92. Extensions of the Reconciliation Sublayer (RS) and Physical Coding Sublayer (PCS) / Physical Media Attachment (PMA) for 10GBASE-PR and 10/1GBASE-PRX for multipoint links and forward error correction

Editors Note: Proposed Changes to c92.2.2 only - all other sections not effected by this proposal. Task Force decissions in resolution of this comment:
Option 1a: Keep figures 92-4 and 92-5 as is; paired complementary transmitters and receiver.
Option 1b: Change figure 92-4 and 92-5 to illustrate OLT PCS (fig 92-4) and ONU PCS (fig 92-5).

92.1 Extensions of the Reconciliation Sublayer (RS) for point-to-point emulation

92.2 Extensions of the physical coding sublayer for data detection and forward error correction

Editors Note: The following section rreflect a reorganization of the clauses and text from Draft 1.0. Additional changes will be incurred, that are not shown here, due to Comment Resolutions agreed to at the Portland Interim meeting of the Task Force. Numerous comment resolutios are included in this presentation, these are listed below. In the clause text each change associated with a secondary comment is followed by the CommentID surrounded by bracket, this tag is identified by red italic font. (see example). In the event final disposition of a secondary comment disagrees with this proposal the disposition in the Comment Database takes precidence.
Text that is suggested by this change that has been added is shown in blue.
Text that has been deleted is shown as ~~Bluegray Strikethru~~.
Due to the extensive nature of the change, Changebars are not included.
An example of a secondary change: Commented text with ~~strikeout and~~ addition in line. *{999}*
The end result of this in Draft 1.1 would be: Commented text with addition in line.

***Primary Comment**:193, 404*

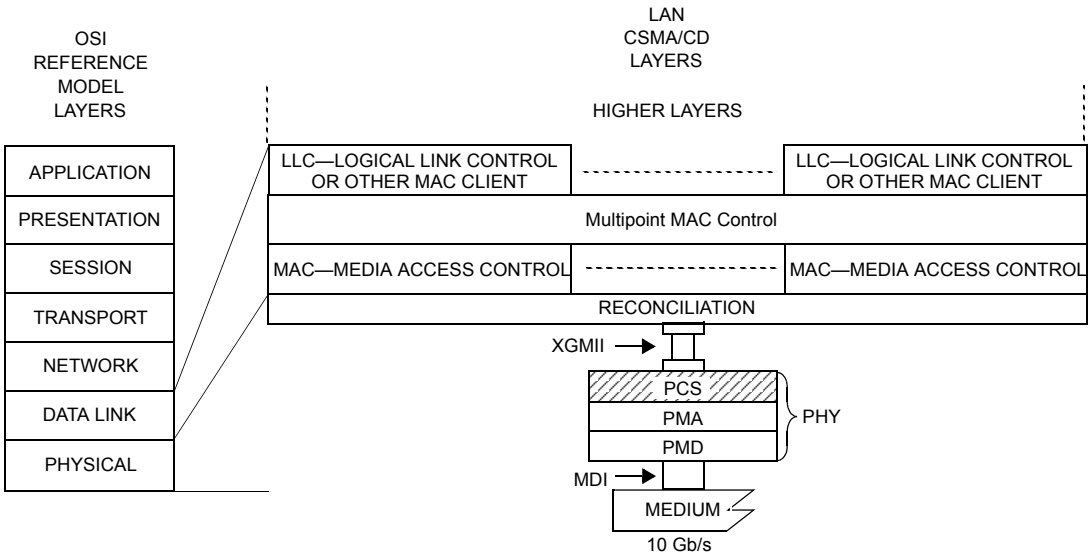
***Secondary Comments** included: 352,* 353, 8, 82

92.2.1 Overview

This subclause extends the physical coding sublayer described in Clause 46 to support burst mode operation of the point-to-multipoint physical medium. This subclause also specifies a forward error correction (FEC) mechanism to increase the optical link budget or the fiber distance. Figure 92–1 shows the relationship between the extended PCS sublayer and the ISO/IEC OSI reference model. Auto-Negotiation, as defined in Clause 37, establishes a point-to-point handshaking mechanism for allowing Physical Layer devices to achieve a highest common denominator link. The P2MP aspect of a 10G EPON network prohibits the use of the auto-negotiation protocol. The XAUI interface as defined in Clause 47 is prohibited from use in XEPON_PCS physical interfaces at ONU.

This subclause presents 10GBASE-PX PCS transmitter functions first and then addresses receiver functions. {193.404}

Editors Note: Figure 92-1 will be changed per comment 80.



GMII/XGMII = GIGABIT MEDIA INDEPENDENT INTERFACE
MDI = MEDIUM DEPENDENT INTERFACE
PHY = PHYSICAL LAYER DEVICE
PCS = PHYSICAL CODING SUBLAYER
PMA = PHYSICAL MEDIUM ATTACHMENT
PMD = PHYSICAL MEDIUM DEPENDENT

Figure 92–1—PCS location in the OSI protocol stack

Editors Note: Currently c92.2 includes two figures, 92-4 and 92-5, illustrating the PCS as OLT/ONT transmitter/receiver complementary pairs. These figures will be included in this Overview section with a short introduction added for each. It may be desirable to change the figures so they reflect 1) OLT PCS tx and rx 2) ONU PCS tx and rx. The TF must decide what is preferred:
Option 1a: Keep figures 92-4 and 92-5 as is; paired complementary transmitters and receiver.
Option 1b: Change figure 92-4 and 92-5 to illustrate OLT PCS (fig 92-4) and ONU PCS (fig 92-5).

