AN 537: Implementing UNH-IOL Test Suite Compliance in Arria GX and Stratix II GX Gigabit Ethernet Designs

© September 2008 AN-537-1.0

Introduction

Gigabit Ethernet (GIGE) is the most widely implemented physical and link layer protocol today. In addition to network backbones and data centers, 1000 Mbps connectivity is deployed widely, including switch stacks, servers, and desktops. Altera's Stratix® II GX and Arria® GX FPGAs provide a GIGE solution that is compliant with the IEEE 802.3 1000 BASE-X specification.

GIGE applications include line cards, NIC cards, and switches. Implementing The University of New Hampshire—InterOperability Laboratory (UNH-IOL) test suite compliance in GIGE designs using Altera's Stratix II GX or Arria GX devices help inter-operability with other devices for applications requiring link auto-negotiation.

This application note describes implementation details required for GIGE designs that do not use Altera's TSE MegaCore® and require UNH-IOL compliance. It also describes how to target Stratix II GX and Arria GX GIGE designs for UNH-IOL compliance using the Quartus® II MegaWizard® Plug-In Manager and additional logic where applicable.

Use this application note in conjunction with the following literature:

- *Arria GX ALT2GXB Megafunction User Guide* in volume 2 of the *Arria GX Device Handbook*
- *Arria GX Transceiver Architecture* in volume 2 of the *Arria GX Device Handbook*
- *Arria GX Transceiver Protocol Support and Additional Features* in volume 2 of the *Arria GX Device Handbook*
- *Stratix II GX ALT2GXB Megafunction User Guide* in volume 2 of the *Stratix II GX Device Handbook*
- *Stratix II GX Transceiver Architecture Overview* in volume 2 of the *Stratix II GX Device Handbook*
- *Triple Speed Ethernet MegaCore Function User Guide*

UNH-IOL GIGE Test Suite Compliance Overview

UNH-IOL has developed a GIGE Test Suite for vendors to verify conformance to the GIGE standard. The test suite helps implementers identify problems that GIGE physical layer devices may have in establishing a link and exchanging packets with each other.

The tests do not determine if a product conforms to the IEEE 802.3 standard. Rather, they provide one method to verify that the two devices can exchange packets within the bit error ratio specifications established by the IEEE 802.3 standard when operating over a worst-case compliant channel.

The interoperability test suite focuses on two areas of functionality to simulate a real-world environment: the exchange of packets to produce a packet error ratio that is low enough to meet a desired bit error ratio and the ability to detect and establish a link at the optimal speed between two devices that make up a link segment. A third area covers specific cable testing.

Successful completion of all tests contained in the test suite provides a high level of confidence that the device under test (DUT) will function well and successfully inter-operate with other devices in most environments.

Making the GIGE implementation using Altera's Stratix II GX or Arria GX devices UNH-IOL complaint ensures successful inter-operability with other devices that are UNH-IOL compliant. This is shown in the data path example in Figure 1, where the Altera® device successfully inter-operates with the other devices on the line card that are UNH-IOL compliant.

GIGE Application Examples

GIGE is used in a variety of applications, including the following:

- Connect multiple devices to a local CPU
- Interconnect multiple boards across a backplane for data
- Control signaling between line cards and the host CPU within an embedded system

Figure 1 represents typical networking equipment architecture.

Figure 1. Network Equipment Architecture

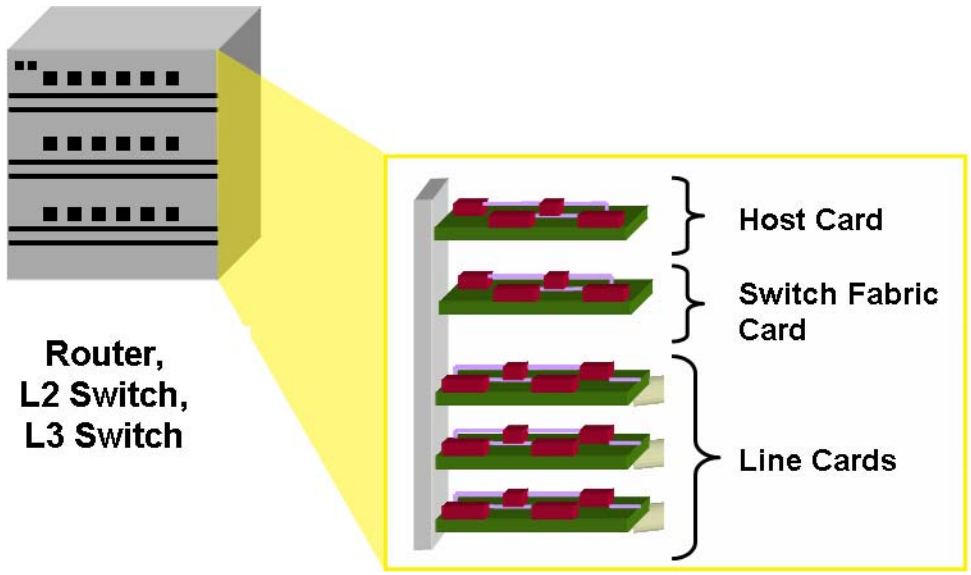
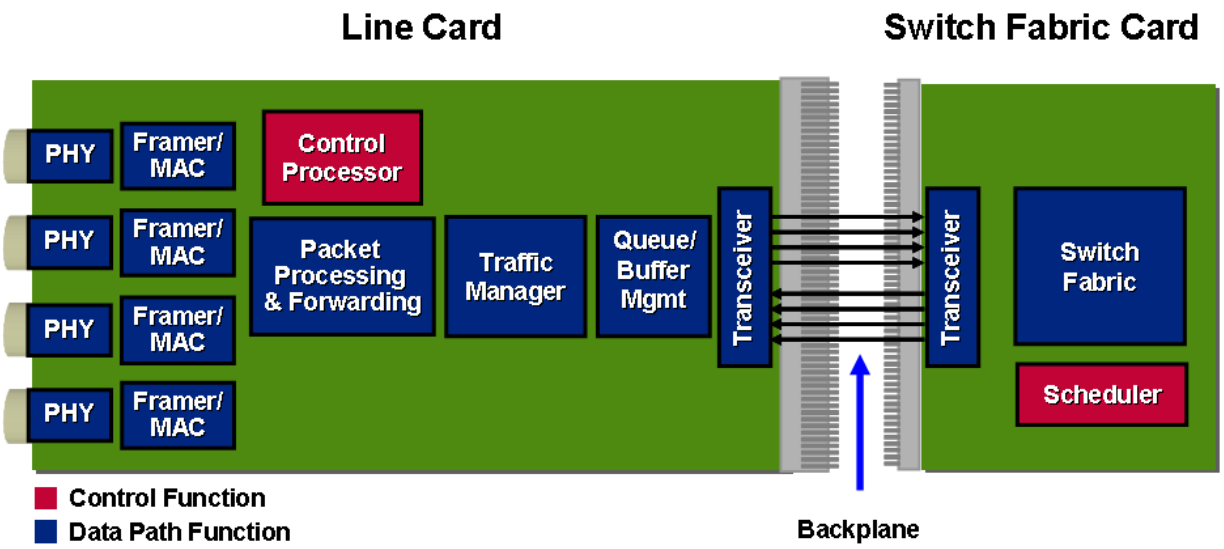


Figure 2 shows a block diagram of a line card used in the network equipment shown in Figure 1.

Figure 2. Block Diagram of a Line Card



The control processor in Figure 2 receives and sends 1.25 Gbps data in GIGE mode. When used in a control plane application such as this, the Stratix II GX or Arria GX device is configured in GIGE functional mode and sends signals to control the other blocks in the line card. The only requirement in such a control plane application is that traffic is 1.25 Gbps GIGE.

For data traffic between the blocks shown in the line card in Figure 2, the different blocks require features such as auto-negotiation to broadcast their capabilities back and forth to their link partner. When Stratix II GX or Arria GX devices are used in data path applications such as those shown in Figure 2, UNH-IOL compliance ensures successful inter-operability with other devices on the line card that are UNH-IOL compliant.

Stratix II GX and Arria GX Transceivers in GIGE Mode

Altera's Stratix II GX and Arria GX transceivers offer native GIGE functional mode support. To configure the devices in GIGE mode, you can use the ALT2GXB megafunction and specify the protocol as GIGE. When configured in GIGE functional mode, the transceivers have built-in circuitry to support the PCS functions of synchronization state machine, 8b/10b encoder/decoder, rate matcher with capability to insert/delete idle characters, and PMA functions defined in the 1000BASE-X IEEE 802.3 specification.

Triple Speed Ethernet MegaCore

Altera's Triple Speed Ethernet (TSE) MegaCore© function provides an integrated Ethernet MAC and PCS solution for Ethernet applications, such as line cards, NIC cards, and switches, operating at 10/100 Mbps (Fast Ethernet) or 1000 Mbps (GIGE). Altera's TSE MegaCore (version 8.0 of the Quartus II software) has been tested and successfully validated by the University of New Hampshire (UNH) InterOperability Lab. All GIGE designs that use Altera's TSE MegaCore are UNH-IOL compliant from version 8.0 of the Quartus II software onward.

For details about Altera's TSE MegaCore, refer to the *Triple Speed Ethernet MegaCore Function User Guide*.

Stratix II GX and Arria GX Transceivers in GIGE Enhanced Mode

You can use the ALT2GXB megafunction to configure the Stratix II GX or Arria GX device in GIGE functional mode. In this mode, the following features are available:

- 8b/10b encoding/decoding
- 10-bit word alignment
- Synchronization state machine
- Upstream transmitter and local receiver clock frequency compensation
  - Rate matcher with capability to insert/delete /I2/ idle ordered sets
- Clock recovery from encoded data at the receiver
- Serialization/deserialization
- The ALT2GXB megafunction included with version 8.0 and later of the Quartus II software offers a GIGE Enhanced mode sub-protocol within GIGE mode. GIGE Enhanced mode offers the following additional features and ports for designs targeting UNH-IOL compliance.

  For details about the ALT2GXB megafunction port list, refer to the *Arria GX ALT2GXB Megafunction User Guide*.

