Page 169 line 4

When the receive channel is in normal or test-pattern mode, the PCS Synchronization process continuously monitors *inst*:IS_SIGNAL.indication(SIGNAL_OK). When SIGNAL_OK indicates OK, then the PCS Synchronization process accepts data-units via the *inst*:IS_UNITDATA_*i*.indication primitive. It attains block synchronization based on the 2-bit synchronization headers on each one of the PCS lanes. Once block synchronization is found on ~~all~~ a PCS ~~lanes~~lane, then ~~PCS~~ alignment marker lock can be attained by searching for valid alignment markers. After alignment markers are found on all PCS lanes, the PCS lanes can be reordered and deskewed. Note that a particular transmit PCS lane can be received on any receive ~~PCS~~ lane of the service interface due to the skew and multiplexing that occurs in the path.

.

Page 180, line 29

0.0.1 PCS Lane deskew

Once the receiver has ~~PCS lane~~ block lock on ~~each PCS lane (4 or 20 lanes)~~a lane, then ~~the process of deskewing the PCS lanes can begin. This is accomplished by first~~ it begins obtaining alignment marker lock as specified in the ~~PCS~~ alignment marker lock state diagram shown in Figure 0–4~~.~~. This process identifes the PCS lane number received on ~~each PCS lane~~a particular lane of the service interface. After alignment marker lock is ~~achieved~~achieved on all lanes (4 or 20 lanes), then any lane to lane skew ~~can be~~ is removed as shown in the PCS deskew state diagram in Figure 82–15. The skew budget that the PCS receiver shall support is shown in Table 82–5~~.~~.

0.0.2 PCS Lane Reorder

~~The PCS reorders lanes if they are received out of order.~~ Transmit PCS ~~transmit~~ lanes can be received on different lanes of the service inteface than they were originally transmitted on due to skew and multiplexing, and so the receive PCS shall handle receiving any transmit PCS lane on any receive lane of the service interface. The receive ~~lane~~PCS orders the recieved PCS lanes according to the PCS lane number.

Page 181, line 30

0.0.3 Test-pattern checker

When the receive channel is operating in scrambled idle test-pattern mode, the scrambled idle test-pattern checker checks the bits received via *inst*:IS_UNITDATA_*i*.indication primitives.

The scrambled idle test-pattern checker utilizes the ~~PCS lane~~ block lock state diagram, the alignment marker state diagram, the PCS deskew state diagram and the descrambler operating as they do during normal data reception. The BER monitor state diagram is disabled during receive test-pattern mode. When align_status is true and the scrambled idle receive test-pattern mode is active, the scrambled idle test-pattern checker observes the output from the descrambler. When the output of the descrambler is the all idle pattern, a match is detected. When operating in scrambled idle test pattern, the test-pattern error counter counts blocks with a mismatch. Any mismatch indicates an error and shall increment the test-pattern error counter.

If a Clause 45 MDIO is implemented, then control of the test pattern reception is from the BASE-R PCS test-pattern control register (register 3.42.2). In addition errors are counted in the BASE-R PCS test-pattern error counter register (3.43.15:0)

