142. Physical Coding Sublayer and Physical Media Attachment for 100G-EPON

142.1 Overview

This clause describes the Physical Coding Sublayer (PCS) with FEC and Physical Medium Attachment (PMA) used with {NG-EPON type} point-to-multipoint (P2MP) networks. These are passive optical multipoint networks (PONs) that connect multiple DTEs using a single shared fiber. The architecture is asymmetric, based on a tree and branch topology utilizing passive optical splitters. This type of network requires that the Multipoint MAC Control sublayer exists above the MACs, as described in Clause 144.

142.1.1 Conventions

The notation used in the state diagrams in this clause follows the conventions in 21.5. Should there be a discrepancy between a state diagram and descriptive text, the state diagram prevails. The notation ++ after a counter indicates it is to be incremented by 1. The notation −− after a counter indicates it is to be decremented by 1. The notation −= after a counter indicates that the counter value is to be decremented by the following value. The notation += after a counter indicates that the counter value is to be incremented by the following value. Code examples given in this clause adhere to the style of the “C” programming language.

142.1.2 Delay constraints

{TBD}

142.2 Physical Coding Sublayer (PCS) for 100G-EPON

142.2.1 Overview

This subclause defines the physical coding sublayers {NG-EPON type} supporting burst mode operation over the point-to-multipoint physical medium. The {NG-EPON type, symmetric} PCS is specified to support {NG-EPON types}, where both the receive and transmit paths operate at multiples of 25.78125 Gb/s rate. The {NG-EPON type, asymmetric} PCS supports {NG-EPON types}, in which OLT transmit path and ONU receive path operate at 25.78125 Gb/s, while the ONU transmit path and the OLT receive path operate at 10.3125 Gb/s rate. Figure XXX and Figure XXX show the relationship between the PCS sublayer and the ISO/IEC OSI reference model.

The PCS functional block diagram is shown in 0.

This subclause also specifies a forward error correction (FEC) mechanism to increase the optical link budget or the fiber distance.

Figure 142–1—PCS Functional Block Diagram

142.2.1.1 {NG-EPON, asymmetric} PCS

{TBD}

142.2.1.2 {NG-EPON, symmetric} PCS

{TBD}

142.2.2 PCS transmit function

This subclause defines the transmit direction of the physical coding sublayers for {NG-EPON type}. In the OLT, the PCS transmit function operates at a 25.78125 Gb/s rate in a continuous mode. In the ONU, the PCS transmit function may operate at a 25.78125 Gb/s rate, as specified herein ({NG-EPON type, symmetric}), or at a {TBD} Gb/s rate, as specified in {TBD} ({NG-EPON type, asymmetric}). For all {NG-EPON type}, the ONU PCS operates in a burst mode in the transmit direction. The PCS includes a mandatory LDPC FEC encoder. The functional block diagram for the PCS transmit function is shown in 0. The transmit function consists of the following functional blocks.

— Transmit/Encode block (see 142.2.2.1),
— Data Detect block (ONU only, see 142.2.2.2),
— 64B/66B to 256B/257B Transcoder (see 142.2.2.3),
— Scrambler (see 142.2.2.4),
— FEC Encoder (see 142.2.2.5), and
— Gear Box (see 142.2.2.6).

142.2.2.1 Transmit/Encode

The Transmit/Encode functional block accepts data from the one 25GMII interface and converts two consecutive 36-bit transfers into a single 72-bit tx_raw vector which is then encoded into a single 64B/66B block. The 64B/66B block structure is as defined in 49.2.4 with exceptions as noted in this subclause. The state diagram of the Transmit/Encode block is shown in Figure 142–2.

142.2.2.1.1 Block Structure

The 25BGASE-PR PCS supports all the block type fields in Figure 49-7 except block type field values of: 0x2d, 0x33, 0x66, 0x55, and 0x4b.

142.2.2.1.2 Control codes

The {NG EPON type} PCS supports the control codes shown in Table 142–1. The representations of the control characters are the control codes. Control characters are transferred over the 25GMII as an 8-bit value. The 25GBASE-PR PCS encodes the start and terminate control characters implicitly using the block type field. The 25GBASE-PR PCS does not encode the ordered set control codes. The 10GBASE-R PCS encodes each of the other control characters into a 7-bit C code.

The control characters and their mappings to 25GBASE-PR control codes are specified in Table 142–1. All 25GMII and 25GBASE-PR control code values that do not appear in the table shall not be transmitted and are treated as an error if received.

Table 142–1—Control Codes

Control Character	Notation	25GMII control code	25GBASE-PR control code
Idle	/I/	0x07	0x00
Inter Envelope Idle	/IEI/	0x08	0x08
Parity Placeholder	/P/	0x09	0x09
Start	/S/	0xFB	Encoded by block type field

Table 142–1—Control Codes

Control Character	Notation	25GMII control code	25GBASE-PR control code
Terminate	/T/	0xFD	Encoded by block type field
Error	/E/	0xFE	0x1E

142.2.2.1.3 Constants

EBLOCK_T - see 49.2.13.2.1.

LBLOCK_T - see 49.2.13.2.1.

142.2.2.1.4 Variables

tx_coded – see 49.2.13.2.2.

tx_raw – see 49.2.13.2.2.

142.2.2.1.5 Functions

ENCODE(tx_raw) – see 49.2.13.2.3.

NextTxValid(prev_tx_coded, next_tx_raw)

This function returns a Boolean indicating whether the *next_tx_raw* vector is valid given the classification of the current (*next_tx_raw*) and previously transmitted (*prev_tx_coded*) vectors. The function returns the values according to Table 142–2. Vector classifications used in Table 142–2 are shown in Table 142–3.

