802.3dt D1.0 Editorial Comment Resolution

Eugene Opsasnick (Broadcom), 802.3dt Chief Editor

Introduction

- This slide package was assembled by the 802.3dt editorial team to provide background and detailed resolutions to aid in comment resolution.

Ordered sets and O code

Comment #52 and others

Comment #52 - Ordered Sets in 802.3-2022 and 802.3dt

- Ordered sets in the current Clause 81 & 82
  - 802.3 Clauses 81 & 82 only define “sequence ordered sets” (with O code of 0x0)
  - 802.3 reserves some O code values for use by other SDOs in a footnote of Table 82-1, but does say anything else about them. Their format and use is defined by the other SDO.
    - 0xF for Fiber Channel
    - 0x5 for OIF
    - 0xC for ITU-T
  - These “other” ordered sets (not “sequence ordered sets”), up to now, must use ***non-standard*** methods to transmit and receive them (with non-zero O code).
- A main goal of 802.3dj is to formally define methods within 802.3 to transmit and receive “other” ordered sets in a general way.
  - 802.3dt gives all “other” ordered sets a name: “control ordered sets”, and creates a method to transmit and receive them.
  - Other SDOs can continue to define their own format and use (with their assigned O code)
  - Other SDOs may use the new 802.3dj methods to transmit and receive control ordered sets using their own O code or can continue to use their previous ***non-standard*** methods.

Comment #52 - Ordered Sets - 66-bit block definition

- Changing D4-D7 in block type 0x4B from 0x000_0000 to general (D)ata is backward compatible with all uses of 0x4B since zeros are still valid.
  - Sequence ordered sets defined in Clause 81 set D4-D7 to all zeros on xMII

Table 81–5—Sequence ordered sets

Lane 0	Lane 1	Lane 2	Lane 3	Lane 4	Lane 5	Lane 6	Lane 7	Description
Sequence	0x00	0x00	0x00	0x00	0x00	0x00	0x00	Reserved
Sequence	0x00	0x00	0x01	0x00	0x00	0x00	0x00	Local Fault
Sequence	0x00	0x00	0x02	0x00	0x00	0x00	0x00	Remote Fault
Sequence	0x00	0x00	0x03	0x00	0x00	0x00	0x00	Link Interruption

  - There is no definition for the “other” ordered sets (non-zero O code) in 802.3
    - It is left to the other organizations.

Comment #52 - Ordered Sets - CL 82 PCS: Control code values

Table 82–1—Control codes

Control character	Notation	XLGMII/ CGMII control code	40/100GBASE-R O code[a]	40GBASE-R and 100GBASE-R control code
idle	/I/	0x07		0x00
LPI	/LI/	0x06		0x06
start	/S/	0xFB		Encoded by block type field
terminate	/T/	0xFD		Encoded by block type field
error	/E/	0xFE		0x1E
Sequence ordered set	/Q/	0x9C	0x0	Encoded by block type 0x4B plus O code, control codes are set to 0x00

[a] INCITS T11 Fibre Channel uses O code 0xF for the signal ordered set. OIF uses O code 0x5 for the FlexE [B59] ordered set. ITU-T uses O code 0xC for the G.8312 [B47] MTN path ordered set.

xMII control codes (8-bits) are shown in Table 82-1 (as defined in Tables 81-3 and 81-4) with the mapping to the 7-bit values used in the 66-bit blocks (last column) or if it is otherwise encoded (by the block type filed)

The footnote reserves other O codes:

- Fibre Channel: 0xF (for signal)
- OIF: 0x5 (for FlexE)
- ITU-T: 0xC (for MTN)

This is the only reference to these “other” O code values in the current PCS for 50+ GbE.

