Comments 195, 196, 197, 198: Link Sync State Diagram

G. Zimmerman / CME Consulting

2/25/2026

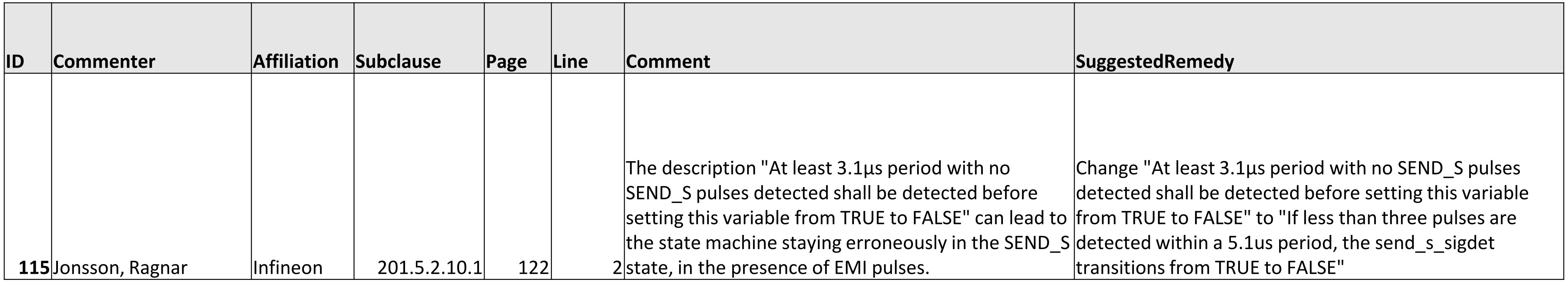
Revised 3/9/2026

NOTE: Comment 115 also has bearing on this

Comments

ID	Commenter	CommenterCo	Subclause	Page	Line	Comment	SuggestedRemedy
195	Zimmerman, George	CME Consulting/ADI,APL Gp, Cisco, Infineon, OnSemi, Sony	201.5.2.10	120	4	This section describes the link synchronization, but lacks the simple requirement that the state diagram (Figure 201-26) shall be followed.	Change "shall establish the start of PHY PMA training as defined in 201.5.2.7." to "shall conform to the state diagram in Figure 201-26. This section describes and defines the function of Figure 201-26, the link synchronization process."
197	Zimmerman, George	CME Consulting/ADI,APL Gp, Cisco, Infineon, OnSemi, Sony	201.5.2.10	120	40	The 5th paragraph here has shalls which duplicate what is in the state diagram. They should be replaced by descriptive text.	Replace "shall output" at lines 42 and 44 with "outputs" Replace "shall repeat" at line 45 with "repeats" Replace "shall stop outputting and enter" at lines 46-47 with "stops outputting and enters" Replace "shall also enter" at line 48 with "enters"
196	Zimmerman, George	CME Consulting/ADI,APL Gp, Cisco, Infineon, OnSemi, Sony	201.5.2.10	120	41	The text here includes a delay for the FOLLOWER sending out a SEND_S pulse that conflicts with the state diagram. Assuming this is the intended behavior, the state diagram needs to be corrected.	**SEE THIS PRESENTATION**
198	Zimmerman, George	CME Consulting/ADI,APL Gp, Cisco, Infineon, OnSemi, Sony	201.5.2.10.1	122	1	The conditions for what TRUE and FALSE mean are contained as 'shalls' in the variable description for send_s_sigdet. Since the state diagram is one 'shall' this makes it a redundant requirement, and isn't good style.	Delete Text (2 sentences) from "At least 3 consecutive… to "setting this variable from TRUE to FALSE". Change definitions of TRUE and FALSE to: TRUE: At least 3 consecutive valid SEND_S pulses have been detected. FALSE: No SEND_S pulses have been detected for at least 3.1 us.

