

802.3dj D1.4 Comment Resolution Logic Track

Gary Nicholl (Cisco), Logic Track Lead Editor
Eugene Opsasnick (Broadcom), Logic Editor
Matt Brown (Alphawave Semi), 802.3dj Chief Editor

Introduction

- This slide package was assembled by the 802.3dj editorial team to provide background and detailed resolutions to aid in comment resolution.
- Specifically, these slides are for the logic track comments

PCS Decode

Comment #239

Comment #239

Cl 119 SC 119.2.5.8.2 P166 L15 # 239
Ran, Adeo Cisco
Comment Type T Comment Status D PCS encode/decode

The stateless decoder assumes that the received data represent valid Ethernet data and does not check it for valid frame structure, unlike the State-diagram decoder.

This should be emphasized for readers familiar with the original decoder defined in Clause 119 to prevent surprises. For example, validation suites may check the PCS with data that is not valid Ethernet and expect it to reject it.

The suggested remedy applies to this subclause (119.2.5.8.2) and to 175.2.5.9. It should also apply to 172.2.5.9.2, but it is currently not in the draft and may be out of scope.

Suggested Remedy

Add a NOTE at the end of 119.2.5.8.2:

NOTE—The stateless decoder relies on the Reed-Solomon decoder for error correction and marking, and unlike the state-diagram decoder, it does not check the validity of Ethernet frames.

Add a similar note at the end of 175.2.5.9.

Add a similar note at the end of 172.2.5.9.2 if it is considered in scope.

Proposed Response Response Status W

PROPOSED ACCEPT IN PRINCIPLE.

The stateless PCS decoder is defined in CL 172 and there are references to it from CL 119 and CL 175. The best place for this note would be in CL 172 with the decoder definition itself, and then notes in 119 and 175 should not be necessary.

Add a note to 172.2.5.9 with editorial license something like:

"NOTE- The stateless decoder does not detect all packet framing errors that the state-diagram decoder can detect. It relies on the RS FEC decoder for error correction and marking as well as the MAC ethernet frame FCS check for frame reliability."

Implement with editorial license.

[Editor's note: CC 172]

Add note after table 172-4.

119.2.5.8.2 Stateless decoder

The stateless decoder generates 200GMII/400GMII transfers based only on the current and preceding 66-bit blocks. The decoder shall decode each 66-bit block $rx_coded_{<65:0>}$ to a 72-bit vector $rx_raw_{<71:0>}$ (see 119.2.6.2.2) according to the rules in Table 172-4. Constants LBLOCK_R and EBLOCK_R are defined in 119.2.6.2.1. Variables reset, rx_raw, and rx_coded are defined in 119.2.6.2.2. Functions R_TYPE and DECODE, and the block types are defined in 119.2.6.2.3.

175.2.5.9 64B/66B decoder

The receive PCS decodes 66-bit blocks to produce RXD<63:0> and RXC<7:0> for transmission to the 1.6TMII. One 1.6TMII transfer is decoded from each 66-bit block. The receive PCS may use either the state-diagram decoder defined by Figure 119-1 or the stateless decoder defined in 172.2.5.9.2.

172.2.5.9.2 Stateless decoder

The stateless decoder generates 800GMII transfers based only on the current and preceding 66-bit blocks. The decoder shall decode each 66-bit block $rx_coded_{<65:0>}$ to a 72-bit vector $rx_raw_{<71:0>}$ (see 172.2.6.2.2) according to the rules in Table 172-4. Constants LBLOCK_R and EBLOCK_R are defined in 172.2.6.2.1. Variables reset, rx_raw, and rx_coded are defined in 172.2.6.2.2. Functions R_TYPE and DECODE, and the block types, are defined in 172.2.6.2.3.

Table 172-4—PCS stateless decoder rules

reset	R_TYPE(rx_coded_{i-1}) ^a	R_TYPE(rx_coded_i) ^b	Resulting rx_raw
1	any block type	any block type	LBLOCK_R
0	any block type	E	EBLOCK_R
0	E	any block type	EBLOCK_R
0	any combination not listed above		DECODE(rx_coded_i)

^a rx_coded_{i-1} is the 66-bit block that immediately precedes rx_coded_i .

