201. Reconciliation Sublayer, Physical Coding Sublayer, and Physical Media Attachment for 100G-EPON

201.1 Overview

This clause describes the Reconciliation Sublayer (RS), Physical Coding Sublayer (PCS) with FEC, and Physical Medium Attachment (PMA) used with 25GBASE-PR 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 202.

201.1.1 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.

201.1.2 Delay constraints

The MPCP relies on strict timing based on the distribution of timestamps. The actual delay is implementation dependent but an implementation shall maintain a combined delay variation through RS, PCS, and PMA sublayers of no more than 1 time_quantum (see 202.2.2.1) so as not to interfere with the MPCP timing.

201.2 Reconciliation Sublayer (RS) for 100G-EPON

201.2.1 Overview

This subclause extends Clause 46 and Clause 106 to enable multiple MACs to interface with at least one Physical Layer, and to enable data links with one data rate (e.g., 25 Gb/s) in one direction but another (e.g., 10 Gb/s) in the opposite direction. The number of MACs supported is limited only by the implementation. It is acceptable for only one MAC to be connected to this Reconciliation Sublayer. Figure 201–1 and Figure 201–2 show the relationship between this RS and the ISO/IEC OSI reference model. The mapping of XGMII/25GMII signals to PLS service primitives is described in 46.1.7 for XGMII and 106.1.7 for 25GMII with exceptions noted herein.

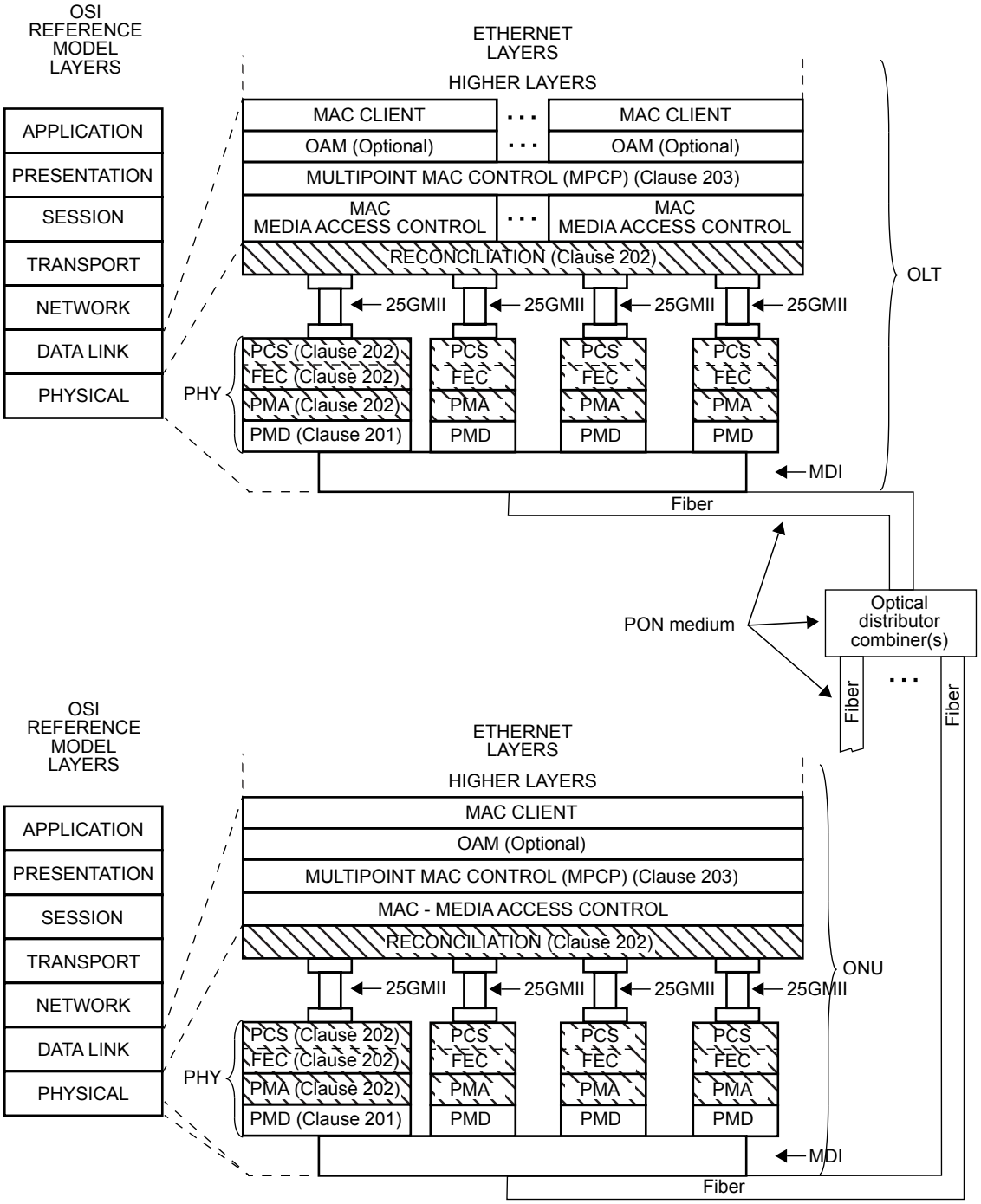


Figure 201–1—Relationship of 100G-EPON P2MP RS, PCS, and PMA to the ISO/IEC OSI reference model and the IEEE 802.3 Ethernet model

201.2.2 Dual-speed Media Independent Interface

In 10G-EPON architectures, the XGMII is the interface used to transfer data between the RS and the PCS. For 100G-EPON architecture, multiple instances of 25GMII interface are used to transfer data between the RS and the PCS: a single 25GMII interface for 25/25G-EPON, two 25GMII interfaces for 50G-EPON, and four 25GMII interfaces for 100G-EPON. When using a 25/10G-EPON architecture, a combination of both 25GMII and XGMII is needed in order to support transmission and reception at different speeds. Through the parallel use of the 25GMII and XGMII, the following modes are supported:

— Symmetric-rate operation for transmit and receive data paths at 25 Gb/s, 50 Gb/s, or 100 Gb/s, utilizing all of the functionality of the 25GMII defined in Clause 106.
— Asymmetric-rate operation for transmit and receive data paths at the OLT, utilizing transmit path functionality of the 25GMII defined in Clause 106 and receive path functionality of the XGMII defined in Clause 46.
— Asymmetric-rate operation for transmit and receive data paths at the ONU, utilizing transmit path functionality of the XGMII defined in Clause 46 and receive path functionality of the 25GMII defined in Clause 106.
— Coexistence of various ONU types by utilizing different data paths within the OLT.

201.2.2.1 25/25G-EPON, 50G-EPON, and 100G-EPON

Figure 201–2(a) depicts the data paths used in 25/25G-EPON. Similar principle applies to 50G-EPON and 100G-EPON, with the only difference in the number of 25GMII interface instances.

201.2.2.2 25/10G-EPON

At the OLT, the transmit path uses 25GMII signals TXD<31:0>, TXC<3:0> and TX_CLK, while the receive path uses XGMII signals RXD<31:0>, RXC<3:0> and RX_CLK. At the ONU, the transmit path uses XGMII signals TXD<31:0>, TXC<3:0> and TX_CLK, while the receive path uses 25GMII signals RXD<31:0>, RXC<3:0> and RX_CLK.

Figure 201–2(b) depicts the data paths used in 25/10G-EPON.

201.2.2.3 Dual-rate mode

To support coexistence of 25/25G-EPON and 25/10G-EPON ONUs on the same outside plant, the OLT may optionally support dual-rate mode. Dual-rate mode supports transmission and reception at both 25 Gb/s and 10 Gb/s. When operating in a dual-rate mode, a combination of 25GMII and XGMII data paths are used for transmission and reception. Figure 201–3 depicts the data paths used in an OLT operating in a dual-rate mode.

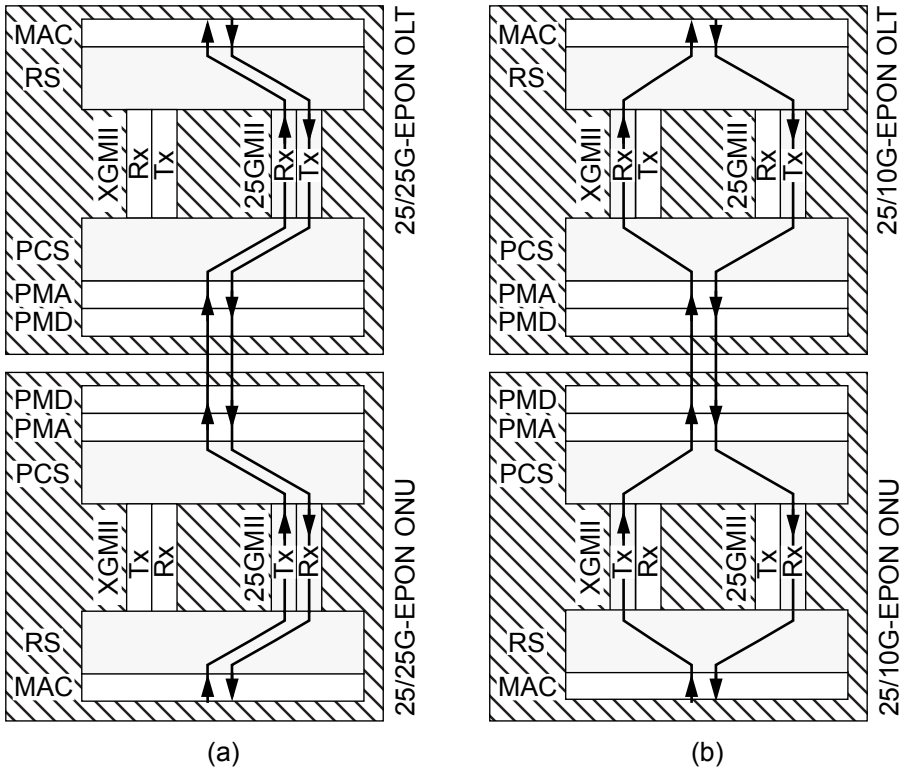


Figure 201–2—25/25G-EPON (a) and 25/10G-EPON (b) operation of OLT and ONU

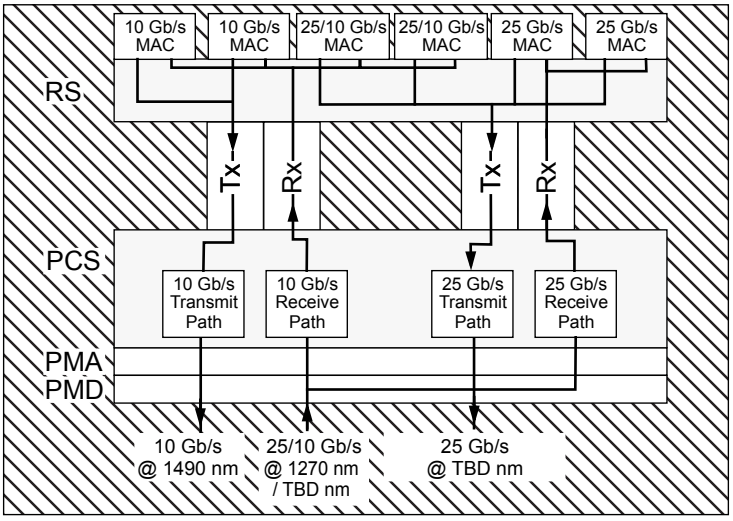


Figure 201–3—PCS and Reconciliation Sublayer for dual rate mode at OLT

201.2.2.4 Mapping of 25GMII and XGMII primitives

The mapping of 25GMII/XGMII signals to the PLS_DATA.request and PLS_DATA.indication primitives is described in 201.2.6. Additional details are provided in Table 201–1, which shows the mapping of PLS_DATA.request primitives to transmit interface signals for different types of OLTs and ONUs. Table 201–2 shows the mapping of PLS_DATA.indication primitives to receive interface signals for different types of OLTs and ONUs.

201.2.3 Summary of major concepts

A successful registration process, described in 202.3.3, results in the assignment of values to the MODE and LLID variables associated with a MAC. This may be one of many MACs in an OLT or a single MAC in an ONU. The MODE and LLID variables are used to identify a packet transmitted from that MAC and how received packets are directed to that MAC. The RS in the OLT shall operate in unidirectional mode as defined in 66.4.

As described in 202.1.2, multiple MACs within an OLT are bound to a single 25GMII in the case of a 25/25G-EPON OLT, or to an 25GMII transmit path and a XGMII receive path in the case of a 25/10G-EPON OLT. Only one PLS_DATA.request primitive is active at any time.

At the ONU, the MAC is either bound to an 25GMII in the case of a 25/25G-EPON ONU, or to an 25GMII receive path and a XGMII transmit path in case the of an 25/10G-EPON ONU.

In the transmit direction, the RS maps the active PLS_DATA.request to either the 25GMII signals (TXD<31:0>, TXC<3:0>, and TX_CLK) or the XGMII signals (TXD<31:0>, TXC<3:0>, and TX_CLK) according to the MAC instance generating the request. The RS replaces octets of preamble with the values of the transmitting MAC's MODE and LLID variables.

In the receive direction, the MODE and LLID values embedded within the preamble identify the MAC to which this packet should be directed. The RS establishes a temporal mapping of either the XGMII signals (TXD<31:0>, TXC<3:0>, and TX_CLK) or the 25GMII signals (RXD<31:0>, RXC<3:0> and RX_CLK) to the correct PLS_DATA.indication and PLS_DATA_VALID.indication primitives.

201.2.3.1 Application

This subclause applies to the interface between the MAC and PHY in an OLT or an ONU. The physical implementation of the interface is primarily intended to be chip-to-chip, but may also be used as a logical interface between ASIC logic modules within an integrated circuit. These interfaces are used to provide media independence, so that an identical media access controller may be used with all 25GBASE-PR and 25/10GBASE-PR PHY types.

201.2.4 XGMII structure

The XGMII structure is discussed in 46.1.6, and Figure 46–2 depicts a schematic view of the RS inputs and outputs.

201.2.5 25GMII structure

The 25GMII structure is discussed in 106.1.6.

201.2.6 Mapping of 25GMII and XGMII signals to PLS service primitives

Except as noted in Table 201–1 and Table 201–2, the mapping of the signals provided at the 25GMII to the PLS service primitives is defined in 46.1.7.