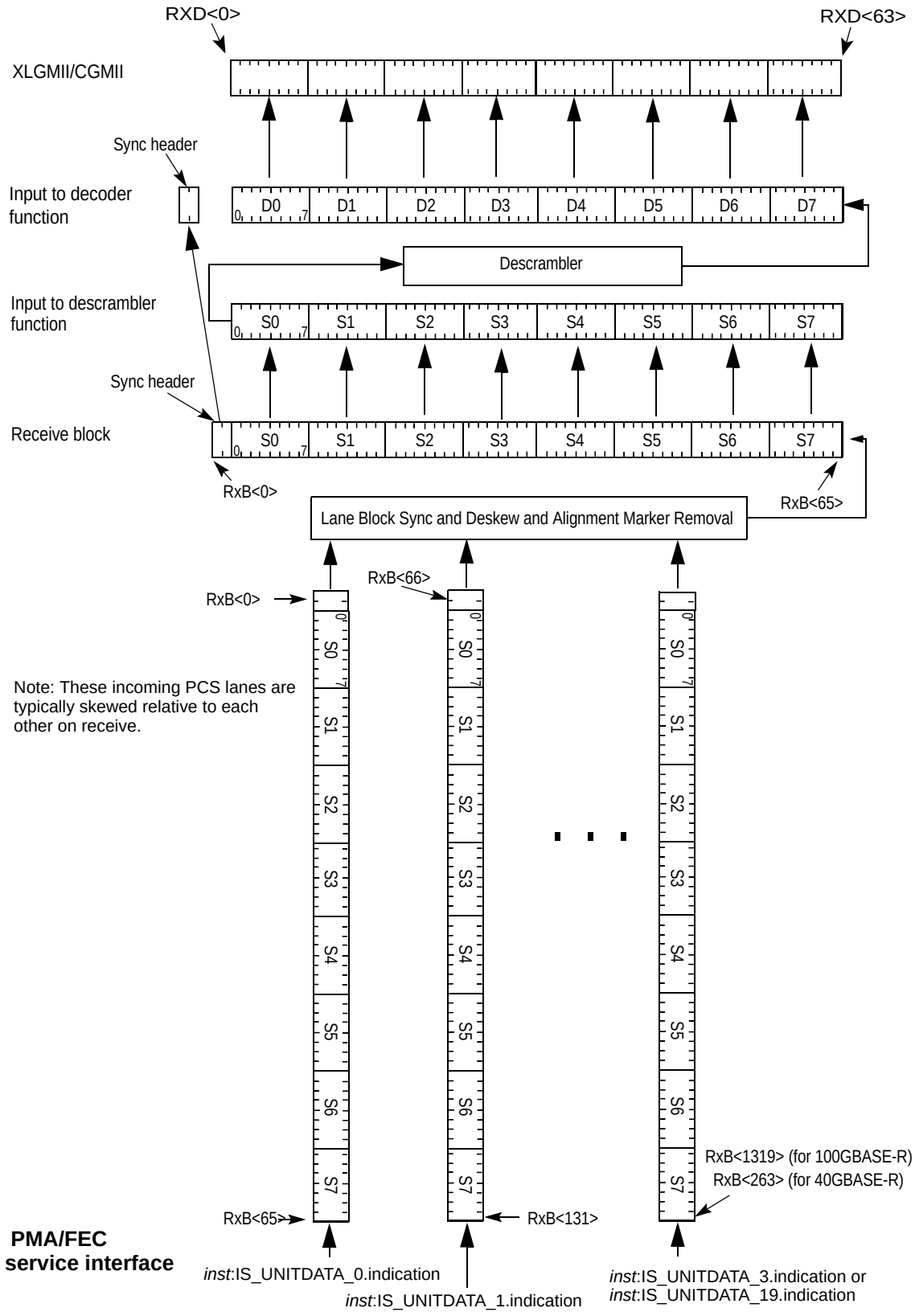


Figure 0–1—PCS Receive bit ordering

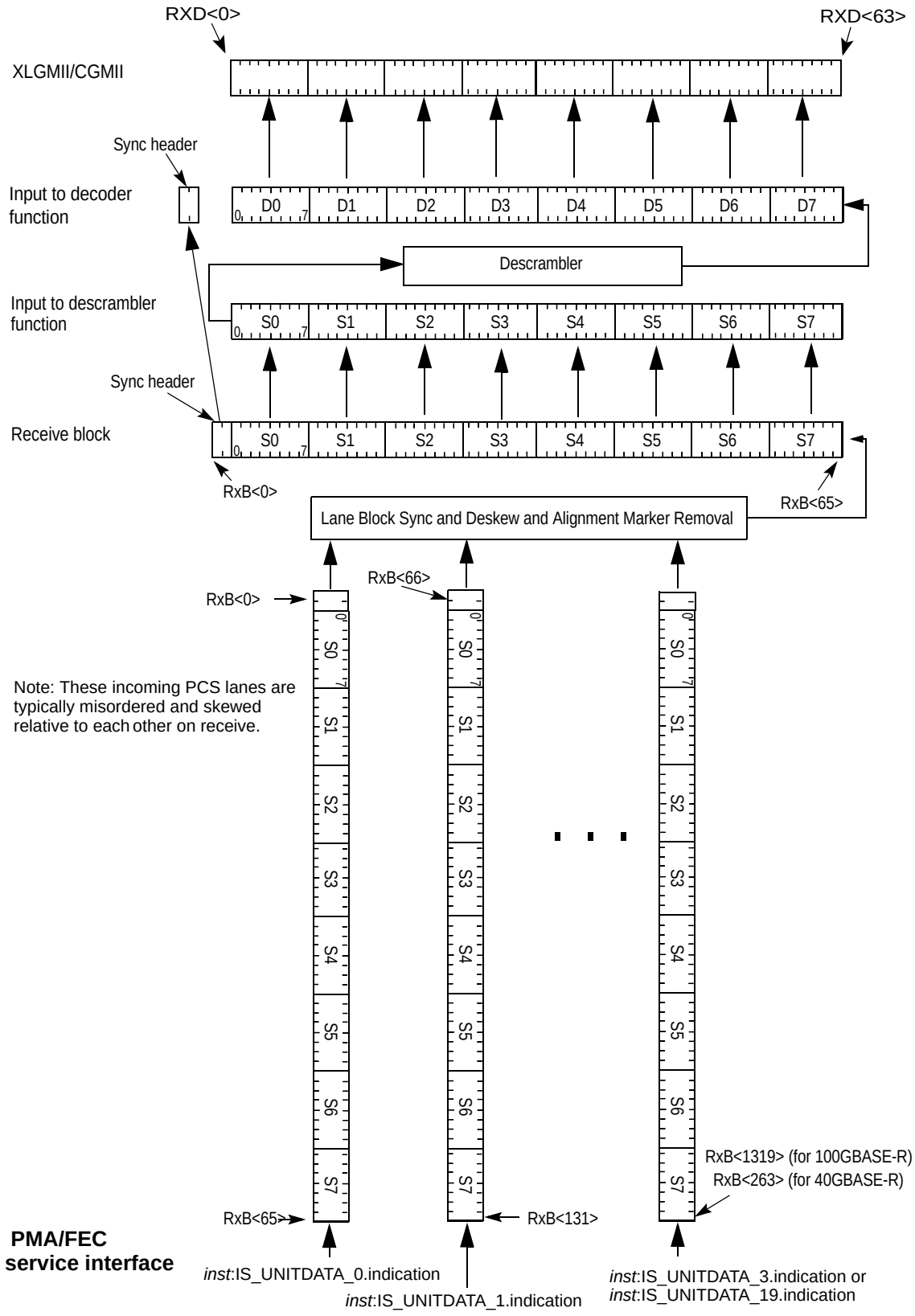


Figure 0–2—PCS Receive bit ordering

Page 182, line 20

0.0.3.0.1 Variables

align_status
: A variable set by the PCS deskew process to reflect the status of the PCS lane-to-lane alignment. Set true when all lanes are synchronized and aligned, set false when the deskew process is not complete.

alignment_valid
: Boolean indication that is set true if all PCS lanes are aligned. It is valid when each ~~PCS~~ lane is in am_lock, with each PCS lane locked to a unique alignment marker from Table 82–2 or 82–3, and when all PCS lanes are deskewed. alignment_valid is false otherwise.

am_lock<x>
: Boolean variable that is set true when receiver acquires alignment marker delineation for a given ~~PCS lane~~lane of the service interface, where x=0:3 for 40GBASE-R and x=0:19 for 100GBASE-R.

am_slip_done
: Boolean variable that is asserted true when the AM_SLIP requested by the ~~PCS~~ alignment marker lock state diagram has been completed indicating that the next candidate 66b block position can be tested.

am_status
: A Boolean variable that represents the following behavior: For all x in am_lock<x>. It is set true when all ~~PCS~~ lanes are in am_lock and false when at least one ~~PCS~~ lane is not in am_lock.

am_counter_done
: Boolean indication that the alignment marker counter is done.

am_valid
: Boolean indication that is set true if received block rx_coded is a valid alignment marker. A valid alignment marker will match one of the encodings in Table 82–2 or 82–3, excluding the BIP_3 and BIP_7 fields, and it will be repeated every 16384 blocks~~. Note that we do not know which marker to expect on which PCS lane~~.

ber_test_sh
: Boolean variable that is set true when a new sync header is available for testing and false when BER_TEST_SH state is entered. A new sync header is available for testing when the Block Sync process has accumulated enough bits from the PMA to evaluate the header of the next block.

