97. Physical Coding Sublayer (PCS), Physical Medium Attachment (PMA) sublayer, Physical Medium Dependent (PMD) sublayer, and baseband medium, type 1000BASE-T1

97.1 Overview

This clause defines the type 1000BASE-T1 Physical Coding Sublayer (PCS), type 1000BASE-T1 Physical Medium Attachment (PMA) sublayer, and type 1000BASE-T1 Physical Medium Dependent (PMD). Together, the PCS, PMA, and PLD sublayers comprise a 1000BASE-T1 Physical Layer (PHY). Provided in this clause are fully functional and electrical specifications for the type 1000BASE-T1 PCS, PMA, and PMD. This clause also specifies the critical parameters of the baseband medium used with 1000BASE-T1 PHY.

The 1000BASE-T1 PHY is one of the Gigabit Ethernet family of high-speed full-duplex network specifications, defining the automotive link capable of operating at 1000 Mb/s and intended to be operated over a single pair of balanced copper cabling, referred to as an automotive link segment (Type A) or additional link segment (Type B), defined in 97.5.4. The cabling supporting the operation of the 1000BASE-T1 PHY is defined in terms of performance requirements between the attachment points (Medium Dependent Interface (MDI)), allowing implementers to provide their own cabling to operate the 1000BASE-T1 PHY as long as the normative requirements included in this Clause are met.

This clause also specifies 1000BASE-T1 Low Power Idle (LPI) as part of Energy-Efficient Ethernet (EEE). This allows the PHY to enter a low power mode of operation during periods of low link utilization as described in Clause 78.

97.1.1 Objectives

TBD

97.1.2 Relationship of 1000BASE-T1 to other standards

TBD

97.1.3 Operation of 1000BASE-T1

TBD

97.1.3.1 Physical Coding Sublayer (PCS)

TBD

97.1.3.2 Physical Medium Attachment (PMA) sublayer

TBD

97.1.3.3 Physical Medium Dependent (PMD) sublayer

TBD

97.1.3.4 EEE capability

A 1000BASE-T1 PHY may optionally support the EEE capability, as described in 78.3. The EEE capability is a mechanism by which 1000BASE-T1 PHYs are able to reduce power consumption during periods of low

link utilization. PHYs can enter this mode of operation after completing training. Each direction of the full duplex link is able to enter and exit the LPI mode independently, supporting symmetric and asymmetric LPI operation. This allows power savings when only one side of the full duplex link is in a period of low utilization. The transition to or from LPI mode shall cause no data frames be lost or corrupted.

In the transmit direction the transition to the LPI transmit mode begins when the PCS transmit function detects an "Assert Low Power Idle" condition on the GMII in the last 80B/81B block of a frame. At the next RS frame the PCS transmits a sleep signal composed of an entire RS frame containing only LP_IDLE. The sleep signal indicates to the link partner that the transmit function of the PHY is entering the LPI transmit mode. Immediately after the transmission of the sleep frame, the transmit function of the local PHY enters the LPI transmit mode. While the transmit function is in the LPI mode the PHY may disable data path and control logic to save additional power. Periodically the transmit function of the local PHY transmits refresh frames that may be used by the link partner to update adaptive filters and timing circuits in order to maintain link integrity. LPI mode may begin with quiet signaling, a full refresh period, or a wake frame. The quiet-refresh cycle continues until the PCS function detects a condition that is not Assert Low Power Idle on the GMII. This condition signals to the PHY that the LPI transmit mode should end. At the next RS frame the PCS transmits a wake frame composed of an entire RS frame containing only Idle. On the next RS frame normal operational mode shall resume.

Support for EEE capability is advertised during Auto-Negotiation. Transitions to and from the LPI transmit mode are controlled via GMII signaling. Transitions to and from the LPI receive mode are controlled by the link partner using sleep and wake signaling.

The PCS 80B/81B Transmit state diagram in Figure 97–11 includes additional states for EEE. The PCS 80B/81B Receive state diagram in Figure 97–12 includes additional states for EEE.

97.1.4 Signaling

TBD

97.1.5 Interfaces

TBD

97.1.6 Conventions in this clause

TBD

97.2 1000BASE-T1 Service Primitives and Interfaces

TBD

97.3 Physical Coding Sublayer (PCS)

Editor's Note: Text written in italics are not approved baseline and are included only for placeholder information. The text will be change to match approved baseline when selected.

97.3.1 PCS service interface (GMII)

The PCS service interface allows the 1000BASE-T1 PCS to transfer information to and from a PCS client. The PCS Interface is precisely defined as the Gigabit Media Independent Interface (GMII) in Clause 35.

97.3.2 PCS functions

The PCS comprises one PCS Reset function and two simultaneous and asynchronous operating functions. The PCS operating functions are: PCS Transmit and PCS Receive. All operating functions start immediately after the successful completion of the PCS Reset function.

The PCS reference diagram, Figure 97–1, shows how the two operating functions relate to the messages of the PCS-PMA interface. Connections from the management interface (signals MDC and MDIO) to other layers are pervasive and are not shown in Figure 97–1.

