802.3cg
On the PCS Receive in clause 147

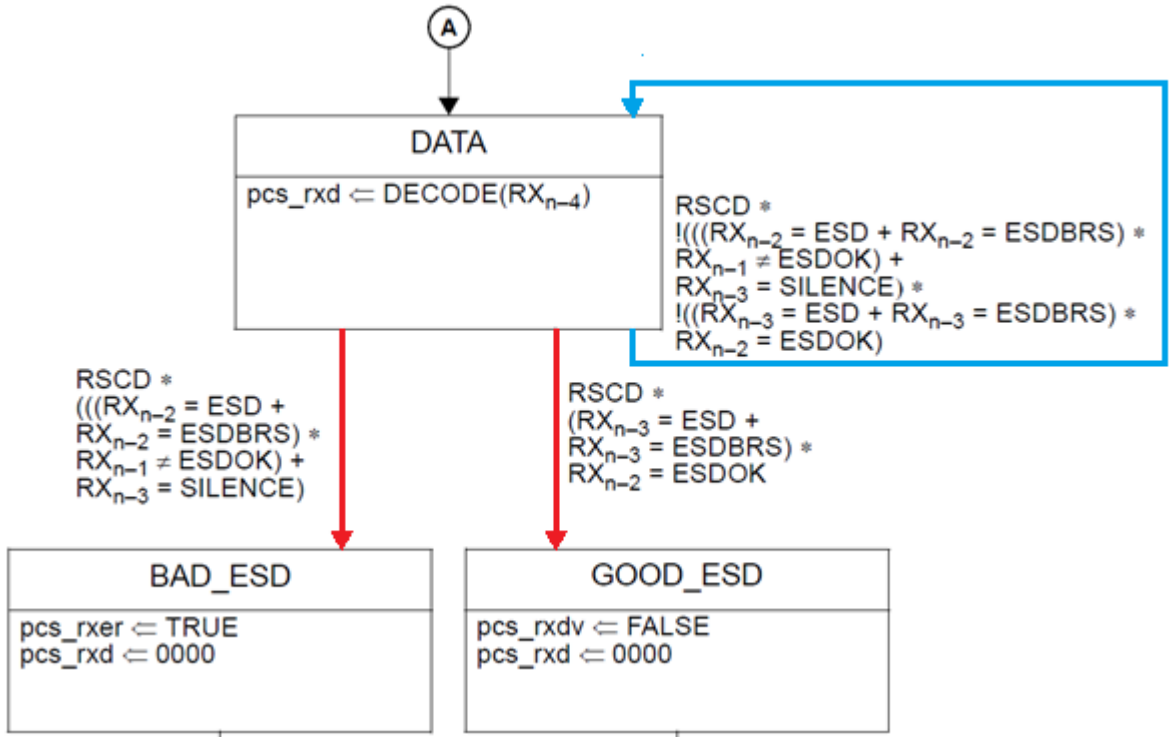
Gergely Huszak, George Zimmerman, Piergiorgio Beruto

(v3)

RX_n indexing and delay line

d3.0 comment i-319
See “Figure 147–8—PCS Receive state diagram (part b)”

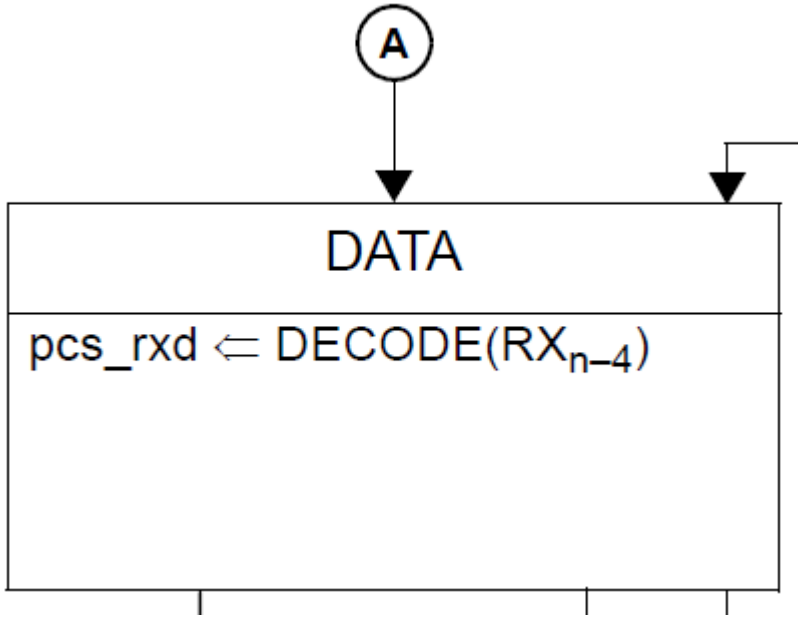
Baseline: PCS RX must include a 4-symbol long delay line
Background: To allow starting actual frame reception upon receiving a valid JJHH, a delay line is there, therefore PCS RX is always 3 symbols behind the channel
Details: The delay line is referred to as RX_{n-x} and it is fed[1] before RSCD, thus when exiting DATA[2] DECODE() is not executed anymore (which is run last upon last DATA→DATA)



