The mapping of the information octets $\boldsymbol{m} = (m_{238}, m_{237}, \ldots, m_0)$ to codeword octets $\boldsymbol{c} = (m_{238}, m_{237}, \ldots, m_0, r_{15}, r_{14}, \ldots, r_0)$ is achieved by computing the remainder polynomial $r(x)$ [Equation (15)]:

$$r(x) = \sum_{k=0}^{15} r_k x^k = x^{16} m(x) \bmod g(x) \tag{15}$$

where $m(x)$ is the information polynomial [Equation (16)]:

$$m(x) = \sum_{k=0}^{238} m_k x^k \tag{16}$$

and r_k, $k = 0, \ldots, 15$, and m_k, $k = 0, \ldots, 238$, are elements of GF(2^8). The message order is as follows: m_{238} is the first octet of the message and m_0 is the last octet of the message.

For a shortened RS(L_{inf} + 16, L_{inf}), 239-L_{inf} zero elements are appended to the incoming L_{inf} octet message as follows:

$m_k = 0$, $k = L_{inf}, \ldots, 238$

These inserted zero elements are not transmitted. A shift-register implementation of the RS encoder RS(L_{inf}+16, L_{inf}) is shown in Figure 166, with additions and multiplications over GF(2^8). After m_0 has been inserted into the shift register, the switch shall be moved from the message polynomial input connection to the shift register output connection (right-to-left).

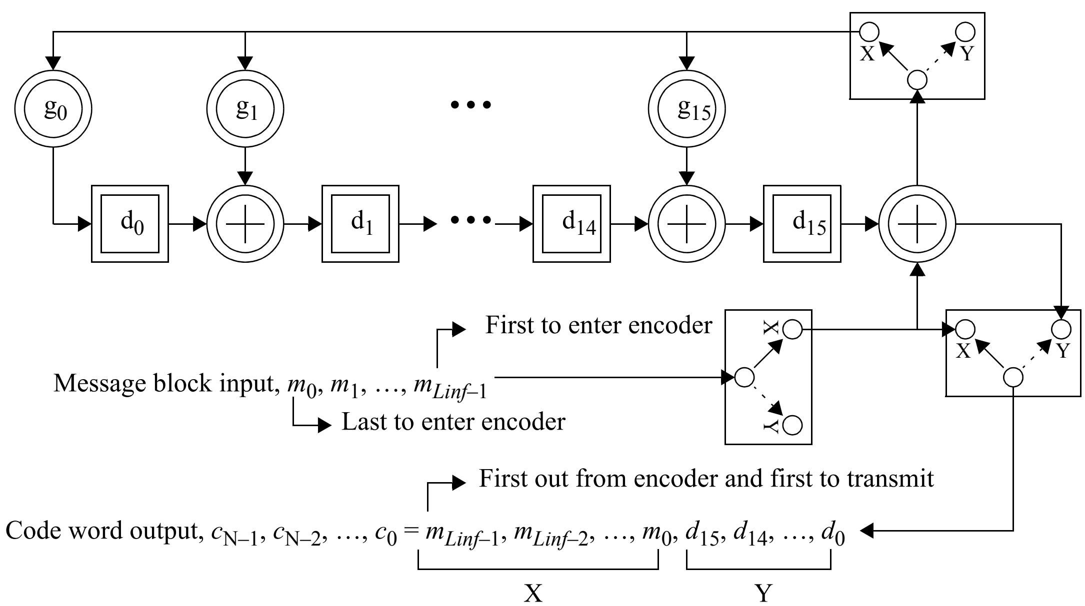

Figure 166—Reed Solomon encoder GF(2^8)

12.2.2.6.2 Irregular LDPC codes

The irregular LDPC codes are used as a high-performance error correction coding technique. The supported FEC rates, information block lengths, and codeword block lengths are described in Table 109.