Table 201–1—Mapping of PLS_DATA.request primitive

MAC location	MAC operating speed	Transmit interface	Signals
OLT	10G-EPON (Tx: 10 Gb/s)	XGMII	TXD<31:0>, TXC<3:0>, TX_CLK
OLT	25/25G-EPON (Tx: 25 Gb/s)	25GMII	TXD<31:0>, TXC<3:0>, TX_CLK
OLT	25/10G-EPON (Tx: 25 Gb/s)	25GMII	TXD<31:0>, TXC<3:0>, TX_CLK
ONU	10G-EPON (Tx: 10 Gb/s)	XGMII	TXD<31:0>, TXC<3:0>, TX_CLK
ONU	25/25G-EPON (Tx: 25 Gb/s)	25GMII	TXD<31:0>, TXC<3:0>, TX_CLK
ONU	25/10G-EPON (Tx: 10 Gb/s)	XGMII	TXD<31:0>, TXC<3:0>, TX_CLK

Table 201–2—Mapping of PLS_DATA.indication primitive

MAC location	MAC operating speed	Receive interface	Signals
OLT	10G-EPON (Rx: 10 Gb/s)	XGMII	RXD<31:0>, RXC<3:0>, RX_CLK
OLT	25/25G-EPON (Rx: 25 Gb/s)	25GMII	RXD<31:0>, RXC<3:0>, RX_CLK
OLT	25/10G-EPON (Rx: 10 Gb/s)	XGMII	RXD<31:0>, RXC<3:0>, RX_CLK
ONU	10G-EPON (Rx: 10 Gb/s)	XGMII	RXD<31:0>, RXC<3:0>, RX_CLK
ONU	25/25G-EPON (Rx: 25 Gb/s)	25GMII	RXD<31:0>, RXC<3:0>, RX_CLK
ONU	25/10G-EPON (Rx: 25 Gb/s)	25GMII	RXD<31:0>, RXC<3:0>, RX_CLK

201.2.6.1 Functional specifications for multiple MACs

201.2.6.1.1 RS Transmit function

The transmit function is described in 76.2.6.1.2.

201.2.6.1.2 RS Receive function

The receive function is described in 76.2.6.1.3.

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

201.3.1 Overview

This subclause defines the physical coding sublayers 25GBASE-PR and 25/10GBASE-PR, supporting burst mode operation over the point-to-multipoint physical medium. The 25GBASE-PR PCS is specified to support 100G-EPON, 50G-EPON, and 25/25G-EPON, where both the receive and transmit paths operate at multiples of 25 Gb/s rate. The 25/10GBASE-PR PCS supports 25/10G-EPON, in which OLT transmit path and ONU receive path operate at 25 Gb/s, while the ONU transmit path and the OLT receive path operate at 10 Gb/s rate.

This subclause also specifies a forward error correction (FEC) mechanism to increase the optical link budget or the fiber distance. Figure 201–1 and Figure 201–2 show the relationship between the PCS sublayer and the ISO/IEC OSI reference model.

201.3.1.1 25/10GBASE-PR PCS

Conceptually, the 25/10GBASE-PR PCS represents a combination of transmit and receive functions defined in the 25GBASE-PR PCS (specified in this clause) and the 10GBASE-PR PCS (specified in Clause 76). At the OLT, the 25/10GBASE-PR consists of the 25GBASE-PR transmit function and the 10GBASE-PR receive function (see Figure 201–4). Reciprocally, at the ONU, the 25/10GBASE-PR PCS consists of the 25GBASE-PR receive function and the 10GBASE-PR transmit function (see Figure 201–5).

Deriving a specification for the 25/10GBASE-PR PCS from the 25GBASE-PR PCS and the 10GBASE-PR PCS specifications as described previously is a straightforward process; therefore, no further explicit specification for the 25/10GBASE-PR PCS is necessary.

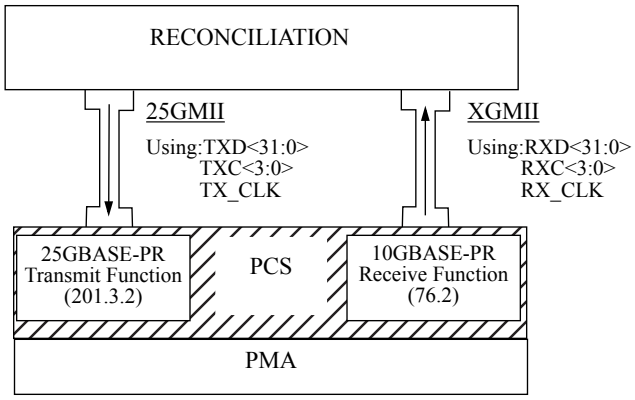


Figure 201–4—Conceptual diagram of 25/10GBASE-PR PCS, OLT Side

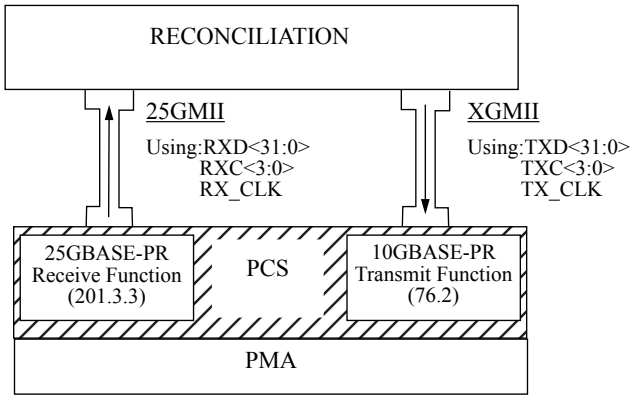


Figure 201–5—Conceptual diagram of 25/10GBASE-PR PCS, ONU Side

201.3.1.2 25GBASE-PR PCS

The 25GBASE-PR PCS extends the physical coding sublayer described in Clause 49 to support burst mode operation over the point-to-multipoint physical medium. Figure 201–6 illustrates the functional block diagram of the downstream path and Figure 201–7 shows the functional block diagram of the upstream path.

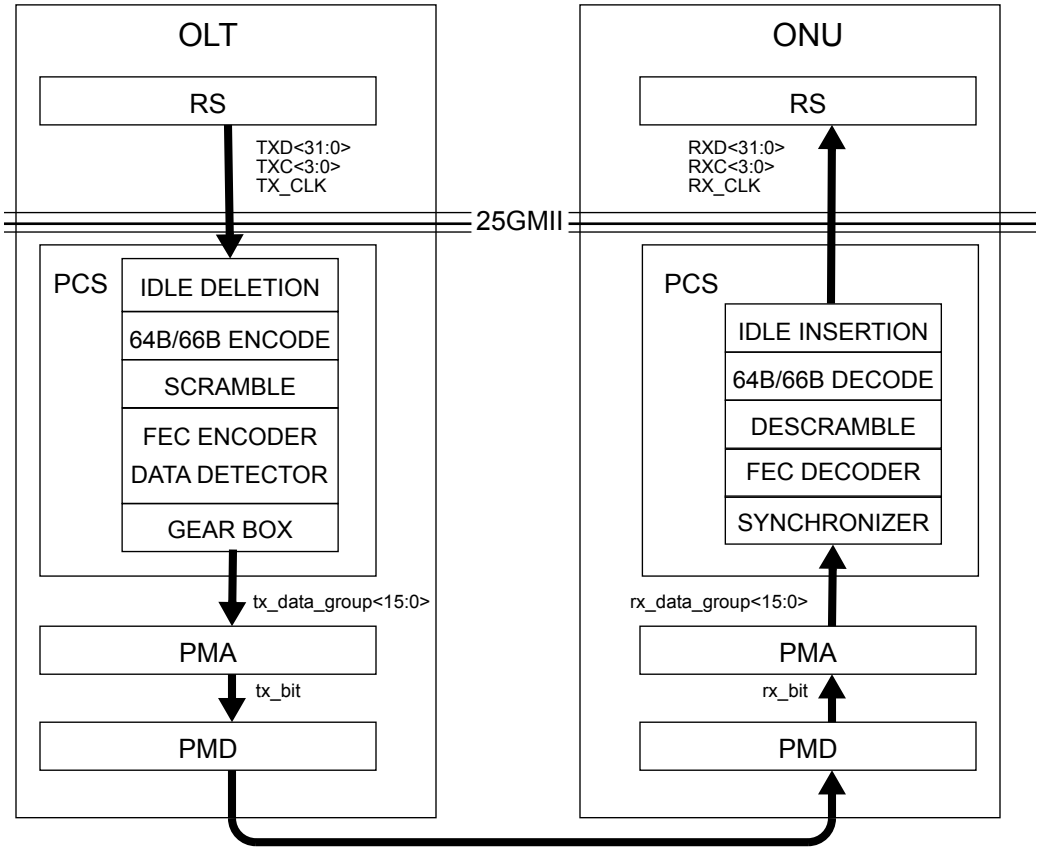


Figure 201–6—PCS functional block diagram, downstream path

201.3.2 PCS transmit function

This subclause defines the transmit direction of the physical coding sublayers for 25GBASE-PR and 25/10GBASE-PR. In the OLT, the PCS transmit function operates at a 25 Gb/s rate in a continuous mode. In the ONU, the PCS transmit function may operate at a 25 Gb/s rate, as specified herein (25GBASE-PR), or at a 10 Gb/s rate, as specified in Clause 76 (25/10GBASE-PR). For both 25GBASE-PR and 25/10GBASE-PR, the ONU PCS always operates in a burst mode in the transmit direction. When operating at the 10 Gb/s rate, the PCS includes a mandatory FEC encoder. Figure 201–6 illustrates the transmit direction of OLT PCS. Figure 201–7 illustrates the transmit direction of the ONU PCS.

201.3.2.1 Idle control character deletion

The Idle Deletion process is responsible for deleting excess Idle characters to allow the parity data to be inserted without increasing the PMD line rate. This process deletes four 72 bit vectors containing Idle characters per every thirty-one 72 bit vectors received from the 25GMII. The MPCP function ensures that sufficient Idle characters occur so that the minimum IPG is always preserved between two adjacent packets.

The Idle Deletion process is depicted in Figure 201–8 for OLTs and in Figure 201–9 for ONUs.

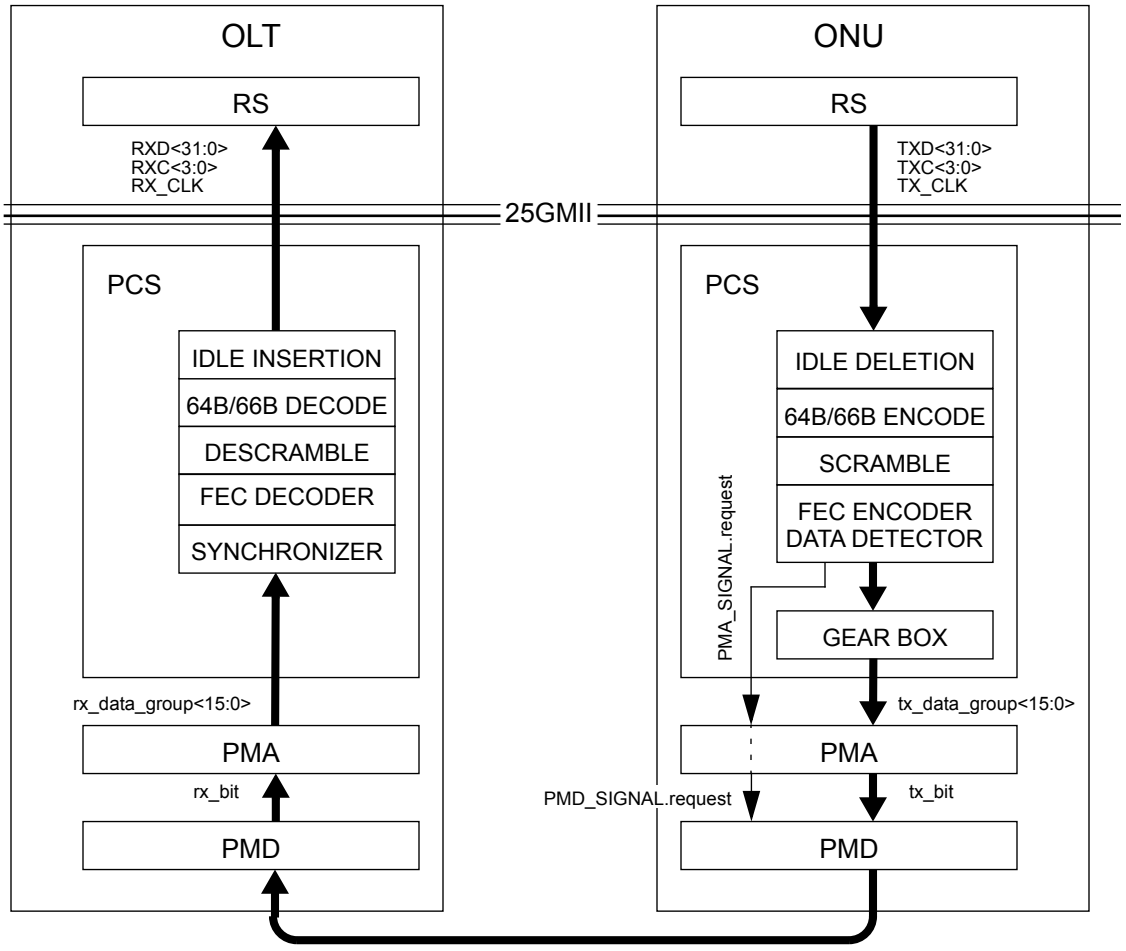


Figure 201–7—PCS functional block diagram, upstream path

201.3.2.1.1 Constants

FEC_DSize

TYPE: 16-bit unsigned integer

The number of 72-bit vectors constituting a payload of a FEC codeword. To normalize pre-FEC data rate, the Idle Deletion function removes FEC_PSize vectors per every FEC_DSize vectors transferred to the 64B/66B encoder.

Value: 27

FEC_PSize

TYPE: 16-bit unsigned integer

The number of 72-bit vectors constituting parity portion of a FEC codeword. To normalize pre-FEC data rate, the Idle Deletion function removes FEC_PSize vectors per every FEC_DSize vectors transferred to the 64B/66B encoder.

Value: 4

201.3.2.1.2 Variables

BEGIN

TYPE: Boolean

This variable is used when initiating operation of the state diagram. It is set to true following initialization and every reset.

DelayBound
TYPE: 16-bit unsigned integer
This value represents the delay sufficient to initiate the laser and to stabilize the receiver at the OLT (i.e., the maximum FIFO size expressed in 66-bit blocks). The value includes LaserOnTime (202.3.3.2), $T_{receiver\_settling}$, T_{CDR}, Burst Delimiter, and the two 66-bit blocks containing Idles, that precede the first packet in the burst. This variable is used only by the ONU.

tx_raw<71:0>
This variable is defined in 49.2.13.2.2.

tx_raw_out<71:0>
72-bit vector sent from the output of the Idle Deletion function to the 64B/66B encoder. The vector contains two 25GMII transfers mapped as shown for tx_raw<71:0>.

201.3.2.1.3 Functions

T_TYPE(tx_raw<71:0>)
This function is defined in 49.2.13.2.3.

201.3.2.1.4 Counters

DelCount
TYPE: 16-bit unsigned integer
Counts the number of 72-bit vectors that need to be deleted.

IdleCount
TYPE: 16-bit unsigned integer
Counts the number of 72-bit vectors containing Idle control characters or other control vectors.

VectorCount
TYPE: 16-bit unsigned integer
Counts the number of 72-bit vectors transmitted.

201.3.2.1.5 State diagrams

The OLT PCS shall perform the Idle Deletion process as shown in Figure 201–8. The ONU PCS shall perform the Idle Deletion process as shown in Figure 201–9. Should there be a discrepancy between a state diagrams and descriptive text, the state diagrams prevail.

