, form, and function of hard disk recording have changed to meet the challenges of faster processors, bigger drives, improved hardware systems, and the ongoing push of marketing forces to sell, sell, sell! As a result, there are numerous hard-disk system types that are designed for various purposes, budgets, and production styles. As new technologies and programming techniques continue to turn out new products on a staggeringly regular basis, many of the long-held limitations and distinctions between system types and even operating systems have gone by the wayside or have been blurred. Although the field of hard-disk recording has matured into a technology that’s become pervasive in all forms of media production, it’s still in a continual form of evolution as hardware, software, and personal working styles change. As with all evolutionary revolutions, it’s always a good idea to keep abreast of these changes by reading the trade magazines, searching the Web, and keeping your eyes and ears open. For the remainder of this chapter, we’ll be looking at the various types of hard disk systems that are currently available on the market . . . as well as their basic principles of operation.

Advantages of Hard-Disk Recording

With the advent of the drum machine (the first practical digital playback system) . . . it was soon discovered that advances in digital and semiconductor technology were opening the doors for recording audio as longer and longer samplefiles into random access memory. It wasn’t long after this that computer technology advanced to the point of being affordable to the average user . . . and through the use of specialized hardware, software, and I/O interfacing, digital audio could be recorded to, edited on, and played back from a computer’s hard disk. Thus, the concept of the hard-disk recorder was born. As most people are aware, there are numerous advantages to using a hard-disk recording system in an audio production environment:

◆ The ability to handle long samplefiles—Hard-disk recording time is often limited only by the size of the disk itself.
◆ Random-access editing—Once audio (or any type of data) is recorded onto a disk, any point within the program can be instantly accessed at any time, regardless of the order in which it was recorded.
◆ Nondestructive editing—This process allows audio segments (often called regions) to be placed back in any context and/or order within a program without changing or affecting the originally recorded soundfile in any way. Once edited, these edited tracks and/or segments can be reproduced to create a single, cohesive program.
◆ DSP—Digital signal processing can be performed on a soundfile and/or segment in either real time or non-real time (often in a nondestructive fashion).

Add to this the fact that computer-based digital audio devices can integrate many of the tasks that are related to both digital audio and MIDI production in a unified fashion.
that's often easy to use, cost effective, and time effective...and you have a system that offers
the artist and engineer an unprecedented degree of production power.

Hard-Disk Multitrack Recorders

Once the modular digital multitrack (MDM) came onto the production scene, the expen-
sive analog multitrack and more expensive digital multitrack recorder became less and less of
an option for the average project and professional studio budget. As technology marched on,
many professionals have chosen not to produce audio using either a tape-based recording
system or DAW...but have instead opted in favor of a multitrack hard-disk recorder
(Figures 6.36 and 6.37).

Unlike software DAWs that use the graphic user interface (GUI) of a personal computer, these
dedicated hardware systems mimic the basic transport, operational, and remote controls of
a traditional multitrack recorder. Often incorporating one or more removable hard-drive
bays and file compatibility with existing multitrack DAW software...the basic allure of such
a system (which is often designed around a PC motherboard and hard drive) is the fact that
it offers a simple, dedicated multitrack hardware interface that's capable of providing the speed,
flexibility, and connectivity benefits of a hard-disk recorder.

Hard-Disk and Flash Memory Portable Studios

Another type of dedicated hardware recording system is the modern-day portable studio that's
capable of recording to hard disk, MD, or solid-state flash memory cards. These all-in-one

Figure 6.36. Tascam MX-2424 PB
multitrack hard-disk recorder. (Courtesy
of Tascam, www.tascam.com.)

Figure 6.37. Mackie HDR 24/96
multitrack hard disk recorder.
(Courtesy of Loud Technologies, Inc.,
www.mackie.com.)
systems include all of the hardware and the control system’s interface to record, edit, mix down, and play back an on-the-spot composition… virtually anywhere when using an AC adapter or batteries (Figure 6.38). These systems, which range in size, features, and track offerings, are often the system of convenience for musicians, as they include all of the necessary mic preamps, mixing surface controls, effects, and integrated CD burning capabilities. In certain cases, these devices can actually get so small as to become a virtual studio in a pocket… offering all of the recording and overdub features that you might expect, plus surprising ones like flash memory, USB data transfer, effects, built-in stereo mic, and built-in rhythm machine (Figure 6.39).

The Digital Audio Workstation

In recent years, the term digital audio workstation (DAW) has increasingly come to signify an integrated computer-based hard disk recording system that commonly offers such
features as:

- Advanced multitrack recording, editing, and mixdown capabilities
- MIDI sequencing, edit, and score capabilities
- Integrated and plug-in signal processing support
- Support for integrating software plug-in instruments (VSTi) and/or peripheral music programs (ReWire)
- Integration of peripheral hardware devices such as controllers and audio and MIDI interface devices

Truth of the matter is...by offering a staggering amount of production power for the buck these software-based programs (Figures 6.40 and 6.41) and their peripherally connected devices have revolutionized the faces of professional, project, and personal studios in a way that touches almost every life within the audio and music production communities.

Integration Now...Integration Forever!

Throughout the history of music and audio production, we’ve become used to the idea that certain devices were only meant to perform a single task: A recorder records and plays back, a limiter limits, and a mixer mixes. Fortunately, the age of the microprocessor has totally broken down these traditional lines...in a way that has created a breed of digital chameleons that can change their functional colors to match the necessary task at hand. Along these same lines, the digital audio workstation isn’t so much a device as a systems concept that can perform a wide range of multichannel audio production tasks with ease and speed. Some of the characteristics that can (or should be) displayed by a DAW include:

- *Integration*—One of the major functions of a workstation is its ability to provide centralized control over the digital audio recording, editing, processing, and signal...
routing functions, as well as to provide transport and/or time-based control over MIDI/electronic music systems, external tape machines, and video recorders.

- **Communication**—A DAW should be able to communicate and distribute pertinent audio-, MIDI-, and automation-related data throughout the connected network system. Digital timing (wordclock) and synchronization (SMPTE time code and/or MTC) should also be supported.

- **Speed and flexibility**—These are probably a workstation’s greatest assets. After you become familiar with a particular system, most production tasks can be tackled in far less time than would be required using similar analog equipment. Many of the extensive signal processing, automation, and system’s communications features would simply be next to impossible to accomplish in the analog domain.

- **Automation**—Because all of the functions are in the digital domain, the ability to instantly recall a session and to undo a performed action becomes a relatively simple matter.

- **Expandability**—Most DAWs are able to integrate new and important hardware and/or software components into the system with little or no difficulty.

- **User-friendly operation**—An important element of a digital audio workstation is its ability to communicate with its central interface unit…you! The operation of a workstation...
should be relatively intuitive and shouldn’t obstruct the creative process by speaking “computerese.”

From the above, I’m sure that you’ve gathered that a software system (and its associated hardware) that’s capable of integrating audio, video, and MIDI under a single, multifunctional umbrella can be a major investment, both in financial terms and in the time that’s spent to learn and master the overall program environment. When choosing a system for yourself or your facility, be sure to take the above considerations into account. Each system has its own strengths, weaknesses, and particular ways of working. When in doubt, it’s always a good idea to research the system as much as possible before committing to it. Feel free to contact your local dealer for a salesroom test drive. As with a new car, purchasing a digital audio workstation can be an expensive proposition that you’ll probably have to live with for a while. Once you’ve taken the time to make the right choice…you can get down to the business of making music.

DAW Hardware

In keeping step with the modern-day truism “technology marches on”…the hardware and software specs of a computer and the connected peripherals continue to change at an ever-increasing pace. This is usually reflected as general improvements in such areas as their:

♦ Need for speed
♦ Increased computing power
♦ Increased disk and RAM memory and speed
♦ Operating system (OS) and peripheral integration
♦ General connectivity (networking and the Web)

In this section, we’ll be taking a look at many of the hardware devices and connected peripheral devices that help to make a DAW work.

