36. Physical Coding Sublayer (PCS) and Physical Medium Attachment (PMA) sublayer, type 1000BASE-X

36.1 Overview

36.1.1 Scope

This clause specifies the Physical Coding Sublayer (PCS) and the Physical Medium Attachment (PMA) sublayer that are common to a family of 1000 Mb/s Physical Layer implementations, collectively known as 1000BASE-X.

1000BASE-X is based on the Physical Layer standards developed by ANSI INCITS 230-1994 (Fibre Channel Physical and Signaling Interface). In particular, this standard uses the same 8B/10B coding as Fibre Channel, a PMA sublayer compatible with speed-enhanced versions of the ANSI 10-bit serializer chip, and similar optical and electrical specifications.

1000BASE-X PCS and PMA sublayers map the interface characteristics of the PMD sublayer (including MDI) to the services expected by the Reconciliation sublayer. 1000BASE-X can be extended to support any other full duplex medium requiring only that the medium be compliant at the PMD level.

36.1.2 Objectives

The following are the objectives of 1000BASE-X:

a) To support the CSMA/CD MAC
b) To support the 1000 Mb/s repeater
c) To provide for Auto-Negotiation among like 1000 Mb/s PMDs
d) To provide 1000 Mb/s data rate at the GMII
e) To support cable plants using 150 Ω balanced copper cabling, or cabled optical fiber compliant with ISO/IEC 11801:1995
f) To allow for a nominal network extent of up to 5 km, including
    1) 150 Ω balanced links of 25 m span
    2) one-repeater networks of 50 m span (using all 150 Ω balanced copper cabling)
    3) one-repeater networks of 200 m span (using fiber)
    4) DTE/DTE links of 5000 m (using fiber)
g) To preserve full duplex behavior of underlying PMD channels
h) To support a BER objective of 10^{-12}

NOTE—The 1000BASE-X PCS and PMA do not constrain the extent of a full duplex network. PMDs in Clause 59 and Clause 60 have ranges beyond 5 km.

36.1.3 Relationship of 1000BASE-X to other standards

Figure 36–1 depicts the relationships among the 1000BASE-X sublayers (shown shaded), the CSMA/CD MAC and reconciliation layers, and the ISO/IEC 8802-2 LLC.

36.1.4 Summary of 1000BASE-X sublayers

The following provides an overview of the 1000BASE-X sublayers.[2]

36.1.4.1 Physical Coding Sublayer (PCS)

The PCS interface is the Gigabit Media Independent Interface (GMII) that provides a uniform interface to the Reconciliation sublayer for all 1000 Mb/s PHY implementations (e.g., not only 1000BASE-X but also other possible types of gigabit PHY entities). 1000BASE-X provides services to the GMII in a manner analogous to how 100BASE-X provides services to the 100 Mb/s MII.

The 1000BASE-X PCS provides all services required by the GMII, including

a) Encoding (decoding) of GMII data octets to (from) ten-bit code-groups (8B/10B) for communication with the underlying PMA
b) Generating Carrier Sense and Collision Detect indications for use by PHY's half duplex clients
c) Managing the Auto-Negotiation process, and informing the management entity via the GMII when the PHY is ready for use

36.1.4.2 Physical Medium Attachment (PMA) sublayer

The PMA provides a medium-independent means for the PCS to support the use of a range of serial-bit-oriented physical media. The 1000BASE-X PMA performs the following functions:

a) Mapping of transmit and receive code-groups between the PCS and PMA via the PMA Service Interface
b) Serialization (deserialization) of code-groups for transmission (reception) on the underlying serial PMD
c) Recovery of clock from the 8B/10B-coded data supplied by the PMD
d) Mapping of transmit and receive bits between the PMA and PMD via the PMD Service Interface
e) Data loopback at the PMD Service Interface

36.1.4.3 Physical Medium Dependent (PMD) sublayer

1000BASE-X Physical Layer signaling for fiber and copper media is adapted from ANSI INCITS 230-1994 (FC-PH), Clauses 6 and 7 respectively. These clauses define 1062.5 Mb/s, full duplex signaling systems that accommodate single-mode optical fiber, multimode optical fiber, and 150 Ω balanced copper cabling. 1000BASE-X adapts these basic Physical Layer specifications for use with the PMD sublayer and media specified in Clause 38 and Clause 39.

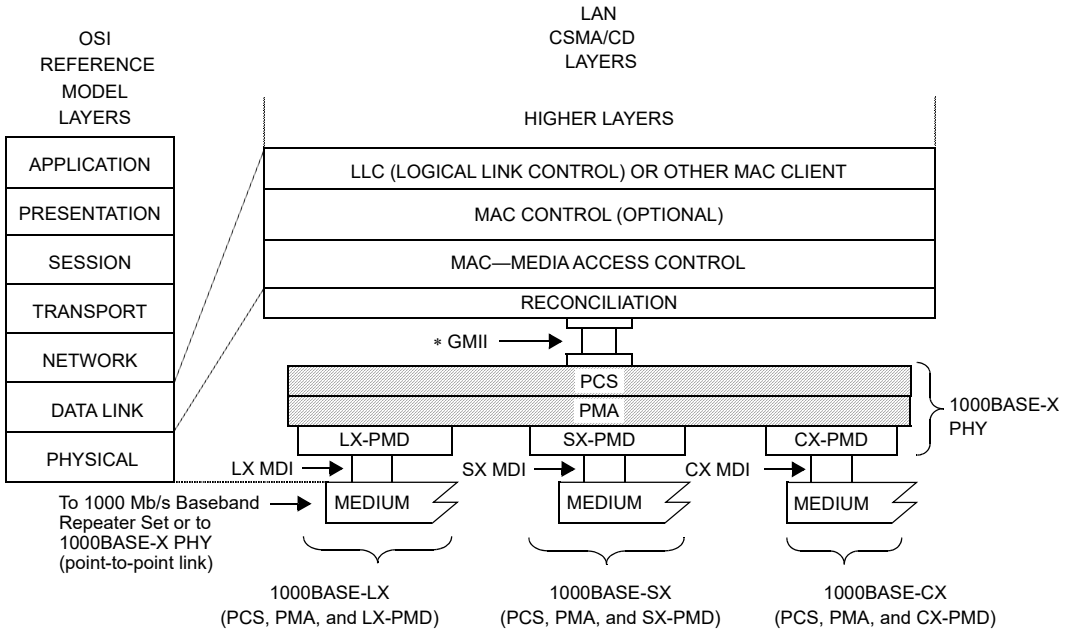
The MDI, logically subsumed within each PMD subclause, is the actual medium attachment, including connectors, for the various supported media.

Figure 36–1 depicts the relationship between 1000BASE-X and its associated PMD sublayers.

36.1.5 Inter-sublayer interfaces

There are a number of interfaces employed by 1000BASE-X. Some (such as the PMA Service Interface) use an abstract service model to define the operation of the interface. An optional physical instantiation of the PCS Interface has been defined. It is called the GMII (Gigabit Media Independent Interface). An optional

[2]The 1000BASE-X PHY consists of that portion of the Physical Layer between the MDI and GMII consisting of the PCS, PMA, and PMD sublayers. The 1000BASE-X PHY is roughly analogous to the 100BASE-X PHY.



MDI=MEDIUM DEPENDENT INTERFACE
GMII=GIGABIT MEDIA INDEPENDENT INTERFACE
PCS=PHYSICAL CODING SUBLAYER
PMA=PHYSICAL MEDIUM ATTACHMENT
PHY=PHYSICAL LAYER DEVICE
PMD=PHYSICAL MEDIUM DEPENDENT
LX-PMD=PMD FOR FIBER—LONG WAVELENGTH, Clause 38
SX-PMD=PMD FOR FIBER—SHORT WAVELENGTH, Clause 38
CX-PMD=PMD FOR 150 Ω BALANCED COPPER CABLING, Clause 39

NOTE—The PMD sublayers are mutually independent.

∗ GMII is optional.

Figure 36–1—Relationship of 1000BASE-X and the PMDs

physical instantiation of the PMA Service Interface has also been defined (see 36.3.3). It is adapted from ANSI Technical Report TR/X3.18-1997 (Fibre Channel—10-bit Interface). Figure 36–2 depicts the relationship and mapping of the services provided by all of the interfaces relevant to 1000BASE-X.

It is important to note that, while this specification defines interfaces in terms of bits, octets, and code-groups, implementers may choose other data path widths for implementation convenience. The only exceptions are a) the GMII, which, when implemented at an observable interconnection port, uses an octet-wide data path as specified in Clause 35, b) the PMA Service Interface, which, when physically implemented as the TBI (Ten-Bit Interface) at an observable interconnection port, uses a 10-bit wide data path as specified in 36.3.3, and c) the MDI, which uses a serial, physical interface.

36.1.6 Functional block diagram

Figure 36–2 provides a functional block diagram of the 1000BASE-X PHY.

36.1.7 State diagram conventions

The body of this standard is comprised of state diagrams, including the associated definitions of variables, constants, and functions. Should there be a discrepancy between a state diagram and descriptive text, the state diagram prevails.

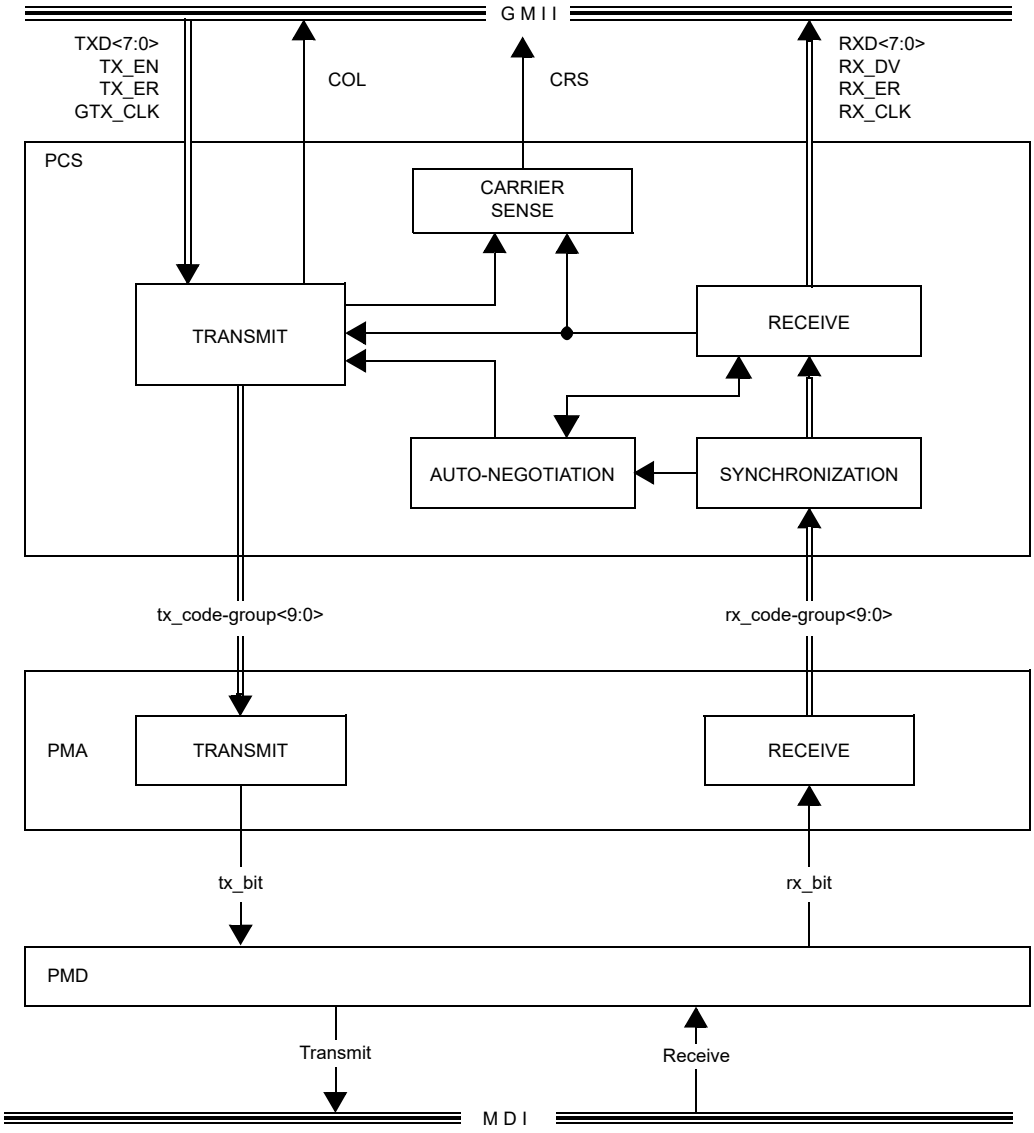


Figure 36–2—Functional block diagram

The notation used in the state diagrams follows the conventions of 21.5. State diagram timers follow the conventions of 14.2.3.2.

36.2 Physical Coding Sublayer (PCS)

36.2.1 PCS Interface (GMII)

The PCS Service Interface allows the 1000BASE-X PCS to transfer information to and from a PCS client. PCS clients include the MAC (via the Reconciliation sublayer) and repeater. The PCS Interface is precisely defined as the Gigabit Media Independent Interface (GMII) in Clause 35.

In this clause, the setting of GMII variables to TRUE or FALSE is equivalent, respectively, to “asserting” or “deasserting” them as specified in Clause 35.

36.2.2 Functions within the PCS

The PCS comprises the PCS Transmit, Carrier Sense, Synchronization, PCS Receive, and Auto-Negotiation processes for 1000BASE-X. The PCS shields the Reconciliation sublayer (and MAC) from the specific nature of the underlying channel. When communicating with the GMII, the PCS uses an octet-wide, synchronous data path, with packet delimiting being provided by separate transmit control signals (TX_EN and TX_ER) and receive control signals (RX_DV and RX_ER). When communicating with the PMA, the PCS uses a ten-bit wide, synchronous data path, which conveys ten-bit code-groups. At the PMA Service Interface, code-group alignment and MAC packet delimiting are made possible by embedding special non-data code-groups in the transmitted code-group stream. The PCS provides the functions necessary to map packets between the GMII format and the PMA Service Interface format.

The PCS Transmit process continuously generates code-groups based upon the TXD <7:0>, TX_EN, and TX_ER signals on the GMII, sending them immediately to the PMA Service Interface via the PMA_UNITDATA.request primitive. The PCS Transmit process generates the GMII signal COL based on whether a reception is occurring simultaneously with transmission. Additionally, it generates the internal flag, transmitting, for use by the Carrier Sense process. The PCS Transmit process monitors the Auto-Negotiation process xmit flag to determine whether to transmit data or reconfigure the link.

The Carrier Sense process controls the GMII signal CRS (see Figure 36–8).

The PCS Synchronization process continuously accepts code-groups via the PMA_UNITDATA.indication primitive and conveys received code-groups to the PCS Receive process via the SYNC_UNITDATA.indicate primitive. The PCS Synchronization process sets the sync_status flag to indicate whether the PMA is functioning dependably (as well as can be determined without exhaustive error-ratio analysis).

The PCS Receive process continuously accepts code-groups via the SYNC_UNITDATA.indicate primitive. The PCS Receive process monitors these code-groups and generates RXD <7:0>, RX_DV, and RX_ER on the GMII, and the internal flag, receiving, used by the Carrier Sense and Transmit processes.

The PCS Auto-Negotiation process sets the xmit flag to inform the PCS Transmit process to either transmit normal idles interspersed with packets as requested by the GMII or to reconfigure the link. The PCS Auto-Negotiation process is specified in Clause 37.

36.2.3 Use of code-groups

The PCS maps GMII signals into ten-bit code groups, and vice versa, using an 8B/10B block coding scheme. Implicit in the definition of a code-group is an establishment of code-group boundaries by a PMA code-group alignment function as specified in 36.3.2.4. Code-groups are unobservable and have no meaning outside the PCS. The PCS functions ENCODE and DECODE generate, manipulate, and interpret code-groups as provided by the rules in 36.2.4.

36.2.4 8B/10B transmission code

The PCS uses a transmission code to improve the transmission characteristics of information to be transferred across the link. The encodings defined by the transmission code ensure that sufficient transitions are present in the PHY bit stream to make clock recovery possible at the receiver. Such encoding also greatly increases the likelihood of detecting any single or multiple bit errors that may occur during transmission and reception of information. In addition, some of the special code-groups of the transmission code contain a distinct and easily recognizable bit pattern that assists a receiver in achieving code-group alignment on the

incoming PHY bit stream. The 8B/10B transmission code specified for use in this standard has a high transition density, is a run-length-limited code, and is DC-balanced. The transition density of the 8B/10B symbols ranges from 3 to 8 transitions per symbol.

The definition of the 8B/10B transmission code in this standard is identical to that specified in ANSI INCITS 230-1994 (FC-PH), Clause 11. The relationship of code-group bit positions to PMA and other PCS constructs is illustrated in Figure 36–3.

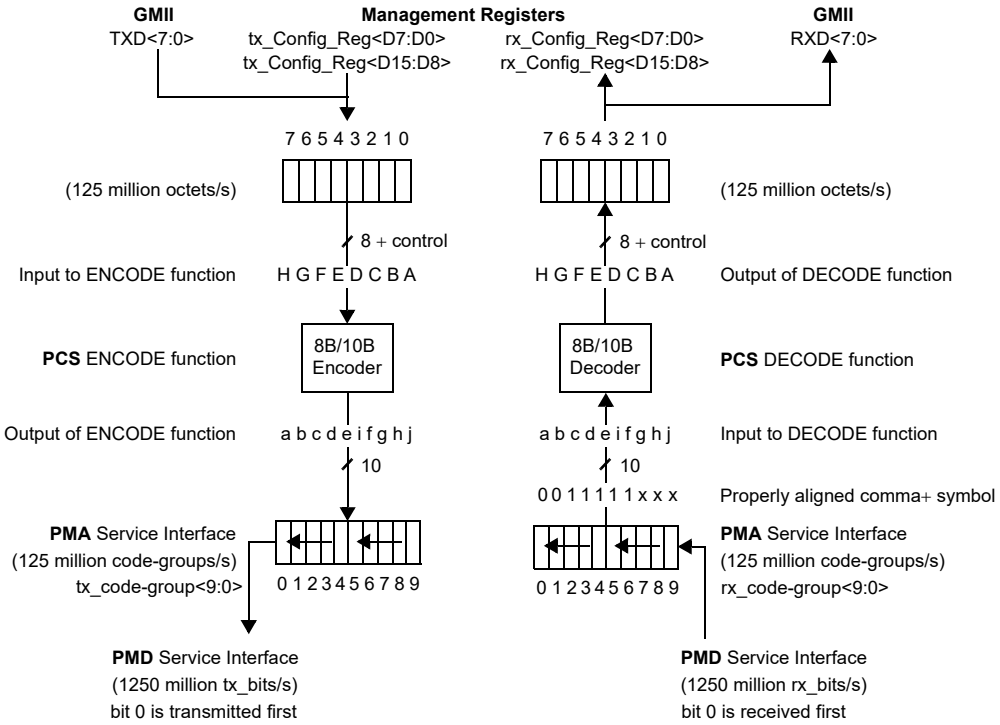


Figure 36–3—PCS reference diagram

36.2.4.1 Notation conventions

8B/10B transmission code uses letter notation for describing the bits of an unencoded information octet and a single control variable. Each bit of the unencoded information octet contains either a binary zero or a binary one. A control variable, Z, has either the value D or the value K. When the control variable associated with an unencoded information octet contains the value D, the associated encoded code-group is referred to as a data code-group. When the control variable associated with an unencoded information octet contains the value K, the associated encoded code-group is referred to as a special code-group.