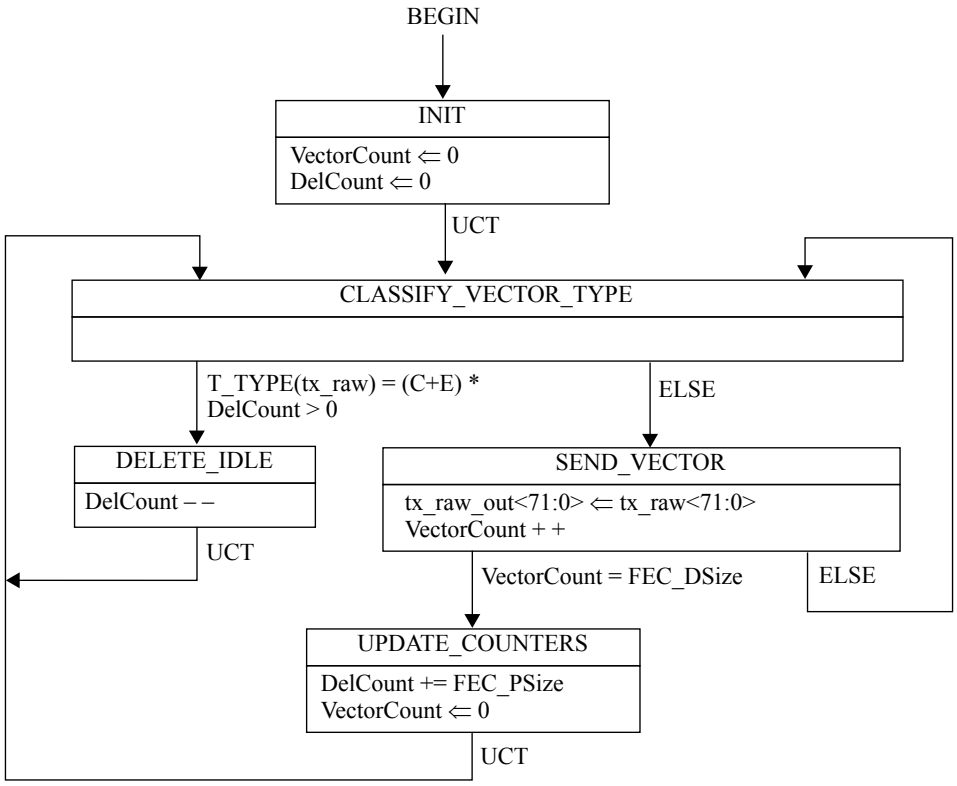


Figure 201–8—OLT Idle Deletion state diagram

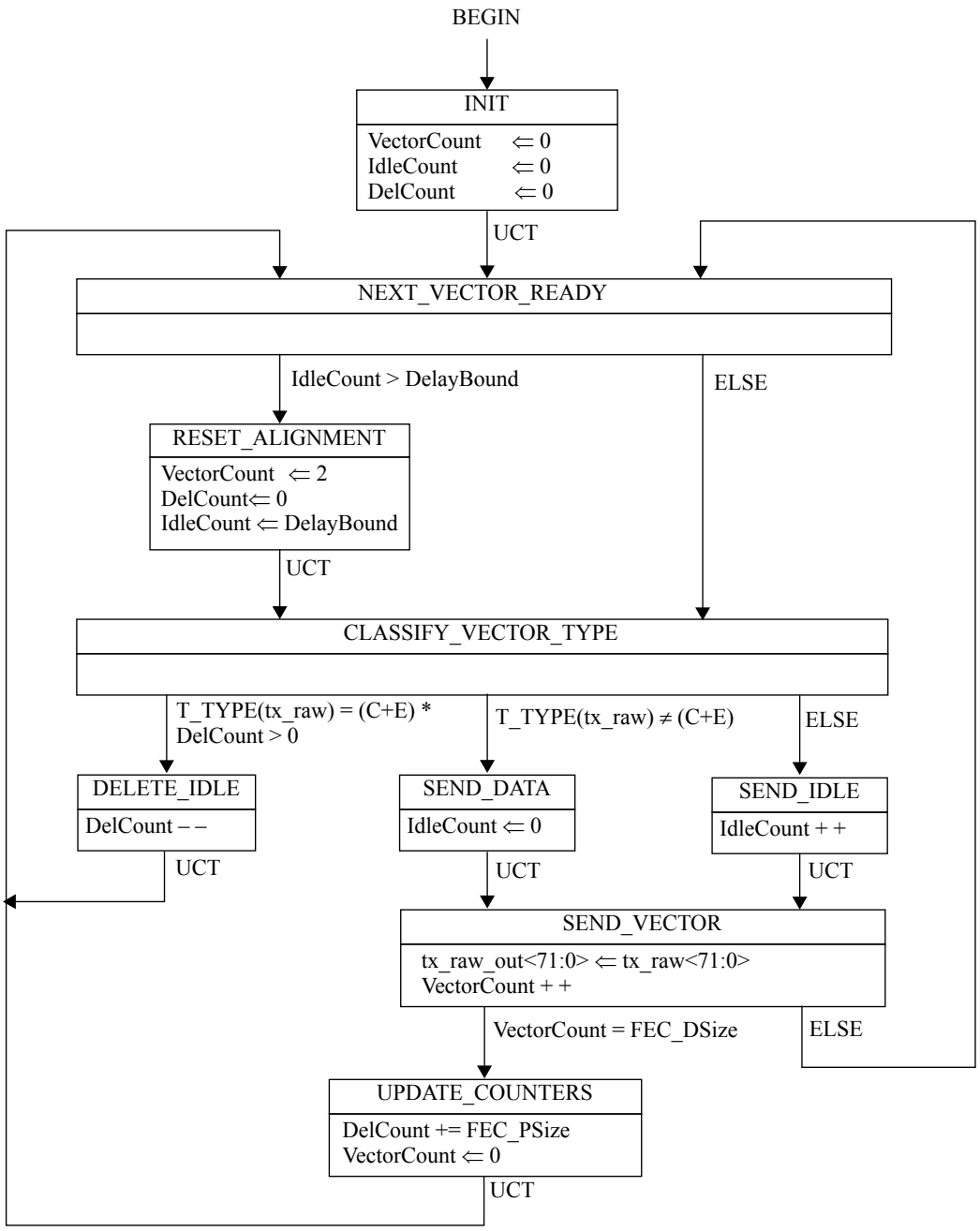


Figure 201–9—ONU Idle Deletion state diagram

201.3.2.2 64B/66B Encode

See 49.2.4. The encoder shall perform the functions specified in the state diagram shown in Figure 49–16.

201.3.2.3 Scrambler

See 49.2.6.

201.3.2.4 FEC encoding

The 25GBASE-PR-D, 25GBASE-PR-U, and 25/10GBASE-PR-D PCS shall encode the transmitted data stream using Reed-Solomon code (255,223). Annex 76A gives an example of RS(255,223) FEC Encoding.

201.3.2.4.1 FEC Algorithm [RS(255,223)]

The FEC code used for 25GBASE-PR links is a linear cyclic block code—the Reed-Solomon code (255, 223) over the Galois Field of GF(2^8)—a code operating on 8-bit symbols. The code encodes 223 information symbols and adds 32 parity symbols. The code is systematic, meaning that the information symbols are not disturbed in any way in the encoder and the parity symbols are added separately to each block.

The code is based on the generating polynomial shown in Equation (201-1).

$$G(Z) = \prod_{i=0}^{31}(Z - \alpha^i) = A_{32}Z^{32} + A_{31}Z^{31} + \ldots + A_0Z^0 \qquad (201\text{-}1)$$

where

α is a root of the binary primitive polynomial $x^8 + x^4 + x^3 + x^2 + 1$ and is represented as 0x02
A is a series representing the resulting polynomial coefficients of $G(Z)$, (A_{32} is equal to 0x01)
Z corresponds to an 8 bit GF(2^8) symbol
x corresponds to a bit position in a GF(2^8) symbol

The parity calculation shall produce the same result as the shift register implementation shown in Figure 201–10. Before calculation begins, the shift register shall be initialized to zero. The contents of the shift register are transmitted without inversion.

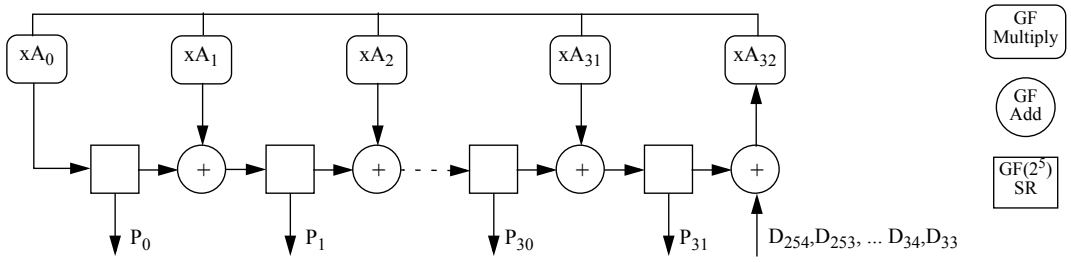


Figure 201–10—Circuit for generating FEC parity vector

A FEC parity vector is represented by Equation (201-2).

$$P(Z) = D(Z) \bmod G(Z) \qquad (201\text{-}2)$$

where

$D(Z)$ is the data vector $D(Z) = D_{222}Z^{254} + D_{221}Z^{253} + \ldots + D_0Z^{32}$. D_{222} is the first data octet and D_0 is the last.
$P(Z)$ is the parity vector $P(Z) = P_{31}Z^{31} + P_{30}Z^{30} \ldots + P_0Z^0$. P_{31} is the first parity octet and P_0 is the last.

A data octet $(d_7, d_6, ..., d_1, d_0)$ is identified with the element: $d_7\alpha^7 + d_6\alpha^6 + ... + d_1\alpha^1 + d_0$ in GF(2^8), the finite field with 2^8 elements. The code has a correction capability of up to sixteen symbols.

NOTE—For the (255, 223) Reed-Solomon code, the symbol size equals one octet. The d_0 is identified as the LSB and d_7 is identified as the MSB for all octets in accordance with the conventions of 3.1.1.

Bit ordering shall be as illustrated in Figure 201–11.

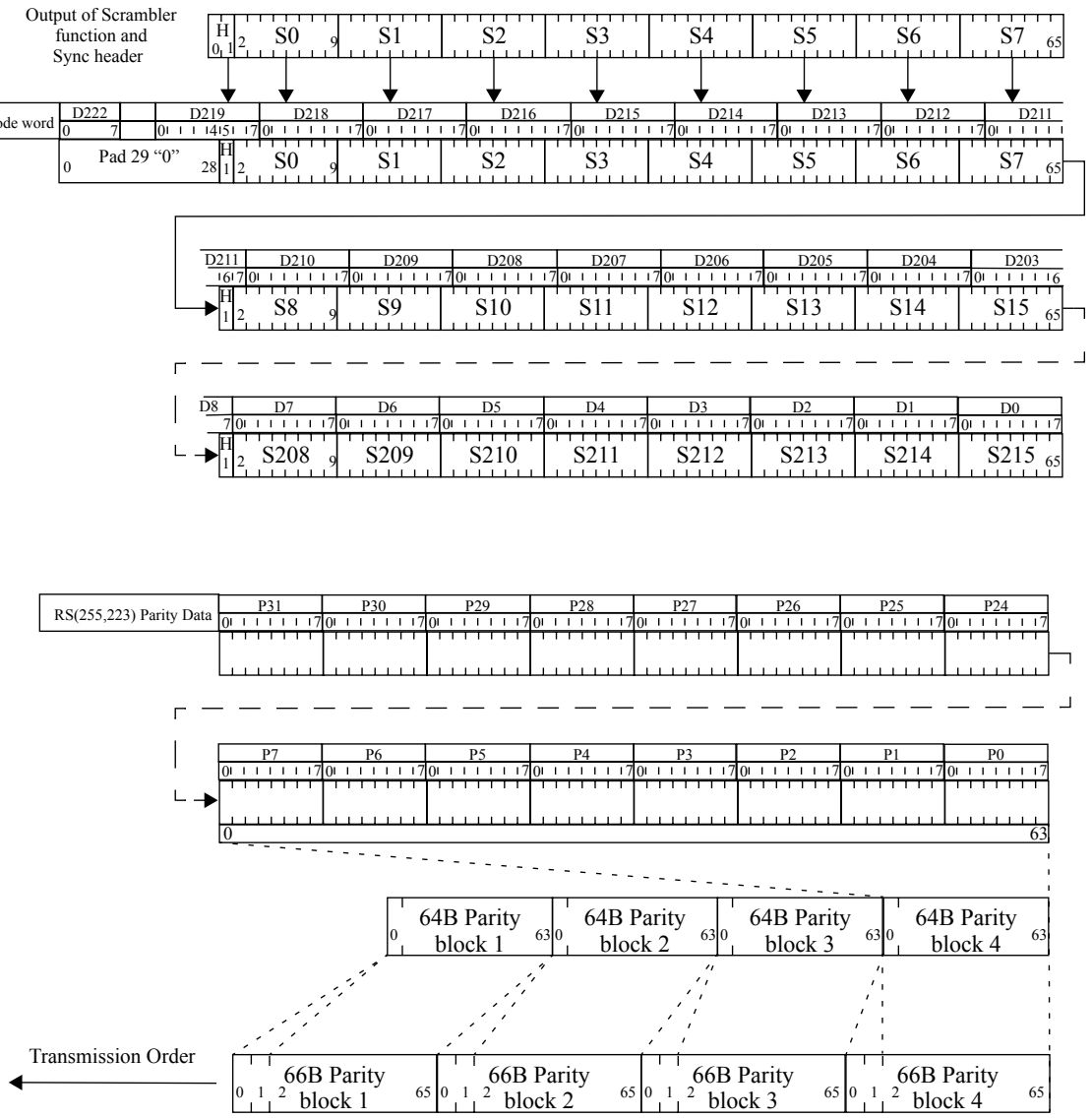


Figure 201–11—Bit ordering in FEC codeword generation

201.3.2.4.2 Parity calculation

Padding of FEC codewords and appending FEC parity octets in the 25GBASE-PR and 25/10GBASE-PR OLT PCS transmitter is illustrated in Figure 201–12. The 64B/66B encoder and scrambler produce 66-bit blocks. The FEC encoder accumulates 27 of these 66-bit blocks to form the basis of an FEC codeword, removing the redundant first bit (i.e., sync header bit <0>) of each block (the first bit is guaranteed to be the complement of the second bit).

The FEC encoder then prepends 29 padding bits (binary 0) to the 27 blocks (65 bits each) to form the 223-octet payload portion of an FEC codeword. This data is then FEC-encoded, resulting in the 32-octet parity portion of the FEC codeword. The 223-octet payload portion and 32-octet parity portion combine to form the 255-octet Reed-Solomon codeword. The padding is used to generate the FEC codeword but is not transmitted.