Table 109—Irregular LDPC parameters

FEC rate, R_{FEC}	LDPC information block length (bits), L_{inf}	LDPC codeword block length (bits), L_{FEC}
1/2	336	672
3/4	504	672
7/8	588	672

The LDPC encoder is systematic, i.e., it encodes an information block of size k, $i = (i_0, i_1, \ldots, i_{(k-1)})$, into a codeword $\mathbf{c}$ of size n, $\mathbf{c} = (i_0, i_1, \ldots, i_{(k-1)}, p_0, p_1, \ldots, p_{(n-k-1)})$, by adding n-k parity bits obtained so that $\mathbf{Hc}^\mathbf{T} = 0$, where $\mathbf{H}$ is an $(n - k) \times n$ parity check matrix.

Each of the parity-check matrices can be partitioned into square subblocks (submatrices) of size $z \times z$ ($z = 21$). These submatrices are either cyclic-permutations of the identity matrix or null (all-zero) submatrices.

The cyclic-permutation matrix $\mathbf{p}^i$ is obtained from the $z \times z$ identity matrix by cyclically shifting the columns to the left by i elements. The matrix $\mathbf{p}^0$ is the $z \times z$ identity matrix.

In the following, an example of cyclic-permutation matrices with $z = 21$ is shown. The matrix $\mathbf{p}^1$ and $\mathbf{p}^2$ are produced by cyclically shifting the columns of the identity matrix $\mathbf{I}_{21\times21}$ to the left by 1 and 2 places, respectively.

$$p^0 = \begin{bmatrix} 1 & 0 & \ldots & \ldots & 0 \\ 0 & 1 & 0 & \ldots & 0 \\ \ldots & 0 & \ldots & 0 & \ldots \\ 0 & \ldots & 0 & 1 & 0 \\ 0 & \ldots & \ldots & 0 & 1 \end{bmatrix}, p^1 = \begin{bmatrix} 0 & \ldots & \ldots & 0 & 1 \\ 1 & 0 & \ldots & \ldots & 0 \\ 0 & 1 & 0 & \ldots & 0 \\ \ldots & 0 & 1 & 0 & 0 \\ 0 & \ldots & 0 & 1 & 0 \end{bmatrix}, p^2 = \begin{bmatrix} 0 & \ldots & 0 & 1 & 0 \\ 0 & \ldots & \ldots & 0 & 1 \\ 1 & 0 & \ldots & \ldots & 0 \\ 0 & 1 & 0 & \ldots & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

where all the above matrices have dimension 21 × 21.

Due to the cyclic permutation, $\mathbf{p}^{21} = \mathbf{p}^0 = \mathbf{I}_{21\times21}$.

Figure 167 displays the matrix permutation indices of parity-check matrices for all three FEC rates at block length $n = 672$ bits. The integer i denotes the cyclic-permutation matrix $\mathbf{p}^i$, as explained in above example. The '–' entries in the table denote null (all zero) submatrices.

For shortened LDPC operation, the k–l zero elements are appended to the incoming l message bits as follows: $r_i = 0$ for $i = l, l+1, \ldots, k-1$. These inserted zero elements are not transmitted.

12.2.2.6.3 Rate 14/15 LDPC code

The rate 14/15 LDPC(1440,1344) code is also systematic, i.e., the LDPC encoder encodes an information block of size k, $\mathbf{i} = (i_0, i_1, \ldots, i_{(k-1)})$, into a codeword $\mathbf{c}$ of size n, $\mathbf{c} = (i_0, i_1, \ldots, i_{(k-1)}, p_0, p_1, \ldots, p_{(n-k-1)})$, by adding $n - k$ parity bits obtained so that $\mathbf{Hc}^\mathrm{T} = 0$, where $\mathbf{H}$ is an $(n - k) \times n$ parity check matrix. Denote the 96×1440 parity check matrix as $\mathbf{H} = (h_{i,j})$, where $h_{i,j}$ consists of $\{0,1\}$, $0 \le i < 96$ and $0 \le j < 1440$. Table 110 shows the matrix elements whose values are '1' in the first 15 columns of parity check matrix.