block_lock<x>
: Boolean variable that is set true when receiver acquires block delineation for a given ~~PCS lane~~lane of the service interface, where x=0:3 for 40GBASE-R and x=0:19 for 100GBASE-R.

current_am
: This variable holds the value of the current alignment marker. This is compared to the variable first_am to determine if we have alignment marker lock and is always n*16384 66b blocks away from the first_am.

enable_deskew
: A Boolean that indicates the enabling and disabling of the deskew process. Blocks may be discarded whenever deskew is enabled. True when deskew is enabled, false when deskew is disabled.

first_am
: A variable that holds the value of the first alignment marker that is recognized on a given ~~PCS~~ lane. This is used later to compare to future alignment markers.

hi_ber
: Boolean variable which is asserted true when the ber_cnt equals or exceeds 97 indicating a bit error ratio $>10^{-4}$

lane_mapping<x>
: This variable indicates which PCS Lane is received on lane x of the service interface when am_lock<x>=true, where x=0:3 for 40GBASE-R and x=0:19 for 100GBASE-R.

r_block_type

Page 185, line 49

0.0.3.1 State diagrams

The 40GBASE-R PCS shall implement four ~~PCS lane~~ block lock processes as depicted in Figure 0–3. The 100GBASE-R PCS shall implement twenty ~~PCS lane~~ block lock processes as depicted in Figure 0–3. A ~~PCS lane~~ block lock process operates independently on each ~~PCS~~ lane. Each ~~PCS lane~~ block lock process looks for 64 valid 66b sync fields in a row to declare lock. A valid sync field is either a 01 or a 10. Once in lock, the lock process looks for 65 invalid sync fields within a 1024 sync window to declare out of lock. An invalid sync field is a 11 or 00. Once ~~PCS lane~~ block lock is achieved on a ~~PCS~~ lane, then the ~~PCS~~ alignment marker process starts.

