Interlaken Reed-Solomon Forward Error Correction Extension

Protocol Definition

February 2020

Revision 1.1

Terms and Conditions

Any company or individual wishing to use all or a part of this document may do so only if they relinquish their proprietary rights to information contained or referenced herein. You agree not to enforce those intellectual property rights against any other party implementing or advocating the implementation of this document.

YOU ACKNOWLEDGE THAT THIS DOCUMENT IS PROVIDED "AS IS" AND WITHOUT WARRANTY OF ANY KIND. FOR THE SAKE OF CLARITY, ALL PARTIES USING OR ADVOCATING THE USE OF THIS DOCUMENT DISCLAIM ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT, INCLUDING WITHOUT LIMITATION ANY AND ALL WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT.

IN NO EVENT WILL ANY PARTY THAT UTILIZES OR ADVOCATES THE USE OF THIS DOCUMENT BE LIABLE TO ANY OTHER PARTY FOR ANY INDIRECT, INCIDENTAL OR CONSEQUENTIAL DAMAGES IN CONNECTION WITH THIS DOCUMENT, AND IN NO EVENT WILL THE CUMULATIVE LIABILITY OF ANY SINGLE PARTY FOR ANY AND ALL CLAIMS IN CONNECTION WITH THIS DOCUMENT EXCEED, IN THE AGGREGATE, ONE HUNDRED DOLLARS ($100.00). YOU AGREE THAT THIS PARAGRAPH IS AN ESSENTIAL BASIS OF THE USE OF THIS DOCUMENT.

Contents

Revision History

Revision 1.0 **7 Dec 2016**
●Initial public release of the document
Revision 1.1 **28 Feb 2020**
●Added information related to 116G lane support

1 Introduction

Interlaken connects components via multiple high-speed serialized links. The Interlaken Protocol Specification defines a high-speed, channelized, chip-to-chip packet interface that is independent of the number of SERDES lanes and rates.

As SERDES speeds increase beyond 30Gbps, to 58Gbps for initial applications, and later to 116Gbps, electrical channel loss and other impairments may limit reach. This motivates the use of signaling schemes such as PAM4 which transmit multiple bits per symbol. However, without increasing signal swing levels, margins are reduced to the point that SERDES bit error rates (BER) in the 10^{-4} to 10^{-6} range are expected instead of the traditional expectation of better than 10^{-15}. This higher SERDES error rate would represent a significant impact to higher layer protocols, and so improvement is required.

Forward Error Correction (FEC) techniques can achieve a BER of $<10^{-15}$ with an input BER of $>10^{-6}$ and are assumed or required by most applicable Electrical Interface Standards using SERDES running between 36-58Gbps (CEI-56G-MR-PAM4 from OIF-CEI-04.0, IEEE Std 802.3bs™-2017, IEEE Std 802.3cd™-2018, among others) and between 72-116G (CEI-112G-MR from upcoming OIF-CEI-05.0, upcoming IEEE 802.3ck™). Future higher rate SERDES will also require FEC.

Gray-coding of the transmitted data and precoding of the data to limit burst-error run lengths are also important considerations.

This extension defines the application of FEC, gray-coding and precoding to an Interlaken interface using PAM4 SERDES at rates between 36-58Gbps and between 72-116Gbps.

This extension does not require alteration of the operation of the regular Interlaken protocol but does cause some changes to the semantics. Future implementations may find that certain existing Interlaken Protocol features become redundant with the addition of FEC.

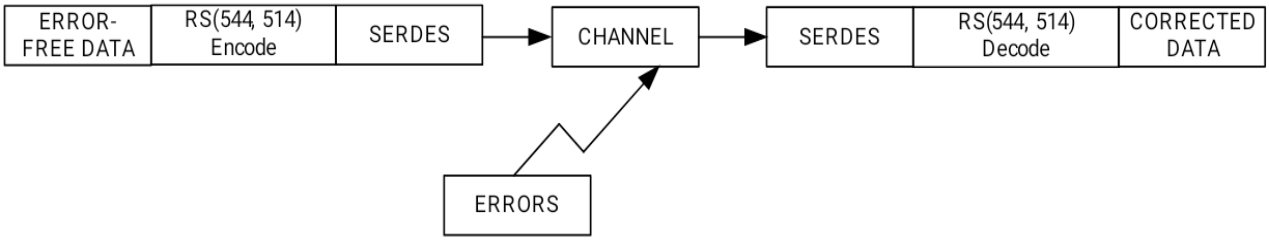
2 Considerations when using 36-58Gbps or 72-116Gbps PAM4 SERDES

2.1 Forward Error Correction (FEC)

Several protocols designed to work with 36-58Gbps PAM4 SERDES have selected an RS(n=544, k=514, t=15, m=10) Forward Error Correction (FEC) code, which is often abbreviated to RS(544, 514). It is a code from the Reed-Solomon group of error-correcting codes with an n=544 symbol codeword, k=514 symbol data field, a symbol size m=10 bits, and has the ability to correct t=15 symbol errors, and detect 2t=30 symbol errors. A FEC scheme is constructed by encoding the data using the RS(544, 514) code to generate the parity or check symbols before transmission, and using a decoder to find and correct the errors after reception.

A simplified transmission system using FEC is illustrated in Figure 1.

Figure 1: Transmission System using Forward Error Correction



2.2 Gray Coding

When the electrical interface is PAM4 encoded, each PAM4 symbol represents 2 bits. A simple receiver views the incoming signal as being at one of 4 levels when it decides on the signal level. Channel losses and other impairments contribute to a change in signal level at the receiver. 1-level errors will be observed much more often than 2-level and 3-level errors. Thus, the data in PAM4 systems is commonly gray-coded such that an error that moves the signal to an adjacent signal level only causes a single bit error at the receiver, and more importantly, only causes a single FEC symbol error.

2.3 Precoding

Some electrical channels and/or receiver architectures may require a decision-feedback equalizer (DFE) architecture with a very large coefficient for the first tap. Any decision errors made at the DFE slicer output can persist over multiple bits as the errors are fed back to the slicer input. If an error of +1 is made, then the error fed back to the input will consist of a scaled alternating -1, +1 sequence superimposed on the correct signal. To counter this effect, a precoder encodes data as output(t)= (input(t)-output(t-1)) MOD 4 before transmission, and after the receiver DFE the decoder decodes it as output(t) = (input(t) + output(t-1)) MOD 4. Since this decoder sums the current and previous decisions from the DFE the alternating error sequence cancels itself after one error. When the DFE

eventually "clears" the error, the transition from the -1, +1 sequence back to an all-zero error sequence will cause the decoder to emit a second error.

With such a precoder and a burst error in the DFE, there is an error output at the start of the error burst, and another error output at the end of the error burst. The potentially long burst error at the DFE output is avoided and is converted to a pair of errors. The downside is that single PAM4 symbol errors are converted into two PAM4 symbol errors. For this reason, the use of the precoder/decoder pair to optimize overall BER is application dependent.

