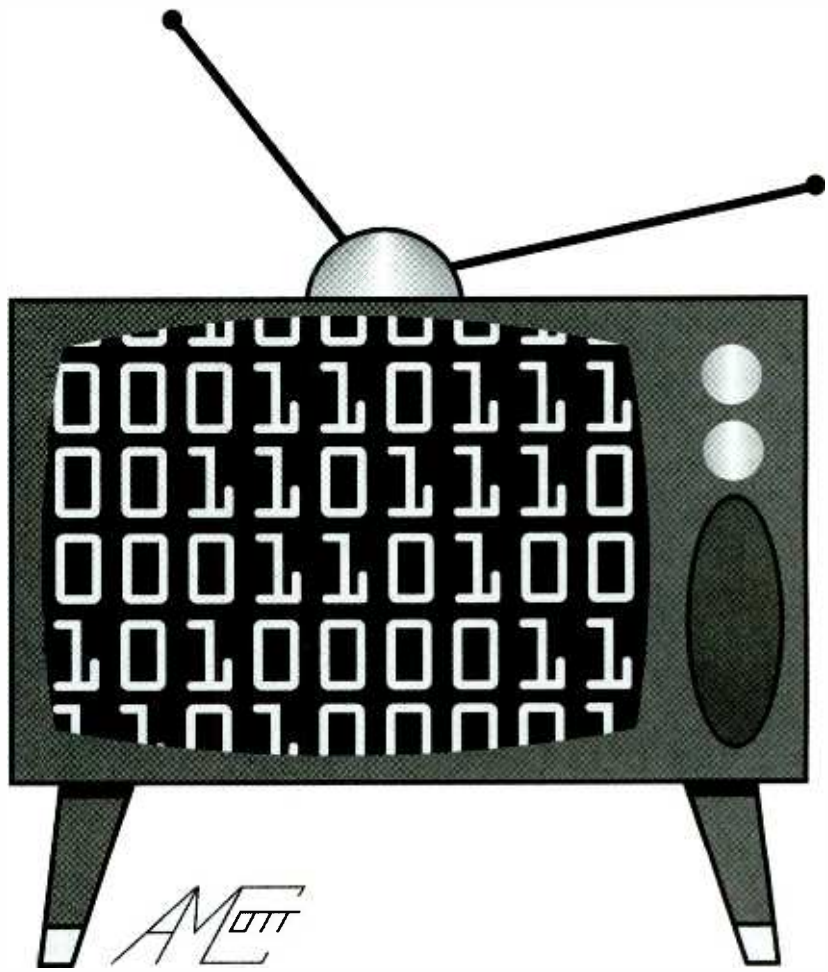
LOOKING FORWARD TO DTV

An advanced peek at television's (not too distant) future. First of two parts.

GEOPHREY J. MCCOMIS



What do you know about video? If you're a typical electronics enthusiast, you probably know a thing or two about NTSC. If you're a videophile, you may have a good idea of what "PLUGE" stands for. If you know what a "color frame" is, you might be a video guru. Well, hold on to your merit badges—it's all about to change!

In 1953, the FCC adopted the NTSC color-video standard. Here in the USA, television's core technology hasn't changed much since—until now. In December 1996, the FCC announced its acceptance of the Advanced-Television-Systems Committee's (ATSC) Digital-Television (DTV) standard. It later revealed a changeover plan that has already brought DTV to within reach of 60% of American television households, with the complete termination of analog broadcasting targeted for 2006. So how will TV look and sound, and how will it work in the next century?

As adopted, the standard describes how the new system will work without dictating its looks. DTV broadcasts occupy the same 6-MHz channels that have been used for NTSC, but instead of a single analog program, they will deliver a full bouquet of digital-programming options. A given channel may offer anything from a single program in multiple formats to a constellation of unrelated audio, video, and data signals.

The ATSC document that describes the standard was almost entirely incorporated into FCC rules. The only item omitted was a chart that would have constrained the number of compression formats. In practice, this means that the demands of the marketplace and program-content developers will determine the available aspect ratios, resolutions, frame rates, and scanning formats. A DTV channel may carry anything from a single HDTV program (High-Definition TV, featuring a resolution of 1920 by 1080 pixels) to three or more standard definition programs (approximately equal in resolution to NTSC).

For better or worse, the new broadcasts will not be compatible with today's television sets. Although you won't be able to build a digital set anytime soon, you can enjoy a good look at the new technologies that make DTV a true creature of the 21st century.

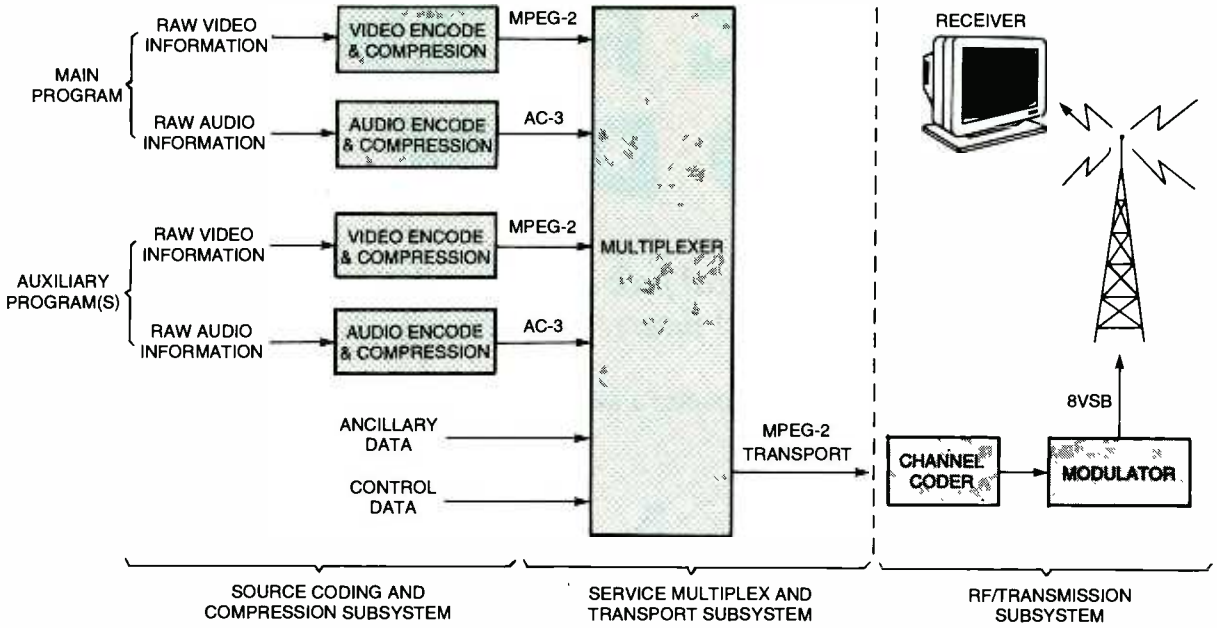


Fig. 1. The DTV terrestrial-broadcast system consists of three subsystems: Source Coding and Compression, Service Multiplex and Transport, and RF/Transport.

System Overview. Figure 1 portrays an overview of DTV. It consists of three basic subsystems: source coding and compression, service multiplex and transport, and RF/transmission. At the source coding and compression stage, audio and video are coded into digital samples and crunched according to Dolby AC-3 and MPEG-2 audio- and video-compression schemes, respectively. Together with some control data, they are then multiplexed and parceled into a single MPEG-2 data stream. Finally, error-correction information is added, and the combined data is modulated and transmitted. For open-air DTV broadcasting, a modulation mode known as *8 VSB*—a vestigial-sideband-modulation scheme with eight discrete amplitude levels—is used. With a 6-MHz channel bandwidth, the system can deliver about 19 Mbps of combined program data.

Video Compression. Delivering an HDTV program within a 19-Mbps data stream requires the use of severe compression; video data must be reduced by a factor of fifty-to-one or more. To achieve this reduction, DTV utilizes the Main Profile subset of the MPEG-2 video-compression standard from the Moving Picture Experts Group.

The MPEG-2 protocol describes a

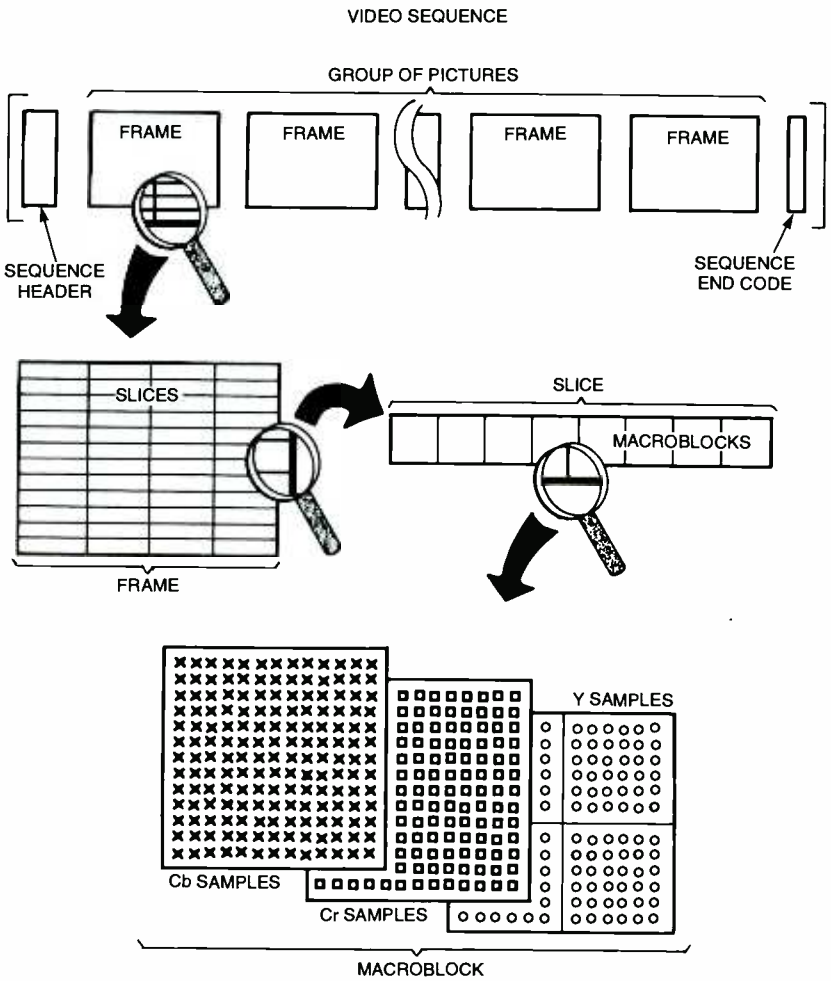


Fig. 2. An MPEG-video sequence is built up from blocks, macroblocks, slices, frames, and groups of pictures (GOPs, which are optional).

series of video-compression techniques that are designed to reduce the amount of data required to represent a video sequence without the compression being noticeable.

