Gigabit Ethernet Consortium
Clause 36 PCS Conformance Test Suite v2.1 Report

UNH-IOL — 121 Technology Drive, Suite 2 — Durham, NH 03824 — +1-603-862-0090
Consortium Manager: Gerry Nadeau — grn@iol.unh.edu — +1-603-862-0166

Vendor X
Company Name
Street Address
City, State, Zip

February 8, 2006
Report Rev. 1.0

Enclosed are the results from the Clause 36 PCS Conformance testing performed on:

Device Under Test (DUT):	Gigabit Ethernet Switch
Hardware Version:	48 port swtich
Firmware Version:	N/A
Software Version:	N/A
Miscellaneous:	Tested on port 5, 10

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

ftp://ftp.iol.unh.edu/pub/ethernet/test_suites/CL36_PCS/CL36_PCS_Test_Suite_v2.1.pdf

Issues Observed While Testing
36.1.1 – Acquire Synchronization- part (b): The DUT was not observed to acquire synchronization from the reception of up to 10,000 repetitions of /K28.1/D0.0/.
36.1.1 – Acquire Synchronization – part (c): The DUT only required the reception of 2 evenly aligned commas and an /I/ ordered_set to acquire synchronization and receive a frame.
36.1.3 – Lose of Synchronization – part (b): The DUT was observed to maintain synchronization upon the reception of four test sequences that should have caused it to lose synchronization.
36.3.2 – Carrier Event Handling – part (b): The DUT was observed to discard all frames proceeded by test sequences that were 1 bit different than /K28.5/.
36.3.2 – Carrier Event Handling – part (c): The DUT failed to respond to frames preceded by a test sequence where /D5.6/, /D16.2/, /D21.5/ or /D2.2/ is replaced with /S/, /T/, /R/ or /V/.

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

Testing Completed 02/08/2006

John Q. Tester

John Q. Tester
johnqtester@iol.unh.edu

Digitally signed by UNH-IOL
Date: 2006.02.08 11:09:32 -05'00'

Review Completed 02/08/2006

John Q. Reviewer

John Q. Reviewer
johnqreviewer@iol.unh.edu

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: 8664 5701 3DC2 368A 0CC0 A1D7 792C D70C
SHA-1 Fingerprint: BF96 86A2 E723 9795 C8EA B9F8 1E10 BF22 1D61 3CE4

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 of 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.

GROUP 1: Synchronization

Test # and Label	Part(s)	Result(s)
36.1.1 – Acquire Synchronization	**a**	**FAIL**
	b	**FAIL**
	c	**FAIL**

Expected Results and Procedural Comments

Purpose: To verify that the device under test (DUT) acquires synchronization upon the reception of three ordered_sets each starting with a code-group containing a comma. Note, following the acquisition of synchronization, one additional /I/ ordered_set is required prior to frame reception.

A manually configured link is established between the DUT and the testing station.

a. The DUT is sent 200 consecutive invalid code-groups followed by each of the sequences listed below, then an ARP request. *Note: sequences 5 through 10 are followed by one /I/ ordered_set.*
   1) /I/I2/I2/I2/ (/I/=/I1/ if beginning RD is positive, otherwise /I/=/I2/)
   2) /I1/I1/I1/I1/
   3) /I2/I2/I2/I2/
   4) /I1/I2/I1/I2/
   5) 3 repetitions of /K28.5/D0.0/
   6) 3 repetitions of /K28.1/D0.0/
   7) 3 repetitions of /K28.5/D21.5/D0.0/D0.0/
   8) 3 repetitions of /K28.5/D2.2/D0.0/D0.0/
   9) 3 repetitions of /K28.5/D0.0/D0.0/D0.0/
   10) 3 repetitions of /K28.5/D0.0/D0.0/D0.0/D0.0/D0.0/

b. All sequences from part a, with which the DUT did not acquire synchronization, are retransmitted with additional repetitions. The DUT should eventually be able to acquire synchronization from all of the test sequences.

c. The Synchronization state diagram (Figure 36-9), beginning in the LOSS_OF_SYNC state, requires that three consecutive Idle ordered_sets be received in order for a device to acquire synchronization. Once sync_status=OK the PCS receive state diagram (figure 36-7a) transitions to the WAIT_FOR_K state. Only one additional Idle ordered_set must be received before a device should be able to receive a frame. According to figures 36-7a and 36-9 of the standard, a device should only require the reception of four consecutive Idle ordered_sets when synchronization is lost in order to receive a frame (three to acquire synchronization and one more in order for the receive state machine to receive a frame).

