101.3.2.3 FEC ~~encoding~~Encode ~~-~~and Data Detector processes (FDD)

The {EPoC_PMD_Name} encodes the transmitted data using a systematic Low-Density Parity-Check (LDPC) (F_C, F_P) code. A LDPC encoder encodes F_P information bits $i_{0 \ldots} i_{F_P - 1}$ into a codeword

$$c = (i_0, \ldots, i_{F_P - 1}, p_{F_P}, \ldots, p_{F_C - 1})$$

by adding F_R parity bits $p_{F_P \ldots} p_{F_C - 1}$ obtained so that

$$Hc^T = 0$$

where H is an $F_R \times F_C$ binary matrix containing mostly '0' and relatively few '1', called low-density parity-check matrix. (see [1] and [2]). The detailed description of such parity check matrices is given in 101.3.2.3.1.

{to be included in informative references: [1] R. G. Gallager, "Low density parity check codes," IRE Trans. Inform. Theory, vol. IT-8, pp. 21–28, Jan. 1962.; [2] T. Richardson and R. Urbanke, "Modern Coding Theory," Cambridge University Press, 2008}

The CLT {EPoC_PMD_Name} PCS operating on amplified CCDN shall encode the transmitted data using one of the LDPC (F_C, F_P) codes per Table 101–1, as selected using register TBD. The CNU {EPoC_PMD_Name} PCS operating on amplified CCDN shall encode the transmitted data using one of the LDPC (F_C, F_P) codes per Table 101–2, as selected using register TBD.

Table 101–1—LDCP codes used by the CLT {EPoC_PMD_Name} PCS for amplified CCDN

Codeword F_C [bits]	Payload F_P [bits]	Parity F_R [bits]	Payload			Parity		
			65-bit blocks B_Q	CRC bits	Padding bits B_P	65 -bit blocks C_Q	Parity bits in last block C_{PL}	Padding bits C_P
16200	14400	1800	220	40	60	28	20	45

Annex 101A gives an example of LDPC (F_C, F_P) FEC encoding. {we will need to select one of the codes from the family of codes we use in either downstream or upstream and then generate examples}

101.3.2.3.1 LDPC matrix definition

The low-density parity check matrix H for LDPC (F_C, F_P) encoder can be divided into blocks of L^2 sub-matrices. Its compact circulant form is represented by an m × n block matrix:

where the submatrix $H_{i,j}$ is an L × L all-zero submatrix or a cyclic right-shifted identity submatrix. The last n–m sub-matrix columns represent the parity portion of the matrix. Moreover, $nL = F_C$, $mL = F_P$ and the code rate is $(n–m)/n = (F_C–F_P)/F_C$. In this specification, the sub-matrix size L is called the lifting factor.

$$H = \begin{bmatrix} H_{1,1} & H_{1,2} & H_{1,3} & \dots & H_{1,n} \\ H_{2,1} & H_{2,2} & H_{2,3} & \dots & H_{2,n} \\ H_{3,1} & H_{3,2} & H_{3,3} & \dots & H_{3,n} \\ \dots & \dots & \dots & & \dots \\ H_{m,1} & H_{m,2} & H_{m,3} & \dots & H_{m,n} \end{bmatrix}$$

Table 101–2—LDCP codes used by the CLT {EPoC_PMD_Name} PCS for amplified CCDN

Codeword F_C [bits]	Payload F_P [bits]	Parity F_R [bits]	Payload			Parity		
			65-bit blocks B_Q	CRC bits	Padding bits B_P	65-bit blocks C_Q	Parity bits in last block C_{PL}	Padding bits C_P
16200	14400	1800	220	40	60	28	20	45
5940	5040	900	76	40	60	14	30	35
1120	840	280	12	40	20	4	60	5

In this specification, the sub-matrix $H_{i,j}$ is represented by a value in {-1, 0,…, L-1}, where a '-1' value represents an all-zero submatrix, and the remaining values represent an L × L identity submatrix cyclically right-shifted by the specified value. Such representation of the parity-check matrix is called a base matrix.

Table 101–3 presents a 5 × 45 base matrix of the low-density parity-check matrix H for LDPC (16200, 14400) code listed in Table 101–1 for downstream and Table 101–2 for upstream, respectively. The lifting factor of the matrix is L=360.