The bit notation of A,B,C,D,E,F,G,H for an unencoded information octet is used in the description of the 8B/10B transmission code. The bits A,B,C,D,E,F,G,H are translated to bits a,b,c,d,e,i,f,g,h,j of 10-bit transmission code-groups. 8B/10B code-group bit assignments are illustrated in Figure 36–3. Each valid code-group has been given a name using the following convention: /Dx.y/ for the 256 valid data code-groups, and /Kx.y/ for special control code-groups, where x is the decimal value of bits EDCBA, and y is the decimal value of bits HGF.

36.2.4.2 Transmission order

Code-group bit transmission order is illustrated in Figure 36–3.

Code-groups within multi-code-group ordered sets (as specified in Table 36–3) are transmitted sequentially beginning with the special code-group used to distinguish the ordered set (e.g., /K28.5/) and proceeding code-group by code-group from left to right within the definition of the ordered set until all code-groups of the ordered set are transmitted.

The first code-group of every multi-code-group ordered set is transmitted in an even-numbered code-group position counting from the first code-group after a reset or power-on. Subsequent code-groups continuously alternate as odd and even-numbered code-groups.

The contents of a packet are transmitted sequentially beginning with the ordered set used to denote the Start_of_Packet (the SPD delimiter) and proceeding code-group by code-group from left to right within the definition of the packet until the ordered set used to denote the End_of_Packet (the EPD delimiter) is transmitted.

36.2.4.3 Valid and invalid code-groups

Table 36–1a–e defines the valid data code-groups (D code-groups) of the 8B/10B transmission code. Table 36–2 defines the valid special code-groups (K code-groups) of the code. The tables are used for both generating valid code-groups (encoding) and checking the validity of received code-groups (decoding). In the tables, each octet entry has two columns that represent two (not necessarily different) code-groups. The two columns correspond to the valid code-group based on the current value of the running disparity (Current RD – or Current RD +). Running disparity is a binary parameter with either the value negative (–) or the value positive (+). Annex 36B provides several 8B/10B transmission code running disparity calculation examples.

36.2.4.4 Running disparity rules

After powering on or exiting a test mode, the transmitter shall assume the negative value for its initial running disparity. Upon transmission of any code-group, the transmitter shall calculate a new value for its running disparity based on the contents of the transmitted code-group.

After powering on or exiting a test mode, the receiver should assume either the positive or negative value for its initial running disparity. Upon the reception of any code-group, the receiver determines whether the code-group is valid or invalid and calculates a new value for its running disparity based on the contents of the received code-group.

The following rules for running disparity shall be used to calculate the new running disparity value for code-groups that have been transmitted (transmitter's running disparity) and that have been received (receiver’s running disparity).

Running disparity for a code-group is calculated on the basis of sub-blocks, where the first six bits (abcdei) form one sub-block (six-bit sub-block) and the second four bits (fghj) form the other sub-block (four-bit sub-block). Running disparity at the beginning of the six-bit sub-block is the running disparity at the end of the last code-group. Running disparity at the beginning of the four-bit sub-block is the running disparity at the end of the six-bit sub-block. Running disparity at the end of the code-group is the running disparity at the end of the four-bit sub-block.

Running disparity for the sub-blocks is calculated as follows:

a) Running disparity at the end of any sub-block is positive if the sub-block contains more ones than zeros. It is also positive at the end of the six-bit sub-block if the six-bit sub-block is 000111, and it is positive at the end of the four-bit sub-block if the four-bit sub-block is 0011.

b) Running disparity at the end of any sub-block is negative if the sub-block contains more zeros than ones. It is also negative at the end of the six-bit sub-block if the six-bit sub-block is 111000, and it is negative at the end of the four-bit sub-block if the four-bit sub-block is 1100.

c) Otherwise, running disparity at the end of the sub-block is the same as at the beginning of the sub-block.

NOTE—All sub-blocks with equal numbers of zeros and ones are disparity neutral. In order to limit the run length of 0's or 1's between sub-blocks, the 8B/10B transmission code rules specify that sub-blocks encoded as 000111 or 0011 are generated only when the running disparity at the beginning of the sub-block is positive; thus, running disparity at the end of these sub-blocks is also positive. Likewise, sub-blocks containing 111000 or 1100 are generated only when the running disparity at the beginning of the sub-block is negative; thus, running disparity at the end of these sub-blocks is also negative.

36.2.4.5 Generating code-groups

The appropriate entry in either Table 36–1a–e or Table 36–2 is found for each octet for which a code-group is to be generated (encoded). The current value of the transmitter's running disparity shall be used to select the code-group from its corresponding column. For each code-group transmitted, a new value of the running disparity is calculated. This new value is used as the transmitter's current running disparity for the next octet to be encoded and transmitted.

36.2.4.6 Checking the validity of received code-groups

The following rules shall be used to determine the validity of received code groups:

a) The column in Table 36–1a–e and Table 36–2 corresponding to the current value of the receiver's running disparity is searched for the received code-group.

b) If the received code-group is found in the proper column, according to the current running disparity, then the code-group is considered valid and, for data code-groups, the associated data octet determined (decoded).

c) If the received code-group is not found in that column, then the code-group is considered invalid.

d) Independent of the code-group's validity, the received code-group is used to calculate a new value of running disparity. The new value is used as the receiver's current running disparity for the next received code-group.

Detection of an invalid code-group does not necessarily indicate that the code-group in which the invalid code-group was detected is in error. Invalid code-groups may result from a prior error which altered the running disparity of the PHY bit stream but which did not result in a detectable error at the code-group in which the error occurred.

The number of invalid code-groups detected is proportional to the bit error ratio (BER) of the link. Link error monitoring may be performed by counting invalid code-groups.

36.2.4.7 Ordered sets

Eight ordered sets, consisting of a single special code-group or combinations of special and data code-groups are specifically defined. Ordered sets which include /K28.5/ provide the ability to obtain bit and code-group synchronization and establish ordered set alignment (see 36.2.4.9 and 36.3.2.4). Ordered sets provide for the delineation of a packet and synchronization between the transmitter and receiver circuits at opposite ends of a link. Table 36–3 lists the defined ordered sets. Certain PHYs include an option (see 78.3) to transmit or receive /LI/, /LI1/ and /LI2/ to support Energy-Efficient Ethernet (see Clause 78).

Table 36–1a—Valid data code-groups

Code Group Name	Octet Value	Octet Bits HGF EDCBA	Current RD –	Current RD +
			abcdei fghj	abcdei fghj
D0.0	00	000 00000	100111 0100	011000 1011
D1.0	01	000 00001	011101 0100	100010 1011
D2.0	02	000 00010	101101 0100	010010 1011
D3.0	03	000 00011	110001 1011	110001 0100
D4.0	04	000 00100	110101 0100	001010 1011
D5.0	05	000 00101	101001 1011	101001 0100
D6.0	06	000 00110	011001 1011	011001 0100
D7.0	07	000 00111	111000 1011	000111 0100
D8.0	08	000 01000	111001 0100	000110 1011
D9.0	09	000 01001	100101 1011	100101 0100
D10.0	0A	000 01010	010101 1011	010101 0100
D11.0	0B	000 01011	110100 1011	110100 0100
D12.0	0C	000 01100	001101 1011	001101 0100
D13.0	0D	000 01101	101100 1011	101100 0100
D14.0	0E	000 01110	011100 1011	011100 0100
D15.0	0F	000 01111	010111 0100	101000 1011
D16.0	10	000 10000	011011 0100	100100 1011
D17.0	11	000 10001	100011 1011	100011 0100
D18.0	12	000 10010	010011 1011	010011 0100
D19.0	13	000 10011	110010 1011	110010 0100
D20.0	14	000 10100	001011 1011	001011 0100
D21.0	15	000 10101	101010 1011	101010 0100
D22.0	16	000 10110	011010 1011	011010 0100
D23.0	17	000 10111	111010 0100	000101 1011
D24.0	18	000 11000	110011 0100	001100 1011
D25.0	19	000 11001	100110 1011	100110 0100
D26.0	1A	000 11010	010110 1011	010110 0100
D27.0	1B	000 11011	110110 0100	001001 1011
D28.0	1C	000 11100	001110 1011	001110 0100
D29.0	1D	000 11101	101110 0100	010001 1011
D30.0	1E	000 11110	011110 0100	100001 1011
D31.0	1F	000 11111	101011 0100	010100 1011
D0.1	20	001 00000	100111 1001	011000 1001
D1.1	21	001 00001	011101 1001	100010 1001
D2.1	22	001 00010	101101 1001	010010 1001
D3.1	23	001 00011	110001 1001	110001 1001
D4.1	24	001 00100	110101 1001	001010 1001
D5.1	25	001 00101	101001 1001	101001 1001
D6.1	26	001 00110	011001 1001	011001 1001
D7.1	27	001 00111	111000 1001	000111 1001
D8.1	28	001 01000	111001 1001	000110 1001
D9.1	29	001 01001	100101 1001	100101 1001
D10.1	2A	001 01010	010101 1001	010101 1001
D11.1	2B	001 01011	110100 1001	110100 1001
D12.1	2C	001 01100	001101 1001	001101 1001
D13.1	2D	001 01101	101100 1001	101100 1001
D14.1	2E	001 01110	011100 1001	011100 1001
D15.1	2F	001 01111	010111 1001	101000 1001
D16.1	30	001 10000	011011 1001	100100 1001
D17.1	31	001 10001	100011 1001	100011 1001
D18.1	32	001 10010	010011 1001	010011 1001
D19.1	33	001 10011	110010 1001	110010 1001
D20.1	34	001 10100	001011 1001	001011 1001
D21.1	35	001 10101	101010 1001	101010 1001
D22.1	36	001 10110	011010 1001	011010 1001
D23.1	37	001 10111	111010 1001	000101 1001
D24.1	38	001 11000	110011 1001	001100 1001
D25.1	39	001 11001	100110 1001	100110 1001
D26.1	3A	001 11010	010110 1001	010110 1001
D27.1	3B	001 11011	110110 1001	001001 1001

(continued)

Table 36–1b—Valid data code-groups

Code Group Name	Octet Value	Octet Bits HGF EDCBA	Current RD –	Current RD +
			abcdei fghj	abcdei fghj
D28.1	3C	001 11100	001110 1001	001110 1001
D29.1	3D	001 11101	101110 1001	010001 1001
D30.1	3E	001 11110	011110 1001	100001 1001
D31.1	3F	001 11111	101011 1001	010100 1001
D0.2	40	010 00000	100111 0101	011000 0101
D1.2	41	010 00001	011101 0101	100010 0101
D2.2	42	010 00010	101101 0101	010010 0101
D3.2	43	010 00011	110001 0101	110001 0101
D4.2	44	010 00100	110101 0101	001010 0101
D5.2	45	010 00101	101001 0101	101001 0101
D6.2	46	010 00110	011001 0101	011001 0101
D7.2	47	010 00111	111000 0101	000111 0101
D8.2	48	010 01000	111001 0101	000110 0101
D9.2	49	010 01001	100101 0101	100101 0101
D10.2	4A	010 01010	010101 0101	010101 0101
D11.2	4B	010 01011	110100 0101	110100 0101
D12.2	4C	010 01100	001101 0101	001101 0101
D13.2	4D	010 01101	101100 0101	101100 0101
D14.2	4E	010 01110	011100 0101	011100 0101
D15.2	4F	010 01111	010111 0101	101000 0101
D16.2	50	010 10000	011011 0101	100100 0101
D17.2	51	010 10001	100011 0101	100011 0101
D18.2	52	010 10010	010011 0101	010011 0101
D19.2	53	010 10011	110010 0101	110010 0101
D20.2	54	010 10100	001011 0101	001011 0101
D21.2	55	010 10101	101010 0101	101010 0101
D22.2	56	010 10110	011010 0101	011010 0101
D23.2	57	010 10111	111010 0101	000101 0101
D24.2	58	010 11000	110011 0101	001100 0101
D25.2	59	010 11001	100110 0101	100110 0101
D26.2	5A	010 11010	010110 0101	010110 0101
D27.2	5B	010 11011	110110 0101	001001 0101
D28.2	5C	010 11100	001110 0101	001110 0101
D29.2	5D	010 11101	101110 0101	010001 0101
D30.2	5E	010 11110	011110 0101	100001 0101
D31.2	5F	010 11111	101011 0101	010100 0101
D0.3	60	011 00000	100111 0011	011000 1100
D1.3	61	011 00001	011101 0011	100010 1100
D2.3	62	011 00010	101101 0011	010010 1100
D3.3	63	011 00011	110001 1100	110001 0011
D4.3	64	011 00100	110101 0011	001010 1100
D5.3	65	011 00101	101001 1100	101001 0011
D6.3	66	011 00110	011001 1100	011001 0011
D7.3	67	011 00111	111000 1100	000111 0011
D8.3	68	011 01000	111001 0011	000110 1100
D9.3	69	011 01001	100101 1100	100101 0011
D10.3	6A	011 01010	010101 1100	010101 0011
D11.3	6B	011 01011	110100 1100	110100 0011
D12.3	6C	011 01100	001101 1100	001101 0011
D13.3	6D	011 01101	101100 1100	101100 0011
D14.3	6E	011 01110	011100 1100	011100 0011
D15.3	6F	011 01111	010111 0011	101000 1100
D16.3	70	011 10000	011011 0011	100100 1100
D17.3	71	011 10001	100011 1100	100011 0011
D18.3	72	011 10010	010011 1100	010011 0011
D19.3	73	011 10011	110010 1100	110010 0011
D20.3	74	011 10100	001011 1100	001011 0011
D21.3	75	011 10101	101010 1100	101010 0011
D22.3	76	011 10110	011010 1100	011010 0011
D23.3	77	011 10111	111010 0011	000101 1100

(continued)

Table 36–1c—Valid data code-groups

Code Group Name	Octet Value	Octet Bits HGF EDCBA	Current RD –	Current RD +
			abcdei fghj	abcdei fghj
D24.3	78	011 11000	110011 0011	001100 1100
D25.3	79	011 11001	100110 1100	100110 0011
D26.3	7A	011 11010	010110 1100	010110 0011
D27.3	7B	011 11011	110110 0011	001001 1100
D28.3	7C	011 11100	001110 1100	001110 0011
D29.3	7D	011 11101	101110 0011	010001 1100
D30.3	7E	011 11110	011110 0011	100001 1100
D31.3	7F	011 11111	101011 0011	010100 1100
D0.4	80	100 00000	100111 0010	011000 1101
D1.4	81	100 00001	011101 0010	100010 1101
D2.4	82	100 00010	101101 0010	010010 1101
D3.4	83	100 00011	110001 1101	110001 0010
D4.4	84	100 00100	110101 0010	001010 1101
D5.4	85	100 00101	101001 1101	101001 0010
D6.4	86	100 00110	011001 1101	011001 0010
D7.4	87	100 00111	111000 1101	000111 0010
D8.4	88	100 01000	111001 0010	000110 1101
D9.4	89	100 01001	100101 1101	100101 0010
D10.4	8A	100 01010	010101 1101	010101 0010
D11.4	8B	100 01011	110100 1101	110100 0010
D12.4	8C	100 01100	001101 1101	001101 0010
D13.4	8D	100 01101	101100 1101	101100 0010
D14.4	8E	100 01110	011100 1101	011100 0010
D15.4	8F	100 01111	010111 0010	101000 1101
D16.4	90	100 10000	011011 0010	100100 1101
D17.4	91	100 10001	100011 1101	100011 0010
D18.4	92	100 10010	010011 1101	010011 0010
D19.4	93	100 10011	110010 1101	110010 0010
D20.4	94	100 10100	001011 1101	001011 0010
D21.4	95	100 10101	101010 1101	101010 0010
D22.4	96	100 10110	011010 1101	011010 0010
D23.4	97	100 10111	111010 0010	000101 1101
D24.4	98	100 11000	110011 0010	001100 1101
D25.4	99	100 11001	100110 1101	100110 0010
D26.4	9A	100 11010	010110 1101	010110 0010
D27.4	9B	100 11011	110110 0010	001001 1101
D28.4	9C	100 11100	001110 1101	001110 0010
D29.4	9D	100 11101	101110 0010	010001 1101
D30.4	9E	100 11110	011110 0010	100001 1101
D31.4	9F	100 11111	101011 0010	010100 1101
D0.5	A0	101 00000	100111 1010	011000 1010
D1.5	A1	101 00001	011101 1010	100010 1010
D2.5	A2	101 00010	101101 1010	010010 1010
D3.5	A3	101 00011	110001 1010	110001 1010
D4.5	A4	101 00100	110101 1010	001010 1010
D5.5	A5	101 00101	101001 1010	101001 1010
D6.5	A6	101 00110	011001 1010	011001 1010
D7.5	A7	101 00111	111000 1010	000111 1010
D8.5	A8	101 01000	111001 1010	000110 1010
D9.5	A9	101 01001	100101 1010	100101 1010
D10.5	AA	101 01010	010101 1010	010101 1010
D11.5	AB	101 01011	110100 1010	110100 1010
D12.5	AC	101 01100	001101 1010	001101 1010
D13.5	AD	101 01101	101100 1010	101100 1010
D14.5	AE	101 01110	011100 1010	011100 1010
D15.5	AF	101 01111	010111 1010	101000 1010
D16.5	B0	101 10000	011011 1010	100100 1010
D17.5	B1	101 10001	100011 1010	100011 1010
D18.5	B2	101 10010	010011 1010	010011 1010
D19.5	B3	101 10011	110010 1010	110010 1010

(continued)