Table 142–2—*NextTxValid* and *NextRxValid* function output

		next_tx_raw/next_rx_coded vector classification						
		/IEI/	/S/	/D/	/T/	/I/	/P/	Other
prev_tx_coded/ prev_rx_raw vector classification	/L/	true	false	false	false	false	false	false
	/IEI/	true	true	false	false	false	true	false
	/S/	true	true	true	true	true	true	false
	/D/	true	true	true	true	true	true	false
	/T/	true	true	true	false	true	true	false
	/I/	true	true	true	false	true	true	false
	/P/	true	true	true	true	true	true	false
	Other	true	true	true	true	true	true	false

Table 142–3—Vector classifications

		Criteria for tx_raw/rx_raw vector	Criteria for tx_coded/rx_coded vector
classification	**/L/**	rx_raw<71:0> = LBLOCK_R (see 49.2.13.2.1). This classification does not apply to tx_raw<71:0>.	tx_coded<65:0> = LBLOCK_T (see 49.2.13.2.1). This classification does not apply to rx_coded<65:0>.
	/IEI/	Vector composed of INTER_ENV_IDLE (see 143.4.3.2)	Vector composed of Inter-envelope idle (vector<1:0> = 10, vector<9:2> = 0x1E, and all control codes = 0x08).
	/S/	Vector beginning with a Start control code symbol (vector<7:0> = 0x80, vector<15:8> = 0xFB)	Vector comprised of a Start control code symbol (vector<1:0> = 10 and vector<9:2> = 0x78)
	/D/	Vector of all data bytes (vector<7:0> = 0x00)	Vector of all data bytes (vector<1:0> = 01)
	/T/	Vector which includes a Terminate control code symbol (vector<7:0> ∈ {0xFF, 0x7F, 0x3F, 0x1F, 0x0F, 0x07, 0x03, 0x01}, 1st control code octet = 0xFD, and all other control characters are valid)	Vector which includes a Terminate control code symbol (vector<1:0> = 10 and vector<9:2> ∈ {0x87, 0x99, 0xAA, 0xB4, 0xCC, 0xD2, 0xE1, 0xFF}, and all control characters are valid)
	/I/	Vector composed of all Idle control code symbols (vector<7:0> = 0xFF and all other octets = 0x07)	Vector composed of all Idle control code symbols vector<1:0> = 10 and vector<65:2> = 0x00..00)
	/P/	Vector composed of PARITY_PLACEHLDR (see 143.4.3.2)	Vector composed of all Parity placeholder (vector<1:0> = 10, vector<9:2> = 0x1E, and all control codes = 0x09)
	/E/	rx_raw<71:0> = EBLOCK_R (see 49.2.13.2.1). This classification does not apply to tx_raw<71:0>.	tx_coded<65:0> = EBLOCK_T (see 49.2.13.2.1). This classification does not apply to rx_coded<65:0>.

NextTxVector()
This function returns the next 72-bit vector from the 25GMII.

142.2.2.1.6 State Diagrams

The OLT and the ONU shall implement the Transmit/Encode process as depicted in Figure 142- 2.

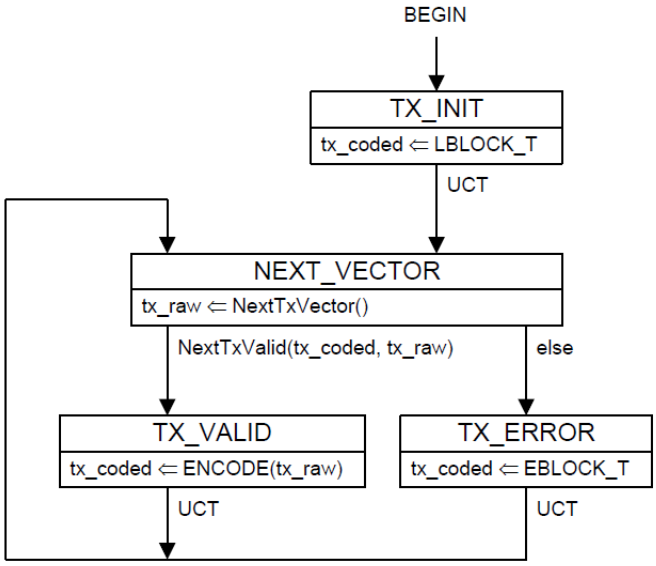

Figure 142–2—Transmit/Encode State Diagram

142.2.2.2 Data detector

{TBD}

142.2.2.2.1 Burst Mode operation (ONU only)

{TBD}

Figure 142–3 presents the details of the ONU burst transmission, in particular, details of the FEC-unprotected and the FEC-protected portions of the normal (granting) upstream burst with three distinct Sync Pattern zones.

The upstream burst begins with a FEC-unprotected area (Sync Pattern), comprising several explicit zones, each playing a separate role: SP_1 zone, optimized for laser on (T_{on}), Automatic Gain Control (AGC, $T_{settling}$); SP_2 zone, optimized for Clock Data Recovery (CDR, T_{CDR}); and SP_3 zone, optimized for Start of Burst Delimiter (SBD) pattern. This arrangement is shown in Figure 142–3 for the normal (granting operation) and in Figure 142–5 for the discovery operation. In some implementations, only two explicit zones are needed: SP_1 zone, optimized for laser on (T_{on}), Automatic Gain Control (AGC, $T_{settling}$), and Clock Data Recovery (CDR, T_{CDR}); and SP_2 zone, optimized for Start of Burst Delimiter (SBD) pattern. This arrangement is shown in Figure 142–4 for the normal (granting operation) and in Figure 142–6 for the discovery operation. Bit patterns transmitted within each Sync Pattern zone are configured using the SYNC_PATTERN MPCPDU (see <new subclause with MPCPDU definition>). Each of the Sync Pattern elements is a multiple of 257 bits, aligning with the PCS line code of 256B/257B.

