Texas Instruments

DSP Solutions for Telephony and Data/Facsimile Modems

Application Book

1997

Digital Signal Processing Solutions

Printed in U.S.A., January 1997 SPRA073

DSP Solutions for Telephony and Data/Facsimile Modems

Tim Massey and Ramesh Iyer
Technical Staff-Western Area DSP Applications
Digital Signal Processing Solutions — Semiconductor Group

SPRA073
January 1997

TEXAS INSTRUMENTS

IMPORTANT NOTICE

Texas Instruments (TI) reserves the right to make changes to its products or to discontinue any semiconductor product or service without notice, and advises its customers to obtain the latest version of relevant information to verify, before placing orders, that the information being relied on is current.

TI warrants performance of its semiconductor products and related software to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are utilized to the extent TI deems necessary to support this warranty. Specific testing of all parameters of each device is not necessarily performed, except those mandated by government requirements.

Certain applications using semiconductor products may involve potential risks of death, personal injury, or severe property or environmental damage ("Critical Applications").

TI SEMICONDUCTOR PRODUCTS ARE NOT DESIGNED, INTENDED, AUTHORIZED, OR WARRANTED TO BE SUITABLE FOR USE IN LIFE-SUPPORT APPLICATIONS, DEVICES OR SYSTEMS OR OTHER CRITICAL APPLICATIONS.

Inclusion of TI products in such applications is understood to be fully at the risk of the customer. Use of TI products in such applications requires the written approval of an appropriate TI officer. Questions concerning potential risk applications should be directed to TI through a local SC sales office.

In order to minimize risks associated with the customer's applications, adequate design and operating safeguards should be provided by the customer to minimize inherent or procedural hazards.

TI assumes no liability for applications assistance, customer product design, software performance, or infringement of patents or services described herein. Nor does TI warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right of TI covering or relating to any combination, machine, or process in which such semiconductor products or services might be or are used.

Contents

Part I DSP Solutions for Telephony

Part II DSP Solutions for Data/Facsimile Modems

List of Figures

Part I DSP Solutions for Telephony

Part II DSP Solutions for Data/Facsimile Modems

List of Tables

Part I
DSP Solutions for Telephony

DSP Solutions for Telephony

ABSTRACT

This application report describes DSP algorithms used in telephony applications. These algorithms include tone detection/generation, dual tone multi-frequency (DTMF) generation/detection, voice compression/decompression (ADPCM), acoustic echo cancelation, line echo cancelation, and caller ID. Speaker phones, modems, voice mail systems, and caller ID units use these algorithms.

1 Introduction

The TI DSP solutions for telephony applications enhance the functionality of older technologies, such as the plain old telephone service (POTS) lines. Telephones, speaker phones, modems, answering machines, and caller identification units are some telephony devices that use DSPs. Figure 1 illustrates some DSP solutions that are available for telephony equipment.

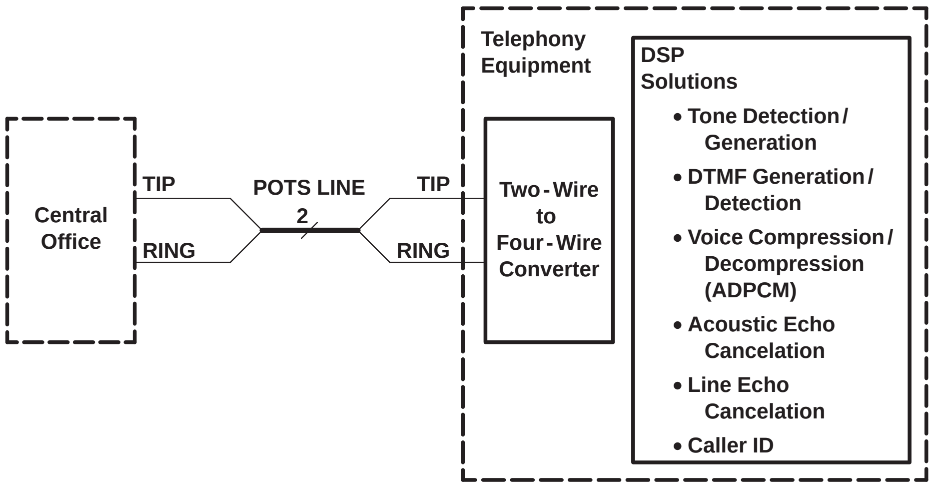


Figure 1. POTS Line Connections

1.1 POTS Line

The plain old telephone service (POTS) lines are a major part of analog telephone networks. The POTS lines connect local telephone facilities, called central offices (CO), to most household telephones (see Figure 1). The connection between the CO and the telephone is called the local loop. The POTS line was designed to carry voice signals cost effectively. A POTS line uses two conductors referred to as tip and ring.

The POTS line bandwidth and impedance is related to the length of the local loop. The local loop can be up to 15,000 feet long. The POTS line has a maximum bandwidth of 4k Hz. When the POTS signal reaches the CO, it is filtered to pass 200 to 3600Hz and converted to a digital signal. As with any transmission line the POTS line impedance varies with frequency.

The POTS line, for safety reasons, has a high impedance to earth ground. It is also resistant to the induction of other signals. Power lines (60Hz) and adjacent POTS lines induce unwanted voltages onto POTS lines. These induced voltages are hum and crosstalk. If the impedances between tip and ground and between ring and ground are similar, hum and crosstalk will be canceled by a common mode rejection technique.

Typical conversational voltages between tip and ring are in the 0- to 100-mV_{rms} range. A 1-V_{rms} signal would be considered loud.

1.2 Two-Wire to Four-Wire Converter

The early telephone circuit designers developed the two-wire to four-wire converter circuit to reduce the number of conductors required to operate the analog telephone system. The telephone's speaker and microphone circuits require four wires: two for the speaker and two for the microphone

CAUTION:

Mandatory safety components are not shown in Figure 2.

The converter circuit (see Figure 2) reduces the number of system conductors from four to two, but introduces an undesirable signal component. This component is produced by the transmitter and is called echo. Echo if too loud, is undesirable in voice and data communications. For improved voice and data communications, DSPs provide an improved echo canceler function.

The typical two-wire to four-wire converter circuit functions as an analog echo canceler. The discrete analog components model the POTS line impedance. The circuit components must approximate an *average* impedance between the many possible POTS line connections.

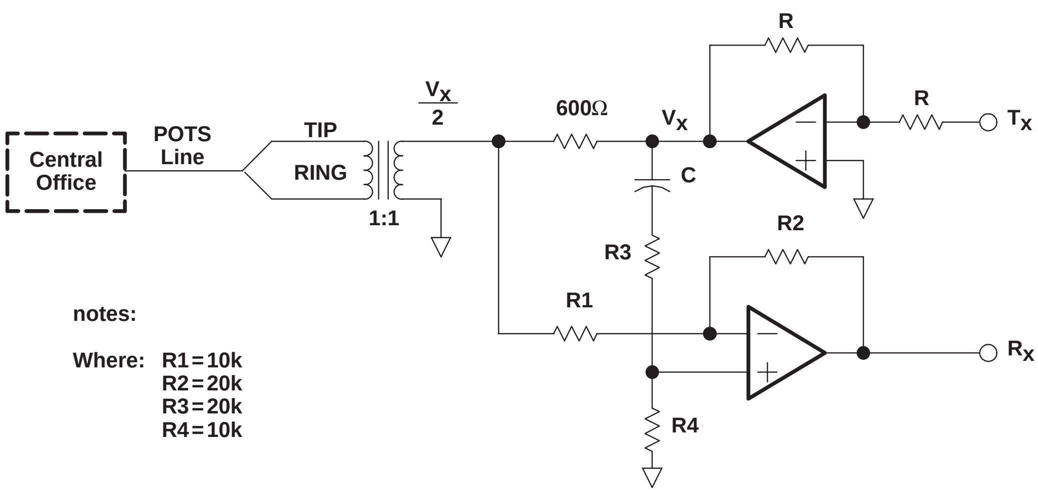


Figure 2. Simplified Two-Wire to Four-Wire Converter Circuit

The transmit op amp output signal V_x (see Figure 2) is ideally canceled completely in the receive path. However, if the analog echo canceler shown in Figure 2 is not optimum, a residual echo or sidetone will be heard in the near-end receiver. People expect to hear their own voice (sidetone) in the receiver. If sidetone is eliminated there is a perception that the circuit is "dead". Therefore, in voice applications some sidetone is desirable.

Sidetone volume levels can interfere with the receive path signal. In noisy environments (such as a noisy shop) ambient noise is fed back into the earpiece adding interference to the received signal. Increasing the volume under these conditions does not necessarily make the receive signal more intelligible. A volume control potentiometer increases the loudness of both the sidetone and the receive voice.

The two-wire to four-wire converter circuit's receive path is shown in Figure 2. The receive path starts at tip and ring and passes through the transformer. If superposition is used, the gain through the receive path op amp's two inputs can be analyzed separately.

To compute the gain of the inverting side assume the noninverting side of the op amp is connected to ground through R4. This path can be considered connected to ground due to the high input impedance of the op amp. It is also connected through a series circuit, comprised of R3 and a capacitor, to the output of the transmit path's op amp, V_x. In this circuit the transmit path op-amp output terminal is considered an ac ground since it is a voltage reference source.

Equation 1 computes the gain of an inverting op amp.

$$V_{out} = -\frac{R_F}{R_I} = V_{in} \tag{1}$$

The R_x signal begins at the inverting input of the receive path op amp. The output of the transmit op amp is defined as V_x. If an ideal line impedance of 600 ohms is assumed, the transmit op amp output signal V_x appears across the transformers as $V_x/2$. If the output R_x is expressed in terms of the transmit signal that appears at the receive path op amp inverting input, then:

$$V_{invert} = -\frac{R_2}{R_1} \cdot \frac{V_x}{2} = -V_x \tag{2}$$

Equation 3 computes the gain of a noninverting op amp.

$$V_{out} = \left(1 + \frac{R_F}{R_I}\right) V_{in} \tag{3}$$

The gain through the noninverting side of the receive op amp can be expressed in terms of V_x as (assume the capacitor to be an ac short):

$$V_{inverting} = \left(1 + \frac{R_2}{R_1}\right)\left(\frac{R_4}{R_3 + R_4}\right) V_x = V_x \tag{4}$$

If the two outputs are summed, then:

$$V_{rx} = V_{noninverting} + V_{inverting} = V_x - V_x = 0 \tag{5}$$

Therefore in the ideal case all of the transmit signal (echo) has been canceled. This is seldom the case. The simple RC circuit does a poor job of modeling the transformer and transmission line impedance. Any receive signal coming from the CO will pass through the transformer and will see a gain of –2 through the R_x op amp.

Note that approximations have been made for ease of explanation. $V_x/2$ was computed without consideration of R1 in parallel with 600 ohms.

Matching a complex impedance is difficult as the loop length varies. There is likely an impedance mismatch for most connections. The impedance for that connection also changes with frequency. This mismatch causes an echo that can be especially obvious in international calls and can interfere with modulation schemes in digital transmission. The echo is undesirable and can be reduced by more elaborate DSP-based echo cancelers (see section 2.2 for more information about echo cancelers).

2 Voice Type Applications

Since most telecommunication operations are based around voice transmission, there are many applications in which DSPs improve voice channels. The improved channel results in better voice clarity and higher data transfer rates. The following lists some DSP applications and their specific DSP functions:

Voice mail systems applications

- Tone detection and generation
- DTMF generation and detection
- Voice compression and decompression (ADPCM)

Full duplex speakerphone and modem applications

- Line echo cancellation
- Acoustic echo cancellation

2.1 Voice Mail Systems

Voice mail system products may be packaged in an answering machine or may reside in a PC. In the business environment, PC cards are commonly used to interface to phone lines. These PC cards provide record and playback functions, dual tone multi-frequency (DTMF) generation and detection, and call progress tone detection. Voice mail systems often use voice compression and decompression techniques such as ADPCM. These cards enable a PC to function as a multi-port voice mail system.

2.1.1 Tone Detection and Generation

DSPs are used in applications that detect and generate tones such as DTMF, multifrequency (MF), busy, and dial tones. DTMF detection is performed in small and large applications such as answering machines, telephones, PBXs, and CO equipment. Intra-central office signaling in E1 applications (European equivalent to T1) and US telephone frame equipment are applications that perform multi-frequency tone detection and generation functions. Busy and dial tone generation and detection capabilities are required by various telephony applications. All of these detect and generate functions are performed by DSPs on the voice channel.

The DTMF digits and their corresponding frequencies are listed in Table 1.

Table 1. DTMF Digits vs. Frequencies

Digit	Frequency 1	Frequency 2
0	941Hz	1336Hz
1	697Hz	1209Hz
2	697Hz	1336Hz
3	697Hz	1447Hz
4	770Hz	1209Hz
5	770Hz	1447Hz
6	770Hz	1447Hz
7	852Hz	1209Hz
8	852Hz	1336Hz
9	852Hz	1447Hz
*	941Hz	1209Hz
#	941Hz	1447Hz
a	697Hz	1633Hz
b	770Hz	1633Hz
c	852Hz	1633Hz
d	941Hz	1633Hz

2.1.2 DTMF Generation

DTMF tones are created by summing two sine waves. Three methods of creating tones are listed below:

- Table look-up
- Taylor series
- Harmonic resonator

To create DTMF tones, sum the proper tones together as indicated in Table1.

2.1.2.1 Table Look-Up Method

The table look-up method retrieves previously computed sine wave values from memory. The sine function is periodic and only one period must be computed. Since this is sampled data, an accurate sine wave generator must confirm that the sample's starting and ending point are the same. The easiest way to determine this is to find the smallest value of I (an integer) that when multiplied by the ratio below will result in an integer.

$$\frac{F_s}{F_o} \cdot I = \text{integer \# of samples} \tag{6}$$

Where:

F_s = sampling frequency
F_o = frequency of interest

The period of the frequency to be generated must be evenly divisible by a multiple of the sampling rate. This method can require large amounts of memory if the frequency is not an easy divisor of the sampling rate. If there are numerous frequencies to generate, or the frequency is unknown beforehand, then the table look-up method may not be the best solution.

2.1.2.2 Taylor Series Expansion

The Taylor series expansion method reduces the memory required to compute an approximation of the sine value. The accuracy can be selected. The Taylor series expansion method expresses a function by polynomial approximation. The expansion for a sine function order 5 is:

$$\begin{aligned} sin(x) &= 3.140625x + 0.02026367x^2 - 5.325196x^3 + 0.544678x^4 \\ &\quad + 1.800293x^5 \end{aligned} \tag{7}$$

where $0 < x < \pi/2$. Note that x is in radians and that the other three quadrants must be accounted for by manipulating the sign and the input value, x.

The Taylor series expansion method requires more computations but less memory than the table look-up method.

2.1.2.3 Harmonic Resonator

The third method for generating a tone is the use of a harmonic resonator. This is a direct implementation of the Z-transform of a discrete sine function, $\sin(n\omega T)$, where T = sample rate and ω = frequency to be generated in radians. Let $h(n) = \sin(n\omega T)$ and $x(n) = \delta(n)$, delta function.

$$\begin{aligned} Y(z) &= H(z)X(z) \\ Y(z) &= \frac{z sin \omega T}{z^2 - 2z cos \omega T + 1} \cdot 1 \end{aligned} \tag{8}$$

which leads to the difference equation:

$$y(n) = a_0 x(n-1) + 2cos(\omega T) y(n-1) - y(n-2) \tag{9}$$

Note that the poles are complex conjugates which lie on the unit circle and are not stable for all inputs. However, it is stable for an impulse input. It is necessary that the difference equation be implemented in this fashion whereas one would not want to perform a multiplication on the y(n–2) term. This coefficient must be 1 to avoid coefficient round-off error and possible unstable conditions. Since the input is only non-zero for one sample in time it can be implemented by pre-calculating its effect on the filter’s initial conditions such that:

$$\begin{aligned} &y(0) = 0,\ y(1) = a_0,\ y(n) = 2cos(\omega T) y(n-1) - y(n-2), \\ &n = 2, 3, \ldots \end{aligned} \tag{10}$$

Once this recursive filter is “hit” with the impulse, it will ring forever. Note that a resonator is required for each tone that is generated, and the frequency is determined by ω.

To create the required DTMF digits, eight tones must be generated and properly summed.

2.1.3 Tone Recognition With the Goertzel Algorithm

The FFT provides an efficient spectrum of tone detection over a given bandwidth, but may not be as computationally efficient as other methods of tone recognition for a given application. FFTs provide the most efficient means of spectrum analysis for evenly spaced frequencies. The FFT is derived from the discrete Fourier transform (DFT):

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{\frac{-j\,2\pi nk}{N}} \quad k = 0, 1, \ldots, N\text{–}1 \tag{11}$$

where x(n) is the time domain sequence and X(k) is the frequency domain sequence. Note that with an entire time domain sequence present, any or all values of X(k) may be evaluated. It is possible to compute one point in the frequency sequence which corresponds to one frequency. This is a computational advantage if a small number of frequencies are sought. It is important to select the sample set length, N, and the sample rate such that the frequency of interest falls on one of the points in the frequency sequence.

$$k = \frac{f_{tone}}{f_s} \cdot N \quad k = 1, 2, \ldots, N\text{–}1 \tag{12}$$

Instead of computing the DFT directly, the Goertzel algorithm is implemented as an infinite impulse response (IIR) filter. The computations are performed recursively as the sample set is being gathered instead of delaying the major processing until after the samples have been collected. If only magnitude information is required, complex multiplication can be avoided. A filter can be derived from the DFT as:

$$\begin{aligned} y(n) &= x(n) * h(n) \\ y(n) &= \sum_{m=0}^{N\text{–}1} x(m) h(n\text{–}m) \\ y(n) &= \sum_{m=0}^{N\text{–}1} x(m) e^{\frac{-j\,2\pi k(n-m)}{N}} \end{aligned} \tag{13}$$

where the filter

$$h(n) = e^{\frac{-j\,2\pi kn}{N}} \cdot u(n) \overset{z}{\leftrightarrow} \frac{1}{1 - e^{\frac{-j\,2\pi kn}{N}} z^{-1}} \tag{14}$$

To avoid a complex multiply, the above filter H(z) can be multiplied by the complex conjugate pole divided by itself to make the denominator real. This produces the filter:

$$H_k(z) = \frac{1 - e^{\frac{-j2\pi k}{N}} z^{-1}}{1 - 2cos(2\pi k/N)\ z^{-1} + z^{-2}} \tag{15}$$

This filter will evaluate the DFT at the frequency corresponding to the sequence member k for:

$$X(k) = y_k(n)|_{n=N} \tag{16}$$

For implementation into an IIR filter, let:

$$H(z) = \frac{Y(z)}{W(z)} \cdot \frac{W(z)}{X(z)} \tag{17}$$

which leads to the difference equations:

$$w_k(n) = 2cos\left(\frac{2\pi k}{N}\right) w_k(n-1) - w_k(n-2) + x(x)$$

and (18)

$$y_k = w_k(n) - e^{\frac{-j2\pi k}{N}} w_k(n-1)$$

which is illustrated in Figure 3.

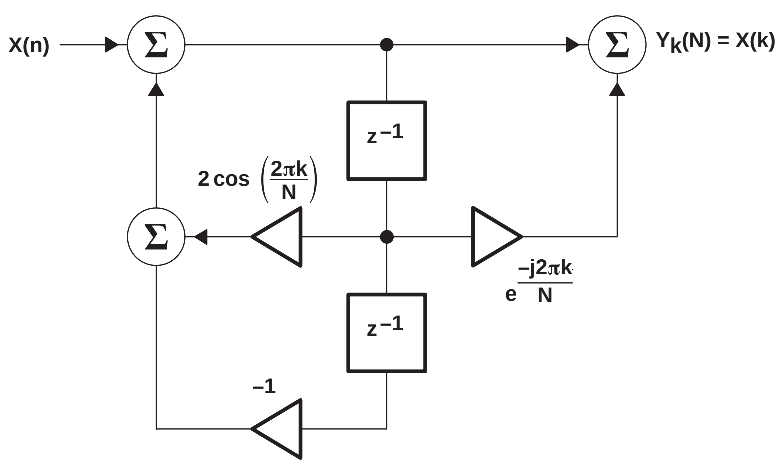


Figure 3. Goertzel Filter for Computing DFT

The filter output is only required when n=N. The value w(n) must be computed N times. If only the power magnitude is necessary the complex multiplication is not required to produce the feed-forward section since:

$$|y_k(N)|^2 = w_k^2(N) + w_k^2(N-1) - 2cos\left(\frac{2\pi k}{N}\right) w_k(N) w_k(N-1) \tag{19}$$

The feedback section stays the same. For DTMF tone recognition eight of these sections will be required; one for each frequency.

A series of logical decisions must be made after the sample set and the power of all the frequencies of interest have been computed. These decisions assure that the magnitude, relative amplitude (twist), and duration of the tones is acceptable. The duration of the off time must also be tested.

Some applications require tone detection during speech. To differentiate between speech and tones is more difficult than just recognizing tones. A safeguard must be implemented to eliminate false detects due to the tonal content of voice. An approach that has had success is to evaluate eight more frequencies, using the above method, at the second harmonics of the eight fundamental DTMF frequencies. Speech often has components at second harmonics. When there is significant second harmonic power present at these frequencies the decision-making process will deactivate the detector.

2.1.4 Voice Compression and Expansion

Audio can require a lot of storage space. How much space the audio actually requires depends on the quality of the audio and how much complexity is required to compress and decompress it. For example telephone quality audio requires 8k bytes per second with little compression/expansion (companding) complexity. That is about 1/2 M byte per minute. There are many different audio compression schemes with varying degrees of complexity. Generally, the higher the compression ratio the more complex the encoding and decoding algorithm.

2.1.4.1 Companding

Companding is a process of compressing a pulse code modulation (PCM) signal at the transmitter and expanding it at the receiver. There are two accepted companding standards, μ-law and A-law, used by the telecommunications industry. North America and Japan use μ-law while most other countries use A-law. Both standards are very similar and support low complexity companding. Toll quality is often associated with these standards, and they have a bandwidth of about 3 kHz which is not acceptable for high fidelity audio. The goal of these standards is to reduce the word length and therefore the bit-rate while maintaining the equivalent dynamic range. This is achieved by using a logarithmic step size instead of a linear one for quantization. The equation for μ-law is:

$$F(x) = sgn(x)\,\frac{ln(1 + \mu|x|)}{ln(1 + \mu)} \tag{20}$$

Where:

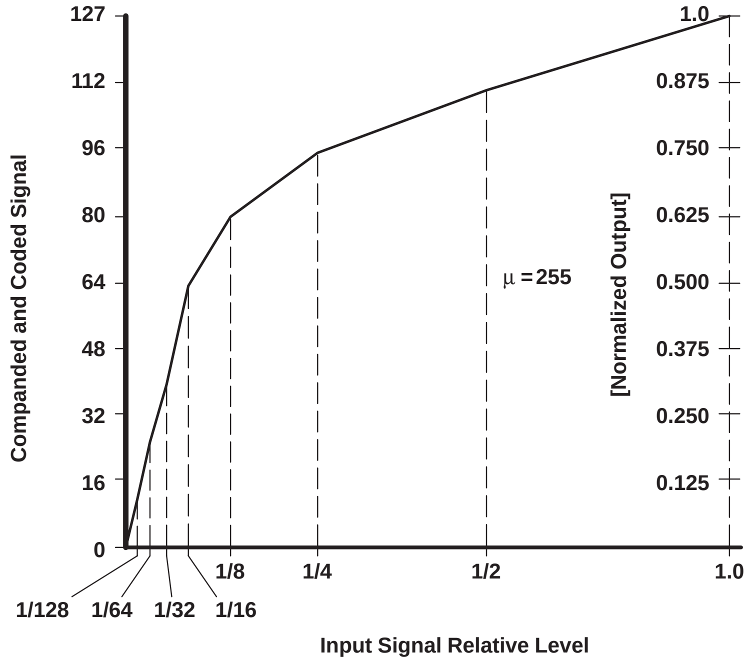
F(x) is the compressed output value
x is the normalized input signal (between –1 and 1)
μ is the compression parameter (=255 in North America)
sgn(x) is the sign (±) of x

The larger step sizes are used for larger amplitude signals. The end result is that there is more quantization error (noise) for the larger amplitudes, yet the signal-to-noise ratio is not dramatically changed. In effect, the louder sound masks the louder noise which is a common audio phenomenon. This results in compressing 13 bits of dynamic range into 7 bits - almost a 2:1 compression.

Figure 4 shows a μ-law companding curve. Eight-bit sign-magnitude words can represent 255 different code words. This made 255 the most convenient choice for the μ-law companding parameter. This companding characteristic exhibits the valuable property of being closely approximated by a set of eight straight-line segments, as shown in Figure 4. This figure shows how the input sample values of successively larger intervals are compressed into intervals of uniform size. The slope of each segment is exactly one-half that of the preceding one. The step size between adjacent code words is doubled in each succeeding segment. This property allows the conversion to and from a linear format to be done efficiently.

