Figure 8.12 Fractionally Spaced Equalizers for a 2 Mile, 26-Gauge Loop

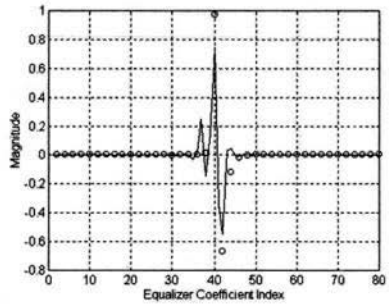
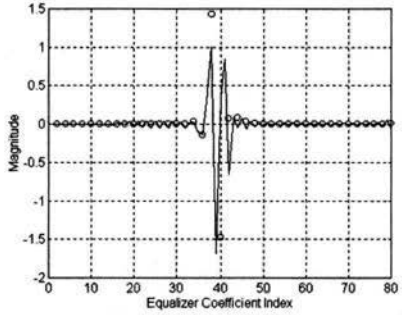


Figure 8.13 Fractionally Spaced Equalizers for a 3 Mile, 26-Gauge Loop



The expression

$$\xi_{\min} = E\left[a_{k-d}^2\right] - E\left[a_{k-d}\begin{bmatrix} X_k^1 \\ X_k^2 \end{bmatrix}^T\right]\begin{bmatrix} Q_{opt}^1 \\ Q_{opt}^2 \end{bmatrix}$$

describes the MMSE between the desired signal and the output of a fractionally spaced equalizer, which is also determined by receiver front-end noise, the twisted-pair loop channel, and the length of the channel equalizer.

For the same simple example of a 26-gauge, 10,560 foot loop with receiver front-end noise power density levels of –110 dBm/Hz, –120 dBm/Hz, and –130 dBm/Hz, we have MMSE levels of –20.1 dB, –30 dB, and –39.6 dB, respectively.

The equalizer coefficients of a fractionally spaced equalizer are also identified iteratively with the LMS algorithm and updated at the signaling baud rate, but not at the sampling rate.

The LMS algorithm for the updating of a two times over sampled equalizer coefficients is expressed as

$$\begin{bmatrix} Q_{k+1}^1 \\ Q_{k+1}^2 \end{bmatrix} = \begin{bmatrix} Q_k^1 \\ Q_k^2 \end{bmatrix} + \mu \begin{bmatrix} X_k^1 \\ X_k^2 \end{bmatrix} \left(a_{k-d} - \begin{bmatrix} Q_k^1 \\ Q_k^2 \end{bmatrix}^T \begin{bmatrix} X_k^1 \\ X_k^2 \end{bmatrix} \right)$$

In a finite bit resolution hardware implementation, the fractionally spaced equalizer can become unstable when coefficient values exceed the hardware limitation. A tap leakage algorithm[6] is usually required for the adaptation of fractionally spaced equalizer coefficients, as expressed in the following:

$$\begin{bmatrix} Q_{k+1}^1 \\ Q_{k+1}^2 \end{bmatrix} = (1-\alpha) \begin{bmatrix} Q_k^1 \\ Q_k^2 \end{bmatrix} + \mu \begin{bmatrix} X_k^1 \\ X_k^2 \end{bmatrix} \left(a_{k-d} - \begin{bmatrix} Q_k^1 \\ Q_k^2 \end{bmatrix}^T \begin{bmatrix} X_k^1 \\ X_k^2 \end{bmatrix} \right)$$

The value α is usually smaller than 0.1.

8.1.5 *Noise Power After Equalization*

The effect of intersymbol interference is minimized by the equalizer, and the energy of the received signal is concentrated on to a single baud interval sampling point; however, because of the concentration effect of the equalizer, the noise power is also amplified. Assuming the noise component has a power of σ^2 at every sampling point, the noise power after the equalization is then

$$\sigma^2 \sum_{i=1}^{m} q_i^2$$

The signal-to-noise ratio after equalization is

$$SNR_{eq} = \frac{1}{\sigma^2 \sum_{i=1}^{m} q_i^2}$$

The signal-to-noise ratio at a receiver front end with n sampling point is

$$SNR_{rf} = \frac{\sum_{i=1}^{n} h^2(i)}{n\sigma^2}$$

The relationship between receiver front-end signal-to-noise ratio and equalizer output signal-to-noise ratio is expressed as

$$SNR_{eq} = SNR_{rf} \frac{n}{\sum_{i=1}^{n} h^2(i) \sum_{i=1}^{m} q_j^2}$$

The noise enhancement introduced by the equalization is[7]

$$\frac{\sum_{i=1}^{n} h^2(i) \sum_{i=1}^{m} q_j^2}{n}$$

The receiver front-end peak signal-to-noise ratio is also defined as

$$SNR_{rf} = \frac{Max\left[h^2(i)\right]}{\sigma^2}$$

The relationship between receiver front-end peak signal-to-noise ratio and equalizer output signal-to-noise ratio is

$$SNR_{eq} = SNR_{rf} \frac{1}{Max\left[h^2(i)\right] \sum_{i=1}^{m} q_j^2}$$

The corresponding noise enhancement is

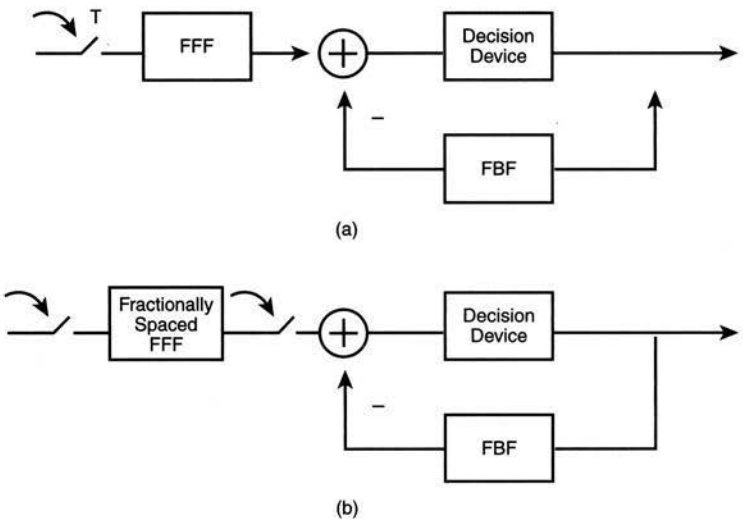
$$Max\left[h^2(i)\right] \sum_{i=1}^{m} q_j^2$$

For an example of a 2 mile, 26-gauge twisted-pair loop, the noise enhancement is 1.34 and 1.05 for baud rate and fractionally spaced equalizers, respectively, compared with receiver front-end maximum signal-to-noise ratio.

8.2 Decision Feedback Channel Equalization

The amount of noise enhancement discussed in section 8.1.4, "Fractionally Spaced Equalizer Performance," can be reduced by minimizing the required number of equalizer coefficients while still achieving the desired equalization effect. Using a *Decision Feedback Equalizer* (DFE) enables the realization of this less noise enhancement objective.[8,9,10] Figure 8.14 shows a DFE, which consists of a *FeedForward Filter* (FFF) and a *FeedBack Filter* (FBF). The FFF also can be fractionally spaced, as depicted in part b of Figure 8.14.

Figure 8.14 The Structure of a Decision Feedback Equalizer



The received signal is only used by the FFF while the estimated signal is used by the FBF. An FFF filtered version of the tail portion of the channel impulse response is canceled by the FBF weighted estimated signals. The noise enhancement of a DFE is only caused by the equalization of the remaining much smaller portion of the channel impulse response.

8.2.1 Determining DFE Coefficients

The DFE equalizer coefficients also can be derived in the discrete time domain. The instantaneous error between the desired signal and the DFE output is

$$e_k = a_{k-d} - X_k^T Q^f + A_{k-d-1}^T Q^b$$

a_{k-d} is the desired data symbol after d symbol intervals of channel delay.

$X_k^T = \begin{bmatrix} x_k & x_{k-1} & \dots & x_{x-n+1} \end{bmatrix}$ is the received signal vector.

$Q^{fT} = \begin{bmatrix} q_1^f & q_2^f & \dots & q_n^f \end{bmatrix}$ is the FFF coefficient vector.

$A_{k-d-1}^T = \begin{bmatrix} a_{k-d-1} & a_{k-d-2} & \dots & a_{x-d-m} \end{bmatrix}$ is the estimated data symbol vector for previous m data symbols.

$Q^{bT} = \begin{bmatrix} q_1^b & q_2^b & \dots & q_m^b \end{bmatrix}$ is the FBF coefficient vector.

e_k is the error between the desired signal and the decision feedback equalizer output.

The expected value of the square error or mean squared error for a particular set of equalizer coefficient vectors Q^f and Q^b is

$$E\left[e_k^2\right] = E\left[\left(a_{k-d} - X_k^T Q^f + A_{k-d-1}^T Q^b\right)^2\right]$$

Derivatives of the mean square error with respect to the equalizer coefficient vectors Q^f and Q^b are

$$\frac{\partial\left(E\left[e_k^2\right]\right)}{\partial Q^f} = -2E\left[\left(a_{k-d} - X_k^T Q^f + A_{k-d-1}^T Q^b\right) X_k^T\right]$$

$$\frac{\partial\left(E\left[e_k^2\right]\right)}{\partial Q^b} = -2E\left[\left(a_{k-d} - X_k^T Q^f + A_{k-d-1}^T Q^b\right) A_{k-d-1}^T\right]$$

By setting derivatives equal to zero, we obtain the optimal equalizer coefficient vectors Q_{opt}^f and Q_{opt}^b for the minimum mean squared error as

$$Q_{opt}^{fT} E\left[X_k X_k^T\right] - Q_{opt}^{bT} E\left[A_{k-d-1} X_k^T\right] = E\left[a_{k-d} X_k^T\right]$$

$$Q_{opt}^{fT} E\left[X_k A_{k-d-1}^T\right] - Q_{opt}^{bT} E\left[A_{k-d-1} A_{k-d-1}^T\right] = E\left[a_{k-d} A_{k-d-1}^T\right]$$

or

$$Q_{opt}^{fT} = \left(E\left[X_k X_k^T\right] - E\left[X_k A_{k-d-1}^T\right] E\left[A_{k-d-1} X_k^T\right]\right)^{-1} E\left[a_{k-d} X_k^T\right]$$

$$Q_{opt}^{bT} = Q_{opt}^{fT} E\left[X_k A_{k-1}^T\right]$$

This solution also can be written in matrix form as

$$\begin{bmatrix} Q_{opt}^{f} \\ Q_{opt}^{b} \end{bmatrix} = \begin{bmatrix} R_{XX} & R_{XA} \\ R_{XA}^T & I \end{bmatrix}^{-1} \begin{bmatrix} R_{Xa} \\ 0 \end{bmatrix}$$

in which

$$R_{XX} = E\left[X_k X_k^T\right]$$

$$R_{Xa} = E\left[X_k a_{k-d}\right]$$

and

$$R_{XA} = E\left[X_k A_{k-d-1}^T\right] = \overline{H}^T E\left[\begin{bmatrix} a_k \\ a_{k-1} \\ \cdots \\ a_{k-n-m+2} \end{bmatrix} \begin{bmatrix} a_{k-d-1} & a_{k-d-2} & \cdots & a_{k-d-m} \end{bmatrix}\right]$$

$$= \overline{H}^T E\begin{bmatrix} a_k a_{k-d-1} & \cdots & \cdots & a_k a_{k-d-m} \\ a_{k-1} a_{k-d-1} & \cdots & \cdots & a_{k-1} a_{k-d-m} \\ \cdots & \cdots & \cdots & \cdots \\ a_{k-d-1} a_{k-d-1} & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & a_{k-d-m} a_{k-d-m} \\ \cdots & \cdots & \cdots & \cdots \\ 0 a_{k-n-m+2} a_{k-d-1} & \cdots & \cdots & a_{k-n-m+2} a_{k-d-m} \end{bmatrix} = \overline{H}^T \begin{bmatrix} 0 & \cdots & \cdots & 0 \\ 0 & \cdots & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 1 & \cdots & \cdots & \cdots \\ \cdots & 1 & \cdots & \cdots \\ \cdots & \cdots & \cdots & 1 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & \cdots & \cdots & 0 \end{bmatrix} = \overline{H}^T P$$

Notice that the DFE solution also can be obtained by expanding the received signal vector to include transmitting data symbols. Assume a super received signal vector to be

$$\tilde{X}_k^T = \left[X_k^T \quad A_{k-d-1}^T\right] = \left[x_k \quad x_{k-1} \quad \ldots \quad x_{k-n+1} \quad a_{k-d-1} \quad a_{k-d-2} \quad \ldots \quad a_{k-d-m}\right]$$

$$\tilde{X}_k^T = \left[a_k \quad a_{k-1} \quad \ldots \quad a_{k-n-m+2}\right] \begin{bmatrix} h(1) & 0 & \ldots & 0 & 0 & \ldots & 0 & 0 \\ h(2) & h(1) & \ldots & 0 & 0 & \ldots & 0 & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ \ldots & \ldots & \ldots & \ldots & 1 & \ldots & 0 & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & 1 & 0 \\ 0 & \ldots & \ldots & \ldots & 0 & \ldots & 0 & 1 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & h(n) & 0 & \ldots & 0 & 0 \end{bmatrix}$$

$$+\left[n_k \quad n_{k-1} \quad \ldots \quad n_{k-n+1} \quad 0 \quad \ldots \quad 0\right] = \tilde{A}_k^T \tilde{H} + \tilde{N}_k^T$$

$$R_{\tilde{X}\tilde{X}} = E\left[\tilde{X}_k \tilde{X}_k^T\right] = \left[\tilde{H}^T \tilde{H}\right] + E\left[\tilde{N}_k \tilde{N}_k^T\right] = \begin{bmatrix} H^T H & H^T P \\ \left(H^T P\right)^T & I \end{bmatrix} + \begin{bmatrix} E\left[N_k N_k^T\right] & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} R_{XX} & R_{XA} \\ \left(R_{XA}\right)^T & I \end{bmatrix}$$

We can use this method to derive equalizer coefficient vectors for the DFE with a fractionally spaced FFF. For example, we set two super received signal vectors for two sampling phases of the received signal

$$\left(\tilde{X}_k^1\right)^T = \left[\left(X_k^1\right)^T \quad [0 \quad 0 \quad \ldots \quad 0]^T\right]$$

and

$$\left(\tilde{X}_k^2\right)^T = \left[\left(X_k^2\right)^T \quad \left(A_{k-d-1}^m\right)^T\right]$$

Replacing these super received signal vectors into the solution for the fractionally spaced equalizer, we then have

$$\begin{bmatrix} \tilde{Q}^1_{opt} \\ \tilde{Q}^2_{opt} \end{bmatrix} = \left(\begin{bmatrix} E\left[\tilde{X}^1_k \tilde{X}^{1T}_k\right] & E\left[\tilde{X}^1_k \tilde{X}^{2T}_k\right] \\ E\left[\tilde{X}^2_k \tilde{X}^{1T}_k\right] & E\left[\tilde{X}^2_k \tilde{X}^{2T}_k\right] \end{bmatrix} \right)^{-1} E\left[a_{k-d} \begin{bmatrix} \tilde{X}^1_k \\ \tilde{X}^2_k \end{bmatrix} \right]$$

$$= \left(\begin{bmatrix} E\left[X^1_k X^{1T}_k\right] & 0 & E\left[X^1_k X^{2T}_k\right] & E\left[X^1_k \left(A^m_{k-d-1}\right)^T\right] \\ 0 & 0 & 0 & 0 \\ E\left[X^2_k X^{1T}_k\right] & 0 & E\left[X^2_k X^{2T}_k\right] & E\left[X^2_k \left(A^m_{k-d-1}\right)^T\right] \\ E\left[A^m_{k-d-1}\left(X^1_k\right)^T\right] & 0 & E\left[A^m_{k-d-1}\left(X^2_k\right)^T\right] & I \end{bmatrix} \right)^{-1} E\left[a_{k-d} \begin{bmatrix} X^1_k \\ 0 \\ X^1_k \\ A^m_{k-d-1} \end{bmatrix} \right] = \begin{bmatrix} Q^1_{opt} \\ 0 \\ Q^2_{opt} \\ Q^b_{opt} \end{bmatrix}$$

By eliminating the second zero row and the second zero column, we obtain the equalizer coefficient solution for the DFE with a fractionally spaced FFF as

$$\begin{bmatrix} Q^1_{opt} \\ Q^2_{opt} \\ Q^b_{opt} \end{bmatrix} = \left(\begin{bmatrix} E\left[X^1_k X^{1T}_k\right] & E\left[X^1_k X^{2T}_k\right] & E\left[X^1_k \left(A^m_{k-d-1}\right)^T\right] \\ E\left[X^2_k X^{1T}_k\right] & E\left[X^2_k X^{2T}_k\right] & E\left[X^2_k \left(A^m_{k-d-1}\right)^T\right] \\ E\left[A^m_{k-d-1}\left(X^1_k\right)^T\right] & E\left[A^m_{k-d-1}\left(X^2_k\right)^T\right] & I \end{bmatrix} \right)^{-1} = E\left[a_{k-d} \begin{bmatrix} X^1_k \\ X^2_k \\ A^m_{k-d-1} \end{bmatrix} \right]$$

This fractionally spaced FFF DFE solution also can be obtained from that for the baud rate DFE by using

$$R_{XX} = E\begin{bmatrix} X^1_k X^{1T}_k & X^1_k X^{2T}_k \\ X^2_k X^{1T}_k & X^2_k X^{2T}_k \end{bmatrix}$$

$$R_{XA} = E\begin{bmatrix} X^1_k \left(A^m_{k-d-1}\right)^T \\ X^2_k \left(A^m_{k-d-1}\right)^T \end{bmatrix}$$

$$R_{Xa} = E\begin{bmatrix} X^1_k a_{k-d} \\ X^2_k a_{k-d} \end{bmatrix} \quad \text{and}$$

$$Q^f_{opt} = \begin{bmatrix} Q^{f1}_{opt} \\ Q^{f2}_{opt} \end{bmatrix}$$

Figures 8.15, 8.16, and 8.17 show normalized baud rate FFF and FBF coefficients as well as baud rate linear equalizer coefficients, as a comparison, for 1, 2, and 3 miles of 26-gauge twisted-pair loops, respectively. Again, the channel impulse response is normalized to produce compensation filter coefficients and compensated channel impulse response of the similar magnitudes for easy comparison.

Figure 8.15 Channel Equalizers for a 1 Mile, 26-Gauge Loop

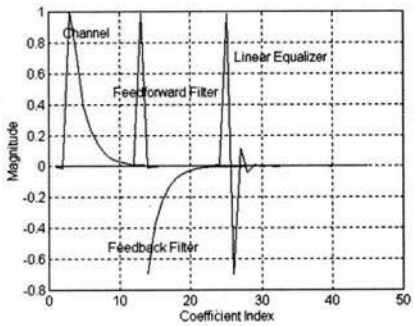
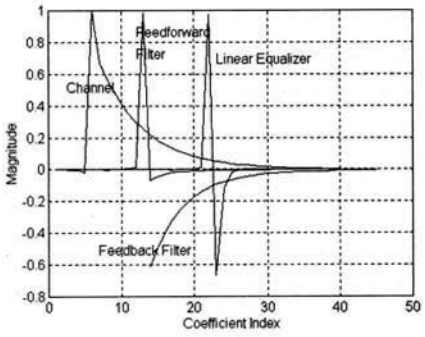


Figure 8.16 Channel Equalizers for a 2 Mile, 26-Gauge Loop



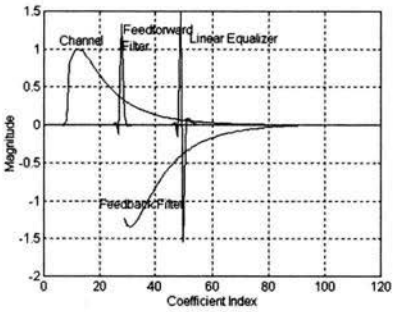
8.2.2 *DFE MMSE Levels*

The MMSE between the desired signal and the output of a DFE is also determined by receiver front-end noise, the twisted-pair loop channel, and the structure of a DFE. However, the amount of MMSE is only related to the FFF due to the fact that the FBF does not contribute to the noise enhancement. The expression

$$\xi_{\min} = E\left[a_k^2\right] - R_{Xa}^T Q_{opt}^f$$

describes the MMSE between the desired signal and the output of a DFE.

Figure 8.17 Channel Equalizers for a 3 Mile, 26-Gauge Loop



For the simple example of a 26-gauge, 10,560 foot loop with receiver front-end noise power density levels of –110 dBm/Hz, –120 dBm/Hz, and –130 dBm/Hz, we have MMSE levels of –18.38 dB, –28.3 dB, and –38 dB, respectively. A DFE with a fractionally spaced FFF can produce lower MMSE levels. The performance of an equalizer also is often judged by the SNR at the equalizer output. We have

$$SNR \approx \frac{1}{\xi_{\min}}$$

Table 8.1 summarizes equalizer output SNRs for different equalizer structures as well as the receiver front-end peak signal-to-noise ratio for similar transmission conditions. A 2 mile, 26-gauge twisted-pair loop is used for receiver front-end noise density levels of –110 dBm/Hz, –120 dBm/Hz, and –130 dBm/Hz. SNR estimates in Table 8.1 do not include the effect of finite precision implementation of channel equalizers. The sampling rate is 300 kHz for a bandwidth of 150 kHz.

Table 8.1 Receiver Front-End and Equalizer Output SNRs

Noise Level	Receiver Front	Linear	Fraction Linear	DFE	Fraction DFE
–110 dBm/Hz	20.9	18.3	20.1	18.4	22.72
–120 dBm/Hz	30.9	27.9	30.0	28.3	32.7
–130 dBm/Hz	40.9	36.0	39.6	38.0	42.7

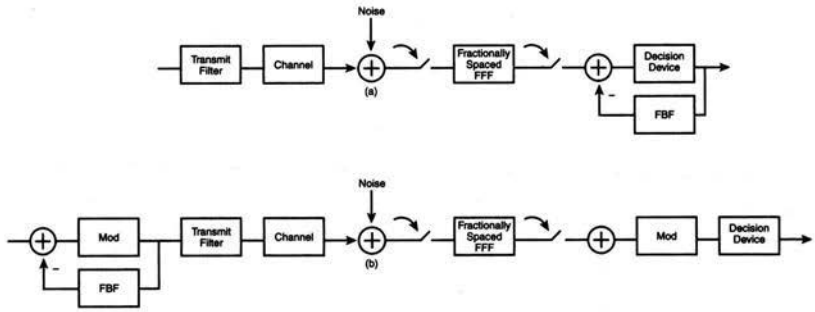
8.2.3 DFE Error Propagation and Precoding

As indicated by results of Table 8.1, a DFE has a better performance compared with a linear equalizer. Also, a DFE with a fractionally spaced FFF has the best performance. The drawback of a DFE, however, is the error propagation problem.[11]

Because the tail cancellation signal from the FBF is generated according to the estimated signal, a wrong estimation of the signal could reduce the chance of correct estimation of subsequent signals. In fact, for a channel with severe intersymbol interference, which can be characterized by a channel impulse response with a long tail or a long FBF with significant magnitudes, multiple errors do occur because of the phenomenon of error propagation. It might take many symbol intervals for a DFE to recover from the error propagation.

The Tomlinson[12]/Harashima[13] precoding method can be used to avoid the DFE error propagation problem. In this precoding method, the FBF (depicted in part (a) of Figure 8.18) is moved into the transmitter to filter the original data symbols (depicted in part (b) of Figure 8.18).

Figure 8.18 Comparison Between a DFE System and a Tomlinson Precoding System



In addition, a modulo operation, *Mod*, is used in the feedback loop to make the operation stable. Without this modulo operation, the FBF would behave like a recursive filter, which could become unstable depending on values of filter coefficients. Another modulo operation is used after the FFF in the receiver to recover the original signal magnitude. The modulo operation is designed to contain the output signal to be in the same magnitude range as that of the input signal after the loopback of FBF. For a line code with M equally spaced signal levels, the modulo operation is implemented as

$$y(t) = remain\left(\frac{x(t)+(2n+1)M}{2M}\right) - M$$

for $y(t) > M$ or $y(t) < -M$ in which $n = 0,1,2...$ is chosen such that $x(t) + (2n+1)M > 0$.

In a normal DFE, the FBF does not form a feedback loop by itself, but through a decision device. As long as decisions are correct, the FBF is equivalent to an FIR filter with data symbols as its input. Even when the decision device makes wrong decisions, the magnitude of the equalizer output is still limited by the decision device. Therefore, a DFE is always stable as an FIR filter.

FBF filter coefficients for a straight twisted-pair loop can be approximated by

$$h_{FBF}(k) = Ae^{-Bk}$$

B is larger than zero. The z transformation of this FIR filter is

$$H_{FBF}(z) = A\frac{z}{z - e^{-B}}$$

and is always stable because $e^{-B} < 1$. When this FBF is used in a Tomlinson/Harashima transmitter without a modulo operation, we have

$$y(k) = x(k) - y(k-1)h(1) - y(k-2)h(2) - ...$$

This relationship in z domain is

$$Y(z) = X(z)\frac{1}{1 + z^{-1}H_{FBF}(z)}$$

The z transformation of the FBF in a transmitter is

$$H'_{FBF}(z) = \frac{1}{1 + \dfrac{A}{z - e^{-B}}} = \frac{z - e^{-B}}{z - \left(e^{-B} - A\right)}$$

The feedback loop becomes unstable when $e^{-B} - A$ is close to 1 for a pole near the unit circle. A twisted-pair loop usually consists of many different sections and with bridged taps. There are many possible loop configurations where the FBF used in a Tomlinson/Harashima transmitter could become unstable.

You can examine the effect of modulo operations by assuming that the FFF normalizes received signal level and equalizes part of the channel distortion such that the remaining channel distortion can be completely compensated by the FBF. In other words, the equivalent channel transfer function after FFF can be described as

$$H_{channel}(z) = 1 + z^{-1}H_{FBF}(z)$$

$H_{FBF}(z)$ is the transfer function of the FBF. The $z^{-1}H_{FBF}(z)$ part of the equivalent channel causes the expansion of the received signal.[14] The received signal expansion is similar to the PAR increase caused by a filter or a channel. In a conventional DFE, the received signal magnitude expansion caused by the channel is compensated by FBF before a decision device.

In a Tomlinson/Harashima transceiver, the signal magnitude expansion at the transmitter is limited by the modulo operation. However, the signal magnitude expansion caused by the channel remains because of the mission of a FBF. A modulo operation is used instead to map the received signal to the original signal constellation.

Figure 8.19 shows the following:

- The effect of modulo operation after FBF feedback loop, $Mod_N(x-a)$
- The effect of channel, $Mod_N(x-a)+a$
- The modulo operation in the receiver, $Mod_N\big(Mod_N(x-a)+a\big)$
- The identity of $Mod_N(x) = Mod_N\big(Mod_N(x-a)+a\big)$

The duo-modulo operation makes the modulo operation transparent under the channel-distortion-only condition; however, the received signal before a decision device does not have to be confined by maximum and minimum levels of a particular line code. The receiver front-end additive noise also can cause the received signal to exceed signal maximum or minimum levels. This slight signal magnitude expansion should be preserved for making correct data symbol estimate decisions; otherwise, *data-flipping* will occur.[15]

When data-flipping occurs, a data symbol corresponding to the maximum signal level can be mistaken as the data symbol with the minimum signal level because an additive noise caused the received signal to exceed the maximum level and the modulo operation to bring it down to near the minimum signal level, as shown in Figure 8.20.

Figure 8.19 Effects of Modulo Operations

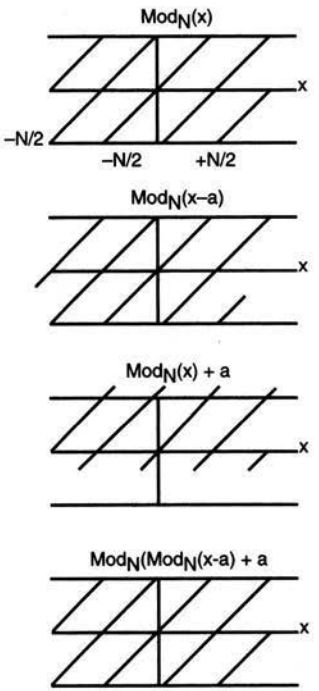
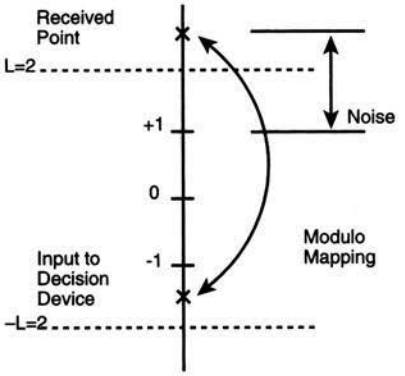


Figure 8.20 The Data-Flipping Effect of a Tomlinson/Harashima System



The effect of data-flipping can be minimized by using a slightly larger allowed signal magnitude range for the receiver modulo operation. The additional allowed magnitude range is related to the receiver front-end noise level and the degree of channel distortion.

8.3 Passband Equalization and Carrierless AM/PM

The direct application of data symbols of different magnitudes creates a baseband signal spectrum. Spectrum shaping of a baseband signal can be realized using transmit filters that correlate to adjacent data symbols. Modulating data symbols with a carrier of a particular frequency can generate a passband signal.

A Quadrature Amplitude Modulation (QAM) line code is realized by modulating the cosine phase and sine phase of a particular carrier with independent data sequences. A Carrierless AM/PM (CAP) line code can be considered as a variation of the QAM line code with the omission of a phase rotator in the transmitter and a phase derotator in the receiver. Transmitter shaping filters are used by passband line codes for the control of signal bandwidth.

8.3.1 QAM Baseband and Passband Implementation

A passband Quadrature Amplitude Modulation (QAM) signal is usually generated with a pair of transmitter shaping filters and a carrier modulator consisting of a cosine phase and a sine phase. A QAM signal after shaping filter and modulation can be described as

$$f(t) = \sum_{i=-\infty}^{\infty} a_i g(t - iT)\cos\omega_c t - \sum_{i=-\infty}^{\infty} b_i g(t - iT)\sin\omega_c t$$

The values are as follows:

a_is are in-phase channel data symbols.

b_is are quadrature channel data symbols.

$g(t)$ is the baseband shaping filter.

Denoting a complex data symbol as $C_i = a_i + jb_i$ results in

$$f(t) = \mathrm{Re}\left[\sum_{i=-\infty}^{\infty} C_i g(t - iT)e^{j\omega_c t}\right]$$

This can be rearranged to

$$f(t) = \mathrm{Re}\left[\sum_{i=-\infty}^{\infty} C_i e^{j\omega_c iT} g(t - iT)e^{j\omega_c (t-iT)}\right]$$

or in another format as

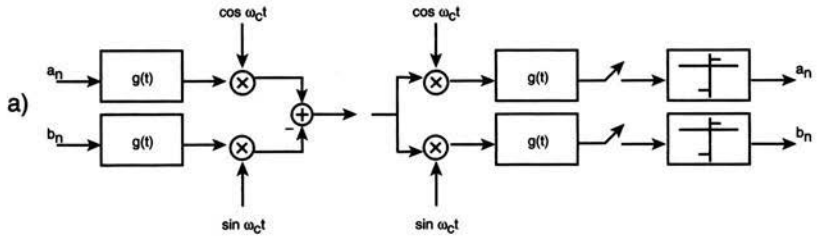
$$f(t) = \sum_{i=-\infty}^{\infty} (a_i \cos \varpi_c iT - b_i \sin \omega_c iT) g(t - iT) \cos \omega_c (t - iT)$$

$$- \sum_{i=-\infty}^{\infty} (a_i \sin \omega_c iT + b_i \cos \omega_c iT) g(t - iT) \sin \omega_c (t - iT)$$

The preceding expression describes an alternative QAM signal generation process (see Figure 8.22) in which the data symbols are phase rotated first and then convolved with a pair of passband shaping filters. The in-phase channel passband shaping filter is the product of the baseband shaping filter and the cosine phase of the carrier. The quadrature channel passband shaping filter is the product of the baseband shaping filter and the sine phase of the carrier.

For an ideal channel, data symbols carried by a QAM signal are usually recovered first through demodulation and then through lowpass filtering, as depicted in Figure 8.21.

Figure 8.21 A Conventional QAM Transceiver Pair



The in-phase channel signal is demodulated by the cosine phase of the carrier, and the quadrature channel signal is demodulated by the sine phase of the carrier. There is a lowpass filter for each channel. The transmission characteristics of the lowpass filter could be the same as those of the transmitter shaping filter. The demodulation and lowpass filtering process for the recovery of data symbols is expressed as

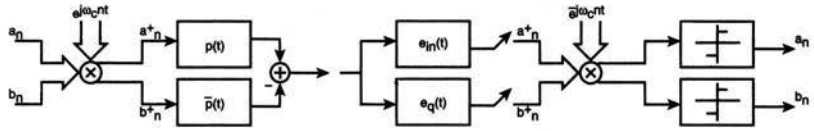
$$\hat{a}_k + j\hat{b}_k = \left[\left(f(t) e^{-j\omega_c t} \right) * g(t) \right]_{t=kT}$$

By changing the order of demodulation and lowpass filtering, data symbols carried by a QAM signal also can be recovered first by in-phase and quadrature bandpass filtering and then followed by a phase derotating process similar to the alternative approach of QAM

transmitter implementation, as shown in Figure 8.22. The passband filtering and phase derotation process for the recovery of data symbols is expressed as

$$\hat{a}_k + j\hat{b}_k = \left[\left(f(t) * \left(g(t)e^{j\omega_c t}\right)\right)e^{-j\omega_c t}\right]_{t=kT}$$

Figure 8.22 An Alternative QAM Implementation with a Phase Modulator and a Phase Demodulator



Linear channel equalizers might be necessary to compensate for the channel distortion in a practical implementation. A linear channel equalizer can be placed after these lowpass filters. The demodulation, lowpass filtering, and linear equalization process for the recovery of data symbols is expressed as

$$\hat{a}_k + j\hat{b}_k = \left[\left(f(t)e^{-j\omega_c t}\right) * g(t) * Q_{IQ}(t)\right]_{t=kT}$$

The linear equalizer consists of a real part, in-phase equalizer, and an imaginary part, cross-coupled equalizer, expressed as

$$Q_{IQ}(t) = Q_I(t) + jQ_Q(t)$$

Similar to the lowpass filter case, we can move the position of the channel equalizer around at different positions. Replacing the received signal, $f(t)$, with

$$\left(\mathrm{Re}\left(\left(C_i * g(t)\right)e^{j\omega_c t}\right)\right) * h(t)$$

The value $h(t)$ is the channel impulse response; the received signal in the front of a channel equalizer is

$$\left(\left(\left(\mathrm{Re}\left(\left(C_i * g(t)\right)e^{j\omega_c t}\right)\right) * h(t)\right)e^{-jw_c t}\right) * g(t)$$

By changing the order of multiplication and convolution, we obtain

$$\left(\mathrm{Re}\left((C_i * g(t))e^{j\omega_c t}\right) * h(t)\right)e^{-j\omega_c t} = \left(\mathrm{Re}\left((C_i * g(t))e^{j\omega_c t}\right)e^{-j\omega_c t}\right) * h(t)e^{-j\omega_c t}$$

In particular, we have

$$\mathrm{Re}\left((C_i * g(t))e^{j\omega_c t}\right)e^{-j\omega_c t} = \left((a_k * g(t))\frac{e^{j\omega_c t} + e^{-j\omega_c t}}{2} - (b_k * g(t))\frac{e^{j\omega_c t} - e^{-j\omega_c t}}{2j}\right)e^{-j\omega_c t}$$

$$= \frac{a_k * g(t)}{2} + j\frac{b_k * g(t)}{2} + F\left(e^{-j2\omega_c t}\right)$$

Using the preceding relationships, the received signal in the front of a channel equalizer also can be described as

$$\left(\left(\left(\mathrm{Re}\left((C_i * g(t))e^{j\varpi_c t}\right)\right) * h(t)\right)e^{-j\omega_c t}\right) * g(t)$$

$$= \left(\frac{a_k * g(t)}{2} + j\frac{b_k * g(t)}{2}\right) * \left(h(t)\cos\omega_c t - jh(t)\sin\omega_c t\right) * g(t)$$

$$= \frac{1}{2}\left(\begin{array}{l} a_k * g(t) * h(t)\cos\omega_c t + b_k * g(t) * h(t)\sin\omega_c t \\ +j\left(b_k * g(t) * h(t)\cos\omega_c t - a_k * g(t) * h(t)\sin\omega_c t\right) \end{array}\right) * g(t)$$

The previous expressions assume that the high frequency components $F\left(e^{-j2\omega_c t}\right)$ were eliminated by the lowpass filter $g(t)$. We therefore identify the QAM in-phase passband equivalent channel as

$$h_I = g(t) * h(t)\cos\varpi_c t * g(t) - jg(t) * h(t)\sin\omega_c t * g(t) = h_{REAL}(t) - jh_{IMAG}(t)$$

and the QAM quadrature passband equivalent channel as

$$h_Q = g(t) * h(t)\cos\varpi_c t * g(t) + jg(t) * h(t)\sin\omega_c t * g(t) = h_{REAL}(t) + jh_{IMAG}(t)$$

We further call

$$h_{REAL}(t) = g(t) * h(t)\cos\varpi_c t * g(t)$$

the real passband equivalent channel and

$$h_{IMAG}(t) = g(t) * h(t) \sin \omega_c t * g(t)$$

the imaginary passband equivalent channel.

8.3.2 *Calculation of QAM Linear Equalizer*

Equalizer coefficients for a QAM transceiver can be similarly calculated. For the in-phase passband equivalent channel, we have the relationship between the desired data symbol and in-phase and cross-coupled equalizer combined output as

$$e_k = a_{k-d} - X_k^T Q_I - Y_k^T Q_Q$$

a_{k-d} is the desired data symbol after d baud rate intervals of channel delay.

$X_k^T = [x_k \quad x_{k-1} \quad \dots \quad x_{x-n+1}]$ is the received signal vector from the real part of the passband equivalent channel.

$Y_k^T = [y_k \quad y_{k-1} \quad \dots \quad y_{x-n+1}]$ is the received signal vector from the imaginary part.

$Q_I^T = [q_{1I} \quad q_{2I} \quad \dots \quad q_{nI}]$ is the in-phase equalization coefficient vector.

$Q_Q^T = [q_{1Q} \quad q_{2Q} \quad \dots \quad q_{nQ}]$ is the cross-coupled equalizer coefficient vectors.

e_k is the error between the desired signal and the combined equalizer output.

We can further express the received signal as a result of data symbols convolved with equivalent channels plus noise. We thus have

$$x_k = \sum_{i=1}^{l} a_{k-i+1} h_{REAL}(i) + \sum_{i=1}^{l} b_{k-i+1} h_{IMAG}(i) + n_{I,k}$$

$$= [a_k \quad a_{k-1} \quad \dots \quad a_{k-l+1}] \begin{bmatrix} h_{REAL}(1) \\ h_{REAL}(2) \\ \dots \\ h_{REAL}(l) \end{bmatrix} + [b_k \quad b_{k-1} \quad \dots \quad b_{k-l+1}] \begin{bmatrix} h_{IMAG}(1) \\ h_{IMAG}(2) \\ \dots \\ h_{IMAG}(l) \end{bmatrix} + n_{I,k}$$

and

$$y_k = \sum_{i=1}^{l} b_{k-i+1} h_{REAL}(i) - \sum_{i=1}^{l} a_{k-i+1} h_{IMAG}(i) + n_{Q,k}$$

$$= \begin{bmatrix} b_k & b_{k-1} & \cdots & b_{k-l+1} \end{bmatrix} \begin{bmatrix} h_{REAL}(1) \\ h_{REAL}(2) \\ \cdots \\ h_{REAL}(l) \end{bmatrix} - \begin{bmatrix} a_k & a_{k-1} & \cdots & a_{k-l+1} \end{bmatrix} \begin{bmatrix} h_{IMAG}(1) \\ h_{IMAG}(2) \\ \cdots \\ h_{IMAG}(l) \end{bmatrix} + n_{Q,k}$$

We further have

$$X_k^T = \begin{bmatrix} a_k & a_{k-1} & \cdots & a_{k-l-n+2} \end{bmatrix} \begin{bmatrix} h_{REAL}(1) & 0 & 0 & 0 \\ h_{REAL}(2) & h_{REAL}(1) & 0 & 0 \\ h_{REAL}(3) & h_{REAL}(2) & 0 & 0 \\ \cdots & \cdots & \cdots & h_{REAL}(1) \\ h_{REAL}(l) & h_{REAL}(l-1) & 0 & h_{REAL}(2) \\ 0 & h_{REAL}(l) & 0 & h_{REAL}(5) \\ 0 & 0 & 0 & \cdots \\ 0 & 0 & 0 & h_{REAL}(l) \end{bmatrix}$$

$$+ \begin{bmatrix} b_k & b_{k-1} & \cdots & b_{k-l-n+2} \end{bmatrix} \begin{bmatrix} h_{IMAG}(1) & 0 & 0 & 0 \\ h_{IMAG}(2) & h_{IMAG}(1) & 0 & 0 \\ h_{IMAG}(3) & h_{IMAG}(2) & 0 & 0 \\ \cdots & \cdots & \cdots & h_{IMAG}(1) \\ h_{IMAG}(l) & h_{IMAG}(l-1) & 0 & h_{IMAG}(2) \\ 0 & h_{IMAG}(l) & 0 & h_{IMAG}(5) \\ 0 & 0 & 0 & \cdots \\ 0 & 0 & 0 & h_{IMAG}(l) \end{bmatrix} + \begin{bmatrix} n_{Q,k} & n_{Q,k-1} & \cdots & n_{Q,k-n+1} \end{bmatrix}$$

$$= \tilde{A}_k^T \overline{H}_{REAL} + \tilde{B}_k^T \overline{H}_{IMAG} + N_{I,k}$$

$$Y_k^T = \begin{bmatrix} b_k & b_{k-1} & \cdots & b_{k-l-n+2} \end{bmatrix} \begin{bmatrix} h_{REAL}(1) & 0 & 0 & 0 \\ h_{REAL}(2) & h_{REAL}(1) & 0 & 0 \\ h_{REAL}(3) & h_{REAL}(2) & 0 & 0 \\ \cdots & \cdots & \cdots & h_{REAL}(1) \\ h_{REAL}(l) & h_{REAL}(l-1) & 0 & h_{REAL}(2) \\ 0 & h_{REAL}(l) & 0 & h_{REAL}(5) \\ 0 & 0 & 0 & \cdots \\ 0 & 0 & 0 & h_{REAL}(l) \end{bmatrix}$$

$$- \begin{bmatrix} a_k & a_{k-1} & \cdots & a_{k-l-n+2} \end{bmatrix} \begin{bmatrix} h_{IMAG}(1) & 0 & 0 & 0 \\ h_{IMAG}(2) & h_{IMAG}(1) & 0 & 0 \\ h_{IMAG}(3) & h_{IMAG}(2) & 0 & 0 \\ \cdots & \cdots & \cdots & h_{IMAG}(1) \\ h_{IMAG}(l) & h_{IMAG}(l-1) & 0 & h_{IMAG}(2) \\ 0 & h_{IMAG}(l) & 0 & h_{IMAG}(5) \\ 0 & 0 & 0 & \cdots \\ 0 & 0 & 0 & h_{IMAG}(l) \end{bmatrix} + \begin{bmatrix} n_{Q,k} & n_{Q,k-2} & \cdots & n_{Q,k-n+1} \end{bmatrix}$$

$$= \tilde{B}_k^T \overline{H}_{REAL} - \tilde{A}_k^T \overline{H}_{IMAG} + N_{Q,k}$$

The expected value of the squared error, or mean squared error, for a particular Q is

$$E\left[e_k^2\right] = E\left[\left(a_{k-d} - X_k^T Q_I - Y_k^T Q_Q\right)^2\right]$$

The derivative of the mean squared error with respect to the equalizer coefficient vectors Q_I and Q_Q are as follows:

$$\frac{\partial\left(E\left[e_k^2\right]\right)}{\partial Q_I} = -2E\left[\left(a_{k-d} - X_k^T Q_I - Y_k^T Q_Q\right) X_k^T\right]$$

$$\frac{\partial\left(E\left[e_k^2\right]\right)}{\partial Q_Q} = -2E\left[\left(a_{k-d} - X_k^T Q_I - Y_k^T Q^2\right) Y_k^T\right]$$

To obtain the optimal equalizer coefficient vector, Q_{opt}, set the derivative equal to zero, as follows:

$$Q_{I,opt}^T E\left[X_k X_k^T\right] + Q_{Q,opt}^T E\left[X_k Y_k^T\right] = E\left[a_{k-d} X_k^T\right]$$

$$Q_{I,opt}^T E\left[Y_k X_k^T\right] + Q_{Q,opt}^T E\left[Y_k Y_k^T\right] = E\left[a_{k-d} Y_k^T\right]$$

Or in another form as

$$\begin{bmatrix} E\left[X_k X_k^T\right] & E\left[X_k Y_k^T\right] \\ E\left[Y_k X_k^T\right] & E\left[Y_k Y_k^T\right] \end{bmatrix} \begin{bmatrix} Q_{I,opt} \\ Q_{Q,opt} \end{bmatrix} = E\left[a_{k-d} \begin{bmatrix} X_k \\ Y_k \end{bmatrix} \right]$$

The QAM linear equalizer solution is

$$\begin{bmatrix} Q_{I,opt} \\ Q_{Q,opt} \end{bmatrix} = \left(\begin{bmatrix} R_{XX} & R_{XY} \\ R_{YX} & R_{YY} \end{bmatrix} \right)^{-1} \begin{bmatrix} R_{aX} \\ R_{aY} \end{bmatrix}$$

in which

$$R_{XX} = E\left[X_k X_k^T\right] = \overline{H}_{REAL}^T \overline{H}_{REAL} + \overline{H}_{IMAG}^T \overline{H}_{IMAG} + \sigma_{N,I}^2 I$$

$$R_{YY} = E\left[Y_k Y_k^T\right] = \overline{H}_{REAL}^T \overline{H}_{REAL} + \overline{H}_{IMAG}^T \overline{H}_{IMAG} + \sigma_{N,Q}^2 I$$

$$R_{XY} = R_{YX}^T = E\left[X_k Y_k^T\right] = \overline{H}_{REAL}^T \overline{H}_{IMAG} + \overline{H}_{IMAG}^T \overline{H}_{REAL}$$

$$R_{aX} = \left(\overline{H}_{REAL}^T - \overline{H}_{IMAG}^T\right) p$$

$$R_{aY} = \left(\overline{H}_{REAL}^T + \overline{H}_{IMAG}^T\right) p$$

Denoting, $\sigma_{N,I}^2$ and $\sigma_{N,Q}^2$ as noise powers for the in-phase and quadrature channels respectively, we have $\sigma_{N,I}^2 = \sigma_{N,Q}^2$ and $R_{XX} = R_{YY}$ in the normal condition.

Figure 8.24 shows real and imaginary channel impulse response for the 2 mile, 26-gauge twisted-pair loop with a carrier frequency of 200 kHz and a 150 kHz 20 percent excessive bandwidth squared root Nyquist filter $g(t)$. The impulse response of $g(t)$ is expressed as

$$g(t) = \frac{\sin[\pi(1-\alpha)t'] + 4\alpha t' \cos[\pi(1+\alpha)t']}{\pi t'\left[1-(4\alpha t')^2\right]}$$

when $t \neq 0$ and $g(0) = (1-\alpha) + \frac{4\alpha}{\pi}$. Figure 8.23 shows the impulse response of the transmit shaping filter with 200 filter coefficients. Real and imaginary impulse responses can be different for using a different shaping filter $g(t)$.

Figure 8.23 Impulse Response of Squared Root Nyquist Filter

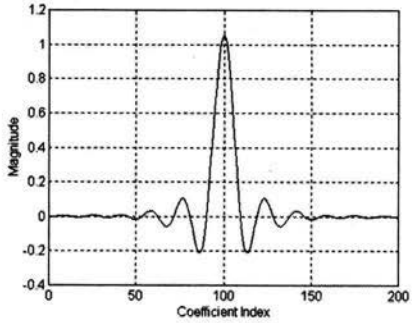
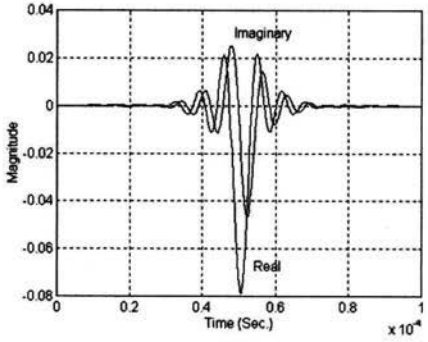


Figure 8.24 Real and Imaginary Channel Impulse Response



Figures 8.25 through 8.27 show QAM in-phase and cross-coupled equalizer coefficient values for 1, 2, and 3 miles of 26-gauge twisted-pair loops, respectively. The length of the passband equalizer is also dependent on the length of the channel impulse response. Required numbers of equalizer coefficients are 15, 20, and 60 for 1, 2, and 3 miles of 26-gauge twisted-pair loops, respectively.

