

# Proposed Changes to State Diagrams

April 3, 2019

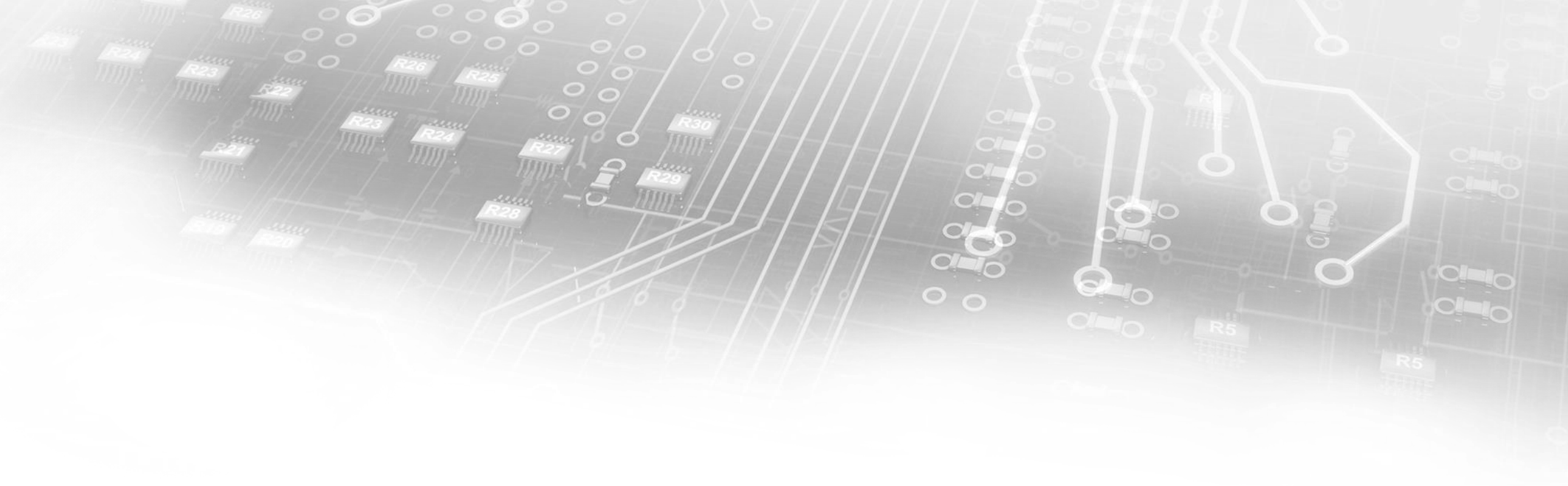
Mike Tu [tum@broadcom.com](mailto:tum@broadcom.com)

Steven Chen [steven.chen@broadcom.com](mailto:steven.chen@broadcom.com)

Tom Souvignier [tom.souvignier@broadcom.com](mailto:tom.souvignier@broadcom.com)

# Introduction

- Several issues in the state diagrams were discussed in [http://www.ieee802.org/3/ch/public/mar19/tu\\_3ch\\_01a\\_0319.pdf](http://www.ieee802.org/3/ch/public/mar19/tu_3ch_01a_0319.pdf)
- **Issue 1: incorrect pcs\_status**
  - pcs\_status set to “true” while still in the PAM2 training state
- **Issue 2: long delay and incorrect link\_status if link drops in data mode**
  - If the PHYC enters SEND\_DATA state and then the loc\_rcvr\_status goes down, it will take another full “maxwait\_timer” expiration to restart the autoneg, while the link\_status stays as OK.
- We propose the following changes to resolve these issues:
  1. Utilize variable “pcs\_data\_mode” even without optional EEE. It is set to “true” only after PHYC enters the SEND\_DATA state.
  2. Let “pcs\_status” be dependent on “pcs\_data\_mode”
  3. Once in SEND\_DATA, do not try to re-initiate another PAM2 training within PHYC state diagram. Always go through autoneg or the LINK SYNC state machine (takes ~1 msec).



