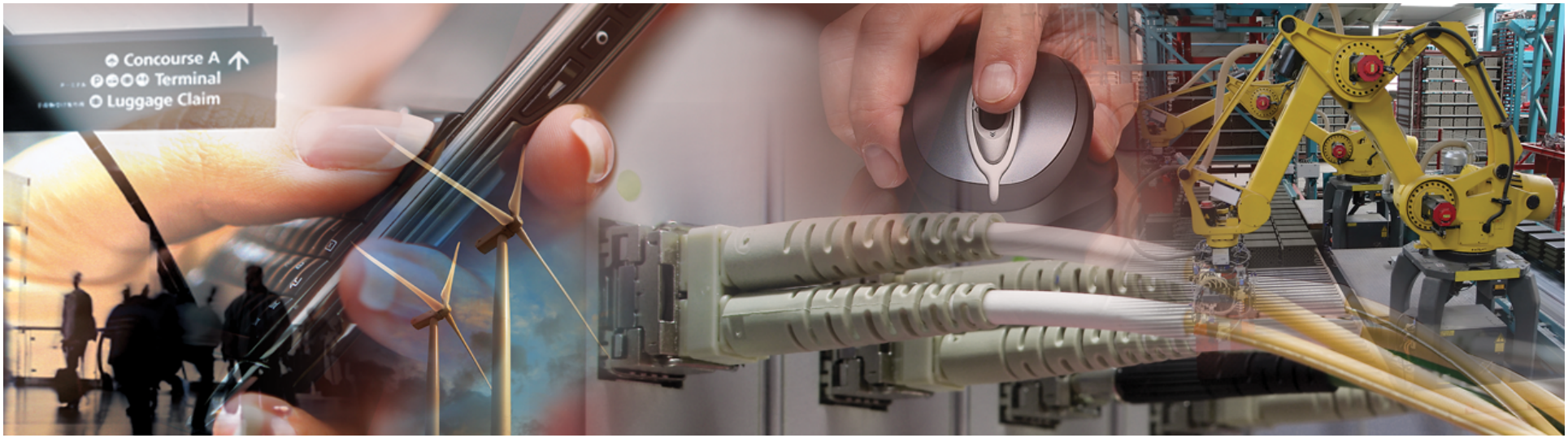
Support of comments #169 & #171



Jeff Slavick

AVAGO TECHNOLOGIES

Supporters

- Steve Trowbridge – ALU
- Anil Mehta – Brocade
- Venu Balasubramonian – Marvell
- Adrian Butter - IBM

Comments

- **#169 – EEE broken**
  - Sending out unscrambled data breaks DC balance among other things.
- **#171 – CWM existence**
  - CWM are not a required method for achieving FEC codeword lock and are more complicated approach then SnT (Search and Test) used by CI74.

Past presentations

1) OTN Support for 25GbE – Atlanta Interim
   http://www.ieee802.org/3/by/public/Jan15/trowbridge_3by_01_0115.pdf
2) Cost of CWMs – Berlin Plenary
   http://www.ieee802.org/3/by/public/Mar15/slavick_3by_01a_0315.pdf
3) CWMs in IEEE 25GE (802.3by) – April 16 ad-hoc
   http://www.ieee802.org/3/by/public/adhoc/architecture/baden_041515_25GE_adhoc-v2.pdf
4) Options for the RS-encoded 25GE – May 6 ad-hoc
   http://www.ieee802.org/3/by/public/adhoc/architecture/calderon_050615_25GE_adhoc.pdf
5) Support for CWM removal – May 13 ad-hoc
   http://www.ieee802.org/3/by/public/adhoc/architecture/slavick_051315_25GE_adhoc.pdf

Summary of conclusions from presentations

	With CWM	No CWM
OTN	Must de-transcode & rate compensate when using RS-FEC on entry/exit of OTN network, which includes descrambling of data stream.	Must de-transcode only when using RS-FEC. similar to 802.3bj
IEEE 1588	Packet delay variation, due to rate compensation being done below PCS, makes it difficult to optimally support this standard	Supportable
Flex-Ethernet	Breaks frame construction; RS-FEC operation only to minimize exceptions defined	Supportable for all 3 25G operating modes
EEE	constant pattern search (Rapid CWM); down count for tracking Rapid -> normal CWM interval; data always scramble	Constant pattern search; data always scrambled
32GFC	Same RS-FEC engine	Entire data path
Lock Time (<5ms requirement)	Mean: 300us WC: 800us	Mean: 500us WC: 3ms
Area of CR PHY	~500k gates	~480k gates

Conclusion – why remove CWMs from 25GE

- **Looks like a sped up 10GE PHY**
  - No FEC – just rate increase
  - Cl 74 FEC – just rate increase
  - Cl108 RS-FEC – looks like Cl74 FEC stream with larger FEC codeword
- **Simplify the specification**
  - Less complex logic being specified. (scrambler v. CWM insert & delete)
  - Fewer exceptions (ie. error marking)
- **Ease use of 25GE by other specifications**
  - OTN, FlexE, 1588 won't require exceptions when 25GE RS-FEC mode is being used due to rate compensation needed by CWMs