Figure 8.25 Channel Equalizers for a 1 Mile, 26-Gauge Loop

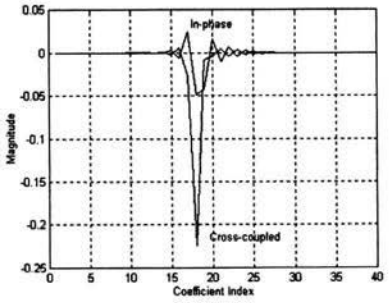


Figure 8.26 Channel Equalizers for a 2 Mile, 26-Gauge Loop

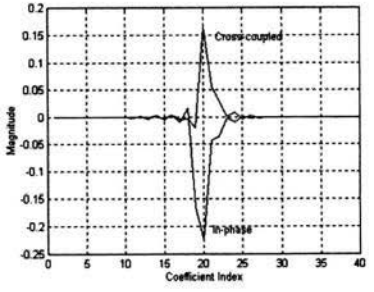
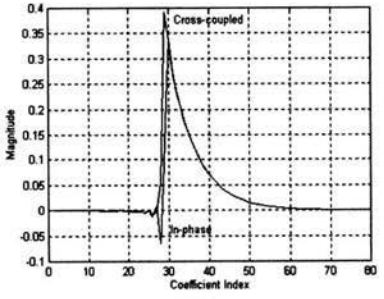


Figure 8.27 Channel Equalizers for a 3 Mile, 26-Gauge Loop



The MMSE at the output of a QAM equalizer is related to the twisted-pair loop channel characteristics, the receiver front-end noise level, and the length of equalizers, and is expressed as

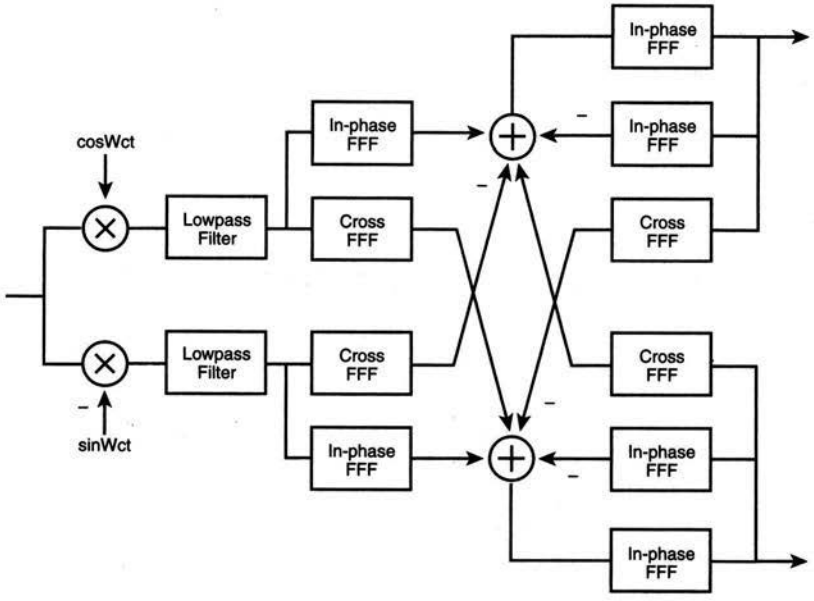
$$\xi_{\min} = E\left[a_k^2\right] - \begin{bmatrix} R_{Xa}^T & R_{Xb}^T \end{bmatrix} \begin{bmatrix} Q_{I,opt} \\ Q_{Q,opt} \end{bmatrix}$$

For the simple example of a 26-gauge, 10,560 foot loop with receiver front-end noise power density levels of –110 dBm/Hz, –120 dBm/Hz, and –130 dBm/Hz, we have MMSE levels of –20.1 dB, –30 dB, and –39.6 dB, respectively.

8.3.3 *Calculation of QAM DFE*

A QAM equalizer also can have a DFE structure consisting of two in-phase FFFs, two cross-coupled FFFs, two in-phase FBFs, and two cross-coupled FBFs, as shown in Figure 8.28. Equalizers for the in-phase and quadrature channels are identical.

Figure 8.28 The Structure of a QAM DFE



Based on the equalizer solution derived for a linear QAM equalizer, we let

$$\left(\tilde{X}_k\right)^T = \begin{bmatrix} X_k^T & A_{k-d-1}^T \end{bmatrix}$$

and

$$\left(\tilde{Y}_k\right)^T = \left[Y_k^T \quad B_{k-d-1}^T\right]$$

to obtain the DFE solution as

$$\begin{bmatrix} Q_{I,opt}^f \\ Q_{Q,opt}^f \\ Q_{I,opt}^b \\ Q_{Q,opt}^b \end{bmatrix} = \begin{bmatrix} R_{XX} & R_{XY} & R_{XA} & R_{XB} \\ R_{YX} & R_{YY} & R_{YA} & R_{YB} \\ R_{XA}^T & R_{YA}^T & I & 0 \\ R_{XB}^T & R_{YB}^T & 0 & I \end{bmatrix}^{-1} \begin{bmatrix} R_{aX} \\ R_{aY} \\ 0 \\ 0 \end{bmatrix}$$

$Q_{I,opt}^f$ is the in-phase FFF.

$Q_{Q,opt}^f$ is the cross-coupled FFF.

$Q_{I,opt}^b$ is the in-phase FBF.

$Q_{Q,opt}^b$ is the cross-coupled FBF.

Figures 8.29 through 8.31 show QAM DFE in-phase and cross-coupled FFF and FBF coefficient values for 1, 2, and 3 miles of 26-gauge twisted-pair loops, respectively.

Figure 8.29 Channel Equalizers for a 1 Mile, 26-Gauge Loop

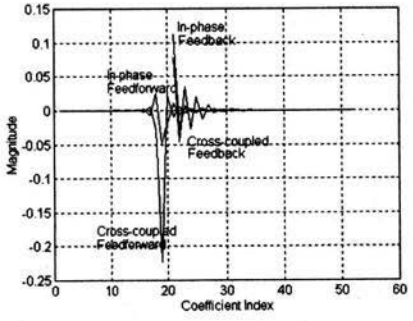


Figure 8.30 Channel Equalizers for a 2 Mile, 26-Gauge Loop

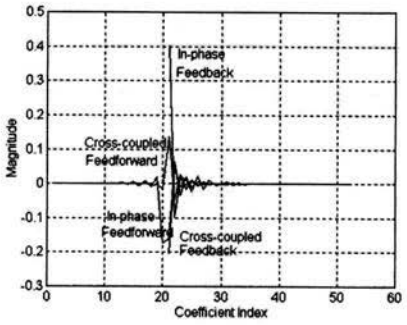
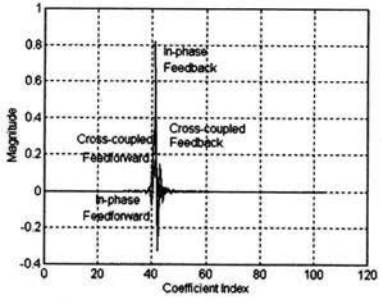


Figure 8.31 Channel Equalizers for a 3 Mile, 26-Gauge Loop



The MMSE at the output of a QAM DFE equalizer is related to the twisted-pair loop channel characteristics, the receiver front-end noise level, and only the length of in-phase and cross-coupled FFFs, and is expressed as

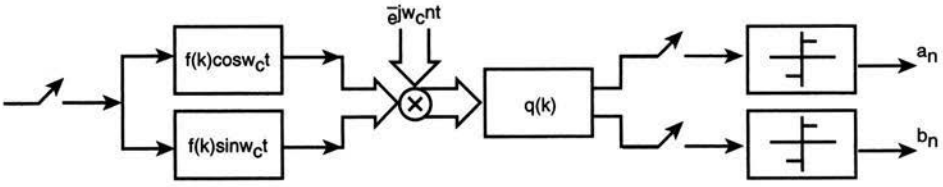
$$\xi_{\min} = E\left[a_k^2\right] - \begin{bmatrix} R_{Xa}^T & R_{Xb}^T \end{bmatrix} \begin{bmatrix} Q_{I,opt}^f \\ Q_{Q,opt}^f \end{bmatrix}$$

A QAM equalizer also can have fractionally spaced linear equalizer or fractionally spaced FFF. Equalizer solutions can be derived based on those of baud rate linear equalizer and baud rate DFE by replacing the single-phase, received-signal vectors with multiple-phased, received-signal vectors.

8.3.4 *From QAM to CAP*

The order of demodulation and lowpass filtering also can be exchanged in the case of using channel equalizers for a QAM system resulting to a pair of passband filters followed by a phase derotator, as depicted in Figure 8.32. A fractionally spaced QAM equalizer is required after the phase derotator.

Figure 8.32 QAM Demodulation After Passband Filtering



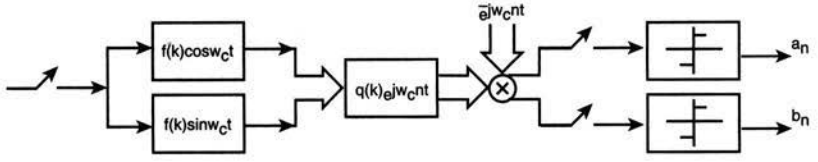
The passband filtering, demodulation, and linear equalization process for the recovery of data symbols are expressed as

$$\hat{a}_i + j\hat{b}_i = \left(f(t) * \left(g(t)e^{j\omega_c t}\right)\right)e^{-j\omega_c t} * Q_{IQ}(t)$$

By again changing the order of filtering and convolution and its mathematical equivalence, we can derive a QAM transceiver with passband equalizers, as shown in Figure 8.33. The passband filtering, passband equalization, and phase derotation process for the recovery of data symbols is expressed as

$$\hat{a}_i + j\hat{b}_i = \left[f(t) * \left(g(t)e^{j\omega_c t}\right) * Q_{IQ}(t)e^{j\omega_c t}\right]e^{-j\omega_c t}$$

Figure 8.33 A QAM Receiver with Passband Equalizers



We recognize that the received signal is first filtered by a pair of passband filters and then by a pair of passband equalizers. Therefore, the function of passband filtering and passband equalization can be further combined in the following expression:

$$\hat{a}_i + j\hat{b}_i = [f(t) * Q(t)]e^{-j\omega_c t}$$

The combined equalizers are the result of the convolution of filters and equalizers. Specifically, a pair of in-phase and quadrature passband equalizers can be used in place of a pair of passband filters followed by passband channel equalizers, as follows:

$$Q(t) = \left(g(t)e^{j\omega_c t}\right) * \left(Q_{IQ}(t)e^{j\omega_c t}\right) = Q_{in}(t) + jQ_{qu}(t)$$

This passband equalization structure has been studied and implemented for many practical QAM systems.[16,17,18]

A Carrierless AM/PM (CAP) system is the result of removing the phase rotator and the phase derotator from a QAM system with passband transmitter filters and passband receiver filters or passband equalizers. A CAP transmit signal is generated with a pair of passband shaping filters whose impulse response is the product of a baseband shaping filter and the in-phase or quadrature phase of the center frequency. The passband shaping filtering process for the generation of a CAP line signal is expressed as

$$f(t) = \mathrm{Re}\left[C_i g(t - iT)e^{j\omega_c(t-iT)}\right]$$

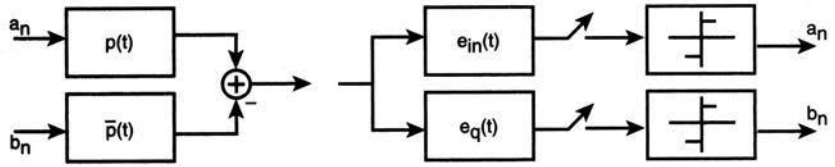
For a distortionless channel, the data symbols can be recovered with the same pair of passband shaping filters. The passband filter process for the recovery of data symbols carried by a CAP line code is expressed as

$$\hat{a}_i + j\hat{b}_i = \left(f(t) * \left(g(t)e^{j\omega_c t}\right)\right)$$

The receiver passband shaping filters need to be replaced with a pair of in-phase and quadrature passband equalizers for a twisted-pair loop channel, as shown in Figure 8.34. The equalizers are the same as those in a direct passband-implemented QAM system. The passband equalization process for the recovery of data symbols carried by a CAP line code is expressed as

$$\hat{a}_i + j\hat{b}_i = [f(t) * Q(t)]$$

Figure 8.34 A CAP Transceiver Pair Resulting from the Removal of Phase Modulator and Demodulator



8.3.5 *Calculation of CAP Linear Equalizer*

CAP equalizer solutions are derived for a three times over-sampled implementation as an example. Instantaneous errors between desired in-phase and quadrature data symbols and outputs from a pair of in-phase and quadrature passband equalizers are

$$e_{ik} = a_{k-d} - X_k^{1T}Q_i^1 - X_k^{2T}Q_i^2 - X_k^{3T}Q_i^3$$

$$e_{qk} = b_{k-d} - X_k^{1T}Q_q^1 - X_k^{2T}Q_q^2 - X_k^{3T}Q_q^3$$

The values are as follows:

a_{k-d} is the desired in-phase data symbol after d baud intervals of channel delay.

$a_{k-d}b_k$ is the desired quadrature data symbol also after d baud intervals of channel delay.

$X_k^{1T} = \begin{bmatrix} x_k & x_{k-3} & \cdots & x_{k-n+3} \end{bmatrix}$ is the first phase of the received-signal vector.

$X_k^{2T} = \begin{bmatrix} x_{k-1} & x_{k-3-1} & \cdots & x_{x-n+2} \end{bmatrix}$ is the second phase of the received-signal vector.

$X_k^{3T} = \begin{bmatrix} x_{k-2} & x_{k-3-2} & \cdots & x_{x-n+1} \end{bmatrix}$ is the third phase of the received-signal vector.

$Q_i^{1T} = \begin{bmatrix} q_{i1} & a_{i4} & \cdots & a_{i(n-2)} \end{bmatrix}$ is the first phase of the in-phase channel equalizer filter coefficient vector.

$Q_i^{2T} = \begin{bmatrix} q_{i2} & a_{i5} & \cdots & a_{i(n-1)} \end{bmatrix}$ is the second phase of the in-phase channel equalizer filter coefficient vector.

$Q_i^{3T} = \begin{bmatrix} q_{i3} & a_{i6} & \cdots & a_{in} \end{bmatrix}$ is the third phase of the in-phase channel equalizer filter coefficient vector.

$Q_q^{1T} = \begin{bmatrix} q_{q1} & a_{i4} & \cdots & a_{q(n-2)} \end{bmatrix}$ is the first phase of the quadrature channel equalizer filter coefficient vector.

$Q_q^{2T} = \begin{bmatrix} q_{q2} & a_{q5} & \cdots & a_{q(n-1)} \end{bmatrix}$ is the second phase of the quadrature channel equalizer filter coefficient vector.

$Q_q^{3T} = \begin{bmatrix} q_{q3} & a_{q6} & \cdots & a_{qn} \end{bmatrix}$ is the third phase of the quadrature channel equalizer filter coefficient vector.

e_{ik} is the error between the desired in-phase data symbol and the in-phase channel equalizer output.

e_{qk} is the error between the desired quadrature data symbol and the quadrature channel equalizer output.

The received signal can be expressed as

$$x_k = \sum_{i=1}^{l} a_{k-i+1} h_I(1+3(i-1)) + \sum_{i=1}^{l} b_{k-i+1} h_Q(1+3(i-1)) + n_{I,k}$$

$$= \begin{bmatrix} a_k & a_{k-1} & \cdots & a_{k-l+1} \end{bmatrix} \begin{bmatrix} h_I(1) \\ h_I(4) \\ \cdots \\ h_I(l-2) \end{bmatrix} + \begin{bmatrix} b_k & b_{k-1} & \cdots & b_{k-l+1} \end{bmatrix} \begin{bmatrix} h_Q(1) \\ h_Q(4) \\ \cdots \\ h_Q(l-2) \end{bmatrix} + n_{I,k}$$

for k=1, 4, 7...

$$x_k = \sum_{i=1}^{l} a_{k-i+1} h_I(2+3(i-1)) + \sum_{i=1}^{l} b_{k-i+1} h_Q(2+3(i-1)) + n_{I,k}$$

$$= \begin{bmatrix} a_k & a_{k-1} & \cdots & a_{k-l+1} \end{bmatrix} \begin{bmatrix} h_I(2) \\ h_I(5) \\ \cdots \\ h_I(l-1) \end{bmatrix} + \begin{bmatrix} b_k & b_{k-1} & \cdots & b_{k-l+1} \end{bmatrix} \begin{bmatrix} h_Q(2) \\ h_Q(5) \\ \cdots \\ h_Q(l-1) \end{bmatrix} + n_{I,k}$$

for k=2, 5, 8...

$$x_k = \sum_{i=1}^{l} a_{k-i+1} h_I(3+3(i-1)) + \sum_{i=1}^{l} b_{k-i+1} h_Q(3+3(i-1)) + n_{I,k}$$

$$= \begin{bmatrix} a_k & a_{k-1} & \cdots & a_{k-l+1} \end{bmatrix} \begin{bmatrix} h_I(3) \\ h_I(6) \\ \cdots \\ h_I(l) \end{bmatrix} + \begin{bmatrix} b_k & b_{k-1} & \cdots & b_{k-l+1} \end{bmatrix} \begin{bmatrix} h_Q(3) \\ h_Q(6) \\ \cdots \\ h_Q(l) \end{bmatrix} + n_{I,k}$$

for k=3, 6, 9... in which

$$h_I(t) = \left(g(t)\cos\varpi_c t\right) * h(t)$$

and

$$h_Q(t) = \left(g(t)\sin\varpi_c t\right) * h(t)$$

The three phases of received signal vectors are expressed as

$$X_k^{1T} = \tilde{A}_k^T \overline{H}_I^1 + \tilde{B}_k^T \overline{H}_Q^1 + N_k$$

$$X_k^{2T} = \tilde{A}_k^T \overline{H}_I^2 + \tilde{B}_k^T \overline{H}_Q^2 + N_k$$

$$X_k^{3T} = \tilde{A}_k^T \overline{H}_I^3 + \tilde{B}_k^T \overline{H}_Q^3 + N_k$$

for which

$$\overline{H}_I^1 = \begin{bmatrix} h_I(1) & 0 & 0 & 0 \\ h_I(4) & h_I(1) & 0 & 0 \\ h_I(7) & h_I(4) & 0 & 0 \\ \cdots & \cdots & \cdots & h_I(1) \\ h_I(l-2) & h_I(l-5) & 0 & h_I(4) \\ 0 & h_I(l-2) & 0 & h_I(7) \\ 0 & 0 & 0 & \cdots \\ 0 & 0 & 0 & h_I(l-2) \end{bmatrix}$$

$$\overline{H}_I^2 = \begin{bmatrix} h_I(2) & 0 & 0 & 0 \\ h_I(5) & h_I(2) & 0 & 0 \\ h_I(8) & h_I(5) & 0 & 0 \\ \cdots & \cdots & \cdots & h_I(2) \\ h_I(l-1) & h_I(l-4) & 0 & h_I(5) \\ 0 & h_I(l-1) & 0 & h_I(8) \\ 0 & 0 & 0 & \cdots \\ 0 & 0 & 0 & h_I(l-1) \end{bmatrix}$$

$$\overline{H}_I^3 = \begin{bmatrix} h_I(3) & 0 & 0 & 0 \\ h_I(6) & h_I(3) & 0 & 0 \\ h_I(9) & h_I(6) & 0 & 0 \\ \cdots & \cdots & \cdots & h_I(3) \\ h_I(l) & h_I(l-3) & 0 & h_I(6) \\ 0 & h_I(l) & 0 & h_I(9) \\ 0 & 0 & 0 & \cdots \\ 0 & 0 & 0 & h_I(l) \end{bmatrix}$$

$$\overline{H}_Q^1 = \begin{bmatrix} h_Q(1) & 0 & 0 & 0 \\ h_Q(4) & h_Q(1) & 0 & 0 \\ h_I(7) & h_Q(4) & 0 & 0 \\ \cdots & \cdots & \cdots & h_Q(1) \\ h_Q(l-2) & h_Q(l-5) & 0 & h_Q(4) \\ 0 & h_Q(l-2) & 0 & h_Q(7) \\ 0 & 0 & 0 & \cdots \\ 0 & 0 & 0 & h_Q(l-2) \end{bmatrix}$$

$$\overline{H}_Q^2 = \begin{bmatrix} h_Q(2) & 0 & 0 & 0 \\ h_Q(5) & h_Q(2) & 0 & 0 \\ h_Q(8) & h_Q(5) & 0 & 0 \\ \cdots & \cdots & \cdots & h_Q(2) \\ h_Q(l-1) & h_Q(l-4) & 0 & h_Q(5) \\ 0 & h_Q(l-1) & 0 & h_Q(8) \\ 0 & 0 & 0 & \cdots \\ 0 & 0 & 0 & h_Q(l-1) \end{bmatrix}$$

$$\overline{H}_Q^3 = \begin{bmatrix} h_Q(3) & 0 & 0 & 0 \\ h_Q(6) & h_Q(3) & 0 & 0 \\ h_Q(9) & h_Q(6) & 0 & 0 \\ \cdots & \cdots & \cdots & h_Q(3) \\ h_Q(l) & h_Q(l-3) & 0 & h_Q(6) \\ 0 & h_Q(l) & 0 & h_Q(9) \\ 0 & 0 & 0 & \cdots \\ 0 & 0 & 0 & h_Q(l) \end{bmatrix}$$

The expected value of square errors or mean square errors for a particular set of equalizer coefficient vectors, Q_i and Q_q, are as follows:

$$E\left[e_{ik}^2\right] = E\left[\left(a_{k-d} - X_k^{1T}Q_i^1 - X_k^{2T}Q_i^2 - X_k^{1T}Q_i^3\right)^2\right]$$

$$E\left[e_{qk}^2\right] = E\left[\left(b_{k-d} - X_k^{1T}Q_q^1 - X_k^{2T}Q_q^2 - X_k^{1T}Q_q^3\right)^2\right]$$

Derivatives of the mean square error with respect to the equalizer coefficient vectors Q_i^1, Q_i^2, Q_i^3, Q_q^1, Q_q^2, and Q_q^3 are

$$\frac{\partial\left(E\left[e_{ik}^2\right]\right)}{\partial Q_i^1} = -2E\left[\left(a_{k-d} - X_k^{1T}Q_i^1 - X_k^{2T}Q_i^2 - X_k^{3T}Q_i^3\right)X_k^{1T}\right]$$

$$\frac{\partial\left(E\left[e_{ik}^2\right]\right)}{\partial Q_i^2} = -2E\left[\left(a_{k-d} - X_k^{1T}Q_i^1 - X_k^{2T}Q_i^2 - X_k^{3T}Q_i^3\right)X_k^{2T}\right]$$

$$\frac{\partial\left(E\left[e_{ik}^2\right]\right)}{\partial Q_i^3} = -2E\left[\left(a_{k-d} - X_k^{1T}Q_i^1 - X_k^{2T}Q_i^2 - X_k^{3T}Q_i^3\right)X_k^{3T}\right]$$

$$\frac{\partial\left(E\left[e_{qk}^2\right]\right)}{\partial Q_q^1} = -2E\left[\left(b_{k-d} - X_k^{1T}Q_q^1 - X_k^{2T}Q_q^2 - X_k^{3T}Q_q^3\right)X_k^{1T}\right]$$

$$\frac{\partial\left(E\left[e_{qk}^2\right]\right)}{\partial Q_q^2} = -2E\left[\left(b_{k-d} - X_k^{1T}Q_q^1 - X_k^{2T}Q_q^2 - X_k^{3T}Q_q^3\right)X_k^{2T}\right]$$

$$\frac{\partial\left(E\left[e_{qk}^2\right]\right)}{\partial Q_q^3} = -2E\left[\left(b_{k-d} - X_k^{1T}Q_q^1 - X_k^{2T}Q_q^2 - X_k^{3T}Q_q^3\right)X_k^{3T}\right]$$

Setting derivatives equal to zero, we obtain the optimal equalizer coefficient vectors Q_i^1, Q_i^2, Q_i^3, Q_q^1, Q_q^2, and Q_q^3 for Minimum Mean Square Errors, as

$$Q_{i,opt} = \begin{bmatrix} Q_{i,opt}^1 \\ Q_{i,opt}^2 \\ Q_{i,opt}^3 \end{bmatrix} = \begin{bmatrix} R_{XX}^{11} & R_{XX}^{12} & R_{XX}^{13} \\ R_{XX}^{21} & R_{XX}^{22} & R_{XX}^{23} \\ R_{XX}^{31} & R_{XX}^{32} & R_{XX}^{33} \end{bmatrix}^{-1} \begin{bmatrix} R_{Xa}^1 \\ R_{Xa}^2 \\ R_{Xa}^3 \end{bmatrix} = \left(R_{XX}\right)^{-1} R_{Xa}$$

$$Q_{q,opt} = \begin{bmatrix} Q_{q,opt}^1 \\ Q_{q,opt}^2 \\ Q_{q,opt}^3 \end{bmatrix} = \begin{bmatrix} R_{XX}^{11} & R_{XX}^{12} & R_{XX}^{13} \\ R_{XX}^{21} & R_{XX}^{22} & R_{XX}^{23} \\ R_{XX}^{31} & R_{XX}^{32} & R_{XX}^{33} \end{bmatrix}^{-1} \begin{bmatrix} R_{Xb}^1 \\ R_{Xb}^2 \\ R_{Xb}^3 \end{bmatrix} = \left(R_{XX}\right)^{-1} R_{Xb}$$

for which

$$R_{XX}^{11} = E\left[X_k^1 X_k^{1T}\right]$$

$$R_{XX}^{22} = E\left[X_k^2 X_k^{2T}\right]$$

$$R_{XX}^{33} = E\left[X_k^3 X_k^{3T}\right]$$

$$R_{XX}^{12} = \left(R_{XX}^{21}\right)^T = E\left[X_k^1 X_k^{2T}\right]$$

$$R_{XX}^{13} = \left(R_{XX}^{31}\right)^T = E\left[X_k^1 X_k^{3T}\right]$$

$$R_{Xa}^1 = E\left[a_k X_k^1\right]$$

$$R_{Xa}^2 = E\left[a_k X_k^2\right]$$

$$R_{Xa}^3 = E\left[a_k X_k^3\right]$$

$$R_{Xb}^1 = E\left[b_k X_k^1\right]$$

$$R_{Xb}^2 = E\left[b_k X_k^2\right]$$

$$R_{Xb}^3 = E\left[b_k X_k^3\right]$$

Figure 8.35 shows CAP in-phase and quadrature passband shaping filter impulse responses for a center frequency of 166.67 kHz, a baud rate of 250 kHz, and an excessive bandwidth of 20 percent.

Figure 8.35 In-Phase and Quadrature Transmit Filter Coefficients

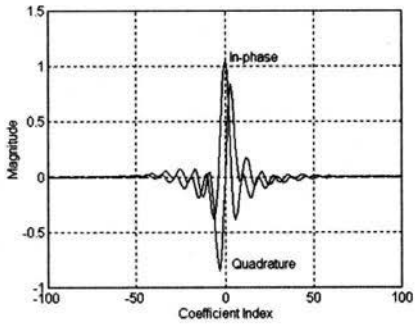
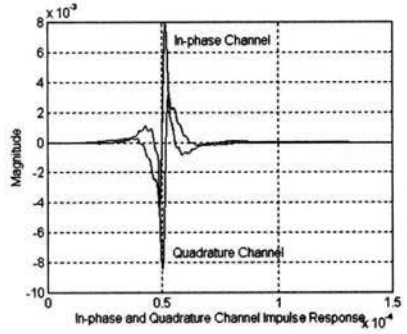


Figure 8.36 shows possible in-phase and quadrature channel impulse responses for a twisted-pair loop.

Figure 8.36 In-Phase and Quadrature Channel Impulse Responses



Figures 8.37 through 8.39 show fractionally spaced CAP equalizer coefficient values with three times of oversampling for 1, 2, and 3 miles, respectively, of 26-gauge twisted-pair loops.

Figure 8.37 Channel Equalizers for a 1 Mile, 26-Gauge Loop

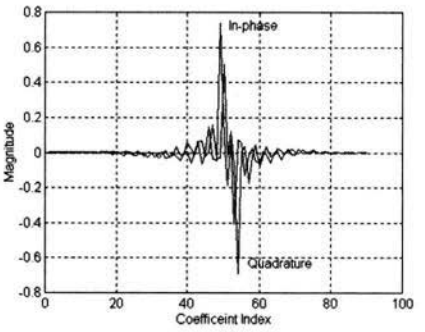


Figure 8.38 Channel Equalizers for a 2 Mile, 26-Gauge Loop

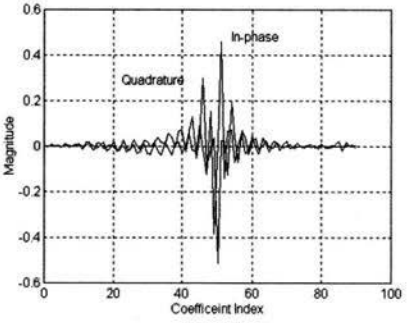
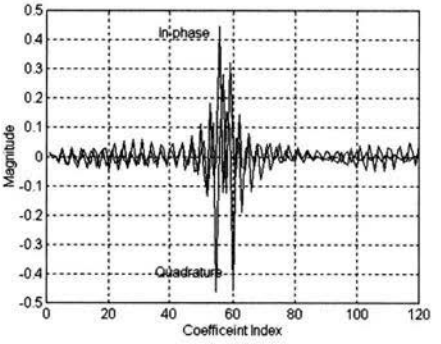


Figure 8.39 Channel Equalizers for a 3 Mile, 26-Gauge Loop



The MMSE at the output of a CAP linear equalizer is related to the twisted-pair loop channel characteristics, the receiver front-end noise level, and the length of equalizers. In-phase and Quadrature MMSEs are expressed separately as

$$\xi_{iin,\min} = E\left[a_k^2\right] - \begin{bmatrix} R_{Xa}^{1T} & R_{Xa}^{2T} & R_{Xa}^{3T} \end{bmatrix} \begin{bmatrix} Q_{i,opt}^1 \\ Q_{i,opt}^2 \\ Q_{i,opt}^3 \end{bmatrix}$$

$$\xi_{qu,\min} = E\left[b_k^2\right] - \begin{bmatrix} R_{Xb}^{1T} & R_{Xb}^{2T} & R_{Xb}^{3T} \end{bmatrix} \begin{bmatrix} Q_{q,opt}^1 \\ Q_{q,opt}^2 \\ Q_{q,opt}^3 \end{bmatrix}$$

8.3.6 *Calculation of CAP DFE*

You can derive CAP DFE equalizer coefficients as well as its structure by expanding the length of received-signal vectors and let one phase of the expanded portion of the data vector contain the estimated data symbols and expanded portions of two other received signal vectors filled with zeros. For an example of using a phase to contain the estimated data symbols, we have

$$\left(\tilde{X}_k^1\right)^T = \left[\left(X_k^1\right)^T \quad [0 \quad 0 \quad \ldots \quad 0]^T\right]$$

$$\left(\tilde{X}_k^2\right)^T = \left[\left(X_k^2\right)^T \quad [0 \quad 0 \quad \ldots \quad 0]^T\right]$$

and

$$\left(\tilde{X}_k^3\right)^T = \left[\left(X_k^3\right)^T \quad A_{k-d-1}^T\right]$$

or

$$\left(\tilde{X}_k^3\right)^T = \left[\left(X_k^3\right)^T \quad B_{k-d-1}^T\right]$$

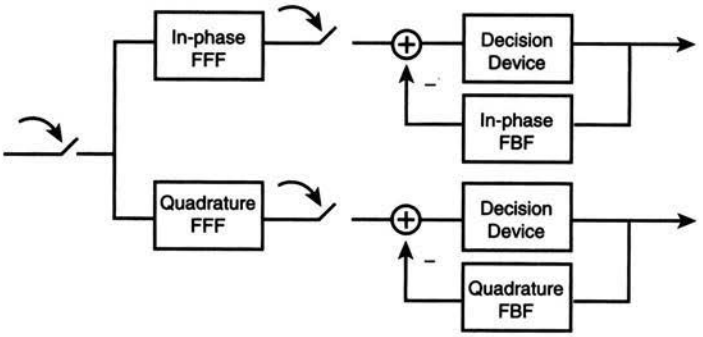
We have the CAP DFE equalizer solution as

$$\begin{bmatrix} Q^1_{i,opt} \\ Q^2_{i,opt} \\ Q^3_{i,opt} \\ Q^b_{i,opt} \end{bmatrix} = \left(\begin{bmatrix} E\left[X^1_k X^{1T}_k\right] & E\left[X^1_k X^{2T}_k\right] & E\left[X^1_k X^{3T}_k\right] & E\left[X^1_k \left(A^m_{k-d-1}\right)^T\right] \\ E\left[X^2_k X^{1T}_k\right] & E\left[X^2_k X^{2T}_k\right] & E\left[X^2_k X^{3T}_k\right] & E\left[X^2_k \left(A^m_{k-d-1}\right)^T\right] \\ E\left[X^3_k X^{1T}_k\right] & E\left[X^3_k X^{2T}_k\right] & E\left[X^3_k X^{3T}_k\right] & E\left[X^3_k \left(A^m_{k-d-1}\right)^T\right] \\ E\left[A^m_{k-d-1}\left(X^1_k\right)^T\right] & E\left[A^m_{k-d-1}\left(X^2_k\right)^T\right] & E\left[A^m_{k-d-1}\left(X^3_k\right)^T\right] & I \end{bmatrix} \right)^{-1} E\left[a_{k-d} \begin{bmatrix} X^1_k \\ X^2_k \\ X^3_k \\ 0 \end{bmatrix} \right]$$

$$\begin{bmatrix} Q^1_{i,opt} \\ Q^2_{q,opt} \\ Q^3_{q,opt} \\ Q^b_{q,opt} \end{bmatrix} = \left(\begin{bmatrix} E\left[X^1_k X^{1T}_k\right] & E\left[X^1_k X^{2T}_k\right] & E\left[X^1_k X^{3T}_k\right] & E\left[X^1_k \left(B^m_{k-d-1}\right)^T\right] \\ E\left[X^2_k X^{1T}_k\right] & E\left[X^2_k X^{2T}_k\right] & E\left[X^2_k X^{3T}_k\right] & E\left[X^2_k \left(B^m_{k-d-1}\right)^T\right] \\ E\left[X^3_k X^{1T}_k\right] & E\left[X^3_k X^{2T}_k\right] & E\left[X^3_k X^{3T}_k\right] & E\left[X^3_k \left(B^m_{k-d-1}\right)^T\right] \\ E\left[B^m_{k-d-1}\left(X^1_k\right)^T\right] & E\left[B^m_{k-d-1}\left(X^2_k\right)^T\right] & E\left[B^m_{k-d-1}\left(X^3_k\right)^T\right] & I \end{bmatrix} \right)^{-1} E\left[b_{k-d} \begin{bmatrix} X^1_k \\ X^2_k \\ X^3_k \\ 0 \end{bmatrix} \right]$$

A CAP DFE consists of two, in-phase and quadrature, separate DFEs, as shown in Figure 8.40. Cross-coupled FBFs are not necessary. The orthogonality between in-phase and quadrature channels is maintained in the transmitter by these passband shaping filters.

Figure 8.40 The Structure of a CAP DFE



Figures 8.41 through 8.43 show CAP DFE equalizer coefficients for 1, 2, and 3 miles, respectively, of 26-gauge twisted-pair loops.

Figure 8.41 Channel Equalizers for a 1 Mile, 26-Gauge Loop

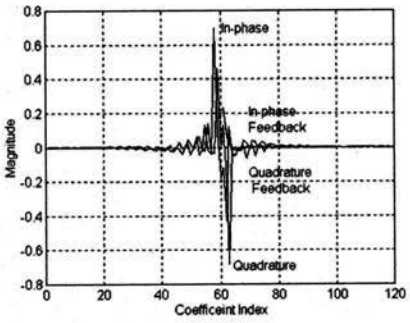


Figure 8.42 Channel Equalizers for a 2 Mile, 26-Gauge Loop

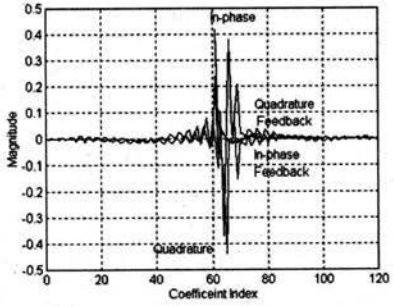
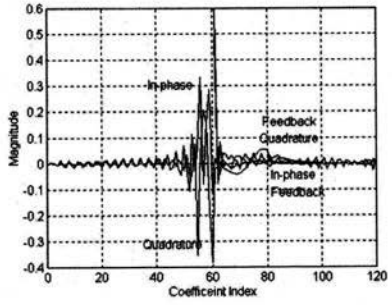


Figure 8.43 Channel Equalizers for a 3 Mile, 26-Gauge Loop



Because FBFs does not contribute to the noise enhancement, MMSEs for a CAP DFE system are the same as those described for the CAP linear equalizer system. The additional performance improvement comes from a different set of CAP FFF filter coefficients:

$$\xi_{iin,\min} = E\left[a_k^2\right] - \begin{bmatrix} R_{Xa}^{1T} & R_{Xa}^{2T} & R_{Xa}^{3T} \end{bmatrix} \begin{bmatrix} Q_{i,opt}^{1} \\ Q_{i,opt}^{2} \\ Q_{i,opt}^{3} \end{bmatrix}$$

$$\xi_{qu,\min} = E\left[b_k^2\right] - \begin{bmatrix} R_{Xb}^{1T} & R_{Xb}^{2T} & R_{Xb}^{3T} \end{bmatrix} \begin{bmatrix} Q_{q,opt}^{1} \\ Q_{q,opt}^{2} \\ Q_{q,opt}^{3} \end{bmatrix}$$

Equalizer output SNRs for QAM and CAP systems are summarized in Table 8.2 for different equalizer structures under different receiver front-end noise power density levels for the 2 mile, 26-gauge twisted-pair loop. Results show that DFE performance is only slightly better for the passband system. However, a QAM equalizer with a fractionally spaced FFF could make performance robustness irrelevant to the sample timing phase.

Performances of a QAM system and a CAP system with fractionally spaced FFFs should be about the same. The difference in the table is caused by the sample timing phase of the channel because the QAM system used a two times over-sampled FFF and the CAP system used a three times over-sampled FFF.

Table 8.2 Passband Equalizer Output SNRs

Noise Level	QAM Linear	QAM DFE	Fractional QAM Linear	CAP Linear	CAP DFE
–110 dBm/Hz	22.0	22.2	25	26.0	26.8
–120 dBm/Hz	32.0	32.2	35	36.0	36.8
–130 dBm/Hz	42.0	42.2	45	45.9	46.7

8.4 *Equalizers for Discrete Multitone Systems*

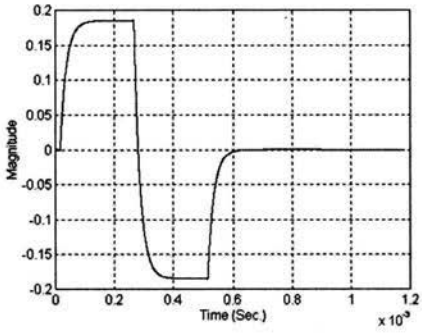
The severe channel intersymbol interference of a twisted-pair loop also can be avoided by signaling at a lower baud rate while letting each baud carry many bits of information. For a 1 mile, 26-gauge twisted-pair loop, the tail of the impulse response is about 30 microseconds. To make the intersymbol interference negligible, the signaling baud interval can be chosen as 100 times that of the tail. This results in a signaling baud rate of 333 Hz.

Data can be carried in this long baud interval with many waveforms of different frequencies. In other words, bits of information are encoded in the frequency domain. This information encoding process can be carried out with the *Inverse Discrete Fourier Transform* (IDFT). Bits of information are recovered in a receiver with the *Discrete Fourier Transform* (DFT). This approach has been called *Discrete MultiTone* (DMT) line code for the use of multiple numbers of carriers. QAM and CAP are called *single-carrier systems.*

A slower baud rate can lead to less intersymbol interference. On the other hand, a slower baud rate requires a higher number of carriers for certain desired transmission throughputs. This in turn results in a bigger IDFT/DFT size. The implementation of a large size IDFT/DFT requires more buffering memories and a longer process delay time.

As an alternative for DMT-based systems, a higher baud rate can be used also, thus lowering the size of the IDFT/DFT requirement. The intersymbol interference at a particular baud rate also can be minimized by the introduction of a guard period where repetitive information is encoded.[19] Figure 8.44 shows the relationship between a long duration baud interval and the tail of the impulse response.

Figure 8.44 Baud Interval Pulse and Residual Tail



Incidentally, the duration of the impulse response tail also can be reduced using the general channel equalization technique. The idea is to use a linear channel equalizer in the receiver front end of a DMT system to minimize the MMSE between the linear equalizer output and an unknown channel with a desired impulse response tail. The solution of this DMT time-domain equalizer has been examined by many studies[20,21] and for other receiver structures.[22,23]

The DMT time-domain equalizer solution also can be derived in the discrete time domain. We assume that the equalized unknown channel is the convolution of the channel and the equalizer and can be expressed as

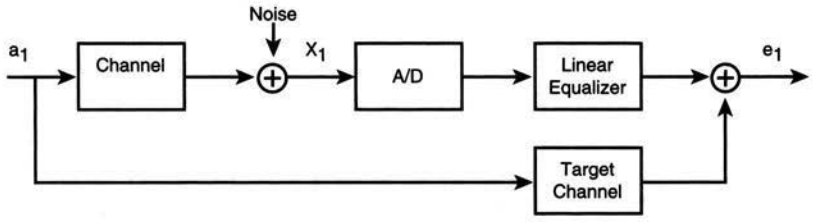
$$h_{equ}(k) = h(k) * Q(k)$$

To constrain the length of the equalized unknown channel, for an example of an original channel length of 5, an equalizer length of 5, and a equalized channel length of 3, we need to have

$$\begin{bmatrix} h(1) & 0 & 0 & 0 & 0 \\ h(2) & h(1) & 0 & 0 & 0 \\ h(3) & h(2) & h(1) & 0 & 0 \\ h(4) & h(3) & h(2) & h(1) & 0 \\ h(5) & h(4) & h(3) & h(2) & h(1) \\ 0 & h(5) & h(4) & h(3) & h(2) \\ 0 & 0 & h(5) & h(4) & h(3) \\ 0 & 0 & 0 & h(5) & h(4) \\ 0 & 0 & 0 & 0 & h(5) \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \\ q_5 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ h_{equ}(1) \\ h_{equ}(2) \\ h_{equ}(3) \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

In general, the equalized unknown channel can have an infinite number of solutions. The optimal equalizer solution is called the target channel, which produces the MMSE in conjunction with the equalized channel. Figure 8.45 illustrates the mean square error between the output of a DMT time domain linear channel equalizer and the unknown target channel. The size of the target channel of a DMT system is usually defined by the guard period of the DMT system, and that is normally part of the design parameters.

Figure 8.45 Equalizer and Target Channel



For the special case of a casual target channel, the solution to this mean square error minimization problem is the decision feedback channel equalizer. Therefore, the linear equalizer and the target channel for this special case can be solved as

$$\begin{bmatrix} Q_{opt} \\ \bar{h}_{equ} \end{bmatrix} = \begin{bmatrix} R_{XX} & R_{XA} \\ R_{XA}^T & I \end{bmatrix}^{-1} \begin{bmatrix} R_{Xa} \\ 0 \end{bmatrix}$$

The values are as follows:

$$R_{XX} = E\left[X_k X_k^T\right]$$

$$R_{XA} = E\left[X_k A_{k-d-1}^T\right]$$

$$R_{Xa} = E\left[X_k a_{k-d}\right]$$

For the general case of a DMT time-domain linear channel equalizer, we remove the constraint of a casual target channel. Therefore, the data symbol vector can have the following form:

$$A_{k-i}^T = \left[a_{k-i} \quad \cdots \quad a_{k-d+1} \quad a_{k-d-1} \quad \cdots \quad a_{k-i-l+1}\right]$$

The data symbol vector can have data symbols before or after, but not include the currently desired data symbol within a certain time window. The mean square error between the output of a DMT time-domain linear equalizer and the target channel is

$$\xi_{\min} = E\left[a_k^2\right] - R_{XA}^T Q_{opt}^i$$

Q_{opt}^i stands for the equalizer solution corresponding to the symbol vector A_{k-i}^T.

The optimal DMT time-domain linear channel equalizer solution is chosen such that its corresponding mean square error is minimal:

$$\underset{i}{Min}\left(\xi_{\min}\right) = \underset{i}{Min}\left(E\left[a_k^2\right] - R_{Xa}^T Q_{opt}^i\right)$$

for $d - l < i < d + 1$ among different A_{k-i}^Ts.

Figures 8.46 through 8.51 show MMSEs corresponding to different A_{k-i}^{T}s, the optimal DMT time-domain linear equalizer coefficients, and the target channel impulse responses for 1, 2, and 3 miles, respectively, of 26-gauge twisted-pair loops.

Figure 8.46 Mean Square Errors for a 1 Mile, 26-Gauge Loop

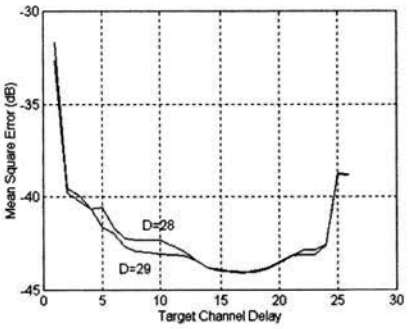


Figure 8.47 Equalizer Coefficient and Target Channel Impulse Response

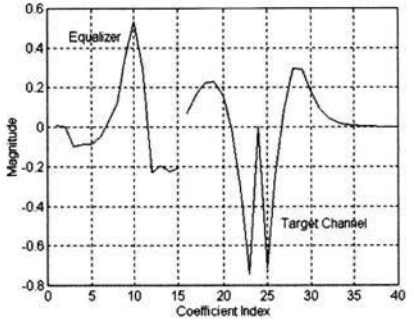


Figure 8.48 Mean Square Errors for a 2 Mile, 26-Gauge Loop

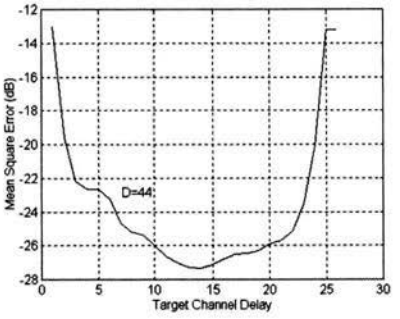


Figure 8.49 Equalizer Coefficient and Target Channel Impulse Response

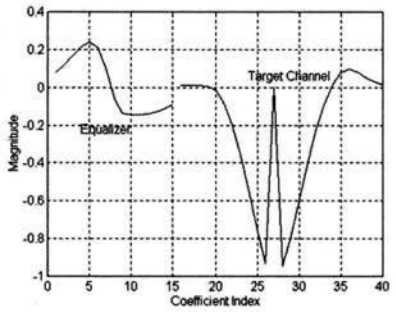


Figure 8.50 Mean Square Errors for a 3 Mile, 26-Gauge Loop

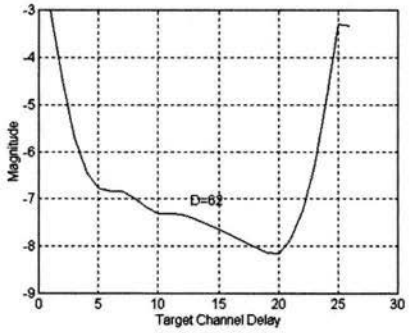
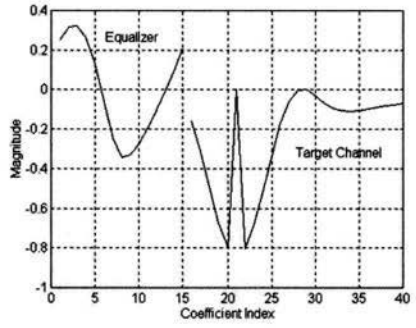


Figure 8.51 Equalizer Coefficient and Target Channel Impulse Response



It should be pointed out that the optimal DMT time-domain linear equalizer is not necessarily obtained based on this MMSE criteria. An optimal DMT linear channel equalizer can

be defined to maximize the transmission throughput under certain error rate requirements in conjunction with a given channel and noise environment.[24]

End Notes

1. R. W. Lucky, "Automatic Equalization for Digital Communication," *B.S.T.J.*, vol. 44, no. X (April 1965): 547–588.
2. A. Papoulis, *Signal Analysis* (McGraw-Hill Book Company, 1977).
3. D. W. Lin, "MMSE Echo Cancellation and Equalization for Digital Subscriber Line Transmission: Part I—Theory and Computation," *IEEE Trans. on Commun.*, vol. 38, no. 1 (January 1990): 31–38.
4. B. Widrow and S. D. Stearns, *Adaptive Signal Processing* (Prentice-Hall 1985).
5. R. D. Gitlin and S. B. Weinstein, "Fractionally-Spaced Equalization: An Improved Digital Transversal Equalizer," *B.S.T.J.*, vol. 60, no. 2 (February 1981): 275–296.
6. R. D. Gitlin, H. C. Meadors, Jr., and S. B. Weinstein, "The Tap-Leakage Algorithm: An Algorithm for the Stable Operation of a Digitally Implemented, Fractionally Spaced Adaptive Equalizer," *B.S.T.J.*, vol. 61, no. 8 (October 1982).
7. E. A. Lee and D. G. Messerschmitt, *Digital Communication* (Kluwer Academic Publishers, 1988).
8. M. E. Austin, *Decision Feedback Equalization for Digital Communication Over Dispersive Channels* (M.I.T. Lincoln Laboratory, August 1967).
9. J. Salz, "Optimum Mean-Square Decision Feedback Equalizer," *B.S.T.J.*, vol. 52 (October 1973): 1341–1373.
10. C. A. Belfiore and J. H. Park, "Decision Feedback Equalization," *Proceedings of the IEEE*, vol. 67, no. 8 (August 1979).
11. D. L. Duttweiler, J. E. Mazo, and D. G. Messerschmitt, "Error Propagation in Decision Feedback Equalizer," *IEEE Trans. On Information Theory*, vol. IT-20 (July 1974): 490–497.
12. M. Tomlinson, "New Automatic Equalizer Employing Modulo Arithmetic," *Electron, Lett.*, vol. 7 (March 1971): 138–139.
13. H. Harashima and H. Miyakawa, "Matched-Transmission Technique for Channels with Intersymbol Interference," *IEEE Trans. on Comm. Tech.*, vol. COM-20 (August 1972): 774–780.

14. N. J. Lynch-Aird, "Tomlinson Precoding and DFE PAM: An Analytical Comparison for Recursive and Non-Recursive Equalizers," *BTL T1E1 Contribution,* T1E1.4/90-146 (September 1990).

15. A. K. Aman, R. L. Cupo, and N. A. Zervos, "Combined Trellis Coding and DFE Through Tomlinson Precoding," *IEEE JSAC,* vol. 9, no. 6 (August 1991): 876–884.

16. K. H. Mueller and J. J. Werner, "A Hardware Efficient Passband Equalizer Structure for Data Transmission," *IEEE Trans. Commun.,* vol. COM-30 (March 1982): 538–541.

17. G. H. Im and C. K. Un, "A Reduced Structure for the Passband Fractionally Spaced Equalizer," *Proc. IEEE,* vol. 75 (June 1987): 847–849.

18. F. Ling and S. U. H. Qureshi, "Convergence and Steady-State Behavior of a Phase-Splitting Fractionally Spaced Equalizer," *IEEE Trans. On Commun.,* vol. 38, no. 4 (April 1990): 418–425.

19. S. B. Weistein, "Data Transmission by Frequency-Division Multiplexing Using the Discrete Fourier Transform," *IEEE Trans. on Commun.,* vol. COM-19, no. 5 (October 1971): 628–634.