Table 101–3—LDPC (16200, 14400) code matrix

Columns	Rows				
	1	**2**	**3**	**4**	**5**
1	93	274	134	-1	253
2	271	115	355	-1	273
3	-1	329	175	184	90
4	83	338	24	70	-1
5	26	124	253	247	-1
6	208	-1	242	14	151
7	245	293	-1	22	311

Table 101–3—LDPC (16200, 14400) code matrix *(continued)*

Columns	Rows				
	1	**2**	**3**	**4**	**5**
8	200	-1	187	7	320
9	-1	69	94	285	339
10	175	64	26	54	-1
11	331	342	87	-1	295
12	17	-1	302	352	148
13	86	88	-1	26	48
14	-1	139	191	108	91
15	337	-1	323	10	62
16	-1	137	22	298	100
17	238	212	-1	123	232
18	81	-1	245	139	146
19	-1	157	294	117	200
20	307	195	240	-1	135
21	-1	357	84	336	12
22	165	81	76	49	-1
23	-1	194	342	202	179
24	47	1	345	359	-1
25	76	159	174	342	-1
26	73	56	269	-1	232
27	150	72	329	224	-1
28	349	126	-1	106	21
29	139	277	214	-1	331
30	331	156	-1	273	313
31	118	32	-1	177	349
32	345	111	-1	245	34
33	27	175	-1	98	97
34	294	-1	218	355	187
35	-1	306	104	178	38
36	145	224	40	176	-1
37	279	-1	197	147	235
38	97	206	73	-1	52
39	106	-1	229	280	170

Table 101–3—LDPC (16200, 14400) code matrix *(continued)*

Columns	Rows				
	1	**2**	**3**	**4**	**5**
40	160	29	63	-1	58
41	143	106	-1	-1	-1
42	-1	334	270	-1	-1
43	-1	-1	72	221	-1
44	-1	-1	-1	208	257
45	-1	-1	-1	-1	0

Table 101–4 presents a 5 × 33 base matrix of the low-density parity-check matrix H for LDPC (5940, 5040) code listed in Table 101–2 for upstream. The lifting factor of the matrix is L=180.

Table 101–4—LDCP (5940, 5040) code matrix

Columns	Rows				
	1	**2**	**3**	**4**	**5**
1	142	54	63	28	52
2	158	172	11	160	159
3	113	145	112	102	75
4	124	28	114	44	74
5	92	55	61	8	46
6	44	19	123	84	71
7	93	159	72	126	42
8	70	22	55	9	11
9	172	96	114	169	108
10	3	12	20	174	153
11	25	85	53	147	-1
12	44	-1	114	24	72
13	141	128	42	145	-1
14	160	5	33	-1	163
15	50	158	4	26	-1
16	45	120	66	-1	9
17	118	51	163	-1	2
18	84	171	50	-1	168

Table 101–4—LDCP (5940, 5040) code matrix *(continued)*

Columns	Rows				
	1	**2**	**3**	**4**	**5**
19	-1	65	46	67	158
20	64	141	17	82	-1
21	66	-1	175	4	1
22	97	42	-1	177	49
23	1	83	-1	151	89
24	115	7	-1	131	63
25	8	-1	92	139	179
26	108	39	-1	117	10
27	-1	121	41	36	75
28	-1	84	138	18	161
29	11	101	-1	-1	-1
30	-1	171	34	-1	-1
31	-1	-1	74	23	-1
32	-1	-1	-1	8	177
33	-1	-1	-1	-1	19

Table 101–5 presents a 5 × 20 base matrix of the low-density parity-check matrix H for LDPC (1120, 840) code listed in Table 101–2 for upstream. The lifting factor of the matrix is L=56.

Table 101–5—LDCP (1120, 840) code matrix

Columns	Rows				
	1	**2**	**3**	**4**	**5**
1	5	0	12	0	36
2	14	35	28	51	6
3	12	1	22	16	3
4	1	26	46	31	51
5	2	0	3	13	4
6	37	10	16	39	19
7	45	16	51	27	4
8	26	16	2	33	45
9	24	34	25	8	48

Table 101–5—LDCP (1120, 840) code matrix *(continued)*

Columns	Rows				
	1	**2**	**3**	**4**	**5**
10	0	4	29	27	9
11	3	2	19	53	-1
12	-1	23	18	13	11
13	34	0	52	-1	22
14	7	51	-1	52	23
15	46	-1	37	33	43
16	10	49	-1	-1	-1
17	-1	20	34	-1	-1
18	-1	-1	39	38	-1
19	-1	-1	-1	7	14
20	-1	-1	-1	-1	1