  - Rate matcher with capability to insert/delete the first two bytes of /C2/ configuration ordered sets
  - 7-bit word alignment
  - `rx_rmfifodatainserted` port to indicate insertion of ordered sets by the rate matcher
  - `rx_rmfifodatadeleted` port to indicate deletion of ordered sets by the rate matcher
  - `rx_runningdisp` port to be used by carrier detect logic

The following section describes how to achieve UNH-IOL compliance in GIGE designs that do not use Altera's TSE MegaCore.

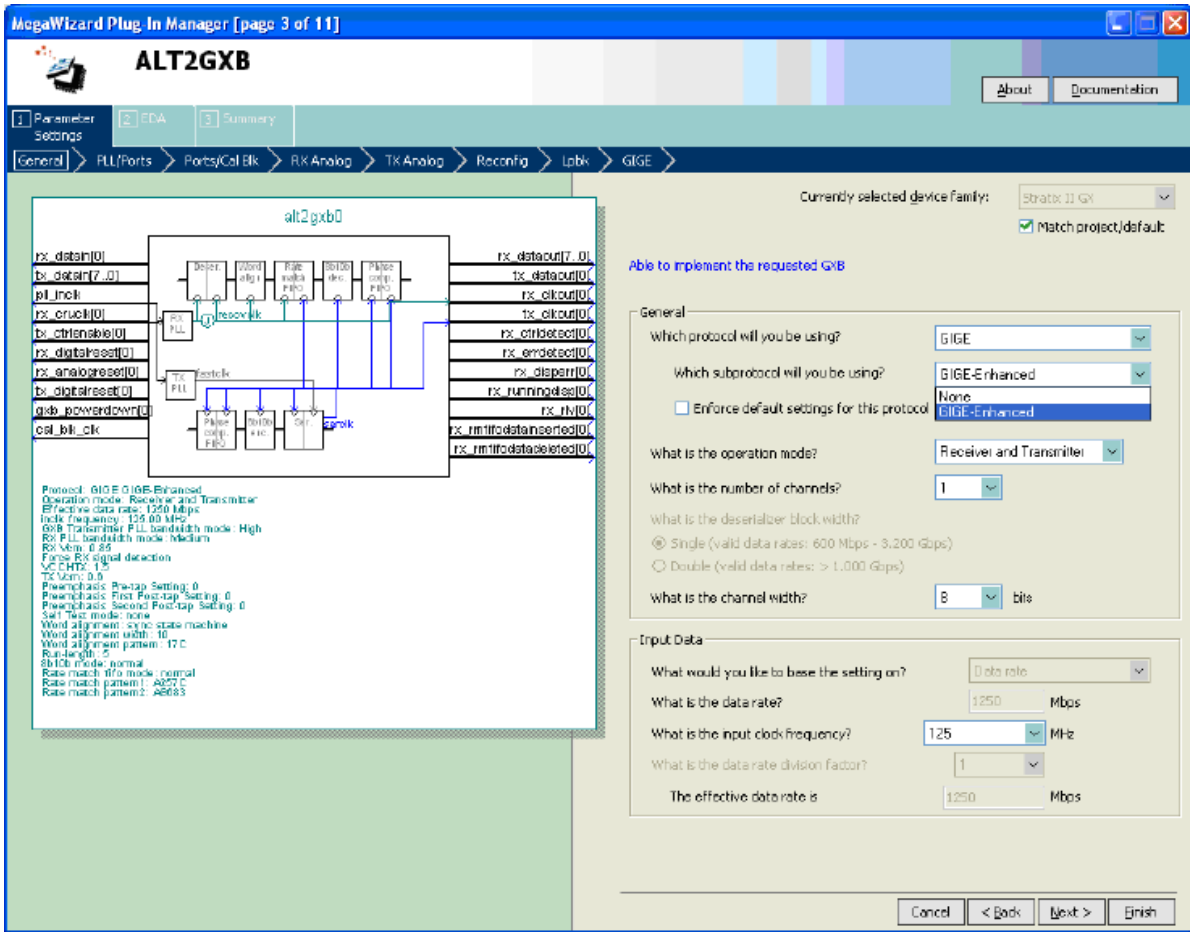
Steps to Implement UNH-IOL Test Suite Compliance Using the ALT2GXB Megafunction

This section describes implementation details required to make your GIGE design UNH-IOL test suite compliant.

Select GIGE Enhanced Mode

To configure the Stratix II GX or Arria GX device in GIGE Enhanced mode, choose **GIGE Enhanced mode**, as shown in Figure 3, in the ALT2GXB MegaWizard Plug-In Manager.

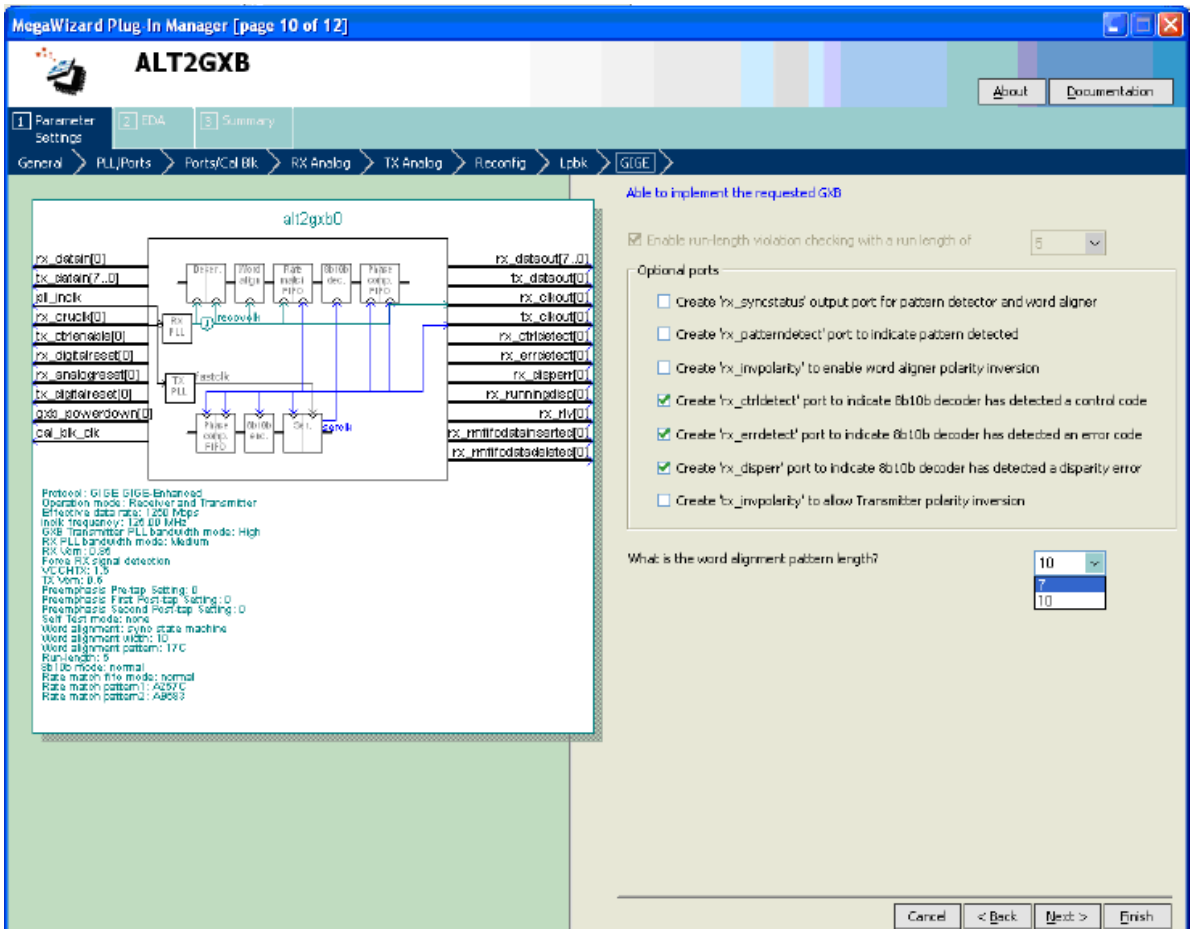
Figure 3. GIGE Enhanced Mode



Select 7-Bit Word Alignment

To enable 7-bit word alignment in GIGE Enhanced mode, select the **GIGE** panel under **Parameter Settings** and choose the word alignment pattern length, as shown in Figure 4, in the ALT2GXB MegaWizard Plug-In Manager.

Figure 4. Word Alignment Pattern Length



Features for UNH-IOL Test Suite Compliance Requiring Custom Implementation

The first two steps configure the Stratix II GX or Arria GX device in GIGE Enhanced mode. To make your design UNH-IOL compliant, additional steps are required for the following features:

- Auto-negotiation feature with UNH-IOL compliance
- `rx-syncstatus` signal for UNH-IOL compliance
- UNH-IOL Carrier Detect