Table 36–1d—Valid data code-groups

Code Group Name	Octet Value	Octet Bits HGF EDCBA	Current RD –	Current RD +
			abcdei fghj	abcdei fghj
D20.5	B4	101 10100	001011 1010	001011 1010
D21.5	B5	101 10101	101010 1010	101010 1010
D22.5	B6	101 10110	011010 1010	011010 1010
D23.5	B7	101 10111	111010 1010	000101 1010
D24.5	B8	101 11000	110011 1010	001100 1010
D25.5	B9	101 11001	100110 1010	100110 1010
D26.5	BA	101 11010	010110 1010	010110 1010
D27.5	BB	101 11011	110110 1010	001001 1010
D28.5	BC	101 11100	001110 1010	001110 1010
D29.5	BD	101 11101	101110 1010	010001 1010
D30.5	BE	101 11110	011110 1010	100001 1010
D31.5	BF	101 11111	101011 1010	010100 1010
D0.6	C0	110 00000	100111 0110	011000 0110
D1.6	C1	110 00001	011101 0110	100010 0110
D2.6	C2	110 00010	101101 0110	010010 0110
D3.6	C3	110 00011	110001 0110	110001 0110
D4.6	C4	110 00100	110101 0110	001010 0110
D5.6	C5	110 00101	101001 0110	101001 0110
D6.6	C6	110 00110	011001 0110	011001 0110
D7.6	C7	110 00111	111000 0110	000111 0110
D8.6	C8	110 01000	111001 0110	000110 0110
D9.6	C9	110 01001	100101 0110	100101 0110
D10.6	CA	110 01010	010101 0110	010101 0110
D11.6	CB	110 01011	110100 0110	110100 0110
D12.6	CC	110 01100	001101 0110	001101 0110
D13.6	CD	110 01101	101100 0110	101100 0110
D14.6	CE	110 01110	011100 0110	011100 0110
D15.6	CF	110 01111	010111 0110	101000 0110
D16.6	D0	110 10000	011011 0110	100100 0110
D17.6	D1	110 10001	100011 0110	100011 0110
D18.6	D2	110 10010	010011 0110	010011 0110
D19.6	D3	110 10011	110010 0110	110010 0110
D20.6	D4	110 10100	001011 0110	001011 0110
D21.6	D5	110 10101	101010 0110	101010 0110
D22.6	D6	110 10110	011010 0110	011010 0110
D23.6	D7	110 10111	111010 0110	000101 0110
D24.6	D8	110 11000	110011 0110	001100 0110
D25.6	D9	110 11001	100110 0110	100110 0110
D26.6	DA	110 11010	010110 0110	010110 0110
D27.6	DB	110 11011	110110 0110	001001 0110
D28.6	DC	110 11100	001110 0110	001110 0110
D29.6	DD	110 11101	101110 0110	010001 0110
D30.6	DE	110 11110	011110 0110	100001 0110
D31.6	DF	110 11111	101011 0110	010100 0110
D0.7	E0	111 00000	100111 0001	011000 1110
D1.7	E1	111 00001	011101 0001	100010 1110
D2.7	E2	111 00010	101101 0001	010010 1110
D3.7	E3	111 00011	110001 1110	110001 0001
D4.7	E4	111 00100	110101 0001	001010 1110
D5.7	E5	111 00101	101001 1110	101001 0001
D6.7	E6	111 00110	011001 1110	011001 0001
D7.7	E7	111 00111	111000 1110	000111 0001
D8.7	E8	111 01000	111001 0001	000110 1110
D9.7	E9	111 01001	100101 1110	100101 0001
D10.7	EA	111 01010	010101 1110	010101 0001
D11.7	EB	111 01011	110100 1110	110100 1000
D12.7	EC	111 01100	001101 1110	001101 0001
D13.7	ED	111 01101	101100 1110	101100 1000
D14.7	EE	111 01110	011100 1110	011100 1000
D15.7	EF	111 01111	010111 0001	101000 1110

(continued)

Table 36–1e—Valid data code-groups

Code Group Name	Octet Value	Octet Bits HGF EDCBA	Current RD –	Current RD +
			abcdei fghj	abcdei fghj
D16.7	F0	111 10000	011011 0001	100100 1110
D17.7	F1	111 10001	100011 0111	100011 0001
D18.7	F2	111 10010	010011 0111	010011 0001
D19.7	F3	111 10011	110010 1110	110010 0001
D20.7	F4	111 10100	001011 0111	001011 0001
D21.7	F5	111 10101	101010 1110	101010 0001
D22.7	F6	111 10110	011010 1110	011010 0001
D23.7	F7	111 10111	111010 0001	000101 1110
D24.7	F8	111 11000	110011 0001	001100 1110
D25.7	F9	111 11001	100110 1110	100110 0001
D26.7	FA	111 11010	010110 1110	010110 0001
D27.7	FB	111 11011	110110 0001	001001 1110
D28.7	FC	111 11100	001110 1110	001110 0001
D29.7	FD	111 11101	101110 0001	010001 1110
D30.7	FE	111 11110	011110 0001	100001 1110
D31.7	FF	111 11111	101011 0001	010100 1110

(concluded)

Table 36–2—Valid special code-groups

Code Group Name	Octet Value	Octet Bits HGF EDCBA	Current RD –	Current RD +	Notes
			abcdei fghj	abcdei fghj	
K28.0	1C	000 11100	001111 0100	110000 1011	1
K28.1	3C	001 11100	001111 1001	110000 0110	1,2
K28.2	5C	010 11100	001111 0101	110000 1010	1
K28.3	7C	011 11100	001111 0011	110000 1100	1
K28.4	9C	100 11100	001111 0010	110000 1101	1
K28.5	BC	101 11100	001111 1010	110000 0101	2
K28.6	DC	110 11100	001111 0110	110000 1001	1
K28.7	FC	111 11100	001111 1000	110000 0111	1,2
K23.7	F7	111 10111	111010 1000	000101 0111	
K27.7	FB	111 11011	110110 1000	001001 0111	
K29.7	FD	111 11101	101110 1000	010001 0111	
K30.7	FE	111 11110	011110 1000	100001 0111	

NOTE 1—Reserved.

NOTE 2—Contains a comma.

36.2.4.7.1 Ordered set rules

Ordered sets are specified according to the following rules:

a) Ordered sets consist of either one, two, or four code-groups.
b) The first code-group of all ordered sets is always a special code-group.
c) The second code-group of all multi-code-group ordered sets is always a data code-group. The second code-group is used to distinguish the ordered set from all other ordered sets. The second code-group provides a high bit transition density.

Table 36–3 lists the defined ordered sets.

Table 36–3—Defined ordered sets

Code	Ordered Set	Number of Code-Groups	Encoding
/C/	**Configuration**		Alternating /C1/ and /C2/
/C1/	Configuration 1	4	/K28.5/D21.5/Config_Reg[a]
/C2/	Configuration 2	4	/K28.5/D2.2/Config_Reg[a]
/I/	**IDLE**		Correcting /I1/, Preserving /I2/
/I1/	IDLE 1	2	/K28.5/D5.6/
/I2/	IDLE 2	2	/K28.5/D16.2/
	Encapsulation		
/R/	Carrier_Extend	1	/K23.7/
/S/	Start_of_Packet	1	/K27.7/
/T/	End_of_Packet	1	/K29.7/
/V/	Error_Propagation	1	/K30.7/
/LI/	**LPI**		Correcting /LI1/, Preserving /LI2/
/LI1/	LPI 1	2	/K28.5/D6.5/
/LI2/	LPI 2	2	/K28.5/D26.4/

[a]Two data code-groups representing the Config_Reg value.

36.2.4.8 /K28.5/ code-group considerations

The /K28.5/ special code-group is chosen as the first code-group of all ordered sets that are signaled repeatedly and for the purpose of allowing a receiver to synchronize to the incoming bit stream (i.e., /C/ and /I/), for the following reasons:

a) Bits abcdeif make up a comma. The comma can be used to easily find and verify code-group and ordered set boundaries of the rx_bit stream.
b) Bits ghj of the encoded code-group present the maximum number of transitions, simplifying receiver acquisition of bit synchronization.

36.2.4.9 Comma considerations

The seven bit comma string is defined as either b'0011111' (comma+) or b'1100000' (comma-). The /I/ and /C/ ordered sets and their associated protocols are specified to ensure that comma+ is transmitted with either equivalent or greater frequency than comma- for the duration of their transmission. This is done to ensure compatibility with common components.

The comma contained within the /K28.1/, /K28.5/, and /K28.7/ special code-groups is a singular bit pattern, which, in the absence of transmission errors, cannot appear in any other location of a code-group and cannot be generated across the boundaries of any two adjacent code-groups with the following exception:

The /K28.7/ special code-group is used by 1000BASE-X for diagnostic purposes only (see Annex 36A). This code-group, if followed by any of the following special or data code-groups: /K28.x/, /D3.x/, /D11.x/, /D12.x/, /D19.x/, /D20.x/, or /D28.x/, where x is a value in the range 0 to 7, inclusive, causes a comma to be

generated across the boundaries of the two adjacent code-groups. A comma across the boundaries of any two adjacent code-groups may cause code-group realignment (see 36.3.2.4).

36.2.4.10 Configuration (/C/)

Configuration, defined as the continuous repetition of the ordered sets /C1/ and /C2/, is used to convey the 16-bit Configuration Register (Config_Reg) to the link partner. See Clause 37 for a description of the Config_Reg contents.

The ordered sets, /C1/ and /C2/, are defined in Table 36–3. The /C1/ ordered set is defined such that the running disparity at the end of the first two code-groups is opposite that of the beginning running disparity. The /C2/ ordered set is defined such that the running disparity at the end of the first two code-groups is the same as the beginning running disparity. For a constant Config_Reg value, the running disparity after transmitting the sequence /C1/C2/ will be the opposite of what it was at the start of the sequence. This ensures that K28.5s containing comma+ will be sent during configuration.

36.2.4.11 Data (/D/)

A data code-group, when not used to distinguish or convey information for a defined ordered set, conveys one octet of arbitrary data between the GMII and the PCS. The sequence of data code-groups is arbitrary, where any data code-group can be followed by any other data code-group. Data code-groups are coded and decoded but not interpreted by the PCS. Successful decoding of the data code-groups depends on proper receipt of the Start_of_Packet delimiter, as defined in 36.2.4.14 and the checking of validity, as defined in 36.2.4.6.

36.2.4.12 IDLE (/I/)

IDLE ordered sets (/I/) are transmitted continuously and repetitively whenever the GMII is idle (TX_EN and TX_ER are both inactive). /I/ provides a continuous fill pattern to establish and maintain clock synchronization. /I/ is emitted from, and interpreted by, the PCS. /I/ consists of one or more consecutively transmitted /I1/ or /I2/ ordered sets, as defined in Table 36–3.

The /I1/ ordered set is defined such that the running disparity at the end of the transmitted /I1/ is opposite that of the beginning running disparity. The /I2/ ordered set is defined such that the running disparity at the end of the transmitted /I2/ is the same as the beginning running disparity. The first /I/ following a packet or Configuration ordered set restores the current positive or negative running disparity to a negative value. All subsequent /I/s are /I2/ to ensure negative ending running disparity.

Distinct carrier events are separated by /I/s.

Implementations of this standard may benefit from the ability to add or remove /I2/ from the code-group stream one /I2/ at a time without altering the beginning running disparity associated with the code-group subsequent to the removed /I2/.

A received ordered set that consists of two code-groups, the first of which is /K28.5/ and the second of which is a data code-group other than /D21.5/ or /D2.2/ (or /D6.5/ or /D26.4/ to support EEE capability), is treated as an /I/ ordered set.

36.2.4.13 Low Power Idle (LPI)

LPI is transmitted in the same manner as IDLE. LPI ordered sets (/LI/) are transmitted continuously and repetitively whenever the GMII is indicating “Assert LPI”. See 35.2.2.6 and 35.2.2.10 for corresponding GMII definitions.

36.2.4.14 Start_of_Packet (SPD) delimiter

A Start_of_Packet delimiter (SPD) is used to delineate the starting boundary of a data transmission sequence and to authenticate carrier events. Upon each fresh assertion of TX_EN by the GMII, and subsequent to the completion of PCS transmission of the current ordered set, the PCS replaces the current octet of the MAC preamble with SPD. Upon initiation of packet reception, the PCS replaces the received SPD delimiter with the data octet value associated with the first preamble octet. A SPD delimiter consists of the code-group /S/, as defined in Table 36–3.

SPD follows /I/ for a single packet or the first packet in a burst.

SPD follows /R/ for the second and subsequent packets of a burst.

36.2.4.15 End_of_Packet delimiter (EPD)

An End_of_Packet delimiter (EPD) is used to delineate the ending boundary of a packet. The EPD is transmitted by the PCS following each deassertion of TX_EN on the GMII, which follows the last data octet comprising the FCS of the MAC packet. On reception, EPD is interpreted by the PCS as terminating a packet. A EPD delimiter consists of the code-groups /T/R/R/ or /T/R/K28.5/. The code-group /T/ is defined in Table 36–3. See 36.2.4.16 for the definition of code-groups used for /R/. /K28.5/ normally occurs as the first code-group of the /I/ ordered set. See 36.2.4.12 for the definition of code-groups used for /I/.

The receiver considers the MAC interpacket gap (IPG) to have begun two octets prior to the transmission of /I/. For example, when a packet is terminated by EPD, the /T/R/ portion of the EPD occupies part of the region considered by the MAC to be the IPG.

36.2.4.15.1 EPD rules

a) The PCS transmits a /T/R/ following the last data octet from the MAC;
b) If the MAC indicates carrier extension to the PCS, Carrier_Extend rules are in effect. See 36.2.4.16.1;
c) If the MAC does not indicate carrier extension to the PCS, perform the following:
    1) If /R/ is transmitted in an even-numbered code-group position, the PCS appends a single additional /R/ to the code-group stream to ensure that the subsequent /I/ is aligned on an even-numbered code-group boundary and EPD transmission is complete;
    2) The PCS transmits /I/.

36.2.4.16 Carrier_Extend (/R/)

Carrier_Extend (/R/) is used for the following purposes:

a) Carrier extension: Used by the MAC to extend the duration of the carrier event. When used for this purpose, carrier extension is emitted from and interpreted by the MAC and coded to and decoded from the corresponding code-group by the PCS. In order to extend carrier, the GMII must deassert TX_EN. The deassertion of TX_EN and simultaneous assertion of TX_ER causes the PCS to emit an /R/ with a two-octet delay, which gives the PCS time to complete its EPD before commencing transmissions. The number of /R/ code-groups emitted from the PCS equals the number of GMII GTX_CLK periods during which it extends carrier;
b) Packet separation: Carrier extension is used by the MAC to separate packets within a burst of packets. When used for this purpose, carrier extension is emitted from and interpreted by the MAC and coded to and decoded from the corresponding code-group by the PCS;
c) EPD2: The first /R/ following the /T/ in the End_of_Packet delimiters /T/R/I/ or /T/R/R/I/;

d) EPD3: The second /R/ following the /T/ in the End_of_Packet delimiter /T/R/R/I/. This /R/ is used, if necessary, to pad the only or last packet of a burst of packets so that the subsequent /I/ is aligned on an even-numbered code-group boundary. When used for this purpose, Carrier_Extend is emitted from, and interpreted by, the PCS. An EPD of /T/R/R/ results in one /R/ being delivered to the PCS client (see 36.2.4.15.1).

Carrier_Extend consists of one or more consecutively transmitted /R/ ordered sets, as defined in Table 36–3.

36.2.4.16.1 Carrier_Extend rules

a) If the MAC indicates carrier extension to the PCS, the initial /T/R/ is followed by one /R/ for each octet of carrier extension received from the MAC;
b) If the last /R/ is transmitted in an even-numbered code-group position, the PCS appends a single additional /R/ to the code-group stream to ensure that the subsequent /I/ is aligned on an even-numbered code-group boundary.

36.2.4.17 Error_Propagation (/V/)

Error_Propagation (/V/) indicates that the PCS client wishes to indicate a transmission error to its peer entity. The normal use of Error_Propagation is for repeaters to propagate received errors. /V/ is emitted from the PCS, at the request of the PCS client through the use of the TX_ER signal, as described in Clause 35. Error_Propagation is emitted from, and interpreted by, the PCS. Error_Propagation consists of the ordered set /V/, as defined in Table 36–3.

The presence of Error_Propagation or any invalid code-group on the medium denotes a collision artifact or an error condition. Invalid code-groups are not intentionally transmitted onto the medium by DTEs. The PCS processes and conditionally indicates the reception of /V/ or an invalid code-group on the GMII as false carrier, data errors, or carrier extend errors, depending on its current context.

36.2.4.18 Encapsulation

The 1000BASE-X PCS accepts packets from the MAC through the Reconciliation sublayer and GMII. Due to the continuously signaled nature of the underlying PMA, and the encoding performed by the PCS, the 1000BASE-X PCS encapsulates MAC frames into a code-group stream. The PCS decodes the code-group stream received from the PMA, extracts packets from it, and passes the packets to the MAC via the Reconciliation sublayer and GMII.

Figure 36–4 depicts the PCS encapsulation of a MAC packet based on GMII signals.

36.2.4.19 Mapping between GMII, PCS and PMA

Figure 36–3 depicts the mapping of the octet-wide data path of the GMII to the ten-bit-wide code-groups of the PCS, and the one-bit paths of the PMA/PMD interface.

The PCS encodes an octet received from the GMII into a ten-bit code-group, according to Figure 36–3. Code-groups are serialized into a tx_bit stream by the PMA and passed to the PMD for transmission on the underlying medium, according to Figure 36–3. The first transmitted tx_bit is tx_code-group<0>, and the last tx_bit transmitted is tx_code-group<9>. There is no numerical significance ascribed to the bits within a code-group; that is, the code-group is simply a ten-bit pattern that has some predefined interpretation.

Similarly, the PMA deserializes rx_bits received from the PMD, according to Figure 36–3. The PCS Receive process converts rx_code-group<9:0>'s into GMII data octets, according to 36.2.5.2.2.

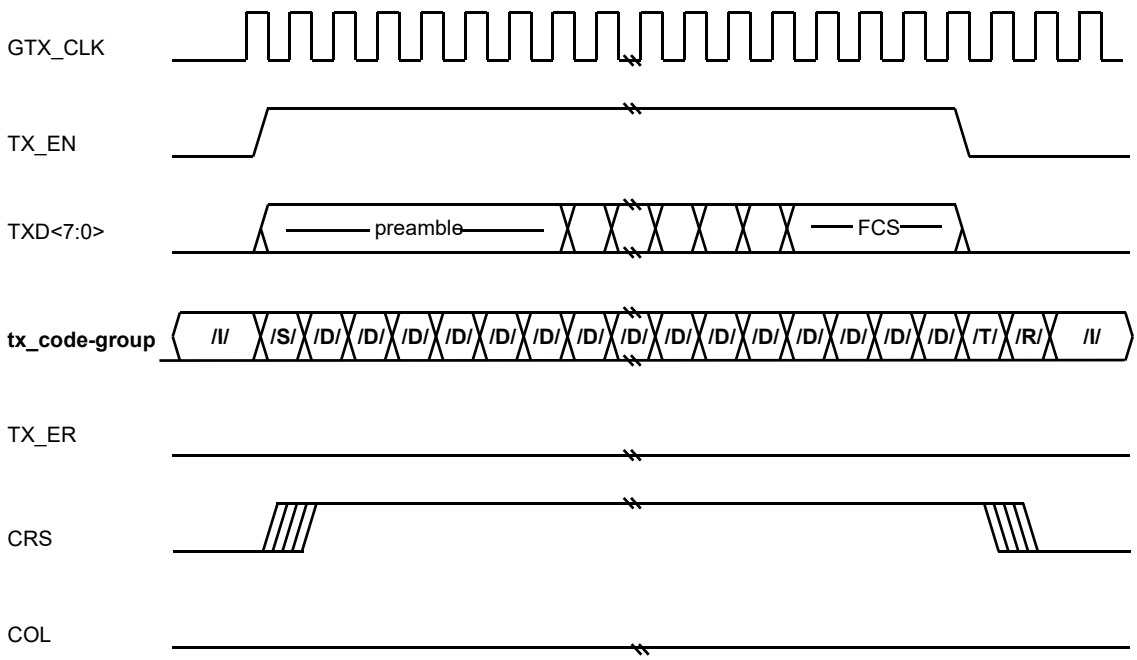


Figure 36–4—PCS encapsulation

36.2.5 Detailed functions and state diagrams

The notation used in the state diagrams in this clause follow the conventions in 21.5. State diagram variables follow the conventions of 21.5.2 except when the variable has a default value. Variables in a state diagram with default values evaluate to the variable default in each state where the variable value is not explicitly set.