# pcs\_data\_mode

---

# “pcs\_data\_mode” in Clause 55 vs. Clause 149 D1.2

- Originally defined for EEE enabled PHY only
- Clause 55
  - PMA\_PCSDATAMODE: defined in 55.2.2:
    - EEE-capable PHYs additionally support the following service primitives:
      - PMA\_PCSDATAMODE.indication (pcs\_data\_mode)
      - It appears in Figure 55-4 10GBASE-T service interfaces
    - pcs\_data\_mode
      - Appears in Figure 55-3 Functional Block diagram, Figure 55-5 PCS reference diagram, Figure 55-21 PMA reference diagram, and Figure 55-28 PHY Control state diagram.
      - It is defined in 55.4.5.1 State diagram variables, in the PMA, not in the PCS.
- Clause 149 D1.2
  - PMA\_PCSEDATAMODE
    - Mentioned but never defined
    - Did not appear in Figure 149-3 service interfaces
  - pcs\_data\_mode
    - Defined in 149.3.6.2.2
    - Defined again in 149.4.4.1
    - Missing from the reference diagrams and the functional block diagrams

## Add “PMA\_PCSDATAMODE” to PMA service interface

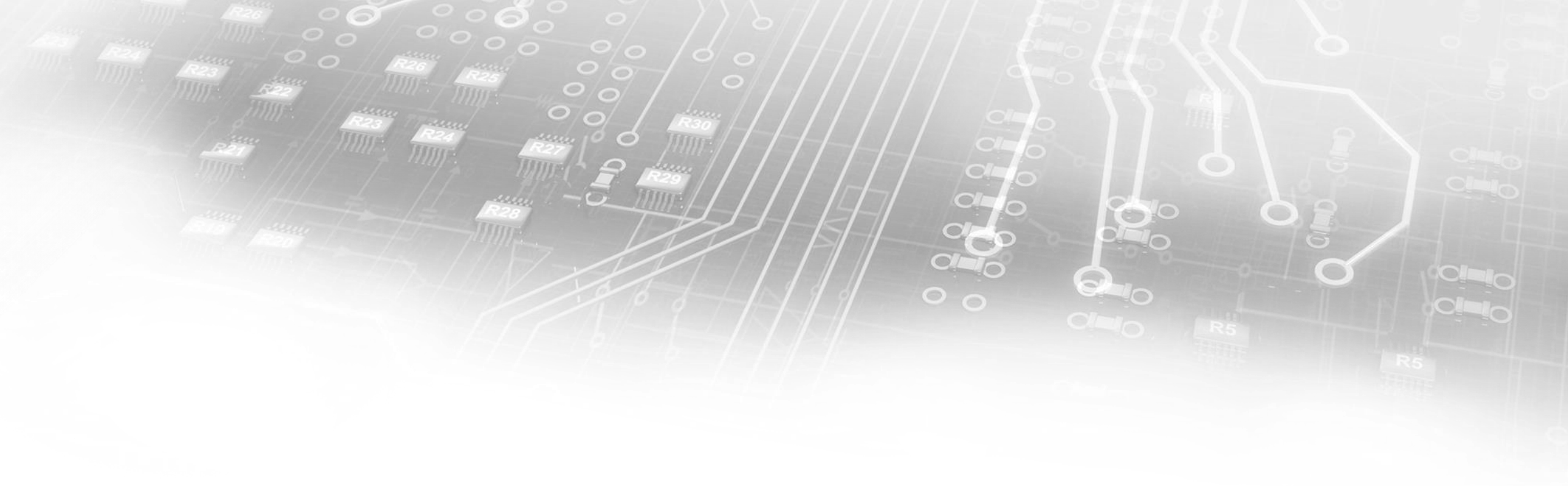
- On page 74, line 22, after 149.2.2.8, insert “PMA\_PCSDATAMODE.indication” based on 55.2.2.11:
  - 149.2.2.8a PMA\_PCSDATAMODE.indication
    - This primitive indicates whether or not the PCS state diagrams are able to transition from their initialization states. The pcs\_data\_mode variable is generated by the PMA PHY Control function. It is passed to the PCS Control function via the PMA\_PCSDATAMODE.indication primitive.
  - 149.2.2.8a.1 Semantics of the primitive
    - PMA\_PCSDATAMODE.indication (pcs\_data\_mode)
  - 149.2.2.8a.2 When generated
    - The PMA PHY Control function generates PMA\_PCSDATAMODE.indication messages continuously.
  - 149.2.2.8a.3 Effect of receipt
    - Upon receipt of this primitive, the PCS performs its transmit function as described in 149.3.2.2.

