10 Gigabit Ethernet Consortium

10GBASE-X PCS Test Suite version 1.3b

UNH-IOL — 121 Technology Drive, Suite 2 — Durham, NH 03824 — +1-603-862-0090
Consortium Manager: Bob Noseworthy — ren@iol.unh.edu — +1-603-862-4342

Vendor X
Company Name
Street Address
City, State, Zip

January 12, 2005
Report Rev. 1.0

Enclosed are the results from the Clause 48 10GBASE-X PCS testing performed on:

Device Under Test (DUT):	10 Gigabit Ethernet Switch
Hardware Version:	N/A
Firmware Version:	N/A
Software Version:	N/A
Miscellaneous:	Tested on port 1

The test suite referenced in this report is available at the UNH-IOL website:

ftp://ftp.iol.unh.edu/pub/10gec/Clause_48_PCS_Test_Suite_V1.3b.pdf

Issues Observed While Testing
48.1.5: good_cgs code count – The DUT was observed to improperly not receive the second frame upon reception of 4 /R/ characters in between Invalid code-groups.
48.2.4: Deskew error identification – The DUT was not observed to recognize RD errors in ||A|| columns as deskew errors.

For specific details regarding issues please see the corresponding test result.

Testing Completed 12/27/2004

John Q. Tester

John Q. Tester
johnqtester@iol.unh.edu
+1603-862-0205

Review Completed 12/27/2004

John Q. Reviewer

John Q. Reviewer
johnqreviewer@iol.unh.edu
+1-603-862-4342

Digital Signature Information

This document was created using an Adobe digital signature. A digital signature helps to ensure the authenticity of the document, but only in this digital format. For information on how to verify this document's integrity proceed to the following site:

http://www.iol.unh.edu/certifyDoc

If the document status still indicates "Validity of author NOT confirmed", then please contact the UNH-IOL to confirm the document's authenticity. To further validate the certificate integrity, Adobe 6.0 should report the following fingerprint information:

MD5 Fingerprint: DB27 087D 94C8 CB63 7679 50E1 2239 C564
SHA-1 Fingerprint: 5411 C271 9458 ECB2 F401 E0C9 0026 25C3 98D3 E8FE

Result Key

The following table contains possible results and their meanings:

Result	Interpretation
PASS	The Device Under Test (DUT) was observed to exhibit conformant behavior.
PASS with Comments	The DUT was observed to exhibit conformant behavior however an additional explanation of the situation is included, such as due to time limitations only a portion of the testing was performed.
FAIL	The DUT was observed to exhibit non-conformant behavior.
Warning	The DUT was observed to exhibit behavior that is not recommended.
Informative	Results are for informative purposes only and are not judged on a pass or fail basis.
Refer to Comments	From the observations, a valid pass or fail could not be determined. An additional explanation of the situation is included.
Not Applicable	The DUT does not support the technology required to perform these tests.
Not Available	Due to testing station or time limitations, the tests could not be performed.
Borderline	The observed values of the specified parameters are valid at one extreme, and invalid at the other.
Not Tested	Not tested due to the time constraints of the test period.

SAMPLE REPORT

Test Setup

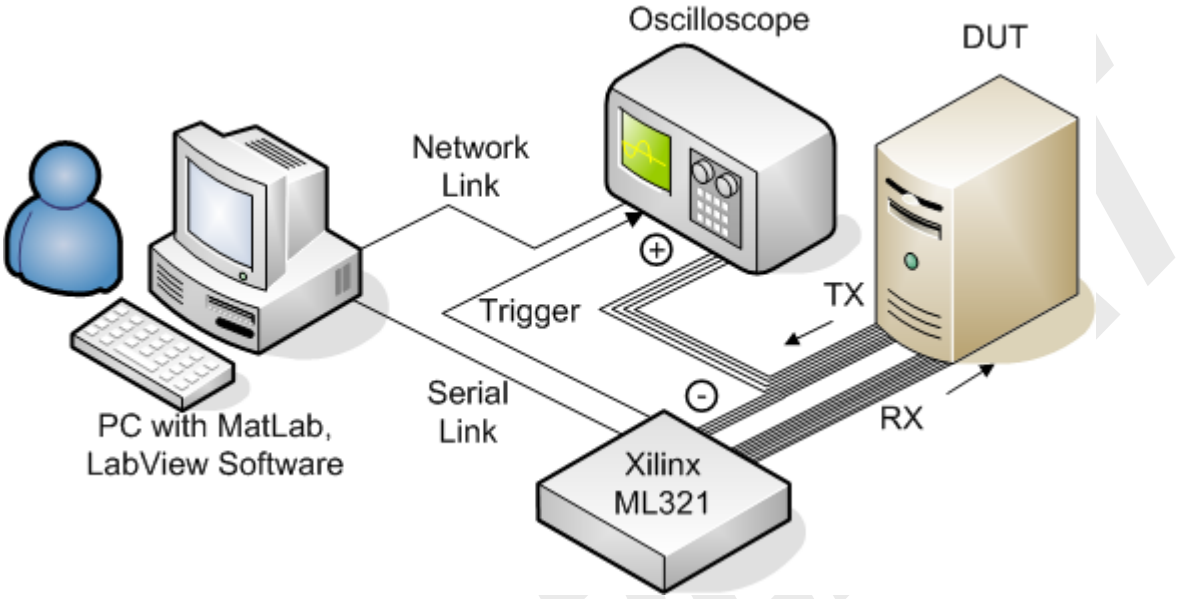


Figure 1 - Test Configuration

Figure 1 above depicts the test setup employed throughout the testing process. Port 1 of the Device Under Test (DUT) was used to provide access to the DUT in all test cases. Control access to the DUT was provided via a serial port interface. The test system consists of the PC, Lecroy SDA 6000 (DSO), and Xilinx ML321.

