Canova Tech

The Art of Silicon Sculpting

PIERGIORGIO BERUTO
ANTONIO ORZELLI

IEEE802.3cg TF

Comments on draft
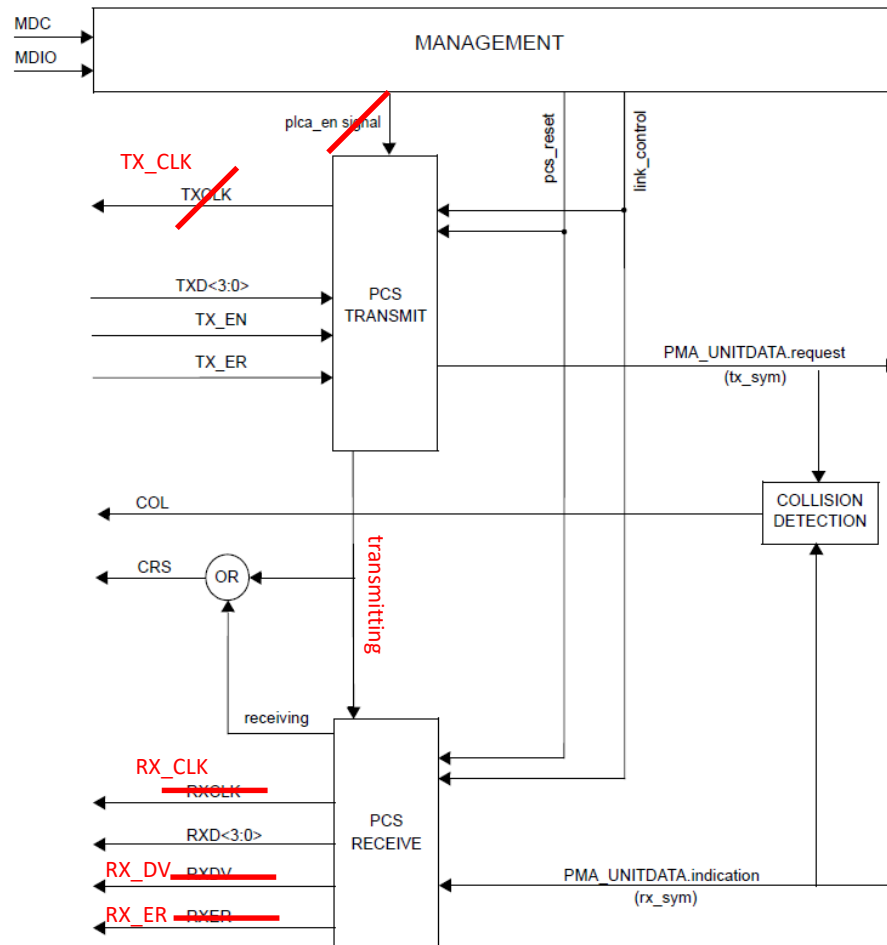
March 5th 2018

Figure 147–2—PCS reference diagram