20. J. S. Chow and J. M. Cioffi, "A Cost-Effective Maximum Likelihood Receiver for Multicarrier System," *Proc. IEEE ICC '92* (June 1992): 948–952.

21. J. S. Chow, J. M. Cioffi, and J. A. C. Bingham, "Equalizer Training Algorithms for Multicarrier Modulation Systems," *Proc. IEEE ICC '93* (June 1993): 761–765.

22. D. D. Falconer and F. R. Magee, Jr., "Adaptive Channel Memory Truncation for Maximum Likelihood Sequence Estimation," *B.S.T.J.,* vol. 52 (November 1973): 1541–1562.

23. D. G. Messerschmitt, "Design of a Finite Impulse Response for the Viterbi Algorithm and Decision Feedback Equalizer," *Proc. IEEE ICC '74* (June 1974): 37D.1–37D.5.

24. N. Al-Dhahir and J. M. Cioffi, "Optimum Finite-Length Equalization for Multicarrier Transceivers," *IEEE Trans. On Commun.,* vol. 44, no. 1 (January 1996): 56–64.

CHAPTER 9

Echo Cancellation

For the voice application, a certain amount of echo was considered a positive feedback for telephone conversations until the satellite link was used for long distance calls. The long delay introduced by the satellite link makes echo an unpleasant experience. Voice echo cancellers are usually installed in COs to significantly reduce the magnitude of the echo. The echo cancellation level for voice applications is between 20 to 30 dB. For data communication, full-duplex transmission over a single pair of twisted-pair telephone loops can be realized using the echo cancellation technique.

The echo for DSL systems occurs at the hybrid circuit where the transmit path and the receiving path are joined together and connected to the twisted-pair telephone loop. An adaptive echo canceller is usually an integrated part of a DSL transceiver. To maintain a low residual echo level compared with the weak received signal, the required echo cancellation level can be as high as 60 dB.

The Basic Rate Access ISDN channel (also called DSL) and HDSL both use echo cancellation for the full-duplex transmission over a single pair of telephone loops using the same frequency band for opposite directions. The DMT ADSL has an option to use echo cancellation for an improved transmission throughput in the downstream direction (from a CO to a telephone subscriber). The echo cancellation for DMT ADSL is asymmetrical because signal bandwidths are different at opposite directions.

The general echo cancellation technique also can be used to reduce the crosstalk noise level if information and signal processing can be shared between DSL transceivers connected to different pairs of telephone loops.

This chapter discusses issues related to the requirements of echo cancellation, implementation of baseband and asymmetrical echo cancellers, and the feasibility of crosstalk cancellation as follows:

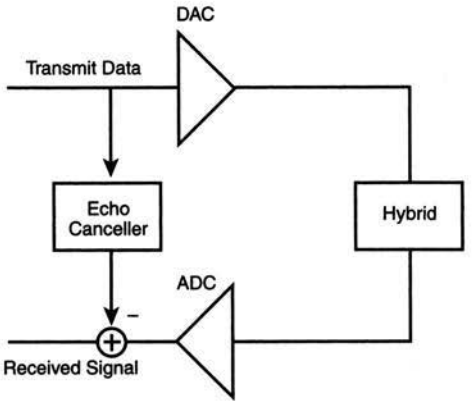
- *Echo cancellation requirements.* The echo path model is analyzed for the determination of required echo cancellation level and the length of an echo canceller. This chapter discusses the effect of line coupling transformer primary inductance.
- *Echo canceller requirements.* Based on the *Least Means Square* (LMS) algorithm, required adaptation step sizes are derived for half-duplex and full-duplex echo cancellation operations. This chapter analyzes the echo canceller convergence behavior and explains how to derive echo canceller implementation precision requirements for the full-duplex adaptation condition.
- *Asymmetrical cancellation.* This chapter introduces the concept of asymmetrical echo cancellation and discusses alternatives for efficient asymmetrical echo canceller implementation.
- *Crosstalk cancellation.* This chapter illustrates how to derive the crosstalk transfer function model as well as discusses the benefit and the general architecture of multiple-input and multiple-output crosstalk noise cancellation.

9.1 Echo Cancellation Requirements

A hybrid circuit can provide only limited echo attenuation due to the uncertain twisted-pair loop impedance for a wide range of frequency band. Echo needs to be filtered out for the case of a frequency division duplex system, or digitally reduced for the case of an echo cancellation based system. Bandpass filters that reject echo can be put in serial with the echo path because the desired signal does not share the same spectrum as the echo.

Digital adaptive echo cancellers, on the other hand, need to be in parallel with the echo path. Figure 9.1 shows the general application of a digital adaptive echo canceller in conjunction with a hybrid circuit. An echo canceller synthesizes the echo path including the DAC, the transmit filter, the hybrid circuit, the receiver filter, and the ADC. An echo canceller can produce an echo replica with the same transmitting data, but with reversed sign to cancel the real echo.

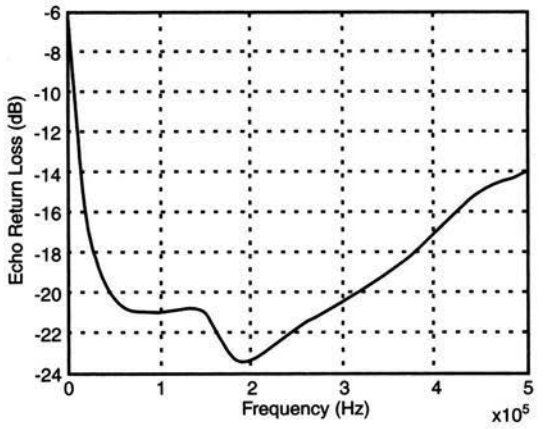
Figure 9.1 Adaptive Echo Cancellation



9.1.1 *Echo Path Transfer Function and Impulse Response*

Figure 9.2 shows the echo path transfer function for a twisted-pair loop with bridged taps. The varying echo attenuation values are caused by the variation of the loop input impedance that is related to bridged taps. The relatively low echo return loss at the low-frequency region is due to the higher impedance of the twisted-pair loop. The relatively low echo return loss at the high-frequency region is due to the leakage inductance of the transformer.

Figure 9.2 Echo Path Transfer Function



We define the echo return loss as the signal attenuation level of the echo path. The average echo return loss can be calculated as

$$K_{ep} = \frac{1}{f_c}\int_0^{f_c} |H_{ec}(f)|^2 \, df$$

$H_{ec}(f)$ is the transfer function of the echo path.

f_c is the corner frequency representing the bandwidth of interest.

The average echo return loss for this twisted-pair loop is 18.5 dB for $f_c = 500kHz$ with the worst echo return loss at only about 6 dB.

The echo return loss with a particular bandwidth constrained by transmit and receive filters could be slightly different compared with the echo return loss without these filters. We have

$$K_{ec} = \frac{1}{f_c}\int_0^{f_c} |G_t(f)H_{ec}(f)G_r(f)|^2 \, df$$

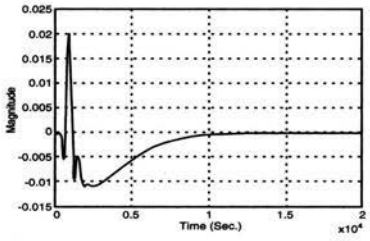
$G_t(f)$ is the frequency response of the transmit filter.

$G_r(f)$ is the frequency response of the receiver filter.

The average echo return loss for the same twisted-pair loop includes additional fourth order Butterworth transmit and receive filters with corner frequencies of 200 kHz and 17 dB.

Although echo path impulse responses are different, they generally consist of a rapidly changing initial portion and a slowly decaying tail portion, as shown in Figure 9.3. In general, the rapidly changing initial portion is due to reflections generated at terminal impedance mismatch and bridged tap locations, and the tail portion is mainly related to the primary inductance of the line transformer.

Figure 9.3 Echo Path Impulse Response

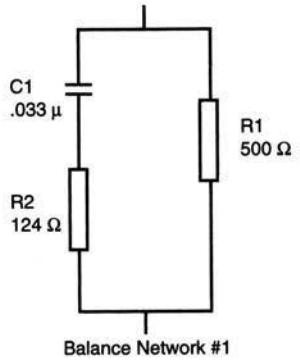


The reflections should die out in a relatively short period compared with the duration of the echo tail. Assuming they become negligible beyond the first round trip across the twisted-pair loop, reflections should not last more than the round trip delay of the longest twisted-pair loop. For an 18 kft, twisted-pair loop, the round trip delay is about 40 μs.

9.1.2 *The Effect of Transformer and Bandpass Filtering*

To see the effect of the primary inductance value on the echo tail, we can derive a simplified echo path model by using the RC network model (see Figure 9.4) in place of the loop impedance Z_i and the primary inductance L_{pri} alone in place of the line transformer equivalent circuit.

Figure 9.4 A Model Twisted-Pair Loop Impedance



For the resistor-only balance network, the simplified echo path model is

$$H(s)_{ec} = \frac{R_{12}}{Z_o + R_{12}} \frac{s(s+\omega_1)}{s^2 + 2\alpha + \omega_0^2} - \frac{1}{2}$$

for which

$$R_{12} = \frac{R_1 R_2}{R_1 + R_2}$$

$$\omega_1 = \frac{1}{R_2 C_1}$$

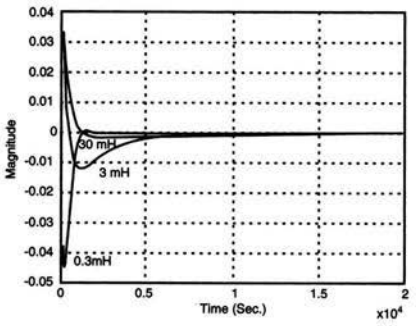
$$\omega_o^2 = \frac{1}{L_{pri}C_1}\frac{\dfrac{R_1 Z_0}{R_1 + Z_0}}{\dfrac{R_1 Z_0}{R_1 + Z_0} + R_2}$$

$$\alpha = \frac{1}{2}\left(\frac{1}{C_1\left(\dfrac{Z_o R_1}{Z_o + R_1} + R_2\right)} + \frac{Z_o R_1 R_2}{\left(R_1 R_2 + Z_o R_2 + Z_o R_1\right)L_{pri}}\right)$$

The simplified echo path is a typical overdamped RLC network with a resonant frequency of ω_o and a damping ratio of α.

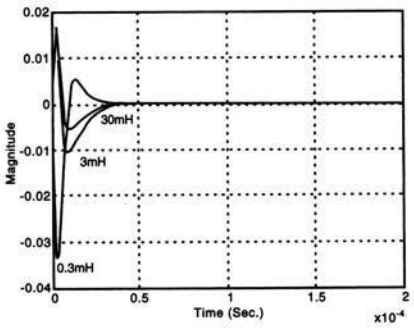
Figure 9.5 shows that the duration of the echo tail is directly related to the primary inductance of the line transformer. We can use a line transformer with a low primary inductance to reduce the length of the echo tail, as long as we satisfy the transformer efficiency, linearity, and system bandwidth requirements.

Figure 9.5 A Baseband Echo Path Impulse Response Model



A passband system has a significantly shorter echo tail due to the avoidance of heavy, low-frequency phase and magnitude distortions. The effect of a passband signal on the echo tail is demonstrated by passing the echo path impulse response of the simplified model through a highpass filter with a corner frequency of 20 kHz, as shown in Figure 9.6.

Figure 9.6 A Passband Echo Path Impulse Response Model



The echo tail disappears at around 100 μs. Similarly, a simple digital highpass filter of $H(z) = 1 - z^{-1}$ also can effectively shorten the echo tail.

Note

Attention also should be paid to the overall system low-frequency performance when using a highpass filter. In general, the removal of low-frequency energy will shorten the echo tail and ease implementation at the expense of some loss in overall system performance in a NEXT dominated environment.[1]

9.1.3 *Required Echo Cancellation Level and Length of Echo Canceller*

Assuming a white residual echo, the required echo cancellation level, K_{EC}, can be calculated according to

$$K_{EC} = SNR + K_{ch} - K_{ep}$$

The values are described as follows:

K_{ch} is the average channel loss.

K_{ep} is the average echo path loss.

SNR is the required average signal-to-residual noise floor ratio.

For the twisted-pair loop depicted in Figure 9.1, we have $K_{ep} = 17\,\text{dB}$ and $K_{ch} = 23\text{dB}$, and for an $SNR = 44\,\text{dB}$, the required echo cancellation level, K_{EC}, is about 60 dB. An SNR of 44 dB, here, corresponds to a noise floor of –40–23–44=–107 dBm/Hz, assuming a transmit signal power level of –40 dBm/Hz. With this required echo cancellation level, we can find the required echo cancellation time span through integration by setting the following:

$$K_{EC} = -10\log_{10}\left(\frac{\int_{t_s}^{\infty} h_{ec}^2(t)dt}{\int_{0}^{\infty} h_{ec}^2(t)dt}\right)$$

For an echo canceller size of 128 taps with an input/output rate of 400 kHz, we need to have a $t_s \leq 320$ μs.

For a self-NEXT dominant environment, where the NEXT noise level is much higher than the background noise or receiver front-end noise level, in conjunction with certain line code, the desired noise floor might not need to be that low. For using a baseband four-level line code, the desired *SNR* at the equalizer output is about 27.5 dB, including a performance margin of 6 dB for an error rate of 10^{-7}. For this case, a receiver front-end *SNR* of about 35 dB might be sufficient.

9.2 Echo Canceller Requirements

An adaptive echo canceller is a typical application of the digital adaptive FIR filter. Figure 9.7 shows the structure of a conventional FIR data driven echo canceller. Transmit data symbols, denoted by *x(k)*, are input to the adaptive filter. The output of this digital adaptive FIR filter is used as an echo replica to cancel the echo, denoted by *y(k)*, coming back from the hybrid circuit. The adaptation of filter coefficients is based on the error between the echo and its replica and the values of the data symbols.

9.2.1 Adaptation Step Size

Echo canceller filter coefficients (denoted by vector h_{ik} s) are estimated and updated iteratively according to the LMS algorithm[2]

$$H_{k+1} = H_k + \mu A_k\left(y(k) - H_k^T A_k\right)$$

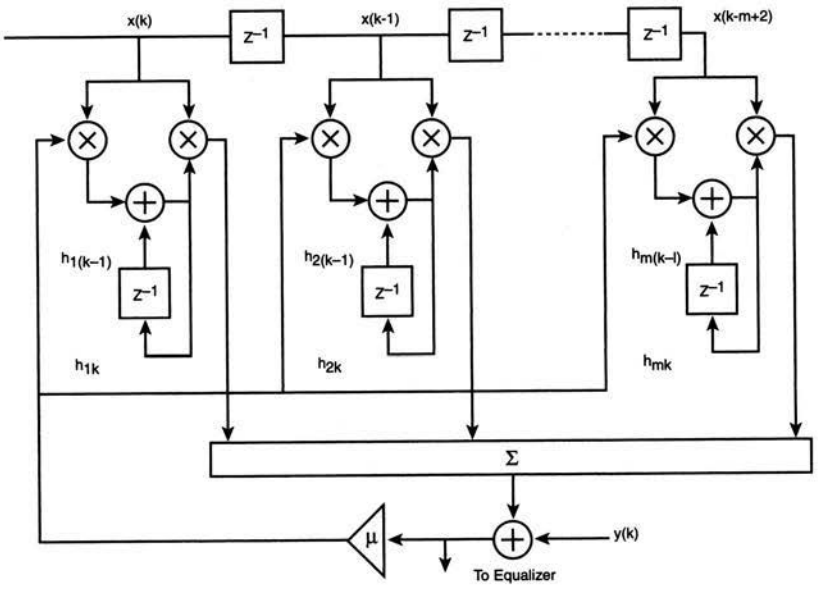
for which

$$H_k = \begin{bmatrix} h_1 \\ h_2 \\ \cdots \\ h_m \end{bmatrix}_2 \quad A_k = \begin{bmatrix} a_k \\ a_{k-1} \\ \cdots \\ a_{k-m+1} \end{bmatrix}$$

a_k is the current signal symbol, and $y(k)$ is the echo. To have a fast initial convergence, the echo canceller needs to be trained in half-duplex mode at startup. The optimal convergence step size under the half-duplex mode can be calculated as

$$\mu_{opt} \le \frac{1}{mE\left[a_k^2\right]} \mu_{opt} \le \frac{1}{(m+1)E\left[a_k^2\right]}$$

Figure 9.7 The Structure of an Echo Canceller



The value m is the number of filter coefficients. For $m = 128$ and $E\left[a_k^2\right] = 1$, we have $\mu_{opt} \le 0.007752$.

The convergence behavior of an echo canceller can be studied through its learning curve, which relates the mean square error to eigenvalues of the input data symbol autocorrelation matrix and the echo path impulse response. We have

$$\xi_k = \xi_{\min} + \sum_{n=1}^{m} h_n^2 \lambda_n \left(1 - 2\mu\lambda_n\right)^{2k}$$

for which eigenvalues are obtained through

$$\det\left[E\left[A_k A_k^T\right] - \lambda I\right] = 0$$

For a normalized white data symbol, we have $E\left[A_k A_k^T\right] = I$.

The preceding expression becomes $(1-\lambda)^m = 0$.

An echo canceller has m equal eigenvalues. For the normalized data symbol case, all eigenvalues are equal to 1. For this equal eigenvalue case, the learning curve can be simplified as follows:

$$\xi_{k=}\xi_{\min} + (1-2\mu)^{2k}\sum_{n=1}^{m} h_n^2$$

Note

Because all eigenvalues are the same, it is possible to use the previous expression to accurately predict the real-time convergence behavior of data-driven echo cancellers.

Assuming $\xi_{\min}$ is negligible, we can estimate the number of iterations required to reach a certain echo cancellation level based on the simplified learning curve. We have

$$\frac{\xi_0}{\xi_k} = (1-2\mu)^{-2k}$$

For $\xi_0 / \xi_k = 10^6$, we have

$$k = -\frac{3}{\log_{10}(1-2\mu)}$$

It will take an echo canceller at least 688 iterations to converge to a –60 dB residual echo level for a step size of $\mu = 0.005$. The real attainable convergence time is usually two to three times the estimation for using an LMS algorithm.

The convergence step size, μ, should be significantly reduced during full-duplex operation to maintain a low level of residual echo while tracking echo path transfer function variations. The attainable echo cancellation ratio is[3]

$$K_{EC} \approx \frac{(2-\mu_{full} m)K_{ch}}{\mu_{full} m}$$

K_{EC} and K_{ch} are expressed in power ratio. Therefore, we have

$$\mu_{full} \approx 2\frac{K_{ch}}{m(K_{EC}+K_{ch})} = \frac{2}{m\left(1+\frac{K_{EC}}{K_{ch}}\right)}$$

For $\frac{K_{EC}}{K_{ch}} = \frac{60}{23} = 37\text{dB}$ or 0.000199, we have $\mu_{full} = 0.0000031$ for m=128.

If the required K_{EC} is 50 dB, we have $\mu_{full} = 0.000031$ for the same m.

9.2.2 *Echo Canceller Precision Requirements*

There could be two precision requirements for echo canceller coefficients:

- Matching the FIR filter output quantization error to the available ADC precision
- Enabling the filter coefficient adaptation process when the step size is small during full-duplex operation

The first precision requirement applies to the half-duplex operation. For an ADC device of n bits, the quantization noise is uniformly distributed between $-\frac{1}{2^n}$ and $\frac{1}{2^n}$. To fully utilize the available bits of the ADC device, the echo replica from the echo canceller also will have n bits with the same or less quantization noise level. If the echo canceller has only one filter coefficient, the required coefficient precision is n bits.

For an echo canceller with more than one filter coefficient, the quantization noise will not be uniformly distributed, but approaches Gaussian as the number of coefficients increases. The filter output quantization noise is required to have the same or a smaller variance as that of an ADC device.

The variance for an ADC device of N bits is $\sigma_{A/D}^2 = \frac{1}{3}\frac{1}{2^{2N}}$.

N is the A/D resolution, and the variance of the echo replica from an echo canceller of length m with a resolution of M bits for filter coefficients is $\sigma_{filter}^2 = m\frac{1}{3}\frac{1}{2^{2M}}$.

By setting $\sigma_{A/D}^2 = \sigma_{filter}^2$, we have $M = N + \frac{1}{2}\log_2 m$.

To make the echo canceller quantization noise equal to, or smaller than, that of an ADC, we need the following:

$$M \geq N + \frac{1}{2}\log_2 m$$

For a 12-bit A/D device, the precision required for a 128 filter coefficient echo canceller is 16 bits. The filter coefficient precision requirements are 17 bits and 18 bits for 13-bit and 14-bit A/D converters, respectively.

To update them adaptively usually requires a higher precision for filter coefficients when a small step size is required for the full-duplex operation. We need to have[4]

$$M \geq -\log_2\left(\mu_{full} 10^{-\frac{K_{EC}}{20}}\right)$$

K_{EC} is expressed in units of dB. Therefore, to keep 60 dB of echo cancellation in full-duplex operation, we have

$$M \geq -\log_2\left(10^{-\frac{60}{20}}\right) - \log_2\left(\mu_{full}\right) = 9.66 + \log_2 \frac{1}{\mu_{full}}$$

For $\mu_{full} = 0.0000031$, we have $M \geq 9.66 + \log_2 \frac{1}{0.0000031} = 9.96 + 18.30 = 28.26$ bits.

For $\mu_{full} = 0.000031$, we have $M \geq 9.96 + \log_2 \frac{1}{0.000031} = 9.96 + 14.98 = 24.94$ bits.

9.3 *Asymmetrical Echo Cancellation*

Transmission throughputs in opposite directions of a twisted-pair loop do not need to be equal. Under this asymmetrical throughput condition, transmit spectra and sampling rates in opposite directions might be very different. At the high-rate transmit and low-rate receive end of a system based on twisted-pair loops, echo is caused by the high-rate data and is digitized at the low sampling rate of the low-rate receiver. At the low-rate transmit and high-rate receive end, echo is caused by the low-rate data and is digitized by the high sampling rate.

Figure 9.8 shows the general transceiver structure and associated echo cancellation task for the cancellation of higher bandwidth echo.[5] This configuration is associated with the high-rate transmit end where the transmit baud rate is higher than the receive sampling rate. The echo canceller can operate at the high baud rate while having its output decimated to the low sampling rate; however, this requires a high speed echo canceller with many adaptive filter coefficients.

Figure 9.8 Higher Bandwidth Echo Cancellation

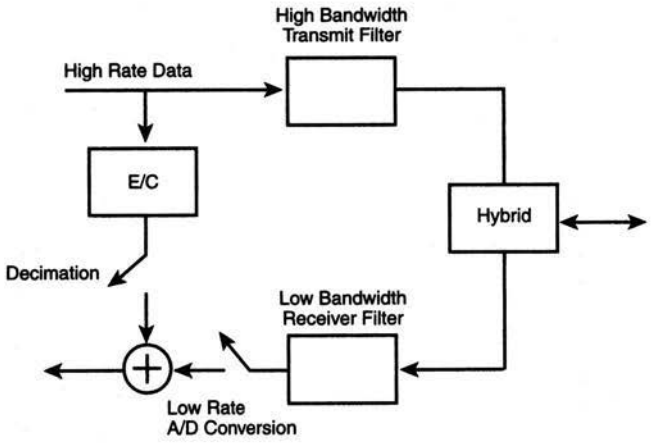
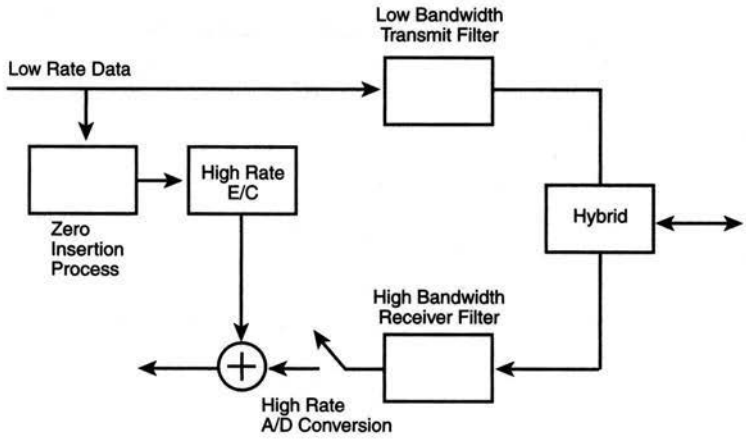


Figure 9.9 shows the general transceiver structure and associated echo cancellation task for the cancellation of lower bandwidth echo. This configuration is associated with the low bit rate transmit end where the transmit baud rate is much lower than the receive sampling rate. The echo canceller can operate at the high sampling rate while having its input interpolated from the low baud rate. The interpolation process can be accomplished by inserting zeros at sampling instances between low-rate sampling points, as well as other methods. However, this also requires a high-speed echo canceller with many adaptive filter coefficients.

Finding the minimum hardware configuration for this asymmetrical adaptive echo cancellation problem involves an understanding of the general topic of multirate signal processing.[6] Cancellation of higher bandwidth echo involves a ***decimation*** process. Cancellation of lower bandwidth echo involves an ***interpolation*** process.

For a conventional data-driven echo canceller for which the symbol rate is the same as the sampling rate, the echo path can be represented by a ***discrete time domain*** model with a delay interval the inverse of the symbol/sampling rate. Hence, a digital echo canceller with the same discrete time delay interval can be constructed, emulating the echo path, to generate an echo replica.

Figure 9.9 Lower Bandwidth Echo Cancellation



When the symbol rate and the sampling rate are different, the delay time interval of the discrete echo path model is the inverse of the higher rate among the symbol and sampling rates. This leads to a high processing speed echo canceller.

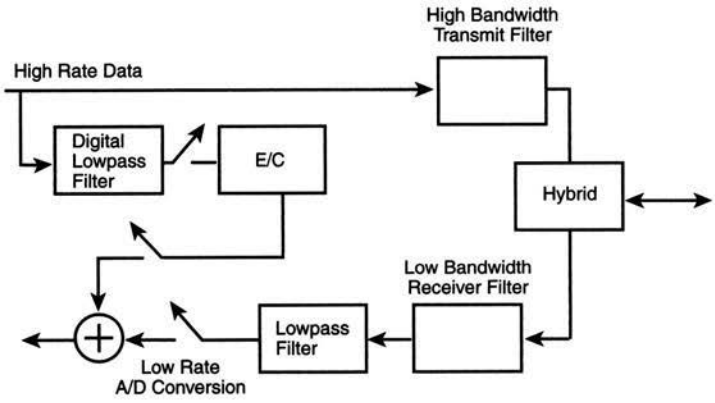
9.3.1 *High Bandwidth Echo Cancellation*

For the case of higher bandwidth echo cancellation, the effective symbol rate through the echo path can be reduced by using a high-order, lowpass filter before or after the receiver lowpass filter. A corresponding digital lowpass filter followed by a decimation process should be used to reduce the effective symbol rate to the digital echo canceller. Thus, a lower processing rate echo canceller can be utilized, as shown in Figure 9.10. This also will result in a much shorter echo canceller time span; however, the data input to the echo canceller is no longer a four-level signal, but of high precision.

Furthermore, the time domain spread of the transmit signal by the lowpass filter will cause an eigenvalue spread of the autocorrelation matrix associated with the adaptive echo canceller. This in turn will slow down the echo canceller initial convergence time.

Because there is always some leakage in the stopband of a practical lowpass filter, the processing rate of the adaptive echo canceller should be at least twice the sampling rate to have a better attenuation of the stopband signal while maintaining a smooth response on the passband signal. Leakage energy from both lowpass filters will appear as residual echo. Furthermore, leakage energy as noise will affect the adaptation process. When using the LMS algorithm, the adjustment step size should be accordingly reduced to minimize the excessive residual noise level.

Figure 9.10 Low-Rate, Higher-Bandwidth Echo Cancellation



9.3.2 *Low Bandwidth Echo Cancellation*

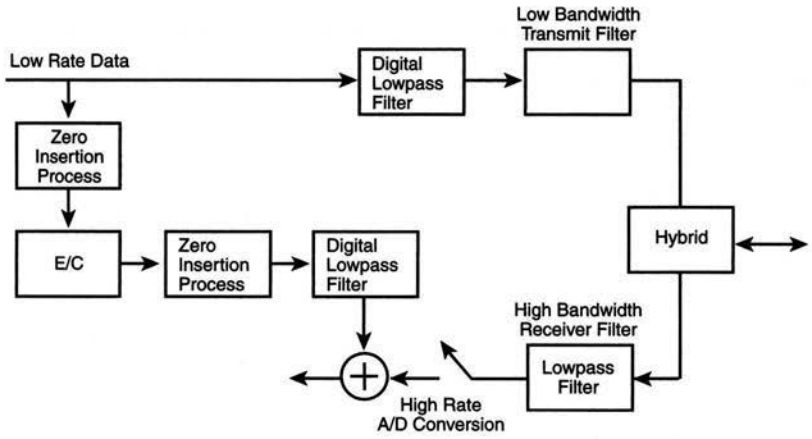
For the case of lower bandwidth echo cancellation, the minimum sampling rate for the purpose of echo cancellation also can be reduced by using a high-order, lowpass filter before or after the transmit lowpass filter. Thus, a lower processing rate echo canceller would be sufficient to generate echo replica for the purpose of echo cancellation.

The corresponding high sampling rate echo replica is generated with an interpolation process with the same lowpass filter as used in the transmit path, as shown in Figure 9.11. This configuration will result also in a much shorter echo canceller time span. To provide some excessive bandwidth for the low throughput channel and to avoid an additional interpolation process, the stopband edge of the high-order, lowpass filter should be twice that of the –3dB bandwidth of the low throughput channel.

Under this configuration, the operation rate of the echo canceller can be reduced to twice that of the low sampling rate. We simply insert zeros between each of the low-rate transmit data samples to provide data input for the echo canceller. The error signal also will be sampled twice for every low-rate baud interval.

The interpolation lowpass filter will take the output of the echo canceller as its input with zeros filling these missing high sampling data points. Proper time delay should be considered to compensate delays caused by these high-order, lowpass filters. Because the data input to the echo canceller is still white, these echo canceller coefficients can be effectively identified by the LMS algorithm.

Figure 9.11 Lower-Rate, Lower-Bandwidth Echo Cancellation



Asymmetrical echo cancellation can be applied to an asymmetrical throughput twisted-pair loop system to yield efficient all digital implementations. Asymmetrical echo cancellation with low processing rate echo cancellers is possible with proper higher-order, lowpass filtering. Additional decimation or interpolation processes are involved. While the hardware savings are very significant for cancellation of lower bandwidth echo with a low-rate echo canceller, additional cost on analog and digital lowpass filtering might be significant.

Modest hardware savings also can be realized for the cancellation of higher bandwidth echo with a low-rate echo canceller. However, the resulting initial convergence time might be slow, and the residual echo level might be high if the LMS algorithm is used.

9.4 *Crosstalk Cancellation*

Echo cancellation can be used to realize the full-duplex transmission in the same frequency bandwidth for opposite directions. The same adaptive signal processing technique can be used also to cancel the crosstalk to reduce the noise level, consequently improving the transmission throughput. The differences for crosstalk cancellation are the need for signal processing across different transceivers and the required signal processing power to handle such a multiple-input and multiple-output problem.

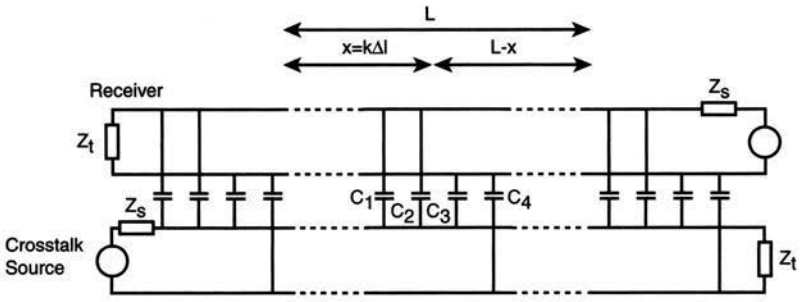
To cancel an interference, such as an echo, we need access to the original disturber signal, such as data symbols, to generate echo replica in conjunction with a simulated echo path. If we have access to data symbols generating crosstalk and we can identify and simulate the crosstalk transfer function, we can avoid the effect of crosstalk by performing crosstalk noise

cancellation. The access of crosstalk-generating data symbols is possible given a cable with a small number of twisted pairs all sharing the same transceiver electronics.

9.4.1 *Derivation of the Crosstalk Impulse Response*

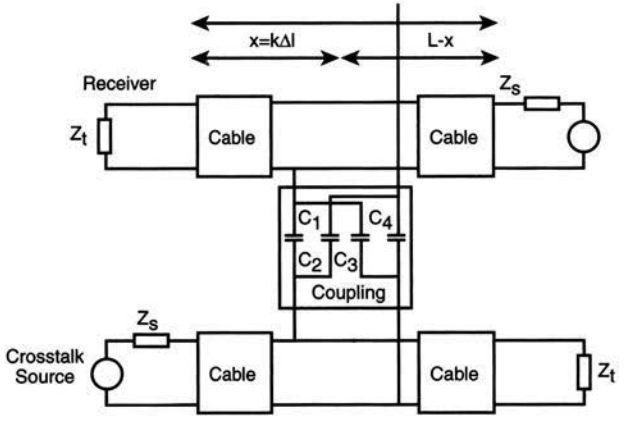
The crosstalk transfer function can be estimated by examining the fundamental mechanism of crosstalk. It has been shown that crosstalk is caused by the unbalance of coupling capacitances between twisted pairs, as depicted in Figure 9.12.[7] We can have four coupling capacitors between two pairs of twisted-pair loops at a particular location.

Figure 9.12 Crosstalk Caused by Coupling Capacitors



There are many such distributed coupling mechanisms along pairs of twisted-pair loops. The overall coupling effects of crosstalk can be obtained by the summation of each individual coupling mechanism at all locations. The effect of an individual coupling mechanism can be examined by using the equivalent two-port network circuit, as shown in Figure 9.13. Transmission characteristics of each two-port network are described by its ABCD parameters.

Figure 9.13 A Two-Port Network Equivalent Circuit for Crosstalk



For the analysis of NEXT, the twisted-pair loop sections beyond the coupling point can be simplified to parallel impedances similar to bridged taps. The simplified two-port network circuit for the analysis of NEXT has two two-port networks representing two cable sections, including parallel twisted-pair loop impedances and one two-port network representing the capacitor coupling mechanism. The ABCD parameters for the crosstalk originating cable section are as follows:

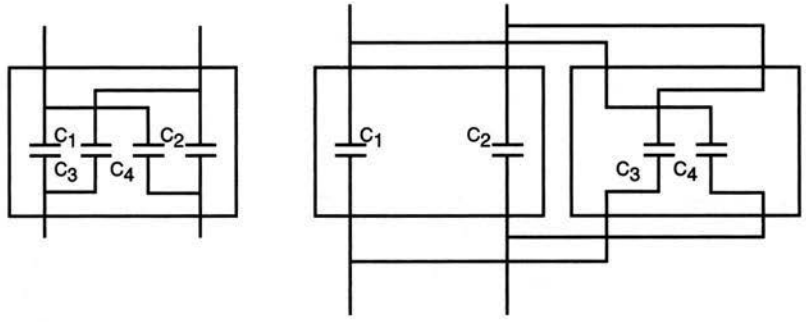
$$\begin{bmatrix} A_O & B_O \\ C_O & D_O \end{bmatrix} = \begin{bmatrix} A_N & B_N \\ C_N & D_N \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \dfrac{C_F Z_t + D_F}{A_F Z_t + B_F} & 1 \end{bmatrix} = \begin{bmatrix} A_N + B_N \dfrac{C_F Z_t + D_F}{A_F Z_t + B_F} & B_N \\ C_N + D_N \dfrac{C_F Z_t + D_F}{A_F Z_t + B_F} & D_N \end{bmatrix}$$

Subscript N denotes the near-end section of the cable and subscript F denotes the far end section of the cable in reference to the location of the crosstalk source and the receiver. The ABCD parameters for the crosstalk receiving cable section are defined as

$$\begin{bmatrix} A_R & B_R \\ C_R & D_R \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \dfrac{C_F Z_t + D_F}{A_F Z_t + B_F} & 1 \end{bmatrix} \begin{bmatrix} A_N & B_N \\ C_N & D_N \end{bmatrix} = \begin{bmatrix} A_N & B_N \\ A_N \dfrac{C_F Z_t + D_F}{A_F Z_t + B_F} + C_N & B_N \dfrac{C_F Z_t + D_F}{A_F Z_t + B_F} + D_N \end{bmatrix}$$

The ABCD parameters for the capacitor coupling mechanism can be developed by splitting these four coupling capacitors into two two-port networks in parallel, as depicted in Figure 9.14.

Figure 9.14 Equivalence of the Coupling Two-Port Network



Then for network A at left (refer to Figure 9.14), we have

$$\begin{bmatrix} A_A & B_A \\ C_A & D_A \end{bmatrix} = \begin{bmatrix} 1 & \dfrac{1}{j\omega(C_1 + C_2)} \\ 0 & 1 \end{bmatrix}$$

And for network B at right (refer to Figure 9.14), we have

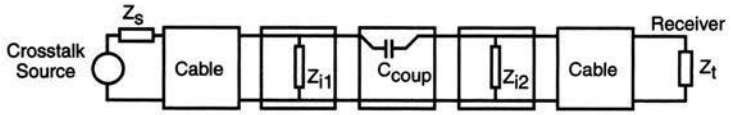
$$\begin{bmatrix} A_B & B_B \\ C_B & D_B \end{bmatrix} = -\begin{bmatrix} 1 & \dfrac{1}{j\omega(C_3 + C_4)} \\ 0 & 1 \end{bmatrix}$$

The ABCD parameters for the capacitor coupling mechanism are as follows:

$$\begin{bmatrix} A_C & B_C \\ C_C & D_C \end{bmatrix} = \begin{bmatrix} 1 & \dfrac{1}{j\omega(C_1 + C_2 - C_3 - C_4)} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & \dfrac{1}{j\omega C_{coup}} \\ 0 & 1 \end{bmatrix}$$

Therefore, the coupling mechanism can be further simplified to one capacitor with the possibility of having negative capacitance, as illustrated in Figure 9.15.

Figure 9.15 A Simplified Two-Port Network Circuit for Crosstalk



The ABCD parameters for the NEXT transfer function due to a particular value of C_{coup} at a particular location of $x = k\Delta l$ are

$$\begin{bmatrix} A_k & B_k \\ C_k & D_k \end{bmatrix} = \begin{bmatrix} A_O & B_O \\ C_O & D_O \end{bmatrix}\Bigg|_{x=k\Delta l} \begin{bmatrix} A_C & B_C \\ C_C & D_C \end{bmatrix} \begin{bmatrix} A_R & B_R \\ C_R & D_R \end{bmatrix}\Bigg|_{x=k\Delta l}$$

in which the twisted-pair loop has been divided into n sections of equal distance Δl.
For a fixed length twisted-pair loop with no bridged taps, the crosstalk at a particular frequency is

$$H_{cross}(\omega_i) = \sum_{k=1}^{n} H_k(\omega_i) H_{cross}(\varpi_i) = \sum_{k=1}^{n} H_k(\varpi_i)$$

We have

$$H_k(\omega_i) = \frac{Z_t}{Z_s\left(C_k(\omega_i)Z_t + D_k(\omega_i)\right) + A_k(\omega_i)Z_t + B_k(\omega_i)}$$

$$H_k(\varpi_i) = \frac{Z_t}{Z_s\left(C_k(\varpi_i)Z_t + D_k(\varpi_i)\right) + A_k(\varpi_i)Z_t + B_k(\varpi_i)}$$

The transfer function of NEXT can be constructed by evaluating $H_{cross}(\omega_i)H_{cross}(\varpi_i)$ for an interested frequency band of $\omega_i\varpi_i$s. Figure 9.16 shows the estimated NEXT transfer function in conjunction with a 26-gauge, 9 kft twisted-pair loop with source and terminal impedances of 100 ohms using uniformly distributed random-coupling capacitances between –3 to 3 pF and $\Delta l = 9$ ft.

Figure 9.16 Estimated NEXT Transfer Function

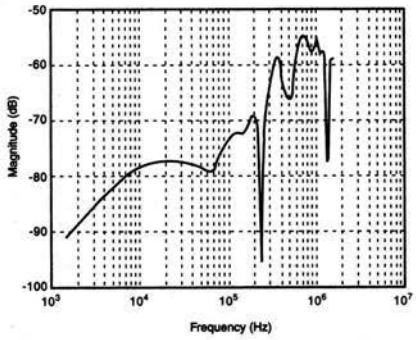
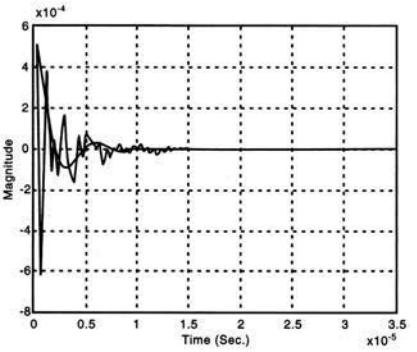


Figure 9.17 shows corresponding impulse response with and without two 200 kHz fourth order Butterworth transmit and receive filters. The impulse response is smoother with filters.

For the NEXT dominant case, in which the NEXT noise level is much higher than background noise, any reduction of NEXT noise level will improve the transmission performance in terms of a higher transmission throughput or a lower error rate. The background noise level can be 30 dB lower than the NEXT noise level. Under this assumption, a NEXT canceller with 30 dB of NEXT noise cancellation can suppress the effect of NEXT.

Figure 9.17 Estimated NEXT Impulse Response

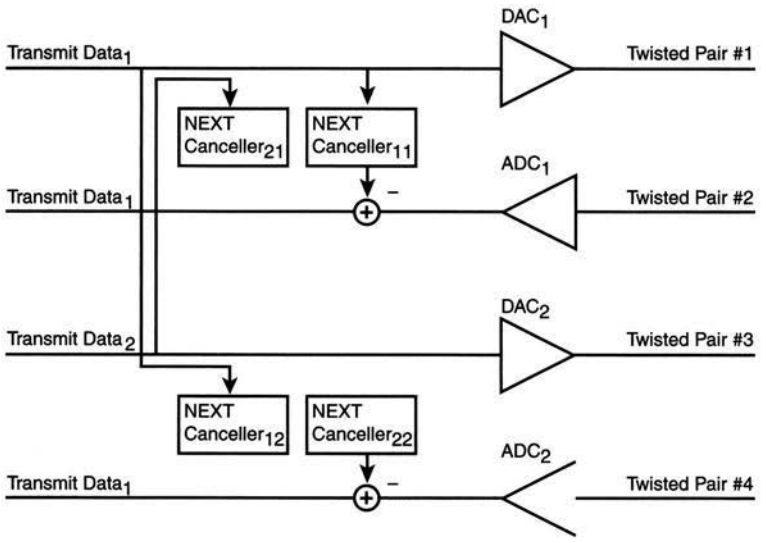


The time span of a NEXT impulse response lasts only about 20 μs. Because there is not much low-frequency content, the NEXT impulse response is relatively short. With a sampling rate of 400 kHz, the NEXT impulse response can be emulated with only 8 taps.

9.4.2 *NEXT Cancellation Architecture*

Figure 9.18 shows the general architecture of the application of NEXT cancellers for two pairs of two-pair transceivers; $n2$NEXT cancellers are necessary for n transceiver pairs.

Figure 9.18 The Configuration of the NEXT Cancellation



The NEXT cancellation requires multiple-input and multiple-output adaptive signal processing. Although the required signal processing is very extensive, it might become feasible depending on the development of semiconductor technology.

End Notes

1. W. Y. Chen, "Echo Cancellation for Passband and Passband Systems," *Bellcore T1E1 Contribution,* T1E1.4/90-181 (September 1990).

2. B. Widrow, *Adaptive Signal Processing* (Prentice-Hall, 1995).

3. K. H. Mueller, "A New, Fast-Converging Mean-Square Algorithm for Adaptive Equalizers with Partial Response Signaling," *B.S.T.J.,* vol. 54 (January 1975): 143–153.

4. R. D. Gitlin and S. B. Weinstein, "On the Required Tap-Weight Precision for Digitally Implemented, Adaptive, Mean-Squared Equalizers," *B.S.T.J.,* vol. 58, no. 2 (February 1979): 301–321.

5. W. Y. Chen, "Asymmetrical Echo Cancellation for a Baseband ADSL," *Bellcore T1E1 Contribution,* T1E1.4/92-156 (August 1992).

6. R. E. Crochiere and L. R. Rabiner, *Multirate Digital Signal Processing* (Prentice Hall, 1993).

7. "Dr. Campbell's Memoranda of 1907 and 1912," *Bell Systems Technical Journal,* vol. 14, no. 4 (October 1935): 552–572.

CHAPTER 10

Error Correction and Trellis Coding

The selection of a proper line code according to the transmission condition in conjunction with the use of an adaptive channel equalizer can maximize the transmission potential for a particular channel environment. Transmission error will occur, however, due to the random nature of the channel noise. Error correction coding can be used to reduce the transmission error rate by introducing certain transmission redundancy. The transmission redundancy can be allocated at either the bit level (such as the Reed-Solomon code) or embedded at the bit-to-symbol conversion level (such as the Trellis coded modulation).

There are many methods for the allocation of bit-level redundancy. A simple and effective approach is parity checking. A simple parity check can be used to identify whether an error has occurred. With multiple applications of parity checking, a certain amount of errors can be automatically corrected. Depending on the size of the code, there could be many different ways of applying parity checking. With a few rules of guidance, efficient codes have been identified by computer searching.

Besides the coding efficiency, the simplicity of encoder and decoder implementation is also a very important concern. Reed-Solomon code has been used for ADSL as well as many other transmission systems for its coding efficiency and relative implementation simplicity.

For a bandwidth-limited transmission environment, the direct application of coding overhead at bit level might result in a negative coding gain because of the necessary signal level or constellation size increase to accommodate the additional transmission throughput. Under the bandwidth limited condition, the Trellis coded modulation has been very effective at providing a reasonably high coding gain depending on the code complexity.

For a Trellis code, the net signal-to-noise ratio decrease due to the increase of signal level or constellation size has been avoided by systematically interrelating possible occurrences of

adjacent symbols. In other words, a random sequence of data is purposely encoded into allowed sequences of symbols. The correlation between adjacent symbols effectively improves the signal-to-noise ratio, therefore providing a coding gain.

This chapter reviews fundamental knowledge and design parameters of coding techniques related to DSL systems, including:

- *Parity check codes.* General concepts of parity check and error syndrome for transmission error identification and correction are discussed. Corresponding matrix representation and manipulation are introduced with calculation examples. Galois field-based polynomial representation also is introduced. Algebra manipulation of the polynomial representation is discussed with numerical examples.
- *Reed-Solomon codes.* The Reed-Solomon code is introduced as a non-binary BCH code. The generation polynomial of the Reed-Solomon code is discussed with numerical examples. Possible Reed-Solomon encoder and decoder implementations are discussed. The use of the Reed-Solomon code for the application of ADSL is discussed with reference to some simulation results.
- *Convolution codes.* The general structure of a convolution encode is introduced, and its generation matrix is defined. Concepts of transition, tree, and Trellis diagrams are discussed. The Viterbi decoder in conjunction with the convolution code is described with numerical examples for both hard and soft decision devices.
- *Trellis code modulation.* The rationale for using Trellis coded modulation for bandwidth-limited transmission systems is explored. Trellis codes and corresponding signal level and constellation mappings are discussed for baseband and passband line codes, respectively.

  Both feedforward and feedback implementation of Trellis codes are described with examples. The analytical representation is also discussed with a numerical example. Examples of linear and non-linear phase invariant Trellis codes for passband line codes are listed.

10.1 Parity Check Codes

When transmitting a binary sequence, the received sequence might be a little different than the original sequence due to the channel uncertainty caused by additive noise and so forth. We cannot tell from the received sequence itself if it is correct unless some redundancy is involved. We can repeat the transmission of the same sequence and compare sequences from each repetition to decide if the received sequence is correct and/or what the correct value should be based on a majority decision if the repetition occurs more than twice.

To check for just the possible occurrence of errors, we also can count the number of 1s or 0s in the sequence, and use a parity check bit to indicate whether it is an even or odd number.[1,2] We then send this check bit along with the original sequence. At the receiver side, a new parity bit is generated according to the received sequence and compared with the received one to decide whether an error or errors have occurred.

To increase the reliability of error-checking, we can divide the sequence into a number of exclusive groups and put a parity check bit for each group. The number of bits in each group can be as small as two. Furthermore, we can define redundant groups—a bit can belong to a number of different groups. Through this redundancy, not only can errors be detected through the parity check bits, but they also can be corrected by cross examining different groups to which these bits belong.

For a binary sequence, the parity checking can be performed through the exclusive-or operation. By denoting $\oplus$ for the exclusive-or operation, the results between two binary digits are as follows:

$$0 \oplus 0 = 0$$
$$0 \oplus 1 = 1$$
$$1 \oplus 0 = 1$$
$$1 \oplus 1 = 0$$

We can, for example, send a sequence of two information bits together with a parity check bit. The receiver will generate its own parity bit and compare it with the received one to detect an error in the received information bits. This is called a $\left(N, k, d_{\min}^{H}\right)$ parity check code. All possible combinations of information bits and associated parity check bits are called as code words. For this example, we have

- The code word dimension, the length of code words, of N=3;
- The information bit sequence of dimension, the length of information sequence, k=2;
- The shortest Hamming distance, the minimum distance or the number of different bits between code words, of $d_{\min}^{H} = 2$;
- The code rate, the ratio of information bit sequence dimension over code word dimension, is $R_c = \frac{k}{N} = \frac{2}{3}$. This code can be used for error detection, but not error correction.

Code words for this example are (000), (011), (101), and (110). In general,

$$t = \left[\frac{d_{\min}^{H} - 1}{2} \right] \text{ errors}$$

can be corrected for a code of $d_{\min}^{H}$.

10.1.1 Achieving Error Correction

To achieve error correction, we can combine two of the code words for this example with more parity check bits. For every two (3,2,2) code words, we generate three more parity bits based on three two-bit groups across these two code words. For two original code words,

$$(i_1, i_2, p_1) \text{and } (i_3, i_4, p_2)$$

in which

$$p_1 = i_1 \oplus i_2 \text{ and } p_2 = i_3 \oplus i_4$$

we have the new code word

$$(i_1, i_2, p_1, i_3, i_4, p_2, p_3, p_4, p_5)$$

in which

$$p_3 = i_1 \oplus i_3$$
$$p_4 = i_2 \oplus i_4$$
$$p_5 = p_3 \oplus p_4$$

This combination is a (9, 4, 4) parity check code and has a code rate of 4/9. This code can be used to detect one error: (4–1)/2. The position of an error can be identified by cross two-parity check bits that indicate errors.