(672,336), Code rate: 1/2

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
1	-	-	-	5	-	18	-	-	-	-	3	-	10	-	-	-	-	-	-	5	-	-	-	-	-	-	-	5	-	7	-	-
2	0	-	-	-	-	-	16	-	-	-	-	6	-	-	-	0	-	7	-	-	-	-	-	-	-	10	-	-	-	-	-	19
3	-	-	6	-	7	-	-	-	-	2	-	-	-	-	9	-	20	-	-	-	-	-	-	-	-	-	19	-	10	-	-	-
4	-	18	-	-	-	-	-	0	10	-	-	-	-	16	-	-	-	-	9	-	-	-	-	-	4	-	-	-	-	-	17	-
5	5	-	-	-	-	-	18	-	-	-	-	3	-	10	-	-	5	-	-	-	-	-	-	-	-	-	-	-	-	-	7	-
6	-	0	-	-	-	-	-	16	6	-	-	-	0	-	-	-	-	-	7	-	-	-	-	-	-	-	-	-	19	-	-	-
7	-	-	-	6	-	7	-	-	-	-	2	-	-	-	-	9	-	20	-	-	-	-	-	-	-	-	-	-	-	10	-	-
8	-	-	18	-	0	-	-	-	-	10	-	-	-	-	16	-	-	-	-	9	-	-	-	-	-	-	-	-	-	-	-	17
9	-	5	-	-	-	-	-	18	3	-	-	-	-	-	10	-	-	5	-	-	4	-	-	-	-	5	-	-	-	-	-	7
10	-	-	0	-	16	-	-	-	-	6	-	-	-	0	-	-	-	-	-	7	-	4	-	-	-	-	-	10	-	19	-	-
11	6	-	-	-	-	-	7	-	-	-	-	2	9	-	-	-	-	-	20	-	-	-	4	-	19	-	-	-	-	-	10	-
12	-	-	-	18	-	0	-	-	-	-	10	-	-	-	-	16	9	-	-	-	-	-	-	12	-	-	4	-	17	-	-	-
13	-	-	5	-	18	-	-	-	-	3	-	-	-	-	-	10	-	-	5	-	-	-	-	-	-	-	5	-	-	-	-	-
14	-	-	-	0	-	16	-	-	-	-	6	-	-	-	0	-	7	-	-	-	-	-	-	-	10	-	-	-	-	-	-	-
15	-	6	-	-	-	-	-	7	2	-	-	-	-	9	-	-	-	-	-	20	-	-	-	-	-	19	-	-	-	-	-	-
16	18	-	-	-	-	-	0	-	-	-	-	10	16	-	-	-	-	9	-	-	-	-	-	-	-	-	-	4	-	-	-	-

(672,504), Code rate: 3/4

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
1	0	-	-	5	-	18	16	-	-	-	3	6	10	-	-	0	-	7	-	5	-	-	4	4	-	10	-	5	-	-	-	-
2	-	18	6	-	7	-	-	0	10	2	-	-	-	16	9	-	20	-	9	-	4	12	-	-	4	-	19	-	-	-	-	-
3	5	0	-	-	-	-	18	16	6	-	-	3	0	10	-	-	5	-	7	-	4	-	-	4	5	-	10	-	19	-	-	-
4	-	-	18	6	0	7	-	-	-	10	2	-	-	-	16	9	-	20	-	9	-	4	12	-	-	4	-	19	-	10	-	-
5	-	5	0	-	16	-	-	18	3	6	-	-	-	0	10	-	-	5	-	7	4	4	-	-	-	5	-	-	-	-	-	-
6	6	-	-	18	-	0	7	-	-	-	10	2	9	-	-	16	9	-	20	-	-	-	4	12	19	-	-	-	-	-	-	-
7	-	-	5	0	18	16	-	-	-	3	6	-	-	-	0	10	7	-	5	-	-	4	4	-	10	-	5	-	7	-	19	-
8	18	6	-	-	-	-	0	7	2	-	-	10	16	9	-	-	-	9	-	20	12	-	-	4	-	19	-	4	-	17	-	10