Timeless states are employed as an editorial convenience to facilitate the distribution of transition conditions from prior states. No actions are taken within these states. Exit conditions are evaluated for timeless states. Timeless states are as follows:

a) PCS transmit ordered set state TX_PACKET;
b) PCS transmit code-group state GENERATE_CODE_GROUPS;
c) PCS transmit code-group state IDLE_DISPARITY_TEST;
d) PCS receive state RECEIVE;
e) PCS receive state EPD2_CHECK_END.

36.2.5.1 State variables

36.2.5.1.1 Notation conventions

/x/

Denotes the constant code-group specified in 36.2.5.1.2 (valid code-groups must follow the rules of running disparity as per 36.2.4.5 and 36.2.4.6).

[/x/]

Denotes the latched received value of the constant code-group (/x/) specified in 36.2.5.1.2 and conveyed by the SYNC_UNITDATA.indicate message described in 36.2.5.1.6.

36.2.5.1.2 Constants

/C/
: The Configuration ordered set group, comprising either the /C1/ or /C2/ ordered set, as specified in 36.2.4.10. Conveys the Config_Reg value as tx_Config_Reg<D15:D0> for the PCS Transmit process and rx_Config_Reg<D15:D0> for the PCS Receive process.

/COMMA/
: The set of special code-groups which include a comma as specified in 36.2.4.9 and listed in Table 36–2.

/D/
: The set of 256 code-groups corresponding to valid data, as specified in 36.2.4.11.

/Dx.y/
: One of the set of 256 code-groups corresponding to valid data, as specified in 36.2.4.11.

/I/
: The IDLE ordered set group, comprising either the /I1/ or /I2/ ordered sets, as specified in 36.2.4.12.

/INVALID/
: The set of invalid data or special code-groups, as specified in 36.2.4.6.

/Kx.y/
: One of the set of 12 code-groups corresponding to valid special code-groups, as specified in Table 36–2.

/R/
: The code-group used as either: End_of_Packet delimiter part 2; End_of_Packet delimiter part 3; Carrier_Extend; and /I/ alignment.

/S/
: The code-group corresponding to the Start_of_Packet delimiter (SPD) as specified in 36.2.4.14.

/T/
: The code-group used for the End_of_Packet delimiter part 1.

/V/
: The Error_Propagation code-group, as specified in 36.2.4.17.

The following constant is used only for the EEE capability:

/LI/
: The LP_IDLE ordered set group, comprising either the /LI1/ or /LI2/ ordered sets, as specified in 36.2.4.13.

36.2.5.1.3 Variables

cgbad
: Alias for the following terms: ((rx_code-group∈/INVALID/) + (rx_code-group=/COMMA/ *rx_even=TRUE)) * PMA_UNITDATA.indication

cggood
: Alias for the following terms: !((rx_code-group∈/INVALID/) + (rx_code-group=/COMMA/ *rx_even=TRUE)) * PMA_UNITDATA.indication

COL

The COL signal of the GMII as specified in Clause 35.

CRS

The CRS signal of the GMII as specified in Clause 35.

EVEN

The latched state of the rx_even variable, when rx_even=TRUE, as conveyed by the SYNC_UNITDATA.indicate message described in 36.2.5.1.6.

mr_loopback

A Boolean that indicates the enabling and disabling of data being looped back through the PHY. Loopback of data through the PHY is enabled when Control register bit 0.14 is set to one.

Values: FALSE; Loopback through the PHY is disabled.
TRUE; Loopback through the PHY is enabled.

mr_main_reset

Controls the resetting of the PCS via Control Register bit 0.15.

Values: FALSE; Do not reset the PCS.
TRUE; Reset the PCS.

ODD

The latched state of the rx_even variable, when rx_even=FALSE, as conveyed by the SYNC_UNITDATA.indicate message described in 36.2.5.1.6.

power_on

Condition that is true until such time as the power supply for the device that contains the PCS has reached the operating region. The condition is also true when the device has low power mode set via Control register bit 0.11.

Values: FALSE; The device is completely powered (default).
TRUE; The device has not been completely powered.

NOTE—Power_on evaluates to its default value in each state where it is not explicitly set.

receiving

A Boolean set by the PCS Receive process to indicate carrier activity. Used by the Carrier Sense process, and also interpreted by the PCS Transmit process for indicating a collision. (See also 36.2.5.1.4, carrier_detect(x).)

Values: TRUE; Carrier being received.
FALSE; Carrier not being received.

repeater_mode

A Boolean used to make the assertion of Carrier Sense occur only in response to receive activity when the PCS is used in a CSMA/CD repeater. This variable is set to TRUE in a repeater application, and set to FALSE in all other applications.

Values: TRUE; Allows the assertion of CRS in response to receive activity only.
FALSE; Allows the assertion of CRS in response to either transmit or receive activity.

rx_bit

A binary parameter conveyed by the PMD_UNITDATA.indication service primitive, as specified in 38.1.1.2, to the PMA.

Values: ZERO; Data bit is a logical zero.
ONE; Data bit is a logical one.

rx_code-group<9:0>

A 10-bit vector represented by the most recently received code-group from the PMA. The element rx_code-group<0> is the least recently received (oldest) rx_bit; rx_code-group<9> is the most recently received rx_bit (newest). When code-group alignment has been achieved, this vector contains precisely one code-group.

rx_Config_Reg<D15:D0>

A 16-bit array that contains the data bits received from a /C/ ordered set as defined in 36.2.4.10. Conveyed by the PCS Receive process to the PCS Auto-Negotiation process. The format of the data bits is context dependent, relative to the state of the Auto-Negotiation function, and is presented in 37.2.1.1 and 37.2.4.3.1. For each element within the array:

Values: ZERO; Data bit is a logical zero.
ONE; Data bit is a logical one.

RX_DV

The RX_DV signal of the GMII as specified in Clause 35. Set by the PCS Receive process.

RX_ER

The RX_ER signal of the GMII as specified in Clause 35. Set by the PCS Receive process.

rx_even

A Boolean set by the PCS Synchronization process to designate received code-groups as either even- or odd-numbered code-groups as specified in 36.2.4.2.

Values: TRUE; Even-numbered code-group being received.
FALSE; Odd-numbered code-group being received.

RXD<7:0>

The RXD<7:0> signal of the GMII as specified in Clause 35. Set by the PCS Receive process.

signal_detect

A Boolean set by the PMD continuously via the PMD_SIGNAL.indication(signal_detect) message to indicate the status of the incoming link signal.

Values: FAIL; A signal is not present on the link.
OK; A signal is present on the link.

sync_status

A parameter set by the PCS Synchronization process to reflect the status of the link as viewed by the receiver. The values of the parameter are defined for code_sync_status. The equation for this parameter is

sync_status = code_sync_status + rx_lpi_active

NOTE—If EEE is not supported, the variable rx_lpi_active is always false, and this variable is identical to code_sync_status controlled by the synchronization state diagram.

transmitting

A Boolean set by the PCS Transmit process to indicate that packet transmission is in progress. Used by the Carrier Sense process and internally by the PCS Transmit process for indicating a collision.

Values: TRUE; The PCS is transmitting a packet.
FALSE; The PCS is not transmitting a packet.

tx_bit

A binary parameter used to convey data from the PMA to the PMD via the PMD_UNITDATA.request service primitive as specified in 38.1.1.1.

Values: ZERO; Data bit is a logical zero.
ONE; Data bit is a logical one.

tx_code-group<9:0>

A vector of bits representing one code-group, as specified in Tables 36–1a–e or 36–2, which has been prepared for transmission by the PCS Transmit process. This vector is conveyed to the PMA as the parameter of a PMD_UNITDATA.request(tx_bit) service primitive. The element tx_code-group<0> is the first tx_bit transmitted; tx_code-group<9> is the last tx_bit transmitted.

tx_Config_Reg<D15:D0>

A 16-bit array that contains the data bits to be transmitted in a /C/ ordered set as defined in 36.2.4.10. Conveyed by the PCS Auto-Negotiation process to the PCS Transmit process. The format of the data bits is context dependent, relative to the state of the Auto-Negotiation function, and is presented in 37.2.1.1 and 37.2.4.3.1. For each element within the array:

Values: ZERO; Data bit is a logical zero.
ONE; Data bit is a logical one.

tx_disparity

A Boolean set by the PCS Transmit process to indicate the running disparity at the end of code-group transmission as a binary value. Running disparity is described in 36.2.4.3.

Values: POSITIVE
NEGATIVE

TX_EN

The TX_EN signal of the GMII as specified in Clause 35.

TX_ER

The TX_ER signal of the GMII as specified in Clause 35.

tx_even

A Boolean set by the PCS Transmit process to designate transmitted code-groups as either even- or odd-numbered code-groups as specified in 36.2.4.2.

Values: TRUE; Even-numbered code-group being transmitted.
FALSE; Odd-numbered code-group being transmitted.

tx_o_set

One of the following defined ordered sets: /C/, /T/, /R/, /I/, /LI/, /S/, /V/, or the code-group /D/.

TXD<7:0>

The TXD<7:0> signal of the GMII as specified in Clause 35.

xmit

Defined in 37.3.1.1.

The following variables are used only for the EEE capability:

assert_lpidle

Alias used for the optional LPI function, consisting of the following terms:

(TX_EN=FALSE * TX_ER=TRUE * (TXD<7:0>=0x01))

code_sync_status

A parameter set by the PCS Synchronization process to reflect the status of the link as viewed by the receiver.

Values: FAIL; The receiver is not synchronized to code-group boundaries.
OK; The receiver is synchronized to code-group boundaries.

idle_d

Alias for the following terms:

SUDI(![/D21.5/] * ![/D2.2/])
that uses an alternate form to support the EEE capability:
SUDI(![/D21.5/] * ![/D2.2/] * ![/D6.5/] * ![/D26.4/])

rx_lpi_active

A Boolean variable that is set to TRUE when the receiver is in a low power state and set to FALSE when it is in an active state and capable of receiving data.

rx_quiet

A Boolean variable set to TRUE while in the RX_QUIET state and set to FALSE otherwise.

tx_quiet

A Boolean variable set to TRUE when the transmitter is in the TX_QUIET state and set to FALSE otherwise. When set to TRUE, the PMD will disable the transmitter as described in 70.6.5.

36.2.5.1.4 Functions

carrier_detect

In the PCS Receive process, this function uses for input the latched code-group ([/x/]) and latched rx_even (EVEN/ODD) parameters of the SYNC_UNITDATA.indicate message from the PCS Synchronization process. When SYNC_UNITDATA.indicate message indicates EVEN, the carrier_detect function detects carrier when either:

a) A two or more bit difference between [/x/] and both /K28.5/ encodings exists (see Table 36–2); or

b) A two to nine bit difference between [/x/] and the expected /K28.5/ (based on current running disparity) exists.

Values: TRUE; Carrier is detected.
FALSE; Carrier is not detected.

check_end

Prescient End_of_Packet and Carrier_Extend function used by the PCS Receive process to set RX_ER and RXD<7:0> signals. The check_end function returns the current and next two code-groups in rx_code-group<9:0>.

DECODE ([/x]/)

In the PCS Receive process, this function takes as its argument the latched value of rx_code-group<9:0> ([/x/]) and the current running disparity, and returns the corresponding GMII RXD<7:0>, rx_Config_Reg<D7:D0>, or rx_Config_Reg<D15:D8> octet, per Table 36–1a–e. DECODE also updates the current running disparity per the running disparity rules outlined in 36.2.4.4.

ENCODE(x)

In the PCS Transmit process, this function takes as its argument (x), where x is a GMII TXD<7:0>, tx_Config_Reg<D7:D0>, or tx_Config_Reg<D15:D8> octet, and the current running disparity, and returns the corresponding ten-bit code-group per Table 36–1a–e. ENCODE also updates the current running disparity per Table 36–1a–e.

signal_detectCHANGE

In the PCS Synchronization process, this function monitors the signal_detect variable for a state change. The function is set upon state change detection.

Values: TRUE; A signal_detect variable state change has been detected.
FALSE; A signal_detect variable state change has not been detected (default).

NOTE—Signal_detectCHANGE is set by this function definition; it is not set explicitly in the state diagrams. Signal_detectCHANGE evaluates to its default value upon state entry.

VOID(x)

x ∈ /D/, /T/, /R/, /K28.5/. Substitutes /V/ on a per code-group basis as requested by the GMII.

If [TX_EN=FALSE * TX_ER=TRUE * TXD≠(0000 1111)],

then return /V/;

Else if [TX_EN=TRUE * TX_ER=TRUE],

then return /V/;

Else return x.

xmitCHANGE

In the PCS Transmit process, this function monitors the xmit variable for a state change. The function is set upon state change detection.

Values: TRUE; An xmit variable state change has been detected.
FALSE; An xmit variable state change has not been detected (default).

NOTE—XmitCHANGE is set by this function definition; it is not set explicitly in the state diagrams. XmitCHANGE evaluates to its default value upon entry to state TX_TEST_XMIT.

36.2.5.1.5 Counters

good_cgs

Count of consecutive valid code-groups received.

The following counter is used only for the EEE capability:

wake_error_counter

A counter that is incremented each time that the LPI receive state diagram enters the RX_WTF state indicating that a wake time fault has been detected. The counter is reflected in register 3.22 (see 45.2.3.12).

36.2.5.1.6 Messages

PMA_UNITDATA.indication(rx_code-group<9:0>)

A signal sent by the PMA Receive process conveying the next code-group received over the medium (see 36.3.1.2).

PMA_UNITDATA.request(tx_code-group<9:0>)
A signal sent to the PMA Transmit process conveying the next code-group ready for transmission over the medium (see 36.3.1.1).

PMD_SIGNAL.indication(signal_detect)
A signal sent by the PMD to indicate the status of the signal being received on the MDI.

PUDI
Alias for PMA_UNITDATA.indication(rx_code-group<9:0>).

PUDR
Alias for PMA_UNITDATA.request(tx_code-group<9:0>).

RUDI
Alias for RX_UNITDATA.indicate(parameter).

RX_UNITDATA.indicate(parameter)
A signal sent by the PCS Receive process to the PCS Auto_Negotiation process conveying the following parameters:

Parameters: INVALID; indicates that an error condition has been detected while receiving /C/ or /I/ ordered sets;
/C/; the /C/ ordered set has been received;
/I/; the /I/ ordered set has been received.

SUDI
Alias for SYNC_UNITDATA.indicate(parameters).

SYNC_UNITDATA.indicate(parameters)
A signal sent by the PCS Synchronization process to the PCS Receive process conveying the following parameters:

Parameters: [/x/]; the latched value of the indicated code-group (/x/);
EVEN/ODD; The latched state of the rx_even variable;

Value: EVEN; Passed when the latched state of rx_even=TRUE.
ODD; Passed when the latched state of rx_even=FALSE.

TX_OSET.indicate
A signal sent to the PCS Transmit ordered set process from the PCS Transmit code-group process signifying the completion of transmission of one ordered set.

The following messages are used only for the EEE capability:

PMD_RXQUIET.request(rx_quiet)
A signal sent by the PCS/PMA LPI receive state diagram to the PMD. Note that this message is ignored by devices that do not support EEE capability.

Values: TRUE: The receiver is in a quiet state and is not expecting incoming data.
FALSE: The receiver is ready to receive data.

PMD_TXQUIET.request(tx_quiet)
A signal sent by the PCS/PMA LPI transmit state diagram to the PMD. Note that this message is ignored by devices that do not support the optional LPI mechanism.

Values: TRUE: The transmitter is in a quiet state and may cease to transmit a signal on the medium.
FALSE: The transmitter is ready to transmit data.

36.2.5.1.7 Timers

cg_timer
: A continuous free-running timer.

  Values: The condition cg_timer_done becomes true upon timer expiration.

  Restart when: immediately after expiration; restarting the timer resets the condition cg_timer_done.

  Duration: 8 ns nominal.

  If the GMII is implemented, cg_timer shall expire synchronously with the rising edge of GTX_CLK (see tolerance required for GTX_CLK in 35.5.2.3). In the absence of a GMII, cg_timer shall expire every 8 ns ± 0.01%. In the PCS transmit code-group state diagram, the message PMA_UNITDATA.request is issued concurrently with cg_timer_done.

The following timers are used only for the EEE capability:

rx_tq_timer
: This timer is started when the PCS receiver enters the START_TQ_TIMER state. The timer terminal count is set to T_{QR}. When the timer reaches terminal count, it will set the rx_tq_timer_done = TRUE.

rx_tw_timer
: This timer is started when the PCS receiver enters the RX_WAKE state. The timer terminal count shall not exceed the maximum value of T_{WR} in Table 36–9. When the timer reaches terminal count, it will set the rx_tw_timer_done = TRUE.

rx_wf_timer
: This timer is started when the PCS receiver enters the RX_WTF state, indicating that the receiver has encountered a wake time fault. The rx_wf_timer allows the receiver an additional period in which to synchronize or return to the quiescent state before a link failure is indicated. The timer terminal count is set to T_{WTF}. When the timer reaches terminal count, it will set the rx_wf_timer_done = TRUE.

tx_ts_timer
: This timer is started when the PCS transmitter enters the TX_SLEEP state. The timer terminal count is set to T_{SL}. When the timer reaches terminal count, it will set the tx_ts_timer_done = TRUE.

tx_tq_timer
: This timer is started when the PCS transmitter enters the TX_QUIET state. The timer terminal count is set to T_{QL}. When the timer reaches terminal count, it will set the tx_tq_timer_done = TRUE.

tx_tr_timer
: This timer is started when the PCS transmitter enters the TX_REFRESH state. The timer terminal count is set to T_{UL}. When the timer reaches terminal count, it will set the tx_tr_timer_done = TRUE.

36.2.5.2 State diagrams

36.2.5.2.1 Transmit

The PCS Transmit process is depicted in two state diagrams: PCS Transmit ordered set and PCS Transmit code-group. The PCS shall implement its Transmit process as depicted in Figure 36–5 and Figure 36–6, including compliance with the associated state variables as specified in 36.2.5.1.