Comment #52 - Ordered Sets - CL 82 PCS: 66-bit block mapping

Old:

Input Data	Sync	Block Payload							
Bit Position: / Data Block Format:	0 1	2							65
$D_0 D_1 D_2 D_3 D_4 D_5 D_6 D_7$	01	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7
Control Block Formats:		Block Type Field							
$C_0 C_1 C_2 C_3 C_4 C_5 C_6 C_7$	10	0x1E	C_0	C_1	C_2	C_3	C_4	C_5	C_6	C_7
$S_0 D_1 D_2 D_3 D_4 D_5 D_6 D_7$	10	0x78	D_1	D_2	D_3	D_4	D_5	D_6	D_7
$O_0 D_1 D_2 D_3 Z_4 Z_5 Z_6 Z_7$	10	0x4B	D_1	D_2	D_3	O_0	0x000_0000		
$T_0 C_1 C_2 C_3 C_4 C_5 C_6 C_7$	10	0x87		C_1	C_2	C_3	C_4	C_5	C_6	C_7
$D_0 T_1 C_2 C_3 C_4 C_5 C_6 C_7$	10	0x99	D_0		C_2	C_3	C_4	C_5	C_6	C_7
$D_0 D_1 T_2 C_3 C_4 C_5 C_6 C_7$	10	0xAA	D_0	D_1		C_3	C_4	C_5	C_6	C_7
$D_0 D_1 D_2 T_3 C_4 C_5 C_6 C_7$	10	0xB4	D_0	D_1	D_2		C_4	C_5	C_6	C_7
$D_0 D_1 D_2 D_3 T_4 C_5 C_6 C_7$	10	0xCC	D_0	D_1	D_2	D_3		C_5	C_6	C_7
$D_0 D_1 D_2 D_3 D_4 T_5 C_6 C_7$	10	0xD2	D_0	D_1	D_2	D_3	D_4		C_6	C_7
$D_0 D_1 D_2 D_3 D_4 D_5 T_6 C_7$	10	0xE1	D_0	D_1	D_2	D_3	D_4	D_5		C_7
$D_0 D_1 D_2 D_3 D_4 D_5 D_6 T_7$	10	0xFF	D_0	D_1	D_2	D_3	D_4	D_5	D_6	

Figure 82–5—64B/66B block formats

New:

Input Data	Sync	Block Payload							
Bit Position: / Data Block Format:	0 1	2							65
$D_0 D_1 D_2 D_3 D_4 D_5 D_6 D_7$	01	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7
Control Block Formats:		Block Type Field							
$C_0 C_1 C_2 C_3 C_4 C_5 C_6 C_7$	10	0x1E	C_0	C_1	C_2	C_3	C_4	C_5	C_6	C_7
$S_0 D_1 D_2 D_3 D_4 D_5 D_6 D_7$	10	0x78	D_1	D_2	D_3	D_4	D_5	D_6	D_7
$O_0 D_1 D_2 D_3 D_4 D_5 D_6 D_7$	10	0x4B	D_1	D_2	D_3	O	D'_4	D_5	D_6	D_7
$T_0 C_1 C_2 C_3 C_4 C_5 C_6 C_7$	10	0x87		C_1	C_2	C_3	C_4	C_5	C_6	C_7
$D_0 T_1 C_2 C_3 C_4 C_5 C_6 C_7$	10	0x99	D_0		C_2	C_3	C_4	C_5	C_6	C_7
$D_0 D_1 T_2 C_3 C_4 C_5 C_6 C_7$	10	0xAA	D_0	D_1		C_3	C_4	C_5	C_6	C_7
$D_0 D_1 D_2 T_3 C_4 C_5 C_6 C_7$	10	0xB4	D_0	D_1	D_2		C_4	C_5	C_6	C_7
$D_0 D_1 D_2 D_3 T_4 C_5 C_6 C_7$	10	0xCC	D_0	D_1	D_2	D_3		C_5	C_6	C_7
$D_0 D_1 D_2 D_3 D_4 T_5 C_6 C_7$	10	0xD2	D_0	D_1	D_2	D_3	D_4		C_6	C_7
$D_0 D_1 D_2 D_3 D_4 D_5 T_6 C_7$	10	0xE1	D_0	D_1	D_2	D_3	D_4	D_5		C_7
$D_0 D_1 D_2 D_3 D_4 D_5 D_6 T_7$	10	0xFF	D_0	D_1	D_2	D_3	D_4	D_5	D_6	

Figure 82–5—64B/66B block formats

- The “generic data” in D4-D7 works with current definition of sequence ordered sets (for fault signaling).
- Other SDOs continue to define the details of the format and use of their control ordered sets using 0x4B block type with their assigned O code value. The other SDOs can continue to set D4-D7 as all zeros OR can now “legally” use those 3.5 bytes for data.

