

PCS Receive and Transmit State Machines

Saied Benyamin, Aquantia

January 14-16, 2019

IEEE 802.3ch Interim

149.3.2.2 PCS Transmit function

The PCS Transmit function shall conform to the PCS 64B/65B Transmit state diagram in Figure ~~136-15149-1000~~ and Figure ~~136-16149-1001~~, and to the PCS Transmit bit ordering in Figure 149-4 and Figure ~~136-8149-5~~.

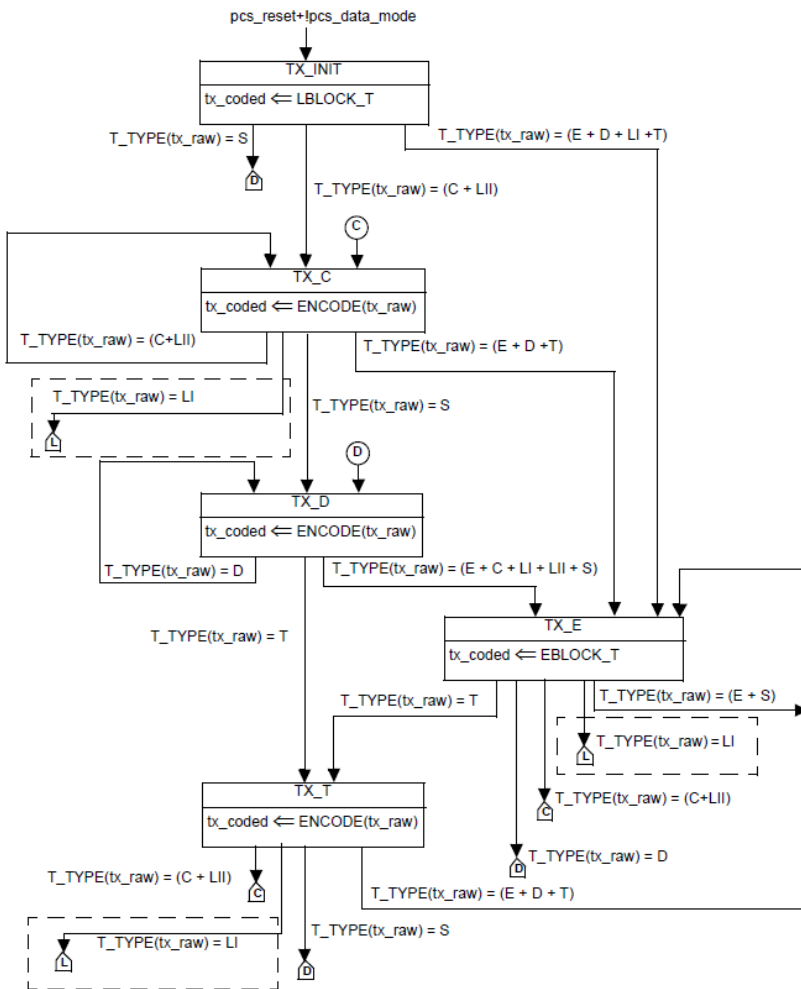
Dashed rectangles in Figure ~~136-15149-1000~~ and Figure ~~136-16149-1001~~ are used to indicate states and state transitions in the transmit process state diagram that shall be supported by PHYs with the EEE capability. PHYs without the EEE capability do not support these transitions.

149.3.2.2.13 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. It takes 1800 PMA_UNITDATA transfers to send an RS-FEC frame of data. Therefore, for 2.5G/5G/10GBASE-T1, 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.