The SBD is followed by a number of FEC codewords, where the last codeword may be shortened to minimize the upstream transmission overhead from long LDPC codewords used. Each FEC codeword comprises a series of 256B/257B encoded and scrambled data blocks, followed by a series of 257-bit long parity blocks. Within a non-shortened FEC codeword, the FEC payload portion includes 56 257-bit data blocks and 10 257-bit parity blocks. Within a shortened FEC codeword, the FEC payload portion may be truncated to a number of data blocks smaller than 56, while the size of the FEC parity portion remains unchanged.

The upstream burst ends with a End of Burst Delimiter (EBD). When received at the OLT, the EBD pattern allows for the rapid reset of the OLT FEC synchronizer, preparing the OLT for the next incoming upstream burst. The EBD pattern is not part of the last FEC codeword.

The OLT announces the number of the Sync Pattern zones (two or three) using the SYNC_PATTERN MPCPDU. The decision to use two or three Sync Pattern zones is implementation-dependent and related with the design of the OLT burst-mode receiver.

The default number of Sync Pattern zones shall be 2, i.e., the *SpCount* field in the SYNC_PATTERN MPCPDU is set to the value of two (2).

The default configuration SP_1 Sync Pattern zone shall be defined as follows:

— covers T_{on}, $T_{setting}$, and T_{CDR};
— *SpValue*: 0x1-55-55-55-55-55-55-55-55-55-55-55-55-55-55-55-55-55-55-55-55-55-55-55-55-55-55-55-55-55-55-55 -55-55-55-55-55-55-55-55-55-55
— *SpBalanced*: True (1)

The default configuration SP_2 Sync Pattern zone shall be defined as follows:

— covers SBD;
— *SpValue*: 0x1-BF-40-18-E5-C5-49-BB-59-6B-F8-D8-12-D8-58-E4-AB-40-BF-E7-1A-3A-B6-44-A6-94-07-27-ED-27-A7-1B-54
— *SpBalanced*: False (0)