In general, the position of errors can be identified using a syndrome vector. The elements of the syndrome vector are the parity check of these parity bits and their corresponding information bits. For example, given a (6,3,3) code, we have

$$p_1 = i_1 \oplus i_2$$
$$p_2 = i_2 \oplus i_3$$
$$p_3 = i_1 \oplus i_3$$

For a syndrome vector $s = (s_1, s_2, s_3)$, we have

$$s_1 = i_1 \oplus i_2 \oplus p_1$$
$$s_2 = i_2 \oplus i_3 \oplus p_2$$
$$s_3 = i_1 \oplus i_3 \oplus p_3$$

A look up table can be constructed to relate the value of the syndrome vector to a particular error pattern. The look up table can be implemented in either a memory or logic-based circuit.

A code vector also can be described in the following matrix format. A code vector, *c*, is a product of the message vector, *m*, and the generator matrix, *G*. We have $c = mG$. For a systematic code, where the first *k* components of any code vector are the components of message vector, *m*, the generator matrix can be further divided as follows:

$$G = [I_k \quad P]$$

I_k is the $k \times k$ dimensional identity matrix and *P* is the $k \times r$ dimensional parity check array in which $r = N - k$. For the (3,2,2) parity check code example, we have the following:

$$G = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

These code words are expressed as the following:

$$[0 \quad 0]\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} = [0 \quad 0 \quad 0]$$

$$[0 \quad 1]\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} = [0 \quad 1 \quad 1]$$

$$[1\ \ 0]\begin{bmatrix}1 & 0 & 1\\ 0 & 1 & 1\end{bmatrix}=[1\ \ 0\ \ 1]$$

$$[1\ \ 1]\begin{bmatrix}1 & 0 & 1\\ 0 & 1 & 1\end{bmatrix}=[1\ \ 1\ \ 0]$$

The addition of matrix elements also is performed with exclusive-or.

For the (6,3,3) parity check code example, we have the following:

$$G=\begin{bmatrix}1 & 0 & 0 & 1 & 0 & 1\\ 0 & 1 & 0 & 1 & 1 & 0\\ 0 & 0 & 1 & 0 & 1 & 1\end{bmatrix}$$

These code words are expressed as

$$[0\ \ 0\ \ 0]\begin{bmatrix}1 & 0 & 0 & 1 & 0 & 1\\ 0 & 1 & 0 & 1 & 1 & 0\\ 0 & 0 & 1 & 0 & 1 & 1\end{bmatrix}=[0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0]$$

$$[0\ \ 0\ \ 1]\begin{bmatrix}1 & 0 & 0 & 1 & 0 & 1\\ 0 & 1 & 0 & 1 & 1 & 0\\ 0 & 0 & 1 & 0 & 1 & 1\end{bmatrix}=[0\ \ 0\ \ 1\ \ 0\ \ 1\ \ 1]$$

$$[0\ \ 1\ \ 0]\begin{bmatrix}1 & 0 & 0 & 1 & 0 & 1\\ 0 & 1 & 0 & 1 & 1 & 0\\ 0 & 0 & 1 & 0 & 1 & 1\end{bmatrix}=[0\ \ 1\ \ 0\ \ 1\ \ 1\ \ 0]$$

$$[0\ \ 1\ \ 1]\begin{bmatrix}1 & 0 & 0 & 1 & 0 & 1\\ 0 & 1 & 0 & 1 & 1 & 0\\ 0 & 0 & 1 & 0 & 1 & 1\end{bmatrix}=[0\ \ 1\ \ 1\ \ 1\ \ 0\ \ 1]$$

$$[1\ 0\ 0]\begin{bmatrix}1&0&0&1&0&1\\0&1&0&1&1&0\\0&0&1&0&1&1\end{bmatrix}=[1\ 0\ 0\ 1\ 0\ 1]$$

$$[1\ 0\ 1]\begin{bmatrix}1&0&0&1&0&1\\0&1&0&1&1&0\\0&0&1&0&1&1\end{bmatrix}=[1\ 0\ 1\ 1\ 1\ 0]$$

$$[1\ 1\ 0]\begin{bmatrix}1&0&0&1&0&1\\0&1&0&1&1&0\\0&0&1&0&1&1\end{bmatrix}=[1\ 1\ 0\ 0\ 1\ 1]$$

$$[1\ 1\ 1]\begin{bmatrix}1&0&0&1&0&1\\0&1&0&1&1&0\\0&0&1&0&1&1\end{bmatrix}=[1\ 1\ 1\ 0\ 0\ 0]$$

A syndrome vector also can be described in the following matrix format. A syndrome vector, s, is a product of the received code vector, y, and the parity check matrix, H. We have $s = yH^T$. Also for a systematic code, the parity check matrix can be further divided as follows:

$$H = \begin{bmatrix} P^T & I_r \end{bmatrix}$$

I_r is the $r \times r$ dimensional identity matrix, and P is the same $k \times r$ dimensional parity check array. For the (3,2,2) parity check code example, we have H=[1 1 1]. For the (6,3,3) parity check example, we have the following:

$$H = \begin{bmatrix}1&1&0&1&0&0\\0&1&1&0&1&0\\1&0&1&0&0&1\end{bmatrix}$$

$$G = \begin{bmatrix}1&0&0&1&0&1\\0&1&0&1&1&0\\0&0&1&0&1&1\end{bmatrix}$$

We always have the orthogonal property of

$$GH^T = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = 0$$

If the received code vector has no error (that is, $y=c$), we have

$$s = cH^T = mGH^T = 0$$

If it contains errors, we represent the received code vector as the sum of code vector and the error vector. We have

$$s = yH^T = (c+e)H^T = cH^T + eH^T = eH^T$$

The dimension of s is r, while the dimension of e is $k+r$ There are 2^k error vectors, es, satisfying above r linear equations. For a Binary Symmetrical Channel, we choose the one that has the minimum non-zero digits as the error vector, e, so that the vector corresponds to the least number of errors, and the probability of receiving a sequence with the least number of errors is higher than the probability of receiving a sequence with more errors. For the (6,3,3) parity check code example, we let

$$y = [1 \quad 1 \quad 1 \quad 0 \quad 0 \quad 1]$$

We have the syndrome

$$s = yH^T = [1 \quad 1 \quad 1 \quad 0 \quad 0 \quad 1] \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = [0 \quad 0 \quad 1]$$

We have following $2^3 = 8$ error vectors, *es*, satisfying $s = eH^T = [0 \quad 0 \quad 1]$:

$[0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1]$
$[0 \quad 0 \quad 1 \quad 0 \quad 1 \quad 0]$
$[0 \quad 1 \quad 0 \quad 1 \quad 1 \quad 1]$
$[0 \quad 1 \quad 1 \quad 1 \quad 0 \quad 0]$
$[1 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0]$
$[1 \quad 0 \quad 1 \quad 1 \quad 1 \quad 1]$
$[1 \quad 1 \quad 0 \quad 0 \quad 1 \quad 0]$
$[1 \quad 1 \quad 1 \quad 0 \quad 0 \quad 1]$

We choose $[0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1]$ as the error vector because it has only one non-zero digit. The corresponding code vector is $[1 \quad 1 \quad 1 \quad 0 \quad 0 \quad 0]$.

10.1.2 *Cyclic Codes*

An important subclass of a linear code is called a cyclic code, in which every cyclic shift of a code vector is also a code vector. The encoding and syndrome computation of cyclic codes can be implemented with shift registers, and the decoding can be handled effectively with its algebraic structure. We can treat components of a message or a code vector

$$m = [i_1 \quad i_2 \quad \ldots \quad i_k]$$

as coefficients of a polynomial, as follows:

$$m(x) = i_1 + i_2 x + \ldots + i_k x^{k-1}$$

Code polynomials, corresponding to code vectors, can be generated as products of message polynomials and a generator polynomial. Table 10.1 shows a (6,3,3) cyclic code that is generated with three-bit message polynomials and a generator polynomial, $1 + x^3$.

Table 10.1 A (6,3,3) Cyclic Code

Messages	Code Vectors	Code Polynomial
[0 0 0]	[0 0 0 0 0 0]	$0(1+x^3)=0$
[0 0 1]	[0 0 1 0 0 1]	$x^2(1+x^3)=x^2+x^5$
[0 1 0]	[0 1 0 0 1 0]	$x(1+x^3)=x+x^4$
[0 1 1]	[0 1 1 0 1 1]	$(x+x^2)(1+x^3)=x+x^2+x^4+x^5$
[1 0 0]	[1 0 0 1 0 0]	$1(1+x^3)=1+x^3$
[1 0 1]	[1 0 1 1 0 1]	$(1+x^2)(1+x^3)=1+x^2+x^3+x^5$
[1 1 0]	[1 1 0 1 1 0]	$(1+x)(1+x^3)=1+x+x^3+x^4$
[1 1 1]	[1 1 1 1 1 1]	$(1+x+x^2)(1+x^3)=1+x+x^2+x^3+x^4+x^5$

10.1.3 *Galois Field Arithmetic*

In dealing with these vector and polynomial manipulations, we have used some algebraic properties of a binary *Galois field, GF(2)*. In general, for a sequence of length m with q values for each bit, there could be q^m different vectors. For q a prime number, we can have a field ranging from 0 to q–1 different elements (values). We sometimes call this a Galois field of q elements—GF(q).

Elements of the field can be manipulated through addition and multiplication operations; however, because we are dealing with a finite field, these operations are defined in a modular sense. The binary exclusive-or operation is actually a modular addition in GF(2). For an example of GF(3), we have the following for addition and multiplication:

Addition	Multiplication
$0+0=0$	$0\times0=0$
$0+1=1$	$0\times1=1$
$0+2=2$	$0\times2=2$

$1+0=1 \qquad 1\times 0=0$

$1+1=2 \qquad 1\times 1=1$

$1+2=0 \qquad 1\times 2=2$

$2+0=2 \qquad 2\times 0=0$

$2+1=0 \qquad 2+1=2$

$2+2=1 \qquad 2\times 2=1$

10.1.4 *Extended Galois Field Arithmetic*

It is sometimes more convenient to consider binary information sequences and corresponding parity check bits in 2^m base as those we often used for binary representation of octal or hexadecimal numbers in computer and electronics circuits. We call GF(2^m) an extension field of the binary Galois field, GF(2). Similar to the binary representation of octal numbers, any element in GF(2^3) can be represented by a sequence of three binary digits, a_0, a_1, and a_2, for example. These are coefficients of a base 2 polynomial. We have the following:

$$b = a + a\times 2 + a\times 2^2 = 1 + 0\times 2 + 1\times 2^2 = 5$$

The corresponding binary representation is (1 0 1). The presentation can be dramatically simplified for long binary information sequences. For an example of a binary information sequence of 128 bits, we need only a binary sequence of seven bits for presentation. In other words, the same information of a 128-bit binary sequence can be represented using seven bits based on a base 2 polynomial notation.

The modular addition between elements of a *GF(2^m)* is defined as the binary modular coefficient addition of the same power. The following is an example:

$$(1\ \ 0\ \ 1)+(1\ \ 1\ \ 0)=(1+0\times 2+1\times 2^2)+(1+1\times 2+0\times 2^2)=(0+1\times 2+1\times 2^2)=(0\ \ 1\ \ 1)$$

The modular multiplication between elements of a *GF(2^m)* is defined as element-to-element modular multiplication of two polynomials. Each element-to-element modular multiplication is carried out by the summation of powers of multiplicands. We need to associate each element with a power representation to carry out the multiplication (see Table 10.2).

Table 10.2 Power Representation for the Elements of GF(2^3)

Power	Polynomial	Binary
0	0	(0 0 0)
1	1	(1 0 0)
α	α	(0 1 0)
α^2	α^2	(0 0 1)
α^3	$1+\alpha$	(1 1 0)
α^4	$\alpha+\alpha^2$	(0 1 1)
α^5	$1+\alpha+\alpha^2$	(1 1 1)
α^6	$1+\alpha^2$	(1 0 1)

For this example, we can still associate α with 2^1 and α^2 with 2^2, but not α^i to 2^i for $i>2$. In fact, α^3 is calculated based on the primitive polynomial for *m=3*. A primitive polynomial of *m*, $p(X)$, divides X^n+1 for $n=2^m-1$. The primitive polynomial for *m=3* is

$$p(X)=1+X+X^3$$

By setting $p(\alpha)=1+\alpha+\alpha^3=0$, we can obtain a representation of α^3 in terms of α. We have

$$1+\alpha+\alpha^3+1+\alpha=1+\alpha$$
$$\alpha^3=1+\alpha$$

According to Table 10.2, we have

$$(1+\alpha^2)(1+\alpha)=1+\alpha+\alpha^2+\alpha^3=1+\alpha+\alpha^2+1+\alpha=\alpha^2$$

Division between two polynomials is better handled by their power presentation, for example

$$\frac{1+\alpha^2}{1+\alpha}=\frac{\alpha^6}{\alpha^3}=\alpha^3=1+\alpha$$

With these defined operations between polynomials, the encoding and decoding can be reduced to algebra operations and solving for the solution of a set of linear equations.

Table 10.3 lists primitive polynomials for the construction of power representations for m=4, 5, 6, 7, and 8.

Table 10.3 Primitive Polynomials

m	Primitive Polynomial
4	$1+X+X^4$
5	$1+X^2+X^5$
6	$1+X+X^6$
7	$1+X^3+X^7$
8	$1+X^2+X^3+X^4+X^8$

10.2 *Reed-Solomon Codes*

Reed-Solomon[3] codes are non-binary Bose, Chaudhuri, and Hocquenghem (BCH) codes. Bose, Chaudhuri, and Hocquenghem published two papers defining basic properties of the BCH cyclic code.[4,5] A code word can be generated by multiplying a message polynomial by a generator polynomial. The specification of the generator polynomial for BCH codes is relatively simple. A BCH code can be designed on the desired code size and code distance.

The generator polynomial of a t-error-correcting BCH code of length 2^m-1 can be found through the least common multiple of minimum polynomials of roots α^i, for $1\le i\le 2t$, where α^is are elements of GF(2^m). A minimal polynomial of α^i has the smallest degree of a polynomial whose coefficient is in GF(2). Minimal polynomials for α, α^3, and α^5 are $1+X+X^4$, $1+X+X^2+X^3+X^4$, and $1+X+X^2$, respectively.

The generator polynomial for the two-error-correcting BCH code of length $n=2^4-1=15$ is

$$g(x)=\left(1+X+X^4\right)\left(1+X+X^2+X^3+X^4\right)=1+X^4+X^6+X^7+X^8$$

The generator polynomial for the three-error-correcting BCH code, also of length $n=2^4-1=15$, is

$$g(x)=(1+X+X^4)(1+X+X^2+X^3+X^4)(1+X+X^2)=1+X+X^2+X^4+X^5+X^8+X^{10}$$

Reed-Solomon codes are used to encode symbols consisting of m binary digits into blocks consisting of $n_s=2^m-1$ symbols (or $n_b=m(2^m-1)$ binary digits). For $m=8$ $m=8$, the symbol block size is $n_s=255$ and the total number of bits in each block is $n_b=8\times255=2040$. For Reed-Solomon codes, the degree of generator polynomial is $r=n-k$ $r=n-k$ and the code word minimum distance is $d_{min}=r+1$, where n is the code word size and k is the message sequence size. The number of errors that can be corrected is defined by

$$t=\frac{d-1}{2}=\frac{r}{2}$$

Shorter than n_s or n_b Reed-Solomon codes can be derived also by using only a subset of the full size code. We can pre-define the last portion of the message as 0s. Therefore, these 0s can be automatically added at the receiver end without transmission of a full size code word. In other words, the code word size becomes $n'=n-k_z$ when the message size is $k-k_z$ by specifying k_z zeros at the end of the message. These k_z zeros need not be included in code words.

10.2.1 *Generating Reed-Solomon Code Words*

Reed-Solomon code words can be generated based on message sequences—$m(x)$, using generator polynomials—$g(x)$. In general, the generator polynomial of a BCH code is the lowest common multiple of minimum polynomials. For a Reed-Solomon code, the minimum polynomials of elements, α^is, of GF(2^m) all have the degree of 1. The Reed-Solomon generator polynomial has the following form:

$$g(x)=(x-\alpha)(x-\alpha^2)(x-\alpha^3)\ldots(x-\alpha^{d-1})$$

For an example of a binary representation with $d=3$ in GF(4), we have

$$g(x)=(x-\alpha)(x-\alpha^2)=x^2-(a+\alpha^2)x+\alpha^3=x^2+x+1$$

for

$$-(\alpha+\alpha^2)=\alpha+\alpha^2=\alpha(1+\alpha)=\alpha\alpha^2=\alpha^3=1 \text{ (see Table 10.4)}$$

because we always have $\alpha^{2^m-1}=1$ for $GF(2^m)$.

Table 10.4 Power Representation

Power	Polynomial	Binary
0	0	(0 0)
1	1	(1 0)
α	α	(0 1)
α^2	$1+\alpha$	(1 1)

For another example of a byte-based Reed-Solomon code with $d=3$ in GF(2^8), we have the following:

$$g(x)=(x-\beta)(x-\beta^2)=x^2+(\beta+\beta^2)x+\beta^3=x^2+(\alpha^7+\alpha^{14})x+\alpha^{21}=x^2+\alpha^{119}x+\alpha^{21}$$

$\beta=\alpha^7$ and $\alpha^7+\alpha^{14}=\alpha^{119}$ according to the GF(256) power representation in Table 10.4.

In practice, a systematic form cyclic code, including the Reed-Solomon code, $c(x)$, is generated in three steps:

1. Pre-multiply the message $m(x)$ by x^d.
2. Obtain the remainder, $r(x)$, by dividing $x^d m(x)$ by the generator polynomial $g(x)$.
3. Combine $r(x)$ and $x^d m(x)$.

We have

$$c(x)=r(x)+x^d m(x)=\text{Re } m\left[x^d m(x)/g(x)\right]+x^d m(x)$$

It is interesting to note that the code-generating method of

$$c(x)=\text{Re } m\left[x^d m(x)/g(x)\right]+x^d m(x)$$

is not the same as multiplying the message polynomial with the generator polynomial, $m(x)g(x)$. For an example of $m(x)=1+x+x^2$ and $g(x)=1+x+x^3$ in GF(2^4) with $d=3$, we have

$$m(x)g(x)=(1+x+x^2)(1+x+x^3)=1+x+x^3+x+x^2+x^4+x^2+x^3+x^5=1+x^4+x^5$$

and

$$\text{Rem}\left[\frac{x^3+x^4+x^5}{1+x+x^3}\right]+x^3+x^4+x^5=\text{Rem}\left[x^2+x+\frac{x}{1+x+x^3}\right]+x^3+x^4+x^5=x+x^3+x^4+x^5$$

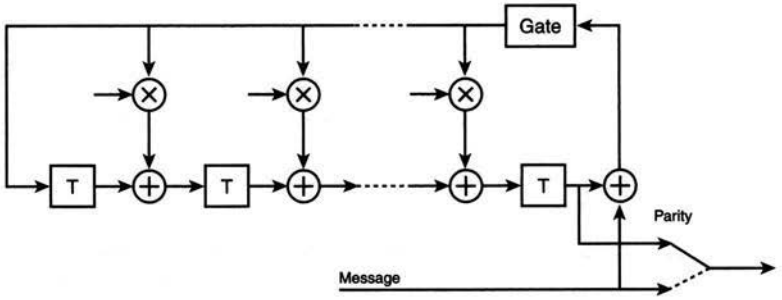
10.2.2 Reed-Solomon Encoder Implementation

The encoder circuit for the code-generating method

$$c(x)=\text{Re } m\left[x^d m(x)/g(x)\right]+x^d m(x)$$

can be conveniently implemented with a division circuit using shift registers, adders, and multipliers, as shown in Figure 10.1. The division circuits can be implemented in any number of bits or bytes, for example.

Figure 10.1 A Systematic Reed-Solomon Encoder



The encoding process is as follows:

1. With the gate turned on, the k information bytes are shifted into the circuit and simultaneously into the communications channel. Shifting the message $m(x)$ into the circuit from the front is equivalent to pre-multiplying $m(x)$ by x^d.
2. As soon as the complete message has entered the circuit, the n-k bytes in the register form the remainder and are thus the parity check bytes. The gate is then blocked.
3. The parity check bytes are shifted out to the channel.

10.2.3 Decoding of Reed-Solomon Code

The decoding of cyclic codes consists of three steps:

1. Syndrome computation

2. Association of the syndrome to an error pattern

3. Error correction

The syndrome computation for cyclic codes can be accomplished with a division circuit whose complexity is linearly proportional to the number of parity check digits.

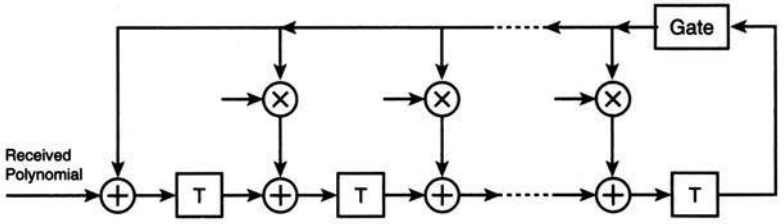
The association of the syndrome to an error pattern can be completely specified by a decoding table. A straightforward approach to the design of a decoding circuit is via a combinational logic circuit that implements the table-lookup procedure. The limit to this approach, however, is that the complexity of the decoding circuit tends to grow exponentially with the code length and the number of errors that we intend to correct.

The error correction step for the binary code case is simply adding (modulo-2) the error pattern to the receiver vector. For the Reed-Solomon code, we need the additional calculation of error values.

Syndrome Computation

The first step of decoding the syndrome computation, corresponding to the previously described encoder circuit, can be accomplished with another division circuit, but with the received polynomial as the input, as shown in Figure 10.2.

Figure 10.2 A Syndrome Computation Circuit



The received polynomial is shifted into the register with all stages initially set to 0. As soon as the entire received polynomial is shifted into the register, the contents in the register form the syndrome.

The syndrome computation can be obtained also by substituting the field elements α, $\alpha^2,\ldots\alpha^d$ into the received polynomial $v(x)=c(x)+e(x)$. For software implementation, substituting α^i into $v(x)$ is best accomplished in the following manner:

$$S_i = v(\alpha^i) = v_{n-1}(\alpha^i)^{n-1} + v_{n-2}(\alpha^i)^{n-2} + \ldots + v_1\alpha^i + v_0$$

$$= \left(\ldots\left(\left(v_{n-1}\alpha^i + v_{n-2}\right)\alpha^i + v_{n-3}\right)\alpha^i + \ldots + v_1\right)\alpha^i + v_0$$

The value n is the code word size. This computation takes $n-1$ addition and $n-1$ multiplication. It requires about $d(n-1)$ additions and nd multiplication to calculate all syndrome components, in which d is the minimum distance and equals the size of the parity check plus 1.

Syndrome/Error Pattern Association

In the second step of decoding, the error-location polynomial, $\sigma(x)$, is defined. We can rewrite the components of syndrome as

$$S_1 = \alpha^{j1}\alpha^{j2}+\ldots+\alpha^{jv}$$
$$S_2 = \left(\alpha^{j1}\right)^2+\left(\alpha^{j2}\right)^2+\ldots+\left(\alpha^{jv}\right)^2$$
$$S_3 = \left(\alpha^{j1}\right)^3+\left(\alpha^{j2}\right)^3+\ldots+\left(\alpha^{jv}\right)^3$$
$$\cdot$$
$$\cdot$$
$$S_d = \left(\alpha^{j1}\right)^d+\left(\alpha^{j2}\right)^d+\ldots+\left(\alpha^{jv}\right)^d$$

in which

$$0 \le j1 < j2 <\ldots< jv < n$$

and v is the number of errors.

Let

$$\beta_l = \alpha^{jl}$$

and the preceding expressions become

$$S_1 = \beta_1+\beta_2+\ldots+\beta_v$$
$$S_2 = \beta_1^{\,2}+\beta_2^{\,2}+\ldots+\beta_v^{\,2}$$
$$S_3 = \beta_1^{\,3}+\beta_2^{\,3}+\ldots+\beta_v^{\,3}$$
$$\cdot$$
$$\cdot$$
$$S_d = \beta_1^{\,d}+\beta_2^{\,d}+\ldots+\beta_v^{\,d}$$

We call $\beta_l = \alpha^{jl}$ error location numbers because they tell us the locations of the errors. Now we define the following error location polynomial:

$$\sigma(x) = (1+\beta_1 x)(1+\beta_2 x)\ldots(1+\beta_v x)$$
$$= \sigma_0 + \sigma_1 x + \sigma_2 x^2 + \ldots + \sigma_v x^v$$

The roots of $\sigma(x)$ are β_1^{-1}, β_2^{-1}, ..., β_v^{-1}, which are the inverse of the error location numbers. The coefficients of $\sigma(x)$ and the error location numbers are related by the following equations:

$$S_1 + \sigma_1 = 0$$
$$S_2 + \sigma_1 S_1 + 2\sigma_2 = 0$$
$$S_3 + \sigma_1 S_2 + \sigma_2 S_1 + 3\sigma_3 = 0$$
$$\vdots$$
$$S_v + \sigma_1 S_{v-1} + \ldots + \sigma_{v-1} S_1 + v\sigma_v = 0$$
$$S_{v+1} + \sigma_1 S_v + \ldots + \sigma_{v-1} S_2 + \sigma_v S_1 = 0$$

It requires about $2(d/2)^2$ addition and multiplication procedures to find $\sigma(x)$.

Error Correction

The third step of decoding is to determine the error location numbers by finding the roots of $\sigma(x)$ and to correct the errors in $v(x)$. The roots of $\sigma(x)$ can be found simply by substituting 1, α, $\alpha^2 \ldots \alpha^{n-1}$ into $\sigma(x)$. If α^l is a root of $\sigma(x)$, α^{n-l} is an error location number, and the received digit is an erroneous digit. This could be up to 2^m evaluations of d multiplication and addition procedures each, for which 2^m is the base of the Galois field.

For the Reed-Solomon code, we need one additional step of error value calculations—we form another polynomial:

$$z(x) = 1 + (S_1 + \sigma_1)x + (S_2 + \sigma_1 S_1 + \sigma_2)x^2 + (S_v + \sigma_1 S_{v-1} + \ldots + \sigma_{v-1} S_1 + \sigma_v)x^v$$

The error value at location $\beta_l = \alpha^{jl}$ is given by the following:

$$e_{jl} = \frac{z\left(\beta_l^{-1}\right)}{\prod_{\substack{i=1 \\ i \neq l}}^{v} \left(1 + \beta_i \beta_l^{-1}\right)}$$

Note
The preceding analysis assumes that modular or Galois field arithmetic operations are available.

10.2.4 DSL Applications of Reed-Solomon Codes

Reed-Solomon coded twisted-pair-based transmission systems have an SNR loss[6,7] relative to uncoded systems because of a higher signaling rate, therefore, a wider bandwidth where the portion of the expanded bandwidth usually has a lower SNR. For using eight bits per symbol in GF(2^8), the symbol error rate is $P_s = 1 - \left(1 - P_e\right)^8$, in which P_e is the bit error rate without coding. The bit error rate (BER) at the output of the (n,k) Reed-Solomon decoder is closely approximated by the following:[8]

$$BER = \frac{n-k+1}{2n} \sum \binom{n}{i} P_S^i \left(1 - P_S\right)^{n-i},$$

Figure 10.3 shows the BER at the output of a Reed-Solomon decoder with 10 percent redundancy, or coding overhead, based on a 16 QAM line code twisted-pair-based transmission system. These curves correspond to single, double, triple, and quadruple error corrections.

Coding overhead-caused SNR losses are included also. The coding loss for 10 percent overhead is about 0.83 dB for this particular twisted-pair loop environment. Table 10.5 summarizes coding losses for differing amounts of overhead.

Table 10.5 Coding Overhead SNR Loss

Percentage Redundancy	SNR Loss (dB)
2%	0.17
5%	0.42
10%	0.83
15%	1.15
20%	1.58

Figure 10.3 BER for 10 Percent Redundancy Reed-Solomon Code (Reproduced from *Bellcore T1E1.4/92-153 Contribution*)

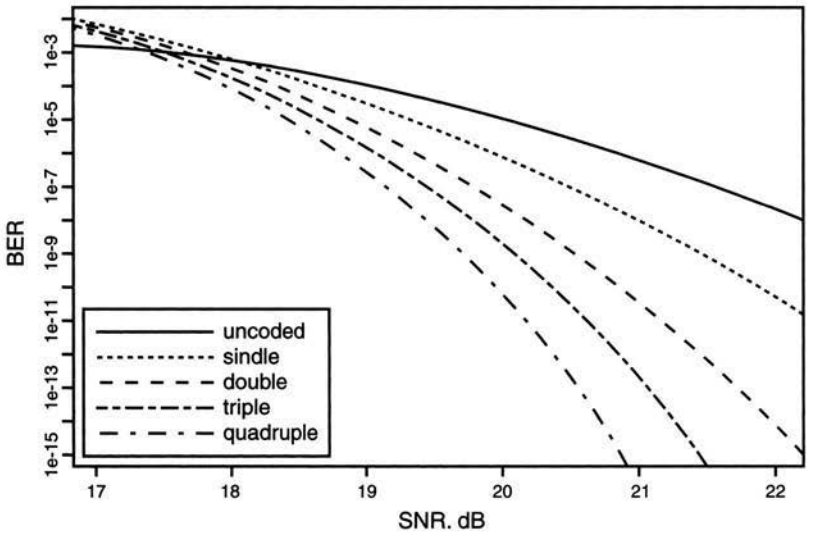


Figure 10.4 shows BERs at the output of a double-byte error correcting Reed-Solomon decoder with different redundancies also based on the 16 QAM line code twisted-pair-based transmission system. Respective coding overhead SNR losses for different amounts of redundancies are included also.

10.3 Convolution Codes

In contrast to the block code, a convolution code does not have a clean boundary between information bits and redundancy bits. In fact, the encoder runs like a sliding window across an information bit stream. A general binary convolution encoder can have a memory size of M with $M-1$ delay elements. k information bits are inputted to the encoder during every clock cycle. n outputs are generated every clock cycle based on modulo combinations of bit groups derived from k input information bits of different delays.

Encoder inputs and outputs are driven by the same synchronized clock. The ratio of the input information bit rate to the encoded output bit rate, k/n, is the coding rate. Figure 10.5 shows the general structure of a convolution encoder.

Figure 10.4 BER for Double-Error Correcting Reed-Solomon Code (Reproduced from *Bellcore T1E1.4/92-153 Contribution*)

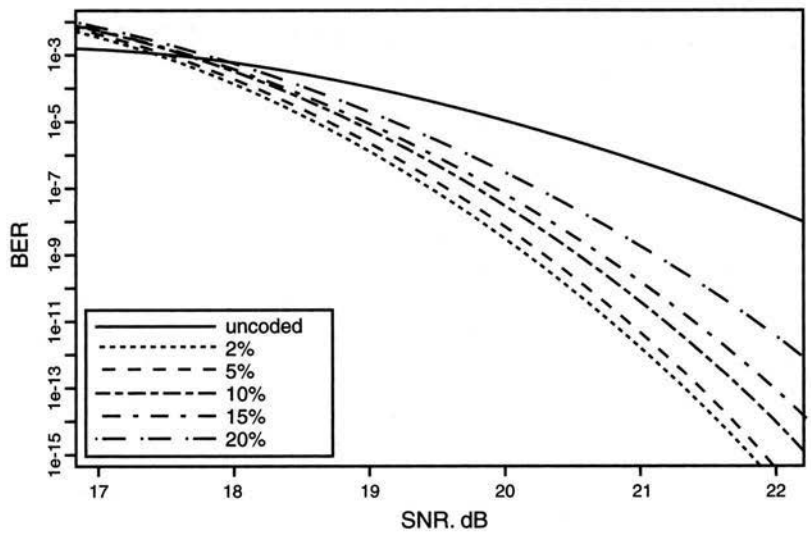
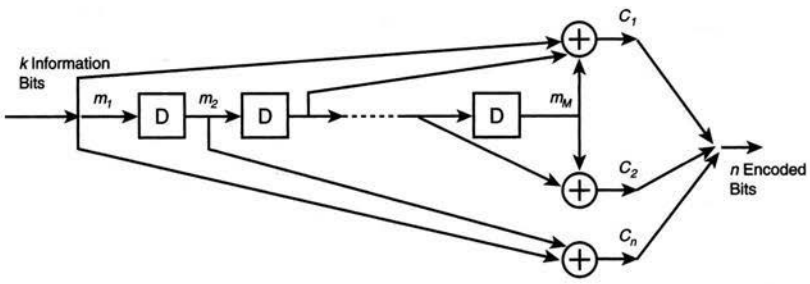


Figure 10.5 A Generic Convolution Encoder



Each encoded output bit is the exclusive-or combination of a selected number of input information bits with up to M clock cycle delays. We have

$$c_i = m_{11}g_{i11} \oplus m_{12}g_{i12} \oplus \ldots \oplus m_{1k}g_{i1k} \oplus m_{21}g_{i21} \oplus m_{22}g_{i22} \oplus \ldots \oplus m_{2k}g_{i2k} \oplus \ldots$$
$$\oplus m_{M1}g_{iM1} \oplus m_{M2}g_{iM2} \oplus \ldots \oplus m_{Mk}g_{iMk}$$

$$= \begin{bmatrix} m_1 & m_2 & \ldots & m_M \end{bmatrix} \begin{bmatrix} g_{i1} \\ g_{i2} \\ \ldots \\ g_{iM} \end{bmatrix} = mg_i$$

in which

$$m_i = \begin{bmatrix} m_{i1} & m_{i2} & \ldots & m_{ik} \end{bmatrix}$$
$$g_{ij} = \begin{bmatrix} g_{ij1} & g_{ij2} & \ldots & g_{ijk} \end{bmatrix}$$

In other words, m is a row vector of dimension $M \times k$, and g_i is a column vector of the same dimension.

For the steady state case (all delay stages are filled with information bits), the encoder output can be represented by

$$c = m \oplus g$$

in which

$$m = \begin{bmatrix} m_1 & m_2 & \ldots & m_M \end{bmatrix}$$
$$g = \begin{bmatrix} g_1 & g_2 & \ldots & g_n \end{bmatrix}$$
$$c = \begin{bmatrix} c_1 & c_2 & \ldots & c_n \end{bmatrix}$$

There are k input information bits and n encoded output bits for every clock cycle. Because $m_i = D^{i-1}m_1$ where $D^{i-1}m_1$ means i–1 clock cycle delayed version of m_1, we have $m = m_1\Gamma$, for which

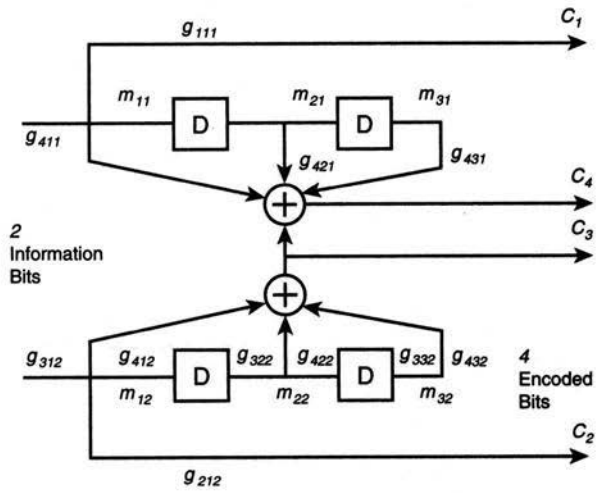
$$\Gamma = \begin{bmatrix} 1 & 0 & 0 & 0 & D & 0 & 0 & 0 & & D^{M-1} & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & D & 0 & 0 & & 0 & D^{M-1} & 0 & 0 \\ & & \ldots & & & & \ldots & & \ldots & & & \ldots & \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & D & & 0 & 0 & 0 & D^{M-1} \end{bmatrix}$$

Consequently, we have $c = m_1 \Gamma g$ or $c = m_1 g(D)$, for which

$$g(D) = \begin{bmatrix} g_{111} + g_{121}D + \ldots + g_{1M1}D^{M-1} & g_{211} + g_{221}D + \ldots + g_{2M1}D^{M-1} & g_{n11} + g_{n21}D + \ldots + g_{nM1}D^{M-1} \\ g_{112} + g_{122}D + \ldots + g_{1M2}D^{M-1} & g_{212} + g_{222}D + \ldots + g_{2M2}D^{M-1} & g_{n12} + g_{n22}D + \ldots + g_{nM2}D^{M-1} \\ & \ldots & \\ g_{11k} + g_{12k}D + \ldots + g_{1Mk}D^{M-1} & g_{21k} + g_{22k}D + \ldots + g_{2Mk}D^{M-1} & g_{n1k} + g_{n2k}D + \ldots + g_{nMk}D^{M-1} \end{bmatrix}$$

Many elements of $g(D)$ are zeros. A convolution code is generally characterized as an (n,k,M) convolution code. Figure 10.6 shows a simple example of a (4,2,3) convolution code.

Figure 10.6 An Example of a (4,2,3) Convolution Code



As represented in Figure 10.6:

m_{ij} denotes the jth information bit with $i-1$ delay.

g_{kij} denotes the weight for the kth encoded bit from m_{ij}.

The generator matrix for this convolution encoder is the following:

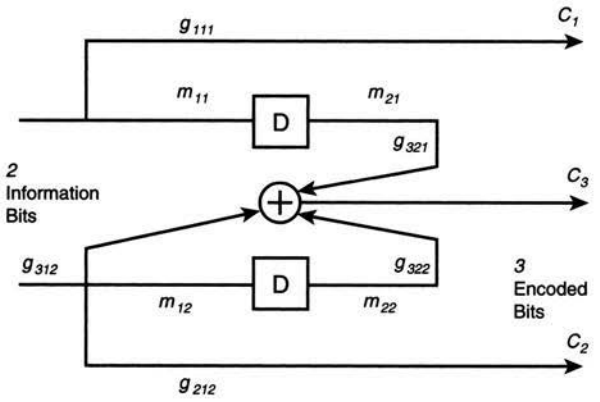
$$g = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix} \text{ or } g(D) = \begin{bmatrix} 1 & 0 & 0 & 1+D+D^2 \\ 0 & 1 & 1+D+D^2 & 1+D+D^2 \end{bmatrix}$$

In analogy to parity check code, the matrix of g or $g(D)$ is in systematic format also.

10.3.1 Convolution Encoder State

We can characterize a convolution encoder also as a finite state machine. The operating condition of a finite state machine can be defined by its input signal vector, its state, and its output signal vector. For the preceding example with four memory elements, we could have a total of 16 distinguishable states. For easy demonstration, we simplify the sample circuit by eliminating two delay elements and one output bit (see Figure 10.7).

Figure 10.7 A (3,2,2) Convolution Encoder



The generator matrix for this simplified convolution encoder is

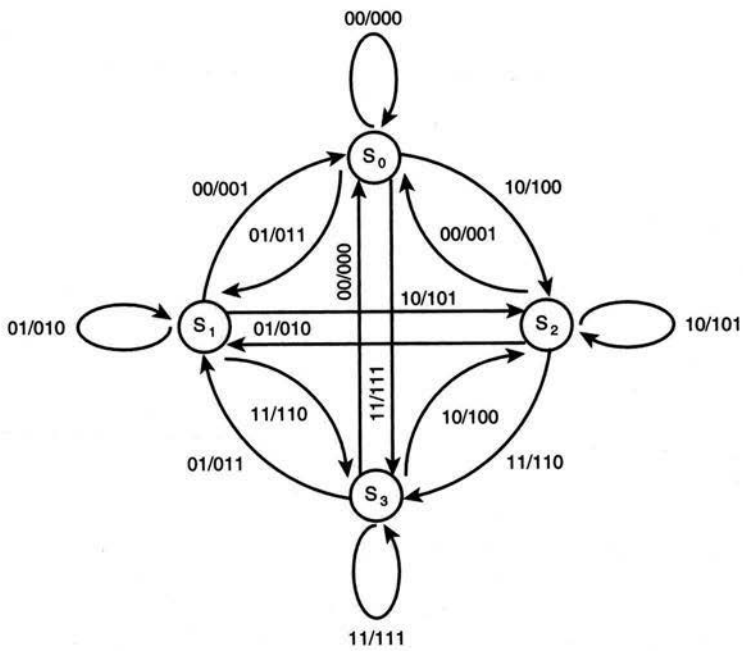
$$g = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix} \text{ or } g(D) = \begin{bmatrix} 1 & 0 & D \\ 0 & 1 & 1+D \end{bmatrix}$$

There are four states for this simplified encoder:

S_0 for $m_2 = 00$
S_1 for $m_2 = 01$
S_2 for $m_2 = 10$
S_3 for $m_2 = 11$

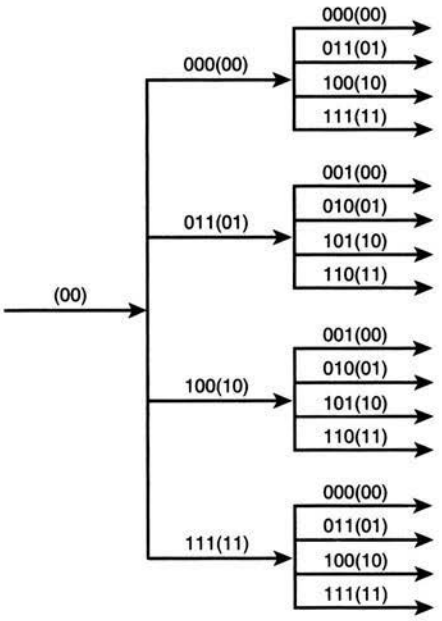
Figure 10.8 shows the state transition diagram of the simplified example of a (3,2,2) encoder. For a transition label of xx/yyy, xx represents the input and yyy represents the output.

Figure 10.8 State Transition Diagram for a (3,2,2) Encoder



Starting from the zero state, possible state transitions and output signal vectors also can be demonstrated with a tree diagram, as shown in Figure 10.9. For a branch label of yyy(zz), yyy represents the output and zz represents the state.

Figure 10.9 Tree Diagram for the (3,2,2) Encoder

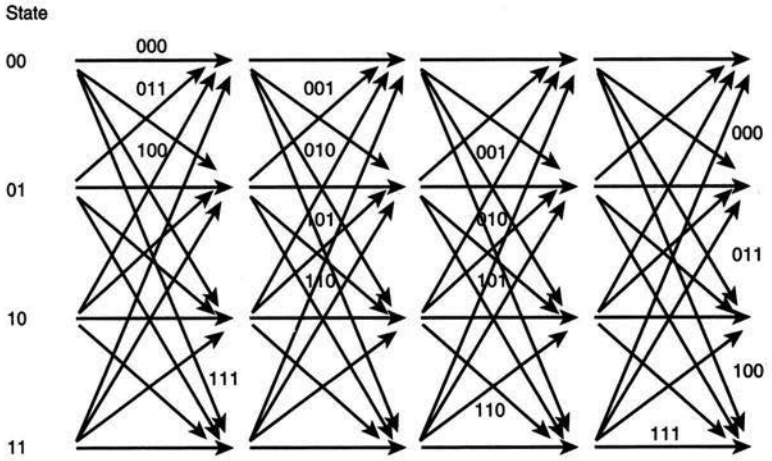


The information contained in a tree diagram also can be displayed with a Trellis diagram of less complexity, as shown in Figure 10.10 for the (3,2,2) convolution encoder.

The encoder output is labeled on the branch from the current state to the next state, as follows:

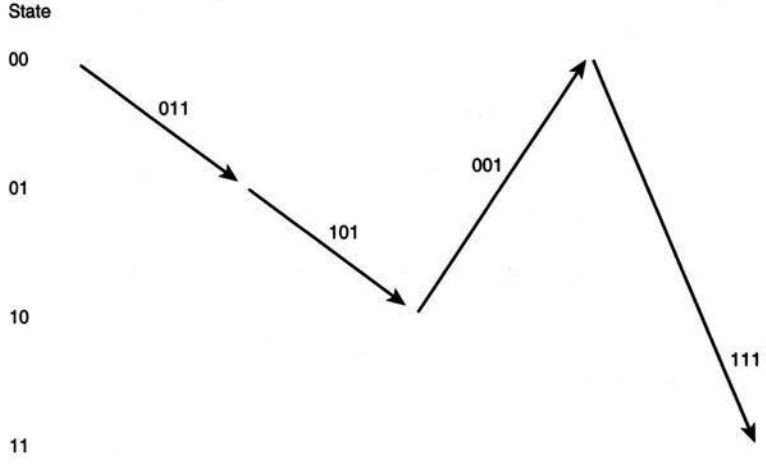
- Outputs from the first state, S_0, to other states are labeled between the first stage and the second stage.
- Outputs from the second state, S_1, to other states are labeled between the second stage and the third stage.
- Outputs from the third state, S_2, to other states are labeled between the third stage and the fourth stage.
- Outputs from the fourth state, S_3, to other states are labeled between the fourth stage and the fifth stage.

Figure 10.10 Trellis Diagram of a (3,2,2) Convolution Encoder



Corresponding input is defined by the encoder matrix or the state transition diagram. Figure 10.11 shows the Trellis for an input sequence of m=(01 10 00 11) from zero state. The output encoded sequence is the combination of all bit sequences labeled on branches—c=(011 101 001 111). For this systematic encoder, the first two bits of the output are the same as the input bits.

Figure 10.11 Trellis of a Message Sequence



Because of the state machine mechanism of the convolution encoder, the encoded bits as well as their sequences are very much related to the state and input information bits, which happened to be the same for this simple example. The information bits are recovered at a decoder based not only on received bits, which might be erroneous due to channel noise, but also on the likelihood of reconstruction of the state transition sequence or of the Trellis.

10.3.2 *Viterbi Decoding*

Assuming an origin from a known state, the maximum likelihood detection[9] or Viterbi[10] algorithm calculates distances—the total code differences—between the received code and codes leading to next possible states. After the accumulation of a certain number of calculations, the Viterbi algorithm then selects the sequence that has the minimal total distances as the right one. The information bits are then recovered based on the sequence, states it traveled through, and its corresponding input bits. The assumed known state can be identified once in awhile when all sequences pass the same state. In other words, a selection of the right sequence can be made when different sequences converge at a certain state.

Specifically, the Viterbi algorithm for the convolution code decoding can be described in the following steps:

1. For every group of received encoded bits $(r_1 \quad r_2 \quad \dots \quad r_n)$ during each clock cycle, we calculate branch metrics from surviving states to all possible next states:

$$B_M^{jl} = \sum_{i=1}^{n} r_i \oplus \beta_i^{jl}$$

B_M^{jl} is the branch metric from the surviving j state to the next l state.
$(\beta_1^{jl} \quad \beta_2^{jl} \quad \dots \quad \beta_n^{jl})$ are right convolution encoder output bits switching from the j state to the next l state.

2. All surviving path metrics are updated as

$$P_M^l = P_M^j + B_M^{jl}$$

3. A small number of paths that have the lowest path metric values are kept for the next iteration. When a path with a significantly lower path metric value is identified or a maximum number of iterations is reached, the path that has the lowest metric value is selected as the right encoded sequence.

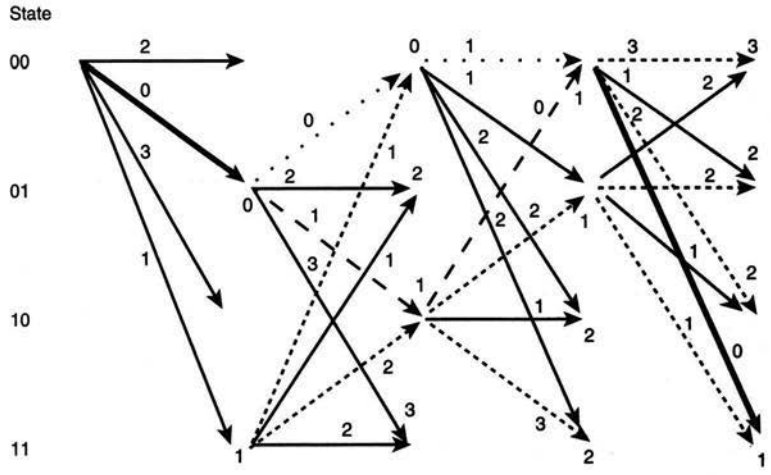
4. Original information bits are recovered according to the input bits, state, and output bits relationship.

The performance of the Viterbi algorithm can be improved if the received encoded bits are not quantized before the metric evaluation algorithm. The branch metric is then evaluated according to

$$B_M^{jl} = \sum_{i=1}^{n} \left(r_i - \beta_i^{jl}\right)^2$$

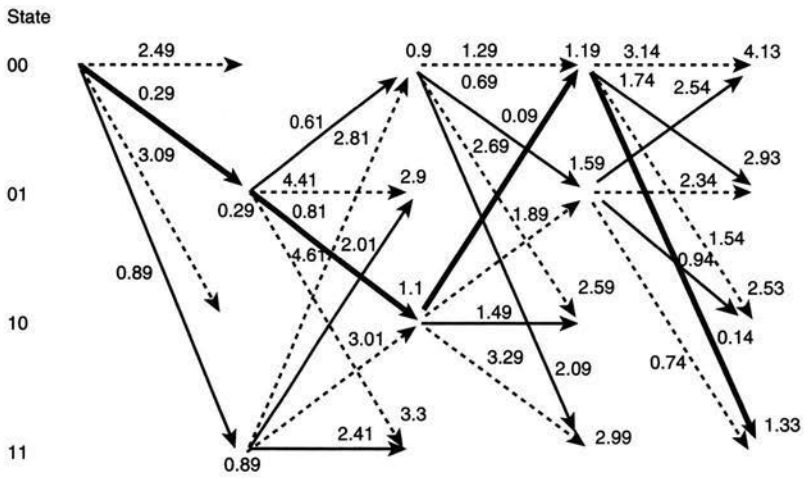
Assuming the received information is $r = (0.2, 1.4, 0.7 \quad 0.4, -0.3, 1.6 \quad -0.2, 0.2, 1.1 \quad 1.3, 0.9, 0.8)$ for the previous example, we have $\ddot{r} = (011 \quad 001 \quad 001 \quad 111)$ after quantization. Figure 10.12 shows branch and path metric values starting from a zero state.