Character Signal Binary Code †

1	0	0	0	0	0	0	0
1	0	0	0	1	1	1	1
1	0	0	1	1	1	1	1
1	0	1	0	1	1	1	1
1	0	1	1	1	1	1	1
1	1	0	0	1	1	1	1
1	1	0	1	1	1	1	1
1	1	1	0	1	1	1	1
1	1	1	1	1	1	1	1



†This is the bit pattern transmitted for positive input values. The left-most bit is a 0 for negative input values

Figure 4. Companding Curve of the μ-Law Compander

2.1.5 ADPCM

The early telephone system was comprised of analog networks. The telecommunication industry has changed from totally analog circuits to a hybrid network that integrates analog and digital circuits. Digital circuits offer the following advantages:

- Improves signal-to-noise ratio
- Simplifies coding and decoding
- Maintains the original signal characteristics
- Provides the ability to manipulate and analyze data.
- Allows random access of stored data

A standard technique for digitizing analog signals, referred to as pulse code modulation (PCM), was developed. PCM uses waveform coding that samples a 4-kHz bandwidth 8,000 times a second, resulting in 64K bits of data per second.

Analysis of speech waveforms revealed a high sample-to-sample correlation. By taking advantage of this property more efficient coding techniques have been developed to meet the need for improved signal quality. These digitization techniques include adaptive PCM (APCM), differential PCM (DPCM), and adaptive differential PCM (ADPCM). These techniques reduce the transmission bit rate while maintaining the overall signal quality. ADPCM combines the features of APCM and DPCM by using sample-to-sample redundancy and by adapting to quantizing step sizes.

2.1.5.1 APCM

APCM is a method that can be applied to both uniform and nonuniform quantizers. It adapts the step size of the coder as the signal changes. This accommodates amplitude variations in a speech signal between one speaker and the next, or even between voiced and unvoiced segments of a continuous signal. The adaptation may be instantaneous, taking place every few samples. Alternatively, it may occur over a longer period of time, taking advantage of more slowly varying features. This is known as syllabic adaptation. The basic concept for an adaptive feedback system, APCM, is shown in Figure 5.

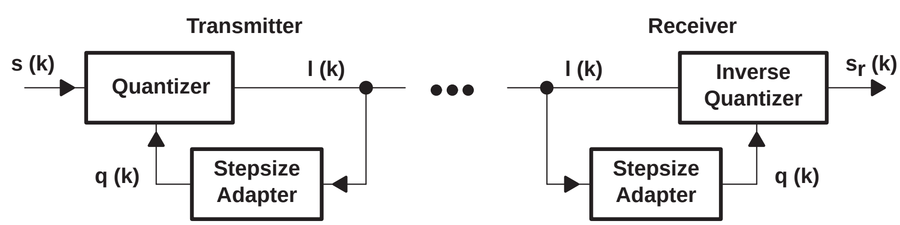


Figure 5. APCM Block Diagram

An input signal, s(k), to the transmitter is quantized and coded to an output, I(k). This output is also processed by the stepsize adapter logic to create a signal, q(k), that adapts the stepsize in the quantizer. Correspondingly in the receiver, the received signal, I(k), is processed by an inverse quantizer (i.e., decoded), producing the reconstructed signal, $s_r(k)$. Like the transmitter, the quantized signal, I(k), is processed by the stepsize adapter logic to create a stepsize control signal, q(k), for the inverse quantizer.

2.1.5.2 DPCM

An audio digitization technique that codes the difference between samples is known as differential PCM (DPCM). The high sample-to-sample correlation of speech waveforms indicates that the difference between adjacent samples produces a waveform with a much lower dynamic range. Correspondingly, an even lower variance can be expected between samples in the difference signal. A signal with a smaller dynamic range may be quantized to a specific signal-to-noise ratio with fewer bits. A DPCM system is shown in Figure 6.

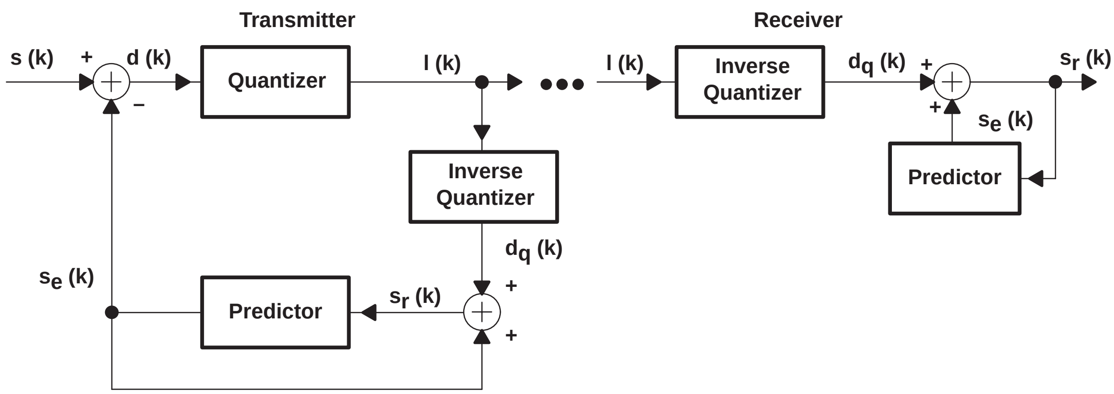


Figure 6. DPCM Block Diagram

In Figure 6, the signal difference, d(k), is determined using a signal estimate, s_e(k), rather than the actual previous sample. By using a signal estimate, s_e(k), the transmitter uses the same information available to the receiver. Each successive coding actually compensates for the quantization error in the previous coding. In this way, the reconstructed signal, s_r(k), can be prevented from drifting from the input signal. s(k), as a result of an accumulation of quantization errors. The reconstructed signal, s_r(k), is formed by adding the quantized difference signal, d_q(k), to the previous signal estimate, s_e(k). The sum is the input to adaptive predictor logic which determines the next signal estimate. A decoding process is used in both the transmitter and receiver to determine the quantized difference signal, d_q(k), from the transmitted signal, I(k).

2.1.5.3 ADPCM

ADPCM is the ANSI de facto standard for digitizing audio signals at 32k bps. Figure 7 shows that both quantizer adaptation and differencing requires the storage (in memory) of one or more samples in both the transmitter and receiver.

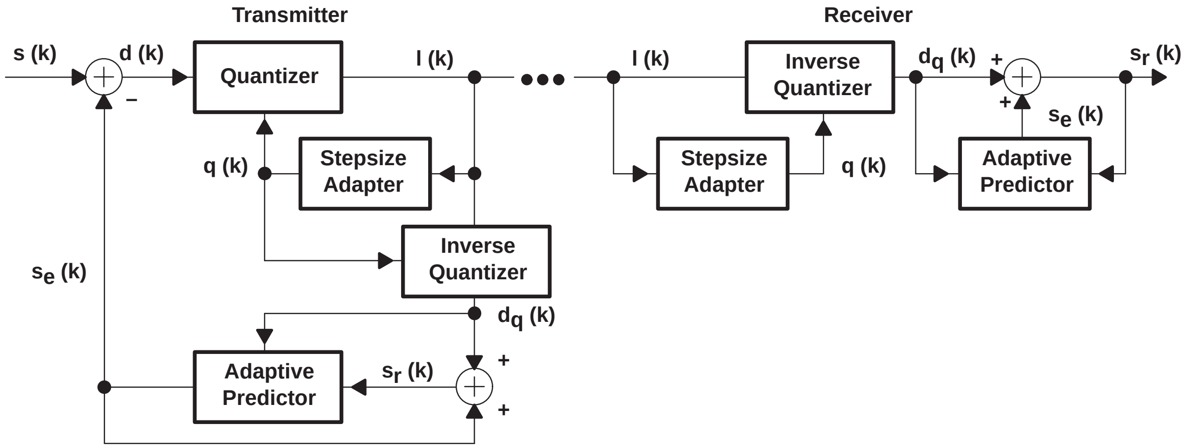


Figure 7. ADPCM Block Diagram

Furthermore, the transmitter must use some method to insure that the receiver is operating synchronously. This is accomplished by using only the transmitter signal, I(k) to determine stepsize adaptation in the quantizer and inverse quantizer and to predict the next signal estimate. In this way, the blocks in the receiver can be identical to those in the receiver. Additionally, the specific adaptation techniques are designed to be convergent and thereby help provide quick recovery following transmission errors.

Companies that manufacture voice processing equipment have made ADPCM their product line's coding algorithm standard. ADPCM is used in many voice mail recorders in the PC environment to store voice data to a hard disk. ADPCM can achieve toll quality speech with a 4-bit data word at 32 kbps: a 2:1 compression ratio compared to companded PCM.

2.2 Echo Cancelers in Full Duplex Speakerphones and Modems

In telephony applications DSPs are used to perform line and acoustic echo canceler functions (see Figures 8 and 9). Because the acoustical impulse response is more complex and lasts longer than the line impulse response, the acoustic echo canceler function requires more DSP processing power than the line echo canceler function.

The most difficult processing task in echo cancellation is identifying the echo path. The echo path is unknown for both the line side and the echo side and it can change relatively slowly over time. The solution requires estimating the echo path and adapting to changes in it. The estimation includes more than just the delay path because attenuation may occur at certain frequencies and many reflections may occur simultaneously. For this reason it is best to look at an acoustic or line echo as a signal which undergoes change by a transfer function.

Two examples of echo canceler applications are high speed modems and full duplex speaker phones. High speed modems use line echo cancelers to reduce inter-symbol interference caused by line impedance discontinuities and reflection. Full duplex speaker phones use both line and acoustic echo cancelers. In this application the line echo canceler function is similar to the modem line echo canceler. The acoustic echo canceler function reduces or removes acoustical echo.

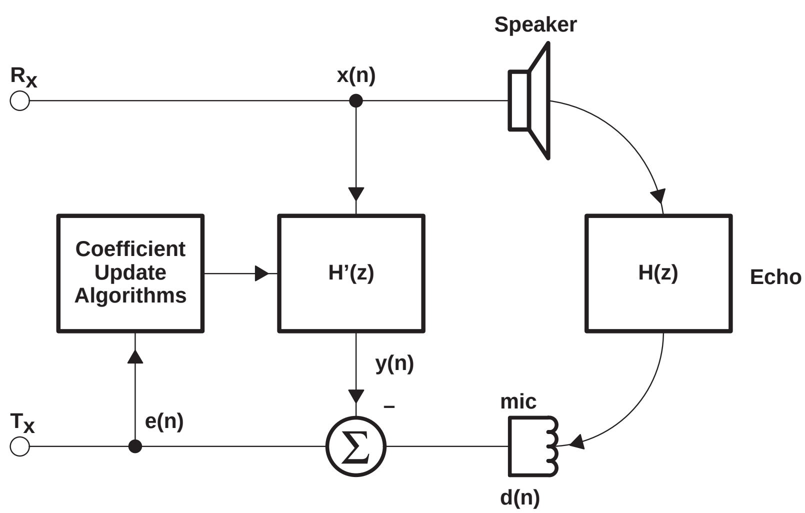


Figure 8. Acoustic Echo Canceler

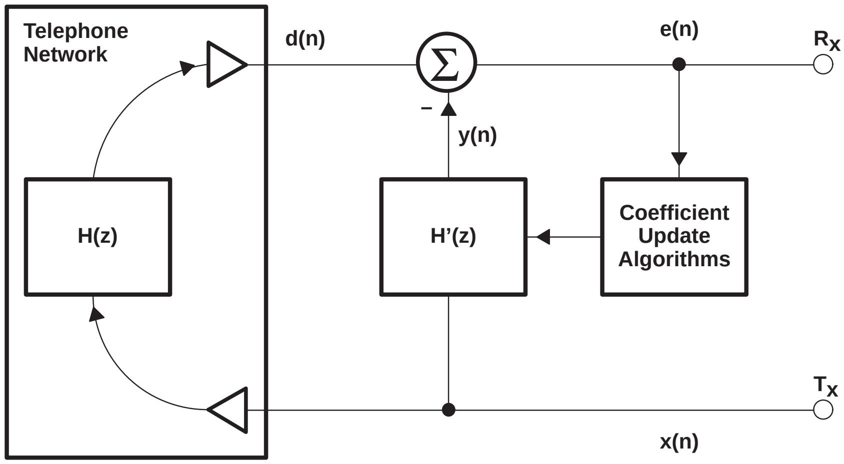


Figure 9. Line Echo Canceler

2.2.1 The Canceler Algorithm

The basic line echo canceler can be seen in Figure 9. A reference signal, x(n), is passed through some environment (telephone network) represented by H(z). When the signal appears again in the receive path, d(n), it has been filtered by H(z). The object is to subtract this exact signal from the receive path which leaves the error signal e(n). This is accomplished by passing the reference signal x(n) through a filter, H'(z), that approximates H(z) and renders y(n). The impedance of the telephone network varies and the acoustic environment varies. Therefore, an algorithmic means of determining what the filter H'(z) should be to match H(z) is needed as the changes occur. There are many adaptive filter types. Finite impulse response (FIR) filters are used most, are more stable, and provide a good basis for understanding canceler algorithms. The FIR implementation is:

$$y(n) = \sum_{k=0}^{N-1} h(k)x(n-k) \tag{21}$$

where the filter coefficient h_k will be changed by the update routine. This filter must be long enough to be able to approximate the impulse response and the associated delay of the transfer function it is trying to match. A typical office acoustic environment has an impulse response and delay of about 50 ms. A typical line connection has an impulse response and delay of about 13 ms. For a sample rate of 8000 samples/sec, this leads to about 400 taps for the acoustic canceler filter and about 100 taps for the line canceler filter.

2.2.2 Coefficient Update

There has been a lot of research done to develop coefficient update algorithms. The recursive least mean square (LMS) algorithm is the most commonly used update algorithm (see Figure 10). This algorithm provides a low cost way to determine the optimum (Wiener) filter coefficients without explicitly computing the cross correlation and auto correlation sequences**.** The goal is to minimize the sum of squared error ***E***:

$$E = \sum_{n=0}^{M} e^2(n) = \sum_{n=0}^{M}\left[d(n) - \sum_{k=0}^{N-1} h(k)x(n-k) \right]^2 \qquad (22)$$

where e(n) is the error signal. To find the minima of the above expression with respect to the filter coefficients we could solve:

$$\frac{\partial E}{\partial h(m)} = 0 \qquad (23)$$

The algorithm is a recursive gradient (steepest descent) that determines the minimum value of *E*. The process starts with any arbitrary choice for the initial values of the filter coefficients (usually all equal to 0). After each new sample input x(n) enters the adaptive FIR filter the corresponding output y(n) and the error signal e(n) are computed and the filter coefficients are updated according to:

$$h_n(k) = h_{n-1}(k) + \Delta e(n)x(n-k) \qquad (24)$$

Where:

K = 0,1,…, N – 1 and n = 1,2,…

where Δ is the step size parameter and x(n–k) is the sample of the input signal located at the k'th tap of the filter at time n. A large value of Δ will lead to a large step size and faster convergence while a small step size may lead to better overall cancellation. However, if the step size becomes too large with respect to the input power the adaptation can become unstable. To ensure stability Δ must be chosen in the range:

$$0 < \Delta > \frac{1}{10NP_x} \qquad (25)$$

where N is the filter length and P_x is an estimate of the power of the input signal x(n). Since P_x can vary quite a bit with speech, it may be necessary to normalize the error signal e(n) based on the P_x. The estimate of the power of the input signal (P_x) can be approximated by:

$$P_x = \frac{1}{M} \sum_{n=0}^{M-1} x^2(n) \qquad (26)$$

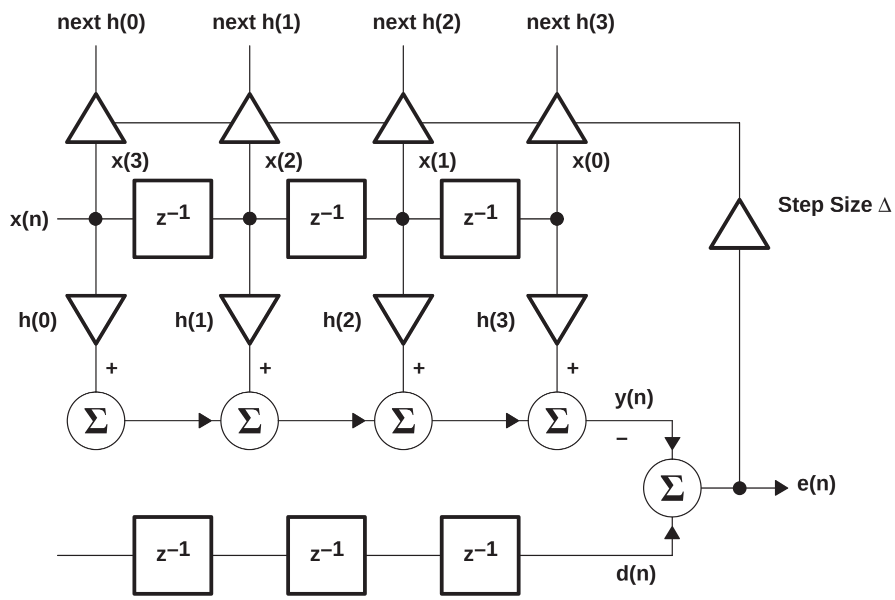


Figure 10. LMS Adaptive Filter Block Diagram

The steepest descent approach updates the coefficients after each new sample. In Figure 8, acoustic echo canceler, if the coefficient is updated when the near end person is talking or if the far end power is too low, a large coefficient error will result. To eliminate these error conditions, coefficient updating must be controlled. This requires supervisory code that monitors far end power and near end voice. The supervisory code is responsible for suspending coefficient updating when required.

2.2.3 System Considerations for Full Duplex Speakerphone

Up to this point only the echo canceler algorithm and implementation has been described. The closed loop nature of the system requires a robust method of determining where the signal originated. For this application the block diagram in Figure 11 shows the line echo canceler and the acoustic echo canceler connected together. In effect, this gives a signal a path around a closed loop. The gain around this path must be less than one for stability. This is especially important when the filters have not converged to some optimum approximation. At the beginning of a conversation when the filters are first beginning to adapt, their error is very large. In addition, when there is more in a signal beside the echo (such as a local talker speaking into the microphone) the error will be large and the filters coefficient should not be updated. When no far-end signal is present, there is essentially no echo and the filters coefficient should not be updated. These points make implementation of the full duplex speakerphone especially difficult. To cope with these constraints several signal detectors are needed around the loop. For instance, near end d(n), and far end x(n) power detectors are needed for each filter in addition to an error monitor. Half duplex speakerphone operation is required until the filters have been trained (the filters converge). Once converged, the gain switching mechanism can be turned off of the system and the phone will achieve full duplex operation.

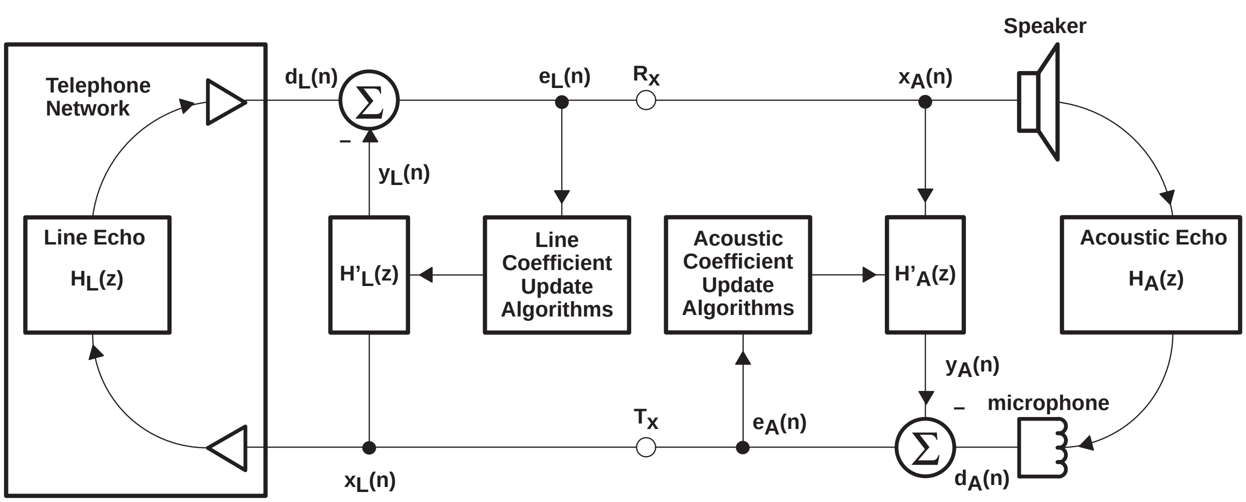


Figure 11. Simplified Full Duplex Speaker Phone Diagram

3 DSP Requirements for Caller Identification

Caller identification (CID) information is transmitted from the telephone company, via the local loop, to the subscriber's CID unit. The information is transmitted between the first and second rings. The CID unit must be capable of interpreting the CID protocol defined in Bellcore TR-NWT-000030 and also of displaying and storing the calling parties name, number, and other types of information.

A half duplex modem operating at 1200 bps is the standard method of communicating CID information. The fundamental frequencies are defined by the Bell 202 specification (see Figure 12). This format is basically continuous phase frequency shift keying (FSK) with a mark frequency (f_m) of 1200 Hz and a space frequency (f_s) of 2200 Hz.

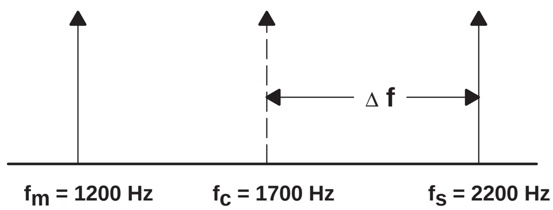


Figure 12. Frequencies for Bell 202 FSK

An advantage of the FSK format is that an FSK detector (see Figure 13) can be non-coherent (not require phase information about the carrier) and perform well. The performance degradation between this type of non-coherent detector compared to a coherent detector is about 3 dB worst case.

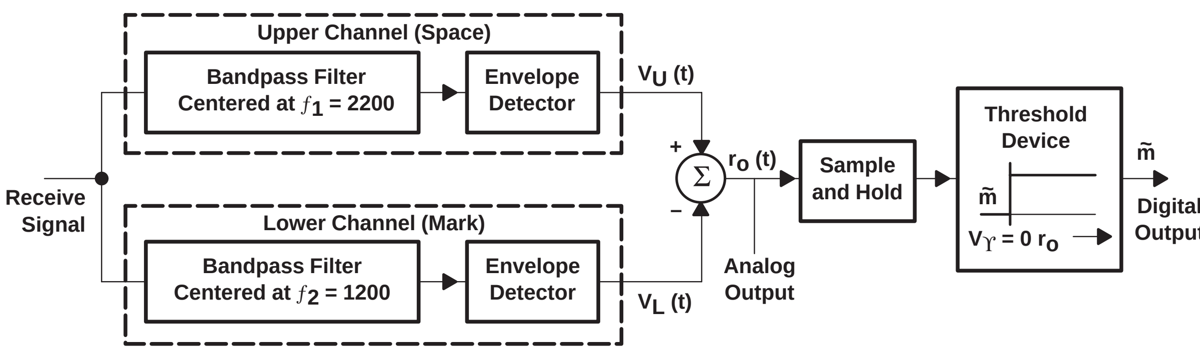


Figure 13. FSK Noncoherent Phase Detector Block Diagram

References

1. Haykin, Simon, *Adaptive Filter* Theory, Prentice-Hall 1986.
2. Proakis, John G, *Digital Signal Processing Principles, Algorithms, and Applications*, Prentice-Hall 1996.
3. Poularikas, Alexander D, *Signals and Systems*, PWS Engineering 1985.
4. Rymer, Jay, ”32 kbps ADPCM with the TMS320C10”, *Digital Signal Processing Applications with the 320 Family,* Texas Instruments Inc., 1989, Literature Number SPRA012A.
5. Couch, Leon, *Digital and Analog Communication Systems*, Macmillan Publishing Company 1990.
6. Proakis, John, *Digital Communications*, McGraw-Hill 1995.

Appendix A Performance Requirements

Table A–1. Analog Telephone System Perfomance Requirements for the TMS320C5x

Function	MIPS	Data Memory Words	Program Memory Words
DTMF Generation	0.7	160	140
DTMF Detection	1.0	350	500
Acoustic Echo Canceler	7.2	600	750
Line Echo Canceler	2.7	200	400
ADPCM Encoder/Decoder	11.0	300	2000
Caller ID	1.0	200	500

Part II
DSP Solutions for Data/Facsimile Modems

DSP Solutions for Data/Facsimile Modems

ABSTRACT

This application report describes the functional blocks of analog data and facsimile modems. The report highlights the critical role of the DSP in these modems. Chapter 5 presents a system level perspective of analog modems. Appendix A provides MIPS/Memory information for V.32bis and V.34. Appendix B provides a quick reference to the common International Telecommunications Union (ITU) standards.

1 Introduction

A modem is a device containing a modulator and a demodulator. The communication channel is the general switched telephone network (GSTN), a two-wire, or a four-wire leased circuit. The bandwidth of the channel ranges from 200 Hz to 3400 Hz. Earlier split-band modems (V.22bis and prior) separated the bandwidth of operation. The transmitting modem sent information in the low band and received in the high band. Similarly, the receiving modem received information in the low band and transmitted in the high band. As the demand for speed grew, modem algorithms became more complex. With the advent of V.32 modems with a data rate of 9600 bps, V.32bis modems with a data rate of 14400 bps, and V.34 modems with a data rate of 28,800 bps required the modem to send and receive within the same band. These modem algorithms incorporated echo cancellation to achieve this channel separation.