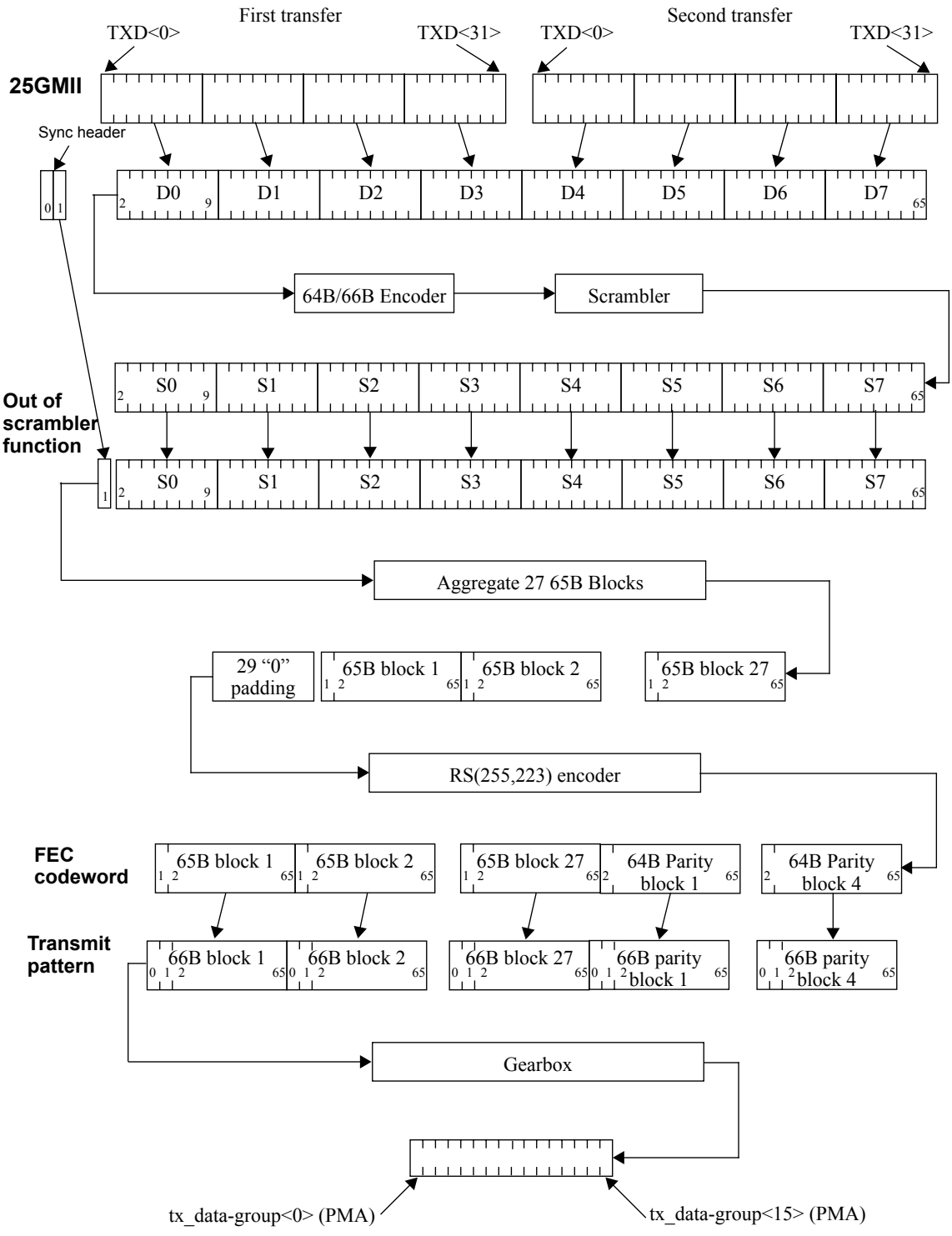


Figure 201–12—PCS Transmit bit ordering

201.3.2.4.3 FEC Transmission Block Formatting

As shown in Figure 201–12, after the Reed-Solomon codeword has been computed, the FEC encoder constructs the transmittable FEC codeword with the original sequence of twenty-seven 66-bit blocks (including the redundant sync bit, but not including the 29 “0” padding bits). The FEC encoder prepends a 2 bit sync header to each group of 64 parity bits to construct a properly formed 66 bit codeword, according to the predefined sync header pattern for the four 64-bit parity blocks: 00 11 11 00. Finally the four 66-bit parity blocks are appended following the twenty-seven 66-bit data blocks and transmitted to the PMA.

201.3.2.5 Data Detector

The 25GBASE-PR-D, 25GBASE-PR-U, and 25/10GBASE-PR-D PCS transmit path includes the Data Detector process. This process contains a delay line (FIFO_DD buffer) that stores the 66-bit blocks to allow insertion of the FEC parity data into the transmitted data stream. The length of the FIFO_DD buffer should be large enough to hold the amount of data equal to the maximum amount of 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 parity data).

201.3.2.5.1 Burst Mode operation (ONU only)

In addition to inserting the parity data into the data stream, the Data Detector process in the 25GBASE-PR-U PCS generates the PMA_SIGNAL.request(tx_enable) primitive to turn the laser on and off at the correct times.

Upon initialization, the laser is turned off. When the first 66-bit block containing data arrives at the buffer, the Data Detector sets the PMA_SIGNAL.request(tx_enable) primitive to the value ON, instructing the PMD sublayer to start the process of turning the laser on.

When the buffer becomes empty (i.e., contains only 66-bit blocks with Idle characters), the Data Detector sends the End of Burst Delimiter and after that sets the PMA_SIGNAL.request(tx_enable) primitive to the value OFF, instructing the PMD sublayer to start the process of turning the laser off. Between packets, Idle blocks arrive at the buffer. If the number of these Idle blocks is insufficient to fill the buffer then the laser is not turned off.

The length of the FIFO_DD buffer at the ONU shall be chosen such that the delay introduced by the buffer together with any delay introduced by the PMA sublayer is long enough to turn the laser on and to allow a laser synchronization pattern, Burst Delimiter pattern, and a predefined number of Idle blocks to be transmitted. The laser synchronization pattern allows the receiving optical detector to adjust its gain ($T_{receiver\_settling}$) and synchronize its receive clock (T_{CDR}). The Burst Delimiter allows the receiver to easily identify the beginning of FEC protected portion of the ONU transmission. The Idle control characters are used to synchronize the descrambler and establish start-of-packet delineation.

Figure 201–13 illustrates the details of the ONU burst transmission. In particular, this figure shows the details of the synchronization time and the FEC protected portions of the burst transmission.

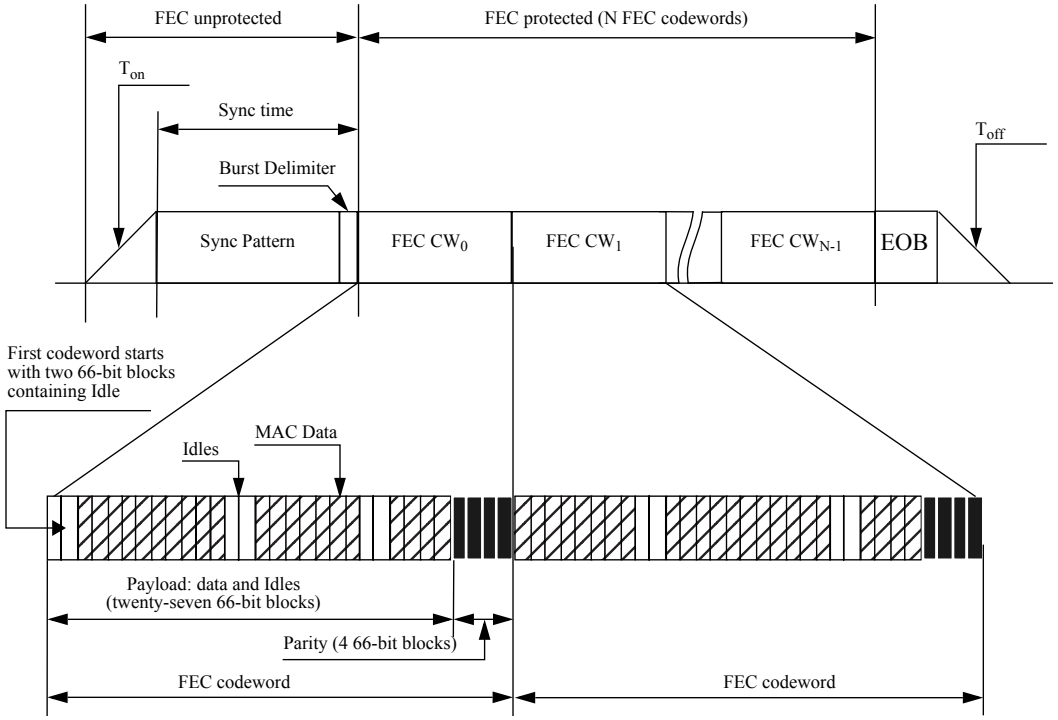


Figure 201–13—Details of burst composition

The ONU burst transmission begins with a Synchronization Pattern (see 201.3.2.5.2), which facilitates receiver clock recovery and gain control at the OLT. To facilitate FEC codeword synchronization, the ONU transmits a 66-bit BURST_DELIMITER (see Figure 201–14). When received at the OLT, the BURST_DELIMITER allows for FEC codeword alignment on the incoming data stream, even in the presence of bit errors. The BURST_DELIMITER is followed by two 66-bit blocks containing Idle codes. The first 66 bit block is used to synchronize the descrambler and a second 66-bit block is needed to provide packet delineation at the RS layer of the OLT. These two 66-bit Idle blocks are part of the first FEC codeword.

The ONU burst transmission ends with an END_BURST_DELIMITER (EOB) pattern of length TERMINATOR_LENGTH (see Figure 201–14). When received at the OLT, the burst terminator allows for the rapid reset of the OLT FEC synchronizer, so that it can search for the next burst. The burst terminator is not part of the last FEC codeword.

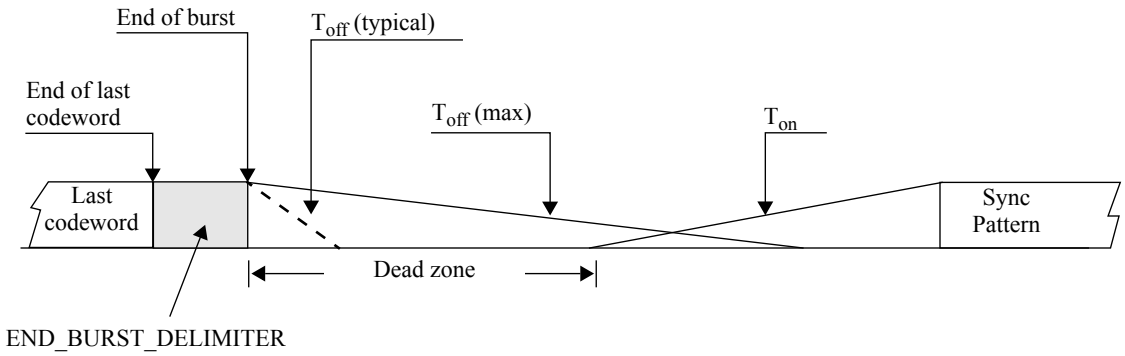


Figure 201–14—ONU burst transmission termination

The body of this subclause comprises state diagrams, including the associated definitions of variables, constants, and functions pertinent to the 25GBASE-PR and 25/10GBASE-PR OLT PCS transmitters. Should there be a discrepancy between a state diagram and descriptive text, the state diagram prevails. The notation used in the state diagrams in this clause follows the conventions in 21.5. State diagram variables follow the conventions of 21.5.2 except when the variable has a default value.

201.3.2.5.2 Constants

BURST_DELIMITER
: TYPE: 66-bit unsigned integer
A 66-bit value used to find the beginning of the first FEC codeword in the upstream burst.
Value: binary 01 followed by 0x 6B F8 D8 12 D8 58 E4 AB (transmission bit sequence: 01 1101 0110 0001 1111 0001 1011 0100 1000 0001 1011 0001 1010 0010 0111 1101 0101)

END_BURST_DELIMITER
: TYPE: 66-bit unsigned integer
A 66-bit value used to identify the end of the upstream burst transmission.
Value: binary 10 followed by 0x 55 55 55 55 55 55 55 55 (transmission bit sequence of 10 1010 1010 1010 1010 1010 1010 1010 1010 1010 1010 1010 1010 1010 1010 1010 1010)

FEC_DSize
: See 201.3.2.1.1.

FEC_PSize
: See 201.3.2.1.1.

SP
: Type: 66-bit unsigned integer
A 66-bit value used to for the burst mode synchronization pattern.
Value: binary 10 followed by 0x BF 40 18 E5 C5 49 BB 59 (transmission bit sequence 10 1111 1101 0000 0010 0001 1000 1010 0111 1010 0011 1001 0010 1101 1101 1001 1010)

TERMINATOR_LENGTH
: Type: 8-bit unsigned integer
Number of END_BURST_DELIMITER blocks that are transmitted at the end of each burst.
Value: 3

SH_DATA
Type: 2-bit unsigned integer
The value of synchronization header indicating that the given 66-bit block is a data block, as defined in 49.2.4.3.
Value: 0x02 (binary representation 10)

SH_CTRL
Type: 2-bit unsigned integer
The value of synchronization header indicating that the given 66-bit block is a control block, as defined in 49.2.4.3.
Value: 0x01 (binary representation 01)

NOTE—The binary representation of the sync header in here is different than that in Clause 49. In Clause 49, binary values are shown with the first transmitted bit (the LSB) on the left.

201.3.2.5.3 Variables

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 should be passed from Data Detector to the GearBox. This variable is reset to false upon read.

DelayBound
This variable is defined in 201.3.2.1.2.

FIFO_DD
TYPE: Array of 66-bit unsigned integer elements
A FIFO array used to store tx_coded<65:0> blocks while the parity is inserted and while burst preamble is generated (at the ONU only).

FifoSize
TYPE: 16-bit unsigned integer
Variable representing a number of elements stored in FIFO_DD.

SyncLength
TYPE: 16-bit unsigned integer
Required number of sync blocks per burst. The value of this variable is derived from the syncTime (excluding BURST_DELIMITER) and laserOnTime parameters defined in 202.3.3.

Transmitting
TYPE: Boolean
Boolean variable indicating whether the device is transmitting or not. At the ONU, the default value of Transmitting is false. At the OLT, this variable is always set to true.

tx_coded<65:0>
TYPE: 66-bit unsigned integer
66-bit block containing the output of the scrambler. The format for this vector 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<65:0>
TYPE: 66-bit unsigned integer

66-bit block containing the output of Data Detector being passed to the Gearbox. The format for this vector 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>.

201.3.2.5.4 Functions

RemoveFifoHead()

This function removes the first block in FIFO_DD and decrements the variable FifoSize by 1.

```
RemoveFifoHead()
   {
      // shift FIFO_DD forward
      FIFO_DD[0] = FIFO_DD[1]
      FIFO_DD[1] = FIFO_DD[2]
          ...
      FIFO_DD[FifoSize-2] = FIFO_DD[FifoSize-1]
      FifoSize --
   }
```

201.3.2.5.5 Messages

PMA_SIGNAL.request(tx_enable)

This primitive is used to turn the laser on and off at the PMD sublayer. In the OLT, this primitive shall always take the value ON. In the ONU, the value of this variable is controlled by the Data detector state diagram (see Figure 201–16).