2.4 Baseline Wander

The requirement to use DC balanced coding is not new to Interlaken as the original protocol definition took additional steps to limit baseline wander and keeps running disparity within +/-96 bits. One benefit of this is enabling the use of smaller AC coupling capacitors which may be integrated into a silicon device. Other protocols such as Ethernet have successfully used scrambling schemes which provide statistical bounds on running disparity, but require a larger AC coupling capacitor to keep baseline wander within acceptable limits.

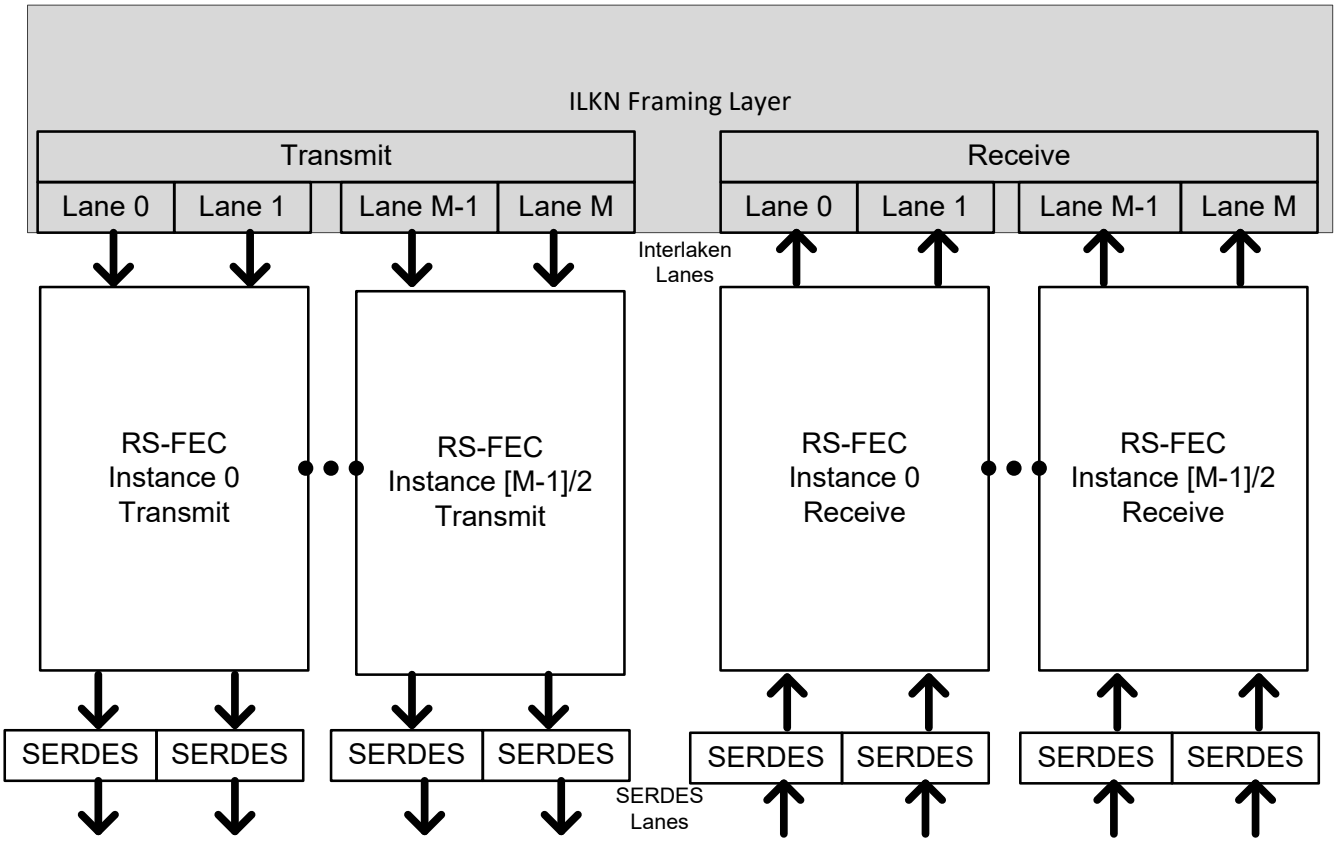
With the use of FEC it would be challenging to add another coding step to bound running disparity to very tight limits. So the goal of this proposal is to achieve running disparity that is no worse than that achieved by Ethernet protocols running on similar channels at similar rates.

3 RS-FEC Extension

The goal of this extension is to maintain the existing Interlaken protocol where possible and choose an appropriate solution to the new challenges of high-speed PAM4 SERDES that leverages existing solutions.

An Interlaken interface with RS-FEC using 36-58Gbps SERDES lanes consists of an even number of Interlaken lanes and an equal number of physical SERDES lanes. An interface will have one FEC instance for every 2 adjacent lanes from lane 0 to lane M. An Interlaken interface with the added FEC instances using 36-58Gbps SERDES lanes is shown in Figure 2a.

Figure 2a: Overview of Interlaken Interface with RS-FEC (36-58Gbps SERDES lanes)



An Interlaken interface with RS-FEC using 72-116Gbps SERDES lanes consists of an even number of Interlaken lanes and half the number of physical SERDES lanes. Each FEC instance is associated with a single 72-116Gbps SERDES lane. An Interlaken interface with the added FEC instances using 72-116Gbps SERDES lanes is shown in Figure 2b.

Figure 2b: Overview of Interlaken Interface with RS-FEC (72-116Gbps SERDES lanes)

