IEEE P802.3z

Task Force

Comments and Responses
on Clauses 36 and 37
(PCS, PMA, LC)
of
802.3z Draft 2

Prepared by:
Rich Taborek, Amdahl
revision date
March 9, 1997

PCS/PMA/LC Comment Resolution

Comments received on 802.3zD2 clauses 36 and 37 are separated into three categories for resolution. All comments received during 802.3z and related meetings, via email, phonemail, etc. are all acceptable. However, comments sent in by email to the 802.3z reflector are preferred.

The three categories are as follows:

I = Issue: Major technical problems the resolution of which probably requires committee decision.

T = Technical: Technical errors where it is believed that there is full agreement as to the function in question.

E = Editorial: Errors of expression where the technical meaning is clear.

In several instances, original comments containing more than a single issue have been divided into two or more comments to allow separate responses to those issues (e.g.accepted, rejected, etc. responses to multiple issues contained in a single original comment.

Comments and associated responses are further grouped into PCS, PMA, and LC sections and consecutively numbered in each section. Comments in each section are numbered according to the date received.

802.3zD2 PCS Comments and Responses

1. (I) GMII Management Functional Requirements (35.2.4)

Source/Date: Interim Meeting, PCS Track comment, 1/27-28/97

Deferred from D1. What controls and status information requirements exist for the GMII management interface? The possibilities include:

a) RESET ➨
b) LOOPBACK ➨
c) POWER DOWN ➨
d) OFFLINE ➨
e) TX_DISABLE ➨
f) ISOLATE ➨
g) COLLISION TEST ➨
h) ➦ FAULT
i) ➦ SIGNAL DETECT

Response: Open.

• Reset: Add support: Defined as GMII Control Register (Register 0) bit 0.15. Add effect om transmission order to 36.3.4.2.

• Loopback: Add support: connected to PMA EWRAP. Defined as GMII Control Register (Register 0) bit 0.14. Define PCS_control.indicate primitive.

• Power Down: Add support: Defined as GMII Control Register (Register 0) bit 0.11

• Offline: Open: Not defined in GMII

• TX_Disable: Delete: Little if any support from PMD track. Not defined in GMII. Not required by PCS, PMA, or LC.

• Isolate: Not applicable. Defined as GMII Control Register (Register 0) bit 0.10. At speeds above 100 Mb/s, bit 0.10 shall have no effect on the PHY.

• Collision Test: Add support: Defined as GMII Control Register (Register 0) bit 0.7. Associated with Loopback.

• Fault: Delete: Little if any support from PMD track. Not defined in GMII. Not required by PCS, PMA, or LC.

• Signal Detect: Delete: Support from PMD deleted by motion. Not defined in GMII. Not required by PCS, PMA, or LC.

2. (T) Figure 36-5, Transmit ordered_set state diagram

Source/Date: Ben Brown, Cabletron, 2/19/97; Dalit Sagi, GEC Plessey, 2/19/97

I'm not sure if anyone is working on the transmit process but I have a question. In the DATA state, TX_EN & TX_ER go true for the first byte of preamble then TX_ER goes false for the remainder of the packet. (I don't know why, let's just say it does.) If the first byte of the packet is odd-byte aligned so that sentO_Set.indicate is false for the first byte and true for the second byte when TX_ER is false, the transmit process will proceed to START_OF_PACKET state and the receiving PCS will never know that there was an error with this packet. Is this the desired operation?

Response: Accepted. Added a new state, “ALIGN_ERR_START” upon exit from the “DATA” state when “TX_EN=TRUE * TX_ER=TRUE”. This state transitions to the “START_ERROR” state upon “sentO_Set.indicate”.

3. (T) Figure 36-7, PCS Receive state diagram

Source/Date: Ben Brown, Cabletron, 2/20/97

The UCT from EARLY_END and EPD_ERROR should actually be PMA_UNITDATA.indicate.

Response: Accepted. The “EPD_ERROR” state has subsequently been subsumed into the “EXTEND_ERR” state which already had a “PMA_UNITDATA.indicate” include in all outputs.

4. (T) Figure 36-5, Transmit ordered_set state diagram

Source/Date: Ben Brown, Cabletron, 2/20/97

On the transition from DATA to START_OF_PACKET, I don't believe you need the even=FALSE condition as sentO_Set.indicate condition covers this.

Response: Accepted.

5. (T) Figure 36-5, Transmit ordered_set state diagram

Source/Date: Ben Brown, Cabletron, 2/20/97

COL should be assigned the value FALSE in END_OF_PACKET_NOEXT state since transmitting equals FALSE here.

Response: Accepted.

6. (T) Figure 36-5, Transmit ordered_set state diagram

Source/Date: Ben Brown, Cabletron, 2/20/97

The transition from EPD2_NOEXT state to A should have the condition even=FALSE not even=TRUE.

Response: Accepted.

7. (T) Figure 36-6, Transmit code_group state diagram

Source/Date: Ben Brown, Cabletron, 2/20/97

Remove ALIGNMENT_TEST and ALIGN_IDLE_TO_EVEN states since EPD3 sets tx_o_set to /R/ always.

Response: Accepted.

8. (T) Figure 36-7, PCS Receive state diagram

Source/Date: Ben Brown, Cabletron, 2/20/97; Dalit Sagi, GEC Plessey, 2/20/97

In the DATA_ERROR state, the assignment of RXD<7:0> should be undefined including not specifying the requirement to use the DECODE(rx_code_group) function. The only parameter that need be specified in the DATA_ERROR state is RX_ER=TRUE.

Response: Accepted.

9. (T) Figure 36-7, PCS Receive state diagram

Source/Date: Dalit Sagi, GEC Plessey, 2/20/97

The way the EXT_ERR is right now is that once we had any error during EXT time, until an /I/ arrives we will stay there. I do not see a problem with it, but I guess it should be written in the text as well, and we should agree on it. One change should be that an /S/ will take us out, otherwise we will throw the full burst on one error in the extend!

Response: Accepted.

10. (T) Figure 36-5, Transmit ordered_set state diagram

Source/Date: Ben Brown, Cabletron, 2/21/97

It seems as though it must be true that you could actually spend 0 time in state ALIGN_ERR_START. Do we want to show it this way? The transition from DATA to ALIGN_ERR_START is the only transition without sentO_Set.indicate. I hope this doesn't result in confusion for people. I'll suggest a more clear set of transitions would have TX_EN=TRUE * TX_ER=TRUE * sentO_Set.indicate going to ALIGN_ERR_START state and TX_EN=TRUE * TX_ER=TRUE * sentO_Set.indicate going to START_ERROR state.

Response: No Change.