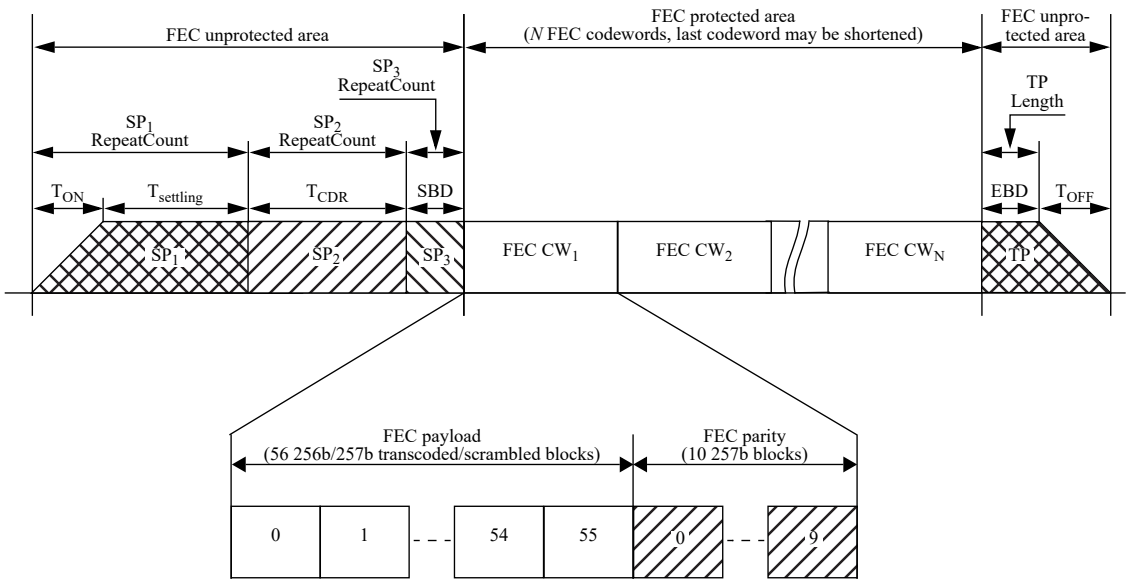

Figure 142–3—ONU burst structure, normal grant, 3 zones

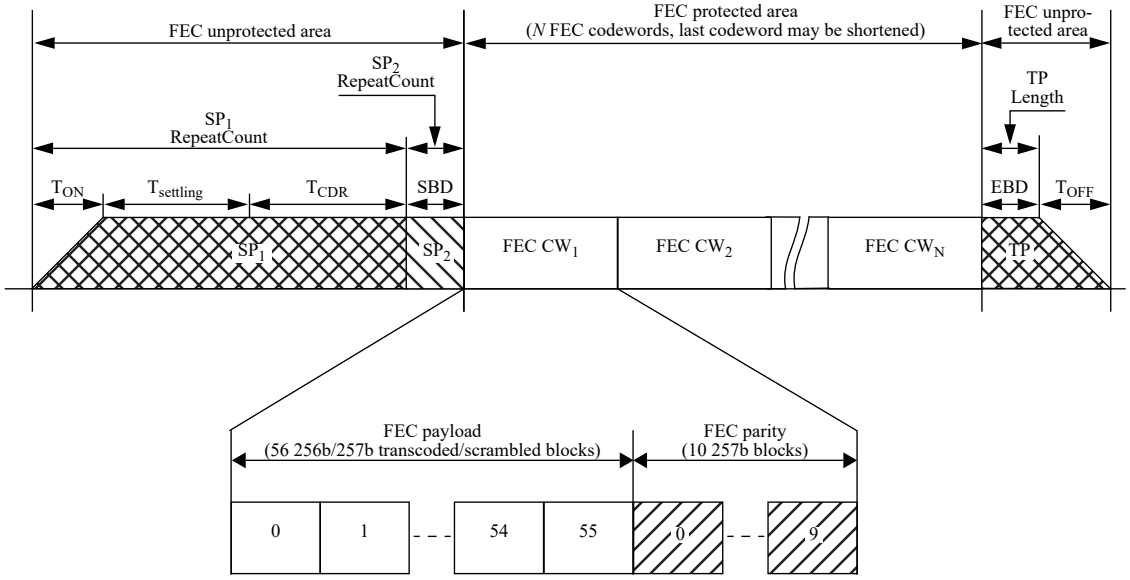

Figure 142–4—ONU burst structure, normal grant, 2 zones

142.2.2.3 64B/66B to 256B/257B transcoder

The 64B/66B to 256B/257B transcoder converts four consecutive 64B/66B blocks the into one 256B/257B block as described in 91.5.2.5 and passes the resulting 257-bit-wide block to the Scrambler functional block. In the OLT the 64B/66B blocks are received from Transmitter/Encoder functional block whereas in the ONU the 64B/66B blocks are received from the Data Detector.

142.2.2.4 Scrambler

See 49.2.6.

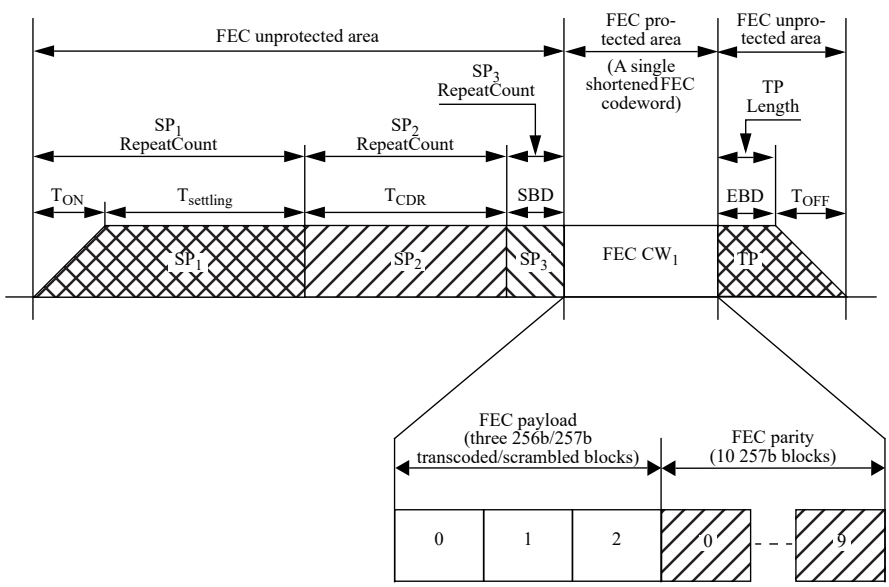

Figure 142–5—ONU burst structure, discovery grant, 3 zones

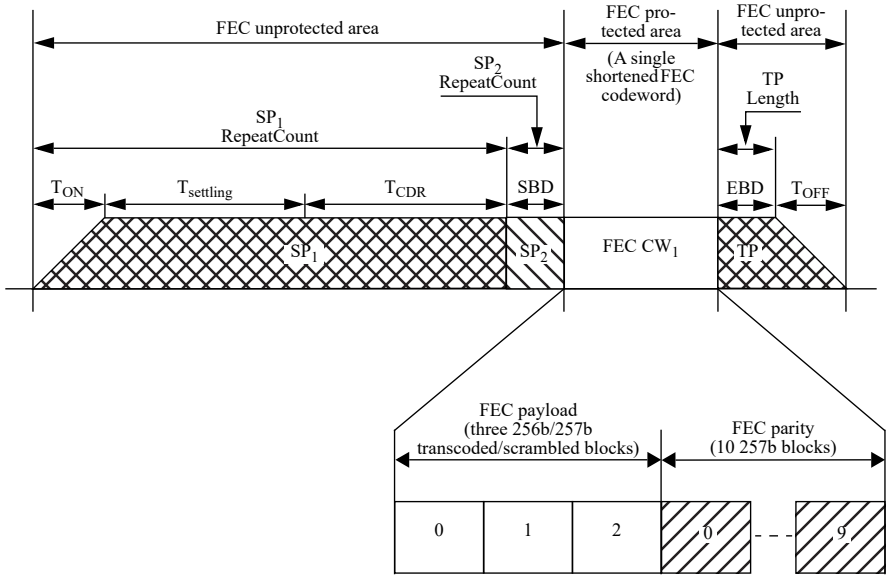

Figure 142–6—ONU burst structure, discovery grant, 2 zones

142.2.2.5 FEC encoding

The {NG-EPON type} PCS shall encode the transmitted data stream using {TBD} FEC. Annex {TBD} gives an example of {TBD} FEC Encoding.

142.2.2.5.1 Low Density Parity Check Coding

The bit sequence input for a given code block to channel coding is denoted by u_1, u_2, ..., u_K, where K is the number of bits to be encoded. The parity check bit sequence produced by FEC Encoder is denoted by p_1, p_2,

..., p_M, where M is the number of parity check bits. The output of FEC Encoder is denoted by $c = [c_1, c_2, ..., c_N] = [u_1, u2, ..., u_K \mid p_1, p_2, ..., p_M]$, where N = K + M is length of encoder output sequence.