# Make “pcs\_data\_mode” available with or without optional EEE

- In D1.2, 149.4.4.1, page 147, line 20 to 25:
  - ~~The following variables are required for PHYs that support the EEE capability:~~
  - pcs\_data\_mode
    - Generated by the PMA PHY Control function and indicates whether or not the local PHY may transition its PCS state diagrams out of their initialization states. The current value of the pcs\_data\_mode is passed to the PCS via the PMA\_PCSDATAMODE.indicate primitive. ~~In the absence of the optional EEE capability, the PHY operates as if the value of this variable is TRUE.~~
- In D1.2, 149.3.6.2.2, page 102, line 37 to 41:
  - Delete “pcs\_data\_mode” and the associated descriptions

## Changes to Figures

- In Figure 149-2, add “pcs\_data\_mode” coming out of “PHY CONTROL”, and enters “PCS TRANSMIT”.
- In Figure 149-4, add “pcs\_data\_mode” entering from “PMA SERVICE INTERFACE” into the “PCS TRANSMIT”. See Figure 55-5 for reference.
- In Figure 149-26, add “pcs\_data\_mode” coming out of “PHY CONOTRL” and going up to the PMA SERVICE INTERFACE”. See Figure 55-21 for reference.
- In Figure 149-33, PHY Control State Diagram:
  - In “DISABLE\_TRANSMITTER” state, add “pcs\_data\_mode <= false”.
  - In “SEND DATA” state, add “pcs\_data\_mode <= true”.



