

A. Scrambler and Descrambler Functions (Informative)

1394B data and control signals are scrambled before 8B10B coding in order to avoid generation of repetitive sequences of identical 10-bit characters. Such repetitive sequences would otherwise be generated during long periods of transmitting a constant control state and during transmission of repetitive data patterns (for example, all zeros). Repetitive sequences may have large amounts of energy concentrated in discrete spectral lines which can lead to electromagnetic compatibility problems. Scrambling causes energy to be spread more uniformly across the transmission frequency band.

Since 1394B signals are scrambled before the 8B10B block code, the properties of the block code (d.c. balance, run length, comma characters) are not affected by the scrambler.

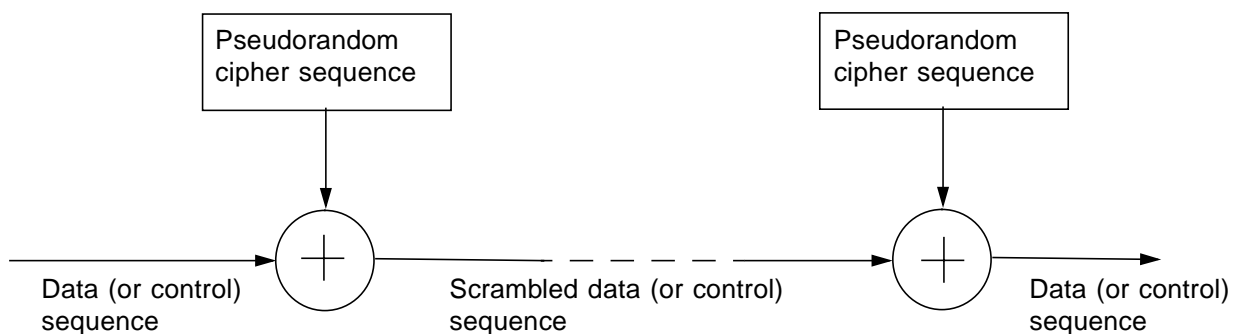
A.1 Scrambler

1394B uses a stream cipher scrambler. In the transmitter, a pseudorandom cipher sequence is added (modulo 2) to the data (or control) sequence to produce a scrambled data (or control) sequence. The pseudorandom cipher sequence is described by the generating polynomial:

$$G(x) = x^{11} + x^9 + 1$$

In the receiver, the same pseudorandom cipher sequence is subtracted (modulo 2) from the scrambled data (or control) sequence to recover the transmitted data (or control) sequence, as illustrated in figure 1.

Figure 1: Overview of scrambler and descrambler operation



The pseudorandom cipher sequence may be generated by a linear feedback shift register which is clocked at the same bit rate as the data sequence. A shift register for the 1394B generating polynomial is illustrated in figure 2. The shift register may also be configured to be clocked at the byte rate of the data sequence, generating successive bytes of the pseudorandom cipher sequence. This configuration is also shown in figure 2. When configured in this way, the state of the shift register at time $k+1$ is related to the state of the register at time k by the transformation matrix T , such that

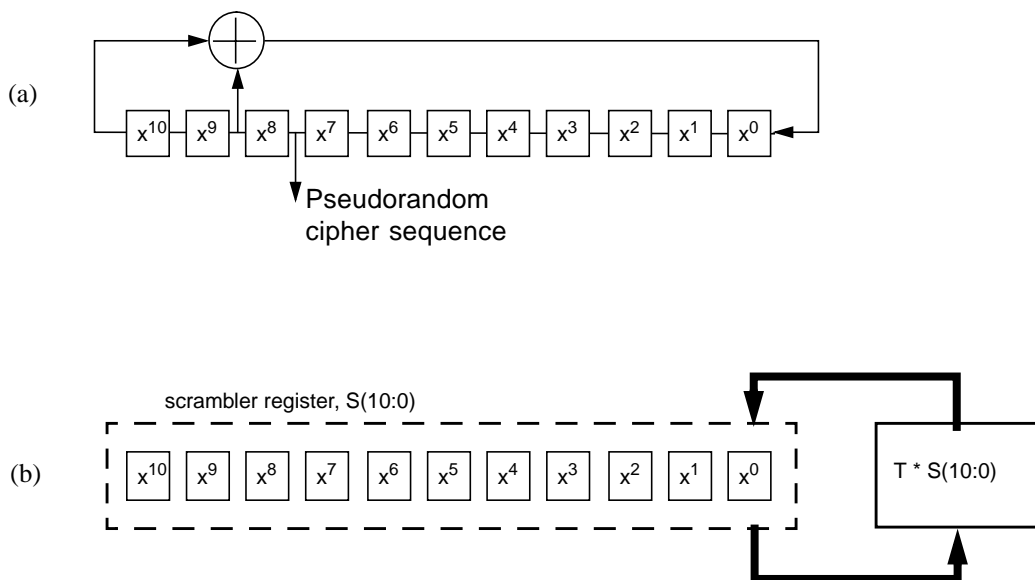
$$S_{k+1} = T \times S_k'$$

where:

$S_k = [S_{10}, S_9, \dots, S_0]$ is the content of the scrambler register at time k

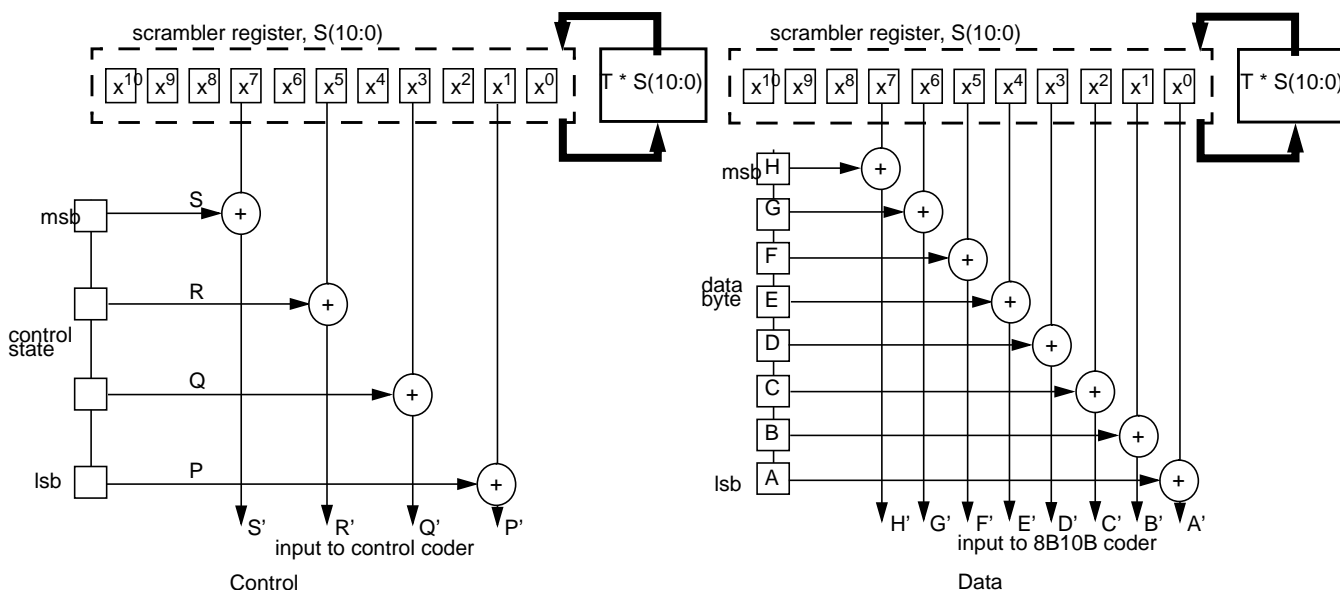
$$T = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0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

Figure 2: Pseudorandom sequence generators (a) clocked at bit rate (b) clocked at byte rate.



During data transmission, the pseudorandom cipher sequence is added modulo 2 to the data sequence. During control transmission control states are represented by four bit symbols, which are then scrambled and mapped to a 10-bit control code character. The effective bit rate during control signaling is therefore half that during data. In order to scramble the control sequence, alternate bits of the pseudorandom cipher sequence are added modulo 2 to the control sequence. (Pseudorandom sequences have the property that samples taken uniformly from the original sequence form an identical pseudorandom sequence at the sampling rate.) This is illustrated in figure 3.

Figure 3: Control and data scrambling



Note: Control and data scrambling is shown separately for clarity. In practice, a single scrambler performs both functions.

A.2 Descrambler

A 1394B receiver descrambles the received data or control sequence by subtracting the same pseudorandom cipher sequence used for scrambling. Since the subtraction is modulo 2, this is equivalent to adding the pseudorandom cipher sequence i.e. the procedure is identical to that used for scrambling in the transmitter. The receiver must ensure that the state of the descrambler register is synchronized with that of the scrambler register.