Since the environment uses analog signals as a means of communication and the capabilities of analog signal processing are limited, analog signals are converted to a digital format before being processed by a DSP. A DSP is a single chip microcomputer dedicated to performing signal processing functions.

Why is the DSP so important? A microprocessor in a PC or a simple microcontroller cannot perform the complex functions that a DSP performs since digital signal processing involves the extensive use of the multiply and add operations. General purpose microprocessors break the multiplication and division operations into a series of shift, add, or subtract operations; therefore, microprocessors do not execute complex signal processing tasks as fast as a dedicated DSP.

2 Modulation Methods

The common modulation techniques are amplitude modulation (AM), frequency modulation (FM), and phase modulation (PM). AM is seldom used since AM is susceptible to noise bursts and signal fading. FM, also known as frequency shift keying, is inexpensive. Modems operating at speeds of 1200 bps or less used FM. For speeds above 1200 bps, modems combine phase and amplitude modulation.

Differential phase shift keying (DPSK) is a technique where the relative phase difference of the current signal element is measured with respect to the previous signal element. The demodulator detects the difference in phase. DPSK systems are classified by the number of phase states provided for signal detection. In a DPSK-2 system, a 0 digit encodes as a +90 degree phase change and a 1 digit as a +270 degree phase change relative to the previous signal element. Using more than one bit per signal increases the data transfer rate. For example, a DPSK-4 system encodes two binary digits (a dibit) every signal element, giving four possible phase states: 00, 01, 10, and 11. A dibit value of 00 represents a phase change of +270 degrees, 01 a phase change of +180 degrees, 10 a phase change of +90 degrees, and 11 a phase change of 0 degrees. A system using DPSK-8 modulation encodes three binary digits (a tribit) every signal element. A high-level of DPSK encoding entails greater complexity in generating and detecting small phase changes and involves an increased susceptibility to jitter in phase during transmission.

A signal element or a symbol is a *change in state* of the circuit.[1] A baud is the number of transitions or state changes that occur in a channel and is expressed in symbols-per-second. Simply speaking, 1 baud = 1 symbol/sec. When a manufacturer advertises a 14400 or 28800 baud modem, the numbers 14400 and 28800 specify the bit rate of the modem, not the baud rate.

For example, a DPSK-4 modulation scheme defines four possible states for a symbol. In this example, the baud is equal to two bits/symbol. If the channel is operating at 600 baud, the effective bit rate is 600 symbol/sec $\times$ 2 bits/symbol = 1200 bps full duplex. This example is the configuration of the Bell 212A modem. Using the same configuration, 2400 bits are transmitted every second in the half-duplex mode of the Bell 201 modem.

Quadrature amplitude modulation (QAM) is another modulation scheme where the amplitude and phase is varied. The QAM system is two-dimensional. Using this modulation, a QAM signal is represented as:

$$x(t) = a(t)\cos(\omega t) + b(t)\sin(\omega t) \qquad (1)$$

where $\omega = 2\pi f$ and f is the carrier frequency. The two bit streams *a(t)* and *b(t)* are obtained from the incoming data stream.

For example, consider the V.29 scheme transmitting at 7200 bps. The transmitting bits are divided into a tribit, which represents eight possible states. Each state represents a phase change relative to the previous signal state. The absolute phase determines the amplitude of each element. If the phase of the element is 0, 90, 180 or 270 degrees, the amplitude is three. If the phase is 45, 135, 225, or 315 degrees, the amplitude is set to $\sqrt{2}$. The combination of phase and amplitude is useful if the state of a symbol undergoes a phase change due to noise in the channel. The plot of the phase and amplitude points on a two-dimensional graph gives a constellation chart. Figure 1 shows the V.29 constellation for 7200 bps QAM encoding.

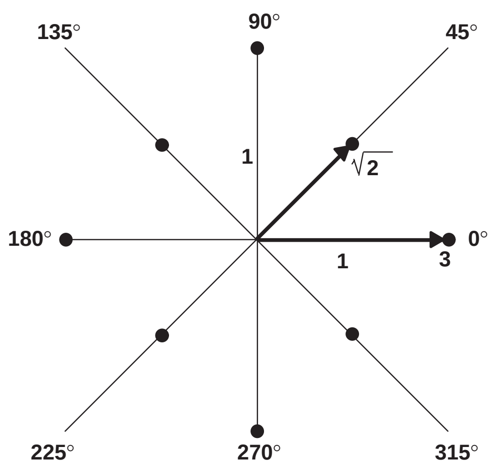


Figure 1. V.29 Constellation for 7200 bps QAM Encoding

A new V.34 standard resulted from the attempt to increase the line rates from V.32bis (14400 bps) and V.32ter (19200 bps). A key distinction between V.34 and the older standards is the availability of different baud rates. V.34 also supports asymmetric baud rates. This support means that line conditions at the originating and answering sides determine the baud rate selected by either side. These two rates may be different. These different baud rates allow the system to maintain a high throughput without having to retrain or retransmit data.

3 Transmitter of the Modem Data Pump

Figure 2 shows the V.32 modem transmitter section of the data pump.[1]

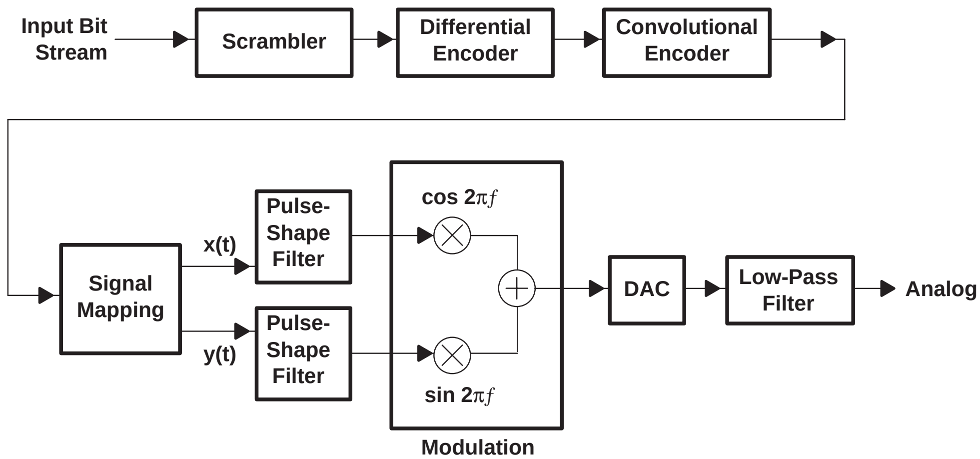


Figure 2. V.32 Modem Transmitter

3.1 Data Scrambling

Are modems synchronous or asynchronous? Modems are asynchronous on the PC side and synchronous with respect to each other. Modems remain synchronous with each other through a clock signal. This clock signal is not part of the data stream, otherwise the clock signal would impose an overhead on the data transmission.

Asynchronous modems derive their clock signal from the data. In some cases, the data is a long stream of unchanging bits. To avoid this case and ensure that the energy of the modulated carrier is white or spread over the frequency band, the ITU specifies using a data scrambler. The scrambler takes the input serial bit stream and converts the stream into a pseudo-random sequence. The scrambler is defined using a polynomial. For example, the V.32bis standard specifies the following:

originating modem polynomial (GPC) as $1 + x^{-18} + x^{-23}$ (2)

answering modem polynomial (GPA) as $1 + x^{-5} + x^{-23}$ (3)

where x is the input sample. The exponent of x signifies the time delay. The value x^{-23} means the twenty-third previous sample of x. The additions are bit-wise XOR functions.

The transmitting modem divides the message sequence by the GPC. The coefficients of the quotient, in descending order, form the output of the scrambler. At the receiver, the data sequence is multiplied by the GPA to recover the message sequence.

The scrambler uses delay lines, XOR functions, and circular buffers. Figure 3 shows a block diagram of a call mode scrambler in the originating modem.[1] The serial input bit stream is x_{in} and the scrambled output is D_s. Each z^{-1} is a delay block (delay line) and each addition is an XOR function.

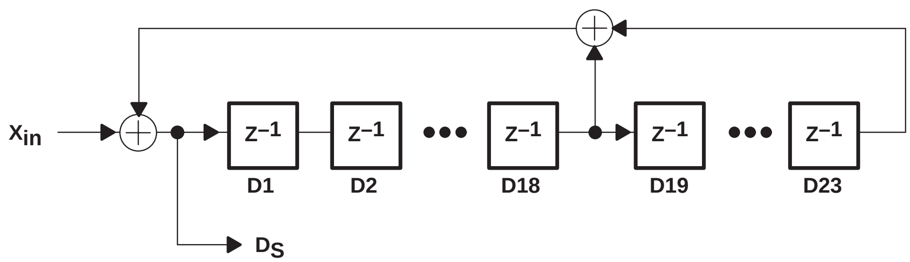


Figure 3. Call Mode Scrambler

3.2 Trellis Coding Fundamentals

For modem standards beyond V.32, as the distinct states move closer together in terms of amplitude and phase, errors become increasingly difficult to correct. The reliability of a QAM system is improved by using a forward error-correction scheme called Trellis coding.

A Trellis coding algorithm accepts m data bits as input and generates m+1 bits as output. The convolutional-coding scheme used with the Trellis coding algorithm generates an extra bit that is redundant information. This scheme was used for the first time with the V.32 standard that specifies using a 16-point constellation or a Trellis coded 32-point constellation. With the Trellis scheme in V.32, the transmit data stream is divided into groups of four bits. The first two bits are differentially encoded and then convolutionally encoded to generate three bits. The other two bits are not encoded, but sent to the output stage. The output is therefore, five bits.

In Figure 4, bits Q1 through Q4 are the four input bits. Bits Q3 and Q4 are not encoded and sent to the output directly. Bits Q1 and Q2 are encoded to give Y1, Y2, and the redundant error correcting bit, Y0.

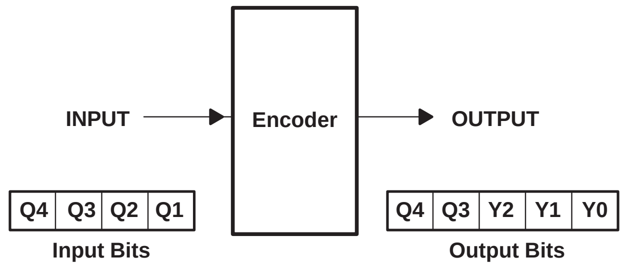


Figure 4. V.32 Encoder

3.2.1 The Redundant Bit

In addition to adding a redundant bit to improve performance, Trellis coding uses a technique to partition the signal. In Figure 4 the five bits provide thirty-two modulator states. The encoded bits, Y0, Y1, and Y2, divide the thirty-two states into eight subsets of four modulator carrier states, so Y0, Y1, and Y2 identify the subset. The uncoded bits, Q3 and Q4, identify a point within the subset. Figure 5 shows the constellation chart for a V.32 modem.

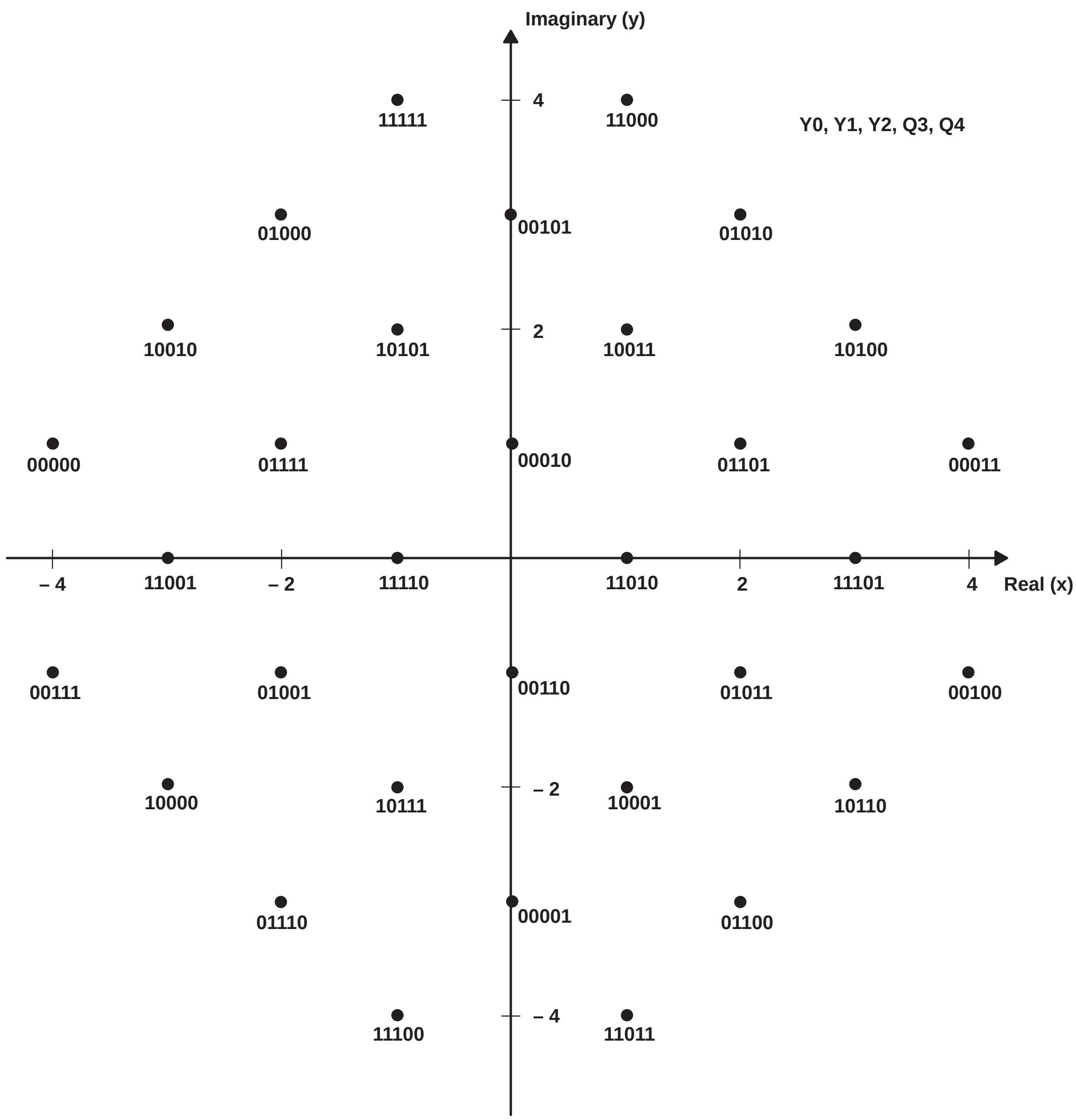


Figure 5. V.32 Signal Constellation

All points beginning with the same three bits (Y0, Y1, Y2) are separated. This type of partitioning and the presence of redundant coding form the basis of Trellis coding.

The V.32bis standard for a bit rate of 14400 bps specifies that the scrambled data stream be divided into groups of six data bits. The output of the encoder provides six bits and one redundant bit for a total of seven bits, thus contributing 2^7 or 128 points on the constellation chart.

The symbols generated in the Trellis encoder, map into the signal space defined in the V.32/V.32bis specification. This mapping generates two outputs or coordinates, the real part and the imaginary part, that are input to the QAM modulator.

For V.32 and V.32bis modems, the output of the scrambler feeds the Trellis encoder. This path does not apply to V.34 modems.

3.3 V.34

Figure 6 shows the transmitter section of a V.34 modem [2].

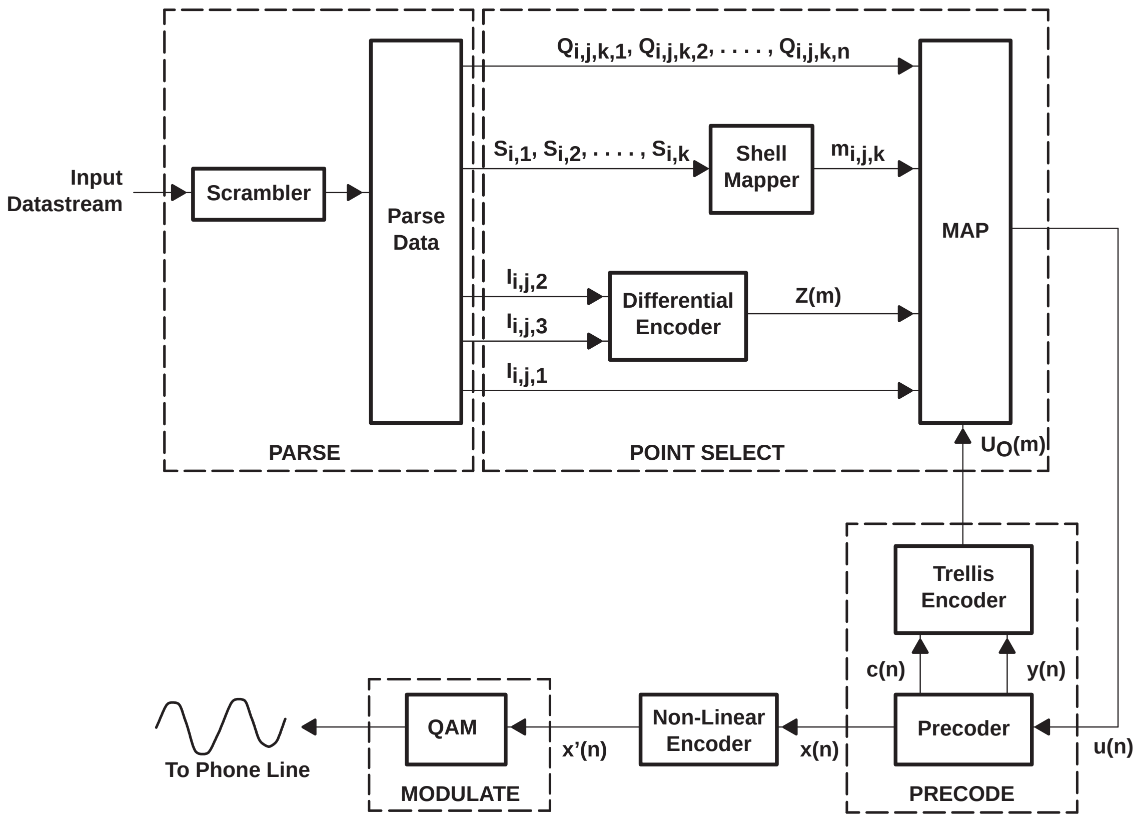


Figure 6. V.34 Modem Transmitter

3.4 Line Probing

Line probing is a bidirectional half-duplex exchange performed immediately after V.8. V.8 is the first start up procedure which uses V.21 modulation to exchange information, such as the type of V.34 modem, modulation modes available, availability of V.42 and V.42bis, etc. Line probing involves the transmission of complex signals, that allow the distant receiver to analyze the characteristics of the line. Based on this information, the modems decide:

- Carrier frequency and symbol rate. The modems decide on an optimum carrier frequency and symbol rate. The six symbol rates are three mandatory rates of 2400, 3000, 3200 symbols/sec and three optional rates of 2743, 2800, 3429 symbols/sec. Each rate has two possible carrier frequencies.
- Pre-emphasis selection. Pre-emphasis is made adaptive based on the line characteristics. The idea behind adaptive pre-emphasis is to compensate for known channel distortions before the distortions happen. The V.34 algorithm adapts by selecting the optimum transmit preemphasis filter from the menu of ten defined filters. The transmitted signal passes through a spectral shaping filter that boosts the signal in some parts of the spectrum while attenuating the signal in others.
- Power level selection. The modems select an optimum transmit level. With the older V.22bis, a high transmit power was better. With the advent of echo-cancellation modems, a high-transmit power introduces unwanted echo at the local end; a very low transmit level lowers the basic signal to noise ratio. Striking a fine balance is necessary.

Line probing is a tonal exchange performed every time the modem makes a new connection and does a full retrain. Line probing allows the modem to adapt to the varying characteristics of a given line.

3.5 Point Selection Unit

V.34 uses a unique method of selecting points from a two-dimensional constellation. The V.34 superconstellation consists of 960 points. Two-dimensional points are selected from a subset of this superconstellation; the subset contains 240 points. The entire superconstellation is obtained from the subset by rotating it by 0, 90, 180 or 270 degrees. Each combination of the symbol rate and data rate uses a particular subset of the superconstellation.

Figure 7 shows the constellation diagram for a subset of the superconstellation. The point with the smallest magnitude is 0 and is closer to the origin of the constellation, the point with the next higher magnitude is labeled 1, and so on. This ordering implies that the points on the outer rings have greater power. To minimize the average power of the transmitted signal, the algorithm uses a point selection scheme that chooses points from the inner rings more often than from the outer rings. The point selection unit contains three sub-units: shell mapper, differential encoder, and mapper.

The parser partitions the binary data from the scrambler into groups of bits. One of these groups is sent to the shell mapper, two groups are sent to the differential encoder, and the others are sent unchanged to the block marked MAP (see Figure 6).

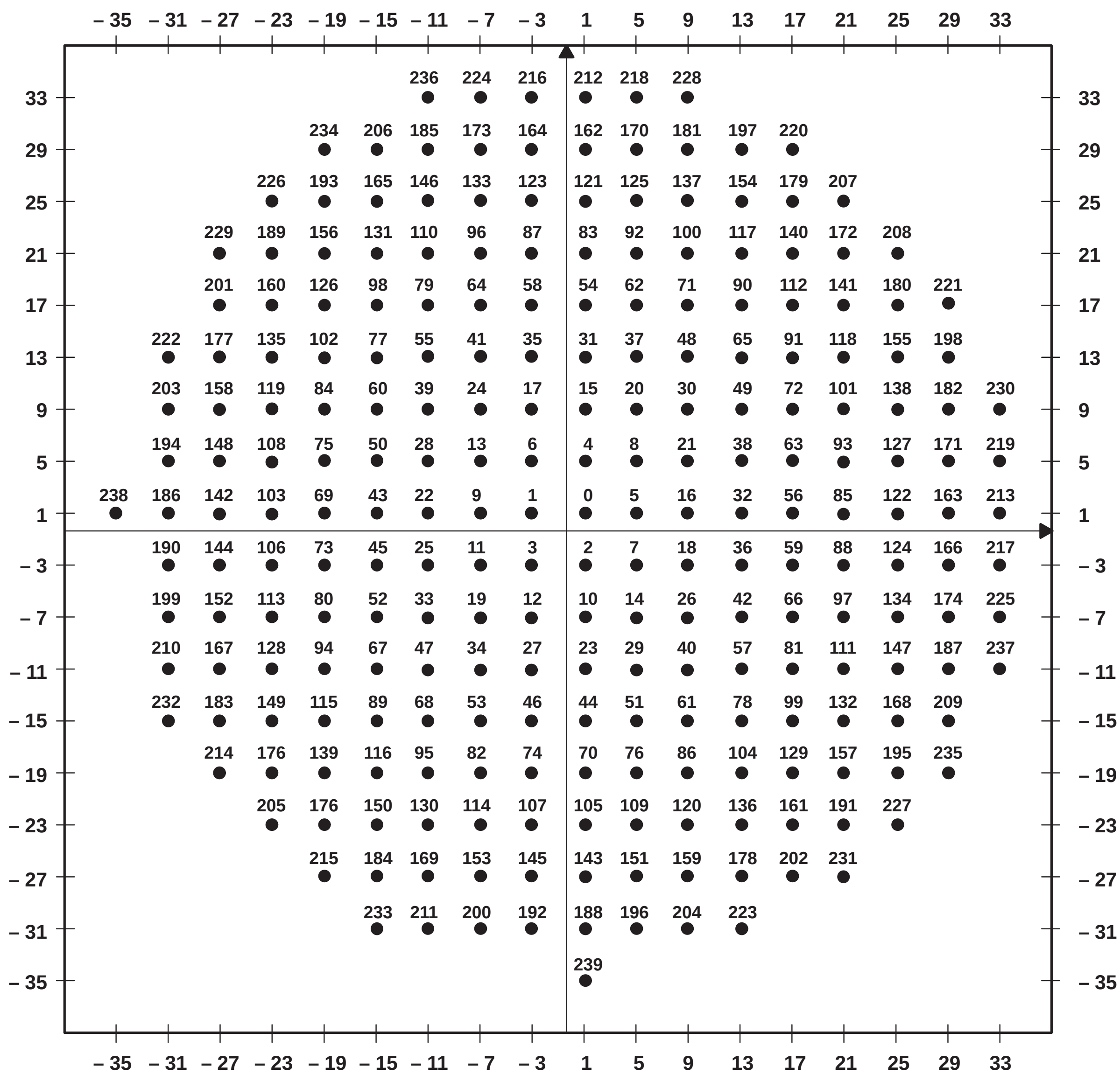


Figure 7. 240-Point Quarter Superconstellation

3.6 V.34 Data Framing

V.32bis encodes four bits of data using a 2400 baud rate to give a total bit rate of 9600 bps. V.34 uses the same principle with a more sophisticated technique of selecting particular sequences of two-dimensional points. This point selection helps in minimizing the transmitted signal power and maximizing the possibility of decoding the encoded data at the receiver correctly.