Figure 2 provides an overview of how video information is organized in MPEG-2. The largest organizational units are called *video sequences*. These may be of any length down to a single frame or picture. Each video sequence consists of a *group of pictures* (GOP), the next organizational layer. GOPs may also be of any length, but are not mandatory divisions and may be omitted entirely.

Frames (or pictures) are like the frames in NTSC video: a frame is one complete still image. Common frame rates for MPEG video streams include 60, 59.94, 30, 29.97, and 24 frames per second. Both interlaced and progressive-scanning types are supported.

An NTSC frame is comprised of two interlaced fields that are combined in the television receiver to form a single image on the screen. The two fields are called the *even field* and the *odd field*. The even field sends scan lines 2, 4, 6, and so on, while the odd field sends—if you haven't guessed by now—scan lines 1, 3, 5, etc. Since it takes 1/60 of a second to send one field, only 30 frames per second are possible. In a progressive-scan frame, *all* of the scan lines are sent in order within one field. The result, on televisions that can handle the bandwidth, is a picture that can rival a computer screen in terms of sharpness. You might be wondering why NTSC doesn't allow progressive scan. The answer is simple: we're talking early 1950s technology. The systems back then didn't have the "horsepower" to create such a picture. In fact, a 525-line frame was considered the ultimate state-of-the-art in available picture-display technology at the time.

Another subtle point about NTSC frames is that they are transferred in real time; that is, the time needed to send them is identical to the time needed to either scan or display them. As a result, an NTSC frame represents a period of time spanning about 1/30 of a second.

MPEG frames are processed as singular points in time like a series of snapshots. In this regard, they are similar to the frames of motion-picture film—hardly a surprise considering that MPEG comes from the motion-picture industry. An MPEG frame exists in memory as an instant of time, even if it will ultimately be parceled out a line at a time to a scanning-display device.

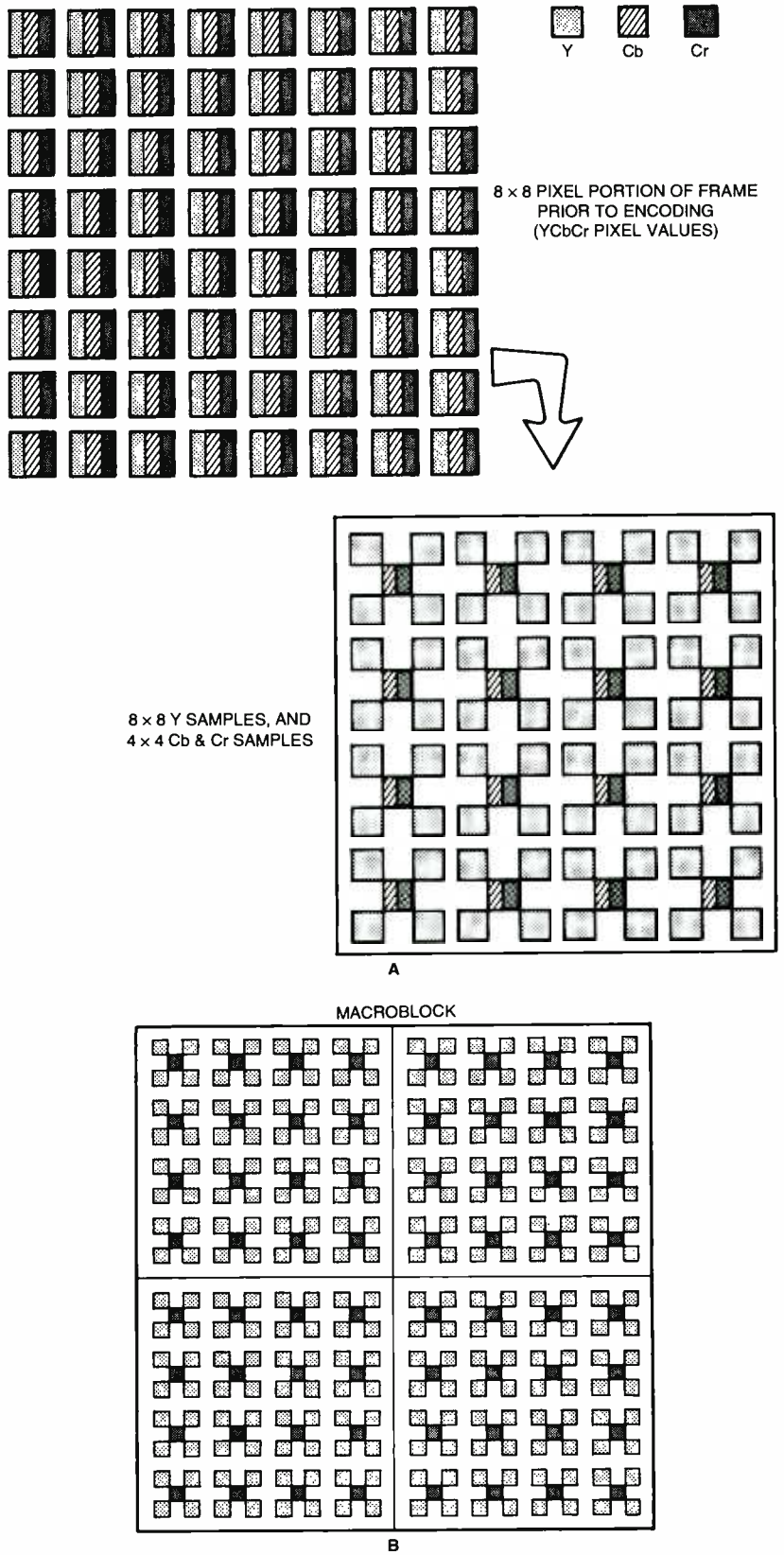


Fig. 3. Blocks and macroblocks are the foundation units of MPEG picture formation.

Anatomy of an MPEG Frame. To understand the composition of an MPEG video frame, start by imagining a single video picture prior to encoding. Think of this picture as a matrix of *pixels* (short for picture elements), like a computer's display. Each pixel contains information on the red, green, and blue (RGB) portions of its color.

Before MPEG encoding, the RGB pixel values are converted to *YCbCr* values. Just like NTSC, luminance (Y) corresponds to the combined red, green, and blue ratios that result in perceived shades of gray, or brightness levels. The Cb and Cr items are the *color-difference* values; Cb for the blue content and Cr for the red content. All together, the three signals convey a pixel's color characteristics. The relative values follow the standard color-video formula Y = 0.30R + 0.59G + 0.11B.

At the beginning of the encoding process, the YCbCr values are grouped into 8 × 8 blocks of samples as shown in Fig. 3A. For every Y-pixel value, a Y sample will be encoded. But the Cb and Cr values are averaged so that every four Cb or Cr values become a single sample. This loss of color resolution represents the first compression gain resulting in data savings. It is not objectionable because human vision is more sensitive to high-resolution luminance information than to the corresponding color information. Since four luminance samples are encoded for every Cb and Cr pair, four blocks of luminance samples, one block of Cb samples, and one block of Cr samples (from a common picture region) are grouped to form a single *macroblock* (Fig. 3B).

Now that we have an understanding of macroblocks, look back at Fig. 2. The macroblocks are further grouped with horizontally adjacent macroblocks to form *slices*. While slices constitute horizontal rows of picture information, they are unlike scan lines in that they are 16 samples high, and there can be several slices in a row.

Squeezing Out The Excess. Video is loaded with redundant information. MPEG compression combats redundancy on two fronts: spatial (or *intra*frame) and temporal (or *inter*frame). Spatial redundancy exists within a frame when significant portions of the picture look the same, such as when an object or background contains large areas of the same pattern, color, or luminance. Temporal redundancy exists from frame to frame when an object or background does not change in appearance or position over time.

The MPEG encoder may select one of three methods to represent a given frame in a sequence: *intraframe* coding (I frame), *predicted* coding (P frame), or *bidirectionally predicted* coding (B frame). While I frames are fully self-contained, P frames are encoded with reference to previous frames (areas of a P frame will be predicted from a previous frame). This eliminates the need to resend similar information from frame to frame. B frames are predicted bidirectionally with reference to previous and/or subsequent frames.

Prediction is MPEG's primary means of "squashing out" temporal redundancy. While P frames use only forward-prediction and intraframe coding, B frames can also take advantage of backward prediction and bi-directional prediction. In both cases, the coding decision is made on a macroblock level. Within a single B frame, for example, you might find different macroblocks encoded in each of the four possible ways (intra-, forward, backward, and bi-directional).

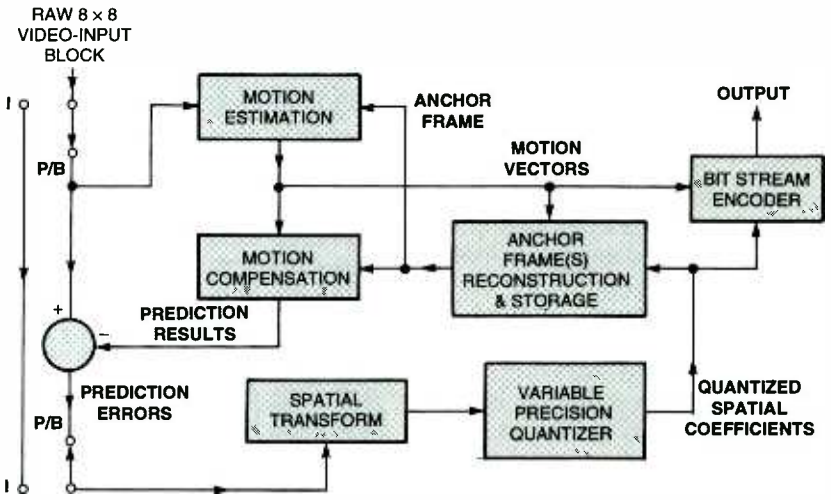


Fig. 4. The MPEG encoder-prediction loop forms the heart of MPEG video encoding.

Bi-directional prediction requires that the frame transmission order be different from the display order. The I- or P-type frames may serve as *anchor frames* (or "referred-to" frames) for other P or B frames, but the anchor frames must be available at the decoder *before* the predicted frames that will refer to them.

I frames are uniquely important to the decoding process. Since P and B frames refer to information from other frames, a decoder cannot be properly initialized until it encounters an I frame. In addition, any errors that find their way into the anchor frames can propagate through a series of P or B frames until replaced by some intraframe-coded blocks. For those reasons, special consideration is given to the use of intraframe coding within the video stream. For example, any given macroblock must be intraframe coded at least once in any 132 consecutive frames. While not required, it is recommended that I frames should be sent at least once every 0.5 second to allow for acceptable channel-change times. It would be annoying if you had to wait two seconds to see the picture after changing channels on your TV.