Comments on Test Results

a) The DUT required the following number of transmissions of each test sequence before it would receive a frame:
   1) Required 3 /I/ ordered_sets transmissions
   2) Required 3 /I/ ordered_sets transmissions
   3) Required 3 /I/ ordered_sets transmissions
   4) Required 3 /I/ ordered_sets transmissions
   5) Required 2 repetitions (plus one /I/ ordered_set)
   6) Wasn't observed to acquire synchronization
   7) Required 2 repetitions (plus one /I/ ordered_set)
   8) Required 2 repetitions (plus one /I/ ordered_set)
   9) Required 2 repetitions (plus one /I/ ordered_set)
   10) Required 2 repetitions (plus one /I/ ordered_set)

b) The DUT was not observed to acquire synchronization upon the reception of up to 10,000 transmissions of sequences 6. It appears that the DUT is unable to acquire synchronization from ordered sets beginning with /K28.1/ rather than /K28.5/. The DUT should be able to acquire synchronization from any ordered set beginning with a comma (not just /K28.5/).

c) The DUT only required the reception of three repetitions of sequences one through four and two repetitions (plus one /I/ ordered_set) of sequences five, and seven through ten in order to acquire synchronization and receive a frame. According to the standard, a device should require the reception of four consecutive Idle ordered_sets or three repetitions of sequences five through ten followed by an /I/ ordered_set when synchronization is lost in order to receive a frame.

Test # and Label	Part(s)	Result(s)
36.1.2 – Maintain Synchronization	**a**	**PASS**

Expected Results and Procedural Comments

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

a. The DUT is sent a test sequence followed by an ICMP request followed by the minimum inter-packet gap followed by another ICMP request. This ICMP request is sent to verify that the testing station is still on the DUT's ARP table. Part a is run for each of the following test sequences starting in an even code-group position directly after Idle:
   1) /K28.5/invalid/
   2) /K28.5/K28.5/
   3) /invalid/invalid/
   4) /invalid/K28.5/
   5) /K28.5/K28.5/invalid/K28.5/
   6) /K28.5/K28.5/invalid/invalid/
   7) /K28.5/invalid/invalid/K28.5/
   8) /K28.5/invalid/invalid/invalid/
   9) /K28.5/invalid/K28.5/invalid/K28.5/invalid/
   10) /K28.5/invalid/I/invalid/D16.2/K28.5/invalid/
   11) /K28.5/invalid/I/K28.5/invalid/I/K28.5/invalid/
   12) /invalid/invalid/invalid/D0.0/I/K28.5/invalid/

Note: invalid = /K28.5/ of the wrong running disparity for even code-groups.
invalid = /D0.0/ of the wrong running disparity for odd code-groups.
I = /I/ ordered_set which sets the running disparity to negative.

Comments on Test Results

a. The DUT was observed properly respond to frames preceded by the following sequences:
   1) /-K28.5/-D0.0/
   2) /-K28.5/+K28.5/
   3) /+K28.5/+D0.0/
   4) /-K28.5/-K28.5/
   5) /-K28.5/+K28.5/+K28.5/-K28.5/
   6) /-K28.5/+K28.5/+K28.5/+D0.0/
   7) /-K28.5/-D0.0/+K28.5/-K28.5/
   8) /-K28.5/-D0.0/+K28.5/+D0.0/
   9) /-K28.5/-D0.0/-K28.5/-D0.0/-K28.5/-D0.0/
   10) /-K28.5/+D0.0/+K28.5/-D5.6/+K28.5/-D16.2/+K28.5/+D0.0/
   11) /-K28.5/-D0.0/-K28.5/+D16.2/-K28.5/-D0.0/-K28.5/+D16.2/-K28.5/-D0.0/
   12) /+K28.5/+D0.0/-K28.5/+D0.0/+K28.5/-D5.6/-K28.5/-D0.0/

Each test sequence was preceded by a long stream of /I2/ ordered_sets (/-K28.5/+D16.2/)

Test # and Label	Part(s)	Result(s)
36.1.3 – Lose of Synchronization	**a**	**FAIL**
	b	**FAIL**