Draft changes to resolve #171

- **Delete 108.5.2.2, 108.5.2.4, 108.5.2.4, 108.5.3.6**
- **Remove total of 7 instances of "marker(s)" in 108.5.3.1 and 108.5.3.3**
- **Delete** "rx_coded_0<1:0> corresponding to the second 257-bit block and" **from 108.5.3.2**
- **Delete variables: cwm_counter_done, cwm_valid, test_cwm, cwm_counter, scrambler_bypass, descrambler_bypass**
- **Replace figures 108-2,3,4,5 with versions on slides 9-12**
- **Add** "and scrambled by the PN-5280" **after "encoded" in 108.5.2.6**
- **Add** "and descrambling them with a PN-5280 generator." **after "they are received" in 108.5.3.1**
- **Add definition for PN-5280 pseudo noise sequence generator between 108.5.2.5 and 108.5.2.6 (see slide 8)**
- **Add definition for parity_valid to 108.5.2.4 (see slide 8)**
- **Update definition for test_cw (see slide 8) (additional Figure reference)**

Draft changes to resolve #171 (cont)

108.5.2.5a PN-5280 pseudo-noise sequence generator

The PN-5280 pseudo-noise generator is identical the one defined in 74.7.4.4.1 with the exception that it's length is 5280.

Scrambling with the PN-5280 sequence at the RS-FEC codeword boundary is necessary for establishing RS-FEC block synchronization (to ensure that any shifted input bit sequence is not equal to another RSFEC codeword) and to ensure DC balance.

108.5.2.4 State Variables

parity_valid

Boolean variable that is set to true if the parity received in the FEC codeword matches the calculated parity and is set to FALSE otherwise.

test_cw

Boolean variable that is set to true when a new FEC codeword is available for decoding and is set to false according to the state diagrams in Figure 108-5 and Figure 108–6.