In V.34, the output sequence of two-dimensional points is divided into sub-sequences of eight points called mapping frames. A mapping frame (see Figure 8) is viewed as the smallest unit of output from the transmitter. The amount of data encoded in a mapping frame varies according to the data transmission and symbol rates.

For example, a bit rate of 28,800 bps and a symbol rate of 3200 symbols/sec gives:

$$28800 \text{ bits/sec} \div 3200 \text{ symbols/sec} = 9 \text{ bits/symbol}$$

Since each mapping frame is eight points or symbols, multiplying nine bits/symbol by eight symbols/mapping frame is seventy-two bits/mapping frame (denoted as ‘b’ in the V.34 spec). The values of J and P in Figure 8 are specified for a given symbol rate.[3 (V.34/Table7)]

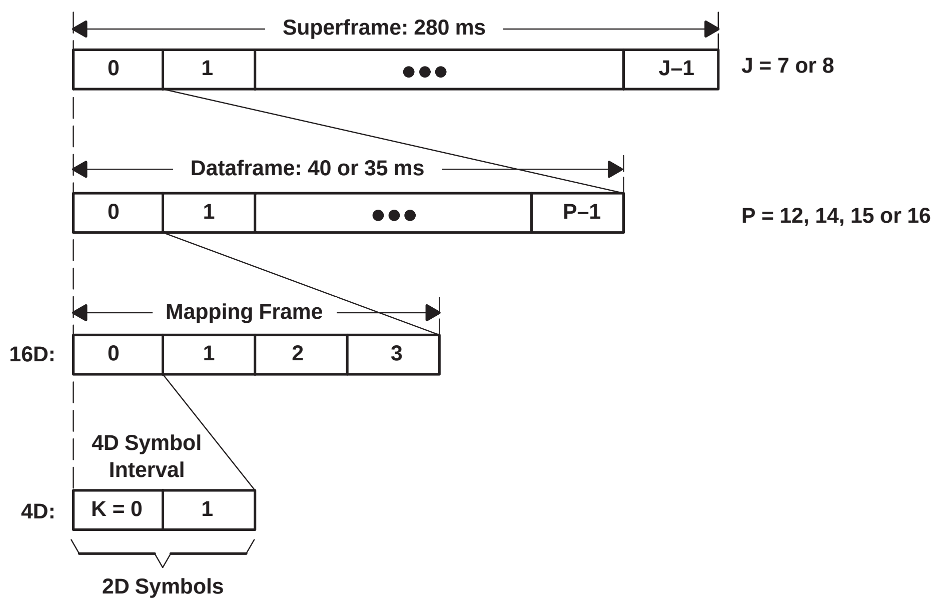


Figure 8. V.34 Framing Conventions

Returning to Figure 6, the input to the shell mapper is a bit stream $S_{i,1}$, $S_{i,2}$, ..., $S_{i,K}$. The subscript *i* refers to the time index of the mapping frame. The value of K is specified in the V.34 specification for a given symbol and data rate. These K data bits are taken from the data stream coming from the scrambler. The value of K for a bit rate of 28800 bps and symbol rate of 3200 symbols/sec is 28 (from Table 10/V.34 of the V.34 specification); therefore, twenty-eight bits out of the seventy-two bits in a mapping frame create the S sequence of bits. The remaining forty-four bits (72–28) are divided into four groups of equal size. The first three bits in each group are denoted by $I1_{i,j}$, $I2_{i,j}$, $I3_{i,j}$ where $0 \le j \le 3$. The remaining $2q = \{[(b–K)/4] –3\}$ bits are divided into two subgroups of size q denoted by $Q_{i,j,k,1}$, $Q_{i,j,k,2}$, ..., $Q_{i,j,k,q}$ where $0 \le k \le 1$. A mapping frame i consists of the following sequence of bits:

$$\begin{aligned}
&(S_{i,1},S_{i,2},\ldots,S_{i,K}), \qquad (4)\\
&(I1_{i,0},I2_{i,0},I3_{i,0}),\ (Q_{i,0,0,1},Q_{i,0,0,2},\ldots,Q_{i,0,0,q}),\ (Q_{i,0,1,1},Q_{i,0,1,2},\ldots,Q_{i,0,1,q}),\\
&(I1_{i,1},I2_{i,1},I3_{i,1}),\ (Q_{i,1,0,1},Q_{i,1,0,2},\ldots,Q_{i,1,0,q}),\ (Q_{i,1,1,1},Q_{i,1,1,2},\ldots,Q_{i,1,1,q}),\\
&(I1_{i,2},I2_{i,2},I3_{i,2}),\ (Q_{i,2,0,1},Q_{i,2,0,2},\ldots,Q_{i,2,0,q}),\ (Q_{i,2,1,1},Q_{i,2,1,2},\ldots,Q_{i,2,1,q}),\\
&(I1_{i,3},I2_{i,3},I3_{i,3}),\ (Q_{i,3,0,1},Q_{i,3,0,2},\ldots,Q_{i,3,0,q}),\ (Q_{i,3,1,1},Q_{i,3,1,2},\ldots,Q_{i,3,1,q}).
\end{aligned}$$

where $S_{i,1}$ is the earliest bit in time and $Q_{i,3,1,q}$ is the latest

In summary, the K bits labeled S are sent to the shell mapper, the four groups of three bits labeled I are sent to the differential encoder, and the eight groups of q bits labeled Q are sent uncoded. Also, the values of K and q depend on the transmission rate.

3.6.1 Use of Bits

Figure 9 shows the shell mapper using the K-S bits to generate a sequence of eight rings.[4]

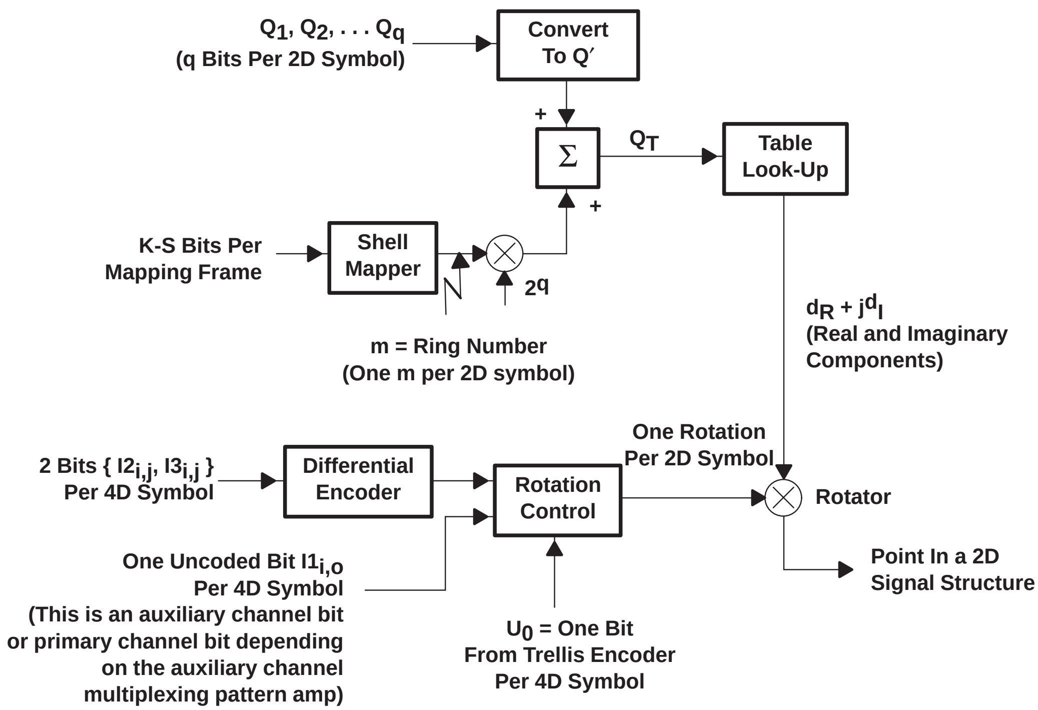


Figure 9. Overall Mapper in Data Mode

The eight groups of q bits, labeled Q, select a point within each ring from the one-quarter superconstellation. The four groups of three bits, labeled I, along with the output of the Trellis encoder are used by the differential encoder to rotate one four-dimensional symbol at a given time. For one mapping frame, the differential encoder is applied four times (see Figure 8) since each mapping frame has four pairs of two-dimensional points. This rotation results in the generation of points from the entire superconstellation.

Shell mapping is part of the overall mapping technique. The shell mapper distributes the signal points from the two-dimensional space into a near-spherical or four-dimensional shape, achieving an expansion of the constellation. This technique improves the signal to noise ratio by approximately 1 dB. The V.34 standard specifies two levels of shell mapping: the first expands the constellation by 12.5 percent and the second by 25 percent.

The telephone channel exhibits nonlinear distortion due to the PCM coding of analog signals. Under excessive nonlinear distortion, it is advisable to omit the constellation expansion. Another technique that combats the effects of nonlinear distortion is warping. Nonlinear distortion affects the outer constellation points more severely than the inner ones; warping compromises the noise immunity of the inner points in favor of the more sensitive outer points. Warping is achieved by increasing the mean distance between the outer points of the constellation. This gives the points a greater noise immunity while reducing the mean distance between the inner points.

3.7 Nonlinear Precoder

In practice, the physical communications channel introduces distortion in the signal. The distortion is compounded by analog components such as the transformer and loading coils. A signal passing through such impairments is subject to inter-symbol interference (ISI).

To combat the effects of ISI, an equalizer is used at the receiver. An equalizer uses a filter to create an equalized response that has zero ISI and channel distortion. Decision feedback equalizing (DFE) is one such technique. DFE is based on a principle that once the value of the current transmitted symbol is determined, the ISI contribution of that symbol to future received symbols can be removed.[5] A decision device determines which symbol of a set of transmitted values is used.

For V.34, DFE is split between the transmitter and the receiver. The receiver calculates the equalizer coefficients and relays them to the transmitter. The transmitter uses these coefficients to pre-emphasize the signal before transmission.

The receiver contains a noise whitening filter, given by the transfer function:

$$H(z) = 1 + F(z) \tag{5}$$

where $F = h_1 z^{-1} + h_2 z^{-2} + h_3 z^{-3}$.

The V.34 transmitter contains the inverse of this filter and precodes the transmitted signal points. This precoding pre-emphasizes the points in a certain way so that the points are decoded properly at the receiver after the noise whitening process.

In the precoder of Figure 26:

$$y_n = U_n + c_n \tag{6}$$

$$x_n = y_n - p_n = U_n + c_n - p_n \tag{7}$$

The quantization error or dither is $d_n = c_n - p_n$. Substituting the dither into Equation 7 gives:

$$x_n = U_n + d_n \tag{8}$$

Since the dither, d_n, and the constellation point, U_n, are constrained in magnitude, x_n is constrained; therefore, the precoder can not become unstable. As $|d_n|$ is very small, $x_n \approx U_n$.

In terms of Z transforms:

$$x(z) = \frac{y(z)}{1 + F(z)} \tag{9}$$

where $F(z) = h_1 z^{-1} + h_2 z^{-2} + h_3 z^{-3}$.

This process of pre-emphasis destroys the Trellis sequence of the transmit signal. To correct this problem, V.34 uses *Trellis coding with the precoder*. The precoder generates a correction term, C(n) once every four-dimensional symbols. The correction term and the output, Y(n), are XORed to form U_0, which is sent to the block marked MAP (see Figure 6).

Precoding enhances the overall modem performance especially on channels with a substantial amplitude-frequency distortion. On most channels (lines), precoding allows the use of a higher symbol rate than those without precoding; therefore the signal constellation size is reduced for a given data rate. Tests show that precoding offers approximately 3.5 dB improvement in the signal-to-noise ratio. This ratio is quite high when compared to using only an equalizer in the receiver.

3.8 Trellis Coding

V.34 uses Trellis coding in a feedback loop. The three Trellis coders specified in V.34 are 16-state (see Figure 10), 32-state, 64-state. The receiving modem selects the type of encoder during phase 4 of the handshake. During this phase, the modems exchange modulation parameters (MP) pertinent to data transmission. The encoder type is exchanged using MP sequences within phase 4 of the V.34 handshake. The MP sequence contains information such as the maximum data signal rates, amount of constellation shaping, nonlinear encoding parameters (used by the remote transmitter), support of asymmetric data signaling rates, etc.

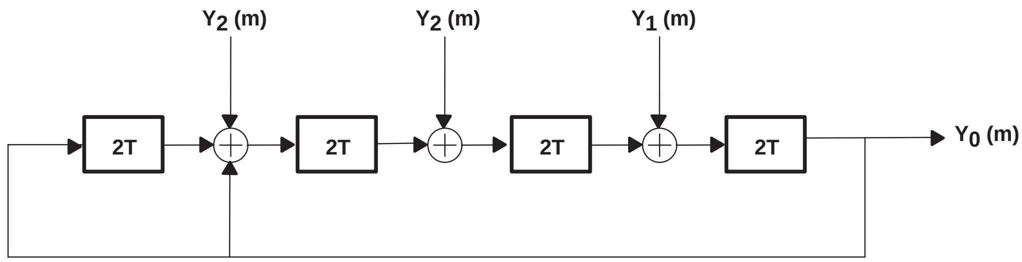


Figure 10. V.34 16-State Convolution Encoder

Figure 11 shows a block diagram of the V.34 Trellis encoder. The term, U_0(m), is generated by the modulo encoder, C_0(m), the convolutional encoder, Y_0(m), and a bit inversion pattern, V_0(m) (used for superframe synchronization). Y_0(m) is computed once every four-dimensional symbols and the resultant term U_0(m) selects the next four-dimensional point.

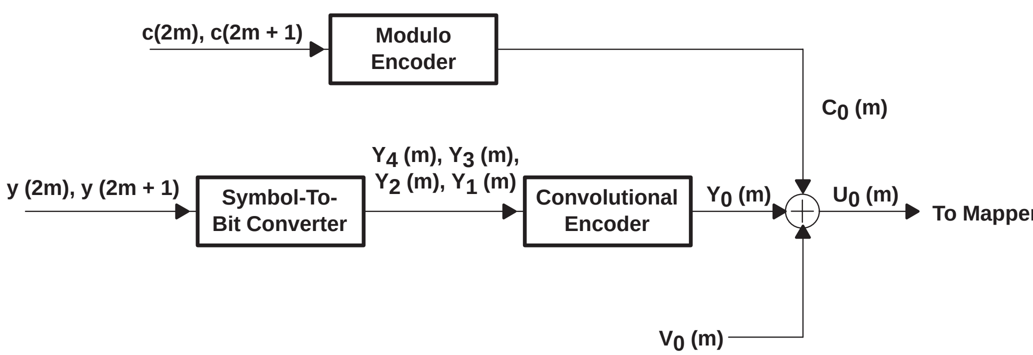


Figure 11. V.34 Trellis Encoder

3.9 Modulator

Figure 12 shows the modulator and pulse shaping filter, which is the final block of the transmitter section of the modem data pump.

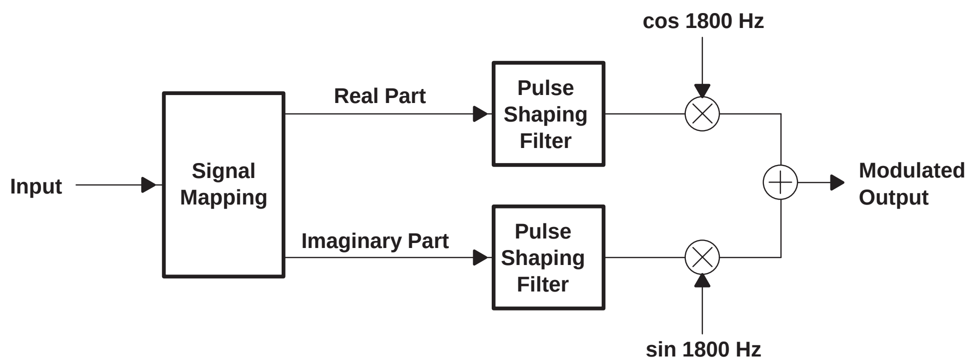


Figure 12. Modulator and Pulse-Shaping Filter

V.34 uses QAM. The encoded digital sequence is modulated using a carrier frequency of 1800 Hz. The D/A converter converts the sequence into an analog signal for transmission over the phone lines. QAM modulation results in the generation of two parts, a real and an imaginary (from Equation 1). The information bearing digital sequences in equation 1 are *a(t)* and *b(t)*.

The sampling rate at the D/A converter is configured during initialization. If the sampling rate of the D/A is 9600 samples/sec and the symbol rate for data transmission is 2400 symbols/sec, every symbol is represented by four samples.

Coupled with the modulator and demodulator are pulse shaping low pass filters. These filters are essential in preventing ISI due to the band-limited GSTN channel. The most common pulse shaping filters are the raised cosine pulse shaping filters.

3.9.1 Raised Cosine Pulse Shapers

For a band-limited channel, it is required that the transmitted signal or pulse does not produce any ISI. This behavior is exhibited by the sinc function which in the time domain is:

$$g(t) = \sin(\pi t/T)/(\pi t/T) \tag{10}$$

In the frequency domain, the sinc function is:

$$\begin{aligned} G(f) &= T \text{ when } f < 1/2T \\ &= 0 \text{ when } f \geq 1/2T \end{aligned} \tag{11}$$

where T is the baud rate.

The zero values of this pulse function occur at multiples of T. ISI occurs when the tails of the receive pulses overlap at the sample points, so the previous symbol contributes to the next symbol. The placement of the nulls (or zeroes) in the sinc function eliminates the ISI due to the previous symbol.

An ideal filter which shapes the input pulse into a sinc function is hard to implement. A raised cosine pulse shaping filter with an impulse response of zero ISI, is easier to implement. The impulse response of such a filter is:

$$p(t) = \frac{\sin(\pi t/T)\ \cos(\propto \pi t/T)}{(\pi t/T)\ (1 - (2 \propto \pi t/T)^2)} \tag{12}$$

where T is the symbol rate in Hz, t is the sampling rate in Hz, and $\propto$ is the roll-off factor.

The roll-off factor determines the amount of excess bandwidth required. As the value of $\propto$ varies from 0 to 1, the excess bandwidth increases from 0 to 100 percent.

The raised cosine pulse shaping filter is implemented as a finite impulse response (FIR) filter.

4 Receiver of the Modem Data Pump

Figure 13 shows the V.34 receiver block diagram. The receiver consists of two blocks: the demodulator and the decoder. The QAM demodulator converts the received analog signal into signal points. The decoder uses the signal points to determine the most likely Trellis sequence of the constellation points and decodes each mapping frame into the original sequence of points.

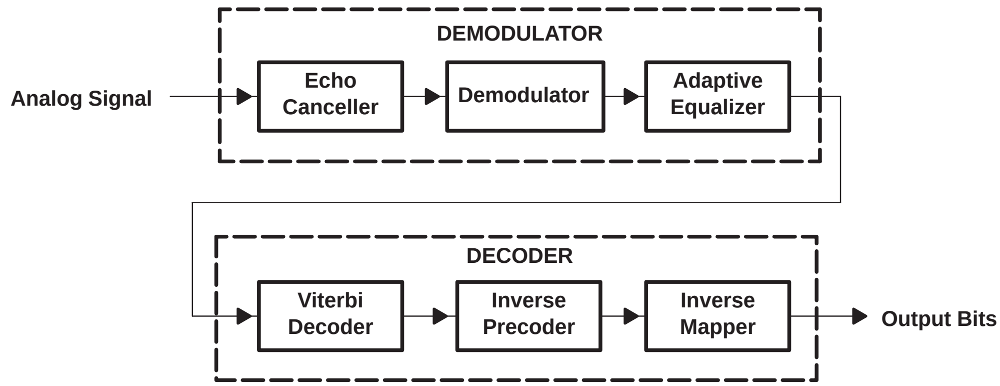


Figure 13. V.34 Receiver Block Diagram

4.1 Echo Cancellation

When a call is placed from the home, the call is routed to the telephone central office through a local loop. The length of the loop varies depending on the proximity to the central office. From there, the call is routed through a digital trunk line or a satellite link. Once the call reaches the destination central office, the call is routed to the final destination (home, office, etc.). The entire process consists of different connections through the network.

The customer phone line is a two-wire connection. This two-wire connection is converted into a four-wire connection at the central office. The process of converting from a two-wire connection to a four-wire connection is accomplished using the hybrid circuit of Figure 14.[1]

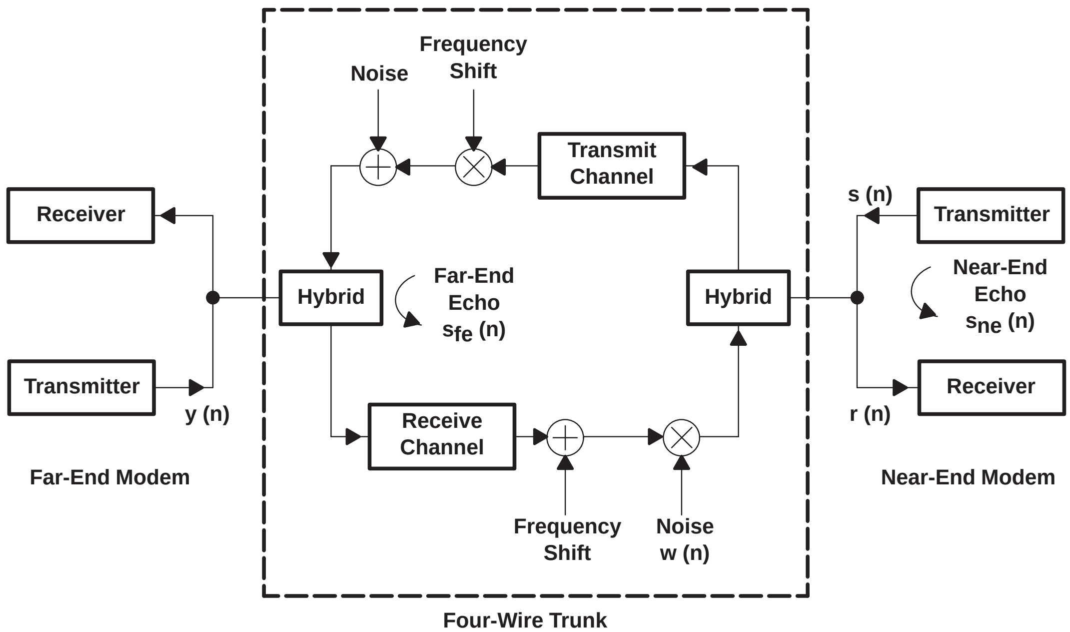


Figure 14. Telephone Channel Showing Hybrids

Modem circuits in the U.S. are matched to a 600 Ω resistive termination. Since the hybrid circuit introduces an impedance mismatch, part of the transmitted signal is reflected back as an echo. The two types of echo in the circuit are: near-end and far-end echo. Near-end echo is the reflection of the transmitted signal from the local hybrid. Far-end echo is the signal reflected from the far-end hybrid.

The receive signal level is quite low. Since all high-speed modems send and receive in the same band, the following components are within the same band: a strong transmit signal, a weak receive signal, a strong local echo, and a weak far-end echo. Distinguishing between the different signals and retrieving only the received signal becomes important.

Two items added to a signal going through the telephone network are white noise and frequency offset. White noise is due to the channel. Frequency offset occurs when the local oscillators used throughout the network are not exactly matched. This mismatch causes a shift between the carrier frequencies of the far-end echo and the transmitted signal. This shift affects the amount of echo cancellation that can be attained. Compensation for this frequency offset is necessary.

If the near-end modem is transmitting a signal, s(n), and the far-end modem is transmitting a signal, y(n), the near-end received signal, r(n), is (see Figure 14):

$$r(n) = y(n) + s_{ne}(n) + s_{fe}(n) + w(n) \tag{13}$$

where $s_{ne}(n)$ is the near-end echo, $s_{fe}(n)$ is the far-end echo, and $w(n)$ is the white noise added due to the channel.

Echo cancellation is achieved by subtracting an estimate of the echo from the received signal. The received signal after echo cancellation is:

$$r'(n) = y(n) + [s_{ne}(n) - \hat{s}_{ne}(n)] + [s_{fe}(n) - \hat{s}_{fe}(n)] + w(n) \tag{14}$$

where $\hat{s}_{ne}(n)$ is an estimate of the near-end echo and $\hat{s}_{fe}(n)$ is an estimate of the far-end echo.

In ideal circumstances, the estimates of the near-end and far-end echoes equal the near-end and far-end echoes respectively, thus forcing the terms in the brackets to equal zero.

An estimate of this echo is generated by feeding the transmitted signal through an adaptive filter. The transfer function of the filter models the communication channel (the phone channel). A least mean squared (LMS) algorithm is used to update the coefficients of the adaptive filter (refer to *DSP Solutions for Telephony*).

