

5 FC-1 transmission codes

5.1 Overview

Transmission codes are a function of the FC-1 level. Communication of words and Special Functions are FC-1 functions. Use of Special Functions is an FC-2P function.

Information to be transmitted over a fibre shall be presented to the FC-1 level as a stream of words and Special Functions. It shall be encoded using one of the transmission codes specified in this clause into a stream of Transmission Words that shall be sent across the link. Information shall be received over the link as a stream of Transmission Words. The stream of Transmission Words shall be decoded using one of the transmission codes specified in this clause into a stream of words and Special Functions that shall be delivered to the FC-2P sublevel.

This standard specifies two types of transmission codes:

- a) frame transfer transmission codes are specified to transfer Upper Level Protocol data; and
- b) other transmission codes (e.g., the Transmitter Training Signal, see 5.6) are specified for purposes other than transferring Fibre Channel frames.

Both types of transmission code provide these functions:

- a) maintaining Bit Synchronization and Transmission Word Synchronization;
- b) communicating link control information; and
- c) increasing the likelihood of detection of transmission errors.

Frame transfer transmission codes additionally provide these functions:

- a) communicating link state machine transitions;
- b) communicating other Special Functions;
- c) denoting frame boundaries; and
- d) communicating Upper Level Protocol data.

The encodings defined by the transmission code ensure that sufficient transitions are present in the serial bit stream to make clock recovery possible at the receiver. Such encoding also increases the likelihood of detecting any single or multiple bit errors that may occur during transmission and reception of information. In addition, the transmission code assures presence of a distinct and easily recognizable bit pattern that assists a receiver in achieving Transmission Word alignment on the incoming bit stream.

An FC-0 standard for a physical variant may specify a transmission code. If an FC-0 standard for a physical variant does not specify a transmission code, then the physical variant shall use the 8B/10B transmission code (see).

5.4 32GFC 256B/257B transmission code

5.4.1 Overview

An FC-0 standard (e.g., FC-PI-6) may specify the use of the 32GFC 256B/257B transmission code as its frame transfer transmission code. If the 32GFC 256B/257B transmission code is specified, then it shall be:

- generated as described in 5.4.2;
- encoded with Reed Solomon coding as described in 5.4.3;
- scrambled as described in 5.4.4;
- descrambled as described in 5.4.5;
- decode with the Reed Solomon decoder as described in 5.4.6; and
- decoded as described in 5.4.7.

5.4.2 64B/66B to 256B/257B Transcoding

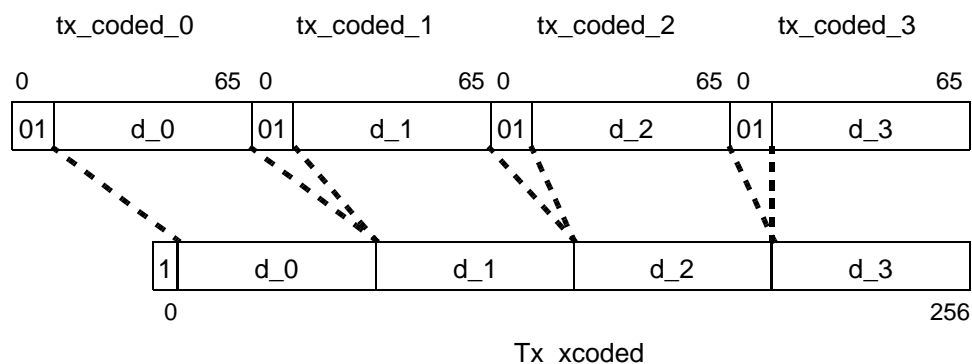
The 256B/257B transmission code specified by this standard operates on 4 consecutive 64B/66B Transmission Words (see 5.3), each group being encoded as a 257-bit Transmission Word.

NOTE 1 - The IEEE 802.3bj-2014 specification of 256B/257B references as “blocks” what this standard references as “Transmission Words”.

The transcoder constructs a 257-bit Transmission Word from a group of 4 x 66-bit Transmission Words to allocate bandwidth for the parity check symbols added by the Reed-Solomon encoder.

The 257-bit Transmission Word $tx_xcoded<256:0>$ shall be constructed as defined in IEEE 802.3bj-201X 91.5.2.5 given 4 x 66-bit Transmission Words denoted as $tx_coded_j<65:0>$ where $j=0$ to 3. The first 5 bits of $tx_xcoded<256:0>$ are not scrambled (i.e., the step that generates $tx_scrambled<256:0>$ is not performed).

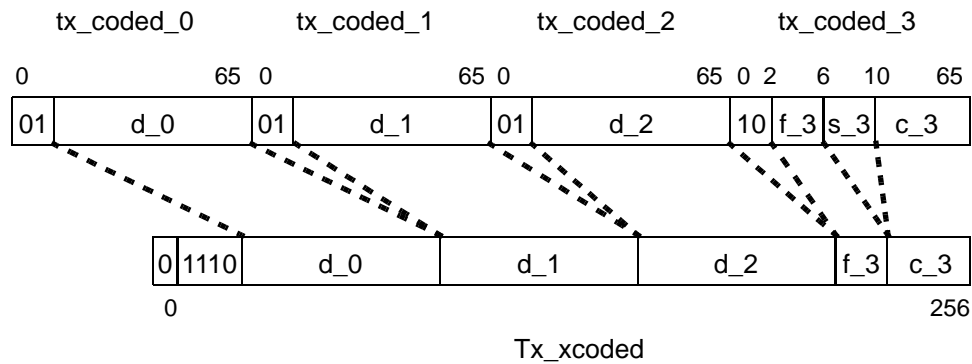
Figure 10 shows the 32GFC 256B/257B encoding of four data words.



Key:
 _x = data from the encoded 64/66b block

Figure 10 - 32GFC 256B/257B encoding of four data words

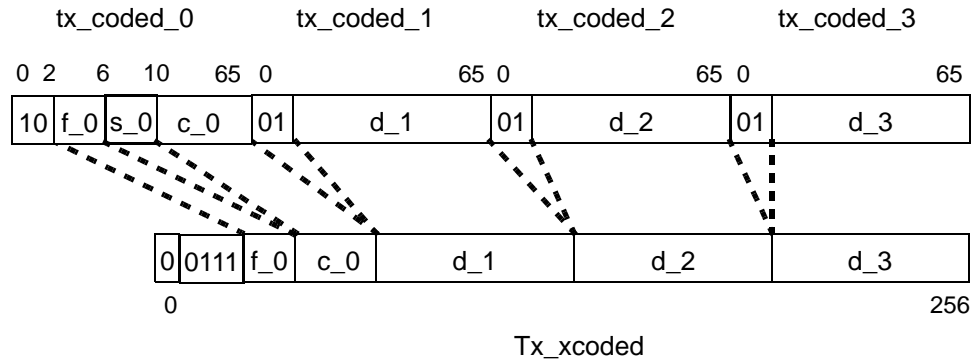
Figure 11 shows the 32GFC 256B/257B encoding of three data words followed by one control word.



Key:
 d_x = data from the encoded 64/66b block
 c_x = control codes from the encoded 64/66b block
 f_x = first 4 bits of the block type field in the encoded 64/66b block
 s_x = second 4 bits of the block type field in the encoded 64/66b block

Figure 11 - 32GFC 256B/257B encoding of three data words followed by one control word

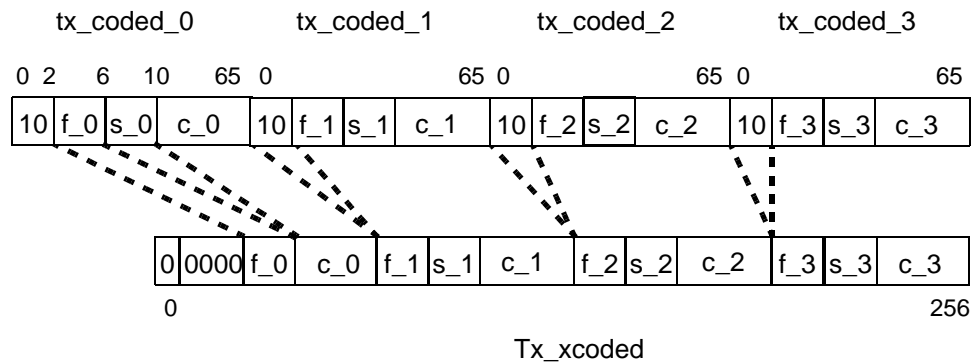
Figure 12 shows the 32GFC 256B/257B encoding of one control word followed by three data words.



Key:
 d_x = data from the encoded 64/66b block
 c_x = control codes from the encoded 64/66b block
 f_x = first 4 bits of the block type field in the encoded 64/66b block
 s_x = second 4 bits of the block type field in the encoded 64/66b block

Figure 12 - 32GFC 256B/257B encoding of one control word followed by three data words

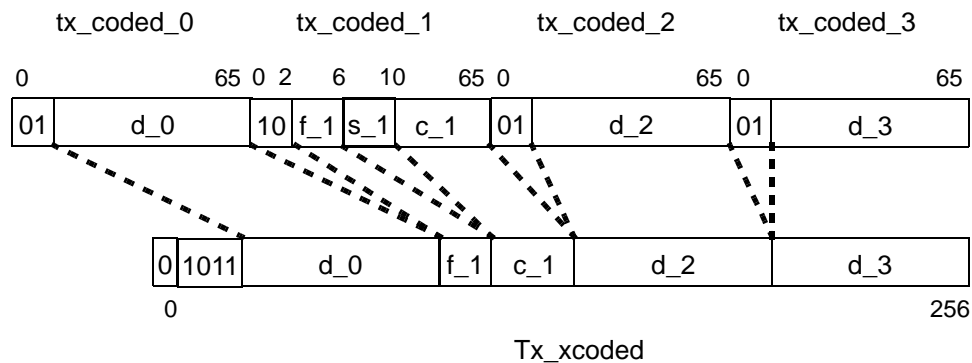
Figure 13 shows the 32GFC 256B/257B encoding of four control words.



Key:
 d_x = data from the encoded 64/66b block
 c_x = control codes from the encoded 64/66b block
 f_x = first 4 bits of the block type field in the encoded 64/66b block
 s_x = second 4 bits of the block type field in the encoded 64/66b block

Figure 13 - 32GFC 256B/257B encoding of four control words

Figure 14 shows the 32GFC 256B/257B encoding of one data word followed by one control word followed by two data words.



Key:
 d_x = data from the encoded 64/66b block
 c_x = control codes from the encoded 64/66b block
 f_x = first 4 bits of the block type field in the encoded 64/66b block
 s_x = second 4 bits of the block type field in the encoded 64/66b block

Figure 14 - 32GFC 256B/257B encoding of one data word, followed by one control word, followed by two data words

A stream of 32GFC 256B/257B Transmission Words on a link shall be further encoded to provide Forward Error Correction (i.e., FEC).

The streams of 32GFC 256B/257B Transmission Words in both directions on the link shall be encoded as specified in 5.4 and then further encoded as specified in subclause 91.5.2.7 of IEEE 802.3bj-2014.

5.4.3 Reed-Solomon encoder

The RS-FEC sublayer employs a Reed-Solomon code (see bibliography Annex M) operating over the Galois Field $GF(2^{10})$ (see bibliography Annex M) where the symbol size is 10 bits. The encoder processes k message symbols to generate $2t$ parity symbols which are then appended to the message to produce a code word of $n=k+2t$ symbols. For the purposes of this clause, a particular Reed-Solomon code is denoted $RS(n, k)$.

The RS-FEC sublayer shall implement $RS(528, 514)$. Each k -symbol message corresponds to twenty 257-bit Transmission Words produced by the transcoder. Each code is based on the generating polynomial given by Equation 91–1 of IEEE 802.3bj-2014.

5.4.4 Scrambler

Each RS-FEC code word is scrambled with a known sequence to randomize the 257-bit Transmission Word headers and to enable robust code word synchronization at the receiver (i.e., ensure that any shifted input bit sequence is not equal to another RS-FEC code word). Scrambling is implemented as modulo 2 addition of the RS-FEC code word and a pseudo-noise sequence 5280 bits in length defined as PN-5280 (see figure 15).

PN-5280 is generated by the polynomial $r(x)$.

$$r(x) = x^{39} + x^{58} + 1$$

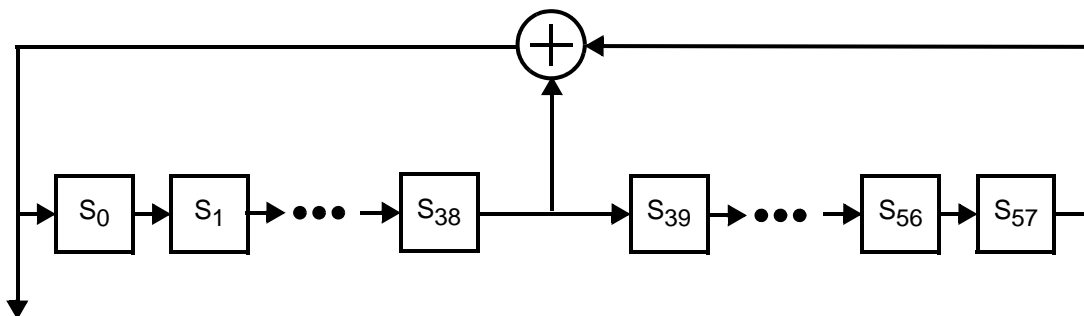


Figure 15 - PN-5280 as a linear feedback shift register

At the start of each RS-FEC code word, the initial state of the pseudo-noise generator is set to:

$$S_{57} = 1$$

$$S_{i-1} = S_i \text{ XOR } 1$$

(i.e., a binary sequence of alternating 1's and 0's).

5.4.5 Descrambler

Each code word shall be descrambled prior to decoding. Descrambling is implemented as the modulo 2 addition of RS-FEC code word and the same pseudo-noise sequence PN-5280 defined for the scrambler (see 5.4.4).

5.4.6 Reed-Solomon decoder

The Reed-Solomon decoder extracts the message symbols from the code word, correcting them as necessary, and discards the parity symbols. The message symbols correspond to 20 x 257-bit Transmission Words.

The Reed-Solomon decoder shall be capable of correcting any combination of up to $t=7$ symbol errors in a code word. It shall also be capable of indicating when a code word contains errors but was not corrected (e.g., it contains a number of errors in excess of the error correction capability).

5.4.7 32GFC 256B/257B to 64B/66B transcoder

The transcoder reconstructs a group of 4 x 66-bit Transmission Words from each received 257-bit Transmission Word.

The 4 x 66-bit Transmission Words, denoted as $rx_coded_j<65:0>$ where $j=0$ to 3, shall be derived from each 257-bit Transmission Word $rx_xcoded<256:0>$ as defined in IEEE 802.3bj-2014 91.5.3.5. As the first 5 bits of $rx_xcoded<256:0>$ are not scrambled, the step defined in 802.3bj that derives rx_xcoded from $rx_scrambled$ is not performed on those bits.

5.4.8 Transmit Bit Ordering

Transmit bit ordering for 32GFC 256B/257B is as shown in figure 16.

SH_n = Synchronization Header n according to figure 10

TWB_n = Scrambled Transmission Word Body n according to figure 10; n = 0 (i.e., earliest word) to n = 3 (i.e., latest word)

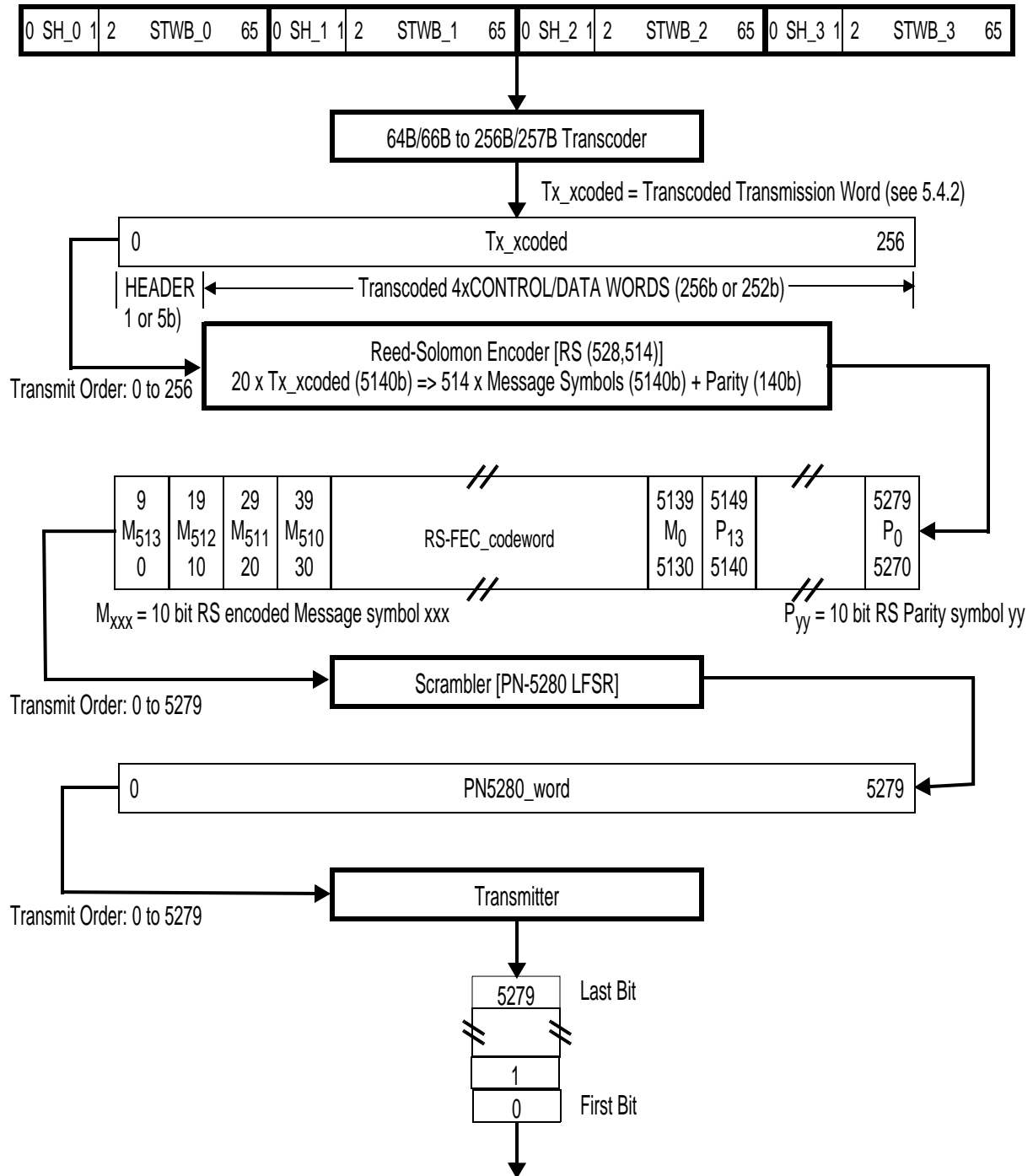


Figure 16 - 32GFC 256B/257B transmit bit ordering

5.4.9 Receive Bit Ordering

Receive bit ordering for [32GFC](#) 256B/257B is as shown in figure 17.

rx_coded_n = Received 66 bit Transmission word (see 5.4.7)

SH_n = Synchronization Header n according to figure 10

TWB_n = Scrambled Transmission Word Body n according to figure 10; n = 0 (i.e., earliest word) to n = 3 (i.e., latest word)

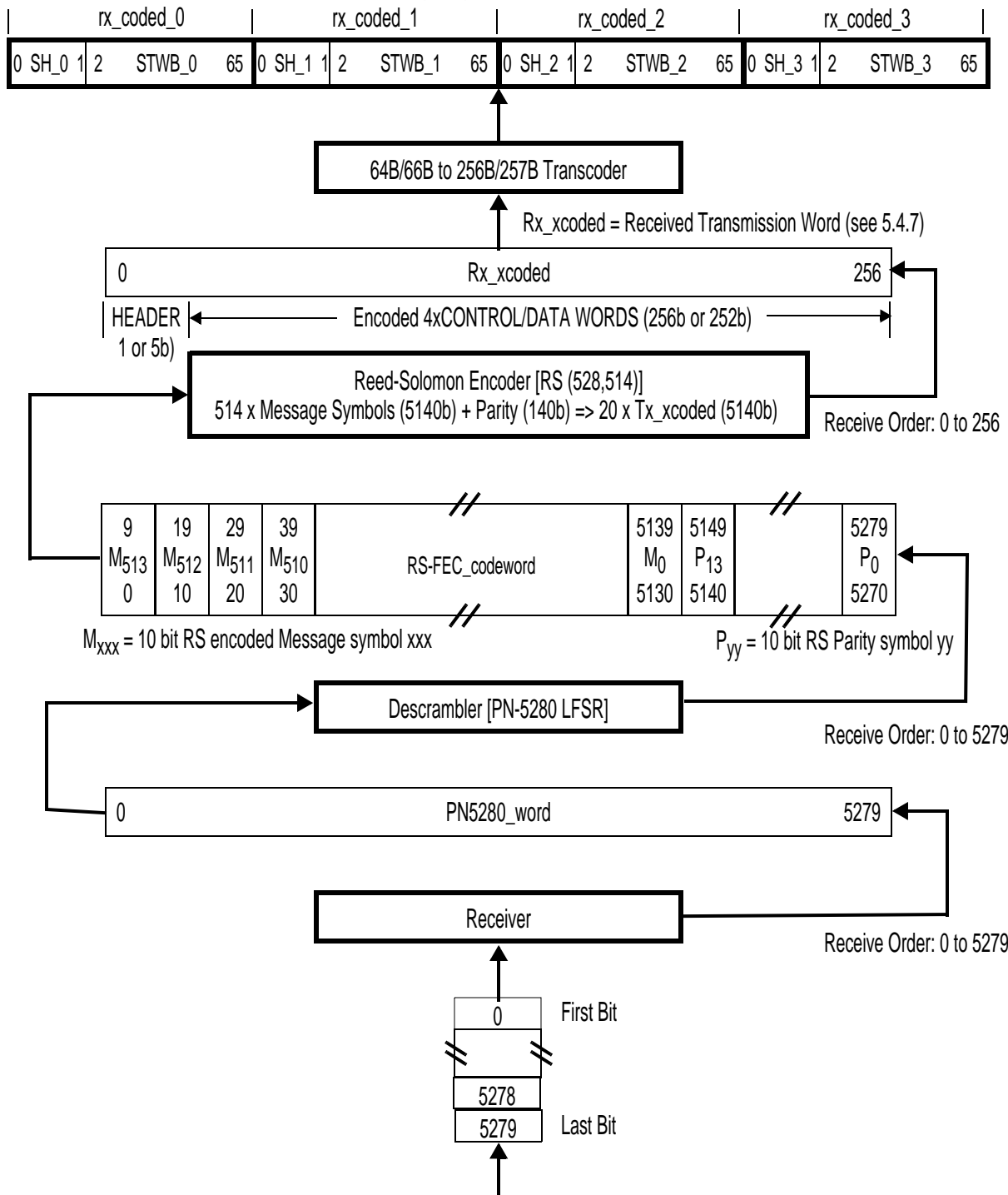


Figure 17 - 32GFC 256B/257B receive bit ordering

5.5 64GFC 256B/257B transmission code

5.5.1 Overview

An FC-0 standard (e.g., FC-PI-7) may specify the use of the 64GFC 256B/257B transmission code as its frame transfer transmission code. If the 64GFC 256B/257B transmission code is specified, then it shall be:

- a) generated as described in 5.5.2 and 5.5.3;
- b) encoded with Reed Solomon coding as described in 5.5.4;
- c) Gray mapped ~~scrambled~~ as described in 5.5.5;
- d) when enabled, precoded as described in 5.5.6;
- e) when enabled, inverse precoded as described in 5.5.7;
- f) inverse Gray mapped ~~descrambled~~ as described in 5.5.8;
- g) decoded with ~~the~~ Reed Solomon decoder as described in ~~5.5.14~~ 5.5.10; and
- h) recovered ~~decoded~~ as described in ~~5.4.7~~ 5.5.11 and 5.5.12.

5.5.2 64B/66B to 64GFC 256B/257B ~~T~~transcoding

64B/66B to 256B/257B transcoding is done as specified in 5.8.2.1.

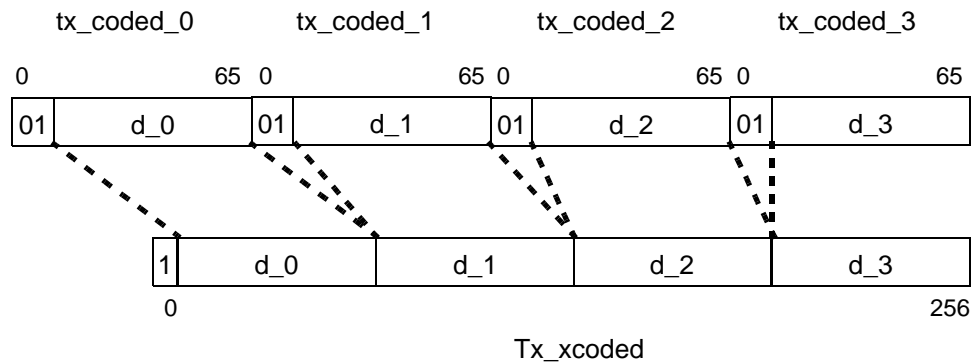
~~The 64GFC 256B/257B transmission code specified by this standard operates on 4 consecutive 64B/66B Transmission Words (see 5.3), each group being encoded as a 257-bit Transmission Word.~~

~~NOTE 2 - The IEEE 802.3bj-2014 specification of 256B/257B references as "blocks" what this standard references as "Transmission Words".~~

~~The transcoder constructs a 257-bit Transmission Word from a group of 4 x 66-bit Transmission Words to allocate bandwidth for the parity check symbols added by the Reed-Solomon encoder.~~

~~The 257-bit Transmission Word tx_xcoded<256:0> shall be constructed as defined in IEEE 802.3bj-2014 91.5.2.5 given 4 x 66-bit Transmission Words denoted as tx_coded_j<65:0> where j=0 to 3. The first 5 bits of tx_xcoded<256:0> are not scrambled (i.e., the step that generates tx_scrambled<256:0> is not performed).~~

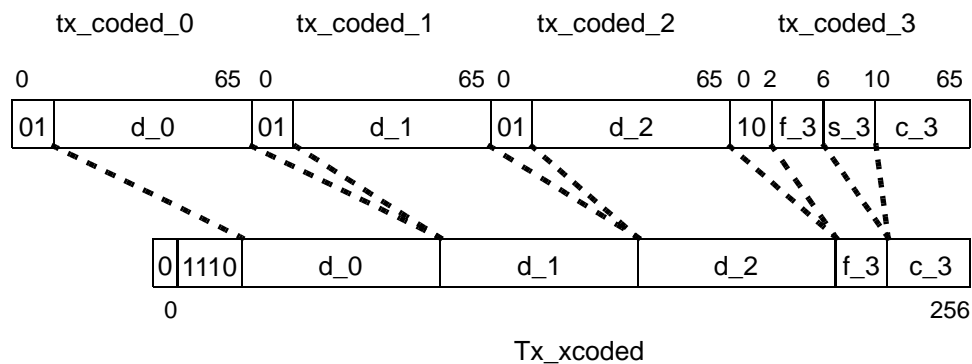
Figure 10 shows the 64GFC 256B/257B encoding of four data words.



Key:
 _x = data from the encoded 64/66b block

Figure 18 - 64GFC 256B/257B encoding of four data words

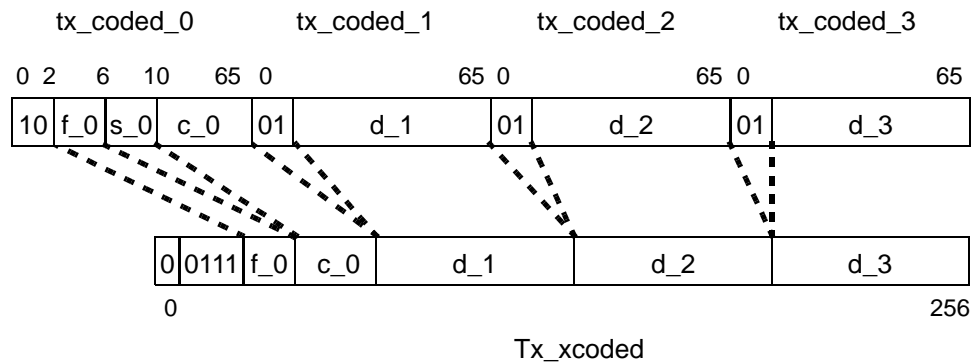
Figure 11 shows the 256B/257B encoding of three data words followed by one control word.