To see how the rest of the encoding process works, take a close look at the encoder-prediction loop shown in Fig. 4. Bear in mind that the decision to use I-, P-, or B-type coding is made separately for each macroblock, outside of this loop. Remember that the opera-

tions found here are carried out one block at a time, with the six blocks of any given macroblock that's processed sequentially.

Begin by imagining that an I frame is processed. Since I frames use no prediction, the "switches" are thrown toward the "I" terminal, letting every block of the I frame bypass the prediction part of the loop. Every I frame passes directly to the spatial-transform function.

The heart of the spatial transform is a process called the *discrete-cosine transform* (DCT). The DCT accepts a block as a matrix of gray-scale or color values in two-dimensional space, and then represents it as a matrix of spatial frequencies spanning the same region. In this way, blocks that initially contain similar values or regular patterns may be fully expressed as blocks containing few spatial-frequency coefficients. This is how multiple expressions of spatially redundant content are filtered out of the video stream.

Transforming blocks into the spatial-frequency domain presents the encoder with yet another opportunity to trim bits. The variable-precision quantizer adjusts the number of spatial-frequency bits according to what we can see. Terms occupying critical regions of the spectrum are given more bits, while precision is reduced for frequencies to which our vision is less sensitive or where the presence of one frequency will mask another.

The processed intraframe-coded block bits pass from the quantizer to the bit-stream encoder, where they will be bundled and sent on to the transport subsystem. Note that they are also passed to the anchor-frame(s) section.

Anchor-frame reconstruction and storage will play a vital role in the coding of subsequent P and B frames. Reconstruction means undoing what was done by the quantizer and spatial-transform functions. This is necessary because the decoded information will *not* be identical to the original. The encoder must work from the same reference information that the decoder will use to recover the predicted frames. Remember that predicted frames are encoded and decoded by *reference* to anchor frames.

"P-rocessing" P Frames. Now that every block of our initial I frame has been processed and there is a complete copy of the same decoded frame in the anchor-frame-storage buffer, we are ready to consider the encoding of a P frame.

P frame blocks go first to the motion-detection function, where they are compared to the corresponding blocks from the anchor frame. The motion detection's job is to answer the question, "What can I do to this anchor block to make it most resemble the block that I want to predict?" The answer can only do two things to the block: moving it x units horizontally and/or y units vertically. Each unit is half the width and height of a pixel. Due to the presence of temporal redundancy (similar information from frame to frame), this process of copying blocks from one frame into another and shifting them slightly does a fair job of conveying a new frame without actually sending the entire frame.

Unfortunately, fair is not always good enough. To the rescue come the motion-compensation and difference functions. Motion compensation takes the referenced block from the anchor frame and shifts it according to the motion vectors, reproducing the predicted block. The difference function compares the predicted block to the actual block and outputs any prediction errors that it finds. The prediction errors are sent to the spatial transform and quantizer functions, and ultimately to the bit-stream encoder. Therefore, the end of the process conveys our P-coded block as a reference to the corresponding block from a previously transmitted frame, modified by motion vectors and transformed/quantized-prediction errors. In practice, this results in a substantial reduction of information, especially if the prediction errors are few.

Simultaneously, the same information is gathered by the anchor-frame section, where our encoded P frame is decoded to serve as a reference for future frames—and so the process continues.

Finally, the bit-stream encoder has a few "bit-squishing" tricks up its sleeve. The result of the spatial-transform and quantization functions is most often a matrix containing zeros interspersed with a few isolated frequency coefficients. The encoder has a choice of two ordering schemes for scanning the matrix sectors. It picks the one that results in the greatest clustering of coefficients separated by the longest runs of zeros. It then applies run-length and Huffman coding to the scanned data.

Run-length coding takes a string of zeros followed by some non-zero value and produces a run-amplitude pair (two numbers (n, v) such that *n* is the number of zeros, and *v* is the value that ended the run). In Huffman coding, data values that are statistically likely to occur are represented by variable length code words—the greater the probability of occurrence, the shorter the assigned code word. In this way, the bulk of the information is conveyed using the shortest possible codes.

At the end of the video-source-encoding process, picture data are punctuated by organizational data, then forwarded to the transport subsystem.

Audio Compression. Like its masterful video processing, DTV's manipu-

TABLE 1
DTV Audio-Service Options

Service Name	Type	Number of Channels
Complete Main (CM)	Main	1 to 5.1
Music and Effects (ME)	Main	1 to 5.1
Visually Impaired (VI)	Associated	1 to 5.1
Hearing Impaired (HI)	Associated	1 to 5.1
Dialog (D)	Associated	1 or 2
Commentary (C)	Associated	1 to 5.1
Emergency (E)	Associated	1
Voice Over (VO)	Associated	1

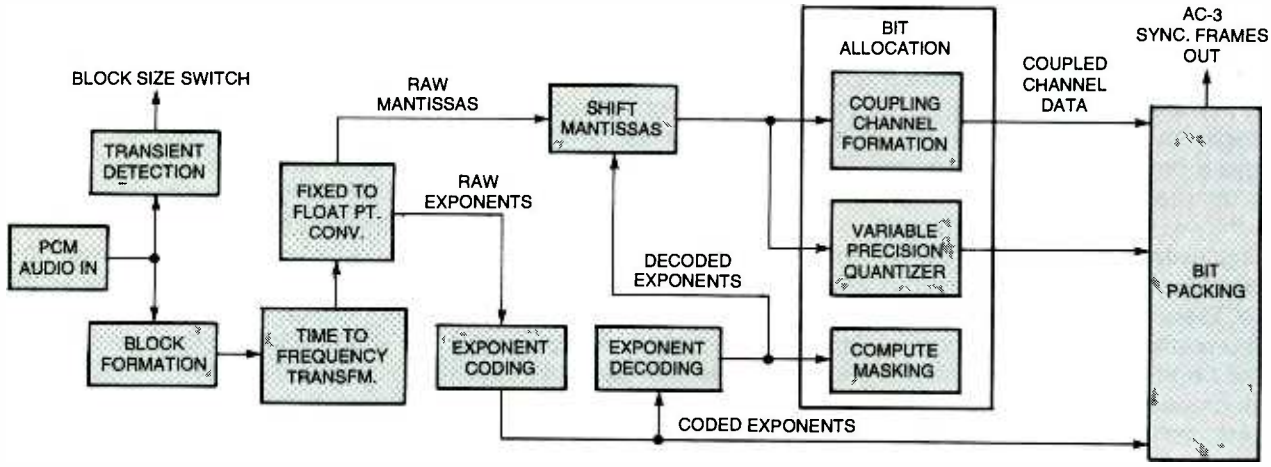


Fig. 5. AC-3 encoding involves a complex series of transformations for each processed channel. Bit allocation and bit packing is carried out globally, at the confluence of individual channels.

lation of audio is no mere sleight-of-hand. The new system will harness Dolby Labs' powerful AC-3 compression technology to deliver up to six main audio channels per program and/or a host of associated services.

AC-3 has been popularized for its promise to bring "5.1"-channel sound to home-theater audiences, as it has to moviegoers since 1992. The 5.1 channel designation refers to the availability of separate left, right, center, left surround, right surround, and subwoofer channels. Not as well known is the fact that AC-3 is an extremely versatile multichannel-coding scheme. As implemented for DTV, it is capable of compressing anywhere from one to six discrete channels into a single audio-bit stream.

Table 1 details the DTV audio-service options. Of those options, *Complete Main* (CM) is most commonly provided, carrying all of the normal program sound in multiple channels. *Music and Effects* (ME) is essentially the same, but without dialog. This is primarily intended to facilitate multi-language programming and may be accompanied by one or more separate *Dialog* (D) services. The *VI* option includes descriptive information for visually impaired viewers. *HI* adds dialog enhancement, offering increased intelligibility for the hearing impaired. *Commentary* (C) service is intended to convey supplemental audio—potentially enhancing but not essential to the program. The VI, HI, and C service types may each be provided as a single-channel augmentation to CM or as a full alternative multi-channel mix. When the *emergency* (E) service is present, all other audio is muted to allow for priority insertion of essential messages. *Voice Over* (VO) is similar, but rather than muting other program elements, VO attenuates them by as much as 24 dB, then muscles its way into the center channel of the audio mix.

As with video compression, the real goal of audio compression is to reduce the amount of program data without compromising quality. At their original sampling frequency of 48 kHz, the main service's 5.1 channels would typically require 5.184 Mbps to convey everything. Instead, they will be compressed into no more than 384 kbps (a ratio of 13.5 to 1).

Taking A "Byte" Out Of Sound. To learn how AC-3 compression works, follow the block diagram of the encoding process as shown in Fig. 5 during the following discussion.

Audio enters the process as pulse-code-modulation (PCM) samples. PCM involves sampling a signal at fixed intervals in time and recording the instantaneous magnitude of the signal at each sampling point. The recorded samples are conveyed as digital words or codes. If you've ever worked with sound files on a computer, you've worked with PCM audio. For DTV, the samples are commonly 16- to 18-bits long, but may be as long as 24 bits.

Within the encoder, blocks of PCM samples are converted into blocks of frequency coefficients that represent the spectral content of the signal—like the display of a very high-resolution spectrum analyzer. When audio data are transformed from time domain (PCM) to frequency domain and back again, special care must be taken to avoid certain audible distortions, especially *blocking artifacts* and *time smearing*.

Blocking artifacts occur at the borders between transformed blocks of samples. To eliminate them, the AC-3 process does away with borders entirely. Figure 6 illustrates the overlap and window functions that achieve this effect. Initially, PCM samples are grouped into blocks of 256, which are then duplicated and regrouped into 50%-overlapping pairs. The pairs are multiplied by windowing coefficients, such that the samples at either end of a pair are multiplied by nearly zero, while those in the middle are multiplied by one. Between those extremes, values of coefficients taper logarithmically. The windowed blocks are then transformed. Since the windowing process effectively cross fades from block to block, blocking artifacts are eliminated.