- XAUI signaling is generated by the "Xilinx ML321" Testing Station. This system continuously sends valid (though not truly randomized) XAUI Idle when not instructed to send a programmable transmit pattern. When the transmit pattern is sent, a trigger signal is sent to the DSO to capture the response of the DUT.
- Labview software controls the generation of the test vectors and programming of the ML321. Labview and Matlab software control the downloading and analysis of the signaling captured on the Lecroy SDA 6000.
- The 4-channel XAUI signaling from the DUT is captured single-endedly on a Lecroy SDA 6000. The positive signals (Tx+) are driven into the 50ohm terminations of the DSO, while the negative signals (Tx-) are driven into negative receive signals of the Xilinx ML321, with 50ohm terminations on the positive receive signals of the Xilinx ML321.
- A PC is used for four purposes:
  - via a serial port to control the DUT
  - via the network to control the DSO and download and process the waveforms in Matlab
  - via the USB/JTAG interface to download the firmware for the Xilinx ML321
  - via a Serial port to control transmissions from the Xilinx ML321

Section1: Detailed Table of Results
Group 1: Synchronization

Test # and Label	Part(s)	Result(s)
48.1.1: COMMA code count	**a, b**	**PASS**
Comments on Test Procedure		
Purpose: To verify how many /COMMA/ codes the device under test (DUT) needs to receive before it can acquire synchronization. When in the LOSS_OF_SYNC state, the PCS attempts to realign its current code-group boundary to the boundary defined by a comma. This process is called code-group alignment. In order for the DUT to acquire synchronization, all four lanes must individually acquire synchronization. As long as the DUT has not acquired synchronization, it is not capable of receiving frames. Configure the testing station to transmit various idle patterns to the DUT, followed by valid frames and observe whether or not the frames are received by the DUT. a. The DUT should receive the frame when *comma_count* is 4 or higher while receiving commas in K28.5. b. The DUT should receive the frame when *comma_count* is 4 or higher while receiving commas in K28.1 or K28.7. All K28.7 codes were followed by D0.0 codes to avoid misaligned commas.		
Comments on Test Results		
a. The DUT was observed to properly receive the frame after receiving 4 /K28.5/ code-groups on each lane. b. The DUT was observed to properly receive the frame after receiving 4 /K28.1/ or /K28.7/ code-groups on each lane.		

SAMPLE REPORT

Test # and Label	Part(s)	Result(s)
48.1.2: Reception of INVALID codes	**a, b**	**PASS**

Comments on Test Procedure

Purpose: To verify that the DUT properly reacts to the reception of various invalid code-groups during the synchronization process.

The synchronization process determines whether the underlying receive channel is ready for operation. In order for the DUT to acquire synchronization, all four lanes must individually acquire synchronization. During the synchronization process, if the DUT receives invalid code-groups, then it will return to the LOSS_OF_SYNC state.

Column of Invalid characters attempted in this test:	
1111111111	0000000000
code with RD error (||K||)	Idle code with single bit error (||K||)

a. This part verifies that the DUT reacts properly to invalid codes when in COMMA_DETECT. To achieve this, the DUT is sent a valid looping sequence containing ||R|| and ||A|| properly spaced to keep the DUT in the LOSS_OF_SYNC state. Once the test begins, the DUT is sent one valid ||K||, one of the invalid codes in the table above, and 2 to 4 additional valid ||K|| columns. The DUT should acquire synchronization after receiving four consecutive valid commas.
b. The part verifies that the DUT reacts properly to invalid codes when in SYNC_ACQUIRED. The DUT should properly lose synchronization on 4 of each invalid code-group and then regain synchronization and receive the frames when *comma_count* ||K|| have been received.

Comments on Test Results

a. The DUT properly acquired synchronization if 4 valid ||K|| columns followed the invalid code for all invalid codes in the above table.
b. The DUT properly lost synchronization upon reception of 4 invalid codes for each of the invalid codes in the above table.

SAMPLE REPORT

Test # and Label	Part(s)	Result(s)
48.1.3: Acquire Synchronization	**a - f**	**PASS**

Comments on Test Procedure

Purpose: To verify that the device under test (DUT) acquires synchronization upon the reception of four columns of four identical Idle Sync code-groups corresponding to the Idle Sync function.

When in the LOSS_OF_SYNC state, the PCS attempts to realign its current code-group boundary to the boundary defined by a comma. This process is called code-group alignment. In order for the DUT to acquire synchronization, all four lanes must individually acquire synchronization. As long as the DUT has not acquired synchronization, it is not capable of receiving frames. Configure the testing station to transmit various idle patterns to the DUT, followed by valid frames and observe whether or not the frames are received by the DUT.