92.2.2 10GBASE-PX Transmitters {193.404}

92.2.2.1 Alignment and Idle Deletion {193.404}

Editors Note: was 92.2.2.1Principle of operation

The ALIGNMENT / IDLE DELETION block receives ts-raw,71:0> data from the XGMII interface. If the start control code is in lane 4 the burst will be shifted to align the start to lane 0. If the minimum IPG has been transmitted after a frame and 14 tx_raw<71:0> transfers have occurred without deleting IDLE then 2 IDLE vectors shall be deleted for every 28 vectors transmitted.

92.2.2.2 64/66b Encoding {193.404}

92.2.2.3 Scrambler {193.404}

92.2.2.4 FEC Encoding {193.404}

Editors note: was 92.2.3.1 FEC code

The FEC code used for 10GBASE-RX links is a linear cyclic block code - the Reed-Solomon code (255, 223, & 16).

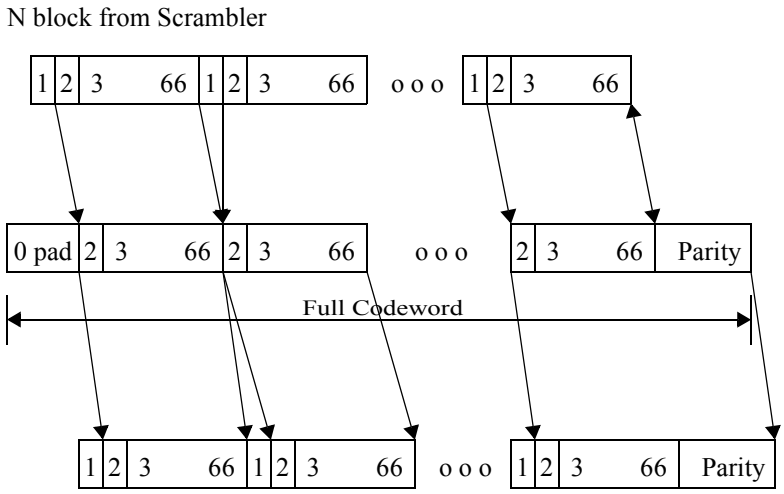


Figure 92–2—FEC Padding

92.2.2.4.1 FEC Algorithm (RS((255, 223)) {193.404}

Editors note: provide general describition of RS(255,223) here.

Editors note: was 92.2.3.2 FEC frame format

~~The frame format of a FEC coded 10G EPON Ethernet frame is herein described.~~

92.2.2.4.2 Parity Calculation {193.404}

Editors note: was 92.2.3.2.1 Placing parity octets

Padding of FEC codewords and appending of FEC parity bytes in the 10GBASE-RX PCS transmitter is illustrated in Figure 92–2. Ethernet packets are received from the PCS scrambler in blocks of 66 bits. ~~The data is partitioned into 27 blocks. Each partition of 27 blocks is then encoded using the RS(255,223) FEC encoder, which results in an additional 4 parity symbols for each block. The block, minus any padding, plus the associated 4 parity symbols form the @tbd@ byte FEC codeword. The additional 4 parity blocks, which are generated from this encoding process for each block, are gathered and added at the end of FEC code word to be transmitted. note that parity is not calculated over the first bit of each 66 bit from the scramble as this bit is redundant. However this first bit is always transmitted over the link.~~ The FEC encoder accumulates 27 66b blocks and removes the first bit of each block (ie. the redundant sync bit of the 66b word). The FEC encoder then prepends 29 "0" padding bits to the 27 blocks form the 233 byte data portion of a FEC codeword. The data is then FEC-encoded, which results in an additional 4 parity symbols for each block - completing the 255-byte Reed-Solomon codeword.

92.2.2.4.3 FEC Transmission Block Formating {193.404}

Editors Note: was92.2.3.2.2 Codeword Padding

~~When dividing the data into FEC payloads there might be a case where the last partition is shorter than the required @tbd@ symbols. In this case sufficient padding of zero bits is added to the font of the payload to fill the payload container. Parity is then calculated as defined in @tbd@. When transmitted across the link the padding is stripped to reduce transmission of the null information.~~

As shown in figure 92-10, after the Reed-Solomon codeword has been computed, the FEC encoder constructs the transmittable FEC frame with the original sequence of 27 66bit 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 66b codeword. Finally the four 66bit parity blocks are appended following the 27 66bit data blocks and transmitted to the PON.

92.2.2.5 Data Detector and Burst Mode Considerations (ONU only) {193.404}

Editors Note: was 92.2.2Burst-mode operation

To avoid spontaneous emission noise from near ONUs obscuring the signal from a distant ONU, the ONUs' lasers must be turned off between transmissions. To control the laser, the US PCS is extended to detect the presence of transmitted data and generate the PMD_SIGNAL.request(tx_enable) primitive to turn the laser on and off at the correct times. This function is performed by the Data Detector shown in the functional block diagram in Figure 92–4.

The DATA DETECTOR contains a delay line (FIFO buffer) storing code-groups to be transmitted. The length of the FIFO buffer 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 characters to be transmitted. The laser synchronization pattern allows the receiving optical detector to adjust its AGC ($T_{receiver\_settling}$) and synchronize its receive clock (T_{cdr}). The Burst Delimiter allows the receiver to easily identify the Start of Data. The IDLE characters are used to synchronize the scrambler and start of packet delineation.

