Digital audio breaks the sound barrier

YOUR CUSTOMERS EXPECT HIGH-QUALITY AUDIO RECORDING AND PLAYBACK, BUT THEY DON'T WANT TO PAY FOR ENORMOUS HARD DRIVES OR T1 LINES OR WAIT FOREVER FOR FILE TRANSFERS. WHAT TO DO? DUST OFF YOUR DIGITAL-SIGNAL-PROCESSING REFERENCE MANUALS AND GET READY TO TRICK THEIR EARS AND BRAINS.

A dual-channel, 16-bit digital audio clip sampled at 44.1 kHz creates a bit stream a little larger than 1.4 Mbps. In other words, that file requires nearly 11 Mbytes/minute of storage capacity. Lossless-compression techniques might shrink the data by a third or a half, but that compressed size is still more than a conventional ADSL or cable-modem connection can reliably stream or quickly download, and it rapidly fills up magnetic-, optical- and semiconductor-based storage media (see **sidebar** "No loss, your gain").

These statistics go a long way toward explaining the booming interest in lossy, or perceptual, compression, which can shrink a file to one-twelfth or even one-twenty-fourth its original size with little to no audible quality difference from the original (**Table 1**). If the audio source can tolerate restrictions on fidelity or frequency, such as a subwoofer-targeted deep-bass channel or a monophonic spoken-word track, the reduction can be even more significant. **Reference 1** explains the audio attributes that enable both lossless and lossy compression to work their magic. Now, it's time to dig into the algorithms themselves to see how they turn compression potential into reality.

COMMON GROUND

Many lossy-compression schemes employ a similar set of high-level functional building blocks, or a subset of them (**Figure 1**). The encoder groups the incoming uncompressed bit stream into multiple-sample overlapping windows (**Figure 2**), and each set routes in parallel to a filter bank and to a perceptual processor. The filter bank executes a time-to-frequency transform on each window, analogous to the location-to-frequency pixel-matrix DCT at the heart of JPEG and MPEG image compression.

Temporal-to-frequency transforms remove sample-to-sample frequency redundancy, collapsing it into common coefficients. If a transform is well-designed and no alteration, such as filtering, occurs while the data is in the frequency domain, a subsequent frequency-to-time retransform exactly reconstructs the original signal. This "input-equals-output" goal is called an identity system.

The perceptual processor determines the frequency and temporal-masking characteristics of the samples in the frame. In frequency masking, a louder tone masks quieter tones of nearby frequencies, whereas in temporal masking, a loud tone masks quieter tones that occur both before and after it (**figures 3** and **4**). The perceptual processor first identifies the highest intensity audio samples and transformed frequency bands and then calculates their masking profiles, which it combines along with the human auditory system's sensitivity curve as a function of frequency.

Using the outputs of the filter bank and the perceptual model, the quantizer determines which tones' data it can significantly attenuate, based on their frequencies and the calculated masking profile (**Figure 5**). It completely discards some tones' data. Because of the ear's inherent behavior and because of masking from louder nearby (in time and frequency) tones, the information would be inaudible even if it were present in the final compressed bit stream. Also, this functional block determines the masking level for each frequency below which noise created by quantization is imperceptible. It quantizes each coefficient to the point at which the added noise is below this just-audible threshold. Some codecs choose a nonlinear-quantization approach, under the assumptions that low-intensity tones are more common than loud ones and that fine resolution of detail is more important during quiet passages (during which an adequate SNR is most difficult to achieve).

AT A GLANCE

- ▷ Audio codecs fundamentally differ based on the processing performance and memory required to compress audio to a given quality level in a given amount of time.
- ▷ Simply converting audio from the time to the frequency domain provides opportunities to identify and remove redundancy.
- ▷ Frequency and temporal masking, along with an understanding of human psychoacoustics, guide quantization and elimination of frequency coefficients.
- ▷ Lossy compression often also incorporates statistical coding to further reduce the bit-stream size.
- ▷ Perceptual codecs might capture most industry attention, but lossless compression has a role to play, too.
- ▷ Encoding and decoding are only two of the many tasks you're asking your audio processor to tackle.

The next optional step in the process involves further losslessly compressing the quantized coefficients, using a Huffman-coding, an arithmetic-coding, or a similar scheme (**Reference 2**). Because quantization typically produces long sequences of repeated zeroes, lossless coding can effectively reduce the bit-stream size. Finally, the encoder packs the coefficients into common-sized data "chunks," sometimes also adding synchronization, error-concealment, buffer-management, info-header, and other overhead bits.

The corresponding decoder is usually far simpler, befitting a one-to-many multimedia-distribution scheme. It first unpacks the compressed audio bit stream, regenerates the frequency coefficients, and then executes a frequency-to-time retransform, often in conjunction with lowpass filtering to remove aliases. It also must handle buffer management, particularly with VBR compression, and error management for cases in which some incoming packets arrive out of sequence and others don't arrive at all.

Sounds straightforward, right? So why, then, are so many codecs available for licensing or royalty-free usage? The most significant differentiators between them are the processing muscle, the memory size, and the corresponding power consumption they need to encode to a given bit-stream size in a given amount of time, as well as the time, processing, memory, and power they require to decode the resultant compressed file (see **sidebar** "Pick a processor for perfect pitch"). One codec might require a microprocessor or dedicated logic circuit running twice as fast as a simpler alternative or might gobble up significantly more program or data memory for comparable compression and decompression speeds. In exchange, though, the more complex algorithm might deliver audibly higher quality than its counterpart at the same bit rate.

Some codecs are tailored for error-filled broadcast environments, whereas others assume a more transmission-friendly model in which the compressed file is locally, versus remotely, stored. For example, RealAudio distributes consecutive-window sample coefficients across multiple network packets, along with

TABLE 1—REPRESENTATIVE LOSSY-COMPRESSION ALGORITHMS

Codec	Web site
ATRAC	www.minidisc.org
AAC	www.aac-audio.com, www.cselt.it/mpeg, www.iis.fhg.de/amm/techinf/aac, www.mpeg.org
ATELP	www.softsound.com/ATELP.html
apt-X	www.aptx.com
DTS	www.dtstech.com, www.fivepoint1.com/home.html
Dolby Digital	www.dolby.com/digital
HDCD	www.hdcd.com
MPEG-1/2/2.5	www.cselt.it/mpeg, www.iis.fhg.de/amm/techinf/basics.html, www.mpeg.org
Ogg Vorbis	www.vorbis.com
PAC/ePAC	www.lucent.com/ldr, www.vedalabs.com
Qdesign	www.qdesign.com
RealAudio	www.real.com
TAC	http://kk-research.hypermart.net
TwinVQ	http://sound.splab.ecl.ntt.co.jp/twinvq-e, www.vqf.com, www.yamaha-xg.com/english/xg/SoundVQ
WMA	www.microsoft.com/windows/windowsmedia