Related Comments

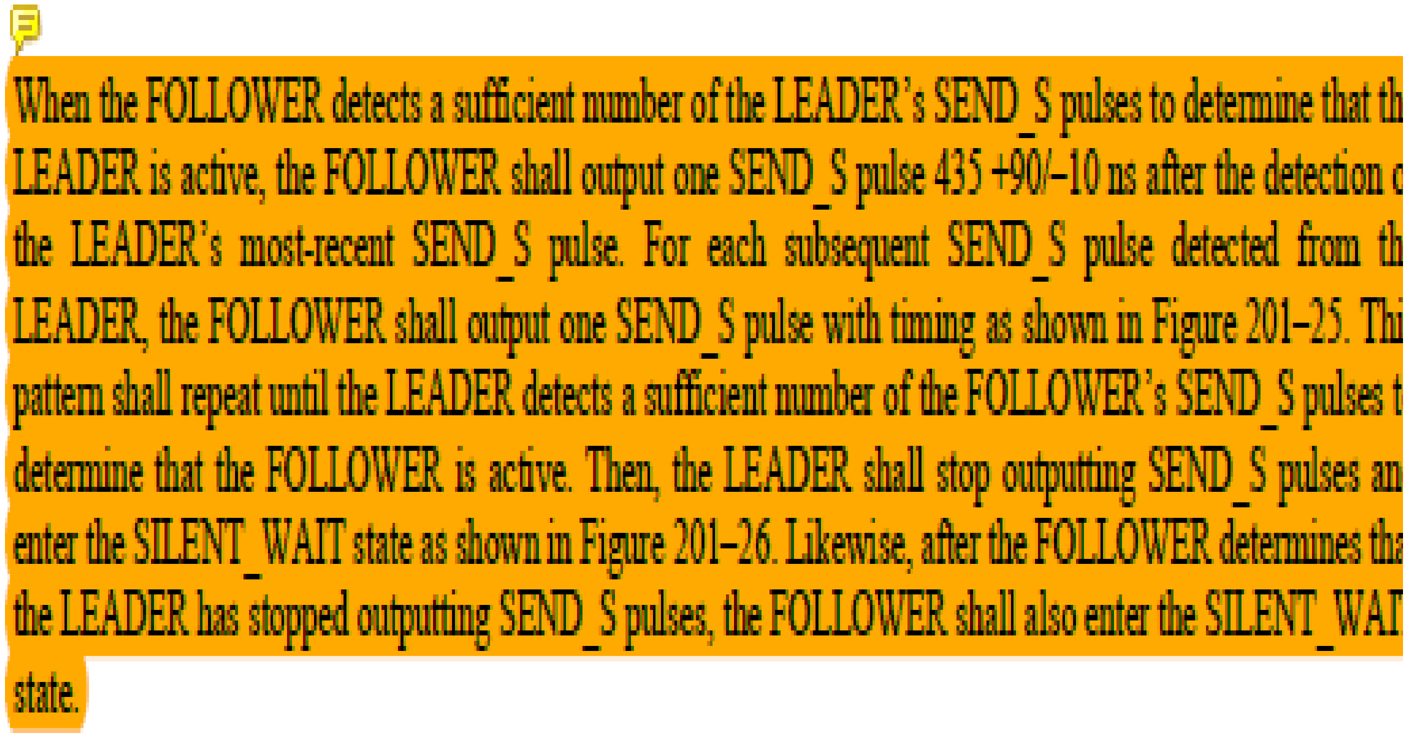


ID	Commenter	Affiliation	Subclause	Page	Line	Comment	SuggestedRemedy
115	Jonsson, Ragnar	Infineon	201.5.2.10.1	122	2	The description "At least 3.1µs period with no SEND_S pulses detected shall be detected before setting this variable from TRUE to FALSE" can lead to the state machine staying erroneously in the SEND_S state, in the presence of EMI pulses.	Change "At least 3.1µs period with no SEND_S pulses detected shall be detected before setting this variable from TRUE to FALSE" to "If less than three pulses are detected within a 5.1us period, the send_s_sigdet transitions from TRUE to FALSE"