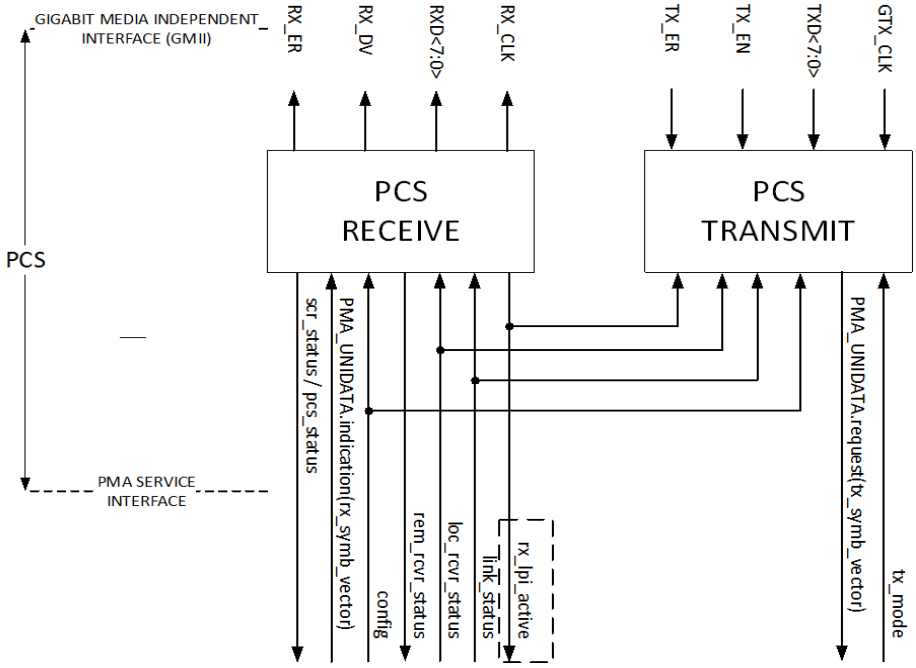

Figure 97–1—PCS reference diagram

97.3.2.1 PCS Reset function

PCS Reset initializes all PCS functions. The PCS Reset function shall be executed whenever one of the following conditions occur:

a) Power on.
b) The receipt of a request for reset from the management entity.

PCS Reset sets pcs_reset=ON while any of the above reset conditions hold true (see 97.3.6.2.2). All state diagrams take the open-ended pcs_reset branch upon execution of PCS Reset. The reference diagrams do not explicitly show the PCS Reset function.

97.3.2.2 PCS Transmit function

The PCS Transmit function shall conform to the PCS 80B/81B Transmit state diagram in Figure 97–11 and the PCS Transmit bit ordering in Figure 97–2 and Figure 97–4.

When communicating with the GMII, the PCS uses an octet-wide, synchronous data path, with packet delimiting being provided by transmit control signals and receive control signals. Alignment to 80B/81B is performed in the PCS. The PMA sublayer operates independently of block and packet boundaries. The PCS provides the functions necessary to map packets between the GMII format and the PMA service interface format.

When the transmit channel is in normal mode, the PCS Transmit process continuously generates 81B blocks based upon the TXD <7:0>, TX_EN and TX_ER signals on the GMII. The subsequent functions of the PCS Transmit process then pack the resulting blocks plus one OAM9 symbol, both of which are then processed by a Reed-Solomon (RS) encoder and subsequently 3B2T mapped into a transmit RS frame of PAM3 symbols. Transmit data-units are sent to the PMA service interface via the PMA_UNITDATA.request primitive. A symbol period, T, is 4/3 ns.

If a PMA_TXMODE.indication message has the value SEND_T, PCS Transmit generates sequences of codes defined in 97.3.4.2 to the PMA via the PMA_UNITDATA.request primitive. These codes are used for training mode and only transmit the values {–1, +1}.

During training mode an InfoField is transmitted at regular intervals containing messages for startup operation. By this mechanism, a PHY indicates the status of its own receiver to the link partner. (See 97.4.2.5.)

In the normal mode of operation, the PMA_TXMODE.indication message has the value SEND_N, and the PCS Transmit function uses an 81B-RS coding technique to generate at each symbol period code-groups that represent data or control. During transmission, the 81B bits are scrambled by the PCS using a PCS scrambler. During data encoding PCS Transmit utilizes a RS frame encoder, then frames are encoded into a sequence of PAM3 symbols and transferred to the PMA.

Dashed rectangles in Figure 97–11 indicate states and state transitions in the transmit process state diagram that shall be supported by PHYs with the EEE capability. PHYs without the EEE capability do not support these transitions.

After reaching the normal mode of operation, EEE-capable PHYs may enter the LPI transmit mode under the control of the MAC via the GMII. The EEE Transmit state diagram is contained within the PCS Transmit function. The EEE capability is described in 97.3.2.2.16.

97.3.2.2.1 Use of blocks

The PCS maps GMII signals into 81-bit blocks inserted into an RS frame, and vice versa, using an 81B-RS coding scheme. The *PAM2* PMA training frame synchronization allows establishment of RS frame and 81B boundaries by the PCS Synchronization process. Blocks and frames are unobservable and have no meaning outside the PCS. The PCS functions ENCODE and DECODE generate, manipulate, and interpret blocks and frames as provided by the rules in 97.3.2.2.2.

97.3.2.2.2 81B-RS transmission code

The PCS uses a transmission code to improve the transmission characteristics of information to be transferred across the link and to support transmission of control and data characters. The encoding defined by the transmission code ensures that sufficient information is present in the PHY bit stream to make clock recovery possible at the receiver. The encoding also preserves the likelihood of detecting any RS frame errors that may occur during transmission and reception of information. In addition, the code enables the receiver to achieve PCS synchronization alignment on the incoming PHY bit stream.