The Transmit ordered set process continuously sources ordered sets to the Transmit code-group process. When initially invoked, and when the Auto-Negotiation process xmit flag indicates CONFIGURATION, the Auto-Negotiation process is invoked. When the Auto-Negotiation process xmit flag indicates IDLE, and between packets (as delimited by the GMII), /I/ is sourced. Upon the assertion of TX_EN by the GMII when the Auto-Negotiation process xmit flag indicates DATA, the SPD ordered set is sourced. Following the SPD, /D/ code-groups are sourced until TX_EN is deasserted. Following the deassertion of TX_EN, EPD ordered sets are sourced. If TX_ER is asserted when TX_EN is deasserted and carrier extend error is not indicated by TXD, /R/ ordered sets are sourced for as many GTX_CLK periods as TX_ER is asserted with a delay of two GTX_CLK periods to first source the /T/ and /R/ ordered sets. If carrier extend error is indicated by TXD during carrier extend, /V/ ordered sets are sourced. If TX_EN and TX_ER are both deasserted, the /R/ ordered set may be sourced, after which the sourcing of /I/ is resumed. If, while TX_EN is asserted, the TX_ER signal is asserted, the /V/ ordered set is sourced except when the SPD ordered set is selected for sourcing.

Collision detection is implemented by noting the occurrence of carrier receptions during transmissions, following the models of 10BASE-T and 100BASE-X.

The Transmit code-group process continuously sources tx_code-group<9:0> to the PMA based on the ordered sets sourced to it by the Transmit ordered set process. The Transmit code-group process determines the proper code-group to source based on even/odd-numbered code-group alignment, running disparity requirements, and ordered set format.

36.2.5.2.2 Receive

The PCS shall implement its Receive process as depicted in Figure 36–7a and Figure 36–7b, including compliance with the associated state variables as specified in 36.2.5.1.

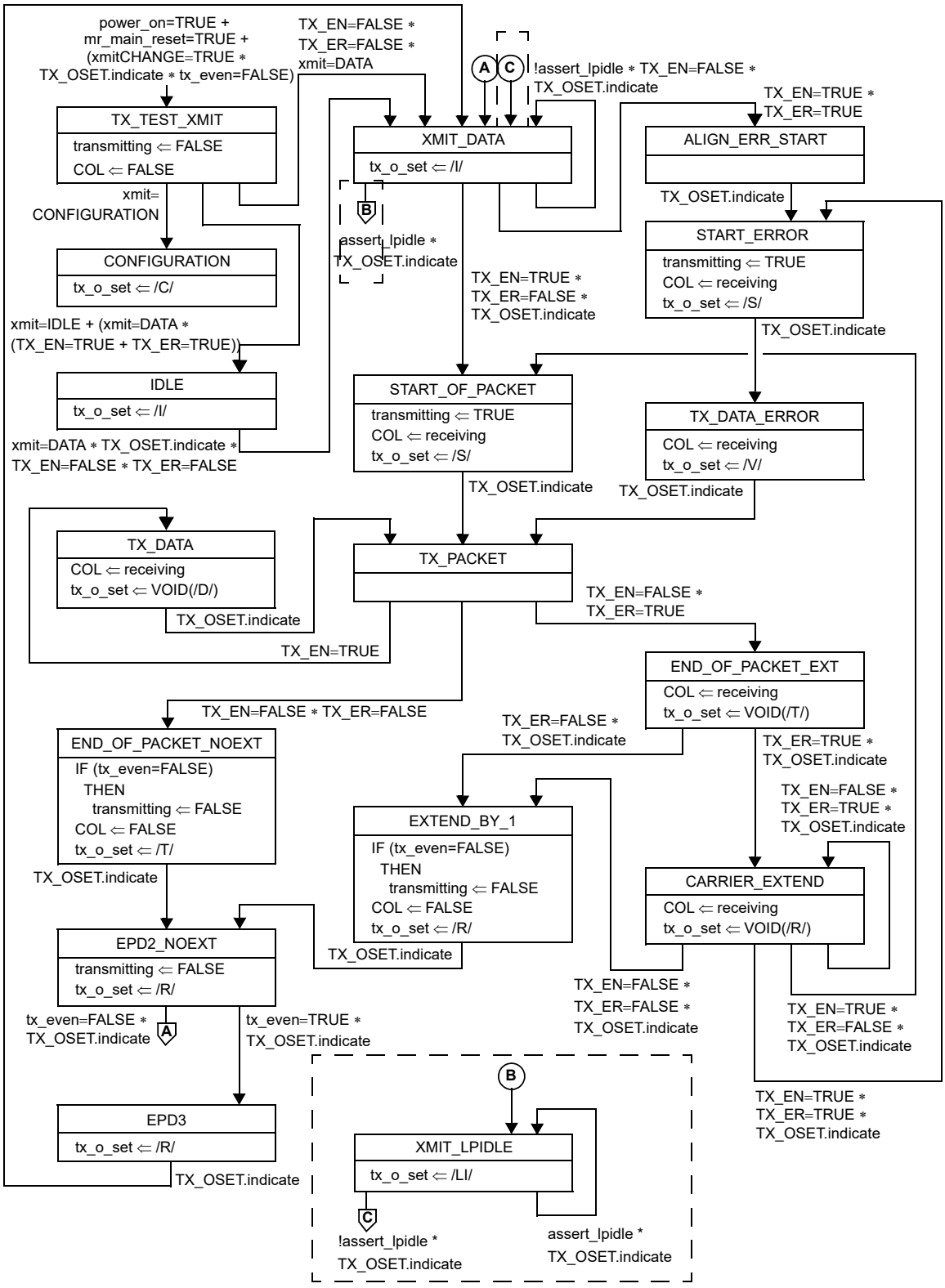
The PCS Receive process continuously passes RXD<7:0> and sets the RX_DV and RX_ER signals to the GMII based on the received code-group from the PMA.

When the Auto-Negotiation process xmit flag indicates CONFIGURATION or IDLE, the PCS Receive process continuously passes /C/ and /I/ ordered sets and rx_Config_Reg<D15:D0> to the Auto-Negotiation process.

36.2.5.2.3 State variable function carrier_detect(x)

The detection of carrier on the underlying channel is used both by the MAC (via the GMII CRS signal and the Reconciliation sublayer) for deferral purposes, and by the PCS Transmit process for collision detection. A carrier event, signaled by the assertion of receiving, is indicated by the detection of a difference between the received code-group and /K28.5/ as specified in 36.2.5.1.4.

A carrier event is in error if it does not start with an SPD. The PCS Receive process performs this function by continuously monitoring incoming code-groups for specific patterns that indicate non-/I/ activity such as SPD. The detection of an SPD carrier event causes the PCS to substitute the value (01010101) for the SPD, set RXD<7:0> to this value, and assert RX_DV. The pattern substituted for the SPD is consistent with the preamble pattern expected by the MAC. The detection of a non-SPD carrier event (false carrier) causes the PCS to substitute the value (00001110) for the code-group received, set RXD<7:0> to this value, and assert RX_ER.



NOTE—Transitions B and C are only required for the EEE capability.

Figure 36–5—PCS transmit ordered set state diagram

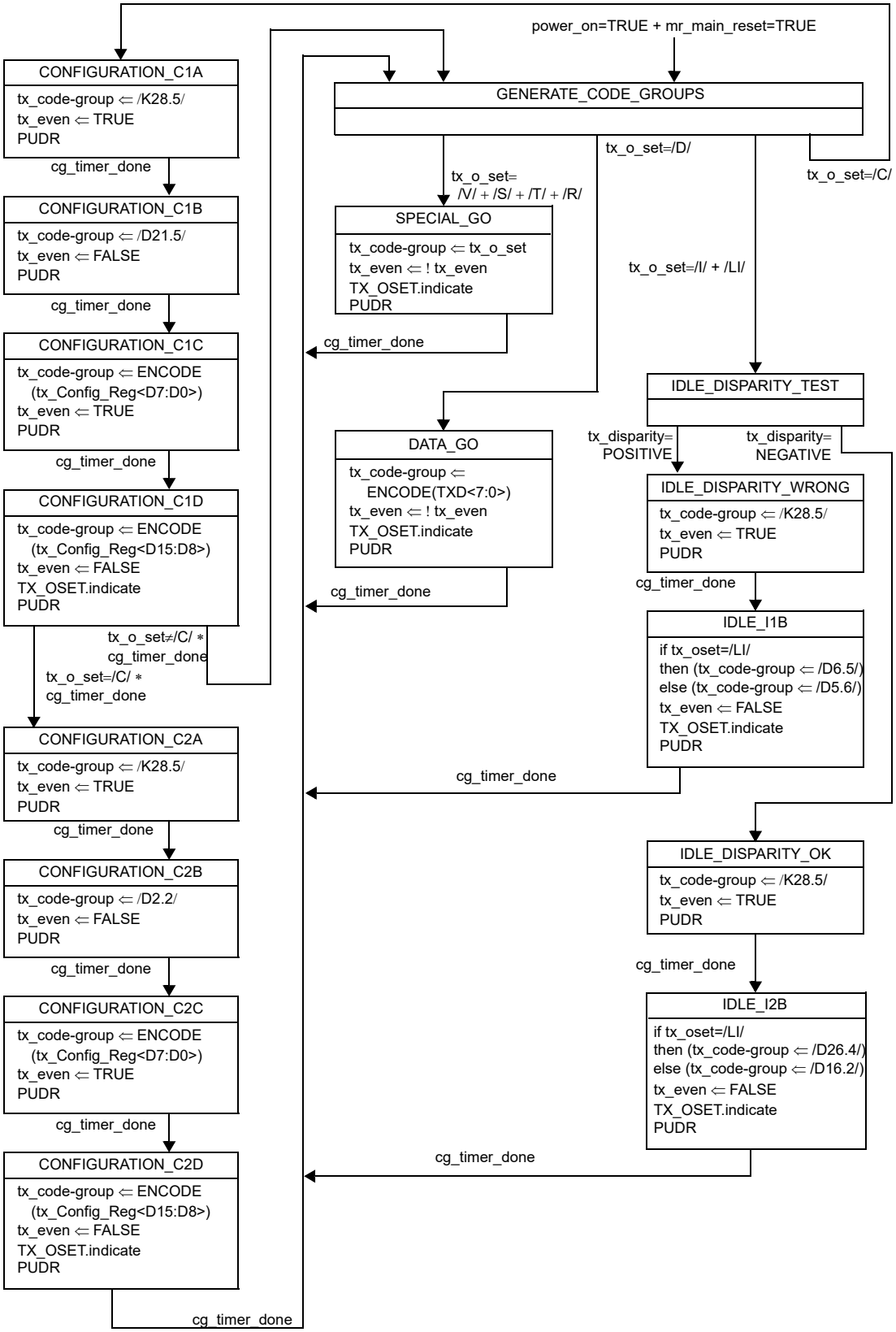
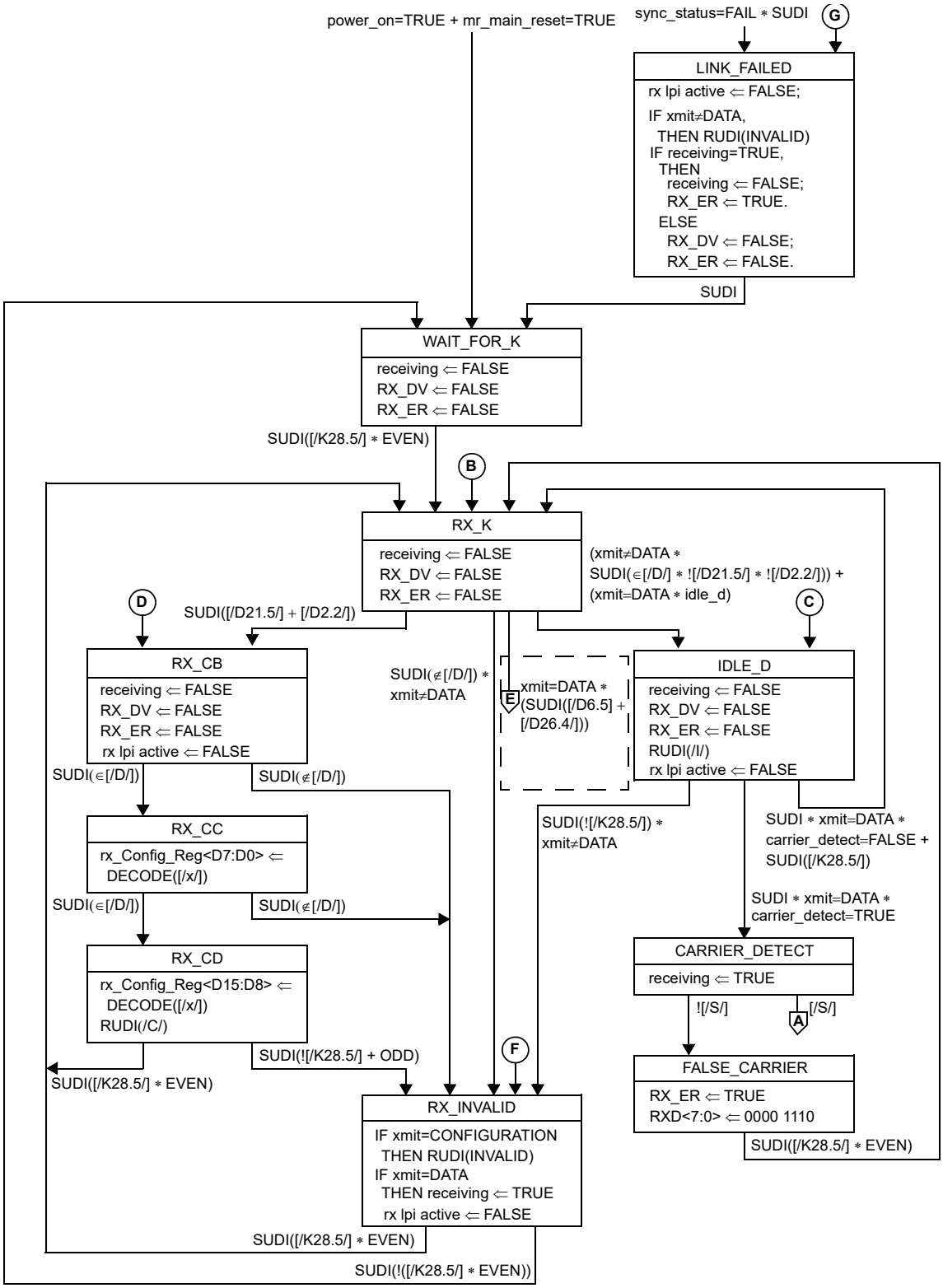
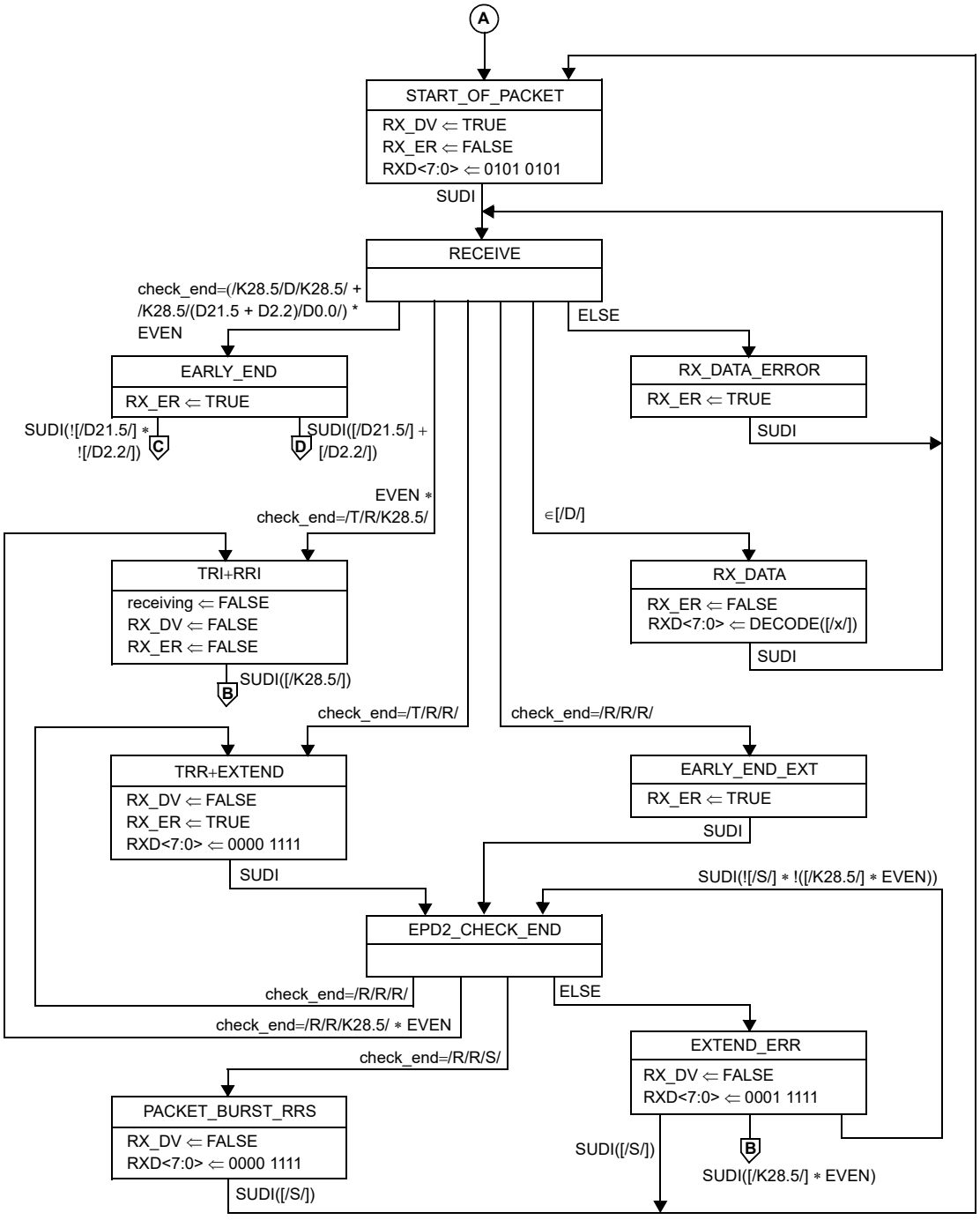


Figure 36–6—PCS transmit code-group state diagram



NOTE—Outgoing arcs leading to labeled polygons flow offpage to corresponding incoming arcs leading from labeled circles on Figure 36–7b and Figure 36–7c, and vice versa.

