201.2.2.9.2 When generated

The PMA PHY Control function generates PMA_PCSDATAMODE.indication messages continuously.

201.2.2.9.3 Effect of receipt

Upon receipt of this primitive, the PCS performs its transmit function as described in 201.3.2.2 for HS_TX and 201.4.2.2 for LS_TX.

201.3 Physical Coding Sublayer (PCS) functions, high speed path (HS_PATH)

201.3.1 PCS service interface (XGMII)

The PCS service interface allows the MultiG+100MBASE-T1/V1 PCS to transfer information to and from a PCS client. The PCS service interface is precisely defined as the 10 Gigabit Media Independent Interface (XGMII) in Clause 46.

201.3.2 PCS functions

The PCS comprises one PCS Reset function and two simultaneous and asynchronous operating functions, one of which is the HS_PATH and the other is the LS_PATH. This subclause discusses the HS_PATH. The HS_PATH PCS operating functions are the HS_TX PCS Transmit in the PHY_S device, and the HS_RX PCS Receive in the PHY_D device. All operating functions start immediately after the successful completion of the PCS Reset function.

The PHY_S PCS reference diagram, Figure 201–7 and the PHY_D PCS diagram, Figure 201–8, show how the two operating functions relate to the messages of the PCS-PMA interface. Connections from the management interface (signals MDC and MDIO) to other layers are pervasive and are not shown in Figure 201–7 and Figure 201–8.

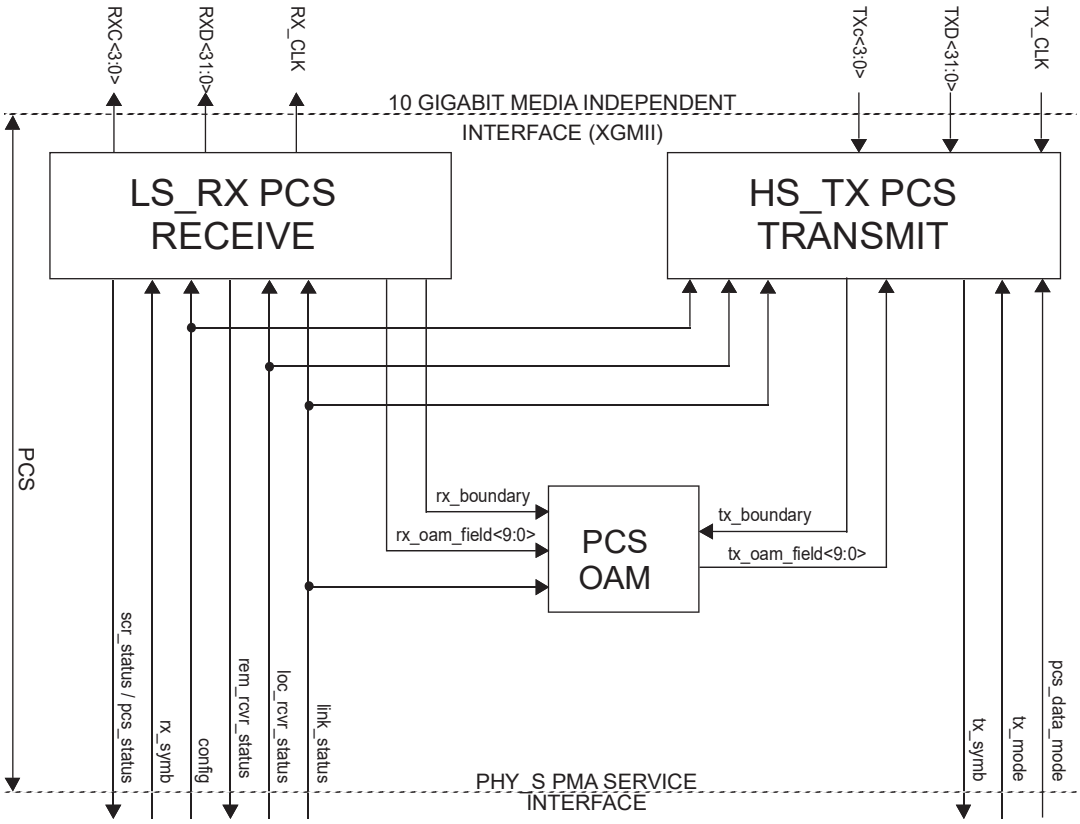


Figure 201–7—PHY_S PCS reference diagram

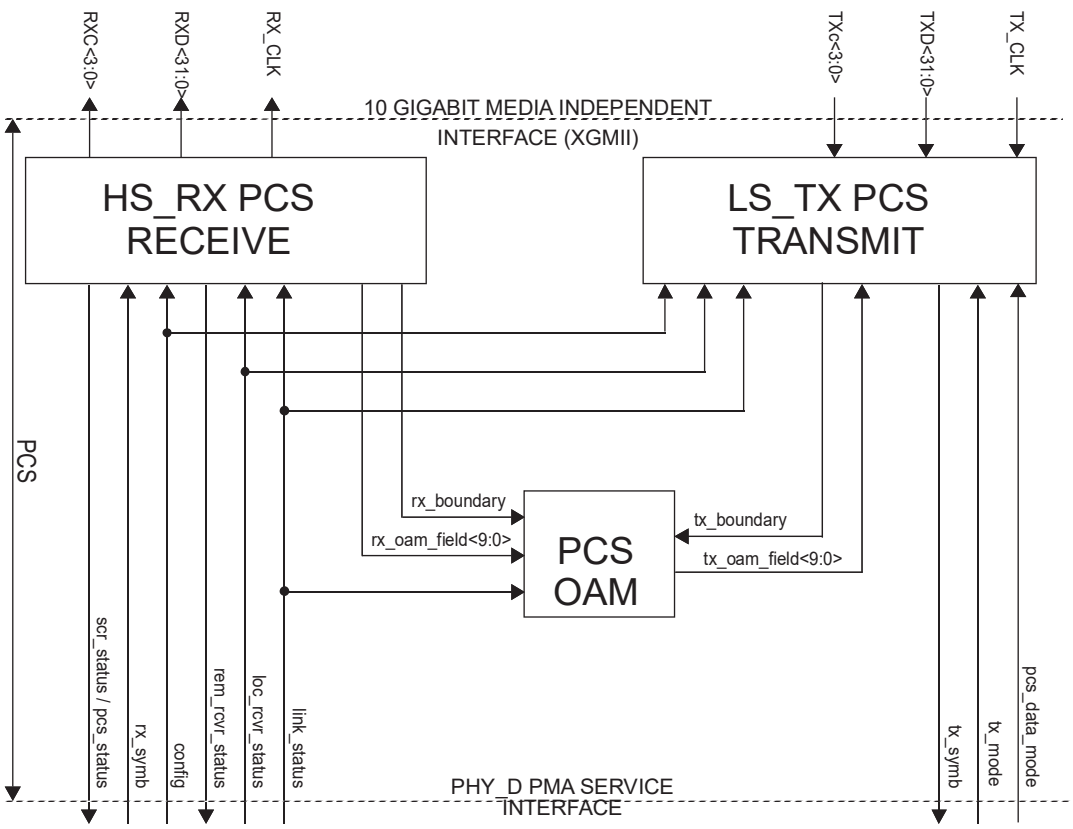


Figure 201–8—PHY_D PCS reference diagram

201.3.2.1 PCS Reset function

PCS Reset initializes all PCS functions. The PCS Reset function shall be executed whenever one of the following conditions occur:

a) Power on (see 201.3.6.2.2).
b) The receipt of a request for reset from the management entity.

PCS Reset sets pcs_reset = TRUE while any of the above reset conditions hold true. All state diagrams take the open-ended pcs_reset branch upon execution of PCS Reset. The reference diagrams do not explicitly show the PCS Reset function.

The control and management interface shall be restored to operation within 10 ms from the setting of bit 3.2322.15.

201.3.2.2 PCS Transmit function

The PCS Transmit function shall conform to the PCS 64B/65B Transmit state diagram in Figure 201–17 and to the PCS Transmit bit ordering in Figure 201–11.

When communicating with the XGMII, the MultiG+100MBASE-T1/V1 PCS uses a four octet-wide, synchronous data path, with packet delimiting being provided by transmit control signals and receive control

signals. Alignment of pairs of XGMII transfers to 64B/65B blocks is performed in the PCS. The PMA sublayer operates independently of PCS block, RS-FEC frames, and higher-layer packet boundaries. The PCS provides the functions necessary to map packets between the XGMII format and the PMA service interface format.