Key:
 d_x = data from the encoded 64/66b block
 c_x = control codes from the encoded 64/66b block
 f_x = first 4 bits of the block type field in the encoded 64/66b block
 s_x = second 4 bits of the block type field in the encoded 64/66b block

Figure 19 - 64GFC 256B/257B encoding of three data words followed by one control word

Figure 12 shows the 64GFC-256B/257B encoding of one control word followed by three data words.

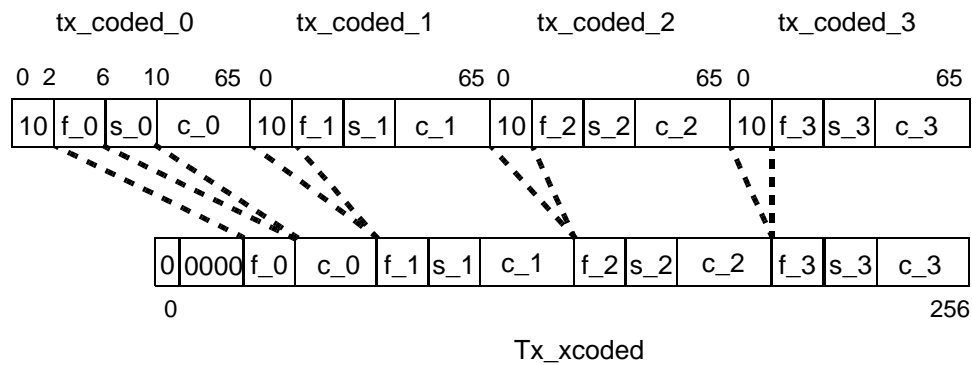


Key:

- d_x = data from the encoded 64/66b block
- c_x = control codes from the encoded 64/66b block
- f_x = first 4 bits of the block type field in the encoded 64/66b block
- s_x = second 4 bits of the block type field in the encoded 64/66b block

Figure 20 - 64GFC-256B/257B encoding of one control word followed by three data words

Figure 13 shows the 64GFC-256B/257B encoding of four control words.

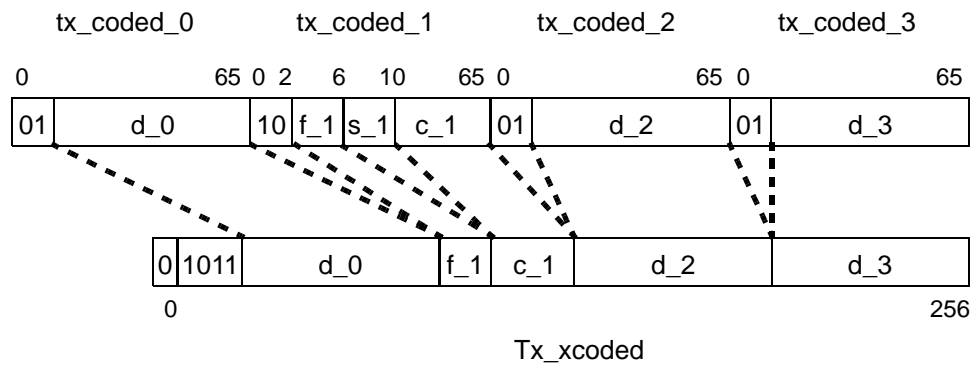


Key:

- d_x = data from the encoded 64/66b block
- c_x = control codes from the encoded 64/66b block
- f_x = first 4 bits of the block type field in the encoded 64/66b block
- s_x = second 4 bits of the block type field in the encoded 64/66b block

Figure 21 - 64GFC-256B/257B encoding of four control words

Figure 14 shows the 64GFC 256B/257B encoding of one data word followed by one control word followed by two data words:



Key:
 d_x = data from the encoded 64/66b block
 c_x = control codes from the encoded 64/66b block
 f_x = first 4 bits of the block type field in the encoded 64/66b block
 s_x = second 4 bits of the block type field in the encoded 64/66b block

Figure 22 - 64GFC 256B/257B encoding of one data word, followed by one control word, followed by two data words

A stream of 64GFC 256B/257B Transmission Words on a link shall be further encoded to provide Forward Error Correction (i.e., FEC).

The streams of 64GFC 256B/257B Transmission Words in both directions on the link shall be encoded as specified in 5.4 and then further encoded as specified in subclause 91.5.2.7 of IEEE 802.3bj-2014.

5.5.3 Alignment marker mapping and insertion

The alignment insertion function inserts a unique data pattern (i.e., Alignment Marker) into the data stream to provide a framing pattern for aligning the FEC code words.

The first 257b of every 1024th FEC code word carries Alignment Marker information. The first 256 bits (0 to 255) of the Alignment Marker (AM) vector are composed of AM0, AM4, AM8 and AM12 from Table 82-2 of IEEE 802.3-2015. AMx is the marker for PCS lane number x in Table 82-2. AM0 is the first marker transmitted on the line, followed by AM4, AM8 and AM12. Each octet in the AM vector is transmitted LSB(rightmost bit) to MSB (leftmost bit) starting from the leftmost octet to the rightmost octet. The last bit (256) is a pad bit that is always transmitted as zero.

The BIP3 octet of AM0 carries Link degrade information associated with the link degrade function (see 5.5.10.1). The bits of this octet are assigned as follows:

BIP3[0]=RD, BIP3[3:1]=0 (Reserved for future use; Transmitted as zero) and BIP3[7:4]=0xA

The RD (Remote Degrade) indicator is in bit 24 of the AM[256:0] vector. The BIP7 octet is the bit-wise inverse of BIP3 but conveys no useful information. As an example, the first 32 bits of the AM vector are ordered as follows:

1000 0011 0001 0110 1000 0100 RD000 0101

For all other markers besides AM0, BIP3=0xAA and BIP7=0x55.

5.5.4 Reed-Solomon encoder

The RS-FEC sublayer employs a Reed-Solomon code (see bibliography Annex M) operating over the Galois Field $GF(2^{10})$ (see bibliography Annex M) where the symbol size is 10 bits. The encoder processes k message symbols to generate $2t$ parity symbols which are then appended to the message to produce a code word of $n=k+2t$ symbols. For the purposes of this clause, a particular Reed-Solomon code is denoted $RS(n, k)$.

The RS-FEC sublayer shall implement $RS(544~~28~~, 514)$ as defined in IEEE 802.3cd D2.2 134.5.2.7. Each k -symbol message corresponds to twenty 257-bit Transmission Words produced by the transcoder. Each code is based on the generating polynomial given by Equation 91–1 of IEEE 802.3-2015.

5.5.5 ~~Scrambler~~Gray mapping

The Gray mapping process shall map consecutive pairs of bits {A, B}, where A is the bit arriving first, to a Gray-coded symbol as follows:

{0, 0} maps to 0.

{0, 1} maps to 1.

{1, 1} maps to 2, and

{1, 0} maps to 3.

~~Each RS-FEC code word is scrambled with a known sequence to randomize the 257-bit Transmission Word headers and to enable robust code word synchronization at the receiver (i.e., ensure that any shifted input bit sequence is not equal to another RS-FEC code word). Scrambling is implemented as module 2 addition of the RS-FEC code word and a pseudo-noise sequence 5280 bits in length defined as PN-5280 (see figure 15).~~

~~PN-5280 is generated by the polynomial $r(x)$.~~

$$~~r(x) = x^{39} + x^{58} + 1~~$$

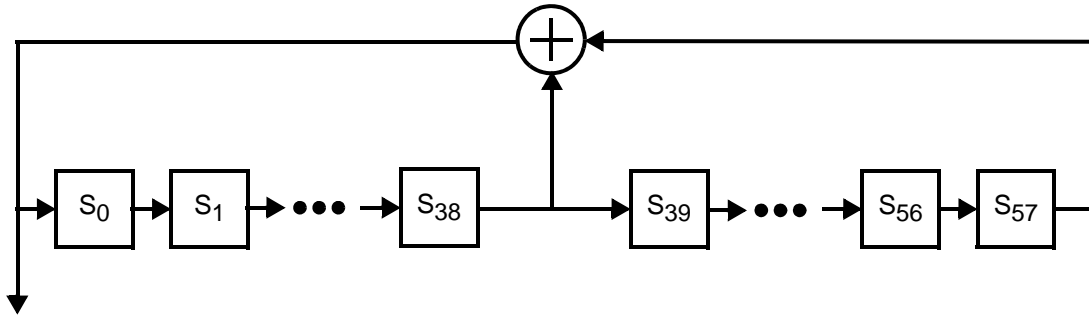


Figure 23 - ~~PN-5280~~ as a linear feedback shift register

At the start of each RS-FEC code word, the initial state of the pseudo-noise generator is set to:-

$$S_{57} = 1$$

$$S_{i-1} = S_i \text{ XOR } 1$$

(i.e., a binary sequence of alternating 1's and 0's).

5.5.6 ~~Descrambler~~ Precoding

The transmit process shall provide $1/(1+D)$ mod 4 precoding capability for electrical variants specified in FC-PI-7.

For each Gray-coded symbol $G(j)$, a precoded symbol $P(j)$ shall be determined by the following algorithm, where j is an index indicating the symbol number:

$$P(j) = (G(j) - P(j-1)) \text{ mod } 4$$

The decision to enable or disable transmitter precoding is determined by the remote receiver during the transmitter training process (refer to section 5.7 Transmitter Training Signal (TTS) for 64GFC Transmitter Training for details).

Each code word shall be descrambled prior to decoding. Descrambling is implemented as the modulo-2 addition of RS-FEC code word and the same pseudo-noise sequence PN-5280 defined for the scrambler (see 5.4.4).

5.5.7 Inverse precoding

The receive process optionally provides inverse precoding for electrical variants specified in FC-PI-7. When implemented and enabled, for each precoded symbol $P(j)$, a Gray-code symbol $G(j)$ shall be determined by the following algorithm:

$$G(j) = (P(j) + P(j-1)) \text{ mod } 4$$

When implemented, the decision to enable or disable remote transmitter precoding and receiver inverse precoding is determined by the receiver during the transmitter training process. The method by which the receiver determines whether or not to enable precoding is implementation dependent and beyond the scope of this standard.

5.5.8 Inverse Gray mapping

The inverse Gray mapping process shall map Gray-coded PAM4 symbols to pairs of bits {A, B} where A is considered to be the first bit as follows:

0 maps to {0, 0},

1 maps to {0, 1},

2 maps to {1, 1}, and

3 maps to {1, 0}.

5.5.9 Alignment lock

The receive function obtains LOCK to the alignment markers as specified by the FEC synchronization state diagram in IEEE802.3cd D2.1 Figure 91-8, using the variable and counter definitions from IEEE802.3cd 134.5.4 but modified for a single FEC lane operation.

5.5.10 Reed-Solomon decoder

5.5.10.1 Overview

The Reed-Solomon decoder extracts the message symbols from the code word, correcting them as necessary, and discards the parity symbols. The message symbols correspond to 20 x 257-bit Transmission Words.

The Reed-Solomon decoder shall be capable of correcting any combination of up to $t=715$ symbol errors in a code word. It shall also be capable of indicating when a code word contains errors but was not corrected (e.g., it contains a number of errors in excess of the error correction capability).

5.5.10.2 Link Degrade Signaling

For 64GFC links, Link Degrade Signaling can be supported by monitoring errors in the FEC logic. The Link Degrade Logic keeps track of the following parameters:

FEC Degrade interval – This is a 32 bit register that specifies the number of RS-FEC code words that make up a Degrade Interval. Bit 0 of this register is ignored so the number of FEC code words within a Degrade Interval is always even.

RD – Remote Degrade Bit to be sent in the Alignment Marker field.

Degrade Activate Threshold – This is a 32 bit register that specifies a symbol error count. The value here controls the threshold used to activate RD.

Degrade Deactivate Threshold – This is a 32 bit register that specifies a symbol error count. The value here controls the threshold used to deactivate RD.

The Reed Solomon Decoder counts the number of symbol errors detected in all the code words within the FEC degrade interval. If a codeword is uncorrectable, the number of symbol errors detected is incremented by 16. When the number of symbol errors detected within a FEC Degrade interval exceeds the FEC degrade activate threshold, RD(Remote Degrade) will be signaled to the remote link partner using a bit in the Alignment Marker. At the end of an interval, if the number of symbol errors is less than the FEC degrade deactivate threshold, RD will be de-asserted in the Alignment Marker.

5.5.11 Alignment marker removal

The first 257 message bits in every 1024th codeword is the vector am_rxmapped<256:0> where bit 0 is the first bit received. The specific codewords that include this vector are indicated by the alignment lock function (see 5.5.9).

The vector am_rxmapped shall be removed prior to transcoding.

5.5.12 ~~64GFC~~ 256B/257B to 64B/66B transcoder

The first five bits of the received block rx_scrambled<256:0>, in reception order, are descrambled Rx_scrambled<256:0> will yield rx_coded<256:0> as follows:

- a) set rx_coded<4:0> to the result of the bit wise Exclusive-OR of rx_scrambled<4:0> and rx_scrambled<12:8>; and
- b) set rx_coded<256:5> to rx_scrambled<256:5>.

Next, a group of four 66 bit transmission words are constructed from each received 257 bit transmission word as specified in 5.4.7.

~~The transcoder reconstructs a group of 4 x 66-bit Transmission Words from each received 257-bit Transmission Word.~~

~~The 4 x 66-bit Transmission Words, denoted as rx_coded_j<65:0> where j=0 to 3, shall be derived from each 257-bit Transmission Word rx_xcoded<256:0> as defined in IEEE 802.3bj 2014 91.5.3.5. As the first 5 bits of rx_xcoded<256:0> are not scrambled, the step defined in 802.3bj that derives rx_xcoded from rx_scrambled is not performed on those bits.~~

5.5.13 Transmit Bit Ordering

Transmit bit ordering for 64GFC 256B/257B is as shown in figure 24.

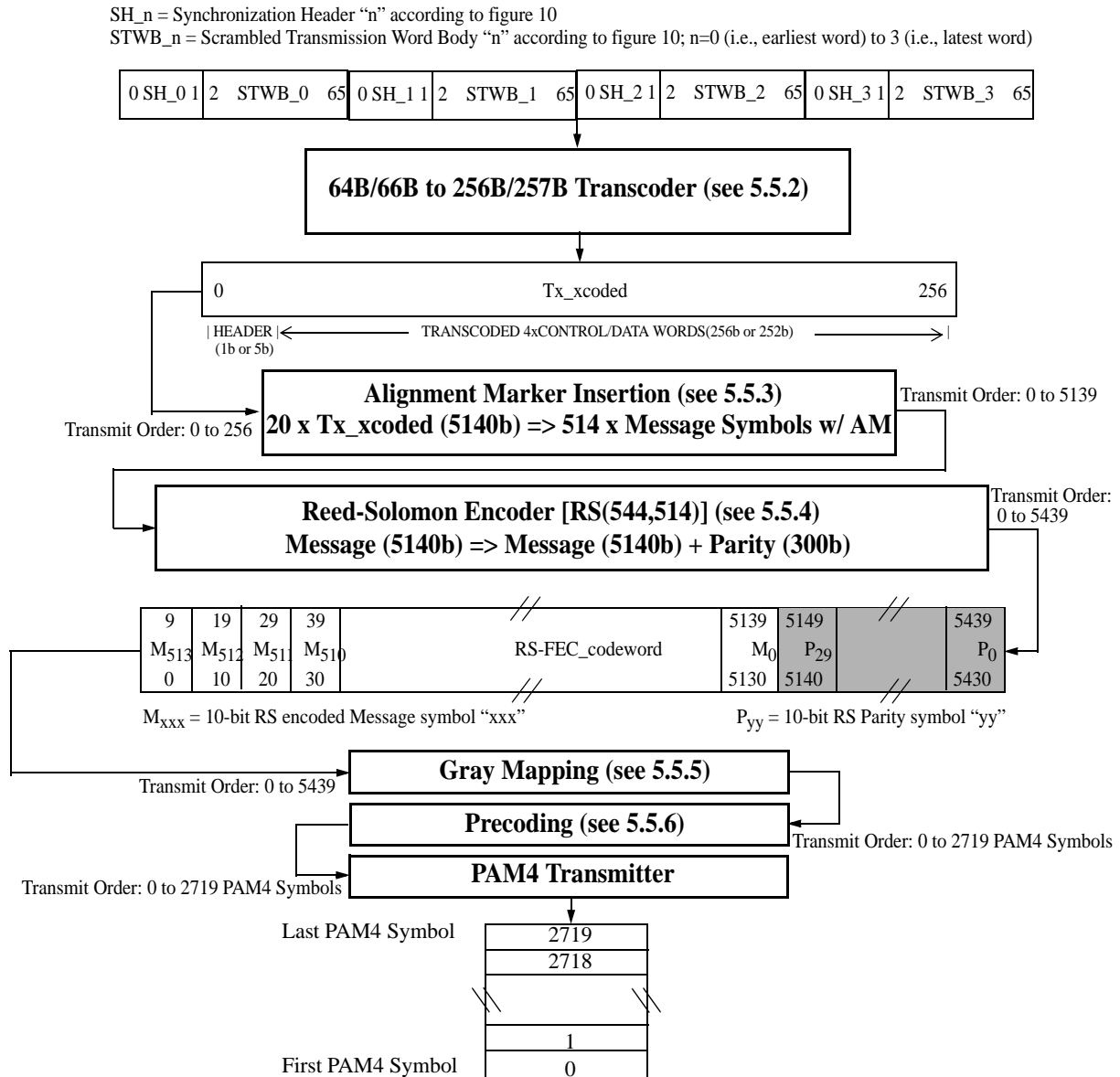


Figure 24 - 64GFC 256B/257B transmit bit ordering

5.5.14 Receive Bit Ordering

Receive bit ordering for 64GFC 256B/257B is as shown in figure 25.

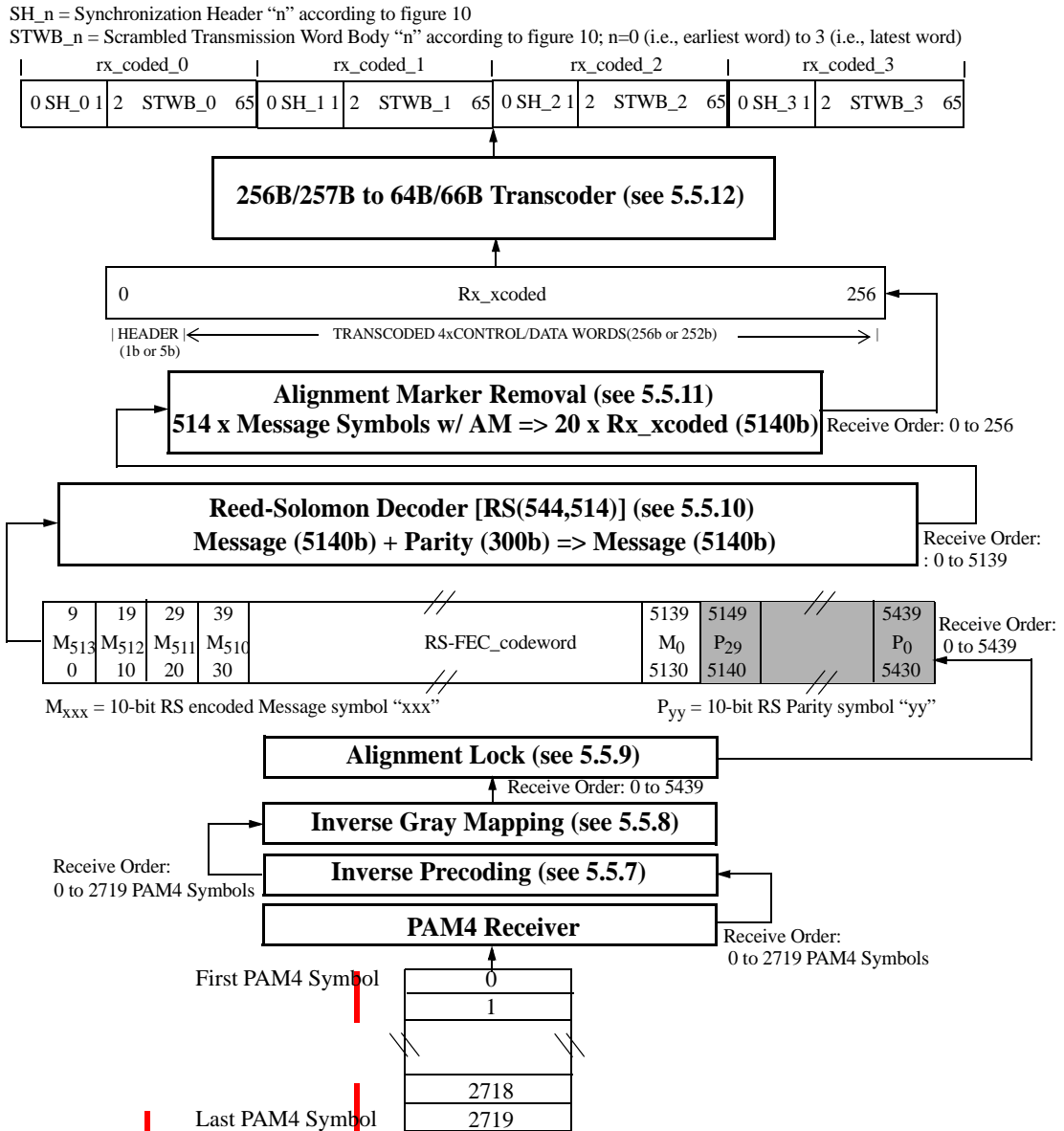


Figure 25 - 64GFC 256B/257B receive bit ordering

5.6 Transmitter Training Signal (TTS) for LSN and 32GFC/16GFC Transmitter Training

5.6.1 Overview

An FC-0 standard (e.g., FC-PI-5) may specify the use of the Transmitter Training Signal. The Transmitter Training Signal shall not be used for communication of Fibre Channel frames.

The Transmitter Training Signal is a transmission code that enables active feedback from a receiver to a transmitter to assist in adapting the transmitter to the characteristics of the link that connects them. Adjustable transmitter coefficients are supported. The use and effect of each coefficient is specified in FC-PI-x. It is expected that two FC_Ports on a link will concurrently send the Transmitter Training Signal allowing each FC_Port to evaluate the received signal quality and recommend adjustments to the transmitter of the other FC_Port. The Transmitter Training Signal may be sent to communicate information without doing transmitter training.

The Transmitter Training Signal allows enabling of Forward Error Correction (FEC) (see 5.3). FEC is optional for 16GFC and mandatory for 32GFC. FEC negotiation is not performed for 32GFC links and 128GFC links (i.e., four parallel lanes of 32GFC in each direction). The Transmitter Training Signal allows enabling parallel lane support (see table 2) by setting Training Frame Control field bit 10 to one, if a lane is capable of running at 32GFC speeds.

The Transmitter Training Signal shall be a repeating series of Transmission Words, each containing two elements (see figure 26):

- 1) A Training Frame (see 5.6.2), which carries recommended adjustments to the transmitter of the receiving FC_Port based on the quality of the signal detected at the receiver of the sending FC_Port. The information in the Training Frame is encoded so as to increase its likelihood of reliable communication when the transmitter is not optimally adjusted for the link; and
- 2) A Training Pattern (see 5.6.3), which allows the receiving FC_Port to formulate recommended adjustments to the transmitter of the sending FC_Port. The Training Pattern is encoded so as to challenge the ability to reliably recover it when the transmitter is not optimally adjusted for the link.

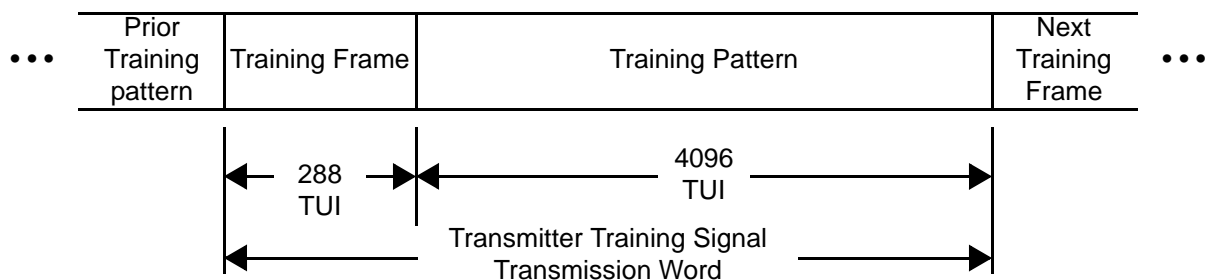
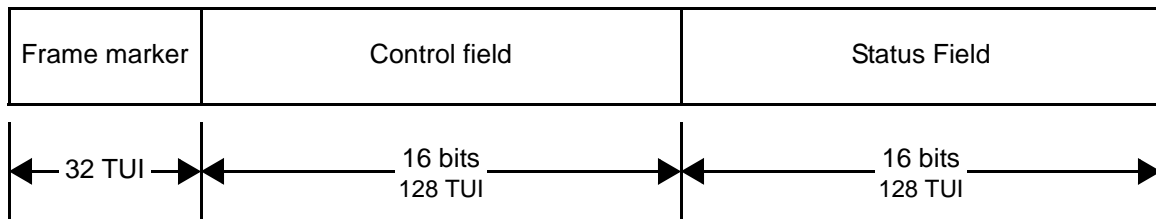


Figure 26 - Transmitter Training Signal

5.6.2 Training Frame

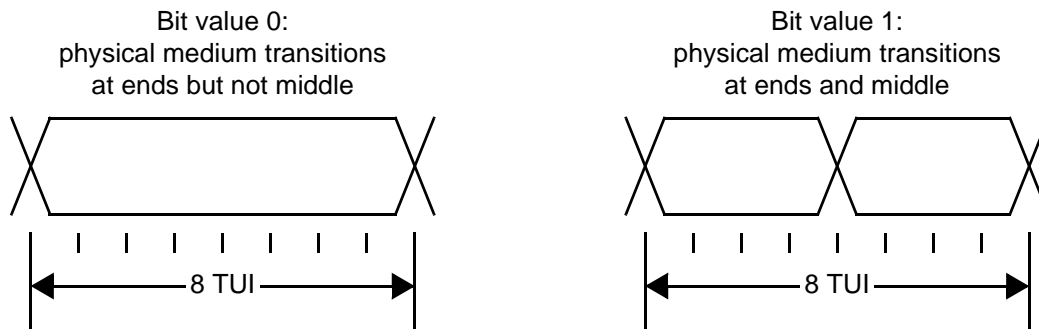
The Training Frame is the element of a Transmitter Training Signal that communicates training information from a receiver to a transmitter. A Training Frame comprises a 32 TUI frame marker followed by a 128 TUI Control field followed by a 128 TUI Status field (see figure 27).



NOTE Each bit of information in the Control field and the Status field is differential Manchester coded in an 8 TUI interval.

Figure 27 - Training Frame format

The Training Frame is intended to communicate information if the transmitter is not optimally adjusted for the link and the selected link speed. The Training Frame also carries information as to whether the physical interface supports parallel lanes and whether FEC is supported. Information in the Training Frame shall be encoded using differential Manchester coding at one eighth the nominal bit rate of the selected link speed (see figure 28).



NOTE Each bit of information in the Control field and the Status field is differential Manchester coded in an 8 TUI interval.

Figure 28 - Differential Manchester coding

The beginning of a Training Frame shall be signaled by a frame marker. A frame marker shall be transmitted by holding the physical medium signal at logical "1" for 16 TUI followed by holding the physical medium at logical "0" for 16 TUI. This is a deliberate violation of one eighth rate differential Manchester coding, and carries no information (see figure 29).

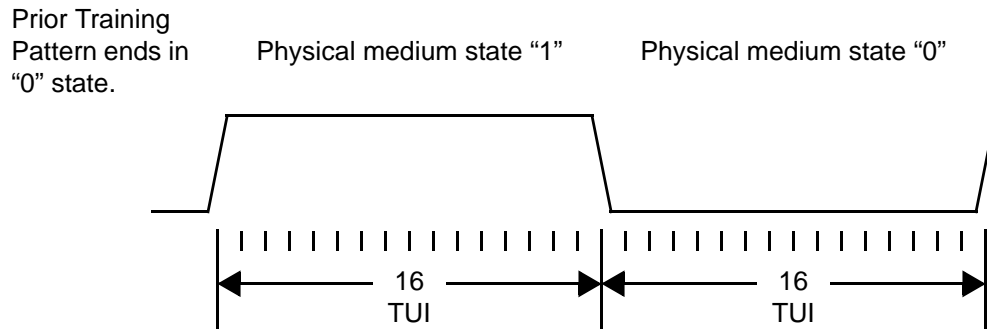


Figure 29 - Frame marker signal

The Control field and the Status field each contain 16 bits of information (i.e., each contain 128 TUI of differential Manchester coded information). The information in these fields shall be transmitted so that more significant encoded information bits are transmitted before less significant encoded information bits. The electrical characteristics of the Transmitter Training Signal are specified in an FC-0 standard, and when indicated in this standard, are indicated informatively.