Figure 10.12 Viterbi Decoding with Quantized Input



These regular dashed lines represent paths with a higher path metric value compared with those of alternative lines. The bold lines represent paths with the lowest path metric values. We can identify the lowest valued path metric at the end. There are, however, two paths producing the same lowest path metric value. In Figure 10.12, the two paths that produce the same lowest path metric values are shown in a dotted line and in a thick dashed line during the second and third stage of the Trellis diagram. Figure 10.13 shows calculations of branch and path metric values with non-quantized received information.

Figure 10.13 Viterbi Decoding with Soft Input



The right path, bold solid lines, is properly identified according to the lowest path metric value at the end.

10.4 *Trellis Code Modulation*

We have observed that the block codes, such as Reed-Solomon codes, can be used on a twisted-pair-loop channel for reducing the transmission error rate. The coding overhead due to parity check bits of a block code will increase the required transmission throughput. The additional transmission throughput requirement can be satisfied by either increasing the transmission bandwidth for a faster signaling rate or expanding the number of symbols for a denser constellation size.

For a non-bandwidth limited case, the transmission bandwidth can be increased for a slight reduction of signal-to-noise ratio of up to 1.58 dB, as for the Reed-Solomon code example. The resulting coding gain measured as a reduction of error rate or an equivalent SNR improvement could still be significant.

For the bandwidth limited case, however, the expansion of the number of symbols could cause a minimum of 3 or 6 dB of signal-to-noise ratio reduction for passband and baseband systems, respectively. Trellis code modulation is an effective coding method that can be used to produce significant coding gain for bandwidth limited cases.[11,12] Trellis code modulation combines a specially selected convolution code with a particular line symbol mapping to be

discussed in more detail in this section. The effective coding gain of a Trellis code modulation system can be in the range of between 3 to 6 dB, depending on the convolution encoder memory size.

The performance of a general purpose convolution code in its binary format is evaluated according to its minimum Hamming distance between code symbols. The direct application of a convolution code in conjunction with a particular baseband or passband system has two problems:

- A soft Viterbi algorithm instead of the conventional DFE is necessary to avoid the loss of information after the encoded information is received.
- There is no direct correlation between the minimum Hamming distance and the Euclidean distance that is crucial for the proper operation of the Viterbi algorithm.

Note

The minimum Hamming distance is related to the binary structure of a convolution code, while the Euclidean distance is related to the constellation structure of a line code.

For a simple (3,2,2) convolution code example of Figure 10.8, the minimum free distance, d_{free}, is 2, which is obtained by examining the corresponding state transition diagram and identifying the minimum weight of all paths that diverge from, and remerge with, the all-zero state, S_0. The asymptotic (or the maximum achievable) coding gain, when SNR is very high, is

$$\gamma = 10\log_{10}\frac{Rd_{free}}{2} = 10\log_{10}\frac{2\times 2}{2} = 3dB$$

R denotes the number of bits per symbol.

The average power for an n-level baseband signal with a voltage difference of 2 between adjacent signal levels is

$$P = \frac{n^2 - 1}{3}$$

For a normalized signal power, the voltage difference between adjacent signal levels for an n-level baseband system is

$$V_{diff} = 2\sqrt{\frac{3}{n^2 - 1}}$$

The normalized SNR for an n-level baseband system is

$$SNR = \frac{1}{\sigma^2}$$

The bit error rate for an n-level baseband system is

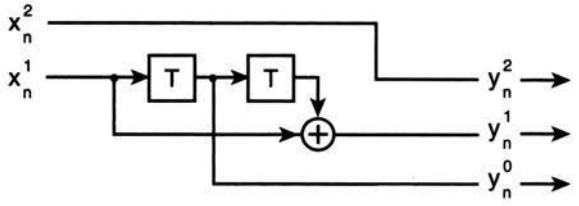
$$P_e = \frac{2}{\sqrt{2\pi}\sigma} \int_{\frac{V_{diff}}{2}}^{\infty} e^{-\frac{y^2}{2\sigma^2}} dy = \frac{2}{\sqrt{2\pi}} \int_{\frac{V_{diff}}{2\sigma}}^{\infty} e^{-\frac{x^2}{2}} dx = \frac{2}{\sqrt{2\pi}} \int_{\sqrt{\frac{3SNR}{n^2-1}}}^{\infty} e^{-\frac{x^2}{2}} dx$$

For a four-level baseband system, the SNR requirement for an error rate of 10^{-7} is about 21.5 dB. The error rate is about 9.3×10^{-3} under the same 21.5 dB SNR, but with a three-bit per symbol, eight-level baseband system to accommodate the extra bit generated by the convolution encoder. The simple convolution code, however, is not powerful enough to reduce the error rate from 9.3×10^{-3} to 10^{-7}. Therefore, the net coding gain is negative, and the asymptotic coding can never be realized.

10.4.1 *Achieving Significant Trellis Coding Gain*

Some special convolution codes have been identified to generate significant Trellis coding gain.[13] Figure 10.14 shows a good two-bit/three-bit or four-level/eight-level (3,2,2) convolution encoder.

Figure 10.14 A 2/3 Convolution Encoder



The generator matrix of this (3,2,2) convolution code to be used in a Trellis code modulation system is

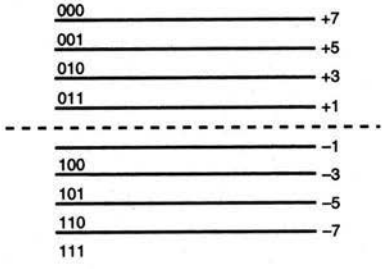
$$g = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

or

$$g(D) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1+D^{-2} & D^{-1} \end{bmatrix}$$

The corresponding eight signal levels should be mapped according to Figure 10.15, assuming the bits are arranged as $(y_2\, y_1\, y_0)$.

Figure 10.15 Eight-Level Bit-to-Signal Mapping



Asymptotic coding gain of about 3.3 dB can be realized by combining the specially selected convolution code and the eight-level bit-to-signal level mapping in conjunction with a soft Viterbi receiver.

The mapping is designed such that y_0 divides the eight-level line code into two four-level line code with wider distance between signal levels, y_1 further divides each four-level line code into two two-level line code, and y_2 selects the signal level among these two-level line codes.

The parity check matrix of this convolution code is

$$H(D) = \begin{bmatrix} 0 & D & 1+D^2 \end{bmatrix}$$

We have the orthogonality of

$$GH^T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1+D^2 & D \end{bmatrix} \begin{bmatrix} 0 \\ D \\ 1+D^2 \end{bmatrix} = \begin{bmatrix} 0 \\ D+D^3+D+D^3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

This parity matrix also leads to a systematic format implementation of the same encoder. Denote encoded bits as $\begin{pmatrix} y_2^i & y_1^i & y_0^i \end{pmatrix}$, where superscripts indicate the time index. We have

$$\begin{pmatrix} y_2^i & y_1^i & y_0^i \end{pmatrix} H^T = \begin{pmatrix} y_2^i & y_1^i & y_0^i \end{pmatrix} \begin{bmatrix} 0 \\ D \\ 1+D^2 \end{bmatrix} = y_1^{i-1} \oplus y_0^i \oplus y_0^{i-2} = 0$$

or

$$y_0^i = y_0^{i-2} \oplus y_1^{i-1}$$

Figure 10.16 shows the corresponding systematic or feedback format encoder.

Figure 10.16 The Systematic Format Implementation of a Convolution Encoder

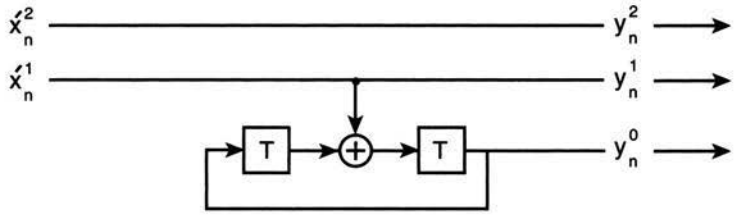
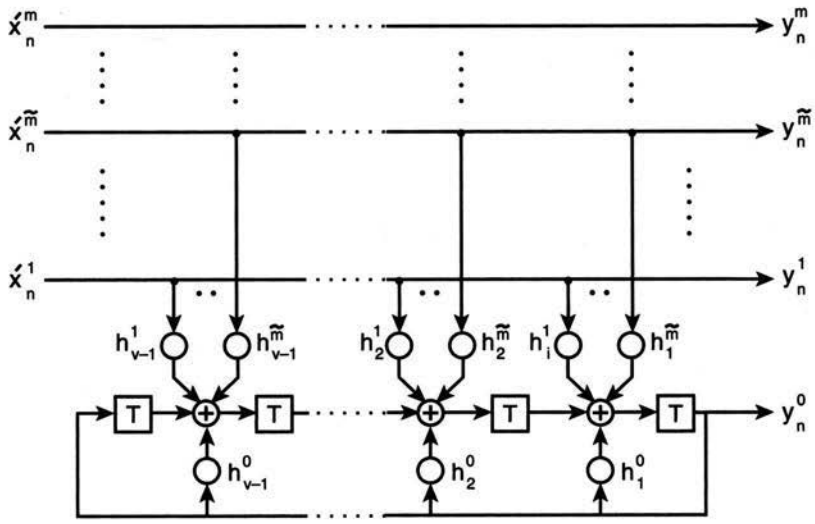


Figure 10.17 shows the general case of systematic format implementation of a convolution encoder with one extra coding bit.[14]

Figure 10.17 Systematic Convolution Encoder Structure



Other convolution encoders with more memory stage can provide even higher coding gains. For three memory stages, the parity check matrix for a convolution code capable of 3.7 dB of coding gain is

$$H(D) = \begin{bmatrix} 0 & D^2 & 1 + D^2 + D^3 \end{bmatrix}$$

The corresponding coding relationship is

$$\begin{pmatrix} y_2^i & y_1^i & y_0^i \end{pmatrix} H^T = \begin{pmatrix} y_2^i & y_1^i & y_0^i \end{pmatrix} \begin{bmatrix} 0 \\ D^2 \\ 1 + D^2 + D^3 \end{bmatrix} = y_1^{i-2} \oplus y_0^i \oplus y_0^{i-2} \oplus y_0^{i-3} = 0$$

or

$$y_0^i = y_0^{i-2} \oplus y_0^{i-3} \oplus y_1^{i-2}$$

For four memory stages, the parity check matrix for a convolution code capable of 4.1 dB of coding gain is as follows:

$$H(D) = \begin{bmatrix} 0 & D^2 & 1 + D + D^2 + D^4 \end{bmatrix}$$

The corresponding coding relationship is

$$\begin{pmatrix} y_2^i & y_1^i & y_0^i \end{pmatrix} H^T = \begin{pmatrix} y_2^i & y_1^i & y_0^i \end{pmatrix} \begin{bmatrix} 0 \\ D^2 \\ 1+D+D^2+D^4 \end{bmatrix} = y_1^{i-2} \oplus y_0^i \oplus y_0^{i-1} \oplus y_0^{i-2} \oplus y_0^{i-4} = 0$$

or

$$y_0^i = y_0^{i-1} \oplus y_0^{i-2} \oplus y_0^{i-4} \oplus y_1^{i-2}$$

For five memory stages, the parity check matrix for a convolution code capable of 4.5 dB of coding gain is as follows:

$$H(D) = \begin{bmatrix} 0 & D+D^4 & 1+D^2+D^3+D^5 \end{bmatrix}$$

The corresponding coding relationship is

$$\begin{pmatrix} y_2^i & y_1^i & y_0^i \end{pmatrix} H^T = \begin{pmatrix} y_2^i & y_1^i & y_0^i \end{pmatrix} \begin{bmatrix} 0 \\ D+D^4 \\ 1+D^2+D^3+D^5 \end{bmatrix} = y_1^{i-1} \oplus y_1^{i-4} \oplus y_0^i \oplus y_0^{i-2} \oplus y_0^{i-3} \oplus y_0^{i-5} = 0$$

or

$$y_0^i = y_0^{i-2} \oplus y_0^{i-3} \oplus y_0^{i-5} \oplus y_1^{i-1} \oplus y_1^{i-4}$$

For six memory stages, the parity check matrix for a convolution code capable of 4.9 dB of coding gain is as follows:

$$H(D) = \begin{bmatrix} 0 & D^3+D^4 & 1+D+D^2+D^5+D^6 \end{bmatrix}$$

The corresponding coding relationship is

$$\begin{pmatrix} y_2^i & y_1^i & y_0^i \end{pmatrix} H^T = \begin{pmatrix} y_2^i & y_1^i & y_0^i \end{pmatrix} \begin{bmatrix} 0 \\ D^3+D^4 \\ 1+D+D^2+D^5+D^6 \end{bmatrix} = y_1^{i-3} \oplus y_1^{i-4} \oplus y_0^i \oplus y_0^{i-1} \oplus y_0^{i-2} \oplus y_0^{i-5} \oplus y_0^{i-6} = 0$$

or

$$y_0^i = y_0^{i-1} \oplus y_0^{i-2} \oplus y_0^{i-5} \oplus y_0^{i-6} \oplus y_1^{i-3} \oplus y_1^{i-4}$$

For seven memory stages, the parity check matrix for a convolution code capable of 5.2 dB of coding gain is as follows:

$$H(D) = \begin{bmatrix} 0 & D + D^2 + D^3 + D^4 + D^5 + D^6 & 1 + D + D^3 + D^5 + D^6 + D^7 \end{bmatrix}$$

The corresponding coding relationship is

$$\begin{aligned} \begin{pmatrix} y_2^i & y_1^i & y_0^i \end{pmatrix} H^T &= \begin{pmatrix} y_2^i & y_1^i & y_0^i \end{pmatrix} \begin{bmatrix} 0 \\ D + D^2 + D^3 + D^4 + D^5 + D^6 \\ 1 + D + D^3 + D^5 + D^6 + D^7 \end{bmatrix} \\ &= y_1^{i-1} \oplus y_1^{i-2} \oplus y_1^{i-3} \oplus y_1^{i-4} \oplus y_1^{i-5} \oplus y_1^{i-6} \oplus y_0^i \oplus y_0^{i-1} \oplus y_0^{i-3} \oplus y_0^{i-5} \oplus y_0^{i-6} \oplus y_0^{i-7} = 0 \end{aligned}$$

or

$$y_0^i = y_0^{i-1} \oplus y_0^{i-3} \oplus y_0^{i-5} \oplus y_0^{i-6} \oplus y_0^{i-7} \oplus y_1^{i-1} \oplus y_1^{i-2} \oplus y_1^{i-3} \oplus y_1^{i-4} \oplus y_1^{i-5} \oplus y_1^{i-6}$$

Figure 10.18 shows the realizable coding gains of a four-level to eight-level Trellis coded baseband system: PAM stands for *Pulse Amplitude Modulation* for the conventional baseband line code.

10.4.2 Feedback/Feedforward Implementations

For each systematic or feedback implementation of a convolution encoder, we can find a corresponding feedforward implementation of the same convolution encoder by deriving the generator matrix, $G(D)$, from the parity check matrix, $H(D)$, according to $G(D)H^T(D) = 0$.[15] Some special techniques of finding feedforward implementation formats might be necessary for multistage convolution codes.[16]

Some convolution codes suitable for two-dimensional Trellis coded modulation are shown in feedforward and feedback implementation formats in Figures 10.19 and 10.20, respectively.

Figure 10.18 PAM Trellis Coding Gain[13]

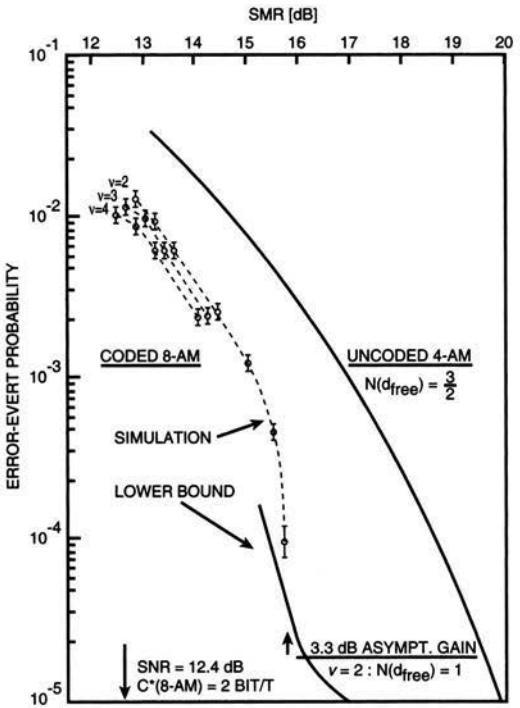


Figure 10.21 shows a 16 QAM (or 16 QASK, for Quadrature Amplitude Shift Keying, as labeled in the figure) constellation mapping division example.

The mapping is such designed that y^0 divides the 16-point constellation into two eight-point constellations with wider distance between adjacent constellation points, y^1 further divides each eight-point constellation into two four-point constellations, y^2 again divides each four-point constellation into two two-point constellations, and y^3 selects the constellation point among these two-point constellations.

Figure 10.22 shows realizable coding gains of a 8 PSK (Phase Shift Keying) or 8 AMPM (Amplitude and Phase Modulation) to 16 QAM (or 16 QASK as labeled in the figure) Trellis coded passband system.

Figure 10.19 Feedforward Convolution Encoder for Trellis Code Modulation

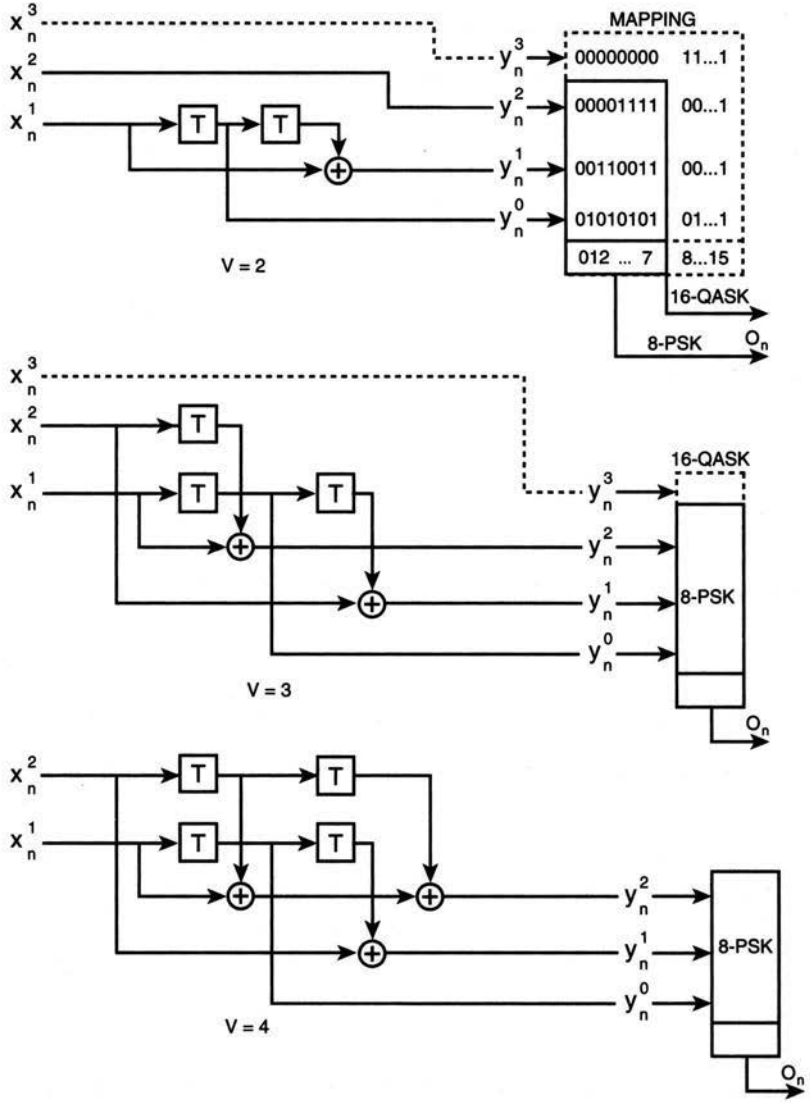
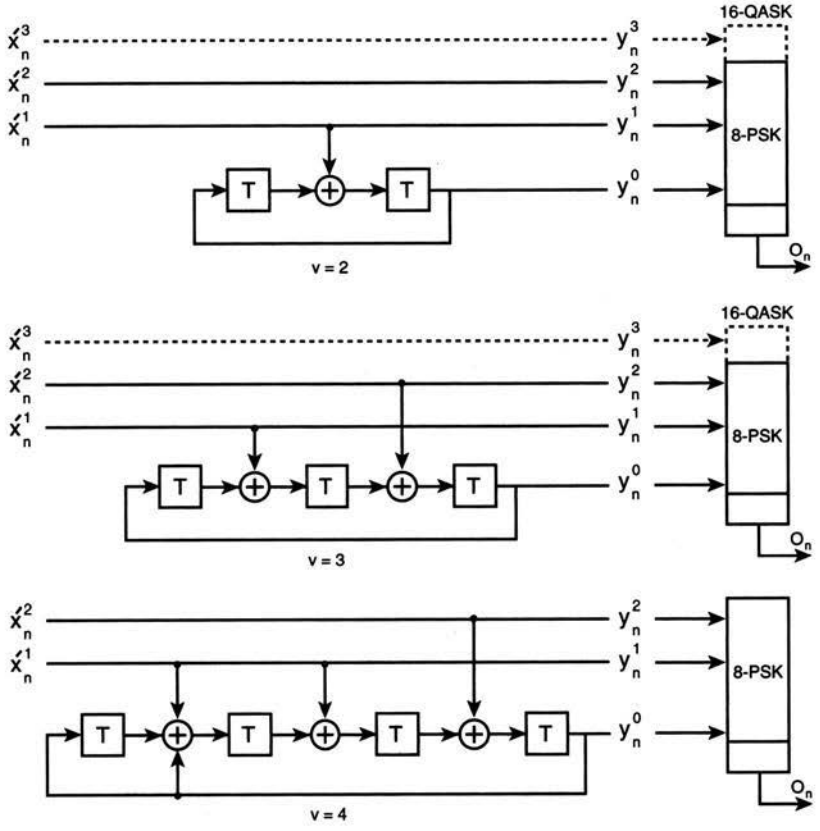


Figure 10.20 Feedback Convolution Encoder for Trellis Code Modulation



10.4.3 An Analytical Trellis Encoder Description

A Trellis encoder that includes the convolution encoder and the signal mapping also can be described in the analytical format.[17] For an example of the two memory stage four-level to eight-level Trellis encoder of Figure 10.14, we let the mapping output be a nonlinear function of input bits and state variables, as follows:

$$f\left(x_1^1 \quad x_0^1 \quad x_0^2 \quad x_0^3\right) = d_0 + d_1x_1^1 + d_2x_0^1 + d_3x_0^2 + d_4x_0^3 + d_{12}x_1^1x_0^1 + d_{13}x_1^1x_0^2 + d_{14}x_1^1x_0^3 + d_{23}x_0^1x_0^2$$
$$+d_{24}x_0^1x_0^3 + d_{34}x_0^2x_0^3 + d_{123}x_1^1x_0^1x_0^2 + d_{124}x_1^1x_0^1x_0^3 + d_{134}x_1^1x_0^2x_0^3 + d_{234}x_0^1x_0^2x_0^3 + d_{1234}x_1^1x_0^1x_0^2x_0^3$$

Figure 10.21 16 QAM Constellation Set Partitioning

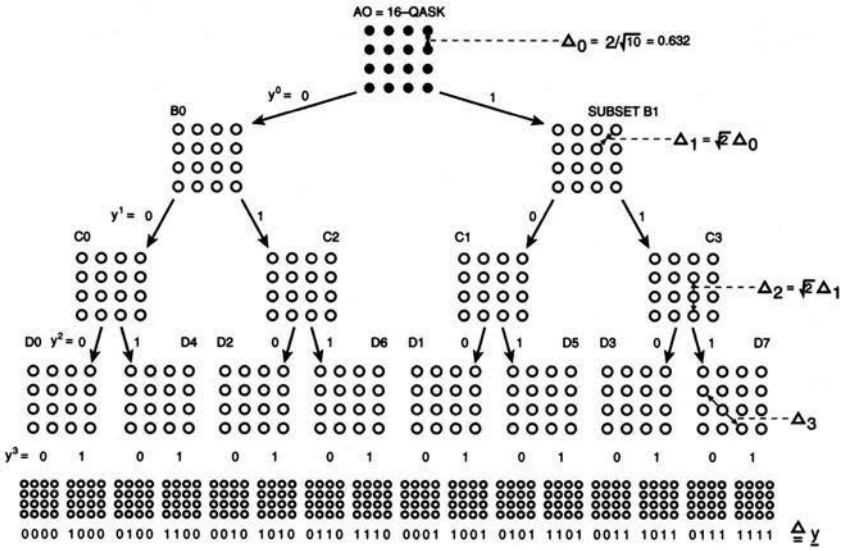
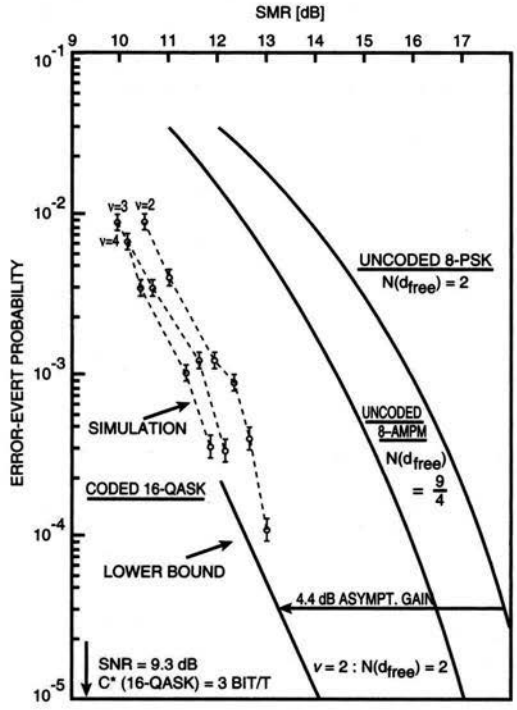


Figure 10.22 16 QAM Trellis Coding Gain[13]



The value x_1^1 corresponds to the input bit of x_n^2 in Figure 10.14, x_0^1 corresponds to the input bit of x_n^1 also in Figure 10.14, x_0^2 is the one stage delay version of x_0^1, and x_0^3 is the two stage delayed version of x_0^1. For the general case, we have the following:

$$d = \frac{1}{2^n} B^T f$$

With the example of n=4, we have

$$d = \begin{bmatrix} d_0 & d_1 & d_2 & d_3 & d_4 & d_{12} & d_{13} & d_{14} & d_{23} & d_{24} & d_{34} & d_{123} & d_{124} & d_{134} & d_{234} & d_{1234} \end{bmatrix}$$

$$f = \begin{bmatrix} f_1 \\ f_2 \\ \cdots \\ f_{2^n} \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ \cdots \\ f_{2^4} \end{bmatrix} \quad B = \begin{bmatrix} B_1 \\ B_2 \\ \cdots \\ B_{2^n} \end{bmatrix} = \begin{bmatrix} B_1 \\ B_2 \\ \cdots \\ B_{2^4} \end{bmatrix}$$

f_i is related to B_i as

$$f_i = f\left(x_1^1 \quad x_0^1 \quad x_0^2 \quad x_0^3\right)$$

for

$$x_1^1 = a_j^1,\ x_0^1 = a_k^1,\ x_0^2 = a_k^2$$

and

$$x_0^3 = a_k^3$$

when

$$B_i = \begin{bmatrix} 0 & x_1^1 & x_0^1 & x_0^2 & x_0^3 & x_1^1x_0^1 & x_1^1x_0^2 & x_1^1x_0^3 & x_0^1x_0^2 & x_0^1x_0^3 & x_0^2x_0^3 & x_1^1x_0^1x_0^2 & x_1^1x_0^1x_0^3 & x_1^1x_0^2x_0^3 & x_0^1x_0^2x_0^3 & x_1^1x_0^1x_0^2x_0^3 \end{bmatrix}$$

also for

$$x_1^1 = a_j^1,\ x_0^1 = a_k^1,\ x_0^2 = a_k^2,\ x_0^3 = a_k^3$$

According to the preceding relationship, we have the following for the example in Figure 10.14:

$$
\begin{aligned}
f_1 &= f(1\ \ 1\ \ 1\ \ 1) = -3 & f_9 &= f(1\ \ 1\ \ 1\ \ -1) = -7\\
f_2 &= f(-1\ \ 1\ \ 1\ \ 1) = 5 & f_{10} &= f(-1\ \ 1\ \ 1\ \ -1) = 1\\
f_3 &= f(1\ \ -1\ \ 1\ \ 1) = -7 & f_{11} &= f(1\ \ -1\ \ 1\ \ -1) = -3\\
f_4 &= f(-1\ \ -1\ \ 1\ \ 1) = 1 & f_{12} &= f(-1\ \ -1\ \ 1\ \ -1) = 5\\
f_5 &= f(1\ \ 1\ \ -1\ \ 1) = -1 & f_{13} &= f(1\ \ 1\ \ -1\ \ -1) = -5\\
f_6 &= f(-1\ \ 1\ \ -1\ \ 1) = 7 & f_{14} &= f(-1\ \ 1\ \ -1\ \ -1) = 3\\
f_7 &= f(1\ \ -1\ \ -1\ \ 1) = -5 & f_{15} &= f(1\ \ -1\ \ -1\ \ -1) = -1\\
f_8 &= f(-1\ \ -1\ \ -1\ \ 1) = 3 & f_{16} &= f(-1\ \ -1\ \ -1\ \ -1) = 7
\end{aligned}
$$

This is equivalent to

$$f = [-3\ \ 5\ \ -7\ \ 1\ \ -1\ \ 7\ \ -5\ \ 3\ \ -7\ \ 1\ \ -3\ \ 5\ \ -5\ \ 3\ \ -1\ \ 7]^T$$

The corresponding B matrix is

$$
B = \begin{bmatrix}
0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1\\
0 & -1 & 1 & 1 & 1 & -1 & -1 & -1 & 1 & 1 & 1 & -1 & -1 & -1 & 1 & -1\\
0 & 1 & -1 & 1 & 1 & -1 & 1 & 1 & -1 & -1 & 1 & -1 & -1 & 1 & -1 & -1\\
0 & -1 & -1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & -1 & -1 & 1\\
0 & 1 & 1 & -1 & 1 & 1 & -1 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & -1 & -1\\
0 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1\\
0 & 1 & -1 & -1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & -1 & -1 & 1 & 1\\
0 & -1 & -1 & -1 & 1 & 1 & 1 & -1 & 1 & -1 & -1 & -1 & 1 & 1 & 1 & -1\\
0 & 1 & 1 & 1 & -1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & -1 & -1 & -1\\
0 & -1 & 1 & 1 & -1 & -1 & -1 & 1 & 1 & -1 & -1 & -1 & 1 & 1 & -1 & 1\\
0 & 1 & -1 & 1 & -1 & -1 & 1 & -1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1\\
0 & -1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & 1 & -1\\
0 & 1 & 1 & -1 & -1 & 1 & -1 & -1 & -1 & -1 & 1 & -1 & -1 & 1 & 1 & 1\\
0 & -1 & 1 & -1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & 1 & -1 & 1 & -1\\
0 & 1 & -1 & -1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 & 1 & 1 & -1 & -1\\
0 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & 1
\end{bmatrix}
$$

We obtain

$$d = \frac{1}{16}[0 \quad 64 \quad 0 \quad -16 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 32 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]$$

Therefore, the (3,2,2) four-level to eight-level Trellis encoder can be described in the analytical form as follows:

$$f\left(x_1^1 \quad x_0^1 \quad x_0^2 \quad x_0^3\right) = 4x_1^1 - x_0^2 + 2x_0^1 x_0^3$$

The analytical format has the advantage of describing the Trellis encoder and corresponding signal level or constellation mapping in a single expression.

10.4.4 *Phase Invariant Trellis Encoders*

For a two-dimensional constellation, such as that of a 16 QAM, the parity of recovered information bits is related to the phase of the carrier in the receiver. The correct carrier phase is usually identified in the initialization process through the use of a known training sequence. Without the reference of the carrier phase, the 16 QAM constellation could become 90° phase invariant.

Note
The automatic recovery of the correct phase is important during a temporary transmission disconnect when the normal training process involving the use of the training sequence is avoided to minimize the recovery time.

Note
180° phase invariant Trellis encoders can be realized with a combination of a one-bit differential encoder, a convolution encoder, and a specially designed bit to symbol mapping.[18] Figures 10.23 and 10.24 show the encoder structure and corresponding constellation mapping for a four-state, 180° phase invariant Trellis encoder.

Figures 10.25 and 10.26 show the encoder structure and corresponding constellation mapping for an alternative four-state, 180° phase invariant Trellis encoder.

180° phase invariant Trellis encoders also can be realized with a combination of a one-bit differential encoder, a convolution encoder in the feedback format, and a specially designed bit-to-symbol mapping in feedback format. Figures 10.27 and 10.28 show the feedback implemented encoder structure and corresponding constellation mapping also for a four-state, 180° phase invariant Trellis encoder.

Figure 10.23 A Four-State 180° Phase Invariant Trellis Encoder

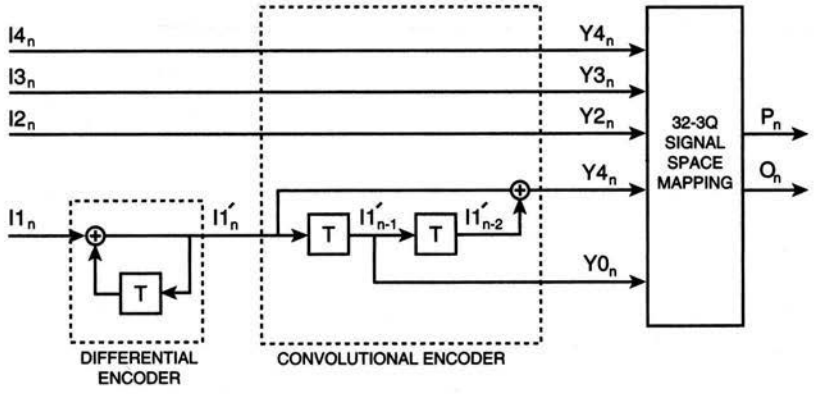


Figure 10.24 A Four-State 180° Phase Invariant Constellation Mapping

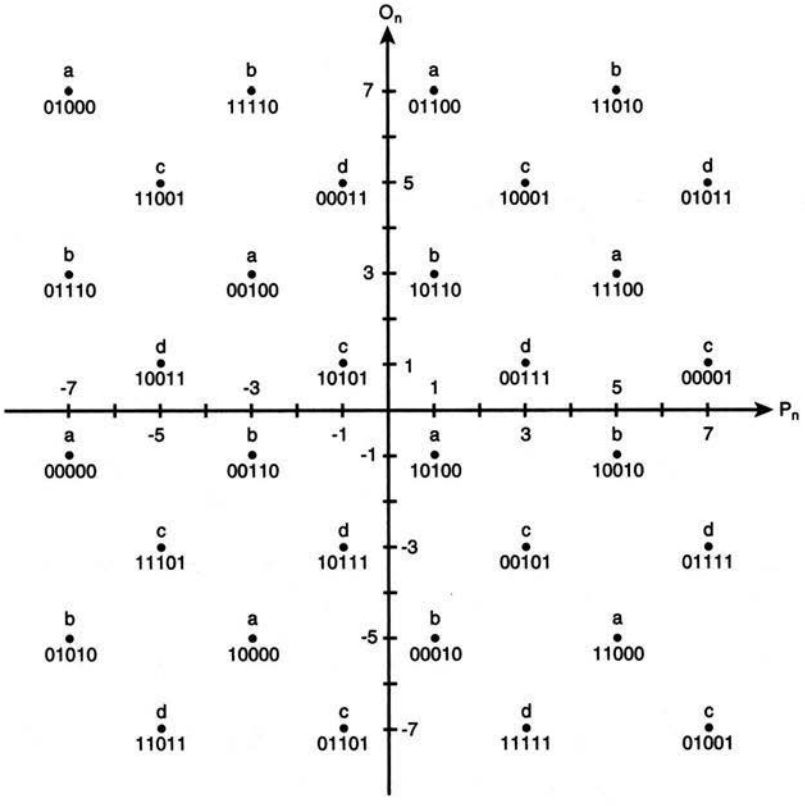


Figure 10.25 Another Four-State 180° Phase Invariant Encoder

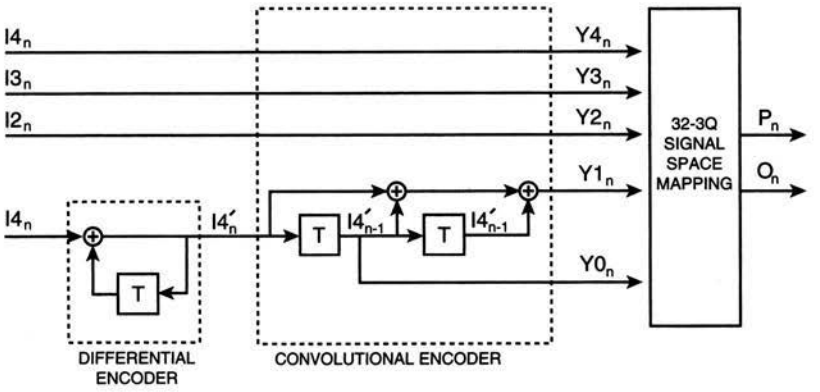


Figure 10.26 Another Four-State 180° Phase Invariant Constellation Mapping

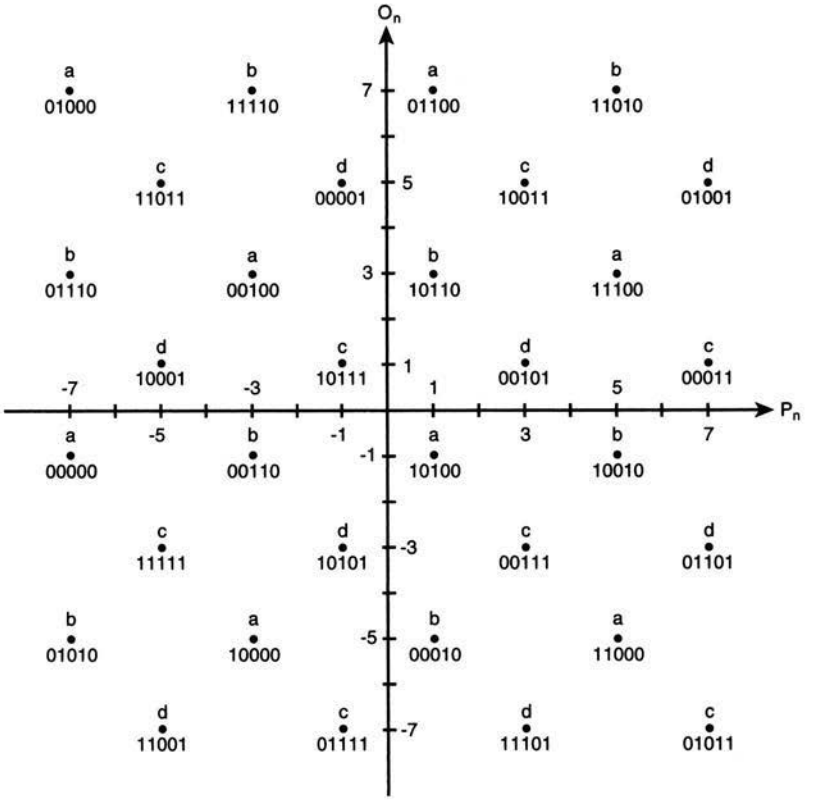


Figure 10.27 A Four-State 180° Feedback Phase Invariant Trellis Encoder

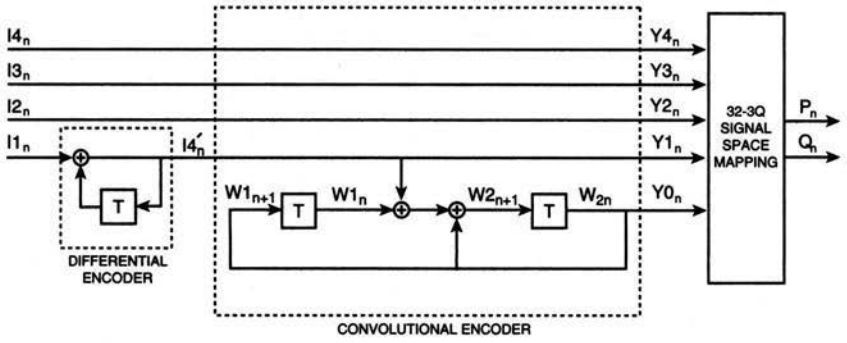
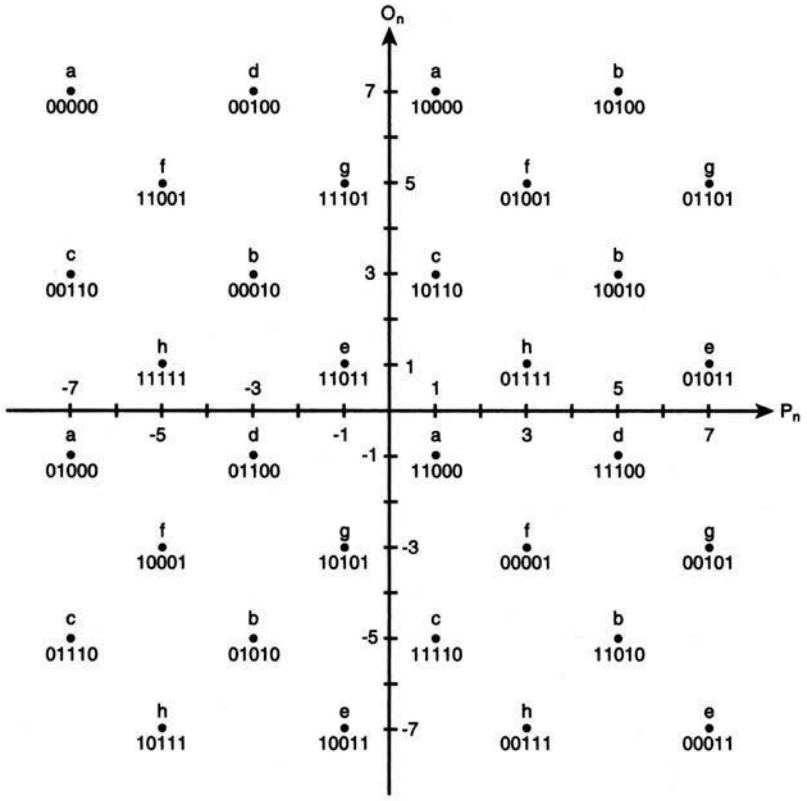
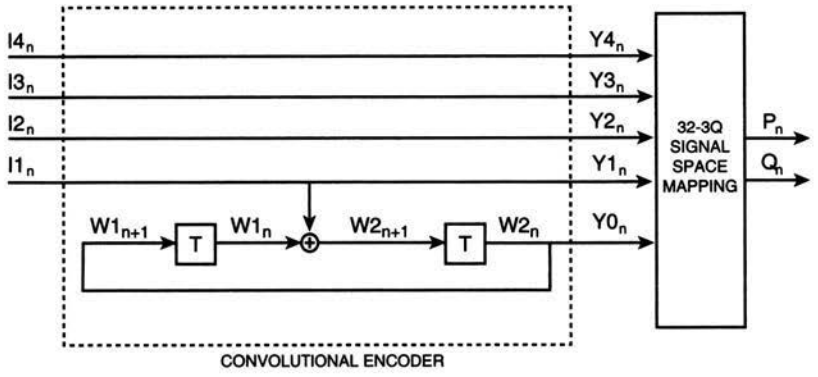


Figure 10.28 A Four-State 180° Feedback Phase Invariant Constellation Mapping



For the feedback implemented Trellis encoder, the differential encoder can sometimes be eliminated. The encoder structure shown in Figure 10.29 in conjunction with the bit mapping of Figure 10.28 also is a four-state, 180° phase invariant Trellis encoder.

Figure 10.29 A 180° Phase Invariant Feedback Trellis Encoder Without Differential Encoding



It has been shown that the 90° phase invariant Trellis encoder does not exist;[19] 90° phase invariant Trellis encoders can be realized, however, with non-linear Trellis code.[20] Figures 10.30 and 10.31 show the encoder structure and the constellation mapping of a 90° phase invariant non-linear Trellis encoder.

Figure 10.30 A 90° Phase Invariant Non-Linear Trellis Encoder

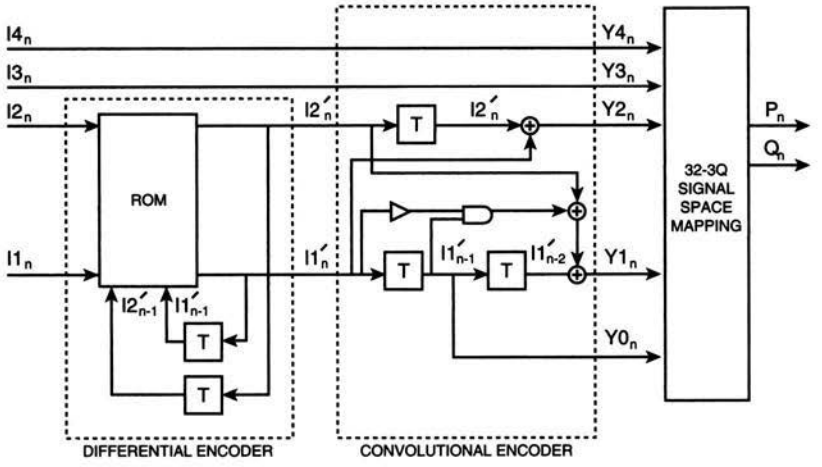
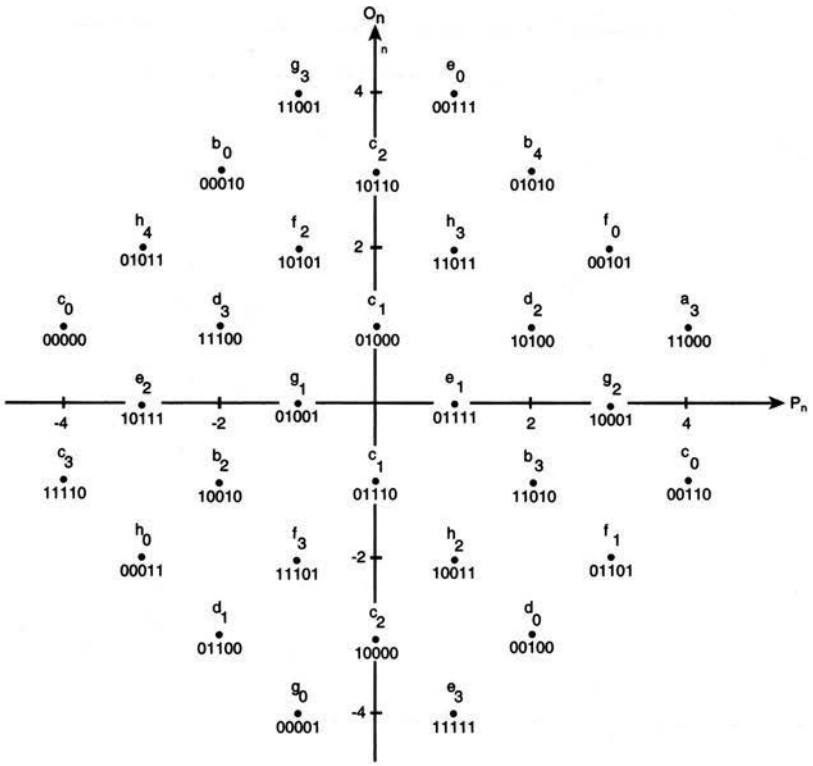


Figure 10.31 A 90° Phase Invariant Non-Linear Trellis Constellation Mapping



The relationship between two input bits

$$I1'_n \text{ and } I2'_n$$

and three state variables

$$I1'_{n-1},\ I1'_{n-2}, \text{ and } I2'_{n-1}$$

and three encoder output bits

$$Y0_n,\ Y1_n, \text{ and } Y2_n$$

are not linear. For example, for

$$I_{k1} = \left[I1'_{k1} \quad I1'_{k1-1} \quad I1'_{k1-2} \quad I2'_{k1} \quad I2'_{k1-1}\right] = [1 \quad 0 \quad 0 \quad 1 \quad 1]$$

we have

$$Y_{k1} = [Y2_{k1} \quad Y1_{k1} \quad Y0_{k1}] = [0 \quad 1 \quad 0]$$

and for

$$I_{k2} = [I1'_{k2} \quad I1'_{k2-1} \quad I1'_{k2-2} \quad I2'_{k2} \quad I2'_{k2-1}] = [0 \quad 1 \quad 0 \quad 0 \quad 0]$$

we have

$$Y_{k2} = [Y2_{k2} \quad Y1_{k2} \quad Y0_{k2}] = [0 \quad 1 \quad 1]$$

However, if we combine I_{k1} and I_{k2} together as the input vector, $I_{k1} \oplus I_{k2} = [1 \quad 1 \quad 0 \quad 1 \quad 1]$, the output is $[0 \quad 1 \quad 1]$, which is not the combination of output vectors, $Y_{k1} \oplus Y_{k2} = [0 \quad 1 \quad 0] \oplus [0 \quad 1 \quad 1] = [0 \quad 0 \quad 1]$, due to each individual input vector.

10.4.5 *Multidimensional Trellis Codes*

Multidimensional Trellis codes[21] also can be realized by Trellis encoding of a *multidimensional constellation*. A multidimensional constellation can be realized by combining a multiple of one- or two-dimensional data symbols. For an example, a four-dimensional constellation can be formed by combining two two-dimensional data symbols. If only one extra Trellis coding bit is introduced for a multidimensional constellation, the amount of SNR reduction can be relatively low. The discovery and the definition of a multidimensional Trellis encoder could be quite involved, however.

End Notes

1. S. Lin and D. J. Costello, Jr., *Error Control Coding: Fundamentals and Applications* (Prentice-Hall, 1983).

2. E. Biglieri, D. Divslar, P. J. McLane, and M. K. Simon, *Introduction to Trellis-Coded Modulation with Applications* (Macmillan Publishing Company, 1991).

3. I. S. Reed and G. Solomon, "Polynomial Codes Over Certain Finite Fields," *J. Soc. Ind. Appl. Math.*, 9 (June 1960): 300–304.

4. A. Hocquenghem, "Codes corecteurs d'erreurs," *Chiffres*, 2 (1959): 147–156.

5. R. C. Bose and D. K. Ray-Chaudhuri, "On a Class of Error Correcting Binary Group Codes," *Inf. Control*, 3 (March 1960): 68–79.

6. K. J. Kerpez, "Forward Error Correction for Asymmetric Digital Subscriber Lines (ADSL)," *Bellcore Globecom '91 Proceedings* (November 1991).