SCRAMBLER_UNITDATA.request(tx_coded<65:0>)

A primitive generated by the SCRAMBLER transmit process conveying the next 66-bit block to be transmitted.

SUDR

Alias for SCRAMBLER_UNITDATA.request(tx_coded<65:0>).

201.3.2.5.6 Counters

IdleBlockCount

TYPE: 32-bit unsigned integer

The number of consecutive non-data blocks ending with the most recently received block. The non-data blocks are represented by sync header 10 (binary).

ParityBlockCount

TYPE: 8-bit unsigned integer

The number of parity blocks transmitted in a current FEC codeword. After reaching the full parity size (FEC_PSize = 4), this counter is reset to 0.

ProtectedBlockCount

TYPE: 8-bit unsigned integer

The number of blocks added to a payload of a current FEC codeword. After reaching the full payload size (FEC_DSize = 27), this counter is reset to 0.

SyncBlockCount

TYPE: 16-bit unsigned integer

The number of synchronization blocks transmitted in current burst.

201.3.2.5.7 State diagrams

The OLT and the ONU shall implement the Data Detector input process as depicted in Figure 201–15. The OLT shall implement the Data Detector output process as depicted in Figure 201–16(a). The ONU shall implement the Data Detector output process as depicted in Figure 201–16(b).

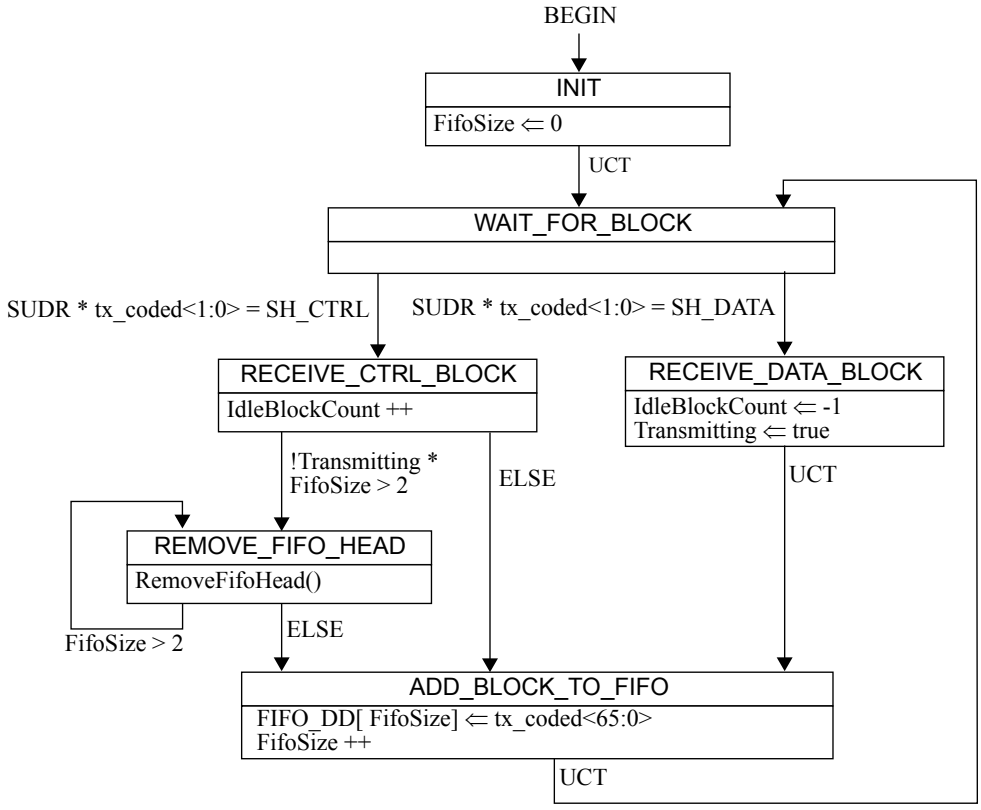
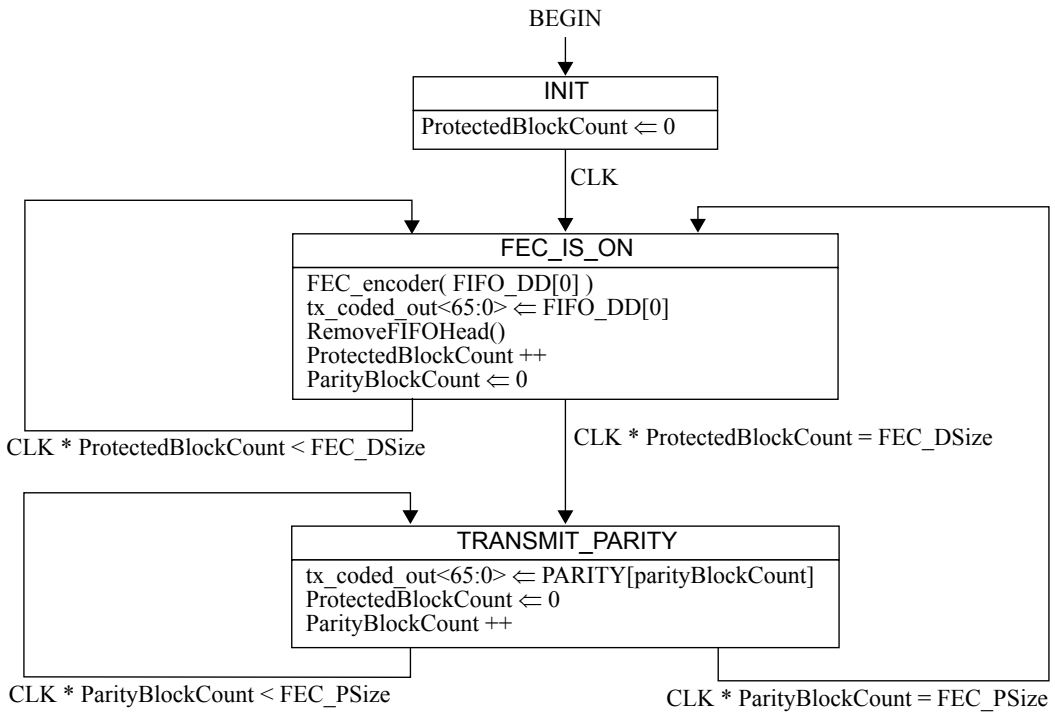


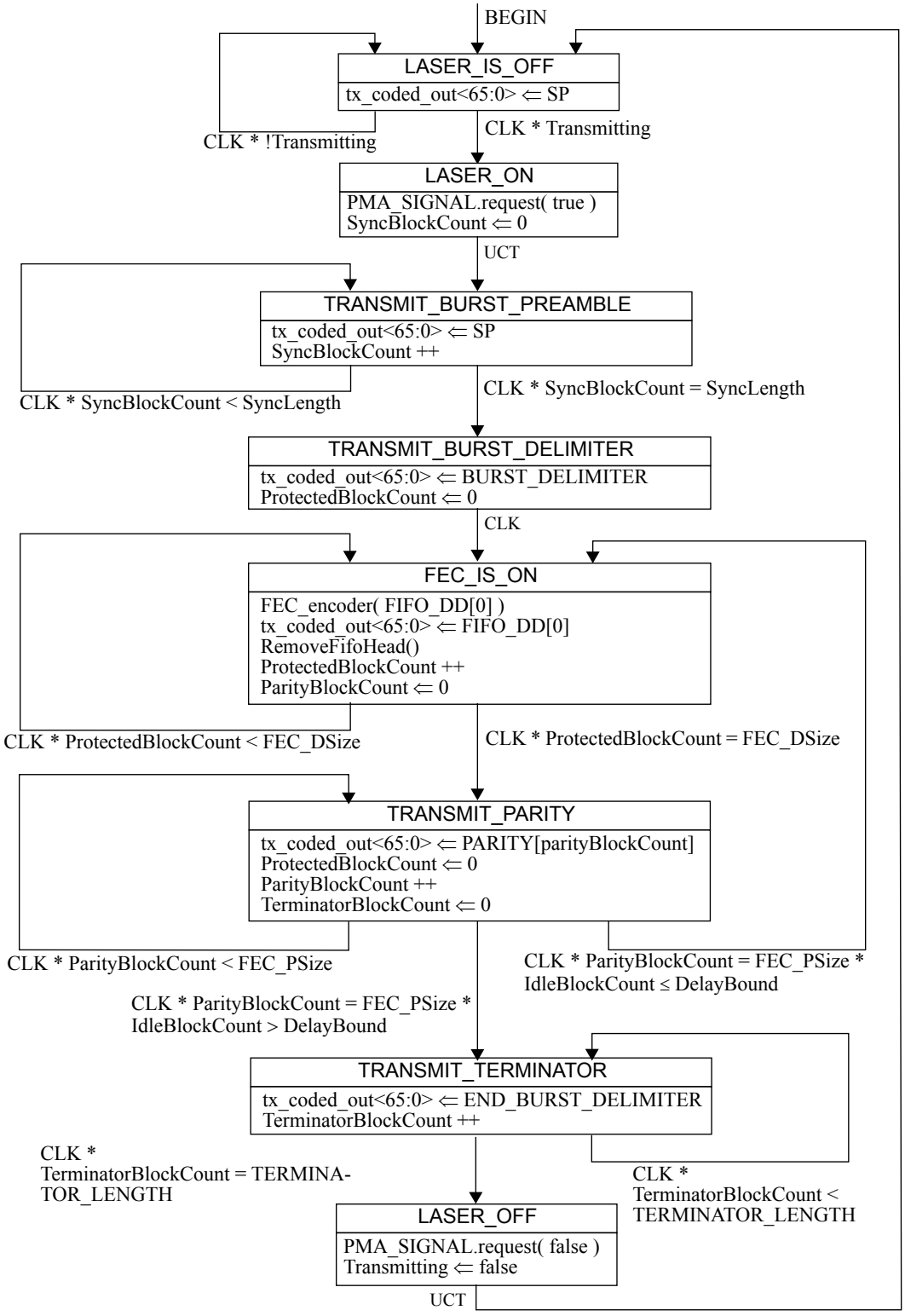
Figure 201–15—Data Detector, input process state diagram

201.3.2.6 Gearbox

See 49.2.7.



(a) OLT state diagram



(b) ONU state diagram

Figure 201–16—Data Detector, output process state diagram

201.3.3 PCS receive Function

This subclause defines the receive direction of physical coding sublayers for 25GBASE-PR and 25/10GBASE-PR. In the ONU, the PCS operates at a 25 Gb/s rate in a continuous mode. In the OLT, the PCS may operate at a 25 Gb/s rate, as specified herein (25GBASE-PR), or at a 10 Gb/s rate, compliant with Clause 76 (25/10GBASE-PR). For both 25GBASE-PR and 25/10GBASE-PR, the OLT PCS always operates in burst mode. When operating at the 25 Gb/s rate or at 10 Gb/s rate, the PCS includes a mandatory FEC decoder. The receive direction of ONU PCS is illustrated in Figure 201–6 and receive direction for the OLT PCS is illustrated in Figure 201–7.

201.3.3.1 OLT synchronizer

The OLT codeword synchronization function receives data via the 16-bit PMA_UNITDATA.indication primitive.

The OLT synchronizer forms a bit stream from the primitives by concatenating requests with the bits of each primitive in order from rx_data-group<0> to rx_data-group<15> (see Figure 201–17). It obtains lock to the thirty-one 66-bit blocks in the bit stream by looking for the burst delimiter. Lock is obtained as specified in the codeword lock state diagram shown in Figure 201–17. 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 looks for the end of the burst. When this is observed, then the state diagram deasserts codeword lock. The state diagram then goes back to searching for the burst delimiter.

201.3.3.1.1 Variables

BD_valid
: TYPE:Boolean
Indication that is set true if received block rx_coded matches the BURST_DELIMITER with less than 12 bits difference, and de-asserted otherwise.

cword_lock
: TYPE: Boolean
Boolean variable that is set true when receiver acquires codeword delineation.

CurrentBlock<65:0>
: TYPE: 66-bit vector
The last 66-bit block received. This variable has an initial value of 0.

decode_success
: TYPE: Boolean
Indication that is set true if the codeword was successfully decoded by the FEC algorithm, and false otherwise.

EOB_valid
: TYPE: Boolean
Indication that is set true if

$$\text{DistanceFromEob}(\text{CurrentBlock}) + \text{DistanceFromEob}(\text{PreviousBlock}) < 11$$

It is set to false otherwise.

inbuffer
: TYPE: bit array
An array of 2040 bits that holds the input to the FEC decoder.

input_buffer_location
TYPE: integer
An integer that points to the next appending location in the input buffer.

persist_dec_fail
TYPE: Boolean
Indication that is set when three consecutive decoding failures have occurred.

PreviousBlock<65:0>
TYPE: 66-bit vector
The 66-bit block received previous to the current block. This variable has an initial value of 0.

reset
See 49.2.13.2.2.

rx_coded<65:0>
See 49.2.13.2.2.

signal_ok
See 49.2.13.2.2.

201.3.3.1.2 Counters

decode_failures
TYPE:
Counter that holds the number of consecutive decoding failures.

sh_wndw_cnt
Count of the number of sync headers checked within the current 62-block window (composed of 2 codewords of 31 blocks each).

201.3.3.1.3 Functions

Append_inbuffer()
Appends the newly arrived 66-bit block into the input buffer of the FEC decoding algorithm, taking care to only insert the bits to be protected, and discarding the unwanted bits.

```
Append_inbuffer()
{
   BlockFromPMA()
   if (sh_wndw_cnt<27)
   {
      inbuffer[input_buffer_location]=rx_coded<1>
      input_buffer_location++
   }
   for(i=2, i<66, i++)
   {
      inbuffer[input_buffer_location]=rx_coded<i>
      input_buffer_location++
   }
}
```

BlockFromPMA()

Function that accepts the next received data from the PMA. Conceptually, this function serializes the 16-bit rx_data_group<15:0> to a bit stream at 25.78125 Gb/s, and then deserializes the resulting bit stream into a 66-bit wide rx_coded<65:0> block of data. It does not return until 66 bits have been transferred. Note that the internal design by which this function is accomplished is an implementation choice; however, the design operates such that a new 66-bit block is made available at the regular interval of 2.56 ns, and the transfers are made synchronous to the 25GMII clock.

Decode()

Triggers the FEC decoding algorithm to accept the contents of the input buffer, and do its decoding work. Note that this function is not blocking, and returns immediately. It is assumed that the FEC decoding algorithm copies the input buffer contents into its own internal memory, so that the input buffer is released to accept the next codeword.