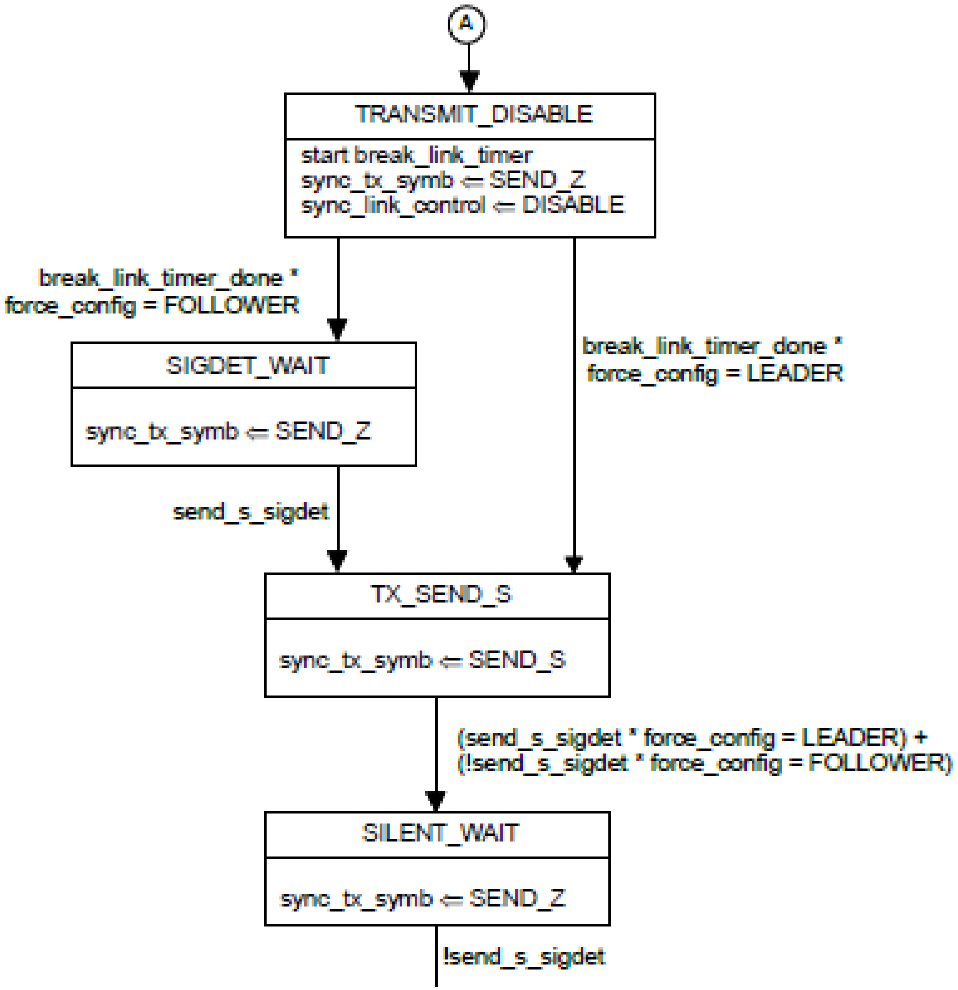
Problem

- 201.5.2.10 contains both a textual description of the Link Sync Process and the formal description of the State Diagram
    - They do not describe the same behaviour for the progress of detecting the link_sync until entry to PAUSE
- The text describes its behavior with “shalls”, which are duplicative of the state diagram’s requirements
- SEND_S is described both as a pulse and as a periodic pulse train
    - The use is different for the LEADER and the FOLLOWER…
- The behavior of the key variable, send_s_sigdet, is defined to be dependent on it’s state – mixing definition and state diagram

Text and State Diagram



When the FOLLOWER detects a sufficient number of the LEADER’s SEND_S pulses to determine that th
LEADER is active, the FOLLOWER shall output one SEND_S pulse 435 +90/–10 ns after the detection c
the LEADER’s most-recent SEND_S pulse. For each subsequent SEND_S pulse detected from th
LEADER, the FOLLOWER shall output one SEND_S pulse with timing as shown in Figure 201–25. Thi
pattern shall repeat until the LEADER detects a sufficient number of the FOLLOWER’s SEND_S pulses t
determine that the FOLLOWER is active. Then, the LEADER shall stop outputting SEND_S pulses an
enter the SILENT_WAIT state as shown in Figure 201–26. Likewise, after the FOLLOWER determines tha
the LEADER has stopped outputting SEND_S pulses, the FOLLOWER shall also enter the SILENT_WAI
state.


A
TRANSMIT_DISABLE
start break_link_timer
sync_tx_symb ⇐ SEND_Z
sync_link_control ⇐ DISABLE
break_link_timer_done * force_config = FOLLOWER
break_link_timer_done * force_config = LEADER
SIGDET_WAIT
sync_tx_symb ⇐ SEND_Z
send_s_sigdet
TX_SEND_S
sync_tx_symb ⇐ SEND_S
(send_s_sigdet * force_config = LEADER) +
(!send_s_sigdet * force_config = FOLLOWER)
SILENT_WAIT
sync_tx_symb ⇐ SEND_Z
!send_s_sigdet

To PAUSE

Behavior: Text vs. Diagram

TEXT

- Leader sends periodic SEND_S
- Follower detects “a sufficient number” of pulses
- Follower outputs a single SEND_S pulse with delay
- Follower then outputs a single SEND_S pulse (with delay) for each subsequent pulse
- When Leader detects “a sufficient number” of follower’s pulses, it enters SILENT_WAIT state & goes silent
- Follower detects Leader’s silence (unspecified criterion) and moves on to SILENT_WAIT & goes silent
- Text does not describe further behavior (progress to PAUSE)

DIAGRAM

- Leader sends SEND_S with each transmit clock
  - No mention of silence then repeating, is SEND_S a pulse or a sequence of pulses?
  - Comment 113 would have “SEND_S signal” & “SEND_S pulse”
- Follower silently waits to detect 3 consecutive SEND_S pulses
- Follower then begins sending SEND_S
  - Overrides delay mentioned in text, and if SEND_S is a sequence of pulses, this differs from text
- When Leader detects 3 consecutive SEND_S pulses, enters SILENT_WAIT state & goes silent
- When Follower misses the first SEND_S pulse from the Leader, it enters SILENT_WAIT and goes silent
- Follower will fall straight through SILENT_WAIT to PAUSE
- Leader will proceed to PAUSE when it misses it’s first Follower SEND_S pulse