7. K. J. Kerpez, "Coding for ADSL," *Bellcore T1E1 Contribution*, T1E1.4/92-153 (August 1992).

8. R. H. Deng and D. J. Costello, "High Rate Concatenated Coding Systems Using Bandwidth Efficient Trellis Inner Codes," *IEEE Trans. Commun.*, vol. 37 (May 1989): 420-427.

9. G. D. Forney, Jr., "Maximum Likelihood Sequence Estimation of Digital Sequences in the Presence of Intersymbol Interference," *IEEE Trans. Inf. Theory*, IT-18 (May 1972): 363–378.

10. A. J. Viterbi, "Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm," *IEEE Trans. Inf. Theory*, IT-13 (April 1967): 260–269.

11. G. Ungerboeck, "Trellis-Coded Modulation with Redundant Signal Sets, Part I: Introduction," *IEEE Communication Magazine*, vol. 25 (February 1987): 5–11.

12. G. Ungerboeck, "Trellis-Coded Modulation with Redundant Signal Sets, Part I: State of The Art," *IEEE Communication Magazine*, vol. 25 (February 1987): 12–21.

13. G. Ungerboeck, "Channel Coding with Multi-Level/Phase Signals," *IEEE Trans. Information Theory*, vol. IT-28 (January 1982): 55–67.

14. G. D. Forney, Jr., "Convolution Codes I: Algebraic Structure," *IEEE Trans. Information Theory*, IT-16 (November 1970): 720–738.

15. E. Biglieri, D. Divsalar, P. J. McLane, and M. K. Simon, *Introduction to Trellis-Coded Modulation with Applications* (Macmillan, 1991).

16. J. E. Porath, "Algorithms for Converting Convolution Codes from Feedback to Feedforward Form and Vice Versa," *Electron. Lett.* (July 1989).

17. A.R. Calderbank and N.J.A. Sloane, "New Trellis Codes Based on Lattices and Cosets," *IEEE Trans. Information Theory*, vol. IT-33 (1987): 177–195.

18. L. F. Wei, "Rotationally Invariant Convolutional Channel Coding with Expanded Signal Space, Part I: 180°," *IEEE JSAC*, vol. SAC-2, no. 5 (September 1984): 659–671.

19. IBM Europe, "Trellis-Coded Modulation Schemes for Use in Data Modems Transmitting 3–7 Bits per Modulation Interval," *CCITT SG XVII Contribution COM XVII*, no. D114 (April 1983).

20. L. F. Wei, "Rotationally Invariant Convolutional Channel Coding with Expanded Signal Space, Part II: Nonlinear Codes," *IEEE JSAC*, vol. SAC-2, no. 5 (September 1984): 672–686.

21. L.F. Wei, "Trellis Coded Modulation with Multi-Dimensional Constellations," *IEEE Trans. Information Theory*, vol. IT-33 (1987): 483–501.

PART II

DSL Systems: System and Transceiver Architectures

11 DSL (Digital Subscriber Lines) 293

12 HDSL (High-Bit-Rate Digital Subscriber Lines) 347

13 ADSL (Asymmetrical Digital Subscriber Lines) 427

14 SHDSL (Single-Pair High-Bit-Rate Digital Subscriber Lines) 517

15 VDSL (Very-High-Bit-Rate Digital Subscriber Lines) 565

Glossary 601

CHAPTER 11

DSL (Digital Subscriber Lines)

Digital Subscriber Lines as a transmission technology was originally developed for the *ISDN (Integrated Services Digital Network) Basic Rate Access channel.* The name *DSL* has been used to describe the transmission technology or physical layer for the ISDN Basic Rate Access channel. Most recently, DSL, as well as xDSL, has been used as a generic name to describe any DSL system. The DSL transmission technology for the ISDN Basic Rate Access channel is sometimes referred to as IDSL. In this chapter, we use *DSL* to describe the physical layer for the ISDN Basic Rate Access channel. In contrast, we use *DSL system* as a generic name in this book.

DSL for the ISDN Basic Rate Access channel is the first bandwidth efficient telephone loop transmission system relying on advanced digital signal processing techniques. The development of DSL has involved the extensive examination of up-to-date advanced signal-processing algorithms in conjunction with the evaluation of capable semiconductor technologies. Specifically, the development of DSL went through initial line code proposal, transmission performance simulation study, and hardware prototype testing stages. Similar stages have been repeated by the development of other DSL systems.

This chapter covers the development history, transmission performance, and interface standards of Digital Subscriber Lines, including:

- *DSL technology development.* This section reviews the development history of DSL. It also discusses different duplex methods, including *Frequency Division Multiplex* (FDM), *Time Compressed Multiplex* (TCM), and *echo cancellation.* You also will have the opportunity to study *candidate line codes* and their transmission characteristics.

- *Transceiver structure and line code.* This section discusses general transceiver architecture and transmission characteristics of the *2B1Q DSL line code.* DSL equalizer and echo canceller requirements are calculated. DSL semiconductor technologies and implementation alternatives are reviewed.
- *Performance objectives.* DSL test loop configurations and corresponding transfer functions as well as NEXT models are presented. DSL transmission performances are simulated for these test loops under the self-NEXT noise condition.
- *Network interfaces.* Possible CO as well as *Digital Loop Carrier* (DLC) interfaces for DSL are discussed. The customer end S and T interfaces, including the framing structure, are also described.
- *DSL standards.* Key features of the ANSI 2B1Q DSL standard, including bit-to-symbol encoding, pulse mask, power spectrum mask, test configuration, frame structure, scrambler, and startup timing sequence, are revealed.

11.1 DSL Technology Development

A Digital Subscriber Line (DSL), as defined for ISDN Basic Access,[1,2] provides transmission capability for a single ISDN Basic Access customer over a nonloaded, two-wire telephone loop.[3,4] The DSL has a transmission throughput of 160 kbps that consists of a payload throughput of 144 kbps and an overhead throughput of 16 kbps. The payload throughput is divided into two B channels, each of which is 64 kbps, and one D channel of 16 kbps. The B channel was designed for digital telephony, and the D channel was designed for signaling and packet applications.

The transmission throughput of DSL is symmetrical in downstream (from a CO to a subscriber) and upstream (from a subscriber to a CO) directions. The DSL technology was developed at the ANSI working group T1D1.3, mainly during 1985 and 1986. An initial version of the standard document for DSL was delivered during 1988.[5]

In the early standardization process, four-wire transmission was ruled out because of the large plant cost that it could incur. There are basically three approaches for the two-wire, full-duplex transmission:

- Frequency Division Multiplex (FDM)
- Time Compression Multiplex (TCM)
- Echo cancellation (EC)

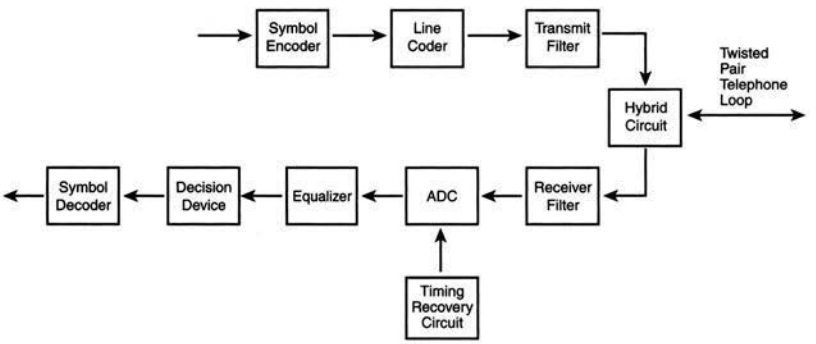
The debate on the choice between TCM and EC approaches to DSL provision has been in progress for some time. The central issues in the comparison of these two approaches were transmission loss, echo level, compatibility with other systems, and system complexity. In summary, the TCM system has the advantage of not requiring an echo canceller, as the separation of the different transmission directions occurs in time. This advantage diminishes as economically realized echo cancellers are made available by advances in *Very Large Scale Integrated-Circuits* (VLSI) technology. The EC system, while potentially more complex, uses less than 50 percent of the transmission bandwidth of its TCM counterparts.

The engineering trade-off between these competing technologies resulted in a decision by the ANSI T1D1.3 working group in favor of the adoption of the echo cancellation technique as the DSL transmission technology standard. One of the major reasons why TCM was not chosen as a standard in the U.S. had to do with synchronization. ISDN signals originating from different COs could share the same twisted-pair cable. This requires different COs to be synchronized for all DSL systems, which was not considered feasible in the U.S. telephone network. However, synchronization was feasible in Japan, and TCM was used for a Japanese version of DSL.

11.1.1 FDM Transmission for DSL

Figure 11.1 shows the block diagram of an FDM transceiver. With FDM, the two directions of transmission are assigned to different frequency bands. With the same line code and, therefore, symmetrical bandwidth, the range of the system was thus determined by the range of the lossier high frequency band. It was considered then that the FDM had a heavy range penalty and, therefore, the FDM was ruled out for the DSL. FDM for DSL systems is also sometimes called FDD for *Frequency Division Duplex.*

Figure 11.1 FDM Transceiver Block Diagram

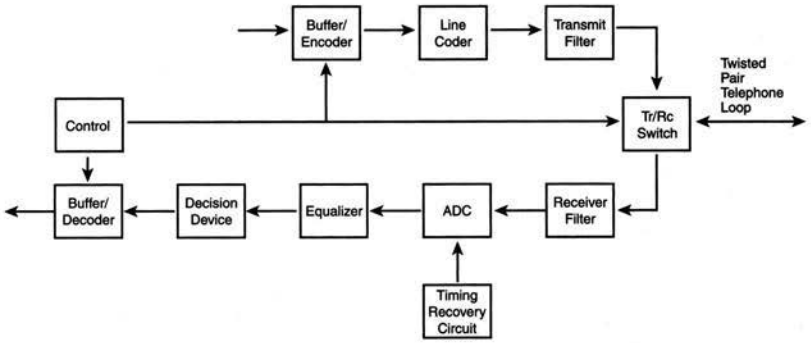


11.1.2 TCM Transmission for DSL

Figure 11.2 shows the block diagram of a TCM transceiver. In TCM, the two directions of transmission are separated in time. With TCM, data transmission actually occurs in half-duplex mode, though to the user it appears to be full-duplex because the gap is smoothed with buffers.

The quiescent period between bursts must be long enough to allow for slight errors in synchronization between opposite ends of the transmission path and to permit transients to die down sufficiently. This requires that the baud rate be approximately 2.25 times that which is required by the throughput for a particular line code. TCM for DSL systems is also sometimes called TDD for *Time Division Duplex.*

Figure 11.2 TCM Transceiver Block Diagram



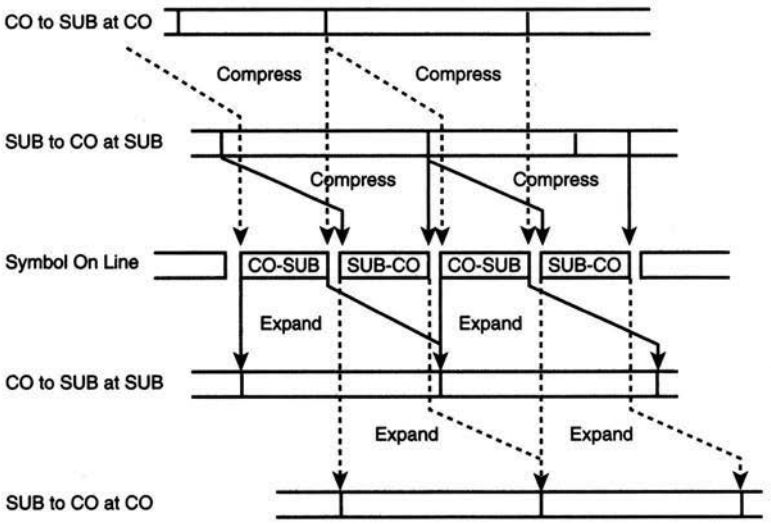
Bursts of transmission alternate between the transmitters on opposite ends of the line, as shown in Figure 11.3. The first row represents continuous data to be transmitted from a CO to a subscriber. The compressed CO-to-subscriber data is shown in the third row. The second row represents continuous data to be transmitted from a subscriber to a CO. The compressed subscriber-to-CO data is shown in the third row. The fourth row is the decompressed CO-to-subscriber data arrived at the subscriber end. The fifth row is the decompressed subscriber-to-CO data arriving at the CO end.

11.1.3 EC-Based Transmission for DSL

Figure 1.4 shows the block diagram of an echo-canceller-based transceiver. The two directions of transmission are separated by a hybrid circuit and the echo canceller. The hybrid circuit can usually provide only about 15 dB of separation. An echo cancellation level of about 60 dB by the echo canceller is required to remove the residual echo. In other words,

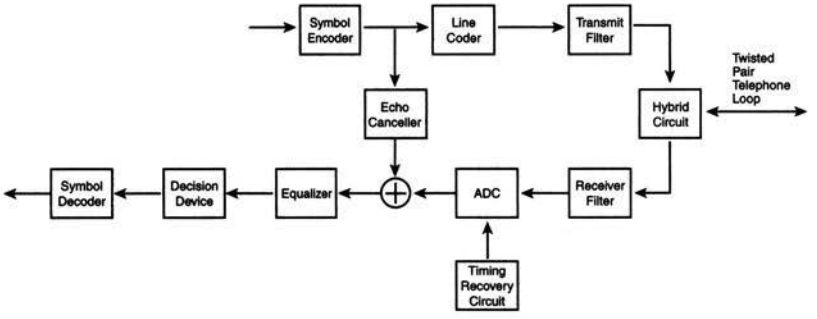
75 dB of combined echo-path attenuation is required to make the residual echo level much lower than the weak received signal level over a long telephone subscriber loop.

Figure 11.3 TCM Burst Principle



The echo canceller is based on a relatively stable transmit/receive clock. Otherwise, the adaptation of echo canceller coefficients might not be fast enough under the full-duplex operation. The transmit clock and the receiver clock can be locked using loop timing, usually at the customer end.

Figure 11.4 Echo Canceller Transceiver Block Diagram



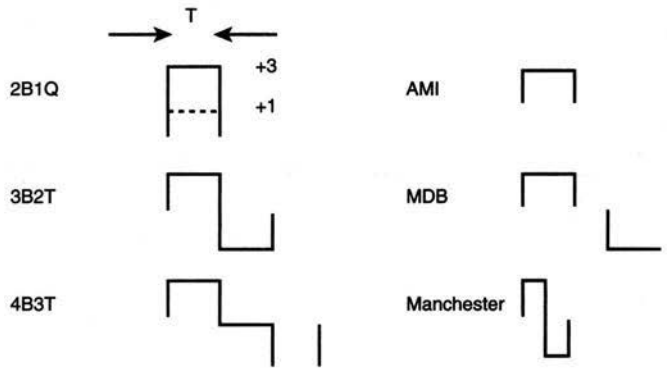
11.1.4 Candidate DSL Line Code Characteristics

There are two general characteristics of line codes that are determinants of their transmission performance: the transmit symbol pulse shape and the logic bit encoding of the symbol. Both symbol pulse shape and logic bit encoding affect the spectrum of a code. Codes considered in the ANSI T1D1.3 working group included the following:

- *2B1Q*.[6] Converts blocks of two consecutive signal bits into a single, four-level pulse. The signaling baud rate is half of the information rate.
- *3B2T*.[7,8] Maps the eight possible combinations of three bits of information into the nine possible combinations of two ternary symbols. This code has lower baud compression than 2B1Q.
- *4B3T*. Maps 16 possible combinations of four bits of information into the 27 possible combinations of three ternary symbols. This block code has less baud rate reduction than 3B2T.
- *AMI (Alternate Mark Inversion)*. Reverses the polarity of a one or mark from that of the previous mark. The baud rate of AMI equals the information rate.
- *MDB (Modified Duobinary)*. Baud rate is the same as the bit rate; however, every baud symbol of the MDB code is repeated with the opposite sign after a baud interval.
- *Manchester*. Also referred to as *Biphase code*, the Manchester code consists of a positive pulse and a negative pulse within a baud. The bandwidth of the Manchester code is twice that of AMI.

Figure 11.5 shows the pulse shapes of these different line codes.

Figure 11.5 Base Pulse Shape of Different Line Codes



MMS43,[9,10,11,12] which was given the most attention at T1D1.3, is also a 4B3T code with three code books. One of the books contains code words that are biased toward positive polarity, one contains words that are negatively biased, and a third is neutral. In addition, the algebraic sum of recently transmitted symbols determines which code book is used.

Note
A code book defines a mapping between bit patterns of input and output for an encoder.

Interleaving is another method of reducing bandwidth. Interleaving involves splitting the transmission sequence into separate, independently encoded sequences and then adding them. The DI43 code, also considered by T1E1.3, is an interleaved code.[13] The U88 code,[14] invented by J. H. W. Unger, is an 8B8T code designed to give spectra properties that would provide good crosstalk and intersymbol interference performance on subscriber loops.

Note
Name interleaving is also used in conjunction with error correction coding for ADSL discussed in Chapter 13, "ADSL (Asymmetrical Digital Subscriber Lines)."

11.1.5 Candidate DSL Line Code Transmission Performance

The power spectrum of a line code is shaped by both the pulse shape and the coding of the amplitude sequence.[15] In particular, we have the line code power spectral density

$$S_s(f) = \frac{1}{T} S_d(f) |P(f)|^2$$

$S_d(f)$ is the power spectral density of the amplitude level sequence.

$|P(f)|^2$ is the power spectral density of the signaling pulse.

For the rectangular pulse, we have

$$|P(f)|^2 = T^2 \left[\frac{\sin \pi f T}{\pi f T} \right]^2$$

For the 2B1Q line code, we have

$$S_d(f) = \frac{5}{9}$$

$$|P(f)|^2 = T^2\left[\frac{\sin \pi fT}{\pi fT}\right]^2$$

$$S_{2B1Q}(f) = \frac{5}{9}T'\left[\frac{\sin \pi fT'}{\pi fT'}\right]^2$$

T' is the 2B1Q baud rate, which is half of the bit rate.

For the 3B2T line code, we have the following:

$$S_d(f) = 1$$

$$|P(f)|^2 = T^2\left[\frac{\sin 2\pi fT}{2\pi fT}\right]^2 + T^2\left[\frac{\sin \pi fT}{\pi fT}\right]^2 \sin^2(\pi fT) + \frac{1}{2}T^2\left[\frac{\sin \pi fT}{\pi fT}\right]^2$$

$$S_{3B2T} = T\left[\frac{\sin 2\pi fT}{2\pi fT}\right]^2 + T\left[\frac{\sin \pi fT}{\pi fT}\right]^2 \sin^2(\pi fT) + \frac{1}{2}T\left[\frac{\sin \pi fT}{\pi fT}\right]^2$$

T is the 3B2T baud rate, which is 2/3 of the bit rate.

For the 4B3T line code, the power spectrum density can be calculated depending on the selection of a particular code book. Because of the 16 information bits to 27 different ternary codes ratio, there are multiple choices of code books.

For the AMI line code, we have

$$S_d(f) = \sin^2(\pi fT)$$

$$|P(f)|^2 = T^2\left[\frac{\sin 2\pi fT}{2\pi fT}\right]^2$$

$$S_{AMI}(f) = T\sin^2(\pi fT)\left[\frac{\sin \pi fT}{\pi fT}\right]^2$$

For the Manchester or Biphase code, we have

$$S_m(f) = T\frac{\sin^4(\pi fT/2)}{(\pi fT/2)}$$

For the MDB code, we have

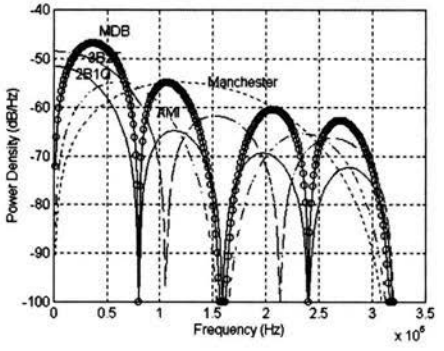
$$S_d(f) = 1$$

$$|P(f)|^2 = 4T^2\sin^2(2\pi fT)\left[\frac{\sin \pi fT}{\pi fT}\right]^2$$

$$S_{MDB}(f) = 4T\sin^2(2\pi fT)\left[\frac{\sin \pi fT}{\pi fT}\right]^2$$

Figure 11.6 illustrates the power spectrum density for the different line codes.

Figure 11.6 Power Spectrum Density of Different Line Codes



Performances of different line codes for the application of DSL were compared,[16,17,18,19,20,21,22] and it was found that the 2B1Q line code was the most favorable

for its higher performance margin. The performance margin is defined as the difference between the signal-to-noise ratio of a particular line code under the selected loop and noise condition and that desired for the defined error rate. The choice of line code was also considered for system cost[23,24,25,26,27,28,29] and VLSI implementation issues.[30,31,32,33,34,35] Table 11.1 illustrates the relative performance parameters of the different DSL line codes.

Table 11.1 DSL Line Codes and Relative Performance Parameters

Code	Input	Output	Bit/Symbol	Redundancy	Margin
2B1Q	2 bits	1 quaternary	2	0	12 dB
3B2T	3 bits	2 ternary	3/2	1/9	9–10 dB
4B3T	4 bits	3 ternary	4/3	11/27	9–10 dB
AMI	1 bit	1 ternary	1	1/3	6–8 dB
MDB	1 bit	1 ternary	1	1/3	6–8 dB
Manchester	1 bit	1 bit	1	0	2–4 dB

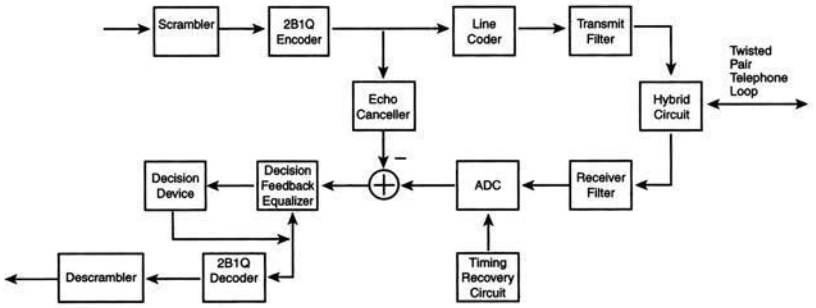
11.2 DSL Transceiver Structure and Line Code

Figure 11.7 shows the general architecture of a DSL transceiver. The data is first scrambled to prevent long streams of zeros or ones. Every two bits of transmit signal are encoded into a symbol through the 2B1Q encoder. The symbol or baud rate is therefore one-half of the bit rate. The symbol is sent to the line encoder, which can be a Digital to Analog Converter (DAC) with or without a digital pulse shaping function.

A second-order Butterworth lowpass filter is used for the transmit filter and receiver filter. The transmit filter reduces the out-of-band energy. The receiver filter minimizes the out-of-band noise. Before using the Analog to Digital Converter (ADC) for analog to digital conversion, a preamplifier is necessary to bring the received signal to the proper voltage level. The ADC is synchronized to the timing recovery circuits.

These same 2B1Q data symbols also are sent to the echo canceller—which resembles the equivalence of the echo path, including transmit filter, hybrid circuits, receiver filter, and ADC at the ADC sampling rate. The 2B1Q encoder's sampling clock is also synchronized to the ADC sampling clock to maintain a stable echo path transfer function. The digitized and echo-canceled received signal is further filtered by the decision feedback channel equalizer before being sent to the *decision device.*

Figure 11.7 A General DSL Transceiver Structure



The decision device is a four-level threshold detector corresponding to the 2B1Q line code. Received symbols generated from the decision device are converted back to information bits through the 2B1Q decoder. The binary bit information is descrambled to obtain the original data stream.

11.2.1 DSL 2B1Q Line Code Parameters

Table 11.2 shows the general transceiver parameters.

Table 11.2 DSL Transceiver Parameters

Line code	2B1Q
Bandwidth (3 dB)	40 kHz
Baud rate	80 kHz
Line rate	160 kbps
Transmit power	12.99 dBm (135 Ohms)

The receiver sampling rate as well as the baud rate is 80 kHz, resulting in a 3 dB transmission bandwidth of 40 kHz. For the 2B1Q line code, each baud carries two bits of information. Therefore, the transmission throughput is 160 kbps. Assuming a power supply of 5 volts and a utilization efficiency of 88 percent, the maximum transmit output voltage is 2.2 volts. For a line impedance of 135 ohms, the total transmit power delivered on the telephone loop is expressed as follows:

$$P = \frac{1}{2}\left(2.2^2 + \left(\frac{2.2}{3}\right)^2\right)\frac{1}{135} = \frac{4.84}{2\times 135}\left(1+\frac{1}{9}\right) = \frac{4.84\times 10}{2\times 135\times 9} = 0.01992 Watts = 12.99 dBm$$

The DSL transmit power density is expressed as

$$PSD = \frac{0.01992}{40 \times 10^3} = 4.98 \times 10^{-7} Watts/Hz = -33.0 dBm/Hz$$

11.2.2 DSL Equalizer Requirements

The length of the decision feedback equalizer should be long enough to cover the duration of the equivalent channel impulse response. The equivalent channel response is the combination of the transmit filter, the transmit transformer, the twisted-pair loop channel, the receiver transformer, and the receiver filter. Figure 11.8 shows the impulse response of the 15 kft, 26-gauge loop.

Figure 11.8 Impulse Response of the 15 kft, 26-Gauge Loop

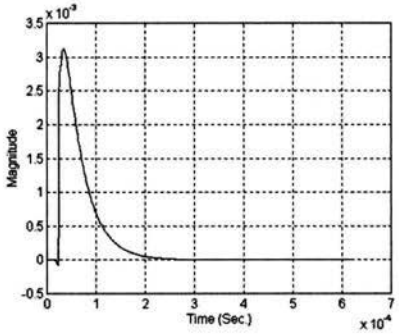


Figure 11.9 shows the impulse response of a second-order Butterworth lowpass filter that was recommended for the transmit/receiver filter.

Figure 11.9 Impulse Response of the Second-Order Butterworth Lowpass Filter

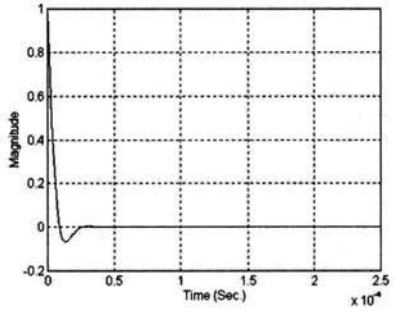
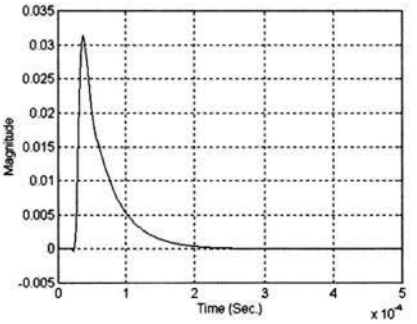


Figure 11.10 shows the impulse response of the equivalent channel that includes two second-order lowpass filters and the 15 kft, 26-gauge twisted-pair loop in cascade assuming the transmission characteristics of transformers are almost ideal.

Figure 11.10 Impulse Response of the Equivalent Channel



The equivalent channel impulse response approaches zero at about 300 microseconds. Considering 30 microseconds of peak delay, the feedback filter should last about 270 microseconds. For a baud sampling rate of 80 kHz, the time between each sample is 12.5 microseconds. Therefore we need about 22 (the round up of 270/12.5=21.6) feedback filter taps. The feedforward filter should have sufficient length to cover the equivalent channel impulse response up to the peak. The required number of feedforward filter taps is therefore three (the round up of 30/12.5=2.4).

11.2.3 DSL Echo Canceller Requirements

The length of the echo canceller should be long enough to cover the duration of the equivalent echo path impulse response. The equivalent echo path response is the combination of the transmit filter, the echo path, and the receiver filter. Assuming for an ideal transformer, the echo path transfer function for the hybrid circuit of Figure 11.11 can be described as:

$$H_{ec}(s) = \frac{RZ_i(s) - RR_b}{(Z(s) + R)(R_b + R)}$$

$Z_i(s)$ is the input impedance of the subscriber loop.

Figure 11.11 A Simplified Echo Path Circuit

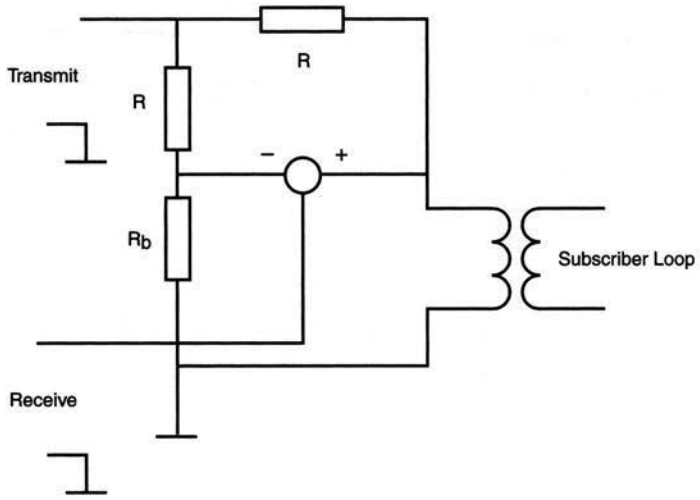


Figure 11.12 shows the impulse response of the echo path in conjunction with a 15 kft, 26-gauge loop.

Figure 11.12 The Impulse Response of an Echo Path

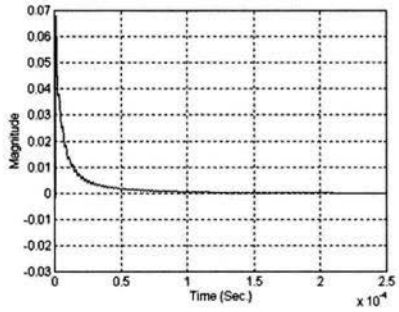


Figure 11.13 shows the impulse response of the equivalent echo path, which includes two second-order lowpass filters and the echo path, in conjunction with a 15 kft, 26-gauge twisted-pair loop in cascade, assuming the transmission characteristics of transformers are almost ideal.

The echo path impulse response approaches zero at about 150 microseconds. Therefore, we need about 150/12.5=12 filter taps for the echo canceller.

The performance of DSL is limited by self-NEXT for long loops. Figure 11.14 depicts the signal-to-NEXT noise ratio from about 65 dB at near DC to about 0 dB at about 135 kHz. The 49 disturber NEXT model is used.

Figure 11.13 The Impulse Response of the Equivalent Echo Path

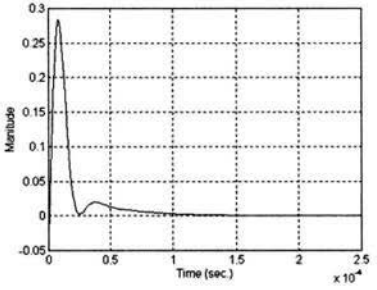
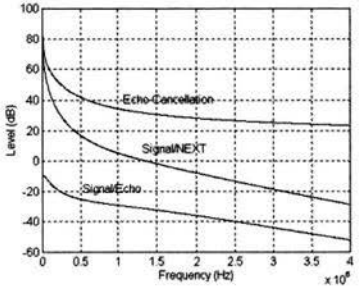


Figure 11.14 Required Echo Cancellation Level



On the other hand, we also see from Figure 11.14 that the signal-to-echo ratio is negative at all frequencies because the echo path return loss is always smaller than the channel loss. Figure 11.14 also shows the required echo cancellation level, which is the difference between the signal-to-noise ratio and the signal-to-echo ratio. The required echo cancellation level, which ranges from 80 dB at near DC to 30 dB at about 135 kHz, is expressed as follows:

$$K_{ec}(f) = \frac{\dfrac{|H(f)|^2}{NEXT(f)}}{\dfrac{|H(f)|^2}{|H_{ec}(f)|^2}} = \frac{|H_{ec}(f)|^2}{NEXT(f)}$$

$H(f)$ is the channel transfer function.
$H_{ec}(f)$ is the echo path transfer function.
$NEXT(f)$ is the NEXT power coupling function.

The required average echo cancellation level is

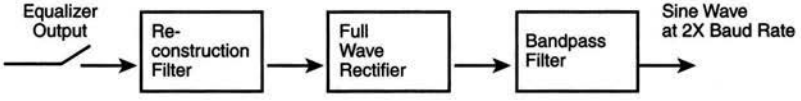
$$\overline{K}_{ec} = \frac{1}{f_{\max}} \int_0^{f_{\max}} K_{ec}(f)df = 68dB$$

in which $f_{\max} = 135kHz$. Although the main energy of DSL is concentrated at below 40 kHz, the signal spectrum of DSL extends above 100 kHz due to its side lopes and a low order transmit filter. The required echo cancellation level for an echo canceller implementation can be relaxed a little because the larger than 60 dB signal to residual echo ratio at low frequencies might not be necessary due to hardware quantization noises.

11.2.4 DSL Timing Recovery Circuits

The timing recovery circuit of a DSL transceiver can be implemented with the spectral line method.[36] Figure 11.15 shows the block diagram of the timing recovery circuit using the spectral line method.

Figure 11.15 A Spectral Line Timing Recovery Circuit



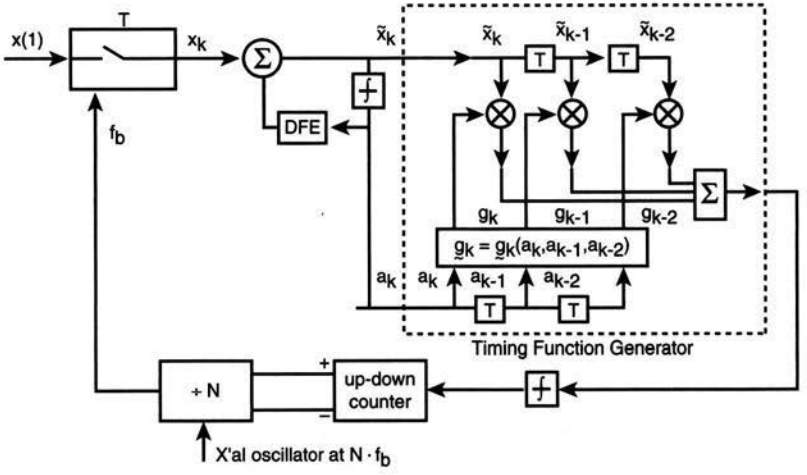
The input of the timing recovery circuit is connected to the output of the decision feedback channel equalizer. The timing recovery circuit consists of the following components:

- *Reconstruction filter.* Generates an analog version of the equalized and echo-canceled signal. The frequency characteristics of the reconstruction filter can be optimized to minimize the timing jitter.
- *Full wave rectifier.* Generates harmonics of the received signal. A square operation can sometimes be used instead of the rectifying operation for the same purpose of harmonics generation.
- *Bandpass filter.* Selects the desired sampling frequency. The bandwidth of the bandpass filter is usually relatively narrow, and the Q relatively high.

The timing recovery circuit can be implemented with digital circuits with a sampling rate that is at least three to four times the baud rate.

The timing recovery circuit of a DSL transceiver can also be implemented with the baud rate sampling techniques.[37] Figure 11.16 illustrates a block diagram of the timing recovery circuit using the baud rate sampling method in conjunction with a decision feedback channel equalizer.

Figure 11.16 A Baud Rate Sampling Timing Recovery Circuit



The timing function generator functional block uses both the equalized and echo-canceled signal and the estimated data symbol to generate the timing information to drive an up-down counter. The function g_k can be chosen as simple as

$$g_k = a_k x_{k-1} - a_{k-1} x_k$$

The output of the timing function generator is used to control the up-down counter. We obtain the nominal sampling clock rate by dividing a master clock by a nominal N. The up count action will increase the value of N and subsequently reduce the sampling rate. A down count action will decrease the value of N and subsequently increase the sampling rate. The timing recovery circuit can be implemented with digital circuits operating at the baud rate.

11.3 DSL Semiconductor Technologies

The capability of semiconductor technology has been a primary consideration during the development of DSL. The economic feasibility of DSL was studied based on the cost structure of VLSI technologies. The performance and integration requirements in turn pushed the state of the art of related semiconductor technologies. The adaptive echo cancellation for DSL was quite a challenge for semiconductor digital signal processing circuits operating at a sampling rate of 160 kHz compared with that of voice band modem at only 8 kHz at that time. At such a high sampling rate, the implementation of ADC and DAC also demanded some special analog circuit techniques.

Overall, the partitioning of different functions among digital and analog semiconductor circuits was itself an interesting task. To reduce system cost, the amount of discrete components in addition to the semiconductor chips was minimized. A single-chip solution that included both analog and digital circuits also was part of the cost minimization objective.

11.3.1 A COM DSL Analog Front-End Chip from Siemens

The analog front-end part of a DSL chip set was implemented using the 2 μm double-metal CMOS technology. It consists of about 58,000 transistors for a chip size of 8.2 by 7.3 mm². The power consumption of this analog front-end chip is about 125 mW.[38] Figure 11.17 shows the functional block diagram of this analog front-end chip, which has four major functional blocks: pulse shaper, line driver, wake up signal detector, and A/D converter.

Figure 11.17 Block Diagram of a Transceiver

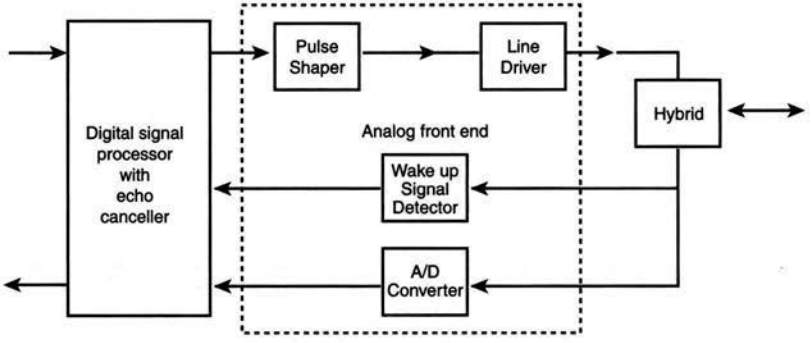
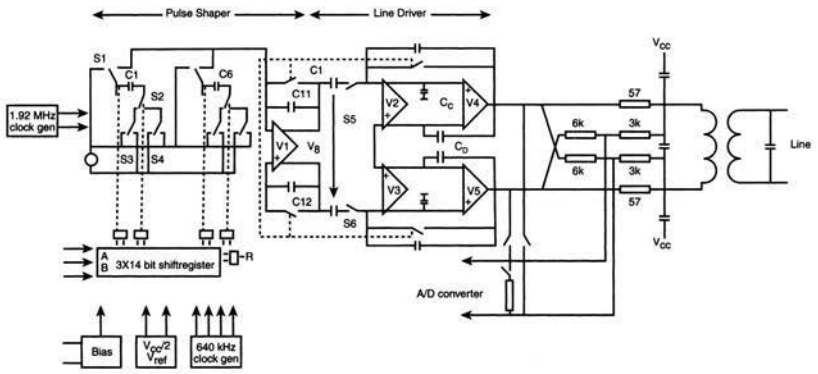


Figure 11.18 shows the functional circuit diagram of the transmit direction for a 2B1Q-coded signal with levels of +3, +1, −1, and −3. At the pulse shaper, positive or negative pulses are formed using charge redistribution between an array of six switched capacitors C_1 to C_6 and the capacitor C_{i1} in the feedback loop of an operational amplifier V_1. To get a fully

differential approach, a second set of six switched capacitors, not shown in Figure 11.18, and the capacitor C_{i2} are implemented at the positive input of V_1. A reset pulse R is used to close switches across C_{i1} and C_{i2}, thereby setting a voltage V_s=0 at the differential outputs of V_1 at an appropriate time.

Figure 11.18 Transmit Part



The line driver is built up by two stages of push-pull power amplifiers: V_2, V_3, V_4, and V_5. The gain of the power amplifiers is set to 1.2, corresponding to a capacitor feedback with C_A=3 pF and C_B=2.5 pF. By using a capacitor feedback, there are no common-mode signals at the input of the line drivers. The hybrid function of the interface is performed with six external resistors if the switches S are nonconducting. With zero output impedance of the line driver and a 150 Ω load, they act as a bridge circuit with high echo return loss to the input of the A/D converter. The amplitude of a single +3 pulse at the load is 2 volts.

The capacitors in the external coupling circuit are part of a low pass filter. The filter is used to smooth the staircase of the transmit pulses. The capacitors connected to *Vcc* avoid oscillation of the line driver at high frequencies. For testing the transceiver, the transmit pulse can be looped back in the receive direction. To do this, the switches S, realized as low-impedance MOS transistors, are put in a conducting state. The resistor in series with one of the switches together with other switches build a voltage divider to prevent overloading of the A/D converter in the loopback mode.

Figure 11.19 shows a simplified schematic for a second-order sigma-delta modulator.[39] Two integrators in the forward path, two 1-bit DACs in the feedback path, and a 1-bit ADC (a comparator) are used. An over-sampling frequency of 15.36 MHz and a second-order noise shaping filter guarantee an adequate accuracy, which is independent of capacitor or resistor ratio. The quantization noise is less than –65 dB at the output of a following digital filter with 120 kHz bandwidth, which is implemented in the digital part of the transceiver.

Figure 11.19 A Second-Order Sigma-Delta Modulator

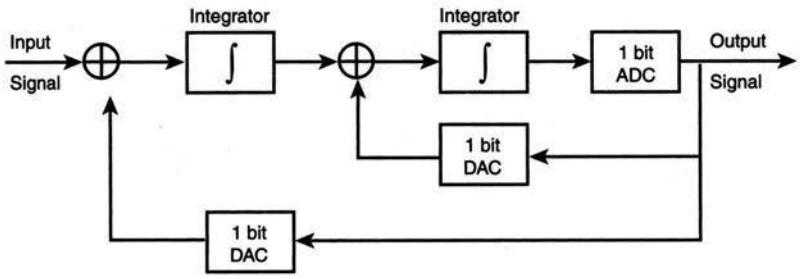
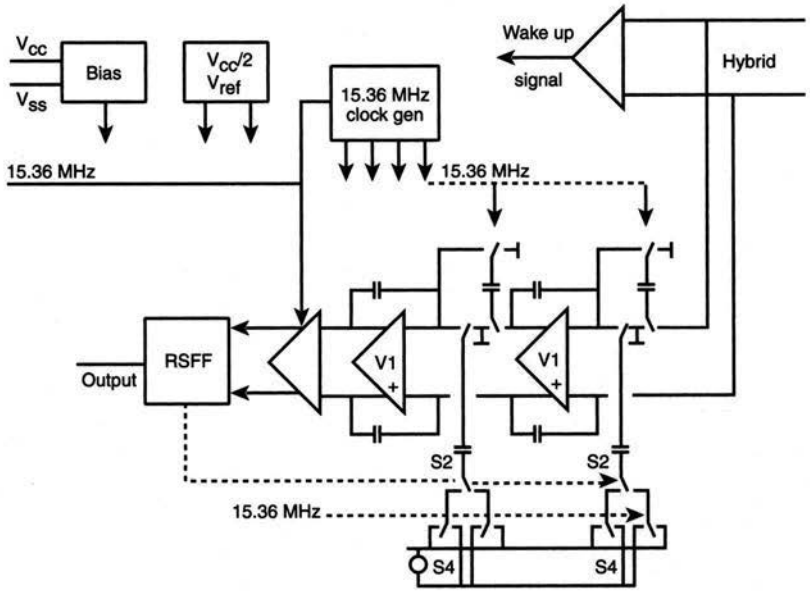


Figure 11.20 shows a functional circuit schematic of the realized sigma-delta modulator. Switched-capacitor integrators in a fully differential configuration are used in the forward path. The operational amplifiers are folded cascade amplifiers with dynamic common-mode feedback. To ensure proper settling in the switched-capacitor integrators working with a clock rate of 15 mHz requires operational amplifiers with bandwidth above 100 mHz. The 1-bit ADC following these two integrators consists of a clocked comparator and a subsequently added RS flip-flop. Using a four-phase clock in the switched-capacitor circuitry minimizes nonlinearities caused by feedthrough.[40]

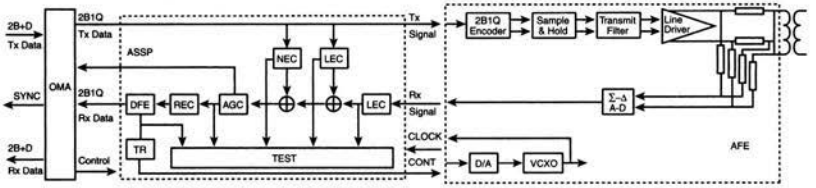
Figure 11.20 Receiver Part



11.3.2 A DSL Analog and Digital Chip Set from AT&T

Another DSL chip set, the AT&T ISDN Chip Set (see Figure 11.21), has a digital *Algorithm Specific Signal* (ASSP) part implemented using the 1.75 µm double-poly CMOS technology and an analog front-end (AFE) part implemented using the 1.25 µm double-metal CMOS technology with chip sizes of 14.5 mm² and 47 mm², respectively.[41] The ASSP consists of *Decision Feedback Equalizer* (DFE), *RECeiver filter* (REC), *Timing Recovery* (TR), *Automatic Gain Control* (AGC), *Nonlinear Echo Canceller* (NEC), *Linear Echo Canceller* (LEC), *DECimation* (DEC), and *TEST functional blocks.* The AFE consists of 2B1Q encoder, sample and hold, transmitter filter, line driver, sigma-delta A to D converter, and D to A converter in conjunction with the *Voltage Controlled Crystal Oscillator* (VCXO).

Figure 11.21 The Functional Block Diagram of the AT&T ISDN Chip Set

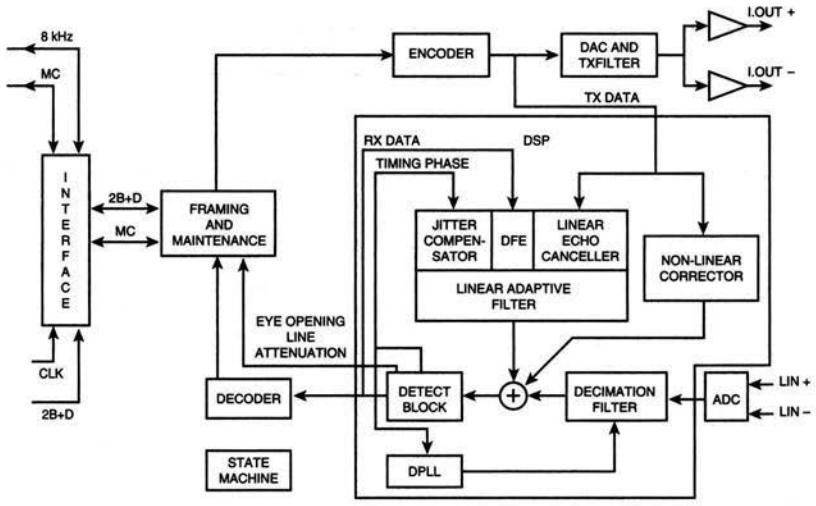


11.3.3 A Single-Chip DSL Solution from Mitel Semiconductor

A single-chip DSL solution was also implemented using a 2 µm single-metal CMOS technology.[42] The single-chip solution DSL chip consists of about 60,000 transistors with chip size of 65.5 mm². Figure 11.22 shows the functional block diagram of this *Digital Subscriber Line Integrated Circuit* (DSLIC). At the encoder, the data stream is scrambled, a 12-bit CRC is calculated, and a nine-symbol synchronization word is inserted before the stream is converted into a two-bit parallel word at 80 kbaud/s. A four-level encoding sampled-analog finite impulse response (FIR) transmit filter, smoothing filter, and dual line drivers are used to differentially drive the line through a transformer.

On the receiver side, analog-to-digital conversion is performed by an over-sampled, second-order delta-sigma converter running at 10.24 mHz master clock rate. All echo cancellation and equalization is then performed in the digital domain using dedicated baud-rate sampling DSP hardware. After frame and superframe recovery, decoding, and descrambling, the data stream is recovered. An on-chip PLA state machine implements a training sequence generator to control the adaptive structures and perform the ANSI network-termination (NT)/line-termination (LT) activation/deactivation procedures. Clock recovery is accomplished by digital phase-locked loops within the DSP and interface, driven by timing decisions derived from the received 2B1Q line signal.

Figure 11.22 DSLIC Block Diagram



The *Linear Adaptive Filter* (LAF) implements the transceiver's *Linear Echo Canceller* (LEC) and DFE. For both functions, the processing in each line symbol interval consists of

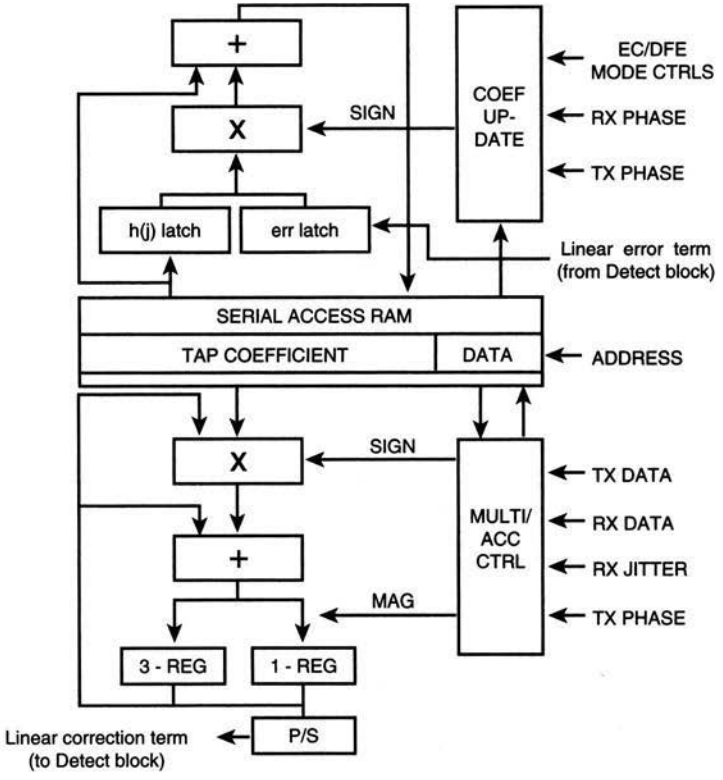
- A convolution of recent quaternary data values with the tap weights of a FIR fitter, generating a linear correction to be applied to the received signal
- An update of tap weights using the clipped LMS algorithm

The LAF is implemented in three main sections:

- A block of three-transistor DRAM for storage of 58 LEC, JC (jitter compensator), DFE, taps, and 58 corresponding transmit and receive data values
- A data path far convolution
- A data path for update

Each data path has an associated controller implemented in random logic. The RAM is addressed cyclically with each tap weight, and its corresponding data value, being both read and written once per line symbol interval. Figure 11.23 shows the functional block diagram of the Linear Adaptive Filter.