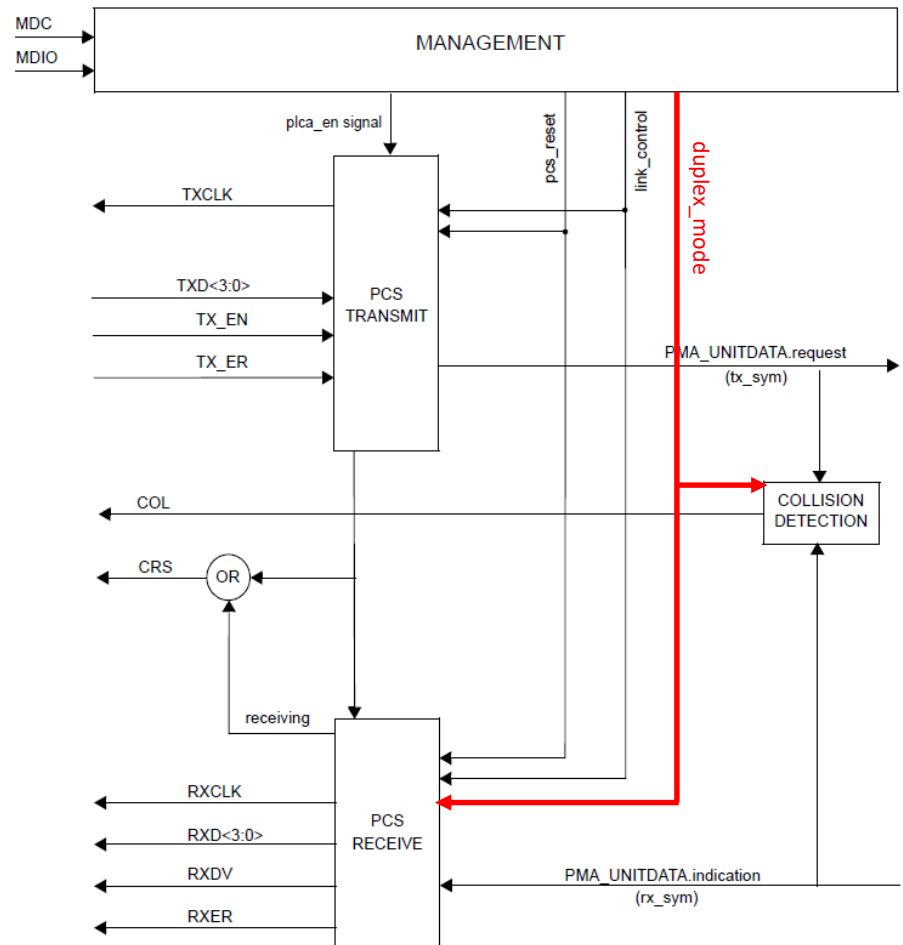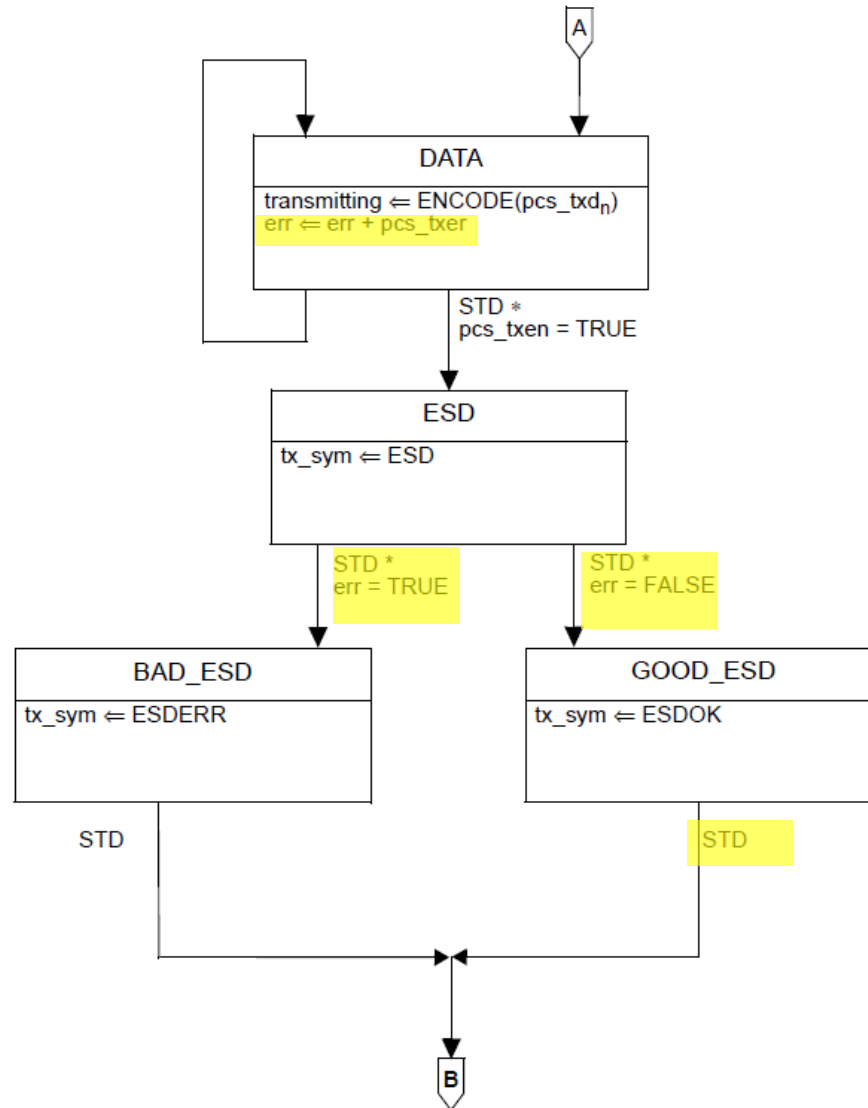Collision detection shall be disabled when operating in full-duplex mode
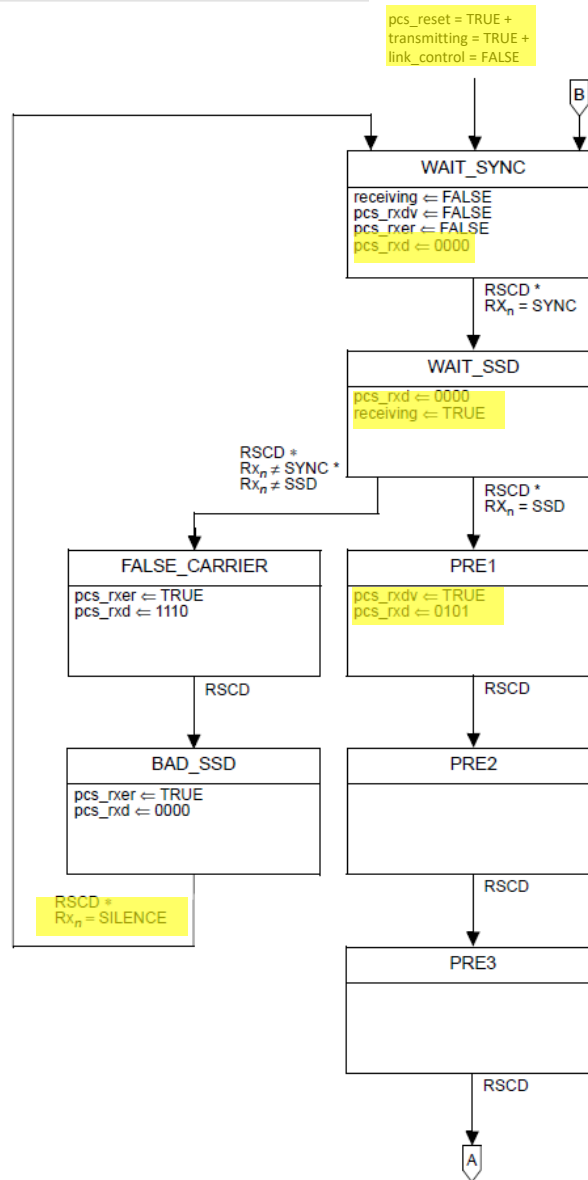