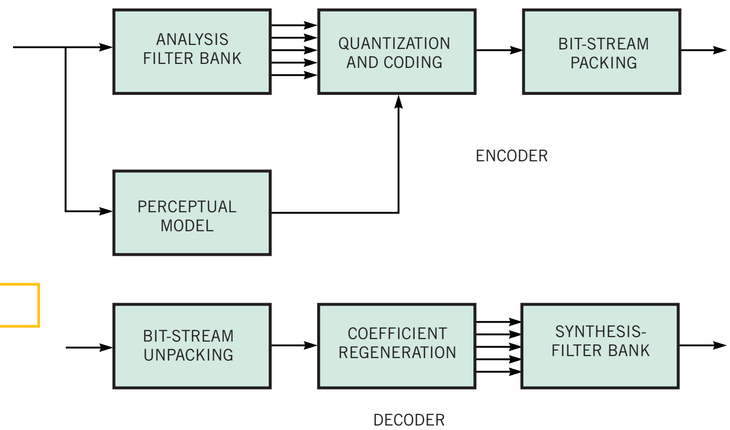
buffering both at the encoder and the decoder, to minimize the error-time interval a dropped packet creates. Instead of outputting often-disagreeable-sounding error data, a streaming decoder might instead mute the volume during the error time frame, re-output the last error-free sample, or interpolate between valid samples to construct an artificial replacement for the error-filled data.

TRANSFORM TRADE-OFFS

The list of codec trade-offs begins with and often centers on the techniques used in time-to-frequency transforms. Your eye can quickly scan an entire digital image horizontally and vertically and rescan pixels as greater detail emerges. Listening to audio, on the other hand, is fundamentally a one-shot, sample-by-sample sequential process. Streaming delivery requires a low-latency delay from when the first incoming data appears at the audio receiver until the music begins to play. Also, a playback device often has insufficient temporary memory to store an entire uncompressed audio file. These factors suggest that an algorithm that simultaneously transforms all of a given audio file's data samples from the time to the frequency domain (akin to the full-image location-to-frequency wavelet transform at the heart of JPEG 2000), although technically feasible, is impractical for most audio applications (**Reference 3**).

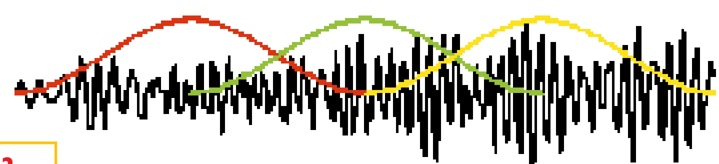
Other variables also factor into your choice of a transform approach, as well as the number of transformed samples

Figure 1



These high-level functional blocks are common to many perceptual lossy encoders and decoders (courtesy Creative Technology).

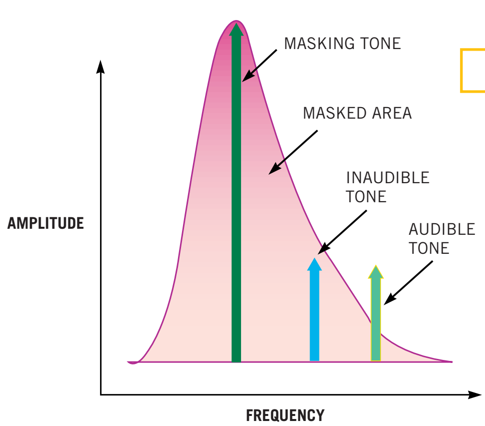
Figure 2



The encoder divides the sample set to be compressed into multiple overlapping windows which it time-to-frequency transforms (courtesy Berkeley Design Technology Inc).

(the window size) that make up each coefficient set. Time and frequency are inversely proportional, and fixing one limits the other. For transform filter banks, this time-versus-frequency trade-off means that the more frequency information they create, the less time resolution they have. The more samples you transform at once, the more accurate the frequency detail becomes and the fewer the total number of transforms you need to calculate for a given-duration audio clip. The longer the sample window, however, the less accurately those frequencies get reallocated to the time domain during decoding. Choose too large a window, and music transitions become muddled and distorted. On the other hand, overall music presentation might be "brighter" than that of a small-window alternative. Most music transitions are gradual, and, in many cases, a high percentage of the overall audio energy concentrates in lower frequency bands. When these assumptions prove false, however, the differences between algorithms most clearly emerge (see **sidebar** "Lend me your ears—and your eyes)

Figure 3



Frequency masking enables the encoder to discard data that would be inaudible even if it existed in the decoded digital-audio bit stream (courtesy Creative Technology).

The time-versus-frequency trade-off manifests itself in echo (**Figure 6**). Abrupt audio transitions, such as the crash of a cymbal or the shatter of breaking glass, create quantization noise that spreads through all of the samples in the window. Echo manifests itself as a colored noise burst that precedes and follows the onset of a transition. If the window is small enough, temporal masking can obscure the added noise before—that is, *pre-echo*—and after the transition. Echo artifacts are of most concern before the transition because temporal-masking effects extend much further beyond a tone than before it.

How do you combat pre-echo? One common approach transforms the incoming samples into multiple sub-bands of frequency data, versus one large coefficient set. This technique restricts the quantization noise to a narrow frequency range, versus leaking it throughout the full frequency spectrum. Comparatively simple codecs subdivide the total frequency range of the incoming material into multiple same-sized subbands. More sophisticated approaches use different-sized subsets that mimic the critical frequency bands of the human ear, as well as taking advantage of the fact that the ear is most sensitive to information in the region of 4 kHz.

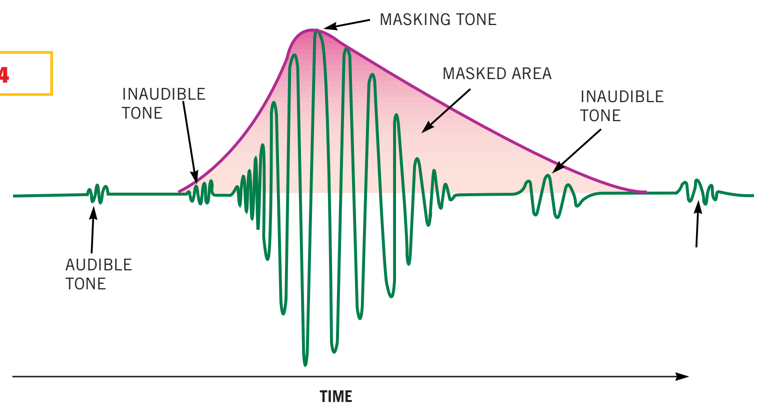
You could also incorporate support into the algorithm for multiple window-size options (**Figure 7**). The compressor selects a small sample window after detecting a transient and a larger window for more moderate passages. Keep in mind, though, that the more flexible an algorithm is, the more corresponding control (also known as side or ancillary) information you must put into the compressed bit stream to guide the decoder. These control bits take the place of audio sample data, reducing the bit stream efficiency. You should incorporate such flexibility, therefore, only if it increases the average resulting quality at a given bit rate versus a less flexible algorithm that uses more of the bit stream to store actual audio data.