Desired Behavior and Consequences

- I believe we want the behavior in the text, particularly
  - Follower first detects ‘a sufficient number’ of Leader pulses
  - Follower responds with a single pulse to each subsequent pulse from the Leader, pulse by pulse
  - SEND_S signal detections (send_s_sigdet) is more robust than just consecutive pulses – “a sufficient number”
  - Detecting silence from the partner is more robust than missing a single pulse
- This means:
  - SEND_S is a pulse (contrary to comment 113)
  - The state diagram needs to generate the periodic pulse train at the Leader
  - We either need a new variable to detect silence, or else states in the diagram to describe the behavior of send_s_sigdet

Comment 115

- Comment 115 changes the transition condition for detecting that the link partner as transitioned to SEND_Z to detecting a specific number of pulses within a specific time.
- This removes the issue where a single missed pulse can cause a link partner to advance to the next state (or fall through a state)
- The suggested resolution of comment 115 is incorporated into the solution proposed here

Proposed solution

- Add a new variable for detecting silence (incorporating comment 115) & modify send_s_sigdet
  - **quiet_detect**: A Boolean variable that indicates whether the link partner has ceased sending SEND_S pulses.

    VALUES:
    TRUE: No SEND_S pulses have been detected at the specified times (for LEADER or FOLLOWER see Figure 201-25) within the preceding 5.1 usec.
    FALSE: At least one send_s pulse has been detected at the specified times (for LEADER or FOLLOWER see Figure 201-25) within the preceding 5.1 usec
  - **send_s_sigdet**: A Boolean variable indicating whether sufficient SEND_S pulses have been detected. The definition of a sufficient number of pulses is implementation dependent.

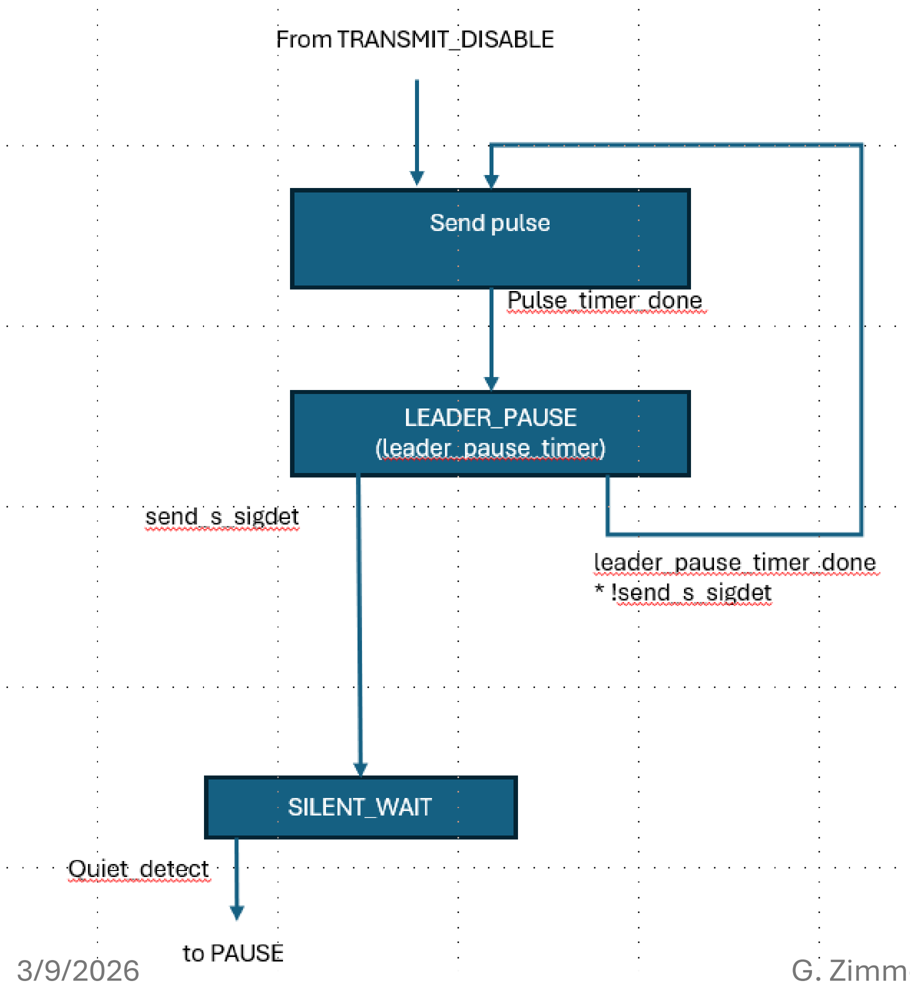
    VALUES:

    TRUE: Sufficient SEND_S pulses have been detected

    FALSE: Sufficient SEND_S pulses have not been detected
- Other new variables: single_send_s_detect
- Add new timers: follower_delay_timer, leader_pause_timer, pulse_timer
- Delete “shalls” in the descriptive text of 201.5.2.10

Proposed diagram: TRANSMIT_DISABLE exit to PAUSE entry

Leader



From TRANSMIT_DISABLE

Send pulse

Pulse_timer_done

LEADER_PAUSE
(leader_pause_timer)

send_s_sigdet

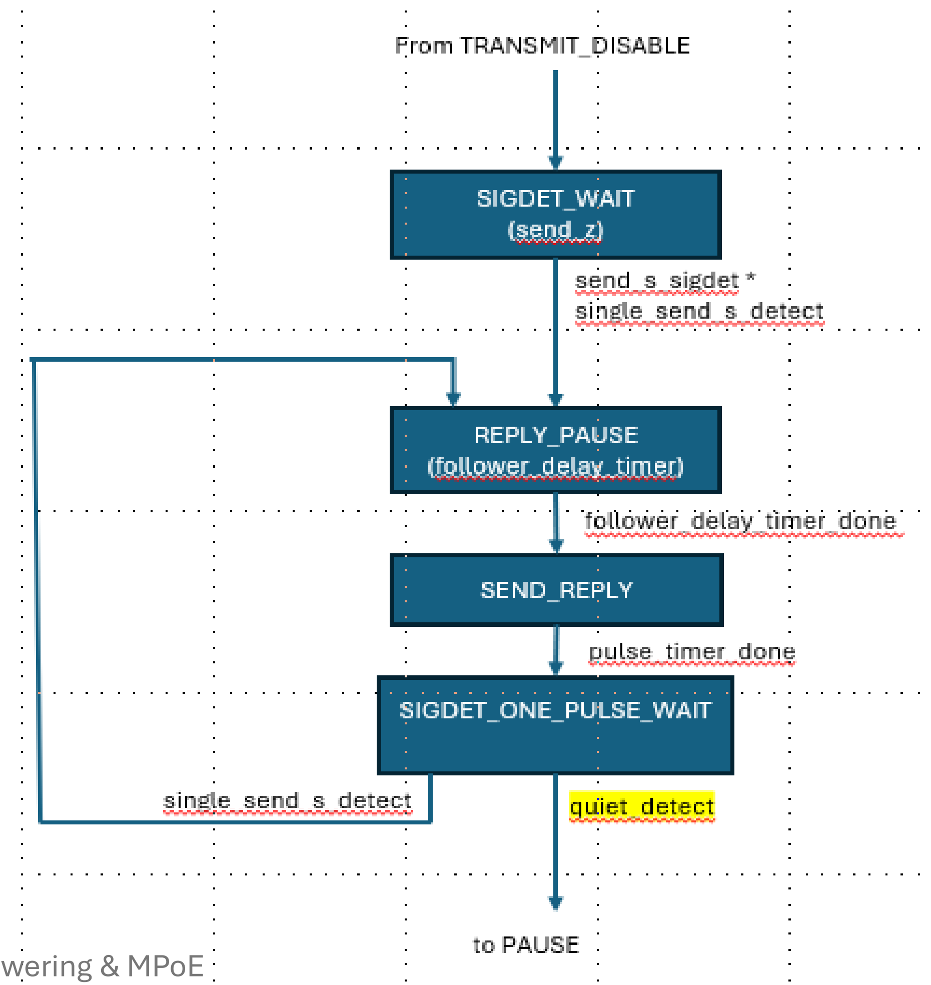
leader_pause_timer_done
* !send_s_sigdet

SILENT_WAIT

Quiet_detect

to PAUSE

Follower



From TRANSMIT_DISABLE

SIGDET_WAIT
(send_z)

send_s_sigdet *
single_send_s_detect

REPLY_PAUSE
(follower_delay_timer)

follower_delay_timer_done

SEND_REPLY

pulse_timer_done

SIGDET_ONE_PULSE_WAIT

single_send_s_detect

quiet_detect

to PAUSE

These two state chains go (as parallel vertical branches) replacing the states in the blue square in the PHY Link Synchronization state diagram at right.