The relationship of block bit positions to GMII, PMA, and other PCS constructs is illustrated in Figure 97–2 for transmit and Figure 97–3 for receive. These figures illustrate the processing of a multiplicity of blocks

97.3.2.2.7 Valid and invalid blocks

A block is invalid if any of the following conditions exists:

a) The block contains an invalid pointer.
b) Any control character contains a value not in Table 97–1.
c) The RS frame containing this 80B/81B block is uncorrectable.

97.3.2.2.8 Idle

Idle (Normal Interframe) control characters are transmitted when TX_EN is not asserted and no other supported control code is present at the GMII. Idle characters may be added or deleted by the PCS to adapt between clock rates. They shall not be added while data is being received. When deleting, the first four Idles after a TX_EN is deasserted shall not be deleted.

97.3.2.2.9 LP_IDLE

The low power idle control characters (LP_IDLEs) are transmitted when TX_EN is not asserted, TX_ER is asserted, and TXD<7:0> = 0x1. A continuous stream of LPI control characters is used to maintain a link in the LPI transmit mode. Idle control characters are used to transition from the LPI transmit mode to the normal mode. EEE compliant PHYs respond to the Assert Low Power Idle condition on the GMII using the procedure outlined in 97.1.3.4.

If EEE is not supported, then LP_IDLE isn't a valid control character.

97.3.2.2.10 Error

The Error is sent when TX_ER is asserted. It is also sent when invalid blocks are received. Error allows physical sublayers such as the PCS to propagate received errors. See R_BLOCK_TYPE and T_BLOCK_TYPE function definitions in 97.3.6.2.4 for further information.

97.3.2.2.11 Transmit process

The transmit process generates blocks based upon the TXD<7:0>, TX_EN and TX_ER signals received from the GMII. Ten GMII data transfers are encoded into each block. It takes 2700 PMA_UNITDATA transfers to send an RS frame of data. Where the GMII and PMA sublayer data rates are not synchronized to that ratio, the transmit process needs to insert idles, or delete idles to adapt between the rates.

The transmit process generates blocks as specified in the transmit process state diagram. The contents of each block are contained in a vector tx_coded<80:0>, which is passed to the scrambler. tx_coded<0> contains the data/ctrl header and the remainder of the bits contain the block payload.

97.3.2.2.12 RS encoder

The 1000BASE-T1 PCS shall encode the transmitted data stream using Reed-Solomon code (450,406). The RS encoder shall follow the notation described in 97.3.2.2.3 where the LSB is the first bit into the RS encoder and the first transmitted bit.

The FEC code used for 1000BASE-T1 links is a linear cyclic block code—the Reed-Solomon code (450,406) over the Galois Field of GF(2^9)—a code operating on 9 bit symbols, as shown in Figure 97–6. The code encodes 406 information symbols and adds 44 parity symbols. The code is systematic, meaning that the information symbols are not disturbed in any way in the encoder and the parity symbols are added separately to each block.

Table 97–2—3B2T Mapping to PAM3

B[2], B[1], B[0]	T[1], T[0]
000	-1,-1
001	0,-1
010	-1,0
011	-1,+1
100	+1,0
101	+1,-1
110	+1,+1
111	0,+1

97.3.2.2.15 81B-RS framer

The 81B-RS framer adapts between the 81-bit width of the 81B blocks and the PAM3 input to the PMA. When the transmit channel is operating in normal mode, the 81B-RS sends one PAM3 symbol of transmit data at a time via PMA_UNITDATA.request primitives. The PMA_UNITDATA.request primitives are fully packed with bits.

97.3.2.2.16 EEE capability

The optional 1000BASE-T1 EEE capability allows compliant PHYs to transition to an LPI mode of operation when link utilization is low. EEE compliant PHYs shall implement the transmit state diagram including the EEE portion, noted by dotted lines in Figure 97–11, within the PCS.

When there is an Assert Low Power Idle while in the SEND_DATA state the PHY transmits the sleep signal to indicate to the link partner that it is transitioning to the SEND_LPI state. The sleep signal is one RS frame composed entirely of LP_IDLE characters. If the LP_IDLE character occurs in the last 80B/81B block then the sleep signal is the next RS frame. The PHY shall transmit no RS frames partially filled with LP_IDLES.

Following the transmission of the sleep signal, quiet-refresh signaling begins, as described in 97.3.5.

While the PMA asserts SEND_N, the lpi_tx_mode variable shall control the transmit signal through the PMA_UNITDATA.request primitive described as follows:

When the PHY is not in the normal state, the lpi_tx_mode variable is ignored.

When the lpi_tx_mode variable takes the value NORMAL the PCS passes coded data to the PMA via the PMA_UNITDATA.request primitive.

When the lpi_tx_mode variable takes the value QUIET the PCS passes zeros to the PMA through the PMA_UNITDATA.request primitive.

When the lpi_tx_mode variable takes the value REFRESH the PMA maps the scrambler output into PAM3.

The quiet-refresh cycle is repeated until Assert Low Power Idle isn't detected at the GMII. This indicates that the local system is requesting a transition back to the normal operational mode. At the next RS frame the PCS transmits a wake frame composed of an entire RS frame containing only Idle. The wake frame shall be sent only during alternating RS frame counts.