Another technique for reducing the audibility of pre-echo and other noise and distortion effects involves reallocating the total bits per window to favor less quantization of more important frequency information and consequently more aggressive compression of (particularly) high-frequency data. Such an approach, *frequency noise shaping*, shifts the quantization noise away from the human hearing "sweet spot," 1 to 4 kHz. Temporal noise shaping is also possible, in which bits get reallocated from one window to another (via constantly varying window sizes), upon detection of a transition to reduce pre-echo effects. High-frequency data, both the most random on a sample-to-sample basis and among the least perceptible to the human auditory system, is often the primary focus of any audio-compression scheme, and different algorithms incorporate a variety of techniques to subdue it.

One brute-force but effective approach is to insert a "brick-wall" lowpass filter that obliterates all audio information higher than a certain frequency, such as 16 kHz. Another technique, joint, or intensity, stereo, sums the left and right channels' tones above a certain frequency and also stores the difference between the two channels (left minus right). During decoding, you use the sum and difference information to reconstruct left- and right-channel high-frequency data. Flexible algorithms can select between conventional left and right or sum and difference compression on a sample group-by-group basis, depending on which alternative results in the smallest data set.

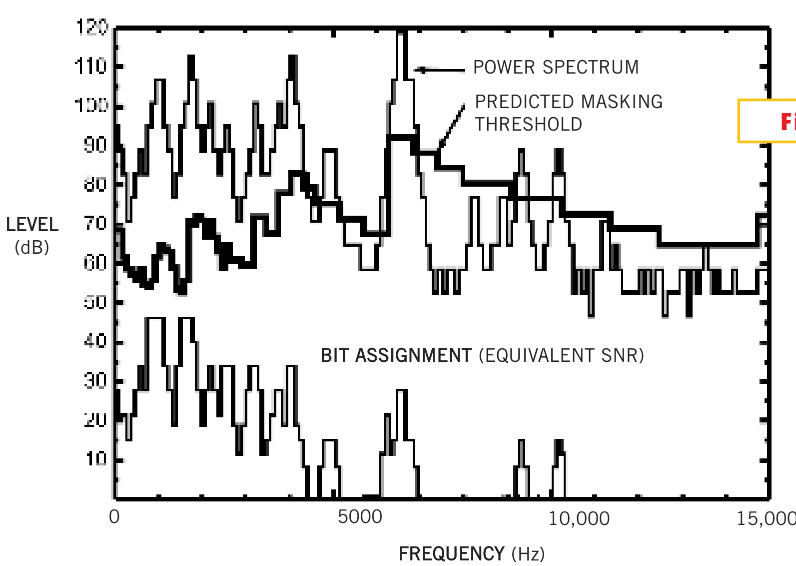
An extrapolation of joint stereo coding takes advantage of the fact that at high frequencies the human auditory system localizes sound based principally on the envelopes of signals reaching the ears versus the signals themselves. As frequencies rise above the sweet spot, the ear increasingly doesn't accurately follow interaural phase differences but instead relies on intensity analysis to determine source location. You can therefore selectively code the per-channel envelope information with greater precision than the carrier information, and, if necessary, you can selectively unite, or couple, multiple channels' carrier components into one. The decoder recombines the common carrier with each channel's envelope data to construct an approximation of the fine-grained high-frequency spectral components.

Figure 4



A tone of given frequency can mask quieter nearby frequency tones that follow and even precede it (courtesy Creative Technology).

Figure 5



The quantizer combines frequency- and temporal-masking information from the perceptual model along with the human auditory system's sensitivity curve across frequency to determine which frequency bands it can attenuate or even discard (courtesy The Audio Engineering Society and Dolby Labs).

ENOUGH THEORY!

Perhaps the most underrated strength of many of today's codecs is that, although they specify the encoded bitstream format and, therefore, the decoder function, they give encoder developers tremendous latitude about which techniques they employ and trade-offs they make. Documenting the compressed file format is necessary to avoid having decoders become obsolete. Consumers would be unhappy if their audio equipment quit working after a codec upgrade. A sufficiently flexible standardized file format allows for a variety of encoder optimizations targeting specific hardware platforms and digital-audio applications, and it also allows for evolutionary encoder improvements reflecting both increased learning and listener feedback.

The MPEG-1 audio standard is a good case study of this flexibility. The specification divides into three layers. (MP3 is a shorthand notation for MPEG-1 Layer 3 along with, sometimes, MPEG-2 Layer 3.) The specification stipulates that a decoder targeting a higher layer must also support decoding of lower layer bit streams. All three MPEG-1 layer encoders subdivide the data into 32 frequency subbanks as part of the transform process and support 32-, 44.1-, and 48-kHz sampling. The algorithms trace their heritage back to Musicam, and MPEG-1 Layer 1 is bit-stream-compatible with the PASC Musicam derivative, used in Philips' DCC system. MPEG-1 encoders allocate bits among multiple audio channels to maximize quality at a given bit rate, unlike some older algorithms, which encode a channel at a time and can't exploit channel-to-channel redundancy or trade bits between the channels on a sample set-by-set basis.

MPEG-1 Layer 1 employs a fixed-length 384-sample transform window and uses same-sized sub-bands generated by a 512-point polyphase filter. The algorithm takes advantage of frequency masking, generates 32- to 448-kbps bit streams, and has quality that is generally considered indistinguishable from that of an audio CD source at bit-stream sizes at and larger than 256 kbps. MPEG-1 Layer 2, used in DAB and CD-interactive, incorporates a three-times-longer window, which comprises 384 previous, 384 current, and 384 future samples. This sample combination helps MPEG-1 Layer 2 to exploit temporal redundancy. The transform is now a 1024-point polyphase

LEND ME YOUR EARS—AND YOUR EYES

I'm curious to find out which and how much of the bit-shrinking techniques codec developers use. So, over the next month, I'll be tossing a number of test tones, solo-instrument and vocal clips, and well-known song segments at the nearly three dozen lossy and lossless encoders and decoders in my possession.

I'm undertaking this task *not* to decide which algorithm is "better" or "worse" than the others. If one codec sounds significantly higher quality to me (who is less adept than most at hearing audio distortions or seeing dropped video frames and image-compression artifacts) than another at the same bit rate, I'll pass the news along. And if I hear or see extreme distortions of test tones, I'll also highlight these discrepancies. But, as I previously noted, I'm not enamored with quality testing that uses sine waves, white noise, short tone spikes, or other unnatural sounds. Benchmarking methods such as these imply that the person doing the study has preconceptions of encoder and decoder weaknesses and has created artificial signals designed to expose these weaknesses. Also, although you could argue that music is nothing more than a combination of multiple sine waves, music's inherent variability does a good job of hiding sins that an infinitely repeating fixed pattern would reveal. Just as the still or slow-transition display of MPEG-2 frames might lead you to an excessively negative opinion of MPEG-2 video quality, testing audio codes with inputs other than the voice, music, or other audio material that the codec will see in real life is of questionable value (**Reference A**).