The Desktop Computer

Desktop computers are often (but not always) too large and cumbersome to lug around. As a result, these systems are most often found as a permanent install in the professional, project, and home studio (Figures 6.42 through 6.44) One of the most commonly asked questions is “Which one…Mac or PC?” The actual answer to what OS to invest in actually depends upon:

♦ Your preference
♦ Your needs
♦ The kind of software you currently have
♦ The kind of computer platform and software your working associates have
Truth of the matter is, beyond these important questions, the choice it strictly up to you. Once you’ve decided which side of the platform tracks that you’d like to live on, the more important questions that you should be asking are:

- Is my computer fast and powerful enough for the tasks at hand?
- Does it have enough hard disks that are large and fast enough for my needs?
Is there enough RAM memory?

Do I have enough monitor space (real estate) to see the important things at a glance?

On the “need for speed” front, it’s always a good idea to buy (or build) a computer at the top of its performance range at any given time. Keeping in mind that technology marches on, the last thing that you’ll want to do is buy a new computer only to soon find out that it’s underpowered for the tasks ahead.

There’s never been a better time for choices on the hard-disk front. With today’s faster and higher capacity IDE, serial ATA, and SCSI drives, it’s a simple matter to install cost-effective drives, each with a capacity of hundreds of gigabytes. In addition, portable drive cases (Figure 6.45) can be plugged into either a FireWire or USB2 port (and in some cases both) . . . making it easy to take your own personal drive with you into the studio.

Speed can be an issue when buying a hard drive for either audio or video applications. The speed at which the disc platters turn will often affect a drive’s access time. Modern drives that “spin” at 7200 or higher are often preferable . . . and it should be noted that production drives often work best when connected directly to the system’s motherboard. On-board buffer memory can also be helpful in transferring data and in freeing up the system for other processing functions. It should be mentioned that the portable FireWire and USB2 drives mentioned above will often have reduced access times (over their internal counterparts), often making them a better medium for backing up data . . . although, with the introduction of FireWire 800 and the onward march of technology, this could change at a moment’s notice.

Regarding random access memory . . . it’s always a good idea to use as much (and as fast) RAM as you can muster. If a system doesn’t have enough RAM, data will often have to be “swapped” to the system’s hard drive . . . often slowing things down and affecting overall performance. When dealing with video and digital images, having a sufficient amount of RAM becomes even more of an issue.

Just like there never seems to be enough space around the house or apartment, having a single, undersized monitor can leave you feeling cramped for visual “real estate.” For starters,
a sufficiently sized monitor (either LCD or CRT) that’s capable of working at higher resolutions will greatly increase the size of your visual desktop; however, if one is a good thing, two can be better! Both Windows XP and newer OS versions for the Mac offer support for dual monitors (Figure 6.46). Through the addition of a “dual head” video card or by simply adding another video card, these systems can be easily configured so that the two monitors will literally double your working space for less bucks than you might think. In short, it’s truly a joy to have your edit window, mixer, effects sections, and transport controls in their own place...all in plain and accessible view.

The Laptop Computer

One of the most amazing characteristics of the digital age is miniaturization. At the forefront of the studio-a-go-go movement is the laptop computer (Figures 6.47 and 6.48). Out of the advent of smaller, lighter, and more powerful notebooks has come the technological Phoenix of the portable DAW. With the advent of USB and FireWire audio interfaces, controllers, and other peripheral devices, these systems are now capable of handling most (if not all) of
the edit and processing functions that can be handled in the studio. In fact, these AC/battery-
powered systems have become powerful enough to handle advanced DAW edit/mixing
functions, as well as being able to happily handle a wide range of plug-in effects and virtual
instruments...all in the comfort of...anywhere!

That's the good news! Now, the downside of all this portability is the fact that, since laptops
are optimized to run off of a battery with as little power drain as possible, their:

◆ Processors will often run slower.
◆ BIOS (the important subconscious brains of a computer) might be different
  (especially with regards to battery-saving features).
◆ Hard drives might not spin as fast.
◆ Video display capabilities are sometimes limited when compared to a desktop.
◆ Internal audio interface usually isn’t so great.

Although the central processing unit (CPU) will often run slower (often to reduce power
consumption in the form of heat), most modern laptops are more than powerful enough to
perform on the road. For this reason, it’s always best to get a system with the fastest CPU
that you can afford.

Most often, the primary problems in a laptop often rest with the basic BIOS and OS
battery-saving features. When it comes to making music with a laptop, there is actually a
real difference between the Mac and a PC. Basically, there’s little difference between a Mac
laptop and a Mac tower, as the BIOS’s are virtually identical. Conversely, the BIOS of a laptop
PC is often limited in power and functional capabilities when compared to a desktop. When
shopping for a PC laptop, it’s often good to research how a particular BIOS chipset will
work for music...particularly with regard to certain audio interface devices (some interfaces
won’t work well or at all with certain chipsets).
Both the PC and Mac laptops often have an automatic power-saving feature (respectively called “speed step” and “processor cycling”) that changes the CPU’s speed in much the same way that a vehicle changes gears in order to save energy. Often these gear changes will wreak havoc on many of the DSP functions of an audio workstation. Turning them off will greatly improve performance...at the expense of reduced battery life.

Hard-drive speeds on a laptop are often limited, when compared to a desktop computer...resulting in slower access times and fewer track counts on a multitrack DAW. Even though these speeds are often more than adequate for general music applications, these speed issues can often be improved through the use of an external FireWire drive.

Again, when it comes to RAM, it’s often a good idea to pack as much into the laptop as you can. This will reduce data swapping to disk with larger audio applications (especially software synths and samplers). Within certain systems, the laptop’s video card capabilities will run off of the system’s RAM (which can severely limit audio processing functions in a DAW). Memory-related problems can also crop up when using a motherboard to run a dual-monitor (LCD and external monitor) setup.

It almost goes without saying that the internal audio quality of most laptops ranges from being okay to abysmal. As a result, about the only true choice is to find an external audio interface that works best for you and your applications (Figures 6.49 and 6.50). Fortunately, there are a ton of audio interface choices for either FireWire or USB...ranging from a simple stereo I/O device to those that include multitrack audio, MIDI, and controller capabilities in a small, on-the-go package.

In addition to being a studio-on-the-go, the laptop can act as an “expansion module” to a desktop setup. You could trigger a loaded software synth or sampler via MIDI or through the use of VST System Link (a Steinberg product that allows multiple computers to act as a single, connected system via a network connection), and any number of effects, instruments, and VST software devices can work in unison with the main DAW.
In the not-too-distant past, installing a device into a computer or connecting between computer systems could be a major hassle. With the development of the USB and FireWire protocols (as well as the improved general programming of hardware drivers), hardware devices such as mice, keyboards, cameras, soundcards, modems, MIDI interfaces, CD and hard drives, MP3 players...even portable fans...can be plugged into an available port, installed, and up and running in no time, (usually) without a hassle.

With the development of a standardized network protocol, it’s now possible to link computers together in a way that allows for the fast and easy sharing of data throughout a connected system. Using such a system, whole businesses are able to share files with a computer across an administered network on the other side of the country, studios can share soundfiles and videofiles throughout an entire production, and houses can share files and a single high-speed Internet connection with relative ease.