Figure 147–2—PCS reference diagram

- In order to support full-duplex mode, the PCS RX block should be configured accordingly

147.2.3.1 Variables

…

duplex_mode

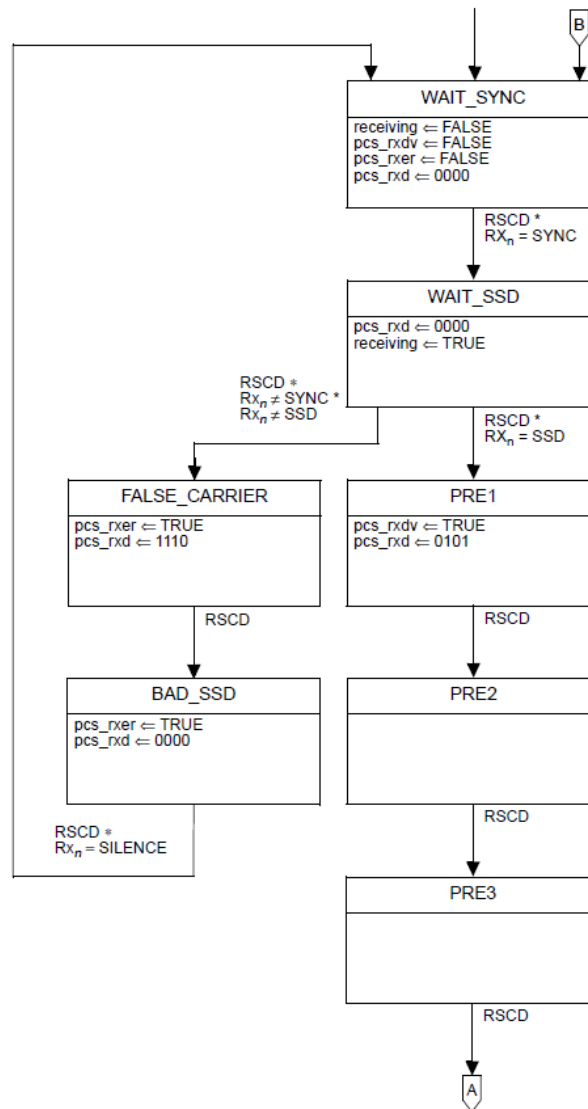> indicates whether the PHY is configured for full-duplex operation (DUPLEX_FULL) or half-duplex operation (DUPLEX_HALF). This ariable is set after bit 8 in MDIO register 0 defined in table 22-7

(transmitting = TRUE * duplex_mode = DUPLEX_HALF)