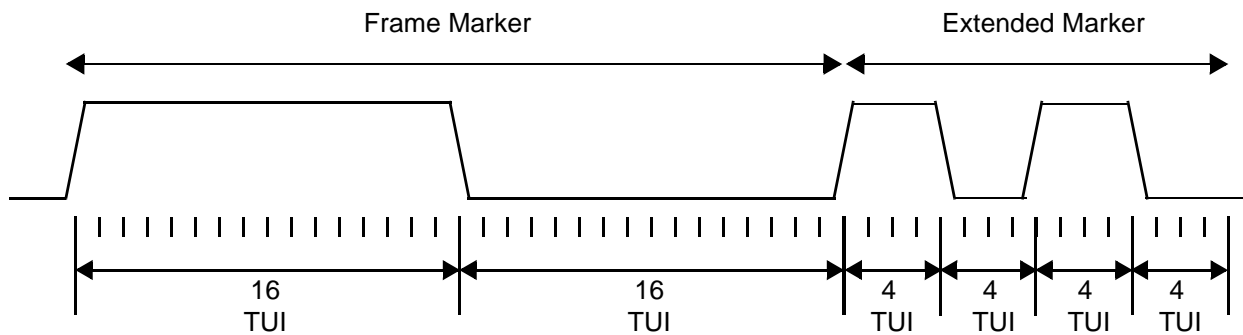


Figure 30 - 32GFC frame marker signal

An extended marker was specified in the Training Frame Control field for 32GFC since the 16GFC Training Frame Control field could be incorrectly recognized as the 32GFC frame marker and a 32GFC port could synchronize on the 16GFC Training Frame Control field. The extended marker is for 16 TUI as shown in figure 30 of alternating highs and lows to uniquely identify 32GFC. 32GFC locks onto the frame marker plus extended marker to preclude the potential of a false lock at 16GFC speeds. The extended marker shall be transmitted after the frame marker whenever a 32GFC Training Frame is transmitted.

Fields in the Control field shall be set as specified in table 2. Fields in the Status field shall be set as specified in table 3. See clause 9 For the use of these fields.

Table 2 - Training Frame Control field

Bits	Field name	Content
15-14	Extended Marker	Set to 11b: Extended marker for 32GFC. Set to 10b: reserved Extended marker for 64GFC. Set to 01b: reserved. Set to 00b: for 16GFC.
13	Preset	Set to one: the transmitter should set all coefficients to preset values. Set to zero: no transmitter action advised.
12	Initialize	Set to one: The Transmitter should set all coefficients to initialize values. Set to zero: no transmitter action.
11	FECReq	Set to one: the FC_Port is requesting the use of Forward Error Correction (FEC) (see 5.3) in association with 64B/66B. Set to zero: the FC_Port is directing not to use Forward Error Correction (FEC) in association with 64B/66B.
10	Parallel Lane Support	Set to one: parallel lanes are supported. Set to zero: parallel lanes are not supported.
9-6		Reserved
5-4	C1Upd	Set to 11b: reserved. Set to 10b: transmitter should decrement coefficient 1 one step. ^a Set to 01b: transmitter should increment coefficient 1 one step. ^a Set to 00b: transmitter should not change coefficient 1.
3-2	C0Upd	Set to 11b: reserved. Set to 10b: transmitter should decrement coefficient 0 one step. ^a Set to 01b: transmitter should increment coefficient 0 one step. ^a Set to 00b: transmitter should not change coefficient 0.
1-0	C-1Upd	Set to 11b: reserved. Set to 10b: transmitter should decrement coefficient -1 one step. ^a Set to 01b: transmitter should increment coefficient -1 one step. ^a Set to 00b: transmitter should not change coefficient -1.
^a See FC-PI-5.		

Table 3 - Training Frame Status field

Bits	Field name	Content
15	TC	Set to one: transmitter training is complete. Set to zero: request to begin or continue transmitter training.
14	SN	Set to one: the transmitter is using and has not completed Speed Negotiation. Set to zero: the transmitter has completed or did not use Speed Negotiation.
13	FECCap	Set to one: FC_Port has Forward Error Correction (FEC) capability (see 5.3). Set to zero: FC_Port does not have Forward Error Correction (FEC) capability.
12	TF	Set to one: the transmitter is operating with fixed transmitter coefficients. Set to zero: the transmitter coefficients may be trained by the receiver.
11-6		Reserved
5-4	C1Stat	Set to 11b: transmitter coefficient 1 acknowledges an update that left it at its maximum value. ^a Set to 10b: transmitter coefficient 1 acknowledges an update that left it at its minimum value. ^a Set to 01b: transmitter coefficient 1 acknowledges an update that is complete. ^a Set to 00b: transmitter coefficient 1 is ready for another update.
3-2	C0Stat	Set to 11b: transmitter coefficient 0 acknowledges an update that left it at its maximum value. Set to 10b: transmitter coefficient 0 acknowledges an update that left it at its minimum value. ^a Set to 01b: transmitter coefficient 0 acknowledges an update that is complete. ^a Set to 00b: transmitter coefficient 0 is ready for another update.
1-0	C-1Stat	Set to 11b: transmitter coefficient -1 acknowledges an update that left it at its maximum value. ^a Set to 10b: transmitter coefficient -1 acknowledges an update that left it at its minimum value. ^a Set to 01b: transmitter coefficient -1 acknowledges an update that is complete. ^a Set to 00b: transmitter coefficient -1 is ready for another update.
^a See FC-PI-5.		

5.6.3 Training Pattern

The Training Pattern is the element of a Transmitter Training Signal that allows a receiver to evaluate its ability to achieve reliable Fibre Channel communication across the link on which the Training Pattern is sent. The Training Pattern shall be composed of 4094 TUI of PRBS-11 followed by two TUI of zero. PRBS-11 (see figure 31) shall be equivalent to the output of an 11-bit linear feedback shift register that is initialized to a value that is randomized to a non-zero value for each training frame, and that implements the polynomial

$$x^{11} + x^9 + 1$$

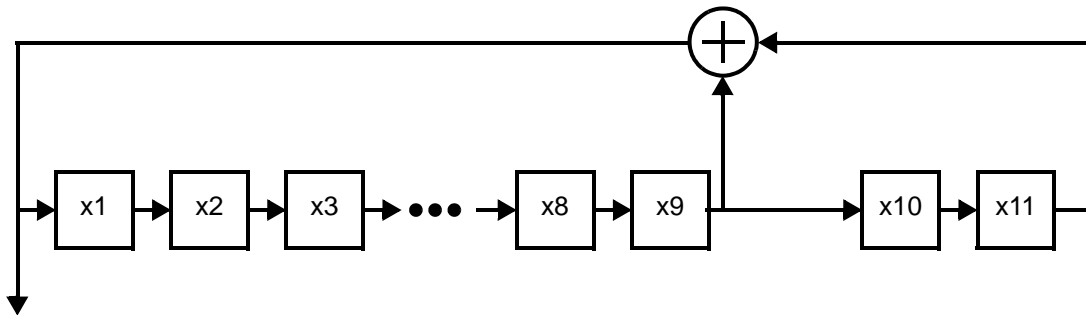


Figure 31 - PRBS-11 as a linear feedback shift register

5.7 Transmitter Training Signal (TTS) for 64GFC Transmitter Training

The training frame structure is specified in IEEE 802.3-2015 136.8.11.1.

The training frame marker is specified in IEEE 802.3-2015 136.8.11.1.1.

Control and status field behavior is specified in IEEE 802.3-2015 136.8.11.1.2. The Control field is specified in table 4.

Table 4 - 64GFC Training Frame Control field

<u>Bits</u>	<u>Field name</u>	<u>Content</u>
<u>15-14</u>	<u>Extended Marker</u>	Set to 11b: Extended marker for 32GFC. Set to 10b: Extended marker for 64GFC. Set to 01b: reserved. Set to 00b: for 16GFC.
<u>13-12</u>	<u>Initial Condition request</u>	Set to 11b: Preset 3. Set to 10b: Preset 2. Set to 01b: Preset 1. Set to 00b: Individual Coefficient Control.
<u>11</u>	<u>Reserved</u>	Transmit as zero, ignore on receipt.
<u>10</u>	<u>Parallel Lane Support</u>	Set to one: parallel lanes are supported. Set to zero: parallel lanes are not supported.
<u>9-8</u>	<u>Modulation and Precoding request</u>	Set to 11b: PAM4 with precoding. Set to 10b: PAM4 Set to 01b: reserved Set to 00b: PAM2.
<u>7-5</u>	<u>Reserved</u>	Transmit as zero, ignore on receipt.
<u>4-2</u>	<u>Coefficient Select</u>	Set to 110b: c(-2) Set to 111b: c(-1) Set to 000b: c(0) Set to 001b: c(1)
<u>1-0</u>	<u>Coefficient Request</u>	Set to 11b: No equalization Set to 10b: Decrement Set to 01b: Increment Set to 00b: Hold
a		

The Status field is specified in table 5.

Table 5 - 64GFC Training Frame Status field

<u>Bits</u>	<u>Field name</u>	<u>Content</u>
<u>15</u>	<u>Receiver Ready</u>	<u>Set to one: training is complete and receiver is ready for data.</u> <u>Set to zero: request for Training to continue.</u>
<u>14</u>	<u>SN</u>	<u>Set to one: transmitter has not completed LSN.</u> <u>Set to zero: transmitter has completed LSN.</u>
<u>13</u>	<u>Reserved</u>	<u>Transmit as zero, ignore on receipt.</u>
<u>12</u>	<u>TF</u>	<u>Set to one: transmitter is operating with Fixed Coefficients.</u> <u>Set to zero: transmitter coefficients may be trained by the receiver.</u>
<u>11-10</u>	<u>Modulation and Precoding Status</u>	<u>Set to 11b: PAM4 with precoding.</u> <u>Set to 10b: PAM4.</u> <u>Set to 01b: reserved</u> <u>Set to 00b: PAM2</u>
<u>9</u>	<u>Receiver frame lock</u>	<u>Set to one: frame boundaries identified.</u> <u>Set to zero: frame boundaries not identified.</u>
<u>8</u>	<u>Initial Condition Status</u>	<u>Set to one: updated.</u> <u>Set to zero: not updated.</u>
<u>7</u>	<u>Parity</u>	<u>Parity bit to provide DC balance.</u>
<u>6</u>	<u>Reserved</u>	<u>Transmit as zero, ignore on receipt.</u>
<u>5-3</u>	<u>Coefficient Select Echo</u>	<u>Set to 110b: c(-2)</u> <u>Set to 111b: c(-1)</u> <u>Set to 000b: c(0)</u> <u>Set to 001b: c(1)</u>
<u>2-0</u>	<u>Coefficient Status</u>	<u>Set to 111b: reserved</u> <u>Set to 110b: coefficient at limit and maximum voltage</u> <u>Set to 101b: reserved</u> <u>Set to 100b: maximum voltage</u> <u>Set to 011b: coefficient not supported</u> <u>Set to 010b: coefficient at limit</u> <u>Set to 001b: updated</u> <u>Set to 000b: not updated</u>
a		

The training pattern is specified in IEEE 802.3-2015 136.8.11.1.3.

The zero pad is specified in IEEE 802.3-2015 136.8.11.1.4.

5.8 FEC for 128GFC

5.8.1 Overview

This clause specifies how Forward Error Correction (FEC) is implemented on 128GFC ports. FEC usage is mandatory on 128GFC ports. Streams of 64/66B Transmission Words in both directions on a 128GFC link are encoded by the FEC layer as specified below.

5.8.2 Functional block diagram

A functional block diagram of the 128GFC RS-FEC sub layer is shown in figure 32.

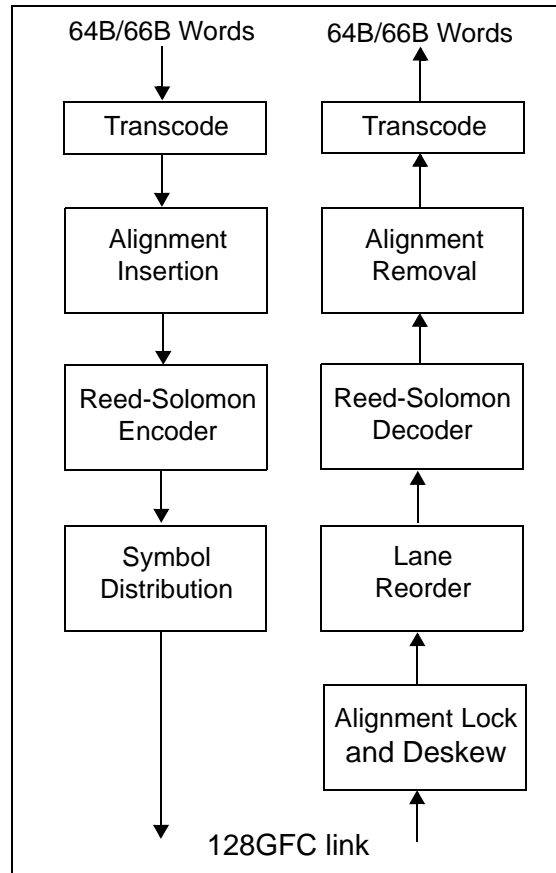


Figure 32 - 128GFC RS-FEC sub layer functional block diagram

5.8.2.1 64B/66B to 256B/257B Transcoder

Transcoding is done as specified in 5.4.2.

In addition, as a final step, the first five bits are scrambled in transmission order as specified in IEEE 802.3bj-2014 91.5.2.5.

After this step, $tx_xcoded\langle 256:0 \rangle$ will yield $tx_scrambled\langle 256:0 \rangle$ as follows:

- a) Set $tx_scrambled\langle 4:0 \rangle$ to the result of the bit wise Exclusive-OR of $tx_xcoded\langle 4:0 \rangle$ and $tx_xcoded\langle 12:8 \rangle$; and
- b) Set $tx_scrambled\langle 256:5 \rangle$ to $tx_xcoded\langle 256:5 \rangle$.

5.8.2.2 Alignment marker mapping and insertion

The alignment insertion function inserts a unique data pattern (i.e., Alignment Marker) for each link into the data stream to enable identification of which of the four links is which FEC lane. This function enables the receiver to map the physical links to logical lanes allowing for random connections of the Transmit links to the Receive links within the group of 4 links, in addition to providing a framing pattern for aligning the FEC code words.

The first 514b of every 4096th FEC code word carries Alignment Marker information.

The alignment marker bit sequence is identical to the first two re-mapped AM TC blocks specified in Clause 82.2.7 and Clause 91.5.2.6 when replacing the BIP3 field in all four instances of the AM0 blocks with the value 0xCA, the BIP3 for AM4 with 0x9D, the BIP3 for AM5 with 0xD7, the BIP3 for AM6 with 0x6F, and the BIP3 for AM7 with 0xA1. Additionally the first bit of AM8 and AM9 that are part of the sequence is changed from 0->1 to maintain DC balance.

Table 6 shows the data stream that will appear on each of the 4 lanes after the RS symbol distribution of the AM pattern is done. The 'd' is the first 6b of data from TC block that follows the AM pattern. The underlined values are the replaced BIP3 and BIP7 fields in the AM blocks.

Table 6 - 128GFC FEC Alignment Marker

AM bits	Lane3	Lane2	Lane1	Lane0
[39:0]	0011000001	0011000001	0011000001	0011000001
[79:40]	0001011010	0001011010	0001011010	0001011010
[119:80]	<u>0010100010</u>	<u>0010100010</u>	<u>0010100010</u>	<u>0010100010</u>
[159:120]	00111110 <u>11</u>	00111110 <u>11</u>	00111110 <u>11</u>	00111110 <u>11</u>
[199:160]	1010010111	1010010111	1010010111	1010010111
[239:200]	<u>0101110111</u>	<u>0101110111</u>	<u>0101110111</u>	<u>0101110111</u>
[279:240]	111011 <u>0011</u>	011010 <u>0011</u>	011101 <u>0011</u>	110101 <u>0011</u>
[319:280]	0100010101	0100101010	0001010011	0000011111
[359:320]	<u>0101100110</u>	<u>1100100110</u>	<u>1111000010</u>	<u>0100001001</u>
[399:360]	0100 <u>101000</u>	0101011011	0010110101	10101 <u>00111</u>
[439:400]	1110101000	1101010110	1010110010	1110000000
[479:440]	1001100110	1101100110	0011110111	1111011011
[513:480]	dddddd <u>1110</u>	01 <u>10010000</u>	01 <u>00101000</u>	01 <u>01100010</u>

5.8.2.3 Reed-Solomon encoder

Reed-Solomon encoding is done as specified in 5.4.3.

5.8.2.4 Symbol distribution

Once the data has been encoded, it is distributed to 4 lanes, in groups of 10 bit symbols.

Symbol distribution is done as specified in IEEE 802.3bj-2014 91.5.2.8.

5.8.2.5 Transmit bit ordering

Transmit bit ordering is as shown in figure 33.

5.8.2.6 Alignment lock and deskew

The receive function creates 4 bit streams after concatenating the bits received on each lane. It then obtains LOCK to the alignment markers on each lane as specified by the FEC synchronization state diagram in IEEE802.3bj-2014 91.5.3.1.

After alignment marker lock is achieved on all four lanes, all inter lane skew is removed as specified by the FEC alignment state diagram in IEEE802.3bj-2014 91.5.3.1. The FEC receive function will support a maximum skew of 180ns between lanes and a maximum skew variation of 4ns.

5.8.2.7 Lane reorder

FEC lanes may be received on different lanes of the service interface from which they were originally transmitted.

The FEC receive function shall order the FEC lanes according to the FEC lane number per IEEE802.3bj-2014-91.5.3.2. The FEC lane number is defined by the alignment marker that is mapped to each FEC lane.

After all FEC lanes are aligned, deskewed, and reordered, the FEC lanes are multiplexed together in the proper order to reconstruct the original stream of FEC code words.

5.8.2.8 Reed-Solomon decoder

Decoding is done as specified in 5.4.6.

5.8.2.9 Alignment marker removal

The first 514 bits in every 4096 code words are the mapped alignment marker bits. These are removed before sending the data to the transcode block.

5.8.2.10 256B/257B to 64B/66B transcoder

The first five bits of the of the received block $rx_scrambled\langle 256:0\rangle$, in reception order, are descrambled. $Rx_scrambled\langle 256:0\rangle$ will yield $rx_coded\langle 256:0\rangle$ as follows:

- a) Set $rx_coded\langle 4:0\rangle$ to the result of the bit wise Exclusive-OR of $rx_scrambled\langle 4:0\rangle$ and $rx_scrambled\langle 12:8\rangle$; and
- b) Set $rx_coded\langle 256:5\rangle$ to $rx_scrambled\langle 256:5\rangle$.

Next, a group of four 66bit transmission words are constructed from each received 257 bit transmission word as specified in 5.4.7.

5.8.2.11 Receive bit ordering

Receive bit ordering is as specified in figure 34.

SH_n = Synchronization Header n according to figure 10

STWB_n = Scrambled Transmission Word Body n according to figure 10; n = 0 (i.e., earliest word) to n = 3 (i.e., latest word)

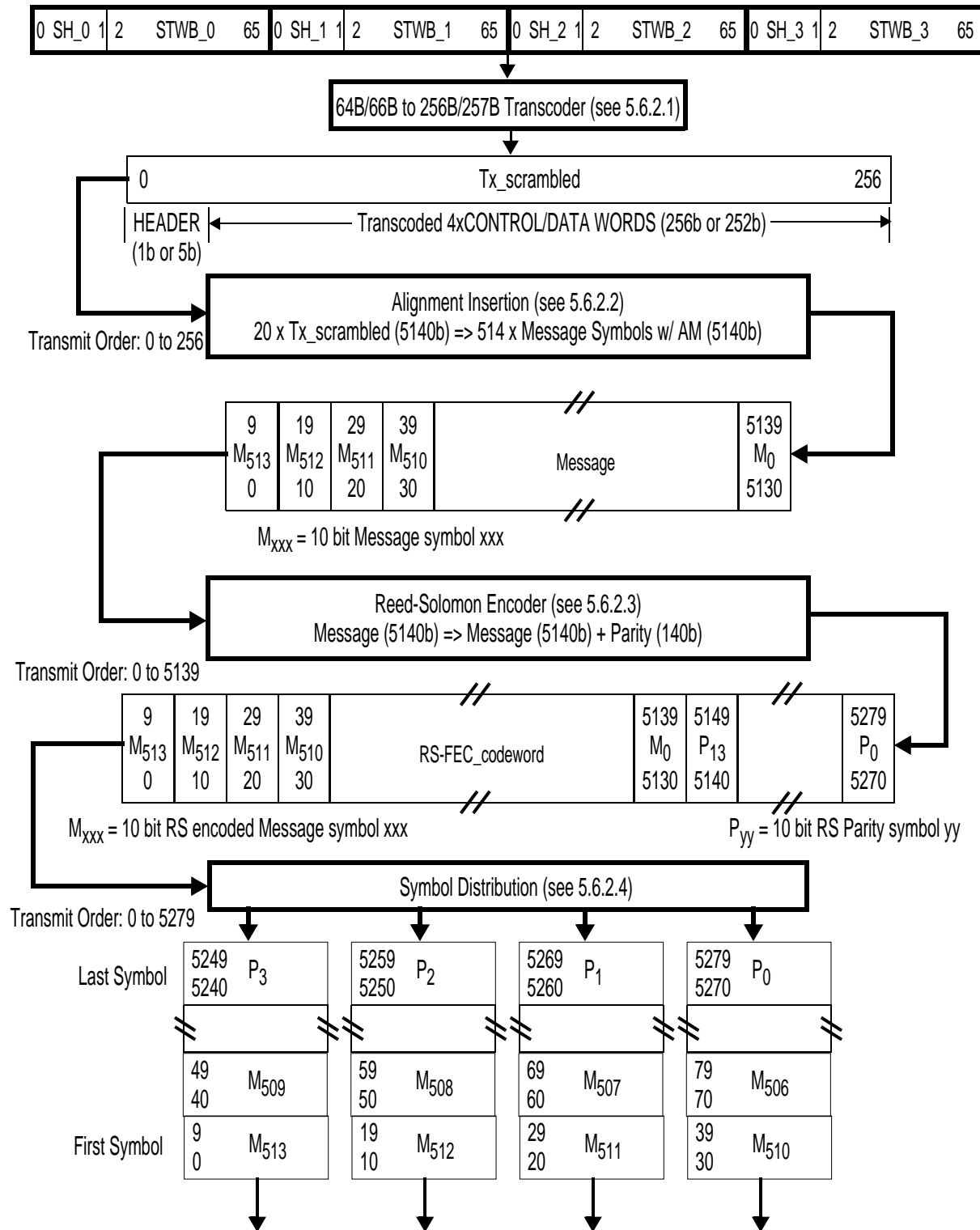


Figure 33 - Transmit bit ordering

SH_n = Synchronization Header n according to figure 10

STWB_n = Scrambled Transmission Word Body n according to figure 10; n = 0 (i.e., earliest word) to n = 3 (i.e., latest word)

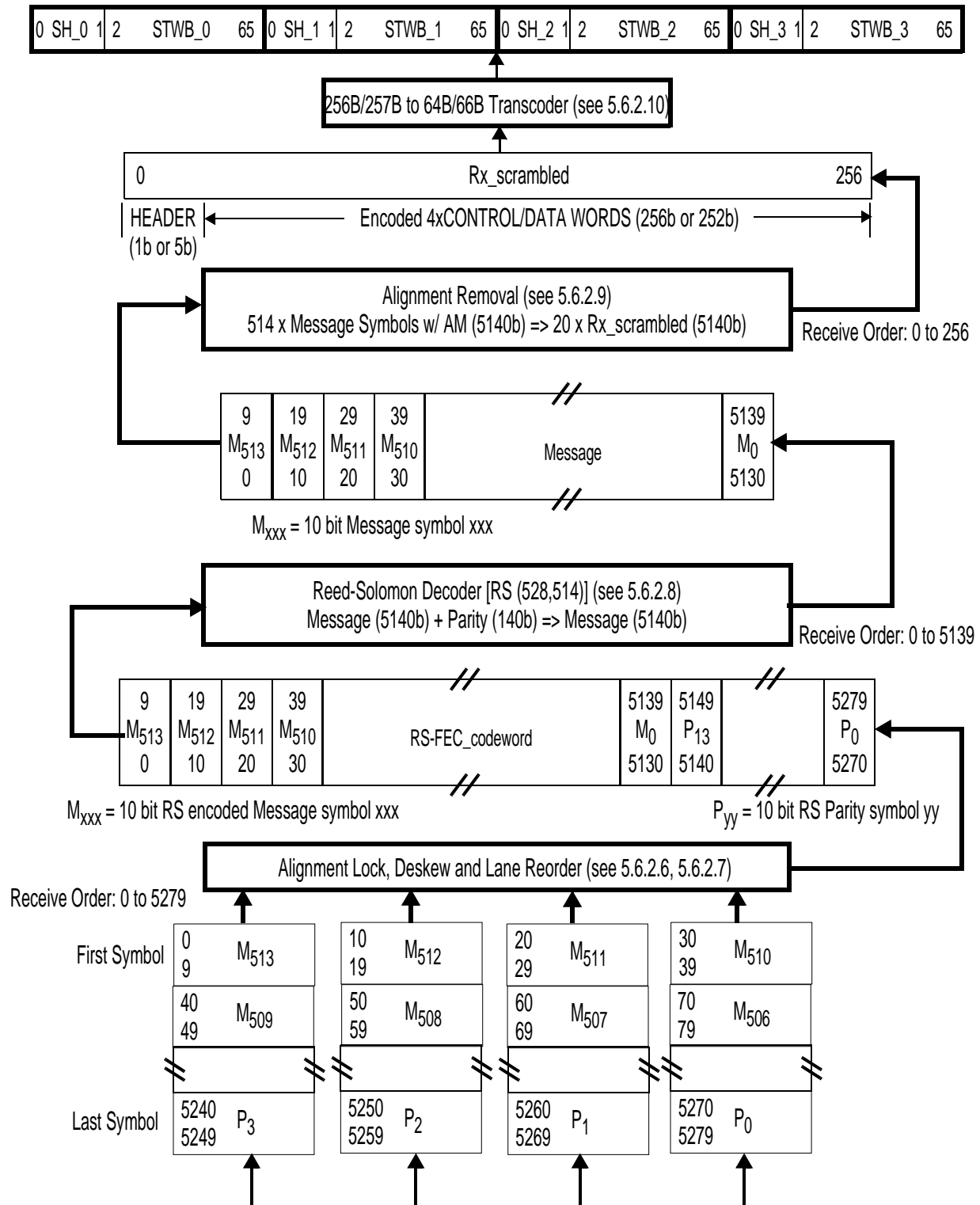


Figure 34 - Receive bit ordering

5.9 FEC for 256GFC

5.9.1 Overview

This clause specifies how Forward Error Correction (FEC) is implemented on 428256GFC ports. FEC usage is mandatory on 428256GFC ports. Streams of 64/66B Transmission Words in both directions on a 428256GFC link are encoded by the FEC layer as specified below.

5.9.2 Functional block diagram

A functional block diagram of the 428256GFC RS-FEC sub layer is shown in figure 35.

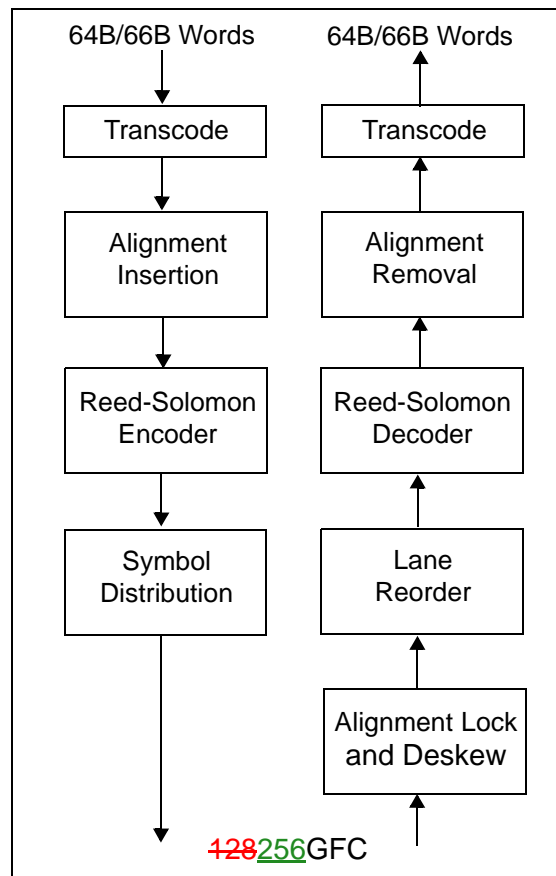


Figure 35 - 428256GFC RS-FEC sub layer functional block diagram

5.9.2.1 64B/66B to 256B/257B Transcoder

64B/66B to 256B/257B transcoding is done as specified in 5.8.2.1.

