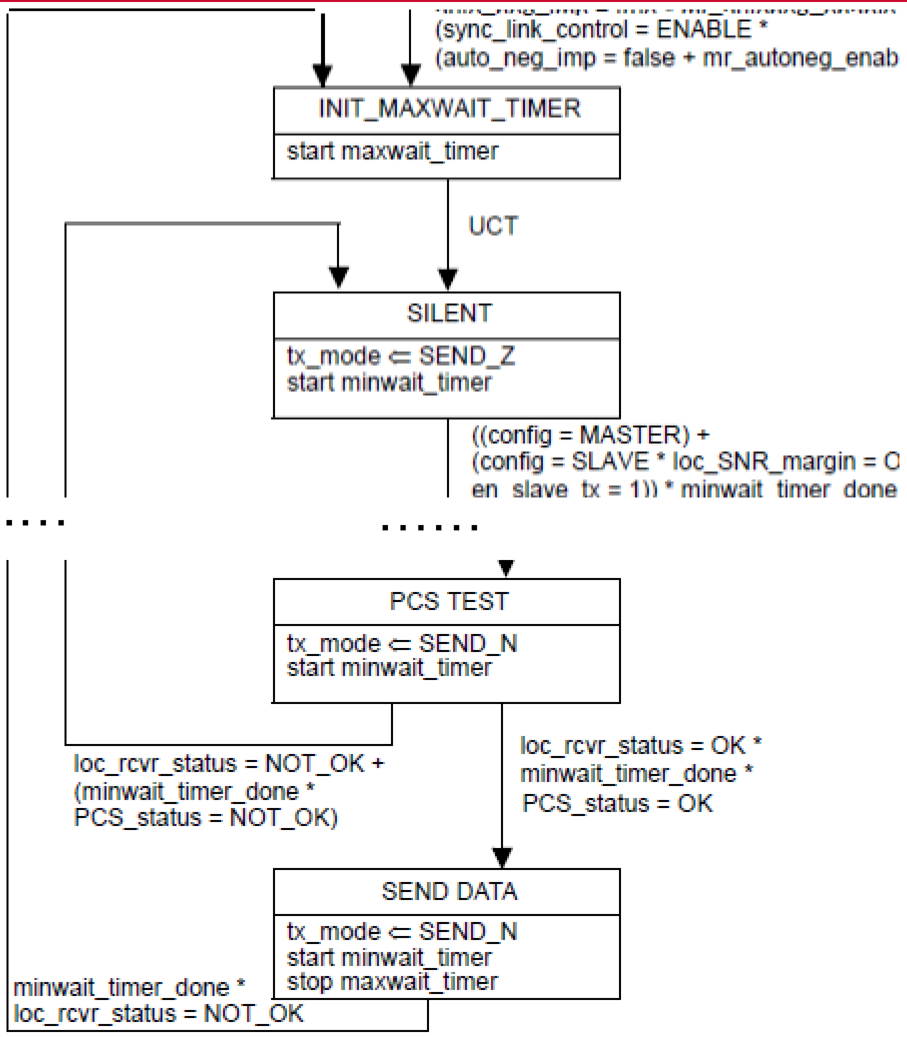
Issues in D1.1 State Diagrams

March 12, 2019

Mike Tu tum@broadcom.com
William Lo will@axonne.com
Steven Chen steven.chen@broadcom.com
Tom Souvignier tom.souvignier@broadcom.com



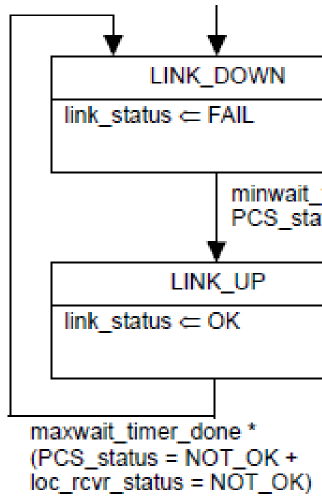


Figure 149-31 SEND_DATA and Figure 149-32 LINK_UP

- PHYC enters SEND_DATA state, with maxwait timer not yet expired
  - maxwait_timer stopped
- Link monitor enters LINK_UP state
  - link_status <= OK
- Assume loc_rcvr_staus drops when PHYC minwait_timer expires
  - PHYC exits SEND_DATA and goes to INIT_MAXWAIT_TIMER, then to SILENT
- Link monitor unable to exit LINK_UP state, as maxwait_timer already stopped, so never expires
  - link_status stays OK although PHYC went back to redo PAM2 training
- One work-around is to add the watchdog_status which monitors the line becoming SILENT
  - Add watchdog_status to the exit condition from the LINK_UP state, as shown in page 2 of Lo_3ch_01_0319.pdf.