Synchronization of a 1394B descrambler may be achieved during reception of certain control states, in particular the control states IDLE and TX_REQUEST. These control states are mapped to control symbols 0000 and 0001 respectively. During IDLE and TX_REQUEST, provided that the receiver has achieved 10-bit character synchronization with respect to the transmitter, the scrambled control symbol at the output of the control decoder at time k, P'Q'R'S', will always have the following properties:

$$S' = S_k(7)$$

$$Q' = S_k(3)$$

As a result, regular samples of the scrambler state are transmitted to the receiver during IDLE and TX_REQUEST. These samples may be used to train the descrambler until its state is synchronized with that of the scrambler¹. In general, at most 11 valid samples of the scrambler state are required to synchronize the receiver descrambler. Training is accomplished by adding a correction matrix to the descrambler update function:

$$D_{k+1} = T \times D_k' + C$$

where:

$D_k = [D_{10}, D_9, \dots, D_0]$ is the content of the descrambler register at time k

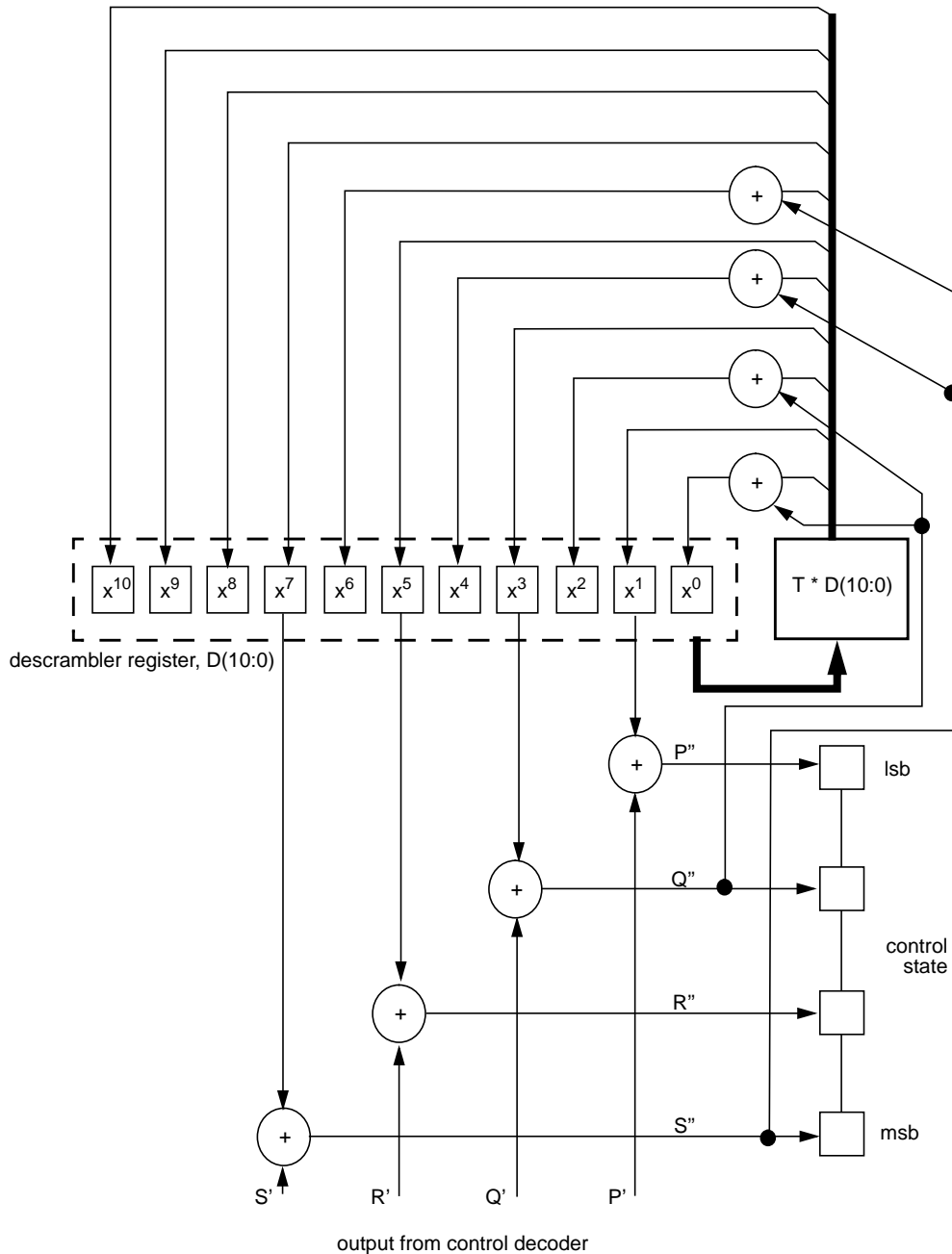
¹ Further details may be found in "Synchronization of shift register generators in distributed sample scramblers", Kim, S.C. and Lee, B. G., IEEE Trans. Comms., vol.42, Feb 1994, pp.1400-1408.

$$C = \begin{bmatrix} 0 & 0 & 0 & 0 & S'' & 0 & S'' & 0 & Q'' & 0 & Q'' \end{bmatrix}$$

Q'' and S'' are the results of descrambling bits P' and S' of the scrambled control sequence

This training procedure is illustrated in figure 4.

Figure 4: Descrambler training.



When the descrambler has been trained, the training procedure may be disabled by forcing the correction matrix to be null. This should happen before the received sequence changes from IDLE or TX_REQUEST.

Alistair Coles, September 19, 1997