The point of my study is to understand what the algorithm does as it compresses, purely to satisfy my (and your) engineering curiosity. How well do the lossless codecs approach the natural entropy of the source material and at what trade-off of required processing horsepower and memory and encoding and decoding speed? Are the lossy algorithms attenuating high-frequency information? Are they combining multiple channels into one above or below a certain frequency threshold? How well do they handle channel-to-channel phase differentials? How well do their chosen sampling-"window" sizes enable them to respond to fast-changing audio transitions? How aggressively do they exploit frequency and temporal masking? And how does this behavior change at different input sampling frequencies, and output bit rates?

For both lossless and lossy algorithms, I'll measure encoding and decoding speed, as well as CPU usage at multiple PC-processor frequencies. I'll also contrast the sizes of compressed files generated by the various lossless-codec alternatives. Finally, I'll compare the original audio files to those that the decoders output to ensure that the algorithms are indeed *lossless*.

For the lossy algorithms, I will compare spectrum-analyzer displays of the original audio information with those of the compressed versions and will also evaluate original-versus-compressed audio versions on an oscilloscope display to search for echo and other artifacts. All comparisons will be of two-channel source material, even for codecs such as Dolby Digital, which are better known in their multichannel implementations. Testing will occur at 64-, 96-, 128-, 192- and 256-kbps bit rates.

When necessary, I'll digitally transfer 44.1-kHz-sampled audio information between my PC and Sony D8 portable DAT deck using Digigram's VXpocket, Zefiro's ZA-2, and Zoltrix's Nightingale sound cards. Unlike most others, these cards don't resample or otherwise alter incoming and outgoing bit streams. Because the ATRAC codec resides exclusively within digital-audio recorders and players, not as PC-based software, I'll also try out Sharp's MD-MT15 MiniDisc unit. I'll use audio software, including Sonic Foundry's Sound Forge and XFX1 and XFX2 plug-ins and Syntrillium Software's Cool Edit Pro, to create test tones and analyze sound outputs. Check out the September issue of *CommVerge* (www.commvergemag.com) magazine for all the details.

Reference

A. Dipert, Brian, "Now hear this," *EDN*, Feb 3, 2000, pg 50.

filter. Bit streams range from 32 to 384 kbps, and rates of 192 kbps and higher achieve near-CD quality.

MPEG-1 Layer 3 retains Layer 2's 1152-sample window, as well as the polyphase filter for backward compatibility but adds a modified DCT filter. DCTs' advantages over DFTs include half as many multiply-accumulate operations and half the generated coefficients because the sinusoidal portion of the calculation is absent, and the DCT generally involves simpler math. The finite lengths of a conventional DCTs' bandpass impulse responses, however, may result in block-boundary effects. MDCTs overlap the analysis blocks and lowpass-filter the decoded audio to remove aliases, eliminating these effects. MDCTs also have a higher transform coding gain than standard DCTs, and their basis functions correspond to better bandpass response.

MPEG-1 Layer 3's DCT sub-bands are unequally sized and correspond to the human auditory system's critical bands. In Layer 3 encoders and decoders, particularly at lower bit rates, joint- and intensity-stereo high-frequency compression techniques commonly appear, and only Layer 3 decoders must support both

PICK A PROCESSOR FOR PERFECT PITCH

Now that you understand the basics of how audio codecs work and how complex they can be, you might be relieved to know that there's little need for you to code your own algorithms. You might be able to obtain the software you need either from the codec developer or from the silicon vendor, usually in processor-specific, compiled object code but sometimes even in assembly or high-level language source. Otherwise, you can contract the codec creation to a software-development and consulting company, such as Berkeley Design Technology Inc.

You *will*, however, have to pick your processor and corresponding memory. Your first decision will be between a media-optimized DSP, which for all but the simplest user and system interfaces requires a separate microcontroller, and a more general-purpose microprocessor, which may have insufficient hardware-acceleration and instruction-set robustness to run codecs with adequate speed and power stinginess.

Once you make the processor-versus-DSP selection, you've got more tough decisions to tackle. Is a 16-bit architecture adequate, perhaps with double-precision instruction assistance? Should you go with a 24-bit approach instead? Or, as samples reach 24 bits in the era of DVD Audio, will you need a full 32-bit processor to retain adequate audio fidelity through numerous interim arithmetic calculations and other data manipulations? With all other factors equal, wider data words translate to better numeric fidelity but at the cost of greater memory usage, higher chip cost, and higher power consumption.

Should you go with a fixed- or floating-point approach? Remember that the number of bits represents each sample's dynamic range, and the number of samples-per-second represents the nonaliased frequency range. Human hearing generally has a 120-dB dynamic range–from the thermal noise of electrons hitting your eardrum to the sound-intensity level that begins to cause pain. Every added bit doubles the absolute dynamic range, or adds 6 dB, to the SNR. You get a 96-dB dynamic range with 16-bit sampling, whereas a 32-bit IEEE floating-point number with a 24-bit mantissa translates to a whopping 1530 dB.

Floating-point processors are generally more expensive, but they can encode and decode in fewer instructions and may be simpler to program. However, those instructions often take longer to execute than their integer counterparts. If you use a floating-point data format, you still need to convert from fixed-point integer at the A/D-converter stage and back to integer at the D/A converter.

Manufacturers can incorporate features that will enhance the CPU's or DSP's audio-codec capabilities. Zero-overhead looping, bit-twiddling, and bit-reversed addressing are all useful in unscrambling a frequency transform's output. Some DSPs now integrate multiple on-chip multiply-accumulate units. Hardware-assisted RLE and Huffman or arithmetic coding and decoding support is also useful for bit-stream generation and parsing, whereas indexed addressing finds use in vector-quantization algorithms.

Audio encoding and decoding are only parts of the task your processor must undertake. Even with two-channel stereo configurations, you might need to support the HDCD algorithm, which hides control information in sample LSBs and uses other techniques to deliver an oversampled 20-bit equivalent dynamic range. Tone-manipulation options include loudness control and bass boost, as well as shelf, parametric, and or multiband graphic equalizers. Virtual surround-sound processing and pitch and time scaling for fast forward and rewind effects also gobble up MIPS. With media security now a hot topic, you need to comprehend encryption and decryption, as well as watermark generation and detection. Your customers might want you to support voice-quality codecs and MIDI. And, for recording devices, automatic level control, and multiple-source mixing capabilities are important.

Turn your focus to the home-theater stack or to DVD Video playback in automobiles and the additional audio-processing requirements skyrocket. For example, karaoke incorporates echo and reverb (to make you sound as good as you think you do in the shower) and pitch and time scaling (to ensure that you sound like you're singing in tune, despite the fact that you're not). Multichannel-to-two-channel downmixing, along with phantom-center and -surround modes, is necessary to support simple speaker configurations and older amplifiers. And THX processing, including THX Crossover, Bass Peak Management, Loudspeaker Position Time Synchronization, Re-Equalization and Timbre Matching can itself bring some DSPs to their knees.