pcs_reset = TRUE +
~~transmitting = TRUE~~ +
link_control = FALSE

B

WAIT_SYNC

receiving ⇐ FALSE
pcs_rxdv ⇐ FALSE
pcs_rxer ⇐ FALSE
pcs_rxd ⇐ 0000

RSCD *
RX_n = SYNC

WAIT_SSD

pcs_rxd ⇐ 0000
receiving ⇐ TRUE

RSCD *
Rx_n ≠ SYNC *
Rx_n ≠ SSD

RSCD *
RX_n = SSD

FALSE_CARRIER

pcs_rxer ⇐ TRUE
pcs_rxd ⇐ 1110

PRE1

pcs_rxdv ⇐ TRUE
pcs_rxd ⇐ 0101

RSCD

RSCD

BAD_SSD

pcs_rxer ⇐ TRUE
pcs_rxd ⇐ 0000

PRE2

RSCD *
Rx_n = SILENCE

RSCD

PRE3

RSCD

A

In order to support full-duplex mode, the PCS RX block should behave accordingly
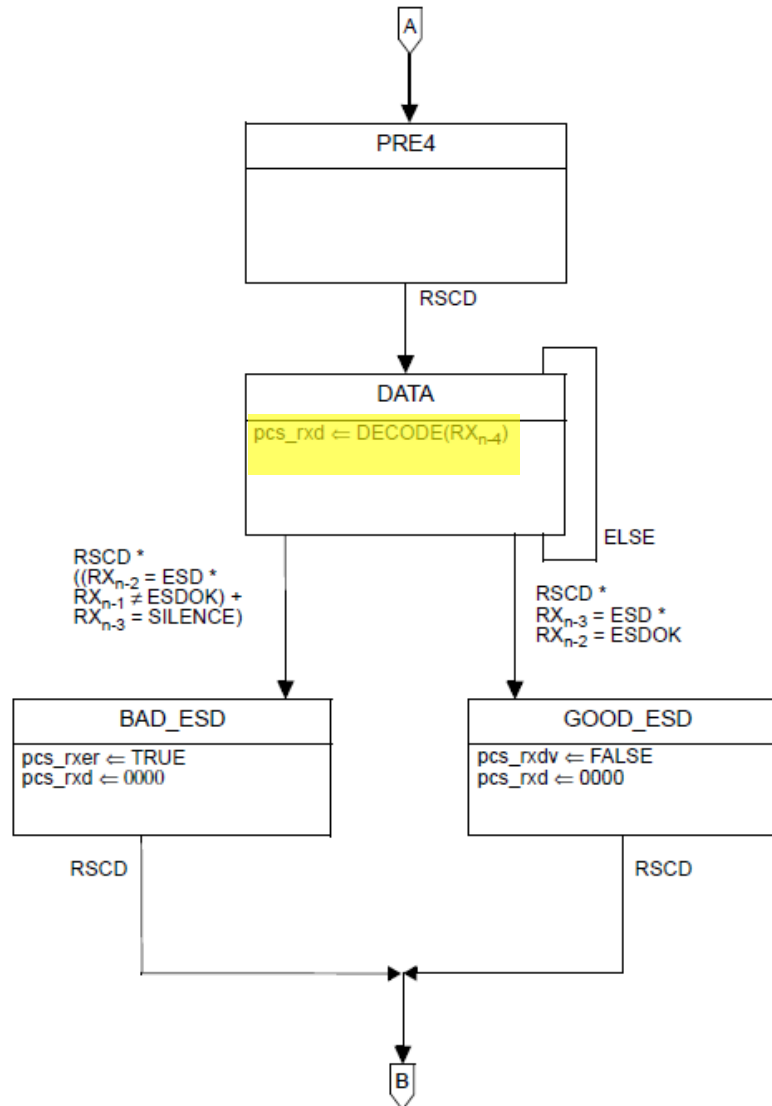
Figure 0–5—Example DME encoding of 5B symbols
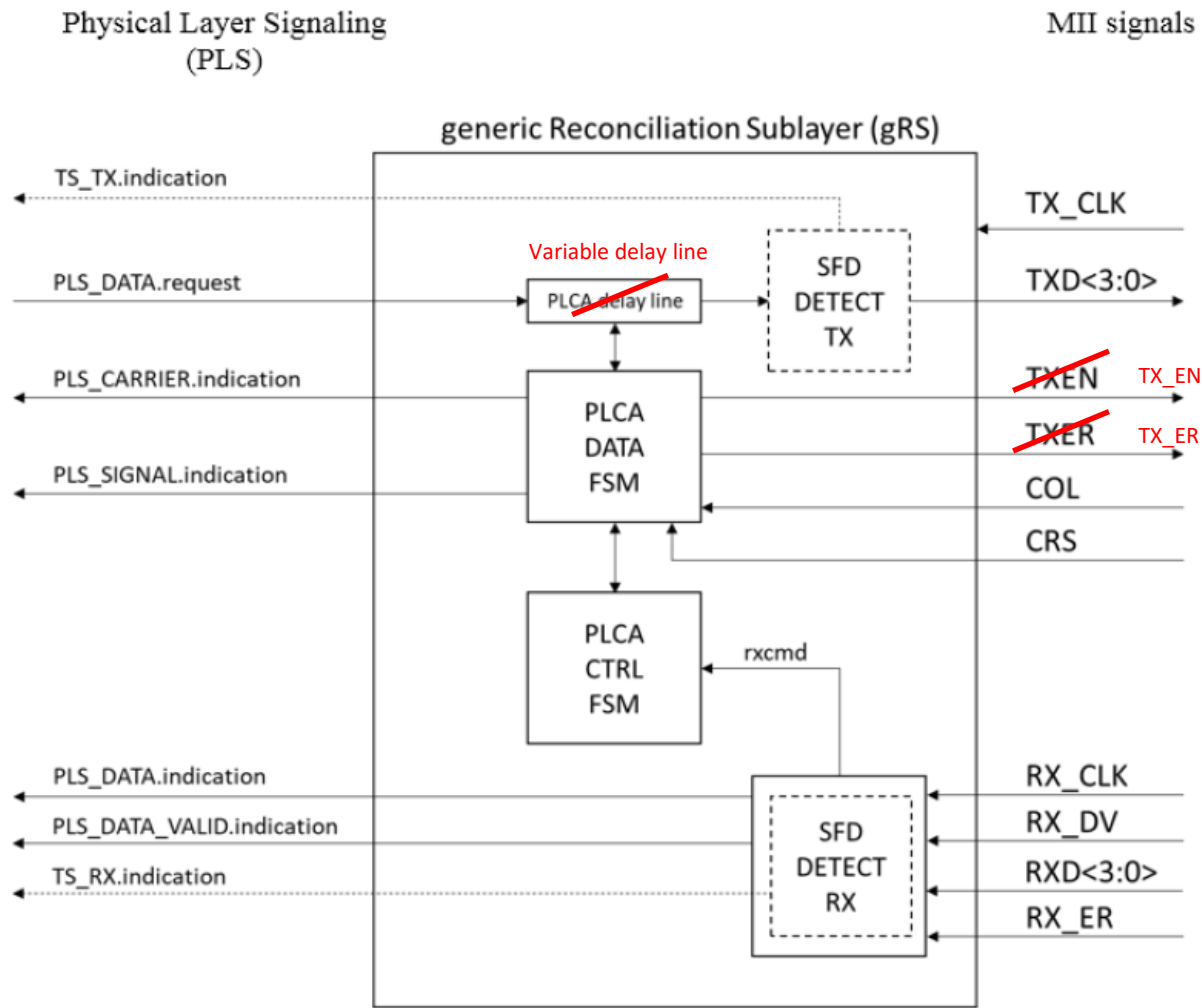
Figure 148–2—PLCA functions within the generic Reconciliation Sublayer (gRS)