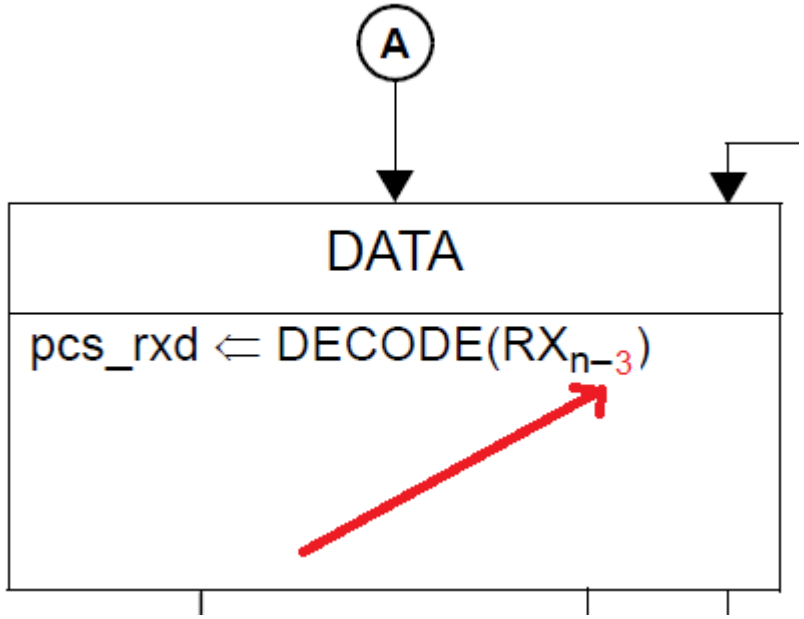
[1] PMA_UNITDATA.indication(rx_sym): RX_{n-2}→RX_{n-3}, RX_{n-1}→RX_{n-2}, RX_n→RX_{n-1}, rx_sym→RX_n
[2] E.g. when RX_{n-4}=*data*, RX_{n-3}=ESD, RX_{n-2}=ESDOK, RX_{n-1}=SILENCE and RX_n=SILENCE

Problem: Incorrect sequence number brought over from clause 96[1]
Result: The last symbol of the frame remains undecoded (upon DATA→GOOD_ESD)
Solution:
1. Correct PCS RX in clause 147 → being done now
2. submit maintenance request to correct clause 96 → done by George Zimmerman

Current text:



Proposed text (correction highlighted):



[1] See for example state DATA in "Figure 96–10—PCS Receive state diagram"

Mutual exclusivity of the 2 forward exit conditions of DATA

d3.0 comment i-278
See “Figure 147–8—PCS Receive state diagram (part b)”

Problem: The condition on DATA→BAD_ESD and that on DATA→GOOD_ESD may both evaluate to TRUE at the same time
Result: PHY behavior becomes implementation-dependent (see interoperability)
Example: RX_{n-3}=ESDOK/ESDBRS, RX_{n-2}=ESDOK/ESDBRS[1], RX_{n-1}=SILENCE, RX_n=SILENCE

Current text:

- *DATA→BAD_ESD:*
  RSCD *
  (((RX$_{n-2}$=ESD +
  RX$_{n-2}$=ESDBRS) *
  RX$_{n-1}$≠ESDOK) +
  RX$_{n-3}$=SILENCE)
- *DATA→DATA:*
  RSCD *
  !(((RX$_{n-2}$=ESD +
  RX$_{n-2}$=ESDBRS) *
  RX$_{n-1}$≠ESDOK) +
  RX$_{n-3}$=SILENCE) *
  !((RX$_{n-3}$=ESD +
  RX$_{n-3}$=ESDBRS) *
  RX$_{n-2}$=ESDOK)

[1] The 5B symbol of ESDOK and ESDBRS is the same (R)

Proposed text (additions highlighted):