101.3.2.3.2 Data Detector process within CLT (downstream)

The {EPoC_PMD_Name} CLT PCS transmit path includes the Data Detector process. This process contains a delay line (represented by the *FIFO_FEC_TX* buffer) that stores 66-bit blocks received from the output of the 64B/66B encoder to allow insertion of the FEC parity data into the transmitted data stream. The length of the *FIFO_FEC_TX* buffer is selected in such a way that it is large enough to compensate for the insertion of the FEC parity data as well as any additional FEC-related overhead, as defined in 101.3.2.3.3. For the Data Detector process included in the {EPoC_PMD_Name} CLT PCS transmit path, the length of the *FIFO_FEC_TX* buffer is set to be equal to the maximum amount of FEC parity data that may be inserted within the transmission time of one packet of a maximum length (i.e., at most forty 66-bit blocks of FEC parity data).

NOTE: the last statement in yellow above must be recalculated and revised. The value comes from 10G-EPON and it is likely incorrect.

101.3.2.3.3 LDPC ~~encoding~~Encode process within CLT (downstream)

The process of padding FEC codewords and appending FEC parity octets in the {EPoC_PMD_Name} CLT ~~transmitter~~PCS transmit path is illustrated in Figure 101–1.

The 64B/66B encoder produces a stream of 66-bit blocks, which are ~~then~~ delivered to the FEC Encode and Data Detector input process, as shown in Figure 101–2~~FEC encoder~~. The FEC encoder accumulates B_Q (see Table 101–1) of these 66-bit blocks to form the payload portion of ~~a~~the FEC codeword, removing the redundant first bit (i.e., sync header bit <0>) in each 66-bit block received from the 64B/66B encoder. The first bit <0> of the sync header in the 66-bit block in the transmit direction is guaranteed to be the complement of the second bit <1> of the sync header – see 49.2.4.3 for more details.

Next, the FEC encoder calculates CRC40 (see 101.3.2.3.8) over the aggregated B_Q 65-bit blocks, placing the resulting 40 bits of CRC40 code immediately after the B_Q 65-bit blocks, forming the payload portion of the

FEC codeword. Finally, the FEC encoder prepends B_P (see Table 101–1) padding bits (with the binary value of "0") to the payload portion of the FEC codeword as shown in Figure 101–1.

This resulting data is then LDPC-encoded, ~~resulting in~~producing ~~the~~ F_R bits of parity data. The first 25 bits of parity data are inserted into the 65-bit block carrying CRC40 code, complementing it. The remaining F_R-25 bits of parity data is then divided into C_Q 65-bit blocks. Note that 65-bit blocks carrying CRC40 data and parity data do not include sync header. The last 65-bit block of the parity data contains C_{PL} bits of parity data, and the remaining C_P bits are filled with padding (binary "0").

101.3.2.3.4 LDPC codeword transmission order within CLT (downstream)

Once the process of calculating FEC parity is complete, the payload portion of the FEC codeword and the parity portion of the FEC codeword are then transferred towards ~~the Data Detector~~PMA, one 65-bit block at a time. Note that the B_P padding bits used to generate the FEC codeword are not transmitted towards ~~the Data Detector~~PMA. The C_P padding bits in the last parity codeword (block number C_Q) are transmitted towards ~~the Data Detector~~PMA.