~~Transcoding is done as specified in 5.4.2.~~

~~In addition, as a final step, the first five bits are scrambled in transmission order as specified in IEEE 802.3bj-2014 91.5.2.5.~~

After this step, tx_xcoded<256:0> will yield tx_scrambled<256:0> as follows:

- a) Set tx_scrambled<4:0> to the result of the bit-wise Exclusive OR of tx_xcoded<4:0> and tx_xcoded<12:8>; and
- b) Set tx_scrambled<256:5> to tx_xcoded<256:5>.

5.9.2.2 Alignment marker mapping and insertion

The alignment insertion function inserts a unique data pattern (i.e., Alignment Marker) for each link into the data stream to enable identification of which of the four links is which FEC lane. This function enables the receiver to map the physical links to logical lanes allowing for random connections of the Transmit links to the Receive links within the group of 4 links, in addition to providing a framing pattern for aligning the FEC code words.

The first 514b of every 4096th FEC code word carries Alignment Marker information.

The alignment marker bit sequence is identical to the first two re-mapped AM TC blocks specified in Clause 82.2.7 and Clause 91.5.2.6 when replacing the BIP3 field for Lane 1, Lane 2, and Lane 3 in all four instances of the AM0 blocks with the value 0xCA, the BIP3 for AM4 with 0x9D, the BIP3 for AM5 with 0xD7, the BIP3 for AM6 with 0x6F, and the BIP3 for AM7 with 0xA1. For Lane0, the BIP3 field of AM0 carries Link degrade information associated with the link degrade function (see 5.5.10.2). Additionally the first bit of AM8 and AM9 that are part of the sequence is changed from 0->1 to maintain DC balance.

Table 7 shows the data stream that will appear on each of the 4 lanes after the RS symbol distribution of the AM pattern is done. The 'd' is the first 6b of data from TC block that follows the AM pattern. The underlined values are the replaced BIP3 and BIP7 fields in the AM blocks.

Table 7 - 428256GFC FEC Alignment Marker

AM bits	Lane3	Lane2	Lane1	Lane0
[39:0]	0011000001	0011000001	0011000001	0011000001
[79:40]	0001011010	0001011010	0001011010	0001011010
[119:80]	<u>0010100010</u>	<u>0010100010</u>	<u>0010100010</u>	<u>10000</u> RD,00101000 10
[159:120]	00111110 <u>11</u>	00111110 <u>11</u>	00111110 <u>11</u>	00111110 11 <u>10</u>
[199:160]	1010010111	1010010111	1010010111	1010010111
[239:200]	<u>0101110111</u>	<u>0101110111</u>	<u>0101110111</u>	0101111 <u>~RD,110111</u>
[279:240]	111011 <u>0011</u>	011010 <u>0011</u>	011101 <u>0011</u>	110101 00110 <u>101</u>
[319:280]	0100010101	0100101010	0001010011	0000011111
[359:320]	<u>0101100110</u>	<u>1100100110</u>	<u>1111000010</u>	<u>0100001001</u>
[399:360]	0100 <u>101000</u>	0101 <u>011011</u>	0010 <u>110101</u>	1010 <u>100111</u>
[439:400]	1110101000	1101010110	1010110010	1110000000
[479:440]	1001100110	1101100110	0011110111	1111011011
[513:480]	<u>dddddd1110</u>	<u>0110010000</u>	<u>0100101000</u>	<u>0101100010</u>

5.9.2.3 Reed-Solomon encoder

Reed-Solomon encoding is done as specified in 5.5.4.

5.9.2.4 Symbol distribution

Once the data has been encoded, it is distributed to 4 lanes, in groups of 10 bit symbols.

Symbol distribution is done as specified in IEEE 802.3bj-2014 91.5.2.8.

5.9.2.5 Gray mapping

Gray mapping is done as specified in 5.5.5.

5.9.2.6 Precoding

Precoding is done as specified in 5.5.6.

5.9.2.7 Transmit bit ordering

Transmit bit ordering is as shown in figure 36.

SH_n = Synchronization Header “n” according to figure 10
 STWB_n = Scrambled Transmission Word Body “n” according to figure 10; n=0 (i.e., earliest word) to 3 (i.e., latest word)

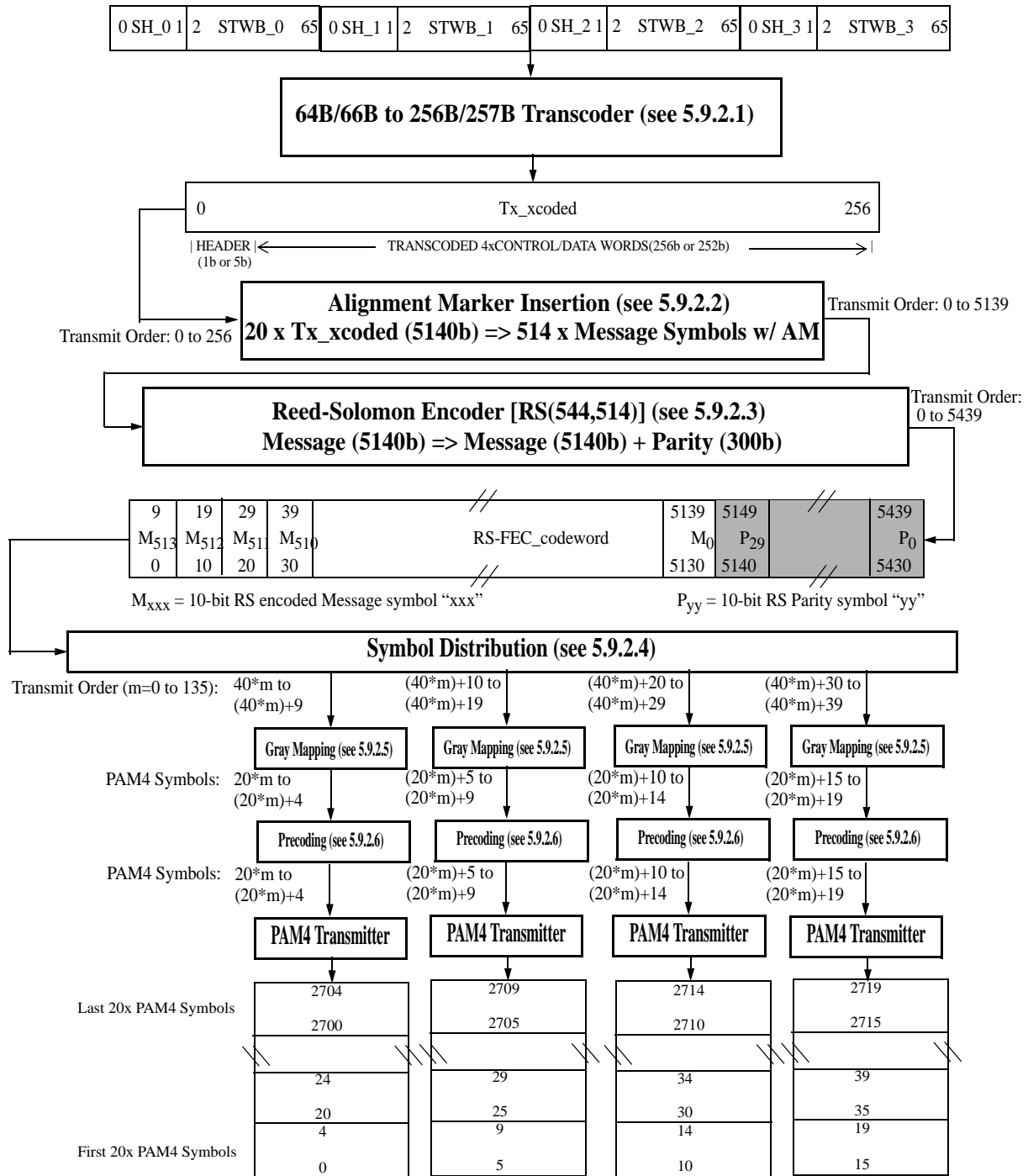


Figure 36 - 25GFC 256B/257B transmit bit ordering

5.9.2.8 **Inverse precoding**

[Inverse precoding is done as specified in 5.5.7.](#)

5.9.2.9 **Inverse Gray mapping**

[Inverse Gray mapping is done as specified in 5.5.8.](#)

5.9.2.10 **Alignment lock and deskew**

The receive function creates 4 bit streams after concatenating the bits received on each lane. It then obtains LOCK to the alignment markers on each lane as specified by the FEC synchronization state diagram in IEEE802.3bj-2014 91.5.3.1.

After alignment marker lock is achieved on all four lanes, all inter lane skew is removed as specified by the FEC alignment state diagram in IEEE802.3bj-2014 91.5.3.1. The FEC receive function will support a maximum skew of 180ns between lanes and a maximum skew variation of 4ns.

5.9.2.11 **Lane reorder**

FEC lanes may be received on different lanes of the service interface from which they were originally transmitted.

The FEC receive function shall order the FEC lanes according to the FEC lane number per IEEE802.3bj-2014-91.5.3.2. The FEC lane number is defined by the alignment marker that is mapped to each FEC lane.

After all FEC lanes are aligned, deskewed, and reordered, the FEC lanes are multiplexed together in the proper order to reconstruct the original stream of FEC code words.

5.9.2.12 **Reed-Solomon decoder**

Decoding is done as specified in 5.5.10.1. [In addition, link degrade signaling is done as specified in 5.5.10.2.](#)

5.9.2.13 **Alignment marker removal**

The first 514 bits in every 4096 code words are the mapped alignment marker bits. These are removed before sending the data to the transcode block.

5.9.2.14 **256B/257B to 64B/66B transcoder**

The first five bits of the of the received block $rx_scrambled\langle 256:0\rangle$, in reception order, are descrambled. $Rx_scrambled\langle 256:0\rangle$ will yield $rx_coded\langle 256:0\rangle$ as follows:

- a) Set $rx_coded\langle 4:0\rangle$ to the result of the bit wise Exclusive-OR of $rx_scrambled\langle 4:0\rangle$ and $rx_scrambled\langle 12:8\rangle$; and
- b) Set $rx_coded\langle 256:5\rangle$ to $rx_scrambled\langle 256:5\rangle$.

Next, a group of four 66bit transmission words are constructed from each received 257 bit transmission word as specified in 5.4.7.

5.9.2.15 Receive bit ordering

Receive bit ordering is as specified in figure 37.

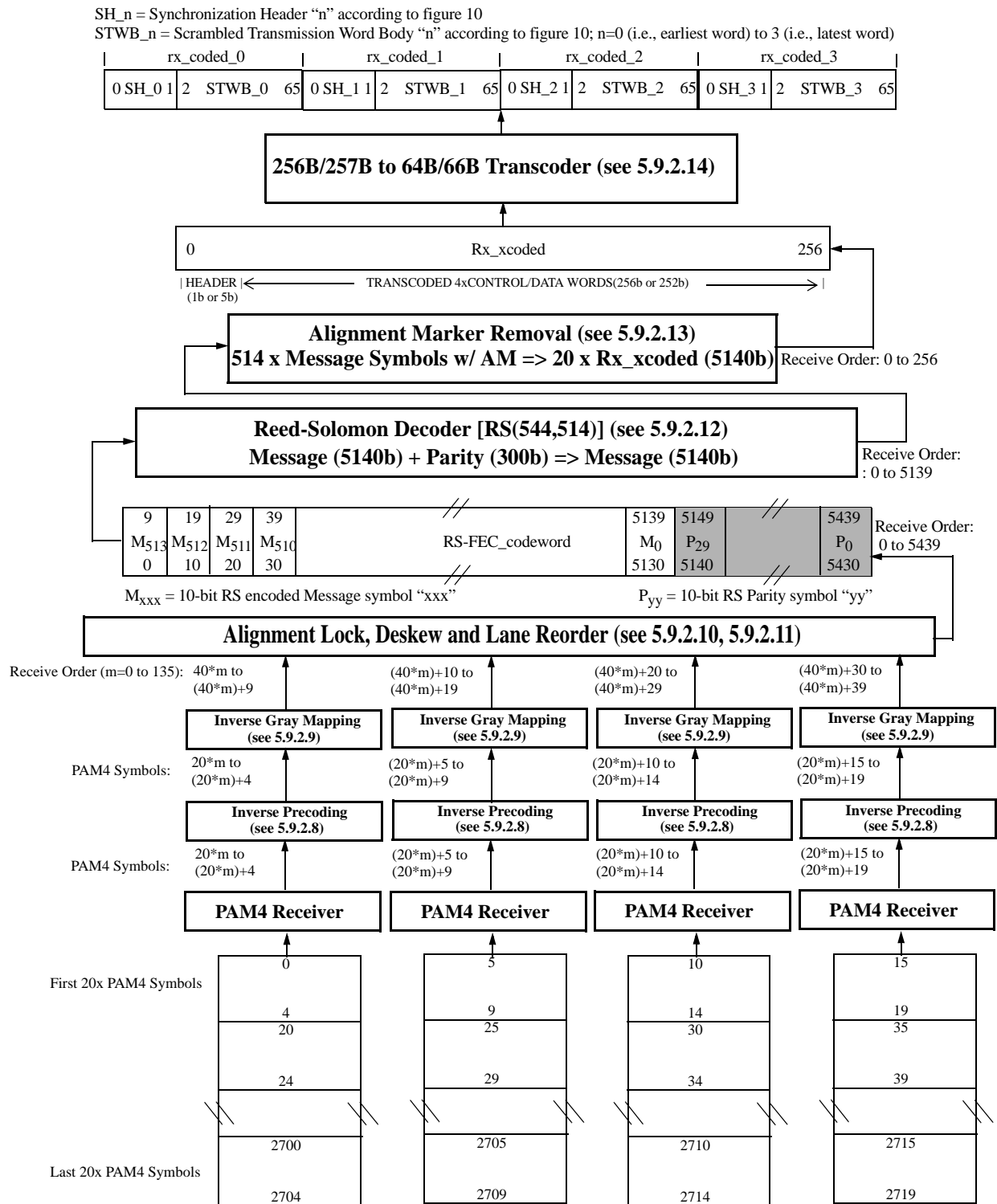


Figure 37 - 256GFC 256B/257B receive bit ordering

6 FC-1 Transmission Word Synchronization

6.1 Scope

Transmission Word Synchronization is a function of the FC-1 level.

6.2 Introduction

In the Fibre Channel architecture, the FC-0 level is responsible for bit transmission and reception (see FC-PI-x). The FC-1 level is responsible for providing a stream of bits for the FC-0 level to transmit. No state information is needed to accomplish this other than that necessary for 64B/66B scrambling and 8B/10B running disparity. The FC-1 level is also responsible for deriving Transmission Word Synchronization and Transmission Words from the received bit stream.

Whenever a signal (see FC-PI-x) is detected on a fibre, the receiver attached to that fibre shall attempt to achieve synchronization on both bit and Transmission Word boundaries of the received encoded bit stream. Bit Synchronization is defined in FC-PI-x. Transmission Word Synchronization is defined in this clause. Synchronization failures on either bit or Transmission Word boundaries are not separately identifiable; both cause Loss-of-Synchronization errors.

An FC_Port receiver has two mutually exclusive receiver Transmission Word Synchronization states, Word Synchronization Acquired and Loss of Synchronization. In the Word Synchronization Acquired state, the FC-1 level shall decode the received signal and pass information to the FC-2P level. In the Loss of Synchronization state, the FC-1 level shall not pass information to the FC-2P level.

A receiver may provide an indication of a Loss-of-Signal condition (see FC-PI-x).

6.3 8B/10B Transmission Word Synchronization

6.3.1 State Diagram Overview

The Receiver State Diagram for 8B/10B Transmission Word Synchronization is shown in figure 39.

The Receiver states are as follows:

- a) Loss of Synchronization state;
- b) No Invalid Transmission Word Detected state;
- c) First Invalid Transmission Word Detected state;
- d) Second Invalid Transmission Word Detected state;
- e) Third Invalid Transmission Word Detected state; and
- f) Reset state.

Being in one of the Word Synchronization Acquired states refers to being in any of:

- a) No Invalid Transmission Word Detected state;
- b) First Invalid Transmission Word Detected state;
- c) Second Invalid Transmission Word Detected state; or
- d) Third Invalid Transmission Word Detected state.

The receiver state transitions are defined as follows:

- a) Transition 1: Power-on;
- b) Transition 2: Acquisition of Word Synchronization (see 6.3.3.2.2);
- c) Transition 3: An invalid Transmission Word is detected (see 6.3.4.2);
- d) Transition 4: A detection of a Loss-of-Signal condition (see 6.2);
- e) Transition 5: Two consecutive Transmission Words that are not Invalid Transmission Words are detected (see 6.3.4.2);
- f) Transition 6: Reset condition imposed on the receiver (see 6.3.5.4); and
- g) Transition 7: Exiting of receiver reset condition (see 6.3.5.4).

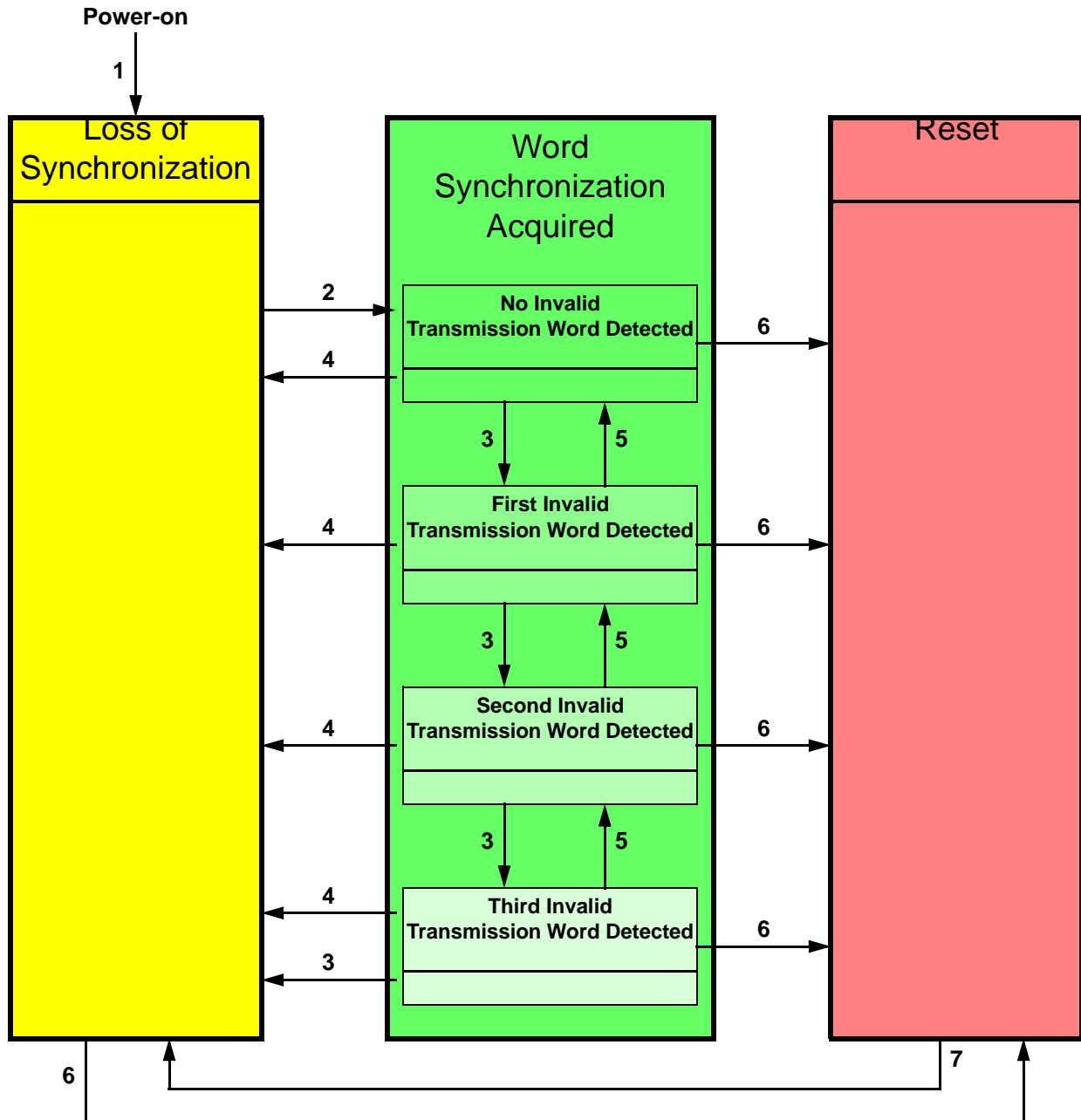


Figure 39 - Receiver state diagram

6.3.2 Operational and not operational conditions

When the receiver is operational, it shall be in either the Loss of Synchronization state or in one of the Word Synchronization Acquired states.

When the receiver is Not operational, it shall be in the Reset state.

6.3.3 Transmission Word Synchronization Procedure

The Transmission Word Synchronization procedure consists of first achieving Bit Synchronization (see 6.3.3.1), followed by achieving Transmission Word Synchronization (see 6.3.3.2).

6.3.3.1 Bit Synchronization

An operational receiver that is in the Loss of Synchronization state shall first acquire Bit Synchronization before attempting to acquire Transmission Word Synchronization. Bit Synchronization is defined in FC-PI-x. After achieving Bit Synchronization, the receiver shall remain in the Loss of Synchronization state until it achieves Transmission Word Synchronization.

6.3.3.2 Transmission Word Synchronization detection

6.3.3.2.1 Introduction

The comma contained within the K28.5 special character is a singular bit pattern that in the absence of transmission errors shall not appear in any other location of a Transmission Character and shall not be generated across the boundaries of any two adjacent Transmission Characters. This bit pattern is sufficient to identify the Transmission Word alignment of the received bit stream. Some implementations (e.g., those that choose to implement the Transmission Word alignment function in Continuous-mode alignment) may choose to align on the full K28.5 Ordered Set to decrease the likelihood of false alignment when bit errors are present in the received bit stream.

Placement of a K28.5 Transmission Character at the left-most position of a received Transmission Word ensures proper alignment of that Transmission Word and of subsequently received Transmission Words. Ordered Set detection shall include both detection of the individual Transmission Characters that make up an Ordered Set and proper alignment of those characters (i.e., the Special Character used to designate an Ordered Set shall be aligned in the leading (left-most) character position of the received Transmission Word).

6.3.3.2.2 Achieving Transmission Word Synchronization

A receiver that is in the Loss of Synchronization state and has acquired Bit Synchronization shall attempt to acquire Transmission Word Synchronization. Transmission Word Synchronization is acquired by the detection of three Ordered Sets containing commas in their left-most bit positions without an intervening invalid Transmission Word, as specified in 6.3.4.2. The third detected Ordered Set shall change the state from the Loss of Synchronization state to the No Invalid Transmission Word Detected state using transition 2. The third detected Ordered Set shall be considered valid information and shall be decoded and provided by the receiver to its FC_Port. A receiver in any of the Word Synchronization Acquired states shall provide information that has been received from its attached fibre and decoded.

The method used by the receiver to implement the Transmission Word alignment function and to detect Ordered Sets is not defined by this standard.

6.3.3.2.3 8B/10B Transmission Word Synchronization for speed negotiation

If the link speed negotiation algorithm (see 8.6) is performed using 8B/10B, then the pass sync_test count shall be 1 000.

6.3.3.2.4 Transmission Word alignment methods

6.3.3.2.4.1 Continuous-mode alignment

Continuous-mode alignment allows the receiver to reestablish Transmission Word alignment at any point in the incoming bit stream while the receiver is operational. Such realignment is likely (but not guaranteed) to result in code violations and subsequent Loss-of-Synchronization. Under certain conditions, it may be possible to realign an incoming bit stream without Loss-of-Synchronization. If such a realignment occurs within a received frame, detection of the resulting error condition is dependent upon higher-level function (e.g., invalid CRC, missing EOF Delimiter).

6.3.3.2.4.2 Explicit-mode alignment

Explicit-mode alignment allows the receiver to reestablish Transmission Word alignment under controlled circumstances (e.g., while in the Loss of Synchronization State). Once synchronization has been acquired, the Transmission Word alignment function of the receiver is disabled.

6.3.4 Loss of Transmission Word Synchronization

6.3.4.1 Introduction

Loss of Transmission Word Synchronization shall occur in the following conditions:

- a) a Loss-of-Signal is detected when in any of the Word Synchronization Acquired states; or
- b) an invalid Transmission Word is detected in the Third Invalid Transmission Word Detected state.

6.3.4.2 Detection of an invalid Transmission Word

In each of the Word Synchronization Acquired states each received Transmission Word is tested to determine the validity of the Transmission Word.

An invalid Transmission Word shall be recognized by the receiver when one of the following conditions is detected:

- a) a code violation, as specified by the 8B/10B transmission code (see 5.2), is detected within a Transmission Word. This is referred to as a code violation condition;
- b) a K30.7 special character is detected in any character position of a Transmission Word. This indicates an error condition has been detected at a lower implementation level within the receiver;
- c) any valid special character is detected in the second, third, or fourth character position of a Transmission Word. This is referred to as an invalid special code alignment condition; or
- d) a defined Ordered Set (see clause 5) is received with improper beginning running disparity (e.g., a SOF delimiter is received with positive beginning running disparity, an EOF delimiter specified for positive beginning running disparity is received when beginning running disparity for that Transmission Word is negative). This is referred to as an invalid beginning running disparity condition.

6.3.5 State transitions

6.3.5.1 Default State

A receiver shall enter the Loss of Synchronization state on power-on (i.e., default).

6.3.5.2 Loss of Synchronization state

The Loss of Synchronization State shall be entered upon the following conditions:

- a) completion of the Loss-of-Synchronization procedure while in the Third Invalid Transmission Word Detected state using transition 3;
- b) detection of Loss-of-Signal while in the No Invalid Transmission Word Detected state, the First Invalid Transmission Word Detected state, the Second Invalid Transmission Word Detected state, or the Third Invalid Transmission Word Detected state using transition 4; or
- c) completion of the reset while in the Reset state using transition 7.

While in the Loss of Synchronization State, the receiver may attempt to reacquire Bit Synchronization. In some instances, this may allow the receiver to regain Transmission Word Synchronization when it otherwise would not be possible. However, initiation of bit re synchronization may also delay the synchronization process by forcing the receiver to reestablish a clock reference when such reestablishment is otherwise unnecessary (see FC-PI-x for a detailed discussion of Bit Synchronization).

When Transmission Word Synchronization is acquired the receiver shall enter the No Invalid Transmission Word Detected state using transition 2. Imposing a reset condition upon the receiver shall cause any state to transition to the Reset state using transition 6.

6.3.5.3 Word Synchronization Acquired states

6.3.5.3.1 Loss-of-Synchronization procedure

The following four states are defined as Word Synchronization Acquired states:

- a) No Invalid Transmission Word Detected state;
- b) First Invalid Transmission Word Detected state;
- c) Second Invalid Transmission Word Detected state; or
- d) Third Invalid Transmission Word Detected state.

NOTE 10 - The rationale for the Loss-of-Synchronization procedure is to reduce the likelihood that a single error results in a Loss-of-Synchronization. A single two-bit error positioned to overlap two Transmission Words could result in the detection of three invalid Transmission Words; the two Transmission Words directly affected by the error and a subsequent Transmission Word that was affected by a disparity change resulting from the error. The procedure described above would maintain synchronization in such a case.

6.3.5.3.2 No Invalid Transmission Word Detected state

When the procedure is in the No Invalid Transmission Word Detected state, checking for an invalid Transmission Word shall be performed. Any invalid Transmission Word shall cause the No Invalid Transmission Word Detected state to transition to the First Invalid Transmission Word Detected state (transition 3). A Loss-of-Signal condition shall cause the No Invalid Transmission Word Detected state to transition to the Loss of Synchronization state (transition 4). A reset condition imposed upon the receiver shall cause the No Invalid Transmission Word Detected state to transition to the Reset state (transition 6).

6.3.5.3.3 First Invalid Transmission Word Detected state

When the procedure is in the First Invalid Transmission Word Detected state, checking for an invalid Transmission Word shall be performed. Any invalid Transmission Word shall cause the First Invalid Transmission Word Detected state to transition to the Second Invalid Transmission Word Detected state (transition 3). If two consecutive Transmission Words that are not Invalid Transmission Words are received, the First Invalid Transmission Word Detected state shall transition to the No Invalid Transmission Word Detected state (transition 5). A Loss-of-Signal condition shall cause the First Invalid Transmission Word Detected state to transition to the Loss of Synchronization state (transition 4). A reset condition imposed upon the receiver shall cause the First Invalid Transmission Word Detected state to transition to the Reset state (transition 6).