USB

In recent computer history, few protocols for interconnecting devices to a host computer have affected our lives like the Universal Serial Bus (USB). In short, USB is an open specification for connecting external hardware devices to the personal computer, as well as a special set of protocols for automatically recognizing and configuring them. The first of the following two speeds are supported by USB 1.0, while all three are supported by USB 2.0:

- USB 1.0 (1.5 Mbits/second)—A low speed for the attachment of low-cost peripherals (such as a joystick or mouse).
- USB 1.0 (12 Mbits/second)—For the attachment of devices that require a higher throughput (such as data transfer, soundcards, digitally compressed video cameras, and scanners)
- USB 2.0 (480 Mbits/second)—For high-throughput and fast transfer of the above applications
The basic characteristics of USB include the following:

- Up to 127 external devices can be added to a system without having to open up the computer. As a result, the industry is moving toward a “sealed case” or “locked-box” approach to computer hardware design.
- Newer operating systems will often automatically recognize and configure a basic USB device that’s shipped with the latest device drivers.
- Devices are “hot pluggable,” meaning that they can be added (or removed) while the computer is on and running.
- The assignment of system resources and bus bandwidth is transparent to the installer and end user.
- USB connections allow data to flow bidirectionally between the computer and the peripheral.
- USB cables can be up to 5 meters in length (up to 3 meters for low-speed devices) and include two twisted pairs of wires, one for carrying signal data and the other pair for carrying a DC voltage to a “bus-powered” device. Those that use less than 500 milliamps (1/2 amp) can get their power directly from the USB cable’s 5-V DC supply, while those having higher current demands will need to be externally powered.
- Standard USB cables have two types of connectors at each end. For example, a cable between the PC and a device would have an “A” plug at the PC (root) connection and a “B” plug for the device’s receptacle.
- Cable distribution and “daisy-chaining” are done via a data “hub” (Figure 6.51). These devices act as a traffic cop, in that they cycle through the various USB inputs in a sequential fashion . . . routing the data into a single data output line. It should be noted that not all hubs are created equal. In certain situations, the chipset that’s used within a hub might not be compatible with certain MIDI and audio interface

Figure 6.51. The Griffin 4-Port USB audio hub. ( Courtesy of Griffin Technology, www.griffintechnology.com.)
systems. . . Should a connection problem arise, contact the manufacturer of your audio-related device for advice.

**FireWire**

Originally created in the mid-1990s by Apple (and later standardized as IEEE-1394), the FireWire protocol is similar to the USB standard in that it uses a twisted-pair wiring to communicate bidirectional, serial data within a hot-swappable, connected chain. Unlike USB (which can handle up to 127 devices per bus), up to 63 devices can be connected within a connected FireWire chain. FireWire supports two speed modes:

- **FireWire 400 or IEEE-1394a (400 Mbits/second)**—Capable of delivering data over cables up to 4.5 meters in length, FireWire 400 is ideally for communicating large amounts of data to such devices as hard drives, video camcorders, and audio interface devices.

- **FireWire 800 or IEEE-1394b (800 Mbits/second)**—FireWire 800 is capable of communicating large amounts of data over cables up to 100 meters in length. When using fiberoptic cables, lengths in excess of 90 meters can be achieved in situations that require long-haul cabling (such as sound stages and studios).

Unlike USB, compatibility between the two modes is mildly problematic, as FireWire 800 ports are configured differently from their earlier predecessor . . . and therefore require adapter cables to ensure compatibility.

**Networking**

Beyond the concept of connecting external devices to a single computer, another concept hits at the heart of the connectivity age . . . *networking*. The ability to set up and make use of a *local area network* (LAN) can be extremely useful in the home, studio, and/or office, in that it can be used to link multiple computers with various data, platforms, and OS types. In short, a network can be set up in a number of different ways . . . with varying degrees of complexity and administrative levels; however, basically there are two common ways that data can be handled over a LAN (Figure 6.52):

- The first is a system whereby the data that’s shared between linked computers resides on the respective computers and is communicated back and forth in a decentralized manner.

- The second makes use of a centralized computer (called a *server*) that’s basically an array of high-capacity hard drives that is used to store “all” of the data that relates to the everyday production aspects of a facility. Often, such a system will have a redundant set of drives that actually clone the entire system on a moment-to-moment basis . . . as a safety backup procedure. Alternatively (and in some cases, in addition to), a set of backup tapes may be made on a daily basis for extra insurance and archival purposes.
No matter what level of complexity is involved, some of the more common uses for working with a network connection include:

- **Sharing files**—Within a connected household, studio or business, a LAN can be used to share files, soundfiles, video images... virtually anything, throughout the connected facility. This means that various productions rooms, studios, and offices can simultaneously share and swap data and/or mediafiles in a way that's often transparent to the users.

*Figure 6.52. Local Area Network (LAN) connections. (a) Data may be shared between independent computers in a home or workplace LAN environment. (b) Computer terminals may be connected to a centralized server, allowing data to be stored, shared, and distributed from a central location.*
Shared Web connection—One of the cooler aspects of using a LAN is the ability to share an Internet connection over the network from a single, connected computer or server. The ability to connect from any computer with ease is just another reason why you should strongly consider wiring your studio and/or house with LAN connections.

Archiving and backup—In addition to the benefits of archiving and backing up data with a server system... even the simplest LAN can be a true lifesaver. For example, let’s say that we need to make a backup DVD of a session but don’t have the time to tie up our production DAW. In this situation, we could simply burn the disc on another computer that’s connected to the system... and continue working away, without interruption.

Accessing soundfiles and sample libraries—It goes without saying that sound- and samplefiles can be easily accessed from any connected computer... Hey! If you’re wireless (or have a long enough cable), go out to the pool and soak up the sun while you’re working!

On a final note, those who are unfamiliar with networking are urged to learn about this powerful and easy-to-use data distribution and backup system. For a minimal investment in cables, hubs, and educational reading, you might be surprised at the time- and trouble-saving benefits that will be almost instantly realized.

The Audio Interface

An important device that deserves careful consideration when putting together a DAW-based production system is the digital audio interface. These devices can have a single, dedicated purpose, or they might be multifunctional in nature... in either case, their main purpose in the studio is to act as a connectivity bridge between the outside world of analog audio and the computer’s inner world of digital audio (Figures 6.53 through 6.57). Audio interfaces

Figure 6.53. Mackie Spike USB audio/MIDI interface. (Courtesy of Loud Technologies, Inc., www.mackie.com.)
come in all shapes, sizes, and functionalities; for example:

- Built into a computer (although, more often than not, these devices are often limited in quality and functionality)
- A simple, two-I/O audio device
- Multichannel offering eight analog I/Os and numerous I/O expansion options
- Fitted with one or more MIDI I/O ports
- Offering digital I/O, wordclock, and sync options
Fitted with a controller surface (with or without motorized faders) that provides for hands-on DAW operation

These devices may be designed as hardware cards that fit directly into the computer, or they might plug into the system via USB or FireWire. They may offer a limited number of sample-rate and bit-depth options...or might be capable of handling rates up to 96 kHz/24 bits or higher. Unless you buy a system that has been designed to operate with a specific piece of hardware (most notably, Digidesign users), you should weigh the vast number of interface options and capabilities with patience and care—the system you might save could be your own.

Native Versus Non-Native Processing

In the past, computer technology was far too slow to handle the intense number of processing and I/O functions that were required for multichannel DAW production; however, with the introduction of faster CPUs, hard disks, and memory, many computer systems are now able to handle the workload of dealing with basic program operation, signal processing, and passing audio to multiple I/O ports, without the need for special, dedicated hardware. As such, newer software systems that work in the native processing environment have now begun to place the burden of processing and I/O signal routing entirely upon the personal computer, its operating system, and the DAW software. In short, a DAW that has been programmed to work in a native environment is able to route all of its processing, file structure, and I/O functions through the computer’s CPU and operating system. The advantage to such a system is that any audio and interface (and many other devices) that can be seen by the computer’s OS can be accessed by the DAW or host application. That’s to say...if you already have an XYZ interface, chances are it’ll work with your system. If your friend has a better one for a specific session...no problem.

Non-native devices, on the other hand, have been designed to work with a specific piece of supporting interface hardware...period. If you already have an interface, you won’t be able to use it with that particular DAW software. If a new system comes out that requires...
A hardware upgrade, chances are your hardware, software, and possibly your plug-ins will have to be sold on the used market and then be replaced by the new workstation’s hardware. Although a non-native system has the appeal of being a complete WYSIWYG (“what you see is what you get”), all-in-one package, you should be aware of the differences and personal options... should you wish to upgrade or add extra functions to your system later.