11. (T) Figure 36-5, Transmit ordered_set state diagram

Source/Date: Ben Brown, Cabletron, 2/21/97

When this machine is in the CARRIER_EXTEND state and TX_EN=FALSE * TX_ER=FALSE * sentO_Set.indicate, shouldn't the transition take you to EXTEND_BY_1? The way this looks, packets with 0, 1 or 2 Carrier Extends from the MAC will look identical to the PMA. The following table should describe what the Ends should look like for a given packet ending from a MAC:

```
MAC Output                                      PCS Output
=============================================================
Data followed by IFG                            /T/ then 1 or 2 /R/s
Data followed by 1 Carr-Ext followed by IF\
```

Response:

Response: G /T/ then 2 or 3 /R/s

```
Data followed by 2 Carr-Ext followed by IFG  /T/ then 3 or 4 /R/s
Data followed by 3 Carr-Ext followed by IFG  /T/ then 4 or 5 /R/s
Data followed by 4 Carr-Ext followed by IFG  /T/ then 5 or 6 /R/s
```

Response: Accepted.

12. (T) Figure 36-7, PCS Receive state diagram

Source/Date: Dalit Sagi, GEC Plessey, 2/20/97

Response: The EXTEND_ERR state goes only to start, idle, and it does not change the RX_DV assignment it can cover for both states now. The only problem I see is that we force a RX_DV=TRUE, RX_ER=TRUE and RXD='h1F, which I do not like (even if allowed).

RX_DV=TRUE while in EXTEND_ERR if EXTEND_ERR were entered from EPD1_CHECK_END. Since RX_ER=TRUE, the MAC still thinks it's receiving packet data. We're probably receiving /R/'s and possibly /junk/ while here. We'll wait for /I/ or /S/ to get out. If we get an /S/, RX_DV will never transition, and the second packet will likely be corrupted. I believe it would be proper to deassert RX_DV if /R/ is received while in EXTEND_ERR.- Rich

Response: Accepted. Added the term “RX_DV = FALSE” to state EXTEND_ERR.

13. (T) Figure 36-7, PCS Receive state diagram

Source/Date: Ben Brown, Cabletron, 2/21/97

The states EARLY_END and EPD_ERROR go back to IDLE immediately while state EXTEND_ERR stays there until it sees an /I/ or /C/ or /S/. Are we being inconsistent, especially between EPD_ERROR and EXTEND_ERR?

Response: Accepted. State “EPD_ERROR” deleted and the leftmost output of block “EPD1_CHECK_END” now goes to “EXTEND_ERR”.

14. (T) Figure 36-7, PCS Receive state diagram

Source/Date: Linda Chen, Sun, 2/25/97

There is a transition from EPD1_CHECK_END to EXTEND_ERR on the condition that (check_end ≠ /R/R/R/ + / R/R/K28.5/ + /R/R/S/) + (check_end = /R/R/K28.5/ * even = FALSE).

However, Since EPD1_CHECK_END takes zero time, and to enter that state rx_code_group must = /T/, the condition as written above will never check for errors such as /T/Junk/Junk/. The error checks for end of packet delimiter should be in a different state block than error checks during extension.

You can use check_end ≠ (/T/R/R/ + /T/R/K28.5/ * even = TRUE) as a transition into EPD_ERROR which you crossed out. I think it should stay.

Response: No Change. The latest Receive state diagram (txrxsmd2_plus on the editors site) has EPD1_CHECK_END going to EXTEND_ERR on the condition (check_end ≠ /T/R/R/) + / T/R/K28.5/ + (check_end = /T/R/K28.5/ * even=FALSE). This will check for /T/ junk/junk/. I don't believe we need the old EPD_ERROR state.

15. (T) Figure 36-6, Transmit code_group state diagram

Source/Date: Linda Chen, Sun, 2/25/97

This is a question. In the Transmit code_group state diagram there is some states for Link configuration as we knew it prior to the San Diego meeting. Who has the action item to update this for “Auto-Fibre Negotiation”?

Response: Accepted. AFN should use all PCS TX and RS state machines.

16. (T) Figure 36-5, Transmit ordered_set state diagram

Source/Date: Linda Chen, Sun, 2/25/97

It looks like the Transmit state machine makes some fragments into packets.

Consider the case where a 511 byte packet is sent. (7 bytes of preamble, 1 byte SFD, 64 bytes packet, 439 bytes Extension).

We would get

```
ev od    ev od  ev od .. od  ev od ev   od     ev     od
p1 p2 .. p7 sfd d1 d2 .. d64 r1 r2 r439 end_r1 end_r2 end_r3
```

end_r1 comes from EXTEND_BY_1 state
end_r2 comes from EPD2_NOEXT state
end_r3 comes from EPD3 state

This fragment will then be treated as a 512 byte packet at the receiver when end_r2 and end_r3 get stripped off, but end_r1 is interpreted as part of the regular extension.

I think it would work better if EXTEND_BY_1 was entered on the condition that a preamble byte was consumed. That is the only condition that would create the 3 ending R's case after tx_er and tx_en go low. After tx_er and tx_en go low, you want to add one or two R's depending on alignment, unless a preamble byte was consumed in which case you want to add two or three R's. This is because PCS needs to preserve the length of the carrier event which the MAC signalled.

Response: No Change.

Steve Haddock: It seems to me that the PCS extending a 511 byte fragment to a 512 byte packet is unavoidable without putting a byte counter in the PCS. The PCS has to do something at the end of an odd length packet to create the even alignment for the next Idle. Its options are pretty limited. Dropping a symbol rather than adding one would be a possibility, but it means we would have to accept /T/I/ as a valid end delimiter. We could define a new symbol that is used only for evenizing, so the receiver can distinguish it from intended carrier extension, but that seems like overkill for this case.

I don't see a danger in allowing a 511 byte fragment being extended to an apparently valid packet as long as there is no situation where a MAC would attempt to retransmit the packet that came out as a 511 byte fragment. The only case I can think of where the MAC might attempt a retransmission would be IF we change our minds and allow retransmission after late collisions in which case we would have to make sure that we jam with something other than /R/ so that the fragment is clearly identifiable by something other than length (because the length may end up being longer than 512).

Linda Chen: I agree that if the solution for collision, including late collisions is to Jam with Error or anything besides R then we are ok at the receiver end. So Clause 4 GOE's ought to remember that we are relying on this capability. The status at then end of San Diego was that the Jam can be anything and the problem would be solved if the MAC didn't retransmit. But I think thinking has since shifted to Jam with Error which would make this problem a non-issue.

Jacob Twersky: Carrier extension ensures 512 byte times of carrier from the Destination Address field of the packet. Therefore, for a 64 byte packet, the MAC will add 448 extend bytes (not 440) so that the total carrier duration *with the preamble* will be 520 byte times when no preamble byte was consumed, or 519 byte times when one preamble byte was consumed. In both cases it's not a fragment.

In the case where the MAC transmitted less extension bytes due to a collision during the extension, this is easily detected by the receiver because the transmitting MAC will JAM with an "extentionError" bytes.