a. *Transition through COMMA_DETECT_1 with valid codes*. – The DUT is sent a sequence containing /K/R/ followed by additional /K/ codes. The DUT should receive the frame after receiving a total of *comma_count* /K/ on each lane.
b. *Transition through COMMA_DETECT_1 with invalid codes*. – The DUT is sent a sequence containing /K/invalid/ followed by additional /K/ codes. The DUT should receive the frame only after each lane receives *comma_count* consecutive /K/ code-groups without any intervening invalid code-groups.
c. *Transition through COMMA_DETECT_2 with valid codes.* – The DUT is sent a sequence containing /K/K/R/ followed by additional /K/ codes. The DUT should receive the frame after receiving a total of *comma_count* /K/ on each lane.
d. *Transition through COMMA_DETECT_2 with invalid codes* – The DUT is sent a sequence containing /K/K/invalid/ followed by additional /K/ codes. The DUT should receive the frame only after each lane receives *comma_count* consecutive /K/ code-groups without any intervening invalid code-groups.
e. *Transition through COMMA_DETECT_3 with valid codes.* – The DUT is sent a sequence containing /K/K/K/R/ followed by additional /K/ codes. The DUT should receive the frame after receiving a total of *comma_count* /K/ on each lane.
f. *Transition through COMMA_DETECT_3 with invalid codes.* – The DUT is sent a sequence containing /K/K/K/invalid/ followed by additional /K/ codes. The DUT should receive the frame after each lane receives *comma_count* consecutive /K/ code-groups without any intervening invalid code-groups.

Note: Invalid codes of 0000000000 were utilized for this test. Each part is comprised of 5 test vectors, with the test sequence on each lane independently and once with the test sequence on all lanes.

Comments on Test Results

a. With sequence A the DUT properly required a total of 4 ||K|| codes.
b. With sequence B the DUT properly required 4 ||K|| codes after the invalid code.
c. With sequence C the DUT properly required a total of 4 ||K|| codes.
d. With sequence D the DUT properly required 4 ||K|| codes after the invalid code.
e. With sequence E the DUT properly required a total of 4 ||K|| codes.
f. With sequence F the DUT properly required 4 ||K|| codes after the invalid code.

SAMPLE REPORT

Test # and Label	Part(s)	Result(s)
48.1.4: INVALID code count	**a**	**PASS**

Comments on Test Procedure

Purpose: To verify the number of consecutive /Invalid/ codes that will cause the DUT to lose synchronization once it has arrived in the SYNC_ACQUIRED_1 state.

After acquiring synchronization, the DUT tests received code-groups and uses multiple sub-states, effecting hysteresis, to move between the SYNC_ACQUIRED_1 and LOSS_OF_SYNC states. After receiving multiple invalid code-groups, the DUT may be forced into the LOSS_OF_SYNC state and will be unable to properly respond to frames while in this state. Configure the testing station to bring the DUT through these states by transmitting various idle patterns followed by valid frames, and observe whether or not the frames are received by the DUT.

a. The testing station transmitted 3 frames, with the second frame preceded by *invalid_count* invalid codes. The DUT should discard the second frame in the sequence when *invalid_count* is set to the value of 4 or higher.

Note: Invalid codes of /0000000000/ were utilized for this test.

Comments on Test Results

a. The DUT was observed to properly discard the middle frame upon reception of 4 Invalid codes.

Test # and Label	Part(s)	Result(s)
48.1.5: good_cgs code count	**a**	**FAIL**

Comments on Test Procedure

Purpose: To determine the number of valid code groups that is necessary to move the DUT from one of the SYNC_ACQUIRED_xA states to the SYNC_ACQUIRED_(x-1) states.

After acquiring synchronization, the DUT tests received code-groups and uses multiple sub-states, effecting hysteresis, to move between the SYNC_ACQUIRED_1 and LOSS_OF_SYNC states. After receiving multiple invalid code-groups, the DUT may be forced into the LOSS_OF_SYNC state and will be unable to properly respond to frames while in this state. Configure the testing station to bring the DUT through these states by transmitting various idle patterns followed by valid frames, and observe whether or not the frames are received by the DUT. Note *invalid_count* is determined in test 48.1.4.

a. The testing station transmitted 3 frames, with the second frame preceded by *n* invalid codes, *good_cgs_count* /R/ codes, and *invalid_count-n* invalid codes (with *n* varying from 1 to 3). The DUT should receive the middle frame in the sequence when *good_cgs_count* is set to the value of 4 or higher.

Note: Invalid codes of /0000000000/ were utilized for this test.

Comments on Test Results

a. The DUT was observed to improperly not receive the second frame upon reception of 4 /R/ characters in between Invalid code-groups. The DUT was observed to require 5 /R/ characters in between Invalid code-groups before it would accept the second frame.

SAMPLE REPORT

Test # and Label	Part(s)	Result(s)
48.1.6: Loss of Synchronization	**a - d**	**PASS**

Comments on Test Procedure

Purpose: To verify that a DUT will lose synchronization after the reception of code-group sequences that should cause it to return to the LOSS_OF_SYNC state.

After acquiring synchronization, the DUT tests received code-groups and uses multiple sub-states, effecting hysteresis, to move between the SYNC_ACQUIRED_1 and LOSS_OF_SYNC states. After receiving multiple invalid code-groups, the DUT may be forced into the LOSS_OF_SYNC state and will be unable to properly respond to frames while in this state. Configure the testing station to bring the DUT through these states by transmitting various idle patterns followed by valid frames, and observe whether or not the frames are received by the DUT.

Test case	Description
A1	1 invalid code / ‘n-1’ valid codes / ‘m-1’ invalid codes / 1 valid code
A2	1 invalid code / ‘n’ valid codes / ‘m’ invalid codes / 1 valid code
B1	2 invalid codes / ‘n-1’ valid codes / ‘m-2’ invalid codes / 1 valid code
B2	2 invalid codes / ‘n’ valid codes / ‘m-1’ invalid codes / 1 valid code
C1	3 invalid codes / ‘n-1’ valid codes / ‘m-3’ invalid codes / 1 valid code
C2	3 invalid codes / ‘n’ valid codes / ‘m-2’ invalid codes / 1 valid code
D	4 repetitions of: 1 invalid code / 1 valid code

‘n’ = *good_cgs_count* found in test 48.1.5, ‘m’ = *invalid_count* found in test 48.1.4

The testing station transmitted 3 frames, with the second frame preceded by the sequences described in the test cases in the above table.