**Audio Driver Protocols**

Audio driver protocols are software programs that set standards for allowing data to be communicated between the system’s software and hardware. A few of the more common protocols are:

- **WDM**—The “Windows Driver Model” is a robust driver implementation that’s directly supported by Microsoft’s Windows. Software and hardware that conform to this basic standard can communicate audio to and from the computer’s basic audio ports.
- **ASIO**—The “Audio Stream Input/Output” architecture forms the backbone of VST. It does this by supporting variable bit depths and sample rates, multichannel operation, and synchronization. This commonly used protocol offers low latency, high performance, easy set up, and stable audio recording within VST.
- **MAS**—The “MOTU Audio System” is a system extension for the Mac that uses an existing CPU to accomplish multitrack audio recording, mixer, bussing, and real-time effects processing. In addition to working with popular interface systems, such as the MOTU 2408 and 1224, this system works with a growing number of interfaces, including Digidesign Audiomedia II/III, Sonorus Studl/O, any PCI-based ProTools system via Direct I/O, Yamaha DSP Factory, Event Layla, and the Korg 1212.
- **CoreAudio**—This Digidesign driver allows compatible single-client, multichannel applications to record and play back through most Digidesign audio interfaces on Mac OSX. It supports full-duplex recording and playback of 16-/24-bit audio at sample rates up to 96 kHz (depending on your Digidesign hardware and CoreAudio client application).

Of course, the above listing is far from complete, and further reading can be found from the respective companies. In most circumstances, it won’t be necessary for you to be familiar with the protocols... simply that your software and hardware are compatible for use with a driver protocol that works best for you.

**Latency**

Quite literally, latency refers to the buildup of delays (measured in milliseconds) in an audio signal, as they pass through the audio circuitry of the audio interface, CPU, internal mixing structure, and I/O routing chains. When monitoring a signal directly through a computer’s signal path, latency can be experienced as short delays between the input...
and monitored signal. If the delays are excessive, they can be unsettling enough to throw a performer off time. For example, when recording a synth track, you might actually hear the delayed monitor sound shortly after hitting the keys (not a happy prospect). With the advent of faster computers, improved audio drivers, and better programming, latency has now been reduced to levels that are so small as to not be unnoticeable. For example, the latency of a standard Windows audio driver can be truly pitiful (upward to 500 ms). By switching to a supported ASIO driver and by reducing the audio interface (and possibly the DAW) buffers to their lowest operating size (without causing stuttering)... these delay values could easily be reduced down to an unnoticeable range.

**DAW Controllers**

Often, one of the more common complaints that some people have against the digital audio editor and workstation environment (particularly when relating to the use of on-screen mixers) is the lack of a hardware controller that gives the user access to hands-on controls. In recent years, this has been addressed by major manufacturers and third-party companies in the form of a hardware DAW controller interface (Figures 6.58 through 6.61). These controllers generally mimic the design of an audio mixer in that they offer slide or rotary gain faders, pan pots, solo/mute, and channel select buttons... with the added bonus of a full transport remote. A channel select button is used to actively assign a specific channel to a section that contains a series of grouped pots and switches that relate to EQ, effects, and dynamic functions. Often, these controllers offer direct mixing control over eight input strips at a time. By switching between the banks in groups of 8 (1–8, 9–16, 17–24,...), any number of the grouped inputs can be access by the virtual mixer. These devices will also often include software function keys that can be programmed to give quick and easy access to the DAW’s more commonly used program keys.

*Figure 6.58. Tascam FW-1884 FireWire audio/MIDI interface and control surface. (Courtesy of Tascam, www.tascam.com.)*
Controller commands are most commonly transmitted between the controller and audio editor via device-specific MIDI System-Exclusive messages. As such, in order to be able to integrate a controller into your system, the DAW’s current version must be specifically programmed to accept the control codes from a particular controller... unless the DAW and controller make use of a new plug-in architecture that allows compatible devices to freely connected. Most controller surfaces communicate these messages to the DAW host via the easy-to-use USB or FireWire protocols.

Certain controllers also offer all-in-one capabilities that can be straightforward and cost-effective devices for first-time buyers. Often, these devices include a multichannel audio interface, MIDI interface port(s), monitor capabilities, and full controller functions. Others may already have an existing digital mixer that can actually be used as a fully functional
controller (and in certain circumstances, as a multichannel audio interface) when connected to a DAW host program. For these and other reasons, taking the time to research your needs and current equipment capabilities can save time and money... or, at the worst, can simply be educational.

**Soundfile Formats**

An amazingly varied number of soundfile formats exist within audio and multimedia production. Here are the most commonly used audio production formats that don’t use data compression:

- **Wave (.wav)**—The Microsoft Windows format supports both mono and stereo files at a variety of resolutions and sample rates. WAV files contain PCM coded audio (uncompressed Pulse Code Modulation formatted data) that follows the Resource Information File Format (RIFF) spec, which allows extra user information to be embedded and saved within the file itself.

- **Broadcast wave (.wav)**—In terms of audio content, broadcast wave files are the same as regular wave files; however, text strings for supplying additional information can be imbedded in the file according to a standardized data format.

- **Wave64 (.w64)**—This proprietary format was developed by Sonic Foundry, Inc. (now operating under the Sony name). In terms of audio quality, Wave64 files are identical to wave files... except that their file headers use 64-bit values (instead of Wave’s 32-bit values). As a result, Wave64 files can be considerably larger than standard wave files, and this format is a good choice for long recordings (e.g., surround files and file sizes over 2 GB).
Apple AIFF (.aif or .snd)—This standard soundfile format from Apple supports mono or stereo, 8-bit or 16-bit audio at a wide range of sample rates. Like broadcast wave files, AIFF files can contain embedded text strings.

Sound Designer I & II (.sd and .sd2)—The Sound Designer is used by Digidesign as a soundfile format for the Mac. SDI was first released in 1985 and can still be found on many CD-ROM and soundfile discs; it was primarily used to store 16-bit, mono samples of short duration (often on the order of seconds). As a later incarnation, SDII can now encode 16- or 24-bit soundfiles of any practical length at a variety of sample rates.

Soundfile Sample Rates

The sample rate of a recorded bitstream directly relates to the resolution at which a recorded sound will be digitally captured. Just as with a moving image...if you take more “samples” of the image as it moves through time, you’ll have a more accurate representation of that recorded image. If the number of samples are too low, the resolution will be substandard and “lossy.” On the other hand, too high of a rate might result in a recorded bandwidth that’s so high that the gain in resolution is lost on the audience’s ability to discriminate it...or the storage requirements might become so great that the files become inordinately large. Beyond the basic adherence to certain industry sample rate standards...such are the choices and personal decisions that must be made regarding which is the best sample rate to use on a project. Although other sample-rate standards exist, the following are the most commonly used in the professional, project, and general audio production community:

- **32 k**—This rate is often used by broadcasters to transmit/receive digital data via satellite. With it’s overall 15-kHz bandwidth and reduced data requirements, it is also used by certain devices in order to conserve on memory. Although this rate isn’t generally used by the procommunity, it’s surprising just how good a sound can be captured at 32 k, given a high-quality converter.

- **44.1 k**—The long-time standard of consumer and pro audio production, 44.1 is the chosen rate of the CD-audio standard. With its overall 20-kHz bandwidth, the 44.1-k rate is generally considered to be the minimum sample rate for professional audio production. Assuming that high-quality converters are used, this rate is capable of recording lossless audio, while conserving on memory storage requirements.

- **48 k**—This standard was adopted early on as a standard sample rate for professional audio applications (particularly when referring to hardware digital audio devices).

- **96 k**—With the onset of 24-bit recording capabilities, higher-rate and bit-rate recordings have made it feasible for recordings to be encoded at 96 kHz and higher rates (e.g., 24/96). 96 kHz is also the accepted rate for DVD audio production.

- **192 k**—This is also an accepted rate for DVD audio production.
Soundfile Bit Rates

The bit rate of a digitally recorded soundfile directly relates to the number of quantization steps that are encoded into the bitstream. As a result, the bit rate (or bit depth) is directly correlated to the:

- Accuracy at which a sampled level (at one point in time) is to be encoded
- Signal-to-error figure...and thus the overall dynamic range of the recorded signal

If the number of encoded bits are too low to accurately encode the sample, the resolution will be substandard (i.e., distorted). On the other hand, too high of a bit depth might result in a resolution that’s so high that the resulting gain in resolution is lost on the audience’s ability to discriminate it...or the storage requirements might become so high that the files become inordinately large.