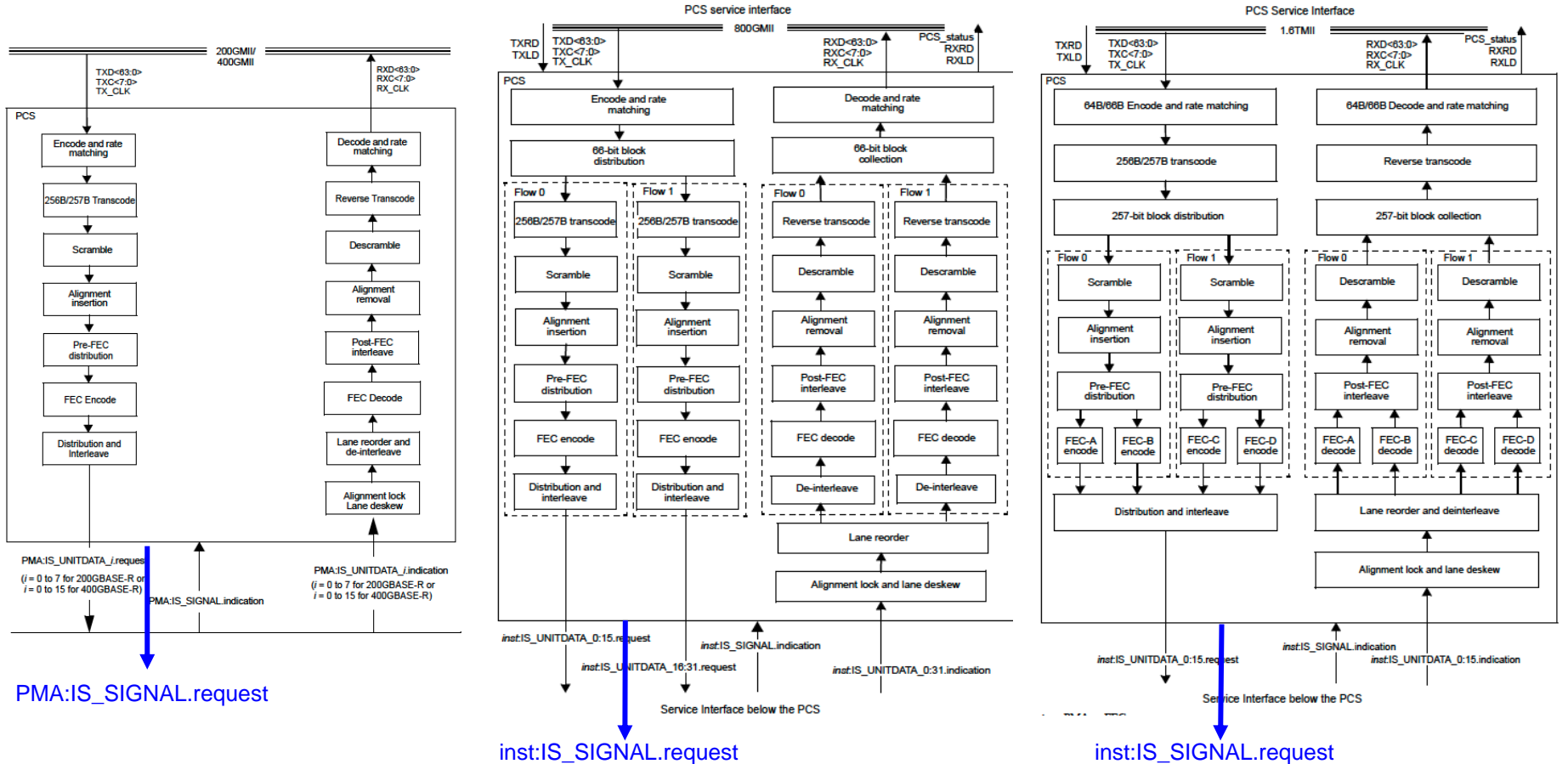
^b rx_coded_i is the 66-bit block that is being decoded.

Note - The stateless decoder does not detect ...