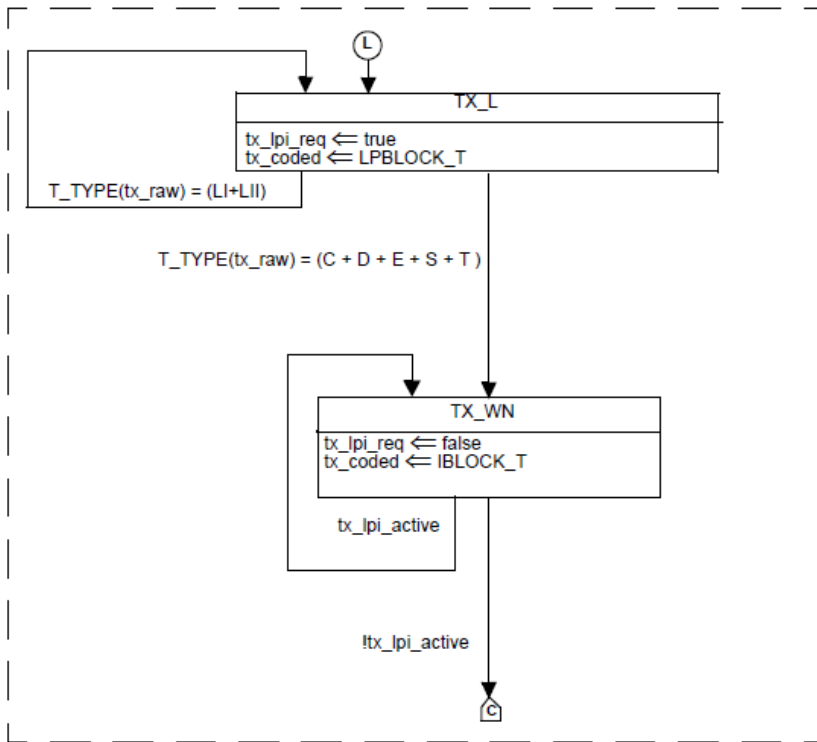
The transmit process generates blocks as specified in the PCS 64B/65B Transmit state diagram (see Figure ~~136-15149-1000~~ and Figure ~~136-16149-1001~~). The contents of each block are contained in a vector `tx_coded<64:0>`, which is passed to the transcoder/ scrambler. `tx_coded<0>` contains the data/ctrl header and the remainder of the bits contain the block payload.

The following images come after section 149.3.6.2.5 Counters



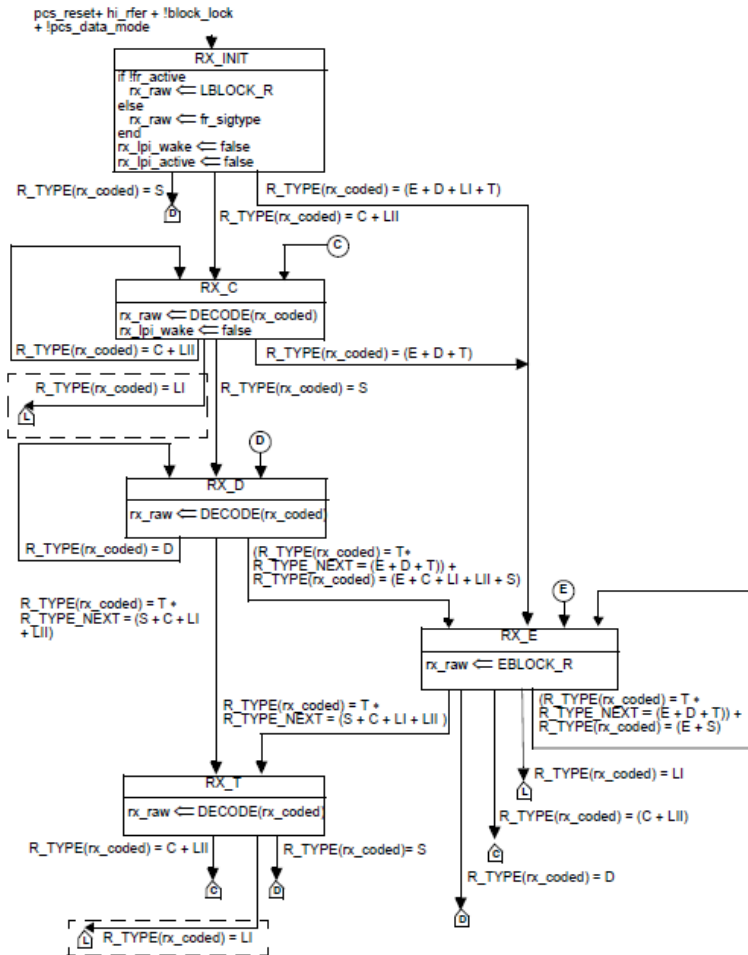
NOTE—Transitions inside dashed boxes are only required for the EEE capability.