Although other bit-rate standards exist, the following are the most commonly used in the pro, project, and general audio production community:

- 16 bits—The long-time standard of consumer and professional audio production, 16 bits is the chosen bit depth of the CD-audio standard (offering a theoretical dynamic range of 97.8 dB). It is generally considered to be the minimum depth for high-quality professional audio production. Assuming that high-quality converters are used, this rate is capable of recoding lossless audio, while conserving on memory storage requirements.
- 20 bits—Before the 24-bit rate came onto the scene, 20 bits was considered to be the standard for high-bit-depth resolution. Although it is used less commonly, it can still be found in high-definition audio recordings (offering a theoretical dynamic range of 121.8 dB).
- 24 bits—Offering a theoretical dynamic range of 145.8 dB, this standard bit rate is often used in professional-audio, high-definition, and DVD-audio applications.

Format Interchange and Compatibility

At the soundfile level, most software editors and DAWs are able to read a wide range of uncompressed and compressed formats, which can then be saved into a new format. At the session level, there are several standards that allow for the exchange of data for an entire session, from one platform, OS, or hardware device to another. These include:

- Open Media Framework Interchange (OMFI) is a platform-independent session file format intended for the transfer of digital media between different DAW applications; it is saved with an .omf file extension. OMF (as it is commonly called) can be saved in either of two ways: (1) “export all to one file,” when the OMF file includes all of the soundfiles and session references that are included in the session (be prepared for this file to be extremely large), and (2) “export media file references,” when the OMF file will not contain the soundfiles themselves but will contain all of the session’s
region, edit, mix settings; effects (relating to the receiving DAW’s available plug-ins and ability to translate effects routing); and I/O settings. This second type of file will be small by comparison; however, the original soundfiles must be transferred into the session folders.

- Developed by the Audio Engineering Society, the AES31 standard is an open file interchange format that was designed to overcome format incompatibility issues between different software and hardware systems. Transferred files will retain event positions, mix settings, fades, etc. AES31 makes use of Microsoft’s FAT32 file system with broadcast wave as the default audio file format. This means that an AES31 file can be transferred to any DAW that supports AES31, regardless of the type of hardware and software used, as long as the workstation can read the FAT32 file system, broadcast wave, or regular wave files.

- OpenTL is a file exchange format that was developed for Tascam hard disk recording systems. An imported OpenTL project file will contain all audio files and edits that were made within the Tascam system, with all events positioned correctly in the Project window. Conversely, a session can be edited and then exported to a disk in the OpenTL format, making it possible to transfer all edits and audio files back to the Tascam hard-disk device.

**DAW Software**

Probably one of the strongest playing cards in the modern digital audio deck is the digital audio workstation. By their very nature, DAWs (Figures 6.62 through 6.65) are software programs that integrate with computer hardware and functional applications to create...
a powerful and flexible audio production environment. These programs commonly offer extensive record, edit, and mixdown facilities through the use of such production tools as:

- Extensive soundfile recording, edit, and region definition and placement
- MIDI sequencing and scoring
- Real-time, on-screen mixing
Real-time effects
Mixdown and effects automation
Soundfile import/export and mixdown export
Support for video/picture synchronization
Systems synchronization
Audio, MIDI, and sync communications with other audio programs (i.e., ReWire)
Audio, MIDI, and sync communications with other software instruments (i.e., VST technology)

The above list is but a smattering of the functional capabilities that can be offered by an audio production DAW.

Suffice it to say that these powerful software production tools are extremely varied in their form and function. Even with their inherent strengths, quirks, and complexities—the basic look, feel, and operational capabilities have, to some degree, become unified between the major DAW competitors. Having said this, it goes without saying that there are enough variations in features, layout, and basic operation that individuals, from aspiring beginner to seasoned professional, will have their favorite DAW make and model. With the growth of the DAW and computer industries, people have begun to customize their computers with features, added power, and peripherals that rival their love for souped-up cars and motorcycles. In the end, though...as with many things in life...it’s doesn’t matter which type of DAW you use—it’s how you use it that counts!
For the rest of this section on DAWs, we’ll be taking a look at that various functional aspects of the digital audio workstation (and hard disk recorders, in general). Of course, there’s no way that all of the general features of a DAW can be covered (let alone specific features of a particular workstation); for that, I’ll refer you to the tons of books and manuals that have been written on each DAW. These writings can be amazing gems that often offer specific insights into production tools and techniques that can fine-tune your production habits.

Soundfile Recording, Editing, Region Definition, and Placement

Hard-disk recorders are capable of recording mono, interleaved stereo (where the L/R or multichannel data is alternately encoded within a single file) and multitrack soundfiles directly to disk in a graphic working environment.

Do-It-Yourself Tutorial: Recording a Soundfile to Disk

- Consult your editor’s manual regarding recording a soundfile to disk.
- Assign the track to an interface input sound source.
- **Name the track!** It’s almost always best to name the track (or tracks) before going into record. In this way, the file will be saved to disk within the session folder under a descriptive name instead of an automatically generated filename (e.g., killerkick.wav... instead of track16-01.wav).
- Save the session and assign the input to another track and overdub a track along with the previously recorded track.

Most hard-disk recording systems graphically display soundfile information within the created tracks of a main graphic window (Figure 6.66). These tracks contain drawn waveforms that graphically represent the amplitude of a soundfile over time in a WYSIWYG fashion. Depending on the system type, soundfile length, and the degree of zoom, the entire waveform

*Figure 6.66. Graphic display of a soundfile region.*
can be shown on the screen, or only a portion will be shown with it continuing to scroll off one or both sides of the screen.

Graphic editing differs greatly from the “razor blade” approach that’s used to cut analog tape in that the waveform gives us both visual and audible cues as to where a precise edit point should be. Using this common display technique, any position, cut/copy/paste, gain, and time changes to the waveform will be instantly reflected on the screen. Usually, these edits are nondestructive (a process whereby the original file isn’t altered... only the way that the region in/out points are accessed or the file is processed as to gain, spectrum, etc.).

Only when a waveform is zoomed-in fully is it possible to see the individual waveshapes of a soundfile (Figure 6.67). At this zoom level, it becomes simple to locate zero-crossing points (points where the level is at the “0,” center-level line). In addition, when a soundfile is zoomed-in to a level that shows individual sample points, the program might allow the sample points to be redrawn, in order to remove potential offenders (such as clicks and pops) or to smooth out amplitude transitions between loops or adjacent regions.

When working in a graphic editing environment, regions can usually be defined by positioning the cursor over the waveform, pressing and holding the mouse or trackball button, and then dragging the cursor to the left or right, which highlights the selected region for easy identification. After the region has been defined, it can be edited, marked, named, maimed, or otherwise processed.

As one might expect, the basic cut and paste techniques used in hard-disk recording are entirely analogous to those used in a word processor or other graphics-based programs:

- **Cut**—Places the highlighted region into memory and deletes the selected data (Figure 6.68)
- **Copy**—Places the highlighted region into memory and doesn’t alter the selected waveform in any way (Figure 6.69)
- **Paste**—Copies the waveform data that’s within the system’s clipboard memory into the soundfile beginning at the current cursor position (Figure 6.70)

*Figure 6.67. Zoomed in area of a soundfile showing sample edit points.*
Do-It-Yourself Tutorial: Copy and Paste

- Consult your editor’s manual regarding basic cut and paste commands.
- Open a soundfile and define a region that includes a musical phrase or sentence.
- Cut the region and try to paste it into another point in the soundfile in a way that makes sense (musical or otherwise).
- Feel free to cut, copy, and paste to your heart’s desire to create an interesting or totally wacky soundfile.
Besides basic cut and paste techniques, processing the amplitude of a signal is one of the most common types of changes that are likely to be encountered. These include such processes as gain changing, normalization, and fading. Gain changing relates to the altering of a region or track’s overall amplitude level, such that a signal can be proportionally increased or reduced to a specified level (often in dB or percentage value). In order to increase a soundfile or region’s overall level, a function known as normalization can be used. Normalization (Figure 6.71) refers to an overall change in a soundfile or defined region’s signal level, whereby the file’s greatest amplitude will be set to 100% (or a set percentage level), with all other levels in the soundfile or region being proportionately changed in gain level. The fading of a region (either in or out) is accomplished by increasing or reducing a signal’s relative amplitude over the course of a defined duration. For example, fading in a file (Figure 6.72a) proportionately increases a region’s gain from infinity (zero) to full gain. Likewise, a fade-out (Figure 6.72b) has the opposite effect of creating a transition from full gain to infinity. These DSP functions have the advantage of creating a much smoother transition than would otherwise be humanly possible when performing a manual fade. A cross-fade (or X-fade) is often used to smooth the transition between two audio segments that either are sonically dissimilar or don’t match in amplitude at a particular edit point (a condition that would otherwise lead to an audible “click” or “pop”). This functional tool basically overlaps a fade-in and fade-out between the two waveforms to create a smooth transition from one segment to the next (Figure 6.73). Technically, this process averages the amplitude of the signals over a user-definable length of time in order to mask the offending edit point.