17. (T) Preamble Alignment

Source/Date: Ben Brown, Cabletron, 2/26/97

One thing that seems to keep coming up is the fact that preamble alignment can or should somehow affect how a packet is ended. I just can't seem to understand this concept. Can you explain it to me? Packets can start even or odd aligned and so may or may not lose a byte of preamble. Those same packets may be even or odd length so by the time you get to the end of the packet, regardless of preamble alignment, you can still be even or odd aligned.

Response: No Change.

Linda Chen: When a preamble byte is consumed because tx_en came on an odd boundary, then the carrier event is reduced in length by one byte.

From say 7 preamble + 1 SFD + 64 pkt + 440 extend = 512 byte times to 6 preamble + 1 SFD + 64 pkt + 440 extend = 511 byte times

You just have to make sure that the PCS puts a compensating R out to make up for the preamble removal done at the beginning of the packet. I believe that the current Tx state machine achieves this with the EXTEND_BY_1 state. If a preamble is consumed the state machine traverses EXTEND_BY_1, EPD2_NOEXT, and EPD3 states. If no preamble is consumed then the state machine traverses only EXTEND_BY_1 and EPD2_NOEXT states. This is because if no preamble is consumed the R sent in state EXTEND_BY_1 is always even. The only thing which could throw off the alignment is consumption of the preamble byte.

18. (T) State Diagram Interpretation

Source/Date: Linda Chen, Sun, 2/26/97

I think we need someone with previous standard experience to comment on how these diagrams are to be read. I see conditions which bring you into a box to be still true in that box. Whereas you see conditions, such as “even=true” in the example below, to evaluate as “even=false” when in box EPD3.

```
+--------------+
| EPD2_NOEXT   |
| tx_o_set<= R |
+--------------+
   | even = true
   v
+--------------+
|    EPD3      |
| tx_o_set<= R |
+--------------+
```

Response: Accepted. All cases checked. Defined messages cause state transitions. The SentO_Set.indicate is the message for the above case.

19. (T) Figure 36-7, PCS Receive state diagram

Source/Date: John Wolcott, Intel, 2/28/97

We've been looking over the PMA Receive State Diagrams (figure 36-7) and have a few questions which relate to a packet EPD of TRRI.

If you have TRRI as an EPD you will drop into the “TRR + EXTEND” state on receipt of a “T” and detection of the “TRR” code_groups (via the “check_end” function). At this point RX_ER goes TRUE immediately (RX_DV=FALSE and RXD=0x0F).

Then upon receipt of the first “R” and detection of “RRI” as the next 3 code_groups in the “EPD2_CHECK_END” state, you drop into the “TRI + RRI” state. A this point RX_DV and RXD remain unchanged, but RX_ER is set to FALSE.

It seems that there is an indication of Carrier Extend for one octet time (RX_ER=TRUE and RX_DV=FALSE and RXD=0x0F per Table 35-2).

This appears to be an error. One way to correct this RX_ER “error pulse” would be to expand the “check_end” function to observe 4 octets for the TRRI condition prior to asserting RX_ER... latency issues???

Have we misinterpreted this flow?

Response: No change. Your interpretation of the PCS Receive state diagram is correct. For the /T/R/R/I/ case, the PCS will reflect one octet of carrier extension to the receiving MAC. This is regardless of whether the transmitting MAC requested one octet or carrier extension or the transmitting PCS added one extra /R/ for alignment purposes. The receiving PCS cannot distinguish these two cases and it's my understanding that the receiving MAC also need not distinguish these two cases.

20. (E) Figure 36-7, PCS Receive state diagram

Source/Date: Rich Taborek, Amdahl, 2/28/97

There’s and extraneous “rx_code_group = SPD” in the transition condition “rx_code_group = SPD * PMA_UNITDATA.indicate” from the “DATA” and “DATA_ERROR” states to the “RECEIVE” block.

Response: Accepted.

21. (E) Figure 36-1, Relationship of 1000BASE-X and the PMDs

Source/Date: PCS/PMA/LC Working Meeting, Amdahl, 3/3/97

The dotted line from the top of the OSI Reference Model Layer should extend to the top pf the LAN CSMA/CD LLC Layer.

Response: Accepted.

22. (E) Figure 36-2, Functional block diagram

Source/Date: PCS/PMA/LC Working Meeting, Amdahl, 3/3/97

The arrow from the PCS TRANSMIT block to the PCS RECEIVE block should be reversed. An arrowhead should be added to the line from the PMA RECEIVE block to the PCS LINK MONITOR block.

Response: Accepted.

23. (E) Table 36-3, Defined ordered_sets

Source/Date: PCS/PMA/LC Working Meeting, Amdahl, 3/3/97

The font size used for note 3 is too large.

Response: Accepted.

24. (E) 36.2.4.14.1 Carrier_Extend (/R/)

Source/Date: PCS/PMA/LC Working Meeting, Amdahl, 3/3/97

For clarity, add a purpose summary to points a, b, and c.

Response: Accepted.

25. (E) 36.2.4.14.1 Carrier_Extend (/R/)

Source/Date: PCS/PMA/LC Working Meeting, Amdahl, 3/3/97

“extend” is misspelled as “entend” in the third line of item a)

Response: Accepted.

26. (E) 36.2.4.16 Interpacket gap (IPG) considerations

Source/Date: PCS/PMA/LC Working Meeting, Amdahl, 3/3/97

The third sentence of this subclause constitutes a rule and should be placed ahead of the first two sentences, which constitute examples.

Response: Accepted.

27. (E) Figure 36-4, PCS Encapsulation

Source/Date: PCS/PMA/LC Working Meeting, Amdahl, 3/3/97

/I/ should be shown as two code_groups spanning two GTX_CLK periods.

Response: Accepted.

28. (E) 36.2.5.1 Constants, Kx.y

Source/Date: PCS/PMA/LC Working Meeting, Amdahl, 3/3/97

Kx.y is one of the set of 12 code_groups, not 256.

Response: Accepted.

29. (E) 36.2.5.2 Variables

Source/Date: PCS/PMA/LC Working Meeting, Amdahl, 3/3/97

In the first sentence and for CRS, the reference should be to clause 35 (GMII), not 22.

Response: Accepted.

30. (E) 36.2.5.2 Variables, receiving

Source/Date: PCS/PMA/LC Working Meeting, Amdahl, 3/3/97

Reword “non-IDLE and non-/C/ activity” to “carrier activity”. Add reference to the carrier_detect(x) function in 36.2.5.3.

Response: Accepted.

31. (T) 36.2.5.2 Variables, rcv/C/

Source/Date: PCS/PMA/LC Working Meeting, Amdahl, 3/3/97

Add a variable to the PCS Receive process which indicates the receipt of (/C/) while xmit = DATA. Variable to be used by the Link Configuration process.

Response: Accepted.