For every 512 windowed-PCM samples that it receives, the transform function outputs 256 coefficients, indicating the signal's relative power level at each of 256 narrow frequency bands during the 10.66-millisecond time interval spanned by the block of input samples. That's how the transform counter-

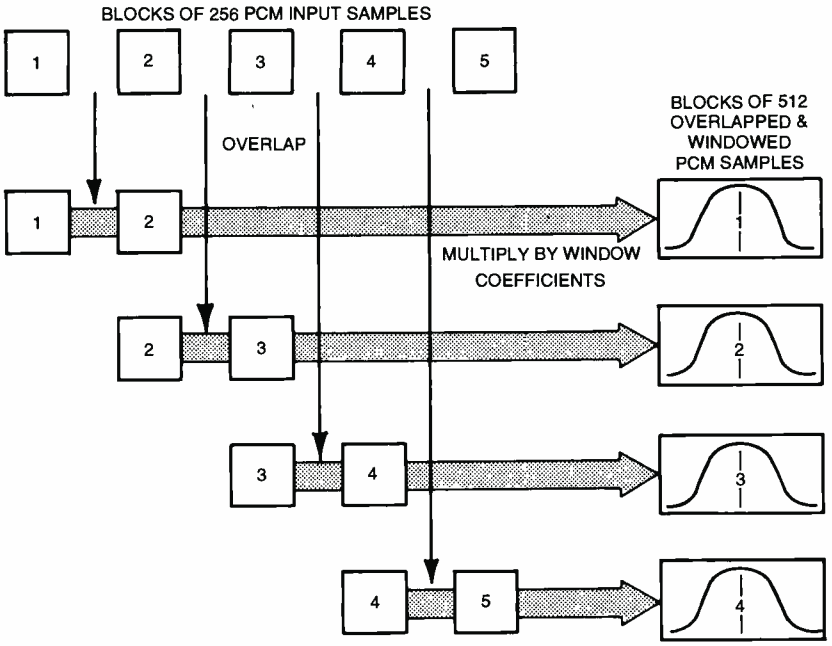


Fig. 6. AC-3 uses a clever technique to prepare blocks of PCM samples for transformation to the frequency domain. This approach effectively cross fades from one group of samples to the next, eliminating blocking artifacts.

acts the blocking function's doubling of samples. The number of frequency bands is a measure of the transform's spectral resolution, while the shortness of the time interval indicates its temporal resolution.

Later in the bit allocation routine, the encoder will save bits by neither attempting to represent sounds nor by hesitating to create noise that humans can't hear, thanks to the principle of *masking*. Masked sound is perceptually obscured by the audible part of the signal. However, in attempting to utilize masking at a temporal resolution of 10.66 mS, strong transients may result in audible distortion.

The trouble with transients is that their duration may be significantly shorter than the evaluation period, so that noise becomes audible before or after the transient that the encoder assumes will mask it—that's *time smearing*.

If the temporal resolution were doubled, so that each block of transform coefficients represented 5.33 mS of audio, then any such noise would fall within the temporal masking period of the transient and would not be perceivable. Unfortunately, that would double the amount of audio data, wrecking the compression economy.

Transient detection and block-size switching (Fig. 5) are the solution to that quandary. In the presence of a transient, AC-3 splits the blocks into two shortened blocks, transforming each separately. The transform for a short block produces only 128 frequency coefficients, so ultimately within the same number of transform coefficients, the spectral resolution is halved in exchange for doubling the temporal resolution.

To summarize the process thus far, after the time-to-frequency transform, we find blocks (or twin short blocks) of 256 frequency coefficients. Those blocks represent the spectral content of an equal number of PCM samples, spanning a time interval of 10.66 milliseconds. If we afford equal precision to both the time and frequency domains (16-bit coefficients for 16-bit samples), there is no data reduction so far.

Sound As Numbers. Raw frequency coefficients are initially represented as fixed-point binary numbers (a set number of digits follow the decimal point). These are converted to floating-point binary pairs. In a floating-point number, one number indicates the quantity of zeros to the right of the decimal point (the *negative exponent*), and another number is comprised of the remaining digits (the *mantissa*). As an example, the coefficient 0.00000 00010110110 would be represented with an exponent of 1000 (binary 8 for 8 zeros) and a mantissa of 10110110. Hereafter, the exponents and mantissas will be encoded separately.

Although exponents and mantissas follow separate paths, they will remain organized in blocks. By the end of the encoding process, groups of six blocks will combine to form larger organizational units called *synchronization frames*. Within a sync frame, a great deal of information will be shared. As you consider the operations leading up to the sync-frame formation, remember that each coefficient belongs to a unique frequency bin (or slot) within a specific transformed block. Blocks of coefficients retain this original association despite the manipulation to which they will be subjected.

Transform coefficients possess certain exploitable qualities that are not obvious at first. Chief among these is the fact that within a block, the magnitude of adjacent exponents rarely differs by more than 2. As a result, exponents can be coded differentially using one of only five possible increments (-2, -1, 0, 1, or 2). The first exponent in a block (the zero Hz or DC term) is represented as an absolute, and all subsequent exponents are coded as the *delta* (difference) between the current and previous terms. In this way, groups of three exponents can be conveyed in no more than seven bits.

The encoder will attempt to economize even more by applying each delta to as many exponents as possible. To this end, the exponent coding operation may select one of several exponent-coding strategies, depending on the extent to which the spectral content of the program varies across a sync frame. If the spectrum is relatively consistent, the first block will have one delta assigned to each exponent. The remaining five blocks will then reuse the same exponent set. Alternatively, if the spectrum is less stable, deltas may be shared across groups of two or four adjacent

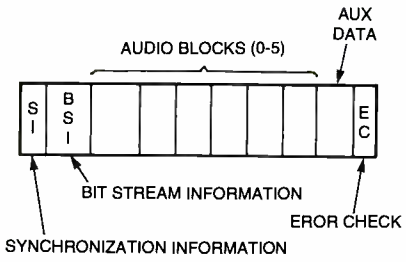


Fig. 7. The sync frame is AC-3's primary organizational unit.

exponents within a single block. In any case, exponent sharing is only permitted within the six blocks of a sync frame. Note that no data is shared across frames. Using these techniques, AC-3 typically achieves an impressive 2.5-coded exponents per bit.

In the process of exponent coding, the value of an exponent may be reduced to allow for the use of a more efficient strategy. In that case, the mantissa is required to take on leading zeros so that the whole coefficient is not changed. As shown in Fig. 5, decoded exponents are used to shift the mantissas, compensating for altered exponents.

Fitting The Available Space. Sync frames carry a limited number of bits that are divided amongst the various channels. Coded exponents are packed first. The bit-allocation routine apportions the remaining bits so that psychoacoustically-critical information is conveyed with the greatest precision.

The encoder's bit-allocation routine employs a mathematical model of the human auditory system. The same model is used in the receiver when the bit stream is decoded. In both instances, decoded exponents serve as a rough spectral representation of the audio program. This is used to compute a "masking threshold" across the signal's spectrum. Since any noise below this threshold will be obscured by the audible signal, the encoder can quantize mantissas with just enough precision to keep the quantization noise inaudible. When the mantissas are unpacked at the receiver, the decoder uses the same masking calculation to determine which quantizer was used.

If the number of bits encoding all of a program's channels still exceed the available bit pool, the encoder may resort to *channel coupling*. Above a frequency of about 2 kHz, we perceive directionality based not on the actual waveforms that we hear, but on the ear-to-ear difference in the fine spectral envelope. AC-3 takes advantage of this effect by combining the high-frequency content of selected channels while preserving the channels' original envelopes. The encoder determines the frequency at which coupling should begin and which channels will participate. It then forms a separate coupling channel.

When coupling is active, coupling coordinates for each original channel carried in block zero of every frame. They contain the envelope information, indicating the extent to which the coupling channel data should be applied to any other channel. If the envelopes are relatively consistent across a frame, then the same coordinates will be applied to all six blocks, but they may be updated as often as every block.

In the end, the sync frame (see Fig. 7) packages the encoded audio together with the synchronization and bit-stream information needed to properly parse and unpack the data, as well as auxiliary data and error-check fields. Each of a sync frame's six audio blocks carries coded exponents and/or mantissas for all of the program's channels, plus the coupling channel (when active). Every block spans 10.66 mS, but since the time intervals were overlapped during block formation, a complete frame encompasses only 32 mS of audio.

Dynamic Range. Finally, another aspect of DTV's unparalleled audio versatility may prove to be challenging for broadcasters. With old-fashioned NTSC, audio has occupied a very narrow dynamic range. In contrast, AC-3 will afford DTV programs a dynamic range in excess of 100 dB. Since this is a far greater range than most receiving systems can reproduce, it would seem to be important to know what the nominal program level should be.

In fact, no standard program level has been defined. Producers are free to set a program's average level anywhere within the overall dynamic window, allowing plenty of headroom for powerful sound effects and ample clearance from nuance to noise floor.

Instead of conforming to a standard level, AC-3 sync frames include two special indicators in the BSI field: one for the level at which dialog is encoded, the other for the overall dynamic range. The receiver uses the dialog indicator to scale the program levels so that dialog is reproduced at a constant level from program to program and channel to channel. The dynamic-range indicator will allow a program's range to be transposed to fit that of the receiving system (or to fit the viewer's preferences). If, for any given program, these values are not correct, the program's audio level will not be scaled correctly. Therefore, broadcasters are required to verify their accuracy. Imagine if every discount merchant in America learned that he could advertise at the level of a full-scale explosion! If abuses of DTV's extreme range prove to be widespread, new regulations will surely follow.

That's all the time and space we have for this month. Next month, we'll put the video and audio together and send it out over the airwaves. Be sure to tune in again next month; same digital time, same digital channel! P

SIGNAL GENERATOR

(continued from page 42)

Once the NTSC/PAL Signal Generator is tested, adjusted, and working, place the completed unit in an appropriately-sized metal case.

Using the Signal Generator. The NTSC/PAL Signal Generator outputs color bars when JP5 is open, but by shorting JP5, you can generate color black video (black video with a color-burst signal). By shorting JP4 as well, any full-screen color can be created by adjusting R8, R9, and R10. Be careful to keep the overall video level to the 1-volt peak-to-peak standard.

I hope that you have learned something about color video and will find this project useful. Fortunately, everything is neatly broken into simple blocks so that you can be assured of success. Have fun. P

Looking Ahead to DTV: Part 2

How digital television combines compressed video and audio signals and delivers them to a digital-ready television near you. Second of two parts.