The 40GBASE-R PCS shall implement four ~~PCS~~ alignment marker lock processes as depicted in Figure 0–4. The 100GBASE-R PCS shall implement twenty ~~PCS~~ alignment marker lock processes as depicted in Figure 0–4. ~~A PCS~~ An alignment marker lock process operates independently on each ~~PCS~~ lane. The ~~PCS~~ alignment marker lock state diagram shown in Figure 0–4 determines when the PCS has obtained alignment marker lock to the received data stream for a given ~~PCS lane~~lane of the service interface. Each alignment marker lock process looks for two valid alignment markers 16384 66b blocks apart to gain alignment marker lock. On a given ~~PCS~~ lane of the service interface lane markers must match each other and an entry from Table 82–2 for 100GBASE-R or Table 82–3 for 40GBASE-R. Note that the BIP_3 and BIP_7 fields are excluded from the markers when making a match to each other or the tables. Once in lock, a ~~PCS~~ lane will go out of alignment marker lock if four markers are received in a row that do not match the alignment marker that the ~~PCS~~ lane is currently locked to.

The PCS shall run the deskew process as depicted in Figure 82–15. The PCS deskew process is responsible for determining if the PCS is capable of presenting coherent data to the XLGMII/CGMII. The deskew process ensures that all PCS lanes have alignment marker lock, are locked to different alignment markers, and that the skew is within the boundaries of what the PCS can deskew.

The BER Monitor state diagram shown in Figure 82–16 monitors the received aggregate signal for high bit error ratio. The high BER state shall be entered if 97 invalid 66b sync fields are detected within a 500 µs window for 100GBASE-R, or a 1.25 ms window for 40GBASE-R. The high BER state is exited once there are less than 97 invalid sync fields in the same window.

The Transmit state diagram shown in Figure 82–17 controls the encoding of transmitted blocks. It makes exactly one transition for each transmit block processed. Though the Transmit state diagram sends Local Fault ordered sets when reset is asserted, the scrambler may not be operational during reset. Thus, the Local Fault ordered sets may not appear on the PMA/FEC service interface.

The Receive state diagram shown in Figure 82–18 controls the decoding of received blocks. It makes exactly one transition for each receive block processed.

The PCS shall perform the functions of ~~PCS lane~~ block lock, alignment marker lock, PCS deskew, BER Monitor, Transmit and Receive as specified in these state diagrams.