Comment #52 - Ordered Sets in Clause 46 RS and Clause 49 PCS (10GBASE-R)

Table 46–5—Sequence ordered sets

Lane 0	Lane 1	Lane 2	Lane 3	Description
Sequence	0x00	0x00	0x00	Reserved
Sequence	0x00	0x00	0x01	Local Fault
Sequence	0x00	0x00	0x02	Remote Fault
Sequence	0x00	0x00	0x03	Link Interruption
Sequence	≥ 0x00	≥ 0x00	≥ 0x04	Reserved

NOTE—Values in Lane 1, Lane 2, and Lane 3 columns are in hexadecimal, most significant bit to least significant bit (i.e., <7:0>). The link fault signaling state diagram allows future standardization of reserved Sequence ordered sets for functions other than link fault indications

Table 49–1—Control codes

Control character	Notation	XGMII Control Code	10GBASE-R Control Code	10GBASE-R O Code	8B/10B Code[a]
Sequence ordered set	/Q/	0x9C	Encoded by block type field plus O code	0x0	K28.4
Signal ordered set[d]	/Fsig/	0x5C	Encoded by block type field plus O code	0xF	K28.2

[a]For information only. The 8B/10B code is specified in Clause 36. Usage of the 8B/10B code for 10 Gb/s operation is specified in Clause 48.
[b]See 48.2.4.2.
[c]The codes for /A/, /K/, and /R/ are used on the XAUI interface to signal idle. They are not present on the XGMII when no errors have occurred, but certain bit errors cause the XGXS to send them on the XGMII.
[d]Used by INCITS T11 Fibre Channel.

In Clause 46&49, ordered sets are 4-byte values on XGMII (Sequence = 0x9C), and only pass 3 bytes of data into the 66-bit blocks. The O code is not transmitted on the xMII. O code of 0xF is used for “signal ordered sets” and is reserved for Fibre Channel (/Fsig/). The XGMII control code (0x5C) distinguishes it from the sequence ordered sets (0x9C) and allows the PCS to translate the XGMII values to the corresponding O code values.

802.3dj uses the 8-byte ordered set format which allows for the O code to be included on the xMII. 802.3dj uses the character 0x5C on the xMII for “control ordered sets” even though 0x9C could have been used since the O code value would distinguish them. 0x5C make it easier in the draft to talk about these “other” ordered sets. The xMII is an “internal interface and is not exposed for 50+ GbE.