On “PCS_status”

- hi_rfer
  - Figure 97-13 was proposed as RFER monitor state diagram for 802.3ch (comment #101 and #221).
  - In Figure 97-13, hi_rfer is initialized to false after either PCS reset or block_lock=false
- block_lock
  - Page 97 line 39: “…as well as an InfoField, which is inserted in the 16th PCS partial PHY frame. When the PCS Synchronization process is synchronized to this pattern, block_lock is asserted.”
  - So block_lock will be asserted while still in PAM2 training
- PCS_status
  - D1.1, page 107 line 47
  - Indicates whether the PCS is in a fully operational state. It is only true if block_lock is true and hi_rfer is false.
- Therefore the PCS_status will get set during PAM2 training mode according to D1.1.
- One solution is to set hi_rfer<=true by default
- Another way is to define PCS_status differently

Comments from George

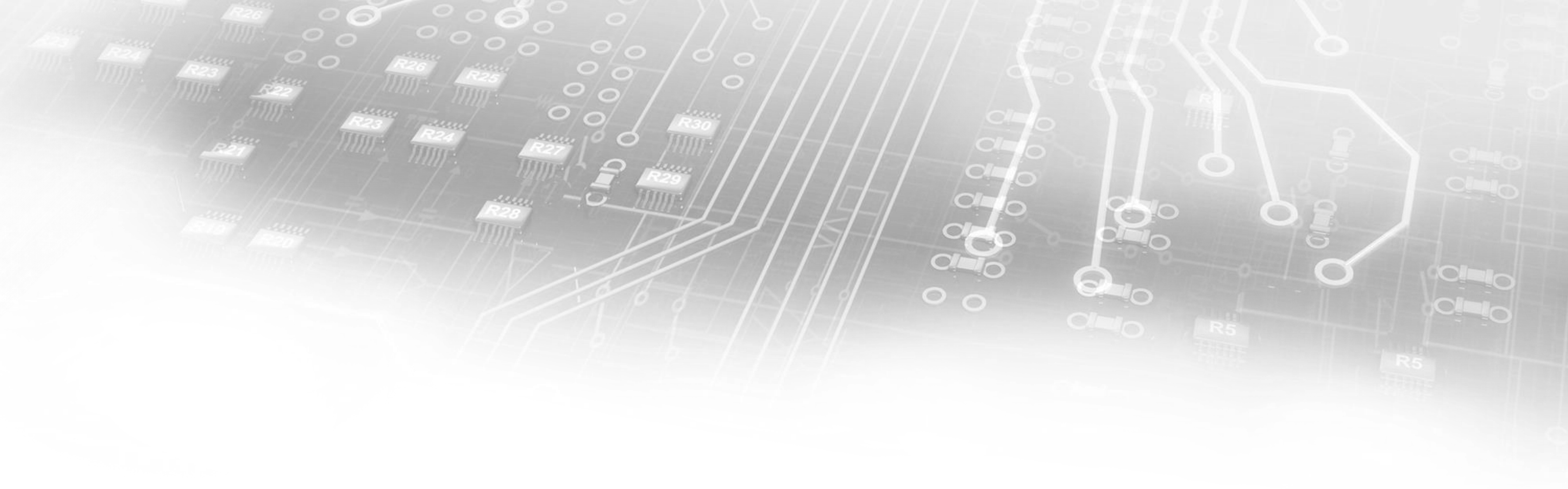
- Mike – at first, I thought you may have uncovered a VERY old problem on slide 2, that EVERY BASE-T PHY since 100BASE-T2 has carried forward (yes, the same relationship of maxwait_timer_done to the phy control and link monitor diagrams is present in 100BASE-T2, 1000BASE-T, 10GBASE-T, 2.5G/5GBASE-T, 25GBASE-T, 40GBASE-T, and 100BASE-T1).
- They all stop maxwait_timer upon reaching data (or idle) transmission, and they all rely on maxwait_timer_done as a condition going from link_up to link_fail. They all have an identical structure.
- Adding a 'watchdog' is not the way to fix this, if it is, in fact, a problem. Changing the maxwait_timer_done for something else on the link monitor might be better – IF anything needs to be done. I'm not sure it does, but I'm not entirely sure that it does. Here's what it looks like happens from the state diagrams (which doesn't agree with my memory):
- The maxwait timer will not be done when the phy successfully trains. Entry to the data (or idle) state stops maxwait_timer. If maxwait_timer_done is true, the PHY has failed to train. Link goes down, link fail inhibit dies with it.
- BUT, reading the state diagrams (again, I think implementations might be different) all of these phys, according to the state diagrams have hysteresis in their link state which seems to allow a retrain attempt. If the local receiver status drops to FAIL in the data or idle state will attempt a retrain, first resetting maxwait_timer (a "start timer" always resets the timer). They return to the training state in PHY control where maxwait_timer is started. In 802.3ch this is INIT_MAXWAIT_TIMER. If data mode is established without maxwait_timer expiring, no problem. Link stays up. If not, they drop link, and autoneg starts again.
- On the one hand, this seems to make sure you don't have to re-autoneg if you don't need it, but again, not sure this is what really happens - I would have thought link would drop.
- I've asked UNH if my interpretation above is correct – because it seems weird.

Comments from George (cont.)

- On high_rfer, I wouldn’t want to set it true by default. That is indicating an error condition that shouldn’t happen. I suggest defining pcs_status to be:
- Pcs_status = block_lock * (!hi_rfer) * (txmode = SEND_N)
- That way, PCS_status is what it says it is:
- The parameter pcs_status conveys to the PMA Receive function the information that the PCS is operating reliably in the data mode.

Comments from William

- For issue #1 simply remove stop maxwait_timer from the SEND DATA state. This is what is done in 1000BASE-T1.
- I'm ambivalent about the watchdog. I made a comment on that in case we want to adopt it but I'm also ok not having it.
- For issue #2 I think we can redefine PCS_status by adding one more condition and that is the PCS datapath is decoding properly.
- I don't like the solution to set hi_rfer to default to true since it may have other side effects like setting registers.



THANK YOU