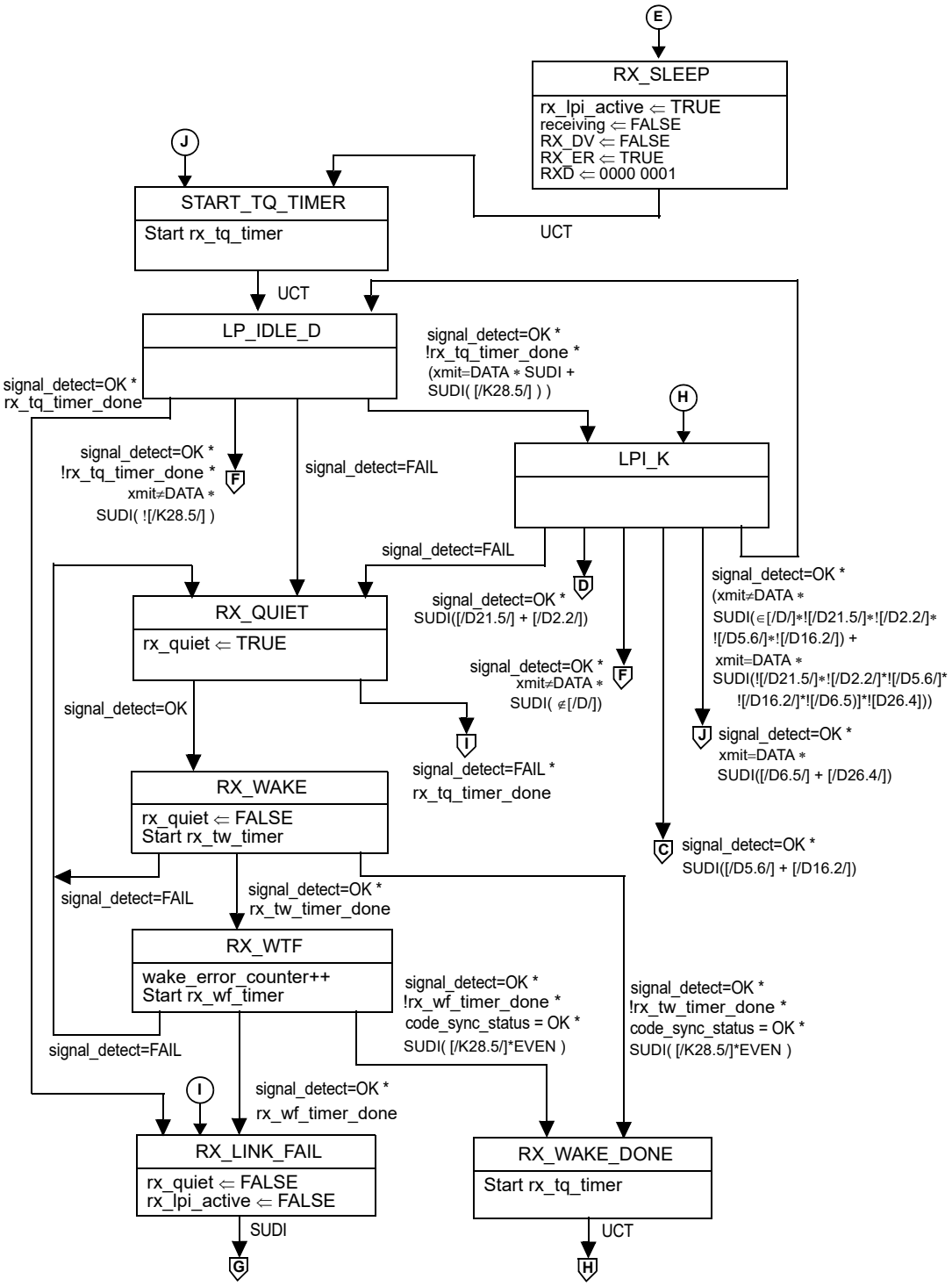
Figure 36–7a—PCS receive state diagram, part a



NOTE 1—Outgoing arcs leading to labeled polygons flow offpage to corresponding incoming arcs leading from labeled circles on Figure 36–7a, and vice versa.

NOTE 2—In the transition from RECEIVE to RX_DATA state the transition condition is a test against the code-group obtained from the SUDI that caused the transition to RECEIVE state.

Figure 36–7b—PCS receive state diagram, part b



NOTE—Outgoing arcs leading to labeled polygons flow off page to corresponding incoming arcs leading from labeled circles on Figure 36–7a, and vice versa.

Figure 36–7c—PCS Receive state diagram, part c
(only required for the optional EEE capability)

36.2.5.2.4 Code-group stream decoding

Subsequent to the detection of an SPD carrier event, the PCS Receive process performs the DECODE function on the incoming code-groups, passing decoded data to the GMII, including those corresponding to the remainder of the MAC preamble and SFD. The GMII signal RX_ER is asserted upon decoding any code-group following the SPD that neither is a valid /D/ code-group nor follows the EPD rules in 36.2.4.15.1.

Packets are terminated with an EPD as specified in 36.2.4.15. The PCS Receive process performs the check_end function to preserve the ability of the MAC to properly delimit the FCS at the end of a packet.

Detection of /T/R/R/ or /T/R/K28.5/ by the check_end function denotes normal (i.e., non-error) packet termination. Detection of /R/R/R/ by the check_end function denotes packet termination with error and Carrier_Extend processing. Detection of /K28.5/D/K28.5/ by the check_end function denotes packet termination with error. Detection of /K28.5/(D21.5 or D2.2)/D0.0 by the check_end function denotes packet termination with error.

36.2.5.2.5 Carrier sense

The Carrier Sense process generates the signal CRS on the GMII, which (via the Reconciliation sublayer) the MAC uses for deferral.

The PCS shall implement the Carrier Sense process as depicted in Figure 36–8 including compliance with the associated state variables as specified in 36.2.5.1.

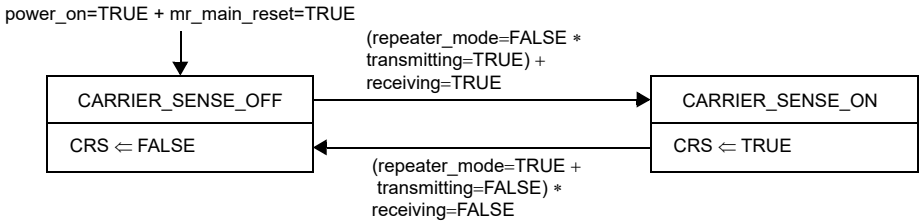


Figure 36–8—Carrier sense state diagram

36.2.5.2.6 Synchronization

The PCS shall implement the Synchronization process as depicted in Figure 36–9 including compliance with the associated state variables as specified in 36.2.5.1. The Synchronization process is responsible for determining whether the underlying receive channel is ready for operation. Failure of the underlying channel typically causes the PMA client to suspend normal actions.

A receiver that is in the LOSS_OF_SYNC state and that has acquired bit synchronization attempts to acquire code-group synchronization via the Synchronization process. Code-group synchronization is acquired by the detection of three ordered sets containing commas in their leftmost bit positions without intervening invalid code-group errors. Upon acquisition of code-group synchronization, the receiver enters the SYNC_ACQUIRED_1 state. Acquisition of synchronization ensures the alignment of multi-code-group ordered sets to even-numbered code-group boundaries.

Once synchronization is acquired, the Synchronization process tests received code-groups in sets of four code-groups and employs multiple sub-states, effecting hysteresis, to move between the SYNC_ACQUIRED_1 and LOSS_OF_SYNC states.

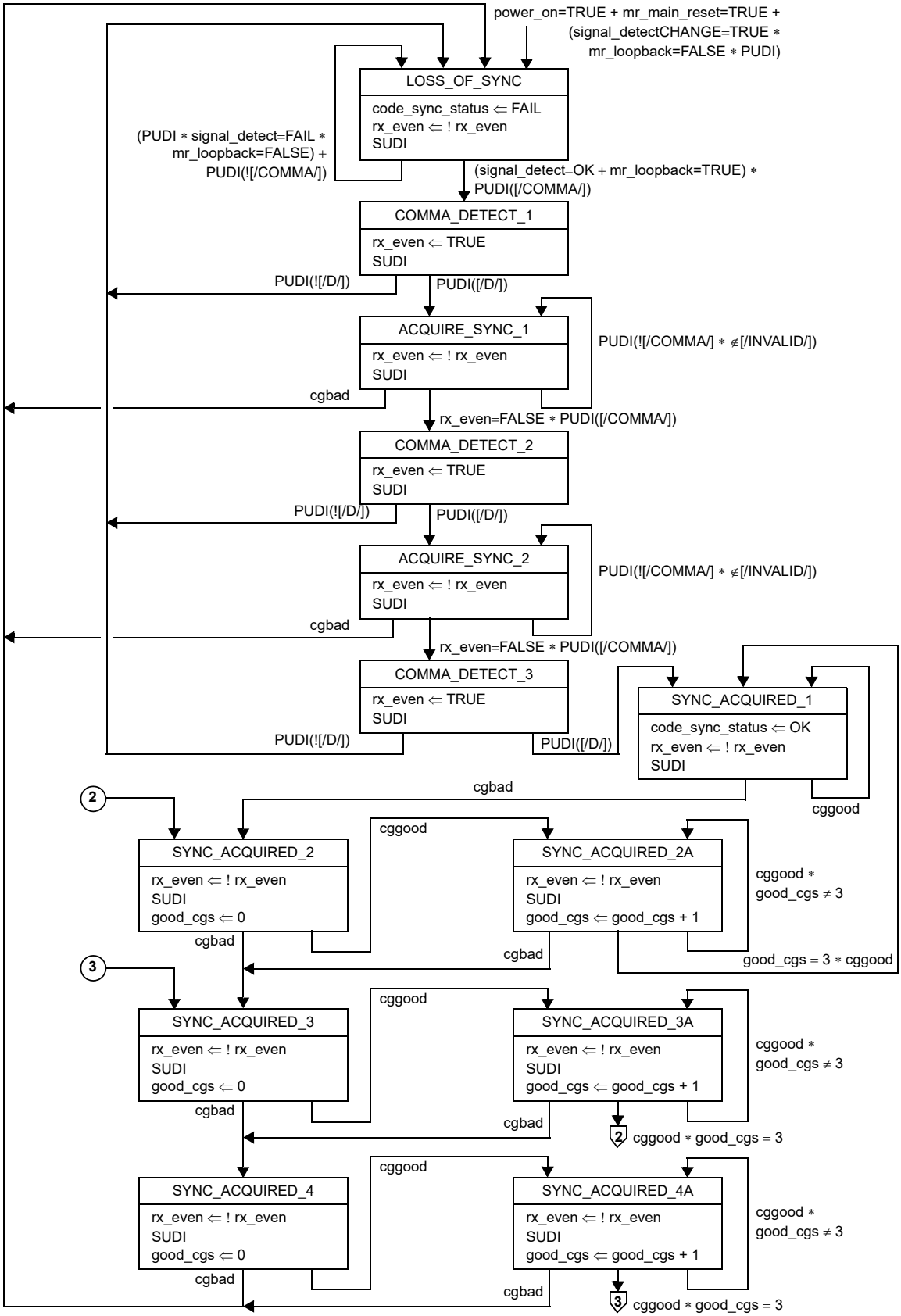


Figure 36–9—Synchronization state diagram

For EEE capability the relationship between sync_status and code_sync_status is given by Figure 36–7c; otherwise sync_status is identical to code_sync_status.

The condition sync_status=FAIL existing for ten ms or more causes the PCS Auto-Negotiation process to begin and the PCS Transmit process to begin transmission of /C/. Upon reception of three matching /C/s from the link partner, the PCS Auto-Negotiation process begins. The internal signal receiving is deasserted in the PCS Receive process LINK_FAILED state when sync_status=FAIL and a code-group is received.

36.2.5.2.7 Auto-Negotiation process

The Auto-Negotiation process shall provide the means to exchange configuration information between two devices that share a link segment and to automatically configure both devices to take maximum advantage of their abilities. When the PCS is used with a PMD other than 1000BASE-KX, see Clause 37 for a description of the Auto-Negotiation process and Config_Reg contents.

Upon successful completion of the Clause 37 Auto-Negotiation process, the xmit flag is set to DATA and normal link operation is enabled. The Clause 37 Auto-Negotiation process utilizes the PCS Transmit and Receive processes to convey Config_Reg contents.

When the PCS is used with a 1000BASE-KX PMD, see Clause 73 for a description of the Auto-Negotiation process. The following requirements apply to a PCS used with a 1000BASE-KX PMD. The PCS shall support the primitive AN_LINK.indication(link_status) (see 73.9). The parameter link_status shall take the value FAIL when sync_status=FAIL and the value OK when sync_status=OK. The primitive shall be generated when the value of link_status changes. If Clause 37 Auto-Negotiation is not present, xmit shall be DATA. If Clause 37 Auto-Negotiation is present the variable mr_an_enable should be false when 1000BASE-KX operation is negotiated through Clause 73 Auto-Negotiation.

36.2.5.2.8 LPI state diagram

A PCS that supports the EEE capability shall implement the LPI transmit process as shown in Figure 36–10. The transmit LPI state diagram controls tx_quiet, which disables the transmitter when true.

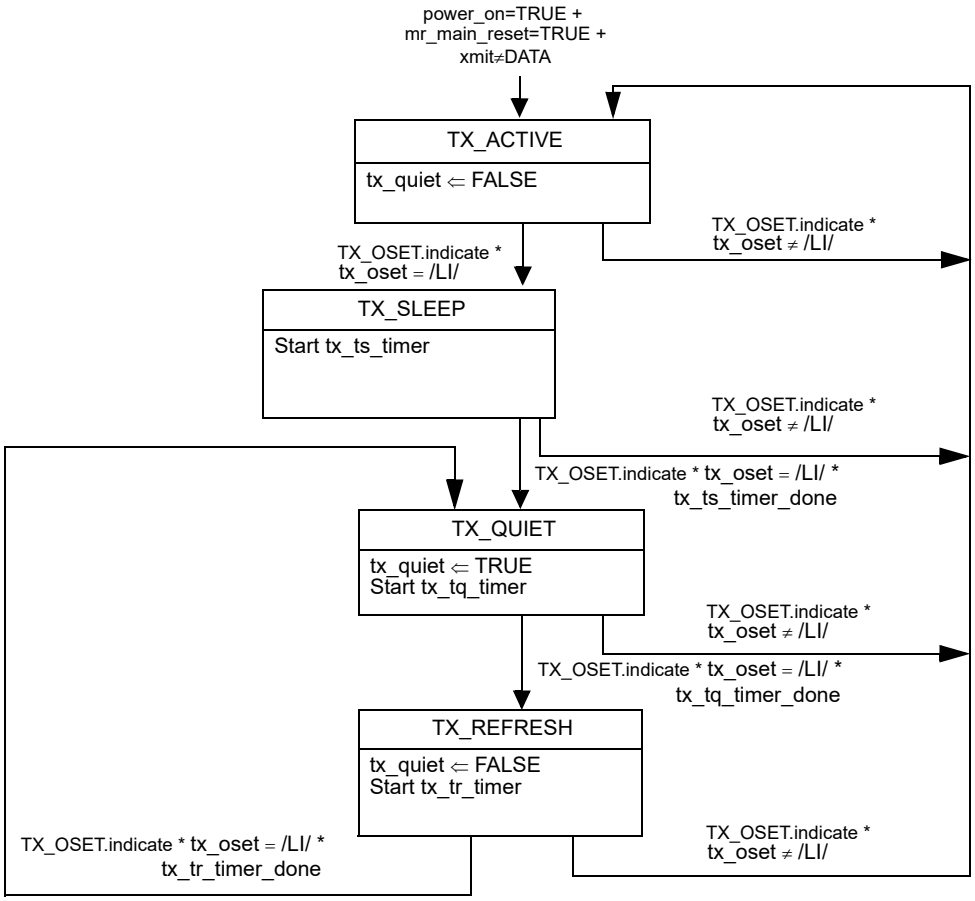


Figure 36–10—LPI Transmit state diagram

The timer values for these state diagrams are shown in Table 36–8 for transmit and Table 36–9 for receive.

Table 36–8—Transmitter LPI timing parameters

Parameter	Description	Min	Max	Units
T_{SL}	Local Sleep Time from entering the TX_SLEEP state to when tx_quiet is set to TRUE	19.9	20.1	µs
T_{QL}	Local Quiet Time from when tx_quiet is set to TRUE to entry into the TX_REFRESH state	2.5	2.6	ms
T_{UL}	Local Refresh Time from entry into the TX_REFRESH state to entry into the TX_QUIET state	19.9	20.1	µs

Table 36–9—Receiver LPI timing parameters

Parameter	Description	Min	Max	Units
T_{QR}	The time the receiver waits for signal detect to be set to OK while in the LP_IDLE_D, LPI_K and RX_QUIET states before asserting a rx_fault	3	4	ms
T_{WR}	Time the receiver waits in the RX_WAKE state before indicating a wake time fault (WTF)		11	µs
T_{WTF}	Wake time fault recovery time		1	ms

36.2.5.2.9 LPI status and management

For EEE capability, the PCS indicates to the management system that LPI is currently active in the receive and transmit directions using the status variables shown in Table 36–10.

Table 36–10—MDIO status indications

MDIO status variable	Register name	Register address	Note
Tx LPI received	PCS status register 1	3.1.11	Latched version of 3.1.9
Rx LPI received	PCS status register 1	3.1.10	Latched version of 3.1.8
Tx LPI indication	PCS status register 1	3.1.9	TRUE when not in state TX_ACTIVE
Rx LPI indication	PCS status register 1	3.1.8	TRUE when not in state RX_ACTIVE

36.3 Physical Medium Attachment (PMA) sublayer

36.3.1 Service Interface

The PMA provides a Service Interface to the PCS. These services are described in an abstract manner and do not imply any particular implementation. The PMA Service Interface supports the exchange of code-groups between PCS entities. The PMA converts code-groups into bits and passes these to the PMD, and vice versa. It also generates an additional status indication for use by its client.

The following primitives are defined:

PMA_UNITDATA.request(tx_code-group<9:0>)
PMA_UNITDATA.indication(rx_code-group<9:0>)

36.3.1.1 PMA_UNITDATA.request

This primitive defines the transfer of data (in the form of code-groups) from the PCS to the PMA. PMA_UNITDATA.request is generated by the PCS Transmit process.

36.3.1.1.1 Semantics of the service primitive

PMA_UNITDATA.request(tx_code-group<9:0>)

The data conveyed by PMA_UNITDATA.request is the tx_code-group<9:0> parameter defined in 36.2.5.1.3.

36.3.1.1.2 When generated

The PCS continuously sends, at a nominal rate of 125 MHz, as governed by GTX_CLK, tx_code-group<9:0> to the PMA.

36.3.1.1.3 Effect of receipt

Upon receipt of this primitive, the PMA generates a series of ten PMD_UNITDATA.request primitives, requesting transmission of the indicated tx_bit to the PMD.

36.3.1.2 PMA_UNITDATA.indication

This primitive defines the transfer of data (in the form of code-groups) from the PMA to the PCS. PMA_UNITDATA.indication is used by the PCS Synchronization process.

36.3.1.2.1 Semantics of the service primitive

PMA_UNITDATA.indication(rx_code-group<9:0>)

The data conveyed by PMA_UNITDATA.indication is the rx_code-group<9:0> parameter defined in 36.2.5.1.3.

36.3.1.2.2 When generated

The PMA continuously sends one rx_code-group<9:0> to the PCS corresponding to the receipt of each code-group aligned set of ten PMD_UNITDATA.indication primitives received from the PMD. The nominal rate of the PMA_UNITDATA.indication primitive is 125 MHz, as governed by the recovered bit clock.