~~To increase burst efficiency it is desirable to align the start of a burst to Lane 0 of the XGMII interface.~~ Two consecutive XGMII transfers provide eight characters that are encoded into one 66-bit transmission block. To increase burst efficiency the start of a burst is aligned to the first of these two transfers. {352} If this is not done the burst transmitter may occasionally be required to transmit and extra 4 bytes of data, causing the data burst to extend into the next grant. To ensure the start of a burst aligns to lane 0 of the XGMII the PCS is extended to allow removal of leading IDLE control codes.

Upon initialization, the FIFO buffer is filled with ~~/I/ ordered_sets~~ idle control characters and the laser is turned off. When the first code-group that is not ~~/I/~~ idle arrives at the buffer, the Data Detector sets the PMD_SIGNAL.request(tx_enable) primitive to the value ON, instructing the PMD sublayer to start the process of turning the laser on (see Figure 92–4). {353}

When the buffer empties of data (i.e., contains only ~~/I/ ordered_sets~~ idle control characters), the Data Detector sets the PMD_SIGNAL.request(tx_enable) primitive to the value OFF, instructing the PMD sublayer to start the process of turning the laser off. Between packets, ~~/I/ or /R/ ordered_sets~~ idle control characters will arrive at the buffer. If the number of these ~~/I/ or /R/ ordered_sets~~ idle control characters is insufficient to fill the buffer then the laser is not turned off. {353}

Figure 92–5 shows the relationship of filling the buffer and the generation of laser_control. In the OLT, the laser always remains turned on. Correspondingly, therefore the OLT's Data Detector does not need a delay line or buffer in the data path for this purpose.

The ONU burst transmission begins with a synchronization pattern (binary ~~1010~~ 0101...) which facilitates receiver clock recovery and gain control at the OLT. To facilitate ~~byte level~~ FEC codeword synchronization the ONU transmits a 66 bit ~~Start of Data (SOD) delimiter composed of a Barker link sequence (see Figure 92–6)~~ BURST_DELIMITER (see Figure 92–7). When received at the OLT the delimiter allows ~~byte~~ FEC codeword alignment of the incoming data stream, even in the presence of bit errors. The ~~SOD~~ BURST_DELIMITER is followed by ~~@tbd (two)@~~ one IDLE block~~s which are used to synchronize the SCRAMBLER at the OLT.~~ to provide IPG at the OLT. These two IDLE blocks are part of the FEC codeword. {8}

