Supporting materials for comments 87 and 88

Jeff Slavick

BROADCOM

Comment #87

Cl **136** *SC* **136.8.11.6** *P* **203** *L* **28** # 87

Slavick, Jeff — Broadcom Limited

Comment Type **T** *Comment Status* **X**

The definition for a request is solely based on the control field changing. We added a parity bit in D1.2, and don't preculde designs from ignoring frames with invalid parity (you're allowed to ignore it if you want). So I think the timing now needs to account for the parity bit being validly set as well.

SuggestedRemedy

Change "A new request is defined to be a received training frame whose control field differs from the control field of the precedingtraining frame."
to "A new request is defined to be a received training frame whose control field differs from the control field of the preceding training frame and the received parity bit is properly set."
Since the acknowledgement already states "status field encoding" I think that covers parity transmission.

The issue

136.8.11.3.5 Parity bit

The parity bit is calculated based on the other bits in the control field and status field to create even parity for these fields. Even parity ensures that the resulting pattern is DC balanced. This field may be ignored on receipt.

- The parity bit **MAYBE** ignored on receipt.
  - This mean a design is allowed to check this bit for validity.
  - The only reason to check the bit is to use it as a qualification for parsing the other bits. A failed parity bit check would cause the design to ignore the control and status field that was received.
- Since designs are allowed to use the parity bit to affect their receive operations we should account for it. The only place this affects is the 2ms response time requirement.

Response time

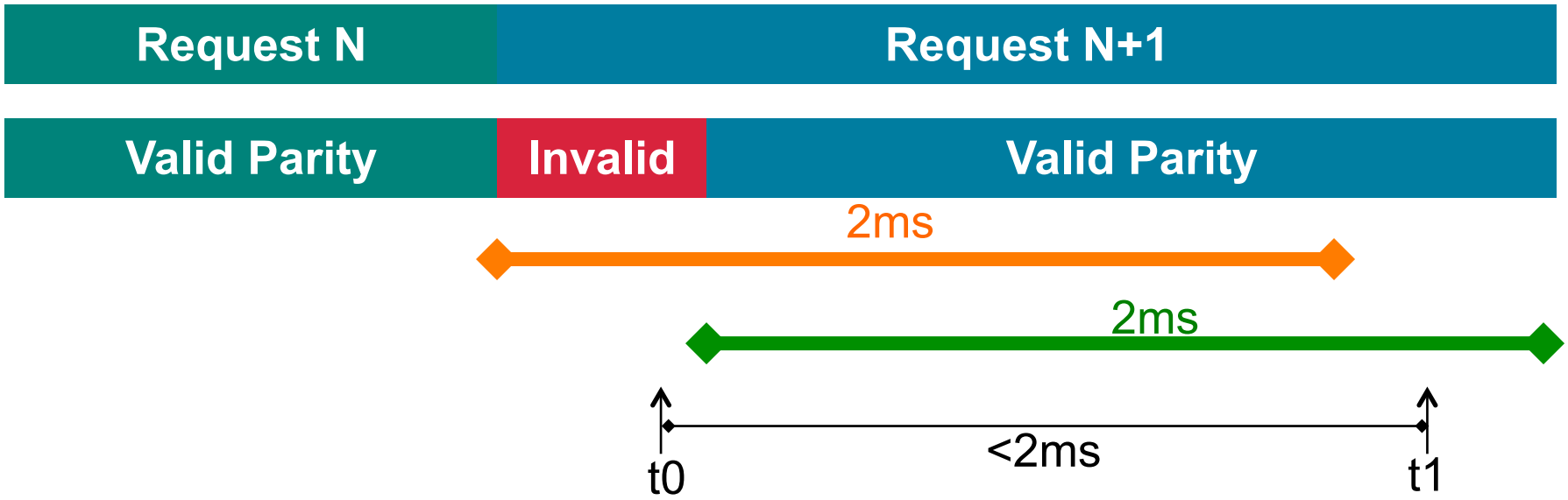
136.8.11.6 Handshake timing

Changes to the configuration of the transmitter equalizer or the transmitted training pattern may occur at any point between the time the request for the new configuration is received and the time that the request is acknowledged.