6.3.5.3.4 Second Invalid Transmission Word Detected state

When the procedure is in the Second Invalid Transmission Word Detected state, checking for an invalid Transmission Word shall be performed. Any invalid Transmission Word shall cause the Second Invalid Transmission Word Detected state to transition to the Third Invalid Transmission Word Detected state (transition 3). If two consecutive Transmission Words that are not Invalid Transmission Words are received, the Second Invalid Transmission Word Detected state shall transition to the First Invalid Transmission Word Detected state (transition 5). A Loss-of-Signal condition shall cause the Second Invalid Transmission Word Detected state to transition to the Loss of Synchronization state (transition 4). A reset condition imposed upon the receiver shall cause the Second Invalid Transmission Word Detected state to transition to the Reset state (transition 6).

6.3.5.3.5 Third Invalid Transmission Word Detection state

When the procedure is in the Third Invalid Transmission Word Detected state, checking for an invalid Transmission Word shall be performed. Any invalid Transmission Word shall cause the Third Invalid Transmission Word Detected state to transition to the Loss of Synchronization state (transition 3). If two consecutive Transmission Words that are not Invalid Transmission Words are received, the Third Invalid Transmission Word Detected state shall transition to the Second Invalid Transmission Word Detected state (transition 5). A Loss-of-Signal condition shall cause the Third Invalid Transmission Word Detected state to transition to the Loss of Synchronization state (transition 4). A reset condition imposed upon the receiver shall cause the Third Invalid Transmission Word Detected state to transition to the Reset state (transition 6).

6.3.5.4 Reset state

When a receiver reset condition is imposed on a receiver, either internally or externally, the receiver shall enter the Reset state (transition 6). Once the Reset state is entered, the receiver shall become not operational and shall remain in the Reset state until it is subsequently made operational by exiting the receiver reset condition.

NOTE 11 - A typical use of receiver reset is to force a receiver in the Loss of Synchronization State to attempt reacquisition of Bit Synchronization. Entry into this state does not necessarily indicate loss of Bit Synchronization.

When the receiver is operational after exiting from a receiver reset condition imposed upon it, either externally or internally, the receiver shall enter the Loss of Synchronization state.

NOTE 12 - The conditions required for a receiver in the Reset state to exit that state are not defined by this standard. Such conditions may be based on explicit indications. They may also be time-dependent in nature.

6.4 64B/66B Transmission Word Synchronization

6.4.1 Overview

64B/66B Transmission Word Synchronization state shall be maintained as specified by the Lock state machine and the BER monitor state machine of the Physical Coding Sublayer (PCS) for 64B/66B, type 10GBASE-R (see subclause 49.2.13 of IEEE 802.3-2012):

- a) if the `block_lock` flag of the Lock state machine is TRUE, the `hi_ber` flag of the BER monitor state machine is FALSE, and the receiver is not indicating Loss-of-Signal, the receiver Transmission Word Synchronization state shall be Word Synchronization Acquired; and
- b) if the `block_lock` flag of the Lock state machine is FALSE, the `hi_ber` flag of the BER monitor state machine is TRUE, or the receiver is indicating Loss-of-Signal, the receiver Transmission Word Synchronization state shall be Loss of Synchronization.

If a receiver is decoding 64B/66B that has been further encoded with FEC (see 5.3.1 and 9.3.7.2.1), loss of FEC block synchronization (see subclause 74.10 of IEEE 802.3-2012) is indicated by the value of the `fec_signal_ok` variable of the FEC block synchronization state machine. A value of FALSE for the `fec_signal_ok` variable of the FEC block synchronization state machine shall be treated as a Loss-of-Signal indication by the receiver.

The Lock state machine relies on the property of the 64B/66B Transmission code that a bit value transition is always encoded between the two least significant bits of a Transmission Word, and because of scrambling is unlikely to occur consistently at any other 66-bit period in the encoded bit stream.

Other than loss of Bit Synchronization, signal conditions (e.g., code violation detection) detected between expected synchronization headers do not affect the receiver Transmission Word Synchronization state during use of the 64B/66B transmission code.

6.4.2 64B/66B Transmission Word Synchronization for speed negotiation

If the link speed negotiation algorithm (see 8.6) is performed using 64B/66B, then the pass `sync_test` count shall be 1 000.

6.4.3 Detection of an invalid 64B/66B Transmission Word

An invalid 64B/66B Transmission Word shall be recognized by the receiver:

- a) if both bits in the Synchronization Header have the same value, then the Transmission Word shall cause a code violation (i.e., Invalid Synchronization Header, see 5.3.4) to be reported;
- b) if a Transmission Word type is decoded that is restricted in table 10, then the Transmission Word shall cause a code violation (i.e., Restricted Transmission Word type, see 5.3.6) to be reported;
- c) if a control code value other than Idle or LPI (i.e., if the FC_Port supports Energy Efficient Fibre Channel), is decoded, then the Transmission Word shall cause a code violation (i.e., Restricted Control Code, see 5.3.6) to be reported;
- d) if a restricted order code value is decoded, the Special Function shall cause a code violation (i.e., Restricted Order Code, see 5.3.6) to be reported;
- e) an Idle followed by SOF Transmission Word shall cause a code violation (i.e., Idle followed by SOF error, see 5.3.6.2) to be reported if the Transmission Word received prior to receiving an Idle followed by SOF Transmission Word:
 - A) was a data Transmission Word;
 - B) was any Transmission Word containing an SOF; or

- C) caused a coding violation to be reported;
- f) an EOF followed by Idle or LPI Transmission Word shall cause a code violation (i.e., EOF followed by Idle or LPI error, see 5.3.6.3) to be reported if the Transmission Word received following receiving an EOF followed by Idle or LPI Transmission Word:
 - A) is a data Transmission Word;
 - B) is any Transmission Word containing an EOF; or
 - C) causes a coding violation to be reported;
- g) an Other Special Function/SOF Transmission Word shall cause a code violation (i.e., Other Special Function / SOF error, see 5.3.6.7) to be reported if the Transmission Word received prior to receiving an Other Special Function/SOF Transmission Word:
 - A) was a data Transmission Word;
 - B) was any Transmission Word containing an SOF; or
 - C) caused a coding violation to be reported;
- h) a SOF/data Transmission Word shall cause a code violation (i.e., SOF/data error, see 5.3.6.8) to be reported if the Transmission Word received prior to receiving an SOF/data Transmission Word:
 - A) was a data Transmission Word;
 - B) was any Transmission Word containing an SOF; or
 - C) caused a coding violation to be reported;
- i) a data/EOF Transmission Word shall cause a code violation (i.e., data/EOF error, see 5.3.6.9) if the Transmission Word received following receiving a data/EOF Transmission Word:
 - A) is a data Transmission Word;
 - B) is any transmission word containing an EOF; or
 - C) causes a coding violation to be reported.,
- j) if an Error Transmission Word is received, then a code violation (i.e., receiver detected error, see 5.3.6.10) shall be reported;
- k) an RX_E transition error shall be reported if a transition from the:
 - A) RX_INIT state to the RX_E state;
 - B) RX_C state to the RX_E state;, or
 - C) RX_D state to the RX_E state occurs (see IEEE Std 802.3-2012, figure 49-17).

6.5 Transmitter Training Signal Transmission Word Synchronization

6.5.1 Introduction

Transmitter Training Signal Transmission Word Synchronization state shall be maintained as specified by the Frame lock state machine of the Physical Medium Dependent Sublayer and Baseband Medium, Type 10GBASE-KR (see subclause 72.6.10.4.1 of IEEE 802.3-2012), except that the condition for entry to the state machine is that the FC_Port initiates use of the Transmitter Training Signal. The training variable of the 10GBASE-KR Frame lock state machine shall be ignored:

- a) if the frame_lock variable of the 10GBASE-KR Frame lock state machine is set to one and the receiver is not indicating Loss-of-Signal, the receiver Transmission Word Synchronization state shall be Word Synchronization Acquired; and
- b) if the frame_lock variable of the 10GBASE-KR Frame lock state machine is set to zero or the receiver is indicating Loss-of-Signal, the receiver Transmission Word Synchronization state shall be Loss of Synchronization.

NOTE 13 - The offset between Frame Marker fields is different for Transmitter Training frames used for 64GFC Transmitter Training (see 5.7) and the Transmitter Training Signal for LSN and 32GFC/16GFC Transmitter Training (see 5.5). The frame lock state machine needs to take this into account based on which type of Transmitter Training signal it is trying to lock to.

Transmitter Training Signal Transmission Word Synchronization relies on the properties of the Transmitter Training Signal that each Transmission Word begins with a 32 TUI frame marker pattern that appears nowhere else in any Transmission Word.

Other than an indication of Loss-of-Signal, the signal between expected frame markers shall not affect Transmitter Training Signal Transmission Word Synchronization state.

In the case of a DME coding violation, the Transmitter Training packet shall be ignored. See IEEE 802.3-2012 for definition of DME code violation.

6.5.2 Transmitter Training Transmission Word Synchronization for speed negotiation

If the link speed negotiation algorithm (see 8.6) is performed using Transmitter Training Signal, then the pass sync_test count shall be 300.

6.6 256B/257B Transmission Word Synchronization for 32GFC RS-FEC

6.6.1 Overview

Transmission Word Synchronization is performed on the stream of 64B/66B Transmission Words as follows:

- 1) given a candidate starting bit position for an RS-FEC code word, descramble the Transmission Word and compute the syndrome and if the syndrome is:
 - a) not zero, then choose the next candidate starting bit position and return to step 1; and
 - b) zero, then set good transmission words count to 1 and go to step 2;
- 2) descramble the next Transmission Word received, starting at the candidate bit position, and attempt to correct it and if the Transmission Word:
 - a) contains errors but is not corrected, then choose the next candidate starting bit position and return to step 1; and
 - b) is error-free or corrected, then:
 - i) increment the good transmission words count;
 - ii) If the good transmission words count is less than 2, then go step 2; and
 - iii) If the good transmission words count is not less than 2, then set codeword_sync to true, set bad transmission words count to 0, and go to step 3;

and
- 3) while codeword_sync is true, descramble and attempt to correct next received code word, and if the Transmission Word:
 - a) is error-free or corrected, then set bad transmission words count to 0 and return to step 3;
 - b) contains errors but is not corrected, then:
 - i) increment the bad transmission words count;
 - ii) if the bad transmission words count is less than 3, then return to step 3;
 - iii) if the bad transmission words count is not less than 3, then set codeword_sync to false and return to step 1.

6.6.2 RS-FEC rapid code Word Synchronization process

The RS-FEC rapid code Word Synchronization process identifies the starting bit position of an RS-FEC code word and provides it to the Transmission Word Synchronization process to greatly reduce the time to achieve lock. It performs this function by searching for either of two known patterns that are sent by the transmitter when `scr_bypass` is set to TRUE (i.e., one pattern includes Idle control codes while the other includes LPI control codes).

Upon a transition from `rx_mode=QUIET` to `rx_mode=DATA`, the receiver suspends the Transmission Word Synchronization process and starts a timer whose duration is `Trs`. During this time, the RS-FEC rapid code Word Synchronization process attempts to identify either of the known patterns in the received bits.

When a known pattern is found, the corresponding starting bit position for the RS-FEC Codeword is passed to the Transmission word synchronization process which is then released and resumes normal operation.

If the timer expires before the known pattern is found, then the Transmission Word Synchronization resumes normal operation.

6.7 Transmission Word Synchronization for 64GFC RS-FEC

Transmission Word Synchronization is achieved by obtaining Lock to an Alignment marker as specified in IEEE 802.3cd D2.2 134.5.3.1 except that:

- 1) for 64GFC there is only one lane for transmit and receive;
- 2) there is a single bit stream so forming two bit streams by concatenating bits is not required; and
- 3) Alignment Marker lock is achieved on the single FEC lane. Since there is a single FEC lane, no inter lane skew removal is required.

8 Link speed negotiation

8.1 Scope

Link speed negotiation is a function of the FC-2P sublevel.

8.2 Speed negotiation overview

The optional speed negotiation method may be used to enable ports that are capable of multiple data transfer rates to establish in-band communications on a link (all port types). The term "speed" as used in this clause refers to the bit transfer rate. This method finds the highest speed common to the ports and to the infrastructure connecting the ports. Each port may support up to a maximum of 4 speeds in the negotiation process. The exact speeds are not specified. Different ports may negotiate with different speed ranges up to a maximum of 4 speeds each and speed negotiation shall converge provided there is at least one common speed. The link quality for speed negotiation purposes is error free Transmission Word Synchronization for a minimum number of Transmission Words specified in clause 6 as the pass sync_test count for the transmission code being used.

Because the link quality requirements for speed negotiation are not as stringent as for other operations it is possible to complete speed negotiation yet have an excessive error rate in other operations. Determination of excessive error rate outside of speed negotiation may be as specified for transmitter training (see 9.2) or by vendor specific methods. The response to a determination of excessive error rate in transmitter training is to re-enter speed negotiation, having eliminated the faulty speed from negotiation. The response to a vendor specific determination of excessive error rate may also be to re-enter speed negotiation, having eliminated the faulty speed from negotiation. A speed, having been eliminated, is restored to subsequent speed negotiation upon vendor specific determination that the reliability of the link at that speed may have improved (e.g., detection of physical disconnect and reconnect of the link, or an administrative action out of scope of this standard).

Transceivers may be able to transmit and detect error free bit streams even though they and other link elements were not designed or specified for operation at the speed being used. This condition may allow links to achieve Transmission Word Synchronization and satisfactory error rates but with degraded margin. It is up to the implementer to ensure that the elements of the physical plant are designed to comply with the requirements specified for operation at the set speed.

Once a particular speed has been established, speed negotiation is not attempted again unless a Signal Failure is detected. Speed negotiation may disrupt communication in excess of a second. An FC_Port capable of the speed negotiation procedure shall initiate Speed negotiation upon power on or Signal Failure. For this purpose, Signal Failure shall be presumed to pertain only in the following circumstances:

- a) the port receiver circuit has indicated Loss of Signal;
- b) the port receiver has remained in "Loss of Synchronization" state for a time in excess of R_T_TOV;
or
- c) the port transceiver has been reset by means other than power on.

An FC_Port should not initiate speed negotiation for other reasons.

8.3 Link physical architecture and requirements

The physical architecture of the link is assumed to be as shown in figure 41.

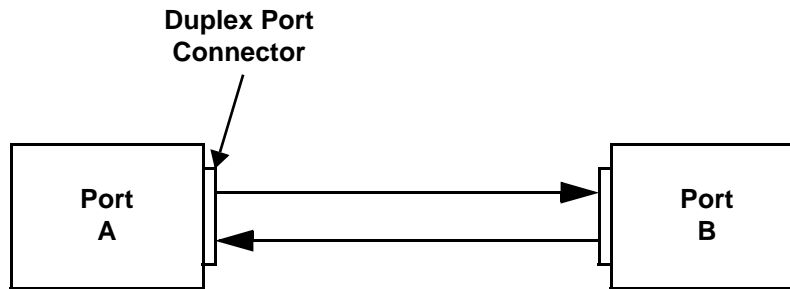


Figure 41 - Physical architecture of the speed negotiating link

There are several points derived from this physical architecture that bear on the speed negotiation algorithm:

- a) only point-to-point links are supported;
- b) loop configurations that negotiate speeds shall present a single port to the other negotiating port for speed negotiation purposes;
- c) the speed negotiation algorithm is specified for only one port at a time (i.e., when port "A" is involved, the term transmitter applies only to the transmitter in port "A" and the term receiver applies only to the receiver in port "A"). The algorithm may be executing on both ports at the same time;
- d) no requirements are explicitly placed by the algorithm on the means for controlling the transceiver speed capabilities. However:
 - A) ports implementing this algorithm shall not attain Transmission Word Synchronization unless the incoming signal is within $\pm 10\%$ of the receive rate set by the port implementing the algorithm;
 - B) the transmitter shall have a Transmitter Stabilization Time for each speed it negotiates (see 8.6.7);
 - C) the receiver shall have a Receiver Stabilization Time for each speed it negotiates (see 8.6.7); and
 - D) if the sum of the Receiver Stabilization Time plus one fifth of the Transmitter Stabilization Time exceeds 28 ms for a speed (see 8.6.7), speed negotiation shall not be conducted for that speed;
- e) a stable physical environment (fully mated connectors, no power cycles, no cable flexing, no transient noise sources, etc.) is expected during speed negotiation. Otherwise, speed negotiation may settle to a sub-optimum speed. The algorithm is capable of handling the normal connection start up transients caused by the connector insertion process (e.g., such transients include contact bounce and partial optical coupling). Sub-optimal speed may result if the connection start up transient conditions persist for more than a few milliseconds. Sub-optimal speed may also result if connectors between devices in the process of negotiating are demated and then remated within three seconds;
- f) the transmitter and receiver shall be capable of working at different speeds at the same time during speed negotiation;
- g) the algorithm supports ports capable of up to a maximum of any 4 speeds; and

- h) if an L_Port configured for speed negotiation is attached to a loop, the L_Port either:
 - A) is being attached to a port in the loop that presents a single speed and does not perform speed negotiation; or
 - B) is being attached to a port in the loop that completes the speed negotiation algorithm described here before inserting the L_Port into the loop.

8.4 Speed negotiation requirements on L_Ports

Removal of an L_Port from a loop shall not cause speed negotiation to occur on the remaining loop. This requirement applies even if the removal of the L_Port allows negotiation of a higher common speed.

As an option to negotiating each hub port per the algorithm, multiple speed hubs may be set to a single speed during speed negotiation by some out-of-band means.

8.5 Primitives

8.5.1 Overview

For FC_Ports that do not support the Transmitter Training Signal, either OLS or NOS (for ports not operating in Arbitrated Loop topology) or LIP (for ports operating in Arbitrated Loop topology) shall be the only signals transmitted during speed negotiation.

For FC_Ports that support the Transmitter Training Signal:

- a) if the FC_Port is transmitting using media and speeds that support the Transmitter Training Signal (see FC-PI-x), then the Transmitter Training Signal shall be transmitted during speed negotiation;
- b) if the Transmitter Training Signal (see 5.5.2) is transmitted during speed negotiation, then the SN field in the Training Status field shall be set to one;
- c) if the FC_Port is transmitting using media and speeds that do not support the Transmitter Training Signal, then either OLS or NOS (for ports not operating in Arbitrated Loop topology) or LIP (for ports operating in Arbitrated Loop topology) shall be transmitted using the required frame transfer transmission code (see FC-PI-x) during speed negotiation;
- d) if the FC_Port is receiving on media at speeds that support the Transmitter Training Signal, then Transmitter Training Signal Transmission Word Synchronization shall be attempted during speed negotiation;
- e) if the Transmitter Training Signal is received during speed negotiation, then the settings of fields in the Training Control field and the Training Status field shall be ignored; and
- f) if the FC_Port is receiving on media at speeds that do not support the Transmitter Training Signal, then Transmission Word Synchronization for the required frame transfer transmission code shall be attempted during speed negotiation.

If a PN_Port negotiates among multiple physical variants that use different transmission codes, the transmission code changes (e.g., from Transmitter Training Signal to 8B/10B and back) during speed negotiation, and the transmitter uses a different transmission code than the receiver at some times.

8.5.2 32GFC speed negotiation

For 32GFC the Transmitter Training Signal is used for speed negotiation. For copper links, transmitter training is performed if requested. For optical links transmitter training shall not be used. Bit 10 in the Control field of the Training Frame shall be set to zero during speed negotiation.

8.5.3 64GFC speed negotiation

For 64GFC, the Transmitter Training Signal specified in 5.6 is used for speed negotiation. 64GFC capability is indicated by setting Training Frame Control Field bits 15:14 (Extended Marker) to 10b.

For links that negotiate to 64GFC, the completion of LSN is always followed by mandatory Transmitter Training for all link types. For optical links the Training Frame Status field bit 12 (TF) is set to one which signals that the transmitter is operating with fixed coefficients. Even though the Transmitter is operating with fixed coefficients, Transmitter Training allows time for the Local Rx Equalization to complete adaptation to a PAM4 signal, enabling robust link performance. The completion of Transmitter Training is signaled by setting Training Field Status bit 15 (Receiver Ready) to one.

8.5.4 128GFC speed negotiation

For 128GFC links speed negotiation shall be performed independently on all lanes. A link that supports 128GFC operation shall set bit 10 in the Control field of the Training Frame (see table 16) on every lane to one if it desires to come up as a 128GFC link. The state machine transitions for speed negotiation on a 128GFC link are as shown in figure 42.

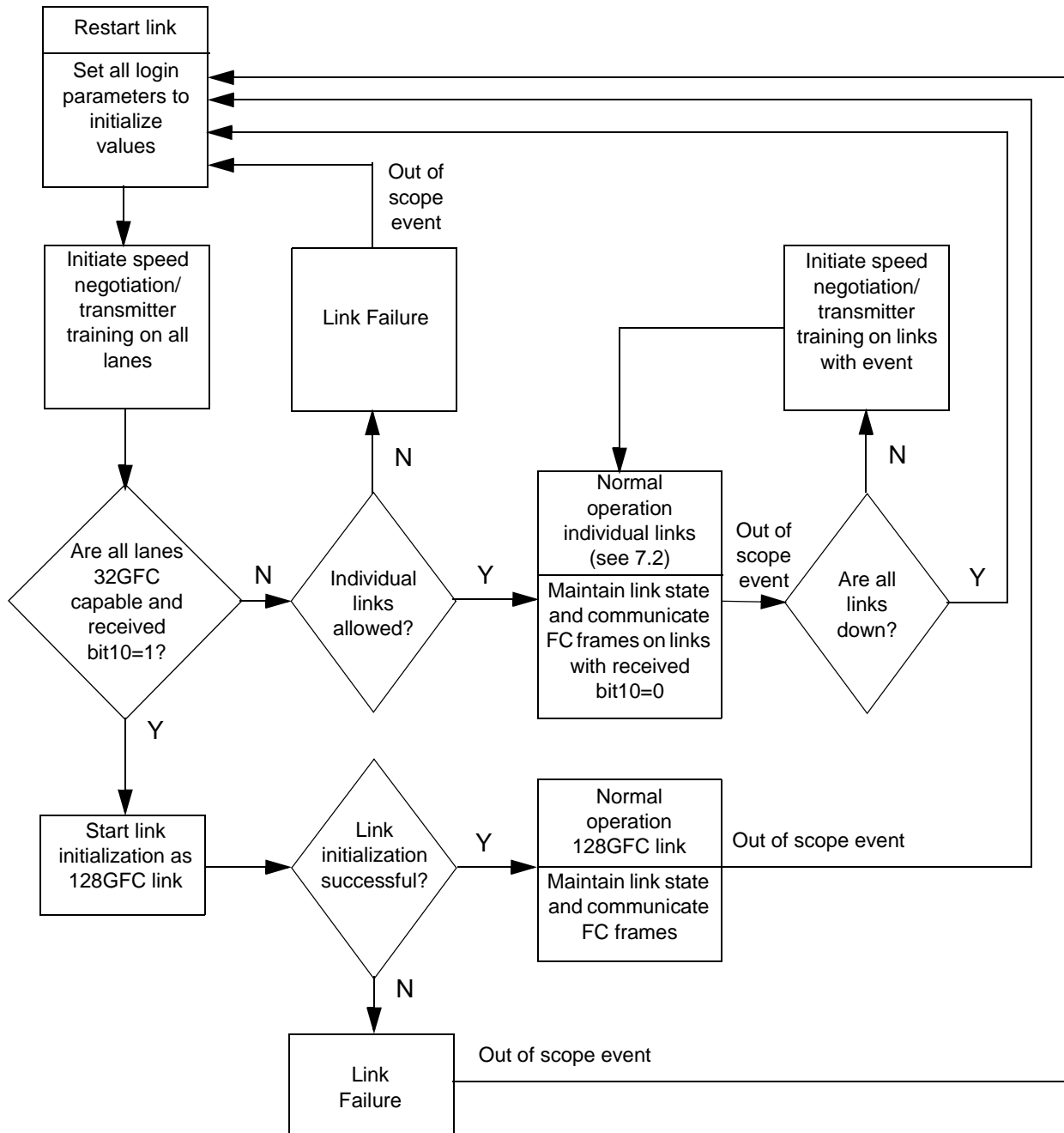


Figure 42 - 128GFC speed negotiation state machine

The 'Out of scope event' in the state diagram occurs if any of the following conditions are true on a 128GFC link:

- a) Loss-of-Signal; or
- b) Loss-of-Synchronization.

If parallel lanes are supported as indicated by receiving Training Frame Control field bit 10 set to one on all lanes and all the lanes negotiate to a speed of 32GFC, then the link may be able to operate at 128GFC. If link initialization is successful, then the link shall enter normal operation as a 128GFC link. If link initialization is unsuccessful as a 128GFC link, then the link transitions to the Link Failure State and transitions to the Restart Link state if an out of scope event occurs.

If any of the lanes do not support 32GFC or parallel lanes are not supported as indicated by receiving Training Frame Control field bit 10 set to 0 on any lane, then 128GFC is not supported and the lanes may operate as individual links at the highest negotiated speed. A link that supports 128GFC operation may support individual links of 16GFC and 32GFC. Support for individual 32GFC links is allowed only if the value of bit 10 in the Training Frame Control field received is zero during speed negotiation.

If a lane is operating as an individual link and it becomes inoperable due to an out of scope event, and all four lanes are in the link failure state, then the state machine transitions to the Restart Link state and speed negotiation is performed as a 128GFC link. If all four lanes are not in the link failure state, then speed negotiation is performed only on the failed link.

8.6 Speed negotiation algorithm

8.6.1 Algorithm overview

Figure 43 shows an overview of the speed negotiation algorithm. Dashed lines indicate optional features.

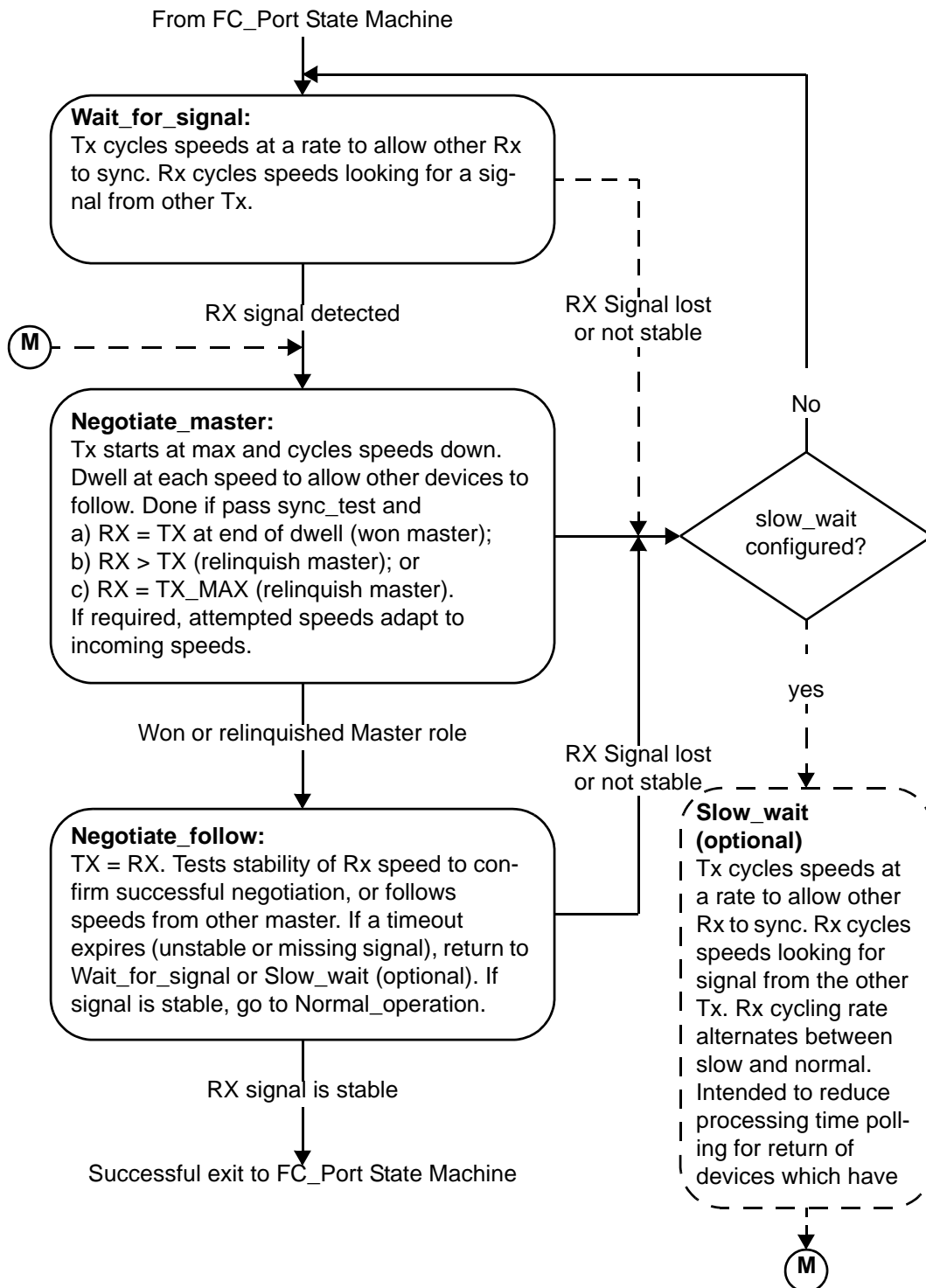


Figure 43 - Overview of the speed negotiation algorithm stages

8.6.2 Speed Negotiation stage specification conventions

8.6.2.1 Diagramming conventions

A stage is a period of time during which a PN_Port conducting Speed Negotiation performs a repeating series of activities in order to determine some major condition of the link to which it is attached (see figure 43). Each stage is specified by a stage diagram and its associated text.