In all cases, the DUT should not receive the second frame.

Note: Invalid codes of /0000000000/ and valid codes of /R/ were utilized for this test. Frame 1 and the test cases were separated by minimum IPG, as well as frames 2 and 3. Frame 2 was transmitted immediately after the last code in the test case. All test cases were run 5 times, with the invalid codes in each of the 4 lanes individually and in all lanes together.

Comments on Test Results

a. The DUT was observed to properly discard the second frame in the sequence.
b. The DUT was observed to properly discard the second frame in the sequence.
c. The DUT was observed to properly discard the second frame in the sequence.
d. The DUT was observed to properly discard the second frame in the sequence.

SAMPLE REPORT

Test # and Label	Part(s)	Result(s)
48.1.7: Maintain Synchronization	**a - e**	**PASS**

Comments on Test Procedure

Purpose: To verify that the DUT is able to maintain synchronization for a specific set of invalid code-group sequences.

After acquiring synchronization, the DUT tests received code-groups and uses multiple sub-states, effecting hysteresis, to move between the SYNC_ACQUIRED_1 and LOSS_OF_SYNC states. After receiving multiple invalid code-groups, the DUT may be forced into the LOSS_OF_SYNC state and will be unable to properly respond to frames while in this state. However, while going through the hysteresis, the DUT will be in one of the SYNC_ACQUIRED_X or SYNC_ACQUIRED_XA states. While in one of these states, the DUT should maintain synchronization and be able to reply to frames. Configure the testing station to bring the DUT through these states by transmitting various idle patterns followed by valid frames, and observe whether or not the frames are received by the DUT.

Test case	Description
A1	1 invalid code / 1 valid code
A2	1 invalid code / ‘n-1’ valid codes / ‘m-2’ invalid codes / 1 valid code
A3	1 invalid code / ‘n’ valid codes / ‘m-1’ invalid codes / 1 valid code
B1	2 invalid codes / 1 valid code
B2	2 invalid codes / ‘n-1’ valid codes / ‘m-3’ invalid codes / 1 valid code
B3	2 invalid codes / ‘n’ valid codes / ‘m-2’ invalid codes / 1 valid code
C1	3 invalid codes / 1 valid code
C2	3 invalid codes / ‘n’ valid codes / ‘m-3’ invalid codes / 1 valid code
D1	4 repetitions (4x) of: 1 invalid code / 1 valid code
D2	2x(1 invalid code / ‘n-1’ valid code) / 1 invalid code / ‘n’ valid codes / 1 invalid code / valid code
E	Invalid in lane 0 / invalid in lane 1 / invalid in lane 2 / invalid in lane 3 / 1 valid code

‘n’ = *good_cgs_count* found in test 48.1.5, ‘m’ = *invalid_count* found in test 48.1.4

The testing station transmitted 3 frames, with the second frame preceded by the sequences described in the test cases in the above table.

In all cases, the DUT should receive the second frame.

Note: Invalid codes of /0000000000/ and valid codes of /R/ were utilized for this test. Frame 1 and the test cases were separated by minimum IPG, as well as frames 2 and 3. Frame 2 was transmitted immediately after the last code in the test case. Parts a – d were run 5 times, with the invalid codes in each of the 4 lanes individually and in all lanes together.

Comments on Test Results

a. The DUT was observed to properly maintain synchronization and accept all three frames.
b. The DUT was observed to properly maintain synchronization and accept all three frames.
c. The DUT was observed to properly maintain synchronization and accept all three frames.
d. The DUT was observed to properly maintain synchronization and accept all three frames.
e. The DUT was observed to properly maintain synchronization and accept all three frames.

SAMPLE REPORT

Test # and Label	Part(s)	Result(s)
48.1.8: Reaquire synchronization	**a**	**PASS**

Comments on Test Procedure

Purpose: To verify that the DUT properly reacquires synchronization once it has established a valid link.

After acquiring synchronization, the DUT tests received code-groups and uses multiple sub-states, effecting hysteresis, to move between the SYNC_ACQUIRED_1 and LOSS_OF_SYNC states. After receiving multiple invalid code-groups, the DUT may be forced into the LOSS_OF_SYNC state and will be unable to properly respond to frames while in this state. However, while going through the hysteresis, the DUT will be in one of the SYNC_ACQUIRED_X or SYNC_ACQUIRED_XA states. While in one of these states, the DUT should maintain synchronization and be able to reply to frames. Configure the testing station to bring the DUT through these states by transmitting various idle patterns followed by valid frames, and observe whether or not the frames are received by the DUT.

a. A misaligned stream was generated by inserting 5 bits into a valid stream. The DUT should properly lose synchronization on the misaligned stream and then acquire synchronization on this new comma alignment.

Comments on Test Results

a. The DUT was observed to properly lose synchronization and then reaquire synchronization to the new data stream.

SAMPLE REPORT

Group 2: Alignment

Test # and Label	Part(s)	Result(s)
48.2.1: Determination of align_count	**a**	**PASS**

Comments on Test Procedure

Purpose: To determine the number of ||A|| the DUT needs to receive before it will acquire alignment.