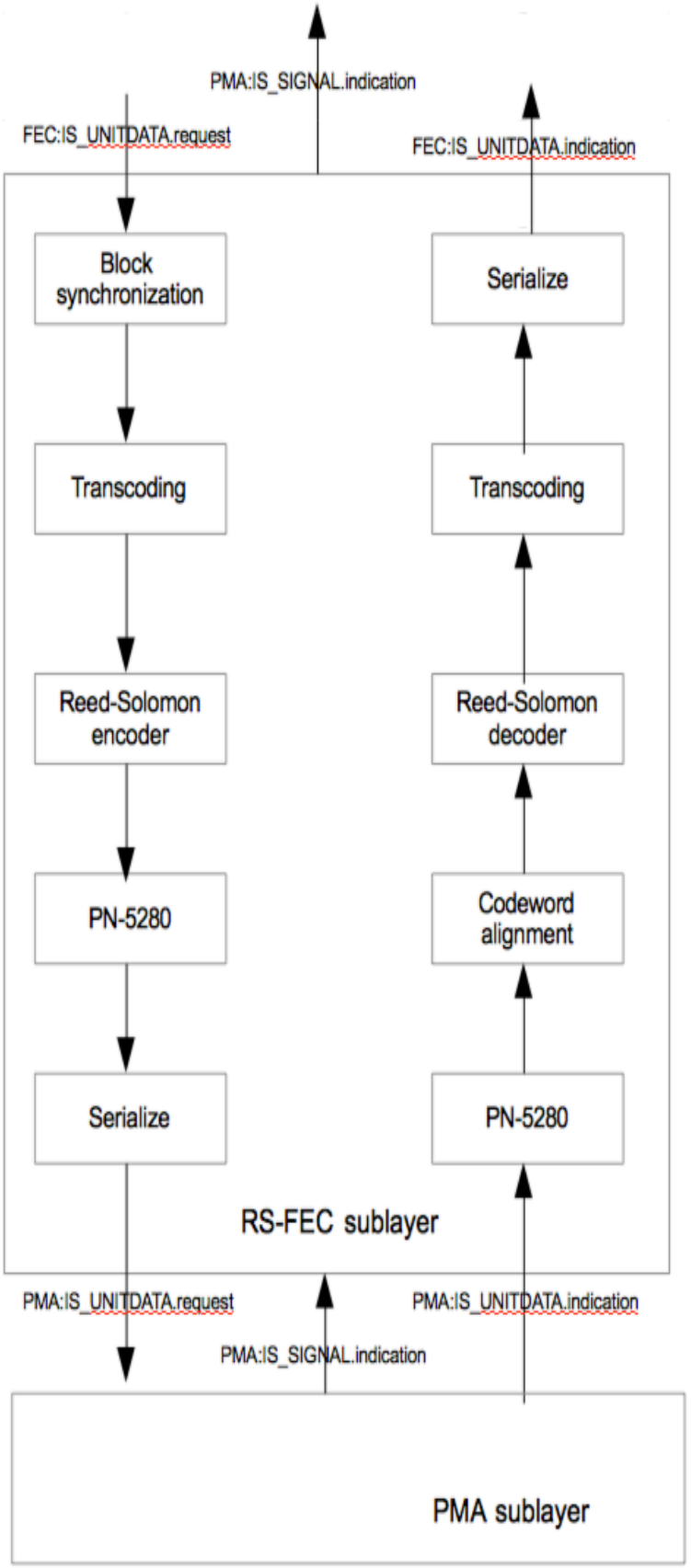


Figure 108-2 Functional Block Diagram

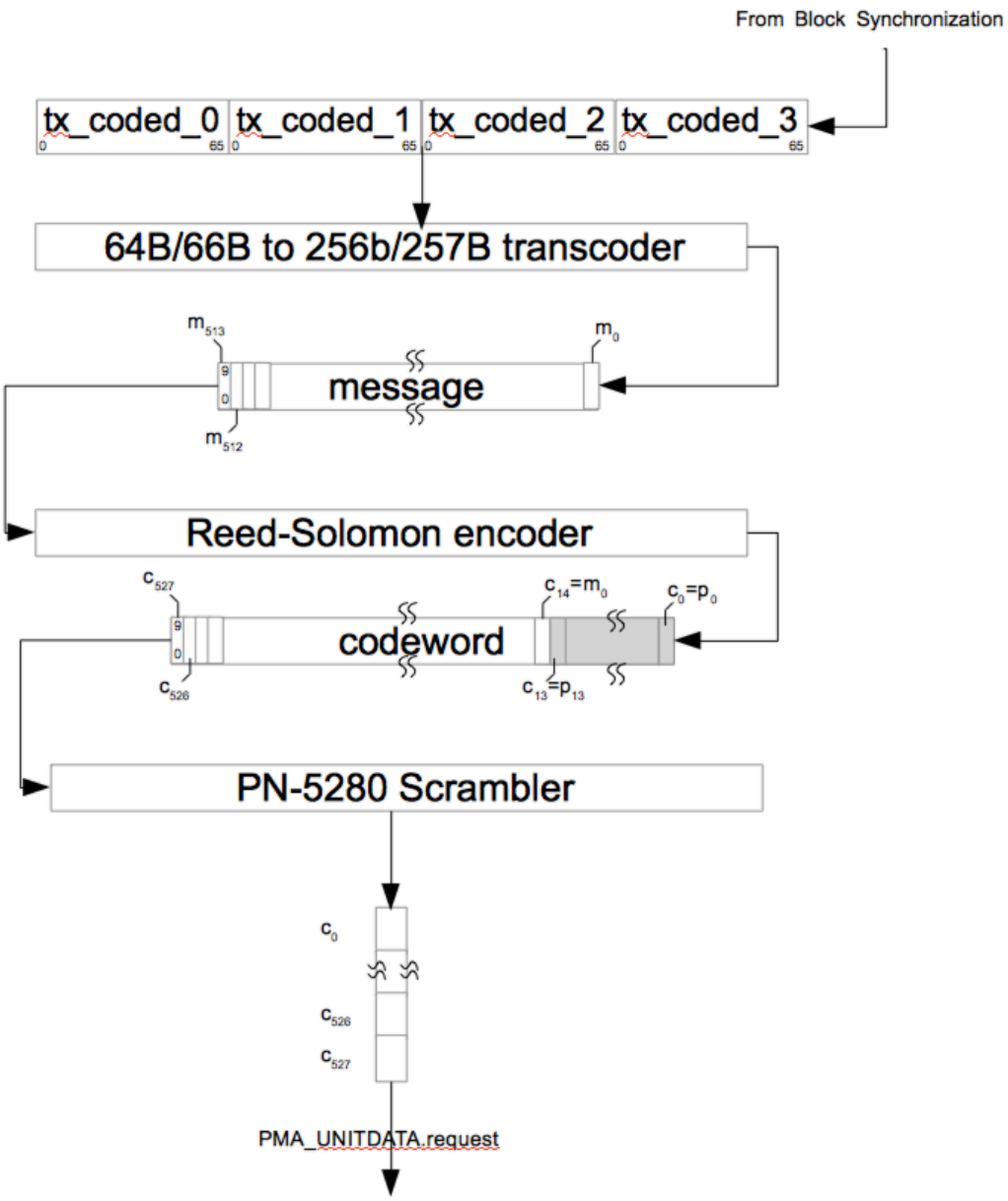


Figure 108-3 Transmit bit ordering

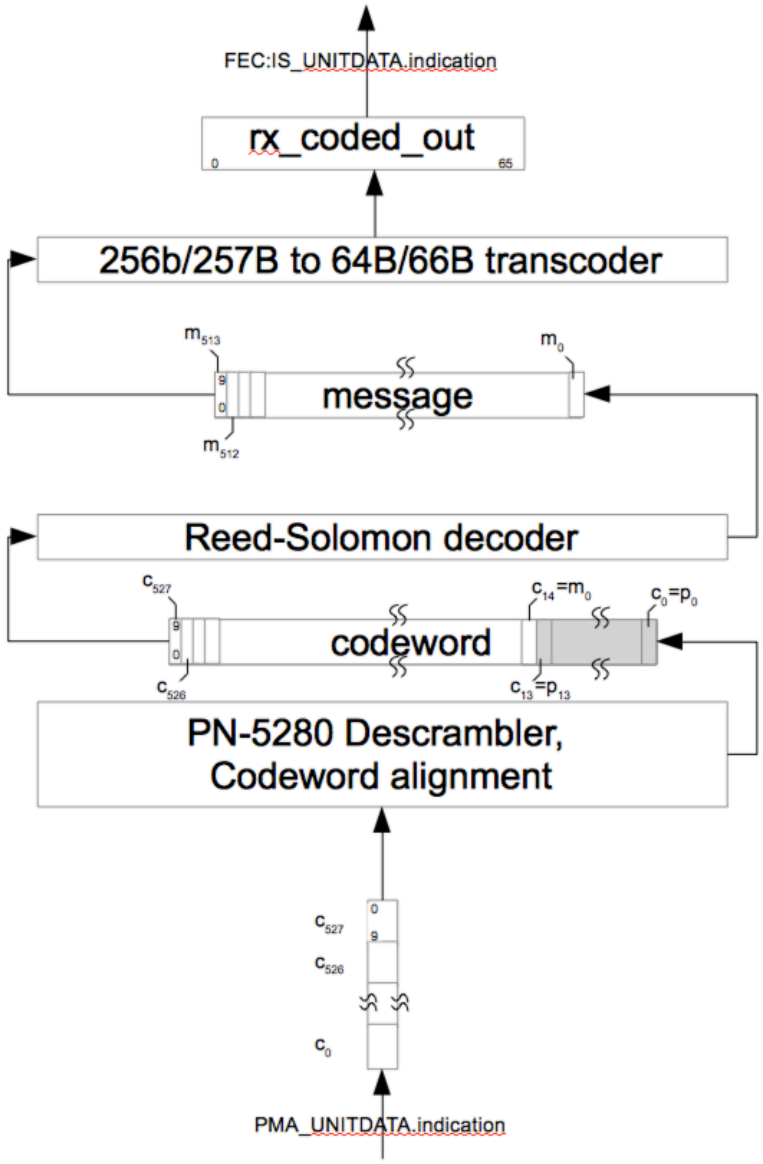


Figure 108-4 Receive bit ordering

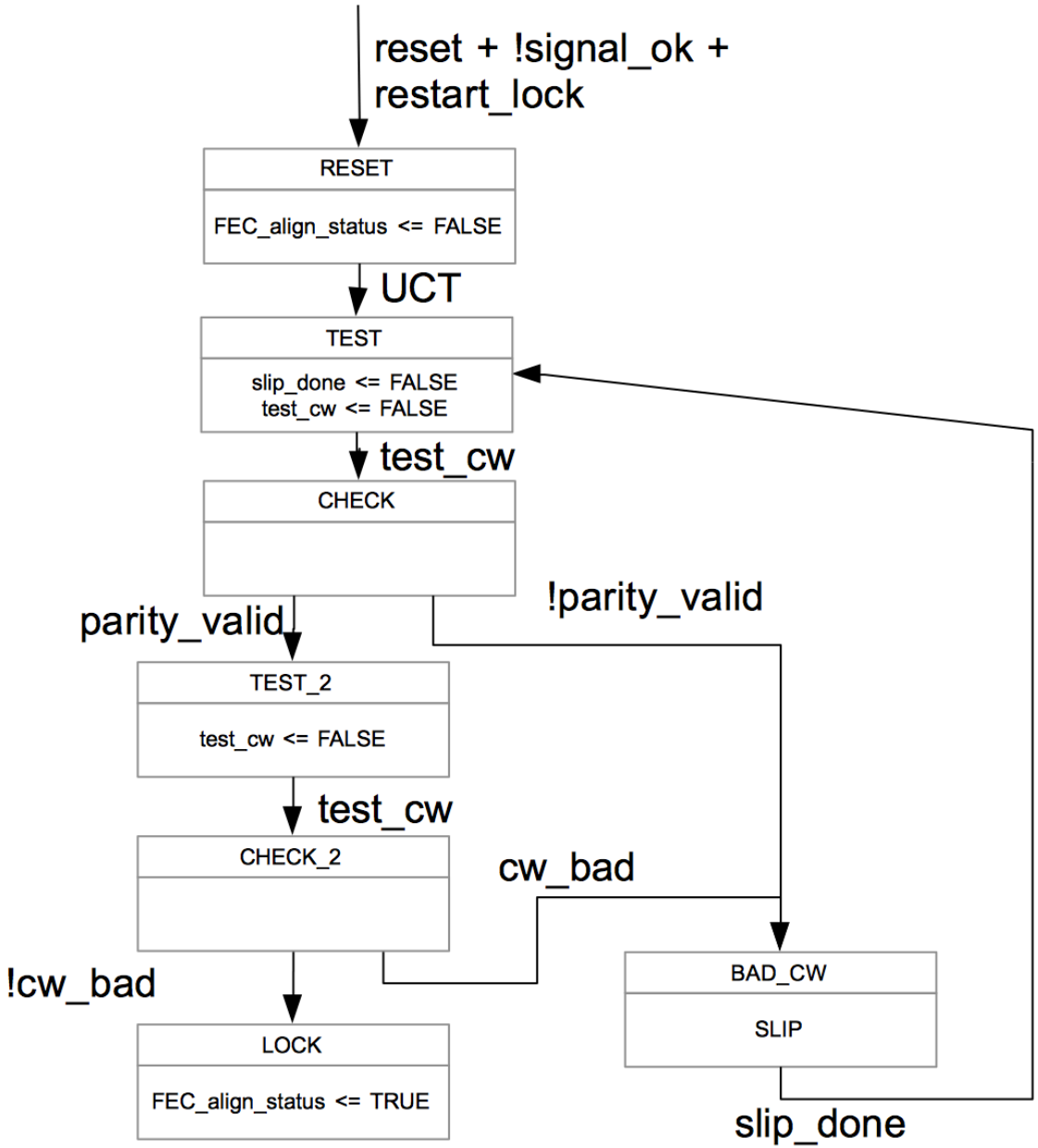


Figure 108-5 FEC synchronization state diagram

Draft Changes to resolve #169

- **In 108.5.2.7 replace all text after a) with**

a) Transmit data per normal for a period of 1.1us to 1.3us

b) Descramble the 66b blocks provided (see 49.2.10) to the transcoder for a period of 1.1us to 1.3us