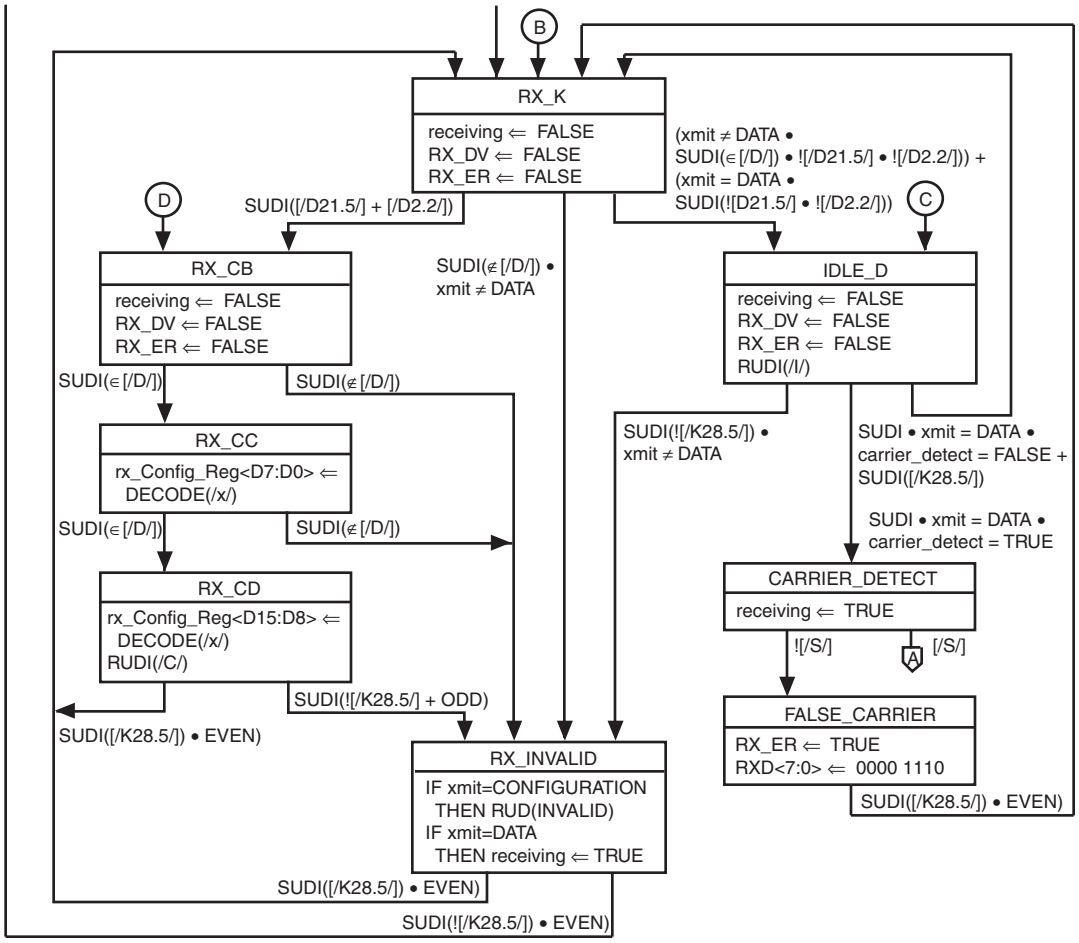
Auto-Negotiation Feature

Auto-negotiation is a function that can be started when link synchronization is acquired during system start up. During auto-negotiation, the PCS function advertises its device features and exchanges them with a link partner device. UNH-IOL test suite compliance for GIGE requires that a link partner be able to continuously send and receive /C1/ and /C2/ configuration ordered sets to and from the Altera device until it sees an acknowledgement from the device. During this time, no /I1/, /I2/ ordered sets are sent across the link.

When the Stratix II GX or Arria GX device is configured in GIGE Enhanced mode, the rate matcher is capable of inserting or deleting the first two bytes of /C2/ ordered sets during the auto-negotiation phase. However, the insertion or deletion of the first two bytes of /C2/ ordered sets could cause the GIGE Soft-IP PCS state machine to function incorrectly.

Figure 5 shows the receiver PCS state diagram as described in the IEEE802.3 specification. To successfully implement the auto-negotiation feature and pass UNH-IOL compliance, modify the PCS state machine logic you implemented in the Altera FPGA fabric as described in the following sections.

Figure 5. IEEE 802.3 GIGE Receiver PCS State Diagram *(Note 1)*



Note to Figure 5:

(1) Image taken from IEEE Std. 802.3™ -2002, IEEE Standard for Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications.

In the following five cases, the PCS state machine must be modified for auto-negotiation only and for auto-negotiation with UNH-IOL compliance.

- **Case 1**—Insertion of first two bytes of /C2/ ordered set during auto-negotiation
- **Case 2**—Deletion of first two bytes of /C2/ ordered set during auto-negotiation

- **Case 3**—Deletion of first two bytes of /C2/ ordered set following an /I2/ ordered set during auto-negotiation
- **Case 4**—Deletion /I2/ ordered set during auto-negotiation
- **Case 5**—Insertion of /I2/ ordered set during auto-negotiation

Cases 1, 2, and 3 are applicable for designs requiring the auto-negotiation feature and UNH-IOL test suite compliance; cases 4 and 5 are applicable for designs requiring UNH-IOL compliance only.

Case 1 – Auto-Negotiation and UNH-IOL Test Suite Compliance

Insertion of first two bytes of /C2/ ordered set during Auto-Negotiation

If the local receiver clock is running at a higher parts per million (ppm) frequency compared to the upstream transmitter clock, the rate match logic inserts the configuration ordered set /K28.5/D2.2/ during auto-negotiation, as shown in Figure 6.

Figure 6. Case 1: Incorrect Receiver State Machine Transitions

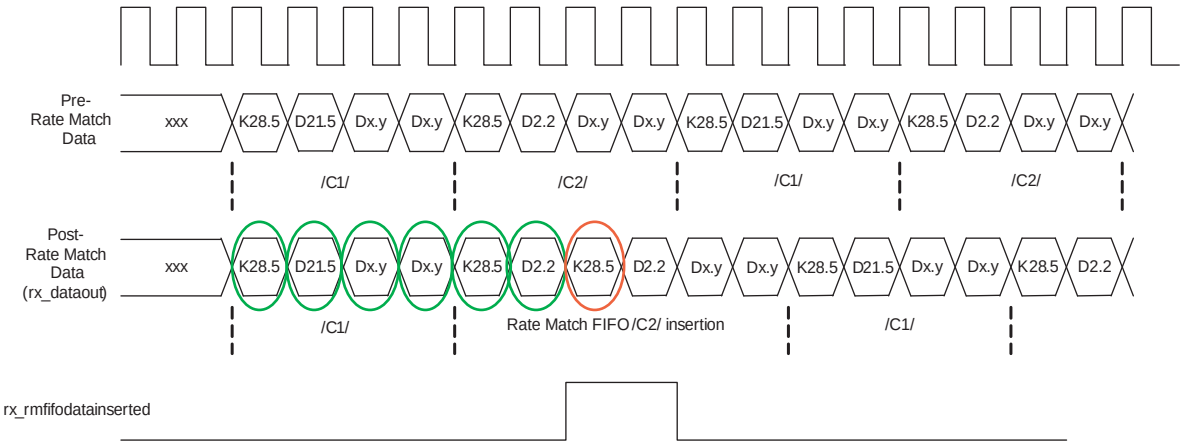


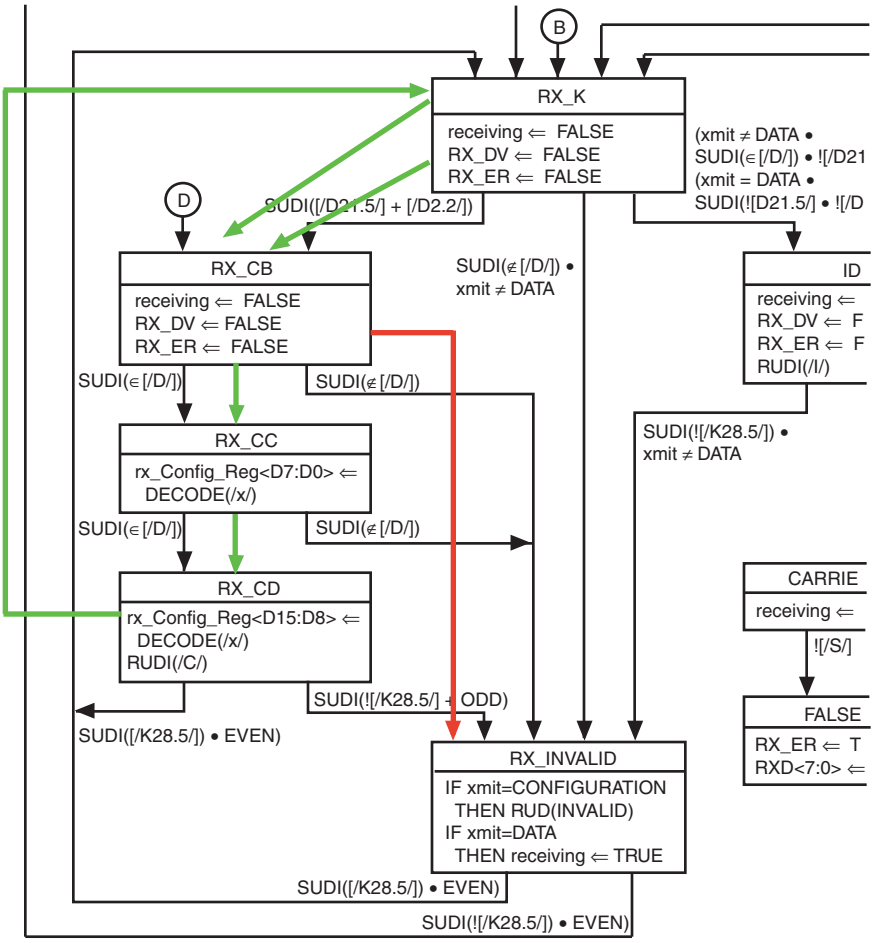
Table 1 shows the receiver state machine transitions for the case when the first two bytes of /C2/ ordered set are inserted during auto-negotiation.

Table 1. Case 1: Incorrect Receiver PCS State Machine Transitions

Clock	N	n+1	n+2	n+3	n+4	n+5	n+6	n+7	n+8	n+9
rx_dataout	K28.5	D21.5	Dx.y	Dx.y	K28.5	D2.2	K28.5	D2.2	Dx.y	Dx.y
rx_rmfifodatainserted	0	0	0	0	0	0	1	1	0	0
RX PCS SM State	RX_K	RX_CB	RX_CC	RX_CD	RX_K	RX_CB	RX_INVALID	x	x	x

In cycle n+6, the receiver PCS state machine is expecting to receive a valid data code group. Due to the insertion of the first two bytes of the /C2/ ordered set during cycles n+6 and n+7, the state machine goes into RX_INVALID state. This could cause the auto-negotiation phase to restart erroneously, as shown in Figure 7 by the green and red arrows corresponding to the post-rate-matched data in Figure 6.

Figure 7. Case 1: Erroneous Restart During the Auto-Negotiation Phase *(Note 1)*



Note to Figure 7:

(1) Image taken from IEEE Std. 802.3™ -2002, IEEE Standard for Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications.