The DUT is brought to the SYNC_ACQUIRED_1 state and sent a frame followed by various idle patterns containing a fixed number of properly spaced ||A|| immediately followed by a frame, and observing whether or not the second frame is received by the DUT. No other ||A|| is sent to the DUT than those preceding the second frame.

a. The DUT should receive the frames when the value of *align_count* is 4.

Comments on Test Results

a. The DUT was observed to receive frames upon reception of 4 ||A|| columns.

Test # and Label	Part(s)	Result(s)
48.2.2: Acquire Alignment	**a, b, c**	**PASS**

Comments on Test Procedure

Purpose: To determine that a DUT will acquire alignment after the reception of 4 identical Idle Align code-groups corresponding to the Idle Align function.

The DUT is brought to the SYNC_ACQUIRED_1 state and sent a frame followed by various idle patterns containing deskew errors immediately followed by a frame, and observing whether or not the second frame is received by the DUT.

Test case	Description
A	1 ||A|| column, 1 deskew error, and then enough ||A|| columns until the DUT acquires alignment.
B	2 ||A|| column, 1 deskew error, and then enough ||A|| columns until the DUT acquires alignment.
C	3 ||A|| column, 1 deskew error, and then enough ||A|| columns until the DUT acquires alignment.

The DUT is sent the test cases above with sufficient valid idle (/K/ /R/) between each ||A|| or deskew error.

In all cases, the DUT should receive the frames after receiving 4 consecutive aligned ||A|| columns.

Note: Deskew errors were created by replacing one /A/ character from a valid ||A|| column with /R/.

Comments on Test Results

a. The DUT was observed to require 4 ||A|| columns after the deskew error to acquire alignment.
b. The DUT was observed to require 4 ||A|| columns after the deskew error to acquire alignment.
c. The DUT was observed to require 4 ||A|| columns after the deskew error to acquire alignment.

SAMPLE REPORT

Test # and Label	Part(s)	Result(s)
48.2.3: Loss of alignment count	**a**	**PASS**

Comments on Test Procedure

Purpose: To determine the number of misaligned ||A|| code-group sequences that should cause the DUT to return to the LOSS_OF_ALIGNMENT state.

The DUT is brought to the ALIGN_ACQUIRED_1 state and sent a frame followed by various idle patterns containing deskew errors immediately followed by a frame, and observing whether or not the second frame is received by the DUT.

a. The DUT was sent an idle pattern containing enough deskew errors to cause the DUT to lose alignment. The value of *lose_align_count* should be 4.

Note: Deskew errors were created by replacing one /A/ character from a valid ||A|| column with /R/.

Comments on Test Results

a. The DUT was observed to properly lose alignment after receiving 4 deskew errors.

Test # and Label	Part(s)	Result(s)
48.2.4: Deskew error identification	**a**	**FAIL**

Comments on Test Procedure

Purpose: To determine what the DUT considers to be a deskew error.

The DUT is brought to the ALIGN_ACQUIRED_1 state and sent a frame followed by various idle patterns containing deskew errors immediately followed by a frame, and observing whether or not the second frame is received by the DUT.

a. The DUT was sent several Idle patterns, each containing identical *lose_align_count* deskew errors. The list of deskew errors attempted are listed below. The DUT should lose alignment upon reception of *lose_align_count* worth of each deskew error without intervening valid ||A||.

/A/A/R/R/	/R/A/A/R/	/A/R/A/A/	/R/R/A/A/	/A/R/R/A/	/A/R/A/R/	/R/A/R/A/
/A/A/A/R	R/A/A/A/	/A/A/R/A/	/A/A/A/*/	/A/A/*/A/	/A/*/A/A/	/*/A/A/A/
/A/A/A/#/	/A/A/#/A/	/A/#/A/A/	/#/A/A/A/			

Notes: * represents any code that is one bit different than /A/. # represents an otherwise valid /A/ that has the incorrect running disparity.

Comments on Test Results

a. The DUT was observed to not recognize RD errors in ||A|| columns as deskew errors.

SAMPLE REPORT

Test # and Label	Part(s)	Result(s)
48.2.5: Loss of Alignment	**a - d**	**PASS**

Comments on Test Procedure

Purpose: To verify that a DUT will lose alignment after the reception of code-group sequences that should cause it to return to the LOSS_OF_ALIGNMENT state.

The DUT is brought to the ALIGN_ACQUIRED_1 state and sent a frame followed by various idle patterns containing deskew errors immediately followed by a frame, and observing whether or not the second frame is received by the DUT.

Pattern 1	Pattern 2	Pattern 3	Pattern 4
Frame 1 Valid Idle – min IPG 'lac' deskew errors Frame 2 Valid Idle – min IPG Frame 3	Frame 1 Valid Idle – min IPG 1 deskew error 1 ||A|| 'lac' deskew errors Frame 2 Valid Idle – min IPG Frame 3	Frame 1 Valid Idle – min IPG 2 deskew errors 1 ||A|| 'lac-1' deskew errors Frame 2 Valid Idle – min IPG Frame 3	Frame 1 Valid Idle – min IPG 3 deskew error 1 ||A|| 'lac-2' deskew errors Frame 2 Valid Idle – min IPG Frame 3

'lac' = lose_align_count found from test 48.2.3

a. The testing station transmitted the patterns from the above table. The deskew error used was A/R/R/R.
b. The testing station transmitted the patterns from the above table. The deskew error used was R/A/R/R.
c. The testing station transmitted the patterns from the above table. The deskew error used was R/R/A/R.
d. The testing station transmitted the patterns from the above table. The deskew error used was R/R/R/A.

In all cases, the DUT should not accept the middle frame.