Expected Results and Procedural Comments

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

a. The DUT is sent a test sequence followed by an ARP request. The sequence, consisting of 200 /D0.0/ with the incorrect running disparity, should be long enough to ensure that the device is capable of losing synchronization.
b. Part a is run for each of the following test sequences starting in an even code-group position directly after Idle. Each test sequence is followed by one /I/ ordered_set and then the ICMP request.
    1) /K28.5/K28.5/invalid/K28.5/invalid/D0.0/
    2) /K28.5/K28.5/invalid/invalid/invalid/D0.0/
    3) /K28.5/invalid/invalid/K28.5/invalid/D0.0/
    4) /invalid/K28.5/invalid/K28.5/
    5) /invalid/invalid/invalid/K28.5/
    6) /invalid/K28.5/invalid/invalid/
    7) /invalid/invalid/invalid/invalid/
    8) /invalid/D0.0/invalid/D0.0/invalid/D0.0/invalid/D0.0/
    9) /invalid/D0.0/K28.5/invalid/K28.5/D0.0/invalid/D0.0/K28.5/invalid/
    10) /invalid/D0.0/K28.5/D0.0/invalid/D0.0/K28.5/D0.0/invalid/D0.0/K28.5/D0.0/invalid/D0.0/

Note: invalid = /K28.5/ of the wrong running disparity for even code-groups.
invalid = /D0.0/ of the wrong running disparity for odd code-groups.

Comments on Test Results

a) The DUT was not observed to lose synchronization with any sequence sent to it in part a. It is unlikely that the DUT will lose synchronization with any sequence sent in part b.
b) The DUT responded to all of the frames that should have been discarded. The frames were preceded by the following sequences:
    1) /-K28.5/+K28.5/+K28.5/-K28.5/-K28.5/+D0.0/
    2) /-K28.5/+K28.5/+K28.5/+D0.0/-K28.5/+D0.0/
    3) /-K28.5/-D0.0/+K28.5/-K28.5/-K28.5/-D0.0/
    4) /+K28.5/-K28.5/-K28.5/+K28.5/
    5) /+K28.5/+D0.0/-K28.5/+K28.5/
    6) /+K28.5/-K28.5/-K28.5/-D0.0/
    7) /+K28.5/+D0.0/-K28.5/-D0.0/
    8) /+K28.5/-D0.0/+K28.5/-D0.0/+K28.5/-D0.0/+K28.5/-D0.0/
    9) /+K28.5/-D0.0/-K28.5/-D0.0/-I2/+K28.5/-D0.0/-K28.5/-D0.0/
    10) /+K28.5/-D0.0/-I2/+K28.5/-D0.0/-I2/+K28.5/-D0.0/-I2/+K28.5/-D0.0/-I2/

Each test sequence was preceded by a long stream of /I2/ ordered_sets (/-K28.5/+D16.2/).

Test # and Label	Part(s)	Result(s)
36.1.4 – Fail to Acquire Synchronization	**a**	**PASS**

Expected Results and Procedural Comments

Purpose: To verify that a station in the LOSS_OF_SYNC state will not acquire synchronization if it receives packet sequences that should not allow it to acquire synchronization.

a. A manually configured link is established between the DUT and the testing station. The DUT is then sent 200 invalid code-groups (/D0.0/ of the wrong running disparity). The DUT is sent a test sequence 100 times followed by one /I/ ordered_set followed by an ARP request followed by a minimum inter-packet gap followed by another ARP request. Part a is run for each of the following test sequences starting in an even code-group position directly after Idle:
   1) /K28.5/invalid/
   2) /K28.5/K28.5/
   3) /K28.5/D0.0/invalid/
   4) /K28.5/D0.0/K28.5/invalid/
   5) /K28.5/D0.0/K28.5/K28.5/
   6) /K28.5/D0.0/K28.5/D0.0/invalid/
   7) /K28.5/D0.0/K28.5/D0.0/K28.5/K28.5/
   8) /K28.5/D0.0/K28.5/D0.0/K28.5/invalid/
   9) /K28.5/D2.2/D0.0/D0.0/K28.5/D21.5/D0.0/D0.0/K28.5/invalid/
   10) /K28.5/D0.0/D0.0/D0.0/D0.0/D0.0/D0.0/invalid/
   11) /K28.5/D0.0/D0.0/D0.0/D0.0/D0.0/K28.5/D0.0/D0.0/D0.0/D0.0/D0.0/K28.5/invalid/

Note: invalid = /K28.5/ of the wrong running disparity for even code-groups.
invalid = /D0.0/ of the wrong running disparity for odd code-groups.