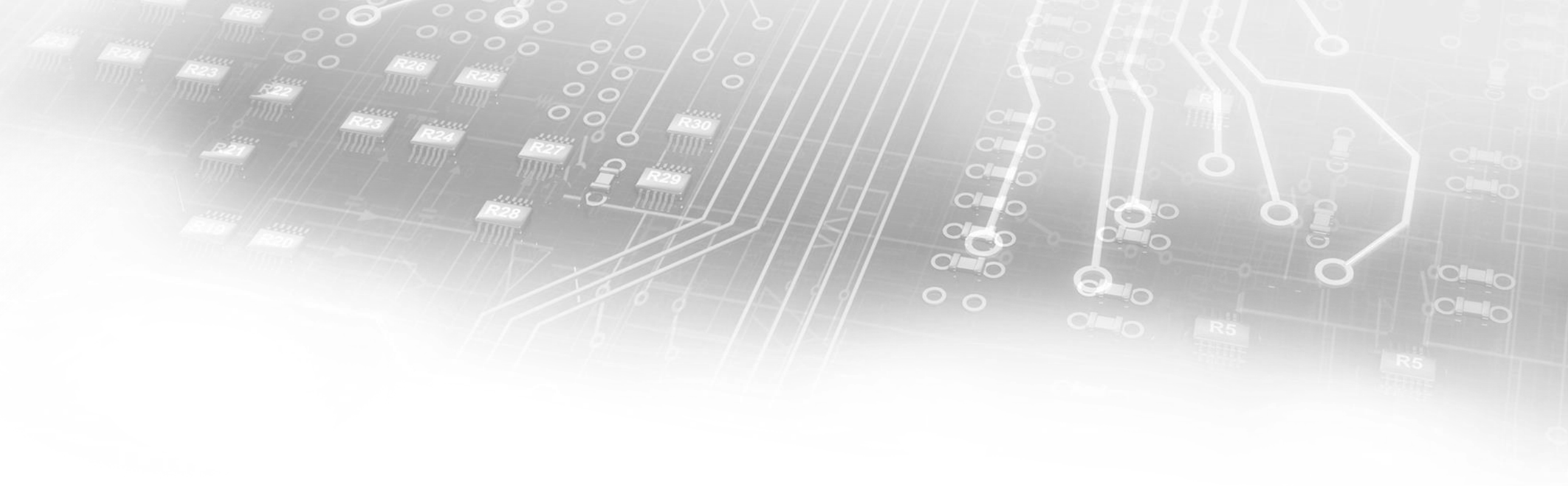
# pcs\_status

---



## Redefine pcs\_status

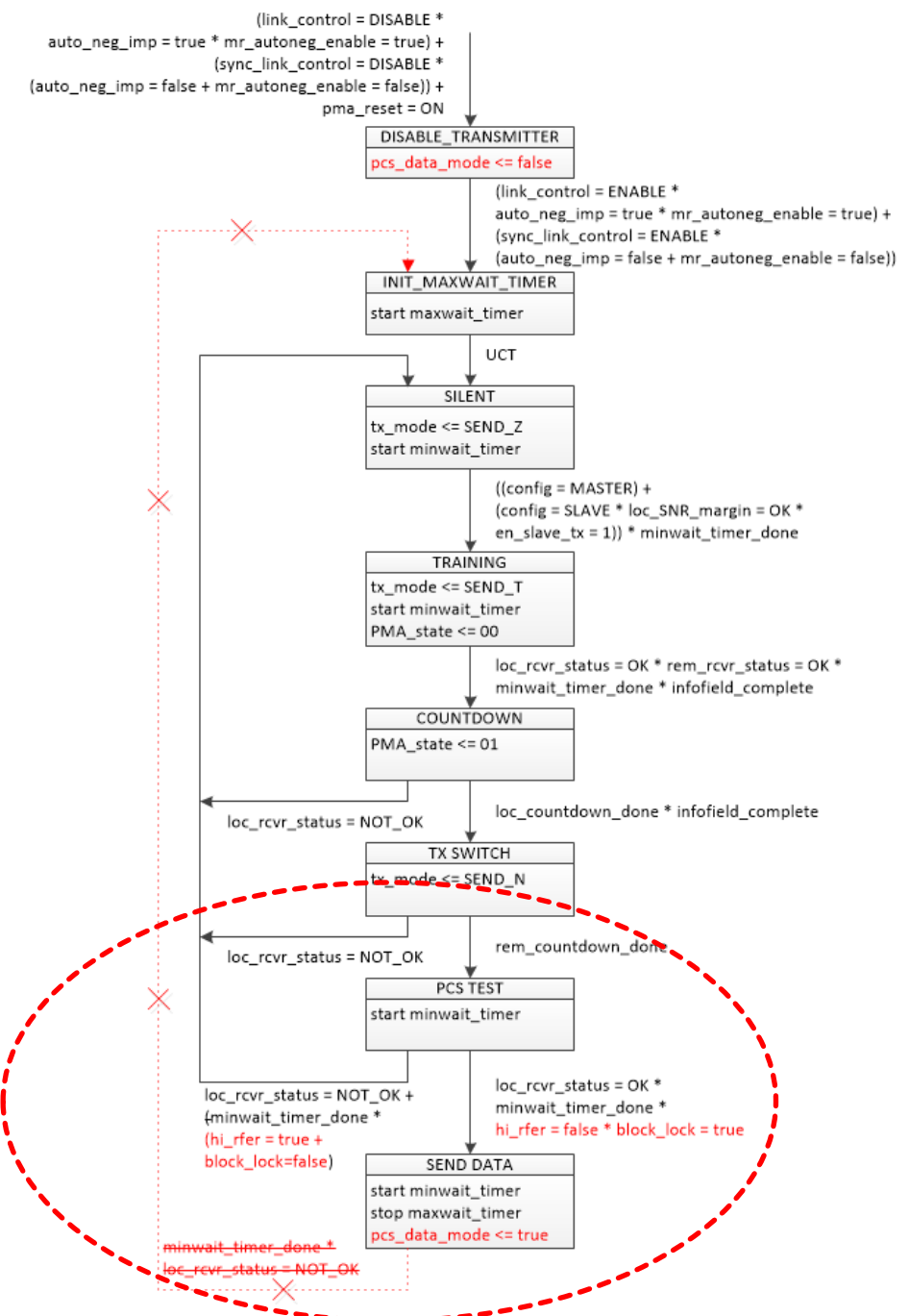
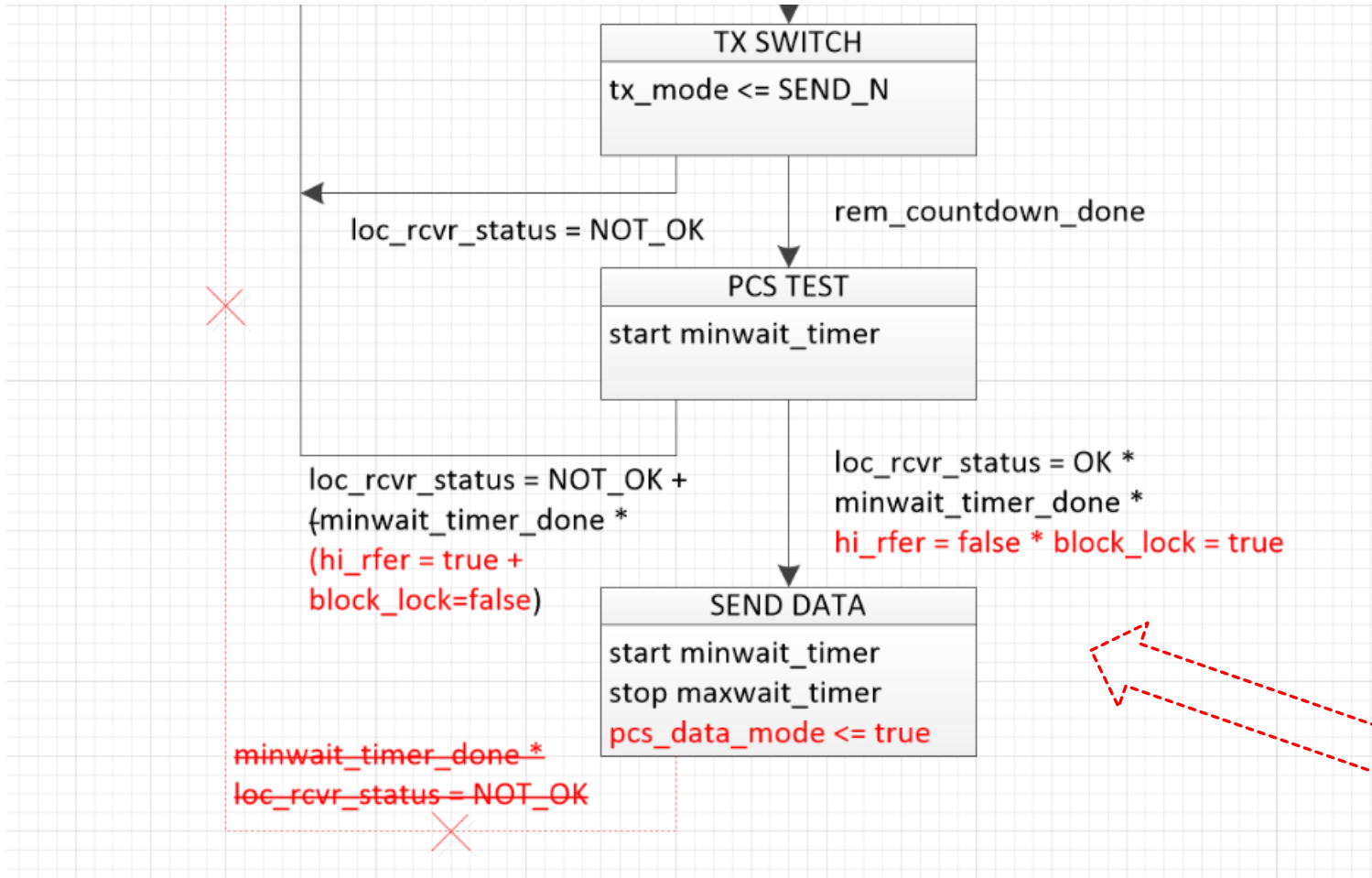
- In 149.3.7.1, page 106, line 22
- pcs\_status:
  - Indicates whether the PCS is in a fully operational state. It is only true if pcs\_data\_mode is true, block\_lock is true, and hi\_rfer is false. This status is reflected in MDIO register 3.2324.10. A latch low view of this status is reflected in MDIO register 3.2323.2 and the inverse of this status is reflected in MDIO register 3.2323.7.