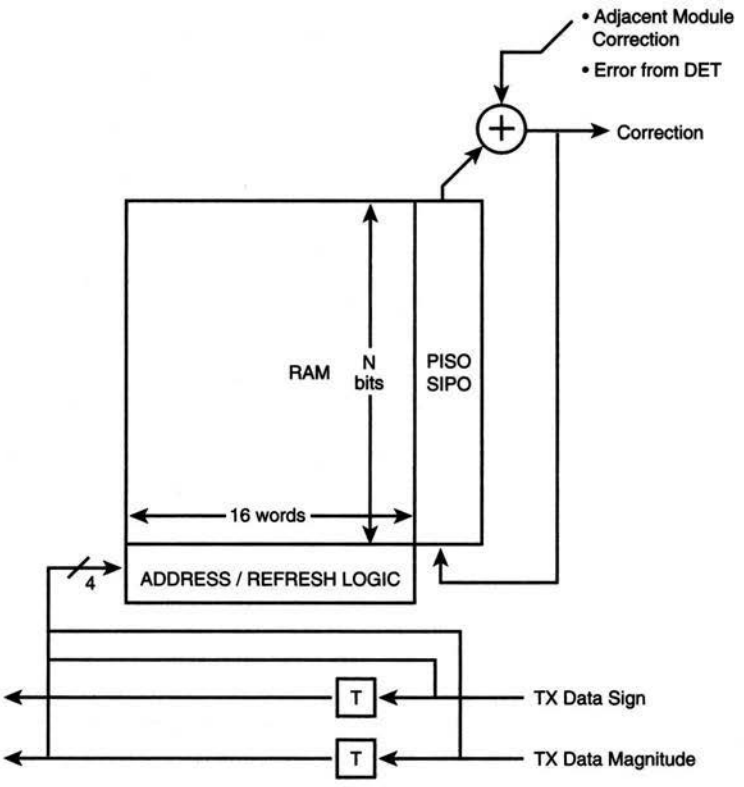
Figure 11.23 Linear Adaptive Filter Block Diagram



The *Nonlinear Corrector* (NLC) implements a modular RAM-based table look-up adaptive filter. Each NLC module maintains 16 values in a look-up table. Each value is the correction to be applied when one of 16 possible pairs of successive transmit data values is transmitted. The NLC models nondispersive and dispersive nonlinearity is caused by the interaction between two successive transmit symbols. This structure was chosen because the dominant nonlinearity is in the analog transmit circuitry, where pulses overlap about 60 percent of the time due to two successive transmit symbols.

Each NLC module, as shown in Figure 11.24, consists of a block of three-transistor DRAM of size 16 words×24 bits, address/refresh logic, a PISO/SIPO, and a serial adder.

Figure 11.24 Nonlinear Corrector Module Block Diagram



In each line symbol interval, the RAM is accessed three times in the following sequence:

1. The chosen data value is read, serialized, and added to values from other NLC modules to form an overall nonlinear correction.
2. The same value is read again, serialized, and added to an error value, generating the updated value.
3. The updated value is converted to parallel form and written back to RAM. When not in use for read/write access, the address/refresh logic carries out a cyclical refresh of the RAM.

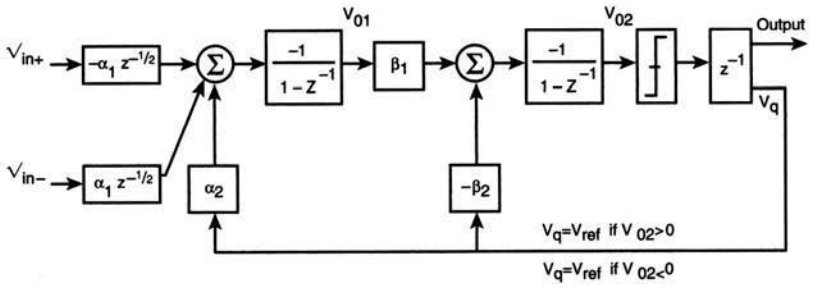
The performance requirements of the ADC can be determined by analyzing the variation, in input level, and required echo cancellation, as one varies the loop length (attenuation) of the transmission system. Due to the strong presence of the near-end component, the input

range varies only by some 10 dB, while the highest SNR is required at the lowest signal level because the far end is so heavily attenuated.

A total SNR of approximately 70 dB was found to be necessary to achieve the desired margin. A second-order delta-sigma modulator with an over-sampling ratio of 128 (master clock to decimated output clock) was chosen due to its capability to meet the requirements, its suitability for MOS integration, and after consideration of the inherent tradeoffs between silicon area and design time.

Figure 11.25 illustrates a block diagram of the ADC. Nondelaying integrators were chosen because they result in lower output levels and reduced slew requirements in the operational amplifiers. Time-domain simulations, with sinusoidal inputs, of this structure predict an SNR of 84 dB for a 40 kHz bandwidth (brickwall filter) and 71 dB for an 80 kHz bandwidth. The actual decimation filter used results in an effective bandwidth between these two extremes.

Figure 11.25 Block Diagram of Second-Order Delta-Sigma Modulator



11.3.4 *A Single-Chip DSL Solution from National Semiconductor*

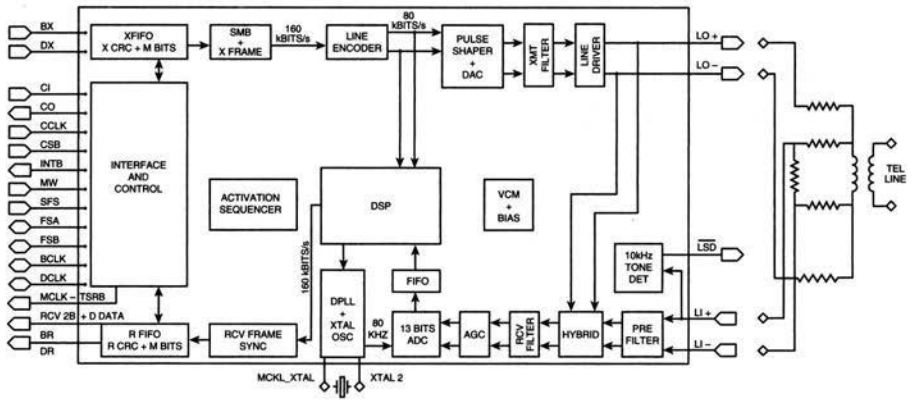
Another single-chip DSL solution was implemented using a 1.5 μm double-metal CMOS technology.[43] This single-chip solution DSL chip has a chip size of 77 mm². Analog signal processing preconditions the signal by maximizing the dynamic range utilization of the 13-bit A/D converter. The digital signal processor performs the high-pass filtering, precursor equalization, linear echo cancellation, far-end signal equalization, and timing recovery functions. The analog signal preconditioning technique enables the entire DSP section to be designed without a single dedicated multiplier.

As shown by Figure 11.26, there are three main functional sections:

- The digital interface on the left of the diagram

- The digital signal processor (DSP) at the center
- The analog front end at the right

Figure 11.26 Transceiver Block Diagram



The digital interface adapts the data rate from 144 to 160 kbps, formats the transmit data, and inserts cyclic redundancy check (CRC) and maintenance bits. The resulting data stream is then scrambled, and 18-b synchronization words are inserted.

The analog front end (AFE) of the DSL transceiver is tailored completely to improving the far-end signal-to-echo ratio, band-limiting this composite signal to achieve higher immunity to noise and to enable baud-rate sampling. As shown by Figure 11.27, the AFE contains the following circuit functions:

- A pulse-density-modulation (PDM) 2B1Q signal generator and line driver
- An adaptive echo precanceller or hybrid balance filter
- An antialiasing filter
- A lowpass receiver filter
- An adaptive gain control (AGC) circuit
- An Analog to Digital Converter (ADC)

The *Digital Signal Processor* (DSP) completes the echo cancellation task and performs far-end signal equalization and timing recovery utilizing a sampled signal strongly preconditioned by the AFE. Figure 11.28 shows the major circuit blocks in the DSP section, which include a highpass filter, a precursor equalizer, an adaptive echo canceller, an adaptive decision feedback equalizer, and the timing recovery function.

Figure 11.27 Analog Front-End Block Diagram

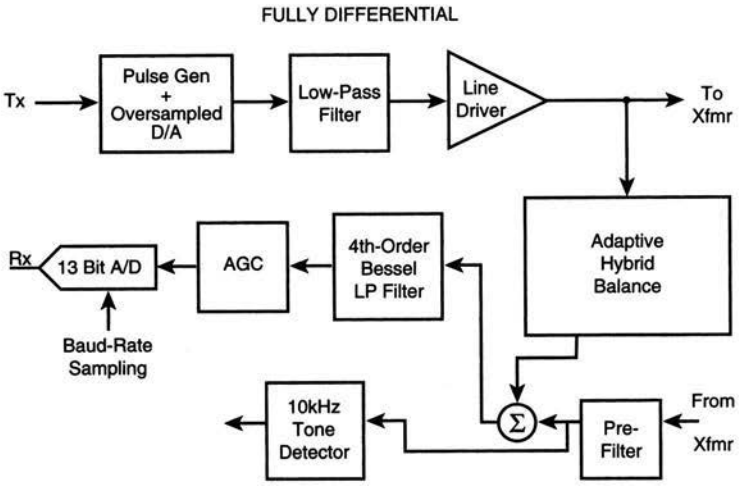
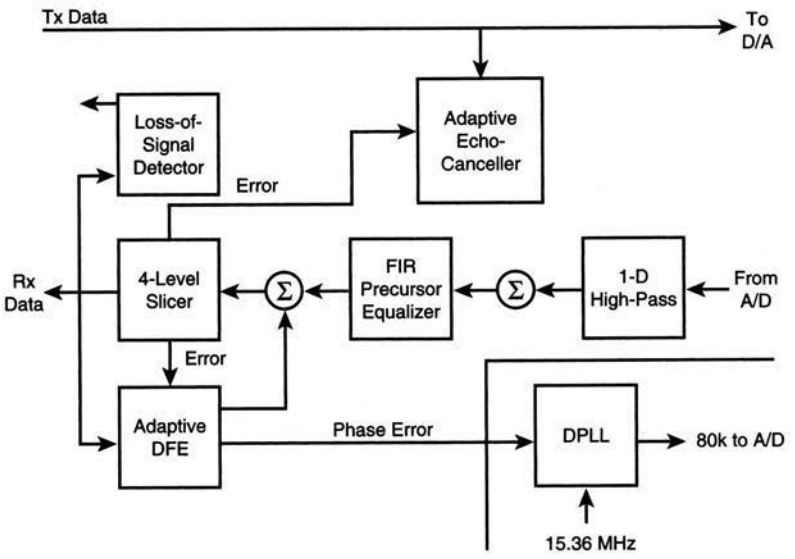
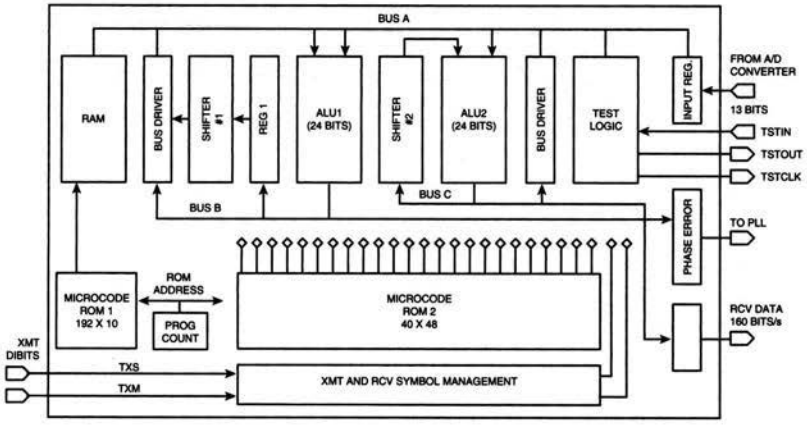


Figure 11.28 DSP Functional Block Diagram



As shown in Figure 11.29, the DSP computation tasks were divided between two identical arithmetical and logical units—ALU1 and ALU2—each of which operated at 7.68 mHz.

Figure 11.29 DSP Architecture



The instruction set executed by each of the two ALUs was limited to addition, subtraction, pass-through (data transfer), shift one location left (X2), and shift two locations left (X4). Rather than dedicating one ALU to each of the filtering tasks, it was more economical to share ALU2 between both the EC and DFE convolutions, while ALU1 performs the EC and DFE coefficient updating. This architecture minimizes busing by enabling the two ALUs to share the same input bus *A*. A shorter bus *B* returns the updated coefficients to the RAM, and transfers the phase error information to the main phaselocked loop.

The 80-word RAM stores the 34 EC and the 34 DFE coefficients, as well as several intermediary variables. During each cycle of the 7.68 mHz clock, one coefficient is read from the RAM, while the previous updated coefficient is returned to the RAM. The RAM address is provided by ROM1. Another microcode ROM, ROM2, generates the commands for the two ALUs and for the data-path registers and shifters.

11.4 DSL Performance Objectives

The performance of the DSL system is decided by the transmission characteristics of the twisted-pair telephone loop, the receiver front-end noise level, and the structure of the DSL transceiver. Figures 11.30, 11.32, and 11.34 show configurations of 15 selected DSL test loops.[44] The transmission characteristics of these loops are representative for the 10 percent worst nonloaded resistance design loops.

As indicated by their transfer functions, the first five loops have the most transmission losses (see Figure 11.31), the next five loops have fewer losses (see Figure 11.33), and the last five loops have the least losses (see Figure 11.35). These transfer functions are characterized by a 15 dB loss at about 1 kHz, and about a 35 to 45 dB loss at about 50 kHz. Bridged taps cause nulls at different frequencies.

Figure 11.30 Test Loops 1–5

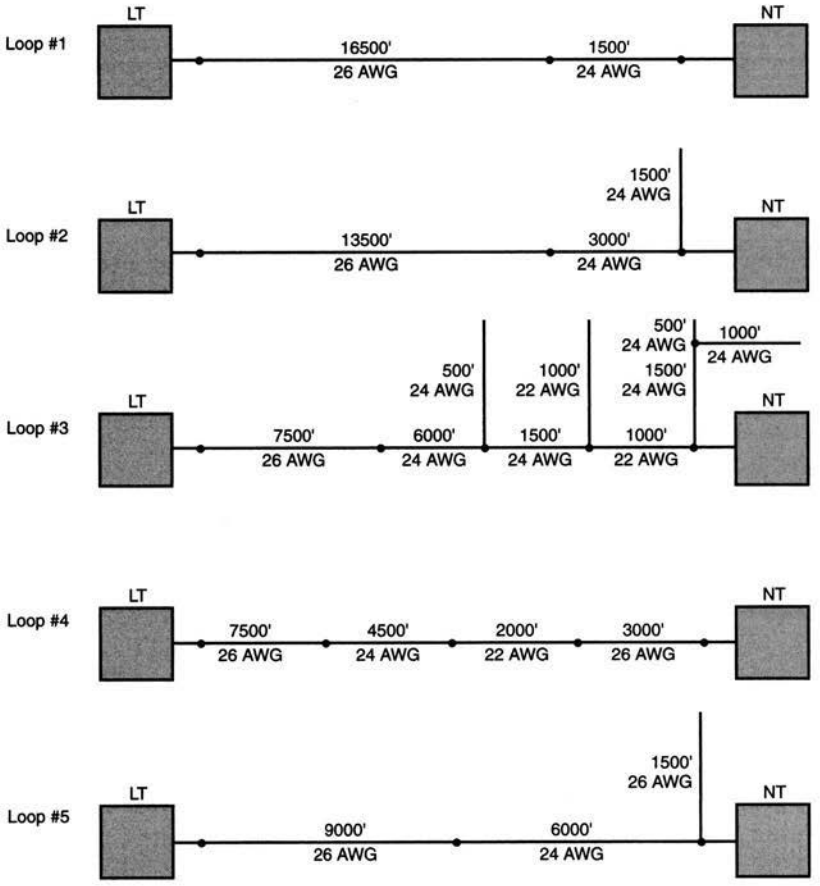
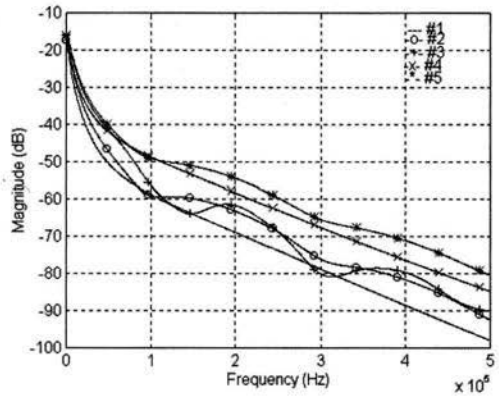


Figure 11.31 Transfer Functions of ISDN Test Loops 1–5



Because of the use of the echo-cancellation-based full-duplex method, the dominant noise component at the receiver front end is the self Near End Crosstalk (NEXT). The self-NEXT is induced by neighboring DSL transmitters. The NEXT noise level depends on the transmit signal level and the NEXT loss. Figure 11.36 shows the DSL NEXT loss model derived from the 1 percent, 49-disturber NEXT loss model.

The performance of a DSL transceiver can be estimated through computer simulation. A white binary sequence is generated and encoded as a random data symbol sequence. The binary sequence is generated at the data transmission rate and 2B1Q encoded into the symbol rate. Meanwhile, a white Gaussian sequence is also generated to drive the NEXT filter. The white Gaussian sequence is generated at the simulation sampling rate. The simulation sampling rate is usually 5 to 10 times that of the baud rate to emulate the channel covering excessive bandwidth signal energy and to show the effect of sampling phase.

The performance of a DSL transceiver is simulated using the virtual DSL system, as shown in Figure 11.37, assuming that the residual echo level is insignificant and, therefore, can be ignored. The received signal is the combination of the data symbol after the equivalent channel and the white Gaussian sequence after the equivalent NEXT coupling function.

The equivalent channel consists of the transmit filter, the channel, and the receiver filter. The equivalent NEXT coupling function consists of the transmit filter, the NEXT filter, and the receiver filter. Both transmit and receiver filters are second-order Butterworth filters with a 3 dB corner frequency of 40 kHz. The channel models are 15 selected DSL test loops. The NEXT filter is an FIR filter designed according to the DSL NEXT loss model.

Figure 11.32 Test Loops 6–10

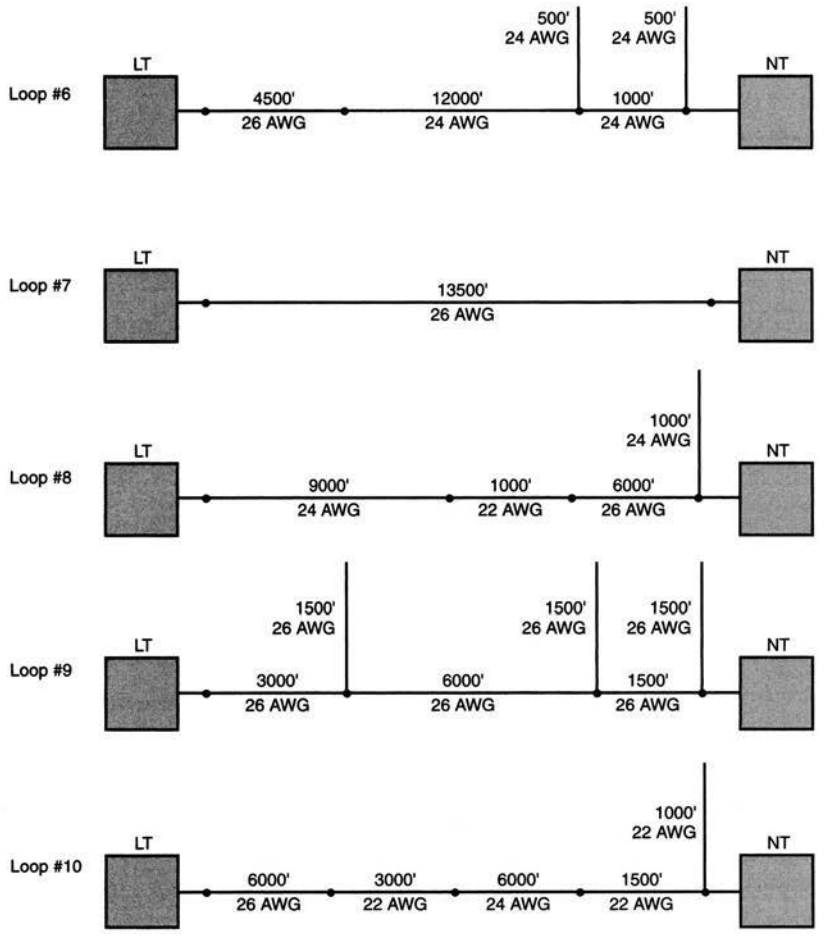


Figure 11.38 shows the impulse response of the NEXT FIR filter, and Figure 11.39 shows the transfer function of the NEXT FIR filter. For the simplicity of computer simulation, the effect of the analog transmit shaping filter is also included in this NEXT filter.

Figure 11.33 Transfer Functions of ISDN Test Loops 6–10

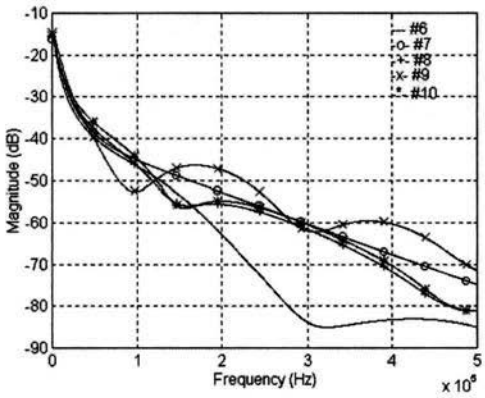


Table 11.3 shows simulated DEF output SNR for the 15 test loops. The required SNR is 21.5 dB for an error rate of 10^{-7}. A 6 dB measured performance margin is required for DSL test loops 5 through 15. The 6 dB of performance margin on test loops 1 through 4 is difficult to reach, even for simulation results in which timing error and some other hardware tolerance have not been accounted for.

The simulation results in Table 11.3 are based on a baud rate DFE with 5 FFF and 32 FBF coefficients. Results might be different for different DFE size and other hardware parameter assumptions.

Table 11.3 Simulated Performance on 15 DSL Test Loops

Test Loop Number	SNR	Margin
1	23.6	2.1
2	20.5	–1.0
3	25.8	4.3
4	23.4	1.9
5	32	10.5
6	32.9	11.4
7	33.2	11.7
8	32.5	11.0
9	30.4	8.9
10	32.0	10.5

Test Loop Number	SNR	Margin
11	34.6	13.1
12	35.5	14.0
13	34.1	12.6
14	40.2	18.7
15	37.5	16.0

Figure 11.34 Test Loops 11–15

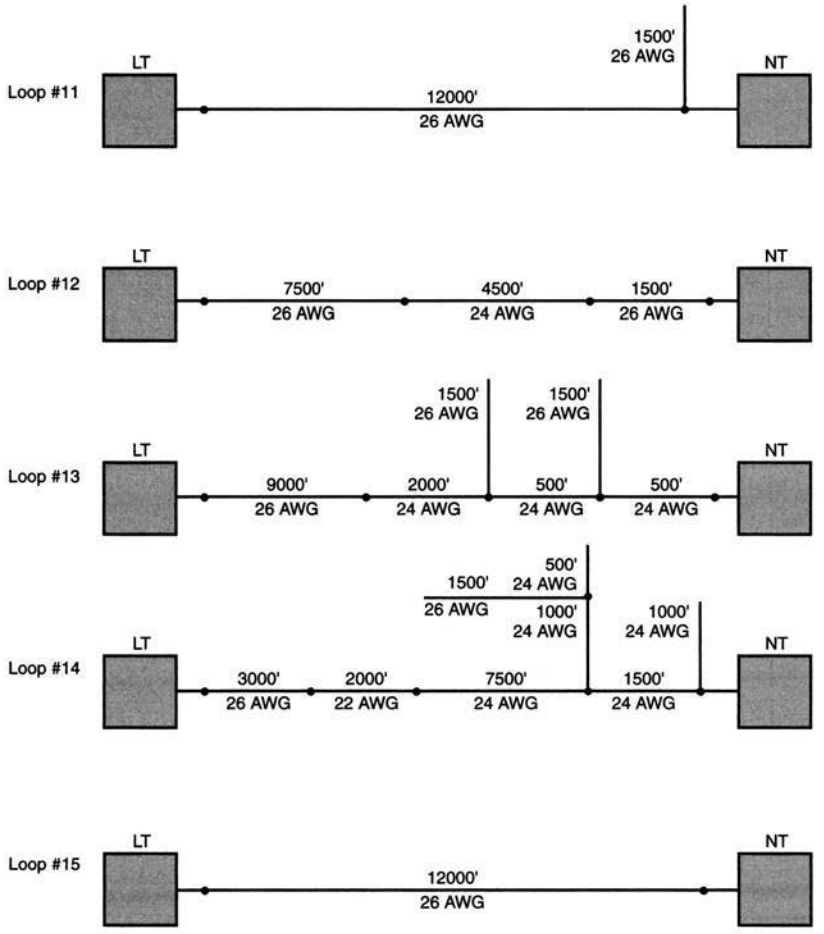


Figure 11.35 Transfer Functions of ISDN Test Loops 11–15

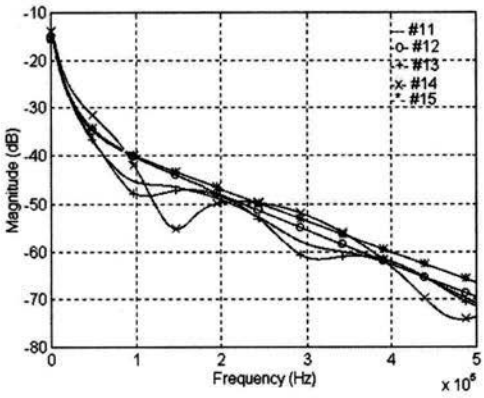


Figure 11.36 NEXT Model

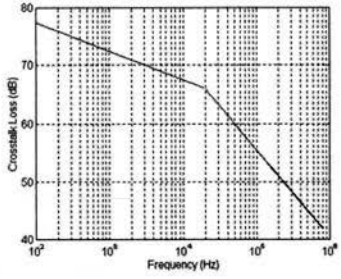


Figure 11.37 The Generation of DSL Received Signal

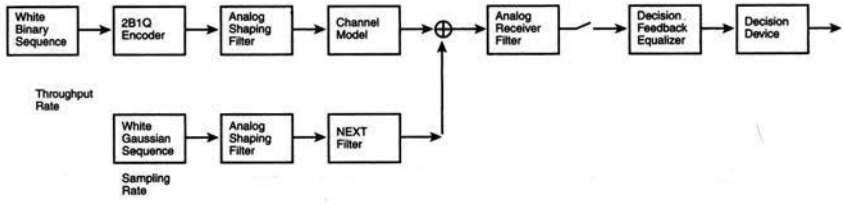


Figure 11.38 Impulse Response of the NEXT Filter

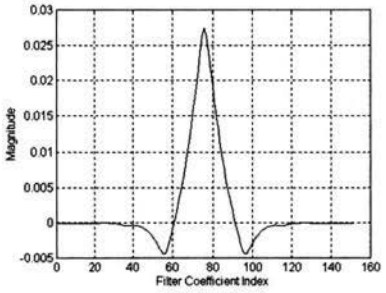
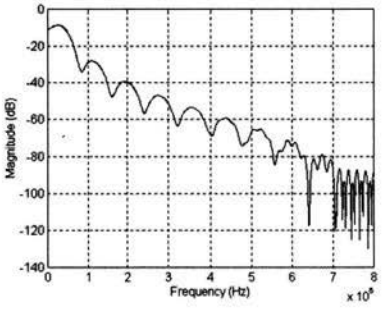


Figure 11.39 Transfer Function of the NEXT Filter



11.5 DSL Network Interfaces

The concept of the Integrated Service Digital Network (ISDN) was gradually formed from aspects of end-to-end digitization of telephony services, data users, integration of services, Intelligent Network Architecture (INA), and the improvement of traffic efficiency.[45]

It was discovered that the digital technology can be used to improve telephone equipment efficiency during these telephone service expansion days. Now, Central Offices and the network connections in between them are all digital. Today, the digitization of POTS happens right at its local connecting Central Office. Major corporations are early data users to interconnect their own computer networks. Small businesses become important data users when their productivity depends on the access of commercial data networks. Individual data users work at home either during or after regular hours to reach their business data networks.

With digitization of all applications, services with different rates and different transmission requirements can all be processed with the same switch and network equipment. Features and capabilities of different services can be all realized at user end equipment with the help of properly defined communication protocols. The concept of INA is based on the coordination of trunk, network, service, and management capabilities. With the concentration of switch and transmission capabilities, the equipment efficiency is increased and the service quality is improved.

The concept of ISDN is to have integrated access—which is defined as a single point of access to multiple networks and to different kinds of networks—for all digitized services. Access interfaces of ISDN are defined as Basic Rate Access with two B channels and one D channel, Primary Rate Access with 23 B channels and one D channel, and Multiple Access. The throughput of Multiple Access can be H0 of 384 kbps or H1 of 1.536 Mbps.

DSL was developed for the Basic Rate Access ISDN interface with two B channels and one D channel. The total transmission throughput for two B channels and one D channel is 144 kbps, while the transmission throughput of DSL is 160 kbps. The difference of 16 kbps is allocated for the overhead channel. The D channel can be used to send switching signals, such as SS7, for the connection of these B channels. Depending on the user terminal type, B channels can be used for either POTS or data applications.

Two B channels can be reverse multiplexed to provide a digital link between end users with a transmission throughput of 128 kbps. The D channel can also be used for packet-oriented applications, via X.25, for example, by itself. The overhead channel connects DSL transceivers at both ends after the initialization of the DSL link. The overhead channel consists of timing and framing information, consuming a throughput of 12 kbps, and networking operation information, consuming a throughput of 4 kbps.

A DSL can be used to connect an ISDN capable CO, a *remote terminal* (RT), which is extended from an ISDN capable CO via a Digital Loop Carrier (DLC), or a serving CO that is extended from an ISDN capable CO via an integrated interoffice digital carrier to an end Basic Rate Access ISDN user, as shown in Figure 11.40.

A DSL is designed to provide Basic Access Rate ISDN connections between an ISDN capable Central Office and subscriber ISDN terminal equipment. Figure 11.41 illustrates the components of the DSL link between the CO and the telephone subscriber.

Figure 11.40 Architecture for ISDN Basic Rate Access

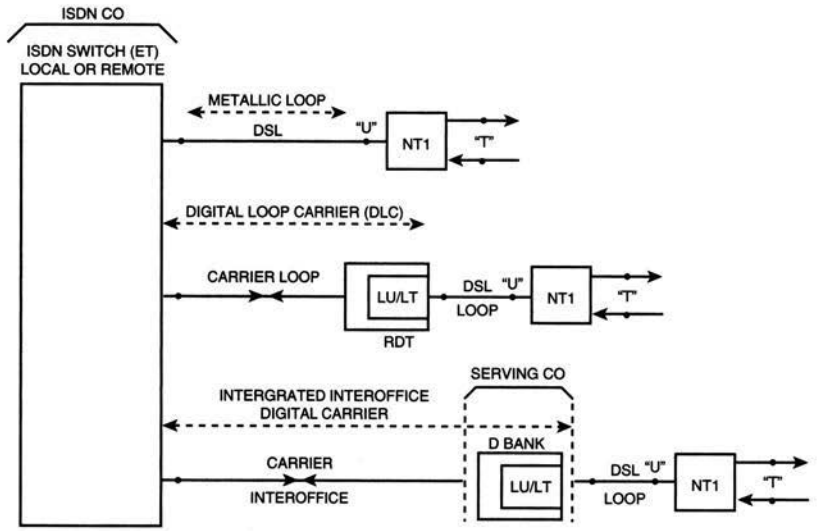
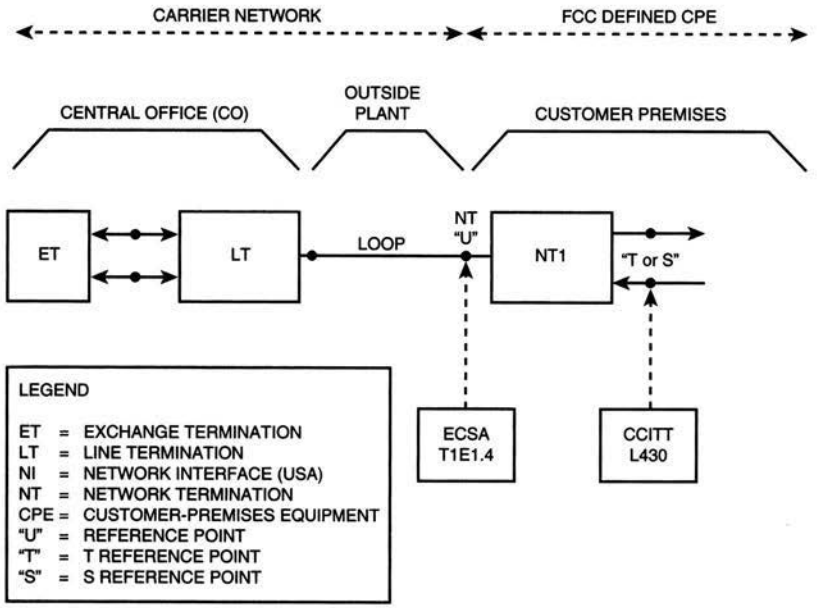


Figure 11.41 Components of a DSL Link



The definition and functions of the DSL link components are as follows:

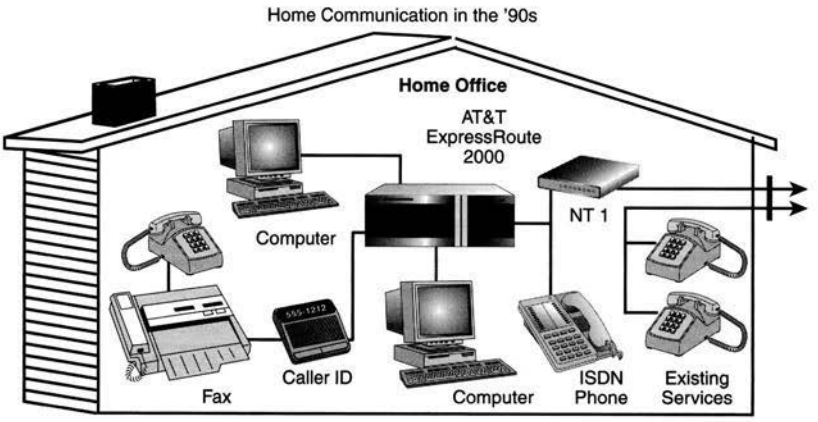
- *Line Termination (LT).* The DSL transceiver at the Central Office side.
- *Network Termination 1 (NT1).* The DSL transceiver at the subscriber side.
- *U interface.* The connection between a DSL transceiver and the twisted-pair loop for both LT and NT1.
- *T interface.* The connection between an NT1 and subscriber terminal equipment (TE), which only specifies network layer 1 multiplexing and maintenance functions.
- *Network Termination 2 (NT2).* Can be placed between an NT1 and a TE.
- *S interface.* The connection between a NT2 and a TE, which specifies network layers 2 and 3 protocol handling, switching, and maintenance functions. The S interface is more suitable for PBX connections.
- *Exchange Termination (ET).* Connects the telephone network to an LT.

The Basic Rate Access provided by a DSL is available to subscriber ISDN terminal equipment through the T or S interfaces. Typical ISDN telephony terminal equipment as originally defined has the capability of 2 POTS lines and a data port of 16 kbps corresponding to these two B channels and one D channel. The T interface can be implemented as a bus with two pairs of twisted pairs, enabling multiple ISDN terminal equipment to be connected to the same T interface. A simpler ISDN telephony terminal could have the capability of only one POTS channel supported by the choice of either of these two B channels plus the access to the D channel for switch signaling.

An ISDN-to-POTS Converter (IPC) can also be used to connect a conventional telephone set, a conventional FAX machine, or a voice band modem to the ISDN line. The IPC also could have a serial port or an Ethernet interface for computer networking. The IPC can have either a U interface or a T interface for the ISDN line, and it should carry out the A/D and D/A conversion and the signaling and power supply functions for the POTS line.

For the application of computer networking, the IPC should also include the ***router function***. With some built-in intelligence, the IPC should also be able to perform some data/voice multiplexing functions. The multiplexing can be implemented within two B channels or within each B channel if packetized applications are available. A variety of applications can be realized with the combination of an NT1 and a multiple function IPC, as illustrated by Figure 11.42.

Figure 11.42 An ISDN-to-POTS Converter



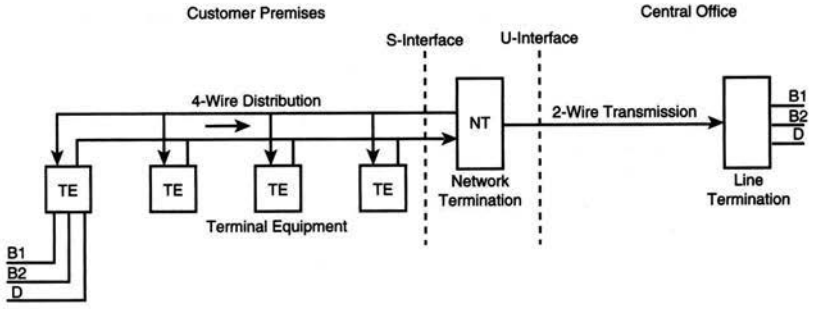
11.5.1 DSL T Interfaces

The T interface is defined[46] based on a two-pair, twisted-pair wiring system. The twisted-pair cable can be either *Category 3* or *Category 5*. Category 3 and Category 5 twisted-pair cables are designed for data transmission applications. Transmission characteristics of Category 5 cable are better than those of Category 3 cable. One twisted-pair is for the transmission from the NT1 to terminal equipment (TE) and the other pair is for the transmission from TE to the NT1, as illustrated in Figure 11.43.

In the case of only one TE, a point-to-point configuration is realized. Two twisted pairs connect the NT1 and the TE. In the case of multiple TE installations, a point-to-multipoint configuration is realized. For this late case, two twisted pairs start from the NT1 and end at a location of physical convenience. A TE at the other end of twisted pairs is not necessary; however, termination resistors at the other end of twisted pairs are always required. TE can be connected to twisted pairs anywhere between the NT1 and the end of the twisted pairs. Twisted pairs are connected to the NT1 and TE via the RJ45 plugs.

Binary rectangle pulse of zero-to-peak amplitude 750 mV is used for the transmission on these twisted pairs. The nominal pulse width is 5.21 μs, resulting to a signaling rate of about 192 kHz. The additional transmission overhead is necessary for the implementation of the T interface frame, which enables the access of multiple TE installations at different locations. The nominal T interface terminal impedance is 100 ohms.

Figure 11.43 Reference Customer Premises Configuration



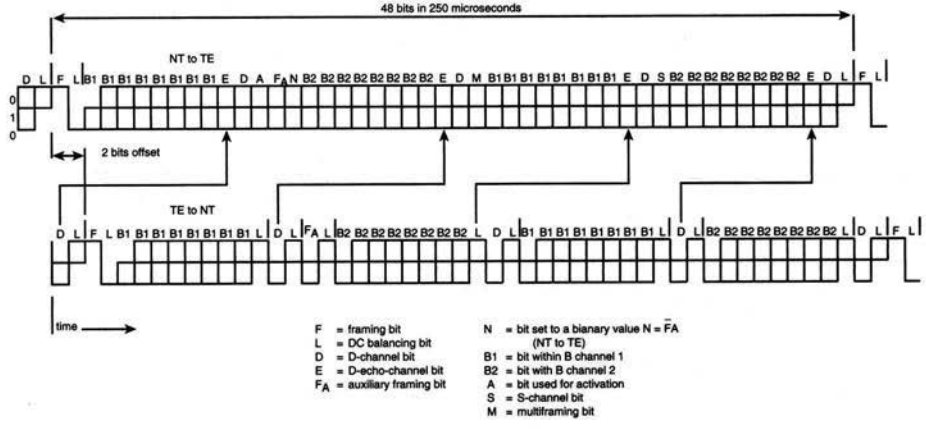
Specifically, as illustrated by Figure 11.44, a T interface frame has a length of 48 bits in 250 μs. The frame starts with a frame bit of value +1 and is followed by a DC balance bit of value –1. There are two bytes each for two B channels for a total of four B channel bytes.

In the NT-to-TE direction, each B channel byte is followed by an E bit, which is the repetition of the last D bit from the TE-to-NT direction. The E bit is followed by the D bit. The D bit is followed by an A, M, S, or L bit for B byte number 1, 2, 3, and 4, respectively. The A bit is used for activation, M is the multiframe bit, S is the S-channel bit, and L is the DC balancing bit. In the TE-to-NT direction, the E bit is replaced by an L bit.

Note

There are an F_A bit and an N bit following the A bit and before the second B byte. The F_A bit is an auxiliary framing bit, and the N bit has the negative value of the F_A bit.

Figure 11.44 Frame Structure at S Interface



There is a two-bit offset between NT-to-TE and TE-to-NT frames. Because the transmission delay is relatively small, the two-bit offset can be approximately maintained among the NT and all TE. There could be up to eight TE installations sharing the same T interface; however, not all eight TE installations can be active if multiplexing of B bytes in the T interface is not implemented.

There is a contention resolution procedure for the TE-to-NT D bit positions. The contention can be detected by monitoring values of E bits. In the case of contention, retransmission will be necessary. To avoid further contention, a random wait will be necessary for TE. The contention of the B channel is totally avoided if the connection of the B channel is established through the D channel with a proper protocol.

The S interface defines some additional signaling protocols between an NT and TE installations for advanced functions, as required by a PBX with the T interface as the physical transmission layer. The U interface is defined as between the twisted-pair telephone loop and a DSL transceiver. The transmission physical layer of a DSL transceiver is also called the U interface.

11.6 DSL Standards

The T1.601-1992[47] addresses interface details for the 2B1Q DSL line code. Specifics covered include the following:

- Transmission methods, which include
  - Pulse shape
  - Transmit power spectrum
  - Transmitter linearity
  - Test loops
  - Test procedures
- Functional characteristics, which include
  - Frame structure
  - Superframe
  - Scrambling method
  - Startup sequence

- M-channel bit functions, which include
  - Cyclic Redundancy Check (CRC)
  - Overhead bits
  - Embedded operation channel (eoc)

Primary constants of typical telephone cables are also included in Annex G of T1.601-1992.

11.6.1 Bit-to-Symbol Encoders

The 2B1Q line code uses the gray code to encode every two bits into a quaternary symbol, as detailed in Table 11.4. A quaternary symbol can have four different possibilities designated by its magnitude for this baseband case.

Table 11.4 The DSL Gray Encoder

First Bit (Sign)	Second Bit (Magnitude)	Quaternary Symbol (Quat)
1	0	+3
1	1	+1
0	1	–1
0	0	–3

11.6.2 DSL Pulse Mask

The DSL line signal from a transmitter should confirm the pulse mask defined in the time domain, as depicted in Figure 11.45. The compliance can be achieved by choosing the proper combination of a transmitter filter and a hybrid circuit. The transmit pulse shape can be measured in real time using either a single pulse or a known pulse sequence with some signal processing. Labels A through H are used for easy reference of any pulse mask violation.

11.6.3 DSL Transmit Power Mask

The upper boundary of the average power spectral density of the transmit signal is shown by Figure 11.46, which can be summarized as that the power density should be –30 dBm/Hz from 0 Hz to 50 kHz, –50 dB/decade attenuation from –30 dBm/Hz at 50 kHz to –80 dBm/Hz at 500 kHz, and –80 dBm/Hz above 500 kHz.

Figure 11.45 DSL Pulse Mask

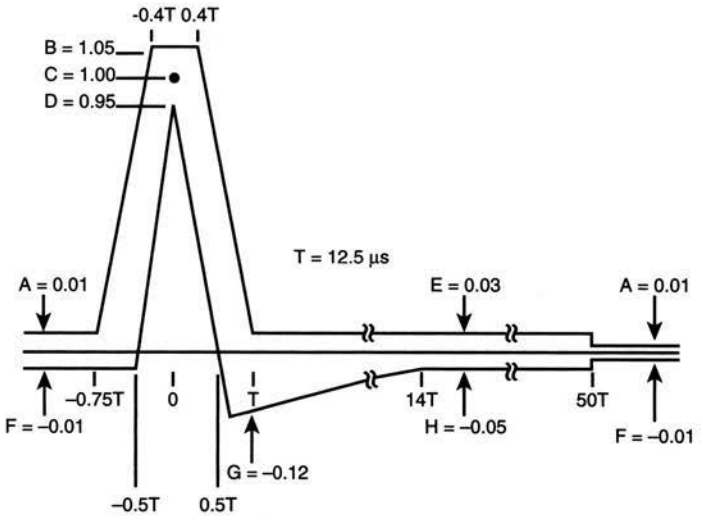
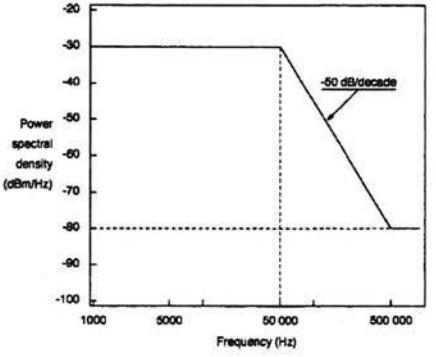


Figure 11.46 DSL Power Mask

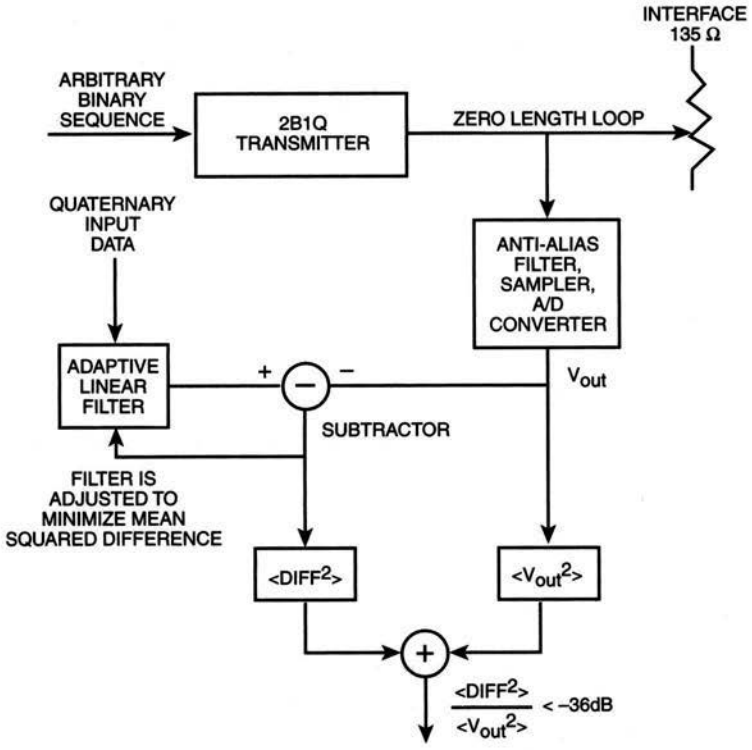


The –50 dB/decade attenuation slope can be obtained with a second-order Butterworth lowpass transmit filter in conjunction with the natural decay of the transmit power above one-half of the baud rate. The total power of the transmit signal is between 13 dBm and 14 dBm over the frequency band between 0 Hz–80 kHz.

11.6.4 Nonlinearity Measurements

The transmitted and received signals should have sufficient linearity so that the residual noise caused by nonlinearity is at least 36 dB below the signal power. The measurement of the transmitter linearity can be performed using a test set up, as illustrated by Figure 11.47. The adaptive linear filter identifies the combination of the 2B1Q transmitter, anti-alias filter, sampler, and A/D converter. The nonlinear noise is obtained by subtracting adaptive filter output from those from the A/D converter.

Figure 11.47 Measurement of Nonlinearity



11.6.5 DSL Transmission Performance Measurement

Figure 11.48 shows a general test setup for the measurement of error rates for a pair of DSL transceivers under certain performance margins. The DSL transceivers are made active in both directions, while the bit error rate of only one direction is measured at a time. Real loops or loop emulators can be used for these test loops. Simulated NEXT noise as well as

power related noise are applied to the receiver front end through high-impedance coupling circuits.

Figure 11.48 Test Environment

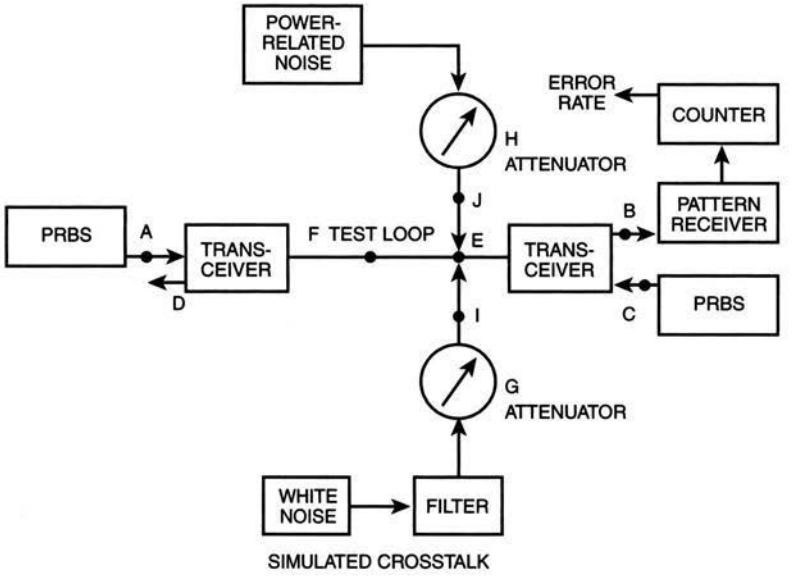
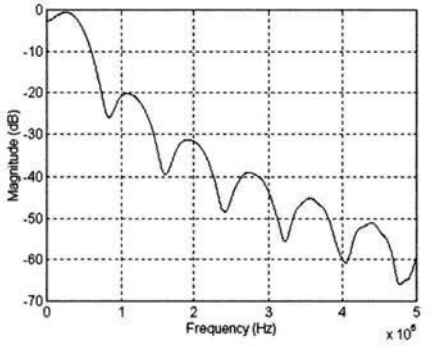


Figure 11.49 shows the transfer function of the NEXT emulation filter—which combines the power spectrum of the DSL transmitter, including the second-order Butterworth transmitter filter and the NEXT coupling function.