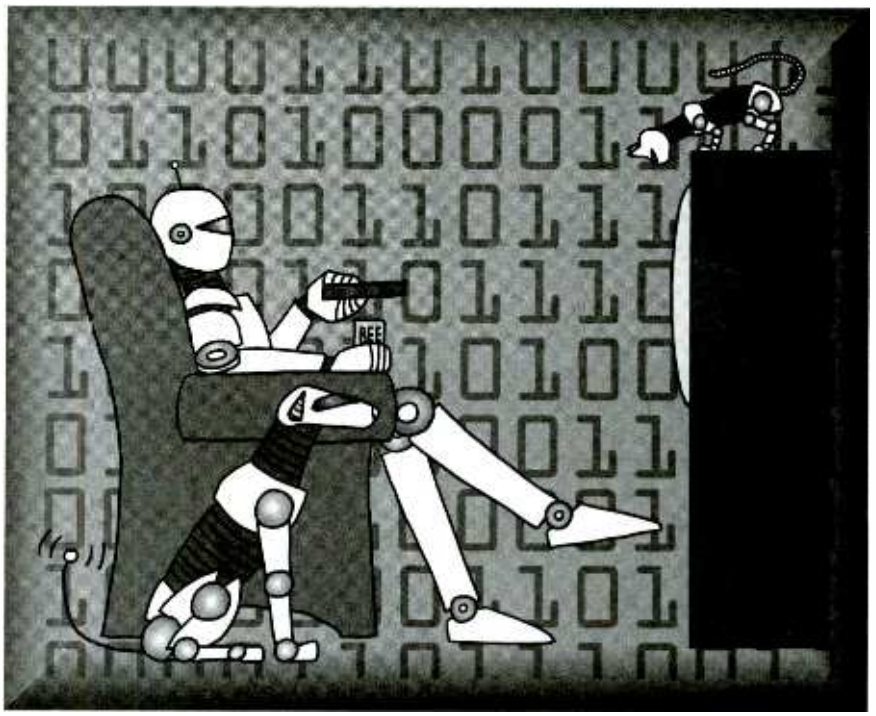
GEOPHREY MCCOMIS

Last month, we looked at the mechanics of how digital television works, including how video (in a single high-definition format or several standard-definition channels) is compressed, squeezed, and otherwise mashed down to fit within a 6-MHz bandwidth. We also saw how the same techniques are able to supply home-theater-quality surround sound in a 5.1 format (four surround channels plus a center channel and subwoofer) with Dolby noise reduction to boot.

Now that we have the two major components of television (video and audio) digitized and compressed, let's see how we put them together. Be sure to have last month's issue handy; there will be occasional references to some of the charts and figures that were published in the first part of the article. To avoid confusion over which figure number belongs to which section, we're going to continue on with the numbering scheme as if this is one large book-length article.

Now that we have all of the disclaimers out of the way, let's plunge into this month's subjects. We'll start with...

The Service-Multiplex and Transport Subsystem. When you look back to Fig 1, the transport subsystem's primary function is obvious: it combines the elements from multiple programs with ancillary and control data to form a single transport stream. As with video, DTV transport streams conform to standards defined under MPEG-2 and constrained for DTV.



A very general summary of the transport subsystem might stop at that, but there are far more interesting issues that lurk just beneath the obvious.

Consider the synchronization of a program's audio and video. In analog-television systems, audio and video information is sent simultaneously and in real time—both are sent, received, and presented concurrently. A DTV program's audio and video are interspersed with other data, so even though the overall bit rate is constant, any one elementary stream appears in bursts in a fraction of the time required to decode and display it. Additionally, MPEG compression often results in frames being reordered for transmission, so there is no inherent synchronism between audio and video information.

For DTV, the secret to synchronism lies between the source and transport-encoding processes (Fig. 8). Elementary audio and video data are first grouped into *Packetized Elementary Stream* (PES) packets. PES packets are variable-length-data structures that tag audio and video frames with packet-start codes and various other header elements.

Among the PES header elements are two "time stamps:" the *Presentation Time Stamp* (PTS) and *Decode Time*

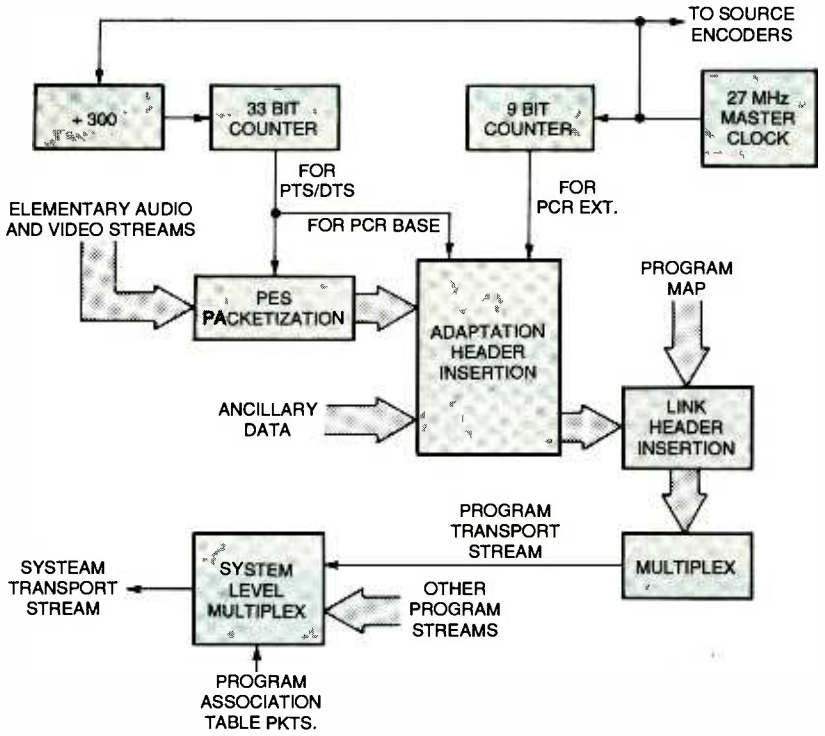


Fig. 8. Construction of DTV's transport streams is a fairly complex affair, involving multiple programs. All of the timing for the transport subsystem is based on a 27-MHz master clock. A 9-bit counter provides a high resolution snapshot of the time elapsed from packet to packet, while a phase-locked 90-kHz clock rate drives a 33-bit counter to place each packet within a large window of time.

Stamp (DTS). The Presentation Time Stamp indicates exactly when a given packet's contents should be presented. The Decode Time Stamp is included whenever frame reordering occurs; it notifies the decoder that the DTS-stamped packet will be required prior to its presentation time (usually in order to decode intervening B frames).

All of the timing for source encoding and decoding is based on a 27-MHz master clock. Both PTS and DTS values are snapshots of a 33-bit counter driven by a 90-kHz divided-down version of the master clock (Fig. 8). Thus, they provide accurate placement information within the overall context of a large window of time.

One step up from the PES level is the adaptation layer as shown in Figs. 8 and 9. Adaptation headers are variable in length—mandatory for audio and video and optional for other data. They exist primarily to facilitate the synchronization of program elements.

Chief among the adaptation header elements is the *Program-Clock Reference* (PCR). The PCR is comprised of two parts: the PCR base (taken from the same 33-bit counter used to generate time stamps) and the PCR extension (a sampling of a nine-bit counter driven directly by the master clock).

The master clock is also needed at the receiver to decode the incoming signal. The decoder generates a local representation of the master clock at the receiver, aligning it with the PCR embedded in the incoming bit stream. In this way, it establishes its own program clock, which becomes the reference for the presentation and decode times indicated by the incoming time stamps. That's how audio and video are synchronized.

The adaptation layer also provides indicators for random access and local program-insertion points.

Recall that once an MPEG decoder has acquired a given program stream, it must be initialized with an I frame. Thus, within a particular bit stream, the start of an I frame is a valid random entry point with respect to the video decoder. Such random entry points are flagged by the state of a special field in the adaptation header. This allows for faster redisplay when switching channels or programs.

And Now For a Word From Our Sponsor. Where would television be without commercials? Commercials are prominent examples of local programming. However, local program insertion might adversely affect the PCR and its representation at the receiver. Imagine, for example, that you have just enjoyed six minutes and twenty-seven seconds of your favorite TV drama; the current PCR and time stamp values will indicate "6:27:xxxxx." Then the local network affiliate cuts to five minutes of commercials. What happens to the time values? Must the PCR be yanked back to 0:00 at the start of each commercial, only to jump back to 6:28 when the program resumes? How might this affect decoder synchronization across brands of receivers?

The adaptation header's splice-countdown field embodies one attempt to provide respectable and consistent answers to those questions. During a normal program, it indicates the number of packets remaining before the occurrence of a splice point. During inserted programming, it reflects the anticipated time before resumption of the featured program.

The discontinuity indicator is another important adaptation-header field. It gives the decoder advance warning that the PCR is about to change (before the start of a new program, for instance).

While it is still up to the receiver to track a potentially shifting time base, the adaptation layer provides the additional support required to maintain reliable performance across the full spectrum of program insertion/change scenarios.

The link level is the final stage in the assembly of a transport packet. Link headers are fixed in length at

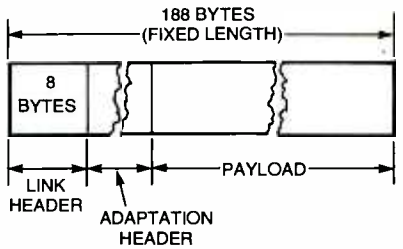


Fig. 9. Transport packets are the common currency of the transport subsystem. Most of the subsystem's per-program operations relate to the assembly of transport packets. System-level multiplexing amounts to shuffling them together, like cards.

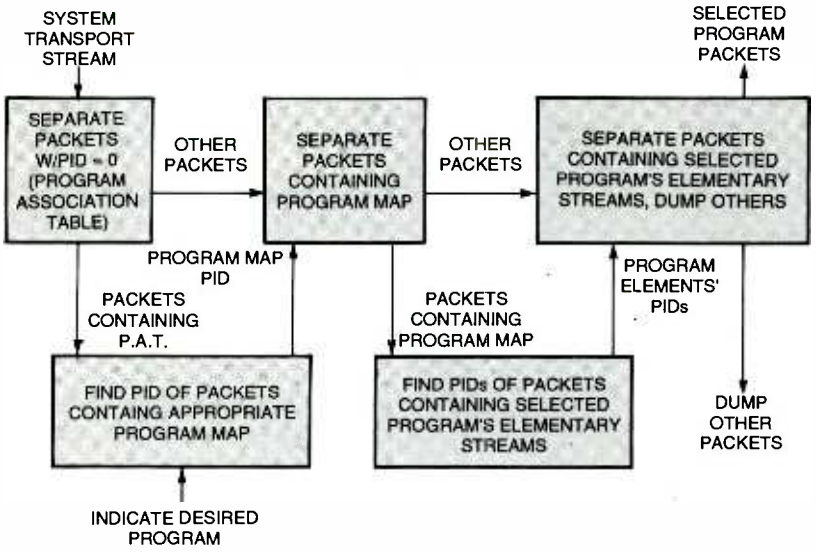


Fig. 10. De-multiplexing the system-transport stream is like a treasure hunt: several pointers must be followed to get to any single program's transport packets.

four bytes. As indicated in Fig. 9, the primary functions facilitated at the link level are packet synchronization and identification, error detection, and conditional-access notification.

The first eight bits of every packet are the link header's *sync byte*. Each sync byte carries the same value for all MPEG-2 bit streams. This allows ready detection and constant verification of the location of transport packet boundaries.