32. (E) 36.2.5.2 Variables, restart_config

Source/Date: PCS/PMA/LC Working Meeting, Amdahl, 3/3/97

Delete the “restart_config” variable as it is defined (as mr_restart_negotiation) in clause 37.

Response: Accepted.

33. (E) 36.2.5.2 36.2.5.2 Variables, transmitting

Source/Date: PCS/PMA/LC Working Meeting, Amdahl, 3/3/97

Reword “non-IDLE and non-/C/ activity transmission in progress” to “that packet transmission is in progress”. For the values, change “TRUE; the PCS’s client is transmitting” to “TRUE; the PCS is transmitting a packet”, and “FALSE; the PCS’s client is not transmitting” to “TRUE; the PCS is not transmitting a packet”.

Response: Accepted.

34. (E) 36.2.5.3 Functions, ENCODE

Source/Date: PCS/PMA/LC Working Meeting, Amdahl, 3/3/97

“an GMII” should be “a GMII”

Response: Accepted.

35. (E) 36.2.6.1.1 Transmit ordered_set

Source/Date: PCS/PMA/LC Working Meeting, Amdahl, 3/3/97

Change most occurrences of “IDLE” corresponding to “/I/” to prevent confusion with the “xmit” flag value “IDLE”.

Response: Accepted.

36. (E) 36.2.6.1.1 Transmit ordered_set

Source/Date: PCS/PMA/LC Working Meeting, Amdahl, 3/3/97

Change “DATA” to “IDLE” in the third sentence.

Response: Accepted.

37. (T) Figure 36-6, Transmit code_group state diagram

Source/Date: PCS/PMA/LC Working Meeting, Amdahl, 3/3/97

Reverse the bit ordering of “Config_Reg” bits used in the ENCODE function. Also define “Config_Reg” as a parameter of the ENCODE function.

Response: Accepted.

38. (T) Figure 36-6, Transmit code_group state diagram

Source/Date: PCS/PMA/LC Working Meeting, Amdahl, 3/3/97

Correct the ENCODE parameter in the DATA_GO state to be “TXD<7:0>”.

Response: Accepted.

39. (T) Figure 36-6, Transmit code_group state diagram

Source/Date: PCS/PMA/LC Working Meeting, Amdahl, 3/3/97

Add the term “tx_even ‹= ! tx_even” to states DATA_GO and SPECIAL_GO to provide proper generation of the tx_even flag in all Transmit code_group states.

Response: Accepted.

40. (T) Figure 36-7, PCS Receive state diagram

Source/Date: PCS/PMA/LC Working Meeting, Amdahl, 3/3/97

Delete the term “rx_code_group = /R/” from the exit test in states TRR+EXTEND and PACKET_BURST_RRS.

Response: Accepted.

41. (T) Figure 36-7, PCS Receive state diagram

Source/Date: PCS/PMA/LC Working Meeting, Amdahl, 3/3/97

Add the term “rcv/C/ <= FALSE” to state CONFIGURATION to indicate to the Link Configuration state machine that the PCS Receive state machine is receiving /C/.

Response: Accepted.

42. (T) Figure 36-7, PCS Receive state diagram

Source/Date: PCS/PMA/LC Working Meeting, Amdahl, 3/3/97

Add the state C_CODE containing the same parameters as the state CONFIGURATION so FALSE CARRIER is not reported to the MAC when this situation occurs. This state is also used to set rcv/C/. Enter C_CODE from EARLY_END and IDLE upon detection of the first two code groups of /C/.

Response: Accepted.

43. (T) Figure 36-7, PCS Receive state diagram

Source/Date: PCS/PMA/LC Working Meeting, Amdahl, 3/3/97

Rename the state IDLE to IDLE_K. Add the state IDLE_D containing the same parameters as the state IDLE_K. Enter IDLE_K from IDLE_D and EARLY_END upon detection of an ordered_set which is not /C/ (e.g. /I/), and from C_CODE on an odd_numbered code_group. Enter IDLE_D from IDLE_K and FALSE_CARRIER when carrier is not detected and from states TRI+RRI and EXTEND_ERR when K28.5 is detected.

Response: Accepted.

44. (T) Figure 36-7, PCS Receive state diagram

Source/Date: PCS/PMA/LC Working Meeting, Amdahl, 3/3/97

Delete the state PACKET_BURST_RS. Delete the terms “RX_ER <= TRUE” and “RXD<7:0> <= 0000 1111” from state the term PACKET_BURST_RRS. The exit from PACKET_BURST_RRS should go to the START_OF_PACKET state.

Response: Accepted.

45. (T) Figure 36-7, PCS Receive state diagram

Source/Date: PCS/PMA/LC Working Meeting, Amdahl, 3/3/97

The test to enter CONFIGURATION should be changed to “xmit=(CONFIGURATION+IDLE) * PMA_UNITDATA.indicate”. The test to stay in state CONFIGURATION should be changed to “rx_code_group ≠ K28.5 * xmit=(CONFIGURATION+IDLE+DATA) * PMA_UNITDATA.indicate”. The test to exit state CONFIGURATION to IDLE_D should be changed to xmit=DATA * rx_code_group ≠ K28.5 * PMA_UNITDATA.indicate”.

Response: Accepted.

46. (T) Figure 36-9, Synchronization state diagram

Source/Date: PCS/PMA/LC Working Meeting, Amdahl, 3/3/97

Add the term “rx_even <= TRUE” to states COMMA_DETECT_1, COMMA_DETECT_2 and COMMA_DETECT_3. Add the term “rx_even <= ! rx_even” to states LOSS_OF_SYNC, ACQUIRE_SYNC_1, ACQUIRE_SYNC_2, SYNC_ACQUIRED_1, SYNC_ACQUIRED_2, SYNC_ACQUIRED_3 and SYNC_ACQUIRED_4.

Response: Accepted.

47. (T) Figure 36-9, Synchronization state diagram

Source/Date: PCS/PMA/LC Working Meeting, Amdahl, 3/3/97

The term “link_status <= OK” should be deleted from states SYNC_ACQUIRED_2, SYNC_ACQUIRED_3 and SYNC_ACQUIRED_4.

Response: Accepted.

48. (E) Variables, link_status

Source/Date: Devendra Tripathi, 3/4/97

Like link_control, link_status is also used and defined by clause 28 (although we do not intend to use it). It will be preferable to avoid using this name and change it to some thing related to “sync”.

Response: Accepted. link_status is renamed to sync_status.

49. (T) 36.2.6.1.5 Code_group stream decoding

Source/Date: Rich Taborek, Amdahl 3/5/97

Changed “Premature packet termination is caused by the detection of an IDLE code_group...” in the third paragraph to “Premature packet termination is caused by the detection of a K28.5 code_group...” to be consistent with PCS Receive state machine operation.

Response: Accepted. link_status is renamed to sync_status.