PCS IS_SIGNAL.request

Comments #248, #251, #236

Comment #248 - 200G/400G PCS, 800G PCS, 1.6T PCS



Comment #251 - PMA IS_SIGNAL Generation

Table 176-6—*inst:IS_SIGNAL.request(SIGNAL_OK)* generation

<i>PMA:IS_SIGNAL.request</i> ^a SIGNAL_OK	<i>align_status_mux</i> ^b or <i>all_locked_demux</i> ^c	<i>inst:IS_SIGNAL.request</i> ^d SIGNAL_OK
OK	true	OK
OK	false	READY
READY	don't care	READY
IN_PROGRESS	don't care	IN_PROGRESS
FAIL	don't care	FAIL
no_primitive^e	true	OK
no_primitive^e	false	READY

^a From the sublayer above the PMA.

^b For m:n PMAs (see 176.4.4.2.1).

^c For n:m PMAs (see 176.4.4.2.1).

^d To the service interface below the PMA.

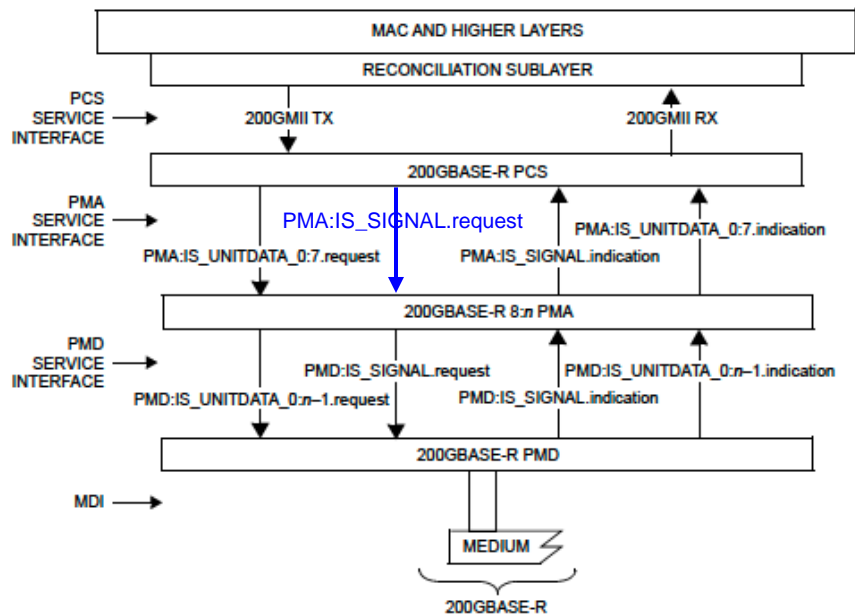
~~^e When *PMA_IS_SIGNAL.request* input is not present. For example, when the sublayer above the PMA is a PCS or DTE XS.~~

PMA:IS_SIGNAL.request (from the sublayer above the PMA) can come from:

- Another PMA
- AUI
- PCS (newly added input)
- DTE XS (newly added input)

inst:IS_SIGNAL.request generation is more clear and consistent.

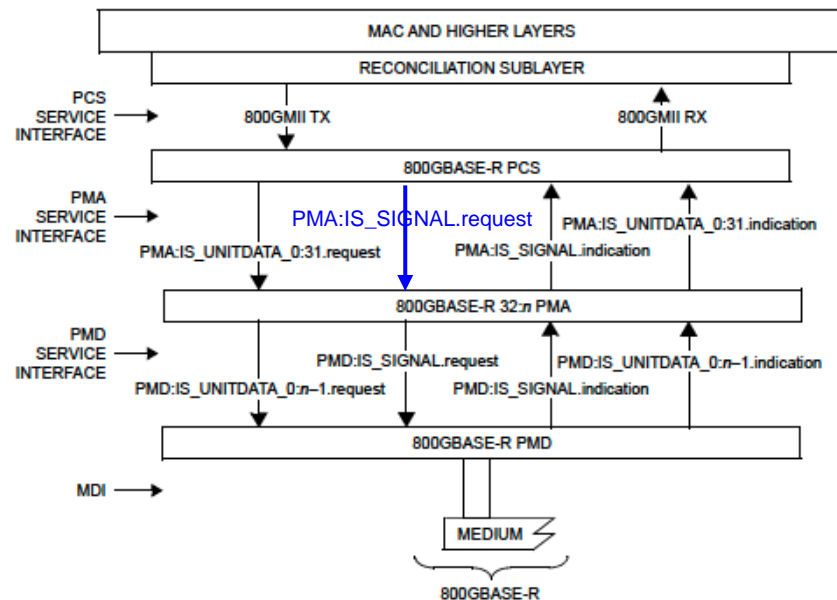
Comment #236 - Example overview figures for 200G and 800G