Figure 15 gives an overview of a modem circuit echo canceller.[4]

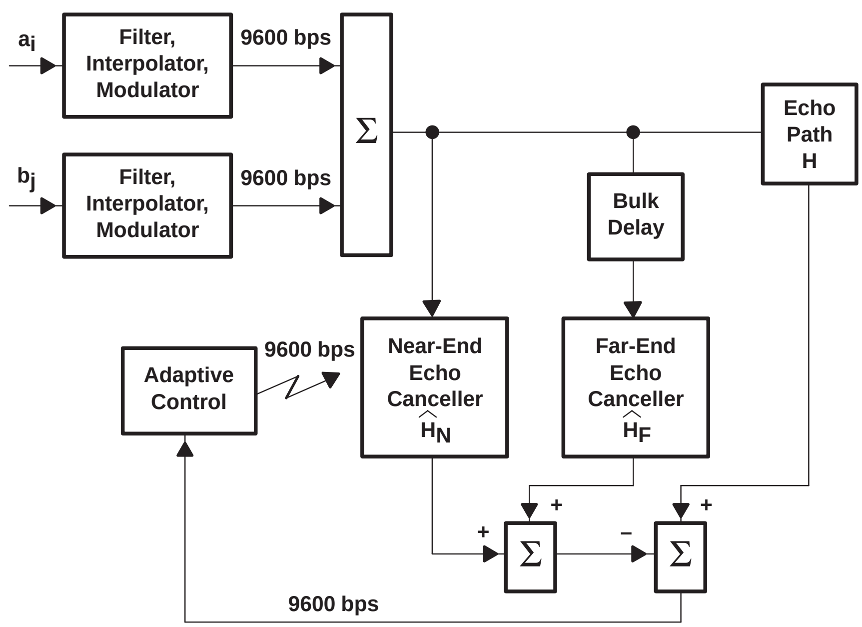


Figure 15. Echo Canceller Overview – Data Mode

Training of the echo canceller takes place during phase 3 of the V.34 handshake using the signal TRN. Signal TRN is a sequence of symbols generated by applying binary ones to the input of the scrambler.

4.1.1 Fast Training of the Echo Canceller

V.34 specifies fast training of the echo canceller. The entire training process of the echo canceller, the equalizer, and the fine training that follows the fast training must be completed in less than two seconds. Employing a fast training scheme for near-end and far-end echoes is difficult due to:

- the range of the far-end echo delay
- the echoes spreading the train sequence causing the near-end and far-end responses to overlap in time.

Designers use a training scheme with a transmitted sequence possessing special auto-correlation and cross-correlation properties to measure the impulse response of the echo path. Since the schemes are proprietary, details are not available.

Fast training of the far-end echo canceller necessitates an estimation of the frequency shift or translation on the round-trip echo path. This estimation is done during phase 2 of the handshake (probing/ranging) using tones A, A~, B, B~, and probing signals L1 and/or L2. Fast training of the echo canceller occurs during phase 3 of the handshake.

The probing signal, L1, is a periodic signal with a repetition rate of 150 ± 0.01% Hz. The signal consists of a set of tones (cosines) spaced 150 Hz apart with frequencies from 150 Hz to 3750 Hz. The tones at 900 Hz, 1200 Hz, 1800 Hz, and 2400 Hz are omitted. L1 is transmitted for 160 ms at 6 dB above the nominal power level. Usually L1 is the probing signal measuring frequency shifts and L2 refines the shift measured. Tones A, A~, B, and B~ are used to estimate the round-trip delay.

The V.34 standard specifies the instance at which estimation of the round-trip delays and measurement of the local echoes of the probing signals occur. The generation and detection of probing signals is an essential function of the DSP. A wrong echo estimate causes a potentially bad connection.

4.2 QAM Demodulator

The QAM demodulator recovers the original bit stream from the incoming analog signal. The incoming signal is fed through a Hilbert transformer that splits the analog signal into two components that are 90 degrees out of phase with respect to each other. The two resulting signals are multiplied by the quadrature components of the carrier and applied to a low pass filter to recover the baseband components (see Figure 16).

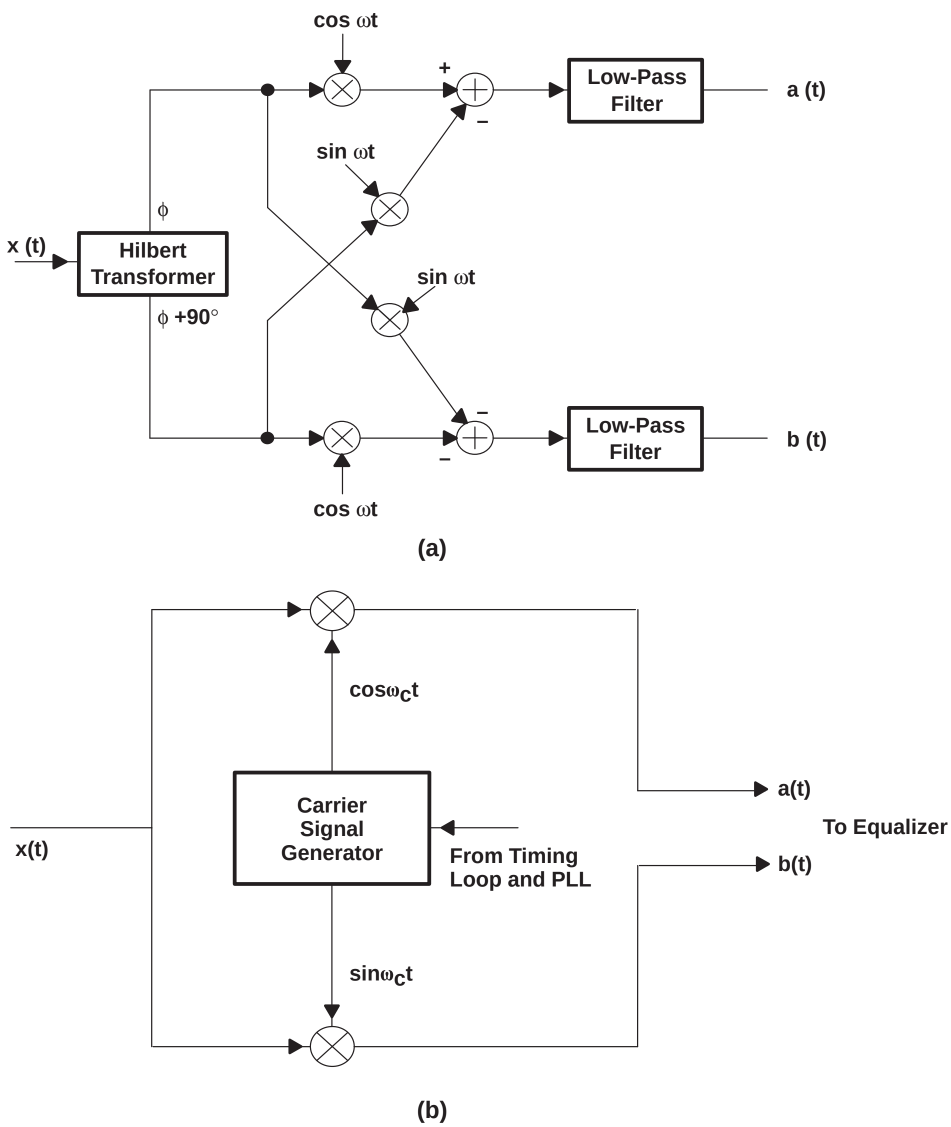


Figure 16. QAM Demodulators

Another method of demodulating the incoming analog signal is to multiply the signal by two quadrature carriers. The multiplication results in the following terms:

$$\begin{aligned} x(t)cos(\omega_c t) &= a(t)cos^2\omega_c t - b(t)sin\omega_c t \; cos\omega_c t \\ &= \frac{a(t)}{2} + \frac{a(t)}{2}cos2\omega_c t - \frac{b(t)}{2}sin2\omega_c t \end{aligned} \tag{15}$$

$$\begin{aligned} x(t)sin(\omega_c t) &= a(t)cos\omega_c t \; sin(\omega_c t) - b(t)sin^2\omega_c t \\ &= \frac{a(t)}{2}sin2(\omega_c t) - \frac{b(t)}{2} + \frac{b(t)}{2}cos2\omega_c t \end{aligned} \tag{16}$$

The information bearing components *a(t)* and *b(t)* are recovered by passing each sequence through a filter that filters the terms centered around $2\omega_c$.

The received signal is not the same as the transmitted signal since the signal undergoes distortion in the communications channel. To state that the receiver decodes the exact sequence of bits as sent by the transmitter is an understatement and a complex process.

Ideally, if the receiver samples the same received signal sent by the transmitter at the same symbol rate as the transmitter, then the exact sequence of symbols is obtained. This is not practical due to:

- The sampling rate at the receiver is not the same as the symbol rate at the transmitter because of different operating environments, clock frequencies, etc. Crystals from the same manufacturer, rated at the same frequency, generate differing clock signals.
- The received signal does not match the transmitted signal because of non-linearity in its propagation path.

The first problem is handled by the clock recovery mechanism, that corrects the sampling frequency of the receiver to match the symbol rate. The second problem is addressed by the adaptive equalizer.

4.3 Clock Recovery Mechanism

A sampling rate of 9600 samples/second at the analog front end of a modem and a symbol rate of 2400 symbols/second, yields four samples for every symbol. The sampling instants are (1/9600) + t, where t represents the clock phase. The clock recovery mechanism keeps the value of t approximately equal to zero. The analog front ends (AFE) or codecs used for V.34 feature a programmable sampling rate. This programmable sampling rate allows the selection of a sampling frequency that is an integer multiple of the selected symbol rate. For example, symbol rates of 2400 and 3200 use a sampling rate of 9600 Hz.

Figure 17 shows an approach to the clock recovery problem.[4]

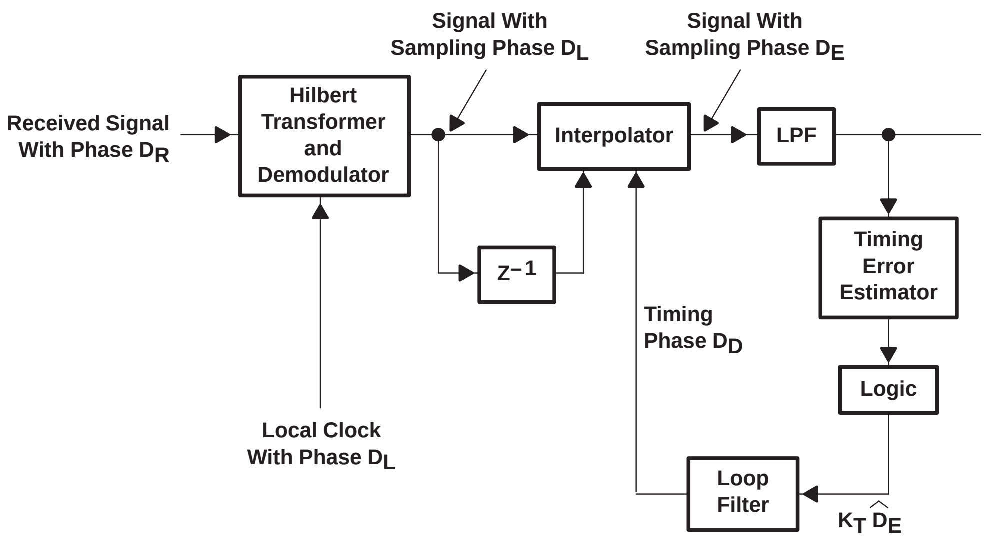


Figure 17. Clock Recovery

The incoming signal phase is D_R. The local clock at the receiver generates samples with a phase of D_L (local samples) and feeds it into the interpolator. The output of the timing error estimator is $k_T\hat{D}_E$ where $\hat{D}_E$ is the estimated timing phase error and k_T is a constant. The loop, consisting of the timing error estimator, logic, and loop filter, attempts to drive the timing phase of the derived signal in the loop D_D, towards the phase of the received signal, D_R. The interpolator is an up sampler.

4.4 Adaptive Equalizer

The adaptive equalizer compensates for the amplitude and phase distortions encountered in the channel. Figure 18 shows an adaptive equalizer block diagram.

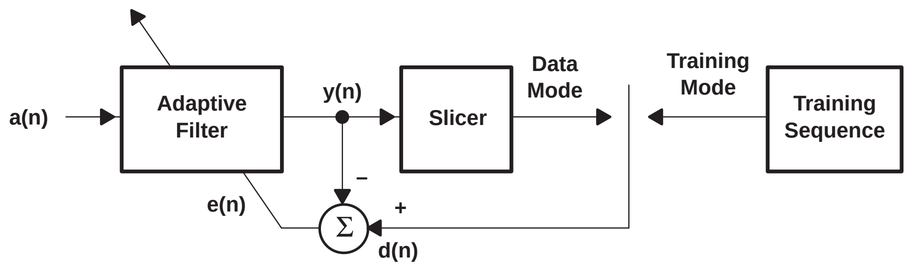


Figure 18. Adaptive Equalizer Block Diagram

An adaptive filter consists of two parts: a filter that performs a desired processing function and an adaptive algorithm that adjusts the coefficients of the filter to improve its performance. As shown in Figure 18, a(n) is the received signal plus channel noise, d(n) is the estimated (detected) signal if in data mode or the training signal if in a training mode, y(n) is the equalized signal used to detect (estimate) received data, and e(n) is the difference between the estimated signal and the equalized signal. The block labeled slicer is a decision device that estimates which symbol is most likely transmitted.

When the impulse response of the channel is convolved with the impulse response of the equalizer, the combined channel/equalizer system should have a one at the main sample and zero at all other sample points. This results in minimal or zero ISI. Equalizers are defined using a general equation:

$$C_{k+1} = C_k - \Delta\, \partial E/\partial C_k \tag{17}$$

where C_k are tap weights, E is the performance index that needs to be optimized, and is a function of the tap weights of the filter C_k. The vector $\partial E/\partial C_k$ indicates the direction of the adjustment needed to minimize E. E is minimized by adjusting the filter tap weights in small steps of Δ . This equation converges when $\partial E/\partial C_k$ is zero. A practical and popular index for performance (E) is the mean squared error (MSE). The error is measured as the difference between the received signal and the ideal signal. When the energy in the error signal is averaged over an interval, the average gives the MSE index.

As mentioned in the section on nonlinear precoding, equalizers minimize the distortion to a signal due to the characteristics of the channel, by compensating for the amplitude and phase distortions inflicted on a signal during its passage through a physical channel. Since channel characteristics vary over time, an *adaptive* equalizer ensures a constant transmission quality.

4.4.1 Types of Equalizers

Two configurations of an equalizer are possible using either an FIR or an infinite impulse response (IIR) filter. The FIR filter contains delay lines in the forward path only, while the IIR filter contains delay lines in the forward and feedback paths. A FIR is called a nonrecursive or transversal equalizer.

Modems using QAM modulation deal with real and imaginary components. The components necessitate the use of complex arithmetic in the design of equalizers.

Another type of equalizer is the fractionally spaced equalizer, where the delay elements are spaced closer than the symbol rate. If there are two inputs to the delay line, one output is computed; the input samples are shifted in at twice the sampling frequency and the output is computed once every sampling interval. This method increases the effective bandwidth of the equalizer.

In the FIR based LMS update equalizer, the coefficients of the filter are gradually adjusted until the mean square error between the equalized signal and a stored reference is at a minimum. Adjustment of the filter tap weights occurs during a training phase. The filter coefficients update using the following equation:

$$C_n(k+1) = C_n(k) - \propto e_k r_{n-k} \tag{18}$$

where $\propto$ is the adaptation step size, e_k is the error at time k, and r_{n-k} is the received signal sampled at time (n–k). The second term in the equation is a correction term. This term is the current input sample multiplied by the current error value scaled by the adaptation rate, $\propto$.

Figure 19 shows an LMS adaptive equalizer flowchart used in V.32 modems.[1]

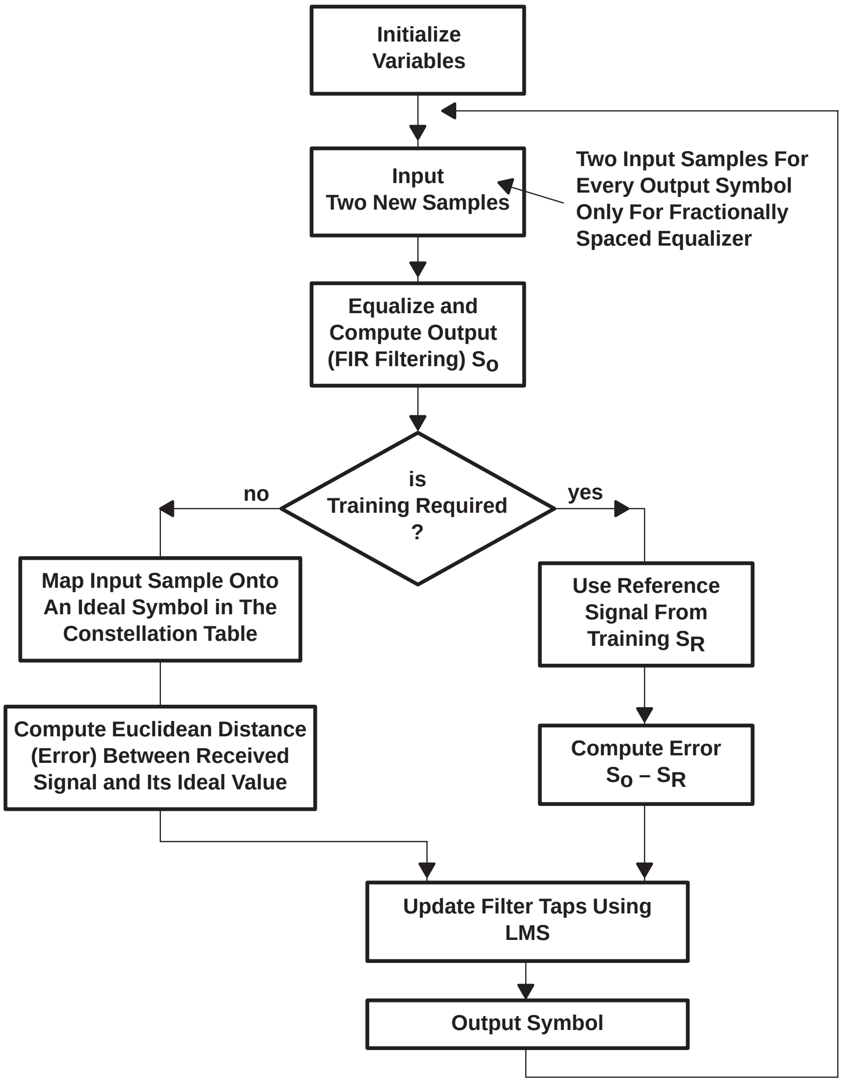


Figure 19. Adaptive Equalizer Flowchart

4.4.2 Training

When the equalizer filter resets, the taps of the filter are in an arbitrary state; some programmers initialize the taps to zero. The filters are far from the optimum state. If a filter is used in this condition, the receiver decisions (as in a decision making block like a slicer or a predictor) that are based on the equalizer output, have a high probability of error; therefore, decision directed adaptation (DDA) does not work.

A training process trains the equalizer. This process uses a training sequence known at the receiver. For example, the V.34 standard specifies that training occurs during phase 3 of the handshake using a signal PP (article 10.1.3.6 of V.34). First, the training signal is sent by the remote modem to train the local modem receiver. Some time later, the local modem repeats the procedure, training the receiver of the remote modem. During training, the difference of the equalizer output and the ideal values from the training sequence form the error that adapts the filter.

Once training is complete, any channel characteristic variations are tracked using DDA.

4.4.3 Decision Directed Adaptation (DDA)

During DDA, the decision making block (slicer or predictor) attempts to map the sample from the FIR filter onto a constellation table containing the ideal symbol. The mapping attempts to quantize the sample to the nearest ideal constellation point. In the process, the mapping generates an error. This error is the Euclidean distance between the received signal and the ideal value. The difference represents the complex error signal. The estimated error is scaled by the adaptation step size, $\propto$. The coefficients of the equalizer filter are adapted to reduce the error.

4.4.4 Decision Feedback

V.34 uses an equalization technique called decision feedback. In Figure 20, if the current transmitted symbol is estimated, the ISI contribution of that symbol is removed from any future received symbol.[5] This means that once the decision device quantifies the incoming sample to the closest constellation point, the filter can calculate the ISI effect on the subsequent received symbols and can attempt to compensate the input of the decision device.

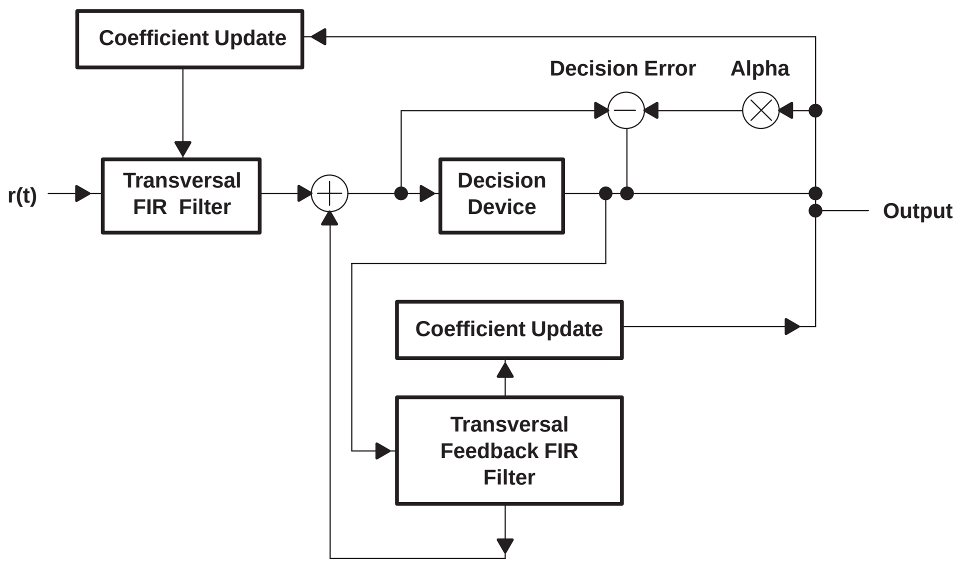


Figure 20. Decision Feedback Equalization Block Diagram

The decision device output feeds into a transversal filter, the output of which compensates the decision device. The second transversal filter in the feedback path is responsible for canceling the ISI on future symbols due to the current symbol.

V.34 splits the operation between the transmitter and the receiver. The V.34 answering modem calculates the optimum equalizer coefficients as the modem does for a normal DFE. The answering modem relays the information to the calling modem during phase 4 of the handshake, where the transmitted signal is precoded (equalized) before transmission. The nonlinear precoding in the transmitter of the data pump is a unique way of using adaptive equalization. (Refer to article 9.6.2 of the V.34 specification for the recommended procedure using the equalizer coefficients and the three recent precoded symbols). Table 21/V.34 defines the bits that form part of the MP Sequence (Type 1). The precoding coefficients are carried as part of the MP sequence.

The ITU specifies fast training of the equalizer. Fast training uses the signal PP (V.34 section 10.3.1.6). The answering modem starts the process of training the calling modem equalizer by using the signal PP. Two complex signals S and S~, transmitted in that order, precede the signal PP from the answering modem. The receiver at the calling modem detects this reversal in phase and uses it as a basis for all subsequent measurements. Further refinement of the equalizer training at the calling modem occurs using the TRN signal.

4.5 Viterbi Decoding

The output of the demodulator and the equalizer is a sequence of noise corrupted points. The Viterbi decoder attempts to map these points to their ideal constellation points; simply, the decoder determines the *most likely sequence* of transmitted points.

Viterbi decoding is made possible from the usage of the Trellis encoder in the transmitter to create a Trellis sequence. The following paragraphs explain the process of Viterbi decoding for V.32 modems. This process also applies to V.34 modems.

For a V.32 modem, the output of the Trellis coder consists of Y0 Y1 Y2 Q3 Q4 where Q3 and Q4 are the original un-encoded bits and Y1 and Y2 are the output of the differential encoder. Y0 is the redundant bit used for error-correction and is generated from the convolutional encoding of Y1 and Y2 (refer to section 3.8, Trellis Coding). Y0 Y1 Y2 split the set of thirty-two points into eight subsets, each with four possible states; therefore Y0 Y1 Y2 identify the subset and Q3 Q4 identify the point within the subset.

S0, S1, S2 are delay states and Y0, Y1, Y2 are path states (see Figure 23). For a given set of delay states (S0, S1, S2), all path states are not possible. From delay state 000, only four path states 000, 010, 011, and 001 are possible. At any given instant of time, only one delay state defines the state of the encoder. This delay state has four path states leading to the next four delay states. Out of the eight original delay states, only four are possible at the next instant of time. For example, if path state 010 is taken, the path leads from delay state 000 to 010, delay state 001 to 000, delay state 010 to 110, and delay state 011 to 100.

Examining this concept in another way, the Trellis encoder for V.32 contains three delay elements and all three are initialized to zero. A two-bit input is given to this encoder every time period. The delay element contents within the encoder change correspondingly and a three-bit output is generated. If each delay element uses one bit, then the states of the delay elements are represented within the encoder using a three-bit value. Based on this sequence, Table 1 is generated for the convolutional encoder:

Table 1. State Table for a Convolutional Encoder

Beginning State	Input	Output	End State
000	00	000	000
000	01	101	011
000	10	010	010
000	11	111	001
001	00	000	100
001	01	101	101
001	10	110	111
001	11	011	110
010	00	100	001
.	.	.	.
.	.	.	.
.	.	.	.