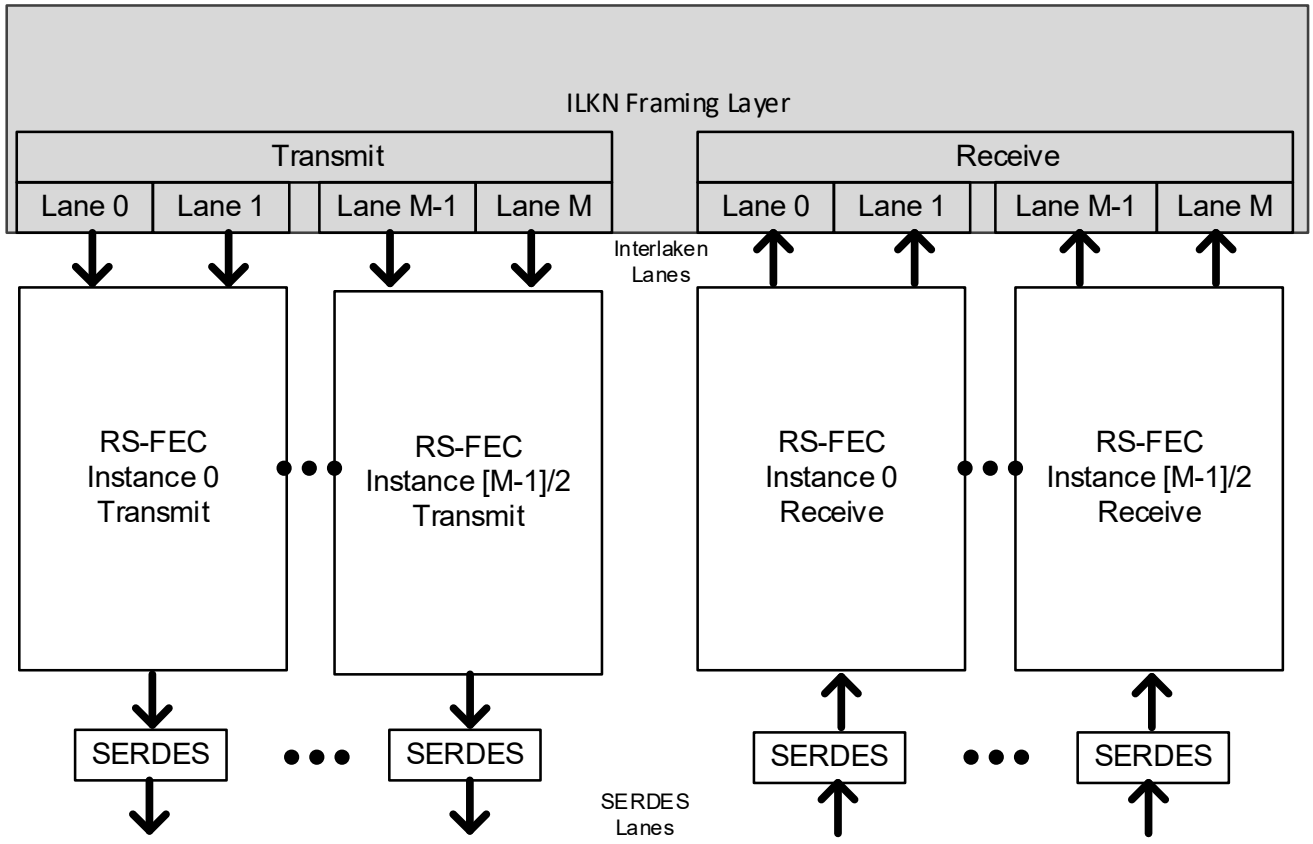


Figure 3a illustrates the extended functions (a single instance of the RS-FEC extension) for a pair of Interlaken Lanes that are to be carried on a pair of 36-58Gbps PAM4 SERDES lanes.

Figure 3b illustrates the extended functions (a single instance of the RS-FEC extension) for a pair of Interlaken Lanes that are to be carried on a single 72-116Gbps PAM4 SERDES lane.

Figure 3a: Detailed FEC Slice Functional Block Diagram (36-58Gbps SERDES lanes)

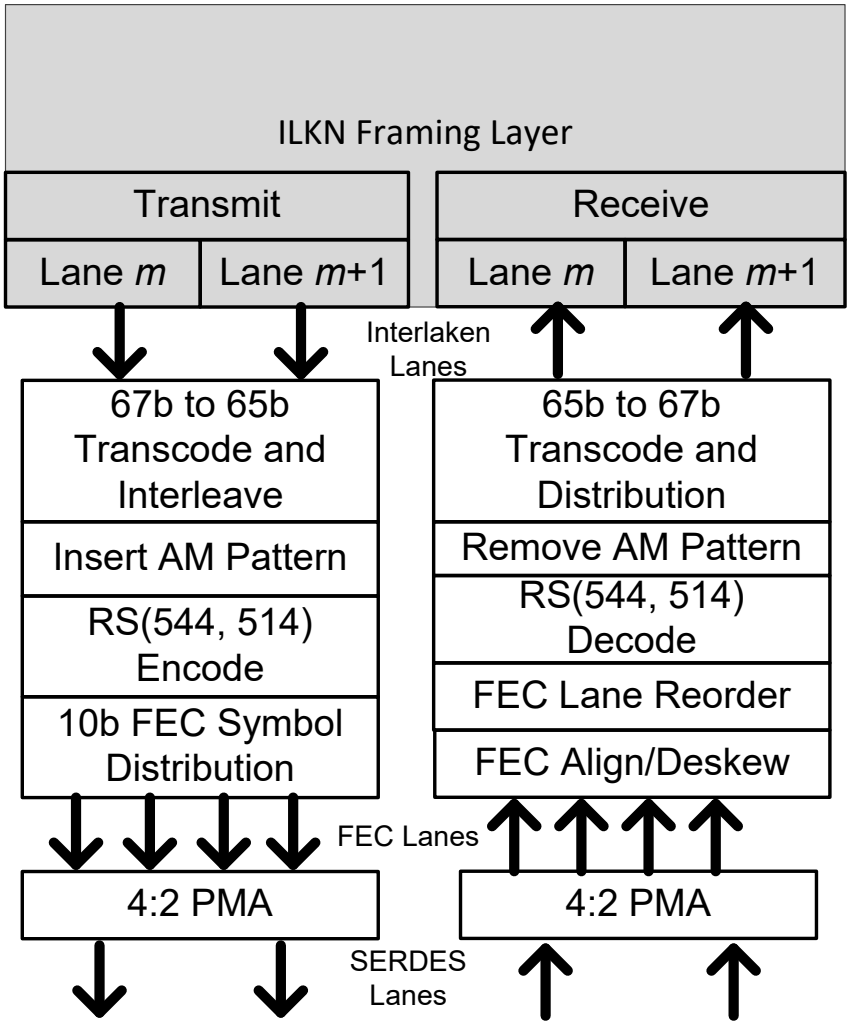
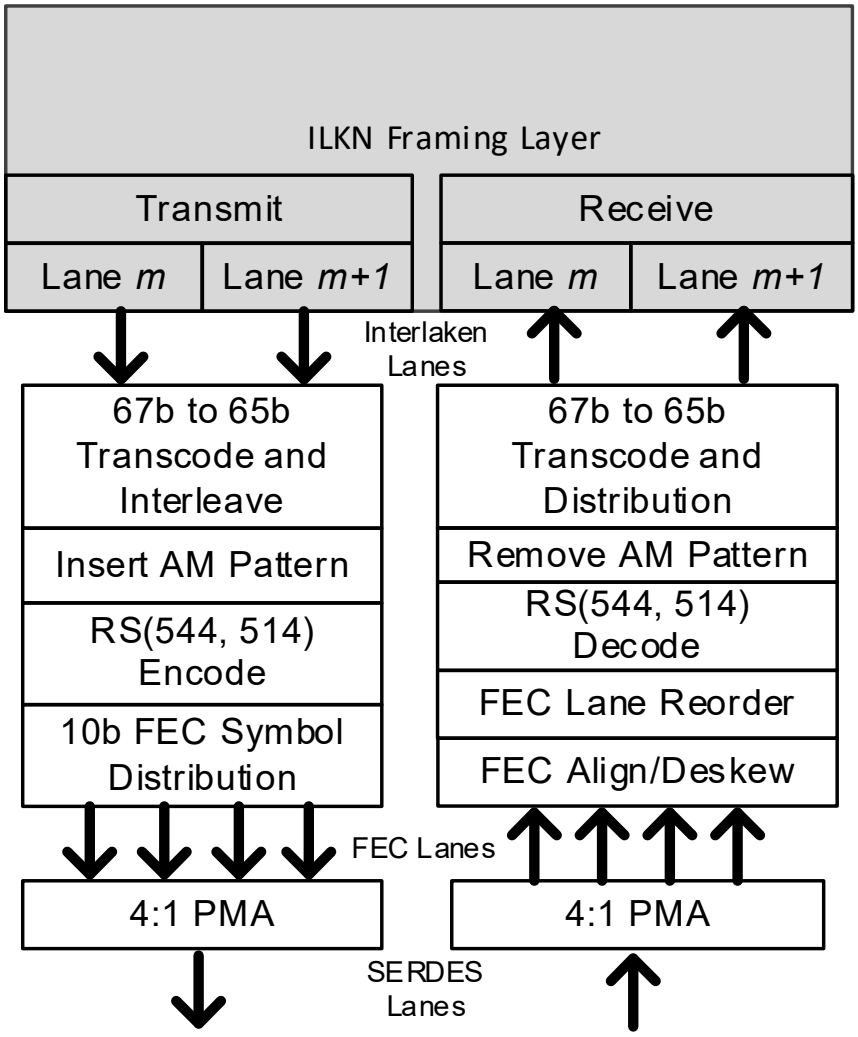