The FEC encoding scheme is shown in Figure 142–7. The scheme consists of a systematic QC–LDPC encoder and a shortening and puncturing mechanism. The parameters of the FEC encoding scheme are:

— the LDPC parity check matrix is a 12–by–69 quasi-cyclic matrix, with circulant size Z = 256; LDPC user bit length before shortening is 57 × 256 = 14592, the parity bit length before puncturing is 12 × 256 = 3072; the codeword length before any shortening and puncturing is 17664;
— the number of transmitted information bits, K (with maximum user length K_{max} = 14328);
— the number of shortened information bits, S (S_{min} = 264);
— the number of punctured parity check bits, P (P = 512);
— the number of parity-check bits after puncturing, M (M = 3072 - 512 = 2560);
— the number of output bits, N (N = K + M, FEC codeword, whose size depends on the burst length pattern to determine shortening length); $N_{max} = K_{max} + M$ = 16888;
— the code rate, R = K / N, defined as the code rate after puncturing and after shortening.

The encoder supports highest code rate $R_{max} = K_{max} / N_{max}$ = 0.8484. Codes with lower code rates/shorter block length shall be obtained through shortening. The puncturing length and location are fixed for all scenarios.

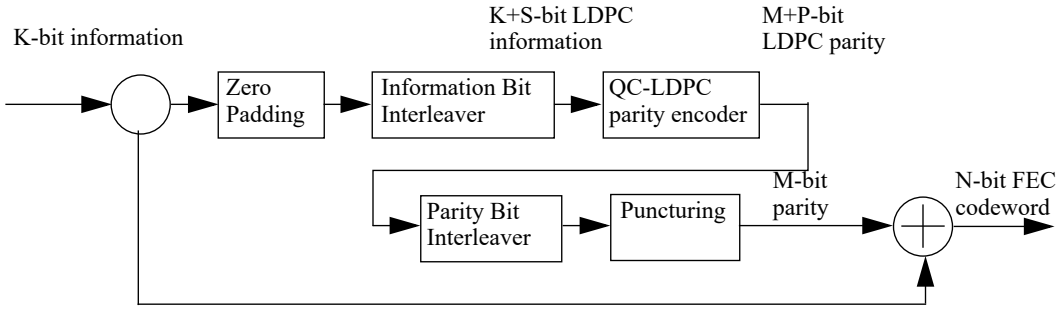

Figure 142–7—FEC encoder

142.2.2.5.2 LDPC Encoder

The full LDPC code is defined by a (M + P) × (K + S + M + P) = 3072 × 17664 size parity-check matrix H composed by a 12 × 69 array of 256 × 256 sub-matrices $A_{i,j}$:

$$H = \begin{bmatrix} A_{1,1} & \cdots & A_{1,69} \\ \cdots & & \cdots \\ A_{12,1} & \cdots & A_{12,69} \end{bmatrix}$$

The sub-matrices $A_{i,j}$ are either a cyclic shifted version of identity matrix or a zero matrix, and have a size of 256 × 256. The parity-check matrix can be described in its compact form:

$$H_C = \begin{bmatrix} a_{1,1} & \cdots & a_{1,69} \\ \cdots & & \cdots \\ a_{12,1} & \cdots & a_{12,69} \end{bmatrix}$$

where $a_{i,j} = -1$ for a zero sub-matrix in position (i,j), and a positive integer number $a_{i,j}$ defines the number of right column shifts of the identity matrix.

Editor's Note (to be removed prior to publication): Before entering WG ballot, content of Table 142-1 will be published under http://standards.ieee.org/downloads/802.3/ in a machine readable format

The compact form of parity-check matrix H_c is shown in Table 142–1.

Table 142–1—Compact form of parity-check matrix H_c

C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12
80	-1	-1	105	-1	-1	137	-1	–1	0	209	53
–1	0	91	–1	170	46	-1	118	208	-1	–1	-1
–1	–1	–1	-1	250	–1	204	15	0	–1	252	93
60	0	74	87	01	37	–1	–1	–1	123	–1	–1
169	-1	-1	-1	–1	–1	238	93	0	–1	39	216
-1	0	237	43	195	49	–1	–1	–1	41	–1	–1
11	-1	202	-1	139	150	–1	–1	0	191	–1	–1
-1	0	-1	165	–1	–1	228	228	–1	–1	159	57
143	-1	-1	-1	-1	65	-1	-1	0	211	69	9
-1	0	201	180	135	-1	225	78	-1	-1	-1	-1
-1	-1	136	-1	-1	-1	247	-1	0	217	37	130
222	0	-1	80	92	177	-1	16	-1	-1	-1	-1
-1	-1	178	227	-1	144	-1	0	-1	243	134	-1
59	0	-1	-1	147	-1	191	-1	251	-1	-1	130
-1	-1	239	221	-1	70	-1	48	0	97	-1	-1
218	0	-1	-1	1	-1	177	-1	-1	-1	201	238
-1	-1	183	77	-1	95	-1	0	-1	252	49	-1
-1	0	-1	-1	-1	-1	255	-1	44	-1	-1	-1
178	0	-1	-1	-1	-1	-1	-1	123	-1	-1	-1
-1	-1	217	0	-1	221	-1	-1	-1	-1	-1	-1
-1	0	-1	-1	13	-1	-1	62	-1	-1	-1	-1

