119.2.4.9 Test-pattern generators

The PCS shall have the ability to generate a scrambled idle test pattern which is suitable for receiver tests and for certain transmitter tests. When a scrambled idle pattern is enabled, the test pattern is generated by the PCS. The test pattern is an idle control block (block type=0x1E) with all idles as defined in Figure 82-5. The test pattern is sent continuously and is transcoded, scrambled, alignment markers are inserted and finally encapsulated by the FEC.

When the transmit channel is operating in test-pattern mode, the encoded bit stream is distributed to the PCS Lanes as in normal operation (see 119.2.4.7).

If a Clause 45 MDIO is implemented, then control of the test-pattern generation is from the BASE-R PCS test-pattern control register (bit 3.42.3).

119.2.5 Receive function

119.2.5.1 Alignment lock and deskew

The receive PCS forms *n* separate bit streams by concatenating the bits from each of the *n* PMA:IS_UNITDATA_*i*.indication primitives in the order they are received (where *n* = 8 for a 200GBASE-R PCS and *n* = 16 for a 400GBASE-R PCS). It obtains lock to the alignment markers as specified by the alignment marker lock state diagram shown in Figure 119–12. Note that alignment marker lock is achieved before FEC codewords are processed and therefore the alignment markers are processed in a high error probability environment.

After alignment marker lock is achieved on each of the *n* lanes (bit streams), all inter-lane Skew is removed as specified by the PCS synchronization state diagram shown in Figure 119–13. The PCS receive function shall support a maximum Skew of 180 ns, and maximum Skew Variation of 4 ns, between PCS lanes.

119.2.5.2 Lane reorder and de-interleave

PCS lanes can be received on different lanes of the service interface from which they were originally transmitted. The PCS receive function shall order the PCS lanes according to the PCS lane number. The PCS lane number is defined by the unique portion (UM_0 to UM_5) of the alignment marker that is mapped to each PCS lane (see 119.2.4.4).

After all PCS lanes are aligned, deskewed, and reordered, the two FEC codewords are de-interleaved to reconstruct the original stream of two FEC codewords.

119.2.5.3 Reed-Solomon decoder

The Reed-Solomon decoder extracts the message symbols from the codeword, corrects them as necessary, and discards the parity symbols.

The Reed-Solomon decoder shall be capable of correcting any combination of up to t=15 symbol errors in a codeword. The Reed-Solomon decoder shall also be capable of indicating when an errored codeword was not corrected. The probability that the decoder fails to indicate a codeword with t+1 errors as uncorrected is not expected to exceed 10^{-16}. This limit is also expected to apply for t+2 errors, t+3 errors, and so on.

If bypass error indication is not supported or not enabled, when the Reed-Solomon decoder determines that a codeword contains errors that were not corrected, it shall cause the PCS receive function to set every 66-bit block within the two associated codewords to an error block (set to EBLOCK_R). This may be achieved by setting the synchronization header to 11 for all 66-bit blocks created from these codewords by the 256B/257B to 64B/66B transcoder.

The Reed-Solomon decoder may optionally provide the ability to bypass the error indication feature to reduce the delay contributed by the FEC function. The presence of this option is indicated by the assertion of the FEC_bypass_indication_ability variable (see 119.3). When the option is provided it is enabled by the assertion of the FEC_bypass_indication_enable variable (see 119.3).

When FEC_bypass_indication_enable is asserted, additional error monitoring is performed by the Reed-Solomon decoder to reduce the likelihood that errors in a packet are not detected. The Reed-Solomon decoder counts the number of symbol errors detected on all PCS lanes in consecutive non-overlapping blocks of 8192 codewords. When the number of symbol errors in a block of 8192 codewords exceeds 5560, the Reed-Solomon decoder shall ~~cause the PCS receive function to set every 66-bit block to an error block (set to EBLOCK_R)~~ assert hi_ser for a period of 60 ms to 75 ms, causing the PCS to lose alignment. ~~This may be achieved by setting the synchronization header to 11 for all 66-bit blocks created by the 256B/257B to 64B/66B transcoder for this time period.~~

The Reed-Solomon decoder may optionally provide the ability to signal a degradation of the received signal. The presence of this option is indicated by the assertion of the FEC_degraded_SER_ability variable (see 119.3). When the option is provided it is enabled by the assertion of the FEC_degraded_SER_enable variable (see 119.3).