Due to the wake signal constrained to occur at the beginning of every second RS frame boundary the PHY wake time may range from 3.6 µs to 10.8 µs (lpi_wake_timer=T_{w_phy} as defined by Clause 78).

97.3.2.3 PCS Receive function

The PCS Receive function shall conform to the PCS 80B/81B receive state diagram in Figure 97–12 and the PCS Receive bit ordering in Figure 97–3 including compliance with the associated state variables as specified in 97.3.6.

The PCS Receive function accepts received code-groups provided by the PMA Receive function via the parameter rx_symb_vector. The PCS receiver uses knowledge of the PMA training alignment to correctly align the 81B-RS frames. The received 81B-RS frames are decoded with error correction; the framing is checked; and the 80B/81B ordered sets are converted to 10 data blocks to obtain the signals RXD<7:0>, RX_DV and RX_ER for transmission to the GMII.

During PMA training mode, PCS Receive checks the received PAM2 framing and signals the reliable acquisition of the descrambler state by setting the parameter scr_status to OK.

When the PCS Synchronization process has obtained synchronization, the RS frame error rate (RFER) monitor process monitors the signal quality asserting hi_rfer if excessive RS frame errors are detected (RS parity error). If 40 consecutive RS frame errors are detected, the block_lock flag is de-asserted. When block_lock is asserted and hi_rfer is de-asserted, the PCS Receive process continuously accepts blocks. The PCS Receive process monitors these blocks and generates RXD<7:0>, RX_DV and RX_ER for transmission to the GMII.

When the receive channel is in training mode, the PCS Synchronization process continuously monitors PMA_RXSTATUS.indication (loc_rcvr_status). When loc_rcvr_status indicates OK, then the PCS Synchronization process accepts data-units via the PMA_UNITDATA.request primitive. It attains frame and block synchronization based on the PMA training frames and conveys received blocks to the PCS Receive process. The PCS Synchronization process sets the block_lock flag to indicate whether the PCS has obtained synchronization. The PMA training sequence includes 1 bit pattern every 180 PAM2 symbols, which is aligned with the PCS Partial RS frame boundary. When the PCS Synchronization process is synchronized to this pattern, block_lock is asserted.

PHYs with the EEE capability support transition to the LPI mode when the PHY has successfully completed training. Transitions to and from the LPI mode are allowed to occur independently in the transmit and receive functions. The PCS receive function is responsible for detecting transitions to and from the LPI receive mode and indicating these transitions using signals defined in 97.3.9.

The link partner signals a transition to the LPI mode of operation by transmitting one frame composed entirely of 80B/81B blocks of LP_IDLES. When blocks of LP_IDLES are detected at the output of the 80B/81B decoder, rx_lpi_active is asserted by the PCS receive function and the LPI character is continuously asserted at the receive GMII. After the sleep frame the link partner begins transmitting zeros, and it is recommended that the receiver power down receive circuits to reduce power consumption. The receive function uses RS frame counters to maintain synchronization with the remote PHY and receives periodic refresh signals that are used to update coefficients, so that the integrity of adaptive filters and timing loops in the PMA is maintained. LPI signaling is defined in 97.3.5. The quiet-refresh cycle continues until the PHY detects the wake frame. The PHY receive function sends Idles to the GMII for the remainder of the wake frame and then resumes normal operation.

97.3.2.3.1 Frame and block synchronization

When the receive channel is operating in normal mode, the frame and block synchronization function receives data via PAM3 PMA_UNITDATA.request primitive. It shall form a PAM3 stream from the primi-

representing the side-stream scrambler state are arbitrarily set. The initialization of the scrambler state is left to the implementer. In no case shall the scrambler state be initialized to all zeros.

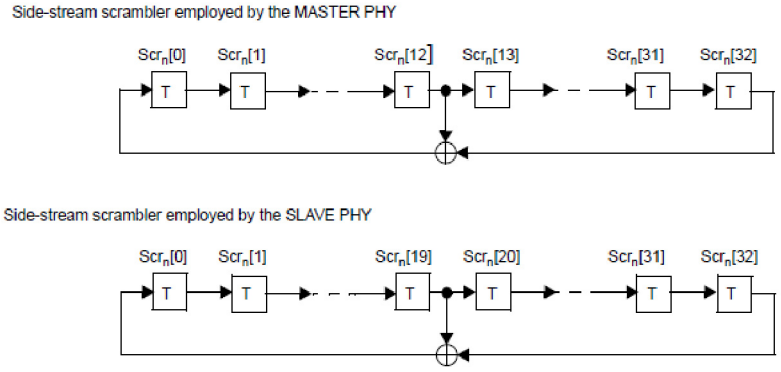

Figure 97–8—A realization of side-stream scramblers by linear feedback shift registers Scr_n

97.3.4.1 Generation of S_n

During PMA training, the training pattern is embedded with indicators to establish alignment to the RS FEC block and the 15 partial frames that comprise the block. The last partial frame is embedded with an information field used to exchange messages between link partners. PMA training signal encoding is based on the generation, at time n, of the bit S_n. The first bit is inverted in the first 14 partial frames of each RS FEC block. The first 96 bits of the 15th partial frame is XOR'd with the contents of the Infofield.

$$S_n = \begin{cases} Scr_n[0] \oplus 1 Infofield_{(n \bmod 180)} & 2519 < (n \bmod 2700) < 2616 \\ Scr_n[0] \oplus 1 & (n \bmod 180) = 0 \\ Scr_n[0] & otherwise \end{cases} \quad (97\text{–}7)$$

97.3.4.2 Generation of symbol T_n

The bit S_n is mapped to the transmit symbol T_n as follows: if $S_n = 0$ then $T_n = +1$, if $S_n = 1$ then $T_n = -1$.