After mapping the XGMII transfers to 64B/65B blocks, the subsequent functions of the PCS Transmit process take L groups of 50 65B blocks and append a 10-bit OAM field to each group. This forms the input to an L-interleaved RS-FEC which adds L × 340 parity bits. The resulting L × 3600 bits are then scrambled. These bits are then mapped, two at a time, into a PAM4 symbol for 10G and one at a time into a PAM2 symbol for 2.5G and 5G. Transmit data-units are sent to the PMA service interface via the PMA_UNITDATA.request primitive.

In each symbol period, when communicating with the PMA, the PCS Transmit generates a PAM4 symbol for 10G and a PAM2 symbol for 2.5G and 5G that is transferred to the PMA via the PMA_UNITDATA.request primitive. The symbol period, *T*, is 1000 / (5.625 × *S)* ps. See Table 201–1 for the definition of *S*.

The operation of the PCS Transmit function is controlled by the PMA_TXMODE.indication message received from the PMA PHY Control function.

If a PMA_TXMODE.indication message has the value SEND_Z, PCS Transmit shall pass a vector of zeros at each symbol period to the PMA via the PMA_UNITDATA.request primitive.

If a PMA_TXMODE.indication message has the value SEND_T, PCS Transmit shall generate a sequence (T_n) defined in 201.3.5.1 to the PMA via the PMA_UNITDATA.request primitive. These code-groups are used for training mode and only transmit the values {–1, +1}.

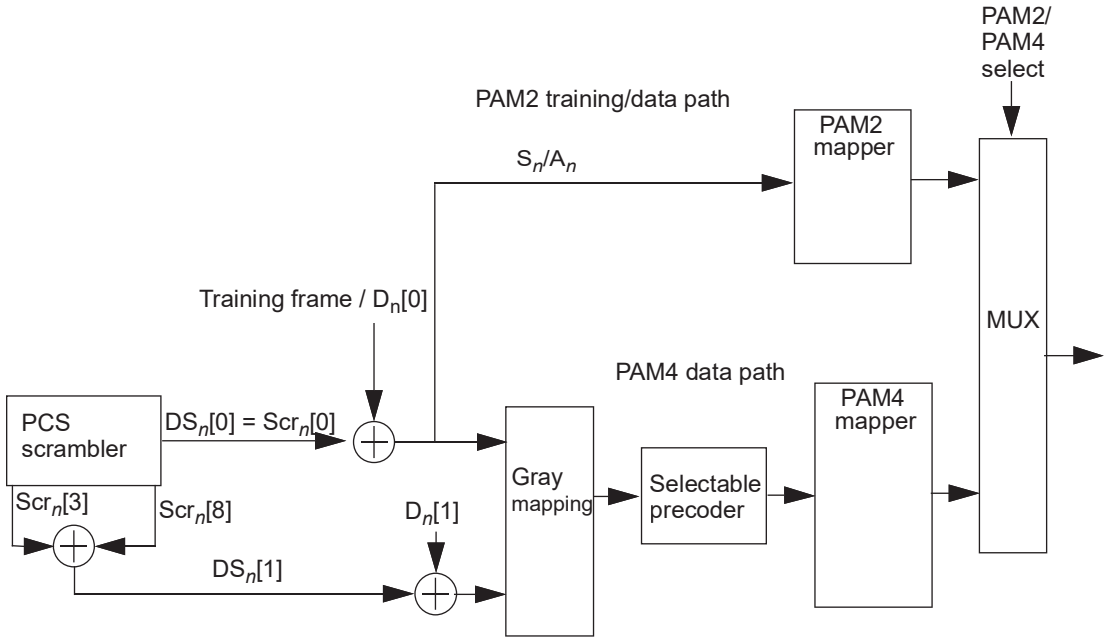
During training mode an Infofield is transmitted at regular intervals containing messages for startup operation. By this mechanism, a PHY indicates the status of its own receiver to the link partner and makes requests for remote transmitter settings. (See 201.5.2.6.)

If a PMA_TXMODE.indication message has the value SEND_N, the PCS is in the normal mode of operation and the PCS Transmit function shall use a 65B coding technique to generate, at each symbol period, code-groups that represent data or control. During transmission, the 50 blocks of 65B encoded bits are appended with a 10-bit OAM field to form the RS-FEC input frame. During data encoding, PCS Transmit utilizes L-interleaved (L = 1, 2, or 4) Reed-Solomon encoders to generate and append 340 parity check bits to form 3600-bit (360,326) RS-FEC frames that are interleaved into an L-interleaved RS-FEC superframe.

Each RS-FEC input superframe consists of 3260 × L bits, or 326 × L Reed-Solomon message symbols. The interleaving function is integrated with the RS-FEC encoding, applying a round-robin interleaving scheme and distributing the 10-bit Reed-Solomon message symbols into L RS-FEC encoders. After encoding, the RS-FEC frames from each encoder are recombined into one single interleaved RS-FEC superframe, which consists of 360 × L symbols, or 3600 × L bits. The bits of the RS-FEC superframe are then scrambled by the PCS using an additive scrambler, encoded in PAM4 symbols, and transferred to the PMA.

L is called the interleaving depth, and the possible choices of L are 1, 2, and 4. The interleaver settings requested in each direction of transmission may be different, and the value of L used by the transmitter is determined by the link partner and signaled during the PAM2 training mode Infofield exchange.

The block diagram of HS_TX PCS Transmit functions is shown in Figure 201–9.



NOTE 1—The S_n are from the training frame, and the A_n are from the 2.5G and 5G HS_PATH when the PAM2 mapper is used.

Figure 201–9—HS_TX PCS Transmit function block diagram

201.3.2.2.1 Use of blocks

The PCS maps XGMII signals into 65-bit blocks inserted into an RS-FEC frame, and vice versa, using a 65B RS-FEC coding scheme. The PAM2 PMA training frame synchronization allows establishment of RS-FEC frame and 65B boundaries by the PCS Synchronization process. Blocks and frames are unobservable and have no meaning outside the PCS. The PCS functions ENCODE and DECODE generate, manipulate, and interpret blocks and frames as provided by the rules in 201.3.2.2.2.

201.3.2.2.2 65B RS-FEC transmission code

The PCS uses a transmission code to improve the transmission characteristics of information to be transferred across the link and to support transmission of control and data characters.