Comments on Test Results

a) The DUT properly discarded all frames preceded by the sequences in part a (listed below). These sequences do not contain enough valid code-groups to cause the DUT to reach the SYNC_ACQUIRED_1 state, and therefore should be discarded.
   1) /-K28.5/-D0.0/
   2) /-K28.5/+K28.5/
   3) /-K28.5/+D0.0/-K28.5/
   4) /-K28.5/+D0.0/+K28.5/+D0.0/
   5) /-K28.5/+D0.0/+K28.5/-K28.5/
   6) /-K28.5/+D0.0/+K28.5/-D0.0/+K28.5/
   7) /-K28.5/+D0.0/+K28.5/-D0.0/-K28.5/+K28.5/
   8) /-K28.5/+D0.0/+K28.5/-D0.0/-K28.5/-D0.0/
   9) /-K28.5/+D2.2/-D0.0/-D0.0/-K28.5/+D21.5/+D0.0/+D0.0/+K28.5/+D0.0/
   10) /-K28.5/+D0.0/+D0.0/+D0.0/+D0.0/+D0.0/+D0.0/-D0.0/
   11) /-K28.5/+D0.0/+D0.0/+D0.0/+D0.0/+D0.0/+K28.5/-D0.0/-D0.0/-D0.0/-D0.0/-D0.0/-K28.5/-D0.0/

GROUP 2: Transmission

Test # and Label	Part(s)	Result(s)
36.2.1 – 8B/10B Encoding	**a**	**PASS**
	b	**Not Applicable**
	c	**Not Applicable**

Expected Results and Procedural Comments

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

a. Bring the DUT to the state where xmit=DATA. Once xmit=DATA, force the DUT to transmit a packet containing both forms of every valid data code-group listed in Table 36-1.
b. If the DUT is capable of packet bursting, force the DUT to transmit a burst of two or more packets. Ensure that the running disparity after the last byte of data in the first packet is positive.
c. If the DUT performs carrier extension, force the DUT to issue a packet that requires extension. Ensure that the running disparity after the last byte of data is negative. Instruct the testing station to collide with the packet while extension is being sent. Repeat, ensuring that the running disparity after the last byte of data issued by the DUT is negative.

Comments on Test Results

a) The DUT properly replied to the frame containing both running disparity values of every code-group found in Table 36-1. There were no running disparity or CRC errors within the frame sent by the DUT.
b) This test was not performed because the DUT does not support Half Duplex MAC operation.
c) This test was not performed because the DUT does not support Half Duplex MAC operation.

Test # and Label	Part(s)	Result(s)
36.2.2 – Idle Generation	**a**	**PASS**
	b	**PASS**

Expected Results and Procedural Comments

Purpose: To verify that the first /I/ ordered_set following the EPD or /C/ ordered_set ensures that the running disparity is negative

a. The first /I/ ordered_set following a packet with a positive ending running disparity should be /I1/. This /I1/ should be followed by /I2/ ordered_sets until the next packet is transmitted.
b. The first /I/ ordered_set following a packet with a negative ending running disparity should be /I2/. This /I2/ should be followed by /I2/ ordered_sets until the next packet is transmitted.

Comments on Test Results

a) The DUT was instructed to transmit even and odd sized Ethernet packets with a positive ending running disparity. The transmission of the DUT was monitored for three minutes and it was never observed to send anything but one /I1/ ordered_set followed by /I2/ ordered_sets after the EPD.
b) The DUT was instructed to transmit even and odd sized Ethernet packets with a negative ending running disparity. The transmission of the DUT was monitored for three minutes and it was never observed to send anything but /I2/ ordered_sets after the EPD.

Test # and Label	Part(s)	Result(s)
36.2.3 – Idle Alignment	**a**	**PASS**
	b	**PASS**

Expected Results and Procedural Comments

Purpose: To verify that the device under test (DUT) transmits the correct number of /R/ code-groups so that /I/ begins in an even code-group position.

a. If a packet ends such that the /T/ ordered_set must be transmitted in an even code-group position, it should be followed by only one /R/ ordered_set and then Idle.
b. If a packet ends such that the /T/ ordered_set must be transmitted in an odd code-group position, it should be followed by two /R/ ordered_sets and then Idle.