The *Packet-Identification* (PID) field occupies 13 bits in the center of the link header. Within a given system multiplex, packets that belong to any particular data stream carry a unique PID value. At the receiver, packets are ultimately sorted out according to their PIDs.

Two separate link-header fields provide error-handling utilities: the four-bit *continuity counter* and the one-bit *transport-packet-error indicator*. The latter is simply a flag that may be set by the modulator or demodulator to indicate that a given packet is known to be in error and should not be used.

The continuity counter confirms the delivery of successive packets of each payload-bearing PID. As the program stream is assembled within each set of packets for a particular PID, it cycles from zero through 15. So, for example, if one packet of PIDs carries a continuity-counter value of 7 and the next packet received for the same PID carries a value of 9, the decoder will recognize that data has been lost and should take steps to control the damage.

The DTV standard does not specify a particular method of encrypting data for conditional access (as with pay-per-view or premium services), but it does provide the means for program providers to do so. Link headers must be sent without encryption, but the balance of a transport packet could easily carry encrypted data. For this reason, the link header's transport-scrambling control identifies packets bearing scrambled payloads.

Transport packets are the common currency of the transport subsystem. They are fixed in length at 188 bytes and are easily multiplexed. Multiplexing at the program level is simply a matter of alternating a program's audio-, video-, and ancillary-transport packets. Together, they constitute a *program stream*. Multiplexing at the system level is more complex. Here, transport packets from multiple program streams must be interwoven. Still, transport packets will remain intact and will merely be placed end-to-end with packets of other programs' various elements. In that patchwork of program packets, each packet's PID identifies the program and element to which it belongs.

PID number zero is reserved for a special class of transport packets—those containing the program-association table for the system-level transport stream. The program association table is like a guidebook to the system stream. It identifies the stream's programs and the PID numbers that contain each program's program-map table. Each program-map table in turn identifies the PID numbers belonging to its associated program elements.

Figure 10 illustrates the process of unraveling the system-level transport stream to get at a particular program. First, packets whose PID is zero are gathered and interpreted to find the PID of the program map for the desired program. Those packets are then gathered and interpreted to find the PID numbers of

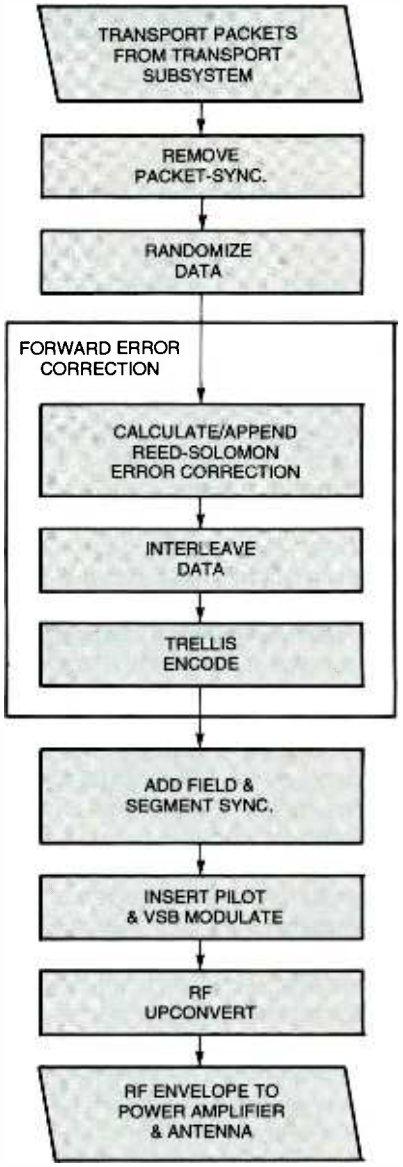


Fig. 11. Error correction is the central facet of DTV's terrestrial-transmission system.

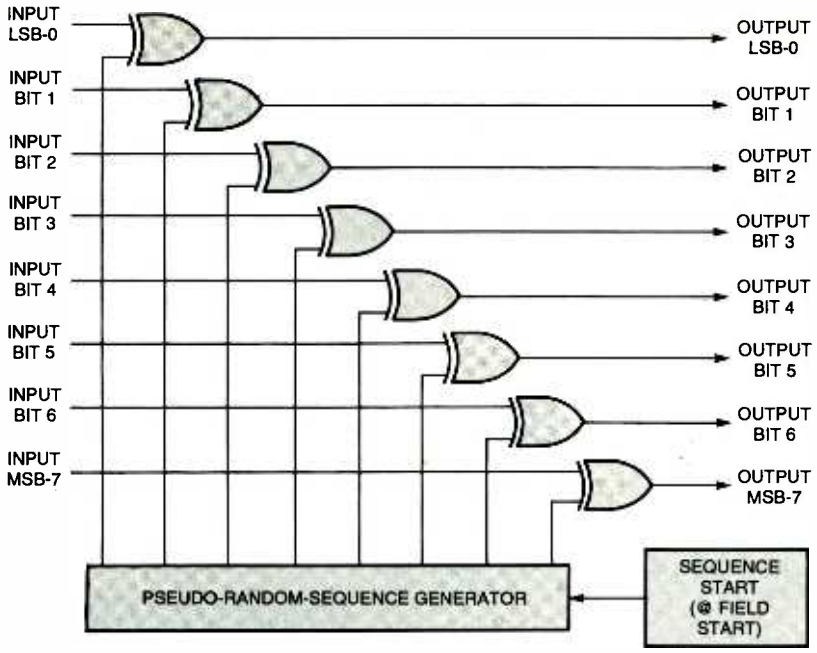


Fig. 12. The randomizer's XOR gates invert the information bits whenever a bit from the sequence generator is a one. In the receiver, the same sequence is XOR-ed with the randomized data, reversing the process.

the selected program's elementary streams, which in the end are separated for decoding, while the remainder of the system transport packets are ignored.

Program selection may be enhanced by the presence of a master-program guide. This is one example of an ancillary data type. While ancillary data may ultimately include anything from stock quotes to software, a unique PID (number 8189) has been reserved for it due to the program guide's essential role.

The RF/Transmission Subsystem. DTV's transport stream is already a complex entrée of digital information; serving it up over the open air is yet another challenge. This is the province of the RF/transmission subsystem. It reprocesses the contents of transport packets to blast them across the miles of noisy air, while maximizing the likelihood that they will emerge intact at their multiple destinations.

The ATSC standard actually describes two transmission modes: *8VSB* for terrestrial broadcast (open air) and *16VSB* for use in cable and other delivery systems in which high signal-to-noise ratios are easily maintained. VSB stands for *vestigial-sideband* modulation, in which either 8 or 16 discrete-amplitude levels are used to convey three or four bits at a time. While 8VSB uses robust-coding techniques to move 19.28 Mbps across a potentially noisy channel, 16VSB trades resilience for data-carrying capacity, delivering 38.57 Mbps. The balance of this article will focus on the 8VSB mode.

Figure 11 summarizes the progression of information through the terrestrial transmission subsystem.

First, transport packets are gathered and stripped of their sync bytes. These are unnecessary in transmission since the RF subsystem creates data frames of its own, and the positions of data from successive transport packets are clearly defined. Packet sync will be restored in the receiver, at the output of the RF stage.

Next, incoming data are randomized. Randomization gives each bit an equal chance of being a zero or a one, which optimizes the bit stream for the rest of the RF subsystem. Figure 12 shows how that is achieved. The start of every data frame triggers the start of an 8-bit pseudo-random sequence, which is XORed (EXCLUSIVE-ORed) with the incoming data bytes. The sequence appears to be random, but it actually emerges from a mathematical function that always yields the same result. Since the XORs invert the data bits any time a bit from the sequence generator is a one, the output data also appear to be random. In the receiver, the same sequence is XORed with the recovered randomized data, magically revealing the original.

Fixing Mistakes. *Forward-Error Correction* (FEC) refers to the addition of special error-correction data on the part of the sender in a one-way digital-communication system. A large part of the transmission subsystem is devoted to FEC. Without it, ATSC reception would falter due to interfering signals, multipath distortion, and adverse atmospheric conditions.

DTV employs three stages of FEC: Reed-Solomon coding, data interleaving, and trellis coding.

It's difficult to imagine how effective DTV's three-part FEC ensemble is. During evaluation while the system was in development, it was found that the threshold for the visibility of errors occurs at a signal-to-noise ratio (S/N) of 14.9 dB. From an analog perspective, this is astounding. It means that roughly one sixth of the signal voltage seen by the receiving system must be noise before any negative effects are observable.

Unfortunately, error correction can only forestall the inevitable. The slope of 8VSB-error probability as a function of S/N is so sharp that the system can't function as the S/N approaches 14 dB. This is the "brick-wall effect" often associated with DTV reception: you get either a perfect picture or no picture at all.

Reed-Solomon (RS) coding is no stranger to consumer electronics, since it's at the heart of the Compact Disc format. Though its mathematical underpinnings are inaccessible to most mortals, think of it as a sort of multi-dimensional parity. For every randomized packet (187 bytes), 20 RS parity bytes are calculated and tacked on. The parity bytes allow the reconstruction of up to ten corrupted bytes within a packet, even if some of the parity bytes are damaged. They are therefore very effective in the correction of random-bit errors.