plca_reset = ON +
plca_en = OFF

DISABLE

tx_cmd ⇐ NONE
committed ⇐ FALSE
curID ⇐ 0
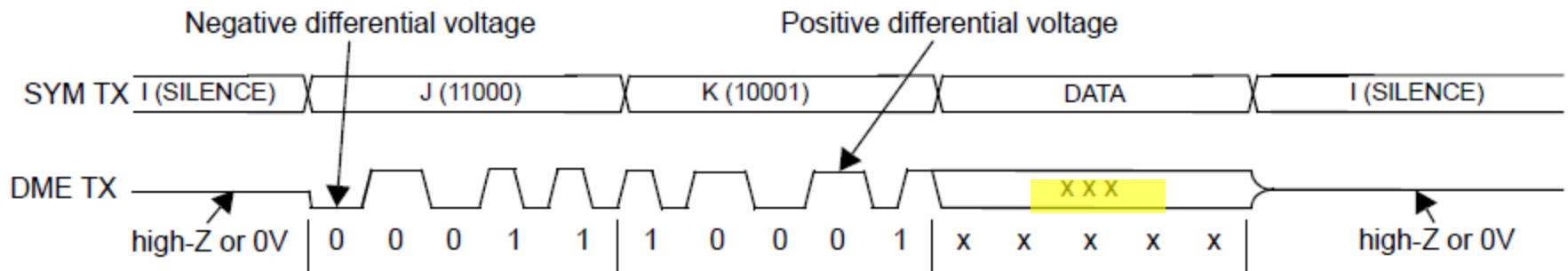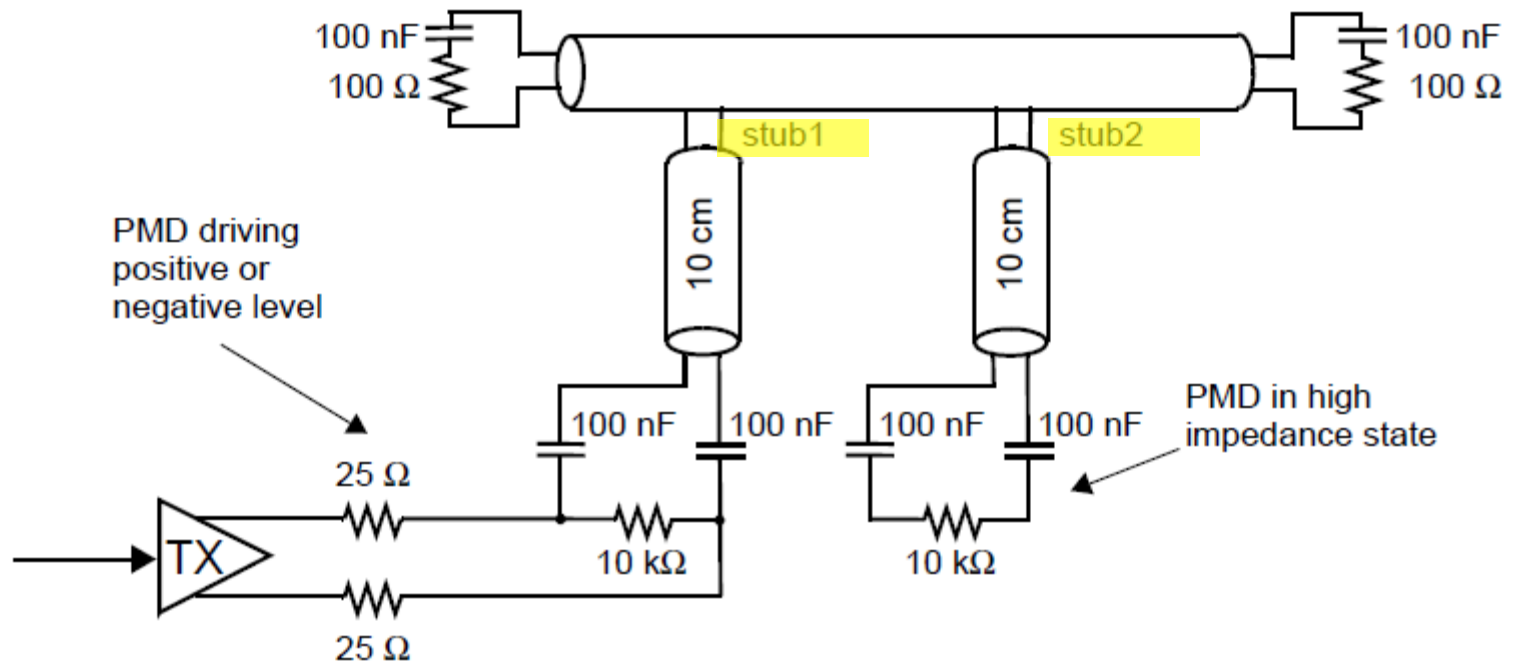
plca_en = ON