To avoid this failure, Altera recommends modifying the receiver PCS state machine logic as follows:

If the current state is `RX_CB`, the next state (priority encoded) is:

- `RX_CB`—if the received data byte is valid, `rx_rmfifodatainserted` is high, and `rx_rmfifodatadeleted` is low; otherwise
- `RX_INVALID`—if the received data byte is a /K/ control code group or an invalid data code group (disparity error or code group violation); otherwise
- `RX_CC`

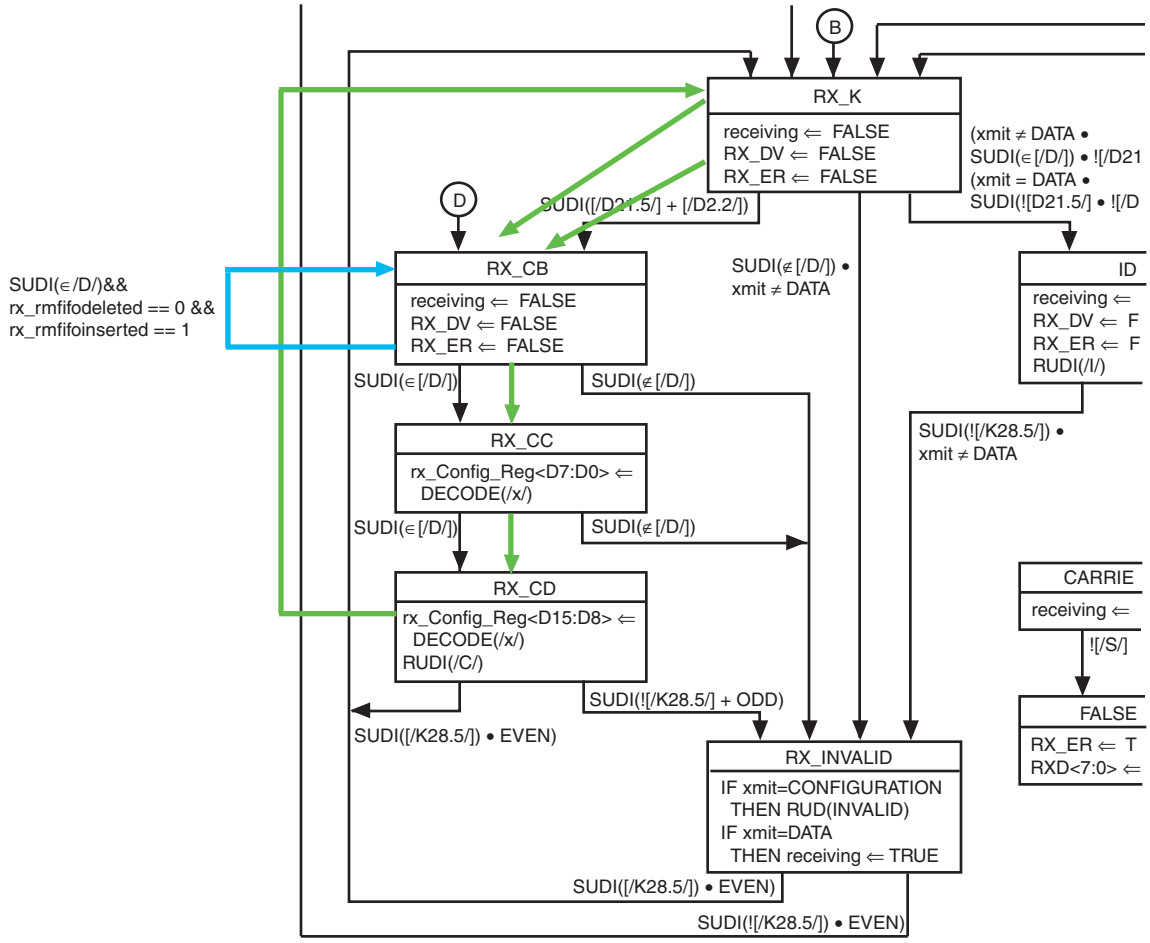
Table 2 shows the modified state machine.

Table 2. Case 1: Modified Receiver PCS State Machine

Clock	n	n+1	n+2	n+3	n+4	n+5	n+6	n+7	n+8	n+9
rx_dataout	K28.5	D21.5	Dx.y	Dx.y	K28.5	D2.2	K28.5	D2.2	Dx.y	Dx.y
rx_rmfifodatainserted	0	0	0	0	0	0	1	1	0	0
RX PCS SM State	RX_K	RX_CB	RX_CC	RX_CD	RX_K	RX_CB	RX_CB	RX_CB	RX_CC	RX_CD

Figure 8 shows the modified receiver PCS state machine. The modified path is shown with blue arrows.

Figure 8. Case 1: Modified Receiver PCS State Machine *(Note 1)*



Note to Figure 8:

(1) Image taken from IEEE Std. 802.3™ -2002, IEEE Standard for Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications.

This modification to the receiver PCS state machine is required to successfully implement the auto-negotiation feature in a GIGE system and to pass UNH-IOL compliance.

Case 2 – Auto-Negotiation and UNH-IOL Test Suite Compliance

Deletion of first two bytes of /C2/ ordered set during auto-negotiation

If the local receiver clock is running at a lower ppm frequency compared to the upstream transmitter clock, the rate match logic deletes /K28.5/D2.2/ during auto-negotiation, as shown in Figure 9.

Figure 9. Case 2: Deletion of First Two Bytes of /C2/ Ordered Set During Auto-Negotiation

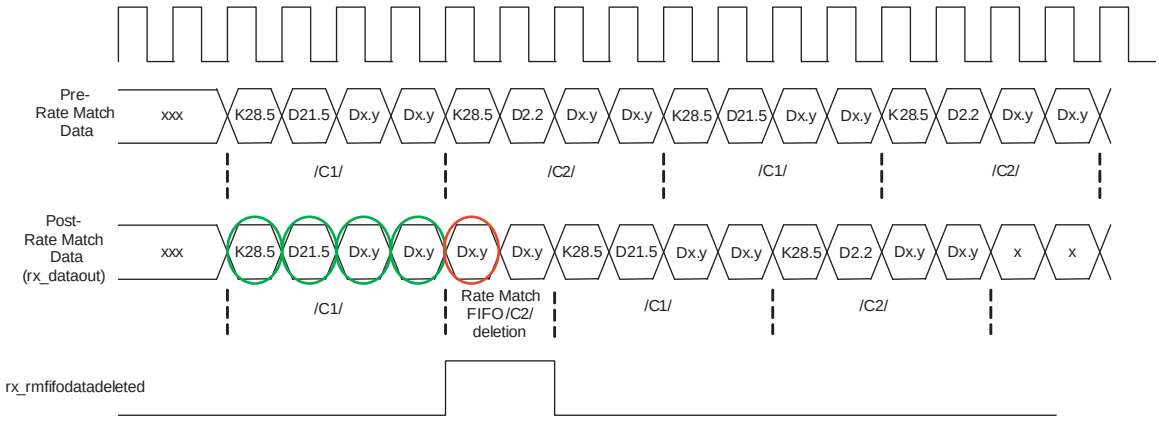


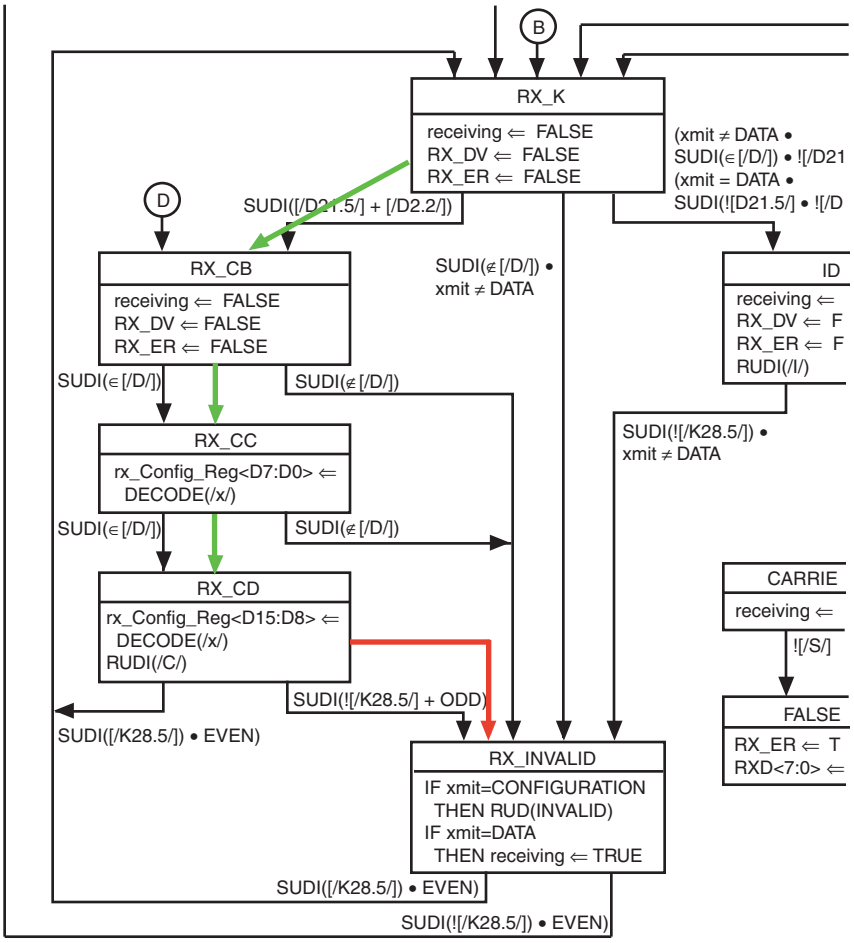
Table 3 shows the receiver state machine transitions for the case when the first two bytes of /C2/ ordered set are deleted during auto-negotiation.

Table 3. Case 2: Incorrect Receiver PCS State Machine Transitions

Clock	n	n+1	n+2	n+3	n+4	n+5	n+6	n+7	n+8	n+9
rx_dataout	K28.5	D21.5	Dx.y	Dx.y	Dx.y	Dx.y	K28.5	D21.5	Dx.y	Dx.y
rx_rmfifodatadeleted	0	0	0	0	1	1	0	0	0	0
RX PCS SM State	RX_K	RX_CB	RX_CC	RX_CD	RX_INVALID	x	x	x	x	x

In cycle n+4, the receiver PCS state machine is expecting to receive a /K28.5/ control code group. Due to the deletion of the first two bytes of the /C2/ ordered set during cycles n+4 and n+5, the state machine sees a /Dx.y/ data code group in cycle n+4. This causes the state machine to go into RX_INVALID state. This could cause the auto-negotiation phase to restart erroneously, as shown in Figure 10 by the green and red arrows corresponding to the post-rate matched data, shown in Figure 9.

