Interpretation Number: 5-11/03
Topic: 10GBASE-X check_end function
Relevant Clause: 48.2.6.1.4
Classification: Unambiguous

Interpretation Request

Clause 48.2.6.1.4 of IEEE Std. 802.3ae-2002 defines the check_end function. The definition of this function is given here:

> Prescient Terminate function used by the PCS Receive process to set the RXD<31:0> and RXC<3:0> signals to indicate Error if a running disparity error was propagated to any Idle code-groups in ||T||, or to the column following ||T||. The XGMII Error control character is returned in all lanes less than n in ||T||, where n identifies the specific Terminate ordered-set ||Tn||, for which a running disparity error or any code-groups other than /A/or /K/are recognized in the column following ||T||. The XGMII Error control character is also returned in all lanes greater than n in the column prior to ||T||, where n identifies the specific Terminate ordered-set ||T n ||, for which a running disparity error or any code group other than /K/is recognized in the corresponding lane of ||T||. For all other lanes the value set previously is retained.

The first sentence clearly states that the purpose of this function is to catch errors that have propagated into idle code-groups either in ||T|| or in the column following ||T||. This also implies that the function does not intend to catch errors that could not possibly have propagated into the idle code-groups. There is no additional information in this sentence, and it is largely a description of the function.

The second sentence is: "The XGMII Error control character is returned in all lanes less than n in ||T||, where n identifies the specific Terminate ordered-set ||Tn||, for which a running disparity error or any code-groups other than /A/ or /K/ are recognized in the column following ||T||."

It is not clear from this sentence what the desired behavior should be. Please examine the following example with the /T/ contained in lane 2. In this example, * refers to a running disparity error or code-group other than /A/ or /K/. There are four cases shown in this example, each with the error occurring in a different lane. Each of the options listed below corresponds to the XGMII data after being received by the PCS.

For the second sentence of the check_end function, there are two interpretations over which frames get discarded, and two interpretations on which code-groups are changed to /E/. It is not clear from the reading of the function which interpretations are correct.

Input of PCS receiver

Case 1	Case 2	Case 3	Case 4
0 1 2 3	0 1 2 3	0 1 2 3	0 1 2 3
D D D D	D D D D	D D D D	D D D D
D D T K	D D T K	D D T K	D D T K
* K K K	K * K K	K K * K	K K K *

In the first option, the error control character is returned for each instance; therefore assuring that the frames will not be accepted. No matter which lane has the error, the error control character will be returned in all lanes less than *n*.

Option 1

0 1 2 3	0 1 2 3	0 1 2 3	0 1 2 3
D D D D	D D D D	D D D D	D D D D
E E T K	E E T K	E E T K	E E T K
E K K K	K E K K	K K E K	K K K E

In the second option, the error control character is only pushed back into the frame when the error occurs on a lane less than *n*. When the error occurs in lanes 2 or 3, it is not necessary to push the error back since these errors could not have been propagated through the frame. This allows the frames to be accepted when the error occurs in lanes 2 or 3. Also in this option, when the error is pushed back into the frame, all lanes less than *n* receive the error control character.

Option 2

0 1 2 3	0 1 2 3	0 1 2 3	0 1 2 3
D D D D	D D D D	D D D D	D D D D
E E T K	E E T K	D D T K	D D T K
E K K K	K E K K	K K E K	K K K E

In the third option, the error control character is only pushed back into the frame when the error occurs on a lane less than *n*. In this option, the error control character is only returned in those lanes, which actually had the error, and not in any other lanes.

Option 3

0 1 2 3	0 1 2 3	0 1 2 3	0 1 2 3
D D D D	D D D D	D D D D	D D D D
E D T K	D E T K	D D T K	D D T K
E K K K	K E K K	K K E K	K K K E

Which of the above interpretations (1, 2, or 3) is correct?

The original intent of the check_end function was that potentially valid frames not be invalidated by the PCS. When an error occurs in a column directly following a valid /T/,

/K/, or /A/ codegroup, then the error could not have propagated through data code-groups in the frame, as the /T/, /K/, and /A/ code-groups will effectively prevent errors such as running disparity errors from going through them. Since the error could not possibly have occurred within the data portion of the frame, there should be no need for the PCS to invalidate the frame. One possible interpretation of the current text is that such action should be taken by the PCS.