plca_eri = TRUE

plca_crs = TRUE

B   C

RESYNC   RECOVER

curID ⇐ 0   restart RECV_BEACON_TMR

myID = 0   RECV_BEACON_TMR done

SEND_BEACON

myID ≠ 0 *
rx_cmd = BEACON   start BEACON_TMR
tx_cmd ⇐ BEACON

BEACON_TMR done

SYNCING

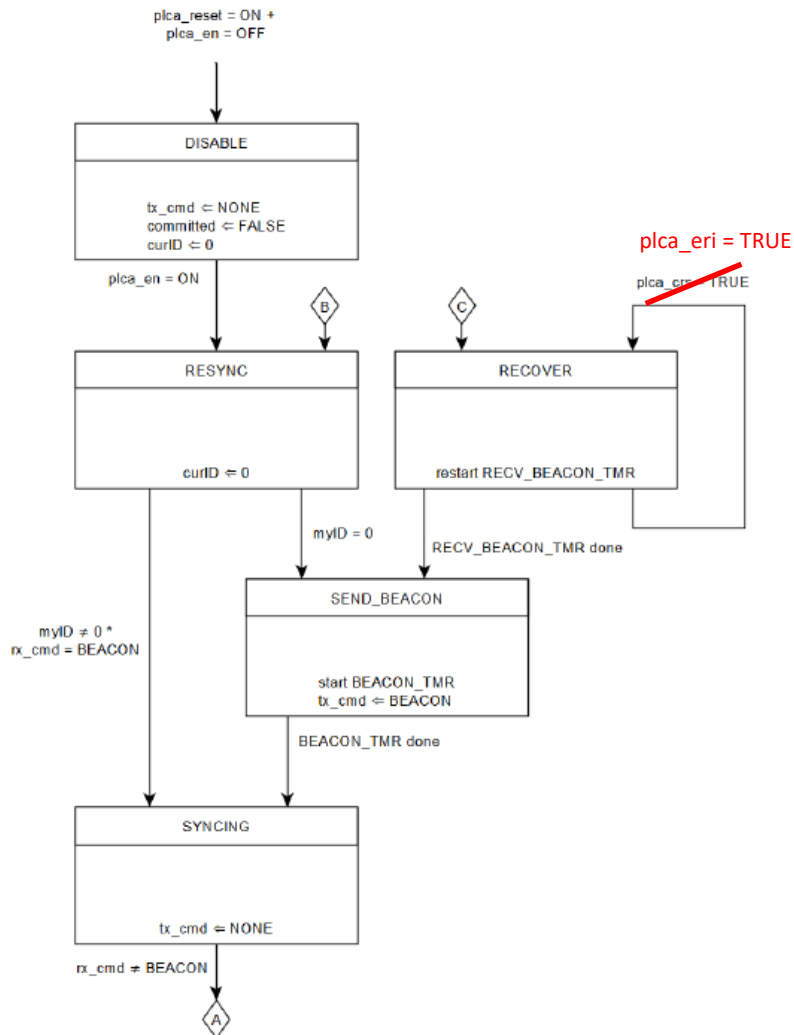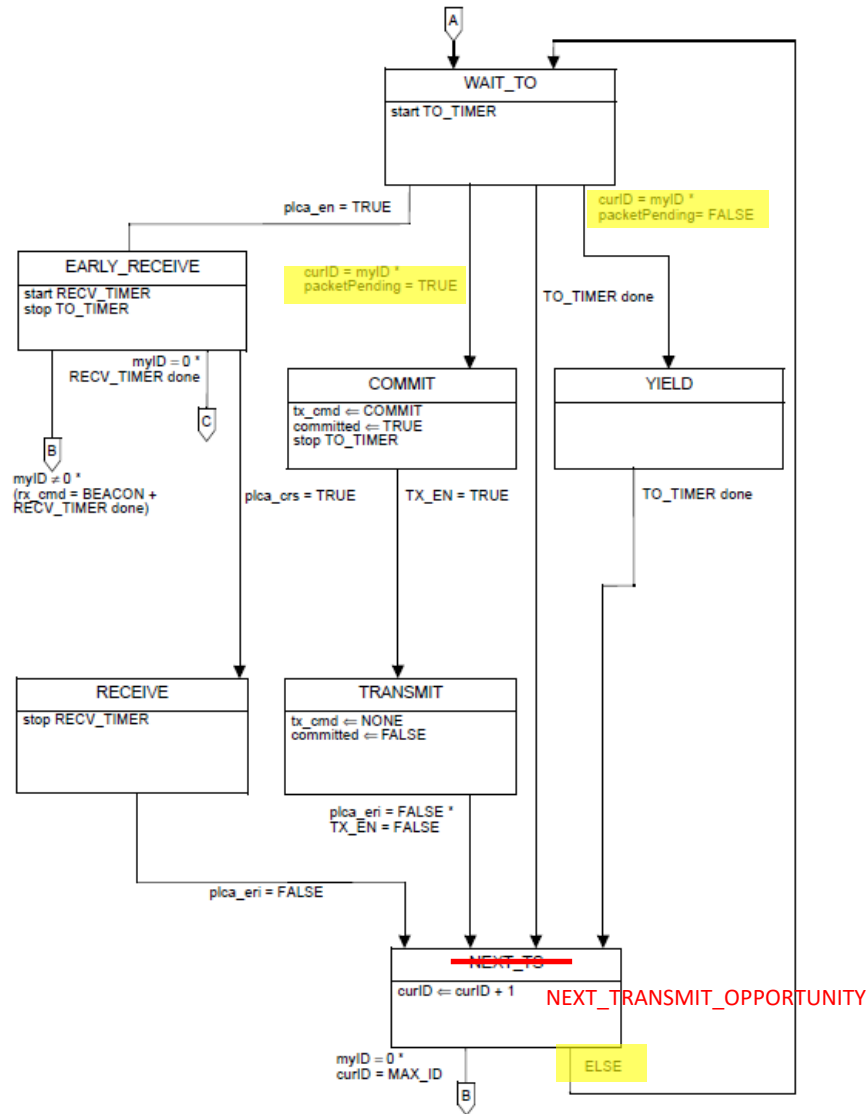tx_cmd ⇐ NONE