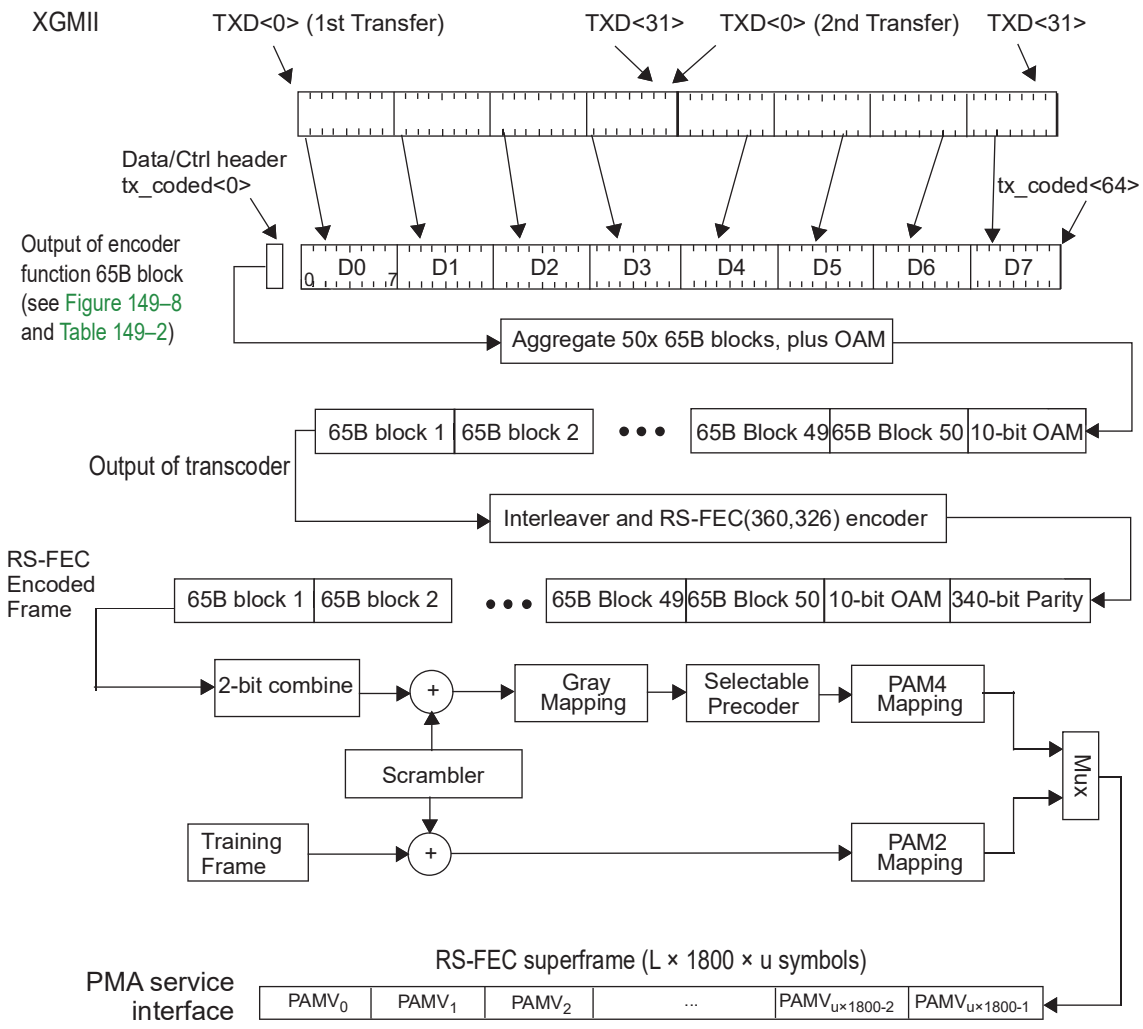
The relationship of block bit positions to XGMII, PMA, and other PCS constructs is illustrated in Figure 201–10 for 10G transmit, Figure 201–11 for 2.5G and 5G transmit, Figure 201–12 for 10G receive, and Figure 201–13 for 2.5G and 5G receive. These figures illustrate the processing of a multiplicity of blocks containing 8 data octets. See 149.3.2.2.4 for information on how blocks containing control characters are mapped.

201.3.2.2.3 Notation conventions

See 149.3.2.2.3.

201.3.2.2.4 Block structure

See 149.3.2.2.4.



NOTE 1—This figure shows the mapping from the XGMII to a 64B/65B block for a block containing eight data characters.
NOTE 2—Figure shown for L = 1.
NOTE 3—For PAM2 path, V=2 and u=2. For PAM4 path, V=4 and u=1.

Figure 201–10—PCS 10G Transmit bit ordering for data mode and training mode, HS_TX

201.3.2.2.5 Control codes

See 149.3.2.2.5.

201.3.2.2.6 Ordered sets

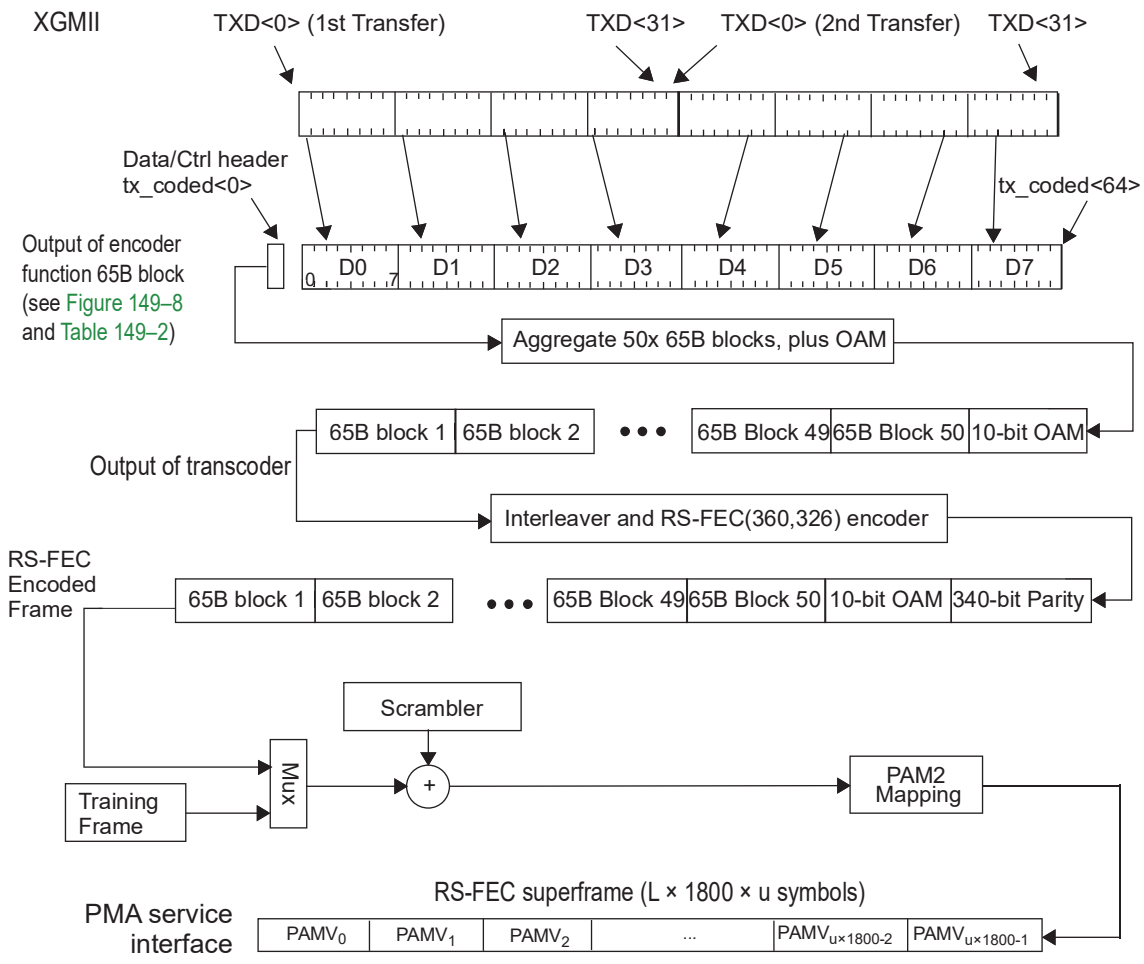
See 149.3.2.2.6. LPI, /LI/ is not used by MutltiG+100MBASE-T1/V1 PHYs.

201.3.2.2.7 Idle (/I/)

See 149.3.2.2.7.

201.3.2.2.8 Start (/S/)

See 149.3.2.2.9.



NOTE 1—This figure shows the mapping from the XGMII to a 64B/65B block for a block containing eight data characters.
NOTE 2—Figure shown for L = 1.
NOTE 3—For PAM2 path, V=2 and u=2.

Figure 201–11—PCS 5G and 2.5G Transmit bit ordering for data mode and training mode, HS_TX