Comments on Test Results

a) The DUT was instructed to send even sized Ethernet packets with both positive and negative ending running disparity. The transmission of the DUT was monitored for three minutes and it was never observed to end a packet with anything but /T/R/.
b) The DUT was instructed to send odd sized Ethernet packets with both positive and negative ending running disparity. The transmission of the DUT was monitored for three minutes and it was never observed to end a packet with anything but /T/R/R/.

Test # and Label	Part(s)	Result(s)
36.2.4 – /C/ Transmission Order	**a**	**Refer to Comments**
	b	**Refer to Comments**

Expected Results and Procedural Comments

Purpose: To verify that device under test (DUT) transmits /C/ ordered_sets as alternating /C1/ and /C2/ ordered_sets.

a. When the DUT is set to auto-negotiate and the receiver of the DUT is receiving no signal it is expected to transmit /C/ ordered_sets containing /D0.0/ in its both of its configuration registers. These /C/ ordered_sets should be transmitted as alternating /C1/ and /C2/ ordered_sets.
a. When the DUT is set to auto-negotiate and the receiver of the DUT is receiving /I/ ordered_sets, the DUT is expected to transmit Break Link for a period of link timer and then constantly transmit /C/ ordered_sets containing its abilities. All /C/ ordered_sets should be transmitted as alternating /C1/ and /C2/ ordered_sets. The DUT should maintain this alternating pattern while it transitions from Break Link to its abilities.

Comments on Test Results

This test was performed during Auto-Negotiation testing (Test 37.1.1(a)) and the results can be found in that report.

GROUP 3: Reception

Test # and Label	Part(s)	Result(s)
36.3.1 – 8B/10B Decoding	**a**	**Not Available**
Expected Results and Procedural Comments		
Purpose: To verify that the device under test (DUT) can distinguish between valid and invalid code-groups. a. Bring the DUT to the state where xmit=DATA. Once xmit=DATA, instruct the testing station to transmit a three packet sequence where each packet is separated by the minimum inter-packet gap. The first and third packets shall be valid echo request packets. The second packet shall be a valid echo request packet with one valid code-group substituted with an invalid one. Repeat step 2 until every invalid code-group has been substituted for every valid code-group (both positive and negative running disparity encodings).		
Comments on Test Results		
This test is currently under development.		

Test # and Label	Part(s)	Result(s)
36.3.2 – Carrier Event Handling	**a**	**PASS**
	b	**FAIL**
	c	**FAIL**

Expected Results and Procedural Comments

Purpose: To verify that the device under test (DUT) detects carrier events and handles them properly.

a. Instruct the testing station to transmit a two-packet sequence. The first packet shall be a valid echo request packet preceded by a two code-group sequence. The first code-group shall be a code-group other than /S/ and the second code-group shall be any data code-group other than /D21.5/ or /D2.2/. This code-group sequence should be considered part of the inter-packet gap. Repeat until /K28.5/ has been replaced with every code-group having a 2-bit difference from /K28.5/ of the negative running disparity. Any code-group with a two or more bit difference from /K28.5/ causes carrier_detect=TRUE. When such a code group is received in the IDLE_D state, the PCS receive state diagram (part a) should transition to the CARRIER_DETECT state and then move directly to the FALSE_CARRIER state where it waits for the reception of /K28.5/ in an even code-group position. The second packet shall be a valid echo request packet.

b. Instruct the testing station to transmit a two-packet sequence. The first packet shall be a valid echo request packet preceded by a two code-group sequence. The first code-group shall be a code-group other than /S/ that is 1-bit different from /K28.5/ and the second code-group shall be any data code-group other than /D21.5/ or /D2.2/, this code-group sequence should be considered part of the inter-packet gap. Repeat until /K28.5/ has been replaced with every code-group having a 1-bit difference from /K28.5/ of the negative running disparity. The second packet shall be a valid echo request packet.

c. Instruct the testing station to transmit a two-packet sequence. The first packet shall be a valid echo request packet preceded by a two code-group sequence. The first code-group shall be /K28.5/ and the second code-group shall be any valid code-group other than /D16.2/, /D5.6/, /D21.5/ or /D2.2/, this code-group sequence should be considered part of the inter-packet gap. The second packet shall be a valid echo request packet. Due to time restrictions, every valid code-group cannot be used to replace /D16.2/, so /D16.2/ is replaced with the following code-groups:
    1) /D5.6/
    2) /D6.6/
    3) /D10.1/
    4) /D3.3/
    5) /D27.7/
    6) /D3.0/
    7) /D30.2/
    8) /D12.4/
    9) /D8.6/
    10) /D13.7/
    11) /S/
    12) /T/
    13) /R/
    14) /V/