97.3.4.3 PMA training mode descrambler polynomials

The PHY shall acquire descrambler state synchronization to the PAM2 training sequence and report success through scr_status. For side-stream descrambling, the MASTER PHY shall employ the receiver descrambler generator polynomial $g'_M(x) = 1 + x^{20} + x^{33}$ and the SLAVE PHY shall employ the receiver descrambler generator polynomial $g'_S(x) = 1 + x^{13} + x^{33}$.

97.3.5 LPI signaling

PHYs with EEE capability have transmit and receive functions that can enter and leave the LPI mode independently. The PHY can transition to the LPI mode when the PHY has successfully completed training. The transmit function of the PHY initiates a transition to the LPI transmit mode when it generates a sleep signal composed of 80B/81B blocks containing only LPI control characters, as described in 97.3.2.2.16. When the transmitter begins to send the sleep signal, it asserts tx_lpi_active and the transmit function enters the LPI transmit mode.

Within the LPI mode PHYs use a repeating quiet-refresh cycle (see Figure 97–9). The first part of this cycle is known as the quiet period and lasts for a time lpi_quiet_time equal to 354 partial RS frame periods. The quiet period is defined in 97.3.5.2. The second part of this cycle is known as the refresh period and lasts for a time lpi_refresh_time equal to 6 partial RS frame periods. The refresh period is defined in 97.3.5.3. A cycle composed of one quiet period and one refresh period is known as an LPI cycle and lasts for an lpi_qr_time equal to 24*15 = 360 partial RS frame periods.

lpi_offset, lpi_quiet_time, lpi_refresh_time, and lpi_qr_time are timing parameters that are integer multiples of the partial RS frame period. lpi_offset is a fixed value equal to lpi_qr_time/2 + 15. It is used to ensure refresh signals and wake start times are appropriately offset by the link partners.

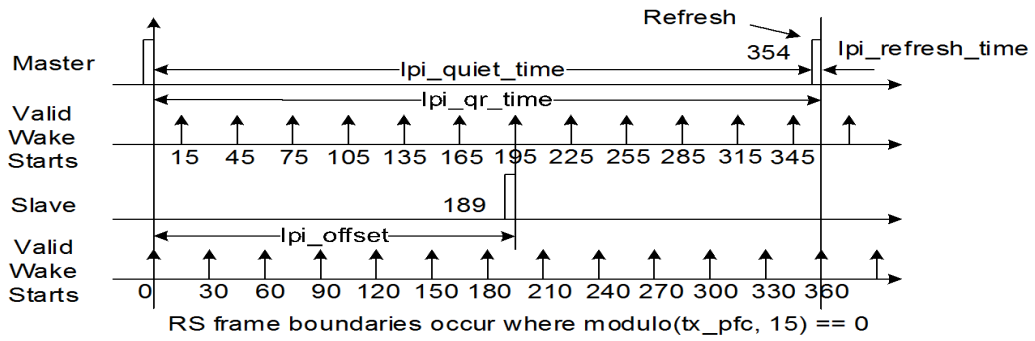

Figure 97–9—LPI signal timing

PHYs begin the transition from the LPI receive mode when they detect the wake frame.

97.3.5.1 LPI Synchronization

To maximize power savings, maintain link integrity, and ensure interoperability, EEE-capable PHYs must synchronize refresh intervals during the LPI mode. The quiet-refresh cycle is established from the Master Partial Frame Count (PFC24) during PMA Training. At the master, partial frame zero and all multiples of 360 partial frames thereafter denote the start of the cycle.

An EEE-capable PHY in slave mode is responsible for synchronizing its partial frame count to the master's partial frame count during link up. The slave shall ensure that its partial frame count is synchronized to the master's partial frames within 1 partial frame. The start of the slave quiet-refresh cycle is delayed from the master by 13 frames (195 partial frames). This offset ensures that the master and slave wake/sense windows are offset from each other and that the refresh periods are nearly a half cycle offset.

Following the transition to PAM3, the PCS continues to count transmitted partial RS frames (tx_pfc), and uses the counter to generate refresh and wake control signals for the transmit functions.

Wake frames may be sent at the beginning of every second RS frame boundary starting at the beginning of the refresh RS frame. This sets wake_period to 30 partial RS frames. The master and slave allowable wake positions do not overlap. The wake frame may start in the same RS frame as a planned refresh and obviate this refresh.

The master and slave shall derive the refresh_active and wake_start signals from the transmitted partial RS frames (tx_pfc) as shown in Table 97-3 and Table 97-4.

