802.3dj D2.1 Comment Resolution Logic Track

Gary Nicholl (Cisco), Logic Track Lead Editor
Matt Brown (Alphawave Semi), 802.3dj Chief Editor
Eugene Opsasnick (Broadcom), Logic 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 various common-track comments.

PHY delay

Comment #250

PHY delay - Clause 31B and 174 (for 1.6TE)
Comment # 250

Annex 31B

(normative)

MAC Control PAUSE operation

31B.3 Detailed specification of PAUSE operation

31B.3.7 Timing considerations for PAUSE operation

Insert the following paragraph into 31B.3.7 (as amended by IEEE Std 802.3df-2024) after the "At operating speeds of 800 Gb/s, …") paragraph:

At operating speeds of 1.6 Tb/s, a station shall not begin to transmit a (new) frame more than 3620 pause_quanta after the reception of a valid PAUSE frame that contains a non-zero value of pause_time, as measured at the MDI.

Pause response time req'd: 3620 pause quanta.

Adding Sublayer Delay Constraints:
For CR8: 768 (MAC) + 2500 (PCS) + 48 (PMA) + 232 (PMD) = 3548
For DR8-2: 768 + 2500 + 48 + 232 + 270 (InnerFEC) = 3818

Table 174–4—Sublayer delay constraints

Sublayer	Maximum (bit time)[b]	Maximum (pause_quanta)[c]	Maximum (ns)	Notes[a]
1.6T MAC, RS, and MAC Control	393 216	768	245.76	See 170.1.4.
1.6TBASE-R PCS or 1.6TXS[d]	1 280 000	2500	800	See 175.5.
1.6TBASE-R 8:16 or 16:8 PMA	24 576	48	15.36	See 176.8.
1.6TBASE-R 8:8 or 16:16 PMA	24 576	48	15.36	See 176.8.
1.6TBASE-R Inner FEC	138240	270	86.4	See 177.8.
1.6TAUI-4 C2C or C2M component	36 864	72	46.08	Includes allocation of 5 ns for one direction through AUI channel. See 176C.5 and 176D.5.
1.6TBASE-KR8 PMD	118 784	232	74.24	Includes allocation of 14 ns for one direction through the backplane medium. See 178.6.
1.6TBASE-CR8 PMD	118 784	232	74.24	Includes allocation of 14 ns for one direction through the cable medium. See 179.6.
1.6TBASE-DR8 PMD	118 784	232	74.24	Includes 2 m of fiber. See 180.4.1.
1.6TBASE-DR8-2 PMD	118 784	232	74.24	Includes 2 m of fiber. See 182.4.1.

[a] Should there be a discrepancy between this table and the delay requirements of the relevant sublayer clause, the sublayer clause prevails.
[b] For 1.6TBASE, 1 bit time is equal to 0.625 ps. (See 1.4.215 for the definition of bit time.)
[c] For 1.6TBASE, 1 pause_quantum is equal to 320 ps. (See 1.4.459 for the definition of pause_quanta.)
[d] If an implementation includes the 1.6TMII Extender, the delay associated with the 1.6TMII Extender includes two 1.6TXS sublayers.

State Diagram Variable Definitions

(comments 118, 119, 120, 121, 122, 123, and others)

State diagram variable definitions
Comments 118-123 plus many in bucket #1.

- Many comments submitted against the state variable definitions
  They fall into three general categories:
  1. The definition describes when a boolean variable is set to true but not when set to false.
  2. Boolean variable definitions wording: “set to true/false”, not “asserted/deasserted”.
  3. It is unclear if the definition is complete or if a state diagram also sets the variable value.

- Editor’s Observations:
  - Case A: Some variable definitions add a description of the variable usage, but they are set to values completely within the state diagram - the state diagram prevails.
  - Case B: Some variables are not set at all in the state diagram (e.g. only used for state transitions). These should be (but are not always) completely specified by the definition text
  - Case C: Some variables are set to values in the description and to the same or to other values by the state diagram.

→ ***It is hard to tell when reading the variable definitions if the definition is complete or if the state diagram also sets its value or if the text exactly matches what the state diagram does (but the state diagram always prevails).***

State diagram variable definitions - update approach for better clarity

- If the variable value is set by a state diagram:
  - Remove “... is set to <*a value*>” wording in the description and refer the state diagram for how the value is set with a cross-reference to the state diagram figure.
  - Keep descriptions that help user understand the intent of the variable usage.
- If the variable value is not set by a state diagram:
  - Use “... is set to <*a value*>” in the definition itself. Be sure to cover possible values, e.g. may need to add “Otherwise, the variable is set to false.”
- Remove wording where the same values are assigned in text and as are assigned in the state diagram.
- Limit the times (case C) when some values are set by definition and other values are set by the state diagram. But, if necessary, be explicit about what it is.

State diagram variable definitions - Case A

variable set by state diagram

<Variable Type> variable that <High level description>. The value of <variable> is set by the <name of FSM> diagram (see Figure X).

Examples:

training_failure

Boolean variable that is true when training failed to complete. The value of training_failure set by the training control state diagram (see Figure 178B-10).

align_status_mux

Boolean variable that is true when the alignment marker lock and deskew processes are complete. Its value is set by the PMA multiplex synchronization state diagram (see Figure 176-10).

State diagram variable definitions - Case B

variable set by definition

<Variable Type> variable that is set to <a value> <description of how the variable takes on ALL of its possible values>.

Examples:

local_tf_lock
: Boolean variable that is set to true when mr_training_enable is true and the training frame marker positions have been identified on a lane of the interface and is set to false otherwise.

all_locked_demux
: Boolean variable that is set to true when pmal_locked_demux<y> is true for all y, where y = 0 to (n - 1), which indicates all PCS lanes within all PMA lanes have achieved alignment marker lock. Otherwise, this variable is set to false.

State diagram variable definitions - Case C

variable set by state diagram and definition

<Variable Type> variable that is set to <values> by the <name of FSM> diagram (see Figure X) and set to <description of when it take on other values>. <High level description>.

Examples:

new_marker

Boolean variable that is set to false by the training control state diagram (see Figure 178B-10) in the TEST_MARKER state and set to true when a new candidate frame marker is available for testing. A new frame marker is available for testing when the training frame lock process has accumulated 32 consecutive symbols starting at the candidate frame start position.

slip_done

Boolean variable that is true when the next candidate 128-bit block position can be tested by the Inner FEC self-synchronization process. It is set to true when the SLIP function completes and is set to false upon entering the GET_BLOCK state of the Inner FEC self-synchronization state diagram (see Figure 177-12).

State diagram variable definitions - Some common exceptions & variations

- The *reset* variable does not have an exact “set to value” definition but is used almost all state diagrams.

  Example:

  reset

  Boolean variable that controls the resetting of the PCS sublayer. It is true whenever a reset is necessary including when PCS_reset is true and during power on, otherwise it is false.

- A counter (or timer) is initialized when a “start counter” statement occurs in a state which can include resetting a counter_*done* variable that is later set when the counter has reached its “terminal value”. The terminal value is usually defined by the function definition. (Some counters just count without a counter_*done* variable and the value is set and tested in the state diagram.)

  Example (see Figure 177-12 on page 358 of D2.1):

Variable → fas_cnt_done

Boolean variable that is set to true when the counter fas_cnt has reached its terminal count and is set to false when starting the counter (see Figure 177-12).

Counters → fas_cnt

Counts the number of Inner FEC codewords between two adjacent pads as defined in 177.4.7.

bad_fas_cnt

Counts the number of consecutive FAS that do not match the expected values.