Figure 10. Case 2: Erroneous Restart During Auto-Negotiation Phase *(Note 1)*



Note to Figure 10:

(1) Image taken from IEEE Std. 802.3™ -2002, IEEE Standard for Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications.

To avoid this failure, Altera recommends modifying the receiver PCS state machine logic as follows:

If the current state is `RX_CD`, the next state (priority encoded) is:

- `RX_CC` — if the received data byte is a valid /D/ control code group, `rx_rmfifodatadeleted` is high, and `rx_rmfifodatainserted` is low; otherwise
- `RX_K` — if the received data byte is a /K28.5/ control code group on an even code group boundary (`rx_even` = TRUE); otherwise
- `RX_INVALID`

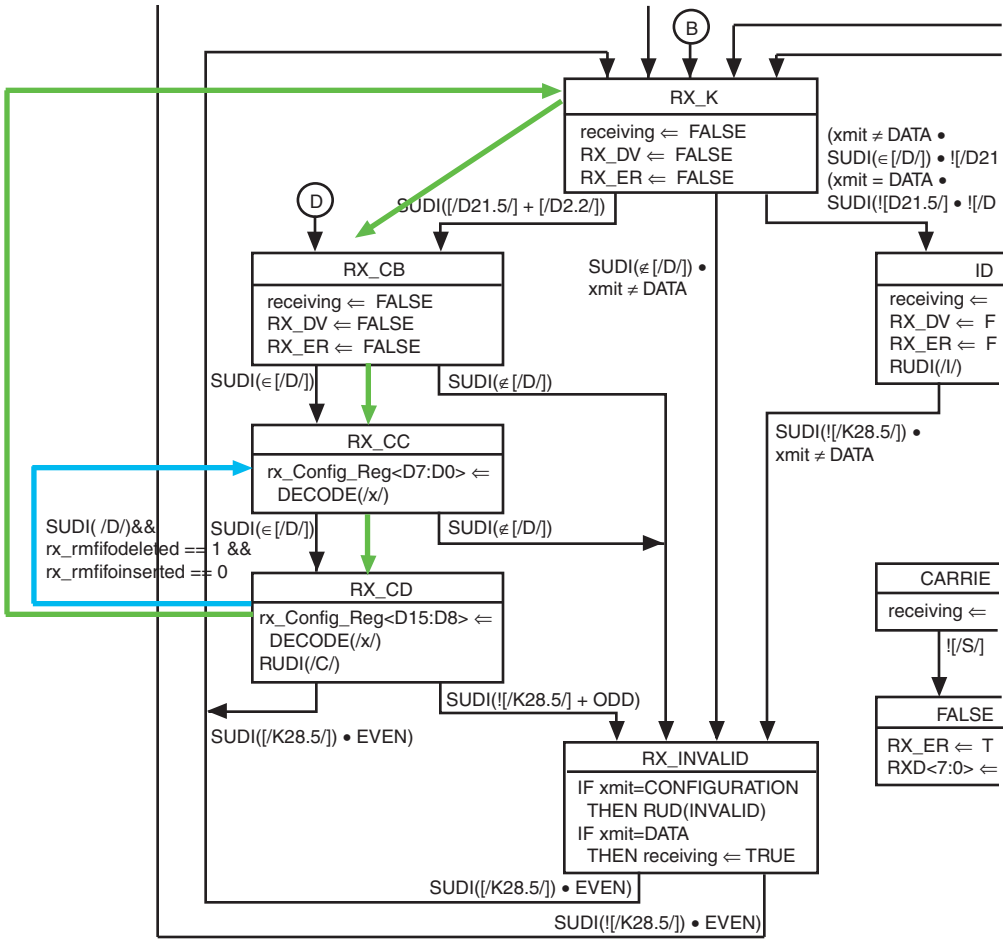
Table 4 shows the modified state machine.

Table 4. Case 2: Modified Receiver PCS State Machine

Clock	n	n+1	n+2	n+3	n+4	n+5	n+6	n+7	n+8	n+9
rx_dataout	K28.5	D21.5	Dx.y	Dx.y	Dx.y	Dx.y	K28.5	D21.5	Dx.y	Dx.y
rx_rmfifodatadeleted	0	0	0	0	1	1	0	0	0	0
RX PCS SM State (Figure 6)	RX_K	RX_CB	RX_CC	RX_CD	RX_CC	RX_CD	RX_K	RX_CB	RX_CC	RX_CD

Figure 11 shows the modified receiver PCS state machine. The modified path is highlighted with blue arrows.

Figure 11. Case 2: Modified Receiver PCS State Machine *(Note 1)*



Note to Figure 11:

(1) Image taken from IEEE Std. 802.3™ -2002, IEEE Standard for Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications.

This modification to the receiver PCS state machine is required to successfully implement the auto-negotiation feature in a GIGE system and to pass UNH-IOL compliance.

Case 3 – Auto-negotiation and UNH-IOL Test Suite Compliance

Deletion of first two bytes of /C2/ ordered set following an /I2/ ordered set during auto-negotiation

If the local receiver clock is running at a lower ppm frequency compared to the upstream transmitter clock, the rate match logic deletes /K28.5/D2.2/ during auto-negotiation. If the first two bytes of the /C2 / ordered set following an idle ordered set is deleted, it could take the receiver PCS state machine to RX_INVALID state, as shown in Figure 12.

Figure 12. Case 3: Deletion of First Two Bytes of /C2/ Ordered Set Following an /I2/ Ordered Set

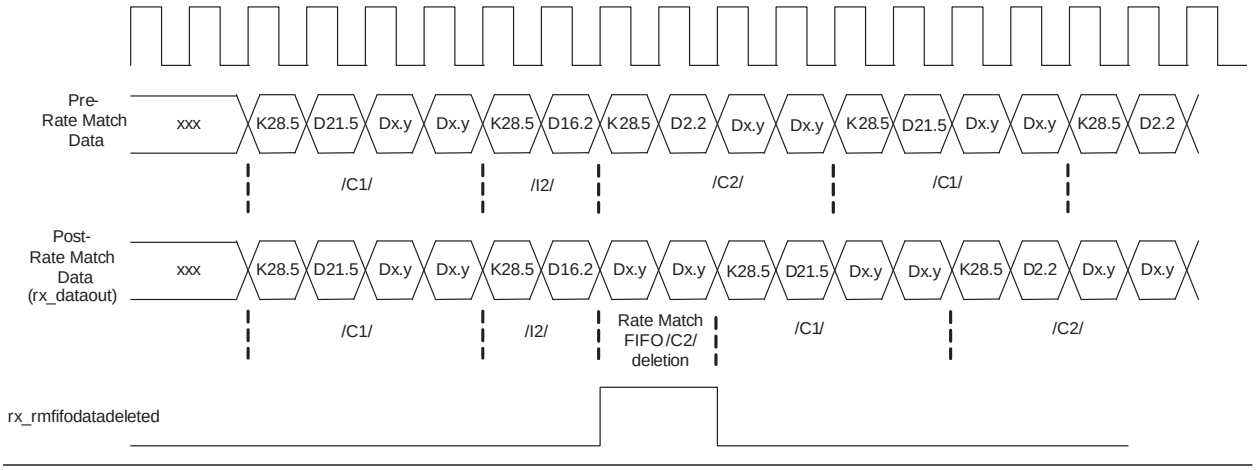


Table 5 shows the receiver state machine transitions for the case when the first two bytes of /C2/ ordered set following an /I2/ ordered set are deleted during auto-negotiation. In IDLE_D state during cycle n+5, since xmit ≠ DATA and a non-/K28.5/ is received, the state machine transitions into the RX_INVALID state.

Table 5. Case 3: Incorrect Receiver PCS State Machine Transitions

Clock	K28.5	n+1	n+2	n+3	n+4	n+5	n+6	n+7	n+8	n+9
rx_dataout	K28.5	D21.5	Dx.y	Dx.y	K28.5	D16.2	Dx.y	Dx.y	K28.5	D2.2
rx_rmfifodatadeleted	0	0	0	0	0	0	1	1	0	0
RX PCS SM State	RX_K	RX_CB	RX_CC	RX_CD	RX_K	IDLE_D	RX_INVALID	x	X	x

To avoid this failure, Altera suggests modifying the receiver PCS state machine to add the following state transition logic:

If the current state is IDLE_D, the next state (priority encoded) is:

- RX_CC — if the received data byte is a valid /D/ control code group, rx_rmfifodatadeleted is high, rx_rmfifoinserted flag is low, and rx_even = TRUE; otherwise
- Follow the state transitions as outlined in the IEEE802.3 specification