92.2.2.6 Gearbox {193.404}

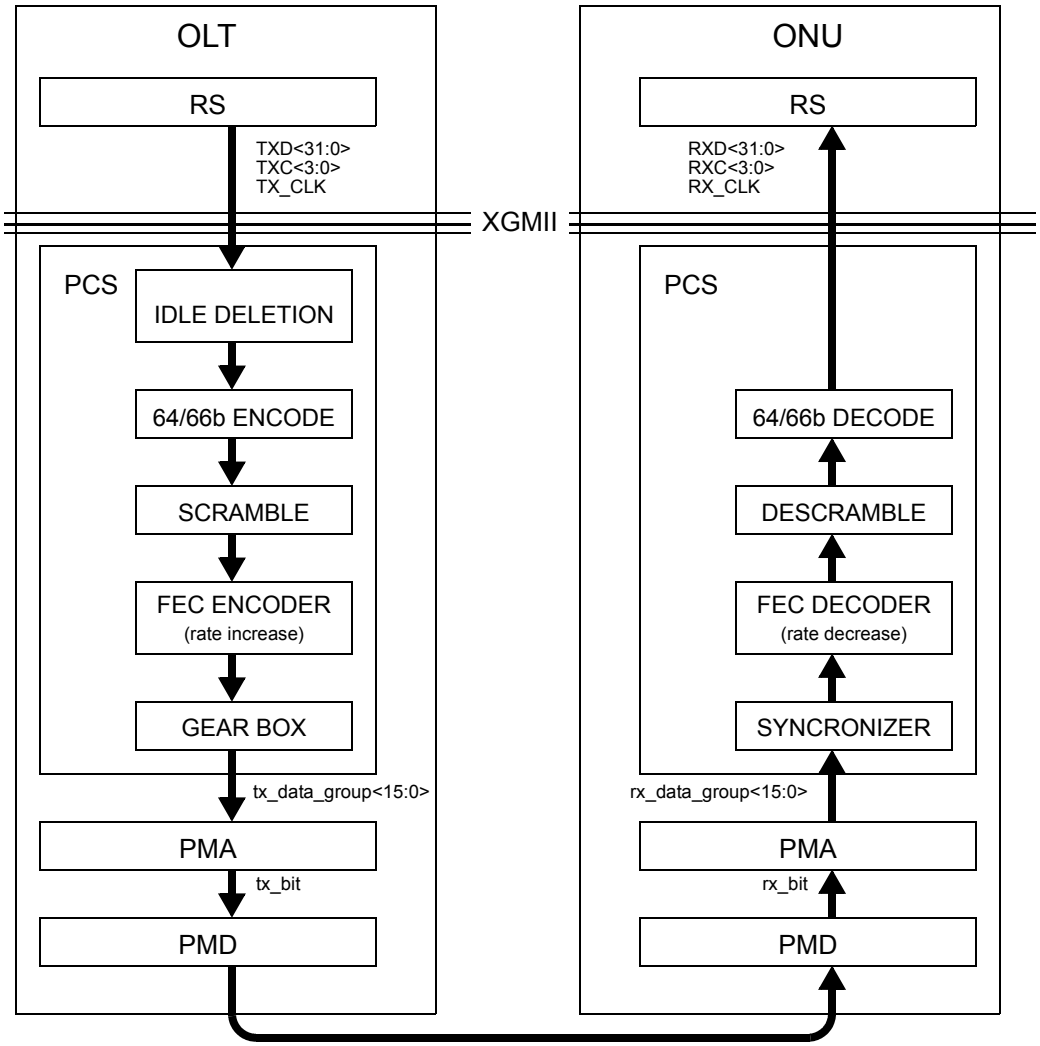


Figure 92–3—PCS Extension functional block diagram, DS

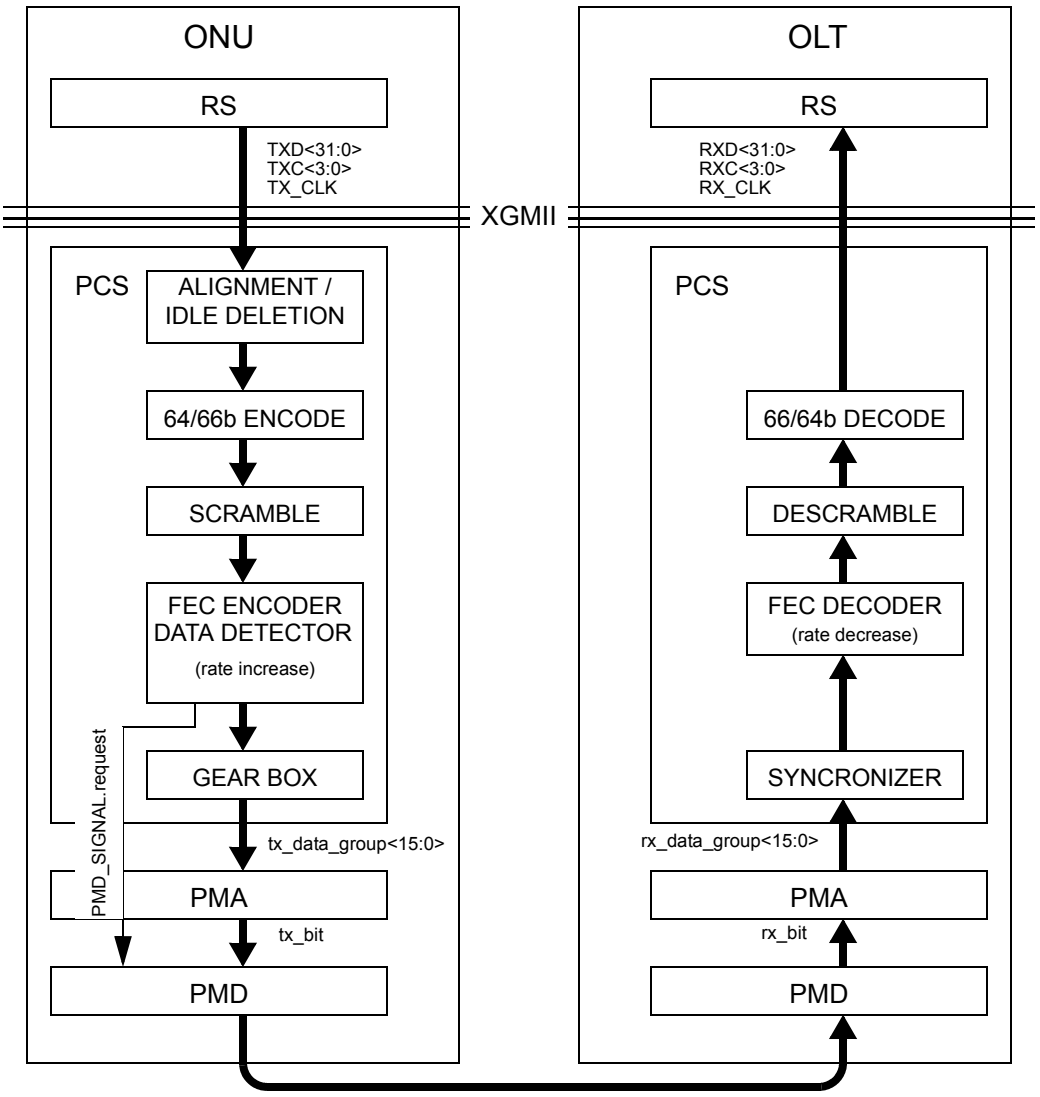


Figure 92–4—PCS Extension functional block diagram, US

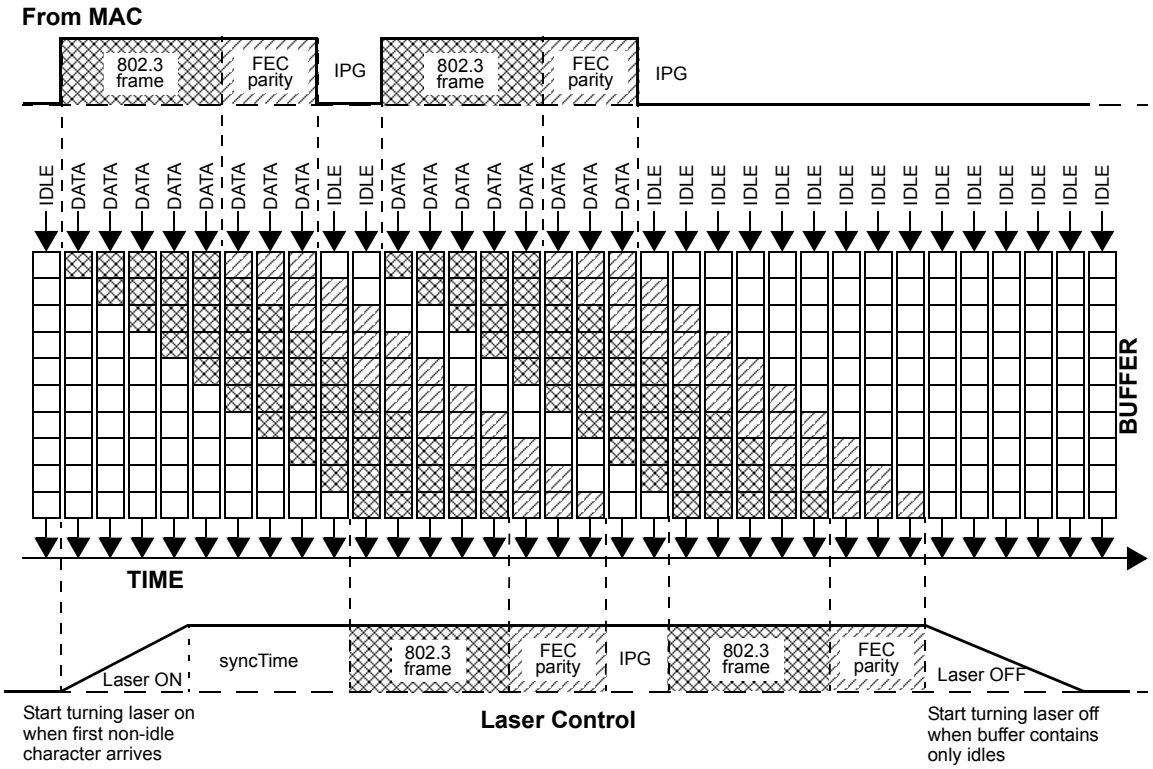


Figure 92–5—Laser control as a function of buffer fill

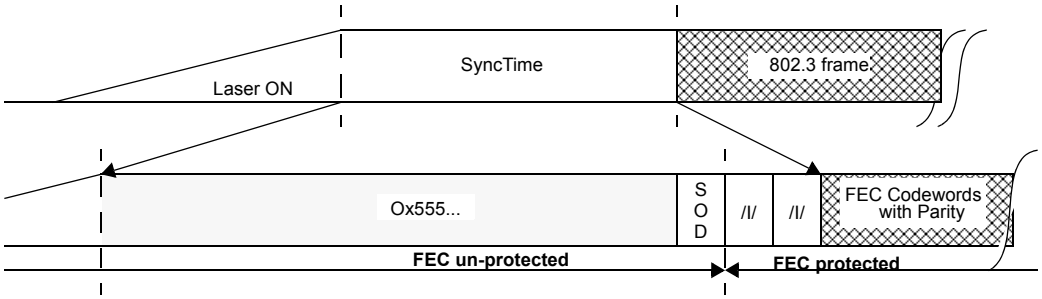


Figure 92–6—SyncTime Detail

92.2.2.7 Detailed functions and state diagrams {193.404}

The body of this subclause comprises state diagrams, including the associated definitions of variables, constants, and functions pertanent to the 10GBASE-X PCS transmitters.{193.404} 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. Variables in a state diagram with default values evaluate to the variable default in each state where the variable value is not explicitly set.

92.2.2.7.1 Constants

FecRatio
TYPE: 16-bit unsigned
The number of 72-bit vectors to transmit before deleting idle.
Default: 14

LsrOffBound
TYPE: 16-bit unsigned
The number of 72-bit vectors needed in the FIFO before laser is turned off.
Default: tbd

MinIpg
TYPE: ??
The number of 72-bit vectors consisting of IDLE that constitute minimum IPG.
Default: 2

BURST_DELIMITER
TYPE: ??
A 66-bit value used to find the beginning of a FEC codeword
Default: ??

SYNC_LENGTH
TYPE: ??
Required number of sync blocks per burst. The value of this constant is derived from Sync-Time parameter passed from the OLT to ONUs. See 64.3.3.2 for details. {82}
Default: ??

92.2.2.7.2 Variables

BEGIN
TYPE: Boolean
This variable is used when initiating operation of the state machine. It is set to true following initialization and every reset.

DelayBound
TYPE: 16-bit unsigned
DEFAULT VALUE: 00-6A (106 code-groups = 848 ns)
This represents the delay sufficient to initiate the laser and to stabilize the receiver at the OLT. The default value of DelayBound is based on default values of laserOnTime (64.3.5.1) and SyncTime (64.3.3.2). This variable is only used by the ONU.

dtx_code-group
A 10-bit vector representing one code-group, as specified in Tables 36–1a through 36–2, which has been prepared for transmission by the Data Detector process. This vector is conveyed to the PMA as the parameter of a PMA_UNITDATA.request(dtx_code-group) service primitive. The element dtx_code-group<0> is the first bit transmitted and dtx_code-group<9> is the last bit transmitted.