Figure 3b: Detailed FEC Slice Functional Block Diagram (72-116Gbps SERDES lanes)



The Interlaken protocol Framing Layer interface supports a number of lanes each carrying 67b line-coded blocks. A pair of Interlaken Lanes will be bound together for the purpose of forward error correction and transmission. For a group of M+1 Interlaken lanes indexed 0 through M there will be N+1= (M+1)/2 instances of the RS-FEC extension indexed 0 through N. The Framing Layer must use an even number of Interlaken lanes.

Interlaken Lane pairs are transcoded from 67b to 65b, interleaved, and inserted into an Alignment Marker (AM) delineated FEC frame. The FEC frame is encoded, and FEC symbols are distributed round-robin to 4 logical FEC lanes. The 4 FEC lanes are 4:2 bit muxed into two 36-58Gbps physical lanes, or 4:1 bit muxed into one 72-116Gbps physical lane. Physical lanes are gray-coded and then optionally precoded prior to transmission by the PAM4 SERDES.

In the receive direction the appropriate inverse functions occur as optional precode decoding, gray-decoding, and either 2:4 or 1:4 bit muxing to form 4 logical FEC lanes. Alignment of the FEC lanes is found so they can be deskewed, reordered, and FEC symbol-multiplexed prior to FEC decoding. After error corrections the 65b blocks are extracted from the AM-delineated FEC frame, transcoded to 67b, and distributed to the Interlaken Lanes for processing by the Interlaken Framing Layer.

3.1 Transmit Functions

Many functions are based on IEEE Std 802.3™-2018 functions. In all functions optional EEE deep sleep capability is not supported.

3.1.1 67b to 65b Transcode and Interleave

67b coded Interlaken words have 4 valid encodings of the Sync Bits (bits 66:64 of the word). These bits are to be mapped to a single Control bit as show in Table 1.

Table 1: Sync Bits to Control Bit Mapping

Bits [66:64]	Meaning	Control Bit
001	Data Word, No Inversion	0
010	Control Word, No Inversion	1
101	Data Word, bits [63:0] Inverted	0
110	Control Word, bits [63:0] Inverted	1
All others	Illegal states	NA

To limit baseline wander the control bit must be scrambled. This is achieved by XOR of the control bit with another bit in the word. Since the other bits are scrambled, they will appear to be random relative to the control bit, and by XOR with the control bit a new random sequence is created.

Since PAM4 transmits 2 adjacent bits in the same symbol, and the first bit controls the sign of the transmitted signal, it is best to choose a bit which is non-adjacent and an odd number of bits away from the control bit in the transmitted bit stream.

Therefore, XOR the calculated Control bit with bit [55] of the input word after removing inversion.

The overall transcode function from 67b to 65b is achieved by the following process (or its logical equivalent) where tx_data[66:0] is the 67b word-aligned Interlaken word from one lane of the framing layer.

For bit *I* in 0 to 63:

tx_data_transcoded[*I*] = tx_data[66] XOR tx_data[*I*]

tx_data_transcoded[64] = tx_data[65] XOR tx_data_transcoded[55]

This is done for all *m*=0..M lanes so that tx_data_transcoded[*m*][64:0] is the tx_data_transcoded[64:0] of lane *m*.

Each of the N+1 pairs of lanes is then interleaved on 65b block boundaries as:

For *n*=0..N

tx_data_interleaved[*n*][129:0]={ tx_data_transcoded[2*n][64:0],
tx_data_transcoded[2*n+1][64:0]}

3.1.2 Alignment Marker Generation and Insertion

Alignment markers are inserted for the purpose of FEC lane delineation and deskew, however they are inserted prior to FEC encoding and distribution so that they can be included in the FEC parity generation.

To facilitate reuse of RS(544, 514) FEC defined in IEEE Std 802.3™-2018 a similar alignment marker generation and insertion process is used. The alignment marker generation function creates a set of 20x64-bit alignment markers plus 5 bits of padding that are truncated to a total of 1240 bits. The last 4 alignment markers are ignored by the alignment process at the receiver so truncation is acceptable.

A 1240-bit truncated alignment marker group am_mapped_trunc[1239:0] is logically created by generating am_mapped<1284:0> using the process described in IEEE Std 802.3™-2018 clause 91.5.2.6 with these modifications:

1. Do not set y=16 when x>=16. Instead, always set y=x when x>3.
2. the variable bytes BIP3 and BIP7 shall instead use fixed unique padding fields derived from a PRBS sequence. These values are not checked at the receiver. The value in the BIP7 position is the inverse of the value in the BIP3 position to maintain DC balance. The values are unique within an alignment marker group, but the same pattern shall be used in every alignment marker group.
3. truncate the last 45 bits of am_mapped<1284:0> so that am_mapped_trunc[1239:0] is the same as am_mapped<1239:0>.

am_mapped_trunc[1239:0] is inserted so that it appears after every 161 940 x 130-bit blocks of each tx_data_interleaved[*n*]. As described in IEEE Std 802.3™-2018, this corresponds to the data field of 4096 RS-FEC codewords. The truncated alignment marker group, am_mapped_trunc, is transmitted as the first 1240 data bits of every 4096th codeword.

The truncated alignment marker group is to be inserted simultaneously on all tx_data_interleaved[*n*][129:0] so that transmitted alignment markers and FEC parity bits will appear at the same time nominally, on all lanes across the full interface.

Figure 4 shows the AM insertion and repetition rate. Figure 5 illustrates the bit ordering within the FEC codewords of the alignment marker (AM field) and the first 130-bit aligned data block that follows the AM field. The AM field is am_mapped_trunc[1239:0] where am_mapped_trunc[0] appears in bit column 0 and so on for am_mapped_trunc[1] through am_mapped_trunc[1239]. This is followed by the first tx_data_interleaved[*n*] in columns 1240 to 1369 with tx_data_interleaved[*n*][129] in column 1240 and so on for tx_data_interleaved[*n*][128] through tx_data_interleaved[*n*][0]. In this way the relative bit ordering remains the same as an Interlaken interface without FEC.