It is not clear what the original intent of the check_end function was with respect to replacing data code-groups with error code-groups. The replacement of a single data code-group with an error code-group is sufficient to force the frame to be discarded. Also, since the lanes are independent of each other, it is not possible for an error to propagate from one lane to another. Although the insertion of multiple error code-groups will have the same result as the insertion of a single error code-group, it is possible that certain error counters may be caused to increment needlessly.

A survey of 4 different vendors showed at least 3 different interpretations of the check_end function. One vendor follows option 1 and discards frames that are otherwise valid. A different vendor follows option 2 and allows those frames to be received. A third and fourth vendor have implemented yet another interpretation, one that allows frames which should be discarded to be accepted. It seems clear that the current wording of the check_end function appears to be overly complicated and is easily given to misunderstanding and misinterpretation. Based on the outcome of this request, we are prepared to submit a maintenance request to clarify the wording of the function so that future implementations may benefit.

Summarizing the tables from above:

Case 1	Case 2	Case 3	Case 4
0 1 2 3	0 1 2 3	0 1 2 3	0 1 2 3
D D D D	D D D D	D D D D	D D D D
D D T K	D D T K	D D T K	D D T K
* K K K	K * K K	K K * K	K K K *

Option 1

0 1 2 3	0 1 2 3	0 1 2 3	0 1 2 3
D D D D	D D D D	D D D D	D D D D
E E T K	E E T K	E E T K	E E T K
E K K K	K E K K	K K E K	K K K E

Option 2

0 1 2 3	0 1 2 3	0 1 2 3	0 1 2 3
D D D D	D D D D	D D D D	D D D D
E E T K	E E T K	D D T K	D D T K
E K K K	K E K K	K K E K	K K K E

Option 3

0 1 2 3	0 1 2 3	0 1 2 3	0 1 2 3
D D D D	D D D D	D D D D	D D D D
E D T K	D E T K	D D T K	D D T K
E K K K	K E K K	K K E K	K K K E

The standard states in subclause 48.2.6.1.4 that "The XGMII Error control character is returned in all lanes less than n in ||T||, where n identifies the specific Terminate ordered-set ||Tn||, for which a running disparity error or any code-groups other than /A/or /K/are recognized in the column following ||T||. The XGMII Error control character is also returned in all lanes greater than n in the column prior to ||T||, where n identifies the specific Terminate ordered-set ||T n ||, for which a running disparity error or any code group other than /K/is recognized in the corresponding lane of ||T||. For all other lanes the value set previously is retained."

The correct option is therefore 3.

In Cases 1 to 4 of the "Input to PCS receiver" shown in the request, n=2 as the /T/ appears in lane 2. Therefore all lanes less than n in these examples mean lanes 0 and 1.

In 'Case 1' shown an error appears in lane 0 which meets the condition of an error in lane less than n. Therefore an error is inserted in the ||T|| for lane 0 because it meets the condition of a disparity error in a lane less than n with a running disparity error in the

following column. No error is inserted in lane 1 because there is no error in the following column.

In case 3 a error appears in lane 2 which does not meet the condition of an error in lane less than n. No errors are inserted.

Interpretation Number: 2-03/05
Topic: 10GBASE-LX4 skew definition
Relevant Clauses: 48.2.4.2.2, 53.1
Classification: Unambiguous

Interpretation Request

Table 48-5 of clause 48.2.4.2.2 in the IEEE802.3ae-2002 standard specifies the skew budget of the PMA sublayer for 10GBASE-X transponder types. Potential sources for skew are PMA TX, PCB, Medium, and PMA RX. The total skew budget is stated to be <41UI.

With respect to Fig. 44A-5 and Fig. 44A-6 it is now unclear if this specification can be applied to the transponders PMD side as well, and if, how the values given in Table 48-5 would then be translated to the Test-Point (TP) terminology that is used in clause 53.1. In detail, a clear statement what the allowed skew values at TP[1:4] (ref. to Fig. 53.2 of clause 53.4.1) are is missing.

Interpretation for IEEE Std 802.3ae-2002

Per Table 48-5, allowed skew at TP2 is 3 UI. The medium is allocated 18 UI, giving an allowed skew at TP3 of 21 UI. TP1 and TP4, shown in Figure 53-2, are provided for reference only. The specification of performance at TP1 and TP4 is implementation specific.