Halfshift
TYPE: Boolean
True if data is currently shifted by one word. False if data is not currently shifted.

laser_control
This variable represents the status of the laser. The value on corresponds to the laser being turned on, and the value off corresponds to laser being off.
TYPE: boolean.

tx_code-group
A 10-bit vector of bits representing one code-group, as specified in Table 36-1a or Table 36-2, which has been prepared for transmission by the PCS Transmit process. The element tx_code-group<0> is the first tx_bit transmitted; tx_code-group<9> is the last tx_bit transmitted.

Wp

TYPE: 32-bit binary
Holds the previously transmitted XGMII word.
Default: na

Wn
TYPE: 32-bit binary
Holds the next XGMII word to be transmitted (what is leftover from the most recent ts_raw).
Default: na

W0,W1
TYPE: 32-bit binary
Hold a single XGMII transfer that is written into tx_raw before being passed to the encoder.
Default: na

IdleBlockCount
TYPE: ??
The number of consecutive non-data blocks ending with the most recently received block. The non-data blocks are represented by sync header 10.
Default: ??

ProtectedBlockCount
TYPE: ??
The number of blocks added to a payload of a current FEC codeword. After reaching the full payload size (28), this variable is reset to 0.
Default: ??

UnprotectedBlockCount
TYPE: ??
The similar to ProtectedBlockCount, but counts payload size outside of grant. This variable is used to determine a number of overhead blocks to be added to maintain proper PHY rate outside the FEC-protected grant area.
Default: ??

SyncBlockCount
TYPE: ??
The number of synchronization blocks transmitted in current burst.
Default: ??

92.2.2.7.3 Functions

IsIdle(tx_code-group)
This function is used to determine whether tx_code-group is a code-group in /I/, the IDLE ordered_set, or /C/, the Configuration ordered_set. This function returns true if tx_code-group is /K28.5/ or any code-group that follows a /K28.5/ or any two consecutive /D/ code-groups that follow /K28.5/D21.5/ or /K28.5/D2.2/. Otherwise, the IsIdle function returns false.

FIFO.RemoveHead()
This function removes the first code-group from the FIFO buffer and advances all remaining code-groups one position ahead. This function returns the 10-bit vector representing the removed code-group.

FIFO.Append(tx_code-group)
This function appends a new 10-bit vector to the end of the FIFO buffer.

TransmitBlock(tx_block<65:0>)
This function passes its argument to the GearBox for further transmission to the PMA.

ReceiveNextBlock()
This function is used to receive the next 66-bit block from the scrambler. RecevieNextBlock() is a blocking function – it does not return until a next block becomes available and is stored in the tail position in the FIFO, as represented by the code FIFO[N] = tx_data<65:0>.

```
ReceiveNextB
{
    // shift FIFO forward
```

```
        FIFO[0] = FIFO[1]
        FIFO[1] = FIFO[2]
                  ...
        FIFO[N-1] = FIFO[N]
        // receive next 66-bit block from scrambler
        FIFO[N] = tx_block<65:0>
    }
```

92.2.2.7.4 Messages

PMD_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 92–8).

PUDR

Alias for PMA_UNITDATA.request(tx_code-group<9:0>).

92.2.2.7.5 Counters

IdleLength

This counter represents the length of the consecutive interval of idles ending with the most recent tx_code-group. If the most recent tx_code-group represents a non-idle character, the IdleLength is reset to 0.

TYPE:32-bit unsigned

IdleCount

TYPE: 16-bit unsigned

Counts the number of 72 bit vectors containing idle or other control vectors.

VectorCount

TYPE: 16-bit unsigned

Counts the number of 72 bit vectors transmitted.

92.2.2.7.6 State Diagrams

The Alignment and Idle Detection function shall be implemented in the PCS as depicted in Figure 92–7, including compliance with the associated state variables as specific in 92.2.2.7.