Table 142–1—Compact form of parity-check matrix *(continued)*H_c

C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12
-1	-1	232	-1	-1	-1	-1	-1	-1	0	104	-1
-1	-1	-1	-1	-1	-1	192	0	-1	-1	-1	144
-1	-1	-1	-1	98	192	-1	-1	0	-1	-1	-1
105	0	-1	16	-1	-1	-1	-1	-1	-1	-1	-1
-1	-1	169	-1	-1	128	-1	0	-1	-1	-1	-1
-1	-1	-1	-1	142	-1	-1	-1	0	-1	129	-1
19	0	-1	-1	-1	-1	51	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	214	-1	-1	-1	0	-1	162
-1	-1	-1	252	-1	-1	-1	-1	-1	-1	157	0
126	-1	-1	-1	225	-1	-1	0	-1	-1	-1	-1
-1	-1	-1	96	-1	-1	-1	-1	0	41	-1	-1
-1	0	129	-1	-1	-1	195	-1	-1	-1	-1	-1
-1	-1	60	0	-1	-1	-1	-1	-1	-1	222	-1
211	-1	-1	-1	-1	51	0	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1	-1	0	29	-1	175
-1	0	-1	-1	23	-1	-1	112	-1	-1	-1	-1
-1	-1	-1	-1	108	-1	172	-1	-1	0	-1	-1
-1	-1	-1	17	-1	100	-1	0	-1	-1	-1	-1
-1	0	19	-1	-1	-1	-1	-1	-1	-1	-1	145
247	-1	76	-1	-1	-1	-1	-1	0	-1	-1	-1
-1	-1	-1	-1	-1	19	-1	-1	-1	-1	139	0
255	-1	-1	-1	-1	-1	-1	-1	-1	0	39	-1
-1	0	-1	-1	-1	-1	219	-1	153	-1	-1	-1
-1	-1	-1	219	0	235	-1	-1	-1	-1	-1	-1
85	-1	-1	-1	-1	-1	-1	0	-1	-1	-1	36
-1	-1	77	-1	0	-1	236	-1	-1	-1	-1	-1
-1	0	-1	198	-1	-1	-1	-1	-1	193	-1	-1
-1	-1	-1	165	-1	-1	-1	-1	0	-1	203	-1
-1	-1	-1	-1	-1	-1	136	0	-1	145	-1	-1
-1	-1	2	-1	-1	-1	-1	0	-1	-1	94	-1
-1	-1	-1	-1	235	-1	-1	-1	0	-1	-1	91
246	0	-1	-1	-1	4	-1	-1	-1	-1	-1	-1
94	-1	-1	36	-1	-1	0	-1	-1	-1	-1	-1

Table 142–1—Compact form of parity-check matrix *(continued)*H_c

C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12
-1	-1	101	-1	-1	-1	-1	-1	-1	0	-1	22
-1	-1	-1	-1	-1	251	-1	22	0	-1	-1	-1
-1	0	-1	-1	121	-1	-1	-1	-1	-1	194	-1
-1	-1	217	-1	0	-1	259	-1	-1	-1	-1	-1
-1	-1	-1	171	-1	109	-1	-1	-1	-1	-1	0
242	-1	-1	-1	-1	-1	-1	-1	-1	-1	3	0
-1	0	-1	-1	-1	-1	10	-1	-1	-1	-1	212
-1	-1	48	-1	-1	-1	-1	0	-1	140	-1	-1
-1	-1	-1	-1	-1	-1	-1	0	-1	46	43	-1
-1	-1	-1	228	0	-1	-1	-1	-1	-1	153	-1
129	-1	-1	-1	-1	140	-1	-1	-1	-1	-1	0
-1	-1	-1	-1	-1	-1	5	-1	0	58	-1	-1
19	-1	-1	-1	46	-1	-1	-1	0	-1	-1	-1
58	0	172	39	242	193	25	120	16	202	207	69
27	-1	42	234	228	241	94	192	0	215	109	88

A fixed amount (512 bits) and locations of the parities are punctured on the full LDPC matrix; a minimum amount (264 bits) and locations of the user bits are shortened on the full LDPC matrix. The effective maximum code rate 0.8484.

Figure 142–8—Codeword Information/Parity Location assignments

142.2.2.5.3 Encoding Operation

The encoding process shall be as follows:

— A group of K information bits $u = [u_1, u_2, ..., u_K]$ are collected and copied to the output of the encoder to form a block of systematic code bits. They are also the input to the zero-padding block (see Figure 142–7).

— A total of S zero bits are appended at the end of u to form the full-length information bit vector $u^* = [u \mid 0, ..., 0]$, which is then sent to the information bit interleaver module, which in turn produces the bit-interleaved sequence $u'' = \pi_{info}(u^*)$.

— The interleaved LDPC information bits u" is sent to the QC-LDPC parity encoder, and used to compute parity-check bits p" with the parity-check matrix H, which is then interleaved to get $p^* = \pi_{info}(p'')$.

— M + P parity bits p* = [p_1, p_2, ..., p_M | p_{M+1}, ..., p_{M+P}] are sent to the puncturing block.
— The last P bits of p* are truncated, and M parity bits p = [p_1, p_2, ..., p_M] are being copied to the output of the encoder to form the parity check bits.
— At the encoder output c = [u | p] = [u_1, u_2, ..., u_K | p_1, p_2, ..., p_M], such that [u" | p"] H^T = 0.

142.2.2.6 Gearbox

{TBD}

142.2.3 PCS receive Function

This subclause defines the receive direction of physical coding sublayers for {NG-EPON type}. In the ONU, the PCS receive function operates at a 25.78125 Gb/s rate in a continuous mode. In the OLT, the PCS receive function may operate at a 25.78125 Gb/s rate, as specified herein ({NG-EPON type, symmetric}), or at a 10.3125 Gb/s rate, compliant with Clause {TBD} ({NG-EPON type, asymmetric}). For all {NG-EPON types}, the OLT PCS receive function operates in burst mode. The PCS includes a mandatory FEC decoder. The functional block diagram for the PCS receive function is shown in 0. The receive function consists of the following functional blocks:

— Synchronizer block (see 142.2.3.1 and 142.2.3.2),
— FEC Decoder (see 142.2.3.4),
— Descrambler (see 142.2.3.5),
— 256B/257B to 64B/66B Transcoder (see 142.2.3.6), and
— Receiver/Decode block (see 142.2.3.7).