rx_cmd ≠ BEACON

A

Figure 148–3—PLCA Control state diagram (part 1 of 2)

In figure 148-3, the transition from RECOVER state to RECOVER state should be done whenever some activity is sensed on the media ("plca_eri"), not only when a good receiving is ongoing ("plca_crs"). This to avoid collision when BEACON is sent

A

WAIT_TO
start TO_TIMER

plca_en = TRUE

curID = myID *
packetPending= FALSE

EARLY_RECEIVE
start RECV_TIMER
stop TO_TIMER

curID = myID *
packetPending = TRUE

TO_TIMER done

myID = 0 *
RECV_TIMER done

COMMIT
tx_cmd ⇐ COMMIT
committed ⇐ TRUE
stop TO_TIMER

YIELD

B

C

myID ≠ 0 *
(rx_cmd = BEACON +
RECV_TIMER done)

plca_crs = TRUE

TX_EN = TRUE

TO_TIMER done

RECEIVE
stop RECV_TIMER

TRANSMIT
tx_cmd ⇐ NONE
committed ⇐ FALSE

plca_eri = FALSE *
TX_EN = FALSE

plca_eri = FALSE

~~NEXT_TO~~
curID ⇐ curID + 1

NEXT_TRANSMIT_OPPORTUNITY
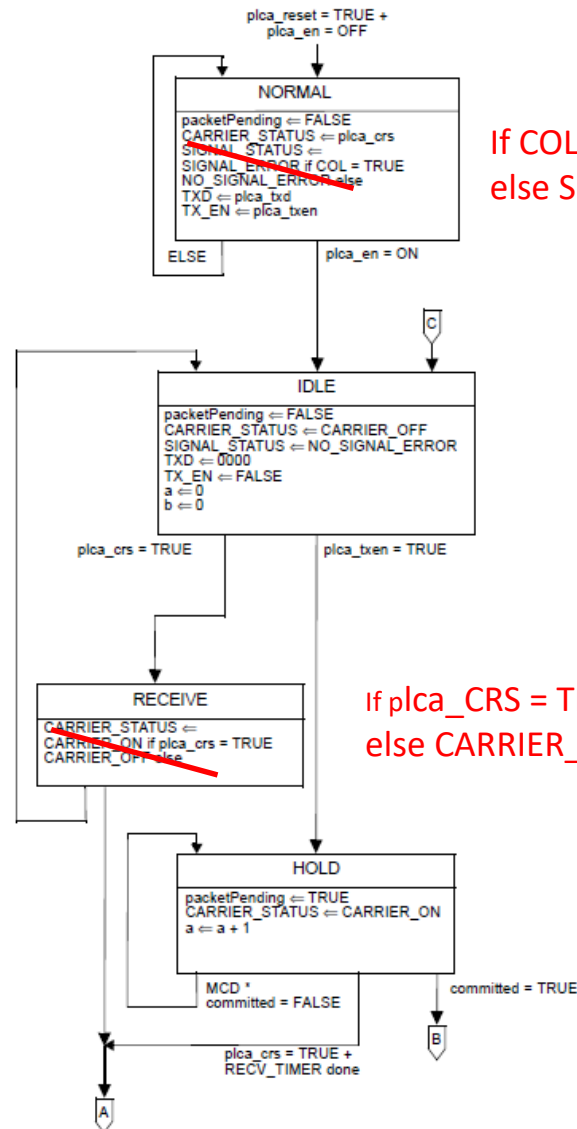
myID = 0 *
curID = MAX_ID

ELSE

B

148.4.5.2:

- MAX_ID can be left unconfigured on slave devices, myID shall not depend on it

myID

ID representing the PLCA transmit opportunity assigned to the PHY. Generated by the management interface (register TBD). May also be set by the Auto-Negotiation protocol as described in Clause 98. The special value '0' is assigned to the master node, indicating the PHY shall generate BEACON signals.