**MIDI Sequencing and Scoring**

Most DAWs include extensive support for MIDI (Figure 6.74), allowing electronic instruments, controllers, effects devices, and electronic music software to be integrated with
Figure 6.72. Examples of various fade curves: (a) fade-in; (b) fade-out.

![Examples of various fade curves](image)

Figure 6.73. Example of a cross-faded soundfile. (Courtesy of Steinberg Media Technologies GMBH, www.steinberg.net.)

![Example of a cross-faded soundfile](image)

Figure 6.74. MIDI edit window within the Cubase SE audio production software. (Courtesy of Steinberg Media Technologies GMBH, www.steinberg.net.)

![MIDI edit window](image)
multitrack audio and video tracks. This important feature often includes the full implementation for:

- MIDI sequencing, processing and editing
- Score editing and printing
- Drum pattern editing
- MIDI signal processing
- Support for linking the timing and I/O elements of an external music application (ReWire)
- Support for software instruments (VSTi)

Further reading about the wonderful world of MIDI can be found within Chapter 7.

Real-Time, On-Screen Mixing

In addition to their abilities to offer extensive region edit and definition, one of the most powerful, cost- and time-effective features of a digital audio workstation is its ability to offer on-screen mixing (Figure 6.75). Essentially, most DAWs include a digital mixer interface that offers most (if not all . . . or more) of the capabilities that are offered by larger analog and/or digital consoles that are far more cost and space prohibitive. In addition to the basic input strip fader, pan, solo/mute and select controls . . . most DAW software mixers offer extensive support for EQ, effects plug-ins (offering a staggering amount of DSP flexibility that will be covered later in the chapter), spatial positioning (pan and possibly surround-sound positioning), total automation (both mixer and plug-in automation), external mix, function and transport control from a supported external hardware controller, support for exporting a mixdown to a file . . . the list goes on and on and on and . . .

Figure 6.75. Nuendo 2.0 on-screen mixer. (Courtesy of Steinberg Media Technologies GMBH, www.steinberg.net.)
DSP Effects

In addition to being able to cut, copy, and paste regions within a soundfile, it’s also possible to alter a soundfile, track, or segment using digital signal processing techniques. In short, DSP works by directly altering the samples of a soundfile or defined region according to a program algorithm (a set of programmed instructions), so as to achieve a desired result. These processing functions can be performed either in real time or non-real time:

- **Real-time DSP**—Commonly used in most modern-day DAW systems, this process makes use of the computer’s CPU or additional acceleration hardware to perform complex DSP calculations during actual playback. Because no calculations are written to disk in an off-line fashion, significant savings in time and disk space can be realized when working with productions that involve complex or long processing events. In addition, the automation instructions for real-time processing are imbedded within the saved session file, allowing any effect or set of parameters to be changed, undone, and redone...without affecting the original soundfile.

- **Non-real-time DSP**—Using this method, signal processing (such as changes in level, EQ, dynamics or reverb) that is too calculation intensive to be carried out during playback will be calculated (in an off-line fashion). In this way, the newly calculated file (containing the effect, sub-mix, etc.) will be played back, without having to use up the extra resources that are now available to the CPU for other functions. DAWs will often have a specific term for tracks or processing functions that have been written to disk...such as “locking” or “freezing” a file. When DSP is performed in non-real time, its almost always wise to save both the original and the effected soundfiles...just in case you need to make changes at a later time.

Most DAWs offer an extensive array of DSP options, ranging from options that are built into the basic I/O path of the input strip (e.g., basic EQ and gain-related functions) to DSP effects and plug-ins that come bundled with the DAW package...to third-party effects plug-ins that can be either inserted directly into the signal path (direct insertion) or offered as a master effect path that numerous tracks can be assigned to and/or mixed into (side chain).

Although the way that effects are implemented into a DAW will vary from one make and model to the next, the basic fundamentals will be much the same. The following discussion describes but a few of the possible effects that can be “plugged” into the signal path of DAW; however, further reading on effects processing can be found in Chapter 12 (Signal Processing) and Appendix A (DSP Basics).

- **Equalization**—EQ is, of course, a feature that’s often implemented at the basic level of a virtual input strip (Figures 6.76 and 6.79). Most systems give full parametric control over the entire audible range...offering overlapping control over several bands, with a variable degree of bandwidth control (Q). Beyond the basic EQ options, many third-party EQ plug-ins are available on the market that vary in complexity, musicality, and market appeal (Figures 6.81 and 6.82).
Figure 6.76. Nuendo’s EQ within the Channel Settings window. (Courtesy of Steinberg Media Technologies GMBH, www.steinberg.net.)

Figure 6.77. Pro Tools EQ screen. (Courtesy of Digidesign, a division of Avid Technology, Inc., www.digidesign.com.)

Figure 6.78. Waves Renaissance equalizer plug-in. (Courtesy of Waves, Ltd., www.waves.com.)
Dynamic range—Dynamic range processors (Figures 6.80 and 6.81) can be used to change the signal level of a program. Processing algorithms are available which emulate a compressor (a device that reduces gain by a ratio that’s proportionate to the input signal), limiter (reduces gain at a fixed ration above a certain input threshold), or expander (increase the overall dynamic range of a program). These gain changers can be inserted directly into a track, used as a grouped master effect, or inserted into the final output path for use as a master gain processing block.

In addition to the basic complement of dynamic range processors, wide assortments of multi-band dynamic plug-in processors (Figure 6.82) are available for general and mastering DSP
applications. These processors (which are further covered within Chapter 16, Mastering) allow the overall frequency range to be broken down into various frequency bands. For example, a plug-in such as this could be inserted into a DAW’s main output path, which allows the lows to be compressed, while the mids are lightly limited and the highs are de-essed to reduce sibilance.

Delay—Another important effects category that can be used to alter and/or augment a signal revolves around delays and regeneration of sound over time. These time-based effects use delay (Figures 6.83 and 6.84) in order to add a perceived depth to a signal or change the way that we perceive the dimensional space of a recorded sound.
A wide range of time-based plug-in effects exist that are all based upon the use of delay (and/or regenerated delay) to achieve such results as:
- Delay
- Chorus
- Flanging
- Reverberation

As was stated toward the beginning of this section, further reading on the subject of delay (and the subject of signal processing in general) can be found in Chapter 12 (Signal Processing).

- **Pitch and Time Change**—Pitch change functions make it possible to shift the relative pitch of a defined region or track either up or down by a specific percentage ratio or musical interval. Most systems can shift the pitch of a soundfile or defined region by determining a ratio between the current and the desired pitch and then adding (lower pitch) or dropping (raise pitch) samples from the existing region or soundfile (Figure 6.85).

In addition to raising or lowering a soundfile’s relative pitch, most systems can combine variable sample rate and pitch shift techniques to alter the duration of a region or track. These pitch- and time-shift combinations make it possible for such changes as:
- **Pitch shift only**—A program’s pitch can be changed while recalculating the file so that its length remains the same.
- **Change duration only**—A program’s length can be changed while shifting the pitch so that it matches that of the original program.
- **Change in both pitch and duration**—A program’s pitch can be changed while also having a corresponding change in length.