201.3.2.2.9 Terminate (/T/)

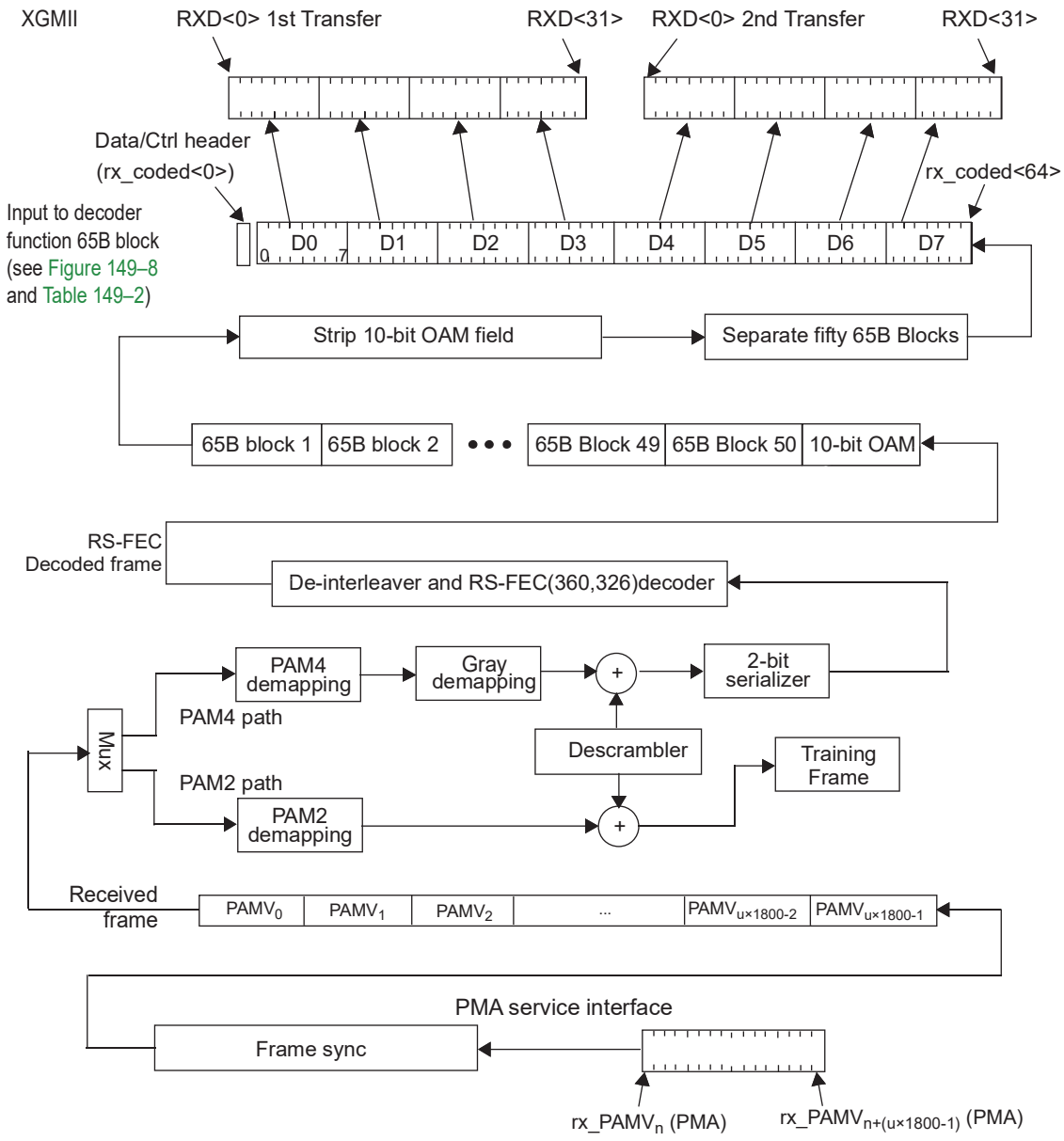
See 149.3.2.2.10.

201.3.2.2.10 Ordered set (/O/)

See 149.3.2.2.11.

201.3.2.2.11 Error (/E/)

See 149.3.2.2.12.



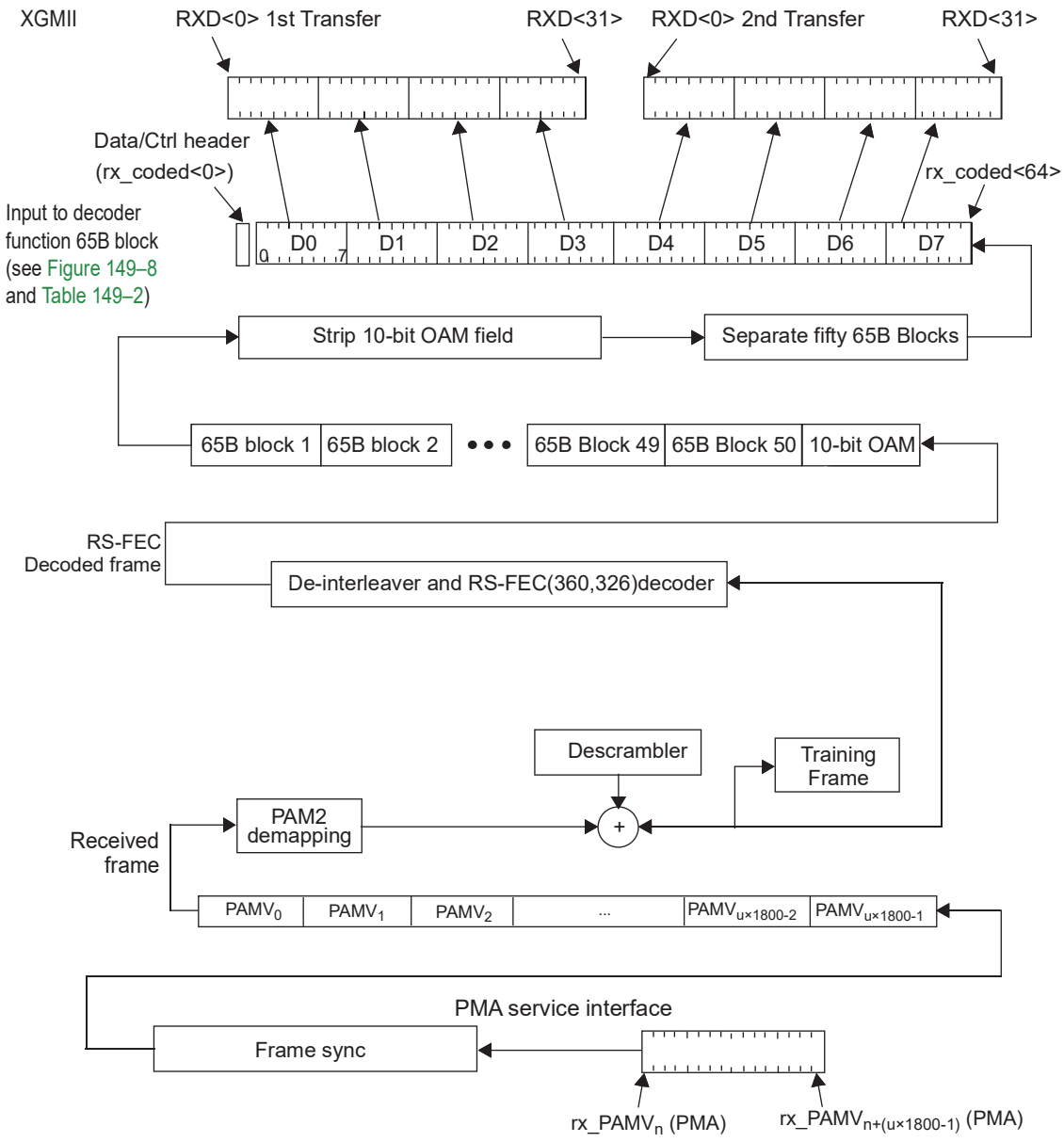
NOTE 1—This figure shows the mapping from a 64B/65B block a block containing eight data characters to the XGMII.
NOTE 2—Figure shown for L = 1.
NOTE 3—For PAM2 path, V=2 and u=2. For PAM4 path, V=4 and u=1.

Figure 201–12—PCS 10 G Receive bit ordering for data mode and training mode, HS_RX

201.3.2.2.12 Transmit process

The transmit process generates blocks based upon the TXD and TXC signals received from the XGMII. 100 XGMII data transfers are encoded into an RS-FEC frame.

For PAM4 path HS_TX (10 Gb/s), it takes 1800 PMA_UNITDATA transfers to send an RS-FEC frame of data, and a XGMII to PMA transfer rate of exactly 1:18.



NOTE 1—This figure shows the mapping from a 64B/65B block a block containing eight data characters to the XGMII.
NOTE 2—Figure shown for L = 1.
NOTE 3—For PAM2 path, V=2 and u=2. For PAM4 path, V=4 and u=1.

Figure 201–13—PCS 5G and 2.5G Receive bit ordering for data mode and training mode, HS_RX

For PAM2 path HS_TX (5 Gb/s and 2.5 Gb/s), it takes 3600 PMA_UNITDATA transfers to send an RS-FEC frame of data, and a XGMII to PMA transfer rate of exactly 1:36.

Therefore, for MultiG+100MBASE-T1/V1, if the PCS is connected to an XGMII and PMA sublayer where the ratio of their transfer rates is exactly 1:18, then the transmit process does not need to perform rate adaptation. Where the XGMII and PMA sublayer data rates are not synchronized to that ratio, the transmit process needs to insert idles, delete idles, or delete sequence ordered sets to adapt between the rates.