The table continues for all eight states, from 000 to 111. Each of the eight combinations, 000, 001, 010 until 111 is a delay state. For each delay state, the input (Y1 Y2) is represent by four values. These values cause the three delay elements to go to the End State in the table and produce an output from the encoder. Each of the values listed in the Output column is a path state; the path state corresponds to the path taken by a delay state to traverse from the Beginning State to the End State for any given input.

Each node in the Trellis diagram represents a delay state. A line, drawn from a delay state in one time window to a delay state in the next time window, is based on the two-bit input to the Trellis encoder. The three-bit value along the path taken is the path state.

Since the Trellis coder, namely the internal delay elements, is initialized to zeroes, the Trellis chart starts at 000. Delay state 000 has four possible combinations using the two-bit input and hence, there are four lines coming out of delay state 000 at time frame 1 to corresponding delay states at time frame 2. As an example, an input of 10 causes the delay state 000 to transition to delay state 010 with an output (path state) of 000.

4.5.1 The Viterbi Algorithm With the Trellis Chart

The two types of decoding schemes are hard-decision and soft-decision. The hard decision scheme takes the received signal and assigns the signal a code word, so that the received word and the code word differ from each other in the smallest number of positions. Simply, this scheme chooses a code word at the minimum distance from the received word/sequence.[6]

The basis of the Viterbi algorithm is a soft-decision scheme that uses the past history and a cost function to assign or decode the received sequence. The cost function selected is based on the distance between the received sequence as it is mapped on the constellation chart and an ideal point on the same chart.

Given a five-bit symbol for V.32 namely Y0 Y1 Y2 Q3 Q4, there are eight possible combinations of Y0 Y1 Y2 and four possible combinations of Q3 Q4. Each possible combination of Y0 Y1 Y2 defines a set of four points (Q3 Q4) arranged so that the points are symmetrical and equally spaced on the constellation.

At the beginning of each sample interval, the received signal is compared with each of the eight combinations of Y0 Y1 Y2. For example, assume that the received signal at time frame 1 is mapped into the V.32 constellation at coordinate 2, 2 (see Figure 21). This mapping does not represent a valid five-bit code because the value of the symbol could have been corrupted during its travel in the channel.

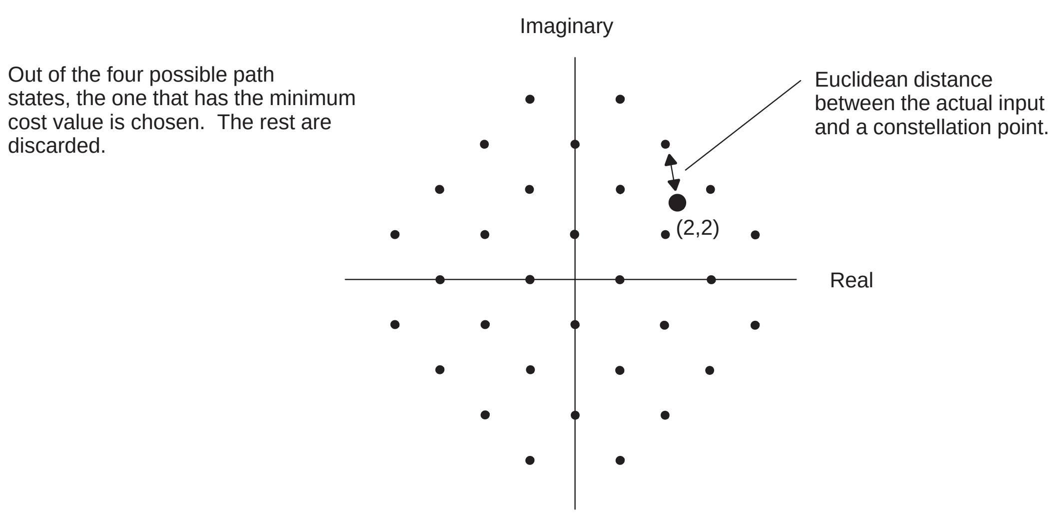


Figure 21. Received Signal at Time Frame 1

The process starts with path state, Y0 Y1 Y2, equal to 000. This path state has four possible points depending on the values of Q3 Q4. These points are 0000, 0001, 0010, 0011 (see Figure 22). The received symbol is compared with each of the four possibilities and the shortest distance computed. This shortest distance corresponds to path state 000 and is recorded in the MIN_DIST table (see Figure 25). This minimum distance computation is the Euclidean distance.

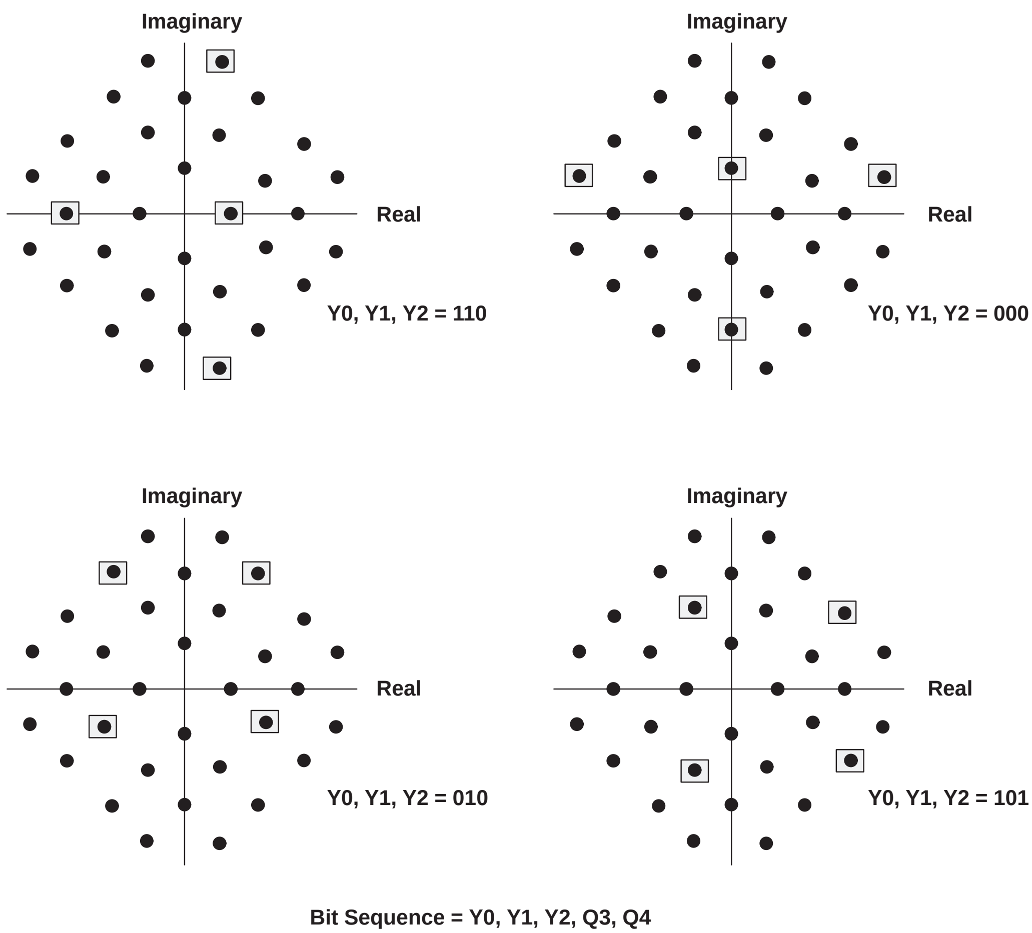
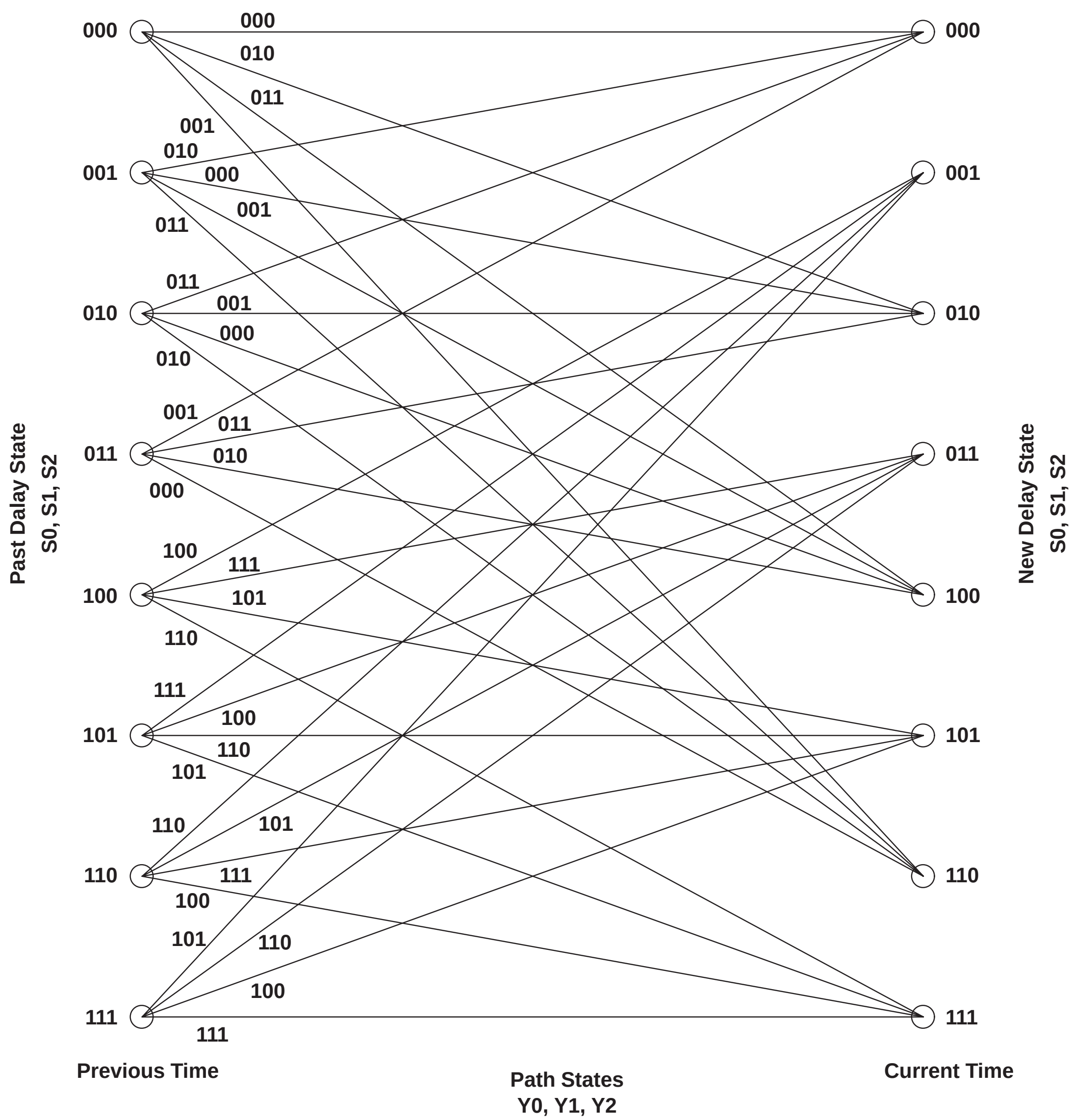


Figure 22. V.32 Modem - Signal Element Mapping

The distance between a received point at co-ordinate (x, y) and any given point on the constellation (x_k, y_k) where k = [0,..., 7] is

$$Dist\ [k]\ =\ \left[\left(x - x_k\right)^2\ +\ \left(y - y_k\right)^2\right]^{1/2} \tag{19}$$

Eight comparisons (for the eight path states from 000 to 111) produce eight constellation points. Each point belongs to a different subset of Y0 Y1 Y2 and each point satisfies the minimum distance criterion from the received sequence.



NOTE: Finite state diagram for the convolution encoder showing the relationship between delay and path States. Not all delay states can be reached from a previous delay state.

Figure 23. V.32 Modem Trellis Diagram

The next step is to compute the accumulated distance for each delay state at the current time. At every time interval, there are eight delay states S0 S1 S2. Every delay state in the current time interval is connected through four possible paths to the previous delay states. These are the only four possible paths for this delay state. Figure 23 shows the delay state 000 in the current time interval is reached through the four paths from the previous time interval:

- Path 000 from delay state 000
- Path 010 from delay state 001
- Path 011 from delay state 010
- Path 001 from delay state 011

The shortest path is chosen and the others are discarded. The chosen path state link is added to the accumulated cost of the delay state from which it originates. If the path state 010 is the shortest distance, the path state 010 is chosen and the other path states are discarded. Also, path state 010 originates from delay state 001, therefore the distance of path state 010 is added to the accumulated cost for delay state 001. This addition gives the accumulated cost for the delay state 000 in the current time interval. This cost accumulation updates in table, ACC_DIST (see Figure 25).

The process repeats for all the delay states in the current time interval (000 to 111). The accumulated cost for all the delay states in the current time interval is updated in the ACC_DIST (see Figure 25) table. Figure 24 shows the eight running paths.

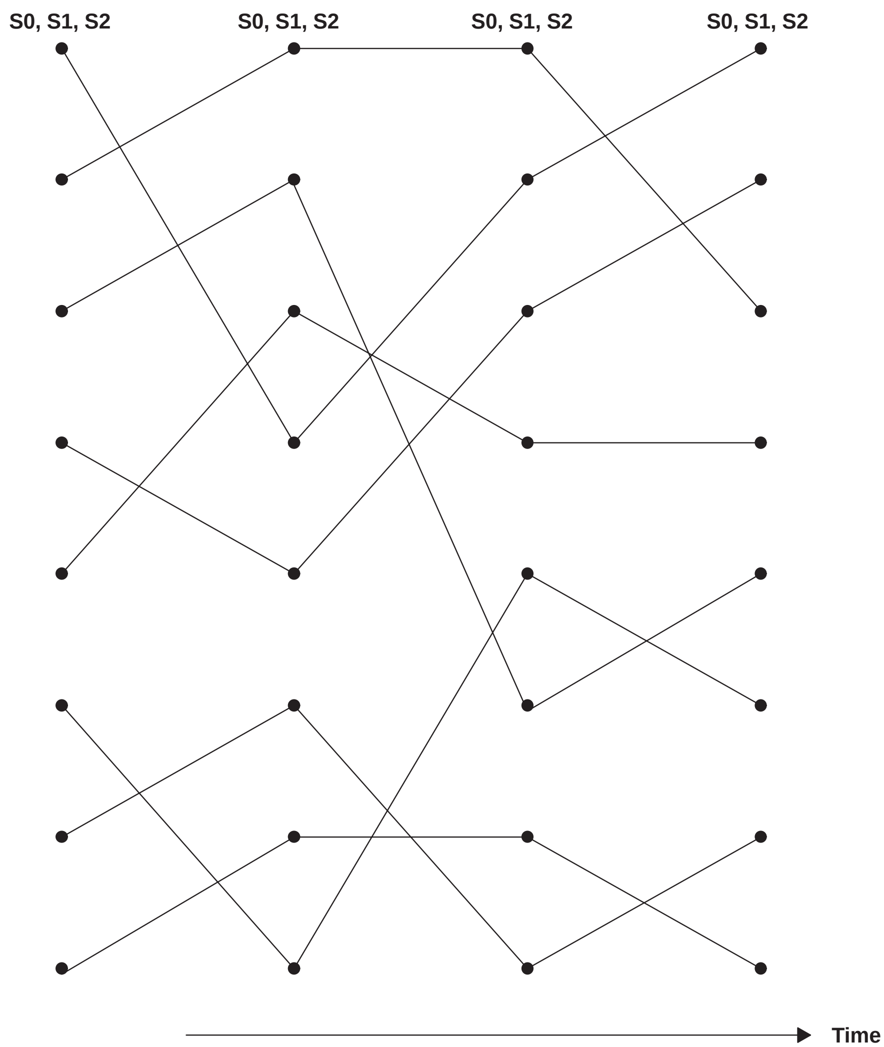


Figure 24. Eight Running Path Histories

Along with the accumulated distance for each path, two other parameters are maintained:

- The path state that is associated with the selected path
- The delay state of the previous time interval that is linked to the current delay state.

This maintenance of the path history allows the algorithm to return to the start of the Trellis diagram.

Out of the eight running paths, the path with the shortest accumulated distance (namely minimum cost) is selected; the minimum of the eight entries in the ACC_DIST (see Figure 25) table is chosen. If this path is associated with delay state 000 at the current time interval, the path state terminating at this delay state 000 is traced back for sixteen time intervals (or to the beginning of the path history).

4.5.1.1 Sixteen Time Intervals

The length of the sixteen time intervals signifies the decoder path history. The greater the path history, the greater the chance of retrieving the original symbol. Increasing the path history results in an increase in the amount of required memory, the number of computations, and slows the process. The optimal length of the decoder is four or five times the constraint length of the convolutional encoder. The constraint length is the number of bits required to define the delay states at each time window, namely S0 S1 S2. Here, the constraint length is three so the path history chosen is sixteen. The choice of the time window is the discretion of the programmer.

After tracing back to the beginning of the Trellis chart, the path state (Y0 Y1 Y2) associated with the last link (namely, at the origin of the Trellis chart) stored in memory is the result of the Viterbi algorithm. This path state corresponds to four possible points on the constellation chart (as determined by Q3 Q4). Of these four points, the one that is closest to the actual input in that time period is selected. This selection gives the five-bit value Y0 Y1 Y2 Q3 Q4.

The redundant bit, Y0, served its purpose and is discarded. Bits Y1 and Y2 are differentially decoded to obtain Q1 and Q2:

$$Q1_n = Y1_n \oplus Y1_{n-1} \tag{20}$$

$$Q2_n = (Q1_n \cdot Y1_{n-1}) \oplus Y2_{n-1} \oplus Y2_n \tag{21}$$

4.5.1.2 Backtracking and the Relevance to the Transmitted Data

The convolutional encoder for V.32 contains two input bits Y1 and Y2, and a three bit shift register represented by S0 S1 S2. The contents of the shift register shift by one bit. If the delay state moved from delay state 000 to delay state 001 for the first input, the next input causes the delay state to move from 001 to say 011. Every input is a set of four bits, since the V.32 encoder works on four bits at a time to generate five bits. Each input is related to the next input, ie, the next state of four bits. Before the actual decoding process starts, the entire decode buffer is filled with known paths. When the process starts for the first time, paths originate from delay state 000.

In this example, the V.32 decoding scheme uses a time frame of sixteen windows. If the input sequence, A, is received, the decoded result is not available until sixteen time windows later. The Viterbi decoding scheme relies on a past history and the fact that one input sequence is correlated to the next sequence due to the encoding scheme at the transmitter.

4.5.2 Viterbi Decoding on a TI DSP

Figure 25 shows the flowchart of a Viterbi decoding scheme.[6] Soft-decision encoding and decoding techniques use looped code, conditional executions, minimum-maximum searches, and pointer addressing techniques.

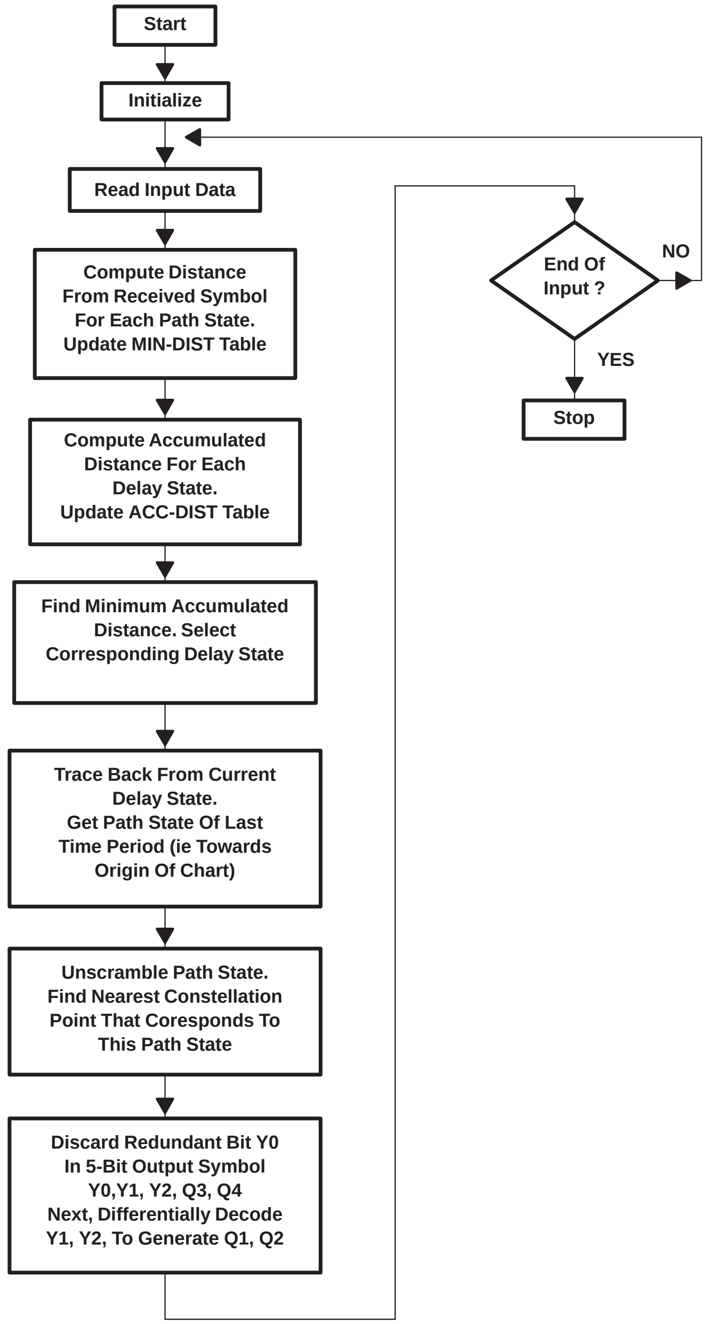


Figure 25. A V.32 Viterbi Decoding Flowchart

4.5.2.1 Viterbi Decoding Using the TMS320C5x DSP

The TI TMS320C5x family of DSPs enable Viterbi decoding using circular buffers, zero-overhead looping, delayed jumps and calls to minimize code execution time, min-max instructions to use efficient search algorithms, and post-modified indirect addressing techniques. The indirect addressing techniques include indexed, circular, and bit reversed addressing modes. A V.32 benchmark is:[6]

Table 2. Viterbi V.32 Benchmark

	Code Size (Words)	Data Size (Words)	CPU Loading Per Symbol, Excluding Initialization (n Machine Cycles)
V.32 Encoder	79	10	90
V.32 Decoder	768	837	963–973

4.5.2.2 Viterbi Decoding Using a TMS320C54x DSP

The TMS320C54x family contains a compare, select, and store unit (CSSU).[7] The CSSU is an application specific unit dedicated to add/compare/select operations of the Viterbi algorithm. In this specific V.32 application, the first calculation is the distance of the received symbol to all the combinations of the eight path states. The shortest path from each path state is stored in a table.

Next, the accumulated distance is calculated for each delay state and the minimum cost link is chosen to return to the beginning of the Trellis chart. This operation uses the CMPS instruction in the CSSU unit. The CMPS instruction uses the CSSU unit to compare two sixteen-bit parts of the specified accumulator and shifts the decision into TRN (bit 0). This decision is also stored in the TC bit of ST0. Based on the decision of the compare operation (between the upper 16 bits and the lower 16 bits of the specified accumulator), the higher quantity is stored in the specified data memory location. TRN contains the information of the path transition decisions to the new state. This information can be used during the back tracking routine.

4.5.3 Inverse Precoding

The output of the Viterbi decoder corresponds to the estimated Trellis coding of the signal at the transmitter. Since the signal is precoded at the transmitter before Trellis coding, the Trellis sequence at the transmitter is not the same as the one selected by the mapping unit at the transmitter. For this reason, inverse precoding is applied at the receiver.