Check out www.bdti.com/audiocomp.htm for a table, which Berkeley Design Technology Inc created for *EDN*. The table analyzes processor and memory types and speeds required to support various codecs. BDTI will also provide an interactive discussion forum at this site where you can ask questions, post comments, and contribute your own results, and the company will frequently update and enhance the table.

References

A. Cravotta, Nicholas, "You say you want a revolution?" *CommVerge*, June 2000, pg 38.

B. Cravotta, Nicholas, "The Internet audio revolution," *EDN*, Feb 3, 2000, pg 101.

C. Nass, Richard, "Programmability is key to portable digital audio players," *Portable Design*, January 2000, pg 73.

CBR and VBR bit streams. (However, many Layer 1 and 2 decoders also handle VBR.) Finally, Layer 3 encoders Huffman-code the quantized coefficients before archiving or transmission for additional lossless compression. Bit streams range from 32 to 320 kbps, and 128-kbps rates achieve near-CD quality, an important achievement when dual-channel ISDN audio delivery was believed to be the future high-bandwidth pipe to the home.

MPEG-2 BC audio, also available in three layers, leverages the MPEG-1 heritage in an evolutionary manner. It adds support for lower mono and stereo, 16-, 22.05-, and 24-kHz sampling rates and corresponding bit rates as low as 8 kbps. MPEG-2 also supports more-than-two-channel audio, again in a backward-compatible manner. The first two channels, within the primary bit stream, contain left and right audio information, as well as matrix-encoded left-surround, right-

NO LOSS, YOUR GAIN

Most digital-audio applications value small files over absolute best quality. However, a small but vocal segment of the audio-listening public refuses to accept any degradation of its CD-audio source material or 48-kHz-sampled DAT recordings. These same audiophiles are the target market for the upcoming DVD Audio (with 192-kHz sampling rates and 24-bit samples) and SACD technologies.

One rarely discussed aspect of lossy compression is that its effects are additive. *Cascading*—repeatedly decoding and then re-encoding an audio clip or transcoding it from one format to another—progressively degrades the sound quality, just as with analog-audio copies and JPEG image files. During audio creation, then, it's important to keep the various sound clips you're editing and mixing at their highest quality levels. For both audiophiles and content creators, lossless compression is an attractive tool for retaining audio excellence while minimizing file sizes (**Table A**).

Traditional compression schemes, such as those in PKZIP and WinZIP, aren't necessarily the best approaches for audio data, because they assume highly random source files. Audio has a higher than normal percentage of sample-to-sample redundancy. As an indication of this discrepancy, another popular compression utility, RAR, includes a multimedia option, and it automatically switches algorithms if it detects that it's compressing audio, image, or other highly redundant material.

RLE and arithmetic coding are common means of compressing low-entropy data. Many lossy codecs, which eliminate perceptual redundancy, also incorporate these techniques, eliminating statistical redundancy, to further compress quantized frequency coefficients. *Differencing*, another lossless approach, stores the variation from one sample to another versus encoding the absolute value of each sample. This technique is tricky, because you don't want to waste unnecessary bits in situations having little to no sample-to-sample variation. On the other hand, you want to retain sufficient bit resolution to capture the occasional significant sample-to-sample variation without resorting to less-than-perfect lossy quantization. The documentation for DAKX compression gives some interesting ideas on how to accomplish this balancing act.

Taking differencing techniques to the next step, you might choose to use a series of past sample values to predict the next value and then store the difference, or residue, between this prediction and the actual sample. Predictive, or delta, codecs differ from each other mainly in the predictive algorithm they use and the number of past samples the algorithm incorporates in its calculation.

TABLE A–REPRESENTATIVE LOSSLESS-COMPRESSION ALGORITHMS

Codec	URL
DAKX	www.dakx.com
LPAC	www-ft.ee.tu-berlin.de/~liebchen/lpac.html
LTAC	www-ft.ee.tu-berlin.de/~liebchen/ltac.html
MKW	http://home.att.net/~mkw
MLP	www.meridian.co.uk/p_mlp.htm
Monkey's Audio	www.monkeysaudio.com
MUSICompress	http://members.aol.com/sndspace, www.gadgetlabs.com/wavezip.htm
RAR	www.rarsoft.com
RKAU	http://members.tripod.com/~malcolmt
Shorten	www.softsound.com/Shorten.html
SPS	www.krishnasoft.com/sps.htm, www.pegasusimaging.com/sound.html
Waveform Archiver	www.simtel.net/pub/simtelnet/msdos/arcers/wavarc10.zip
Wavpack	www.wavpack.com

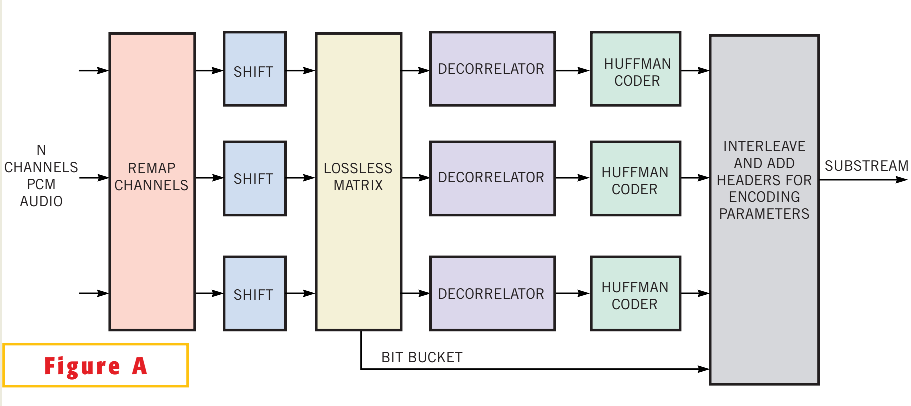


Figure A

The MLP encoder leverages a variety of techniques to minimize file size while retaining full audio fidelity (courtesy Meridian Audio).

surround, and center-channel data. MPEG-1 decoders downmix and output conventional two-channel audio. MPEG-2 decoders use the primary bit stream, plus the somewhat-redundant surround- and center-channel data in additional bit streams to decode the full five-channel mix. MPEG-2 BC also supports multi-language audio, particularly important in Europe, where the MPEG standardization efforts were based.

The MPEG-2 specification also comprehends the AAC NBC algorithm. AAC supports as many as 48 distinct audio channels and 8- to 96-kHz sampling frequencies and retains MP3's MDCT but drops the backward-compatible polyphase filter. Similar to MPEG-1's three layers, the AAC specification includes main, LC, and SSR profiles. LC and SSR target implementations with restricted processing capabilities. Depending on which profile you use, you have access to a number of encoding enhancements.

Some algorithms take prediction to multiple derivatives, predicting not only the sample but also its residue, its residue of residue, and so forth. Many lossless codecs include the option for additional lossy compression, which the codecs usually implement by dropping bits to reduce sample size in exchange for decreased dynamic range.