The transmit process generates blocks as specified in the PCS 64B/65B Transmit state diagram (see Figure 201–17). The contents of each block are contained in a vector tx_coded<64:0>, which is passed to the transcoder and scrambler. tx_coded<0> contains the data/ctrl header and the remainder of the bits contain the block payload.

201.3.2.2.13 RS-FEC framing and RS-FEC encoder

The resulting RS-FEC frame of 50 65B blocks, followed by the 10-bit OAM field and 340 parity bits is 3600 bits. See Figure 201–10 and Figure 201–11 for details on PCS bit ordering. See 201.3.2.2.16 for details on RS-FEC encoding.

The RS-FEC encoding takes the 3260-bit vector, consisting of tx_group50x65B, and the 10-bit OAM_field, and shall generate the 34 10-bit parity symbols (340 bits total).

201.3.2.2.14 RS-FEC superframe and round-robin interleaving

As specified in 149.3.2.2.15.

201.3.2.2.15 RS-FEC recombine

As specified in 149.3.2.2.16.

201.3.2.2.16 Reed-Solomon encoder

As specified in 149.3.2.2.17.

201.3.2.2.17 PCS scrambler PAM4

As specified in 149.3.2.2.18.

201.3.2.2.18 PCS scrambler PAM2

The bits of the interleaved RS-FEC superframe are presented as D_n, where n is an index indicating the symbol number, and are scrambled using an additive scrambler. The scrambling sequence DS_n is equal to $Scr_n[0]$ defined in 201.3.4.

All incoming PAM2 path HS_RX (5 Gb/s and 2.5 Gb/s) data bits are D_n, which are represented in Figure 201–7 as $D_n[0]$. The DS_n are applied as an additive scrambler sequence to each incoming data bit, D_n, to generate a single scrambled data bit, A_n, as shown in Equation (201–1).

$$A_n = DS_n \oplus D_n \quad (201–1)$$

201.3.2.2.19 Gray mapping for PAM4 encoding

As specified in 149.3.2.2.19.

201.3.2.2.20 Selectable precoder

As specified in 149.3.2.2.20 for PAM4 only.

201.3.2.2.21 PAM4 encoding

As specified in 149.3.2.2.21.

201.3.2.2.22 PAM2 encoding

The PCS transmit process shall encode each output symbol to one of two PAM2 levels as specified in this subclause.

The PAM2 encoded symbols are denoted M(*n*), where:

n is an index indicating the symbol number.

Each consecutive output symbol, A_n, is mapped to one of two PAM2 levels and assigned to the PAM2 encoder output M(*n*).

Mapping from the output symbol A_n to a PAM2 encoded symbol M(*n*) is as follows:
0 maps to +1, and
1 maps to –1.

201.3.2.3 PCS Receive function

The PCS Receive function shall conform to the PCS 64B/65B Receive state diagram in Figure 201–18 and the PCS Receive bit ordering in Figure 201–12 including compliance with the associated state variables as specified in 201.3.6.2.2.

The PCS Receive function accepts received code-groups provided by the PMA Receive function via the parameter rx_symb. The PCS receiver uses knowledge of the encoding rules and PMA training alignment to correctly align the 65B RS-FEC frames. The received symbols are demapped and descrambling is performed.

Following descrambling, the L-interleaved RS-FEC superframe is de-interleaved and the Reed-Solomon frames are decoded with Reed-Solomon error correction. Frames that cannot be corrected are marked with error symbols by the decoder. The RS-FEC decoded frame is then separated into a 10-bit OAM field and 50 64B/65B blocks. This process generates the 64B/65B block vector rx_coded<64:0>, which is then decoded to form the XGMII signals RXD<31:0> and RXC<3:0> as specified in the PCS 64B/65B Receive state diagram (see Figure 201–18). Two XGMII data transfers are decoded from each block. Where the XGMII and PMA sublayer data rates are not synchronized, the receive process inserts idles, deletes idles, or deletes sequence ordered sets to adapt between rates.

During PMA training mode, PCS Receive checks the received PAM2 framing and signals the reliable acquisition of the descrambler state by setting the scr_status parameter of the PMA_SCRSTATUS.request primitive to OK.

When the PCS Synchronization process has obtained synchronization, the RS-FEC frame error ratio (RFER) monitor state diagram shown in Figure 201–16 monitors the received signal for high RS-FEC frame error ratio and asserts hi_rfer to indicate excessive RS-FEC frame errors. If 40 consecutive RS-FEC frame errors are detected, the block_lock flag is de-asserted. The block_lock flag is re-asserted upon detection of a valid RS-FEC frame. When block_lock is asserted and hi_rfer is de-asserted, the PCS Receive process continuously accepts blocks. The PCS Receive process monitors these blocks and generates RXD <31:0> and RXC <3:0> on the XGMII.

When the receive channel is in training mode, the PCS Synchronization process continuously monitors PMA_RXSTATUS.indication(loc_rcvr_status). When loc_rcvr_status indicates OK, then the PCS Synchronization process accepts data-units via the PMA_UNITDATA.indication primitive. It attains frame and block synchronization based on the PMA training frames and conveys received blocks to the PCS Receive process. The PCS Synchronization process sets the block_lock flag to indicate whether the PCS has obtained synchronization. The PMA training frame includes an alignment bit every 450 PAM2 symbols,

which is aligned with the PCS partial PHY frame boundary, as well as an Infofield, which is inserted in the 16th PCS partial PHY frame. When the PCS Synchronization process is synchronized to this pattern, block_lock is asserted.

201.3.2.3.1 Frame and block synchronization

When operating in the data mode, a 100M+10GBASE-T1/V1 PHY's HS_RX PCS shall form a PAM4 stream from the PMA_UNITDATA.indication primitive by concatenating requests in order from rx_PAM4_0 to rx_PAM4_1799 (see Figure 201–12). It obtains block lock to the PHY frames during PAM2 training using synchronization bits provided in the training frames.

When operating in the data mode, a 100M+2.5GBASE-T1/V1 PHY and a 100M+5GBASE-T1/V1 PHY's HS_RX PCS shall form a PAM2 stream from the PMA_UNITDATA.indication primitive by concatenating requests in order from rx_PAM2_0 to rx_PAM2_3599 (see Figure 201–12). It obtains block_lock to the PHY frames during PAM2 training using synchronization bits provided in the training frames.

201.3.2.3.2 PCS descrambler

The descrambling process is as specified in 149.3.2.3.2, except Equation (149-6) shall be applied regardless of whether PHY_D is LEADER or FOLLOWER.

201.3.2.3.3 Invalid blocks

As specified in 149.3.2.3.3.

201.3.3 Test-pattern generators

As specified in 149.3.3.

201.3.4 Side-stream scrambler polynomials

The side stream scrambler shall employ Equation (149-6) as specified for $g_S(x)$ in 149.3.4. Equation (149-6) is applied regardless of whether PHY_S is LEADER or FOLLOWER.

201.3.5 HS_PATH PMA training frame

During PMA training, the training frames are embedded with indicators to establish alignment to the RS-FEC superframe composed of 16 partial PHY frames that comprise the block. The last partial PHY frame is embedded with an information field used to exchange messages between link partners.

For 10Gb/s, the timing relationship among training frame, partial frame, RS-FEC frame, superframe, and partial PHY frame count (PFC24) are shown in Figure 201–15.

For 2.5Gb/s and 5Gb/s, the timing relationship among training frame, partial frame, RS-FEC frame, superframe, and partial PHY frame count (PFC24) are shown in Figure 201–14.