When FEC_degraded_SER_enable is asserted, additional error monitoring is performed by the PCS. The Reed-Solomon decoder counts the number of symbol errors detected on all PCS lanes in consecutive non-overlapping blocks of FEC_degraded_SER_interval (see 119.3.1) codewords, where the least significant bit of FEC_degraded_SER_interval is ignored (evaluated as 0) to make the number of codewords even. If the decoder determines that a codeword is uncorrectable, the number of symbol errors detected is increased by 16. When the number of symbol errors exceeds the threshold set in FEC_degraded_SER_activate_threshold (see 119.3.1), the FEC_degraded_SER bit (see 119.3.1) is set. At the end of each interval, if the number of symbol errors is less than FEC_degraded_SER_deactivate_threshold, the FEC_degraded_SER bit is cleared. If either FEC_degraded_SER_ability or FEC_degraded_SER_enable is de-asserted then the FEC_degraded_SER bit is cleared.

119.2.5.4 Post FEC interleave

After the Reed-Solomon decoder processes the data, data is interleaved on a 10-bit basis into rx_scrambled_am from two codewords corresponding to 40 transcoded blocks in order to recreate the transmitted data stream.

119.2.5.5 Alignment marker removal

For the 200GBASE-R PCS, every 81 920 x 257-bit blocks (corresponds to 4096 codewords) the first 1028 bits of rx_scrambled_am blocks is the vector am_rx<1027:0> where bit 0 is the first bit received. The specific codewords that include this vector are indicated by the alignment lock and deskew function. The 3-bit receive alignment marker status field is assigned from the variable am_rx as follows:

rx_am_sf<2:0> = am_rx<1027:1025>

For the 400GBASE-R PCS, every 163 840 x 257-bit blocks (corresponds to 8192 codewords) the first 2056 bits of rx_scrambled_am blocks is the vector am_rx<2055:0> where bit 0 is the first bit received. The specific codewords that include this vector are indicated by the alignment lock and deskew function. The 3-bit receive alignment marker status field is assigned from the variable am_rx as follows:

rx_am_sf<2:0> = am_rx<2055:2053>

The vector am_rx shall be removed from rx_scrambled_am to create rx_scrambled prior to descrambling.

119.2.5.8 Decode and rate matching

The receive PCS decodes blocks to produce RXD<63:0> and RXC<7:0> for transmission to the 200GMII/400GMII. One 200GMII/400GMII data transfer is decoded from each block. The receive PCS may insert idle control characters to compensate for the removal of alignment markers. If the receive PCS spans multiple clock domains, it may also perform clock rate compensation via the deletion of idle control characters or sequence ordered sets or the insertion of idle control characters (see 82.2.3.6 and 82.2.3.9 for insertion and deletion rules).

The PCS receive decodes blocks as specified in the receive state diagram shown in Figure 119–15.

119.2.6 Detailed functions and state diagrams

119.2.6.1 State diagram conventions

The body of this subclause is composed of state diagrams, including the associated definitions of variables, functions, and counters. 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. The notation ++ after a counter or integer variable indicates that its value is to be incremented.

119.2.6.2 State variables

119.2.6.2.1 Constants

EBLOCK_R<71:0>
: 72-bit vector to be sent to the 200GMII/400GMII containing /E/ in all the eight character locations.

EBLOCK_T<65:0>
: 66-bit vector to be sent to the transcoder containing /E/ in all the eight character locations.

LBLOCK_R<71:0>
: 72-bit vector to be sent to the 200GMII/400GMII containing one Local Fault ordered set. The Local Fault ordered set is defined in 119.3.

LBLOCK_T<65:0>
: 66-bit vector to be sent to the transcoder containing one Local Fault ordered set.

119.2.6.2.2 Variables

align_status
: A variable set by the PCS synchronization process to reflect the status of PCS lane-to-lane alignment. Set to true when all lanes are synchronized and aligned and set to false when the PCS synchronization process is not complete.all_locked
A Boolean variable that is set to true when amps_lock<*x*> is true for all *x* and is set to false when amps_lock<*x*> is false for any *x*.

amp_counter_done
: Boolean variable that indicates that amp_counter has reached its terminal count.

amp_match
: Boolean variable that holds the output of the function AMP_COMPARE.