c) Return to normal data transmission

As a result of transmitting descrambled 66b blocks at least 4 RS-FEC codewords with known data patterns are produced at the output of the PN-5280 pseudo-noise generator. The known data patterns can then be used by the receive process to achieve a rapid codeword lock.

Draft Changes to resolve #169 (cont)

- **In 108.5.3.7**
  - remove a)
  - Remove all text after “CW_GOOD state” on page 109 line 41 to line 46
  - Change 1) to read “Two 64B/66B after the transition from deterministic FEC block to normal scrambled FEC block”

Backup

Entry to OTN network

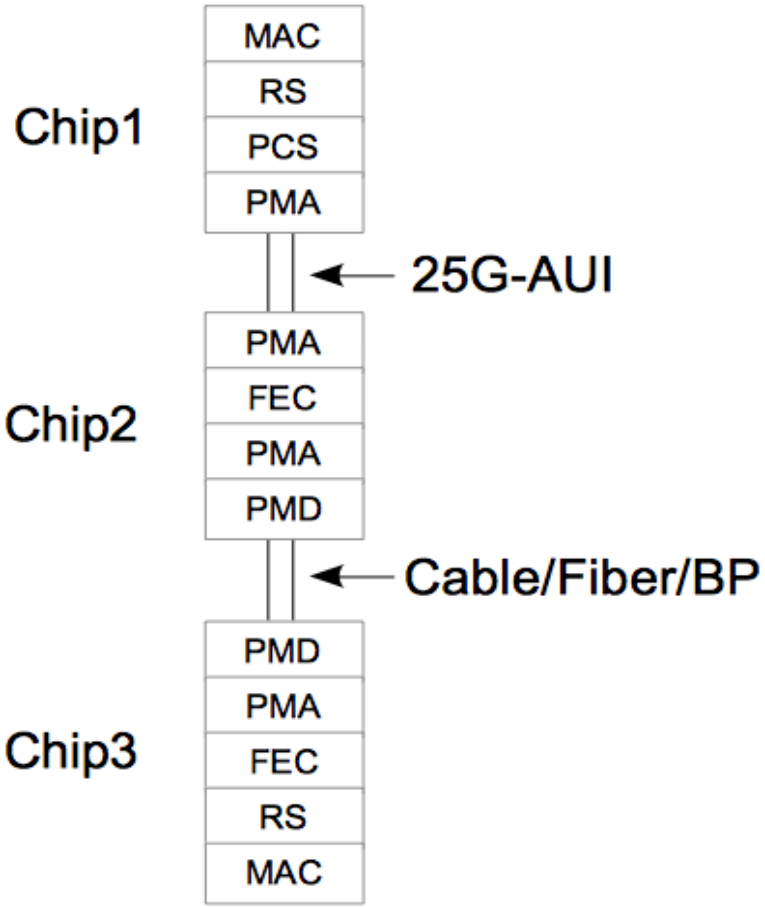
Operation	noFEC	base-R FEC	100G-RS	25G RS-CWM	25G RS-SnT
Block/CW alignment	✔	✔	✔	✔	✔
FEC decode		✔	✔	✔	✔
Marker Removal[1]				✔	
Transcode to 66b		✔	✔	✔	✔
Descramble[2]				✔	
Find IPG & Insert /I/,/LI/[2]				✔	
Scramble[2]				✔	
ODU map	✔	✔	✔	✔	✔

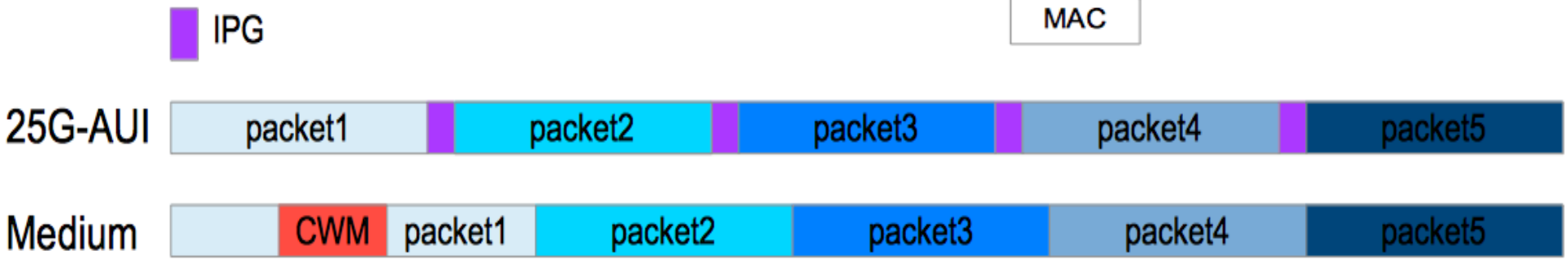
[1] CWM removal is needed since they're added by Cl 108 FEC, not needed in 100G since they have were added in by the PCS and transcoder has 66b mappings for them.