Figure 149-1000—PCS 64B/65B Transmit state diagram, part a



NOTE—This figure is mandatory for PHYs with the EEE capability.

Figure 149-1000—PCS 64B/65B Transmit state diagram, part b



NOTE—Signals and functions shown with dashed lines are only required for the EEE capability.

Figure 149-1001—PCS 64B/65B Receive state diagram, part a

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.

Formatted: Font: 9 pt

Formatted: SC.13.270344, Font: (Default) +Body (Calibri), 11 pt

Formatted: Indent: Left: 0.5", Space After: 8 pt, Line spacing: Multiple 1.08 li, Adjust space between Latin and Asian text, Adjust space between Asian text and numbers

Formatted: Font: 9 pt

Formatted: Indent: First line: 0.5"

LBLOCK_T<64:0>

65 bit vector to be sent to the LDPC encoder containing two Local Fault ordered sets.

Formatted: Font: 9 pt

Formatted: Indent: First line: 0.5"

LPBLOCK_R<71:0>

72 bit vector to be sent to the XGMII containing /L/ in all the eight character locations.

Formatted: Font: 9 pt

Formatted: Indent: First line: 0.5"

LPBLOCK_T<64:0>

65 bit vector to be sent to the LDPC encoder containing /L/ in all the eight character locations.

Formatted: Font: 9 pt

Formatted: Indent: First line: 0.5"

IBLOCK_R<71:0>

72 bit vector to be sent to the XGMII containing /I/ in all the eight character locations.

Formatted: Font: 9 pt

Formatted: Indent: First line: 0.5"

IBLOCK_T<64:0>

65 bit vector to be sent to the LDPC encoder containing /I/ in all the eight character locations.

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.

Formatted: Font: 9 pt

Formatted: Indent: First line: 0.5", Space After: 0 pt, Line spacing: single, Don't adjust space between Latin and Asian text, Don't adjust space between Asian text and numbers

Formatted: Font: 9 pt

149.3.6.2.2 Variables

rfer_test_lf

Boolean variable that is set true when a new Reed Solomon frame is available for testing and false when RFER_TEST_LF state is entered. A new Reed Solomon frame is available for testing when the Block Sync process has accumulated enough symbols from the PMA to evaluate the next Reed Solomon frame.

Formatted: Indent: First line: 0.5"

block_lock

Boolean variable that is set true when receiver acquires block delineation.

Formatted: Indent: First line: 0.5"

hi_rfer

Boolean variable that is asserted true when the rfer_cnt reaches 16 errors in one rfer_timer interval.

Formatted: Indent: First line: 0.5"

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.

Formatted: Indent: First line: 0.5"

rx_coded<64:0>

Vector containing the input to the 64B/65B decoder. The format for this vector is shown in Figure 149-6. The leftmost bit in the figure is rx_coded<0> and the rightmost bit is rx_coded<64>.

Formatted: Indent: First line: 0.5"

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>.

Formatted: Indent: First line: 0.5"

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.

Formatted: Normal, Space Before: 0 pt, After: 0 pt

Formatted: Default Paragraph Font

Formatted: Indent: First line: 0.5"

tx_coded<64:0>

Vector containing the output from the 64B/65B encoder. The format for this vector is shown in Figure 149-6. The leftmost bit in the figure is tx_coded<0> and the rightmost bit is tx_coded<64>.

Formatted: Indent: First line: 0.5"

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>.