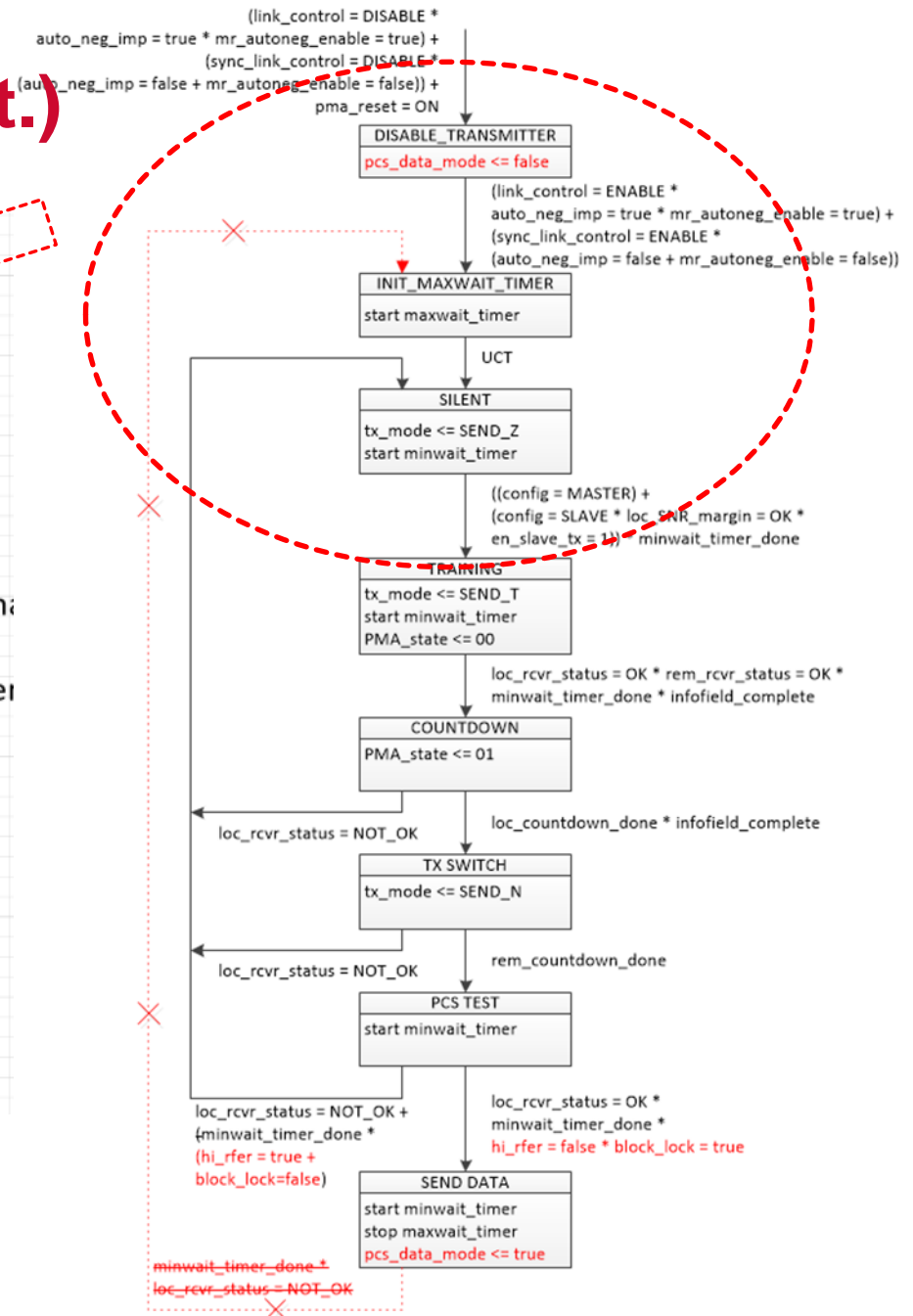
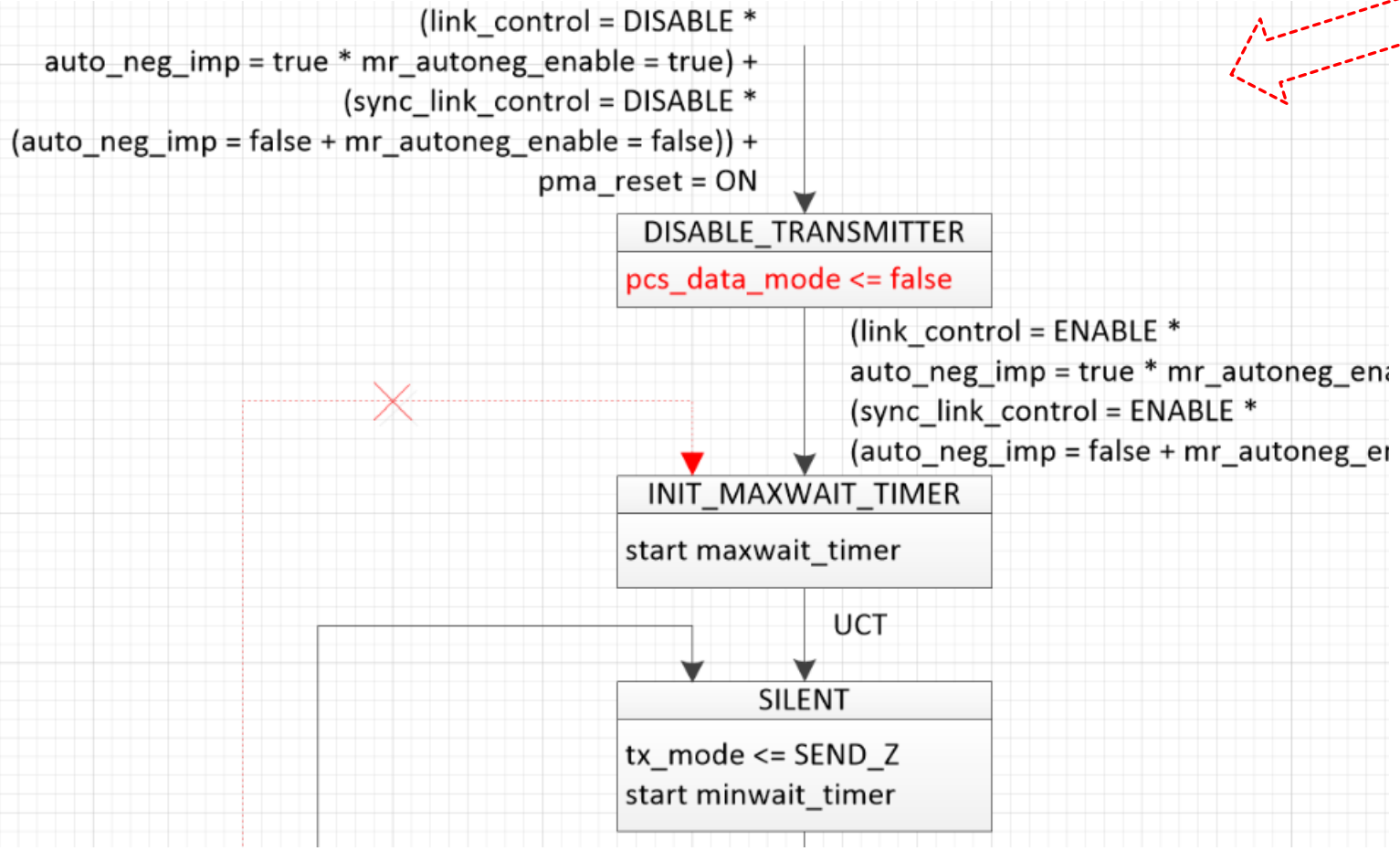
# Figure 149-33 PHY Control state diagram