50. (E) Figure 36-7, PCS Receive state diagram

Source/Date: GEA Technical meeting, PCS review 3/6/97

Swap the state names of IDLE_K and IDLE_D so that a K character takes you to IDLE_K instead of from IDLE_K, similar to other state transitions.

Response: Accepted.

51. (T) Figure 36-7, PCS Receive state diagram

Source/Date: GEA Technical meeting, PCS review 3/6/97

Rename the state CONFIGURATION to IN_CONFIG and the state C_CODE to RCV_C_CODE as the former are aliases of each other and do not accurately convey the reason the state was entered.

Response: Accepted.

52. (T) Figure 36-7, PCS Receive state diagram

Source/Date: GEA Technical meeting, PCS review 3/6/97

Delete the setting of RXD in states IDLE_D, IDLE_K, IN_CONFIG, and RCV_C_CODE since its value has no meaning when RX_DV=FALSE and RX_ER=FALSE.

Response: Accepted.

53. (T) Figure 36-7, PCS Receive state diagram

Source/Date: GEA Technical meeting, PCS review 3/6/97

Corrected the loop condition for state IN_CONFIG to "(xmit=CONFIGURATION+ xmit=IDLE + (xmit=DATA * rx_code_group ≠ K28.5)) * PMA_UNITDATA.indicate".

Response: Accepted.

54. (T) Figure 36-7, PCS Receive state diagram

Source/Date: GEA Technical meeting, PCS review 3/6/97

Since the state PACKET_BURST_RS was deleted, the term RX_ER=FALSE must be added to the START_OF_PACKET state.

Response: Accepted.

55. (T) Figure 36-7, PCS Receive state diagram

Source/Date: GEA Technical meeting, PCS review 3/6/97

Change all "even" variables to "rx_even".

Response: Accepted.

56. (T) Figure 36-7, PCS Receive state diagram

Source/Date: GEA Technical meeting, PCS review 3/6/97

Add the term "rx_code_group = /S/" to the exit of state PACKET_BURST_RRS.

Response: Accepted.

57. (T) Figure 36-7, PCS Receive state diagram

Source/Date: GEA Technical meeting, PCS review 3/6/97

The variable RX_ER should be set to FALSE in states IN_CONFIG, and RCV_C_CODE since it is not necessary to communicate the reception or transmission of /C/ to the MAC.

58. (T) Figure 36-9, Synchronization state diagram

Source/Date: GEA Technical meeting, PCS review 3/6/97

If an alignment error occurs, and we don't lose sync, all the SYNC_ACQUIRED_X states only toggle rx_even. It seems that they need to correct rx_even in this case or flag an error.

Response: Accepted. Fixed by treating a misaligned comma in all SYNC_ACQUIRED_X states as an error similar to rx_code_group = INVALID. Also redefined the 2_good_cgs function to treat a misaligned comma as an error.