When combined with shifts in time (delay), changes in pitch make it possible for a multitude of effects to be created (such as flanging, which results from random fluctuations in...
delay and time shifts that are mixed with the original signal to create an ethereal “phasey” kind of sound).

**DSP Plug-Ins**

Workstations often offer a number of DSP effects that come bundled with the program; however, a staggering range of third-party plug-in effects can be inserted into a signal path which perform functions for any number of tasks ranging from the straightforward to the wild 'n' zany (Figures 6.86 through 6.88). These effects can be programmed to seamlessly integrate into a host DAW application that conform to such plug-in platforms as:

- **DirectX**—A DSP platform for the PC that offers plug-in support for sound, music, graphics (gaming), and network applications running under Microsoft Windows (in its various OS incarnations)
- **AU (Audio Units)**—Developed by Apple for audio and MIDI technologies in OSX; allows for a more advanced GUI and audio interface
- **VST (Virtual Studio Technology)**—A native plug-in format created by Steinberg for use on either a PC or Mac; all functions of a VST effect processor or instrument are directly controllable and automatable from the host program
- **MAS (MOTU Audio System)**—A real-time native plug-in format for the Mac that was created by Mark of the Unicorn as a proprietary plug-in format for Performer and Digital Performer; MAS plug-ins are fully automatable and do not require external DSP in order to work with the host program
Figure 6.86. Universal Audio RealVerb Pro reverb plug-in for the UAD-1. (Courtesy of Universal Audio, www.uaudio.com.)

Figure 6.87. Waves Morphoder Vocoder plug-in. (Courtesy of Waves, Ltd., www.waves.com.)

Figure 6.88. Hyperprism Gold OSX signal processing plug-in in HyperVerb mode. (Courtesy of Arboretum Systems, Inc., www.arboretum.com.)
AudioSuite—A file-based plug-in that destructively applies an effect to a defined segment or entire soundfile...meaning that a new, effected version of the file is rewritten in order to conserve on the processor's DSP overhead; when applying AudioSuite, it’s often wise to apply effects to a copy of the original file so as to allow for future changes.

RTAS (Real-Time Audio Suite)—A fully automatable plug-in format that was designed for Digidesign’s Pro Tools LE; available on Digi ToolBox and Digi 001 (any system with Pro Tools LE) and runs on the power of the host CPU (host-based processing) on either the Mac or PC.

TDM (Time Domain Multiplex)—A plug-in format that can only be used with Digidesign Pro Tools systems (Mac or PC) that are fitted with Digidesign Farm cards; this 24-bit, 256-channel path integrates mixing and real-time digital signal processing into the system with zero latency and under full automation.

These popular software applications (which are being programmed by major manufacturers and third-party startups alike) have helped to shape the face of hard-disk recording, by allowing us to pick and choose those plug-ins that best fit our personal production needs. As a result, new companies, ideas, and task-oriented products are constantly popping up on the market...literally on a monthly basis.

Accelerator Cards

In most circumstances, the CPU of a host DAW program will have sufficient power and speed to perform all of the DSP effects and processing needs of a project. Under extreme production conditions, however, the CPU might run out of computing steam and choke during real-time playback. Under these conditions, two choices could be made in order to reduce the workload on a CPU: On the one hand, the track(s) could be “frozen,” meaning that the processing functions would be calculated in non-real time and then written to disk as a separate file. On the other hand, an accelerator card (Figure 6.89) could be placed into the system that’s capable of adding an extra CPU into the circuit, giving the system extra processing power to perform the necessary effects calculations.

Figure 6.89. The tc electronic PowerCore FireWire rack-mountable effects accelerator. (Courtesy of tc electronic, www.tcelectronic.com.)
Mixdown and Effects Automation

One of the great strengths of the digital age is how easily all of the mix and effects parameters can be automated and recalled within a mix. A DAW is particularly strong in this area. The ability to change levels, pan, and virtually control any parameter within a project makes it possible for a session to be written to disk, saved, and recalled at a seconds notice. In addition to grabbing a control and moving it (either on-screen or from a physical controller) . . . “rubberband” controls let you view, draw, and edit various parameters as a representative graphic line that details the various parameter moves over time (Figure 6.90). Generally, the edit moves that have been made within a mix can be undone, redone, or recalled back to a specific point in the edit stage. Often (but not always), the moves within a mix can’t be “undone” and reverted back to a specific point in the mix. Obviously, one of the best ways to save (and revert to) a particular version of a mix in progress (or various versions of an alternate mix) is simply to save the version under a unique (and descriptive) session file title . . . and then keep on working.

Exporting a Mixdown to File

Most DAWs systems are able to export (print) part or an entire session to a single file or set of soundfiles (Figure 6.91). The former refers to multiple channels that are interleaved together in a L–R–L–R . . . or multichannel fashion, while the latter renders the channels as individual channel files. Often, the session can be exported in non-real time (a faster than real-time process that can include all mix, plug-in effects, automation, and virtual instrument calculations) or in real-time (a process that’s capable of sending and receiving real-time analog signals through the audio interface so as to allow for the insertion of external effects devices, etc.). Usually, a session can be mixed down to a number of final soundfile and bit/sample-rate formats. Certain DAWs might also allow third-party plug-ins to be inserted into the final (master) output section . . . allowing for the export of a session to a specific output.
file format. For example, a discrete surround mix could be folded down into a two-channel Dolby ProLogic surround-sound file, or the same file could be rendered as a Dolby Digital 5.1 file for insertion into a DVD video soundtrack.

Support for Video and Picture Sync

Speaking of video . . . most high-end DAWs include support for displaying a video track within a session, both as a video window that can be displayed on the monitor desktop as well as in the form of a video thumbnail track that will often appear in the track view as a linear guide track. Both of these provide important visual cues for tracking live music, sequencing MIDI tracks, and accurately placing effects (sfx) at specific hit points within the scene (Figure 6.92).

Figure 6.91. Many DAWs are capable of exporting a session soundfiles, effects, and automation to a final set of mixdown tracks.

Figure 6.92. Most high-end DAW systems are capable of importing a videofile directly into the project session window.
Systems Synchronization

Through the use of SMPTE time code, MTC, and wordclock, the timing elements of a DAW can be locked to various media devices within a studio or media production house. Transport control over external media devices can also be accomplished through the use of Sony 9-pin and MMC (MIDI machine control).

Audio, MIDI, and Sync Communications with Other Audio Programs

Through the use of a standard protocol (such as Propellerhead’s ReWire technology), many DAWs are able to communicate audio, MIDI, and timing information between one or more independent music programs to the host DAW. In this way, a music program that was designed to perform to a specific media task could be linked to the DAW’s I/O mix and transport functions...effectively allowing multiple programs to work in tandem as a unified production system.

Loop-Based Audio Editors

Loop-based audio editors are groove driven music programs (Figures 6.93 and 6.94) that are designed to let you drag and drop prerecorded or user-created loops and audio tracks into a graphic multitrack production interface. With the help of custom, royalty-free loops (available from the manufacturer and/or third-party companies), users can quickly and easily experiment with setting up grooves, backing tracks, and creating a sonic ambience by simply dragging the loops into the program’s main soundfile view...where they can be arranged in a multitrack or special GUI as a session file that can be saved to disk.

Figure 6.93. Apple GarageBand.
One of the most interesting aspects of these editors is their ability to match the tempo of a specially created loop soundfile to the tempo of the current session. Amazingly enough, this process isn’t that difficult to perform, as the program extracts the length, native tempo, and pitch information form the imported file’s header and (using various digital time and/or pitch change techniques) adjusts the loop to fit the native time/pitch parameters of the current session. This means that loops of various tempos and musical keys can be automatically adjusted in length and pitch so as to fit in time with previously existing loops (a process that would otherwise take a great deal of patience to manually pull off).