142.2.3.1 OLT synchronizer

{TBD}

142.2.3.2 ONU Synchronizer

The ONU synchronization receives data via the {TBD}-bit PMA_UNITDATA.indication primitive. The synchronizer forms a bit stream from the primitives by concatenating requests with the bits of each primitive in order from rx_data-group<xx> to rx_data-group<xx> (see Figure was 76–19). It obtains lock to the FEC codewords within the bit stream using the mechanism shown in Figure 142- 3 and outputs {TBD} codewords to the FEC decoder function.

{TBD description of block handling}

While in codeword lock, the synchronizer copies the FEC-protected bits from each data block and the parity bits of the codeword into an input buffer. When the codeword is complete, the FEC decoder is triggered, and the input buffer is freed for the next codeword.

When in codeword lock, the state diagram continues to check for sync header validity. If 16 or more sync headers in a codeword pair (62 blocks) are invalid, then the state diagram deasserts codeword lock. In addition, if the *persist_dec_fail* signal becomes set, then codeword lock is deasserted (this check ensures that certain false-lock cases are not persistent.)

142.2.3.2.1 Constants

FEC_CW_SZ
TYPE: Integer

The size of the FEC Codeword in bits.
VALUE: {TBD}

Editorial Note (to be removed prior to publication): Note FEC_CW_SZ will likely be defined before this section and could just be cross referenced.

FecFailLimit
TYPE: Integer
The number of FEC decoding failures allowed while in codeword lock before declaring out of lock
VALUE: {TBD}

MatchTarget
TYPE: Integer
The number of parity delimiters required to transition from a codeword out of lock start to a codeword lock state.
VALUE: {TBD}

PD
TYPE: binary array of {TBD}-bits
The burst delimiter bit pattern found at the beginning of each FEC Parity block.
VALUE: {TBD}

142.2.3.2.2 Variables and counters

FecDecodeFail
TYPE: Boolean
This clear on read variable indicates the most recent completed FEC codeword decoding failed.

FecDecodeSucceed
TYPE: Boolean
This clear on read variable indicate the most recently completed FEC codeword decoding succeeded.

FecFailCount
TYPE: Integer
This counter track the number of consecutive FEC decoding failures.

Match
TYPE: Boolean
This variable holds the most recent result of the Compare() function.

MatchCount
TYPE: Integer
This counter tracks the number of consecutive successful parity delimiter matched.

rx_buffer
TYPE: binary array
This array hold the sequence of concatenated bits received from the PMA_UNITDATA.indication-primitive.

142.2.3.2.3 Functions

Compare(v, p)
This function compares bit by bit its two arguments and returns a Boolean ‘true’ if the number of

bits that are different is less or equal to the Hamming threshold of {TBD} otherwise the function returns false.

Slip(v, bc)
This function removes “bc” bits from the passed array “v”.

142.2.3.2.4 State Diagrams

The ONU Synchronizer shall implement the state diagram as depicted in Figure 76–20.

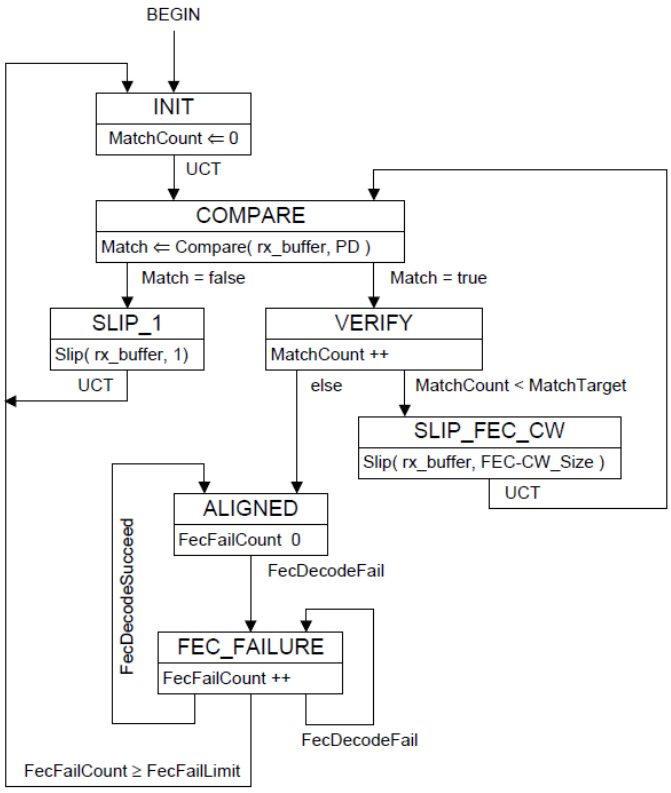

Figure 142–9—Transmit/Encode State Diagram

142.2.3.3 BER monitor

{TBD}

142.2.3.4 FEC decoder

{TBD}

142.2.3.5 Descrambler

See 49.2.10.

142.2.3.6 256B/257B to 64B/66B transcoder

The 256B/257B to 64B/66B transcoder converts one 256B/257B block received from the Descrambler functional block into four consecutive 64B/66B blocks as described in 91.5.3.5 and passes these to the Receiver/Decoder functional block.

142.2.3.7 Receive/Decode

See 49.2.11. The decoder shall perform functions specified in the state diagram shown in Figure 49–17.