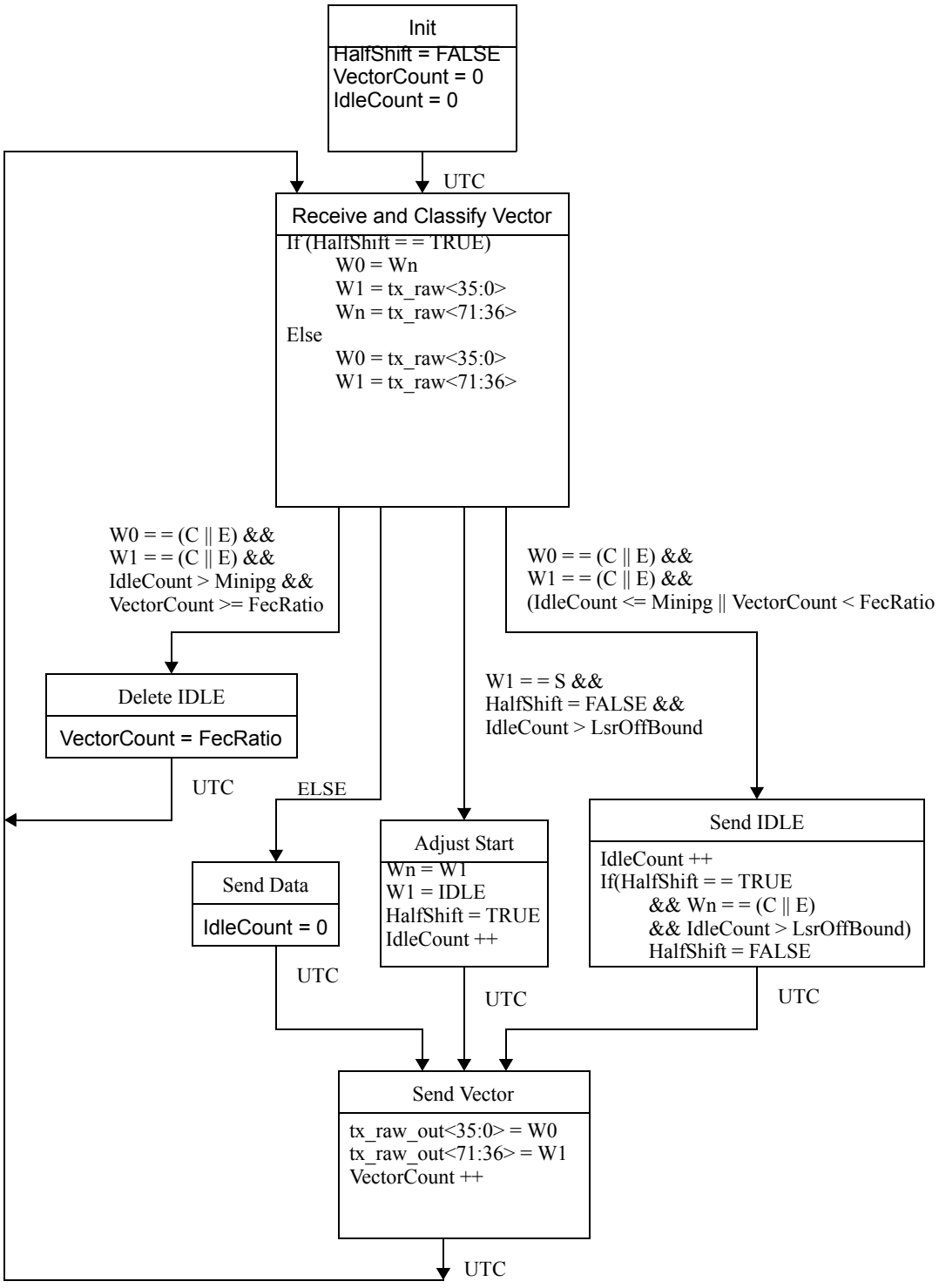


Figure 92–7—ONU Alignment / Idle Detector state diagram

The Data Detector shall be implemented for an ONU as depicted in Figure 92–8, including compliance with the associated state variables as specified in 92.2.2.7.

92.2.3 10GBASE-PX Receivers {193.404}

92.2.3.1 Synchronizer {193.404}

The codeword synchronization function receives data via 16-bit PMA_UNITDATA.request primitive.

The synchronizer shall form 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 92-##). It obtains lock to the 31*66-bit blocks in the bit stream using the sync headers and outputs 66-bit blocks, with the codeword structure being indicated by a locally generated sync header pattern. Lock is obtained as specified in the codeword lock state machine shown in Figure 92-##.

The incoming sync header pattern is 27 conventional (clause 49) sync headers (01 or 10), and then 00, 11, 11, and 00. The state machine performs a search for this pattern, and when it finds a perfect match of two full codewords (62 blocks), it then asserts codeword lock.

When codeword lock is true, the decoder guarantees that the sync header of the last block in the codeword will be "11", and that no other sync header will have this pattern, even in the face of errors. This is achieved by forcing the first 27 sync headers to be conventional headers, and forcing the last four headers to be 00, 00, 00, and 11. This locally forced pattern then allows the subsequent FEC decoder logic to find the last block in the codeword with a trivial match of the sync header to 11.

When in codeword lock, the state machine continues to check for sync header validity. If 16 or more sync headers in a codeword pair (62 blocks) are invalid, then the state machine deasserts codeword lock.

92.2.3.2 FEC Decoder {193.404}

The FEC decoder corrects or confirms the correctness of the 27 66b blocks contained in the frame based on the four 66b blocks of parity information. The decoder then forwards the 66bit data blocks to the descrambler and discards the parity blocks.

If the FEC decoder determines that the frame is not correctable (due to an excess of symbols containing errors), the data blocks are nevertheless passed to the descrambler to maintain descrambling synchronization. The data blocks of the frame must then be replaced by /E/ blocks before being passed to the PCS.

92.2.3.3 Descrambler {193.404}

92.2.3.4 66/64b Decode {193.404}

92.2.3.5 Idle Insertion {193.404}

92.2.3.6 Detailed functions and state diagrams {193.404}

The body of this subclause comprises state diagrams, including the associated definitions of variables, constants, and functions pertanent to the 10GBASE-X 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. State diagram variables follow the conventions of 21.5.2 except when the variable has a default value. Variables in a state diagram with default values evaluate to the variable default in each state where the variable value is not explicitly set.