DecodeWhenReady()

Determines if the inbuffer contains a full codeword, and if so, it triggers the Decode function, and then clears the inbuffer for the next codeword.

```
DecodeWhenReady()
{
   if (sh_wndw_cnt=0 or sh_wndw_cnt=31)
   {
      if (cword_lock)
      {
         Decode();
      }
      Flush_inbuffer();
   }
}
```

DistanceFromEob(block<65:0>)

Returns the Hamming distance between the supplied block and the END_BURST_DELIMITER

Flush_inbuffer()

Flushes the input buffer of the FEC decoding algorithm block.

```
Flush_inbuffer()
{
   for(i=0, i<2040, i++)
   {
      inbuffer[i]=0
   }
   input_buffer_location = 29
}
```

SLIP_One_Bit

Causes the next candidate block sync position to be tested. The next candidate is exactly one bit later than the previous candidate – no burst alignments may be skipped. Following the conceptual model mentioned in “BlockFromPMA,” this function transfers one more bit from the 16-bit serializer to the 66-bit deserializer.

201.3.3.1.4 State diagram

The OLT Synchronizer shall implement the state diagram as depicted in Figure 201–17. Should there be a discrepancy between a state diagram and descriptive text, the state diagram prevails.

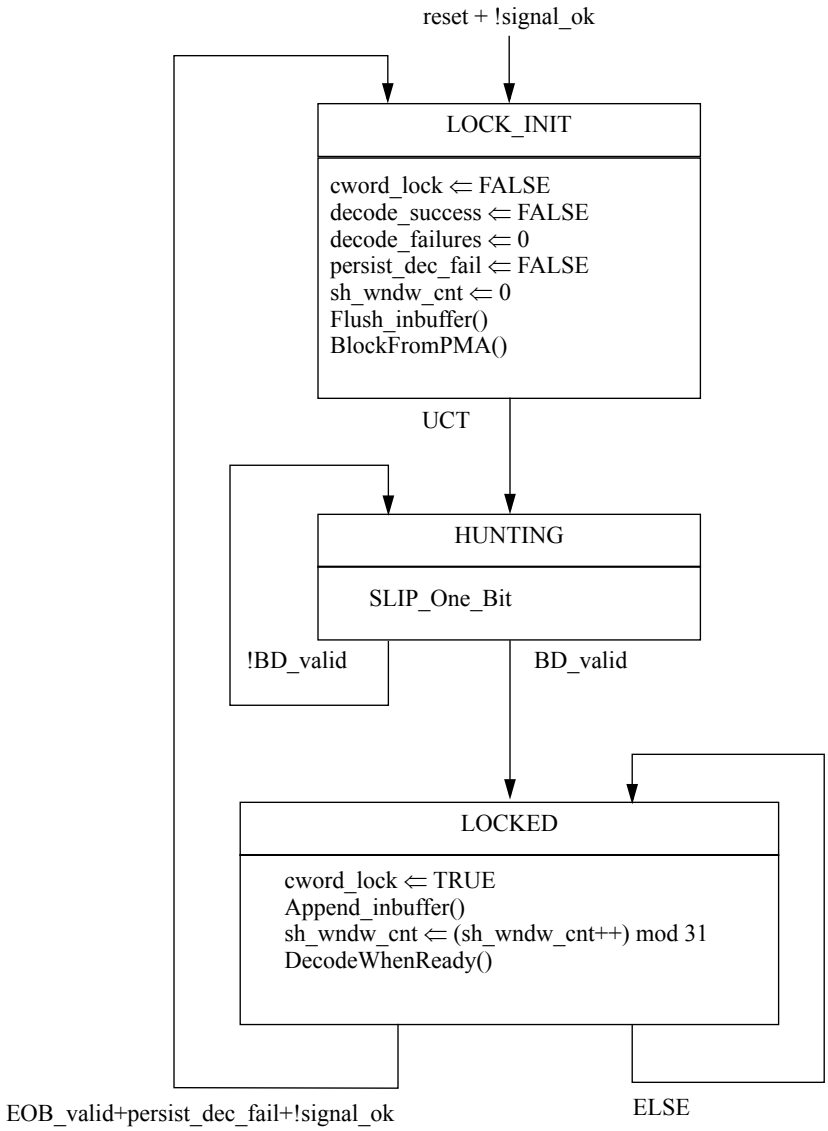


Figure 201–17—OLT Synchronizer state diagram

201.3.3.2 ONU Synchronizer

The codeword synchronization function receives data via the 16-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<0> to rx_data-group<15> (see Figure 201–18). It obtains lock to the thirty-one 66-bit blocks in the bit stream using the sync headers and outputs 2040-bit codewords to the FEC decoder function.

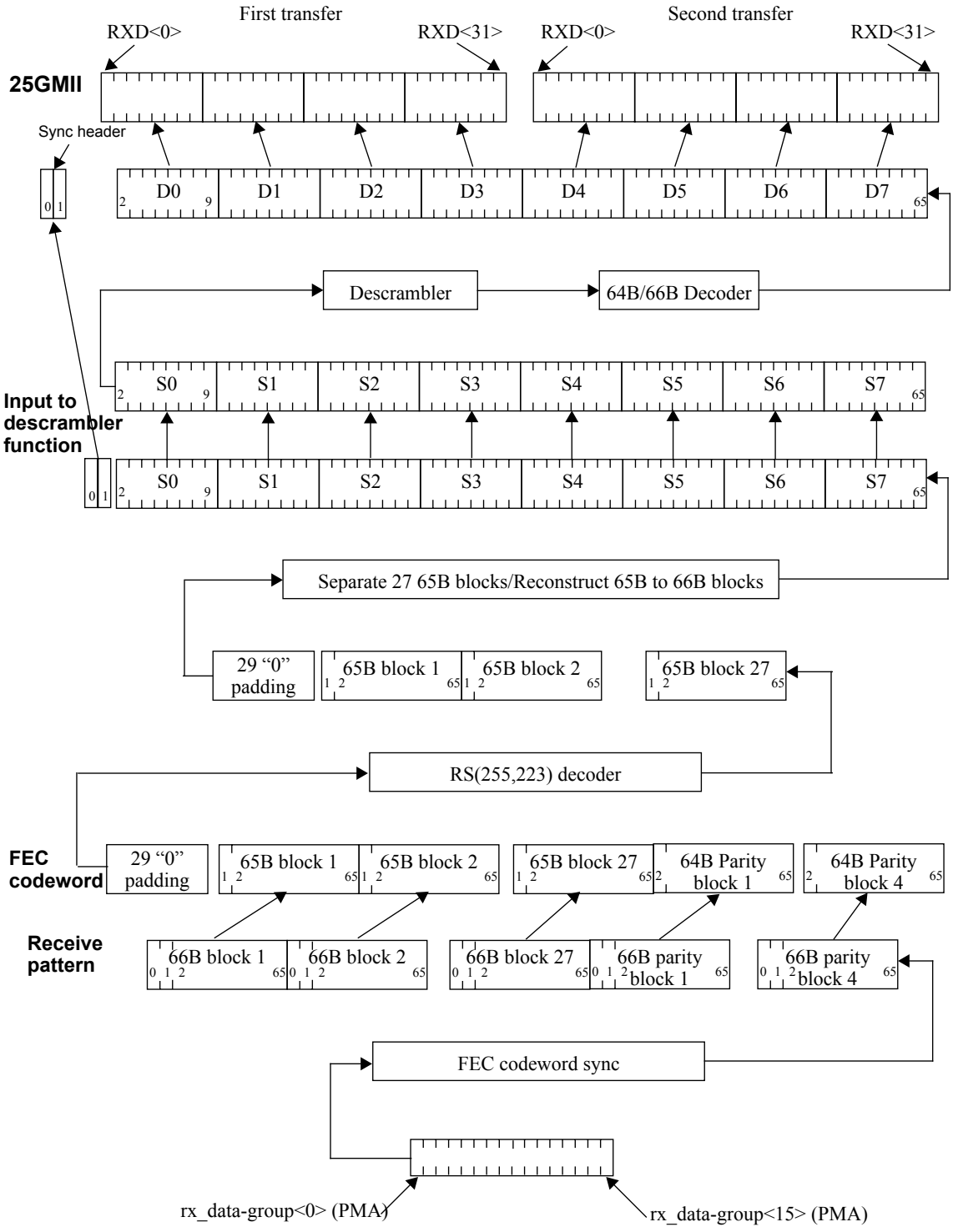


Figure 201–18—PCS Receive bit ordering

The incoming sync header pattern is 27 conventional (Clause 49) sync headers (binary 01 or 10), and then binary 00, 11, 11, and finally binary 00. The ONU synchronizer attempts to match this pattern to the

received data stream, and when it finds a perfect match of two full codewords (62 blocks), it then asserts codeword lock.

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.)

201.3.3.2.1 Constants

All the relevant constants defined in 49.2.13.2.1 are inherited. In addition, the following items are defined.

SH_CW_PATTERN
: TYPE: array of 8-bit unsigned integer
31 element array of codeword sync header bit counts, where each element is set to the value 1 except for:
Value:
SH_CW_PATTERN[27]=0
SH_CW_PATTERN[28]=2
SH_CW_PATTERN[29]=2
SH_CW_PATTERN[30]=0

201.3.3.2.2 Variables

cword_lock
: See 201.3.3.1.1.

decode_success
: See 201.3.3.1.1.

persist_dec_fail
: See 201.3.3.1.1.

reset
: See 49.2.13.2.2.

sh_valid
: TYPE: Boolean array
Indication that is set true if received block rx_coded has valid sync header bits for the supposed current position in the FEC codeword. That is, sh_valid[i] is asserted if (rx_coded<0> + rx_coded<1>) = SH_CW_PATTERN[i mod 31] and de-asserted otherwise.

signal_ok
: See 49.2.13.2.2.

slip_done
: See 49.2.13.2.2.

test_sh
: See 49.2.13.2.2.

201.3.3.2.3 Counters

decode_failures
See 201.3.3.1.2.

FEC_cnt
TYPE: 8-bit unsigned integer
This counter keeps track of the parity sync header index that is currently being tested.

sh_wndw_cnt
See 201.3.3.1.2.

sh_valid_cnt
See 49.2.13.2.4.

201.3.3.2.4 Functions

Append_inbuffer()
See 201.3.3.1.3.

DecodeWhenReady()
See 201.3.3.1.3.

SLIP
See 49.2.13.2.3.

201.3.3.2.5 State diagram

The ONU Synchronizer shall implement the state diagram as depicted in Figure 201–19. Should there be a discrepancy between a state diagram and descriptive text, the state diagram prevails.

201.3.3.3 FEC decoding process

The 25GBASE-PR-D, 25GBASE-PR-U, and 25/10GBASE-PR-U PCS shall correct errors in the received data stream using Reed-Solomon code (255,223). The FEC decoder corrects or confirms the correctness of the twenty-seven 66-bit blocks contained in the FEC codeword based on the four 66-bit blocks of parity information. The FEC decoding process then forwards the 66-bit data blocks to the descrambler and discards the parity blocks. The FEC decoding process is also responsible for setting bit 0 of the sync header to the inverse of bit 1 of the sync header. The handling of data leaving the FEC decoder and going to the descrambler is specified in the FEC decoding process state diagram shown in Figure 201–20. Implementations shall be capable of correcting up to 16 symbols in a codeword and detecting uncorrectable codewords.

The synchronizer state diagram accumulates a full codeword in a buffer. If the synchronizer is locked, then the FEC decoding process is triggered. The FEC algorithm then processes the buffer. The algorithm produces two outputs: the decode_success signal and (if successful) the corrected buffer. The data portion of the buffer is then read out to the descrambler logic in 66-bit blocks, as normal. Note that the rate of 66 bit transfers here is reduced due to the removal of the FEC parity blocks. This is corrected in the Idle Insertion step (see Figure 201–22).

If decode_success is false, then a counter is incremented (see 45.2.3.40). If there are three decoding failures in a row, then the Persist_dec_fail signal is asserted. This signal then resets the synchronizer.

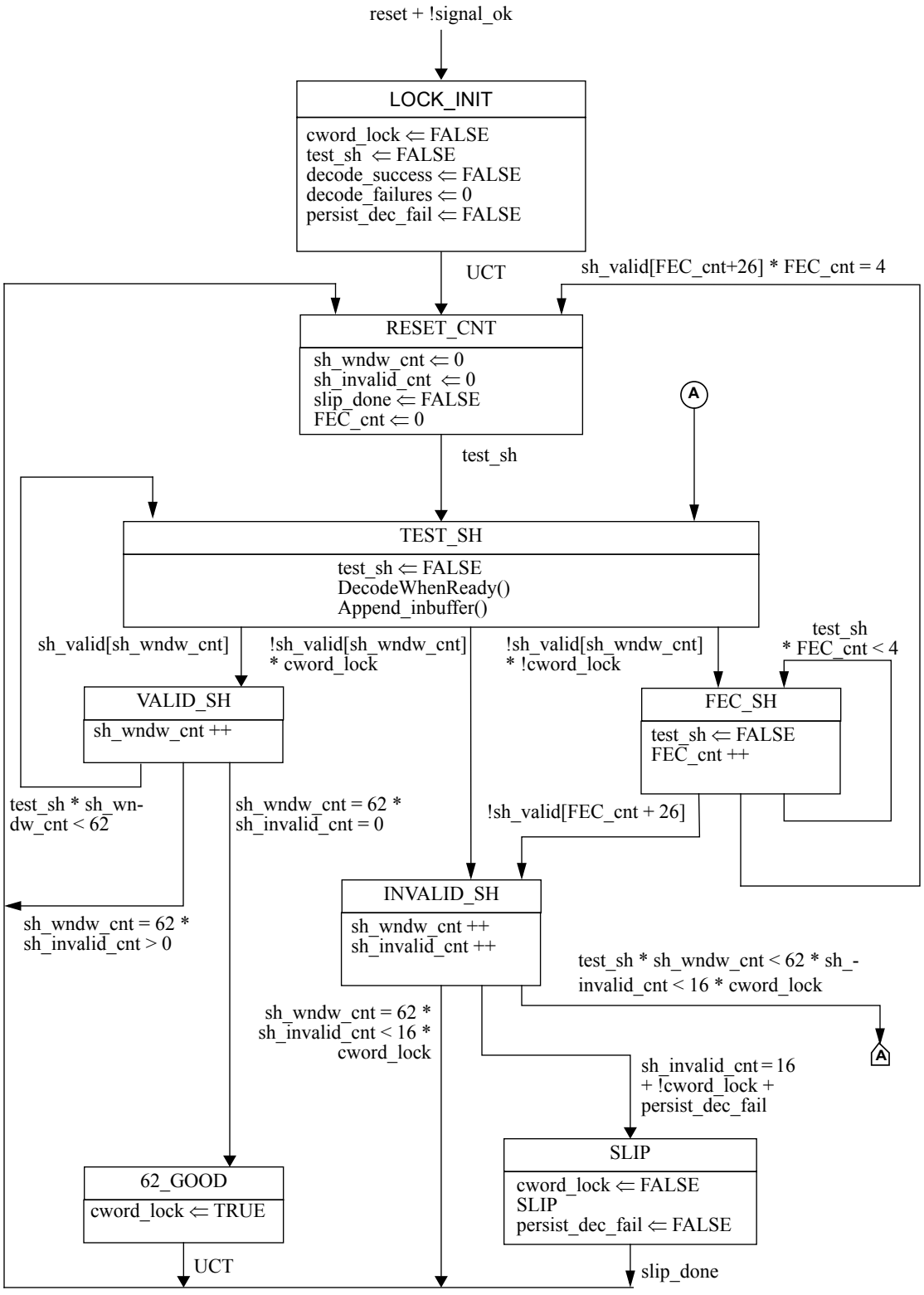


Figure 201–19—ONU Synchronizer state diagram

The FEC decoding process shall provide a user option to indicate an uncorrectable FEC codeword (due to an excess of symbols containing errors) to higher layers. If this option is set to be true, the FEC decoding process checks for the value of decode_success. If the variable decode_success is set to false, then each sync header of the received payload blocks in the FEC codeword is set to a value of binary 00. However, the data blocks are nevertheless passed to the descrambler to maintain descrambling synchronization. When this option is set to FALSE and decode_success is FALSE then each received payload block is passed unchanged.