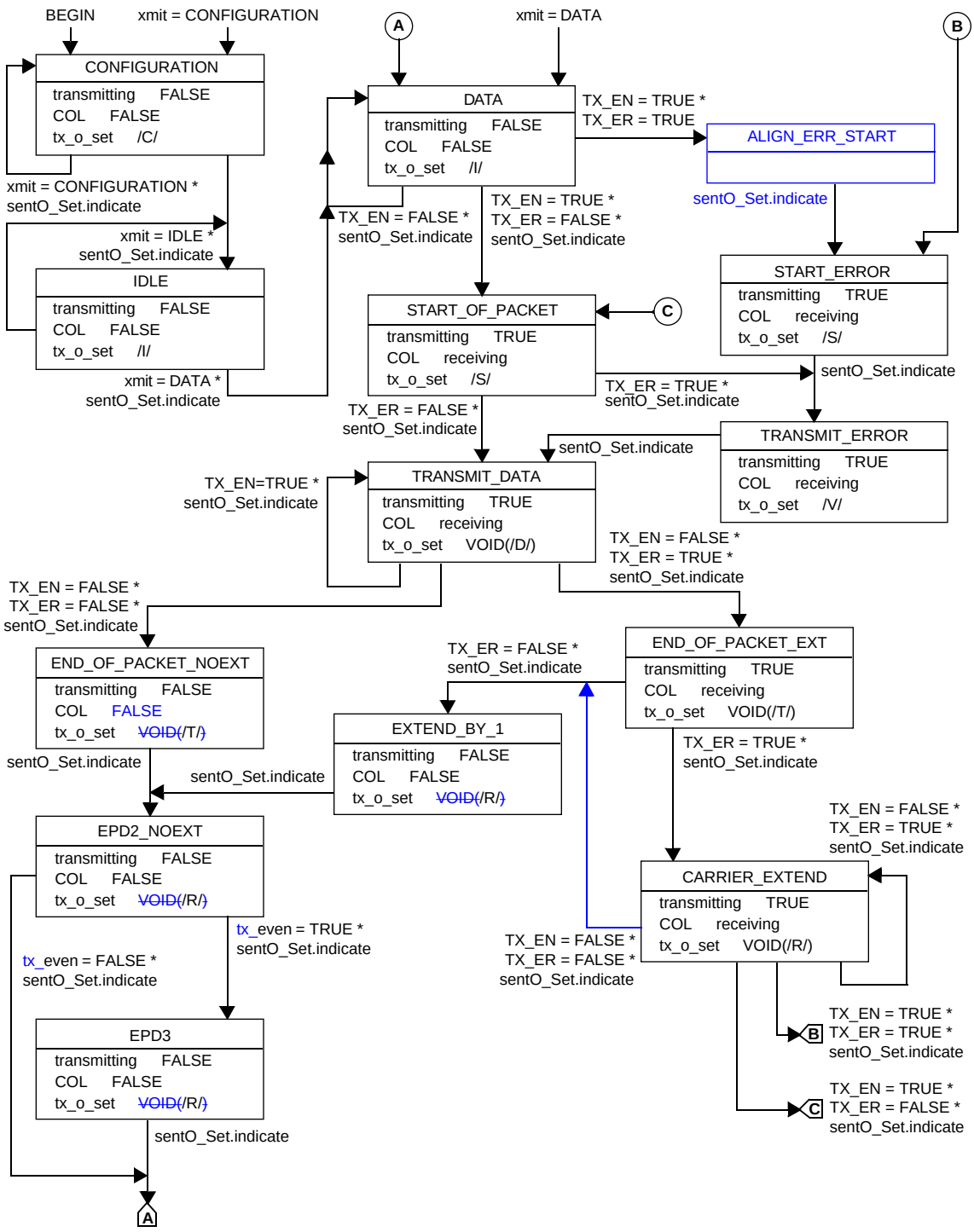

Figure 36-5—Transmit ordered_set state diagram

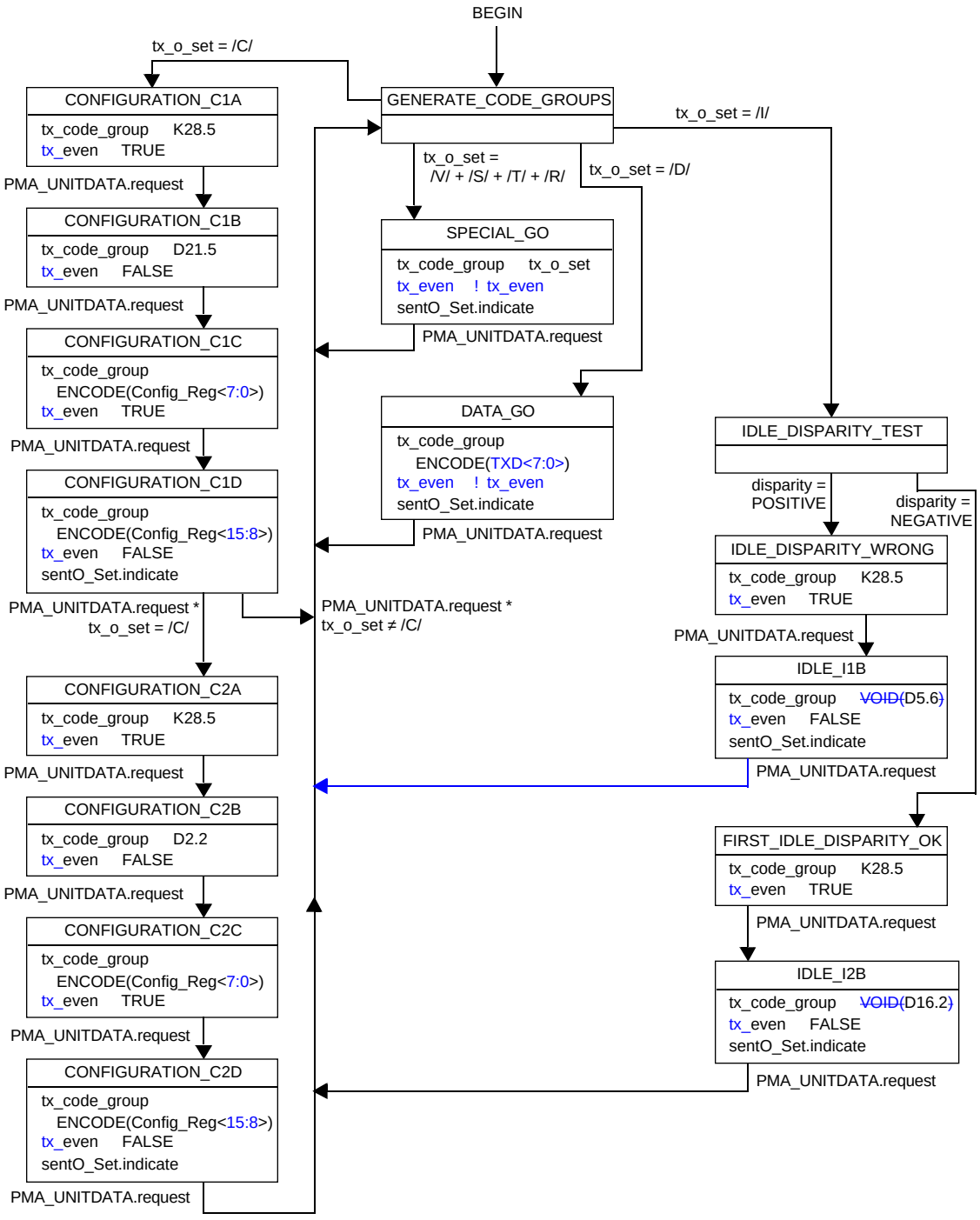

Figure 36-6—Transmit code_group state diagram

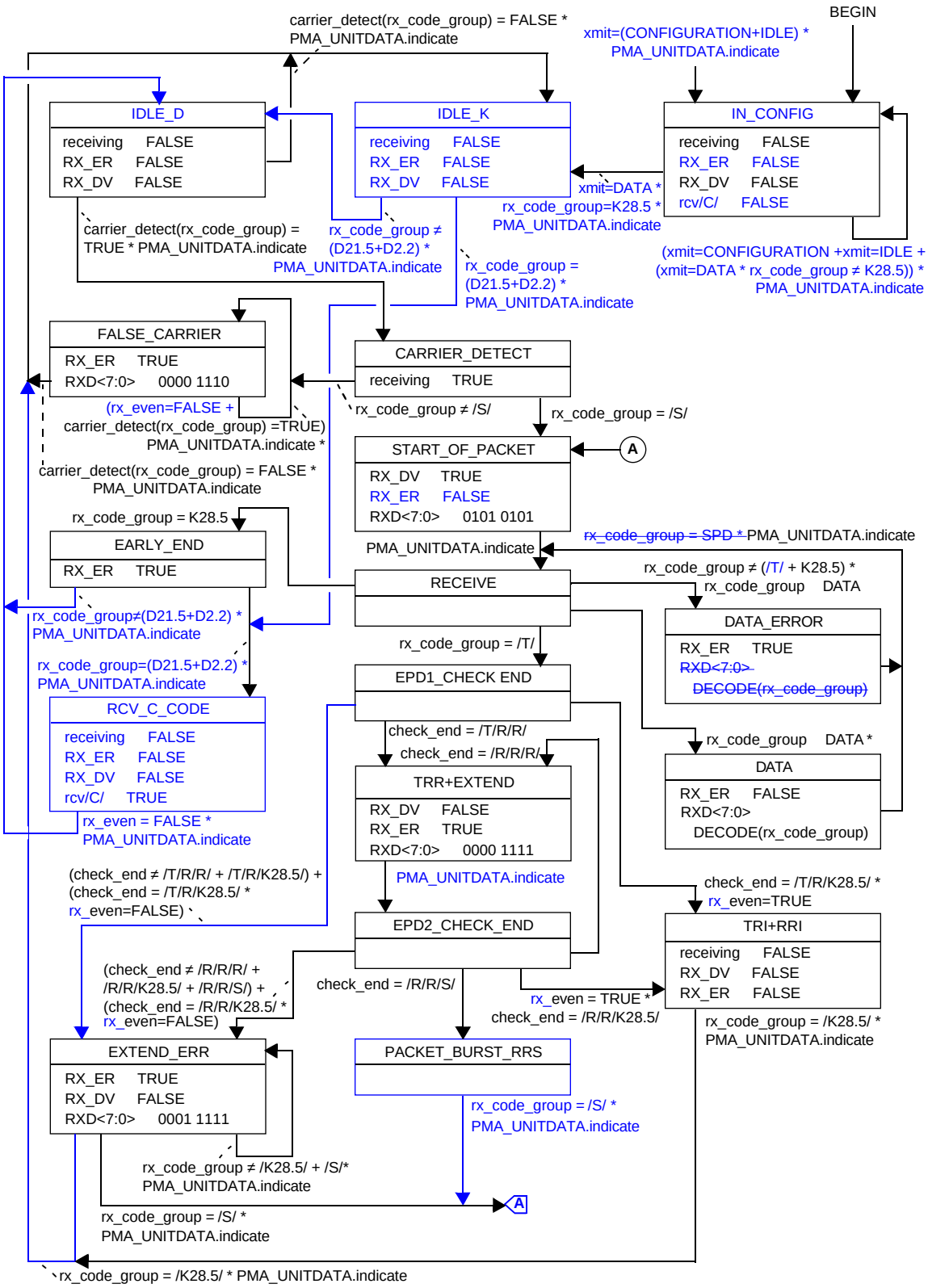

Figure 36-7—PCS Receive state diagram

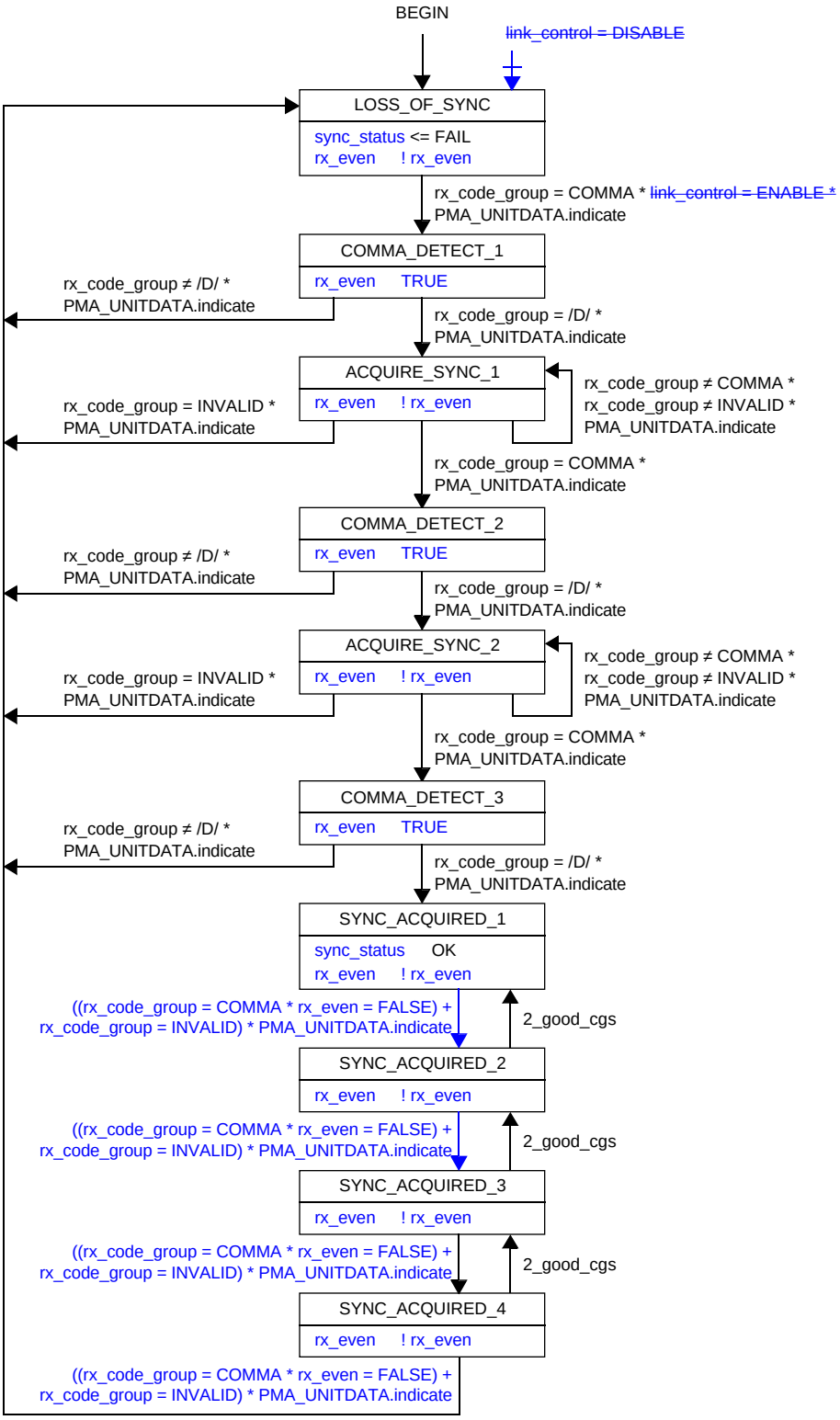

Figure 36-9—Synchronization state diagram

802.3zD2 PMA Comments and Responses

1. (T) 36.2.4.18 GMII clocking

Source/Date: PCS/PMA/LC Working Meeting, Amdahl, 3/3/97

GMII and PMA clocks should be associated, and PMA clocks should be renamed as follows: PMA_TX_CLK is derived from GMII TX_CLK. GMII RX_CLK is derived from PMA_RX_CLK<0> and PMA_RX_CLK<1>.

Response: Accepted.

2. (T) Figure 36-10, PMA reference diagram

Source/Date: PCS/PMA/LC Working Meeting, Amdahl, 3/3/97

The signal EN_CDET is missing from this figure.

Response: Accepted.

3. (E) 36.3.1 Service interface

Source/Date: PCS/PMA/LC Working Meeting, Amdahl, 3/3/97

In paragraph 3, “PMA_UNIT.request...” should be “PMA_UNITDATA.request...”.

Response: Accepted.

4. (E) 36.3.2.3 PMA Receive function

Source/Date: PCS/PMA/LC Working Meeting, Amdahl, 3/3/97

In the second sentence, “tx_bits” should be “rx_bits”.

Response: Accepted.

5. (T) 36.3.2.4 Comma detect function

Source/Date: PCS/PMA/LC Working Meeting, Amdahl, 3/3/97

In the third sentence, add “or modify” after “...but shall delete”.

Response: Accepted.

6. (T) 36.3.3.1 Required signals, COM_DET

Source/Date: PCS/PMA/LC Working Meeting, Amdahl, 3/3/97

Rewrite the definition of COM_DET as follows:

An indication that the code_group associated with the current PMA_RX_CLK<1> contains a valid comma. The PMA is required to detect and code_group-align to the comma+. Optionally, the PMA may also detect and code_group-align to the comma-. The PMA shall provide this signal as an output, but it is may not be used by the PCS.

Response: Accepted.

7. (E) 36.3.3.1 Required signals, EN_CDET

Source/Date: PCS/PMA/LC Working Meeting, Amdahl, 3/3/97

Change “This signal is optionally used by the PCS.” to “This signal is optionally generated by the PMA client.”

Response: Accepted.