SoftSound's MUSICompress, an all-integer algorithm provided in ANSI C source as well as object code for several CPUs and DSPs, takes advantage of the fact that most audio is oversampled and therefore contains more low-frequency components than high-frequency components. Inventor Al Wegener developed an approach that separates the original audio into *subset* samples (every nth—usually every second—sample) and *removed* samples. He then applies a simple approximation routine (a four-tap FIR filter) to the subset samples to approximate the removed samples. The error between the removed samples and their approximation is usually small, and MUSICompress therefore replaces a sampled data stream with the subset array and the error array.

Subset samples then run through a first, second, and third derivative generator. The algorithm determines which derivative creates the smallest subset array. Normally, the lower the energy, or frequency, of the original audio, the higher the derivative that results in the best bit-packing. Wegener estimates that the MUSICompress object library for Microsoft Visual C v6.0 takes 60 kbytes. On a Motorola 563xx DSP, MUSICompress requires 700 words of program memory and 1800 words of data memory. It takes 1.7 MIPS to compress and 1.4 MIPS to decompress 44.1-kHz-sampled, dual-channel, 16-bit audio. Wegener also estimates that an all-hardware MUSICompress encoder would require 4700 logic gates and 20,500 memory bits, and the corresponding decoder would require 3800 logic gates and 1.5 kbits of RAM.

Shorten, another lossless codec popular with live music tapers and traders, performs linear prediction, (*autoregression*) based on autocorrelation coefficients. It can perform either a general prediction-and-error-calculation combination:

$$\hat{s}(t)=\sum_{i=1}^{p} a_i s(t-i).$$

$\epsilon(t)=\hat{s}(t)-\hat{s}(t),$

or a more restrictive set of prediction derivatives:

$\hat{s}_0(t)=0,$
$\hat{s}_1(t)=s(t-1),$
$\hat{s}_2(t)=2s(t-1)-s(t-2),$
$\hat{s}_3(t)=3s(t-1)-3s(t-1)-3s(t-2)+s(t-3),$
$\epsilon_0(t)=s(t),$
$\epsilon_1(t)=\epsilon_0(t)-\epsilon_0(t-1),$
$\epsilon_2(t)=\epsilon_1(t)-\epsilon_1(t-1),$ and
$\epsilon_3(t)=\epsilon_2(t)-\epsilon_2(t-1),$

from which the algorithm selects the best estimate. It then Huffman-codes resulting error values, which it assumes to be uncorrelated. The default block size, which you can override, is 256 samples. Too short a set results in excessive calculation and parameter-transmission overhead, and too long of a set, due to excessive changes in signal characteristics, results in a poor signal model.

Perhaps the best-known lossless audio codec is MLP (**Figure A**). The MLP algorithm enables a DVD Audio disc to store 77 to 133 minutes of six-channel, 24-bit, 96-kHz sampled sound. The 77-minute figure is the same as that of a two-channel, 44.1-kHz audio CD. MLP-enhanced DVDs can also hold 122 to 136 minutes of 192-kHz, dual-channel info.

Beyond the normal lossless techniques to reduce sample-to-sample and channel-to-channel correlation, MLP also shrinks the bit stream, if it detects, for example, that the source material uses 16-bit samples or a 44.1-kHz sampling frequency. SACD (**Figure B**) also incorporates a lossless codec—in this case, the Philips-developed Direct Stream Transfer algorithm, which delivers 2-to-1 average lossless compression.

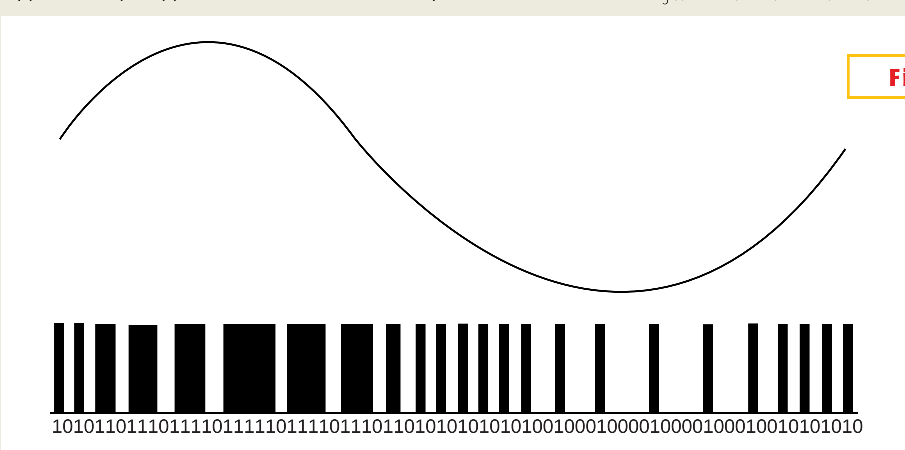


Figure B

SACD's Direct Stream Digital process uses 2.8224-MHz sampled, single-bit, pulse-density modulation to encode audio information (courtesy Sony).

Like MP3, AAC uses Huffman coding, quantization and scaling, and joint- and intensity-stereo techniques, all improvements of their implementation in MP3. AAC also employs forward and backward adaptive prediction to enable storage of only the residual difference between actual samples and algorithmically calculated sample estimates. AAC supports both 2048- and 256-sample-long blocks and employs temporal-noise shaping to reduce pre-echo and other quantization noise effects at low bit rates. Listeners generally agree that AAC produces near-CD quality beginning at 96-kbps rates.

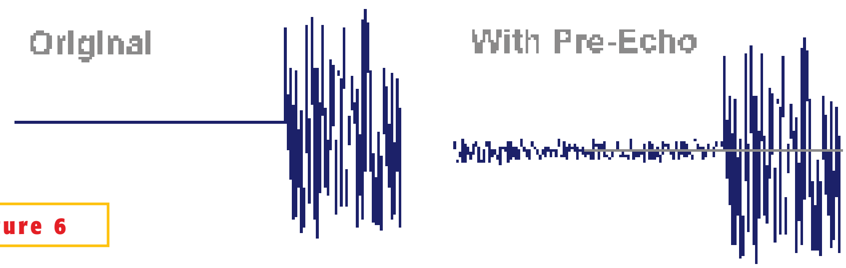


Figure 6

Pre-echo is a side effect of the quantization and time-to-frequency conversion and, if the sample window is too wide, can result in audible distortion (courtesy Berkeley Design Technology Inc).

LESSER KNOWN, BUT HEAR THEM OUT TOO

AAC forms the high-bit-rate, high-fidelity audio foundation of the MPEG-4 specification in a further enhanced derivative of its MPEG-2 predecessor. MPEG-4 AAC improves the prediction algorithms and incorporates *perceptual-noise substitution*. With an eye toward multiprocessor encoders, MPEG-4 AAC uses the BSAC kernel tailored for scalable systems. For lower bit-rate, high-fidelity audio, MPEG-4 turns to the TwinVQ algorithm and also supports other codecs for reduced-fidelity transmissions, such as voice. TwinVQ and a few other codecs, including, according to rumor, a Voxware-developed approach that Microsoft uses in its closely guarded WMA, employ vector quantization. This technique, like the DCT in JPEG and MPEG, also finds use with still- and video-image compression.