Comments on Test Results

a. The DUT properly did not accept the middle frame.
b. The DUT properly did not accept the middle frame.
c. The DUT properly did not accept the middle frame.
d. The DUT properly did not accept the middle frame.

SAMPLE REPORT

Test # and Label	Part(s)	Result(s)
48.2.6: Maintain Alignment	**a**	**PASS**

Comments on Test Procedure

Purpose: To verify that the DUT is able to maintain alignment for receiving a specific set of deskew errors.

After acquiring alignment, the DUT tests received ordered_sets and uses multiple sub-states, effecting hysteresis, to move between the ALIGN_ACQUIRED_1 and LOSS_OF_ALIGNMENT states. After receiving multiple deskew errors, the DUT may be forced into the LOSS_OF_ALIGNMENT state and will be unable to properly respond to frames while in this state. However, while going through the hysteresis, the DUT will be in one of the ALIGN_ACQUIRED_X. While in one of these states, the DUT should maintain alignment and be able to reply to frames. Configure the testing station to bring the DUT through these states by transmitting various idle patterns followed by valid frames, and observe whether or not the frames are received by the DUT.

Pattern 1	Pattern 2	Pattern 3
Frame 1 Valid Idle – min IPG 1 deskew error Frame 2 Valid Idle – min IPG Frame 3	Frame 1 Valid Idle – min IPG 2 deskew errors Frame 2 Valid Idle – min IPG Frame 3	Frame 1 Valid Idle – min IPG 3 deskew errors Frame 2 Valid Idle – min IPG Frame 3
Pattern 4	**Pattern 5**	**Pattern 6**
Frame 1 Valid Idle – min IPG 1 deskew error 1 ||A|| 'lac-1' deskew errors Frame 2 Valid Idle – min IPG Frame 3	Frame 1 Valid Idle – min IPG 2 deskew errors 1 ||A|| 'lac-2' deskew errors Frame 2 Valid Idle – min IPG Frame 3	Frame 1 Valid Idle – min IPG 3 deskew errors 1 ||A|| 'lac-3' deskew errors Frame 2 Valid Idle – min IPG Frame 3

a. The testing station transmitted the patterns from the above table. The deskew error used was A/R/R/R.
b. The testing station transmitted the patterns from the above table. The deskew error used was R/A/R/R.
c. The testing station transmitted the patterns from the above table. The deskew error used was R/R/A/R.
d. The testing station transmitted the patterns from the above table. The deskew error used was R/R/R/A.

In all cases, the DUT should accept the middle frame.

Comments on Test Results

a. The DUT was observed to properly maintain alignment and received all frames.
b. The DUT was observed to properly maintain alignment and received all frames.
c. The DUT was observed to properly maintain alignment and received all frames.
d. The DUT was observed to properly maintain alignment and received all frames.

SAMPLE REPORT

Test # and Label	Part(s)	Result(s)
48.2.7: Reacquire alignment	**a**	**PASS**

Comments on Test Procedure

Purpose: To verify that the DUT is able to reacquire alignment upon reception of a skewed Idle stream.

Once the DUT has achieved alignment, it is sent a set of data that has additional codewords inserted into some but not all lanes, creating a sudden lane-to-lane skew. This should cause the DUT to lose alignment, and then regain alignment on the new lane-to-lane skew value even though synchronization is not lost.

a. The DUT should lose alignment on the skewed idle stream and then regain alignment. The Idle stream was skewed by adding 10 bits to one lane.

Comments on Test Results

a. The DUT was observed to properly lose alignment and then regain alignment on the new bit stream.

Test # and Label	Part(s)	Result(s)
48.2.8: Skew Tolerance	**a**	**PASS**

Comments on Test Procedure

Purpose: To verify that the DUT is able to properly achieve alignment and receive frames when skew is introduced to the XAUI link.

Across all lanes, the DUT should be tolerant of at least 41 UI total skew on its receiver.

The test station was configured to send idle with ||A|| columns separated by a constant 16 columns of ||K|| and ||R||.
For each test case, the test stations transmitter's would be powered down, the output skew would be set on one lane at a time (either + or – relative to the other lanes), and then the test stations transmitter's would be powered up again. Frame transmission would then be attempted from the test station. Link status and frame transmission was observed.
For each skew setting, several trials were conducted consisting of powering down the testing station's transmitters, powering up the transmitters, sending a frame, and repeating.

Comments on Test Results

The DUT was observed to establish link and receive, loopback and transmit frames with following skew applied (in Unit Intervals).

Lane 0 +62 UI to –57 UI
Lane 1 +59 UI to –64 UI
Lane 2 +59 UI to –57 UI
Lane 3 +62 UI to –61 UI

Note, with the transmission of ||A|| columns separated by a constant 16 columns of ||K|| and ||R||, it would be possible to properly deskew and receive up to 85bits of skew. The standard requires a tolerance of at least 41 UI of skew between the PMA Tx and PMA Rx, however the PMA Rx alone is only required to tolerate 20 UI of skew. The values indicated above correspond to skew generated at the ML321 test station. Lane to lane skew inherent in the physical channel from the Xilinx Virtex II Pro to the backplane is suspected to be 3 UI or less and therefore assumed negligible.

SAMPLE REPORT

Group 3: Transmit Related

Test # and Label	Part(s)	Result(s)
48.3.1: 8B/10B Encoding	**a, b**	**PASS**

Comments on Test Procedure

Purpose: To verify that the device under test (DUT) selects the proper encoding for transmitted code-groups.