For the stage diagrams of 8.6, the following concepts and diagramming symbols (see figure 44) are used:

- a) a state is a specific activity within a specific stage. Depending on the type of state, different symbols are used. For reference from text, the symbol for each state has a numeric identifier in one corner;
- b) a path specifies that a state may be followed by a successor state. The symbol for a path is a line with an arrowhead directed from the state to the successor state;
- c) an action state sets variables and conditions that control subsequent action or capture the results of prior action. The symbol for an action state is a rounded rectangle shape;
- d) a decision state has more than one successor among which it selects by the result of a test. The symbol for a decision state is a diamond shape, each path from which is labelled with the result that causes it to be selected. A “yes” result may be abbreviated as “Y”, and a “no” result may be abbreviated as “N”;
- e) a delay-and-test state is a decision state that operates for a specific time period at current settings before performing the indicated test (see figure 45). The symbol for a delay-and-test state is a boldface diamond shape, each path from which is labelled with the result that causes it to be selected; and
- f) within diagrams for required stages, paths and states that are optional to implement are indicated by symbols composed of dashed lines.

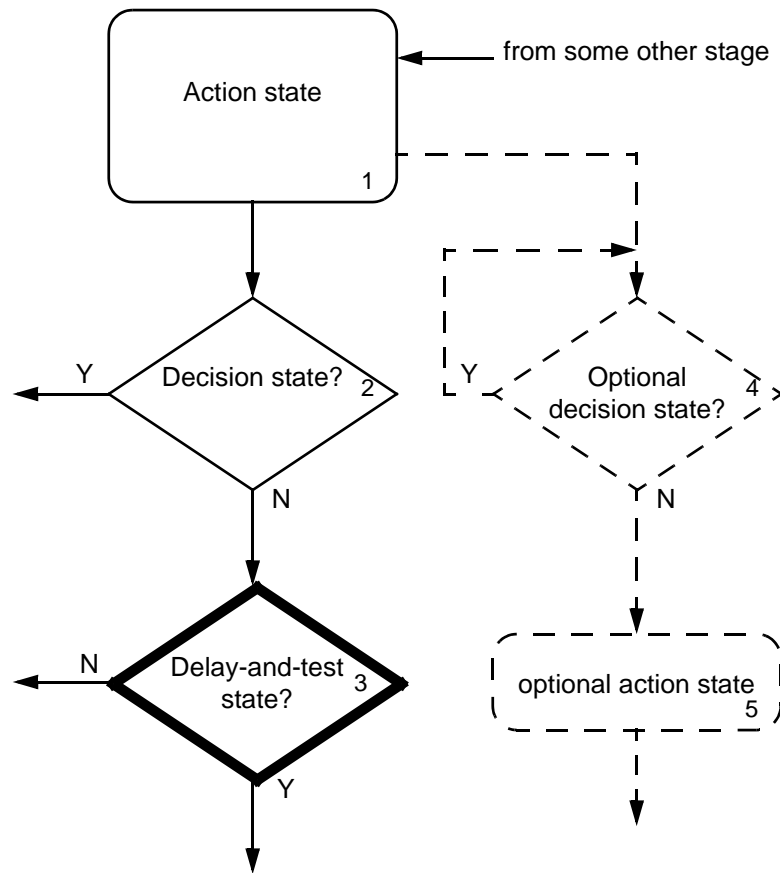
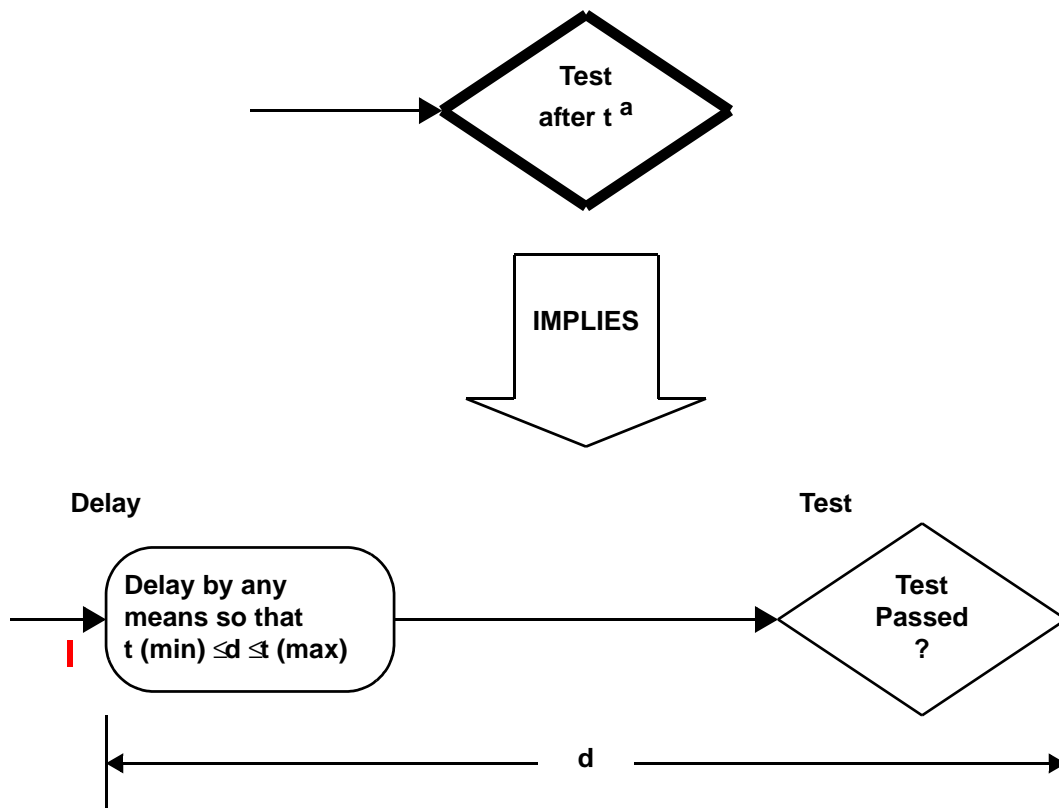


Figure 44 - Stage diagram symbols



^a t is a timing variable with minimum value t (min) and maximum value t (max)

Figure 45 - Delay / test operations

8.6.2.2 Terminology

In the stage diagrams in 8.6, the following terminology is used:

Speeds

- Tx speed list refers to the set of speeds that are currently available for negotiation by the Port. The Tx speed list may change during Negotiate_master. Transmit speed changes in the algorithm shall always be based on the Tx speed list that is currently set;
- there is no explicit Rx speed list, since the receiver is always cycled through all speeds it supports;
- recorded Rx list refers to a list of the signal speeds at which pass_sync_test has succeeded;
- RX_MAX refers to the maximum Rx speed; TX_MAX refers to the maximum speed in the current Tx speed list;
- TX refers to the present transmitter speed; RX refers to the receiver speed;
- TxNext(xxx) is the next speed less than xxx in the Tx speed list if there is a lower speed; otherwise it is the highest speed in the Tx speed list; and
- RxNext(xxx) is the next speed less than xxx among all speeds supported by the port if there is a lower speed; otherwise it is the highest speed supported by the port.

Timing

- a) pass sync_test decision blocks (states 11, 21, 27, 34, 52, 56) requires that Transmission Word Synchronization be maintained for a monitoring period that shall equal or exceed receiving the pass sync_test count (see clause 6) of consecutive Transmission Words for the transmission code being used. The period of monitoring shall not exceed 100 microseconds. Counting of code violations may be used for the monitoring period to ensure robustness, if available to the firmware. If 64B/66B transmission code is used, then code violations shall be counted for the monitoring period. If counted, then the number of errors allowed shall be zero. If the number of errors is not zero, then Transmission Word Synchronization (Pass sync_test decision blocks) is not considered to have occurred and a different speed is negotiated or the algorithm does not converge;
- b) in contrast, Sync decision block in state 31 is Transmission Word Synchronization per clause 6;
- c) in figures 46, 47, 48, and 49 a decision diamond with a bold-face outline indicates that a delay and a test are combined (see figure 45). In operations so indicated:
 - A) other activity may be implemented before the test is performed;
 - B) the test shall be completed after the minimum and before the maximum values of the delay time parameter; and
 - C) the actual delay time may vary from test to test, but the test shall fall within the specified limits;
- d) all flowchart atoms (action boxes or decision diamonds) that do not have a bold-face outline shall execute in less than 100 microseconds, and no delays shall accrue between atoms (bold-face outline or not);
- e) elapsed-time timers are compared against constants in several places:
 - A) t_{tx} , t_{neg} , and t_{sync} start where shown being (re)set to 0 in the algorithm;
 - B) t_{tx} is compared against t_{txcycl} ;
 - C) t_{neg} is compared against t_{fail} ;
 - D) t_{sync} is compared against t_{stbl} ; and
 - E) t_{nc} is compared against t_{ncycl} and may be set at several different places;
 and
- f) the R_T_TOV watchdog timer begins anytime Transmission Word Synchronization is lost during Normal_operation. Because elapsed-time counters are tested at intervals determined by a preceding delay-and-test decision (see bullet above relating to decision diamonds), the actual elapsed time determined by the elapsed-time counter test may vary from the value of the counter up to its value plus the length of the delay. In most instances, the delay may be as much as the maximum value of the range of t_{rxcycl} . This value was chosen to tolerate the response times of typical operating system kernels.

8.6.3 Stage 1 - Wait_for_signal

Figure 46 shows the flowchart for the Wait_for_signal stage.

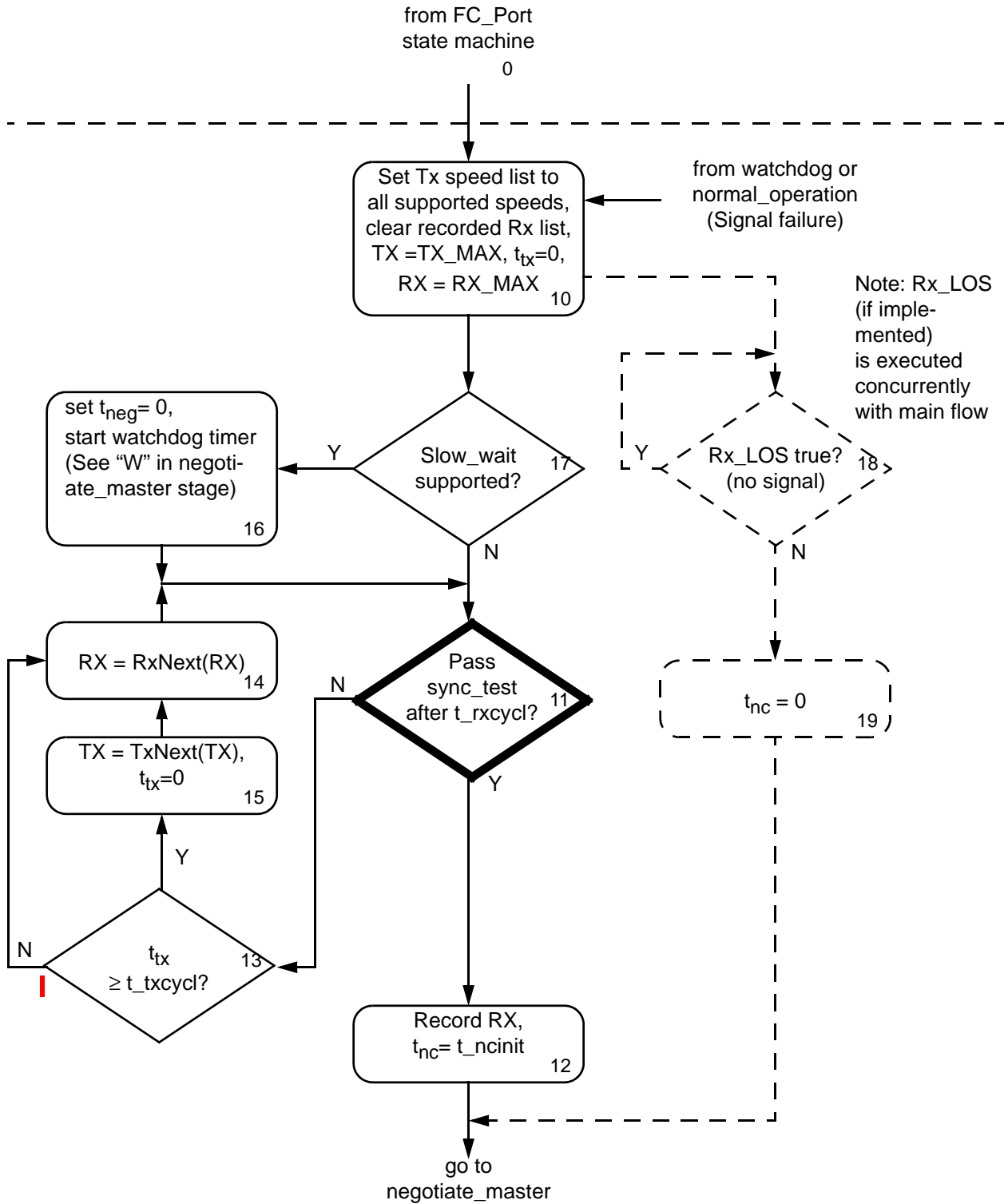


Figure 46 - Wait_for_signal flowchart

Description

- a) the device sets default parameters in state 10. States 11, 13, 14 cycle Rx speeds looking for the presence of an incoming signal from the other device that is adequate to pass the Pass sync_test. If found, RX is recorded, and the device moves onto Negotiate_master;
- b) Tx speeds are cycled slowly compared to the time spent in 1 Rx speed. This allows the receiving side of the opposite Port to cycle through at least 5 Rx speeds at each transmitted speed before the transmitted speed changes;
- c) monitoring for synchronization is performed as part of the test in state 11. Should the period of monitoring satisfy the definition of "Pass sync_test" decision blocks above, the reception of this speed shall be recorded and t_{nc} shall be set to t_{ncinit} (state 12);
- d) if the slow_wait optional stage is implemented, the watchdog timer diagrammed in figure 47 and described in 8.6.4 shall be initiated after entry to the wait_for_signal stage. If the slow_wait optional stage is not implemented, the watchdog timer shall be initiated at entry to the Negotiate_master stage but not initiated in the Wait_for_signal stage; and
- e) prior to entering state 10 from power on and ready condition, a port capable of speed negotiation shall be considered incapable of participating in normal protocol, so its transmitter shall be disabled and nothing shall be transmitted until its transmitter is enabled in the course of step 10 (see FC-PI-x).

Rx_LOS, if implemented (see dashed lines in figure 46), may be used in addition to periodically monitoring for receiver synchronization. If this option is implemented, Rx_LOS may be monitored by any means and at any time during the wait_for_signal stage after execution of block 10. If Rx_LOS becomes false, the algorithm transitions to the Negotiate_master stage without recording a received speed. In some configurations, Rx_LOS negation may occur in the absence of an active attached device. This may cause spurious entry into Negotiate_master.

8.6.4 Stage 2 - Negotiate_master and Watchdog timer

Figure 47 shows the flowchart for Negotiate_master and Watchdog timer.

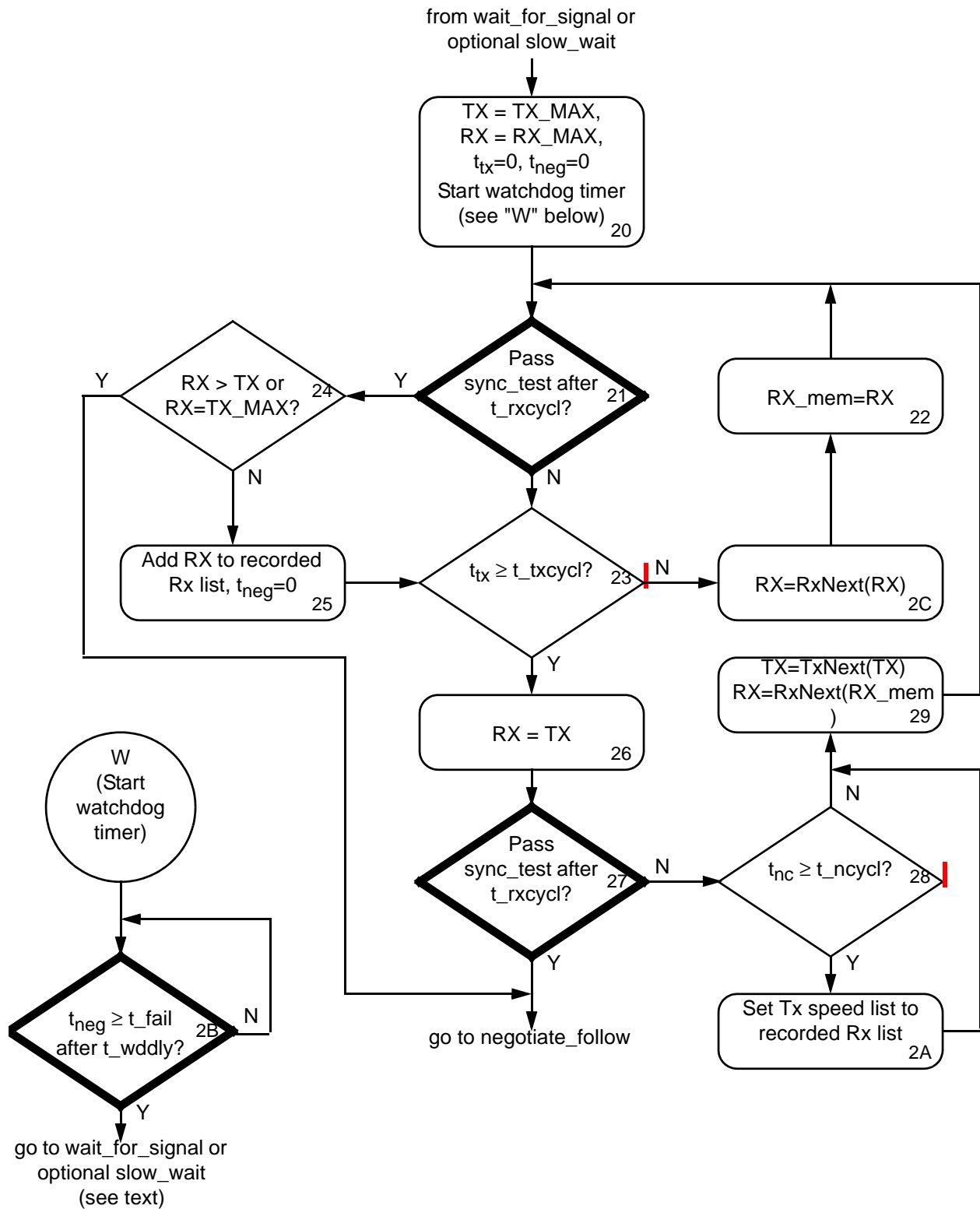


Figure 47 - Negotiate_master and watchdog timer flowchart

Description:

- a) the general operation of the algorithm is to start at the highest speed and step down until both devices agree on a speed. Lower speeds are tried only if higher speeds fail;
- b) states 23 & 26-2A control TX. A transmit speed is forced (starting at the highest speed) for sufficient time ($t_{txcycl} + t_{rxcycl}$) for the other device to pass the Pass sync_test and follow (see 8.6.5) and return TX back to the master. If NO from state 27, another (lower) Tx speed is attempted; if YES, the master role is assumed to be successful, and the algorithm moves to state 30 in Negotiate_follow;
- c) there are two conditions that may cause YES in state 27: (1) the other device is in follow mode as described above, and (2) the other device is also in master mode transmitting at the same speed. If the latter, the master role is effectively relinquished to the other master;
- d) if the port has had sufficient time to detect all possible speeds (maximum of 4 speeds) from the other port, but master role has not completed, states 28 & 2A adapt the Tx speed list to the incoming speeds recorded in the Rx list (state 25). This usually does not occur unless the cable plant only supports a limited set of speeds;
- e) states 21-25 control RX. To constantly monitor for an incoming rate that is higher than TX or equal to the maximum rate in the Tx speed list. If such a speed is found (Pass sync_test passed), the device relinquishes its master role to the other device, and transfers to the Negotiate_follow stage; and
- f) a watchdog timer, t_{neg} , keeps track of time between passed Pass sync_tests (states 11, 21, 27, and 34). If t_{neg} exceeds t_{fail} the port enters Slow_wait if the optional slow_wait stage is implemented and enabled. If the optional Slow_wait stage is not implemented the Port returns to wait_for_signal if t_{neg} exceeds t_{fail} .

Rx_LOS shall not be used during Negotiate_master stage.

8.6.5 Stage 3 - Negotiate_follow

Figure 48 shows the flowchart for the Negotiate_follow stage.

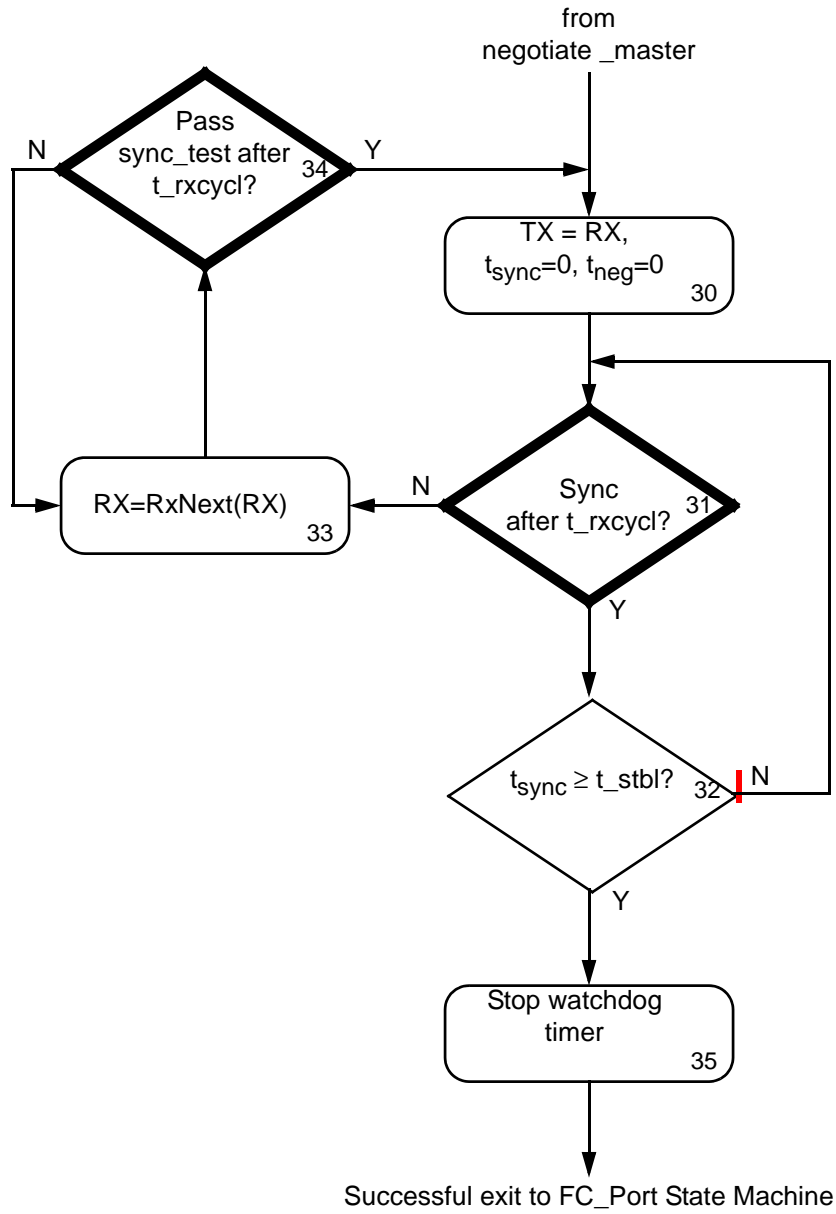


Figure 48 - Negotiate_follow flowchart

Description:

- a) a Port in the Negotiate_follow stage attempts to transmit its incoming speed;
- b) if sync is lost (e.g., due to an incoming signal speed change), the port seeks sync at another Rx speed. If obtained, TX is adjusted to follow the new RX, and the test for t_{stbl} starts over;
- c) this continues until sync is held for at least t_{stbl} in state 31 (in the case where the other master is not driving other speeds). During this time, TX and RX have been matched, allowing the other device to come to a YES decision in state 27. After t_{stbl} , the Port returns to the FC_Port state machine (see 7.2) indicating successful Speed Negotiation; and
- d) the same watchdog timer used in Negotiate_master is also used in Negotiate_follow.

Rx_LOS shall not be used during Negotiate_follow stage.

8.6.6 Optional Stage 5 - Slow_wait

Upon watchdog timer expiration, the Slow_wait stage may be entered as an alternative to returning to the Wait_for_signal stage. Its implementation is optional, and if implemented, its use may be a configuration option. Use of this optional stage reduces by approximately 80% the processing time required to monitor a Port that does not receive a valid signal at any supported speed (e.g., not cabled). However, the response to a signal being presented may be delayed by up to t_{sleep} (see table 22). Figure 49 shows the flowchart for the optional slow_wait stage.

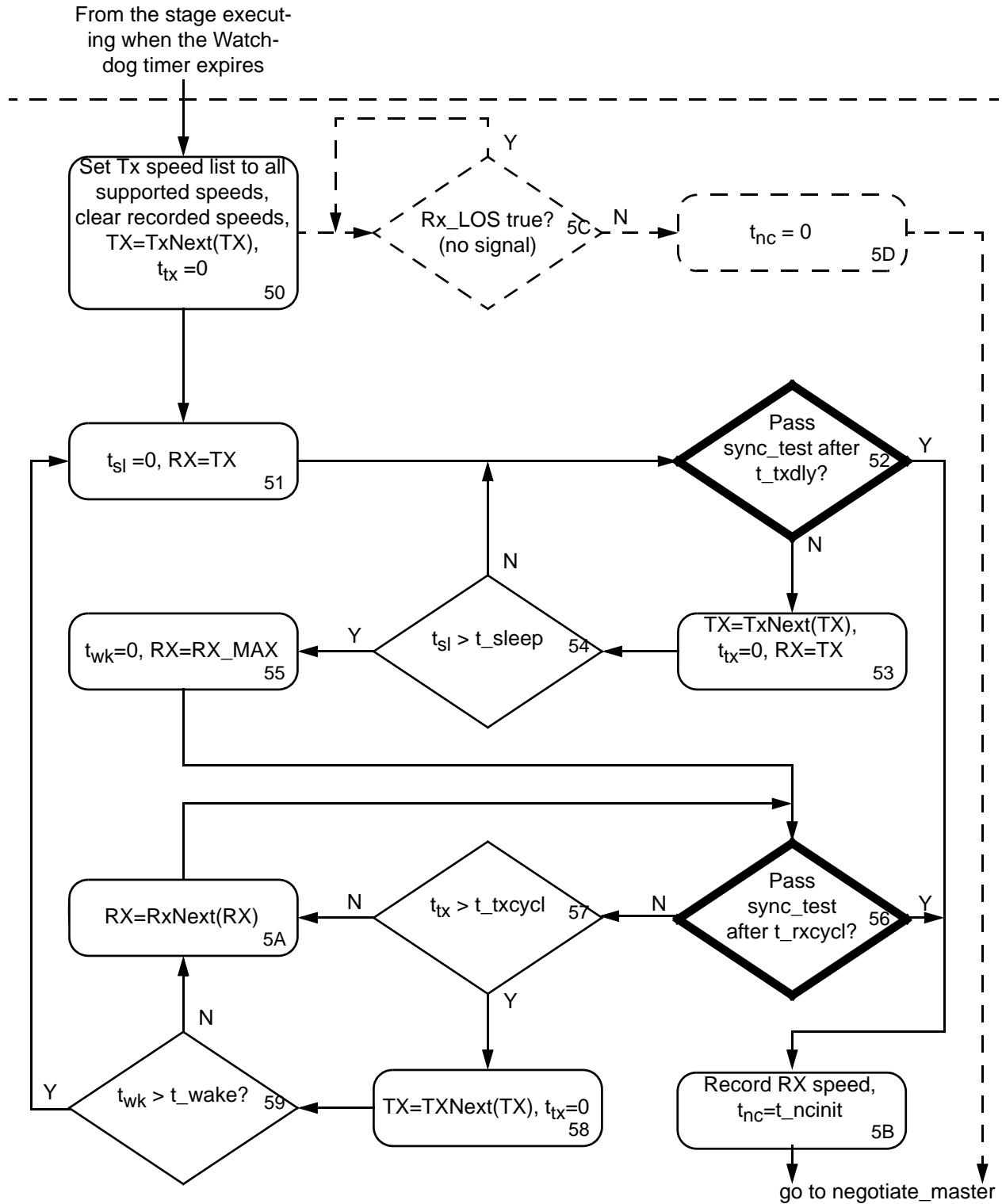


Figure 49 - Slow_wait flowchart

Description:

- a) entry into the Slow_wait stage occurs when the watchdog timer expires regardless of the stage executing when the expiration occurs;
- b) the device sets default parameters in state 50. Transmit speed cycling begins here and is uninterrupted throughout this stage, independent of cycling between slow and normal receiver speed changes;
- c) state 51 initializes the sleep timer that defines the low processing load portion of the algorithm (states 52, 53, and 54);
- d) states 52, 53, and 54 cycle both transmitter and receiver speeds at the normal transmitter speed cycling rate, checking for synchronization with any incoming signal from the upstream device before each speed change. Limiting the transmit time at each speed allows a downstream device to detect sync but not exit prematurely from Negotiate_follow. The synchronization test enables prompt synchronization to a fixed speed upstream device, reducing loop disruption upon attachment to a hub, or to an upstream device in Negotiate_follow stage. Processing load is reduced because the normal transmitter speed cycle is approximately one fifth the rate of the normal receiver speed cycle. This loop exits after operating for time t_{sleep} , or it transits to the Negotiate_master stage if synchronization is detected at the transmitted speed;
- e) states 55 initializes the receive speed and the wake timer for a period of normal receiver speed cycling; and
- f) states 56, 57, 58, 59, and 5A continue to cycle transmitter speeds but now cycle receiver speeds at their normal rate. This continues to present a signal for the downstream device to synchronize, while now attempting to synchronize with a negotiating upstream device. During this period, the behavior and processing load of the slow_wait stage is the same as the wait_for_signal stage. If synchronization is achieved, the receiver speed is recorded and the algorithm proceeds to the Negotiate_master stage. If the wake timer expires, the algorithm returns to the low processing load mode of operation.