[2] IDLE restoration needed to have all 25G rates running with same throughput. If you just remove Markers then 1 of the 3 operating modes runs with a different data throughput.

1588 Issue

Chip1 and Chip3 have MAC,RS,PCS integrated to support 1588. Addition of Chip2 causes variation in the arrival times to occur during transmission since the CWM are inserted at interval that don't always align with IPGs.





FlexEthernet Issue

Initial FlexE Shim could account for 25GE CWM, however if something converts the 25GE-RSFEC w/ CWM to 25GE non RS-FEC you will corrupt FlexE Frame.

This affect would also occur even when the end is 25G RS-FEC w/ CWM if you've removed the CWM and re-inserted them. There is no guarantee that IDLEs that were added/deleted for the CWM are the same location.

Edits to D1.0 inline

~~*108.5.2.2 Rate compensation for codeword markers in the transmit direction*~~

~~The RS-FEC transmit process periodically inserts codeword markers into the transcoded block stream (see 108.5.2.4). In order to maintain the same bit rate after codeword marker insertion, the RS-FEC transmit process shall perform the rate compensation function described below, or its functional equivalent:~~

~~a) Decode the PCS blocks received by descrambling (see 49.2.10) and applying the PCS receive process (see 49.2.11) to obtain the 25G-MII character stream.~~

~~b) Delete idle characters and ordered sets, according to the rules in 49.2.4.7 and 49.2.4.10, to create room as necessary for the periodic codeword markers.~~

~~c) Re-encode the data stream obtained, by applying the PCS transmit process (see 49.2.5) and scrambler (see 49.2.6) to obtain 64B/66B blocks.~~

~~If the optional EEE deep sleep capability is supported, re-encoding in item c) is affected by the value of~~

~~scrambler_bypass. When scrambler_bypass is true, the descrambled data is passed to the transcoder, rather than the data from the scrambler output. The scrambler continues to operate normally, shifting input data into the delay line. When scrambler_bypass is set to false, the data from the scrambler output is passed to the transcoder.~~

Edits to D1.0 inline (cont)

~~108.5.2.4 Codeword marker insertion~~

~~In order to support codeword alignment in the receive direction, the 25GBASE-R RS-FEC shall periodically insert codeword markers into the stream of transcoded blocks at predefined locations, as the first 257 bits of every 1024th RS-FEC codeword. The distance between the beginning of successive codeword markers is therefore 20480 257-bit transcoded blocks, equivalent to 81920 64B/66B blocks. Room for codeword markers is created by the rate compensation for codeword markers in the transmit direction process (see 108.5.2.2) such that the bit rates at the input and the output of the 25GBASE-R RS-FEC sublayer are equal.~~

~~The transmitted codeword marker is a 257-bit block, tx_cwm, constructed as follows:~~

~~1) tx_cwm<0:23> are set to the content of the three octets M0, M1 and M2 of the alignment marker of PCS lane 0 as defined in Table 82-2.~~
~~2) tx_cwm<24:31> are set to 0x33.~~
~~3) tx_cwm<32:55> are set to the content of the three octets M4, M5 and M6 of the alignment marker of PCS lane 0 as defined in Table 82-2.~~
~~4) tx_cwm<56:63> are set to 0xCC.~~
~~5) tx_cwm<64:87> are set to the content of the three octets M0, M1 and M2 of the alignment marker of PCS lane 1 as defined in Table 82-3.~~
~~6) tx_cwm<88:95> are set to 0x33.~~
~~7) tx_cwm<96:119> are set to the content of the three octets M4, M5 and M6 of the alignment marker of PCS lane 1 as defined in Table 82-3.~~
~~8) tx_cwm<120:127> are set to 0xCC.~~
~~9) tx_cwm<128:151> are set to the content of the three octets M0, M1 and M2 of the alignment marker of PCS lane 2 as defined in Table 82-3.~~
~~10) tx_cwm<152:159> are set to 0x33.~~
~~11) tx_cwm<160:183> are set to the content of the three octets M4, M5 and M6 of the alignment marker of PCS lane 2 as defined in Table 82-3.~~
~~12) tx_cwm<184:191> are set to 0xCC.~~
~~13) tx_cwm<192:215> are set to the content of the three octets M0, M1 and M2 of the alignment marker of PCS lane 3 as defined in Table 82-3.~~
~~14) tx_cwm<216:223> are set to 0x33.~~
~~15) tx_cwm<224:247> are set to the content of the three octets M4, M5 and M6 of the alignment marker of PCS lane 3 as defined in Table 82-3.~~
~~16) tx_cwm<248:255> are set to 0xCC.~~
~~17) tx_cwm<256> is set to 0.~~

Edits to D1.0 inline (cont)

108.5.2.5a PN-5280 pseudo-noise sequence generator

The PN-5280 pseudo-noise generator is identical the one defined in 74.7.4.4.1 with the exception that it's length is 5280.