200GMII = 200 Gb/s MEDIA INDEPENDENT INTERFACE
 MAC = MEDIA ACCESS CONTROL
 MDI = MEDIUM DEPENDENT INTERFACE
 PCS = PHYSICAL CODING SUBLAYER

PMA = PHYSICAL MEDIUM ATTACHMENT
 PMD = PHYSICAL MEDIUM DEPENDENT
 n = NUMBER OF PARALLEL STREAMS OF DATA UNITS

Figure 116-2—200GBASE-R inter-sublayer service interfaces



800GMII = 800 Gb/s MEDIA INDEPENDENT INTERFACE
 MAC = MEDIA ACCESS CONTROL
 MDI = MEDIUM DEPENDENT INTERFACE
 PCS = PHYSICAL CODING SUBLAYER

PMA = PHYSICAL MEDIUM ATTACHMENT
 PMD = PHYSICAL MEDIUM DEPENDENT
 n = NUMBER OF PARALLEL STREAMS OF DATA UNITS

Figure 169-2—800GBASE-R inter-sublayer service interfaces not including 800GMII Extender

Comment #236 - examples with extender

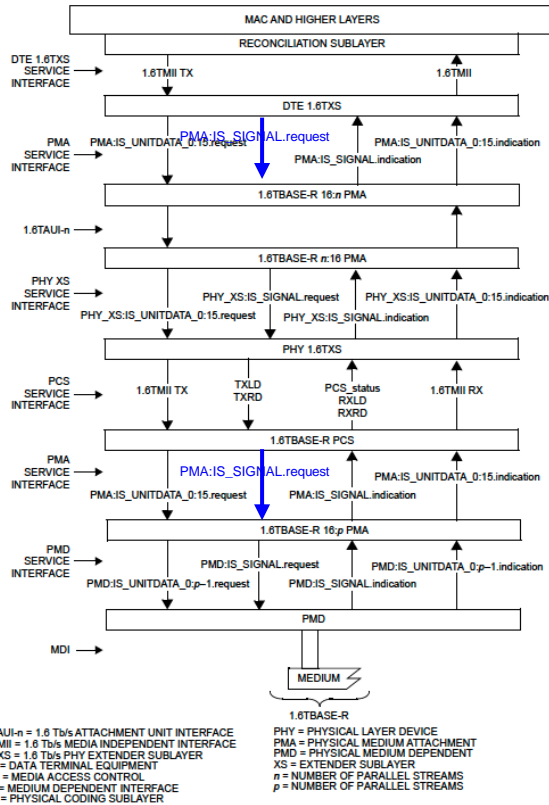


Figure 174-3—Inter-sublayer service interfaces for a 1.6TBASE-R PHY and 1.6TMII Extender

Fig. 174-3 Is a typical xBASE-R PHY with extender (Copper/IMDD PHYs).

- IS_SIGNAL.request added to both DTE and PCS Tx outputs

Fig. 185-2 and 185-3 are a special case for coherent PHY 800GBASE-LR1

- Coherent links do not support ILT
- Per comment #21, Fig. 185-3 is removed Instructions for changes to Fig. 185-3 in comment #236 can be ignored.
- Per comment #21, Fig. 185-2 is adding an 800G-AUI.
- IS_SIGNAL.request and IS_SIGNAL.indication should be added to the new inter-sublayer interfaces in Fig. 185-2

Convolutional Interleaver

Comment #47

Comment #47 - Clause 177, Convolutional Interleaver

CI 177 SC 177.4.2 P 318 L 34 # 47

Bruckman, Leon

Nvidia

Comment Type TR Comment Status D convolutional interleaver

The relationship between the position of the input and output switches in Figure 177-4 is not defined.

Suggested Remedy

Add the following sentence at the end of the paragraph: "The input and output switches are always aligned to the same row."

Proposed Response Response Status W

PROPOSED REJECT.