Figure 4: Alignment Marker Insertion Period

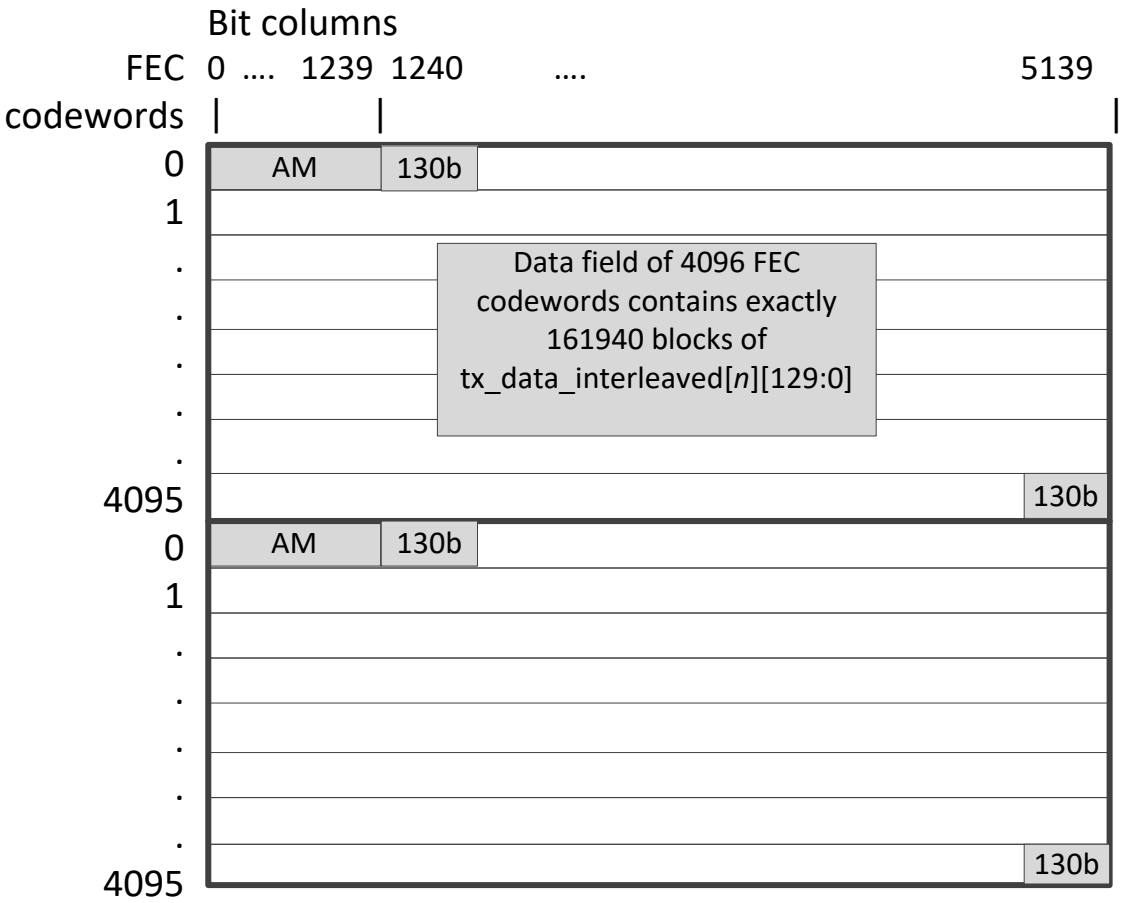
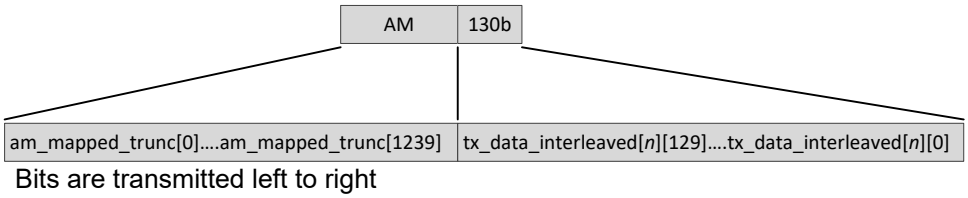


Figure 5: Alignment Marker and Data Bit Ordering

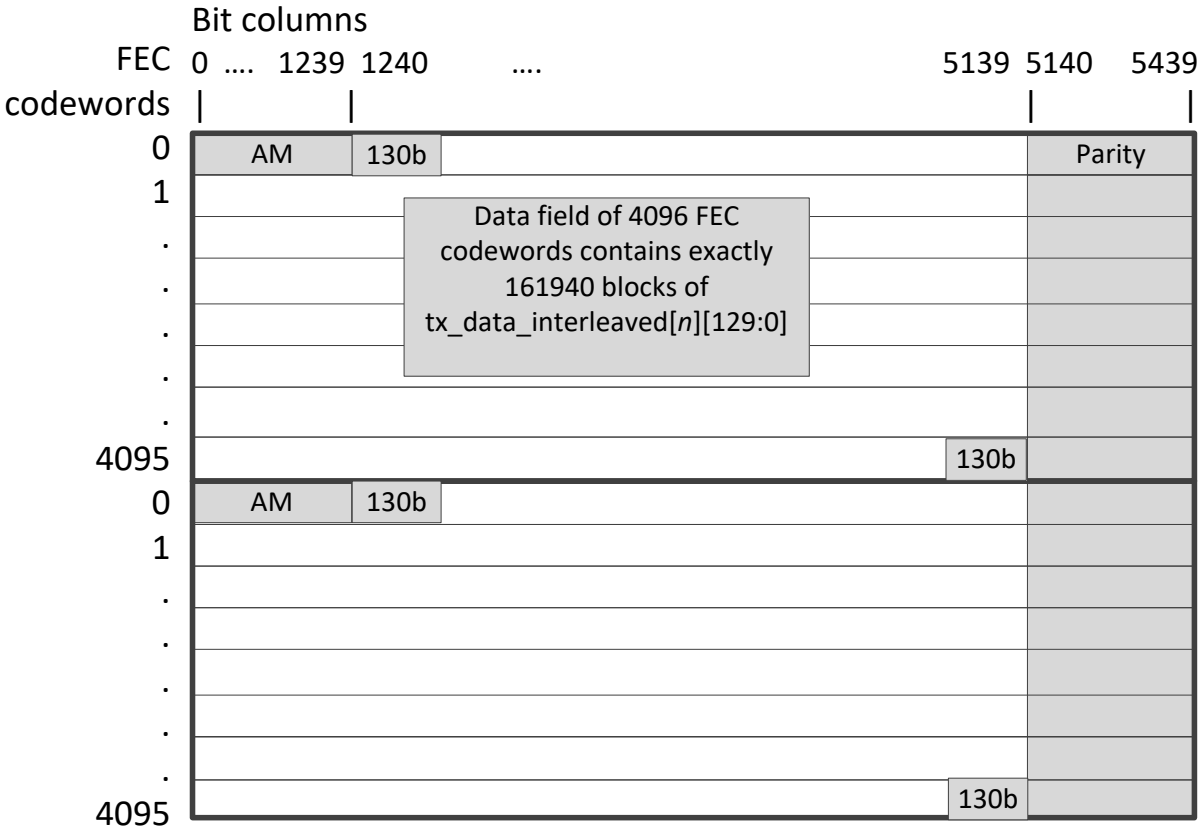


3.1.3 RS(544, 514) Encoder

The RS(544, 514) encoder is the same as that defined in IEEE Std 802.3™-2018 clause 91.5.2.7.