Scrambling with the PN-5280 sequence at the RS-FEC codeword boundary is necessary for establishing RS-FEC block synchronization (to ensure that any shifted input bit sequence is not equal to another RS-FEC codeword) and to ensure DC balance.

108.5.2.6 Codeword serialization

Once the data has been Reed-Solomon encoded and scrambled by the PN-5280, it shall be serialized and sent to the PMA using the PMA:IS_UNITDATA.request primitive with the transmit bit ordering illustrated in Figure 108–3.

Edits to D1.0 inline (cont)

108.5.2.7 RS-FEC encoding for rapid codeword lock (EEE deep sleep)

If the optional EEE deep sleep capability is supported, the RS-FEC transmit function changes its signaling during LPI refresh and wake periods, as described in this subclause. The transition of the PCS LPI transmit state diagram (Figure 49–12) to TX_WAKE state is detected by the change of the tx_mode parameter of the FEC:IS_TX_MODE.request primitive from ALERT to DATA. Following this transition, the transmit function behaves as follows:

~~a) The variable scrambler_bypass is set to TRUE for a period of 0.9 µs to 1.1 µs. This causes the rate compensation function (108.5.2.2) to generate unscrambled data.~~

~~b) The first codeword marker insertion (see 108.5.2.4) is performed at the beginning of the second full codeword (after the first full codeword has been transmitted). Subsequent codeword markers are inserted at regular offsets from this first codeword marker.~~

~~NOTE—During the TX_WAKE state of the PCS LPI transmit state diagram (Figure 49–12), the data encoded by the PCS comprises one of two control characters, either /I/ or /LI/. See 78.1.3.1 and 78.1.3.3.1 for more details about LPI wake cycles.~~

~~As a result of the transmit function behavior and the PCS data stream content during TX_WAKE state, at least four RS-FEC codewords sent over the channel comprise deterministic transcoded blocks, and a codeword marker is inserted at a predetermined location. This enables rapid synchronization of the remote RS-FEC receive function and determination of the block types and receive LPI state by the remote PCS.~~

a) Transmit data per normal for a period of 1.1us to 1.3us

b) Descramble the 66b blocks provided (see 49.2.10) to the transcoder for a period of 1.1us to 1.3us

c) Return to normal data transmission

As a result of the transmitting descrambled 66b blocks at least 4 RS-FEC codewords with known data patterns are produced at the output of the PN-5280 pseudo-noise generator. The known data patterns can then be used by the receive process to achieve a rapid codeword lock.

Edits to D1.0 inline (cont)

108.5.3.1 Codeword ~~marker~~ lock

The 25GBASE-R RS-FEC shall implement the codeword ~~marker~~ lock process as described in this subclause. The RS-FEC receive function forms a bit stream by concatenating the bits from the PMA:IS_UNITDATA.indication primitive in the order they are received and descrambling them with a PN-5280 generator. This process obtains lock to the codeword ~~markers~~ as shown in FEC synchronization state diagram (Figure 108–5). The status of the codeword ~~marker~~ lock process shall be reflected by the state variable FEC_align_status.

108.5.3.2 Reed-Solomon decoder

....

In addition, it shall ensure ~~rx_coded_0<1:0> corresponding to the second 257-bit block and~~ rx_coded_3<1:0> corresponding to the last (20th) 257-bit block in the codeword are set to 11.

Edits to D1.0 inline (cont)

108.5.3.3 Codeword monitor

After codeword ~~marker~~ lock has been achieved, this process continuously checks codeword validity as indicated by the codeword monitor state diagram (Figure 108–6). When three consecutive uncorrected codewords are detected, the codeword monitor shall restart the codeword ~~marker~~ lock process. In such event, it is likely that multiple blocks are marked as bad until codeword ~~marker~~ alignment is found, leading to hi_ber being set by the PCS. When Auto-Negotiation is supported and enabled, this event causes Auto-Negotiation to restart.

~~*108.5.3.4 Alignment marker removal*~~

~~The first 257 message bits in every 1024th codeword is the vector rx_cwm<256:0> where bit 0 is the first bit received. The specific codewords that include this vector are indicated by the codeword marker lock function (108.5.3.1).~~

~~The vector rx_cwm shall be removed prior to transcoding.~~

Edits to D1.0 inline (cont)

~~*108.5.3.6 Rate compensation for codeword markers in the receive direction*~~

~~After the codeword markers have been discarded from transcoding and the stream of rx_coded<65:0> vectors has been obtained, to compensate for the deleted codeword markers, the RS-FEC receive process shall perform the rate compensation function described below, or its functional equivalent:~~

~~a) Decode the stream of rx_coded vectors by descrambling (see 49.2.10) and applying the PCS receive process (see 49.2.11) to obtain the 25G-MII character stream.~~

