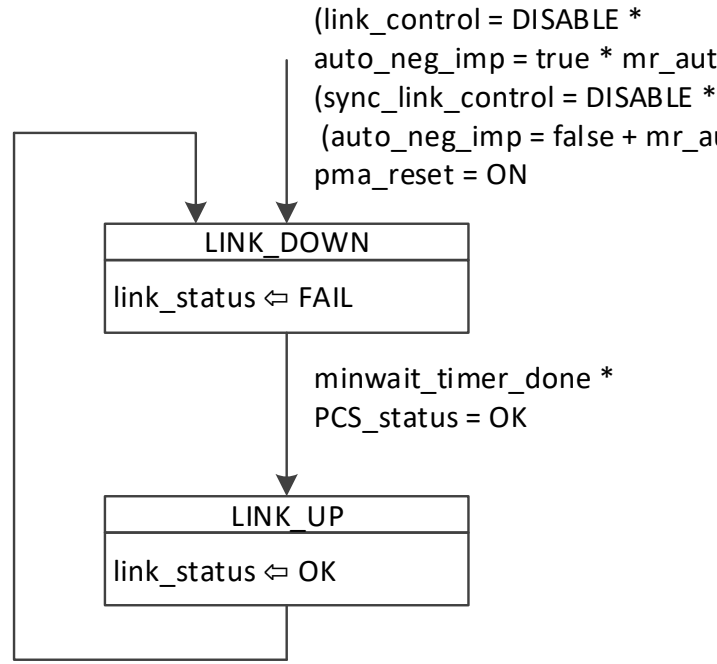
Draft 1.1 Proposed Comment Details

William Lo

1) Figure 149-32 change in red



PMA_watchdog_status
Variable indicating the status of the PAM4 monitor.
Values:

OK: the local device has received sufficient transitions
NOT_OK: the local device has not received sufficient transitions

During normal operation NOT_OK is assigned when no transitions are seen on the line for longer than 2us +/- 0.1us

During Low Power Idle operation NOT_OK is assigned when no transitions are seen on the line during the refresh window

2) Ignore OAM parity symbols except during low power idle refreshes

- Current the OAM frame symbols are protected by the RS(360, 326). The RS(16, 14) to calculate/check the parity is redundant except when used during refresh cycles.
- The RS(16, 14) is unnecessary circuitry for PHYs that does not implement EEE.
- Recommend transmitting the actual values of the parity symbols be optional and the parity checks not be performed at the receiver if all OAM symbols in the OAM frame are embedded in the RS(360, 326) frame.

Required text changes

149.3.8.4.5 – Replace with the following text

rs (14 OAM symbols)

This function outputs the 2 parity symbols as defined in 149.3.8.2.13.

If all 16 symbols of the OAM frame are embedded in tx_RSmessage<9:0> as described in 149.3.2.2.15 then this function can return any arbitrary value.

rs_correct (16 OAM symbols)

This function outputs the 14 corrected data symbols as defined in 149.3.8.2.13 sets/clears the rs_check variable when at least one of the 16 symbols of the OAM frame is embedded in a LPI refresh.

Otherwise this function returns the 14 data symbol unmodified and sets rs_check to GOOD if none of the 16 RS(360, 326) frames are uncorrectable.

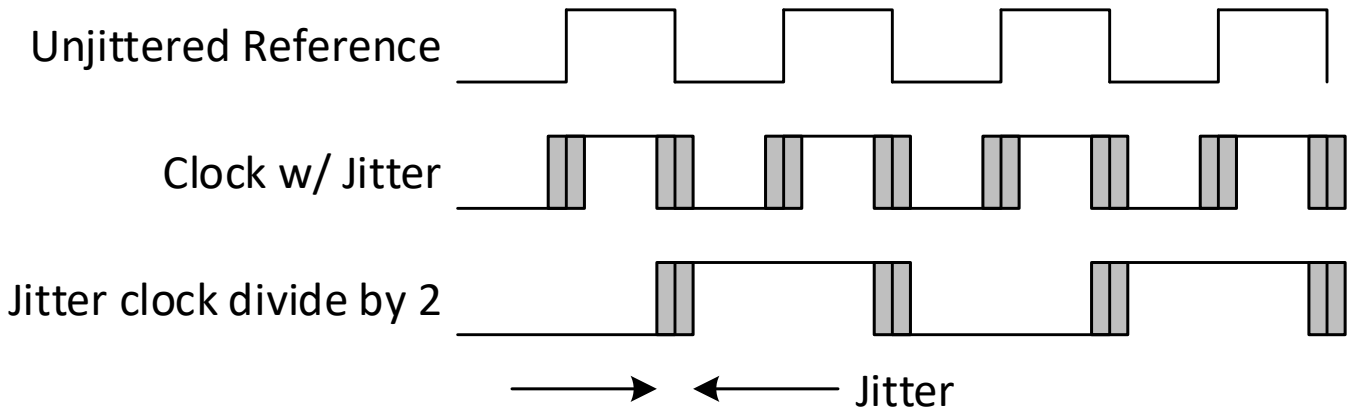
149.3.8.2.13 – Add the following at the beginning of the clause

The OAM Reed Solomon parity symbol generation/correction is required only when the EEE is implemented. When all 16 symbols of the OAM are embedded in tx_RSmessage<9:0> as described in 149.3.2.2.15 then OAM<15:14><9:0> are dummy symbols and are ignored at the receiver.

3) TX_TCLK_DIV (Test mode 1) 149.5.1

- The baud rate is at 10G is TX_TCLK = 5625MHz. Currently TX_TCLK_DIV = TX_TCLK / 16.
- Transmit jitter is measured at TX_TCLK_DIV
- Question: Will dividing by 32 be ok so we can use a slower speed I/O to output TX_TCLK_DIV
- Answer: Yes. Jitter is the same after clock division. Only difference is half the edges disappear.

Graphically – infinite persistence triggering on unjittered reference:



- Recommendation: Divide by 32 or even 64.

THANK YOU