These shifts in time to match a loop to the session’s native tempo can actually be performed in a number of ways. For example, using basic DSP techniques to time-stretch and pitch-shift a recorded loop will often work well over a given plus-or-minus percentage range (which is often dependant upon the quality of the program algorithms). Beyond this range, the loop will often begin to distort and become jittery. At such extremes, other beat slice detection techniques can be used to make the loop sound more natural. For example, drums or percussion can be stretched in time by adding additional silence between the various hit-points within the loop, at precisely calculated intervals. In this way, the pitch will remain the same while the length is altered. Of course, such a loop would sound choppy and broken up when played on its own; however, when buried within a mix, it might work just fine...it’s all up to you and the current musical context.

Preprogrammed loops that will work with a number of groove editors can be obtained from any number of sources, such as:

- The Web (both free and for purchase)
- Commercial CDs
- Rolling your own (creating your own loops can add a satisfying and personal touch)
It’s important to note that at any point during the creation of a composition, audio and/or MIDI tracks (such as vocals or played instruments) can easily be recorded into a loop session in order to give the performance a fluid and more dynamic feel. It’s even possible to record live instruments into a session with a defined tempo... and then edit these tracks into defined loops that can be dropped into the current and future sessions to add a live touch.

As these programs often include many of the features that you’ll find in a full-featured DAW (including real-time effects, mixing, multiple I/O, and synchronization)... the completed session can be exported as a mixdown file. However, in addition to acting as a stand-alone music application, the I/O, transport, and timing elements of many of these programs can be fully integrated into a DAW (most often through the use of ReWire technology). In this way, groove tracks can be integrated into a master session, where they can be processed and mixed down as a master file for mastering to its intended media.

**Beat Slicing**

As we now know, it’s often necessary to alter the beats per minute (bpm) of one loop to that of another. A loop-based audio editor (and/or a DAW that includes provisions for intelligent beat matching) will be able to use a variety of time-stretching, pitch-shifting, and formant-shifting algorithms. Another method, called *beat slicing*, actually breaks an audiofile into a number of small segments (often called *slices*). Such a system can be used to preserve the pitch, timbre, and sound quality of a file by altering the time between these slices rather than by changing the speed and pitch at which it’s played. This process can be done by slicing the loop into equal segments that are based upon its original tempo (e.g., cutting the loop into 8th or 16th intervals)... or by detecting the transient events within the loop and placing the slices at the appropriate points (often according to user-definable sensitivity and detection controls). If the slices are accurately detected to correspond to the “hits” within the loop (e.g., occurring on the various beats of a drum loop), timing modifications can be made in order to extend or shorten its length (while the pitch remains unaltered) to match the session’s current tempo. Given that the loop is now divided into smaller segments (that might be small enough so that each segment is made up of one or more instruments... such as an individual kick):

- Each of those segments could actually be replaced with another instrument (e.g., a different kick).
- The various segments could be shuffled in a directed or random order to create a totally new pattern.
- The beat’s emphasis could be shifted to create a new “feel” (e.g., make/remove a shuffle groove).
- Different slices could be assigned to different tracks in order to add different effects to various, individual slices, etc.
Certain programs are able to match and/or distort the beat timing of a loop by assigning the various slices to a MIDI note scale (Figure 6.95). By cycling through (and altering the tempo, notes, and velocities of) the scale, the loop can be changed, replaced, and otherwise mutilated in any number of imaginable ways.

**DJ Software**

In addition to music production software, there’s a growing number of software players, loopers, groovers, effects, and digital devices on the market for the digital DJ of the 21st century. These hardware/software devices make it possible for digital grooves to be created from a laptop, controller, specially fitted turntable, or digital turntable (jog/scratch CD player) with an unprecedented amount of preprogrammable and/or live performance interactivity that can be used on the floor, on stage, or in the studio (Figure 6.96).

**Backup and Archive Strategies**

The phrase “nothing lasts forever” is often especially true in the digital domain of lost 1’s and 0’s, damaged media, dead hard drives, and lost data... you know, the “oops factor.” It’s a basic fact that you never quite know what lies around the techno bend... and, of course, it’s extremely important that you protect yourself as much as is humanly possible against the inevitable. Of course, the answer to most digital dilemmas is to back up your data in the most reliable (or redundant) way. Hardware and program software can (usually) be replaced; on the other hand, when valuable session soundfiles are lost... they’re lost!
Backing up a session can be done in several ways, depending upon the level of assured security that the files will be able to be played in the future with a minimum of hassle and potential problems. Here are a few tips on this important topic:

- As you might expect, the most straightforward backup system is to copy the session data, in its entirety, to the most appropriate media. This works well in the short run, as it assumes that all of your program and plug-in data is loaded and up to date.
- In the longer run (5+ years), the most ironclad way to back up the track data of a session is to print each track as its own .wav, .aif, or .sd2 file. This file should always be recorded or exported to a file as a contiguous file that flows from the beginning of the session (00:00:00:00 or appropriate begin point) to the end of that particular track. In this way, the individual track files could be loaded into any type of DAW, at the beginning point, for processing and mixdown.
- In such a track-by-track safety restoration situation, you might want to save two copies of a track that has a particular effect... one that contains the original and effected sound and another that simply contains the original, unaltered sound.
- For those that might want additional protection against the degradation of unproven digital media, you might also want to back each track (or group of tracks) to the individual tracks of an analog recorder.
- For those sessions that contain MIDI tracks, you should always keep these tracks within the session (i.e., don’t delete them). These tracks might come in very handy during a remix or future mixdown.
- Speaking of MIDI... It’s always wise to export all of the MIDI tracks/data within a session as standard MIDI file. You should at least save all of the tracks as a type 1 file (where all of the multichannel track/data information is left intact... and whenever
possible save it as both a type 0 and a type 1 (you never know what file format obstacles might haunt you in the future). For those wanting additional safety, it might be wise to create a contiguous soundfile (in the same manner as above) that’s a recorded file of each MIDI instrument track . . . just in case the instrument is no longer available for any reason.

Whenever possible, make multiple backups and store them in separate locations. Having a backup copy in your home as well as the studio can save your proverbial butt in case of a fire or unforeseen situation.

As you might expect, all of these tracks can add up to a ton of data. For example, a 3-minute song that’s been recorded onto 24 tracks at 16 bit/4.1 kHz would add up to 720 Mb (10 Mb min/track × 3 minutes × 24 tracks = 720 Mb). When you add up the sessions tracks from multiple songs, original DAW session data, and alternate track takes . . . you could easily end up backing up huge amounts of data onto such backup media as:

- **Hard disk**—Large amounts of session/track data can be backed to a large removable drive (or duplicated onto multiple drives).
- **DVD+R or DVD-R**—This removable media lets you back up to 5 Gb (gigabytes) of data onto a single disc. (My good buddy, Craig Anderton, recommends that you individually back each song onto its own disc . . . In case something happens to the media, just that song would be lost, not all of them.)
- **CD-R**—This removable media can hold up to 800 Mb of data and is reliable assuming that proper care is taken in labeling the disc with a water-based CD marker and that proper storage and handling precautions are taken.

Just think of how you would feel if all of your most precious, original session material would be lost . . . forever! All of the frustration and deep sense of loss can be avoided (or at least cushioned) by thinking ahead and saving the day . . . by backing up your data.

**Session Documentation**

Closely related to backing up your session data is the need for comprehensive session documentation. There are few things that are more frustrating than going back to an archived session and finding that no information exists on what instrument patch, mic type, or outboard effect was used on a DAW session (or any session, for that matter). The importance of documenting a session within a separate written document and/or within the notepad apps within a DAW can’t be overemphasized (Figure 6.97). Just a few of the important information and useful parameters that should be included in your session documentation include:

- Session tempo
- Participants in the project and important dates (for future credit references)
- Mic choice and placement (for future overdub reference)
- Outboard equipment type and their settings
Plug-in effects and their settings or general descriptions (you never know if they’ll be available at a future time, so a description can help you to duplicate it with another app)

In short, the more information that can be archived with a session (and its backups) . . . the better chance that you’ll be able to duplicate the session in greater detail, at some point in the future. Just remember that it’s up to us to save and document the music of today for the fans and future playback/mix technologies of tomorrow.

Figure 6.97. Nuendo Notepad apps. (Courtesy of Steinberg Media Technologies GMBH, www.steinberg.net.)