In vector quantization, both the encoder and the decoder have identical "code books" containing vector sets of coefficients. After calculating a coefficient set based on a window-sample series, the encoder searches for the closest approximation in its code book and, instead of sending the actual coefficients, sends the much shorter code-book index. The decoder uses this index to find the same vector in its code book, which it then retransforms from the frequency domain to the time domain. The quality of results with vector quantization depend highly on the robustness of the code book and on how well the encoder determines the best code-book match. Vector-quantization encoding also takes significantly longer than perceptual coding, all other factors being equal, though the incremental performance that the decoder demands is usually trivial.

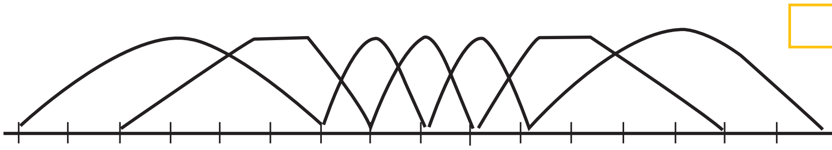
Other perceptual coders, including ATELP, QDesign, and RealAudio, employ conceptually similar techniques to those for MPEG-1 and AAC with a few unique twists. Fraunhofer has developed MPEG-2.5, a proprietary variant of MPEG-2 that further lowers the allowable sampling rates to 8, 11.05, and 12 kHz. Fraunhofer has also developed LD-AAC, a specialized AAC encoder and an example of the encoder flexibility you can achieve when only the bit stream is standardized. LD-AAC doesn't necessarily give equivalent quality to full-featured AAC at a given bit rate, but it compensates with 20-msec maximum encoding delay. Such a feature would be useful in, for example, two-way live communication.

Dolby Laboratories, Fraunhofer, Lucent Technologies, and Sony together own the fundamental patents on which AAC is based. Not surprisingly, in creating AAC, each company drew from its experience in developing its own codecs, and each has further developed its proprietary alternatives. Dolby Laboratories' AC-3, which is the perceptual codec inside Dolby Digital, and AC-2 are best known in their multichannel versions. Fraunhofer did much of the R&D work

ACRONYMS

AAC: Advanced Audio Compression
ADSL: asymmetrical digital-subscriber line
ATELP: adaptive-transform-excited-linear prediction
ATRAC: adaptive-transform acoustic coding
BC: backward compatible
BSAC: bit-sliced arithmetic coding
CBR: constant bit rate
Codec: compressor/decompressor, also sometimes used to define a single-chip A/D-plus-D/A converter
DAB: digital-audio broadcast
DCC: digital compact cassette
DCT: discrete cosine transform
DFT: discrete Fourier transform
DTS: Digital Theater Systems
DVD: digital versatile disc
EPAC: Enhanced Perceptual Audio Coder
FFT: fast Fourier transform
FIR: finite-impulse response
HDCD: high-definition compatible digital
IBOC: in-band on-channel
JPEG: Joint Photographic Experts Group
LC: low complexity
LD-AAC: Low Delay Advanced Audio Compression
LPAC: lossless predictive audio compression
LSB: least significant bit
LTAC: lossless transform audio coding
MDCT: modified discrete-cosine transform, synonymous with TDAC
MIDI: musical-instrument digital interface
MIPS: millions of instructions per second
MLP: Meridian lossless packing
MPEG: Moving Picture Experts Group
NBC: non-backward compatible
PAC: Perceptual Audio Coder
PASC: Perceptual Audio Sub-band Coding
PCM: pulse-code modulation
RLE: run-length encoding
SACD: super audio compact disc
SPS: sound-processing software
SSR: scalable sampling rate
TDAC: time-domain alias cancellation, synonymous with MDCT
TwinVQ: transform-domain weighted interleaved vector quantization
VBR: variable bit rate
WMA: Windows Media Audio

Figure 7



Variable-window sizes are one key means of combating pre-echo (courtesy Berkeley Design Technology Inc).

that resulted in MPEG-1 and, therefore, MPEG-2. Sony's ATRAC codec, which it implements in MiniDisc recorders and players, runs at a dual-channel bit rate of 292 kbps and has gone through numerous bit-stream-compatible improvements since its 1992 introduction. The version of ATRAC that Sony's Music Clip implements uses a 128-kbps bit stream.

Technology from Lucent Technologies' PAC codec also made its way into AAC, and Lucent has made further improvements to come up with ePAC. The Lucent Digital Radio subsidiary put an interesting spin on ePAC to come up with a codec that is one of the two finalists for IBOC digital-radio broadcasting; the other, which USA Digital Radio advocates, is AAC. Lucent Digital Radio transmits four simultaneous 32-kbps streams, any of which produce an audibly understandable presentation when they reach their destination. Multiple received streams incrementally improve the quality, and if you can tune in all four streams, you get CD-transparent, two-channel reception, according to the company. Lucent Digital Radio's implementation of ePAC conceptually more closely mimics the behavior of traditional-analog radio and is unlike, say, a digital-cellular phone, with which users experience a binary "all-there" or "silence" reception, which is not what happens with the progressive signal degradation of an earlier generation analog cellular phone.

MULTICHANNEL MAKES AMAZING MUSIC

Expand the audio beyond conventional two-channel stereo, and the difficulty of squeezing a high-quality presentation into a reasonably sized bit stream increases. Multichannel MPEG-2 has

FOR MORE INFORMATION...

For more information on products such as those discussed in this article, enter the appropriate numbers at www.ednmag.com/infoaccess.asp. When you contact any of the following manufacturers directly, please let them know you read about their products in *EDN*.