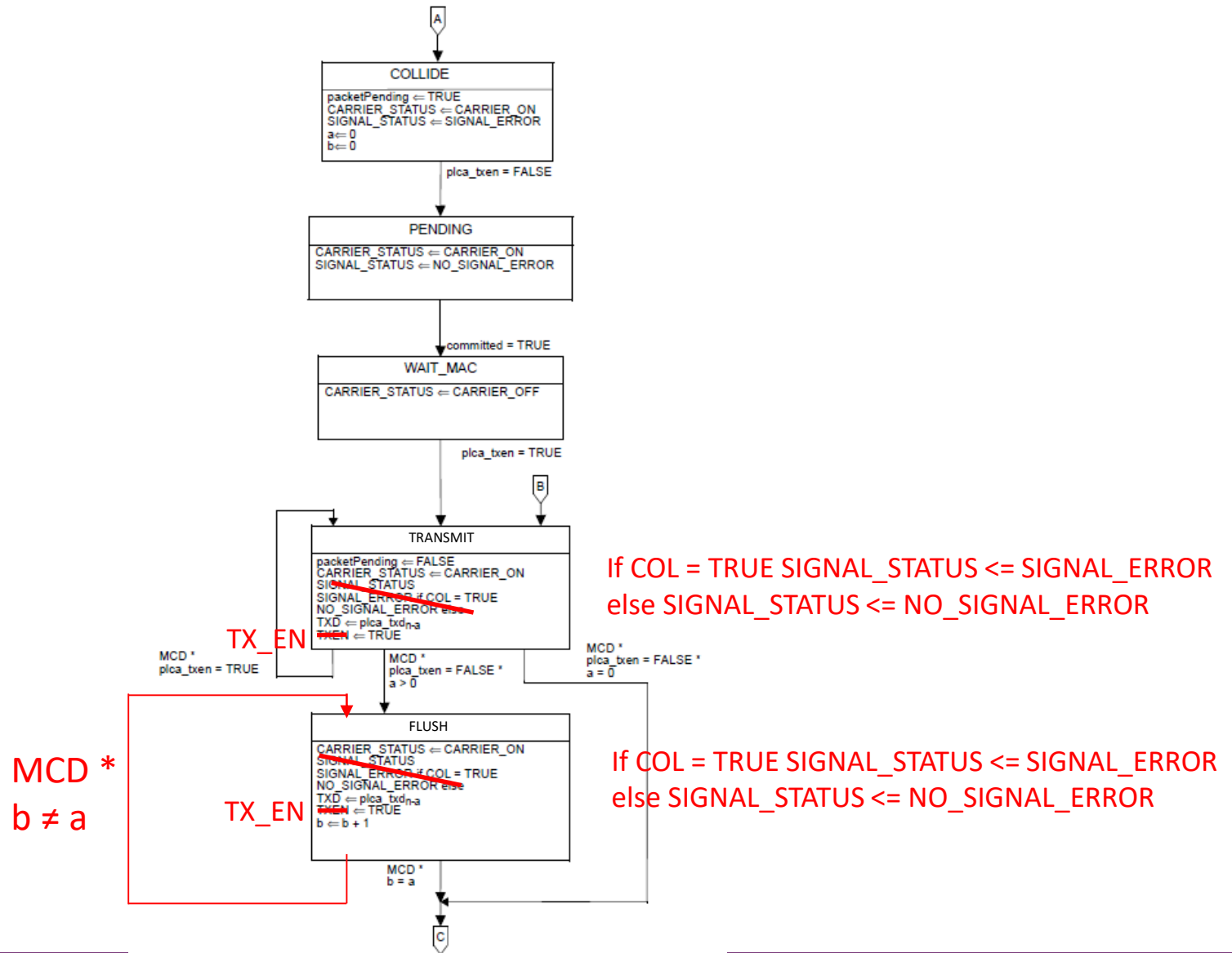Values: integer value from 0 (master) to ~~MAX_ID~~255.

plca_reset = TRUE +
plca_en = OFF

NORMAL
packetPending ⇐ FALSE
CARRIER_STATUS ⇐ plca_crs
SIGNAL_STATUS ⇐
SIGNAL_ERROR if COL = TRUE else
NO_SIGNAL_ERROR
TXD ⇐ plca_txd
TX_EN ⇐ plca_txen

ELSE

plca_en = ON

C

If COL = TRUE SIGNAL_STATUS <= SIGNAL_ERROR
else SIGNAL_STATUS <= NO_SIGNAL_ERROR

IDLE
packetPending ⇐ FALSE
CARRIER_STATUS ⇐ CARRIER_OFF
SIGNAL_STATUS ⇐ NO_SIGNAL_ERROR
TXD ⇐ 0000
TX_EN ⇐ FALSE
a ⇐ 0
b ⇐ 0

plca_crs = TRUE

plca_txen = TRUE

RECEIVE
CARRIER_STATUS ⇐
CARRIER_ON if plca_crs = TRUE
CARRIER_OFF else

If plca_CRS = TRUE CARRIER_STATUS <= CARRIER_ON
else CARRIER_STATUS <= CARRIER_OFF"

HOLD
packetPending ⇐ TRUE
CARRIER_STATUS ⇐ CARRIER_ON
a ⇐ a + 1

MCD *
committed = FALSE

committed = TRUE

B

plca_crs = TRUE +
RECV_TIMER done

A

If COL = TRUE SIGNAL_STATUS <= SIGNAL_ERROR
else SIGNAL_STATUS <= NO_SIGNAL_ERROR

MCD *
b ≠ a

TX_EN

If COL = TRUE SIGNAL_STATUS <= SIGNAL_ERROR
else SIGNAL_STATUS <= NO_SIGNAL_ERROR

Thank You !