It is not required to keep the input and output switches aligned to the same row.

From 177.4.2:

The data from deskewed PMA lane is fed into each delay line one RS-FEC symbol-quartet at a time. The input data round-robins between the three delay lines in the order Delay Line 0, then Delay Line 1 and lastly Delay Line 2.

The output of the convolutional interleaver round-robins between the three delay lines, receiving one RS-FEC symbol-quartet from each at a time, in the order Delay Line 0, then Delay Line 1 and lastly Delay Line 2.

Figure 177-4 illustrates the delay lines of the convolutional interleaver using Q instances of a RS-FEC symbol-quartet sized base delay element labeled "D".

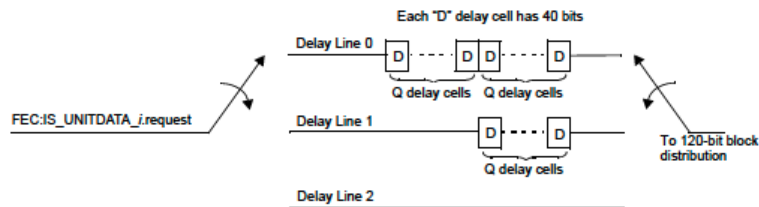


Figure 177-4—Convolutional interleaver

Comment #47 - Comparison to CL 184 Convolutional Interleaver

From 184.4.5:

The following list describes the convolutional interleaver process:

- a) The input and output switches are always aligned to the same row b , where $b = 0$ to 2
- b) A block of 40 bits is read from row b .
- c) The contents of row b are shifted to the right by 40 bits.
- d) A block of 40 bits is written to row b .
- e) The position of the switches is updated to $(b + 1) \bmod 3$

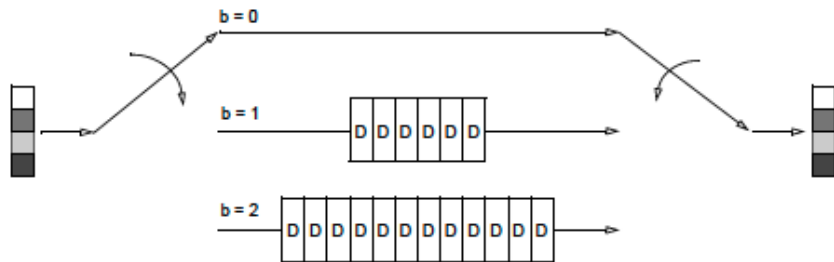


Figure 184-4—Convolutional interleaver

184.4.5: Convolutional Interleaver description

Part (a) Input and Output switches are always aligned to the same row.

It is a necessary part of the logical description of the CI functionality.

Change response to comment #47 for 177.4.2
From: REJECT
To: ACCEPT IN PRINCIPLE
Implement the suggested remedy with editorial license.