The following variables are required for PHYs that support the EEE capability:

To be added when we have consensus on full EEE

The following variable is only required for PHYs that support the fast retrain capability:

To be added when we agree on fast retrain

Formatted: Normal, Space Before: 0 pt, After: 0 pt, Don't adjust space between Latin and Asian text, Don't adjust space between Asian text and numbers

149.3.6.2.4 Functions

DECODE(rx_symb_vector<64:0>)

In the PCS Receive process, this function takes as its argument 65-bit rx_coded<64:0> from the LDPC decoder and decodes the 65B-LDPC bit vector returning a vector rx_raw<71:0>, which is

Formatted: Font: 11 pt, Font color: Auto

Formatted: Indent: First line: 0.5"

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 [149.3.2.2.2](#).

Formatted: Indent: First line: 0.5"

R_BLOCK_TYPE = {C, S, T, D, E, I, LI, LII}

When the EEE capability is not supported, 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.

Formatted: Indent: First line: 0.5"

When the EEE capability is supported, this function classifies each 65-bit rx_coded vector as belonging to the eight types depending on its contents. A vector may simultaneously belong to the C and I types when it contains eight valid control characters that are all /I/, but in every other case the 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/ and /LI/;

Formatted: Indent: Left: 0.5", First line: 0.5"

b) A block type field of 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:

Formatted: Indent: First line: 0.5"

a) A block type field of 0x33 and four valid control characters;

Formatted: Indent: Left: 0.5", First line: 0.5"

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.

Formatted: Indent: First line: 0.5"

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

I; If the optional EEE capability is supported, then the I type is a special case of the C type where the vector contains a data/ctrl header of 1, a block type field of 0x1e, and eight control characters of /I/.

LI: If the optional EEE capability is supported, then the LI type occurs when the vector contains a data/ctrl header of 1, a block type field of 0x1e, and eight control characters of /LI/.

LII: If the optional EEE capability is supported, then the LII type occurs when the vector contains a data/ctrl header of 1, a block type field of 0x1E, and one of the following:

a) Four control characters of /LI/ followed by four control characters of /I/;

Formatted: Indent: Left: 0.5", First line: 0.5"

b) Four control characters of /I/ followed by four control characters of /LI/.

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

Formatted: Indent: First line: 0.5"

A valid control character is one containing a 2.5G/5GBASE-T control code specified in Table 149-1. A valid O code is one containing an O code specified in Table 126-1.

R_TYPE(rx_coded<64:0>)

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

Formatted: Indent: First line: 0.5"

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.

Formatted: Indent: First line: 0.5", Space After: 0 pt, Line spacing: single, Don't adjust space between Latin and Asian text, Don't adjust space between Asian text and numbers

T_BLOCK_TYPE = {C, S, T, D, E, I, LI, LII}

When the EEE capability is not supported, 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.

Formatted: Indent: First line: 0.5"

When the EEE capability is supported, this function classifies each 72-bit tx_raw vector as belonging to the eight types depending on its contents. A vector may simultaneously belong to the C and I types when it contains eight valid control characters that are all /I/, but in every other case the vector belongs to only one type.

Values:

C; The vector contains one of the following:

Formatted: Indent: First line: 0.5"

- a) Eight valid control characters other than /O/, /S/, /T/, /E/, and /LI/;
- b) One valid ordered set and four valid control characters other than /O/, /S/, and /T/;
- c) Two valid ordered sets.

Formatted: Indent: Left: 0.5", First line: 0.5"

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.

Formatted: Indent: First line: 0.5"

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.

I; If the optional EEE capability is supported, then the I type is a special case of the C type where the vector contains eight control characters of /I/.

LI: If the optional EEE capability is supported, then the LI type occurs when the vector contains eight control characters of /LI/.

LII: If the optional EEE capability is supported, then the LII type occurs when the vector contains one of the following:

- a) Four control characters of /LI/ followed by four control characters of /I/;
- b) Four control characters of /I/ followed by four control characters of /LI/.

Formatted: Indent: Left: 0.5", First line: 0.5"

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

Formatted: Indent: First line: 0.5"

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 126–1. 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 149–1.

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.

← **Formatted:** Indent: First line: 0.5"

← **Formatted:** Indent: First line: 0.5"