Page 187, line 1

0.0.4 PMD MDIO function mapping

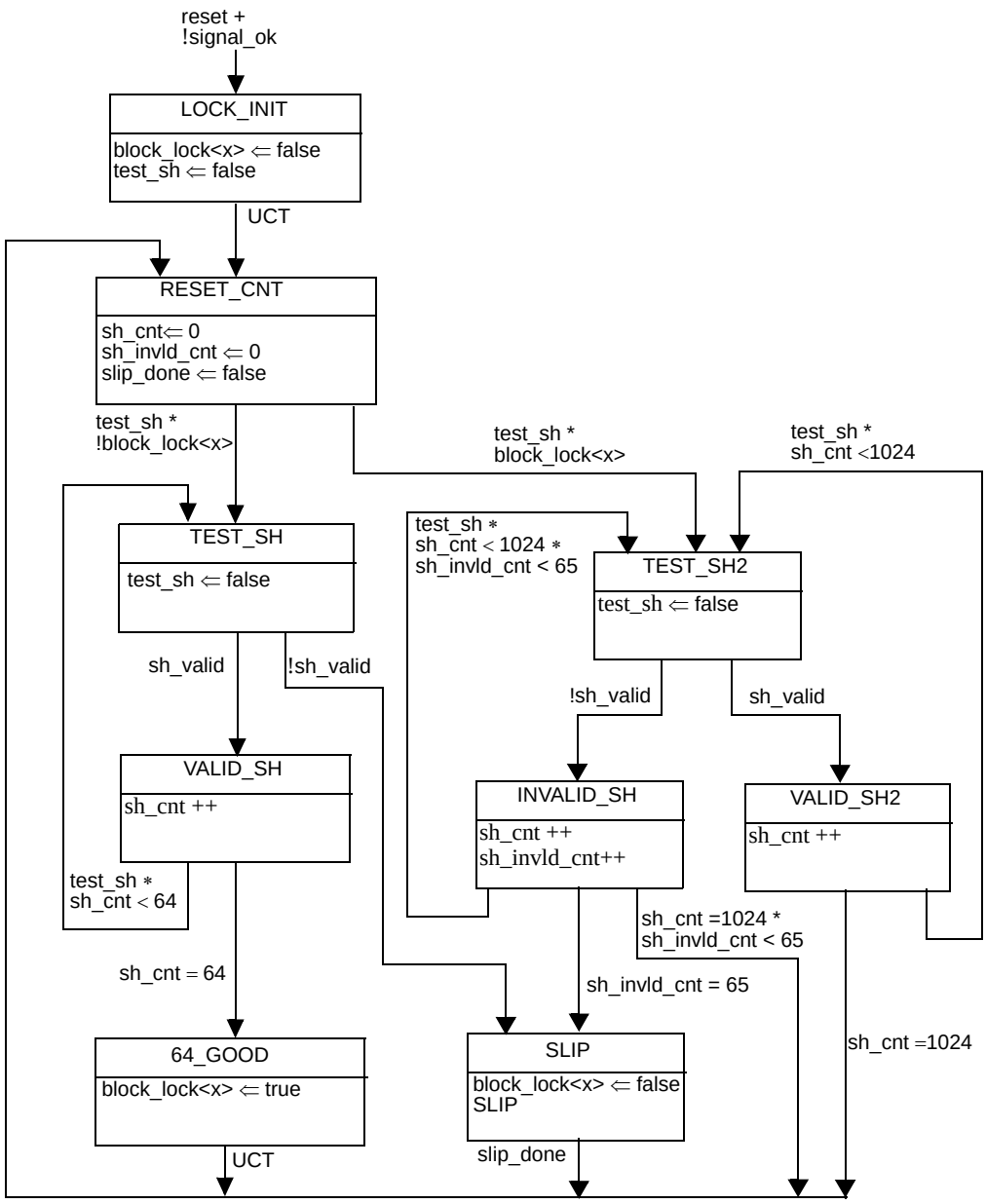
The optional MDIO capability described in Clause 45 defines several variables that may provide control and status information for and about the PCS. Mapping of MDIO control variables to PCS control variables is shown in Table 0–1. Mapping of MDIO status variables to PMD status variables is shown in Table 0–2.

Table 0–1—MDIO/PMD control variable mapping

MDIO control variable	PCS register name	Register/ bit number	PCS control variable
Reset	Control register 1	3.0.15	reset
Loopback	Control register 1	3.0.14	Loopback
Transmit test-pattern enable	BASE-R PCS test-pattern control register	3.42.3	tx_test_mode
Receive test-pattern enable	BASE-R PCS test-pattern control register	3.42.2	rx_test_mode