Rx_LOS, if implemented (see dashed lines in figure 49), may be used in addition to periodically monitoring for receiver synchronization. If this option is implemented, Rx_LOS may be monitored by any means and at any time during the slow_wait stage. If Rx_LOS becomes false, the algorithm transitions to the Negotiate_master stage without recording a received speed. In some configurations, Rx_LOS negation may occur in the absence of an active attached device. This may cause spurious entry into Negotiate_master.

8.6.7 Timing requirements**8.6.7.1 Overview**

This section describes the timing requirements for the speed negotiation algorithm.

The following are variables implemented to execute the algorithm:

- a) t_{tx} : a timer that indicates the length of time since a transmitter has most recently been instructed to switch speeds. It is compared against t_{txcycl} to control duration of a transmitted speed;
- b) t_{neg} : a timer that indicates the length of time since the most recent:
 - A) successful Pass sync_test;
 - B) entry into Negotiate_master;
 - C) entry into Negotiate_follow; or
 - D) entry into Wait_for_signal if the optional Slow_wait stage is implemented.
- c) t_{neg} is compared against t_{fail} to timeout in case of Loss-of-Signal quality during negotiation; and

- d) t_{sync} : a timer that indicates the length of time that a receiver maintains `Word_sync` at a single speed. T_{sync} is used to determine that the remote transmitter is stable and is not actively changing speeds.

The following are parameters that define part of the criteria for decision points in the algorithm:

- a) transmitter Stabilization Time: The maximum time that it takes for a transmitter to achieve compliant transmission of the signal it uses for speed negotiation in a speed when administratively requested to change transmission to that speed. For any variant that does not specify a Transmitter Stabilization Time, including those specified in FC-PI-2, FC-PI-3, FC-PI-4, 10GFC, the Transmitter Stabilization Time shall be one millisecond. Other variants (e.g., those specified in FC-PI-5) may specify the Transmitter Stabilization Time to be greater than one millisecond. The Transmitter Stabilization time refers to the time for the transmitter in the Host ASIC. For module stabilization times, refer to the section on 'Timing requirements for module Speed Changes during LSN';
- b) receiver Stabilization Time: The maximum time that it takes for a receiver to stabilize detection of the signal it uses for speed negotiation in a speed when administratively requested to change reception to that speed, or when the signal presented to the receiver changes from any other speed to the speed at which the receiver is operating. For any variant that does not specify a Receiver Stabilization Time, including those specified in FC-PI-2, FC-PI-3, FC-PI-4, 10GFC, the Receiver Stabilization Time shall be one millisecond. Other variants (e.g., those specified in FC-PI-5) may specify the Receiver Stabilization Time to be greater than one millisecond. The receiver stabilization time refers to the time for the receiver in the host ASIC. For module receiver stabilization times, refer to the section on 'Timing requirements for module Speed Changes during LSN'. For 64GFC variants, the sum of the receiver stabilization time and $T_{modulrxcStbl}$ shall be less than or equal to t_{rxcycl} minus one millisecond;
- c) receiver Cycle Time, t_{rxcycl} : The limits for the time the receiver is set to a particular speed during portions of the algorithm where the receiver is cycling speeds;
- d) master_Transmitter Cycle Time, t_{txcycl} : The time threshold used to govern the transmission time of a particular speed in the `Wait_for_signal`, `Negotiate_master`, and `Slow_wait` stages;
- e) speed stability time, t_{stbl} : Time threshold required to ensure stability of the speed received from the other Port;
- f) watchdog timer threshold, t_{fail} : Time allowed for the algorithm to continue without passing the `Pass_sync_test` at any supported speed;
- g) low processing load sleep time, t_{sleep} : Threshold time for which the receiver may be cycled at the transmitter cycling rate in the `Slow_wait` stage. During this interval, attachment to a negotiating Port may not be detected, but attachment to a fixed speed Port is detected;
- h) periodic sync search wake time, t_{wake} : Threshold time for normal cycling of receiver speeds in the `Slow_wait` stage required to allow synchronization if the upstream Port is executing speed negotiation;
- i) speed recording time, t_{ncycl} : A threshold time that ensures that all possible speeds from another negotiating Port have been sampled by the receiver during the `Negotiate_master` stage;
- j) speed recording time initial value, t_{ncinit} : a constant describing the initial value for t_{nc} when `Pass_sync_test` has been achieved at a speed before entry to `Negotiate_master` stage;
- k) timer test delay, t_{wddly} : Denotes the limits on the delay that shall be included in each cycle of the watchdog timer test (state 2B); and
- l) `slow_wait` stage transmit cycle delay, t_{txdly} : Denotes the limits on the delay that shall be included in each cycle of the low processing overhead loop implemented by states 52-54.

Table 21 lists the range of values allowed for the specified timing parameter. ~~Table 22 lists the value of timing parameters used only in comparison or as a value that is set, t_ncinit.~~

Table 21 - Timing parameters with a range

Timing Parameter	Min (ms)	Max (ms)
t_rxcycl	$\geq 2^a$	30.2^b
t_wddly	0	32
t_txdly	154	184

^a t_rxcycl shall be no less than 2 ms and no less than the Receiver Stabilization Time plus one ms.
^b ~~t_rxcycl shall be no more than 30.2 ms minus one fifth of the Transmitter Stabilization Time.~~

Table 22 lists the value of timing parameters used only in comparison or as a value that is set, t_ncinit.

Table 22 - Constant timing parameters

Timing Parameter	Value (ms)
t_txcycl	154
t_stbl	217
t_ncycl	1 652
t_ncinit	370
t_fail	1 620
t_sleep	5 000
t_wake	900

8.6.7.2 Timing requirements for Module for speed changes during LSN

This section defines values for the timing parameters using figure 50 as content.

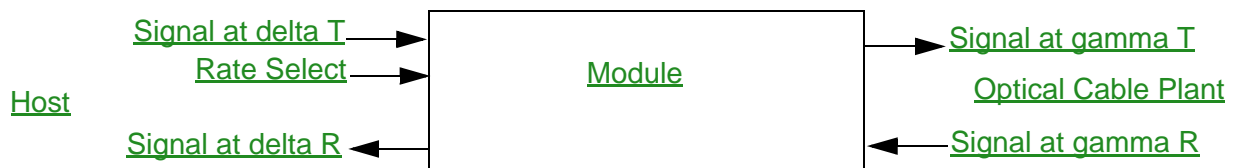


Figure 50 - Module timing parameters

ThostUndef – Amount of time that the Host ASIC has to change the Rate Select and Signal at delta T to the module. The Rate Select and Signal at delta T may conflict during this time. The signal may be at an undetermined value during this time including all zero, all one or any other non-deterministic pattern.

TmoduletxStable – Time for module to stabilize Signal at gamma T after Rate Select and Signal at delta T are at their proper values.

TmoduleroxStabl – Time for module to stabilize Signal at delta R once Rate Select and Signal at gamma R are at their proper values.

Module timing parameter values are specified in table 23.

Table 23 - Module timing parameter values

<u>Timing Parameter</u>	<u>Value maximum (ms)</u>
<u>ThostUndef</u>	1
<u>TmoduletxStable</u>	4
<u>TmoduleroxStabl</u>	4

9 Transmitter training

Note: Consider changing all references to Transmitter Training. Call it Link Training instead because it involves adjusting the Transmitter as well as adaptive equalization circuits in the receiver.

9.1 Scope

Transmitter training is a function of the FC-2P sublevel.

9.2 Overview

Transmitter training negotiates capabilities between the transmitters and receivers connected by a link:

- a) values of transmitter equalizer coefficients that result in most reliable signal reception across the link;
- b) values for receiver adaptive equalization circuits for reliable signal reception;
- c) use of FEC;
- d) Parallel Lane Support; and
- e) Extended Marker Present.

This clause specifies the protocol by which these capabilities shall be negotiated.

Transmitter training includes two steps:

- a) active training; and
- b) link quality check.

9.2.1 Transmitter training for 32GFC/16GFC

Active training is performed while transmitting and receiving the Transmitter Training Signal (see 5.5). Information in the Training Frame (see 5.5.2) portion of the Transmitter Training Signal is used to implement the protocol for negotiation of capabilities. The Training Pattern (see 5.5.3) portion of the Transmitter Training Signal allows each FC_Port to evaluate the received signal quality and recommend adjustments to the transmitter of the other FC_Port.

Training of transmitter equalizer coefficients is based on modeling the transmitter equalizer as having up to three coefficients that may be controlled by information in the Training Frame of the Transmitter Training Signal (see 5.5.2). The use of each coefficient is specified by FC-PI-x for each FC-0 variant that supports transmitter training. Each coefficient in the model has a minimum value, a maximum value, an initialize value, a preset value, and a step size by which it may be adjusted. These values are specified by FC-PI-x for each FC-0 physical variant that supports transmitter training.

An FC_Port that does not support training of transmitter equalizer coefficients acknowledges transmitter training commands but takes no action on its transmitter.

Training of transmitter equalizer coefficients presumes an adaptation process that determines the feedback requests to send to the remote transmitter and adjusts the local transmitter in response to feedback requests from the remote transmitter. The adaptation process observes the sequence of events specified by this standard, but the process by which it determines the need to send requests and adapts to received requests is not within the scope of this standard.

Link quality check confirms the ability of the link to be used for normal operation. Link quality check is performed while transmitting and receiving 64B/66B transmission code. Link quality check for frame transfer transmission codes other than 64B/66B is out of the scope of this standard.

9.2.2 Transmitter training for 64GFC

Transmitter training for 64GFC is performed using the Transmitter Training Signal for 64GFC specified in 5.7. The following sections describe how the control and status field values are generated during Transmitter Training.

Initial condition setting is done as specified in IEEE 802.3cd D2.1 136.8.11.4.1.

Coefficient update is done as specified in IEEE 802.3cd D2.1 136.8.11.4.2.

Handshake timing is done as specified in IEEE 802.3cd D2.1 136.8.11.6.

Variables, functions, timers, counters and state diagrams are specified in IEEE 802.3cd D2.1 136.8.11.7 and the max_wait_timer value is specified in 9.4.

9.3 Transmitter training state machines

9.3.1 Overview

Transmitter training shall cause link behavior equivalent to the state machines specified in 9.3.

Active training is specified by seven state machines operating concurrently:

- a) Training_Sequencer;
- b) a C_n_Controller for each coefficient n (i.e., n=-1, 0, 1) in the equalizer model; and
- c) a C_n_Responder for each coefficient n (i.e., n=-1, 0, 1) in the equalizer model.

Link quality check is specified by a single state machine, Link_Qual_Check.

The transitions among these state machines and with the FC_Port state machine are specified by the transmitter training flow diagrammed in figure 50.

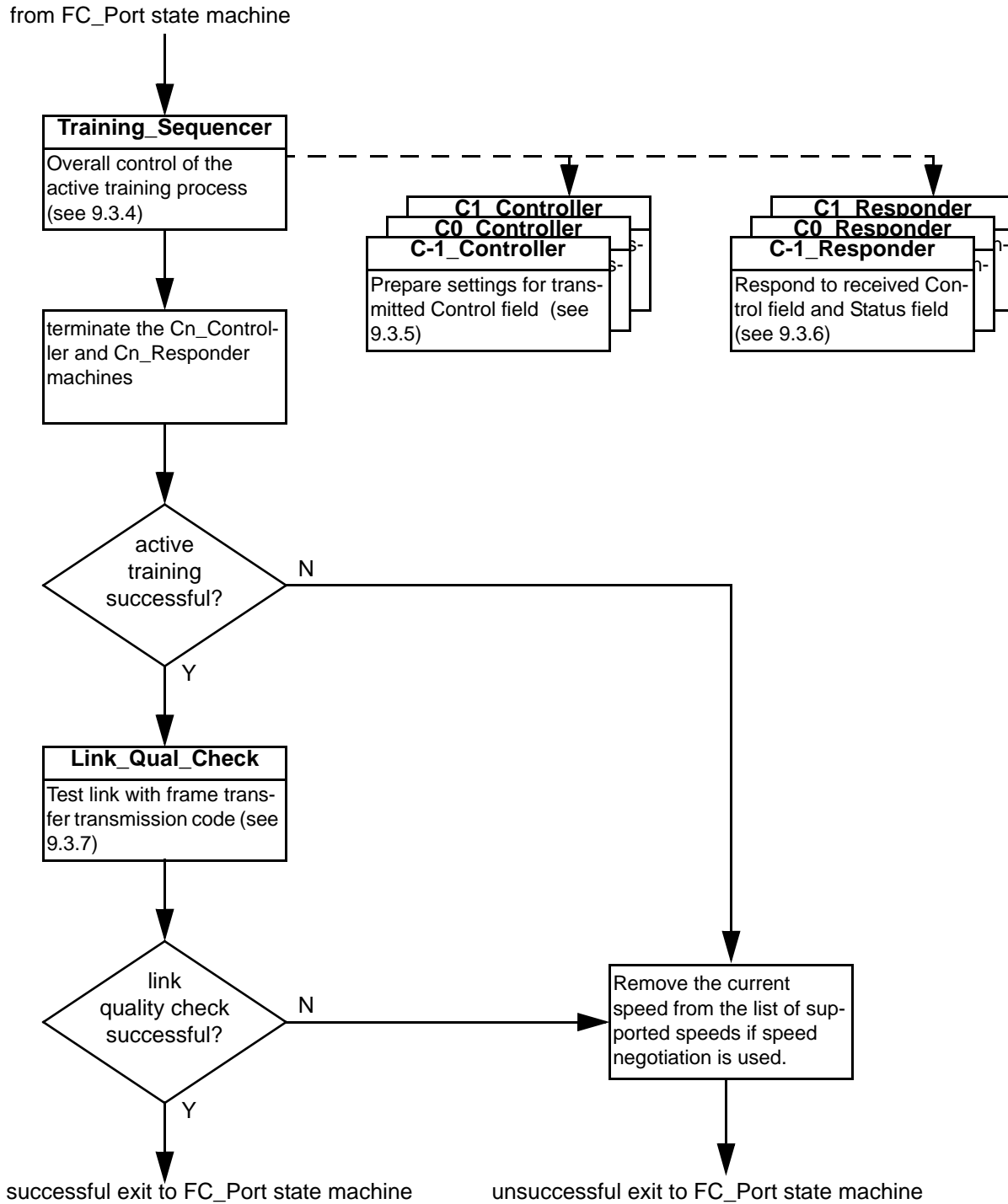


Figure 50 - Transmitter training flow

9.3.2 Timers

The timers specified in this subclause are visible to all state machines specified in 9.3.

train_fail_timer: a timer that limits the duration of active training. The train_fail_timer expires between 1 s and 1.5 s from the time it is started.

link_wait_timer: a timer that limits the duration in which the transmitter will transmit the Transmitter Training Signal at fixed settings after the remote FC_Port indicates training complete to ensure that remote FC_Port correctly detects the local interface state. The link_wait_timer expires between 32 μ s and 96 μ s from the time it is started.

link_test_timer: a timer that determines the delay in the LINK_TEST state before sampling of the link quality. The link_test_timer expires between 30 ms and 45 ms from the time it is started.

9.3.3 Variables

The variables specified in this subclause are visible to all state machines specified in 9.3.

These variables are set on decoding the values received in arriving Training Frames (see table 16 and table 17) during transmitter training. They are only set while the Transmitter Training Signal Transmission Word Synchronization state is Word Synchronization Acquired (see 6.5.1):

- a) **rcv_Preset:** the value in the Preset field of the Control field in the most recently received Training Frame;
- b) **rcv_Initialize:** the value in the Initialize field of the Control field in the most recently received Training Frame;
- c) **rcv_FECReq:** the value in the FECReq field of the Control field in the most recently received Training Frame;
- d) **rcv_C1Upd:** the value in the C1Upd field of the Control field in the most recently received Training Frame;
- e) **rcv_C0Upd:** the value in the C0Upd field of the Control field in the most recently received Training Frame;
- f) **rcv_C-1Upd:** the value in the C-1Upd field of the Control field in the most recently received Training Frame;
- g) **rcv_TC:** the value in the TC field of the Status field in the most recently received Training Frame;
- h) **rcv_SN:** the value in the SN field of the Status field in the most recently received Training Frame;
- i) **rcv_FECCap:** the value in the FECCap field of the Status field in the most recently received Training Frame;
- j) **rcv_TF:** the value in the TF field of the Status field in the most recently received Training Frame;
- k) **rcv_C1Stat:** the value in the C1Stat field of the Status field in the most recently received Training Frame;
- l) **rcv_C0Stat:** the value in the C0Stat field of the Status field in the most recently received Training Frame; and
- m) **rcv_C-1Stat:** the value in the C-1Stat field of the Status field in the most recently received Training Frame.

The term rcv_CnUpd is used to reference some member of rcv_C-1Upd, rcv_C0Upd, or rcv_C1Upd, specified by the context of the reference. The term rcv_CnStat is used to reference some member of rcv_C-1Stat, rcv_C0Stat, or rcv_C1Stat, specified by the context of the reference.

These variables contain the values that are set in transmitted Training Frames (see table 16 and table 17) while the Transmitter Training Signal is being used during transmitter training:

- a) **send_Preset**: the value to set in the Preset field of the Control field of subsequently sent Training Frames;
- b) **send_Initialize**: the value to set in the Initialize field of the Control field of subsequently sent Training Frames;
- c) **send_FECReq**: the value to set in the FECReq field of the Control field of subsequently sent Training Frames. The value of send_FECReq does not change;
- d) **send_C1Upd**: the value to set in the C1Upd field of the Control field of subsequently sent Training Frames;
- e) **send_C0Upd**: the value to set in the C0Upd field of the Control field of subsequently sent Training Frames;
- f) **send_C-1Upd**: the value to set in the C-1Upd field of the Control field of subsequently sent Training Frames;
- g) **send_TC**: the value to set in the TC field of the Status field of subsequently sent Training Frames;
- h) **send_SN**: the value to set in the SN field of the Status field of subsequently sent Training Frames;
- i) **send_FECCap**: the value to set in the FECCap field of the Status field of subsequently sent Training Frames. The value of send_FECCap does not change;
- j) **send_TF**: the value to set in the TF field of the Status field of subsequently sent Training Frames. The value of send_TF does not change;
- k) **send_C1Stat**: the value to set in the C1Stat field of the Status field of subsequently sent Training Frames;
- l) **send_C0Stat**: the value to set in the C0Stat field of the Status field of subsequently sent Training Frames; and
- m) **send_C-1Stat**: the value to set in the C-1Stat field of the Status field of subsequently sent Training Frames.

The term send_CnUpd is used to reference some member of send_C-1Upd, send_C0Upd, or send_C1Upd, specified by the context of the reference. The term send_CnStat is used to reference some member of send_C-1Stat, send_C0Stat, or send_C1Stat, specified by the context of the reference.

9.3.4 Training_Sequencer state machine

9.3.4.1 Overview

This state machine starts the concurrent state machines that manage training of individual equalizer coefficients (see 9.3.5 and 9.3.6), and then conducts the protocol to determine when training has become stable or failed. A diagram for the Training_Sequencer state machine is given in figure 51.

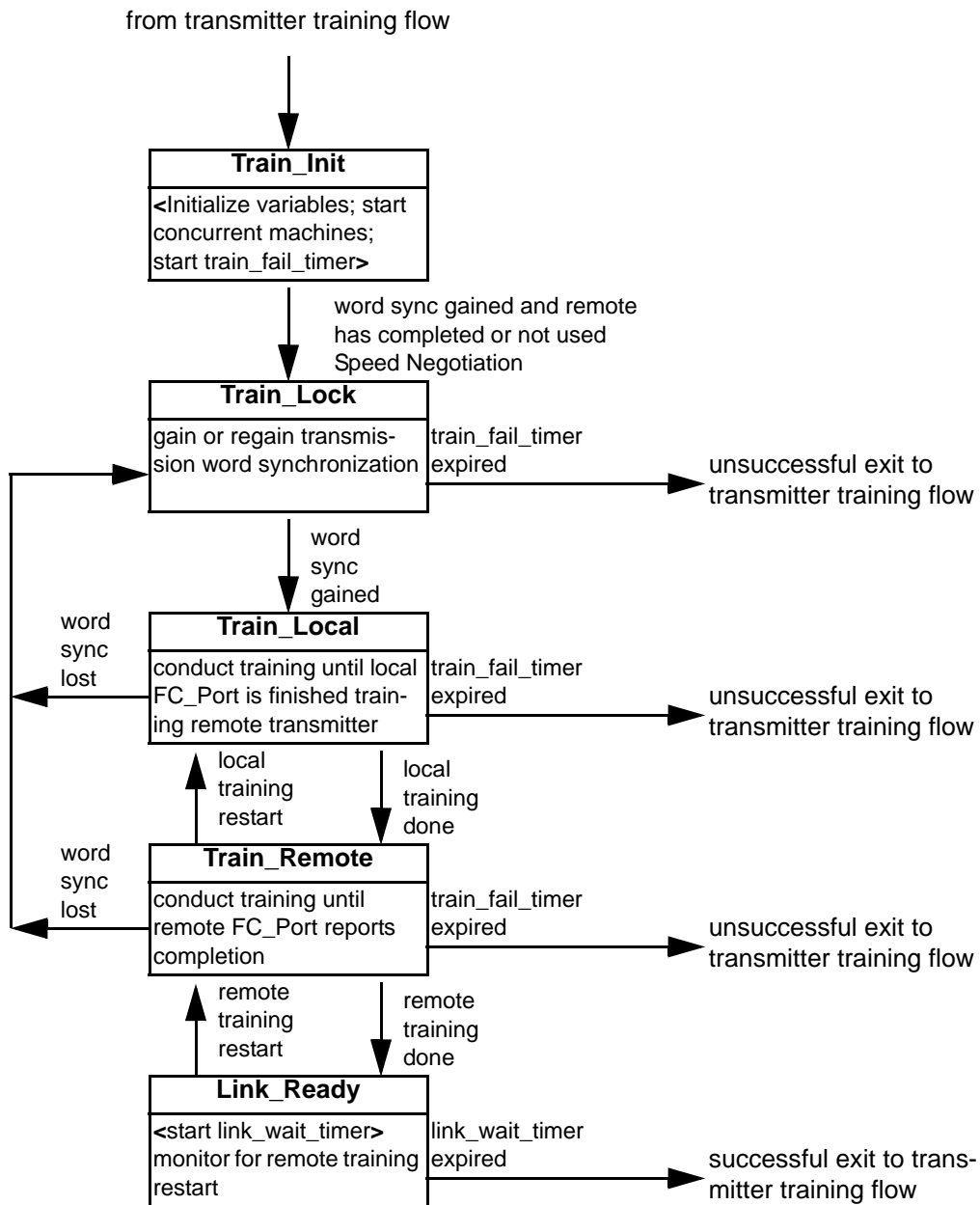


Figure 51 - Diagram of Training Sequencer state machine flow

9.3.4.2 States

9.3.4.2.1 Train_Init

The Train_Init state initializes the variables and watchdog timer used by the state machine that specifies the process of actively negotiating transmitter capabilities, and then awaits the remote FC_Port indicating readiness for negotiation.

The actions on entry to the Train_Init state are:

- 1) initialize the variables (see 9.3.3) necessary for the operation of the remaining state machines:
 - a) set rcv_Preset to zero;
 - b) set rcv_Initialize to zero;
 - c) set rcv_FECReq to zero;
 - d) set all rcv_CnUpd to 00b;
 - e) set rcv_TC to zero;
 - f) set rcv_SN to one;
 - g) set rcv_FECCap to zero;
 - h) set rcv_TF to one;
 - i) set all rcv_CnStat to 00b;
 - j) set send_Preset to zero;
 - k) set send_Initialize to zero;
 - l) if the port does not request use of FEC, then set send_FECReq to zero;
 - m) if the port requests use of FEC, then set send_FECReq to one;
 - n) set all send_CnUpd to 00b;
 - o) set send_TC to zero;
 - p) set send_SN to zero;
 - q) if the port does not support use of FEC, then set send_FECCap to zero;
 - r) if the port supports use of FEC, then set send_FECCap to one; and
 - s) if the FC_Port allows training of transmitter coefficients, then set send_TF to zero;
 - t) if the FC_Port does not allow training of transmitter coefficients, then set send_TF to one;
 - u) set all send_CnStat to 00b;
 - v) for 32GFC and 128GFC set Extended Marker (see table 16) to 11b, other speeds set to 00b; and
 - w) for training the Parallel Lane Support (see table 16) field is not meaningful;
- 2) set all of the transmitter equalizer coefficients to their initialize values (see FC-PI-x);
- 3) begin or continue transmitting the Transmitter Training Signal (see 5.5);
- 4) if the FC_Port supports training of transmitter coefficients, then start the Cn_Controller state machines (see 9.3.5);
- 5) start the Cn_Responder state machines (see 9.3.6); and
- 6) start the train_fail_timer (see 9.3.2).

The actions while remaining in the Train_Init state are:

- a) transmit and receive the Transmitter Training Signal (see 5.5);
- b) monitor rcv_SN; and
- c) monitor the train_fail_timer.

The transitions from the Train_Init state are:

- a) if the value of rcv_SN is zero, then transition to the Train_Lock state; or
- b) if the train_fail_timer expires, then exit from the Training Sequencer state machine indicating active training is unsuccessful.

9.3.4.2.2 Train_Lock

The Train_Lock state establishes or recovers Transmitter Training Signal Transmission Word Synchronization (see 6.5) during the process of actively negotiating transmitter equalization.

There are no actions on entry to the Train_Lock state.

The actions while remaining in the Train_Lock state are:

- a) transmit the Transmitter Training Signal;
- b) monitor Transmitter Training Signal Transmission Word Synchronization (see 6.5); and
- c) monitor the train_fail_timer.

The transitions from the Train_Lock state are:

- a) if Transmitter Training Signal Transmission Word Synchronization is detected, then transition to the Train_Local state; or
- b) if the train_fail_timer expires, then exit from the Training Sequencer state machine indicating active training is unsuccessful.

9.3.4.2.3 Train_Local

The Train_Local state establishes or re-establishes stable equalization of the remote transmitter by the local FC_Port during the process of actively negotiating transmitter capabilities.

The actions on entry to the Train_Local state are:

- a) set send_TC to zero.

The actions while remaining in the Train_Local state are:

- a) if the value of send_TF is zero, then monitor for completion of the adaptation process in the local FC_Port, which is not within the scope of this standard;
- b) monitor Transmitter Training Signal Transmission Word Synchronization (see 6.5); and
- c) monitor the train_fail_timer.