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22

ER1 AM Location Function

Comments #70

Comment #70 - Figure 186-20

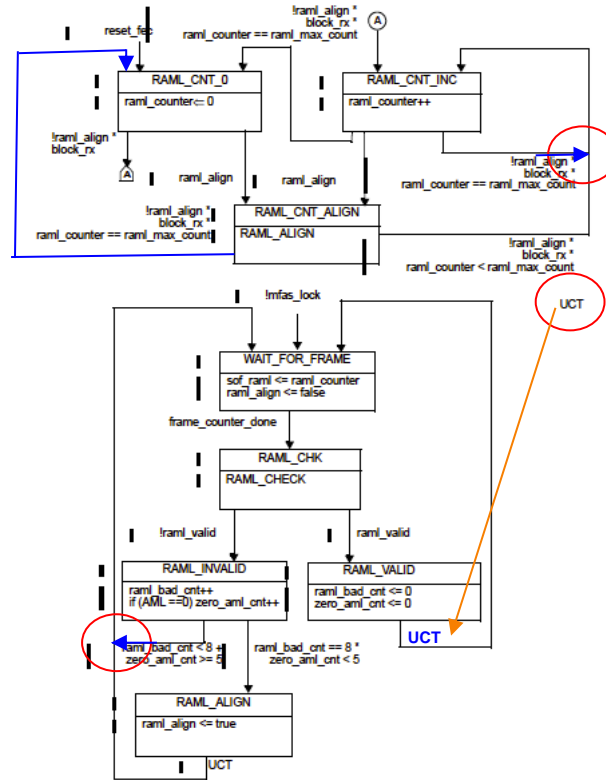
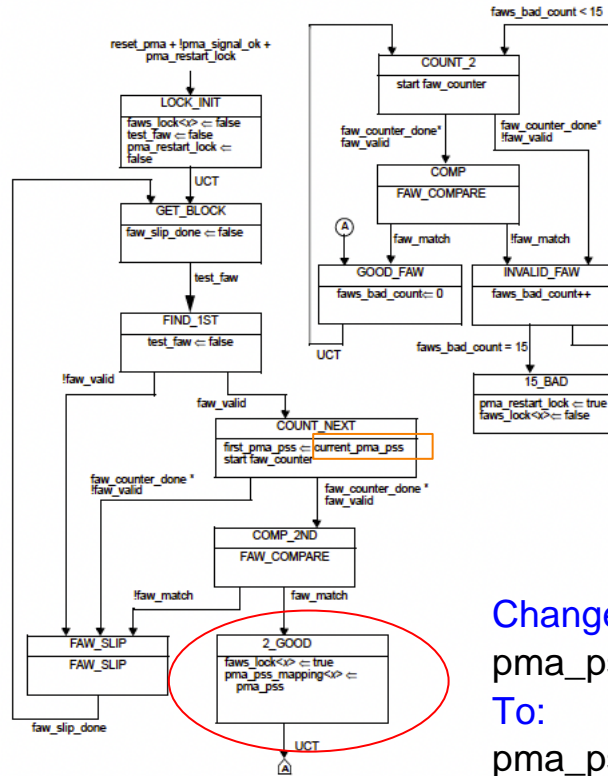


Figure 186-20—800GBASE-ER1 FEC Alignment marker location state diagram

ER1 PMA Frame Alignment

Comment #69

Comment #69 - Figure 186-16



Change from:

`pma_pss_mapping<x> <- pma_pss`

To:

`pma_pss_mapping<x> <- current_pma_pss`

Figure 186-16—800GBASE-ER1 PMA FAW field lock state diagram

Comment #69 - Figure 186-16 - [current_]pma_pss

current_pma_pss

A variable that holds the 800GBASE-ER1 PMA polarization symbol stream identifier corresponding to the current frame alignment word sequence that is recognized on a given polarization symbol stream of the 800GBASE-ER1 PMD service interface. It is compared to the variable first_pma_pss to confirm that the location of the frame alignment word sequence has been detected.

pma_pss

A variable that holds the 800GBASE-ER1 PMA polarization symbol stream identifier received on the 800GBASE-ER1 PMA service interface when faws_lock<x> = true. The 800GBASE-ER1 PMA polarization symbol stream identifier is determined by matching the received 22-symbol frame alignment word sequence to the values in one of the columns of Table 186–6.

ER1 MDIO

Comments #106, #107

Comment #106 and #107

Cl **186** SC **186.7.1** P **607** L **25** # **106**

Huber, Thomas Nokia

Comment Type **T** Comment Status **D** ER1 MDIO

In tables 186-7 and 186-8, there are a number of rows that are missing a variable reference. These are all variables that are related to the "Inverse RS FEC" function that is specified by reference to clause 172.

Suggested Remedy

Determine if all these variables are needed, add references for the ones that are and delete any that are not needed.

Proposed Response Response Status **W**

PROPOSED ACCEPT IN PRINCIPLE.

For the variable reference, list both the subclause of the inverse FEC function (186.2.x) and the subclause of 172 where the variable is defined.

Details to be provided in editorial team slides.

Cl **186** SC **186.7.1** P **607** L **25** # **107**

Huber, Thomas Nokia

Comment Type **T** Comment Status **D** ER1 MDIO

In tables 186-7 and 186-8, there are a number of rows that are missing MDIO register/bit numbers and pointers to clause 45.

Suggested Remedy

Add the missing register/bit numbers and pointers to clause 45.

Proposed Response Response Status **W**

PROPOSED ACCEPT IN PRINCIPLE.

Specify the registers as indicated in the editorial slides.