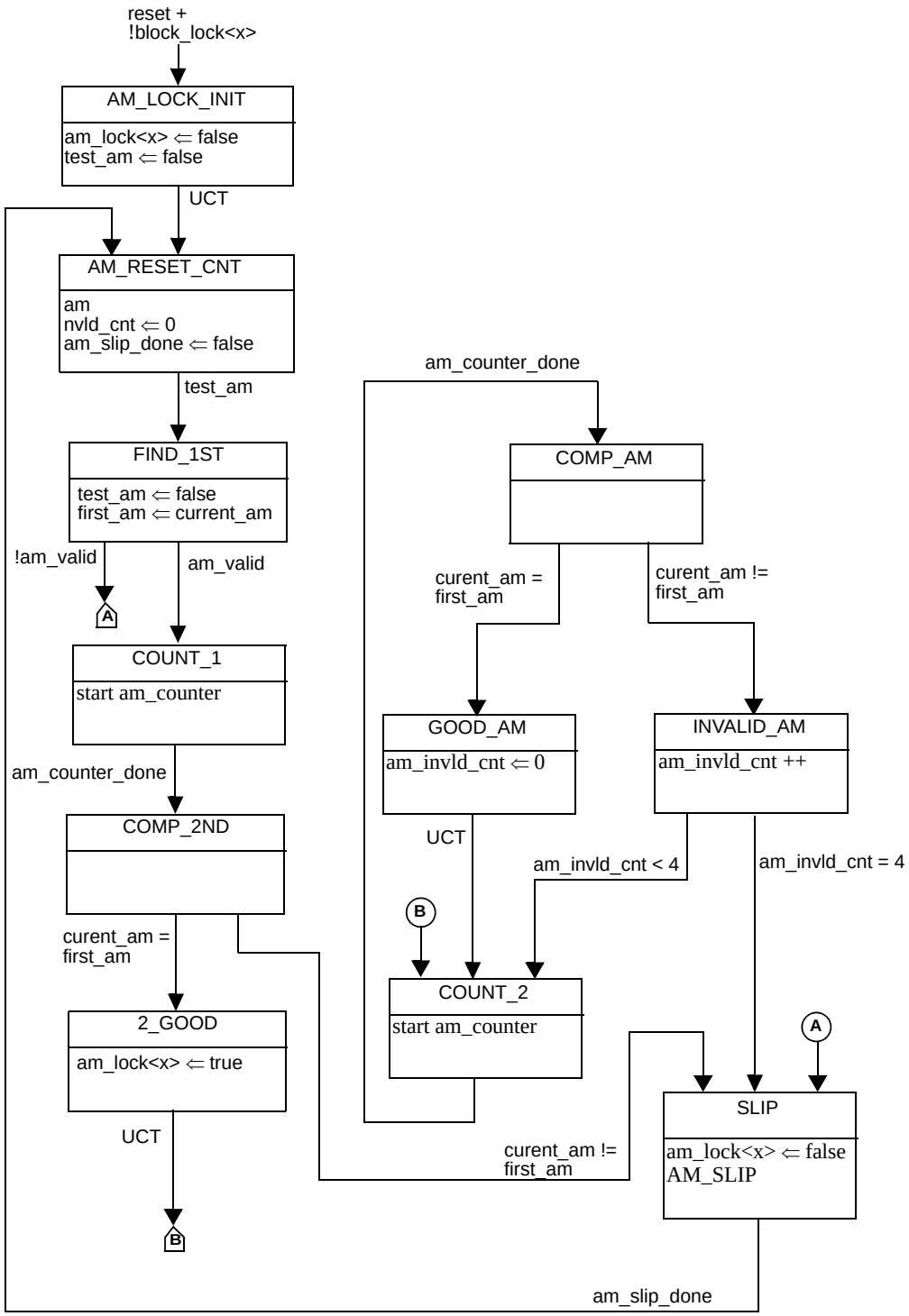
Table 0–2—MDIO/PMD status variable mapping

MDIO status variable	PCS register name	Register/ bit number	PCS status variable
10/40/100GBASE-R and 10GBASE-T receive link status	10/40/100GBASE-R and 10GBASE-T PCS status 1 register	3.32.12	PCS_status
10/40/100GBASE-R and 10GBASE-T PCS high BER	10/40/100GBASE-R and 10GBASE-T PCS status 1 register	3.32.1	hi_ber
~~PCS Lane~~ Block x lock	Multi-lane BASE-R PCS alignment status register 1 and 2	3.50.7:0 3.51.11:0	block_lock<x>
~~PCS~~ Lane x aligned	Multi-lane BASE-R PCS alignment status register 3 and 4	3.52.7:0 3.53.11:0	am_lock<x>
PCS lane alignment status	Multi-lane BASE-R PCS alignment status register 1	3.50.12	align_status
BER	10/40/100GBASE-R and 10GBASE-T PCS status 2 register	3.33.13:8	ber_count
Errored blocks	10/40/100GBASE-R and 10GBASE-T PCS status 2 register	3.33.7:0	errored_block_count
Test-pattern error counter	BASE-R PCS test-pattern error counter register	3.43.15:0	test_pattern_error_count
BIP error counters	BIP error counter	3.90 through 3.99	bip_counter



