number of bits in the DATA field, N_{DATA}; and the number of pad bits, N_{PAD}, are computed from the length of the PSDU (LENGTH) as follows:

$$N_{SYM} = \text{Ceiling}\left((16 + 8 \times \text{LENGTH} + 6)/N_{DBPS}\right) \quad (18\text{-}11)$$

$$N_{DATA} = N_{SYM} \times N_{DBPS} \quad (18\text{-}12)$$

$$N_{PAD} = N_{DATA} - (16 + 8 \times \text{LENGTH} + 6) \quad (18\text{-}13)$$

The function Ceiling (.) is a function that returns the smallest integer value greater than or equal to its argument value. The appended bits ("pad bits") are set to 0 and are subsequently scrambled with the rest of the bits in the DATA field.

An example of a DATA field that contains the SERVICE field, DATA, tail, and pad bits is given in L.1.5.1.

18.3.5.5 PLCP DATA scrambler and descrambler

The DATA field, composed of SERVICE, PSDU, tail, and pad parts, shall be scrambled with a length-127 frame-synchronous scrambler. The octets of the PSDU are placed in the transmit serial bit stream, bit 0 first and bit 7 last. The frame synchronous scrambler uses the generator polynomial *S(x)* as follows, and is illustrated in Figure 18-7:

$$S(x) = x^7 + x^4 + 1 \quad (18\text{-}14)$$

The 127-bit sequence generated repeatedly by the scrambler shall be (leftmost used first), 00001110 11110010 11001001 00000010 00100110 00101110 10110110 00001100 11010100 11100111 10110100 00101010 11111010 01010001 10111000 1111111, when the all ones initial state is used. The same scrambler is used to scramble transmit data and to descramble receive data. When transmitting, the initial state of the scrambler shall be set to a pseudorandom nonzero state. The seven LSBs of the SERVICE field shall be set to all zeros prior to scrambling to enable estimation of the initial state of the scrambler in the receiver.

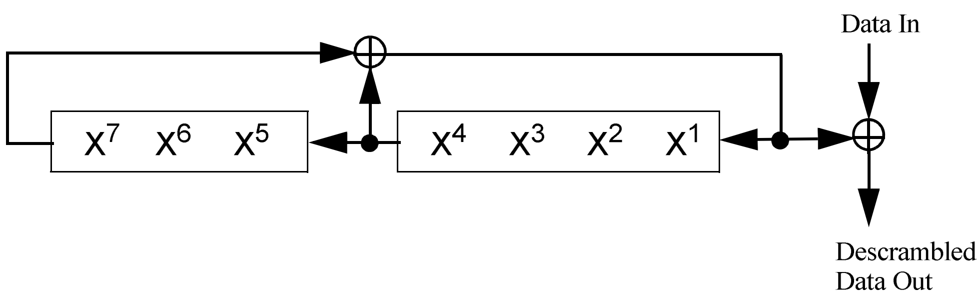

Figure 18-7—Data scrambler

An example of the scrambler output is illustrated in L.1.5.2.

18.3.5.6 Convolutional encoder

The DATA field, composed of SERVICE, PSDU, tail, and pad parts, shall be coded with a convolutional encoder of coding rate R = 1/2, 2/3, or 3/4, corresponding to the desired data rate. The convolutional encoder shall use the industry-standard generator polynomials, $g_0 = 133_8$ and $g_1 = 171_8$, of rate R = 1/2, as shown in Figure 18-8. The bit denoted as "A" shall be output from the encoder before the bit denoted as "B." Higher rates are derived from it by employing "puncturing." Puncturing is a procedure for omitting some of the