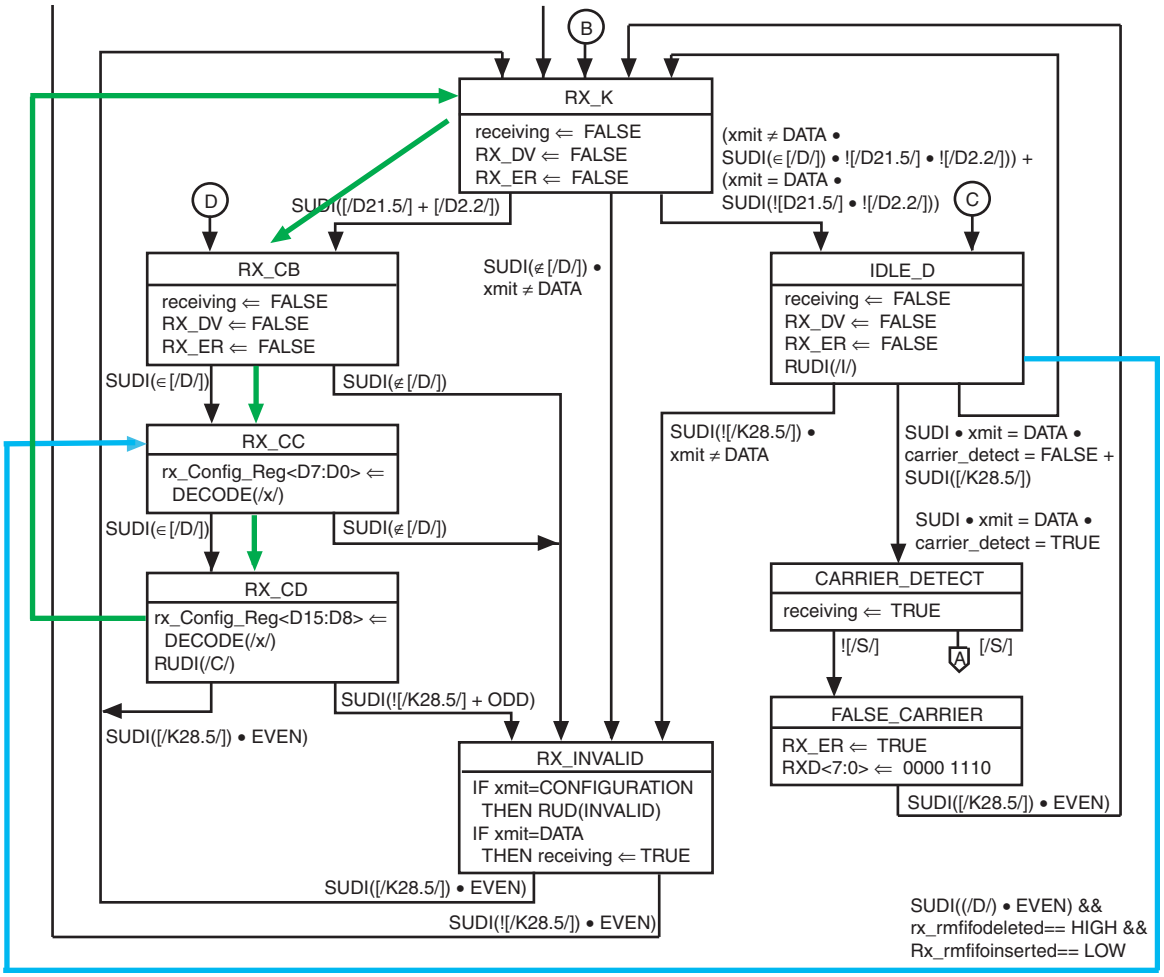
Table 6 shows the modified state machine.

Table 6. Case 3: Modified Receiver PCS State Machine

Clock	n	n+1	n+2	n+3	n+4	n+5	n+6	n+7	n+8	n+9
rx_dataout	K28.5	D21.5	Dx.y	Dx.y	K28.5	D16.2	Dx.y	Dx.y	K28.5	D2.2
rx_rmfifodatadeleted	0	0	0	0	0	0	1	1	0	0
RX PCS SM State (Figure 6)	RX_K	RX_CB	RX_CC	RX_CD	RX_K	IDLE_D	RX_CC	RX_CD	RX_K	RX_CB

Figure 13 shows the modified receiver PCS state machine. The modified path is shown with blue arrows.

Figure 13. Case 3: Modified PCS State Machine *(Note 1)*



Note to Figure 13:

(1) Image taken from IEEE Std. 802.3™ -2002, IEEE Standard for Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications.

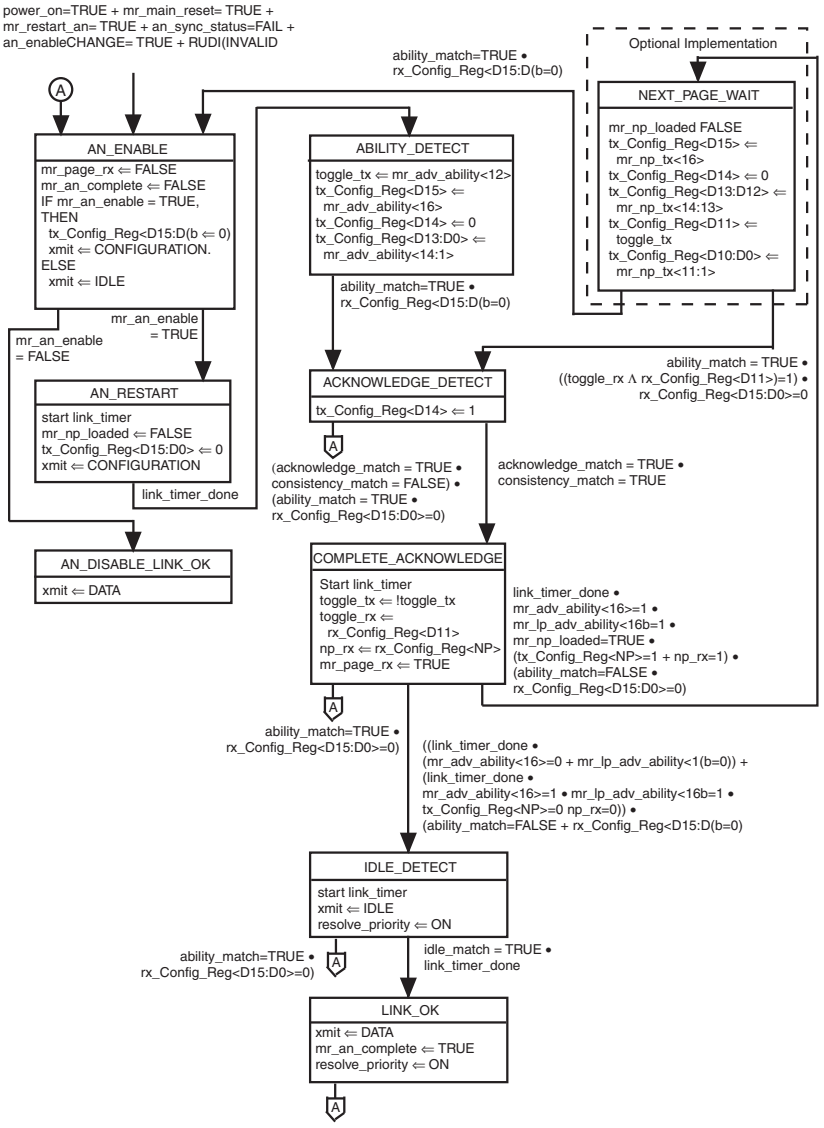
This modification to the receiver PCS state machine is required to successfully implement the auto-negotiation feature in a GIGE system and to pass UNH-IOL compliance.

Case 4 – UNH-IOL Test Suite Compliance Only

Deletion of /I2/ ordered set during auto-negotiation

During the ABILITY_DETECT state of the auto-negotiation state machine, the receiver must receive three consecutive non-zero configuration /C/ ordered sets (ability_match = TRUE) to move into the ACKNOWLEDGE_DETECT state, as shown in Figure 14.

Figure 14. Case 4: IEEE 802.3 Receiver PCS State Machine Showing ABILITY_DETECT *(Note 1)*



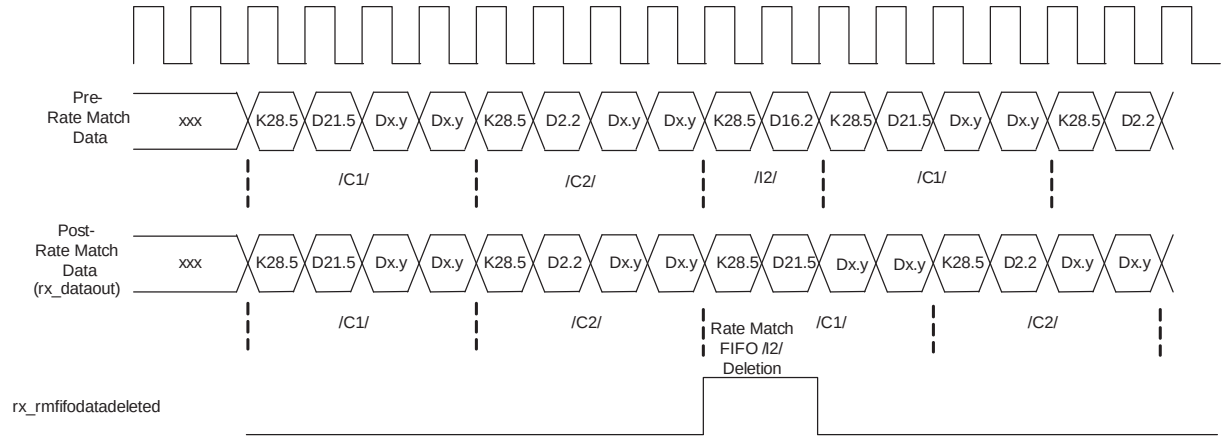
Note to Figure 14:

(1) Image taken from IEEE Std. 802.3™ -2002, IEEE Standard for Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications.

Test 37.3.1 in the UNH-IOL test suite sends test patterns with one, two, three, and more consecutive non-zero /C/ ordered sets and checks if the auto-negotiation state machine moves into the ACKNOWLEDGE_DETECT state. The test suite sends Idle ordered sets in between /C/ ordered sets to create these test patterns. For example, to create a test pattern with two consecutive non-zero /C/ ordered sets, the test suite sends two /C/ ordered sets followed by a stream of Idle /I/ ordered sets repeatedly.

If the test suite sends two consecutive /C/ ordered sets followed by an /I2/ ordered set repeatedly and if the rate matcher deletes the /I2/ ordered set for clock rate compensation, the auto-negotiation state machine sees four consecutive /C/ ordered sets. This incorrectly takes the state machine into ACKNOWLEDGE_DETECT state and the UNH-IOL test fails, as shown in Figure 15.