Configure the device under test to transmit all valid data code-groups and all valid special code-groups, /K/, /R/, /A/, /S/, /T/, /K28.1/, /K28.6/, and /K23.7/ for both the positive and negative running disparity. Both forms of each valid data code-group can be generated as part of normal packet data.

a. The DUT should be able to properly transmit all valid data and control code-groups for both positive and negative running disparity.
b. The DUT should be able to properly transmit all valid special code-groups for both positive and negative running disparity.

Comments on Test Results

a. The DUT was observed to properly transmit all valid data and control code-groups for both positive and negative running disparity.
b. The DUT was observed to properly transmit all valid special code-groups for both positive and negative running disparity.

SAMPLE REPORT

Test # and Label	Part(s)	Result(s)
48.3.2: Idle Sequencing	**a – e**	**PASS**

Comments on Test Procedure

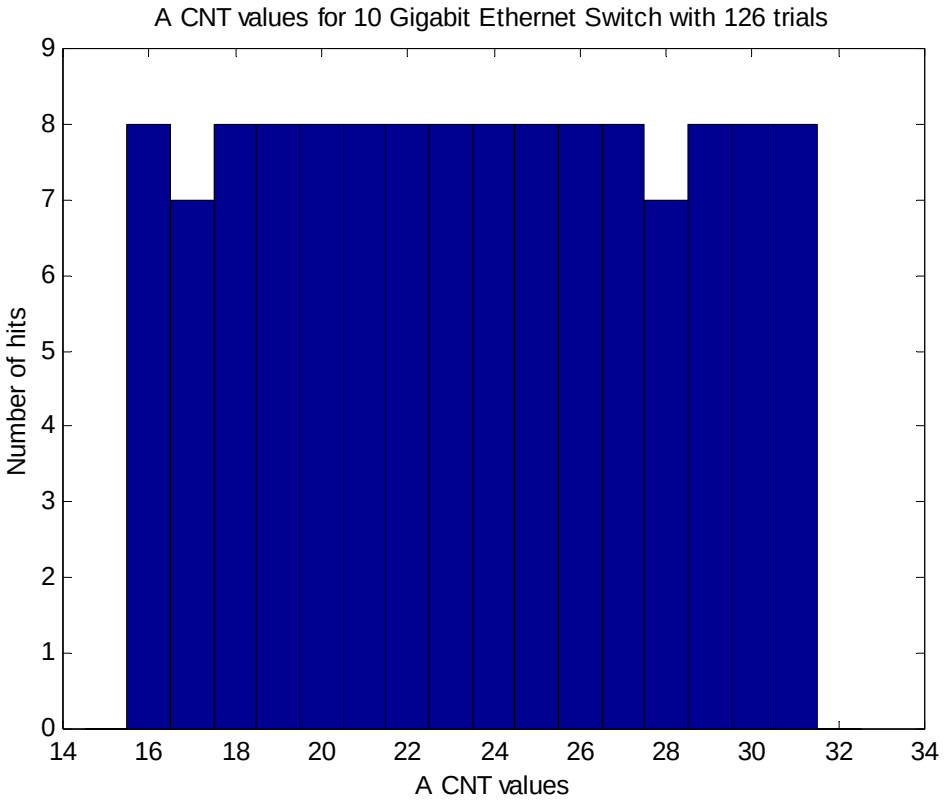
Purpose: To verify that the DUT follows the proper rules for the sequencing of Idle.

Configure the DUT to send combinations of frames and idle to verify that the rules for the sequencing of idle are properly followed:

a. The first ||I|| following ||T|| alternates between ||A|| or ||K|| (maintaining ||A|| spacing)
b. If less than A_CNT, and the previous frame followed ||T|| with ||K|| the first ||I|| following ||T|| should be another ||K||
c. ||R|| is always the second full column of idle after ||T||
d. ||A|| is sent after *r* non-||A|| columns, where *r* is uniformly distributed integer between 16 and 31, inclusive. This establishes minimum and maximum ||A|| spacing.
e. ||K|| and ||R|| are sent with a uniform distribution in the absence of ||A||

Comments on Test Results

a. When transmitting frames at least 32 columns in length, the DUT was observed to properly alternate between ||A|| and ||K|| at the end of the frame.
b. When transmitting 64 byte frames in length, the DUT was observed to properly alternate between ||A|| and ||K|| at the end of the frame. The DUT was observed to sometimes alternate as ||A||, ||K||, and ||K||.
c. The DUT was observed to always properly transmit ||R|| in the second column after ||T||.
d. The DUT was observed to properly transmit 16-31 columns of ||K|| and ||R|| between ||A|| columns.
e. The distribution of ||K|| and ||R|| follows the standard defined polynomial X^7+X^3+1.



Test # and Label	Part(s)	Result(s)
48.3.3: cvtx_terminate	**a**	**PASS**
Comments on Test Procedure		
Purpose: To verify that the DUT properly fills in the column containing the /T/ with /K/ code-groups. Configure the DUT to transmit a frame of length *n*, *n+1*, and *n+2*, where *n* is an integer multiple of 4. The DUT should fill the remaining lanes in ||T|| with /K/. a. The DUT should complete the column containing the /T/ with /K/ code-groups.		
Comments on Test Results		
a. The DUT was always observed to properly complete the column containing /T/ with /K/.		

Test # and Label	Part(s)	Result(s)
48.3.4: Fault Transmission	**a**	**PASS**
Comments on Test Procedure		
Purpose: To verify that the DUT properly transmits sequence ordered_sets. Configure the DUT such that the RS is transmitting continuous local or remote fault ordered_sets to the PCS. The PCS should only transmit ||Q|| ordered_sets following an ||A||. a. When continuous sequence ordered_sets are being transmitted, the DUT should only place ||Q|| ordered_sets after ||A|| ordered_sets.		
Comments on Test Results		
a. The DUT was observed to properly follow each column of ||A|| with a ||Q|| ordered set.		

SAMPLE REPORT

Group 4: Receive Related

Test # and Label	Part(s)	Result(s)
48.4.1: Fault Reception	**a - c**	**Not Applicable**
Comments on Test Procedure		
Purpose: To verify that the DUT properly receives sequence ordered_sets. Instruct the testing station to transmit local and remote faults to the DUT. The DUT should pass local and remote fault ordered_sets to the PCS client, and should generate local fault ordered_sets while in the LOCAL_FAULT_INDICATE state. a. While in the LOCAL_FAULT_INDICATE state, the DUT should pass local fault to the PCS client. b. When receiving local fault, the DUT should pass local fault sequence ordered_sets to the PCS client. c. When receiving remote fault, the DUT should pass remote fault sequence ordered_sets to the PCS client.		
Comments on Test Results		
This test could not be completed as there was not an interface available to observe the transmissions to the PCS client.		

SAMPLE REPORT

Test # and Label	Part(s)	Result(s)
48.4.2: check_end	**a, b**	**PASS**

Comments on Test Procedure

Purpose: To verify that the DUT properly implements the check_end function.

The frame termination test sequences sent to the DUT are as follows:

Interpretation Note: "/ TKKK / RD fmmm KKKK / " signifies two columns, the first with /T/ on lane 0, the second all /K/s where the Running Disparity (RD) is flipped (f) on lane 0 (and thus an RD error), and maintained (m) on lanes 1,2,3 (and thus not an error)

Number	Test Sequence	Valid Checkend?
1	/ TKKK / RDfmmm KKKK /	Yes
2	/ TKKK / RDmfmm KKKK /	Yes
3	/ TKKK / RDmmfm KKKK /	Yes
4	/ TKKK / RDmmmf KKKK /	Yes
5	/ DTKK / RDfmmm KKKK /	No
6	/ DTKK / RDmfmm KKKK /	Yes
7	/ DTKK / RDmmfm KKKK /	Yes
8	/ DTKK / RDmmmf KKKK /	Yes
9	/ DDTK / RDfmmm KKKK /	No
10	/ DDTK / RDmfmm KKKK /	No
11	/ DDTK / RDmmfm KKKK /	Yes
12	/ DDTK / RDmmmf KKKK /	Yes
13	/ DDDT / RDfmmm KKKK /	No
14	/ DDDT / RDmfmm KKKK /	No
15	/ DDDT / RDmmfm KKKK /	No
16	/ DDDT / RDmmmf KKKK /	Yes
17	/ TKKK / KKKR /	Yes
18	/ TKKK / KKRK /	Yes
19	/ TKKK / KRKK /	Yes
20	/ TKKK / RKKK /	Yes
21	/ DTKK / KKKR /	Yes

Number	Test Sequence	Valid Checkend?
22	/ DTKK / KKRK /	Yes
23	/ DTKK / KRKK /	Yes
24	/ DTKK / RKKK /	No
25	/ DDTK / KKKR /	Yes
26	/ DDTK / KKRK /	Yes
27	/ DDTK / KRKK /	No
28	/ DDTK / RKKK /	No
29	/ DDDT / KKKR /	Yes
30	/ DDDT / KKRK /	No
31	/ DDDT / KRKK /	No
32	/ DDDT / RKKK /	No
33	/ RDmfmm TKKK /	No
34	/ RDmmfm TKKK /	No
35	/ RDmmmf TKKK /	No
36	/ TKKR /	No
37	/ TKRK /	No
38	/ TRKK /	No

a. In Part A, the DUT should always receive the first and third frames. When the error occurs in a column that immediately follows a /D/ code-group, the second frame should always be discarded. When the error occurs in a column that immediately follows a /T/ or /I/ code-group, the second frame should not be discarded.
b. In Part B, the DUT should always receive the first and third frames but never receive the second frame.

Comments on Test Results

a. When sent 3 frames, where the 2nd frame has the termination indicated in the table above, the DUT properly discarded the second frame in the cases where the checkend result for the termination is invalid (indicated as "No" in the "Valid Checkend?" column in the table above).
b. The DUT was observed to properly reject frames containing running disparity errors and non-K codes in ||T||.

Test # and Label	Part(s)	Result(s)
48.4.3: Tolerance to ||A|| spacing	**a, c**	**PASS**
	b	**Not Available**

Comments on Test Procedure

Purpose: To determine if the DUT is insensitive to ||A|| Spacing.

The DUT is brought to the SYNC_ACQUIRED_1 state and sent a frame followed by validly encoded idle patterns with *align_count* or greater ||A|| with fixed spacing transmitted immediately before a frame, and observing whether or not the second frame is received by the DUT.

a. The minimum ||A|| spacing that allows the DUT to acquire alignment is determined.
b. The maximum ||A|| spacing that allows the DUT to acquire alignment is determined.
c. Determine if the DUT can maintain alignment without receiving any ||A||.

Comments on Test Results

a. The DUT was observed to acquire alignment with as little as 3 columns separating ||A|| columns.
b. This portion of the test cannot be completed with the current testing system, however the DUT was observed to acquire alignment with an ||A|| spacing of up to 230 columns.
c. The DUT was observed to maintain alignment for at least 5 minutes without receiving ||A|| columns.

SAMPLE REPORT