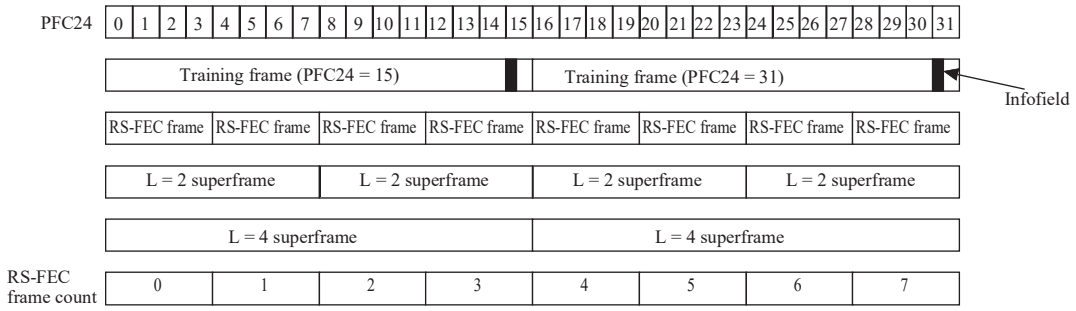


Figure 201–14—Timing relationship to PFC24 for 10G

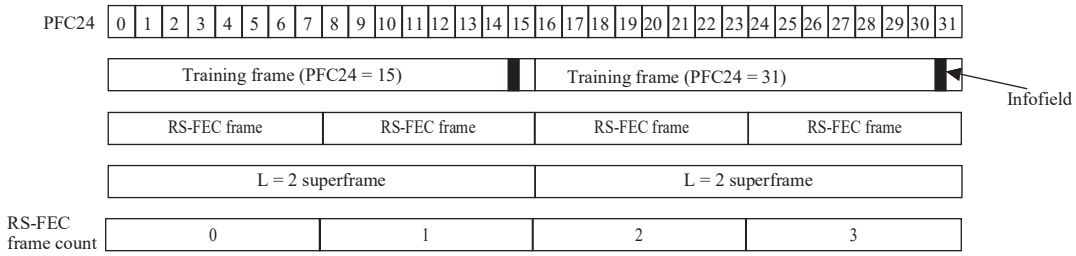


Figure 201–15—Timing relationship to PFC24 for 2.5G and 5G

PMA training frame encoding is based on the generation, at time n, of the bit S_n. The first bit is inverted in the first 15 partial PHY frames of each RS-FEC block. The first 96 bits of the 16th partial PHY frame are XORed with the contents of the Infofield. Each partial PHY frame is 450 bits long, beginning at S_n where (n mod 450) = 0. See Equation (201–2).

$$S_n = \begin{cases} Scr_n[0] \oplus InfoField_{(n \bmod 450)} & 6750 \le (n \bmod 7200) \le 6845 \\ Scr_n[0] \oplus 1 & \text{else if } (n \bmod 450) = 0 \\ Scr_n[0] & \text{otherwise} \end{cases} \quad (201–2)$$

201.3.5.1 Generation of symbol Tn

The bit S_n is mapped to the transmit symbol T_n as follows: if $S_n = 0$ then $T_n = +1$, if $S_n = 1$ then $T_n = –1$.

201.3.5.2 PMA training mode descrambler polynomials

The PHY shall acquire descrambler state synchronization to the PAM2 training sequence and report success through scr_status. For side-stream descrambling, the high speed receiver employs the receiver descrambler generator polynomial per 201.3.4.

201.3.6 Detailed functions and state diagrams

201.3.6.1 State diagram conventions

The body of this subclause is composed of state diagrams, including the associated definitions of variables, constants, and functions. Should there be a discrepancy between a state diagram and descriptive text, the state diagram prevails.

201.3.6.2 State diagram parameters

201.3.6.2.1 Constants

EBLOCK_R<71:0>
: 72-bit vector to be sent to the XGMII interface containing /E/ in all the eight character locations.

EBLOCK_T<64:0>
: 65-bit vector to be sent to the RS-FEC encoder containing /E/ in all the eight character locations.

LBLOCK_R<71:0>
: 72-bit vector to be sent to the XGMII interface containing two Local Fault ordered sets. The Local Fault ordered set is defined in 46.3.4.

LBLOCK_T<64:0>
: 65-bit vector to be sent to the RS-FEC encoder containing two Local Fault ordered sets.

RFER_CNT_LIMIT
: TYPE: Integer
VALUE: 16
Number of Reed-Solomon frames with uncorrectable errors.

RFRX_CNT_LIMIT
: TYPE: Integer
VALUE: 88
Number of Reed-Solomon frames received over bit error ratio interval.

UBLOCK_R<71:0>
: 72-bit vector to be sent to the XGMII containing two Link Interruption ordered sets.
The Link Interruption ordered set is defined in 46.3.4.

201.3.6.2.2 Variables

block_lock
: Boolean variable that is set TRUE when receiver acquires block delineation.

hi_rfer
: Boolean variable that is asserted TRUE when the rfer_cnt reaches 16 errors in one RFRX_CNT_LIMIT interval.

pcs_data_mode
: Variable set by the PMA PHY Control function. See 201.5.2.8.2.

pcs_reset
: Boolean variable that controls the resetting of the PCS. It is TRUE whenever a reset is necessary

including when reset is initiated from the MDIO, during power on, and when the MDIO has put the PCS into low-power mode.

rx_coded<64:0>
Vector containing the input to the 64B/65B decoder. The format for this vector is shown in Figure 149–8. The leftmost bit in the figure is rx_coded<0> and the rightmost bit is rx_coded<64>.

rx_raw<71:0>
Vector containing two successive XGMII output transfers. RXC<3:0> for the first transfer are taken from rx_raw<3:0>. RXC<3:0> for the second transfer are taken from rx_raw<7:4>. RXD<31:0> for the first transfer are taken from rx_raw<39:8>. RXD<31:0> for the second transfer are taken from rx_raw<71:40>.

rf_valid
Boolean indication that is set TRUE if received Reed-Solomon frame is valid. Reed-Solomon frame is valid if and only if all parity checks of the Reed-Solomon code are satisfied.

tx_coded<64:0>
Vector containing the output from the 64B/65B encoder. The format for this vector is shown in Figure 149–8. The leftmost bit in the figure is tx_coded<0> and the rightmost bit is tx_coded<64>.

tx_raw<71:0>
Vector containing two successive XGMII transfers. TXC<3:0> for the first transfer are placed in tx_raw<3:0>. TXC<3:0> for the second transfer are placed in tx_raw<7:4>. TXD<31:0> for the first transfer are placed in tx_raw<39:8>. TXD<31:0> for the second transfer are placed in tx_raw<71:40>.

201.3.6.2.3 Timers

rfer_timer
Timer that is triggered every $125/(4 \times S)$ μs +1%, –25%. When the timer reaches its terminal count, rfer_timer_done = TRUE. See Table 201–1 for the definition of *S*.

201.3.6.2.4 Functions

DECODE(rx_coded<64:0>)
In the PCS Receive process, this function takes as its argument 65-bit rx_coded<64:0> from the RS-FEC decoder and decodes the 65B RS-FEC bit vector returning a vector rx_raw<71:0>, which is sent to the XGMII. The DECODE function shall decode the block based on code specified in 149.3.2.2.2.

ENCODE(tx_raw<71:0>)
Encodes the 72-bit vector received from the XGMII, returning 65-bit vector tx_coded. The ENCODE function shall encode the block as specified in 201.3.2.2.2.

R_BLOCK_TYPE = {C, S, T, D, E}
This function classifies each 65-bit rx_coded vector as belonging to one of the five types {C, S, T, D, E} depending on its contents.
A vector belongs to only one type.
Values: C; The vector contains a data/ctrl header of 1 and one of the following:
a) A block type field of 0x1E and eight valid control characters other than /E/;

b) A block type field 0x2D or 0x4B, a valid O code, and four valid control characters;
c) A block type field of 0x55 and two valid O codes.

S; The vector contains a data/ctrl header of 1 and one of the following:
a) A block type field of 0x33 and four valid control characters;
b) A block type field of 0x66 and a valid O code;
c) A block type field of 0x78.

T; The vector contains a data/ctrl header of 1, a block type field of 0x87, 0x99, 0xAA, 0xB4, 0xCC, 0xD2, 0xE1 or 0xFF and all control characters are valid.

D; The vector contains a data/ctrl header of 0.

E; The vector does not meet the criteria for any other value.

A valid control character is one containing a MultiGBASE-T1 control code specified in Table 149–2. A valid O code is one containing an O code specified in Table 149–2.

R_TYPE(rx_coded<64:0>)

Returns the R_BLOCK_TYPE of the rx_coded<64:0> bit vector.

R_TYPE_NEXT

Prescient end of packet check function. It returns the R_BLOCK_TYPE of the rx_coded vector immediately following the current rx_coded vector.

T_BLOCK_TYPE = {C, S, T, D, E}

This function classifies each 72-bit tx_raw vector as belonging to one of the five types {C, S, T, D, E} depending on its contents. A vector belongs to only one type.