Figure 15. Case 4: Deletion of /I2/ Ordered Set in Test 37.3.1 in ABILITY_DETECT State

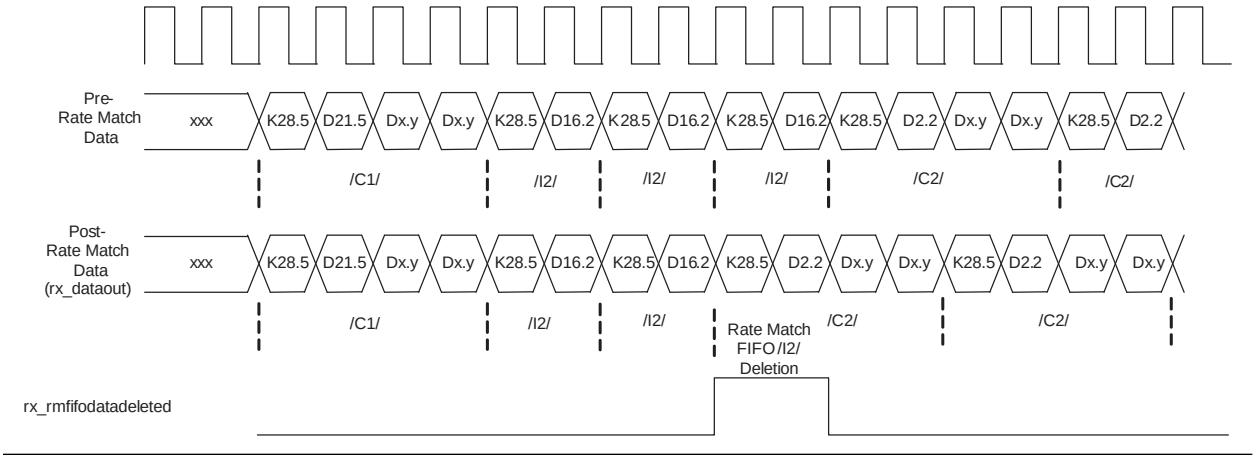


A second scenario in which deletion of an /I2/ ordered set may cause UNH-IOL auto-negotiation test suite failure is described below.

After the two devices forming the GIGE link have negotiated and acknowledged their abilities (COMPLETE_ACKNOWLEDGE state), the auto-negotiation state machine moves into the IDLE_DETECT state, as shown in Figure 14. In this state, each device transmits consecutive Idle ordered sets and monitors the number of Idle ordered sets received until the link timer expires. At the end of the link timer, if the device has received three consecutive Idle ordered sets (Idle_Match = TRUE), the auto-negotiation state machine must move to the LINK_OK state.

Test 37.3.4 in the UNH-IOL test suite sends test patterns with one, two, three, or more consecutive Idle ordered sets in between configuration ordered sets and checks if the auto-negotiation state machine moves into the LINK_OK state. It expects the state machine to move into the LINK_OK state only on reception of three or more consecutive Idle ordered sets. However, when it sends the test pattern that has three consecutive Idle ordered sets, there is a possibility of the rate matcher deleting one of the three /I2/ ordered sets for clock rate compensation. Due to deletion of an /I2/ ordered set, the three Idle ordered sets transmitted by the UNH-IOL tester is seen as two Idle ordered sets by the auto-negotiation state machine, keeping the state machine in the IDLE_DETECT state. This results in UNH-IOL failure, as shown in Figure 16.

Figure 16. Case 4: Deletion of /I2/ Ordered Set in Test 37.3.2 in IDLE_DETECT State



To avoid these two failures, Altera recommends incrementing the Idle count appropriately when the `rx_rmfifodatadeleted` flag is asserted high, corresponding to the deleted /I2/ ordered set. The `rx_rmfifodatadeleted` flag goes high even when the first two bytes of /C2/ get deleted. You must implement the Idle counter that only increments the count when the `rx_rmfifodatadeleted` flag goes high in response to a deleted /I2/ ordered set.

This modification to the receiver PCS state machine is not required if you are not targeting UNH-IOL compliance. In a real GIGE system, the situation in which only three consecutive Idle ordered sets are received to move into the `LINK_OK` state does not arise, as both link partners transmit a stream of Idle ordered sets until the link timer expires.

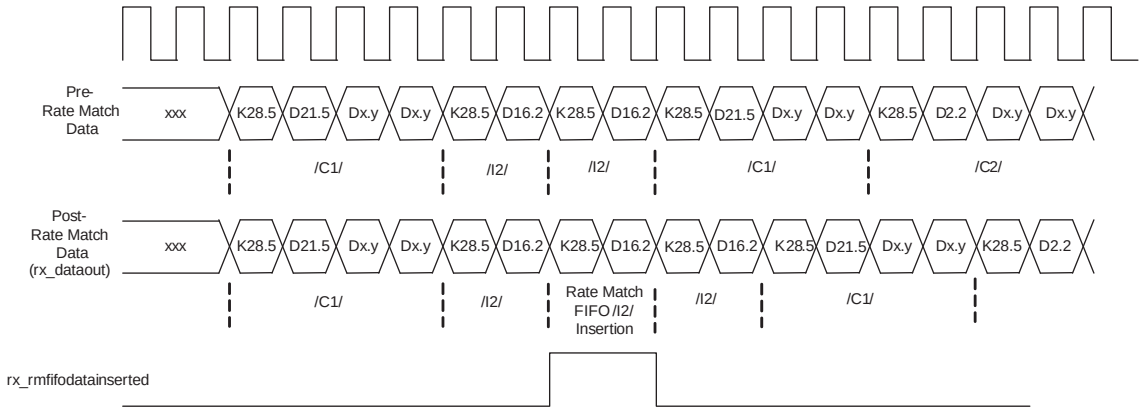
Case 5 – UNH-IOL Test Suite Compliance Only

Insertion of /I2/ ordered set during auto-negotiation

After the two devices forming the GIGE link have negotiated and acknowledged their abilities (the `COMPLETE_ACKNOWLEDGE` state), the auto-negotiation state machine moves into the `IDLE_DETECT` state. In this state, each device transmits consecutive Idle ordered sets and monitors the number of Idle ordered sets received until the link timer expires. At the end of the link timer, if the device has received three consecutive Idle ordered sets (`Idle_Match` = TRUE), the auto-negotiation state machine must move to the `LINK_OK` state, as shown in Figure 14 on page 16.

Test 37.3.4 in the UNH-IOL test suite sends test patterns with one, two, three, or more consecutive Idle ordered sets in between configuration ordered sets and checks if the auto-negotiation state machine moves into the `LINK_OK` state. It expects the state machine to move into the `LINK_OK` state only on reception of three or more consecutive Idle ordered sets. However, when it sends the test pattern that has two consecutive Idle ordered sets, there is a possibility of the rate matcher inserting an /I2/ ordered set for clock rate compensation. Due to insertion of an /I2/ ordered set, the two Idle ordered sets transmitted by the UNH-IOL tester are seen as three Idle ordered sets by the auto-negotiation state machine taking it into the `LINK_OK` state. This results in UNH-IOL failure, as shown in Figure 17.

Figure 17. Case 5: Insertion of /I2/ Ordered Set in Test 37.3.4 in IDLE_DETECT State



To avoid this failure, Altera recommends not incrementing the Idle count if an /I2/ ordered set is received on the `rx_dataout` port with the `rx_rmfifodatainserted` flag asserted high. In other words, you must modify your design to ignore the /I2/ ordered set if the `rx_rmfifodatainserted` flag is high.

This modification to the receiver PCS state machine is not required if you are not targeting UNH-IOL compliance. In a real GIGE system, the situation in which only two consecutive Idle ordered sets are received in the `Idle_Detect` state of the auto-negotiation state machine does not arise as both link partners transmit a stream of Idle ordered sets until the link timer expires.

rx_syncstatus Signal for UNH-IOL Compliance

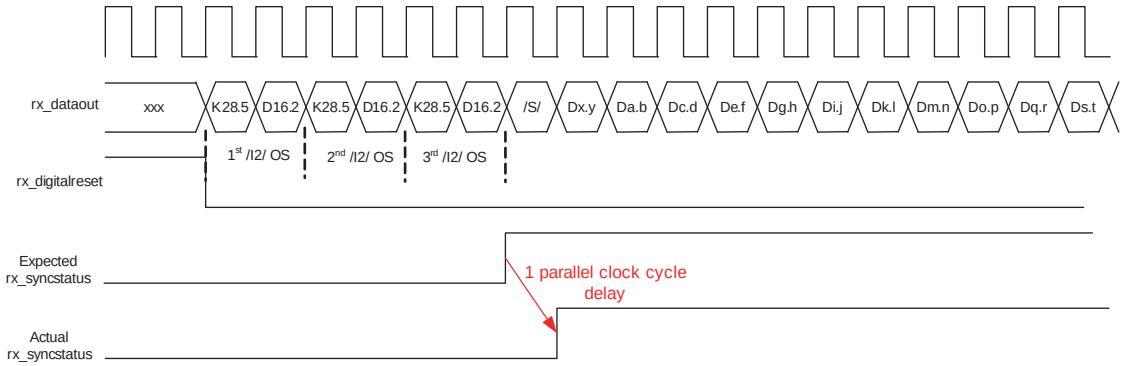
The word aligner in the Stratix II GX and Arria GX data path implements the synchronization state machine specified in Clause 36.2.5.2.6 of the IEEE 802.3-2002 standard. The standard requires the receiver to acquire synchronization on reception of three Idle ordered sets (/I1/ or /I2/).

Stratix II GX and Arria GX transceivers indicate synchronization status on the `rx_syncstatus` port. You may use this signal as an indication of a synchronized link in the receiver PCS state machine implemented in the GIGE protocol IP. The receiver PCS state machine might expect the `rx_syncstatus` signal to assert, along with the byte received immediately after the third Idle ordered set. However, `rx_syncstatus` gets asserted one clock cycle after the byte received immediately after the third Idle ordered set. If the receiver PCS state machine uses the `rx_syncstatus` signal to decide if the data on the `rx_dataout` port is valid, the byte immediately following the third /I2/ ordered set is considered invalid. This could result in a UNH-IOL *Acquire Synchronization* failure.

Consider the example scenario shown in Figure 18. Three /I2/ ordered sets are received immediately after coming out of reset. After the three /I2/ ordered sets, a start of packet /S/ is received followed by data bytes /Dx.y/, /Da.b/, and /Dc.d/.

After the synchronization state machine detects the third /I2/ ordered set, it asserts the `rx_syncstatus` signal to indicate that synchronization is acquired. The `rx_syncstatus` signal does not get asserted along with /S/ following the /D16.2/ of the third /I2/ ordered set. It gets asserted one clock cycle later, along with /Dx.y/. If your receiver PCS state machine uses the `rx_syncstatus` signal to validate the data on the `rx_dataout` port, the /S/ following the third /I2/ ordered set is neglected by the state machine. This could result in a UNH-IOL *Acquire Synchronization* failure.

Figure 18. rx_syncstatus Behavior in GIGE Mode



To achieve UNH-IOL compliance, simply delay `rx_dataout` and other status signals by one clock cycle to match the latency of the `rx_syncstatus` signal, as shown in Figure 19.

Figure 19. Modified rx_syncstatus Behavior for UNH-IOL Compliance

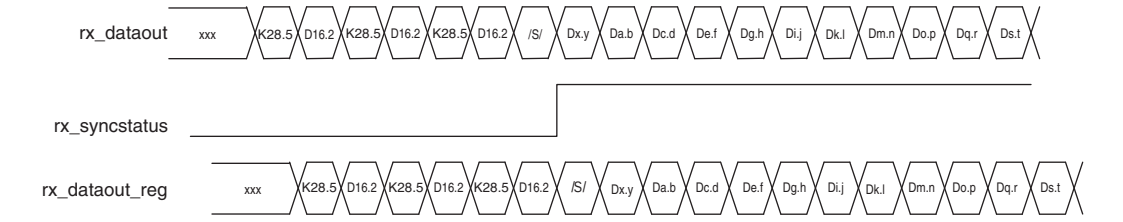


Figure 19 shows the `rx_dataout` signal delayed by one clock cycle. Delay all status signals that you intend to use in the logic by one clock cycle to keep the signals synchronous to the data. The status signals include:

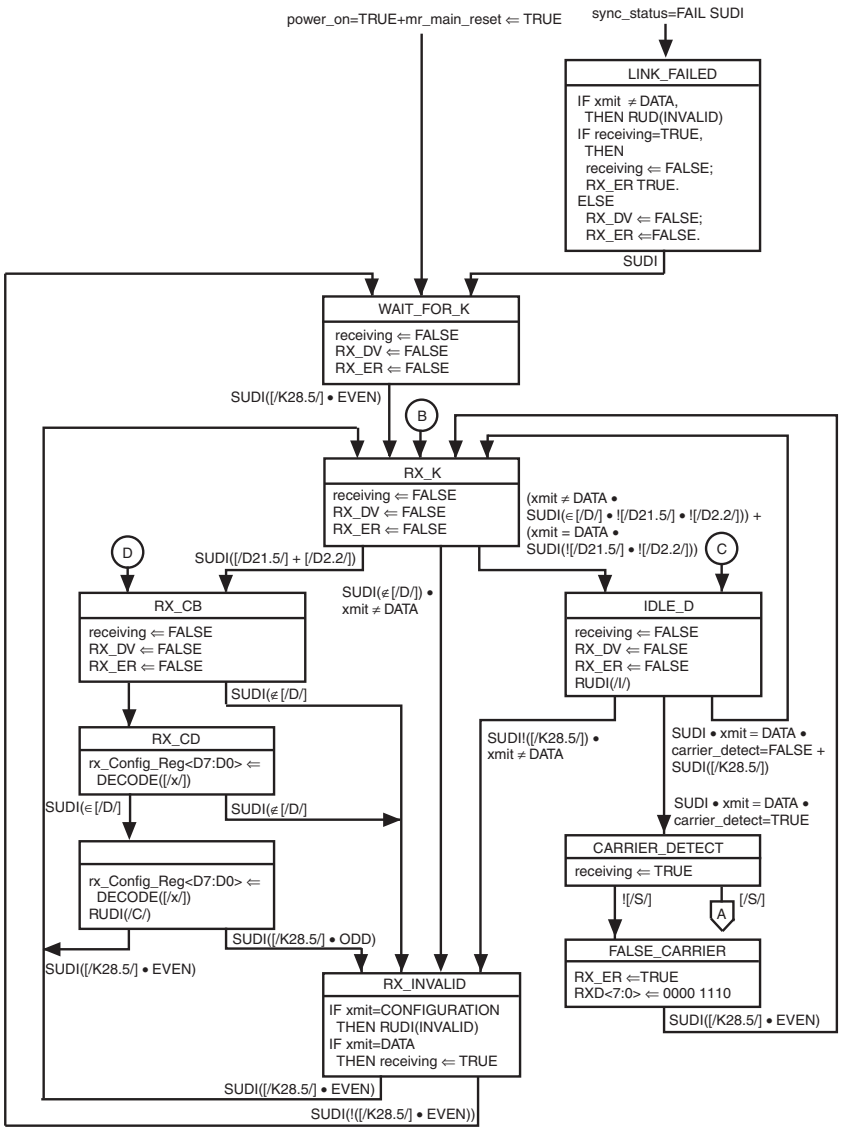
- `rx_ctrldetect`
- `rx_patterndetect`
- `rx_errdetect`
- `rx_disperr`
- `rx_rmfifodatainserted`
- `rx_rmfifodatadeleted`
- `rx_runningdisp`
- `altpcs_carrierdetect`.

The `rx_rmfifodatainserted`, `rx_rmfifodatadeleted`, and `rx_runningdisp` signals are available as ports on the ALT2GXB megafunction in GIGE Enhanced mode. If you instantiate the `carrier_detect` logic block in your design as described in "UNH-IOL Carrier Detect" on page 1–21, `altpcs_carrierdetect` is available as a port on the `carrier_detect` logic block.

UNH-IOL Carrier Detect

The IEEE802.3 specification requires the receiver PCS state machine to check for carrier event, as shown in Figure 20.

Figure 20. IEEE 802.3 Receiver PCS State Machine CARRIER_DETECT *(Note 1)*



Note to Figure 20:

(1) Image taken from IEEE Std. 802.3™ -2002, IEEE Standard for Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications.

The specification defines `carrier_detect` function as follows:

The `carrier_detect` function detects carrier when either:

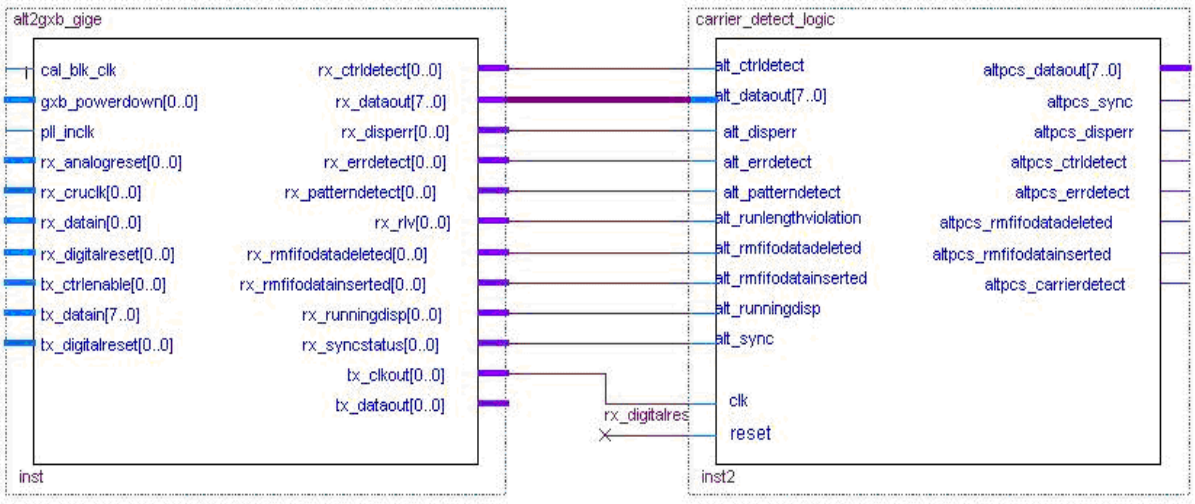
a) A two or more bit difference between [/x/] and both /K28.5/ encodings exists; or

b) A two to nine bit difference between [/x/] and the expected /K28.5/ (based on current running disparity) exists.

The Stratix II GX or Arria GX receiver configured in GIGE Enhanced mode generates ports required to implement the carrier detect function. The carrier detect function is implemented in the FPGA fabric outside the Altera transceiver.

Figure 21 shows the block diagram of the ALT2GXB megafunction in GIGE Enhanced mode and the carrier detect logic.

Figure 21. GIGE Enhanced Mode with Carrier Detect Block



To implement carrier detect in your design, unzip the file **carrier_detect_logic.v** or **carrier_detect_logic.vhd** to your Quartus II installation directory. Instantiate the verilog/vhdl block in your design and connect the ports, as shown in Figure 21

The `carrier_detect_logic` module infers a `carrier_detect` event from the 8-bit decoded data on the `rx_dataout[7:0]` port and receiver status signals:

- The `carrier_detect` status is driven on the `altpcs_carrierdetect` port and has a two-clock cycle latency with respect to the input data and control signals. Output data and status signals (`altpcs_*`) are registered twice to match the two-clock cycle latency of the `altpcs_carrierdetect` signal.
- The `carrier_detect_logic` block for each channel utilizes 17 adaptive logic modules (ALMs) or approximately 43 equivalent logic elements (LEs).

Conclusion

GIGE designs using Altera's TSE MegaCore on Altera's Stratix II GX or Arria GX devices are UNH-IOL test suite compliant. You can modify GIGE designs that do not use Altera's TSE MegaCore and require UNH-IOL test suite compliance as explained in this document to achieve UNH-IOL test suite compliance.

Referenced Documents

This application note references the following documents:

- *Arria GX ALT2GXB Megafunction User Guide* in volume 2 of the Arria GX Device Handbook
- *Arria GX Transceiver Architecture* in volume 2 of the *Arria GX Device Handbook*
- *Arria GX Transceiver Protocol Support and Additional Features* in volume 2 of the Arria GX Device Handbook
- *Stratix II GX ALT2GXB Megafunction User Guide* in volume 2 of the Stratix II GX Device Handbook
- *Stratix II GX Transceiver Architecture Overview* in volume 2 of the Stratix II GX Device Handbook
- *Triple Speed Ethernet MegaCore Function User Guide*

Revision History

Table 7 shows the revision history for this application note.

Table 7. Document Revision History

Date and Version Number	Changes Made	Summary of Changes
September 2008, v1.0	Initial release.	—