Comment #52 - Ordered Sets - CL 49 PCS: 66-bit block mapping

Input Data	Sync	Block Payload							
Bit Position: / Data Block Format:	0 1	2							65
$D_0\,D_1\,D_2\,D_3/D_4\,D_5\,D_6\,D_7$	01	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7
Control Block Formats:		Block Type Field							
$C_0\,C_1\,C_2\,C_3/C_4\,C_5\,C_6\,C_7$	10	0x1E	C_0	C_1	C_2	C_3	C_4	C_5	C_6	C_7
$C_0\,C_1\,C_2\,C_3/O_4\,D_5\,D_6\,D_7$	10	0x2D	C_0	C_1	C_2	C_3	O_4	D_5	D_6	D_7
$C_0\,C_1\,C_2\,C_3/S_4\,D_5\,D_6\,D_7$	10	0x33	C_0	C_1	C_2	C_3		D_5	D_6	D_7
$O_0\,D_1\,D_2\,D_3/S_4\,D_5\,D_6\,D_7$	10	0x66	D_1	D_2	D_3	O_0		D_5	D_6	D_7
$O_0\,D_1\,D_2\,D_3/O_4\,D_5\,D_6\,D_7$	10	0x55	D_1	D_2	D_3	O_0	O_4	D_5	D_6	D_7
$S_0\,D_1\,D_2\,D_3/D_4\,D_5\,D_6\,D_7$	10	0x78	D_1	D_2	D_3	D_4		D_5	D_6	D_7
$O_0\,D_1\,D_2\,D_3/C_4\,C_5\,C_6\,C_7$	10	0x4B	D_1	D_2	D_3	O_0	C_4	C_5	C_6	C_7
$T_0\,C_1\,C_2\,C_3/C_4\,C_5\,C_6\,C_7$	10	0x87		C_1	C_2	C_3	C_4	C_5	C_6	C_7
$D_0\,T_1\,C_2\,C_3/C_4\,C_5\,C_6\,C_7$	10	0x99	D_0		C_2	C_3	C_4	C_5	C_6	C_7
$D_0\,D_1\,T_2\,C_3/C_4\,C_5\,C_6\,C_7$	10	0xAA	D_0	D_1		C_3	C_4	C_5	C_6	C_7
$D_0\,D_1\,D_2\,T_3/C_4\,C_5\,C_6\,C_7$	10	0xB4	D_0	D_1	D_2		C_4	C_5	C_6	C_7
$D_0\,D_1\,D_2\,D_3/T_4\,C_5\,C_6\,C_7$	10	0xCC	D_0	D_1	D_2	D_3		C_5	C_6	C_7
$D_0\,D_1\,D_2\,D_3/D_4\,T_5\,C_6\,C_7$	10	0xD2	D_0	D_1	D_2	D_3	D_4		C_6	C_7
$D_0\,D_1\,D_2\,D_3/D_4\,D_5\,T_6\,C_7$	10	0xE1	D_0	D_1	D_2	D_3	D_4	D_5		C_7
$D_0\,D_1\,D_2\,D_3/D_4\,D_5\,D_6\,T_7$	10	0xFF	D_0	D_1	D_2	D_3	D_4	D_5	D_6	

Figure 49–7—64B/66B block formats

In Clause 49 (XGMII) packets and control blocks are 4-byte aligned. Note: Ordered sets use only 3.5 bytes (0 code + 3 bytes data).

Two ordered sets can be stuffed into a single 66-bit block (0x55).

A single ordered set (in block type 0x4B) contains control characters (i.e. Idles) in C4-C7 (28 bits), since ordered sets are defined as 4-bytes on the XGMII.

XGMII control code of 0x9C can get mapped to several block types, but always uses O code 0x0. The block type may be 0x2D, 0x66, 0x55, or 0x4B.
Fibre Channel uses control code 0x5C (not 0x9C) on the MII so that a PCS used for FC can map 0x5C to the same block type and set the O code value to 0xF.

Comment #52 - Ordered Sets - CL 49 PCS: Ordered Set description

From Clause 49 (PCS for 10GBASE-R):

49.2.4.10 ordered set (/O/)

The ordered set control characters (/O/) indicate the start of an ordered set. There are two kinds of ordered sets: the sequence ordered set and the signal ordered set (which is reserved). When it is necessary to designate the control character for the sequence ordered set specifically, /Q/ will be used. /O/ is only valid on the first octet of the XGMII. Receipt of an /O/ on any other octet of TXD indicates an error. Block type field values implicitly encode an /O/ as the first or fifth character of the block. The 4-bit O code encodes the specific /O/ character for the ordered set.

Sequence ordered sets may be deleted by the PCS to adapt between clock rates. Such deletion shall only occur when two consecutive sequence ordered sets have been received and shall delete only one of the two. Only Idles may be inserted for clock compensation. Signal ordered sets are not deleted for clock compensation.