Proposed resolution to #106 and #107

Variable references and MDIO registers related to the inverse FEC sublayer

Update references to the RS FEC decoder subclause in the control variables:

Align the set of variables to clause 172 and add references to FEC decoder and lane alignment subclauses in the status variables:

Table 186-7—FEC control variables and MDIO mapping

Control variable	Variable reference	MDIO register/bit number	MDIO register/bit reference
FEC_reset	186.4.2.1	1.0.15	45.2.1.1
tx_test_mode	186.2.3.12		
fec_alignment_marker_location_enable	186.2.3.5.10	1.1825.14	45.2.1.178c
IFEC_bypass_indication_enable	186.2.3.1.3	1.200.1	45.2.1.116
IFEC_degraded_SER_enable	186.2.3.1.3	1.200.4	45.2.1.116
IFEC_degraded_SER_activate_threshold	186.2.3.1.3	1.650, 1.651	45.2.1.146
IFEC_degraded_SER_deactivate_threshold	186.2.3.1.3	1.652, 1.653	45.2.1.147
IFEC_degraded_SER_interval	186.2.3.1.3	1.654, 1.655	45.2.1.148

rx_local_degraded			
IFEC_align_status	186.2.3.1.1	1.201.14	45.2.1.117
IFEC_amps_lock<0:31>	186.2.3.1.1		
IFEC_cw_counter	186.2.3.1.3		
IFEC_codeword_error_bin<1:15>	186.2.3.1.3		
IFEC_pcs_lane_mapping<0:31>	186.2.3.1.3		
IFEC_symbol_error_counter<0:31>	186.2.3.1.3		
IFEC_bypass_indication_ability	186.2.3.1.3	1.201.1	45.2.1.117
IFEC_hi_ser_combined	186.2.3.1.3		
IFEC_degraded_SER_ability	186.2.3.1.3	1.201.3	45.2.1.117
IFEC_degraded_SER	186.2.3.1.3	1.201.4	45.2.1.117
IFEC_rx_rm_degraded	186.2.3.1.3		
IFEC_rx_local_degraded	186.2.3.1.3		
IFEC_corrected_cw_counter	186.2.3.1.3	1.202, 1.203	45.2.1.118
IFEC_uncorrected_cw_counter	186.2.3.1.3	1.204, 1.205	45.2.1.119

Update existing references for IFEC_abc variables to use the IFEC registers in 1.22xx rather than 1.20x or 1.65x registers. Define MDIO registers/bit assignments (1.xxxx) and add subclauses to 45.2.1 as necessary for the variables that are missing those references (see subsequent slides), implement with editorial license.

Proposed resolution to #106 and #107

Update references and define new IFEC registers - control variables

- Define a register and bit for tx_test_mode. This exists in the PCS, but we can't reuse that
- Change IFEC_bypass_indication_enable to refer to 1.2200.1 (currently points at 1.200.1)
- Add a register/bit for IFEC_degraded_SER_enable (currently points at 1.200.4)
- Add registers for IFEC_degraded_SER_activate_threshold, IFEC_degraded_SER_deactivate_threshold, and IFEC_degraded_SER_interval (currently point at 1.650 through 1.655)

Proposed resolution to #106 and #107

Update references and define new IFEC registers - status variables

- Add a register/bit for fec_alignment_marker_location_ability. This is a new variable
- Change IFEC_align_status to refer to 1.2201.14 (currently points to 1.201.14)
- Figure out what to do for amps_lock<0:31>. 1.2201.8:11 are defined for lanes 0-3; we need to either put lanes 4-31 somewhere else, or define an entirely new set of 4 bytes that has 32 lanes.
- Add registers for IFEC_cw_counter and the 16 bins IFEC_codeword_error_bin<0:15>
- Change IFEC_corrected_cw_counter and IFEC_uncorrected_cw_counter to refer to 1.2202 through 1.2205
- Figure out what to do about IFEC_PCS_lane_mapping (like amps_lock, there are registers for lanes 0-3, but we need 32)
- Add IFEC_symbol_error_counter registers for each lane
- Change IFEC_bypass_indication_ability to refer to 1.2201.1 (currently points to 1.201.1)
- Add IFEC_hi_ser_combined, IFEC_degraded_SER, IFEC_rx_rm_degraded, IFEC_rx_local_degraded registers