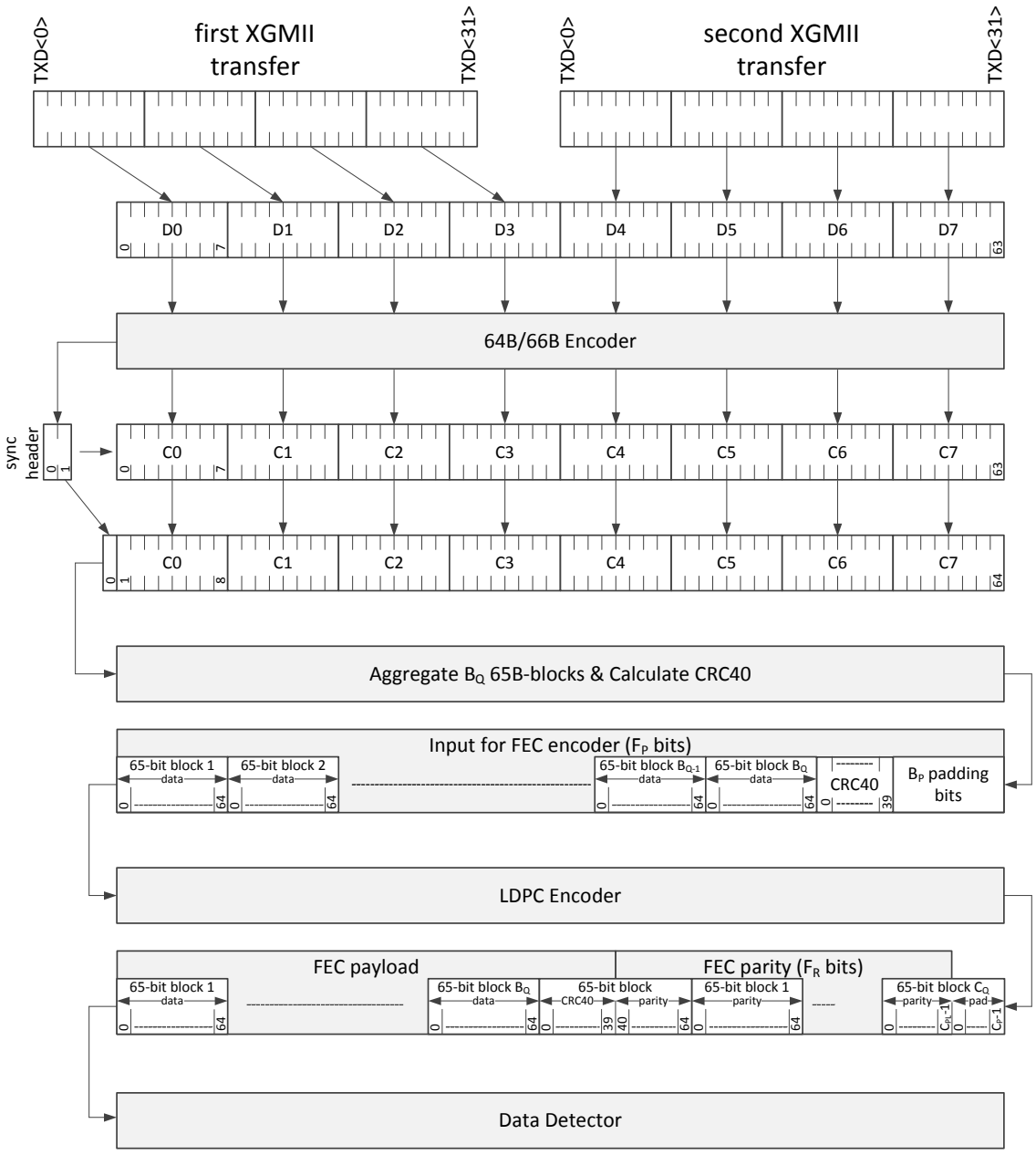

Figure 101–1—PCS Transmit bit ordering within CLT (downstream)

101.3.2.3.5 LDPC encoding process within CNU (upstream)

{the upstream FEC encoding for CNU will be described when we have a consistent proposal on how to mix three different FEC codes into a single transmission slot}

101.3.2.3.6 Data Detector process within CNU (upstream)

{Data Detector process in the upstream will be much more complex, since it needs to account for burs transmission as well and burst structure - text is TBD at this time}

101.3.2.3.7 LDPC codeword transmission order within CNU (upstream)

{the content of this subclause ought to be quite similar with the content of 101.3.2.3.5}

101.3.2.3.8 CRC40

{the content of this subclause will provide details about CRC40 used in EPoC to guarantee MTTFPA}

101.3.2.3.9 State diagrams

101.3.2.3.9.1 Constants

B_P

VALUE: see Table 101–1 for downstream FEC, Table 101–2 for upstream FEC
This constant represents the number of padding bits within the payload portion of the FEC codeword.

B_Q

VALUE: see Table 101–1 for downstream FEC, Table 101–2 for upstream FEC
This constant represents the number of 65-bit blocks within the payload portion of the FEC codeword.