Values: C; The vector contains one of the following:
a) Eight valid control characters other than /O/, /S/, /T/, and /E/;
b) One valid ordered set and four valid control characters other than /O/, /S/, and /T/;
c) Two valid ordered sets.

S; The vector contains an /S/ in its first or fifth character. Any characters before the S character are valid control characters other than /O/, /S/ and /T/ or form a valid ordered set, and all characters following the /S/ are data characters.

T; The vector contains a /T/ in one of its characters, all characters before the /T/ are data characters, and all characters following the /T/ are valid control characters other than /O/, /S/ and /T/.

D; The vector contains eight data characters.

E; The vector does not meet the criteria for any other value.

A tx_raw character is a control character if its associated TXC bit is asserted. A valid control character is one containing an XGMII control code specified in Table 149–2. A valid ordered set consists of a valid /O/ character in the first or fifth characters and data characters in the three characters following the /O/. A valid /O/ is any character with a value for O code in Table 149–2.

T_TYPE(tx_raw<71:0>)

Returns the T_BLOCK_TYPE of the tx_raw<71:0> bit vector.

T_TYPE_NEXT

Prescient end of packet check function. It returns the FRAME_TYPE of the tx_raw vector immediately following the current tx_raw vector.

201.3.6.2.5 Counters

rfer_cnt

Count up to a maximum of RFER_CNT_LIMIT of the number of invalid Reed-Solomon frames within the current RFRX_CNT_LIMIT Reed-Solomon frame period.

rfrx_cnt
: Count number Reed-Solomon frames received during current period.

201.3.6.2.6 Messages

RX_FRAME
: A signal sent to PCS Receive indicating that a full Reed-Solomon frame has been decoded and the variable rf_valid is updated.

201.3.6.3 State diagrams

The RFER monitor state diagram shown in Figure 201–16 monitors the received signal for high RS-FEC frame error ratio.

The PCS 64B/65B Transmit state diagram shown in Figure 201–17 controls the encoding of 65B transmitted blocks. It makes exactly one transition for each 65B transmit block processed. Though the state diagram sends Local Fault ordered sets when reset is asserted, the scrambler and 65B RS-FEC are not guaranteed to be operational during reset. Thus, the Local Fault ordered sets are not guaranteed to appear on the PMA service interface.