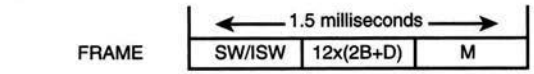
Figure 11.49 NEXT Generation Filter Spectrum



11.6.6 The DSL Frame Structure

A DSL frame has 120 quats with a length of 1.5 milliseconds. Figure 11.50 illustrates a DSL frame that starts with a Synchronization Word (SW) of 9 quats, followed by 108 of information payload quats, which consist of 12 groups of two B channel bytes and two D channel bits (shown in Figure 11.51), and ends with 3 quats of M bits.

Figure 11.50 DSL Frame Structure



Function	Sync word	2B+D	Overhead
# Quats	9	108	3
Quat positions	1-9	10-117	118-120
# Bits	18	216	6
Bit Positions	1-18	19-234	235-240

Frames in the NT-to-network direction are offset from frames in the network-to-NT direction by 60 ± 2 quats.

Figure 11.51 2B1Q Encoding of 2B+D Bit Fields

Data	Time → B_1				B_2				D
Bit pairs	$b_{11}b_{12}$	$b_{13}b_{14}$	$b_{15}b_{16}$	$b_{17}b_{18}$	$b_{21}b_{22}$	$b_{23}b_{24}$	$b_{25}b_{26}$	$b_{27}b_{28}$	d_1d_2
Quat # (relative)	q_1	q_2	q_3	q_4	q_5	q_6	q_7	q_8	q_9
# Bits			8				8		2
# Quats			4				4		1

11.6.7 The DSL Super Frame Structure

A super frame consists of eight frames. The super frame is identified by the first frame whose SW is inverted. The normal SW is +3 +3 −3 −3 −3 +3 −3 +3 +3, and the inverted SW (ISW) is −3 −3 +3 +3 +3 −3 +3 −3 −3. The super frame consists of the following:

- 24 eoc bits
- 12 crc bits
- one far-end block error (febe) bit
- one turn-off (dea) (network to NT) bit
- one startup (act) (network to NT and NT to network) bit
- two power-status (ps) (NT to network) bits

- one NT-in-test-mode (ntm) (NT to network) bit
- one cold-start-only (cso) (NT to network) bit
- one U-interface-only activation (uoa) (network to NT) bit
- one S/T interface activity indicator (sai) (NT to network) bit
- one alarm indicator bit (aib) (network to NT) bit
- three reserved bits

The total is 48 M-bits, which are distributed through the superframe structure, as illustrated in Figure 11.52.

Figure 11.52 DSL Superframe Structure

		FRAMING	2B+D	Overhead Bits (M_1–M_6)					
	Quat positions	1-9	10-117	118s	118m	119s	119m	120s	120m
	Bit positions	1-18	19-234	235	236	237	238	239	240
Superframe #	Basic frame #	Sync word	2B+D	M_1	M_2	M_3	M_4	M_5	M_6
A	1	ISW	2B+D	eoc_{a1}	eoc_{a2}	eoc_{a3}	act	1	1
	2	SW	2B+D	eoc_{dm}	eoc_{i1}	eoc_{i2}	dea	1	febe
	3	SW	2B+D	eoc_{i3}	eoc_{i4}	eoc_{i5}	1	crc_1	crc_2
	4	SW	2B+D	eoc_{i6}	eoc_{i7}	eoc_{i6}	1	crc_3	crc_4
	5	SW	2B+D	eoc_{a1}	eoc_{a2}	eoc_{a3}	1	crc_5	crc_6
	6	SW	2B+D	eoc_{dm}	eoc_{i1}	eoc_{i2}	1	crc_7	crc_8
	7	SW	2B+D	eoc_{i3}	eoc_{i4}	eoc_{i5}	uoa	crc_9	crc_{10}
	8	SW	2B+D	eoc_{i6}	eoc_{i7}	eoc_{i6}	aib	crc_{11}	crc_{12}
B, C, ...									

a) Network ⟶ NT

		FRAMING	2B+D	Overhead Bits (M_1–M_6)					
	Quat positions	1-9	10-117	118s	118m	119s	119m	120s	120m
	Bit positions	1-18	19-234	235	236	237	238	239	240
Superframe #	Basic frame #	Sync word	2B+D	M_1	M_2	M_3	M_4	M_5	M_6
1	1	ISW	2B+D	eoc_{a1}	eoc_{a2}	eoc_{a3}	act	1	1
	2	SW	2B+D	eoc_{dm}	eoc_{i1}	eoc_{i2}	ps_1	1	febe
	3	SW	2B+D	eoc_{i3}	eoc_{i4}	eoc_{i5}	ps_2	crc_1	crc_2
	4	SW	2B+D	eoc_{i6}	eoc_{i7}	eoc_{i6}	ntm	crc_3	crc_4
	5	SW	2B+D	eoc_{a1}	eoc_{a2}	eoc_{a3}	cso	crc_5	crc_6
	6	SW	2B+D	eoc_{dm}	eoc_{i1}	eoc_{i2}	1	crc_7	crc_8
	7	SW	2B+D	eoc_{i3}	eoc_{i4}	eoc_{i5}	sai	crc_9	crc_{10}
	8	SW	2B+D	eoc_{i6}	eoc_{i7}	eoc_{i6}	1'	crc_{11}	crc_{12}
2,3, ...									

b) NT ⟶ Network

11.6.8 DSL Scrambler/Descrambler

All data bits, except the Synchronization Words (SW), are scrambled with a 23rd-order polynomial prior to the 2B1Q encoder. In the NT-to-network direction, the polynomial is

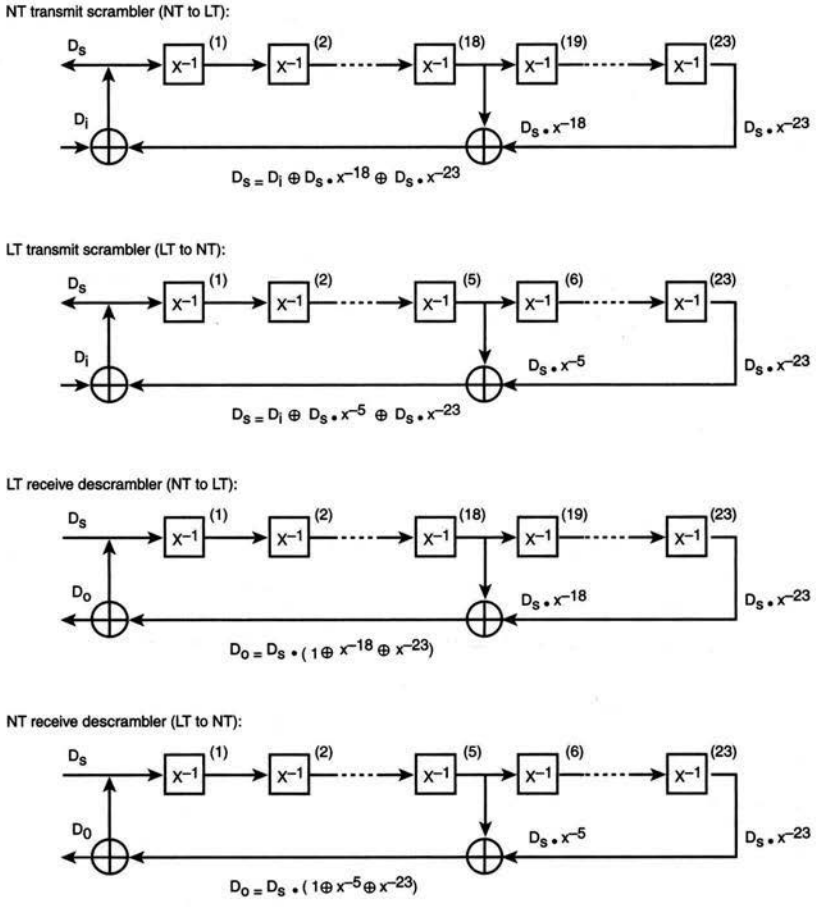
$$x^{-23} \oplus x^{-18} \oplus 1$$

In the network-to-NT direction, the polynomial is

$$x^{-23} \oplus x^{-5} \oplus 1$$

in which $\oplus$ denotes a modular 2 addition. The scramblers and descramblers are self synchronized, as illustrated in Figure 11.53.

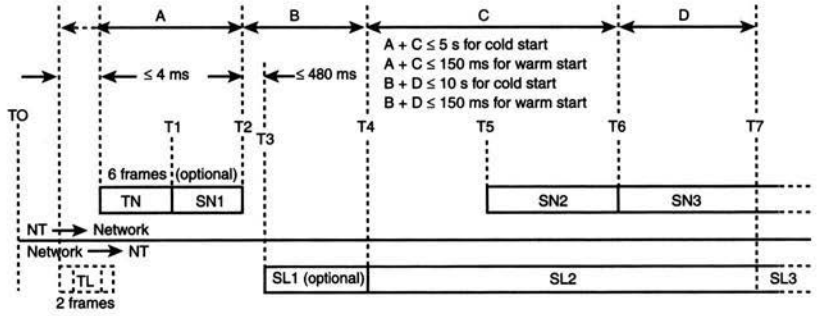
Figure 11.53 Scrambler and Descrambler



11.6.9 The DSL Startup Sequence

The startup sequence involving the network and the NT lasts less than 15 seconds. Figure 11.54 illustrates the timing relationship of the startup sequence.

Figure 11.54 DSL Startup Sequence



The startup is initiated by a wake up tone: either TL (network to NT) or TN (NT to network). The signals TN and TL are 10 kHz tones generated by repeating the following unscrambled and unframed symbol pattern: ... +3 +3 +3 +3 −3 −3 −3 −3.... Six frames after TN, optional SN1 can be applied from NT to network. The contents of different segments are described as follows:

- SN1 and SN2 are scrambled 1s with SW, but no ISW and with 1s for M bits.
- SN3 is scrambled 1s with SW and ISW and normal M bits.
- SL1 is scrambled 1s with SW, but no ISW and with 1s for M bits.
- SL2 and SL3 are scrambled 0s with SW and ISW and normal M bits.

The periods during SN1 and SL1 can be used to train echo cancellers at NT and network ends, respectively. With echo levels, reduced channel equalizers can be trained quickly under the full-duplex mode.

More detailed descriptions and explanations of the standard DSL interface can be found in the ANSI T1.601 standards document at **http://web.ansi.org/**.

End Notes

1. "Integrated Service Digital Network (ISDN): Basic Access Interface for Use on Metallic Loops for Application on the Network Side of the NT-Layer 1 Specification," ANSI T1.601-1992.

2. "ISDN Basic Access Digital Subscriber Lines," *Bellcore Technical Reference,* TR-TSY-000393, Issue 1 (May 1988).

3. N. S. Lin and C-P. J. Tzeng, "Full-Duplex Data Over Local Loops," *IEEE Communication Magazine,* vol. 26, no. 1 (February 1988): 31–42.

4. D. T. Huang and C. F. Valenti, "Digital Subscriber Lines: Network Considerations for ISDN Basic Access Standard," *Proc. IEEE,* vol. 79, no. 2 (February 1991): 125–144.

5. "Integrated Service Digital Network (ISDN): Basic Access Interface for Use on Metallic Loops for Application on the Network Side of the NT-Layer 1 Specification," ANSI T1.601-1988.

6. British Telecom Research Laboratories, "2B1Q Measurement Results," *ECSA Contribution* T1D1.3/86-177.

7. Siemens Communication Systems, "A Comparison of 4B3T and 3B2T Proposals for the Interface at the Network Side of the NT1," *ECSA Contribution* T1D1.3/86-036 (January 1986).

8. J. W. Lechleider, "A Comparison of Crosstalk Performance of Dicode, PR-4 and 3B2T: Using Decision Feedback Equalization," *ECSA Contribution* T1D1.3/86-005 (January 1986).

9. Siemens Communication Systems, Inc., "Some Characteristics of the MMS43 Line Code," *ECSA Contribution* T1D1.3/85-133.

10. J. W. Lechleider, "Estimated Coverage of the Population of Non-Loaded Loops in the 1983 Survey Using AMI, U88, MMS43, and DI43 Line Codes," *ECSA Contribution* T1D1.3/85-154.

11. C. F. Valenti, "Calculation of RMS Loop Insertion Loss for AMI, U88, MMS43, and DI43 Codes for Loops in the 1983 Loop Survey," *ECSA Contribution* T1D1.3/85-159 (August 15, 1985).

12. L. Wu and K. Laker, "Cost Estimates for AMI and MMS43," *ECSA Contribution* T1D1.3/86-007.

13. P. Fleischer and L. Wu, "The DI43 Code and Its Spectral Properties," *ECSA Contribution* T1D1.3/85-154.

14. J. Lechleider, "A Block Code for ISDN Access Studies," *ECSA Contribution* T1D1.3/85-157.

15. R. D. Gitlin, J. F. Hayes, and S. B. Weinstein, *Data Communication Principles* (New York: Plenum Press, 1992).

16. J. Lechleider, "Line Codes for Digital Subscriber Lines," *IEEE Communications Magazine,* vol.9 (September 1989).

17. J. W. Lechleider and R. A. McDonald, "Capability of the Telephone Loop Plant for ISDN Basic Access," *IEEE Int'l Symp. Subs. Loops Serv.* (1986): 156–161.

18. G. Brand, et al., "Comparison of Line Codes with Optimal DFE Design and Bridged Taps," *ECSA Contribution* T1D1.3/86-018.

19. Bell Communications Research, Inc., "The Calculated Performance of Reference Configurations of AMI, MDB, and MMS43 Are Compared," *ECSA Contribution* T1D1.3/86-155.

20. J. W. Lechleider, "DSLs for Use with Correlated Line Codes," *IEEE Trans. On Comm.,* vol. COM-35, no. 10 (October 1987): 1029–1036.

21. Bell Communications Research, Inc., "The DI43 Line Code and Its Spectral Properties," *ECSA Contribution* T1D1.3/85-154.

22. Bell Communications Research, Inc., "Simulation of 2B1Q Performance," *ECSA Contribution* T1D1.3/86-210.

23. AT&T Technology, Inc., "ISDN Access Cost Model," *ECSA Contribution* T1D1.3/86-081.

24. Bellcore, "Sensitivity of LT Line Card Cost to Transceiver Line Code Selection," *ECSA Contribution* T1D1.3/86-008.

25. ITT Europe, "Cost and Complexity of Various Line Codes," *ECSA Contribution* T1D1.3/85-205 (September 1985).

26. BellSouth Services, "Cost, Complexity, and Risk Versus Reach and Coverage for U-Interface Technology," *ECSA Contribution* T1D1.3/85-191 (October 1986).

27. Pacific Bell, "Economic Considerations for Evaluating the Line Code at the U Reference Point," *ECSA Contribution* T1D1.3/85-225 (September 1985).

28. D. V. Batorsky, D. T. Huang, and R. A. McDonald, "Economic Model for Line Code Selection for Digital Subscriber Lines," *ECSA Contribution* T1D1.3/85-230.

29. D. V. Batorsky, "Summary of Line Code Performance and Hardware Costs Supporting Selection of a Block Code as a Standard Line Code," *ECSA Contribution* T1D1.3/86-009 (January 1986).

30. National Semiconductor Corporation, "Cost/Performance Comparison of Various Echo-Cancellation Systems for Digital Loop Transmission," *ECSA Contribution* T1D1.3/85-171.

31. Intel Corporation, "Cost, Performance, and Complexity Comparison of Various Line Codes for Basic Access on the Network Side of NT1," *ECSA Contribution* T1D1.3/85-171.

32. AT&T Technology, Inc., "Estimates of Impact of Chip Area on IC Price," *ECSA Contribution* T1D1.3/86-079.

33. Intel Corporation, "Die Size Versus Price Estimates," *ECSA Contribution* T1D1.3/86-009 (April 7, 1986).

34. ITT Telecom, "Comparison of Die Size Estimates for Implementation of Systems Based on 4B3T and AMI-Dicode (1+D)," *ECSA Contribution* T1D1.3/86-012 (June 1986).

35. D. V. Batorsky and M. L. Liou, "Comparison of Chip Complexity," *ECSA Contribution* T1D1.3/86-109.

36. L. E. Franks and J. P. Bubrouski, "Statistics Properties of Timing Jitter in a PAM Timing Recovery Scheme," *IEEE Trans. Commun.*, vol. COM-22, no. 7 (July 1974): 913–920.

37. J. Tzeng, D.A. Hodges, and D.G. Messerschmitt, "Timing Recovery in Digital Subscriber Loops Using Baud Rate Sampling," *IEEE Trans. on Selected Areas in Communications* (November 1986).

38. B. Roessler and E. Wolter, "CMOS Analog Front End of a Transceiver with Digital Echo Cancellation for ISDN," *IEEE Journal on Solid State Circuits*, vol. 23, no. 2 (April 1988): 311–317.

39. J. C. Candy, "A Use of Double Integration in Sigma-Delta-Modulation," *IEEE Trans. Commun.*, vol. COM-33, no. 3 (1985): 249–258.

40. D. G. Haigh and B. Singh, "A Switching Scheme for Switched Capacitor Filters Which Reduces the Effect of Parasitic Capacitances Associated with Switch Control Terminals," *Proc. ISCAS 1983*: 586–589.

41. H. Khorramabadi, et al., "An ANSI Standard ISDN Transceiver Chip Set," *IEEE ISSCC '89*: 256–257.

42. R. P. Colbeck, et al., "A Single-Chip 2B1Q U-Interface Transceiver," *IEEE Journal on Solid State Circuits*, vol. SC-24, no. 6 (December 1989): 1614–1624.

43. R. Batruni, et al., "Mixed Digital/Analog Signal Processing for a Single-Chip 2B1Q U-Interface Transceiver," *IEEE Journal on Solid State Circuits,* vol. SC-25, no. 6 (December 1990): 1414–1425.

44. "ISDN Basic Access Digital Subscriber Lines," *Bellcore Tech. Ref.* TR-TSY-000393, issue 1 (May 1988).

45. P. Verma, *ISDN Systems* (Prentice-Hall, 1990).

46. American National Standards Institute, Inc., "Integrated Services Digital Network (ISDN) Basic Access Interface for Use on Metallic Loops for Application on the Network Side of the NT (Layer 1 Specification)," ANSI T1.605-1992.

47. American National Standards Institute, Inc., "Integrated Services Digital Network (ISDN): Basic Access Interface for Use on Metallic Loops for Application on the Network Side of the NT (Layer 1 Specification)," ANSI T1.601-1992.

CHAPTER 12

HDSL (High-Bit-Rate Digital Subscriber Lines)

The capability of a digital subscriber line system depends on a particularly defined telephone loop plant transmission environment and also on the required signal processing power supported by current semiconductor technologies. HDSL is simply a technological extension of DSL based on the same 2B1Q baseband line code and the same echo cancellation method.

An HDSL transceiver operates at five times that of DSL. The required signal processing power, however, could be 25 times more because channel and echo path impulse responses become five times longer under the five times higher sampling rate. The transmission throughput improvement of HDSL over DSL is due to the shorter CSA operation range instead of that defined by all non-loaded loops.

T1 lines can provide a transmission throughput of about 1.544 Mbps. T1 lines can be used to provide a high-speed digital link between a CO and a customer through the existing telephone loop plant. The provisioning of the T1 line, however, involves the removal of bridged taps and the installation of repeaters with the traditional T1 technology developed during the early 1960s.

HDSL has been successfully used as the repeaterless T1 technology, achieving a huge provisioning cost savings. With HDSL, the T1 service can be provided whenever transceivers can be installed at both ends of a twisted-pair loop, which is much simpler compared with bridged tap removal and repeater installation. HDSL technology can also be used to provide high-speed data links in a campus or business environment over previously installed twisted-pair cables.

This chapter covers development activities, transmission performance, and interface standards of High-bit-rate Digital Subscriber Lines, including the following:

- *HDSL technology development.* Technical discussions leading to the development of HDSL are reviewed. Early performance measurements on prototype HDSL systems and corresponding transmit power levels are presented.
- *Transceiver structures and line codes.* The general HDSL transceiver architecture and major transmission parameters are discussed. HDSL equalizer and echo canceller requirements are calculated. A symbol rate timing recovery circuit for the application of HDSL is also discussed. HDSL semiconductor technologies and implementation alternatives are reviewed.
- *Performance objectives.* HDSL test loop configurations and corresponding transfer functions as well as NEXT models are presented. HDSL transmission performances are simulated for these test loops under the self-NEXT noise condition. The effect of transceiver termination resistance to the transmission performance of HDSL is studied. Options for dealing with the slight difference in transmission throughputs in opposite directions are discussed.
- *Network interfaces.* Network interface for using HDSL as a dual-duplex repeaterless T1 technology is defined. The HDSL frame structure is reviewed. Possibilities of line powering HDSL transceivers are also discussed.
- *HDSL technical report.* Key features of the ANSI HDSL technical report, including bit-to-symbol encoding, pulse mask, power spectrum mask, additional test loops, performance test configuration, crosstalk coupling circuit, error rate measurement procedures, scrambler, and startup sequence, are revealed. Key features of CAP HDSL are also discussed.

12.1 HDSL Technology Development

The development of HDSL can be summarized as initial system definition discussions, followed by simulation studies, and concludes with prototype performance measurements. In this section, many technical discussions and study results are briefly mentioned for background and historical reasons. Some details are further discussed in later sections of this chapter and other details can be found through referenced T1E1.4 contribution documents.

12.1.1 The Initial Definition of HDSL

TIEI.4 initiated a study project for HDSL with a data rate of 1.544 Mbps in 1988.[1] Early in the study, it was suggested that there were a number of issues that required further study before an HDSL could be implemented,[2] including the following:

- A/D conversion
- Timing
- Signal processing (including exploiting NEXT cyclostationarity)
- Echo cancellation
- Viterbi receivers
- Adaptive transmitters
- Wire center and customer premises wiring characteristics
- Impulse noise
- Other services sharing the same cable sheath as the HDSL
- Architecture issues

In the following year, 1989, it was also pointed out that the technical feasibility and the type of service to be offered would have a bearing on the desirability of establishing a new standard for HDSL.[3] For example, if HDSL is used for ISDN primary access transport, no new standard would be required. The assumptions used in the Bellcore studies of HDSL were presented in an effort to gain a common frame of performance reference.[4]

The critical issues were that the required performance would include a bit error ratio of 10^{-7}, and that the margin used in theoretical and simulation studies would be 12 dB, while the margin on a measured piece of equipment need only be 6 dB. The crosstalk model has a NEXT loss of about 57 dB at 80 kHz and decreases at about 15 dB per decade for frequencies above about 20 kHz. Plant models were taken from the 1983 loop survey, and impulse noise data was taken from a survey of impulse noise by the NYNEX Corporation, the results of which were used in basic access studies. The issue of defining a 6 dB margin was further discussed.[5,6]

12.1.2 *Studies on Baseband 2B1Q Line Code*

Early study results indicated that at DS1 rate (1.544 Mbps), *single-pair HDSLs* using modified basic access technology would have a range of about a mile, or 5.3 kft.[7] It was also pointed out that HDSLs using modified basic access technology, but operating at 800 kbps, would perform satisfactorily on most Carrier Serving Area (CSA) loops, but would have a theoretical margin of about 6 dB on the lossiest loop in a population of CSA loops.[8]

A study of using Trellis coded modulation to make up an extra 4 dB of performance improvement was also reported.[9] The Shannon capacity of subscriber loops using a simple

crosstalk model and identical signal spectra in the two directions of transmission was studied.[10] It was found that the capacity of subscriber loops far exceeds the DS1 rate.

Some possible enhancements[11] to basic access technology can achieve performance that is acceptable on a large percentage of loops. This contention was later supported by data on the distribution of losses in a CSA.[12] The performance of a Decision Feedback Equalizer (DFE) with an infinitely long feedback filter on a population of CSA loops was also calculated.[13] It was found that performance is not enhanced by increasing the length of the feedforward filter in the DFE to more than nine taps, with baud-rate sampling.

12.1.3 Studies on Passband QAM Line Code with Coding

Coded Quadrature Amplitude Modulation (QAM) was proposed as the line code for the HDSL.[14] It was suggested that two-wire transmission at DS1 rates was feasible in a CSA. It was also pointed out that because QAM is a passband code, it could be used above the voice band.

Calculations of the theoretical performance of the 64 QAM HDSL line code[15] were based on a bandwidth of 257.33 kHz and a Trellis coding gain of 4 dB. These theoretical results showed that performance was adequate for the crosstalk model used, which was less conservative than the Bellcore model. In addition, there was no margin. Performances of four-wire simplex and duplex transmission were compared.[16] It was pointed out that the range penalty was suffered by the simplex approach. Further analysis was pursued for the simplex/duplex issue, noting that the best line code used in these two cases might be different.[17]

12.1.4 Studies Toward an HDSL Technical Report

A technical report for the study of the HDSL project was proposed.[18] Cable models for the HDSL study were proposed.[19,20] The performances of HDSL under other common types of systems, such as ISDN and T1 NEXT, were studied.[21,22] The CSA cable loss at 200 kHz was studied.[23] The benefit of coding for band limited channel was discussed.[24] The performance of a combined coding and precoding approach was studied.[25,26] The idea of Tomlinson precoding was also discussed in more detail.[27] A tutorial on the Carrierless AM/PM was also given.[28]

Performances of baseband and passband systems on a 9 kft, 26-gauge loop and on a 12 kft, 24-gauge loop were compared.[29] It was found that self-NEXT, baseband, and passband systems have roughly the same performance, and that signaling at two bits per dimension leads to better performance than signaling at three.

Another result was that baseband systems perform better in the presence of interference from ISDN basic access systems in the same cable. The reason for this is that interference from basic access systems is very low at frequencies below 20 kHz so that the low-frequency portion of the HDSL band is a good place to put power to mitigate the effects of basic access interference. These comparison results were generally confirmed on a large population of CSA loops by another Bellcore study.[30]

Impulse noise wave forms were presented for the HDSL performance study.[31] Comparative study results of impulse noise performance for 2BIQ PAM systems and QAM systems were also reported.[32] In an effort to narrow the possibilities for consideration for HDSL implementation, it was suggested that HDSL configurations to be considered in further study be limited to repeaterless DS1 applications in CSA.[33]

It was discovered that there might be complications with HDSLs that are connected to DS1 lines that have slight frequency or baud offsets.[34] Techniques used in voice band modems to overcome this asynchronous problem were discussed.[35] Further discussions revealed that the asynchronous problem could be avoided by the use of a variable frame.[36] Other variable frame formats were also proposed.[37,38] Performances for different variable frame proposals were compared and the idea of a Double Barker code was introduced.[39]

12.1.5 *Performance Measurements of HDSL Prototypes*

The measurements of experimental HDSLs were first reported.[40] A margin of 3.7 dB on a 12 kft, 24-gauge loop for a full-duplex 772 kbps system that uses the 2B1Q line code was achieved. The measurements of a Carrierless AM/PM HDSL system were also reported.[41] Margins of greater than 6 dB on some CSA loops for a half DS1 rate two-pair simplex system with simulated crosstalk noise were achieved.

Measurements of the performance of a 3B1O (3 Bits per Octal symbol) system were also reported.[42] An operating margin of 3.8 dB in a 49-disturber simulated NEXT environment was achieved. Although the number of Pulse Amplitude Modulation (PAM) levels that achieve optimal performance in a self-NEXT environment is between five and six, the eight-level code was used for simplicity of implementation.

A plan for the laboratory testing of proposed HDSL systems in a normalized environment was proposed,[43] which was similar to that used for the testing of ISDN basic access transceivers. A loop emulator testbed that permits the rapid emulation of a large number of loop configurations for the testing of proposed HDSL architectures was also discussed.[44]

The loop emulator testbed consists of twisted-pair cable sections of different lengths and sizes and mechanical relays. A particular loop configuration can be established through a convenient PC user interface. A collection of eight CSA loop configurations that are representative of the population and may be used for simulation studies of HDSL architecture was presented.[45]

A 6B+D HDSL using the 2B1P (two bits per Penternary [5 level] symbol) line code that was being studied for use in Europe was also reported[46] along with a framing structure for the system.[47] DMT was proposed for HDSL.[48,49] The effect of terminal impedance was further examined for PAM and QAM systems.[50] A 2B1Q HDSL VLSI solution was discussed.[51] The performance of a decision feedback equalizer-based 2B1Q HDSL was further studied.[52] The effect of impulse noise on baseband and passband HDSL systems was further analyzed.[53] The ISDN test loop #12 was also proposed to be used for stretched HDSL performance testing.[54] The adoption of a variable frame format for HDSL was gradually finalized.[55,56,57]

The possibility of including a POTS channel with an HDSL was discussed.[58] The HDSL frame structure was further discussed.[59,60] A VLSI chip for the implementation of an HDSL transceiver was discussed.[61] The effect of temperature variation on the subscriber loop transmission characteristics was reported.[62] HDSL transceiver performance test results were published for units from British Telecom Research Laboratories,[63] Bell-Northern Research,[64] AT&T,[65] Tellabs,[66] and PairGain Technologies.[67] After sharing these test results, it was generally agreed that a dual-duplex 2B1Q HDSL system will be defined mainly as a repeaterless T1 technology[68] with the Carrierless AM/PM system as an alternative line code.[69]

Table 12.1 summarizes results of NEXT margin performance tests for different HDSL transceivers. While performance margins are generally consistent for the 2B1Q line code, the higher terminal impedance seems to help the performance. CAP line code is only used by AT&T's HDSL transceiver.

Table 12.2 shows HDSL transceiver transmit power levels. Signal[1] represents the transmitter power measured with a spectrum analyzer. Signal[2] represents the transmitter power measured with a wide-band meter. NEXT is the calculated NEXT level at 0 dB margin. The CAP HDSL transmitter power and the NEXT power are relatively lower compared with other HDSL systems.

Table 12.1 NEXT Performance Margin Measurements (dB)

	BT Labs.		Bell Northern Research				AT&T		Tellabs				PairGain	
	110 ohms		110 ohms		120 ohms		110 ohms		110 ohms		120 ohms		110 ohms	
Loop #	For.	Rev.	For.	Rev.	For.	Rev.	For.	Rev.	For.	Rev.	For.	Rev.	For.	Rev.
1	4.5	4.5	5.0	5.0	7.0	7.0	8.5	9.0	4.0	4.0	5.0	5.0	5.0	5.5
2	4.0	3.5			7.0	5.5	9.5	5.5		1.5		2.0	5.5	3.5
3	3.0	3.0	4.0	4.5	6.0	6.0	6.5	7.0	3.0	3.5	3.5	3.5	4.0	4.0
4	3.0	3.5	2.5	3.0	4.0	5.0	5.0	6.0	0.5	2.0	-0.5	3.0	3.0	3.5
5	3.0	3.0	3.5	4.5	6.0	6.5	7.0	7.5	3.5	3.5	4.0	4.0	4.5	5.0
6	1.5		3.5		5.5		7.0		3.5		3.5		4.0	
7	4.0	4.0	9.0	5.0	11.0	7.5	14.0	9.0	5.0	5.0	6.5	5.0	9.0	5.5
8	3.5		6.5		8.5		10.5		6.5		6.5		7.0	

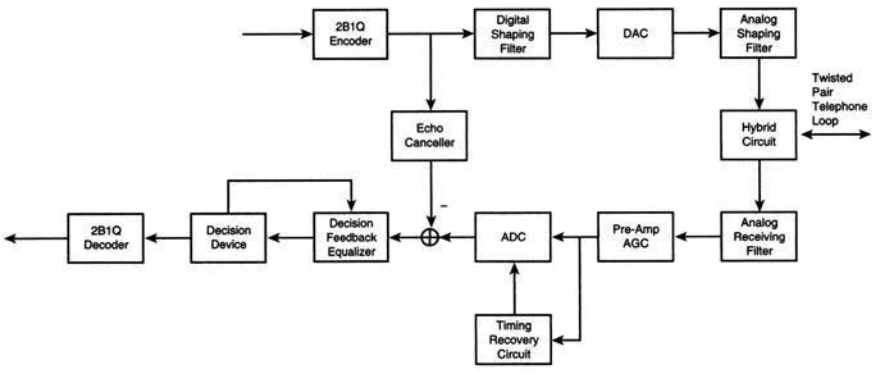
Table 12.2 HDSL Transmit and NEXT Power Level (dBm)

	BT Labs.	Bell Northern Research		AT&T	Tellabs		PairGain
	110 ohms	110 ohms	120 ohms	110 ohms	110 ohms	120 ohms	110 ohms
Signal[1]	12.1	13.4	13.4	-1.1	13.9	13.9	14.1
Signal[2]	12.4	13.5	13.5	-1.0	14.0	14.0	14.3
NEXT	-45.8	-40.2	-40.2	-53.5	-40.3	-40.3	-40.7

12.2 *HDSL Transceiver Structures and Line Codes*

A pair of HDSL transceivers is required at both ends of a subscriber loop to form an HDSL link. A pair of HDSL links is necessary for a dual-duplex repeaterless T1 line. In other words, two HDSL links, each of which has a data rate of 784 kbps, are required to form a T1 line. Figure 12.1 shows the general architecture of an HDSL transceiver. Detailed descriptions of HDSL transceiver components and processes follow.

Figure 12.1 A General HDSL Transceiver Structure



Every two bits of transmit signal are encoded into a symbol through the 2B1Q encoder. The symbol or baud rate is therefore one-half of the bit rate. The symbol is sent to the Digital to Analog Converter (DAC) with an optional digital transmit shaping filter. A fourth order Butterworth lowpass filter is usually sufficient for the transmit analog shaping filter. The purpose of the transmit shaping filter is to control the out-of-band energy. The use of the optional transmit digital shaping filter could reduce the overall transceiver component count.

The filtered transmit signal is connected to the twisted-pair telephone subscriber loop through a hybrid circuit, which has two unidirectional ports (one for transmitting and one for receiving) and one bi-directional port. The bi-directional port is connected to the subscriber loop. If the impedance of the subscriber loop matches the design impedance of the hybrid, there will be a perfect isolation between the transmitting port and the receiving port. For this ideal case, the hybrid return loss is infinity. In reality, the loop impedance is a variable of frequency and is very different for different loops. Therefore, even a well-designed hybrid can sometimes provide a return loss as low as 15 dB at certain frequencies.

A fourth-order Butterworth lowpass filter can also be used for the analog receiver filter. The function of the receiver filter is mainly to minimize out-of-band noise. The preamplifier is necessary to bring the received signal level to that close to the normal transmit signal level for subsequent digital signal processing. A preamplifier gain of about 35 dB might be necessary to compensate for the average CSA loop loss at a frequency of around 200 kHz.

The preamplifier gain should be adjustable to 0 dB for short loops. The gain of the preamplifier should be controlled by the digital signal processing function of the HDSL transceiver. The Analog to Digital Converter is synchronized to the timing recovery circuits. The timing recovery circuits derive the timing information based on the receiver signal after the preamplifier.

These same transmit data symbols are also sent to the echo canceller. The echo canceller resembles the equivalence of the echo path, including transmit shaping filters, DAC, hybrid circuits, analog receiver filter, preamplifier, and ADC at the ADC sampling rate. The DAC's sampling clock is also synchronized to the ADC sampling clock to maintain a stable echo path transfer function. Possible phase jitter between transmit signal and received signal is handled with bit stuffing defined by the HDSL frame structure. The echo path transfer function is identified with an adaptive signal processing algorithm.

The digitized and echo-canceled received signal is further filtered by the decision feedback channel equalizer before being sent to the decision device. A decision feedback equalizer consists of a feedforward filter and a feedback filter. The feedforward filter equalizes the

precursor of the subscriber loop channel impulse response, while the feedback filter cancels the effect of the post-cursor of the channel impulse response.

The decision feedback equalizer is necessary for an HDSL transceiver to maintain a minimal noise enhancement during the channel equalization process. The decision device is a four level threshold detector corresponding to the 2B1Q line code. Received symbols generated from the decision device are converted back to original signal bits through the 2B1Q decoder.

12.2.1 HDSL Transceiver Parameters

Table 12.3 shows the general HDSL transceiver parameters.

Table 12.3 HDSL Transceiver Parameters

Line code	2B1Q
Bandwidth (3 dB)	200 kHz
Baud rate	400 kHz
Throughput	800 kbps
Transmit power	13.38 dBm (135 ohms)

The receiver is based on a baud rate decision feedback channel equalizer. The receiver sampling rate (as well as the baud rate) is 400 kHz, resulting in a 3 dB transmission bandwidth of 200 kHz. For the 2B1Q line code, each baud carries two bits of information. Therefore, the transmission throughput is 800 kbps. Assuming a power supply of five volts and a utilization efficiency of 92 percent, the maximum transmit output voltage is $Z_r(f)2.3$ volts. For a line impedance of 135 ohms, the total transmit power delivered on the telephone loop is as follows:

$$P = \frac{1}{2}\left(2.3^2 + \left(\frac{2.3}{3}\right)^2\right)\frac{1}{135} = \frac{5.29}{2\times 135}\left(1+\frac{1}{9}\right)$$

$$= \frac{5.29\times 10}{2\times 135\times 9} = 0.02177 \text{ Watts} = 13.38dBm$$

The HDSL transmit power spectral density is defined as follows:

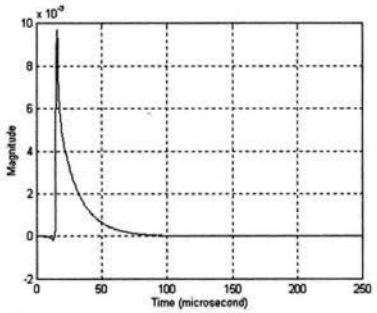
$$PSD = \frac{0.02177}{200\times 10^3} = 1.08\times 10^{-7} \text{ Watts/}Hz = -39.7dBm/Hz$$

12.2.2 HDSL Channel Equalizer Requirements

The length of the decision feedback equalizer should be sufficiently long compared with the equivalent channel impulse response. The equivalent channel response represents the combination of the transmit filter, the transmit transformer, the twisted-pair loop channel, the receiver transformer, and the receiver filter. Assuming that the transmit and the receiver transformers are the same and that the transmit and receiver filters are also the same, the equivalent channel impulse response is the convolution of the twisted-pair loop impulse response twice convolved with the impulse response of the transformer and twice convolved with the transmit/receiver filter.

Figure 12.2 shows the impulse response of the 9 kft, 26-gauge CSA loop.

Figure 12.2 Impulse Response of the 9 kft, 26-Gauge Loop



An HDSL transformer should be designed to have an almost flat frequency response from DC (direct current) to about 200 kHz; however, the transmission of low-frequency energy near DC could lead to excessive requirements on the DFE and echo canceller. In practice, the frequency components near DC occur only with a very low probability. The effective minimum frequency occurs when n consecutive baud symbols are of the same sign. For n=30 corresponding to an occurrence probability of $p=0.5^{30}=9.313\times10^{-10}$, the effective minimum frequency is as follows:

$$f_{em} = \frac{200\times10^3}{30} = 6.667kHz$$

For a loop impedance of 100 ohms, the transformer primary impedance is

$$L_{pri} = \frac{R}{2\pi f_{em}} 2.387mH$$

Figure 12.4 shows the impulse response of a transformer with an equivalent circuit illustrated in Figure 12.3.

Figure 12.3 A Transformer Equivalent Circuit

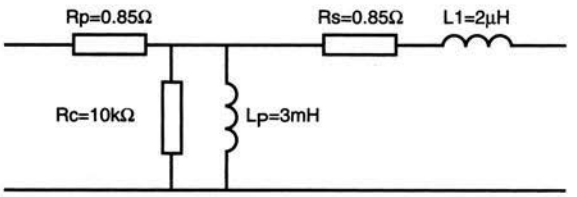


Figure 12.4 Impulse Response of the Transformer

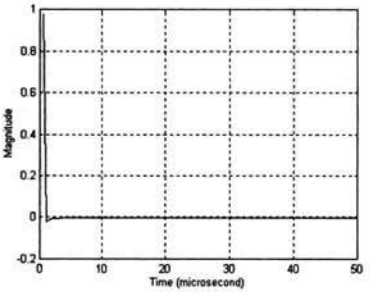


Figure 12.5 shows the impulse response of a fourth-order Butterworth filter, which was recommended for the transmit/receiver filter.

Figure 12.5 Impulse Response of the Fourth-Order Butterworth Lowpass Filter

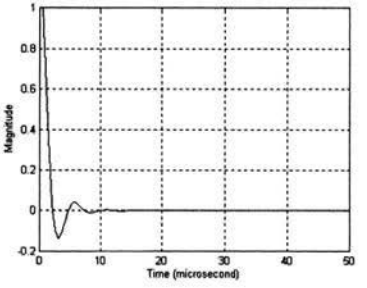
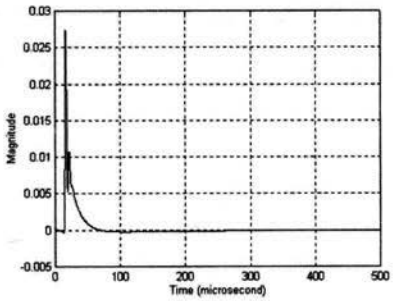


Figure 12.6 shows the impulse response of the equivalent channel, which includes two transformers, two Butterworth lowpass filters, and the twisted-pair loop in cascade.

Figure 12.6 Impulse Response of the Equivalent Channel



The equivalent channel impulse response approaches zero at about 300 microseconds. Considering 20 microseconds of peak delay, the feedback filter should last about 280 microseconds. For a baud sampling rate of 400 kHz, the time between each sample is 2.5 microseconds. Therefore, we need about 280/2.5=122 feedback filter taps, assuming the feedforward filter does not affect the length of the feedback filter too much. The feedforward filter should have a sufficient length to cover the equivalent channel impulse response up to the peak. The required number of feedforward filter taps is therefore 20/2.5=8.

12.2.3 HDSL Echo Canceller Requirements

The echo path transfer function can be calculated according to the following expression:[70]

$$H_{ec}(f) = H_{ec0}(f)\frac{1+\dfrac{Z_n(f)}{Z_i(f)}}{1+\dfrac{Z_d(f)}{Z_i(f)}}$$

The values are as follows:

$$H_{ec0}(f) = \frac{1}{2} - \frac{R_p + R_c \,//\, j2\pi fL_p}{100 + R_p + R_c \,//\, j2\pi fL_p}$$

$$Z_d(f) = R_p + R_c \,//\, j2\pi fL_p \,//\, Z_1(f)$$

$$Z_n(f) = R_p + R_c \,//\, j2\pi fL_p \,//\, Z_2(f)$$

$$Z_1(f) = j2\pi fL_l + R_s + 90.9$$

$$Z_2(f) = j2\pi fL_l + R_s - 1053$$

Further, $Z_l(f)$ is the impedance of the twisted-pair loop. This expression for $H_{ec}(f)$ is for the single-ended hybrid circuit. For the 9 kft, 26-gauge CSA loop, the echo path transfer function is shown by Figure 12.7, and the echo path impulse response is shown by Figure 12.8.

Figure 12.7 Echo Path Transfer Function for the 9 kft, 26-Gauge Loop

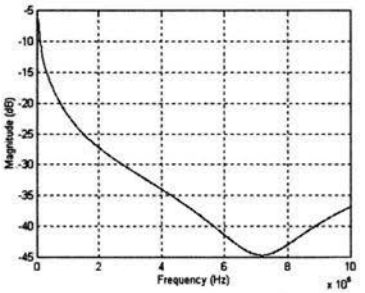
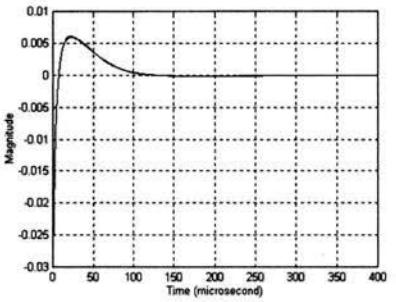


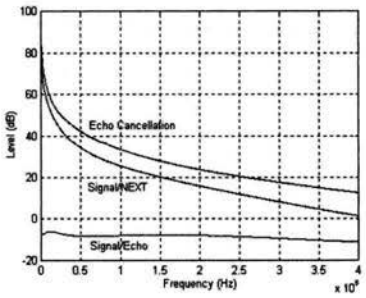
Figure 12.8 Echo Path Impulse Response for the 9 kft, 26-Gauge Loop



The echo path impulse response approaches zero at about 280 microseconds. Echo path impulse responses for other loops exhibit a similar duration. Therefore, we need about 280/2.5=122 filter taps for the echo canceller.

The performance of HDSL is limited by self-NEXT. Figure 12.9 defines the Signal-to-NEXT-Noise ratio from about 65 dB at near DC to about 0 dB at about 400 kHz. This figure also shows the required echo cancellation level, which is the difference between the signal-to-noise ratio and the signal-to-echo ratio.

Figure 12.9 Required Echo Cancellation Level



On the other hand, Figure 12.9 also shows that the signal-to-echo ratio is negative at all frequencies because the echo path return loss is always smaller than the channel loss. The required echo cancellation level, which ranges from 80 dB at near DC to 13 dB at about 400 kHz, is expressed as

$$K_{ec}(f) = \frac{\frac{|H(f)|^2}{NEXT(f)}}{\frac{|H(f)|^2}{|H_{ec}(f)|^2}} = \frac{|H_{ec}(f)|^2}{NEXT(f)}$$

in which

H(f) is the channel transfer function.

$H_{ec}(f)$ is the echo path transfer function.

NEXT(f) is the NEXT power coupling function.

The required average echo cancellation level is defined by

$$K_{ec} = \frac{1}{f_{\max}} \int_0^{f_{\max}} K_{ec}(f)df = 3.09 \times 10^6 = 64.9dB$$

$F_{max} = 400$ kHz.

12.2.4 HDSL Timing Recovery Circuits

The Decision Feedback channel Equalizer (DFE) and the Echo Canceller (EC) of an HDSL transceiver operate on baud sampled signals generated from the receiver's front-end Analog to Digital Converter (ADC). Sampling frequencies of HDSL transceivers at each end of a telephone subscriber loop should be synchronized. Using crystal oscillators, the baud frequency difference between transceivers at each end of the loop can be kept within a few parts per million. However, the accumulation of the frequency difference in a short period of operation could result in a huge sampling frequency discrepancy if a proper synchronization mechanism is not in place.

Among a pair of HDSL transceivers at both ends of a loop, one can use its own crystal to generate the baud frequency driving its digital signal processing circuits, while the other should recover the baud frequency from the received signal. The HDSL transceiver that recovers the baud frequency from the received signal is said to be in *loop timing mode* of operation.

A timing recovery circuit should be used by the loop timing HDSL transceiver to synchronize its digital signal processing to those of the transmitter at the other end of the loop. The timing recovery circuit is usually used to select the optimal sampling phase in conjunction with a particular receiver. The performance of a baud rate DFE is very sensitive to the sampling phase of an ADC. The optimal sampling phase enables the realization of the whole performance potential of a baud-rate-based DFE.

Because the received signal is available at the baud rate, only baud-rate-based timing recovery circuits can be used if additional higher sampling rate ADC is not used. Mueller and Muller proposed an effective baud rate timing recovery circuit architecture in 1976.[71] A timing function can be constructed using a few sampling points around the optimal sample point of a Nyquist channel impulse response. The first proposed timing function is

$$f_1(\tau) = \frac{1}{2}\big(h(\tau + T) - h(\tau - T)\big)$$

$h(t)$ is the channel impulse response.

τ is the sampling instance.

T is the baud interval.

For a Nyquist channel, we have

$$f_1(\tau)=\frac{1}{2}\frac{\sin\pi\left(\frac{\tau+T}{T}\right)}{\pi\left(\frac{\tau+T}{T}\right)}\frac{\cos\alpha\pi\left(\frac{\tau+T}{T}\right)}{1-4\alpha^2\left(\frac{\tau+T}{T}\right)^2}-\frac{1}{2}\frac{\sin\pi\left(\frac{\tau-T}{T}\right)}{\pi\left(\frac{\tau+T}{T}\right)}\frac{\cos\alpha\pi\left(\frac{\tau-T}{T}\right)}{1-4\alpha^2\left(\frac{\tau+T}{T}\right)^2}$$

The value α is the excessive bandwidth. For $\alpha=0$, we have

$$f_1(\tau)=\frac{1}{\pi}\frac{\sin\pi\frac{\tau}{T}}{\frac{\tau^2}{T^2}-1}$$

The second proposed timing function is

$$f_2(\tau)=h(\tau+T)$$

For a Nyquist channel, we have

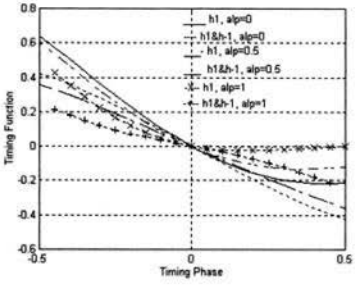
$$f_2(\tau)=\frac{1}{2}\frac{\sin\pi\left(\frac{\tau+T}{T}\right)}{\pi\left(\frac{\tau+T}{T}\right)}\frac{\cos\alpha\pi\left(\frac{\tau+T}{T}\right)}{1-4\alpha^2\left(\frac{\tau+T}{T}\right)^2}$$

which is just a one-baud shifted version of the Nyquist channel impulse response. For $\alpha=0$, we have the one-baud shifted version of the sine function:

$$f_2(\tau)=\frac{\sin\pi\left(\frac{\tau+T}{T}\right)}{\pi\frac{\tau+T}{T}}$$

Figure 12.10 illustrates the sensitivities of these timing functions in relationship to the duration of the baud interval.

Figure 12.10 Timing Function Versus Timing Phase



Such a channel impulse response-based timing function can be derived based on received signal and corresponding data symbols. Data symbols can be recovered from the received data through a threshold decision device for the case of the Nyquist channel. We can define a timing function

$$F\left(\bar{x}_k(\tau), \bar{a}_k(\tau)\right)$$

such that

$$E\left\{F\left(\bar{x}_k(\tau), \bar{a}_k(\tau)\right)\right\} = f(\tau)$$

in which

$$\bar{x}_k(\tau) = \left[x_{k-m+1}(\tau), x_{k-m+2}(\tau), \ldots, x_k(\tau)\right]^T$$

is the received signal vector and

$$\bar{a}_k(\tau) = \left[a_{k-m+1}(\tau), a_{k-m+2}(\tau), \ldots, a_k(\tau)\right]^T$$

is the data symbol vector. Figure 12.11 illustrates the general structure of such a timing recovery circuit.