~~b) Insert idle characters, according to the rules in 49.2.4.7, to fill in as necessary for any deleted codeword markers.~~

~~c) Re-encode the data stream obtained, by applying the PCS transmit process (see 49.2.5) and scrambling (see 49.2.6) to obtain 64B/66B blocks rx_coded_out<65:0>.~~

~~If the optional EEE deep sleep capability is supported, the process in list item a) is affected by the value of descrambler_bypass. When descrambler_bypass is true, the received data is used without descrambling.~~

~~The descrambler continues to operate normally, shifting input data into the delay line. When descrambler_bypass is set to false, data from the descrambler output is used.~~

~~If rx_coded<1:0> is either 00 or 11, the process in list item c) shall set rx_coded_out<1:0> to rx_coded<1:0> and the process in list item b) shall not insert idle characters at the next block after rx_coded_out.~~

Edits to D1.0 inline (cont)

108.5.3.7 Rapid codeword lock for EEE deep sleep

If the optional EEE deep sleep capability is supported, the RS-FEC receive function performs rapid codeword lock during LPI refresh and wake periods, as described in this subclause. When rx_mode (or rx_tx_mode if appropriate) transitions from QUIET to DATA:

a) ~~Set descrambler_bypass to true. This causes the rate compensation function to use the receive data without descrambling (see 108.5.3.6).~~

b) Start a hold-off timer whose duration is greater than or equal to 13.7 µs.

c) Enable the RS-FEC rapid codeword lock mechanism, which attempts to determine the start location of the RS-FEC codeword based on the deterministic patterns sent by the remote RS-FEC transmit function (see 108.5.2.7).

NOTE—The rapid codeword lock mechanism is implementation dependent and outside the scope of this standard.

When the start location is found, FEC_align_status is set to true and test_cw is asserted for each subsequent codeword, enabling aligned codeword decoding. Assuming the rapid codeword lock has determined the correct start of codeword location, the RS-FEC codeword monitor state diagram (Figure 108–6) reaches the CW_GOOD state~~, and then the decoding in item a) of 108.5.3.6 results in one of two deterministic blocks, comprised of either /I/ or /LI/ control characters, with no decoding errors. When the decoding in item a) of 108.5.3.6 starts generating decoding errors while codeword monitor is in CW_GOOD state and descrambler_bypass is true, it is an indication that the remote RS-FEC transmitter has re-enabled scrambling, and descrambler_bypass is set to false.~~

The FEC sublayer shall hold off asserting signal_ok until one of the following two events occurs:

1) Two 64B/66B blocks ~~generated by the rate compensation function (108.5.3.6) after the transition of descrambler_bypass from true to false.~~after the transition from deterministic FEC block to normal scrambled FEC block

2) Expiration of the hold-off timer.

Edits to D1.0 inline (cont)

108.5.4.2 State variables

~~**cwm_counter_done**~~

~~Boolean variable that indicates that cwm_counter has reached its terminal count.~~

~~**cwm_valid**~~

~~Boolean variable that is set to true if the received block rx_cwm<0:256> (as defined in 108.5.3.4) is a valid codeword marker. Codeword marker validity is tested by comparing bits 0:23 and 32:55 of rx_cwm<0:256>, on a nibble-wide basis, against their respective values in tx_cwm<0:256> (as defined in 108.5.2.4). If nine or more nibbles in the candidate block match the corresponding known nibbles in the codeword marker, the candidate block is considered a valid codeword marker.~~

~~Editor's note: response to comment #34 against D0.1 included different text for the definition of cwm_valid, which, in the editor's view, was incorrect. The text above is based on the original suggested remedy of comments #34 and #116.~~

~~**descrambler_bypass**~~

~~Boolean variable that controls bypassing the descrambler in the rate compensation in the receive direction (108.5.3.6), in order to assist rapid synchronization following LPI refresh or wake. If the optional EEE deep sleep capability is supported, descrambler_bypass is controlled by the rapid codeword lock process during LPI wake cycles (see 108.5.3.7). If the optional EEE deep sleep capability is not supported, descrambler_bypass is always false.~~

~~**scrambler_bypass**~~

~~Boolean variable that controls bypassing the scrambler in the rate compensation in the transmit direction (108.5.2.2), in order to assist rapid synchronization following LPI refresh or wake. If the optional EEE deep sleep capability is supported, scrambler_bypass is set to true during LPI wake cycles (see 108.5.2.7). If the optional EEE deep sleep capability is not supported, scrambler_bypass is always false.~~

~~**test_cwm**~~

~~Boolean variable that is set to true when a candidate block position is available for testing, and is set to false according to the FEC synchronization state diagram in Figure 108–5.~~

test_cw

Boolean variable that is set to true when a new FEC codeword is available for decoding and is set to false according to the ~~codeword monitor~~ state diagrams in Figure 108-5 and Figure 108–6.

parity_valid

Boolean variable that is set to true if the parity received in the FEC codeword matches the calculated parity and is set to FALSE otherwise.

108.5.4.4 Counters

~~**cwm_counter**~~

~~This counter counts the received codewords that separate the ends of two consecutive codeword markers. An RS-FEC codeword is 5280 bits. The terminal count of this counter is equal to the codeword offset between transmitted codeword markers, 1024.~~