(672,588), Code rate: 7/8

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
1	0	18	6	5	7	18	16	0	10	2	3	6	10	16	9	0	20	7	9	5	4	12	4	4	4	10	19	5	10	-	-	-
2	5	0	18	6	0	7	18	16	6	10	2	3	0	10	16	9	5	20	7	9	4	4	12	4	5	4	10	19	19	10	-	-
3	6	5	0	18	16	0	7	18	3	6	10	2	9	0	10	16	9	5	20	7	4	4	4	12	19	5	4	10	17	19	10	-
4	18	6	5	0	18	16	0	7	2	3	6	10	16	9	0	10	7	9	5	20	12	4	4	4	10	19	5	4	7	17	19	10

Figure 167—Matrix permutation indices of structured parity check matrices

Table 110—Positions of '1's in the first 15 columns of parity check matrix H (codeword block length L_{FEC} = 1440)

$h_{0,0}$ $h_{1,0}$ $h_{4,0}$
$h_{32,1}$ $h_{34,1}$ $h_{39,1}$
$h_{64,2}$ $h_{70,2}$ $h_{78,2}$
$h_{8,3}$ $h_{18,3}$ $h_{95,3}$
$h_{31,4}$ $h_{42,4}$ $h_{54,4}$
$h_{63,5}$ $h_{76,5}$ $h_{91,5}$

Table 110—Positions of '1's in the first 15 columns of parity check matrix H (codeword block length L_{FEC} = 1440) *(continued)*

$h_{14,6}$ $h_{45,6}$ $h_{94,6}$
$h_{30,7}$ $h_{47,7}$ $h_{83,7}$
$h_{17,8}$ $h_{62,8}$ $h_{80,8}$
$h_{28,9}$ $h_{48,9}$ $h_{82,9}$
$h_{22,10}$ $h_{60,10}$ $h_{81,10}$
$h_{27,11}$ $h_{49,11}$ $h_{84,11}$
$h_{7,12}$ $h_{53,12}$ $h_{77,12}$
$h_{19,13}$ $h_{44,13}$ $h_{85,13}$
$h_{6,14}$ $h_{46,14}$ $h_{75,14}$

For $15 \le j$, the matrix element can be obtained by using Equation (17).

$$h_{i,j} = h_{\mathrm{mod}(i + \lfloor j/15 \rfloor,\ 96),\ \mathrm{mod}(j,\ 15)} \tag{17}$$

where mod(x, y) is the modulo function and is defined as $x - n \times y$ where n is the nearest integer less than or equal to x/y.

The LDPC(1440, 1344) code is a quasi-cyclic code such that every cyclic shift of a codeword by 15 symbols yields another codeword.

For shortened LDPC operation, the $k–l$ zero elements are appended to the incoming l message bits as follows: $r_i = 0$ for $i = l, l+1, \ldots, k–1$. The message order is $r_{k–1}$ as the first bit of the message with r_0 as the last bit of the message. These inserted zero elements are not transmitted.

12.2.2.7 Stuff bits

Stuff bits shall be added to the end of the encoded MAC frame body if the number of the encoded data bits is not an integer multiple of the length of the data portion in the subblock. The number of stuff bits is computed for each subframe if standard aggregation is employed. The calculation of stuff bits is as follows.

In the encoded MAC frame body, the number of FEC codewords, N_{FEC} is given by Equation (18).

$$N_{FEC} = \mathrm{CEIL}[(L_{MFB} \times 8)/(L_{FEC} \times R_{FEC})] \tag{18}$$

where L_{FEC} is the FEC codeword length, L_{MFB} is the length of the MAC frame body in octets, and R_{FEC} is the FEC rate. The FEC codeword length, L_{FEC}, is 2040 for the RS FEC specified in 12.2.2.6.1, 672 for the irregular LDPC specified in 12.2.2.6.2, and 1440 for the rate 14/15 LDPC specified in 12.2.2.6.3.

The encoded MAC frame body shall be concatenated with stuff bits of length L_{STUFF} so that the resulting MAC frame body is aligned on the subblock symbol boundary. The stuff bits shall be set to zero and then scrambled using the continuation of the scrambler sequence that scrambled the MAC frame body in 12.2.2.10. The length of bits in the encoded MAC frame body, L_{ebits} is given by Equation (19).

$$L_{ebits} = 8 \times L_{MFB} + N_{FEC} \times (1 - R_{FEC}) \times L_{FEC} \tag{19}$$