Figure 26 shows a block diagram of the section between the precoder at the transmitter and the inverse precoder at the receiver.[4]

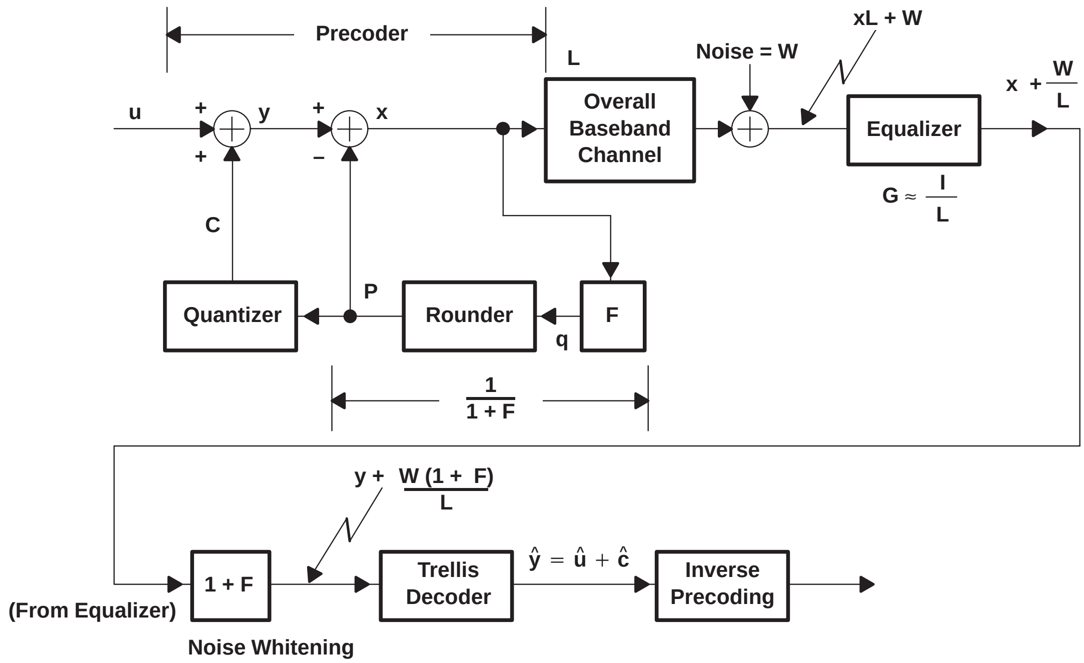


Figure 26. Block Diagram of Precoding and Inverse Precoding

V.34 modems use DFE so that the function is split between the transmitter and the receiver. This method is chosen rather than using DFE in the receiver for the following reasons:

- DFE performs well only if all the decisions are correct
- Trellis coding and decoding introduces a delay; the longer the delay, the more accurate the decisions.

If the ISI due to the most recent decisions needs to be canceled, delay-free decisions are needed. Therefore, the function of the DFE is split so that the learning occurs at the receiver during training and all the learned coefficients are transferred to the transmitter.

In Figure 26, G is the transfer function of the equalizer and W is the channel noise. The channel noise is assumed to be white. The output of the precoder, x, appears as xL + W at the input to the equalizer. Figure 26 also shows the other signals.

If there are no errors in the Trellis decoder, then the estimate of the Trellis decoder output, $\hat{y}$, is:

$$\hat{y} = y = u + c \tag{22}$$

At the input to the Trellis decoder, the signal component without noise is y. The noise component is W(1+F)/L. The training of the noise whitening filter produces 1+F ≈ L. Hence, the Gaussian noise at the input to the Trellis decoder is mainly white.

If the input to the inverse precoder (see Figure 26) is $\hat{y} = y$, the output of the inverse precoder should be the same as the input to the precoder u.

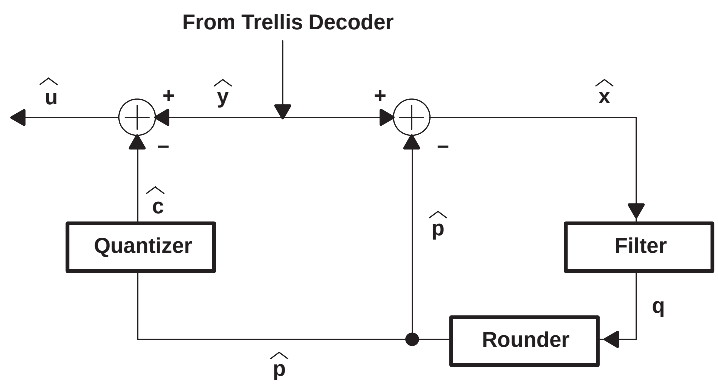


Figure 27. Inverse Precoder

The input to the inverse precoder comes from the Trellis decoder. The steps are:[4]

1. Compute the filter output estimate $\hat{q}(n)$.
   $\hat{q}(n) = f(1)\hat{x}(n-1) + f(2)\hat{x}(n-2) + f(3)\hat{x}(n-3)$ where $\hat{x}$'s are the three previous symbols.
2. Round $\hat{q}(n)$ to get $\hat{p}(n)$ as in the precoder. Save the $\hat{p}(n)$.
3. Compute $\hat{x}(n) = \hat{y}(n) - \hat{p}(n)$ and shift $\hat{x}(n)$ in the filter.
4. Quantitize $\hat{p}(n)$ to obtain $\hat{c}(n)$ as in the precoder.
5. Compute $\hat{u}(n) = \hat{y}(n) - \hat{c}(n)$.

The quantizer and rounder are identical to those in the transmitter.

The complex filter coefficients are learned by the receiver during phase 4 of the training using TRN. These coefficients are then sent to the transmitter.

With the inverse precoder in the system and the noise whitening filter, the goal is:

- G(f) ≈ 1/L(f)
- 1 + F(f) ≈ L(f)

The signal, y, goes through 1/(1+F) at the precoder, L through the channel, G in the equalizer, and (1+F) in the noise whitener, giving [1/(1+F)](LG)(1+F) ≈ 1 at the Trellis decoder. The channel noise, W, goes through G(1+F) ≈ (1+F)/L in arriving at the Trellis decoder. Since (1+F) ≈ L, the noise is coarsely white.

In summary, the precoder and the inverse precoder are the inverse of each other. If the scheme provides $\hat{y}$ approximately equal to y, the scheme also provides a satisfactory overall operation, and $\hat{U} = U$, with a high probability. Precoding and the equalizer scheme correct a wider range and variety of channel characteristics than a linear equalizer by itself. This correction allows the use of a higher symbol rate.

4.6 Inverse Mapping

The V.34 specification does not provide algorithms for shell demapping. The algorithms are specific to individual modem vendors.

The goal of the inverse mapper is to retrieve the exact sequence of input bits S, Q, and I bits, from one mapping frame. Inverting the operation of the differential encoder, the shell mapper, the parser, and the scrambler achieves this goal.

The I bits (I1, I2, I3) are retrieved by reversing the equations of the differential encoder. In the process, the output of the Trellis encoder at the transmitter (U0) is also obtained. The three I bits are used at the transmitter to rotate the four pairs of two-dimensional points in each mapping frame.

The Q bits are determined by using an inverse constellation lookup table. The x and y coordinates of the constellation points are an index into this table. The entries in the table correspond to the point indices. Masking the lower q bits of the index gives the q-Q bits. The remaining high order bits, that constitute the ring index of the point, are stored in a memory array location until a block of eight ring indices is obtained. When eight ring indices are restored, the end of a mapping frame is reached. The next step is to obtain the final K-S bits of the sequence.

The shell demapper reverses the process of shell mapping to create the index of the received eight ring sequence.

After decoding the S, Q, and I bits, the inverse mapper reverses the operation of the parser to combine these bits into one final array. The array is unscrambled. The descrambler uses the same polynomial as the transmitter, accomplishing the reverse operation (see Figure 28).

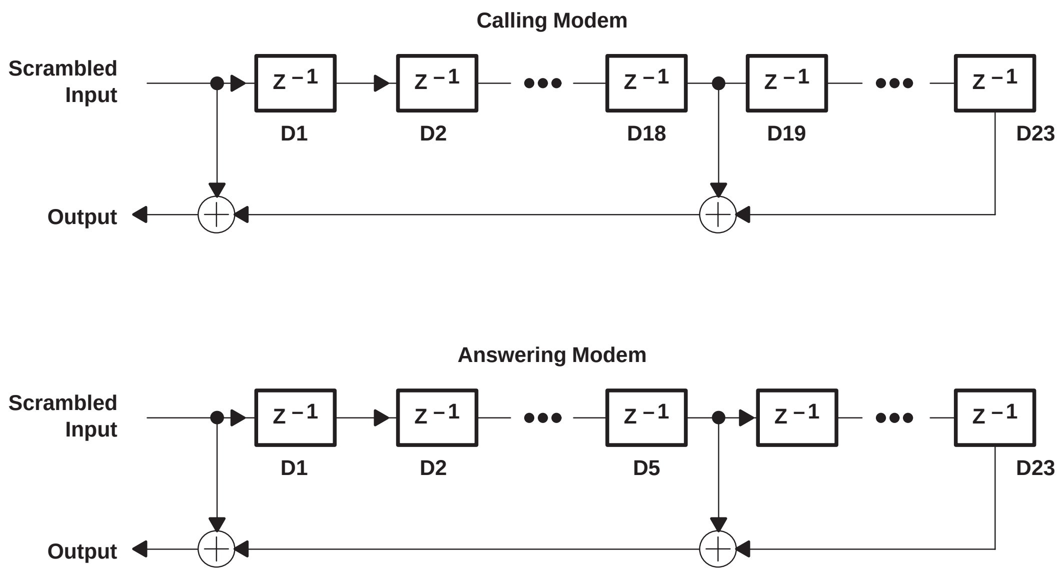


Figure 28. Calling and Answering Modem Descramblers

5 Modems - A System Level Perspective

Modems are internal or external. An internal modem is inside a PC, communicating through the ISA bus. An external modem is outside a PC, communicating through the RS-232 interface. Now, modems contain a Plug-n'Play interface that allows automatic detection of the COM port. The user does not set the DIP switches to select a COM port and IRQ number.

Figure 29 shows the essential components of a modem. These components are:

- A DSP to handle signal-processing functions
- An AFE. The AFE communicates with the phone line circuit on one end and the DSP at the other end. The AFE provides for a microphone input and speaker output. This I/O allows for multimedia applications.
- A data access arrangement (DAA) circuit. The DAA is the phone line circuit responsible for interfacing to the phone line. The DAA contains protective components which play an important role during FCC 15, FCC 68, and UL tests. The other components are a 600 Ω termination, a transformer, and a current holding circuit. This circuit provides a path for the DC current from the phone lines. The circuit contains ring-detect circuitry to detect incoming rings and caller-ID circuitry to provide a path for caller-ID data. Caller-ID data is sent between the first and the second rings.
- The static RAM that holds static data. V.34 modems use a 64K word SRAM to store data tables and time critical program code. The speed of a SRAM depends on the access time of a DSP peripheral. For example, a TMS320C51 based modem operating at 57 MHz (cycle time of 35 ns), needs a peripheral (SRAM) access time of 20 ns.
- The ROM that contains the DSP program code. Often, time critical program code is downloaded into the SRAM, since the ROM is slower. A ROM should satisfy the DSP access time requirement. A TMS320C51 based modem operating at 57 MHz needs a ROM rated for 20 ns. Such speeds do not exist. If such a ROM exists, its cost is prohibitive. To handle the speed requirements, DSPs contain a wait-state generator that is programmed during power up. Each wait state prolongs the read and write cycles by one machine cycle (35 ns). For the TMS320C51 in the example, one wait state prolongs the read and write cycle by one machine cycle (35 ns). The number of wait-states should be chosen to satisfy the access speed criteria. Using one wait-state requires a ROM with a speed of 55 ns (= 20 ns access time requirement for the TMS320C51 + 35 ns). The designer chooses the wait state of the modem since faster memories tend to be expensive or in short supply. The choice of wait states also depends on the performance needed of

the overall system, since additional wait-states slow the entire system. V.34 modems contain a ROM size of 128K words.

- An EEPROM (nonvolatile memory) holds data relevant to Plug-n'Play and modem profiles.
- An ISA bus interface (for internal or parallel modems) or an RS-232 interface (for external or serial modems). Internal modems contain an ASIC that contains the ISA interface logic, Plug-n'Play logic, UART (16550) logic, and DMA logic (for voice playback/recording in multimedia modems).

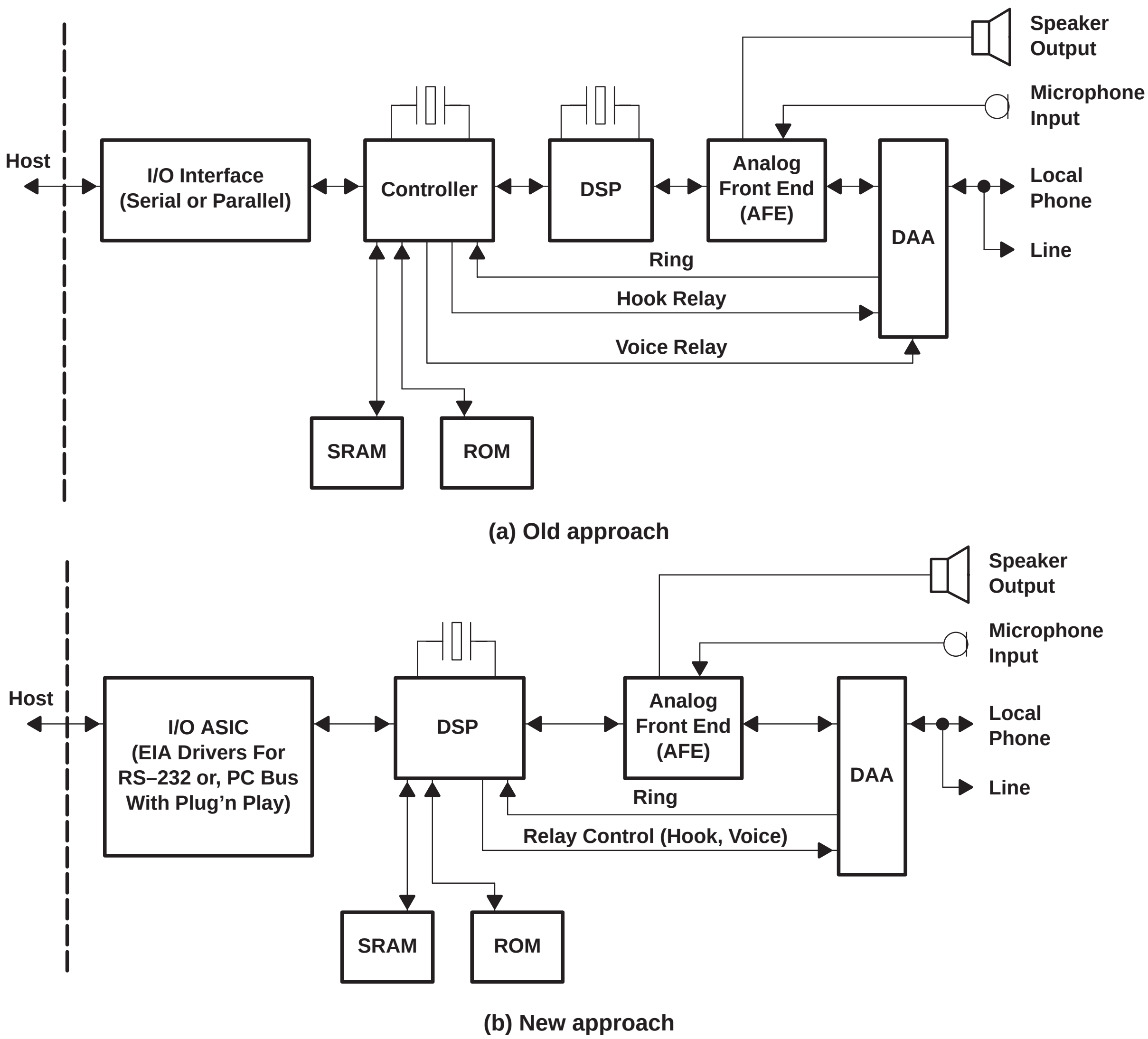


Figure 29. Modem Block Diagram

Some modems contain a dedicated microcontroller that interfaces to the RS-232 or ISA bus on one side and the DSP on the other side. The microcontroller performs the following functions:

- AT command parsing and execution
- Call progress monitoring, such as monitoring for the presence of dial, busy, answer, and facsimile calling tone. The microcontroller does not perform tone detection; detecting tones is a DSP function. The DSP reports the tone detection results to the microcontroller, that takes suitable action.
- Handshake monitoring. The DSP performs the signal processing functions that form handshaking; the microcontroller monitors the progress of handshake and makes decisions on moving to the next or previous state of a state machine.
- Data link layer protocols (V.42 error correction and V.42bis compression)
- Sending or receiving data. The microcontroller retrieves data from the PC and sends the data to the DSP. The DSP passes the data to the AFE which in turn passes the data to the DAA and then to the phone line. Received data from the phone line follows a similar route in the reverse order.

Most V.34 modems do not use a microcontroller. All the microcontroller functions are now performed in a DSP. Although DSPs are inefficient in performing non-signal processing functions, removing the microcontroller from the modem allows vendors to keep system costs and chip counts low.

By default, a modem powers up in data mode or AT+FCLASS = 0 mode. When a user dials a number, the modem goes off-hook and looks for a dial tone. The DSP tone detectors qualify the presence of a valid dial tone (see Table 3) before the digits are dialed by the DTMF generator in the DSP.

Table 3. Network Call Progress Tones[8]

Tone	Frequency (Hz)	On Time (sec)	Off Time (sec)
Dial	350 + 440	Continuous	
Busy	480 + 620	0.5	0.5
Ringback, Normal	440 + 480	2	4
Ringback, PBX	440 + 480	1	3
Congestion (Toll)	480 + 620	0.2	0.3
Reorder (Local)	480 + 620	0.3	0.2
Receiver Off–Hook*	1400 + 2060 + 2450 + 2600	0.1	0.1
No such number	200 to 400	Continuous, Frequency Modulated at 1 Hz Rate	
Fax Calling Tone	1100	0.5	3

☆Receiver off–hook is a very loud tone, 0 dBm per frequency

After the digits are dialed, the calling modem goes into a call-progress monitor mode. During this time the modem listens for: a busy, answer, facsimile, voice tones, etc. The remote modem is not allowed to send an answer or a calling (CNG) tone immediately as per the central office (CO) requirement. This requirement stipulates that the answering equipment maintains a silence for two seconds (the billing delay). The silence gives the CO sufficient time to start billing the caller.

5.1 The Handshake Process

This section presents a brief overview of the V.34 handshake process. For a detailed description, refer to the ITU recommendation for the modem standard.

5.1.1 Phase 1

The remote modem sends a 2100 Hz answer tone. The calling modem DSP qualifies the presence of the answer tone for a period of 1 second before detecting subsequent tones that apply to the particular mode of operation. The handshaking tones/sequences used for V.22bis (2400 bps) differ from those tones used for V.32 (9600bps) that differ from those tones used for V.32bis (14400 bps).

Phase 1 is markedly different for V.34 as compared to the older V series standards. This phase is also called network interaction. The calling modem DSP, namely the receiver in the datapump, is conditioned to receive either an answer tone or the tones compliant with V.8.

Modem standards prior to the V.34 standard stipulated that the answering modem send an answer tone. If an answer tone is detected, the handshake proceeds based on the recommendations set by V.32bis Annex A, T.30 Facsimile Recommendation, or any suitable recommendation. The detection of V.8 tones by the calling modem, is an indication that the remote (answering) modem is also a V.34 modem.

V.8 is an intelligent procedure. V.34 modems use the V.8 to perform feature and mode negotiation quickly, utilizing V.21 (300 bps FSK) modulation to exchange information. Negotiation parameters include information such as:

- Data mode or text phone operations
- Modulation modes available
- V.42 and V.42bis support
- Wireline or cellular operation

The modem enters phase 2 based on parameters learned during V.8 of phase 1 or based on the detection of an answer tone signifying a non-V.34 mode.

5.1.2 Phase 2 - Probing/Ranging

The calling and answering modems proceed with the exchange of tone configurations A, A~, B, B~. Tone A is a 2400 Hz tone transmitted by the answering modem, whereas tone A~ is the tone obtained by a 180 degree phase reversal of tone A. Tone B is a 1200 Hz tone transmitted by the calling modem and tone B~ is obtained by a 180 degree phase reversal of tone B. The four tones are used by the modems in the estimation of the round-trip delay. Phase 3 uses the round-trip delay estimate for the fast training of echo canceller and equalizer. The modems use two probing signals, L1 and L2, to analyze the characteristics of the telephone channel.

5.1.3 Phase 3 - Equalizer and Echo Canceller Training

The modems decide on a symbol rate, carrier frequency, pre-emphasis filter, and power level by phase 3. The following signals/tones are exchanged:[3]

- S, S~: A transition from S to S~ is the basis for detection of signal PP
- PP: consists of six periods of a forty-eight symbol sequence. The remote modem uses PP to train the equalizer of the other modem receiver.
- MD: an optional manufacturer-defined signal used for training
- TRN: a sequence of symbols generated by applying binary one to the input of the V.34 scrambler. TRN is used for fine training of the equalizers following the use of signal PP.
- J: repetitions of a specified bit pattern indicating the constellation size used by the remote modem for transmitting sequences TRN, MP, MP’, and E during phase 4 of the handshake

- J’: J’ terminates the sequence J.

This phase is MIPS intensive for the DSP, during which the DSP detects incoming tones or signals.

5.1.4 Phase 4 - Final Training

Phase 4 is the final training of the modem and exchange of final data modulation parameters. Phase 4 uses the following signals/tones:[3]

- S, S~
- TRN
- J, J’
- MP: sequences contain MP for data mode transmission. Bit fields for the MP sequence, types 0 and 1, are in the V.34 recommendation
- MP’: an MP sequence with the acknowledge bit set to 1.
- E: a 20-bit sequence of binary ones signaling the end of the MP field.
- B1: this sequence consists of one data frame of scrambled ones at the end of phase 4 using the selected data mode MP. Prior to the transmission of B1, the scrambler, Trellis encoder, differential encoder, and the precoding filter tap delay line are initialized to zero.

At the end of phase 4, if both modems support V.34, a V.34 connection is made. The connection speed is dependent upon line conditions.

At this time, the DCEs are connected, ie. the modem DSPs are connected. The DTEs are still not connected.

5.1.5 Link Layer Protocols

The bottom layer of the seven-layer OSI model is the physical layer. The modems achieve a connection at best-case line speed, which for V.34 is 28,800 bps. There is an emerging standard, an Annex to V.34, that attains a speed of 33,600 bps. The connection is at the physical layer at both ends.

All modems use one of two error correcting protocols for errors in the received data. Such protocols are data link layer protocols. The data link layer is the second layer in the OSI structure.

The first error correction protocol is based on the family of Microcom networking protocol (MNP) standards. The second error correction protocol is a link access procedure for modems (LAPM). The protocols are specified by the ITU as part of the V.42 standard. The LAPM protocol is the primary method while MNP4 is the alternative method. Alternative does not mean optional. A modem termed V.42 *compliant* implements LAPM and MNP2 through MNP4 protocols. A modem termed V.42 *compatible* implements the LAPM protocol. Unless specified by the user, two V.42 modems attempt a LAPM connection first. If one or both of the modems support MNP, the modems communicate using that protocol.[9]

The ITU V.42 standard using a LAPM protocol is based on High-level Data Link Control (HDLC) formats and procedures (refer to the ITU V.42 recommendation for additional information).

Once a V.42 connection is established, the modems attempt a data compression scheme using V.42bis. If the error correction protocol is MNP4, the modems use MNP5 for data compression. V.42bis uses a dictionary based Lempel-Ziev-Welch (LZW) algorithm. This scheme attains up to 4:1 compression depending on the type of data. Text files are easier to compress than binary files or graphic files. The MNP5 scheme uses a combination of adaptive Huffman encoding and run-length encoding (RLE) to attain a compression of 2:1 (refer to the ITU V.42bis recommendation for additional information).

6 Facsimile

At the physical level, data and facsimile modems are indistinguishable. The principal difference between the two is the communication session protocol. Facsimile modems support the protocols described by the ITU specifications T.4, T.6, and T.30.

The scenario is slightly different when a facsimile modem is used as part of a facsimile machine. The machine has a scanning system that sends an electrical signal representing the image to be transmitted to a modulation system that is part of the facsimile modem. A reverse process occurs while receiving a facsimile where the demodulation unit of the facsimile modem sends electrical signals to the replication unit that prints the received image on paper.

Depending on the signal type, the modulation method, and the communication capabilities, the CCITT provides four classifications of facsimile equipment.

Group 1 and 2 facsimile machines, which use the ITU standards T.2 and T.3 respectively, use analog signals for encoding and transmitting. Group 2 machines use a more efficient method of modulation than Group 1 machines.

All facsimile machines are classified as Group 3 using ITU standard T.4. Group 3 facsimile machines use digital techniques and view an image as a series of discrete dots or pixels. Each pixel assumes a value of 1 (black) or 0 (white). No intermediate shades are possible. The bits representing an image are compressed and transmitted. T.4 also specifies supporting a resolution of 15.4 lines per millimeter vertically and 16 pixels per millimeter horizontally.

Figure 30 shows a facsimile transmitter block diagram.