The transitions from the Train_Local state are:

- a) if completion of the adaptation process in the local FC_Port is detected, then transition to the Train_Remote state;
- b) if the value of send_TF is one, then transition to the Train_Remote state;
- c) if the value of rcv_TF is one, then transition to the Train_Remote state;
- d) if loss of Transmitter Training Signal Transmission Word Synchronization is detected, then transition to the Train_Lock state; or
- e) if the train_fail_timer expires, then exit from the Training Sequencer state machine indicating active training is unsuccessful.

9.3.4.2.4 Train_Remote

The Train_Remote state establishes stable equalization of the local transmitter by the remote FC_Port during the process of actively negotiating transmitter capabilities.

The actions on entry to the Train_Remote state are:

- a) set send_TC to one.

The actions while remaining in the Train_Remote state are:

- a) monitor the value of rcv_TC;
- b) monitor the value of rcv_TF;
- c) if the value of send_TF is zero and rcv_TF is set to zero, then monitor for resumption of the adaptation process in the local FC_Port, which is not within the scope of this standard;
- d) monitor Transmitter Training Signal Transmission Word Synchronization (see 6.5); and
- e) monitor the train_fail_timer.

The transitions from the Train_Remote state are:

- a) if the value of rcv_TC is one, then transition to the Link_Ready state;
- b) if the value of send_TF is zero and the value of rcv_TF is zero and resumption of the adaptation process in the local FC_Port is detected, then transition to the Train_Local state;
- c) if loss of Transmitter Training Signal Transmission Word Synchronization is detected, then transition to the Train_Lock state; or
- d) if the train_fail_timer expires, then exit from the Training Sequencer state machine indicating active training is unsuccessful.

9.3.4.2.5 Link_Ready

The Link_Ready state confirms stable negotiation between the local and remote FC_Ports during the process of actively negotiating transmitter equalization.

The actions on entry to the Link_Ready state are:

- a) start the link_wait_timer.

The actions while remaining in the Link_Ready state are:

- a) monitor the value of rcv_TC; and
- b) monitor the link_wait_timer.

The transitions from the Link_Ready state are:

- a) if the value of rcv_TC is zero, then transition to the Train_Remote state; or
- b) if the link_wait_timer expires, then exit from the Training Sequencer state machine indicating active training is successful.

9.3.5 Cn_Controller state machines

9.3.5.1 Overview

If the FC_Port supports training of transmitter coefficients, then there is an instance of the Cn_Controller state machine specific to each of the coefficients of the model transmitter equalizer (i.e., C1_Controller, C0_Controller and C-1_Controller). Each Cn_Controller controls the setting of its specific send_CnUpd variable, and acts on the setting of its specific rcv_CnStat variable. Cn_Controller state machines are instantiated at the start of the Training_Sequencer state machine, and terminated when the Training_Sequencer state machine terminates. When a Cn_Controller state machine is instantiated, it enters the Tx_Ready state. A diagram for the Cn_Controller state machine is given in figure 52.

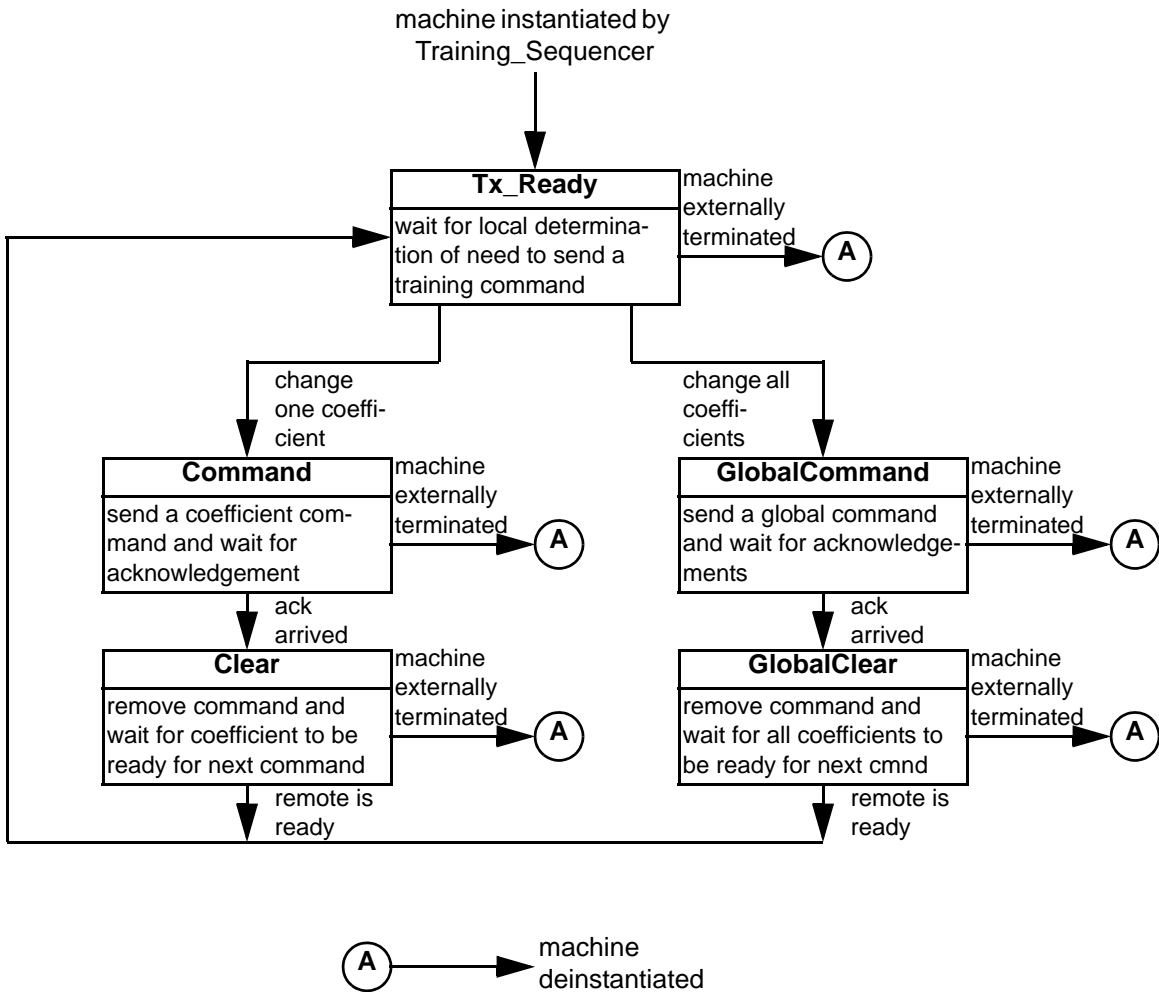


Figure 52 - Diagram of Cn_Controller state machine flow

9.3.5.2 States

9.3.5.2.1 Tx_Ready

In the Tx_Ready state, the Cn_Controller state machine has confirmed completion of its prior update command and does not need to update its coefficient further.

The actions on entry to the Tx_Ready state are:

- a) set send_Preset to zero;
- b) set send_Initialize to zero; and
- c) set send_CnUpd to 00b for the coefficient managed by this Cn_Controller state machine.

The actions while remaining in the Tx_Ready state are:

- a) monitor the value of rcv_TF. If the value of rcv_TF is zero, then:
 - A) monitor for the need to set all coefficients to their initialize values (see FC-PI-x);
 - B) monitor for the need to set all coefficients to their preset values (see FC-PI-x); and
 - C) monitor for the need to increment or decrement the coefficient negotiated by this Cn_Controller state machine.

The processes by which a Cn_Controller determines the need to update the coefficient in the remote transmitter that it negotiates and reset the negotiation are not within the scope of this standard; however, these processes shall not indicate the need for more than one command at the same time that affects the same coefficient.

The transitions from the Tx_Ready state are:

- a) if the value of rcv_TF is zero, the values of rcv_CnStat for all coefficients are 00b, and the Cn_Controller state machine determines the need to set all coefficients to their initialize values, then transition to the GlobalCommand state;
- b) if the value of rcv_TF is zero, the values of rcv_CnStat for all coefficients are 00b, and the Cn_Controller state machine determines the need to set all coefficients to their preset values, then transition to the GlobalCommand state; or
- c) If the value of rcv_TF is zero and the value of the rcv_CnStat for the coefficient to be adjusted is 00b, and the Cn_Controller state machine determines the need to increment or decrement the coefficient negotiated by this Cn_Controller state machine, then transition to the Command state.

9.3.5.2.2 Command

In the Command state, the Cn_Controller is sending a command that affects only its coefficient, and is waiting for acknowledgement that the command was received.

The actions on entry to the Command state are:

- a) if the Cn_Controller state machine has determined the need to increment the coefficient negotiated by this Cn_Controller state machine, then set send_Preset to zero, set send_Initialize to zero, and set send_CnUpd to 01b for the coefficient negotiated by this Cn_Controller state machine; or
- b) if the Cn_Controller state machine has determined the need to decrement the coefficient negotiated by this Cn_Controller state machine, then set send_Preset to zero, set send_Initialize

to zero, and set send_CnUpd to 10b for the coefficient negotiated by this Cn_Controller state machine.

The actions while remaining in the Command state are:

- a) monitor the value of rcv_CnStat for the coefficient negotiated by this Cn_Controller state machine.

The transitions from the Command state are:

- a) if the value of rcv_CnStat for the coefficient negotiated by this Cn_Controller state machine is not 00b, then transition to the Clear state.

9.3.5.2.3 Clear

In the Clear state, the Cn_Controller has received acknowledgement for a command that affects only its coefficient, and is waiting for notification that the remote FC_Port is ready for another command.

The actions on entry to the Clear state are:

- a) set send_Preset to zero;
- b) set send_Initialize to zero; and
- c) set send_CnUpd to 00b for the coefficient managed by this Cn_Controller state machine.

The actions while remaining in the Clear state are:

- a) monitor the value of rcv_CnStat for the coefficient negotiated by this Cn_Controller state machine.

The transitions from the Clear state are:

- a) if the value of rcv_CnStat for the coefficient negotiated by this Cn_Controller state machine is 00b, then transition to the Tx_Ready state.

9.3.5.2.4 GlobalCommand

In the GlobalCommand state, the Cn_Controller is sending a command that affects all coefficients, and is waiting for acknowledgement that the command was received.

The processes by which a Cn_Controller determines the need to update the coefficient in the remote transmitter that it negotiates and reset the negotiation are not within the scope of this standard; however, these processes shall not indicate the need for more than one command at the same time that affects the same coefficient.

The actions on entry to the GlobalCommand state are:

- a) if the Cn_Controller state machine determines the need to reset all coefficients to their preset values, then set send_Preset to one, set send_Initialize to zero, and set send_CnUpd to 00b for all coefficients; or
- b) if the Cn_Controller state machine has determined the need to set all coefficients to their initialize values, then set send_Preset to zero, set send_Initialize to one, and send_CnUpd to 00b for all coefficients.

The actions while remaining in the GlobalCommand state are:

- a) monitor the values of rcv_CnStat for all coefficients.

The transitions from the GlobalCommand state are:

- a) if the values of rcv_CnStat for all coefficients are nonzero, then transition to the GlobalClear state.

9.3.5.2.5 GlobalClear

In the GlobalClear state, the Cn_Controller has received acknowledgement for a command that affects all coefficients, and is waiting for notification that the remote FC_Port is ready for another command.

The actions on entry to the GlobalClear state are:

- a) set send_Reset to zero;
- b) set send_Initialize to zero; and
- c) set send_CnUpd to 00b for all coefficients.

The actions while remaining in the GlobalClear state are:

- a) monitor the values of rcv_CnStat for all coefficients.

The transitions from the GlobalClear state are:

- a) if the values of rcv_CnStat for all coefficients are 00b, then transition to the Tx_Ready state.

9.3.6 Cn_Responder state machines

9.3.6.1 Overview

There is an instance of the Cn_Responder state machine specific to each of the coefficients of the model transmitter equalizer (i.e., C1_Responder, C0_Responder and C-1_Responder). Each Cn_Responder acts on the setting of its specific rcv_CnUpd variable, and controls the setting of its specific send_CnStat variable. Cn_Responder state machines are instantiated at the start of the Training_Sequencer state machine, and terminated when the Training_Sequencer state machine terminates. When a Cn_Responder state machine is instantiated, it enters the Rx_Ready state. A diagram for the Cn_Responder state machine is given in figure 53.

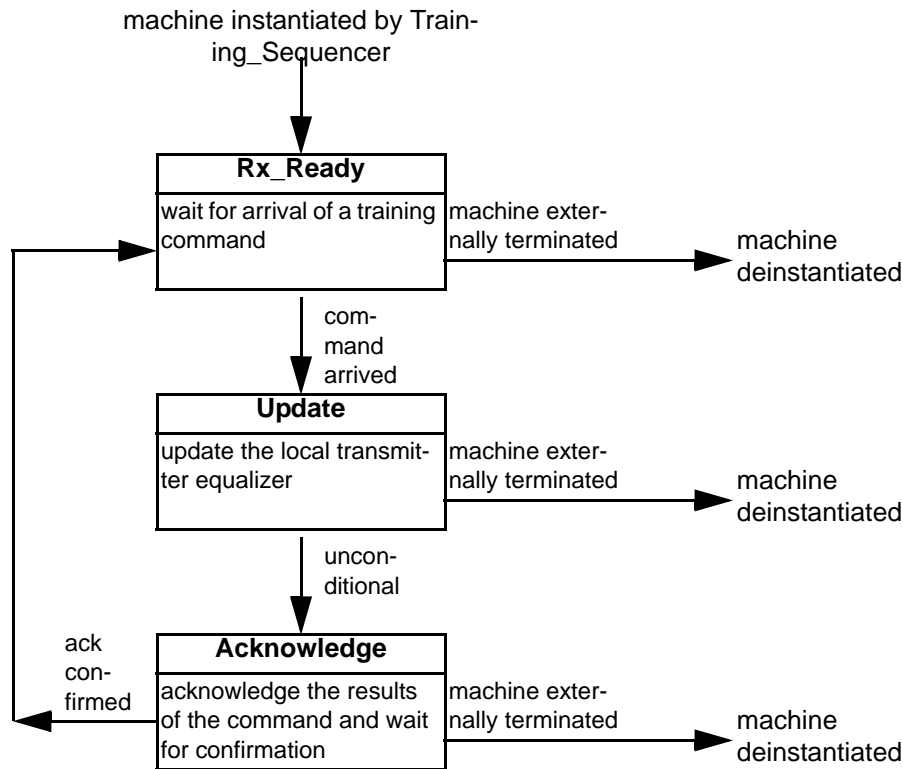


Figure 53 - Diagram of Cn_Responder state machine flow

9.3.6.2 States

9.3.6.2.1 Rx_Ready

In the Rx_Ready state, the Cn_Responder state machine is ready to process another request to change the transmitter equalizer coefficient managed by this Cn_Responder state machine.

The actions on entry to the Rx_Ready state are:

- a) set send_CnStat to 00b for the coefficient managed by this Cn_Responder state machine.

The actions while remaining in the Rx_Ready state are:

- a) monitor the value of rcv_CnUpd for the coefficient managed by this Cn_Responder state machine;
- b) monitor the value of rcv_Initialize; and
- c) monitor the value of rcv_Preset.

The transitions from the Rx_Ready state are:

- a) if the value of rcv_CnUpd is nonzero for the coefficient managed by this Cn_Responder state machine, then transition to the Update state;
- b) if the value of rcv_Initialize is nonzero, then transition to the Update state; and
- c) if the value of rcv_Preset is nonzero, then transition to the Update state.

9.3.6.2.2 Update

In the Update state, the Cn_Responder state machine processes a command that affects the coefficient managed by this Cn_Responder state machine and reports the resulting status to the sender of the command.

The actions on entry to the Update state are:

- a) if the value of send_TF is one, then set send_CnStat to any nonzero value for the coefficient managed by this Cn_Responder state machine;
- b) if:
 - A) the value of send_TF is zero; and
 - B) the value of rcv_Preset is one,
 then set the coefficient managed by this Cn_Responder state machine to its preset value (see FC-PI-x) and then:
 - A) if the coefficient managed by this Cn_Responder state machine is not at its minimum value and not at its maximum value, then set send_CnStat to 01b for the coefficient managed by this Cn_Responder state machine;
 - B) if the coefficient managed by this Cn_Responder state machine is at its minimum value, then set send_CnStat to 10b for the coefficient managed by this Cn_Responder state machine; or
 - C) if the coefficient managed by this Cn_Responder state machine is at its maximum value, then set send_CnStat to 11b for the coefficient managed by this Cn_Responder state machine;
- c) if:
 - A) the value of send_TF is zero;
 - B) the value of rcv_Preset is zero; and
 - C) the value of rcv_Initialize is one,
 then set the coefficient managed by this Cn_Responder state machine to its initialize value (see FC-PI-x) and set send_CnStat to 01b for the coefficient managed by this Cn_Responder state machine;
- d) if
 - A) the value of send_TF is zero;
 - B) the value of rcv_Preset is zero;
 - C) the value of rcv_Initialize is zero; and
 - D) the value of rcv_CnUpd is 01b for the coefficient managed by this Cn_Responder state machine,
 then:
 - A) if the coefficient managed by this Cn_Responder state machine is at its maximum value, then set send_CnStat to 11b for the coefficient managed by this Cn_Responder state machine; or
 - B) if the coefficient managed by this Cn_Responder state machine is not at its maximum value then increment the coefficient managed by this Cn_Responder state machine by its step size (see FC-PI-x) and then:

- a) if the coefficient managed by this Cn_Responder state machine is not at its maximum value, then set send_CnStat to 01b for the coefficient managed by this Cn_Responder state machine; or
 - b) if the coefficient managed by this Cn_Responder state machine is at its maximum value, then set send_CnStat to 11b for the coefficient managed by this Cn_Responder state machine;
- or
- e) if
 - A) the value of send_TF is zero;
 - B) the value of rcv_Preset is zero;
 - C) the value of rcv_Initialize is zero; and
 - D) the value of rcv_CnUpd is 10b for the coefficient managed by this Cn_Responder state machine,
 then:
 - A) if the coefficient managed by this Cn_Responder state machine is at its minimum value, then set send_CnStat to 10b for the coefficient managed by this Cn_Responder state machine; or
 - B) if the coefficient managed by this Cn_Responder state machine is not at its minimum value then decrement the coefficient managed by this Cn_Responder state machine by its step size (see FC-PI-x) and then:
 - a) if the coefficient managed by this Cn_Responder state machine is not at its minimum value, then set send_CnStat to 01b for the coefficient managed by this Cn_Responder state machine; or
 - b) if the coefficient managed by this Cn_Responder state machine is at its minimum value, then set send_CnStat to 10b for the coefficient managed by this Cn_Responder state machine.

There are no actions while remaining in the Update state.

The Update state transitions to the Acknowledge state on completing its actions on entry.

9.3.6.2.3 Acknowledge

In the Acknowledge state, the Cn_Responder maintains the status of its most recently processed command until the sender of the command indicates that the status has been received.

There are no actions on entry to the Acknowledge state.

The actions while remaining in the Acknowledge state are:

- a) monitor the value of rcv_Preset;
- b) monitor the value of rcv_Initialize; and
- c) monitor the value of rcv_CnUpd for the coefficient managed by this Cn_Responder state machine.

The transitions from the Acknowledge state are:

- a) If:
 - A) the value of rcv_Preset is zero;
 - B) the value of rcv_Initialize is zero; and

- C) the value of `rcv_CnUpd` is zero for the coefficient managed by this `Cn_Responder` state machine,
then transition to the `Rx_Ready` state.

9.3.7 Link_Qual_Check state machine

9.3.7.1 Overview

This state machine verifies that the `FC_Port` is able to reliably communicate over the link using 64B/66B frame transfer transmission protocol (see 5.3). In this state machine, the NOS Primitive Sequence is transmitted.

9.3.7.2 States

9.3.7.2.1 Link_Test

The `Link_Test` state is the only state in the `Link_Qual_Test` state machine. It begins using the 64B/66B transmission code, delays long enough for both the local and remote `FC_Port` to synchronize to the 64B/66B transmission code, and then verifies 64B/66B Transmission Word Synchronization has been achieved.

The actions on entry to the `Link_Test` state are:

- 1) begin transmitting 64B/66B transmission code (see 5.3) with FEC determined by:
 - A) if either `send_FECCap` or `rcv_FECCap` is set to zero, then do not use FEC;
 - B) if neither `send_FECReq` nor `rcv_FECReq` is set to one, then do not use FEC; and
 - C) if both `send_FECCap` and `rcv_FECCap` are set to one and either `send_FECReq` or `rcv_FECReq` is set to one, then use FEC;
 and
- 2) start the `link_test_timer` (see 9.3.2).

The actions while remaining in the `Link_Test` state are:

- 1) continue transmitting 64B/66B transmission code, with use of FEC as determined on entry to the `Link_Test` state, until the `link_test_timer` expires; and
- 2) Determine if Transmission Word Synchronization (see 6.4.1) is indicated.

The transitions from the `Link_Test` state are:

- a) if Transmission Word Synchronization is indicated, then exit from the `Link_Qual_Check` state machine indicating that link quality check was successful; or
- b) if Transmission Word Synchronization is not indicated, then exit from the `Link_Qual_Check` state machine indicating that link quality check was unsuccessful.

If the `Link_Test` state exits indicating that link quality check was successful, then the transmission code selected on entry to the `Link_Test` state continues to be used in normal operation.

9.4 Link Speed Negotiation/Transmitter Training flow diagram for 64GFC

An example of the flow as the link moves from LSN to Transmitter Training for 64GFC links is shown in Figure 54.

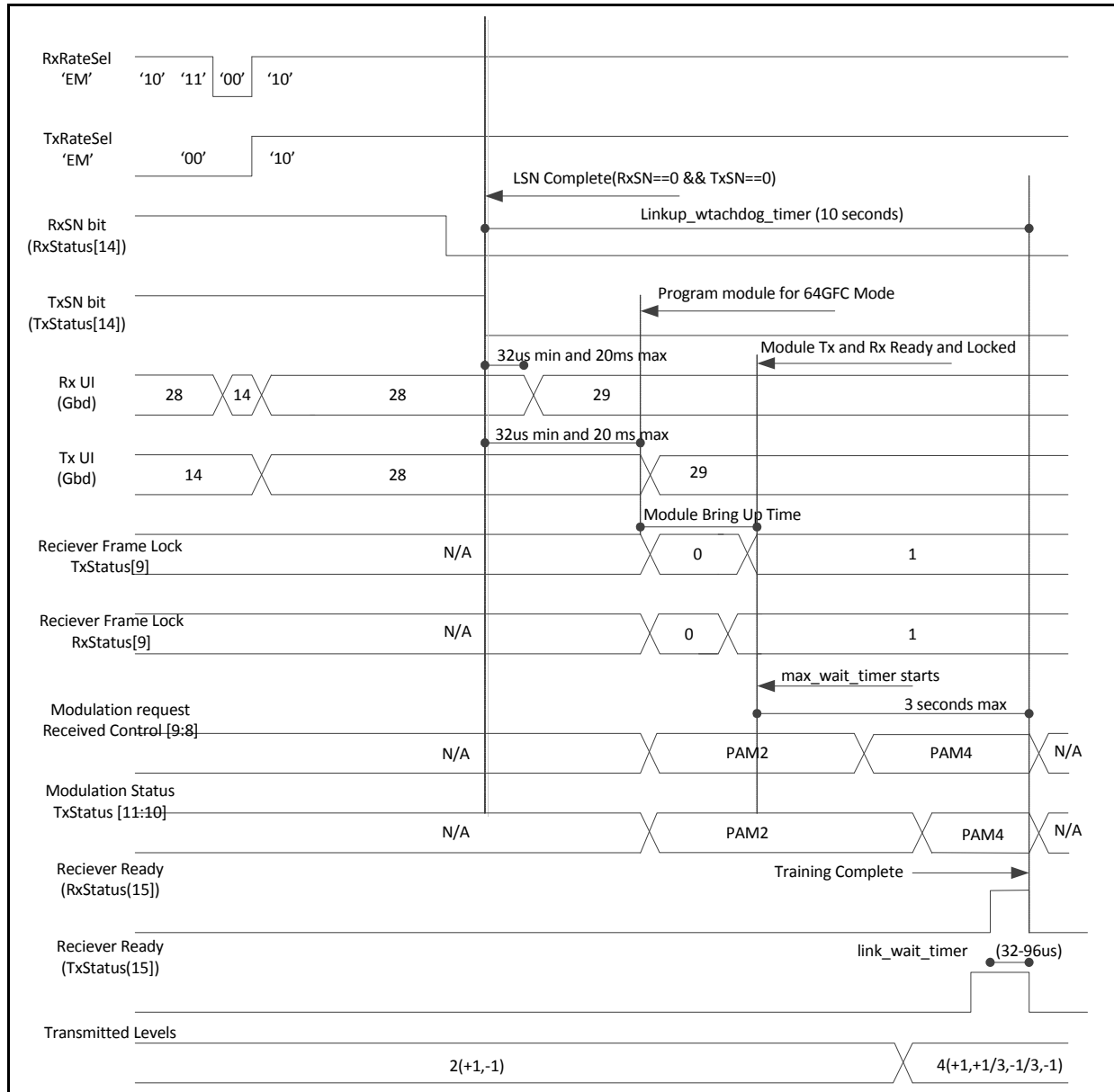


Figure 54 - Link speed negotiation to Transmitter Training for 64GFC links

When the link has set Training Field Status Bit 14(SN) to zero in the transmitted TTS and receives Training Field Status bit 14(SN) as zero in the received TTS, LSN is complete. The link shall transmit the TTS for LSN for a minimum of 32us (lsn_end_wait_timer) after LSN is complete.

The link must start transmitting the 28.9Gbaud TTS frame for 64GFC within a maximum time of 20ms (lsn_end_training_start_timer) from when LSN is complete.

If the transmitter training is being run across an optical link, the optical module shall be programmed to run at the 64GFC data rate upon expiration of the lsn_end_training_start timer. After completion of module programming, the link will wait for the optical module to indicate that it is ready to transmit and receive data at the 64GFC data rate by reading appropriate status bits in the optical module. Once the module indicates a ready status, the host ASIC shall acquire lock to the 64GFC TTS frame.

If the transmitter training is being run across an electrical link the steps related to module programming and checking for status shall not be executed. Instead the host ASIC shall attempt to acquire lock to the 64GFC TTS frame upon expiration of the lsn_end_training_start timer.

Once lock is achieved to the 64GFC TTS frame in the host ASIC, the link shall set Training Frame Status Field Bit 9(Receiver Frame Lock) to let the remote side know that it has achieved lock to the TTS signal. When it receives Training frame Status Field Bit 9(Receiver Frame Lock) from the remote side, this indicates that tuning of the link parameters can start, and the max_wait timer shall be started. Transmitter Training starts with a PAM2 Training Pattern (section 5.7). It switches to PAM4 once tuning of parameters for PAM2 is complete.

When the link determines that tuning of link parameters is complete it shall set Training Frame Status Field Bit 15(Receiver Ready) to one. When Training Frame Status Field Bit 15 in transmit and receive TTS is one, Transmitter training is complete. The process from LSN complete to Transmitter Training complete must be completed before linkup_watchdog timer expires.

The link will now enter the LINK_READY state (see IEEE 802.3cd D2.1 figure 136-7). From the Link Ready state, upon expiration of the link_wait timer, the link will move to the Link Quality Check State Machine (see 9.3.7) to perform a Link test. FEC is mandatory for 64GFC links, so exchanged FEC capability bits are ignored while performing the Link test.

lsn_end_wait timer

This timer is started after LSN is complete. The link sends additional TTS frames until this timer expires to ensure that the remote link partner receives a sufficient number of training frames to detect the link state. The minimum value of this timer shall be 32us.

lsn_end_training_start timer

This timer is started after LSN is complete. The link starts switching its host ASIC to transmit the TTS frames (see 5.7) for 64GFC transmitter training after meeting the requirements specified in lsn_end_wait timer and must complete this switch before lsn_end_training_start timer expires. The maximum value of this timer shall be 20ms.

max_wait timer

This timer sets the limit on how long transmitter training is allowed to operate to find the optimal transmit coefficients and receiver adaptive equalization values for reliable link operation. For 64GFC links, the value of the max_wait timer shall be 3 seconds.

linkup_watchdog timer

This timer is started when LSN is complete. This timer sets the maximum amount of time from LSN complete to transmitter training complete. The value of this timer shall be set to 10 seconds.

link_wait timer

A timer that limits the duration in which the transmitter will transmit the Transmitter Training Signal at fixed settings after the remote FC Port indicates training complete to ensure that remote FC Port correctly detects the local interface state. The link_wait timer expires between 32us and 96us from the time it is started.