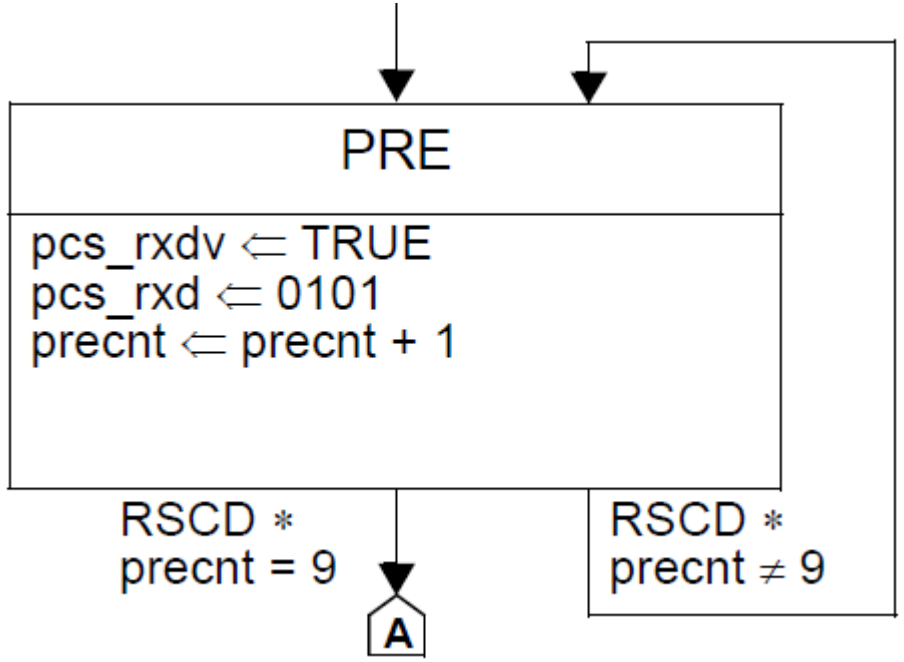
- *DATA→BAD_ESD:*
  RSCD *
  (((RX$_{n-2}$=ESD +
  RX$_{n-2}$=ESDBRS) *
  RX$_{n-1}$≠ESDOK *
  RX$_{n-3}$≠ESD *
  RX$_{n-3}$≠ESDBRS) +
  RX$_{n-3}$=SILENCE)
- *DATA→DATA:*
  RSCD *
  !(((RX$_{n-2}$=ESD +
  RX$_{n-2}$=ESDBRS) *
  RX$_{n-1}$≠ESDOK *
  RX$_{n-3}$≠ESD *
  RX$_{n-3}$≠ESDBRS) +
  RX$_{n-3}$=SILENCE) *
  !((RX$_{n-3}$=ESD +
  RX$_{n-3}$=ESDBRS) *
  RX$_{n-2}$=ESDOK)

Descrabmler locking

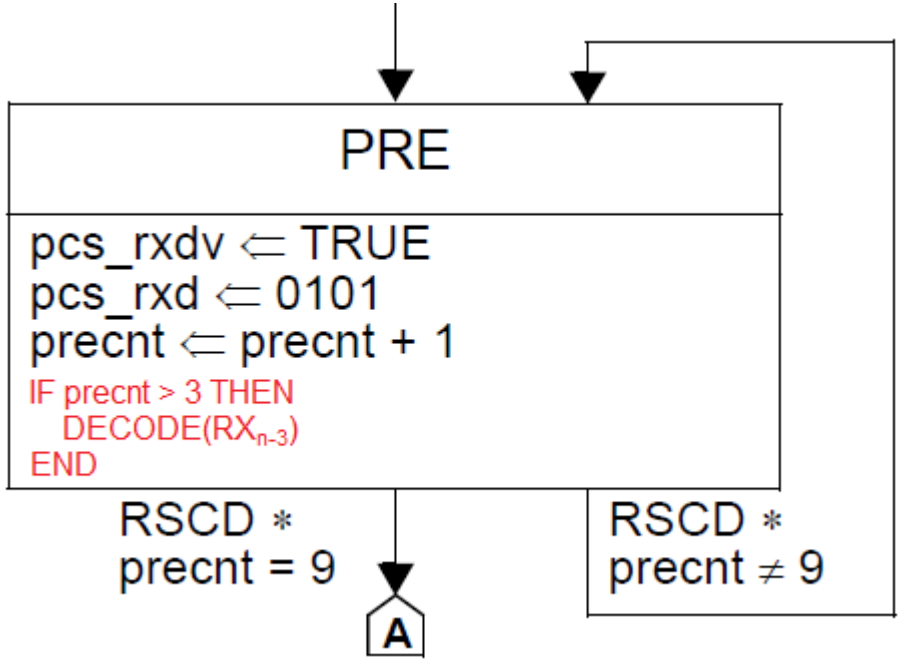
d3.0 comment i-281
See “Figure 147–7—PCS Receive state diagram (part a)”

Baseline: Self-synchronous descrambler needs initial 17 bits (≈5 symbols) to lock
Problem: If implemented as specified by figures 147–7 and 147–8, descrambler will start locking in DATA state, this way making first data 17 bits undecodable
Solution: Start feeding descrambler immediately after receiving valid "JJHH" in PRE[1]

Current text:



Proposed text (additions highlighted):



[1] The index x of the delay line RX_{n-x} is 3 (x=3), as per the explanation at pages 2-4 (on comment i-319)