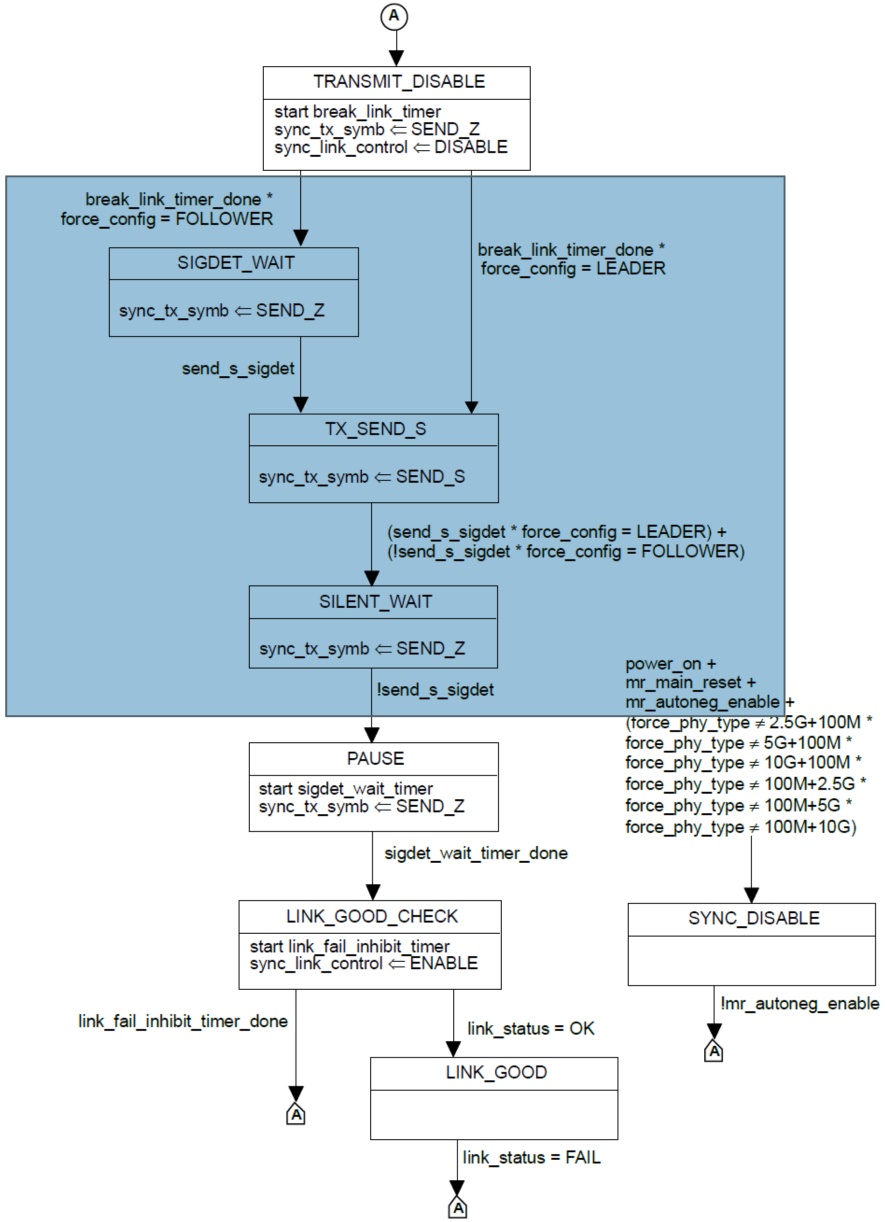
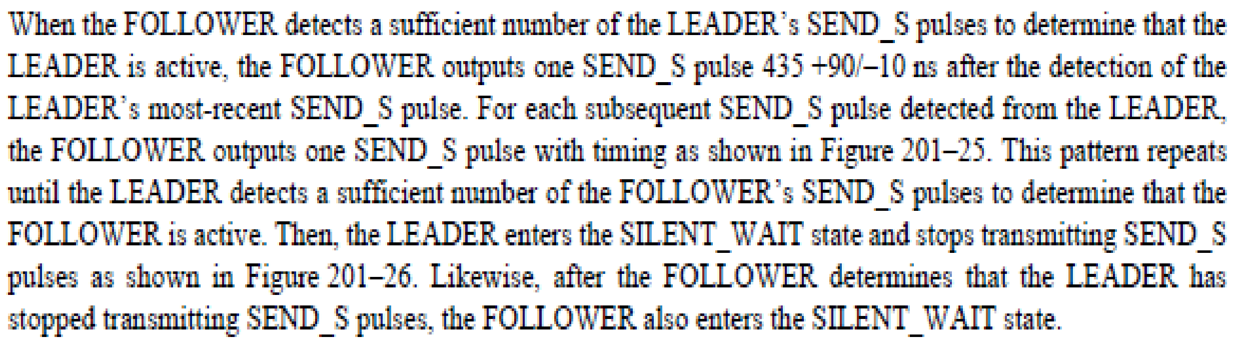