201.3.3.3.1 Variables

decode_done
: TYPE: Boolean
Indication that is transiently set when the FEC decoder algorithm has completed its processing and the corrected data is present in the output buffer.

decode_success
: See 201.3.3.1.1.

mark_uncorrectable
: TYPE: Boolean
Control variable that is set to true if the uncorrectable errors are to be marked.
Default: TRUE

outbuffer
: TYPE: bit array
An array of 2040 bits that holds the output of the FEC decoder.

persist_dec_fail
: See 201.3.3.1.1.

rx_code_corrected
: Type: 66-bit vector
The next block of data to be sent to the scrambler.

201.3.3.3.2 Counters

decode_failures
: See 201.3.3.1.2.

corrected_FEC_codewords_counter
: A corrected block is an FEC codeword that was received with one or more errored symbols, and that has been corrected by the FEC decoder.
The corrected_FEC_codewords_counter counts once for each corrected FEC codeword processed. This is a 32-bit counter. This variable is provided by a management interface that may be mapped to the 45.2.3.39 register (3.76, 3.77).

FEC_uncorrected_blocks_counter
: An uncorrected block is an FEC codeword that was received with 17 or more errored symbols, and that has not been corrected by the FEC decoder.
FEC_uncorrected_blocks_counter counts once for each uncorrected FEC codeword processed. This is a 32-bit counter. This variable is provided by a management interface that may be mapped to the 45.2.3.40 register (3.78, 3.79).

201.3.3.3.3 Functions

All the relevant functions defined in 49.2.13.2.3 are inherited. In addition, the following items are defined.

BlockFromPMA()
See 201.3.3.1.3.

BlockToDescrambler()
Function that sends the next rx_coded_corrected<65:0> block to the descrambler. It does not return until the transfer is completed, and each transfer takes 2.56 ns and is synchronized to the 25GMII clock.

Flush_inbuffer()
See 201.3.3.1.3.

Read_outbuffer(i)
Passes output buffer contents to the descrambler, with the appropriate format.

```
Read_outbuffer[i]
{
   int offset = 29+i*65
   for(j=0, j<65, j++)
   {
      rx_coded_corrected<j+1> = outbuffer[j+offset]
   }
   if (!decode_success AND mark_uncorrectable)
   {
      rx_coded_corrected<0> = 0
      rx_coded_corrected<1> = 0
   }
   else
   {
      rx_coded_corrected<0> = !rx_coded_corrected<1>
   }
   BlockToDescrambler()
}
```

SLIP
See 49.2.13.2.3.

201.3.3.3.4 State diagrams

The body of this subclause comprises state diagrams, including the associated definitions of variables, constants, and functions pertinent to the 25GBASE-PR-D, 25GBASE-PR-U, and 25/10GBASE-PR-U PCS receivers. Should there be a discrepancy between a state diagram and descriptive text, the state diagram prevails. The notation used in the state diagrams in this clause follows the conventions in 21.5.

The FEC decoding process shall be implemented in the PCS as depicted in Figure 201–20. Should there be a discrepancy between a state diagram and descriptive text, the state diagram prevails.

201.3.3.4 BER monitor

The BER monitor is described in Figure 201–21. This BER monitor function operates on the uncorrected incoming data stream.

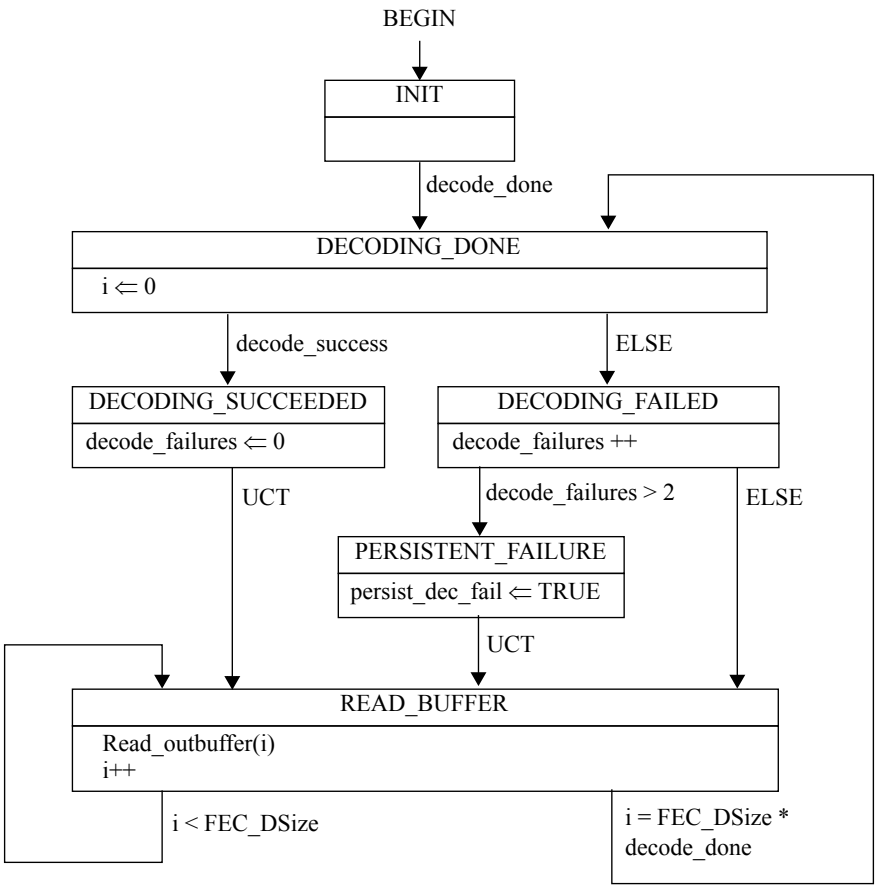


Figure 201–20—FEC decoding process state diagram

201.3.3.4.1 Variables

BER_Monitor_Interval
: Indicates the time window associated with the BER monitor function. The timers in the BER monitor state diagram depend on this configurable variable. This value is reflected in MDIO register 3.80.

ber_test_sh
: See 49.2.13.2.2.

BER_Threshold
: Indicates the threshold value of invalid sync headers associated with the BER monitor function. When BER_Threshold bad sync headers are encountered within the BER Monitor_Interval period, the BER monitor raises the hi_ber flag. When the number of bad sync headers encountered within the BER_Monitor_interval period less than the BER_Threshold, the hi_ber flag is turned off. This value is reflected in MDIO register 3.82.

hi_ber
: See 49.2.13.2.2.

reset
: See 49.2.13.2.2.

ber_test_sh
See 49.2.13.2.2.

201.3.3.4.2 Timers

State diagram timers follow the conventions of 14.2.3.2.

interval_timer
Timer that is triggered every BER_monitor_interval us +1%, –25%.

201.3.3.4.3 Counters

ber_cnt
See 49.2.13.2.4.

201.3.3.4.4 State diagrams

The BER monitor state diagram is present only in the ONU. It is shown in Figure 201–21.

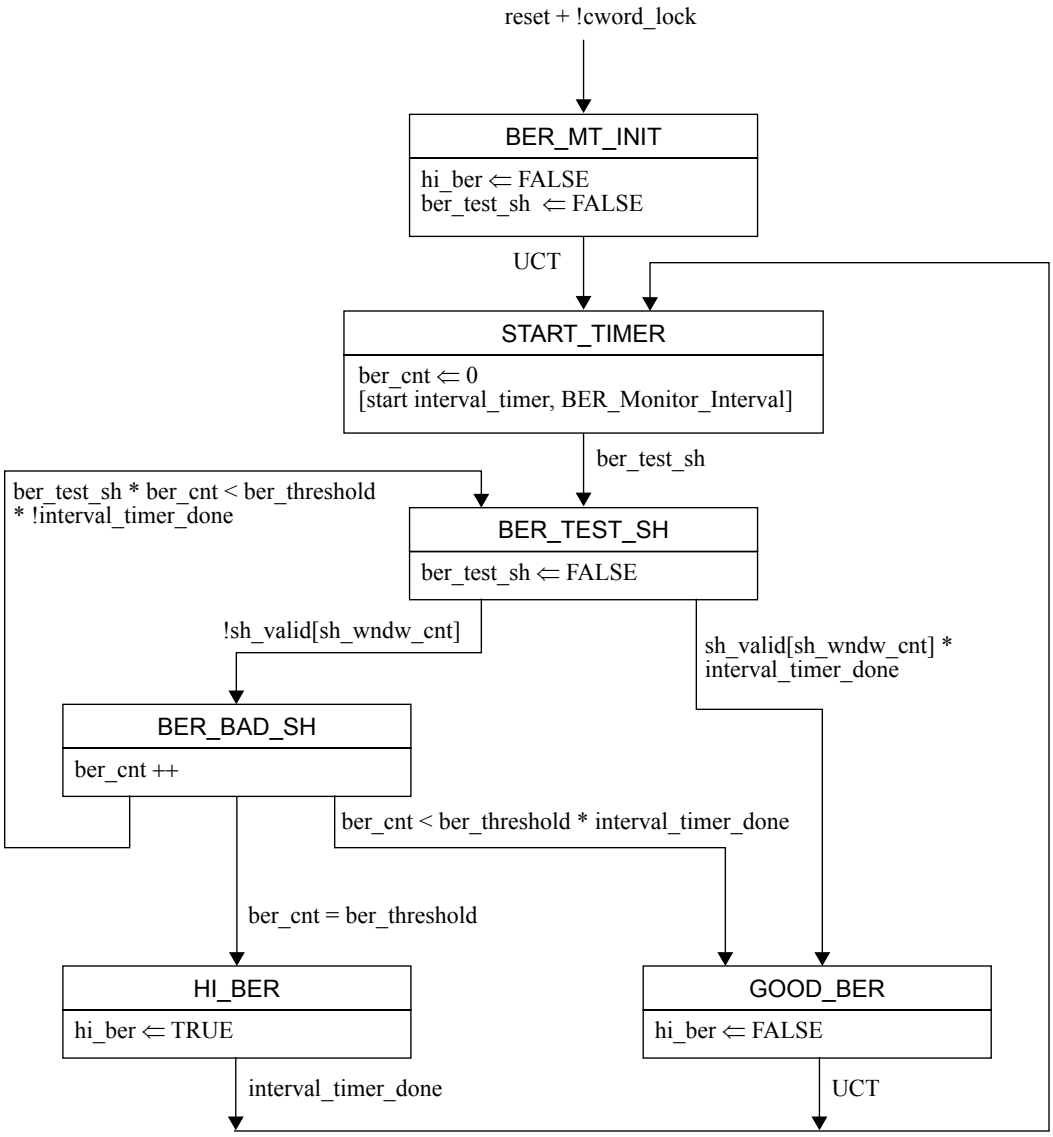


Figure 201–21—BER monitor state diagram (ONU only)

201.3.3.5 Descrambler

See 49.2.10.

201.3.3.6 64B/66B Decode

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

201.3.3.7 Idle Insertion

The receiving PCS inserts the Idle control characters to compensate for the removed FEC parity octets. The Idle Insertion function (see Figure 201–22) receives 72-bit vectors from the 64B/66B decoder and writes

them into the Idle Insertion FIFO (called FIFO_II) and reads 72-bit vectors from the FIFO_II and transfers them to the XGMII.

The Idle Insertion process receives 72-bit vectors at a slower rate than the nominal 25GMII rate due to the fact that the FEC parity blocks are removed by the FEC decoder and not put through the descrambler and 64B/66B decoder. The Idle Insertion process outputs 72-bit vectors at the nominal 25GMII rate. To match the input and output rates, the Idle Insertion process inserts additional 72-bit vectors containing Idle codes. The additional blocks are inserted between packets and not necessarily at the same locations where parity blocks have been removed.

The body of this subclause comprises state diagrams, including the associated definitions of variables, constants, and functions pertinent to the 25GBASE-PR-D, 25GBASE-PR-U and 25/10GBASE-PR-U PCS receivers. Should there be a discrepancy between a state diagram and descriptive text, the state diagram prevails. The notation used in the state diagrams in this clause follows the conventions in 21.5.

201.3.3.7.1 Constants

FEC_DSize
See 201.3.2.1.1.

FEC_PSize
See 201.3.2.1.1.

FIFO_II_SIZE
TYPE: 16-bit unsigned integer
This constant represents the size of Idle Insertion FIFO buffer. This buffer should be able to accommodate the number of 66-bit blocks sufficient to fill the gap introduced by removing the parity blocks from a maximum size MAC frame.
Value: 42

IDLE_VECTOR
TYPE: 72-bit binary
This constant represents a 72-bit vector containing Idle characters.

LBLOCK_R
See 49.2.13.2.1.

201.3.3.7.2 Variables

FIFO_II
TYPE: Array of 72-bit vectors received from 64B/66B decoder.
This FIFO is internal to the Idle Insertion process. Upon initialization, all elements of this array are filled with IDLE_VECTORs. FIFO_II is a zero-based array of size FIFO_II_SIZE (see 201.3.3.7.1).

RX_CLK
TYPE: Boolean
See 46.3.2.1.

rx_raw_in<71:0>
TYPE: 72-bit binary
Vector received from the output of the 64B/66B decoder. RXD<0> through RXD<31> for the second transfer are placed in rx_raw<40> through rx_raw<71>, respectively.

rx_raw_out<71:0>
TYPE: 72-bit binary
72-bit vector passed from the Idle Insertion process to XGMII. The vector is mapped to two XGMII transfers as follows:
Bits rx_raw<3:0> are mapped to RXC<3:0> for the first transfer;
Bits rx_raw<7:4> are mapped to RXC<3:0> for the second transfer;
Bits rx_raw<39:8> are mapped to RXD<31:0> for the first transfer;
Bits rx_raw<71:40> are mapped to RXD<31:0> for the second transfer.

VectorCount
TYPE: 16-bit unsigned integer
This variable tracks the number of 72-bit vectors stored in the FIFO_II.

201.3.3.7.3 Functions

T_TYPE(rx_raw)
See 49.2.13.2.3.

201.3.3.7.4 Messages

DECODER_UNITDATA.indicate(rx_raw_in<71:0>)
A signal sent by the PCS Receive process conveying the next received 72-bit vector.

DUDI
Alias for DECODER_UNITDATA.indicate(rx_raw_in<71:0>).