The time from the receipt of a new request to the time that request is acknowledged shall be less than 2 ms when the receiver frame lock bit in the status field of transmitted training frames is set to 1. A new request is defined to be a received training frame whose control field differs from the control field of the preceding training frame. An acknowledgment is defined as the first transmitted training frame that contains a status field encoding that is an appropriate response for the requested action. For example, a change in the coefficient select bits in the control field would be acknowledged by a change in the coefficient select echo bits of the status field. Similarly, a change in the modulation and precoding request bits in the control field would be acknowledged by a change in the modulation and precoding status bits in the status field. All timing measurements are referred to a common reference point within the training frame (e.g., the 3 to 0 transition in the training frame marker).

- But the parity bit is sent in the Status Field.

Example issue



- Design is checking the parity bit for validity, at time t0 it's invalid so it doesn't process Request N+1until t1. The interval between t1 and t2 time is less then required 2ms.
- However if 2ms window begins when Control field update is received, then it would violate the response time. If the 2ms window starts when both Control and Status field are valid, then it'd be compliant.

Why is the parity bit invalid?

- One possibility is the transmitter sends wrong value.
- Another would be the receiver has a data error

- In either case, making the specification clear on how to deal with this situation removes any ambiguity.

Solutions

- Follow suggested remedy
  - Add in the parity bit to qualify the start of the 2ms window
- Alternate approach would be
  - Change the parity bit to be ignored on receipt. (convert may be to is)

Comment #88

Cl **136** *SC* **136.8.11.7.1** *P* **205** *L* **12** # 88

Slavick, Jeff — Broadcom Limited

Comment Type **T** *Comment Status* **X**

remote_rx_rdy is a direct mirror of the status bit received in the training frames. In clause 72 this variable is only updated to TRUE when 3 consecutive training frames with the status bit are received.

SuggestedRemedy

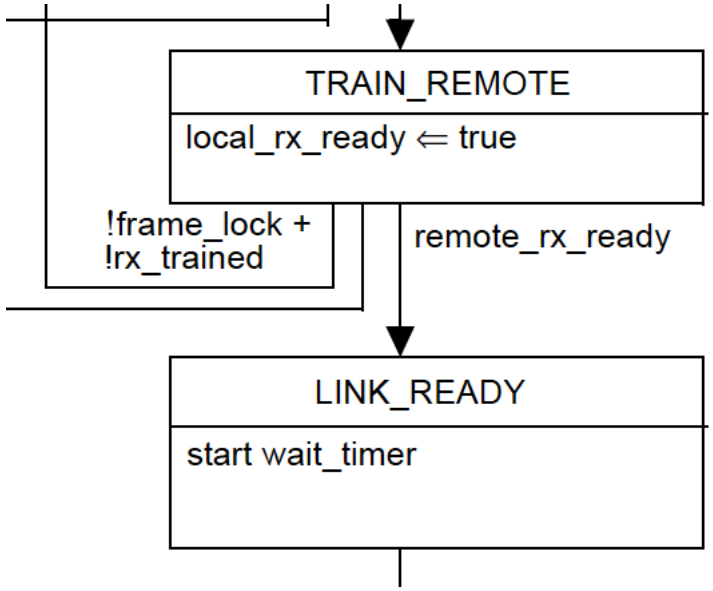
Change remote_rx_rdy and remote_tf_lock to be set to TRUE once 3 consecutive training frames are received with the appropriate field set.

Clause 72

remote_rx_ready

Boolean variable that is set to FALSE upon entry into the SEND_TRAINING state. The value of remote_rx_ready shall not be set to TRUE until no fewer than three consecutive training frames have been received with the receiver ready bit asserted.

- In all previous generation PMDs we wait on getting 3 frames from the remote end indicating it's "done" before beginning the shutdown process



BROADCOM

What does the 3x do?

- During a Link Train process you will see one side end first (begin sending receiver ready) and it sits there waiting for the other end to complete.
- The three consecutive receptions causes
  - A) Prevents a single mis-decode of the status from causing the “already done” side to begin shutdown early
  - B) Allows for the “slower” side time to enter/transition through the same state

Why remote_tf_lock

- remote_tf_lock is used to start-up the 1.5s timeout timer.
- During the SEND_TF state, the amount of equalization applied to the channel maybe minimal. So ensuring that the other end is sending you a tf_lock status before you start the 1.5s timeout timer provides a more robust start-up process.

Conclusion

- Adding the 3x qualification to remote_rx_rdy and remote_tf_lock provides a more tolerant receiver operation.