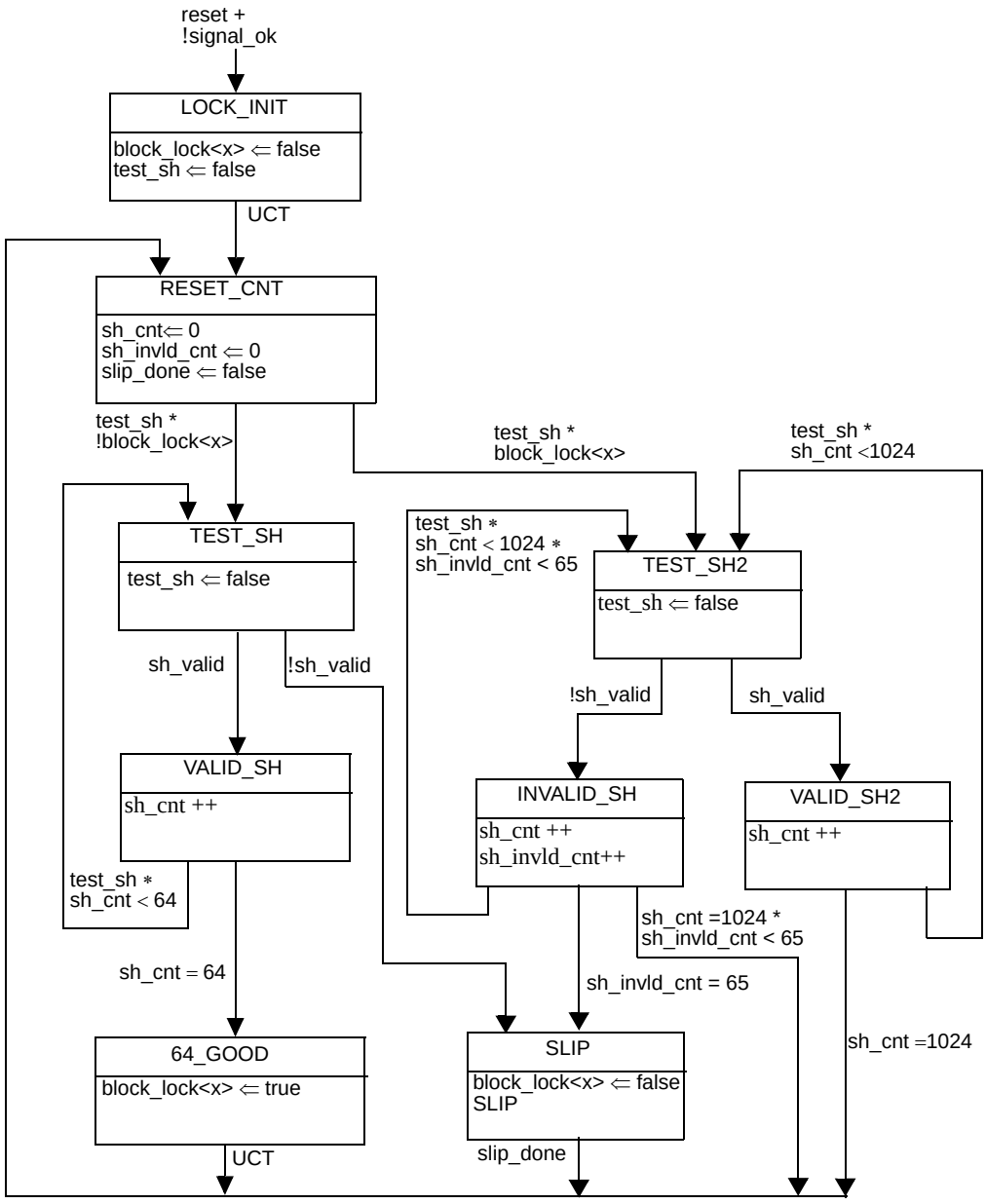
Note - block_lock<x> refers to the received PCS lane x, where x = 0:3 (for 40GBASE-R) or 0:19 (for 100GBASE-R)