EMS ability/enable

Comment #7

Comment #7 - New functionality in old PCS/FEC clauses

- 802.3dt adds new functionality to old PCS and FEC clauses
  - The new functionality must be optional to implement (old implementation won't have it.)
  - But to support EMS, the new functionality is required.
- New functions:
  - **Use the new definition of 66-bit block type 0x4B to support control order sets.**
    - Needed for control ordered sets be able to use 6.5 bytes of data (not just 3 bytes of data and 3.5 bytes of all zero).
  - **Stateless 64B/66B coding in Clause 82 (same function that is optional in Clause 119)**
    - Can be used as an alternative to the state-diagram TX/RX coding that is fully interoperable with the state-diagram version.
  - **"FEC full error marking" in RS-FEC decoder for 50G & 100G RS-FEC clauses**
    - Needed for data integrity of metadata in preamble/SFD and control ordered sets
    - This is already how the RS-FEC decoder operates for 200GE+
  - **Ability to disable to disable the BER monitor in 50G & 100G PCS**
    - Required when optional "FEC full error marking is enable" is used in the RS-FEC

Comment #7 - New functionality in old PCS/FEC clauses in D1.0

- Current approach in D1.0 to describe new functions and enable them separately
  - For each new function:
    - Add a new “ability” MDIO status variable which is set if the new function is supported
      - User can query these ability bits to see if a device supports EMS
    - Add a new “enable” MDIO control variable if the new function takes the place of an existing functionality. For example, a device might implement both the old and new version of the function and the enable bit can be used to select between the old functionality and the new functionality
- The “set” of functions needed to support EMS, both new and existing options, for each family of xBASE-R PHYs are listed in a single list in a new PCS subclause for the xBASE-R family of PHYs instead of spreading out across multiple Clauses (PCS and FEC Clauses for 50G and 100G). Each PCS clause has a short subclause with a short list of the required functionality.
- This makes the new optional functionality available for new features (other than EMS) in any combination without major changes to the EMS-related text.
- A list of PHYs is not required in the Clause 195. New projects should not need to update CL 195.
- New implementations that want to support EMS do not need to implement the “old functionality.”

Comment #7 - New functionality in old PCS/FEC clauses as proposed

- Comment #7 (and other related comments) suggests an alternate approach
  - Define a set of 4 EMS ability/enable variables for the entire PHY
    - Separate ability/enable variables for link_metadata and packet_metadata
  - The EMS ability status variable is set only if all new functions are available
  - All new functions for EMS are enabled when the EMS enable variable is set by management.
  - Are separate MDIO variables needed for PCS and FEC and RS sublayers? (12 variables total)

- This is an “all or none” type of approach.

- Likely, does not make significant changes to the PICS - may change some dependencies.
  - All optional functionality is still needed.

- Requires implementations that want to support EMS to implement both old and functionality.

Comment #7 - status/control variable options

Feature	Current D1.0 MDIO variables	New proposal	Notes
Supports CtlOS	control_ordered_set_ability	ems_link_ability	Enable should not needed
Supports BER disable	BER_monitor_disable_ability BER_monitor_disable	(ems_link_ability or pcs_ems_packet_ability) (ems_link_enable or ems_packet_enable)	Enable only if FEC full error marking is enabled.
Supports PCS Stateless 64B/66B	PCS_64B/66B_stateless_ability PCS_64B/66B_stateless_enable	ems_link_ability ems_link_enable	Is already independent of metadata support for 200GE+. Would be nice for 50G/100G also.
Supports RS-FEC full error marking	FEC_full_error_marking_ability FEC_full_error_marking_enable	(ems_link_ability or ems_packet_ability) (ems_link_enable or ems_packet_enable)	A valid option on it's own. No need to tie it to EMS.

Comment #7 - Need a consensus decision on direction

- TF needs to decide on an approach to enabling the EMS functionality in the PCS and FEC.

    1. Enable each new optional function individually
    2. Enable all new optional functions together as a group based on EMS enabled

- This does not affect the optional functionality itself, just how it is enabled.

- Deciding this first will determine how to respond to many of the remaining comments and if changes are needed to the PICS.