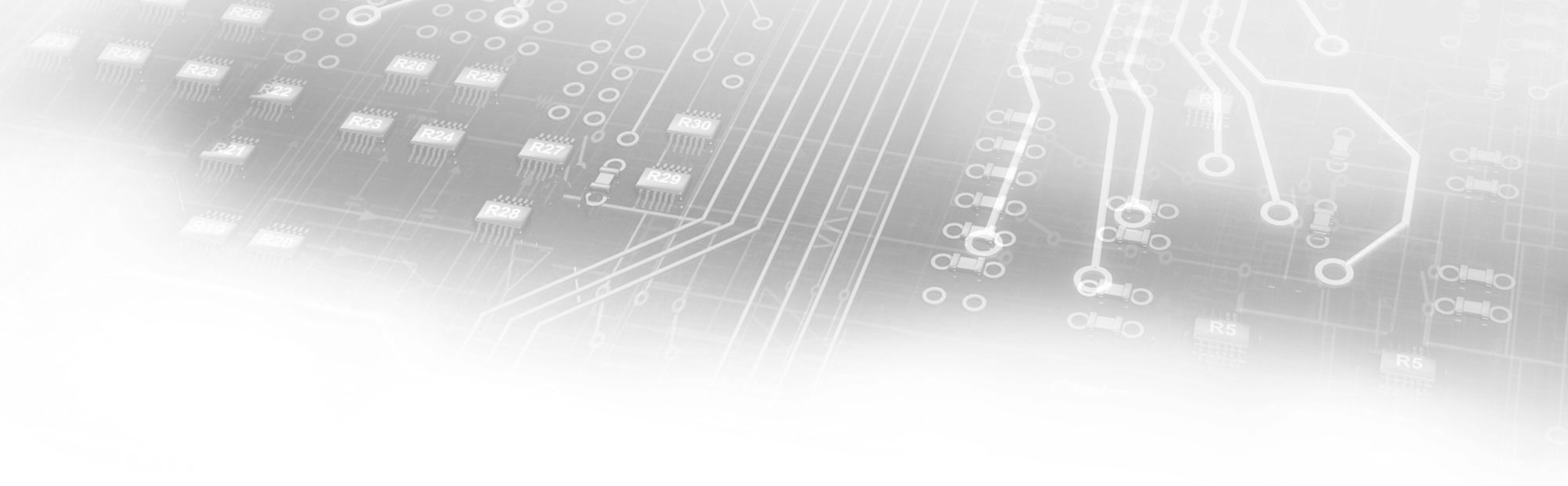
---

# PHY Control State Diagram Changes



# PHY Control State Diagram Changes (cont.)

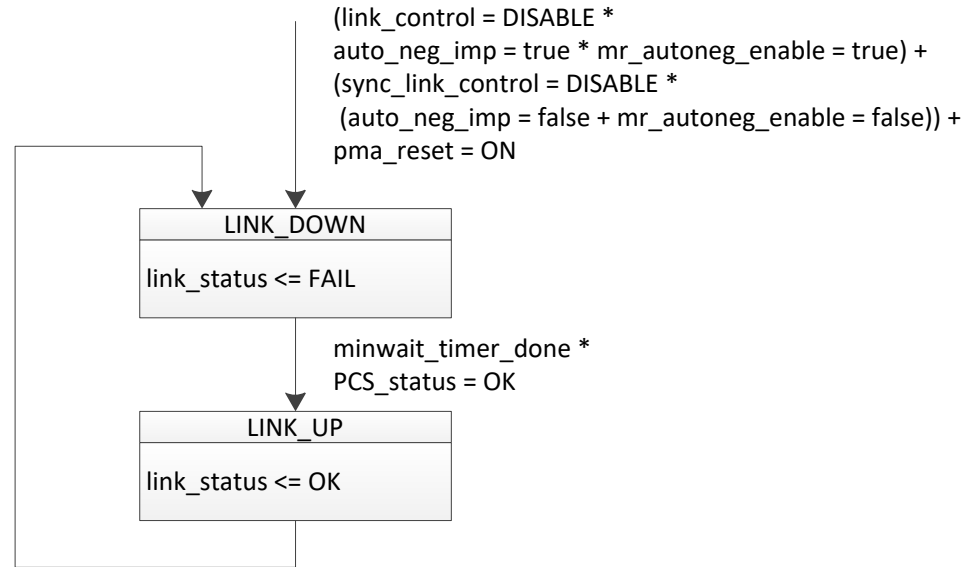




# Figure 149-34 Link Monitor state diagram

---

# Updated Figure 149-34

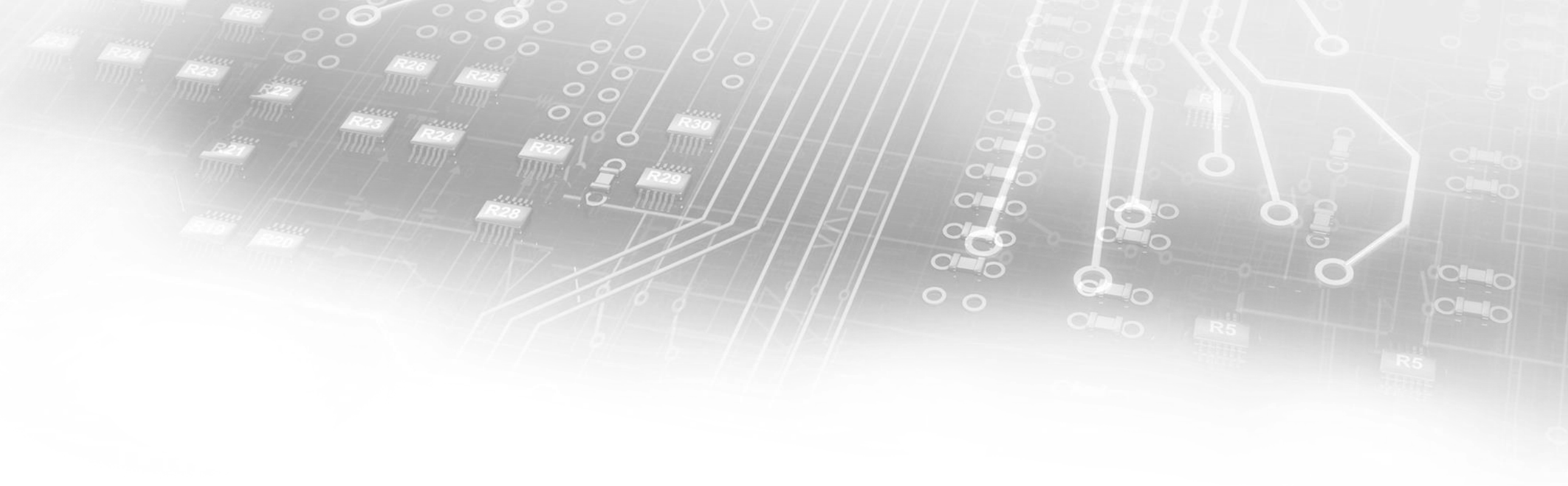


~~maxwait\_timer\_done \*~~  
 (PCS\_status = NOT\_OK + loc\_rcvr\_status = NOT\_OK) +  
 PMA\_refresh\_status = FAIL+  
~~PMA\_watchdog\_status = NOT\_OK~~

~~NOTE 1 – maxwait\_timer is started in PHY Control state diagram (see Figure 149-16).~~

NOTE 1 – The variables link\_control and link\_status are designated as link\_control\_mGigT1 and link\_status\_mGigT1, respectively, by the Auto-Negotiation Arbitration state diagram (Figure 98-7) if the optional Auto-Negotiation function is implemented.

- PCS\_state is dependent on pcs\_data\_mode now. So it will enter LINK\_UP only after pcs\_data\_mode is true
- The maxwait\_timer would have been stopped when PHYC enters “SEND\_DATA” state → do not check for “maxwait\_timer\_done”.
- If link\_status drops to FAIL, autoneg or LINK SYNC state machines will take over and restart link attempt.



# THANK YOU

---