Table 97–3—Synchronization logic derived from slave signal partial RS frame count

Slave-side Variable	u = tx_pfc
tx_refresh_active=true	lpi_offset – lpi_refresh_time ≤ mod(u, lpi_qr_time) < lpi_offset
tx_wake_start=true	mod(u, wake_period) = 0

Table 97–4—Synchronization logic derived from master signal partial RS frame count

Master-side variable	v = tx_pfc
tx_refresh_active=true	lpi_quiet_time ≤ mod(v, lpi_qr_time)
tx_wake_start=true	mod(v, wake_period) = wake_period/2

97.3.5.2 Quiet period signaling

During the quiet period the transmitters shall put zeros on to the MDI. During the quiet period the transmitter and may be turned off to save power.

97.3.5.3 Refresh period signaling

During the LPI mode 1000BASE-T1 PHYs use staggered, out-of-phase refresh signaling to maximize power savings. PAM3 refresh symbols are generated from the output of the data mode PCS side-stream scrambler polynomials described in 97.3.2.2.13 with PCS transmit data masked to zero. The scramblers run continuously regardless of the transmit mode. The refresh occupies the last 6 partial RS frames of where the RS frame would occur if it were transmitted.

The OAM symbol is XOR'ed with the scrambler stream at the same relative position to the RS boundaries as it occupies during normal mode.

97.3.6 Detailed functions and state diagrams

97.3.6.1 State diagram conventions

The body of this subclause 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.

The notation used in the state diagrams follows the conventions of 21.5. State diagram timers follow the conventions of 14.2.3.2. The notation ++ after a counter or integer variable indicates that its value is to be incremented.

97.3.6.2 State diagram parameters

97.3.6.2.1 Constants

EBLOCK_R<99:0>
TYPE: bit vector
100-bit vector to be sent to the GMII containing symbol errors in all 10 character locations.

IBLOCK_R<99:0>
TYPE: bit vector
100-bit vector to be sent to the GMII containing idles in all 10 character locations.

~~EBLOCK_R~~IBLOCK_T<99:0>
TYPE: bit vector
100-bit vector to be sent to the ~~GMII~~ encoder containing ~~symbol errors~~ idles in all 10 character locations.

~~IBLOCK_R~~LPBLOCK_R<99:0>
TYPE: bit vector
100-bit vector to be sent to the GMII containing ~~idles~~ LP_IDLEs in all 10 character locations.

~~IBLOCK_T~~LPBLOCK_T<99:0>
TYPE: bit vector
100-bit vector to be sent to the encoder containing ~~idles~~ LP_IDLEs in all 10 character locations.

RFER_CNT_LIMIT
TYPE: TBD
Number of Reed Solomon frames with uncorrectable errors.

RFRX_CNT_LIMIT
TYPE: TBD
Number of Reed Solomon frames received over bit error rate interval.

97.3.6.2.2 Variables

RFER_test_lf
Boolean variable that is set true when a new RS frame is available for testing and false when RFER_TEST_LF state is entered. A new RS frame is available for testing when the Block Sync process has accumulated enough symbols from the PMA to evaluate the next RS frame.

block_lock
Boolean variable that is set true when receiver acquires block delineation.

hi_rfer
Boolean variable which is asserted true when the rfer_cnt exceeds RFER_CNT_LIMIT indicating a bit error ratio > 4×10^{-4}.

pcs_reset
Boolean variable that controls the resetting of the PCS. It is true whenever a reset is necessary including when reset is initiated from the MDIO, during power on, and when the MDIO has put the PCS into low-power mode.

rx_coded<81:0>
Vector containing the input to the 80B/81B decoder including a block valid flag. The format for rx_coded<80:0> is shown in Figure 97–3. The leftmost bit in the figure is rx_coded<0> and the rightmost bit is rx_coded<80>. rx_coded<81> (not shown in the figure) is set to 1 if all parity checks of the Reed Solomon frame are satisfied, otherwise it is set to 0.

rf_valid
Boolean indication that is set true if received Reed Solomon frame is valid. The frame is valid if all parity checks of the coded bits are satisfied.

rx_lpi_active
: This variable is set TRUE upon detection of LP_IDLE. Set FALSE upon wake_detection.

rx_raw<99:0>
: Vector containing 10 successive GMII output transfers. Each transfer is numbered from 0 to 9 with the first transfer numbered as the 0th transfer. The nth GMII transfer is labeled as RX_DV[n], RX_ER[n], RXD[n][7:0].
For n = 0 to 9, rx_raw<8n> = RX_DV[n], rx_raw<8n+1> = RX_ER[n], rx_raw<8n+9:8n+2> = RXD[n][7:0]

rx_wake_frame_complete
: This variable is set TRUE at end of WAKE RS frame, otherwise FALSE.

tx_coded<80:0>
: Vector containing the output from the 80B/81B encoder. The format for this vector is shown in Figure 97–11. The leftmost bit in the figure is tx_coded<0> and the rightmost bit is tx_coded<80>.

tx_data_mode
: Set true when tx_mode = SEND_N, otherwise false.

tx_lpi_active
: This variable is set FALSE at next wake frame if non- LP_IDLE is detected on GMII in any block. This variable is set TRUE on next RS frame if LP_IDLE detected on GMII in the last 80/81 block.

tx_raw<99:0>
: Vector containing 10 successive GMII transfers. Each transfer is numbered from 0 to 9 with the first transfer numbered as the 0th transfer. The nth GMII transfer is labeled as TX_EN[n], TX_ER[n], TXD[n][7:0].
For n = 0 to 9, tx_raw<8n> = TX_EN[n], tx_raw<8n+1> = TX_ER[n], tx_raw<8n+9:8n+2> = TXD[n][7:0]

tx_wake_frame_complete
: This variable is set TRUE at the end of the RS WAKE frame, otherwise FALSE.

lpi_tx_mode
: A variable indicating the signaling to be used from the PCS to the PMA across the PMA_UNITDATA.request (tx_symb_vector) interface.
lpi_tx_mode controls tx_symb_vector only when tx_mode is set to SEND_N.
The variable is set to NORMAL when !tx_lpi_active, indicating that the PCS is in the normal mode of operation.
The variable is set to REFRESH when (tx_lpi_active * tx_refresh active).
The variable is set to QUIET when (tx_lpi_active * !tx_refresh active).