Comments on Test Results

a) The DUT properly discarded all of the frames preceded by the sequences as described in part a.
b) The DUT properly failed to respond to any of the echo requests preceded by the sequences of code-groups where /K28.5/ is replaced with a code group with a 1 bit difference. Since these sequences should be considered part of the inter-packet gap, the DUT fails this part.
c) The DUT properly failed to respond to any of the echo requests preceded by the sequences of code-groups where either /D21.5/ or /D2.2/ is replaced with a code groups 11-14. Since these sequences should be considered part of the inter-packet gap, the DUT fails this part.

Test # and Label	Part(s)	Result(s)
36.3.3 – Detecting End of Packet	**a**	**PASS**
	b	**PASS**

Expected Results and Procedural Comments

Purpose: To verify that the device under test (DUT) can distinguish valid EPDs from invalid EPDs and detect the premature end of a packet.

a. The DUT is sent a repeating two-packet sequence consisting of even sized packets with 8 bytes of preamble. The first packet contains a valid echo request to the DUT and the second packet is the same echo request packet except that it has an EPD consisting of:
    1. /T/D0.0/
    2. /T/R/D0.0/D0.0/
    3. /R/R/R/D0.0/I/
    4. /I/I/
    5. /K28.5/D21.5/D0.0/D0.0/
    6. /K28.5/D2.2/D0.0/D0.0/

b. The DUT is sent a repeating two-packet sequence consisting of odd sized packets with 8 bytes of preamble. The first packet contains a valid echo request to the DUT and the second packet is the same echo request packet except that it has an EPD consisting of:
    1. /T/R/K28.5/
    2. /T/D0.0/R/
    3. /T/R/D0.0/
    4. /R/R/R/

Comments on Test Results

a) The DUT properly discarded the even sized packets that had bad EPD's.
b) The DUT properly discarded the odd sized packets that had bad EPD's.

Such frames should be received as CRC errors by the DUT. The DUT failed to increment the "CRC" error counter for any of the frames with bad EPDs. The reception of these frames did not affect the reception of frames with good EPDs.

Test # and Label	Part(s)	Result(s)
36.3.4 – Reception of /C/ during IDLE	**a**	**Refer to Comments**

Expected Results and Procedural Comments

Purpose: To verify that the device under test (DUT) detects carrier events and handles them properly.

Configure the DUT to auto-negotiate. Force the testing station to send a repeating pattern consisting of 100 /C/ ordered_sets, with the ACK bit set, followed by 100 /I/ ordered_sets. The /C/ ordered_sets are configured to be consistent with the abilities of the DUT and are not configured to be break-link. The DUT is expected to reach the LINK_OK state in the Auto-Negotiation state diagram based on this repeating sequence. Once the DUT reaches the LINK_OK state, it should get an ability match (ability_match=TRUE) and restart Auto-Negotiation. The test is run three times by sending the /C/ ordered_sets the following ways:

a) /C1/C2/C1/C2/
b) /C1/C1/C1/C1/
c) /C2/C2/C2/C2/

Comments on Test Results

This test was performed during Auto-Negotiation testing (Test 37.1.1(a)) and the results can be found in that report.

Annex A: Test Setup 1

Test Equipment

The following test equipment was used in performing all Clause 36 PCS testing:

Testing Equipment	Brand and Version Information
PC Requirements	Win2K with LabVIEW 7.1 and a GPIB interface
Software	SmartWindows 8.00.162, UNH-IOL TIGER System Software, UNH-IOL Custom PCS Testing software v3.0
Logic Analyzer	HP 16500B with Logic Analyzer module 16555A and Pattern Generator module 15552A
Traffic Generator/Sniffer	SMB 2000 Chassis with GX-1405B 1000BASE-SX module
TIGER Board with Cub	Custom Testing System designed by UNH-IOL in collaboration with Texas Instruments
Splitter	Two AMP Multimode Splitters: 2-107842-3

Test Configuration

The following configuration was used in performing all Clause 36 PCS testing: