5. XAUI Mode

SGX52005-1.2

Introduction

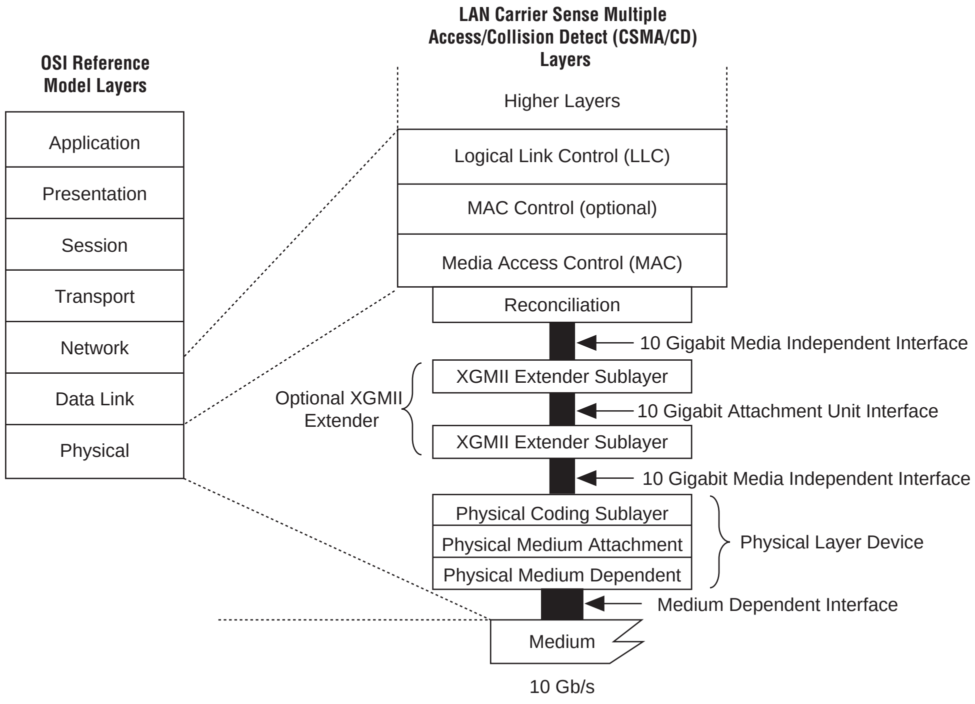
The 10 Gigabit Attachment Unit Interface (XAUI) is an optional, self-managed interface that can be inserted between the reconciliation sublayer and the PHY layer to transparently extend the physical reach of the 10 Gigabit Media Independent Interface (XGMII).

XAUI addresses several physical limitations of the XGMII. XGMII signaling is based on the HSTL class 1 single-ended I/O standard, which has an electrical distance limitation of approximately 7 cm. Because XAUI uses low voltage differential signaling method, the electrical limitation is increased to approximately 50 cm. Another advantage of XAUI is simplification of backplane and board trace routing. XGMII is composed of 32 transmit channels, 32 receive channels, 1 transmit clock, 1 receive clock, 4 transmitter control characters, and 4 receive control characters for a 74-pin wide interface in total. XAUI, on the other hand, only consists of 4 differential transmitter channels and 4 differential receiver channels for a 16-pin wide interface in total. This reduction in pin count significantly simplifies the routing process in the layout design. Figure 5–1 shows the relationships between the XGMII and XAUI layers.

Stratix® GX devices offer the following XAUI features:

- Serial data rate range from 500 Mbps to 3.1875 Gbps
- Input reference clock range from 25 to 637.5 MHz
- Parallel interface width of 16 bits
- 8B/10B encoder and decoder
- Word aligner supports 10-bit code-group
- Channel deskew
- Rate compensation or elastic buffer
- XGMII-to -PCS code conversion on transmit
- PCS-to -XGMII code conversion on receive
- Byte deserializer

Figure 5–1. XGMII & XAUI Relationship to ISO/IEC Open Systems Interconnection (OSI) Reference Model & IEEE 802.3 CSMA/CD LAN Model


OSI Reference Model Layers
Application
Presentation
Session
Transport
Network
Data Link
Physical
LAN Carrier Sense Multiple Access/Collision Detect (CSMA/CD) Layers
Higher Layers
Logical Link Control (LLC)
MAC Control (optional)
Media Access Control (MAC)
Reconciliation
10 Gigabit Media Independent Interface
XGMII Extender Sublayer
10 Gigabit Attachment Unit Interface
XGMII Extender Sublayer
Optional XGMII Extender
10 Gigabit Media Independent Interface
Physical Coding Sublayer
Physical Medium Attachment
Physical Medium Dependent
Physical Layer Device
Medium Dependent Interface
Medium
10 Gb/s

As noted earlier, the XGMII interface consists of 4 lanes of 8 bits. At the transmit side of the XAUI interface, the data and control characters are converted within the XGXS into an 8B/10B encoded data stream. Each data stream is then transmitted across a single differential pair running at 3.125 Gbps. At the XAUI receiver, the incoming data is decoded and mapped back to the 32 bit XGMII format. This process provides a transparent extension of the physical reach of the XGMII and also reduces the interface pin count.

XAUI functions as a self-managed interface because code-group synchronization, channel deskew, and clock domain decoupling are handled with no upper layer support requirements. These features are accomplished based on Physical Coding Sublayer (PCS) code-groups that are used during the Inter-Packet Gap (IPG) time and during idle periods. PCS code-groups are mapped by the XGMII Extender Sublayer (XGXS) to XGMII characters, as specified in Table 5–1.

Table 5–1. XGMII Character to PCS Code-Group Mapping *Note (1)*

XGMII TXC	XGMII TXD	PCD Code-Group	Description
0	00 through FF	Dxx,y	Normal data transmission
1	07	K28.0 or K28.3 or K28.5	Idle in ||**I**||
1	07	K28.5	Idle in ||**T**||
1	9C	K28.4	Sequence
1	FB	K27.7	Start
1	FD	K29.7	Terminate
1	FE	K30.7	Error
1	Any other value	K30.7	Invalid XGMII character

Note to Table 5–1:
(1) Values in TXD column are in hexadecimal.

Figure 5–2 shows an example of the mapping between XGMII characters to the PCS code-groups used in XAUI. The idle characters are mapped to a pseudorandom sequence of ||A||, ||R||, and ||K|| code-groups.

Figure 5–2. Example of Mapping XGMII Characters to PCS Code-Groups

XGMII

T/RXD<7..0>	I	I	S	Dp	D	D	D	---	D	D	D	D	I	I	I	I	I	I
T/RXD<15..8>	I	I	Dp	Dp	D	D	D	---	D	D	D	T	I	I	I	I	I	I
T/RXD<23..16>	I	I	Dp	Dp	D	D	D	---	D	D	D	I	I	I	I	I	I	I
T/RXD<31..24>	I	I	Dp	Ds	D	D	D	---	D	D	D	I	I	I	I	I	I	I

PCS

Lane 0	K	R	S	Dp	D	D	D	---	D	D	D	D	A	R	R	K	K	R
Lane 1	K	R	Dp	Dp	D	D	D	---	D	D	D	T	A	R	R	K	K	R
Lane 2	K	R	Dp	Dp	D	D	D	---	D	D	D	K	A	R	R	K	K	R
Lane 3	K	R	Dp	Ds	D	D	D	---	D	D	D	K	A	R	R	K	K	R

The PCS code-groups are sent via PCS ordered sets. PCS ordered sets consist of combinations of special and data code-groups defined as a column of code-groups. These ordered sets are composed of four code-groups beginning in lane 0. Table 5–2 lists the defined idle ordered sets (||I||) that are used for the self managed properties of XAUI.

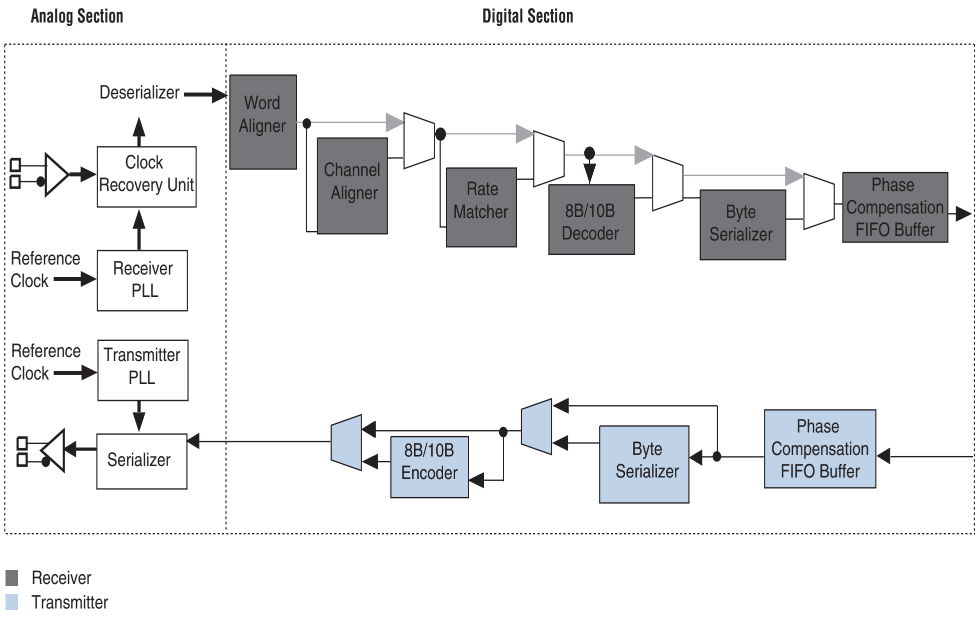
Table 5–2. Defined Idle Ordered Set

Code	Ordered_Set	Number of Code_Groups	Encoding
`||I|`	Idle		Substitute for XGMII Idle
`||K||`	Sync column	4	`/K28.5/K28.5/K28.5/K28.5/`
`||R||`	Skip column	4	`/K28.0/K28.0/K28.0/K28.0/`
`||A||`	Align column	4	`/K28.3/K28.3/K28.3/K28.3/`

This section briefly introduces XAUI, along with the code-groups and ordered sets associated with this self-managed interface. For full details on the XAUI standard, refer to clauses 47–48 in the 10 Gigabit Ethernet standard (IEEE 802.3ae).

XAUI mode enables the configuring of transceiver blocks to support XAUI synchronization, channel alignment, rate compensation, XGMII to PCS code-group conversion, and PCS code-group-to-XGMII conversion. This section describes the supported digital architecture, clocking schemes, and software implementation of the XAUI mode. Figure 5–3 shows a block diagram of a duplex channel configured in XAUI mode.

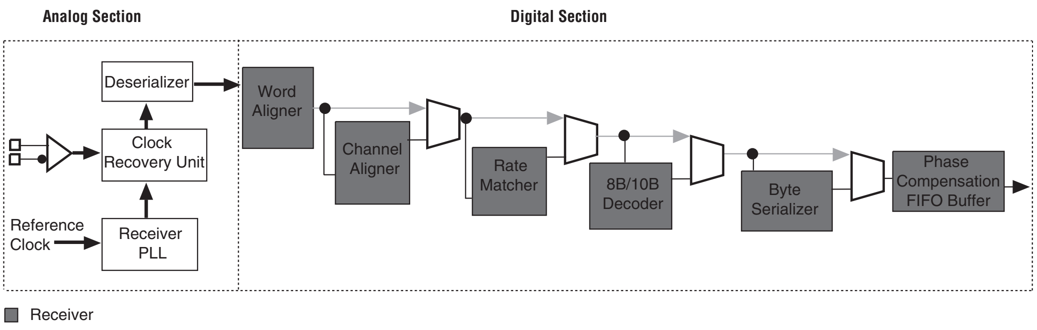
Figure 5–3. Block Diagram of a Duplex Channel Configured in XAUI Mode



XAUI Mode Receiver Architecture

Figure 5–4 diagrams the digital components of the receiver in XAUI mode.

Figure 5–4. Block diagram of Receiver Digital Components in XAUI Mode

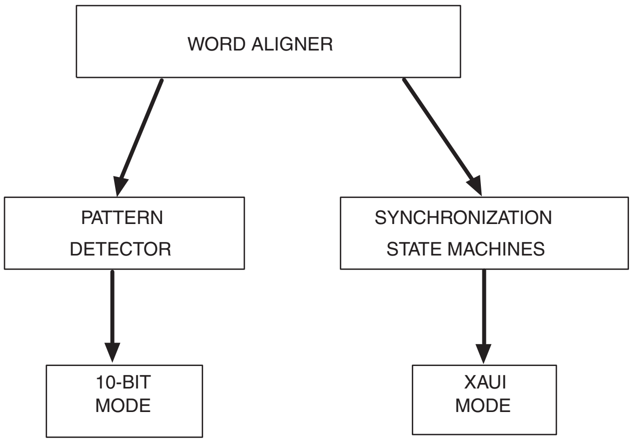


Word Aligner

For embedded clocking schemes, the clock is recovered from the incoming data stream based on the transition density of the data. This feature eliminates the need for you to factor in receiver skew margins between the clock and data. However, with this clocking methodology, the word boundary of the re-timed data can be altered. Stratix GX transceivers offer an embedded word alignment circuit that can be used in conjunction with the pattern detector to align the word boundary of the re-timed data to a specified comma. You can configure this embedded circuit to synchronize with the XAUI protocols.

The word aligner is composed of a pattern detector and synchronization state machines. The word aligner cannot be bypassed, but if the `rx_enacdet` signal is not used, the word aligner does not alter the data. Figure 5–5 shows the various components of the word aligner. The functionality of each component is described in the following sections.

Figure 5–5. Stratix GX Word Aligner Components



Pattern Detector Module

The pattern detector matches a pre-defined comma to the current byte boundary. If the comma is found, the optional `rx_patterndetect` signal is asserted for the duration of one clock cycle to signify that the comma exists in the current word boundary. The pattern detector module indicates only that the signal exists and does not modify the word boundary. Modification of the word boundary is discussed in the word alignment and synchronization sections. You can program a 10-bit pattern for the pattern detector to recognize.

This module matches the 10-bit comma specified in the MegaWizard® Plug-In Manager with the data and its complement in the current word boundary. Both positive and negative disparities are checked in this mode. For example, if a /K28.5/ (`b'0011111010`) pattern is specified as the comma, the `rx_patterndetect` signal is asserted if `b'0011111010` or `b'1100000101` is detected in the incoming data.

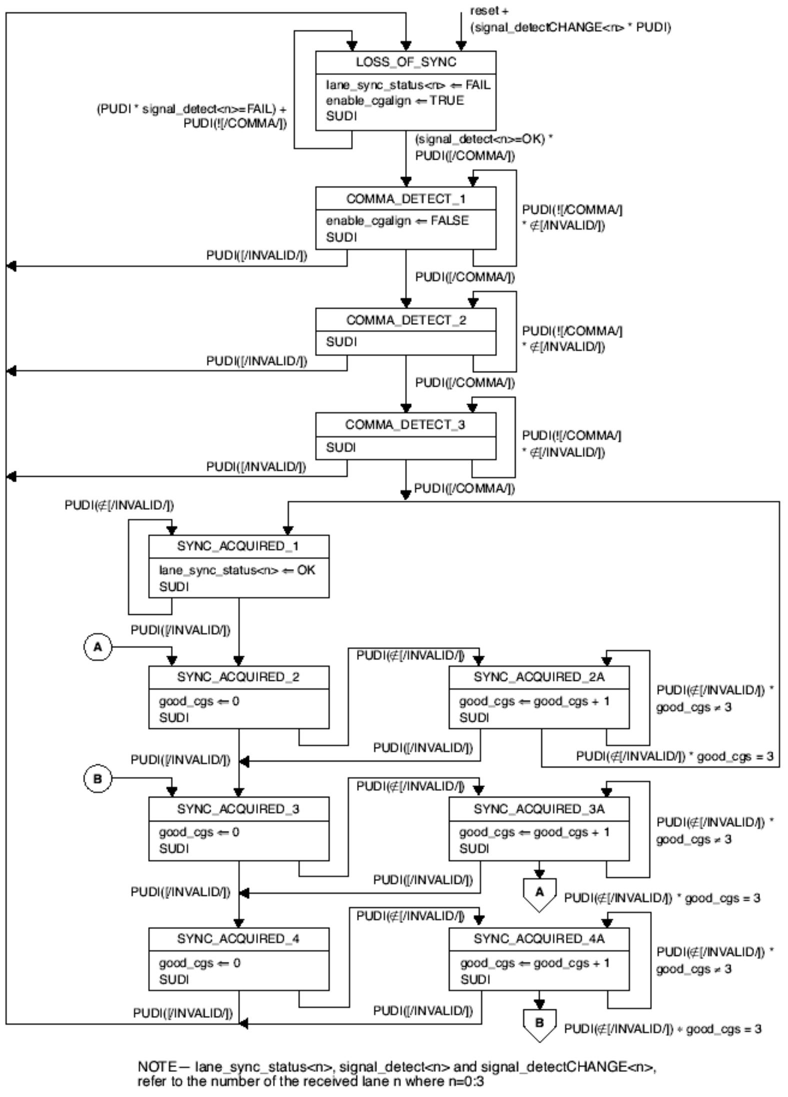
XAUI uses an embedded clocking scheme that re-times the data, which can also alter the code-group boundary. The boundaries of the code-groups are realigned through a synchronization process specified in clause 48 of the IEEE 802.3ae standard, which states that synchronization is achieved upon the reception of four /K28.5/ commas. Each comma can be followed by any number of valid code-groups. Invalid code-groups are not supported during the synchronization stage.

XAUI Synchronization Mode

When a Stratix GX transceiver is configured to the XAUI protocol, the built-in pattern detector, word aligner, and XAUI state machines adhere to the PCS synchronization specification. The code-group synchronization is achieved upon the reception of four /K28.5/ commas. Each comma is followed by any number of valid code-groups. Invalid code-groups are not supported during the synchronization stage. When code-group synchronization is achieved, the optional `rx_syncstatus[]` signal is asserted. Refer to clause 47-48 of the IEEE P802.3ae standard for more information regarding the operation of the synchronization phase. If the `rx_sigdet` signal is valid and the reset is deasserted, the synchronization state machine begins the synchronization process. If either of the two signals are not valid, the state machine falls out of the synchronization process and waits for the valid `rx_sigdet` signal and reset.

For reference, the PCS synchronization state diagram specified in clause 48 of the IEEE P802.3ae is shown in Figure 5–6.

Figure 5–6. IEEE 802.3ae PCS Synchronization State Diagram


reset + (signal_detectCHANGE<n> * PUDI)
LOSS_OF_SYNC
lane_sync_status<n> ⇐ FAIL
enable_cgalign ⇐ TRUE
SUDI
(PUDI * signal_detect<n>=FAIL) + PUDI(![/COMMA/])
(signal_detect<n>=OK) * PUDI([/COMMA/])
COMMA_DETECT_1
enable_cgalign ⇐ FALSE
SUDI
PUDI(![/COMMA/] * ∉[/INVALID/])
PUDI([/INVALID/])
PUDI([/COMMA/])
COMMA_DETECT_2
SUDI
COMMA_DETECT_3
PUDI(∉[/INVALID/])
SYNC_ACQUIRED_1
lane_sync_status<n> ⇐ OK
SUDI
SYNC_ACQUIRED_2
good_cgs ⇐ 0
SUDI
SYNC_ACQUIRED_2A
good_cgs ⇐ good_cgs + 1
SUDI
PUDI(∉[/INVALID/]) * good_cgs ≠ 3
PUDI(∉[/INVALID/]) * good_cgs = 3
SYNC_ACQUIRED_3
SYNC_ACQUIRED_3A
SYNC_ACQUIRED_4
SYNC_ACQUIRED_4A
A
B
NOTE— lane_sync_status<n>, signal_detect<n> and signal_detectCHANGE<n>, refer to the number of the received lane n where n=0:3

Note to Figure 5–6:

(1) `lane_sync_status<n>`, `signal_detect<n>`, and `signal_detectCHANGE<n>` refer to the number of the received lane n where n = 0 to 3.

Channel Aligner

You use the channel aligner when implementing the XAUI protocol, to ensure that the channels are aligned. The channel aligner uses a 16-word-deep FIFO module.

Ordered sets can be misaligned with respect to one another because of board skew or differences between the independent clock recoveries per serial lane. Channel alignment re-aligns the ordered sets. This process is commonly referred to as deskew or channel bonding. Channel alignment is accomplished by using the alignment code-group, referred to as /A/. The /A/ code-group is transmitted simultaneously on all four lanes, constituting an ||A|| ordered set, during idles or inter packet gaps (IPG). XAUI receivers use these code-groups to resolve any lane to lane skew. Skew between the lanes can be up to 40 UI (12.8ns) as specified in the standard, which relaxes the board design constraints. Figure 5–7 shows lane skew at the receiver input, and how the deskew circuitry uses the /A/ code-group to deskew the channels.

Figure 5–7. Example of Lane Skew at Receiver Input

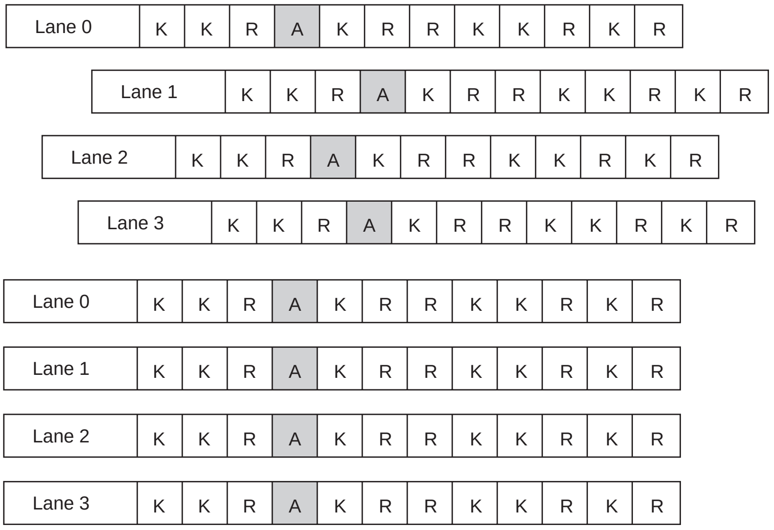
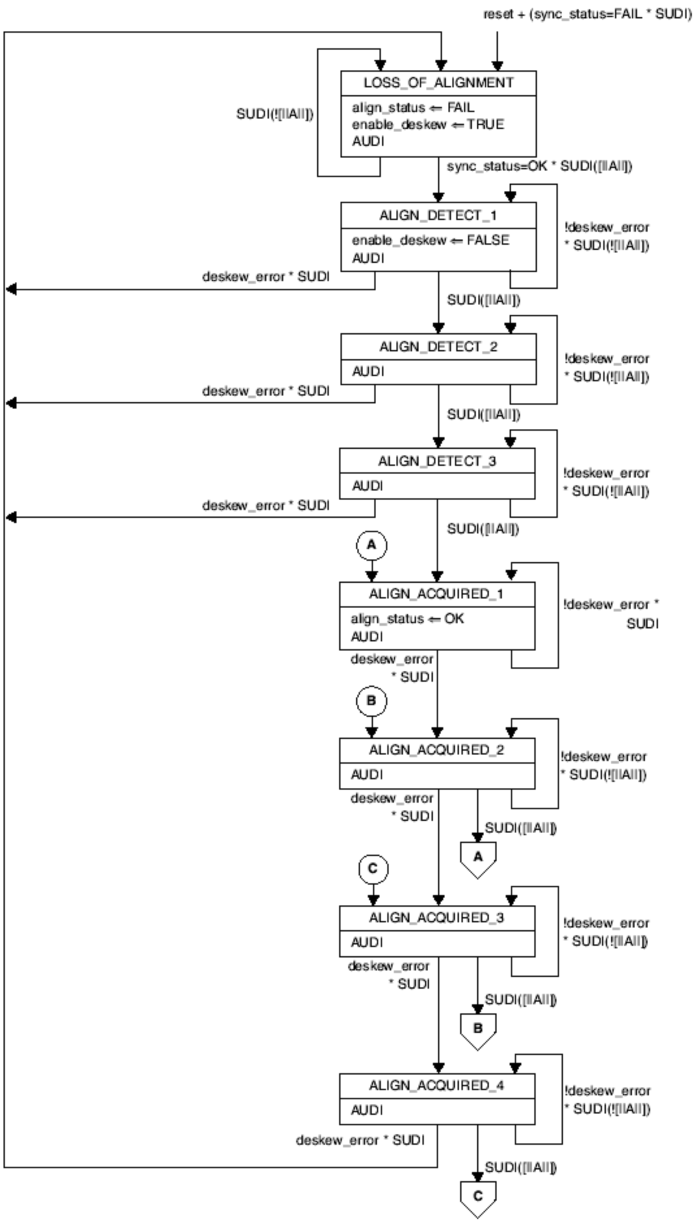


Figure 5–8. IEEE802.3ae PCS Deskew State Diagram



Stratix GX transceivers handle XAUI channel alignment with a dedicated deskew macro consisting of a 16-word-deep FIFO module that is controlled by a XAUI deskew state machine. The XAUI deskew state machine first looks for the /A/ code-group within each channel. When /A/ is detected in each channel, the deskew FIFO module is enabled. The deskew state machine then monitors the reception of /A/ code-groups. When four aligned /A/ code-groups are received, the `rx_channelaligned[]` signal is asserted. The deskew state machine continues to monitor the reception of /A/ code-groups and deasserts the `rx_channelaligned[]` signal if alignment conditions are lost. This built-in deskew macro is only enabled for the XAUI protocol. For reference, the PCS deskew state diagram specified in clause 48 of the IEEE P802.3ae is shown in Figure 5–8.

Rate Matcher

XAUI can operate in multi-crystal environments, which can tolerate a frequency variation of ±100 ppm between crystals. Stratix GX transceivers have embedded circuitry to perform clock rate compensation. This is achieved by the insertion or removal of the PCS SKIP code-group (/R/) from the inter packet gap (IPG) or idle stream. This process is called *rate matching* and is sometimes referred to as clock rate compensation.

The rate matcher in Stratix GX transceivers consists of a 12-word-deep FIFO module along with control logic. In XAUI mode, the controller begins to write data into the FIFO module whenever the `rx_channelaligned` signal is asserted. Within the control logic, there is a FIFO module counter that keeps track of the read and write executions. When the FIFO module is near an overflow condition, the receivers delete the /R/ code-group simultaneously across all channels during IPG or idle conditions. If the FIFO counter is near an underflow condition, the receivers insert the /R/ code-group simultaneously across all channels during IPG or idle conditions. This circuitry compensates for ±100 ppm frequency variations.

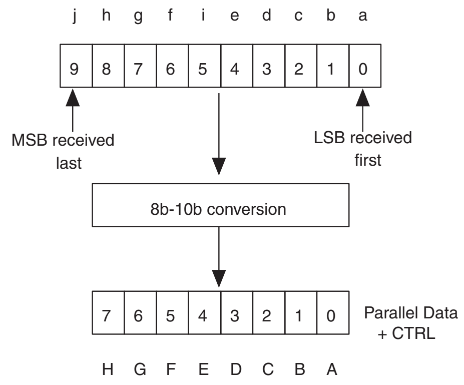
8B/10B Decoder

The 8B/10B decoder is part of the Stratix GX transceiver blocks. The purpose of the 8B/10B decoder is to restore the 8-bit data + 1-bit control identifier from the 10-bit code.

10-Bit Decoding

The 8B/10B decoder translates the 10-bit encode code into the 8-bit equivalent data or control code. The 10-bit encoded code is received LSB to MSB. The data that is received must be from the supported Dx.y or Kx.y list. All 8B/10B control signals (disparity error, control detect, and code error) are pipelined with the data in the Stratix GX receiver block and are edge-aligned with the data. Figure 5–9 diagrams the 10- to 8-bit conversion.

Figure 5–9. 10-Bit to 8-Bit Conversion



Reset

The `rxdigitalreset` signal governs the reset condition of the 8B/10B decoder. In reset, the disparity registers are cleared. Upon exiting reset, the 8B/10B decoder can start with either a positive or negative disparity. The decoder calculates the initial running disparity based on the first valid code received.

The receiver block must be word-aligned after reset before the 8B/10B decoder can decode valid data or control codes.

Code Error Detect

The `rx_errdetect` signal indicates when the code received contains an error. This port is optional and if not in use, there is no way to detect whether a code received is valid or not. The `rx_errdetect` goes high if code that is received is invalid or if it contains a disparity error. If code that is received is not part of the valid Dx.y or Kx.y list, the `rx_errdetect` signal goes high. This signal is aligned to the invalid code word that is received at the FPGA logic array.

Disparity Error Detector

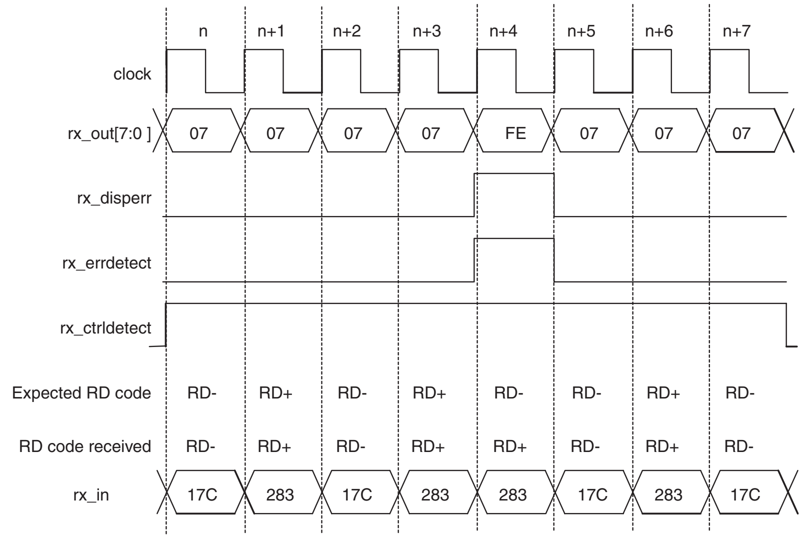
The 8B/10B decoder detects disparity errors based on which 10-bit code it received. The disparity error is indicated at the optional `rx_disperr` port. The current running disparity is based on the disparity calculation of the last code received. The disparity calculation is described in Appendix A, Data & Control Codes.

If negative disparity is calculated for the last 10-bit code, a neutral or positive disparity 10-bit code is expected. If the decoder does not receive a neutral or positive disparity 10-bit code, the `rx_disperr` signal goes high, which indicates that the code received had a disparity error.

If a positive disparity is calculated, a neutral or negative disparity 10-bit code is expected. `Rx_disperr` goes high if the code received is not as expected. When the `rx_disperr` signal is high, the `rx_errdetect` signal also transitions high.

In XAUI mode, idle character 8'h07 is converted to 8'hbc, 8'h1c, or 8'h7c. Figure 5–10 shows 8'hbc encoded into RD+ and RD-, and also shows a case where the disparity is violated. A K28.5 code has an 8-bit value of `8'hbc` and a 10-bit value (`jhgfiedcba`). The 10-bit value is 10'b0011111010 (10'h17c) for RD- or 10'b1100000101 (10'h283) for RD+. Assume that the running disparity at time n-1 is negative, so the expected code at time *n* is from the RD- column. Since a K28.5 does not have a balanced 10-bit code (equal number of 1's and 0's), the expected RD code toggles back and forth between RD- and RD+. At time n+3, the 8B/10B decoder received a RD+ K28.5 code (10'h283), which makes the current running disparity negative. At time n+4, because the current disparity is negative, a K28.5 from the RD- column is expected, but a K28.5 code from the RD+ is received instead. This code prompts `rx_disperr` to go high during time n+4 to indicate that the K28.5 code had a disparity error. The current running disparity at the end of time n+4 is negative, because a K28.5 from the RD+ column was received. Based on the current running disparity at the end of time n+5, a positive disparity K28.5 code (from the RD-) column is expected at time n+5.

Figure 5–10. Disparity Error

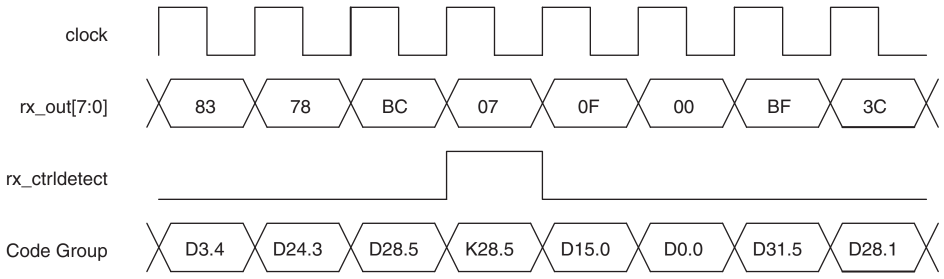


Control Detect

The 8B/10B has the ability to differentiate between data and control codes via the `rx_ctrldetect` port. If this port is not in use, there is no way to differentiate Dx.y from Kx.y.

Figure 5–11 shows an example waveform demonstrating the detection of a K28.5 code (BC + ctrl). The `rx_ctrldetect`=1'b1 is aligned with `8'hbc`, which indicates that this code is a control code. The reset of the code received is data.

Figure 5–11. Control Code Detection



PCS - XGMII Code Conversion

In XAUI mode, the 8b/10b decoder in Stratix GX transceivers is controlled by a global receiver state machine that maps various PCS code-groups into specific 8-bit XGMII codes. Table 5–3 lists the PCS code group to XGMII character mapping.

Table 5–3. PCS Code-Group to XGMII Character Mapping *Note (1)*

XGMII RXC	XGMII RXD	PCS Code Group	Description
0	00 through FF	Dxx.y	Normal data reception
1	07	K28.0 or K28.3 or K28.5	Idle in ||**I**||
1	07	K28.5	Idle in ||**T**||
1	9C	K28.4	Sequence
1	FB	K27.7	Start
1	FD	K29.7	Terminate
1	FE	K30.7	Error
1	FE	Invalid code-group	Received code-group

Note to Table 5–3:
(1) Values in RXD column are in hexadecimal.

Byte Deserializer

The byte deserializer module reduces the speed at which the FPGA logic array must run to meet performance specifications. It is possible to bring the data rate down from the line rate by a factor of 20.

The input to this module is 8 bits of data; the output to the FPGA logic array is deserialized to 16 bits. The byte deserializer does not process the data, and therefore, the control signals fed to the module are simply processed to match the latency to the data.

The byte deserializer in the receiver block takes in up to 13 bits. It is possible to feed the following to the byte deserialzer:

- 8 bits of data and up to 2 control signals (`rx_patterndetect`, `rx_syncstatus`)
- 8 bits of data and up to 5 control signals (`rx_patterndetect`, `rx_syncstatus`, `rx_disperr`, `rx_ctrldetect`, and `rx_errdetect`)

The byte deserializer outputs up to 26 bits, depending on the number of bits that was passed to it. When the input includes data and control signals, the data and the control signals are deserialized to include double the data bits and 2 bits of each control signal, one for the MSB and one for the LSB. This case is shown in the XAUI mode where the inputs to the Byte Deserializer are `datain[7..0]`, `patterndetect`, `syncstatus`, `disperr`, `ctrldet`, and `errdet`. These 13 input signals feeding the byte deserializer and 26 output signals are fed to the FPGA logic array. The signals are sent into the logic array as two 13-bit buses. The aggregate bandwidth does not change by using the byte deserializer because the logic array data width is doubled.

Figure 5–12 demonstrates input and output signals of the byte deserializer when deserializing an 8-bit data input to 16 bits. In this case, the alignment pattern B (10111100) is located in the MSB of the 16-bit output, and this is reflected with `patterndetect[1]` going high. The output of the byte deserializer is BA, DC, FE, and so on.

Figure 5–12. Receiver Byte Deserializer in 8/16-Bit Mode with Alignment Pattern in MSB

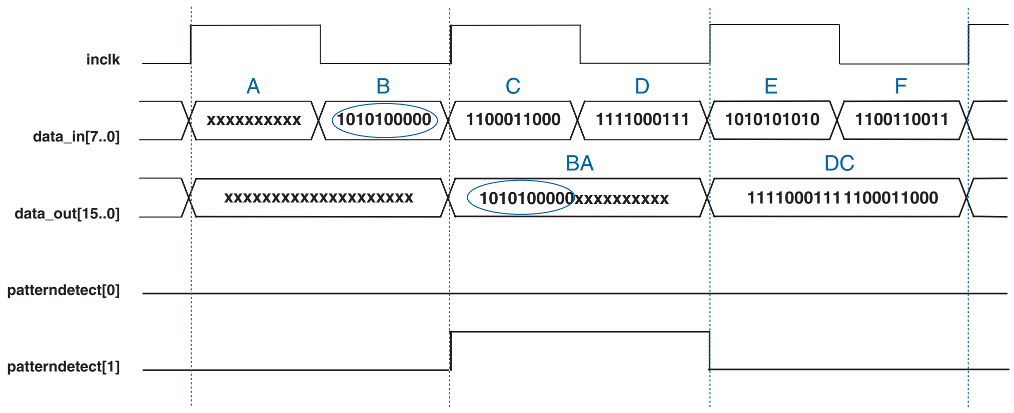
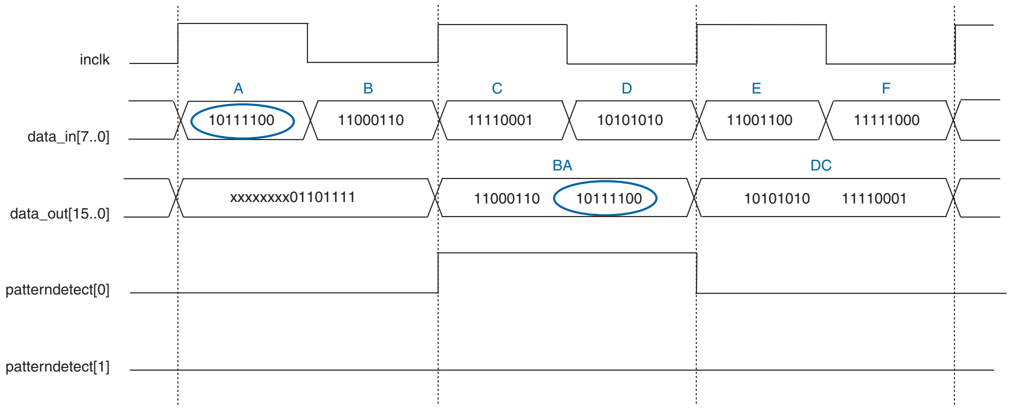


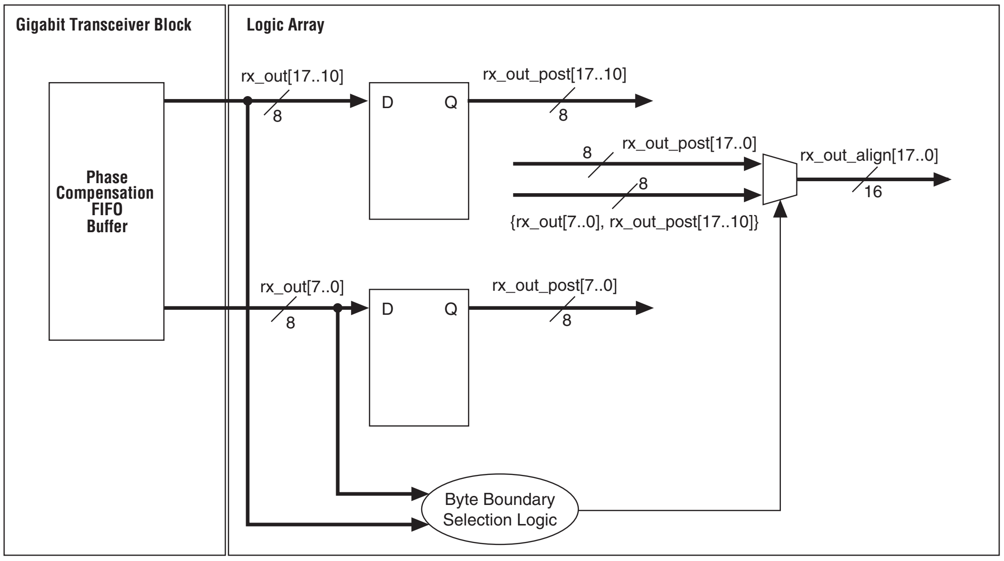
Figure 5–13 demonstrates the alternate case of the alignment pattern found in the LSB of the 16-bit output. Correspondingly, `patterndetect[0]` goes high. In this case, the output is BA, DC, FE, and so on.

Figure 5–13. Receiver Byte Deserializer in 8/16-Bit Mode With Alignment Pattern in LSB



The logic array must include logic to perform byte position alignment when the data enters the logic array, as seen in Figure 5–14. In this example, the byte position selection logic determines the proper byte position based on the pattern detect signal.

Figure 5–14. Receiver Byte Deserializer Data Recovery in Logic Array



Receiver Phase Compensation FIFO Module

The receiver phase compensation FIFO module is located at the FPGA logic array interface in the receiver block and is four words deep. The FIFO module compensates for the phase difference between the clock in the FPGA and the operating clocks in the transceiver block.

In XAUI mode, the write port is clocked by the `refclk` from the transmitter PLL. This clock is half the rate if the byte deserializer is used. The read clock is clocked by `coreclk` (output from the transmitter PLL).

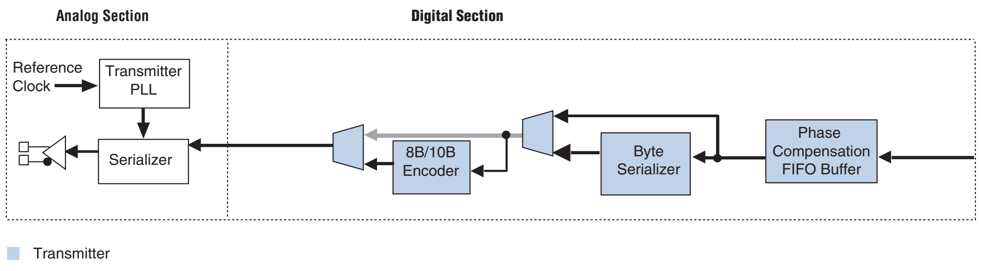
The receiver phase compensation FIFO module only accounts for phase differences.

The receiver phase compensation FIFO module is always used and cannot be bypassed.

XAUI Mode Transmitter Architecture

Figure 5–15 diagrams the transmitter digital components in XAUI mode.

Figure 5–15. Block Diagram of Transmitter Digital Components in XAUI Mode



Transmitter Phase Compensation FIFO Module

The Transmitter Phase Compensation FIFO module is located at the FPGA logic array interface in the transmitter block and is four words deep. The FIFO module compensates for the phase difference between the clock in the FPGA and the operating clocks in the transceiver block.

The read port of the phase compensation FIFO module is clocked by the transmitter PLL clock. The write clock is clocked by `tx_coreclk`. You can select the `tx_coreclk` as an optional transmitter input port to receive a clock supply. In this case, there must be no frequency difference between the tx_coreclk and the transmitter PLL clock. The transmitter Phase Compensation FIFO module can only account for phase differences.

If the tx_coreclk is not selected as an optional input transmitter port, tx_coreclk is fed by coreclk_out. This connection occurs using the logic array routing. As such, the software defaults to using an FPGA global clock, regional clock, or fast regional clock resource.

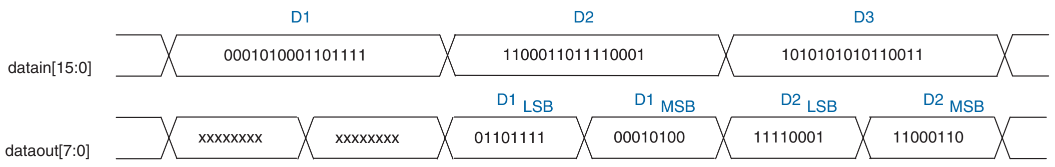
The transmitter phase compensation FIFO module is always used and cannot be bypassed. The input to this FIFO module is the data from the FPGA logic array. If they are used, the tx_ctrlenable and tx_forcedisparity signals are also passed through the FIFO module to ensure that they are synchronized with the data when they feed to the subsequent module.

Byte Serializer

The byte serializer in the transmitter block takes a 16-bit input from the FPGA logic array and serializes it to 8 bits. It transmits from the least significant byte to the most significant byte. The transmitter digital reset must always be used to reset the FIFO module pointers whenever an unknown state is encountered, such as when the transmitter PLL loses lock. Refer to the chapter *Reset Control & Power Down* for further details on the reset sequence.

Figure 5–16 demonstrates input and output signals of the byte serializer when serializing a 16-bit input to 8 bits. The tx_in[] signal is the input that has already passed from the FPGA logic array through the transmitter phase compensation FIFO module.

Figure 5–16. Transmitter Byte Serializer in 16- to 8-Bit Mode



The LSB is transmitted before the MSB in the transmitter byte serializer. Figure 5–16 shows the order of data transmitted. For the input of D1, the output is D1LSB and then D1MSB.

XGMII Character to PCS Code-Group Mapping

In XAUI mode, the 8b/10b encoder in the Stratix GX transceiver is controlled by a global transmitter state machine that maps various 8-bit XGMII codes to 10-bit PCS code-groups. This state machine complies with the IEEE 802.3ae PCS transmit specification. For reference, the PCS transmit source state diagram, specified in clause 48 of the IEEE P802.3ae specification, is shown in Figure 5–17.

Figure 5–17. IEEE 802.3ae PCS Transmit Source State Diagram

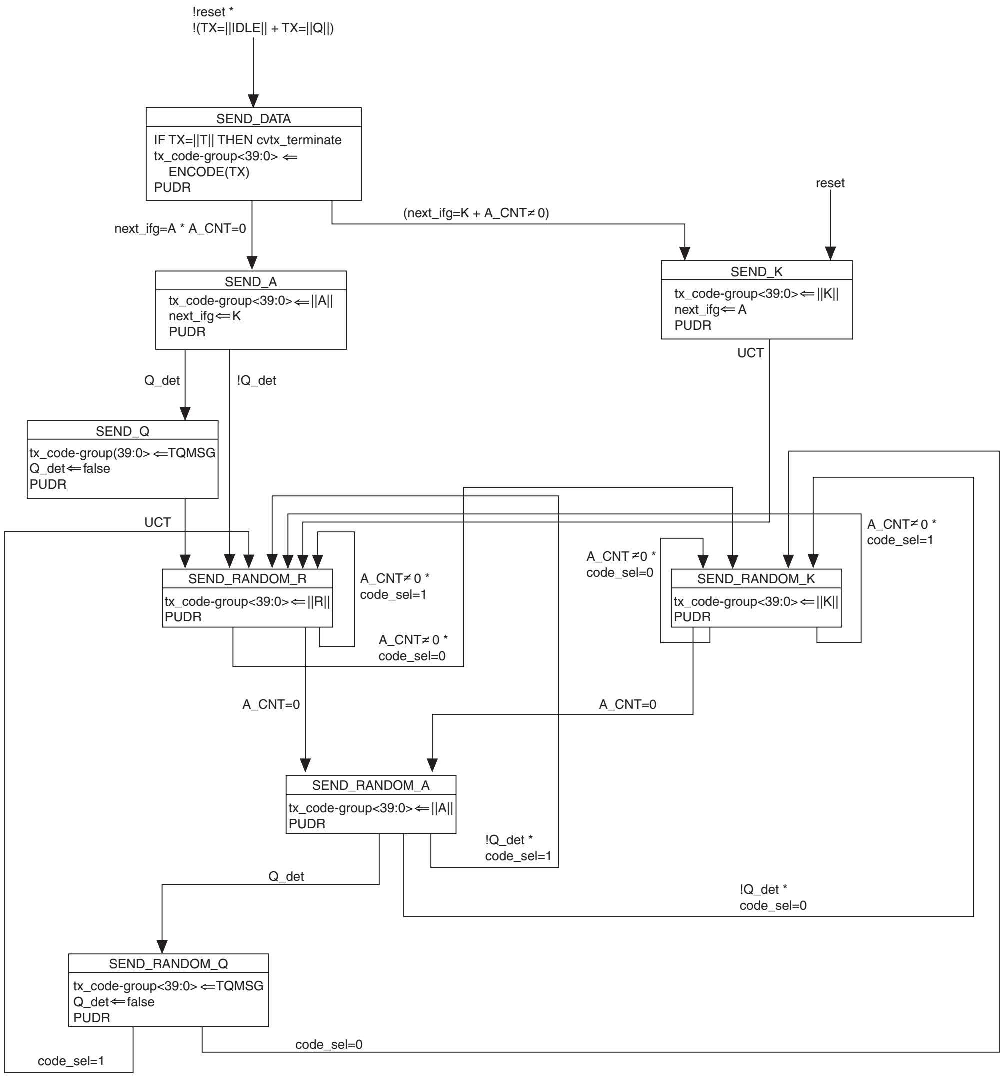


Table 5–4 lists the XGMII character-to -PCS code-group mapping.

Table 5–4. XGMII Character to PCS Code-Group Mapping *Note (1)*

XGMII TXC	XGMII TXD	PCS Code-Group	Description
0	00 though FF	Dxx,y	Normal data transmission
1	07	K28.0 or K28.3 or K28.5	Idle in ||**I**||
1	07	K28.5	Idle in ||**T**||
1	9C	K28.4	Sequence
1	FB	K27.7	Start
1	FD	K29.7	Terminate
1	FE	K30.7	Error
1	Any other value	K30.7	Invalid XGMII character

Note to Table 5–4:

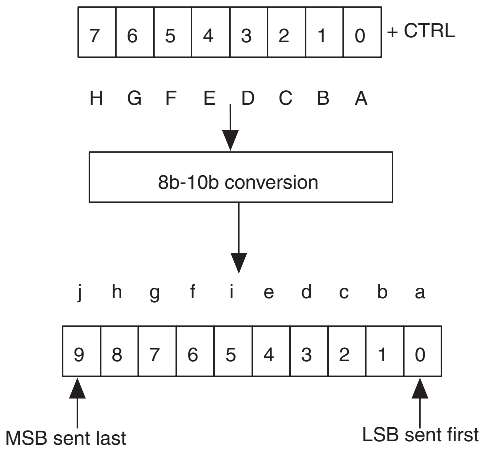
(1) Values in TXD column are in hexadecimal.

8B/10B Encoder

The 8B/10B encoder is part of the Stratix GX transceiver blocks. The purpose of the 8B/10B encoder is to translate 8-bit data and a 1-bit control identifier (via `tx_ctrlenable`) into a 10-bit DC balanced data stream.

For additional information about the 8B/10B code itself, refer to the *Data & Control Codes* chapter of the *Stratix GX Device Handbook, Volume 2*. The 8B/10B encoder translates the 8-bit data or 8-bit control character to its 10-bit equivalent. The conversion format is shown in Figure 5–18. The10-bit resultant data is transmitted LSB first by the serializer.

Figure 5–18. 8B/10B Conversion Format



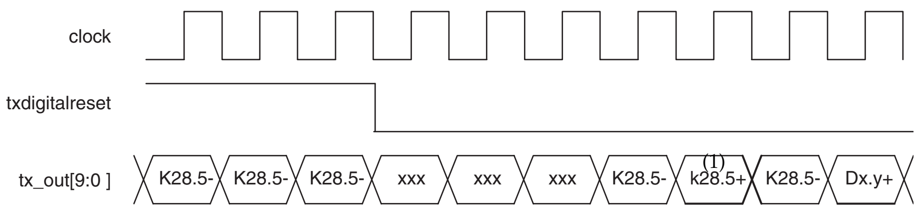
8B/10B Reset Condition

The `txdigitalreset` controls the reset of the 8B/10B encoder. To reset the 8B/10B encoder, `txdigitalreset` must be high. During reset, the running disparity registers are cleared, along with the data registers. Also, the 8B/10B encoder outputs a K28.5 pattern from the RD- column continuously until `txdigitalreset` goes low. The `tx_in[]` and `tx_ctrlenable[]` are ignored during the reset state. Once out of reset, the 8B/10B encoder starts with a bias towards negative disparity (RD-) and transmits three K28.5 codes for synchronizing before it starts encoding and transmitting the data on `tx_in[]`.

If the reset for the 8B/10B encoder is asserted, the 8B/10B decoder receiving the data may receive an invalid code error, sync error, control detect, and/or disparity error while `txdigitalreset` is high.

Figure 5–19 shows the reset behavior of the 8B/10B encoder. When in reset (`txdigitalreset` is high) a K28.5- (K28.5 10-bit code from the RD- column) is sent continuously until `txdigitalreset` goes low. Because of pipelining of the transmitter channel, there are several don't-care values (10'hxxx) until the first of three K28.5 is sent (Figure 5–19 shows three don't-cares). Normal user data follows the third K28.5.

Figure 5–19. Transmitter Output During Reset Conditions



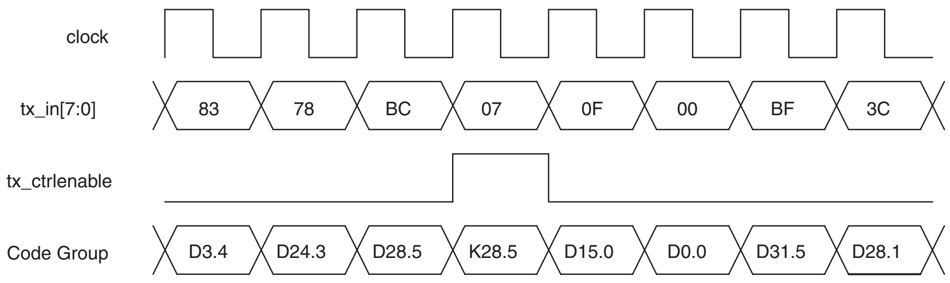
Note to Figure 5–19:

(1) K28.5 is an example, but an 07 control generates an idle sequence based on the 802.3 specification.

Control Code Encoding

The `tx_ctrlenable[]` signal dictates when a control code is to be inserted in the encoded data flow. When `tx_ctrlenable[]` is low, the byte at `tx_in[]` is encoded as data. When `tx_ctrlenable[]` is high, `tx_in[]` is encoded as a control word. The waveform in Figure 5–20 shows that 0x07 is encoded as a control code. The other values of `tx_in[]` are encoded as data.

Figure 5–20. Control Word Identification Waveform



The 8B/10B encoder does not check to see if the code word that is entered is one of the 12 valid codes. If an invalid control code is entered, the resulting 10-bit code might be encoded as an invalid code, which does not map to a valid Dx.y or Kx.y code), or a valid Dx.y code, depending on the value entered.

An example is the invalid encoding of a K24.1 (data = 8'h38 + `tx_ctrlenable` = 1'b1). Depending on the current running disparity, you can encode the K24.1 to be 10'b0110001100 (0x18C), which is equivalent to a D24.6+ (0xD8 from the RD+ column). An 8B/10B decoder decodes this value incorrectly (based on the 8B/10B Fibre Channel specification).

XAUI Mode Clocking

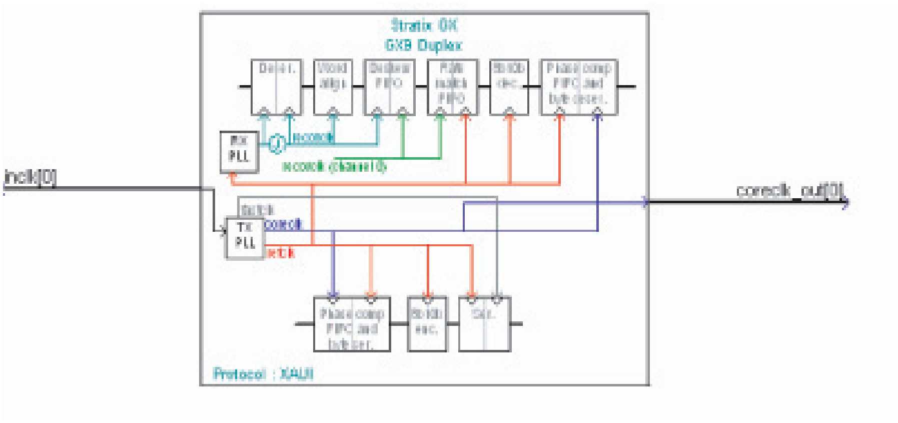
This section describes the clocking supported by the Stratix GX device in XAUI mode.

XAUI Mode Channel Clocking

This section describes clocking of the transceiver, internal clocking details, and external clock ports in XAUI mode. Each block diagram shows the input and output port clocks. Most of the settings are based on per transceiver block (4 channels) basis. By default, the MegaWizard Plug-In Manager selects a set of clocks for transmitters and receivers in a transceiver block when XAUI mode is selected. The MegaWizard Plug-In Manager also offers clock options other than the default selection, which facilitates the clocking scheme.

Figure 5–21 shows that the `altgxb` megafunction is configured such that the train receiver PLL with transmitter PLL is enabled. The transmitter PLL is fed from an `inclk` port that can itself be fed from a dedicated `REFCLKB`, global clock, regional clock, or fast regional clock source. The receiver logic is clocked by the recovered clock from the clock recovery unit up to a deskew FIFO module in the data path. Rate matching is done between recovered clock of channel 0 and `refclk` from the transmitter PLL. The data from the receive parallel interface, which is also from the phase compensation FIFO module, is clocked by `coreclk_out` from the transmitter PLL. On the transmitter channel, the output of the transmitter PLL, `coreclk_out`, is sent out of the logic array as an output and also loops back to clock the write side of the transmit phase compensation FIFO module and the read side of the receive phase compensation FIFO module.

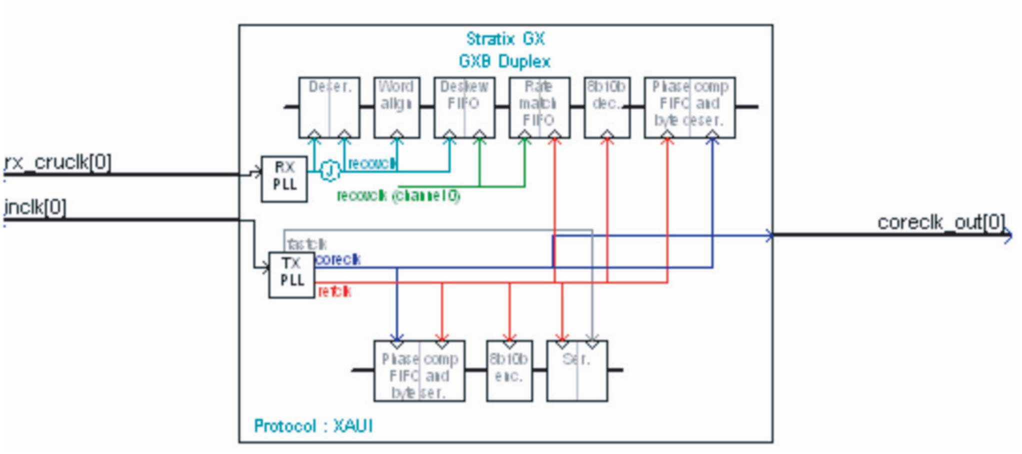
Figure 5–21. Default Configuration of altgxb Megafunction in XAUI Mode



You can disable the train receiver PLL CRU clock from transmitter PLL feature in the `altgxb` MegaWizard Plug-In Manager. Deselecting this option enables an additional `rx_cruclk` input reference clock port for the receiver PLL. You can use this feature to support additional multiplication factors for the receiver PLL, because it supports the separation of receiver and transmitter reference clocks. This separation is necessary if the output reference clock frequency from the transmitter PLL exceeds the 325 MHz phase frequency detector of the receiver PLL (for more information, refer to the *Stratix GX Analog Description* chapter of the *Stratix GX Device Handbook, Volume 2*). This configuration is shown in Figure 5–22.

Refer to the *Stratix GX Device Family Data Sheet* section of the *Stratix GX Device Handbook, Volume 1* for information on parallel interface speeds for other device speed grades.

Figure 5–22. Train Receiver PLL CRU Clock From Transmitter PLL Feature Is Disabled With Added Port RX_CRUCLK



If tx_coreclk is enabled and the train receiver CRU clock from transmitter PLL is disabled, and if other default options are also enabled, this configuration has an independent rx_cruclk that feeds the receiver PLL reference clock. This input clock port is only available when the receiver PLL is not trained by the transmitter PLL.

You can enable the write clock of the transmitter phase compensation FIFO module to manually feed in a clock from the FPGA logic array. You can use this option to optimize the global clock usage. For instance, if all transmitter channels between transceiver blocks are from a common clock domain, the transceiver instantiations use a total of one global resource instead of one global per transceiver block if the tx_coreclk option is not enabled. On the transmitter functionality screen under the optional port of transmitter section, if tx_coreclk is selected as an input port, the default clocking scheme changes by using tx_coreclk as the write clock for the phase compensation FIFO module.

There are two ways to connect tx_coreclk. To use coreclk_out, connect coreclk_out to tx_coreclk by using either gclk/rclk/fclk or logic array routing. Alternatively, tx_coreclk can be supplied from a crystal or any other clock source, as long as tx_coreclk is frequency-locked to the read side of the phase compensation FIFO module on the transmit side.

The tx_coreclk must be frequency matched with its respective read ports. The phase compensation FIFO module can only correct for phase, not for frequency differences. The receiver parallel interface clocks the data to the FPGA based on coreclk_out, which is the default option in the MegaWizard Plug-In Manager.

Figure 5–23 shows the clock configuration with these optional input ports enabled.

Figure 5–23. TX_CORECLK Enabled With RX_CRUCLK Port

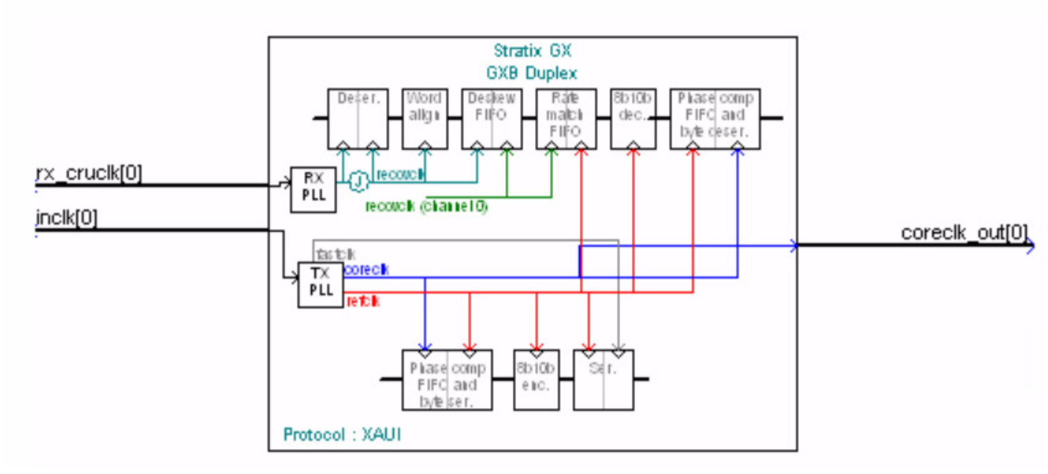


Table 5–5 describes the input and output ports shown in Figure 5–23.

Table 5–5. List of Input & Output Ports Available in XAUI Mode

Clock	Port	Description
inclk	Input	Input to the transmitter PLL. Available as a port when the transmitter PLL is instantiated.
rx_cruclk	Input	Input to CRU. Available as a port when CRU is not trained by the transmitter PLL.
tx_coreclk	Input	Clocks write port of the transmitter phase compensation FIFO module. Optional port in the Quartus® II software. Must be frequency matched to tx_pll_clk. If not available as a port, is fed by coreclk_out through logic array routing.
coreclk_out	Output	Output clock from the transmitter PLL equivalent to tx_pll_clk. Available as port if the transmitter PLL is used.

XAUI Inter-Transceiver Block Clocking

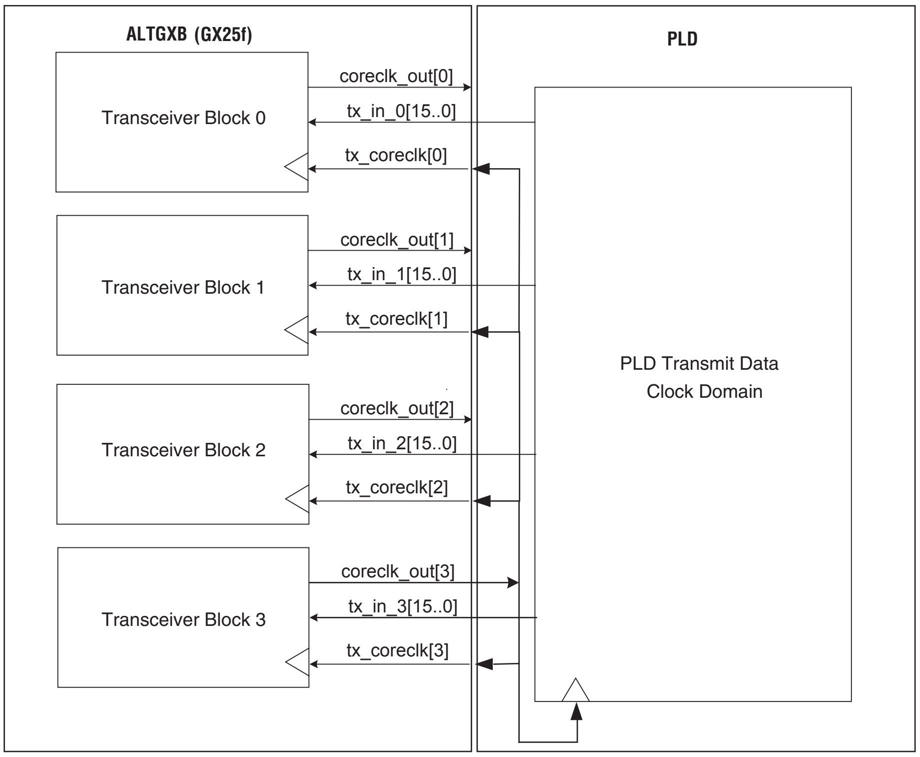
This section describes guidelines for the transceiver interface clocking that is used inside the FPGA logic array when multiple transceiver blocks are active. The transceiver blocks for each mode are supported by transceiver-to-FPGA interface clocking, unique to the Stratix GX transceiver. Different input and output clocks are available based on the options provided by the Quartus II MegaWizard Plug-In Manager's built-in functions. The number of supported channels varies based on the type of Stratix GX device you select (for example, EP1SGX40G, EP1SGX25F, and so on). Consider the clocking schemes at a system level with multiple lanes carefully to prevent pitfalls later in the design cycle. XAUI mode is transceiver-block-based and can only support lanes in multiples of four.

One of the clocking interfaces in the Stratix GX device is the interface between the transceiver and the FPGA, which can be further divided into FPGA-transmit of a transceiver and FPGA-receive of a transceiver. In XAUI mode, depending on the options set in the MegaWizard Plug-In Manager, you can use either the `coreclk_out` or `tx_coreclk` clock to send the data into the transmit of the transceiver. However, the `tx_coreclk` must be frequency locked with the transmit system clock of each transceiver block. (In each transceiver block, one transmitter PLL is shared among four transmitters.)

In a multi-transceiver block scenario, if there are synchronous data transfers based on transmit clocks when `tx_coreclk` is enabled for each channel, each enabled transceiver block must connect to one of the `coreclk_out` outputs. When `tx_coreclk` is not enabled, the Quartus II software automatically routes the `coreclk_out` signal to write the clock of the phase compensation FIFO module using a global, regional, or fast regional resource. In a multi-transceiver block configuration, this feature can lead to timing violations because the `coreclk_out` per transceiver block cannot guarantee a phase relationship. For this reason, Altera recommends clocking the `tx_coreclk` with a common clock for synchronous transmission.

In the multi-transceiver block case, use the transmit clock by enabling `tx_coreclk` and connecting one of the `coreclk_out` clock signals output from one of the transceiver blocks that is active. An illustration of this scheme is shown in Figure 5–24.

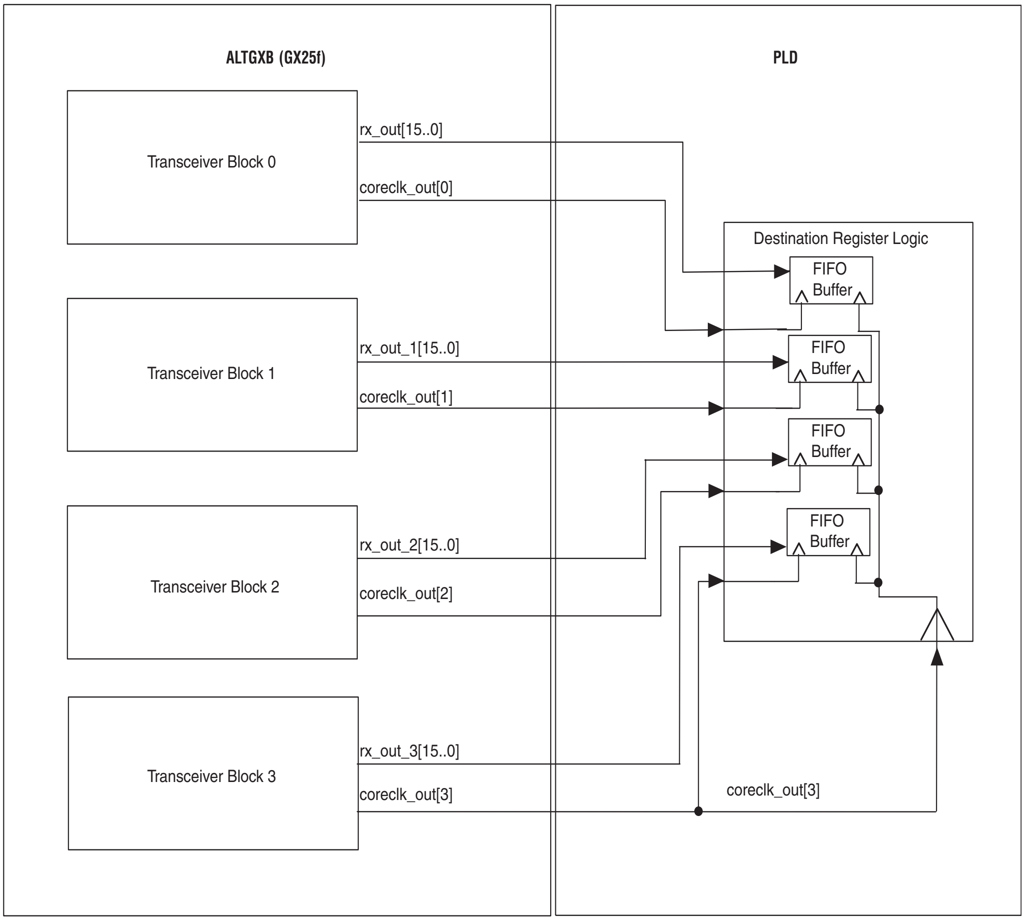
Figure 5–24. PLD to Transmit Interface Clocking Scheme in a Multi-Channel Application



At the FPGA-receive interface, there is no receive parallel interface clock option in the MegaWizard Plug-In Manager; the default is the transmitter PLL output clock, which is a transceiver internal clock.

Altera recommends implementing channel bonding across the transceiver blocks used in Stratix GX devices to ensure that there is no skew between the transceiver blocks (if each transceiver is operating, no channel bonding is required and the data can simply go to destination registers, as shown in Figure 5–25). Also, all traces in your design should match.

Figure 5–25. Clocking Scheme in Multi-Channel, Only CORECLK_OUT Is Enabled



XAUI mode applications are typically transceiver block-based. The previous recommendations are valid in a multi-transceiver block situation. In a multi-transceiver block situation, data striping across the channels is common. Skew introduced between transceiver blocks by passive and active elements of the link must be de-skewed in the PLD core (channel alignment) to ensure error-free data.

Another multi-transceiver block issue is the selection of the dedicated `refclkb` pin. Stratix GX channels are arranged in banks of four, which are called transceiver blocks. Each transceiver block has the ability to share a common reference clock through the Inter-Transceiver (IQ) lines. You can reduce the Stratix GX logic array clock usage by using the IQ lines. The IQ lines are used when a `refclkb` input port from one transceiver block or channel drives any other transceiver blocks or channels. The IQ line usage is determined automatically by the Quartus II software.

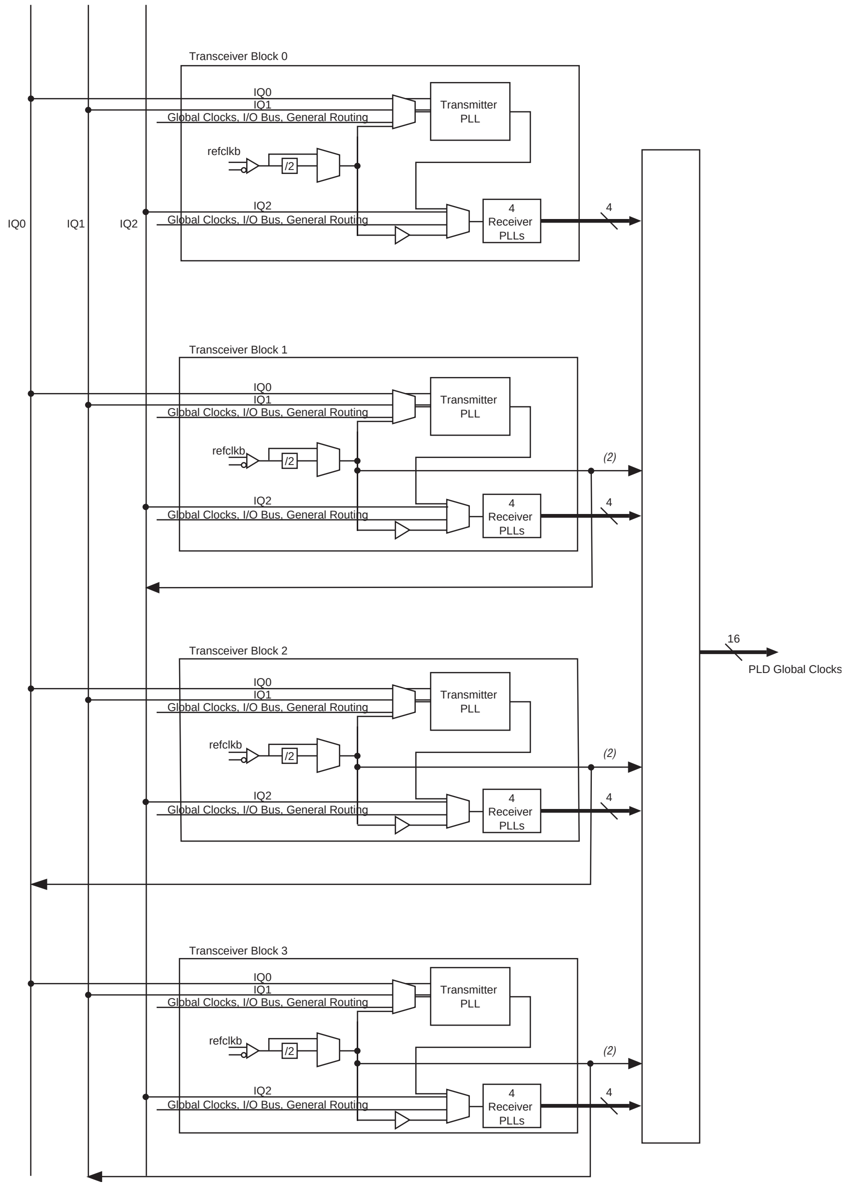
When determining the location of `refclkb` pins, consider what is fed by the pin you select. Table 5–6 shows the available IQ lines and which transceiver blocks are driven by `refclkb`. This information is based on the number of transceiver channels in the Stratix GX device.

Table 5–6. REFCLKB to Inter-Transceiver Line connections

Channel Density	REFCLKB in Transceiver Block Number	Channels in Transceiver Block	IQ Line Driven by REFCLKB
8 channels (EP1SGX10)	0	[3:0]	IQ2
	1	[7:4]	IQ0
16 channels (EP1SGX25)	0	[3:0]	N/A
	1	[7:4]	IQ2
	2	[11:8]	IQ0
	3	[15:12]	IQ1
20 channels (EP1SGX40)	0	[3:0]	N/A
	1	[7:4]	IQ2
	2	[11:8]	IQ0
	3	[15:12]	IQ1
	4	[19:16]	N/A

Figure 5–26 shows the transceiver routing with respect to Inter-Transceiver lines. This information is vital when placing `refclkb` pins. (When placing `refclkb` pins, refer to the *REFCLKB Pin Constraints* chapter of the *Stratix GX Device Handbook, Volume 2* for information about analog reads and `refclkb` pin usage constraints.) For example, if a `refclkb` pin is required to feed a transmitter PLL using an IQ line, the `refclkb` pin cannot be in transceiver block 1, because IQ2 only feeds the receiver PLLs.

Figure 5–26. IQ Line Connections for EP1SGX25 Device *Note (1)*



Notes to Figure 5–26:

(1) IQ lines are inter-transceiver block lines.

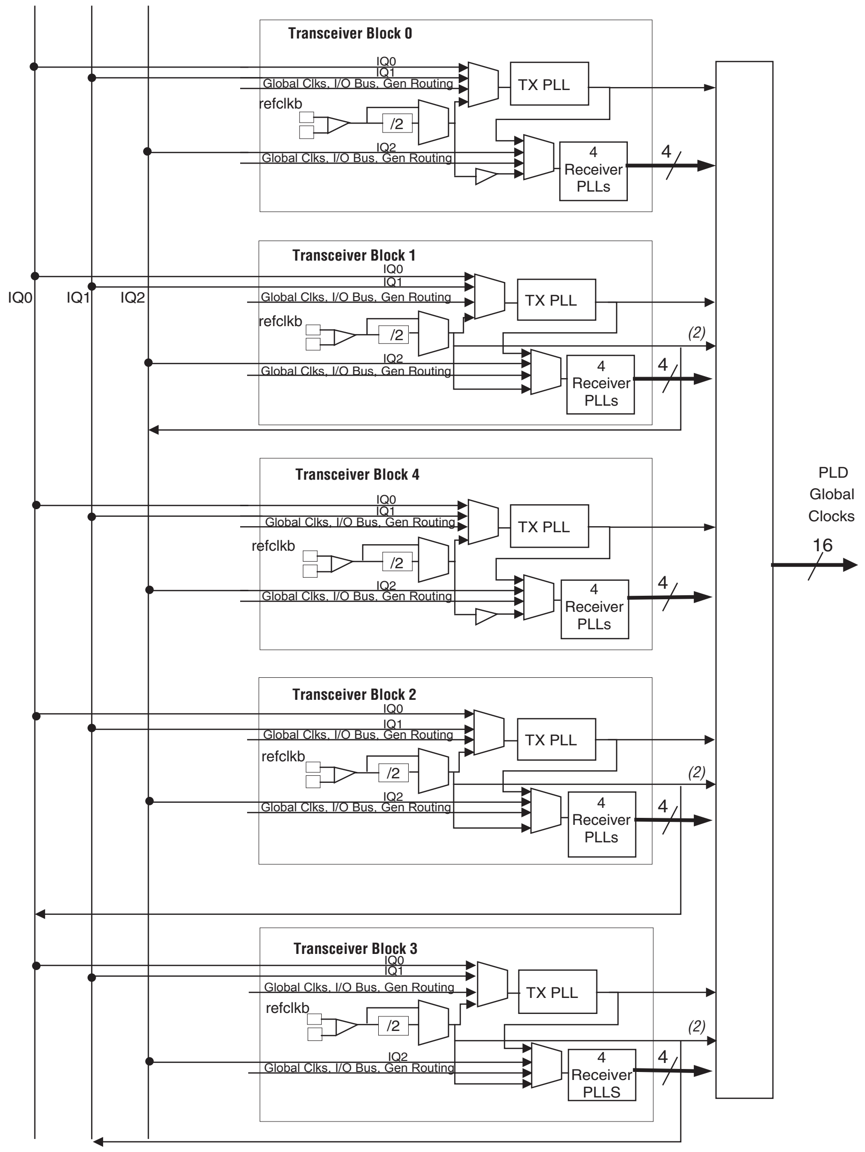
(2) If the /2 pre-divider is used, the path to drive the PLD logic array, local, or global clocks is not allowed.

(3) There are four receiver PLLs in each transceiver block.

Figure 5–27 shows the transceiver routing with respect to IQ lines for the EP1SGX40G device. This device has an extra transceiver block (transceiver block 4), which is in the middle of the other transceiver blocks, as shown. It is important to use this information when placing `refclkb` pins. (When placing `refclkb` pins, refer to the *REFCLKB Pin Constraints* chapter of the *Stratix GX Device Handbook, Volume 2* for information about analog reads and `refclkb` pin usage constraints.)

For example, if a `refclkb` pin is required to feed a transmitter PLL using an IQ line, the `refclkb` pin cannot be in transceiver block 1, because IQ2 only feeds the receiver PLLs.

Figure 5–27. IQ Line Connections for EP1SGX40G *Note (1)*



Notes to Figure 5–27:

(1) IQ lines are inter-transceiver block lines.

(2) If the /2 pre-divider is used, the path to drive the PLD logic array, local, or global clocks is not allowed.

(3) There are four receiver PLLs in each transceiver block.

XAUI Mode MegaWizard Plug-In Manager

This section describes the `altgxb` megafunction MegaWizard® Plug-In Manager options in XAUI mode. Altera recommends that the Stratix GX transceiver block be instantiated and parameterized through the MegaWizard Plug-In Manager in the Quartus II software. The Quartus II MegaWizard Plug-In Manager offers a graphical user interface (GUI) that organizes the `altgxb` options in easy-to-use sections. The MegaWizard Plug-In Manager also sets the proper ports and parameters automatically, based on the selected options and parameters. Invalid settings are automatically flagged to help prevent illegal configurations. The MegaWizard Plug-In Manager grays out any options that do not apply to XAUI mode.

Although it is possible to instantiate the Stratix GX block directly by calling out the `altgxb` megafunction, Altera recommends using the MegaWizard Plug-In Manager to instantiate the `altgxb` megafunction to reduce the chance of invalid settings.

XAUI Mode MegaWizard Plug-In Manager Considerations

Each `altgxb` MegaWizard Plug-In Manager instantiation can use one or more transceiver blocks based on the number of channels you select. There are four channels per transceiver block. In XAUI mode, the number of channels are in multiples of four or transceiver block based.

Each MegaWizard Plug-In Manager instantiation must have similar functionality and data rates. To use transceiver blocks that differ in functionality and/or data rates, create a separate MegaWizard Plug-In Manager instantiation for each transceiver block.

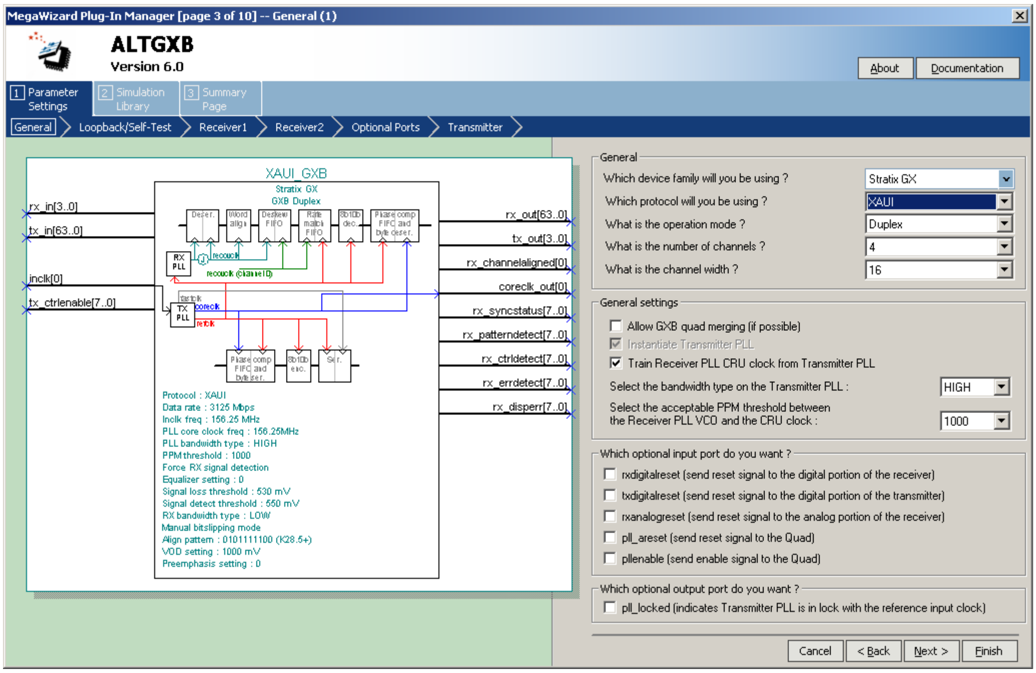
As mentioned in the clocking section, the MegaWizard Plug-In Manager displays the configuration of the `altgxb` megafunction. This diagram changes dynamically based on the selected mode, options, and clocking schemes.

XAUI Mode altgxb MegaWizard Plug-In Manager Options

This section shows the MegaWizard Plug-In Manager pages where you select the options for a XAUI mode configuration.

Figure 5–28 shows page 3 of the `altgxb` MegaWizard Plug-In Manager in XAUI mode.

Figure 5–28. MegaWizard Plug-In Manager - altgxb (Page 3)



Protocol : XAUI
Data rate : 3125 Mbps
Inclk freq : 156.25 MHz
PLL core clock freq : 156.25MHz
PLL bandwidth type : HIGH
PPM threshold : 1000
Force RX signal detection
Equalizer setting : 0
Signal loss threshold : 530 mV
Signal detect threshold : 550 mV
RX bandwidth type : LOW
Manual bitslipping mode
Align pattern : 0101111100 (K28.5+)
VOD setting : 1000 mV
Preemphasis setting : 0

General

Which device family will you be using ? Stratix GX
Which protocol will you be using ? XAUI
What is the operation mode ? Duplex
What is the number of channels ? 4
What is the channel width ? 16

General settings

- [ ] Allow GXB quad merging (if possible)
- [✓] Instantiate Transmitter PLL
- [✓] Train Receiver PLL CRU clock from Transmitter PLL

Select the bandwidth type on the Transmitter PLL : HIGH
Select the acceptable PPM threshold between the Receiver PLL VCO and the CRU clock : 1000

Which optional input port do you want ?

- [ ] rxdigitalreset (send reset signal to the digital portion of the receiver)
- [ ] txdigitalreset (send reset signal to the digital portion of the transmitter)
- [ ] rxanalogreset (send reset signal to the analog portion of the receiver)
- [ ] pll_areset (send reset signal to the Quad)
- [ ] pllenable (send enable signal to the Quad)

Which optional output port do you want ?

- [ ] pll_locked (indicates Transmitter PLL is in lock with the reference input clock)

Table 5–7 describes the available options on page 3 of the MegaWizard Plug-In Manager for your `altgxb` custom megafunction variation.

Table 5–7. MegaWizard Plug-In Manager Options (Page 3 for XAUI Mode) (Part 1 of 2)

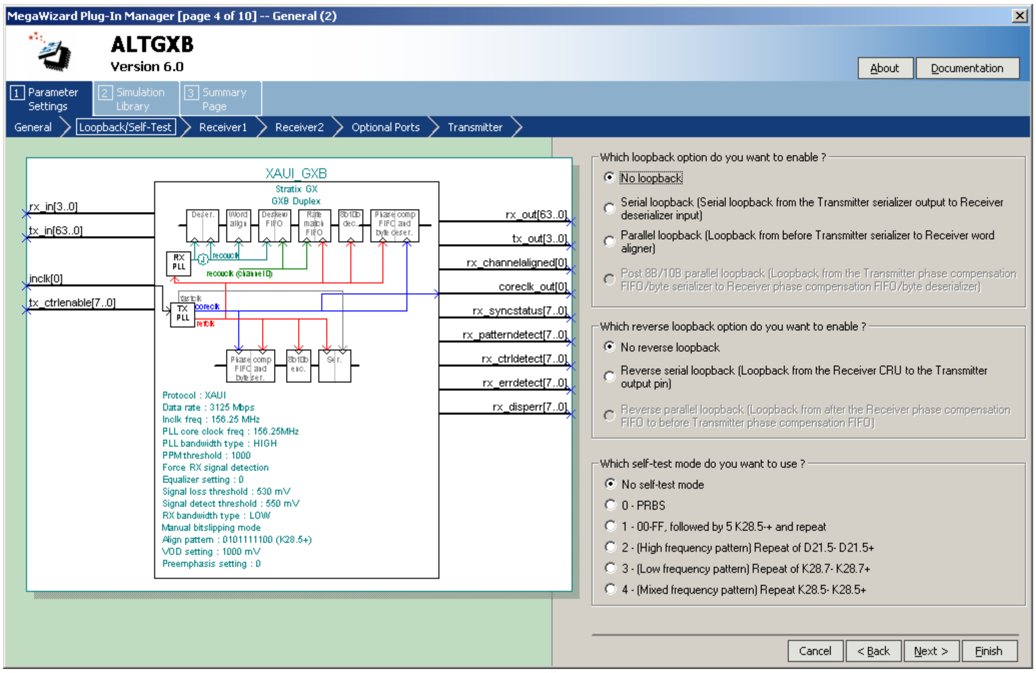
altgxb Setting	Description
Which device family will you be using?	Stratix GX is the only option available.
Which protocol will you be using?	For the XAUI mode, you must select the XAUI protocol.
What is the operation mode?	XAUI protocol mode supports duplex only mode.
What is the number of channels?	This value can be from 4 to the maximum number of channels available on the device in increments of 4.
What is the channel width?	16 bits is double width.
Allow GXB quad merging (if possible)	For information about this option, refer to the section "Stratix GX Transceiver Merging" on page 5–47.
Instantiate Transmitter PLL	For more information, refer to the *Stratix GX Analog Description* chapter in volume 2 of the *Stratix GX Device Handbook*.

Table 5–7. MegaWizard Plug-In Manager Options (Page 3 for XAUI Mode) (Part 2 of 2)

altgxb Setting	Description
Train Receiver PLL CRU clock from Transmitter PLL	For more information, refer to the *Stratix GX Analog Description* chapter in volume 2 of the *Stratix GX Device Handbook*.
Select the bandwidth type on the Transmitter PLL	For more information, refer to the *Stratix GX Analog Description* chapter in volume 2 of the *Stratix GX Device Handbook*.
Select the acceptable PPM threshold between the Receiver PLL VCO and the CRU clock	For more information, refer to the *Stratix GX Analog Description* chapter in volume 2 of the *Stratix GX Device Handbook*.
`rxdigitalreset` (send reset signal to the digital portion of the receiver)	The `rxdigitalreset` port resets the digital blocks in the receiver channel. Each active receiver channel has its own digital reset.
`txdigitalreset` (send reset signal to the digital portion of the transmitter)	The `txdigitalreset` port resets the digital blocks of the transmitter channel. Each active transmitter channel has its own digital reset.
`rxanalogreset` (send reset signal to the analog portion of the receiver)	The `rxanalogreset` port resets the receiver's analog circuits, including the receiver PLL. Each active receiver channel has its own analog reset.
`pll_areset` (send reset signal to the Quad)	The `pll_areset` port resets the entire transceiver block (all receiver and transmitter digital and analog circuits, including receiver and transmitter PLLs).
`pllenable` (send enable signal to the Quad)	The `pllenable` port enables the entire transceiver block; if deasserted, the entire transceiver block is held in the reset condition.
`pll_locked` (indicates Transmitter PLL is in lock with the reference input clock)	For more information, refer to the *Ports & Parameters* chapter in volume 2 of the *Stratix GX Device Handbook*.

Figure 5–29 shows page 4 of the `altgxb` MegaWizard Plug-In Manager in XAUI mode.

Figure 5–29. MegaWizard Plug-In Manager - altgxb (Page 4)



Protocol : XAUI
Data rate : 3125 Mbps
Inclk freq : 156.25 MHz
PLL core clock freq : 156.25MHz
PLL bandwidth type : HIGH
PPM threshold : 1000
Force RX signal detection
Equalizer setting : 0
Signal loss threshold : 530 mV
Signal detect threshold : 550 mV
RX bandwidth type : LOW
Manual bitslipping mode
Align pattern : 0101111100 (K28.5+)
VOD setting : 1000 mV
Preemphasis setting : 0

Which loopback option do you want to enable ?
- [✓ No loopback]
- Serial loopback (Serial loopback from the Transmitter serializer output to Receiver deserializer input)
- Parallel loopback (Loopback from before Transmitter serializer to Receiver word aligner)
- Post 8B/10B parallel loopback (Loopback from the Transmitter phase compensation FIFO/byte serializer to Receiver phase compensation FIFO/byte deserializer)

Which reverse loopback option do you want to enable ?
- [✓ No reverse loopback]
- Reverse serial loopback (Loopback from the Receiver CRU to the Transmitter output pin)
- Reverse parallel loopback (Loopback from after the Receiver phase compensation FIFO to before Transmitter phase compensation FIFO)

Which self-test mode do you want to use ?
- [✓ No self-test mode]
- 0 - PRBS
- 1 - 00-FF, followed by 5 K28.5-+ and repeat
- 2 - (High frequency pattern) Repeat of D21.5- D21.5+
- 3 - (Low frequency pattern) Repeat of K28.7- K28.7+
- 4 - (Mixed frequency pattern) Repeat K28.5- K28.5+

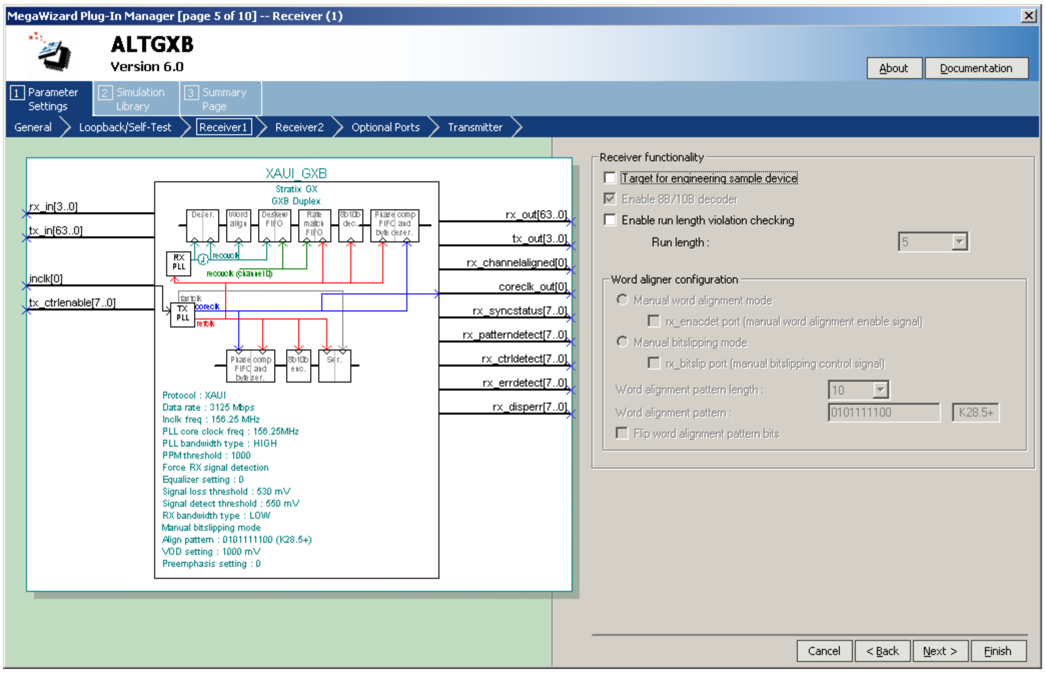
Table 5–8 describes the available options on page 4 of the MegaWizard Plug-In Manager for your `altgxb` custom megafunction variation.

Table 5–8. MegaWizard Plug-In Manager Options (Page 4 for XAUI Mode)

altgxb Setting	Description
Which loopback option do you want to enable?	For more information, refer to the *Loopback Modes* chapter in volume 2 of the *Stratix GX Device Handbook*.
Which reverse loopback option do you want to enable?	For more information, refer to the *Loopback Modes* chapter in volume 2 of the *Stratix GX Device Handbook*.
Which self-test mode do you want to use?	For more information, refer to the *Stratix GX Built-In Self Test (BIST)* chapter in volume 2 of the *Stratix GX Device Handbook*.

Figure 5–30 shows page 5 of the `altgxb` MegaWizard Plug-In Manager in XAUI mode.

Figure 5–30. MegaWizard Plug-In Manager - altgxb (Page 5)



Protocol : XAUI
Data rate : 3125 Mbps
Inclk freq : 156.25 MHz
PLL core clock freq : 156.25MHz
PLL bandwidth type : HIGH
PPM threshold : 1000
Force RX signal detection
Equalizer setting : 0
Signal loss threshold : 530 mV
Signal detect threshold : 550 mV
RX bandwidth type : LOW
Manual bitslipping mode
Align pattern : 0101111100 (K28.5+)
VOD setting : 1000 mV
Preemphasis setting : 0

Receiver functionality
- ☐ Target for engineering sample device
- ☑ Enable 8B/10B decoder
- ☐ Enable run length violation checking
- Run length : 5

Word aligner configuration
- ○ Manual word alignment mode
  - ☐ rx_enacdet port (manual word alignment enable signal)
- ○ Manual bitslipping mode
  - ☐ rx_bitslip port (manual bitslipping control signal)
- Word alignment pattern length : 10
- Word alignment pattern : 0101111100 K28.5+
- ☐ Flip word alignment pattern bits

Table 5–9 describes the available options on page 5 of the MegaWizard Plug-In Manager for your `altgxb` custom megafunction variation.

Table 5–9. MegaWizard Plug-In Manager Options (Page 5 for XAUI Mode) (Part 1 of 2)

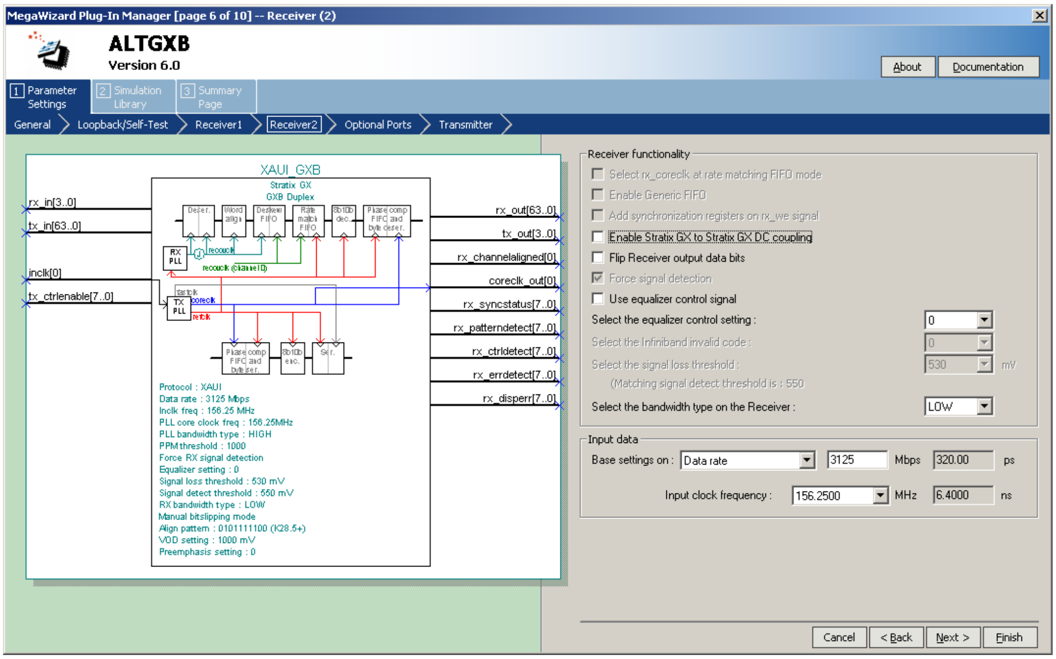
altgxb Setting	Description
Target for engineering sample device	You must select this option if the design is targeted for an engineering sample (ES) device.
Enable 8B/10B decoder	In XAUI mode, this option is always enabled because 8B/10B data is always encoded.
Enable run-length violation checking	For more information, refer to the *Stratix GX Analog Description* chapter in volume 2 of the *Stratix GX Device Handbook*.
Manual word alignment mode	This option is not available in XAUI mode.
`rx_enacdet` port (manual word alignment enable signal)	This option is not available in XAUI mode.
Manual bitslipping mode	This option is not available in XAUI mode.
`rx_bitslip` port (manual bitslipping control signal)	This option is not available in XAUI mode.

Table 5–9. MegaWizard Plug-In Manager Options (Page 5 for XAUI Mode) (Part 2 of 2)

altgxb Setting	Description
Word alignment pattern length	This option is not available in XAUI mode because the word alignment pattern is always set to a 10-bit K28.5 pattern.
Word alignment pattern	The word aligner in XAUI mode is always set as a 10-bit K28.5 pattern. Both positive and negative disparities are checked.
Flip word alignment pattern bits	This option is not available in XAUI mode.

Figure 5–31 shows page 6 of the `altgxb` MegaWizard Plug-In Manager in XAUI mode.

Figure 5–31. MegaWizard Plug-In Manager - altgxb (Page 6)



Protocol : XAUI
Data rate : 3125 Mbps
Inclk freq : 156.25 MHz
PLL core clock freq : 156.25MHz
PLL bandwidth type : HIGH
PPM threshold : 1000
Force RX signal detection
Equalizer setting : 0
Signal loss threshold : 530 mV
Signal detect threshold : 550 mV
RX bandwidth type : LOW
Manual bitslipping mode
Align pattern : 0101111100 (K28.5+)
VOD setting : 1000 mV
Preemphasis setting : 0

Receiver functionality

- [ ] Select rx_coreclk at rate matching FIFO mode
- [ ] Enable Generic FIFO
- [ ] Add synchronization registers on rx_we signal
- [ ] Enable Stratix GX to Stratix GX DC coupling
- [ ] Flip Receiver output data bits
- [x] Force signal detection
- [ ] Use equalizer control signal

Select the equalizer control setting : 0
Select the Infiniband invalid code : 0
Select the signal loss threshold : 530 mV
(Matching signal detect threshold is : 550
Select the bandwidth type on the Receiver : LOW

Input data

Base settings on : Data rate 3125 Mbps 320.00 ps
Input clock frequency : 156.2500 MHz 6.4000 ns

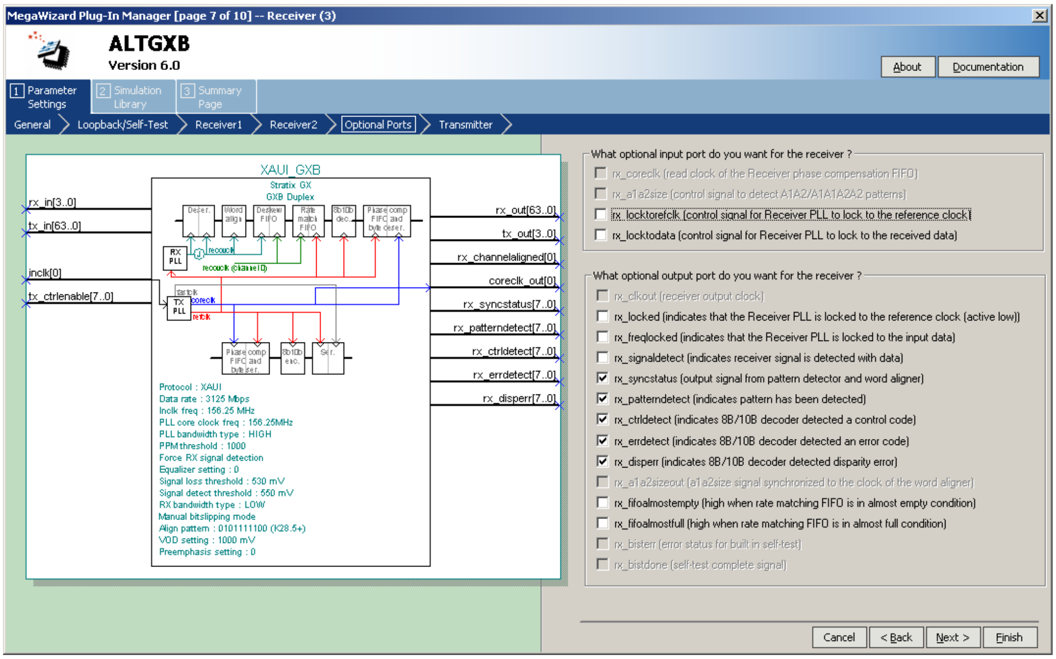
Table 5–10 describes the available options on page 6 of the MegaWizard Plug-In Manager for your `altgxb` custom megafunction variation.

Table 5–10. MegaWizard Plug-In Manager Options (Page 6 for XAUI Mode)

altgxb Setting	Description
Select `rx_coreclk` at rate matching FIFO mode	This option is not available in XAUI mode.
Enable Generic FIFO	This option is not available in XAUI mode, because a dedicated rate matching FIFO is included in the receiver data path by default.
Add synchronization registers on `rx_we` signal	This option is not available in XAUI mode.
Enable Stratix GX to Stratix GX DC coupling	For information about this option, refer to the *Stratix GX Analog Description* chapter in volume 2 of the *Stratix GX Device Handbook.*
Force signal detection	The Force Signal Detect option is always on and cannot be turned off. The signal detect circuitry is always forced, so the `rx_signaldetect` is always set in XAUI mode.
Use equalizer control signal	For information about this option, refer to the *Stratix GX Analog Description* chapter in volume 2 of the *Stratix GX Device Handbook.*
Select the equalizer control setting	For information about this option, refer to the *Stratix GX Analog Description* chapter in volume 2 of the *Stratix GX Device Handbook.*
Select the Infiniband invalid code	This option is not available in XAUI mode.
Select the signal loss threshold	This option is not available in XAUI mode.
Select the bandwidth type on the Receiver	For information about this option, refer to the *Stratix GX Analog Description* chapter in volume 2 of the *Stratix GX Device Handbook.*
Base settings on	By default, the data rate for XAUI is set to 3125 Mbps. Other data rates are possible, but they must adhere to the set multiplication factor of 2, 4, 5, 8, 10, 16, 20 of the input clock. Multiplication factors of 2, 4, 5 must use the `refclkb` pins. A multiplication factor of 2 also requires that the receiver PLL be trained by the transmitter PLL.

Figure 5–32 shows page 7 of the `altgxb` MegaWizard Plug-In Manager in XAUI mode.

Figure 5–32. MegaWizard Plug-In Manager - altgxb (Page 7)



Protocol : XAUI
Data rate : 3125 Mbps
Inclk freq : 156.25 MHz
PLL core clock freq : 156.25MHz
PLL bandwidth type : HIGH
PPM threshold : 1000
Force RX signal detection
Equalizer setting : 0
Signal loss threshold : 530 mV
Signal detect threshold : 550 mV
RX bandwidth type : LOW
Manual bitslipping mode
Align pattern : 0101111100 (K28.5+)
VOD setting : 1000 mV
Preemphasis setting : 0

What optional input port do you want for the receiver ?

- ☐ rx_coreclk (read clock of the Receiver phase compensation FIFO)
- ☐ rx_a1a2size (control signal to detect A1A2/A1A1A2A2 patterns)
- ☐ rx_locktorefclk (control signal for Receiver PLL to lock to the reference clock)
- ☐ rx_locktodata (control signal for Receiver PLL to lock to the received data)

What optional output port do you want for the receiver ?

- ☐ rx_clkout (receiver output clock)
- ☐ rx_locked (indicates that the Receiver PLL is locked to the reference clock (active low))
- ☐ rx_freqlocked (indicates that the Receiver PLL is locked to the input data)
- ☐ rx_signaldetect (indicates receiver signal is detected with data)
- ☑ rx_syncstatus (output signal from pattern detector and word aligner)
- ☑ rx_patterndetect (indicates pattern has been detected)
- ☑ rx_ctrldetect (indicates 8B/10B decoder detected a control code)
- ☑ rx_errdetect (indicates 8B/10B decoder detected an error code)
- ☑ rx_disperr (indicates 8B/10B decoder detected disparity error)
- ☐ rx_a1a2sizeout (a1a2size signal synchronized to the clock of the word aligner)
- ☐ rx_fifoalmostempty (high when rate matching FIFO is in almost empty condition)
- ☐ rx_fifoalmostfull (high when rate matching FIFO is in almost full condition)
- ☐ rx_bisterr (error status for built in self-test)
- ☐ rx_bistdone (self-test complete signal)

Table 5–11 describes the available options on page 7 of the MegaWizard Plug-In Manager for your `altgxb` custom megafunction variation.

Table 5–11. MegaWizard Plug-In Manager Options (Page 7 for XAUI Mode) (Part 1 of 2)

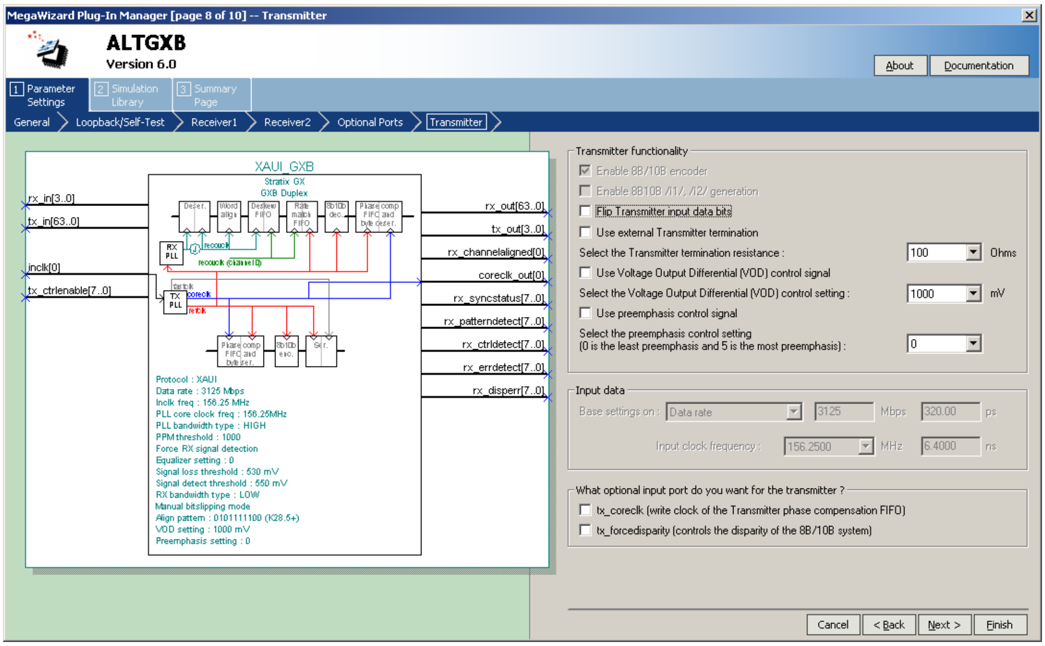
altgxb Setting	Description
`rx_coreclk` (read clock of the Receiver phase compensation FIFO)	This option is not available in XAUI mode.
`rx_a1a2size` (control logic signal to detect A1A2/A1A1A2A2 patterns)	This option is not available in XAUI mode.
`rx_locktorefclk` (control signal for Receiver PLL to lock to the reference clock)	Refer to the *Ports & Parameters* chapter in volume 2 of the *Stratix GX Device Handbook*.
`rx_locktodata` (control signal for Receiver PLL to lock to the received data)	Refer to the *Ports & Parameters* chapter in volume 2 of the *Stratix GX Device Handbook*.
`rx_clkout` (receiver input clock)	This option is not available in XAUI mode.

Table 5–11. MegaWizard Plug-In Manager Options (Page 7 for XAUI Mode) (Part 2 of 2)

altgxb Setting	Description
`rx_locked` (indicates that the Receiver PLL is locked to the reference clock (active low))	Receiver PLL lock indicator. For `rx_locked`, Low = receiver PLL is locked to the reference clock.
`rx_freqlocked` (indicates that the Receiver PLL is locked to the input data)	Refer to the *Ports & Parameters* chapter in volume 2 of the *Stratix GX Device Handbook*.
`rx_signaldetect` (indicates receiver signal is detected with data)	`rx_signaldetect` is only available in XAUI or GIGE mode. The signal detect circuitry is always forced, so the `rx_signaldetect` signal is always set in XAUI and GIGE modes, supporting backward compatibility with existing designs. Refer to the *Ports & Parameters* chapter in volume 2 of the *Stratix GX Device Handbook*.
`rx_syncstatus` (output signal from pattern detector and word aligner)	Indicates when the word aligner has aligned to the byte boundary. The `rx_syncstatus` signal goes high for one `rx_clkout` period when the word aligner aligns to the new byte boundary. In 16-bit mode, each high and low byte has a separate `rx_syncstatus` signal.
`rx_patterndetect` (indicates pattern has been detected)	Similar to `rx_syncstatus`, except that `rx_patterndetect` asserts only when the word alignment pattern appears in the data stream within the synchronized byte boundary.
`rx_ctrldetect` (indicates 8B/10B decoder detected a control code)	Refer to the *Ports & Parameters* chapter in volume 2 of the *Stratix GX Device Handbook*.
`rx_errdetect` (indicates 8B/10B decoder detected an error code	Refer to the *Ports & Parameters* chapter in volume 2 of the *Stratix GX Device Handbook*.
`rx_disperr` (indicates 8B/10B decoder detected disparity error)	Refer to the *Ports & Parameters* chapter in volume 2 of the *Stratix GX Device Handbook*.
`rx_a1a2sizeout` (a1a2size signal synchronized to the clock of the word aligner)	Refer to the *Ports & Parameters* chapter in volume 2 of the *Stratix GX Device Handbook*.
`rx_fifoalmostempty` (high when rate matching FIFO is in almost empty condition)	You can include this rate matching FIFO status signal by enabling this option. When driven HIGH, this signal indicates an almost empty condition for the rate matching FIFO.
`rx_fifoalmostfull` (high when rate matching FIFO is in almost full condition)	You can include this rate matching FIFO status signal by enabling this option. When driven HIGH, this signal indicates an almost full condition for the rate matching FIFO.
`rx_bisterr` (error status for built-in self-test)	Refer to the *Ports & Parameters* chapter in volume 2 of the *Stratix GX Device Handbook*.
`rx_bistdone` (self-test complete signal)	Refer to the *Ports & Parameters* chapter in volume 2 of the *Stratix GX Device Handbook*.

Figure 5–33 shows page 8 of the `altgxb` MegaWizard Plug-In Manager in XAUI mode.

Figure 5–33. MegaWizard Plug-In Manager - altgxb (Page 8)



MegaWizard Plug-In Manager [page 8 of 10] -- Transmitter

ALTGXB
Version 6.0

XAUI_GXB
Stratix GX
GXB Duplex

Protocol : XAUI
Data rate : 3125 Mbps
Inclk freq : 156.25 MHz
PLL core clock freq : 156.25MHz
PLL bandwidth type : HIGH
PPM threshold : 1000
Force RX signal detection
Equalizer setting : 0
Signal loss threshold : 530 mV
Signal detect threshold : 550 mV
RX bandwidth type : LOW
Manual bitslipping mode
Align pattern : 0101111100 (K28.5+)
VOD setting : 1000 mV
Preemphasis setting : 0

Transmitter functionality
- [✓] Enable 8B/10B encoder
- [ ] Enable 8B10B /I1/, /I2/ generation
- [ ] Flip Transmitter input data bits
- [ ] Use external Transmitter termination

Select the Transmitter termination resistance : 100 Ohms
- [ ] Use Voltage Output Differential (VOD) control signal

Select the Voltage Output Differential (VOD) control setting : 1000 mV
- [ ] Use preemphasis control signal

Select the preemphasis control setting
(0 is the least preemphasis and 5 is the most preemphasis) : 0

Input data
Base settings on : Data rate 3125 Mbps 320.00 ps
Input clock frequency : 156.2500 MHz 6.4000 ns

What optional input port do you want for the transmitter ?
- [ ] tx_coreclk (write clock of the Transmitter phase compensation FIFO)
- [ ] tx_forcedisparity (controls the disparity of the 8B/10B system)

Table 5–12 describes the available options on page 8 of the MegaWizard Plug-In Manager for your `altgxb` custom megafunction variation.

Table 5–12. MegaWizard Plug-In Manager Options (Page 8 for XAUI Mode) (Part 1 of 2)

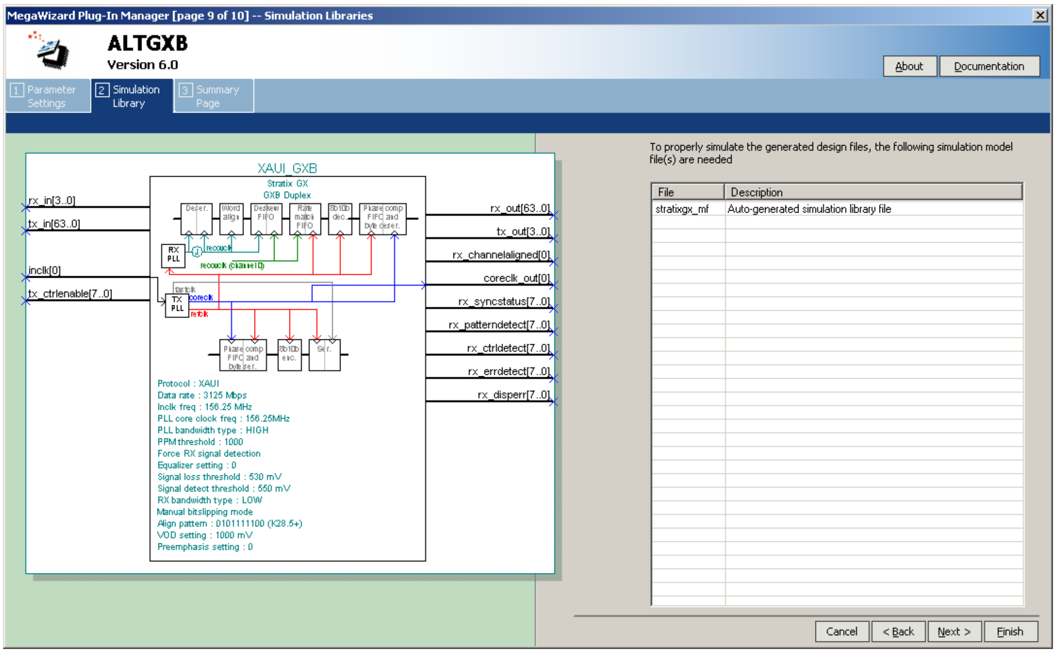
altgxb Setting	Description
Enable 8B/10B encoder	In XAUI mode, this option is always enabled because data is always 8B/10B encoded.
Enable 8B/10B /I1/, /I2/ generation	This option is not available in XAUI mode.
Use external Transmitter termination	For information about this option, refer to the *Stratix GX Analog Description* chapter in volume 2 of the *Stratix GX Device Handbook*.
Use Voltage Output Differential (VOD) control signal	For information about this option, refer to the *Stratix GX Analog Description* chapter in volume 2 of the *Stratix GX Device Handbook*.
Select the Voltage Output Differential (VOD) control setting	For information about this option, refer to the *Stratix GX Analog Description* chapter in volume 2 of the *Stratix GX Device Handbook*.
Use preemphasis control signal	For information about this option, refer to the *Stratix GX Analog Description* chapter in volume 2 of the *Stratix GX Device Handbook*.

Table 5–12. MegaWizard Plug-In Manager Options (Page 8 for XAUI Mode) (Part 2 of 2)

altgxb Setting	Description
Select the preemphasis control setting (0 is the least preemphasis and 5 is the most preemphasis)	For information about this option, refer to the *Stratix GX Analog Description* chapter in volume 2 of the *Stratix GX Device Handbook*.
`tx_coreclk` (write clock of the Transmitter phase compensation FIFO buffer	You can optionally choose the write clock of the transmitter phase compensation FIFO buffer. This clock should be frequency locked with the internal reference clock because the phase compensation FIFO buffer cannot tolerate frequency variations and contains no error flags.
`tx_forcedisparity` (controls the disparity of the 8B/10B system)	Refer to the *Ports & Parameters* chapter in volume 2 of the *Stratix GX Device Handbook*.

Figure 5–34 shows page 9, the Simulation Libraries page, of the MegaWizard Plug-In Manager for the XAUI protocol set up.

Figure 5–34. MegaWizard Plug-In Manager - altgxb (Page 9)



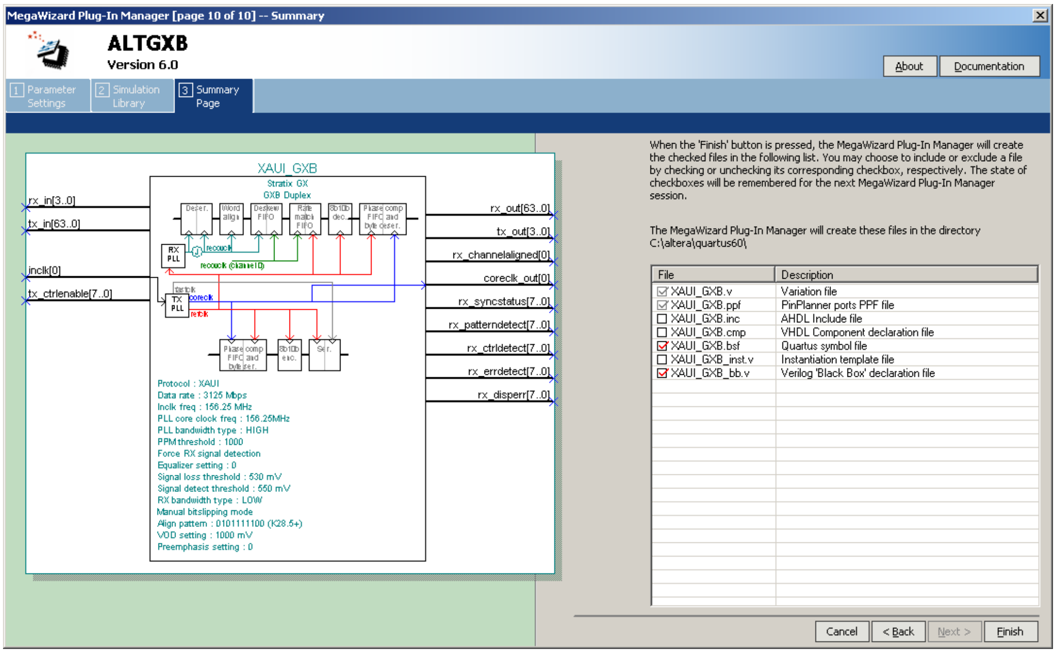
Protocol : XAUI
Data rate : 3125 Mbps
Inclk freq : 156.25 MHz
PLL core clock freq : 156.25MHz
PLL bandwidth type : HIGH
PPM threshold : 1000
Force RX signal detection
Equalizer setting : 0
Signal loss threshold : 530 mV
Signal detect threshold : 550 mV
RX bandwidth type : LOW
Manual bitslipping mode
Align pattern : 0101111100 (K28.5+)
VOD setting : 1000 mV
Preemphasis setting : 0

To properly simulate the generated design files, the following simulation model file(s) are needed

File	Description
stratixgx_mf	Auto-generated simulation library file

Figure 5–35 shows page 10 of the MegaWizard Plug-In Manager for the XAUI protocol set up. You can select optional files on this page. After you make your selections, click **Finish** to generate the files.

Figure 5–35. MegaWizard Plug-In Manager - altgxb (Page 10)



Protocol : XAUI
Data rate : 3125 Mbps
Inclk freq : 156.25 MHz
PLL core clock freq : 156.25MHz
PLL bandwidth type : HIGH
PPM threshold : 1000
Force RX signal detection
Equalizer setting : 0
Signal loss threshold : 530 mV
Signal detect threshold : 550 mV
RX bandwidth type : LOW
Manual bitslipping mode
Align pattern : 0101111100 (K28.5+)
VOD setting : 1000 mV
Preemphasis setting : 0

When the 'Finish' button is pressed, the MegaWizard Plug-In Manager will create the checked files in the following list. You may choose to include or exclude a file by checking or unchecking its corresponding checkbox, respectively. The state of checkboxes will be remembered for the next MegaWizard Plug-In Manager session.

The MegaWizard Plug-In Manager will create these files in the directory C:\altera\quartus60\

File	Description
✓ XAUI_GXB.v	Variation file
✓ XAUI_GXB.ppf	PinPlanner ports PPF file
☐ XAUI_GXB.inc	AHDL Include file
☐ XAUI_GXB.cmp	VHDL Component declaration file
✓ XAUI_GXB.bsf	Quartus symbol file
☐ XAUI_GXB_inst.v	Instantiation template file
✓ XAUI_GXB_bb.v	Verilog 'Black Box' declaration file

Stratix GX Transceiver Merging

A transceiver block contains four transceivers. In a design, an `altgxb` instantiation is placed in one or more transceiver blocks and potentially leaves unused transceivers in a block. For example, a six transceiver instantiation completely fills one transceiver block and half fills a second, taking up two full transceiver blocks. If another instantiation is in the design, it is placed the same way. For example, an instantiation of two transmitters takes up a third transceiver block. Merging two of the partially filled transceiver blocks into one transceiver block reduces the resources used and allows a design to fit into a device with fewer transceiver blocks.

The `altgxb` MegaWizard Plug-In Manager in the Quartus II software has a feature that allows merging of similar quads (transceiver blocks). With a few exceptions, transceiver blocks can be merged if the options chosen in the wizard are the same. The Quartus II software merges the transceiver blocks automatically if you turn on the merging option for each instantiation. Table 5–13 shows the considerations for merging.

Table 5–13. Stratix GX Merging Considerations

Merging Rules	Required to Match Between Transceiver Blocks	Not Required to Match Between Transceiver Blocks

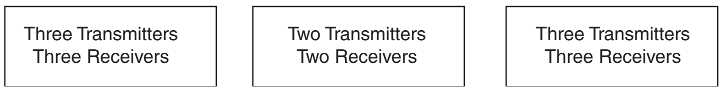
MegaWizard Plug-In Manager options	`USE_8B_10_MODE`
	`USE_DOUBLE_DATA_MODE`
	`CHANNEL_WIDTH`
	`SYNC_MODE`
	`DATA_RATE`
	`TRANSMIT_PROTOCOL`
		VOD
		Pre-emphasis
		Equalization
		Number of transmitters
		Number of receivers

Input control signals must be shared across transceiver blocks and must be from the same source	Clock
	`CRU_CLOCK`
	`PLL_RESET`
	`PLL_ENABLE`	

The merging partial transceiver blocks must reduce the number of transceiver block when merged	The total number of receivers and transmitters must be tx ≤4 x *n* and rx ≤4 x *n*, where *n* is the reduced number of transceiver blocks remaining after merging.	

The Quartus II software does not merge two transceiver blocks if they won’t completely fit into one. It can, however, merge three transceiver blocks into two. Figure 5–36 shows a configuration with three transceiver blocks that can potentially be merged.

Figure 5–36. Three Transceiver Configuration



In Figure 5–36, two transceiver blocks cannot merge because there would be extra channels remaining. But because there are three transceiver blocks, the Quartus II software can merge them into two with no remaining channels.

If you turn on the merging option, the Quartus II software automatically merges transceiver blocks if possible or provides a warning during compilation if merging is not possible. The merging feature can reduce the transceiver block resources used in your design.