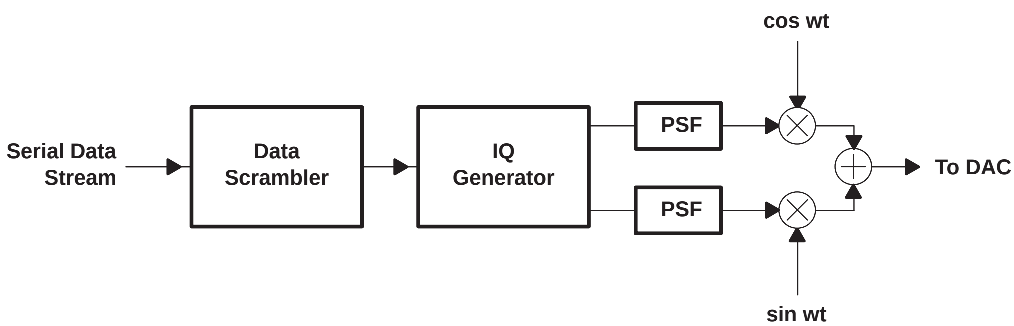


Figure 30. Facsimile Transmitter Block Diagram

Group 3 facsimile machines/modems operate at the following speeds: ITU V.27ter at speeds of 2400 or 4800 bps; ITU V.29 at speeds of 7200 or 9600 bps; ITU V.17 at speeds of 7200, 9600, 12000, and 14400 bps. The facsimile operation is half-duplex as opposed to data modems that are full-duplex.

Any facsimile transmission involves:

- Conformance to the T.4 standard. The T.4 standard specifies the guidelines to be followed for document transmission namely coding or compression schemes, resolution of the image (number of scan lines/mm and number of pixels/mm, transmission time per total coded scan line, etc).
- Conformity to the T.30 standard. The T.30 standard describes the procedures and handshake signals used when the facsimile equipment is operated over a GSTN.
- Modulation standard (V.17, V.29, etc.)

6.1 ITU V.17 Modulation Standard

ITU recommendation V.17 is a 14400 bps facsimile standard which specifies: data signaling rates of 14400, 12000, 9600, and 7200 bps; QAM at 2400 symbols/second; carrier frequency of 1800 ± 1 Hz; inclusion of data scramblers, adaptive equalizers and eight-state Trellis coding.[10]

The data scrambler uses the polynomial:

$$1 + x^{-18} + x^{-23} \tag{23}$$

At the transmitter, the scrambler divides the message sequence by the generating polynomial. At the receiver, the received data sequence is multiplied by the same polynomial to recover the message sequence. Efficient use of circular buffers in the DSP achieves scrambling/descrambling.

The scrambled data stream to be transmitted is divided into groups of six bits which are ordered according to their time of occurrence. The first two bits $Q1_n$ and $Q2_n$ are differentially encoded according to a table specified in the recommendation (Table 1/V.17) to form $Y1_n$ and $Y2_n$. The two differentially encoded bits $Y1_n$ and $Y2_n$ are used as inputs to a convolutional encoder which generates a redundant bit $Y0_n$. The six bits formed from $Y1_n$ $Y2_n$ $Q3_n$ $Q4_n$ $Q5_n$ $Q6_n$ are combined with the redundant bit $Y0_n$ and mapped into a constellation space for transmission. The use of pulse shaping filters assures a zero ISI.

Refer to the ITU V series recommendation for more details on facsimile modulation schemes.

6.2 T.4 Coding Schemes

Facsimile modems use a run-length modified Huffman encoding scheme. This scheme is one-dimensional or two-dimensional. The specifications are provided in the T.4 recommendation. The encoding scheme is irrelevant to the facsimile modem, since the encoding is performed in the DTE (ie. PC) connected to the modem (refer to the T.4 recommendation).

6.2.1 One-Dimensional Coding

The T.4 standard describes a one-dimensional run-length encoding scheme to compress data in a scan line. Each scan line is a series of variable length code words as defined in the T.4 recommendation.

6.2.2 Two-Dimensional Coding

The T.4 standard recommends an optional two-dimensional encoding scheme. After each scan line is coded one dimensionally, at most K–1 lines following this line are encoded relative to it. The maximum value of K is two for standard vertical resolution or four for a higher vertical resolution.

6.3 T.30

Figure 31 shows a T.30 facsimile call. The T.30 recommendation divides the facsimile operation into the following five phases:

- Phase A: call establishment.
- Phase B : pre-message procedure.
- Phase C : message transmission.
- Phase D: post-message procedure.
- Phase E: call release.

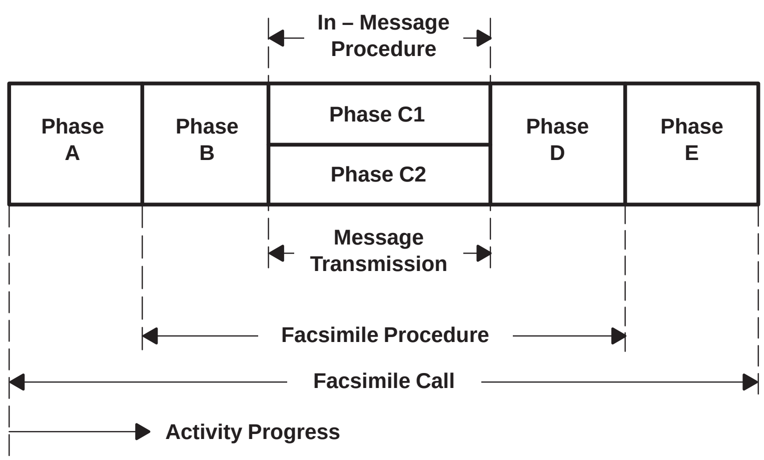


Figure 31. A T.30 Facsimile Call

In the older designs that relied on a dedicated microcontroller, the DSP detected the different tones that formed part of a facsimile process and the modulation and demodulation of facsimile signals. The microcontroller performed T.30 and communicated with the DTE. This is not applicable where a single DSP performs all the tasks previously performed by the microcontroller.

6.4 A Facsimile Session

The first phase of a facsimile session begins with the placement of a call. Facsimile equipment uses a tonal exchange or binary coded messages throughout the handshake. Binary coded message signalling is attempted first. All binary coded control procedures are transmitted using V.21 at 300 bps and an HDLC frame structure. (Refer to T.30 recommendation for further details regarding the different tones used as part of the tonal exchange and for salient differences in the handshake process using tonal sequences versus binary coded messages). Figure 32 presents a typical facsimile session.[9]

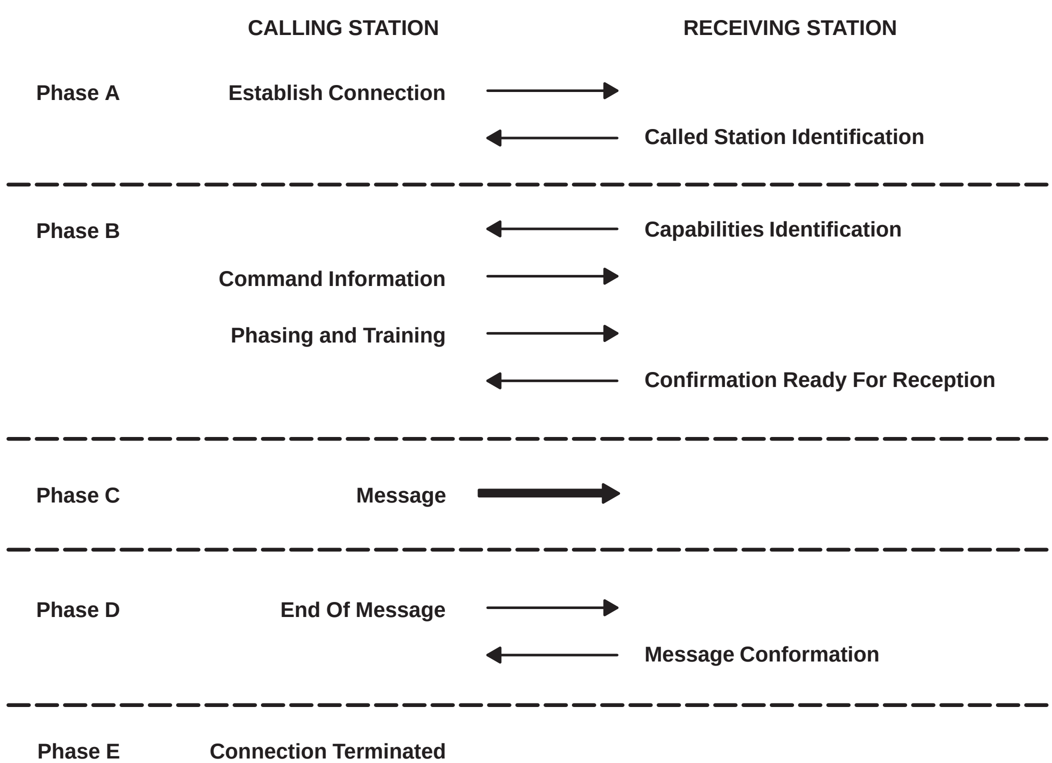


Figure 32. A Typical Facsimile Session

When a call is placed by the calling machine, the machine transmits a calling tone. The DSP at the called station detects this tone and the handshake proceeds to Phase B.

During the pre-message procedure (phase B), the two facsimile equipments negotiate, select, and confirm the parameters of the connection using the digital identification signal (DIS) and digital command signal (DCS). After a confirmation, the two sides exchange phasing and training signals and establish a synchronization for reliable communications.

Phase C is comprised of in-message procedures and message transmission. The message transmission occurs using the appropriate V. series recommendation for the equipment. In-message procedures occur along with message transmission and involve commands and responses that control synchronization, error detection, error correction, and line supervision.

The post-message procedure (phase D) involves messages regarding the transmission of additional pages, confirmation or indication of an end-of-message, or end-of-facsimile signaling.

Call release (phase E) signifies the final period when the facsimile equipment terminates the call.

6.5 EIA Class Structure

The EIA developed a series of standards that define the protocols and commands for use between a DTE and a facsimile DCE. This structure created three classes depending on the ability of the facsimile modem to conduct a facsimile session.

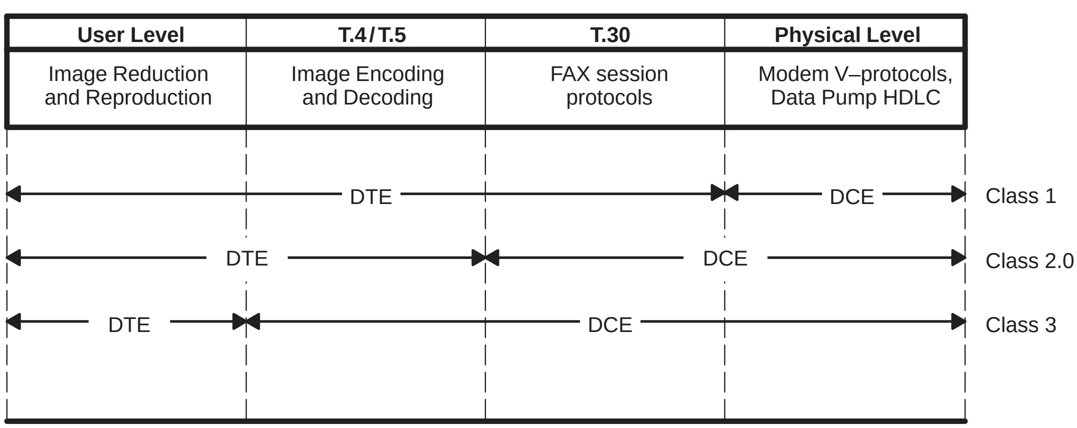


Figure 33. EIA Fax Modem Class Interface Partitioning

6.5.1 Class Definitions

The classes create a dividing line regarding the duties of the DTE and the DCE for a given facsimile session.

6.5.1.1 Class 1

Class 1 modems are defined by the EIA in the specification EIA/TIA-578. This class of modems provide the minimum services for a Group 3 facsimile session. The DTE is responsible for encoding the image per T.4 and managing the session protocols per T.30. A Class 1 modem performs the following functions:

- Provides a GSTN interface.
- Autodialing
- Modulation and demodulation per the V series recommendation.
- HDLC framing and error detection.
- Commands and responses.

The application facsimile software resident on the PC always manages the facsimile session.

Since Class 1 was the first specification for modem designers, the specification fails to address several issues needed for a full Group 3 specification. Some specification faults include data overrun/underrun, incapability of meeting facsimile timing constraints, and no error correction mode (ECM) support.

6.5.1.2 Class 2

A Class 2 modem exhibits more intelligence than a Class 1 modem. The T.30 session is off-loaded from the PC to the modem. The PC still retains the responsibility for preparing and compressing the image data for transmission.

The Class 2 specification developed by Sierra Semiconductor, was an unwritten standard. This specification suffered from many of the same problems that plagued Class 1 including overrun, underrun, and no ECM support. This standard is obsolete.

6.5.1.3 Class 2.0

The EIA interim recommendation SP-2388A was modified extensively and converted to TIA/EIA-592. The TR29.2 committee ratified this standard. Modems compliant with this specification are Class 2.0. This specification addresses many of the basic defects that existed in the Class 1 and Class 2 implementations. The following highlights some of the changes:

- The computer overrun problem is resolved by using a simple data packetizing protocol that operates on the modem-computer interface. This protocol allows the computer to detect data overrun on the receive path and allows for a retransmission of the data packet. This protocol is supported by three seconds of data buffering in the modem to compensate for the interruption in the flow of data to the computer.
- The data underrun between the computer and the modem is resolved by requiring the modem to parse the T.4 image scan lines to determine the appropriate positions where an idle state is established on the PSTN link. This feature allows the computer to underrun data to the modem for a period of approximately five seconds.
- The flow control problem is resolved by introducing a facsimile specific flow control setting command.
- Earlier modems had to autobaud at the high DTE baud rates required by facsimile equipment. The high autobaud caused difficulty for the less functional modems that could not achieve autobauding at the higher speeds. This problem is solved by introducing a specific command for baud rates.
- Error correction mode (ECM) is addressed simply by requiring the modem to implement the protocol without assistance from the computer. This requirement prevents a computer-to-modem interface change.
- The requirement for receive data quality check is specified. This data quality check calls for the decoding of the T.4 scan lines to determine any corruption in the image data. This check allows for decisively realizing the quality of the received facsimile. This measure of quality determines the T.30 response to the page reception. This feature is an inherent part of the T.30 protocol and is required by European countries.
- The T.30 polling capability allows an originating Group 3 apparatus to receive a document from an answering Group 3 apparatus.

A Class 2.0 modem performs these functions:

- Provides a GSTN interface, autodialing, modulation and demodulation per the V series recommendation.
- T.30 protocol implementation.
- Session status reporting.
- Padding for minimum scan line.
- Quality check on received data.

6.5.1.4 Class 3

Class 3 is currently under study. Class 3 continues to offload functions from the PC to the modem. The facsimile modem is expected to handle the task of converting image data per T.4 or T.6. This class of modems will support some form of tagged image file format (TIFF) graphics files.

6.5.1.5 Identifying a Facsimile Modem

The AT+FCLASS=? command on a facsimile modem prints the following responses based on the class definition supported by the modem:

Table 4. Modem Class Definitions

Response	Compliant Standard
0	Data Modem
1	EIA/TIA–578
2	SP–2388A (now obsolete)
2.0	TIA/EIA–592

References

1. *Digital Signal Processing Applications Using The ADSP-2100 Family* - Volume 2, Prentice-Hall, 1992.
2. Chris Buehler, et al., *A Practical Implementation of a V.34 Modem Transmitter and Receiver Using TMS320C50 DSPs*, University of Maryland, 1995.
3. *Draft Recommendation V.34*, ”A Modem Operating at Data Signalling Rates of Up to 28,800 bps for Use on the General Switched Telephone Network and on Leased Point-to-Point Two-wire Telephone Type Circuits” , ITU, June 1994.
4. Floreat Inc., Saratoga, CA.
5. David Smalley, ”Equalization Concepts: A Tutorial”, *Telecommunications Applications With The TMS320C5x DSPs*, 1994.
6. Mansoor A. Chishtie, ”Viterbi Implementation on the TMS320C5x for V.32 Modems”, *Telecommunications Applications With The TMS320C5x DSPs*, 1994.
7. *TMS320C54x User’s Guide*, Texas Instruments Inc., 1995.
8. Stephen J. Bigelow, *Understanding Telephone Electronics*, SAMS, 1993.
9. Robert L. Hummel, *Programmer’s Technical Reference: Data and Facsimile Communications*, Ziff-Davis Press, 1993.
10. *Recommendation V.17*, ”A 2-Wire Modem for Facsimile Applications with Rates Up to 14,400 bps”, ITU, 1991.

Appendix A MIPS/Memory Breakdown of V.32bis and V.34

This section presents rough estimates of the V.32bis code from Floreat Inc., on a TMS320C5x platform.

Table A–1. MIPS Estimate for V.32bis on a TMS320C5x Platform [2]

Function	MIPS
V.32bis	18
V.32	15
V.22bis	5
V.22	5
V.17 (14,400 fax)	10
V.29	7
V.27ter	4
V.21	4
V.42 and V.42bis	8
Voice (DTAD, Floreat SVD etc)	10
DTMF Detection (for DTAD)	5
TI Speakerphone	14
OS and miscellaneous	3

Table A–2. Program Memory Estimates for V.32bis on a TMS320C5x Platform

Function	Program Memory In Decimal Kwords
V.32bis (including Handshake code)	14.5
V.32	14.5
V.22bis (including V.22 bis stand–alone handshake code)	6
V.22	6
V.17	14.5
V.29 and V.27ter	10
V.42bis	2.5
V.42	6
Vector Quantization (Floreat proprietary Voice compression)	4.5
Vector Quantization code book	4.5
Main System control	1
AT commands:	
Basic and parser	5.5
Extended AT	5.8
Fax AT	1.6
Voice AT	2.0
Fax Class 1	0.8
Fax Class 2	4
Fax Class 2 Receiver	2.2
Miscellaneous (including Tone or Pulse dialing, Hardware init., PC serial communication, Autobauding, Call Progress, NVRAM access etc.)	4
TI speakerphone	1.4
Operating system	1

Table A–3. Data Memory Estimates for V.32bis on a TMS320C5x Platform

Function	Data Memory In Decimal Kwords
V.32bis and V.32	5.2
V.22bis and lower	0.5
V.29 and V.27ter	0.5
V.17	1
V.42	3.4
V.42bis	5
Vector Quantization (Floreat proprietary voice compression)	4.4
AT commands (Basic, AT parser)	4
AT commands (extended, voice, fax)	0.5
TI speakerphone	0.4
C system stack	1
Operating system	1.5

This section presents anticipated estimates for V.34 on a TMS320C5x platform.

Table A–4. MIPS Estimate for V.34 on a TMS320C5x Platform

Function	MIPS
Digital section of V.34 datapump:	18
Scrambler/descrambler	1
Parser, shell mapper, Trellis encoder, Precoder	4
Demapper, deparser	6
Viterbi decoder	7
Analog section of V.34 datapump	17
V.42 and V.42bis	8
DSVD vocoder (G.729A)	12
DSVD protocol	2 to 3

Appendix B Selected Communication Standards

T.2	Standardization of Group 1 Facsimile Apparatus for Document Transmission This standard provides a minimal definition of some terms and parameters used for G1 facsimile operations.
T.3	Standardization of Group 2 Facsimile Apparatus for Document Transmission This standard provides a definition of the protocols and signals used during G2 facsimile operations.
T.4	Standardization of Group 3 Facsimile Apparatus for Document Transmission This standard provides definitions of the protocols and signals used during G3 facsimile operations including supported resolutions, one- and two-dimensional encoding, and optional error control and error limiting modes.
T.6	Facsimile Coding Schemes and Coding Control Functions for Group 4 Facsimile Apparatus This standard provides a brief definition of the line encoding schemes and control functions that are to be used in the G4 facsimile service.
T.10	Document Facsimile Transmissions on Leased Telephone Type Circuits
T.10bis	Document Facsimile Transmissions in the General Switched Telephone Network
T.20	Standardized Test Chart for Facsimile Transmissions
T.21	Standardized Test Charts for Document Facsimile Transmissions
T.30	Procedures for Document Facsimile Transmissions in the General Switched Telephone Network This standard describes the procedures and signals used when operating G1, G2, and G3 facsimile services. Descriptions of the HDLC framing system, facsimile information control fields and set-up procedures are described.
T.434	Binary File Transfer Protocol for the Telematic Services

T.503	A Document Application Profile for the Interchange of Group 4 Facsimile Documents This standard provides a brief outline of the formats that are used to interchange facsimile documents that contain only graphics.
T.563	Terminal Characteristics for Group 4 Facsimile Apparatus
V.8bis	Procedures for the Identification and Selection of Common Modes of Operation Between DCE & DTE Over the GSTN or 2-Wire Leased Line This standard defines the first phase in the connection between any 2 multimedia terminals such as DSVD, videotelephony, H.324 etc. This standard has not been ratified by the ITU because of IP issues from Radish Communications.
V.17	A 2-Wire Modem for Facsimile Application with Rates Up to 14,400 Bits/Second This standard defines the half duplex modulation methods and operating sequences for facsimile operation at speeds of 7200, 9600, 12000, and 14400 bps.
V.21	300Bits/Second Duplex Modem Standardized For Use in the General Switched Telephone Network
V.22	1200 bps Duplex Modem Standardized for Use in the General Switched Telephone Network and On Point-To-Point 2-Wire Leased Telephone Type Circuits
V.22bis	2400 bps Duplex Modem Using the Frequency Division Technique Standardized on the GSTN and on Point-to-Point 2-Wire Leased Telephone Type Circuits
V.23	600/1200 Baud Modem Standardized for Use in the GSTN
V.24	This standard provides a list of definitions for Interchange circuits between Data Terminal equipment (DTE) and Data circuit terminating equipment (DCE).
V.25bis	Automatic Answering Equipment and/or Parallel Automatic Calling Equipment on the GSTN This standard provides procedures for disabling of echo control devices for both manually and automatically established calls.
V.25ter	Serial Asynchronous Automatic Dialing & Control

V.27ter	4800/2400 bps Modem with Automatic Equalizer Standardized for Use on Leased Telephone Type Circuits This standard defines an early standard supported by G3 facsimile.
V.29	9600 bps Modem Standardized for Use on Point-to-Point 4-Wire Leased Telephone Type Circuits This document defines a standard used by G3 facsimile and some non-standard data modems.
V.32	A Family of 2-Wire, Duplex Modems Operating at Data Signaling Rates of Up to 9600 bps for Use on the GSTN and on Leased Telephone Type Circuits This standard describes the modulation methods used for communications up to 9600 bps, speed negotiation options, echo cancellation and Trellis coding.
V.32bis	A Duplex Modem Operating at Data Signaling Rates of Up to 14,400 bps for Use on the GSTN and on Leased Point-to-Point 2-Wire Telephone Type Circuits This specification is a self contained extension of the CCITT V.32 specification. V.32bis defines modulation speeds up to 14,400 bps with fallback speeds of 12000, 9600, 7200, and 4800 bps.
V.33	14400 bps Modem Standardized for Use on Point-to-Point 4-Wire Leased Telephone Type Circuits
V.42	Error Correcting Procedures for DCE’s Using Asynchronous to Synchronous Conversion This standard describes an error control procedure that provides for error detection and automatic retransmission of data. The standard describes the LAPM (Link Access Procedure for Modems) as the primary method and the MNP2-4 protocols as an alternate.
V.42bis	Data Compression Procedures for DCE’s Using Error Correcting Procedures This standard defines the data compression method that can be provided during a LAPM error control link.
V.FAST	Rockwell Proprietary Standard for Modulation Methods Up to 28800 bps

V.34	A Modem Operating at Data Signaling Rates of Up to 28800 bps for Use on the GSTN and on Leased Point-to-Point 2-Wire Telephone Type Circuits Describes Symbol Rates, Primary Channel Data Signaling Rates, Trellis Coding, an Optional Auxiliary Channel, Automoding Techniques to V.Series Modems.
V.34Q	Modem Incorporating Simultaneous Voice & Data with Audio/Data Signaling Up to 31,200 bps and Data-Only Signaling Up to 33,600 bps, for Use on GSTN & 2-Wire Leased Line This is basically V.34plus (data rates of up to 33,600 bps) and Analog Simultaneous Voice and Data. This is still in the draft stage with Study Group 14 as of June 1996.
V.54	Loop Test Devices For Modems
V.70	This standard is referred to as DSVD-C describing the Control Procedures for a Digital Simultaneous Voice and Data (DSVD) Terminal.
V.75	This standard is referred to as DSVD-S describing the System Procedures for a Digital Simultaneous Voice and Data (DSVD) Terminal.
V.76	This standard is referred to as V.gmux describing the Multiplexing procedures for a DSVD terminal using slightly modified V.42/LAPM
V.80	This standard describes the in-band DCE control and synchronous data modes for asynchronous DTE. This was previously known as V.ib. It was ratified by the CCITT (now ITU) at the Study Group 14 meeting in March 1996. V80 also defines the application interface for H.324.
Bell103/113	Frequency Division Multiplexed Full-Duplex Asynchronous Signaling at Speeds From 0 to 300 bps Using Frequency Shift Keying
Bell 201	Full Duplex (201B) and Half-Duplex (201B/C). Synchronous Signaling at Speeds Up to 2400 bps Using 4-State Differential Phase Shift Keying (DPSK-4)
Bell 202	Half Duplex Asynchronous Signaling at Speeds Up to 1200 bps Using FSK and an On-Off Keyed Backchannel

Bell 212A	Frequency Division Multiplexed Full-Duplex Synchronous Signaling at Speeds Up to 1200 bps Using 4-State Differential Phase Shift Keying (DPSK-4)
G.729A	8000 bps Codec Used in a Digital Simultaneous Voice and Data (DSVD) Terminal.