Figure 0–3—PCS lane lock state diagram



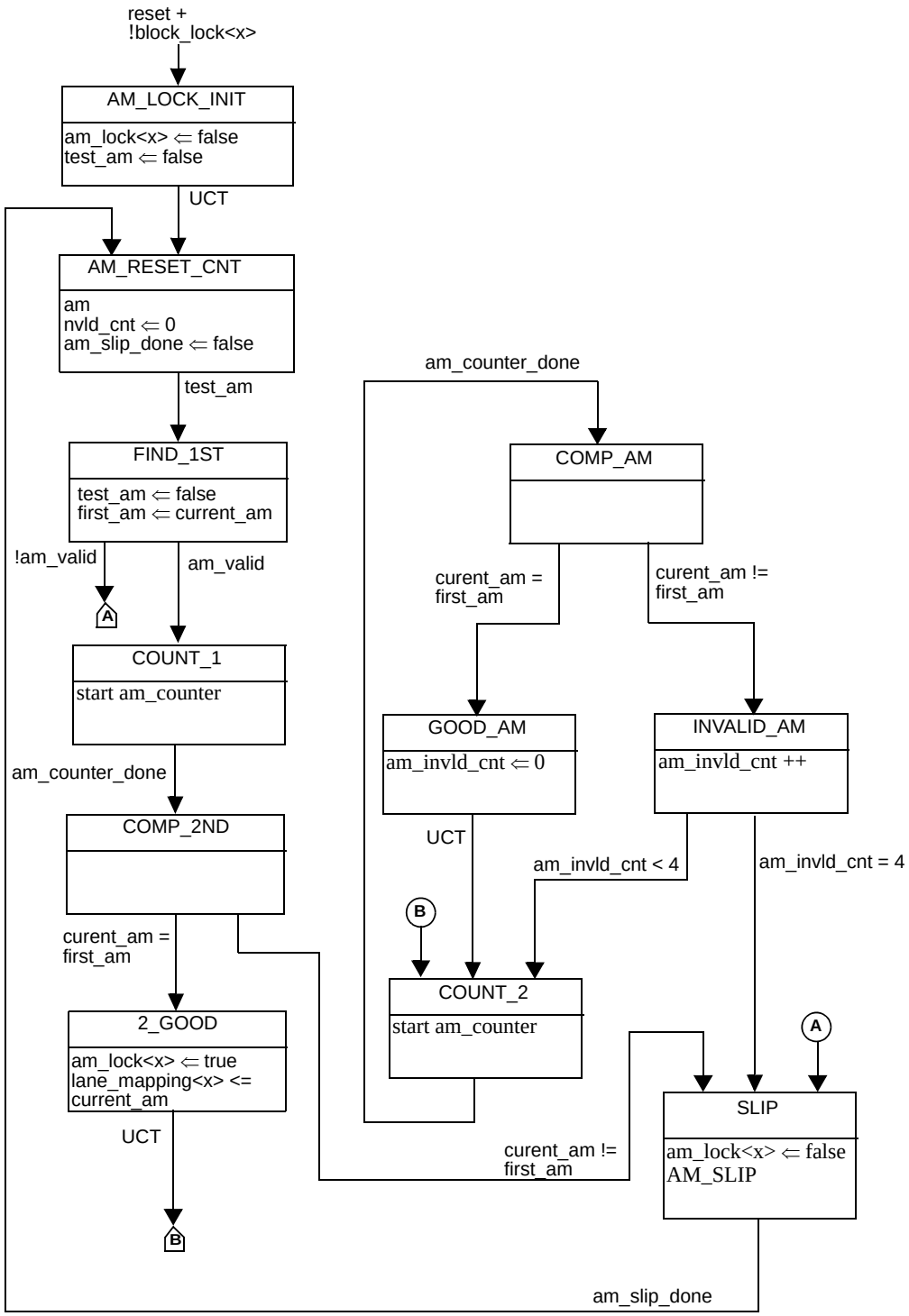
Note - am_lock<x> refers to the received lane x, where x = 0:3 (for 40GBASE-R) or 0:19 (for 100GBASE-R)

Figure 0–4—PCS alignment marker lock state diagram



Note - block_lock<x> refers to the received lane x of the service interface, where x = 0:3 (for 40GBASE-R) or 0:19 (for 100GBASE-R)

Figure 0–5—Block lock state diagram



Note - am_lock<x> refers to the received lane x of the service interface, where x = 0:3 (for 40GBASE-R) or 0:19 (for 100GBASE-R)

Figure 0–6— Alignment marker lock state diagram

Page 199, line 1

0.0.4.1 State diagrams

Item	Feature	Subclause	Value/Comment	Status	Support
SM1	40GBASE-R ~~PCS Lane~~ Block Lock	0.0.3.1	Implements 4 ~~lane~~ block lock processes as depicted in Figure 0–3	M	Yes [ ] No [ ]
SM2	100GBASE-R ~~PCS Lane~~ Block Lock	0.0.3.1	Implements 20 ~~lane~~ block lock processes as depicted in Figure 0–3	M	Yes [ ] No [ ]
SM3	The SLIP function evaluates all possible bit positions	82.2.18.2.3		M	Yes [ ] No [ ]
SM4	40GBASE-R Alignment Marker Lock	0.0.3.1	Implements 4 alignment marker lock processes as depicted in Figure 0–4	M	Yes [ ] No [ ]
SM5	100GBASE-R Alignment Marker Lock	0.0.3.1	Implements 20 alignment marker lock processes as depicted in Figure 0–4	M	Yes [ ] No [ ]
SM6	The AM_SLIP functions evaluates all possible blocks	82.2.18.2.3		M	Yes [ ] No [ ]
SM7	40GBASE-R and 100GBASE-R PCS deskew state diagram	0.0.3.1	Meets the requirements of Figure 82–15	M	Yes [ ] No [ ]
SM8	40GBASE-R BER Monitor	0.0.3.1	Meets the requirements of Figure 82–16 with xus_timer_done set to 1.25 msec	M	Yes [ ] No [ ]
SM9	100GBASE-R BER Monitor	0.0.3.1	Meets the requirements of Figure 82–16 with xus_timer_done set to 500 µs	M	Yes [ ] No [ ]
SM10	40GBASE-R and 100GBASE-R Transmit process	0.0.3.1	Meets the requirements of Figure 82–17	M	Yes [ ] No [ ]
SM11	40GBASE-R and 100GBASE-R Receive process	0.0.3.1	Meets the requirements of Figure 82–18	M	Yes [ ] No [ ]