amp_valid
: Boolean variable that is set to true if the received 120-bit block is a valid alignment marker payload. The alignment marker payload, mapped to a PCS lane according to the process described in 119.2.4.4, consists of 96 known bits. The 48 bits of the common marker portion are compared on a nibble-wise basis (12 comparisons). If 9 or more nibbles in the candidate block match the corre-

sponding known nibbles in the common portion of the alignment marker payload, the candidate block is considered a valid alignment marker payload.

amps_lock<*x*>
: Boolean variable that is set to true when the receiver has detected the location of the alignment marker payload sequence for a given lane on the PMA service interface, where *x* = 0:7 for 200GBASE-R and *x* = 0:15 for 400GBASE-R

current_pcsl
: A variable that holds the PCS lane number corresponding to the current alignment marker payload that is recognized on a given lane of the PMA service interface. It is compared to the variable first_pcsl to confirm that the location of the alignment marker payload sequence has been detected.

cw_A_bad
: A Boolean variable that is set to true if the Reed-Solomon decoder (see 119.2.5.3) fails to correct the current FEC codeword A and is set to false otherwise.

cw_B_bad
: A Boolean variable that is set to true if the Reed-Solomon decoder (see 119.2.5.3) fails to correct the current FEC codeword B and is set to false otherwise.

deskew_done
: A Boolean variable that is set to true when pcs_enable_deskew is set to true and the deskew process is completed. Otherwise, this variable is set to false.

first_pcsl
: A variable that holds the PCS lane number that corresponds to the first alignment marker payload that is recognized on a given lane of the PMA service interface. It is compared to the PCS lane number corresponding to the second alignment marker payload that is tested.

hi_ser
: ~~This variable is defined when the FEC_bypass_indication_ability variable is set to one~~. When FEC_bypass_indication_enable is set to one, this bit is set to one if the number of RS-FEC symbol errors in a window of 8192 codewords exceeds the threshold (see 119.2.5.3) and is set to zero otherwise. When FEC_bypass_indication_enable is set to zero, this bit is set to zero. This variable is mapped to the bit defined in 45.2.3.47k (3.801.2).

pcs_alignment_valid
: Boolean variable that is set to true if all PCS lanes are aligned. PCS lanes are considered to be aligned when amps_lock<*x*> is true for all *x*, each PCS lane is locked to a unique alignment marker payload sequence (see 119.2.4.4), and the PCS lanes are deskewed. Otherwise, this variable is set to false.

pcs_enable_deskew
: A Boolean variable that enables and disables the PCS synchronization process. Received bits may be discarded whenever deskew is enabled. It is set to true when deskew is enabled and set to false when deskew is disabled.

pcs_lane
: A variable that holds the PCS lane number received on lane *x* of the PMA service interface when amps_lock<*x*>=true. The PCS lane number is determined by the alignment marker payloads based on the mapping defined in 119.2.4.4. The 48 bits that are in the positions of the unique marker bits in the received alignment marker payload are compared to the expected values for a given payload position and PCS lane on a nibble-wise basis (12 comparisons). If 9 or more nibbles in the candidate block match the corresponding known nibbles for any payload position on a given PCS lane, then the PCS lane number is assigned accordingly.

pcs_lane_mapping<*x*>
: A variable that holds the value of the pcs_lane received on physical lane *x*.

PCS_status
: A boolean variable that is true when align_status is true and is false otherwise.

reset
: Boolean variable that controls the resetting of the PCS sublayer. It is true whenever a reset is nec-