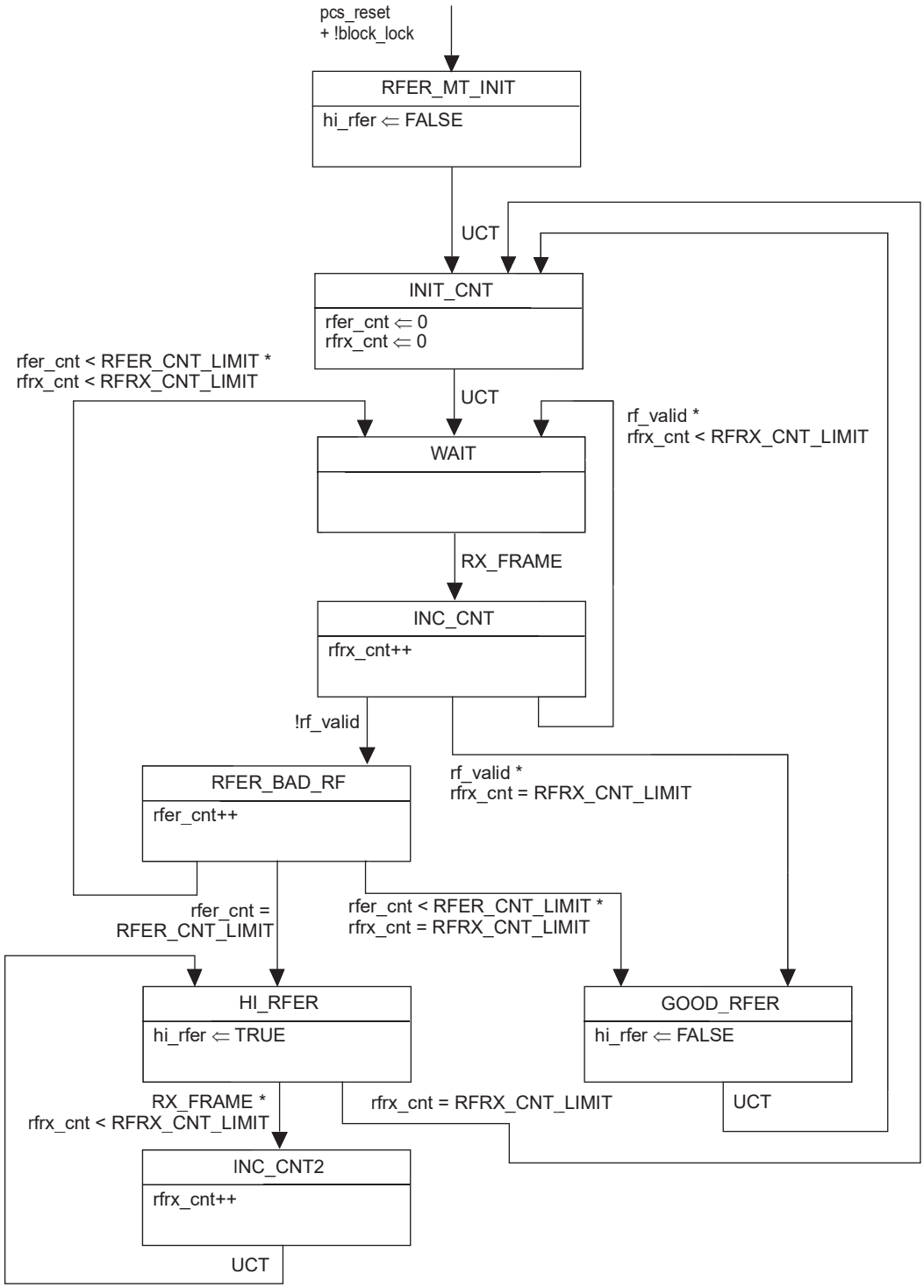


Figure 201–16—RFER monitor state diagram

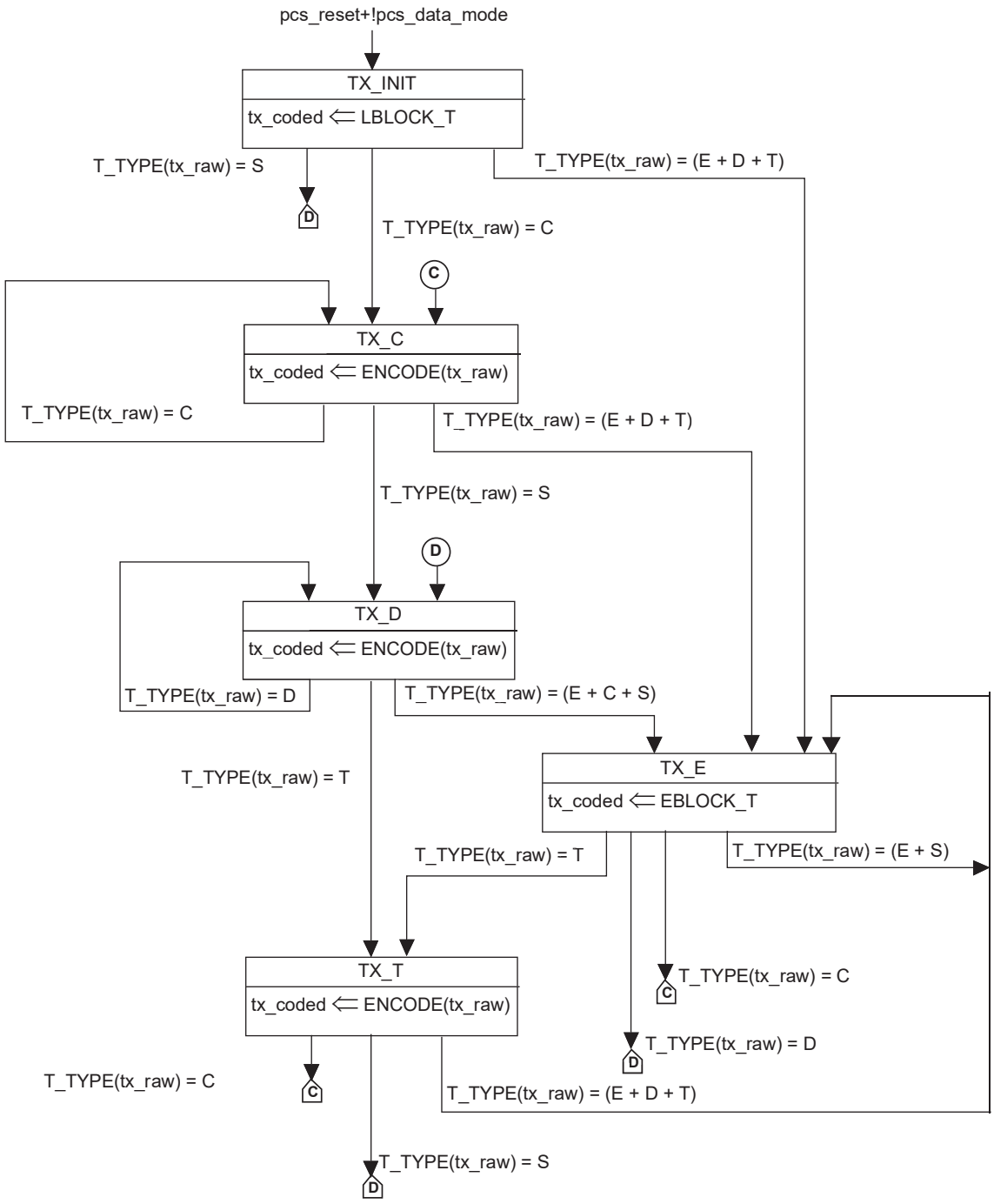


Figure 201–17—PCS 64B/65B Transmit state diagram

The PCS 64B/65B Receive state diagram is shown in Figure 201–18 and controls the decoding of 65B received blocks. It makes exactly one transition for each receive block processed except for the transition from RX_WE to RX_E, which occurs immediately after the RX_WE processes are complete.

The PCS shall perform the functions of RFER monitor, Transmit, and Receive as specified in these state diagrams.

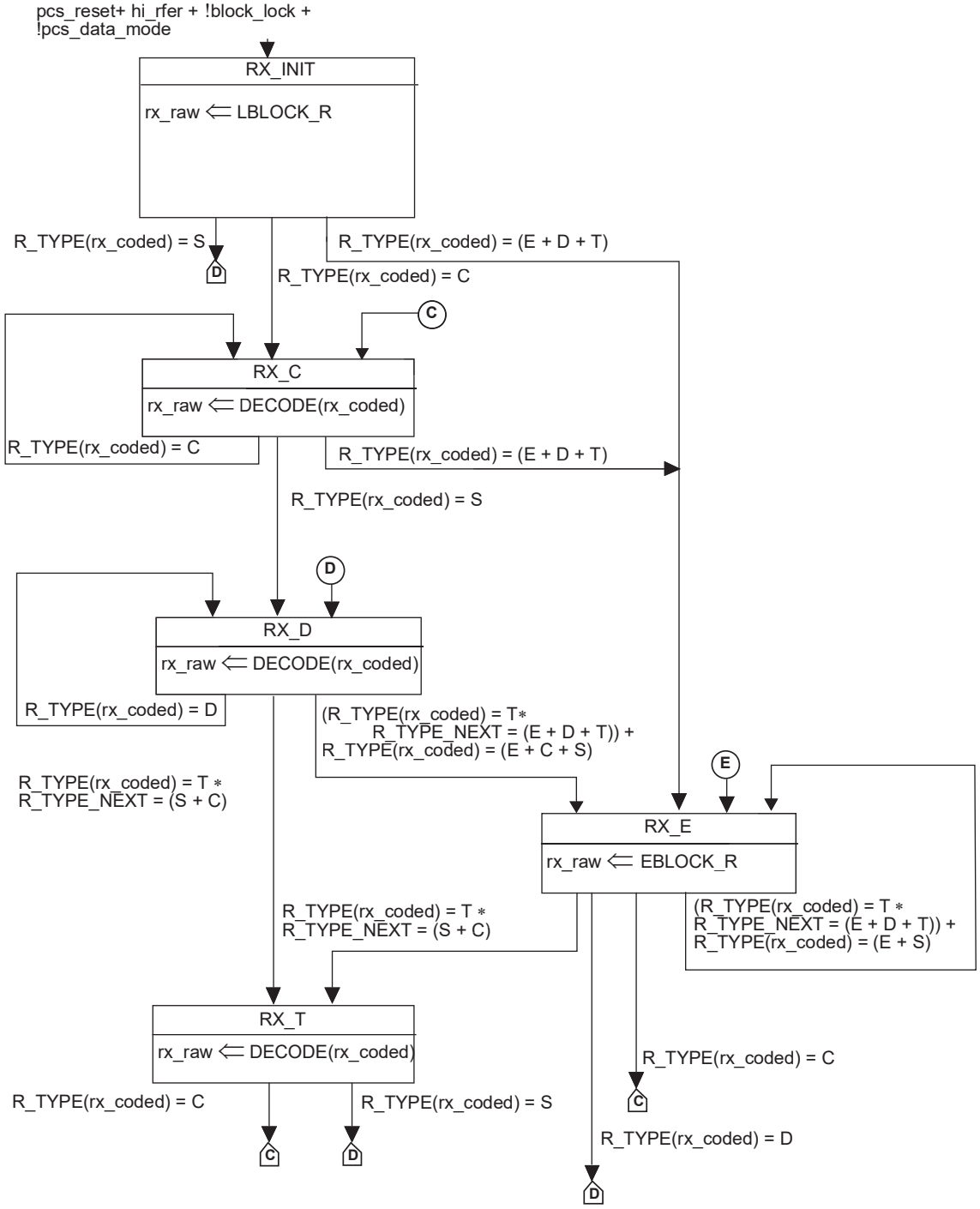


Figure 201–18—PCS 64B/65B Receive state diagram

201.3.7 PCS management

The following objects apply to PCS management. If an MDIO Interface is provided (see Clause 45), they are accessed via that interface. If not, it is recommended that an equivalent access be provided.

201.3.7.1 Status

pcs_status:
: Indicates whether the PCS is in a fully operational state. It is only TRUE if pcs_data_mode is TRUE, block_lock is TRUE, and hi_rfer is FALSE. This status is reflected in MDIO bit 3.2324.10. A latch low view of this status is reflected in MDIO bit 3.2323.2 and the inverse of this status is reflected in MDIO bit 3.2323.7.

block_lock:
: Indicates the state of the block_lock variable. This status is reflected in MDIO bit 3.2324.8. A latching low version of this status is reflected in MDIO bit 3.2324.6.

hi_rfer:
: Indicates the state of the hi_rfer variable. This status is reflected in MDIO bit 3.2324.9. A latching high version of this status is reflected in MDIO bit 3.2324.7.

201.3.7.2 Counter

The following counter is reset to zero upon read and upon reset of the PCS. When it reaches all ones, it stops counting. Its purpose is to help monitor the quality of the link.

RFER_count:
: 6-bit counter that counts each time the RFER_BAD_RF of the RFER monitor state diagram (see Figure 201–16) is entered. This counter is reflected in MDIO register bits 3.2324.5:0. The counter is reset when register 3.2324 is read by management. Note that this counter counts a maximum of RFER_CNT_LIMIT counts per RFRX_CNT_LIMIT period since the RFER_BAD_RF state can be entered a maximum of RFER_CNT_LIMIT times per RFRX_CNT_LIMIT window.

201.3.8 MultiG+100MBASE-T1/V1 operations, administration, and maintenance (OAM)

The MultiG+100MBASE-T1/V1 PCS level operations, administration, and maintenance (OAM) provides an optional mechanism useful for monitoring link operation such as exchanging PHY link health status and message exchange. When OAM is implemented, behavior is defined in 149.3.9, including the state diagrams in Figure 149–24 and Figure 149–25.

The OAM frame data is carried in the OAM 10-bit field described in 201.3.2.2.13 for HS_PATH.

OAM involves both HS_PATH and LS_PATH. The 10-bit symbols are inserted one at a time into the OAM field in each Reed Solomon frame in the HS_PATH and LS_PATH.

201.3.8.1 Definitions

The definitions for OAM are as defined in 149.3.9.1 for OAM frame, OAM symbol, OAM message, and OAM status.

OAM field: A 10-bit field in each RS-FEC frame reserved for the OAM symbol as described in 201.3.2.2.14.