36.3.1.2.3 Effect of receipt

The effect of receipt of this primitive by the client is unspecified by the PMA sublayer.

36.3.2 Functions within the PMA

Figure 36–3 depicts the mapping of the octet-wide data path of the GMII to the ten-bit-wide code-groups of the PMA Service Interface, and on to the serial PMD Service Interface. The PMA comprises the PMA Transmit and PMA Receive processes for 1000BASE-X.

The PMA Transmit process serializes tx_code-groups into tx_bits and passes them to the PMD for transmission on the underlying medium, according to Figure 36–3. Similarly, the PMA Receive process deserializes rx_bits received from the PMD according to Figure 36–3. The PMA continuously conveys ten-bit code-groups to the PCS, independent of code-group alignment. After code-group alignment is achieved, based on comma detection, the PCS converts code-groups into GMII data octets, according to 36.2.5.2.2.

The proper alignment of a comma used for code-group synchronization is depicted in Figure 36–3.

NOTE—Strict adherence to manufacturer-supplied guidelines for the operation and use of PMA serializer components is required to meet the jitter specifications of Clause 38 and Clause 39. The supplied guidelines should address the quality of power supply filtering associated with the transmit clock generator, and also the purity of the reference clock fed to the transmit clock generator.

36.3.2.1 Data delay

The PMA maps a nonaligned one-bit data path from the PMD to an aligned, ten-bit-wide data path to the PCS, on the receive side. Logically, received bits must be buffered to facilitate proper code-group alignment. These functions necessitate an internal PMA delay of at least ten bit times. In practice, code-group alignment may necessitate even longer delays of the incoming rx_bit stream.

36.3.2.2 PMA transmit function

The PMA Transmit function passes data unaltered (except for serializing) from the PCS directly to the PMD. Upon receipt of a PMA_UNITDATA.request primitive, the PMA Transmit function shall serialize the ten bits of the tx_code-group<9:0> parameter and transmit them to the PMD in the form of ten successive PMD_UNITDATA.request primitives, with tx_code-group<0> transmitted first, and tx_code-group<9> transmitted last.

36.3.2.3 PMA receive function

The PMA Receive function passes data unaltered (except for deserializing and possible code-group slipping upon code-group alignment) from the PMD directly to the PCS. Upon receipt of ten successive PMD_UNITDATA.indication primitives, the PMA shall assemble the ten received rx_bits into a single ten-bit value and pass that value to the PCS as the rx_code-group<9:0> parameter of the primitive PMA_UNITDATA.indication, with the first received bit installed in rx_code-group<0> and the last received bit installed in rx_code-group<9>. An exception to this operation is specified in 36.3.2.4.

36.3.2.4 Code-group alignment

In the event the PMA sublayer detects a comma+ within the incoming rx_bit stream, it may realign its current code-group boundary, if necessary, to that of the received comma+ as shown in Figure 36–3. This process is referred to in this document as code-group alignment. The code-group alignment function shall be operational when the EN_CDET signal is active (see 36.3.3.1). During the code-group alignment process, the PMA sublayer may delete or modify up to four, but shall delete or modify no more than four, ten-bit code-groups in order to align the correct receive clock and code-group containing the comma+. This process is referred to as code-group slipping.

In addition, the PMA sublayer is permitted to realign the current code-group boundary upon receipt of a comma-pattern.

36.3.3 A physical instantiation of the PMA Service Interface

The ten-bit interface (TBI) is defined to provide compatibility among devices designed by different manufacturers. There is no requirement for a compliant device to implement or expose the TBI. A TBI implementation shall behave as described in 36.3.3 through 36.3.6.

Figure 36–11 illustrates the TBI functions and interfaces.

As depicted in Figure 36–11, the TBI connects the PCS and PMD sublayers. It is equipped for full duplex transmission of code-groups at 125 MHz. The PCS provides code-groups on tx_code-group<9:0> to the PMA transmit function, which latches the data on the rising edge of the 125 MHz PMA_TX_CLK. An internal Clock Multiplier Unit uses PMA_TX_CLK to generate the internal 1250 MHz bit clock that is used to serialize the latched data out of the PMA outputs, if EWRAP is Low, or internally loop it back to the Receive function input, if EWRAP is High.

The PMA Receive function accepts 1250 Mb/s serial data from either the PMD, if EWRAP is Low, or the PMA transmit function, if EWRAP is High, and extracts a bit clock and recovered data from the serial inputs

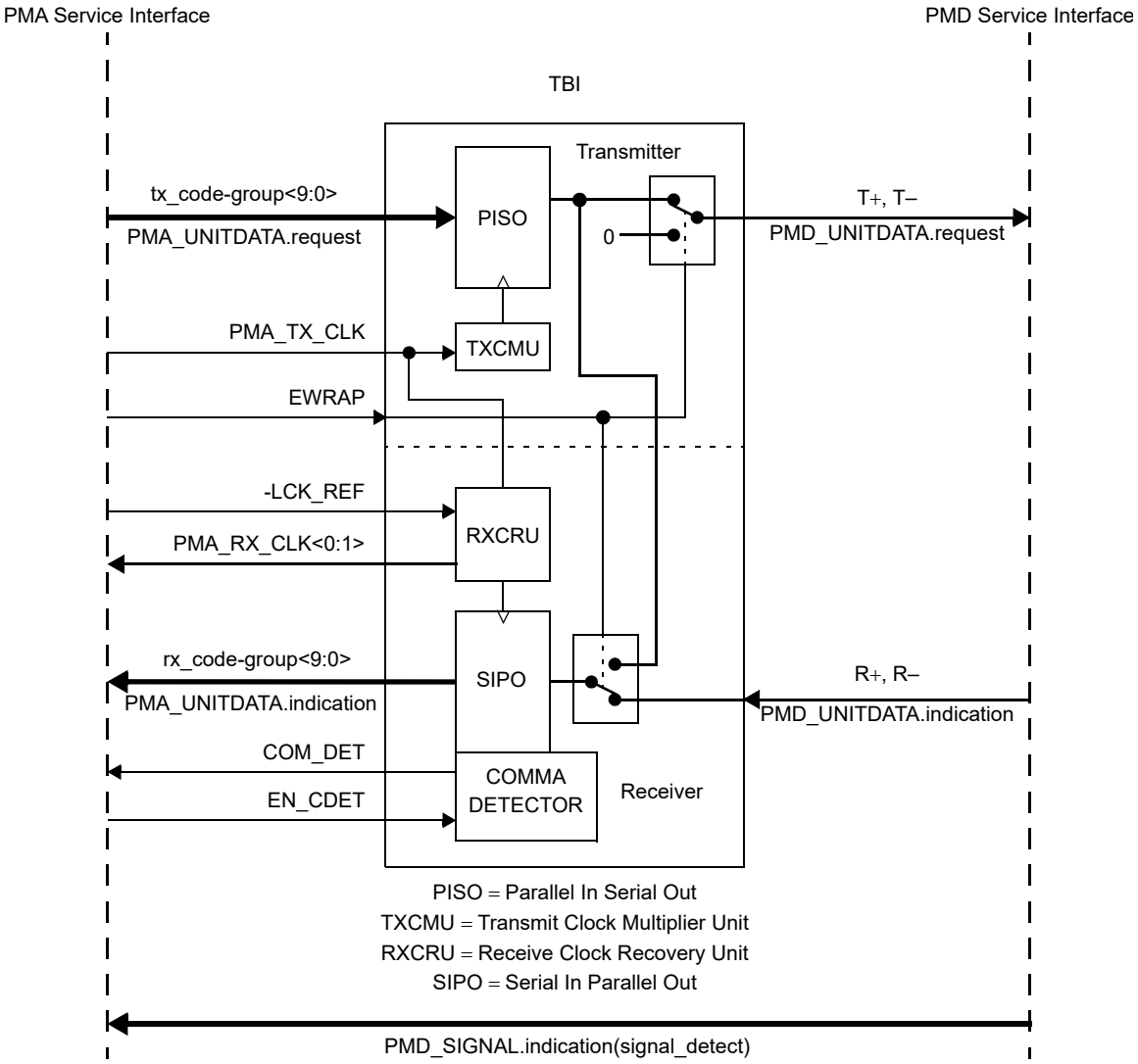


Figure 36–11—TBI reference diagram

in a clock recovery unit. The recovered data is deserialized and conveyed to the PCS on rx_code-group<9:0>. Two recovered clocks, PMA_RX_CLK<0> and PMA_RX_CLK<1>, which are at 1/20th the baud (62.5 MHz), and 180° out-of-phase with one another, are used by the PMA to latch the received 10-bit code-groups. Even and odd-numbered code-groups are latched on successive rising edges of PMA_RX_CLK<1> and PMA_RX_CLK<0>, respectively.

Code-group alignment occurs in the PMA Receive function, if enabled by EN_CDET, when a comma pattern occurs in the PHY bit stream. Upon recognition of the comma pattern, the PMA Receive function outputs the ten-bit code-group containing the comma on rx_code-group<9:0> with the alignment specified in Figure 36–3, and clocked on the rising edge of PMA_RX_CLK<1>.

This TBI provides a Lock_to_Reference_Clock (LCK_REF) input, which may be used to lock the clock recovery unit to PMA_TX_CLK rather than incoming serial data. In the absence of serial data or invalid serial data, the PMA Receive function passes many 8B/10B invalid code-groups across to the PCS. A circuit may be constructed to detect those errors and, using LCK_REF, re-center the receiver clock recovery unit to PMA_TX_CLK in preparation for reacquiring lock on the incoming PHY bit stream.

36.3.3.1 Required signals

In the event this TBI is made accessible, the signals listed in Table 36–11 are provided, with the meanings described elsewhere in this section. Note that not all of these signals are used by the PCS.

Table 36–11—TBI required signals

Symbol	Signal Name	Signal Type	Active Level
tx_code-group<9:0>	Transmit Data	Input	H
PMA_TX_CLK	Transmit Clock	Input	↑
EWRAP	Enable Wrap	Input	H
rx_code-group<9:0>	Receive Data	Output	H
PMA_RX_CLK<0>	Receive Clock 0	Output	↑
PMA_RX_CLK<1>	Receive Clock 1	Output	↑
COM_DET	Comma Detect	Output	H
-LCK_REF	Lock to Reference	Input	L
EN_CDET	Enable Comma Detect	Input	H

tx_code-group<9:0>
: The 10-bit parallel transmit data presented to the PMA for serialization and transmission onto the media. The order of transmission is tx_bit<0> first, followed by tx_bit<1> through tx_bit<9>.

PMA_TX_CLK
: The 125 MHz transmit code-group clock. This code-group clock is used to latch data into the PMA for transmission. PMA_TX_CLK is also used by the transmitter clock multiplier unit to generate the 1250 MHz bit rate clock. PMA_TX_CLK is also used by the receiver when -LCK_REF is active. PMA_TX_CLK has a ±100 ppm tolerance. PMA_TX_CLK is derived from GMII GTX_CLK.

EWRAP
: EWRAP enables the TBI to electrically loop transmit data to the receiver. The serial outputs on the transmitter are held in a static state during EWRAP operation. EWRAP may optionally be tied low (function disabled).

rx_code-group<9:0>
: Presents the 10-bit parallel receive code-group data to the PCS for further processing. When code-groups are properly aligned, any received code-group containing a comma is clocked by PMA_RX_CLK<1>.

PMA_RX_CLK<0>
: The 62.5 MHz receive clock that the protocol device uses to latch odd-numbered code-groups in the received PHY bit stream. This clock may be stretched during code-group alignment, and is not shortened.

PMA_RX_CLK<1>
: The 62.5 MHz receive clock that the protocol device uses to latch even-numbered code-groups in the received PHY bit stream. PMA_RX_CLK<1> is 180° out-of-phase with PMA_RX_CLK<0>. This clock may be stretched during code-group alignment, and is not shortened.

COM_DET

An indication that the code-group associated with the current PMA_RX_CLK<1> contains a valid comma. When EN_CDET is asserted, the TBI is required to detect and code-group-align to the comma+ bit sequence. Optionally, the TBI may also detect and code-group-align to the comma- bit sequence. The TBI provides this signal as an output, but it may not be used by the PCS.

-LCK_REF

Causes the TBI clock recovery unit to lock to PMA_TX_CLK. The TBI attains frequency lock within 500 ms. This function is not used by the PCS.

NOTE—Implementers may find it necessary to use this signal in order to meet the clock recovery requirements of the PMA sublayer.

EN_CDET

Enables the TBI to perform the code-group alignment function on a comma (see 36.2.4.9, 36.3.2.4). When EN_CDET is asserted the code-group alignment function is operational. This signal is optionally generated by the PMA client. The PMA sublayer may leave this function always enabled.

36.3.3.2 Summary of control signal usage

Table 36–12 lists all possible combinations of control signals on this TBI, including the valid combinations as well as the undefined combinations.

Table 36–12—TBI combinations of control signals

EWRAP	-LCK_REF	EN_CDET	Interpretation
L	L	L	Undefined
L	L	H	Lock receiver clock recovery unit to PMA_TX_CLK
L	H	L	Normal operation; COM_DET disabled
L	H	H	Normal operation; COM_DET enabled
H	L	L	Undefined
H	L	H	Undefined
H	H	L	Loop transmit data to receiver; COM_DET disabled
H	H	H	Loop transmit data to receiver; COM_DET enabled

36.3.4 General electrical characteristics of the TBI

In the event this TBI is made accessible, the following subclauses specify the general electrical characteristics of the TBI.

36.3.4.1 DC characteristics

Table 36–13 documents the required dc parametric attributes required of all inputs to the TBI and the dc parametric attributes associated with the outputs of the TBI. The inputs levels to the TBI may be greater than the power supply level (i.e., 5 V output driving V_{OH} into a 3.3 V input), tolerance to mismatched input levels is optional. TBI devices not tolerant of mismatched inputs levels that meet Table 36–13 requirements are still regarded as compliant.

Table 36–13—DC specifications

Symbol	Parameter	Conditions		Min	Typ	Max	Units
V_{OH}	Output High Voltage	I_{OH} = –400 µA	V_{CC} = Min	2.2	3.0	V_{CC}	V
V_{OL}	Output Low Voltage	I_{OL} = 1 mA	V_{CC} = Min	GND	0.25	0.6	V
V_{IH}	Input High Voltage			2.0	—	V_{CC}[a] + 10%	V
V_{IL}	Input Low Voltage			GND	—	0.8	V
I_{IH}	Input High Current	V_{CC} = Max	V_{IN} = 2.4 V	—	—	40	µA
I_{IL}	Input Low Current	V_{CC} = Max	V_{IN} = 0.4 V	—	—	600	µA
C_{IN}	Input Capacitance			—	—	4.0	pf
t_R	Clock Rise Time	0.8 V to 2.0 V		0.7	—	2.4	ns
t_F	Clock Fall Time	2.0 V to 0.8 V		0.7	—	2.4	ns
t_R	Data Rise Time	0.8 V to 2.0 V		0.7	—	—	ns
t_F	Data Fall Time	2.0 V to 0.8 V		0.7	—	—	ns

[a]Refers to the driving device power supply.

36.3.4.2 Valid signal levels

All ac measurements are made from the 1.4 V level of the clock to the valid input or output data levels as shown in Figure 36–12.

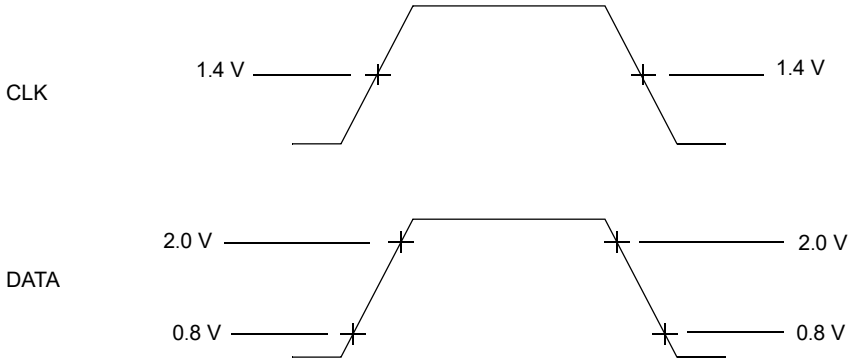


Figure 36–12—Input/output valid level for ac measurements

36.3.4.3 Rise and fall time definition

The rise and fall time definition for PMA_TX-CLK, PMA_RX_CLK<0>, PMA_RX_CLK<1>, and DATA is shown in Figure 36–13.

36.3.4.4 Output load

All ac measurements are assumed to have the output load of 10 pF.

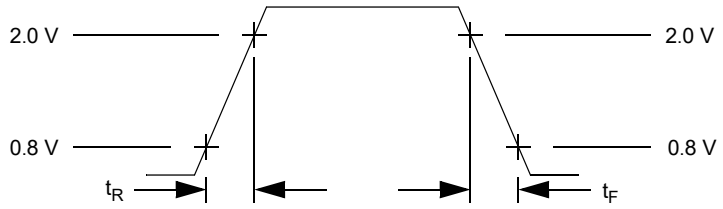


Figure 36–13—Rise and fall time definition

36.3.5 TBI transmit interface electrical characteristics

In the event this TBI is made accessible, the electrical characteristics of the TBI transmit interface are specified in the following subclauses.

36.3.5.1 Transmit data (tx_code-group<9:0>)

The tx_code-group<9:0> signals carry data from the PCS to PMA to be serialized to the PMD in accordance with the transmission order shown in Figure 36–3. All tx_code-group<9:0> data conforms to valid code-groups.

36.3.5.2 TBI transmit interface timing

The TBI transmit interface timings in Table 36–14 defines the TBI input. All transitions in Figure 36–14 are specified from the PMA_TX_CLK reference level (1.4 V), to valid input signal levels.

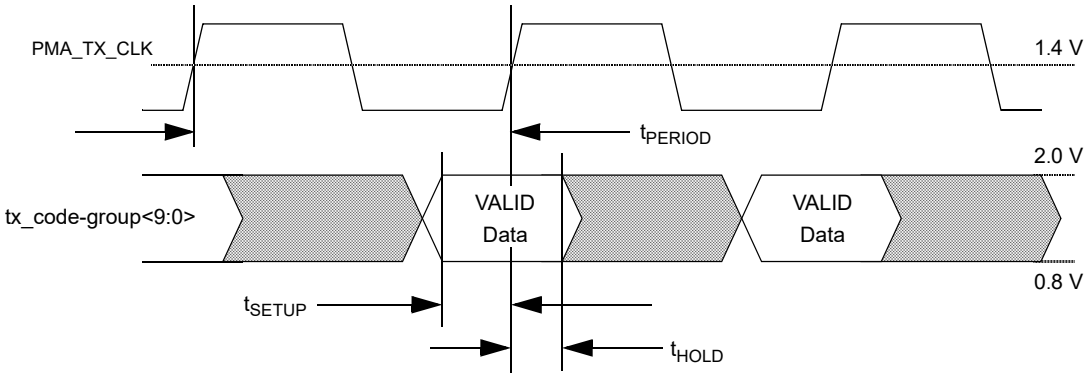


Figure 36–14—TBI transmit interface timing diagram

Table 36–14—Transmit AC specification

Parameter	Description	Min	Typ	Max	Units
t_{PERIOD}	PMA_TX_CLK Period[a]	—	8	—	ns
t_{SETUP}	Data Setup to ↑ PMA_TX_CLK	2.0	—	—	ns
t_{HOLD}	Data Hold from ↑ PMA_TX_CLK	1.0	—	—	ns
t_{DUTY}	PMA_TX_CLK Duty Cycle	40		60	%

[a] ±100 ppm tolerance on PMA_TX_CLK frequency.

