Titanium Ethernet 10GBase-KR User Guide

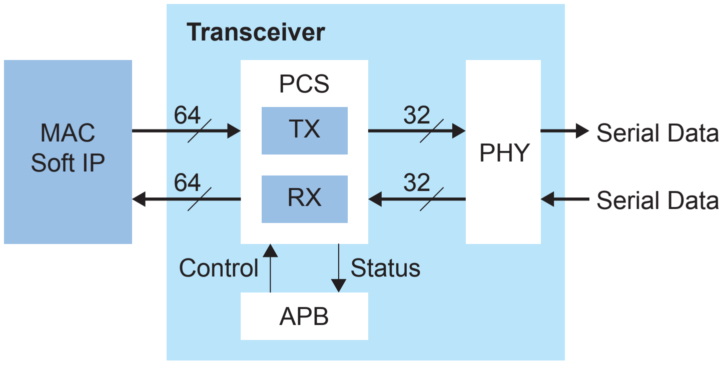
UG-Ti10GBASEKR-v1.1
November 2024
www.efinixinc.com

Contents

Introduction

Titanium transceivers consist of a physical medium attachment (PMA) and a physical coding sublayer (PCS). The PMA connects the FPGA to the lane, generates the required clocks, and converts the data from parallel to serial or serial to parallel. The PCS contains the digital processing interface between the PMA and the FPGA fabric. The PCS supports SGMII, 10GBase-KR, and PCIe Gen4 as well as PMA Direct. This user guide provides the specifications for the 10GBase-KR interface.

Figure 1: System-Level Block Diagram



Features

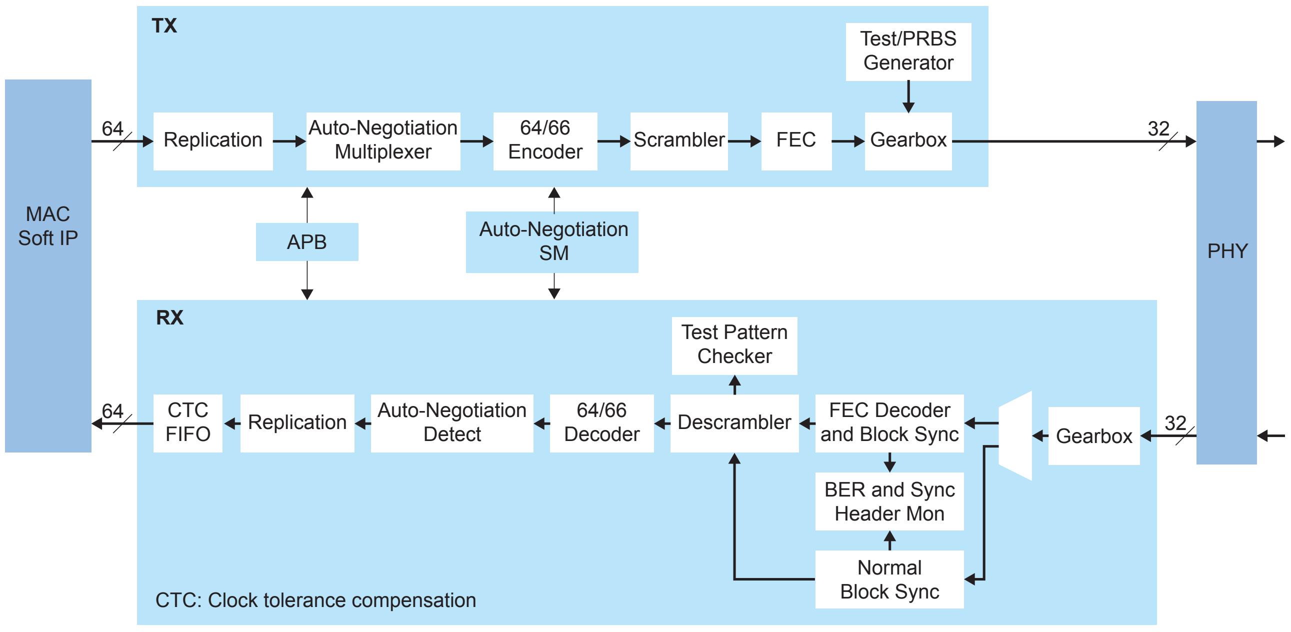
- Per-lane, 64-bit USXGMII interface
  - Supports 100 Mbps, 1 Gbps, and 10 Gbps MAC Ethernet data rates
  - SerDes rate of 10.3125 Gbps
- Auto-negotiation (Clause 37, non-backplane Ethernet) link status notification
- Compliant with IEEE Std. 802.3 Clauses 49 and 129
- 64b/66b encoding/decoding
- Data scrambling/descrambling on TX and RX path
- ±100 ppm clock drift between local clock and recovered clock
- 16 KB jumbo frame
- PRBS pattern generators and checkers:
  - PRBS31 and PRBS9 pattern generators and checkers
  - Pseudo-random pattern generators and checkers for the local fault and zero data patterns
  - Scrambled idle data pattern generator and checker
  - Square wave pattern generator
- Forward Error Correction (FEC) support (Clause 74)
- APB control status register interface

Functional Description

The 10GBase-KR PCS includes all the functionality of a standard 10GBase-R PCS along with functionality to replicate XGMII data stripes to adapt slower Ethernet data rates to the fixed speed of the SerDes. It also includes functionality for a modified Clause 37 auto-negotiation state machine to pass status information from the PHY to the MAC as defined by the USXGMII standard. (The PCS does not support the packet control header (PCH) non-standard preamble described in the USXGMII standard.)

The 10GBase-KR PCS complies with the Cisco Universal SXGMII Interface and IEEE Std. 802.3 Clause 49. Additionally, it supports Base-R FEC (as per IEEE Std. 802.3 Clause 74). You can configure the management and configuration through an APB interface. The SerDes connection is a 32-bit interface.

Figure 2: Funtional Block Diagram

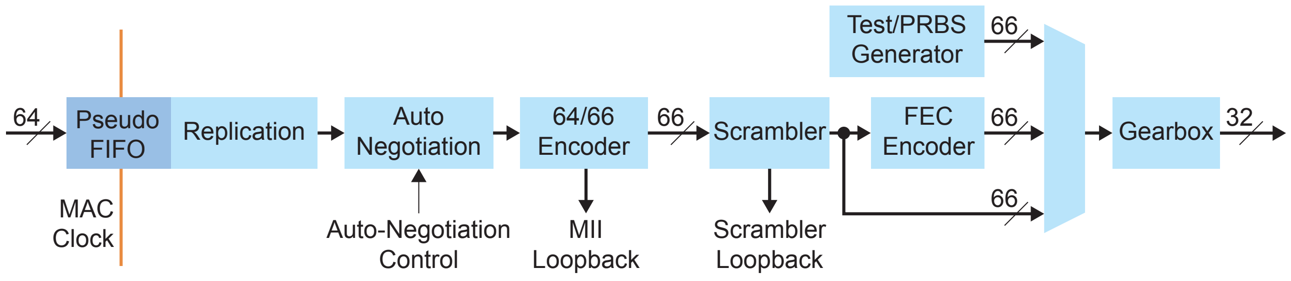


Transmitter (TX)

The transmitter performs the following functions:

- Replicates XGMII data for slower Ethernet MAC data rates compared to the fixed speed of the SerDes (10.3125 Gbps)
- Performs TX auto-negotiation in which transmit data is sourced from the PCS block.
- Performs 64b/66b encoding to generate 64-bit data with a 2-bit sync header.
- Scrambles the 66-bit data.
- Performs FEC coding and framing in FEC mode.
- Uses the gearbox to convert the 66-bit transmission block into SerDes 32b width data.
- Generates test patterns in test mode.

Figure 3: 10GBase-KR Transmitter



Pseudo-FIFO and Replication

The MAC rate can be 100 Mbps, 1 Gbps, or 10 Gbps. When the MAC rate is not the same as the SerDes link rate, the XGMII words are replicated. For example, if the MAC rate is 1 Gbps and the SerDes link rate is 10 Gbps, each XGMII 32-bit word is replicated 10 times when going into the 64/66B encoder (which always runs at the SerDes link rate).

To handle the clock domain difference between the MAC and the internal PCS, a pseudo-FIFO bridges between the host `mac_tx_clk` and the `internal pcs_tx_clk`. The host clock scales according to the MAC rate/SerDes rate ratio. For example, if the SerDes link rate is 10 Gbps and the MAC rate is 1 Gbps, the `mac_tx_clk` is 1/10th of the normal operating frequency.

The TX FIFO allows a standard XGMII-style interface to the host with the only change being the clock frequency.

Auto-Negotiation

This block selects the auto-negotiation controlled data words to the 64/66B encoder.

This module sends the following data types:

- Auto-negotiation configuration ordered sets
- IDLE order set

Note: The 10GBase-KR block does not support the auto-negotiation function for backplane Ethernet, which is specified in Clause 73. It does support auto-negotiation for non-backplane Ethernet (Clause 37).

Transmitter Encoder

The 64/66b encoder converts 8 data octets and 1 control octet into one 66b transmission block according to IEEE 802.3 spec Clause 49.

There are 6 reserved codes along with low power idle, and signal and sequence ordered sets (including the auto-negotiation ordered set). There are only 15 valid data block formats. If none of these formats is detected, the encoder issues an error message.

This module is also supports the scrambled idles test pattern as described in IEEE 802.3 Clause 107. Any input data from the MAC is ignored and the encoder outputs a continuous stream of encoded Idle blocks to the scrambler.

Parallel Scrambler

The scrambler uses the $g(x) = 1 + x39 + x58$ polynomial on the 64-bit transmit data stream (it ignores the sync bits). The scrambler output can be descrambled by a self-synchronizing descrambler using a similar polynomial as specified in IEEE Std. 802.3 Clause 49.

The scrambler has two modes:
- Data mode
- Pseudo-random test mode

Data mode is the normal operating mode and scrambles the 64/66 encoded data.

Pseudo-random test mode is for test pattern generation. The transmitter feeds the scrambler a pre-determined input data block of either 64 zeroes or two local fault sequences, depending on the state of the control input `tx_tst_data_sel`.

FEC Encoder

The FEC encoder takes in 32 66-bit blocks from the scrambler and encodes them into a single FEC block of 2,112 bits. The FEC encoder compresses the two sync bits into one transcode bit as specified in IEEE Std. 802.3 Clause 74. The resulting 32 65-bit blocks are passed through the (2112,2080) encoding process, which generates 32 parity-check bits. The generator polynomial used to generate these bits is:

$g(x) = x32 + x23 + x21 + x11 + x2 + 1$

The encoder appends the parity check bits to the end of the FEC block. Finally, the encoder scrambles the FEC block using the PN-2112 pseudo-noise sequence with a generator polynomial of $r(x) = 1 + x39 + x58$.

Test Pattern Generator

The test pattern generator generates test patterns that are either PRBS 31 or PRBS 9 as specified in IEEE Std. 802.3 Clauses 49 and 68. The square wave and PRBS test patterns have higher priority than the PRBS test.

The generator can create 6 test patterns as shown in the following table:

Table 1: TX Test Pattern Modes

Pattern	tx_tst_en	tx_scr_idle_en	tx_tst_dat_sel	tx_prbs9_en	tx_prbs31_en	tx_sqw_en
Normal operation	0	0	0	0	0	0
Scrambled 0	1	0	1	0	0	0
Scrambled local fault	1	0	0	0	0	0
PRBS 9	X	X	X	1	0	0
PRBS 31	X	X	X	0	1	0
Scrambled idle	1	1	0	0	0	0
Square wave	X	X	X	X	X	1

Gearbox

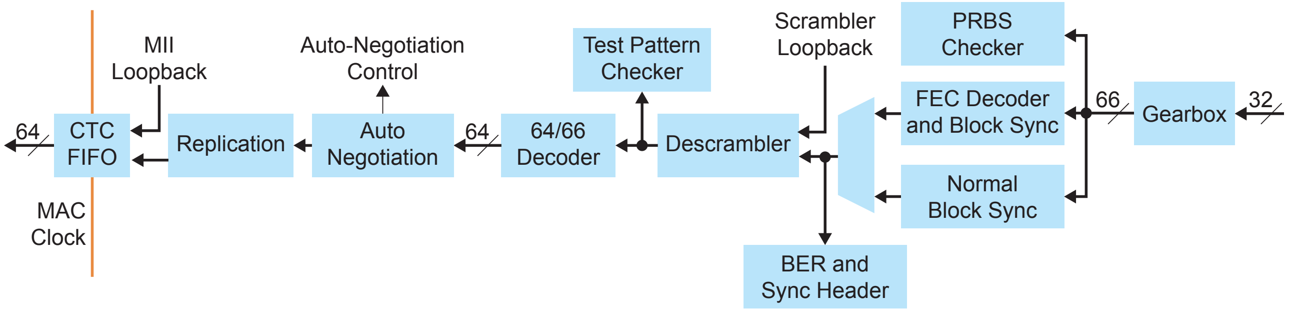
The gearbox converts the 66-bit data output from the scrambler or FEC encoder into 32-bit data to send to the SerDes interface.

Receiver (RX)

The receiver performs the following functions:

- Uses the gearbox to convert the SerDes 32b data into 66b data.
- Monitors the data from the SerDes interface for BER and sync headers when `signal_ok` is asserted.
- When not in FEC mode, the receiver achieves 66b block synchronization by shifting one bit at a time until a succession of valid synchronization bits are seen.
- In FEC mode, it performs FEC descrambling, FEC framing synchronization, FEC decoding, and data correction (if an error happens).
- In FEC mode, the decoder can be configured to indicate errors.
- Descrambles the 66b received data using the $G(x) = 1 + x39 + x58$ polynomial
- Performs 64b/66b decoding on the 66b data block to obtain the 64b data and 8b control data for the MAC.
- De-replicates the received code words to the relevant MAC sub-rate.
- Inserts or deletes idle characters to adapt between the host clock and recovered clock.
- Deletes control sequence characters to adapt between the host clock and recovered clock.
- Performs test pattern checking.

Figure 4: 10GBase-KR Receiver



Gearbox

The RX gearbox converts the 32-bit data from the SerDes to 66-bit blocks that are later used to achieve block synchronization. You reset the gearbox by toggling the Control Register's `signal_ok`.

FEC Decoder and Block Synchronizer

If FEC mode is enabled, the 66B data blocks are passed into the FEC decoder and block synchronization module, which performs FEC framing synchronization, FEC descrambling, and FEC decoding.

The FEC decoder establishes FEC block synchronization based on repeated decoding of the 2112-bit received FEC sequence. Upon reset, the first data bit received via the gearbox is assumed to be the block start position. 2112b of data are subsequently passed through an FEC descrambling and FEC decoding circuit.

The descrambling circuit is a PN-2112 generator based on the polynomial:

$g(x) = x32 + x23 + x21 + x11 + x2 + 1$

From the decoding the 2112b,if the syndrome check is invalid, the block start position is shifted by one bit position and the process is repeated. Once the parity check is valid for a potential block start position, the bit slipping process is halted. If "n" consecutive FEC blocks are received with good parity then Block Sync is reported via the top level output block_lock. If any block within the "n" count fails the parity check, then the bit slipping process is restarted. Once Block Sync is established, "m" consecutive blocks with bad parity are required to drop Block Sync and restart the bit slipping mechanism.

Once FEC Block Sync has been attained, the error-correcting circuit is activated. Any subsequent FEC blocks with 11 or less consecutive bit errors are automatically corrected.

The datapath output from this module functionally matches the output of the Receive Path Synchronizer used when FEC mode is disabled.

Receive Block Synchronizer (Non-FEC)

The receive path synchronizer works with the 66b data from the gearbox. A bit-slip shifter slips the data one bit at a time, as controlled by the state machine, to detect a valid sync header boundary.

If 64 consecutive valid sync headers are detected, the synchronizer reports block sync via the top-level signal `block_lock`. If any sync header within the 64 consecutive 66b blocks is invalid, the bit slipping is restarted. Once block sync is established, 16 invalid sync headers within 64 consecutive 66b blocks are required to drop block sync and restart the bit slipping.

Bit Error Ratio Monitor

The bit error ratio (BER) monitor has a 5-bit counter that counts the number of invalid sync headers detected by the normal block synchronizer or the FEC decoder. If the count reaches a certain value during a specified interval, the BER monitor sets the `hi_ber` register.

The monitor sets `hi_ber` when it detects 16 errors within a 125 μs window, and automatically writes set to the appropriate value in APB register.

Parallel Descrambler

The parallel descrambler takes 64-bit data from the synchronizer and applies the $g(x) = 1 + x39 + x58$ polynomial to the input data to reverse the scrambling function of the transmitter.

Test Pattern Checker

The receive datapath implements various test pattern checkers, which can be used for self-test and target testing of specific areas of the datapaths. The following table shows the possible signal combinations for the different test modes.

Table 2: RX Test Pattern Modes

Pattern	rx_tst_en	rx_scr_idle_en	rx_tst_dat_sel	rx_prbs9_en	rx_prbs31_en
Normal operation	0	0	0	0	0
Scrambled 0	1	0	1	0	0
Scrambled local fault	1	0	0	0	0
PRBS 9	X	X	X	1	0
PRBS 31	X	X	X	0	1
Scrambled idle	1	1	0	0	0

In the PRBS test modes, the 66b input data stream is checked against the relevant polynomial according to the IEEE 802.3 specification. Bit errors are counted and are stored in the optional PRBS test pattern error counter that can be accessed through the APB.

You can enable pseudo-random test pattern checking (which is different from PRBS checking) by asserting `rx_tst_en`. The test pattern checker monitors the descrambler output and counts the number of data mismatches against the expected data pattern. The expected data pattern is either zero or local fault depending on `rx_tst_data_sel`.

Block errors are counted in the test pattern error counter that can be read through the APB.

The scrambled idles test pattern checking operates in a similar way to the pseudorandom test pattern checks, except that the checker expects a constant stream of encoded idle blocks.

Receiver Decoder

The 64/66B decoder decodes the 66b data into 64b data and generates the associated 8b control data. The decoded data and corresponding control bits are stored as a 72b vector (64b data + 8b control).

In the decode circuit, the combinatorial decode of the encoded data happens first and is dependent on the block-type field code that is present. The output of this stage sets the start, control, terminate, error, and data signals, and also assembles the decoded data with the appropriate control block codes.

The appropriate control bit is set to a logic 1 when the octet contains a control character, or to logic 0 for a data character. Reserved characters, low power idle, and signal and sequence ordered sets (including the auto-negotiation ordered set) are also valid input data to the decoder.

Auto-Negotiation Detector

This block detects auto-negotiation configuration ordered sets and signals to the auto-negotiation control state machine. Additionally, the module replaces the auto-negotiation ordered sets with IDLE going to the USXGMII (de)replication block.

Note: The 10GBase-KR block does not support the auto-negotiation function for backplane Ethernet, which is specified in Clause 73. It does support auto-negotiation for non-backplane Ethernet (Clause 37).

Replication

The receive replication block is used in USXGMII mode to sample the data on the receive path and remove replicated data. The module supports the following replication levels (per USXGMII specifications):

- *100 Mbps*—100x
- *1 Gbps*—10x
- *10 Gbps*—1x

Clock Tolerance Compensation

The clock tolerance compensation (CTC) module inserts or deletes IDLEs in the incoming data stream to compensate for the difference between two clocks within +/-100ppm. It can also delete the second sequence ordered set of two consecutive sequence ordered sets received from the incoming data stream.

The PCS uses a 64b datapath. However, the CTC only operates on 32b of data at a time, and only deletes or inserts 32b during a minimum IPG window (to ensure the MAC receives an IPG of at least five bytes when the PCS receives an IPG of nine bytes). A jumbo 16K frame is transmitted in approximately 2,048 cycles. 200 ppm is equivalent to a clock slip (equivalent to 64 bits) every 5,000 cycles, or a slip of 32b every 2,500 cycles, which is slightly more than the cycles needed to transmit the 16K packet. On average, expect the CTC to delete every 1.22 16K packets with a 200 ppm clock difference.

If an under/overflow condition is triggered, the CTC empties itself and recovers automatically.

Loopback

The TX datapath can be looped back to the RX datapath in two places:

- *MII Local Loopback*—Data provided to the MII transmit interface passes through a single sampling register and then loops back and outputs on the MII receive interface.
- *Post Scrambler Loopback*—The loopback occurs just before the TX SerDes gearbox. The data is fed back into the TX datapath just after the receive SerDes gearbox. This loopback allows you to exercise a very large portion of the datapaths.

USXGMII In-Band Control and Status Signals

The USXGMII specification defines a method to exchange link information based on a modification of the IEEE Std. 802.3 Clause 37 defined auto-negotiation state machine. This method uses XGMII ordered sets with a Cisco-specific opcode of `0x03`.

In the 10GBase-KR PCS this opcode is user programmable via the programming registers. The PCS supports autonomous hardware and software auto-negotiation modes as described in the USXGMII specification.

Hardware Auto-Negotiation

The 10GBase-KR PCS uses hardware auto-negotiation mode when the `usx_an_enable` field of the `pcsr_x_control_register` is set to `1'b1`. In this mode, the information exchange is autonomous, and programmable interrupts signal completion. Use this sequence:

1. Program `pcsr_x_usxgmii_link_timer_register` with the SerDes link rate and the required auto-negotiation link timeout value (1 - 2 ms per the USXGMII specification).
2. Program `pcsr_x_usxgmii_an_adv_register` fields with the link parameters according to the Cisco USXGMII specification.
3. (*MAC devices only*) enable the `usx_an_mirror_enable` field of `pcsr_x_control_register` to allow automatic mirroring of the link, duplex and speed fields of the negotiation message.
4. Unmask the `usxgmii_new_link_info` and `usxgmii_link_sts_upd` interrupts.
5. Enable auto-negotiation by setting the `usx_an_enable` field of `pcsr_x_control_register` to `1'b1`.
6. Wait for the interrupts to trigger. The first interrupt received is the `usxgmii_new_link_info` interrupt which occurs when a non-zero configuration word is received from the link partner. For MAC devices, do the following actions when you receive this interrupt:

   a) Optionally update the `pcsr_x_usxgmii_an_adv_register` if you are not using `usx_an_mirror_enable`.

   b) Program the `usx_speed` field of `pcsr_x_control_register` with the speed information sent by the link partner (read it in `pcsr_x_usxgmii_an_lp_register`).

7. When you receive the `usxgmii_link_sts_upd` interrupt, auto-negotiation is complete and normal data transmission/reception is possible.
8. Monitor the interrupts to be notified if/when auto-negotiation restarts.

Note: The 10GBase-KR block does not support the auto-negotiation function for backplane Ethernet, which is specified in Clause 73. It does support auto-negotiation for non-backplane Ethernet (Clause 37).

Software Auto-Negotiation

Software auto-negotiation ensures inter-operability with a wide range of devices. Enable this mode by setting:

- The `usx_an_enable` field of `pcsr_x_control_register` to `1'b0`.
- The `usx_an_tx_type` field to `2'b00`.

For software auto-negotiation, Efinix® recommends that you follow a similar methodology as the Clause 37 state machine:

1. Program `pcsr_x_usxgmii_an_adv_register` to `16'h0000`.
2. Optionally unmask the `usxgmii_new_link_info` interrupt.
3. Program these `pcsr_x_control_register` fields:
   - `usx_an_enable`—1'b0.
   - `usx_an_tx_type`—2'b01.
4. Wait for 1 - 2 ms to ensure that the link partner auto-negotiation state machine is restarted.
5. Program the `pcsr_x_usxgmii_an_adv_register` fields with the link parameters per the Cisco USXGMII specification, ensuring that the acknowledge field is set appropriately.
6. Wait for an interrupt to trigger or periodically poll `pcsr_x_usxgmii_an_lp_register` until the returned value is non-zero.
7. Update the `pcsr_x_usxgmii_an_adv_register` field and set the acknowledge bit.
8. Wait for 1 - 2 ms to ensure that the link partner properly recognizes the acknowledge.
9. Check that `pcsr_x_usxgmii_an_lp_register` has the acknowledge bit set. If it is not set, go back to step 1.
10. Program the `usx_an_tx_type` field to `2'b11` to transmit IDLEs to the link partner.
11. Wait 1 - 2 ms.
12. Program the `usx_an_tx_type` field to `2'b10` to allow transmission of XGMII data.

Software auto-negotiation is complete. Continue monitoring the interrupt to be notified when the link partner restarts auto-negotiation.

Note: The 10GBase-KR block does not support the auto-negotiation function for backplane Ethernet, which is specified in Clause 73. It does support auto-negotiation for non-backplane Ethernet (Clause 37).

Base-KR Training

You enable Base-KR link training with the `KR_TRAINING_ENABLE` and `KR_RESTART_TRAINING` PHY inputs.

- Initiate training at start-of-day by asserting `KR_TRAINING_ENABLE` prior to enabling the link.
- After initial start-up, initiate training by asserting `KR_TRAINING_ENABLE` high followed by strobing `KR_RESTART_TRAINING` high for a mininum of 100 μs. `KR_RESTART_TRAINING` is an active-high reset to the 10G-KR training logic.

Assertion of `KR_SIGNAL_DETECT` indicates successful link training.

Assertion of `KR_TRAINING_FAILURE` indicates failed link training.

`KR_TRAINING`, `KR_FRAME_LOCK`, and `KR_LOCAL_RX_TRAINED` provide status information on the state of the KR training process and upon failure an indication of where the failure occurred.

Figure 5: BASE-KR Training Success Diagram

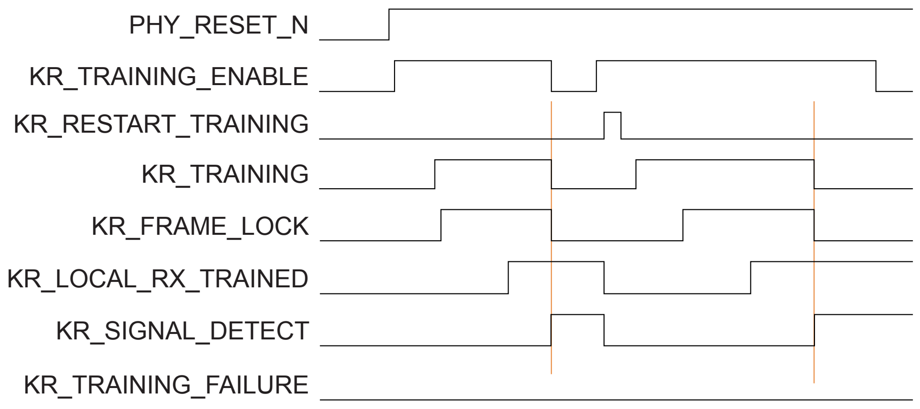
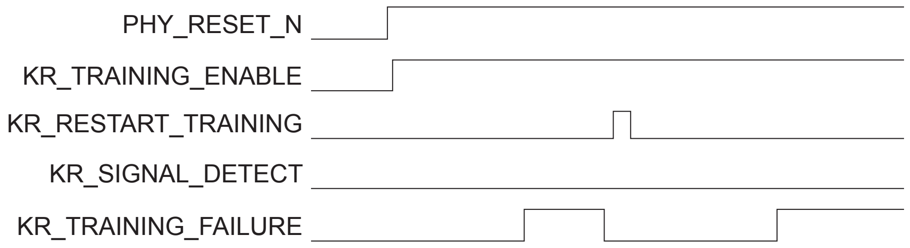


Figure 6: BASE-KR Training Failure Diagram

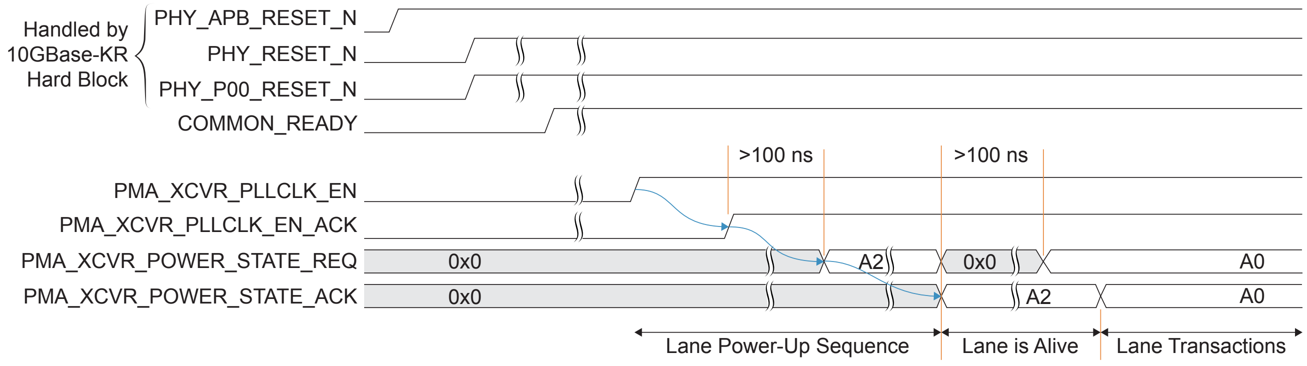


Power Up Sequence

Initially, the 10GBase-KR reset controller controls the `PHY_APB_RESET_N` and `PHY_RESET_N` signals. The PHY reset signals are handed over to the client after `COMMON_READY` is asserted and the soft logic enters user mode. The client needs to drive the PHY reset signals high in the initial state so that the power up sequence is not impacted.

Figure 7: Power-Up Sequence

(*m* is 0, 1, 2, or 3)



When `COMMON_READY` is asserted:

- Set `PMA_XCVR_POWER_STATE_REQ` to `0x0`
- Assert `PMA_XCVR_PLLCLK_EN`

When `PMA_XCVR_PLLCLK_EN_ACK` is asserted, set `PMA_XCVR_POWER_STATE_REQ` to `A2`.

There is a 100 ns (minimum) delay between the assertion of `PMA_XCVR_PLLCLK_EN_ACK` and when you can set `PMA_XCVR_POWER_STATE_REQ` to `A2`.

To start RX operation, the client monitors the assertion of the PHY's `RX_SIGNAL_DETECT` and waits for $t_{rx\_cr\_ceinit}$ or $t_{rx\_cr\_noinit}$ before asserting `SIGNAL_OK` in `control_register`.

Figure 8: Asserting signal_ok

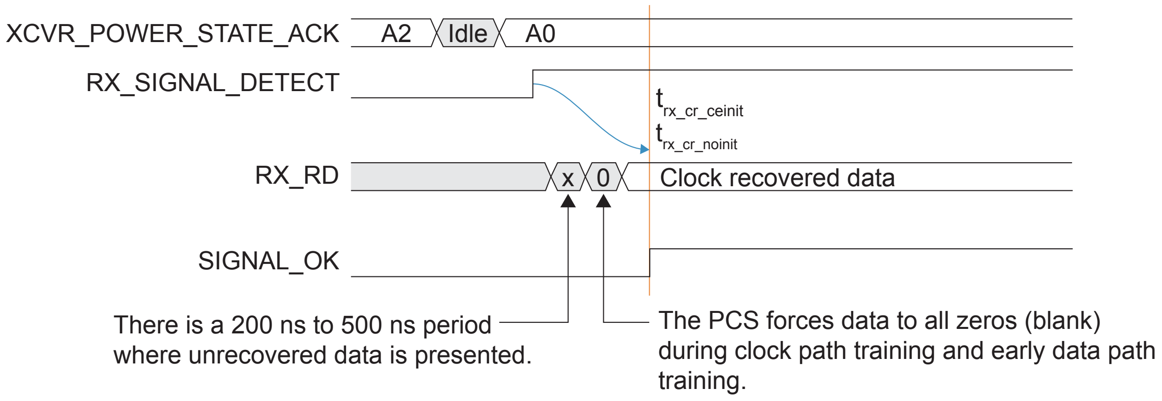

XCVR_POWER_STATE_ACK
A2
Idle
A0
RX_SIGNAL_DETECT
t rx_cr_ceinit
t rx_cr_noinit
RX_RD
x
0
Clock recovered data
SIGNAL_OK
There is a 200 ns to 500 ns period where unrecovered data is presented.
The PCS forces data to all zeros (blank) during clock path training and early data path training.

Table 3: SIGNAL_OK Timing Parameters

Timing Parameters	Minimum	Maximum	Description
$t_{rx\_cr\_ceinit}$	22.3 µs	117.4 µs	Initial time required to lock clock recovery once valid data is received.
$t_{rx\_cr\_noinit}$	428 ns	593 ns	Time required to lock clock recovery once valid data is received, assuming initial adaptation has been previously completed.

Signals

In the Efinity Interface Designer, signals are prefixed with a user-defined instance name. Efinix recommends using an instance name with the format Q*n*_L*m* (where *n* is the quad number and *m* is the lane number) for easier identification.

Table 4: Signals Per Lane

Signal	Direction	Clock Domain	Description
TXD_[63:0]	Input	PCS_CLK_TX	Transmit data.
TXC_[7:0]	Input	PCS_CLK_TX	Transmit control.
RXD_[63:0]	Output	PCS_CLK_TX	Receive data.
RXC_[7:0]	Output	PCS_CLK_TX	Receive control.
PCS_CLK_TX	Input	N/A	Interface transmitter clock.
PCS_CLK_RX	Input	N/A	Interface receiver clock.
PCS_RST_N_TX	Input	Asynchronous	PCS TX reset.
PCS_RST_N_RX	Input	Asynchronous	PCS RX reset.
IRQ	Output	APB_CLK	Interrupt, level sensitive.
BLOCK_LOCK	Output	PCS_CLK_TX	Indicates that block lock has been achieved either through the FEC decoding or the standard RX sync process. In addition, if USXGMII auto-negotiation is enabled, this output only goes high after auto-negotiation completes.
HI_BER	Output	PCS_CLK_TX	Indicates High Bit Error Ratio (BER) status.
PCS_STATUS	Output	PCS_CLK_TX	General PCS ready status. Connect to other blocks for status reporting, e.g., backplane Ethernet auto-negotiation. This signal is defined in IEEE Std. 802.3 Clause 49 as block_lock (not hi_ber).
TX_FWD_CLK	Output	N/A	Soft MAC clock resource. Both pcs_clk_tx and pcs_clk_rx need to connect to this clock source. 10 M: 1.5625 MHz 1 G: 15.625 MHz 10 G: 156.25 MHz
PHY_RESET_N	Input	Asynchronous	PHY per-lane reset. The user application should initialize it to 1.

Signal	Direction	Clock Domain	Description
PMA_TX_ELEC_IDLE	Input	Asynchronous	PMA TX electrical idle. 1: TX lines placed into electrical idle state 0: Transmit data
ETH_EEE_ALERT_EN	Input	Asynchronous	Energy Efficient Ethernet (EEE) alert signaling enable. Selects the preset transmitter de-emphasis setting instead of the trained/equalized value. Asserted high when transmitting the alert signal during EEE operation. Only applies when phy_l*m*_eth_mode == 1 (i.e., only valid for 10G-KR operation). If EEE is unused, tie low.
KR_RESTART_TRAINING	Input	Asynchronous	Restart link training. Strobed high (100 ns minimum) to re-initiate the 10G-KR training process after the process has completed or after training was initially disabled or bypassed. Assert kr_training_enable_ln_*m* high prior to de-asserting kr_resetart_training_ln_*m* for the training process to trigger successfully. Operates as an active-high reset to the 10G-KR training logic for the associated lane.
KR_TRAINING_ENABLE	Input	Asynchronous	Link training enable. When high, enables the 10G-KR training process immediately upon the PMA becoming ready after the link is enabled. Must be asserted prior to de-assertion of reset and remain asserted until either kr_training_ln_*m* or kr_training_failure_ln_*m* is asserted high.
KR_FRAME_LOCK	Output	Asynchronous	10G-KR frame locked, active hig. Indicates that the 10G-KR training process is receiving 10G-KR training frame headers successfully, headers are spaced appropriately, and it is acquiring accurate link training update and status information from the remote link partner.
KR_LOCAL_RX_TRAINED	Output	Asynchronous	10G-KR receiver trained, active high. Indicates that the local receiver has finished evaluation and adjustment of remote transmitter de-emphasis coefficients as part of the 10G-KR training process.
KR_SIGNAL_DETECT	Output	Asynchronous	10G-KR training signal detect, active high. Indicates either a normal completion of the 10G-KR training process, or, if the training process is disabled/bypassed, that the initial transmitter de-emphasis coefficients have been applied.

Signal	Direction	Clock Domain	Description
KR_TRAINING_FAILURE	Output	Asynchronous	Link training failure, active high. Indicates that the maximum time limit (500 ms) allotted for 10G-KR link training was reached without successful convergence of the local and/ or remote equalization algorithms. You can determine the which end of the link inhibited this process by observing the value of kr_local_rx_trained_ln_*m*. If deasserted, the issue lies between the remote transmitter and the local receiver. If asserted, the issue lies between the local transmitter and remote receiver.
KR_TRAINING	Output	Asynchronous	10G-KR training, active high. Indicates that a given lane has applied the initial transmitter de-emphasis coefficient values and is actively training the 10G-KR link.
PMA_XCVR_PLLCLK_EN	Input	Asynchronous	Link PLL clock enable. This signal cleanly gates the pma_pllck_datart_ln_*m* and pma_pllclk_fullrt_ln_*m* clocks for the associated link/port.
PMA_XCVR_PLLCLK_EN_ACK	Output	Asynchronous	Link PLL clock enable acknowledgment:. This signal indicates whether the pma_pllclk_datart_ln_*m* and pma_pllclk_-fullrt_ln_*m* for the associated link/port is running or not.
PMA_XCVR_POWER_ STATE_REQ[3:0]	Input	Asynchronous	Link power state request. This signal changes the raw SerDes link/port's power state. When the link/port has completed the transition to the requested power state, the requested state is reflected on pma_xcvr_power_state_ack_p_*m*. 4'b0000: Idle 4'b0001: A0 - TX/RR active 4'b0010: A1 - Powerdown 1 (low power state with minimum exit latency) 4'b0100: A2 - Powerdown 2 (lower power state with longer exit latency as compared to A1) 4'b1000: A3 - Powerdown 3 (lower power state and longer exit latency as compared to A2) This signal is one hot encoded. A subsequent change request is not signaled until the current request has been acknowledged and pma_xcvr_power_state_req_p_*m* has returned to 0. Upon reset release, the first power state must be A2.

Signal	Direction	Clock Domain	Description
PMA_XCVR_POWER_ STATE_ACK[3:0]	Output	Asynchronous	Link power state acknowledgment. This signal indicates that a power state change request has completed. 4’b0000: Value after reset, prior to first power state request 4’b0001: A0 4’b0010: A1 4’b0100: A2 4’b1000: A3 Once a power state is acknowledged, the value remains unchanged until a new power state is requested and the link has completed the transition to the new power state.
PMA_RX_SIGNAL_DETECT	Output	Asynchronous	PMA receiver signal detect. Asserted high upon detection of a high-speed signal on the RX differential inputs.

Table 5: Common Signals Used for All Lanes in a Quad

Signal	Direction	Clock Domain	Description
APB_CLK	Input	N/A	APB clock source. 200 Mhz maximum.
USER_APB_PADDR[23:0]	Input	APB_CLK	APB address.
USER_APB_PSEL	Input	APB_CLK	APB select.
USER_APB_PENABLE	Input	APB_CLK	APB enable.
USER_APB_PWRITE	Input	APB_CLK	APB write.
USER_APB_PWDATA[31:0]	Input	APB_CLK	APB write data.
USER_APB_PRDATA[31:0]	Output	APB_CLK	APB read data.
USER_APB_PREADY	Output	APB_CLK	APB ready.
PMA_CMN_READY	Output	Asynchronous	PHY ready.

Register Map

The following tables show the PCS registers.

Table 6: Register Map

Register Name	Type	Reset Value	Address Offset[7:0]
control_register	RW	0x0031 1004	0x00
pcsr_test_control_register	RW	0x0000 0000	0x04
status_register	RW	0x0000 0000	0x08
designcfg_register	RO	0x0000 005F	0x0C
test_seed_a_lower	RW	0x0000 0000	0x10
test_seed_a_upper	RW	0x0000 0000	0x14
test_seed_b_lower	RW	0x0000 0000	0x18
test_seed_b_upper	RW	0x0000 0000	0x1C
rx_decoder_error_counter	RW	0x0000 0000	0x20
bit_error_counter	RW	0x0000 0000	0x24
test_pattern_error_counter	RW	0x0000 0000	0x28
prbs_error_counter	RW	0x0000 0000	0x2C
fec_corr_error_counter	RW	0x0000 0000	0x50
fec_uncorr_error_counter	RW	0x0000 0000	0x54
interrupt_status_register	RW	0x0000 0000	0x60
interrupt_enable_register	RW	0x0000 0000	0x64
interrupt_disable_register	RW	0x0000 0000	0x68
interrupt_mask_register	RO	0x0311 010A	0x6C
usxgmii_link_timer_register	RW	0x000A 3D09	0x70
usxgmii_an_adv_register	RW	0x0000 0000	0x74
usxgmii_an_lp_register	RW	0x0000 0000	0x78
revision_register	RO	0x0380 0100	0x7C

Table 7: 10G PCS Access

10G PCS access	Value
user_apb_paddr[23:21]	3’b110
user_apb_paddr[20:11]	X (don’t care)
user_apb_paddr[10:8]	3’b000: Lane 0 3’b001: Lane 1 3’b010: Lane 2 3’b011: Lane 3

The following tables show the bit descriptions for the PCS registers.

Table 8: control_register

Bit	Name	Description	Type	Reset
31	usx_an_enable	USXGMII auto-negotiation enable. Set to 1'b1 to enable USXGMII hardware auto-negotiation state machine. Set to 1'b0 for non-USXGMII operation or if the application requires software controlled negotiation.	RW	0
30	usx_an_restart	USXGMII auto-negotiation restart. Write 1'b1 to trigger restart of the USXGMII hardware auto-negotiation process. Bit 31 must also be set for auto-negotiation to function. The hardware performs a rising-edge detect on this field, therefore, software should always set this register field to 0 after writing.	RW	0
29:28	usx_an_tx_type	USXGMII auto-negotiation transmit data type. This field is used for USXGMII software managed auto-negotiation and is only applicable if usx_an_enable is set to 1'b0. 2'b00: Use transmit data from the MAC interface and do not continue to monitor for new link information. 2'b01: Transmit auto-negotiation ordered sets using the value in usx_an_adv register. 2'b10: Use transmit data from the MAC interface and continue monitoring for new link information. 2'b11: Transmit a stream of IDLEs.	RW	0
27:20	usx_an_os_code	USXGMII auto-negotiation ordered set code. Set the ordered set code field to use when transmitting and detecting auto-negotiation ordered sets. The default value is 8'h03, which is the value defined in the Cisco specification. Only change the value when the RX and TX datapaths are disabled.	RW	0x03
19	usx_an_mirror_enable	USXGMII auto-negotiation auto mirror link info. When set to 1'b1 and hardware auto-negotiation is enabled, the transmitted ability values in the ACK_DET state are taken from the received abilities from the link partner. This function applies to all fields except for EEE capability fields. This function is a debug feature that can be useful for MAC devices. Do not set it for PHY devices.	RW	0
18:17	reserved	Reserved.	RO	0
16:14	usx_speed	USXGMII speed. These bits and the serdes_rate field determine the amount of replication performed to obtain the desired sub-rate. The USXGMII speed must match the data rate of the MAC device in the SoC. 000: 100 Mbps 001: 1 Gbps 100: 10 Gbps Other: reserved Only change the value when the TX and RX datapaths are disabled.	RW	0x4

Bit	Name	Description	Type	Reset
13:12	serdes_rate	SerDes line rate. These bits control how symbols are repeated for USXGMII operation. The following values are valid: 01: 10.3125 Gbps Others - reserved Only change the value when the TX and RX datapaths are disabled.	RW	0x1
11	rx_pol_invert	RX polarity invert. Set high to invert the incoming RX data.	RW	0
10	tx_pol_invert	TX polarity invert. Set high to invert the TX data.	RW	0
9	rx_scr_bypass	RX scrambler bypass. Set high to bypass the RX descrambler. Only change the value when the RX datapath is disabled.	RW	0
8	tx_scr_bypass	TX scrambler bypass. Set high to bypass the TX scrambler. Only change the value when the TX datapath is disabled.	RW	0
7:6	reserved	Reserved.	RO	0
5	fec_err_ind	FEC error forwarding. When high and FEC mode is enabled, upon detection of uncorrectable errors, 66-bit blocks within the errored FEC block are marked as errored as per the IEEE Std. 802.3 Clause 74 to ensure that affected packets to the MAC are corrupted. The configuration option for this feature must be enabled at compile time; when enabled it significantly increases the RX path latency. When low, uncorrectable FEC blocks have no effect on PCS sync headers. Only change the value when the RX datapath is disabled.	RW	0
4	fec_enable	FEC mode enable. 1: FEC mode is enabled. 0: FEC mode is disabled. Only change the value when the TX and RX datapaths are disabled.	RW	0
3	reserved	Reserved.	RO	0
2	rx_sync_reset	RX reset. 1: Reset the receive datapath. 0: RX datapath is enabled.	RW	1
1	tx_datapath_en	TX datapath enable. Drives the tx_datapath_en signal to the TX. 1: Enable the TX datapath. 0: TX datapath is disabled.	RW	0
0	signal_ok	Enable the USXGMII/BASE-R receive PCS. Drives the signal_ok signal to the RX. If this bit is low the RX is in reset unless the post scrambler loopback mode is enabled. This bit is reset low using a hardware reset. Do not set it high until the external SerDes is supplying a suitable recovered receive clock.	RW	0

Table 9: pcsr_test_control_register

Bit	Name	Description	Type	Reset
31:22	reserved	Reserved.	RO	0
21	rx_prbs31_en	Receive PRBS 31 enable. Check for PRBS 31 test pattern.	RW	0
20	rx_prbs9_en	Receive PRBS 9 enable. Check for PRBS 9 test pattern.	RW	0
19	reserved	Reserved.	RO	0
18	rx_tst_dat_sel	Receive test data select. 1: Check for pseudo random zero patterns; 0: Check for pseudo random local fault test patterns.	RW	0
17	rx_scr_idle_en	Receive scrambled idle enable. Check for scrambled idle test pattern.	RW	0
16	rx_tst_en	Receive test enable. Enables receive test pattern checking.	RW	0
15:13	reserved	Reserved.	RO	0
12	tx_sqw_en	Transmit square wave enable. Set before enabling test pattern transmission. 1: Square wave. 0: Pseudo-random.	RW	0
11:10	reserved	Reserved.	RO	0
9	tx_prbs31_en	Transmit PRBS 31 enable. Selects PRBS 31 transmit test pattern.	RW	0
8	tx_prbs9_en	Transmit PRBS 9 enable. Selects PRBS 9 transmit test pattern.	RW	0
7	reserved	Reserved.	RO	0
6	tx_tst_dat_sel	Transmit test data select. Only used for pseudo random test mode 1: Zero pattern sent to test generator 0: Local fault test data.	RW	0
5	tx_scr_idle_en	Transmit scrambled idle enable. Selects scrambled idle test pattern.	RW	0
4	tx_tst_en	Transmit test enable. Enable transmit test pattern transmission.	RW	0
3:2	reserved	Reserved.	RO	0
1	scr_lpbk_en	TX-RX loopback at scrambler. Enables loopback at embedded scrambler; drives scr_lpbk_en and scr_lpbk_clk_ctrl.	RW	0
0	mii_lpbk_en	TX-RX loopback at MII. Enbles loopback at MII; drives mii_lpbk_en and mii_lpbk_clk_ctrl.	RW	0

Table 10: status_register

Bit	Name	Description	Type	Reset
31	ctc_o_u_flow	CTC FIFO overflow/underflow. Indicates the internal clock tolerance compensation FIFOs have overflowed or underflowed.	RW W1toClr	0
30	reserved	Reserved.	RO	0
29	hi_bit_error	Hi BER. High bit error ratio detected.	RW W1toClr	0
28	tx_fault	TX fault. The TX encoder state machine has entered the error state.	RW W1toClr	0
27	rx_fault	RX fault. the RX decoder state machine has entered the error state.	RW W1toClr	0
26:2	reserved	Reserved.	RO	0
1	an_complete	USXGMII auto-negotiation complete. When the USXGMII hardware auto-negotiation feature is enabled, this field indicates the negotiation status. When set to 1'b1, auto-negotiation has completed and the value received from the link partner can be obtained from usxgmii_an_lp_register.	RO	0
0	block_lock	Block lock. A one indicates that the USXGMII/10GBASE-R PCS has achieved block synchronization.	RO	0

Table 11: Test_seed_a_lower

Bits	Name	Description	Type	Reset
31:0	seed_a_low	Test pattern seed A bits 31:0 for pseudo random counter transmission. When in PRBS mode, the scrambler seed is loaded every 128 blocks with a repeating pattern of: seed_a, seed_a_invert, seed_b, seed_b_invert.	RW	0

Table 12: Test_seed_a_upper

Bit	Name	Description	Type	Reset
31:26	reserved	Reserved.	RO	0
25:0	seed_a_upper	Test pattern seed A bits 57:32 for pseudo random counter transmission. When in PRBS mode, the scrambler seed is loaded every 128 blocks with a repeating pattern of: seed_a, seed_a_invert, seed_b, seed_b_invert.	RW	0

Table 13: Test_seed_b_lower

Bit	Name	Description	Type	Reset
31:0	seed_b_low	Test pattern seed B bits 31:0 for pseudo random counter transmission. When in PRBS mode, the scrambler seed is loaded every 128 blocks with a repeating pattern of: seed_a, seed_a_invert, seed_b, seed_b_invert.	RW	0

Table 14: Test_seed_b_upper

Bit	Name	Description	Type	Reset
31:26	reserved	Reserved.	RO	0
25:0	seed_b_upper	Test pattern seed B bits 57:32 for pseudo random counter transmission. When in PRBS mode, the scrambler seed is loaded every 128 blocks with a repeating pattern of: seed_a, seed_a_invert, seed_b, seed_b_invert.	RW	0

Table 15: rx_decoder_error_counter

Bit	Name	Description	Type	Reset
31:16	reserved	Reserved.	RO	0
15:0	block_error_count	Block error count from RX decoder. Writing any value to this register clears this field. Due to internal clock synchronization, it may take a little time for the clear to propagate and take effect.	RW W1toClr	0

Table 16: bit_error_counter

Bit	Name	Description	Type	Reset
31:16	reserved	Reserved	RO	0
15:0	bit_error_count	Bit error count from BER monitor; count of errors in the synchronization bits. Per the IEEE Std. 802.3 specification, if hi_ber is high, all errors may not be counted because a maximum of 16 errors can be counted in any 125 µs window for 10 Gbps. Writing any value to this register clears this field. Due to internal clock synchronization, it may take a little time for the clear to propagate and take effect.	RW W1toClr	0

Table 17: test_pattern_error_counter

Bit	Name	Description	Type	Reset
31:16	reserved	Reserved.	RO	0
15:0	test_error_count	Test pattern checker error count (scrambled idle errors if rx_scr_idle_en is set and pseudo random test pattern errors if rx_scr_idle_en is clear. Writing any value to this register clears this field. Due to internal clock synchronization, it may take a little time for the clear to propagate and take effect.	RW W1toClr	0

Table 18: prbs_error_counter

Bit	Name	Description	Type	Reset
31:16	reserved	Reserved.	RO	0
15:0	prbs_rx_error_count	PRBS RX pattern checking error count. Writing any value to this register clears this field. Due to internal clock synchronization, it may take a little time for the clear to propagate and take effect.	RW W1toClr	0

Table 19: fec_corr_error_counter

Bit	Name	Description	Type	Reset
31:0	fec_block_corrected_count	Number of FEC blocks received that were detected as errored and were corrected. Writing any value to this register clears this field. Due to internal clock synchronization, it may take a little time for the clear to propagate and take effect.	RW W1toClr	0

Table 20: fec_uncorr_error_counter

Bit	Name	Description	Type	Reset
31:0	fec_block_error_count	Number of FEC blocks received that were detected as errored and were uncorrectable. Writing any value to this register clears this field. Due to internal clock synchronization, it may take a little time for the clear to propagate and take effect.	RW W1toClr	0

Table 21: interrupt_status_register

Bit	Name	Description	Type	Reset
31:26	reserved	Reserved.	RO	0
25	usxgmii_new_link_info	USXGMII new link information seen.	RW W1toClr	0
24	usxgmii_link_sts_upd	USXGMII link status update complete.	RW W1toClr	0
23:21	reserved	Reserved.	RO	0
20	fec_correctable_error	FEC correctable error occurred.	RW W1toClr	0
19:17	reserved	Reserved.	RO	0
16	fec_uncorrectable_error	FEC uncorrectable error occurred.	RW W1toClr	0
15:9	reserved	Reserved.	RO	0
8	block_locked	Block lock status change.	RW W1toClr	0
7:4	reserved	Reserved.	RO	0
3	hi_bit_error	High bit error status triggered.	RW W1toClr	0
2	reserved	Reserved.	RO	0
1	buffer_error	Elastic buffer error occurred.	RW W1toClr	0
0	reserved	Reserved.	RO	0

Table 22: interrupt_enable_register

Bit	Name	Description	Type	Reset
31:26	reserved	Reserved.	RO	0
25	usxgmii_new_link_info_en	USXGMII new link information seen enable.	WO	0
24	usxgmii_link_sts_upd_en	USXGMII link status update complete enable.	WO	0
23:21	reserved	Reserved.	RO	0
20	fec_correctable_error_en	FEC correctable error occurred enable.	WO	0
19:17	reserved	Reserved.	RO	0
16	fec_uncorrectable_error_en	FEC uncorrectable error occurred enable.	WO	0
15:9	reserved	Reserved	RO	0
8	block_locked_en	Block lock status change enable.	WO	0
7:4	reserved	Reserved.	RO	0
3	hi_bit_error_en	High bit error status triggered enable.	WO	0
2	reserved	Reserved.	RO	0
1	buffer_error_en	Elastic buffer error occurred enable.	WO	0
0	reserved	Reserved.	RO	0

Table 23: interrupt_disable_register

Bit	Name	Description	Type	Reset
31:26	reserved	Reserved.	RO	0
25	usxgmii_new_link_info_dis	USXGMII new link information seen disable.	WO	0
24	usxgmii_link_sts_upd_dis	USXGMII link status update complete disable.	WO	0
23:21	reserved	Reserved.	RO	0
20	fec_correctable_error_dis	FEC correctable error occurred disable.	WO	0
19:17	reserved	Reserved.	RO	0
16	fec_uncorrectable_error_dis	FEC uncorrectable error occurred disable.	WO	0
15:9	reserved	Reserved.	RO	0
8	block_locked_dis	Block lock status change disable.	WO	0
7:4	reserved	Reserved.	RO	0
3	hi_bit_error_dis	High bit error status triggered disable.	WO	0
2	reserved	Reserved.	RO	0
1	buffer_error_dis	Elastic buffer error occurred disable.	WO	0
0	reserved	Reserved.	RO	0

Table 24: interrupt_mask_register

Bit	Name	Description	Type	Reset
31:26	reserved	Reserved.	RO	0
25	usxgmii_new_link_info_mask	USXGMII new link information seen mask.	RO	1
24	usxgmii_link_sts_upd_mask	USXGMII link status update complete mask.	RO	1
23:21	reserved	Reserved.	RO	0
20	fec_correctable_error_mask	FEC correctable error occurred mask.	RO	1
19:17	reserved	Reserved.	RO	0
16	fec_uncorrectable_error_mask	FEC uncorrectable error occurred mask.	RO	1
15:9	reserved	Reserved.	RO	0
8	block_locked_mask	Block lock status change mask.	RO	1
7:4	reserved	Reserved.	RO	0
3	hi_bit_error_mask	High bit error status triggered mask.	RO	1
2	reserved	Reserved.	RO	0
1	buffer_error_mask	Elastic buffer error occurred mask.	RO	1
0	reserved	Reserved.	RO	0

Table 25: usxgmii_link_timer_register

Bit	Name	Description	Type	Reset
31:21	reserved	Reserved.	RO	0
20:16	usx_link_tim	Link timer value to use when USXGMII hardware auto-negotiation is enabled. USXGMII specifies a link timer range of 1-2 ms adjustable in steps of 0.1 ms. This register field is used with the prescale field to determine the step granularity and the number of steps. Set the prescale field to provide a granularity of 0.1 ms, taking into account the SerDes link rate. Only change this field value when the RX datapath is disabled.	RW	0x0A
15:14	reserved	Reserved.	RO	0
13:0	usx_link_tim_prescale	Link timer prescaler value. Set this field to represent 0.1 ms, taking into account the main datapath frequency (156.25 MHz for a 10 Gbps SerDes. Multiply this frequency by 100 to get the number of clock cycles required to represent 0.1 ms. The default value is set for a 10 Gbps link. Only change this field value when the RX datapath is disabled.	RW	0x3D09

Table 26: usxgmii_an_adv_regsiter

Bit	Name	Description	Type	Reset
31:16	reserved	Reserved.	RO	0
15:0	usx_an_adv	USXGMII auto-negotiation base page advertisement value. This field's value is used when auto-negotiation ordered sets are transmitted (hardware or software controlled). For hardware auto-negotiation, the internal state machine automatically sets the ACK bit if it is not already set in this field. Refer to the Cisco specification for more details on the valid settings for this field.	RW	0

Table 27: usxgmii_an_lp_register

Bit	Name	Description	Type	Reset
31:16	reserved	Reserved	RO	0
15:0	usx_an_lp_adv	USXGMII auto-negotiation link partner base page value. This field's value indicates the configuration word received from the link partner at the end of auto-negotiation when hardware managed negotiation is enabled, or while software-managed negotiation is in process.	RO	0x0

Table 28: revision_register

Bit	Name	Description	Type	Reset
31:28	fix_number	Fix number; incremented for fix releases.	RO	0
27:16	module_identification_number	PCS module identification number; fixed value.	RO	0x380
15:0	module_revision	Module revision. Fixed value specific to the PCS revision that is incremented for each non-fixed release.	RO	0x0100

Revision History

Table 29: Document Revision History

Date	Version	Description
November 2024	1.1	The 10GBase-KR transceiver does not support auto-negotiation for backplane Ethernet (Clause 73). DOC-2191
June 2024	1.0	Initial release.