97.3.6.2.3 Timers

State diagram timers follow the conventions described in 14.2.3.2.

97.3.6.2.4 Functions

DECODE(rx_symb_vector<81:0>)
: In the PCS Receive process, this function takes as its argument 82-bit rx_coded<81:0> from the Reed Solomon decoder and descrambler. If rx_coded<81> = 1 then the decoder decodes the

tion for each receive block processed. The PCS shall perform the functions of RFER Monitor, Transmit, and Receive as specified in these state diagrams.

97.3.7 PCS management

The following objects apply to PCS management. If an MDIO Interface is provided (see Clause 45), they are accessed via that interface. If not, it is recommended that an equivalent access be provided.

97.3.7.1 Status

PCS_status:
: Indicates whether the PCS is in a fully operational state. It is only true if block_lock is true and hi_rfer is false. This status is reflected in MDIO register 3.32.12. A latch low view of this status is reflected in MDIO register 3.1.2 and a latch high of the inverse of this status, Receive fault, is reflected in MDIO register 3.8.10.

block_lock:
: Indicates the state of the block_lock variable. This status is reflected in MDIO register 3.32.0. A latch low view of this status is reflected in MDIO register 3.33.15.

hi_rfer:
: Indicates the state of the hi_rfer variable. This status is reflected in MDIO register 3.32.1. A latch high view of this status is reflected in MDIO register 3.33.14.

Rx LPI indication:
: For EEE capability, this variable indicates the current state of the receive LPI function. This flag is set to TRUE (register bit set to one) when the PCS Transmit Receive state diagram (Figure 97–12) is in the RECEIVE_LPI or RECEIVE_WAKE states. This status is reflected in MDIO register 3.1.8. A latch high view of this status is reflected in MDIO register 3.1.10 (Rx LPI received).

Tx LPI indication:
: For EEE capability, this variable indicates the current state of the transmit LPI function. This flag is set to TRUE (register bit set to one) when the PCS Transmit state diagram (Figure 97–11) is in the SEND_LPI or SEND_WAKE states. This status is reflected in MDIO register 3.1.9. A latch high view of this status is reflected in MDIO register 3.1.11 (Tx LPI received).

97.3.7.2 Counters

The following counters are reset to zero upon read and upon reset of the PCS. When they reach all ones, they stop counting. Their purpose is to help monitor the quality of the link.

RFER_count:
: 6-bit counter that counts each time RFER_BAD_RF state is entered. This counter is reflected in MDIO register bits 3.33.13:8. The counter is reset when register 3.33 is read by management. Note that this counter counts a maximum of RFER_CNT_LIMIT counts per RFRX_CNT_LIMIT period since the RFER_BAD_RF can be entered a maximum of RFER_CNT_LIMIT times per RFRX_CNT_LIMIT window.

97.3.7.3 Loopback

The PCS shall be placed in loopback mode when the loopback bit in MDIO register 3.0.14 is set to a one. In this mode, the PCS shall accept data on the transmit path from the GMII and return it on the receive path to the GMII. In addition, the PCS shall transmit a continuous stream of GMII to 81B-RS encoded PAM3 symbols to the PMA sublayer, and shall ignore all data presented to it by the PMA sublayer.

97.3.9.12 PMA_FR_ACTIVE.indication

97.3.9.12.1 Semantics of the primitive

97.3.9.12.2 When generated

97.3.9.12.3 Effect of receipt

97.4 Physical Medium Attachment (PMA) sublayer

97.4.1 PMA functional specifications

The PMA couples messages from a PMA service interface specified in 97.2.2 (TBD) to the 1000BASE-T1 baseband medium, specified in 97.5 (to be confirmed).

The interface between PMA and the baseband medium is the Medium Dependent Interface (MDI), which is specified in 97.8 (TBD).