Figure 6 shows the FEC codeword sequence with parity bits added, prior to symbol distribution.

Figure 6: FEC Codewords



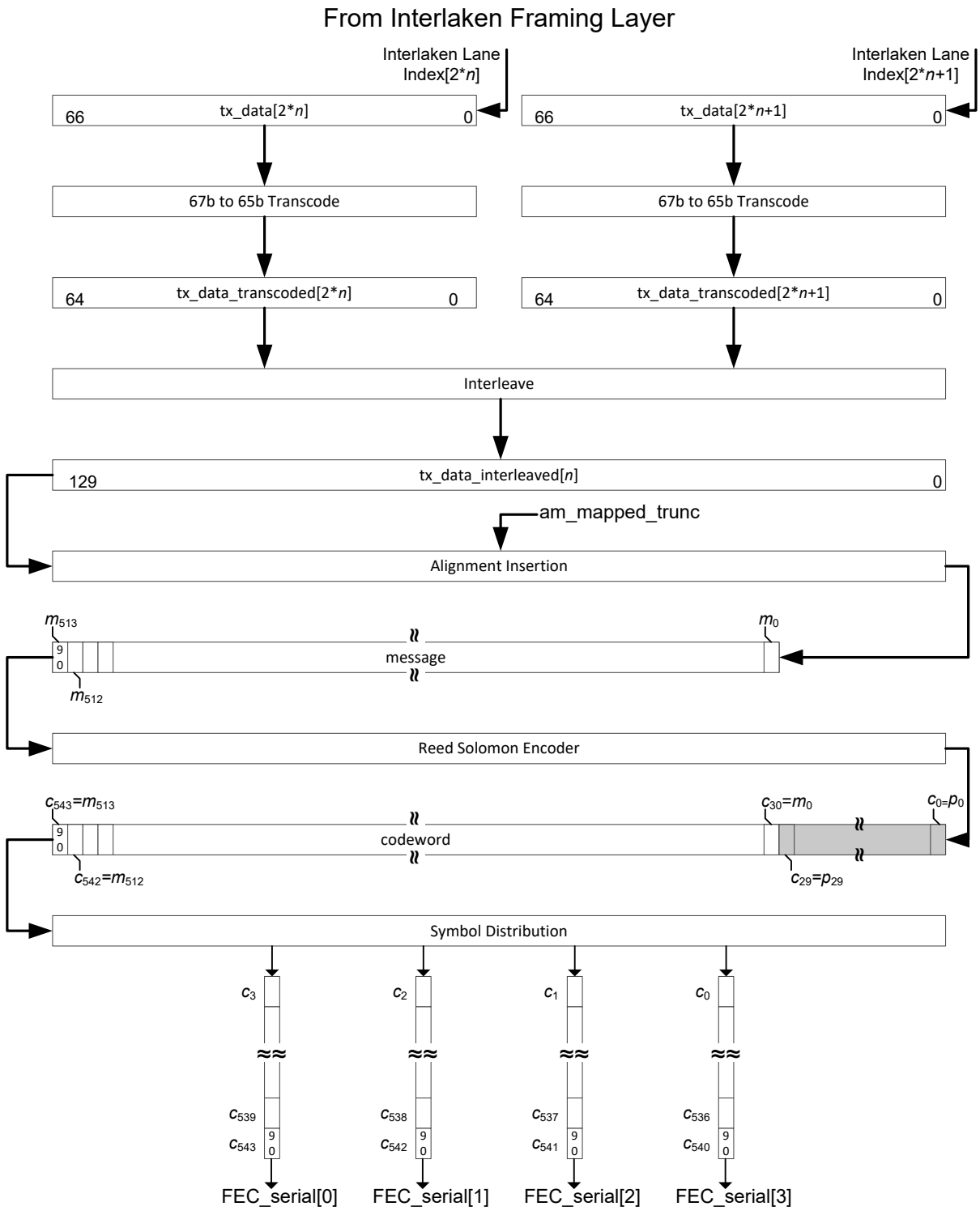
3.1.4 Symbol Distribution

The FEC symbol distribution process is the same as that defined in IEEE Std 802.3™-2018 clause 91.5.2.8. Transmit bit ordering will be the same as that defined in IEEE Std 802.3™-2018 clause 91.5.2.9. At the output of the symbol distribution there are 4 logical serial streams, or FEC lanes, that are labelled FEC_serial[0:3].

3.1.5 Transmit bit ordering

Figure 7 shows the transmit bit ordering.

Figure 7: Transmit Bit Ordering



3.1.6 4:2 Transmit PMA

The 4:2 Transmit PMA is based on the IEEE Std 802.3™ PMA. The functions performed in order in the transmit direction consist of:

1. 4:2 Bit-muxing
2. Gray coding
3. Optional precoding
4. PAM4 encoding

The transmit PMA also contains the transmit SERDES functions which are implementation specific.

3.1.6.1 4:2 Bit-muxing

FEC_serial_interleaved[0] is formed by bit-muxing FEC_serial[0] and FEC_serial[1] by alternately transmitting bits from each.

FEC_serial_interleaved[1] is formed by bit-muxing FEC_serial[2] and FEC_serial[3] by alternately transmitting bits from each.

3.1.6.2 Gray Coding

Gray coding maps bit pairs to Gray-coded symbols. Each FEC_serial_interleaved is Gray-coded according to the mapping specified in the first 2 paragraphs of IEEE Std 802.3™-2018 clause 94.2.2.5. The portions of the clause dependent on termination blocks are not applicable in this case.

This process produces Graycoder_output[1:0] which are 2 separate Gray-coded symbol streams from the FEC_serial_interleaved[1:0] inputs, respectively.

3.1.6.3 Optional precoding

If precoding is enabled, it is done on the two Gray-coded symbols streams independently. The precoding process is based on IEEE Std 802.3™-2018 clause 94.2.2.6, but without termination blocks so the precoding process for generating precoded symbols from Gray-coded symbols is:

Precoder_output[x](t) = (Graycoder_output[x](t) – Precoder_output[x](t-1)) MOD 4

3.1.6.4 PAM4 encoding

PAM4 coding shall be performed on the Gray-coded, or optionally on the precoded symbol streams per the mapping described in IEEE Std 802.3™-2018 clause 94.2.2.7.

3.1.7 4:1 Transmit PMA

The 4:1 Transmit PMA is based on the IEEE Std 802.3™ PMA. The functions performed in order in the transmit direction consist of:

1. 4:1 Bit-muxing
2. Gray coding
3. Optional precoding
4. PAM4 encoding

The transmit PMA also contains the transmit SERDES functions which are implementation specific.