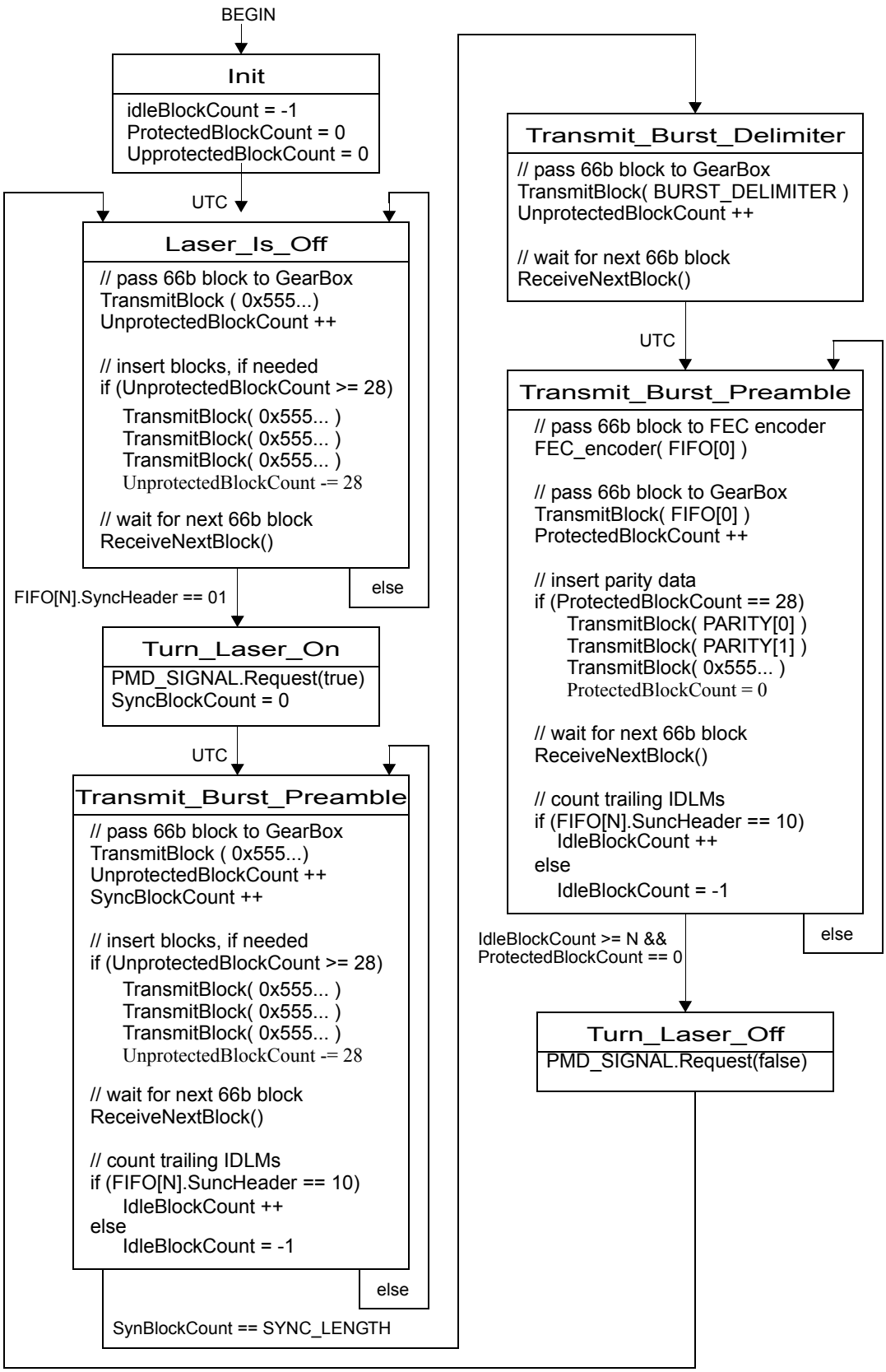


Figure 92–8—ONU data decoder state diagram

92.2.3.6.1 Constants {193.404}

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

SH_CW_PATTERN[0..30]
: 31 element array of codeword sync header bit counts, where each element is set to the value 1 except for:

SH_CW_PATTERN[27]=0
SH_CW_PATTERN[28]=2
SH_CW_PATTERN[29]=2
SH_CW_PATTERN[30]=0

92.2.3.6.2 Variables {193.404}

All the relevant variables defined in 49.2.13.2.2 are inherited. In addition, the following items are defined.

sh_valid[i]
: Boolean 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.

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

92.2.3.6.3 Functions {193.404}

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

Force(i)
: Forces the sync header to the state that preserves FEC frame lock. Note that for parity blocks, the pattern is known a priori. For payload blocks, the first bit is forced to be the complement of the second bit. While this may duplicate a bit error, it will not propagate, as the FEC decoder discards the first bit before decoding.

```
Force(i)
    {
        If ( cword_lock == true )
            If ( i>26 )
                If ( i==30 )
                    rx_coded<0>=1
                    rx_coded<1>=1
                else
                    rx_coded<0>=0
                    rx_coded<1>=0
            else
                rx_coded<0>=!rx_coded<1>
    }
```

92.2.3.6.4 Messages {193.404}

92.2.3.6.5 Counters {193.404}

All the relevant counters defined in 49.2.13.2.4 are inherited.

92.2.3.6.6 State Diagrams {193.404}

The Lock state machine shown in Figure 92-## determines when the PCS has obtained lock to the received data stream. The BER is determined by the FEC decoder function, and so a separate state machine is not required.

Editors Note: Add the figure, as provided in presentation 3av_0801_remein_3.pdf