C_P

VALUE: see Table 101–1 for downstream FEC, Table 101–2 for upstream FEC
This constant represents the number of padding bits within the last 65-bit block of the parity portion of the FEC codeword.

C_Q

VALUE: see Table 101–1 for downstream FEC, Table 101–2 for upstream FEC
This constant represents the number of 65-bit blocks within the parity portion of the FEC codeword.

F_P

VALUE: see Table 101–1 for downstream FEC, Table 101–2 for upstream FEC
This constant represents the number of bits within the payload portion of the FEC codeword.

F_R

VALUE: see Table 101–1 for downstream FEC, Table 101–2 for upstream FEC
This constant represents the number of bits within the parity portion of the FEC codeword.

SH_CTRL

See 76.3.2.5.2

SH_DATA

See 76.3.2.5.2

101.3.2.3.9.2 Variables

blockCount

TYPE: 16-bit unsigned integer
This variable represents the number of either 65-bit blocks or 66-bit blocks.

CLK

TYPE: Boolean
This Boolean is *true* on every negative edge of TX_CLK (see 46.3.1) and represents instances of time at which a 66-bit block is passed from the output of the 64B/66B encoder into the FEC encoder. This variable is reset to *false* upon read.

dataPayload<F_P-1:0>

TYPE: Bit array

This array represents the payload portion of the FEC codeword, accounting for the necessary padding. It is initialized to the size of F_P bits and filled with the binary value of "0".

dataParity<F_R-1+C_P:0>

TYPE: Bit array

This array represents the parity portion of the FEC codeword, accounting for the necessary padding. It is initialized to the size of $F_R + C_P$ bits and filled with the binary value of "0".

FIFO_FEC_TX

TYPE: Array of 65-bit blocks

A FIFO array used to store 65-bit blocks, inserted by the input process and retrieved by the output process in the FEC encoder.

loc

TYPE: 16-bit unsigned integer

This variable represents the position within the given bit array.

~~SH_CTRL~~

~~See 76.3.2.5.2~~

~~SH_DATA~~

~~See 76.3.2.5.2~~

sizeFifo

TYPE: 16-bit unsigned integer

This variable represents the number of 65-bit blocks stored in the FIFO.

tx_coded<65:0>

TYPE: 66-bit block

This 66-bit block contains 64B/66B encoded data from the output of 64B/66B encoder. The format for this data block is shown in Figure 49–7. The left-most bit in the figure is tx_coded<0> and the right-most bit is tx_coded<65>.

tx_coded_out<64:0>

TYPE: 65-bit block

This 65-bit block contains the output of the FEC encoder being passed ~~towards~~to the Data Detector. The left-most bit is tx_coded_out<0> and the right-most bit is tx_coded_out<64>.

101.3.2.3.9.3 Functions

calculateCrc (ARRAY_IN)

This function calculates CRC40 for data included in ARRAY_IN.

calculateParity(ARRAY_IN)

This function calculates LDPC parity (for the code per Table 101–1 or Table 101–2) for data included in ARRAY_IN.

resetArray(ARRAY_IN)

This function resets the content of ARRAY_IN, removing all the elements within ARRAY_IN and setting its size to 0.

removeFifoHead(ARRAY_IN)

This function removes the first block in ARRAY_IN and decrements its size by 1.

```
removeFifoHead( ARRAY_IN )
{
ARRAY_IN[0] = ARRAY_IN[1]
ARRAY_IN[1] = ARRAY_IN[2]
```

```
...
ARRAY_IN[sizeFifo-2] = ARRAY_IN[sizeFifo-1]
sizeFifo --
}
```

101.3.2.3.9.4 Messages

TBD

101.3.2.3.9.5 State diagrams

The CLT PCS shall implement the ~~LDPC~~FEC-Encode and Data Detector ~~encoding~~ process, comprising the input process as shown in Figure 101–2 and the output process as shown in Figure 101–3. The CNU PCS shall implement the ~~LDPC encoding~~FEC Encode and Data Detector process, comprising the input process as shown in Figure 101–2 and the output process as shown in Figure 101–3.

In case of any discrepancy between state diagrams and the descriptive text, the state diagrams prevail.