3.1.7.1 4:1 Bit-muxing

FEC_serial_interleaved is formed by bit-muxing FEC_serial[0], FEC_serial[1], FEC_serial[2] and FEC_serial[3] in order, by transmitting one bit at a time from each of the 4 logical serial streams (FEC lanes).

3.1.7.2 Gray Coding

Gray coding maps bit pairs to Gray-coded symbols. Each FEC_serial_interleaved is Gray-coded according to the mapping specified in the first 2 paragraphs of IEEE Std 802.3™-2018 clause 94.2.2.5. The portions of the clause dependent on termination blocks are not applicable in this case.

This process produces Graycoder_output, a Gray-coded symbol stream from the FEC_serial_interleaved input.

3.1.7.3 Optional precoding

Precoding is optional. The precoding process is based on IEEE Std 802.3™-2018 clause 94.2.2.6, but without termination blocks so the precoding process for generating precoded symbols from Gray-coded symbols is:

Precoder_output(t) = (Graycoder_output(t) – Precoder_output(t-1)) MOD 4

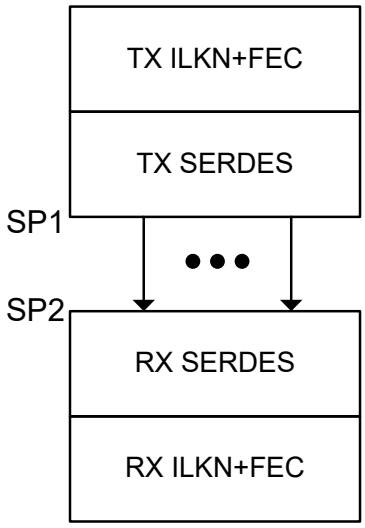
3.1.7.4 PAM4 encoding

PAM4 coding shall be performed on the Gray-coded, or optionally on the precoded symbol streams per the mapping described in IEEE Std 802.3™-2018 clause 94.2.2.7.

3.1.8 Transmit Skew limits

Transmit skew of the alignment marker patterns across all external PAM4 lane interfaces shall be less than 10ns. The measurement point corresponds to the transmit balls/pins on the device package and is specified at SP1 as shown in Figure 8.

Figure 8: Skew Points



3.2 Receive Functions

Many functions are based on IEEE Std 802.3™-2018 functions. In all functions optional EEE deep sleep capability is not supported.

3.2.1 Receive skew limits

Receive skew of the alignment marker patterns across all external PAM4 lane interfaces shall be less than 20ns. The receive skew is specified at SP2 as shown in Figure 8.

Receive skew variation shall be less than 0.2ns at SP2.

3.2.2 4:2 Receive PMA

The functions performed in order in the receive direction consist of:

1. PAM4 decoding
2. Optional precoder decoding
3. Inverse Gray coding
4. 2:4 Bit-demuxing

The receive PMA also contains the receive SERDES functions which are implementation specific.

PAM4 decoding is the reverse mapping of the PAM4 encoding process.

The precode decoding is only performed if precoding is used in the transmit direction. The precoder decoding process produces the Gray-coded symbols stream at its output:

Precode_decoder_output[x](t) = (PAM4_decoder_output[x](t) + PAM4_decoder_output[x](t-1)) MOD 4

Inverse Gray-coding is the reverse mapping of the Gray-coding process and converts each of the 2 Gray-coded symbol streams to a logical bit stream.

For each received bit stream the 2:4 Bit-demuxing distributes every second bit to one of 2 FEC receive lanes. In this way 2 bit streams are converted to 4 FEC lanes. The lane ordering of the 4 received FEC lanes after bit-demuxing is non-deterministic.

3.2.3 4:1 Receive PMA

The functions performed in order in the receive direction consist of:

5. PAM4 decoding
6. Optional precoder decoding
7. Inverse Gray coding
8. 1:4 Bit-demuxing

The receive PMA also contains the receive SERDES functions which are implementation specific.

PAM4 decoding is the reverse mapping of the PAM4 encoding process.

The precode decoding is only performed if precoding is used in the transmit direction. The precoder decoding process produces the Gray-coded symbols stream at its output:

Precode_decoder_output(t) = (PAM4_decoder_output(t) + PAM4_decoder_output (t-1)) MOD 4

Inverse Gray-coding is the reverse mapping of the Gray-coding process and converts each of the 2 Gray-coded symbol streams to a logical bit stream.

For each received bit stream the 1:4 Bit-demuxing distributes every fourth bit to one of 4 FEC receive lanes. In this way the received 72-116Gbps bit stream is converted to 4 FEC lanes. The lane ordering of the 4 received FEC lanes after bit-demuxing is non-deterministic.

3.2.4 FEC Align and Deskew

The FEC lane alignment and deskew processes are the same as that defined in IEEE Std 802.3™-2018 clause 91.5.3.1. The EEE-related functions are not supported and can be ignored. Since Interlaken is used with relatively short interconnects the skew limits can be reduced. The FEC receive deskew function shall support a maximum skew of at least 1060UI at SP2 as shown in Figure 8 between 4 FEC lanes carrying parts of the same FEC codewords. As well, the maximum supported skew between all FEC lanes belonging to the same Interlaken interface shall be at least 1060UI at SP2.

The maximum supported skew variation is at least 10UI at SP2.

For received data where the FEC alignment state is such that "fec_align_status" is false, the Receive Per-Lane State of the affected Interlaken lanes shall be equivalent to the behavior when the receive SERDES loses lock in the Interlaken Protocol Definition Figure 8 (CDR LOCK? = No) and section 5.4.11.1.

3.2.5 FEC Lane Reorder

The Lane Reorder process per 4 FEC lanes is the same as that defined in IEEE Std 802.3™-2018 clause 91.5.3.2. Reordering among all FEC lanes of the Interlaken interface or among Interlaken lanes is not performed in this function.

3.2.6 RS(544, 514) Decoder

The RS(544, 514) decoder is as defined in IEEE Std 802.3™-2018 clause 91.5.3.3 with the following changes:

1. The portions related to indicating errors to the PCS sublayer are not applicable since they are specific to the 66b coding used in Ethernet. Instead, uncorrectable (or uncorrected) errors shall cause the Interlaken receiver to error affected open packets on all channels. Any packet which has a part carried in an errored FEC codeword must be dropped. The behavior is similar to the behavior defined for a Burst CRC24 Error as detailed in section 5.4.11.5 of the Interlaken Protocol Definition.
2. Error indication bypass ability is not provided.
3. Correction bypass option is not provided.

3.2.7 Remove AM Pattern

Similar to IEEE Std 802.3™-2018 clause 91.5.3.4 the specific codeword that includes the 1240-bit AM vector shall be indicated by the Align and Deskew functions.

The 1240-bit AM pattern shall be removed and the data following will be output in 130-bit aligned units to produce a series of 130-bit words rx_data_interleaved[*n*][129:0] for each *n*=0..N.