To achieve partial immunity to longer-burst errors, the data are also interleaved. Interleaving refers to the systematic scrambling of bytes from adjacent packets. Figure 13

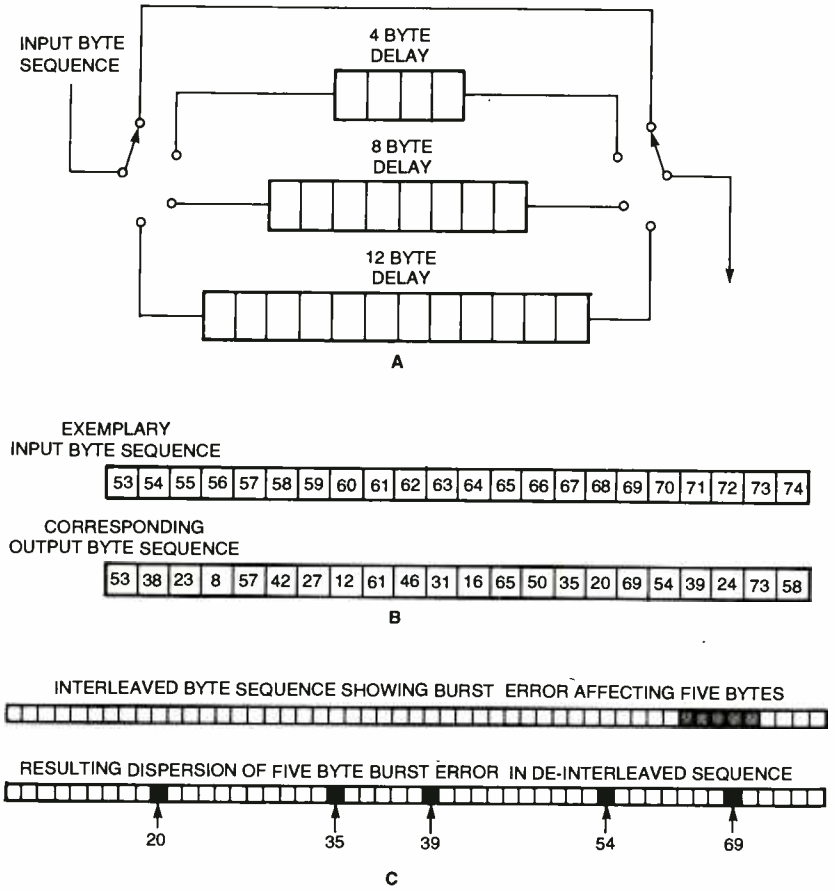


Fig. 13. From the receiver's viewpoint, interleaving causes burst errors to look like random bit errors.

provides an example of a simplified interleaver and shows how interleaving can distribute the effects of burst errors over a broad range of non-adjacent bytes. However, where the interleaver of Fig. 13 is comprised of only four stages with a maximum delay of 12 bytes, DTV's interleaver has 52 stages with a maximum delay of 204 bytes. The interleaving sequence is initialized at the start of every data field, resulting in the mixing of bytes over a range of 52 packets. When the data are "de-interleaved" in the receiver, burst errors are dispersed, affecting small portions of several packets rather than a large part of any one packet. This dispersion allows the use of RS parity to correct much larger errors than would otherwise be possible.

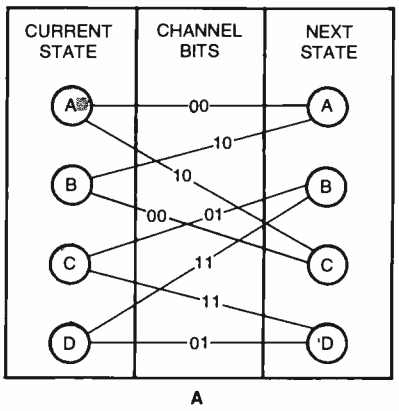
After interleaving, the data directly associated with any one packet are scrambled. When we look ahead to the structure of a VSB data frame, we see that interleaved data and RS parity bytes will be grouped into *segments*. One segment contains the data equivalent to 207 interleaved bytes, which is precisely the amount of data contained in one packet, plus its associated parity.

In the next and final stage of FEC, each interleaved byte (8 bits) will be converted to 12 *channel bits*.

Channel bits are the product of channel coding. The NATO alphabet—often heard in old war movies—provides a good example of channel coding. In a broken or noisy radio transmission, the sequence "Alpha-Bravo-Charlie" has a better chance of being correctly understood than simply "a-b-c," which could come out sounding like "a-e-e." Digital channel coding usually involves adapting a unit of data to a longer code word to enhance its intelligibility in a potentially noisy-transmission system.

Trellis Coding. Trellis codes belong to a family of channel codes that are called "convolutional." They are named after the appearance of the state diagrams that demonstrate their formation and decoding. Take a peek at Figs. 14 and 15—the diagrams look like trellises. They are considered convolutional, because at any point in time the output code word depends not only on the present input data, but also on the state of the encoder—a function of past input data.

Examine Fig. 14 more closely, and it's easy to see how trellis coding works. The encoder has four states, labeled "A" through "D." It



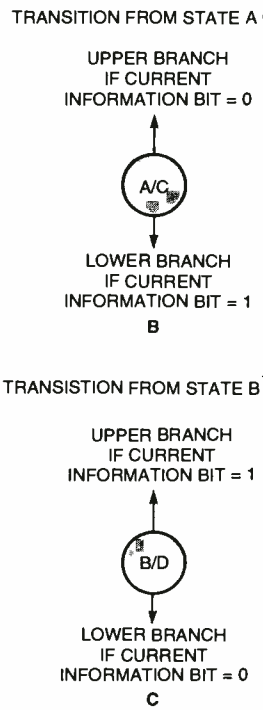


Fig. 14. At any moment, the trellis encoder exists at one of four states. As each information bit is received, the encoder outputs two channel bits and moves to the next state. The transition table and trellis diagram demonstrate the eight possibilities associated with the encoding of any information bit.

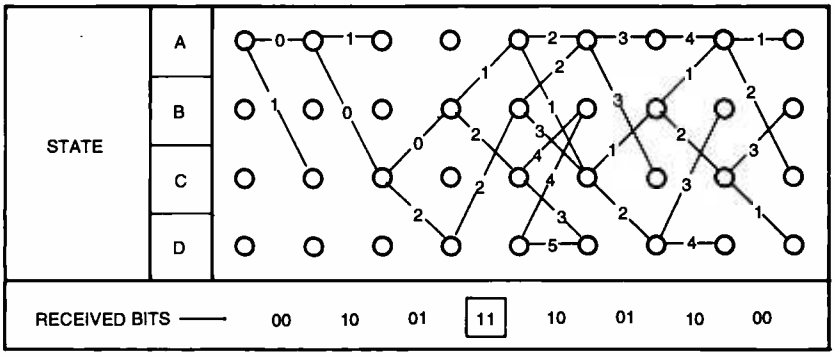


Fig. 15. Viterbi decoding is probabilistic. Each pair of received channel bits leads to a new vector in the trellis. The decoder compares all possible vectors with those suggested by the received data, and, over time, selects the path associated with the least number of errors (the most probable).

starts out at "A." As each bit enters the encoder, two channel bits emerge, and the encoder advances up or down to the next state—the example of Fig. 14 is a *½-rate* code.

The most common method of unraveling trellis-encoded data has been named *Viterbi* decoding after its originator, Andrew Viterbi. For an example of how it works, consider Fig. 15. The initial state is assumed to be "A." The decoder receives the first pair of channel bits: 00. Check the transition table of Fig. 14, and you will see that 00 is a valid channel code from state "A." It is associated with a source bit of 0 and a transition back to state "A." That vector is traced in the decoder's memory with an indication to verify that no bit errors were received If the vector was the actual path intended in the original transmission.

The only other valid channel code from state "A" is 10. If 00 had been falsely detected in place of 10, then one bit would have been received in error. This possibility is also considered. The next two channel bits are 10, suggesting a valid source bit of 1 and a transition to state "C." Again, this vector and its single-error alternative are traced.

Skipping ahead to the fourth pair of received-channel bits, you will see something interesting. The current state is "B," and the received channel bits are 11. Although 11 is not a valid channel code from state "B," the decoder is not bothered by this. It simply traces the two possible vectors from state "B" and notes the number of requisite bit errors. The one-error vector would more likely be the intended path than the two-error vector, but the errors encountered at this point are carried through the next several nodes, and many possible paths are evaluated. As the decoding progresses, improbable paths are closed out and the path of least error emerges as the most likely. Viterbi decoding is probabilistic. It evaluates channel data and recommends the most probable source data.

There's an extra twist hidden in the trellis encoder. 8VSB uses a ½-

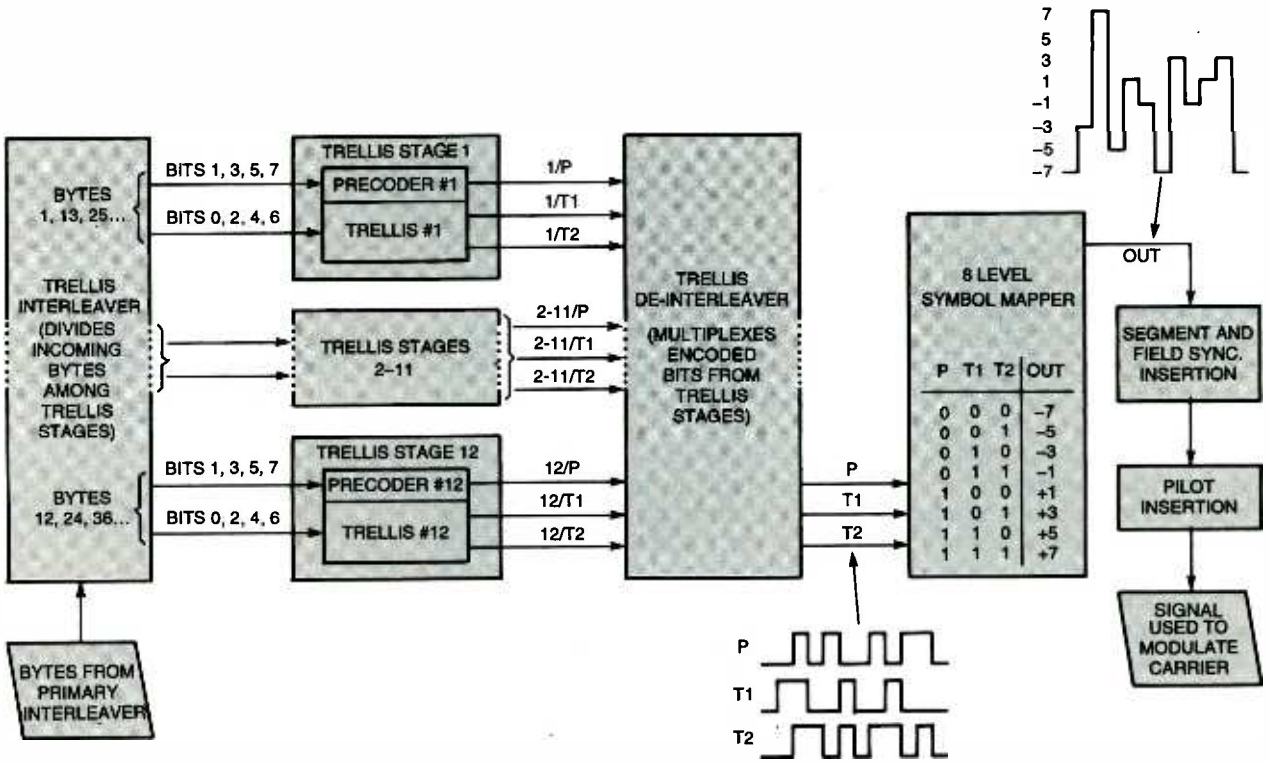


Fig. 16. DTV's trellis encoding approach is ingeniously wed to its 8VSB symbol mapping. Only half of the source bits are actually trellis encoded. The other bits determine the sign of the signal created by the symbol mapper (they are assigned the greatest bit weight), so they are least likely to be misinterpreted at the receiver.