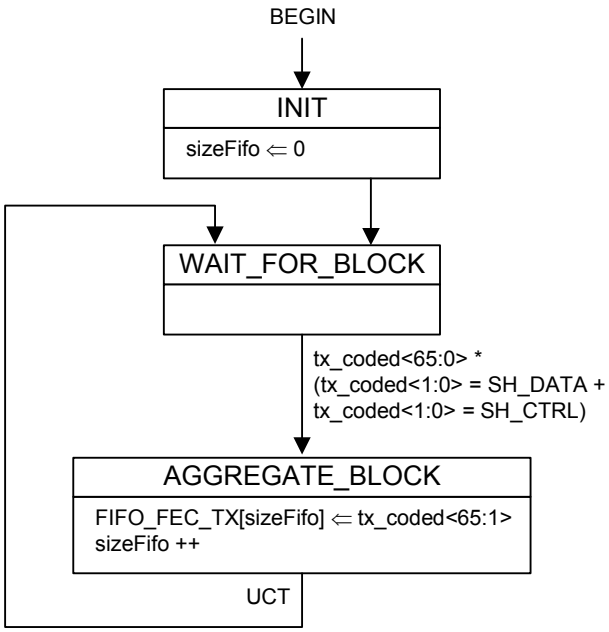

Figure 101–2—FEC ~~encoder~~Encode and Data Detector, input process state diagram

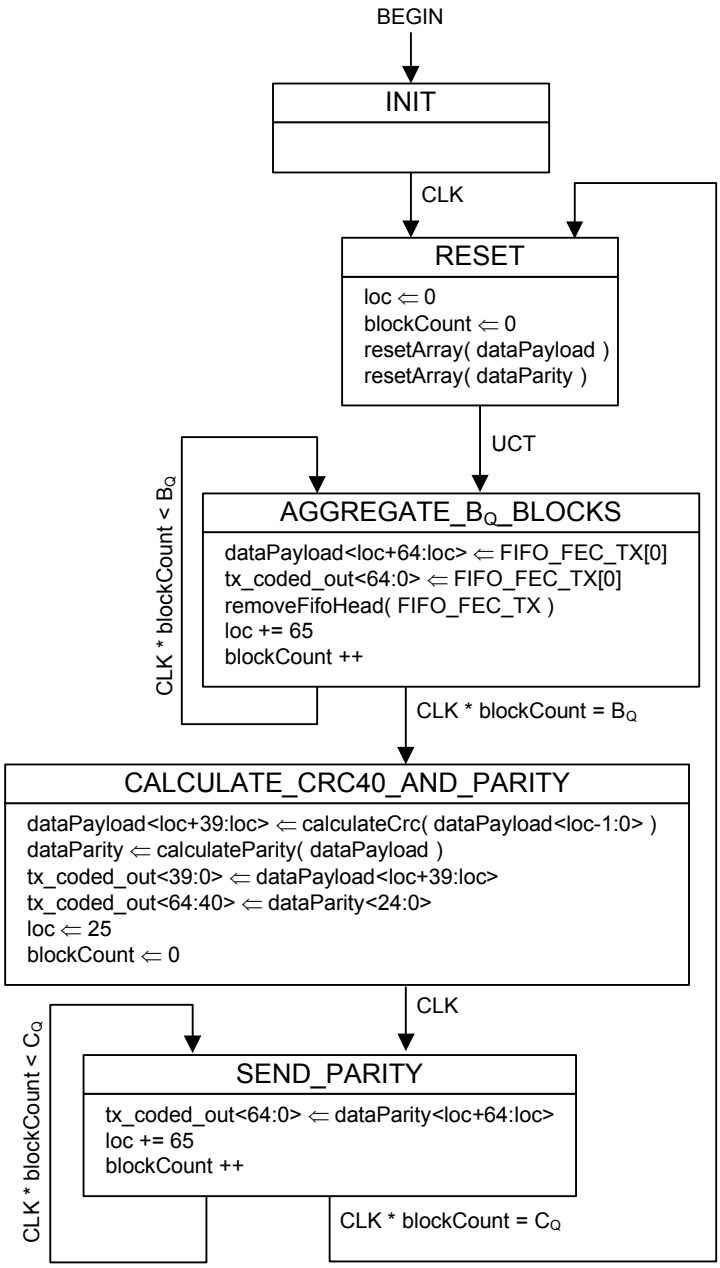

Figure 101–3—FEC ~~encoder~~Encode and Data Detector, output process state diagram (CL1