3.2.8 65b to 67b Transcode and Distribution

The distribution to Interlaken lanes is performed by splitting each sequence of 130-bit words into two sequences of 65-bit words:

rx_data_transcoded[2*n*][64:0] = rx_data_interleaved[*n*][129:65]

rx_data_transcoded[2*n*+1][64:0]} = rx_data_interleaved[*n*][64:0]

67b codewords are generated from 65b codewords by descrambling the control bit, generating the 2-bit sync-header and only using non-inverted 67b words for each lane *m*=0..M:

rx_data[*m*][66:0] = { 0,

rx_data_transcoded[*m*][64] XOR rx_data_transcoded[*m*][55],

rx_data_transcoded[*m*][64] XNOR rx_data_transcoded[*m*][55],

rx_data_transcoded[*m*][63:0]}

At this point the rx_data[*m*] is in a form that can be directly connected to the Interlaken Framing Layer receive interface lane *m*.

3.2.9 Receive Interlaken Lane skew tolerance

The skew between Interlaken lanes which were recovered from the same FEC codeword will be negligible, but skew between Interlaken lanes recovered from different FEC codewords may be larger. Interlaken lanes shall be deskewed using the Interlaken Protocol Definition section 5.4.5 Lane Alignment Process. The skew remaining for this process to remove after the parallel FEC termination is implementation specific.

3.3 Use of Diagnostic Word

The per-lane diagnostic word CRC32 count is no longer traceable to a single physical lane, and FEC statistics are more meaningful. Therefore, when using FEC, all-0's may be inserted into the CRC32 field of the diagnostic word at the Interlaken Transmit and the CRC32 field must be ignored at the Interlaken Receive interface.

3.4 Status Reporting

The following per-FEC instance status variables from Table 91-3 of IEEE Std 802.3™-2018 are recommended to be reported:

- amps_lock<*x*>
  - Per-lane alignment state machine lock status for the alignment marker payload sequence of lane *x* (*x*=0:3).
- fec_align_status
  - Per-FEC instance indication that deskew process is complete and all lanes are aligned and FEC processing can be carried out.
- FEC_lane_mapping<*x*>
  - Per-lane alignment state machine FEC lane number of lane *x* (*x*=0:3).
- FEC_corrected_cw_counter
  - Per-FEC instance count of codewords that contained errors and were corrected.
- FEC_uncorrected_cw_counter
  - Per-FEC instance count of codewords that contained errors and were not corrected.
- FEC_symbol_error_counter_*i*
  - Per-FEC lane *i* (*i*=0:3) count of 10-bit symbol errors corrected on FEC lane *i*.

4 Appendix

4.1 Overhead Comparison

In general overhead can be calculated by dividing the number of overhead bits over the number of useful bits sent. For Interlaken the useful bits can be considered the 64 bits of each data or control word.

The line coding overhead for an Interlaken interface using 67b coding without the RS FEC extension consists of 3 sync bits per 64b block of data. The line coding overhead is (67/64)-1 = 4.7%.

The line coding overhead for an Interlaken interface using the RS FEC extension is similarly calculated over a 4096 codeword RS FEC frame as (5440*4096/(323880*64))-1 = 7.5%.

The percent increase in line coding overhead for the RS FEC extension over 67b coding is then (5440*4096/(323880*67))-1 = 2.68%.

4.2 Alignment Marker Encodings

The purpose of the Alignment Marker Generation and Insertion described in 3.1.2 is to achieve a recognizable, periodic lane alignment and identification pattern on each FEC Lane. Table 2 illustrates the information used to form the alignment markers. Note each octet is transmitted LSB to MSB and only 6 of the 8 bits for the final M_6 value of each lane are included in the transmitted pattern. The values provided here for BIP_3 and BIP_7 are arbitrary and are not checked at the receiver, however the values provided are selected to provide DC balance and transition density including the effects of truncation.

Table 3 illustrates the pattern that will appear on each FEC Lane after Symbol Distribution to the 4 FEC_serial lanes.

Table 2: FEC Lane Alignment Marker Encodings

FEC Lane Number	Alignment marker index *#x*	Alignment marker *#y*	M_0	M_1	M_2	BIP_3	M_4	M_5	M_6	BIP_7
FEC_serial[0]	0	0	0xC1	0x68	0x21	0xD9	0x3E	0x97	0xDE	0x26
	4	4	0xF5	0x07	0x09	0x19	0x0A	0xF8	0xF6	0xE6
	8	8	0xA0	0x24	0x76	0x9F	0x5F	0xDB	0x89	0x60
	12	12	0x5C	0xB9	0xB2	0x5B	0xA3	0x46	0x4D	0xA4
	16	16	0xC4	0x31	0x4C	0xA6	0x3B	0xCE	0xB3	
FEC_serial[1]	1	0	0xC1	0x68	0x21	0x67	0x3E	0x97	0xDE	0x98
	5	5	0xD	0x14	0xC2	0x4E	0x22	0xEB	0x3D	0xB1

			D							
	9	9	0x68	0xC9	0xFB	0xA2	0x97	0x36	0x04	0x5D
	13	13	0x1A	0xF8	0xBD	0xCC	0xE5	0x07	0x42	0x33
	17	17	0xAD	0xD6	0xB7	0x74	0x52	0x29	0x48	
FEC_serial[2]	2	0	0xC1	0x68	0x21	0xFE	0x3E	0x97	0xDE	0x01
	6	6	0x9A	0x4A	0x26	0xEE	0x65	0xB5	0xD9	0x11
	10	10	0xFD	0x6C	0x99	0x04	0x02	0x93	0x66	0xFB
	14	14	0x83	0xC7	0xCA	0xB1	0x7C	0x38	0x35	0x4E
	18	18	0x5F	0x66	0x2A	0xCE	0xA0	0x99	0xD5	
FEC_serial[3]	3	0	0xC1	0x68	0x21	0x84	0x3E	0x97	0xDE	0x7B
	7	7	0x7B	0x45	0x66	0x32	0x84	0xBA	0x99	0xCD
	11	11	0xB9	0x91	0x55	0x71	0x46	0x6E	0xAA	0x8E
	15	15	0x35	0x36	0xCD	0x56	0xCA	0xC9	0x32	0xA9
	19	19	0xC0	0xF0	0xE5	0x4A	0x3F	0x0F	0x1A	

Table 3: FEC Lane Alignment Marker Encodings

FEC Lane	**310-bit Alignment Bit Pattern as Transmitted Left to Right**
FEC_serial[0]	10000011 00010110 … 11011100 01110011 110011
FEC_serial[1]	10000011 00010110 … 01001010 10010100 000100
FEC_serial[2]	10000011 00010110 … 00000101 10011001 101010
FEC_serial[3]	10000011 00010110 … 11111100 11110000 010110

References

1. Interlaken Alliance: Interlaken Protocol Definition Revision 1.2 Oct. 7, 2008.
2. IEEE Std 802.3™-2018, IEEE Standard for Ethernet, 14 June 2018.
3. Optical Interworking Forum Implementation Agreement OIF-CEI-04.0, 29 December 2017.