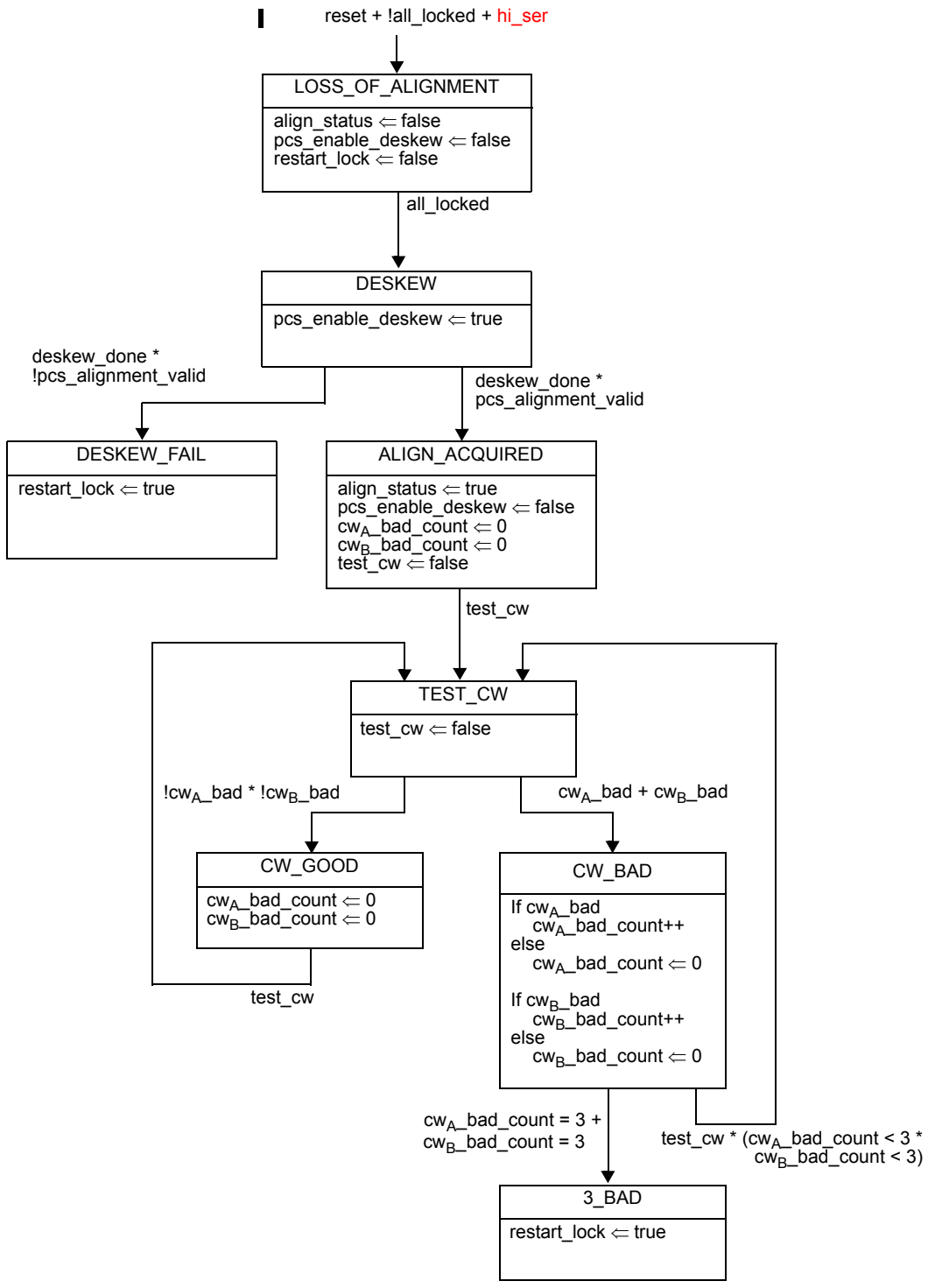

Figure 119–13—PCS synchronization state diagram

119.6.4.2 Receive function

Item	Feature	Subclause	Value/Comment	Status	Support
RF1	Skew tolerance	119.2.5.1	Maximum Skew of 180 ns between PCS lanes and a maximum Skew Variation of 4 ns	M	Yes [ ]
RF2	Lane reorder and de-interleave	119.2.5.2	Order the PCS lanes according to the PCS lane number and de-interleave the FEC codewords	M	Yes [ ]
RF3	Reed-Solomon decoder	119.2.5.3	Corrects any combination of up to t=15 symbol errors in a codeword	M	Yes [ ]
RF4	Reed-Solomon decoder	119.2.5.3	Capable of indicating when a codeword was not corrected.	M	Yes [ ]
RF5	Error monitoring while error indication is bypassed	119.2.5.3	When the number of symbol errors in a block of 8192 codewords exceeds 5560 ~~corrupt 66-bit block synchronization headers~~ assert hi_ser for 60ms to 75ms	BI:M	Yes [ ] N/A [ ]
~~RF6~~	~~Bypass indication error marking~~	~~119.2.5.3~~	~~Synchronization headers are marked for 60 ms to 75 ms when the error threshold is reached.~~	~~BI:M~~	~~Yes [ ]~~ ~~N/A [ ]~~
RF6	256B/257B to 64B/66B transcoder	119.2.5.7	rx_coded_j<65:0>, j=0 to 3 constructed per 119.2.5.7	M	Yes [ ]