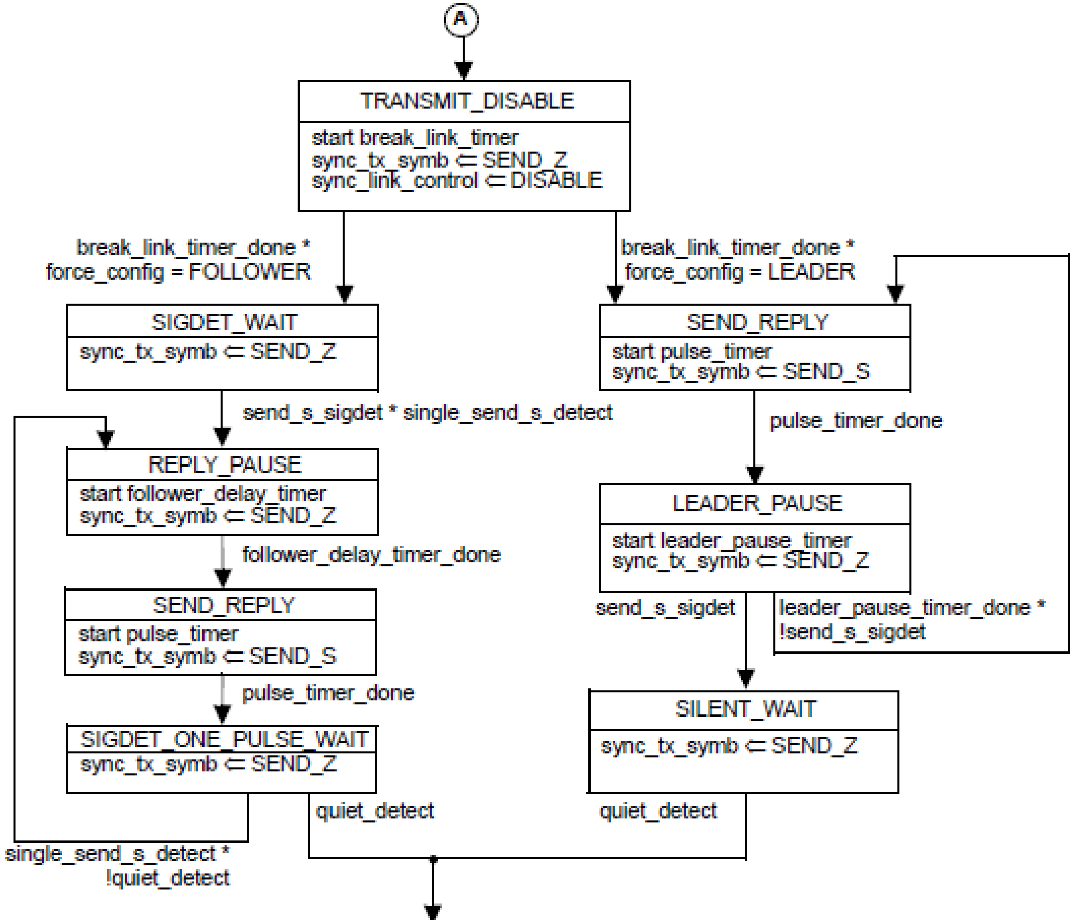