201.3.3.7.5 State diagrams

The PCS Idle Insertion function shall implement the state diagram as shown in Figure 201–22. Should there be a discrepancy between a state diagram and descriptive text, the state diagram prevails.

201.4 25GBASE-PR and 25/10GBASE-PR PMA

The 25GBASE-PR PMA is derived from the 10GBASE-R PMA defined in Clause 51. This clause specifies 10GBASE-R extensions necessary to support P2MP operation. The 25/10GBASE-PR PMA conceptually consists of a combination of transmit and receive functions specified for 25GBASE-PR and 10GBASE-PR defined in 76.3, as shown in Table 201–3.

Table 201–3—Derivation of PMA transmit and receive functions for 25GBASE-PR and 25/10GBASE-PR

PMA	Transmit function	Receive function
25GBASE-PR-U	As specified in Clause 51 with extensions defined in 201.4.1.	
25/10GBASE-PR-U	Identical to 10GBASE-PR-U. See 76.3.1.	Identical to 25GBASE-PR-U. See 201.4.1.
25GBASE-PR-D	As specified in Clause 51 with extensions defined in 201.4.2.	
25/10GBASE-PR-D	Identical to 25GBASE-PR-D.	Identical to 10GBASE-PR-D. See 76.3.2.

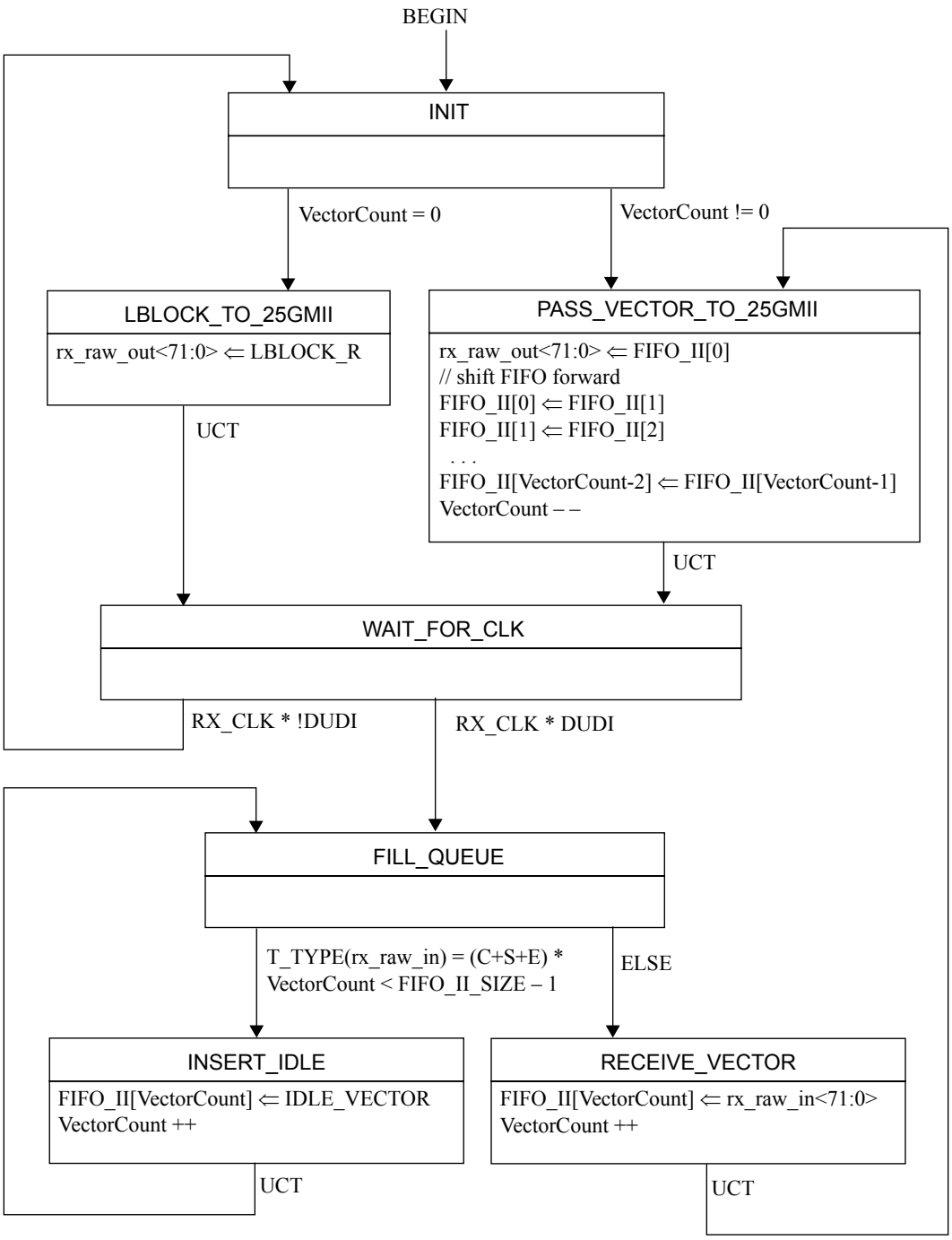


Figure 201–22—PCS Idle Insertion

201.4.1 Extensions for 25GBASE-PR-U and 25/10GBASE-PR-U

201.4.1.1 Physical Medium Attachment (PMA) sublayer interfaces

In addition to the primitives of Clause 51, the following primitive is defined:

PMA_SIGNAL.request(tx_enable)

This primitive is mapped to PMD_SIGNAL.request(tx_enable). It is generated by the PCS's Data Detector. The effect of reception of PMD_SIGNAL.request(tx_enable) is defined in 200.3.1.4.

tx_enable

The tx_enable parameter can take one of two values, ON or OFF.

201.4.1.2 Loop-timing specifications for ONUs

ONUs shall operate at the same time basis as the OLT, i.e., the ONU TX clock tracks the ONU RX clock, which in turn locks to OLT TX clock. Jitter transfer masks are defined in 200.7.

For the 25/10GBASE-PR-U devices, the received clock PMA_RX_CLK is 644.53125 MHz (25.78125 GBd/16), however, the transmit clock PMA_TX_CLK is 125 MHz (1.25 GBd/10). The loop timing is achieved by multiplying the PMA_RX_CLK by 32 and dividing by 165.

201.4.2 Extensions for 25GBASE-PR-D and 25/10GBASE-PR-D

201.4.2.1 CDR lock timing measurement for the upstream direction

CDR lock time (denoted T_{CDR}) is defined as a time interval required by the receiver to acquire phase lock on the incoming data stream. T_{CDR} is measured as the time elapsed from the moment when the electrical signal after the PMD at TP8, as illustrated in Figure 200-3, reaches the conditions specified in 200.7.15 for receiver settling time to the moment when the signal phase is recovered and jitter is maintained for a network with BER of no more than 10^{-3}.

A PMA instantiated in an OLT becomes synchronized at the bit level within 400 ns (T_{CDR}) after the appearance of a valid synchronization pattern (as defined in 201.3.2.5.2) at TP8.

201.4.2.1.1 Test specification

The test of the OLT PMA receiver T_{CDR} time assumes that there is an optical PMD transmitter at the ONU with well known T_{on} time as defined in 200.7.14, and an optical PMD receiver at the OLT with well-known $T_{receiver\_settling}$ time as defined in 200.7.15. After T_{on} + $T_{receiver\_settling}$ time, the parameters at TP8 reach within 15% of their steady state values, measure T_{CDR} as the time from the TX_ENABLE assertion, minus the known T_{on} + $T_{receiver\_settling}$ time, to the time the electrical signal at the output of the receiving PMA reaches up to the phase difference from the input signal of the transmitting PMA assuring BER of 10^{-3}, and maintaining its jitter specifications. The signal throughout this test is the synchronization pattern, as illustrated in Figure 201–13.

201.5 Protocol implementation conformance statement (PICS) proforma for Clause 201, Reconciliation Sublayer, Physical Coding Sublayer, and Physical Media Attachment for 100G-EPON[33]

201.5.1 Introduction

The supplier of a protocol implementation that is claimed to conform to Clause 201, Reconciliation Sublayer, Physical Coding Sublayer and Physical Media Attachment for 100G-EPON, shall complete the following protocol implementation conformance statement (PICS) proforma.

[33] *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.

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.

201.5.2 Identification

201.5.2.1 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).	

201.5.2.2 Protocol summary

Identification of protocol standard	IEEE Std 802.3-201x, Clause 201, Reconciliation Sublayer (RS), 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	

201.5.3 Major capabilities/options

Item	Feature	Subclause	Value/Comment	Status	Support
*OLT	OLT functionality	201.2.1	Device supports functionality required for OLT	O.1	Yes [ ] No [ ]
*ONU	ONU functionality	201.2.1	Device supports functionality required for ONU	O.1	Yes [ ] No [ ]

201.5.4 PICS proforma tables for Reconciliation Sublayer (RS), Physical Coding Sublayer (PCS), and Physical Media Attachment (PMA) for point-to-multipoint media, types 25GBASE-PR and 25/10GBASE-PR

201.5.4.1 Operating modes of OLT MACs

Item	Feature	Subclause	Value/Comment	Status	Support
OM1	Unidirectional mode	201.2.3	Device operates in unidirectional transmission mode	OLT:M	Yes [ ]
OM2	Dual-rate mode	201.2.2.3	Device operates in dual-rate mode	OLT:O	Yes [ ] No [ ]

201.5.4.2 ONU and OLT variables

Item	Feature	Subclause	Value/Comment	Status	Support
FS1	enable variable	65.1.3.1	True for ONU MAC, TRUE for OLT MAC if enabled, FALSE for OLT MAC if not enabled	M	Yes [ ]
FS2	mode variable	65.1.3.1	0 for ONU MAC, 0 or 1 for enabled OLT MAC	M	Yes [ ]
FS3	logical_link_id variable	76.2.6.1.1	Set to 0x7FFD until ONU MAC is registered Set to any value for enabled OLT MAC. Set to any value other then 0x7FFD for registered ONU MAC	M	Yes [ ]
FS4	multicast LLID support	76.2.6.1.1	Supports multicast LLID, multicast_link_id variable	O	Yes [ ] No [ ]

201.5.4.3 Preamble mapping and replacement

Item	Feature	Subclause	Value/Comment	Status	Support
PM1	CRC-8 generation	65.1.3.2	CRC calculation produces same result as serial implementation	M	Yes [ ] No [ ]
PM2	CRC-8 initial value	65.1.3.2	CRC shift register initialized to 0x00 before each new calculations	M	Yes [ ] No [ ]
PM3	SLD parsing	76.2.6.1.3.1	If SLD is not found then discard packet	M	Yes [ ] No[ ]
PM4	SLD replacement	76.2.6.1.3.1	Replace SLD with preamble	M	Yes [ ] No [ ]
PM5	LLID matching	76.2.6.1.3.2	If LLID does not match then discard packet	M	Yes [ ] No [ ]
PM6	multicast LLID matching	76.2.6.1.3.2	If multicast LLID matches accept the packet	*FS4 :M	Yes [ ] No [ ]
PM7	LLID replacement	76.2.6.1.3.2	Replace LLID with preamble	M	Yes [ ] No [ ]
PM8	Reserved LLID	76.2.6.1.3.2	registered ONU shall not transmit frames with a reserved LLID	M	Yes [ ] No [ ]
PM9	CRC-8 checking	65.1.3.3.3	If CRC does not match then discard packet	M	Yes [ ] No [ ]
PM10	CRC-8 replacement	65.1.3.3.3	Replace CRC with preamble	M	Yes [ ] No [ ]

201.5.4.4 Coding Rules

Item	Feature	Subclause	Value/Comment	Status	Support
C1	Encoder implements the code as specified	201.3.2.2		M	Yes [ ] No [ ]
C2	Decoder implements the code as specified	201.3.3.6		M	Yes [ ] No [ ]

201.5.4.5 Data detection

Item	Feature	Subclause	Value/Comment	Status	Support
DD1	Buffer depth	201.3.2.5.1	Depth sufficient to turn on laser and send laser synchronization pattern, Burst Delimiter pattern and a predefined number of Idle control character (receiver settle).	ONU:M	Yes [ ] No [ ]
DD2	OLT laser control	201.3.2.5.5	Always takes the value ON	OLT:M	Yes [ ] No [ ]
DD3	ONU State diagrams	201.3.2.5.7	Meets the requirements of Figure 201–15 and Figure 201–16	ONU:M	Yes [ ] No[ ]
DD4	OLT State diagrams	201.3.2.5.7	Meets the requirements of Figure 201–15 and Figure 201–16	OLT:M	Yes [ ] No[ ]

201.5.4.6 Idle control character deletion

Item	Feature	Subclause	Value/Comment	Status	Support
AIC1	Idle Deletion function implementation in ONU	201.3.2.1.5	Meets the requirements of Figure 201–9	ONU:M	Yes [ ] No [ ]
AIC2	Idle Deletion function implementation in OLT	201.3.2.1.5	Meets the requirements of Figure 201–8	OLT:M	Yes [ ] No [ ]

201.5.4.7 FEC requirements

Item	Feature	Subclause	Value/Comment	Status	Support
FE1	FEC Encoder	201.3.2.4	RS(255,223)	M	Yes [ ] No [ ]
FE2	FEC Decoder	201.3.3.3	RS(255,223)	M	Yes [ ] No [ ]

Item	Feature	Subclause	Value/Comment	Status	Support
FE3	Uncorrectable block indication	201.3.3.3	When activated, mark all 66-bit blocks in an uncorrectable block by setting all sync headers for the received payload blocks of the FEC codeword to the value of 00.	M	Yes [ ] No [ ]
FE4	Correctable codewords	201.3.3.3	Correct up to 16 symbols in a codeword and detect uncorrectable codewords	M	Yes [ ] No [ ]

201.5.4.8 FEC state diagrams

Item	Feature	Subclause	Value/Comment	Status	Support
SM1	Transmit	201.3.2.4.1	Meets the requirements of Figure 201–11	M	Yes [ ]
SM2	ONU synchronization	201.3.3.2.5	Meets the requirements of Figure 201–19	ONU:M	Yes [ ] No [ ]
SM3	OLT synchronization	201.3.3.1.4	Meets the requirements of Figure 201–17	OLT:M	Yes [ ] No [ ]
SM4	FEC decoding process	201.3.3.3.4	Meets the requirements of Figure 201–20	M	Yes [ ] No [ ]

201.5.4.9 PCS Idle Insertion

Item	Feature	Subclause	Value/Comment	Status	Support
PI1	Idle Insertion	201.3.3.7.5	Meets the requirements of Figure 201–22	M	Yes [ ] No [ ]

201.5.4.10 PMA

Item	Feature	Subclause	Value/Comment	Status	Support
BMC1	Loop timing	201.4.1.2	ONU RX clock tracks OLT TX clock	ONU:M	Yes [ ] No [ ]

201.5.4.11 Delay variation

Item	Feature	Subclause	Value/Comment	Status	Support
DV1	Delay variation	201.1.2	Combined delay variation through RS, PCS, and PMA sublayers is limited to 1 time_quantum	M	Yes [ ] No [ ]