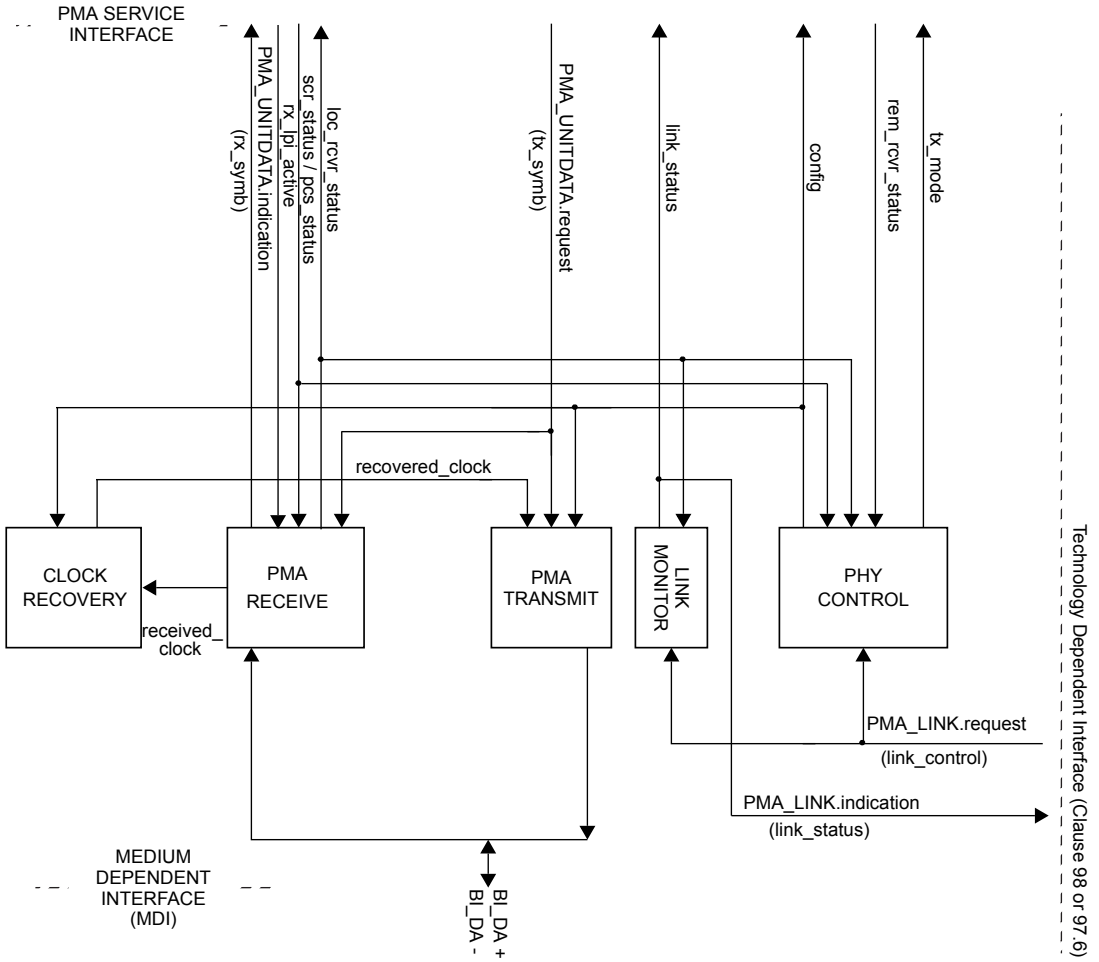

Figure 97–13—PMA reference diagram

NOTE—The recovered_clock arc is shown to indicate delivery of the recovered clock signal back to PMA TRANSMIT for loop timing.

data bandwidth. The OAM is strictly between two 1000BASE-T1 PHYs on the physical layer. Passing OAM information to other layers is outside the scope of this standard.

OAM is operational as long as both PHYs implement this mechanism and link is up. It continues to be operational during low power idle albeit the information is transferred at a slower rate during the refresh cycle.

The 1000BASE-T1 RS frame has a 9-bit reserved field as described in 97.3.2.2.12. This 9-bit is used to exchange OAM frames. The implementation of OAM frame exchange function is optional. However, if 1000BASE-T1 EEE is implemented, then the OAM frame exchange function must be implemented to exchange at minimum the link partner health status.

For the remainder of this subclause, the term OAM is specific to the 1000BASE-T1 PCS level OAM.

97.7.1 Definitions

OAM frame – A frame consisting of 12 byte of data with 12 parity bits

OAM symbol – A 9-bit symbol consisting of one data byte plus a parity bit. 12 OAM symbols makes up an OAM frame.

OAM field – The 9-bit reserved field in each RS frame as described in clause 97.3.2.2.12 or in each refresh cycle as described in ~~TBD (need to insert into EEE section)~~97.3.5.3.

OAM message – A message contains a 4 bit message number plus 8 bytes of message data embedded in an OAM frame. The same OAM message can be repeated on multiple OAM frames.

97.7.2 Functional specifications

97.7.2.1 OAM Frame Structure

Each OAM frame is made up of 12 bytes of data and 12 parity bits. Each symbol consists of 8 bits of data and one parity bit. The parity bit value for symbol 0 should be such that the sum of the number of 1s in the nine bits is even. The parity bit value for symbols 1 to 11 should be such that the sum of the number of 1s in the nine bits is odd.

One OAM frame symbol is placed in the 9-bit OAM field in each RS frame during normal operation. One OAM frame symbol is placed in the 9-bit OAM field in each refresh cycle during low power idle. The 12 OAM frame symbols are consecutively inserted into 12 consecutive RS frames and/or refresh cycles. Once the 12 symbols of the current OAM frame are inserted, the 12 symbols of the next OAM frame are inserted. This process is continuous without any break in the insertion of OAM frame symbols.

Bit 0 of each OAM frame symbol is the first bit transmitted in the 9-bit OAM field. Symbol 0 is the first symbol transmitted in each OAM frame.

The OAM frame boundary can be found at the receiver by determining the symbol parity. Symbol 0 has even parity while all other symbols have odd parity.

If OAM is not implemented then the 9-bit OAM field shall be set to all 0s. If the link partner does not implement OAM, the 9-bit OAM field will remain static and the symbol parity will not change.