36.3.6 TBI receive interface electrical characteristics

In the event this TBI is made accessible, the electrical characteristics of the TBI receive interface are specified in the following subclauses.

The TBI receive interface timings in Table 36–15 define the TBI output. All transitions in Figure 36–15 are specified from the Receive Clock reference level (1.4 V) to valid output signal levels.

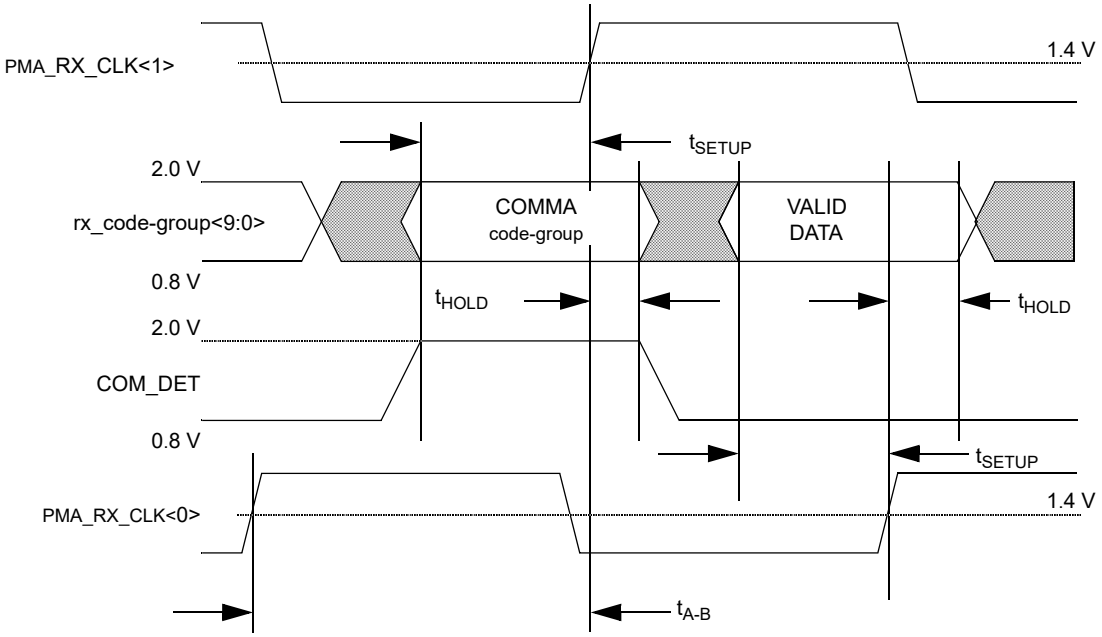


Figure 36–15—TBI receive interface timing diagram

36.3.6.1 Receive data (rx_code-group<9:0>)

The 10 receive data signals rx_code-group<9:0> carry parallel data from the PMA sublayer to the PCS sublayer during the rising edge of the receive clock (i.e., PMA_RX_CLK<1> transitions from Low to High). When properly locked and aligned, data transferred across this interface conforms to valid code-groups.

36.3.6.2 Receive clock (PMA_RX_CLK<0>, PMA_RX_CLK<1>)

The receive clocks supplied to the PCS and GMII are derived from the recovered bit clock. PMA_RX_CLK<0> is 180° out-of-phase with PMA_RX_CLK<1>.

Table 36–15 specifies a receive clock drift (t_{DRIFT}), which is applicable under all input conditions to the receiver (including invalid or absent input signals). However, the restriction does not apply when the receiver is realigning to a new code-group boundary and the receive clocks are being stretched to a new code-group boundary to avoid short pulses. During the code-group alignment process the receive clocks may slow a fixed amount, depending on the bit offset of the new comma and then return to the nominal frequency.

36.3.7 Loopback mode

Loopback mode shall be provided, as specified in this subclause, by the transmitter and receiver of a device as a test function to the device. When Loopback mode is selected, transmission requests passed to the

Table 36–15—Receive bus AC specification

Parameter	Description	Min	Typ	Max	Units
t_{FREQ}	PMA_RX_CLK Frequency	—	62.5	—	MHz
t_{DRIFT}	PMA_RX_CLK Drift Rate[a]	0.2	—	—	µs/MHz
t_{SETUP}	Data Setup Before ↑PMA_RX_CLK	2.5	—	—	ns
t_{HOLD}	Data Hold After ↑PMA_RX_CLK	1.5	—	—	ns
t_{DUTY}	PMA_RX_CLK Duty Cycle	40	—	60	%
t_{A-B}	PMA_RX_CLK Skew	7.5	—	8.5	ns

[a]t_{DRIFT} is the (minimum) time for PMA_RX_CLK to drift from 63.5 MHz to 64.5 MHz or 60 MHz to 59 MHz from the PMA_RX_CLK lock value. It is applicable under all input signal conditions (except where noted in 36.3.2.4), including invalid or absent input signals, provided that the receiver clock recovery unit was previously locked to PMA_TX_CLK or to a valid input signal.

transmitter are shunted directly to the receiver, overriding any signal detected by the receiver on its attached link. A device is explicitly placed in Loopback mode (i.e., Loopback mode is not the normal mode of operation of a device). The method of implementing Loopback mode is not defined by this standard.

NOTE—Loopback mode may be implemented either in the parallel or the serial circuitry of a device.

36.3.7.1 Receiver considerations

A receiver may be placed in Loopback mode. Entry into or exit from Loopback mode may result in a temporary loss of synchronization.

36.3.7.2 Transmitter considerations

A transmitter may be placed in Loopback mode. The external behavior of a transmitter (i.e., the activity of a transmitter with respect to its attached link) in Loopback mode is specified in 22.2.4.1.2.

36.3.8 Test functions

A limited set of test functions may be provided as an implementation option for testing of the transmitter function.

Some test functions that are not defined by this standard may be provided by certain implementations. Compliance with the standard is not affected by the provision or exclusion of such functions by an implementation. Random jitter test patterns for 1000BASE-X are specified in Annex 36A.

A typical test function is the ability to transmit invalid code-groups within an otherwise valid PHY bit stream. Certain invalid PHY bit streams may cause a receiver to lose word and/or bit synchronization. See ANSI INCITS 230-1994 (FC-PH), subclause 5.4, for a more detailed discussion of receiver and transmitter behavior under various test conditions.

36.4 Compatibility considerations

There is no requirement for a compliant device to implement or expose any of the interfaces specified for the PCS or PMA. Implementations of a GMII shall comply with the requirements as specified in Clause 35. Implementations of a TBI shall comply with the requirements as specified in 36.3.3.

36.5 Delay constraints

In half duplex mode, proper operation of a CSMA/CD LAN demands that there be an upper bound on the propagation delays through the network. This implies that MAC, PHY, and repeater implementers must conform to certain delay minima and maxima, and that network planners and administrators conform to constraints regarding the cable topology and concatenation of devices. MAC constraints are contained in 35.2.4 and Table 35–5. Topological constraints are contained in Clause 42.

In full duplex mode, predictable operation of the MAC Control PAUSE operation (Clause 31, Annex 31B) also demands that there be an upper bound on the propagation delays through the network. This implies that MAC, MAC Control sublayer, and PHY implementers must conform to certain delay maxima, and that network planners and administrators conform to constraints regarding the cable topology and concatenation of devices.

The reference point for all MDI measurements is the 50% point of the mid-cell transition corresponding to the reference bit, as measured at the MDI.

36.5.1 MDI to GMII delay constraints

Every 1000BASE-X PHY associated with a GMII shall comply with the bit time delay constraints specified in Table 36–16 for half duplex operation and Table 36–17 for full duplex operation. These figures apply for all 1000BASE-X PMDs. For any given implementation, the assertion and deassertion delays on CRS shall be equal.

Table 36–16—MDI to GMII delay constraints (half duplex mode)

Sublayer measurement points	Event	Min (bit time)	Max (bit time)	Input timing reference	Output timing reference
GMII ⇔ MDI	TX_EN=1 sampled to MDI output	—	136	PMA_TX_CLK rising	1st bit of /S/
	MDI input to CRS assert	—	192	1st bit of /S/	
	MDI input to CRS deassert	—	192	1st bit of /K28.5/	
	MDI input to COL assert	—	192	1st bit of /S/	
	MDI input to COL deassert	—	192	1st bit of /K28.5/	
	TX_EN=1 sampled to CRS assert	—	16	PMA_TX_CLK rising	
	TX_EN=0 sampled to CRS deassert	—	16	PMA_TX_CLK rising	

Table 36–17—MDI to GMII delay constraints (full duplex mode)

Sublayer measurement points	Event	Min (bit time)	Max (bit time)	Input timing reference	Output timing reference
GMII ⇔ MDI	TX_EN=1 sampled to MDI output	—	136	PMA_TX_CLK rising	1st bit of /S/
	MDI input to RX_DV deassert	—	192	1st bit of /T/	RX_CLK rising

36.5.2 DTE delay constraints (half duplex mode)

Every DTE with a 1000BASE-X PHY shall comply with the bit time delay constraints specified in Table 36–18 for half duplex operation. These figures apply for all 1000BASE-X PMDs.

Table 36–18—DTE delay constraints (half duplex mode)

Sublayer measurement points	Event	Min (bit time)	Max (bit time)	Input timing reference	Output timing reference
MAC ⇔ MDI	MAC transmit start to MDI output	—	184		1st bit of /S/
	MDI input to MDI output (worst-case nondeferred transmit)	—	440	1st bit of /S/	1st bit of /S/
	MDI input to collision detect	—	240	1st bit of /S/	
	MDI input to MDI output = Jam (worst-case collision response)	—	440	1st bit of /S/	1st bit of jam

36.5.3 Carrier deassertion/assertion constraint (half duplex mode)

To ensure fair access to the network, each DTE operating in half duplex mode shall, additionally, satisfy the following:

(MAX MDI to MAC Carrier Deassert Detect) – (MIN MDI to MAC Carrier Assert Detect) < 16 bits

36.6 Environmental specifications

All equipment subject to this clause shall conform to the requirements of 14.7 and applicable sections of ISO/IEC 11801:1995.

36.7 Protocol implementation conformance statement (PICS) proforma for Clause 36, Physical Coding Sublayer (PCS) and Physical Medium Attachment (PMA) sublayer, type 1000BASE-X[3]

36.7.1 Introduction

The supplier of a protocol implementation that is claimed to conform to Clause 36, Physical Coding Sublayer (PCS) and Physical Medium Attachment (PMA) sublayer, type 1000BASE-X, shall complete the following protocol implementation conformance statement (PICS) proforma. A detailed description of the symbols used in the PICS proforma, along with instructions for completing the PICS proforma, can be found in Clause 21.

36.7.2 Identification

36.7.2.1 Implementation identification

Supplier (Note 1)	
Contact point for inquiries about the PICS (Note 1)	
Implementation Name(s) and Version(s) (Notes 1 and 3)	
Other information necessary for full identification—e.g., name(s) and version(s) for machines and/or operating systems; System Name(s) (Note 2)	
NOTE 1—Required for all implementations. NOTE 2—May be completed as appropriate in meeting the requirements for the identification. NOTE 3—The terms Name and Version should be interpreted appropriately to correspond with a supplier's terminology (e.g., Type, Series, Model).	

36.7.2.2 Protocol summary

Identification of protocol standard	IEEE Std 802.3-2018, Clause 36, Physical Coding Sublayer (PCS) and Physical Medium Attachment (PMA) sublayer, type 1000BASE-X
Identification of amendments and corrigenda to this PICS proforma that have been completed as part of this PICS	
Have any Exception items been required? No [ ] Yes [ ] (See Clause 21; the answer Yes means that the implementation does not conform to IEEE Std 802.3-2018.)	

Date of Statement	

[3]*Copyright release for PICS proformas:* Users of this standard may freely reproduce the PICS proforma in this subclause so that it can be used for its intended purpose and may further publish the completed PICS.

36.7.3 Major capabilities/options

Item	Feature	Subclause	Value/Comment	Status	Support
*PMA	Ten-bit interface (TBI)	36.4		O	Yes [ ] No [ ]
*GMII	PHY associated with GMII	36.4		O	Yes [ ] No [ ]
*DTE	DTE with PHY not associated with GMII	36.5.2		O	Yes [ ] No [ ]
*FDX	PHY supports full duplex mode	36.5		O	Yes [ ] No [ ]
*HDX	PHY supports half duplex mode	36.5		O	Yes [ ] No [ ]
*LPI	Implementation of LPI	36.2.4.13		O	Yes [ ] No [ ]
NOTE—The following abbreviations are used: *HDGM: HDX and GMII *FDGM: FDX and GMII *HDTE: HDX and DTE					

36.7.4 PICS proforma tables for the PCS and PMA sublayer, type 1000BASE-X

36.7.4.1 Compatibility considerations

Item	Feature	Subclause	Value/Comment	Status	Support
CC1	Test functions Annex 36A support	36.3.8		O	Yes [ ] No [ ]
CC2	Environmental specifications	36.6		M	Yes [ ]

36.7.4.2 Code-group functions

Item	Feature	Subclause	Value/Comment	Status	Support
CG1	Transmitter initial running disparity	36.2.4.4	Transmitter initial running disparity assumes negative value	M	Yes [ ]
CG2	Transmitter running disparity calculation	36.2.4.4	Running disparity is calculated after each code-group transmitted	M	Yes [ ]
CG3	Validating received code-groups	36.2.4.6		M	Yes [ ]
CG4	Running disparity rules	36.2.4.4	Running disparity is calculated after each code-group reception	M	Yes [ ]
CG5	Transmitted code-group is chosen from the corresponding running disparity	36.2.4.5		M	Yes [ ]

36.7.4.3 State diagrams

Item	Feature	Subclause	Value/Comment	Status	Support
SD1	Transmit ordered set	36.2.5.2.1	Meets the requirements of Figure 36–5	M	Yes [ ]
SD2	Transmit code-group	36.2.5.2.1	Meets the requirements of Figure 36–6	M	Yes [ ]
SD3	Receive	36.2.5.2.2	Meets the requirements of Figures 36–7a and 36–7b	M	Yes [ ]
SD4	Carrier sense	36.2.5.2.5	Meets the requirements of Figure 36–8	M	Yes [ ]
SD5	Synchronization	36.2.5.2.6	Meets the requirements of Figure 36–9	M	Yes [ ]
SD6	Auto-Negotiation	36.2.5.2.7	Described in Clause 37	M	Yes [ ]
SD7*	Support for use with a 1000BASE-KX PMD	36.2.5.2.7	AN technology dependent interface described in Clause 73	O	Yes [ ]
SD8	AN_LINK.indication primitive	36.2.5.2.7	Support of the primitive AN_LINK.indication(link_status), when the PCS is used with 1000BASE-KX PMD	SD7:M	Yes [ ]
SD9	link_status parameter	36.2.5.2.7	Takes the value OK or FAIL, as described in 36.2.5.2.7	SD7:M	Yes [ ]
SD10	Generation of AN_LINK.indication primitive	36.2.5.2.7	Generated when the value of link_status changes	SD7:M	Yes [ ]
SD11	Value of xmit, when the PCS is used with 1000BASE-KX PMD	36.2.5.2.7	The value of xmit is DATA, when Clause 37 Auto-Negotiation is not present as described in 36.2.5.2.7	SD7:M	Yes [ ]

36.7.4.4 PMA functions

Item	Feature	Subclause	Value/Comment	Status	Support
PMA1	Transmit function	36.3.2.2		M	Yes [ ]
PMA2	Receive function	36.3.2.3		M	Yes [ ]
PMA3	Code-group alignment	36.3.2.4	When EN_CDET is active	M	Yes [ ]
PMA4	Loopback mode	36.3.7		M	Yes [ ]

36.7.4.5 PMA transmit function

Item	Feature	Subclause	Value/Comment	Status	Support
PMT1	cg_timer expiration	36.2.5.1.7	See 35.5.2.3	GMII:M	Yes [ ] N/A [ ]
PMT2	cg_timer expiration	36.2.5.1.7	8 ns ± 0.01%	!GMII: M	Yes [ ] N/A [ ]

36.7.4.6 PMA code-group alignment function

Item	Feature	Subclause	Value/Comment	Status	Support
CDT1	Code-group alignment to comma-	36.3.2.4		O	Yes [ ] N/A [ ]
CDT2	Code-group slipping limit	36.3.2.4	Deletion or modification of no more than four code-groups	M	Yes [ ]
CDT3	Code-group alignment to comma+	36.3.2.4		O	Yes [ ] N/A [ ]

36.7.4.7 TBI

Item	Feature	Subclause	Value/Comment	Status	Support
TBI1	TBI requirements	36.3.3		PMA:M	Yes [ ] N/A [ ]

36.7.4.8 Delay constraints

Item	Feature	Subclause	Value/Comment	Status	Support
TIM1	Equal carrier deassertion and assertion delay on CRS	36.5.1		HDGM:M	Yes [ ] N/A [ ]
TIM2	MDI to GMII delay constraints for half duplex	36.5.1	Table 36–16	HDGM:M	Yes [ ] N/A [ ]
TIM3	MDI to GMII delay constraints for full duplex	36.5.1	Table 36–17	FDGM:M	Yes [ ] N/A [ ]
TIM4	DTE delay constraints for half duplex	36.5.2	Table 36–18	HDTE:M	Yes [ ] N/A [ ]
TIM5	Carrier deassertion/assertion constraints	36.5.3		HDTE:M	Yes [ ] N/A [ ]

36.7.4.9 LPI functions

Item	Feature	Subclause	Value/Comment	Status	Support
LP-01	Transmit ordered set state diagram	36.2.5.2.1	Support additions to Figure 36–5 for LPI operation	LPI:M	Yes [ ] No [ ]
LP-02	Receive state diagram	36.2.5.2.2	Support additions to Figure 36–7a, Figure 36–7b for LPI operation	LPI:M	Yes [ ] No [ ]
LP-03	LPI transmit state diagram	36.2.5.2.8	Meets the requirements of Figure 36–10	LPI:M	Yes [ ] No [ ]
LP-04	LPI receive state diagram	36.2.5.2.8	Meets the requirements of Figure 36–7c	LPI:M	Yes [ ] No [ ]
LP-03	LPI transmit timing	36.2.5.2.8	Meets the requirements of Table 36–8	LPI:M	Yes [ ] No [ ]
LP-04	LPI receive timing	36.2.5.2.8	Meets the requirements of Table 36–9	LPI:M	Yes [ ] No [ ]