8. (I) MAC and SERDES timing budget (35.3 or 36.3)

Source/Date: Haluk Aytac, H-P, 3/4/97

D1 comment deferred to D2: Include budget equations based on 1/97 San Diego proposal on this subject by Haluk Aytac into the relevant subclause.

Response: Accepted. Added to 36.3.7.

9. (T) 36.2.4.18 GMII clocking

Source/Date: GEA Technical meeting, PCS review 3/6/97

Since all existing PMA components use two receive clocks, the GMII interface should support two receive clocks.

Response: Accepted. GMII RX_ CLK<0:1> changed to RX_ CLK<0:1>

10. (T) Table 36-8, Receive Bus AC Specification

Source/Date: John Wolcott, Intel, 3/7/97

Currently the duty cycle specification here says 40/60% which on the RX_CLK[1:0] signals translates to pulse width times of 6.4ns and 9.6ns respectively. The tA-B specification gives a skew spec which is referenced between rising edges of the two clocks.

It seems that if you have a 40/60 DC on one clock and a 60/40 DC on the other, the only way to meet the skew spec is to intentionally delay(skew) one of them between 1.1ns and 2.1ns.

While there is no explicit requirement that the RX_CLK<1> and RX_CLK<0> signals be electrical complements of one another, this is probably going to be the case and the scenario above may occur.

Do the SERDES vendors agree with this spec (i.e. can they meet it)? What are the issues with spec'ing the skew between complementary edges?

Response: Open.

11. (T) Table 36-7, Transmit AC Specification

Source/Date: Stan Moriya, Synergy Semiconductor, 3/7/97

In IEEE Draft P802.3z/D2, page 36.40, Table 36-7: The tPERIOD in the first row of the table GTX_CLK is listed as 800ps. Shouldn't it be 8ns?

Response: Accepted.

802.3zD2 LC Comments and Response

1. (T) Break Link (Link Configuration Restart) Options

Source/Date: GEA Technical meeting, PCS review 3/6/97

a) Config_Reg = 0's
b) Config_Reg dedicated base page bit
c) Shut down transmitter (i.e., no light)
d) New ordered set

New ordered set affects silicon in progress - Shutting down the transmitter is not applicable to 1000BASE-CX, However, TX can be disabled by biasing T-, T+ - Config_Reg options are the most flexible and simplest to implement. - Config_Reg dedicated base page bit is the most desirable if it is available.

Response: Accepted. Specify Restart using dedicated bit. Post-meeting discussion suggested using D0 for Restart and specifying the protocol using D0 and D15 (Next Page) as follows:

```
D0 ................. D15

0  .................  0     Restart
1  .................  0     Null
X  .................  1     Next Page
```

2. (T) Figure 37-42, Arbitration Diagram

Source/Date: GEA Technical meeting, PCS review 3/6/97

Sufficient /C/s should be transmitted for each link protocol step to be equivalent to the D1 Link Monitor state machine config_time_short to insure interoperability. FDDI and Fast Ethernet implementations for the associated functions were not interoperable.

Response: Accepted. Qualify all state exits which effect a link protocol step with a timer/counter equivalent exceeding the link round trip time.

3. (T) 37.1.1 Scope

Source/Date: GEA Technical meeting, PCS review 3/6/97

Multiple PMA's are specified in clause 37 and only a single PMA is specified for 1000BASE-X. This concept is a carryover from clause 28 and not applicable to 1000BASE-X which specifies a single PMA. Link_Configuration is not applicable to 100BASE-X.

Response: Accepted. Delete support for Multiple PMA's from clause 37.

4. (E) 37 Physical Layer link signaling for 1000 Mb/s AutoFiber-Negotiation on fiber

Source/Date: GEA Technical meeting, PCS review 3/6/97

Link Configuration substantially different from Auto Negotiation. The mapping of terms between Clauses 36 (Link Configuration) and 37 (AFN) is confusing.

Response: Revert to D1 title for Clause 37: Link_Configuration function, type 1000BASE-X

5. (T) Figure 37-42, Arbitration Diagram

Source/Date: GEA Technical meeting, PCS review 3/6/97

A Config_Reg base page change should result in reconfiguration. This may already be covered in the ARB state machine.

Response: Open. Steve Haddock to investigate further

6. (T) Figure 37-42, Arbitration Diagram

Source/Date: GEA Technical meeting, PCS review 3/6/97

ARB state machine: is “transmit_ack <= false” missing in state NEXT_PAGE_WAIT? It appears to be missing, but transmit_ack is defined to have a default value of zero. This appears to violate state machine conventions (clause 21).

Response: Open.

7. (T) Figure 37-42, Arbitration Diagram

Source/Date: GEA Technical meeting, PCS review 3/6/97.

The definition of “ability_match” seems weak in that it does not specify which three Link Code Words are matched. This may already be covered by LC item 5 above.

Response: Open.

8. (T) 37.2.5 Management function requirements

Source/Date: GEA Technical meeting, PCS review 3/6/97.

Management function requirements in 37.2.5 are inconsistent with those in 35.2.5.

Response: Accepted.

• Use 35.2.5.1 for Control Register 0

• Use 35.2.5.2 for Status Register 0

• Use 37.2.5.X definitions for all other registers

• Config_Reg bits such as Pause, Remote Fault, etc. are not assigned to registers in clause 37. They are in clause 35. Use the clause 35 bit assignments in 37.2.5.x register definitions.

9. (T) Figure 37-42, Arbitration Diagram

Source/Date: Devendra Tripathi, XaQti, 3/7/97

In state ABILITY_DETECT and NEXT_PAGE_WAIT set transmit_ack = false.

Response: Open.

10. (T) Figure 37-42, Arbitration Diagram

Source/Date: Devendra Tripathi, XaQti, 3/7/97

In the condition of branch ACKNOWLEDGE_DETECT -> TRANSMIT_DISABLE add "+ rx_restart”.

Response: Open.

11. (T) Figure 37-42, Arbitration Diagram

Source/Date: Devendra Tripathi, XaQti, 3/7/97

In the state TRANSMIT_DISABLE add out put tx_restart = true.

Response: Open.

12. (T) Figure 37-42, Arbitration Diagram

Source/Date: Devendra Tripathi, XaQti, 3/7/97

In the condition NEXT_PAGE_WAIT -> TRANSMIT_DISABLE add “+rx_restart”.

Response: Open.

13. (T) Figure 37-42, Arbitration Diagram

Source/Date: Devendra Tripathi, XaQti, 3/7/97

In the condition IDLE_DETECT -> TRANSMIT_DISABLE add “+rx_restart”.

Response: Open.

14. (T) Figure 37-42, Arbitration Diagram

Source/Date: Devendra Tripathi, XaQti, 3/7/97

In the condition LINK_OK -> TRANSMIT_DISABLE add “+rx_restart”.

Response: Open.