Audio Processing Technology
+44 (0) 2890 371110
www.aptx.com
Enter No. 336

DAKX
1-919-542-5785
www.dakx.com
Enter No. 337

Dolby Labs
1-415-558-0200
www.dolby.com
Enter No. 338

DTS
1-818-706-3525
www.dtsonline.com
Enter No. 339

Fraunhofer Institute
+49 (0) 9131 / 776 0
www.iis.fhg.de
Enter No. 340

K&K Research
(045) 38334048
http://kk-research.hypermart.net
Enter No. 341

Krishna Software
1-732-549-3097
www.krishnasoft.com
Enter No. 342

Lucent Technologies
1-908-582-8500
www.lucent.com
Enter No. 343

Meridian Audio
+44 1480 52144
www.meridian.co.uk
Enter No. 344

Microsoft
1-425-882-8080
www.microsoft.com
Enter No. 345

Musicam USA
1-732-739-5600
www.musicamusa.com
Enter No. 346

Nippon Telephone and Telegraph
www.ntt.co.jp
Enter No. 347

Microsonics
1-510-475-8000
www.hdcd.com
Enter No. 348

QDesign
1-604-688-1525
www.qdesign.com
Enter No. 349

RARSoft
www.rarsoft.com
Enter No. 350

RealNetworks
1-206-674-2700
www.realnetworks.com
Enter No. 351

SoftSound Ltd
+44 1223 421754
www.softsound.com
Enter No. 352

Sony
1-201-930-1000
www.sony.com
Enter No. 353

Soundspace Audio
1-408-221-1191
http://members.aol.com/sndspace
Enter No. 354

OTHER COMPANIES MENTIONED IN THIS ARTICLE

AT&T
www.att.com
Beadgame.com
www.beadgame.com
Berkeley Design Technology Inc
www.bdti.com
Creative Technology
www.creative.com
Digigram
www.digigram.com
Lucasfilm THX
www.thx.com
Motorola
www.motorola.com
Philips
www.philips.com
Sharp
www.sharp.com
Sonic Foundry
www.sonicfoundry.com
S Systems
www.s-systems-inc.com
Syntrillium Software
www.syntrillium.com
USA Digital Radio
www.usadr.com
Voxware
www.voxware.com
Zefiro Acoustics
www.zefiro.com
Zoltrix International
www.zoltrix.com

SUPER INFO NUMBER
For more information on the products available from all of the vendors listed in this box, enter No. 355 at www.ednmag.com/infoaccess.asp.

found use primarily as the preferred audio codec for the European DVD Video standard. Dolby Digital is the most common multichannel audio codec in the United States and the codec for the US digital-television standard.

Dolby Digital's predecessor is Dolby Stereo. (This name is its movie-theater version; the home-theater variants were Dolby Surround and, later, the higher-quality Dolby Pro Logic.) Dolby Stereo matrix-encoded additional audio channels within the normal front-left and front-right stereo signals and achieved its first widespread success in 1976 with *Star Wars*. By the late 1980s, though, the technology was beginning to show its age, and Dolby Surround was also inappropriate for surround-sound high-fidelity music reproduction. The rear surround was monophonic—that is, it didn't provide separate left and right channels—and had a restricted 100- to 7000-Hz frequency range. Matrix-encoding provided less precise spatial effects than true distinct additional channels would allow. And the subwoofer was a lowpass-filtered version of the combination of other channels, again not a distinct channel of its own.

Dolby Digital, which first appeared in theaters with 1992's *Batman Returns*, fixed many of these shortcomings. By employing perceptual-encoding techniques, Dolby Labs was able to squeeze five distinct, full-range channels—front left and right, center, and rear left and right—plus a dedicated, low-frequency effects channel (hence, the common "5.1" moniker) into a bit stream averaging 384 kbps. The Dolby Digital encoder attenuates the subwoofer channel with a brick-wall lowpass filter at frequencies greater than 120 Hz. A flexible-allocation technique assigns bits both across frequencies and across channels as needed from a common bit pool and exploits both intrachannel and interchannel frequency and temporal-masking effects. This approach realizes further coding gain by employing joint- and intensity-stereo techniques—separating and independently coding high-frequency carrier and envelope information.

The Dolby Digital documentation includes an interesting rule of thumb: The average bit demand of multiple channels using perceptual compression is roughly proportional to the square root of the number of channels. AC-3, being an older codec than MPEG-1 or follow-ons, required, in Dolby's estimation, 128 kbits for high-quality, single-channel reproduction: $128\times\sqrt{5.1}=289$ kbps, comfortably within the 384-kbps bit stream, which includes not only sample coefficients but also dialogue and other level-normalizing signals, as well as suggested volume-compressor control data for limited-dynamic-range listening environments. Dolby Digital decoders not only must decode the full 5.1-channel output, but also must downmix to other receiver and speaker configurations, including conventional two-channel stereo and Dolby Surround.

Dolby Digital's primary multichannel-surround-sound competition, DTS, entered the public consciousness in a big way with 1993's *Jurassic Park*. DTS, as it first appeared in theaters, used the apt-X100 codec, whose delta encoder divides the incoming signal into four sub-bands and delivers a 4-to-1 compression ratio. DTS on audio CDs, laser discs, and DVD Video discs uses the Coherent Acoustics codec, whose encoder employs a 32-sub-band frequency transform. Although frequency- and temporal-masking data-reduction techniques are an optional part of Coherent Acoustics, the algorithm doesn't usually use them at CD, laser-disc, and DVD bit rates. DTS instead commonly encodes at 1.509 Mbps to create a claimed higher quality audio presentation. Coherent Acoustics' compression still enables a full-fidelity, six-channel, 20-bit audio stream to fit into roughly the same CD space that a two-channel, 16-bit PCM audio alternative requires.

Comparisons of DTS and Dolby Digital presentations of the same movie soundtrack can sometimes reveal subtle DTS enhancements, especially when you use high-quality DVD players, amplifiers, and speakers in ideal listening environments. But is the difference worthwhile, particularly when some DTS movies have fewer features than their Dolby Digital alternatives? Filmmakers sometimes use the extra DVD space that Dolby Digital frees up to store a two-channel PCM version of the soundtrack; additional language versions; a director's commentary; behind-the-scenes documentaries; or other extra audio, video-, and still-image information.

Both THX and DTS have announced "EX" versions of their formats, which matrix-encode a middle rear channel for supposedly more realistic surround channel-to-channel transitions. EX debuted in 1999 with movies such as *Star Wars: Episode 1* and *The Haunting*. DTS has also developed DTS-ES Discrete, a full 6.1-channel format that adds a distinct surround back channel. The surround-sound world has become even more crowded of late with the news that Japan has standardized a multichannel variant of AAC for its digital-TV format.□

You can reach Technical Editor Brian Dipert at 1-916-454-5242, fax 1-916-454-5101, e-mail bdipert@pacbell.net.

REFERENCES

1. Dipert, Brian, "Now hear this," *EDN*, Feb 3, 2000, pg 50.

2. Dipert, Brian, Image compression article addendum, www.ednmag.com/reg/1998/061898/wavefrc.htm.

3. Dipert, Brian, "Compression puts images on a diet," *EDN*, June 18, 1998, pg 71.

4. Bier, Jeff, and Jennifer Eyre, "DSPs court the consumer," *IEEE Spectrum*, March 1999, pg 47.

5. Partkya, Jeff, "Stop, hey, what's that sound," *EMedia Professional*, January 1999, pg 42.

6. Ranada, David, "Download show-down," *Stereo Review's Sound & Vision*, September 1999, pg 98.

7. Ranada, David, "MP3 sound quality?" *Stereo Review's Sound & Vision*, April 1999, pg 40.

ACKNOWLEDGMENTS

Thanks to Jeff Bier from Berkeley Design Technology Inc (who also reviewed an early article draft), Dana Massie from Beadgame, and John Strawn from S Systems for their presentations at the 2000 Embedded Processor Forum, and to Dave Rossum from Creative Technology for his presentation at the 2000 WinHEC. Both of these classes and the literature that accompanied them were useful in balancing the multitude of understandably biased vendor perspectives I received while researching this article.