Figure 201–26—PHY Link Synchronization state diagram

Modified Description Text and State Diagram



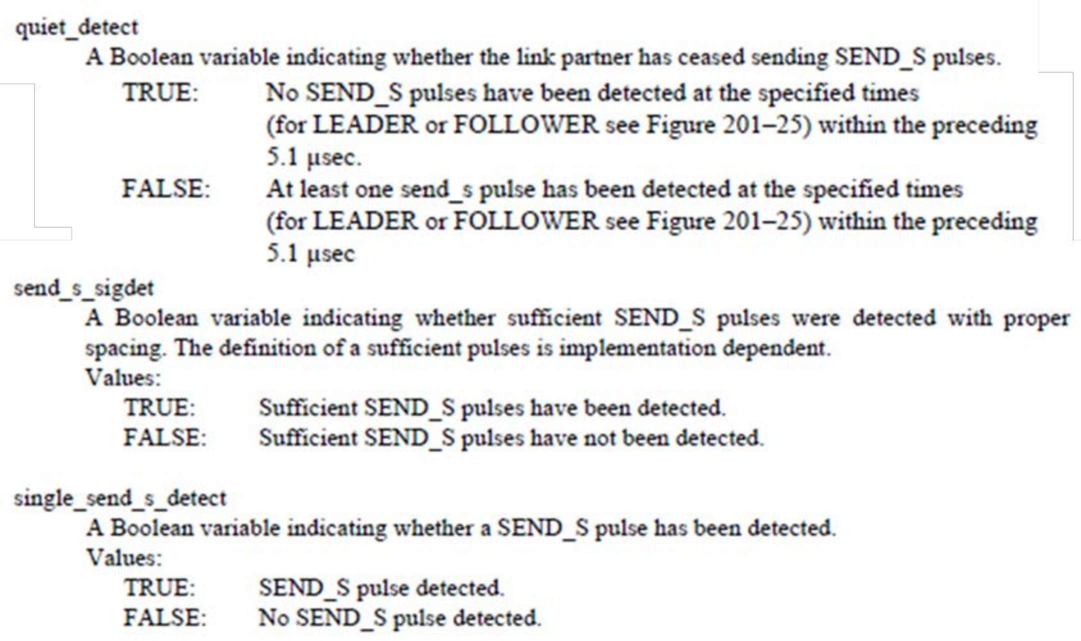
When the FOLLOWER detects a sufficient number of the LEADER's SEND_S pulses to determine that the LEADER is active, the FOLLOWER outputs one SEND_S pulse 435 +90/–10 ns after the detection of the LEADER's most-recent SEND_S pulse. For each subsequent SEND_S pulse detected from the LEADER, the FOLLOWER outputs one SEND_S pulse with timing as shown in Figure 201–25. This pattern repeats until the LEADER detects a sufficient number of the FOLLOWER's SEND_S pulses to determine that the FOLLOWER is active. Then, the LEADER enters the SILENT_WAIT state and stops transmitting SEND_S pulses as shown in Figure 201–26. Likewise, after the FOLLOWER determines that the LEADER has stopped transmitting SEND_S pulses, the FOLLOWER also enters the SILENT_WAIT state.



To PAUSE

Variables & Timers

Variables



quiet_detect

A Boolean variable indicating whether the link partner has ceased sending SEND_S pulses.

TRUE: No SEND_S pulses have been detected at the specified times (for LEADER or FOLLOWER see Figure 201–25) within the preceding 5.1 µsec.

FALSE: At least one send_s pulse has been detected at the specified times (for LEADER or FOLLOWER see Figure 201–25) within the preceding 5.1 µsec

send_s_sigdet

A Boolean variable indicating whether sufficient SEND_S pulses were detected with proper spacing. The definition of a sufficient pulses is implementation dependent.

Values:

TRUE: Sufficient SEND_S pulses have been detected.

FALSE: Sufficient SEND_S pulses have not been detected.

single_send_s_detect

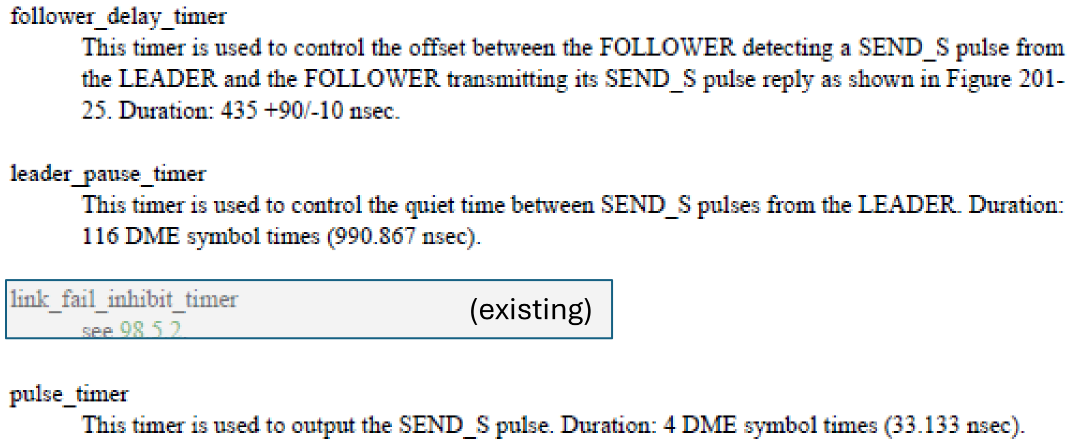
A Boolean variable indicating whether a SEND_S pulse has been detected.

Values:

TRUE: SEND_S pulse detected.

FALSE: No SEND_S pulse detected.

Timers



follower_delay_timer

This timer is used to control the offset between the FOLLOWER detecting a SEND_S pulse from the LEADER and the FOLLOWER transmitting its SEND_S pulse reply as shown in Figure 201-25. Duration: 435 +90/-10 nsec.

leader_pause_timer

This timer is used to control the quiet time between SEND_S pulses from the LEADER. Duration: 116 DME symbol times (990.867 nsec).

link_fail_inhibit_timer

see 98.5.2.

(existing)

pulse_timer

This timer is used to output the SEND_S pulse. Duration: 4 DME symbol times (33.133 nsec).

Questions?
Thank you.