142.2.3.7.1 Constants

EBLOCK_R - see 49.2.13.2.1.

LBLOCK_R - see 49.2.13.2.1.

142.2.3.7.2 Variables

rx_coded – see 49.2.13.2.2.

rx_raw – see 49.2.13.2.2.

142.2.3.7.3 Functions

DECODE(rx_coded) - see 49.2.13.2.3.

NextRxValid(prev_rx_raw, next_rx_coded)
This function returns a Boolean indicating whether the *next_rx_coded* vector is valid given the classification of the previously received *prev_rx_raw vector*. The function returns the values according to Table 142–2. Vector classifications used in Table 142–2 are shown in Table 142–3.

NextRxVector()
function which returns the next 66-bit vector from the Descrambler.

142.2.3.7.4 State Diagrams

The OLT and the ONU shall implement the Receive/Decode process as depicted in Figure 142–10.

142.3 {NG-EPON type} PMA

142.3.1 Differential Encoder

Differential encoding as shown in Figure 142–11 shall be implemented in the OLT TX PMA for downstream. Differential encoding is optional to use by setting the control bit in the register, as defined in <Clause 45>.

142.3.2 Differential Decoder

Differential decoding shall be implemented in the ONU PMA RX function as shown in Figure 142–12. The ONU shall implement automatic detection of RX path differential encoding, and switch the decoder in appropriately.

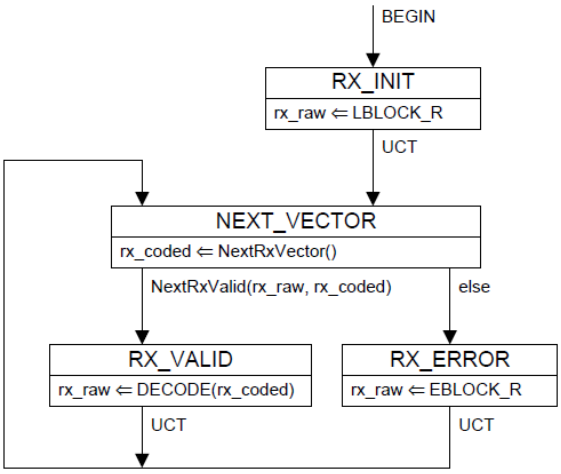

Figure 142–10—Transmit/Encode State Diagram

X_i = Input from OLT PCS FEC encoder
Y_i = Output to OLT PCS gearbox

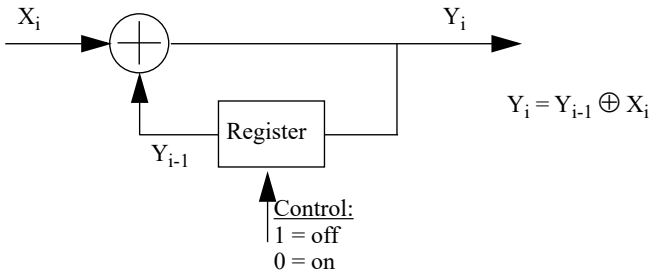

Figure 142–11—Differential encoding

X_i = Input from ONU PMA
Y_i = Output to ONU PCS Synchronizer

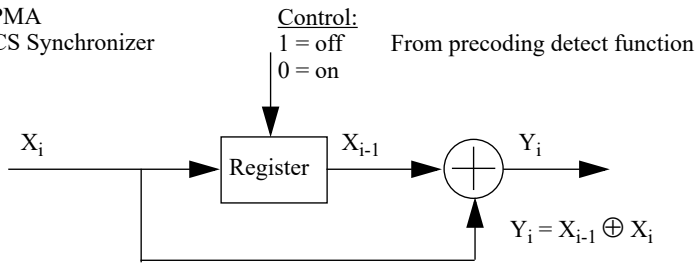

Figure 142–12—Differential decoding

142.4 Protocol implementation conformance statement (PICS) proforma for Clause 142, Physical Coding Sublayer and Physical Media Attachment for 100G-EPON[2]

142.4.1 Introduction

The supplier of a protocol implementation that is claimed to conform to Clause 142, Physical Coding Sublayer and Physical Media Attachment for 100G-EPON, shall complete the following protocol implementation conformance statement (PICS) proforma.

A detailed description of the symbols used in the PICS proforma, along with instructions for completing the PICS proforma, can be found in Clause 21.

142.4.2 Identification

142.4.2.1 Implementation identification

Supplier	
Contact point for inquiries about the PICS	
Implementation Name(s) and Version(s)	
Other information necessary for full identification—e.g., name(s) and version(s) for machines and/or operating systems; System Name(s)	
NOTE 1—Only the first three items are required for all implementations; other information may be completed as appropriate in meeting the requirements for the identification. NOTE 2—The terms Name and Version should be interpreted appropriately to correspond with a supplier's terminology (e.g., Type, Series, Model).	

[2]*Copyright release for PICS proformas:* Users of this standard may freely reproduce the PICS proforma in this subclause so that it can be used for its intended purpose and may further publish the completed PICS.

142.4.2.2 Protocol summary

Identification of protocol standard	IEEE Std 802.3-201x, Clause 142, Physical Coding Sublayer (PCS), and Physical Media Attachment (PMA) for point-to-point media, types 25GBASE-PR and 25/10GBASE-PR
Identification of amendments and corrigenda to this PICS proforma that have been completed as part of this PICS	
Have any Exception items been required? No [ ] Yes [ ] (See Clause 21; the answer Yes means that the implementation does not conform to IEEE Std 802.3-2015.)	
Date of Statement	

142.4.3 Major capabilities/options

Item	Feature	Subclause	Value/Comment	Status	Support