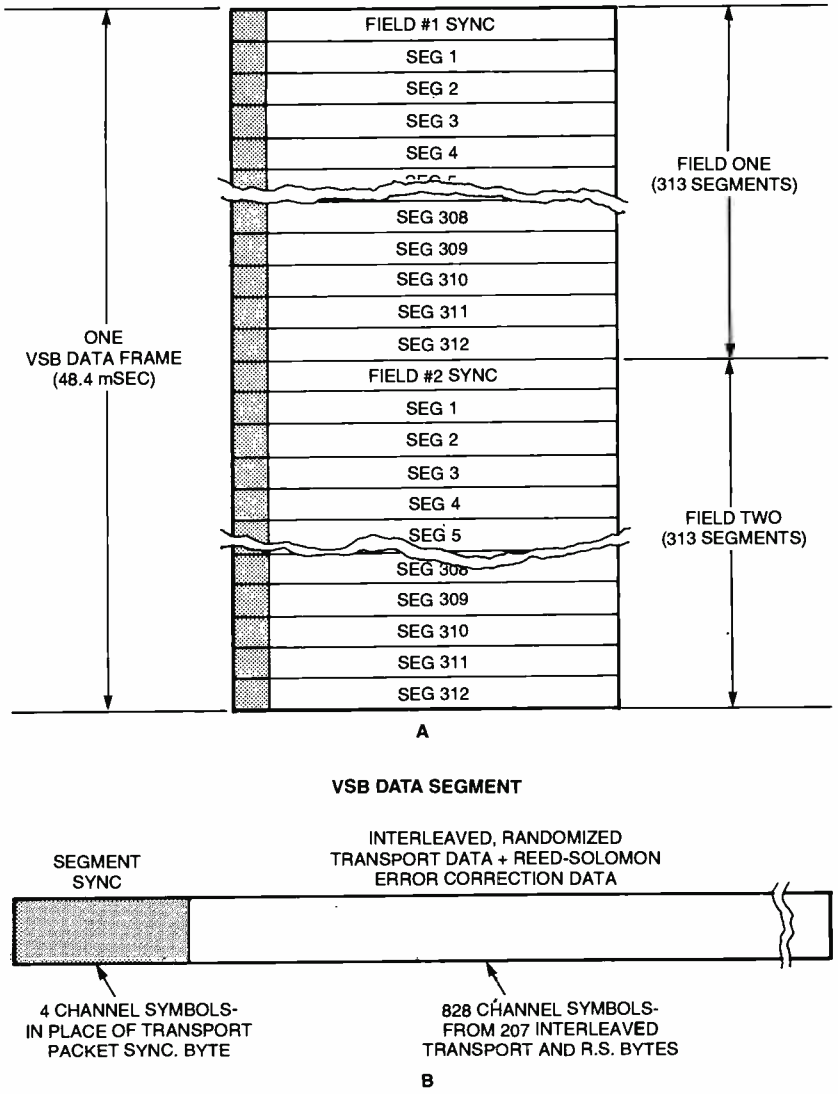


Fig. 17. ATSC transmissions have a framing structure reminiscent of analog video formats. Two fields, each comprised of 313 segments, make up a frame.

rate trellis code, yet its overall coding rate is ⅔. That's because only half of the source bits are trellis coded. As you can see in Fig. 16, at the threshold of the encoder, interleaved bytes pass through yet another interleaver. Unlike the primary interleaver, the second is merely a "traffic cop," directing the incoming bytes to one of 12 trellis stages. Each byte is split as it's fed to one of the 12 stages, and only its even bits (0,2,4,6) are trellis coded.

The odd bits pass through a *precoder*, which performs a simple operation. Starting with an output of zero, it generates a transition when its input is a one and holds its output state when its input is a zero. This treatment is easily undone at the receiver, and it produces a single bit for every bit processed.

The combination of a precoder and trellis coder forms a complete trellis stage. Each of the trellis stages generate three bits for every input pair, making up the overall ⅔ rate. The de-interleaver multiplexes the outputs of the 12 trellis stages into a single three-bit-wide data bus, completing the encoding process.

When you consider the symbol mapper, the elegance of the ⅔ approach becomes clear. The symbol mapper represents a trio of bits as one of eight discrete DC levels (a *symbol*), which will ultimately correspond to carrier-amplitude levels. From the chart on the symbol mapper in Fig. 16, note that the precoded bit determines the sign of the modulating DC level. It is unlikely that the receiver will confuse a positive for a negative level (or *vice versa*), except in the case of +1, and -1, at which point the two trellis-encoded bits are maximally differentiated (00 versus 11). Furthermore, instances of the same trellis-coded bits occurring on opposite sides of the polarity divide are always four levels apart (+1 and -7, +3 and -5, +5 and -3, +7 and -1). This highly efficient approach makes it possible to reap the benefits of trellis encoding, while only encoding half of the source data.

After symbol mapping is done, field- and segment-synchronization signals are added to the train of 8VSB channel symbols. They are binary for easy identification; and they toggle between high and low states, which correspond to 8VSB levels of +5 and -5. Fig. 17 details the positions of the sync signals in the overall framing structure.

Segment sync consists of the four-symbol pattern 1001. It is the only non-random repeating pattern in the transmission, occurring at regular 77.3 -microsecond intervals so it stands out clearly.

Field sync occupies a complete segment and occurs once in every 313 segments. Most of it consists of several repeating pseudo-random patterns, spanning 700 symbols. Of the remaining 128 symbols, 24 indicate the VSB mode of the transmission (currently either 8VSB or 16VSB). The balance is reserved, serving no defined purpose.

The combination of 8VSB channel symbols and sync patterns will be used to modulate the RF carrier, but not until a slight adjustment is made.

Pumping It Out. Any transmission system is most efficient if it fully utilizes all of its available bandwidth. DTV has been designed to use all of its 6-MHz channel bandwidth nearly all of the time. It is therefore said to be noise-like; when observing an ATSC channel over time, one finds an equal distribution of energy from upper to lower band edge—the defining characteristic of white noise.

DTV uses an amplitude-modulation format that takes advantage of a technique called *carrier suppression*. This eliminates AM's char

(Continued on page 67)

DTV: PART 2

(continued from page 43)

acteristic power spike at the carrier frequency, leaving all the available transmitter power for the information-rich sidebands. In vestigial-sideband formats, one of the sidebands is also suppressed. Since the sidebands are mirror images, one of them is superfluous. In DTV's case, only the upper sideband and a slight vestige of the lower sideband are actually transmitted.

Carrier suppression makes the receiver's job a little trickier, since it must essentially regenerate the missing carrier in order to recover the information from the sideband(s). This would be especially difficult in DTV's case, since its use of channel spectrum is basically flat, looking like noise.

To help the receiver identify the carrier, a small "pilot" is added to the transmission at the carrier frequency. In other words, a little bit of the carrier is preserved for the benefit of the receiver. Before the combined channel symbols and sync are sent to the modulator, a DC level that corresponds to exactly 1.25 8VSB level divisions positively biases them. All of the 8VSB and sync levels are raised by that amount. Adjusting the modulating signal in this way is called *pilot insertion*. Although the modulator performs carrier suppression, the added DC content of the modulating signal preserves the carrier as part of the modulated information. While sufficient to aid the receiver, the power level of the pilot signal is slight, remaining 11.3 dB below the average data-signal power.

With the pilot level in place, the 8VSB signal is modulated on an intermediate-frequency carrier, then upconverted to the transmitter's specified operating frequency.

Statistical analysis of DTV's 8VSB mode has shown that 99.9% of the time, the peak transmitter power is within 6.3 dB of the average power. From a practical standpoint, the average- to peak-power relationship allows an ATSC transmitter to achieve the same coverage as an NTSC transmitter at the same frequency with an average power level 12 dB lower than NTSC's peak sync power (where the bulk of NTSC's signal power is concentrated).

Additional Resources. So now you know a thing or two about DTV. Nevertheless, if you plan to remain a videophile or hope to become a guru, you'll probably want to know more.

The Internet is a great resource for information on DTV. You might start with a visit to the ATSC's Web site: *www.atsc.org*. There you can freely download the DTV standard and an excellent companion reference, *Guide to the Use of the ATSC Digital Television Standard*. Another great Web site is the National Association of Broadcasters (NAB): *www.nab.org*. They maintain links to a variety of broadcast-related information resources, including a page dedicated to DTV off their "Current Issues" page.

As the countdown to 2006 continues, it will be fascinating to watch the changes in television broadcasting unfold. May your vantage point be an enlightened one. P

GETTING INSIDE AN NCO

(continued from page 50)

You can use a 4-kB CMOS chip—the 27C32 (Digi-Key NM27 C32BQ200-ND) or the cheaper and faster 8k part—the 27C64 (Digi-Key NM27C64Q-150-ND). The latter part has 28 pins; its unused address pins should be tied to ground.

Since the EPROM generates glitches every time the address inputs change, it needs an output latch (IC11) to get good results. Jumper J1 allows quick swapping from ramp to sine outputs. Wire pin 1 of IC9 to ground if you don't use a sine converter. For best results, you should change the filter to the one shown in Fig. 4.

Modulating The Output. Figure 5 shows how different types of modulation can be applied to the basic NCO. Adding frequency-shift modulation is simple; just add a second frequency source (more DIP switches, for instance) and a bank of two-way multiplexer chips. Driving the multiplexer with a binary signal switches the output between the two frequencies you have set.

Quadratic-phase modulation is also easy. All it takes is two more EXCLUSIVE-OR gates between the adder and the existing EXCLUSIVE-OR bank. The EPROM can also be programmed to generate QPSK modulation from a signal applied to its address inputs.

If you want to experiment with continuous-frequency modulation, you need to digitize the analog signal and add it to the DIP-switch input. You can generate narrow-band FM without adders by using the bottom bits of the frequency-control number as the modulation input and the top bits as the carrier-setting number. Putting adders between the ramp output and the EXCLUSIVE-OR gates implements continuous-phase modulation.

With a little experimentation, you will soon learn the value of these unique devices and will be ready to use an NCO chip in your own designs. P

More Information About NCO Generators

"Digital Frequency Synthesis" *Circuit Cellar Ink*, October 1998 (*www.circuitcellar.com*). This article discusses how to generate accurate, modulated audio-frequency signals with a cheap microcontroller.

"Making Waves with NCOs," *Circuit Cellar Ink*, December 1997–January 1998. A signal-generator construction project using a Harris (now Intersil) NCO chip to generate both sinewave and squarewave output from 1 Hz to 10 MHz.

"Push Numerically-Controlled Oscillators Beyond Their Limits," *EDN*, September 12, 1997 (*www.ednmag.com*). An introduction to NCOs and some ideas for extending their frequency range. This article includes more detail on generating a squarewave without a sine conversion.